



ARTICLE

Intelligent Robust Control of Redundant Smart Robotic Arm Pt II: Quantum Computing KB Optimizer Supremacy

Alena V. Nikolaeva^{1,2} Sergey V. Ulyanov^{3,4*}

1. Dubna State University, Russia
2. INESYS LLC (EFKO GROUP), Russia
3. Physical and Mathematical Sciences; JINR, Russia
4. LIT JINR, INESYS LLC (EFKO GROUP), Russia

ARTICLE INFO

Article history

Received: 7 November 2019

Accepted: 24 April 2020

Published Online: 30 June 2020

Keywords:

Quantum computing supremacy

Quantum-classical correlation

Knowledge base

Fuzzy controller

Quantum fuzzy inference

ABSTRACT

This article is a continuation of the work “Intelligent robust control of redundant smart robotic arm Pt I: Soft computing KB optimizer - deep machine learning IT”. In the first part of the paper, we examined control systems with constant coefficients of the conventional PID controller (based on genetic algorithm) and intelligent control systems based on soft computing technologies. For demonstration, MatLab / Simulink models and a test benchmark of the robot manipulator demonstrated. Advantages and limitations of intelligent control systems based on soft computing technology discussed. Intelligent main element of the control system based on soft computing is a fuzzy controller with a knowledge base in it. In the first part of the article, two ways to implement fuzzy controllers showed. First way applied one controller for all links of the manipulator and showed the best performance. However, such an implementation is not possible in complex control objects, such as a manipulator with seven degrees of freedom (7DOF). The second way use of separated control when an independent fuzzy controller controls each link. The control decomposition due to a slight decrease in the quality of management has greatly simplified the processes of creating and placing knowledge bases. In this Pt II, to eliminate the mismatch of the work of separate independent fuzzy controllers, methods for organizing coordination control based on quantum computing technologies to create robust intelligent control systems for robotic manipulators with 3DOF and 7DOF described. Quantum supremacy of developed end-to-end IT design of robust intelligent control systems demonstrated.

**Corresponding Author:*

Sergey V. Ulyanov,

Physical and Mathematical Sciences; JINR, Russia; LIT JINR, INESYS LLC (EFKO GROUP), Russia;

Email: ulyanovsv@mail.ru

1. Intelligent Control System of Manipulator with Seven Degrees of Freedom

1.1 Intelligent Control System Based on Quantum Computing Knowledge Base Optimizer (QCOptKB™)

Such unconventional phenomena as entangled states, teleportation, superdense coding appear in the quantum logic of computational intelligence and quantum information theory, due to the failure to fulfill the classical law of distributivity.

Quantum computing technologies for creating self-organizing Knowledge Bases (KB) of intelligent regulators are considered in [1,2]. In [1] a new type of quantum search algorithm described on the generalized space of the KB designed for soft computing, and the quantum fuzzy algorithm for designing a generalized robust control signal applied. The Quantum Fuzzy Inference (QFI) [3] implements the self-organization of the KBs, based on the physical laws of the theory of quantum computing [4,5] and the application of four operators: superposition, quantum correlation, interference, and measurement. Operators superposition, quantum correlation and interference are unitary, invertible quantum operators, and the measurement operator is classical non-invertible.

QFI algorithm [3] consists of the following sequence steps:

(1) *Normalization*. Bringing input signals to normalized states.

(2) *Coding*. All normalized states $|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$ and $|1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$ constructed for the current input variables at the coding step.

Next, the probability amplitudes of quantum states $|0\rangle$ and $|1\rangle$ are calculated from the corresponding histograms.

(3) *Choosing a correlation type*. Selection of the type of quantum correlation for constructing control output signals.

We will consider three types of miscellaneous correlations: spatial, spatio-temporal and temporal [6], and in addition to this, correlations of various numbers of fuzzy controllers (FC) types. The dependence of the output signals determined by the correlation of the input coefficients sets, where each set is an entangled state [7].

(4) *Superposition*. The construction of a superposition of entangled states.

(5) *Measurement of an intelligent quantum state*. The choice of the state with the maximum probability amplitude [8,9].

(6) *Decoding*. The calculation of the output value of the corresponding signal.

(7) *Denormalization*. Output value calculation.

The connection diagram of the QFI block in the Intelligent Control System (ICS) is shown in Figure 1 where: Q_{REF} is the reference signal, Q' is the measured variable, E is the control error, $s(t)$ is a limitation of the control action, $m_c(t), m_{MS}(t)$ are noises in the control channels and in the measuring system, U is the control action, $d(t)$ is a delay in the Measuring System (MS), $f_o(t)$ is external action on the CO, Q is an adjustable value.

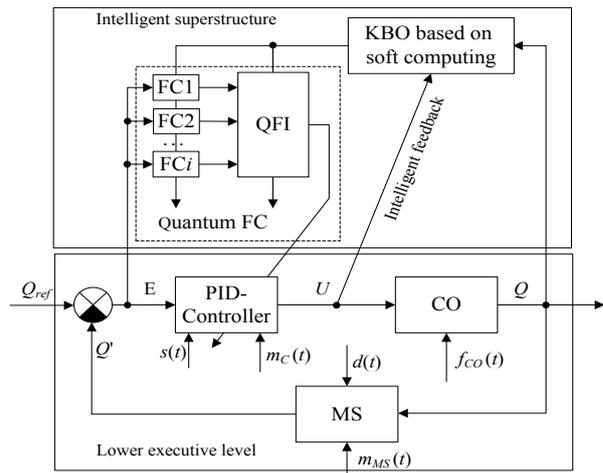


Figure 1. The structure of the ISC based on soft and quantum computing

The design of QFI is produced using the computational intelligence toolkit as Knowledge Base Optimizer (KBO) based on quantum computing [10].

Let us consider the method of embedding QFI block in the ICS structure based on KBO on soft computing.

1.2 Coordination Management Organization

Seven FC are implemented in the selected configuration of the ICS structure, each of FCs independently controls one of the seven links.

As a standard control situation for the i FC, a typical control situation appears, in the conditions of which a teaching signal TS_i received.

Unexpected control situations divided into external and internal. External unexpected situations include disturbing action, such as a forced change in the position of the CO, a change in the reference signals, etc. Internal control situations include changes in the parameters of the ICS components, for example restrictions, noise, and signal delays.

ICS based on soft computing (Figure 1), may contain information about seven control situations (standard or unexpected) for each of the links.

Let us consider an example when, in the designed ICS based on KBO on soft computing with separate control, FC1 contains information about the standard situation 1 for link 1, FC2 contains information about the standard situation 1 for link 2, etc.

Example. The QFI block in the ICS model based on KBO on Soft Computing implements the self-organization of the FC KB in learning situations. In this case, it is possible to combine information about the standard situation 1 for all links, and to extract additional information (by methods of quantum computing and quantum information theory) from the reaction of the designed KBs to implement robust control in regular and unexpected control situations (which are not included in existing designed KB). The KBs, whose responses used to design robust control, in unexpected situations may not be robust.

The connection diagram of the QFI block in the intelligent superstructure of the control system shown in Figure 2, on which blocks designed using soft and quantum-computing technologies highlighted.

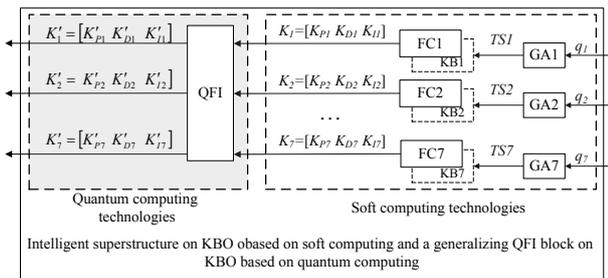


Figure 2. Intelligent add-on structure of the ICS based on KBO on quantum and soft computing

The ICS using the QFI called the ICS based on quantum computing KBO.

1.3 Test Procedure and Quality Criteria

A series of experiments in standard and unexpected control situations carried out to test the robustness of control system models. The following configuration is taken as the initial position of the manipulator: $q_1 = 0$ deg; $q_2 = 0$ deg; $q_3 = 0$ deg; $q_4 = 0$ deg; $q_5 = 0$ deg; $q_6 = 0$ deg; $q_7 = 0$ deg.

Emergencies divided into external and internal. External situations include a forced change of provisions and a change in initial conditions; unexpected situations include limitation of control actions, the introduction of noise into the control channels and errors in the MS.

A system of quality criteria that takes into account methods for assessing transients of automatic control theory [11] was introduced to evaluate and compare the test results of the considered types of control systems (control systems with constant coefficients and ICS for KBO on Soft and Quantum computing). The methods were adapted

for a specific CO (a 7DOF robot manipulator) and described in [12]:

- (1) Position Task Solution in known control situations PTS_{KCS} .
- (2) Position Task Solution in the external considered control situations PTS_{ACCS1} .
- (3) Position Task Solution in the internal considered control situations PTS_{ACCS2} .
- (4) Performance I_T .
- (5) Relative overshoot value σ .
- (6) Relative error in link positioning \hat{a} .
- (7) Time of one iteration t .
- (8) The complexity of implementing control P .
- (9) Full Control Behavior FCB .

1.4 The Quality of an Intelligent Control System with QFI

Control system models of 7DOF robot manipulator in MatLab/Simulink developed to research the quality of control systems.

The initial ICS on KBO on soft and quantum computing contains seven FCs, each of which can correlate with one or more NR systems.

Consider six combinations:

- (1) The correlation of 2 adjacent fuzzy KBs (the information FC i and $i+1$ is used to control the i -th link of the manipulator);
- (2) the correlation of 3 adjacent fuzzy KBs (the information FC i , $i+1$ and $i+2$ is used to control the i -th link of the manipulator, as shown on Figure 3);
- (3) the correlation of 4 adjacent fuzzy KBs;
- (4) the correlation of 5 adjacent fuzzy KBs;
- (5) the correlation of 6 adjacent fuzzy KBs;
- (6) the correlation of all 7 fuzzy KBs.

For selected six combination spatial, spatio-temporal and temporal correlations are considered.

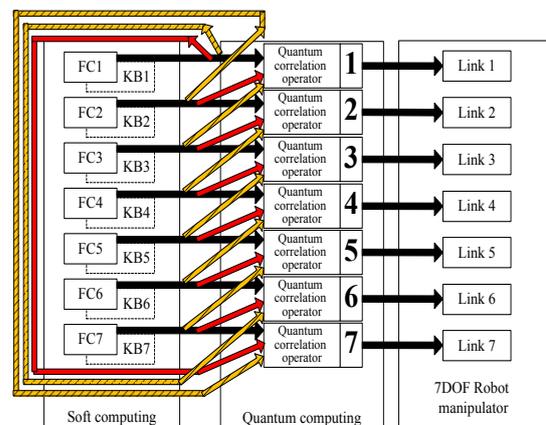


Figure 3. The application of the correlation of three neighboring FC

Comparison of the ICS operation based on Quantum KBO for spatial, spatio-temporal and temporal correlations shown in Figures 4-9 for selected combinations of correlated FC.

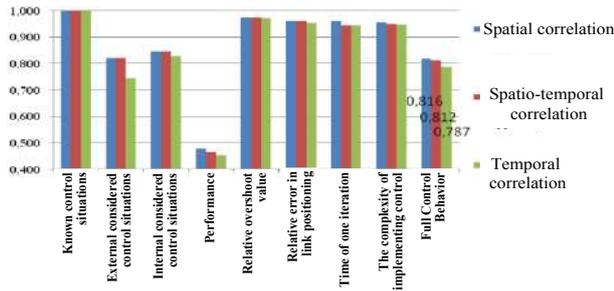


Figure 4. The result of the work of the ICS on quantum computing KBO using the correlation of 2 FC

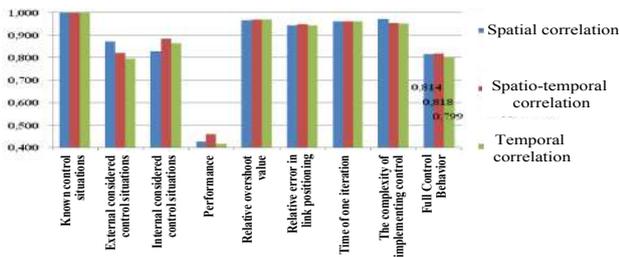


Figure 5. The result of the work of the ICS on quantum computing KBO using the correlation of 3 FC

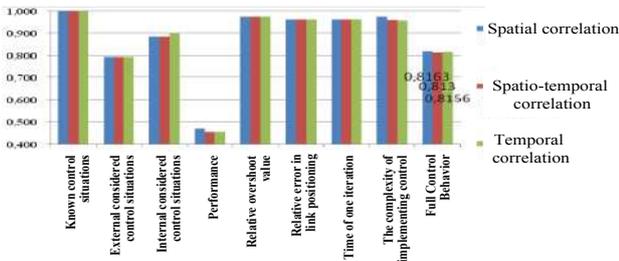


Figure 6. The result of the work of the ICS on quantum computing KBO using the correlation of 4 FC

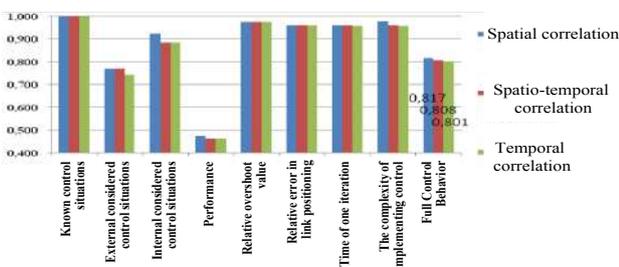


Figure 7. The result of the work of the ICS on quantum computing KBO using the correlation of 5 FC

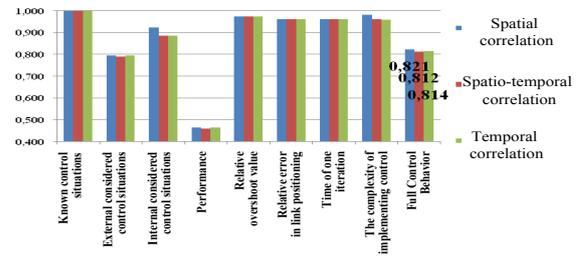


Figure 8. The result of the work of the ICS on quantum computing KBO using the correlation of 6 FC

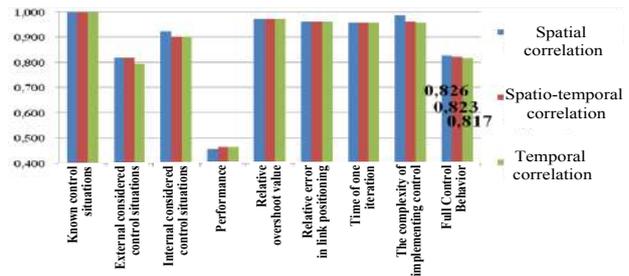


Figure 9. The result of the work of the ICS on quantum computing KBO using the correlation of 7 FC

We conclude from the results:

- (1) for ICS based on quantum KBO with the correlation of 2 FCs, the spatial correlation is the best (Figure 4);
- (2) for ICS based on quantum KBO with the correlation of 3 FCs, the spatio-temporal correlation is the best (Figure 5);
- (3) for ICS based on quantum KBO with the correlation of 4 FCs, the spatial correlation is the best (Figure 6);
- (4) for ICS based on quantum KBO with the correlation of 5 FCs, the spatial correlation is the best (Figure 7);
- (5) for ICS based on quantum KBO with the correlation of 6 FCs, the spatial correlation is the best (Figure 8);
- (6) for ICS based on quantum KBO with the correlation of 7 FCs, the spatial correlation is the best (Figure 9);

Next, we compare the work of the ICS based on quantum KBO using spatial correlation with the work of the ICS based on soft computing KBO (Table 1).

Application of QFI allows the following results:

- (1) to solve the problem of positioning in regular situations using either correlation type;
- (2) to improve the results of the positioning task in conditions of external unexpected situations; results do not improve in internal unexpected situations;
- (3) to increased performance criterion by 5 times;
- (4) the overall control result is improved; the best result is achieved using the spatial correlation of 7 FCs (Figure 10 shows a comparison with ICS on soft computing).

Table 1. Comparison ICS based on quantum computing and ICS based on soft computing

	Control situation models	ICS based on soft KBO	ICS based on quantum KBO					
			Spatial correlation of 2 FC	Spatio-temporal correlation of 3 FC	Spatial correlation of 4 FC	Spatial correlation of 5 FC	Spatial correlation of 6 FC	Spatial correlation of 7 FC
1	Known control situations PTS_{KCS}	0,923	1,000	1,000	1,000	1,000	1,000	1,000
2	External considered control situations $PT-S_{ACCS1}$	0,744	0,821	0,821	0,795	0,769	0,795	0,821
3	Internal considered control situations PTS_{ACCS2}	0,923	0,846	0,885	0,885	0,923	0,923	0,923
4	Performance I_T	0,092	0,477	0,461	0,472	0,476	0,465	0,459
5	Relative overshoot value σ	0,969	0,973	0,968	0,974	0,974	0,974	0,971
6	Relative error in link positioning \hat{a}	0,911	0,962	0,947	0,962	0,962	0,962	0,960
7	Time of one iteration t	0,973	0,961	0,962	0,961	0,961	0,961	0,960
8	The complexity of implementing control P	0,946	0,957	0,953	0,973	0,977	0,982	0,986
9	Full Control Behavior FCB	0,721	0,816	0,818	0,816	0,817	0,821	0,826

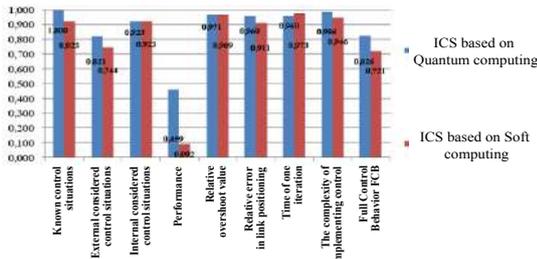


Figure 10. Comparison of ICS based on soft KBO and ICS based on quantum KBO using spatial correlation of 7 FC

Figures 11-12 demonstrate the operation of the manipulator when using the ICS based on soft KBO and the ICS based on quantum KBO in the conditions of a standard control situation and an external unexpected control situation (the initial position has been changed).

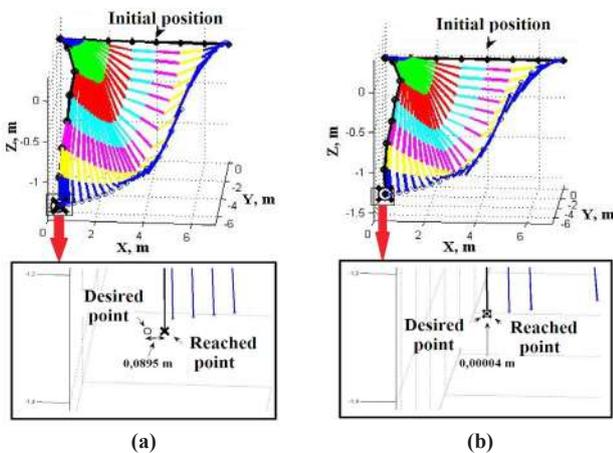


Figure 11. The movement of the robot manipulator in a standard control situation: under control of ICS based on soft computing (a); ISU on ICS based on quantum computing (b)

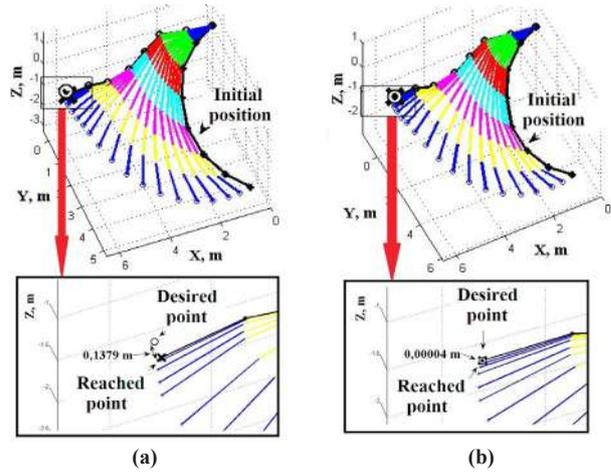


Figure 12. The movement of the robot manipulator in an external unexpected control situation: under control of ICS based on soft computing (a); ISU on ICS based on quantum computing (b)

From Figures 11-12 we can conclude that ICS based on quantum computing KBO under the conditions of the standard and external unexpected control situations solves the problem of precise positioning of a 7DOF robot manipulator, in contrast ICS based on soft KBO.

A comparison of the phase portraits of the ICS based on the KBO on soft computing and the ICS based on the KBO on quantum computing is shown in Figures 13 and 14 for the considered standard and external unexpected control situations.

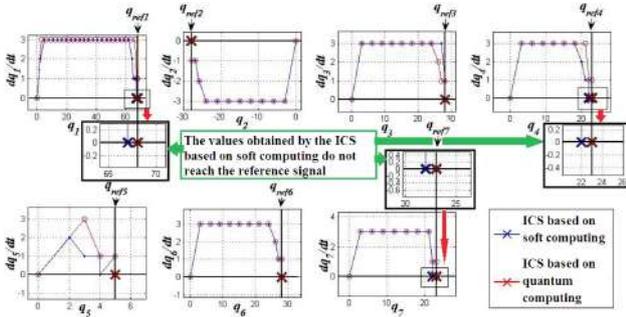


Figure 13. Changing the position of the manipulator links in a standard control situation: a comparison of ICS based on soft computing KBO and ICS based on quantum computing KBO

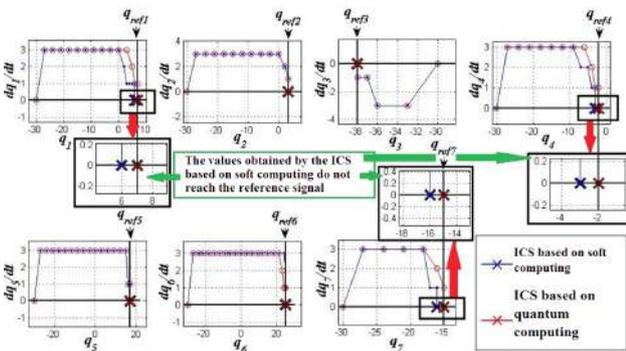


Figure 14. Changing the position of the manipulator links in an external control situation: a comparison of the ICS based on soft computing KBO and the ICS based on quantum computing KBO

The application of ICS based on quantum computing in solving of positioning problems in internal unexpected situations did not help to improve the results compared to ICS based on soft computing. Let us consider in more detail internal unexpected situations and increase internal noises.

2. The Quality of Work of an ICS based on QFI under Conditions of Strong Internal Interference

Earlier, the following internal unexpected situations with the introduction of interference considered:

- (1) Introducing noise into the control channels (noise amplitude = 1);
- (2) Making errors in the MS (1,5 degrees).

Let us consider the behavior of ICS under conditions of strong interference. Previously, the solution to the positioning problem was determined by the most accurate positioning of manipulator links:

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

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 links positions.

However, under conditions of strong interference, it is almost impossible to reach accurate positioning of each link. The criterion Performance IT can be excluded from consideration (time will always be equal to the number of allowed iterations).

We formulate the general problem: to position the functional device of the manipulator as accurately as possible. We introduce a new quality criterion: a metric error in the positioning of the manipulator end effector EEP (End Effector Positioning):

$$EEP = \sqrt{(X_{ref} - X)^2 + (Y_{ref} - Y)^2 + (Z_{ref} - Z)^2},$$

where $\{X, Y, Z\}$ are the coordinates of the reached point, $\{X_{ref}, Y_{ref}, Z_{ref}\}$ are the coordinates of the desired point.

$$EEP_{PERFORMANCE} \equiv \frac{\left(\sum_{i=1}^N \frac{EEP_i}{EEP_{max}} \right)}{N},$$

where N is the number of experiments, EEP_{max} is the maximum metric error from the estimated series of experiments.

Updated system of quality criteria for evaluating the operation of control systems under conditions of strong internal interference:

- (1) Metric positioning error of the manipulator end effector EEP.
- (2) Relative overshoot value σ .
- (3) Relative error in link positioning ε .
- (4) Time of one iteration t .
- (5) The complexity of implementing control P .
- (6) Full Control Behavior FCB.

$$FCB \equiv w_1 \cdot P[EEP] + w_2 \cdot P[\sigma] + w_3 \cdot P[\varepsilon] + w_4 \cdot P[t] + w_5 \cdot P,$$

where $w = [0,4 \ 0,1 \ 0,1 \ 0,2 \ 0,2]$ is the weight's vector.

Consider an unexpected situation with the introduction of a strong interference in the MS.

2.1 The Quality of Control Systems in Conditions of Strong Disturbing Interference in the MS

Previously, an internal unexpected situation was consid-

ered with the introduction of random noise with an amplitude = 1,5 into the MS.

Increase the amplitude of the introduced random noise to 5 degrees.

We carry out the following series of experiments: for 13 points of the manipulator’s working space, 30 random noises generated. The generated noises introduced into the MS of ICS based on the soft KBO with separate control and into the ICS based on the quantum KBO.

In Figure 15 shows a comparison of metric errors in positioning the functional device of the manipulator based on soft and quantum computing for 380 experiments.

Metric error according to the results of 380 experiments, when using QFI, it is less than when using only soft computing (Table 2).

Let us compare the work of the ICS based on soft KBO and the ICS based on quantum KBO in accordance with the introduced system of quality criteria Figure 16.

From Figure 16 can concluded that the ICS based on quantum KBO by all criteria (except for the time of one iteration) is better to the ICS based on soft KBO.

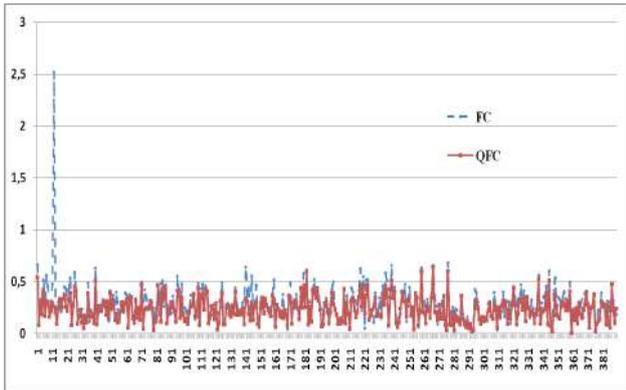


Figure 15. Metric positioning error of the manipulator end effector [m]: *FC* is result of ICS based on soft computing; *QFC* is result of ICS based on quantum computing

Table 2. Comparison of the results of ICS based on KBO on quantum and soft computing

Total number of experiments	390	
The number of experiments in which ICS based on quantum computing worsen the result than when using only soft computing:	87	
The number of experiments where the error increased by N%	0..30%	61
	31..60%	17
	61..90%	5
	91..120%	2
The number of experiments in which ICS based on quantum computing improve the result than using only soft computing:	121..150%	2
	303	

The number of experiments where the error decreased by N%	0..30%	79
	31..60%	92
	61..90%	38
	91..120%	23
	121..150%	11
	150..200%	12
	200..250%	9
	250..300%	6
The result has not changed.		11

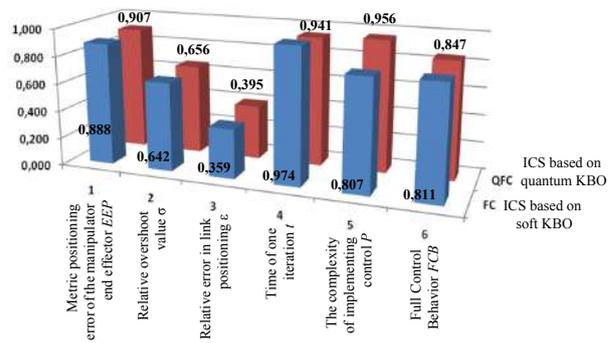


Figure 16. The quality of the ICS based on soft and quantum computing in the conditions of strong interference in the MS

Example. Let us consider an example of the influence of noise in a control channel of ICS based on soft and quantum computing. In Figure 17 shows an example of noise in a control channel. In Figure 18 shows a comparison of the control signals during the operation of the ICS based on soft (a) and quantum (b) computing. In Figure 19 shows the position of the manipulator when operating the ICS based on soft (a) and quantum computing (b).

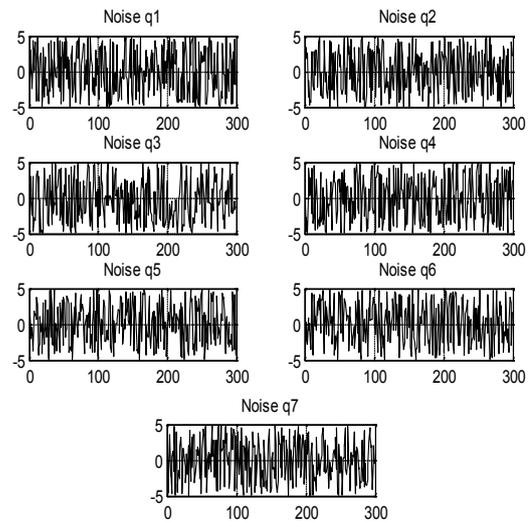
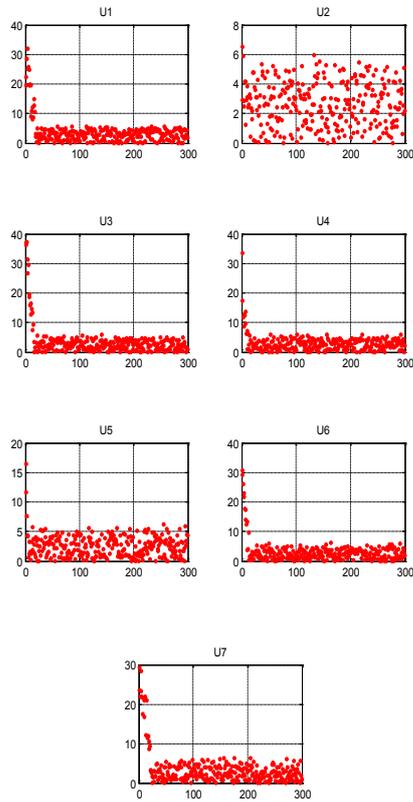
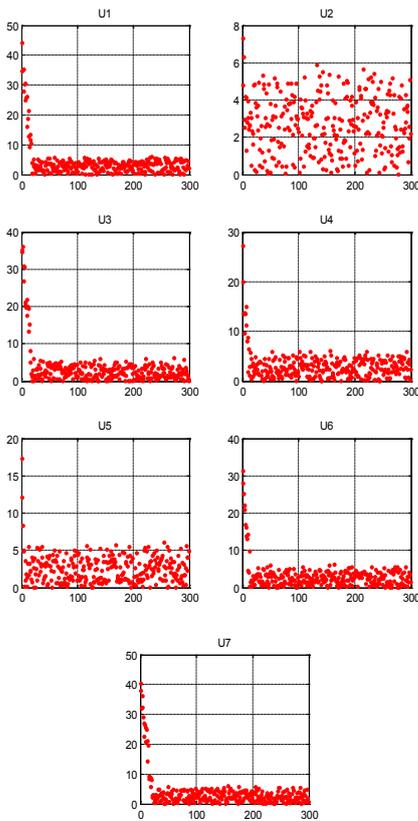


Figure 17. Noise in the control channel



(a)



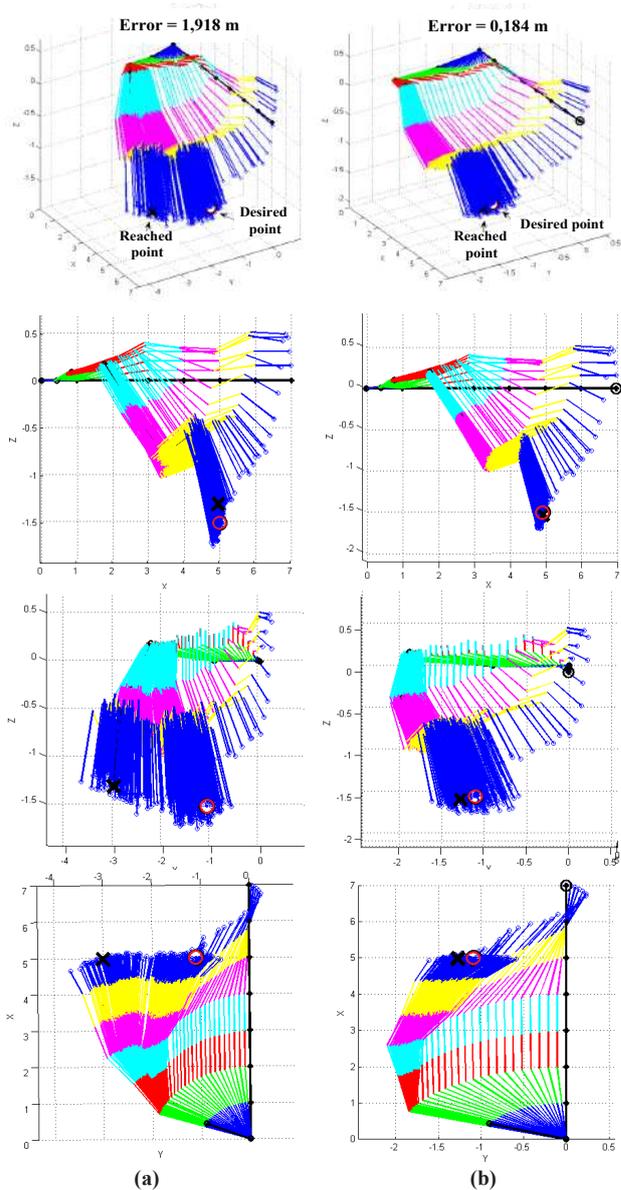
(b)

Figure 18. Control signals generated by ICS based on soft (a) and quantum (b) computing

The error in positioning an end effector when using only soft computing does not exceed 0,3 meters with a length of one link of 1 m.

We will generate noise at which the metric error of positioning the functional device during the operation of the ICS based on the KBO on soft computing will be more than 0, 5 m and, with the same noise, we will evaluate the behavior of the ICS based on quantum computing. To do this, increase the amplitude of the noise introduced into the control channels to 100.

Five random noises generated for 13 workspace points. Generated noises introduced into the control channels of the ICS based on KBO on soft computing with separate control and the ICS based on KBO on quantum computing.



(a)

(b)

Figure 19. The position of the manipulator in space when operating ICS based on soft (a) and quantum computing (b)

Figure 20 shows a comparison of metric errors in positioning the manipulator functional device of the ICS based on KBO on soft and quantum computing for 65 experiments.

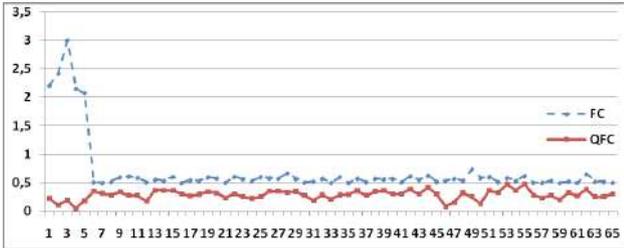


Figure 20. Functional device metric positioning error [m]: *FC* is the result of ICS operation based on soft computing, *QFC* is the result of ICS operation based on quantum computing

According to the results of 65 experiments, we conclude that the metric error obtained when using quantum computing is much smaller than when using only soft computing.

Let us compare the work of the ICS based on KBO on soft and quantum computing in accordance with the introduced system of quality criteria (Figure 21)

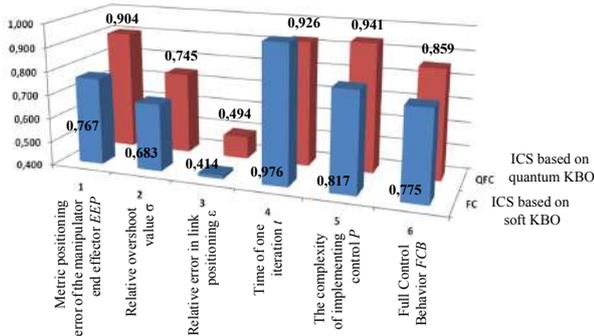


Figure 21. The quality of the ICM based on KBO on soft and quantum computing in the conditions of strong noise in the control channels

It can be seen from Figure 21 that ICS based on KBO on quantum computing by all criteria (with the exception one iteration time) is superior to the ICS than based on KBO on soft computing.

Example. Let us consider an example of the influence of noise in control channel. In Figure 22 is an example of noise in a control channel. In Figure 23 shows a comparison of control signals. In Figure 24 shows the position of the manipulator when operating ICS based on soft (a) and quantum computing (b).

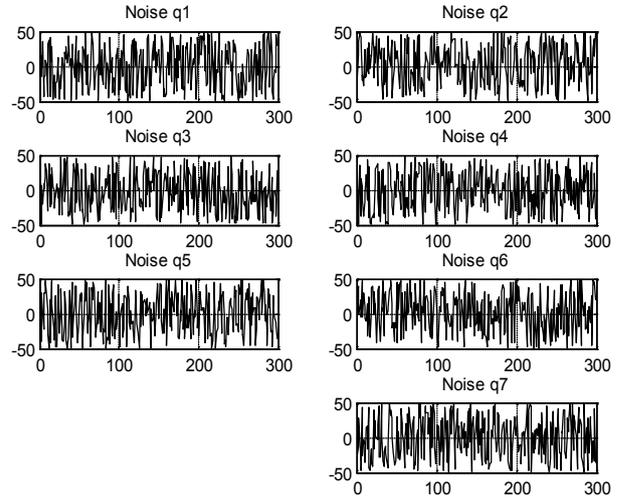
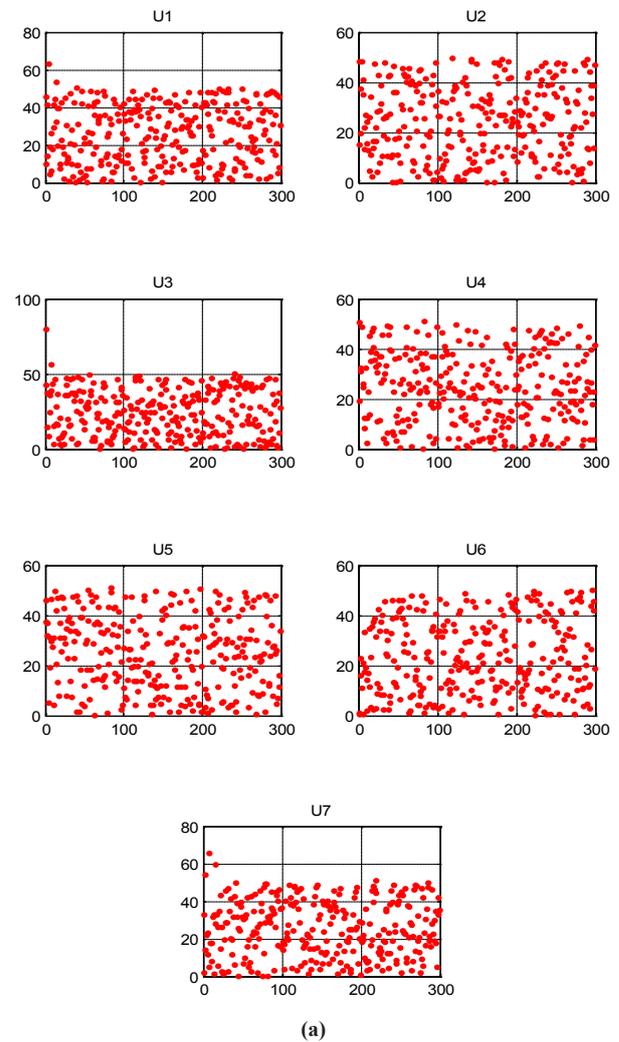


Figure 22. Noise in the control channel



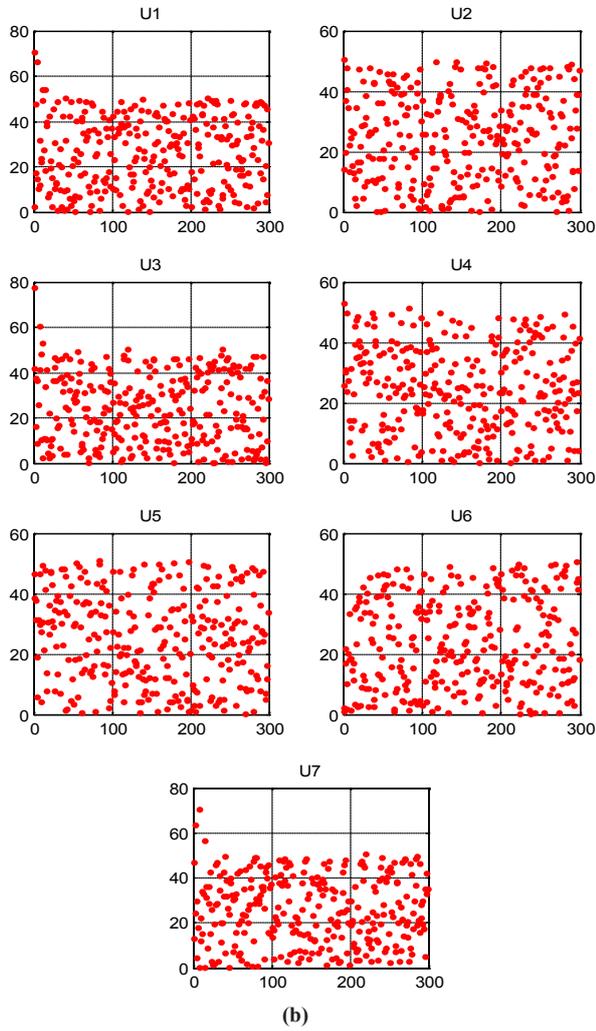


Figure 23. Control signals generated by ICS based on soft (a) and quantum (b) computing

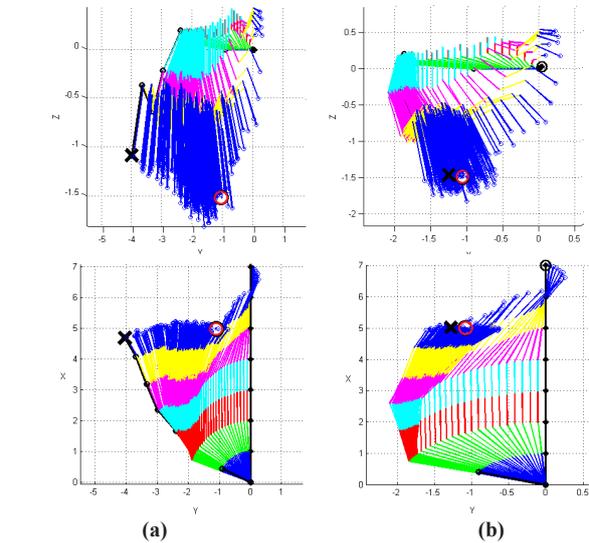
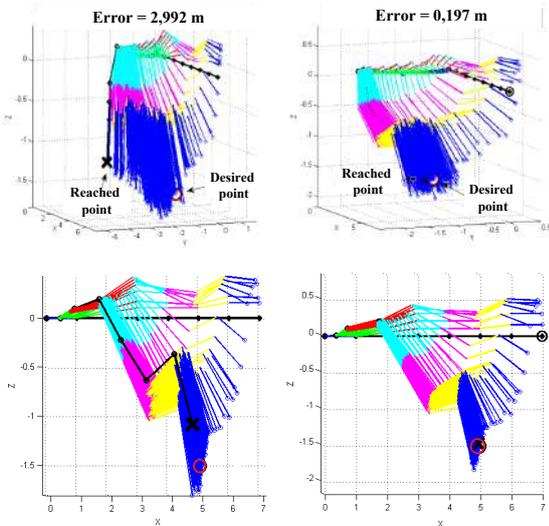


Figure 24. The position of the manipulator in space when operating ICS based on soft (a) and quantum computing (b)

In the considered unexpected situations with the introduction of strong noise into the control channels, the ICS based on KBO on quantum computing showed an advantage over ICS based on KBO on soft computing.

Consider an unexpected situation with the introduction of a strong disturbing effect in MS.

2.2. The Quality of Control Systems in Conditions of Strong Disturbing Influences in the Measurement System

Previously, an internal unexpected situation was considered with the introduction of random noise with an amplitude = 1.5 into MS. Increase the amplitude of the introduced random noise to 5 degrees, 30 random noises generated for 13 points of the manipulator workspace. The generated noise introduced into MS of the ICS based on KBO on soft computing with separate control and ICS on quantum computing.

In Figure 25 shows a comparison of the metric errors in positioning the manipulator functional device of ICS based on KBO on soft and quantum computing for 380 experiments.

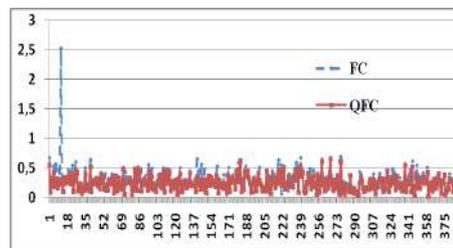


Figure 25. Functional device metric positioning error [m]: *FC* is the result of ICS operation based on soft computing, *QFC* is the result of ICS operation based on quantum computing

According to the results of 380 experiments, we conclude that the metric error obtained when using quantum computing is much smaller than when using only soft computing (Table 3).

It can see from Figure 26 that ICS based on KBO on quantum computing by all criteria (with the exception one iteration time) is superior to the ICS based on KBO on soft computing.

Example. Let us consider an example of the influence of noise in MS. In Figure 27 is an example of noise in MS. In Figure 28 shows a comparison of control signals. In Figure 29 shows the position of the manipulator when operating ICS based on soft (a) and quantum computing

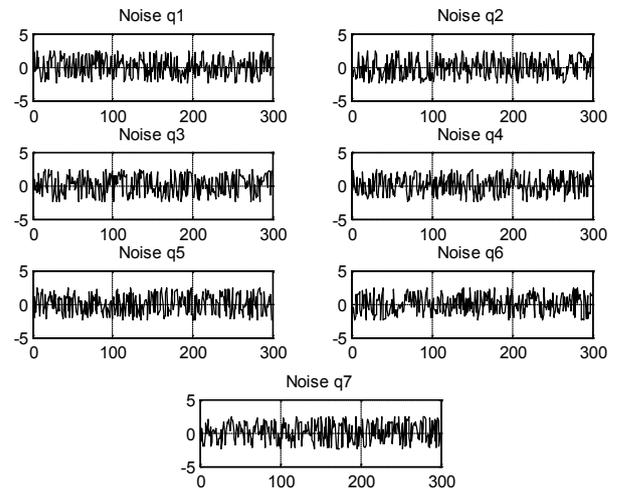


Figure 27. Noise in the control channel

Table 3. Comparison of the results of ICS based on KBO on quantum and soft computing

Total number of experiments		390
The number of experiments in which ICS based on quantum computing worsen the result than when using only soft computing:		105
The number of experiments where the error increased by N%	0..30%	80
	31..60%	21
	61..90%	2
	91..120%	0
	121..150%	1
The number of experiments in which ICS based on quantum computing improve the result than using only soft computing:		285
The number of experiments where the error decreased by N%	0..30%	131
	31..60%	79
	61..90%	24
	91..120%	16
	121..150%	9
	150..200%	4
	200..250%	4
	250..300%	2
300 and more %		4
The result has not changed.		11

Let us compare the work of the ICS based on KBO on soft and quantum computing in accordance with the introduced system of quality criteria (Figure 26)

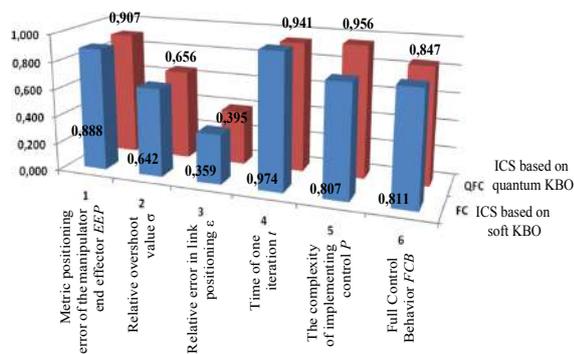
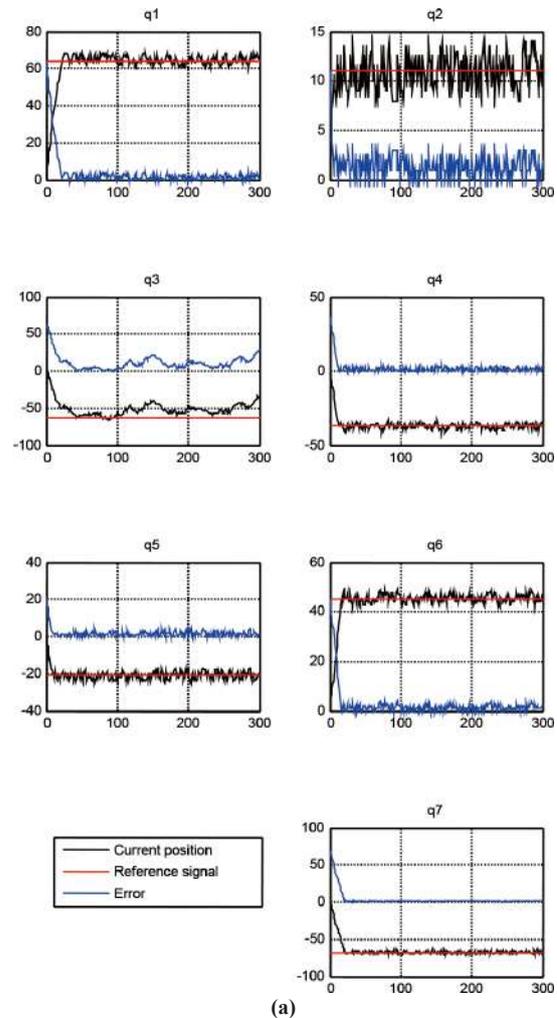


Figure 26. The quality of the ICM based on KBO on soft and quantum computing in the conditions of strong noise in MS



(a)

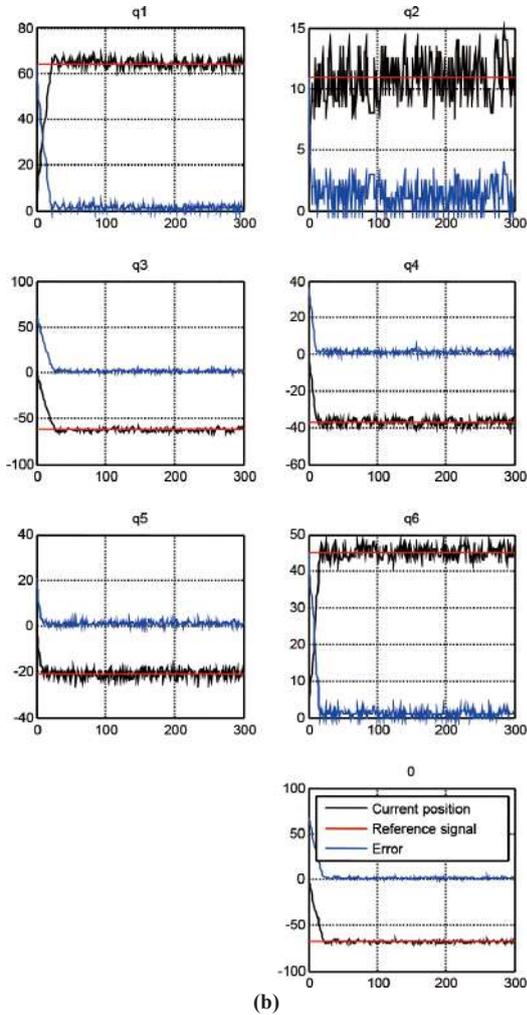


Figure 28. Measured signals during operation ICS based on soft (a) and quantum (b) computing

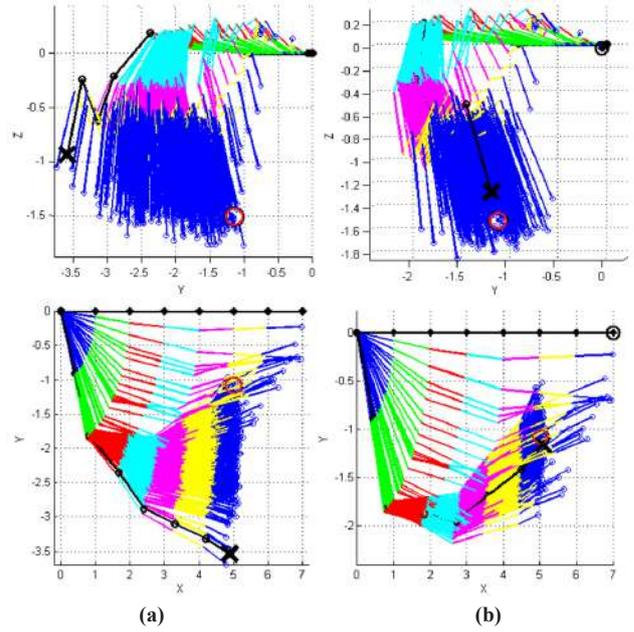
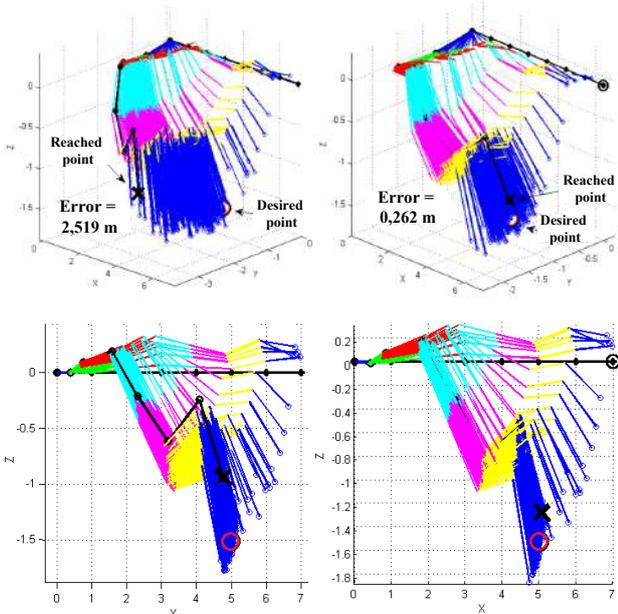


Figure 29. The position of the manipulator in space when operating ICS based on soft (a) and quantum computing (b)

The use of ICS based on KBO on quantum computing allows us slightly improve the performance compared to ICS based on KBO on soft computing, but the improvement is insignificant.

In general, the positioning error of a functional device when using only soft computing does not exceed 0,3 m with a single link length of 1 m.

In Figure 30a comparison of metric errors of positioning the functional device manipulator of the ICS based on soft and quantum KBO for 65 experiments is presented.

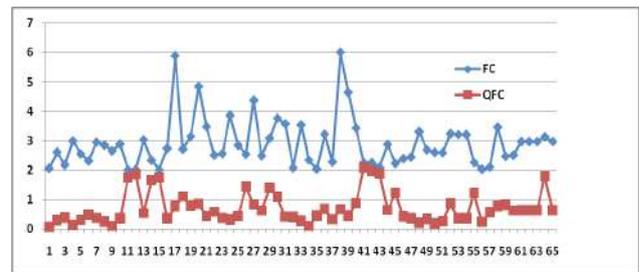


Figure 30. Functional device metric positioning error [m]: *FC* is the result of ICS operation based on soft computing, *QFC* is the result of ICS operation based on quantum computing

We conclude from the results of 65 experiments: the metric error obtained when using quantum computing is much smaller than when using only soft computing.

Let us compare the work of the ICS based on soft and quantum KBO in accordance with the introduced system

of quality criteria (Figure 31).

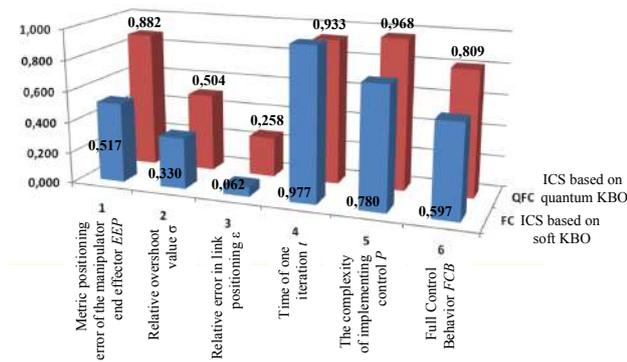


Figure 31. The quality of the ICS based on soft and quantum computing in the conditions of strong interference in the MS

From Figure 31 it can be seen that the ICS based on quantum KBO by all criteria (except for the time of one iteration) is better to the ICS based on soft computing.

Example. Let us consider an example of the interference in a MS on the operation of an ICS in soft and quantum computing. In Figure 32 shows an example of noise in a MS.

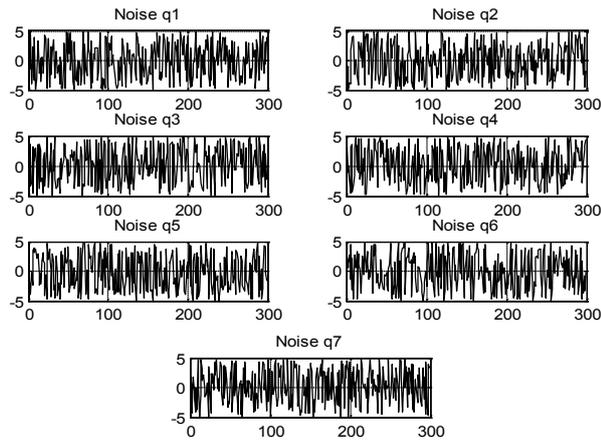


Figure 32. Noise in the MS

In Figure 33 shows a comparison of the measured signals during the operation of the ICS based on soft (a) and quantum (b) computing. In Figure 34 shows the position of the manipulator when operating the ICS based on soft (a) and quantum computing (b).

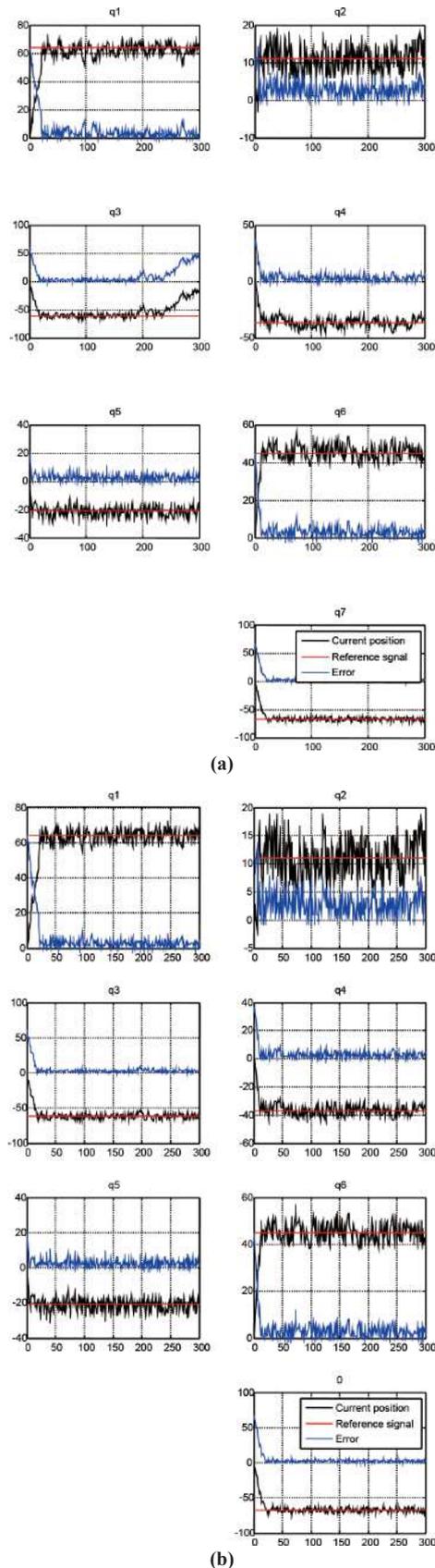


Figure 33. Measured signals during ICS based on soft (a) and quantum (b) computing

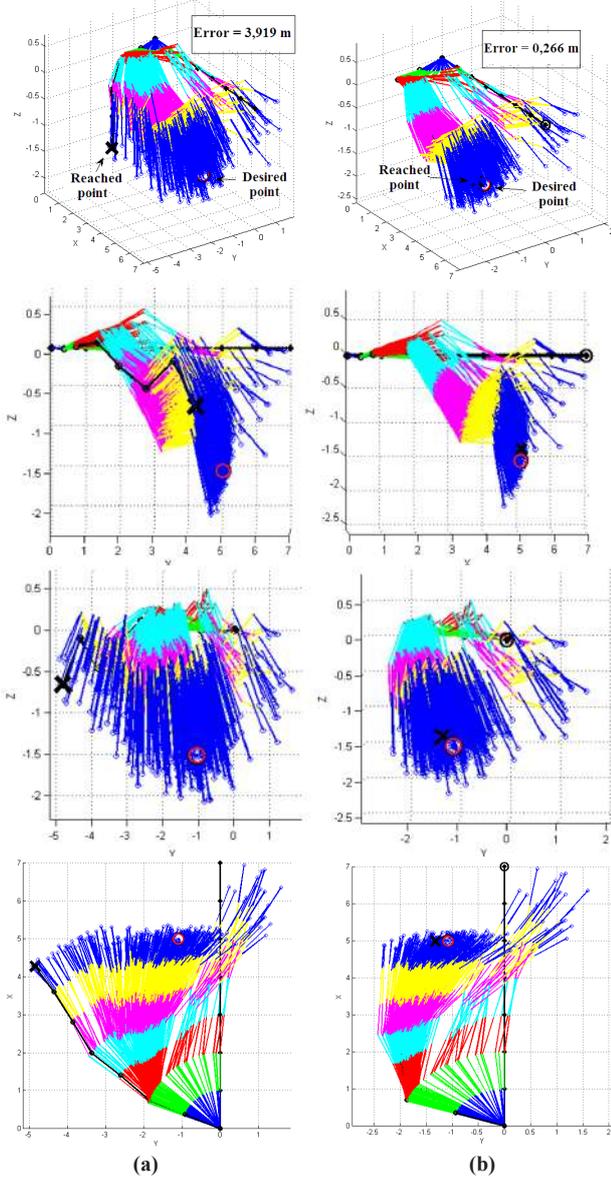


Figure 34. The position of the manipulator in space when operating ICS based on soft (a) and quantum computing (b)

In the considered unexpected situations with the introduction of strong interference into the MS, the ICS based on quantum KBO improves quality than when using only soft computing.

Thus, we considered the method of organizing coordination control using quantum computing technologies on the example of a complex CO a 7DOF robot manipulator.

We have demonstrated at the simulation model (using the MatLab/Simulink package) that the inclusion of generalize QFI unit into the ICS based on soft computing allowed:

- (1) to provide a complete solution to the problem of positioning in standard situations;
- (2) to improve positioning accuracy in conditions of

external unexpected control situations;

(3) to improve the accuracy of the positioning of the functional device in conditions of strong internal interference;

(4) to improve the Performance criterion;

(5) to reduce relative link positioning error.

Next, we consider the application of QFI in the control system of a simpler CO as 3DOF robot manipulator.

3. Control Systems for a Manipulator with 3DOF based on Quantum Computing

In this section, 3DOF manipulator control systems considered both at the simulation level and at the physical test benchmark.

3.1. Management Task

Three FCs implemented in the selected configuration of the ICS structure; each FC independently controls one of the 3links.

In Figure 35 shows a structural diagram of ICS by 3DOF manipulator based on KBO on soft computing with separated control.

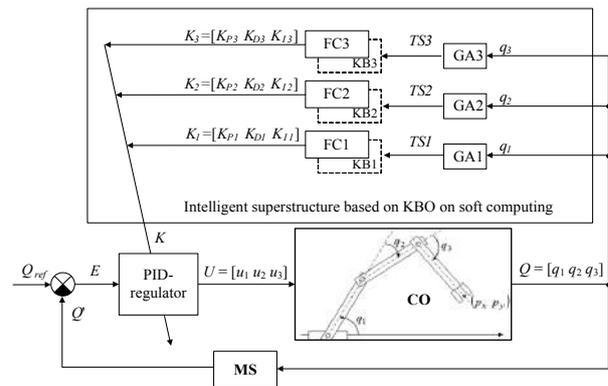


Figure 35. ICS with separated control based on KBO on soft computing for 3DOF manipulator

In Figure 35 K is the matrix of proportional, differential and integral coefficients of the PID controller $K_{p_i}, K_{D_i}, K_{I_i}, i = \overline{1,3}$, i is the number of the corresponding link of the robot of the manipulator, TS_i is the training signal, GA_i is a GA that generates a teaching signal for the formation of the i -th KB.

As a standard control situation for the FC $_i$, a typical control situation appears in the conditions of which a teaching signal TS_i received. Unexpected control situations divided into external and internal. Designed ICS based on KBO on soft computing with separated divided control (Figure 35) contain information about three situations of control (standard or unexpected) for each of three

links.

In the designed ICS based on KBO on soft computing with separated control FC1 contains information about the standard situation 1 (KB1) for link 1, FC2 contains information about the standard situation 1 for link 2 (KB2), and FC3 contains information about the standard situation 1 for link 3 (KB3).

The scheme for extracting hidden information about the relationships between existing FCs designed using soft computing technologies for three manipulator links with KBs obtained for standard control situations using QFI unit shown in Figure 36.

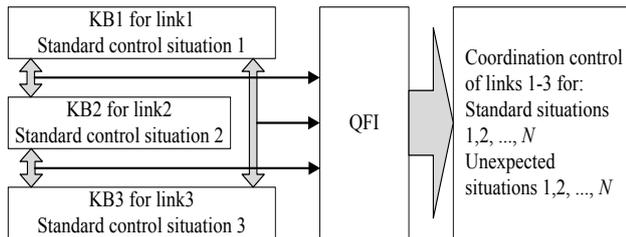


Figure 36. Methodology for extracting hidden information of the relationships of the KB

We will add the QFI block to ICS based on KBO on soft computing, which can realize the self-organization of the KBs.

3.2. Modeling and Experiment: Control Quality

The performance of the considered ICSs evaluated based on the results of MatLab / Simulink modeling and the results of a series of experiments on CO Testbench.

Let us compare the work of ICS based on KBO on quantum computing using spatial, spatio-temporal and temporal correlations and ICS based on soft computing with separated control.

In Figures 37 and 38, a comparison of ICS is given for MatLab/Simulink models and manipulator Testbench according to the introduced system of quality criteria.

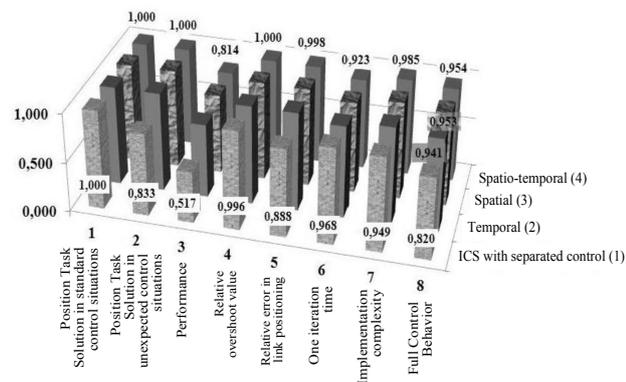


Figure 37. Comparison of the results of ICS based on KBO on soft computing with separate control (1) and ICS

based on quantum computing using temporal (2), spatial (3) and spatio-temporal (4) correlations when testing MatLab/Simulink models

From the comparison results, we conclude that the addition of the QFI unit to ICS with separated control the control task is solved for both standard and unexpected control situations; the performance increases; the accuracy of link positioning improves; and the implementation complexity of control decreases. The complexity of the implementation control depends on the dynamics of the control signal; we consider the robustness of the generated control laws below.

Because of QFI block apply in ICS, all quality criteria are improved as a result of eliminating the mismatch of the work of separated independent KBs by organizing coordination control. Moreover, if for MatLab/Simulink models the best indicator of full control behavior provided when using spatio-temporal correlation, and then manipulator Testbench determines the optimal choice of spatial correlation.

Next, we will consider the ICS based on KBO on quantum computing only with the use of spatial correlation.

In Figure 38 compares the operation of ICS based on KBO on soft computing with separated control in the conditions of an external unexpected control situation in comparison with the ICS based on quantum computing.

The forced displacement of the second link is an unexpected situation in this case.

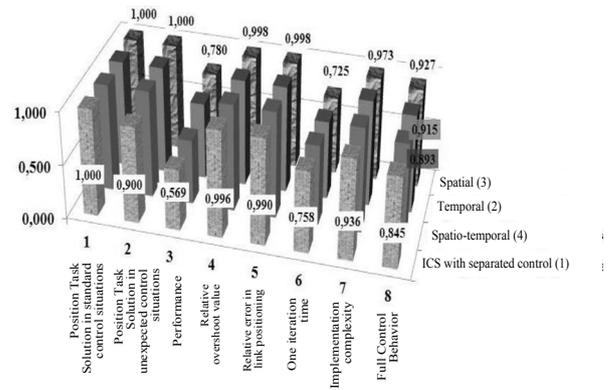


Figure 38. Comparison of the results of ICS based on KBO on soft computing with separate control (1) and ICS based on quantum computing using temporal (2), spatial (3) and spatio-temporal (4) correlations when testing manipulator Testbench

From Figure 39 it can be seen that in the considered unexpected control situation, ICS based on KBO on quantum computing decides with the positioning problem with a given accuracy, in contrast to ICS based on KBO on soft computing with separated control.

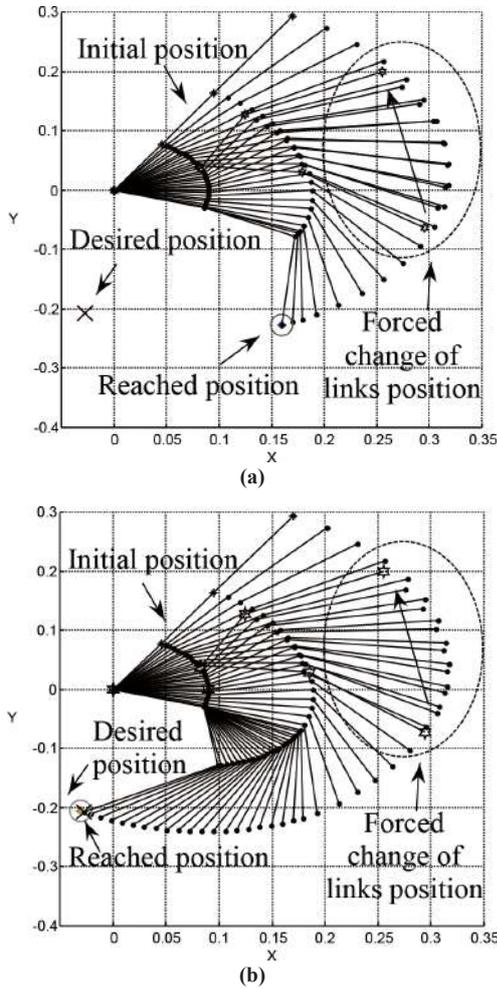


Figure 39. ICS based on KBO on soft computing with separated control in unexpected control situation (a); ICS based on KBO on quantum computing (b)

The inability of ICS based on soft computing to solve the problem of exact position control in Figure 40 also illustrated. FC responsible for managing the second link for the allotted time was not able to rehabilitate after a powerful external impact. Because of which the positioning error of the second link was more than 50 degrees, the control goal has not achieved and the control system as a whole was not robust.

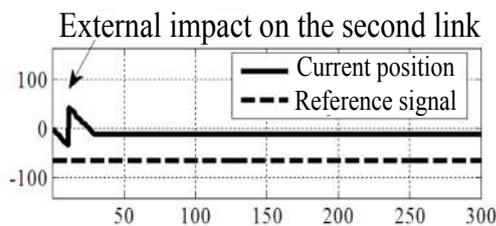


Figure 40. The change in the position of the second link under the control of ICS based on KBO on soft computing with separated control

Consider internal unexpected control situations. Let us compare ICS based on KBO on soft computing with one FC and ICS based on KBO on quantum computing in the conditions of changing restrictions of the control channel.

In Figures 41 and 42 ICS is compared respectively for MatLab / Simulink models and for manipulator Testbench, taking into account internal and external unexpected control situations.

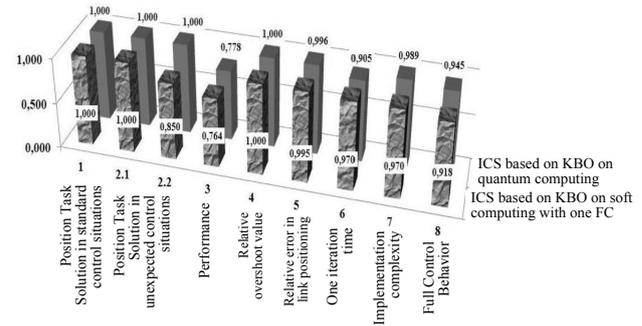


Figure 41. Comparison of the results of ICS based on KBO on soft computing with one FC and ICS based on quantum computing when testing Matlab / Simulink models

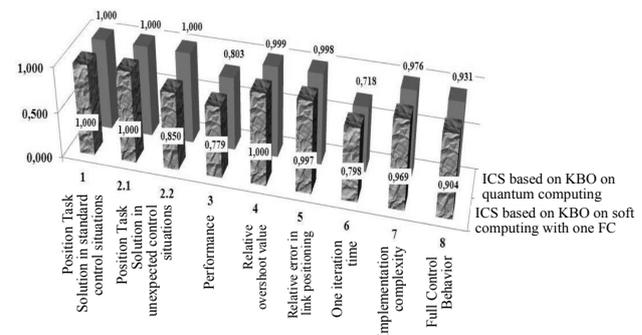


Figure 42. Comparison of the results of ICS based on KBO on soft computing with one FC and ICS based on quantum computing when testing manipulator Testbench

It can see from the comparison results that ICS based on KBO on soft computing and ICS based on quantum computing solves the positioning problems for standard and external unexpected control situations. ICS based on soft computing with one FC not always coping with internal unexpected control situations. Full Control Behavior for ICS on quantum computing is higher, both for MatLab/Simulink models and for the manipulator Testbench.

Let us demonstrate the operation of ICS based on KBO on soft computing with one FC in the conditions of an internal unexpected control situation (Figure 43) in comparison with ICS based on KBO on quantum computing.

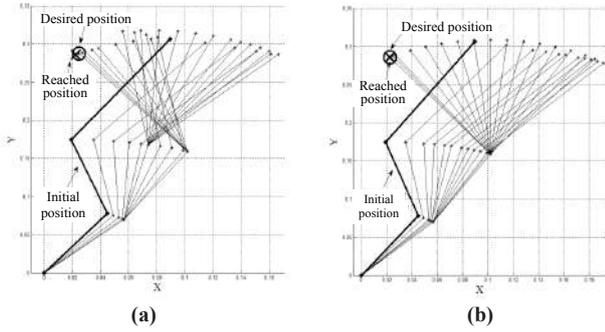


Figure 43. ICS based on KBO on soft computing with one FC in the conditions of an internal unexpected control situation (a); ICS based on quantum computing (b)

From Figure 43 it can be seen that ICS based on KBO on quantum computing provides a better solution quality than ICS based on soft computing with one FC.

Consider the control signals generated by the considered types of ICS based on soft and quantum computing.

In Figure 44 *QFC* are the control signals of ICS based on KBO on quantum computing, *FC* are the control signals generated by ICS on soft computing with one FC, *FC Decomposition* are the control signals generated by ICS on soft computing with separated control.

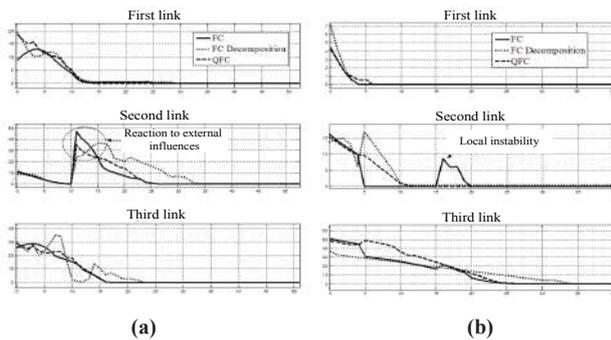


Figure 44. Control signals generated by different types of ICS in unexpected control situations

From Figure 44 (a) it can be seen that the control signals generated by ICS based on KBO on quantum computing under external disturbing influences are similar to those generated by ICS on soft computing with one FC. However, the amplitude of the control signal at the time of external influence is significantly lower for ICS based on quantum computing. The control signal formed by ICS based on soft computing with separated control has become overshoot.

From Figure 44 (b) it can be seen that the control signal generated by ICS based on KBO on soft computing with one FC for the second link has a local instability section, while ICS based on quantum computing generates robust control signals under conditions of external disturbing influences.

Thus, the minimum consumption of useful resource in the formation of control signals ensured when using ICS based on KBO on quantum computing.

4. Conclusion

Using two COs of varying complexity (3DOF and 7DOF manipulators) as an example, the advantages and limitations of using soft and quantum computing technologies in ICS were demonstrated. On the example of 3DOF manipulator, the minimal difference between the results of the physical Manipulator Testbench and the MatLab/Simulink model demonstrated.

Figure 45 shows a generalized comparison of ICS based on KBO on soft and quantum computing for 3DOF and 7DOF manipulators for standard and unexpected control situations of the examples considered in this article (Figures 11, 13, 19, 24, 39, 40).

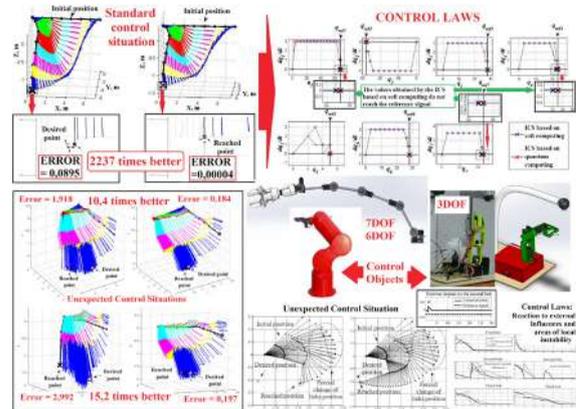


Figure 45. Comparison of ICS based on soft and quantum computing for 3DOF and 7DOF

Further research focused on the development and analysis of the physical Testbench of Manipulator with 7 DOF, as well as its integration with the mobile platform.

Appendix 1. Requirements to QFI-model Design and Its Features in Quantum Algorithm Control of Knowledge Base Self-organization

A1.1 Main Proposals and Features of QFI Model

Main proposals and features of the developed swarm QFI-model in the solution of intelligent control problems are as following:

A. Main proposals

(1) The digital value's set of control signals produced by responses of FC outputs are considered as swarm particles along of classical control trajectories with individual marked intelligent agents;

(2) Communication between particle swarm trajectories through a quantum link is introduced;

(3) Intelligent agents are used different types of quantum correlations (as behavior control coordinator with quantum computation by communication) and information transport (value information);

(4) The (hidden) quantum value information extracted from classical states of control signal classical trajectories (with minimum entropy in “intelligent states” of designed robust control signals).

B. Features

(1) Developed QFI model is based on *thermodynamic* and *information-theoretic* measures of intelligent agent interactions in communication space between macro- and micro-levels (the entanglement-assisted correlations in an active system represented by a collection of intelligent agents);

(2) From computer science viewpoint, QA of QFI model plays the role of the information-algorithmic and SW-platform support for design of self-organization process;

(3) Physically, QFI supports optimally a new developed *thermodynamic trade-off* of control performance (between stability, controllability and robustness) in self-organization KB process.

From quantum information theory viewpoint, QFI reduces the redundant information in classical control signals using four facts^[4] from quantum information for data compression in quantum information processing: 1) efficient quantum data compression; 2) coupling (separation) of information in the quantum state in the form of classical and quantum components; 3) amount of total, classical, and quantum correlation; and 4) hidden (observable) classical correlation in the quantum state.

The gate structure of QFI model with self-organization KB properties that includes all of these QA features (see, below Figure A1) based on above-mentioned proposals.

Let us discuss the following question.

Q. *What is a difference between our approach and Natural (or man-made) models of self-organization?*

A. *Main differences and features* are as followings:

(1) In our approach a self-organization process is described as a *logical algorithmic* process of value information *extraction* from hidden layers (*possibilities*) in classical control laws using quantum decision-making logic of QFI-models based on main facts of quantum information, quantum computing and QA’s theories;

(2) Structure of QFI includes all of natural elements of self-organization (templating, self-assembly, and self-organization structure) with corresponding quantum operators (superposition of initial states, selection of quantum correlation types and classes, quantum oracles, interference, and measurements);

(3) QFI is a new quantum search algorithm (belonging

to so called *QPB*-class) that can solve classical algorithmically unsolved problems;

(4) In QFI the self-organization principle is realized using the on-line responses in a dynamic behavior of classical FC’s on new control errors in unpredicted control situations for the design of robust intelligent control;

(5) Model of QFI supports the thermodynamic interrelations between *stability*, *controllability* and *robustness* for design of self-organization processes.

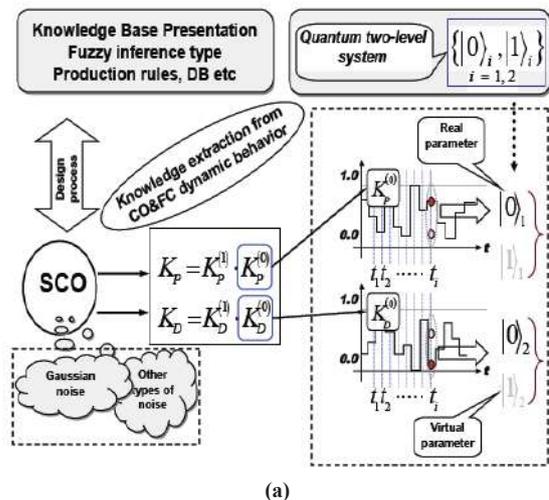
Specific features of QFI applications in design of robust KB. Let us stress the fundamentally important specific feature of operation of the QA (in the QFI model) in the design process of robust laws for the coefficient gain schedules of fuzzy PID controllers based on the individual KB that designed on Soft Computing Optimizer (SCO) with soft computing.

A1.2 Quantum Information Resources in QFI Algorithm

In this section we introduce briefly the particularities of quantum computing and quantum information theory that are used in the quantum block, QFI supporting a self-organizing capability of FC in robust ICS. According to described below algorithm the input to the QFI gate is considered as a superposed quantum state $K_1(t) \otimes K_2(t)$, where $K_{1,2}(t)$ are the outputs from fuzzy controllers FC1 and FC2 designed by SCO (see, below Figure A1) for the given control task in different control situations (for example, in the presence of different stochastic noises).

A1.2.1 Quantum Hidden Information Extraction in QFI

Using the four facts from quantum information theory QFI extracts the hidden quantum value information from classical KB1 and KB2 (see Figure A1).



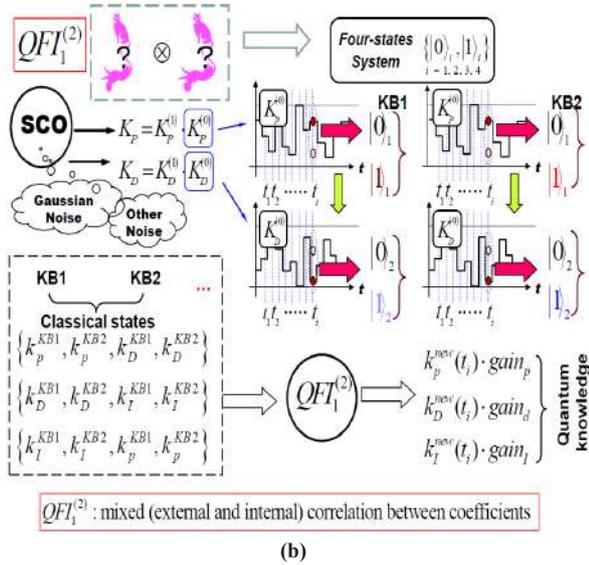


Figure A1. Example of information extraction in QFI

In this case between KB1 and KB2 (from quantum information theory of viewpoint) we organize a communication channel using quantum correlations that is impossible in classical communication theory. The algorithm of superposition calculation is presented below and described in details in [4].

We discuss for simplicity the situation in which an arbitrary amount of correlation is unlocked with a one-way message.

Example. Let us consider the communication process between two KBs as communication between two players *A* and *B* (see Figure A1) and let $d = 2^n$. According to the law of quantum mechanics, initially we must prepare a quantum state description by density matrix ρ from two classical states (KB1 and KB2). The initial state ρ is shared between subsystems held by *A* (KB1) and *B* (KB2), with respective dimensions d ,

$$\rho = \frac{1}{2d} \sum_{k=0}^{d-1} \sum_{t=0}^1 (|k\rangle\langle k| \otimes |t\rangle\langle t|)_A \otimes (U_t |k\rangle\langle k| U_t^\dagger)_B. \quad (A1)$$

Here $U_0=I$ and U_1 changes the computational basis to a conjugate basis $\langle i|U_1|k\rangle = 1/\sqrt{d} \forall i, k$.

In this case, *B* chooses $|k\rangle$ randomly from d states in two possible random bases, while *A* has complete knowledge on his state. The state (A1) can arise from following scenario. *A* picks a random n -bit string k and sends $B|k\rangle$ or $H^{\otimes n}|k\rangle$ depending on whether the random bit $t = 0$ or 1 . Player *A* can send t to player *B* to unlock the correlation later. Experimentally, Hadamard transform, H and measurement on single qubits are sufficient to prepare

the state (A1), and later extract the unlocked correlation in ρ' . The initial correlation is small, i.e. $I_{Cl}^{(l)}(\rho) = \frac{1}{2} \log d$.

The final amount of information after the complete measurement M_A in one-way communication is ad hoc, $I_{Cl}(\rho') = I_{Cl}^{(l)}(\rho) = \log d + 1$, i.e., the amount of *accessible information increase*. This phenomenon is *impossible classically*. However, states exhibiting this behaviour *need not be entangled* and corresponding communication can be organized using Hadamard transform.

Therefore, using the Hadamard transformation and a new type of quantum correlation as the communication between a few KB's it is possible to increase initial information by unconventional quantum correlation (as the quantum cognitive process of a value hidden information extraction in on-line, see, e.g. Figure A1, b). In present section we consider a simplified case of QFI when with the Hadamard transform is organized an unlocked correlation in superposition of two KB's; instead of the difficult defined entanglement operation an equivalent quantum oracle is modelled that can estimates an "intelligent state" with the maximum of amplitude probability in corresponding superposition of classical states (minimum entropy) principle relative to extracted quantum knowledge [5]. Interference operator extracts this maximum of amplitude probability with a classical measurement.

Figure A2 shows the algorithm for coding, searching and extracting the value information from KB's of fuzzy PID controllers designed by SCO and QCO (quantum computing optimizer).

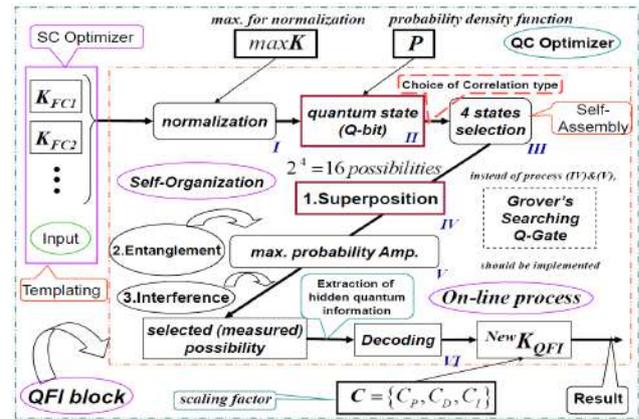


Figure A2. The structure of QFI gate

Optimal drawing process of value information from a few KBs that are designed by soft computing is based on following four facts from quantum information theory [4]: (i) the effective quantum data compression; (ii) the splitting of classical and quantum parts of information in quantum state; (iii) the total correlations in quantum state

are “mixture” of classical and quantum correlations; and (iv) the exiting of hidden (locking) classical correlation in quantum state.

This quantum control algorithm uses these four Facts from quantum information theory in following way: (1) compression of classical information by coding in computational basis $\{|0\rangle, |1\rangle\}$ and forming the quantum correlation between different computational bases (Fact 1); (2) separating and splitting total information and correlations on “classical” and “quantum” parts using Hadamard transform (Facts 2 and 3); (3) extract unlocking information and residual redundant information by measuring the classical correlation in quantum state (Fact 4) using criteria of maximal corresponding amplitude probability. These facts are the informational resources of QFI background. Using these facts, it is possible to extract an additional amount of quantum value information from smart KBs produced by SCO for design a *wise* control using compression and rejection procedures of the redundant information in a classical control signal.

Below we discuss the application of this quantum control algorithm in QFI structure.

A1.3 Structures of Robust ICS and Information Design Technology of Quantum KB Self-organization

The kernel of the above-mentioned FC design toolkit is a so-called SCO implementing advanced soft computing ideas. SCO is considered as a new flexible tool for design of optimal structure and robust KBs of FC based on a chain of genetic algorithms (GAs) with information-thermodynamic criteria for KB optimization and advanced error BP-algorithm for KB refinement. Input to SCO can be some measured or simulated data (called as TS) about the modelling system. For TS design (or for GA fitness evaluation) we use stochastic simulation system based on the control object model.

Figure A3 illustrates as an example the structure and main ideas of self-organized control system consisting of two FC’s coupling in one QFI chain that supplies a self-organizing capability. CO may be represented in physical form or in the form of mathematical model. We will use a mathematical model of CO described in Matlab / Simulink. The kernel of the above-mentioned FC design tools is a so-called SCO implementing advanced soft computing ideas.

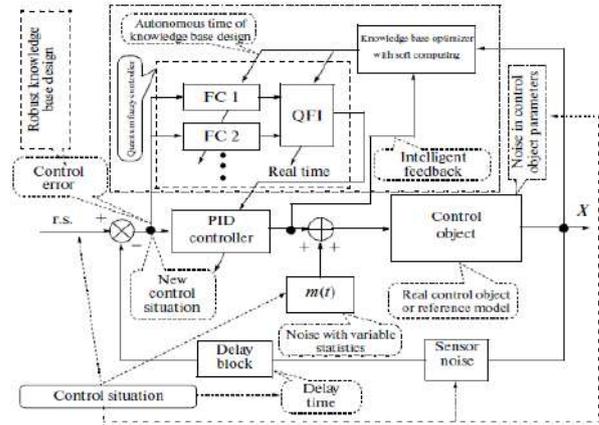


Figure A3. Structure of robust ICS based on QFI

Robustness of control signal is the background for support the reliability of control accuracy in uncertainty environments. The effectiveness of the developed QFI model is illustrated for important case, the application to design of robust control system in unpredicted control situations.

The main technical purpose of QFI is to supply a self-organization capability for many (sometimes unpredicted) control situations based on a few KBs. QFI produces a robust optimal control signal for the current control situation using a reducing procedure and compression of redundant information in KB’s of individual FCs. Process of rejection and compression of redundant information in KB’s uses the laws of quantum information theory. Decreasing of redundant information in KB-FC increases the robustness of control without loss of important control quality as reliability of control accuracy. As a result, a few KB-FC with QFI can be adapted to unexpected change of external environments and to uncertainty in initial information.

Let us discuss in detail the design process of robust KB in unpredicted situations.

A1.3.1 KB self-organization Quantum Algorithm of FC’s Based on QFI

We use real value of a current input control signal to design normalized state $|0\rangle$. To define probability amplitude a_0 we will use simulation results of controlled object behavior in teaching conditions. In this case by using control signal values, we can construct histograms of control signals and then taking integral we can receive probability distribution function and calculate $\alpha_0 = \sqrt{P_0}$. Then we can find $\alpha_1 = \sqrt{1 - P_0}$. After that it is possible to define state $|1\rangle$ as shown on Figure A4 below^[13].

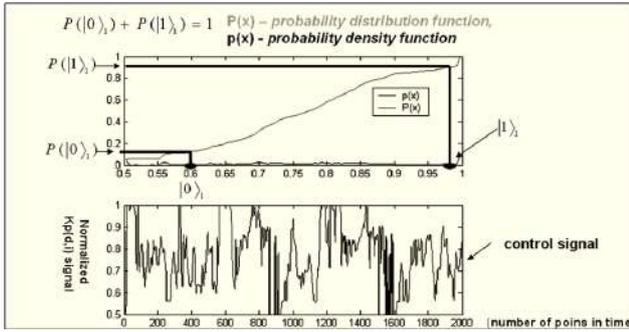


Figure A4. Example of control signal and corresponding probability distribution function

For QA design of QFI it is needed to apply the additional operations to partial KBs outputs that drawing and aggregate the value information from different KBs. Soft computing tool does not contain corresponding necessary operations. The necessary unitary reversible operations are called *superposition*, *entanglement* (quantum correlation) and *interference* that physically are operators of quantum computing.

Consider main steps of developed QFI process that is considered as a QA.

Step 1: Coding

(1) Preparation of all normalized states $|0\rangle$ and $|1\rangle$ for current values of control signal $\{k_p^i(t), k_D^i(t), k_I^i(t)\}$ (index i means a number of KB) with respect to the chosen knowledge bases and corresponding probability distributions, including:

- (2) calculation of probability amplitudes a_0, a_1 of states $|0\rangle$ and $|1\rangle$ from histograms;
- (3) by using a_1 calculation of normalized value of state $|1\rangle$.

Step 2: Choose quantum correlation type for preparation of entangled state. In the Table A1 investigated types of quantum correlations are shown. Take, for example, the following quantum correlation type:

$$\{k_p^1(t), k_D^1(t), k_p^2(t), k_D^2(t)\} \rightarrow k_p^{new}(t), \text{ where 1 and 2 are indexes of KB.}$$

Then a quantum state $|a_1 a_2 a_3 a_4\rangle = |k_p^1(t) k_D^1(t) k_p^2(t) k_D^2(t)\rangle$ is considered as correlated (entangled) state.

Table A1. Types of quantum correlations

1. QFI based on spatial correlations	$k_p^{KB_1, KB_2}(t) k_D^{KB_1, KB_2}(t) \rightarrow k_p^{new}(t) \cdot gain_p$ $k_D^{KB_1, KB_2}(t) k_I^{KB_1, KB_2}(t) \rightarrow k_D^{new}(t) \cdot gain_D$ $k_I^{KB_1, KB_2}(t) k_p^{KB_1, KB_2}(t) \rightarrow k_I^{new}(t) \cdot gain_I$
---	---

2. QFI based on temporal correlations	$k_p^{KB_1, KB_2}(t_i) k_p^{KB_1, KB_2}(t_i - \Delta t) \rightarrow k_p^{new}(t_i) \cdot gain_p$ $k_D^{KB_1, KB_2}(t_i) k_D^{KB_1, KB_2}(t_i - \Delta t) \rightarrow k_D^{new}(t_i) \cdot gain_D$ $k_I^{KB_1, KB_2}(t_i) k_I^{KB_1, KB_2}(t_i - \Delta t) \rightarrow k_I^{new}(t_i) \cdot gain_I$
3. QFI based on spatio-temporal correlations	$k_p^{KB_1}(t_i) k_D^{KB_1}(t_i - \Delta t) k_p^{KB_2}(t_i - \Delta t) k_D^{KB_2}(t_i) \rightarrow k_p^{new}(t_i) \cdot gain_p$ $k_D^{KB_1}(t_i) k_I^{KB_1}(t_i - \Delta t) k_D^{KB_2}(t_i - \Delta t) k_I^{KB_2}(t_i) \rightarrow k_D^{new}(t_i) \cdot gain_D$ $k_I^{KB_1}(t_i) k_p^{KB_1}(t_i - \Delta t) k_I^{KB_2}(t_i - \Delta t) k_p^{KB_2}(t_i) \rightarrow k_I^{new}(t_i) \cdot gain_I$

Step 3: Superposition and entanglement. According to the chosen quantum correlation type construct superposition of entangled states as shown on general Figure A5, where H is the Hadamard transform operator.

Step 4: Denormalization

Calculate final (denormalized) output result as follows:

$$k_p^{output} = k_p^{new}(t) \cdot gain_p, k_D^{output} = k_D^{new}(t) \cdot gain_D, k_I^{output} = k_I^{new}(t) \cdot gain_I.$$

Step 4a: Find robust QFI scaling gains $\{gain_p, gain_D, gain_I\}$ based on GA and a chosen fitness function.

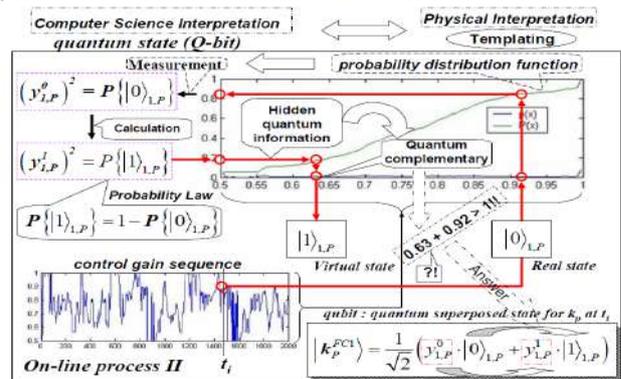
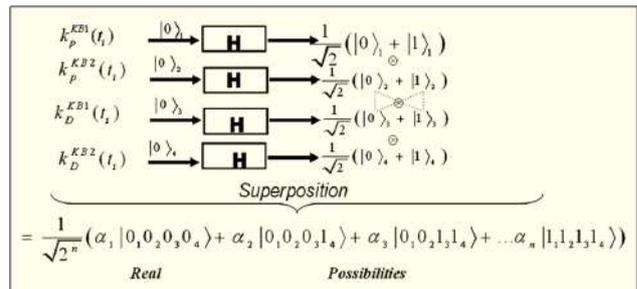


Figure A5. The algorithm of superposition calculation

In proposed QFI we investigated the proposed types of quantum QFI correlations shown in Table A1 where the correlations are given with 2KB, but in general case a few of KBs may be; t_i is a current temporal point and Δt is a correlation parameter.

Let us discuss the particularities of quantum computing that are used in the quantum block QFI (Figure A2) supporting a self-organizing capability of a fuzzy controller. Optimal drawing process of value information from a few of KBs as above-mentioned is based on the following four facts from quantum information theory:

- (1) the effective quantum data compression (*Fact1*);

(2) the splitting of classical and quantum parts of information in quantum state (*Fact 2*);

(3) the total correlations in quantum state are “mixture” of classical and quantum correlations (*Fact 3*);

(4) existing of hidden (locking) classical correlation in quantum state using criteria of maximal corresponding probability amplitude (*Fact 4*).

These facts are the informational resources of QFI background. Using these facts, it is possible to extract the value information from KB1 and KB2. In this case between KB1 and KB2 (from quantum information theory point of view) we organize a communication channel using quantum correlations that is impossible in classical communication. In QFI algorithm with the Hadamard transform an unlocked correlation in superposition of states is organized. The entanglement operation is modelled as a quantum oracle that can estimate a maximum of amplitude probability in corresponding *superposition of entangled states*. Interference operator extracts this maximum of amplitudes probability with a classical measurement. Thus, from two FC-KBs (produced by SCO for design a *smart* control) we can produce a *wise* control by using compression and rejection procedures of the redundant information in a classical control signal. This completes the particularities of quantum computing and quantum information theory that are used in the quantum block supporting a self-organizing capability of FC.

A1.4 Robust FC Design Toolkit: SC and QC Optimizers for Quantum Controller’s Design

To realize QFI process we developed new tools called “QC Optimizer” that are the next generation of SCO tools.

A1.4.1 QC Optimizer Toolkit

QC Optimizer Toolkit is based on Quantum & Soft Computing and includes the following:

(1) Soft computing and stochastic fuzzy simulation with information-thermodynamic criteria for robust KBs design in the case of a few teaching control situations;

(2) QFI-Model and its application to a self-organization process based on two or more KBs for robust control in the case of unpredicted control situations.

Internal structure of QC Optimizer is shown on Figures A6 and A7.

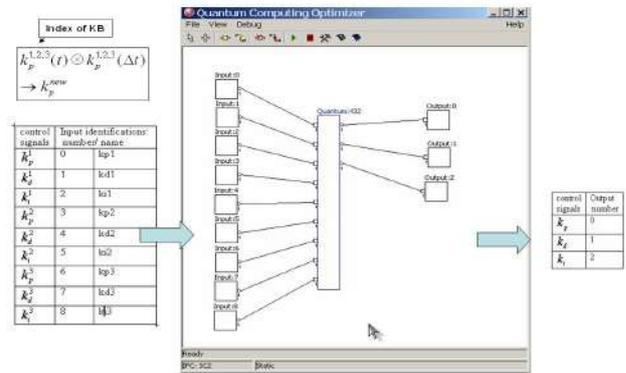


Figure A6. First internal layer of QC Optimizer

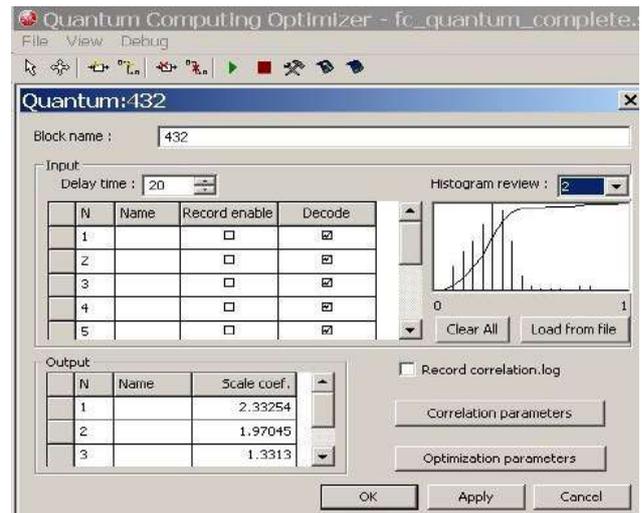


Figure A7. Second internal layer of QC Optimizer

Remark. On Figure A6, the first internal layer of QC Optimizer is shown (inputs/output). On Fig/ A7, the quantum block realizing QFI process based on three KB is described. The “delay time = 20 (sec)” on the Figure A7, corresponds to the parameter “ Δt ” given in temporal quantum correlations description (see Table A1); the knob named “correlation parameters” call other block (see Figure A7) where a chosen type of quantum correlations (Table A1) is described.

On Figure A8 description of temporal quantum correlations is shown. Here “kp1_r” means state $|0\rangle$ for $k_p(t)$ of FC1 (or KB1); “kp1_r_t” means state $|0\rangle$ for $k_p(t+\Delta t)$ of FC1 (or KB1); “kp1_v” means state $|1\rangle$ for $k_p(t)$ of FC1 (or KB1); “kp1_v_t” means state $|1\rangle$ for $k_p(t+\Delta t)$ of FC1 (or KB1); and so on for other FC2 (KB2) and FC3(KB3).

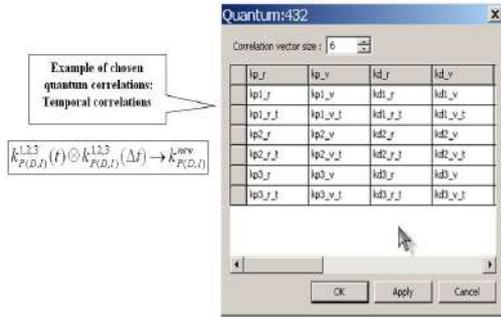


Figure A8. Internal structure of “correlation parameters” block

A1.4.2 Design of Intelligent Robust Control Systems for Complex Dynamic Systems Capable to Work in Unpredicted Control Situations

Describe now key points of Quantum & Soft Computing Application in Control Engineering as follows:

- (1) PID Gain coefficient schedule (control law) is described in the form of KB of a Fuzzy Inference System (realized in FC);
- (2) GA with complicated Fitness Function is used for KB-FC forming;
- (3) KB-FC tuning is based on Fuzzy Neural Networks using error BP-algorithm;
- (4) Optimization of KB-FC is based on *SC optimizer tools* (Step 1 technology);
- (5) Quantum control algorithm of self-organization is developed based on the QFI-model;
- (6) QFI-model realized for the KB self-organization to a new unpredicted KB control situation is based on *QC optimizer tools* (Step 2 technology).

In this Appendix 1 we are introduced briefly the particularities of quantum computing and quantum information theory that are used in the quantum block QFI (Figure A9) supporting a self-organizing capability of FC in robust ICS.

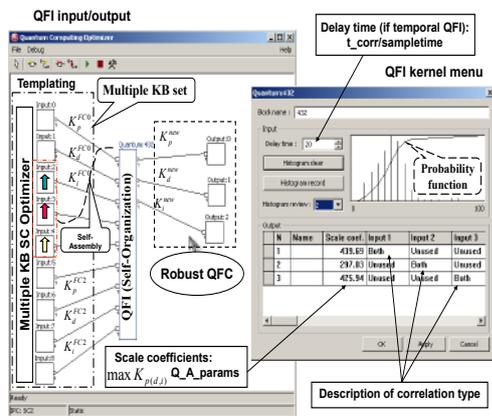


Figure A9. QFI-process by using QC Optimizer (QFI kernel)

Using unconventional computational intelligence toolkit, we propose a solution of such kind of generalization problems by introducing a self-organization design process of robust KB-FC that supported by QFI based on Quantum Soft Computing ideas.

The main technical purpose of QFI is to supply a self-organization capability for many (sometimes unpredicted) control situations based on a few KBs. QFI produces robust optimal control signal for the current control situation using a reducing procedure and compression of redundant information in KB’s of individual FCs. Process of rejection and compression of redundant information in KB’s uses the laws of quantum information theory. Decreasing of redundant information in KB-FC increases the robustness of control without loss of important control quality as reliability of control accuracy. As a result, a few KB-FC with QFI can be adapted to unexpected change of external environments and to uncertainty in initial information.

At the second stage of design with application of the QFI model, we do not need yet to form new production rules. It is sufficient only to receive in on-line the response of production rules in the employed FC to the current unpredicted control situation in the form of the output control signals of the coefficient gains schedule in the fuzzy PID controller. In this case, to provide the operation of the QFI model, the knowledge of particular production rules fired in the KB is not required, which gives a big advantage, which is expressed the form of an opportunity of designing control processes with the required robustness level in on-line.

Note that the achievement of the required robustness level in an unpredicted control situation essentially depends in a number of cases on the quality and quantity of the employed individual KB’s.

Thus, the QA in the QFI model is a physical prototype of production rules, implements a virtual robust KB for a fuzzy PID controller in a program way (for the current unpredicted control situation), and is a problem-independent toolkit. The presented facts give an opportunity to use experimental data of the teaching signal without designing a mathematical model of the CO. This approach offers the challenge of QFI using in problems of CO with weakly formalized (ill-defined) structure and a large dimension of the phase space of controlled parameters.

Appendix 2: Quantum Computing Optimizer of Knowledge Base (QCOptKB™)

Quantum Computing Optimizer of KB (QCOptimizer) is a software toolkit designed for modeling, creation and optimization of complex control structures, including fuzzy and quantum control modules. Working with

QCOptimizer one can:

- (1) create a model of complex control system, consisting from different blocks connected at any fashion;
- (2) analyze performance of modeled control system and optimize it using different teaching signals.

A2.1 Learning Signal for QCOptimizer

In order to perform different optimization algorithms QCOptimizer requires *learning signal*, which presents samples of input values and corresponding output values. QCOptimizer is able to read signal data from Matlab v.4 and v.5 files and from text files.

Text files are processed based on *locale data*, which defines symbols for decimal point, thousands of separators and so on. By default, QCOptimizer uses windows settings for these parameters. If those settings do not match signal file format they can be changed at any moment. Once changed, locale parameters are saved in model and will be used for future processing of data. Locale setting affects reading and writing of text data files and model files.

Using Quantum Computing Optimizer.

A2.2 Starting with Quantum Computing Optimizer

Main module of Quantum Computing Optimizer is a Win32 application, called qcwin.exe. It can be used in Windows 95, 98, NT, Me, 2000 and XP environments. In order to start the program, you should type “QCOptimizer” in command prompt or double-click QCOptimizer icon in windows explorer. During process of model optimization QCOptimizer creates temporary files in current directory. Those files have “mms” extensions. Before starting QCOptimizer check that current directory does not contain files with same extension.

A2.2.1 SCOptimizer Program Window

Main program window of QCOptimizer is shown on Figure A10.

QCOptimizer window is mostly occupied with model display. It displays current control system model as a set of connected blocks. Each block corresponds to some data-processing module in control system. Lines show how inputs and outputs of modules are connected. Model display allows access to each module configuration and allows manipulation of modules and connections.

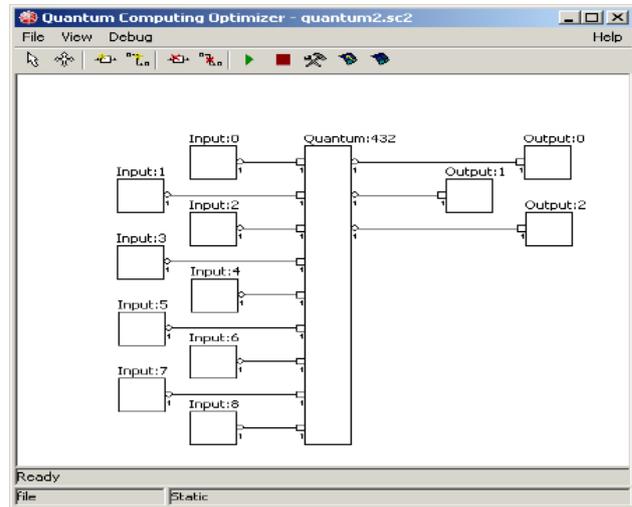


Figure A10. QCOptimizer program window

A2.2.2 QCOptimizer Toolbar

QCOptimizer toolbar is located at the upper part of the main window. Buttons on the toolbar provide access for model-editing and optimization functions. They will be described in detail in next sections.

A2.2.3 QCOptimizer Menu

QCOptimizer also has standard window menu, displayed on the top of program window. To choose a command with the mouse or keyboard, first select the menu and then choose command you want. Each underlined character in menu and command names corresponds to the key you can press to select a menu or to choose a command.

To choose a command by using the mouse you should first click the menu name containing the command you want and then click the command name in the pop-up menu.

To choose a command by using the keyboard first press the *ALT* key. The File menu appears selected. Use left and right arrow keys to select required menu and then press down arrow or *ENTER* to open pop-up menu. Use up and down arrows to select desired command and press *ENTER* to activate it. You can also press one of the underlined characters to quickly select desired menu or command.

A2.2.3.1 FILE Menu

QCOptimizer allows project files to be managed by the means of commands in the *FILE* Menu.

The following items are available in this menu:

New: closes actual model and starts creation of new model.

Open: open existing model from disk file.

Import: import SCOptimizer file to the current model.

Save: save actual model to the file.

Close: close current model without saving.

Signal Manager: activate learning signal manager.

Number Format: set number format conventions for teaching signal and model files.

Exit: closes QCOptimizer.

A2.2.3.2 VIEW Menu

View menu can be used to open signal view windows:

Open signal view: displays new signal window.

A2.2.3.3 HELP Menu

Help menu currently have only one command *About*, which displays version information of QCOptimizer.

A2.2.4 Dialog Boxes

When you choose a command having options, QCOptimizer shows a dialog box. A dialog box may contain fields in which you can enter text, numbers or select some items. Typical dialog box will have *OK* button, which you should press after filling all fields in order to activate the command, and *CANCEL* button, which will abort command execution. You can also use *ENTER* key on the keyboard to activate the same function as *OK* button and *ESC* as *CANCEL*.

If an operation requires many parameters Wizard-style dialog box will be used. Wizard box consists of a sequence of dialog boxes that will guide you through the steps of an operation. Three buttons will be available for navigation of the wizard box. *NEXT>>* button tells wizard that you have successfully completed filling of current page and want to switch to the next page or perform the command if no additional parameters are required. *<<BACK* button should be used to return to one of the previous pages, if you want to change some parameters. *CANCEL* button aborts command execution.

A2.2.5 QCOptimizer Files

SCOptimizer saves project in special files: *model files* (with *.sc2 extension). *Model file* is used to store all model data, including model blocks, their properties, connections and positions, as well as most optimization and other settings.

QCOptimizer can save genetic algorithm optimization state to disk file, so you can continue optimization process interrupted some time ago. Those files have *.st extension and will be called *state file*.

If you wish to copy QCOptimizer files to new location (for example to another computer) you should copy *model file*, *teaching signal* files and *state files*, if it is required. If your *teaching signal* file was originally located in directory other than one with project file, than QCOptimizer may be unable to load it automatically in new location. In this

situation use *Teaching Signal Manager* to point QCOptimizer to new location of *teaching signal* file.

A2.3 Working with QCOptimizer

A2.3.1 Creating New Model

Select *File/New* from menu to create new model. An empty model will be created and you can than add various blocks to it.

A2.3.2 Loading Model from File

If you have a file with previously saved model you should select *File/Open* command. Standard windows *File Open* dialog box will appear. Select file with you model and press *Open* button.

A2.3.3 Saving Model files

You can use *File/Save* command to save you model files. Standard windows *File Save* dialog box will appear. Select file with you model and press *Save* button.

We recommend saving you model regularly, in order to avoid loss of data due to possible program/computer failure or undesired model change. Note that most operations of QCOptimizer do not have “undo” option, so the only way to restore to previous state is to load model from saved file.

A2.3.4. Importing SCOptimizer Files

You can import SCOptimizer model files (.sco extension) in order to use fuzzy control systems developed in SCOptimizer in QCOptimizer projects. Select *File/Import* command to start model input. Standard windows *File Open* dialog box will appear. Select file with SCOptimizer model and press *Open* button.

Import options dialog will than appear as shown in Figure A11.

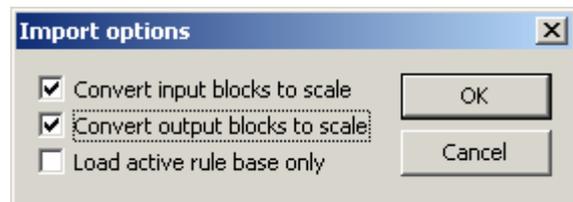


Figure A11. Import options dialog

Check option checkboxes as required and press *OK* to import model. The options are following:

Convert input blocks to scale: by default imported SCOptimizer model includes new input blocks that are used to feed input data to fuzzy control system. If this option is selected the model will be imported with scale

blocks instead, which will use input normalization parameters to scale signal. This option can be useful when creating models, which includes several fuzzy control modules using the same input.

Convert output blocks to scale: same as before, but applied to fuzzy control output.

Load active database only: allow import of only one fuzzy rule database from SCOptimizer model, the active one. By default, all databases are imported and active database is connected to output blocks.

A2.3.5 Changing Locale Settings

This operation is available from *File/Number Format* menu and from New Model creation Wizard. After selection of this operation QCOptimizer will show you dialog box, containing current settings (Figure A12).

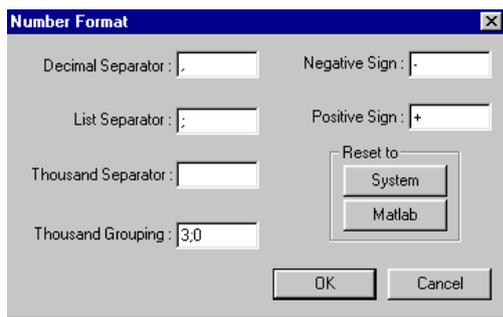


Figure A12. Number Format dialog box

Every string in this list shows one of the locale parameters. Those parameters have following meaning:

List separator

List separator is a symbol or string used to separate several successive numbers in list.

Decimal separator

Separates integer and fractional part of numbers.

Negative sign

Defines negative numbers.

Positive sign

Shows that number is positive. SCOptimizer won't print this string before positive numbers, but will accept numbers with this sign in data files.

Grouping method

Define how digits of integer part of numbers will be grouped. This is a semicolon-separated list of numbers, defining number of digits in each group, from right to left. Trailing 0 means "use previous value for all following groups". SCOptimizer will insert the thousand separators in every place defined by grouping method, but will accept thousand separators in any position in input files.

Thousand's separator

Separates groups of digits, defined by Grouping method.

You can change any parameter by entering desired symbol(s) to the corresponding field. There are also two buttons, which you can use to switch to one of the predefined configurations:

(1) *System:* switch to system defaults, as defined by Windows locale settings.

(2) *Matlab:* switch to format, used by Matlab for text files. Following settings are default for the Matlab:

① *List separator:* “;”

② *Decimal separator:* “.”

③ *Negative sign:* “-”

④ *Positive sign:* “+”

⑤ *Grouping method:* “3;”

⑥ *Thousand's separator:* “ ”

When entering locale parameters be careful not to:

(3) set different parameters to the same symbol

(4) make list separator, decimal separator or negative sign an empty string

(5) do not set any parameter to one of the following symbols: (,),{,},;,',?.

Breaking these rules may cause program not to save and/or load files correctly.

After you set all options to desired characters press OK to apply changes or *CANCEL* if you want to reset you changes.

A2.3.6 Working with Model

You can edit model using model view in the main QCOptimizer window. It displays model as a set of processing modules, called blocks, and connections between them. You can add new blocks, delete existing, create and destroy links between blocks and access block properties.

Each block is displayed as a rectangle with block name written above of it. All blocks have some number of input and output ports, which allow signals to be passed to and from blocks. Input ports are drawn as squares on the left edge of the block, and output ports are drawn as circles on the right edge. The size of the block depends on number of input and output blocks.

Small numbers near input and output ports show port dimensions. For regular signals it is '1', for fuzzified signal it is number of membership functions used for fuzzification+1. Only ports with equal dimensions can be connected. Dimension of the port depend on each module type and configuration.

To access various edit functions QCOptimizer toolbar buttons are used:

(1) *Select tool* (Table A2): use this button to enable select mode (default). Click on the block to select it. Selected block is displayed with altered colors (depend ending on windows color settings). You can select multiple

blocks by holding *CTRL* key and clicking on blocks to be selected. To deselect all blocks click on a free space.

Selected block[s] can be moved across display. To move, click and hold left mouse on selected block, and drag it to desired location. Release mouse button to stop moving.

You can also access block properties in this mode. Double click left mouse button over block to display corresponding option dialog box.

(2) *Move window tool* (Table A2): select this button to move display window if the model does not fit inside screen. Click toolbar button and then click and hold left mouse button on model view and drag mouse to change display position. Release mouse to stop moving.

(3) *Create block* (Table A2): click this button to create new block. After you click it menu with list of available block types will be displayed. Select required block type from this menu. Depending on block type dialog box may appear, where you should enter initial block settings. After finishing with block settings, you will see new block displayed in model view. Drag block to its desired location and click mouse to finish block creation.

(4) *Create link tool* (Table A2): click this button to create link between blocks. Then click to ports (one input and one output, in any order) to create link between ports. Any output port can have as many connections as required. Input ports can have only one connection. If you select input port which already have connection it will be reconnected as instructed.

(5) *Delete block tool* (Table A2): To delete block click this button and then block to be deleted.

(6) *Delete link tool* (Table A2): To delete link click this button and then input port, which should be disconnected. You cannot delete links by clicking on output ports.

Table A2. The icons

	<i>Select tool</i>
	<i>Move window tool</i>
	<i>Create block</i>
	<i>Create link tool</i>
	<i>Delete block tool</i>
	<i>Delete link tool</i>

A2.3.7 Learning Signal Manager

If you want to use *learning signal* for optimization it should be loaded and assigned for model variables before use.

QCOptimizer may load teaching signal from Matlab v.4 and v.5 files and from text files. File type is autodetected. If the file does not look like Matlab file than it assumed to be a text file.

Matlab file should contain an array of real numbers with number of columns equal to sum of number of input and output variables of the model. If version file contains several variables then first one will be loaded.

Text files are processed using *locale settings*. Please set *locale settings* as described in section 4.2.5 before loading text files. If *locale settings* do not match the file format than signal will be loaded incorrectly and no error message will typically be displayed. Text file should contain *learning signal* data separated by any separators. This will include text files produced by Matlab, CSV files and practically any other file of this kind.

To manage learning signals, select *File/Signal Manager* command from QCOptimizer menu. *Signal manager window* will be displayed on Figure A13.

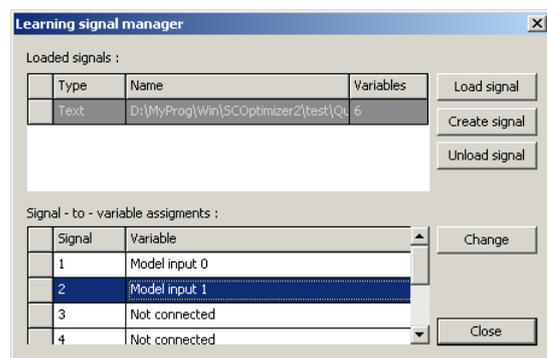


Figure A13. Signal manager dialog box

To load new signal file press *Load signal* button. Standard windows file open dialog will be displayed. Select file with signal and click *Open* to load it. Loaded files will appear in list box labeled *Loaded signals*.

Click on a file name in this list box and its signal to variable assignments will be displayed below. Those define how loaded signal and model variables are related. By default, table shows “not connected” to indicate that signal variable is not assigned to model variable. To set the assignment (or change it) click *Change* button. Another dialog will appear as shown on Figure A14.

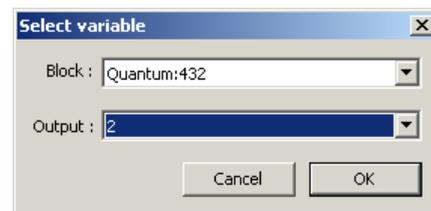


Figure A14. Variable assignment dialog box

In this dialog select block and *output* port name corresponding to loaded teaching signal variable. Press *OK* to save assignment. To assign signal to model input select “Model Input/Output” as block name, then select corresponding input or output in second list.

Loaded and assigned signals can be used during optimization, if teaching signal is selected as a source of fitness function in optimization manager parameters. In this case for model evaluation loaded signals will be presented on model input and block outputs will be compared to assigned signals.

File with learning signal can be unloaded by selecting it in the list and clicking Unload button. *Unloaded* signals will not be used during optimization.

A2.3.8. Available blocks

Blocks are building parts of a control system models. Following block types are supported:

- (1) *Model input*: feeds input data to model;
- (2) *Model output*: used to pass calculated output from model to external world;
- (3) *Fuzzification*: performs fuzzification of signal;
- (4) *Inference*: fuzzy rule database, performing Sugeno inference;
- (5) *Constant*: generates constant signal;
- (6) *Quantum generalization*: perform quantum generalization algorithm;
- (7) *File*: injects data from external file to model;
- (8) *Sum*: calculates weighted sum of its inputs;
- (9) *Scale*: perform scale and offset on signal.

Blocks are described in the following sections.

A2.3.8.1 Model Input Blocks

Input blocks are used to pass incoming data to model. They do not have input ports and receive data from external connection or assigned learning signal file.

Input blocks are numbered from 0, and the index of block is shown after block name. Vector signals from external connection are separated to components and send to input blocks according to their index. Block index depend on the order in which blocks where created and can only be changed by deleting blocks with lower indexes.

Input blocks also support signal normalization. To alter signal normalization parameters double click block. Following dialog will be displayed on Figure A15.

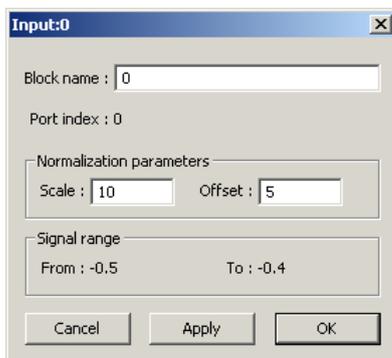


Figure A15. Input block parameter dialog box

In this dialog you can change block name (initially equals block index) and normalization parameters. Signal range corresponding to current not-normalized signal are shown below.

Press *OK* to accept changes and close window. Press *Apply* to change block parameters without closing window. Press *Cancel* to close window without applying modifications.

A2.3.8.2 Model Output Blocks

Output blocks are used to pass results of calculation to external world. Data arriving at output blocks is then sent over external link or compared to assigned learning signals.

Output blocks are numbered from 0, and the index of block is shown after block name. Output signal sent over external connection are compiled from signals of output blocks in order of their indexes. Block index depend on the order in which blocks where created and can only be changed by deleting blocks with lower indexes.

Output blocks support signal denormalization. To alter signal normalization parameters double click block. Following dialog will be displayed on Figure 16.

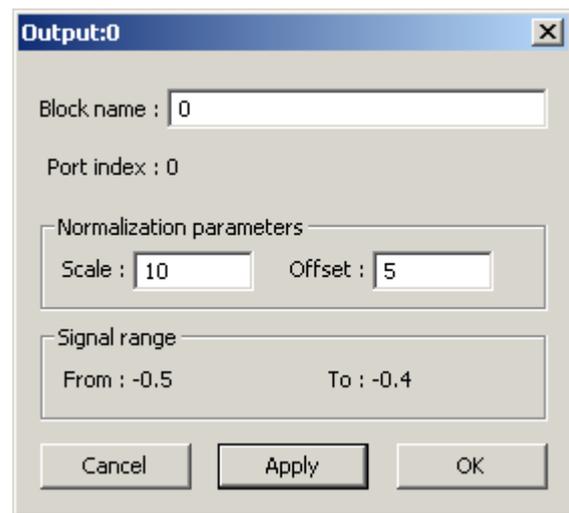


Figure A16. Input block parameter dialog box

In this dialog you can change block name (initially equals block index) and normalization parameters. Signal range corresponding to current not-normalized signal are shown below.

Note that output block perform demoralization on the signal, so that directly connected input and output blocks with equal parameters will leave signal intact. Input block offset signal to offset parameter and than scale it, *output block perform reverse operation*.

Press *OK* to accept changes and close window. Press *Apply* to change block parameters without closing window. Press *Cancel* to close window without applying

modifications.

A2.3.8.3 Fuzzification Blocks

Fuzzification blocks perform fuzzification of input signal. During block creation dialog box will appear where you should enter number of fuzzy membership functions and block name (Figure A17).



Figure A17. Fuzzification block creation dialog

Once created fuzzification block parameters can be set using properties dialog (Figure A18).

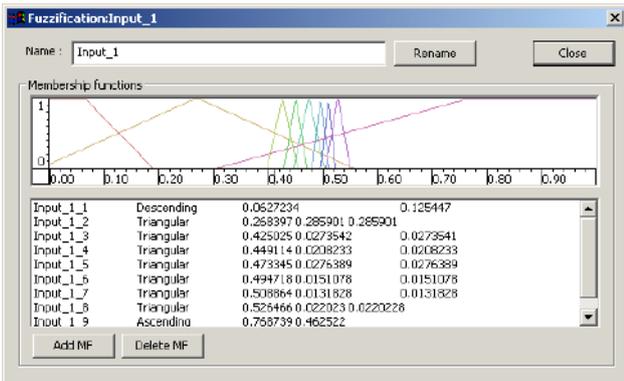


Figure A18. Fuzzification block parameter dialog

There is the name of the block atop of the dialog.

Then there is display *membership functions* (MF's) of the variable.

Graphical window display *distribution functions* of MF's. You can change appearance of this window by the pop-up menu, activated by the right-click of the mouse in the window. Menu items are the following:

Track cursor: Use this feature to see the margins of alpha-levels. When mouse cursor is on the y-axis then lines representing alpha level will be drawn, as well as color lines which will show margins of this level for all MF's. When cursor is somewhere else on the window then vertical lines at the position of cursor and horizontal lines from intersections of this line with MF's will be drawn.

Display MF supports: Display supports of MF's using colored vertical lines.

Display signal interval: Display margins of signal change interval with vertical lines.

Color shapes: Draw functions using filled color figures (default).

Color lines: Draw functions using color lines.

B&W lines: Draw functions using black lines.

Save Image: Save current image to file (Windows BMP format).

You can use this window to change MF distribution parameters. Move mouse to the x coordinate of the modal value or support margin of one of the MF's. Colored line will appear showing selected parameter. Press and hold left mouse button. Move mouse left or right to change the parameter. New shape of the MF will be drawn using outline method. Release left mouse button when you are satisfied with the shape of MF.

The list in the very bottom of the page display *membership functions* and their parameters. First column of the list is MF name, next *distribution* type followed by *distribution parameters*.

You can change those parameters by double clicking list items. If you do, the dialog box shown on Figure A19 will appear.

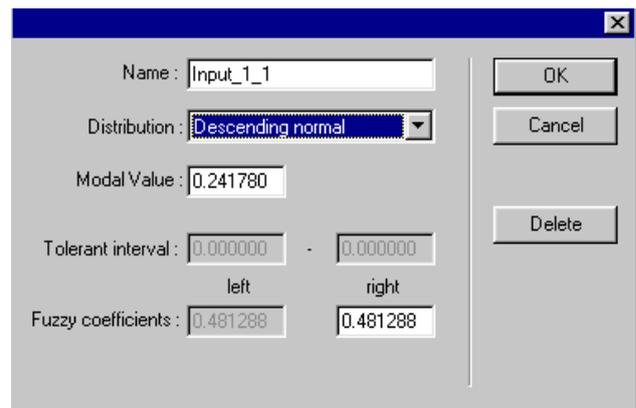


Figure A19. Membership function parameters dialog

This dialog box display all parameters of the membership function. Parameters can be changed by entering new data into corresponding fields. Please note that some parameters may not be available for different distributions. For example, *Descending normal* distribution require only *Modal Value* parameter and *Right Fuzzy Coefficient* parameter. Fields, corresponding to the unused parameters, are grayed and can not be changed.

When you are done with this dialog press *OK* to apply you changes or *Cancel* to return without modifying the variable.

You can also use this dialog to delete *membership functions* of input variables. If you wish to do it press *Delete* button.

Add MF button allow you to manually add new membership function to the current variable. After you press this button the same dialog as was used for editing new variable will appear and you will be able to enter param-

eters of newly added membership function. If you press **OK** MF with those parameters will be added, if you press **CANCEL** it won't. Since rule database structure is highly dependant on number of MF's, rule database will be cleared and recreated if you add a MF.

A2.3.8.4 Inference Blocks

Inference block support fuzzy inference with Sugeno model. When the block is being created following dialog will appear, where you should enter initial block parameters (Figure A20)

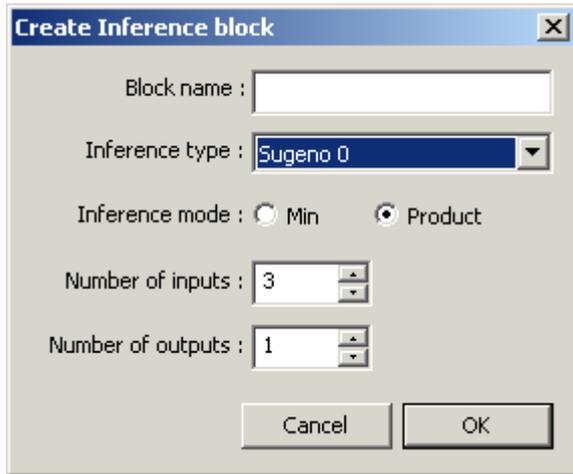


Figure A20. Inference block creation page

To view or edit inference block properties double click the block. Dialog box as shown on Figure A21 will appear.

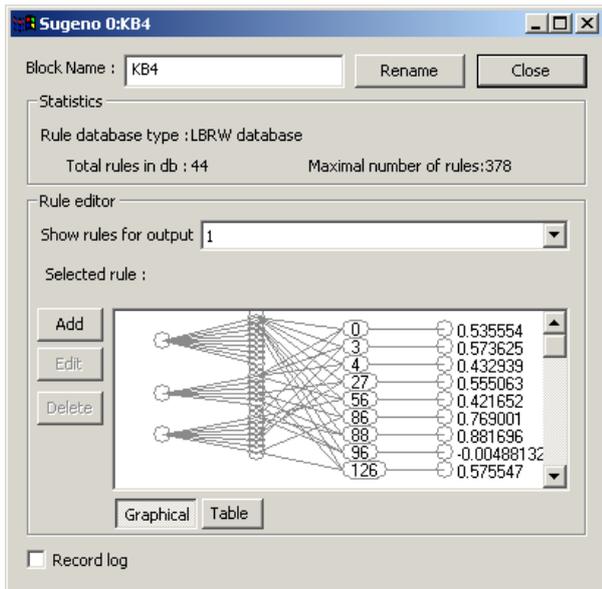


Figure A21. Inference block parameter dialog

This dialog displays following parameters:

Rule database type: type of the database used in model. Can be *Complete rule database* or *LBRW Rule database*.

Maximal number of rules: Maximal number of rules for current model.

Total rules in db: Number of rules stored in the database. For *complete database* this is always equal to maximal number of rules. For *LBRW database* it can be less then maximal.

Show rules for output: Rule database is displayed for one of the outputs. This list selects output variable for which database will be displayed.

Selected rule: Displays textual representation of selected rule, if any.

Rule Database editor display database as network with four layers. First layer is *input variable layer*. Each circle in this layer represents *input variable*. Second layer is *input MF layer*. Circles of this layer represent *membership functions* of variables. Circles in the third layer represent *rules* of the database. Number written inside this circle is a *rule number* of the rule in the database. Last layer is the *output layer*. For *Mamdani model* output layer is composed of circles, corresponding to *membership functions* of selected output variable. For *Sugeno models* output layer displays numerical parameters of the rule.

Database structure is shown with lines, connecting different layers. Each node in the *rule level* is linked with those MF's in *input MF layer*, which are included in the *if-part* of the rule. It is also linked with output MF or numerical parameter of the *then-part*.

Since all rules of the model won't fit on display, scroll bar is implemented which scrolls nodes of the *rule level*.

You can select a *rule* from the database by clicking on the node of the *rule level*. Textual representation of the rule will be shown in the *Selected rule* field and you will be able to edit or delete this rule. When you select a rule in the database its activation level is displayed as red lines on the *Rule Activation Level* page.

If you wish to delete the rule, select it and press *delete* button. Rule will be removed from the database. Note that rules from complete database can not be deleted.

By pressing *Edit* button, you can change selected rule. The dialog window as shown on Figure A22 will appear.

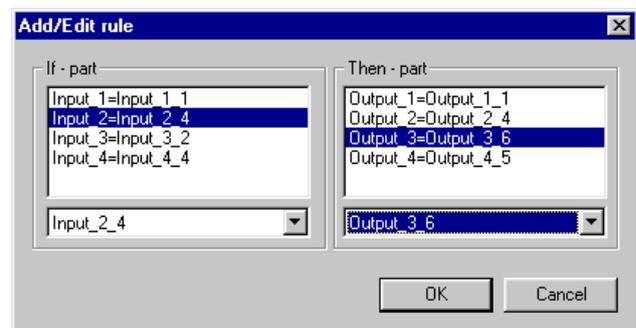


Figure A22. Add/Edit Rule dialog for Mamdani model

The left part of the dialog represents *if-part* of the rule, and the right part corresponds to *the then-part*. You can change parameters of any part by selecting items from the list and changing values in the drop-down box below the list.

For *Sugeno 0* model this dialog will have a slightly different appearance (Figure A23).

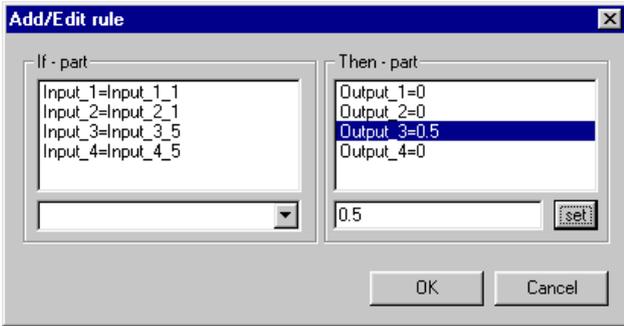


Figure A23. Add/Edit Rule dialog for Sugeno 0 model

To change output parameter for *Sugeno 0* model select corresponding line from the list, enter new value in the text field below and press *set*.

You can add rule to the database by pressing *Add* button. The same dialog as used for rule editing will appear. Change values in the *if-part* and *then-part* as desired and press *OK* to add rule. If the rule with selected *if-part* already exists in the database it will be replaced with new rule.

A2.3.8.5. Constant Blocks

Constant blocks can be used to create constant signals in the model. During block creation the dialog box as shown on Figure A24 will appear.

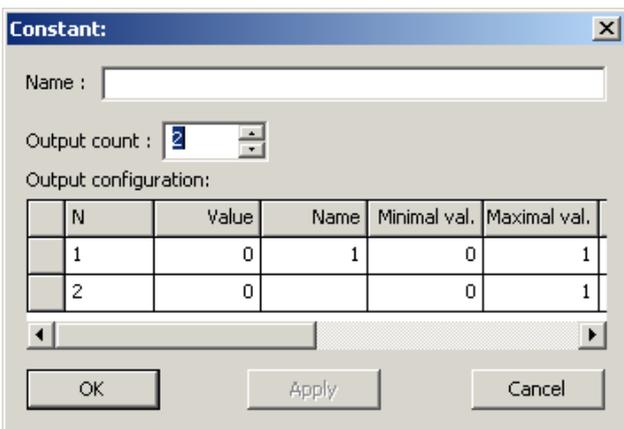


Figure A24. Constant block parameter dialog

Enter number of block outputs desired and configuration of each output, as follows:

- (1) Value is constant value
- (2) Minimal and Maximal values are range of value

search for block optimization

(3) Cyclic is indicate that value have cyclic property which is used during optimization

Number of block outputs cannot be changed after the block is created. Other parameters can be changed by activating parameter dialog by double clicking on the block.

A2.3.8.6 Quantum Generalization Blocks

During creation of quantum generalization block you should enter initial parameters in the dialog box shown on Figure A25.

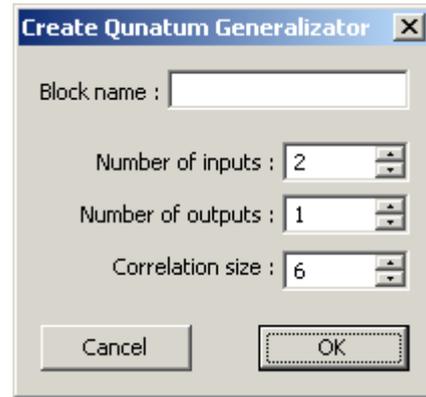


Figure A25. Quantum block creation dialog

Enter block name, number of inputs and outputs, size of correlation vector. Correlation vector size can be changed later, while number of inputs and outputs cannot. When done press *OK* to create block.

Configuration of block can be checked and changed by double clicking on the block. The dialog box as shown on Figure A26 will appear.

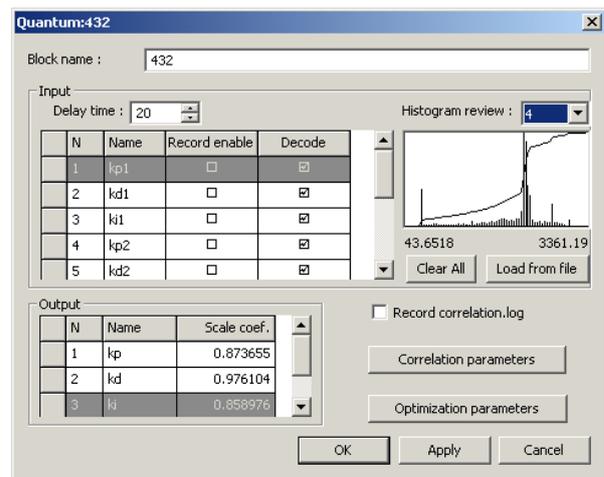


Figure A26. Quantum block parameters dialog

Main page of parameter dialog show configuration of inputs and outputs. Table in the input section lists following configurable parameters for each input:

- (1) *Name*: input name, to be used in correlation configuration;

(2) *Record enable*: check to enable histogram recording for selected channel;

(3) *Decode*: uncheck to skip this input during decoding (required by QPID algorithm).

To the left of table there is a histogram review window. It shows histogram for the channel selected in the drop-down list above it. *Clear All* button can be used to erase all recorded histograms for all channels.

Delay time control in the top part of the section determine amount of delay introduced for temporal correlation.

In the output section there is a table with output parameters:

(1) *Name*: output name, to be used in correlation configuration;

(2) *Scale coef.*: output is scaled to this constant.

Output scale coefficients can be optimized. Optimization settings can be accessed in dialog box activated by *Optimization Parameters* button (Figure A27).

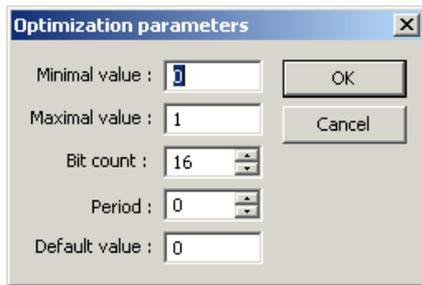


Figure A27. Quantum block optimization parameters dialog

The dialog allow selection of parameter search range, default value, number of bits for binary GA chromosomes and periodic property, if any.

Quantum correlation can be configured by pressing *Correlation parameters* button (Figure A28).

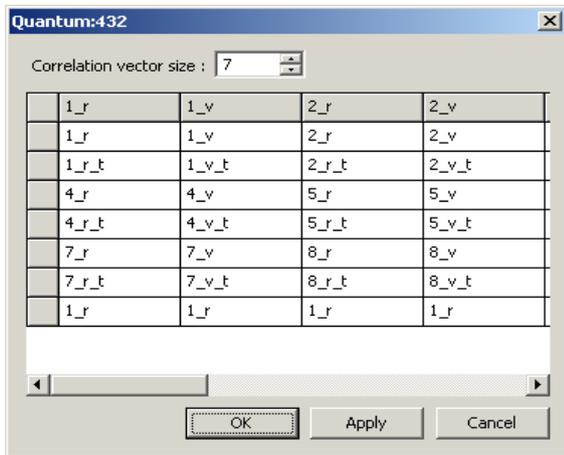


Figure A28. Quantum correlation parameters dialog

Input field at the top part of window allow changing correlation vector size. Table below lists current correla-

tion vectors. Each to columns of the table correspond to real (one with “_r” appended) and virtual (“_v”) parts of a vector for each output.

Entries in the table show which value will be used at which position of a vector. Those values can be changed by clicking on them and selecting new value from menu. Name of the component is composed of input name (or input number if name is not specified), and modifier, as follows:

- (1) *_r*: real component is used;
- (2) *_r_t*: real delayed component used;
- (3) *_v*: virtual component used;
- (4) *_v_d*: virtual delayed.

Quantum algorithm requires the histogram to be loaded for each channel. Histograms can be recorded during block operation or by processing file from disk. In the first case check *Record enable* checkbox in input table of corresponding inputs and run signal through the model. Data appeared on inputs corresponding to checked table lines will be recorded in histogram. To load histogram press “Load from file button” (Figure A29).

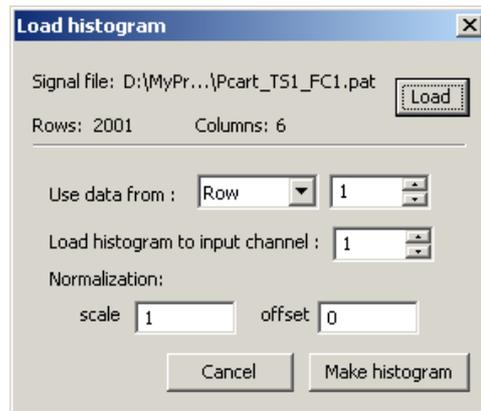


Figure A29. Histogram load dialog

In the dialog appear first press *Load* button and select file to be used. File name and number of rows and columns will be displayed. Then select if the data used will be taken from *row* or *column* (depending on file dimensions and format). Select file row/column index and index of input where histogram should be loaded. If required enter normalization parameters. Press *Make histogram* to load specified data. You can than select another indexes and load histograms to other inputs by pressing *Make histogram* again. Press *Cancel* when done with loading.

A2.3.8.7 File Blocks

File block can be used to insert data from external file to the model. Block support reading of Matlab v.5 and v.6 and text files. File type is detected automatically.

During of block creation the dialog box as shown on Figure A30 will appear.



Figure A30. File block parameter dialog

Select file you wish to use by clicking on “...” button and selecting file name using standard windows file open dialog. Data from file can be normalized before sending to block output. Normalization parameters should be entered to the corresponding fields of the dialog. Press *OK* to create block.

During operation file block output one record from the file for every model calculation step. Current record index is shown in block properties dialog and can be reset to zero by pressing *Rewind* button. Block automatically restarts from first record when all records from file are read.

A2.3.8.8 Sum Blocks

Sum block calculate weighted average of its inputs. During creation of a block dialog window will appear where you should select required number of inputs and initial weights (Figure A31).

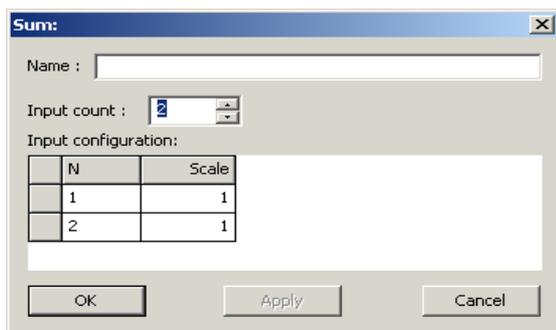


Figure A31. Sum block parameter dialog

Enter desired number of inputs in input count field, weight factors in scale column in input configuration table and press *OK* to create the block.

You cannot change number of inputs of sum block, but input weights can be changed. To do it double click the block. The same dialog window will appear, where you can set weight factors as desired.

A2.3.8.9 Scale Blocks

Scale blocks transform input signal by scaling and offsetting it. To set scale and offset parameters double click

the block (Figure A32).

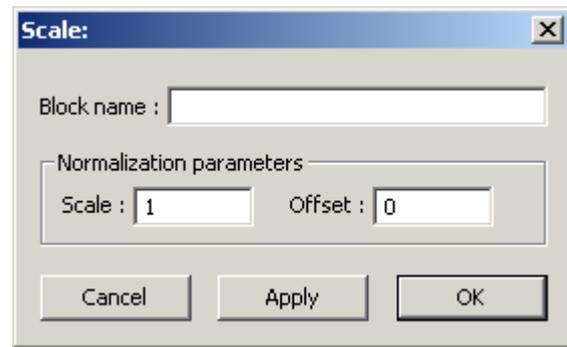


Figure A32. Scale block parameter dialog

Set desired parameters in dialog and press *OK* or *Apply* to change block parameters.

A2.3.9 External Link

QCOptimizer support data exchange with external programs, which can be used to pass data from external program as input to QCOptimizer model and return resulting model output. This link uses the same interface as one available for external inference in SCOptimizer, and can be accessed from SCLib library functions.

If current fitness calculation mode is set to “teaching signal” external link is not used and cannot be activated. Change fitness calculation mode to Matlab/external signal to enable external link access.

To connect via external link QCOptimizer and another program should use the same link name. Default link name is SC2. Link name in SCOptimizer can be changed by pressing  toolbar button. Dialog box will appear where you can enter desired link name (Figure A33).



Figure A33. Link name configuration dialog

Current link name is also displayed in the status line in the left-bottom corner of main QCOptimizer window preceding with “IPC:”.

To enable external link press  toolbar button. To indicate that QCOptimizer is ready for external connection an “*” symbol will be shown after link name in status line. To disable external connection press , “*” will disappear.

Always enable external link first from QCOptimizer and than in external program. Close link in reverse order – first in external program, then in QCOptimizer.

A2.3.10 Viewing Data Passing through the Model

You can view signals passing through the model in the graphical representation. To view signal select *View/Open signal view* menu item. Signal view window will appear, showing signals on model input. To change signal shown in window click right mouse button above window. In the menu appear select block (or model input/output) and than block output.

You can open as many data view windows as you like, by selecting the same menu item several times.

A2.3.11 Testing Model on Input Signal from File

You can make QCOptimizer perform calculations using input signal recorded in file. To do so you should:

- (1) Load signal using *Teaching Signal Manager* and associate it with model inputs.
- (2) Select “*teaching signal*” fitness mode calculation in optimization configuration.

Once those two actions are done, you should see “file” displayed in the status line typically used to show external link name. Press  toolbar button to perform a full run through data available in loaded files. You can view resulting signals using signal view windows, or you can use this mode to record histograms in quantum block.

A2.3.12 Optimization

QCOptimizer support various optimization algorithms and modes. It can perform optimization of single block, or several blocks at a time, including optimization with different algorithms, running at the same time and evaluated over same link.

A2.3.12.1 Selecting Optimization Options

To set optimization settings press  toolbar button (Figure A34).

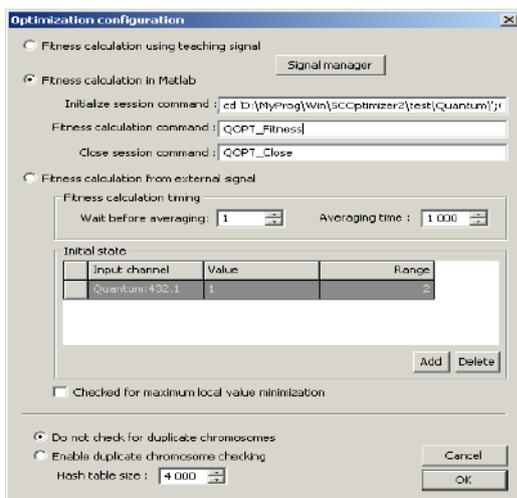


Figure A34. Optimization settings

First select desired fitness calculation mode:

(1) *Teaching signal*: model is evaluated using *teaching signal* loaded in *Teaching Signal Manager* and fitness is computed by comparing actual data from model with data selected as *teaching signal*.

(2) Fitness calculation in matlab: during optimization QCOptimizer set model to the state being evaluated and calls matlab function specified in configuration. Function should evaluate model quality (typically using external link to do some calculations) and return fitness value. To work in this mode matlab should be available and one should create three functions and set it names to following parameters:

- ① Initialize session command: called by QCOptimizer when optimization is started. Should perform initialization, for example load some simulink model.
- ② Fitness calculation command: called by QCOptimizer to calculate fitness for current state.
- ③ Close session command: called when optimization is finished. May free resources allocated for calculations.

(3) Fitness calculation from external signal. Signal coming from external link is used for model input and fitness calculation. Concrete fitness function is configured during optimization start, but here you can enter initial conditions and calculation time. Initial conditions are set in *Initial state* table where you can enter desired initial value for some variables and possible delta. In example shown on figure above optimization will start when output 1 of quantum block 432 will equal 1 ± 2 . Once initial condition is met QCOptimizer will wait for delay time set in wait before averaging field and than calculate fitness function as average over averaging time selected.

In this window you can also select what QCOptimizer will do if it detects that the same model state evaluation is requested during single optimization process. If *do not check for duplicate chromosomes* is selected than duplicate detection will be disabled and all states will be evaluated. You can greatly improve optimization time by selecting single evaluation mode. In this case you may also want to set hash table size used to store processed states.

A2.3.12.2 Starting Optimization

To start optimization press  button. Optimization start dialog will appear (Figure A35).

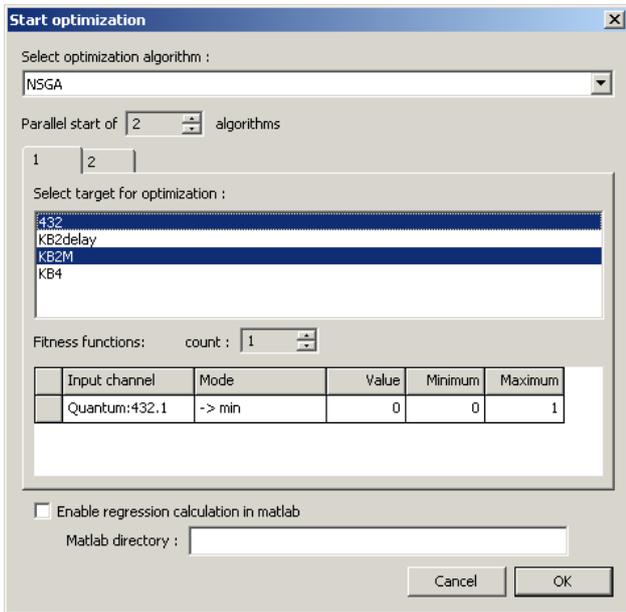


Figure A35. Optimization start dialog

First select optimization algorithm from list. Blocks available for optimization are shown in optimization target lists. Select blocks to be optimized by clicking on them.

Below optimization target list there is a fitness function configuration table. In “fitness calculation in Matlab” mode those settings are not used. For NSGA algorithm you can use more than one fitness function. In the table select desired signal, optimization mode (for minimum, for maximum, or to match a value).

You can start several instances of the same algorithm operating over different blocks by setting parallel start option to desired number. Several pages will appear where you can select targets for all algorithms.

Once you press *OK* you will be prompted to enter algorithm parameters (depending on algorithm selection). When you confirm algorithm settings optimization will start.

A2.3.12.3 Optimization Progress/Evaluation

When optimization is active optimization status dialog is shown on Figure A36.

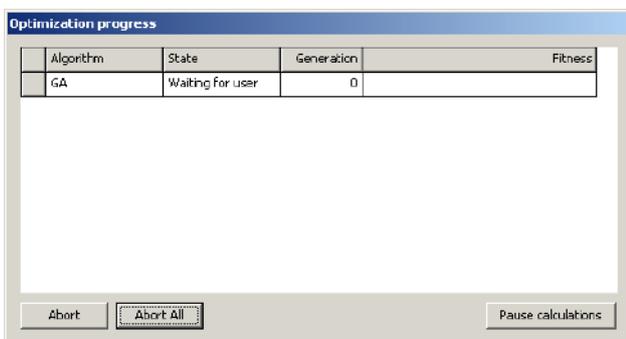


Figure A36. Optimization progress dialog

For each optimization currently active it displays a line, which shows current status, generation and fitness achieved. You can abort one of algorithms by selecting it and pressing *Abort button*. Pressing *Abort All* will abort all active algorithms. Algorithms may be aborted with some delay, if there is a calculation which cannot be stopped immediately.

Press *Pause calculations* to pause all calculations. Current evaluations will be restarted from beginning after calculation is resumed.

When calculation is over (or after each generation if you select corresponding option in algorithm configuration) the dialog will appear showing state of current population (Figure A37).

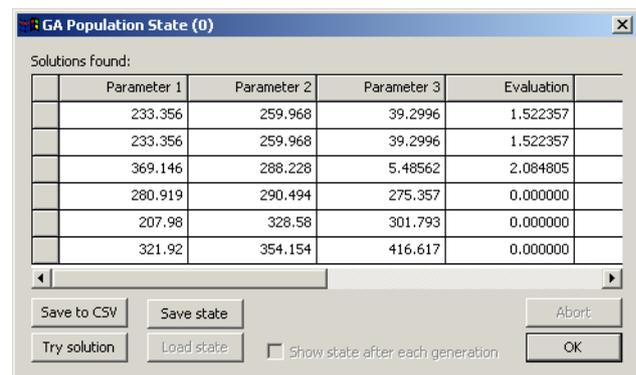


Figure A37. Optimization progress dialog

The table lists all parameters for all chromosomes, as well as evaluation, fitness and other algorithm-dependent parameters. Top line shows best chromosome available, next lists chromosomes of current generation. You can save this data to text (CSV) file by pressing *Save to CSV* button. Select chromosome and press *Try solution* to set model state to chromosome selected, so that you can evaluate it manually.

If this dialog is shown after optimization is over, you should select a chromosome which will be used as optimization result (by default best one shown in top line is used). When you are finished this chromosome will be applied to model.

References

- [1] Ulyanov, S. V. System and method for control using quantum soft computing [P]. US Patent No 7,383,235 B1, 2003; EP PCT 1 083 520 A2, 2001; Efficient simulation system of quantum algorithm gates on classical computer based on fast algorithm [P]. US Patent No 2006/0224547 A1, 2006.
- [2] Ulyanov, S. V. Quantum fast algorithm computational intelligence PT I: SW / HW smart toolkit [J]. Artificial Intelligence Advances, 2019, 1(1): 18-43.

DOI: <https://doi.org/10.30564/aia.v1i1.619>

- [3] Grover, L. K. A fast quantum mechanical algorithm for database search[P]. US Patent US 6,317,766 B1, 2001.
- [4] Litvintseva, L. V. et al. Quantum fuzzy conclusion for creating knowledge bases in robust intelligent regulators[J]. RAN News. Theory and control systems, 2007, 6: 71–126. (in Russian).
- [5] Litvintseva, L. V., Ulyanov, S. V. Intelligent control systems. 1. Quantum computing and self-organization algorithm[J]. RAN News. Theory and control systems, 2009, 6: 102–141. (in Russian).
- [6] Ulyanov, S. V., Sorokin, S. V., Litvintseva, L. V. Knowledge Base Optimizer based on quantum computing for designing self-organizing fuzzy controllers: software tools[J]. System analysis in science and education, 2012, 1: 1–27. (in Russian).
- [7] Nielsen, M. A., Chuang, I. L. Quantum computation and quantum information[M]. UK, Cambridge, University Press, 2000: 700.
- [8] Marinescu, D. C., Marinescu G. M. Approaching quantum computing[M]. New Jersey: Pearson Prentice Hall, 2005: 400.
- [9] Nikolaeva, A. V., Ulyanov, S. V. Design of an intelligent robot control system by a manipulator. Part 3: Modeling and physical experiment based on quantum computing technologies[J]. System analysis in science and education: a network scientific publication, 2013, 1: 1-25. (in Russian).
- [10] 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).
- [11] Mikhailov, V. S. Control Theory[M]. K: Higher School, 1988: 312. (in Russian).
- [12] Nikolaeva, A. V., Ulyanov, S. V., Barkhatova, I. A., Nozdrachev, A. V. Intelligent control system for redundant robotic arm with seven degrees of freedom based on soft computing[J]. System analysis in science and education: a network scientific publication, 2014, 2: 48-55p. (in Russian).
- [13] Ulyanov, S. V. Self-Organized Intelligent Robust Control Based on Quantum Fuzzy Inference [J] Recent Advances in Robust Control - Novel Approaches and Design Methods, ISBN: 978-953-307-339-2, InTech, 2011. Available from: <http://www.intechopen.com/books/recent-advances-in-robust-controlnovel-approaches-and-design-methods/self-organized-intelligent-robust-control-based-on-quantum-fuzzy-inference>.