Comparison of Constraint-handling Techniques Used in Artificial Bee Colony Algorithm for Auto-Tuning of State Feedback Speed Controller for PMSM

Rafal Szczepanski[1],Tomasz Tarczewski[1], Krystian Erwinski[1] and Lech M. Grzesiak[2]

[1]*Department of Automatics and Measurement Systems, Nicolaus Copernicus University, Grudziadzka 5, 87-100 Torun, Poland*

[2]*Institute of Control and Industrial Electronics, Warsaw University of Technology, Koszykowa 75, 00-662 Warsaw, Poland*

Keywords: Constraint-handling Technique, Augmented Lagrangian, Deb's Rules, Optimization, Artificial Bee Colony, State Feedback Speed Controller, PMSM.

Abstract: This article focuses on comparison of two constraint-handling techniques: Deb's Rules (DR) and Augmented Lagrangian (AL) applied to Artificial Bee Colony (ABC) algorithm that is used for auto-tuning of state feedback speed controller (SFC) for permanent magnet synchronous motor (PMSM). The task of the optimization algorithm is to determine the elements of **Q** and **R** weighting matrices in linear quadratic regulator (LQR) optimization process. Chosen matrices guarantee the best performance according to given optimization criteria. Safety and proper operation of the motor requires the use of constraint-handling (C-H) technique. The ABC in its original version cannot handle the constrained optimization problems, therefore necessary modifications of considered optimization algorithm are depicted. Simulation and experimental results showed that AL technique allows to obtain a better convergence of ABC algorithm and a better performance of the PMSM drive than DR technique.

1 INTRODUCTION

Optimization problems are present in all branches of applied sciences and engineering sciences. Most practical applications require limiting of physical variables, which involve equality or inequality constraints. Over the past years, nature-inspired optimization algorithms, such as Artificial Bee Colony (ABC), Particle Swarm Optimization (PSO), Genetic Algorithm (GA), Flower Pollination Algorithm (FPA), Ant-Colony (AC), Grey Wolf Optimizer Algorithm (GWO) and many others, have gained popularity in solving engineering optimization problems (Kaminski and Najdek, 2018), (Wang et al., 2016), (Senberber and Bagis, 2017). Most of nature-inspired algorithms, in their original versions, can solve only unconstrained optimization problems. Therefore researchers had to apply C-H techniques (Gionfra et al., 2017), (Deb, 2000), (Khalilpourazari and Khalilpourazary, 2018), (Long et al., 2017), (Tarczewski and Grzesiak, 2018), (Szczepanski et al., 2017). There are many techniques to handling constraints (Mezura-Montes and Coello, 2011), but the most commonly used groups are: penalty functions and, separation of objective function and constraints. First group allows transformation of constrained optimization problems into unconstrained optimization problems by adding penalty functions to the objective function for each constraint. In order to favor selection of a feasible (e.g. valid) solution, the penalty functions decrease the fitness of an infeasible solution. The opposite idea of C-H technique is based on separation of the objective function and constraints. Keeping both values apart allows for optimization on an unconstrained problem by calculating fitness of solution with different equations for feasible and infeasible solution or tournament selection. Mathematically, the equality constrained optimization problem can be presented as:

$$\begin{aligned} &\text{Minimize} && f(x) \\ &\text{subject to:} && h(x) = 0 \end{aligned} \tag{1}$$

where $f : \mathbb{R}^n \rightarrow \mathbb{R}$, $h : \mathbb{R}^n \rightarrow \mathbb{R}^m$and x_i, $i = 1, ..., n$ is bounded by lower and upper limits $l_i \leq x_i \leq u_i$ which define the search space.

The reliable comparison of C-H techniques requires non-trivial constrained optimization problem. In this paper, an auto-tuning process of state feed-

Szczepanski, R., Tarczewski, T., Erwinski, K. and Grzesiak, L.
Comparison of Constraint-handling Techniques Used in Artificial Bee Colony Algorithm for Auto-Tuning of State Feedback Speed Controller for PMSM.
DOI: 10.5220/0006904002690276
In *Proceedings of the 15th International Conference on Informatics in Control, Automation and Robotics (ICINCO 2018) - Volume 1*, pages 269-276
ISBN: 978-989-758-321-6

back speed controller for PMSM has been chosen. PMSM has wide range of applications (e.g. electrical and hybrid vehicles, CNC machines, ventilating and air conditioning applications (Chan, 1993), (Liu et al., 2016), (Dai et al., 2007), (Lin et al., 2006), (Abrahamsen et al., 2000). Variable speed drives with PMSMs are commonly controlled by cascade of PI controllers or by state feedback controller. The latter control structure is applied in the proposed approach due to superior dynamical properties, with particular emphasis on disturbance compensation (Tarczewski and Grzesiak, 2016). Since all state-space variables of the plant are simultaneously controlled by a single controller, the design process requires selection of all coefficients at the same time. This is a non-trivial task, especially for complex control systems. The trial-and-error method is commonly used to tune SFC regardless of linear-quadratic or pole-placement design technique. In (Franklin et al., 1998), Bryson's method is described for initial guess of diagonal elements of penalty matrices. The pole-placement technique is based on location of poles, which usually requires expert knowledge. A novel usage of nature-inspired optimization algorithm to auto-tuning SFC for PMSM was proposed in (Tarczewski and Grzesiak, 2018), where DR technique has been applied as C-H technique. To the best our knowledge, usage of AL technique for the above described optimization problem and comparison of C-H techniques for auto-tuning of state-space controller were not presented before.

In this paper LQR is used to tune SFC gains. However the coefficients of penalty matrices are obtained by applying ABC algorithm. Two C-H techniques are investigated to analyze the performance of constrained, nature-inspired optimization algorithm.

2 STATE FEEDBACK SPEED CONTROLLER FOR PMSM

The knowledge of state-space description of PMSM fed by voltage-source inverter (VSI) is necessary for synthesis process of state feedback speed controller for PMSM. It was assumed that: (i) all state-space variables of the motor are directly measured using appropriate sensors and calculations and, (ii) a single SFC controls all state-space variables.

2.1 Linearized Model of the PMSM

In order to design SFC for PMSM, state-space representation of the plant (i.e. PMSM fed by VSI) should be introduced. The following assumptions will be adopted to design a linear description of the drive's model:

- a simple feedback linearization procedure will be employed,
- VSI's dynamic behavior and non-linearities will be neglected,
- a PMSM with surface mounted magnets will be considered, and therefore $L_d = L_q = L_s$,
- load torque cannot be measured, and therefore it will be omitted,
- reference signal's internal model will be included.

The aforementioned assumptions lead to the following representation of the considered plant in *d*-*q* reference frame (Tarczewski and Grzesiak, 2018):

$$\frac{d\mathbf{x}_i(t)}{dt} = \mathbf{A}_i\mathbf{x}_i(t) + \mathbf{B}_i\mathbf{u}_i(t) + \mathbf{F}_i r_i(t) \qquad (2)$$

with:

$$\mathbf{A}_i = \begin{bmatrix} -\frac{R_s}{L_s} & 0 & 0 & 0 \\ 0 & -\frac{R_s}{L_s} & 0 & 0 \\ 0 & -\frac{K_t}{J_m} & -\frac{B_m}{J_m} & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}, \quad \mathbf{F}_i = \begin{bmatrix} 0 \\ 0 \\ 0 \\ -1 \end{bmatrix},$$

$$\mathbf{B}_i = \begin{bmatrix} \frac{K_p}{L_s} & 0 \\ 0 & \frac{K_p}{L_s} \\ 0 & 0 \\ 0 & 0 \end{bmatrix}, \quad \mathbf{x}_i(t) = \begin{bmatrix} i_d(t) \\ i_q(t) \\ \omega_m(t) \\ x_\omega(t) \end{bmatrix},$$

$$\mathbf{u}_i(t) = \begin{bmatrix} u_{ld}(t) \\ u_{lq}(t) \end{bmatrix}, \quad r_i(t) = \omega_{mref}(t),$$

where: R_s, L_s – resistance and inductance of the PMSM stator, J_m – moment of inertia, K_t – torque constant, B_m – viscous friction, $i_d(t)$, $i_q(t)$ – current space vector components, $\omega_m(t)$ – angular speed of the PMSM shaft, K_p – gain of VSI, $u_{ld}(t)$, $u_{lq}(t)$ – linear components of control voltages, $\omega_{mref}(t)$ – reference value of angular speed. The last state-space variable has been introduced to ensure steady-state error-free operation for step changes of reference speed and load torque. It is specified by the following formula:

$$x_\omega(t) = \int_0^t [\omega_m(\tau) - \omega_{mref}(\tau)]\, d\tau \qquad (3)$$

Shown in (2), linear components of control voltages are obtained by using feedback linearization procedure described in (Grzesiak and Tarczewski, 2012):

$$u_{ld}(t) = u_{sd}(t) + u_{md}(t) \tag{4}$$

$$u_{lq}(t) = u_{sq}(t) - u_{mq}(t) \tag{5}$$

with:

$$u_{md}(t) = p\omega_m(t)L_s i_q(t)/K_p \tag{6}$$

$$u_{mq}(t) = p\omega_m(t)(L_s i_d(t) + \psi_f)/K_p \tag{7}$$

where: $u_{sd}(t)$, $u_{sq}(t)$ – space vector components of inverter control voltages, $u_{md}(t)$, $u_{mq}(t)$ – non-linear components of control voltages, p – the number of pole pairs, ψ_f – permanent magnet flux linkage.

2.2 State Feedback Controller

A discrete state feedback speed controller obtained for (2) has the following form:

$$\mathbf{u}_i(n) = -\mathbf{K}x_i(n) = -\mathbf{K}_x x(n) - \mathbf{K}_\omega x_\omega(n) \tag{8}$$

with:

$$\mathbf{K} = [\mathbf{K}_x \quad \mathbf{K}_\omega] = \begin{bmatrix} k_{x1} & k_{x2} & k_{x3} & k_{\omega 1} \\ k_{x4} & k_{x5} & k_{x6} & k_{\omega 2} \end{bmatrix} \tag{9}$$

where: n – a discrete sample time index, k_{x1}, k_{x2}, k_{x3}, k_{x4}, k_{x5}, k_{x6}, $k_{\omega 1}$ and $k_{\omega 2}$ – gain coefficients of SFC. Proper selection of gain coefficients is not trivial, because all of them should be simultaneously chosen. These could be determined by using linear-quadratic optimization (Grzesiak and Tarczewski, 2011) or pole placement technique (Grzesiak and Tarczewski, 2012). In the proposed approach the first one is used. The method minimizes discrete performance index for weighting matrices $\mathbf{Q}$ and $\mathbf{R}$:

$$I_{LQR} = \sum_{n=0}^{\infty} \left[\mathbf{x}_i^{\mathrm{T}}(n)\mathbf{Q}\mathbf{x}_i(n) + \mathbf{u}_{li}^{\mathrm{T}}(n)\mathbf{R}\mathbf{u}_{li}(n)\right] \tag{10}$$

with

$$\mathbf{Q} = \mathrm{diag}(\begin{bmatrix} q_1 & q_2 & q_3 & q_4 \end{bmatrix}), \quad \mathbf{R} = \mathrm{diag}(\begin{bmatrix} r_1 & r_2 \end{bmatrix}) \tag{11}$$

where: q_1, q_2, q_3, q_4, r_1 and r_2 are coefficients of penalty matrices. The trial-and-error manual approach to determine those values could be time-consuming and challenging process for the control system considered. In this paper the Artificial Bee Colony algorithm will be employed to obtain $\mathbf{Q}$ and $\mathbf{R}$ values.

3 CONSTRAINED ARTIFICIAL BEE COLONY ALGORITHM

An Artificial Bee Colony optimization algorithm has been proposed by Karaboga in 2005. It is based on the intelligent foraging behavior of honey bee swarm (Karaboga and Basturk, 2007). In the next years Karaboga proved experimentally that ABC has better performance than other popular nature-inspired algorithms (Karaboga and Basturk, 2008).

3.1 Artificial Bee Colony Algorithm

The block diagram of the ABC algorithm is shown in Fig. 1. The algorithm divides the colony into three groups: employed bees, onlooker bees and scouts. Employed bees look for a new food source in the randomly chosen neighbourhood.

Onlooker bees go from their actual food source to another food source depending on the nectar amount in the source. The last group, scouts, only appear when a food source is abandoned and a new one has to be found. An employed bee becomes a scout when the number of failed attempts exceeds the predefined parameter called *limit*. After initialization, the algorithm repeats all aforementioned phases *MCN* times. To reduce the diversity of new food sources produced by employed and onlooker bees, the modification rate *MR* is introduced. The parameter determines probability of change in dimension. Well matched *MR* allows to reduce diversity without convergence reduction. Values of ABC used in the considered auto-tuning problem are listed in Table 1.

Table 1: Artificial Bee Colony parameters.

Parameter & (Symbol)	Value
No of optimized parameters (D)	6
No of colony size (NP)	10
No of food sources (FN)	$NP/2$
Maximum no of cycles (MCN)	50
Control parameter ($limit$)	$FN \times D$
Scout production period (SPP)	$FN \times D$
Modification rate (MR)	0.8
Lower bounds ($lb_1 \div lb_D$)	1×10^{-3}
Upper bounds ($ub_1 \div ub_D$)	1×10^{4}
Weighting coefficient (α)	1×10^{-3}

In order to apply the ABC for automatic selection of $\mathbf{Q}$ and $\mathbf{R}$ values that assure satisfactory behavior of the drive, an optimization performance index should be defined. On the basis of information contained in (Tarczewski and Grzesiak, 2018), the following formula was chosen:

$$f(x) = \sum_{n=0}^{N} \left[e_\omega^2(x,n)nT_s + e_{id}^2(x,n)nT_s + \alpha\Delta u_{sq}^2(x,n)\right] \tag{12}$$

with:

$$e_\omega(x,n) = \omega_m(x,n) - \omega_{mref}(x,n)$$

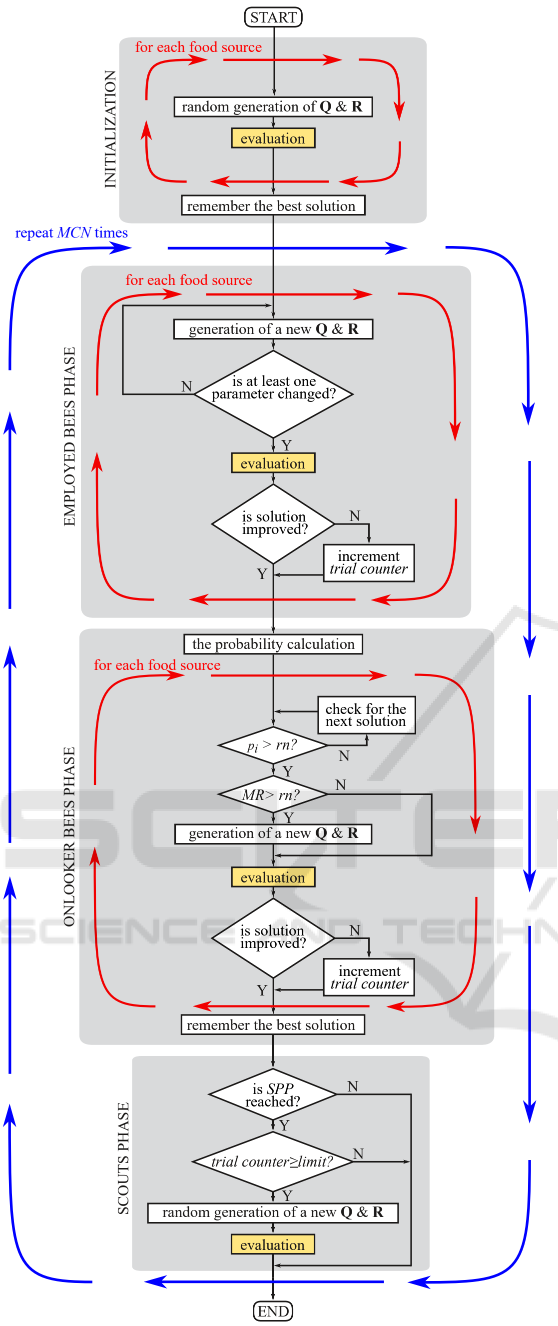

Figure 1: The block diagram of Artificial Bee Colony algorithm.

$$e_{id}(x,n) = i_d(x,n) - i_{dref}(x,n)$$
$$\Delta u_{sq}(x,n) = \left[u_{sq}(x,n) - u_{sq}(x,n-1)\right]/T_s$$

where: α – manually selected coefficient, $i_{dref}(x,n)$ – the reference value of d-axis current. From (12) it can be seen, that the performance index has three components: the first one is responsible for steady-state error-free operation and dynamical properties of the drive's angular velocity, the second one should assure zero d-axis current control strategy, and the last one is responsible for minimization of chattering in q-axis control signal.

It should be noted, that despite of performance index selection, an original version of ABC (i.e. non-constrained) cannot be directly employed for auto-tuning of SFC. Since safe operation of the drive requires limitation of q-axis current and q-axis control signal, a constraint-handling technique should be applied. The constraints are computed from the following formulas:

$$h_{i_q}(x) = \sum_{n=0}^{N} MAX(0, c_{i_q}(x,n))$$
$$h_{u_{sq}}(x) = \sum_{n=0}^{N} MAX(0, c_{u_{sq}}(x,n)) \tag{13}$$

with:

$$c_{i_q}(x,n) = \frac{|i_q(x,n)|}{i_{q\,max}} - 1$$
$$c_{u_{sq}}(x,n) = \frac{|u_{sq}(x,n)|}{u_{sq\,max}} - 1$$

Although C-H method based on Deb's Rules has been recently employed to auto-tuning of SFC (Tarczewski and Grzesiak, 2018), it is worth to examine the impact of other techniques on the convergence of ABC algorithm as well as on the performance of the PMSM drive. For that reason, Powell-Hestenes-Rockafellar Augmented Lagrangian (Powell, 1967), (Hestenes, 1969), (Rockafellar, 1974) will be compared with Deb's Rules (Deb, 2000) C-H technique. Due to this, two variants of the evaluation block will be used. Their contents are shown in Fig. 2.

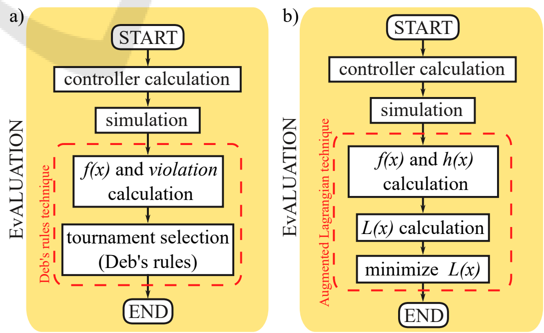

Figure 2: Content of evaluation block.

3.2 Deb's Rules Technique

In 2000 Deb proposed a C-H technique based on separation of objective function and constraints. Nowadays this method is known as Deb's Rules (DR) and it is used with many nature-inspired optimization algorithms (e.g. PSO (Gionfra et al., 2017), ABC

(Tarczewski and Grzesiak, 2018), GA (Deb, 2000)). The DR introduces only one additional variable called *violation*, which is defined as:

$$violation = \|h(x)\|_\infty \tag{14}$$

$$solution \rightarrow \begin{cases} feasible & \text{if} \quad violation \leq 0 \\ infeasible & \text{if} \quad violation > 0 \end{cases}$$

In order to compare two solutions, the following rules are used:

- for feasible solution and infeasible solution, the feasible one is selected,
- for two feasible solutions, the one having better objective function value is selected
- for two infeasible solutions, the one having smaller *violation* parameter value is selected

Above rules guarantee that, the solution will be feasible or the value of *violation* will be minimized. The main advantages of DR are: (i) the lack of parameters that have to be chosen individually for the problem, and (ii) a simple implementation.

3.3 Augmented Lagrangian Technique

Augmented Lagrangian (AL) technique works by using additional formulas, called penalty functions, to the objective function for each violated constraint. The augmented objective function (Augmented Lagrangian) has the following form (Birgin and Martínez, 2008):

$$L(x,\lambda,\rho) = f(x) + \frac{\rho}{2}\sum_{i=1}^{m}\left[h_i(x) + \frac{\lambda_i}{\rho}\right]^2 \tag{15}$$

where $\lambda : \mathbb{R}^m$ and $\rho > 0$. The algorithm updates augmented lagrangian multipliers λ and penalty parameter ρ automatically during optimization process by using following formulas:

$$\rho^{(k+1)} = \gamma\rho^k \tag{16}$$

$$\lambda_i^{(k+1)} = \text{MAX}\left[\lambda_{min}, \text{MIN}\left(\lambda_{max}, \lambda_i^k + \rho^{(k+1)}h_i(x)\right)\right] \tag{17}$$

where: $\lambda_{min} = -10^{20}$, $\lambda_{max} = 10^{20}$, $\rho_{min} = 10^{-6}$, $\rho_{max} = 10$ and $\gamma = 10$. It is worth to point out that aforementioned values are directly taken from (Andreani et al., 2007).

Initially AL multipliers are equal to zeros and the penalty parameter is defined as:

$$\rho^{initial} = \text{MAX}\left[\rho_{min}, \text{MIN}\left(\rho_{max}, \frac{2\mid f(x_0)\mid}{\|h(x_0)\|^2}\right)\right] \tag{18}$$

It was assumed that for $\|h(x_0)\|$ equal to zero, the penalty parameter takes the minimum value. To avoid unnecessary modifications of the penalty parameter ρ the Infeasibility - Complementarity Measure (*ICM*) parameter is introduced using the following formula:

$$ICM = \|ABS(MAX(h_i(x), -\frac{\lambda_i}{\rho})) \quad i = 1...m\|_\infty \tag{19}$$

The final updating formula of the penalty parameter is defined as:

$$\rho^{(k+1)} = \begin{cases} \alpha\rho^k & \text{if } ICM^{(k+1)} > \frac{ICM^k}{2} \\ \rho^k & \text{otherwise} \end{cases} \tag{20}$$

Finally, the last parameter needed for integration of the AL technique with ABC algorithm is an updating frequency of λ and ρ. Considered value was determined by using trial-and-error method. If parameters will be updated too rarely it could cause premature convergence of the ABC algorithm in infeasible local minimum which may result in rejection of every infeasible solution too early. In the second case, the operation of AL technique would be similar to DR. The experimentally selected parameter λ is updated every two iterations of ABC algorithm.

4 NUMERICAL RESULTS

The algorithms were examined on a computer with Intel i5-7500 @ 3.4GHz CPU with 16GB memory in MATLAB/Simulink environment. The main parameters of the PMSM drive employed during auto-tuning process are listed in Table 2. The laboratory stand consists of two PMSM drives (Tarczewski and Grzesiak, 2018). The main drive is used for evaluation of SFC algorithms, while the second one is employed to produce load torque.

Table 2: The main parameters of the PMSM drive.

Parameter (Symbol)	Value [Unit]
Rated power (P_N)	628 [W]
Rated current (I_N)	3 [A]
Rated torque (T_{eN})	1.05 [Nm]
Rated speed (Ω_{mN})	366 [rad/s]
Resistance (R_s)	0.85 [Ω]
Inductance (L_s)	4 [mH]
Torque constant (K_t)	0.35 [Nm/A]
No of pole pairs (p)	3
Viscous friction (B_m)	2.2×10^{-3} [Nms/rad]
Moment of inertia (J_m)	2×10^{-4} [kgm^2]
VSI gain (K_p)	95
Switching frequency	16 [kHz]

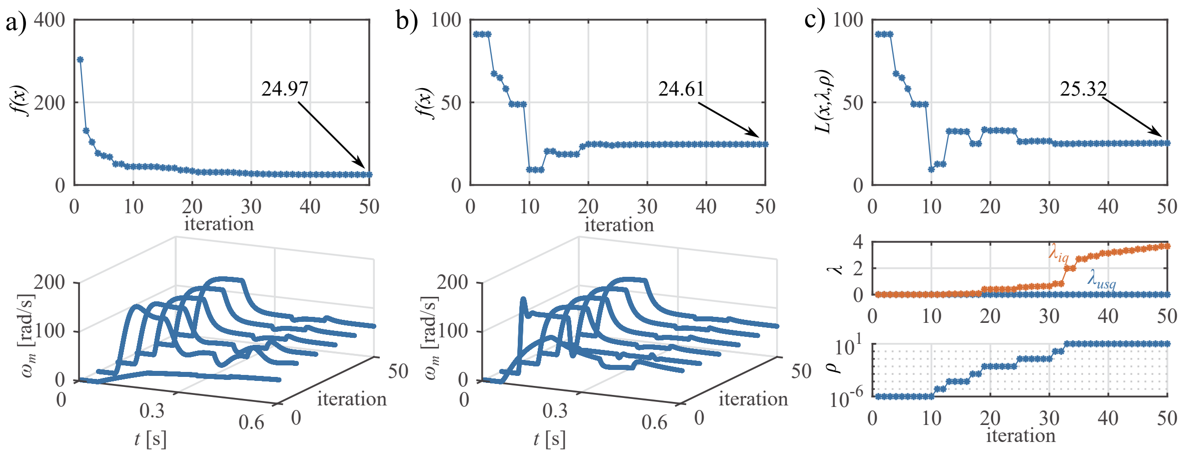

Figure 3: Progress of: a) the objective function and plant response for DR technique; a) the objective function and plant response for AL technique, c) the AL function, AL multiplier and penalty parameter.

The number of objective function's evaluation was the same for both C-H techniques, which results in very similar computation times, c.a. 14 minutes. Repetition of gained result is shown in Fig. 4. It can be seen, that application of AL as C-H technique gives a better repetition and assures a smaller mean value of the objective function in comparison to DR.

The best determined coefficients of **Q** and **R** matrices for both C-H techniques are summarized in Table 3, while coefficients of SFC are listed in Table 4, respectively.

Table 3: Coefficients of: **Q** and **R** matrices.

	DR	AL
q_1	1.25×10^3	5.49×10^3
q_2	1.29×10^2	50.2
q_3	4.3	5.0
q_4	9.38×10^3	9.2×10^3
r_1	7.01×10^3	4.23×10^3
r_2	2.92×10^2	1.51×10^2

Table 4: Coefficients of SFC.

	DR	AL
k_{x_1}	0.3097	0.5749
k_{x_2}, k_{x_3}	0	0
k_{ω_1}, k_{x_4}	0	0
k_{x_5}	0.4436	0.4174
k_{x_6}	0.0839	0.1270
k_{ω_2}	3.3655	5.2356

The progress of the objective functions, AL function, AL parameters and the response of the plant are shown in Fig 3. It is worth to point out that the DR technique never goes under constrained global minimum value of the objective function, which is caused by the tournament selection, and AL technique

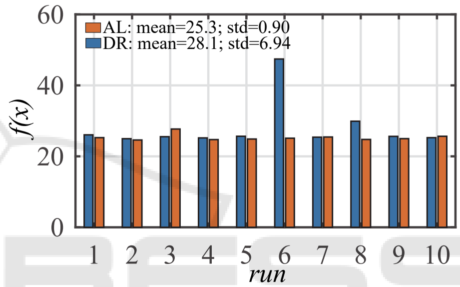

Figure 4: Final objective function values after 10 runs for ABC algorithm with DR and AL techniques.

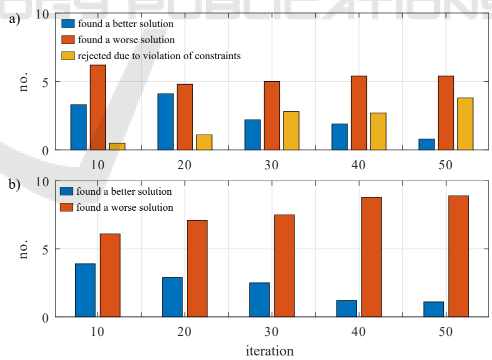

Figure 5: Decisions taken by ABC algorithm during comparison of solutions for: a) DR technique, b) AL technique.

allows to reach infeasible solutions and then the algorithm imposes penalties for violated constraints. The DR technique approaches the solution only from the side of feasible solutions while the AL technique allows for approaching the solution from both sides. The slope from side of feasible solutions is caused by minimizing the objective function and the slope from side of infeasible solutions is caused by minimizing penalty functions (minimizing violation of constraints). In late iterations, when all food sour-

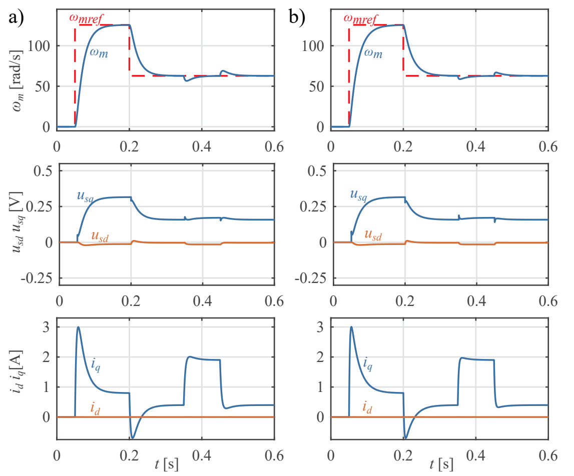

Figure 6: Simulation responses of the PMSM with SFC coefficients found by using ABC algorithm with: a) DR technique, b) AL technique.

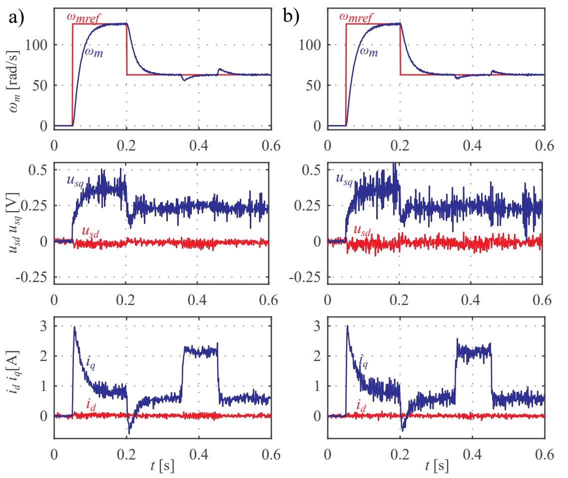

Figure 7: Experimental responses of the PMSM with SFC coefficients found by using ABC algorithm with: a) DR technique, b) AL technique.

ces of ABC algorithm are near the global minimum, the most of new food sources are rejected by DR, because these violate constraints. Taken decisions about new food source for both C-H techniques are shown in Fig. 5.

The comparison of simulation results obtained for PMSM with SFC tuned by ABC with DR and AL C-H techniques is shown in Fig. 6. Result of experiment carried out with the same reference signals (i.e. values of angular speed and load torque) on physical drive is presented in Fig. 7. From simulation and experimental responses it can be seen, that all control objectives are fulfilled. Angular speed is controlled without steady-state error and load torque imposed on the PMSM shaft for $t \in (0.35; 0.45)$ s is properly compensated. Recorded waveforms of current space vector components clearly illustrate, that zero d-axis control strategy is successfully employed and both C-H techniques imposed on q-axis current work well (its maximum value does not exceed rated one).

5 CONCLUSION

This paper presented comparison of popular constraint-handling techniques: Deb's rules and Augmented Lagrangian used with novel, nature-based Artificial Bee Colony algorithm to solve practical engineering problem, which is auto-tuning of SFC for PMSM. Both C-H techniques have been successfully used. These allow to satisfy imposed constraints and to return a feasible solution. DR technique owes its popularity to lack of parameters and easy implementation. AL technique requires selection of several parameters for proper optimization, but in this paper the recommended default parameters have been used. In AL technique a frequency of parameter updating should also be chosen, what was done by using trial-and-error approach. Stability and performance of obtained solution compensates additional parameters that need to be chosen individually for problem. On the basis of simulation and experimental results, it was found that AL technique allows to find a better solution and also to reduce standard deviation between runs.

Regardless of employed C-H technique, obtained weighting matrices assure steady-state error-free operation of the drive and satisfactory dynamical behavior. To the best our knowledge, Augmented Lagrangian technique for auto-tunning of SFC was not presented before. On the basis of obtained results, we recommend to use AL C-H technique in ABC algorithm for the discussed problem.

REFERENCES

Abrahamsen, F., Blaabjerg, F., and Pedersen, J. K. (2000). Efficiency improvement of variable speed electrical drives for hvac applications. In *Energy Efficiency Improvements in Electronic Motors and Drives*, pages 130–135. Springer.

Andreani, R., Birgin, E. G., Martínez, J. M., and Schuverdt, M. L. (2007). On augmented Lagrangian methods with general lower-level constraints. *SIAM J. Optim.*, 18(4):1286–1309.

Birgin, E. G. and Martínez, J. M. (2008). Improving ultimate convergence of an augmented Lagrangian method. *Optim. Method Softw.*, 23(2):177–195.

Chan, C. C. (1993). An overview of electric vehicle technology. *Proc. of the IEEE*, 81(9):1202–1213.

Dai, Y., Song, L., and Cui, S. (2007). Development of pmsm drives for hybrid electric car applications. *IEEE Trans. Magn.*, 43(1):434–437.

Deb, K. (2000). An efficient constraint handling method for genetic algorithms. *Comput. Meth. Appl. Mech. Eng.*, 186(2-4):311–338.

Franklin, G. F., Powell, J. D., and Workman, M. L. (1998). *Digital control of dynamic systems*. Addison-Wesley Menlo Park, CA.

Gionfra, N., Sandou, G., Siguerdidjane, H., Loevenbruck, P., and Faille, D. (2017). A novel distributed particle swarm optimization algorithm for the optimal power flow problem. In *IEEE CCTA Conf.*, pages 656–661.

Grzesiak, L. M. and Tarczewski, T. (2011). Permanent magnet synchronous motor discrete linear quadratic speed controller. In *2011 IEEE ISIE Symp.*, pages 667–672.

Grzesiak, L. M. and Tarczewski, T. (2012). PMSM servodrive control system with a state feedback and a load torque feedforward compensation. *COMPEL*, 32(1):364–382.

Hestenes, M. R. (1969). Multiplier and gradient methods. *J. Optim. Theory Appl.*, 4(5):303–320.

Kaminski, M. and Najdek, K. (2018). Adaptive neural controller based on RBF model applied for electrical drive with PMSM motor. *Przeglad Elektrotechniczny*, pages 94–98 (in Polish).

Karaboga, D. and Basturk, B. (2007). A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm. *J. Glob. Optim.*, 39(3):459–471.

Karaboga, D. and Basturk, B. (2008). On the performance of artificial bee colony (ABC) algorithm. *Appl. Soft. Comput.*, 8(1):687–697.

Khalilpourazari, S. and Khalilpourazary, S. (2018). Optimization of production time in the multi-pass milling process via a robust grey wolf optimizer. *Neural Computing and Applications*, 29(12):1321–1336.

Lin, F.-J., Shieh, H.-J., Shieh, P.-H., and Shen, P.-H. (2006). An adaptive recurrent-neural-network motion controller for X-Y table in CNC Machine. *IEEE Trans. Syst. Man Cybern. Part B-Cybern.*, 36(2):286–299.

Liu, X., Chen, H., Zhao, J., and Belahcen, A. (2016). Research on the performances and parameters of interior pmsm used for electric vehicles. *IEEE Transactions on Industrial Electronics*, 63(6):3533–3545.

Long, W., Liang, X., Cai, S., Jiao, J., and Zhang, W. (2017). An improved artificial bee colony with modified augmented Lagrangian for constrained optimization. *Soft Comput.*

Mezura-Montes, E. and Coello, C. A. C. (2011). Constraint-handling in nature-inspired numerical optimization: past, present and future. *Swarm Evol. Comput.*, 1(4):173–194.

Powell, M. (1967). *"A method for non-linear constraints in minimization problems"*. Atomic Energy Res. Estab. Theoretical Physics Div. ; AERE TP 310. U.K.A.E.A.

Rockafellar, R. T. (1974). Augmented lagrange multiplier functions and duality in nonconvex programming. *SIAM J. Control*, 12(2):268–285.

Senberber, H. and Bagis, A. (2017). Fractional pid controller design for fractional order systems using abc algorithm. In *Electronics, 2017*, pages 1–7. IEEE.

Szczepanski, R., Erwinski, K., and Paprocki, M. (2017). Accelerating PSO based feedrate optimization for NURBS toolpaths using parallel computation with OpenMP. In *22nd Int. MMAR Conf.*, pages 431–436.

Tarczewski, T. and Grzesiak, L. M. (2016). Constrained state feedback speed control of PMSM based on model predictive approach. *IEEE Trans. Ind. Electron.*, 63(6):3867–3875.

Tarczewski, T. and Grzesiak, L. M. (2018). An application of novel nature-inspired optimization algorithms to auto-tuning state feedback speed controller for PMSM. *IEEE Trans. Ind. Appl.*, 54(3):2913–2925.

Wang, X., Ufnalski, B., and Grzesiak, L. M. (2016). Adaptive speed control in the PMSM drive for a nonstationary repetitive process using particle swarms. In *Compatibility, Power Electronics and Power Engineering (CPE-POWERENG), 2016 10th International Conference on*, pages 464–471. IEEE.