Deep Machine Learning SW for Intelligent Control. Part I: Soft computing KB optimizer supremacy

Alena V. Nikolaeva[1], Sergey V. Ulyanov[2], Liudmila V. Litvitseva[3], Victor S. Ulyanov[4]

[1]PhD student; Dubna State University, Russia
INESYS LLC (EFKO GROUP), Russia

[2] Doctor of Physical and Mathematical Sciences; Dubna State University, Russia;
INESYS LLC (EFKO GROUP), Russia

[3] PhD; Professor of Milan University and University of Electro-Communications, Tokyo, Chofu.

[4] PhD, Associate Professor, MISIS R&D University, Russia;
INESYS LLC (EFKO GROUP), Russia

ABSTRACT

The technology and toolkit for development of intelligent control for complex unstable dynamic systems are described. A new approach is founded on the new ideas of soft computing applied to intelligent control system design based on fuzzy PID controllers techniques (further, call it shortly as FC). For design of robust Knowledge Bases of FC the new program toolkit called Soft Computing Optimizer (SCO) is developed. Computational intelligence toolkit SCO is a deep machine learning SW platform with optimal fuzzy neural network structure. It allows designers to realize the principle of optimal intelligent control with a maximum reliability and controllability level in the presence of a complex control object under conditions of uncertainty in a source data, and in the presence of stochastic noises of various physical and statistical characters. The SCO structure, its application for the development of a robust intelligent control system solving a problem of precision positioning of manipulator (with three degrees of freedom) is described.

KEYWORDS

Intelligent control system; knowledge base; soft computing technology; robotic manipulator.

INTRODUCTION

Dynamic systems not easily controlled by traditional control systems (such as P- [I]-D-controllers) in the case of complex, essentially non-linear and ill-defined structures of controlled objects, and especially in a presence of different stochastic noises.*Intelligent Control Systems* (ICS) design methodology provides a main alternative way to the traditional control system's design [1]. Dynamic systems ICS design is usually based on *Fuzzy Controllers* (FC) and *Fuzzy PID Controllers* with *Soft Computing* (SC) application [2-4].

Soft computing methodologies, such as *genetic algorithms* (GA) and *fuzzy neural networks* (FNN) had expanded application areas of FC. But a lot of researchers have demonstrated that fuzzy controllers prepared to maintain control object in the prescribed conditions are often fail to control when such a conditions are dramatically changed (see, for example, [5]).

Let will keep in mind the following peculiarities of *Fuzzy PID Controllers* design with traditional SC application:

- in classical PID-control the PID parameters are constant. In fuzzy PID control they are considered as variable;
- "input-output" linguistic variables of FC must be described and
- teaching "input-output" linguistic relations of FC must be determined;
- laws of coefficient gain's schedule of the time dependent PID-parameters are described in a form of a Knowledge Base (KB) of a Fuzzy Controller.
- an optimization of FC KB is performed by using GA, FNN.

The *learning* and *adaptation* aspects of FC's have always the interesting topic in advanced control theory.

Many learning schemes based on the *back-propagation* (BP) algorithm [2-4]. But BP algorithm is successfully working if we perform control task without a presence of ill-defined stochastic noises in environment or without a presence of unknown noises in sensors systems and control loop, and so on.

For more complicated control situations learning and adaptation methods based on BP-algorithms do not guarantee the required robustness and accuracy of control in imperfect information and hazard situations.

We have conducted series of benchmark simulations which have shown the following. In a case of a global unstable essentially non-linear dynamic control object and in a presence of different stochastic excitations on control object (or random noises in sensor's measurement system in control channel loops), traditional SC approach cannot guarantee a robust and stable control achievement. We also have shown that usage only one principle of control (for example, a minimum of control error as a fitness function in GA) does not always guarantee that we obtain an optimal control. If the control object model is essentially non-linear and excitation on the object is not Gaussian, we need to consider also the physical criteria of minimum of entropy production rate [1].

Experimental results have shown that new SC based approaches are needed to solve a main problem in modern ICS design: how to construct a *robust* Knowledge Base for increasing *self-learning, self-adaptation and self-organizing capabilities* of developed control system.

For all mentioned kind of cases, we develop a new technology of smart control design based on SC and principle of minimum entropy production rate (MEP)**[**1,6,7**].**Based on the MEP principle, we developed SC tools that allow us to form the knowledge base of FC by extracting information from the stochastic simulation of control object behaviour and by using new approach to KB FC optimization with education and industrial applications as intelligent robotics and mechatronics.

1. THE IT STRUCTURE OF INTELLIGENT CONTROL SYSTEMS DESIGN

The general hierarchical structure and stages of execution of information technology embedded in the process of design of integrated fuzzy PID controllers for autonomous and interconnected COs with different physical nature shown in Figure1.

This technology uses computational intelligence toolkit for design of *Knowledge Bases* (KB) in the FC of the lower executive level [1,5-11]. The main role in the structure of this technology 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 shown in Figure1.

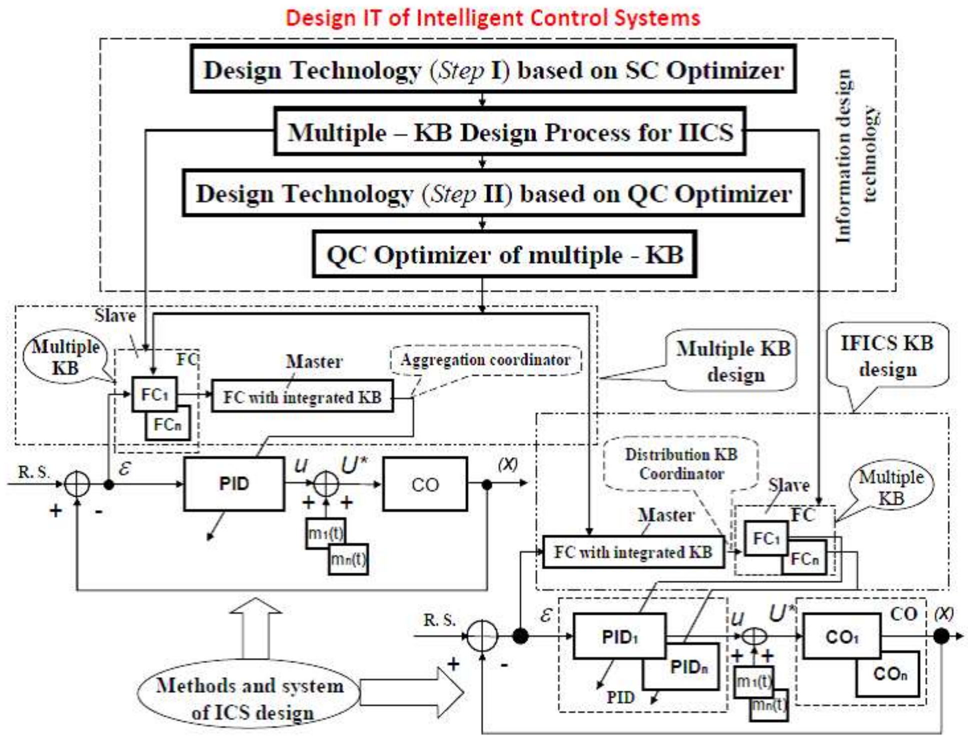


Figure 1. General hierarchical structure of information design technology of robust KBs for integrated fuzzy PID controllers

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 / hazardcontrol situations (see Part III).

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 Figure1) individual KBs for two (or more) FCs for particular control situations (learning situations) formed. Optimizer of KBs used with the technology of soft computing and fuzzy stochastic simulation.

The optimizer of KB SCOptKB™ was developed as a new toolkit of computational intelligence based on the technology of soft computing (first design stage), including the GAs set and fuzzy neural networks (FNNs) for realization of optimization and learning procedures (universal robust approximator) of production rules in KBs, respectively.

The toolkit used for extraction of objective knowledge from the dynamic behavior of weakly (ill-defined) structured models of complex COs and design of robust KBs in FC with deep knowledge representation (see Figure2).

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), 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).

The optimization of control processes with required quality and robustness levels 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 [1,5-12].

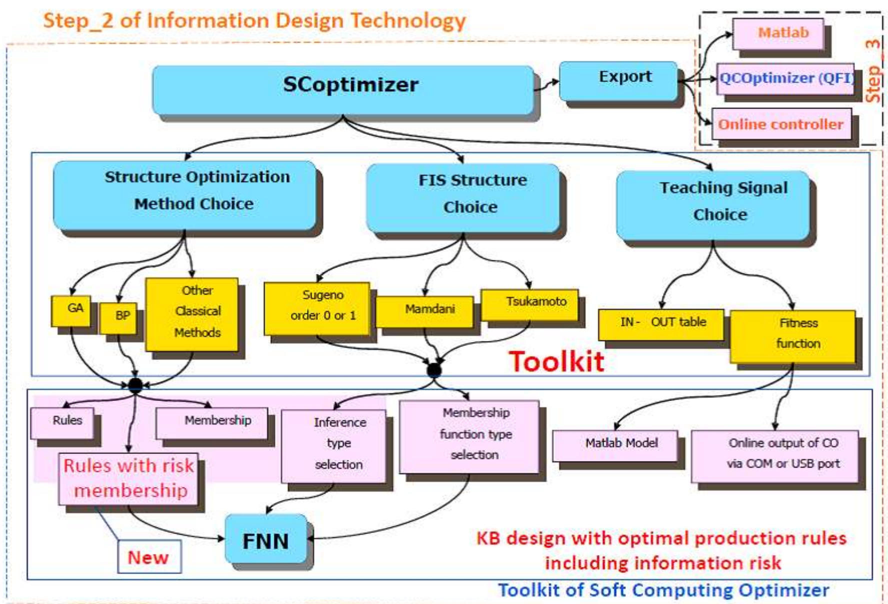


Figure 2. Structure of SCO toolkit of information design technology of robust KBs for integrated fuzzy controllers

Based on new types of computation (soft and quantum computing) SCO 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 unpredicted control situations.

2. THE STRUCTURE AND MAIN STEPS OF THE SCO BASED OPTIMIZATION OF KB

The SCOptKB™ 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).
Remark. For simplicity instead of SCOptKB™we will use abbreviation SCO.
The SCO consists of interrelated GA_1, GA_2, GA_3, which optimize particular components of KB.
The input of the SCO is a teaching signal (TS), which can be obtained either at the stage of stochastic simulation of the behavior of the controlled object (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 object.As new optimization criteria, we take the thermodynamic and information-entropy criteria represented in Table1 (see below).
The structure of the SCO for the design robust ICSs presented on Figures3 and 4.

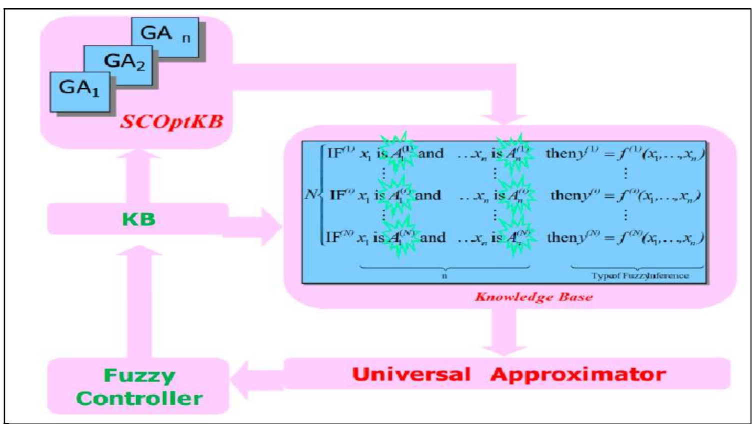


Figure 3. The structure of knowledge base optimization bySCOptKB™

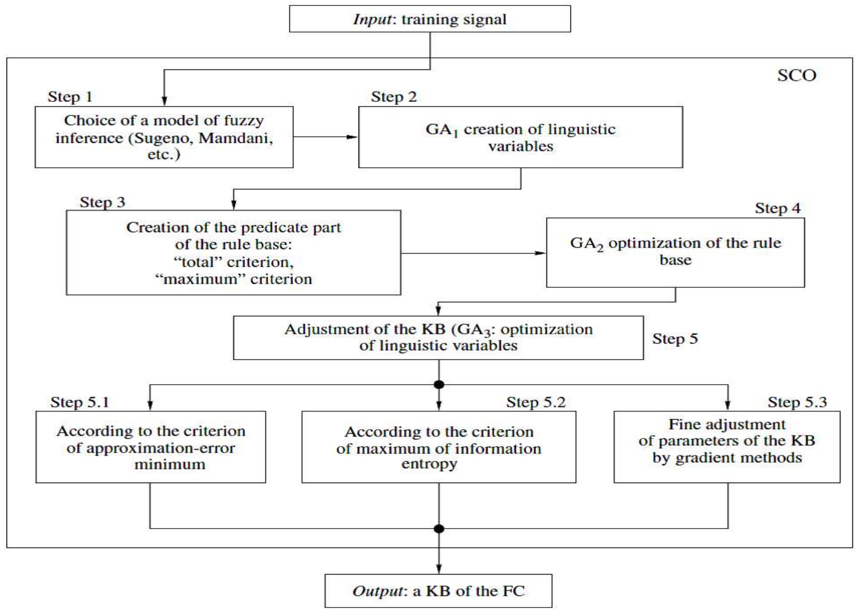


Figure 4. **The general block scheme of the KBoptimization algorithm.**

Figure4 presents the successive implementation of the steps of SCO based KB optimization algorithm.

Let us specify the steps shown on Figure 4.

Step 1.*A choice of the model of a 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.*A creation of linguistic variables.* With the application of GA_1, 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.*A design of the rule base.* At this stage, a special algorithm for selection of the most robust rules used in accordance with the following two criteria:

1) "total" criterion: choose only the rules that satisfy the following condition: $R^l_{total\_fs} \geq TL$,where TL (threshold level) is a given (manually or chosen automatically) level of rule activation, and

$$R^l_{total\_fs} = \sum_{k=1}^{N} R^l_{fs}(t_k),$$

and $R^l_{fs}(t_k) = \prod [\mu^l_{j1}((x_1(t_k)), \mu^l_{j2}((x_2(t_k)), ..., \mu^l_{jn}((x_n(t_k))]$,

where t_k are time instants, $k = 1,...,N$, and N is equal to the number of points in the control signal; $\mu^l_{jk}(x_k)$, $k = 1,...,n$ 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 $\max_t R^l_{fs}(t) \geq TL$.

Step 4. *The optimization of base rules*. With the help of GA_2, the right sides of rules of the KB defined at Step 3 optimized. At this stage, a solution that is close to the global optimum found (minimum error of approximation of the training signal). With the application of the next step, this solution can improved locally.

Step 5.*The adjustment of the base of rules.* With the help of GA_3, 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 Figure4) 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 Figure4).

The verification (testing) of designed knowledge base. Constructed at stages 4, 5.1, 5.2, and 5.3 (on Figure4) KBs of the ICS are tested from the viewpoint of robustness and control quality. For further use, the best KB is investigated in online regime for different control situations.

Examples of KBs simulation based on efficient application of the SCO are considered for control system design of robotic manipulators in section 4.1.

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	$H^j_{X_i} = -p^j_{X_i}\log(p^j_{X_i}) = -p(x_i \mid x_i = \mu^j_{X_i})\log[p(x_i \mid x_i = \mu^j_{X_i})]$ $= -\frac{1}{N}\sum_{t=1}^{N}\mu^j_{X_i}(x_i(t))\log[\mu^j_{X_i}(x_i(t))]$	Elimination of redundancy of the TS Choice of an optimal cardinality of term-sets of linguistic variables of the components of the TS
		$H^{(j,l)}_{X_i \mid X_k} = H\left(x_j\big\rvert_{x_i = \mu^j_{X_i},\, x_k = \mu^l_{X_k}}\right) = -\frac{1}{N}\sum_{t=1}^{N}[\mu^j_{X_i}(x_i(t)) * \mu^l_{X_k}(x_k(t))]$, $\log[\mu^j_{X_i}(x_i(t) * \mu^l_{X_k}(x_k(t)))]$, where * is the chosen operation of fuzzy AND	
GA_2: Optimization of rule base	Minimum approximation error	$E = \sum_p E^p$, where $E^p = 1/2(F(x_1^p, x_2^p, \ldots, x_n^p) - d^p)^2$	Choice of optimal parameters of the right sides of rules
GA_3: Adjustment of the KB	Minimum approximation error or maximum joint information entropy	$E = \sum_p E^p$	Fine adjustment of the parameters of membership functions
		$H^j_{X_i}$	

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, control criteria, and mathematical (or physical) model of CO we extract objective knowledge about control laws independent from human-expert. Processing of obtained TS 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* introduced:

- Combined principle of control (*global negative back relation principle* + *global intelligent back relation principle*) allows us do not 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.

2.1. Extraction, data processing and design of objective knowledge based on soft computing and stochastic simulation

The KB design process uses a teaching control signal to design KB of FC and optimize it. Let us discuss: how to design a teaching control signal for the given control task?

For this aim we use a *stochastic simulation system*. The 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 controlled object (CO). Stochastic noises simulation considered as a random noises simulation with needed probability density function. Random noises simulation realized by the method of forming filter on the base of Fokker-Planck-Kolmogorov equations [12].

The general structure of a stochastic simulation system shown on Figure6.

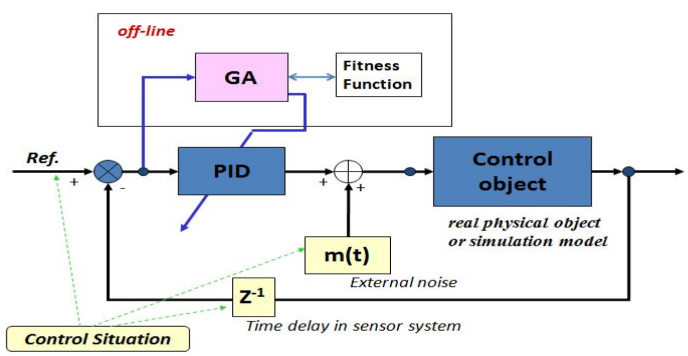


Figure 6. The general structure of stochastic simulation system

At first the following factors must be 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.

Then the stochastic simulation system uses CO model with the simulated stochastic noises and GA with a chosen fitness function.One of the characteristics can be control error, or the minimum of the 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.

By using GA, we obtain a set of optimal control values, which minimize the selected physical characteristics of the stochastic model of CO.

On the Figure 6 the main factors that influent on the control accuracy are shown. These factors are the following: 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.

At this stage of simulation, we conduct simulation with the following aims:

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

At the stage of GA based TS creation, we find a solution $\left\{K_p(t), K_d(t), K_i(t)\right\}$ close to a global optimum. The output of GA is TS (or training patterns) representing a table of 'in-out' patterns as follows: $\left\{E\left(t_i\right)\right\},\ \left\{K\left(t_i\right)\right\},\ i=1,\ldots,n,$ where $E\left(t_i\right)=\left\{e\left(t_i\right),\ \dot{e}\left(t_i\right),\ \int e\left(t_i\right)dt_i\right\}$ is vector, containing control error, its derivative and integral parts correspondingly, and $K(t_i)=\left\{K_P\left(t_i\right),\ K_D\left(t_i\right),\ K_I\left(t_i\right)\right\}$ 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 realized by the button "create signal".

Let us go to SCO main menu description.

Finally, let us summarize the main ideas of SCO. Figure 7 shows the flow chart of SCO operations on macro level and combines several stages.

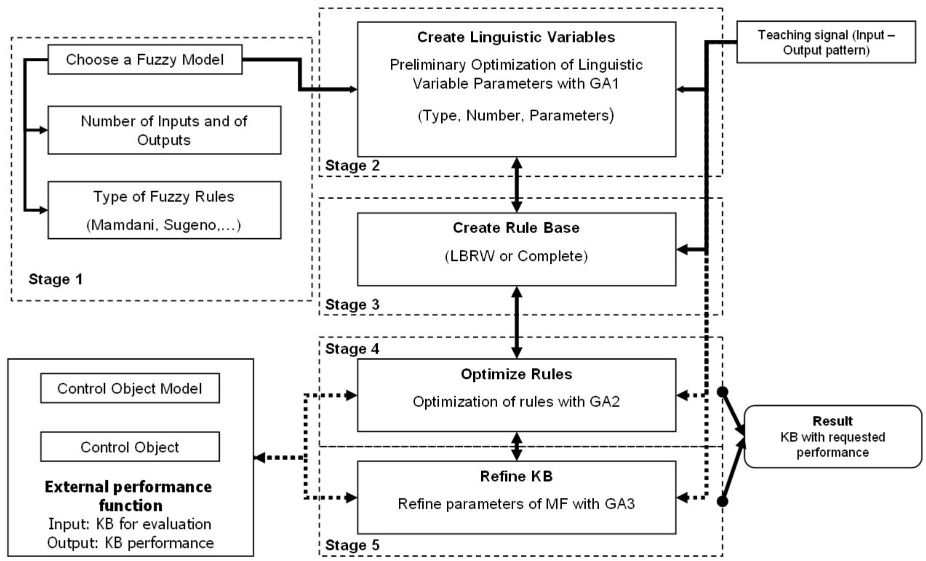


Figure 7. The flow chart of SC Optimizer

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 corresponding criteria.

Let us consider main functions in SCO toolkit.

3.Brief descriptionof SC Optimizer toolkit

At first, we must create a new sco-project.

3.1. New Project creation

SCO tools allows us 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 called by buttons «*File*», «*New*» in main menu.

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

Created model saved into file «name.sco». After the model created or loaded, you will 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.

3.2. Membership functions creation and its optimization

First step is the application of GA_1which solves an optimization problem connected with the optimal choice of number of MFs and their shapes. This process 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.

SCO supplies two ways of MFs determining: creating variables *with uniform distribution algorithm*and creating variables with GA_1 that finds a 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 Figure8 one example of designed MFs is shown.

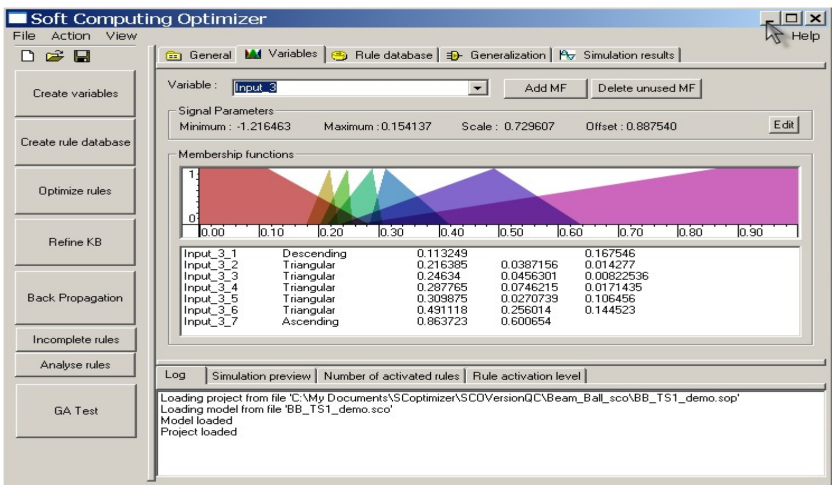


Figure 8. Example of designed MFs

As shown in this figure, for the description of «Input_3» values GA_1 finds seven fuzzy sets with triangle membership functions.

3.3. Rule database creation

After you have created all MFs for FC inputs (in our example they are «input1»,«input2» and «input3») you can create **rule database**. You can do it by pressing "*Create rule database*" command button.

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 9 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.

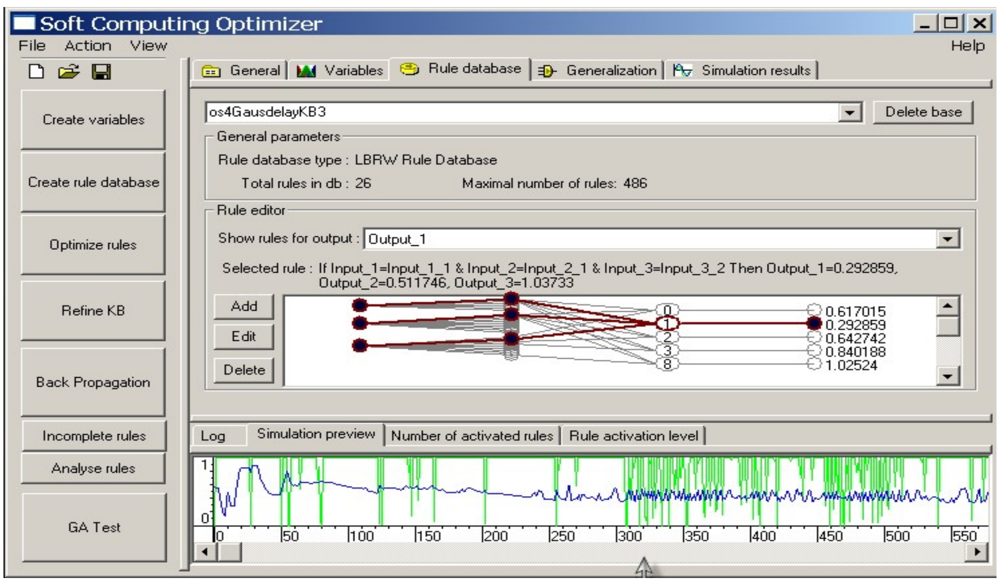


Figure 9. Example of designed rules database

On Figure9 in the line named «*Selected rule*» is shown the chosen fuzzy rule (red bolt line on the FNN structure; order number of the chosen rule = 1). This rule is written in the symbolic form as follows: « **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 in Figure 9, the result of teaching signal (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.

3.4. Rule database optimization

After rule database created, proceed to their optimization by GA_2. Press «**Optimize rules**» and the window is opened.Therearethreepossibilities:

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

You should select output variables for which database should be optimized. By default, optimization selected for all variables and you should not change it when starting algorithm for the first time.During optimization, a progress window will appear. It displays variables currently optimized, 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 pressed and the optimization will switch to the next variable. Press Abort All to stop optimization process and return to SCO.

As the result of GA_2 optimization we obtain the optimal values of right parts of fuzzy rules.

Remark. GA_2 optimization is based on TS. If TS is not optimal (from the control quality criterion), GA_2 optimization may be not optimal too. For that case in SCO toolkit there is an effective way - RD optimization by *Matlab simulation*.

For RD optimization by Matlab simulation,there is a special option «*Matlab simulation*».

3.5. Fine tuning of the model

When rule database optimized you can further improve a control model quality by returning to MFs optimization. This accomplished by the last optimization step model refinement (known as GA_3 algorithm). You can start model refinement by clicking «*Refine KB*» command button. After you activate the command wizard dialog will appear. It will first prompt you which fitness function you would like to use. in this case threevariantsare available:

- Maximization of mutual information entropy: Tells SCO to minimize mutual information entropy between MF fuzzy sets. This is the same function used in GA_1 algorithm, but unlike GA_1, GA_3 will not change number of MF's per variable, only MF parameters will changed.
- Minimization of output error.
- Matlab simulation: use Matlab/Simulink to calculate fitness function.

Now you should select input variables, which should optimized. By default, optimization selected for all variables. While GA_3 algorithm operates, the progress dialog 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 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 (GA_2) 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.

4. Example. 3DOF Manipulator control system

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

4.1. Description of the 3DOF Manipulator Test Bench

Figure 10 shows the test bench which is used to the test control system.As the measurement system (MS) 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 10). Information about the current positions of the links and the characteristics of the quality of control displayed on the LCD and serial interface.

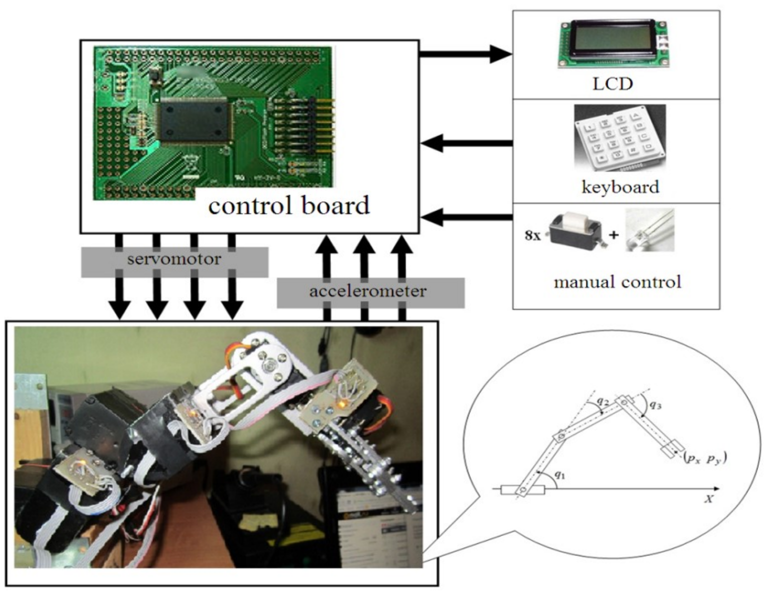


Figure 10. The manipulator test bench

Both automatic and manual control modes supported (the ability to move each of the three links and the manipulator's grip device using the manual control buttons). In robotics, as a rule, a mathematical model of the manipulator built, simulation of the CO, identification of the parameters of the mathematical model. Then comparison of the simulation results on the mathematical model of the CO and test bench of robot manipulator performed. 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 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. Control Tasks

On the Figure 11 shown the direct circuit of the control loop by the 3DOF manipulator to explain the operation with a PID controller.
In Figure 11: $E=[\varepsilon_1\ \varepsilon_2\ \varepsilon_3]$ is a control error $K_{Pi}, K_{Di}, K_{Ii},\ i=\overline{1,3}$ 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. The control task reduced to finding the coefficients of the PID controller $K_{Pi}, K_{Di}, K_{Ii},\ i=\overline{1,3}$, which ensures the desired movement.

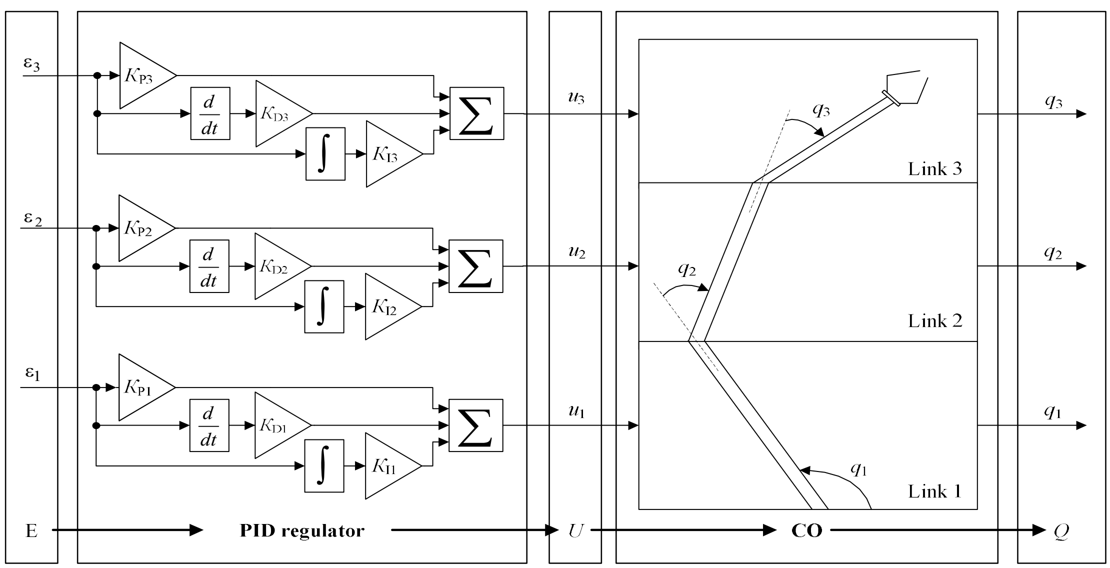


Figure 11. Direct circuit of control system with PID controller

4.3. Test Procedure

A series of experiments 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 carried out in standard and unexpected control situations and evaluated according to the quality criteria introduced above. As standard control situations, ten experiments performed in accordance with a group of workspace test points (Figure 12).

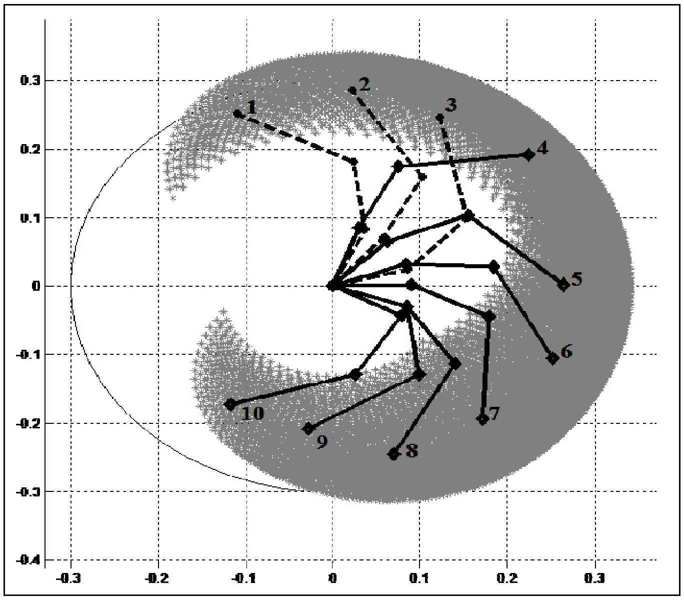


Figure12. Test points

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 $Q=[q_1;q_2;q_3]=[60;\,45;\,-43]$degrees;
3) the initial conditions are changed $Q=[q_1;q_2;q_3]=[60;\,45;\,-43]$degrees: and the position of the second link is changed to the value $q_2 = 45$ degrees at the 11th iteration.
Three unexpected situations tested at ten points in the test space. Thus, 30 experiments conducted for unexpected control situations.
Consider the features of the design of ICS based on SCO for 3DOF robot manipulator.

4.4. ICS based on SCO

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 SCO for a 3DOF robot manipulator is possible both with one FC and with separated control.
Let us consider the process of creating KB for the ICS.

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 defined:

1) the type of fuzzy model: Sugeno 0 (zero order);

2) the 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 × n2 × n3 × n4 × n5 × n6 × n7 × n8 × 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.*

At this stage the traditional method of error back propagating is used.

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. The dependence of the maximum number of fuzzy rules on the number of degrees of freedom of the manipulator increases lineally. But even in this case we have a huge number of rules in designed KB.

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 increase.

We made the following *conclusion*: *if it is difficult to implement a single KB, we will divide the KB into several, and will 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.
Now we describe the process of creating KB.
1. Creating TS.
We created three 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 three 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 defined for each of KB:
1) the type of fuzzy model: Sugeno 0;
2) the 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 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 used at this stage.

4.5. Modeling and test bench: control quality

Figure 13 and 14 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.

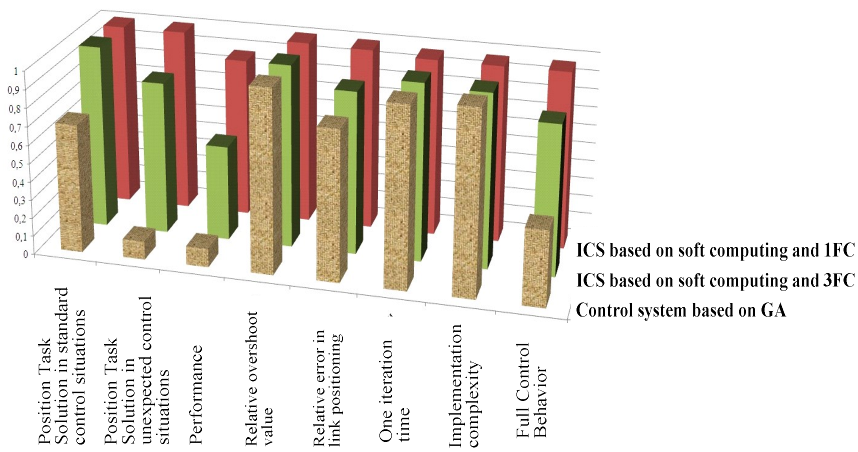


Figure 13.Comparison of quality criteria for a control system based on GA, ICS based on KBO on soft computing with one FC and ICS based on soft computing with separated control for MatLab / Simulink models

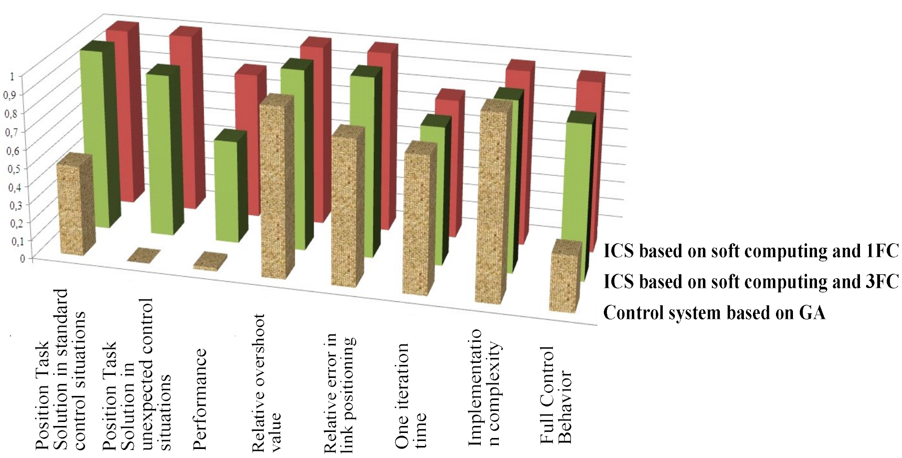
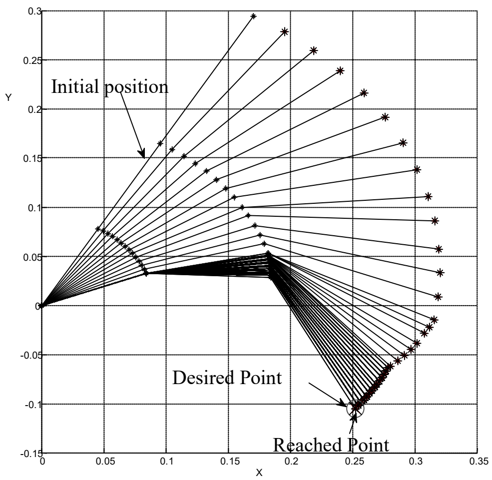
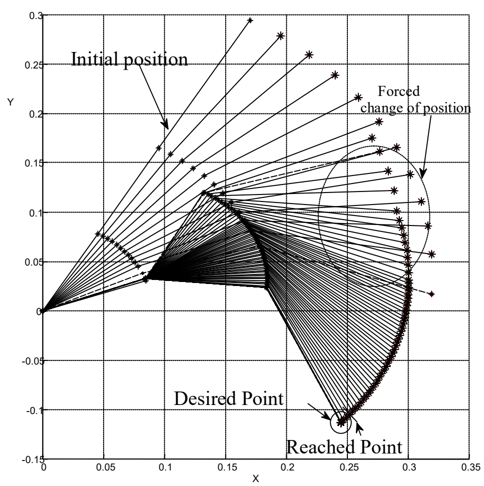


Figure 14.Comparison of quality criteria for a control system based on GA, ICS based on KBO on soft computing with one FC and ICS based on soft computing with separated control for the robot manipulator test bench

It can see 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 15). The full control behavior is rather low.

Figure 15(b) shows the movement of the manipulator in an external unexpected situation.





(a)

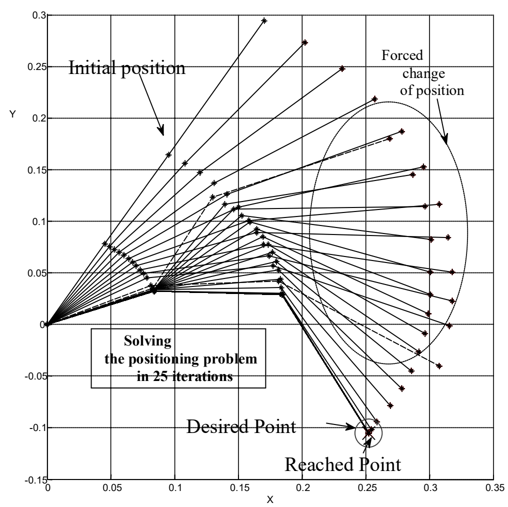
(b)

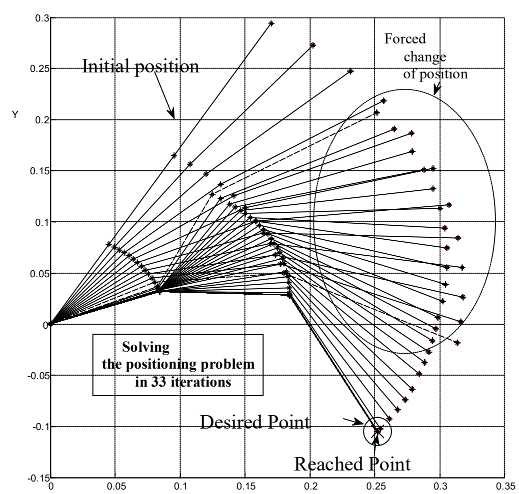
Figure 15. The operation of the control system based on GA: in a standard control situation (a); in an unexpected control situation (b)

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.

Figure 15 shows the work of the ICS based on SCO with one FC and separated control in an unexpected control situation, previously proposed for a control system based on GA.

From Figure 15 and Figure 16, 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.





(a) (b)

Figure 16. 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)

Current conclusions

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

1) to obtain a maximum of a quality criteria position task solution as in standard and in 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;
3) ICS based on KBO on soft computing with one FC allows us 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 three 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 KB optimization on soft computing with divided control with three 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 17 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 17*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.

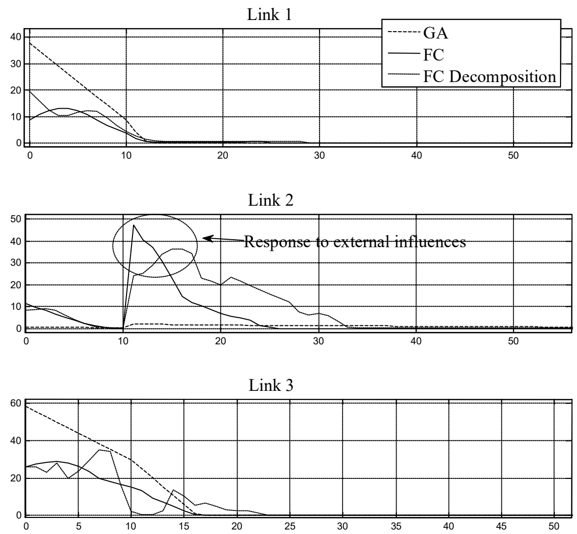


Figure 17. Control signals generated by the control system based on GA, ICS based on KBO on soft computing with one FC and ICS on soft computing with separated control

From Figure 17 you can see that the control signals generated by the control system based on GA for the first and third links have the large amplitude compared to the similar control signals generated by ICS based on SCO. For the second link in the control signal, formed by the control system based on GA, the reaction to external influence not sufficiently reflected, because of which the task of precise positioning not solved.

The control signals generated by ICS based on SCO with separate control, compared with ICS with one FC, with a comparable amplitude, have a greater overshoot.

Thus, the minimum consumption of a useful resource in the formation of control signals ensured when using the ICS based on SCO with one FC.

5. Conclusion

Brief introduction on the SC Optimizer tools for designing robust FC'sintroduced. Robustness capabilities of designed KB's for many control situations investigated. To control robots with manipulators of varying complexity, the following factors considered:1) control systems with constant coefficients of the PID controller; and 2) control systems with adjustable PID controller coefficients depending on the situation.

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 (conventional) situations*[10].

2. Computational intelligence toolkit called as SCO realizes a deep machine learning with an optimal structure of FNN and reduces redundant information in production a robust set of fuzzy logical rules (robust KB).
3. A unified KB of the ICS based on SCO with one FC contains the most complete information about the behavior of all links. It 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. Therefore, the implementation of a single KB, for example, for a complex 7DOF robot manipulator is not possible.
4. Most important decision-makingis a selection of the generalization strategy, which will switch the flow of control signals from different FC, and if necessary will modify their output to fit present control object conditions. For this purpose, the simplest way is the application of weighted aggregation of outputs of each independent FC.But this solution will fail and distribution of weighting factors should be somehow dynamically decided.
5. Solution of such kind of generalization problems by introducing a *self-organization* design process of KB-FC that supported by the *Quantum Fuzzy Inference* (QFI) based on Quantum Soft Computing ideas [13]. This problem considered in the Part III.The method of organizing coordination control using quantumsoft computing technologies to create robust ICS for 3DOF and 7DOF manipulators demonstrated.

In particular, in the next Part II and III, 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 considered.

References

1. Ulyanov, S. V. Self-organized Control System US patent, No. 6, 411, 944, 1997.
2. Ruano, A.Intelligent control systems using computational intelligence techniques. L.: IEE Control Series. – Vol. 70. – 2008.
3. Negnevitsky, M. Intelligent control design with MATLAB and SIMULINK. Intensive Course. Singapore, TEKBAC. 2011.
4. Behera, L. Intelligent systems and control principles and applications. Oxford, University Press. 2010.
5. 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 Comp. Syst. Sci. Intern., Vol. 43, No 3,2004. 615-632.
6. Ulyanov, S.V., Litvintseva, L.V. Soft computing optimizer of intelligent control system structures // US patent No 7,219,087 B2. 2007.
7. Ulyanov, S.V.,L.V. Litvintseva, System for soft computing simulation // US patent No 20060218108 A1. 2006.
8. Ulyanov, S. V., Litvintseva, L. V., 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: Application No. 2011617532 dated

11.10.2011 RF Registered in the Register of computer programs on December 1, 2011 [in Russian].

9. Ulyanov, S. V. Soft Computing Optimizer of Intelligent Control System Structures: US patent, No. 20,050,119,986, 2005.
10. 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 // System analysis in science and education: a network scientific publication, No 4, 2013. Access mode: http://www.sanse.ru/download/193 [in Russian].
11. Litvintseva L. V., Ulyanov S. V. and Ulyanov V. S. Design of robust knowledge bases of fuzzy controllers for intelligent control of substantially nonlinear dynamic systems: II. A soft computing optimizer and robustness of intelligent control systems, J. of Computer and Systems Sciences Intern., 2006, Vol. 45, № 5, pp. 744–771.
12. Ulyanov, S.V., Feng, M.,Yamafuji, K., Fukuda, T. Stochastic analysis of time-invariant non-linear dynamic systems. Pt 1: the Fokker-Planck-Kolmogorov equation approach in stochastic mechanic.Prob. Eng. Mech., 1998, Vol. 13, № 3, Pts 1&2.pp. 183 – 203; 205-226.
13. S.V. Ulyanov, Self-organization quantum robust control methods and systems for situations with uncertainty and risk. // US Patent No 8, 0345 874. – 2014.