Intelligent Scheduling Controller Design for Networked Control Systems Based on Estimation of Distribution Algorithm

LI Hongbo (李洪波), SUN Zengqi (孙增圻) **, CHEN Badong (陈霸东), LIU Huaping (刘华平)

Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China

Abstract: The use of communication networks in control loops has gained increasing attention in recent years due to its advantages and flexible applications. The network quality-of-service (QoS) in those so-called networked control systems always fluctuates due to changes of the traffic load and available network resources. This paper presents an intelligent scheduling controller design approach for a class of NCSs to handle network QoS variations. The sampling period and control parameters in the controller are simultaneously scheduled to compensate for the network QoS variations. The estimation of distribution algorithm is used to optimize the sampling period and control parameters for better performance. Compared with existing networked control methods, the controller has better ability to compensate for the network QoS variations and to balance network loads. Simulation results show that the plant setting time with the intelligent scheduling controller is reduced by about 64.0% for the medium network load and 49.1% for high network load and demonstrate the effectiveness of the proposed approaches.

Key words: networked control systems (NCSs); estimation of distribution algorithm (EDA); network-induced delay; packet dropout; network quality-of-service (QoS) variation

Introduction

Networked control systems (NCSs) are a type of distributed control systems with control loops closed via communication networks. Compared with traditional “point to point” control systems, NCSs facilitate resource sharing, reduce installation and reconfiguration costs, and improve flexibility. Therefore, NCSs have great potential in manufacturing plants, vehicles, aircraft, and spacecraft. With their many advantages and potential applications, NCSs have attracted much attention in the control and computer community in recent years, with many interesting results [1-12].

In NCSs, the controller design is one of the most important topics. Yue et al. [1] investigated the state feedback control for NCSs with network-induced delays and packet dropout via linear matrix inequality (LMI) approach. Luis and Panos [4] and Zhivoglyadov and Middleton [5] proposed a model based control approach for NCSs with network constraints. Nissson et al. [6] and Hu and Zhu [7] investigated the optimal stochastic control approach for NCSs with the network-induced delay shorter and longer than the sampling period. These results have shown many advantages and have solved various problems. However, those studies have assumed that the control parameters and sampling period remain constant regardless of network quality-of-services (QoSs) variations. In practical circumstances, the network QoS always fluctuates due to changes of the traffic load and available network resources. NCSs are functionally related systems and
their performance depends not only on the control algorithms but also on the network QoS. Therefore, despite the progress made in the NCS controller design, it has become evident that advanced control methodologies with QoS variation compensation are required.

Although there has only been a limited amount of research on NCSs controller design with QoS variation compensation, there are some interesting results reported in the literature. In general, there are two main approaches. The first one is the sampling period scheduling approach, which enables NCSs to achieve a higher resource utilization rate and better control performance[^8^-^10^]. The second approach adjusts the control parameters to compensate for the network QoS variations[^11^,^12^]. Both methodologies show promising results. However, these approaches still have some limitations since the two approaches have been studied separately rather than having a networked controller that simultaneously adjusts the sampling period and control parameters. Moreover, the sampling period and control parameters have not been optimized for NCSs.

This paper presents a controller design approach that simultaneously adjusts the sampling period and control parameters for NCSs, where the sampling period and control parameters are optimized by using the estimation of distribution algorithm (EDA). The EDA is a population-based search algorithm introduced by Muhlenbein and Paar[^13^] for evolutionary computations. During the past several years, the EDA has received much attention as one of the fastest growing techniques for genetic and evolutionary computations. Many interesting results have been reported in the literature[^14^,^15^]. Since the EDA has many advantages, such as better speed, better solutions, and less tuned parameters, the EDA is used here to solve the optimization problem.

The objective of this paper is to construct an intelligent scheduling controller to handle network QoS variations by simultaneously scheduling the sampling period and control parameters based on network QoS variations. The method first optimizes the sampling period and control parameters offline for the NCS. Then, the controller schedules the sampling period and control parameters online based on QoS variations.

1 Problem Statement

The NCSs structure is shown in Fig. 1, where networks exist between the sensor and the controller nodes. The controlled plant, $G_p$, is given by

$$\begin{align*}
\dot{x}(t) &= Ax(t) + Bu(t) + v(t), \\
y(t) &= Cx(t)
\end{align*}$$

where $x(t)$ is the state vector, $u(t)$ is the control input vector, $y(t)$ is the plant output and $v(t)$ is the disturbance vector. $A$, $B$, and $C$ are matrices with appropriate dimensions.

![Fig. 1 NCSs structure](image)

The actuator is event-driven with the event here being the arrival of a packet. The sensor is clock-driven with the sampling period denoted as $h$. Therefore, the system has serial pairs $\{y(k), kh\}$. The sampling period can be adjusted online according to some mechanism for better NCSs performance.

Networks are unreliable data transmission paths with packets not only suffering network-induced delay and out-of-order packets, but even transmission loss as illustrated in Fig. 2. Packets arrive at the controller randomly because the network-induced delays are random. Let $T_i (T_{i+1} > T_i)$ denote the packet arrival instant relative to the initial time, where the subscript “$i$” is the number of packets received by the controller. A serial pair $\{\hat{y}(i), T_i\}$ corresponds the arrival of packets at the controller, where $\{\hat{y}(i)\}$ is a subset of $\{y(k)\}$.

![Fig. 2 NCSs timing diagram](image)

The control objective is to design a networked controller based on $\{\hat{y}(i), T_i\}$ to guarantee the NCSs stability and performance. The controller dynamics, $G_c$, can be described as

$$u(k) = f(\{\hat{y}(i), T_i\}, \star)$$

where $\star$ represents other appropriate information such as the plant model.

Then, the closed loop NCSs can be represented as

$$\Sigma(G_p, \{y(k), kh\}, G_c, \{\hat{y}(i), T_i\})$$

**Remark 1** In Eq. (3), $\{\hat{y}(i)\}$ is a subset of $\{y(k)\}$. If the element numbers in the two sets are not equal, some packets have been dropped. If the ele-
ments in the two sets are not in the same order, some packets are out-of-order. Therefore, Eq. (3) can be viewed as a general expression of the concerned NCSs which considers the effects of packet out-of-order, network-induced delay, and packet dropout.

2 Intelligent Scheduling Controller Design

The controller structure is shown in Fig. 3, where the QoS monitor is a high level function which provides the packets to the switching controller through the “data path”, and sends information to the switching controller through the “information path”. The regulator schedules the sampling period and adjusts the control parameters online according to the current network QoS and informs the QoS monitor and the switching controller about the new sampling period. The switching controller sends the control signals to the plant based on packets specified by the QoS monitor.

2.1 Network QoS partition

Two parameters are employed to define the different network QoS levels.

- \( T_{\text{max}} \) denotes the current maximum network-induced delay and is used to indicate the maximum expected packet delay length.
- \( D_{\text{rate}} \) denotes the current packet dropout rate, which is defined as the ratio of the number of lost packets to the total number of transmitted packets. \( D_{\text{rate}} \) is used to indicate the probability that a packet will be lost.

Let \( \hat{T}_{\text{max}} \) and \( \hat{D}_{\text{rate}} \) denote the upper bounds of \( T_{\text{max}} \) and \( D_{\text{rate}} \), and then the universe of discourse \( Q \) for the network QoS can be described as

\[
Q = \{ [0, \hat{T}_{\text{max}}], [0, \hat{D}_{\text{rate}}] \} \quad (4)
\]

Divide \([0, \hat{T}_{\text{max}}] \) into \( N_1 \) different subsets denoted as \( \phi_i \), where \( i = 1, \ldots, N_1 \). Similarly, divide \([0, \hat{D}_{\text{rate}}] \) into \( N_2 \) subsets denoted as \( \phi_j \), where \( j = 1, \ldots, N_2 \). Then by combining \( \phi_i \) with \( \phi_j \), \( Q \) can be divided into \( M \) different levels as illustrated in Fig. 4, where \( M = N_1 N_2, i = 1, \ldots, N_1, j = 1, \ldots, N_2 \). Each QoS level can be expressed as

\[
Q(j-i)N_1 + j = \{ T_{\text{max}} \in \phi_i, D_{\text{rate}} \in \phi_j \} \quad (5)
\]

![Fig. 4 Network QoS partitioning](image)

The partitioning of \( Q \) satisfies

\[
\begin{align*}
Q, Q_j, Q_{ji} \cup \cdots \cup Q_M &= Q; \\
Q_i \cap Q_j &= \emptyset, \quad 1 \leq i, j \leq M, \quad i \neq j
\end{align*}
\]

(6)

2.2 Switching controller design

In the switching controller, there exists a special sampling period \( h_i \) for each QoS level \( Q_i \), where \( i \in \{1, \ldots, M\} \). Denote the \( m \)-th sampling period as \( h_{i,m} \), where \( i \in \{1, \ldots, M\} \) and the value of \( i \) is determined by the network QoS condition at the \( m \)-th sampling period. Let \( t_{\text{sc}} \) denote the network-induced delay from the sensor to the controller. The packet with network-induced delay longer than the current sampling period will be discarded by the controller, we can assume that \( t_{\text{sc}} = +\infty \) if a packet is dropped out or encounters a delay longer than the current sampling period. To simplify the analysis, the sum of the first \( k \) sampling periods is represented by \( \bar{h} \), where \( \bar{h} := \sum_{m=1}^{k} h_{i,m} \). Let \( Q^{\text{real}} \) denote the network QoS condition at time instant \( \bar{h} \). Then the switching controller is designed as

Switching rule \( i \): If \( Q^{\text{real}} \in Q_i \), then

Local controller rules:

- Rule 1: If \( t_{\text{sc}} \in [0, h_i) \), then
  \[
  u(\bar{h}) = -L_x(\bar{h}),
  \]
  \[
  \tilde{x}(\bar{h} + h_i) = F_x(\bar{h}) + G_u(\bar{h}) + K_y[y(\bar{h}) - Cx(\bar{h})];
  \]

- Rule 2: If \( t_{\text{sc}} = +\infty \), then
  \[
  u(\bar{h}) = -L_x(\bar{h}),
  \]
  \[
  \tilde{x}(\bar{h} + h_i) = F_x(\bar{h}) + G_u(\bar{h})
  \]

(7)
where $i = 1, \ldots, M$ and $F_i$ and $G_i$ are
\[ F_i = e^{\omega_i}, \quad G_i = \int_0^h e^{\omega_i} \, drB \] (8)

Note that the switching controller is clock-driven. $Q^\omega$ and $t_{ac}$ represent the switching state of the switching controller. $h_i$, $K_i$, and $L_i$ are parameters to be designed.

### 2.3 Optimization problem

Since the control parameters and sampling period are closely related to the control performance of the closed-loop system, the NCSs performance can be improved by using an optimum sampling period and control parameters. Unfortunately, a closed-form relationship among the network QoS, sampling period, control parameters, and control performance is not available. Therefore, the optimum parameters should be determined using an optimization algorithm. The problem will be formulated as an optimization problem, whose objective is to obtain the optimal $h_i$, $K_i$, and $L_i$ for the NCSs, where $i = 1, \ldots, M$. The input parameter set for the optimization problem is
\[ V = [h_1, L_1, K_1, \ldots, h_M, L_M, K_M] \] (9)

Since $L_i$ and $K_i$ may be matrices, the input parameter set can be rewritten as
\[ V = [v_1, v_2, \ldots, v_{n-1}, v_n] \] (10)
where $S$ is the dimension of the input parameter set $V$.

#### 2.3.1 Fitness function

The fitness function for the control performance is
\[ \text{fitness} = \lambda_1 \text{fitness}_1 + (1 - \lambda_1) \text{fitness}_2 \] (11)
where fitness and fitness$_2$ are used to evaluate the macroscopic and microscopic aspects of the NCSs performance.

The fitness function fitness$_1$ is defined as
\[ f_1(M_{des}, M_{act}) = \begin{cases} 0, & M_{act} \leq M_{des}; \\ (M_{act} - M_{des}) / [(\xi - 1) M_{act}], & M_{des} < M_{act} \leq \xi M_{des}; \\ 1, & M_{act} > \xi M_{des} \end{cases} \] (12)

\[ f_2(T_{end}, T_{act}) = \begin{cases} 0, & T_{act} \leq T_{end}; \\ (T_{end} - T_{act}) / [(\xi - 1) T_{end}], & T_{des} < T_{act} \leq \xi T_{des}; \\ 1, & T_{act} > \xi T_{des} \end{cases} \] (13)

where $f_1(M_{des}, M_{act})$ is a cost function for the actual overshoot, $M_{act}$, and the desired overshoot, $M_{des}$. $f_2(T_{end}, T_{act})$ is a cost function for the actual settling time, $T_{act}$, and the desired settling time $T_{des}$. $\xi$ is a weighting factor larger than 1, which is used to adjust the performance of $f_1(M_{des}, M_{act})$ and $f_2(T_{end}, T_{act})$.

The fitness function fitness$_2$, is defined as
\[ \text{fitness}_2 = \lambda_2 f_2(IAE) + (1 - \lambda_2) f_1(ITAE) \] (15)

\[ f_2(IAE) = \int_0^{T_{end}} | y(t) - r(t) | \, dt / \int_0^{T_{end}} | r(t) - y(0) | \, dt \] (16)

\[ f_1(ITAE) = \int_0^{T_{end}} | y(t) - r(t) | \, dt / \int_0^{T_{end}} | r(t) - y(0) | \, dt \] (17)

where $y(t)$, $r(t)$, and $y(0)$ denote the actual output, the initial output, and the desired output, respectively. $T_{end}$ is a finite time chosen such that the integral approaches steady-state value and is usually chosen as the settling time.

Note that $\lambda_1$, $\lambda_2$, and $\lambda_3$ in Eqs. (11), (12), and (15) are weighting factors ranging from 0 to 1.

#### 2.3.2 Sampling period constraint (SPC)

To ensure the physical feasibility of the sampling period and the system stability, we introduce the sampling period constraint subject to the optimization problem as follows:

\[ \text{SPC:} \quad 0 < h_i < h_{max}, \quad i = 1, \ldots, M \] (18)
where $h_{max}$ is the upper bound of the sampling period.

Let $w_{sys}$ denote the control system bandwidth, which is defined as the maximum frequency at which the system output will track an input sinusoidal in a satisfactory manner. $h_{max}$ will be defined as
\[ h_{max} = 1 / (20 w_{sys}) \] (19)

#### 2.3.2 Control parameters constraint (CPC)

The control parameters constraint guarantees that the controller parameters do not exceed the limitations of the system.

\[ \text{CPC:} \quad \chi^{min} < \chi < \chi^{max} \] (20)
where $\chi$ is one element of $[L, K, \xi]$ and $\chi^{min}$ and $\chi^{max}$ are the lower and upper bounds of $\chi$.

Then, the optimization problem can be expressed as follows:
Minimize fitness
s. t. (i) Plant is subjected to Eq. (1);
(ii) SPC: \( 0 < h_i < h_{\text{max}} \);
(iii) CPC: \( \chi_{\text{min}} < \chi_i < \chi_{\text{max}} \)

\[
\begin{align*}
\text{Minimize} & \quad \text{fitness} \\
\text{s. t.} & \quad \text{Plant is subjected to Eq. (1)}; \\
& \quad \text{SPC: } 0 < h_i < h_{\text{max}}; \\
& \quad \text{CPC: } \chi_{\text{min}} < \chi_i < \chi_{\text{max}}.
\end{align*}
\]

\[\text{(21)}\]

### 2.4 Sampling period and control parameters optimization based on the EDA

The intelligent scheduling controller uses the EDA to optimize the sampling period and control parameters. The EDA is a new non-deterministic, heuristic search algorithm which does not use a crossover operator or a mutation operator. Instead, new individuals are sampled from a probability distribution estimated from a database containing selected individuals from the previous generation. Thus, the relationships between the variables are explicitly and effectively captured and exploited. As a result, the EDA can push toward higher speeds by predicting population movements in the search space without needing many parameters.

Since the input parameter set \( V \) in the search space is assumed to satisfy a multivariate normal distribution \( p(q) = \prod_{i=1}^{S} p(q_i), \) which is the product of \( S \) independent univariate normal distributions \( p(q_i), \) the solution of EDA can be replaced by two vectors, the mean values of Gaussian normal distribution, \( \mu_i, \) and the standard deviation, \( \sigma_i. \) No interactions among variables are considered. The algorithm can be summarized as follows:

1. Generate \( W_2 \) individuals (the initial population) with a multivariate normal distribution \( p(q) = \prod_{i=1}^{S} p(q_i). \)
2. Repeat the following steps until the stopping criterion is met.
   - Select the best \( W_2 \) individuals from the parent generation, where \( W_2 \leq W_1. \) The selected individuals are denoted as \( \bar{D} = (\bar{R}_1, ..., \bar{R}_w), \) where \( \bar{R}_i, i = 1, ..., W_2, \) are the input parameter sets denoted as \( \bar{R}_i = (\bar{R}_{i1}, ..., \bar{R}_{iS}). \)
   - Update the multivariate Gaussian distribution using the selected individuals according to

\[
\begin{align*}
\mu_i &= \frac{1}{W_2} \sum_{j=1}^{W_2} \bar{R}_{ij} \\
\sigma_i &= \left( \frac{1}{\sqrt{W_2}} \sum_{j=1}^{W_2} (\bar{R}_{ij} - \bar{R}^{\text{ave}})^2 \right)^{1/2} \quad \text{(23)}
\end{align*}
\]

where \( \bar{R}^{\text{ave}} \) is the mean of \( \bar{R}_{ij}, \ j = 1, ..., W_2. \)

### 2.5 QoS monitor design

Using synchronized clocks in the sensor and the controller enables the QoS monitor to calculate the arriving packet delay information from the time stamps in the packet. Nilsson\(^{[16]}\) gave more details on clock synchronization. If the network-induced delay of the arriving packet is shorter than the current sampling period \( h_i, \) the QoS monitor will send the packet to the controller through the “data path”, and send the network-induced delay information to the controller through the “information path”. Then the switching controller will use Rule 1 in Eq. (8) to compute the control signal. If the QoS monitor has not obtained the packet from the \( k \)-th sampling period until the time instant \( h_i + h_i, \) the QoS monitor will actuate the switching controller to use Rule 2 in Eq. (8) to compute the control signal via the “information path”.

### 2.6 Regulator design

Based on the EDA, a table with respect to network QoS levels, the optimal sampling period, and the control parameters can be formed. Then, the lookup table technique can be used in the regulator to schedule the optimal sampling period and control parameters for the controller based on the actual network QoS.

### 3 Simulation Results

The effectiveness of this intelligent scheduling controller is illustrated by using the controller in an NCS shown in Fig. 1. The plant is given by

\[
\begin{align*}
\dot{x}(t) &= \begin{bmatrix} 0 & 1 \\ 0 & -0.1 \end{bmatrix} x(t) + \begin{bmatrix} 0 \\ 0.1 \end{bmatrix} u(t) + v(t), \\
y(t) &= \begin{bmatrix} 0 & 1 \end{bmatrix} x(t) \\
\end{align*}
\]

where \( v(t) \in \mathbb{R}^2 \) is the extended disturbance, which is
a zero-mean Gaussian white noise.

In the used network, the network-induced delay distributions corresponding to medium and high network loads are given in Fig. 5, with packet dropout rates of 4% and 8%.

![Fig. 5 Network-induced delay distributions](image)

The network QoS conditions corresponding to the medium and high network loads are defined as levels $Q_1$ and $Q_2$:

$Q_1 = \{ T_{\text{max}} \in [0.05, 0.25], D_{\text{rate}} \in [0.00, 0.05] \}$

$Q_2 = \{ T_{\text{max}} \in [0.25, 0.45], D_{\text{rate}} \in [0.00, 0.08] \}$

The simulation used a population size of 60 for the EDA, with the best 20 individuals selected from the parent generation to update the multivariate Gaussian distribution for the next generation. The controller sampling period and control parameters from the EDA are:

$h_{\text{Med}} = 0.25, \quad L_{\text{Med}} = [7.7646, 4.6569],
\quad K_{\text{Med}} = [1.9506, 3.9012]$ (25)

$h_{\text{High}} = 0.40, \quad L_{\text{High}} = [4.4616, 3.2922],
\quad K_{\text{High}} = [1.9216, 2.4020]$ (26)

The state response of the plant controlled with fixed sampling period and control parameters was also investigated for comparison. The sampling period is set to 0.3 s. The state responses of the NCS with fixed sampling period and control parameters are plotted in Fig. 7. In this case, the plant setting time is 5.0 s for the medium network load and 5.5 s for the high network load.

The state responses of the NCS with fixed sampling period and control parameters are plotted in Fig. 7. In this case, the plant setting time is 5.0 s for the medium network load and 5.5 s for the high network load.

The simulation results show that compared to the controller with fixed sampling period and control parameter, the plant setting time with the intelligent scheduling controller is reduced by about 64.0% for the medium network load and 49.1% for high network load, even with a much worse initial state. Therefore, the intelligent scheduling controller effectively improves the dynamic performance of the NCSs. Analysis of the sampling period of the intelligent scheduling controller indicates an interesting nature. For high network loads, the controller sampling period is relatively long. However, for low network loads, the intelligent scheduling controller increases the sampling rate to obtain better performance. This mechanism helps the NCSs achieve a higher resource utilization rate and avoid overloads. Thus, the intelligent controller more effectively balances network loads.
4 Conclusions

This paper has proposed an intelligent scheduling controller that simultaneously adjusts the sampling period and control parameters to compensate network QoS variations in NCSs. The estimation of distribution algorithm is used to optimize the sampling period and the control parameters for better performance. Simulations illustrated the effectiveness of the intelligent scheduling controller.

This study emphasized the NCS controller design without considering the closed-loop system stability. The stability of NCSs with the intelligent scheduling controller will be analyzed in future work.

References