Chapter 9
Advanced Control and Optimization Techniques in AC Drives and DC/AC Sine Wave Voltage Inverters: Selected Problems

Barłomiej Ufnalski, Lech M. Grzesiak and Arkadiusz Kaszewski

Abstract This chapter presents the application of a particle swarm optimization (PSO) to a controller tuning in selected power electronic and drive systems. The chapter starts with a relatively simple tuning of a cascaded PI speed and position control system for a BLDC servo drive. This example serves as the background for a discussion on selecting the objective function for the PSO. Then the PSO is used in two challenging controller tuning tasks. This includes optimizing selected learning parameters in the adaptive artificial neural network (ANN) based online trained speed controller for an urban vehicle (3D problem) and selecting penalty factors in the LQR with augmented state (i.e. with oscillatory terms) for a three-phase four-leg sine wave inverter (15D problem). It is demonstrated with the help of these case studies why and where the PSO, or any other similar population based stochastic search algorithm, can be beneficial. Engineers encounter many non-straightforward controller tuning problems in power electronic systems and this chapter illustrates that in some cases it is relatively easy to reduce these tasks into the objective function selection problem. The relevant controller parameters are then determined automatically by the PSO.

B. Ufnalski (✉) · L. M. Grzesiak · A. Kaszewski
Institute of Control and Industrial Electronics, Warsaw University of Technology,
75 Koszykowa Str., Warsaw 00-662, Poland
e-mail: bartlomiej.ufnalski@ee.pw.edu.pl
URL: http://www.isep.pw.edu.pl

L. M. Grzesiak
e-mail: lech.grzesiak@ee.pw.edu.pl

A. Kaszewski
e-mail: arkadiusz.kaszewski@ee.pw.edu.pl

T. Orłowska-Kowalska et al. (eds.), *Advanced and Intelligent Control in Power Electronics and Drives*, Studies in Computational Intelligence 531,
DOI: 10.1007/978-3-319-03401-0_9, © Springer International Publishing Switzerland 2014

1 Particle Swarm Optimization

Particle swarm optimization is attributed to Kennedy, Eberhart and Shi [1]. It is an iterative gradient-free search algorithm inspired by biological swarming such as bird flocking, fish schooling, herding of land animals or collective behavior of insects. The search starts with a random set (here called swarm) of solutions (here called particles). The particles travel through a search space and are rated according to a user defined objective function. Their movements are a function of the individual experience and information acquired from other particles. However, the velocity vectors are not deterministic. The instantaneous strength of social and individual behavior varies randomly for each particle in each iteration. In a basic version of the algorithm the only piece of information shared among the particles is the global best solution found so far. Each particle stores also its best solution found so far. The velocity and position update rules are as follows

$$\begin{aligned} \boldsymbol{v}_j(i+1) = c_1 \boldsymbol{v}_j(i) + c_2\, rand()\left(\boldsymbol{p}_j^{pbest} - \boldsymbol{p}_j(i)\right) \\ + c_3 rand()\left(\boldsymbol{p}_j^{gbest} - \boldsymbol{p}_j(i)\right) \end{aligned} \tag{1}$$

$$\boldsymbol{p}_j(i+1) = \boldsymbol{p}_j(i) + \boldsymbol{v}_j(i+1), \tag{2}$$

where: j is the particle identification number, i denotes the iteration number, $\boldsymbol{v}_j$ and $\boldsymbol{p}_j$ are speed and position of the j-th particle, $\boldsymbol{p}_j^{pbest}$ stores the best solution proposed so far by the j-th particle ($pbest$), $\boldsymbol{p}^{gbest}$ denotes the best solution found so far by the swarm ($gbest$), c_1, c_2 and c_3 are the explorative factor (inertia weight), the individuality factor and the social factor, respectively. It is to note that the speed $\boldsymbol{v}_j(i+1)$ in the rule (2) should be multiplied by time to represent physical velocity. However, it is common practice to set the time increment to 1 and thereafter neglect it in (2). An introduction of a different time increment does not influence a behavior of the swarm because coefficients present in (1) have to be divided by this increment. The search path is not deterministic because of the last two terms in (1) that include multiplication by the random numbers `rand()` generated for each particle in each iteration. The random numbers are uniformly distributed in the unit interval. In all experiments described in this chapter, the c_1, c_2 and c_3 factors have been calculated using the constricted PSO formula [1] and are 0.73, $0.73 \cdot 2.05$ and $0.73 \cdot 2.05$, respectively. If the basic velocity update rule does not manifest satisfactory search abilities, numerous refinements are available. A fairly representative survey can be found in [2, 3]. Taking into account the no free lunch theorem for optimization [4] one can conclude that there is no ultimate version of the PSO. The set of modifications should be selected individually, usually by trials and errors, to suit a given optimization problem. These modifications range from simple velocity clamping to substantial adjustments of particle communication principles like, e.g., in the fully-informed PSO [5]. Additionally, there are certain modifications dedicated to online optimization in non-stationary environments [6]. The rules of swarm movements can be rearranged to handle multi-objective [7] and/or multi-modal [8] problems. Various neighborhood topologies are proposed to serve these purposes [9]. Our main goal is to keep the swarm as simple as possible and still effective in the controller tuning tasks encountered in power electronic or drive systems. This chapter deals with offline optimization in such systems, hence the environment is assumed to be stationary even if a measurement noise is included in the model. Moreover, all optimizations are performed on numerical models of the plants, therefore the search space does not need to be bounded to specific territory resulting from safety requirements for the given plant. The numerical model should include all nonlinearities, noises and other parasitic effects crucial from control point of view. No information on gradient is harnessed by the PSO. This gives a lot of freedom to the designer at problem formulation stage—the mathematical model of the system being optimized can include, e.g., discontinuities. From the same reason, the objective function chosen by the designer can be of any type and one can focus on formulating this function to be aligned with the desired behavior of the system without bothering about its mathematical properties in terms of complexity and differentiability.

In the case of all discussed problems no special measures have been taken regarding swarm movements. The very basic formula (1) tends to produce implementable solutions. All problems have been handled as single-modal and no explicit multi-objectivity has been introduced. It is common among control practitioners to reduce problems with contradictory objectives to single-objective ones with the help of weighted terms in the objective function. This methodology has been employed also here. The simplest communication topology has been used, i.e. the *gbest* attracts all particles and no neighborhood operator is present. However, it has been decided to introduce absorbing walls [10] if clear physical or theoretical constraints are identifiable. This means that the speed of a particle is reset to zero if known boundaries are crossed. No other modifications to the standard PSO have been identified as necessary to effectively tune the below discussed controllers.

2 Objective Function

The objective function (also called cost function, energy function, performance index or fitness function) determines the behavior of the optimal controller. There are some commonly used performance measures as the integral of squared error (ISE)

$$J_{\mathrm{ISE}} = \int_{0}^{\infty} e^{2}(t)\,\mathrm{d}t, \qquad (3)$$

where $e(t)$ denotes control error, or the generalized ISE

$$J_{\mathrm{IGSE}} = \int\limits_{0}^{\infty} \left(e^2(t) + \alpha \dot{e}^2(t)\right) \mathrm{d}t, \tag{4}$$

where α is a subjective weighting factor, or the integral of squared error and derivative of control effort

$$J_{\mathrm{ISEDCE}} = \int\limits_{0}^{\infty} \left(e^2(t) + \beta \dot{u}^2(t)\right) \mathrm{d}t, \tag{5}$$

where $u(t)$ denotes control effort (control signal) and β is again a subjective weighting factor. Especially (3) earns its popularity by producing problems that are easy to approach analytically for some classes of control systems ([11] might serve as the example). However, in a gradient free optimization the performance index can be chosen freely to reflect in the best possible way the desired behavior of the system. It has been decided that for all studied cases the performance indices will be positive definite and the optimization problem will be of the minimization type. The philosophy of signals contributing to the performance index its mean squared value has been kept in all discussed tuning procedures. Nevertheless, to promote sometimes contradictory behaviors like fast transients and no chattering at steady state, additional functions are introduced into the performance index definition allowing for selective contribution to the overall value. For example, the dynamics of the control effort is not penalized during the specific time interval after the change in the reference signal whereas this penalty is non-zero at the steady state. These time windows have to be carefully chosen for each term in the performance index according to physical limitations of the plant. Also, it is common practice to add terms that take into account overshooting or crossing acceptable levels for control signal. The nature of the PSO enables the designer to work with any form of performance index. However, our main goal is to keep this stage simple without sacrificing the performance of the resulting system. All proposed here performance measures have the form of

$$\begin{aligned} J_c = \frac{T_s}{t_{\mathrm{stop}}} \sum_{k=1}^{\frac{t_{\mathrm{stop}}}{T_s}} & (f_1(k)\boldsymbol{e}_y^T(k)\boldsymbol{e}_y(k) + f_2(k)\Delta\boldsymbol{u}^T(k)\Delta\boldsymbol{u}(k) \\ & + f_3(k)\boldsymbol{u}_{\mathrm{aux}}^T(k)\boldsymbol{u}_{\mathrm{aux}}(k) + f_4(k)\boldsymbol{y}_{\mathrm{aux}}^T(k)\boldsymbol{y}_{\mathrm{aux}}(k)), \end{aligned} \tag{6}$$

where $\boldsymbol{e}_{\mathrm{y}}$, $\boldsymbol{u}$, $\boldsymbol{u}_{\mathrm{aux}}$ and $\boldsymbol{y}_{\mathrm{aux}}$ are vectors (in the MIMO case, e.g. as in three-phase converters) containing control errors, control signals, auxiliary signals from the controller and auxiliary signals from the plant, respectively (as depicted in Fig. 1). The discrete representation of the performance measure has been chosen to correspond to the assumed digital implementation of a control system. All signals are sampled at the rate of T_s. The t_{stop} denotes the assessment test time for the particle

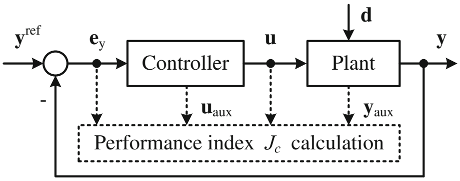


Fig. 1 Selected signals for performance calculation

(equal for all particles). It is to note that the infinite integration limit commonly used in the analytical approach has to be changed to finite test time if the performance evaluation is done by using signal samples recorded during a numerical simulation or a physical experiment. The functions f_1, f_2, f_3 and f_4 are bivalent functions with zero value for intervals with no penalty for a given behavior and positive value, usually different for each function with one of them set to 1, for intervals with a penalty for this behavior. These intervals are correlated with the reference test signal(s) and the test disturbance(s). In some designs they depend on states of a system when, e.g., additional penalty for overshooting is needed. It should be stressed that the bivalency is assumed here to make the design process more intuitive. From now onward the bivalent functions will be referred to as switching functions. The resulting system is optimal for a given shape of the reference and disturbance signals. That is why it is crucial to design the test scenario that includes representative set of anticipated system states. The scaling by the reciprocal of the number of samples present in (6) does not influence the optimization process and is introduced solely to make the value of the performance index easier in interpretation as the mean value of the sum of squares. The test reference signal(s) should take into account physical limits of a plant, e.g. the available acceleration. Otherwise, a dominant contribution to the cost function value coming from demanded behavior outside the physical limits makes determining upper values for switching functions significantly more difficult. It is common practice to implement ramps and s-ramps as reference models for speed or position in electric drive systems. This limits first and second derivative of the reference signal, respectively. It is also practical to use first and second order lag elements if these derivatives are expected to be limited. An example is shown in Figs. 2 and 3. The step reference signal should be avoided in such performance index based assessment tests because this does not reflect most real-life applications. For example, the s-ramp speed reference model is frequently used in drivetrain systems to limit the jerk which is important for a lifetime of a mechanical part of the system and for a comfort of its users, e.g. passengers of a vehicle.

It is common that in the early stage of the search many particles cannot be rated using (6) because the simulation stops before reaching the assumed t_{stop} due to numerical problems. The simulation is also stopped intentionally before t_{stop} if states of a plant reach unacceptable levels from the physical implementation point of view. For some search problems the particles do not carry directly values of parameters of the model but those values are calculated using the values stored in

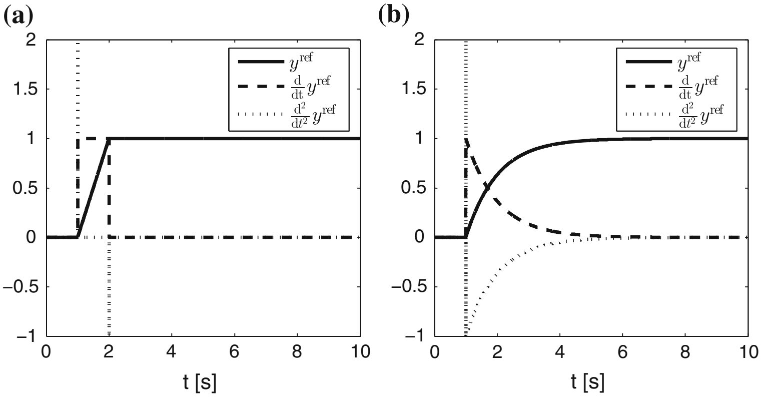


Fig. 2 Reference signal shaped by using rate limiter (**a**) and first order lag element $G(s) = \frac{1}{s\tau+1}$ with $\tau = 1$ s fed by the step signal (**b**)

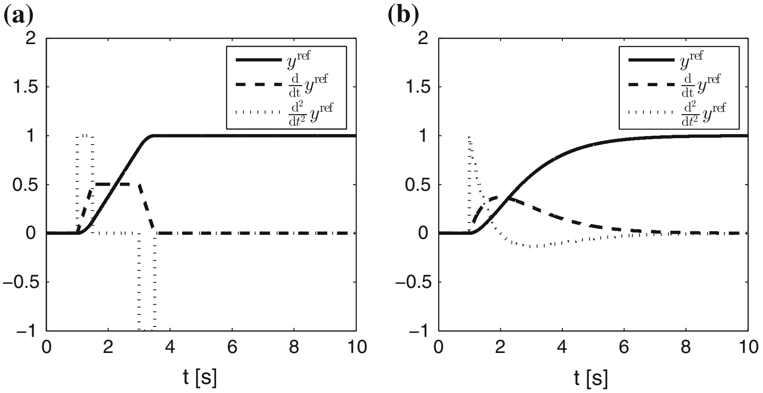
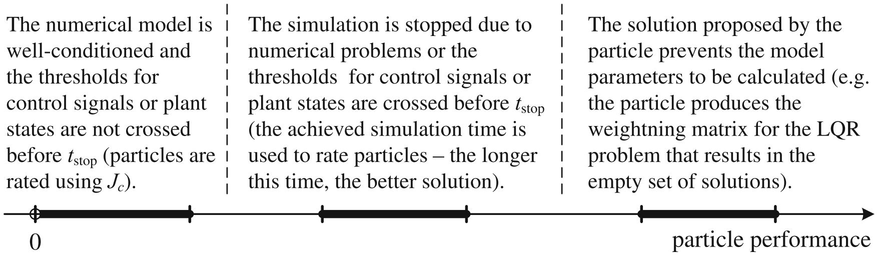


Fig. 3 Reference signal shaped directly as a quadratic spline (**a**) and generated using second order lag element $G(s) = \frac{1}{(s\tau_1+1)(s\tau_2+1)}$ with $\tau_1 = \tau_2 = 1$ s fed by the step signal (**b**)

the particle (see Sect. 6). It is possible then that the identification of poor solutions may takes place even before running the numerical model of the system. It has been tested that rating all such particles as equally poor may impair swarm's capability to keep a good balance between exploration and exploitation. Any functions can be used to rate particles in the event that (6) is not applicable as far as they preserve logical order and their codomains do not overlap. The idea is illustrated in Fig. 4.

The PSO itself puts a minor computational burden on the optimization procedure. The time needed to complete the optimization is dictated by the wall clock time required by the numerical model of the system to be simulated. Non surprisingly, the number of needed simulations depends highly on the form of the objective function. Some guidelines on choosing a good number of particles and



The numerical model is well-conditioned and the thresholds for control signals or plant states are not crossed before t_{stop} (particles are rated using J_c).

The simulation is stopped due to numerical problems or the thresholds for control signals or plant states are crossed before t_{stop} (the achieved simulation time is used to rate particles – the longer this time, the better solution).

The solution proposed by the particle prevents the model parameters to be calculated (e.g. the particle produces the weightning matrix for the LQR problem that results in the empty set of solutions).

Fig. 4 The methodology of distinguishing particles' performance outside the area covered by the definition of J_c

iterations are given in [12]. It has been verified that 50 particles and 100 swarm iterations would suffice in most of the discussed here problems. However, this implies up to 5,000 runs of the model. It could be problematic if the performance of, e.g., a drive with a pulse width modulated (PWM) converter is to be assessed. Usually, a simulation step size two orders of magnitude lower than the controller sampling time is required to obtain trustworthy numerical results. In order to tackle such problems in a reasonable wall clock time, one will need processing capacity extending far beyond the one offered by today's personal computers. On the other hand, many controller tuning tasks encountered in power electronics and drives deal with plant natural frequencies significantly lower than the used PWM frequencies. In such cases it is reasonable to neglect discontinuities introduced by the modulator and simplify the converter to a linear amplifier with a delay. Obtained model usually produces reliable numerical results for a simulation step size equal to the controller sample time. This has helped to reduce computational complexity of the below presented experiments to levels resulting in several-hour-long tuning procedures. The number of swarm iterations is always a subjective choice. Even though there are various indices elaborated for assessing a search progress, none of them is free from subjective choice of thresholds. A fairly representative set of stopping criteria has been described in [13]. The thresholds are usually determined using the guess and check method. For all following optimization problems the number of swarm iterations has been set arbitrary to meet assumed wall-clock time constraints. However, it is advisable to monitor swarm diversity variations which can be helpful in detection of ill-posed problems. A well-established measure of diversity incorporates Euclidean distance of particles to the mean and is defined, see e.g. [14], as follows

$$D_{\text{dist}} = \frac{1}{N_p\sqrt{N_d}} \sum_{j=1}^{N_p} \sqrt{\sum_{n=1}^{N_d} \left(p_{jn} - \overline{p}_n\right)^2}, \tag{7}$$

where N_p is the swarm size, N_d is the dimensionality of the problem and $\overline{p}$ is the average point. Originally this diversity measure uses the length of the longest diagonal in the search space instead of $\sqrt{N_d}$. However, the original definition cannot be applied to swarms with at least one unbounded search direction which is the case in all discussed here systems. If more insight into separate dimensions is needed, a slightly different diversity measure could be used. It contains standard deviations of proposed solutions that can also be merged into a single formula by calculating their mean value per dimension

$$D_{\text{std}} = \frac{1}{N_d}\sum_{n=1}^{N_d}\sqrt{\frac{1}{N_p}\sum_{j=1}^{N_p}\left(p_{jn}-\overline{p}_n\right)^2}. \tag{8}$$

By monitoring the evolution of inner sums in (8) one can detect dimensions with poor convergence. If the swarm does not calm down in selected dimensions, it could suggest that the problem is ill-posed for these dimensions. Such a lack of convergence in selected dimensions can also be intensified by measurement and system noises. Possible solutions are rethinking parameters to be optimized and/or redefining the fitness function.

3 Simultaneous Tuning of Cascaded PI Position and Speed Controllers

There exist numerous analytical and experimental methods for effective PID tuning in a cascaded position, speed and torque control system frequently used in servo drives. Just to mention some of them: modulus and symmetrical optimum methods (Kessler's criteria) [15], Ziegler-Nichols method [16] with different tuning charts, e.g. Pessen recipe, Seborg et al. recipes (some-overshoot rule, no-overshoot rule), Tyreus-Luyben tuning chart. In most cases the resulting control quality is sufficient and there is no need for more elaborated tuning procedures. The example of PSO for a BLDC servo drive serves here only illustrative purposes. However, some easily identifiable advantages of the evolutionary gradientless optimization are present in contrast to the abovementioned methods: the optimizer can work with any user-defined performance index and simultaneous tuning of more than one out of the cascaded controllers is possible. These principles have already been used in [17] for optimizing a cascaded PI control structure with respect to the H_∞ norm. Moreover, the process and the control system can be modeled in any drag-and-drop environment and can include all crucial nonlinearities, i.e. controller saturation and anti-windup, and parasitic effects as the measurement noise.

The dynamics of a three-phase brushless DC machine can be numerically modeled using following mathematical description

$$\begin{cases} u_a(t) = L\frac{\mathrm{d}i_a(t)}{\mathrm{d}t} + Ri_a(t) + e_a(\alpha_e(t), \omega_m(t)) \\ u_b(t) = L\frac{\mathrm{d}i_b(t)}{\mathrm{d}t} + Ri_b(t) + e_b(\alpha_e(t), \omega_m(t)) \\ u_c(t) = L\frac{\mathrm{d}i_c(t)}{\mathrm{d}t} + Ri_c(t) + e_c(\alpha_e(t), \omega_m(t)), \end{cases} \tag{9}$$

where

$$e_x(\alpha_e(t), \omega_m(t)) = K_{\mathrm{BLDC}} k_{ex}(\alpha_e(t))\omega_m(t)|_{x=a,b,c}. \tag{10}$$

The electromagnetic torque produced in the machine can be calculated using formula

$$T_e(t) = \frac{\sum\limits_{x=a,b,c} e_x(\alpha_e(t), \omega_m(t)) i_x(t)}{\omega_m(t)}, \tag{11}$$

where $e_x(x = a, b, c)$ is the phase back-EMF voltage, K_{BLDC} is the back-EMF constant, $k_{ex}(x = a, b, c)$ is the ideal trapezoidal shape function, L and R denote the stator inductance and resistance, $\alpha_e = p\alpha_m$ is the electrical rotor angle equal to the mechanical rotor angle α_m multiplied by the number of pole pairs, ω_m is the rotor angular speed. If the motor is electronically commutated in such a way that the current flows only through two phases, its dynamics can be modeled using equations similar to the ones describing a brushed DC machine

$$u_{\mathrm{BLDC}}(t) = 2L\frac{\mathrm{d}i_{\mathrm{BLDC}}(t)}{\mathrm{d}t} + 2Ri_{\mathrm{BLDC}}(t) + 2K_{\mathrm{BLDC}}\omega_m(t) \tag{12}$$

$$T_e(t) = 2K_{\mathrm{BLDC}} i_{\mathrm{BLDC}}(t) \tag{13}$$

accompanied by the Newton's law for rotation

$$T_e(t) = J\frac{\mathrm{d}\omega(t)}{\mathrm{d}t} + T_{\mathrm{load}}(t) + F_v\omega_m(t), \tag{14}$$

where T_{load} is the load torque and F_v is the viscous friction coefficient. The BLDC servo drive considered here consists of a hypothetical converter-fed motor equipped with a torque/current PI control loop, tuned with the help of the modulus optimum method, and a cascaded speed and position PI controllers. Parameters of the drive are given in Table 1. All controllers include standard anti-windup algorithm depicted in Fig. 5. Input signals of the speed and torque controllers have per-unit values with the motor nominal values as the base ones. The input of the position controller is left unscaled. The model of the servo drive is then connected to the PSO (Fig. 6) and a performance index is defined as follows

$$\begin{aligned} J_c^{\mathrm{BLDC}} = \frac{T_s}{t_{\mathrm{stop}}} \sum_{k=1}^{\frac{t_{\mathrm{stop}}}{T_s}} \big(f_1^{\mathrm{BLDC}}(k) e_\alpha^2(k) + f_2^{\mathrm{BLDC}}(k)\left(\Delta T_e^{\mathrm{ref}}(k)\right)^2 \\ + f_3^{\mathrm{BLDC}}(k) e_\alpha^2(k)\big), \end{aligned} \tag{15}$$

Table 1 Parameters of the torque controlled BLDC drive

Parameter	Value	Units
Nominal angular speed	15	$\mathrm{rad} \cdot \mathrm{s}^{-1}$
Nominal voltage	24.6	V
Nominal current	3.33	A
Nominal torque	4	$\mathrm{N} \cdot \mathrm{m}$
Back-EMF constant K_{BLDC}	0.6	Wb
Resistance R	1	Ω
Inductance L	2.5	mH
Moment of inertia J	0.0075[a]	$\mathrm{kg} \cdot \mathrm{m}^2$
Viscous friction F_v	0.0133[b]	$\mathrm{N} \cdot \mathrm{m} \cdot \mathrm{s} \cdot \mathrm{rad}^{-1}$
Converter gain	32	V
Switching frequency	2	kHz
Delay introduced by the modulator	250[c]	µs
Controller sampling time	500[c]	µs
Current acquisition delay	125[c]	µs

[a] including some hypothetical driven machine

[b] 20 % of the nominal torque at the nominal speed (incl. some hypothetical driven machine)

[c] delays discussed in [18] and taken into account when applying the modulus optimum method

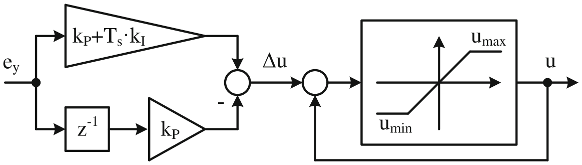


Fig. 5 Discrete implementation of a PI controller with saturation and anti-windup (conditional integration algorithm): k_P—proportional path gain, k_I—integral path gain, T_s—controller sampling time, $u_{\min}$ and $u_{\max}$ are minimal and maximal control signal levels

where the function f_1^{BLDC} introduces penalty for position control error after the time equal to a theoretical time needed to travel a test angle under assumption that the only physical limits are these related to maximal absolute values of electromagnetic torque and angular speed, the function f_2^{BLDC} switches off the penalty for reference torque variations during transients forced by the test reference speed and test disturbance torque, and the function f_3^{BLDC} penalizes for an overshoot.

For the discussed PI controllers tuning task, each particle is a vector of candidate settings for both controllers. These settings can be explicitly stored as the vector components. However, it has been tested that in some tuning problems it is easier and sometimes also more effective to perform the search in an exponential scale. For the purpose of this search, the particle is a vector

$$\boldsymbol{p}^{\mathrm{BLDC}} = [\log_{10} k_{\mathrm{P}\omega}, \log_{10} k_{\mathrm{I}\omega}, \log_{10} k_{\mathrm{P}\alpha}, \log_{10} k_{\mathrm{I}\alpha}], \quad (16)$$

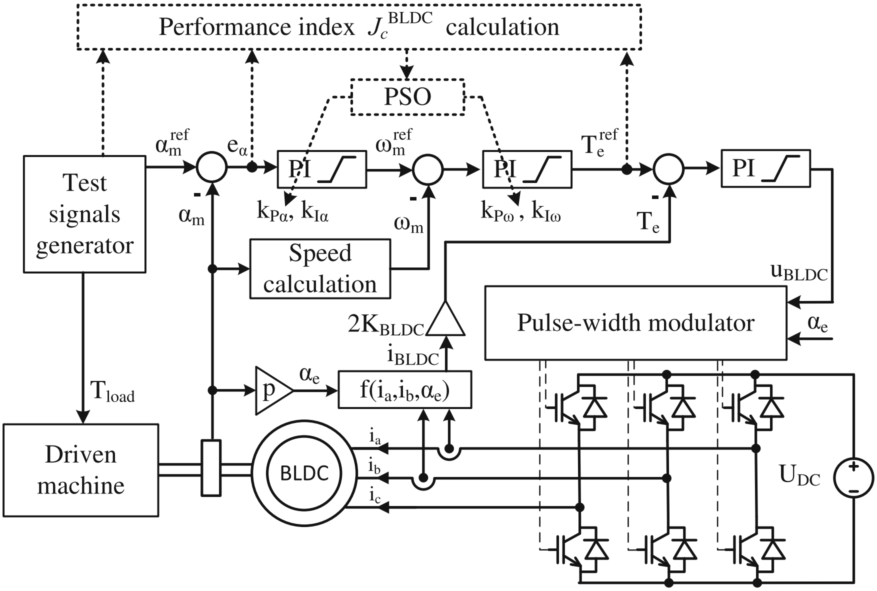


Fig. 6 The BLDC servo drive connected to the PSO system

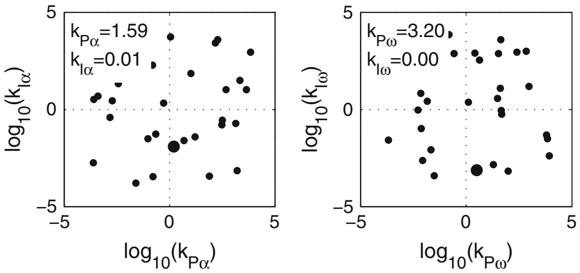


Fig. 7 Initial particles position—the bigger in diameter *black dot* denotes *gbest*

where $k_{P\omega}$, $k_{I\omega}$, k_{Pa} and $k_{I\alpha}$ are the controllers' gains. The swarm consists of 30 particles and is stopped arbitrary after 100 iterations. The diversity (7) and standard deviations present in (8) are inspected visually. In the case of any stochastic search algorithm each search attempt is unique due to the presence of random variables in the speed update rule (1). It is recommended to repeat several times the search, to be able to assess how conditioned the problem is. If the search process is repeatable in terms of a final position of the swarm, the optimization task is well-posed. For illustrative purposes selected iterations are shown in Figs. 7, 8, 9. The evolution of the performance index J_c^{BLDC} for the *gbest* solution is depicted in Fig. 10. However, the most informative are standard deviations shown in Fig. 11. They clearly indicate that the solutions proposed by the swarm

Fig. 8 Position of particles after five iterations

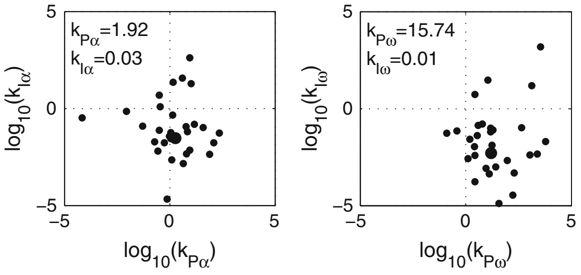


Fig. 9 Position of particles after 100 iterations—swarm near its equilibrium

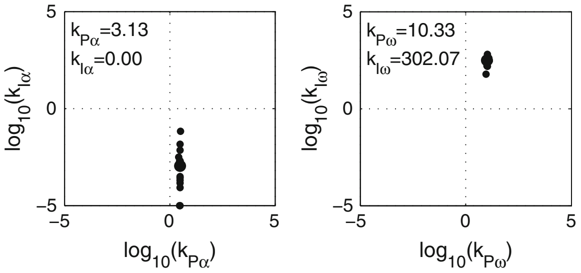
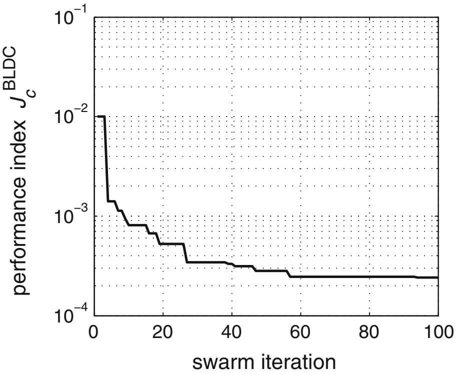


Fig. 10 Evolution of the performance index J_c^{BLDC} for *gbest* solution during the search



for the quadruple of controller gains are convergent except the gain $k_{\mathrm{I}\alpha}$ in the integral path of the position controller. The reason for this is that in the test scenario used for assessing a particle the dominant term comes from a steady state error for a constant reference. A physical integration present in the plant, since the angular position controlled in the outer loop is the integral of the speed controlled in the inner loop, is then sufficient to accomplish the objective. It should be noted that the proposed *gbest* for $k_{\mathrm{I}\alpha}$ is near zero (see Fig. 9). The dynamics of the system for the *gbest* after 100 search iterations is illustrated in Fig. 12. The swarm has identified that the assumed controller topology is excessive for this task. The behavior of the swarm illustrates its ability to prompt to the designer potential

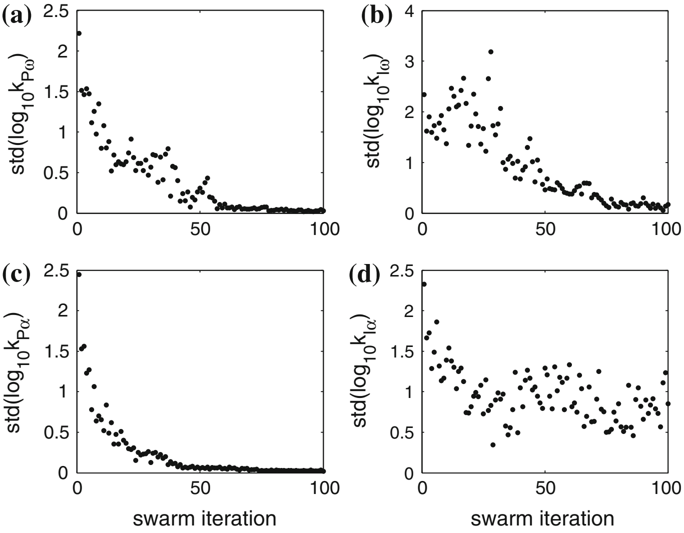


Fig. 11 Evolution of standard deviations during the BLDC servo drive tuning

further refinements of the assumed topology. This can be especially helpful when a controller for a gray-boxed or a black-boxed plant has to be designed. Obviously, there is a possibility to extend such a random search into simultaneous topology selection and gain tuning. Two typical situations could be addressed: optimization of a semi-fixed-structure controller or selecting the best fixed-structure controller from a set of predefined structures. The former refers, e.g., to determining the number of neurons in a neurocontroller. The control structure is fixed—the type of artificial neural network is assumed to be fixed—the only decision variable related to the structure is the number of hidden neurons. On the other hand, the latter refers to a situation where a set of structurally different controllers is tested by the particles, i.e. one entry of the particle vector is a pointer to a set of predefined structures. This is especially useful for black-boxed plants when little or no information about the dynamics of the process is available. The designer can then define a set of potentially applicable control structures: cascaded PIs, augmented-state feedback controllers, various neurocontrollers, iterative learning controllers, repetitive controllers, etc. The PSO will then find the most suitable one for a given black-boxed process.

4 Adaptive Online Trained Speed Neurocontroller

Nonlinear and adaptive speed controllers are often used to cope with inertia variations present in many applications. A robotic arm or an electric vehicle carrying various loads can serve as the examples. A direct method assumes

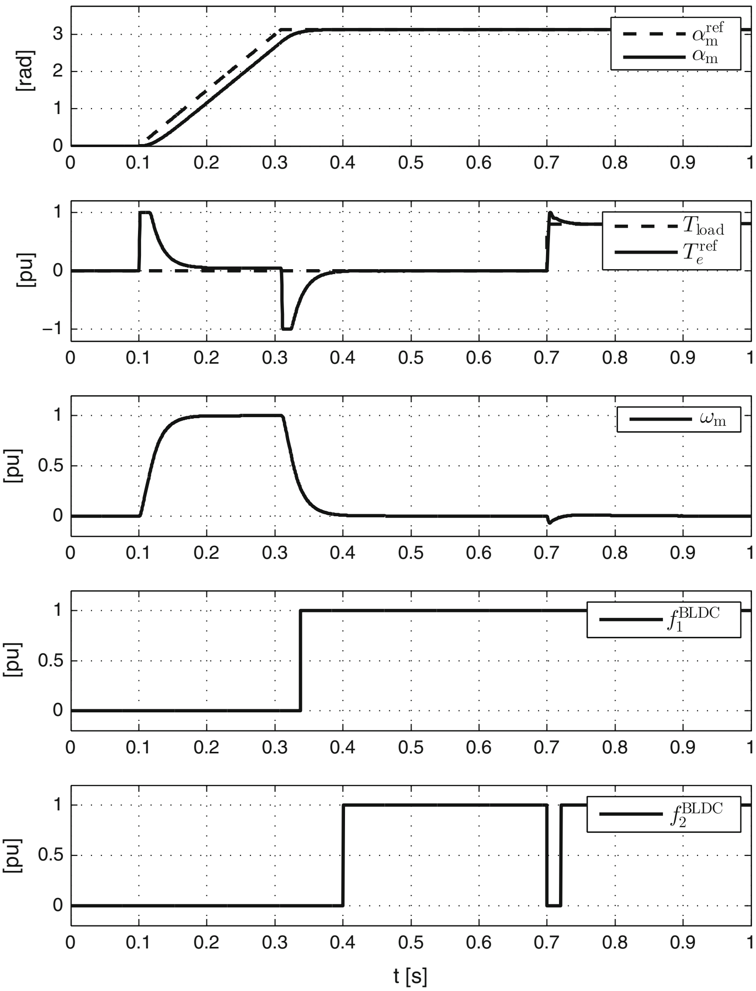


Fig. 12 Response of the BLDC servo drive tuned by the PSO

introduction of an inertia estimator into the system (see e.g. [19]). In indirect methods no inertia is estimated explicitly. These methods usually take advantage of introducing nonlinearity into the controller with the intention to reduce sensitivity to variations of plant parameters. Fuzzy logic (FL), artificial neural networks (ANN) and their combinations are commonly used for implementation of nonlinear controllers. This nonlinearity can be static, i.e. determined in an offline optimization procedure, or can be tuned continuously during the regular operation of a drive. A fairy representative examples of online and offline trained neurocontrollers can be found in [20–28]. The online trained neurocontrollers offer natural capability of adaptation. A learning algorithm is kept active during regular operation of the drive [29, 30]. It has been already identified that the resilient backpropagation Rprop algorithm possesses properties that are especially useful if real-time training is considered. The algorithm is less sensitive to noise in comparison to the original backpropagation algorithm because it takes into account only a sign of the gradient. In many tests the Rprop outperforms other first-order training algorithms in terms of convergence [31]. An Rprop modification known as the Rprop with weight-backtracking [31] has been selected in this study. Its pseudocode is as follows:

```
for each $w_{ij}$ do
    if ${\frac{\partial E}{\partial w_{ij}}}^{(k-1)} {\frac{\partial E}{\partial w_{ij}}}^{(k)} > 0$
        $\delta_{ij}^{(k)} = \min(\delta_{ij}^{(k-1)}\eta^{+}, \delta_{\max})$
        $\Delta w_{ij}^{(k)} = -\operatorname{sgn}\left({\frac{\partial E}{\partial w_{ij}}}^{(k)}\right)\delta_{ij}^{(k)}$
        $w_{ij}^{(k+1)} = w_{ij}^{(k)} + \Delta w_{ij}^{(k)}$
    elseif ${\frac{\partial E}{\partial w_{ij}}}^{(k-1)} {\frac{\partial E}{\partial w_{ij}}}^{(k)} < 0$
        $\delta_{ij}^{(k)} = \max(\delta_{ij}^{(k-1)}\eta^{-}, \delta_{\min})$
        $w_{ij}^{(k+1)} = w_{ij}^{(k)} - \Delta w_{ij}^{(k-1)}$
        ${\frac{\partial E}{\partial w_{ij}}}^{(k)} = 0$
    else
        $\Delta w_{ij}^{(k)} = -\operatorname{sgn}\left({\frac{\partial E}{\partial w_{ij}}}^{(k)}\right)\delta_{ij}^{(k)}$
        $w_{ij}^{(k+1)} = w_{ij}^{(k)} + \Delta w_{ij}^{(k)}$
    end
end,
```

where E is the cost function (usually MSE), w_{ij} is the weight of a neural connection, $\delta_{\min}$ and $\delta_{\max}$ are allowable minimal and maximal absolute values of Δw_{ij}, δ_{ij} is the current weight change, η^{-} and η^{+} are decrease and increase factors for δ_{ij}.

If a speed control task is considered and no repetitiveness of this process is assumed (see [32] for more details), the cost function is as follows

$$E_{\mathrm{ANN}}^{\mathrm{SPEED}}(k) = \frac{1}{2}\left(\omega_m^{\mathrm{ref}}(k) - \omega_m(k)\right)^2. \tag{17}$$

It has been tested in several different systems that the most crucial settings are $\delta_{\max}$, η^{-} and η^{+} as far as the training process is equivalent to the control task. Other parameters as the number of neurons, the length of the tapped delay line (TDL) or $\delta_{\min}$ are easy to tune using the guess and check method. The latter can usually be set to zero or to a very small positive value. Some recommendations on $\delta_{\max}$, η^{-} and η^{+} potentially working settings are available in the literature (e.g. [31]). They are reported as suitable for selected offline benchmarks. It was verified that these recommendations cannot be easily extended to online tasks. This only shows that any optimization task is always problem specific (see Sect. 2) and parameters of the Rprop have to be adjusted for a given drive system. Some level

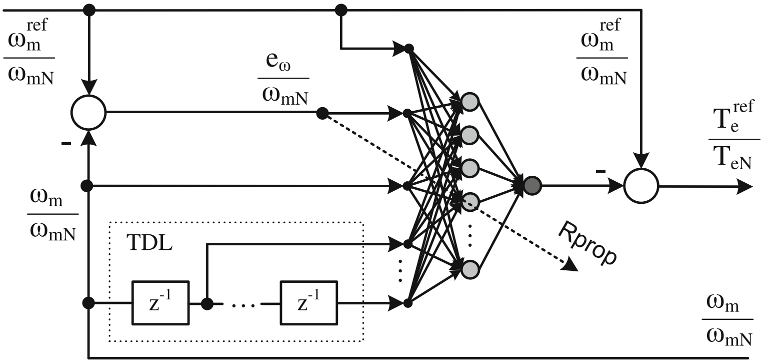

ω_m^{ref} / ω_{mN}
e_ω / ω_{mN}
ω_m^{ref} / ω_{mN}
T_e^{ref} / T_{eN}
-
ω_m / ω_{mN}
-
TDL
z^{-1}
z^{-1}
Rprop
ω_m / ω_{mN}

Fig. 13 Topology of the speed neurocontroller for normalized signals with ω_{mN} and T_{eN} denoting nominal speed and torque values

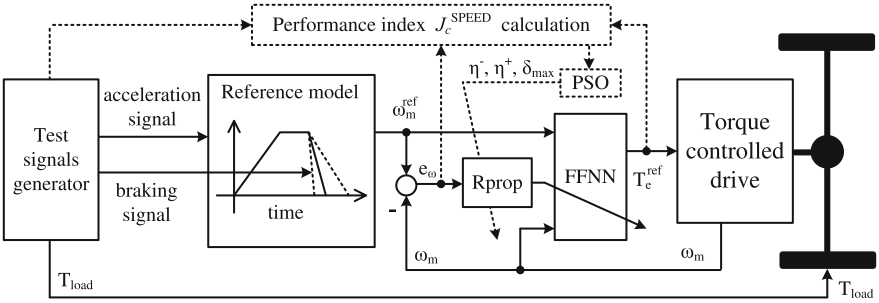


Fig. 14 Adaptive neurocontroller as a part of a vehicle's control system (FFNN stands for the feed-forward neural network)

of automatism can be easily achieved by employing the swarm-based optimization as reported in [33].

For illustrative purposes the controller shown in Fig. 13 has been implemented in a hypothetical drive (Fig. 14) suitable for a passenger city car (of assumed total mass variations equal to $1,500 \pm 300\,\text{kg}$) and then has been optimized using a user defined performance index. Each particle is a vector

$$\boldsymbol{p}^{\text{SPEED}} = [\eta^-, \eta^+, \delta_{\max}] \tag{18}$$

of candidate settings for the `Rprop` adaptation rule. There exist clear search boundaries resulting from the `Rprop` itself. Absorbing walls have been introduced to limit the search to acceptable regions and they are as follows

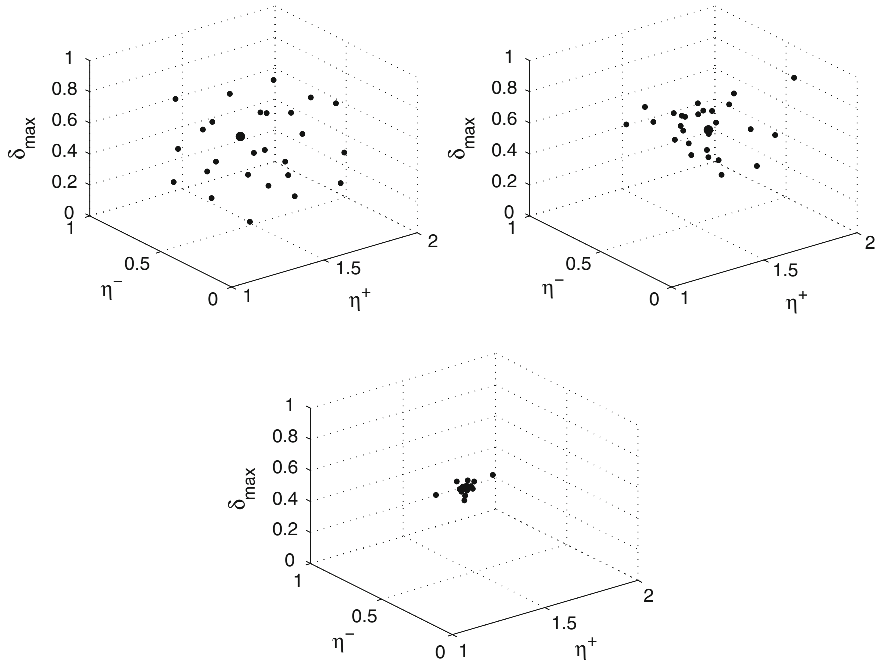


Fig. 15 Swarm position after 1st, 5th and 50th iteration—the bigger in diameter *black dot* denotes *gbest*

$$\begin{cases} 0 < \eta^- < 1 \\ \eta^+ > 1 \\ \delta_{\max} > 0 \end{cases} \tag{19}$$

The swarm consists of 27 particles rated according to the following performance index

$$\begin{aligned} J_c^{\mathrm{SPEED}} = \frac{T_s}{t_{\mathrm{stop}}} \sum_{k=1}^{\frac{t_{\mathrm{stop}}}{T_s}} \Big(& f_1^{\mathrm{SPEED}}(k) e_\omega^2(k) + f_2^{\mathrm{SPEED}}(k) \left(\Delta T_e^{\mathrm{ref}}(k) \right)^2 \\ & + f_3^{\mathrm{SPEED}}(k) e_\omega^2(k) \Big), \end{aligned} \tag{20}$$

where f_3^{SPEED} detects overshooting. The performance index has been changed in comparison to the one proposed in [33] so as to test whether a performance index similar to the one used in Sect. 4 can produce satisfactory results. In the previous work, an expert knowledge about the controller has been incorporated into the performance index definition. Here this knowledge is neglected, i.e. no special measures during the rating related to random initial weights of ANN are taken. The system is assumed to be black-boxed. Variations of the vehicle inertia have been included in the test scenario to optimize the controller for anticipated operating

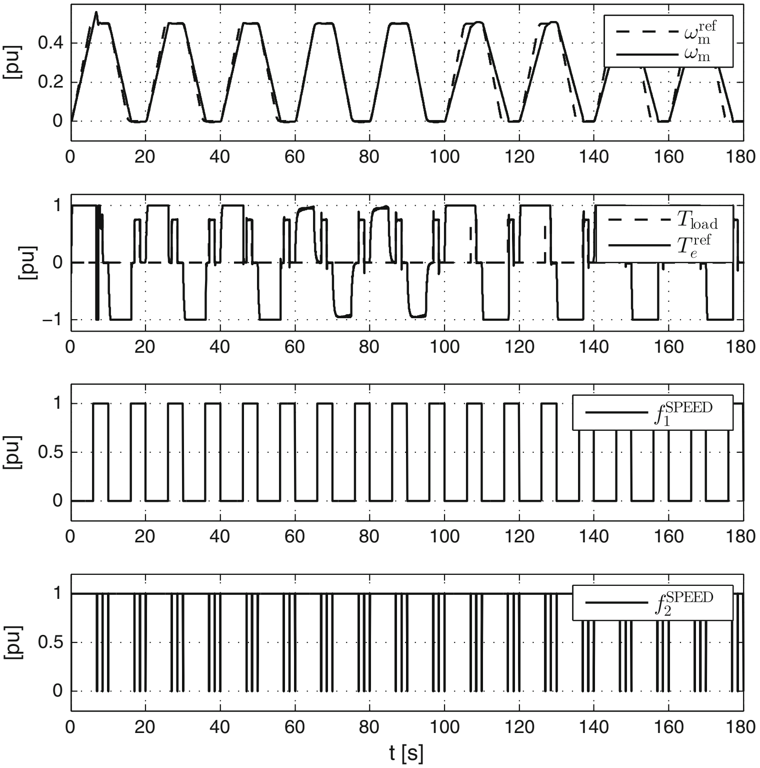


Fig. 16 Response of the drive with the speed neurocontroller tuned by the PSO—moment of inertia drops 20 % at 60 s and rises 20 % at 100 s in comparison to the value set for the first 60 s (assumed total mass variations 1, 500 ± 300 kg)

conditions. The evolution of the swarm is illustrated in Fig. 15. The performance of the drive along with the switching functions used in (20) is shown in Fig. 16. The speed overshoot visible in Fig. 16 during the first acceleration is the result of a random initial weights of the neurocontroller. The training procedure needs some transients on speed to be able to identify the dynamics of the plant. After one vehicle braking no subsequent significant overshoots occur.

5 Augmented Full-State Feedback Controller for a Three-Phase Inverter

Oscillatory controllers have proven to be one of the highest performance alternatives for AC voltage control in many applications, including grid converters, active power filters and sine wave inverters. Examples of such solutions are

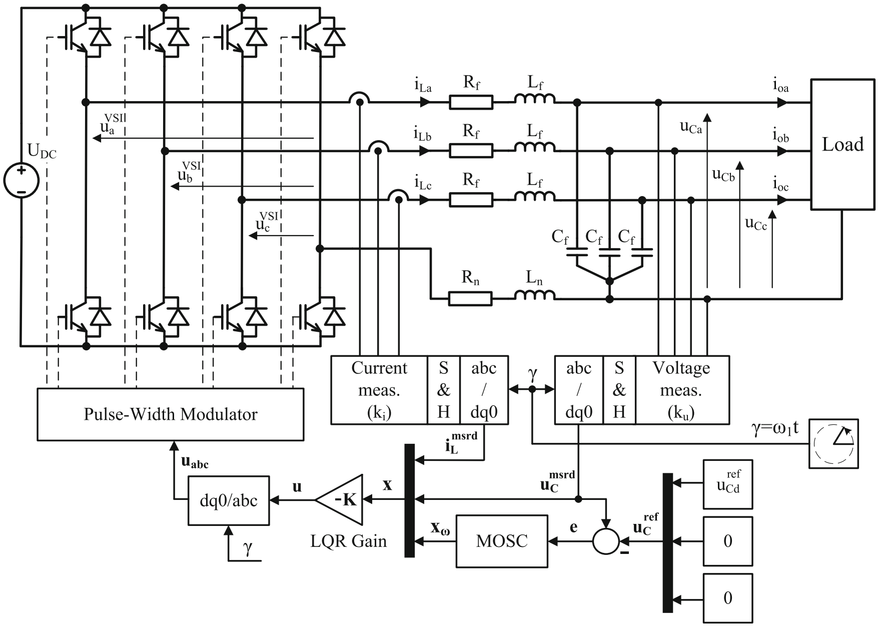


Fig. 17 Three-phase four-leg inverter with LC filter and augmented full state feedback controller

described in [34–38]. The solution considered here relies solely on the LQR design method and has been proposed in [39]. A full-state feedback controller with a state vector augmented to include integral plus multiple oscillatory actions has been implemented in the *dq*0 rotating reference frame with all its gains calculated in one pass using the LQR approach. The LQR design method is known to deliver good performance in practical systems. The resulting system is inherently stable and the controller is relatively simple in coding. The procedure comes down to preparation of a state-space description of an augmented system (plant plus auxiliary controller states), setting weighting matrices in the quadratic cost function and calling the function that solves analytically the optimization problem, known as the discrete-time algebraic Riccati equation (DARE). A three-phase four-leg inverter with an output LC filter depicted in Fig. 17 is considered as the plant to be controlled. Therefore, its mathematical model in the *dq*0 rotating reference frame is as follows

$$\frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{x}_f(t) = \boldsymbol{A}_f^{\mathrm{cont}}\boldsymbol{x}_f(t) + \boldsymbol{B}_f^{\mathrm{cont}}\boldsymbol{u}(t) + \boldsymbol{E}_f^{\mathrm{cont}}z(t), \tag{21}$$

where

$$\boldsymbol{A}_f^{\text{cont}} = \begin{bmatrix} -\frac{R_f}{L_f} & \omega_1 & 0 & -\frac{k_i}{k_u L_f} & 0 & 0 \\ -\omega_1 & -\frac{R_f}{L_f} & 0 & 0 & -\frac{k_i}{k_u L_f} & 0 \\ 0 & 0 & -\frac{R_f+3R_n}{L_f+3L_n} & 0 & 0 & -\frac{k_i}{k_u(L_f+3L_n)} \\ \frac{k_u}{k_i C_f} & 0 & 0 & 0 & \omega_1 & 0 \\ 0 & \frac{k_u}{k_i C_f} & 0 & -\omega_1 & 0 & 0 \\ 0 & 0 & \frac{k_u}{k_i C_f} & 0 & 0 & 0 \end{bmatrix} \tag{22}$$

$$\boldsymbol{B}_f^{\text{cont}} = \begin{bmatrix} \frac{k_{dc}k_i}{L_f} & 0 & 0 \\ 0 & \frac{k_{dc}k_i}{L_f} & 0 \\ 0 & 0 & \frac{k_{dc}k_i}{L_f+3L_n} \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{23}$$

$$\boldsymbol{E}_f^{\text{cont}} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ -\frac{k_u}{C_f} & 0 & 0 \\ 0 & -\frac{k_u}{C_f} & 0 \\ 0 & 0 & -\frac{k_u}{C_f} \end{bmatrix} \tag{24}$$

and

$$\boldsymbol{x}_f(t) = \left[i_{Ld}^{\text{msrd}}(t), i_{Lq}^{\text{msrd}}(t), i_{L0}^{\text{msrd}}(t), u_{Cd}^{\text{msrd}}(t), u_{Cq}^{\text{msrd}}(t), u_{C0}^{\text{msrd}}(t) \right]^T \tag{25}$$

$$\boldsymbol{u}(t) = \left[u_d(t), u_q(t), u_0(t) \right]^T \tag{26}$$

$$z(t) = \left[i_{od}(t), i_{oq}(t), i_{o0}(t) \right]^T, \tag{27}$$

where L_f, R_f, L_n, R_n, C_f and ω_1 denote respectively inductances and resistances of phase and neutral filter legs, capacitance of the filter, and angular speed of the reference frame $dq0$ equal to the fundamental angular frequency of the desired output voltage. The load current vector $z(t)$ represents the unmeasured disturbance. This description already accommodates k_u, k_i, k_{dc} gains that model current and voltage transducers, and the voltage source inverter, respectively. The superscript $\bullet^{\text{msrd}}$ denotes the output signal of a measurement transducer. Thereafter, auxiliary states are introduced to achieve control objectives, i.e. zero steady-state error for the reference frequency and good disturbance rejection for the anticipated load current dominant harmonics. Thus, the auxiliary state variables are as follows

$$\frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{x}_0(t) = \boldsymbol{e}(t) \tag{28}$$

with

$$\boldsymbol{e}(t) = [\underbrace{u_{Cd}^{\mathrm{msrd}}(t) - u_d^{\mathrm{ref}}(t)}_{e_d(t)}, \underbrace{u_{Cq}^{\mathrm{msrd}}(t) - u_q^{\mathrm{ref}}(t)}_{e_q(t)}, \underbrace{u_{C0}^{\mathrm{msrd}}(t) - u_0^{\mathrm{ref}}(t)}_{e_0(t)}]^T \tag{29}$$

and

$$\begin{cases} \frac{\mathrm{d}}{\mathrm{d}t}x_1(t) = x_2(t) \\ \frac{\mathrm{d}}{\mathrm{d}t}x_2(t) = -\omega^2 x_1(t) + e(t), \end{cases} \tag{30}$$

where ω is the desired resonant angular frequency and $e(t)$ denotes a selected control error component from (29). The number and the value of resonant angular frequencies ω are selected individually for each voltage component and have to reflect anticipated load current harmonics seen in the $dq0$ reference frame. The integral term can be regarded as the special case of the oscillatory term with zero resonant frequency. As a result, the auxiliary subsystem can be categorized as a sole multi-oscillatory (MOSC) subsystem

$$\frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{x}_\omega(t) = \boldsymbol{A}_\omega^{\mathrm{cont}}\boldsymbol{x}_\omega(t) + \boldsymbol{B}_\omega^{\mathrm{cont}}\boldsymbol{e}(t), \tag{31}$$

where the auxiliary state vector x_ω refers to all three voltage components and accommodates any desired frequency from available bandwidth. The auxiliary states are merged with the plant states and the augmented state matrix and the input matrix are composed as follows

$$\boldsymbol{A}^{\mathrm{cont}} = \begin{bmatrix} \boldsymbol{A}_f^{\mathrm{cont}} & 0 \\ \begin{bmatrix} 0 & \boldsymbol{B}_\omega^{\mathrm{cont}} \end{bmatrix} & \boldsymbol{A}_\omega^{\mathrm{cont}} \end{bmatrix} \tag{32}$$

$$\boldsymbol{B}^{\mathrm{cont}} = \left[\boldsymbol{B}_f^{\mathrm{cont}}, 0\right]^T. \tag{33}$$

This description is then transformed into the discrete-time domain using ZOH method and weighting matrices **Q** and **R** have to be determined in the quadratic cost function

$$J_{\mathrm{LQ}} = \sum_{k=1}^{\infty}\left(\boldsymbol{x}^T(k)\boldsymbol{Q}\boldsymbol{x}(k) + \boldsymbol{u}^T(k)\boldsymbol{R}\boldsymbol{u}(k)\right) \tag{34}$$

being the part of the LQR definition. The dynamics of the non-disturbed augmented full-state feedback system

$$\boldsymbol{x}(k+1) = (\boldsymbol{A} - \boldsymbol{B}\boldsymbol{K})\boldsymbol{x}(k) \tag{35}$$

Table 2 Selected parameters of the laboratory setup

Symbol	Value and units	Description
DSC	TMS320F28335	Digital signal controller (150 MHz)
U_n	325 V	Nominal output voltage amplitude
ω_1	$2\pi \cdot 50\ \mathrm{s}^{-1}$	Output reference angular frequency
f_{PWM}, T_s	10 kHz, 100 μs	Switching frequency, sampling time
Dead-time	1.3 μs	Dead-time for IGBT gate signals
U_{DC}	620 V	DC-link voltage
L_f	250 μH	Inductance of the phase leg
R_f	165 mΩ	Series resistance of the phase leg
C_f	85 μF	Capacitance of the output filter
L_n	250 μH	Inductance of the neutral leg
R_n	165 mΩ	Series resistance of the neutral leg
k_u	$1/U_n$	Voltage transducer gain
k_i	$0.01\ \mathrm{A}^{-1}$	Current transducer gain
k_{dc}	U_{DC}	Voltage source inverter gain
Noise	1 %	Measurement noise level

for the zero reference signals is then shaped by $\boldsymbol{K}$ designed using the LQR method. Selecting $\boldsymbol{Q}$ and $\boldsymbol{R}$ is the crucial step and in the most common approach of guessing and checking this step involves an expert knowledge combined with usually numerous trials. Moreover, the resulting controller though optimal according to (34) is not optimal according to commonly used control performance indices as (3), (4) or (5). On the other hand, with the help of a population based optimizer, a full-state feedback controller can be tuned according to a user defined cost function [40–42]. It has been verified that in the case of the discussed controller the performance index

$$J_c^{\mathrm{LQR}} = \frac{T_s}{t_{\mathrm{stop}}} \sum_{k=1}^{\frac{t_{\mathrm{stop}}}{T_s}} \left(\boldsymbol{e}^T(k)\boldsymbol{e}(k) + \beta \Delta \boldsymbol{u}^T(k) \Delta \boldsymbol{u}(k) \right), \tag{36}$$

where β is the weighting factor that directly influences dynamics of the control signal by penalizing this dynamics, is able to produce practical controller gains applicable in the real system in the presence of a measurement noise. In the discussed system the LQR design approach has been kept. However, it should be noticed that a swarm could also perform direct search for $\boldsymbol{K}$ entries [40] or closed-loop poles [41]. If no presumptions concerning the poles or the gains are available, the LQR approach seems to be the most effective if the convergence rate of the stochastic search is taken into account. This conclusion is similar if the augmented state feedback controller is tuned by the human using a trial and error method.

The optimization takes place in offline mode on numerical model of the physical converter with parameters as in Table 2. The method has been verified for the case of the auxiliary states covering 2nd, 3rd, 4th, 6th, 8th, 9th, 10th and 12th harmonic in the dq paths and 1st, 3rd, 5th, 6th, 7th, 9th, 11th and 12th harmonic for

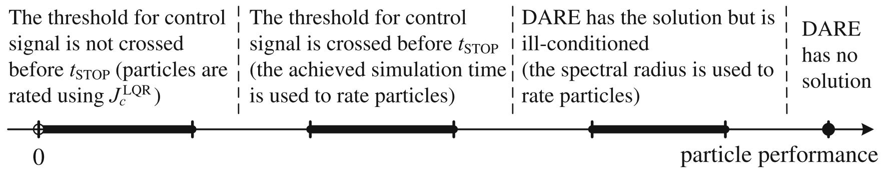

The threshold for control signal is not crossed before t_{STOP} (particles are rated using J_c^{LQR})
The threshold for control signal is crossed before t_{STOP} (the achieved simulation time is used to rate particles)
DARE has the solution but is ill-conditioned (the spectral radius is used to rate particles)
DARE has no solution
0
particle performance

Fig. 18 The order of performance regions for the LQR

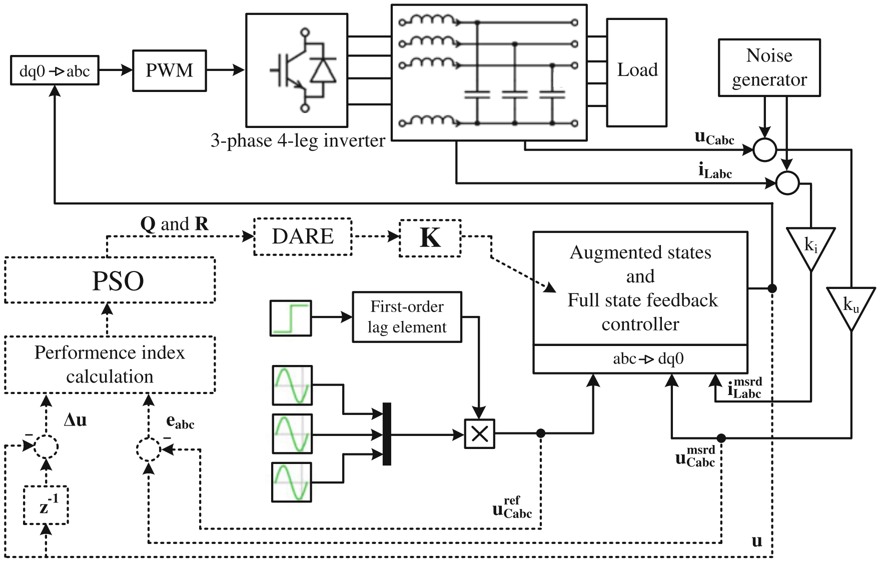


Fig. 19 PSO connected to the numerical model of the three-phase four-leg inverter with the output LC filter

the 0-component. The auxiliary states related to integral actions for all error components are also included. This produces the 59-dimensional optimization problem: 3 entries of $\boldsymbol{R}$ and 57 entries of $\boldsymbol{Q}$ with one entry set arbitrary to 1 as scaling of (34) does not influence the result. The entries in $\boldsymbol{Q}$ are related to: 6 measured state variables, 3 integral actions and $2 \cdot 3 \cdot 8$ oscillatory state variables. It has been decided to merge selected search dimensions to get less challenging problem from wall-clock time perspective. First of all, the Bryson's rule [43] has been applied to normalize the weighting entries of $\boldsymbol{Q}$ and $\boldsymbol{R}$. Next, penalties are not varied for a given harmonic (regardless to its occurrence in the different axis). Moreover, it has been tested that the search performed in an exponential scale is more effective in comparison to the search in a linear scale. This gives entries with the decision variables as exponents, e.g. for 3rd harmonic of the form of $3^2\omega_1^2 10^{q_3}$ and 10^{q_3} for the two auxiliary states introduced by the oscillatory term. The optimization can then be run in 15D space. The particle is a vector

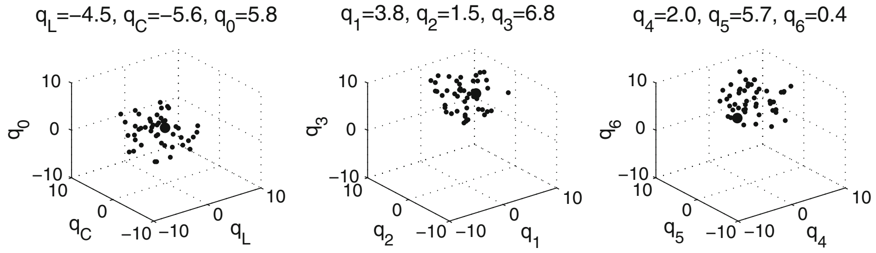


Fig. 20 Initial position of the swarm—the bigger in diameter *black dot* denotes *gbest*

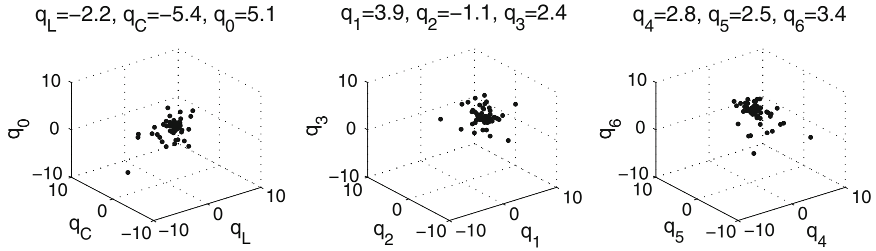


Fig. 21 Position of the particles at 15th iteration, i.e. at the early stage of the search

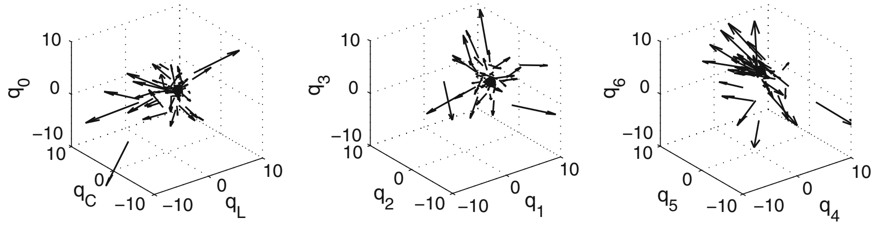


Fig. 22 Velocity vectors of the particles at 15th iteration

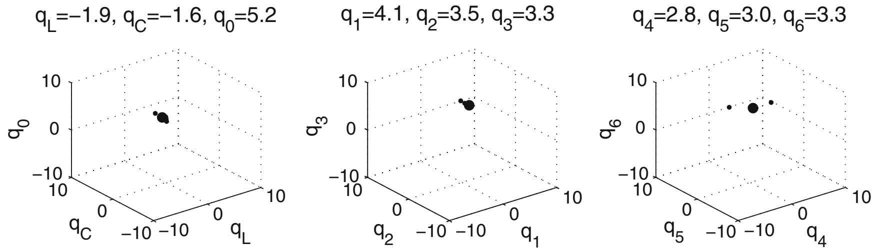


Fig. 23 Position of the swarm near its equilibrium after 75 iterations

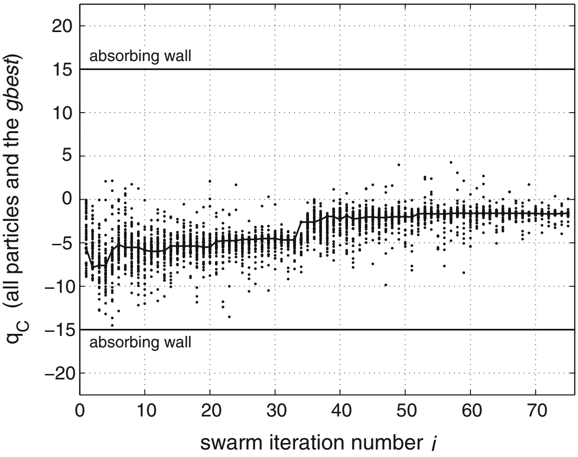


Fig. 24 The evolution of *gbest* (*curved line*) and the swarm position (*dots*) over optimization iterations for q_C

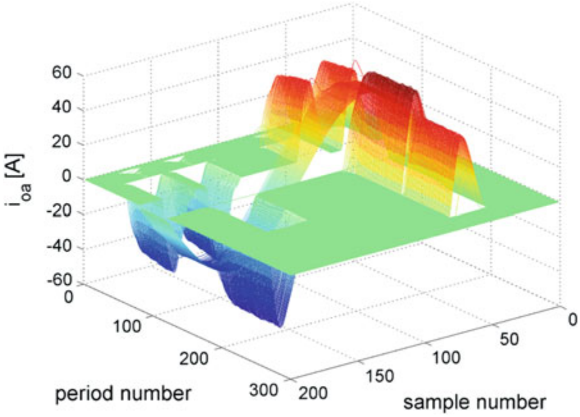
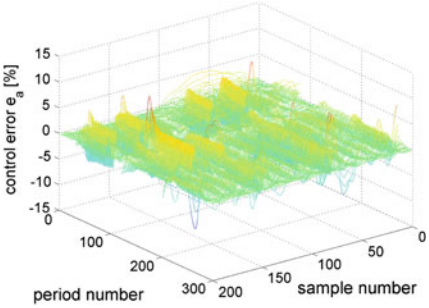


Fig. 25 The test load current (phase *a*)



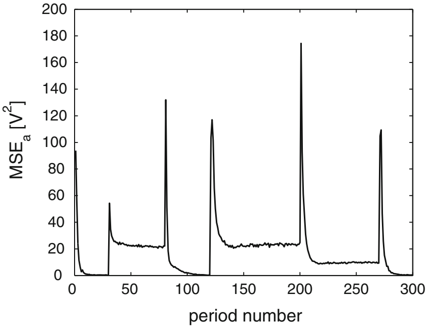


Fig. 26 The performance of the system optimized according to the performance index J_c^{LQR}—instantaneous control error (*on the left*) and MSE per period of the reference signal (*on the right*)

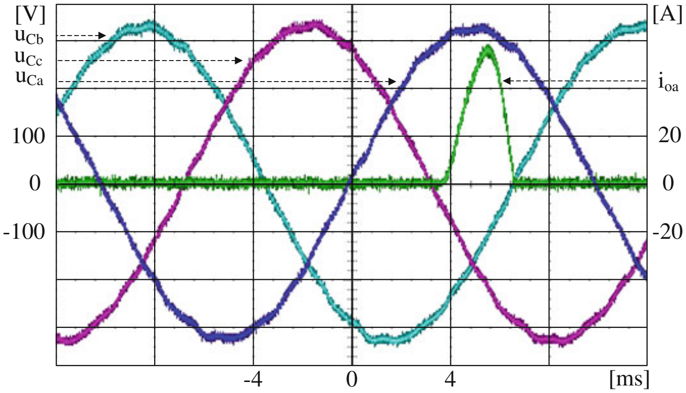


Fig. 27 Output voltage (u_{Ca}, u_{Cb} , u_{Cc}) and output current (i_{oa}) waveforms under the three-pulse diode rectifier load: $U_{\mathrm{Ca}} = 230$ Vrms, $I_{\mathrm{oa}} = 14.14$ Arms, $\mathrm{THD}_{U_{\mathrm{Ca}}} = 1.58$ %, $\mathrm{THD}_{I_{\mathrm{oa}}} = 179$ %

$$\boldsymbol{p}^{\mathrm{LQR}} = [q_L, q_C, q_0, q_1, \ldots, q_{12}] \tag{37}$$

of candidate exponents determining weighting coefficients for the LQR cost function (34). All search directions could have been left unconstrained because of lack of clear physical boundaries. The entries of **Q** are positive for any real-valued particle (37). However, the absorbing walls have been introduced at −15 and 15 to turn the particles back to practical search regions. The walls have been set with considerable surplus according to the experience gathered during previously used trial and error tuning method. Particles that cannot be rated using (36) are handled as in Fig. 18. The swarm of 50 particles is connected to the numerical model of the system as in Fig. 19. Envelopes of the reference voltages seen in natural reference frame are shaped using a first order lag element (see Fig. 2) with a time constant of 0.05 s to avoid excessive contribution to the performance index due to zero initial conditions for the LC filter. Alternatively, a switching function could be introduced in (36) as discussed in Sect. 1.

Position and speed graphs have been broken into 3D plots. An illustrative selection of such graphs is shown in Figs. 20, 21, 22 and 23. The evolution of the swarm in one selected dimension is shown in Fig. 24. The performance of the resulting system under the load current from Fig. 25 is shown in Fig. 26. The obtained matrix $\boldsymbol{K}$ is transferred to the physical controller without any further alterations. Selected parameters of the laboratory setup are given in Table 2. The tuning procedure from the designer side is only one-dimensional and finding a good β for (36) that produces desired behavior of the physical system usually takes less than five trials. The performance of the physical system under nonlinear loads is illustrated in Figs. 27 and 28. The transient state caused by the step resistive load in one phase (and no-load operation of other phases) is shown in Fig. 29. The harmonic contents are compared in Table 3.

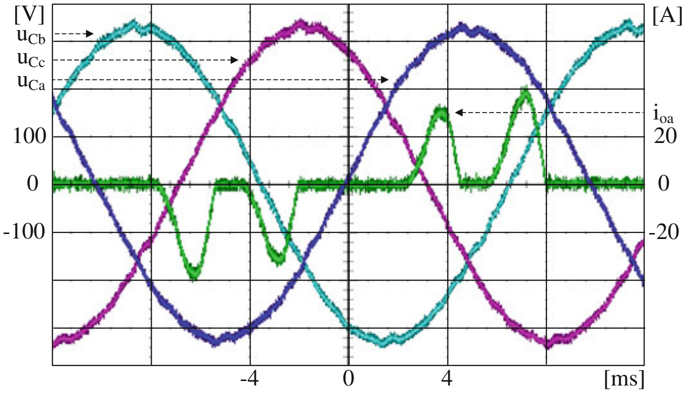


Fig. 28 Output voltage $(u_{\mathrm{Ca}}, u_{\mathrm{Cb}}\,, \ u_{\mathrm{Cc}})$ and output current (i_{oa}) waveforms under the six-pulse diode rectifier load: $U_{\mathrm{Ca}} = 230$ Vrms, $I_{\mathrm{oa}} = 15.11$ Arms, $\mathrm{THD}_{U_{\mathrm{Ca}}} = 1.68\ \%$, $\mathrm{THD}_{I_{\mathrm{oa}}} = 105\ \%$

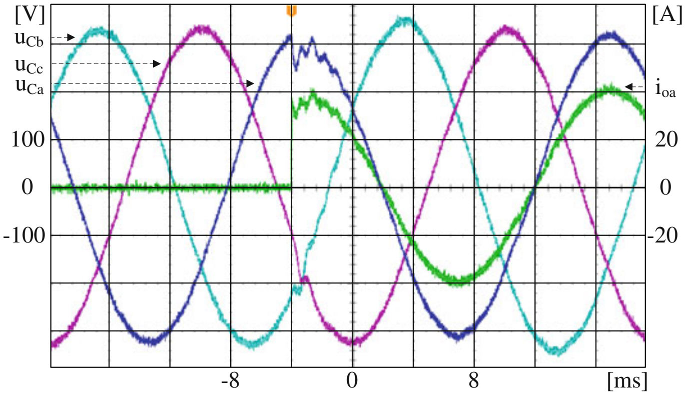


Fig. 29 Transient state on capacitors voltage and output current for the step load $\left(R_{\mathrm{step}} = 8\,\Omega\right)$ in phase *a* (phases *b*, *c* left unloaded)

Table 3 Harmonic spectra of the output voltage (u_{Ca}) and the load current (i_{oa})

	Open loop operation under 3-pulse diode rectifier load		Closed loop operation under 3-pulse diode rectifier load		Closed loop operation under 6-pulse diode rectifier load	
	$U_{\mathrm{rms}} = 227$ Vrms		$U_{\mathrm{rms}} = 230$ Vrms		$U_{\mathrm{rms}} = 230$ Vrms	
	$I_{\mathrm{rms}} = 12.24$ Arms		$I_{\mathrm{rms}} = 14.14$ Arms		$I_{\mathrm{rms}} = 15.11$ Arms	
	$\mathrm{THD}_U = 4.38\,\%$		$\mathrm{THD}_U = 1.58\,\%$		$\mathrm{THD}_U = 1.68\,\%$	
	$\mathrm{THD}_I = 152\,\%$		$\mathrm{THD}_I = 179\,\%$		$\mathrm{THD}_I = 105\,\%$	
	$U_1 = 227$ Vrms		$U_1 = 230$ Vrms		$U_1 = 230$ Vrms	
	$I_1 = 6.14$ Arms		$I_1 = 6.42$ Arms		$I_1 = 10.38$ Arms	
h	U (%)	I (%)	U (%)	I (%)	U (%)	I (%)
1	100	100	100	10	100	100
2	0.574	92	0.410	95	0.313	14
3	3.149	80	0.226	86	0.339	5
4	0.612	64	0.128	76	0.139	13
5	0.697	48	0.534	64	0.484	79
6	2.054	32	0.063	52	0.129	2
7	0.190	19	0.276	38	0.558	58
8	0.064	10	0.152	27	0.125	4
9	0.728	7	0.121	17	0.037	3
10	0.081	8	0.110	11	0.092	3
11	0.218	8	0.102	8	0.114	27
12	1.488	7	0.031	8	0.094	1
13	0.163	5	0.163	9	0.212	12

6 Conclusions

Nowadays, control systems become more and more elaborated and related tuning procedures often require theoretical insight into the system. And even then, analytical tuning procedures available for these systems usually forces a designer to guess some parameters that are needed as their input arguments. Also often these arguments do not directly shape the dynamics of the closed-loop system, and consequently achieving the desired behavior of the system is not a straightforward task. That is why still cascaded PI controllers are dominant in the industrial practice. They earn their popularity because of relatively simple tuning methods and low computational complexity. Theoretically, any controller tuning task can be redefined into a performance index optimization problem. Relevant controller settings turn into decision variables. Such an optimization problem can be solved with little or no insight into the system by using gradientless population-based optimizers. This approach can be applied to problems that do not have yet an analytical solution as well as to problems that do have one. In the case of the latter this can result in much more straightforward procedure from the designer point of view in comparison to the original analytical solution. It has been illustrated that a swarm of particles can support control engineers in a simultaneous tuning of cascaded controllers, in identifying potential excessiveness of a control structure (as in Sect. 4), in reducing dimensionality of the problem in terms of number of parameters that have to be passed to the function by the user (as in Sect. 6). Moreover, swarms are extremely useful when a user-defined performance index should be addressed in the system and analytical solution has not been yet developed. It has been shown that swarms can help to tune adaptive neurocontrollers (as in Sect. 5) that otherwise would have to be tuned by using a time-consuming trial and error method accompanied by an expert knowledge. It should be noted that the PSO is itself a trial-and-error-like method. However, points that are to be visited in the solution space are determined by the swarm itself in the iterative manner. The visual inspection of a performance often used during human made trials is replaced by rating solutions according to a user defined real-valued performance index. In most practical engineering problems no analytical solution is expected as far as stochastic search is able to deliver good suboptimal solution and this turns out to be achievable if some basic expert knowledge is incorporated into the performance index definition.

References

1. R.C. Eberhart, Y. Shi, J. Kennedy, *Swarm Intelligence*, 1st edn. (The Morgan Kaufmann Series in Evolutionary Computation, Morgan Kaufmann Publishers, Burlington, 2001)
2. A. Banks, J. Vincent, C. Anyakoha, A review of particle swarm optimization. Part I: background and development. Nat. Comput. **6**(4), 467–484 (2007)
3. A. Banks, J. Vincent, C. Anyakoha, A review of particle swarm optimization. Part II: hybridisation, combinatorial, multicriteria and constrained optimization, and indicative applications. Nat. Comput. **7**(1), 109–124 (2008)
4. D. Wolpert, W. Macready, No free lunch theorems for optimization. IEEE Trans. Evolut. Comput. **1**(1), 67–82 (1997)
5. R. Mendes, J. Kennedy, J. Neves, The fully informed particle swarm: simpler, maybe better. IEEE Trans. Evolut. Comput. **8**(3), 204–210 (2004)
6. J. Fernandez-Marquez, J. Arcos, Adapting particle swarm optimization in dynamic and noisy environments, in *IEEE Congress on Evolutionary Computation* (*CEC* 2010) (2010), pp. 1–8
7. X. Hu, R. Eberhart, Multiobjective optimization using dynamic neighborhood particle swarm optimization, in, *Proceedings of the* 2002 *Congress on Evolutionary Computation* (*CEC* 2002), vol. 2, pp. 1677–1681, 2002
8. A. Röhler, S. Chen, An analysis of sub-swarms in multi-swarm systems, in *AI 2011: Advances in Artificial Intelligence*, vol. 7106, Lecture Notes in Computer Science, ed. by D. Wang, M. Reynolds (Springer, Berlin, 2011), pp. 271–280
9. J. Kennedy, R. Mendes, Neighborhood topologies in fully informed and best-of-neighborhood particle swarms. IEEE Trans. Syst. Man Cyber. Part C Appl. Rev. **36**(4), 515–519 (2006)
10. J. Robinson, Y. Rahmat-Samii, Particle swarm optimization in electromagnetics. IEEE Trans. Antennas Propag. **52**(2), 397–407 (2004)
11. M.A. Rahimian, M.S. Tavazoei, Improving integral square error performance with implementable fractional-order PI controllers. Optimal Control Appl. Methods (In press) (2013)

12. I.C. Trelea, The particle swarm optimization algorithm: convergence analysis and parameter selection. Info. Process. Lett. **85**(6), 317–325 (2003)
13. K. Zielinski, R. Laur, Stopping criteria for a constrained single-objective particle swarm optimization algorithm. Informatica (Slovenia) **31**(1), 51–59 (2007)
14. J. Riget, J. Vesterstrøm, A diversity-guided particle swarm optimizer—the ARPSO. Technical report (Aarhus Universitet, Aarhus, 2002)
15. K.J. Åström, T. Hägglund, *PID Controllers: Theory*, *Design*, *and Tuning*, 2 edn. (Instrument Society of America, Research Triangle Park, Charlotte, 1995)
16. K.J. Åström, T. Hägglund, Revisiting the Ziegler–Nichols step response method for PID control. J. Process Control **14**(6), 635–650 (2004)
17. M. Joost, K. Zielinski, B. Orlik, R. Laur, Robust PI cascade control for a multi-mass system optimized by evolutionary algorithms, in, *The 13th Power Electronics and Motion Control Conference*, (*EPE-PEMC* 2008) (2008), pp. 1064–1070
18. T. Nussbaumer, M. Heldwein, G. Gong, S. Round, J. Kolar, Comparison of prediction techniques to compensate time delays caused by digital control of a three-phase buck-type PWM rectifier system. IEEE Trans. Ind. Electron. **55**(2), 791–799 (2008)
19. J.W. Choi, S.C. Lee, H.G. Kim, Inertia identification algorithm for high-performance speed control of electric motors. IEEE Proc. Electr. Power Appl. **153**(3), 379–386 (2006)
20. M.O. Efe, in *Neural Network-based Control*, ed. by B.M. Wilamowski, J.D. Irwin. The Industrial Electronics Handbook: Intelligent Systems, vol. 3 (CRC Press, Boca Raton, 2011)
21. J. Sarangapani, *Neural Network Control of Nonlinear Discrete-Time Systems*, (CRC Press, Boca Raton, 2006)
22. P. Potocnik, I. Grabec, Adaptive self-tuning neurocontrol. Math. Comput. Simulat. **51**, 201–207 (2000)
23. Y.C. Wang, C.J. Chien, D.T. Lee, An output recurrent fuzzy neural network based iterative learning control for nonlinear systems, in *IEEE International Conference on Fuzzy Systems* (*FUZZ-IEEE* 2008), *IEEE World Congress on Computational Intelligence* (2008), pp. 1563–1569
24. Q. Zhu, L. Guo, Stable adaptive neurocontrol for nonlinear discrete-time systems. IEEE Trans. Neural Netw. **15**(3), 653–662 (2004)
25. J.H. Park, S.H. Kim, C.J. Moon, Adaptive neural control for strict-feedback nonlinear systems without backstepping. IEEE Trans. Neural Netw. **20**(7), 1204–1209 (2009)
26. J. Levin, N. Perez-Arancibia, P. Ioannou, T.C. Tsao, A neural-networks-based adaptive disturbance rejection method and its application to the control of hard disk drives. IEEE Trans. Magnet. **45**(5), 2140–2150 (2009)
27. T. Pajchrowski, K. Zawirski, Application of artificial neural network to robust speed control of servodrive. IEEE Trans. Ind. Electron. **54**(1), 200–207 (2007)
28. T. Orlowska-Kowalska, K. Szabat, Control of the drive system with stiff and elastic couplings using adaptive neuro-fuzzy approach. IEEE Trans. Ind. Electron. **54**(1), 228–240 (2007)
29. L. Grzesiak, V. Meganck, J. Sobolewski, B. Ufnalski, On-line trained neural speed controller with variable weight update period for direct-torque-controlled AC drive, in *12th International Power Electronics and Motion Control Conference* (*EPE-PEMC* 2006) (2006), pp. 1127–1132
30. L. Grzesiak, V. Meganck, J. Sobolewski, B. Ufnalski, Genetic algorithm for parameters optimization of ANN-based speed controller, in *The International Conference on Computer as a Tool* (*EUROCON* 2007) (2007), pp. 1700–1705
31. C. Igel, M. Hüsken, Empirical evaluation of the improved `Rprop` learning algorithms. Neurocomputing **50**, 105–123 (2003)
32. B. Ufnalski, L. Grzesiak, Artificial neural network based voltage controller for the single phase true sine wave inverter—a repetitive control approach. Electr. Rev. (Przeglad Elektrotechniczny, paper in English, open access at pe. org. pl), **89**(4), 14–18 (2013)
33. B. Ufnalski, L. Grzesiak, Particle swarm optimization of artificial-neural-network-based online trained speed controller for battery electric vehicle. Bull. Polish Acad. Sci. Tech. Sci. **60**(3), 661–667 (2012)

34. D. De, V. Ramanarayanan, A proportional + multiresonant controller for three-phase four-wire high-frequency link inverter. IEEE Trans. Power Electron. **25**(4), 899–906 (2010)
35. G. Bonan, O. Mano, L. Pereira, D. Coutinho, Robust control design of multiple resonant controllers for sinusoidal tracking and harmonic rejection in uninterruptible power supplies, in *IEEE International Symposium on Industrial Electronics* (*ISIE* 2010) (2010), pp. 303–308
36. M. Dai, M. Marwali, J.W. Jung, A. Keyhani, A three-phase four-wire inverter control technique for a single distributed generation unit in island mode. IEEE Trans. Power Electron. **23**(1), 322–331 (2008)
37. C. Liu, F. Blaabjerg, W. Chen, D. Xu, Stator current harmonic control with resonant controller for doubly fed induction generator. IEEE Trans. Power Electron. **27**(7), 3207–3220 (2012)
38. A. Hasanzadeh, C.S. Edrington, B. Maghsoudlou, F. Fleming, H. Mokhtari, Multi-loop linear resonant voltage source inverter controller design for distorted loads using the linear quadratic regulator method. Power Electron. IET **5**(6), 841–851 (2012)
39. A. Kaszewski, L. Grzesiak, B. Ufnalski, Multi-oscillatory LQR for a three-phase four-wire inverter with L3nC output filter, in *38th Annual Conference on IEEE Industrial Electronics Society* (*IECON* 2012) (2012), pp. 3449–3455
40. S. Karanki, M. Mishra, B. Kumar, Particle swarm optimization-based feedback controller for unified power-quality conditioner. IEEE Trans. Power Deliv. **25**(4), 2814–2824 (2010)
41. J. Wang, B. Brackett, R. Harley, Particle swarm-assisted state feedback control: from pole selection to state estimation, in *American Control Conference* (*ACC* 2009) (2009), pp. 1493–1498
42. A. Hasanzadeh, C. Edrington, J. Leonard, Y. Liu, VSI optimal controller tuning with LQR-based gain space determination and PSO finalization in LEV drive, in *IEEE Vehicle Power and Propulsion Conference* (*VPPC* 2011) (2011), pp. 1–6
43. F. Gene, J.D.P. Franklin, M.L. Workman, *Digital Control of Dynamic Systems*, 3rd edn. (Addison Wesley Longman, Boston, 1998)