

ARTICLE

Intelligent Robust Control of Redundant Smart Robotic Arm Pt I: Soft Computing KB Optimizer - Deep Machine Learning IT

Alena V. Nikolaeva Sergey V. Ulyanov*

Dubna State University, Universitetskaya Str.19, Dubna, Moscow Region, 141980, Russia
ISESYS LLC (EFKO GROUP), Naberezhnaya Ovchinnikovskaya Str.20 Bld.2, Moscow, Russia

ARTICLE INFO

Article history

Received: 29 October 2019

Accepted: 8 April 2020

Published Online: 15 April 2020

Keywords:

Intelligent control system

Knowledge base

Soft computing technology

Decomposition

Redundant robotic manipulator

ABSTRACT

Redundant robotic arm models as a control object discussed. Background of computational intelligence IT on soft computing optimizer of knowledge base in smart robotic manipulators introduced. Soft computing optimizer is the sophisticated computational intelligence toolkit of deep machine learning SW platform with optimal fuzzy neural network structure. The methods for development and design technology of control systems based on soft computing introduced in this Part 1 allow one to implement the principle of design an optimal intelligent control systems with a maximum reliability and controllability level of a complex control object under conditions of uncertainty in the source data, and in the presence of stochastic noises of various physical and statistical characters. The knowledge bases formed with the application of soft computing optimizer produce robust control laws for the schedule of time dependent coefficient gains of conventional PID controllers for a wide range of external perturbations and are maximally insensitive to random variations of the structure of control object. The robustness is achieved by application a vector fitness function for genetic algorithm, whose one component describes the physical principle of minimum production of generalized entropy both in the control object and the control system, and the other components describe conventional control objective functionals such as minimum control error, etc. The application of soft computing technologies (Part I) for the development a robust intelligent control system that solving the problem of precision positioning redundant (3DOF and 7 DOF) manipulators considered. Application of quantum soft computing in robust intelligent control of smart manipulators in Part II described.

1. Introduction

The approach based on Soft Computing Optimizer (SCO) for design intelligent control systems (ICS) allows one to design an optimal ICS with a maximum reliability and controllability level for the set of dynamic systems under the presence of uncertainty in

the source data; to reduce the number of sensors both in the control channel and in the Measurement System (MS) without loss of precision control quality and accuracy. The robust ICS based on this approach requires minimum source data on both the behavior of the Control Object (CO) and the external perturbations. SCO is the SW toolkit of deep machine learning platform with optimal struc-

*Corresponding Author:

Sergey V. Ulyanov,

ISESYS LLC (EFKO GROUP), Naberezhnaya Ovchinnikovskaya Str.20 Bld.2, Moscow, Russia;

Email: ulyanovsv@mail.ru

ture of fuzzy neural network (FNN).

Let us consider the design features of the ICS IT structure and the SCO of knowledge base (SCOptKBTM). Analysis of the simulation results made it possible to establish that the application of the FNN-based technique does not guarantee the required accuracy achievement of the Teaching Signal (TS) approximation. As a result, the level of sensitivity of CO increases, and the reliability of ICS decreases. SCO based on soft computing techniques increases the level of ICS reliability. Consider an SCO structure containing the optimal FNN configuration. The main features of SCO, the design of reliable Knowledge Bases (KB) of Fuzzy Controllers (FC) are described in the Appendix. The methodology of fuzzy and joint stochastic modeling of control system based on SCO is discussed to assess the stability and limitations of ICS. The effectiveness of SCO-based control processes using specific typical examples (standards) of COs such as a robotic manipulator is demonstrated under conditions of incomplete information about the CO structure and unpredicted control situations.

1.1. Background of physical laws ICS design

Figures 1 and 2 demonstrate typical criteria for control quality, their interrelations with different types of computations and simulation types, as well as the hierarchy of levels of control quality depending on the required level of intelligence of the Automatic Control System (ACS).

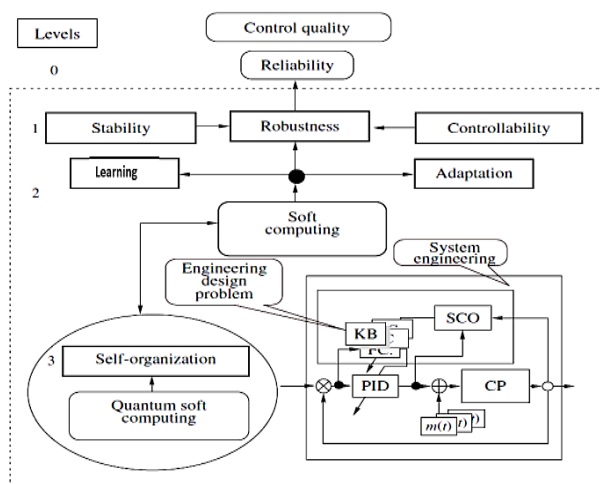


Figure 1. The interrelation between the types and hierarchical levels of control quality criteria

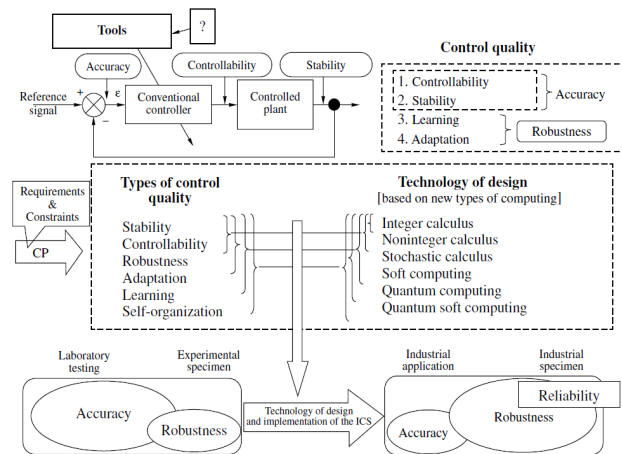


Figure 2. The interrelation between the control quality criteria, types of intelligent computing, and simulation in designing robust KBs of the FC

The key point of this design IT is the use of the method of eliciting objective knowledge about the control process irrespective of the subjective experience of experts and the design of objective KBs of a FCs which is principal component of a robust ICS.

Figure 3 presents the main components and their interrelations in the information design technology based on new types of computing (soft and quantum computational intelligence).

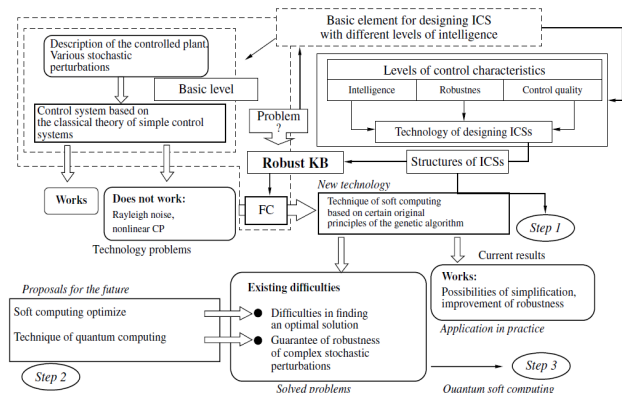


Figure 3. The process of development and creation of information technology for design an integrated ICS

A robust KB of the FC is the result of application considered technology. We can change the property of the system without changing the intrinsic passive property using the generalized canonical transformation. Indeed, if a given system fails to satisfy the stabilizable conditions by the feedback: positive definiteness of the Hamiltonian function and zero-state detectability, then still we may be able to transform the system into an appropriate Hamiltonian system which can be stabilized by the intelligent feedback.

Figure 4 shows the role of thermodynamic trade-off in robust control design.

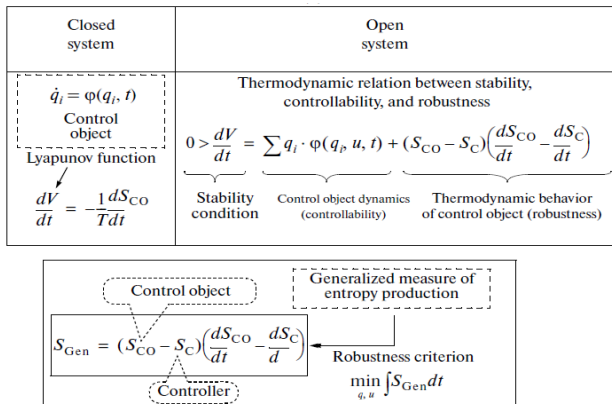


Figure 4. Physical law of intelligent control as background of ICS design technology

Remark. This approach was firstly presented in [1]. It was introduced the new physical measure of control quality to complex non-linear controlled objects described as non-linear dissipative models. This physical measure of control quality is based on the physical law of minimum entropy production rate in ICS and in dynamic behavior of complex object. The problem of the minimum entropy is equivalent with the associated problem of the maximum released mechanical work as the optimal solutions of corresponding Hamilton-Jacobi-Bellman equations. It has shown that the variational fixed-end problem of the maximum work W is equivalent to the variational fixed-end problem of the minimum entropy production. In this case both optimal solutions are equivalent for the dynamic control of complex systems and the principle of minimum of entropy production guarantee the maximal released mechanical work with *intelligent* operations. This new physical measure of control quality applied as fitness function of Genetic Algorithm (GA) in optimal control system design. The introduction of physical criteria (the minimum entropy) can guarantee the stability and robustness of control. This method differs from aforesaid design method in that a new intelligent global feedback in control system introduced. The interrelation between the stability of CO (the Lyapunov function) and controllability is used. The basic peculiarity of the given method is the necessity of model investigation for CO and the computing of entropy production rate through the parameters of the developed model. The integration of joint systems of equations (the equations of mechanical model motion and the equations of entropy production rate) enable to use the result as the fitness function in GA as a new type of CI. Acceleration method of integration for these equations is described in [2].

A continuous-time system in the feedback interconnection with the resetting controller is considering in [3]. Every time the emulated energy of the controller reaches its maximum, the states of the controller reset in such a way that the controller's emulated energy becomes zero. Alternatively, the controller states can be made reset every time the emulated energy is equal to the actual energy of the plant, enforcing the second law of thermodynamics that ensures that the energy flows from the more energetic system (the plant) to the less energetic system (the controller). The proof of asymptotic stability of the closed-loop system in this case requires the non-trivial extension of the hybrid invariance principle, which in turn is a very recent extension of the classical Barbashin-Krasovskii invariant set theorem. The subtlety here is that the resetting set is not a closed set and as such a new transversality condition involving higher-order Lie derivatives is needed. A system theoretic foundation for thermodynamics is developed in [4].

Main goal of robust intelligent control is support of optimal trade-off between stability, controllability and robustness with thermodynamic relation as thermodynamically stabilizing compensator (see Figure 4). The hybrid energy dissipating controller provides effectively one-way energy transfer between the CO and the controller [4].

The hybrid controller with resetting set is a thermodynamically stabilizing compensator. Analogous thermodynamically stabilizing compensators can be constructed for lossless dynamical systems. Detail description of interrelations between energy-based and thermodynamic-based controller design is given in [4, 5].

On Figure 4 joint in analytic form different measures of control quality such as stability, controllability, and robustness supporting the required level of reliability and accuracy. Consequently, the interrelation between the Lyapunov stability and robustness is the main physical law for designing ACS. This law provides the background for an applied technique of robust ICS's (with different levels of intelligence designing KB's) based on the application based of soft computing technologies.

2. The structure of ICS design IT

The general hierarchical structure and stages of execution of information technology embedded in the process of design of integrated fuzzy ICS for autonomous and interconnected COs with different physical nature (so called port-controlled Hamiltonian systems) is shown in Figure 5.

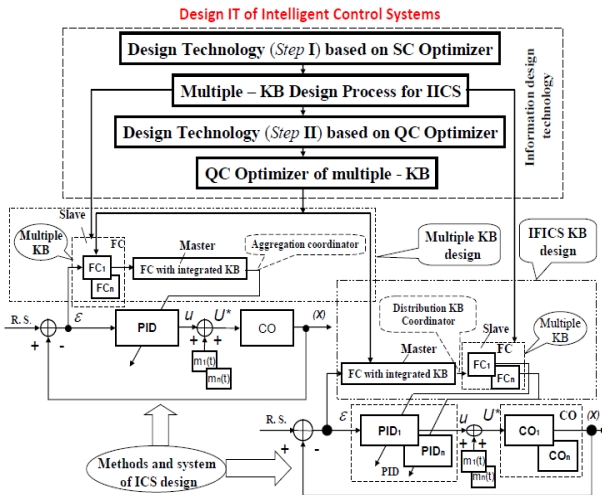


Figure 5. General hierarchical structure of information design technology of robust KBs for integrated fuzzy ICS

This technology uses computational intelligence toolkit for design of KBs in the FC of the lower executive level. The main role in the structure of this technology is played by the development of robust KBs based on corresponding optimizers (see the block “Information design technology” labeled by dashed lines). Note some structural and functional specific features of design stages in Figure 5.

At the *first* stage the technology of design of optimizer KBs with soft computing SCOptKB™ forms robust KBs for fixed learning control situation. At the *second* stage quantum optimizer QCOptKB™ used to realize the process of design of the generalized robust KB of hybrid fuzzy PID controllers operating in contingency control situations.

Thus, the process of design of robust KBs consists of two interconnected stages based on soft and quantum computing, respectively. Functionally, at the first design stage (see Figure 5) individual KBs for two (or more) FCs for particular control situations (learning situations) are formed. Optimizer of KBs are used with the technology of soft computing and fuzzy stochastic simulation. The optimizer of KB SCOptKB™ was developed in [6, 7] as the new toolkit of computational intelligence based on the technology of soft computing (first design stage), including the GAs and FNNs for realization of optimization and learning procedures (universal robust approximation) of production rules in KBs, respectively. The toolkit was used for extraction of objective knowledge from the dynamic behavior of weakly structured models of complex COs and design of robust KBs in FC with deep knowledge representation (see Figure 6).

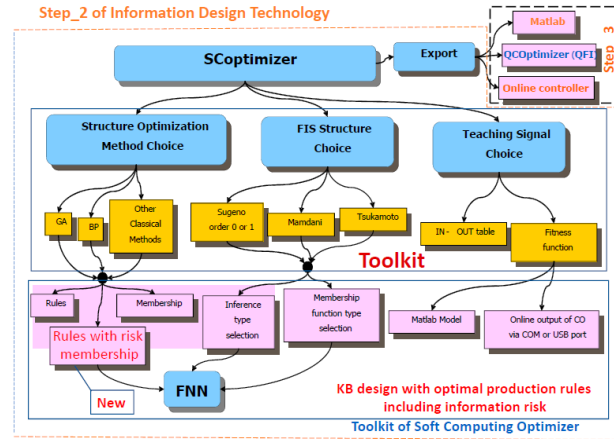


Figure 6. Structure of computational intelligence toolkit of design IT

It should be underlined that the toolkit of Knowledge Base Optimizer (KBO) realizes in the stochastic fuzzy simulation *global intelligent feedback* (new type of feedback [8]), which makes it possible to objectively extract and compress valuable information from the dynamic behavior of the CO and applied controller type. For guaranteed achieving the required robustness level and control quality in the form of fitness functions of GA information and physical criteria are introduced (information-thermodynamic criterion of optimal distribution of physically achievable levels of stability, controllability, and robustness in ICSs [6, 8]). The optimization of control processes with required quality and robustness levels is achieved for fixed search space and type of fitness functions of the GA. The developed new toolkit of computational intelligence is the generalization of methodology and methods in [5, 9-11].

The application of the self-organization principle based on quantum computing is the algorithmic essence of the second stage for increasing the robustness of the KB. The block diagram of design of robust KBs based on the principle of self-organization of ICSs (using quantum effects) and the structure of information flows in the design technology are shown in Figure 7; in this figure the main objectives and content of stages of design technology (Figure 5) are explained.

The structure and software support of quantum KBO QCOptKB™ are considered below in Part II. Let us elucidate some specific features and technical details of realization of technologies of intelligent computing in processes of design of robust KBs shown in Figures 5 and 6.

Studies performed in [6, 12] demonstrated the existence of a rather broad domain of preservation of robustness of individual KBs designed at the first stage based on optimizer KBs. The introduction of the technology of soft computing (whose kernel is comprised of GAs and FNNs)

extended the domains of efficient application of FCs due to the addition of new functions in the form of learning and adaptation. Multiple results of simulation and practical application showed [6, 13, 14] that for random events and control situations with known perturbation probability distribution density functions optimizer KBs with soft computing can be used to design robust KBs in FCs, which do not lose the robustness property in many contingency control situations.

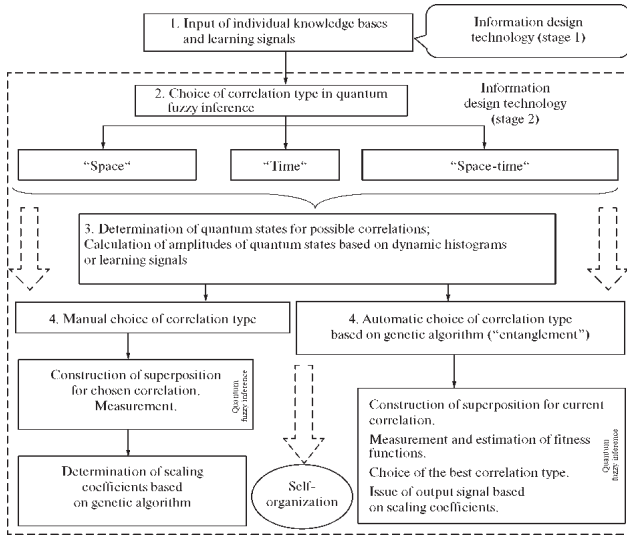


Figure 7. Block diagram of design of robust KBs and structure of information flows in technology of design of robust KBs based on the principle of self-organization of ICS

The SCO is a new, efficient software tool for KBs design of robust ICSs based on soft computing with the use of new optimization criteria (in the form of new fitness functions of Gas; see in details Appendix). As these criteria, we take the thermodynamic and information-entropy criteria represented in Table 1.

Table 1. The types and the role of the fitness function of the GA in the SCO

Type of GA	Criteria	Fitness function (FF)	Role of the FF
GA ₁ : Optimization of linguistic variables	Maximum joint information entropy and minimum information about signals separately	$H_{x_i}^j = -p_{x_i}^j \log(p_{x_i}^j) = -p(x_i x_i = \mu_{x_i}^j) \log[p(x_i x_i = \mu_{x_i}^j)]$ $= \frac{1}{N} \sum_{i=1}^N \mu_{x_i}^j(x_i(t)) \log[\mu_{x_i}^j(x_i(t))]$ $H_{x_i}^{j,l} = H(x_i _{x_i = \mu_{x_i}^j, x_i = \mu_{x_i}^l}) = \frac{1}{N} \sum_{i=1}^N [\mu_{x_i}^j(x_i(t)) * \mu_{x_i}^l(x_i(t))]$ $\log[\mu_{x_i}^j(x_i(t)) * \mu_{x_i}^l(x_i(t))],$ where * is the chosen operation of fuzzy AND	Elimination of redundancy of the TS Choice of an optimal cardinality of term-sets of linguistic variables of the components of the TS
GA ₂ : Optimization of rule base	Minimum approximation error	$E = \sum_p E^p,$ where $E^p = 1/2(F(x_1^p, x_2^p, \dots, x_n^p) - d^p)^2$	Choice of optimal parameters of the right sides of rules
GA ₃ : Adjustment of the KB	Minimum approximation error or maximum joint information entropy	$E = \sum_p E^p$ $H_{x_i}^j$	Fine adjustment of the parameters of membership functions

The structure of the SCO for design robust ICSs is pre-

sented in Figure 8.

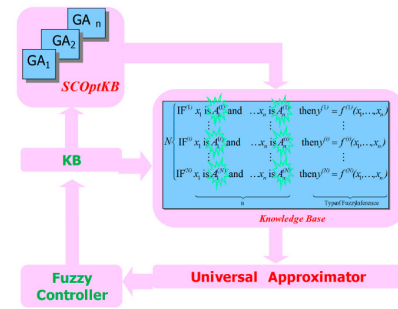


Figure 8. Structure of SCO of knowledge base SCOptKB™

The SCO consists of interrelated GA1, GA2, GA3, which optimize particular components of KB.

The input of the SCO is TS, which can be obtained either at the stage of stochastic simulation of the behavior of the controlled plant (with the use of its mathematical model) or experimentally, i.e., directly from the measurement of the parameters of the physical model of the controlled plant.

Figure 9 also presents the successive implementation of the stages of designing the SCO.

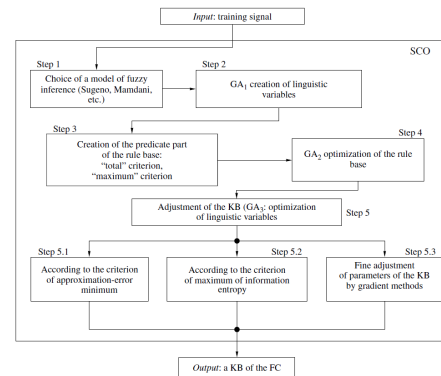


Figure 9. The algorithm of interaction of operations in the SCO

Let us specify the steps of the optimization algorithm.

Step 1. Choice of the model of fuzzy inference. The user specifies the particular type of model of fuzzy inference (Sugeno, Mamdani, etc.) and the number of input and output variables.

Step 2. Creation of linguistic variables. With the application of GA₁, an optimal number of membership functions (MF) is determined for each input linguistic variable, and an optimal form for the representation of its MFs (triangular, Gaussian, etc.) is chosen.

Step 3. Design of the rule base. At this stage, a special algorithm for selection of the most robust rules is used in accordance with the following two criteria:

- 1) "total" criterion: choose only the rules that satisfy

the following condition: $R_{total_fs}^l \geq TL$, where TL (threshold level) is a given (manually or chosen automatically)

level of rule activation, and $R_{total_fs}^l = \sum_{k=1}^N R_{fs}^l(t_k)$, and

$$R_{fs}^l(t_k) = \prod [\mu_{j1}^l((x_1(t_k))), \mu_{j2}^l((x_2(t_k))), \dots, \mu_{jn}^l((x_n(t_k)))] ,$$

where t_k are time instants, $k = 1, \dots, N$, and N is equal to the number of points in the control signal; $\mu_{jk}^l(x_k)$, $k = 1, \dots, n$ are membership functions of input variables, l is the index of the rule in the KB; and symbol “ Π ” means the operation of fuzzy conjunction (in particular, it may be interpreted as a product);

2) “maximum” criterion: choose only the rules that satisfy the condition $\max_t R_{fs}^l(t) \geq TL$.

Step 4. *Optimization of base rules.* With the help of GA₂, the right sides of rules of the KB defined at Step 3 are optimized. At this stage, a solution that is close to the global optimum is found (minimum TS approximation error). With the application of the next step, this solution can be improved locally.

Step 5. *Adjustment of the base of rules.* With the help of GA₃, the left and right sides of the rules of the KB are optimized; i.e., optimal parameters of the MFs of the input/output variables are chosen (from the viewpoint of a given fitness function of the GA). In this optimization process, three different fitness functions chosen by the user (steps 5.1 and 5.2 in Figure 9) are used. In addition, there is also the opportunity to adjust the KB with the help of conventional error-back-propagation method (step 5.3 in Figure 9).

Verification (testing) of the designed knowledge base. Constructed at stages 4, 5.1, 5.2, and 5.3 on Figure 9 KBs of the ICS are tested from the viewpoint of robustness and control quality. For further use, the best functionally KB is chosen, which is tested in the functional mode in online.

Examples of KBs simulation on the basis of efficient application of the SCO below on redundant robotic manipulators considered.

2.1. Software implementation of the soft computing optimizer

The SCO was implemented as a software system [9, 15-17]. As a programming language, C++ (Microsoft Visual Studio.net) was chosen. The algorithmic part devoted to the implementation of the main stages of optimization algorithms was implemented as a platform-independent tool (see, Appendix). The graphical interface presented in Figure 10 was developed for operating systems of the Win32

family and was tested on personal computers with different versions of the Windows operating system. The main menu of the optimizer was divided into several sections (Figure 10) devoted to execution of the main functions and visualization of the results of algorithm operation.

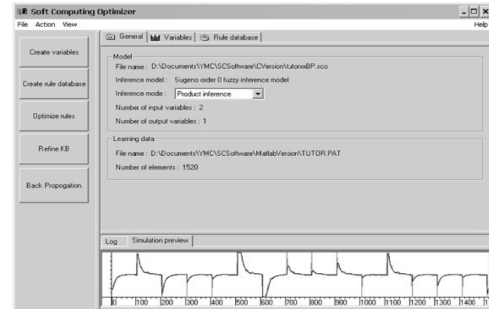


Figure 10. The main menu of the SCO

In the *left section* of the main menu, a group of buttons is located. These buttons run different optimizing components such as following:

- creation of linguistic variables (Create variables) with the help of GA₁;
- algorithm of generation of the predicate part of fuzzy rules (Create rule base);
- GA₂ for optimization of the consequence part of fuzzy rules (Optimize rules);
- GA₃, which represent the algorithm of readjustment of the parameters of linguistic variables for a more accurate approximation of TS by the obtained rules (Refine KB). The error-back-propagation algorithm is also included (Back propagation), which guarantees a given accuracy of the approximation of TS of the designed KB.

In the *central section* of the main menu of the optimizer, the basic information about the designed fuzzy system is located, such as the type, address of the main file of the KB, the number of input and output variables, as well as generic information about the TS. Here, can also find the *editor of linguistic variables* and the editor of rules.

Figure 11 presents the editor of linguistic variables. The membership functions of fuzzy variables can be edited both manually, by dragging the corresponding values, and by manual input of parameters.

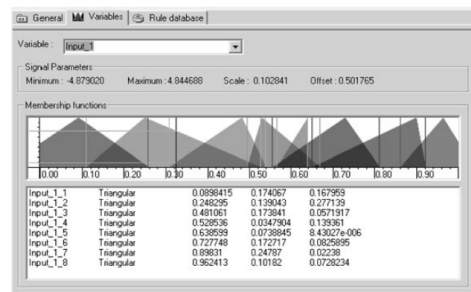


Figure 11. The editor of linguistic variables

Figure 12 presents the *editor of the base of fuzzy rules*. The fuzzy rules are structurally represented in the form of FNN. The number of neurons of the first layer corresponds to the number of input signals, while the number of neurons of the second layer corresponds to the total number of MFs involved in the linguistic variables describing the corresponding input signals.

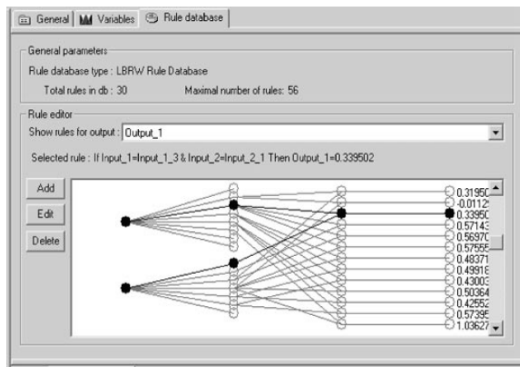


Figure 12. The editor of the base of fuzzy rules

The number of neurons of the third layer is given by the set of fuzzy rules involved in a given KB. To choose a particular rule, it is necessary to choose the corresponding neuron of the third layer. The chosen rule can be further changed and appended.

In the bottom part of the main menu of the optimizer, the window for the output of system messages is located, in which the parameters of algorithms and all actions made by the user are copied. The constantly updated result of fuzzy inference is output together with the approximating TS. Any actions aimed at a change of parameters of the designed KB results in updating the approximation results. Thus, the user can visually control the effect of modification of parameters of the KB on the result of the approximation.

In the design of this system, it was initially planned to use it together with Matlab, which allows one to flexibly compute the values of the fitness functions of GA. Note that, together with the TS, it is possible to apply the results of numerical integration of models of the controlled plant executed in the Simulink environment controlled by FC with the synthesized SCO. An approach that allows one to compute the fitness function in Matlab with the subsequent transfer of the results to the GA of the optimizer was developed^[18]. For this purpose, the corresponding library of units of the Simulink environment was designed. This library supports the loading of the KB and fuzzy inference (in the simulation mode), as well as the communication with the optimizer (in the optimization mode). The unit of fuzzy inference for Simulink was written in C++, in the form of the corresponding *S*-func-

tion of Simulink. To simulate fuzzy inference (without using Simulink models), the corresponding *.mex file was prepared, which allows one to obtain the results of fuzzy inference with the help of the command line and executed scripts of Matlab. The program is compatible with Matlab 6.1 and subsequent versions.

Since the main chain in the technology for designing ICSs is the stage of designing the corresponding KB, the design of robust KBs under the types of unpredicted control situations specified above allows one to establish in a general the accordance between the conditions of functioning of the controlled plant and the robustness level required for the ICS. Consider the results of simulation of robust structures of ICSs with efficient application of the SCO.

Remark. We are described a methodology for designing robust KBs and the corresponding software tools in the form of SCO based on soft computing, which allows one to solve the problem posed within the framework of processes of learning and adaptation. In what follows, we consider particular examples of application of the SCO in the problems of testing and evaluating the levels of structural robustness of the designed ICS based on the joint technique of stochastic and fuzzy simulation. As simulation objects, we chose benchmarks that allow us to demonstrate clearly the efficiency and advantage of the developed tools for designing the SCO.

The employed models of the controlled plant possess both local and global dynamic instabilities, high sensitivity to variation of the initial conditions, parameters of the CO structure, and random parametric, internal, and external perturbations. We present the results of simulation and practical recommendations for using them in the problems of designing robust ICS. The methodology of stochastic simulation is described in short below.

2.2. A system of stochastic fuzzy simulation of robust intelligent control systems

Fuzzy simulation of robust KBs with the SCO is based on the process of extraction of valuable information by simulation and investigation of individual (statistically represented) informative trajectories describing the behavior of the controlled plant and a conventional PID controller under the effect of stochastic processes. Within the scope of correlation theory, stochastic processes, which are different in their statistical nature (i.e., having different density functions of probability distribution), can be indistinguishable in their correlation properties. The density function of probability distribution is the complete statistical characteristics of stochastic processes. Therefore, the output process of the forming filter simulating the external

environment must be represented by the informatively significant selective trajectory of the stochastic process that allows one to investigate individual parameters of dynamic fuzzy systems. Selective trajectories should meet these requirements if their density function of the probability distribution is known. Stochastic processes with a required density function of probability distribution are simulated by the method of nonlinear forming filters.

In this section, we use the methodology of designing the structures of ICSs functioning in the external environment under the presence of stochastic processes having the same autocorrelation function and different distribution functions of the probability density. The method of nonlinear forming filters for describing stochastic processes with a required density function for the probability distribution based on the Fokker-Planck-Kolmogorov equations is described in ^[17]. This approach allows us to develop a generalized methodology for investigating the robustness of ICSs based on stochastic fuzzy simulation.

Figure 13 presents the generalized structure of the system of stochastic fuzzy simulation, which was applied for evaluating the robustness and limiting capabilities of the structures of ICSs with specifying the main factors that affect the sensitivity and reliability of control.

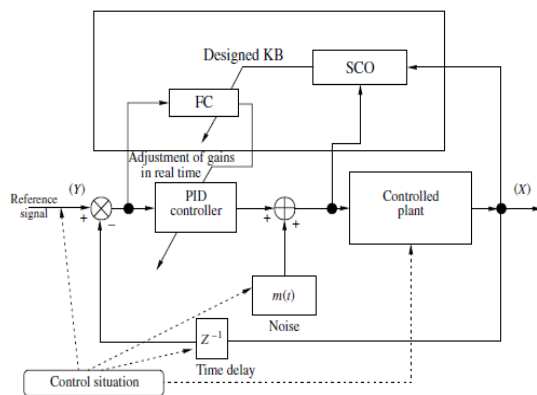


Figure 13. The block diagram of stochastic fuzzy simulation with unpredicted control situations

The efficiency of application of the SCO is demonstrated by particular typical examples of models of controlled plants, the so-called benchmarks of redundant robotic manipulators. In particular, the investigated models of physical controlled plants and their functioning environment are characterized by the following specific features typical of real dynamic controlled plants:

- they have local and global dynamic instability with respect to the generalized coordinates;
- they have essentially nonlinear cross constraints (stochastic nonlinearities) in the generalized dynamic coordinates, which mutually affect (antagonistically) the dynam-

ic, stability, and controllability of the controlled plant;

- they operate under unpredicted control situations.

As unpredicted control situations, we consider four control models under the conditions of uncertainty of the source information: (1) with statistical information about the external and parametric random time dependent perturbations (selective trajectories of stochastic processes with density functions of probability distribution depending on time); (2) with uncertainty of information about the variation of parameters of the structure of the controlled plant; (3) under the presence of random delay time in the loops of control and measurement systems; and (4) when the control (reference signal) goals are changed.

The developed model of the ICS and controlled plant was simulated in the Matlab/Simulink system presented in Figure 14.

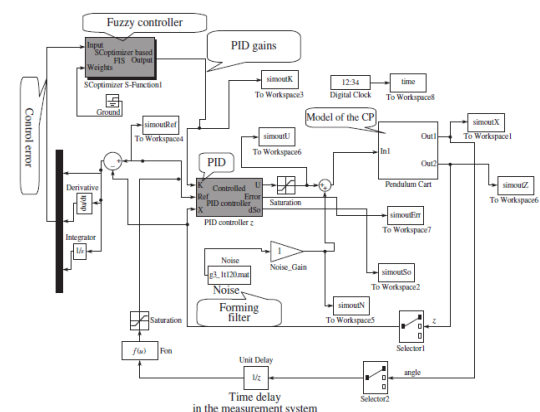


Figure 14. A Matlab/Simulink Model of the control system

As typical random noise, three types of stochastic processes with the corresponding density functions of probability distribution were simulated.

Figure 15 presents the form of the density functions of probability distribution and the simulation results of the output stochastic processes from the corresponding forming filters.

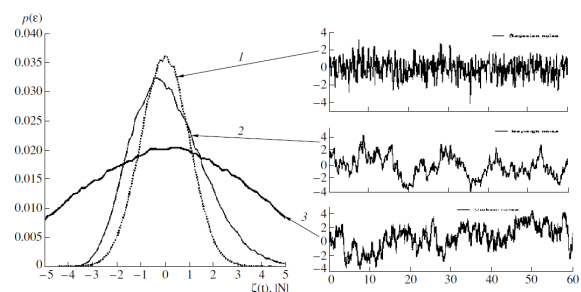


Figure 15. The form of the density function of the probability distribution and the results of simulation of output stochastic processes from the corresponding forming filters: (1) Gaussian; (2) Rayleigh; (3) uniformly distributed

Varying the structure of the forming filters, the parameters in the models of the controlled plant, the delay time in the channel for measuring the control error, and the form of the reference signal (control goal), we can simulate unpredicted control situations and evaluate the sensitivity and the robustness level of the designed ICS.

In this section, we present the results of simulating the robust control laws for intelligent fuzzy PID controllers by complex essentially nonlinear dynamic controlled plants as robotic redundant manipulators. To demonstrate the capabilities of simulation of the processes of intelligent control of a dynamic controlled plant and the conditions of functioning, the results of simulation of the following three typical controlled plants (benchmarks) are considered: (1) a nonlinear oscillator with essential dissipation and local dynamic instability; (2) an inverted pendulum mounted on a moving cart (so-called “cart-pole” system) and with global dynamic instability; and (3) an essentially nonlinear oscillator with local and global dynamic instability in cross constraints of the generalized coordinates of the controlled plant.

These oscillators are of independent interest for problems in robotics and mechanics (e.g., a stroboscopic manipulator robot with complex behavior dynamics and considerable dissipation) and allow one to compare our results with the results obtained by methods based on FNN^[19].

Remark 6. In view of the large amount of the simulation results, we consider the first version of an oscillator containing all the qualitative specific features of the two types of oscillators listed above.

3. Control System of 7DOF Manipulator

Redundant manipulators have a greater number of Degrees of Freedom (DOF) than is necessary for the task solution more than the dimension of the workspace. Redundancy DOF allows the structure of the manipulator to adapt under conditions of insufficient information about an external changing environment, as well as in conditions of changing parameters of the manipulator (for example, an obsolescence or unit failure). Redundancy DOF also allow to specify the behavior of the robot manipulator with a minimum consumption of useful resource. The control tasks for redundant robot manipulator (positioning of the end effector, trajectory describing, solving the inverse dynamics problem, etc.), with increasing CO complexity, increasing performance requirements in unexpected situations, are being solved applying computational intelligence technologies GA^[20, 21], neural and fuzzy neural networks^[22, 23], fuzzy logic^[24, 25]. The application of soft computing technologies^[26] to build a robust ICS for solv-

ing the problem of precision positioning redundant (3DOF and 7 DOF) manipulators considered.

The control system is a combination of one or more COs and a control system. In general, a control system consists of a control link, an CO, and a Measuring System (MS) in a feedback circuit. To provide the given dynamic indicators in control systems, any types of controllers are used. Widespread is Proportional Differential Integral (PID) controller. The integral component of the controller allows eliminating the static error in the system, and the differential component allows improving the dynamic performance, and forcing the overshoot process.

3.1. Control systems with constant controller parameters

In the general case, it is necessary to find the coefficients $K_P, K_D, K_I, i = \overline{1, 7}$ of the PID controller.

Initial knowledge of the control system and of CO^[27, 28] are necessary for determining the coefficients by analytical methods, correct determination of PID controller coefficients K_P, K_D, K_I is possible with the help of an expert.

The inclusion of elements of intelligent computing in the control system may allow us to describe the requirements for the control system in terms of quality criteria.

For example, we can define control parameters using GA. It is necessary to correctly determine the fitness function of the GA, for example as follows:

$$fitness = (PTS = 1) \cap (I_T \rightarrow 0),$$

where *PTS* (Position Task Solution) is the solution to the positioning problem by the manipulator, and I_T is the ICS performance.

Based on the fitness function, the choice of coefficients $K_P, K_D, K_I, i = \overline{1, 7}$ is determined on the basis of providing a guaranteed solution to the positioning problem with maximum performance.

An intelligent GA superstructure without destroying the lower executive level allows to operate with qualitative criteria of the system.

The block diagram of the ICS based on GA is shown in Figure 16, where Q_{ref} is the master reference signal, Q' is the measured variable, $K = [K_{P1} K_{D1} K_{I1}; \dots; K_{P7} K_{D7} K_{I7}]$ is the coefficient matrix of the PID controller, $s(t)$ is the limitation of the control action, $d(t)$ is the delay in the MS, $m(t)$ is the external influence.

The selection of the PID controller coefficients in the control system based on GA is made once for one or a number of cases (regular control situations) and remain unchanged during operation. As a result, control system based on GA gets a good result with the task of accurately positioning the manipulator in standard situations. However, control system does not provide guaranteed control in unexpected control situations, which will be demonstrated below.

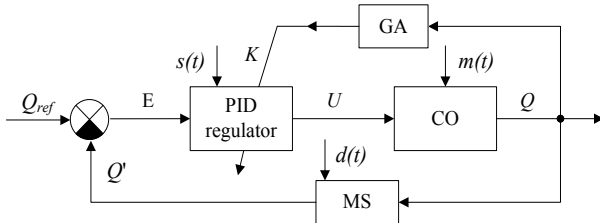


Figure 16. The block diagram of the control system based on GA

The use of the control system based on GA is limited by the requirement for a description of the constant environmental conditions and known structures of the control unit and CO.

Expanding the scope of the control system is possible by increasing the intelligence of the control system: using dynamic tuning of the PID controller coefficients, which is possible with the elements of soft computing technology.

3.2. Designing an intelligent control system based on Soft Computing Optimizer

FC is the main element of the ICS based on soft computing technologies^[28], FC manages the gain of the PID controller due to the integrated KB, which includes data on the form and parameters of MFs of input and output fuzzy variables, and fuzzy production rules.

KBs are created using the intelligent tools the KBO based on soft computing^[29] in the following sequence:

- 1) creating TS: determining a typical control situation (for example, a standard situation), generating a table of PID controller coefficients and control errors using a GA;
- 2) organization of a fuzzy inference model: determining of the type of fuzzy model, interpreting fuzzy operations, the number of input and output variables;
- 3) creating linguistic variables for input values;
- 4) creating a rule base;
- 5) setting up the rule base;
- 6) optimization of the left and right parts of the rules of the KB.

ICS based on KBO on soft calculations may contain one or more FC depending on the complexity of the sys-

tem and CO. In the case of a simple CO, it is possible to implement one FC, respectively, with a single KB (Figure 17). However, with the increasing complexity of the CO, the time of creating the KB increases, the requirements for the computing resources of the processor on which the KB is created and the amount of memory of the system in which the KB is located increase.

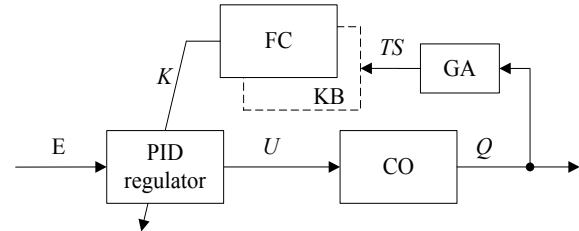


Figure 17. ICS based on KBO on soft computing with one FC

When the complexity of implementing a single KB is high, several KBs are created that are located in different FCs (Figure 18).

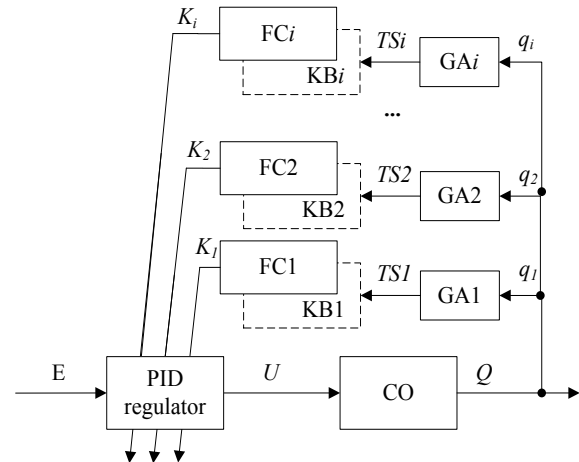


Figure 18. ICS based on KBO on soft computing with shared control

Separation of management somewhat reduces the quality of the system. However, the creation of several FCs is often the only way to organize intellectual management of complex CO.

Let us consider in more detail the process of creating a KB for ICS based on KBO on soft computing for a robot manipulator with 7 DOF.

Due to the complexity of the CO under consideration, the implementation of a single KB is impossible, therefore, we will initially organize a divided link management (one FC controls one link, as shown in Figure 19).

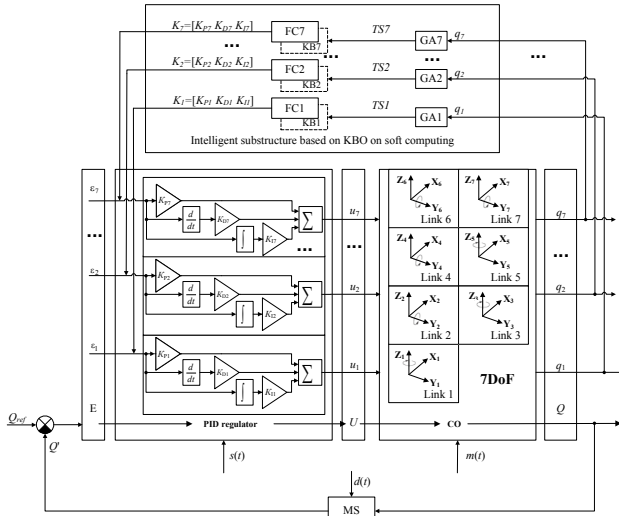


Figure 19. ICS 7DOF manipulator based on KBO on soft computing

In Figure 19: Q_{ref} is the reference signal, Q' is the measured variable, $E = [\varepsilon_1 \ \varepsilon_2 \ \dots \ \varepsilon_7]$ is the control error, K is the matrix of proportional, differential and integral coefficients of the PID controller $K_{pi}, K_{Di}, K_{Ii}, i = \overline{1, 7}$, where i is the number of the corresponding link of the robot manipulator, $s(t)$ is the limitation of the control action, $U = [u_1 \ u_2 \ \dots \ u_7]$ is the control action, $d(t)$ is delay in the Measuring System (MS), $TS_i, i = \overline{1, 7}$ is TS of the corresponding FC, $m(t)$ is external environmental impact, $Q = [q_1 \ q_2 \ \dots \ q_7]$ is adjustable value [30, 31].

The modeling of 7DOF manipulator control systems was carried out to study the quality of the considered control systems in the environment of MatLab/Simulink.

3.3. The model of Control Object

A formalized model of the 7DOF manipulator was built under the assumption that the links of the robot of the manipulator can rotate in the range of $(-70 \ +70)$ degrees. The degree of freedom configuration corresponds to:

- 1 link) vertical axis of rotation α_{Z1} ;
- 2 link) transverse α_{Y2} ;
- 3 link) vertical α_{Z3} ;
- 4 link) transverse α_{Y4} ;
- 5 link) vertical α_{Z5} ;
- 6 link) transverse α_{Y6} ;
- 7 link) transverse α_{Y7} .

The CO model and formulas for determining the coordinates of the links of the manipulator were available in earlier works [32].

Creating a real CO model allowed accelerating the

identification of the CO model, obtaining acceptable control parameters for different types of control systems and with a different level of intelligence.

To demonstrate the advantages and disadvantages of the considered types of control systems as applied to 7DOF manipulator, a series of experiments for MatLab/Simulink models was performed in this work.

Consider the test procedure order.

3.4. Test Procedure

A series of experiments is necessary to identify the advantages of various types of control systems of the 7DOF manipulator in both standard and unexpected control situations.

To test the robustness of control system models, a series of experiments is carried out, consisting of two stages: 1) work in standard control situations; 2) work in unexpected management situations.

As standard control situations, thirteen experiments are performed in accordance with the group of test points of the working space (Figure 20). The configuration is taken as the initial position of the manipulator:

$$Q = [q_1 \ q_2 \ q_3 \ q_4 \ q_5 \ q_6 \ q_7] = [0000000] \text{deg}.$$

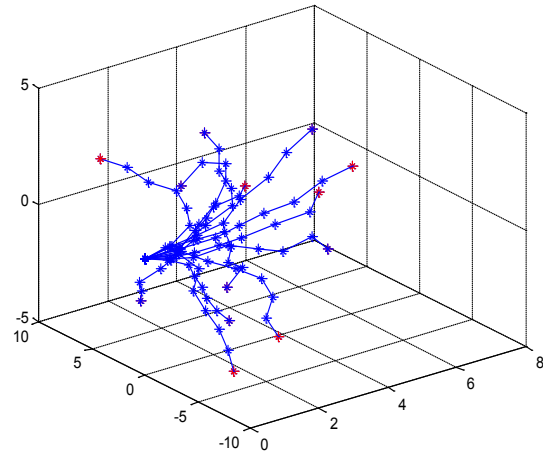


Figure 20. Test workspace

Unexpected situations are divided into external and internal. External unexpected situations:

- 1) forced change in the position of the links (Figure 21):
 - the first link to a value of -30 degrees at the 25th iteration and to a value of 30 degrees at the 75th iteration;
 - the second link to a value of -30 degrees at the 50th iteration and to a value of 30 degrees at the 100th iteration;
 - the third link to a value of -30 degrees at the 50th iteration and to a value of 30 degrees at the 100th iteration;
 - the fourth link to a value of -30 degrees at the 50th iteration and to a value of 30 degrees at the 100th iteration;
 - the fifth link to a value of -30 degrees at the 50th iteration and to a value of 30 degrees at the 100th iteration;

- the sixth link to a value of -30 degrees at the 50th iteration and to a value of 30 degrees at the 100th iteration;
- the seventh link to a value of -30 degrees at the 50th iteration and to a value of 30 degrees at the 100th iteration;

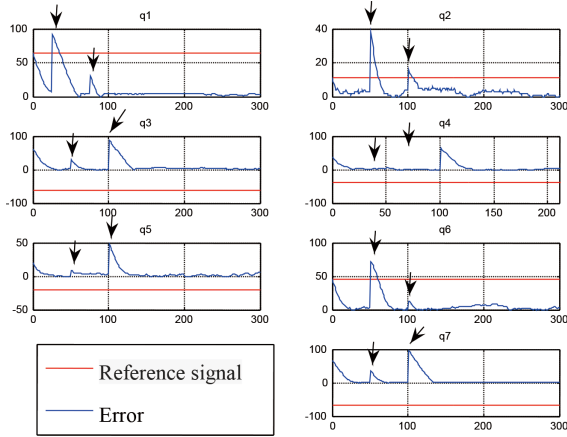


Figure 21. Forced change of links position

2) the initial conditions are changed

$$Q = [q_1; q_2; q_3; q_4; q_5; q_6; q_7] = [-30; -30; -30; -30; -30; -30; -30] \text{ degrees ;}$$

3) at the same time a forced change in the positions of the links and a change in the initial conditions are carried out.

Each of the three external unexpected situations is tested at thirteen points of the test space; thus, 39 experiments are conducted for external contingency management situations.

Internal unexpected situations:

- 1) reduction of restrictions of control actions;
- 2) an increase in the limitations of control actions;
- 3) introduction of noise into the control channels (Figure 22);

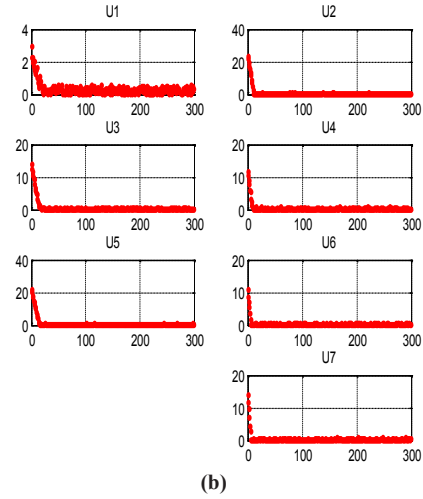
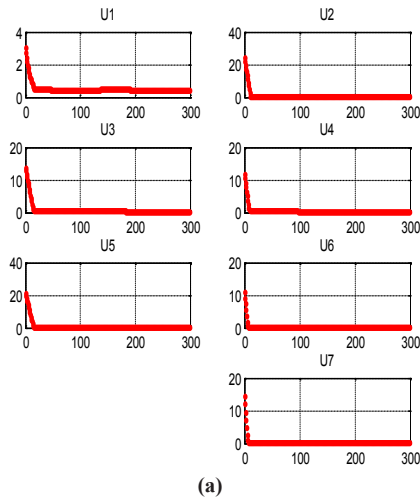


Figure 22. Control actions: initial (a); after making noise (b)

4) the introduction of errors in the MS ($\pm 1,5$ degrees).

Each internal unexpected situation is tested at thirteen points of test space; thus, 52 experiments are conducted for internal unexpected control situations.

3.5. Definition of quality criteria

We introduce a system of quality criteria that takes into account methods of the theory of automatic control^[28] to evaluate and compare the results of tests of control systems with constant PID regulator coefficients and ICS based on KBO on soft calculations.

These methods have been adapted for a specific CO 7DOF manipulator in the following form:

1. Position Task Solution in known control situations PTS_{KCS} .

The positioning problem is considered to be solved if, upon completion of a given number of iterations $I_{max} = 300$, the condition is satisfied:

$$\begin{cases} PTS = 1, \text{ if } |q_{1ref} - q_1| + |q_{2ref} - q_2| + \dots + |q_{7ref} - q_7| \leq 2 \text{ deg} \\ PTS = 0, \text{ else} \end{cases}$$

where $q_{1ref}, q_{2ref}, \dots, q_{7ref}$ are the desired positions of the links, $q_1, q_2, \dots, q_7$ are the current positions of links

$$PTS_{\text{implementation}} \equiv \frac{\sum_{i=1}^N PTS}{N},$$

where N is the number of experiments.

2. Position Task Solution in the external above considered control situations PTS_{ACCS1} .

3. Position Task Solution in the internal above consid-

ered control situations PTS_{ACCS2} .

4. Performance I_T

The number of iterations from the beginning of the impact during which each of the links is positioned with an allowable error $2\Delta < 1 \text{ deg}$:

$$I_T = I \left| \left(|q_{1ref} - q_1| < 1 \text{ deg} \right) \cap \left(|q_{2ref} - q_2| < 1 \text{ deg} \right) \cap \dots \cap \left(|q_{7ref} - q_7| < 1 \text{ deg} \right) \right|,$$

$$I_T \text{ implementation} \equiv 1 - \frac{\sum_{i=1}^N I_T}{NI_{\max}}.$$

5. Relative overshoot value σ

The ratio of the maximum deviation of the current position of the link from the steady-state value to the steady-state value:

$$\sigma = \max \left[\max \left| \frac{q_{1ref} - q_1}{q_{1ref}} \right|, \max \left| \frac{q_{2ref} - q_2}{q_{2ref}} \right|, \dots, \max \left| \frac{q_{7ref} - q_7}{q_{7ref}} \right| \right],$$

$$\sigma \text{ implementation} \equiv 1 - \frac{\sum_{i=1}^N \sigma}{N}.$$

6. Relative error in link positioning after completion of a given number of iterations ε

$$\begin{cases} \ddot{a} = \left| \frac{q_{1ref} - q_1}{q_{1ref}} \right|, f \left(|q_{1ref} - q_1| > |q_{2ref} - q_2| \right) \cap \left(|q_{1ref} - q_1| > |q_{3ref} - q_3| \right) \cap \dots \cap \left(|q_{1ref} - q_1| > |q_{7ref} - q_7| \right) \\ \ddot{a} = \left| \frac{q_{2ref} - q_2}{q_{2ref}} \right|, f \left(|q_{2ref} - q_2| > |q_{1ref} - q_1| \right) \cap \left(|q_{2ref} - q_2| > |q_{3ref} - q_3| \right) \cap \dots \cap \left(|q_{2ref} - q_2| > |q_{7ref} - q_7| \right) \\ \dots \\ \ddot{a} = \left| \frac{q_{7ref} - q_7}{q_{7ref}} \right|, f \left(|q_{7ref} - q_7| > |q_{1ref} - q_1| \right) \cap \left(|q_{7ref} - q_7| > |q_{2ref} - q_2| \right) \cap \dots \cap \left(|q_{7ref} - q_7| > |q_{6ref} - q_6| \right) \end{cases}$$

$$\varepsilon \text{ implementation} \equiv 1 - \frac{\sum_{i=1}^N \varepsilon}{N}.$$

7. One iteration time t

Execution time of one iteration I :

$$t \text{ implementation} \equiv 1 - \frac{t}{t_{const}}, t < t_{const}.$$

8. Implementation complexity P

Evaluation of changes in control coefficients:

$$P \text{ implementation} \equiv 1 - \frac{\sqrt{\frac{1}{N} \int_{i=1}^N \left(\frac{dK}{dt} \right)^2 dt}}{\max(K)}.$$

9. Full Control Behavior FCB

$$FCB \equiv w_1 \cdot P[PTS_{KCS}] + w_2 \cdot P[PTS_{ACCS1}] + w_3 \cdot P[PTS_{ACCS2}] + w_4 \cdot P[I_T] + w_5 \cdot P[\sigma] + w_6 \cdot P[\varepsilon] + w_7 \cdot P[t] + w_8 \cdot P,$$

where $w = [0,1 \ 0,2 \ 0,2 \ 0,2 \ 0,05 \ 0,1 \ 0,1 \ 0,05]$ are weights.

3.6. PID Constant Control Systems

The control task is reduced to finding the coefficients of the PID controller $K_P, K_D, K_I, i = \overline{1,7}$, which ensures the desired nature of the movement of the manipulator. In this section, we consider two types of control systems with constant coefficients: a control system on a PID controller and based on GA.

A comparison of the operation of 7DOF manipulator ACS based on the PID controller and based on GA in accordance with the introduced system of quality criteria is given in Table 2, and in Figure 23.

Table 2. Comparison of the operation of control systems with constant coefficient

Quality Criteria		based on PID	based on GA
1	PTS_{KCS}	0,000	0,615
2	PTS_{ACCS1}	0,000	0,256
3	PTS_{ACCS2}	0,058	0,308
4	I_T	0,000	0,008
5	σ	0,892	0,956
6	ε	0,379	0,657
7	t	0,998	0,998
8	P	1,000	1,000
9	FCB	0,244	0,439

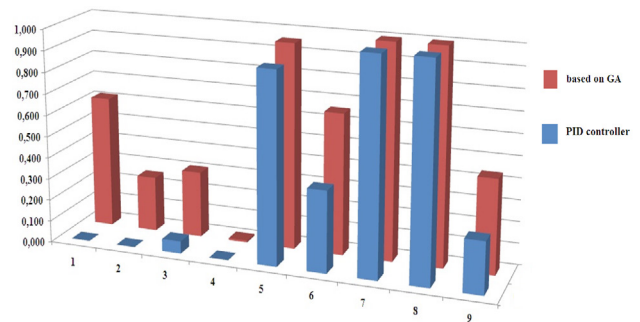


Figure 23. Comparison of the results of MatLab/Simulink control systems models based on the PID controller and with the use of GA

From the results of comparing control systems on PID controller and using the GA, we conclude:

1) applying the control system based on PID controller,

the positioning task was not solved in normal situations and external unexpected control situations, insignificant positive results (3/52 experiments) were obtained for internal unexpected control situations;

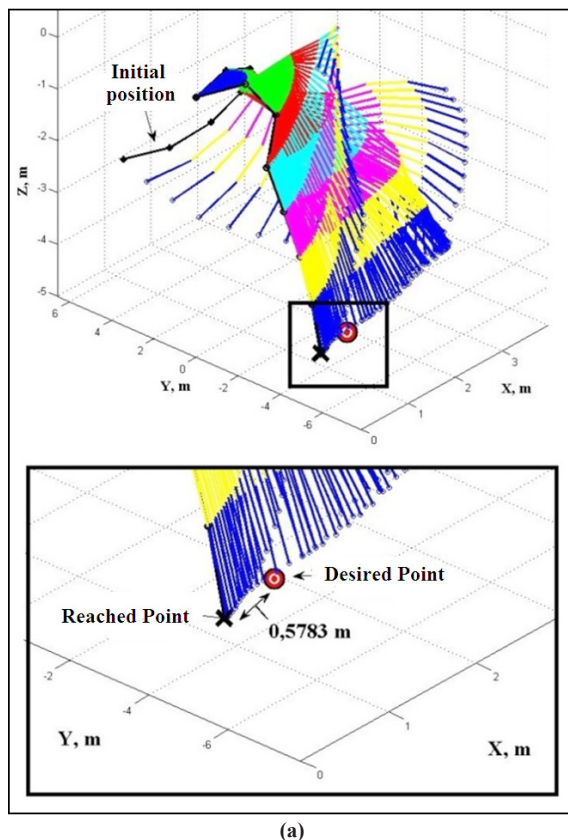
2) some improvement is achieved by using a control system based on GA: the positioning problem is solved in standard control situations in most experiments, but in unexpected control situations (external and internal), the solution is achieved in less than a third of the experiments;

3) both systems with constant PID controller coefficients have low speed;

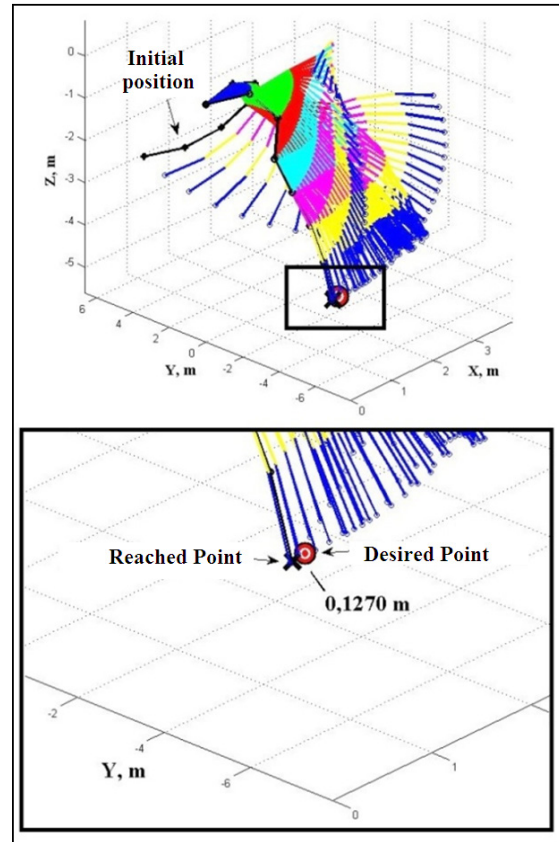
4) applying the control system based on the GA, the relative values of overshoot and positioning errors are significantly improved compared to the control system on the PID controller;

5) applying the control system based on the GA, the Full Control Behavior in comparison with the control system on the PID controller improves.

In Figure 24 demonstrates the operation of the manipulator when using control systems on the PID controller and using the GA in the conditions of the third external unexpected control situation (the initial position has been changed and the links are forced to move at different times). In this experiment, the control system based on GA solves the control problem, unlike to the control system on the PID controller.



(a)



(b)

Figure 24. The movement of the manipulator in an external unexpected situation: ACS based on PID controller (a); GA control systems (b)

Despite the fact that the control system on the GA significantly improves the assessment of quality criteria compared to the ACS on the PID controller, the overall quality of control provided by the control system on the GA is rather low.

In the process of control, the PID controller coefficients for the considered structures do not change. This simplifies the control system design, but at the same time deprives the control system of the possibility of rebuilding and adaptation.

Next, we consider a structure with dynamic adaptation of the PID controller coefficients, implemented on the basis of soft computing technologies.

3.7. Quality of Control System based on Soft Computing Technologies

Testing of the obtained KB1 - KB7, respectively, FC1 - FC7 is carried out as part of the ICS based on soft computing.

The results of ICS based on soft computing tests and control systems with constant coefficients (based on the PID controller and using GA) in accordance with the introduced quality criteria are shown in Table 3 and in Figure 25.

Table 3. Results comparison of the of control systems with constant coefficients and ICS based on KBO on soft computing

Quality Crite- ria	ICS based on soft comput- ing	based on GA	based on PID
1 PTS_{KCS}	0,923	0,615	0,000
2 PTS_{ACCS1}	0,744	0,256	0,000
3 PTS_{ACCS2}	0,923	0,308	0,058
4 I_T	0,092	0,008	0,000
5 σ	0,969	0,956	0,892
6 ε	0,911	0,657	0,379
7 t	0,973	0,998	0,998
8 P	0,946	1,000	1,000
9 FCB	0,728	0,439	0,244

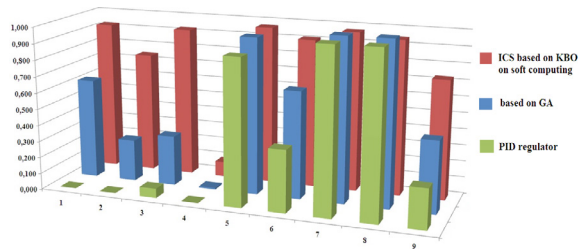


Figure 25. Comparison of the work of MatLab/Simulink ICS models based on KBO on soft computing and control systems with constant coefficients

From the results of the comparison of control systems (ICS based on soft computing, based on the PID controller and using GA) we conclude that when using the ICS based on soft computing:

- 1) the quality criterion position task solution in known control situations has increased compared to control systems with constant coefficients (based on the PID and using GA), the solution is positive in 12 out of 13 experiments;
- 2) the position task solution in the unexpected considered control situations has increased significantly compared to control systems with constant coefficients: 2,9 times for external unexpected situations and 3 times for internal unexpected situations (in comparison with the ICS based on GA);
- 3) the performance has increased significantly: more than 10 times in comparison with the ICS based on GA; however, as before, the performance is rather low;
- 4) the quality criterions relative overshoot value and relative error in link positioning improved compared to control systems with constant coefficients; but criterions one iteration time and implementation complexity have deteriorated somewhat;
- 5) the full control behavior is improved 1,7 times compared with the control system using GA and 3 times compared with the PID controller based control system.

In Figure 26 demonstrates the operation of the manipulator when using a control system based on GA and ICS based on KBO on soft computing in the conditions of the

first external unexpected control situation (the links are forced to move at different points in time).

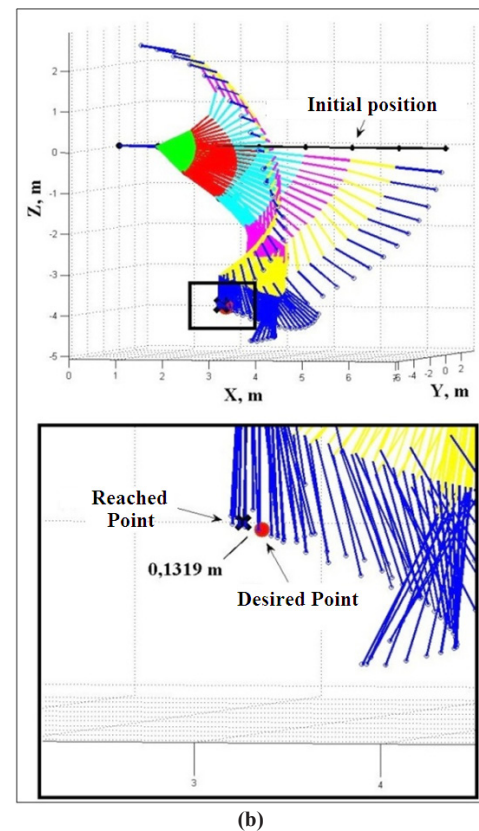
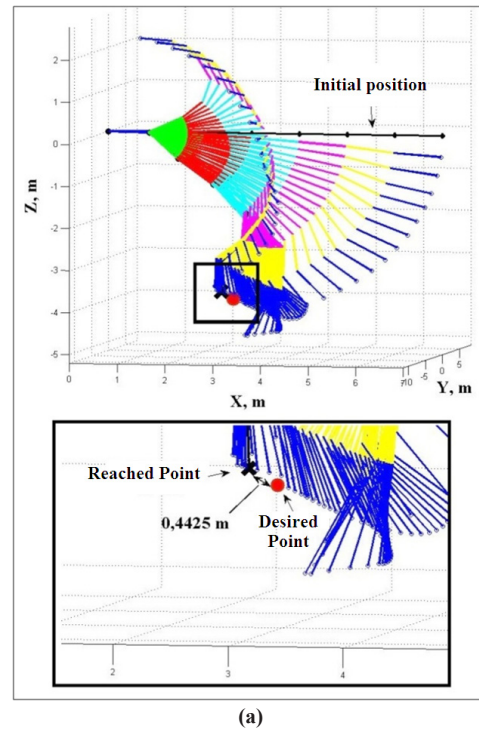


Figure 26. The movement of the manipulator in an external unexpected situation: the result of the control system based on the GA (a); ICS based on KBO on soft computing (b)

In Figure 27 shows the operation of the manipulator when using a control system based on GA and ICS based on KBO on soft computing in the conditions of the fourth internal unexpected control situation (introducing errors into the MS).

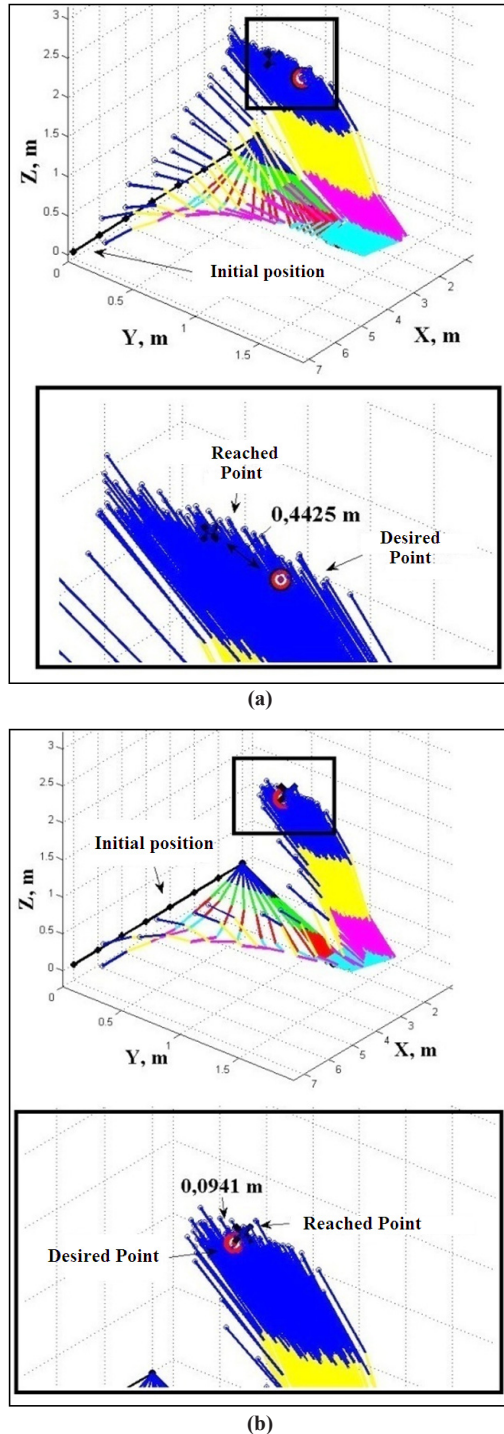


Figure 27. The movement of the manipulator in an internal unexpected situation: the result of the control system based on the GA (a); ICS based on KBO on soft computing (b)

In Figure 28 shows a comparison of phase portraits when using a control system based on GA and ICS on KBO on soft computing for the considered control situation.

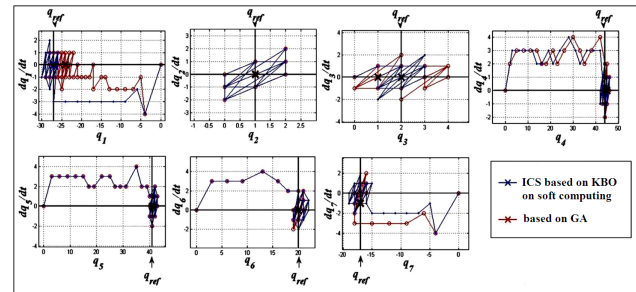


Figure 28. Changing the position of the links of the manipulator in the conditions of an internal unexpected situation: control systems based on GA and ICS based on KBO on soft computing.

The ICS by the 7DOF manipulator based on KBO on soft computing significantly improves the quality of control compared to control systems with constant coefficients (based on the PID controller and using the GA), however, the performance indicator remains at a fairly low level.

The ICS based on KBO on soft computing was organized with a separation of control: each link of the manipulator corresponds to one independent FC due to the fact that the CO is complex. Decomposition of control leads to a mismatch of work and some decrease in the quality of management.

It is possible to organize coordination control without significantly increasing the complexity of the system by introducing additional generalizing superstructure, the implementation of which is possible using quantum computing technologies, which will be discussed in the next part of the article.

Next, we consider a simpler example of an CO: this is a 3DOF robot manipulator, often used both in industry and in training.

4. 3DOF Manipulator control systems

The robot control systems for the 3DOF manipulator will be considered both at the simulation level and at the physical level. To demonstrate the quality of control systems, a test bench of 3DOF robot manipulator was developed.

4.1. Description of the 3DOF Manipulator Test Bench

In Figure 29 shows the test bench used to test control systems.

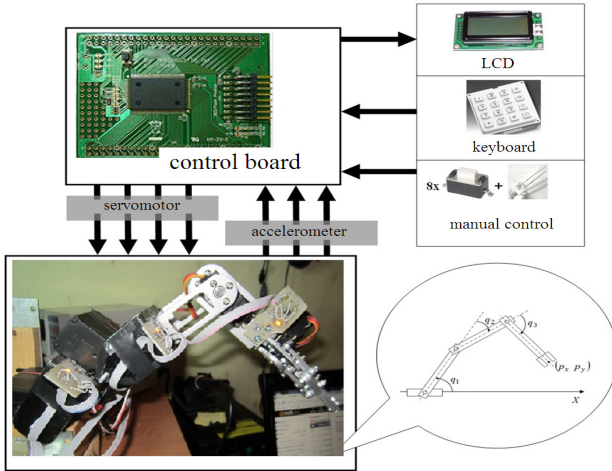


Figure 29. The manipulator test bench

As the MS (accelerometer on Figure 14), the board uses three boards with accelerometer installed on them with 3DOF ADXL335. The Renesas microcontroller is the core of the system (control board on Figure 14). Information about the current positions of the links and the characteristics of the quality of control is displayed on the LCD and serial interface. Both automatic and manual control modes are supported (the ability to move each of the 3 links and the manipulator's grip device using the manual control buttons). In robotics, as a rule, a mathematical model of the manipulator is built, simulation of the CO, identification of the parameters of the mathematical model, and then comparison of the simulation results on the mathematical model of the CO and a test bench robot manipulator are performed^[27, 28]. In contrast to the traditional approach, in this case, the behavior of the links of the robot test bench was formalized by the correspondence tables “width of the servo drive control pulse ~ angle of movement”, which allowed us to describe the behavior of the test bench in the MatLab/Simulink environment. The manipulator test bench was created without involving the mathematical model.

The creation of a formalized manipulator model allowed accelerating the identification of the CO model and obtaining acceptable control parameters.

4.2. Management Tasks

We examine the direct circuit of the control loop by the 3DOF manipulator to explain the operation of the PID controller.

In Figure 30: $E = [\varepsilon_1 \ \varepsilon_2 \ \varepsilon_3]$ is a control error, $K_{Pi}, K_{Di}, K_{Ii}, i = 1, 3$ is the proportional, differential and integral coefficients of the PID controller, i is the number of the corresponding link of the robot manipulator, $U = [u_1 \ u_2 \ u_3]$ is the control action, $Q = [q_1 \ q_2 \ q_3]$ is

an adjustable value^[30].

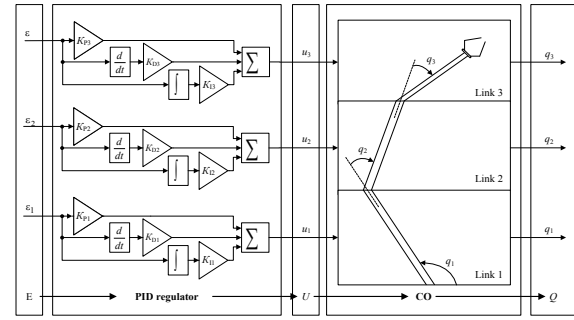


Figure 30. Direct circuit of control system with PID controller

The control task is reduced to finding the coefficients of the PID controller $K_{Pi}, K_{Di}, K_{Ii}, i = 1, 3$, which ensures the desired movement.

4.3. Test Procedure

A series of experiments is carried out for each of the considered types of control systems: based on GA, ICS based on KBO on soft computing with one FC and ICS based on soft computing with separated control.

A series of experiments is carried out in standard and unexpected control situations and is evaluated according to the quality criteria introduced above. As standard control situations, ten experiments are performed in accordance with a group of workspace test points (Figure 31).

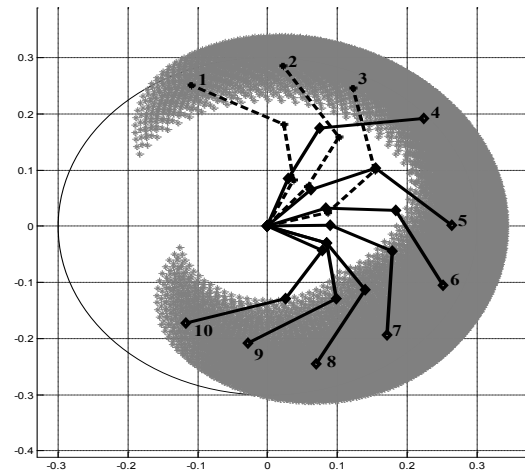


Figure 31. Test points

Configuration $Q = [q_1; q_2; q_3] = [60; 0; 0]$ degrees taken as the initial position of the manipulator.

Three cases act as unexpected control situations:

- 1) the position of the second link is changed to a value $q_2 = 45$ degrees at the 11th iteration;
- 2) initial conditions are changed

$$Q = [q_1; q_2; q_3] = [60; 45; -43] \text{ degrees};$$

3) the initial conditions are changed

$Q = [q_1; q_2; q_3] = [60; 45; -43] \text{ degrees}$; and the position of the second link is changed to the value $q_2 = 45 \text{ degrees}$ at the 11th iteration.

Three unexpected situations are tested at ten points in the test space. Thus, 30 experiments are conducted for unexpected control situations.

Consider the features of the design of ICS based on KBO on soft computing for 3DOF robot manipulator.

4.4. ICS based on SCSOptKB™

FC with a built-in KB that controls the gain of the PID controller is the main elements of the ICS based on soft computing technologies. Implementation of the ICS based on KBO on soft computing for a 3DOF robot manipulator is possible both with one FC and with separated control.

Let us consider the process of creating KB for ICS based on KBO on soft computing.

1. *Creating TS.* Define a typical control situation. As typical control situations, we will consider standard control situations.

Three of the standard experiments were used to create TS1, TS2 and TS3, for which control situations in which the parameters of the PID controller were determined using GA were reproduced using MatLab/Simulink models.

The considered TS1-TS3 are tables where columns 1-9 are input values [errP1, errD1, errI1, errP2, errD2, errI2, errP3, errD3, errI3], and columns 10-18 are output values [KP1, KD1, KI1, KP2, KD2, KI2, KP3, KD3, KI3].

Input values are vectors of input variables of proportional, differential and integral errors of the first, second and third links of the manipulator. The output values are the vectors of the output of certain GA variables of proportional, differential and integral coefficients of the PID controller of the first, second and third links of the manipulator.

The final TS used to obtain the KB consists of sequentially connected TS1, TS2 and TS3.

2. Definition of a fuzzy inference model.

The following parameters must be defined:

1) type of fuzzy model: Sugeno 0 (zero order);
2) interpretation of fuzzy operations: fuzzy conjunction as a product;

3) the number of input and output variables: 9 and 9.

3. Creating linguistic variables for input values.

The optimal number and form of MFs are determined using the GA from the KBO software.

At the first stage of creating the KB, we set the task of creating five MFs for each of the nine input variables, i.e. the vector $[n1 \ n2 \ n3 \ n4 \ n5 \ n6 \ n7 \ n8 \ n9] = [5 \ 5 \ 5 \ 5 \ 5 \ 5 \ 5 \ 5 \ 5]$, which would lead to the creation of $n1 \times n2 \times n3 \times$

$n4 \times n5 \times n6 \times n7 \times n8 \times n9 = 1953125$ fuzzy rules. At the second stage, as a result of the GA operation, the vector $[n1 \ n2 \ n3 \ n4 \ n5 \ n6 \ n7 \ n8 \ n9]$ took the value $[4 \ 4 \ 4 \ 4 \ 3 \ 4 \ 4 \ 3 \ 3]$, and the maximum number of fuzzy rules was 110592.

4. Creating a rule base.

As a result of the work, the algorithm for selecting rules (passing the specified activation threshold) selected 33 of the most robust rules out of 110592.

5. Setting up the rule base and optimization of the left and right parts of the rules of the KB.

The traditional method of error back propagating is used at this stage.

In the considered example, the maximum number of fuzzy rules for 3-4 MFs was 110592 rules. We calculate the maximum number of fuzzy rules for 3,4,5,6 and 7 MFs for each input variable. Then the dependence of the maximum number of fuzzy rules on the number of degrees of freedom of the manipulator has the form shown in Figure 32.

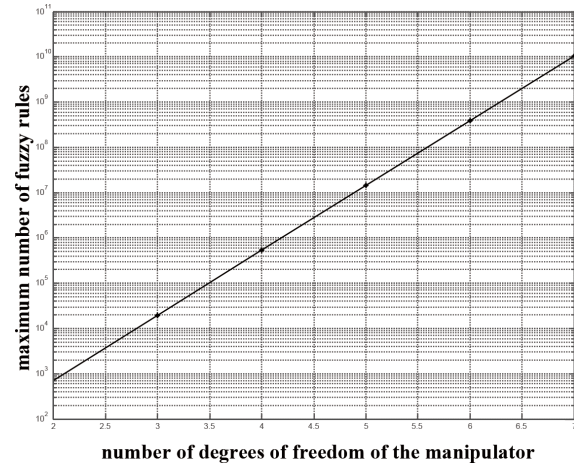


Figure 32. The dependence of the maximum number of fuzzy rules on the number of degrees of freedom of the manipulator

The introduction of additional links, the expansion of the functions of existing units, or the addition of other devices requiring coordination control will increase the maximum number of fuzzy rules by more than one and a half orders of magnitude. As a result, the complexity and time of creating KB will increase, the requirements for the computing resources of the processor and the memory capacity of the system in which the KB is located will also increase.

If it is difficult to implement a single KB, we will divide the KB into several, and use several FCs.

Consider the separation of control, in which one FC controls one link of the manipulator.

It is necessary to create 3 KBs for 3 FC respectively. The number of input and output variables for each of the KBs will decrease 3 times, and the maximum number of

fuzzy rules will decrease.

Let us consider the process of creating KB

1. Creating TS.

We created 3 TSs for 3 KBs. Each of the TS, consists of two TSs based on two different experiments.

TS1, TS 2 and TS 3 for creating 3 independent KSs contain a vector of input variables in the left columns, and vectors of output variables of certain GAs in the right columns. Input variables are proportional, differential and integral errors ([errP1, errD1, errI1], [errP2, errD2, errI2] and [errP3, errD3, errI3] for the first, second and third links of the manipulator. Output variables are proportional, differential and integral coefficients of the PID controller [KP1, KD1, KI1], [KP2, KD2, KI2] and [KP3, KD3, KI3] for the first, second and third links of the manipulator.

2. Definition of a fuzzy inference model.

The following parameters must be defined for each of KB:

- 1) type of fuzzy model: Sugeno 0;
- 2) interpretation of fuzzy operations: fuzzy conjunction as a product;
- 3) the number of input and output variables: 3 and 3.

3. Creating linguistic variables for input values.

The optimal number and form of MFs are determined using the GA1 from the KBO software.

The number of functions during the creation of KB1, KB 2 and KB 3 and optimization of GA1 was [3 3 5], [5 5 9] and [7 7 8], the number of fuzzy rules corresponds to 45, 225 and 392.

4. Creating a rule base.

18 out of 45 rules were selected for KB1, 26 out of 225 rules were selected for KB2, 48 out of 392 rules were selected for KB3.

The maximum number of fuzzy rules when creating single KB with one FC was 110592, of which 33 most robust ones were selected. The maximum number of rules in the case of separated control is 392 for KB3, which significantly reduces the time for selecting the most robust rules.

However, the total number of selected rules $18 + 26 + 48 = 92$ is more than 2 times higher than the number of selected rules when using one FC.

Consequently, the placement of the final KBs when using the ICS based on soft computing with separate control will require a larger amount of memory of the final device in which the control system is located.

5. Setting up the rule base and optimization of the left and right parts of the rules of the KB.

The traditional method of error back propagating is used at this stage.

4.5. Modeling and test bench: control quality

In Figure 33 and Figure 34 show a comparison of control

quality criteria for a control system based on GA, ICS based on KBO on soft computing with one FC and ICS based on soft computing with separated control for MatLab/Simulink models and the robot manipulator test bench.

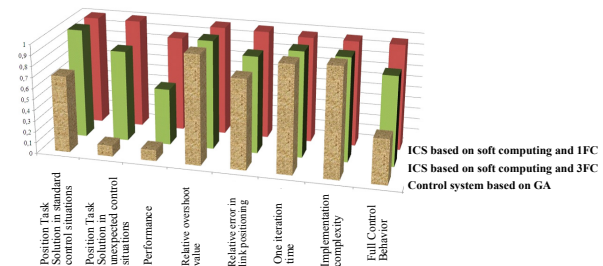


Figure 33. Comparison of quality criteria for a control system based on GA, ICS based on KBO on soft computing with one FC and ICS based on soft computing with separated control for MatLab/Simulink models

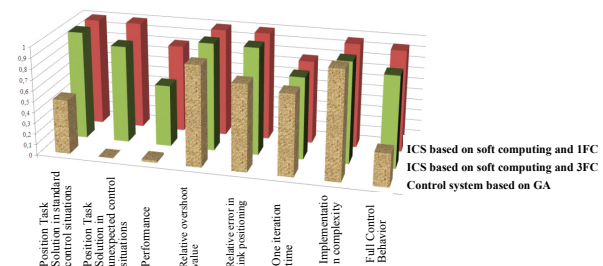


Figure 34. Comparison of quality criteria for a control system based on GA, ICS based on KBO on soft computing with one FC and ICS based on soft computing with separated control for the robot manipulator test bench

It can be seen from the comparison results that the use of the control system based on GA solves the problem of accurate positioning in half of the standard situations. The control system based on GA does not provide guaranteed control in unexpected control situations (as shown in Figure 35). The full control behavior is rather low. In Figure 35 shows the movement of the manipulator in an external unexpected situation.

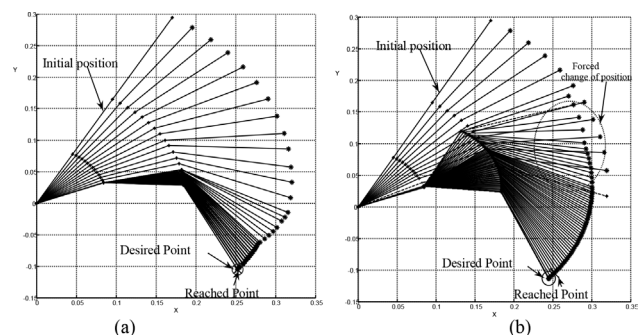


Figure 35. The operation of the control system based on GA: in a standard control situation (a); in an unexpected control situation (b)

The coefficients of the PID controller in the control system based on GA do not change. This facilitates the design of the control system, but deprives the control system of the possibility of rebuilding and adaptation.

In Figure 36 shows the work of the ICS based on KBO on soft computing with one FC and separated control in an unexpected control situation, previously proposed for a control system based on GA (Figure 35).

From Figure 35 and Figure 36, we conclude that both of ICS based on the KBO using soft computing technologies, in contrast to the control system based on GA, solve the problem of accurate positioning. ICS using a single KB provides a solution for fewer iterations than the structure of ICS with separated control.

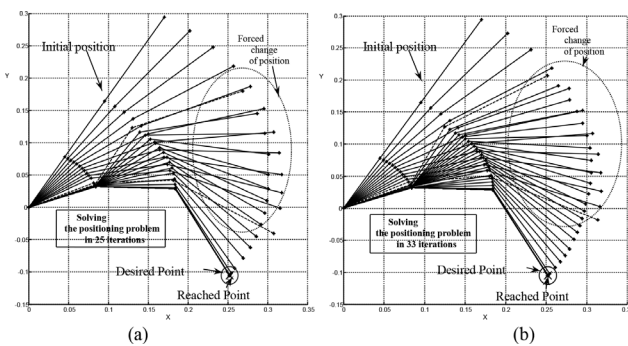


Figure 36. The operation of the ICS based on KBO on soft computing with one FC in an unexpected control situation (a); ICS based on soft computing with separated control (b)

The use of ICS based on KBO on soft computing with one FC allows:

- 1) to obtain maximum of quality criteria position task solution in standard and unexpected control situations;
- 2) to improve all quality criteria, except for the one iteration time and the implementation complexity, because dynamic adjustment of coefficients requires additional calculations;

ICS based on KBO on soft computing with one FC allows you to collect in a single KB information on the mutual behavior of 3 links of the robot manipulator at the same time, however, the high complexity of the implemented KB requires significant computational resources to create and placement.

Dividing of the control link into 3 independent FCs (one KB controls one link) allows, due to a certain decrease in the quality of management, to significantly simplify the processes of creating, optimizing and placing the KB.

It can be seen from the comparison results that when using the ICS based on KBO on soft computing with divided control with 3 FCs, all quality indicators are somewhat deteriorated, which occurs as a result of the mismatch of the work of the separated independent KBs.

4.6. Control actions

Consider the control actions generated by the considered types of control systems. In Figure 37 shows the control actions generated by the control system based on GA, ICS based on KBO on soft computing with one FC and ICS on soft computing with separated control. In Figure 37 *GA* is the signal generated by the control system based on the GA, *FC* is the signal generated by the ICS based on KBO on soft computing with one FC, *FC Decomposition* is the signal formed by the ICS on soft computing with separated control.

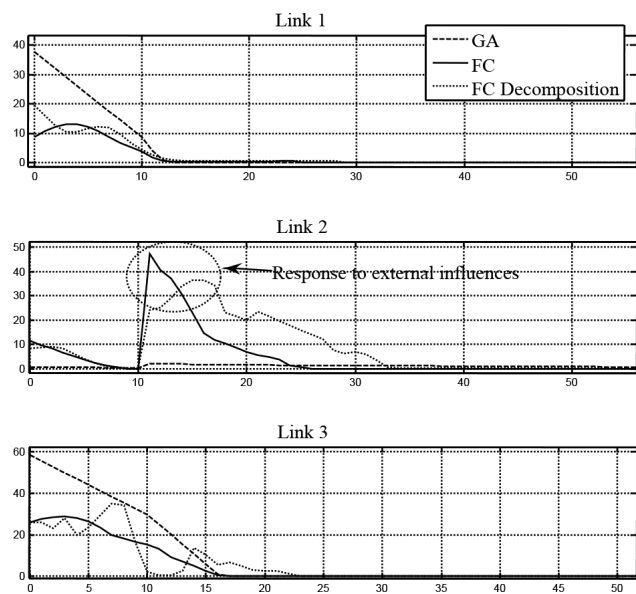


Figure 37. Control signals generated by the control system based on GA, ICS based on KBO on soft computing with one FC and ICS on soft computing with separated control

From Figure 37 it can be seen that the control signals generated by the control system based on the GA for the first and third links have a large amplitude compared to similar control signals generated by the ICS based on KBO on soft computing. For the second link in the control signal, formed by the control system based on GA, the reaction to external influence is not sufficiently reflected, as a result of which the task of precise positioning is not solved. The control signals generated by ICS based on KBO on soft computing with separate control, compared with ICS on KBO on soft computing with one FC, with a comparable amplitude, have greater overshoot.

Thus, the minimum consumption of useful resource in the formation of control signals is ensured when using the ICS based on KBO on soft computing with one FC.

5. Conclusion

To control robots with manipulators of varying complexity, the following were considered:

- 1) control systems with constant coefficients of the PID controller;
- 2) control systems with adjustable PID controller coefficients depending on the situation.

It has been shown that:

1. Control systems with constant coefficients based on GA are attractive because of the simplicity of implementation, however, due to the constancy of control parameters, the solution of the problem of accurate positioning is possible only for regular situations.

2. The unified KB of the ICS based on KBO on soft computing with one FC contains the most complete information about the behavior of all links, which allows the ICS to work both in standard and unexpected control situations. However, the creation of a single KB is a complex and long temporal process that requires significant computing resources. So, the implementation of a single KB for the complex CO 7DOF robot manipulator is not possible.

3. The decomposition of the control in the structure of the ICS based on KBS on soft computing with separate control, due to a slight decrease in the quality of control due to the mismatch of the behavior of the links as a result of the independence of the creation and functioning of the KBs, can significantly simplify the processes of creating and placing the KB.

4. Computational intelligence toolkit SCOptKB™ realized deep machine learning with optimal structure of FNN and reduce redundant information in production logical rules of robust KB.

In the next Part II, to eliminate the mismatch of the work of the separated independent KBs, the method of organizing coordination control using quantum computing technologies to create robust ICS 3DOF and 7DOF manipulators will be considered ^[33].

Appendix: Soft Computing Optimizer toolkit

ICS based on new types of computation (soft and quantum computing) have the following advantages:

- maintain basic advantages of conventional, classical, control systems such as controllability and stability;
- have optimal (from a given criteria of control quality) KB;
- guarantee the achievement of the given control quality on the base of designed KB;
- have the property of robustness. It means that ISC allows to maintain the given control quality in the case of

unexpected control situations.

A1. Peculiarities of the information technology for intelligent control system design based on Soft Computing Optimizer toolkit

For design of robust KBs of FC we developed the new program toolkit called *Soft Computing Optimizer* based on soft computing. SCO allows to design smart control systems with needed level of robustness.

Discuss the peculiarities of SCO and developed information technology.

We use *Genetic Algorithms* (GA) to find an optimal control signal and construct *teaching control signal* (TS). By using different GA fitness functions describing information-thermodynamic and control criteria and mathematical (or physical) model of CO we extract objective knowledge about control laws independent from human-expert. Processing of obtained TS is based on SCO with new types of computing. It allows us to design KB FC with a needed level of intelligence that supplies the needed level of robustness. Main components of SCO are the different GA structures with different constrains and fitness functions. Mutual actions of these components supply extraction, processing and design of KB, that is the main problem of Artificial Intelligence.

As summary list main factors of the information technology for ICS design: if we want to add to the known criteria stability and controllability a new one, we must use new types of computing.

New criterion of control quality *robustness* is introduced:

- Combined principle of control (*global negative back relation principle* + *global intelligent back relation principle*) allows us don't destroy the lowest control level (PID) and use the high level of control with the corresponding level of intelligence.

Introduction of *global intelligent back relation principle* allows realizing three steps of knowledge processing: extract information from dynamic behavior CO with PID control; use GA to construct *teaching control signal*; use a set of GA to design KB and optimize it.

By SCO we can design the given level of intelligence of control system and, hence, the given level of robustness.

A2. Main steps of the information technology for intelligent control system design based on Soft Computing Optimizer toolkit

Main steps of the information technology for ICS design based on SCO toolkit are shown on Figure A1.

Remark. Step 4 on Figure A1 is not considered in this Appendix. It is realized by SCO on quantum computing.

A2.1 Extraction, processing and design of objective knowledge based on stochastic simulation and soft computing

Describe main steps of developed KB FC design technology. At first consider briefly steps 1 - 3 (Figure A1), and then consider one example of KB FC design for the chosen dynamic CO.

The KB design process can be realized by the following steps.

Step 0. In this step one or a few typical *teaching situations* are defined. Here the following factors are described: parameters of the mathematical (or physical) model of CO; initial conditions; reference signal (a goal of control); external stochastic noise; presence/absence of time delay in the channel of CO state measurement and so on.

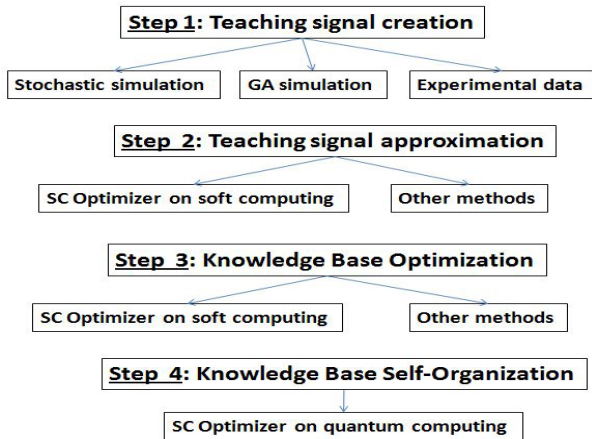


Figure A1. Main steps of the information technology for ICS design

Step1. Stochastic simulation system for teaching control signal design

For robust KBs design we will use *stochastic simulation system* in order to find robust teaching control signal.

Stochastic simulation is based on information extraction process by investigation of individual trajectories of dynamic object behavior under influence of stochastic noises acting on the object.

Stochastic noises simulation is considered as a random noises simulation with needed probability density function. Random noises simulation is realized by the method of forming filter on the base of Fokker-Planck-Kolmogorov equations^[17] (see Appendix 2 to this Chapter 1).

Stochastic simulation system uses CO model with simulated stochastic noises and GA with a chosen fitness function. By using GA, we obtain a set of optimal control values, which minimize the selected physical character-

istics of the stochastic model of CO. One of the characteristics can be control error, or the minimum entropy production rate of the control system and of the CO. In some complicated cases, the fitness function may include a weighted sum of different motion characteristics of the CO like accelerations, velocities, spectral characteristics. Thus, the resulted motion under control will tend to reduce all of them simultaneously. At this stage of simulation, we conduct simulation with the following aims:

- investigation free motion of CO in order to determine type of dynamic behavior, stable or locally/globally unstable motion,
- investigation an influence of different types of stochastic excitations on dynamic behavior and control laws,
- investigation an influence of type of traditional controllers (PID, PD, P) on type of control laws in a fuzzy control,
- investigation an influence of different GA fitness functions on type of control laws,
- comparison control quality of traditional PID control with *constant* gains and GA-PID control with *variable* gains obtained by GA,
- choice the best GA solution and designing a *teaching control signal* (TS) for the next steps of technology.

The general structure of stochastic simulation system is shown on Figure A2. On the figure the main factors that influent on the control accuracy are shown. They are: a presence of stochastic noises (as external and internal), a presence of time delay in the channel of CO state measurement, a presence of stochastic noises in the channel of CO state measurement. Moreover, we must consider also such factors as incompleteness of CO model, incorrectness of model parameters and so on.

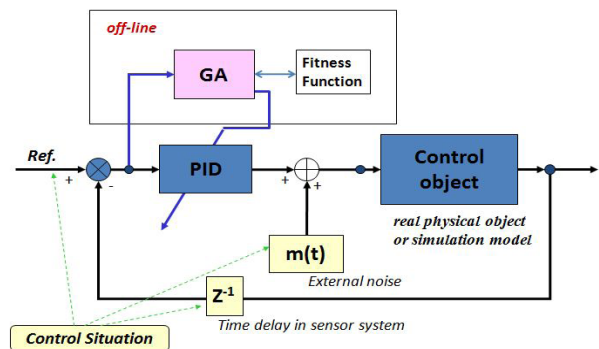


Figure A2. The general structure of stochastic simulation system

At the stage of GA based TS creation, we find a solution $\{K_p(t), K_d(t), K_i(t)\}$ close to a global optimum. The output of GA is TS (or training patterns) representing a table of ‘in-out’ patterns as fol-

low s: $\{E(t_i)\}, \{K(t_i)\}, i=1,...,n$, where $E(t_i)=\{e(t_i), \dot{e}(t_i), \int e(t_i)dt_i\}$ is vector, containing control error, its derivative and integral parts correspondingly, and $K(t_i)=\{K_P(t_i), K_D(t_i), K_I(t_i)\}$ are PID gains at time moments t_i .

SC Optimizer has tools to create TS using genetic optimization and Matlab model of control system (or physical model). This step is realized by the button “create signal”. On Figure A3 the main menu SCO and GA parameters window is shown.

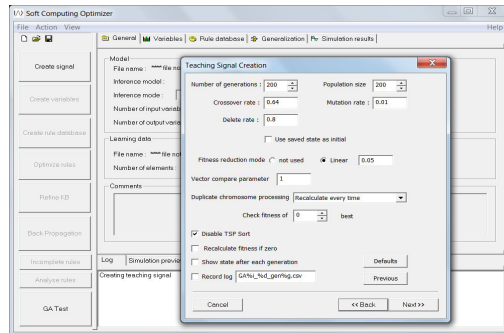


Figure A3. Main menu SCO and GA parameters window

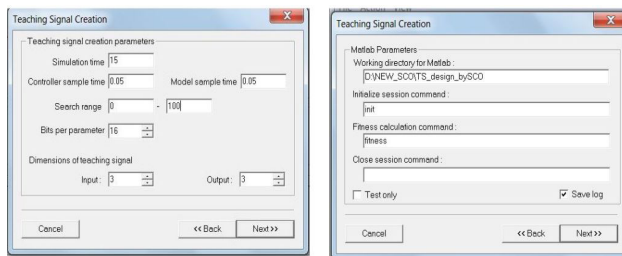


Figure A4. Windows of TS creation

By button “next” we go to the next windows shown on Figure A4. In the left window (Figure A4) signal creation parameters should be entered:

- Simulation time: time of simulation used to create TS.
- Controller sample time: sample time of control system.
- Model sample time: sample time of simulation.
- Search range: control signal search range.
- Bits per parameter: number of bits in GA chromosome per each control signal parameter. Greater values increase precision and optimization time.

Dimensions of TS: number of components in input and output parts of TS.

In the right window (Figure A4) the path to Matlab-model, initiation commands and fitness function are given. In the end of the given session, designed TS is saved in the «name».pat format as shown in Figure A5.

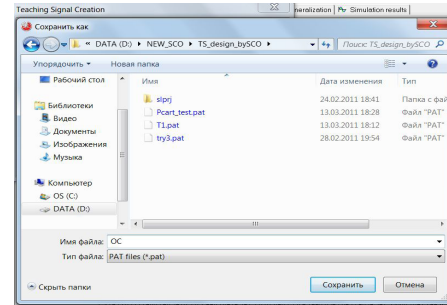


Figure A5. Window of saving TS

A3. Robust Knowledge Base Design based on SC Optimizer

Designed TS will be approximated by a fuzzy model chosen by a user.

Remark. TS also may be obtained experimentally from measurements of dynamic parameters of physical objects.

A3.1. Short general description

SCO uses the chain of GAs (GA_1, GA_2, GA_3) and approximates measured or simulated data (TS) about the modeled system with desired accuracy. GA_1 solves optimization problem connected with the optimal choice of number of MFs and their shapes. GA_2 searches optimal KB with given level of rules activation. Introduction of activation level of rules (LA) allows us to sort fuzzy rules in accordance with value information and design robust KB. GA_3 refines KB by using two criteria (see below). Figure A6 shows the flow chart of SCO operations on macro level and combines several stages.

Stage 1: *Fuzzy Inference System (FIS) Selection*. The user makes the selection of fuzzy inference model with the featuring of the initial parameters.

Stage 2: *Create linguistic values*. GA optimizes linguistic variable parameters, using the initial parameters, and TS, obtained from the in-out patterns, or from dynamic response of CO (real or simulated in Matlab).

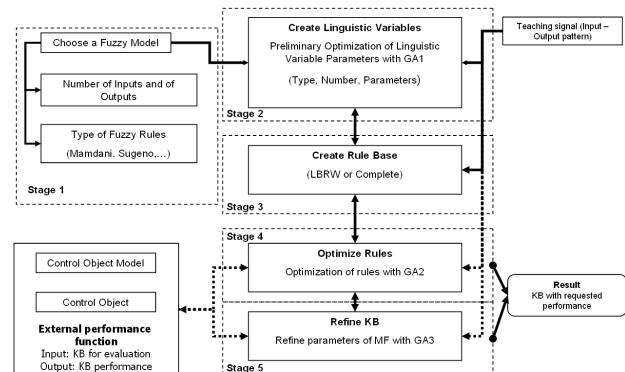


Figure A6. Flow chart of SC Optimizer

Stage 3: *Rule base creation*. At first, we use the rule rating algorithm (LBRW) for selection of the certain number of rules. The “Level of activation” (*LA*) criteria is a parameter given by a user. At this stage the *total firing strength of each rule* ($R_{total}^i = \sum_t R^i(t)$, where *t* is a time, *i* is a rule index) is calculated. Then the “Sum of firing strength” and “Max of firing strength” criterias are used for design KB^[17]. Output of this stage is the rule base designed according to the chosen criteria and activation level.

Stage 4: *Rule base optimization*. GA_2 optimizes the rule base (Stage 3), using the fuzzy model (Stage 1), optimal linguistic variable parameters (Stage 2), and TS. If you are still not satisfied with model quality you can use *Error Back Propagation* algorithm.

Stage 5: *Refine KB*. On this stage, the structure of KB is already specified and close to global optimum. In order to reach the optimal structure, two criteria can be used. First criterion is based on the minimum error, and in this case KB refining is similar to classical derivative based optimization procedures (like error back propagation algorithm for FNN tuning). Second criterion is based on the maximum of mutual information entropy^[17]. The result of the Stage 4 is a specification of fuzzy inference structure, optimal for solution of a current problem. In order to have robust solution, Stage IV can be bypassed, and the robust structure obtained with GAs of stages 2 - 3 can be used.

A4. Description of steps in SC Optimizer toolkit

Designed TS is used on the next step of technology (step 2 on the Figure A1). At first, we must create a new sco-project.

A4.1 New Project creation SCO allows to create a new model or load previously created model from file.

If you choose to create a new model the system will prompt you about model parameters, including inference model, number of input and output variables, number of fuzzy sets for each variable and so on. New model creation window is called by buttons «File», «New» in main menu. The window is shown on Figure A7 (a). Then following the button “next” we go to the window for TS input shown on Figure A7 (b).

(a)

(b)

Figure A7. New model creation windows

After TS is inputted, it must be adopted for SCO data processing format. For that purpose there is the window (Figure A7 (b)) where you must push the button «Change».

Created model is saved into file «name.sco», for example, «Pcart_TS1.sco». After the model was created or loaded you will be presented with main program menu, allowing you to view model parameters, start different optimization algorithms or edit model manually.

After new model is created go to the next step create variables.

A4.2. Membership functions creation and its optimization

First step is GA_1 which solves optimization problem connected with the optimal choice of number of MFs and their shapes. This process is called by button «Create variables» and then you go forward according to menu.

When working with GA_1 algorithm you can run signal filtering algorithm which will remove redundant signal

lines. This can improve quality of fuzzy sets created by GA_1 algorithm. If you wish to use this mode select Filter Signal checkbox on the first page of the dialog and enter desired filter threshold level (see Figure A8).

Next window will be the window with GA parameters. Fill it and press NEXT>> to switch to the next page. Select variables, which should be optimized, by holding CONTROL key and clicking items in the list. If you are running this algorithm for the first time it is recommended to leave all variables selected. Use this feature in order to improve quality of some variables later.

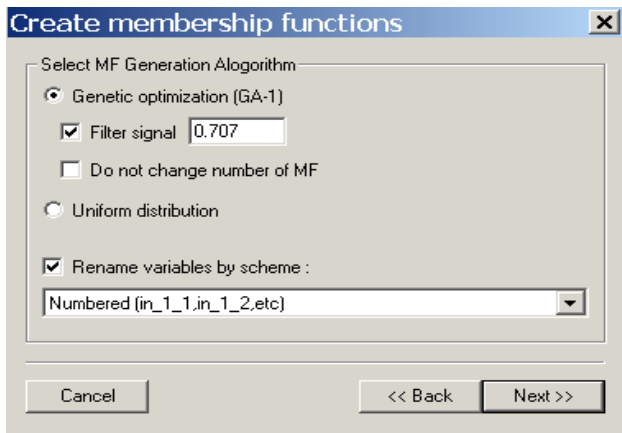


Figure A8. Create MFs window

SCO supplies two ways of MFs determining: creating variables *with uniform distribution algorithm* and creating variables *with GA_1* that finds best (from the fitness function view) combination of fuzzy sets for each input variable. Also, GA_1 finds optimal form (type) of MFs and optimal value of intersection between neighbor fuzzy sets

On Figure A9 one example of designed MFs is shown. As shown in this figure, for «Input_3» values description GA_1 finds seven fuzzy sets (membership functions).

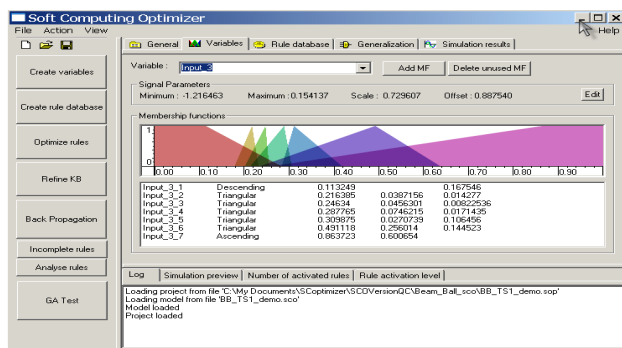


Figure A9. Example of designed MFs

A5. Rule database creation

After you have created variables and MFs you can create rule database. You can do this by pressing Create

rule database command button or with Action/Create rule database menu. After pressing «Create rule database» the following window is shown (Figure A10).

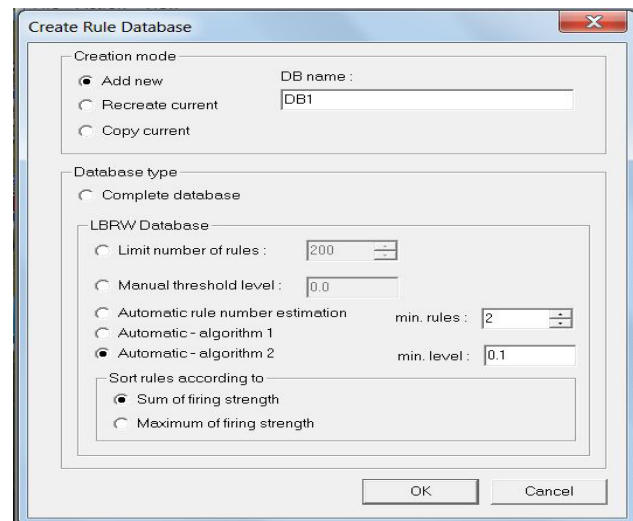


Figure A10. Create rules window

SCO support two types of rules database (RD): complete database and LBRW database (LBRW from “Let the Best Rule Win”). Complete database consists of all possible combinations of fuzzy sets describing input variables. The number of rules in complete RD equals the product of numbers of fuzzy sets for each input variables. If in the model there are more than three input variables then the complete RD has a large number of rules. Usually such kind of RD contains redundant information, and control with this RD is not effective.

LBRW algorithm chooses only valuable (robust) rules. Decreasing number of rules gives greater velocity of RD optimization without loss of accuracy. When creating LBRW database you can specify exact number of rules or minimal level of firing strength (threshold level). In the latter case created database will include all rules with firing strength greater than or equal to one you specify.

On Figure A11 an example of designed rules database is shown. As you can see, complete database contains 486 rules, but designed LBRW database consists only of 26 rules.

Next window will be the window with GA parameters. Fill it and press NEXT>> to switch to the next page. Select variables, which should be optimized, by holding CONTROL key and clicking items in the list. If you are running this algorithm for the first time it is recommended to leave all variables selected. Use this feature in order to improve quality of some variables later.

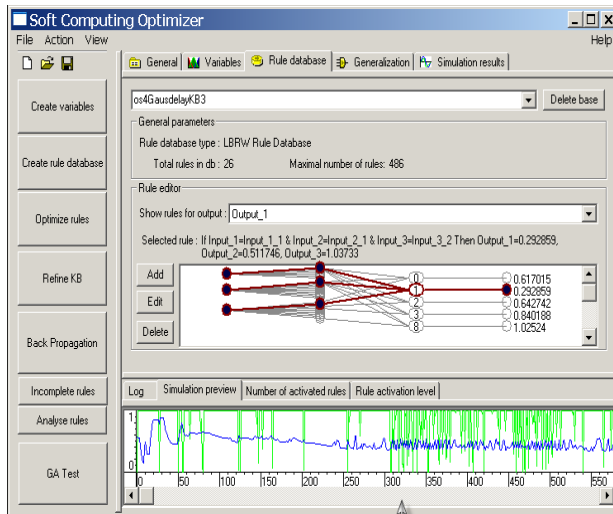


Figure A11. Example of designed rules database

On Figure A11 in the line named «Selected rule» a chosen rule (red bolt line on the FNN structure; order number of the chosen rule = 1) is shown in symbolic form:

« If Input_1 = Input_1_1 & Input_2 = Input_2_1 & Input_3 = Input_3_2 Then Output_1 = 0.292859, Output_2 = 0.511746, Output_3 = 1.03733 »

In the low part of the window (Figure A11) the result of TS approximation is shown. Green line represents a TS, blue line represents approximation of TS by chosen fuzzy system with designed rule database with 26 rules.

A6. Rule database optimization

After rule database is created, proceed to their optimization by GA₂. Press «Optimize rules» and the window shown on Figure A12 is opened.

There are three possibilities:

- RD optimization with complete TS,
- RD optimization with optimized TS,
- RD optimization by Matlab simulation.

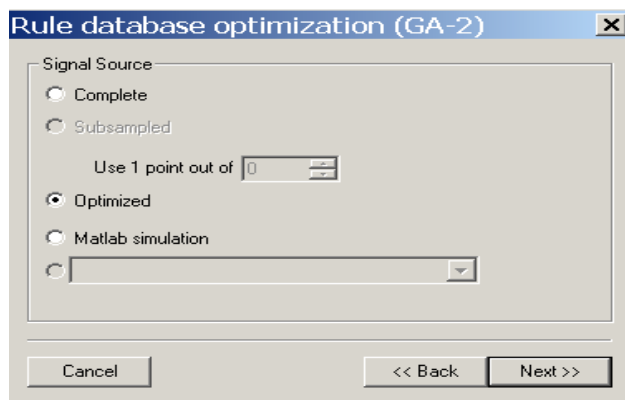


Figure A12. Rule database optimization window

Choose one way, press NEXT>> and the following

windows are opened (see Figure A13).

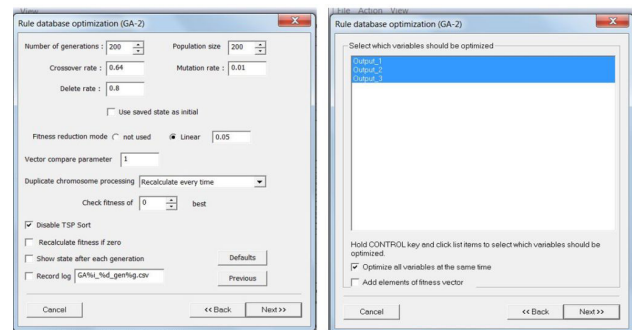


Figure A13. Choice of GA parameters and selecting variables

You should select output variables for which database should be optimized. By default, optimization is selected for all variables and you shouldn't change it when starting algorithm for the first time.

During optimization a progress window will appear (see Figure A14). It displays variables currently optimized, number of current generation and achieved level of evaluation function.

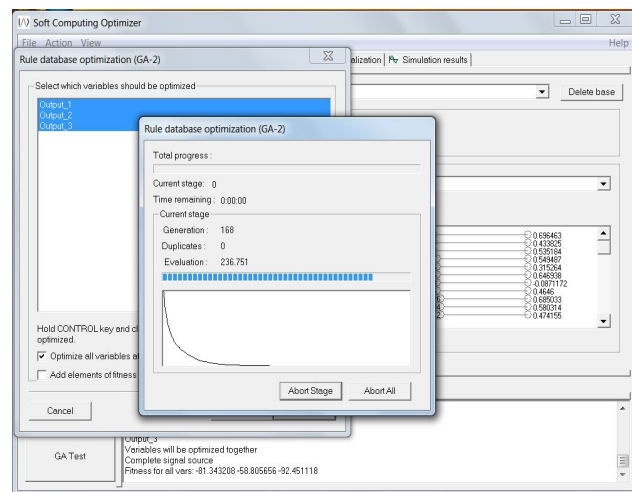


Figure A14. Progress window of optimization process

You can press Abort Stage button if you want to stop optimization for the current stage. The state of the variables will be set to the best state found before abort button was pressed and the optimization will switch to the next variable. Press Abort All to stop optimization process and return to SCO.

So, as result of GA₂ optimization we obtain optimal values of right parts of fuzzy rules.

Remark. GA₂ optimization is based on TS. If TS is not optimal (from the control quality criterion) GA₂ optimization may be not optimal too. For that case in SCO toolkit there is an effective way - RD optimization by Matlab simulation.

A7. Rule database optimization with Matlab simulation

For RD optimization by Matlab simulation choose option «*Matlab simulation*» in window on Figure A12 (see above) and press NEXT>>. After two windows as shown on Figure A13 (see above) is fulfilled, by pressing again NEXT>> we get into the window shown on Figure A15.

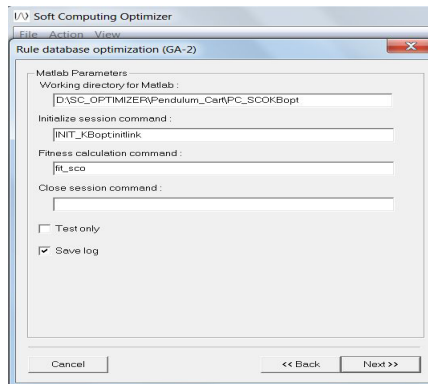


Figure A15. Window for connection to Matlab/Simulink model

In this window the path to Matlab model, initiation commands and fitness function are given.

A8. Fine tuning of the model

When rule database is optimized you can further improve model quality by returning to MFs optimization. This is accomplished by the last optimization step model refinement (known as GA₃ algorithm). You can start model refinement by clicking Refine KB command button or selecting Action/Refine KB menu item.

After you activate the command wizard dialog will appear. It will first prompt you which fitness function you would like to use. Three choices are available:

- Maximization of mutual information entropy: Tells SCO to minimize mutual information entropy between MF fuzzy sets. This is the same function used in GA₁ algorithm, but unlike GA₁, GA₃ won't change number of MF's per variable, only MF parameters will be changed.
- Minimization of output error: Minimize output error.
- Matlab simulation: Use Matlab/Simulink to calculate fitness function.

Select one of the variants and press NEXT>>. Enter genetic algorithm parameters on the second page and press NEXT>> to switch to the next page.

Now you should select input variables, which should be optimized. By default, optimization is selected for all variables. You can change selection by holding CTRL and clicking left mouse button on the list items. You also have an option to optimize all variables at the same time (if you check

“optimize all the variables at the same time” check box). If you leave this checkbox unchecked program will optimize variables one after another. If you check Add elements of the fitness vector box then elements of the resulting fitness vector will be added together. Otherwise vector fitness function will be used. Press NEXT>> to start optimization.

While GA₃ algorithm operates the progress dialog will be shown. It will display number of current generation and achieved level of evaluation function. You can press Abort Stage button if you want to stop optimization for the current stage. The state of the variables will be set to the best state found before abort button was pressed and the optimization will switch to the next variable. Press Abort All to stop optimization process and return to SCO.

If you are still not satisfied with model quality you can run rule database optimization (GA₂) again or use Error Back Propagation algorithm. Error Back Propagation algorithm implements classical gradient optimization method, which provides an effective way to further improve model output after genetic optimization. You can start Back Propagation algorithm by clicking Back Propagation command button or selecting Action/Back Propagation menu item.

References

- [1] Ulyanov, S. V. Self-organized Control System [P] US patent, No. 6, 411, 944, 1997.
- [2] Aliev, R. A., Ulyanov, S. V. Fuzzy Models of Control Processes and Systems [J] Itog. Nauk. Tekh. (VINITI), Ser. Eng. Cybern., 29 (1990); 32 (1991).
- [3] Haddad, W. M., Chellaboina, V., Nersesov, S. G. Energy-and entropy-based stabilization for lossless dynamical systems via hybrid controllers [J] IEEE Transactions on Automatic Control, Vol. 52, No 9, 2007: 1604-1614 pp.
- [4] Haddad, W. M., Chellaboina, V., Nersesov, S. G. Thermodynamics: A Dynamical Systems Approach, Princeton Series in Applied Mathematics Princeton [M] NJ: Princeton University Press, 2005.
- [5] Haddad, W. M., Chellaboina, V., Nersesov, S. G. Impulsive and Hybrid Dynamical Systems: Stability, Dissipativity, and Control, Princeton Series in Applied Mathematics [M] Princeton, NJ: Princeton University Press, 2006.
- [6] Ulyanov, S. V., Yamafuji, K., Kurawaki, I. et al. Computational Intelligence for Robust Control Algorithms of Complex Dynamic Systems with Minimum Entropy Production [J] Advanced Comput. Intelligence and Intelligent Informatic, Vol. 3, No 2, 1999.
- [7] Bologna, M., Grigoloni, P., Karagiorgios, M. et al. Trajectory Versus Probability Density Entropy [J] Phys.

- Rev. 64E (1), 2001.
- [8] Petrov, B. N., Gol'denblat, I. I., Ulanov, G. M., Ul'yanov, S. V. The Theory of Models in Control Processes: Information Thermodynamic Aspects [M] Nauka, Moscow, 1978 [in Russian].
- [9] Litvintseva, L. V., Takahashi, K., Ulyanov, S.V., et al. Intelligent Robust Control Design Based on New Types of Computations [J] Note del Polo Ricerca. Milano: Universitadegli Studidi Milano Publ., Vol. 60, 2004.
- [10] Petrov, B. N., Ulanov, G. M., Ul'yanov, S. V. et al. Control Problems for Relativistic and Quantum Dynamic Systems [M], Nauka, Moscow, 1982 [in Russian].
- [11] Petrov, B N., Pugachev, V. S., Ulyanov, S. V. et al. Informational Foundations of Qualitative Theory of Control Systems [J] in Proceedings of 7th IFAC Congress, Helsinki, Finland, Vol. 3, 1978.
- [12] Granichin, O. N., Polyak, B. T. Randomized Estimation and Optimization Algorithms under Almost Arbitrary Noise [M] Nauka, Moscow, 2003 [in Russian].
- [13] Polyak, B. T., Shcherbakov, P. S. Robust Stability and Control [M] Nauka, Moscow, 2002 [in Russian].
- [14] Perroud, M., Saucier, A. Thermodynamics of Dissipative Systems [J] Helvetica Physica 60 (8), 1987.
- [15] Ulyanov, S. V., Hagiwara, T. Intelligent Mechatronic Control Suspension System Based on Soft Computing [P] US patent, No. 6,701,236, 2001.
- [16] Panfilov, S. A., Litvintseva, L. V., Ul'yanov, S. S. et al. Software Support of the Processes of Generation, Eliciting, and Design of Knowledge Bases of Robust Intelligent Control Systems [J] Program. Prod. Syst., No. 2, 2004.
- [17] Fujii, S., Watanabe, H., Panfilov, S. A. et al. Intelligent Robust Control System for Motorcycle Using Soft Computing Optimizer [P] US patent, No. 20050197994, 2005.
- [18] Ulyanov, S. V. et al. System and method for stochastic simulation of nonlinear dynamic systems with a high degree of freedom for soft computing applications [P] Patent US 2004/0039555 A1. Pub. Date: Feb. 26, 2004.
- [19] Kurawaki, I, Litvintseva, L. V., Takakhashi, K. et al. Design of Robust Knowledge Bases of Fuzzy Controllers for Intelligent Control of Substantially Nonlinear Dynamic Systems, I. Application of Soft Computing [J] Izv. Ross. Akad. Nauk, Teor. Sist. Upr., No. 4, 2004 [Comp. Syst. Sci. Intern., Vol. 43, No 3, 2004: 615-632 pp.].
- [20] Khoogar, A. R. et al. Obstacle Avoidance for Redundant Manipulators Using a Genetic Algorithm [J]. Southeastcon'91 Conference, Williamsburg, VA, 7-10 Apr., Vol. 1, 1991: 317-320.
- [21] Secara, C., Vladareanu, L. Iterative genetic algorithm-based strategy for obstacles avoidance of a redundant manipulator [J] Wseas Transaction on Mathematics, Vol. 9, No 3, 2010: 211-221.
- [22] Yu, W., Rosen, J. Neural PID Control of Robot Manipulators with Application to an Upper Limb Exoskeleton [J] Cybernetics, IEEE Transactions, Vol. 43, No 2, 2013: 673-684.
- [23] Jasour, A. M, Farrokhi, M. Path Tracking and Obstacle Avoidance for Redundant Robotic Arms Using Fuzzy NMPC [J] American Control Conference, Hyatt Regency Riverfront, St. Louis, MO, USA, 10-12 June, 2009: 1353-1358.
- [24] Meza, J. L. et al. Fuzzy Self-Tuning PID Semiglobal Regulator for Robot Manipulators [J] Industrial Electronics, IEEE Transactions, Vol. 59, No 6, 2012: 2709-2717.
- [25] Salas, F. G., Santibanez, V., Llama, M. A. Variable Gains PD Tracking Control of Robot Manipulators: Stability Analysis and Simulations [J] World Automation Congress (WAC), Puerto Vallarta, Mexico, 24-28 June, 2012: 1-6.
- [26] Ulyanov, S. V. et al. Intelligent robust control: soft computing technologies [M] M.: VNIIGeosystem, 2011: 408 p [in Russian].
- [27] Panda, R. C. Introduction to PID Controllers - Theory, Tuning and Application to Frontier Areas [J] Rijeka, Croatia: InTech, 2012: 258 p.
- [28] Mikhailov, V. S. Control Theory [M] K: Higher School, 1988: 312 p [in Russian].
- [29] Ulyanov, S. V., Litvintseva, L. V., Mishin, A.A., Sorokin, S. V. Certificate of state registration of computer programs No. 2011619257. Optimizer of robust knowledge bases for the design of intelligent control systems on soft computing: [P] Application No. 2011617532 dated 11.10.2011 RF Registered in the Register of computer programs on December 1, 2011 [in Russian].
- [30] Phillips, C., Harbor, R. Control systems with feedback [M] M.: Laboratory of Basic Knowledge, 2001: 616 p.
- [31] Ulyanov, S. V. Soft Computing Optimizer of Intelligent Control System Structures [P]: US patent, No. 20,050,119,986, 2005.
- [32] Nikolaeva, A. V., Ulyanov, S. V. Design of an intelligent control system for an excess robot with a manipulator with seven degrees of freedom. Part 1: Soft Computing Technologies [J] System analysis in science and education: a network scientific publication, No 4, 2013. Access mode: <http://www.sanse.ru/download/193> [in Russian].
- [33] Ulyanov, S. V. Self-organized robust intelligent control [M] Saarbrücken: LAP Lambert Academic Publishing, 2015: 412 p.