2014 IEEE Congress on Evolutionary Computation (CEC)
July 6-11, 2014, Beijing, China

Lion Algorithm for Standard and Large Scale Bilinear System Identification: A Global Optimization based on Lion's Social Behavior

B R Rajakumar, *Member, IEEE*

***Abstract*—Nonlinear system identification process, especially bilinear system identification process exploits global optimization algorithms for betterment of identification precision. This paper attempts to introduce a new optimization algorithm called as Lion algorithm to accomplish the system characteristics precisely. Our algorithm is a simulation model of the lion's unique characteristics such as territorial defense, territorial takeover, laggardness exploitation and pride. Experiments are conducted by identifying a nonlinear rationale digital benchmark system using standard bilinear model and comparisons are made with prominent genetic algorithm and differential evolution. Subsequently, curse of dimensionality is also experimented by defining a large scale bilinear model, i.e. bilinear system with 1023 bilinear kernel models, to identify the same digital benchmark system. Lion algorithm dominates when using standard bilinear model, whereas it is equivalent to differential evolution and better than genetic algorithm when using large scale bilinear model.**

Keywords— Lion Algorithm (LA); bilinear system; system identification; territorial defense; territorial takeover.

I. Introduction

NONLINEAR system identification is a vital process to forecast the output of all the real world control systems, signal processing systems, chemical processing systems, etc [1-6]. Differential equations are suggested to be promising solutions for a simple system with well-known physical properties. Nevertheless, such systems are unrealistic and rare in practice because most of the times, only input and output signals are available for system identification [7]. Hence, numerous auto-regression models [8-10], auto-regression models with exogenous variables (e.g. ARX, ARMAX, NARMAX, etc) [11-15] and other nonlinear models such as Polynomial models [21, 22], Winner-Hammerstein models [20], Volterra series [16, 17, 19] and bilinear series [18] were reported in the literature. Bilinear series is an extended form of Infinite Impulse Response (IIR) filter. It was initially formulated as a non-recursive structure (no kernel model), but later, it has been framed as recursive structures (with kernel models for precise system identification [7].

Numerous bilinear system identification methods have been reported in the literature in the recent past [27-30]. For example, Least Mean Square algorithm (LMS) and its variants [31], blind identification methods and many more [27-29]. In the past decade, global optimization algorithms have also been reported for bilinear system identification, among which Genetic Algorithm (GA) [35] and Differential Evolution (DE) [7, 32-34] are the most prominent algorithms.

This paper makes an attempt to introduce a novel optimization algorithm, which is based on lion's unique social behaviour, hence termed as Lion Algorithm (LA), for bilinear system identification. LA introduced in this paper is an extension of our previous work [37] in which a simple LA model was proposed to solve a benchmark minimization problem. Hence, the main contributions of the paper can be pinpointed as follows

- Introduces a new optimization algorithm based on lion's social behaviour
- Applies the optimization algorithm for nonlinear system identification through bilinear series model
- Expands the system identification problem as a Large scale global optimization problem, which is further referred as large scale system identification problem
- Studies the performance competency of the algorithms when they are subjected to solve large scale system identification problem.

The rest of the paper is organized as follows. Section II gives preliminaries about the bilinear series model and LA. Section III and IV details the system identification procedure and the simulations of steps of LA, respectively with required illustrations and mathematical models. Section V discusses the experimental results and Section VI concludes the paper.

II. Preliminaries

A. Bilinear Series Model

Bilinear model is one of the classes of nonlinear models, which has highly attracted researchers for deriving models for various real-time systems [19-25]. Given an input sequence, $u_n : n = 0,1,\ldots,T-1$, the output y_n from a bilinear series model can be given by Eq. (1), in which $\{a_1, a_2, \ldots, a_N \; b_0, b_1, \ldots, b_M, c_{0,1}, c_{0,2}, \ldots, c_{M,N}\}$ is the set of kernels to be identified.

$$y_n = \sum_{k=1}^{N} a_k y_{n-k} + \sum_{k=0}^{M} b_k u_{n-k} + \sum_{k_1=0}^{M} \sum_{k_2=1}^{N} c_{k_1 k_2} u_{n-k_1} y_{n-k_2} \qquad (1)$$

where, N and M are the number of past outputs and inputs, respectively. The cardinality of the kernel set, $|K|$ can be derived as

$$|K| = 1 + N[M+2] + M \quad (2)$$

B. Lion Algorithm (LA)

In 2012, LA (termed as lion's algorithm) was introduced by us [37] based on the raw inspiration from lion's unique social behaviour [36]. This can be interpreted in algorithmic perspective as follows

- A solution (territorial lion) should be strong enough to defeat a random solution (nomadic lion)
- The weak solutions (weak lions, may be cubs also) are either vanished from the solution pool or driven out from the solution pool
- A solution, which is derived from the successful solution (succeeded lion in territorial defense or territorial takeover) is stronger than a solution, which is derived from failed solution (laggard lion in territorial defense or territorial takeover)

However, we have restructured the basic model of the algorithm proposed in [37] along with the introduction of fertility evaluation phase. We have also changed crossover operation and gender clustering method with subsequent strengthening in the survival fight. The restructured version of the proposed LA is illustrated in Fig. 1. However, the algorithm steps are detailed in the Section IV.

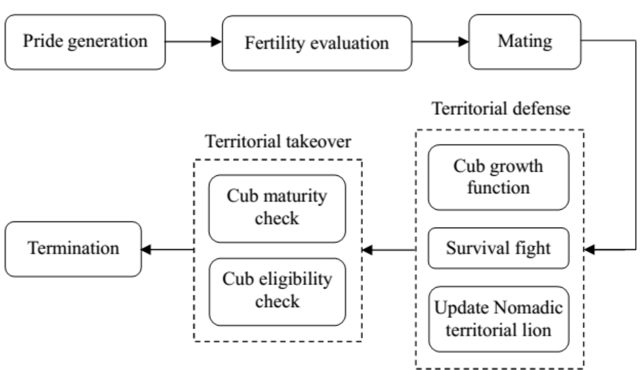

Fig 1. Major blocks/functions involved in the proposed LA

III. System Identification Procedure

The system identification method using LA is illustrated in Fig 2. It interprets that the LA tunes the kernel models based on the output variation between the actual system and the bilinear model. Hence, the system identification procedure can be modelled as a minimization problem as follows

$$K^{opt} = \arg\min_{K} \frac{1}{T}\left|\sum_{n=0}^{T-1}[A_n - y_n]\right| \quad (3)$$

where, K^{opt} is the optimal set of kernel models, y_n is the bilinear system model as given in Eq. (1) and A_n is the actual system to be modelled. LA attempts to solve the minimization problem and determines the optimal kernel models.

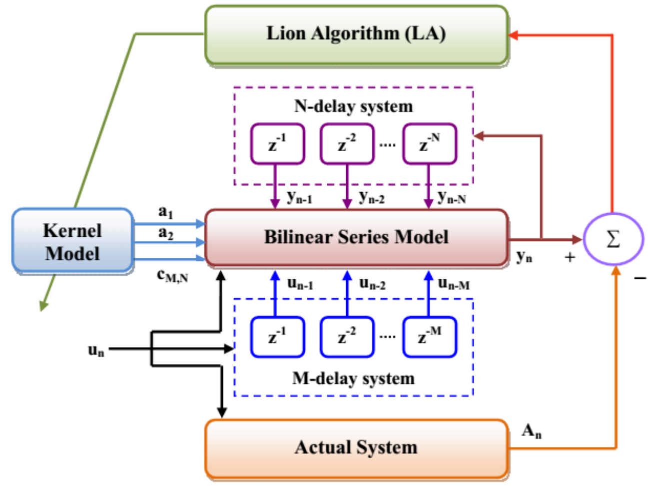

Fig 2. Bilinear System Identification Process using LA

IV. Simulation Steps

Step 1: Pride Generation

Initialize X^{male}, X^{female} and X_1^{nomad} of which X^{male} and its lioness X^{female} constitute pride. The lions interpret the solution vectors i.e. set of kernel models K. The elements of X^{male}, X^{female} and X_1^{nomad}, i.e., $x^{male}(l)$, $x^{female}(l)$ and, $x_1^{nomad}(l)$ respectively are arbitrary integers generated within the minimum and maximum limits, where, $l = 1, 2, \ldots, L$. Here, L refers number of kernel models to be optimized, i.e., $|K|$. Here, one of the two nomadic lions is initialized while the other nomadic lion will be initialized at the time of territorial defense only.

Step 2: Fitness Evaluation

The fitness of all the three lions, termed as $f(X^{male})$, $f(X^{female})$ and $f(X_1^{nomad})$, are determined using Eq. (1) followed by the error calculation in Eq. (3). For the further steps, we set $f^{ref} = f(X^{male})$ and $N_g = 0$, where N_g is the generation counter, which may be used for checking termination criterion. Also, we store X^{male} and $f(X^{male})$

Step 3: Fertility Evaluation

This stage evaluates and ensures the fertility of the territorial lion and lioness. In other words, the stage intends to avoid converging in local optima. The processing steps are discussed in the pseudo code given in Fig. 3.

Process: Fertility Evaluation
Input: X^{male}, X^{female}, f^{ref}, L_r and S_r
Output: X^{male}, X^{female}, f^{ref}, L_r and S_r
// X^{male} Evaluation

```
If f^{ref} ≤ f(X^{male})
        L_r ← L_r + 1
else
        Reset L_r
        f^{ref} ← f(X^{male})
End if
// X^{female} evaluation
If S_r is not tolerable
        Set u_c and g_c to zero
        Do
                Calculate X^{female+}
                g_c ← g_c + 1
                If f(X^{female+}) < f(X^{female})
                        u_c ← 1
                        X^{female} ← X^{female+}
                        Reset S_r
                End if
        Until g_c reaches g_c^{max}
End if
```

Fig. 3. Pseudo code for Fertility evaluation

In Fig. 3, $X^{female+}$, f^{ref}, L_r, S_r, u_c and g_c are updated female lion, reference fitness, Laggardness rate, sterility rate, female update count and female generation count respectively. When the LA begins, L_r and S_r are initialized as zero, and at every call for fertility evaluation, L_r and S_r get lastly determined value. Checking the tolerance of S_r is nothing but checking whether they exceed their maximum limit $S_r^{\max}$. $S_r^{\max}$ is set as four as the median duration of estrus is four days for lions [38]. However after some trial and error evaluation, we finalized this as three. In order to avoid conflict and to maintain uniformity, we include both $L_r^{\max}$ (will be referred in Step 6) and $S_r^{\max}$ irrespective of gender. $g_c^{\max}$ is set as 10 based on trial and error method (Results are affixed in Section V). $X^{female+}$ ($\supset x_l^{female+}$) can be calculated as

$$x_l^{female+} = \begin{cases} x_k^{female+}; if\ l = k \\ x_l^{female}; otherwise \end{cases} \quad (4)$$

$$x_k^{female+} = \min\left[x_k^{\max}, \max\left(x_k^{\min}, \nabla_k\right)\right] \quad (5)$$

$$\nabla_k = \left[x_k^{female} + (0.1r_2 - 0.05)\left(x_k^{male} - r_1 x_k^{female}\right)\right] \quad (6)$$

where, $x_l^{female+}$ and $x_k^{female+}$ are the l^{th} and k^{th} vector elements of $X^{female+}$, respectively, k is a random integer generated within the interval $[1, L]$, ∇ is the female update function, r_1 and r_2 are random integers generated within the interval $[0,1]$.

Step 4: Mating

In the mating, we include two primary steps called as crossover and mutation, which are found as significant operators for any evolutionary optimization [39-42]. We follow the maximum natural littering rate, i.e., four cubs (mostly) in a lioness pregnancy [43] and so we get four cubs 'X^{cubs}' from the crossover, which is of uniform in nature with random crossover probability C_r. The mathematical representation of the crossover operation can be given as

$$X^{cubs}(p) = B_p \circ X^{male} + \overline{B}_p \circ X^{female} \ : \ p = 1,2,3,4 \quad (7)$$

where, B is crossover mask of length L in which 1s and 0s are filled randomly based on C_r, $\overline{B}$ is the one's complement of B, vector operator '$\circ$' represents Hadamard product or schur product and $X^{cubs}(p)$ is the p^{th} cub obtained from crossover. The X^{cubs} are subjected to uniform mutation with the mutation probability as M_r and hence equal number of new cubs X^{new} are obtained. The obtained X^{new} (from mutation) and X^{cubs} (from crossover) are placed in the cub pool and subjected to further processes.

A secondary step, called gender clustering [37], is also included here to extract a single male cub and a female cub from the cub pool. Based on the lion's physical nature [44-47], we select the cubs, which have the first and second best fitness, as the male cub X^{m_cub} and female cub X^{f_cub}, respectively. Once the X^{m_cub} and X^{f_cub} are obtained, set their ages (commonly referred as cubs' age) A_{cub} as zero.

Step 5: Cub Growth Function

Cub growth function is a local solution search function in which the X^{m_cub} and X^{f_cub} are subjected to uniform random mutation at a rate of G_r. If the mutated X^{m_cub} and X^{f_cub} are better than old X^{m_cub} and X^{f_cub}, then the mutated X^{m_cub} and X^{f_cub} replace old X^{m_cub} and X^{f_cub}, respectively. A_{cub} is incremented by one at every execution of cub growth function to illustrate the cubs' growth towards maturity. G_r should be less enough (possibly <0.2) to search the fine local solutions of X^{m_cub} and X^{f_cub} and need not be equal to M_r.

Step 6: Territorial Defense

Territorial defense [37] is one of the primary lion operators to direct the algorithm to analyze the search space in a wider way. The territorial defense can be sequenced here as forming nomad coalition, survival fight and then pride and nomad coalition updates. Pseudo code for Territorial defense is given in Fig. 4.

Process: Territorial Defense
Get nomad coalition
    Select x^{e_nomad}
    if x^{e_nomad} wins
        $x^{male} \leftarrow x^{e\_nomad}$
        Remove x^{e_nomad} from nomad world
        Kill x^{m_cub} & x^{f_cub}
        Reset age(cubs)
        Defense result←1
    Else
        Update nomad coalition
        Defense result←0
    End if

Fig. 4. Pseudo code for Territorial defense

Getting nomad coalition as given in Fig. 4 represents introducing two nomadic lions of which X_1^{nomad} has been initialized at Step 1, whereas X_2^{nomad} is initialized as based on L_r. Succinctly, X_2^{nomad} is initialized as like X_1^{nomad}, when X^{male} is not laggard (i.e., $L_r \le L_r^{\max}$). Otherwise, X_2^{nomad} is initialized as an updated version of X^{male} through uniform mutation (as described in Step 4), with a mutation rate of $1 - M_r$.

Survival fight takes place between one of the two lions of nomad coalition and the pride, despite coalition between nomadic lions are also common [48]. Instead of engaging nomad coalition for territorial defense, we apply winner take all approach [49] so that only winning nomadic lion X^{e_nomad} among the coalition engage in territorial defense. The survival fight result comes in favour of the selected X^{e_nomad} if the following criteria are met

$$f\left(X^{e\_nomad}\right) < f\left(X^{male}\right) \tag{8}$$
$$f\left(X^{e\_nomad}\right) < f\left(X^{m\_cub}\right) \tag{9}$$
$$f\left(X^{e\_nomad}\right) < f\left(X^{f\_cub}\right) \tag{10}$$

Pride is updated by replacing X^{male} by X^{e_nomad} after removing it from nomad coalition, which happens only when X^{male} is defeated in the territorial defensre. Likely, nomad coalition is updated only when X^{e_nomad} is defeated. The update process is done by selecting only one X^{nomad}, which has E^{nomad} greater than or equal to exponential of unity. For instance, X_1^{nomad} is selected if $E_1^{nomad} \ge e$, X_2^{nomad} otherwise, where, E_1^{nomad} can be calculated as follows

$$E_1^{nomad} = \exp\left(\frac{d_1}{\max(d_1, d_2)}\right)\frac{\max\left(f\left(X_1^{nomad}\right), f\left(X_2^{nomad}\right)\right)}{f\left(X_1^{nomad}\right)} \tag{11}$$

In Eq. (11), d_1 is the Euclidean distance between X_1^{nomad} and X^{male}, d_2 is the Euclidean distance between X_2^{nomad} and X^{male}. If the defense result is zero, X^{male} and $f\left(X^{male}\right)$ are stored and the process is reiterated from Step 3.

Step 7: Territorial Takeover

Territorial takeover takes place only if $A_{cub} \ge A_{\max}$, otherwise the process is reiterated from Step 5. It is a process of giving territory to the X^{m_cub} and X^{f_cub} after they mature and become stronger than X^{male} and X^{female}. The pseudo code, which is given in Fig. 5, depicts the processing steps of the territorial takeover operation. If X^{f_cub} is found to be better than X^{female}, X^{f_cub} retains X^{female} position. Such X^{f_cub} is probably fertile and hence S_r is set back to zero in the territorial takeover. Here $A_{\max}$ is associated with attaining sexual maturity by cubs. Naturally, cubs mature in 1.5-2 years [38] or 2-4 years [50]. We refer the later, and hence we fix the median as maximum age for cub maturity, i.e., $A_{\max} = 3$. Now, one generation shall be considered as completed and hence is N_g incremented by one.

Process: Territorial Takeover
If $f\left(X^{male}\right) > f\left(X^{m\_cub}\right)$
    $X^{male} = X^{m\_cub}$
Endif
$X^{old} = X^{female}$
If $f\left(X^{female}\right) > f\left(X^{f\_cub}\right)$
    $X^{female} = X^{f\_cub}$
Endif
If $X^{female} \ne X^{old}$
    Clear S_r
Endif

Fig. 5. Pseudo code for Territorial takeover

Step 8: Termination Criteria

Here, the algorithm execution is terminated when any one of the following two termination criteria is met, otherwise the process is reiterated from Step 3, after storing X^{male} and $f\left(X^{male}\right)$.

$$f\left(X^{male}\right) \leq e_T \tag{12}$$

$$N_f > N_f^{\max} \tag{13}$$

where, N_f is the number of function evaluations, which is also initialized with zero and incremented by one, when a fitness evaluation is performed, $N_f^{\max}$ and e_T are the maximum number of function evaluations and target error, respectively.

V. Experimental Results

The algorithm is executed in MATLAB R2013a and experimented in a PC with Intel Pentium Dual Core Processor @ 2.13GHz Clock speed, 2 GB RAM and Windows 7 OS. An extremely nonlinear rationale digital system presented by Y. Liu *et. al.* [57] as given in Eq. (14) is considered.

$$y_n = \frac{y_{n-1}y_{n-2}y_{n-3}\left[y_{n-3}-1\right]u_{n-2}+u_{n-1}}{1+y_{n-2}^2+y_{n-3}^2} \tag{14}$$

This exemplary system takes the sinusoidal signal as input, which is represented as

$$u_n = 0.5\sin\left(\frac{n\pi}{125}\right)+0.2\sin\left(\frac{2n\pi}{25}\right); 0 \leq n \leq 500 . \tag{15}$$

The experimental investigation is carried out by setting the bilinear series model parameters M and N as *4* and *5*, respectively, as recommended in [7]. Hence, $|K|=35$,i.e., the dimension of the optimization problem becomes *35*. Error deviation between the actual system output and the identified system output for *50* test executions are noted along with the number of function evaluations. For all those measures, the best and worst among every measure, mean, median and standard deviation are calculated and tabulated in Table I.

However, many real world problems have high dimensional parameters to be optimized, which results in curse of dimensionality [51, 52]. Hence, in our second experiment, we investigate the system identification problem as a large scale optimization problem, as already referred as large scale system identification. The parameters M and N defines the number of kernel models to be optimized as illustrated in Fig. 6. A large scale system identification problem can be formulated when $M, N > 31$, because the number of kernel models to be optimized becomes greater than 1000. Conventional meta-heuristic search algorithms [53-55] have been reported as lagging to solve large scale optimization problems because of its degenerating performance when search space dimension increases [56]. Hence, we further subjected LA to solve this large scale system identification problem by setting *M=31* and *N = 32*, so that $|K| = 1023$. This means that the number of kernel models to be optimized is *1023* and so the length of the lion, $L = 1023$. For simple observation, five executions are made in this investigation and similar measures (as mentioned in Table I) are tabulated in Table II.

TABLE I
SYSTEM IDENTIFICATION PERFORMANCE OF GA, DE AND LA USING STANDARD BILINEAR MODEL

Performance measures	**|Error|**			**Number of function evaluations**		
Methods	**GA (Rank)**	**DE (Rank)**	**LA (Rank)**	**GA (Rank)**	**DE (Rank)**	**LA (Rank)**
Target	1×10^{-6}	1×10^{-6}	1×10^{-6}	1×10^{5}	1×10^{5}	1×10^{5}
Best	0.00251 (3)	0.002218 (1)	0.002373 (2)	100005 (2)	100020 (3)	100002 (1)
Worst	0.006018 (1)	0.016543 (3)	0.006881 (2)	100005 (1)	100020 (2)	100032 (3)
Mean	0.00391 (2)	0.005767 (3)	0.003892 (1)	100005 (1)	100020 (3)	100015.9 (2)
Median	0.003739 (2)	0.004652 (3)	0.003573 (1)	100005 (1)	100020 (3)	100018 (2)
Standard deviation	0.000823 (1)	0.003423 (3)	0.001009 (2)	0 (1)	0 (1)	8.153421 (3)
Average Rank	1.8	2.6	1.6	1.2	2.4	2.2
Final Rank	**2**	**3**	**1**	**1**	**3**	**2**

TABLE II
SYSTEM IDENTIFICATION PERFORMANCE OF GA, DE AND LA USING LARGE SCALE BILINEAR MODEL

Performance measures	**|Error|**			**Number of function evaluations**		
Methods	**GA (Rank)**	**DE (Rank)**	**LA (Rank)**	**GA (Rank)**	**DE (Rank)**	**LA (Rank)**
Target	1×10^{-6}	1×10^{-6}	1×10^{-6}	1×10^{5}	1×10^{5}	1×10^{5}
Best	4.32631 (3)	0.270486 (1)	2.56987 (2)	100005 (1)	100020 (3)	100008 (2)
Worst	7.52724 (3)	5.08779 (2)	3.54709 (1)	100005 (1)	100020 (2)	100021 (3)
Mean	5.641038 (3)	1.676307 (1)	2.93411 (2)	100005 (1)	100020 (3)	100015.2 (2)
Median	5.22757 (3)	0.966493 (1)	2.74654 (2)	100005 (1)	100020 (3)	100018 (2)
Standard deviation	1.365057 (2)	1.935896 (3)	0.429375 (1)	0 (1)	0 (1)	6.685806 (3)
Average Rank	2.8	1.6	1.6	1	2.4	2.4
Final Rank	**3**	**1**	**1**	**1**	**2**	**2**

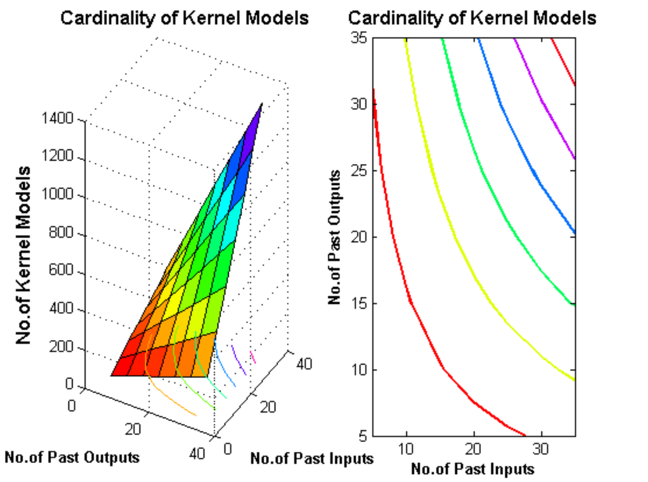

Fig. 6. Increase in no. of kernel models to be optimized with respect M and N, which are no. of past inputs and outputs, respectively. No. of kernel models crosses *1000* , when M and N are greater than *30*.

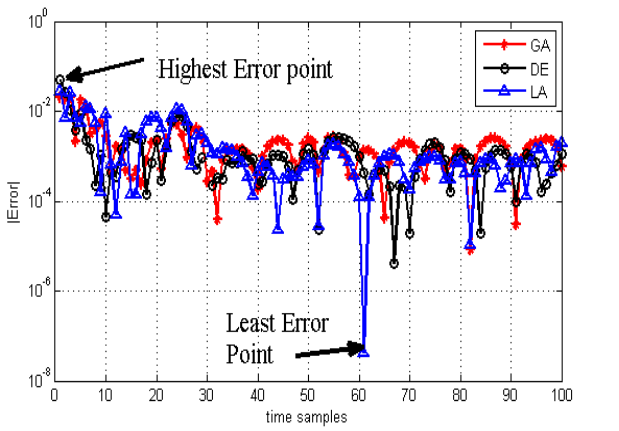

Fig. 7. Error absolute calculated between the actual system and the system identified by the algorithms. It depicts best and worst error minimization at every time sample along with the cumulative minimization capability throughout the 100 time samples.

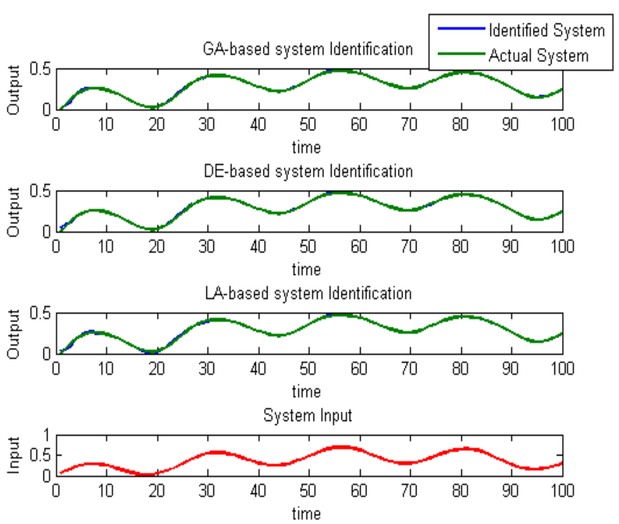

Fig. 8. Output characteristics of bilinear systems identified by global optimization algorithms versus actual system and the corresponding input.

However, many real world problems have high dimensional parameters to be optimized, which results in curse of dimensionality [51, 52]. Hence, in our second experiment, we investigate the system identification problem as a large scale optimization problem, as already referred as large scale system identification. The parameters M and N defines the number of kernel models to be optimized as illustrated in Fig. 6. A large scale system identification problem can be formulated when $M, N > 31$, because the number of kernel models to be optimized becomes greater than 1000. Conventional meta-heuristic search algorithms [53-55] have been reported as lagging to solve large scale optimization problems because of its degenerating performance when search space dimension increases [56]. Hence, we further subjected LA to solve this large scale system identification problem by setting *M=31* and *N = 32*, so that $|K| = 1023$. This means that the number of kernel models to be optimized is *1023* and so the length of the lion, $L = 1023$. For simple observation, five executions are made in this investigation and similar measures (as mentioned in Table I) are tabulated in Table II.

A. Discussion

LA is operated by various parameters such as $S_r^{\max}$, $L_r^{\max}$, etc, for better searching. As stated earlier, those parameters get values from the biological motivation, except $g_c^{\max}$. In Section IV, $g_c^{\max}$ was said as set through trial and error method. The illustration, which is given in Fig. 9, could better justify that *10* is the best fit value for $_c$, when experimenting on standard bilinear model with $g_c^{\max}$ varied between *1* to *25*. Coming to the experimental investigation, Table I demonstrates that LA dominates over the other two algorithms in terms of error minimization. However, number of function evaluations carried out by LA is more than GA, but less than DE. Please note that the ranks are based on five measures such best, worst, mean and median error convergences and the standard deviation of convergences in the *50* test executions. An illustration for error deviation is given in Fig. 7 to depict the absolute error between the actual system and identified system at every time sample. Least error point marked here should not be confused with the best error value (0.00251) mentioned in Table I, because values in Table I are the average of error absolute obtained at all the time samples (here, 100 samples). The system characteristics is plotted against the input and system identified by the algorithms along with the actual system in Fig. 8. These two illustrations construe the nature of the algorithms on accomplishing error minimization.

From the summary of Table II, we can say that GA fails to minimize the error in all the cases (best, worst, mean and median performances). LA and DE compete equally if we observe LA domination in consistent error minimization and relatively less worst convergence, whereas DE accomplished best convergence among the five executions, average and median error accomplishment. However, DE is found to be the last in minimizing the error consistently. The eventual ranking system based on the average ranks reveals LA and DE are equivalent to each other, but better than GA. On the other hand, GA consumes relatively less number of function evaluations, whereas LA and DE are equivalent to each other. Nevertheless, LA is not constant on evaluating fitness functions. As a performance remark based on only mean and median of Table I and Table II, LA is found to be first and second dominating algorithm for system identification using standard and large scale bilinear model, respectively.

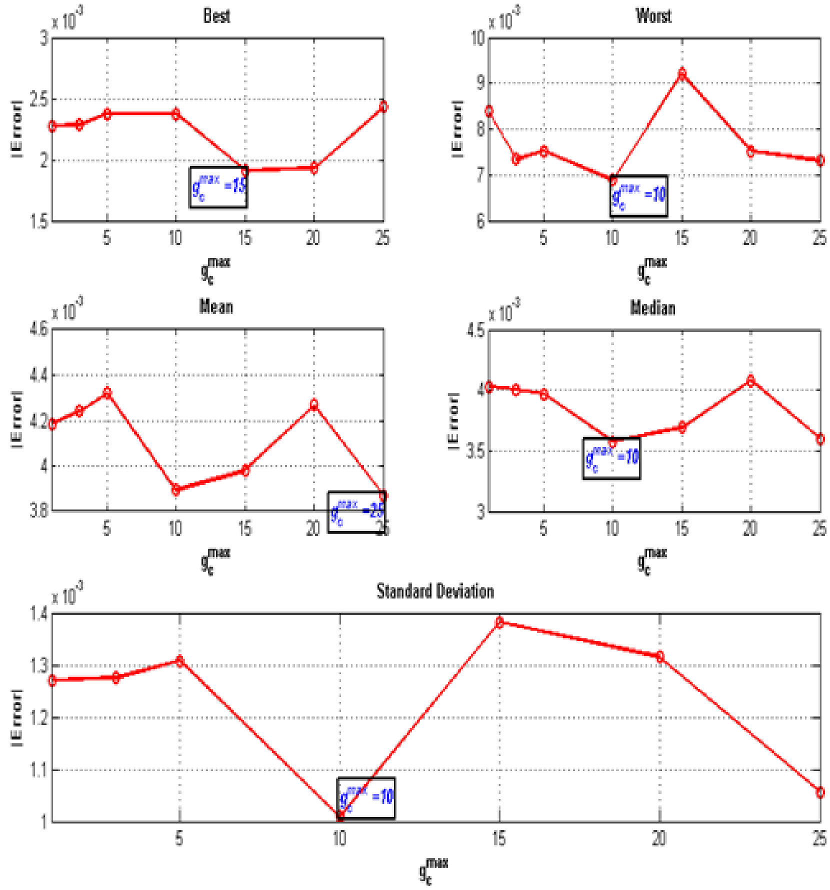

Fig. 9. Impact of $g_c^{\max}$ on LA performance

VI. Conclusion and Future Work

We have introduced LA for solving nonlinear system identification problem for which Bilinear series model was used. Experiments were carried out to estimate the behaviour model of a nonlinear rationale benchmark digital system. In the first case, standard bilinear model was used in which LA dominated over the standard GA and DE. In the second case, large scale bilinear model was used to test the algorithms in which LA outperformed GA and proved as equivalent to DE. The obtained results are encouraging, if the depth of experimentation is not considered. Hence, the future work has to be conducted with wide experimental study in terms of large scale problems along with systematic comparisons. Further, we have planned to extend the work for other nonlinear models such as volterra series, cognitive systems, etc.

References

[1] B. Subudhia and D. Jenab, “A differential evolution based neural network approach to nonlinear system identification”, *Applied Soft Computing*, vol. 11, no. 1, pp. 861–871, January 2011.

[2] S. Chen, S.A. Billings, “Representation of non-linear systems: the NARMAX model”, *Int. J. Control* vol. 49, no. 1013–1032, 1989.

[3] M. Adjrad, A. Belouchrani, “Estimation of multi component polynomial phase signals impinging on a multisensor array using state-space modelling”, *IEEE Trans. Signal Process*. vol. 55, no.1, pp. 32–45, 2007.

[4] H. Hujiberts, H. Nijmeijer, R. Willems, “System identification in communication with chaotic systems”, *IEEE Trans. Circuits Syst. I* vol. 47, no. 6, pp. 800–808, 2000.

[5] K. Watanbe, I. Matsuura, M. Abe, M. Kebota, D.M. Himelblau, “Incipient fault diagnosis of chemical processes via artificial neural networks”, AICHE J. vol. 35, no. 11, pp. 1803–1812, 1989.

[6] Y. Xie, B. Guo, L. Xu, J. Li, P. Stoica, Multistatic adaptive microwave imag-ing for early breast cancer detection, *IEEE Trans. Biomed. Eng.* vol. 53, no.8, pp.1647–1657, 2006.

[7] W. Chang, “Differential evolution-based nonlinear system modeling using a bilinear series model”, *Applied Soft Computing*, vol. 12, no.11, pp. 3401–3407, November 2012.

[8] J. Ding, L. Han, X. Chen, “Time series AR modeling with missing observations based on the polynomial transformation”, *Mathematical and Computer Modeling*. vol. 51, pp. 527–536, 2010.

[9] I. Rojas, O. Valenzuela, F. Rojas, L.J. Herrera, H. Pomares, L. Marquez, M. Pasadas, “Soft-computing techniques and ARMA model for time series prediction”, *Neurocomputing*, vol. 71, pp. 519–537, 2008.

[10] A. Kizilkaya, A. H. Kayran, “ARMA model parameter estimation based on the equivalent MA approach”, *Digital Signal Processing* vol. 16, pp. 670–681, 2006.

[11] M. N. Nounou, H. N. Nounou, "Improving the prediction and parsimony of ARX models using multiscale estimation", *Applied Soft Computing* vol.7, pp. 711–721, 2007.
[12] M. Muddu, A. Narang, S. C. Patwardhan, "Reparametrized ARX models for predictive control of staged and packed bed distillation columns", *Control Engineering Practice* vol. 18, pp. 114–130, 2010.
[13] B. Wang, N. L. Tai, H.Q. Zhai, J. Ye, J. D. Zhu, L. B. Qi, "A new ARMAX model based on evolutionary algorithm and particle swarm optimization for short-term load forecasting", *Electric Power Systems Research* vol. 78, pp. 1679–1685, 2008.
[14] C. Lim, M. McAleer, J. C. H. Min, "ARMAX modelling of international tourism Demand", *Mathematics and Computers in Simulation* vol. 79, pp. 2879–2888, 2009.
[15] J. S. H. Tsai, C. T. Wang, C. C. Kuang, S. M. Guo, L. S. Shieh, C. W. Chen, "A NARMAX model-based state-space self-tuning control for nonlinear stochastic hybrid systems", *Applied Mathematical Modeling* vol. 34, pp. 3030–3054, 2010.
[16] X. Zhao, V. Z. Marmarelis, "Nonlinear parametric models from Volterra kernels measurements", *Mathematical and Computer Modeling* vol. 27, no. 5, pp. 37–43, 1998.
[17] R. J. G. B. Campello, G. Favier, W. C. doAmaral, "Optimal expansions of discrete-time Volterra models using Laguerre functions", *Automatica* vol. 40, pp. 815–822, 2004.
[18] J. Zhang, H. Zhao, "A novel adaptive bilinear filter based on pipelined architecture", *Digital Signal Processing* vol. 20, pp. 23–38, 2010.
[19] M. Schetzmen, "The Voltera and Winner Theories on Nonlinear Systems", *Wiley*, *New York*, 1980.
[20] F. Dinga, T. Chenb, "Identification of Hammerstein nonlinear ARMAX systems", *Automatica* vol. 41, pp.1479–1489, 2005.
[21] E. Hernandez, Y. Arkun, Control of nonlinear systems using polynomial ARMA models, AICHE J. vol. 39, no. 3, pp. 446–460 1993.
[22] T. T. Lee, J. T. Jeng, "The Chebyshev polynomial-based unified model neural net-works for functional approximation", *IEEE Trans. Syst. Man Cybern*. *B* vol. 28, pp.925–935, 1998.
[23] C. Bruni, G. DiPillo, and G. Koch, "Bilinear Systems: An Appealing Class of 'Nearly Linear Systems' in Theory and Applications," *IEEE Trans. Automat. Control*, vol. AC-19, pp. 334-348, 1974.
[24] W. K. Kim, L. Billard, and V. Basawa, "Estimation for the First-Order Diagonal Bilinear Time Series Model," *Journal of Time Series Analysis*, vol. 11, no. 3, pp. 215-229, 1990.
[25] R. R. Mohler and W. J. Kolodziej, "An Overview of Bilinear System Theory and Applications," *IEEE Trans. Systems*, Man and Cybernetics, vol. SMC-IO, pp. 683- 688, Oct,1980.
[26] T. D. Pham and L. T. Tran, "On the First-Order Bilinear Time Series Model," 1. Applied Probability, vol. 18, pp. 617-627, 1981.
[27] N. Kalouptsidis, P, Koukoulas, V. J. Mathews, "Blind identification of bilinear systems", IEEE Transactions On Signal Processing vol. 51, pp. 484–499, 2003.
[28] P. Koukoulas, V. Tsoulkas, N. Kalouptsidis, "A cumulant based algorithm for the identification of input output quadratic systems", *Automatica* vol. 38, pp. 391–407, 2002.
[29] V. J. Mathews, T. K. Moon, "Parameter estimation for a bilinear time series model", *in Proc. IEEE Int. Conf. Acoust. Speech Signal Process*. pp. 3513–3516, 1991.
[30] V. Tsoulkas, P. Koukoulas, N. Kalouptsidis, "Identification of input output bilinear systems using cumulants", *IEEE Trans. Signal Processing* vol. 49, pp. 2753–2761, 2001.
[31] V. J. Mathews, G. L. Sicuranza, "Polynomial signal processing".(New York: Wiley), 2000.
[32] N. Karaboga, "Digital IIR Filter Design Using Differential Evolution Algorithm", EURASIP Journal on Applied Signal Processing vol. 8, pp. 1269–1276, 2005.
[33] B. Subudhi, D. Jena, "Differential Evolution and Levenberg Marquardt Trained Neural Network Scheme for Nonlinear System Identification", Neural Process Lett. pp. 285–296, 2008.
[34] B. Subudhi, D. Jena, M. M. Gupta, "Memetic Differential Evolution Trained Neural Networks for Non linear System Identification", *IEEE Region 10 Colloquium and the Third International Conference on Industrial and Information Systems*, Kharagpur, INDIA, pp. 8–10, 2008.
[35] Z. Wang, H. Gu, "Parameter Identification of Bilinear System Based on Genetic Algorithm", LNCS vol. 4688, pp.83–91, 2007.
[36] H. Bauer, H.H. de Iongh and I. Silvestre, "Lion social behaviour in the West and Central African Savanna belt.Mammalian Biology, vol. 68, no.1, pp. 239-243, 2003.
[37] B.R. Rajakumar, "The Lion's Algorithm: A New Nature-Inspired Search Algorithm", *Second International Conference on Communication*, Computing & Security, vol. 6, pp. 126–135, 2012, http://dx.doi.org/10.1016/j.protcy.2012.10.016
[38] C. Packer and A. E,Pusey, "Male takeovers and female reproductive parameters: A simulation of oestrous synchrony in lions (Panthera leo)", Animal Behavior, vol. 31, no. 2, pp. 334-340, 1983
[39] L. J. Fogel, A. J. Owens, and M. J. Walsh, "Artificial Intelligence through Simulated Evolution", *New York: Wiley Publishing*, 1966.
[40] B. Doerr, E. Happ, "Christian Klein: Crossover can provably be useful in evolutionary computation". Theor. Comput. Sci. vol. 425, pp.17-33 2012.
[41] T, Back, F. Hoffmeister and H. P. Schwefel, "An Overview of Evolutionary Algorithms for Parameter Optimization. Journal of Evolutionary Computation, ed. K. De Jong, Cambridge, MA: MIT Press, vol. 1 no. 1, pp.1-24, 1993.
[42] K. A. D. Jong, "An Analysis of the Behavior of a Class of Genetic Adaptive Systems". *Doctoral thesis, Dept. Computer and Communication Sciences*, University of Michigan, Ann Arbor, 1975.
[43] C. Packer, A. E. Pusey, "Divided we fall: Cooperation among lions", *Scientific American*, vol. 276, pp. 52-59, 1997.
[44] R. I. Pocock, "The Fauna of British India, including Ceylon and Burma". *Mammalia. Taylor and Francis Ltd*., London. vol. 1, pp. 212–222, 1939.
[45] S. J. O'Brien, P. Joslin, G. L. Smith, III, R, Wolfe, N. E,Schaffer, Heath, J, Ott-Joslin, P. P. Rawal, K. K. Bhattacharjee, and J. S. Martenson, "Evidence for African origins of founders of the Asiatic lion Species Survival Plan". Zoo Biology vol. 6 no. 2, pp. 99–116,1987. doi:10.1002/zoo.1430060202.
[46] R. Chellam, and A. J. T. Johnsingh, "Management of Asiatic lions in the Gir Forest, India. In N. Dunstone and M. L. Gorman (eds.) Mammals as predators: the proceedings of a symposium held by the Zoological Society of London and the Mammal Society", London. Volume 65 of Symposia of the Zoological Society of London. Zoological Society of London, London, 1993.
[47] R. A. Sterndale, "Natural History of the Mammalia of India and Ceylon", Thacker, Spink and Co., Calcutta, 1884.
[48] Packer and Pusey, "Cooperation and competition within coalition of male lions: Kin selection or game theory", Macmillan journals, vol. 296, no. 5859, pp. 740-742, 1982.
[49] T. Kohonen, "Self-organization and associative memory", Berlin, Germany: Springer- Verlag, vol. 8, 1984
[50] J. Ginsburg, S.Okolo, G, Prelevic, P. Hardiman "Residence in the Londonarea and sperm density". Lancet. pp. 343-230, 1994.
[51] M. Olhofer, Y. Jin, and B. Sendhoff. "Adaptive encoding for aerodynamic shape optimization using evolution strategies". In Proc. of IEEE Congress on Evolutionary Computation), vol. 2, pp. 576–583. IEEE Press, May 2001.
[52] Bellman, R.E, "Dynamic Programming. Dover Books on Mathematics. Princeton University Press", Princeton 1957.
[53] T. Back. "Evolutionary Algorithms in Theory and Practice: Evolution Strategies", *Evolutionary Programming, Genetic Algorithms. ser. Dover Books onMathematics*. OxfordUniversity Press, 1996.
[54] T.Back, D. B. Fogel, and Z. Michalewicz, editors. "Handbook of Evolutionary Computation". Institute of Physics Publishing, Bristol, and Oxford University Press, New York, 1997.
[55] C. A. C. Coello, D. A. V. Veldhuizen, and G. B. Lamont. "Evolutionary Algorithms for Solving Multi-Objective Problems. Kluwer Academic Publishers", New York, USA, 2002.
[56] R.V. Kulkarni, G.K.Venayagamoorthy, "Bio-inspired Algorithms for Autonomous Deployment and Localization of Sensor Nodes", *IEEE Transactions on Systems*, Man, and Cybernetics, Part C: Applications and Reviews", vol. 40, no. 6, pp. 663-675, 2010.
[57] Y. Liu, H. Wang, J. Yu, P. Li, "Selective recursive kernel learning for online identification of nonlinear systems with NARX form", *Journal of Process Control* vol. 20, pp.181–194, 2010.