Introduction

- **Real-Time Control Systems**
  - Controlled by one **Computer Processor**
    - Centralized control systems
    - Real-time operating systems
  - Controlled by one **Communication Medium**
    - Distributed control systems
    - Real-time communications

References

- Abstract:
  - Integrated communication and control systems (ICCS) consist of several distributed control processes which share a network medium. Performance of several feedback control loops in the ICCS is subject to the network-induced delays from sensor to controller and from controller to actuator. The network-induced delays are directly dependent upon the data sampling times of the control components which share a network medium. In this study, a scheduling algorithm of determining data sampling times is developed using the window concept, where the sampled data from the control components in the ICCS share a limited number of windows, so that the performance requirement of each control loop is satisfied as well as the utilization of network resources is considerably increased. The scheduling algorithm is verified by discrete-event/continuous-time simulation model of an example of ICCS

References
Example of NCS Design

- **5 Loops with 10 Data-TX Nodes**
  - Packetized data tx time (L): 2 ms
  - Server overhead (σ): 0.1 ms
  - Maximum allowable loop delays:
    - \[[Φ₁, Φ₂, Φ₃, Φ₄, Φ₅]\] = [25, 60, 100, 200, 400] (ms)
  - Periods of each loop:
    - \[[T₁, T₂, T₃, T₄, T₅]\] = [9, 18, 36, 72, 144] (ms)
    - \[[k₁, k₂, k₃, k₄, k₅]\] = [1, 2, 4, 8, 16] (ms)
    - \[[t₁, t₂, t₃, t₄, t₅, t₆, t₇, t₈, t₉, t₁₀]\] = [0, 0, 0, 0, 9, 9, 27, 27, 63, 63] (ms)
    - r = 4
  - Network and system utilizations:
    - U = 86.1% and Uₛ = 96.8%
### Network-induced delays in control loop $i$:

$T_i$ : sampling time in loop $i$

$\theta_i$ : sensor-controller delay

$\ell_i$ : controller-actuator delay

### Loop delays in control loop $i$:

$D_i$ : loop delay, $D_i = \left[\frac{1}{T_i} + \ell_i\right] T_i + 2\theta_i$

### Data rejection and Vacant Sampling:

$\Phi_i$ : pre-determined max allowable loop delay

$\ell_i < T_i, m = 1, 2$ : eliminate data rejection & vacant sampling

$D_i \leq \Phi_i$ : loop delay should not exceed its limitation
**Time-Varying Delays:**

- \( t_k \) : the \( k \)th instant
  - when actuator command arrives at plant
- \( \theta_i \) is time-varying,
  - \( D_i \) and \( (t_k, t_{k+1}) \) are also time-varying
- \( (t_k, t_{k+1}) \) becomes max
  - when \( \min^2 \theta_i \) occurs at the \( k \)th arrival
    - followed by \( \max^2 \theta_i \) occurs at the \((k+1)\)th arrival

\[ \sup(t_k, t_{k+1}) = T_i + (\sup^2 \theta_i - \min^2 \theta_i) \]

\( D'_i \) : modified constant loop delay
\[ D'_i = 2T_i + (\sup^2 \theta_i - \min^2 \theta_i) \]

\[ \Rightarrow D'_i = 2T_i + (\sup^2 \theta_i - \min^2 \theta_i) \leq \Phi_i \]

**Stability Region for Time-Varying Delays:**

\[ \dot{x}(t) + \alpha x(t) + \beta x(t - \theta(t)) = 0 \]

---

**Algorithm of Determining Data Sampling Times**

**Parameters & Variables:**

- \( T = [T_1, T_2, \cdots, T_M] \), with \( T_i \leq T_{i+1}, \forall i \)
- \( r_i \) : number of data served by the server
during the worst-case data latency
  - \( \max m^i \theta_i = r_iL + N\sigma, \ m = 1, 2 \)
  - because \( \max m^i \theta_i \leq T_i \)
  - \( r_i = \left[ \frac{T_i - N\sigma}{L} \right] \)

**Parameters & Variables:**

- \( N \) nodes share the \( r \) windows \((r = r_1, N > r)\)
- \( T_j = k_j T_1, \ (k_j \geq 1) \)
  - During \( T_1 \), if number of data served \( \leq r \)
  - During \( T_j \), data served \( \leq r_j = k_j r \)
  - Node \( j \) never overflow
Algorithm of Determining Data Sampling Times

**Parameters & Variables:**

\[ D'_i = 2T_i + (\sup \theta_i - \min \theta_i) \]

\[ r_i = \frac{T_i - N\sigma}{L} \]

\[ \Rightarrow \min \theta_1 = L \]

\[ \sup \theta_1 = T_1 \]

\[ D' = \Phi_1 (= \min [\Phi_i, \, i = 1, \ldots, M]) \]

\[ T_1 = \frac{\Phi_1 + L}{3} \]

\[ r = \frac{(\Phi_1 + L)/3 - N\sigma}{L} \]

\[ K = [k_1, k_2, \cdots, k_M], \quad k_i = \frac{T_i}{T_1}, \quad k_i \leq k_{i+1}, \forall i \]

\[ \alpha_K = 2 \sum_{i=1}^{M} \frac{1}{k_i} \]

\[ \Rightarrow \text{If } \alpha_K \leq r \text{ and } \text{Rem}[k_j, k_i] = 0, \forall i, j \geq i \]

\[ \Rightarrow \text{all the data sampled during } T_M (= k_M T_1) \]

\[ \text{can be accommodated by offered windows} \]

**Scheduling Feasibility:**

\[ \Rightarrow \text{For any node } j \text{ with } T_j \leq T_M \]

\[ \Rightarrow \text{If Rem}[k_M, k_j] = 0 \]

\[ \Rightarrow \text{the number of data generated from node } j \text{ during } T_M \text{ is fixed to } k_M/k_j \]

\[ \Rightarrow \text{the total number of data generated from all nodes during } T_M \text{ is} \]

\[ k_M \sum_{j=1}^{N} (1/k_j) = 2k_M \sum_{j=1}^{M} (1/k_j) = k_M\alpha_K \]

\[ \Rightarrow \text{If } \alpha_K > r \]

\[ \Rightarrow \text{the network system is overloaded} \]

\[ \Rightarrow \{ \text{increase data rate } (B), \text{ reduce overhead } (\sigma), \text{ reduce number of nodes } (N) \} \]
Algorithm of Determining Data Sampling Times

**Scheduling Algorithm:**

- $A_i$: the beginning instant of the $i$th $T_1$
- $u^N(A_i)$: number of sampled data among node 1 upto $n$ at $A_i$
- $t_j$: sampling instant of node $j$ in $T_M$
  - $t_1 = A_1$
  - $u^l(A_i) = 1, l = 1, ..., k_M$
  - $t_j = \inf \{ A_i \geq A_{i-1} : u^l(A_i) \leq r \}, j = 2, ..., N$

⇒ No more than $r$ data can be served at $T_1$
  - $\sup^m \theta_i = T_1, \ \min^m \theta_i = L, \ \forall i, m$

Example of ICCS Design

- 5 Loops with 10 Data-TX Nodes:
Example of NCS Design

- **5 Loops with 10 Data-TX Nodes**
  - Packetized data tx time (L): 2 ms
  - Server overhead (σ): 0.1 ms
  - Maximum allowable loop delays:
    - \([\Phi_1, \Phi_2, \Phi_3, \Phi_4, \Phi_5] = [25, 60, 100, 200, 400] \text{ (ms)}\)
  - Periods of each loop:
    - \([T_1, T_2, T_3, T_4, T_5] = [9, 18, 36, 72, 144] \text{ (ms)}\)
    - \([k_1, k_2, k_3, k_4, k_5] = [1, 2, 4, 8, 16] \text{ (ms)}\)
    - \([t_1, t_2, t_3, t_4, t_5, t_6, t_7, t_8, t_9, t_{10}] = [0, 0, 0, 0, 9, 9, 27, 27, 63, 63] \text{ (ms)}\)
    - \(r = 4\)
  - Network and system utilizations:
    - \(U = 86.1\%\) and \(U_s = 96.8\%\)

Example of ICCS Design

- **For Loop 1:**
  - NORMAL: \(T_1 = 9\)
  - CASE 1: \(T_1 = 6.3\)
  - CASE 2: \(T_1 = 13.5\)

  \([-9, 18, 36, 72, 144]\)

  \([-9, 17, 37, 83, 161]\)

  \([-9, -9, 18, 36, -72]\)

  \(U_3 = U_4, U_4 > 1\)