QUANTUM SOFTWARE ENGINEERINGSUPREMACY PT 1. QUANTUM COMPUTING OPTIMIZER TOOLKIT OF ROBUST KNOWLEDGE BASES DESIGN FOR FUZZY CONTROLLER

ULYANOV SERGEY[1], LITVINTSEVA LIUDMILA[2], ULYANOV VIKTOR [3].

[1]Dubna State University, Institute of system analysis and control, Dubna, Russia,INESYS LLS, Moscow, Russia

[2]Intelligent information technology Dept, INESYS LLS MISIS, Moscow, Russia.

[3]Intelligent robotics and control lab., INESYS MISIS, Moscow, Russia;

Abstract

Quantum computing technologies for self-organizing Knowledge Bases (KB) designof intelligent regulators considered. 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) implements the self-organization of the KBs, based on the physical laws of the theories of quantum information and quantum computing, 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. Quantum software toolkit for optimization of knowledge base introduced. Quantum supremacy of developed computational intelligence toolkit demonstrated on intelligent robotics as robust control of two globally unstable objects. New synergy effect of robust KB design from two imperfect KB (designed on soft computing optimizer in learning situations) for unpredicted control situation in on line on concrete example described.

Keywords

Intelligent control systems, fuzzy controller, quantum soft computing.

Introduction

The general hierarchical structure and stages of execution of information technology embedded in the process of design of integrated fuzzy PID controllers for autonomous and interconnected COs with different physical nature described in [1, 2].

This technology applied computational intelligence toolkit for design of *Knowledge Bases* (KB) in the fuzzy controller (FC) of the lower executive level. The main role in the structure of this technology played by the development of *robust* KBs based on corresponding optimizers (see the

DOI: 10.5121/ije.2020.05104

block "*Information design technology*", Fig. 1 in [1] labeled by dashed lines). We have considered an application of *quantum computing ideas* for solution of control problems and have developed a quantum control algorithm based on Quantum Fuzzy Inference (QFI) model.The *quantum principles* (such as quantum parallelism, quantum complementary, quantum long-distance correlation, quantum bio-inspired searching, etc) can be used for applications of quantum strategies for optimal decision making with conventional computers in much the same way that genetic principles of evolution are used in genetic optimizers. Nature also uses the principles of quantum mechanics to solve problems, including quantum-like optimization-type problems, searching-type problems, selection-type problems, etc. [3].

The *quantum operators*, such as *superposition, entanglement* and *interference*, give rise to the quantum logic used in quantum computations. Moreover, the usefulness of these quantum operators gives rise to the new viewpoint on control and self-organization algorithms.

QFI system is a quantum algorithm (QA) block, which performs post processing of the results of fuzzy inference of each independent FC and produces the generalized control signal output. In this case, the output of QFI is an optimal robust control signal, which combines best features of the each independent FC outputs. Therefore, the operation area of such a control system can expanded greatly as well as its robustness. Robustness of control signal is the background for support the reliability of control accuracy in uncertainty environments. Successful sophisticated search solutions of intractable robotic task's as global robust intelligent and cognitive smart control in unpredicted (unconventional) / hazard control situations or multi-criteria imperfect control goal is based on quantum control principles (as quantum neural network for deep machine learning or quantum genetic optimization algorithm).

It is important in these cases to choose types and kind of quantum correlations, as example, between PID-controller in coefficient gain schedule. Extracted from classical states (as example, from modeling of control coefficient gain's laws) quantum hidden correlations (that physically rigor and mathematically strong correctness, and corresponds to main qualitative properties in general of ill-defined control object) are considered as an additional physical computing and hidden quantum information resources.

These information resources changes the time-dependent laws of the coefficient gains schedule of the traditional controllers as PID-controllers with guarantees the achievement of control goal in hazard situations. This article discusses the application of quantum genetic algorithm to automatically choice the optimal type and kind of correlations in the quantum fuzzy inference. Efficiency of quantum search algorithm in imperfect KB self-organization on the Benchmark system "cart – pole" demonstrated.

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, 5] 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 structure of QFI model with self-organization KB properties that includes all of these QA features and quantum genetic algorithm (see, below Figure 1) based on above-mentioned proposals [6] and described as following:

$$QAG=\left[\left(Int\otimes^{n}I\right)\cdot U_{F}\right]^{h+1}\cdot\left[QGA\right]\left[{}^{n}H\otimes^{m}S\right] \quad (1)$$

The first part in Eq. (1) is the choice of the type of the entangled state of operator U_F. The basic unit of such an intelligent control system (ICS) is the quantum genetic search algorithm (QGSA) (see, [7]).

Results of simulation show computing effectiveness of robust stability and controllability of (QFI + QGA)-controller and new information synergetic effect: from two fuzzy controllers with imperfect knowledge bases can be created robust intelligent controller (extracted hidden quantum information from classical states is the source of value work for controller [1]) in on-line (see below Fig. 8). Intelligent control systems with embedding intelligent QFI-controller can be realized either on classical or on quantum processors (as an example, on D-Wave processor type).

Main peculiarities of QFI model are following:

- 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 [4, 5];
- 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);
- QFI is a new quantum search algorithm (belonging to so called *QPB*-class) that can solve classical algorithmically unsolved problems [7, 8];
- 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;
- Model of QFI supports the thermodynamic interrelations between *stability*, *controllability* and *robustness* for design of self-organization processes.

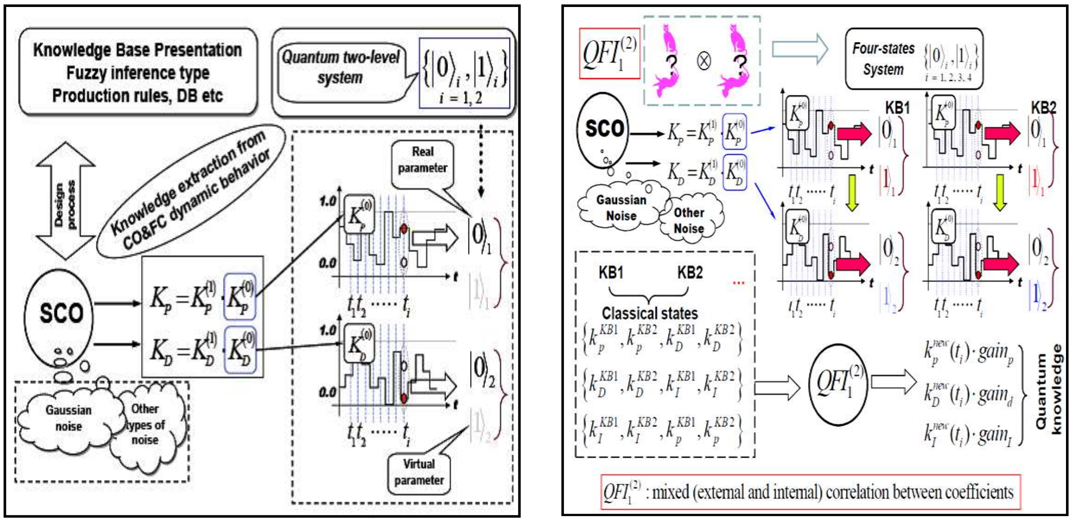
Specific features of QFI applications in design of robust KB. Let us stress the fundamentally important specific feature of operation of the QA (in the QFI model) in the design process of robust laws for the coefficient gain schedules of fuzzy PID controllers based on the individual KB that designed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

1.2. Quantum information resources in QFI algorithm

In this section, we introduce briefly the particularities of quantum computing and quantum information theory that 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 1) for the given control task in different control situations (for example, in the presence of different stochastic noises).

1.2.1. 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 Fig. 1).


Knowledge Base Presentation
Fuzzy inference type
Production rules, DB etc
Quantum two-level system
$\{|0\rangle_i, |1\rangle_i\}$
$i = 1, 2$
Design process
Knowledge extraction from CO&FC dynamic behavior
Real parameter
Virtual parameter
SCO
$K_P = K_P^{(1)} \cdot K_P^{(0)}$
$K_D = K_D^{(1)} \cdot K_D^{(0)}$
1.0
0.0
$t_1 t_2 \cdots t_i$
Gaussian noise
Other types of noise
$QFI_1^{(2)}$
Four-states System
$\{|0\rangle_i, |1\rangle_i\}$
$i = 1, 2, 3, 4$
KB1
KB2
Classical states
$\{k_P^{KB1}, k_P^{KB2}, k_D^{KB1}, k_D^{KB2}\}$
$\{k_D^{KB1}, k_D^{KB2}, k_I^{KB1}, k_I^{KB2}\}$
$\{k_I^{KB1}, k_I^{KB2}, k_P^{KB1}, k_P^{KB2}\}$
$k_P^{new}(t_i) \cdot gain_P$
$k_D^{new}(t_i) \cdot gain_d$
$k_I^{new}(t_i) \cdot gain_I$
Quantum knowledge
$QFI_1^{(2)}$: mixed (external and internal) correlation between coefficients

(a) (b)

Figure 1.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 presented below and described in details in [8].

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 shared between subsystems held by *A* (KB1) and *B* (KB2), with respective dimensions *d*,

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

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

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 (*A*) 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 (1), 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 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. Fig. 1, 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 [2, 3]. Interference operator extracts this maximum of amplitude probability with a classical measurement.

Figure 2 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).

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, 5]: (*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: (i) compression of classical information by coding in computational basis $\{|0\rangle,|1\rangle\}$ and forming the quantum correlation between different computational bases (Fact 1); (ii) separating and splitting total information and correlations on "classical" and "quantum" parts using Hadamard transform (Facts 2 and 3); (iii) 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.

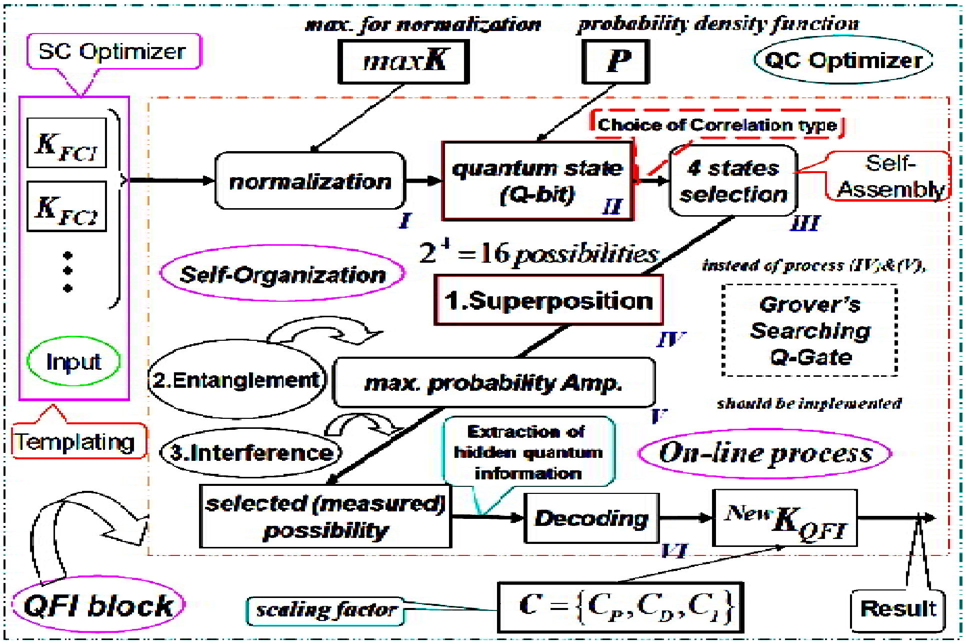


Figure 2. The structure of QFI gate

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

1.3. 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 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 [1].

Figure 3 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 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.

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.

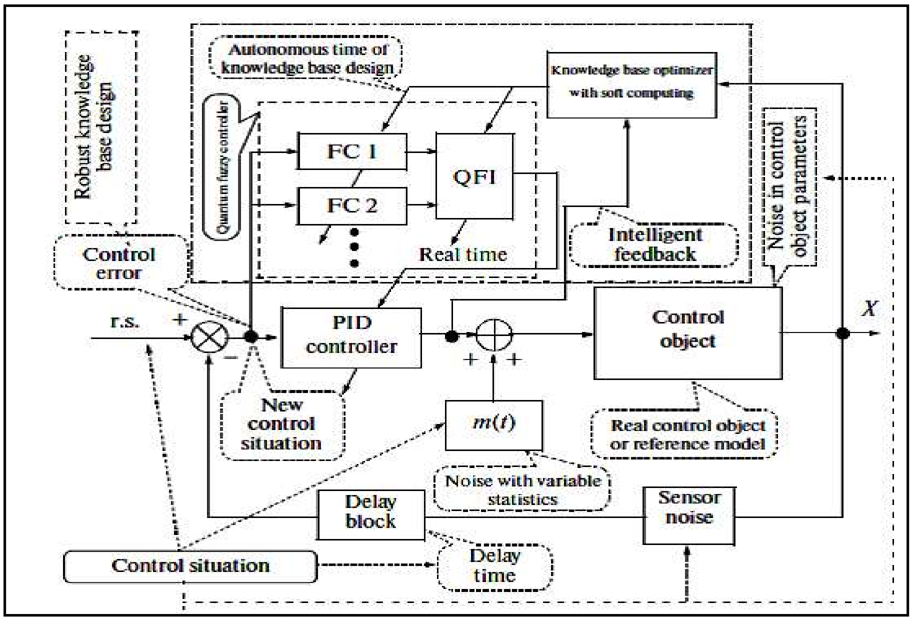

Robust knowledge base design
Autonomous time of knowledge base design
Knowledge base optimizer with soft computing
Quantum fuzzy controller
FC 1
FC 2
QFI
Real time
Intelligent feedback
Noise in control object parameters
Control error
r.s.
+
−
PID controller
+
+
Control object
X
New control situation
m(t)
Real control object or reference model
Noise with variable statistics
Delay block
Sensor noise
Control situation
Delay time

Figure 3. Structure of robust ICS based on QFI

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

1.3.1. 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 α_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 4 below [8, 9].

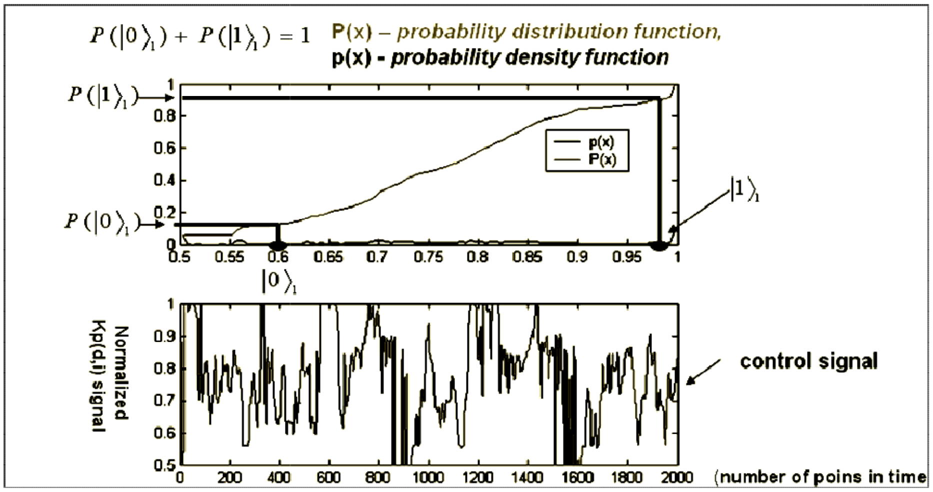


Figure 4. Example of control signal and corresponding probability distribution function

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 called *superposition*, *entanglement* (quantum correlation) and *interference* that physically are operators of quantum computing.Consider main steps of developed QFI process that is QA.

Step 1: *Coding*

- 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:
- calculation of probability amplitudes α_0, α_1 of states $|0\rangle$ and $|1\rangle$ from histograms;
- by using α_1 calculation of normalized value of state $|1\rangle$.

Step 2: *Choose quantum correlation type for preparation of entangled state.* In the Table 1 investigated types of quantum correlations are shown. Take, for example, the following quantum correlation type: $\{k_P^1(t), k_D^1(t), k_P^2(t), k_D^2(t)\} \to k_P^{new}(t)$, where 1 and 2 are indexes of KB.

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

Table 1. Types of quantum correlations

1. QFI based on *spatial* correlations	$k_P^{KB_1,KB_2}(t_i)k_D^{KB_1,KB_2}(t_i)\to k_P^{new}(t_i)\cdot gain_P$ $k_D^{KB_1,KB_2}(t_i)k_I^{KB_1,KB_2}\to k_D^{new}(t_i)\cdot gain_D$ $k_I^{KB_1,KB_2}(t_i)k_P^{KB_1,KB_2}(t_i)\to k_I^{new}(t_i)\cdot gain_I$
2. QFI based on *temporal* correlations	$k_P^{KB_1,KB_2}(t_i)k_P^{KB_1,KB_2}(t_i-\Delta t)\to k_P^{new}(t_i)\cdot gain_P$ $k_D^{KB_1,KB_2}(t_i)k_D^{KB_1,KB_2}(t_i-\Delta t)\to k_D^{new}(t_i)\cdot gain_D$ $k_I^{KB_1,KB_2}(t_i)k_I^{KB_1,KB_2}(t_i-\Delta t)\to k_I^{new}(t_i)\cdot gain_I$
3. QFI based on *spatio-temporal* correlations	$k_P^{KB1}(t_i)k_D^{KB1}(t_i-\Delta t)k_P^{KB2}(t_i-\Delta t)k_D^{KB2}(t_i)\to k_P^{new}(t_i)\cdot gain_P$ $k_D^{KB1}(t_i)k_I^{KB1}(t_i-\Delta t)k_D^{KB2}(t_i-\Delta t)k_I^{KB2}(t_i)\to k_D^{new}(t_i)\cdot gain_D$ $k_I^{KB1}(t_i)k_P^{KB1}(t_i-\Delta t)k_I^{KB2}(t_i-\Delta t)k_P^{KB2}(t_i)\to k_I^{new}(t_i)\cdot gain_I$

Step 3: *Superposition and entanglement.* According to the chosen quantum correlation type construct superposition of entangled states as shown on general Figure 5, where **H** is the Hadamard transform operator.

Step 6: *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 6a: Find robust QFI scaling gains $\{gain_P, gain_D, gain_I\}$ based on GA and a chosen fitness function.

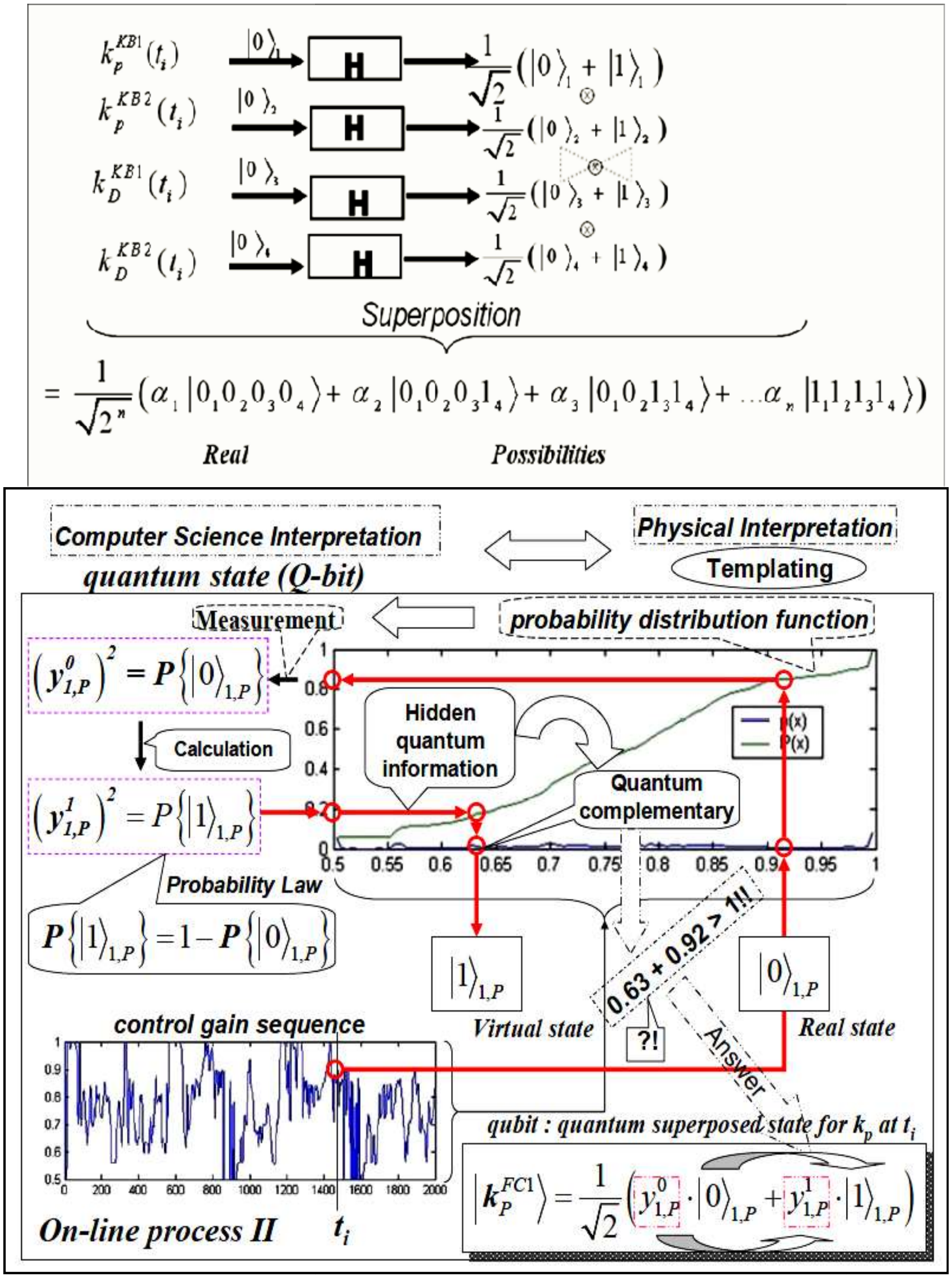

$k_p^{KB1}(t_i)$
$|0\rangle_1$
H
$\frac{1}{\sqrt{2}}(|0\rangle_1 + |1\rangle_1)$
$k_p^{KB2}(t_i)$
$|0\rangle_2$
H
$\frac{1}{\sqrt{2}}(|0\rangle_2 + |1\rangle_2)$
$k_D^{KB1}(t_i)$
$|0\rangle_3$
H
$\frac{1}{\sqrt{2}}(|0\rangle_3 + |1\rangle_3)$
$k_D^{KB2}(t_i)$
$|0\rangle_4$
H
$\frac{1}{\sqrt{2}}(|0\rangle_4 + |1\rangle_4)$
Superposition
$= \frac{1}{\sqrt{2^n}}(\alpha_1 |0_1 0_2 0_3 0_4\rangle + \alpha_2 |0_1 0_2 0_3 1_4\rangle + \alpha_3 |0_1 0_2 1_3 1_4\rangle + \dots \alpha_n |1_1 1_2 1_3 1_4\rangle)$
Real
Possibilities
Computer Science Interpretation
Physical Interpretation
Templating
quantum state (Q-bit)
Measurement
probability distribution function
$(y_{1,P}^0)^2 = P\{|0\rangle_{1,P}\}$
Calculation
$(y_{1,P}^1)^2 = P\{|1\rangle_{1,P}\}$
Probability Law
$P\{|1\rangle_{1,P}\} = 1 - P\{|0\rangle_{1,P}\}$
Hidden quantum information
Quantum complementary
p(x)
P(x)
1
0.8
0.6
0.4
0.2
0
0.5
0.55
0.6
0.65
0.7
0.75
0.8
0.85
0.9
0.95
1
$|1\rangle_{1,P}$
Virtual state
0.63 + 0.92 > 1!!
?!
Answer
$|0\rangle_{1,P}$
Real state
control gain sequence
On-line process II
t_i
0 200 400 600 800 1000 1200 1400 1600 1800 2000
0.5 0.6 0.7 0.8 0.9 1
qubit : quantum superposed state for k_p at t_i
$|k_P^{FC1}\rangle = \frac{1}{\sqrt{2}}(y_{1,P}^0 \cdot |0\rangle_{1,P} + y_{1,P}^1 \cdot |1\rangle_{1,P})$

Figure 5. The algorithm of superposition calculation

In proposed QFI we investigated the proposed types of quantum QFI correlations shown in Table 1 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 (Fig. 2) supporting a self-organizing capability of FC. Optimal drawing process of value information from a few of KBs as above-mentioned based on the following four facts from quantum information theory.

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 organized. The entanglement operation 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 used in the quantum block supporting a self-organizing capability of FC.

2. Robust FC design toolkit:SC and QC Optimizers for quantum controller's design

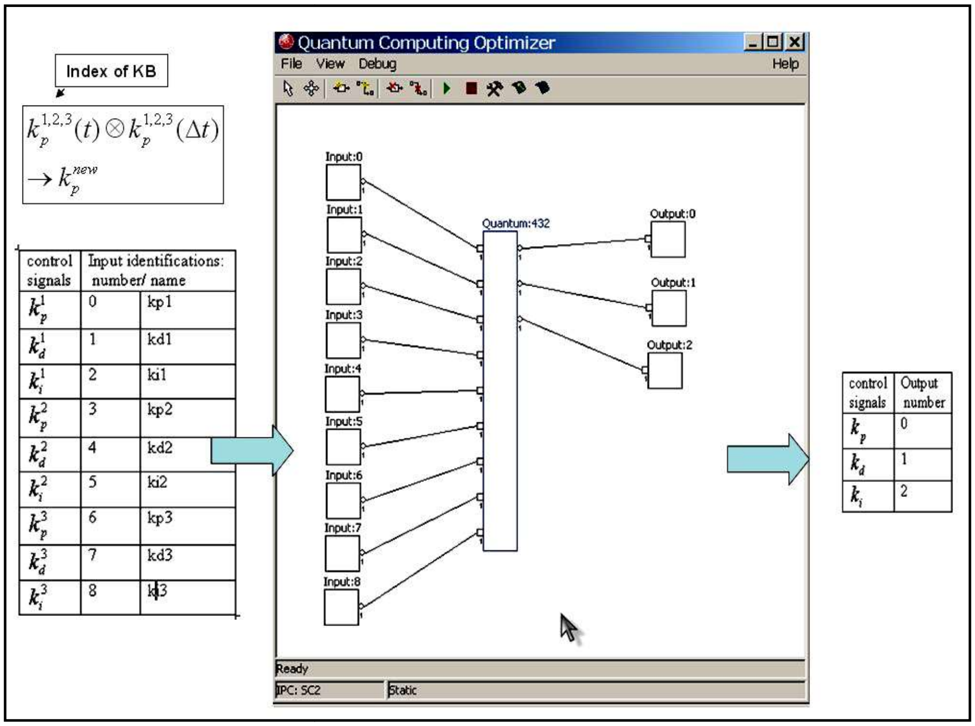
To realize QFI process we developed new tools called "QC Optimizer" that are the next generation of SCO tools [1].

2.1. QC Optimizer Toolkit

*QC Optimizer Toolkit*based on Quantum & Soft Computing includes next items:

- Soft computing and stochastic fuzzy simulation with information-thermodynamic criteria for robust KBs design in the case of a few teaching control situations;
- 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 shown on Figs 6 and 7.

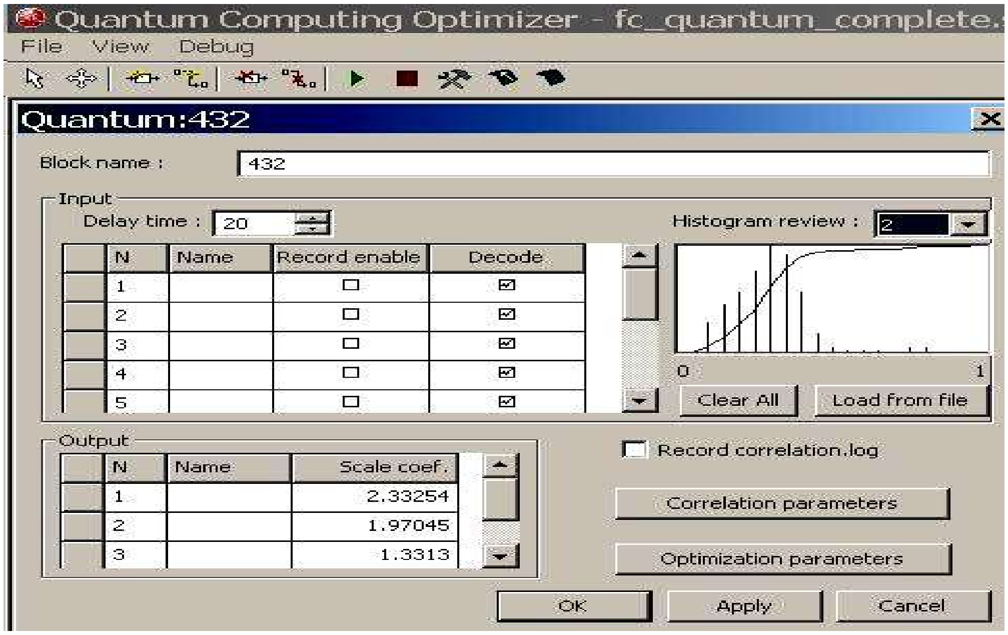


$k_p^{1,2,3}(t)\otimes k_p^{1,2,3}(\Delta t)$
$\to k_p^{new}$

control signals	Input identifications: number/ name	
k_p^1	0	kp1
k_d^1	1	kd1
k_i^1	2	ki1
k_p^2	3	kp2
k_d^2	4	kd2
k_i^2	5	ki2
k_p^3	6	kp3
k_d^3	7	kd3
k_i^3	8	ki3

control signals	Output number
k_p	0
k_d	1
k_i	2

Figure 6. First internal layer of QC Optimizer



Block name : 432

Delay time : 20

Histogram review : 2

N	Name	Record enable	Decode
1		□	☑
2		□	☑
3		□	☑
4		□	☑
5		□	☑

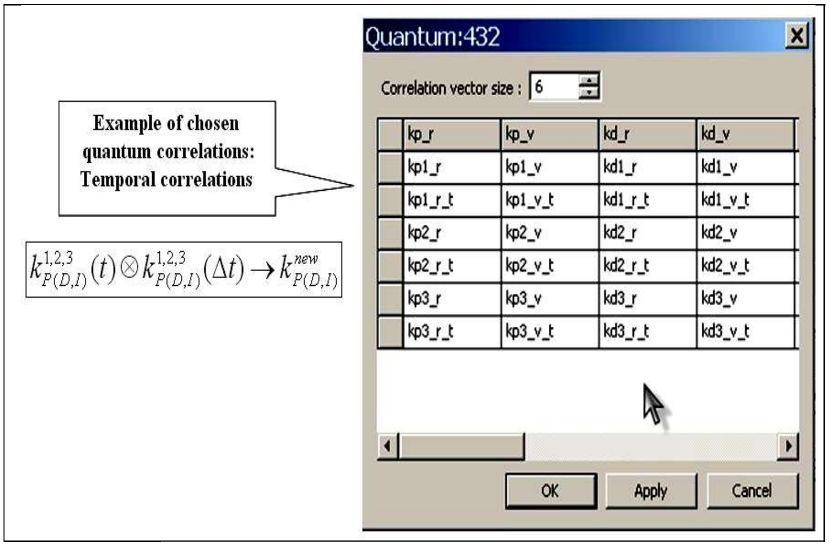
N	Name	Scale coef.
1		2.33254
2		1.97045
3		1.3313

□ Record correlation.log

Figure 7. Second internal layer of QC Optimizer

Remark. On Figure 6, the first internal layer of QC Optimizer shown (inputs/output). On Figure 7, the quantum block realizing QFI process based on three KB described. The “delay time = 20 (sec)” on the Fig. 7, corresponds to the parameter “Δt” given in temporal quantum correlations description (see Table 1); the knob named “correlation parameters” call other block (see Fig. 7) where a chosen type of quantum correlations (Table 1) is described.

On Fig. 8 description of temporal quantum correlations 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).



Example of chosen quantum correlations: Temporal correlations

$k^{1,2,3}_{P(D,I)}(t)\otimes k^{1,2,3}_{P(D,I)}(\Delta t)\rightarrow k^{new}_{P(D,I)}$

kp_r	kp_v	kd_r	kd_v
kp1_r	kp1_v	kd1_r	kd1_v
kp1_r_t	kp1_v_t	kd1_r_t	kd1_v_t
kp2_r	kp2_v	kd2_r	kd2_v
kp2_r_t	kp2_v_t	kd2_r_t	kd2_v_t
kp3_r	kp3_v	kd3_r	kd3_v
kp3_r_t	kp3_v_t	kd3_r_t	kd3_v_t

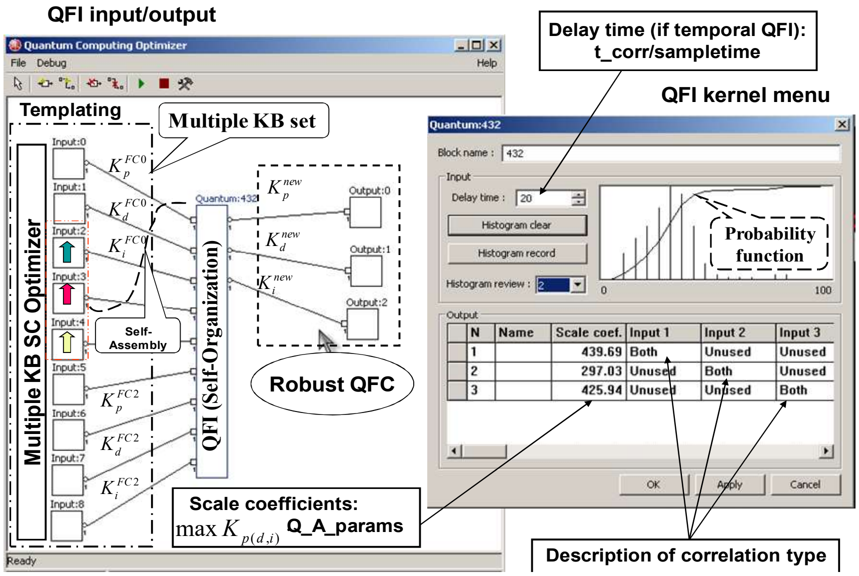
Figure 8.Internal structure of “correlation parameters” block

2.2. 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:

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

In this part we have introduced briefly the particularities of quantum computing and quantum information theory that are used in the quantum block QFI (Fig. 9) supporting a self-organizing capability of FC in robust ICS.



Block name : 432

Delay time : 20

Histogram review : 2

N	Name	Scale coef.	Input 1	Input 2	Input 3
1		439.69	Both	Unused	Unused
2		297.03	Unused	Both	Unused
3		425.94	Unused	Unused	Both

Figure 9.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.

3. Quantum Computing Optimizer of Knowledge Base (QCOptKB^{TM})

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.

3.1. 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 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 changed at any moment. Once changed, locale parameters saved in model and will used for future processing of data. Locale setting affects reading and writing of text data files and model files.

3.2 QCOptimizer menu

Main module of Quantum Computing Optimizer is a Win32 application, called qcowin.exe. It can 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.

The main program window of the QCOptimizer shown on Figure 10.

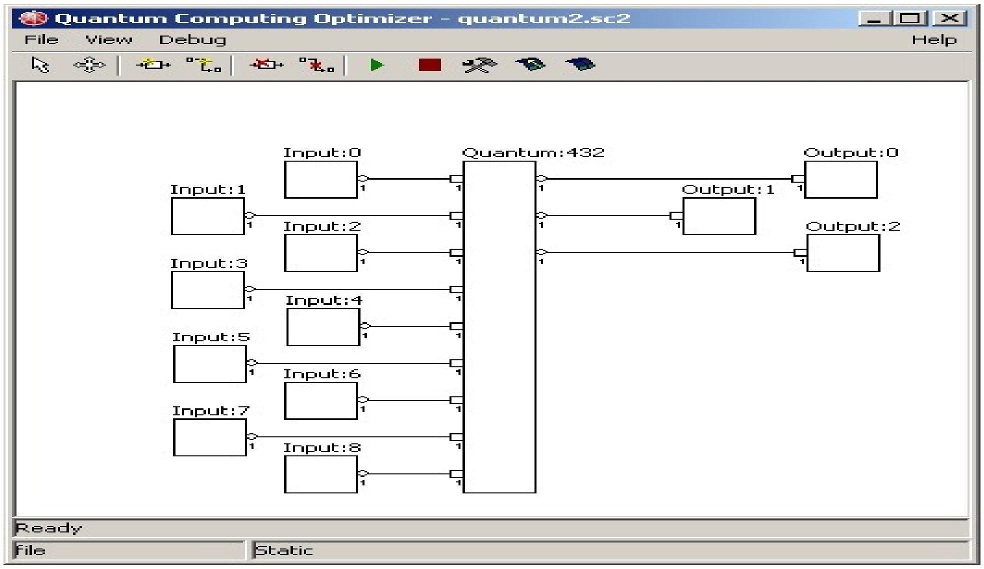


Figure 10. QCOptimizer program window

QCOptimizer window 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.
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 described in detail in next sections.
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.

FILE Menu

QCOptimizer allows project files 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.

QCOptimizer saves project in special files: *model files* (with ***.sc2** extension). *Model file* 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 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.

VIEW Menu

View menu can used to open signal view windows:
Open signal view: displays new signal window.

HELP Menu

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

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 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 use to return to one of the previous pages, if you want to change some parameters. ***CANCEL*** button aborts command execution.

4. Working with QCOptimizer

4.1. 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.

4.2. 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.

4.3. Saving model files

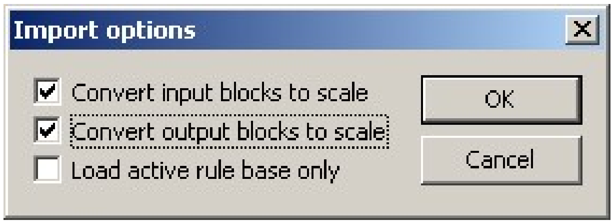
You can use ***File/Save*** command to save you model files. Standard windows ***File Save*** dialog box will appear. Select file with you model and press ***Save*** button.

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.

4.4. Importing SCOptimizer files

You can import SCOptimizer model files (.sco extension) in order to use fuzzy control systems developed in SCOptimzier 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 Fig. 11.



Import options

☑ Convert input blocks to scale

☑ Convert output blocks to scale

□ Load active rule base only

Figure 11.Import options dialog

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

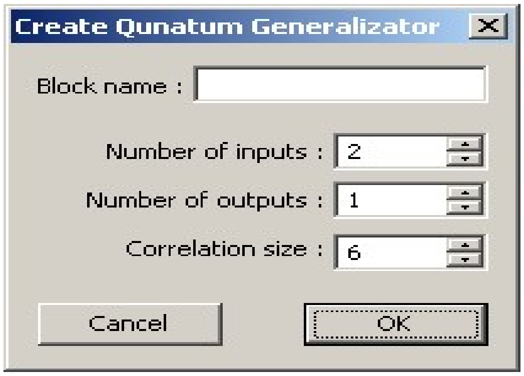
Convert input blocks to scale: by default, imported SCOptimizer model includes new input blocks that are used to feed input data to fuzzy control system. If this option is selected the model will be imported with scale blocks instead, which will use input normalization parameters to scale signal. This 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 connected to output blocks.

4.5. 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 (Fig. 12).



Create Qunatum Generalizator

Block name :

Number of inputs : 2

Number of outputs : 1

Correlation size : 6

Figure 12. Quantum block creation dialog

4.7. 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 auto detected. If the file does not look like Matlab file than it assumed to be a text file.

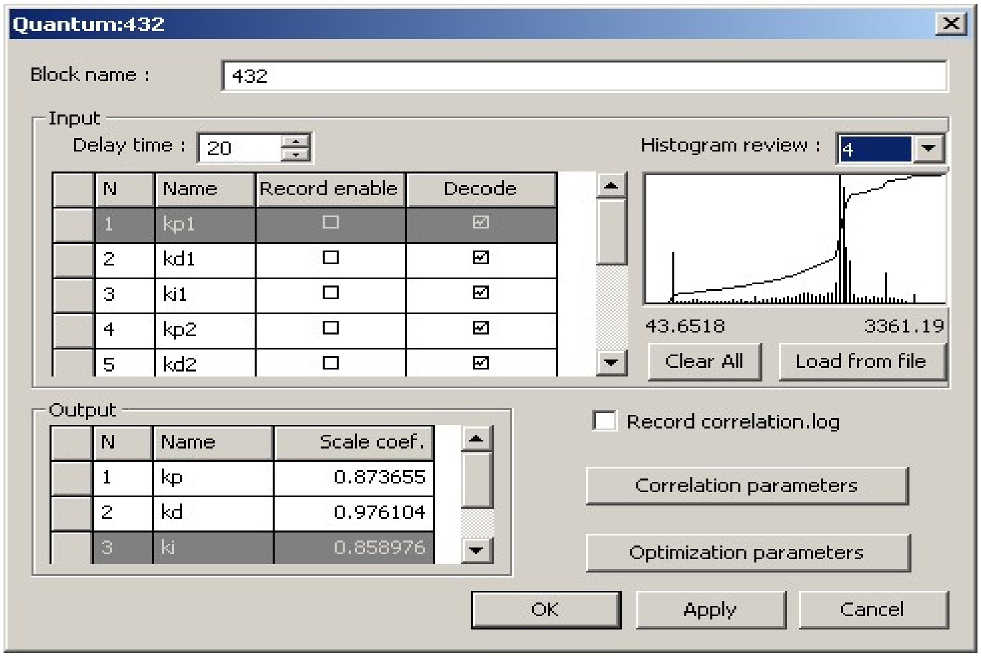
Matlab file should contain an array of real numbers with number of columns equal to sum of number of input and output variables of the model. If version file contains several variables then first one will be loaded.Text files are processed using *locale settings*. Please set *locale settings* as described in section 0 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 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.The main block in the SCOptimizer toolkit is a quantum generalization, which realizes QFI method described in section 1.1.3.

5. Quantum Generalization Blocks

During creation of quantum generalization block, you should enter initial parameters in the dialog box shown on Figure12.

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

Configuration of the block can be checked and changed by double clicking on the block. The dialog box as shown on Fig. 13 will appear.



Quantum:432

Block name : 432

Input

Delay time : 20

Histogram review : 4

N	Name	Record enable	Decode
1	kp1	□	☑
2	kd1	□	☑
3	ki1	□	☑
4	kp2	□	☑
5	kd2	□	☑

Output

N	Name	Scale coef.
1	kp	0.873655
2	kd	0.976104
3	ki	0.858976

□ Record correlation.log

Figure 13. 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:

- ***Name***: input name, to be used in correlation configuration;
- ***Record enable***: check to enable histogram recording for selected channel;
- ***Decode***: uncheck to skip this input during decoding (required by QPID algorithm).

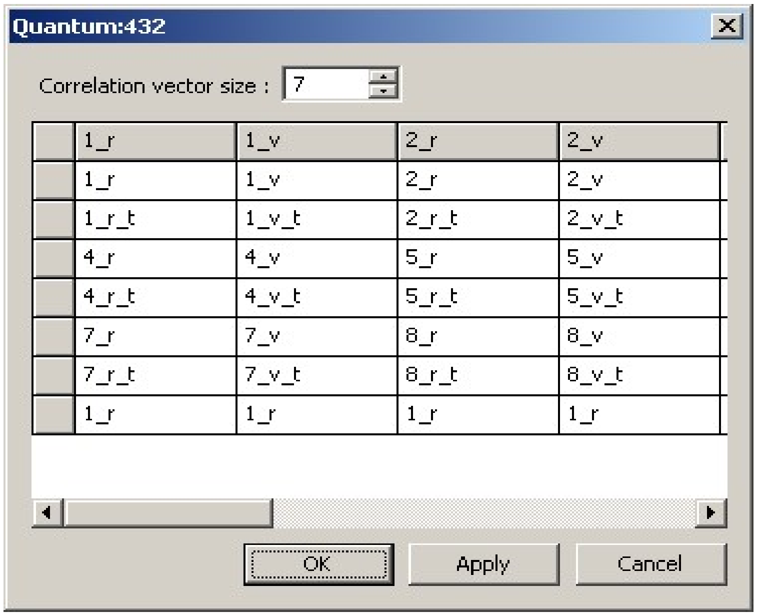
To the left of table there is a histogram review window. It shows histogram for the channel selected in the drop-down list above it. ***Clear All*** button can used to erase all recorded histograms for all channels.

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:

- ***Name***: output name, to be used in correlation configuration;
- ***Scale coefficient***: output scaled to this constant.

Quantum correlation can configured by pressing ***Correlation parameters*** button (Fig. 14).



Correlation vector size : 7

	1_r	1_v	2_r	2_v
	1_r	1_v	2_r	2_v
	1_r_t	1_v_t	2_r_t	2_v_t
	4_r	4_v	5_r	5_v
	4_r_t	4_v_t	5_r_t	5_v_t
	7_r	7_v	8_r	8_v
	7_r_t	7_v_t	8_r_t	8_v_t
	1_r	1_r	1_r	1_r

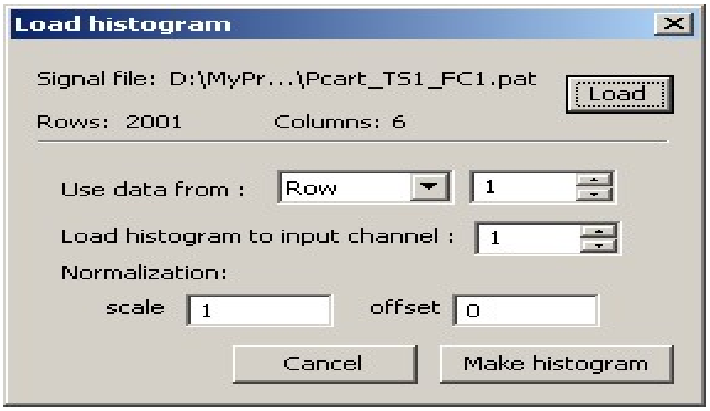
Figure 14. Quantum correlation parameters dialog

Input field at the top part of window allow changing correlation vector size. Table below lists current correlation 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 used at which position of a vector. Those values can 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 not specified), and modifier, as follows:

- _r: real component is used;
- _r_t: real delayed component used;
- _v: virtual component used;
- _v_d: virtual delayed.

Quantum algorithm requires the histogram to be loaded for each channel. Histograms can 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 checked table lines will recorded in histogram. To load histogram press "Load from file button" (Fig. 15).



Signal file: D:\MyPr...\Pcart_TS1_FC1.pat

Rows: 2001 Columns: 6

Use data from : Row 1

Load histogram to input channel : 1

Normalization:

scale 1 offset 0

Figure 15. Histogram load dialog

In the dialog appear first press ***Load*** button and select file used. File name and number of rows and columns will displayed. Then select if the data used 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.

5.1. 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.

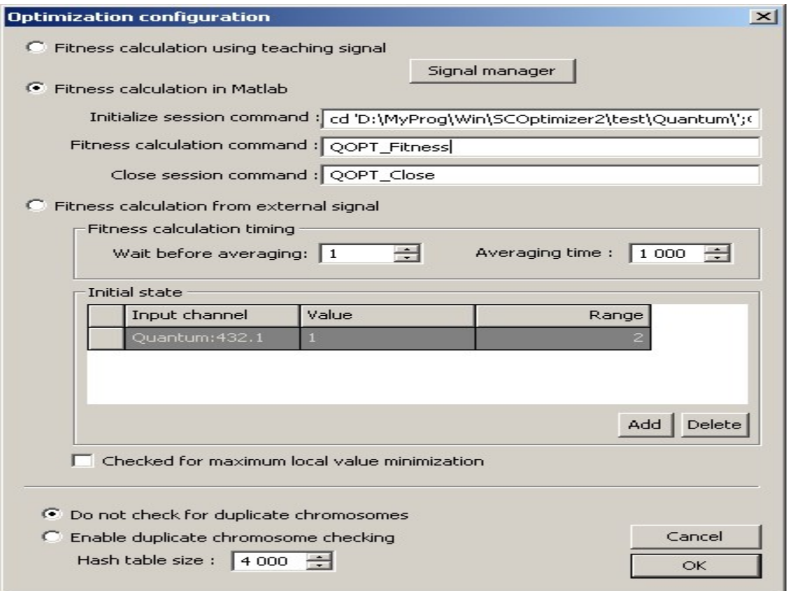
Once those two actions done, you should see "file" displayed in the status line typically used to show external link name. Press ▶ toolbar button to perform a full run through data available in loaded files. You can view resulting signals using signal view windows, or you can use this mode to record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

5.2. 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.

5.2.1. Selecting optimization options

To set optimization settings press toolbar button (Fig. 16).



Optimization configuration

○ Fitness calculation using teaching signal

● Fitness calculation in Matlab

Initialize session command : cd 'D:\MyProg\Win\SCOptimizer2\test\Quantum\';

Fitness calculation command : QOPT_Fitness

Close session command : QOPT_Close

○ Fitness calculation from external signal

Fitness calculation timing

Wait before averaging: 1

Averaging time : 1 000

Initial state

	Input channel	Value	Range
	Quantum:432.1	1	2

□ Checked for maximum local value minimization

● Do not check for duplicate chromosomes

○ Enable duplicate chromosome checking

Hash table size : 4 000

Figure 16. Optimization settings

First select desired fitness calculation mode:

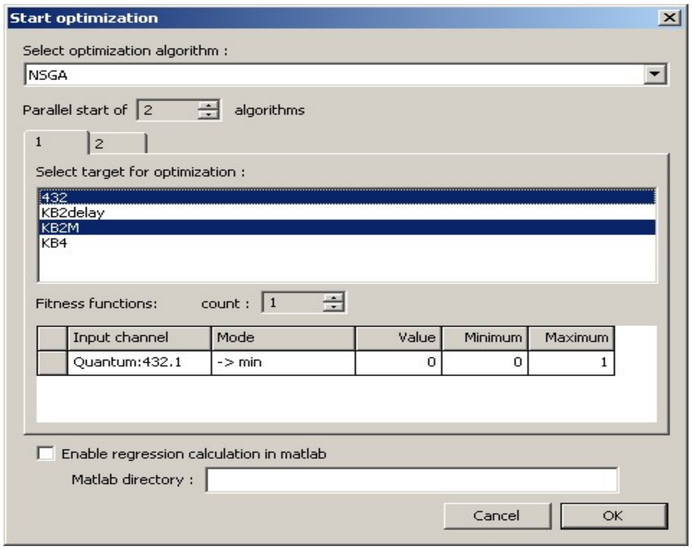
- Teaching signal: model is evaluated using teaching signal loaded in teaching signal manager and fitness computed by comparing actual data from model with data selected as teaching signal.
- Fitness calculation in matlab: During optimization, QCOptimizer set model to the state evaluated and calls matlab 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:
    - Initialize session command: called by QCOptimizer when optimization started. Should perform initialization, for example load some simulink model.
    - Fitness calculation command: called by QCOptimizer to calculate fitness for current state.
    - Close session command: called when optimization is finished. May free resources allocated for calculations.
- Fitness calculation from external signal. Signal coming from external link used for model input and fitness calculation. Concrete fitness function 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 2 will equal. Once initial condition met QCOptimizer will wait for delay time set in wait before averaging field and then 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 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.

5.2. Starting optimization

To start optimization press button. Optimization start dialog will appear (Fig. 17).



Start optimization

Select optimization algorithm : NSGA

Parallel start of 2 algorithms

Select target for optimization :

- 432
- KB2delay
- KB2M
- KB4

Fitness functions: count : 1

	Input channel	Mode	Value	Minimum	Maximum
	Quantum:432.1	-> min	0	0	1

☐ Enable regression calculation in matlab

Matlab directory :

Figure 17.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.

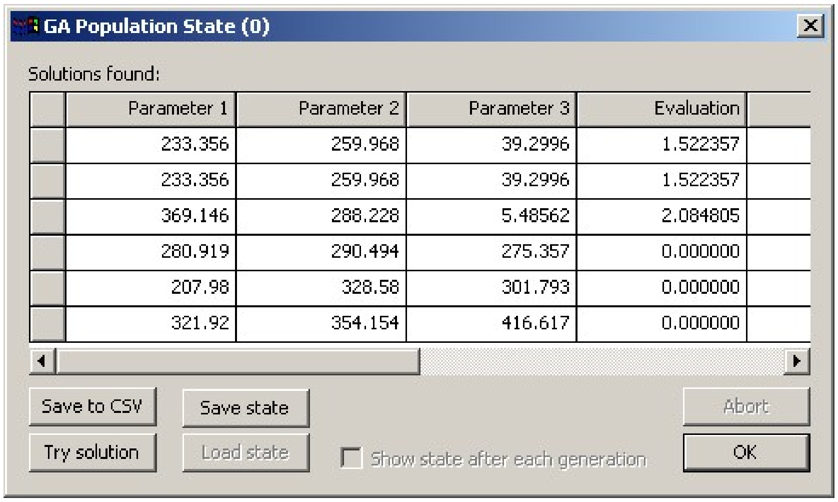
Once you press OK you will be prompted to enter algorithm parameters (depending on algorithm selection). When you confirm algorithm settings optimization will start.

5.3. Optimization progress/evaluation

For each optimization currently active, it displays a line, which shows 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 aborted with some delay, if there is a calculation, which cannot stopped immediately. Press ***Pause calculations*** to pause all calculations. Current evaluations will restarted from beginning after calculation resumed.

When calculation is over (or after each generation if you select corresponding option in algorithm architecture,) the dialog will appear showing state of current population (Fig. 19).

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 ***Trysolution*** to set model state to chromosome selected, so that you can evaluate it manually.



	Parameter 1	Parameter 2	Parameter 3	Evaluation
	233.356	259.968	39.2996	1.522357
	233.356	259.968	39.2996	1.522357
	369.146	288.228	5.48562	2.084805
	280.919	290.494	275.357	0.000000
	207.98	328.58	301.793	0.000000
	321.92	354.154	416.617	0.000000

Figure 19. Optimization progress dialog

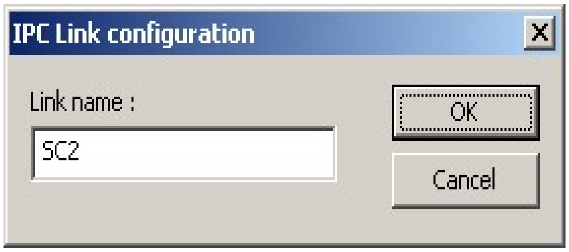
If this dialog shown after optimization is over, you should select a chromosome, which will used as optimization result (by default best one shown in top line is used). When you are finished this chromosome will applied to model.

6. External link

QCOptimizer support data exchange with external programs, which can 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 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 changed by pressing ⚒ toolbar button. Dialog box will appear where you can enter desired link name (Fig. 20).



IPC Link configuration

Link name : SC2

Figure 20. Link name configuration dialog

Current link name 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.

Selection of development toolkit.

Simulator access is as simple as possible and it implemented as a non-typical web application. The diagram of the sequence of the user's work with the system and the interaction of the model, the presentation and the template presented on Fig. 21.

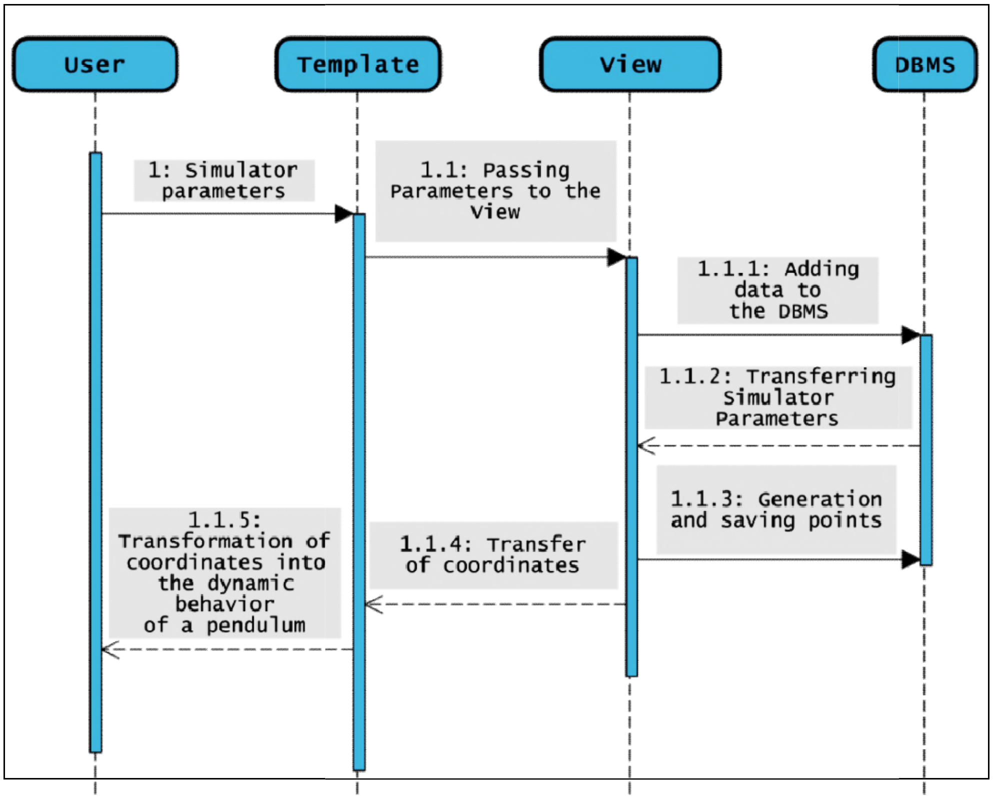

User
Template
View
DBMS
1: Simulator parameters
1.1: Passing Parameters to the View
1.1.1: Adding data to the DBMS
1.1.2: Transferring Simulator Parameters
1.1.3: Generation and saving points
1.1.4: Transfer of coordinates
1.1.5: Transformation of coordinates into the dynamic behavior of a pendulum

Figure21. Sequence diagram of system

Most of the server-side work is math. It is necessary to calculate the position of the carriage, the angle of inclination of the pendulum in space. For this reason, Python and the Django framework, which implements the model-view-controller (MVC) approach, chosen as the programming language (or in Django, this is the model-view-template (MVT)). MySQL used to store all data, and the architecture has developed for adding Redis to be faster, if the MySQL operation speed is insufficient.

Figure 22 show Benchmark results of quantum intelligent control simulation of “cart - pole” system with QGA (box for the type choice of “Quantum correlation” on Fig. 2).

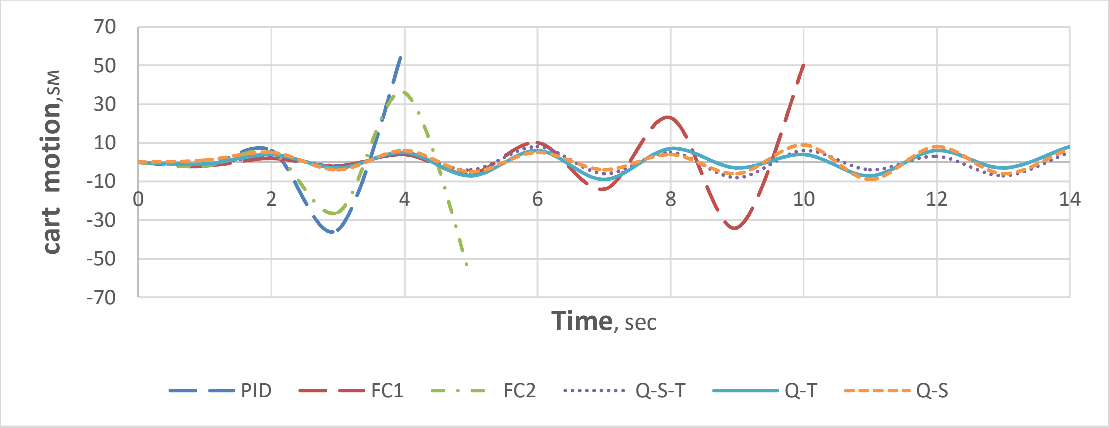


Figure22. Cart behavior (Q-S-T – quantum spatial-temporal correlation; Q-T – quantum temporal correlation; Q-S – quantum spatial correlation; FC – fuzzy controller)

On Fig. 23 are demonstrated results of simulation and experimental results comparison.

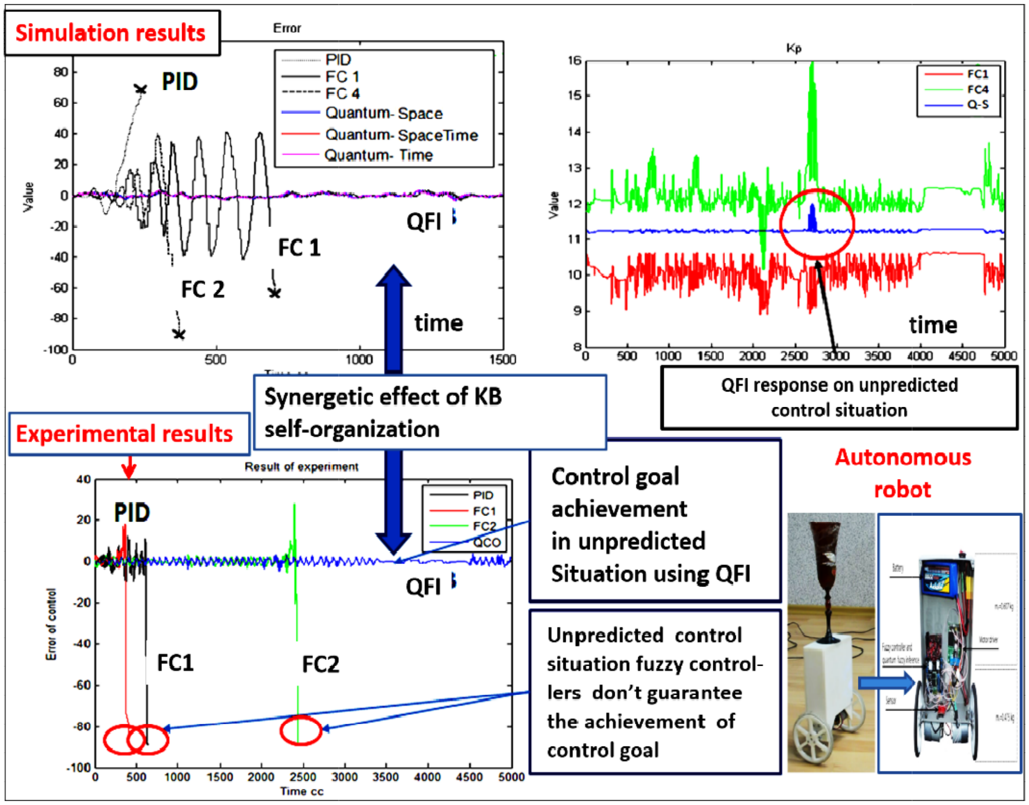


Figure23. Simulation & experimental results comparison for unpredicted control situation in cases of PID-controller, fuzzy controller and QFI-controller (b)

Mathematical modeling and experimental results received for the case of unpredicted control situation and knowledge base of fuzzy controller was designing with SW of QCOPTKB^{TM} for teaching signal measured directly from control.

As model of unpredicted control situation on Fig. 3 (Box Z^{-1}) was the situation of feedback sensor signal delay on three times. Results of controller's behavior comparison confirm the existence of synergetic self-organization effect in the design process of robust KB on the base of imperfect (non-robust) KB of fuzzy controllers on Fig. 3. In unpredicted control situation, control error is dramatically changing and KB responses of fuzzy controllers (FC 1 and FC 2) that designed in learning situations with soft computing are *imperfect* and do not can achieve the control goal.

Using responses of imperfect KB (as control signals for design the schedule of time dependent coefficient gain in PID-controller on Fig. 3) in Box QFI the robust control formed in on line. This effect is based on the existence of additional information resource that extracted by QFI as quantum information hidden in classical states of control signal as response output of imperfect KB's on new control error (QFI algorithm structure on line in Fig. 2). Classically this effect impossible to achieve.

QGA in Fig. 23 for this case recommended the spatial quantum correlation [7] as was early received in [8].

CONCLUSIONS

- Ideas of quantum computation to realize control algorithm to design a robust intelligent control system, when it can operate not only in teaching control conditions but also in unpredicted control situations too applied.
- Quantum control algorithm of knowledge base self-organization based on a special form of *quantum fuzzy inference* relative to quantum knowledge extraction from a few of Knowledge Bases designed by SC Optimizer tools [1].
- Description of QFI model and QCOptimizer tools supremacy realized this method demonstrated.
- The results demonstrate the effective application possibility of end-to-end quantum technologies and quantum computational intelligence toolkit based on quantum soft computing for the solution of intractable classical and algorithmically unsolved problems as design of global robustness of ICS in unpredicted control situations and intelligent robotics [10-15].
- New circuit implementation design method of quantum gates for fast classical efficient simulation of quantum algorithms is developed. Benchmarks of design application as Grover's QSA and QFI based on QGA demonstrated.
- Applications of QAG approach in intelligent control systems with quantum self-organization of imperfect knowledge bases described on concrete examples.

References

[1] NikolaevaA.V., Ulyanov S.V., Litvintseva L.V., Ulyanov V.S. Deep Machine Learning SW for Intelligent Control. Part I: Soft computing KB optimizer supremacy. Intern. J. Soft ware Engineering and Applications. – 2020 (in print)

[2] Ulyanov, S. V. System and method for control using quantum soft computing. 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. US Patent No 2006/0224547 A1, 2006.

[3] Ulyanov, S. V. Quantum fast algorithm computational intelligence PT I: SW / HW smart toolkit. Artificial Intelligence Advances, No 1, 2019: pp.18-43. DOI:https://doi.org/10.30564/aia.v1i1.619.

[4] Nielsen, M. A., Chuang, I. L. Quantum computation and quantum information. UK, Cambridge, University Press, 2000: 700 p.

[5] Marinescu, D. C., Marinescu G. M. Approaching quantum computing. New Jersey: Pearson Prentice Hall, 2005: 400 p.

[6] Ulyanov S.V.: Self-organizing quantum robust control methods and systems for situations with uncertainty and risk. Patent US 8788450 B2. 2014.

[7] Ulyanov S.V. Quantum fuzzy inferencebased on quantum genetic algorithm: Quantum simulator in intelligent robotics. R. A. Aliev et al. (Eds.): ICSCCW 2019, Springer AISC 1095, pp. 1–8, 2020. https://doi.org/10.1007/978-3-030-35249-3_9.

[8] Ulyanov S. V. Self-Organized Intelligent Robust Control Based on Quantum Fuzzy Inference. Recent Advances in Robust Control - Novel Approaches and Design Methods, InTech, 2011. ISBN: 978-953-307-339-2.

[9] Ulyanov S.V.:Quantum soft computing in control processes design: Quantum genetic algorithms and quantum neural network approaches. In: R. Aliev (Ed.) WAC (ISSCI') 2004 (5th Intern. Symp. on Soft Computing for Industry), Seville Spain, 2004. vol. 17. pp.99-104. Springer, Heidelberg.2004.

[10] Ulyanov S.V.: Self-organized robust intelligent control - Quantum Fuzzy Inference in Unpredicted / Hazard Environments and Quantum Soft Computational Intelligence Toolkit in MatLab. LAP Lambert Academic Publishing. 2015.

[11] Ulyanov S.V., Albu V. A. et all: Quantum Algorithmic Gates: Information Analysis & Design System in MatLab. LAP Lambert Academic Publishing, Saarbrücken. 2014.

[12] Ulyanov S.V., Albu V. A. et all: Design IT of Quantum Algorithmic Gates: Quantum search algorithm simulation in MatLab. LAP Lambert Academic Publishing, Saarbrücken. 2014.

[13] Lahoz-Beltra R.: Quantum genetic algorithms for computer scientists. Computers. 2016. Vol. 5. No 4. Pp. 31-47.

[14] Litvintseva L.V., Ulyanov S.V. Quantum fuzzy inference for knowledge base design in robust intelligent controllers. J. of Computer and Systems Sci. Intern. 2007. Vol. 46. No 6. Pp.908–961.

[15] US Patent No 6,578,018 B1: System and method for control using quantum soft computing (Inventor: S.V.Ulyanov) (2003); US Patent No 7,383,235 B1. 2003; EP PCT 1 083 520 A2. 2001.