SPRING 2010

即時控制系統設計
Design of Real-Time Control Systems

Lecture 34
Scheduling Sampling Times of Networked Control Systems

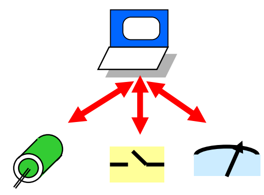
Feng-Li Lian

NTU-EE

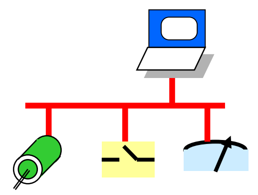
Feb10 – Jun10

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



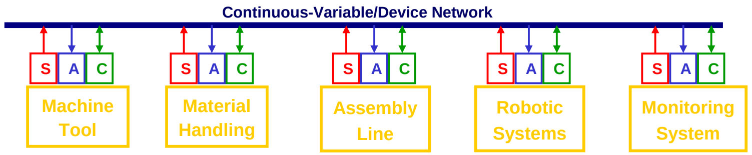
Centralized Control System

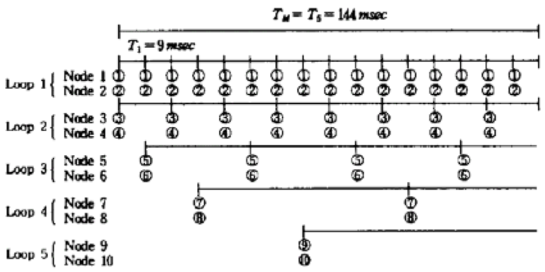


Distributed Control System

- S.H. Hong, "Scheduling algorithm of data sampling times in the integrated communication and control systems," IEEE-CST, Vol. 3, No. 2, pp. 225-230, June 1995
- S.H. Hong, "Bandwidth allocation scheme for cyclic-service fieldbus networks," IEEE/ASME Mechatronics, Vol. 6, No. 2, pp. 197-204, June 2001
- S.H. Hong and Y.C. Kim, "Implementation of a bandwidth allocation scheme in a token-passing fieldbus network," IEEE Instrumentation & Measurement, Vol. 51, No. 2, pp. 246-251, Apr. 2002
- E.Tovar, F. Vasques, and A. Burns, "Supporting real-time distributed computer-controlled systems with multi-hop P-NET networks," Control Engineering Practice, Vol. 7, No. 8, pp. 1015-1025, Aug. 1999
- S. Cavalieri, A.D. Stefano, and O. Mirabella,, "Impact of fieldbus on communication in robotic systems," IEEE-RA, Vol. 13, No. 1, pp. 30-48, Feb. 1997

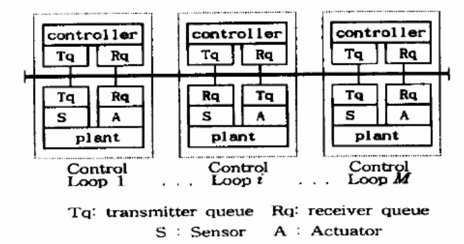
- S.H. Hong,
- "Scheduling algorithm of data sampling times in the integrated communication and control systems,"
- IEEE-CST, Vol. 3, No. 2, pp. 225-230, June 1995
- Abstract:
  - Integrated communication and control systems (ICCS) consist of several distributed control processes which share a network 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





- 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] (ms)
  - Periods of each loop:
    - $[T_1, T_2, T_3, T_4, T_5]$ = [9, 18, 36, 72, 144] (ms)
    - $[k_1, k_2, k_3, k_4, k_5]$ = [1, 2, 4, 8, 16] (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] (ms)
    - r = 4
  - Network and system utilizations:
    - U = 86.1% and U_s = 96.8%

- Schematic diagram:



$\Rightarrow M$: control loops

$\Rightarrow N = 2M$: node number

- Key Parameters:

$\Rightarrow T_i$: sampling time in loop i

$\Rightarrow L'$: actual message size

$\Rightarrow \bar{L}$: packetized message size

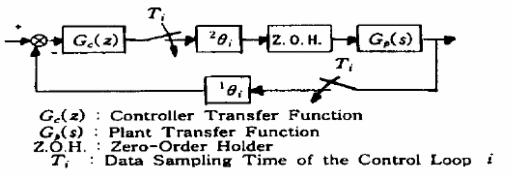
$\Rightarrow p = \lceil L'/\bar{L} \rceil$: number of segmented packets

$\Rightarrow T_i/p$: interval inserting tx queue

$\Rightarrow B$: data rate of network

$\Rightarrow L = \bar{L}/B$: packetized data tx time

- Network-induced delays in control loop i:

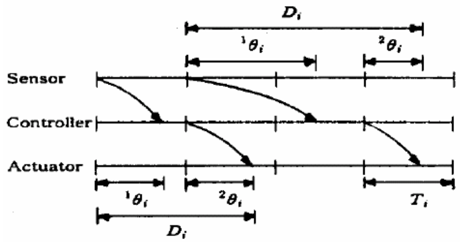


$\Rightarrow T_i$: sampling time in loop i

$\Rightarrow {}^1\theta_i$: sensor-controller delay

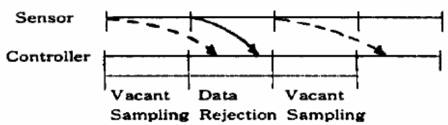
$\Rightarrow {}^2\theta_i$: controller-actuator delay

- Loop delays in control loop i:



$\Rightarrow D_i$: loop delay, $D_i = \left\lceil \frac{^1\theta_i}{T_i} \right\rceil T_i + {}^2\theta_i$

- Data rejection and Vacant Sampling:



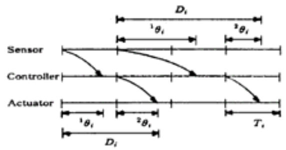
- Design Objective:

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

$\Rightarrow {}^m\theta_i < T_i, m = 1, 2$:
eliminate data rejection & vacant sampling

$\Rightarrow D_i \leq \Phi_i$:
loop delay should not exceed its limitation

▪ Time-Varying Delays:



⇒ t_k : the kth instant

when actuator command arrives at plant

⇒ Since ${}^2\theta_i$ is time-varying,

→ D_i & $[t_k, t_{k+1})$ are also time-varying

⇒ $[t_k, t_{k+1})$ becomes max

when $\min {}^2\theta_i$ occurs at the kth arrival

followed by $\max {}^2\theta_i$ occurs at the $(k+1)$th arrival

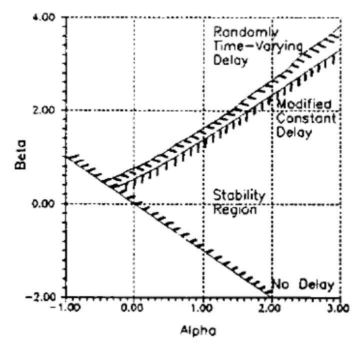
⇒ $\sup[t_k, t_{k+1}) = T_i + (\sup {}^2\theta_i - \min {}^2\theta_i)$

⇒ D_i' : modified constant loop delay

$$D_i' = 2T_i + (\sup {}^2\theta_i - \min {}^2\theta_i)$$

▪ Stability Region for Time-Varying Delays:

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



$$\Rightarrow D_i' = 2T_i + (\sup {}^2\theta_i - \min {}^2\theta_i) \le \Phi_i$$

▪ Parameters & Variables:

⇒ $T = [T_1, T_2, \cdots, T_M]$, with $T_i \le T_{i+1}, \forall i$

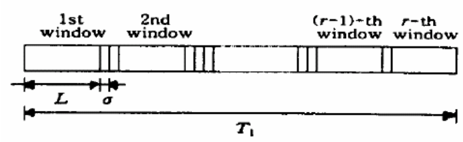
⇒ r_i : number of data served by the server during the worst-case data latency

→ $\max {}^m\theta_i = r_i L + N\sigma, \quad m = 1, 2$

→ because $\max {}^m\theta_i \le T_i$

→ $r_i = \left\lfloor \frac{T_i - N\sigma}{L} \right\rfloor$

▪ Parameters & Variables:



⇒ N nodes share the r windows $(r = r_1, N > r)$

⇒ $T_j = k_j T_1, \quad (k_j \ge 1)$

→ During T_1, if number of data served $\le r$

→ During T_j, data served $\le r_j = k_j r$

⇒ Node j never overflow

$$D_i' = 2T_i + (\sup {}^2\theta_i - \min {}^2\theta_i)$$

$$r_i = \left\lfloor \frac{T_i - N\sigma}{L} \right\rfloor$$

- Parameters & Variables:

$\Rightarrow \min \; {}^2\theta_1 = L$

$\Rightarrow \sup \; {}^2\theta_1 = T_1$

$\Rightarrow D' = \Phi_1 \; (= \min [\, \Phi_i, \; i = 1, ..., M \,])$

$$\rightarrow T_1 = \frac{\Phi_1 + L}{3}$$

$$\rightarrow r = \left\lfloor \frac{(\Phi_1 + L)/3 - N\sigma}{L} \right\rfloor$$

- Parameters & Variables:

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

$$\Rightarrow \alpha_K = 2 \sum_{i=1}^{M} \frac{1}{k_i}$$

$\rightarrow$ If $\alpha_K \le r$ and $\mathrm{Rem}[k_j, k_i] = 0, \forall i, j, j \ge i$

$\rightarrow$ all the data sampled during $T_M \; (= k_M T_1)$ can be accommodated by offered windows

- Scheduling Feasibility:

$\Rightarrow$ For any node j with $T_j \le T_M$

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

$\rightarrow$ the number of data generated from node j during T_M is fixed to k_M / k_j

$\Rightarrow$ the total number of data generated from all nodes during T_M is

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

- Scheduling Feasibility:

$\Rightarrow$ If $\alpha_K \le r$

$\rightarrow$ all the data sampled during $T_M \; (= k_M T_1)$ can be accommodated by offered windows

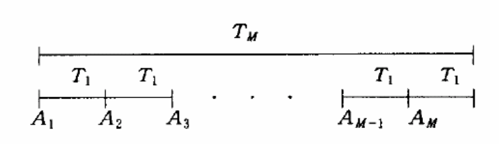
$\rightarrow$ feasibility condition of scheduling algorithm

$\Rightarrow$ If $\alpha_K > r$

$\rightarrow$ the network system is overloaded

$$\rightarrow \begin{cases} \text{increase data rate } (B) \\ \text{reduce overhead } (\sigma) \\ \text{reduce number of nodes } (N) \end{cases}$$

- **Scheduling Algorithm:**



⇒ A_l : the beginning instant of the lth T_1

⇒ $u^n(A_l)$: number of sampled data among node 1 upto n at A_l

⇒ t_j : sampling instant of node j in T_M

→ $t_1 = A_1$

→ $u^1(A_l) = 1, l = 1, ..., k_M$

→ $t_j = \inf[A_l \geq A_{l-1} : u^j(A_l) \leq r], j = 2, ..., N$

- **Scheduling Algorithm:**

⇒ Since $\text{Rem}[k_j, k_{j-1}] = 0$,

→ $T_{j-1} = nT_j$, n: integer

→ $u^j(A_l + nT_j) \leq r, \forall n = 1, 2, ...$

→ Hence, $u^N(A_l) \leq r, \forall l = 1, ..., k_M$

⇒ No nodes experience overflow

⇒ No more than r data can be served at T_1

→ $\sup^m \theta_i = T_1, \quad \min^m \theta_i = L, \quad \forall i, m$

→ $2T_i + (T_1 - L) \leq \Phi_i, \quad i = 2, ..., M$

- **Scheduling Algorithm:**

⇒ Other T_i, $i = 2, ...M$

→ $T_i = k_i T_1, \quad k_i = \left\langle \frac{\Phi_i - (T_1 - L)}{2T_1} \right\rangle$

⇒ For light loaded network,

→ $r \geq N$

→ $T_i = \frac{\Phi_i - (T_1 - L)}{2}, \quad \forall i = 1, ..., M$

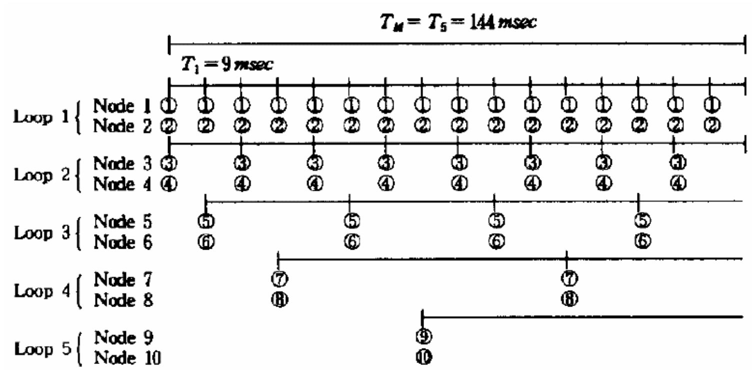
⇒ Network Utilization:

→ $U = \sum_{j=1}^{N} \frac{L}{T_j} = \frac{2L}{T_1} \sum_{i=1}^{M} \frac{1}{k_i}$

⇒ System Utilization:

→ $U_s = \frac{\#\ \text{windows used during}\ T_M}{\#\ \text{windows offered}} = \frac{\alpha_K}{r}$

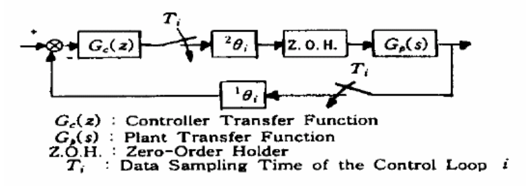
- **5 Loops with 10 Data-TX Nodes:**



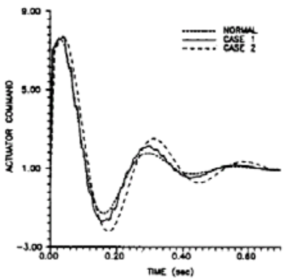
▪ 5 Loops with 10 Data-TX Nodes

- 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] (ms)
- Periods of each loop:
  - $[T_1, T_2, T_3, T_4, T_5]$ = [9, 18, 36, 72, 144] (ms)
  - $[k_1, k_2, k_3, k_4, k_5]$ = [1, 2, 4, 8, 16] (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] (ms)
  - r = 4
- Network and system utilizations:
  - U = 86.1% and U_s = 96.8%

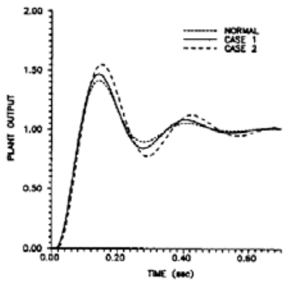
▪ For Loop 1:

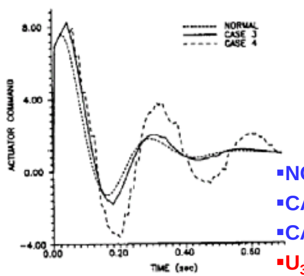


$$G_c(s) = \frac{7(s+5)}{s} \qquad G_p(s) = \frac{1}{(0.3s+1)(0.03s+1)}$$



▪NORMAL: T_1 = 9
▪CASE 1: T_1 = 6.3
▪CASE 2: T_1 = 13.5





▪NORMAL: 9, 18, 36, 72, 144
▪CASE 3: 9, 17, 37, 83, 161
▪CASE 4: 9, 9, 18, 36, 72
▪U_3 == U_N, U_4 > 1