AiMT Advances in Military Technology
Vol. 11, No. 1, June 2016

Influence of State Space Topology on Parameter Identification Based on PSO Method

M. Dub[1*] and A. Štefek[2]

[1] Department of Aircraft Electrical Systems, University of Defence, Brno, Czech Republic
[2] Department of Air Defence Systems, University of Defence, Brno, Czech Republic

The manuscript was received on 30 November 2015 and was accepted after revision for publication on 19 May 2016.

Abstract:

Motion control of electromechanical systems still plays a very important role in a wide area of weapon systems. Modern control systems use not only data from various sensors, but also state parameters of controlled system. The article explores the influence of state space topology on parameter identification of real simple electromechanical system based on the Particle Swarm Optimization (PSO) method. Four different but equivalent mathematical models of the second order were used to create different state spaces of the system parameters. A general recommendation for the PSO method setup and two independent program tools were applied to evaluate the state space searching by the PSO method. The PSO simulations were focused on both narrow and wide state spaces around the fitness function global minimum. A novel approach to set the PSO method initial agents' positions has been introduced since the traditional random uniform distribution failed when wide state spaces were used.

Keywords:

Parameter estimation, particle swarm optimization, system identification

1. Introduction

With the increasing power of computers, the stochastic optimization methods became very popular. One of the first methods which were implemented was the Particle Swarm Optimization Method (PSO). Even though many other methods were introduced, the PSO method still remains one of the best methods [1, 2].

Various modifications of the PSO method have been introduced by many authors either to improve the PSO method itself or to fuse the PSO method with other optimization methods, recently e.g. [3 – 6].

---

[*] *Corresponding author: Department of Aircraft Electrical Systems, Faculty of Military Technology, University of Defence, Kounicova 65, 662 10 Brno, Czech Republic. Phone: +420-973-44 50 61, Fax: +420-973-445235, E-mail: michal.dub@unob.cz*

This article proposes the hypothesis that the traditional random uniform distribution used for the PSO method initial agents' positions can fail within a considerable wide state space. The idea is applied on different but comparable state spaces to evaluate the influence of state space topology on the behaviour of the PSO method. Mutual transformation of state spaces is also considered as a base for advanced simulations.

Modelling the swarm moving across a state space is represented by a set of simple equations Eq. (1), (2). Each step of the simulation is followed by calculating fitness functions and reporting the good and the best agents' positions.

$$v_{k+1} = c_1 v_k + r_2 c_2 \left(x_p - x_k\right) + r_3 c_3 \left(x_s - x_k\right), \tag{1}$$

$$x_{k+1} = x_k + v_{k+1} . \tag{2}$$

In the automation control theory, a transfer function is widely used as an easy description of any realistic linear system. However, the identification of the transfer function parameters still remains the main challenge, even if the transfer function structure is known. A system time response can be used for both determined and stochastic identification. For the purpose of this article, only the system of second order will be considered. These systems can be described by transfer functions with various state space topology based on its parameters [7, 8].

The quality of the identification process called the fitness function is given by the difference between the actual measured system response and the mathematical model simulated response. The fitness function can be described by various mathematical functions. One of the possible formulas for an integral fitness function is Eq. (3). For a simulation, the integral fitness function can be transformed into the form of Eq. (4).

$$L(m) = \int_0^T \left(y_m(t) - y_r(t)\right)^2 dt , \tag{3}$$

$$L(m) = \sum_0^{n-1} \left(y_m(k\Delta t) - y_r(k\Delta t)\right)^2 \Delta t . \tag{4}$$

A simple electromechanical system is represented by a separately excited DC motor. The system data measured were both the input armature voltage and the output rotational speed, while the nominal voltage step function was used as a test input function. Rotational speed was measured by a DC tachogenerator, and therefore, ripple voltage and additional noise can be seen in Fig. 1. Voltages were measured with a sampling frequency of 20 kHz. The effect of measurement errors, e.g. aliasing and quantisation errors of A/D converters, on resulting digital value of rotational speed was not analysed, as the paper is focused on optimisation methods.

Four equivalents (A), (B), (C), and (D) were used as three-parameter models for the PSO implementation in such a way that model (A) was chosen as the primary model, while models (B), (C) and (D) were derived from it (A). Moreover, each model has its own state space based on its parameters.

Tab. 1 shows the parameters of all models in detail. The transformation equations can be used for direct transformation from the primary model (A) into the other models (B), (C) and (D). The reversal transformations from other models to primary model (A) can lead to complex numbers of primary model (A) parameters even if other models initial parameters are randomly chosen real numbers.

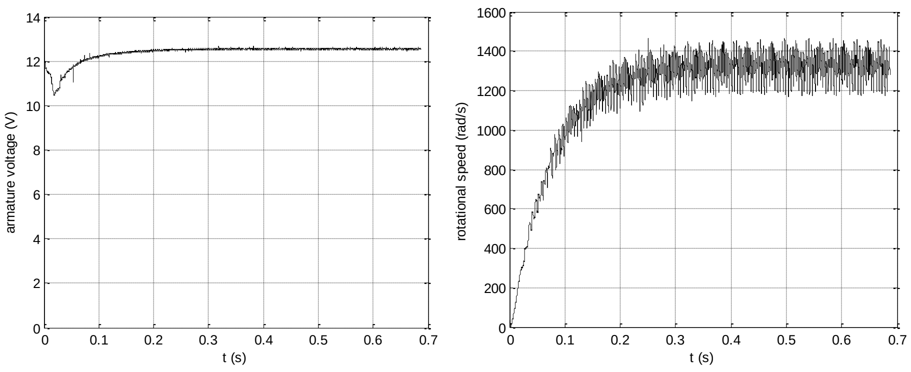

Fig. 1 Real time system input armature voltage and output rotational speed

Tab. 1 Models of simple electromechanical systems

Model	Transfer function	State vector
A	$F(s)=\dfrac{K_A}{(T_{1A}+s)(T_{2A}+s)}$	$[K_A, T_{1A}, T_{2A}]$
B	$F(s)=\dfrac{K_B}{T_{1B}T_{2B}s^2+(T_{1B}+T_{2B})s+1}$	$[K_B, T_{1B}, T_{2B}]$
C	$F(s)=\dfrac{K_C}{T_{1C}T_{2C}s^2+T_{2C}s+1}$	$[K_C, T_{1C}, T_{2C}]$
D	$F(s)=\dfrac{K_D}{T_{1D}s^2+T_{2D}s+1}$	$[K_D, T_{1D}, T_{2D}]$

2. State Space Topology

The PSO method is a stochastic method, so setting up the initial conditions for different models is rather complicated. Random seed settings and model transformations were used to remove the stochastic part of the experiments, since they ensure the same position of agents at the beginning of each experiment with all the different models (A), (B), (C) and (D). They also guarantee that the only factor that has an impact on the results is the state space structure.

The PSO method has three parameters that can vary. Setting for the PSO method given by Eq. (5), (6) was confirmed as the (sub)optimal and only those PSO coefficients were later used in all experiments [9].

$$v_{k+1}=0.7v_k+1.4r_2(x_p-x_k)+1.4r_3(x_s-x_k), \tag{5}$$

$$x_{k+1}=x_k+v_{k+1}. \tag{6}$$

As the PSO is a method that has no general rule for ending the optimization process, not each one of them finds the optimal solution. Moreover, the PSO method very often finds just a suboptimal solution. Therefore, the PSO optimization results were stored throughout many different stages and snapshots were taken at 25th, 50th, 100th, 150th and 200th step of the optimization process. During the experimental phase, dozens of experiments were calculated and stored. For later evaluation, the database was defined with the following structure:

- Random generator seed.
- Model type (A), (B), (C) or (D).
- Number of steps in the optimization process.
- Fitness function value.
- Calculated suboptimum.

The fitness function global minimum was found at different locations from the system state vectors and its value was not only confirmed a number of times during the conducted experiments, but it was also calculated independently by the Nelder-Mead simplex search method using the MATLAB program tool. The value of the fitness function global minimum is:

$$L(m) = 2\,343.174412\ . \tag{7}$$

The PSO simulations were performed in two different state spaces of the initial estimates, a narrow and a wide state space around fitness function global minimum, see Tab. 2. Both state spaces were derived from the optimum global minimum given by Eq. (7), transformed into state space of each model. The narrow state space limits were calculated as one order higher and one order lower than the magnitude of each element of the model state vector. The wide state space limits were calculated in a similar way as five orders higher and five orders lower.

Models (A) and (B) have within their state spaces two optima, as it is shown in Tab. 1 and Tab. 2. This suggests that those models would be much more successful than models (C) and (D). The global minimums and their close proximities were graphically displayed in the MATLAB program for each model. State space of model (A), shown in Fig. 2 (on the left), is totally different from the others. The state spaces of models (C) and (D) are "smooth" as the state space of model (B) shown in Fig. 2 (on the right).

Tab. 2 State space limits

Model	Global minimum	Narrow state space	Wide state space
A	$4.8831\cdot10^{5}$ $2.8248\cdot10^{2}$ $1.6224\cdot10^{1}$	$<10^{4}; 10^{6}>$ $<10^{1}; 10^{3}>$ $<10^{0}; 10^{2}>$	$<10^{0}; 10^{10}>$ $<10^{-3}; 10^{7}>$ $<10^{-4}; 10^{6}>$
B	$1.0655\cdot10^{2}$ $6.1637\cdot10^{-2}$ $3.5401\cdot10^{-3}$	$<10^{1}; 10^{3}>$ $<10^{-3}; 10^{-1}>$ $<10^{-4}; 10^{-2}>$	$<10^{-3}; 10^{7}>$ $<10^{-7}; 10^{3}>$ $<10^{-8}; 10^{2}>$
C	$1.0655\cdot10^{2}$ $3.3481\cdot10^{-3}$ $6.5179\cdot10^{-2}$	$<10^{1}; 10^{3}>$ $<10^{-4}; 10^{-2}>$ $<10^{-3}; 10^{-1}>$	$<10^{-3}; 10^{7}>$ $<10^{-8}; 10^{2}>$ $<10^{-7}; 10^{3}>$
D	$1.0655\cdot10^{2}$ $2.1820\cdot10^{-4}$ $6.5179\cdot10^{-2}$	$<10^{1}; 10^{3}>$ $<10^{-5}; 10^{-3}>$ $<10^{-3}; 10^{-1}>$	$<10^{-3}; 10^{7}>$ $<10^{-9}; 10^{1}>$ $<10^{-7}; 10^{3}>$

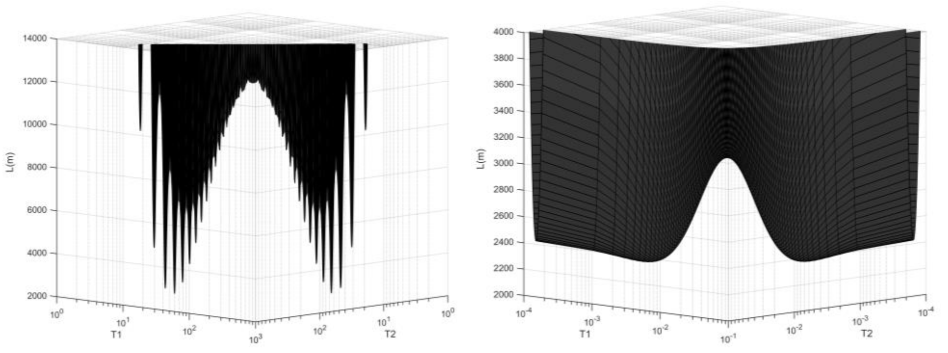

Fig. 2 Model (A) and model (B) state spaces for constant motor gains K

3. Basic Results

Two program environments were chosen for conducting the experiments – the MATLAB Routine and the Open Optimization Routine (OOR). Both tools used controlled random number generators to assure simulation repeatability in the future. Moreover, the random number generator was the same for both routines during all the experiments and was calculated by a particular program tool to suit the PSO method.

The MATLAB Routine was compiled both in MATLAB release R2009a and 2014b because of the updated random number generator syntax. The MATLAB routine follows the PSO algorithm utilizing MATLAB embedded function *"tf"* to create a model transfer function and a function *"lsim"* to evaluate the fitness function.

The Open Optimization Routine (OOR) has been developed for the open optimization process [9, 10]. The inner structure allows the scientists to define the optimization problems independently and then to compare their results. The OOR was compiled using C# language and .NET environment. For the evaluation of the fitness function, the theory of "Differential Equation Numerical Solution" was used. As models (A), (B) and (C) can be transformed into model (D) (and for fitness function evaluation the models were transformed), only the state differential equations for model (D) can be used.

For the purpose of these experiments, an acceptable level of accuracy of the proposed results had to be established. The examples of the PSO results are shown in Fig. 3 on the left. The limit of 5 % accuracy chosen for the fitness function global minimum was based on the step response error range described by automation control theory. The examples of the PSO results falling into the 5 % accuracy limit are then shown in Fig. 3 on the right. Both graphs are related to model (A) when 100 steps of the PSO method were initialized in narrow state space by random uniform distribution in MATLAB (72.3 % of PSO simulation results entered the 5 % accuracy limit).

To validate the experiments, snapshots after 25th, 50th, 75th, 100th, 150th and 200th step of the PSO method were taken. Tab. 3 and Tab. 4 show the percentage of successful results for selected sets of steps performed by OOR. As expected, the more steps in the PSO method, the higher the probability for getting good results. At the number of 100 steps, there is a very high probability for gaining suitable results.

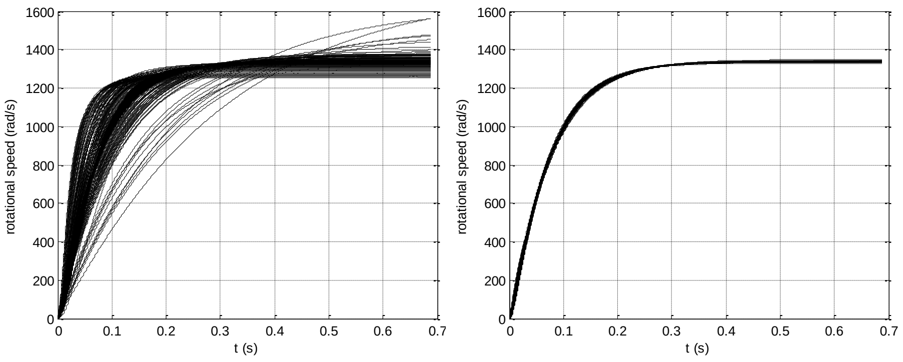

Fig. 3 Examples of PSO results: no limits and 5 % accuracy limit

The PSO method uses a random initial placement of agents within the state space. This placement can be done with different distributions, however, generally recommended random uniform distribution proved to be very successful except model (D) average success for the narrow state space; nonetheless it was completely ineffective for exploring the wide state space.

An innovative approach to set the PSO method initial agents' positions within a wide state space had to be introduced. A random log-uniform distribution using a probability distribution of random variables whose logarithms are uniformly distributed was chosen. Such distribution grants the same probability of success for numerous intervals <10^0; 10^1>, <10^1; 10^2>, <10^2; 10^3>, etc.

Tab. 3 PSO OOR within the 5 % accuracy limit (uniform distribution)

Steps	**Narrow state space**				**Wide state space**			
	A	**B**	**C**	**D**	**A**	**B**	**C**	**D**
25	25.5	18.1	15.5	8.8	0.0	0.0	0.0	0.0
50	58.9	52.3	47.1	30.6	0.1	0.0	0.0	0.0
75	74.0	72.5	64.9	43.4	0.4	0.0	0.0	0.0
100	79.1	77.7	71.9	48.5	0.7	0.0	0.0	0.0
150	82.4	80.6	75.1	53.2	3.8	0.0	0.0	0.0
200	82.9	80.6	75.4	53.3	8.6	0.0	0.0	0.0

Tab. 4 PSO OOR within the 5 % accuracy limit (log-uniform distribution)

Steps	**Narrow state space**				**Wide state space**			
	A	**B**	**C**	**D**	**A**	**B**	**C**	**D**
25	22.2	31.5	34.0	29.6	1.9	2.0	1.3	1.1
50	43.4	61.6	62.4	54.7	10.7	10.3	7.2	6.4
75	52.9	69.1	71.1	64.0	21.3	21.0	16.5	15.2
100	57.5	72.5	73.6	67.2	31.7	29.7	27.0	24.8
150	61.9	74.7	75.3	69.8	46.3	38.8	33.5	34.2
200	62.4	75.0	76.0	69.9	54.0	39.7	34.8	35.9

The use of log-uniform distribution revealed very good success except model (A) average success when exploring the narrow state space and, at the same time, much better success than uniform distribution when exploring the wide state space. We can conclude that the hypothesis about better behaviour of the PSO method within wide state space initialized by random log-uniform distribution is true. Moreover, we can recommend using the random log-uniform distribution instead of the random uniform distribution in all cases where the dimensions of state space are not known.

According to the theory of probability, it is better to run 100 steps of the PSO method twice than to run 200 steps of the PSO method once. Tab. 5 shows the percentage of MATLAB 1000 runs of PSO-100 step simulations falling within the 5 % accuracy limit for model A, evaluated as 1000 single runs, 500 pairs, 333 triplets and 250 quadruplets.

Tab. 5 MATLAB repeated runs probability (100 steps)

Distribution	**Interval**	**One run**	**Two runs**	**Three runs**	**Four runs**
uniform	narrow	72.30	92.20	97.30	99.60
uniform	wide	49.20	75.20	87.09	94.00
log-uniform	narrow	2.00	3.60	5.71	7.20
log-uniform	wide	30.20	50.80	62.76	76.00

Due to the space restriction of the article, only 100 steps of PSO simulations were chosen to compare simulation results. Comparing the two independent experimental tools in Tab. 6 shows almost the same influence of state space topology on the PSO method results when exploring the state space was initialised by uniform distribution. Some distinction can be found between MATLAB and OOR results when exploring the state space was based on log-uniform distribution of initial agents' positions but both methods confirmed our hypothesis about better behaviour of the PSO method within wide state space.

Tab. 6 PSO within the 5 % accuracy limit (basic results)

Distribution	**Interval**	**MATLAB**				**OOR**			
		A	**B**	**C**	**D**	**A**	**B**	**C**	**D**
uniform	narrow	72.3	69.3	64.9	41.0	79.1	77.7	71.9	48.5
uniform	wide	2.0	0.0	0.0	0.0	0.7	0.0	0.0	0.0
log-uniform	narrow	48.3	71.3	66.4	64.6	57.5	72.5	73.6	67.2
log-uniform	wide	30.2	44.0	35.1	35.5	31.7	29.7	27.0	24.8

4. Advanced Results

All the gained results were very interesting and more or less anticipated [12]. The next issue to explore was the influence of the model transformation on PSO simulation success. New sets of simulation were performed with initial estimates of models (B), (C) and (D) calculated from the initial estimates of model (A) according to the transformation equations.

Comparing uniform distribution for the narrow state space, there is a little progress in model (B) and (C) success; though there is a noticeable leap in model (D) success. However, the other three sets of simulations were less successful, compare Tab. 6 and Tab. 7.

Tab. 7 PSO within the 5 % accuracy limit (advanced results)

Distribution	**Interval**	**MATLAB**				**OOR**			
		A	**B**	**C**	**D**	**A**	**B**	**C**	**D**
uniform	narrow	72.3	72.9	79.8	88.0	80.5	75.7	84.5	89.8
uniform	wide	2.0	0.4	0.5	0.3	1.9	0.3	0.3	0.7
log-uniform	narrow	48.3	50.2	54.7	56.2	57.5	53.6	59.9	60.8
log-uniform	wide	30.2	21.1	34.3	23.1	33.9	14.3	22.4	15.6

Having results for all models allowed to evaluate equal success for each model. Evaluation was done by comparing the success of separate models initialised by the same random generator seed. Tab. 8 shows the number of successful results (unlike the percentage of successful results in the previous tables) based on success of exactly one model (A, B, C or D), exactly two models (AB, AC, AD, BC, BD or CD), exactly three models (ABC, ABD, ACD or BCD) and success of exactly all four models (ABCD).

Tab. 8 MATLAB PSO within the 5 % accuracy limit

Distribution	**Interval**	**Original (see Tab. 6)**				**Transformed (see Tab. 7)**			
		1	**2**	**3**	**4**	**1**	**2**	**3**	**4**
uniform	narrow	149	331	320	176	83	127	163	576
uniform	wide	20	0	0	0	22	3	0	1
log-uniform	narrow	123	310	385	152	221	228	175	223
log-uniform	wide	298	214	178	47	234	152	119	48

As mentioned above, the basic experiments were initialized by the same random generator seed but different agent start positions, while the advanced experiments were initialized by the same random generator seed and the same agent start positions transformed into each model state space. Many questions arose and the main question is whether we could get better results, if the different models were combined together.

To an inexperienced user who would like to utilize the particular results presented in this article, model (A) can be recommended for any identification processes using the PSO method. On the other hand, in some special cases, model cooperation based on initial estimates transformation gives the best results.

5. Conclusion

The PSO method is a probabilistic method, thus numerous tests must be conducted in order to get precise results (a suboptimal solution is very close to an optimal solution). Each test described in this article was run and evaluated at least 1000 times and thus the stochastic behaviour of the PSO method should have minimum effect on the presented results.

The use of two different tools (MATLAB and OOR) offers the validation that is often needed. While MATLAB is a universal tool for many applications, the OOR is a very specific application (and library) that enables faster experiment execution. Using the same hardware for running particular experiments, the OOR needs approximately half computation time than the MATLAB.

The hypothesis that the traditional random uniform distribution used for PSO method initial agents' positions fails within the really wide state space was fully confirmed. Experiments based on the proposed random log-uniform distribution showed better success than the uniform distribution when the wide state space was explored. Such basic results may imply general recommendation to use log-uniform initialisation of the PSO method when the state space dimensions are really not known.

Advanced results have shown the influence of the model transformation, more precisely the influence of the transformed initial position estimates on the PSO method. The idea models (C) and (D) would be much less successful than models (A) and (B) because having only one optimum within the state space was not confirmed during basic experiments. However, the transformed initial position estimates based on state space of model (A) helped in some cases other models to get better results.

To conclude, the use of different models resulting in different state spaces has a noticeable impact on the identification based on the PSO method. Different shape of appropriate fitness function prevents particles from getting stuck in local optimum. The trend to stay near local optimum is the main problem of the PSO method and a number of articles are discussing it. Therefore we suggest that a natural combination of different models (fitness functions) can considerably reduce the problem.

Acknowledgement

The work presented in this article has been supported by the Czech Republic Ministry of Defence (K206 Department development program "Complex aviation electronic system for unmanned aerial systems").

References

[1] KENNEDY, J. and EBERHART, R. Particle Swarm Optimization. In *Proceedings of IEEE International Conference on Neural Networks*. New York: IEEE, 1995, p. 1942-1948, ISBN 0-7803-2769-1.

[2] KENNEDY, J. and EBERHART, R. *Swarm Intelligence.* San Francisco: Morgan Kaufmann, 2001, ISBN 1-55860-595-9.

[3] DENG, X. System Identification Based on Particle Swarm Optimization Algorithm. *In Proceedings of 2009 International Conference on Computational Intelligence and Security*. Washington: IEEE, 2009, p. 259-263, 2009, ISBN 978-0-7695-3931-7.

[4] DAI, Y., LIU, L. and SONG J. Complex Nonlinear System Identification Based On Cellular Particle Swarm Optimization. In *Proceedings of 2013 IEEE International Conference on Mechatronics and Automation*. IEEE, p. 1486-1491, 2013, ISBN 978-1-4673-5560-5.

[5] CHEN, S., MEI, T., LUO, M. and YANG, X. Identification of Nonlinear System Based on a New Hybrid Gradient-Based PSO Algorithm. In *Proceedings of 2007*

International Conference on Information Acquisition. IEEE, p. 265-268, 2007, ISBN 978-1-4244-1219-8.

[6] KINCL, Z. and KOLKA, Z. Test Frequency Selection Using Particle Swarm Optimization. In *Advances in Electrical and Electronic Engineering*, 2013, vol. 11, no. 3, p. 507-513, ISSN 1804-3119.

[7] KUO, B. C. *Automatic control systems.* 7th ed. Prentice-Hall, USA, 1995. ISBN 0-13-304759-8.

[8] PHILLIPS, Ch. L. and HARBOR, R. D. *Feedback Control Systems*. Prentice-Hall, USA, 1996, ISBN 0-13-371691-0.

[9] STEFEK, Alexandr. Benchmarking of heuristic optimization methods. In *Proceedings of 14th International Conference on Mechatronics MECHATRONIKA 2011*. Trencin: Alexander Dubcek University of Trencin, 2011, p. 68-71, ISBN 978-808075477-8.

[10] STEFEK, A. Distributed Optimization – Concepts, Ideas and Solutions. In *Croatian Journal of Education, 2012, v*ol. 14, Issue SPECIAL.ISS, p. 161-167, ISSN 1848-5189.

[11] DUB, M. and STEFEK, A. Evaluation of PSO Method Application to DC Machine Experimental Identification. In *Proceedings of the International Conference on Military Technology.* Brno: University of Defence, 2013, p. 887-892, ISBN 978-80-7231-917-6.

[12] DUB, Michal and STEFEK, Alexandr. Using PSO method for System Identification. In *Mechatronics 2013. Recent Technological and Scientific Advances.* New York: Springer, 2013, p. 143-150, ISBN 978-3-319-02293-2.