Integrated predictive control and scheduling co-design for networked control systems

Y.B. Zhao, G.P. Liu and D. Rees

Abstract: A predictive control and scheduling co-design approach is proposed to deal with the controller and scheduler design for a set of networked control systems which are connected to a shared communication network. In the proposed approach, a predictive controller is applied to generate the control predictions for each system using delayed sensing data and previous control information, and a time delay compensator is designed at the actuator side to actively compensate for the network-induced delay in the forward channel when the control action is taken. Two different scheduling algorithms, the existing static rate monotonic (RM) scheduling algorithm and a new dynamic scheduling algorithm called dynamic feedback scheduling (DFS), are considered to schedule the transmissions of the control signals generated by the predictive controller, which are packed and transmitted to the actuator in one packet simultaneously. Both the scheduling algorithms are designed with the guarantee of the stability of all the systems, which is achieved by ensuring that the time delay of the systems do not exceed the upper bound under which the systems are stable. It is also pointed out that the RM algorithm is a special case of the proposed DFS algorithm, in the sense that the former can work only in a private network environment, whereas the latter extends its application to such networks where other components occupying the network. Simulations for both the RM and the DFS algorithms, illustrate the validity of the proposed approach.

1 Introduction

With the rapid development of control theory and communication technology, a new area of research activity called ‘networked control systems’ (NCSs) has received much attention in recent years. Although the notion of NCSs is quite new and the theory is still in its infancy, fruitful research work can be found in the most popular journals in both fields of control theory and communication networks, considering different issues in NCSs, such as the network-induced delay, data packet dropout, and so on. [1–4].

So far, most research work on NCSs, especially in dealing with the network-induced delay, which is introduced by the inserted network and greatly degrades the performance of the system at certain conditions, has been done by the control theory community. Various methodologies in conventional control theory, such as the theories of time-delay systems, switched systems, stochastic control, optimal control and so on, have found their applications in NCSs [5, 6]. In this kind of research, the characteristics of the network are assumed to be given in advance, and thus a conventional time-delay system, rather than an NCS is considered.

However, it is the communication network which replaces the direct connections between sensors, controllers and actuators in conventional control systems that makes NCSs distinct from the latter. Thus, it is necessary to take the characteristics of the network into account in the study of NCSs, and actually, this kind of research (the so called ‘co-design’ approach) has been an emerging trend in recent years. The co-design approach to NCSs generally considers NCSs with such communication constraints as network-induced delays, data packet dropout, medium access constraints and so on, which are not assumed to be given as parameters or constraints for conventional control systems in advance, but act as desirable factors using techniques of communications and networks. It is therefore reasonable to expect that a better performance can be obtained using the co-design approach. For further information of the co-design approach to NCSs, the reader is referred to recent papers [7–15] and the references therein.

In this paper, the design and analysis of a set of linear NCSs which share the network with limited resource to transmit their control signals, are considered. (See Fig. 1 for the general configuration. Note that there are also random delays in the backward channel which is not shown in Fig. 1.) A similar problem setup can be found in [16, 17]. Hristu Varsakelis and Kumar [16] use the technique of ‘communication sequence’ (see also in [18, 19]) to deal with medium access constraint for such a system configuration and model the subsystems as switched systems with two modes ‘open loop’ and ‘closed loop’ which switch according to whether the current subsystem has access to the medium or not. Branicky et al. [17] consider a special case of Fig. 1 where the channel from controller to actuator is linked directly, and the rate monotonic (RM) scheduling algorithm is applied to schedule the transmissions of the sensing data of the subsystems. Both of the papers do not explicitly take the network-induced delay nor data packet dropout into consideration. In this paper, however, we will consider all the communication constraints, network-induced delay, data...
packet dropout and medium access constraint, for the system configuration shown in Fig. 1. To this end, a co-design approach is proposed with the integration of the model predictive control (MPC) method and the scheduling algorithm.

In conventional time-delay systems (TDS), there are mainly two ways to deal with the case when there is no current control signal available at the plant side due to the delay. This is to use either the last control signal or zero control. In both methods, the previous information of the system, including the system states, outputs and inputs and the structure information of the system have not been considered. However, if this information is well organised to derive a predictive control signal, it is reasonable to expect that a better performance can be obtained. On the basis of this insight, a modified MPC method is applied to design the predictive controller and a time-delay compensator is used at the actuator side to compensate for the network-induced delay in NCSs [20]. This method is validated using both simulation and a practical experiment. A similar idea is also used to deal with the network-induced delays and medium access constraints in [21]. In this paper, different from the input–output form in [20], a modified predictive control method in state-space form is applied and a delay compensation scheme is designed at both the controller and the actuator sides to compensate for both the network-induced delay and data packet dropout.

In order to optimise the resource allocations of the shared network, scheduling theory is applied in this paper to allocate the medium access of the transmissions of the predictive control sequence [22, 23]. Two different scheduling algorithms, the existing static RM algorithm and a novel dynamic feedback scheduling (DFS) algorithm, are adopted to schedule the communications under different environments by defining carefully the transmission task for each system, with the guarantee of the stability of all the systems by using the notion of ‘Stable Supremum of Round Trip Time (SSRTT)’.

The remainder of the paper is organised as follows. The problem being studied is described in Section 2; then the design of the predictive controller and the time delay compensator is given in Section 3. After that the scheduling algorithms are presented in Section 4, and the simulation results are illustrated in Section 5. Section 6 concludes the paper.

2 Problem description

Consider a set of N continuous-time LTI systems $(S^i_j)_{1 \leq i \leq N}$ described by the state equations

\[
S^i_j: \begin{cases}
   \dot{x}^i_j(t) = A^i_j x^i_j(t) + B^i_j u^i_j(t) \\
   y^i_j(t) = C^i_j x^i_j(t)
\end{cases}
\]

where $x^i_j(t) \in \mathbb{R}^{n_i}$, $u^i_j(t) \in \mathbb{R}^{m_i}$ and $y^i_j(t) \in \mathbb{R}^{r_i}$. In a digital control environment, a discrete-time representation $S^i$ of system $S^i_j$ is obtained using a sampling period $T_i$

\[
S^i: \begin{cases}
   x^i(k+1) = A^i x^i(k) + B^i u^i(k) \\
   y^i(k) = C^i x^i(k)
\end{cases}
\]

with $x^i(k) = x^i_j(k T_i)$, $u^i(k) = u^i_j(k T_i)$, $y^i(k) = y^i_j(k T_i)$, $A^i = e^{A^i T_i}$ and $B^i = \int_0^{T_i} e^{A^i s} ds B^i_j$.

Suppose that the time delays in the backward channel of all the systems are random but bounded and the transmissions from the controllers to the actuators share a communication network with limited resource. The communication resource is limited in the sense that, at each time instant, only one controller can access the network for transmission. Thus, the time delay in the forward channel depends not only on the delay when the data are transmitted through the network but also on the scheduling algorithm used for the medium access control of the transmissions of all the systems (Fig. 1).

Thus the problem here is not only to design a controller for each system $S^i$ but also to design the scheduling scheme for the communication resource allocation for all the systems $(S^i_j)_{1 \leq i \leq N}$, in an environment of network-induced delays and data packet dropouts.

3 Predictive controller and time delay compensation scheme design

In this section, the predictive controller with a delay compensation scheme at the controller side and a modified delay compensation at the actuator side for each system are first presented and then the stability theorem of the closed-loop system is obtained, from which the important notion SSRTT is derived. This notion will be used in the scheduling algorithm design covered in the next section.

Note that though the design is for a separate system $S^i$, the sub-system $i$ of all the parameters for system $S^i$ is ignored in this section for the simplicity of notation due to the fact that the design method is exactly the same for all the systems $(S^i)_{1 \leq i \leq N}$.

3.1 Design of the predictive controller

Assume that the objective function has the form

\[
J(N_1, N_2, N_0) = \sum_{j = 1}^{N_0} \delta_j \left( y_j(k + j - \tau_{wc, k}) - \alpha_j(k + j) \right)^2 + \sum_{j = 1}^{N_2} \lambda_j \left( \Delta u_k(k + j - 1) \right)^2
\]

where $N_1$ and $N_2$ are the minimum and maximum prediction horizons, $N_0$ is the control horizon, $\delta_j, j = N_1, \ldots, N_2, \lambda_j, j = 1, \ldots, N_2$, the weighting factors, $\alpha_j(k + j), j = N_1, \ldots, N_2$, the set points, $\Delta u_k(k + j - \tau_{wc, k}), j = N_1, \ldots, N_2$, the predictive outputs based on previous information up to time $k - \tau_{wc, k}$ and $\tau_{wc, k}$ the time delay in the backward channel at time $k$.

Let $\tilde{x}(k) = x(k) u(k - 1)\dagger$, then system $S^i$ (any system chosen from $(S^i)_{1 \leq i \leq N}$) can be represented by $S^i$

\[
S^i: \begin{cases}
   \tilde{x}(k+1) = M \tilde{x}(k) + N \Delta u(k) \\
   y(k) = Q \tilde{x}(k)
\end{cases}
\]

where $M = \left( \begin{array}{cc} A & B \\ 0 & I \end{array} \right)$, $N = \left( \begin{array}{c} B \\ I \end{array} \right)$, $Q = \left( \begin{array}{c} C \\ 0 \end{array} \right)$ and $\Delta u(k) = u(k) - u(k - 1)$.
Thus the \( j \) step forward output prediction at time \( k' \) is

\[
\hat{y}(k' + j|k') = Q M^j \hat{x}(k') + \sum_{l=0}^{j-1} Q M^{j-1-l} N \Delta u(k' + l)
\]

Let \( j = j + \tau_{sc,k}, \) \( k' = k - \tau_{sc,k} \) and \( l' = l + \tau_{sc,k} \), then the predictive outputs at time \( k \) based on the information of the state up to time \( k - \tau_{sc,k} \) and the control sequence from \( k_{sc,k} \) is

\[
\hat{y}(k + j|k - \tau_{sc,k}) = Q M^{j+\tau_{sc,k}} \hat{x}(k - \tau_{sc,k}) + \sum_{l=0}^{j-1} Q M^{j-1-l} N \Delta u(k + l)
\]  \hspace{1cm} (2)

If the state vector \( x \) is not available, an observer must be included

\[
\hat{x}(k + 1|k) = A \hat{x}(k|k - 1) + Bu(k) + L(y_u(k) - C \hat{x}(k|k - 1))
\]  \hspace{1cm} (3)

where \( y_u(k) \) is the measured output. If the plant is subject to white noise disturbances affecting the process and the output with known covariance matrices, the observer becomes a Kalman filter and the gain \( L \) is calculated solving a Riccati equation.

In [20], the previous control sequence \( u(k - 1), \ldots, u(k - \tau_{sc,k}) \) is used to calculate the predictive control sequence at the controller side at time \( k \). However, the previous control signals from \( u(k - \tau_{sc,k}) \) to \( u(k - 1) \) are not available for the controller due to the random time delay in the forward channel. As will be discussed further in Section 3.2, in a networked predictive control environment, a sequence of future control signals is packed to send to the actuator, and the actuator only picks out one from the sequence of the data corresponding to the specific time delay in the forward channel. It can therefore be seen that the controller does not know the real control signal adopted by the actuator unless it receives the information about the previous control signals applied to the actuator. Only in the special case where there is no delay in the forward channel, the previous control sequence is known immediately by the controller. Therefore in this paper, we develop a new method to deal with this problem, in which only the control and output information before \( k - \tau_{sc,k} \) are used to generate the predictive control sequence by including the control sequence from time \( k - \tau_{sc,k} \) to \( k - 1 \) as part of the predictive control sequence.

Let \( \hat{y}(k - \tau_{sc,k}) = [\hat{y}(k + N_1|k - \tau_{sc,k}) \cdots \hat{y}(k + N_2|k - \tau_{sc,k})]^T \), \( \Delta u(k - \tau_{sc,k}) = [\Delta u(k - \tau_{sc,k} - N_2|k - \tau_{sc,k}) \cdots \Delta u(k + N_u - 1|k - \tau_{sc,k})]^T \), then

\[
\hat{y}(k - \tau_{sc,k}) = E \hat{x}(k - \tau_{sc,k}) + F \Delta u(k - \tau_{sc,k})
\]  \hspace{1cm} (4)

where \( E = [(QM^{N_1+\tau_{sc,k}})^T (QM^{N_1+\tau_{sc,k}+1})^T \cdots (QM^{N_2+\tau_{sc,k}})^T]^T \) and \( F \) is a block lower triangular matrix with its non-null elements defined by \( (F)_{ij} = Q M^{N_i+\tau_{sc,k}+i-j-1} I_{N_j} \), \( i - j \leq N_1 + \tau_{sc,k} - 1 \). Let \( \sigma_k = (o(k + N_1) \cdots o(k + N_2))^T \), then the optimal control increment sequence from \( k - \tau_{sc,k} \) to \( k + N_u - 1 \) can be calculated by letting \( \partial \ell(\cdot) / \partial u(k) = 0 \), using equations (1) and (4)

\[
\Delta U^*(k|k - \tau_{sc,k}) = (F \hat{x}(k - \tau_{sc,k}))^T W_1 \hat{y}(k - \tau_{sc,k})
\]  \hspace{1cm} (5)

where \( W_1 \) and \( W_2 \) are the weighting matrices, which are diagonal with the entries \( (W_1)_{ii} = \delta_{i1} \), \( i = 1, 2, \ldots, N_2 - N_1 + 1 \) and \( (W_2)_{ii} = \lambda_i, i = 1, 2, \ldots, N_u \). The optimal predictive control sequence from \( k \) to \( k + N_u - 1 \) is

\[
U^*(k|k - \tau_{sc,k}) = Gu(k - \tau_{sc,k} - 1|k - \tau_{sc,k}) + H \Delta U^*(k|k - \tau_{sc,k})
\]  \hspace{1cm} (6)

where \( U^*(k|k - \tau_{sc,k}) = [u(k - \tau_{sc,k}) \cdots u(k + N_u - 1|k - \tau_{sc,k})]^T \), \( G = I_{m} \cdots I_{m} l_{m}^T I_{m} \cdots I_{m} l_{m}^T \), \( mN_{s} \), \( m \) is the identity matrix with rank \( m \) and

\[
H = \begin{bmatrix}
I_m & \cdots & I_m & 0 & \cdots & 0 \\
I_m & \cdots & I_m & I_m & \cdots & 0 \\
\vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\
I_m & \cdots & I_m & I_m & \cdots & I_m
\end{bmatrix}_{mN_s \times m(N_s + \tau_{sc,k})}
\]

Remark 1: Although we have not specially pointed this out earlier, it is a fact that the complexity of the calculation of the control predictions \([(5) \text{ and } (6)] \) seriously depends on the backward channel delay \( \tau_{sc} \) since the matrices \( E, F \) and \( H \) vary with \( \tau_{sc} \). Thus, for the online implementation, it is a great burden for the controller to calculate the control predictions if \( \tau_{sc} \) varies over a large range. However, all these matrices, actually, can be calculated offline for a given \( \tau_{sc} \) given the nature of these matrices. This advantage enables us to calculate offline all the matrices relating to the specific \( \tau_{sc} \), store them in the controller and just pick out the appropriate ones when calculating online the control predictions, according to the current value of \( \tau_{sc} \), which can be known to the controller by using a time stamp for each sensing data packet as described in the following time-delay compensator design.

### 3.2 Design of the time-delay compensator

The network introduces to the NCSs delays which greatly degrade the performance of the system, even making the system unstable under certain conditions, while at the same time, the network also brings an advantage to the system in that a sequence of signals can be packed and transmitted simultaneously \([6, 20]\). The time-delay compensator adopted in this paper takes advantage of this property of NCSs.

The following assumptions are made in the design of the time delay compensator.

1. For the sake of the calculation of the predictive control sequence, the time delay of the backward channel, \( \tau_{ca} \), is known to the controller, which can be easily done by issuing a time stamp on each data packet from the sensor side to the controller side.
2. The round trip time (RTT), noted by \( \tau \), the total time delay of a packet, i.e. \( \tau = \tau_{sc} + \tau_{ca} \), is known to the actuator, which can also be done by using the time stamps.
3. The predictive control sequences are packed and transmitted to the actuator simultaneously.
4. The sum of the forward time delay \( \tau_{sc} \) and the maximum of continuous data packet dropout is less than the control horizon \( N_u \).

The time delay compensator works as follows. At every time instant \( k \), the predictive controller calculates a sequence of predictive control signals based on the outputs and control sequence up to time \( k - \tau_{sc,k} \), the future control sequence part of which \( U^*(k - \tau_{sc,k}) \) (6) is then transmitted in one packet to the actuator with a time stamp \( k \) and its backward channel delay \( \tau_{sc,k} \). When a packet of a control sequence arrives at the actuator side (different packets may experience different time delays), it is compared with the one already in the cache of the actuator according to the time stamp and only the later one is saved.

As for the actuator, it picks out the control action \( u(k + \tau_{ca,k} - \tau_{ac,k}) \) from the control sequence in its cache if the time stamp of the control sequence in its cache is \( k \).

Note that the time instant \( k \) in the time-delay compensator described above is based on the controller. Let \( \tau_{ac,k} \) denote the time delay in the forward channel of the control sequence which is applied to the actuator at time instant \( k \) (the time at the plant side), then the time stamp of this sequence (the time when it is sent at the controller side) is

\[
k^* = k - \tau_{ac,k} = \max_j \{ j | U^*(j) / \tau_{ac,k} \in \Gamma_k \} \tag{7}
\]

where \( \Gamma_k \) is the set of the control sequences that are available at time interval \( (k - 1, k] \) at the actuator side.

From (5), (6) and (7) and the definition of \( \bar{x}(k) \), the control signal adopted by the actuator at time \( k \) is obtained as

\[
u(k) = d_{ca,k}U^*(k - \tau_{ca,k} - \tau_{ac,k})
\]
\[
= d_{ca,k}G(k - \tau_{ac,k} + \tau_{ac,k} - \tau_{ac,k}) + d_{ca,k}H(k - \tau_{ac,k} - \tau_{ac,k})
\]
\[
= u(k - \tau_{ac,k} - \tau_{ac,k}) + d_{ca,k}H(F^T W_2^2 + W_2)^{-1}
\]
\[
\times F^T W_2 (\sigma_{k - \tau_{ac,k}} - E_k(k - \tau_{ac,k}))
\]
\[
= (I_m - K_{ch}E_2)u(k - \tau_{ac,k} - \tau_{ac,k})
\]
\[
+ K_{ch}E_1k - \tau_{ac,k} - \tau_{ac,k} - E_1x(k - \tau_{ac,k} - \tau_{ac,k}) \tag{8}
\]

where \( d_{ca,k} \) is a \( N \times N \) block matrix with all entries are 0 except the \( \tau_{ca,k} \) th row of \( I_{m} \), \( \tau_{ac,k} \) is the RTT with respect to \( \tau_{ac,k} \), that is, \( \tau_{ac,k} = \tau_{ac,k} + \tau_{ac,k} + \tau_{ac,k} \) the delay in the backward channel corresponding to \( \tau_{ac,k} \). \( K_{ch} = d_{ca,k}H(F^T W_2^2 + W_2)^{-1}F^T W_2, \) and \( E \) can be written as \( E = [E_1, E_2] \) with appropriate dimensions such that

\[
E_k(k - \tau_{ac,k} - \tau_{ac,k}) = [E_1, E_2] \left( \frac{x(k - \tau_{ac,k} - \tau_{ac,k})}{u(k - \tau_{ac,k} - \tau_{ac,k})} \right) = E_1x(k - \tau_{ac,k} - \tau_{ac,k})
\]
\[
+ E_2u(k - \tau_{ac,k} - \tau_{ac,k}) \tag{8}
\]

### 3.3 Stability analysis

It is assumed in this section that the RTT is bounded by a finite value \( \bar{\tau} \), \( \bar{\tau} = \bar{\tau}_{ca} + \bar{\tau}_{sc} \), where \( \bar{\tau}_{ca} \) and \( \bar{\tau}_{sc} \) are the upper bounds of the delay in the forward and backward channels respectively, and \( \bar{\alpha} = 0 \) without loss of generality. Let \( U(k) = [u(k - 1) \cdots u(k - \bar{\tau} - \bar{\tau} - 1)]^T, X(k) = [x(k) \cdots x(k - \bar{\tau})]^T \) and \( Z(k) = [X^T(k)U^T(k)]^T \), then an enhanced system can be obtained from (8) and the system description of \( S_i \) [any system chosen from \( (S_i)_{1 \leq i \leq N} \)]

\[
Z(k + 1) = A(\tau_{sc,k}^*, \tau_{ac,k}^*)Z(k) \tag{9}
\]

where

\[
\Lambda(\tau_{sc,k}^*, \tau_{ac,k}^*) = \begin{pmatrix}
A & BK_{c1}^2 & BK_{c2}^2 \\
I_m & 0 & 0 \\
& K_{c1}^2 & K_{c2}^2 \\
& I_m & 0 & \vdots & I_m
\end{pmatrix}
\]

which varies with \( \tau_{ac,k}^* \) and \( \tau_{sc,k}^* \). \( K_{c1}^2 = -K_{c1}E_1, K_{c2}^2 = I_m - K_{c2}E_2, \) and \( I_n \) is the identity matrix with rank \( n \). If the state vector is not available, then the state observer (3) is also included in the enhanced system.

The theory of switched systems is applied to derive the following stability theorem. A similar theorem can also be found in [20].

**Theorem 1:** The closed-loop system (9) is stable if there exists a positive definite matrix \( P \) such that

\[
\Lambda^T(\tau_{sc,k}^*, \tau_{ac,k}^*)PA(\tau_{sc,k}^*, \tau_{ac,k}^*) - P \leq 0 \tag{10}
\]

for all \( \tau_{sc,k}^* \in \{0, 1, \ldots, \bar{\tau}_{ca}\} \) and \( \tau_{ac,k}^* \in \{0, 1, \ldots, \bar{\tau}_{sc}\} \).

**Proof:** Let \( V(k) = Z^T(k)PZ(k) \) be a Lyapunov candidate. Then the incremental \( V \) for system (9) is

\[
\Delta V(k + 1) = Z^T(k)\Lambda(k)Z(k) - PZ(k)
\]

\( \leq 0 \), thus enabling the theorem to be proved given the assumption above.

**Remark 2:** Theorem 1 implies that the linear system \( S_i \) which uses MPC method and the delay compensation scheme described above to compensate for the network-induced delay and data packet dropout in NCSs (called a ‘network predictive control system (NPCS)’ hereafter), is stable under certain conditions if the RTT is less than a fixed value (Fig. 2). In other words, given a linear system, the least upper bound, or the supremum of the RTT, under which the system is stable can be found from Theorem 1. We call this supremum of RTT that guarantees the stability of the system the SSRTT, which is an inherent characteristic of a given system (Note that the notion of SSRTT here is similar to ‘maximum allowable delay bound’ (MADB) which has been used in a number of papers; see [24] for an overview. We prefer SSRTT to MADB in this paper since the former can better express the particular requirement of the RTT in Theorem 1 for the stability of the system.) Techniques such as the LMI tool-box are useful in the process of finding the SSRTT of a given system. It is also necessary to point out that if other performance constraints besides stability are considered, a smaller supremum of RTT than SSRTT is needed. For convenience, denote the SSRTTs of the systems \( (S_i)_{1 \leq i \leq N} \) by \( \bar{D}_i > 0, 1 \leq i \leq N \).

### 4 Scheduling

In this section, the scheduling theory is applied to allocate the limited network resource for the transmission of the control information for the systems \( (S_i)_{1 \leq i \leq N} \) in the forward channel. The static, priority-based scheduling algorithm RM is applied to the set of systems to schedule their transmissions under a private network environment first, and then a dynamic, feedback-based scheduling algorithm DFS is presented to extend the application to the public network.

![NPCS with a time-delay compensator](image)
When a scheduling algorithm is applied to schedule the transmission tasks of a set of NPCSSs, the stability of the systems need to be guaranteed as a precondition. To ensure this, the concept of ‘stable schedulability’ is defined.

Stable schedulability: A set of NPCSSs sharing the network resource is said to be stable schedulable by a scheduling algorithm if the transmissions of all the systems can be scheduled so that all the systems are stable.

4.1 Static scheduling

In the static scheduling case, the network is assumed to be used only by the systems \((S_i)_{1 \leq i \leq N}\), that is, it is private to the set of systems. Since the transmissions of the predictive control sequences for each system are viewed as different real-time tasks in scheduling theory, in the following the transmission tasks of the NPCSSs will be defined by analysing the formation of the network-induced delays of NPCSSs first, and then the RM algorithm is described over these transmission tasks and the feasibility theorem is obtained as well.

4.1.1 Transmission tasks of the NPCSSs: As shown in [25], the time delay of the forward channel \(\tau_{ca}\) is mainly composed of the following three parts.

1. The propagation delay, which is the time from when a packet is put onto the network till it successfully arrives at its destination. Since the network is private to the systems \((S_i)_{1 \leq i \leq N}\), the propagation delay depends merely on the speed of signal transmission and the distance between the source and the destination, which are fixed in the system model of this paper, and hence this delay is assumed to be known as a constant \(\tau_{ca}\) in the static scheduling case.

2. The frame time delay, which is the time for the source to place a packet on the network. Suppose that the size of the packet which contains the predictive control sequence is expressed as \(\gamma N_u\), where \(\gamma\) is a constant to the number of bytes contained in the one step ahead predictive control signal. This can be viewed as the same for all the systems, then the frame time delay for a transmission of system \(S_i\) is

\[
e_i = \frac{N_u}{B}\gamma
\]  

where \(B\) is the bandwidth of the network and \(N_u\) the control horizon of system \(S_i\). In the networked predictive control scheme proposed in Section 3, it is normal to assume that all the systems use the same control horizon \(N_u\), since the selection of \(N_u\) mainly depends on the RTT of the network and all the systems endure similar network-induced delays and data packet dropouts by sharing the network. Hence

\[
e_i = e = \frac{\gamma N_u}{B}, \quad i = 1, 2, \ldots, N
\]

The frame time delay \(e\) serves as the execution time in the transmission tasks of the NPCSSs.

3. The waiting time delay is defined as the time a predictive control sequence has to wait for queuing and network availability before actually being sent. From Remark 2, the SSRTT for system \(S_i\) is \(\tilde{D}_i\), and therefore to ensure the stability of all the systems, the waiting delay for the transmission of the control signals of each system should not be larger than

\[
D_i = \tilde{D}_i - \frac{\varphi}{\tau_{ca}} - \tau_{nc} - e
\]  

Note here that the upper bound of the delay in the backward channel \(\tau_{nc}\) is used since the RM algorithm assigns the priority of each task statically and the stability of the systems needs to be guaranteed under the worst case. It is only the stability of the system that we care about in the RM algorithm, and thus the transmission period of system \(S_i\) needs to be no longer than \(D_i\). Therefore the transmission period \(h_i\) of system \(S_i\) is assumed to be equal to \(D_i\) in the static RM scheduling algorithm, that is

\[
h_i = D_i, \quad i = 1, 2, \ldots, N
\]

4.1.2 Scheduling of NPCSSs by RM: RM is a widely used scheduling algorithm, where tasks with shorter periods have higher priorities. It is a fixed-priority assignment: priorities are assigned to tasks before execution and do not change over time. Liu and Layland [26] have shown that RM is superior to other fixed-priority assignments in the sense that no other fixed-priority algorithm can schedule a task set that cannot be scheduled by RM.

Now consider the set of real-time transmission tasks \(\mathcal{F}_i, 1 \leq i \leq N\) defined in (15). These tasks are periodic, independent, non-pre-emptive and the period of each task equals its deadline. These characteristics are just what the operation of the RM algorithm needs. Therefore the RM scheduling algorithm can be applied to schedule the set of transmission tasks in NPCSSs, by which the transmission with shorter deadlines (or periods) are assigned higher priorities. Thus, the predictive control sequence can be transmitted first if the network is idle, that is

\[
\text{if } h_i < h_j, \text{ then } Y_i > Y_j, \quad i, j = 1, 2, \ldots, N
\]

where \(Y_i\) represents the priority of the transmission task of system \(S_i\).

Theorem 2: A set of \(N\) NPCSSs sharing the network resource in their forward channel (indexed by the increasing order of their transmission periods, that is, \(h_i \leq h_{i+1}, i = 1, 2, \ldots, N - 1\)) are stable schedulable if for all \(i = 1, \ldots, N\)

\[
\mathcal{U}(i) \leq f(i)
\]

where \(f(i) = i(2^{1/i} - 1)\) and

\[
\mathcal{U}(i) = \begin{cases} 
\frac{i - 1}{h_i} + \frac{1}{h_i} & i = 1, 2, \ldots, N - 1 \\
\frac{1}{h_i} & i = N
\end{cases}
\]
Proof: From [27, Theorem 16], a set of non-pre-emptive periodic real-time tasks are schedulable if
\[
e_{i_1} + e_{i_2} + \cdots + e_{i_j} + \frac{\bar{b}_{i_j}}{h_i} \leq (2^{1/j} - 1) \tag{18}
\]
where \(e_i\) is the frame time, \(h_i\) the transmission period, each for the \(i\)th task, and \(\bar{b}_{i_j}\) is task \(i\)'s worst-case blocking time by the lower priority tasks, that is
\[
\bar{b}_{i_j} = \max_{j = i+1, \ldots, N} e_j \tag{19}
\]
As has been pointed out above, in the proposed networked predictive control systems, (12) holds, and therefore \(\bar{b}_{i_N} = 0\) from (19). Hence, the theorem holds.

Corollary 1: If (1) \(h_{i+1} \leq 2h_i, i = 1, 2, \ldots, N-2\) and (2) \(h_{N}/(h_{N-1}) \leq ((N-1)/N) \left(2^{1/(N-1)} - 1\right)/(2^{1/N} - 1)\), then the set of tasks of NPCSs is stable schedulable if
\[
e \sum_{i=1}^{N} \frac{1}{h_i} \leq N\left(2^{1/N} - 1\right) \tag{20}
\]
Proof: From assumption (1), we obtain for \(i = 1, 2, \ldots, N-2\)
\[
\mathcal{U}(i+1) - \mathcal{U}(i) = e\left(\frac{2}{h_{i+1}} - \frac{1}{h_i}\right) \geq 0
\]
and it is apparent that \(\mathcal{U}(N) \geq \mathcal{U}(N-1)\) from assumption (2). Thus
\[
\mathcal{U}(N) = \max_{1 \leq i \leq N} \mathcal{U}(i)
\]
On the other hand, it is easy to show that the function \(f(t)\) is non-increasing, and therefore \(\mathcal{U}(N) \leq f(N)\) implies \(\mathcal{U}(i) \leq f(i), i = 1, 2, \ldots, N-1\), which completes the proof recalling Theorem 2.

Corollary 2: If the transmission periods of all the systems are the same, that is, \(h_i = h, i = 1, 2, \ldots, N\), then the collection of systems are stable schedulable if
\[
e \leq 2^{1/N} - 1 \tag{21}
\]
Proof: It can be obtained directly from Corollary 1. \(\square\)

4.2 Dynamic feedback scheduling

In the static RM scheduling scheme presented above, the transmission periods \(h_i\) for all the systems are assigned a priori to ensure the stability of the systems and do not change in the process of their operation. In the case of the network being shared only by systems \((S_i), 1 \leq i \leq N\), this method works though the performance of the systems may not be optimum because the network is not fully used. However, if the network is not private to these systems, that is, there are other components occupying the network, it cannot be assumed that the propagation delay is constant due to the change of the network loads. On the basis of this reality, a DFS scheme is designed. In this scheme, a higher-level feedback scheduler is proposed, which obtains the information of the network utilisation from the network and the control performances from all the systems as well, and regularly calculates and reassigns the transmission period for each system. During the interval of two successive reassignments of periods, the RM algorithm still works. The framework of the DFS scheme is depicted in Fig. 3.

4.2.1 Period of DFS: This period, noted by \(T_{DFS}\), needs to be chosen carefully. Generally, its value depends on the speed at which the condition of the network changes. A small \(T_{DFS}\) is needed if the network condition changes rapidly, whereas a larger one can still guarantee the performance of the system without overloading the network if the parameters of the network do not change much over a long time. However, in any case, \(T_{DFS}\) should be always not less than the transmission periods of all the systems, that is, \(T_{DFS} \geq \max_{i=1}^{N} h_i\).

4.2.2 Measurement of the network utilisation: To obtain the utilisation information of the network, a packet containing this information is sent to the feedback scheduler using the period of \(T_{DFS}\). This information is mainly reflected by the propagation time delay in the forward channel of each system. The propagation time will increase to a certain extent with the increase of the network load. Another factor affecting the stability of each system is the change of the delay in the backward channel. In order to take this factor into account and for simplicity, we assume that the upper bound of the delay in the backward channel during the \(k\)th period of DFS (noted by \(\tau_{ca}^0(k)\) for system \(S_1\)) can be obtained from the network and the network does not change too much during this period, thus enabling us to use \(\tau_{ca}(k)\) to estimate its value during the \((k+1)\)th period. Then the deadline \(h_i\) of each task for each system will be recalculated by updating the propagation time \(\tau_{ca}^0(k)\) and the upper bound of the delay in the backward channel every \(T_{DFS}\) seconds as follows
\[
D_i(k + 1) = \hat{D}_i - \mu(\tau_{ca}^0(k) + \tau_{ca}(k)) - e \tag{22}
\]
where \(\mu\) close to 1 is a smoothing factor satisfying
\[
\mu(\tau_{ca}^0(k) + \tau_{ca}(k)) \geq \tau_{ca}^0(k + 1) + \tau_{ca}(k + 1), \quad \forall k
\]
4.2.3 Reseasignment of the transmission periods of all the systems: In order to obtain the control performance of each system, an obvious idea is to use the predictive quality of performance (QoP) during the next DFS period. This QoP during the \(k\)th period of DFS can be defined for
system $S_i$ as

$$
\hat{P}_i(k) = \sum_{j=[kT_{DFS}/h_i]}^{[k+1]T_{DFS}/h_i} (\hat{y}_i(j) - \tau_{w,j})^2
$$

(23)

where $[x]$ is the nearest integer to $x$ satisfying $[x] \geq x$.

The calculation of the new transmission periods for all the systems can then be modelled as an optimisation problem $\mathcal{P}$

$$
\mathcal{P}_i:\begin{align*}
\text{Select } h_i, & \quad i = 1, 2, \ldots, N \\
\text{s.t. } & \quad \min \sum_{i=1}^{N} \hat{P}_i, \\
\text{subject to } & \quad \mathcal{H}(i) \leq f(i), \quad i = 1, 2, \ldots, N \\
& \quad \bar{h}_i \geq T_i, \quad i = 1, 2, \ldots, N
\end{align*}
$$

where $\mathcal{H}(i)$ and $f(i)$ are defined in Theorem 2 and $T_i$ is the sampling period for system $S_i$ as defined in Section 2.

In practice, the predictive outputs $\hat{y}_i(j) - \tau_{w,j}$ of the systems can be obtained using the open-loop prediction as shown in (2), whereas the online operation of the optimisation problem $\mathcal{P}$ is not a simple one. Therefore not the predictive QoP but the previous QoP, $\bar{J}_i(k)$, for system $S_i$ and $k$ for the $k$th period of DFS, is used to represent the performance of the system, which is defined as follows and can be easily obtained

$$
\bar{J}_i(k) = \sum_{J(i) \in \Pi_i} J_i(k; N_i^1, N_i^2, N_i^3)
$$

(24)

where $J_i(k; N_i^1, N_i^2, N_i^3)$ is the objective function of system $S_i$ in the predictive controller defined in (1) and $\Pi_i$ is the set of objective functions during the $k$th period of DFS, or from the $[(k-1)T_{DFS}/h_i]$th transmission period of system $S_i$ to the $[kT_{DFS}/h_i]$th.

Let the new transmission periods chosen in this way be

$$
\frac{1}{h_i(k+1)} = \kappa(k+1) = \frac{\bar{J}_i(k)}{\sum_{i=1}^{N} \bar{J}_i(k)} = \kappa(k+1)\theta_i(k)
$$

(25)

where $\kappa$ is a proportion factor and can be chosen as follows to include the constraints of stable schedulability of Theorem 2

$$
\kappa(k+1) = \max_{i=1,\ldots,N-1} \left\{ \frac{f(i)}{e(\sum_{i=1}^{N} \theta_i(k) + \theta_i(k))}, \frac{f(N_i)}{e} \right\}
$$

(26)

Considering the fact that the network load may change greatly between two periods of DFS, a smoothing factor $\rho(0 < \rho \leq 1)$is introduced to avoid network overload. Also taking account of the fact that the transmission period $h_i$ can never exceed $D_i$ for the stability of the system, then the transmission periods are obtained as

$$
h_i(k+1) = \min \left\{ \frac{\rho}{\kappa(k+1)\theta_i(k)}, D_i(k+1) \right\}
$$

(27)

4.2.4 Algorithm of DFS scheme: The algorithm of DFS can then be described as follows.

S1. Initialise $T_{DFS}, k = 1, t = 0, h_i = \bar{D}_i - e, \quad 1 \leq i \leq N$.

S2. During time $t \in [(k-1)T_{DFS}/h_i, kT_{DFS}]$, do the following: $S2a$. Generate the predictive control sequences $U_i^*(\cdot)$, $1 \leq i \leq N$ using (6) for all the systems.

S2b. Apply RM algorithm described in Section 4.1 to determine the order of the transmission tasks of the NPCSs.

S2c. Transmit the predictive control sequence as $S2b$ determines.

S3. When $t = kT_{DFS}$, the DFS module checks the utilisation information of network. If the network is in full use, set $k = k + 1$, return to $S2$; else go to $S4$.

S4. The DFS module calculates the new transmission periods using (25) and reassign the priorities for all the systems; set $k = k + 1$, return to $S2$.

4.2.5 Stability of DFS

Theorem 3: A set of NPCSs is stable under DFS if the transmission tasks of the NPCSs are always stable schedulable.

Proof: The application of DFS to the set of NPCSs in this paper does not change the delay in the backward channel, since the DFS module is on the controller side, whereas it does change the delay in the forward channel by reassigning the transmission period $h_i, i = 1, 2, \ldots, N$ to the systems. However, by taking account of (22) and (27), we obtain

$$
\begin{align*}
& h_i(k+1) \leq D_i(k+1) \leq \bar{D}_i - \tau_0(k+1) - \tau_e(k+1) - e, \\
& \forall k, i = 1, 2, \ldots, N
\end{align*}
$$

(28)

which implies

$$
\sup_{k} \left\{ h_i(k) + \tau_0(k) + \tau_e(k) + e \right\} \leq \bar{D}_i, \quad i = 1, 2, \ldots, N
$$

(29)

Note that the left-hand side of (29) is the effective maximum of RTT for system $S_i$, which is always no more than the required SSRTT. Thus, the theorem is valid recalling Theorem 1. $\square$

5 Simulation

An example of the co-design method proposed above by using simulation is given in this section.

5.1 Simulation parameters

Three second-order linear systems are considered in the simulation, with the expressions in continuous form

$$
\begin{align*}
A_1^e & = \begin{pmatrix} -11.1572 & -106.0132 \\ -110.3637 & -5.2680 \end{pmatrix}, \\
A_2^e & = \begin{pmatrix} -23.7783 & -48.9313 \\ -107.2959 & -34.0550 \end{pmatrix}, \\
A_3^e & = \begin{pmatrix} -91.6291 & -160.9438 \\ -69.3147 & -35.6675 \end{pmatrix}, \\
B_1^e & = \begin{pmatrix} -0.1295 \\ 2.6890 \end{pmatrix}, \quad B_2^e = \begin{pmatrix} 4.3801 \\ 2.3278 \end{pmatrix}, \\
B_3^e & = \begin{pmatrix} 9.1471 \\ 4.0444 \end{pmatrix}, \quad C_1^e = C_2^e = C_3^e = 1
\end{align*}
$$

and sampling periods $T_1 = 0.02 \ s, \ T_2 = 0.015 \ s, \ T_3 = 0.01 \ s$, respectively. The corresponding discrete
and systems that the set point
For the simplicity of simulation, assume for all the three
assumptions, the closed-loop system can be obtained
so that the state observer (3) is not required. Under these
Fig. 4 State evolution using RM algorithm
Only the first state is illustrated

Table 1: Simulation parameters

<table>
<thead>
<tr>
<th></th>
<th>System 1</th>
<th>System 2</th>
<th>System 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>$T$</td>
<td>0.02</td>
<td>0.015</td>
<td>0.01</td>
</tr>
<tr>
<td>$\phi$</td>
<td>0</td>
<td>0.05</td>
<td>0.06</td>
</tr>
<tr>
<td>$x_0$</td>
<td>$[-1 -1]^T$</td>
<td>$[-1 -1]^T$</td>
<td>$[-1 -1]^T$</td>
</tr>
<tr>
<td>$P_1$</td>
<td>$[1 \ 30 \ 20 \ 0 \ / \ /]$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$P_2$</td>
<td>[0.008 0.002 0.01 10]</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

$T$ is the sampling period
$\phi$ is the first release time
$x_0 = [x_{01} \ x_{02}]^T$ is the initial state
$P_1 = [N_1 \ N_2 \ N_3 \ \omega \ W_1 \ W_2]$ is the predictive parameters
$P_2 = [\sigma^2 \ \tau_{sc} \ T_{sim}]$, $T_{sim}$ is the simulation time

For the simplicity of simulation, assume for all the three
systems that the set point $\omega = 0$, weighting factors $W_1 = 1$
and $W_2 = I$ and the state vector can be obtained directly
so that the state observer (3) is not required. Under these
assumptions, the closed-loop system can be obtained from (9).

Other parameters of the simulation are shown in Table 1.
A Gaussian white noise with standard deviation 0.1 is also
introduced as the disturbance of the state.

Table 2: SSRTT and transmission periods

<table>
<thead>
<tr>
<th></th>
<th>System 1</th>
<th>System 2</th>
<th>System 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\hat{D}_i$</td>
<td>5</td>
<td>7</td>
<td>8</td>
</tr>
<tr>
<td>$h_i$</td>
<td>4</td>
<td>5</td>
<td>6</td>
</tr>
</tbody>
</table>

systems can be obtained as

\[
A_1 = \begin{bmatrix}
0.8 & 0.12 \\
0.11 & 0.9
\end{bmatrix},
A_2 = \begin{bmatrix}
0.7 & 0.48 \\
0.2 & 0.6
\end{bmatrix},
A_3 = \begin{bmatrix}
0.4 & 0.2 \\
0.5 & 0.7
\end{bmatrix},
B_1 = \begin{bmatrix}
0.02 \\
0.05
\end{bmatrix},
B_2 = \begin{bmatrix}
0.08 \\
0.06
\end{bmatrix},
B_3 = \begin{bmatrix}
0.08 \\
0.1
\end{bmatrix},
C_1 = C_2 = C_3 = 1.
\]

Note that the execution time of each job is $e = 0.008$ s,
then the value of the utilization function $U_i$ can be
obtained as follows, $U_i(i) = 0.2, 0.3133, 0.34, i = 1, 2, 3$
whereas $f(i) = 1, 0.8284, 0.7798, i = 1, 2, 3$, thus (17)
holds, and by Theorem 2, the set of NCSs is stable schedu-
able under RM.

The state evolution of the first state of the three systems
under RM is illustrated in Fig. 4.

5.2 Simulation of RM algorithm
Using the LMI toolbox in Matlab, the SSRTT of each
system $\hat{D}_i$ can be obtained by Theorem 1, thus enabling
the transmission periods $h_i$ to be calculated according to
(13) and (14), as shown in Table 2.

5.3 Simulation of DFS algorithm
It is noted that the SSRTT obtained in Theorem 2 is con-
servative. In the simulation of DFS, the deadlines of the
three systems are set to be 8, 10 and 12 steps, respectively,
and the propagation delays of the systems in the forward
channel are set to be randomly changing under the con-
straint that the real RTT are no more than the new
SSRTT, in order to simulate the changes of the network
loads. All the other parameters remain the same as in RM
algorithm.

The simulation result (Fig. 5) shows that the systems are
still stable under this larger SSRTT and with fluctuating
propagation delays.

6 Conclusion
In this paper, a co-design approach is proposed to deal with
the communication constraints for a set of NCSs which are
connected to a shared network. To reduce the negative
effect of the network-induced delay and data packet
dropout, predictive control theory is applied to produce
future control inputs to the systems, whereas the schedul-
ing theorem, both the static algorithm RM and a dynamic
DFS which takes advantage of the feedback information of
the system performances, is applied to schedule the trans-
missions of the predictive control sequences of all the
systems. Simulation results illustrate the validity of the
integration of both predictive control and scheduling
theories.
7 References


