WYŻSZA SZKOŁA ZARZĄDZANIA i ADMINISTRACJI W OPOLU
WSZiA
OPOLE
1996

THE ACADEMY OF MANAGEMENT
AND ADMINISTRATION IN OPOLE

INNOVATION MANAGEMENT IN AGRICULTURE. AGROTRONICS AND DESIGN OF OPTIMAL CONTROLLERS BASED ON NEW MODIFICATIONS OF PARTICLE SWARM OPTIMIZATION

THE ACADEMY OF MANAGEMENT AND ADMINISTRATION IN OPOLE

Yuriy Romasevych, Viatcheslav Loveikin, Mikola Ohiienko, Lyubov Shymko,
Kazimierz Łukawiecki

INNOVATION MANAGEMENT IN AGRICULTURE. AGROTRONICS AND DESIGN OF OPTIMAL CONTROLLERS BASED ON NEW MODIFICATIONS OF PARTICLE SWARM OPTIMIZATION

Monograph

Opole 2021

ISBN 978-83-66567-39-9

Innovation management in agriculture. Agrotronics and design of optimal controllers based on new modifications of particle swarm optimization. *Monograph.*, 2021; ISBN 978-83-66567-39-9; pp. 122, illus., tabs., bibls.

Editorial Office:
Wyższa Szkoła Zarządzania i Administracji w Opolu 45-085 Polska, Opole, ul. Niedziałkowskiego 18 tel. 77 402-19-00/01 E-mail: info@poczta.wszia.opole.pl

Recommended for publication
by the Academic Council of Research Institute of Engineering and Technology
of National University of Life and Environmental Science of Ukraine
(Protocol No. 11 of May 21, 2021)

Reviewers:
dr hab. Wojciech Duczmal, prof. dr hab. Henryk Sobczuk,
prof. dr hab. Eugeniusz Krasowski

Authors of Monograph
Yuriy Romasevych, Viatcheslav Loveikin, Mikola Ohiienko, Lyubov Shymko,
Kazimierz Łukawiecki

Publishing House:
Wyższa Szkoła Zarządzania i Administracji w Opolu 45-085 Polska, Opole,
ul. Niedziałkowskiego 18 tel. 77 402-19-00/01

200 copies

Authors are responsible for content of the materials.

ISBN 978-83-66567-39-9

TABLE OF CONTENTS

PREFACE

Automatic controllers are used in modern production operations. Their application influences the quality indicators of the automated processes. That is why there are a lot of scientific researches, which are dedicated to the methods of development of optimal controllers and optimal tuning of classical controllers (for instance, PI-controllers). However, the development of technologies causes new problems of control, which are cannot be solved with the known scientific methods. Here we may state the following thesis „new problems should be solved with the new methods”. In the presented monograph authors have tried to develop new approaches in that specific area.

In chapter 1 authors have focused their attention on the particle swarm optimization method. They analyzed the areas of its application and stated the main features of PSO utilization for different problems. Based on the PSO weakness, which is connected with the possibility of stacking into local minima, the authors have proposed to overcome it with a swarm reinitialization technique. Another positive modification of PSO algorithm was an increase of swarm diversity. Both modifications have been investigated in the chapter and they have shown promising results in terms of localization of global minimum of benchmark functions and efficiency of computation recourses using.

The main goal of chapter 2 is connected with the development of the general methodology of PI-controllers optimization. Authors have stated the problem in a general form (including constraints), and reduced it to the problem of unconstrained optimization. In order to prove the efficiency of the developed approach, several transfer functions were used in the carried out calculations. They refer to the plants of second, third and forth orders, with delays, with zeros and poles, etc. Such a variety of transfer functions is caused by the desire to prove the generality of the methodology. ME-PSO and ME-D-PSO have been used for finding the solutions of the optimization problems. Obtained results have been compared with those, which refer to common in engineering practice tuning methods. Optimal-based results are better almost for all of the quality indicators.

Chapter 3 is the largest one. Here researchers have unwinded the previous methodology to the problem of neural controller training. All the carried out calculations refer to the training of artificial neural network in „reinforced” paradigm. The number of arguments, which should be calculated, is much bigger, than for PI-controllers. However, obtained results support the statement, that optimal neurocontrollers may be used for quite complicated control problems (for example, with non-linear, unstable, multivariable plants etc.). In the chapter authors have proved, that neurocontroller training may be effectively provided with PSO-based approaches.

The presented in the monograph insights may be used in the allied sciences.

The practical usefulness of the research relates to the recommendations for tuning of P-, PI-, or PID-controllers, and neurocontrollers; development of efficient linear or non-linear controllers; solving of complicated scientific and applied problems in the area of technical sciences, etc. Their application provides increasing control quality of different automated processes.

Most of the monograph’s content has been obtained in the frame of scientific research supported by the public grant of Ukraine „Development of high-efficient automatic controllers” (registration number 0119U100758). It was conducted by scholars of National University of Life and Environmental Sciences of Ukraine.

The monograph is useful for specialists in the area of automation and optimal control, scientists, workers, and developers of the control systems, students, and graduate students of higher technical institutions.

CHAPTER 1. AGROTRONICS OF DEVELOPMENT OF NEW MODIFICATIONS OF PSO

1.1 Engineering management of ME-PSO

Bio-inspired optimization methods have great spread in many fields of the human activities [1]. The reasons are linked to their calculation advantages and implementation simplicity in different applications. One of the most powerful methods within such a class of methods is PSO [2]. The number of its' applications is huge [2-6]. PSO and various modifications have been used for: learning and designing of artificial neural networks [2], calculations of various control problems [3, 4], signal processing [5], design [6], sentiment analysis [7], programming problems [8] etc. Almost in all of the referenced works were used modifications of the canonical PSO.

Indeed, many optimization problems have complications (stochastic influences [9], non-linearity [10, 11], multidimensionality and multi-extremal features [12], multi-objectivity [8, 13, 14], necessity to find the global extremum, etc.) which cause attempts of deep modification of PSO.

In this paper, we present a proposal, called „Multi-Epoch Particle Swarm Optimization” (ME-PSO). This method allows to improve significantly the exploration ability of a swarm and makes possible to use computation resources in a more effective manner.

According to PSO, at the beginning of the search process every particle in the swarm has random position. Rather quickly PSO finds local optimum, after that better local optimum and so on and so forth. As the algorithm continues the successive local optimums slightly differs and the quantity of iterations, required to reach further local optimums, is extending. Hence, the swarm tends to stagnation and efficiency of the PSO respectively reduces. This problem is known as a premature convergence.

In order to overcome the premature convergence and improve PSO exploration ability many of its modifications have been proposed. Such modifications envisage various strategies: mutation [7], different topologies of the particles' connections [15] and topology variation [16], alteration of a swarm population [17], changing parameters of a swarm [18], varying the initial position and velocity of a swarm [19], adding extra terms in velocity expression or modification of canonical velocity formula [20], using many swarms in co-evolution interaction [21], integration PSO with other optimization methods [22] etc. Note that presented classification is not completed.

Some of the mentioned modifications have shown a good performance for test optimization problems. Nevertheless, there is a lack of the PSO modifications which allow to overcome the premature convergence in a simple manner. Here we mean the algorithms without high calculation complication, the algorithms which are similar in simplicity to the canonical PSO. Hence, the further studies in the area of PSO-based techniques should be continued.

1.1.1 ME-PSO description

In the PSO method, a swarm is a set of particles which move on the surface of a minimized function in order to find global minimum of the function. During its movement particles improve the found minima and exchange information with their neighbours. The position of the i-th particle is a set of its coordinates (x_{i1}, x_{i2},…, x_{iD}) in the search domain with dimensionality D.

At the initial stage of the PSO algorithm, the particles' positions are randomly initialized. Each particle also described by the velocity vector, which is usually zero-vector for the initial iteration. During subsequent iterations, the components of the position and velocity vectors of a particle are updating according to the formulas:

$$\begin{cases} \tilde{v}_{ij} = wv_{ij} + c_1 r_1 (p_{ij} - x_{ij}) + c_2 r_2 (g_j - x_{ij}); \\ \tilde{x}_{ij} = x_{ij} + \tilde{v}_{ij}, \end{cases} \tag{1.1}$$

where $\tilde{v}_{ij}$ and $\tilde{x}_{ij}$ – are the new j-th component of velocity and position vectors of the *i*-th particle; p_{ij} – the best position, that has been found by the *i*-th particle on the previous iterations (personal best); g_j – the best position, that has been found by the all swarm on the previous iterations (global best); *w* – the inertial coefficient; c_1 and c_2 – cognitive and social coefficients respectively; r_1, r_2 – random numbers that are generated on the interval [0, 1]. The inertial coefficient w determines the influence of the previous velocity of the particle to the $\tilde{v}_{ij}$. The value of the cognitive coefficient c_1 characterizes the degree of individual particle behaviour, its „desire” to move towards personal best. The value of the social coefficient c_2 reflects the degree of collective behaviour, the „desire” to move towards global best.

For the very first iteration the initial positions of the particles are considered as the best.

An iteration of PSO algorithm includes applying the formulas (1.1) and updating the of p_{ij} and g_j values according to the rules:

$$\begin{cases} p_j = x_j, \ \ if \quad f(x_j) < f(p_j); \\ g = p_j, \ \ if \quad f(p_j) < f(g), \end{cases} \tag{1.2}$$

where *f* – is a function to be minimized.

PSO execution provides advantageous exploration of the minimized function. For simple functions PSO commonly finds global minimum. As for topologically complicated functions PSO algorithm finds bad local minimum. This disadvantage is caused by premature convergence of the algorithm. In the following we proposed simple modification of PSO which eliminates mentioned disadvantage.

Proposed ME-PSO algorithm is based on the monitoring of the global optimum search performance (herein, for clarity sake, we will refer as minimum of a function). The main idea of the novel technique is the following: if the rate of the swarm global best reducing is low then all the swarm particles positions must be reinitialized in a random way (the new epoch of the swarm commences). The global best of the swarm (in the new epoch) for the first iteration is the same as it was for the last iteration of the swarm in the previous epoch.

Moving on the surface of a function a particle may trap in a minimum that would be better than the previous epoch global best. It should be noted that for some number of iterations (just after reinitialization of the swarm) the particles move without any improvement of the global best. Soon a particle may find the local minimum that is the better than the current global best.

We should set a criterion of swarm stagnation. Such criterion may be global best reduction rate described by the following expression:

$$R = \frac{GB_i - GB_{i-1}}{GB_i}, \tag{1.3}$$

where GB_i and GB_{i-1} – are global bests of a swarm for the i-th (current) and (i-1)-th (previous) iterations. Equation (1.3) shows how many the global best of a swarm reduces during an iteration. Thus, value of R must be calculated at the end of every iteration. If the global best reduction rate is low, then the swarm must be reinitialized. The condition of the swarm reinitialization is:

$$AR \geq R, \tag{1.4}$$

where AR – is acceptable rate of the global best reduction (this value must be set by ME-PSO user).

The value of the AR must be set in view of the recommendations: big value of AR causes frequent reinitialization of the swarm and, as a result, during many iterations particles move without improving of the global best; small value of AR leads to jamming of the particles in local minima and stagnation of the swarm (note, if the AR=0 ME-PSO reduces to canonical PSO). The issue of assigning AR value is still open for discussions and it is necessary to investigate it in the further studies. In the following we set AR=0.01. Such value of AR provides quite good balance between function exploration and reinitialization. Other criteria, which may be used for the swarm reinitialization, are presented in the Table 1. They should be checked at the end of each iteration as well. In the article we use only (1.1) and (1.2) expressions as condition of the swarm reinitialization. Note, all or just only one of the described criteria (Table 1.1) may be used for this purpose.

In order to provide the high search ability of particles we propose to eliminate their inertial feature. It requires to set inertia coefficient equal to zero w=0. It causes rapid movement of particles on a function surface and, as a result, the bigger area of the function domain may be explored.

Table 1.1 Possible criteria which may be used as condition of a swarm reinitialaization

Formula	Description
$GB_i = GB_{i-q}, \quad q = \overline{(1,\ Q)}$, where *Q* – is the number of the iterations which must be set before run of the algorithm.	If the value of the global best has not been changed during *Q* iterations, it may mean that the particles had trapped in local minima and they cannot leave it. The number *Q* may be set as a fraction of the total number of iterations *N*. For instance, *Q*=*N*(0.01…0.10).
$\frac{1}{E}\sum_{e=1}^{E} LB_i^e \approx (0.99...\ 0.90) GB_i$, where *E* – is some quantity of particles which is lesser than the swarm population; *LB* – local bests of the particles. Subscript means the *i*-th iteration, superscript means a number of the particle.	If the *E* particles are close to the best particle in a swarm that mean they have a little chance to find the minimum that would be better than the current global best. In order to avoid swarm stagnation about the global best it is necessary to reinitialize the swarm. The number E must be set as an integer, for example, *E*=*SP*(0.1…0.5), where *SP* – is the swarm population.

The novel ME-PSO technique can be clarified within support of following pseudocode:

```
    Set the parameters $c_1$, $c_2$, $SP$, stop criterion (number of iterations, cost function
    value, etc.) and $AR$;
    Initialize particles positions and velocities;
    Calculate global best;
Do
    Update the particles positions and velocities;
    For each particle check the excess search domain condition;
    Calculate personal bests and global best;
    Calculate $R$;
If $R \leq AR$ then
    Reinitialize particles positions and velocities;
Until stop condition is met.
```

During initialization and reinitialization, all the components of particles' positions should be set as a random number in the search domain and all the components of a particles' velocity should be set equal to zero. Described approach allows to use computation resources more efficiently. Further study makes possible to establish how ME-PSO copes with the different optimization problems.

1.1.2 Numerical experiments

In order to show advantages of the ME-PSO the numerical experiments have been performed. We choose ten benchmark functions: uni- and multimodal (Table 1.2). All of the chosen functions have different topology features but each function has global minima which are equal to zero.

In order to establish how benchmark functions dimensions (D) influences to ME-PSO performance, experiments were carried out for numbers of D: 10, 30, 50 and 200 (for the last experimental series). As indicators of the algorithms efficiency average and median values were used. Standard deviation SD indicates dispersion of the reached minima in relation to average values. All calculations were carried out for PSO, IA-PSO [16] and ME-PSO techniques. Comparison different approaches, which are implemented in the ME-PSO and IA-PSO algorithms, will give the information about the efficiency of overcoming the premature convergence.

In all experiments the number of iterations N as a stop criterion was used.

Table 1.2 Ten benchmark function for numerical experiment

Benchmark function	Formula	Search domain
1	2	3
Spherical	$f1=\sum_{i=1}^{D} x_i^2$	-20≤x_i≤20
Elliptical	$f2=\sum_{i=1}^{D}(10^6)^{\frac{i-1}{D-1}} x_i^2$	-2≤ x_i ≤2
Schwefel №1	$f3=\sum_{i=1}^{D}\left(\sum_{j=1}^{i} x_j\right)^2$	-10≤ x_i ≤10
Rosenbrock	$f4=\sum_{i=1}^{D-1}\left(100(x_{i+1}-x_i^2)^2+(1-x_i)^2\right)$	-10≤ x_i ≤10
Rastrigin	$f5=\sum_{i=1}^{D}\left(x_i^2-10\cos(2\pi x_i)+10\right)$	-5≤ x_i ≤5
Griewank	$f6=4000^{-1}\sum_{i=1}^{D} x_i^2-\prod_{i=1}^{D} x_i i^{-0.5}+1$	-100≤ x_i ≤100
Alpine	$f7=\sum_{i=1}^{D}\lvert x_i\sin(x_i)+0.1x_i\rvert$	-10≤ x_i ≤10
Schwefel №2	$f8=D^{-1}\sum_{i=1}^{D}\left(x_i\sin(\lvert x_i\rvert^{-0.5})\right)+418.983$	-500≤ x_i ≤500

Table 1.2 continuation

1	2	3
Ackley	$f9=-20\exp\left(-0{,}2\left(D^{-1}\sum_{i=1}^{D}x_i^2\right)^{0.5}\right)-$ $-\exp\left(D^{-1}\sum_{i=1}^{D}\cos(2\pi x_i)\right)+20+e$	-30≤ x_i ≤30
Weierstrass	$f10=D^{-1}\sum_{i=1}^{D}\sum_{k=0}^{20}\left(0.5^k\cos(2\pi 3^k(x_i+0.5))\right)-$ $-\sum_{k=0}^{20}\left(0.5^k\cos(\pi 3^k)\right)$	-0.5≤ x_i ≤0.5

In order to obtained proper statistical results each numerical experiment has run 100 times. In each run the particles' positions were random. Parameters of the swarm were the same for all experiments (Table 1.3).

Table 1.3 Swarm parameters for all numerical experiments

Parameters of the swarm	Value
c_1	1.19
c_2	1.19
w	0.72
swarm population	30
connection topology	full

1.1.3 Results and discussion

Results of the first series of experiments (N=250) allow to determine the algorithms' performance at early stages of the exploration (Table 1.4).

Table 1.4 Results of the first experimental series

Functions	PSO			IA-PSO			ME-PSO		
	Average	Median	SD	Average	Median	SD	Average	Median	SD
1	2	3	4	5	6	7	8	9	10
D=10									
*f*1	1,11E-17	2,26E-18	2,78E-17	4,09E-30	4,89E-32	1,69E-29	1,02E-07	4,02E-08	1,70E-07
*f*2	8,69E01	1,30E-10	8,62E02	1,11E-30	3,98E-33	1,01E-29	5,00E-01	4,26E-01	4,52E-01
*f*3	2,14E-06	1,69E-08	9,85E-06	7,21E-11	1,75E-14	4,32E-10	2,20E-02	1,42E-2	2,44E-02
*f*4	1,16E01	4,59E00	3,37E01	7,35E00	7,12E00	9,39E-01	2,04E01	7,63E00	3,10E01
*f*5	1,42E01	1,29E01	6,42E00	1,58E01	1,47E01	6,77E00	5,68E00	5,05E00	2,27E00
*f*6	1,01E-01	8,73E-02	6,53E-02	2,57E-01	2,39E-01	1,41E-01	1,69E-01	1,43E-01	1,00E-01
*f*7	2,41E-03	1,67E-05	1,57E-03	1,09E00	7,33E-01	1,07E00	3,37E-03	9,07E-04	7,58E-03
*f*8	4,03E02	4,03E02	1,86E00	4,10E02	4,10E02	1,13E00	4,13E02	4,12E02	3,71E-01
*f*9	3,00E-01	4,73E-09	5,98E-01	1,60E00	2,04E00	1,33E00	2,05E-03	6,58E-04	7,25E-03
*f*10	8,40E-02	4,47E-02	9,51E-02	7,90E-03	1,40E-16	3,89E-02	8,30E-03	1,47E-03	2,31E-02
D=30									
*f*1	1,87E00	4,08E-01	4,34E00	5,19E-08	8,25E-09	3,37E-07	7,94E-02	6,50E-02	5,23E-02
*f*2	7,86E02	3,35E02	1,20E02	4,35E-11	5,53E-12	1,35E-10	8,44E01	8,14E01	3,74E01

Table 1.4 continuation

1	2	3	4	5	6	7	8	9	10
*f*3	3,89E01	3,40E01	2,37E01	3,62E-01	2,54E-01	3,42E-01	1,46E01	1,44E01	4,10E00
*f*4	4,78E02	2,37E02	6,84E02	2,83E01	2,86E01	1,14E00	1,47E02	1,36E02	7,52E01
*f*5	9,73E01	9,84E01	2,25E01	1,11E02	1,09E02	2,64E01	5,71E01	5,77E01	1,24E01
*f*6	3,67E-01	3,15E-01	2,58E-01	3,36E-02	5,18E-08	3,88E-02	3,05E-01	2,35E-01	2,03E-01
*f*7	1,89E00	1,67E00	1,25E00	1,23E01	1,24E01	3,73E00	8,04E-01	6,49E-01	5,37E-01
*f*8	4,03E02	4,03E02	2,01E00	4,13E02	4,13E02	1,38E00	4,10E02	4,10E02	3,09E-01
*f*9	6,73E00	6,45E00	1,93E00	3,18E00	3,47E00	1,01E00	2,10E00	2,10E00	5,26E-01
*f*10	5,08E-01	5,04E-01	9,92E-02	5,02E03	2,07E04	4,36E-02	2,09E-01	2,07E-01	6,06E-02
D=50									
*f*1	7,08E01	5,98E01	4,63E-01	2,01E-04	1,76E-04	1,42E-04	2,28E00	2,17E00	7,99E-01
*f*2	7,77E03	4,99E03	8,73E03	7,58E-06	2,88E-06	3,03E-05	6,16E02	6,01E02	2,15E02
*f*3	7,08E02	6,47E02	2,79E02	1,18E01	1,02E01	6,50E00	7,25E01	7,22E01	1,73E01
*f*4	2,00E04	1,96E04	5,75E02	4,85E01	4,85E01	5,18E-01	4,72E02	4,47E02	1,52E02
*f*5	3,46E02	3,52E02	6,75E01	2,49E02	2,46E02	4,31E01	1,31E02	1,28E02	2,55E01
*f*6	1,43E00	1,39E00	2,37E-01	5,63E-02	1,69E-04	7,52E-02	1,04E00	1,04E00	2,02E-02
*f*7	1,16E01	1,16E01	3,95E00	2,79E01	2,77E01	6,33E00	5,51E00	5,12E00	2,09E00

Table 1.4 continuation

1	2	3	4	5	6	7	8	9	10
*f*8	4,05E02	4,05E02	1,05E-01	4,16E02	4,16E02	5,00E-01	4,15E02	4,15E02	7,01E-02
*f*9	1,70E01	1,11E01	3,84E00	3,21E00	3,90E00	1,47E00	3,77E00	3,71E00	4,69E-01
*f*10	7,01E0-1	6,97E0-1	8,45E-02	7,87E-03	4,07E-03	1,21E-02	3,51E-01	3,58E-01	5,69E-01

Table 1.5 Results of the second experimental series

Functions	PSO			IA-PSO			ME-PSO		
	Average	Median	SD	Average	Median	SD	Average	Median	SD
1	2	3	4	5	6	7	8	9	10
D=10									
*f*4	5,18E00	4,33E00	6,56E00	6,43E00	6,22E00	1,13E00	2,18E00	1,84E00	1,89E00
*f*5	1,14E01	1,04E01	5,54E00	7,84E00	7,33E00	4,55E00	8,55E-37	1,97E-47	5,48E-36
*f*8	4,15E02	4,15E02	2,19E-01	4,15E02	4,15E02	2,11E-01	4,03E02	4,03E02	1,33E00
D=30									
*f*4	2,08E02	1,71E02	1,56E02	2,78E01	2,79E01	1,00E00	5,06E01	2,72E01	3,53E01

Table 1.5 continuation

1	2	3	4	5	6	7	8	9	10
*f*5	5,13E01	4,93E01	1,77E01	4,62E01	4,64E01	1,58E01	9,58E00	9,95E00	2,45E00
*f*8	4,03E02	4,03E02	4,32E-01	4,16E02	4,16E02	2,34E-01	4,15E02	4,15E02	1,61E-11
*f*9	3,47E00	3,28E00	9,32E-01	2,58E00	2,94E00	1,03E00	6,39E-12	4,62E-12	6,28E-12
D=50									
*f*3	2,19E02	2,16E02	6,35E01	3,73E-11	1,06E-14	3,14E-10	2,29E00	2,20E00	6,37E-01
*f*4	5,35E03	3,37E03	5,16E03	4,79E01	4,85E01	9,86E-01	9,02E01	9,48E01	4,08E01
*f*5	1,94E02	1,92E02	3,94E01	2,50E02	2,54E02	4,30E01	4,30E01	4,31E01	7,91E00
*f*7	1,90E01	1,89E01	5,82E00	1,07E01	1,03E01	4,50E00	2,59E-02	1,80E-02	2,57E-02
*f*8	4,15E02	4,15E02	2,25E-01	4,16E02	4,16E02	1,87E-01	4,15E02	4,15E02	1,07E-02
*f*9	1,16E01	1,17E01	1,62E00	2,00E00	2,00E00	1,86E-03	6,78E-08	5,36E-08	4,65E-08

In Table 1.4 and further tables the best values of average and median are in bold.

Figures in Table 1.4 show that on early stage of exploration ME-PSO has reached not good minima values for almost all benchmark functions. The only one exception is the function *f*5.

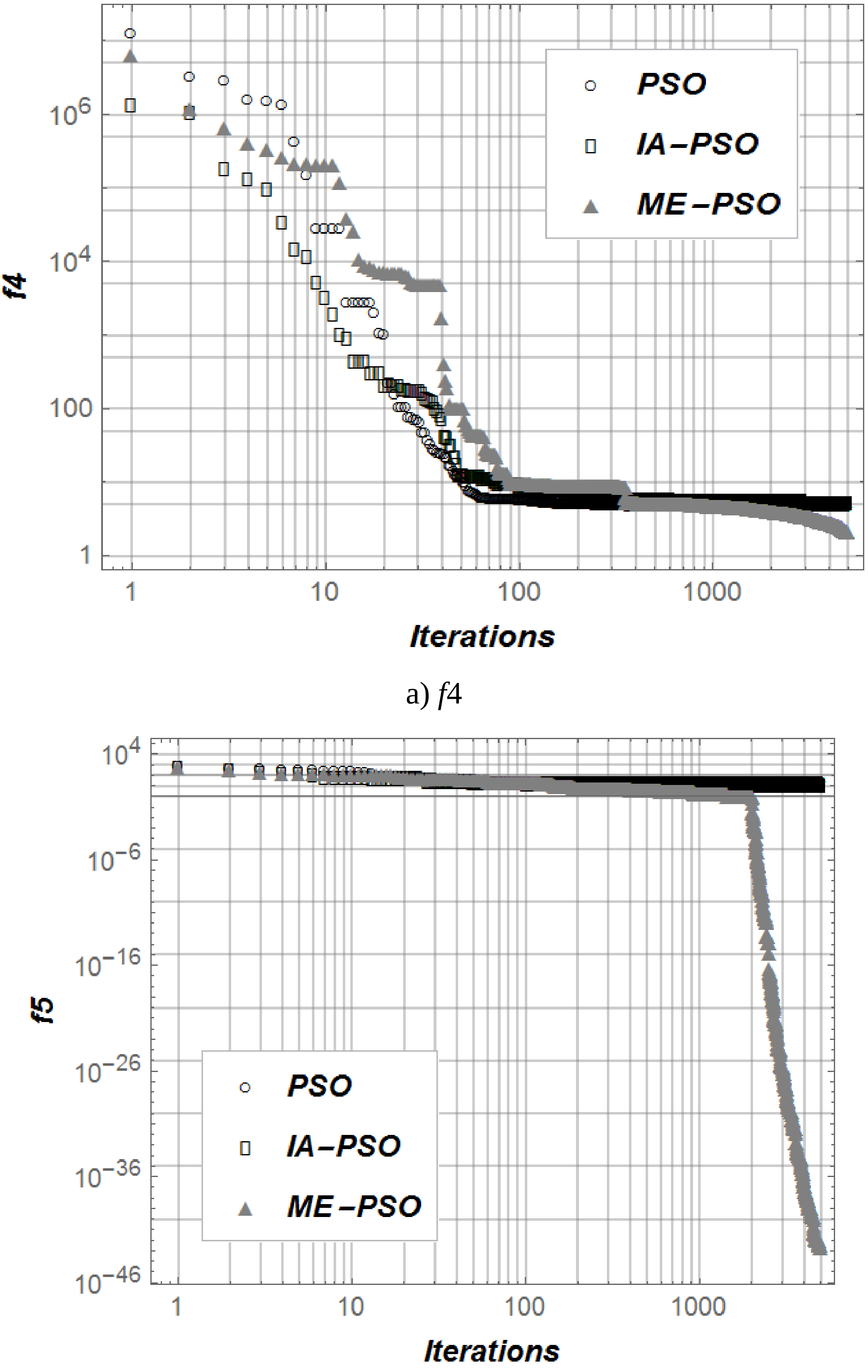
The best performance in this experimental series relates to IA-PSO technique. For *D*=10 it allows to find local minima which are very close (in topology sense) to the global minimum of the functions *f*1, *f*2, *f*3, *f*10. The worst results ME-PSO has shown for the functions: *f*4 and *f*8. It is caused by small number of iterations.

Thus, there is a need to study how ME-PSO works with big number of *N*. It was the purpose of the second series of experiment in which *N*=5000 (Table 1.5). The calculations were carried out for the most difficult functions to minimize. Comparison of data in Table 1.4 and Table 1.5 supports the statement that an increasing of the iteration number *N* makes it possible to reduce the average and median values of reached minima.

Moreover, for some cases ME-PSO has reached almost computer zero (function *f*5 with *D*=10). Almost all results that are related to ME-PSO are better than those that were obtained with PSO and IA-PSO techniques.

The exceptions are functions *f*3, *f*4 and *f*8. IA-PSO finds good local minimum of the *f*3 (with *D*=50) quite rapid. However, complicated topology of function *f*4 causes the premature convergence both PSO and IA-PSO, especially for *D*=50. In order to compare PSO, IA-PSO and ME-PSO performances the graphs have been built (Fig. 1.3 and Fig. 1.4).

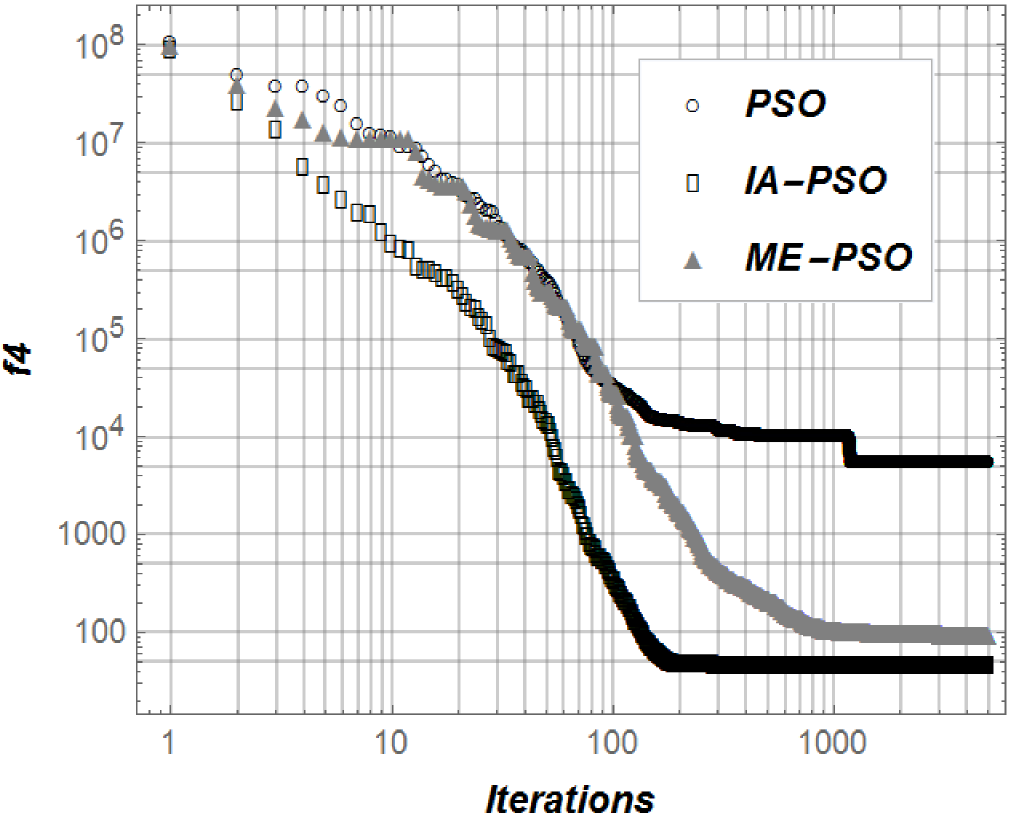
Vertical and horizontal axes of the following graphs are presented in logarithmic scale. The graphs on Fig. 1.4 (b, c, d) make it obvious that ME-PSO has no premature convergence: the algorithm execution provides reduction of the global best during all iterations.



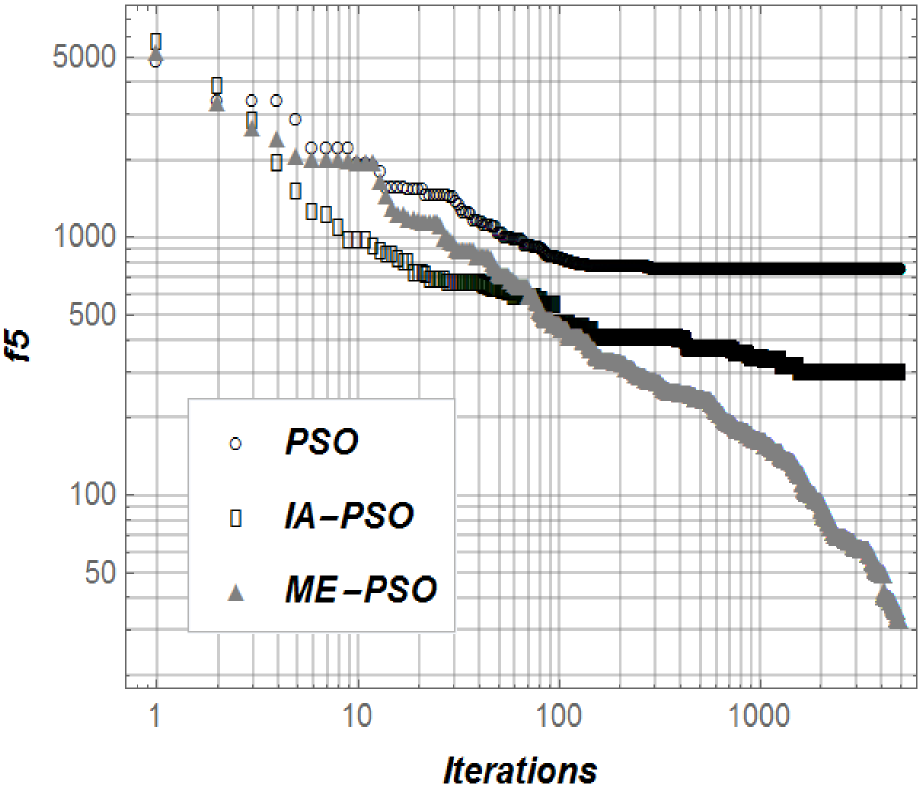
a) *f*4

b) *f*5

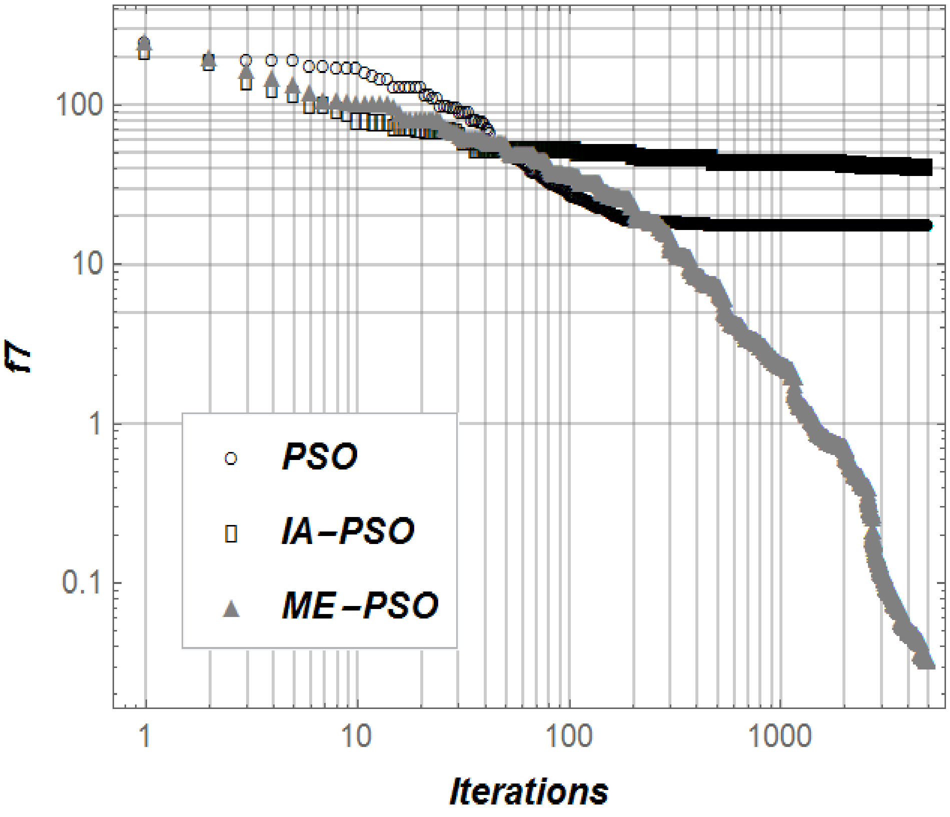
Fig. 1.3 PSO, IA-PSO and ME-PSO performances during minimizing of the benchmark functions *f*4 and *f*5 (*D*=10)


10^8
10^7
10^6
10^5
10^4
1000
100
f4
1
10
100
1000
Iterations
○ PSO
▯ IA–PSO
▲ ME–PSO

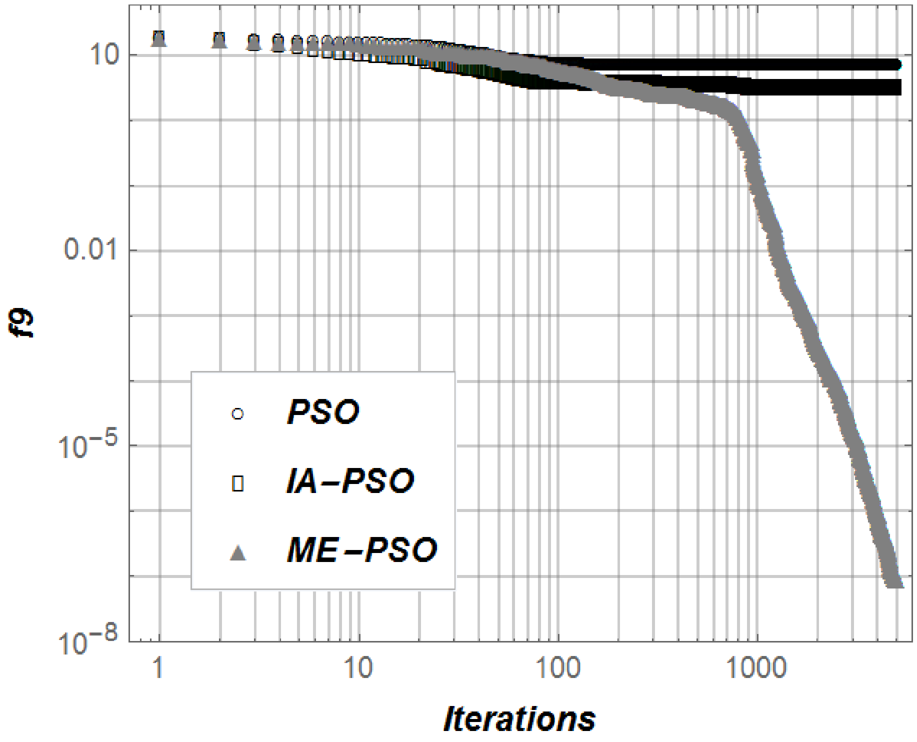
a) *f*4


5000
1000
500
100
50
f5
1
10
100
1000
Iterations
○ PSO
▯ IA–PSO
▲ ME–PSO

b) *f*5



c) *f*7



d) *f*9

Fig. 1.4 PSO, IA-PSO and ME-PSO performances during minimizing of the benchmark functions *f*4, *f*5, *f*7 and *f*9 (*D*=50)

This is the biggest difference between ME-PSO and algorithms PSO and IA-PSO. Graphs on the Fig. 1.3 and Fig. 1.4 clearly show that PSO and IA-PSO converge rather quickly. In contrast, ME-PSO continues to minimize almost all benchmark functions during all iterations.

In order to support that suggestion, the third series of experiment was performed. All the calculations were carried out with the $f4$ and $f5$ functions with D=50.

We choose f4 because IA-PSO has better than ME-PSO performance for its minimization. In order to investigate the impact of iterations number N on the IA-PSO and ME-PSO performances we set N=50000.

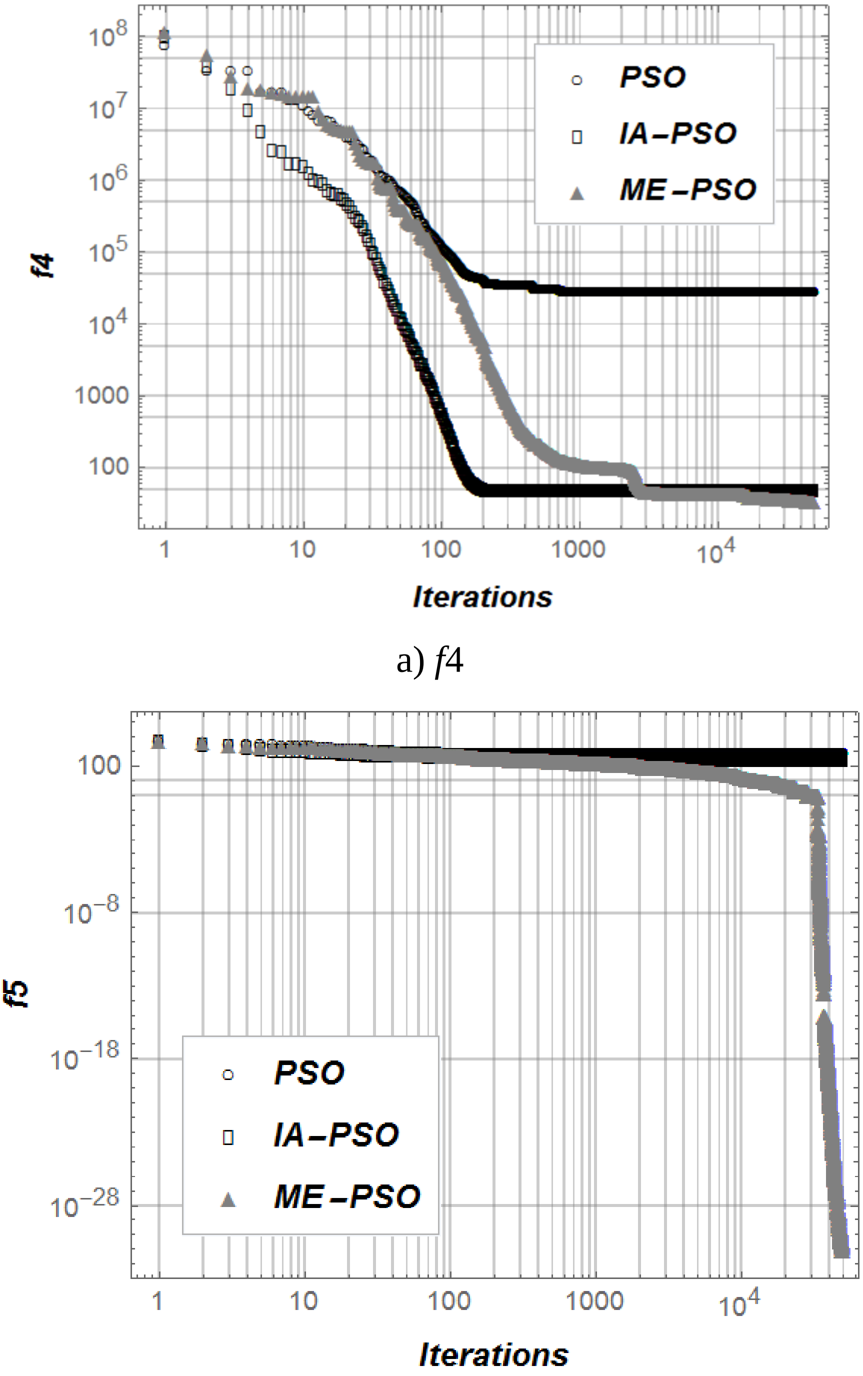
The choice of $f5$ for third experimental series is caused by the fact of bad efficiency of ME-PSO for that function on the previous experimental series. All the figures were set to the Table 1.6. Obtained results show that advantages of ME-PSO (especially for complicated functions) are revealed at big number of iterations. The graphs on Fig. 1.5 show the bigger N the better ME-PSO performance.

Table 1.6 Results of the third experimental series

Function	PSO			IA-PSO			ME-PSO		
	Average	Median	SD	Average	Median	SD	Average	Median	SD
*f*4	1,98E04	1,94E04	5,61E02	4,72E01	4,88E01	8,99E-01	3,62E01	3,79E01	2,02E00
*f*5	3,44E02	3,43E02	6,66E01	2,44E02	2,51E02	5,301E01	3,34E-35	2,50E-36	8,32E-37

The most difficult benchmark function for all algorithms is f8. None of them have found a good solution. At early stages of the exploration these algorithms find bad local minima of $f8$. Even reinitialization of a swarm does not solve the problem: all the particles in a new epoch swarm have a great tendency to move toward previous global best. They have no time for proper exploration of the $f8$. This ME-PSO weakness (only for some of the complicated optimization problems) causes the necessity for further improving of the proposed algorithm.

One of the possible ways to solve that problem is varying parameter *AR* during optimization process. For instance, *AR* can be a function of the current global best or current iteration. That issue is the matter for further investigations.



a) *f*4

b) *f*5

Fig. 1.5 PSO, IA-PSO and ME-PSO performances during minimizing of the benchmark functions *f*4 and *f*5 (*D*=50)

Some optimization algorithms fail when the dimension of the cost function is more than 100. That is why the fourth series of experiment was conducted under condition *D*=200. Two benchmark functions (*f*4 and *f*7) were chosen for that series. They have different features: the first one is unimodal and the second one is multimodal. For these high-dimensional problems we set *N*=50000. All the obtained figures are in Table 1.7.

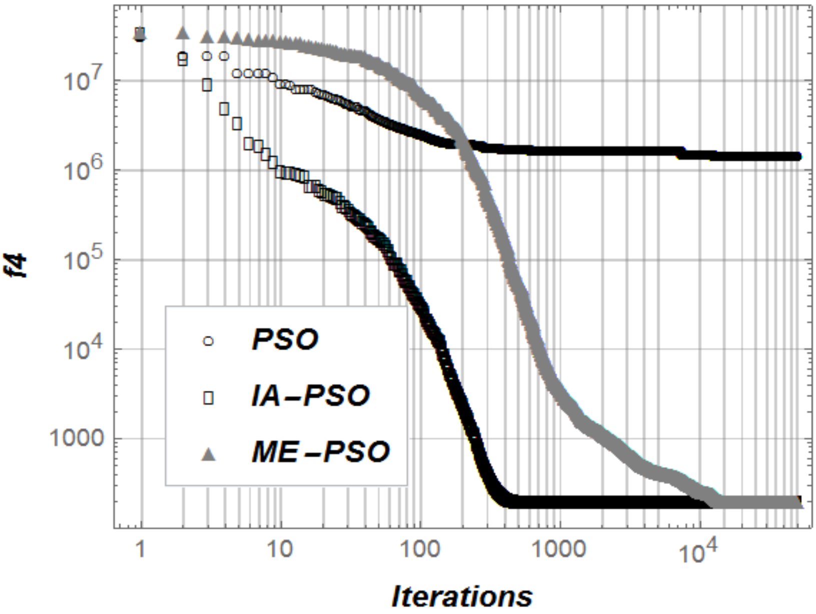
Table 1.7 Results of the fourth experimental series

Function	PSO			IA-PSO			ME-PSO		
	Average	Median	SD	Average	Median	SD	Average	Median	SD
*f*4	1.46E06	1.42E06	2.09E04	1.98E02	1.97E02	7.90E00	1.96E02	1.94E02	5.04E00
*f*7	1.92E02	1.83E02	4.90E01	7.90E01	7.33E01	1.01E01	1.38E01	1.33E01	4.22E00

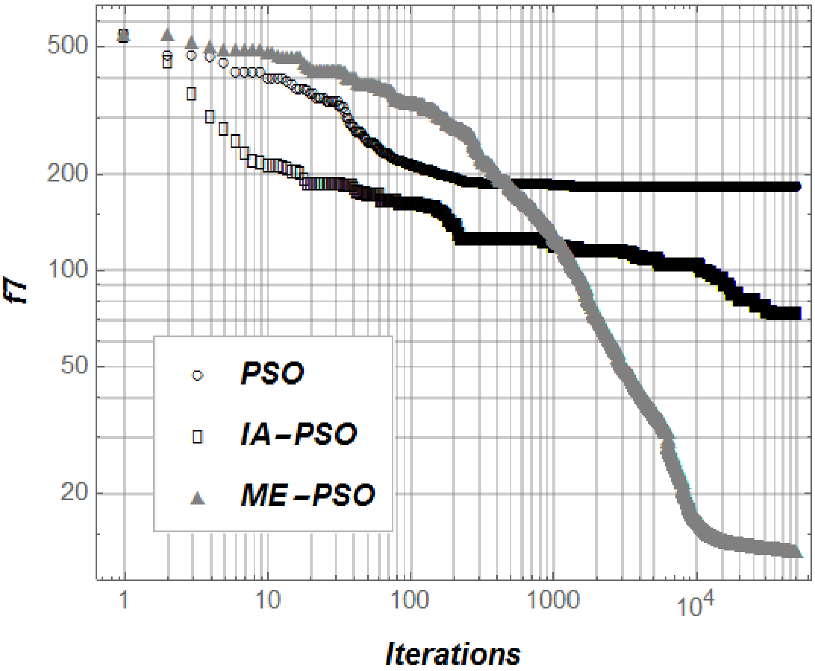
Graphs which related to PSO, IA-PSO and ME-PSO performances for the high-dimensional optimization problems are shown on fig 1.6.

Data in table 1.7 and graphs on fig. 1.6 clearly prove the superiority of ME-PSO. Although for *f*4 the difference between IA-PSO and ME-PSO is slight.

Fig. 1.6 (a) shows the convergence of all algorithms. It is an obstacle for further function minimization.



a) *f*4



b) *f*7

Fig. 1.6 PSO, IA-PSO and ME-PSO performances during minimizing of the benchmark functions *f*4 and *f*7 (*D*=200)

In order to prevent it further improvement of ME-PSO should be carried out. The ultimate goal is to find a PSO modification which is invariant to problem dimensionality and has high exploration abilities.

1.2 Engineering management of ME-D-PSO

One of the most powerful methods for function minimization is known as PSO (particle swarm optimization) proposed by Kennedy and Eberhart [2]. Since then there were hundreds of modifications of the canonical PSO-algorithm. Some of them have quite good searching abilities and allow their applying for a wide range of the real-world problems [23]. One of such problems is optimal control, particularly tuning of PI or PID-controllers for: quadrotors [24], active magnetic bearing suspends [25], variable frequency brushless synchronous generator [26], bioreactors [27], overhead crane [28], energy conversion system [29] and many others. Almost all of these works involve following criteria to minimize: Integral Absolute Error (IAE) [24], Integral of Time-Absolute Error multiplication (ITAE) [25], overshoot [26]; [28, 29], Integral Square Error (ISE) [27], rise time and settling time [28]; [29], steady-state error [28]. However, other ones were ignored: maximal control, Integral Square Control (ISC), Integral Time-Absolute Control (ITAC), etc. Researches in that area are continuing.

The researches aim is to adjust the controller in such a manner, that the most important criteria reduced to a minimum (or, at least, decreased as much as possible) and constrains are met. Nevertheless, constrains are very infrequent in the optimal tuning problems statements.

The efficiency of the problem solving depends on the features of the chosen optimization method (algorithm). For PSO-based class of algorithms, their features depend heavily on their parameters. Appropriate adjusting of the parameters provides good algorithm performance.

Another way to improve the PSO-based class of algorithms – is enhancing with different mechanisms of particles' search, swarm reinitialization, local minima avoiding, etc. Such approaches involve additional lines of an algorithm's code and, sometimes, additional parameters.

It is a quite difficult task to classify all of the known PSO-based techniques. We should mention the work [30] where were presented all the most important directions in PSO methods evolution. Since then, hundreds have been developed.

One of the approaches to swarm searching abilities improving is to increase its diversity [31-34]. Base on the definition of the swarm population diversity given by Shi, Eberhart and Zhan [35-37]; authors of the work [31] divided it into three parts: position diversity, velocity diversity, and cognitive diversity. All of these definitions mean some norm value (for example, position diversity means the farness of particles positions from swarm's average point). The analysis of these diversities is presented in the work [32] and it has been found that the method based on the current position and the average of current velocities has the best performance. In the article [33] position diversity governs the switching between attractive and repulsive phases of the swarm in ARPSO method. In scientific work [34] has been briefly analyzed the PSO-base techniques with a diversity mechanism as well as presented a novel diversity enhanced mechanism, which is similar to the crossover operation of Differential Evolution [38]. It involves initializing of some trial particles and exploits greedy mechanism for their selection. Work [34] gives a different definition of diversity: with the intention of usefulness in quantifying swarm exploration and exploitation. Thus, diversity may be exploited in different manners.

In contrast to mentioned investigations, in the current article diversity denote another feature: the variety of the particles' movements.

In the article 39 a reinitializing mechanism for the PSO-algorithm (ME-PSO) has been developed. The main idea was to run a new swarm after the previous one had shown a low rate of minimized function reducing. That mechanism prevents the algorithm from premature convergence and improves its chances to continue the searching activity. In the work [40] ME-PSO method has been used for optimization of PI-controllers tuning.

However, many areas of the control issues, particularly in the automation, are needed powerful techniques for synthesis of the optimal closed-loop systems of control. That statement is especially important in the cases of different contractions being placed on the control function or phase coordinates. The problem to solve may be complicated with non-linearities of the contractions, isolation and/or farness of the admissible search domains, distortion of the cost function, etc. In the current article, the PSO-based technique is developed, that might overcome such difficulties.

The article is arranged in the following manner: the describing of the canonical PSO-algorithm is given in the first part of the third section. The second one contains the explanation of the new modified PSO-based method. The fourth section provides a description of the numerical experiments and gives raw data for further analysis. The fifth one is dedicated to the estimation of the compared algorithms efficiency based on the worked-out two criteria. The algorithm which has shown the best efficiency is used for optimal tuning of PI-controllers with constraints in the sixth section. That part of the article also includes analysis of the new PSO-based algorithm applying to the problem of constrained optimization of the PI-controller tuning. Some suggestions and conclusions complete the article.

1.2.1 ME-D-PSO description

First of all, we will describe ME-PSO algorithm. As was mentioned earlier the main idea of that method is monitoring the algorithm efficiency during its execution. If a swarm tends to stagnation, which means that it has trapped to the bad minimum and there is a low possibility to leave it in the further iterations, the efficiency of the algorithm is low. The criterion, which reflects the swarm's stagnation has been presented (1.3)-(1.4). For the very first iteration, the global best of the new (reinitialized) swarm is it was for the previous swarm.

Taking into consideration quite a positive effect of the reinitializing mechanism in ME-PSO method [39] we will equip the new PSO modification with it.

In order to improve search ability of the new PSO modification, it was enhanced with the diversity mechanism. The term „diversity" means the different features of the particles, which provide a different manner of their movements. From iteration to iteration the movement patterns of the particles are changing. It, in turn, improves the chances of finding the good (or even global) minimum of the cost function. In the frame of the research, the diversity mechanism is provided with a permanent variation of the coefficients w, c_1, and c_2. We have used a stochastic technique for their changing. In particular coefficients w, c_1, and c_2 were pseudorandom numbers in some ranges: $w_{\min}$ and $w_{\max}$ – for w; $c_{1\min}$ and $c_{1\max}$ – for c_1; $c_{2\min}$ and $c_{2\max}$ – for c_2. From iteration to iteration coefficients w, c_1, and c_2 are changing. However, the more general case of the algorithm allows some slowing-down in the coefficients w, c_1, c_2 changes: variations may occur not on each of the iteration. Such a case was not investigated in the current research. It is the subject of further study.

Thus, the reinitialization prevents the algorithm's premature convergence, and diversity of the particles in the swarm provides good search ability. The algorithm which contains mentioned techniques was called ME-D-PSO.

1.2.2 Numerical experiments

In order to show the performance of the ME-D-PSO method, numerical experiments should be carried out. The standard approach for such a purpose is to determine the performance indicators of compared algorithms for a range of different benchmark cost functions. In the frame of the research, we have used a set the following benchmarks (Table 1.8).

All the given benchmarks have the global minimum, which is equal to zero.

The benchmark functions have different topological features. Such a variety of benchmarks is dictated by the requirement to obtain to the extent possible general results. Indeed, the algorithm, which has shown good performance on the set of the benchmarks (Table 1.8), potentially will allow finding the global (or, at least, good local) minimum of a complicated real-world optimization problem.

One of the most important features of the benchmark is its dimensionality. It influences the complication (in terms of optimization problem) of the cost functions. The high dimensionality of a cost function is almost always connected with its strong complexity.

For multimodal functions increasing dimensions almost always leads to appearing of additional local minima. That is why we have used a quite big number of dimensions for all benchmarks D=50.

The swarm population P for all algorithms is equal to 50. The number of iterations J is equal to 1000 for each algorithm's run.

Table 1.8 The set of the benchmarks and some of their features

Benchmark function	Formula	Search domain	Location of the global minimum x_i	Separa-bility	Unimo-dality
1	2	3	4	5	6
Spherical	$f1=\sum_{i=1}^{D}x_i^2$	$-20\le x_i\le 20$	0	+	+
Rosenbrock	$f2=\sum_{i=1}^{D-1}\left(100(x_{i+1}-x_i^2)^2+(1-x_i)^2\right)$	$-10\le x_i\le 10$	1	-	+
Rastrigin	$f3=\sum_{i=1}^{D}\left(x_i^2-10\cos(2\pi x_i)+10\right)$	$-5\le x_i\le 5$	0	+	-
Griewank	$f4=4000^{-1}\sum_{i=1}^{D}x_i^2-\prod_{i=1}^{D}x_i i^{-0.5}+1$	$-100\le x_i\le 100$	0	-	-
Alpine	$f5=\sum_{i=1}^{D}\lvert x_i\sin(x_i)+0.1x_i\rvert$	$-10\le x_i\le 10$	0	-	-
Schwefel №2	$f6=D^{-1}\sum_{i=1}^{D}\left(x_i\sin(\lvert x_i\rvert^{-0.5})\right)+418.983$	$-500\le x_i\le 500$	420.9687	-	-

Table 1.8 continuation

1	2	3	4	5	6
Ackley	$f7=-20\exp\left(-0{,}2\left(D^{-1}\sum_{i=1}^{D}x_i^2\right)^{0.5}\right)-$ $-\exp\left(D^{-1}\sum_{i=1}^{D}\cos(2\pi x_i)\right)+20+e$	-30≤x_i≤30	0	+	-
Weierstrass	$f8=D^{-1}\sum_{i=1}^{D}\sum_{k=0}^{20}\left(0.5^k\ \cos(2\pi 3^k\,(x_i+0.5))\right)-$ $-\sum_{k=0}^{20}\left(0.5^k\ \cos(\pi 3^k)\right)$	-0.5≤x_i≤0.5	0	+	-
Schwefel 2.20	$f9=\sum_{i=1}^{D}\lvert x_i\rvert$	-10≤x_i≤10	0	+	+
Qing	$f10=\sum_{i=1}^{D}\left(x_i^2-i\right)^2$	-500≤x_i≤500	$\pm i^{0.5}$	-	-
Salomon	$f11=1-\cos\left(2\pi\left(\sum_{i=1}^{D}x_i^2\right)^{0.5}\right)+0.1\left(\sum_{i=1}^{D}x_i^2\right)^{0.5}$	-100≤x_i≤100	0	-	-
Xin-She Yang №2	$f12=\sum_{i=1}^{D}\lvert x_i\rvert\cdot\exp\left(-\sum_{i=1}^{D}\sin\left(x_i^2\right)\right)$	-2π≤x_i≤2π	0	-	-

In order to obtain statistically-valid results for each benchmark function, all the algorithms have been run 50 times.

In each run the particles' positions were random.

The comparable algorithms were: LDW-PSO [41], IA-PSO [18], Ring-PSO [42], ME-PSO [39], ME-D-PSO. Their parameters are given in Table 1.9.

Table 1.9 Parameters of PSO-based algorithms

Parameters of the swarm	Algorithm				
	LDW-PSO	IA-PSO	Ring-PSO	ME-PSO	ME-D-PSO
c_1	1.19	1.19	1.19	0.10	var
c_2	1.19	1.19	1.19	2.10	var
w	var	var	0.72	0	var
connection topology	full	full	non-full	full	full

The notations „var" for different algorithms given in Table 1.9 are explained below:

1) for LDW-PSO weight coefficient according to the work [41] is described as follows:

$$w = (w_{\min} - w_{\max})\frac{J-j}{J} + w_{\min}, \tag{1.5}$$

where $w_{\min}$ and $w_{\max}$ – minimum and maximum limits of the weight coefficient domain ($w_{\min}$=0.4, $w_{\max}$=0.9);

2) for IA-PSO weight coefficients are different for different particles. They are changing during algorithm execution [18] with accordance to the following formula:

$$w_p = rand(0.5,\ 1)\cdot\left(1 - \frac{\left\|g_j - \mathrm{x}_{jp}\right\|}{\max\limits_p(\left\|g_j - \mathrm{x}_{jp}\right\|)}\right),\quad p \in (\overline{1,\ P}), \tag{1.6}$$

where w_p – the weight coefficient of the p-th particle; P – swarm population; g_j – global best, which has been found during 1...j iterations; x_{jp} – the position vector of the p-th particle on the j-th iteration;

3) for ME-D-PSO the variation of the coefficients w, c_1, and c_2 has been explained in the following subsection. Now we set the domain limits: $c_{1\min}=c_{2\min}=w_{\min}=0$; $c_{1\max}=c_{2\max}=2$, $w_{\max}=1$.

The obtained results have been estimated with the set of statistical indicators: maximum (the worst), minimum (the best), average, median, and standard deviation (*SD*) values.

All the figures which illustrate the obtained results are given in Table 1.10 (the best figures are in bold).

Table 1.10 Numerical experiments outcome

Functions	Indicators	Algorithms				
		LDW-PSO	IA-PSO	Ring-PSO	ME-PSO	ME-D-PSO
1	2	3	4	5	6	7
*f*1	Max	1.979E+00	**7.943E-26**	5.807E-07	1.293E-06	4.785E+03
	Min	4.258E-02	**9.352E-30**	9.815E-08	3.497E-08	1.092E-11
	Average	5.184E-01	**8.519E-27**	2.664E-07	1.903E-07	9.569E+01
	Median	3.643E-01	**3.360E-27**	2.584E-07	1.379E-07	1.581E-09
	SD	4.395E-01	1.402E-26	1.063E-07	2.023E-07	6.766E+02
*f*2	Max	1.236E+03	**5.090E+01**	1.855E+02	7.023E+06	6.230E+06
	Min	1.146E+02	4.606E+01	**3.924E+01**	4.362E+04	4.259E+01
	Average	3.282E+02	**4.807E+01**	8.676E+01	8.135E+05	2.372E+05
	Median	2.650E+02	**4.843E+01**	8.600E+01	1.583E+02	1.534E+02
	SD	2.244E+02	9.105E-01	4.217E+01	2.058E+06	1.175E+06
*f*3	Max	2.736E+02	2.802E+02	**1.897E+02**	8.273E+02	7.304E+02
	Min	1.221E+02	1.048E+02	1.010E+02	2.405E+02	**7.578E+01**
	Average	2.030E+02	1.807E+02	**1.467E+02**	4.072E+02	1.603E+02
	Median	2.054E+02	1.767E+02	1.485E+02	3.537E+02	**1.356E+02**
	SD	3.739E+01	3.506E+01	2.462E+01	1.642E+02	1.214E+02

Table 1.10 continuation

1	2	3	4	5	6	7
*f*4	Max	1.005E+00	3.487E-02	**1.232E-02**	3.428E+01	3.801E-02
	Min	3.269E-02	**1.324E-32**	9.401E-08	9.931E-09	1.739E-10
	Average	2.514E-01	8.503E-03	5.720E-04	1.908E+00	**6.988E-03**
	Median	1.926E-01	**2.935E-30**	7.021E-07	3.705E-03	5.886E-08
	SD	1.954E-01	1.322E-02	2.242E-03	7.612E+00	1.016E-02
*f*5	Max	4.360E+04	2.878E+01	6.904E+00	1.169E+02	**4.475E+00**
	Min	3.951E-01	9.716E+00	1.803E-02	4.242E-03	**6.589E-04**
	Average	3.889E+00	2.014E+01	1.943E+00	5.823E+00	**9.738E-02**
	Median	2.671E+00	2.031E+01	1.387E+00	1.634E-01	**6.896E-03**
	SD	2.933E+00	4.579E+00	1.883E+00	1.661E+01	6.317E-01
*f*6	Max	**1.470E+02**	3.325E+02	1.866E+02	3.674E+02	3.691E+02
	Min	7.023E+01	2.681E+02	1.282E+02	1.047E+02	**4.565E+01**
	Average	1.144E+02	2.989E+02	1.567E+02	1.608E+02	**9.095E+01**
	Median	1.153E+02	2.987E+02	1.562E+02	1.380E+02	**8.040E+01**
	SD	1.800E+01	1.627E+01	1.476E+01	7.422E+01	5.905E+01
*f*7	Max	6.681E+00	4.112E+00	**1.155E+00**	2.081E+01	1.185E+00
	Min	2.971E+00	**3.908E-14**	3.658E-04	2.730E+00	2.156E-04
	Average	4.589E+00	3.283E+00	6.239E-02	1.805E+01	**6.637E-02**
	Median	4.658E+00	3.559E+00	**1.050E-03**	1.861E+01	4.394E-03
	SD	9.127E-01	1.049E+00	2.461E-01	3.056E+00	2.336E-01
*f*8	Max	7.181E-01	**6.631E-16**	1.299E-02	1.751E+00	1.656E+00
	Min	3.508E-01	**6.075E-16**	1.078E-03	6.473E-01	4.143E-02
	Average	4.840E-01	**6.572E-16**	2.609E-03	8.916E-01	2.269E-01
	Median	4.634E-01	**6.630E-16**	2.025E-03	8.664E-01	1.875E-01
	SD	9.396E-02	1.684E-17	2.242E-03	2.047E-01	2.195E-01
*f*9	Max	1.417E+01	**2.501E-23**	2.714E-04	2.124E+02	2.070E+02
	Min	1.190E+00	**2.044E-26**	1.188E-04	7.489E-05	2.718E-05
	Average	6.873E+00	**3.180E-24**	1.744E-04	2.972E+01	4.552E+00
	Median	5.710E+00	**1.248E-24**	1.657E-04	1.006E+01	1.912E-04
	SD	4.177E+00	4.923E-24	3.345E-05	5.829E+01	2.929E+01

Table 1.10 continuation

1	2	3	4	5	6	7
*f*10	Max	1.219E+06	5.444E+03	**9.795E+01**	4.139E+11	3.917E+11
	Min	7.318E+02	1.786E+03	7.999E-01	6.117E-03	**1.356E-05**
	Average	5.336E+04	3.159E+03	**2.309E+01**	8.277E+09	7.835E+09
	Median	1.004E+04	2.832E+03	1.430E+01	3.397E-02	**5.360E-04**
	SD	1.791E+05	9.887E+02	2.336E+01	5.853E+10	5.540E+10
*f*11	Max	4.200E+00	**8.999E-01**	1.500E+00	1.900E+00	3.445E+01
	Min	1.400E+00	**5.999E-01**	8.999E-01	9.999E-01	7.999E-01
	Average	2.782E+00	**7.639E-01**	1.133E+00	1.384E+00	1.661E+00
	Median	2.700E+00	**7.999E-01**	1.100E+00	1.300E+00	9.999E-01
	SD	5.759E-01	6.312E-02	1.342E-01	2.198E-01	4.733E+00
*f*12	Max	**1.456E-19**	5.866E-11	1.108E-18	2.908E-04	2.668E-05
	Min	4.471E-20	1.163E-16	3.265E-20	3.270E-20	**2.428E-20**
	Average	**8.429E-20**	2.695E-12	1.236E-19	1.341E-05	5.335E-07
	Median	7.399E-20	4.806E-14	7.170E-20	4.011E-20	**3.494E-20**
	SD	2.652E-20	9.472E-12	1.854E-19	5.516E-05	3.773E-06

The most valuable statistical indicator in the list is a median. It is the least subjected by the influence of the random character of the algorithms' performance. Thus, for each of the compared algorithms, we have counted the total number of the cases then it is the best (in terms of median value): LDW-PSO – 0, IA-PSO – 6, Ring-PSO – 1, ME-PSO – 0, ME-D-PSO – 5. Such preliminary analysis shows the advantages of IA-PSO and ME-D-PSO algorithms. However, in order to establish the most powerful algorithm, we should carry out a deeper analysis.

1.3 Engineering management of algorithms performance analysis

1.3.1 Development of estimation criteria

In order to establish the algorithms' efficiency, the numbers given in Table 1.10 should be properly analyzed. We will conduct the analysis with two indicators which are based on the reached median values of the cost functions.

The first estimation criterion – is the ranks' sum of an algorithm. In order to calculate that indicator for each of the cost function, all the algorithms have been ranked by median values. Then we added all the algorithms ranks and obtained for each of the algorithm the numerical values of the first estimation criterion.

The drawback of such an estimation approach is in the fact that all the criterion values are „unweighted". They do not reflect the numerical values of the median. In order to overcome that disadvantage following criterion has been proposed:

$$LogCr = \frac{\log_{10}\left(\dfrac{f_{zj=1}}{f_{zj=J}}\right)}{\max\left(\left\{\log_{10}\left(\dfrac{f_{zj=1}}{f_{zj=J}}\right)\right\}\right)}, \tag{1.7}$$

where $f_{zj=1}$ – the median value of the z-th algorithm (for its fifty runs), which is related to the first algorithm's iteration (the first iteration of all algorithms is the same – it is calculation of a cost function for randomly initialized arguments); $f_{zj=J}$ – the median value of the z-th algorithm, which is related to the J-th (the last) iteration of the algorithm. Criterion (1.7) shows the algorithm's performance with relation to the best one (of course, for particular cost function). For the best algorithm (in terms of the smallest median values of the particular cost function) *LogCr* equals to one. Proposed criterion (1.7) should be applied for each of the benchmarks.

1.3.2 Estimation of the compared algorithms

All the calculated values of the estimation criteria are given in Table 1.11 and Table 1.12.

Table 1.11 The first estimation criterion (algorithms' ranks sum)

Functions	Algorithms' ranks				
	LDW-PSO	IA-PSO	Ring-PSO	ME-PSO	ME-D-PSO
*f*1	1	5	2	3	4
*f*2	1	5	4	2	3
*f*3	2	3	4	1	5
*f*4	1	5	3	2	4
*f*5	2	1	3	4	5
*f*6	4	1	2	3	5
*f*7	2	3	5	1	4
*f*8	2	5	4	1	3
*f*9	2	5	4	1	3
*f*10	1	2	3	4	5
*f*11	2	5	3	1	4
*f*12	2	1	3	4	5
Criterion value	**22**	**41**	**40**	**27**	**50**

Overall estimation of the algorithm's performance for the first criterion shows that ME-D-PSO is the best one. The worst algorithm among estimated is LDW-PSO.

The calculated values of the second criterion are given in Table 1.12.

Based on the *LogCr* criterion we may argue that ME-D-PSO is the best algorithm. However, IA-PSO is characterized by a similar performance.

Table 1.12 The second estimation criterion (*LogCr* values)

Functions	Algorithms				
	LDW-PSO	IA-PSO	Ring-PSO	ME-PSO	ME-D-PSO
*f*1	0.141	1.000	0.344	0.353	0.417
*f*2	0.861	1.000	0.953	0.903	0.906
*f*3	0.782	0.861	0.952	0.498	1.000
*f*4	0.075	1.000	0.250	0.130	0.284
*f*5	0.403	0.200	0.469	0.683	1.000
*f*6	0.781	0.204	0.597	0.672	1.000
*f*7	0.152	0.180	1.000	0.013	0.856
*f*8	0.041	1.000	0.193	0.023	0.066
*f*9	0.062	1.000	0.235	0.053	0.233
*f*10	0.517	0.554	0.706	0.880	1.000
*f*11	0.692	1.000	0.919	0.877	0.944
*f*12	0.983	0.686	0.984	0.997	1.000
The sum of *LogCr* criterion values	**5.493**	**8.685**	**7.602**	**6.083**	**8.705**

Based on two criteria we have identified the best algorithm. It is ME-D-PSO. Thus, such a simple enhancing of ME-PSO algorithm leads to significant improvement in its efficiency [43].

Conclusions to chapter 1

1. In the research we proposed the novel PSO-based technique (ME-PSO). The basic idea of it is in reinitialization of the stagnant swarm. The article contains the description of the stagnation criteria and one of them has been used in the presented above calculations. The used in calculations criterion appeals to the

rate of global best reduction. If it is low then a swarm should be reinitialized (the new epoch of swarm is commencing).

2. The value of *AR* (acceptable rate) is a matter for further studies. It is necessary to found the connections between *AR* and parameters of optimization problem: its dimensionality, search domain, a function topology, etc.
3. The main advantage of the ME-PSO is the following: the greater the number of iterations the better the value of a reached extremum of a function. Proposed ME-PSO algorithm may be combined with other PSO modifications. The reasonable combinations of ME-PSO and other PSO-based techniques, impact of parameters on the algorithm performance are the issues for further studies.
4. Starting from PSO-algorithm with reinitialization of the swarm (ME-PSO) we enhanced it with diversity mechanism. In order to measure the algorithm performance, two criteria have been proposed. Their values clearly showed the superiority of the new algorithm ME-D-PSO over other ones under comparison (for example, ME-D-PSO algorithm has the 8.705 estimation of the sum of *LogCr* criterion values – the best one among others; the same refers to the algorithms' ranks). In the calculations, we have used ten well-known benchmark functions. Such criteria may be exploited in the optimization methods comparison.

References to chapter 1

1. Bozorg-Haddad O., Solgi M., Loáiciga H.A. Meta-Heuristic and Evolutionary Algorithms for Engineering Optimization, John Wiley & Sons Inc, Hoboken, USA, 2017.
2. Kennedy J., Eberhart R.C. Particle swarm optimization. – In Proceedings of IEEE International conference on Neural Networks, 1995, 1942-1948.

3. Kiranyaz S., Ince T., Yildirim A., Gabbouj M. Evolutionary artificial neural networks by multi-dimensional particle swarm optimization. – Neural Networks, Vol. 22, 2009, Issue 10, 1448-1462.
4. Heo J.S., Lee K.Y., Garduno-Ramirez R. Multiobjective control of power plants using particle swarm optimization techniques. IEEE Transactions on Energy Conversion, Vol. 21, 2006, Issue 10, 552-561.
5. Zamani M., Karimi-Ghartemani M., Sadati N., Parniani M. Design of a fractional order PID controller for an AVR using particle swarm optimization. – Control engineering practice, Vol. 17, 2009, Issue 12, 1380-1387.
6. Chander, A., Chatterjee A., Siarry P. A new social and momentum component adaptive PSO algorithm for image segmentation. – Expert systems with applications, Vol. 38, 2011, Issue 5, 4998-5004.
7. Samira Bordbar, Pirooz Shamsinejad. A New Opinion Mining Method based on Fuzzy Classifier and Particle Swarm Optimization (PSO) Algorithm – Cybernetics and Information Technologies, Vol. 18(2), 2018, 36-50.
8. Qingping He, Yibing Lv. Particle Swarm Optimization Based on Smoothing Approach for Solving a Class of Bi-Level Multiobjective Programming Problem, Cybernetics and Information Technologies, Vol. 17(3), 2017, 59-74.
9. Bao G.Q., Zhang D., Shi J.H., Jiang J.Z. Optimal design for cogging torque reduction of transverse flux permanent motor using particle swarm optimization algorithm. – Power Electronics and Motion Control Conference, Vol. 4, 2004.
10. Shurub Yu.V., Dudnyk A.O., Lavinskiy D.S. Optimization of regulators of frequency controlled induction electric drives under the stochastic loadings. – Journal Tekhnichna elektrodynamika, Vol. 4, 2016, 53-55.
11. Taher N. A new fuzzy adaptive hybrid particle swarm optimization algorithm for non-linear, non-smooth and non-convex economic dispatch problem. – Applied energy, Vol. 87, 2010, Issue 1, 327-339.

12. Rao R.V., Savsani V.J., Vakharia D.P. Teaching-Learning-Based Optimization: An optimization method for continuous non-linear large scale problems. – Information sciences, Vol. 183, 2012, Issue 1, 1-15.

13. Liu Dasheng, Tan K.C., Goh C. K., Ho W.K. A multiobjective memetic algorithm based on particle swarm optimization. – IEEE Transactions on Systems Man and Cybernetics, Part B (Cybernetics), Vol. 37, 2007, Issue 1, 42-50.

14. Loveikin V.S., Romasevych Yu.O. Dynamic optimization of a mine winder acceleration mod. – Naukovyi Visnyk Natsionalnoho Hirnychoho Universytetu, Vol. 4, 2017, 55-61.

15. Clerc M. Back to random topology [Electronic resource]. – Access mode: http://clerc.maurice.free.fr/pso/random_topology.pdf.

16. Richards M., Ventura D. Dynamic Sociometry in Particle Swarm Optimization. – Proceedings of the Joint Conference of Information Sciences, 2003, 1557–1560.

17. Clerc M. Variable PSO [Electronic resource]. – Access mode: http://clerc.maurice.free.fr/pso/2011-01-20_Variable_PSO.zip

18. Suresh K., Ghosh S., Kundu D., Sen A. Inertia-adaptive particle swarm optimizer for improved global search, Intelligent Systems Design and Applications, 2, 2008, 253-258.

19. Yong D., Wu Chuansheng W., Haimin G. Particle Swarm Optimization Algorithm with Adaptive Chaos Perturbation. – Cybernetics and Information Technologies. Special Issue on Logistics, Informatics and Service Science, Vol 15, 2015, No 6, 70-80.

20. Jordan J., Helwing S., Wanka R. Social interaction in particle swarm optimization, the ranked FIPS, and adaptive multi-swarms. – Proceedings of the 10th annual conference on Genetic and evolutionary computation, 2008, 49-56.

21. Parsopoulos K. E. Parallel cooperative micro-particle swarm optimization: A master-slave model. – Applied soft computing, Vol 12, 2012, Issue 11, 3552-3579.
22. Garg H. A hybrid PSO-GA algorithm for constrained optimization problems. – Applied Mathematics and Computation, Vol 274, 2016, 292-305.
23. Pluhacek M., Senkerik R., Viktorin A., Kadavy T., Zelinka I. A review of real-world applications of particle swarm optimization algorithm. Recent Advances in Electrical Engineering and Related Sciences: Theory and Application. Lecture Notes in Electrical Engineering: Vol. 465, 2018. Springer: Cham
24. Nazaruddin Y.Y., Andrini A.D., Anditio B. PSO based PID controller for quadrotor with virtual sensor. Proceedings of the 3rd IFAC Conference on Advances in Proportional-Integral-Derivative Control, 2018, Vol. 51, Issue 4, 358-363
25. Štimac G., Braut S., Žigulić, R. Comparative analysis of PSO algorithms for PID controller tuning. Chinese journal of mechanical engineering, 2014, 27(5), 928-936.
26. Smail, H., Smail, M.K., Fetha, C., & Bahi, T. PSO optimized PID regulator for a variable frequency brushless synchronous generator. Power engineering and electrical engineering, 2018, 16(4), 414-425.
27. Latha, K., Rajinikanth, V. & Surekha, P.M. (2013). PSO-based PID controller design for a class of stable and unstable systems, ISRN Artificial Intelligence, 2013 (article ID 543607), 11 pages.
28. Nur Iffah, M.A., Nafrizuan M.Y., Ho J.F., Wan A. & Wan Y. (2019). Optimization of the PID-PD parameters of the overhead crane control system by using PSO algorithm. Engineering Application of Artificial Intelligence Conference 2018 (EAAIC 2018), 255, 04001, 8 pages. Sabah: MATEC Web of Conferences.
29. Atacak I., Küçük B. PSO-based PID controller design for an energy conversion system using compressed air, Tehnički vjesnik, 2017, 24(3), 671-679.

30. Yudong Z., Shuihua W., Genlin J. A Comprehensive survey on particle swarm optimization algorithm and its applications. Mathematical Problems in Engineering, 2015 (article ID 931256), 2015, 38 pages.

31. Shi C., Yuhui S. Diversity control in particle swarm optimization, 2011 IEEE Symposium on Swarm Intelligence, 2011, 9 pages. Paris: IEEE.

32. Riget J., Vesterstrøm J.S. A diversity-guided particle swarm optimizer – the ARPSO., 2002, (EVALife Technical Report no. 2002-02).

33. Hui W., Hui S., Changhe L., Shahryar R., Jeng-Shyang P. Diversity enhanced particle swarm optimization with neighborhood search. Information Sciences, 2013, 223, 119-135.

34. Olorunda O., Engelbrecht, A.P. Measuring exploration/exploitation in particle swarms using swarm diversity. 2008 IEEE Congress on Evolutionary Computation (IEEE World Congress on Computational Intelligence), 2008, pages 7. Hong Kong: IEEE.

35. Shi Y., Eberhart R. Population diversity of particle swarms. Proceedings of the 2008 Congress on Evolutionary Computation (CEC2008), 2008, pp. 1063-1067. Hong Kong: IEEE.

36. Shi Y., Eberhart R. Monitoring of particle swarm optimization. Frontiers of Computer Science, 2009, 3(31) 31-37.

37. Zhan Z., Zhang J., Shi Y. Experimental study on PSO diversity. Third International Workshop on Advanced Computational Intelligence, 2010, 310-317. China: Suzhou Jiangsu.

38. Storn R., Price K. Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces, Journal of Global Optimization, 11, 1997, 341-359.

39. Romasevych Yu., Loveikin V. A novel multi-epoch particle swarm optimization technique, Cybernetics and Information Technologies, 2018, 18(3) 62-74.

40. Romasevych Yu., Loveikin V., Usenko S. PI-controller tuning optimization via PSO-based technique, Przegląd elektrotechniczny, 95, 2019, NR 7/2019, 33-37.

41. Shi Y., Eberhart, R.C. A modified particle swarm optimizer. Proceedings of the IEEE Conference on Evolutionary Computation, 1998, pp. 69-73. Anchorage: IEEE.

42. Xiaodong L. Niching without niching parameters: particle swarm optimization using a ring topology. IEEE Transactions on Evolutionary Computation, 2010, Vol. 14, Issue 1, 150-169.

43. Romasevych Y., Loveikin V., Makarets V. Optimal constrained tuning of PI-controllers via a new PSO-based technique. International Journal of Swarm Intelligence Research, 2020, 11(4), pp. 87-105.

CHAPTER 2. AGROTRONICS OF GENERAL APPROACH TO PI-CONTROLLERS TUNING OPTIMIZATION

2.1 Engineering management of constrained optimization of the PI-controller tuning

2.1.1 The general statement of the optimization problem

In this section, we have investigated PI-controllers, as they have the most common usage in many areas of automated processes. The problem of constrained optimal PI-controller tuning includes the following elements: a set of criteria to minimize, set of constraints, and the stability requirement. Let us assume that the mathematical model of the controlled system is known (it is a differential equation). The general setting of the optimization of a PI-controller tuning problem may be presented in the following form:

$$\begin{cases} \sum\limits_{k=0}^{K} A_k \dfrac{d^k x(t)}{dt^k} = \sum\limits_{q=0}^{Q} B_q \dfrac{d^q u}{dt^q},\ Q < K; \\ \lim\limits_{t\to\infty} x = r,\ \lim\limits_{t\to\infty} \dfrac{d^k x(t)}{dt^k} = 0; \\ u = K_p(r - x(t-\tau)) + T_i^{-1}\int\limits_0^t (r - x(t-\tau))dt; \\ f_e(\dfrac{d^k x(t)}{dt^k}, \dfrac{d^q u}{dt^q}, A_k,\ B_q) \le F_e,\ k = (\overline{0, K}),\ q = (\overline{0, Q}),\ e = (\overline{1, E}); \\ Cr_i \to \min,\ i \in (\overline{1,\ I}), \end{cases} \tag{2.1}$$

where Cr_i – i-th criterion to minimize (in the frame of the current research we accept that all the criteria reflect various undesirable features of the plant); I – total number of criteria to minimize; x – plant variable; u – control function (in the following we will denote it as „control”); K_p and T_i – proportional and integral coefficients of PI-controller respectively; K – the order of the plant (the order of polynomial denominator of the plant transfer function); Q – the order of polynomial numerator of the plant transfer function; A_k and B_q – the k-th and the q-th coefficients which depend on the parameters of the plant; f_e – e-th function, which in the general case depends on the plant variable x, and its higher derivatives with time, control u and its higher derivatives with time, as well as the coefficients A_k and B_q; F_e – the limit value of the e-th function; E – total number of inequalities; t – time; τ – time delay; r – set-point. The solution of the problem (2.1) is such values of the coefficients K_p and T_i, which require all the conditions (2.1). It should be pointed out, that it is desirable to find a pareto-optimal solution of the problem (2.1).

Note, in the sense of computational complexity problem (2.1) is quite difficult, especially for quite big numbers K and E. That is why optimization algorithm performance is crucial. In the process of an algorithm applying, on each of the iteration inequality (2.1) should be checked. This complicates the operation of the algorithm. Therefore, there is a need to reduce the initial problem (2.1) to a single complex criterion.

2.1.2 Reducing the initial problem to the unconstrained optimization (development of the generalized optimization criterion)

The crucial requirement to the PI-controller tuning – is the stability of the plant, which mathematically presented with the fourth line of the system (2.1). These expressions may be substituted with a single terminal criterion – the norm of the error and its higher derivatives with time. However, commonly accepted, that the plant is stable if the variable x has achieved a range $(0.9\ldots1.05)r$ and the higher derivatives of the variable x with time are minor. That is why we may substitute the stability requirements with the following terminal criterion:

$$Ter = \begin{cases} 0, \ if \ \ 0.95r \le x(T) \le 1.05r \wedge \dfrac{d^k x(T)}{dt^k} < sv_k; \\ \delta_T \sqrt{(x(T)-r)^2 + \sum\limits_{k=1}^{K} \left(\dfrac{d^k x(T)}{dt^k} \right)^2}, \ if \ \ 0.95r > x(T) \vee x(T) > 1.05r \vee \dfrac{d^k x(T)}{dt^k} > sv_k, \end{cases} \tag{2.2}$$

where *T* – the moment when the stability conditions are met; *sv_k* – a small admissible value of *k*-th derivative of *x* with time; δ_T – termination coefficient, which reflect the importance of achieving of stability conditions. The absolute minimum of criteria (2.2) is equal to zero. When it is achieved, stability requirements are met.

The problem of reducing the set of criteria Cr_i to the single one (complex criterion) may be solved in the following manner:

$$CC = \sum_{i=1}^{I} w_i m_i Cr_i \to \min, \tag{2.3}$$

where w_i – is the *i*-th weighting coefficient, that reflects the importance of the criterion Cr_i; m_i - *i*-th coefficient, which reduces the product $m_i Cr_i$ into dimensionless value. The set of the inequalities (the second line of the system (2.1)) might be substituted with the following criterion:

$$Cr_{ineq} = \begin{cases} 0, \ if \ F_e - f_e(\dfrac{d^k x(t)}{dt^k}, \dfrac{d^q u}{dt^q}, A_k, \ B_q) \ge 0; \\ i\delta_p \sum\limits_{y=1}^{Y} (F_y - f_y(\dfrac{d^k x(t)}{dt^k}, \dfrac{d^q u}{dt^q}, A_k, \ B_q)), \ if \\ F_1 - f_1(\dfrac{d^k x(t)}{dt^k}, \dfrac{d^q u}{dt^q}, A_k, \ B_q) < 0 \vee ... \\ ... \vee F_2 - f_2(\dfrac{d^k x(t)}{dt^k}, \dfrac{d^q u}{dt^q}, A_k, \ B_q) < 0 \vee \\ F_E - f_E(\dfrac{d^k x(t)}{dt^k}, \dfrac{d^q u}{dt^q}, A_k, \ B_q) < 0, \end{cases} \tag{2.4}$$

where *y* – index of violated inequality in the requirements (2.1); *Y* – total number of the violated inequalities; δ_p – penalty coefficient.

Now, we may develop the generalized complex optimization criterion. Before we present its expression one note should be done.

Each of the criteria (2.2)-(2.4) has a different importance. In order to take that into account, the generalized complex optimization criterion should have proper topology features. We proposed it like pit-in-pit topology:

$$Cr = Ter + Cr_{ineq} + CC \rightarrow \min, \tag{2.5}$$

The top pit reflects the topology of criterion (2.2), the under pit responds to the criterion (2.4), and, finally, the bottom one has the criterion (2.3) topology. Nesting of the pits is provided with coefficients δ_T and δ_p, they must be set in the following manner $\delta_T >> \delta_p > 0$. Criterion (2.5) may be calculated as the mathematical model of the plant is known and the coefficients K_p and T_i are set. In the frame of the investigation, we consider the cost function as some MISO-function. Its inputs are coefficients K_p and T_i, the body of the MISO-function is a numerical integration of the differential equation of the plant, and the output is the calculated criterion (2.5). Now, we may illustrate the efficiency of the proposed approach for different plants, which are described with the last line of the system (2.1).

2.2 Engineering management of ME-D-PSO application

2.2.1 Conditions of the numerical experiments

In order to prove the superiority of the developed tuning technique, we state four problems to solve. All the calculations were performed for the well-known transfer functions given in Table 2.1. They were proposed by K.J. Åström and T. Hägglund in the work [1].

In 2.1 subsection we have used the differential equation.

However, in Table 2.1 we gave the transfer functions as they are common in the automatic control theory. There are no contradictions; one may obtain transfer function as soon as the differential equation is known, and vice versa.

Table 2.1 Transfer functions for constrained optimization of PI-controller tuning

Transfer function	Search domain		Note
	K_p	T_i	
1	2	3	4
$G_1(s)=\frac{e^{-4s}}{(0.1s+1)^2}$	0…10	0…10	With time delay
$G_2(s)=\frac{1}{(s+1)^3}$	0…10	0…10	It has been used by controller manufacturers as a test case for a long time
$G_3(s)=\frac{(1-2s)}{(s+1)^3}$	0…10	0…10	Difficult to control
$G_4(s)=\frac{1}{(s+1)(1+0.1s)(1+0.01s)(1+0.001s)}$	0…10	0…10	There are dramatic improvements when going from PI to PID control

The inequalities for all of the considered cases are presented below:

$$\begin{cases} \max(x-r)\le 0; \\ u\le 1.5. \end{cases} \tag{2.6}$$

The first inequality (2.6) requires zero overshoot, the second one allows avoiding the saturation element in the output of the PI-controller.

For all experiments r=1, sv_k=0.005.

In the investigation the following complex criterion has been used:

$$CC = 0.2t_s^{-1}\int_0^{t_s}|u|dt + 0.6t_s, \tag{2.7}$$

where t_s – settling time. Numerical coefficients near the summands are the product of weighting coefficients and the coefficients for reducing the values of summands to dimensionless value. Finally, the values of coefficients δ_T and δ_p are 10^6 and 10^3 respectively.

In order to detect the tuning efficiency the following quality indicators have been used: mean integral error $t_s^{-1}\int_0^{t_s} edt$ (MIE), mean integral control $t_s^{-1}\int_0^{t_s} udt$ (MIC), and settling time t_s. In the carried out calculations, we used the following ME-D-PSO parameters: J=50, S=20, AR=0.001.

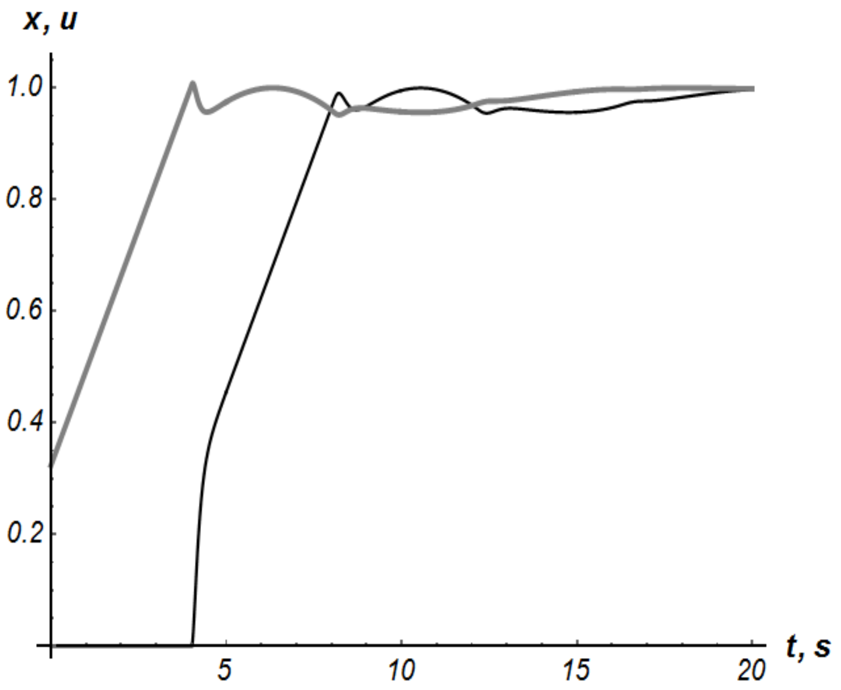
2.2.2 Obtained data analysis

All the obtained data are given in Table 2.2. In order to illustrate the reaching of the set-point r, we have built plots (Fig. 2.1).

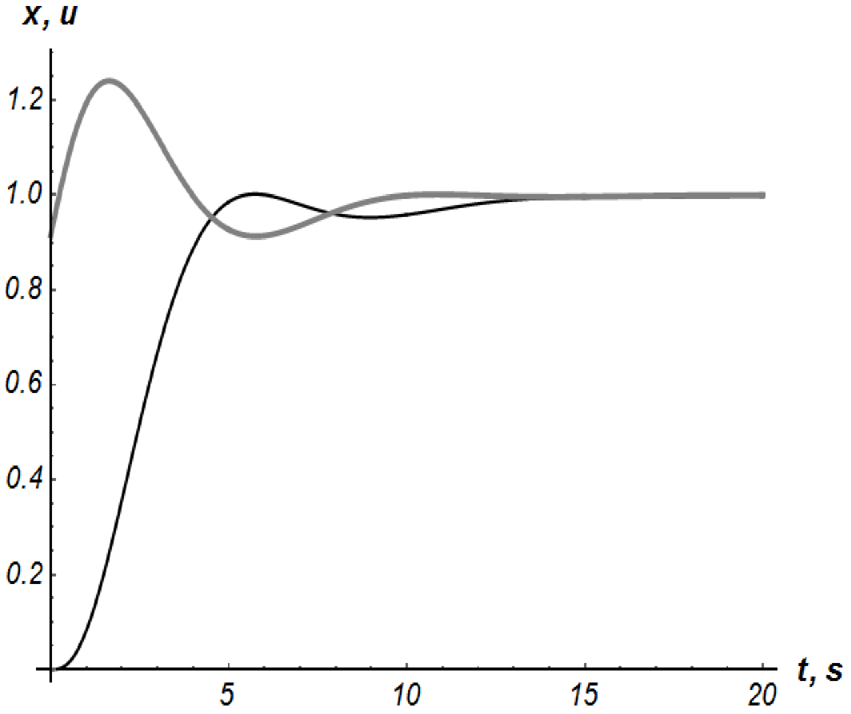
Table 2.2 Obtained PI-controller coefficients and quality indicators

Transfer function	Coefficients values		Quality indicators		
	K_p	T_i	MIE	MIC	t_s
$G_1(s) = \dfrac{e^{-4s}}{(0.1s+1)^2}$	0.317	5.917	0.70	0.82	7.95
$G_2(s) = \dfrac{1}{(s+1)^3}$	0.902	2.811	0.56	1.11	4.55
$G_3(s) = \dfrac{(1-2s)}{(s+1)^3}$	0.468	5.031	0.59	0.90	8.14
$G_4(s) = \dfrac{1}{(s+1)(1+0.1s)(1+0.01s)(1+0.001s)}$	1.342	0.736	0.37	1.18	1.94

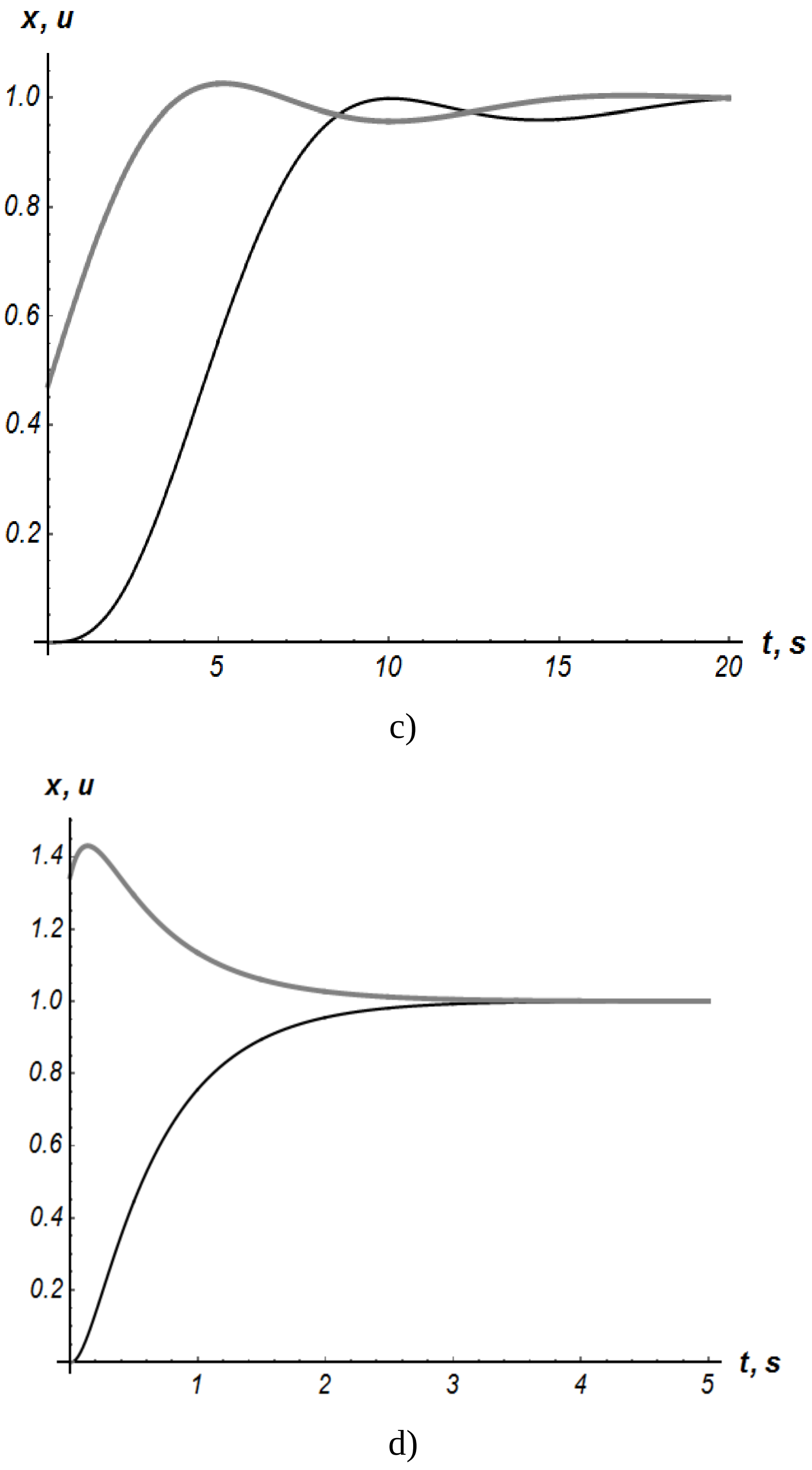
It is obvious, that for all the numerical experiments overshoot is equal to zero and control is within the set limit 1.5. Both numerical values of the quality indicators given in Table 2.2 and plots in Fig. 2.1 confirm the high efficiency of optimal PI-controller tuning.



a)



b)

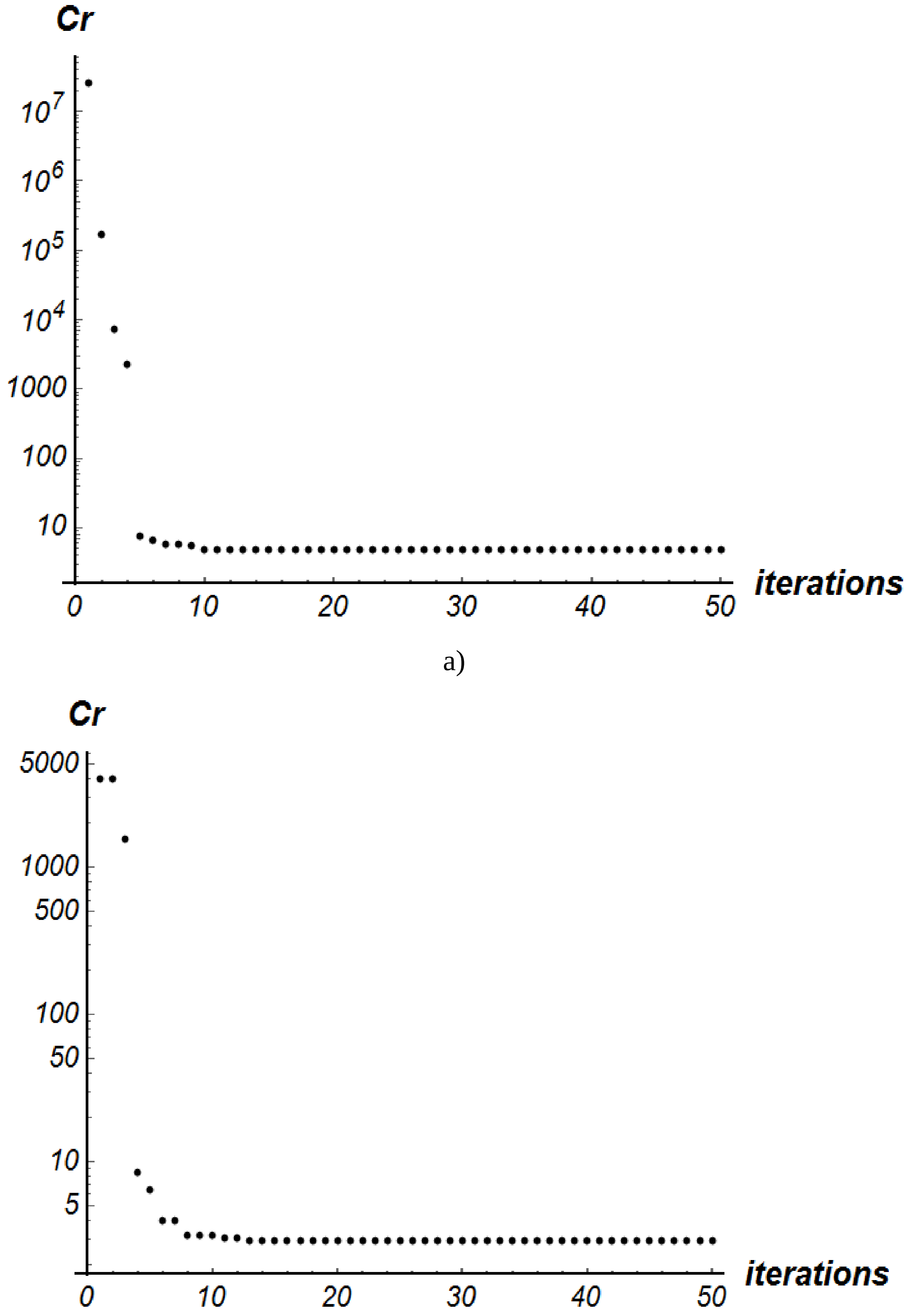


c)

d)

Fig. 2.1 Control *u* (gray curve) and controlled variable *x* (black curve) for transfer functions: a) $G_1(s)$; b) $G_2(s)$; c) $G_3(s)$; d) $G_4(s)$

In order to illustrate the decreasing of value *Cr* during the algorithm’s execution, we have built plots in the logarithmic scale (Fig. 2.2).



a)

b)

Fig. 2.2 Decreasing of *Cr* value during algorithm execution for transfer functions: a) $G_1(s)$; b) $G_2(s)$

In Fig. 2.2, *a* one may observe the pit-in-pit topology of criterion *Cr*. Indeed, the first and the second iterations of ME-D-PSO took place on the top pit. In the following two iterations, the global best of the swarm was in the middle pit. The rest of the iterations occurred in the bottom pit where the value of criterion (2.7) was decreasing. It may happen that from the very first iteration algorithm finds a stable solution of the problem and the global best of the swarm is located in the middle pit (Fig. 2.2, *b*). After three executed iteration algorithm fell into the bottom pit.

Dot plots in Fig. 2.2 confirm the high efficiency of the developed criterion (2.5) as well as the approach to find its minimum (ME-D-PSO). Thus, we may propose to exploit it for more general problems of optimal control [2].

2.3 Engineering management of ME-PSO application

2.3.1 Conditions of numerical experiments

In order to prove the efficiency of the ME-PSO application, we have stated five problems of tuning PI-controllers for different real-world plants. In Table 2.3 we have shown their transfer functions, as well as the search domains for coefficients *P* and *I* of PI-controllers.

Table 2.3 Transfer functions of real-world plants and PI-controller coefficients search domains

Transfer function	Search domains	
	P	*I*
$G_1(s)=\dfrac{7.00\cdot10^{-1}e^{-5s}}{2.20\cdot10^{2}s+1}$	$0\ldots1\cdot10^{2}$	$0\ldots1\cdot10^{1}$
$G_2(s)=\dfrac{1.83}{1.68\cdot10^{4}s^{2}+4.60s+1}$	$0\ldots5\cdot10^{0}$	$0\ldots5\cdot10^{2}$

$G_3(s)=\dfrac{2.60\cdot10^{-2}}{2.73\cdot10^5 s^2+1.35\cdot10^3 s+1}$	$0\ldots1\cdot10^2$	$0\ldots1\cdot10^2$
$G_4(s)=\dfrac{4.47\cdot10^{-1}e^{-28.6s}}{1.83\cdot10^5 s^3+1.53\cdot10^4 s^2+9.40\cdot10^1 s}$	$0\ldots5\cdot10^0$	$0\ldots5\cdot10^7$
$G_5(s)=\dfrac{1}{8.20\cdot10^{-5}s^2+6.17\cdot10^{-3}s+2.30\cdot10^{-2}}$	$0\ldots2\cdot10^0$	$0\ldots5\cdot10^0$

Used in calculations transfer functions refer to the following plants: $G_1(s)$ – to the fruit storage; $G_2(s)$ – to the grain dryer; $G_3(s)$ – to the greenhouse; $G_4(s)$ – to the heating boiler КВГМ-100, and $G_5(s)$ – to the mobile robot, that is used in a greenhouse [3]. For the transfer functions $G_1(s)$-$G_4(s)$ the control channel is „input power – temperature”, for the transfer function $G_5(s)$ it refers to „power supply voltage – the angular speed of robot drive”. Such variety of plants is caused by desire to evidentiate the generality of the methodology.

Tuning of PI-controllers for five plants should require a condition of zero overshoot. For all of the numerical experiments following parameters have been set: $\delta_T=10^8$, $\delta_p=10^6$, $r=1$; $\Delta=0.005r$, $s_n=0.001$.

In order to find values K_p and T_i, ME-PSO [4] method was applied. Note, that in work [5] it has been used for similar problems, but there were no constraints. Parameters of method ME-PSO have a dramatic impact on its performance. That is why we have listed them in Table 2.4.

Table 2.4 Parameters of ME-PSO algorithm

Parameters	Value
Social coefficient c_1	1.19
Cognitive coefficient c_2	1.19
Swarm population	25
Acceptable rate AR	0.05
Particles connection topology	full
Number of iterations	50

Optimization criterion for all the cases is described as follows:

$$C_c = 0.5t_s^{-1}\int_0^{t_s} udt + 0.3t_s^{-1}\int_0^{t_s} edt + 0.6t_s. \tag{2.8}$$

Coefficients of quality indicators in (2.8) are given in the numerical form: they reduce their dimensions to dimensionless form and refer to the importance of each of them. In the carried out calculations, apart from the optimal methodology, we have used rules of well-known in engineering practice methods: Ziegler-Nichols [6], Kappa-Tau [7], AMIGO [8], Chien-Hrones-Reswick [9], Tyreus-Luyben [10]. For transfer functions $G_1(s)$, $G_2(s)$, $G_3(s)$ and $G_5(s)$ were used Ziegler-Nichols, Kappa-Tau, AMIGO and Chien-Hrones-Reswick [9] methods. For the transfer function $G_4(s)$ Chien-Hrones-Reswick method was substituted to Tyreus-Luyben one by virtue of the fact, that the former method leads to unstable control.

The proof of the superiority of the approach is based on the comparison of quality indicators, which refer to the tuning methods mentioned above. Quality indicators, used in the frame of the research, are: mean integral error $t_s^{-1}\int_0^{t_s} edt$ (MIE), mean integral control $t_s^{-1}\int_0^{t_s} udt$ (MIC), settling time t_s, and overshoot (OS).

2.3.2 A brief analysis of tuned PI-controllers performance

The results of the calculations are presented in Table 2.5 below. The best values of the indicators are in bold.

Table 2.5 Results of numerical experiments

Tuning method	Tuned coefficients		MIE	MIC	t_s, s	OS, %
	K_p	T_i				
1	2	3	4	5	6	7

Table 2.5 continuation

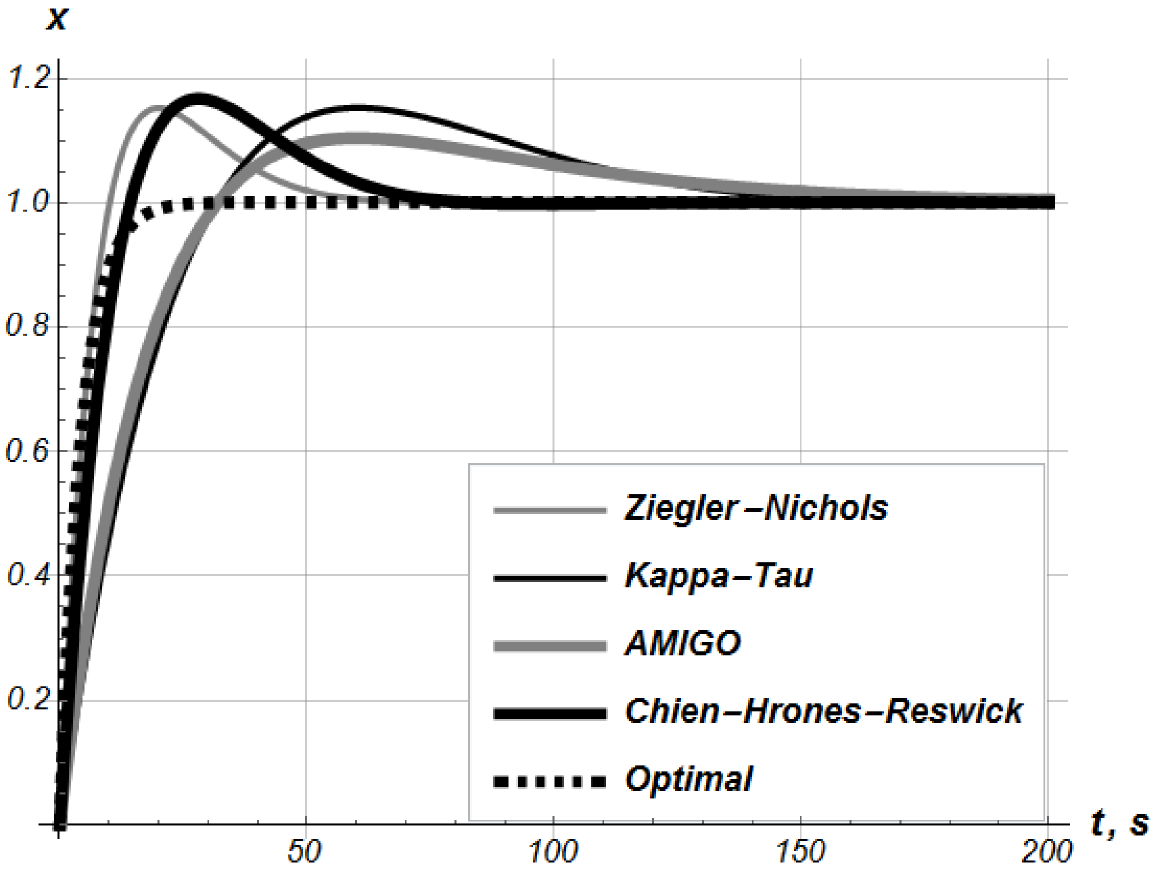
1	2	3	4	5	6	7
Transfer function $G_1(s)$						
Ziegler-Nichols	56.571	0.295	0.15	8.22	49.8	15.2
Kappa-Tau	17.198	2.238	0.16	3.83	133.0	15.2
AMIGO	20.849	2.527	**0.13**	**3.61**	145.7	10.4
Chien-Hrones-Reswick	37.714	0.530	0.17	6.48	65.1	16.7
Optimal tuning	70.720	3.181	0.26	19.95	**17.0**	**0.0**
Transfer function $G_2(s)$						
Ziegler-Nichols	5.446	23.319	**0.20**	1.36	493	40.9
Kappa-Tau	1.439	139.220	0.22	**0.81**	809	15.2
AMIGO	1.740	140.617	**0.20**	0.82	777	11.1
Chien-Hrones-Reswick	3.631	41.975	0.31	1.24	355	30.2
Optimal tuning	2.234	209.674	0.42	1.32	**255**	**0.0**
Transfer function $G_3(s)$						
Ziegler-Nichols	5.446	23.319	**0.23**	40.32	8605	26.1
Kappa-Tau	1.439	139.22	0.32	**29.33**	16640	**0.0**
AMIGO	1.740	140.617	0.31	29.44	17060	**0.0**
Chien-Hrones-Reswick	3.631	41.975	0.28	37.43	7980	11.4
Optimal tuning	85.688	15.668	0.44	63.64	**1265**	**0.0**
Transfer function $G_4(s)$						
Ziegler-Nichols	0.988	646.696	0.24	0.24	2440	55.9
Kappa-Tau	0.429	3551.73	0.21	0.08	3822	25.7
AMIGO	0.384	6684.990	**0.15**	**0.04**	6035	17.0
Tyreus-Luyben	1.675	688.077	0.18	0.30	2586	56.0
Optimal tuning	0.327	$4.683 \cdot 10^7$	0.36	0.12	**1683**	**0.0**
Transfer function $G_5(s)$						
Ziegler-Nichols	0.387	0.118	0.26	0.11	0.178	57.6
Kappa-Tau	0.110	0.798	0.23	**0.04**	0.300	21.0

Table 2.5 continuation

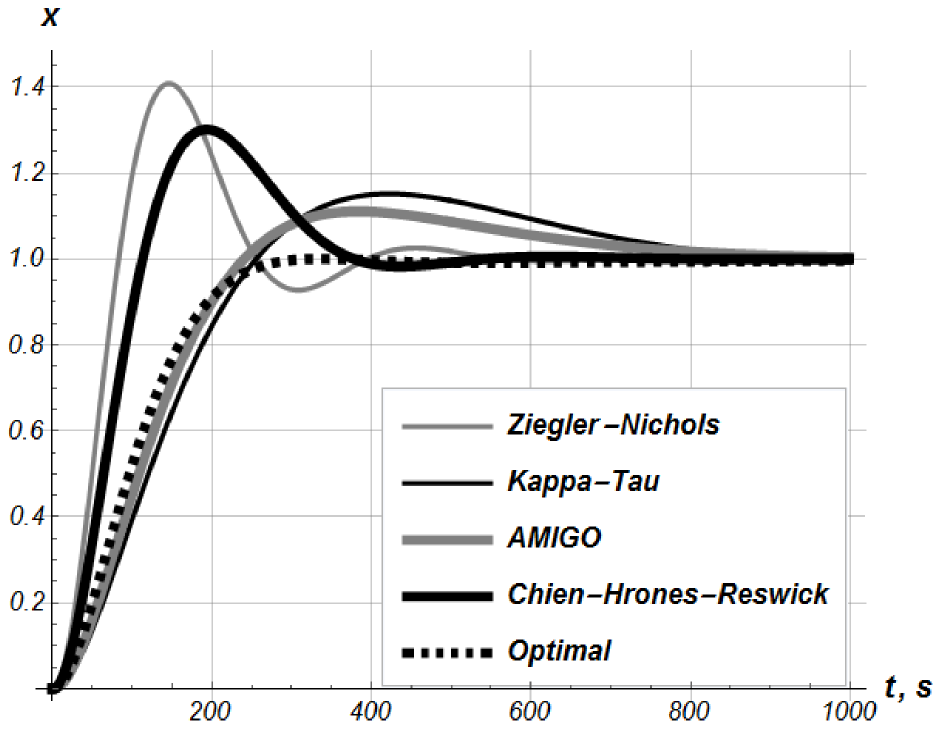
1	2	3	4	5	6	7
AMIGO	0.133	0.837	**0.19**	**0.04**	0.314	15.8
Chien-Hrones-Reswick	0.258	0.212	0.28	0.08	0.168	43.1
Optimal tuning	0.122	2.311	0.39	0.07	**0.126**	**0.0**

In order to illustrate the control dynamics and confirm numbers in Table 2.5 we have built plots (Fig. 2.3).

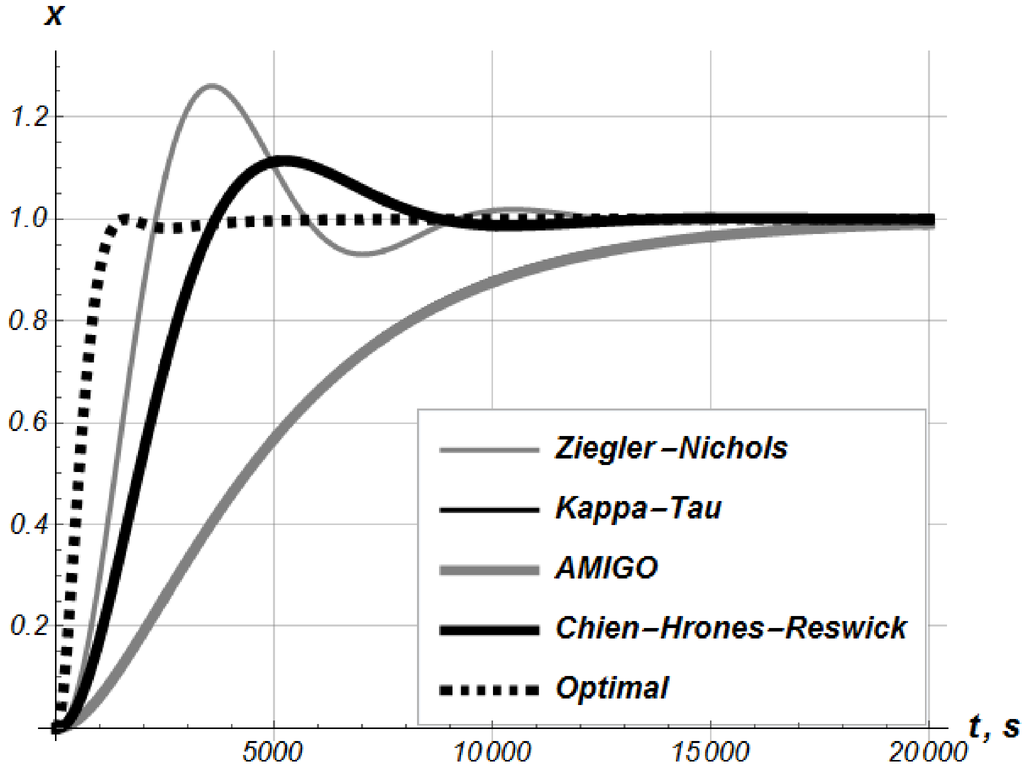
Indeed, all the plots show zero overshoot performance refers to optimal tuned PI-controllers. Another strong tendency is the shortest transitional mode: settling time indicator for optimal tuned PI-controllers is the smallest.



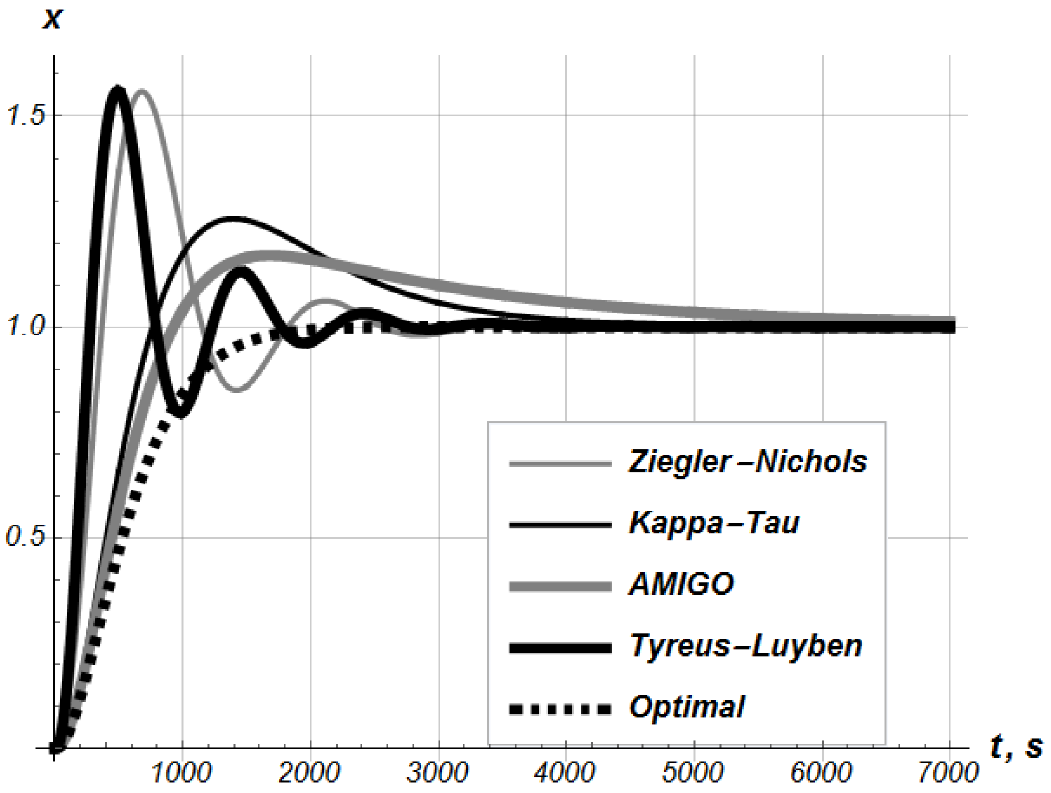
a)


x
1.4
1.2
1.0
0.8
0.6
0.4
0.2
200
400
600
800
1000
t, s
Ziegler–Nichols
Kappa–Tau
AMIGO
Chien–Hrones–Reswick
Optimal

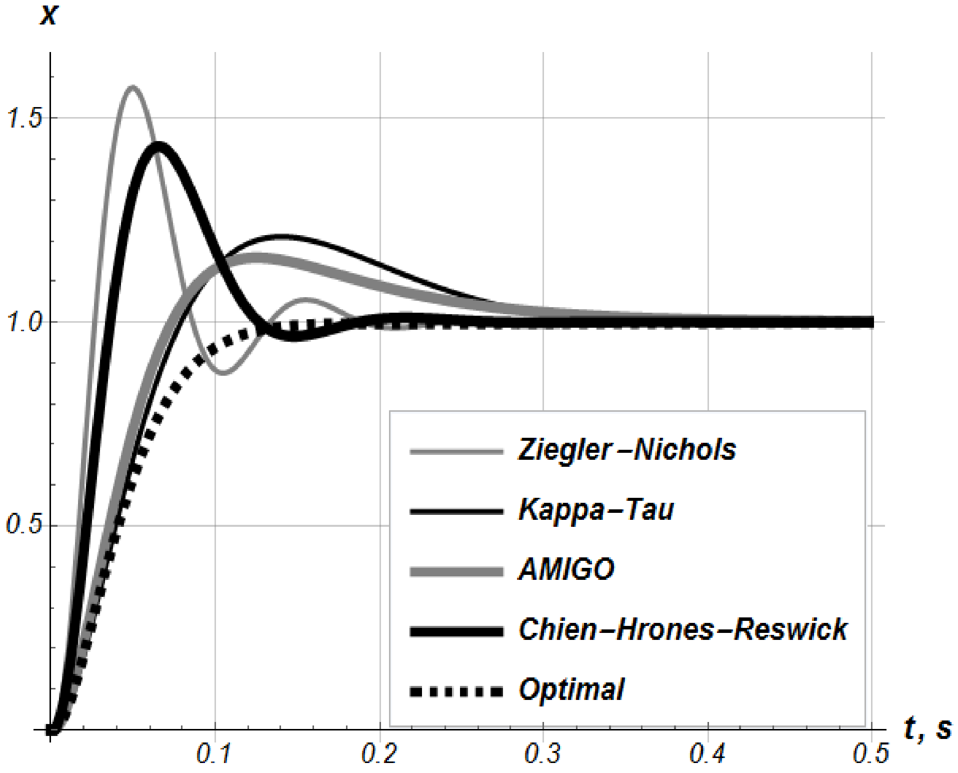
b)


x
1.2
1.0
0.8
0.6
0.4
0.2
5000
10000
15000
20000
t, s
Ziegler–Nichols
Kappa–Tau
AMIGO
Chien–Hrones–Reswick
Optimal

c)



d)



e)

Fig. 2.3 Step response of the real-world plants with transfer functions:

a) $G_1(s)$; b) $G_2(s)$; c) $G_3(s)$; d) $G_4(s)$; e) $G_5(s)$

For the plant described with transfer function $G_1(s)$ t_s is 2.93…8.57 times shorter, than for the rest of tuning methods; for the plant with transfer function $G_2(s)$ t_s – is 1.39…3.17 times shorter; for the plant with transfer function $G_3(s)$ t_s – is 6.31…13.48 times shorter; for the plant with transfer function $G_4(s)$ t_s – is 1.45…3.56 times shorter; for the plant with transfer function $G_5(s)$ t_s – is 1.33…2.49 times shorter.

Such a reduction in duration of transitional modes has a negative effect: other indicators (MIC and MIE) for all of the plants, which refer to the optimal tuned PI-controller, are bigger on the whole.

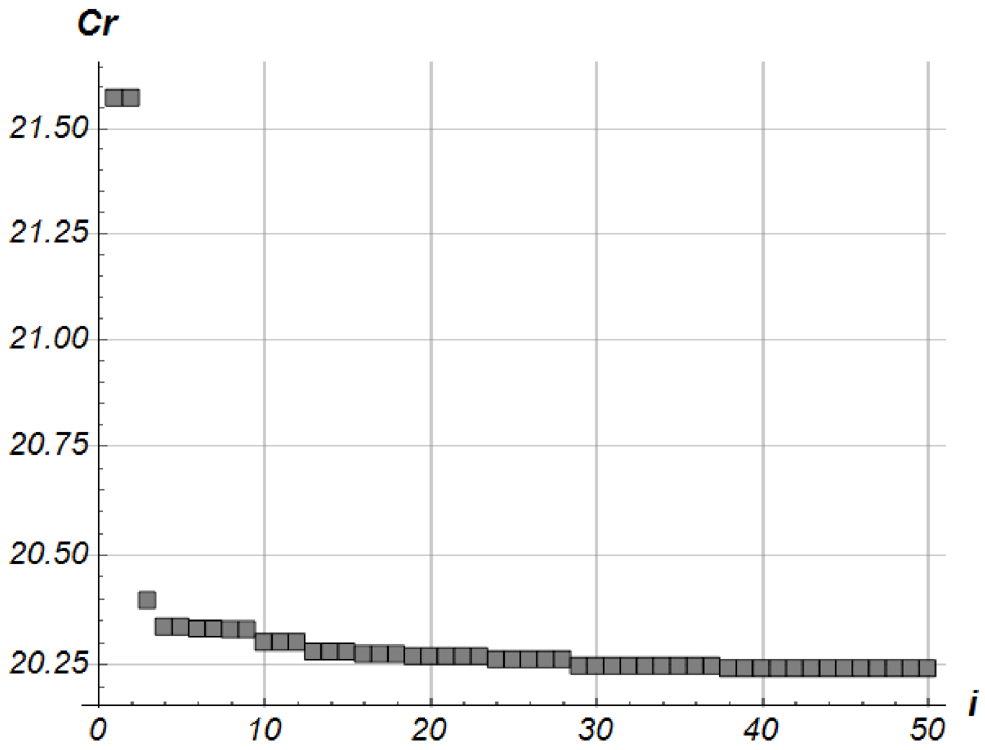
We may point the cause: the duration of the transitional modes is bigger than indicators MIC and MIE. Thus, their impact on *Cr* value is weak. In order to reduce them one should magnify their coefficients. One expects that the duration, in this case, will increase.

Summing up what has been investigated, we may state, that whatever solution has been found, it is Pareto-optimal one. It shifts the research focus on the importance of the quality indicators.

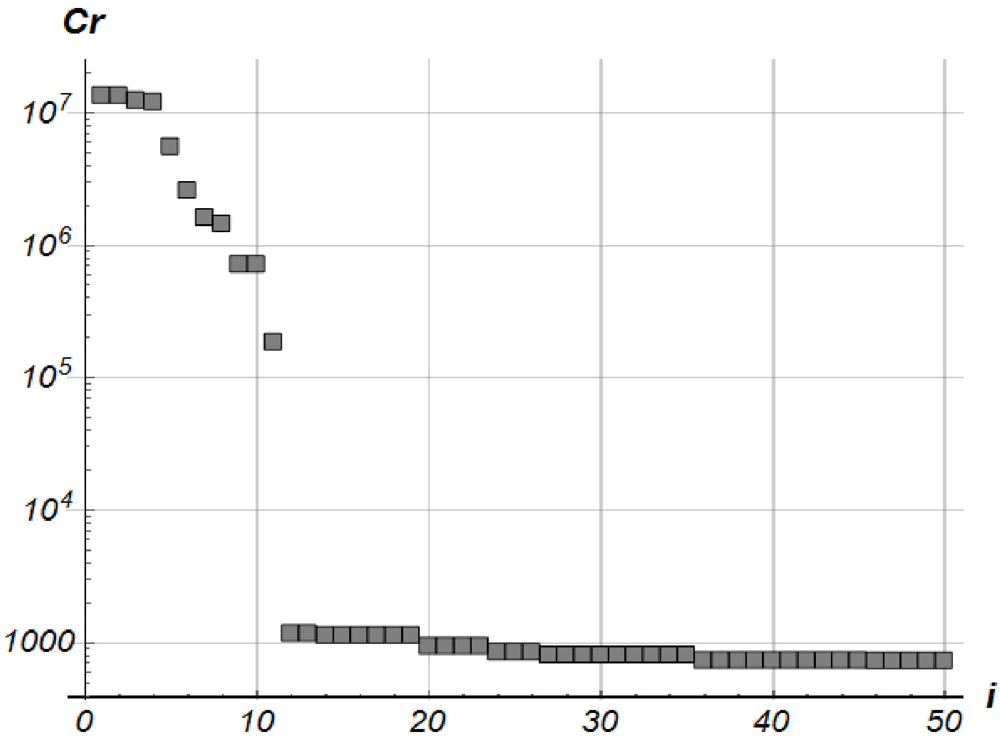
2.3.3 A brief analysis of computational efficiency of ME-PSO algorithm

In this subsection, the computational characteristics will be shown. The importance of this issue is grounded by the natural worries: whether ME-PSO provides the global minimum of the generalized criterion, or it is a local one. Such a question is natural, because of the strong desire to obtain as much advantage as possible (in sense of criterion value).

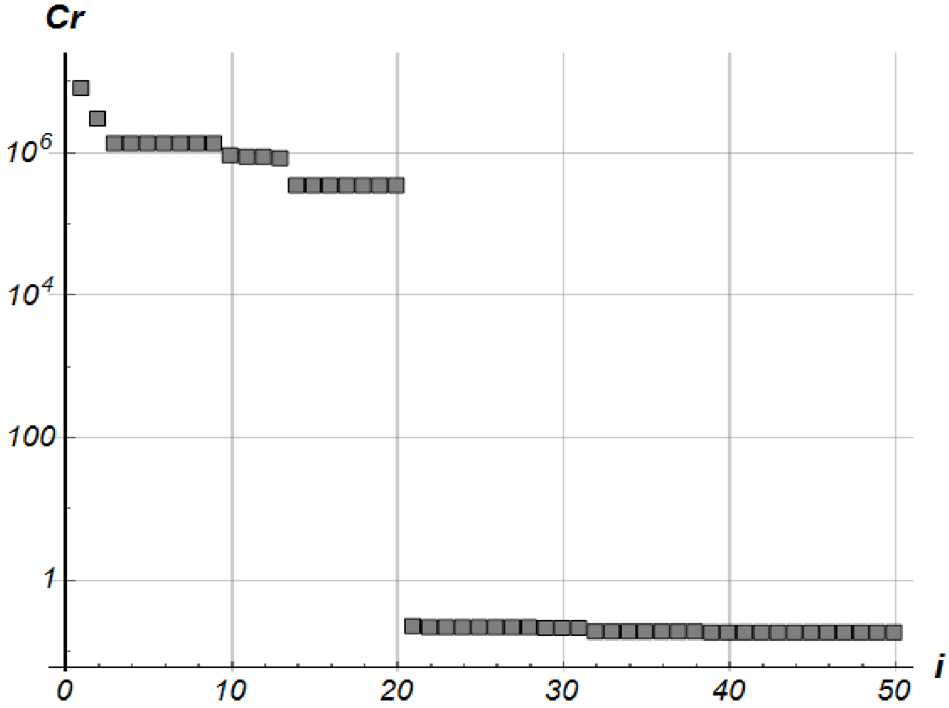
In order to investigate the issue the plots of the algorithm’s convergence were built (Fig. 2.4). Plots on Fig. 2.4 (b) (c) clearly show that great decrease of *Cr* value occurred during first algorithm iterations. As for the first optimization problem (tuning the plant with transfer function $G_1(s)$), initialization of the swarm was quite advantageous: one of the particles was initialized near to the global minimum of the criterion *Cr* and further iterations of the algorithm did not lead to the great improvement of the very first solution.



a)



b)



c)

Fig. 2.4 Convergence of ME-PSO algorithm for transfer functions:
a) $G_1(s)$; b) $G_3(s)$; c) $G_5(s)$

During algorithm’s execution, the swarm was few times reinitialized. One may observe such an effect on the plots by edges of the *Cr* value decreasing. For instance, in Fig. 2.4 (c) edges are observed on 9-th, 14-th and 21-th iterations. They prevent the algorithm from premature convergence and enhance the probability of finding the criterion’s global minimum. Thus, we have quite a strong prerequisite for a statement, that the found solutions are global minimums of the generalized criterion [11].

Conclusions to chapter 2

1. The second chapter of the monograph is devoted to the development of an approach to the constrained optimization of PI-controller tuning. In order to use ME-D-PSO for that aim, we reduced the initial problem to the unconstrained optimization problem. It is based on the criterion with pit-in-pit topology. In the researched case, we have used the three-leveled pit. However, the more

general case may include more levels (for instance, when the inequalities or/and optimization criteria are unequal with influence).

2. Having the powerful method for functions minimization (ME-D-PSO) and developed criterion we have carried out the calculation of PI-controller coefficients for four benchmark transfer functions. Analysis of the received data confirms the high efficiency of the proposed approach.
3. The efficiency of ME-PSO application was tested via tuning of PI-controllers for five real-world plants with different properties (orders and time-constants). Carried out analysis allowed to confirm the advantageousness of the approach (in a sense of satisfaction of all of the optimization problem conditions). We hope, that the methodology may be useful for other PI or/and PID-controllers tuning problems, as well as for non-classic control systems (based on fuzzy-logic, artificial neural networks, etc.). Current research is limited by linear plants. Thus, further investigations should involve non-linear ones. Another direction for further researches is connected with developing optimal controllers for MIMO-systems.

References to chapter 2

1. Åströn K.J., Hägglund T. Benchmark Systems for PID Control, IFAC Proceedings Volumes, 2000, 33(4), pp. 165-166. Terrassa: Elsevier.
2. Romasevych Y., Loveikin V., Makarets V. Optimal constrained tuning of PI-controllers via a new PSO-based technique. International Journal of Swarm Intelligence Research, 2020, 11(4), pp. 87-105.
3. Lysenko V., Bolbot I., Romasevych Y., Loveykin V., Voytiuk V. Algorithms of robotic electrotechnical complex control in agricultural production. Control Systems: Theory and Applications. River Publishers, 2018, ch. 11, pp. 271-289.

4. Romasevych Yu., Loveikin V. A Novel Multi-Epoch Particle Swarm Optimization Technique. Cybernetics and Information Technologies, 2018, vol. 18(3), pp. 62-74.

5. Romasevych Yu., Loveikin V., Usenko S. PI-controller tuning optimization via PSO-based technique. Przegląd elektrotechniczny, 95, NR 7/2019, pp. 33-37.

6. Ziegler J.G., Nichols N.B. Optimum Settings for Automatic Controllers. Transaction of the ASME, 1942, vol. 64, pp. 759-768.

7. Åström K.J., Hägglund T., PID Controllers: Theory, Design and Tuning, Instrument Society of America NC.: Research Triangle Park, 2 edition, 1995.

8. Åström K.J., Hägglund T. Revisiting the Ziegler-Nichols step response method for PID control. Journal of Process Control, 2004, vol. 14, pp. 635-650.

9. Chien K.L., Hrones J.A., Reswick J.B. On the automatic control of generalized passive systems. Transaction of the ASME, 1952, vol. 74, No. 2, pp. 175-185.

10. Luyben W.L, Luyben M.L. Essentials of Process Control. 1997. McGraw-Hill. 594 p.

11. Romasevych Y., Loveikin V., Dudnyk A., Loveikin Y. Optimal constrained PI-controllers tuning for real-world plants. 2020 IEEE KhPI Week on Advanced Technology, KhPI Week 2020 - Conference Proceedings, 2020, pp. 47–52, 9250108

CHAPTER 3. ARTIFICIAL NEURAL NETWORKS AS OPTIMAL CONTROLLERS

One of the current trends in the development of the theory of automatic control is connected with the application of artificial neural networks. This trend is caused by the fact that the latter may be used to develop automatic control systems for quite complex (often non-linear) plants. In this case, all or some components of the phase-vector of the plant, their higher time derivatives, all or some components of control (neural network output) may be under constraints.

Additionally, the problem of synthesis of the controller, requires that a certain criterion (or set of criteria) should be minimized. These requirements in many cases complicate the problems of synthesis of the controller.

The artificial neural network can be considered as a universal approximator [1], so the trained neural network can satisfy all (or almost all) conditions of the problem of the automatic controller synthesis.

Thus, the initial problem is reduced to the problem of training the artificial neural network. The latter is considered as an optimization problem.

Most commonly the method of backpropagation (BP) of the error [2] and the method of stochastic gradient [2] are used to train the artificial neural network. It is known that gradient methods in the process of finding a solution of a problem may get stuck in the local extremums of the loss function. This reduces the effectiveness of their application.

Metaheuristic (non-gradient) optimization methods are also used to train artificial neural networks. It should be noted that with a small number of parameters that need to be found (weights and biases of network neurons) it is advisable to use these particular methods.

However, with a large number of parameters (thousands or even millions of parameters) metaheuristic optimization methods in terms of the number of calculations becomes quite expensive. Therefore, metaheuristic approaches should be used in training of the artificial neural networks, including in the field of automatic control, because the number of adjustable parameters of neurocontrollers is relatively small.

Thus, there is a need to develop the method of training the artificial neural networks that may be used to control technological processes and that would meet the requirements of minimizing one or more optimization criteria and provide the constraints imposed on the process.

3.1 Analysis of the scientific and applied publications on the topic of the study

3.1.1 The quantitative analysis of the publications

In order to conduct a quantitative analysis of scientific and technical documents on the subject of the study, we use data from scientometric databases Scopus, Web of Science, Google Scholar, as well as data provided by the website of German Patent and Trade Mark Office [3].

In order to perform the analysis in these databases, queries were made for the word „neurocontroller”, the phrase „neural network in control” and the phrase „neural network PSO” (neural network with the method of particle swarm optimization). Such inquiries allow us to establish the main trends in the application of artificial neural networks in the field of automatic control, their training and application for these problems of the method of particle swarm optimization (PSO), and its modifications.

All data obtained correspond to the date of the request February 5, 2020.

When working with the scientometric database Scopus, the queries were performed by the title of the article, a brief description, and keywords. As a result of the review of the number of publications on the generated queries, the obtained data were listed in Table 3.1-3.4.

Table 3.1 Number of publications on the study topic in the scientometric database Scopus by year

Year										Total in the database
2020	2019	2018	2017	2016	2015	2014	2013	2012	2011	
The query for the word „neurocontroller”										
3	11	4	8	9	11	11	12	12	16	507
The query for the phrase „neural network in control”										
967	8857	7763	6244	5085	4697	4525	4403	4235	4423	51199
The query for the phrase „neural network PSO”										
155	978	786	618	570	494	494	494	474	424	7185

Table 3.2 Top 10 countries by number of the publications in the scientometric database Scopus during 2020-2011

The query for the word „neurocontroller”										
Country	USA	China	Ukraine	Germany	Poland	Mexico	Russia	India	Canada	Estonia
1	2	3	4	5	6	7	8	9	10	11
Number of publications	17	11	8	6	6	5	5	4	3	3
The query for the phrase „neural network in control”										
Country	China	USA	India	UK	Germany	Canada	Iran	Italy	Japan	South Korea
Number of publications	17122	9131	3288	2757	2414	1850	1641	1498	1478	1366

Table 3.2 continuation

1	2	3	4	5	6	7	8	9	10	11
The query for the phrase „neural network PSO”										
Country	China	India	Iran	USA	Taiwan	Malaysia	Japan	UK	Australia	Turkey
Number of publications	3386	834	581	409	307	241	184	176	166	133

Table 3.3 Number of the publications on the topic in the scientometric database Web of Science by year

Year										Total in the database
2020	2019	2018	2017	2016	2015	2014	2013	2012	2011	
Query for the word "neurocontroller"										
8	6	5	7	10	8	9	7	4	9	367
1	1	1	4	3	1	-	3	3	2	126
Query for phrase "neural network in control"										
7441	7278	6123	5133	4328	3823	3371	3046	2841	2462	72412
84	88	83	63	65	50	53	47	49	53	1514
Query for phrase "neural network PSO"										
625	558	453	404	381	338	277	244	250	185	4728
32	34	29	35	33	34	14	29	19	21	427

Table 3.3 shows the data obtained when processing queries in the scientometric database Web of Science. The top rows of Table 3.3 correspond to the topic of the query, and the bottom – to the name of the query.

The query „neurocontroller” in the scientometric database Google Scholar for 2020-2011 allows access to about 10300 documents. During the same period, the number of publications available on query „neural network in control” is about 1350000 documents. The query for the phrase „neural network PSO” in the Google Scholar database provides access to 24,200 documents that were published during 2020-2011 years.

The query on the website of the German Patent and Trade Mark Office was executed according to patent classes G05B13/027 „ Adaptive control systems, i.e. systems automatically adjusting themselves to have a performance which is optimum according to some preassigned criterion electric the criterion being a learning criterion using neural networks only” and G06N3/02 „Computer systems based on biological models using neural network models”.

Table 3.4 Number of the patents obtained during 2020-2011 under classes G05B13/027 and G06N3/02

Patent class	Year									
	2020	2019	2018	2017	2016	2015	2014	2013	2012	2011
G05B13/027	702	546	296	168	82	63	38	67	58	48
G06N3/02	4481	3091	2006	1220	723	809	589	422	376	306

Thus, the number of the scientific documents that are placed in scientometric databases Scopus, Web of Science, Google Scholar, and patents obtained over the past ten years in the world, as well as trends in their number shows a significant interest of the scientific community in developing and using technologies which are based on the capabilities of the artificial neural networks.

3.1.2 The content analysis of the publications

The first practical application of the PSO method for problems of training of the artificial neural networks was proposed together with the development of the method itself [4] in 1995. In this subsection, we will analyze some scientific papers related to the training of the neural networks using the PSO method or its modifications.

In the paper [5], a thorough analysis of modifications of the PSO method and the problems in which they were applied are given. In particular, the authors cited the problems in which PSO modifications were used to train the artificial neural networks. For example, Zhang and Wu in the paper [6] proposed the modification of the adaptive chaotic method (ACPSO) to determine the weights and biases of a two-layer (by the number of hidden layers) neural network of direct propagation. The latter was used to develop a hybrid classifier of plant images.

The authors of the paper [7] used the PSO method to adjust the weights of the neurocontroller, which controlled the mobile robot in real time.

In the paper [8], three different types of neural networks were constructed. Here it has been shown that the use of the neural networks, the parameters of which are selected by using PSO method and genetic algorithm (GA), have much better prediction characteristics than individual models or linear combinations of models.

In the paper [9] the improved PID controller for the problem of gas pressure control is presented. The developed algorithm, which is based on the neural network on radial basis functions and the PSO method, was used to adjust the parameters of the PID-controller. The theoretical analysis and simulation of the controller operation performed by the authors showed a reduction of the settling time and an improvement of the control in the setpoint monitoring mode.

In the paper [10], Perng and other authors proposed the combination of the PSO method and the artificial neural network based on radial basis functions to determine the optimal operating point of the wind turbine PID controller and to identify the stability area of the automatic system in the parameter space.

Mandal and others in the paper [11] solved the problem of short-term forecasting of the energy production by using the windmills based on meteorological information. The paper presents aggregate forecasts for wind farms using an intelligent algorithm based on wavelet filtering of data and the artificial neural network, which was optimized by using the PSO algorithm.

The authors of the paper [12] used the adaptive method of variant of the PSO – ACPSO – to train the artificial neural network. The latter acted as a classifier of brain images obtained during magnetic resonance imaging. The accuracy of the classifier based on a sample of 160 images was 98.75%.

In the paper [13], the PSO algorithm was used to supplement the BP method, which is widely used to train artificial neural networks. This training strategy was used to identify a nonlinear system of the bioreactor of yeast fermentation. In addition, it has been shown that this approach is better than BP network training.

Saraswathi and other researchers in the paper [14] used the combination of the GA and PSO methods in combination with the neural network for the problem of gene selection and cancer classification.

A new hybrid model combining the artificial neural network and the chaotic PSO method was proposed in the paper [15] to increase the prediction accuracy. In the paper [16], the modification of PSO without configurable parameters for the problem of training of the artificial neural network is proposed.

In the paper [17], Asadnia and other scientists used an improved PSO method to train the artificial neural network to predict the water level of the Heshui watershed (Gansu Province, China). The results presented in the paper showed a fairly good forecast of low and high water levels in the area in comparasion with other models.

In the paper [18], Das and other researchers used the artificial neural network with the PSO method to solve the problem of the frequency correction of the data channel. In this paper, PSO was used not only to determine the weights of the neurons but also to obtain the optimal network topology and activation functions of the neurons.

Conradie and other scientists in [19] investigated the possibility of extending standard neurocontrollers in industrial processes by using a PSO-based algorithm. The PSO method was chosen in studies because it allows changing the weight of the neurocontroller when changing the environment without turning it off. The researches of using the nonlinear bioreactor model have shown that the profitability of the process can be significantly improved without destabilizing it.

The PSO method has also been successfully applied to fuzzy neural networks. In particular, in the paper [20] Lin and other authors used the hybrid of the PSO method and the local approximation method, and the multi-elite strategy to improve the training of the Takagi-Sugeno-Kang neural network. This hybrid method has replaced BP. Its application has shown a more accurate and efficient training process of the neural network.

In the paper [21], the comparative analysis of different methods of initialization of the swarm population in the problem of training the neural networks, which acted as classifiers, was conducted. The results obtained on the basis of solving eight test problems showed the advantage of using a logarithmic-logistic sequence for the initialization of new particle positions.

In the paper [22], the hybrid of PSO and Cuckoo Search (CS) methods were developed, which the authors used to train the direct propagation neural network that acted as a classifier. The results obtained on the axial analysis of two test problems of classification showed that the new method coincides faster and has a lower tendency to fall into local minima, so it is recommended for use for training neural networks.

As mentioned above, the standard approach to training an artificial neural network is to use the BP method. Thus, BP and PSO are competitors in this sense. The effectiveness of these methods is the subject of many studies. In particular, in the paper [23] it was found out that for the neural networks of the direct propagation the PSO algorithm in terms of the number of calculations is more efficient than BP. PSO is best for applications that require rapid training. In addition, the effectiveness of the neural network training using BP depends on the choice of initial bias values, while for PSO this factor is not influential. The authors also drew an important conclusion: in the case when the object of the training sample is small, the use of PSO allows to reduce the number of calculations by six times than when using the BP method. In this study, the recurrent neural networks, with similar conclusions, were also investigated.

Similar studies were conducted by Nasser Mohammadi and Seyed Javad Mirabedini. In the paper [24], they cited the main disadvantages of the BP method:

the slow convergence and the ability to reach the local minimum. In contrast, PSO avoids these shortcomings. The experimental results showed that the algorithm based on the Levenberg–Marquardt method (which is essentially one of the types of BP) achieved better network training performance than all other BP algorithms (multilayer perceptrons were used in the study). The comparative analysis of these results with those which were found out during the use of PSO, showed the advantage of the latter method.

In the paper [25], the algorithm was developed for the problem of training the single-layer neural network, which allows the application of optimization methods belonging to the specific class: PSO, Intelligent Water Drops (IWD), Ant Colony Optimization (ACO), Artificial Bee Colony (ABC). Their advantages and disadvantages for problems of this type are established. As for PSO, this method showed the premature and rapid convergence, the impossibility of the convergence in the local search area, and its residual error depends on the application problem for which it is used.

In the paper [26], the authors compared the methods of BP and PSO to adjust the weights of the neural network-classifier. The results of the study did not allow the authors to give an unambiguous answer to the question: which of the algorithms is best for the problems of training the neural networks. However, the data obtained by the authors suggested that PSO provides better training accuracy for linear problems. In addition, the authors argue [27] that the advantage of PSO over BP is that it allows to change other network parameters (for example, the activation function of the neuron), and BP does not have this capability.

In the paper [28], the authors proposed the approach that involves using the ACO method to determine the architecture of the neural network and then using the PSO method to perform its training. In the paper [29], a hybrid of the PSO method and another method (simulated annealing), was used to determine the weights and biases of the neural network.

In other works, such as the paper [30], the adjustable elements in the problem of the neural network training were its architecture, activation function of the neuron, and set of weights. In this paper, the authors proposed a new model of the PSO algorithm (NMPSO), while in the paper [31] the same authors solved the problem by using the method of differential evolution.

3.2 The development of a method for training the artificial neural networks

The first stage in the study is to specify the class of networks for which the calculations are performed. We will consider the feedforward neural networks. In the general case, there are no constraints on the dimensionality of the input and output vector of the network, the number of hidden layers, and the number of neurons in the network. In addition, there are no constraints on the type of the activation function of neurons, except in some cases of the output layer: for example, for many automatic control problems, the values of the components of the output vector may include both negative and positive values. In this case, the ReLu-type activation functions do not meet this requirement and other activation functions (e.g., sigmoid) should be used. There is also no requirement for the differentiation of the activation functions. In the research, we will use the model of the neuron, which is presented in Fig. 3.1.

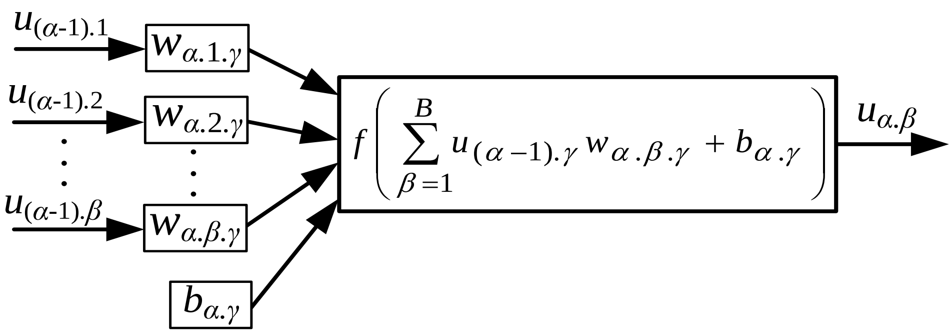

$u_{(\alpha\text{-}1).1}$
$w_{\alpha.1.\gamma}$
$u_{(\alpha\text{-}1).2}$
$w_{\alpha.2.\gamma}$
$u_{(\alpha\text{-}1).\beta}$
$w_{\alpha.\beta.\gamma}$
$b_{\alpha.\gamma}$
$f\left(\sum_{\beta=1}^{B} u_{(\alpha-1).\gamma} w_{\alpha.\beta.\gamma} + b_{\alpha.\gamma}\right)$
$u_{\alpha.\beta}$

Fig. 3.1 The graphical representation of the artificial neuron model

The mathematical model of the neuron is described by the following expression:

$$u_{\alpha.\gamma} = f\left(\sum_{\beta=1}^{B} u_{(\alpha-1).\gamma} \cdot w_{\alpha.\beta.\gamma} + b_{\alpha.\gamma}\right),\ \alpha \in (\overline{2,\ A}),\ \beta \in (\overline{1,\ B}),\ \gamma \in (\overline{1,\ \Gamma}), \tag{3.1}$$

where $w_{\alpha.\beta.\gamma}$ is the weight of the synapse of the γ-th neuron in the α-th layer, which transmits a signal from the β-th neuron of the previous (i.e. from (α-1)-th) layer; A is the number of layers of the neural network, including input and output layers; B is the number of neurons in the previous layer; Γ is the number of neurons in the current layer; $b_{\alpha.\gamma}$ is the bias (signal shift) of the γ-th neuron in the α-th layer; f is the activation function of the neuron (without reducing the generality, we assume that the activation functions for all neurons are the same); $u_{\alpha.\gamma}$ is the output signal of the γ-th neuron in the α-th layer; $u_{(\alpha-1).\beta}$ is the output signal of the β-th neuron in the (α-1)-th hidden layer. The general structure of the artificial neural network is shown in Fig. 3.2.

Given the expression (3.1) and the structure shown in Fig. 3.2, we can write the following dependencies, which describe the components of the output vector Y of the network:

$$y_m = u_{A.m},\ m \in (\overline{1,\ M}). \tag{3.2}$$

The problem of training the artificial neural network is to determine such values of the weights of the synapses $w_{\alpha.\beta.\gamma}$ and biases $b_{\alpha.\gamma}$, at which a certain criterion reaches an extremum (type of extremum: minimum or maximum is determined in the statement of the problem). The essence of this criterion will be disclosed below.

Next, we determine the number of unknown parameters of the neural network that need to be determined during its training. For the network, the structure of which is shown in Fig. 3.2, for this purpose it is necessary to use the following formula:

$$V = V_w + V_b = \left((N+M)\Gamma + (A-3)\Gamma^2\right) + \left((A-2)\Gamma + M\right), \tag{3.3}$$

where V_w and V_b are the numbers of weights of the synapses and the number of biases, respectively.

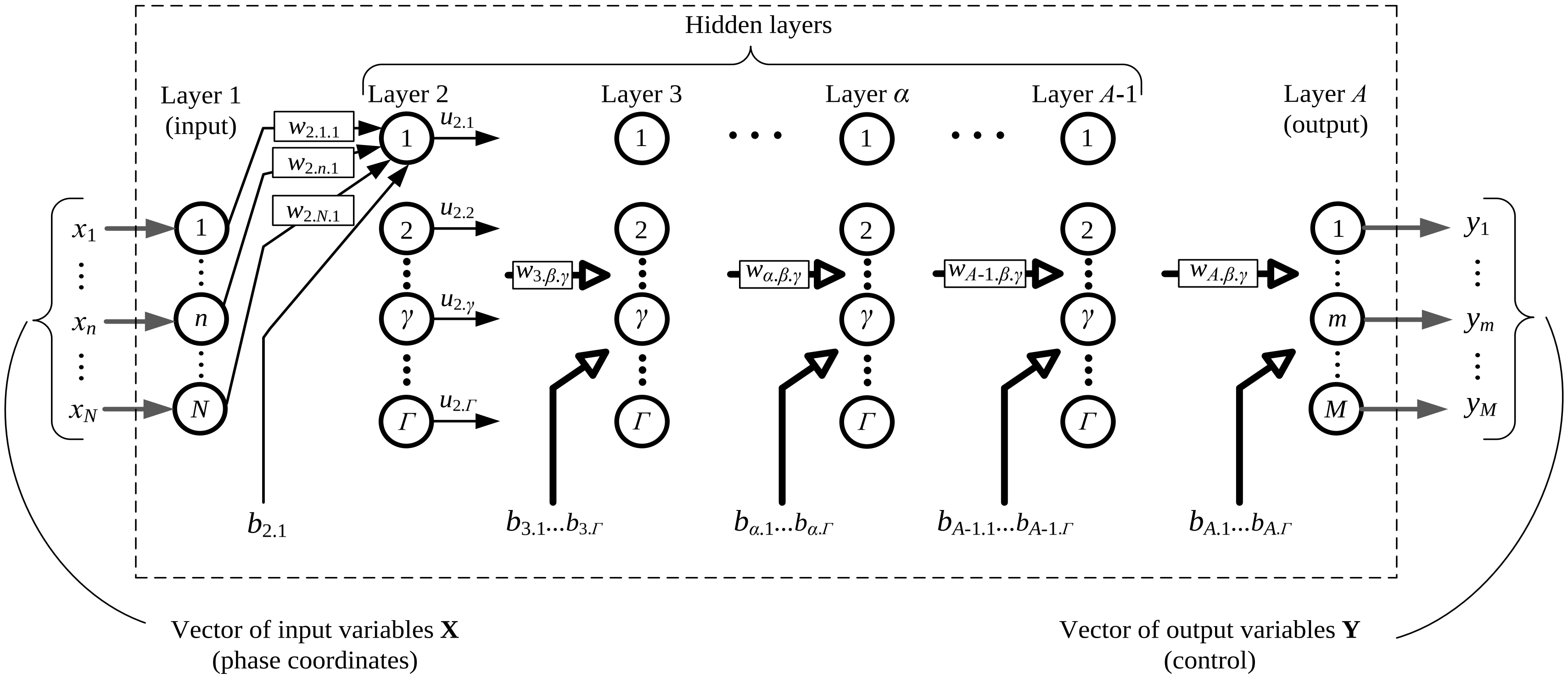


Fig. 3.2 The general structure of the artificial neural network*

* the input (first) layer of the artificial neural network does not perform the signal conversion. It only propagates the signals (components) of the input vector X to the hidden layers of the network. In Fig. 3.2 the dimension of the input vector X is denoted by *N*, and the output Y by *M*. The phase coordinates in the control problems, as a rule, correspond to the control error and its higher time derivatives. The components of the control vector correspond to the signals for actuators of the automatic control system.

Note that expression (3.3) is valid only for the case when the number of neurons in the hidden layers is the same, which, however, does not reduce the generality of the method of training artificial neural networks, which will be described below.

In order to train the neural network, it is necessary to choose the most suitable among the known paradigm: supervised learning, unsupervised learning, reinforcement learning. In general, the choice of paradigm should be based on available learning data. If one knows the training sample (the set of training pairs „input-output”, which reflect the dynamics of the plant), it is necessary to choose the supervised learning paradigm. If the mathematical model of the plant is known (for example, its transfer function), it is necessary to choose the reinforcement learning paradigm. In this case, the agent is the neural network and the environment is the plant. This is the paradigm that will be used in the current study. Hereafter, the neural network that will act as a controller will be called a neurocontroller. In the general case, the plant is a MIMO system (multiple input multiple output). For the linear plants, their mathematical model can be described by using the matrix transfer function G:

$$\mathrm{G} = \begin{bmatrix} G_{1.1} & \cdots & G_{1.m} & \cdots & G_{1.M} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ G_{n.1} & \cdots & G_{n.m} & \cdots & G_{n.M} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ G_{N.1} & \cdots & G_{N.m} & \cdots & G_{N.M} \end{bmatrix}, \tag{3.4}$$

where $G_{n.m}$ is the transfer function of the m-th component of the input vector of the plant Y (the neurocontroller output vector) to the n-th component of the output vector of the plant X (the neurocontroller input vector) (Fig. 3.3).

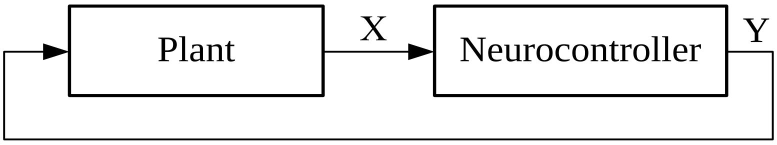


Fig. 3.3 Relationships between the plant and the neurocontroller

For the linear and nonlinear plants, their dynamics can also be described using the differential equations system:

$$\dot{\mathrm{X}} = \mathrm{F}(\mathrm{X},\ \mathrm{Y}), \tag{3.5}$$

where F is a nonlinear vector function. The dot above the symbol means the derivatives by time.

The key concept of this paradigm is the reward, which is the measure of the network quality approximation to a predetermined ideal. The trained network, interacting with the environment, receives the reward, and the training of the network is to select such unknown values of $w_{\alpha.\beta.\gamma}$ and $b_{\alpha.\gamma}$ at which reward will be maximum. In this study, we will not use this concept by replacing it with the concept of „optimization criterion" and noting that there is an inverse relationship between them.

In order to describe the optimization criterion that will be used to train the neurocontroller, we present a statement of the problem of the neurocontroller optimization:

$$\begin{cases} \mathrm{G} = \dfrac{\mathrm{Y}}{\mathrm{X}}; \\ \mathrm{Y} = \mathrm{P}(\mathrm{X}); \\ \lim\limits_{t\to\infty} \mathrm{X} = 0; \\ f_{\lambda}(\mathrm{X},\ \mathrm{Y}) \le F_{\lambda},\ \lambda = (\overline{1,\ L}); \\ Cr_{\kappa} \to \min,\ \kappa \in (\overline{1,\ K}), \end{cases} \tag{3.6}$$

where P is the matrix of nonlinear operators of the neurocontroller, which corresponds to the transformation of the input vector X into the output Y; Cr_{κ} is the κ-th criterion of minimization (without reducing the generality of the problem, we assume that all criteria reflect the undesirable characteristics of the controlled process and therefore should be minimized); K is the total number of optimization criteria; f_{λ} is the λ-th function, which in the general case depends on the vectors X and Y; F_{λ} is the limit value of the λ-th function; L is the total number of constraints in the optimization problem. The first line of the system (3.6) reflects the mathematical model of the plant, the second reflects the neurocontroller, the third one – reflects the requirement of control stability, the fourth reflects the constraints imposed on the control process, and the fifth reflects the set of criteria to be minimized. Note that the statement part of the problem is similar to that used in the study [32]. This analogy makes it possible to perform the transformation of the system (3.6) and replace it with only one criterion, which will be used in the neurocontroller training process. This criterion is described as follows:

$$Cr(\mathrm{W},\mathrm{B}) = Ter + Cr_{ineq} + CC \to \min, \tag{3.7}$$

where *Ter*, Cr_{ineq}, and *CC* are the criteria: terminal, caused by constrains and complex, respectively. The terminal criterion is described by the following dependence:

$$Ter = \begin{cases} 0, \ \ if \ \ x_1(T) < v_1 \wedge \ldots \ x_n(T) < v_n \wedge \ldots x_N(T) < v_N; \\ \delta_T \left\| \mathrm{X}(T) \right\|, \ if \ \ \ x_1(T) < v_1 \vee \ldots \ x_n(T) < v_n \vee \ldots x_N(T) < v_N, \end{cases} \tag{3.8}$$

where T is the moment when the stability conditions are met; v_n is the small allowable value of the n-th component of the vector X; δ_T is the terminal coefficient, which reflects the importance of achieving stability conditions. The absolute minimum of criterion (3.8) is zero: when it is reached, the requirements for the stability of the control process are met. The criterion caused by the constraints may be written as follows:

$$Cr_{ineq} = \begin{cases} 0, \ if \ \ F_\lambda - f_\lambda(\mathrm{X},\ \mathrm{Y}) \ge 0; \\ \delta_p \sum\limits_{y=1}^{Y} (F_y - f_y(\mathrm{X},\ \mathrm{Y})), \ if \ \ F_1 - f_1(\mathrm{X},\ \mathrm{Y}) < 0 \vee \\ \vee F_2 - f_2(\mathrm{X},\ \mathrm{Y}) < 0 \ldots \vee F_L - f_L(\mathrm{X},\ \mathrm{Y}) < 0, \end{cases} \tag{3.9}$$

where y is the index of the unfulfilled inequality in the requirements (3.6); Y is the total number of the unfulfilled inequalities; δ_p is the penalty coefficient. According to the study [32], the essence of criterion (3.9) is that if at least one inequality is not fulfilled, the criterion will have a significant value. On the other hand, if all inequalities are met, the Cr_{ineq} criterion becomes zero and does not affect the further minimization of the generalized criterion (3.7).

The set of optimization criteria Cr_κ can be replaced by one, which is described by the following expression [32]:

$$CC = \sum_{\kappa=1}^{K} w_\kappa g_\kappa Cr_\kappa \to \min, \tag{3.10}$$

where w_κ is the κ-th weight coefficient, which reflects the importance of the criterion Cr_κ; g_κ is the coefficient of the κ-th criterion, which reduces the product $Cr_\kappa g_\kappa$ to the dimensionless form. This form of the comprehensive criterion allows on a compromise basis to perform the optimization criteria convergence of different importance to one criterion. The latter is represented as the linear convolution [32].

The values of the coefficients δ_p and δ_T make it possible to form the required topology of the criterion (3.7). The developed criterion (3.7) allows application of the method for the neurocontroller training. In order to illustrate the developed method, Fig. 3.4 shows its block diagram.

At the initial stage of the algorithm, the constant parameters of the ME-D-PSO method are initialized [32]: the number of iterations *R*, the size of the swarm population *D*, and the acceptable rate *AR* the global best *GB* decreasing. The next step is to initialize the positions and velocities of the swarm particles. The position of each particle will be characterized by a *V*-dimensional vector. The first branch of the block diagram (Fig. 3.4) means checking the condition $r<R$, which indicates the end of all iterations of the algorithm. In case if all iterations have not been performed yet, it is necessary to go to the cycle. The first operation of the body of the cycle is to determine the values of *Cr*, which correspond to the initialized particles’ positions. It is necessary to access the function *Cr*, which receives the input *V*-dimensional vector whose components are the values of the weights $w_{\alpha.\beta.\gamma}$

and biases $b_{\alpha.\gamma}$ of the neurocontroller, and at the output forms a scalar value of Cr. The operation of this function will be discussed later.

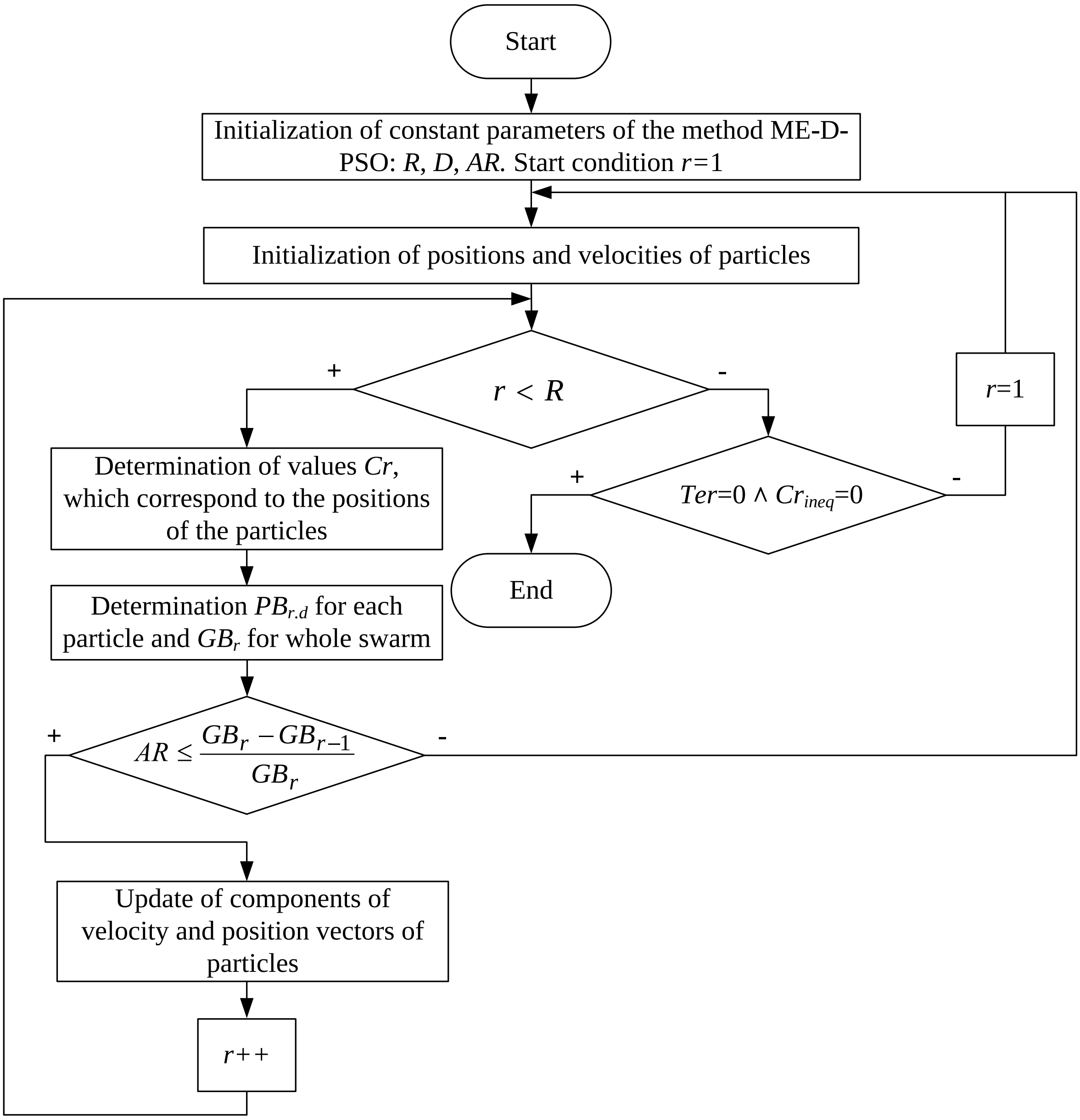

Start
Initialization of constant parameters of the method ME-D-PSO: R, D, AR. Start condition r=1
Initialization of positions and velocities of particles
$r < R$
+
-
r=1
Determination of values Cr, which correspond to the positions of the particles
+
$Ter=0 \wedge Cr_{ineq}=0$
-
End
Determination $PB_{r.d}$ for each particle and GB_r for whole swarm
+
$AR \leq \frac{GB_r - GB_{r-1}}{GB_r}$
-
Update of components of velocity and position vectors of particles
r++

Fig. 3.4 Block diagram illustrating the neurocontroller training method

The personal best $PB_{r.d}$ for each particle and the global best GB_r for the whole swarm should be then calculated (the lower index r in the $PB_{r.d}$ and GB_r records indicates the iteration number at which these values were obtained, and the lower index d in the $PB_{r.d}$ record indicates the number of the particle that has the value of personal best). Branching in the body of the cycle allows testing the effectiveness of the global minimum search. Failure to meet the condition

$AR \leq \frac{GB_r - GB_{r-1}}{GB_r}$ is an indicator of swarm stagnation. In this case, the swarm must be reinitialized (a new epoch of the swarm should be launched [33]). If the swarm effectively searches for the minimum of the *Cr* function, then the velocity vectors and positions of each of the swarm particles are updated. At the end of all iterations of the cycle (i.e. under the condition $r=R$) the condition of stability of control $Ter=0$ and the conditions of constraints satisfaction $Cr_{ineq}=0$ is checked. If these conditions and constraints are not met, the algorithm must be run for re-execution. This is due to the mandatory fulfillment of these conditions. In order to fully understand how the method works, it is necessary to describe how to access the *Cr* function. To do this, in Fig. 3.5 we present its block diagram.

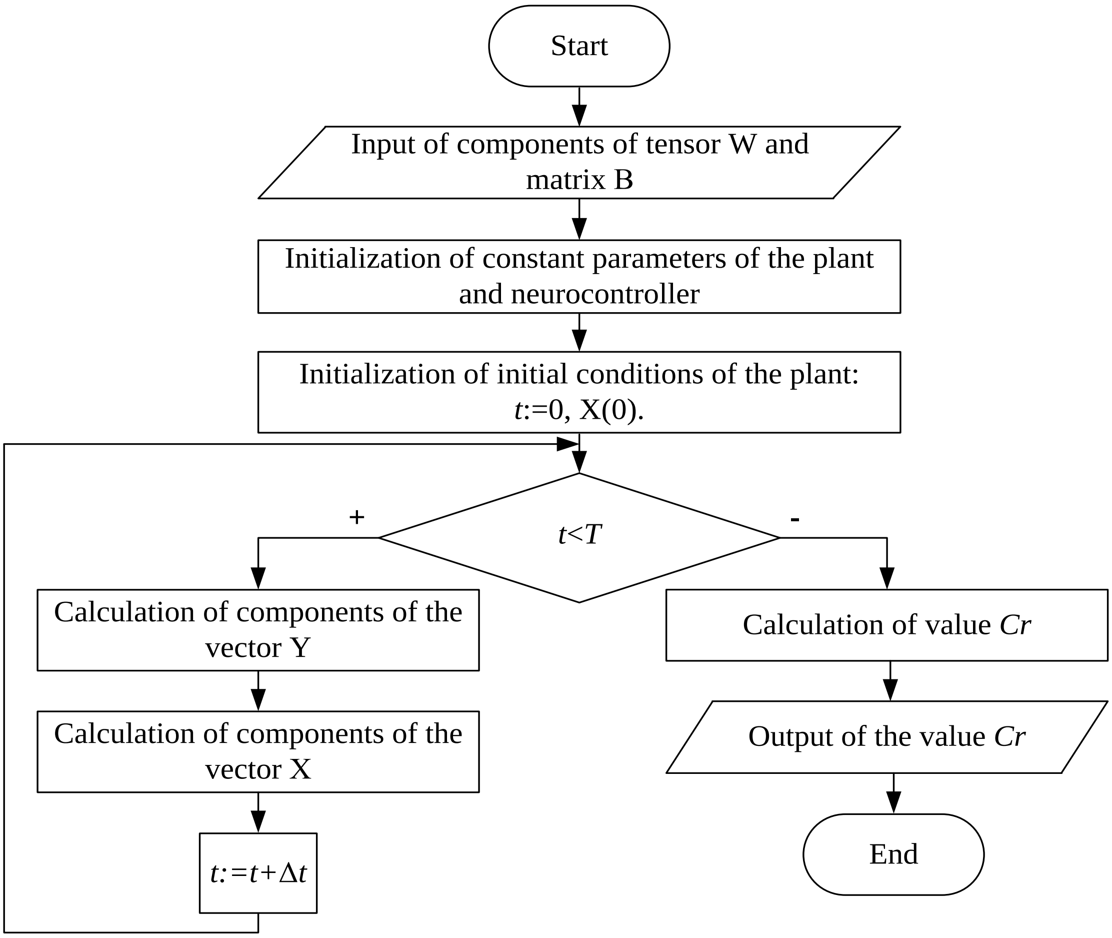

Start
Input of components of tensor W and matrix B
Initialization of constant parameters of the plant and neurocontroller
Initialization of initial conditions of the plant: t:=0, X(0).
t<T
+
-
Calculation of components of the vector Y
Calculation of components of the vector X
t:=t+Δt
Calculation of value Cr
Output of the value Cr
End

Fig. 3.5 Block diagram of the *Cr* function

At the initial stage of the function, the components of the tensor W and the vector B are introduced. Then the initial parameters of the plant (coefficients of the matrix transfer function) and the neurocontroller (for example, the type of activation function), as well as the initial conditions (phase coordinates of the plant and settings at the beginning of the control process) are initialized.

After that, the algorithm goes into the cycle: after passing the body of the cycle, the condition $t<T$ of the end of the simulation of the control process is checked. The components of the vector Y (using the neurocontroller model) and the vector X (using the model of the plant) are calculated in the body of the cycle for a certain time step. When all time steps are performed, the cycle is exited and *Cr* is calculated. At the last stage, *Cr* is gone as an output.

In order to illustrate the application of the developed method, the calculations of the neurocontroller were performed for several plants. These studies are described in the following.

3.3 The results of the neurocontroller operation simulation

For all of the calculations below, the neurocontroller was the artificial feedforward neural network consisting of three layers (one of which was hidden with five neurons). The activation function for all neurons is arctangent (Fig. 3.6).

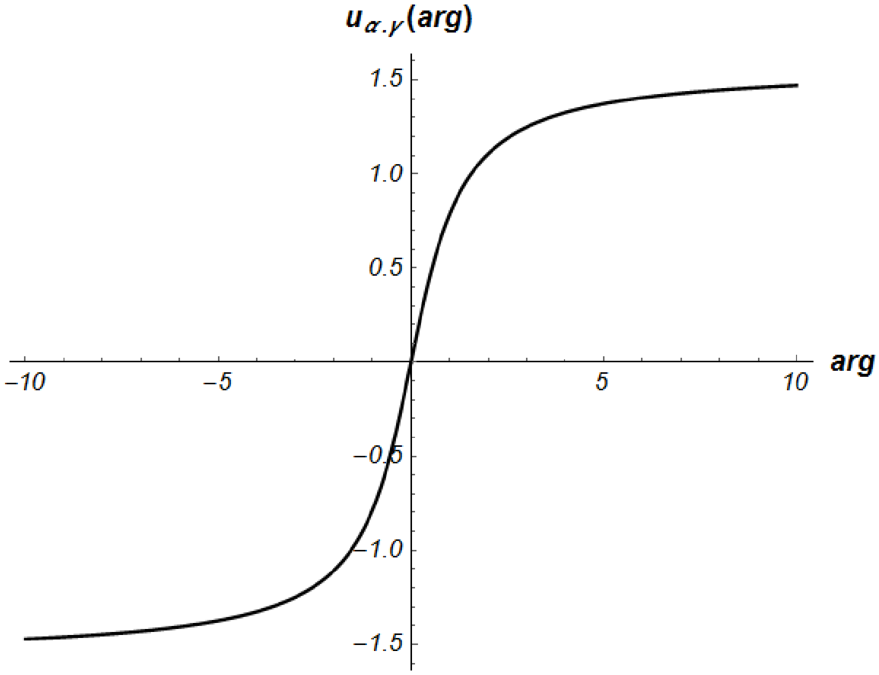


Fig. 3.6 Plots of the neurons network activation function

Hereafter we will consider several options for the synthesis of the optimal neurocontrollers.

3.3.1 Linear MISO plants

At the first stage of the study of the method efficiency we will use some transfer functions of the plants which are used in the papers [34, 35] (Table 3.5).

Table 3.5 Transfer functions of the plants and their features

Transfer function	Features
1	2
Test transfer functions	
$G_1(s)=\frac{1}{s^2-1}$	The unstable system. The transfer function is a model of an inverted pendulum in the position of unstable equilibrium.
$G_2(s)=\frac{\omega_0^2}{(s+1)(s^2+2\xi\omega_0+\omega_0^2)}$, $\omega_0=1.$	The oscillating system. The use of the PID-controllers for the systems of this class with low damping ξ does not allow to obtain the high-quality control. The plants described by these transfer functions are better controlled at significant ω_0. The quality of the control can be improved by using more general structures of the controllers.
Transfer functions of real plants	
$G_3(s)=\frac{1{,}83}{16800s^2+460s+1}$	The grain dryer. The plants of this type are usually described by the second-order models and have a significant time constant.

Table 3.5 continuation

1	2
$G_4(s) = \dfrac{1}{8.20 \cdot 10^{-5} s^2 + 6.17 \cdot 10^{-3} s + 2.30 \cdot 10^{-2}}$	The electric drive of the mobile robot operating in the greenhouse. Such an electromechanical system has extremely fast transients and is described by the second-order model.

The real plants are selected in such a way that their time constants differ by orders of magnitude. This variety of the parameters will allow confirming the method applicability for the different plants, i.e. those whose features and properties differ significantly. In turn, this will confirm the generality of the developed approaches.

For control systems, in which the neural network must be located, the constraints are imposed on zero overshoot. This constraint is presented as follows:

$$\max(x - r) \le 0. \tag{3.11}$$

The optimization criterion in the conducted studies was the following value:

$$CC = \sqrt{\int_0^{t_s} u^2 dt} + \sqrt{\int_0^{t_s} e^2 dt} + t_s \to \min, \tag{3.12}$$

which on a compromise basis reflects the requirement to minimize RMS control, the error, and the settling time. It should be noted that the optimization criterion may be another value. The optimization methodology would not have changed.

In addition, this study did not take into account the importance of the individual criteria, as they depend on the specifics of the problem. The purpose of this study was to show the possibility of applying the developed approach to the optimal neurocontrollers synthesis.

For the plants (Table 3.5) it is set that the allowable error v_0 (for all control problems r=1) is 0.005, and for higher derivatives of the error in time v_1=v_2=0.005. The parameters of the ME-D-PSO method are as follows: the number of iterations R=100, the size of the swarm population D=20, the value of the criterion acceptable rate AR = 0.01.

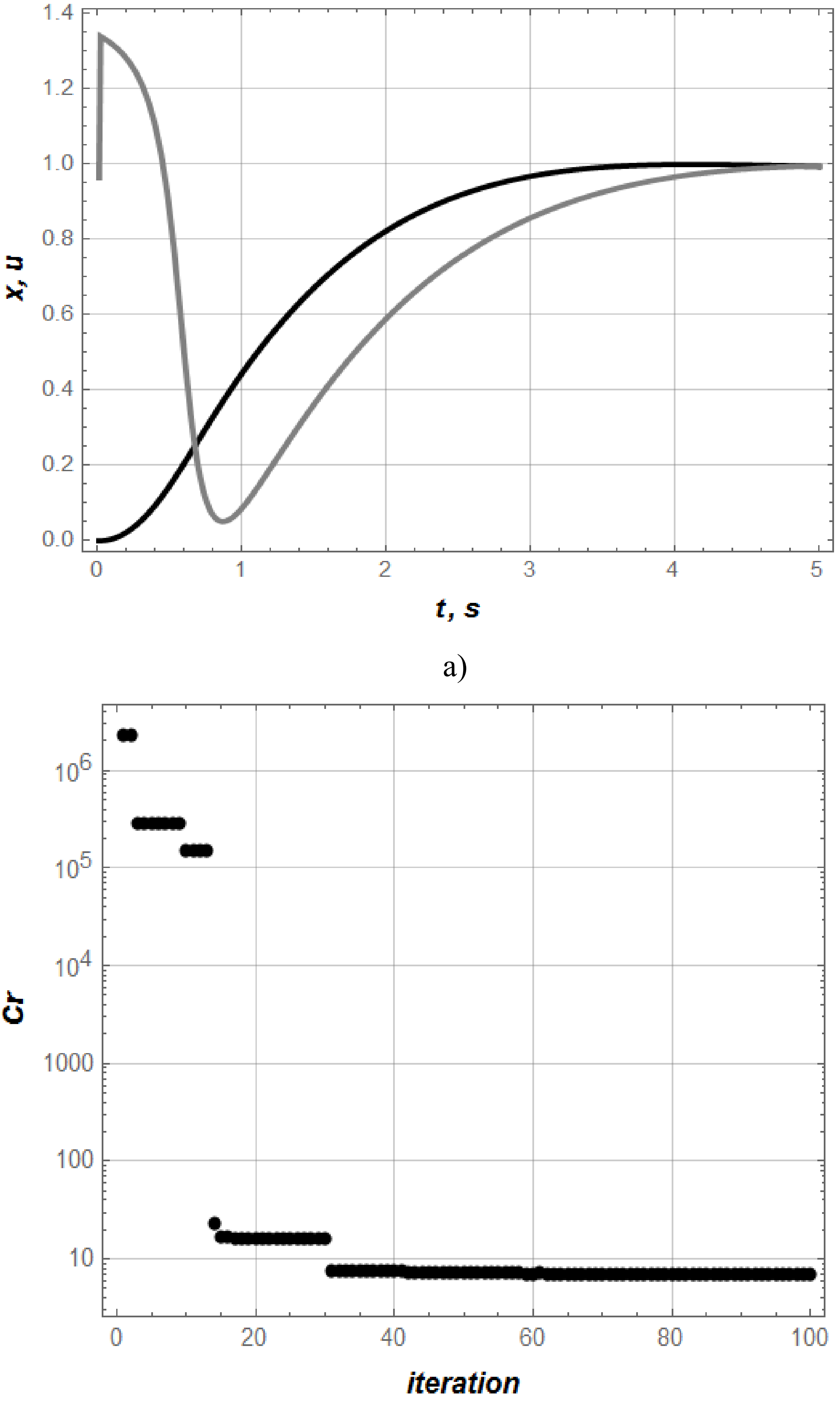
Using expression (3.3), the number of the unknown values to be found out was obtained. For the transfer functions $G_1(s)$, $G_3(s)$, $G_4(s)$ V=21, and for $G_2(s)$ V=26. The search domain for the optimal value of each argument (weights and biases of the neural network) is given by the limits -4 and 4.

As a result of the developed method application, the components of the tensor W and the vector B are obtained, which minimize the criterion (3.12) for all plants. The graphical dependencies are shown to illustrate the obtained results.

Fig. 3.7 shows the plots that correspond to the training process of the neural network and its operation in the control loop of the plant with the transfer function $G_1(s)$ (hereinafter, all plots correspond to the setpoint mode). Note that from Fig. 3.7 (a) it is seen that the constraints (3.11) are met, i.e. the system goes to the setpoint without the overshoot.

Analyzing the plots shown in Fig. 3.7 (b), we can see the topology of the criterion Cr. As indicated in the paper [32], this is a topology of the „pit in the pit" type: the first two iterations are performed in the „upper pit" criterion, the next seventeen iterations – in the „middle pit", and the rest – in the „bottom pit". Since both conditions Ter=0 and Cr_{ineq}=0 were met, there was no need to restart the ME-D-PSO algorithm.

Similar results were obtained for the plant, which is described by the transfer function $G_2(s)$ (Fig. 3.8): the first iteration corresponds to the swarm movement in the „upper pit", from the twenty-third iteration the algorithm „collapses" into the „lower pit" and then performs search for the minimum of the criterion only (3.12).

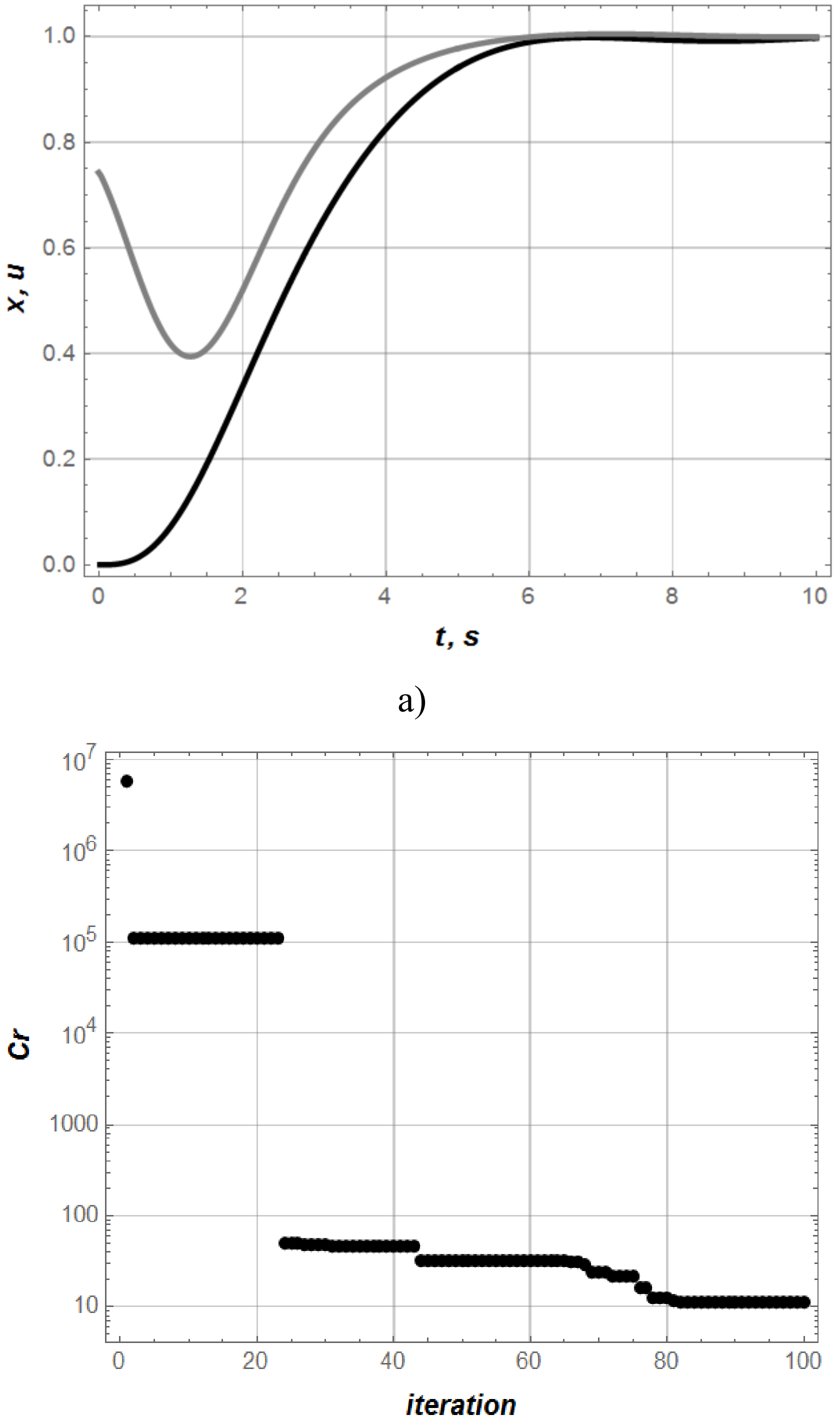


a)

b)

Fig. 3.7 Plots of the dynamics of the controlled variable (black curve) and control (gray curve) of the plant with the transfer function $G_1(s)$ (a) and the generalized criterion reduction during the optimization procedure (b)

In the „bottom pit", the ME-D-PSO algorithm searches for the global minimum of the criterion (3.21), which was accompanied by several re-initializations of the swarm [33].



a)

b)

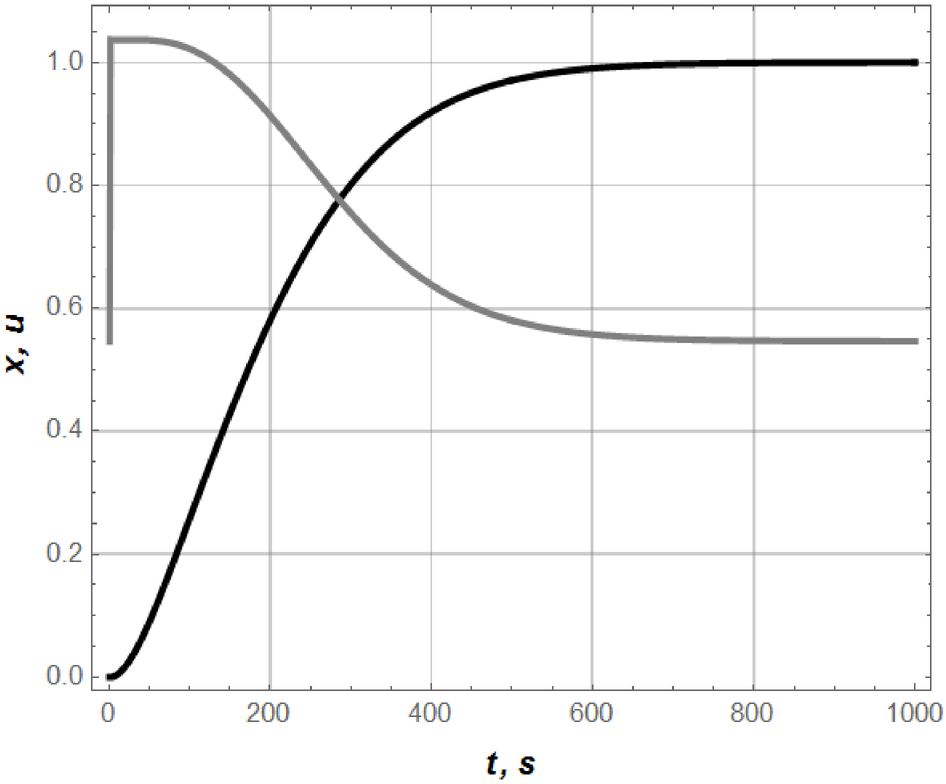
Fig. 3.8 Plots of the dynamics of the controlled variable (black curve) and control (gray curve) of the plant with the transfer function $G_2(s)$ (a) and the generalized criterion reduction during the optimization procedure (b)

From Fig. 3.8 (a) it is seen that the control signal (the neurocontroller output signal) is a smooth function, and the controlled variable has no overshoot, i.e. the condition (3.11) is met.

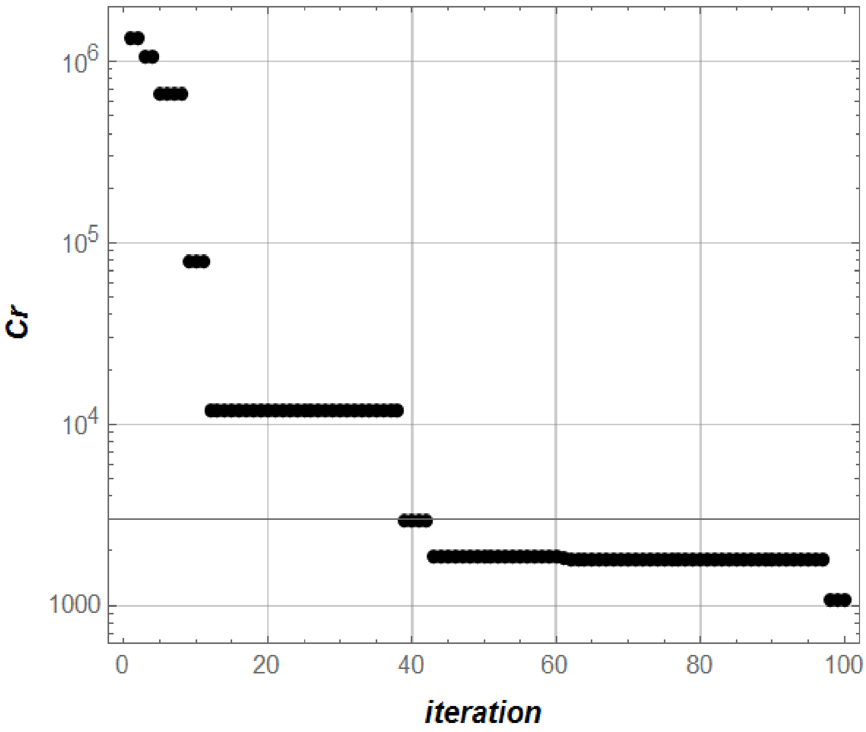
In order to establish the method suitability of training the neural networks for the control of the real plants, the problems were solved for which it was necessary to minimize the value (3.12) with the constraint (3.11) for the plants described by the transfer functions $G_3(s)$ and $G_4(s)$.

As a result of the calculations, the values of the weights and biases, corresponding to the conditions of the problem, were obtained. The obtained results will be presented in the form of graphical dependences (Fig. 3.9 and Fig. 3.10).

The analysis of the dynamics of the temperature setpoint (relative value) reaching the grain dryer shows a smooth, without an overshoot, output to a steady state. The obtained control mode allows minimizing the criterion (3.12).



a)



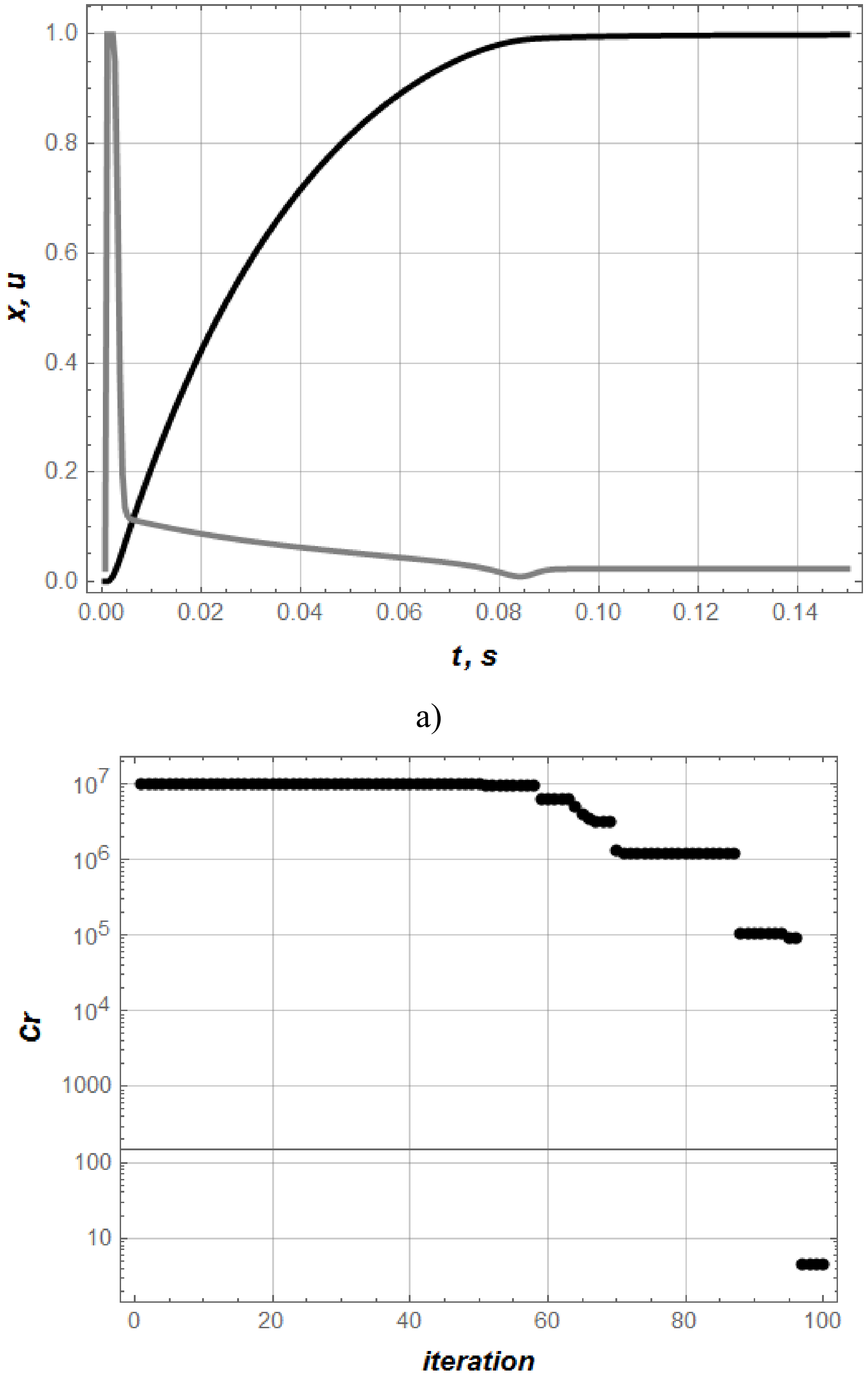
b)

Fig. 3.9 Plots of the dynamics of the controlled variable (black curve) and control (gray curve) of the plant with the transfer function $G_3(s)$ (a) and the generalized criterion reduction during the optimization procedure (b)

Similar dynamics is observed for the electric drive robot, i.e. the system described by the transfer function $G_4(s)$ (Fig. 3.10, a). The settling time for this plant is small (about 0.08 s).

However, as for the algorithm's operation, there are significant differences. For the plant with the transfer function $G_3(s)$, the convergence to the minimum of the criterion (3.12) is typical: first, a significant decrease in the value of the criterion due to the transition from the „upper pit" of the criterion (3.12) to „middle pit" during the first ten iterations; further „collapsing" of the swarm into the „bottom pit" on the thirty-ninth iteration.

For the algorithm that performed the minimization of the criterion (3.12), which corresponds to the parameters of the electric drive robot (described by the transfer function $G_4(s)$), the first fifty-eight iterations were inefficient. In the subsequent iterations, the algorithm was able to find the first local minimum of the criterion, followed by others.



a)

b)

Fig. 3.10 Plots of the dynamics of the controlled variable (black curve) and control (gray curve) of the plant with the transfer function $G_4(s)$ (a) and the generalized criterion reduction during the optimization procedure (b)

This feature was also noticed when the algorithm was restarted: the minimization of the criterion (3.12) was not performed on the first iterations of the algorithm. This feature should be taken into account in such cases by increasing the number of iterations.

The neural network training method was also used for the synthesis of the optimal motion controller of the „crane-load” system (Fig. 3.11).

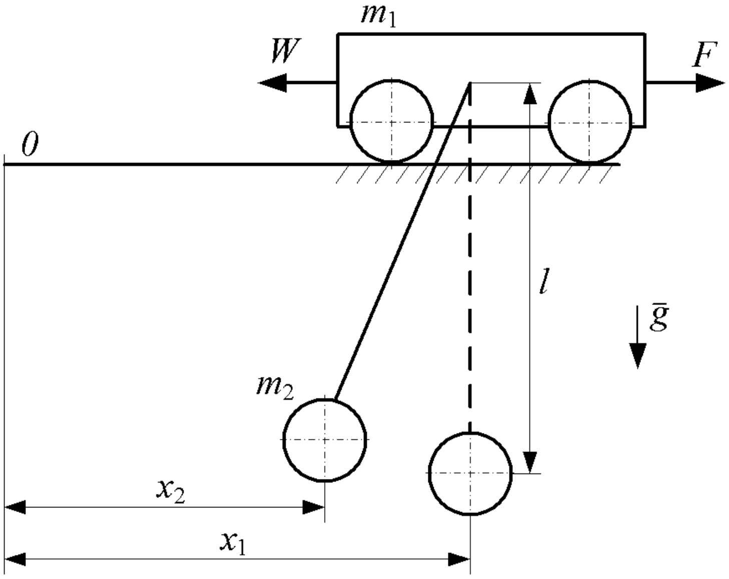


Fig. 3.11 Dynamic movement model of the „crane-load” system

The plant, the dynamic model of which is shown in Fig. 3.11, is described by the following system of differential equations:

$$\begin{cases} x_1 = x_2 + \ddot{x}_2 \dfrac{l}{g}; \\ F - W = m_1 \ddot{x}_1 + m_2 \ddot{x}_2, \end{cases} \qquad (3.13)$$

where m_1 and m_2 are the reduced masses of the crane and the load, respectively; x_1 and x_2 are generalized coordinates of the reduced masses m_1 and m_2, respectively; l is the length of the flexible suspension; F is the driving force; W is the force of the crane resistance of movement.

The neurocontroller operation for this plant is that it is necessary to bring it to a steady speed, and eliminating the oscillations of the load on the flexible suspension and minimize the optimization criterion. The boundary conditions of the system may be written as follows:

$$\begin{cases} x_1(0) = \dot{x}_1(0) = x_2(0) = \dot{x}_2(0) = 0; \\ x_1(t_s) = x_2(t_s); \;\; \dot{x}_1(t_s) = \dot{x}_2(t_s) = v_n, \end{cases} \tag{3.14}$$

where v_n is the nominal speed of the crane.

The constraints imposed on the movement of the system are due to the limited overload capacity of the engine of the crane movement mechanism (limitation on the amount of the driving force) and the limited supply voltage (it causes the constraints on the speed of the driving force change). Both constraints may be written as follows:

$$\begin{cases} F_{\min} \le F \le F_{\max}; \\ \dot{F}_{\min} \le \dot{F} \le \dot{F}_{\max}, \end{cases} \tag{3.15}$$

where $F_{\min}$ and $F_{\max}$ are the maximum and minimum values of the driving forces of the drive, respectively; $\dot{F}_{\min}$ and $\dot{F}_{\max}$ are the maximum and minimum values of the rates of the driving forces change, respectively.

In addition, in the problem the constraint on the crane speed is imposed: it should not exceed the nominal value (i.e. the speed overshoot should be absent):

$$\dot{x}_1 \le v_n. \tag{3.16}$$

The optimization criterion in this problem is a complex one, which on a compromise basis reflects the duration of the control and the root mean square value of the dynamic component of the drive power. It is described by the following dependence:

$$CC = I_{P_{dyn}} = 10^4 t_s + \sqrt{t_s^{-1} \int_0^{t_s} P_{dyn}^2 dt} = 10^4 t_s + \sqrt{t_s^{-1} \int_0^{t_s} \left(F\dot{x}_1\right)^2 dt} =$$

$$= 10^4 t_s + \sqrt{t_s^{-1} \int_0^{t_s} \left(m_1 \ddot{x}_1 \dot{x}_1 + m_2 \ddot{x}_2 \dot{x}_1\right)^2 dt} \to \min, \tag{3.16}$$

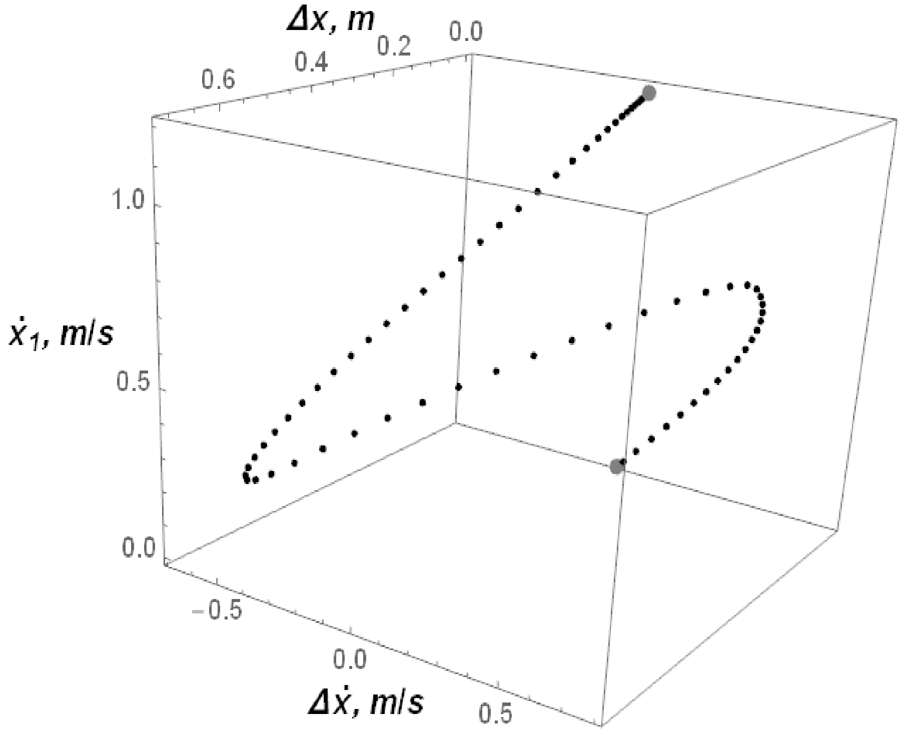
where P_{dyn} is the dynamic component of the drive power.

The system model (3.13) is linear; however, criterion (3.16) is nonlinear. In addition, the calculations are complicated by the presence of the constraints (3.15) and (3.16).

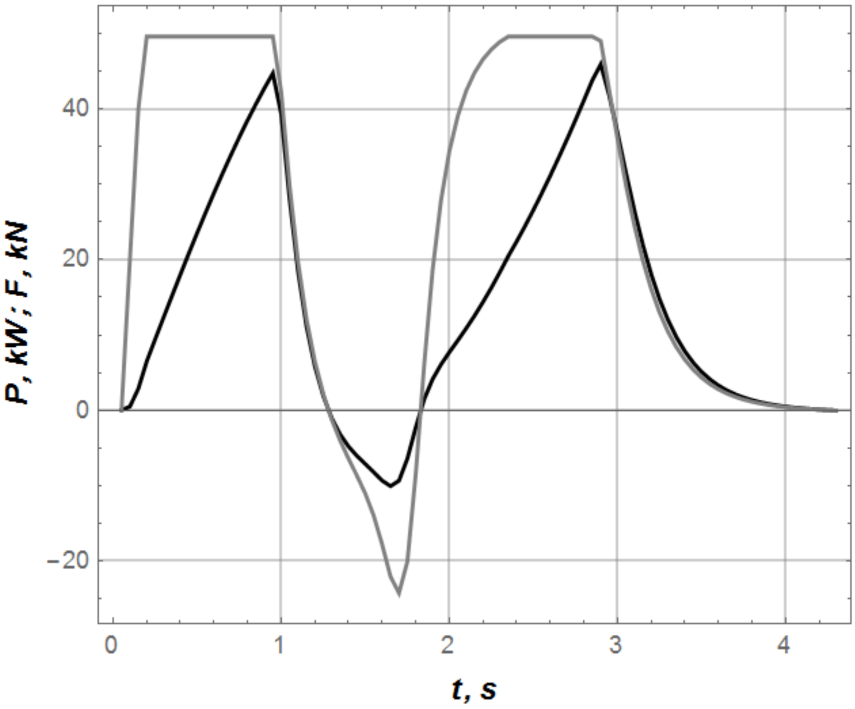
The calculations were performed for the crane with the following parameters: m_1=42000 kg, m_2=25000 kg, l=5 m, v_n=1.2 m/s, $F_{\min}$=-56250 H, $F_{\max}$=56250 H, $\dot{F}_{\min}$=-400 kH/s, $\dot{F}_{\max}$=400 kH/s, v_n=1.2 m/s; W=6573 H.

The application of the neurocontroller training method for this problem allowed us to obtain such values of the components of the tensor W and the vector B, at which all the conditions of the problem (3.13)-(3.16) are met. In order to illustrate the obtained results, the plots of functions were built (Fig. 3.12).

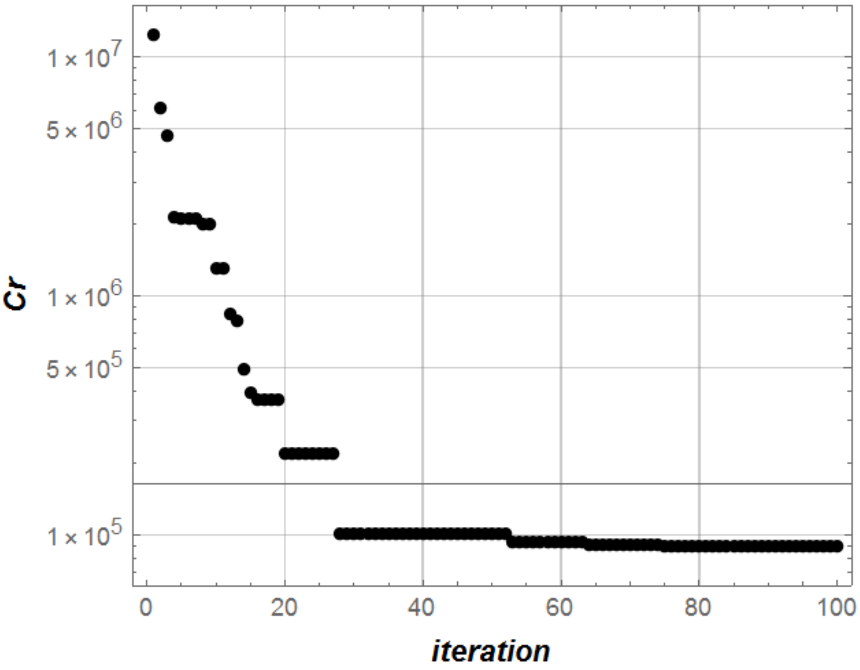
In Fig. 3.12 (a) the initial and final values of the phase vector are shown in gray dots. The phase trajectory of the system begins at the origin and ends at a point that corresponds to the steady motion of the system with velocity v_n. As can be seen from Fig. 3.12 (a) the load oscillations at the end of the crane start are absent, as evidenced by the zero values of Δx and $\Delta\dot{x}$ at the end point of the phase trajectory.



a)



b)



c)

Fig. 3.12 Plots of the dynamics of the control of the plant „crane-load": non-classical phase portrait of the system in the coordinates „the difference between the positions of the crane and load – the difference between the speeds of the crane and load – crane speed" (a); the driving force (gray curve) and the power (black curve) of the crane drive (b); the generalized criterion reduction during the optimization procedure (c)

Thus, the system terminal conditions (3.14) are satisfied. In addition, Fig. 3.12 (a) shows that the magnitude of $\dot{x}_1$ does not exceed the nominal value of the crane speed. Fig. 3.12 (b) shows that the driving force is twice limited and changes its sign.

However, criterion (3.16) also reflects the requirement to minimize the RMS value of the drive dynamic power, which has an impact on the obtained optimal law of the system motion.

The power changes its sign, which is undesirable because it requires the ability to recover the electric energy from the system. In order to prevent such a phenomenon, it is necessary to set an asymmetric constraint on the driving force (3.15), as done in the study [36].

Thus, based on the method of training artificial neural networks, it was possible to synthesize the optimal neurocontrollers for several linear plants, which are characterized by different properties (unstable, oscillatory, with small and large time constants). In addition, the nonlinear constraint problem was solved for the problem of moving the „crane-load” system. All this suggests that the developed method of training the neural networks is an effective tool for the synthesis of the optimal controllers for the linear MISO plants.

In the following, we will show its efficiency for the problems of the neurocontrollers synthesis that are operated in the control loops of the nonlinear MISO plants.

3.3.2 Nonlinear MISO plants

The next step in illustrating the effectiveness of the method of training the neural networks for control problems is involving of the underactuated systems as plants to control. Such systems are nonlinear and therefore the problems of optimal controllers synthesis for them are quite complex.

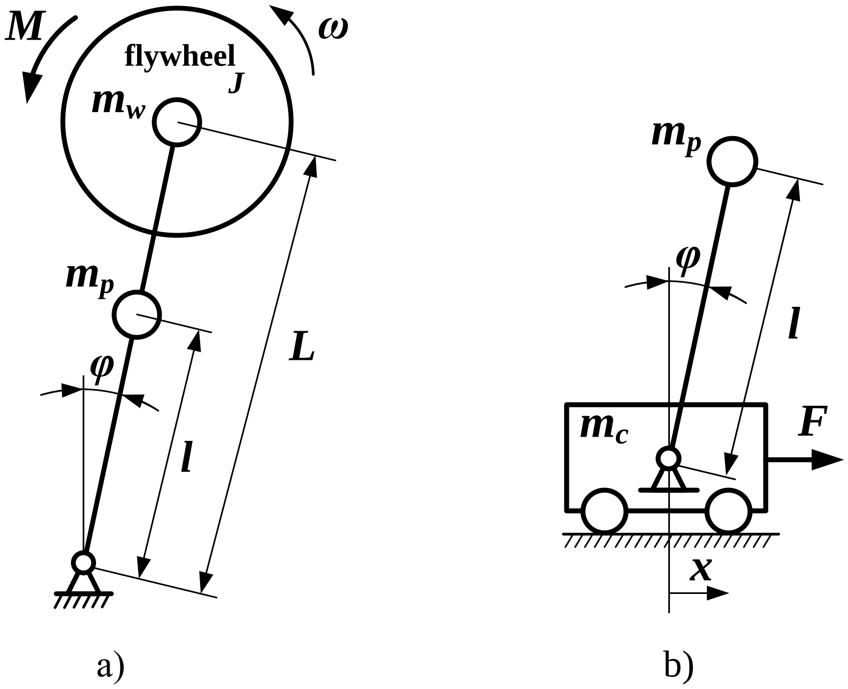
The neurocontrollers synthesis will be performed for the systems whose dynamic models are shown in Fig. 3.13.

The first system is an inverted wheel pendulum or inverted pendulum with an inertial flywheel (Fig. 3.13, a) [37]. The flywheel is driven by the torque generated by the engine (Fig. 3.13, not shown).

The inertial characteristics of the pendulum are represented by the reduced masses m_w and m_w. The inertial properties of the flywheel are represented by the consolidated moment of inertia J.

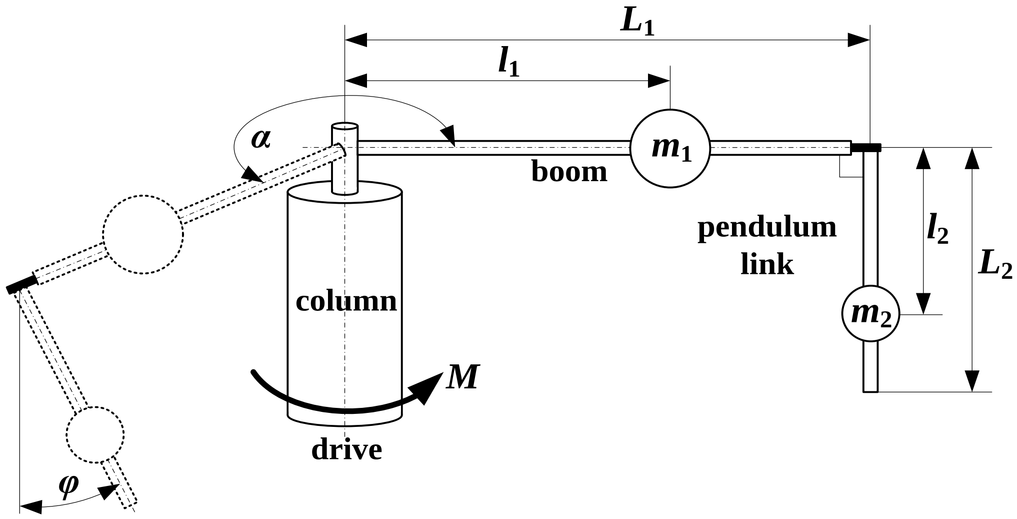
The inverted pendulum (Fig. 3.13, b) is another model from the class of the underactuated nonlinear systems (such a system is also known as „cart-pole”, „inverted pendulum on a cart”) [38]. The cart moves in a horizontal plane, and the pendulum oscillates around the axis that is attached to the cart. The system movement is due to the driving force, which can change the sign. It is created by a drive (usually from a DC motor with PWM, or a stepper motor).

The Furuta pendulum [39] (Fig. 3.13, c) is a vertical column, which is driven by the engine (in Fig. 3.13 is not shown). A boom is attached to the end of the column, and a pendulum link is attached to the end of the boom through a pivot. The masses m_1 and m_2 are located on the boom and the pendulum link, respectively.



a)

b)



c)

Fig. 3.13 The dynamic models of the underactuated systems: a) the inverted pendulum with flywheel; b) the inverted pendulum; c) the Furuta pendulum

The parameters of the underactuated systems determine the properties of their control (stabilization). In order to illustrate the developed method of training the artificial neural networks, we will assume that the system's parameters are constant.

The mathematical model of the first underactuated system (Fig. 3.13, a) is presented below:

$$\begin{cases} \alpha_1 \ddot{\varphi} + J\dot{\omega} = \alpha_2 \sin(\varphi); \\ J(\ddot{\varphi} + \dot{\omega}) = M, \end{cases} \tag{3.17}$$

where φ is the generalized coordinate of the pendulum (the angle of deviation of the pendulum from the vertical); ω is the angular velocity of the pendulum flywheel; g is the acceleration of the free fall; l is the distance from the center of mass of the pendulum to the axis of its rotation; L is the equivalent length of the pendulum; J is the inertia moment of the flywheel and rotor of the electric motor; M is the electromagnetic moment created by the engine; α_1 and α_2 are the coefficients determined from the following expressions: $\alpha_1 = J_p + m_w L^2$, $\alpha_2 = (m_p l + m_w L)g$; m_p is the mass of the pendulum; m_w is the mass of the flywheel and the electric motor; J_p is the moment of the pendulum inertia. The numerical data used in the calculations for the first system are given below: L=2.9·10^{-1} m, l=2.6·10^{-1} m, J=1.5·10^{-3} kgm^2, J_p =4.2·10^{-2} kgm^2, m_p =3.0·10^{-1} kg, m_w =2.8·10^{-1} kg, $M_{\max}$=6.0·10^{-1} Nm (the maximum force of the system drive). The boundary conditions of the system are as follows:

$$\begin{cases} \varphi(0) = \pi;\ \dot{\varphi}(0) = \omega(0) = 0; \\ \varphi(t_S) = \dot{\varphi}(t_S) = \omega(t_S) = 0. \end{cases} \tag{3.18}$$

The initial conditions of the system movement (3.18) mean the state of the rest of the pendulum (in the lower stable position) and the flywheel. The final (desired) conditions of the system mean the state of the rest of the pendulum in the upper unstable equilibrium position and the state of the rest of the flywheel.

The inverted pendulum, the model of which is shown in Fig. 3.13 (b), is described by the system of the following nonlinear differential equations:

$$\begin{cases} (m_c + m_p)\ddot{x} + m_p l(\ddot{\varphi}\cos\varphi - \dot{\varphi}^2 \sin\varphi) = F; \\ \ddot{x}\cos\varphi + l\ddot{\varphi} + g\sin\varphi = 0, \end{cases} \tag{3.19}$$

where x is the generalized coordinate of the cart; φ is the generalized coordinate of the pendulum; l is the equivalent length of the pendulum; m_p and m_c are the reduced masses of load and cart, respectively; F is the driving force acting on the cart. Numerical data for the second underactuated system, which are used in the calculations, are given below: l=5.0·10^{-1} m, m_p =3.0·10^{-1} kg, m_c=2.0·10^{-1} kg. In addition, the maximum drive force is limited by 5.0 H.

The boundary conditions of this system are recorded as follows:

$$\begin{cases} \varphi(0) = \pi;\ \dot{\varphi}(0) = x(0) = \dot{x}(0) = 0; \\ \varphi(t_S) = \dot{\varphi}(t_S) = x(t_S) = \dot{x}(t_S) = 0. \end{cases} \tag{3.20}$$

The initial conditions of the system movement (3.20) correspond to the rest of the cart and the position of the pendulum in the lower stable position of the equilibrium. The final (desired) conditions of the system mean the state of the rest of the pendulum in the upper unstable equilibrium position, and the position of the cart must locate in its initial position. In addition, the movement of the cart is constrained with the following expression:

$$\left| x(t) \right| \le x_{\max}, \tag{3.21}$$

where $x_{\max}$ is the maximum permissible deviation of the cart from the initial position. This constraint is justified by the fact that the cart moves on rails, which in the conditions of real installation have a limited length. In the calculations below, the constraint $x_{\max}$=0.5 m was used.

The mathematical model of the third underactuated system (the Furuta pendulum) is represented by the nonlinear differential equations system:

$$\begin{cases} \ddot{\alpha}(J_0 + J_{21}\sin^2(\varphi)) + \ddot{\varphi} m_2 L_1 l_2 \cos(\varphi) - m_2 L_1 l_2 \sin(\varphi)\dot{\varphi}^2 + \dot{\alpha}\dot{\varphi} J_{21}\sin(2\varphi) = M; \\ \ddot{\alpha} m_2 L_1 l_2 \cos(\varphi) + \ddot{\varphi} J_{21} - \frac{1}{2}\dot{\varphi}^2 J_{21}\sin(2\varphi) g m_2 l_2 \sin(\varphi) = 0, \end{cases} \tag{3.22}$$

where φ, α are the generalized coordinates of the system (the angle of the pendulum deflection and the angle of the boom rotation, respectively); M is the driving moment created by the engine; L_1 and L_2 are the lengths of the boom and pendulum link, respectively, relative to the axes of their rotation; l_1 and l_2 are the distance from the center of mass m_1 to the rotation axis of the boom link and the distance from the center of mass m_2 to the rotation axis of the pendulum link, respectively; m_1 and m_2 are the reduced masses, which are placed on the boom and the pendulum link, respectively; J_0 is the reduced moment of the column inertia relative to the axis of its rotation, provided that the pendulum link is in the lower (stable) position of the equilibrium ($J_0 = J_1 + m_1 l_1^2 + m_2 L_1^2$); J_1 is the total moment of the column inertia and the engine relative to their axis of rotation; J_2 is the reduced moment of the pendulum link inertia relative to the axis of its rotation; J_{21} is the reduced moment of the pendulum link inertia relative to its axis of rotation ($J_{21} = J_2 + m_2 l_2^2$). The dot above the symbol means time derivative. Numerical data for the first system are given below: L_1=0.278 m, L_2=0.3 m, l_1=0.15 m, l_2=0.148 m, m_1=0.3 kg, m_2=0.075 kg, J_1=2.48·10^{-2} kg·m^2, J_2=3.86·10^{-3} kg·m^2. In addition, the maximum drive torque should not exceed 1.0 Hm.

The boundary conditions for the Furuta pendulum movement are as follows:

$$\begin{cases} \varphi(0) = \pi;\ \dot{\varphi}(0) = \alpha(0) = \dot{\alpha}(0) = 0; \\ \varphi(t_S) = \dot{\varphi}(t_S) = \alpha(t_S) = \dot{\alpha}(t_S) = 0. \end{cases} \tag{3.23}$$

The initial movement conditions of the system mean the state of the rest of the system at the position of the pendulum in the lower stable position of the equilibrium, and the final conditions correspond to the state of the rest of the pendulum in the upper unstable position of the equilibrium. In general, the start and end positions of the boom are not required to be the same. However, we have accepted this condition in order to complicate the problem.

The optimization criterion in the studies for all underactuated systems was the following value:

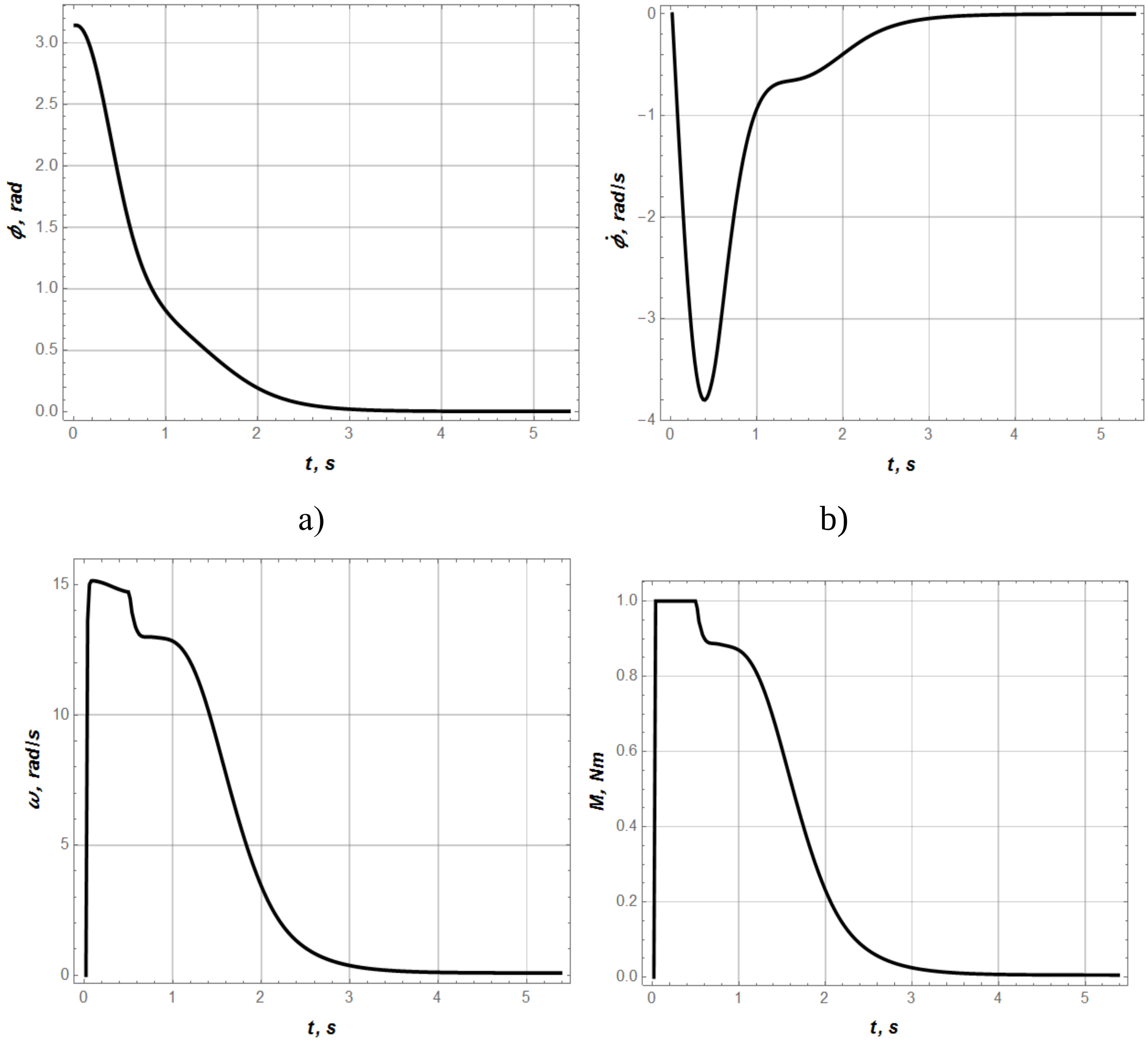
$$CC = 2t_S + \sqrt{t_S^{-1} \int_0^{t_S} u dt} \to \min, \tag{3.24}$$

where *u* is the control, which for the system (19) denotes the driving force, and for the systems (3.17) and (3.22) the driving torque. The optimization criterion (3.24) is the sum of the settling time and the root-mean-square control value. The first component of criterion minimization (3.24) causes a rapid transition from the initial position to the final one, and the second improves the energy efficiency of the underactuated systems (the second component corresponds to the energy consumption of the drive).

The parameters of the ME-D-PSO method for determining the unknown components of the tensor W and the vector B are left the same as those used in the previous subsection. As the problems are more complex, the number of iterations has been increased to 200.

The difference in the training problems of the neurocontroller is the number of unknown parameters of the artificial neural network (neurocontroller). Formula (3.3) was used to establish the number of unknown neural network parameters. For the first system (3.17) we obtained *V*=26, for the second one (3.19), and the third one (3.22) systems *V*=31. The different number of parameters that need to be found is related to the different dimensions of the neurocontrollers input vectors (or, similarly, the different orders of the underactuated systems).

As a result of applying the method, the components of the tensor W and the vector B are obtained, which minimizes the criterion (3.24). The control dynamics for the first underactuated system (3.17) is shown in Fig. 3.14. The reduction of the generalized optimization criterion is shown in Fig. 3.14 (e).

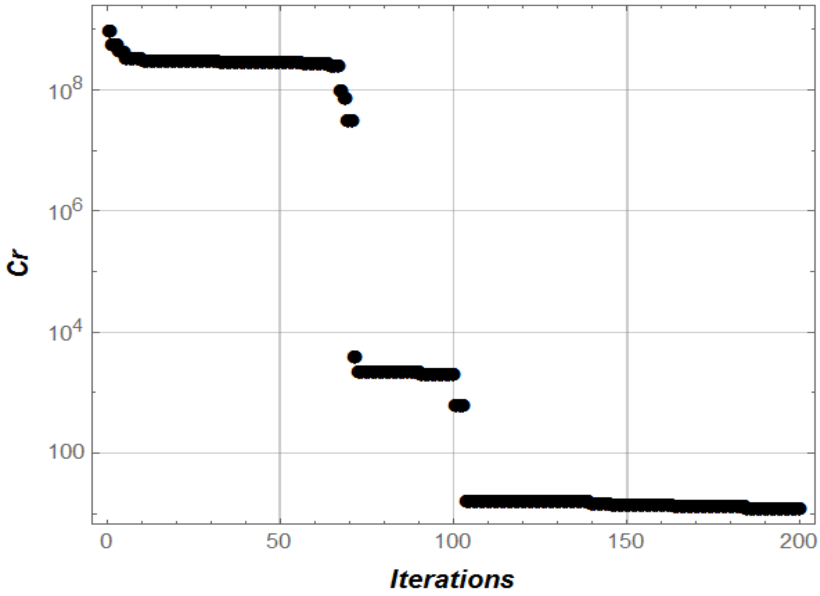


a)

b)

c)

d)



e)

Fig. 3.14 Plots of the control dynamics of the underactuated system inverted pendulum with flywheel: the pendulum angular position (a), the pendulum angular velocity (b), the flywheel angular velocity (c), the flywheel driving torque (d), (e)

The graphical dependencies analysis, which is presented in Fig. 3.14 (a) and (b), shows that the obtained neurocontroller effectively transfers the pendulum into the upper unstable position. However, as can be seen from Fig. 3.14 (b), (c), (d), the synthesized controller has the disadvantage: a rapid increase of torque at the beginning of the control process.

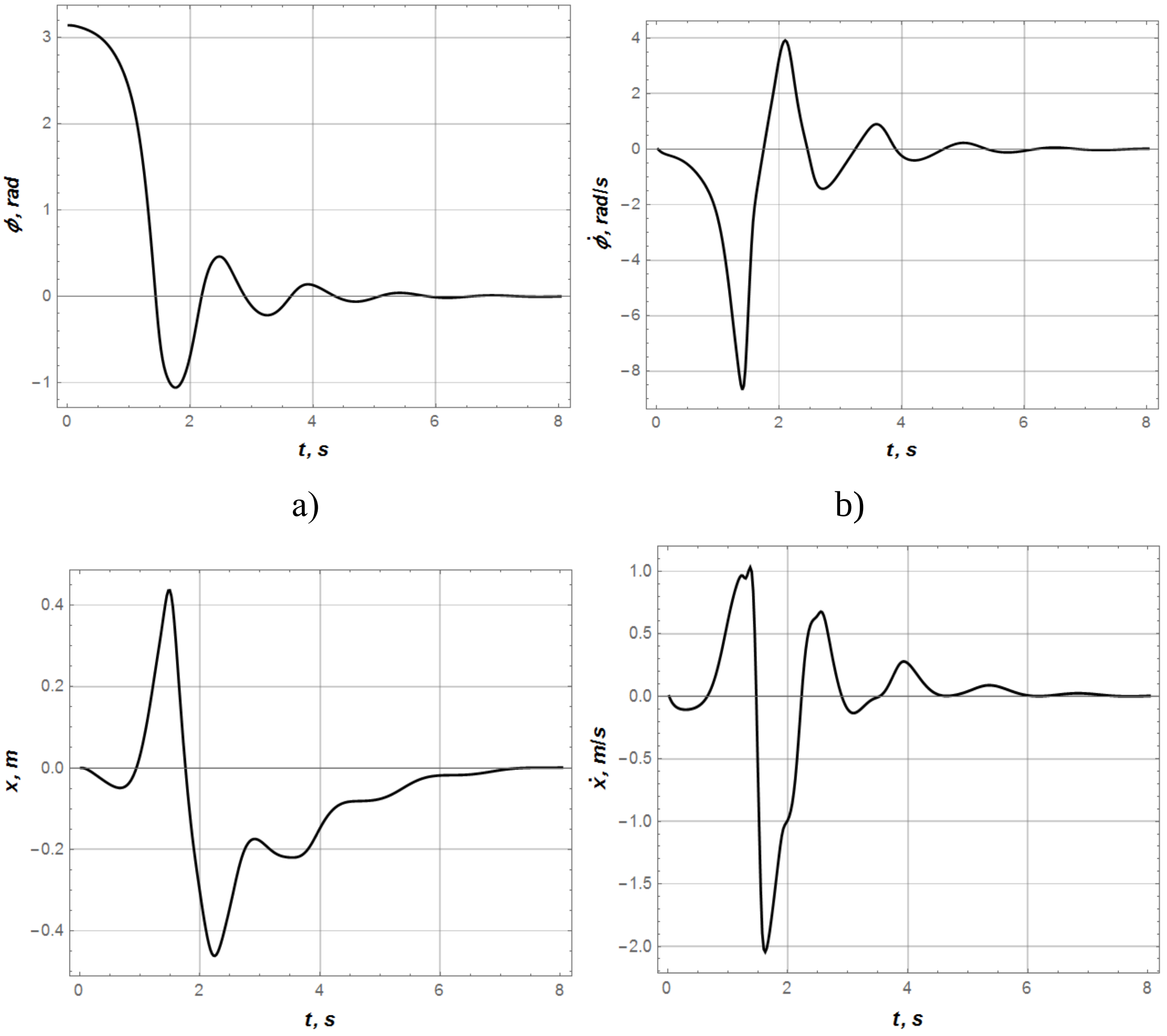
Obviously, in order to reduce the rate of increase of torque applied to the flywheel, it is necessary to introduce some constraints. Such constraints can be mathematically represented as a component of the integrand of the integral criterion (3.24) or in the form of a constraint on the rate of increase of torque (this value must be specified in the statement part of the problem). In the latter case, in the generalized criterion (3.7) it is necessary to enter the value that would reflect the number of the constraint violations. Then, in the course of minimizing the generalized criterion, the number of constraint violations will decrease, moving to zero (i.e., the constraints will be met).

Such problems are beyond the scope of this study, they are the subject of further authors’ research.

The nature of the decrease in the value of the generalized criterion *Cr* shows the rapid convergence of the algorithm (Fig. 3.14 (d)) to the minimum of this criterion.

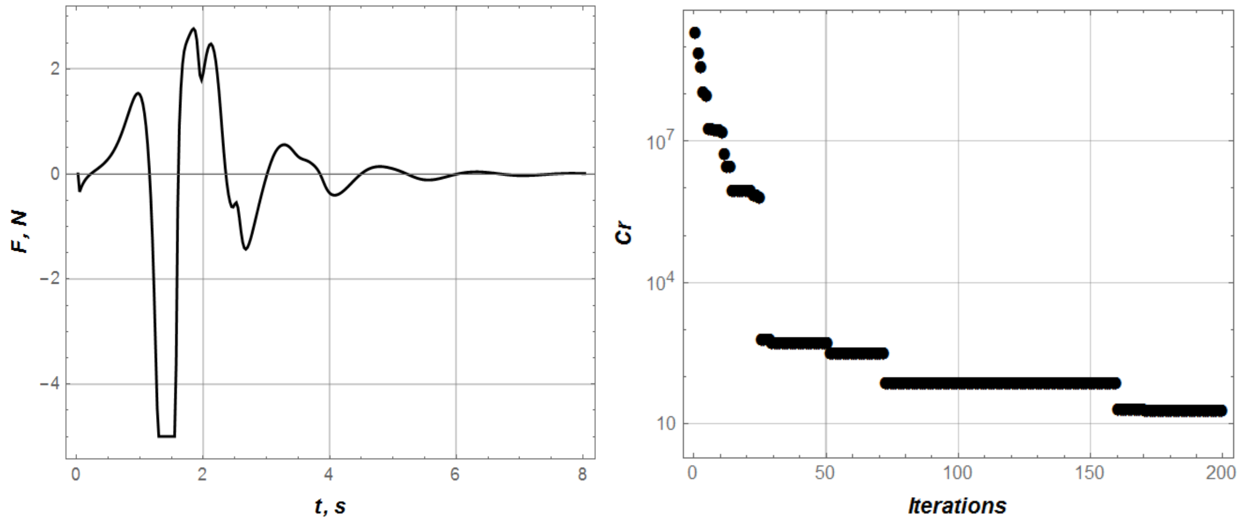
The results of the optimal control of the system inverted pendulum by using the synthesized optimal neurocontroller are shown in Fig. 3.15.

Fig. 3.15 shows that the boundary conditions of the system movement (3.20), as well as the constraints (3.21), are met. The initial position of the pendulum is equal to π, and the final positions of all the system elements are zero. The movement direction of the cart changed six times (Fig. 3.15, d). During the system movement, its drive force is limited (Fig. 3.15, e).



a) b)

c) d)



e) f)

Fig. 3.15 Plots of the dynamics of control of the underactuated system "inverted pendulum": the pendulum angular position (a), the pendulum angular velocity (b), the position of the cart (c), the speed of the cart (d), the driving force of the engine (e), decreasing of the optimization criterion (e)

It should be noted that in the problem statement, in addition to the constraints (3.21), one may consider others. For example, the numbers of changes in the direction of movement of the cart, the number of changes in the values of the driving force, the number of outputs to the constraints, and so on. They will complicate the problem. The problem in such statements will be investigated in the following authors’ studies.

The generalized criterion reduction during the algorithm application (Fig. 3.15, f) is typical: first, a rapid decrease of the criterion with a gradual decrease in tempo. The last iterations do not lead to a significant improvement in the value of the criterion (3.24). It should be mentioned that increasing the number of iterations to 200 was justified. Indeed, 100 iterations for this problem would not be enough to achieve an acceptable magnitude of the optimization criterion.

The neurocontroler corresponding to the conditions of the problem (3.22)-(3.24) was also obtained for the Furuta pendulum. Fig. 3.16 shows the obtained results.

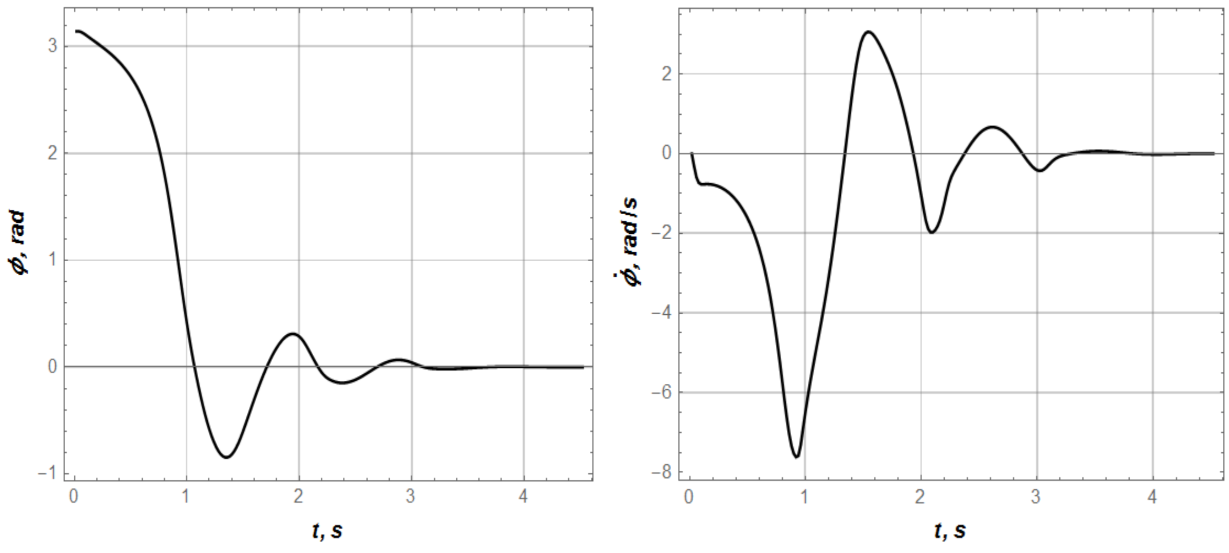
The plots presented in Fig. 3.16 show that during the control, the Furuta pendulum with the help of the neurocontroller was moved from the initial position to the final: the pendulum link at the end of the movement was in the position of vertical unstable equilibrium (Fig. 3.16, a).

The boom changed direction seven times (Fig. 3.16, d) and at the end of the control stopped in the initial position (Fig. 3.16, c). The function of the driving torque reached the maximum and minimum (Fig. 3.16, d) and at the end of the movement is zero. This means that in the future the system will be at rest.

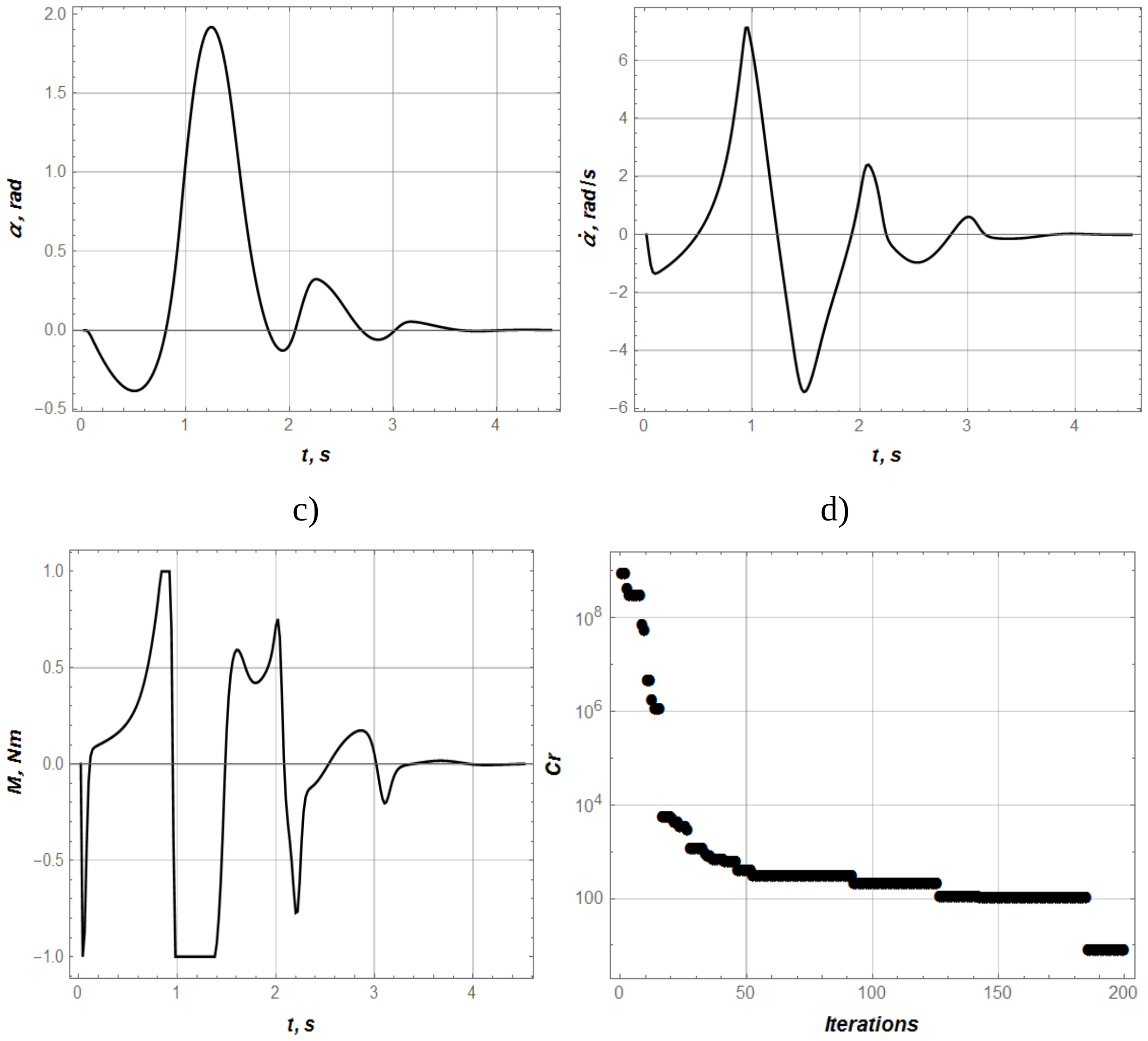
The analysis of the convergence of the algorithm to the minimum criterion (Fig. 16, e) shows that the solution of the problem (the required values of the components of the vector of bias B and the tensor of weights W) was found in the last iterations.

If such a solution is not found, it is necessary to start a new calculation cycle according to the method (Fig. 3.4) (sometimes it is advisable to increase the number of iterations).

In general, it can be concluded that the proposed approach to the synthesis of optimal neurocontrollers for nonlinear MISO systems (for example, the underactuated plants) has shown its effectiveness. It can be recommended for solving other problems, or problems that were solved in this subsection, but in a different statement. This will be the subject of further authors’ research.



a) b)



c) d)

e) f)

Fig. 3.16 Plots of dynamics of the Furuta pendulum control: the pendulum links angular position (a), the pendulum link angular speed (b), the angular position (c), the boom angular speed (c), the engine torque (e), the reduction of the generalized criterion during optimization procedure (e)

3.3.3 Linear MIMO plants

A significant number of the plants belong to the class of multidimensional systems (MIMO systems). Such systems have multiple inputs and multiple outputs. In the simplest case, the MIMO system has two inputs and two outputs and is described by the matrix transfer function of size 2x2.

In order to illustrate the applicability of the method of training the artificial neural networks, we will show how it was used for the synthesis of the MIMO systems neurorcontrollers. The first system is a distillation column for the separation of water and methanol. The mathematical model of such a plant was proposed by Vinante and Luyben in [40]. It is described via a matrix transfer function of size 2x2:

$$\mathrm{G}(s)=\begin{bmatrix} \dfrac{-2{,}2e^{-s}}{7s+1} & \dfrac{1{,}3e^{-0{,}3s}}{7s+1} \\ \dfrac{-2{,}8e^{-1{,}8s}}{9{,}5s+1} & \dfrac{4{,}3e^{-0{,}35s}}{9{,}2s+1} \end{bmatrix} \tag{3.20}$$

The output (controlled) parameters of the plant are the temperatures on the fourth T_4 and the seventeenth T_{17} plate, and the input (control): the intensity of irrigation (reflux) R and the intensity of the flow V at the bottom of the distillation column. The controlled parameters are measured in Celsius degrees, and input (control) in kg/h. It should be noted that the feature of the model (3.20) is the time delays, which complicate the problem of the optimal neurocontroller synthesis. However, it is assumed that the method will be able to overcome them, because the neural network is a universal approximator of the control function.

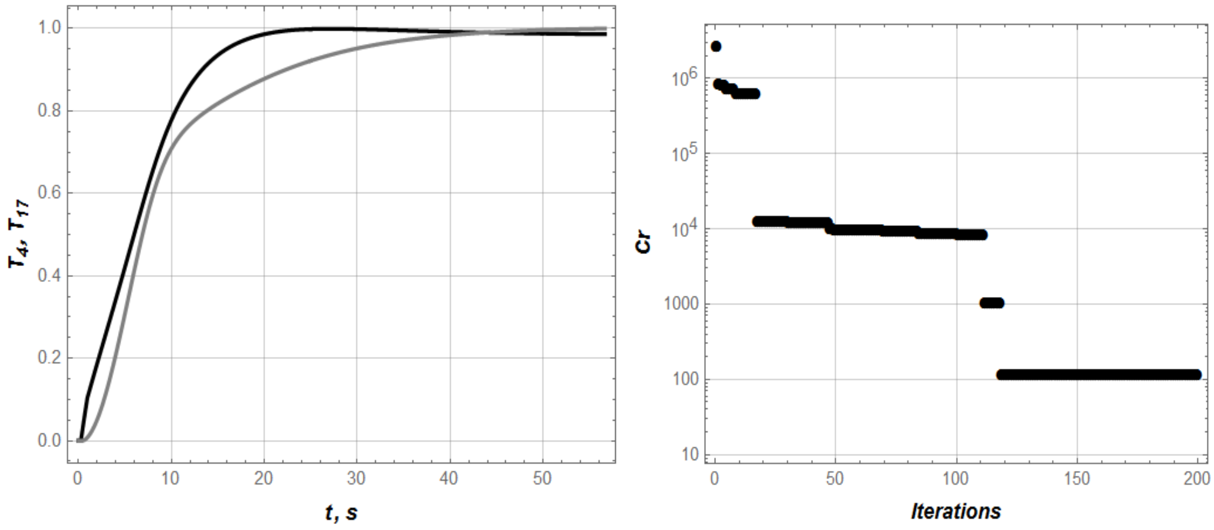
In the case of the neural network with two inputs, two outputs, one hidden layer with five neurons in it, it is necessary to determine its 32 unknown parameters. Their values must satisfy the following conditions:

$$\begin{cases} 1{,}5\max(t_{s1},\, t_{s2})+20\left(t_{s1}^{-1}\int\limits_0^{t_{s1}}\left|\overline{T}_4-T_4\right|dt+t_{s2}^{-1}\int\limits_0^{t_{s2}}\left|\overline{T}_{17}-T_{17}\right|dt\right)\to \min; \\ \begin{cases} \overline{T}_4-T_4\le 0; \\ \overline{T}_{17}-T_{17}\le 0, \end{cases} \end{cases} \tag{3.21}$$

where $\overline{T}_4$ and $\overline{T}_{17}$ are temperature setpoints on the fourth and seventeenth plates, which are taken as equal to one; t_{s1} and t_{s2} are settling times of the first (T_4) and second (T_{17}) variables. Criterion (3.21) is complex: it meets the desire to minimize the weighted sum of the maximum regulation duration and the sum of the average modules of the errors (all components of the complex criterion are undesirable).

The weights of 1.5 and 20, which are located near each of the components, correspond to their importance and reduce the individual components to a dimensionless form.

In addition, we will require that the temperature overshoots on the fourth and seventeenth plates should be zero. This expresses the second line of the system (3.21). As a result of using the method of training the artificial neural network, the components of the tensor W of the network weights and the matrix B of biases were obtained, which allowed meeting the requirements (3.21). The process of the criterion minimization (3.21), which was performed as a result of the algorithm application, is illustrated in Fig. 3.17 (b) (for this problem the number of the iterations was chosen equal to 500, because with a smaller number the process did not have enough time to converge).



a) b)

Fig. 3.17. Plots of the temperature control dynamics on the fourth and seventeenth plates of the distillation column Vinante and Luyben: the temperature on the fourth plate T_4 (black curve), the temperature on the seventeenth plate T_{17} (gray curve) (a), the generalized criterion reduction during the optimization procedure (b)

Fig. 3.17 (a) shows, that the controlled variables reach the setpoint quite quickly. The overshoots for both temperatures (on the fourth and seventeenth plates) are zero. The settling times are equal to t_{s1}=15.9 s, t_{s2}= 29.6 s. The values of the average modules of errors are as follows: $t_{s1}^{-1}\int_0^{t_s}\left|\overline{T_4}-T_4\right|dt = 0.41$ and $t_{s2}^{-1}\int_0^{t_{s2}}\left|\overline{T}_{17}-T_{17}\right|dt = 0.32$ Thus, the problem of the neurorcontroller synthesis is solved.

Another multidimensional system, which is quite common as a test in the development of closed control systems, is the model of the Wood-Berry distillation column [41], which is designed for methanol and water separation. It is represented as a matrix transfer function of size 2x2:

$$\mathrm{G}(s) = \begin{bmatrix} \dfrac{12{,}8e^{-s}}{16{,}7s+1} & \dfrac{-18{,}9e^{-3s}}{21s+1} \\ \dfrac{-6{,}6e^{-7s}}{10{,}9s+1} & \dfrac{-19{,}4e^{-3s}}{14{,}4s+1} \end{bmatrix} \tag{3.22}$$

The controlled parameters of the plants with the matrix transfer function are the concentrations of products in the rectifier X_B and in the vat residue X_D, respectively, and the input (controls): the irrigation intensity R supplied to the column and the flow of water vapor S supplied to the boiler.

Matrix transfer function may be presented in the form of sysmem of differential equation as it is shown in Appendix.

Model (3.22) differs from the model (3.20) in that it has greater time delays and in that the differential equations describing the evolution of both variables are of the second order (for model (3.20) the first equation that describes the change in temperature T_4 is the first order, and the equation to describe the temperature T_{17} is the second order). Therefore, the input vector dimension of the system neurocontroller, which is described by the matrix transfer function (3.22) is four.

Now we can determine the number of the unknown parameters that need to be calculated in the process of training the artificial neural network. We perform the following calculations by using formula (3.3) and obtain for a single-layer network with five neurons in the hidden layer *V*=37. For this problem of the neurocontroller synthesis we use the optimization criterion similar to (3.21):

$$1{,}5\max(t_{s1},\, t_{s2}) + 20\left(t_{s1}^{-1}\int_0^{t_{s1}} \left|\overline{X}_B - X_B\right| dt + t_{s2}^{-1}\int_0^{t_{s2}} \left|\overline{X}_D - X_D\right| dt \right) \to \min \qquad (3.23)$$

and a zero overshoot requirement for both variables:

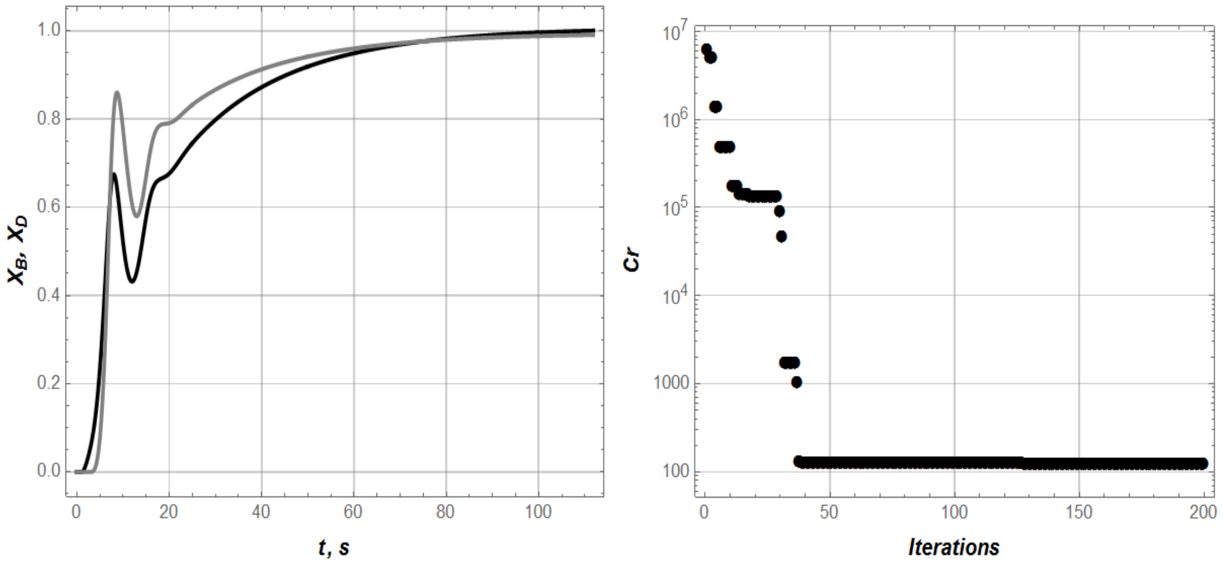
$$\begin{cases} \overline{X}_B - X_B \le 0; \\ \overline{X}_D - X_D \le 0, \end{cases} \qquad (3.24)$$

where $\overline{X}_B$ and $\overline{X}_D$ are the corresponding setpoints of controlled variables, which in both cases are equal to one; t_{s1} and t_{s2} are the settling times of the first (X_B) and second (X_D) variables.

The application of the developed method of training the artificial neural network made it possible to obtain the components of the tensor W of the weights of the network and its matrix B of bias, which minimizes the criterion (3.23) and meets the constraints (3.24).

In order to illustrate the obtained results, we present graphical dependences (Fig. 3.18). The minimization of the criterion process (3.23) is illustrated by the plots in Fig. 3.18 (b). From Fig. 3.18, (a) it is seen that both controlled values reach the setpoint without overshoots.

The settling times are equal to: t_{s1}= 60.2 s, t_{s2}= 54.0 s. The values of the average modules of the errors are as follows: $t_{s1}^{-1}\int_0^{t_{s1}} \left|\overline{X}_B - X_B\right| dt = 0.29$ and $t_{s2}^{-1}\int_0^{t_{s2}} \left|\overline{X}_D - X_D\right| dt = 0.26$. Therefore, this problem of neurocontroller synthesis has been successfully solved.



a) b)

Fig. 3.18 Plots of the control dynamics of the concentration of products in the rectifier X_B and in the vat residue X_D: product concentration in the rectifier X_B (black curve), product concentration in the rectifier X_D (gray curve) (a), reduction of the generalized criterion during the optimization procedure (b)

Conclusions to chapter 3

1. A method of training artificial neural networks for the synthesis of optimal automatic controllers has been developed. It allows taking into account the constraints imposed on the components of the phase vector of the plant and the vector of control. The method is based on the minimization of the generalized optimization criterion in the space of values of the components of the tensor weights of the neural network and the components of the matrix of its biases. The criterion was minimized by using the method ME-D-PSO.
2. In order to evaluate the efficiency of the developed method, the results of training neural networks (neurocontrollers) for several linear and nonlinear (underactuated) plants were performed. Analysis of the dynamics of their control and generalized criterion during the operation of the ME-D-PSO

algorithm confirmed the applicability of the method for the synthesis of optimal controlers.

3. In particular, for the Wood-Berry distillation column, the settling time was 60.2 and 54.0 s, and the average error was 0.29 and 0.26. For the Vinante-Libuben distillation column, the settling time was 15.9 and 29.6 s, and the average error was 0.41 and 0.32. These values support the efficiency of trained neurocontrollers application for MIMO plants.

References to chapter 3

1. Cybenko, G. V. Approximation by Superpositions of a Sigmoidal function. Mathematics of Control Signals and Systems, 1989. T. 2, № 4. pp. 303-314.
2. Back-Propagation and Other Differentiation Algorithms / Goodfellow Ian, Bengio Yoshua, Courville Aaron. Deep Learning. MIT Press. 2016 pp. 200-220.
3. Deutsches Patent-und Markenamt: веб-сайт. URL: https://depatisnet.dpma.de/DepatisNet/depatisnet?window=1&space=menu&content=index&action=einsteiger&switchToLang=en (date of access 28.03.2021)
4. Kennedy J., Eberhart R. Particle swarm optimization. Proceedings of the IEEE International Conference on Neural Networks, vol. 4, pp. 1942–1948, December 1995.
5. Yudong Zhang, Shuihua Wang, Genlin Ji. A Comprehensive Survey on Particle Swarm Optimization Algorithm and Its Applications. Mathematical Problems in Engineering. Vol. 2015, Article ID 931256, 38 pages.
6. Zhang Y., Wu L. Crop classification by forward neural network with adaptive chaotic particle swarm optimization. Sensors. 2011. vol. 11, no. 5, pp. 4721–4743.

7. Munoz D.M., Llanos C.H, Coelho L.D.S., Ayala-Rincon M. Hardware opposition-based PSO applied to mobile robot controllers. Engineering Applications of Artificial Intelligence. 2014. Vol. 28. pp. 64–77.

8. Xiao Y., Xiao J., Lu F., Wang S. Ensemble ANNs-PSO-GA approach for day-ahead stock E-exchange prices forecasting. International Journal of Computational Intelligence Systems. 2014. vol. 7, no. 2, pp. 272–290.

9. Zhong Y., Huang X., Meng P., Li F. PSO-RBF neural network PID control algorithm of electric gas pressure controller. Abstract and Applied Analysis. 2014. vol. 2014, Article ID 731368, 7 pages

10. Perng J.-W., Chen G.-Y., Hsieh S.-C. Optimal PID-controller design based on PSO-RBFNN for wind turbine systems. Energies. 2014. vol. 7, no. 1, pp. 191-209

11. Mandal P., Zareipour H., Rosehart W.D. Forecasting aggregated wind power production of multiple wind farms using hybrid wavelet-PSO-NNs. International Journal of Energy Research. 2014. vol. 38, no. 13, pp. 1654–1666.

12. Zhang Y., Wang S., Wu L. A novel method for magnetic resonance brain image classification based on adaptive chaotic PSO. Progress in Electromagnetics Research. 2010. vol. 109, pp. 325–343.

13. Nasimi R., Irani R. Identification and modeling of a yeast fermentation bioreactor using hybrid particle swarm optimization-artificial neural networks. Energy Sources, Part A: Recovery, Utilization and Environmental Effects. 2014. vol. 36, no. 14, pp. 1604–1611.

14. Saraswathi S., Sundaram S., Sundararajan N., Zimmermann M., Nilsen-Hamilton M. ICGA-PSO-ELM approach for accurate multiclass cancer classification resulting in reduced gene sets in which genes encoding secreted proteins are highly represented. IEEE/ACMTransactions on Computational Biology and Bioinformatics. 2011. vol. 8, no. 2, pp. 452–463.

15. He H.-D., Lu W.-Z., Xue Y. Prediction of particulate matter at street level using artificial neural networks coupling with chaotic particle swarm optimization algorithm. Building and Environment. 2014. vol. 78, pp. 111–117.

16. Yeh W.-C. New parameter-free simplified swarmoptimization for artificial neural network training and its application in the prediction of time series. IEEE Transactions on Neural Networks and Learning Systems. 2013. vol. 24,no. 4, pp. 661–665.

17. Asadnia M., Chua L.H.C., Qin X.S., Talei A. Improved particle swarmoptimization-based artificial neural network for Rainfall-Runoff modeling. Journal of Hydrologic Engineering. 2014. vol. 19, no. 7, pp. 1320–1329.

18. Das G., Pattnaik P.K., Padhy S.K. Artificial Neural Network trained by Particle SwarmOptimization for non-linear channel equalization./ Expert Systems with Applications. 2014. vol. 41, no. 7, pp. 3491–3496.

19. Conradie AVE, Miikkulainen R, Aldrich C. Adaptive control utilising neural swarming. In: Proceedings of the genetic and evolutionary computation conference, New York, USA. 2002

20. Lin C-J, Hong S-J, Lee C-Y. The design of neuro-fuzzy networks using particle swarm optimization and recursive singular value decomposition. 2006 International joint conference on neural networks, Sheraton Vancouver Wall Centre Hotel, Vancouver, BC, Canada, 16–21 July 2006

21. Hafiz T.R., Waqas H.B., Jamil A., Saad A.B. Training of Artificial Neural Network Using PSO With Novel Initialization Technique. 2018 International Conference on Innovation and Intelligence for Informatics, Computing, and Technologies (3ICT)

22. Jeng-Fung C., Quang H.D., Ho-Nien H. Training Artificial Neural Networks by a Hybrid PSO-CS Algorithm. Algorithms. 2015. 8, pp. 292-308.

23. Venayagamoorthy G.K., Gudise V.G. Comparison of Particle Swarm Optimization and Backpropagation as Training Algorithms for Neural

Networks. Proceedings of the 2003 IEEE Swarm Intelligence Symposium, 2003. SIS '03, Institute of Electrical and Electronics Engineers (IEEE), Jan 2003.

24. Nasser M., Seyed J.M. Comparison of Particle Swarm Optimization and Backpropagation Algorithms for Training Feedforward Neural Network. Journal of mathematics and computer science. 2014. 12, pp. 113-123.

25. Kaur J., Kalr, A., Sharma D. Comparative Survey of Swarm Intelligence Optimization Approaches for ANN Optimization. Intelligent Communication, Control and Devices. 2018. pp. 305–314.

26. Garro B.A., Sossa H., V´azquez R.A. Back-propagation vs particle swarm optimization algorithm: which algorithm is better to adjust the synaptic weights of a feed-forward ANN? International Journal of Artificial Intelligence. 2011. vol. 7, no. 11, pp. 208-218.

27. Garro B.A., Sossa H., Vazquez R.A. Design of artificial neural networks using a modified particle swarm optimization algorithm. Proceedings of IEEE International Joint Conference on Neural Networks (IJCNN'09), Atlanta, GA, USA, 2009, pp. 938-945.

28. Conforth M., Meng Y. Toward evolving neural networks using bio-inspired algorithms. IC-AI, H. R. Arabnia and Y. Mun, Eds. 2008 pp. 413–419, CSREA Press.

29. Da Y., Xiurun G. An improved PSO-based ANN with simulated annealing technique. Neurocomputing. 2005. vol. 63, pp. 527-533.

30. Garro B.A., Sossa H., Vazquez R.A. Design of artificial neural networks using a modified particle swarm optimization algorithm. Proceedings of the International Joint Conference on Neural Networks (IJCNN '09), pp. 938-945, IEEE, Atlanta, Ga, USA, June 2009.

31. Garro B., Sossa H., Vazquez R. Design of artificial neural networks using differential evolution algorithm. Neural Information Processing. Models and Applications, K. Wong, B. Mendis, and A. Bouzerdoum, Eds., vol. 6444 of

Lecture Notes in Computer Science, pp. 201–208, Springer, Berlin, Germany, 2010.

32. Romasevych Y., Loveikin V., Makarets V. Optimal Constrained Tuning of PI-Controllers via a New PSO-Based Technique. International Journal of Swarm Intelligence Research (IJSIR), Volume 11, Issue 4, p 87-105.

33. Romasevych Yu., Loveikin V. A Novel Multi-Epoch Particle Swarm Optimization Technique. Cybernetics and Information Technologies. 2018. Vol. 18 (3). pp. 62–74.

34. Åströn K. J., Hägglund T. Benchmark Systems for PID Control. International Federation of Automatic Control. 2000. P. 165–166.

35. Lysenko V., Bolbot I., Romasevych Y., Loveykin V., Voytiuk V. Algorithms of robotic electrotechnical complex control in agricultural production. Control Systems: Theory and Applications. River Publishers, 2018, ch. 11, pp. 271-289

36. Loveikin V.S., Romasevich Yu.A., Khoroshun S.A., Shevchuck A.G. Time-optimal control of a simple pendulum with a movable pivot. Part 1. International Applied Mechanics. 2018. 54, N 3. pp. 358 – 365.

37. Spong M.W., Corke P., Lozano R. Nonlinear control of the inertia wheel pendulum. Automatica, 37, 2001, pp. 1845-1851.

38. Russ Tedrake. Underactuated Robotics: Algorithms for Walking, Running, Swimming, Flying, and Manipulation (Course Notes for MIT 6.832). Downloaded on 06.04.2020 from http://underactuated.mit.edu/

39. Furuta K., Yamakita M., Kobayashi S. Swing-up control of inverted pendulum using pseudo-state feedback. Journal of Systems and Control Engineering. 1992. 206(6), pp. 263-269.

40. Vinante, Luyben Experimental studies of distillation decoupling. Control Eng. Practice.1972. Vol. 1, Nº 6 pp. 999-1008.

41. Wood R.K., Berry M.W. Terminal composition control of a binary distillation column. Chem. Eng. Sci. 1973. 28, pp. 707-1717.

42. Qing-Guo W., Zhen Y., Wen-Jian C., Chang-Chieh H. PID Control for Multivariable Processes. Lecture Notes in Control and Information Sciences 373. Editors: M. Thoma, M. Morari. Springer-Verlag Berlin Heidelberg. 2008. p. 273.

APPENDIX. CONVERTING A MATRIX TRANSFER FUNCTION INTO SYSTEM OF DIFFERENTIAL EQUATIONS (ILLUSTRATED BY WOOD-BERRY COLUMN)

In many problems of synthesis of control systems transfer functions and systems of differential equations are used. Both presented a mathematical model of a plant. In the development of the control systems transfer functions are more common. However, in many cases systems of differential equations are more euphemistic. Thus, it causes the problem of converting one kind of mathematical model of a plant into another one.

In the current work, an example of converting is presented. We have chosen the plant – Wood-Berry column. It has two inputs and two outputs and it is described by the matrix transfer function:

$$\mathrm{G}(s)=\begin{bmatrix}\dfrac{12{,}8e^{-s}}{16{,}7s+1} & \dfrac{-18{,}9e^{-3s}}{21s+1}\\ \dfrac{-6{,}6e^{-7s}}{10{,}9s+1} & \dfrac{-19{,}4e^{-3s}}{14{,}4s+1}\end{bmatrix}, \qquad (1)$$

where *s* – Laplace image. Taking into consideration the transfer function (1) we may write down the equation of Laplace transform of variables X_D and X_B:

$$\begin{bmatrix}X_D(s)\\ X_B(s)\end{bmatrix}=\begin{bmatrix}\dfrac{12{,}8e^{-s}}{16{,}7s+1} & \dfrac{-18{,}9e^{-3s}}{21s+1}\\ \dfrac{-6{,}6e^{-7s}}{10{,}9s+1} & \dfrac{-19{,}4e^{-3s}}{14{,}4s+1}\end{bmatrix}\begin{bmatrix}R(s)\\ S(s)\end{bmatrix}, \qquad (2)$$

where *S* and *R* – steam flow and reflux respectively; X_D and X_B – concentrations of the separated components: water and methanol respectively.

The matrix form (3) we may present in the following form:

$$\begin{cases} X_D(s) = \dfrac{12{,}8e^{-s}}{16{,}7s+1}R(s) + \dfrac{-18{,}9e^{-3s}}{21s+1}S(s); \\ X_B(s) = \dfrac{-6{,}6e^{-7s}}{10{,}9s+1}R(s) + \dfrac{-19{,}4e^{-3s}}{14{,}4s+1}S(s). \end{cases} \tag{3}$$

Taking some transformations of the system (3) we may obtain the following form:

$$\begin{cases} (16{,}7s+1)(21s+1)X_D(s) = 12{,}8e^{-s}(21s+1)R(s) - 18{,}9e^{-3s}(16{,}7s+1)S(s); \\ (10{,}9s+1)(14{,}4s+1)X_B(s) = -6{,}6e^{-7s}(14{,}4s+1)R(s) - 19{,}4e^{-3s}(10{,}9s+1)S(s). \end{cases} \tag{4}$$

Simplification of the system of equation (4) leads to the next system of equations:

$$\begin{cases} (350{,}7s^2 + 37{,}7s+1)X_D(s) = (268{,}8s+12{,}8)e^{-s}R(s) - (315{,}63s+18{,}9)e^{-3s}S(s); \\ (156{,}96s^2 + 25{,}3s+1)X_B(s) = -(95{,}04s+6{,}6)e^{-7s}R(s) - (211{,}46s+19{,}4)e^{-3s}S(s). \end{cases} \tag{5}$$

Carrying out the inverse Laplace transformation leads to the final result – the system of differential equations, which corresponds to the matrix transfer function (1):

$$\begin{cases} \ddot{X}_D(t)350{,}7 + 37{,}7\dot{X}_D(t) + X_D(t) = \\ = 268{,}8\dot{R}(t-1) + 12{,}8R(t-1) - 315{,}63\dot{S}(t-3) - 18{,}9S(t-3); \\ 156{,}96\ddot{X}_B(t) + 25{,}3\dot{X}_B(t) + X_B(t) = \\ = -95{,}04\dot{R}(t-7) - 6{,}6R(t-7) - 211{,}46\dot{S}(t-3) - 19{,}4S(t-3). \end{cases} \tag{6}$$

The found mathematical model is the same as presented in the work [1]. It will be exploited in the area of the development of optimal control systems.

Reference:

1. Qing-Guo W., Zhen Y., Wen-Jian C., Chang-Chieh H. PID Control for Multivariable Processes. Lecture Notes in Control and Information Sciences 373. Editors: M. Thoma, M. Morari. Springer-Verlag Berlin Heidelberg. – 2008. – p. 273.

Future

WYŻSZA SZKOŁA ZARZĄDZANIA i ADMINISTRACJI W OPOLU

WSZiA OPOLE

1996

НУБіП

1898

TRADERE PROGREDI PROSPICERE

ISBN 978-83-66567-39-9

9 788366 567399