2021 18th International Conference on Electrical Engineering, Computing Science and Automatic Control (CCE)
Mexico City, Mexico November 10-12, 2021

Bio-Inspired Optimization to Improve Neural Identifiers for Discrete-time Nonlinear Systems

J. Felipe Guerra, Ramon Garcia-Hernandez, Miguel A. Llama
División de Estudios de Posgrado e Investigación
Tecnológico Nacional de México/I.T. La Laguna
Torreón, Coahuila de Zaragoza, México
Email: [m.jfguerrac, rgarciah, mllama]@correo.itlalaguna.edu.mx

***Abstract*—This work aims to apply metaheuristic optimization, offered by bio-inspired algorithms, to enhance the behavior of neural identifiers for unknown nonlinear systems in discrete-time whose model, for this purpose, is assumed unknown. Due to this fact, a new and efficient training algorithm based on the UKF approach is used in combination with the bio-inspired algorithms that are used in order to tune the covariance matrices for the training algorithm. Simulations on a two degree of freedom robot arm are performed obtaining encourage results.**

***Index Terms*—neural networks, Kalman filters, optimization, metaheuristics**

I. Introduction

In the middle of the 19th century, two great scientists conceived the meaning of evolution, each one of them working on equivalent experiments, but without any manner of mutual collaboration. On the one hand, Charles Darwin published his book *On the origin of species* in 1859, being recognized as the father of evolutionary biology, solving extremely complex evolutionary problems and, above all, offering the most convincing how evolution happens. However, it was the work of Gregor Johann Mendel *Plant hybridization experiments* who allowed us to know the mechanisms of inheritance, giving explanations missing in Darwin's work, explaining how natural selection works, to what would later be known as Mendel's laws of inheritance. These concepts, together with those specifically related to evolutionary biology, founded the concept of bio-inspired algorithms [1].

One of the most prominent fields of artificial intelligence is evolutionary computation due to its versatility in solving problems from multiple techniques instituted in the principles of biological evolution [2]. For this reason, it is not surprising that its use has become widespread in solving optimization problems.

As its name indicates, evolutionary algorithms (EA) are optimization methods corresponding to an outstanding concept of evolutionary computation, yielded by models based on biology. A population of possible solutions, in EAs, consists of individuals with the ability to compare themselves concerning their ability to improve the solution. In addition, according to an objective function (or fitness function), those who obtain better evaluations are selected as the most qualified candidates. In this way, it takes place to the evolution of the population a series of operations (reproduction, mutation, recombination, or selection), which, applied to the individuals of the population iteratively, emulates the process of natural selection and biological evolution. The use of the previously mentioned operations maintains a high probability of generating better solutions. The way the population evolves possible solutions and chooses the new best global solutions is inherent in every EA.

No one is surprised that the number of algorithms proposed is increasing day by day, so carrying out a study on all the optimization methods based on EAs that fall into the category of bio-inspired algorithms is practically impossible. Due to this, it is necessary to classify and delimit the bio-inspired algorithms, which for this work will be separated into methods based on swarm intelligence (SI) and based on evolutionary computation (EC).

Figure 1 shows an example of the classification of metaheuristic algorithms inspired by nature, divided into two classes. In the first instance, there are algorithms strongly related to Mendel's laws of inheritance and natural selection, for example, genetic algorithms (GA) [3] and differential evolution (DE) [4]; It also includes novel algorithms such as the Black Widow Optimization Algorithm (BWO) [5] and the Polar Bear Optimization Algorithm (PBO) [6], which also share certain abilities of swarm intelli-

The authors are grateful for the financial support of the projects of investigation of the Tecnológico Nacional de México/I. T. La Laguna and CONACYT 860723.

gence, all this, however, does not take away their core of operation within the basic concepts of evolution.

In the case of swarm intelligence (SI) algorithms are based on the population behavior of swarms (herds, colonies, etc.) that occur in nature [7]; some of the most prominent algorithms were included such as particle swarm optimization (PSO) [8], ant colony algorithm (ACO) [9], artificial bee colony algorithm (ABC) [10], the bat algorithm (BA) [11], the cuckoo search (CS) [12], and the firefly algorithm (FA) [13]. Also included are methods that derive from a combination of evolutionary techniques and swarm intelligence, such as the gray wolf optimizer (GWO) [14], the ant-lion optimizer (ALO) [15], the optimization flame-moth (MFO) [16], whale optimization algorithm (WOA) [17] and accelerated particle swarm optimization (APSO) [18].

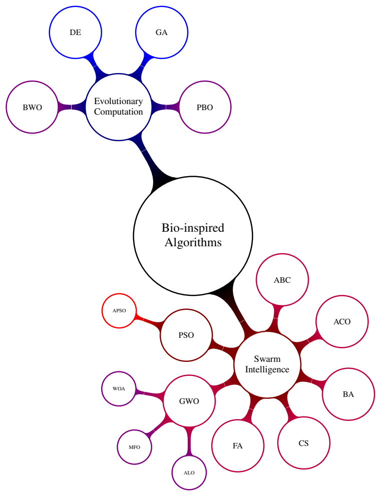

Fig. 1: Examples of bio-inspired algorithms

As in [19], the neural identification of a linear induction motor is proposed through an algorithm based on the extended Kalman filter (EKF), using particle swarm optimization to improve its performance. It is worth noticing that EKF is one of the most widely used filtering methods for estimating nonlinear systems, even though this filter has two serious drawbacks that make it difficult to apply [20]. First, it does not take into account the probabilistic uncertainty of the random variables of the system and the noise that contaminates it when performing the linearization. Second, the precision of the covariance and the propagated mean is limited to first order, due to the use of the linearization method based on truncated first order Taylor series. The Unscented Kalman filter (UKF) was developed to overcome the limitations of the EKF owing to the first order linearization of nonlinear systems [21].

For this work, the identification schemes based on the Unscented Kalman filter was used together with the bio-inspired algorithms.

II. Preliminaries

Let be a class of discrete-time nonlinear and interconnected system, which may be represented in the nonlinear block-controllable (NBC) form, incorporated by r blocks

$$\begin{aligned} \chi_i^1(k+1) &= f_i^1\left(\chi_i^1\right) + B_i^1\left(\chi_i^1\right)\chi_i^2 + \Gamma_{i\ell}^1, \\ \chi_i^2(k+1) &= f_i^2\left(\chi_i^1,\chi_i^2\right) + B_i^2\left(\chi_i^1,\chi_i^2\right)\chi_i^3 + \Gamma_{i\ell}^2, \\ &\vdots \\ \chi_i^r(k+1) &= f_i^r\left(\boldsymbol{\chi}_i\right) + B_i^r\left(\boldsymbol{\chi}_i\right)u_i + \Gamma_{i\ell}^r \end{aligned} \tag{1}$$

where $\boldsymbol{\chi}_i \in \mathbb{R}^{n_i}$, $\chi_i^j \in \mathbb{R}^{n_{ij}\times 1}$, $i = 1,\dots,N$, $l = 1,\dots,n_{ij}$. N is the number of subsystems and $u_i \in \mathbb{R}^{m_i}$ is the input vector. f_i^j, B_i^j and Γ_i^j are assumed smooth and bounded functions, with $f_i^j(0) = 0$ and $B_i^j(0) = 0$; in addition $n_{i1} \le n_{i2} \le \cdots \le n_{ij} \le m_i$ define the different subsystem structures. The interconnection terms are given by

$$\begin{aligned} \Gamma_{i\ell}^1 &= \sum_{\ell=1,\ell\neq i}^{N} \gamma_{i\ell}^1\left(\chi_\ell^1\right), \\ &\vdots \\ \Gamma_{i\ell}^r &= \sum_{\ell=1,\ell\neq i}^{N} \gamma_{i\ell}^r\left(\boldsymbol{\chi}_\ell\right) \end{aligned} \tag{2}$$

where $\boldsymbol{\chi}_\ell = \left[\chi_\ell^1,\chi_\ell^2,\cdots,\chi_\ell^j\right]^\top$ represents the state vector of the ℓ-th subsystem with $1 \le \ell \le N$ and $\ell \neq i$. Terms in (2) reflect the interaction between the i-th subsystem and the other ones.

A. *Discrete-time Recurrent High Order Neural Networks (RHONNs).*

The following decentralized discrete-time RHONN structure is proposed to identify (1):

$$\begin{aligned} x_i^1(k+1) &= w_i^1(k)S\left(\chi_i^1(k)\right) + w_i'^1\chi_i^2(k), \\ x_i^2(k+1) &= w_i^2(k)S\left(\chi_i^1(k),\chi_i^2(k)\right) + w_i'^2\chi_i^3(k), \\ &\vdots \\ x_i^r(k+1) &= w_i^r(k)S\left(\chi_i^1(k),\cdots,\chi_i^r(k)\right) + w_i'^r u_i(k) \end{aligned} \tag{3}$$

where $x_i^j(k+1) = \begin{bmatrix} x_i^1 & x_i^2 & \cdots & x_i^r \end{bmatrix}^\top$ is the j-th block neuron state with $i = 1,\dots,N$ and $j = 1,\dots,r$; $w_i'^j$ are fixed parameters with $rank(w_i'^j) = n_{ij}$; $S(\bullet)$ is the activation function, and $u_i(k)$ represents the input vector.

The effects of the interconnection terms are compensated by the neural network weights update.

B. *UKF learning law*

The search for optimal weight values $w_i^j(k)$ that mitigate the prediction error is the training task. Therefore, this work aims to implement a new and efficient training algorithm built on the UKF framework described by

$$\begin{aligned}K_i^j(k) &= P_i^{jxy}(k)\left(P_i^{jyy}(k)\right)^{-1},\\ w_i^j(k+1) &= w_i^j(k) + K_i^j(k)e_i^j(k),\\ P_i^j(k+1) &= P_i^j(k) - K_i^j(k)P_i^{jyy}(k)K_i^j(k)^\top,\end{aligned} \tag{4}$$

with

$$\begin{aligned}P_{i,k}^{jxy} &= \sum_{i=0}^{2L}\eta_i^c[\mathcal{X}_{i,k|k-1}^j - \hat{x}_{i,k}^{j-}][\mathcal{Y}_{i,k|k-1}^j - \hat{y}_{i,k}^{j-}]^T\\ P_{i,k}^{jyy} &= R_{i,k}^j + \sum_{i=0}^{2L}\eta_i^{jc}[\mathcal{Y}_{i,k|k-1}^j - \hat{y}_{i,k}^{j-}][\mathcal{Y}_{i,k|k-1}^j - \hat{y}_{i,k}^{j-}]^T\\ P_{i,k}^{j} &= Q_{i,k}^j + \sum_{i=0}^{2L}\eta_i^{jc}[\mathcal{X}_{i,k|k-1}^j - \hat{x}_{i,k}^{j-}][\mathcal{X}_{i,k|k-1}^j - \hat{x}_{i,k}^{j-}]^T\\ e_{i,k}^j &= [\chi_{i,k}^j - x_{i,k}^j]\end{aligned} \tag{5}$$

where $e_i^j(k)$ is the identification error, $P_i^j(k)$ is the prediction error covariance matrix, $P_i^{jyy}(k)$ is the covariance of predicted output matrix, $P_i^{jxy}(k)$ is the cross-covariance of state and output matrix; $w_i^j(k)$ is the j-the weight (state) of the i-th subsystem, η_i^{jc} is a design parameter, $\chi_i^j(k)$ is the j-th plant state and $x_i^j(k)$ is the j-th neural network state. L is the number of states, $K_i^j(k)$ is the Kalman gain matrix, $Q_i^j(k)$ is the measurement noise covariance matrix, $R_i^j(k)$ is the state noise covariance matrix, $\hat{x}_i^{j-}(k)$ is the mean of predicted state, $\hat{y}_i^{j-}(k)$ is the mean of predicted output, $\mathcal{X}_i^j(k|k-1)$ and $\mathcal{Y}_i^j(k|k-1)$ are propagation of sigma-points through prediction and observation respectively. It is important to remark that $K_i^j(k)$, $P_i^j(k)$, $R_i^j(k)$ and $Q_i^j(k)$ for the UKF are bounded.

Finding the values of $w_i^j(k)$, requires to define the dynamics of the prediction error in (5) expressed as

$$e_i^j(k+1) = \tilde{w}_i^j(k)z_i^j(k) + \epsilon_{z_i^j}. \tag{6}$$

On the other hand, the dynamics of weight estimation error $\tilde{w}_i^j(k)$ is

$$\tilde{w}_i^j(k+1) = \tilde{w}_i^j(k) - K_i^j(k)e_i^j(k). \tag{7}$$

In [22] is defined the stability analysis for the i-th subsystem of the RHONN (3) and the identification of the i-th subsystem of the nonlinear system (1).

III. Neural Identifiers Optimization

Usually P_i^j, Q_i^j, and R_i^j are initialized as diagonal matrices with entries $P_i^j(0)$, $Q_i^j(0)$, and $R_i^j(0)$, respectively. Given that typically this entries are defined heuristically, in this section, actual work propose the use of a multiple bio-inspired algorithms in order to compute on-line such entries to improve the UKF training algorithm, as explained below.

A. Ant Lion Optimizer

In [15] the author introduces a method based on the larva of the ant lion to solve optimization problems. This insect, also known as Myrmeleon, is part of the Myrmelentidae family and is widely found in the hottest regions of the world. The ant lion has two stages during its life: larval and adult. This insect is best described by its name *ant lion* because of the technique it uses to hunt other insects in their larval stage. This process can be described in the following stages:

1) The ant lion makes funnel holes in soft sand by digging backward, in this process, it makes funnel-shaped spiral trails of different sizes.
2) The larva waits patiently at the bottom of the trap for its prey.
3) As it slides down the funnel, the prey is caught and captured by the ant lion. If an attempt is made to escape through a funnel, the larva will throw sand on the edge of the trap to prevent the prey from escaping.

Theoretically, the ALO algorithm claim to improve solutions due to the following reasons:

- The random selection of ant lion and random walks around it guaranteed the search space exploration.
- Shrinking and adaptive limits of ant lion traps ensured the exploitation of the search space.
- A population-based algorithm has an intrinsically high ability to avoid stagnation in a local optimum.
- The best ant lion from each iteration is compared to the best ant lion obtained so far (elite).

All these characteristics, joint with a simple structure, as shown in Figure 2, make the ALO algorithm a viable option in solving engineering problems [23].

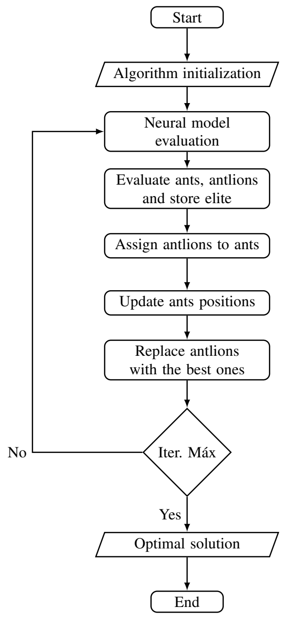

Fig. 2: ALO Algorithm

B. Black Widow Optimization Algorithm

The novel metaheuristic black widow optimization algorithm (BWOA) proposed in [5] is suitable for

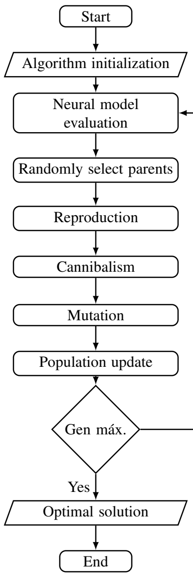

Fig. 3: BWOA flowchart

nonlinear continuous optimization problems. Based on the mating behavior of black widow spiders and including the cannibalism of the species, the algorithm presents unique traits. The inclusion of this stage allows for the screening of individuals with inapposite aptitude, which in theory leads to a quick convergence. Figure 3 shows the flow diagram of the BWOA algorithm. Every one of the spiders that make up the initial population represents a potential solution. The new generations will arise from the attempted mating of pairs of spiders in the initial population. As its name implies, the female spider eats the male during or after mating. It is then released into egg sacs and, 11 days after laying, the young hatch from the egg sacks. Coexistence as newborns after hatching, the phase of cannibalism between congeners takes place. Finally, the addition of the mutation factor increases the randomness, ending this process, which is repeated until the optimal solution is reached.

IV. Experimental evaluation for a two DOF robot manipulator

For this work, the identification schemes based on the nonlinear Kalman filters used in previous works were used [24], [25], together with the bio-inspired algorithms mentioned in previous sections.

The implementation of bio-inspired algorithms in nonlinear systems to improve neural identifiers in discrete-time is illustrated using a two DOF robot manipulator.

The aforementioned robot is illustrated in Figure 4 and in the Table I are included the numerical values of the different parameters.

In pursuit of identifying the dynamic model of the robot, several authors [26], [27] recommend using a

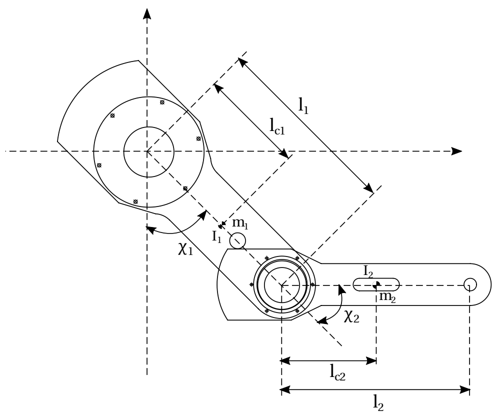

Fig. 4: Two DOF robot manipulator

TABLE I: Two DOF robot manipulator parameters

Description	Notation	Value	Unit
Mass link 1	m_1	23.902	[kg]
Mass link 2	m_2	3.880	[kg]
Length link 1	l_1	0.450	[m]
Length link 2	l_2	0.450	[m]
Inertia link 1	I_1	1.266	[kg·m^2]
Inertia link 2	I_2	0.093	[kg·m^2]
Center of mass link 1	l_{c1}	0.091	[m]
Center of mass link 2	l_{c2}	0.048	[m]

decentralized series-parallel NN for each joint defined by (3).

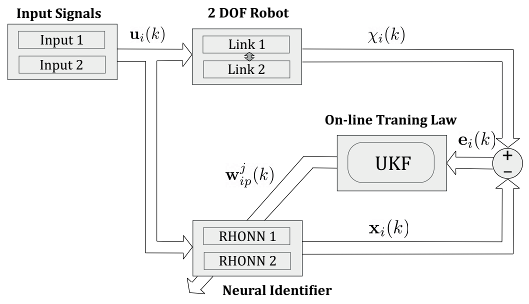

Fig. 5: Neural identification scheme

Using an identification scheme as shown in Figure 5 and an objective function similar to the one proposed in [28]

$$f_{obj} = \varrho_1 MSE(e_i(k)) + \sum_{k=0}^{\frac{T}{t_s}} \left[\varrho_2 \left(u_i(k-1) - u_i(k)\right)\right] \quad (8)$$

where T is the total simulation time, t_s is the sampling period, ϱ_1 and ϱ_2 are scale factors meant to bring all the terms of the objective function to a similar order. The term MSE refers to the mean squared error of the identification errors, which is calculated as:

$$MSE = \frac{T}{t_s} \sum_{k=0}^{\frac{T}{t_s}} \left[\chi_i(k) - x_i(k)\right]^2$$

In the performance comparison between the different algorithms, it was necessary to use the same conditions as indicated in [29]. In these tests, 25 iterations and a population of 50 were used, performing 25 simulations with these characteristics for each algorithm. Both algorithms have restrictions limiting the search space in a closed set that ensures that matrices $P_i^j(k)$, $Q_i^j(k)$, and $R_i^j(k)$ are positive definite, avoiding producing a malfunction in the UKF algorithm.

Figures 6 and 7 show the input signals and respective positions for each of the links. The identification errors are presented in Figures 8 to 11. Table II presents the results of the performance of the fitness function in the bio-inspired algorithms ALO and BWO, comparing them with a selection of heuristic parameters (without optimization). In addition, Figure 12 is included with the convergence curves, where the evolution of the performance of the algorithms is appreciated.

Worth noting that the computational cost of the SI algorithms is at least ten times lower than most of the EC methods; e. g. the ALO algorithm averaged 3 minutes of CPU usage, as opposed to the BWO algorithm, which averaged 80 minutes of CPU usage.

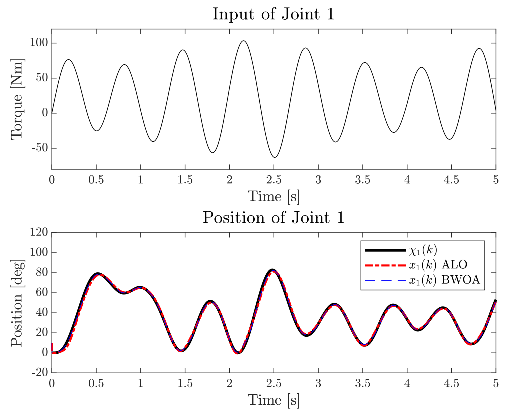

Fig. 6: Input and position of Joint 1

V. Conclusions

From the results obtained from the two bio-inspired algorithms it is possible to conclude the following:

- The poor performance of the PBO and FA algorithms is mainly due to the number of individuals in the initial population and the number of iterations performed.
- Generally, SI algorithms are faster than EC algorithms. However, the DE algorithm has an acceptable computational cost and showing a headlong convergence.
- Analysis of convergence curves in Figure 12 indicates which algorithms tend to be more exploitable than exploratory. For example, the EC

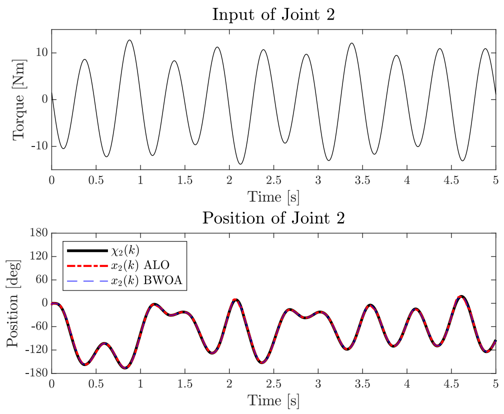

Fig. 7: Input and position of Joint 2

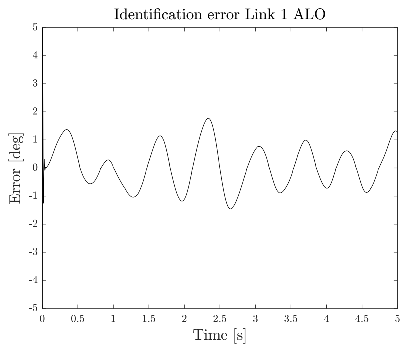

Fig. 8: Identification error Link 1 with ALO

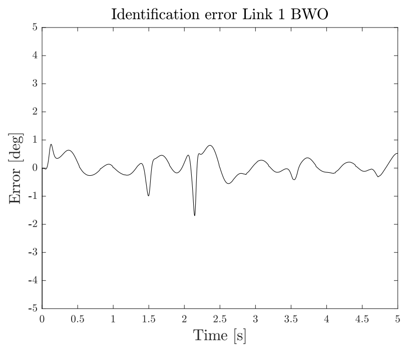

Fig. 9: Identification error Link 1 with BWOA

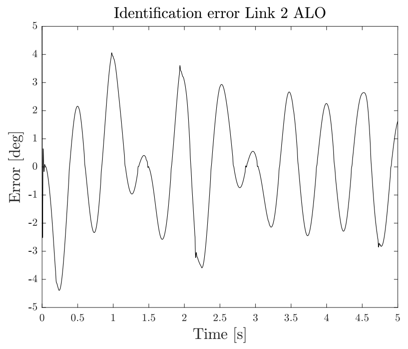

Fig. 10: Identification error Link 2 with ALO

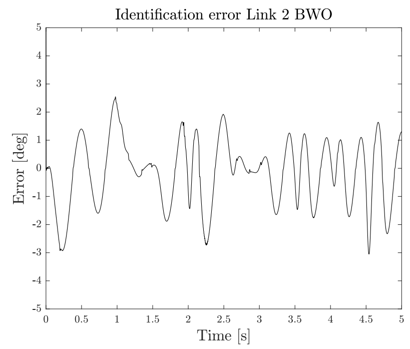

Fig. 11: Identification error Link 2 with BWOA

algorithms tend more to the solution exploitation than the SI methods that constantly explore the search space.

Algorithm	w/o Opt.	ALO	BWO
f_{obj}	0.2865	0.0988	0.0593

TABLE II: Performance of bio-inspired algorithms

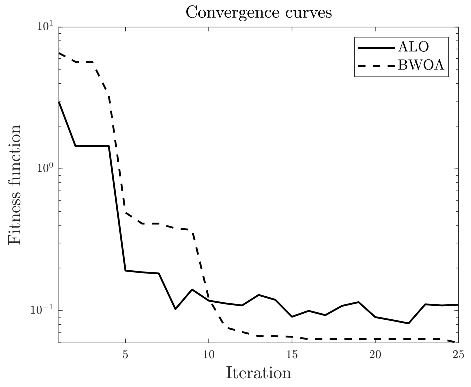

Fig. 12: Convergence of the bio-inspired algorithms

References

[1] N. Bizzo and C. N. El-Hani, "Darwin and mendel: evolution and genetics," *Journal of Biological Education*, vol. 43, no. 3, pp. 108–114, 2009.

[2] D. Simon, *Evolutionary optimization algorithms*. John Wiley & Sons, 2013.

[3] J. H. Holland, *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence*. MIT press, 1992.

[4] K. Price, R. M. Storn, and J. A. Lampinen, *Differential evolution: a practical approach to global optimization*. Springer Science & Business Media, 2006.

[5] V. Hayyolalam and A. A. P. Kazem, "Black widow optimization algorithm: A novel meta-heuristic approach for solving engineering optimization problems," *Engineering Applications of Artificial Intelligence*, vol. 87, p. 103249, 2020.

[6] D. Polap and M. Wozniak, "Polar bear optimization algorithm: Meta-heuristic with fast population movement and dynamic birth and death mechanism," *Symmetry*, vol. 9, no. 10, p. 203, 2017.

[7] Y. Shi and R. C. Eberhart, "Empirical study of particle swarm optimization," in *Proceedings of the 1999 congress on evolutionary computation-CEC99 (Cat. No. 99TH8406)*, vol. 3, pp. 1945–1950, IEEE, 1999.

[8] R. Eberhart and J. Kennedy, "Particle swarm optimization," in *Proceedings of the IEEE international conference on neural networks*, vol. 4, pp. 1942–1948, Citeseer, 1995.

[9] K. Socha, "Aco for continuous and mixed-variable optimization," in *International Workshop on Ant Colony Optimization and Swarm Intelligence*, pp. 25–36, Springer, 2004.

[10] D. Karaboga and B. Basturk, "A powerful and efficient algorithm for numerical function optimization: artificial bee colony (abc) algorithm," *Journal of global optimization*, vol. 39, no. 3, pp. 459–471, 2007.

[11] X.-S. Yang and X. He, "Bat algorithm: literature review and applications," *International Journal of Bio-inspired computation*, vol. 5, no. 3, pp. 141–149, 2013.

[12] X.-S. Yang and S. Deb, "Cuckoo search via lévy flights," in *2009 World congress on nature & biologically inspired computing (NaBIC)*, pp. 210–214, Ieee, 2009.

[13] X.-S. Yang, "Firefly algorithms for multimodal optimization," in *International symposium on stochastic algorithms*, pp. 169–178, Springer, 2009.

[14] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey wolf optimizer," *Advances in engineering software*, vol. 69, pp. 46–61, 2014.

[15] S. Mirjalili, "The ant lion optimizer," *Advances in engineering software*, vol. 83, pp. 80–98, 2015.

[16] S. Mirjalili, "Moth-flame optimization algorithm: A novel nature-inspired heuristic paradigm," *Knowledge-based systems*, vol. 89, pp. 228–249, 2015.

[17] S. Mirjalili and A. Lewis, "The whale optimization algorithm," *Advances in engineering software*, vol. 95, pp. 51–67, 2016.

[18] X.-S. Yang, *Engineering Optimization: An Introduction with Metaheuristic Applications*. John Wiley & Sons, July 2010. Google-Books-ID: kTi6ul2g9VYC.

[19] A. Y. Alanis, N. Arana-Daniel, and C. Lopez-Franco, *Bio-inspired algorithms for engineering*. Butterworth-Heinemann, 2018.

[20] S. Haykin, *Kalman filtering and neural networks*, vol. 47. John Wiley and Sons, 2004.

[21] E. A. Wan and R. Van Der Merwe, "The unscented kalman filter for nonlinear estimation," in *Proceedings of the IEEE 2000 Adaptive Systems for Signal Processing, Communications, and Control Symposium*, pp. 153–158, 2000.

[22] R. Garcia-Hernandez, M. Lopez-Franco, E. N. Sanchez, A. Y. Alanis, and J. A. Ruz-Hernandez, *Decentralized Neural Control: Application to Robotics*, vol. 96 of *Studies in Systems, Decision and Control*. Switzerland: Springer, 2017.

[23] M. Mani, O. Bozorg-Haddad, and X. Chu, "Ant lion optimizer (alo) algorithm," in *Advanced optimization by nature-inspired algorithms*, pp. 105–116, Springer, 2018.

[24] R. García-Hernández, E. N. Sanchez, V. Santibáñez, M. A. Llama, and E. Bayro-Corrochano, "Real-time decentralized neural block controller for a robot manipulator," in *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 906–911, IEEE, 2009.

[25] J. Guerra, R. Garcia-Hernandez, and M. Llama, "Decentralized neural block control for a robot manipulator based in ukf training," in *2019 International Conference on Electronics, Communications and Computers (CONIELECOMP)*, pp. 88–94, IEEE, 2019.

[26] R. A. Felix, E. N. Sanchez, and A. G. Loukianov, "Avoiding controller singularities in adaptive recurrent neural control," *IFAC Proceedings Volumes*, vol. 38, no. 1, pp. 109–114, 2005.

[27] E. N. Sanchez, A. Y. Alanís, and A. G. Loukianov, *Discrete-time high order neural control*. Springer, 2008.

[28] M. Llama, A. Flores, R. Garcia-Hernandez, and V. Santibañez, "Heuristic global optimization of an adaptive fuzzy controller for the inverted pendulum system: Experimental comparison," *Applied Sciences*, vol. 10, no. 18, p. 6158, 2020.

[29] M. Mernik, S.-H. Liu, D. Karaboga, and M. Črepinšek, "On clarifying misconceptions when comparing variants of the artificial bee colony algorithm by offering a new implementation," *Information Sciences*, vol. 291, pp. 115–127, 2015.