Optimal Bandwidth Scheduling for Resource-constrained Networks

LI Zu-Xin[1, 2] WANG Wan-Liang[2] CHENG Xin-Min[1]

Abstract In networked control systems (NCSs) with resource constraints, there is an unavoidable tradeoff between the control performance and the quality of service. To address these problems, we present multi-objective programming with a set of constraints to optimize control performance and bandwidth consumption for the first time. Thanks to robust nonlinear approximate function and computational cost, a feed-forward neural network as optimal approximator is employed. The role of the neural network, which provides a good approximation to the optimal solution, dynamically allocates the bandwidth of each control loop so that the overall system performance is maximized while bandwidth consumption is minimized. Preliminary simulation results show that the proposed optimal strategy is an effective tradeoff method between the control performance and bandwidth consumption in networked control applications.

Key words Networked control systems (NCSs), bandwidth scheduling, multi-objective programming, neural network (NN), optimization

Networked control systems (NCSs) are typically spatially distributed systems, in which the communication between sensors, actuators, and controllers occurs through a shared resources-limited communication network[1−3]. These distributed systems, which may operate in an asynchronous manner, have their operation coordinated to achieve desired overall objectives. Compared with the conventional control systems, in which the components are connected via hardwired connections, these distributed systems where information can be transmitted reliably via shared networks or even wireless connections have caused two main changes in control system analysis and design. The first has to do with the explicit consideration of the interconnections. The second change has to do with a renewed emphasis on distributed control systems[4]. To attack these issues, there are various methodologies from different perspectives. One fashionable way is to adopt a suitable control technique, which can compensate time delay induced by network[2, 5−6]. Another effective way is to schedule and optimize the sharing resources, including CPU resources and bandwidth resources, in order to achieve desired overall objectives.

In many application areas, maximizing control performance while exploiting the available network bandwidth as low as possible is crucial, especially for an NCS with resource constraints. Most traditional resource management techniques are based on static resource allocation. The control system is shared by all nodes according to the pre-established allocation strategies regardless of the dynamic control process at runtime[7]. These static strategies, namely fixed bandwidth allocation (FBA) in this paper, work well because they guarantee a constant bandwidth to share for each control loop, allowing it to meet given control performance specifications. The issues of these allocation methods have gained considerable research attention, as reported in [8−11] and therein references.

However, at a close look at the behavior of control loops and the relationship between control performance and requirement of bandwidth, these static strategies may not be optimal for an NCS with bandwidth constraints. For many networked control applications, the desired overall objective is usually to pursue the result that the system performance is maximized while bandwidth consumption is minimized. A scheduling algorithm (thus resources allocation) is said to be optimal if it minimizes or maximizes some given performance measure functions defined over NCS[12]. It is a natural extension issue to select feasible schedule sets and objective functions, which can optimize overall performance. Generally speaking, the optimization is usually a multi-objective mathematical programming with a set of constraints. Consequently, there is an unavoidable tradeoff between the control performance and the quality of service in networked control systems. In [12−13], each NCS was assumed to be associated with a performance measure function subject to rate monotonic (RM) schedulability constraints and NCS stability constraints, and the optimal sampling periods were found by using Matlab function "*fmincon*". This optimal technique is still a fixed resource allocation in spite of improved control performance.

In recent years, it has been reported that the bandwidth resources are allocated dynamically by varying sampling period of control loops (thus their bandwidth consumption). One of the reasons may be the limited bandwidth or the different communication capability of each control loop within multi-loop NCS. Another reason may be resource optimization so as to dynamically adjust the sampling period. Consequently, the key of these novel methods is to sufficiently utilize the sharing resources and effectively improve the control performance (see examples in [7, 14−16]). The algorithm proposed in [15] adopts an extending state-space model and uses an exponential function to dynamically allocate available bandwidth. However, this strategy is not an optimal technique because the control performance depends on selecting model and adjusting the criticalness parameter. In [7, 16], considering single optimal objective of control performance subject to global available bandwidth, the suggested optimal problem solved by linear programming technique was also simplified. These dynamic allocation strategies employ limited bandwidth to dedicate themselves to improve control performance as mentioned above.

Taking these observations as a baseline for our current work on optimizing overall performance and bandwidth consumption, we present a multiple objective optimization (MOO) technique to maximize overall system performance and minimize the bandwidth consumption for the first time. A neural network (NN) as a good and robust nonlinear function approximator is employed. The optimal algorithm, combining with expert knowledge expressed as rules, is used as a teacher to label the data samples for the NN well-trained offline. Consequently, the NN is employed to allocate resources at runtime based on feedback information, i.e., state error. This mechanism can maximize control performance and minimize bandwidth consumption simultaneously.

Received January 8, 2008; in revised form September 7, 2008
Supported by National Natural Science Foundation of China (60573123, 60573056)
1. School of Information Engineering, Huzhou Teachers College, Huzhou 313000, P. R. China 2. College of Information Engineering, Zhejiang University of Technology, Hangzhou 310032, P. R. China
DOI: 10.3724/SP.J.1004.2009.00443

1 System architecture and problem description

In the system considered in this work as shown in Fig. 1, there are n control loops that share the control network together with other non-control applications. The bandwidth manager based on an optimal NN can be embedded in controller node so as to manage bandwidth locally at runtime according to communication data fed back from the control network. All actuator nodes and controller nodes are event-triggered, and all sensor nodes are time-triggered. Information gathered using the network is easy to realize in some multi-point communication systems, i.e., CAN bus. For each control loop, it is assumed that the required messages to carry out all sensor-controller-actuator transaction be finished within each sampling period (i.e., deadline is equal to the sampling period). And it is also assumed that the architecture is based on a real-time network that guarantees all the deadlines upon any bandwidth allocation among all control loops.

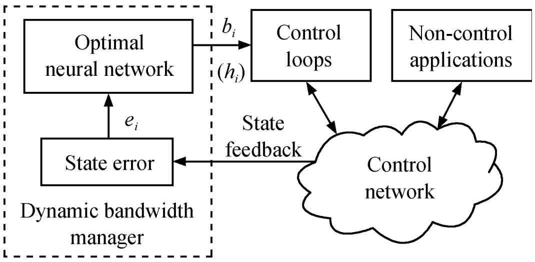

Fig. 1 The multi-loop of NCS based on optimal NN bandwidth management

The linear time invariant system model of the i-th control loop is given by

$$\begin{cases} \dot{\boldsymbol{x}}_i(t) = A_i\boldsymbol{x}_i(t) + B_i\boldsymbol{u}_i(t) \\ \boldsymbol{y}_i(t) = C_i\boldsymbol{x}_i(t), \quad i = 1, \cdots, n \end{cases} \tag{1}$$

where $\boldsymbol{u}_i(t)$ is the control input, A_i, B_i, C_i are matrices with appropriate dimensions, and the vector $\boldsymbol{x}_i(t) = [x_i^1(t), \cdots, x_i^n(t)]$ is the state of the system at time t, and its elements are called state-variables (we assume full state availability).

The proposed method in this paper, namely, MOO as mentioned above, tries to maximize control performance and minimize bandwidth consumption. Dynamic bandwidth manager adjusts the next sampling period (h_i) of each control loop according to the state-variables fed back from network. Therefore, the relation between bandwidth and sampling period for each control loop is given by

$$b_i = \frac{c_i}{h_i} \tag{2}$$

where b_i is assignable partial bandwidth to the i-th control loop, and c_i is the time spent on the messages required to perform each closed loop operation (which may include communication data exchanged from sensor to controller and from controller to actuator, as well as the time spent on executing the controller)[15−16].

By assuming c_i in (2) constant, any change on b_i will directly imply a change on h_i (and vice versa). Henceforth, either b_i or h_i will be used to denote bandwidth (or sampling period). Without loss of generality, we assume that the equilibrium point to be zero. The Euclidean norm of the state vector, also called error, is defined as

$$e_i = \|\boldsymbol{x}_i(t)\| \tag{3}$$

Dynamic bandwidth manager based on MOO strategy obeys the following rationale: when a control process is affected by perturbations so as to deviate its equilibrium, an optimal magnitude b_i will hasten system recovery. However, when a control process tends to the neighborhood of equilibrium, i.e., small e_i, bandwidth manager should distribute optimal minimum magnitude b_i to the relevant control loop.

2 Optimal bandwidth management

2.1 The boundaries of bandwidth allocation

In feedback control systems, it is important that sampled data should be transmitted within a sampling period in order to guarantee stability of control systems. While a shorter sampling period is preferable in most control systems, it can be lengthened up to a certain bound within which stability of the system is guaranteed in spite of the performance degradation. This certain bound is called a maximum allowable delay bound (MADB)[11]. In addition, the MADB depends only on parameters and configurations of the given plant and the controller. It is noted that the MADB can be obtained from the plant model independent of network protocols. Thus, MADB can be treated as the upper bound of sampling period[11, 17−18].

Let control law be $\boldsymbol{u} = -K_i\boldsymbol{x}_i$; (1) can be rewritten as

$$\begin{cases} \dot{\boldsymbol{x}}_i(t) = A_i\boldsymbol{x}_i(t) + M_i\boldsymbol{x}_i(t-\tau_i) \\ \boldsymbol{x}_i(t) = \boldsymbol{\varphi}_i(t), \quad t \in [-\tau_i, 0] \end{cases} \tag{4}$$

where $M_i = -B_iK_i$, $\boldsymbol{\varphi}_i(t)$ is initial condition, $\tau_i = \tau_i^{sc} + \tau_i^{ca} + \tau_i^{c}$, $0 \leq \tau_i \leq \tau_{i,\max} \leq h_i$, τ_i^{sc}, τ_i^{ca}, τ_i^{c} are delays of the i-th control loop from sensor to controller, controller to actuator, and runtime of controller, respectively. The minimum bandwidth of each control loop (maximum sampling period) can be calculated from Theorem 1.

Theorem 1[19]**.** Assume sampling period of the i-th control loop to be h_i. If there exist matrices with appropriate dimensions $P_i > 0$, $Q_i > 0$, X_i, Y_i, and Z_i such that

$$\begin{bmatrix} \Pi_i & P_iM_i - Y_i & h_iA_i^{\mathrm{T}}Z_i \\ M_i^{\mathrm{T}}P_i - Y_i^{\mathrm{T}} & -Q_i & h_iM_i^{\mathrm{T}}Z_i \\ h_iZ_iA_i & h_iZ_iM_i & -h_iZ_i \end{bmatrix} < 0 \tag{5}$$

$$\begin{bmatrix} X_i & Y_i \\ Y_i^{\mathrm{T}} & Z_i \end{bmatrix} \geq 0 \tag{6}$$

where $\Pi_i = A_i^{\mathrm{T}}P_i + P_iA_i + Y_i + Y_i^{\mathrm{T}} + Q_i + h_iX_i$, then the system (4) is asymptotically stable.

In fact, (5) and (6) are linear matrix inequalities (LMI) problems. Therefore, minimum b_i of the i-th control loop (maximum sampling period) can be solved by using LMI toolbox according to Theorem 1. In an NCS, the high sampling rate can increase network load and result in longer delay of the messages. So the maximum b_i (minimum sampling period) must be considered in (7).

$$b_i^{\max} = U_d - \sum_{j=1, j\neq i}^{n} b_j^{\min}, \quad \text{s.t.} \sum_{j=1}^{n} b_j \leq U_d \tag{7}$$

where U_d is a global available bandwidth. Hence, the minimum sampling period of the i-th control loop can be calculated by (2). Consequently, the boundaries of bandwidth allocation should be considered as constraints in order to guarantee the stability of NCS at the time of allocating bandwidth dynamically.

2.2 Performance optimization

As mentioned above, the dynamic bandwidth manager optimizes the overall control performance and bandwidth consumption by adjusting the sampling period of each control loop. Generally speaking, this will formulate a multiple objectives optimization problem with a set of constraints since the optimal object is to maximize control performance and minimize bandwidth consumption. Control performance can denote a certain performance criterion (measured using standard quadratic or linear performance index) so as to evaluate the controlled system response. To do so, we use the integral of the absolute error (IAE) index since the aim of controllers is to minimize the error. That is, there is an inverse relationship between the IAE index and the control performance, of which better control performance will correspond to a smaller IAE.

In fact, the relationship between control performance and a range of allowed periods (i.e., the boundaries of bandwidth allocation) can be approximated by a linear relationship[7, 16, 20]. The cost function is approximated as $J(h_i) = \alpha_i + \beta_i h_i$. Given n control loops with allocated bandwidth vector $\boldsymbol{b} = [b_1, \cdots, b_n]^{\mathrm{T}}$, vector $\boldsymbol{C} = [1, \cdots, 1]^{\mathrm{T}}$, and considering the bandwidth consumption to be minimized, the optimal control performance and bandwidth consumption should solve the problem

$$\min_{\boldsymbol{b}} J_1 = \sum_{i=1}^{n} \left(\alpha_i + \frac{\beta_i c_i}{b_i} \right) \tag{8}$$

$$\min_{\boldsymbol{b}} J_2 = \sum_{i=1}^{n} b_i \tag{9}$$

$$\text{s.t. } \boldsymbol{C}^{\mathrm{T}} \boldsymbol{b} \leq U_d \tag{10}$$

$$b_i^{\min} \leq b_i \leq b_i^{\max} \tag{11}$$

where J_1 and J_2 denote cost function of control performance (measured using IAE index) and bandwidth consumption, respectively. The parameter c_i is transmission time, and α_i and β_i are specific for each control loop, which depend on plant and controller. The parameter solutions can be seen in [16, 20] in detail. Note that the constant α_i can be disregarded when the gradient is calculated. Hence, it is sufficient to estimate the curvature (β_i) of the cost function.

From the solution approach perspective in solving multi-objective programming, one may present some of the goals as constraints to be satisfied while the other objectives can be weighted to make a single composite objective function. In addition, some approaches (e.g., weighted-sum approach, utility function approach, and compromise approach) also can make multiple objectives to a single objective function so as to be solved by traditional mathematic program tools. For this case, multiple objectives are weighted to make a composite objective function. Thus, (8) and (9) can be rewritten as

$$\min_{\boldsymbol{b}} J = \sum_{i=1}^{n} \left(\alpha_i + \frac{\beta_i c_i}{b_i} \right) + \gamma_i \sum_{i=1}^{n} b_i \tag{12}$$

where γ_i is weight, and selected γ_i should make the magnitude level between J_1 and J_2 be satisfied. According to the Karush-Kuhn-Tucker conditions, if $\boldsymbol{b}^* = [b_1^*, \cdots, b_n^*]^{\mathrm{T}}$ is the optimal solution, then

$$\begin{cases} \nabla J(\boldsymbol{b}^*) + \boldsymbol{\lambda}_a - \boldsymbol{\lambda}_b + \lambda \boldsymbol{C} = 0 \\ U_d - \boldsymbol{C}^{\mathrm{T}} \boldsymbol{b}^* \geq 0, \quad b_i^{\min} \leq b_i^* \leq b_i^{\max}, \quad i = 1, \cdots, n \\ \lambda (U_d - \boldsymbol{C}^{\mathrm{T}} \boldsymbol{b}^*) = 0, \ \lambda_i (b_i^{\max} - b_i^*) = 0, \ \lambda_{n+i} (b_i^* - b_i^{\min}) = 0 \\ \lambda \geq 0, \quad \boldsymbol{\lambda}_a \geq 0, \quad \boldsymbol{\lambda}_b \geq 0 \end{cases} \tag{13}$$

where ∇J is the gradient vector, and λ, $\boldsymbol{\lambda}_a = [\lambda_1, \cdots, \lambda_n]^{\mathrm{T}}$, $\boldsymbol{\lambda}_b = [\lambda_{n+1}, \cdots, \lambda_{2n}]^{\mathrm{T}}$ is the Lagrange multiplier.

Although there are several techniques to solve such a constrained optimization problem, e.g., sequential quadratic programming method, feasible direction algorithm, and penalty function method, solving the optimization problem exactly involves a large amount of computations because an optimal algorithmic approach often needs to calculate the gradients or the Jacobian matrix, and a significant number of iterations.

Moreover, from the experience of applications perspective, we can use expert knowledge as connotative constraints in order to further save bandwidth consumption. For this case, we employ the rules expressed as: if $e_i \leq e_i^{th}$, then $b_i = b_i^{\min}, i = 1, \cdots, n$. Here, e_i^{th} denotes threshold at a sufficiently small value for control loop i. Hence, the multi-objective programming with a set of constraints can be rewritten as

$$\min_{\boldsymbol{b}} J = \sum_{i=1}^{n} \left(\alpha_i + \frac{\beta_i c_i}{b_i} \right) + \gamma_i \sum_{i=1}^{n} b_i \tag{14}$$

s.t. (10) and (11)

$$b_i = b_i^{\min}, \text{ if } e_i \leq e_i^{th} \tag{15}$$

Obviously, conventional mathematics program methods actually have difficulty in solving this optimization problem. Some heuristic optimization algorithms (e.g., genetic algorithms) may solve the problem above. However, a significant number of iterations in solving optimization problem may not be feasible at runtime, which may introduce non-negligible overhead because of the computational complexity. To address these problems, we intend to exploit a simple and effective structure that uses a feed-forward NN for dynamic bandwidth allocation, demonstrating a novel application of NN at the same time. The optimal algorithm combining expert knowledge expressed as rules is used as a teacher to label the data samples for the NN training. Once well-trained offline, the NN is used to deliver almost optimal solution to bandwidth scheduling so that the control performance of overall system is maximized while the bandwidth consumption is minimized.

2.3 Optimal scheduling strategy based on NN

Compared with the heuristic algorithms, the feed-forward-NN-based solutions can be delivered in real-time, reducing the scheduling overhead. On the other hand, NN can offer very accurate solutions because of its powerful nonlinear approximation capacities. These attractive twofold reason makes NN to be used in our optimal bandwidth scheduling. A three-layer feed-forward NN (with one hidden layer) is selected as a universal approximator to the bandwidth scheduling optimizer (see Fig. 2), where s is number of neurons in the hidden layer, n is number of elements in input and output vectors. The input vector $\boldsymbol{P} = [e_1, \cdots, e_n]$ is chosen as the network inputs, where the error of each control loop e_i is defined in (3), while network output vector $\boldsymbol{Y} = [h_1, \cdots, h_n]$ is the sampling periods (thus bandwidth). Generally speaking, the appropriate number of hidden neurons is problem dependent, mainly determined by the size of the training sets and the number of input variables. According to our experience and repetitious simulations, the number of hidden neurons is 7.

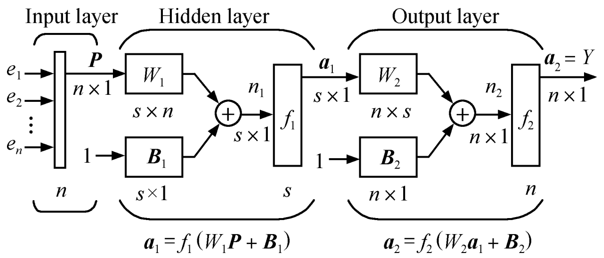

Fig. 2 Neural network structure using abbreviated notation

The relationship between network inputs and outputs is given by

$$\boldsymbol{Y} = f_2(W_2 f_1(W_1\boldsymbol{P} + \boldsymbol{B}_1) + \boldsymbol{B}_2) \tag{16}$$

where W_i, $\boldsymbol{B}_i$ are the weight matrices and bias vectors with appropriate dimensions, respectively. The activation functions used are the log-sigmoid transfer function $f_1(\cdot)$ in the hidden layer and the linear transfer function $f_2(\cdot)$ in the output layer. As we can see from (16), the computational complexity of the NN mainly depends on the number of hidden neurons and the number of control loop. Once weight matrices and bias vectors are trained offline, the bandwidth manager based on NN can distribute the optimal allocated partial bandwidth to each loop at runtime.

In order to collect data samples for NN training, we implement it by the following steps. Firstly, for a given appropriate e_i^{th}, we can easily obtain portion of data pairs by employing expert knowledge expressed as rules in (15). Secondly, to solve the optimal object (14) subject to (10) and (11), another portion of data pairs can be obtained by an algorithmic optimization process. Consequently, these twofold data samples collected are used as optimal solutions to train and validate the backpropagation (BP) neural network. Ideally, the data sample set for NN training should cover the whole range of the input in order to capture all possible scenarios. Before training, the network inputs and targets should perform certain preprocessing steps so that they always fall within a specified range. This process can make neural network training more efficient.

For this work, the BP NN is trained offline using the Levenberg-Marquardt (LM) algorithm which combines the filled function method[21] in order to improve the training process and its convergence to the global minimum. LM algorithm is an iterative technique that locates the minimum of a multivariate function, expressed as the sum of squares of non-linear real-valued functions[22]. Since LM can be thought of as a combination of steepest descent and the Gauss-Newton method, it may be the fastest method for training feed-forward networks up to several hundred weights. During the training process, the LM training algorithm finds one of local minimal points first, the filled function method finds the point that is lower than the minimal point previously found. By repeating these processes, a global minimal point can be obtained at last. Moreover, early stopping method is used for improving generalization. For this technique, the validation set should be representative of all points in the training set. After it is well trained in supervised mode, the NN can be used to provide a good approximation to the optimal solution for real-time bandwidth scheduling.

2.4 Notion of mean network utilization

In this subsection, we firstly illustrate the notion of mean network utilization (MNU) in order to evaluate the bandwidth consumption in NCS with variable sampling periods. The MNU can effectively express the merit of multiple objectives optimization.

The network utilization factor is defined as $U = \sum_i c_i/h_i$ in [12], but it aims at invariable sampling period (or bandwidth) in n control loops. In order to evaluate the bandwidth consumption of n control loops in an NCS with dynamic resource allocation, we define MNU as (17). The integral upper limit could be any time marking the evaluation time interval.

$$U_m = \sum_{i=1}^{n} \lim_{T\to\infty} \frac{1}{T} \int_0^T \frac{c_i(t)}{h_i(t)} \mathrm{d}t \tag{17}$$

In (17), c_i and h_i denote transmission time and sampling period at the k-th sampling instance of the i-th control loop. Obviously, c_i/h_i is a subsection function. It is also fixed in each sampling period. By taking into account that a certain dynamic bandwidth allocation strategy is given at each sampling period, bandwidth consumption can be compared when different bandwidth scheduling strategies are employed.

3 Evaluation

3.1 Simulation setup

We consider an instance of the model (1) for three independent Ball & Beam processes, which can be represented as a linear invariant state-space model given by

$$\dot{\boldsymbol{x}} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \boldsymbol{x} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} \boldsymbol{u}, \quad \boldsymbol{y} = \begin{bmatrix} 1 & 0 \end{bmatrix} \boldsymbol{x} \tag{18}$$

Obviously, the equilibrium position of control system is original. The disturbing node is used to represent other classes of non-control applications which share the communication resources. With such a high priority disturbing node, the utilization of network bandwidth for control loops is restricted to a certain scale. More details of setup are that the disturbing node sharing bandwidth is 70 %. Thus, the global available bandwidth U_d of the three control loops is 30 %. All the weight coefficients of three control loops of NCS are set to be the same. The control law is designed by LQ approach without network. Its matrix coefficient $\boldsymbol{K}$ is [0.5916 1.2382]. The upper and lower bounds of the sampling period are 0.5 s and 0.2857 s inferred by (2), (5), and (7), respectively.

In order to capture all possible scenarios for three different bandwidth scheduling strategies, we run the control system for 3 200 s. At the same time, the random perturbations are generated for each control process with different average perturbation intervals. The distance between two consecutive perturbation intervals in the same system varies in such a way that the system may be continuously perturbed or almost never perturbed.

For comparison, three different cases are considered: 1) fixed bandwidth allocation (FBA), where all control loops always share the global available bandwidth equally (thus the same sampling period) regardless of when the perturbations occur; 2) the suggested optimal strategy mentioned in [7, 16], namely single object optimization (SOO) based on linear programming technique; and 3) our solution with MOO.

3.2 Results

Figs. 3 and 4 show the bandwidth occupancies of the control system with different bandwidth scheduling strategies within partial runtime, respectively. When three control loops are affected by respective perturbations, (see the bottom part marked as "Perturbation"), the bandwidth of the corresponding control loop is allocated dynamically using the SOO strategy and our proposed MOO strategy, also shown in the middle part marked "Each loop BW". The summation of the bandwidth consumption for three control loops is changed dynamically according to deviating equilibrium. The range of bandwidth allocation is from 24 % to 30 %, shown in two figures. However, for the bandwidth occupancy using FBA strategy, it always points to 30 % at all runtime because three control loops always share the available bandwidth equally.

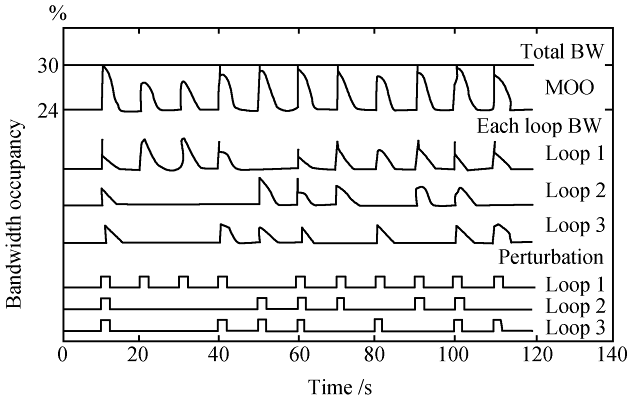

Fig. 3 The bandwidth consumption of three control loops with MOO strategy affected by stochastic perturbation intervals

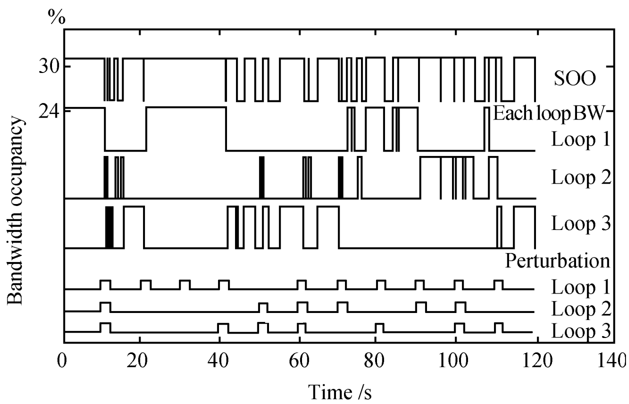

Fig. 4 The bandwidth consumption of three control loops with SOO strategy affected by stochastic perturbation intervals

When the MOO strategy is employed, the bandwidth occupancy of each control loop should be subjected to its assignable bandwidth resources. When there are no perturbations in three control loops, all sampling periods are always the allowable maximum value, and total bandwidth occupancy points to 24 %. However, when SOO strategy is employed, the bandwidth occupancy points to 30 % in most runtime. Actually, for this strategy mentioned in [7, 16], if all of the parameters are the same, e.g., the same control algorithm and plant, all of the available bandwidth will be assigned to the control loop with the largest error e_i. Obviously, this optimal strategy based on linear programming technique dedicates to optimize the control performance and neglects to save the bandwidth resources.

For each control loop affected by different perturbation interval periods T (thus stochastic perturbation interval is an integral multiple of T) within runtime, simulation results show that our proposed solution is highly effective in improving the control performance while saving bandwidth of control network (see Table 1). For the SOO strategy suggested in [7, 16], the control performance, i.e., IAE performance criterion, can effectively improve in contrast to FBA strategy at different partial runtime. However, its mean network utilization (MNU) almost equals to the FBA′s. The respective MNU of FBA and MOO are 30 % and 29.5 %. For our optimal strategy, the MNU is only 24.4 %. It is evident that the MNU of our proposed method is lower than FBA′s and SOO′s. Meanwhile, its IAE is almost equal to the SOO′s though the values in different perturbation interval periods are slightly more or less than the corresponding value using SOO strategy at relevant partial runtime, as shown in Table 1 in detail.

Table 1 The IAE and MNU with three scheduling strategies affected by different perturbation interval periods T at runtime, respectively

T	Run time	FBA Strategy		SOO Strategy		MOO Strategy	
		IAE	MNU	IAE	MNU	IAE	MNU
10 s	120 s	18.429	30.0 %	16.369	28.6 %	16.949	25.3 %
20 s	300 s	15.864	30.0 %	14.284	29.5 %	13.384	24.5 %
30 s	400 s	15.647	30.0 %	13.460	29.7 %	13.678	24.4 %
40 s	480 s	11.080	30.0 %	10.070	29.7 %	10.677	24.2 %
50 s	500 s	9.916	30.0 %	9.352	29.8 %	9.415	24.2 %
60 s	600 s	10.114	30.0 %	9.218	29.8 %	9.080	24.2 %
80 s	800 s	13.112	30.0 %	11.592	29.7 %	12.280	24.2 %
Total	3 200 s	94.163	30.0 %	84.344	29.5 %	85.462	24.4 %

For total IAE, SOO strategy, and MOO strategy are almost equal to improve control performance in contrast to FBA strategy. Furthermore, MOO strategy can save considerable bandwidth resources while SOO strategy can not (see Fig. 5). For our proposed optimal strategy, all these significant results benefit from multiple objectives optimization.

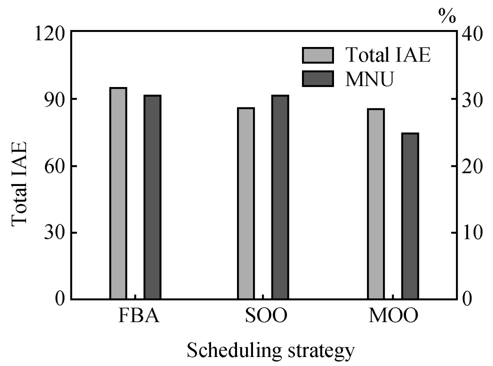

Fig. 5 The total IAE and MNU using different scheduling strategies in all runtime

4 Conclusion

Traditional mathematical programming methods have difficulty in solving multiple objectives programming with a set of constraints combining expert knowledge expressed as rules. Some heuristic algorithms may solve the problem, which need a large amount of computations. The computational cost may become a bar to networked control applications.

However, neural network can be employed as optimal approximator because of its robust nonlinear function and computational speed. The optimization problem combing rules can be utilized as a teacher to generate data samples for NN training. Exactly as what we see, the proposed strategy can maximize the control performance and minimize the bandwidth consumption. Simulation results prove that the optimal strategy is an effective tradeoff method between the control performance and bandwidth consumption in networked control applications. These proved results are useful to networked control applications with resource constraints.

References

1 Baillieul J, Antsaklis P J. Control and communication challenges in networked real-time systems. *Proceedings of the IEEE*, 2007, **95**(1): 9−28

2 Yang T C. Networked control system: a brief survey. *IEE Proceedings — Control Theory and Applications*, 2006, **153**(4): 403−412

3 Hespanha J P, Naghshtabrizi P, Xu Y G. A survey of recent results in networked control systems. *Proceedings of the IEEE*, 2007, **95**(1): 138−162

4 Antsaklis P, Baillieul J. Special issue on technology of networked control systems. *Proceedings of the IEEE*, 2007, **95**(1): 5−8

5 Tipsuwan Y, Chow M Y. Control methodologies in networked control systems. *Control Engineering Practice*, 2003, **11**(10): 1099−1111

6 Li Y Q, Fang H J. Control methodologies of large delays in networked control systems. In: Proceedings of the 5th International Conference on Control and Automation. Budapest, Hungary: IEEE, 2005. 1225−1230

7 Marti P, Lin C X, Brandt S A, Velasco M, Fuertes J M. Optimal state feedback based resource allocation for resource-constrained control tasks. In: Proceedings of the 25th International Real-Time Systems Symposium. Lisbon, Portugal: IEEE, 2004. 161−172

8 Hong S H, Kim W H. Bandwidth allocation scheme in CAN protocol. *IEE Proceedings — Control Theory and Applications*, 2000, **147**(1): 37−44

9 Hong S H. Bandwidth allocation scheme for cyclic-service field bus networks. *IEEE/ASME Transactions on Mechatronics*, 2001, **6**(2): 197−204

10 Hong S H, Kim Y C. Implementation of a bandwidth allocation scheme in a token-passing field-bus network. *IEEE Transactions on Instrumentation and Measurement*, 2002, **51**(2): 246−251

11 Park H S, Kim Y H, Kim D S, Kwon W H. A scheduling method for network-based control systems. *IEEE Transactions on Control Systems Technology*, 2002, **10**(3): 318−330

12 Zhang W. Stability Analysis of Networked Control Systems [Ph. D. dissertation], Case Western Reserve University, USA, 2001

13 Branicky M S, Phillips S M, Zhang W. Scheduling and feedback co-design for networked control systems. In: Proceedings of the 41st Conference on Decision and Control. Las Vegas, USA: IEEE, 2002. 1211−1217

14 Ren X D, Li S B, Wang Z, Yuan M Z, Sun Y X. A QoS management scheme for paralleled networked control systems with CAN bus. In: Proceedings of the 29th Annual Conference of the IEEE Industrial Electronics Society. Virginia, USA: IEEE, 2003. 842−847

15 Velasco M, Fuertes J M, Lin C X, Marti P, Brandt S. A control approach to bandwidth management in networked control systems. In: Proceedings of the 30th Annual Conference of the IEEE Industrial Electronics Society. Busan, South Korea: IEEE, 2004. 2343−2348

16 Velasco M, Marti P, Frigola M. Bandwidth management for distributed control of highly articulated robots. In: Proceedings of the IEEE International Conference on Robotics and Automation. Barcelona, Spain: IEEE, 2005. 265−270

17 Kim Y H, Park H S, Kwon W H. A scheduling method for network-based control systems. In: Proceedings of the 17th American Control Conference. Philadelphia, USA: IEEE, 1998. 718−722

18 Kim D S, Lee Y S, Kwon W H, Park H S. Maximum allowable delay bounds of networked control systems. *Control Engineering Practice*, 2003, **11**(11): 1301−1313

19 Li Zu-Xin, Wang Wan-Liang, Lei Bi-Cheng, Chen Hui-Ying. An approach to bandwidth management based on fuzzy logic. *Engineering Science*, 2008, **10**(7): 104−111 (in Chinese)

20 Cervin A, Eker J, Bernhardsson B, Arzen K E. Feedback-feedforward scheduling of control tasks. *Real-Time Systems*, 2002, **23**(1-2): 25−53

21 Li Huan-Qin, Wan Bai-Wu. A new global optmzation algorithm for training feedforward neural networks and its application. *System Engineering Theory and Practice*, 2003, **23**(8): 42−47 (in Chinese)

22 Hagan M T, Menhaj M. Training feed-forward networks with the Marquardt algorithm. *IEEE Transactions on Neural Networks*, 1994, **5**(6): 989−993

LI Zu-Xin Associate professor at Huzhou Teachers College. He received his Ph. D. degree from Zhejiang University of Technology in 2008. His research interest covers networked control systems and intelligent control. Corresponding author of this paper. E-mail: lzx@hutc.zj.cn

WANG Wan-Liang Professor at Zhejiang University of Technology. His research interest covers computer control, computer net, CMIS, and production scheduling.
E-mail: wangwanliang@zjut.edu.cn

CHENG Xin-Min Associate professor at Huzhou Teachers College. He received his master degree from Southeast University in 2004. His research interest covers image and speech processing. E-mail: chengxinmin@hutc.zj.cn