Integrated Communication and Control Systems: Part II—Design Considerations

Asok Ray
Mem. ASME.

Yoram Halevi
Mechanical Engineering Department,
The Pennsylvania State University,
University Park, PA 16802

1 Introduction

Integrated Communication and Control Systems (ICCS) essentially belong to a class of distributed digital control for application to complex dynamical processes like advanced aircraft, spacecraft, and autonomous manufacturing plants. System components exchange information via an asynchronous time-division multiplexed network which potentially enhances resource utilization and reduces space, power and wiring requirements of the integrated system. However, multiplexed networks are also a source of time-varying and possibly stochastic delays in the feedback control loop. The objective of this on-going research is to develop a comprehensive methodology for analysis and design of ICCS. This necessitates interactions between the disciplines of communication systems and control systems engineering.

In our earlier work [2, 3], we presented the performance analysis of ICCS networks and demonstrated, by combined discrete-event and continuous-time simulation, how the network-induced delays could degrade the control system performance. In Part I [1], which is a companion paper, we have developed a discrete-time, finite-dimensional, time-varying model of the closed loop ICCS. A necessary and sufficient condition for the stability of delayed control systems has been established.

This paper is the second of two parts, and addresses the ICCS design issues for nonperiodic and stochastic delays and outlines the framework of alternative design procedures. Using the model developed in Part I, the impact of network-induced delays on the system stability was investigated, and their physical significance has been exemplified by simulation. The flight control system of an advanced aircraft [3] has been simulated to assess the detrimental effects of a combination of time-varying data latency and mis-synchronization between system components.

The paper is organized in five sections and two appendices. The symbols that were defined in Part I and in this paper are listed in the nomenclature. Section 2 deals with the usage of the model developed in Part I and other criteria for ICCS design. Section 3 presents the simulation results using (1) a simple model to elucidate the intrinsic characteristics of time-varying delayed control systems and (2) a flight control system model of an advanced aircraft to illustrate the detrimental effects of vacant sampling and message rejection at the controller. Alternative analytic procedures for ICCS design are outlined in Section 4. The summary and conclusions of this paper are presented in Section 5 along with recommendations for future research. Appendix A contains a supporting proposition. Appendix B provides a brief description of the network testbed where the experimental evaluation of the ICCS design techniques are planned for this on-going research project.

2 Usage of the Delayed System Model for ICCS Design

In this section we first address the key features of the discrete-time, finite-dimensional, time-varying model of the delayed control system, which was developed in Part I [1]. Then we specify the criteria for avoiding the vacant sampling
and message rejection phenomena from the perspectives of ICCS design.

The augmented state vector \( \mathbf{X} \) of the delayed system [1] includes past values of the plant input \( u \) and output \( y \). The amount of the information that must be included in the augmented state vector may change at individual sampling instants because of the time-varying nature of the network-induced delays. This is explained next with reference to the delayed control system model developed in Part I. (The symbols are defined in the nomenclature as well as in Part I.)

The maximum value of the sensor-controller delays \( \Delta_s \) is \( \Delta_s + jT \) if \( \Delta_{max} = \left( (j-1)T, (j+1)T \right) \) and that of the controller-actuator delay \( \Delta_a \) is \( \Delta_a + jT \). Whether these two maxima would occur on the same signal (as it traverses around the control loop) depends on several factors such as the traffic distribution in the network and the relative location of the sensor and controller terminals. In view of the controller-actuator delay, the control input \( u_{k-j} \) generated at the \( (k-j) \)th controller sample, may arrive at the actuator during the sensor sampling interval \( [kT, (k+1)T) \) if \( \Delta_s + \delta_s + \Delta_{max}(\delta_s \epsilon [jT, (j+1)T), \) and the oldest input that may drive the actuator during the initial part of this interval is \( u_{k-j-1} \). In view of the ICCS design requirements, we consider a special case where each of \( \Delta_s \), \( \delta_s \), and \( \Delta_{max} \) is less than \( T \).

(a) Since \( \Delta_s + \delta_s + \Delta_{max} < 3T \) is considered, the first input arriving at the actuator during the interval \( [kT, (k+1)T) \) cannot be older than \( u_{k-1} \). Therefore, \( u(t) \) will assume one or more of the four different values, \( u_{k-3} \), \( u_{k-2} \), \( u_{k-1} \), and \( u_k \) during this interval. Similarly, as the maximum of the sensor-controller delay \( \Delta_s \) is \( \Delta_s + jT \) for \( \Delta_{max} < T \), the oldest data that may be used by the controller in computing \( u_k \) is \( y_{k-1} \).

(b) In terms of the derivations of Section 4 of Part I, if \( \Delta_s + \delta_s + \Delta_{max} < 3T \), \( i = 1, 2, 3 \), then the maximum number of actuator commands in one sensor sampling period, \( \leq i \).

(c) If \( \Delta_s + \delta_s + \Delta_{min} > jT, j = 1, 2 \), then the input matrices \( B_{m} = 0, \forall m \leq (j-1) \) and \( \forall k \). (See Section 4 in Part I.) In this case, the latest control input that could arrive at the actuator in \( [kT, (k+1)T) \) is \( u_{k-2} \) and the input matrices for any subsequent inputs must be zero.

(d) Since sensor to controller data latency is assumed to be always less than \( T \), the sensor data \( y_k \) can be used as the latest plant output for generating the inputs \( u_k \) or \( u_{k+1} \) but not for \( u_{k+j}, j \geq 2 \). Therefore, \( p(k) = 0 \) or \( p(k) = 1 \).

(e) If \( \Delta_s > \Delta_{max} \), then \( p(k) = 0 \forall k \). In this case \( y_{k} \) always arrives at the controller before the controller sampling instant \( k \).

(f) If \( \Delta_s \leq \Delta_{min} \), then \( p(k) = 1 \forall k \). In this case \( y_{k} \) always arrives at the controller after the controller sampling instant \( k \).

These observations show that the augmented state vector \( \mathbf{X} \) for the general case under observations \( (a) \) to \( (f) \), is given as

\[
\mathbf{X}_k = [\mathbf{x}_k \mathbf{y}_{k-1} T \mathbf{y}_{k-2} T \mathbf{y}_{k-3} T]^T \tag{2.1}
\]

The above equation indicates the maximum size of \( \mathbf{X} \) under the assumption that \( (\Delta_s + \delta_s + \Delta_{max}) < 3T \). \( \mathbf{X} \) can be reduced in size by eliminating some of the past inputs and outputs under specific conditions. A few examples follow.

If \( i = 2 \) in observation \( (b) \), then the term \( u_{k-3} \) is deleted from (2.1); if \( i = 1 \), then both \( u_{k-3} \) and \( u_{k-2} \) are deleted. If the observation \( (e) \) holds, then \( y_{k-2} \) should be deleted from (2.1).

Observations \( (d) \) and \( (f) \) may not specifically reduce the order of the augmented system but some of the time-varying terms are reduced to constants.

Observations \( (a) \) to \( (f) \) identify different cases based on \( \Delta_{min} \) and \( \delta_{max} \). The sequence \( (\mathbf{y}_k) \), by which the sensor samples are delayed before being processed at the controller, gives rise to four possible cases as described below.

Case #1. Constant \( \Theta_s \): Smaller Delay. \( p(k) = p(k+1) = 0 \) implies that the sensor data \( y_k \) and \( y_{k+1} \) reach the controller's receiver before their respective sampling instants. In this case, the sensor data at both instants are subject to a delay of \( \Delta_s \) before being processed by the controller.

Case #2: Constant \( \Theta_a \): Larger Delay. \( p(k) = p(k+1) = 1 \) implies that the sensor data \( y_k \) and \( y_{k+1} \) reach the controller's receiver after their respective sampling instant at the controller. In this case, the sensor data at both instants are subject

---

Nomenclature for Symbols Used in Parts I and II

- **A**: plant system matrix \((n \times n)\)
- **\( A_0 \)**: plant state transition matrix \((n \times n)\)
- **B**: plant input matrix \((n \times m)\)
- **C**: plant output matrix \((r \times n)\)
- **F**: controller system matrix \((q \times q)\)
- **\( F(*) \)**: probability distribution function
- **\( f(*) \)**: probability density function
- **G**: controller input matrix \((q \times r)\)
- **H**: controller output matrix \((m \times q)\)
- **J**: controller direct coupling matrix \((m \times r)\)
- **J**: performance index
- **K**: controller gain
- **\( \mu_{re} \)**: probability of message rejection at the controller's receiver
- **\( \nu_{re} \)**: probability of vacant sampling at the controller's receiver

\( \Delta_s \): sensor-to-controller data latency
\( \delta_s \): processing delay at the controller computer
\( \delta_{sc} \): sensor-to-controller data latency
\( \delta_{ca} \): controller-to-actuator data latency
\( \eta \): controller state vector \((q \times 1)\)
\( \theta_s \): average delay in the control loop
\( \Theta_s \): sensor-controller delay
\( \Theta_{ca} \): controller-actuator delay
\( \lambda \): lumped delay
\( p \): maximum # of delayed samples before processing of sensor data
\( l \): maximum # of actuator commands in one sensor sampling period
\( \Phi \): augmented system matrix \((N \times N)\)
problems in ICCS design is to determine the optimal value of the constant parameter $\delta$. We discuss this issue in the following paragraph.

The sensor-controller delay $\delta_s$ remains a constant at $\Delta_s$ if $\Delta_s > \delta_{\text{max}}$ (Case #1) and at $T + \Delta_s$ if $\Delta_s < \delta_{\text{min}}$ (Case #2). If the network is lightly loaded, i.e., $\delta_{\text{max}} < T$, a sufficiently small $\Delta_s > \delta_{\text{max}}$ would perhaps satisfy the system stability requirements. However, if the network traffic is moderate to heavy, i.e., $\delta_{\text{max}} = T$, or if, as a result of multiple sampling rates, the traffic is light on the average but $\delta_{\text{max}}$ is not small, then the resulting delay $\Delta_s > \delta_{\text{max}}$ may not be acceptable for a satisfactory system design performance. In that case $\Delta_s$ should be selected to be smaller than $\delta_{\text{max}}$, which would cause vacant sampling and message rejection at the controller and $\delta_s$ will be either $T + \Delta_s$ or $\Delta_s$ depending on whether the sensor-to-controller data latency exceeds $\Delta_s$ or not. Apparently, the system dynamic performance depends on the average value of $\delta_s$ as well as on the frequency of vacant sampling and message rejection. This is discussed further in Section 4.

So far we have assumed that $\Delta_s$ is maintained at a constant desired value by periodic synchronization of the system components. This could be achieved by transmitting high-priority synchronization signals via the network medium or by additional wiring, and would require additional efforts to meet the system reliability requirements. An alternative approach to synchronization is to deliberately assign dissimilar sampling periods to the sensor and the controller. If the sensor sampling period is $T_s$ and that of the controller is $T_c$, such that $T_s = T_c(1 - \epsilon)$ for $1 < \epsilon < 1$, then $\Delta_s$ completes a cycle at every $1/\epsilon - 1$ samples of the controller. By choosing $\epsilon$ sufficiently large (which should still satisfy $1 < \epsilon < 1$), $\Delta_s$ could be prevented from remaining in the vicinity of the worst value for a long time. In other words, the time averaging will reduce the detrimental effects of the worst case. By increasing $\epsilon$ either vacant sampling or message rejection (but not both) at the controller can be eliminated. This fact is supported by the following proposition.

**Proposition 2.1:** The controller receives at least one sensor data during each of its sampling interval provided that

$$\epsilon > (\delta_{\text{max}} - \delta_{\text{min}})/T_c \quad \text{where} \quad \epsilon = (T_s - T_c)/T_c$$

**Proof:** Let $\delta^k$ be the data latency of the $k$th message from the sensor to controller. The $k$th sensor data generated at $t_0 + kT_s$ is received at the controller at $t_0 + kT_s + \delta^k$. The next sensor data are received by the controller at $t_0 + (k + 1)T_s + \delta^{k+1}$. Therefore, the interval between the reception of two consecutive sensor data, $\Delta T = \delta^k + \delta^{k+1}$, and $\sup \Delta T = T_s + \delta_{\text{max}} - \delta_{\text{min}}$, $\sup \Delta T$ being less than or equal to $T_s$, suffices that the controller always receives at least one sensor data during each of its sampling period.

**Corollary to Proposition 2.1:** The controller receives at most one sensor data during each of its sampling interval provided that

$$- \epsilon > (\delta_{\text{max}} - \delta_{\text{min}})/T_s$$

**Proof:** Let $\inf \Delta T = T_s - \delta_{\text{max}} + \delta_{\text{min}}$. $\inf \Delta T$ being greater than or equal to $T_s$, suffices that controller always receives at most one sensor data during each of its sampling period.

**Remark 2.1:** Although no vacant sampling occurs under the condition of Proposition 2.1, some of the sensor messages are rejected at the receiver buffer of controller. Similarly, no sensor message is rejected under the condition of the corollary to Proposition 2.1 but some of the controller sampling periods remain vacant.

**Remark 2.2:** The average rate of message rejection is equal to that of vacant sampling only if $T_s = T_c$. 

---

*Transactions of the ASME*
3 Simulation of the Delayed Control System

As a first step to ICCS design, a delayed control system was simulated. To elucidate the intrinsic characteristics of the discrete-time, finite-dimensional model, developed in Part I [1], we selected a simple example where the plant model is given as

$$\frac{dx}{dt} = -x + u, \quad y = x \tag{3.1}$$

where \(x, u, y\) are scalar functions of time. The control law is chosen to be purely proportional as given by

$$u_k = K(x_k - z_k) \tag{2.2}$$

where \(r_k\) is the reference signal, and \(z_k = y_k - p(k)\) is the delayed sensor data at the controller whose proportional gain is the scalar \(K\). On the basis of the above two equations, the pertinent matrices in (4.9) of Part I reduce to scalars and are set as: \(A = -1\), \(B = C = 1\), \(H = 0\), \(J = K\), and \(E_0 = 1 - p(k) = E_1 = p(k)\).

For \(r_k = 0\) and the maximum data latency \(\delta_{\text{max}}\) being less than \(T\), the finite-dimensional, discrete-time model of the delayed control system is

\[
\begin{bmatrix}
    x_{k+1} \\
    y_k \\
    u_k \\
    u_{k-1}
\end{bmatrix} =
\begin{bmatrix}
   A - B_0 k (1 - p(k)) & -B_0 k p(k) & B_1 k & B_2 k \\
   1 & 0 & 0 & 0 \\
   -K (1 - p(k)) & -K p(k) & 0 & 0 \\
   0 & 0 & 1 & 0
\end{bmatrix}
\begin{bmatrix}
    x_k \\
    y_{k-1} \\
    u_{k-1} \\
    u_{k-2}
\end{bmatrix}
\tag{3.3}
\]

where \(A = \exp(-T)\), \(B_0 k = 1 - \exp(t_k - T), B_1 k = \exp(t_k - T) - \exp(t_k - T), B_2 k = \exp(t_k - T) - \exp(-T)\). The control inputs could arrive at the actuator at the instants \(kT + t_k\) and \(kT + t_k\) during the \(k\)th sensor sampling interval \([kT, (k + 1)T]\) with the constraint \(0 \leq t_k \leq t_k \leq T\). This implies that if \(u_k\) arrives at the actuator after the \(k\)th sensor sampling instant, then \(t_k = T\) and \(0 \leq t_k \leq T\). Similarly, if \(u_{k-1}\) arrives before the \(k\)th sensor sampling instant, then \(t_k = 0\) and \(0 \leq t_k \leq T\).

The significance of the state equation (3.3) under different scenarios is explained below.

No delays: \(p(k) = 0\) and \(t_k = t_k = 0\) implying that \(B_0 k = 1 - \exp(-T), B_1 k = B_2 k = 0\). Thus, \(x_{k+1} = \exp(-T) x_k\). This is equivalent to a conventional digital control with a nondelayed feedback.

Delayed Control Signal: \(p(k) = 0\) or \(1\), \(t_k = T\) and \(0 \leq t_k \leq T\). This implies that \(B_0 k = 0\) and possibly \(B_1 k \neq 0\) and \(B_2 k = 0\). Therefore, \(x_{k+1} = \exp(-T) x_k\). This is equivalent to a digital control system where the actuator command is generated using a weighted average of delayed control inputs.

Delayed Control Signal: \(p(k) = 0\) or \(1\), \(t_k = T\) and \(0 \leq t_k \leq T\). This implies that \(B_0 k = 0\) and possibly \(B_1 k \neq 0\) and \(B_2 k \neq 0\). Therefore, \(x_{k+1} = \exp(-T) x_k + B_1 k u_{k-1} + B_2 k u_{k-2}\). This is equivalent to a digital control system where the actuator command is generated using a weighted average of delayed control inputs.

Now we present the simulation results of the delayed control system using the following numerical values.

The sampling time and the control signal processing time were set at \(T = 0.1\) s and \(\phi = 0.015\) s, respectively. For the purpose of illustrating the effects of periodic delays, the data latencies \(\delta_{\text{min}}\) and \(\delta_{\text{max}}\) (in seconds) were chosen to follow sinusoidal profiles with \(\omega = 13\) rad/s.

\[
\delta_{\text{min}} = 0.015 + 0.005 \sin(\omega T) \tag{3.4a}
\]

\[
\delta_{\text{max}} = 0.015 + 0.005 \sin(\omega T + \phi) \tag{3.4b}
\]

where \(t\) is the time in s, and \(\phi\) is the phase differences in rad. However, since \(\phi \in (-\pi, \pi)\) depends on several factors including the traffic and relative location of the controller and sensor terminals, we selected \(\phi = 0\). In that case, the continuous-time functions in (3.4a) and (3.4b) reduce to discrete sequences by sampling at the sensor and controller sampling instants \(JT\) and \(JT + \Delta_s\), respectively.

\[
\delta_{\text{min}} = 0.015 + 0.005 \sin(\omega JT) \tag{3.5a}
\]

\[
\delta_{\text{max}} = 0.015 + 0.005 \sin(\omega JT + \Delta_s) \tag{3.5b}
\]

We note, from (3.5a) and (3.5b), that \(\delta_{\text{min}} = 0.010\) s, \(\delta_{\text{max}} = 0.020\) s. The data latencies \(\delta_{\text{max}}\) and \(\delta_{\text{max}}\) cannot be treated as perfectly periodic for the given values of \(\omega\) and \(T\) since there is no finite integer \(M\) such that \(\delta_{\text{max}} + M = \delta_{\text{max}} + \delta_{\text{max}}\). Therefore, the analytical technique, reported in Part I [1] for periodic delays, cannot be readily applied for evaluating the system stability in this case.

A series of simulation runs were conducted to determine the system stability for different combinations of \(\Delta_s\) and \(K\). Figure 2 shows the stability region of the delayed control system as a function of the feedback gain \(K\) and the time skew \(\Delta_s\). If \(\Delta_s \leq \delta_{\text{min}}\), then the sensor-controller delay \(\delta_{\text{max}}\) on the other hand, if \(\Delta_s > \delta_{\text{max}}\), then \(\delta_{\text{max}} = \Delta_s\). Whereas \(\Delta_s < \delta_{\text{min}}\) would result in large delays, values of \(\Delta_s\) in excess of \(\delta_{\text{max}}\) may not be desirable as well. In the simulation, we set \(\Delta_s\) close to \(\delta_{\text{max}}\) in the range of 0.019 to 0.020 where stability was maintained for \(K = 22.5\) with negligible message rejections. Figure 2 shows that values of \(\Delta_s < \delta_{\text{min}} = 0.010\) may result in loss of stability even for low gains \((K > 9.5)\). If \(\Delta_s\) cannot be maintained at the desired value, \(K\) should be reduced below 9.5 to ensure stability for all \(\Delta_s\).

The above problem was partially circumvented by setting the sensor sampling period \(T_s\), smaller than the controller sampling period \(T_c\). To show that the improved stability is caused by the fact that \(T_s < T_c\) and not by a smaller sensor sampling period, \(T_s\) was kept constant at the previous value of 0.1 s while \(T_c\) was increased. Therefore the relationship \(T_s = (1 + \epsilon) T_c, \epsilon \in [0, 1]\) was used instead of \(T_s = (1 - \epsilon) T_c\). The resulting observations are summarized below.

The rate of sensor message rejection monotonically increases with \(\epsilon\). If \(T_s = T_c\), the time skew \(\Delta_s\) oscillates between 0

![Fig. 2 Stability region](https://example.com/stability_region.png)
and $T$, and the rate of change of $\Delta$ directly depends on the magnitude of $\epsilon$.

Apparently even small positive values of $\epsilon$ could improve stability and, therefore, allow higher gains to achieve a better system dynamic performance. For example, the simulated system was found to be stable at $K = 11$ and $\epsilon = 0.002$. The simulation results indicate that, for $K \leq 11.5$, the system exhibits stability with $\epsilon$ in the range of 0.003 to 0.5. (Values of $\epsilon$ in excess of 0.5 were not simulated.) System stability for $K$ in the vicinity of 11.5 is sensitive to small values of $\epsilon$ but this sensitivity significantly decreases for larger values of $\epsilon$. At higher $K$, small ranges of $\epsilon$ for which the system is stable may exist but it is apparently difficult to provide physical interpretations of these stability regions from the simulation results only. An analytical method is required for evaluating the bounds of $\epsilon$ as a function of the controller parameters, which is one of the areas of the current research.

Better stability with $\epsilon > 0$ results from the reduction of vacant sampling. Loosely speaking, if the system is stable for a wide range of $\Delta$, a small decrement in $T$ could lead to a stable system; if this range of $\Delta$ is narrow, changes in $T$ have a smaller bearing on the system stability.

We conclude this example by noting that the stability analysis cannot be made solely on the basis of average values of the time-varying delays. This is exemplified below for the case of identical sensor and controller sampling periods.

Let the arrival time (relative to the sensor sampling instant) of control inputs $[u_k]$ at the actuator be $t = T/2 + kT$, and, on the average, one out of three sensor messages arrives at the controller after the respective sampling instant. Assuming periodic delays with a period of $6T$, the three different sequences of $p(k)$ (where $z_k = y_k - p(k)$) along with the stable ranges of the controller gain $K$ are listed below.

<table>
<thead>
<tr>
<th>Number</th>
<th>Sequence of $p(k)$</th>
<th>Stable range of $K$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1 0 0 1 0 0</td>
<td>$K &lt; 11.1$ and $18.2 &lt; K &lt; 22.5$</td>
</tr>
<tr>
<td>2</td>
<td>1 0 1 0 0 0</td>
<td>$K &lt; 18.75$</td>
</tr>
<tr>
<td>3</td>
<td>1 1 0 0 0 0</td>
<td>$K &lt; 17.5$</td>
</tr>
</tbody>
</table>

While the average system characteristics of the above three sequences are similar, the dynamic behavior of the Sequence #1 is significantly different from those of #2 and #3. This is apparently attributed to the even distribution of delayed sensor arrivals for #1 resulting in relatively more dominant effects of vacant sampling and message rejection. More extensive simulation studies are needed for a better understanding of these phenomena.

Next we investigate the physical phenomena resulting from vacant sampling and message rejection, and assess their detrimental effects on the system dynamic performance. To this effect, the flight control system consisting of the longitudinal motion dynamics of an advanced aircraft and a linear time-invariant control algorithm, described in Appendix A of our previous publication [3], was simulated under the following conditions.

- The network access protocol is the SAE Linear Token Passing Bus [4].
- The network medium is shared by 31 terminals.
- Each terminal has a single receiver of queue capacity equal to one.
- The network traffic is periodic with a sampling period of 10 ms for all terminals.
- The terminal #1 operates as both sensor and actuator terminals with its transmitter queue serving the sensor and its receiver queue serving the actuator and the terminal #2 as the controller.
- Every terminal, except the controller terminal, simultaneously generates a message at the beginning of each sample, i.e., the sampling instants of all terminals except the controller are synchronized with the sensor terminal.
- The sensor and controller terminals have a fixed message length with their information part being 64 bit long. The remaining terminals have identical message lengths which depend on the offered traffic.
- Processing delays at the controller is 1 ms.

Our earlier publication [3] prescribed that a network should be designed such that the offered traffic $G$ has a safe margin relative to its critical value $G_c$. Therefore, $G$ was selected to be 0.2, which is substantially less than the critical value, for conducting simulation runs. (For the condition given above, $G_c = 0.993$.)

Three cases with different values of skew $\Delta$ between the sensor and the controller terminals were considered.

- 5000 $\mu$s which is greater than $\delta_{\text{max}}$ ($\approx 2000 \mu$s);
- 2 $\mu$s which is less than $\delta_{\text{min}}$ ($= 4 \mu$s);
- 1000 $\mu$s which is approximately equal to $(\delta_{\text{max}} + \delta_{\text{min}})/2$.

The simulation run time was selected from 0.5 to 1.5 s, and 100 messages were generated for each terminal at an interval of 10 ms. At $\Delta = 5000 \mu$s, being greater than $\delta_{\text{max}}$, the sensor-controller delay $\delta_{\text{sc}}$ is always equal to $\Delta$. At $\Delta = 2 \mu$s, being less than $\delta_{\text{min}}$, $\delta_{\text{sc}}$ is always equal to $\Delta + T$. In both cases, exactly one sensor data reaches the receiver buffer of controller during every sampling period. At $\Delta = 1000 \mu$s, $\delta_{\text{sc}}$ would be either $\Delta$ or $\Delta + T$. This fluctuation in $\delta_{\text{sc}}$ results in vacant sampling and message rejection at the controller.

The transient responses of the control input $u(t)$ at the actuator, as a result of a unit step disturbance in the reference input, is given in Fig. 3 for three different values of $\Delta$. When a vacant sample occurs, the control signal is generated on the basis of the sensor data which was used in the previous sample since no fresh sensor data is received during that interval. When the next sensor data arrives at the controller terminal, the control input is derived on the basis of the new data. Figure 3 shows the distortion of the control signal under the above-mentioned transient disturbances for $\Delta = 1000 \mu$s as a result of vacant sampling and message rejection. The distortion of the control input causes high frequency noise in the actuator leading to excessive wear. The distortion may also excite the flexible modes of the aircraft. Implications of vacant sampling and message rejection are explained below.

In view of the four cases of sensor data arrival at the controller, discussed in Section 2, the relationship $z_k = y_k - p(k)$ between the delayed and true sensor data can be expressed in the following recursive form provided that $p(k)$ is either 0 or 1 $\forall k$. 

![Fig. 3 Distortion of control signal](image-url)
\[ z_k = z_{k-1} + [1-p(k)]\Delta y_k + p(k-1)\Delta y_{k-1} \quad (3.7) \]

where \( \Delta y_k = y_k - y_{k-1} \).

The above equation can be interpreted as \( z_k \) being the output of an integrator driven by the first differences \( \Delta y_k \) and \( \Delta y_{k-1} \) generated by the sequence \{\( y_k \)\}. If \( p(k) = p(k-1) \) as it is in the cases 1 and 2 in Section 2, the input to (3.7) is exactly one of the two first differences. For \( p(k) \neq p(k-1) \) implying message rejection or vacant sampling, either both or none of the two first differences are present. In this perspective, the phenomena of message rejection and vacant sampling can be expressed as

\[ z_k = z_{k-1} + \Delta y_k + e_k \quad (3.8) \]

where \( e_k \) represents a train of impulses with varying amplitudes. It is these impulses that distort the control signal sequence \{\( y_k \)\} as seen in Fig. 3. In the special case of alternate messages being rejected, the controller response could be worse than that with twice the sample period and no message rejection. The problem of vacant sampling can be partially circumvented by an observer which will provide an estimate of the delayed data. In that case the buffer capacity of the controller’s receiver should be increased so that the delayed data can be used for data reconstruction instead of simply being overwritten.

4 Analytical Approach to ICCS Design

The delayed control system model, presented in Part I [1], is time-varying and the currently available tools for analyzing time-varying systems do not offer a practical approach for the ICCS design unless the delays are periodic. For nonperiodic and stochastic delays, an alternative is to base the design procedure on minimizing the average delay in the control loop and the probabilities of vacant sampling and message rejection.

At first we consider the case of the sensor and controller sampling periods being (almost) identical where \( \Delta_s \) is maintained at a desired constant value. By appropriate selection of \( \Delta_s \), the delays in the control loop as well as the probability of vacant sampling may be minimized. This could be achieved without increasing the traffic in contrast to the case where the sensor sampling time is decreased. An analytical approach for selecting the optimal value of \( \Delta_s \) is outlined below.

Since variations in network traffic pattern are usually slow relative to the dynamics of the control system, the probability distribution of data latency can be considered to be stationary over a sufficiently large period of time. In addition, the assumption of ergodicity allows direct evaluation of the statistical characteristics of data latency from simulation or experimental results. In this way, the probability density function \( f_\delta(\cdot) \) of the sensor-to-controller data latency \( \delta_{sc} \) and controller-to-actuator data latency \( \delta_a \) can be obtained using the simulation program developed earlier [3]. Similarly, the joint probability density function \( f_{\delta k} \) of \( \delta_{sc} \) and \( \delta_{ac} \) at two consecutive sensor samples can be generated. It is important to note that \( f_\delta(\cdot) \neq 0 \) and \( f_{\delta k}(\cdot, \cdot) \neq 0 \) only if \( \theta, \xi \in [\delta_{min}, \delta_{max}] \). To select an optimal value of \( \Delta_s \), the following performance index is minimized:

\[ J = (1-\alpha)\phi(\theta_a) + \alpha P_{sr}, \quad \alpha \in [0,1] \quad (4.1) \]

where \( \theta_a = E[\delta_{ac}] + \delta_{ac} \) which is expected value of the total delay in the ICCS loop (see Fig. 2 in Part I), \( P_{sr} \) is the probability of vacant sampling at the controller and \( \phi \) is a given function of \( \theta_a \).

Since the sensor and controller have identical sampling frequencies, \( P_{sr} = P_{mr} \) which is the probability of message rejection. (See Remark 2.2.)

Several special cases of interest fit into the general performance index.

1. Minimum average delay: \( \phi(\theta_a) = 0 \); \( \alpha = 0 \).
2. Minimum vacant sampling/message rejection: \( \alpha = 1 \).
3. Maximum stability margin: \( \phi(\theta_a) = \varphi_0 - \varphi_1 \), where \( \varphi_0 \) and \( \varphi_1 \) are the phase margins of the given control system with no delay and a constant delay of \( \theta_a \), respectively. The reduction in the phase margin as a result of the additional delay is linear with respect to this delay and is given by \( \Delta \varphi = \omega_s \theta_a \) where \( \omega_s \) is the cross-over frequency of the system which includes the effects of sampling and zero-order-hold. Since \( \Delta \varphi \) is proportional to \( \theta_a \), this phenomenon is identical to that in the item 1 above. In contrast, the gain margin is a nonlinear function of \( \theta_a \). However, the use of phase margin does not preclude the use of a nonlinear function of \( \theta_a \) to further penalize delays close to the allowable minimum.

Now we consider the case: \( \phi(\theta_a) = \theta_a \). Since the controller-to-actuator delay \( \theta_{ac} \) is independent of \( \Delta_s \), only \( \theta_{ac} \) in the first term on the right hand side in (4.1) needs to be considered for the optimization procedure.

\[ E[\theta_{ac}] = \int \delta_{ac} \Delta f_\delta(\xi) d\xi + \int \delta_{ac} \Delta f_{\delta k}(\xi, \phi) d\xi \]

\[ = \Delta_s + T \int_{\delta_{ac}}^{\delta_{max}} f_\delta(\xi) d\xi \quad (4.2) \]

The second term \( P_{sr} \) in (4.1) is given as

\[ P_{sr} = \int_{\delta_{min}}^{\delta_{ac}} \int_{\delta_{min}}^{\delta_{ac}} d\phi d\theta_{ac} f_{\delta k}(\xi, \phi) \quad (4.3) \]

Using (4.2) and (4.3) in (4.1), sufficient conditions for minimization are obtained by setting \( \partial J/\partial \Delta_s = 0 \) and \( \partial^2 J/\partial \Delta_s^2 > 0 \) as:

\[ (1-\alpha) \left( 1 - T \int_{\delta_{min}}^{\delta_{ac}} f_\delta(\xi, \Delta_s) d\xi \right) - \alpha T \int_{\delta_{min}}^{\delta_{ac}} f_{\delta k}(\xi, \Delta_s) d\xi + \alpha T \int_{\delta_{ac}}^{\delta_{max}} f_{\delta k}(\xi, \Delta_s, \phi) d\phi = 0 \quad (4.4) \]

and

\[ -T(1-\alpha) \int_{\delta_{min}}^{\delta_{ac}} \int_{\delta_{ac}}^{\delta_{max}} d\phi d\theta_{ac} f_{\delta k}(\xi, \phi)/\partial \Delta_s d\xi \]

\[ + \alpha T \int_{\delta_{ac}}^{\delta_{max}} d\phi f_{\delta k}(\xi, \Delta_s, \phi)/\partial \Delta_s d\phi - 2\alpha T f_{\delta k}(\xi, \Delta_s, \phi) > 0 \quad (4.5) \]

To gain a better insight of the problem, we consider three special cases.

(i) \( \alpha = 0 \): In this case, \( \partial J/\partial \Delta_s = 0 \) yields \( f_\delta(\xi) = 1/T \) using (4.4) and \( \partial^2 J/\partial \Delta_s^2 > 0 \) yields \( f_{\delta k}(\xi, \Delta_s) = \delta_{min} < 0 \) using (4.5). If the network is not overloaded, implying that \( \delta_{max} = \delta_{min} > T \), then the density function \( f_\delta(\xi) \) must exceed \( 1/T \) for some \( \xi \). If \( f_\delta(\xi) > 1/T \forall \xi \in [\delta_{min}, \delta_{max}] \), then \( \Delta_s = \delta_{max} \). Figure 4 illustrates these phenomena.

(ii) From (4.3) or directly from Section 2, it follows that \( P_{sr} = 0 \) if \( \Delta_s \in [\delta_{min}, \delta_{max}] \). In these ranges, the \( \Omega_{ac} \) is held constant at \( \delta_{max} \) if \( \delta_{ac} = \delta_{max} \).

(iii) If the ICCS network serves a large number of terminals and a majority of them has a stationary random traffic, the sensor-controller delays \( \delta_{ac} \) and \( \delta_{ac} + \delta_{ac} \) at consecutive samples could be assumed to be independent and identically distributed, i.e., \( f_{\delta k}(\xi, \phi) = f_{\delta k}(\xi) f_{\delta k}(\phi) \). Then, (4.4) and (4.5) yield less complex expressions for computation of the optimal \( \Delta_s \).
Next we outline the framework of a design approach when the sensor and controller sampling periods are deliberately made different. In this case, $\Delta_e$ will vary periodically depending on the design parameter $\epsilon = (T_e - T_c)/T_e$, and the corresponding performance index needs to be minimized with respect to $\epsilon$ instead of $\Delta_e$. From the Proposition 2.1 and its Corollary, it follows that $\epsilon > 0$ reduces the probability of vacant sampling and increases that of message rejection. While reduced vacant sampling improves the quality of control signals, message rejections increase the network traffic with ineffective utilization of the resources. Therefore the performance index should be structured in terms of average delay and probabilities of vacant sampling and message rejection as:

$$J = (1 - \beta - \gamma)\phi(\theta_e) + \beta TP_{us} + \gamma TP_{mr}$$

(4.6)

where $\beta$ and $\gamma$ are non-negative weights such that $\beta + \gamma \epsilon \in [0, 1]$.

From the system design perspectives, the controller sampling period can be maintained constant, and the sensor sampling period $T_i$ be varied to realize perturbations in $\epsilon$. Then the design task is to determine analytical expressions for $\theta_e$, $P_{us}$, and $P_{mr}$ as functions of $\epsilon$ or $T_i$. This could be accomplished on the basis of simulated network traffic characteristics. The final step is to evaluate an optimal $\epsilon$ using the sufficiency conditions: $\delta J/\partial \epsilon = 0$ and $\delta^2 J/\partial \epsilon^2 > 0$. This is also a subject of the current research.

5 Summary, Conclusions, and Recommendations for Future Work

The paper addresses the design issues of Integrated Communication and Control Systems (ICCS) for complex dynamical processes like advanced aircraft, spacecraft, and autonomous manufacturing plants. The ICCS utilize asynchronous time-division multiplexed networking which introduces time-varying and possibly stochastic delays between system components.

One of the major emphases in this paper is on the usage of the discrete-time, finite-dimensional model of the delayed control system which has been developed in Part I [1] for ICCS design. The impact of network delays and possible mis-synchronization between the sensor and controller was investigated, and their physical significance relative to the system stability has been exemplified. The simulation of the flight dynamics of an advanced aircraft shows that a combination of time-varying data latency in the network and mis-synchronization between system components could cause vacant sampling at the controller and distort the control inputs to the actuator.

Two alternative ICCS design procedures which require a combination of analytical and simulation techniques, have been outlined. Both procedures are based on minimizing a weighted sum of the average time delay in the control loop and the probabilities of message rejection and vacant sampling, and offer the following options: (1) Identical sampling rates of the sensor and controller which are to be periodically synchronized to maintain the desired time skew between their sampling instants; and (2) faster sampling rates for the sensor in order to reduce the occurrence of vacant sampling. Another major research area for analysis and design of ICCS is the parametric evaluation of the system stability under non-periodic and stochastic delays.

The current research is focused on the development of the above analytical techniques and their verification by combined discrete-event and continuous-time simulation. The next planned phase is to experimentally verify these procedures at the network testbed described in Appendix B.

Acknowledgments

The authors acknowledge the contributions of Messrs. S. H. Hong, L-W. Liou, S. Lee, and R. Luck in this paper.

References


APPENDIX A

Supporting Proposition

Proposition A.1: Let the clocks of two terminals be synchronized at $t=0$ and their sampling times be $T+\epsilon$ and $T$, respectively. Then, as $t \to \infty$, the probability distribution of the skew $\Delta e\in[0, T)$ provided that $\epsilon$ is independent of the probability distribution of $\epsilon$ provided that $\epsilon$ is bounded, i.e., $|\epsilon| < \epsilon < \infty$.

Proof: Let the random variable $\epsilon \geq 0$ have a probability distribution function $F_\epsilon(x)$ and density function $f_\epsilon(x)$. At the $k$th sampling instant, the skew $\Delta e(0, T)$ is given as:

$$\Delta e = \text{Rem}[k(T + \epsilon) - kT, T] = \text{Rem}[k\epsilon, T]$$

where $\text{Rem}[\epsilon, \phi]$ is the remainder when $\epsilon$ is divided by $\phi$. The distribution function $F_\epsilon(x)$ of $\Delta e$ can be expressed in terms of the distribution function $F_\epsilon(x)$ of $\epsilon$ as:

$$F_\epsilon(x) = P(\Delta e \leq x) = \sum_{j=0}^{N} P(T < k\epsilon \leq (JT + x))$$

$$= \sum_{j=0}^{N} [F_\epsilon((JT + x)/k) - F_\epsilon((JT)/k)] \quad \text{for } x \in [0, T)$$

(1.1)

where $N =$ Integer part of $(k\epsilon/T)$. Differentiating (1.1) with respect to $x$, we obtain the density function $f_\epsilon(x)$ as:

$$f_\epsilon(x) = (1/k) \sum_{j=0}^{N} f_\epsilon((JT + x)/k), \quad x \in [0, T)$$

(1.2)

As $k \to \infty$ (and consequently $N \to \infty$) under steady state conditions, the summation in (1.2) emerges as
\[ \lim_{k \to \infty} f_k(x) = \frac{1}{\mathcal{E}} \int_0^\infty f_k(\tau)d\tau = \frac{1}{T} \text{ for } x \in [0, T) \]  

(A.3)

Hence \(\Delta\) is uniformly distributed in \([0, T)\) under steady states.

**APPENDIX B**

**Description of the Network Testbed**

The 10 Mbps network testbed is designed to experiment with IEEE 802.4 (linear) token passing bus and IEEE 802.2 logical link control protocols in an ISO compatible network architecture which includes the Transport Protocol (TP) [5]. Under normal operating conditions, the characteristics of IEEE 802.4 are similar to those of SAE linear token passing bus [4].

The testbed is equipped with three terminal interface units, and six host computers including a micro-Vax which can communicate with each other via the network medium or directly using a pair of communication ports. For the ICCS test facility, a tacho-generator and a servomotor function as the sensor and the plant, respectively, and are connected to one microcomputer of the network. Another microcomputer serves as the controller.

A traffic load generator (TLG) has been designed to emulate the scenario of a large number of stations and varying traffic on the network by use of only two terminal interface units. The TLG emulates the major features of the IEEE 802.4 token passing bus protocol. The locations of all odd numbered stations are simulated on the microcomputer host #1 and those of even numbered stations on the microcomputer host #2 where the two microcomputers communicate with each other via the network. The host #1 also provides the A/D and D/A conversion functions for the tacho-generator and the servomotor and is designated as the station #1 in the logical ring of the token passing network of the TLG. The controller is located at host #2 and could be designated with any even station number, i.e., the controller station can be hosted in different positions in the logical ring relative to the sensor/actuator station. Since the host #2 functions as an integral part of the TLG, another microcomputer host #3 is directly connected to host #2. This host #3 provides a multi-processing environment for host #2. The control algorithm is resident in host #3.