ISSN 2176-5480

22nd International Congress of Mechanical Engineering (COBEM 2013)
November 3-7, 2013, Ribeirão Preto, SP, Brazil

RBF NEURAL NETWORK OPTIMIZED WITH QUANTUM PARTICLE SWARM APPLIED TO SWIMMER'S PROPULSIVE FORCE MODELING

Rafael Bartnik Grebogi
Federal University of Parana (UFPR), Graduate Program in Electrical Engineering,
Rua Cel. Francisco Heraclito dos Santos, 100, Curitiba, PR, Brazil
rafagrebogi@gmail.com
Weslly Puchalski
Industrial and Systems Engineering Graduate Program (PPGEPS), Pontifical Catholic University of Parana (PUCPR)
Imaculada Conceição, 1155, Curitiba, PR, Brazil
heremitwolf@gmail.com
Fernando Van den Tempel de Mendonça
Mechatronics Engineering Undergraduate program, Pontifical Catholic University of Parana (PUCPR)
Imaculada Conceição, 1155, Curitiba, PR, Brazil
wilson_mendonca@hotmail.com
Leandro dos Santos Coelho
Industrial and Systems Engineering Graduate Program (PPGEPS), Pontifical Catholic University of Parana (PUCPR)
Imaculada Conceição, 1155, Curitiba, PR, Brazil and Department of Electrical Engineering (DEE/PPGEE)
Federal University of Parana (UFPR), Rua Cel. Francisco Heraclito dos Santos, 100, Curitiba, PR, Brazil
leandro.coelho@pucpr.br

Abstract. *Artificial neural networks (ANNs) can learn from a given data set. In this context, a special kind of the ANN is the radial basis function neural network (RBF-NN). It can be viewed as a curve fitting method in a high-dimensional space. Furthermore, RBF-NN is considered as a good candidate for the forecasting problems due to its rapid learning capacity and, therefore, has been applied successfully to nonlinear time series modeling and forecasts. The contribution of this paper is to present a quantum particle swarm optimization (QPSO) approach to RBF-NN optimization to deal the time series forecasting. To achieve this deal, in this work, the RBF-NN configuration uses the k-means clustering algorithm, QPSO and pseudo-inverse method. The focus of application of the proposed RBF-NN design aims the force modeling in the swimming field. The results show that the proposed RBF-NN design can be considered as a suitable technique for force modeling of a Brazilian elite male swimmer using time series data.*

Keywords: neural networks, radial basis function neural network, time series forecasting, swimming, force modeling.

1. INTRODUCTION

Artificial neural networks have been defined by Kohonen as “massively parallel interconnected networks of simple (usually adaptive) elements and their hierarchical organizations, which are intended to interact with the objects of the real world in the same way as biological nervous system do” (Kohonen, 1988).

Feedforward neural networks possess a number of properties which make them particularly suited to complex pattern classification and time series forecasting problems. Generally, the computation of the output weights in feedforward neural networks is based on the given error cost function (or referred to as target function), which is a function of the network structure and the weights. Particularly, the feedforward radial basis function neural networks (RBF-NNs) are flexible modeling tools that have the ability to rapidly learn complex patterns and a tendency to present data and adapt to changes quickly. Theoretically RBF-NNs can approximate any continuous function defined on a compact set to any prescribed degree of accuracy by sufficiently expanding the size of network (Park and Sandberg, 1991).

The classical RBF-NN consists of three layers: input layer, radial basis functions and output layer. Input layer takes features extracted from training samples and directly delivers them to the RBF layer. Hidden neurons are radial basis functions. The performance through the input layer to the output layer compete the task of classification by dividing the whole input space into several subspaces in the form of a hyperellipsoid. The classical training procedures for RBF neural networks are usually divided into two stages where the centers of the hidden layer are determined first in a self-organizing manner, followed by the computation of the weights that connect the hidden layer with the output layer.

Thus, to train the RBF-NN is a very important task, which requires efficient and practical optimization techniques. In recent years, several RBF-NN training methods have been proposed (Simon, 2002; Karayiannis, 1999; Sarimveis et al., 2003).

ISSN 2176-5480

The main problem in RBF-NNs design concerns establishing the number of hidden neurons to use and their centers and radii. The aim of the present paper is to validate a RBF-NN design combined with k-means clustering method to tune the centers of Gaussian functions (activation functions), a quantum particle swarm optimization (QPSO) approach to optimize the radii values and the fine-tuning of the Gaussian functions centers, and the linear least-squares method to set the output weights. In order to illustrate the behavior of the RBF-NN algorithm, a case study of force modeling of a Brazilian elite male swimmer is realized. The forecasting results show that the proposed RBF-NN has satisfactory performance.

The remainder of this paper is organized as follows: Section 2 presents the RBF-NN fundamentals. After, Section 3 presents the forecasting results using time series data related to force measurements of a Brazilian elite male swimmer. Finally, the conclusion and further research are discussed in Section 4.

2. FUNDAMENTALS OF THE RBF-NN

A RBF-NN which was introduced by Broomhead and Lowe (1988), is a special type of artificial neural network that uses a radial basis function as its activation function. From the aspect of functional representations, RBF-NN is a scheme that represents a function of interest by using members of a family of compactly (locally) supported basis functions to perform curve fitting. The locality of the basis functions makes the RBF-NN more suitable for learning functions with local variations and discontinuities.

In RBF networks, the outputs of the input layer are determined by calculating the distance between the network inputs and hidden layer centers. The second layer is the linear hidden layer and outputs of this layer are weighted forms of the input layer outputs. The neurons of the output layer have input-output relationship that performs simple weighted summations. Each neuron of the hidden layer has a parameter vector called center.

Figure 1 shows a typical structure of RBF-NN, including the input layer, a single hidden layer, and the output layer. The variables *w* represented in Figure 1 are the weights between the hidden layer and the output layer of the RBF-NN. In Figure 1, ϕ_i is the activation function of the *i*-th hidden neuron. The RBF-NN with a single hidden layer of Gaussian neurons are considered to be universal approximators. Therefore, we choose the Gaussian function as the nonlinear function of each neuron. The Gaussian basis functions are given by

$$\phi_i = \exp\left(-\frac{\|x - c_i\|^2}{r_i^2}\right) \tag{1}$$

where $x=[x_1, x_2,\ldots, x_m]$ is the input vector, $c_i=[c_{1i}, c_{2i},\ldots, c_{mi}]$ is the center vector of the *i*-th neuron, r_i is the radii of the function of the *i*-th neuron, and $\|a\|$ denotes the Euclidian norm of the vector *a*. In general, an RBFNN is a multi-dimensional function that depends on the distance between the input vector and a center vector. By building up several basis functions h_i with Euclidean distance between the input and the center, the mapping relationship between inputs and outputs can be obtained.

Since RBF-NNs are linear-in-the-parameters for fixed centers and radii, the coefficients ω_i can be determined using the linear least-squares method. The choice of the values of centers and radii is crucial for the performance of the neural compensation. In this context, clustering algorithms can be used to find a set of centers which more accurately reflects the distribution of the data points.

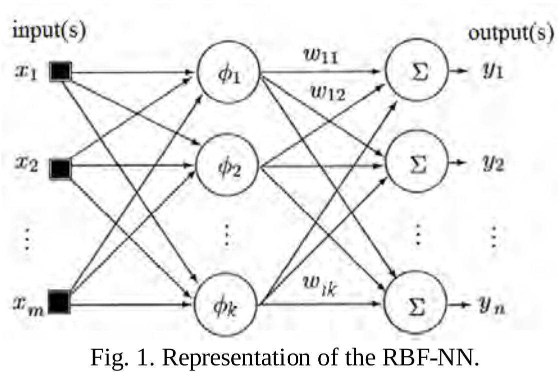

Fig. 1. Representation of the RBF-NN.

ISSN 2176-5480

The performance of an RBF-NN model depends crucially on the model construction method. One of the most important issues in developing an RBF-NN model is to determine the RBF centers, and another is to determine the number of such centers based on the generalization capability (Du et al., 2012).

In this paper, the *k*-means clustering method is adopted to tune the centers of Gaussian functions (activation functions), a quantum particle swarm optimization (QPSO) approach to optimize the radii values and the fine-tuning of the Gaussian functions centers, and the linear least-squares method to set the output weights.

Clustering algorithms are generally used in an unsupervised fashion. They are presented with a set of data instances that must be grouped according to some notion of similarity. K-means (MacQueen, 1967; Hartigan and Wang, 1979) is one of the simplest unsupervised learning algorithms which use prototypes (centroids) to represent clusters by optimizing the squared error function.

The particle swarm optimization (PSO) was first introduced in Kennedy and Eberhart (Kennedy and Eberhart, 1995; Eberhart and Kennedy, 1995). It is a flexible, robust, population-based stochastic optimization algorithm. Furthermore, PSO is a kind of swarm intelligence paradigm that is based on social-psychological principles and provides insights into social behavior, as well as contributing to engineering applications.

In the quantum model of a PSO called here quantum PSO (QPSO) proposed in Sun et al. (2004), the state of a particle is depicted by wavefunction $\psi(x, t)$ (Schrödinger equation), instead of position and velocity of classical PSO. The probability of the particle's appearing in position x_i from probability density function $|\psi(x,t)|^2$, the form of which depends on the potential field the particle lies. Employing the Monte Carlo method, the particles move according to the following iterative equation by Sun et al. (2004) and Sun et al. (2005):

$$\begin{cases} x_{i,j}(t+1) = p_i(t) + \beta \cdot \left| Mbest_j(t) - x_{i,j}(t) \right| \cdot \ln(1/u), \text{if } k \geq 0.5 \\ x_{i,j}(t+1) = p_i(t) - \beta \cdot \left| Mbest_j(t) - x_{i,j}(t) \right| \cdot \ln(1/u), \text{if } k < 0.5 \end{cases} \tag{1}$$

where $x_{i,j}(t+1)$ is the position for the *j*-th dimension of *i*-th particle in *t*-th generation (iteration); $Mbest_j(t)$ is the global point called *Mainstream Thought* or *Mean Best* (*Mbest*) for the *j*-th dimension; β is a design parameter called contraction-expansion coefficient; *u* and *k* are values generated according to a uniform probability distribution in range [0,1]; and $p_i(t)$ is local point (local attractor) defined by Clerc and Kennedy (2002). The *Mainstream Thought* or *Mean Best* (*Mbest*) is defined as the mean of the *pbest* positions of all particles and it given by

$$Mbest_j(t) = \frac{1}{N} \sum_{j=1}^{N} p_{g,j}(t), \tag{2}$$

where *N* represents the dimension of optimization problem and *g* represents the index of the best particle among all the particles' swarm in *j*-th dimension. In this case, the local attractor to guarantee convergence of the algorithm presents the following coordinates:

$$p_i(t) = \frac{c_1 \cdot p_{k,i} + c_2 \cdot p_{g,i}}{c_1 + c_2}, \tag{3}$$

where $p_{k,i}$ (*pbest*) represents the best previous *i*-th position of the *k*-th particle and $p_{g,i}$ (*gbest*) represents the *i*-th position of the best particle of the population. Positive constants c_1 and c_2 are the cognitive and social components, respectively, as in the classical PSO. The procedure for implementing the QPSO is given by the following steps (Coelho, 2008):

Step 1: Initialization of swarm positions: Initialize a population (array) of particles with random positions in the *n* dimensional problem space using a uniform probability distribution function.

Step 2: Evaluation of particle's fitness: Evaluate the fitness value of each particle.

Step 3: Comparison of each particle's fitness with its pbest (personal best): Compare each particle's fitness with the particle's *pbest*. If the current value is better than *pbest*, then set a novel *pbest* value equals to the current value and the *pbest* location equals to the current location in *n*-dimensional space.

Step 4: Comparison of each particle's fitness with its gbest (global best): Compare the fitness with the population's overall previous best. If the current value is better than *gbest*, then reset *gbest* to the current particle's array index and value.

Step 5: Updating of global point: Calculate the *Mbest* using equation (2).

ISSN 2176-5480

Step 6:Updating of particles' position: Change the position of the particles where c_1 and c_2 are two random numbers generated using a uniform probability distribution in the range [0, 1].

Step 7: Repeating the evolutionary cycle: Loop to *Step 2* until a stop criterion is met, usually a sufficiently good fitness or a maximum number of iterations (generations).

3. DESCRIPTION OF THE CASE STUDY AND THE FORECASTING RESULTS

This section describes the case study in the swimming field and forecasting results. First, a brief overview of the case study of the force measurements of a swimmer is provided, and finally the forecasting results of the proposed RBF-NN design are presented.

3.1 Case study

In the sports field, numerical simulation techniques have been shown to provide useful information about performance and to play an important role as a complementary tool to physical experiments. In swimming, this methodology has been applied in order to better understand swimming performance (Marinho et al., 2012).

Maximal swimming velocity, especially at sprint distances, depends on pulling force characteristics besides technical and energetic abilities of swimmers. Viewed kinematically, swimming is a series of cyclic movements performed by alternation of arm and leg strokes. Each stroke results in a characteristic force, which pulls the swimmer forward and is realized by contracting the muscles involved (Doopsaj et al., 2000).

The propulsive force of swimming (Gourgoulis et al., 2008), velocity (Stamm et al., 2011) and technique (Akis and Orcan, 2004) are key factors in the performance of swimmer. A specific method to evaluate the propulsive force dynamometry is fully tethered swimming. Tethered swimming has been in use for long in order to measure and evaluate swimmers propulsive force (Yeater et al., 1981). The results of tethered swimming can be useful to relationship between swimming speed and the propulsive forces, and the instantaneous power.

In this paper, the force data of a Brazilian elite male swimmer in crawl stroke swimming were measured. In this context, the purpose of the measurement was to propose a time series model for this test. The obtained of the force data was carried out using a dynamometer Instrutherm DD-300. The device was linked with the platform using a cable. Furthermore, the dynamometer was connected to a personal computer to store the tension force signals. The swimmer did a 10-second warm-up, followed by a 30-second high intensity trial of tethered swimming in order to get the force data. The sampling time was approximately 200 ms, according to the dynamometer manufacturer. Trials were simultaneously video recorded using an underwater camera. An illustration of procedure is presented in the Figures 2, 3 and 4. The force measures of the Brazilian elite male swimmer are presented in Figure 5 and Table 1.

It is advised that the swimmer should use something to clip their feet, in order to avoid any noise generated by the stride. As the legs tend to sink, it also should be used some floaters to help keeping the body alignment.

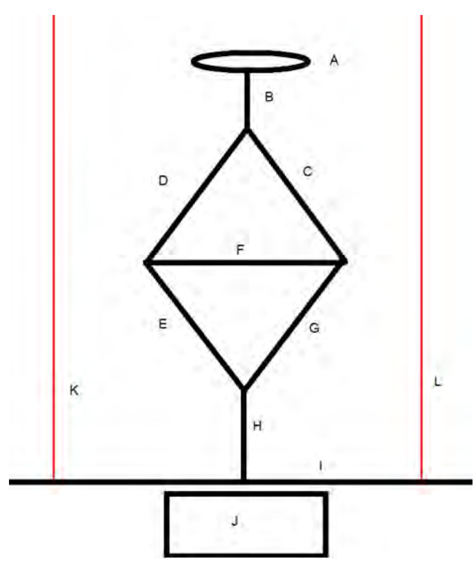

Fig. 2. Illustration of tether mechanism, where 'A' is a belt; 'B', 'C', 'D', 'E', 'G' and 'H' are ropes; 'F' is a bar; 'I' is the pool wall; 'J' is the starting block; 'K' and 'L' are racing lanes.

ISSN 2176-5480

Fig. 3. The dynamometer used to measure tethered force of the swimmer.

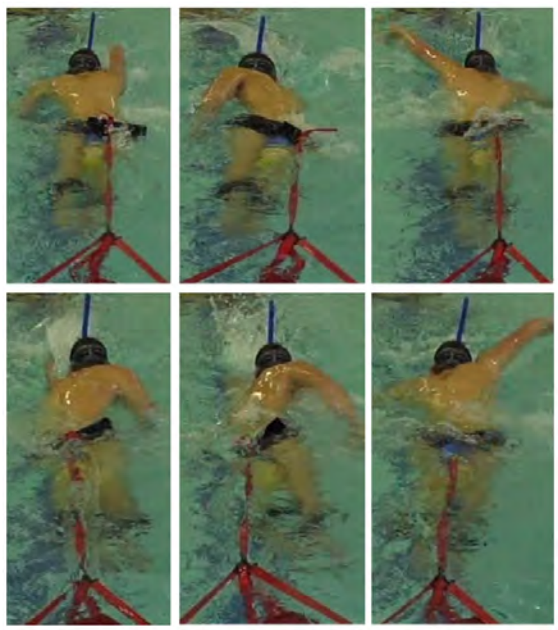

Fig. 4. Part of the tethered force test video realized with the swimmer.

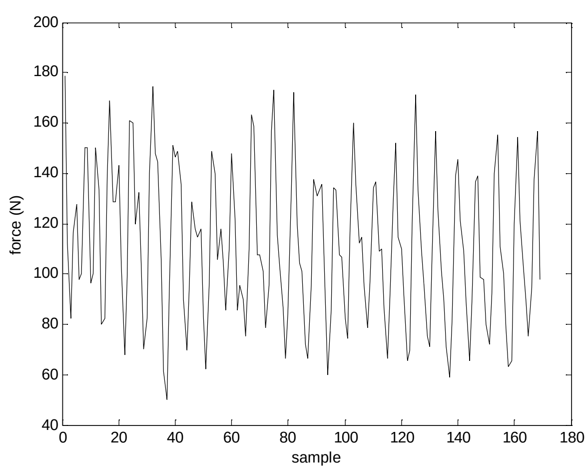

Fig. 5. Measured force data of the male swimmer during the tethered force test.

Table 1. Analyzed data (using a image processing software) of the male swimmer during the tethered force test.

Description	*Value*
Time to peak of force (18 N)	0.2960 s
Mean time to transition of best to worst arm	0.2806 s
Mean time to transition of worst to best arm	0.2879 s
Mean force (worst arm)	7.9902 N
Mean force (best arm)	13.0152 N
Maximum of the force signal	18.0001 N
Mean of the force signal	10.3706 N
Median of the force signal	10.3850 N
Variance of the force signal	7.4227 N

3.2 Forecasting results using RBF-NN

In the RBF-NN design, it was adopted the QPSO using $c_1 = c_2 = 2.05$ and β based on a linear reduction equation with initial and final values of 1 and 0.2. The setup adopted in the QPSO also includes the swarm size (population size) equal to 30 particles and the stopping criterion is 100 generations. The computational programs and functions of the RBF-NN design were run in Matlab® (MathWorks Inc., MA) on a 3.2 GHz Pentium IV processor with 8 MB of Random Access Memory.

The data that we used for our runs consist of 170 data points. Our method exploits the output dependence across time using a delay vector. It was adopted an input vector in the RBF-NN with three delayed outputs (force measurement) in a series-parallel setup.

As splitting criterion we have chosen approximately 59% of the data for training set and the remaining for validation. In other words, forecasting simulations were carried out for the estimation phase of the mathematical model using samples 1 to 100. For the validation phase, the RBF-NN model used signals of samples 101 to 171. The system identification by the RBF-NN model is appropriate if the values of the performance index are permissible for the user's needs. In this case, the objective is the maximization of the harmonic mean of multiple correlation indices of estimation and validation phases. That function (to be maximized) is calculated using the expression of R^2_{est} given by:

$$R^2_{est} = 1 - \frac{\sum_{t=1}^{100} [y(t) - \hat{y}(t)]^2}{\sum_{t=1}^{100} [y(t) - \bar{y}]^2} \tag{4}$$

where R^2_{est} is the multiple correlation index of the estimation phase, $y(t)$ is the output of the real system, $\hat{y}(t)$ is the output estimated by the RBF-NN, and $\bar{y}$ is the mean value of the system's output. For the validation phase (verification of generalization capability) of the RBF-NN, we employed the R^2_{val} index given by

$$R^2_{val} = 1 - \frac{\sum_{t=101}^{170} [y(t) - \hat{y}(t)]^2}{\sum_{t=101}^{170} [y(t) - \bar{y}]^2} \tag{5}$$

where R^2_{val} is the multiple correlation index of the validation phase. When the value R^2 =1.0 (estimation or validation phases), it indicates the model's accurate approach to the system's measured data. A R^2 value between 0.9 and 1.0 is considered sufficient for applications in forecasting and identification applications. QPSO used in the RBF-NN design deals the minimization of the sum of the equations (4) and (5).

Table 2 contains the forecasting results using RBF-NN design using QPSO in 30 runs. It can be seen that the RBF-NN design using 5 to 12 Gaussian functions in the hidden layer with obtain reasonably good results in terms of R^2_{est} and R^2_{val}.

Table 2. Forecasting results using RBF-NN design using QPSO in 30 runs.

Setup		R^2_{est} in 30 runs				R^2_{val} in 30 runs			
Test	*nG*	Maximum	Mean	Minimum	Standard Deviation	Maximum	Mean	Minimum	Standard Deviation
1	2	0.5377	0.4656	0.1304	0.0932	0.5960	0.5282	0.1849	0.1048
2	3	0.5460	0.5312	0.5104	0.0079	0.6196	0.5928	0.5697	0.0110
3	4	0.5960	0.5557	0.4640	0.0257	0.7216	0.6349	0.5887	0.0389
4	**5**	**0.6580**	0.6319	0.5980	0.0141	**0.8062**	0.7908	0.7404	0.0138
5	6	0.6571	0.6461	0.6032	0.0108	0.8125	0.8023	0.7833	0.0069
6	7	0.6587	0.6539	0.6410	0.0045	0.8126	0.8067	0.8005	0.0028
7	8	0.6592	0.6569	0.6504	0.0021	0.8151	0.8086	0.8048	0.0022
8	9	0.6590	0.6574	0.6551	0.0010	0.8121	0.8097	0.8078	0.0013
9	10	0.6588	0.6575	0.6559	0.0009	0.8119	0.8101	0.8083	0.0009
10	11	0.6603	0.6548	0.6358	0.0056	0.8155	0.8083	0.7888	0.0063
11	12	0.6615	0.6520	0.5437	0.0212	0.8141	0.8015	0.7477	0.0153

* *nG*: Number of Gaussian functions in the hidden layer of the RBF-NN

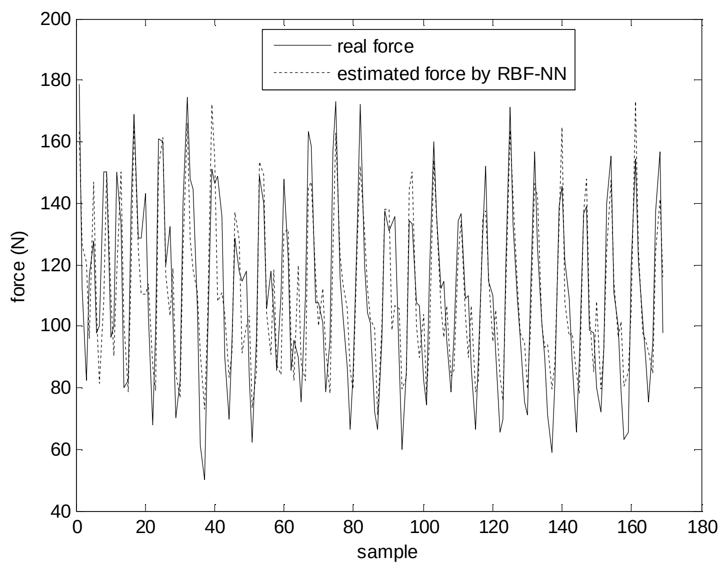

Fig. 6. Forecasting result using RBF-NN with *nG*=5 (best result in 30 runs).

4. CONCLUSION

An important property of the RBF-NN is that multidimensional space nonlinearity can be taken to be a linear combination of the nonlinear RBFs. This ability of learning the relation and structure of a multidimensional data by RBF-NN in a nonlinear way has been exploited in many engineering problems.

RBF-NNs have been trained by different learning algorithms in the literature. In this study we have proposed and investigated a RBF-NN design based on the QPSO approach used to force modeling of a Brazilian elite male swimmer. As the Table 2 shows, our proposed RBF-NN design can be useful to force modeling. Furthermore, the results obtained were used to establish a prediction model and monitoring for competitive level of fitness in swimmers, which is important to athletes and coaches.

In the future work the proposed RBF-NN scheme will be applied to other practical case studies related to swimming biomechanics.

5. ACKNOWLEDGEMENTS

This work was supported by the National Council of Scientific and Technologic Development of Brazil - CNPq - under Grants 307150/2012-7 and 476235/2011-1.

6. REFERENCES

Akis, T. and Orcan, Y., 2004, "Experimental and Analytical Investigation of the Mechanics of Crawl Stroke Swimming", Mechanics Research Communications, Vol. 31, No. 2, pp. 243-261.

Broomhead, D. and Lowe, D., 1988, "Multivariable Functional Interpolation and Adaptive Networks", Complex Systems, Vol. 2, pp. 321-355.

Clerc, M. and Kennedy, J.F., 2002, "The Particle Swarm: Explosion, Stability and Convergence in a Multi-dimensional Complex Space", IEEE Transactions on Evolutionary Computation, Vol. 6, pp. 58-73.

Coelho, L.S., 2008, "A Quantum Particle Swarm Optimizer with Chaotic Mutation Operator", Chaos, Solitons & Fractals, Vol. 37, pp. 1409-1418.

Dopsaj, M., Matković, I. and Zdravković, I., 2000, "The Relationship Between 50 m – Freestyle Results and Characteristics of Tethered Forces in Male Sprint Swimmers: A New Approach to Tethered Swimming Test", FACTA UNIVERSITATIS Series: Physical Education and Sport, Vol. 1, No. 7, pp. 15-22.

Li, X., Fei, M. and Irwin, G. W., 2012, "A Novel Locally Regularized Automatic Construction Method for RBF Neural Models", Neurocomputing, Vol. 98, pp. 4-11.

Eberhart, R.C. and Kennedy, J.F., 1995, "A New Optimizer Using Particle Swarm Theory", in Proceedings of International Symposium on Micro Machine and Human Science, Japan, pp. 39-45.

Gourgoulis, V., Aggeloussis, N., Vezos, N., Kasimatis, P. Antoniou, P. and Mavromatis, g., 2008, "Estimation of Hand Forces and Propelling Efficiency During Front Crawl Swimming with Hand Paddles", Journal of Biomechanics, Vol. 41, No. 1, pp. 208-215.

Hartigan, J. and Wang, M., 1979, "A K-means Clustering Algorithm", Applied Statistics, Vol. 28, pp. 100-108.

Karayiannis, N.B., 1999, "Reformulated Radial Basis Neural Networks Trained by Gradient Descent", IEEE Transactions on Neural Networks, Vol. 10, No. 3, pp. 2230-2235.

Kennedy, J.F. and Eberhart, R.C., 1995, "Particle Swarm Optimization", in Proceedings of IEEE International Conference on Neural Networks, Perth, Australia, pp. 1942-1948.

Kohonen, T., 1988, "An Introduction to Neural Computing", Neural Networks, Vol. 1, pp. 3-16.

MacQueen, J.B., 1967, "Some Methods for classification and Analysis of Multivariate Observations", Proceedings of 5th Berkeley Symposium on Mathematical Statistics and Probability, Berkeley, University of California Press, pp. 281-297.

Marinho, D.A., Barbosa, T. M., Mantha, V., Rouboa, A. and Silva, A.J., 2012, "Modelling Propelling Force in Swimming Using Numerical Simulations", in Fluid Dynamics, Computational Modeling and Applications. Rijeka: InTech, pp. 439-448.

Park, J. and Sandberg, I., 1991, "Universal Approximation using Radial Basis Function Networks", Neural Computing, Vol. 3, No. 2, pp. 246-257.

Sarimveis, H., Alexandridis, A. and Bafas, G., 2003, "A Fast Training Algorithm for RBF Networks Based on Subtractive Clustering", Neurocomputing, Vol. 51, pp. 501-505.

Simon, D., 2002, "Training Radial Basis Neural Networks with the Extended Kalman Filter", Neurocomputing, Vol. 48, Nos. 1-4, pp. 455-475.

Stamm, A., Thiel, D.V., Burkett, B. and James, D.A., 2011, "Towards Determining Absolute Velocity of Freestyle Swimming Using 3-Axis Accelerometers", Procedia Engineering, Vol. 13, pp. 120-125.

Sun, J., Xu, W. and Feng, B., 2005, "Adaptive Parameter Control for Quantum-behaved Particle Swarm Optimization on Individual Level", in Proceedings of IEEE International Conference on Systems, Man and Cybernetics, Big Island, HI, USA, pp. 3049-3054.

Sun, J., Xu, W.B. and Feng, B., 2004, "A Global Search Strategy of Quantum-Behaved Particle Swarm Optimization", in Proceedings of IEEE Conference on Cybernetics and Intelligent Systems, Singapore, pp. 111-116.

Yeater, R.A., Martin, R.B., White, M.K. and Gilson, K.H., 1981, "Tethered Swimming Forces in the Crawl, Breast and Back Strokes and their Relationship to Competitive Performance", Journal of Biomechanics, Vol. 14, No. 8, pp. 527-537.

7. RESPONSIBILITY NOTICE

The authors are the only responsible for the printed material included in this paper.