

RBF Neural Network combined with Self-Adaptive MODE and Genetic Algorithm to Identify Velocity Profile of Swimmers

Luciano F. da Cruz, Roberto Z. Freire and
Gilberto Reynoso-Meza

Industrial and Systems Engineering Graduate Program
PUCPR

Curitiba, Brazil

luciano.cruz@consultant.volvo.com,
roberto.freire@pucpr.br, g.reynosomeza@pucpr.br

Flavia B. Pinto and Leandro dos S. Coelho

Industrial and Systems Engineering Graduate Program

Department of Electrical Engineering

PUCPR, UFPR

Curitiba, Brazil

fla.bernardo@gmail.com, leandro.coelho@pucpr.br

Abstract— On the side of enhancing the execution of skills, specialists in sports are adopting analysis of kinematics to correct actions of an athlete. By means of technological resources used to measure physical variables and to supply relevant data to trainers, results related to improvements on athletes' performance are being achieved. In this context, this work uses the Radial Basis Function Neural Networks (RBF-NNs) combined in cascade with a Genetic Algorithm (GA) as a cross-correlation method, likewise Multiobjective Differential Evolution (MODE) and self-adaptive MODE (using JADE - Adaptive Differential Evolution with Optional External Archive - self-adaptation method) for optimization. The RBF-NNs have been applied to predict the swimmers velocity profile being multiple correlation coefficient (R^2) adopted to evaluate the optimization techniques during both estimation and validation stages. A data acquisition system was used to capture the para-swimming athletes' instantaneous velocity data, swimming 25 meters in crawl, breaststroke, backstroke, and butterfly strokes. Looking at the results achieved, self-adaptive MODE outperforms classical MODE in approach point of view considering all cases studied, finding the best RBF-NN framework to identify the profile of speed in swimming.

Keywords—Multiobjective optimization; RBF neural networks; self-adaptive; swim; time series forecasting.

I. INTRODUCTION

BIOMECHANICS is a branch that aims to increase the achievement of athletes and parathletes, to prevent injuries and to raise results in rehabilitation. Researches in kinematic analysis dedicated to swimming have been concentrated in technological innovations through based on equipment development to measure swimmers velocity. Taken as an important advantage point in swimming, speed is constantly connected to impulsive and resisting forces [1-4], arrangement and harmony of the athletes' movements [5]. Researchers have dedicated attention in velocity studies of swimming aiming a kind of system identification [6-7].

By using data sets acquired from a prototype presented in [7], a method to forecast the swimmers velocity was chosen through the identification of signal characteristics. In this particular case, an Artificial Neural Network (ANN) has been chosen, because it is capable to recognize nonlinear

connections of input variables, by extracting intrinsic information through the learning process based in data for training [12].

A Radial Basis Function Neural Network (RBF-NN) is a particular case of ANNs, which is consisting of three layers. The definition of parameters vector (center, widths and out weights) is the main problem faced by RBF-NN, which will reach advantageous results for a cost function. One way to solve this problem is the association of RBF-NNs and optimization methods.

Evolutionary algorithms (EAs) were initially dedicated to fix optimization problems being it connected with identification methods in distinct areas [13-16]. Differential Evolution (DE) is a particular kind of EA method that has gained popularity [10,17]. Based on DE concepts, authors in [18] has extrapolated researches to multiobjective optimization concerns, achieving then, Multiobjective Differential Evolution (MODE) algorithm which uses Pareto-based rating as criteria to select best fitness in a community of individuals.

The achievement of a DE algorithm, however, is still quite dependent on the setting of control parameters being so the mutation factor and the crossover probability, for instance, according to both experimental studies and theoretical analysis [19]. Although, suggestions for parameter settings are cited on specialized literature [17,20,21]. However, the interaction between the parameter setting and the optimization performance is still complicated and not completely understood. This is mainly because there is no fixed parameter setting that is suitable for a wide range of problems or even at different evolution stages of a single problem.

Self-adaptive parameter control is a method used to conduct the self-adaptation, the parameters are directly associated with population at an individual or population level, where a natural selection occurs. Since better parameter values trend to generate individuals which have more probability to perpetuate the species, these values can be propagated to more offspring. One method used in the self-adaptation of DE and MODE methods is called Adaptive Differential Evolution with

Optional External Archive (JADE), which was firstly introduced by Zhang and Sanderson in [19].

Along these lines, this paper brings a swim speed study concentrated in the computational identification of elite para-swimming athletes through the usage of RBF-NN combined to GA, MODE and self-adaptive MODE algorithms aiming for enhance the RBF-NN achievement in a cascaded methodology. The purpose is to compare the performance of classical MODE and self-adaptive MODE combined with GA. Both the GA with MODE or the GA associated to the self-adaptive MODE have been applied looking for optimizing the adaptability of RBF-NN when seeking for the best solutions by increasing the reliability of variables built.

The paper is structured in the follow way: Section II presents a state of art over the optimization algorithms applied to this work, namely GA, MODE and self-adaptive MODE; Section III gives a contextualization on RBF-NN model as well as suggested learning process; Section IV explains the suggested procedures over optimization algorithms used to RBF-NN; Section V brings case studies in swimming looking at identification of the velocity outline; Section VI presents the achievements and considerations; and finally Section VII states the final comments and future works.

II. EVOLUTIONARY ALGORITHMS

By applying biologically inspired concepts of Darwian evolution, Evolutionary Algorithms (EAs) have been applied to solve computationally hard problems in the area of numerical optimization, combinatorial optimization, machine learning, neural networks, and many other engineering problems [22-25]. The following subsections present the EAs that were applied to this work.

A. Genetic Algorithm (GA)

Genetic Algorithms (GAs) are adaptive heuristic search algorithm revolving around evolutionary ideas of natural selection and genetics. As such they represent an intelligent exploitation of a random search applied to fix optimization issues. GAs exploit the capability of an individual to perpetuate certain genetic information to next generations. The basic techniques of the GAs are designed to simulate processes in natural systems necessary for evolution especially those follow the principles first laid down by Darwin [26]. The GA maps the problem inside a batch of seeds, each seed demonstrating a possible solution. It employs the best encouraging seeds in its seek for enhanced solutions and manipulates according to an elementary sequence, as found in [27].

B. Multiobjective Differential Evolution (MODE)

MODE is an inspired nondominated sorting genetic algorithm where the main target is to identify the Pareto optimal solution set. A Pareto-based approach selects the best individuals. Following a simple cycle, a population is generated randomly and the fitness functions are evaluated. At every generation of evolutionary search, the population is sorted into several ranks based on dominance concept. As general differential evolution algorithms operations are carried out over the individuals of the population. At the end functions of trial vectors, are evaluated. Then, the ranking of global

population is carried out following crowding distance criteria [28]. To this paper MODE steps was applied as described in details in [11]. In case of population takes more individuals than population size, it must be manipulated to make algorithm capable to achieve next statement. The manipulation is based in an arranging of individuals in a sorting without dominance and later assessing the individuals from the equal front with the criteria of crowding distance metric.

C. Adaptive Differential Evolution with Optional External Archive (JADE)

The original population of MODE is randomly generated according to a uniform distribution between lower and upper limits defined for each component of and individual. After initialization, DE enters a loop of evolutionary operations: mutation, crossover and selection. DE/rand/1 is the first mutation strategy developed for DE and has been applied for general DE applications. There are several strategies found for DE, in review of the fast but less reliable convergence performance of greedy strategies, DE/current-to- p -best is adopted as the fundament of the self-adaptive DE algorithm, JADE [19]. The two involved control parameters, F (scaling factor) and CR (crossover ratio), are usually problem dependent and need to be tuned by trial and error. In JADE, F and CR are updated by a self-adaptation mechanism that is based on a simple principle: Better values of control parameters trend to generate individuals that are capable to perpetuate and thus these values should be propagated. According to [19], in DE/current-to- p -best, a mutation vector is generated in the following manner:

$$\mathbf{v}_{i,g} = \mathbf{x}_{i,g} + F_i(\mathbf{x}_{\text{best},g}^p - \mathbf{x}_{i,g}) + F_i(\mathbf{x}_{r1,g} - \mathbf{x}_{r2,g}) \quad (1)$$

where $\mathbf{x}_{\text{best},g}^p$ is uniformly chosen as one of the top $100p\%$ individuals in the current population with $p \in (0,1]$. At each generation g , the crossover probability CR_i of each individual $\mathbf{w}_i$ is independently generated according to a normal distribution of mean μ_{CR} and standard deviation 0.1, an can be described as:

$$CR_i = \text{randni}(\mu_{CR}, 0.1) \quad (2)$$

and then truncated to $[0, 1]$. By considering S_{CR} the collection of all successful crossover probabilities CR_i 's at generation g . The mean μ_{CR} is initialized to be 0.5 and then updated at the end of each generation as:

$$\mu_{CR} = (1 - c) \cdot \mu_{CR} + c \cdot \text{meanA}(S_{CR}) \quad (3)$$

where c is a positive constant between 0 and 1 and $\text{meanA}(\cdot)$ is the arithmetic mean. Similarly, at each generation g , the mutation factor F_i of each individual $\mathbf{w}_i$ is independently generated according to a mixture of a uniform distribution $\text{randi}(0, 1.2)$, and a normal distribution $\text{randni}(\mu_F, 0.1)$ which is the mean μ_F , with standard deviation of 0.1, and truncated to $[0, 1.2]$. That is,

$$\begin{cases} \text{randi}(0, 1.2) & \text{if } i \in I_{1/3} \\ \text{randni}(\mu_F, 0.1) & \text{otherwise,} \end{cases} \quad (4)$$

where $I_{1/3}$ denotes a random collection of one-third indices of the set $\{1, 2, \dots, NP\}$ being NP the population size. S_F is the set of all successful mutation factors F_i 's at generation g , and the mean μ_F , from the normal distribution, is then updated as follows:

$$\mu_F = (1 - c) \cdot \mu_F + c \cdot L(S_F) \quad (5)$$

where $L(\cdot)$ is the Lehmer mean:

$$L(S_F) = \sum_{F \in S_F} F^2 / \sum_{F \in S_F} F. \quad (6)$$

Regarding the constant c in (3) and (5), no parameter adaptation takes place if $c \geq 0$. Otherwise, the *life span* of a successful CR_i or F_i is roughly $1/c$ generations; i.e., after $1/c$ generations, the old value of μ_{CR} or μ_F is reduced by a factor of $(1 - c)^{1/c} \rightarrow 1/e \approx 37\%$, when c is close to zero.

III. RADIAL BASIS FUNCTIONS NEURAL NETWORKS

A RBF-NN is a special type of artificial neural network which uses a radial basis function as its activation function. It has three layers: input layer with x_m nodes, a hidden layer with ϕ_k neurons or Radial Basis Functions (RBFs), and an output layer with y_n nodes, those composed by linear units (Figure 1).

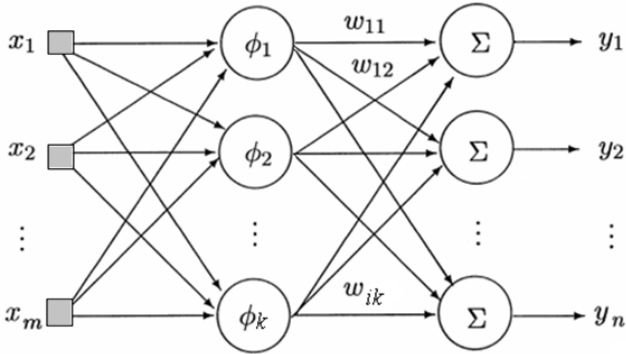


Fig. 1. Representation of a RBF-NN structure [10].

By calculating the distance between network inputs and hidden layer centers it is possible to establish the outputs which are weighted forms of the input layer. The neurons in output layer bring the intrinsic information from input by performing simple weighted summations. The widely applied activation function is the Gaussian function [29].

Given an input vector $\mathbf{r}(t) \in \mathbb{R}^m$ at time t , the output of the hidden layer when using Gaussian RBF is:

$$\varphi_i(\mathbf{r}(t)) = \exp\left\{-\frac{\|\mathbf{r}(t) - \mu_i\|^2}{\sigma_i^2}\right\} \quad (7)$$

where $\mathbf{r}(t)$ is the input vector, μ_i is the center of i -th neuron, σ_i is the radii of the function of the i -th neuron, and finally $\|\mathbf{a}\|$ is the Euclidian norm of the vector $\mathbf{a}$. By considering that, the output layer can now be written as the following summation:

$$\hat{y}(t) = \sum_{i=1}^k w_i \cdot \varphi_i(x) \quad (8)$$

where w_i is each of the synaptic weights which connects the hidden neurons i to the output neuron and k is the total number of neurons in the hidden layer. The performance of a RBF-NN model is connected to the model construction. Important issue, in this way, is to determine the RBF centers and the number of such centers based on the generalization capability [30].

IV. EXPECTED METHODOLOGY

Since the centers, widths and output weights should be set in a RBF-NN, a training phase adjusts the Gaussian basis function centers by applying a normal distribution to generate the centers inside the range $[0, 1]$. The MODE and self-adaptive MODE optimization methods were adopted to improve seek of the widths and locally the centers of the Gaussian basis functions. The RBF-NN identification method splits the acquired data within training (75%) and validation (25%). The selection of the amount to be destined to each phase is made by the authors in line with the number data set available. Once the application of the RBF-NN in swimming velocity profile identification procedures is recent, there are no references about the effectiveness of data split. In this way, the multiple correlation coefficient was adopted, and it is stated as:

$$R^2 = 1 - \frac{\sum_{t=1}^{N_s} [y(t) - \hat{y}(t)]^2}{\sum_{t=1}^{N_s} [y(t) - \bar{y}]^2} \quad (9)$$

where N_s is the number of samples in a given set, $y(t)$ is the output of the real system, $\hat{y}(t)$ is the output estimated by RBF-NN, and $\bar{y}$ is the mean value of the system's output.

The multiple correlation coefficient was used for training (R^2_u) and validation phases (R^2_v) as decision variables inside genetic algorithm. Therefore objective functions adopted by MODE and self-adaptive MODE are given by:

$$f_1(x) = 1/(1 + R^2_{tr}), \quad f_2(x) = 1/(1 + R^2_v) \quad (10)$$

being so, the training procedure has the objective of enhance the accuracy of the model and its generalization. Values for R^2 higher than 0.9 are enough to express a model in identification field according to authors in [31]. By employing a cascaded evolutionary algorithm composed by GA with MODE and self-adaptive MODE, the suggested methodology to improve simultaneously the lags in the time series used as inputs of the RBF-NN model and its parameters. In order to realize these modifications in the neural network, the suggested strategy uses, in the first layer, a simple GA, binary coded, which selects lags used in the series. Therefore, an individual of GA means the presence/absence of a lag in a particular model.

Similarly as in [11], the following objective function (Fobj) has been adopted for the GA:

$$F_{obj} = \text{RMSE} + \sum_{k=-\tau_{max}}^{\tau_{max}} [\phi_{\xi(\tau)}'(\tau)^2 + \phi_{\xi(\tau^2)}'(\tau)^2 + \phi_{(\xi(\tau))'(\xi(\tau))}(\tau)^2] \quad (11)$$

where RMSE means for the root mean squared error metric, $\phi_{ab}(\tau)$ is the cross-correlation function between two sequences $\{a\}$ and $\{b\}$, and τ_{max} is the maximum number of candidates to be tested. Authors in [32] have developed a group of statistical correlation tests concerned to nonlinear model testing and verification. Based in One Step Ahead (OSA) model prediction, authors use the model residual $\xi(\tau)$ to define whether the model will be unpredictable from all linear and nonlinear combinations of past inputs and outputs. In practice the 95% assurance bands, which are approximately $\pm 1.96/\sqrt{N}$ where N is data length, are employed to decide whether the tests are satisfied and the model is validated. The flowchart presented in Figure 2 illustrates the suggested methodology as an interactive way for further understanding. It may be noticed how GA, MODE and self-adaptive MODE (JADE as self-adaptation method) act on the overall optimization process.

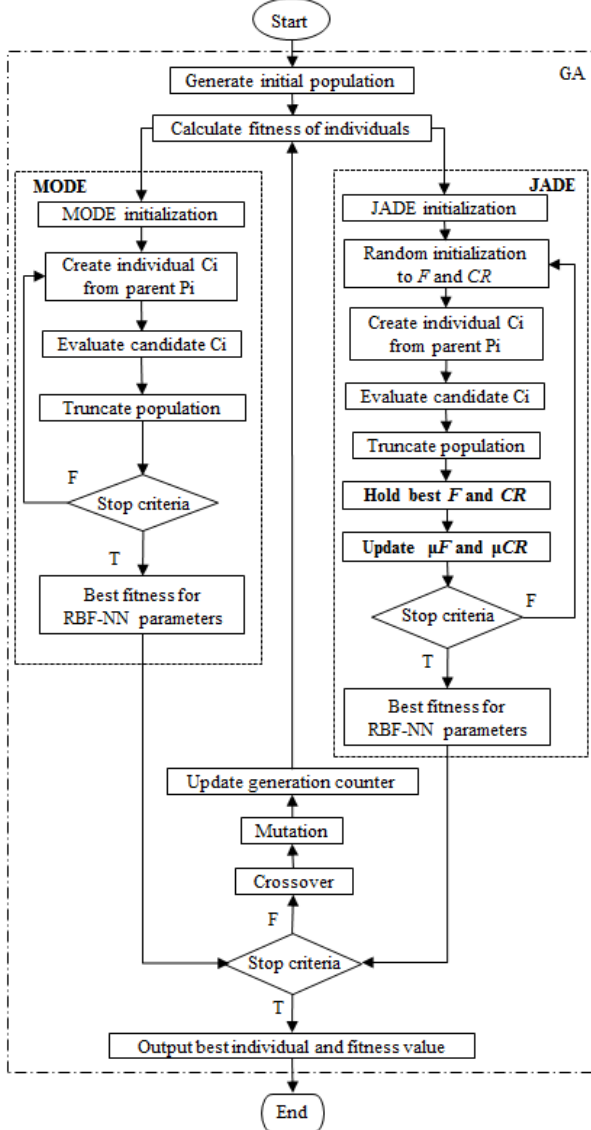


Fig. 2. Flowchart of the planned JADE algorithm compared to MODE.

V. EXPERIMENTS AND IDENTIFICATION PROCEDURES

Through the usage of a prototype which uses an encoder data of instantaneous speed was acquired of elite para-swimming athletes in swimming pool of 25 meters long (Figure 3) with 50ms of settling time being so, 352 samples obtained in every test. Swimmers were directly linked to the acquisition system over a line.

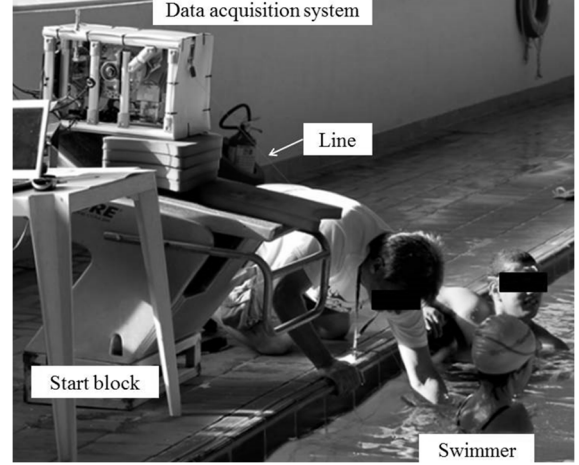


Fig. 3. Training sessions supported by kinematic analysis.

The data acquisition system has also a real time plot screen and data saving mode in a spreadsheet to make trainers able to evaluate and perform live improvements in swimmers' behavior during training sessions (Figure 4). The swim cases acquired were: crawl stroke (male with legs' paralysis), breaststroke and butterfly stroke (female with traumatic amputation of forearm), and backstroke (male with congenital malformation in the right leg). The data set was assessed using EAs for time series approach. The system identification context through RBF-NN applied to support kinematic analysis in swimming aims to correlate swimmers between them or the identification of self-features relevant to performance increase, such as: strokes, hand position, propulsive forces, body position, among others.

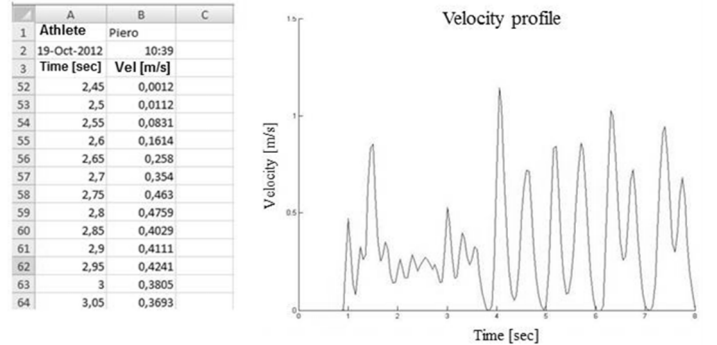


Fig. 4. Data saving and real time plot screen.

VI. SIMULATION RESULTS

Identification achievement for both training and validation procedures considering the presented methodology. Table I summarizes all case studies (different swimmers and strokes)

in terms of numbers of neurons in hidden layer of RBF-NN, R^2 values in training and validation phases, lags selected by GA algorithm in conjunction with comparison between classical MODE and self-adaptive MODE.

TABLE I. RESULTS REACHED COMPARING MODE AND SELF-ADAPTIVE MODE FOR THE EXPECTED PROCEDURE

Stroke	Value					
	Neurons	<i>Regressors chosen</i> $y(t-i) - [i]$	MODE		JADE	
			R^2_{tr}	R^2_v	R^2_{tr}	R^2_v
Crawl	9	1,2,3,4,8,11,12,15,17,20	0,9221	0,9362	0,9287	0,9374
	10	1,2,3,6,7,8,9,11,12,13,16,18	0,9266	0,9401	0,9288	0,9488
	11	1,2,3,4,6,9,19	0,9278	0,9422	0,9325	0,9450
	12	1,2,4,5,6,8,10,11,12,16,17	0,9322	0,9376	0,9330	0,9452
	13	1,2,3,4,5,6,8,9,13,14,16,17	0,9339	0,9400	0,9357	0,9472
Breaststroke	9	1,2,3,4,7,8,10,12,18,19,20	0,9122	0,9297	0,9138	0,9398
	10	1,2,6,7,12,13,14,20	0,9116	0,9353	0,9179	0,9342
	11	1,2,3,4,6,8,9,10,12,14,20	0,9187	0,9337	0,9223	0,9347
	12	1,2,3,4,6,8,9,10,12,14,20	0,9251	0,9305	0,9264	0,9388
	13	1,2,3,5,12	0,9254	0,9376	0,9376	0,9443
Backstroke	9	1,2,3,6,7,9,11,17	0,7608	0,5066	0,7489	0,7789
	10	1,2,3,4,6,7,8,9,10,11,17	0,7591	0,5469	0,7520	0,7843
	11	1,2,3,6,8,9,11,18,19,20	0,7621	0,7660	0,6186	0,8502
	12	1,2,3,6,7,12,18	0,6529	0,8317	0,7068	0,8558
	13	1,2,3,6,7,12,18	0,6686	0,8705	0,6385	0,8679
Butterfly	9	1,2,3,4,7,8,11,12,17,18,19	0,8962	0,9251	0,9113	0,9300
	10	1,2,3,6,13,14,15,16	0,9130	0,9165	0,9061	0,9301
	11	1,2,3,4,6,7,8,12,16,19,20	0,9075	0,9249	0,9152	0,9319
	12	1,2,3,4,5,6,8,10,12,18,20	0,9126	0,9273	0,9143	0,9348
	13	1,2,3,4,5,6,8,14,16,18	0,9179	0,9317	0,9208	0,9389

In general words GA is the main method run before any action from MODE or self-adaptive MODE. By applying a max number of 20 lag candidates the number of possible solutions can be limited. The main reason to apply GA algorithm working in cascade with MODE and self-adaptive MODE is computational cost saving. Setting GA parameters it was applied 50 individuals, 100 generations, 1/20 as mutation factor and 0.9 as crossover ratio. MODE algorithm received 100 individuals, 400 generations, 0.3 as crossover ratio and 0.5 as mutation factor. The self-adaptive MODE was stated with 100 individuals, 100 generations, 0.5 as crossover ratio and 0.5 as mutation factor. For JADE method it was adopted the rate of parameter adaptation c as 0.1 and greediness of mutation strategy p as 0.05. The RBF-NN was experienced from 9 to 13 neurons to every study case.

It can be noticed that, by using the same configuration for the number of neurons in the hidden layer likewise the same set up for the regressors from GA algorithm, self-adaptive MODE outperforms the results from classical MODE taking into

account the R^2 values. It happens to all number of neurons and all swimming strokes experienced. The reason of assuming the same set up for the regressors from GA algorithm cause relationship to the particularity of this method, which is assuming just the real time series to choose the regressors and it does not relation to the prediction process. It means that once the same real time series is adopted, in both classical MODE and self-adaptive MODE cases, the GA algorithm should return the same regressors at each algorithm step.

Figures 5-8 show the estimated outlines from the best GA combined with self-adaptive MODE, and the error variation from each evaluated case (stroke).

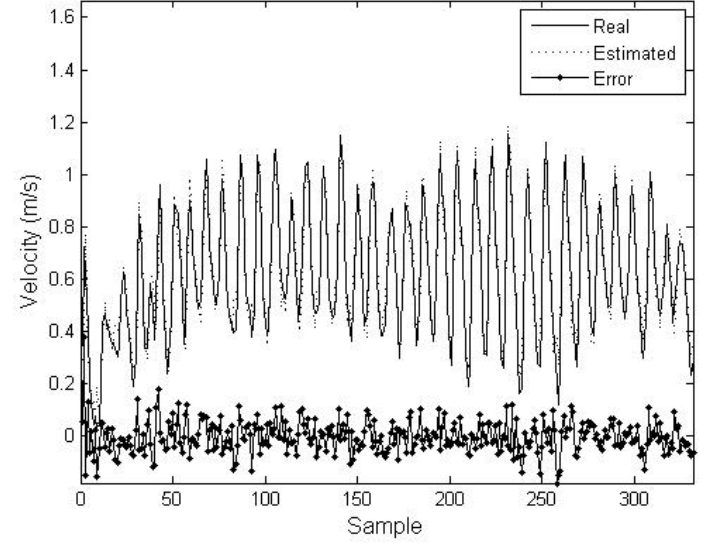


Fig. 5. Estimation and error signal for crawl study case.

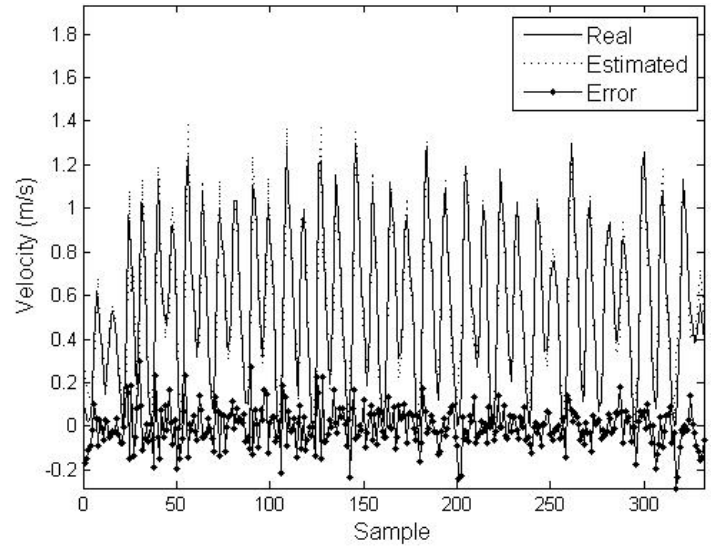


Fig. 6. Estimation and error signal for breaststroke study case.

As it can be seen, the RBF-NN was capable to forecast the time series. The efficiency of this method can still be confirmed again in Table 1, which brings the batch of lags selected R^2 values for training and validation phases. In all these cases, R^2 values greater than 0.9 were achieved.

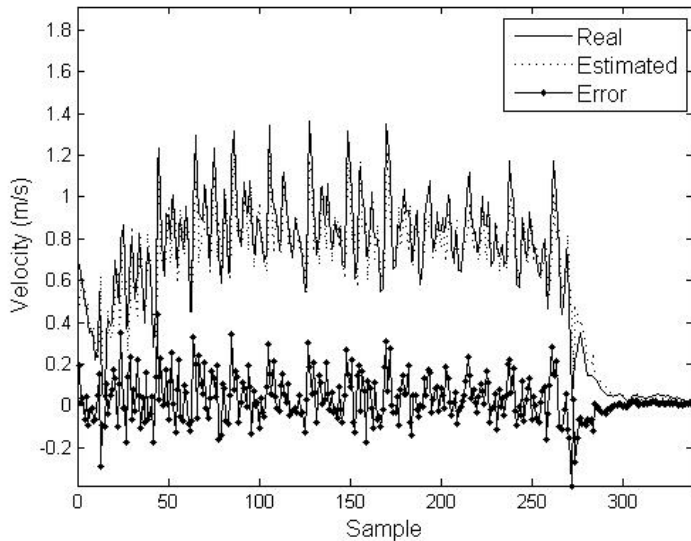


Fig. 7. Estimation and error signal for backstroke study case.

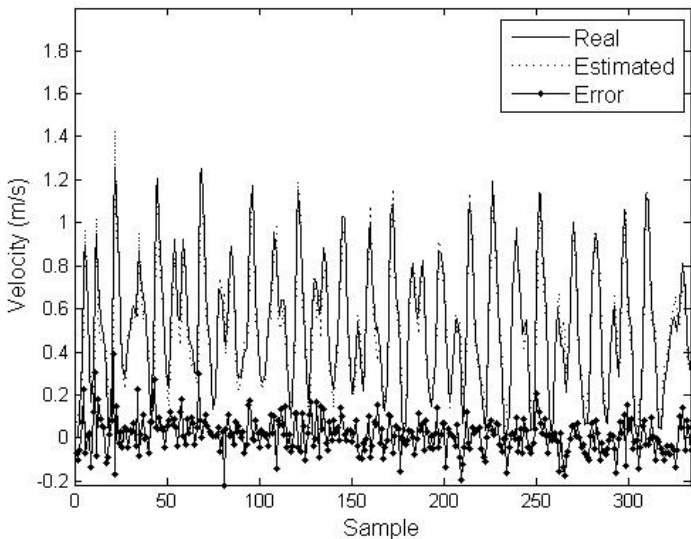


Fig. 8. Estimation and error signal for butterfly stroke study case.

VII. CONCLUSION

An application of RBF-NN associated with cascaded EAs placed on GA and self-adaptive MODE optimization methods could be verified in the current paper. Time series estimating procedures were applied to perform the velocity profile identification in four case studies (crawl, breaststroke, backstroke, and butterfly strokes) of para-swimming athletes. Once more the success of this method might be verified by R^2 values higher than 0.9. The gain by using GA algorithm in time series forecasting of swimming profile is given by computational effort decrement put to run this forecasting. Based on experience and results reach by authors in this research field numerically values of R^2 have been presented small gains through the usage of optimization techniques, in the other hand it is possible to claim that small gains have significant importance to decide training sessions to elite para-swimming athletes where milliseconds can be the difference

between first and second place in a competition environment. The main objective of applying GA algorithm is to allow the designer from selecting the collection of lags in order to avoid empirical parameters and consequently improve the convergence of the method. By improving the models offline, the purpose is to afford a more accurate way to compare athletes from the same team and improve the achievement of the para-swimming athletes. In this way, comparisons among athletes from the same class of competition can be made in terms of velocity, where the correlation with a better athlete may be applied as a standard for the others. Future works are dedicated on the inclusion of inputs in this model. It means to use, for example, inputs signals to this system in a way to seek for a feed-forward system capable to give direct point to be corrected during a swimming behavior.

ACKNOWLEDGMENT

Kindly thanks to CAPES (Coordination of Improvement of Higher Education Personnel) for supporting this research.

REFERENCES

- [1] L. Seifert, C. Schnitzler, G. Bideault, M. Alberty, D. Chollet, and H. M. Toussaint, "Relationships between coordination, active drag and propelling efficiency in crawl", in *Human Movement Science*, vol. 39, pp. 55-64, 2015.
- [2] D. Jandačka, *Biomechanical Basis of Physical Exercises*, *Základy biomechaniky tělesných cvičení*, D. Luleč, 2012.
- [3] H. M. Toussaint, P. E. Roos, and S. Kolmogorov, "The determination of drag in front crawl swimming", in *Journal of Biomechanics*, vol. 37, no. 11, pp. 1655-1663, 2004.
- [4] F. Hildebrand, and D. Kliche, "Relation of swimming propulsion and muscle force moment", in *Proceedings of 23th International Symposium on Biomechanics in Sports (ISBS2005)*, Beijing, China, pp. 927-930, 2005.
- [5] S. Kudo, R. Vennell, and B. Wilson, "The effect of unsteady flow due to acceleration on hydrodynamic forces acting on the hand in swimming", in *Journal of Biomechanics*, vol. 46, no. 10, pp. 1697-1704, 2013.
- [6] A. Stamm, D. A. James, B. B. Burkett, R. M. Hangem, and D. V. Thiel, "Determining maximum push-off velocity in swimming using accelerometers", in *Procedia Engineering*, vol. 60, pp. 201-207, 2013.
- [7] P. Siirtola, P. Laurinen, J. Rönning, and H. Kinnunen, "Efficient accelerometer-based swimming exercise tracking", in *Symposium on Computational Intelligence and Data Mining*, pp. 156-161, 2011.
- [8] L. Cruz, R. Freire, and L. Coelho, "Low-cost prototype development and swim velocity profile identification using neural network associated to generalized extremal optimisation", in *Proceedings of the 12th International Symposium on Biomechanics and Medicine in Swimming (BMS 2014)*, Canberra, Australia, pp. 566-572, 2014.
- [9] H. Ayala, L. Cruz, L. Coelho, and R. Freire, "Swim velocity profile identification through a dynamic self-adaptive multiobjective harmonic search and RBF neural networks", in *Proceedings of European Symposium on Artificial Neural Networks, Artificial Intelligence and Machine Learning (ESANN 2014)*, Bruges, Belgium, pp. 637-643, 2014.
- [10] L. Cruz, L. Coelho, and R. Freire, "Swim velocity profile identification by using a modified differential evolution method associated with RBF neural network", in *Proceedings of International Conference on Innovative Computing Technology*, London, UK, pp. 389-395, 2013.
- [11] H. Ayala, L. Cruz, R. Freire, and L. Coelho, "Cascaded evolutionary multiobjective identification based on correlation function statistical tests for improving velocity analyzes in swimming", in *Proceedings of IEEE Symposium on Computational Intelligence in Multi-Criteria Decision-Making*, v. 1, Orlando, EUA, pp. 185-193, 2014.

- [12] H. Sarimveis, P. Doagnis, and A. Alexandridis, "A classification technique based on radial basis function neural networks", in *Advances in Engineering Software*, vol. 37, no. 4, pp. 218-221, 2006.
- [13] I. Tijani, R. Akmeliawati, A. Legowo, and A. Budiyo, "Nonlinear identification of a small scale unmanned helicopter using optimized NARX network with multiobjective differential evolution", in *Engineering Applications of Artificial Intelligence*, vol. 33, pp. 99-115, 2014.
- [14] J. V. Carrau, S. G. N. Rodrigez, J. V. Salcedo, and R. H. Bishop, "Multi-Objective Optimization for Wind Estimation and Aircraft Model Identification", in *Journal of Guidance, Control, and Dynamics*, vol. 38, no. 2, pp. 372-389, 2016.
- [15] P. Rakshit, A. Konar, "Differential evolution for noisy multiobjective optimization", in *Journal Artificial Intelligence*, vol. 227, pp. 165-189, 2015.
- [16] N. A. Shrivastava, K. Lohia, and B. K. Panigrahi, "A multiobjective framework for wind speed prediction interval forecasts", in *Renewable Energy*, vol. 87, pp. 903-910, 2016.
- [17] S. Das, S. S. Mullick, and P. N. Suganthan, "Recent advances in differential evolution - An updated survey", in *Swarm and Evolutionary Computation*, vol. 27, pp. 1-30, 2016.
- [18] F. Xue, A. Sanderson, R. Graves, "Pareto-based multi-objective differential evolution", in *Proceedings of Congress on Evolutionary Computation (CEC 2003)*, Canberra, Australia, vol. 2, pp. 862-869, 2003.
- [19] J. Zhang, and A. Sanderson, "JADE: Adaptive differential evolution with optional external archive", in *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 5, pp. 945-958, 2009.
- [20] K. K. Mandal and N. Chakraborty, "Parameter study of differential evolution based optimal scheduling of hydrothermal systems", in *Journal of Hydro-environment Research*, vol. 7, no. 1, pp. 72-80, 2013.
- [21] G. R. Meza, J. Sanchis, X. Blasco, and M. Martínez, "An empirical study on parameter selection for multiobjective optimization algorithms using differential evolution", in *2011 IEEE Symposium on Differential Evolution (SDE)*, pp. 1-7, 2011.
- [22] J. Holland, "Outline for a logical theory of adaptive systems", *Journal Association for Computing Machinery*, vol. 3, pp. 297-314, 1962.
- [23] D. Fogel, *Evolutionary Computation: Toward a New Philosophy of Machine Intelligence*, 2nd Edition, Wiley-IEEE Press: New York, 1999.
- [24] I. Rechenberg, *Evolutionsstrategie: Optimierung technischer systeme nach prinzipien der biologischen evolution*, 1st ed., vol. 15. Stuttgart, Germany: Frommann-Holzboog, 1973.
- [25] M. Mitchell, *An Introduction to Genetic Algorithms*, Reprint Edition, MIT Press: Cambridge, 1998.
- [26] D. Goldberg, *Genetic algorithms in search, optimization and machine learning*, 1st ed. Michigan: Addison-Wesley, 1989.
- [27] A. Bastani, and B. Shahalami, "New approach in the prediction of RDC liquid-liquid extraction column parameters", in *Chemical Engineering and Technology*, vol. 31, no. 7, pp. 971-977, 2008.
- [28] X. Wang, and L. Tang, "An adaptative multi-population differential evolution algorithm for continuous multi-objective optimization", in *Information Sciences*, vol. 348, pp. 124-141, 2016.
- [29] S. Qasem, S. Shamsuddin, and A. Zain, "Multi-objective hybrid evolutionary algorithms for radial basis function neural network design", in *Knowledge-Based Systems*, vol. 27, pp. 475-497, 2012.
- [30] R. Grebogi, W. Puchalski, F. Mendonça, and L. Coelho, "RBF neural network optimized with quantum particle swarm approach applied to force modeling", in *Proceedings of 22nd International Congress of Mechanical Engineering (COBEM 2013)*, Sao Paulo, Brazil, pp. 7144-7151, 2013.
- [31] B. Schaible, H. Xie, and Y. Lee, "Fuzzy logic models for ranking process effects", in *IEEE Transactions on Fuzzy Systems*, vol. 5, no. 4, pp. 545-556, 1997.
- [32] S. Billings, and W. Voon, "Structure detection and model validity tests in the identification of nonlinear systems", in *Proceedings of IEEE*, vol. 4, no. 4, pp. 193-200, 1983.