Optimal resource allocation in networked control systems

Gokhan Cetin1, M. Sami Fadali1, Hao Xu1
1Department of Electrical & Biomedical Engineering, University of Nevada, Reno, USA
E-mail: gokhancetin@nevada.unr.edu

Abstract: This study proposes an optimal bandwidth allocation algorithm for a Networked Control System (NCS) that includes a time-driven sensor, event-driven controller and random channels. The authors predict resource demands at the next time step for the random channels, then calculate the resource demands from the event-driven and the time-driven channels. They allocate bandwidths for each channel by solving a convex optimisation problem based on the resource demands. When the total traffic exceeds the total capacity of the network, only the random and event-driven network bandwidths are allocated optimally. For time-driven channels, they increase the sampling period to reduce traffic volume and satisfy the total capacity constraint. This maintains sufficiently slow sampling to improve the quality of service in NCS. An upper bound on the sampling period is set to guarantee sufficiently fast sampling for control dynamics. Simulation results show that the proposed approach minimises network congestion.

1 Introduction

Networked control system (NCS) is a control system in which subsystems such as controllers, actuators, and sensors are connected via a communication network. The network can be shared with other nodes outside of the control system. Because the NCS includes a communication network, it is essential to schedule control loops to prevent package dropout or delay. The latter has a significant effect on the dynamics and stability of the NCS [1, 2].

NCS structure is categorised as centralised, decentralised and distributed based on its configuration [3, 4]. Since the Internet of Things (IoT) is rapidly developing, using communication networks and advanced protocols can bring many advantages in applications of distributed control systems such as distributed transportation networks, distributed electrical power systems and smart grids, remote surgery, cellular phone networks, industrial and manufacturing systems, biological networks and flocking etc. [3]. In an NCS, packet loss or delay of the data of subsystems, such as controllers, actuators and sensors, can occur because the data are transmitted through a shared communication network. Nevertheless, NCS has many advantages such as modularity, scalability, and robustness [4, 5]. Analysis and design of applications of NCS have many challenges due to communication, computation and control limitations. Communication limitations in NCS are mainly caused by limited channel capacity, data dropout, network delay, and time-varying network topology [4, 5].

Researchers have proposed methods to mitigate network limitations using judicious bandwidth allocation. In [1], the authors proposed bandwidth allocation and scheduling for NCS using non-linear-programming techniques. They optimise bandwidth utilisation by using the optimal sampling frequency. To optimise utilisation, they formulated an approximation of bandwidth utilisation in terms of the sampling period.

Bandwidth is directly related to network traffic or sampling rate [3]. In combined networks that include event-driven, time-driven, and random network channels, it is difficult to apply and manipulate a variable sampling period for both random and event-driven channels. In [6], the authors proposed a scheme to allocate bandwidth using a variable sampling period. The sampling period is chosen as the reciprocal of the bandwidth. They maximised a utility function that is a monotonically increasing function of the sampling rate to ensure that higher sampling provides better quality of control. They imposed a lower bound on the sampling rate $r > r_{\text{min}}$, where $r_{\text{min}}$ is a lower limit for the stability of the system [6].

In [7], a bandwidth allocation method is proposed by designing an event-triggered controller that guarantees stability and a higher bound on bandwidth requirement. A robust controller is proposed for norm-bounded uncertainty, which stabilises the NCS and guarantees bandwidth usage.

The authors in [8] propose a dynamic allocation strategy for NCS by restricting the bandwidth of each control loop in a bounded interval and allocating bandwidth without exceeding network capacity. They estimate the error of each control loop using the mean absolute error and allocate bandwidth using the estimated error. A dynamic weight is determined using quadratic minimisation and the sampling period for the time-driven channel is bounded interval and allocating bandwidth using the estimated error.

Although there is a rich literature on bandwidth allocation, to our knowledge, the problem of bandwidth allocation for a network shared by NCS and random traffic has not been addressed. In this paper, novel bandwidth allocation and sampling rearrangement scheme for NCS is proposed. We assume multiple links are sharing a bandwidth including random traffic, time-driven traffic, and event-driven traffic. Using the current traffic of the random network, we predict the traffic at the next sampling point using a Poisson graphical model [9]. Bandwidth is allocated based on the resource demands of each individual channel by solving a convex optimisation problem based on the predicted traffic for the random channel and the known deterministic traffic. We assume that network capacity is time-invariant over the planning horizon of the control system. However, if the lower bound on capacity is known and is adequate to satisfy the quality of service requirements for the NCS, the proposed method can provide a prediction of network traffic subject to the known capacity constraints and is therefore valid if the capacity changes with time.

Since the total capacity of the network is limited, when the total demand exceeds the total capacity of the network, packet dropout and/or delay may occur. To prevent this, the bandwidth of the random and event-driven channels is allocated optimally using a Poisson graphical model [9]. Bandwidth is allocated based on the resource demands of each individual channel by solving a convex optimisation problem based on the predicted traffic for the random channel and the known deterministic traffic. We assume that network capacity is time-invariant over the planning horizon of the control system. However, if the lower bound on capacity is known and is adequate to satisfy the quality of service requirements for the NCS, the proposed method can provide a prediction of network traffic subject to the known capacity constraints and is therefore valid if the capacity changes with time.

Since the total capacity of the network is limited, when the total demand exceeds the total capacity of the network, packet dropout and/or delay may occur. To prevent this, the bandwidth of the random and event-driven channels is allocated optimally using a Poisson graphical model [9]. Bandwidth is allocated based on the resource demands of each individual channel by solving a convex optimisation problem based on the predicted traffic for the random channel and the known deterministic traffic. We assume that network capacity is time-invariant over the planning horizon of the control system. However, if the lower bound on capacity is known and is adequate to satisfy the quality of service requirements for the NCS, the proposed method can provide a prediction of network traffic subject to the known capacity constraints and is therefore valid if the capacity changes with time.

The main contributions of the paper are
i. The paper addresses bandwidth allocation for a network that serves random and deterministic traffic.

ii. The paper includes a novel statistical approach to traffic prediction.

The remainder of the paper is organised as follows: Section 2 describes the data rate prediction of random channels, the convex optimisation problem, and the dynamic sampling problem. In Section 3, simulation results are discussed. Section 4 provides conclusions.

2 Optimisation of resource allocation

Fig. 1 shows a block diagram of a closed-loop NCS, including sensor, actuator, and controller. The sensor samples the actuator output, and sends the output data to the controller as a package. The controller receives and processes the packet for generating a control signal and sends it to the actuator [3, 10]. The time interval between two sample packets is defined as the sampling period, and is denoted by \( h \). Thus, the sensor sends one packet that includes sample data every \( h \) seconds and the rate of transmission between nodes is the reciprocal of the sampling period

\[ r = 1/h \]  

The transmission rate for each channel is the bandwidth capacity for the channel. The transmission rate is a critical factor in NCS stability and performance conditions. If the transmission rate exceeds the bandwidth capacity, network congestion occurs, which leads to packet losses, delays, and jitter in the congested channels. Higher sampling rates give better performance but generate more data and may cause congestion. Hence, the sampling rate must be small enough to avoid congestion but large enough to guarantee stability and good performance for the NCS.

We assume that multiple links share bandwidth and that only some of the network traffic is random while the rest is deterministic. Using the current traffic vector of the random channels, we predict the traffic vector at the next time point utilising a Poisson graphical model [9]. The conditional distribution for the random channel is specified by the univariate Poisson distribution [9]

\[ P_{\text{pois}}(t_i(k + 1)|t_i(k)) = \exp\left\{\Psi(t_i(k))t_i(k + 1)\right\} 
\]

where \( t_i \) is the traffic of the \( i \)th random channel, \( t' \) is the traffic vector of the random channels, \( t_i' \) is a vector including the traffic history for the network, and \( \Psi(t_i') \) is any function that depends on \( t_i' \).

Assume that \( \Psi \) is in the form

\[ \Psi(t_i'(k)) = \ln(\theta^T t_i') \]

where

\[ \theta = [\theta_1, \theta_2, ... \theta_n]^T \]

This choice of the function \( \Psi \) gives the distribution

\[ P_{\text{pois}}(t_i'(k+1)|t_i(k)) = \frac{(\theta^T t_i')^\gamma t_i'(k + 1)! e^{-\theta^T t_i'}}{t_i'(k + 1)!} \]

For \( N \) independent and identically distributed measurements, we have the likelihood function

\[ L(t_i'(k + 1)) = \prod_{j=1}^{N} \frac{(\theta^T t_i')^\gamma t_i'(k + 1)! e^{-\theta^T t_i'}}{t_i'(k + 1)!} \]

The natural logarithm of the likelihood function is

\[ \ln L(t_i'(k + 1)) = \sum_{j=1}^{N} \gamma t_i'(k + 1)! - \theta^T \sum_{j=1}^{N} t_i'(k + 1)! \]

To find the maximum-likelihood estimator of the parameters, we differentiate the log-likelihood, and equate to zero

\[ \frac{\partial \ln L(t_i'(k + 1))}{\partial \theta} \]  

We now have the condition

\[ \sum_{j=1}^{N} t_i'(k + 1)! \frac{\partial \theta}{\partial \theta} \]  

The condition yields an expression that cannot readily be solved for the parameter estimates. We, therefore, rewrite the condition in terms of new parameters. We define

\[ \gamma_j = \frac{1}{\theta^T t_i'(k + 1)!} \quad j = 1, ..., N \]

and \( e = \sum_{j=1}^{N} t_i'(k + 1)! \)

We rewrite the condition (4) in the form

\[ L \hat{\gamma} = e \]

where \( L = [t_i'(k + 1)! \cdots t_i'(k + 1)!] \)

\[ \gamma = [\gamma_1, \cdots, \gamma_N]^T \]

Then we solve the equation for an estimate of \( \gamma \) in terms of the pseudoinverse \( L^+ \) of the matrix \( L \)

\[ \hat{\gamma} = L^+ e \]

To find a parameter estimate \( \hat{\theta} \), we solve the equation

\[ M \theta = \gamma \]

where

\[ M = [t_i' \cdots t_i'] \]

\[ \gamma = [\gamma_1, \cdots, \gamma_N]^T \]

\[ \gamma_{\text{inv}} = [\gamma_{\text{inv}}^1, \cdots, \gamma_{\text{inv}}^N]^T \]

This choice of the function \( \Psi \) gives the distribution

\[ P_{\text{pois}}(t_i'(k+1)|t_i(k)) = \frac{(\theta^T t_i')^\gamma t_i'(k + 1)! e^{-\theta^T t_i'}}{t_i'(k + 1)!} \]

For \( N \) independent and identically distributed measurements, we have the likelihood function

\[ L(t_i'(k + 1)) = \prod_{j=1}^{N} \frac{(\theta^T t_i')^\gamma t_i'(k + 1)! e^{-\theta^T t_i'}}{t_i'(k + 1)!} \]

The natural logarithm of the likelihood function is

\[ \ln L(t_i'(k + 1)) = \sum_{j=1}^{N} \gamma t_i'(k + 1)! - \theta^T \sum_{j=1}^{N} t_i'(k + 1)! \]

To find the maximum-likelihood estimator of the parameters, we differentiate the log-likelihood, and equate to zero

\[ \frac{\partial \ln L(t_i'(k + 1))}{\partial \theta} \]  

We now have the condition

\[ \sum_{j=1}^{N} t_i'(k + 1)! \frac{\partial \theta}{\partial \theta} \]  

The condition yields an expression that cannot readily be solved for the parameter estimates. We, therefore, rewrite the condition in terms of new parameters. We define

\[ \gamma_j = \frac{1}{\theta^T t_i'(k + 1)!} \quad j = 1, ..., N \]

and \( e = \sum_{j=1}^{N} t_i'(k + 1)! \)

We rewrite the condition (4) in the form

\[ L \hat{\gamma} = e \]

where \( L = [t_i'(k + 1)! \cdots t_i'(k + 1)!] \)

\[ \gamma = [\gamma_1, \cdots, \gamma_N]^T \]

Then we solve the equation for an estimate of \( \gamma \) in terms of the pseudoinverse \( L^+ \) of the matrix \( L \)

\[ \hat{\gamma} = L^+ e \]

To find a parameter estimate \( \hat{\theta} \), we solve the equation

\[ M \theta = \gamma \]

where

\[ M = [t_i' \cdots t_i'] \]

\[ \gamma = [\gamma_1, \cdots, \gamma_N]^T \]

\[ \gamma_{\text{inv}} = [\gamma_{\text{inv}}^1, \cdots, \gamma_{\text{inv}}^N]^T \]
The parameter estimate is given by

\[ \hat{\theta} = M^t \varphi_{\text{DIV}} \tag{8} \]

Using the formula of the node conditional distribution, we calculate the expected values of \( \bar{t}(k+1) \) given \( \bar{t}(k) \), the traffic of the random channels at a time \( k \). The expected value is given by

\[ \bar{t}(k) = E\{\bar{t}(k)\} = \sum_{i=1}^{N} t_i P_{\text{min}}(\bar{t}(k+1)|t_i(k - 1)) = \sum_{i=1}^{N} \left[ |\bar{t}(k)| \right] \]

where \( \bar{t}(k) \) is the total capacity of the network. For large \( N \), we obtain

\[ \bar{t}(k) = \exp\{\theta_0 |\bar{t}(k) - 1| \} \tag{9} \]

The two-sided confidence interval of Poisson prediction is calculated using the score CI because this method gives better results for small samples [11]. Score CI is defined as

\[ \left[ t_{\text{min}}(k), t_{\text{max}}(k) \right] = \left( \frac{\lambda + z^2/2}{n} \right) \pm \frac{z}{\sqrt{n}} \left( \frac{\lambda + z^2}{4n} \right)^{1/2} \tag{10} \]

where \( t_{\text{min}}(k), t_{\text{max}}(k) \) are upper and lower bounds of the confidence interval for the random traffic at a time \( k \). \( \lambda \) is the mean value, \( z \) is the standard normal distribution percentile which is 100(1 - \( \alpha \)) and \( n \) is the number of samples.

Since the total bandwidth capacity is limited, bandwidth is optimally allocated based on the resource demands of individual channels. After predicting the traffic at the time \( k \) + 1, given the traffic of the event-driven channels, traffic in each network channels is \( \bar{t}(k+1) \). We allocate bandwidth for the random and event-driven channels. Let the allocated bandwidths be represented as \( x_j \), where \( j = 1, 2, \ldots, n \), where \( n \) is the number of the channels in the network. The bandwidth allocation problem is the \( l_i \) optimisation

\[ \min_{x} \| Ax - t(k+1) \|_1 \tag{11} \]

where \( A \) is a diagonal matrix whose diagonal is equal to the vector of weights \( a \). The vector of weights has entries between 0 and 1 with larger weights for more important channels.

We rewrite (5) as

\[ \min y \quad s.t. \quad -y \leq Ax - t(k+1) \leq y \]

and \( 1^T x = B \)

where \( B \) is the total capacity of a network with \( 1 = [1,1,\ldots,1]^T \). We now have the linear program

\[ \min \bar{x}^T x \quad s.t. \quad \bar{A} \bar{x} \leq \bar{t}(k+1) \]

and \( \bar{A} \bar{x} = \bar{B} \)

where

\[ \bar{A} = \begin{bmatrix} A & -I \\ -A & -I \end{bmatrix} \quad \bar{x} = \begin{bmatrix} x \\ y \end{bmatrix} \]

By assigning a smaller weight to the time-driven channel, the optimisation problem will allocate all other channels maximally if total demand exceeds capacity. To avoid network congestion, package drop, or delay, the sampling periods of the congested time-driven channels can be increased. However, the sampling period cannot exceed an upper bound required for acceptable NCS performance. Therefore, we design a control loop for the sampling period that reduces congestion, maintains NCS performance, and exploits the available bandwidth.

Let the traffic in the random channel at time \( k \) be \( \bar{t}(k) \). The traffic in the time-driven channel, in which all tasks and control actions are periodic, and sampling is at predefined points in time, is \( \bar{t}(k) \), and the traffic in the event-driven channel is \( \bar{t}(k) \). Using the node conditional distribution, the predicted traffic at \( k+1 \) is \( \bar{t}(k+1) \). Since \( \bar{t}(k+1) \) is known, we can change the sampling periods of each congested time-driven channel as follows.

First, all the channels in the network are allocated optimally using \( l_i \) optimisation. If \( \bar{t}(k+1) \neq x_j \), where \( x_j \) is the allocated bandwidth of the ith time-driven channel. We change the sampling period of the time-driven channels to satisfy the bandwidth constraint of the network. Then we calculate the following estimate the sampling period of the time-driven channels

\[ h_i(k+1) = \frac{\epsilon (B - \sum_j x_j) - \bar{t}(k+1)}{\epsilon \sum_j x_j} \tag{12} \]

where \( t_{\text{min}}(k+1) \) is the upper bound of 95% confidence interval for the random traffic at the time \( k+1 \). \( r_i \) is the information transmission time for the ith control loop and \( \epsilon \) is the safety factor required for a satisfactory result. The safety factor can be considered as a design parameter for the network in the range [0, 1] depending on our confidence in the traffic estimate.

The upper bound for the sampling period of the ith time-driven channel is a constant \( h_{\text{OPT}} \), which is the sampling period the ith channel required for the stability and performance of the NCS and \( h_{\text{OPT}} \) is the min sampling period, which is selected based on hardware limitations. The sampling period of the time-driven channels are calculated as

\[ h_i(k+1) = \min \left\{ \max \left\{ h_{\text{OPT}}, \frac{\bar{t}(k+1)}{h_{\text{OPT}}} \right\} \right\} \tag{13} \]

Lemma 1: The sampling period control system using (13) is bounded-input–bounded output (BIBO) stable.

Proof: The result follows from the fact that for any bounded input, the output of the system is upper bounded by \( h_{\text{OPT}} \) and lower bounded by zero. □

Clearly, if \( h_i(k) \) has a lower bound \( h_i \), then the sampling period for the time-driven channels remains bounded in the range \( [h_i, h_{\text{OPT}}] \). However, the control loop for adjusting the sampling period is not Lyapunov stable because we cannot impose an arbitrarily small bound on the sampling period by the appropriate choice of its initial value. In fact, increasing the range in which the sampling period can vary allows the system to effectively use the available bandwidth.

3 Results and discussion
We apply our methodology to a DC motor speed control example and compare our results to the bandwidth allocation strategy of [8]. The first-order state-space model is [6, 12]


\[ x(t) = 0.01x(t) + u(t) \]

where \( x(t) \) is the state and \( u(t) \) is the input. The closed-loop system includes a communication network. A sensor samples the output and generates discrete-time data \( x(kT), k = 1, 2, \ldots, n \), where \( T \) is the sampling period. The sensor sends the data packets to the controller through the network, and the controller generates the control \( u(kT) \) and sends it to the actuator

\[ u(kT) = -K(R - x(kT)) \]

where \( K \) is the controller gain and \( R \) is a reference input.

The authors in [6, 13] propose a utility function for a first-order system. The utility function for the \( i \)th control loop is

\[ U_i(r) = \frac{a_i - b_iK_i}{a_i}e^{-\gamma_1r} \]

where \( r \) is the transmission rate. The term

\[ 1^T \sum_{i=1}^n U_i(r) \]

determines the stability region for the PI controller gains. Since the paper seeks to allocate bandwidth based on random traffic, only one control loop is used in the simulation for simplicity.

The controller gain is \( K = 1.5 \) and the gains for the PI controller are selected as \( K_1 = 3, K_4 = 0.01, N = 100 \).

The steady-state transmission rate for the control loop is \( r_0 = 45 \) pack/10 ms and the information transmission time of the control loop is 10 ms.

The following fixed network characteristics and parameter values are set for the simulation: The frequency-division multiple access (FDMA) network channels have a capacity of 1.8 Mbps and packet size is 80 bit, 225 pac/10 ms, the traffic in the three random channels is Poisson distributed with mean 40, 50, and 55 pac/10 ms, respectively, and the sensor and the controller channel generate 45 pac/10 ms.

Each channel has the same channel capacity of 225/5 = 45 pac/10 ms. We select the fixed sampling period of the time-driven sensor channel as 0.004 s. Since the allocated bandwidths for the controller and the sensor channels are less than their traffic depending on the traffic at random channels, the channel capacity of the sensor and the controller channels are not sufficient to send data without delay or packet drop. This will adversely affect the performance of the NCS (see Fig. 2).

Assuming five nodes are sharing a network, three nodes are generating random traffic and disturbing the NCS of Figs. 1 and 3; the other two nodes are deterministic and include the time-driven sensor node and event-driven controller node.

Using the initial traffic of the random network, we predict the next traffic volume using (2)–(10). Other channels are deterministic and their traffic can be easily calculated, using the same fixed parameter of the network.

The three random nodes generate Poisson distributed random traffic with the means of 40, 50, and 55 pac/10 ms. The time-driven sensor contains a periodic task where the traffic is 300 pac/10 ms. At each invocation, it samples the process and transmits the sample package to the controller node. The controller contains an event-driven task that is triggered each time a sample arrives over the network from the sensor node. The traffic in the controller to the actuator channel is 45 pac/10 ms. Upon receiving the sample, the controller computes a control signal, which is then sent to the event-driven actuator node.

After predicting the traffic of the random channel at a time \( k + 1 \), we have values for all network traffic that can be used for bandwidth allocation. Let the allocation of the bandwidths be \( t(k+1) = [t'(k+1), 45, 45] \), then the convex optimisation problem is solved using (11).

For the initial random channel traffic vector at the time \( k \) of \( t'(k) = [50, 41, 53] \) pac/10 ms, the predicted traffic at a time \( k + 1 \) is

\[ t'(k + 1) = [48, 45, 56] \text{ pac/10 ms} \]

The upper bound for the sampling period of the time-driven channel is \( T_\text{max} \), required for satisfactory NCS is calculated based on the system dynamics. The desired response is required to have a 12% maximum overshoot and 5 ms settling time. For 12% overshoot, we need a damping ratio of about 0.2155. We calculate the undamped natural frequency for the NCS \( \omega_d = 1185 \) rad/s. However, since packet arrival time to the controller must be considerably faster than the natural frequency of the closed-loop system from the specifications as \( \omega_0 = 1403 \text{ rad/s} \). We calculate the upper bound for the sampling period of the time-driven channel as

\[ h(k + 1) = 0.95(225 - 48 - 45 - 56 - 45) = 0.0034 \]

The upper bound for the sampling period of the time-driven channels \( h_\text{max} \), required for satisfactory NCS is calculated based on the system dynamics. The desired response is required to have a 12% maximum overshoot and 5 ms settling time. For 12% overshoot, we need a damping ratio of about 0.2155. We calculate the natural frequency of the closed-loop system from the specifications as \( \omega_0 = 1403 \text{ rad/s} \). However, since packet arrival time to the controller must be considerably faster than the undamped natural frequency \( \omega_d = 1185 \text{ rad} \), the required sampling frequency for the NCS \( \omega_s \) is calculated as \( \omega_s = \omega_0 \alpha \), where \( \alpha \) a scale factor determined is based on a well-known rule of thumb. We selected \( \alpha = 1.5 \) to obtain

4

IET Cyber-Phys. Syst.. Theory Appl.

This is an open access article published by the IET under the Creative Commons Attribution License (http://creativecommons.org/licenses/by/3.0/)
\[ h_{\text{net, max}} = \frac{2\pi}{\alpha_{\text{avg}}} \approx 0.0035 \]

To ensure the required output response max sampling period should be 4 ms and a minimum sampling period is limited to 15 ms. The adjusted sampling period of the time-driven channels is calculated using (13) as

\[ b_i(k+1) = \min \{ \max (0.0015, 0.0034), 0.0034 \} = 0.0034 \]

For each time step, we apply the same procedure to eliminate channel congestion.

We compared the proposed dynamic bandwidth allocation strategy to the strategy of [8]. In [8], bandwidth allocation direction (increasing or decreasing) is determined using the error calculated by subtracting the scheduling period and the sampling period of the related control loop. The mean absolute error for each loop is calculated as

\[
e_i = \frac{1}{M_i} \sum_{k=1}^{M_i} |e(k)|
\]

where \( e_i \) is the absolute mean error for the \( i \)th loop at the \( j \)th. The scheduling period, \( M_i \) is the number of errors between \( b_i \) and \( b_i^+ \) control period for the related loop and the sampling scheduling period.

Based on the mean absolute error, the desired bandwidth \( b_i^{+1} \) and required bandwidth increment \( \Delta b_i^{+1} \) are calculated as follows:

\[
\begin{align*}
\Delta b_i^{+1} &= \left\{ \begin{array}{ll}
\beta_i^{+1} (b_i^m - b_i) & \beta_i^{+1} \geq 0 \\
\beta_i^{+1} (b_i - b_i^m) & \beta_i^{+1} < 0
\end{array} \right. \\
b_i^{+1} &= b_i + \Delta b_i^{+1}
\end{align*}
\]

where \( \beta_i^{+1} \) is the bandwidth feedback coefficient \( \beta_i^{+1} \in [-1, 1] \). \( \beta_i^{+1} \) is calculated using the absolute error as follows:

\[
\Delta b_i^{+1} = \left\{ \begin{array}{ll}
\frac{|e_i|^2 - \mu_i}{\mu_i} |e_i| + \mu_i & \mu_i \leq |e_i| \\
-\mu_i - 2\mu_i - 1 & \mu_i > |e_i|
\end{array} \right.
\]

where \( \mu_i \) is allowed stable boundary conditions.

Finally, the desired bandwidth \( b_i^{+1} \) is calculated as follows: (see equation below). Using the same system and network parameters, the bandwidth allocation strategy of [8] and the proposed bandwidth allocation strategy are simulated utilising the MATLAB TrueTime Network Simulator.

We first use fixed network parameters without bandwidth allocation and sampling rate adjustment. In this case, network congestion occurs since the traffic of some channels exceeds their capacity based on random traffics. The closed-loop response of NCS does not satisfy the desired system response because network congestion causes packet loss and delay which disrupts the control, as can be seen in Fig. 3.

\[
b_i^{+1} = \left\{ \begin{array}{ll}
b_i^{+1} + \Delta b_i^{+1} & \Delta b_i^{+1} > 0 \text{ and } \sum_{i=1}^{N} b_i + \Delta b_i^{+1} < B \\
b_i^{+1} + \frac{\Delta b_i^{+1}}{\sum_{i=1}^{N} \Delta b_i^{+1}} \left( B - \sum_{i=1}^{N} b_i - \sum_{i=1}^{N} \Delta b_i^{+1} \right) & \Delta b_i^{+1} < 0 \text{ and } \sum_{i=1}^{N} b_i + \Delta b_i^{+1} > B
\end{array} \right.
\]

In our approach, we allocate the available bandwidth based on the demand for each channel. Using \( l_i \) optimisation, we allocate bandwidth to the random and the event-driven controller channel. Since the total capacity is limited, the time-driven sensor channel may become congested. We change its sampling rate to satisfy its allocated bandwidth capacity subject to stability and performance constraints.

The random traffic in the network is predicted using the Poisson graphical model and the predicted values are used for optimal bandwidth allocation. Fig. 2 shows the accuracy of predicted random traffic in NCS and Fig. 4 gives upper and lower bounds for the prediction accuracy. The true network traffic lies in the confidence interval of the random traffic. Fig. 4 shows the predicted traffic with a confidence interval. Initially, there is a very short period where the predicted values are outside the confidence interval. The figure shows that prediction is sufficiently accurate after 0.04 s for the random channel-1, after 0.4 s for the random channel 2, and after 0.3 s for channel 3.

Fig. 5 shows that the output square wave response of the proposed algorithm provides the desired response characteristics and adequate network quality of services for the NCS with 7% overshoot and 4 ms settling time in the first pulse and only 2.6% overshoot and 4 ms settling time for the second pulse. In contrast, the approach of [8] results in 70% overshoot and 7 ms settling time for the first pulse and 74% overshoot and 8 ms settling time for the second pulse. Thus, while our approach meets the desired overshoot criterion for both pulses, the approach of [8] does not for both the first and the second pulses. The response in the second pulse improves for the proposed method because all the predictions are within the confidence interval. In contrast, the overshoot is worse in the second pulse for the method of [8] because it does not utilise traffic prediction.

Fig. 6 shows the cumulative usage of the bandwidth for both [8] and the proposed allocation algorithm. The figure shows that the proposed method allocates the bandwidth so as never to exceed the total available bandwidth capacity while the approach of [8] slightly exceeds the total capacity. The figure shows how network
congestion frequently occurs without bandwidth allocation. The proposed algorithm can mitigate congestion by predicting the random traffic to optimally allocate bandwidth. After optimal bandwidth allocation and sampling rate adjustment for the sensor channel, the congestion in the sensor and the controller channels are eliminated and the system remains stable with good performance. Bandwidth allocation in the approach of [8], without the benefit of prediction, results in packet delivery delays when the bandwidth allocated is less than the required bandwidth for the time driven sensor channel and a deterioration in the response of the NCS.

Figs. 7 and 8 show the global network scheduling for the time-driven sensor and event-driven controller channels in the Truetime network simulator for both the proposed method and the approach of [8] in the time interval [0,0.3]. A node can only transmit data when it is scheduled to state ‘high’ (running data). If the node has nothing to transmit, the network states ‘low’ (idle). The figures indicate that the schedule for the time-driven sensor and event controller channels for the proposed approach has more running status with a short-time period, i.e. more packets are transmitted. The time-driven sensor node samples the process periodically and sends the samples over the network to the controller node. Fig. 7 indicates that the proposed algorithm sufficiently allocates bandwidth for the time-driven sensor channel and changes the sampling rate based on the allocation. Fig. 8 indicates that the time-driven sensor node in the approach of [8] sends more delayed data to the sensor node compared to the time-driven sensor node in the proposed approach. The figures show that fewer packets are received by the event-driven controller node from the sensor node in the approach of [8]. This is because bandwidth deficiency at the related transmission time results in lost or delayed packets.

Fig. 9 shows the percentage of bandwidth allocation for each channel based on its resource demand for both our approach and the approach of [8]. The allocation for the time-driven sensor for our approach adapts to demand better than the approach of [8]. The reason is that our approach uses the predicted random network traffic to allocate bandwidth at a time $k+1$ while [8] cannot
predict random traffic. Our approach guarantees sufficiently fast sampling for NCS stability while providing better bandwidth utilisation.

Fig. 10 shows the sampling period of the sensor channel in the interval $[0, 0.5]$. After optimal bandwidth allocation, we change the sampling period of the time-driven sensor channel to prevent network congestion and to maintain a satisfactory output, as shown in Fig. 5. The sampling period barely reaches but does not exceed the critical level of the maximum allowed sampling period of 3.5 ms. Because of the limited capacity, it never reaches the minimum sampling period that causes network congestion.

Fig. 11 shows the variation of the computational time based on the assigned sampling period of the sensor between 0.16 and 0.2 s in the time interval $[0, 0.2]$. The mean value of the assigned sampling period is calculated as 31 ms, the computational time for the period is 0.1810 s, and the mean of the computational time is 0.1855. This shows that computational time depends on the assigned bandwidth and assigned sampling period. Although the assigned bandwidth also affects the computational time, the change in computational time with the sampling period is bounded and acceptable.

We simulated the system, including a dead zone in the ranges $[-0.1, 0.1]$, $[-0.15, 0.15]$, $[-0.2, 0.2]$, $[-0.25, 0.25]$ at the input of the plant for both the proposed method and the approach of [8]. The results of the simulation are shown in Fig. 12. The results show that the non-linearity affects the output transient response of the closed-loop NCS. However, the effect of the non-linearity on bandwidth allocation is minor after a short transient period, particularly in the dead zone range $[-0.1, 0.1]$. The figure shows a large overshoot for the method of [8], with a large peak overshoot for the interval $[-0.25, 0.25]$ that decreases for smaller ranges. A large overshoot results in significant disruption of the NCS.

4 Conclusions

This paper presents a new approach to bandwidth allocation in NCS. The approach optimises bandwidth allocation based on the predicted traffic flow for random channels so as to minimise packet loss. For high traffic volume, traffic is reduced by reducing the sampling rate for time-driven channels. However, sampling must be sufficiently fast to avoid destabilising control loops.

We compared our results for a network that includes a random channel occupying 10% of the bandwidth and is subject to bandwidth constraints to the method of [8]. The method calculates control errors utilising the global network scheduler and allocates bandwidth using the calculated control errors. Simulation results show that our approach can allocate bandwidth optimally and select sampling periods of the time-driven sensor channels to meet bandwidth constraints and reduce or eliminate congestion. In contrast, random error changes due to the random channel result in significant deterioration for the method of [8]. The results show that for the complex NCS that share a network with other nodes, random traffic may cause large errors due to the constraints on bandwidth allocation. Our proposed method provides a solution to this issue. Future work will consider the effect of cyberattacks on the NCS and improve its resilience.

5 References

[1] Dong, J., Kim, W.J.: ‘Bandwidth allocation and scheduling of networked control systems with exponential and quadratic approximations’, Control Eng. Pract., 2014, 26, pp. 72–81
[2] Arzen, K.E., Cervin, A., Eker, J., et al.: ‘An introduction to control and scheduling co-design’. Proc of the 39th IEEE Conf. on Decision and Control, Sydney NSW Australia, December 2000, vol. 5, pp. 4865–4870

[3] Ge, X., Yang, F., Han, Q.L.: ‘Distributed networked control system: a brief overview’, Inf. Sci., 2017, 380, pp. 117–131

[4] Zang, X.M., Han, Q.L., Yu, X.: ‘Survey on recent advances in networked control systems’, IEEE Trans. Ind. Inf., 2015, 12, pp. 1740–1752

[5] Gupta, R.A., Chow, M.Y.: ‘Networked control system: overview and research trends’, IEEE Trans. Ind. Electron., 2010, 57, (7), pp. 2527–2535

[6] Al-Hammouri, A.T., Branicky, M.S., Liberatore, V., et al.: ‘Decentralized and dynamic bandwidth allocation in networked control systems’. IEEE Parallel and Distributed Processing Symp., Greece, June 2006

[7] Roohi, M.H., Izadi, I., Ghaisari, J.: ‘Bandwidth allocation, delay bound analysis, and controller synthesis for event-triggered control systems’, IET Control Theory Appl., 2017, 11, (16), pp. 2870–2878

[8] Tao, Y., Wei, H., Yixin, Z.: ‘A dynamic allocation strategy of bandwidth of networked control systems with bandwidth constraints’. 2016 IEEE Advanced Information Management, Communicates, Electronic and Automation Control Conf. (IMCEC), China, October 2016, pp. 1153–1158

[9] Allen, G.L., Liu, Z.: ‘A local Poisson graphical model for interfering networks from sequencing data’, IEEE Trans. Nanobioscience, 2013, 12, (3), pp. 189–198

[10] Cervin, A., Henriksson, D., Ohlin, M.: ‘Truetime 2.0 – reference manual’, Department of Automatic Control, Lund University, April 2016, Sweden

[11] Tanisut, M.: ‘Two sided confidence interval for the Poisson means’, Int. J. Model. Optim., 2012, 2, (5), pp. 589–591

[12] Silva, G., Datta, A., Bhattacharyya, S.: ‘PID controllers for time-delay systems’ (Birkhauser, Basel, 2005, 1st edn.)

[13] Branicky, M., Phillips, S., Zhang, W.: ‘Scheduling and feedback co-design for networked control systems’. Proc. IEEE Conf. on Decision and Control, Las Vegas, NV, USA, 2002, vol. 2, pp. 1211–1217