A Multi-objective Optimization based on Hybrid Quantum Evolutionary Algorithm in Networked Control System

QU Zheng-geng a,b, ZHANG Xiao-yan a,b

a College of Information Science and Technology, Northwest University, Xi’an, China
b Department of Computer Science, Shangluo University, Shangluo, China

Abstract

This paper proposes a new hybrid quantum clone evolutionary algorithm (HQCEA) in a two-layer networked learning control system (NLCS) architecture. This special architecture achieves better control performance, better interference rejection and increasing the adaptability to varying environment. The proposed scheduling algorithm HQCEA optimizes the network transmission period which increases the diversity of the solution space of functions and avoid trapping into local peak effectively. As thus, network resources are allocated reasonably to reduce delays and dropped packets, improving network utilization with communication constraints. According to the simulation results, the HQCEA overcomes the shortcoming of the traditional QEA, and can deal with the continuous functions with multi-peak and complex plant successfully in a shorter time.

© 2012 Published by Elsevier B.V. Selection and/or peer-review under responsibility of Garry Lee

Keywords: Multi-objective Optimization, Hybrid Quantum Evolutionary Algorithm, Networked Control System

1. Introduction

The introduction of the network complicates the analysis and synthesis problems of control systems [2]. Network-induced delays [3-4] and packet losses [5] are two essential issues that need careful consideration in an NCS design. There is no guarantee for zero delay or even random long delay when a sensor sends messages to a controller or a controller sends control signals to a sensor. Some packets are dropped to either reduce the queue size in the path or to inform the senders to reduce their transmission rate when there is congestion in the communication network. The synchronization of different sensors, actuators and controllers is another problem in NCSs. In real-time systems, particularly control systems, these problems are catastrophic or even cause system instability. Therefore, the quality for performances of network service (QoS) and control system (QoC) rely on not only the design of system architecture and control algorithm, but also the scheduling of network information to reduce information transmission collision and implement resource allocation of network nodes [6].
2. The Scheduling Optimization of Networked Learning Control System

2.1. Problem description

In this paper, two-layer networked learning control system architecture [7-8] is introduced as shown in Fig.1, with the objectives to achieve better control performance, better interference rejection and to increase the adaptability to varying environment. Ci, Ai and Si is a controller, actuator and sensor respectively.

![Fig.1. Two-layer networked learning control system (NLCS)](image)

2.2. The Scheduling of Networked Learning Control System

Multi-objective optimization method dynamically adjusts the sampling of each loop, and using the possible smallest bandwidth requirements to achieve optimal control performance. The relation between bandwidth $b_i$ and sampling period $t_i$ for each control loop is given by (1), where $m_j$ is the time spent on the messaging required to perform each closed loop operation (which may include data exchange from sensor to controller and from controller to actuator, as well as the time spent to execute the controller).

$$b_i = \frac{m_j}{t_i}$$

Without loss of generality, for all axes, if the equilibrium point (or reference signal) is considered to be zero, the Euclidean norm of the state vector $x_i = (x_{i,1}, x_{i,2}, \ldots, x_{i,n})^T$ in the $\mathbb{R}^n$ vector space is the distance that measures how far each axis is from its equilibrium point at any given time $t > 0$. This measure (also called error (2)) is defined as the feedback information that each controller, at each sample, will forward to the bandwidth manager for the run-time bandwidth allocation.

$$e_i = \| x_i \|$$
In fact, in the range of the permissible variable sampling period (the boundary of dynamic bandwidth changes) and no overload network, approximate linear relationship between system IAE and the sampling period could be like:

$$J(t_i) = \alpha_i + \beta_i t_i \quad (3)$$

$n$ control loops being allocated network bandwidth are $b_1, b_2, \ldots, b_n$. Multi-objective optimization problem to minimize bandwidth consumption and optimize the performance of system control in bandwidth-constrained NLCS is described as:

$$\min J_1 = \sum_{i=1}^{n} (\alpha_i + \beta_i m_i / t_i) \quad (4)$$

$$\min J_2 = \sum_{i=1}^{n} b_i \quad (5)$$

subject to

$$C^T b \leq U_d \quad (6)$$

$$b_i^{\text{min}} \leq b_i \leq b_i^{\text{max}} \quad (7)$$

Where (4) describes the optimization problem of system control performance, $m_i$ is the information transmission time, and the parameters $\alpha_i$ and $\beta_i$ depend on the controller and plant in the corresponding control loop. Due to the calculation of the gradient in solving optimization problems, $\alpha_i$ could be ignored, only finding the curvature $\beta_i$. (5) describes the optimization problem of bandwidth consumption, aiming at minimize demand for bandwidth under the best system performance. Consequently, the excess of limited resources is available to the rest of application nodes. (6) is global availability of bandwidth in NLCS, where $U_d$ is applicable global bandwidth resource and $b = [b_1, b_2, \ldots, b_n]^T$, $C = [1,1,\ldots,1]^T$. (7) is the allowing variation range of dynamic allocation of bandwidth for each control loop, namely, ensuring system stability of the determined upper and lower bounds for sampling period. The corresponding control loop parameters of upper and lower bounds can be found by exploiting resource-constrained conditions and Maximum Allowable Delay Bound (MADB) method.

We use Rights Law here compounding (4) and (5) to the following single-objective function for the problem solving of multi-objective optimization:

$$\min J = \sum_{i=1}^{n} (\alpha_i + \beta_i m_i / t_i) + \gamma \sum_{i=1}^{n} b_i \quad (8)$$

Where $\gamma$ is the weight coefficient, which balance the objective functions $J_1$ and $J_2$.

To solve the above optimization problems, we convert constrained optimization into unconstrained one, establishing the following Lagrange function:
According to Karush-Kuhn-Tucker (KTT) conditions, if \( b^* = [b_1^*, b_2^*, \cdots, b_n^*]^T \) is the optimal solution of the optimization problem, then

\[
\nabla J(b^*) + \lambda_a - \lambda_b + \lambda C = 0
\]

\[U_d - C^T b^* \geq 0, \quad b_i^{\min} \leq b_i^* \leq b_i^{\max}, \quad i = 1, 2, \cdots, n\]

\[
\lambda(U_d - C^T b^*) = 0, \quad \lambda \geq 0, \quad \lambda_a \geq 0, \quad \lambda_b \geq 0
\]

Where \( \nabla J \) is gradient vector, \( \lambda = [\lambda_1, \lambda_2, \cdots, \lambda_n]^T \) and \( \lambda = [\lambda_{n+1}, \lambda_{n+2}, \cdots, \lambda_{2n}]^T \) are Lagrange multipliers.

In order to further saving more limited network resources, some expert knowledge is applied to corresponding set of rules as part of the constraints. Control loops require only less bandwidth to maintain the transmission of information near the equilibrium point, using the rules described as: if \( e_i \leq e_i^{\text{th}} \), then \( b_i = b_i^{\min} \), \( i = 1, 2, \cdots, n \), where \( e_i^{\text{th}} \) expresses a sufficiently small error thresholds of the \( i \) th control loop. Thus, the optimization problem with described rules becomes:

\[
\min J = \sum_{i=1}^{n} (\alpha_i + \beta_i m_i / t_i) + \gamma \sum_{i=1}^{n} b_i
\]

subject to

\[C^T b \leq U_d\]

\[b_i^{\min} \leq b_i \leq b_i^{\max}\]

\[b_i = b_i^{\min}, \quad \text{if} \quad e_i \leq e_i^{\text{th}}\]

(11 12 13 14)

Obviously, the conventional mathematical programming is very difficult to solve the optimization problems except some heuristic modern optimization algorithms (such as genetic algorithms, etc.). In this paper, HQCEA will be given to solve the multi-objective scheduling optimization problem in NLCS.
3. The HQCEA-based Multi-objective Scheduling Optimization (MOSO) scheme

3.1. The Description of the HQCEA

The basic unit of information in quantum computation [9-11] is the qubit. A qubit is a two-level quantum system and it can be represented by a unit vector of a two dimensional Hilbert space \((\alpha, \beta \in \mathbb{C})\):

\[
|\psi\rangle = \alpha |0\rangle + \beta |1\rangle \quad , \quad |\alpha|^2 + |\beta|^2 = 1
\]  

(15)

where we denote with \(|0\rangle\) and \(|1\rangle\) the basis states, adopting the ket notation for quantum state vectors. A two-level quantum system is described by a superposition of the basis states, whereas a two-level classical system can be just in one of the basis states 0 or 1.

The state of a qubit can be changed by the operation with a quantum gate. Inspired by the concept of quantum computing, HQCEA is designed with a novel Q-bit representation, a Q-gate as a variation operator, and an observation process. A Q-bit individual as a string of n Q-bits is defined as:

\[
q_j = \begin{bmatrix}
\alpha_{j1} & \alpha_{j2} & \cdots & \alpha_{jm} \\
\beta_{j1} & \beta_{j2} & \cdots & \beta_{jm}
\end{bmatrix}
\]  

(16)

Where \(j = 1, 2, \cdots, n\).

In this paper, HQCEA is made up of these factors differ from traditional quantum evolutionary algorithm as follows:

HQCEA can make sub-swarms distribute around father individual’s search space, which can increase the diversity of the solution space of functions and avoid trapping into local peak effectively. So the algorithm can get balance between depth search and breadth search by doing this and improve its ability to solve the function.

The HQCEA scale of cope is dynamic while the traditional one is fixed in the phase of clone copy.

\[
N_i^j = \text{round}(n - \frac{f(p_i^j)}{\sum_{j=1}^{n} f(p_j^j)})
\]  

(17)

Where \(j = 1, 2, \cdots, n\) and \(N\) is the scale of clone of each individual in the \(t\) th iterative time, and the scale of clone of the swarm is:

\[
N_i^j \approx \text{round}(nN_i^j) = \text{round}(n \sum_{i=1}^{n} \frac{f(p_i^j)}{\sum_{j=1}^{n} f(p_j^j)}) = n
\]  

(18)

3.2. Procedure HQCEA for the Optimization of Continuous Functions with Multi-peak

Every individual of each chromosome in this HQCEA will make its own dynamic clone to build its new sub-swarm; then every new chromosome will be mutation in its low bit; At last, the HQCEA will
update the whole swarm by using random strategy. The procedure begins at $t = 0$, initializing $Q(t) = \{q_1, q_2, \ldots, q_n\}$ and define an empty memory storeroom. By observing the states of $Q(t)$, we make $R(t)$, which is cloned the individual and get a new swarm $R^*(t)$. After that, each sub-swarm makes mutation in their low bit. We evaluate $R^*(t)$ and store the best solutions in memory storeroom. While the termination is false, the swarm is updated by Q-gate to get the new swarm. If $\varepsilon$ has a little changed, we update the swarm randomly. Setting $t = t + 1$, the procedure of HQCEA loops again.

In this paper, HQCEA use the appropriate Q-gate, by which operation the Q-bit should be updated as the step of mutation. The following rotation gate is used as a Q-gate in HQCEA, such

$$
U(\Delta \theta) = \begin{pmatrix}
\cos(\Delta \theta) & -\sin(\Delta \theta) \\
\sin(\Delta \theta) & \cos(\Delta \theta)
\end{pmatrix}
$$

(19)

$$
\begin{bmatrix}
\alpha_i^* \\
\beta_i^*
\end{bmatrix} = U(\Delta \theta_i) \begin{bmatrix}
\alpha_i \\
\beta_i
\end{bmatrix}
= \begin{pmatrix}
\cos(\Delta \theta_i) & -\sin(\Delta \theta_i) \\
\sin(\Delta \theta_i) & \cos(\Delta \theta_i)
\end{pmatrix} \begin{bmatrix}
\alpha_i \\
\beta_i
\end{bmatrix}
$$

(20)

$$
\theta_i = s(\alpha_i, \beta_i) \times \Delta \theta_i
$$

(21)

Where $\Delta \theta$ is a rotation angle of each Q-bit towards either 0 or 1 state depending on its sign.

In (19), Q-gate has a parameter $\Delta \theta$, which can get as follows:

Table 1. The rotation angle strategy for scheduling in NLCS

| $x_i$ | $b_i$ | $f(x) > f(b)$ | $\Delta \theta$ | $s(\alpha_\beta)$ |
|-------|-------|---------------|-----------------|-----------------|
| 0     | 0     | FALSE         | 0               | $\alpha \beta > 0$ |
| 0     | 0     | TRUE          | 0               | $\alpha \beta < 0$ |
| 0     | 1     | FALSE         | 0.05\pi         | $\alpha = 0$     |
| 0     | 1     | TRUE          | 0.05\pi         | $\beta = 0$     |
| 1     | 0     | FALSE         | 0.02\pi         | +1              |
| 1     | 0     | TRUE          | 0.02\pi         | -1              |
| 1     | 1     | FALSE         | 0               | 0               |
| 1     | 1     | TRUE          | 0               | 0               |

As table1 illustrated, where $x_i$ is the i th bit of current individual, $b_i$ is the best individual’s i th bit in current swarm, $f(*)$ is fitness function. $\Delta \theta$ is the quantum rotation angle’s value, which can control the convergence scope of the algorithm. Different rotation angle will induce different results. $s(\alpha_\beta)$ is the direction of rotation angle, which can control the convergence speed of the algorithm. Quantum
rotation angle is very important for the algorithm. If the angle is too small, which will decrease the speed of
the algorithm, contrarily, which will conduce prematurity. So we define the angle is in $[0.01\pi, 0.05\pi]$.

4. Simulation results

In this section, we present the results of a simulation of the HQCEA in NLCS in order to show the
validity of the algorithm. We run program on the computer of Intel(R) Core(TM) 2 Quad CPU and 4G
DDR3 RAM, compiled environment of Matlab 7.0. The following simulation plant is based on [9]. The
fitness function is

$$f = \frac{1}{J}$$  \hspace{1cm} (22)

The parameters of HQCEA and control system are listed: Every swarm contains 50 individuals, and
each individual is encoding by 20 bits. The global availability of bandwidth in NLCS is 10, and the
termination generation is 40.

To illustrate improving level of quality of control (QoC) and bandwidth requirements, we compare
HQCEA with non-optimizing and the traditional quantum evolutionary algorithm (QEA).

When the control loops with different random perturbations cycle, the integral absolute error (IAE) and
mean network utilization (MNU) of three different strategies are as showed in the table 2. The transmission
periods are more close to each other adopting the HQCEA-based scheduling approach. The transmission
error of the system becomes smaller. These show that the HQCEA save as much as possible the limited
network resources and make the system control performance further optimized.

| $T$ (s) | Non-optimizing | Traditional QEA | HQCEA |
|--------|----------------|-----------------|-------|
|        | IAE            | MNU             | IAE   | MNU             | IAE   | MNU             |
| 10     | 17.564         | 40.01%          | 16.012| 28.88%          | 15.976| 25.03%          |
| 20     | 17.456         | 39.99%          | 15.966| 28.56%          | 15.952| 24.52%          |
| 50     | 17.625         | 40.00%          | 15.962| 29.06%          | 15.261| 24.24%          |
| 100    | 16.154         | 39.96%          | 15.161| 29.11%          | 13.815| 24.17%          |
| 150    | 16.012         | 39.98%          | 13.648| 29.20%          | 12.268| 24.03%          |
| Total  | 84.811         | 39.99%          | 76.749| 28.96%          | 73.272| 24.40%          |

To sum up, hybrid quantum clone evolutionary algorithm-Based Scheduling Optimization achieves the
desired objectives in the resource-constrained NLCS.

5. Conclusions

A networked control system (NCS) is the control system whose sensors, actuators and controllers are
closed over communication networks [1], becoming the focus of many recent researches both in control of
academic and control of applications. A NCS has been widely used in control field due to various
advantages including modular system design with greater flexibility, low cost of installation, powerful
system diagnosis and ease of maintenance.
References

[1] Marieke B. Nathan V. Stability of Networked Control Systems With Uncertain Time-Varying Delays, IEEE Trans. on Automatic Control, Vol.54, No.7, 1575-1580, 2009.

[2] Junlin Xiong and James Lam, Stabilization of Networked Control Systems With a Logic ZOH, IEEE Trans. on Automatic Control, Vol.54, No.2, 358-363, 2009.

[3] M. Liu et al., Stabilization of Markovian jump linear system over networks with random communication delay, Automatica, 45, 416–421, 2009.

[4] Zhang, W. Yu, L., A robust control approach to stabilization of networked control systems with time-varying delays, Automatica, 5, 1-6, 2009

[5] Wang, Y., Residual generation and evaluation of networked control systems subject to random packet dropout, Automatica, 6, 1-8, 2009.

[6] Xiangfeng Zhang, Zhijie Wang, Zhaojia Liang, An Immune-Genetic Algorithm-Based Scheduling Optimization in a Networked Control System, IEEE Global Congress on Intelligent Systems, 2009.

[7] Dajun Du, Minrui Fei, Hu Huosheng, Li Lixiong, Two-layer networked learning control using self-learning fuzzy control algorithms, Chinese Journal of Scientific Instrument, Vol.28, No.12, 2124-2131, 2007.

[8] Minrui Fei, Dajun Du, Kang Li, Two-Layer Networked Learning Control of a Nonlinear HVAC System, LSMS, 697–709, 2007.

[9] Andrea Malossini, Enrico Blanzieri, Tommaso Calarco, Quantum Genetic Optimization, IEEE Trans. on evolutionary computation, Vol.12, No.2, 231-241, 2008.

[10] Hongjian Qu, Dawei Zhao, Fangzhao Zhou, New Quantum Clone Evolutionary Algorithm for Multi-objective Optimization, IEEE International Seminar on Business and Information Management, 23-25, 2008.