

Spring 2019

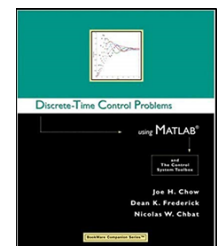
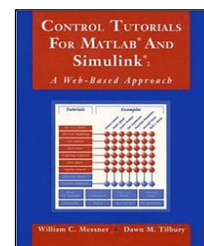
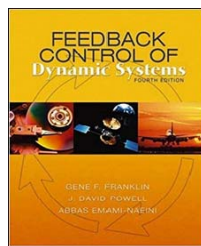
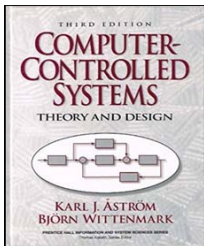
數位控制系統  
Digital Control Systems

DCS-15  
Timing Analysis

Feng-Li Lian

NTU-EE

Feb19 – Jun19

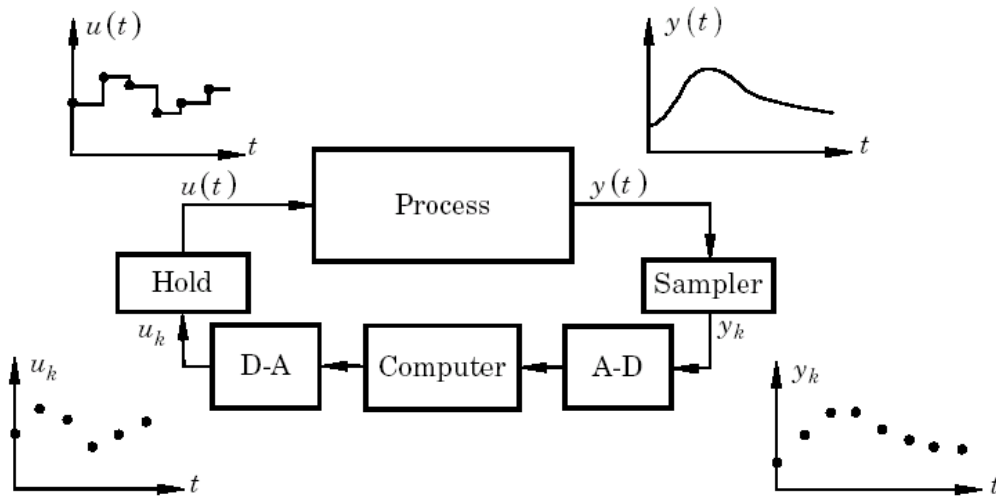


Outline

Feng-Li Lian © 2019  
DCS15-Timing-2

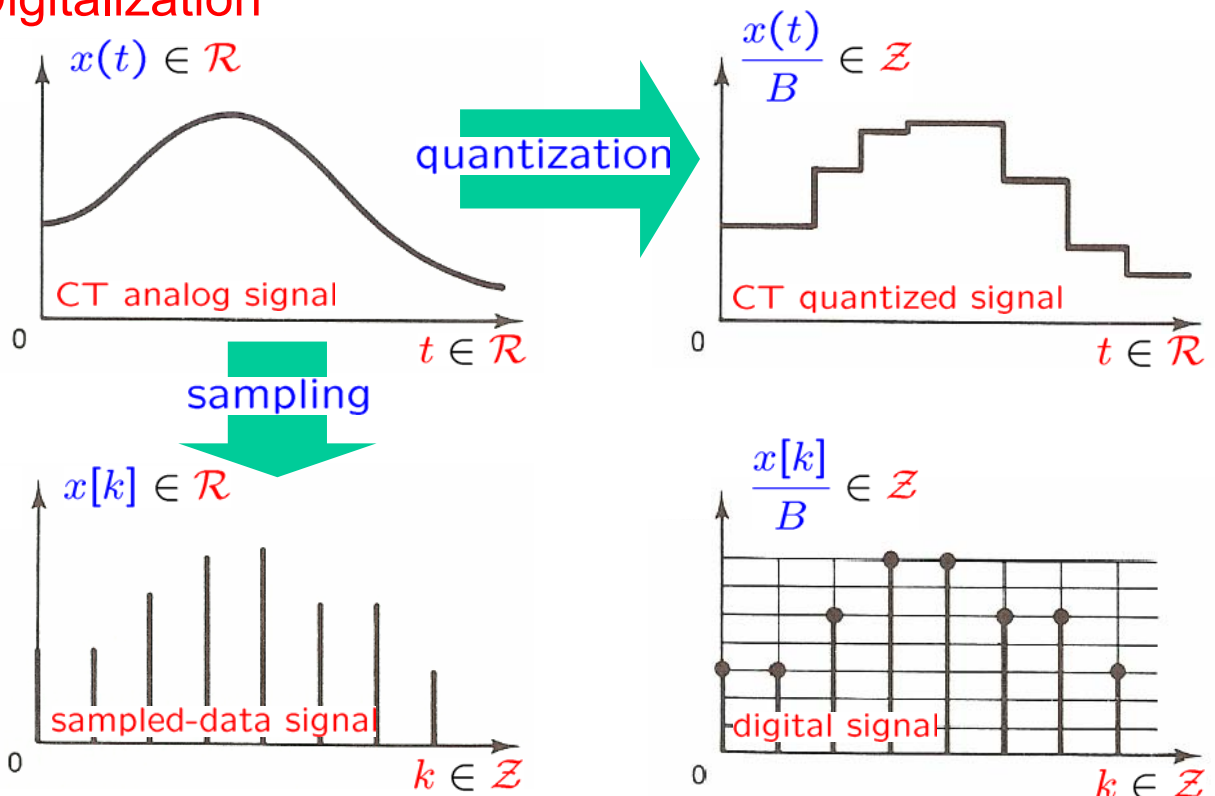
- Digitalization
- Timing Analysis in a Control System
- Some Related Issues:
  - Multiple or Random Time-Delay Systems
  - Types of Jitters
  - Timing Requirements & Control Attributes
  - Multiprocessor Implementation of Digital Engine Control
  - End-to-End Delay of Videoconferencing
  - Timing Analysis for Programs
  - Temporal Characteristics of Task Transmission

Control System Block Diagram



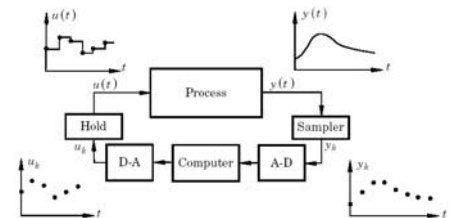
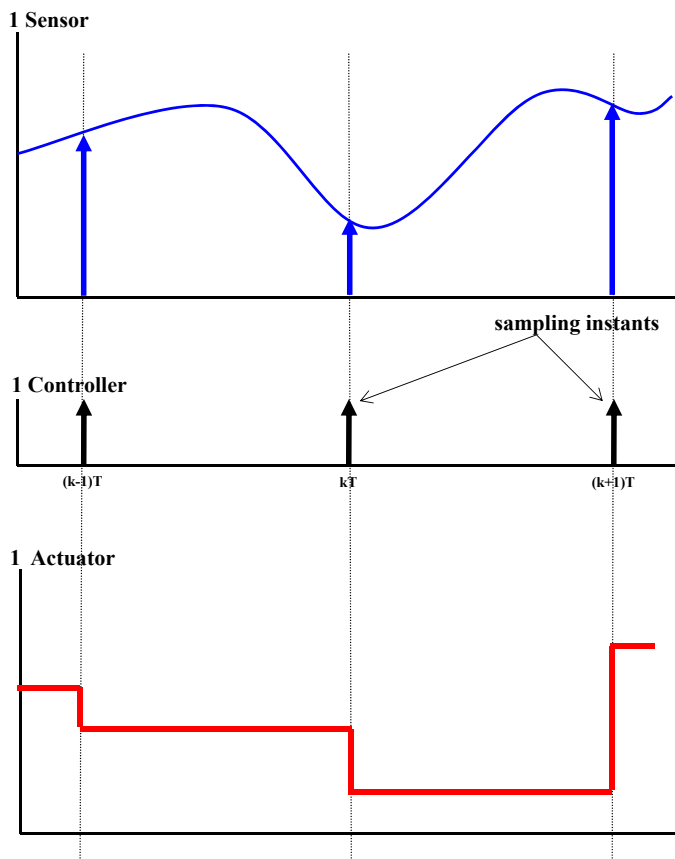
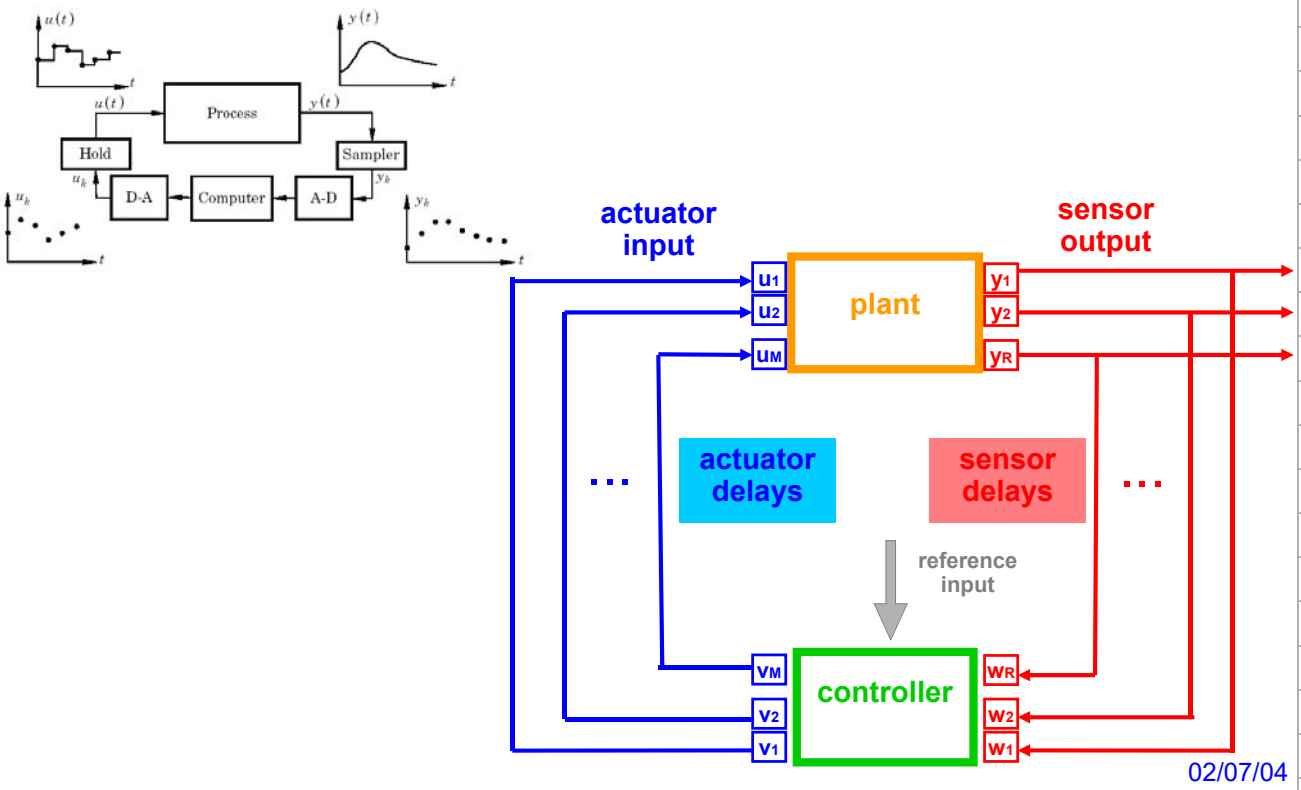
Astrom & Wittenmark 1997

Digitalization

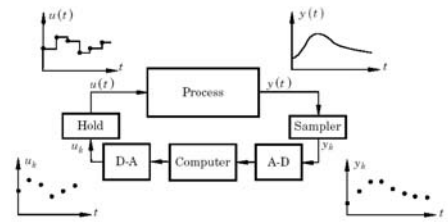
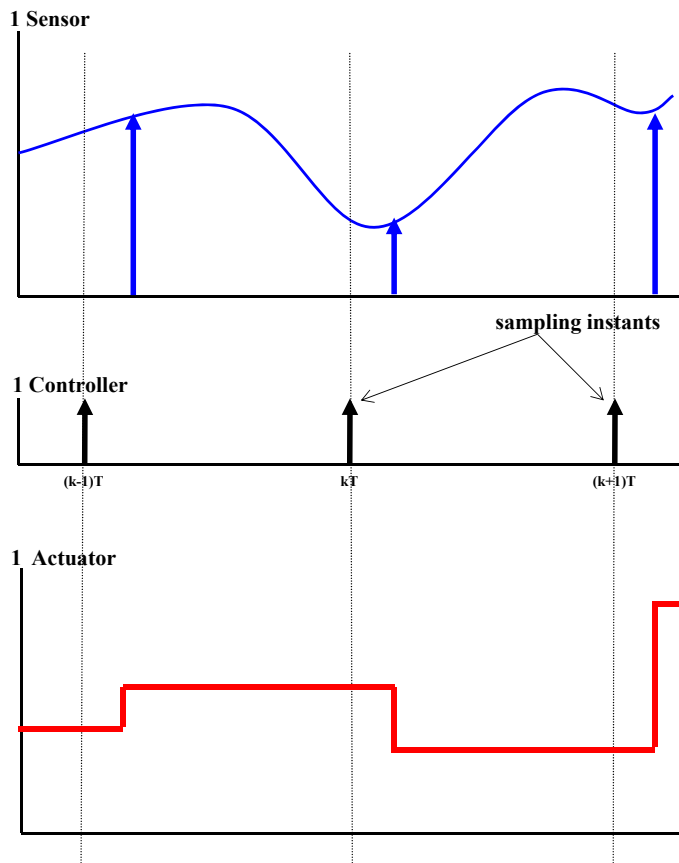


Ogata 1995

- Single-Input-Single-Output and Multiple-Input-Multiple-Output
- Single Delay and Multiple Delays

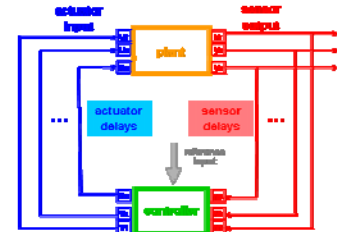
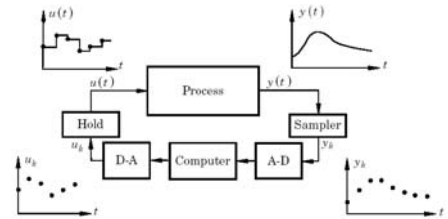
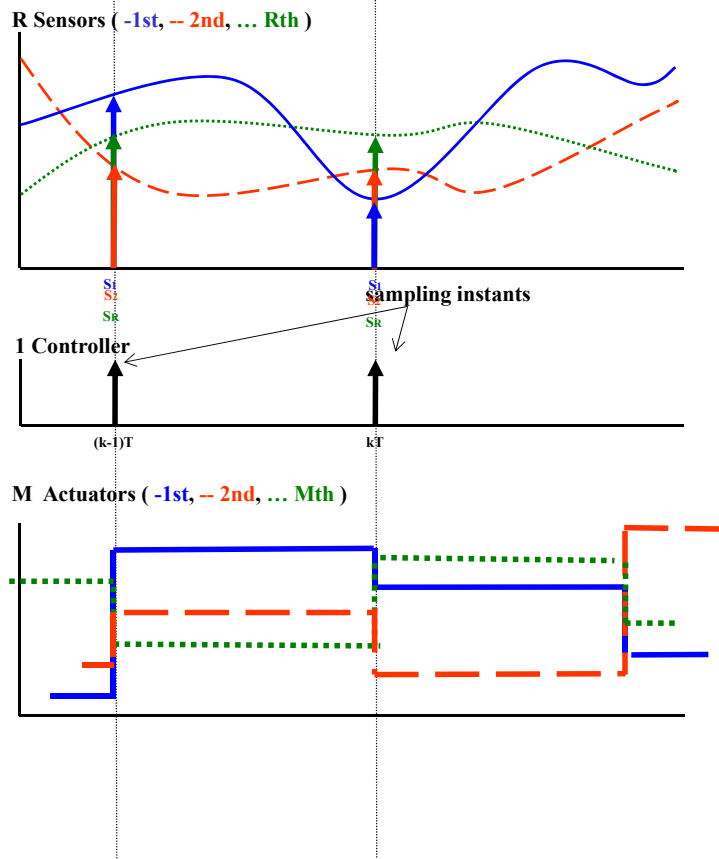


# Timing Analysis: Single-Input-Single-Output with Single Delay

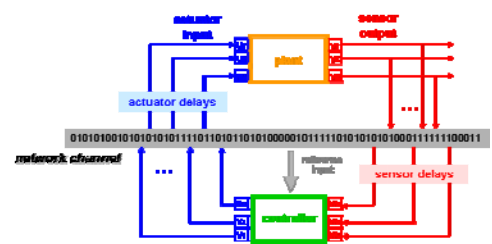
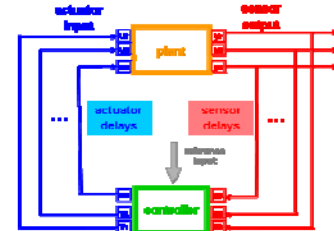
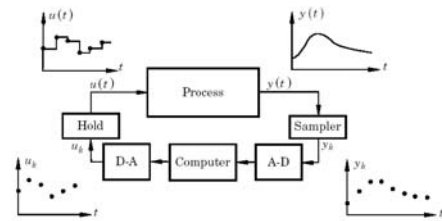
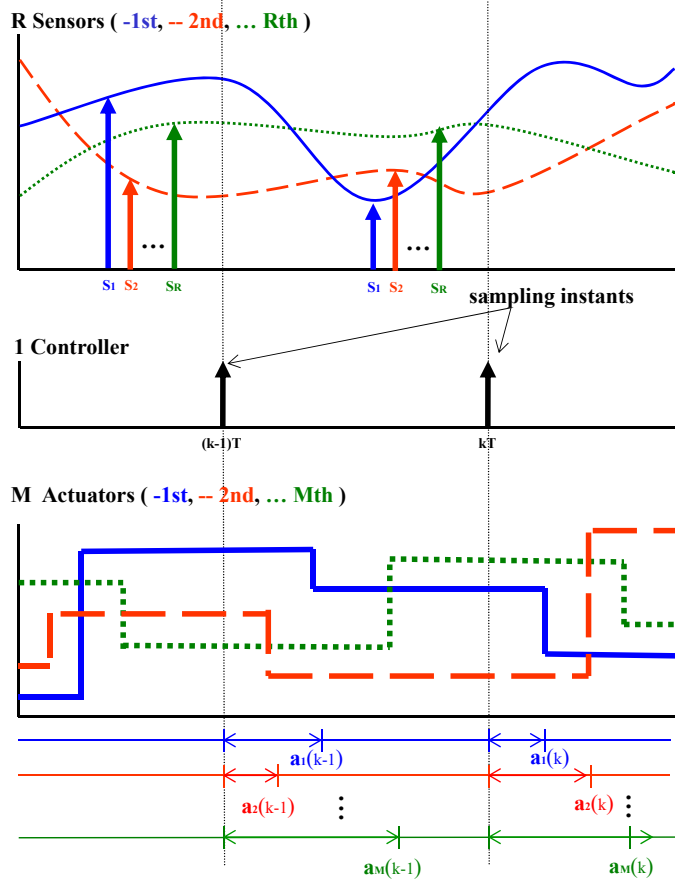


02/24/04

# Timing Analysis: Multiple-Input-Multiple-Output



02/24/04



02/24/04

## Multiple or Random Time-Delay Systems

### Paper:

- B. Wittenmark, J. Nilsson, and M. Torngren,
- "Timing problems in real-time control systems,"
- In Proceedings of American Control Conference, Seattle, Washington, pp. 2000–2004, June 1995.

### Abstract:

- In this paper we have discussed some of the timing problems in real-time control systems. The influence of the scheduling on the models is discussed together with different interesting problem formulations. The effect of the timing problems are exemplified through some simulated examples. The future research will concentrate on analysis of the robustness properties with respect to time-delay variations and jitter in sampled-data systems. The following items will be of great interest: 1) Studying ways of analyzing time-varying systems, in particular influences of jitter and time-varying delays. 2) Applicability of robustness theory to derive jitter specifications. 3) Ways of detecting and compensating for transient errors.

05/03/09

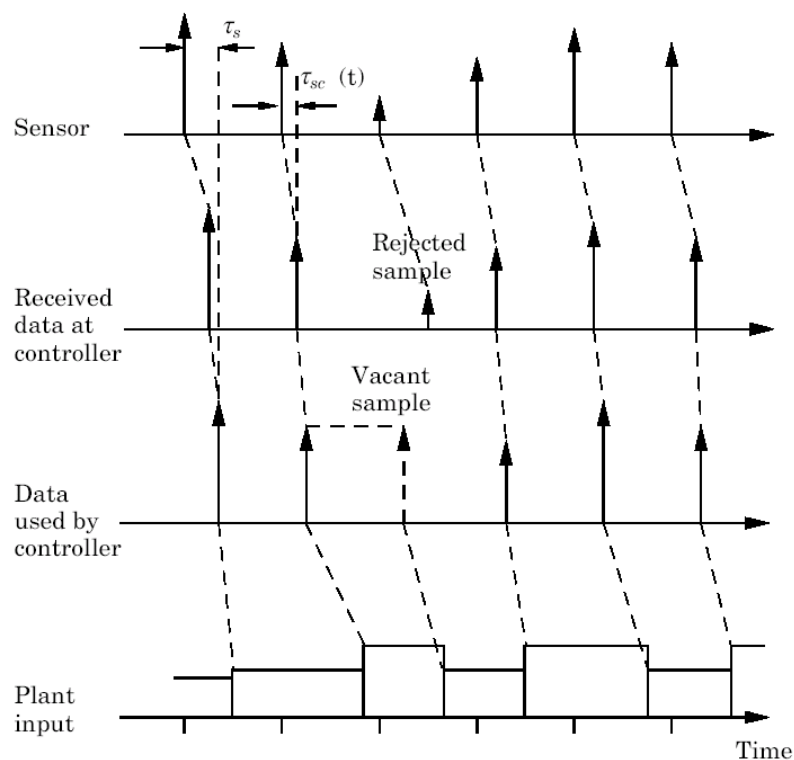
## ▪ Paper:

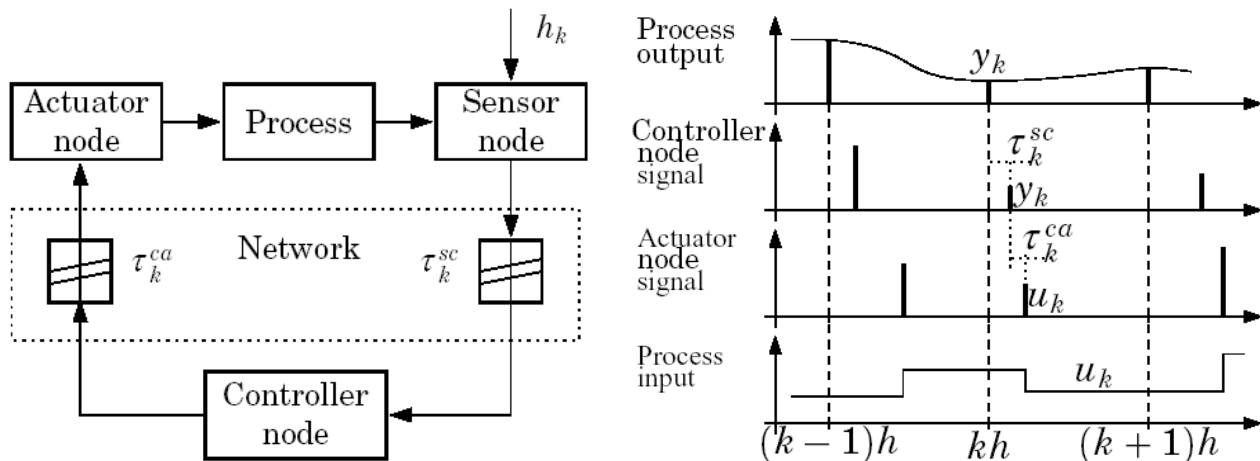
- J. Nilsson, B. Bernhardsson, B. Wittenmark,
- "Stochastic analysis and control of real-time systems with random time delays,"
- Automatica, 34(1):57-64, Jan. 1998.

## ▪ Abstract:

- The paper discusses modeling and analysis of **real-time systems** subject to **random time delays** in the **communication network**. A new method for analysis of different control schemes is presented. The method is used to **evaluate different suggested schemes** from the literature. A new scheme, using so called **timestamps**, for handling the random time delays is then developed and successfully compared with previous schemes. The new scheme is based on **stochastic control theory** and a **separation property** is shown to hold for the **optimal controller**.

05/03/09





$$x_{k+1} = \Phi x_k + \Gamma_0(\tau_k^{sc}, \tau_k^{ca})u_k + \Gamma_1(\tau_k^{sc}, \tau_k^{ca})u_{k-1} + v_k$$

Nilsson, Bernhardsson, Wittenmark 1998

## Types of Jitters

### ▪ Paper:

- P. Marti, J.M. Fuertes, G. Fohler, and K. Ramamritham,
- "Jitter compensation for real-time control systems,"
- In Proc. 22nd IEEE Real-Time Systems Symposium, pp. 39-48, Dec. 2001.

### ▪ Abstract:

- In this paper, we first identify the **potential violations of control assumptions** inherent in **standard real-time scheduling approaches** (because of the presence of **jitters**) that causes, **degradation** in control performance and may even lead to **instability**. We then develop practical approaches founded on **control theory** to deal with these violations. Our approach is based on the **notion of compensations** wherein **controller parameters** are adjusted **at runtime for the presence of jitters**. Through time and memory overhead analysis, and by elaborating on the implementation details, we characterize when **offline and on-line compensations** are feasible. Our experimental results confirm that our approach does compensate for the **degraded control performance** when **EDF** and **FPS** algorithms are used for scheduling the control tasks. Our compensation approach provides us another advantage that leads to **better schedulability of control tasks**. This derives from the potential to derive **more flexible timing constraints**, beyond periods and deadlines necessary to apply EDF and FPS. Overall, our approach provides guarantees offline that **the control system will be stable at runtime**-if temporal requirements are met at runtime-even when actual execution patterns are not known beforehand. With our approach, we can address the problems due to (a) **sampling jitters**, (b) **varying delays** between sampling and actuation, or (c) **both-not addressable** using traditional EDF and FPS based scheduling, or by previous real-time and control integration approaches.

# Types of Jitters

Case	Sketch <sup>1</sup>	S <sup>2</sup>	SA <sup>3</sup>	SP <sup>4</sup>
1		x	-	h_k
2		-	x	tau_k
3		x	x	h_k, tau_k
4		-	-	-
5		-	x	tau_k
6		-	-	-

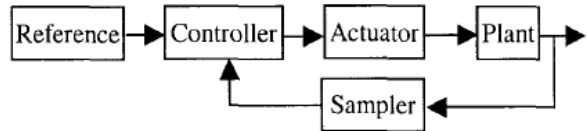
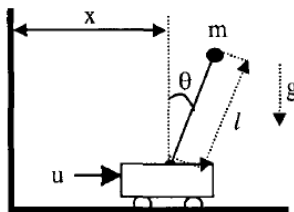


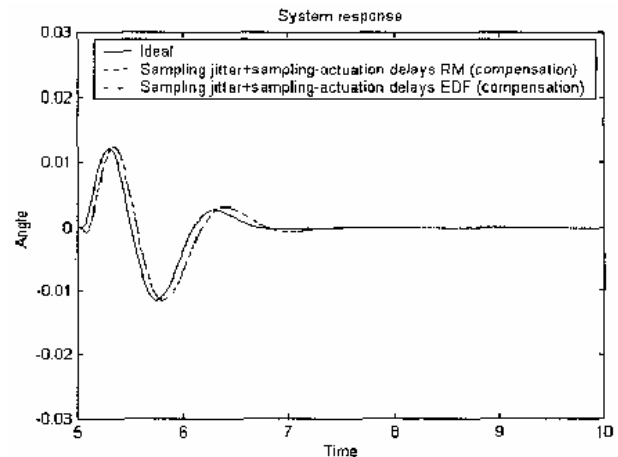
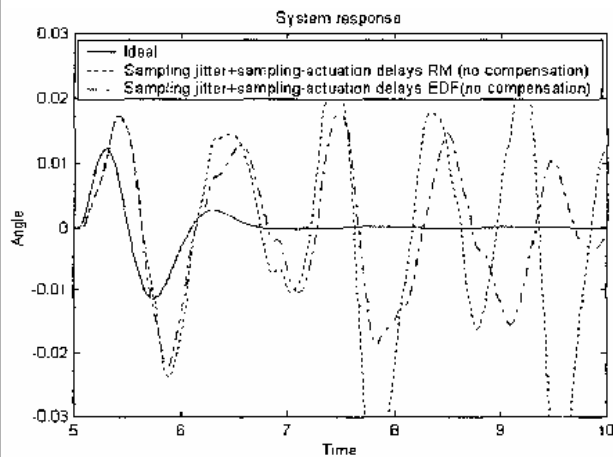
Figure 1. Control loop

<sup>1</sup>  $\Delta$ ,  $\square$  and  $\nabla$  denote sampling time, controller execution and actuation time  
<sup>2</sup> S stands for sampling jitter  
<sup>3</sup> SA stands for sampling-actuation delays  
<sup>4</sup> SP stands for source of problems

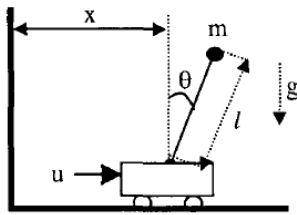
# Types of Jitters: Fixed Timing Constraints



	Task1	Task2	Control task
T	60ms	70ms	80ms
C	10ms	10ms	1ms







	T	C	DL	Offset
Task T1	100ms	60ms		
Task T2	200ms	20ms	20ms	
Control task Cr	$h_k$	20ms	20ms	40ms

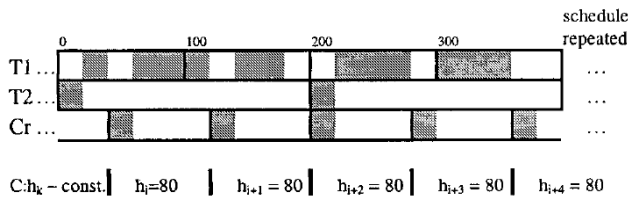


Figure 6. Not feasible schedule

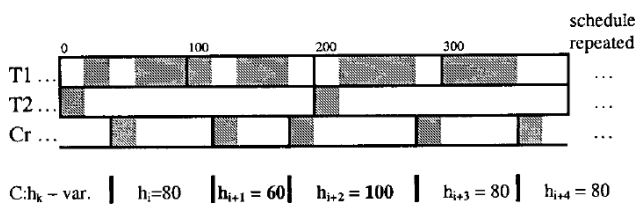
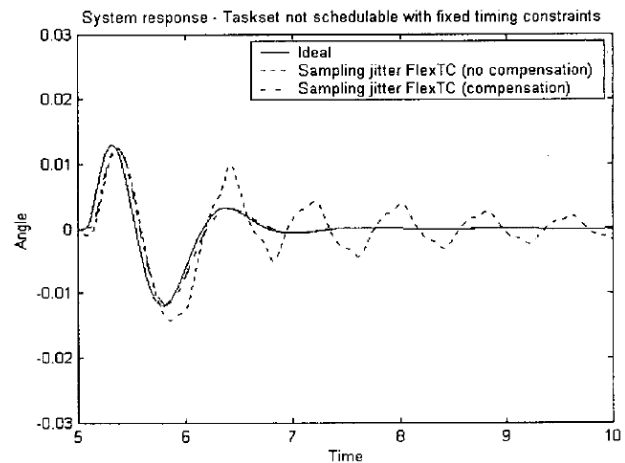


Figure 7. Feasible schedule



03/21/05

Marti et al. 2001

## Timing Requirements & Control Attributes

### ■ Paper:

- I. Bate, P. Nightingale, and A. Cervin,
- "Establishing timing requirements and control attributes for control loops in real-time systems,"
- Proc. 15th Euromicro Conf. on Real-Time Systems, pp. 121-128, July 2003.

### ■ Abstract:

- Advances in scheduling theory have given designers of control systems greater flexibility over their choice of timing requirements. This could lead to systems becoming more responsive, more flexible and more maintainable. However, experience has shown that engineers find it difficult to exploit these advantages due to the difficulty in determining the "real" timing requirements of systems and therefore the techniques have delivered less benefit than expected. Part of the reason for this is that the models used by engineers when developing systems do not allow for emergent properties such as timing. This paper presents an approach and framework for addressing the problem of identifying an appropriate and valid set of timing requirements and their corresponding control parameters based on a combination of static analysis and simulation.

05/03/09

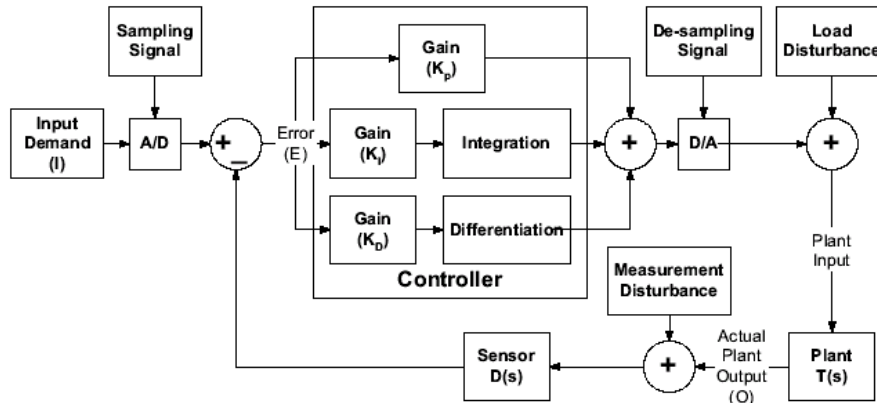
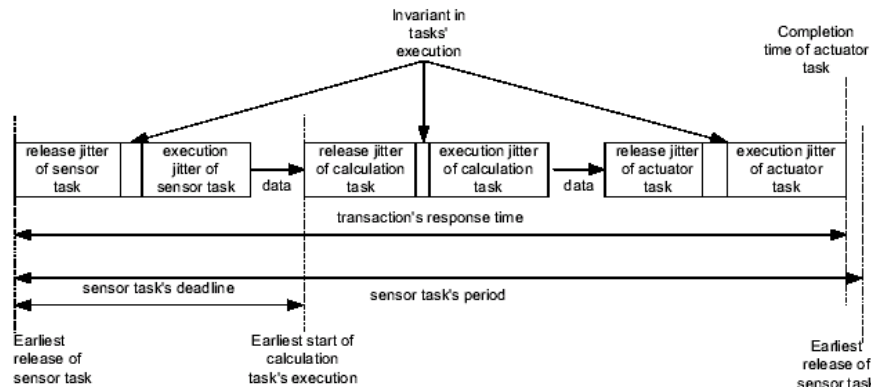


Figure 1 – Typical PID Loop



Bate, Nightingale & Cervin 2003

03/16/04

## End-to-End Delay of Videoconferencing

### ■ Paper:

- M. Baldi and Y. Ofek,
- "End-to-end delay analysis of videoconferencing over packet-switched networks,"
- IEEE/ACM Transactions on Networking, 8(4): 479-492, Aug. 2000.

### ■ Abstract (short):

- In order for the participants in a **videoconference call** to interact naturally, the **end-to-end delay** should be **below human perception**; even though an objective and unique figure cannot be set, **100 ms** is widely recognized as the **desired one-way delay** requirement for interaction. Since the global propagation delay can be about 100 ms, the **actual end-to-end delay budget** available to the system designer (excluding propagation delay) can be no **more than 10 ms**. We identify the components of the end-to-end delay in various configurations with the objective of understanding how it can be kept **below the desired 10-ms bound**. We analyze **these components** step-by-step through **six system configurations** obtained by combining **three generic network architectures** with **two video encoding schemes**. We study the transmission of **raw video** and **variable bit rate (VBR) MPEG video encoding** over 1) **circuit switching**; 2) **synchronous packet switching**; and 3) **asynchronous packet switching**. In addition, we show that **constant bit rate (CBR) MPEG encoding delivers unacceptable delay**—on the order of the **group of pictures (GOP) time interval**—when **maximizing quality** for static scenes.

05/03/09

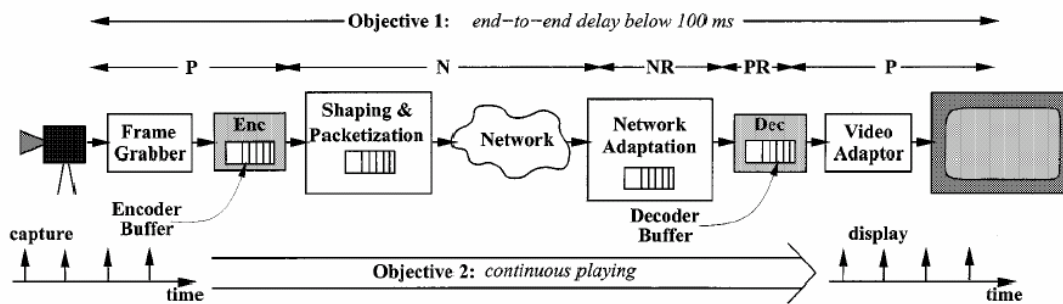
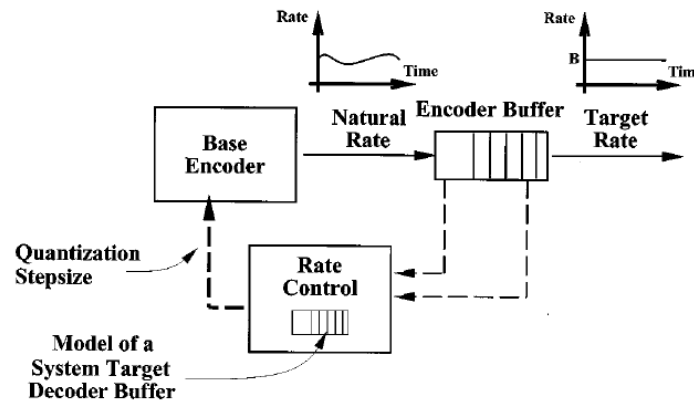


Fig. 1. Model of a videoconferencing system. P: processing delay. PR: processing resynchronization delay. N: network delay. NR: network resynchronization delay.

$$P + N + PR + NR + Pr = \text{CONSTANT} \leq 100 \text{ ms.}$$



Baldi & Ofek 2000

03/16/04

## Timing Analysis for Programs

### ■ Paper:

- S.-S. Lim, J. Kim, and S.L. Min,
- "A worst case timing analysis technique for optimized programs,"
- In Proc. Fifth Int'l Conf. Real-Time Computing Systems and Applications, pp. 151-157, Oct. 1998.

### ■ Abstract:

- We propose a technique to analyze the **worst case execution times (WCETs)** of **optimized programs**. Our work is based on a **hierarchical timing analysis** technique called the **extended timing schema (ETS)**. A major hurdle in applying the ETS to optimized programs is the **lack of correspondences** in the **control structure** between the **optimized machine code** to be analyzed and the **original source program** written in a high-level programming language. We suggest a **compiler-assisted approach** where a **timing analyzer** relies on an optimizing compiler for a **consistent hierarchical representation** and an **accurate source-level correspondence** that are essential for accurate WCET analysis for optimized programs. In order to validate the proposed approach, we implemented a **proof-of-concept version** of a timing analyzer for a **256-bit VLIW processor** and compared the analysis results with the simulation results. The experimental results show that the proposed solution can accurately predict the WCETs of highly-optimized VLIW programs.

05/03/09

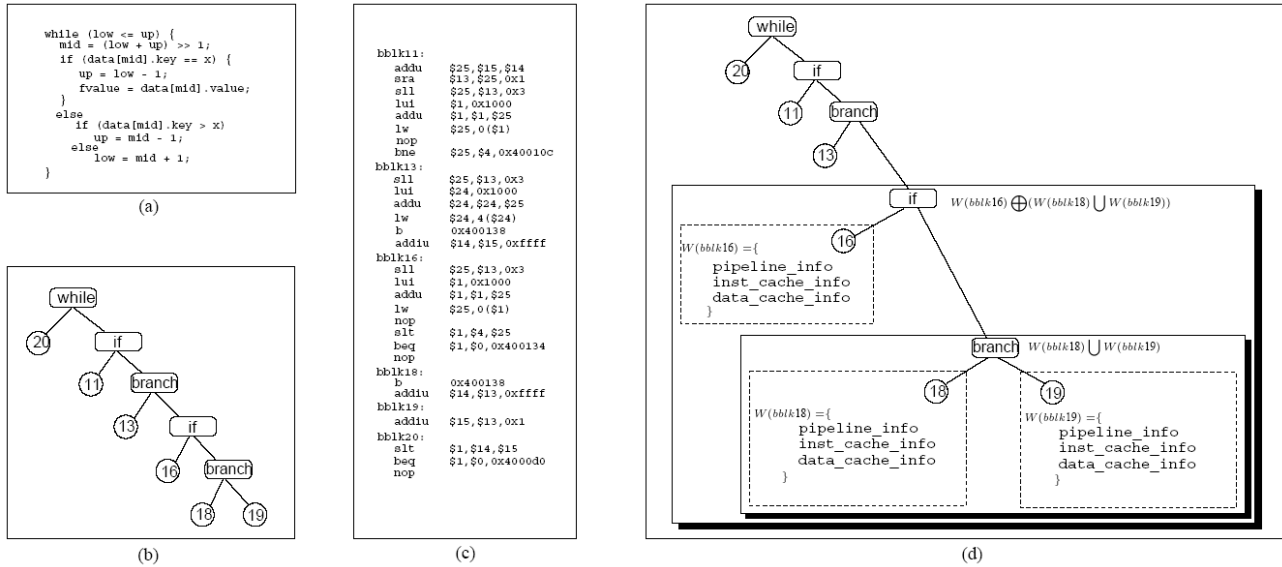


Figure 1. Overall processing steps in the ETS: (a) a sample high-level program, (b) the syntax tree, (c) the assembly code, and (d) the steps for applying the ETS to the sample program.

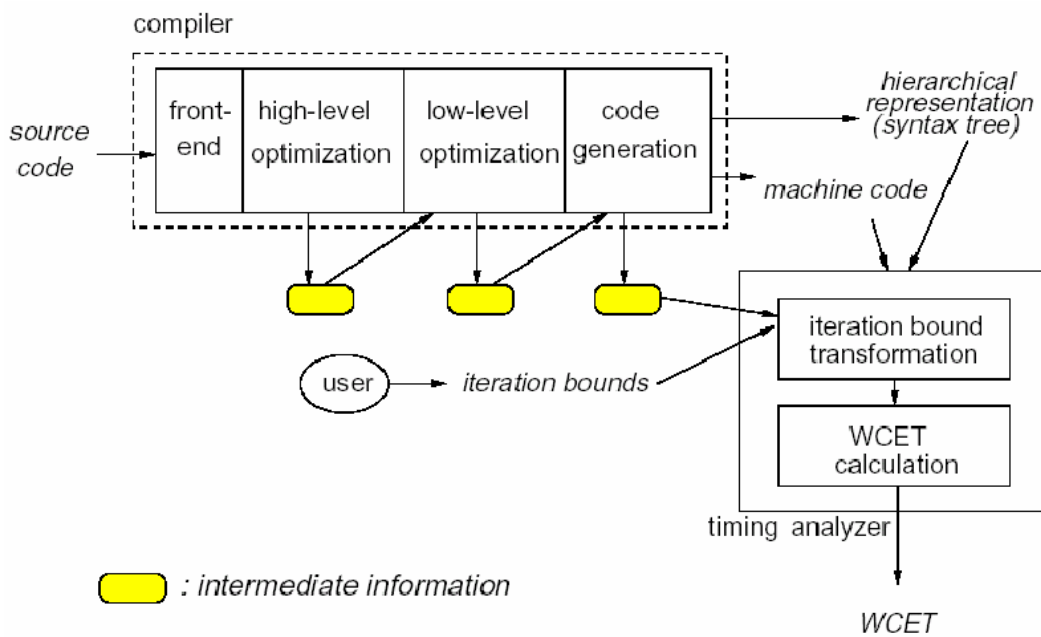


Figure 2. Overview of a compiler-assisted timing analyzer.

Benchmark Programs	Description
<i>MatMul</i>	multiplies two $5 \times 5$ integer matrices
<i>JFDCTINT</i>	performs the forward Discrete Cosine Transform used in JPEG
<i>FIR</i>	performs a 32-taps Finite Impulse Response (FIR) filtering operation
<i>FFT</i>	performs the Fast Fourier Transform (FFT) on 256 floating point numbers

**Table 1. The benchmarks used for the experiments.**

Benchmark Programs	Simulation Results	Analysis Results
<i>MatMul</i>	1673	1739
<i>JFDCTINT</i>	4456	4780
<i>FIR</i>	30940	32218
<i>FFT</i>	2879360	4567872

TMS320C6201

**Table 2. Predicted and measured execution cycles of the benchmark programs.**

## Multiprocessor Implementation of Digital Engine Control

- **Paper:**
  - P.L. Shaffer,
  - "A multiprocessor implementation of real-time control for a turbojet engine,"
  - IEEE Control Systems Magazine, 10(4): 38-42, June 1990.
- **Abstract:**
  - A real-time control program for a turbojet engine has been implemented on a four-processor computer, achieving a speedup of 3.38 times the speed of a sequential version of the same program on a single processor. The concurrent program was produced from a sequential program by subjecting the sequential program to global, hierarchical interprocedural data-flow analysis and timing measurements. A static schedule for the constituent tasks of the control program on the four processors was determined using a heuristic algorithm based on the critical-path method. The approach should be applicable to a variety of control and related programs where iterative tasks with well-bounded execution times are computed in systems with hard real-time requirements.

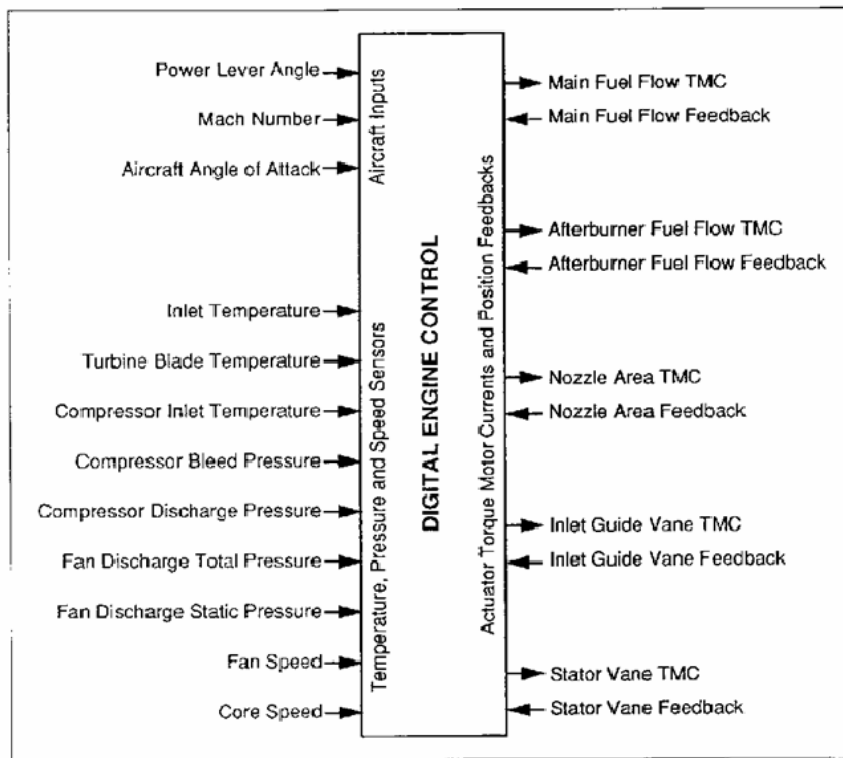


Fig. 1. Digital engine control for a turbojet engine, showing inputs from aircraft, sensor inputs from engine, and actuator torque motor concurrent (TMC) outputs and position feedbacks.

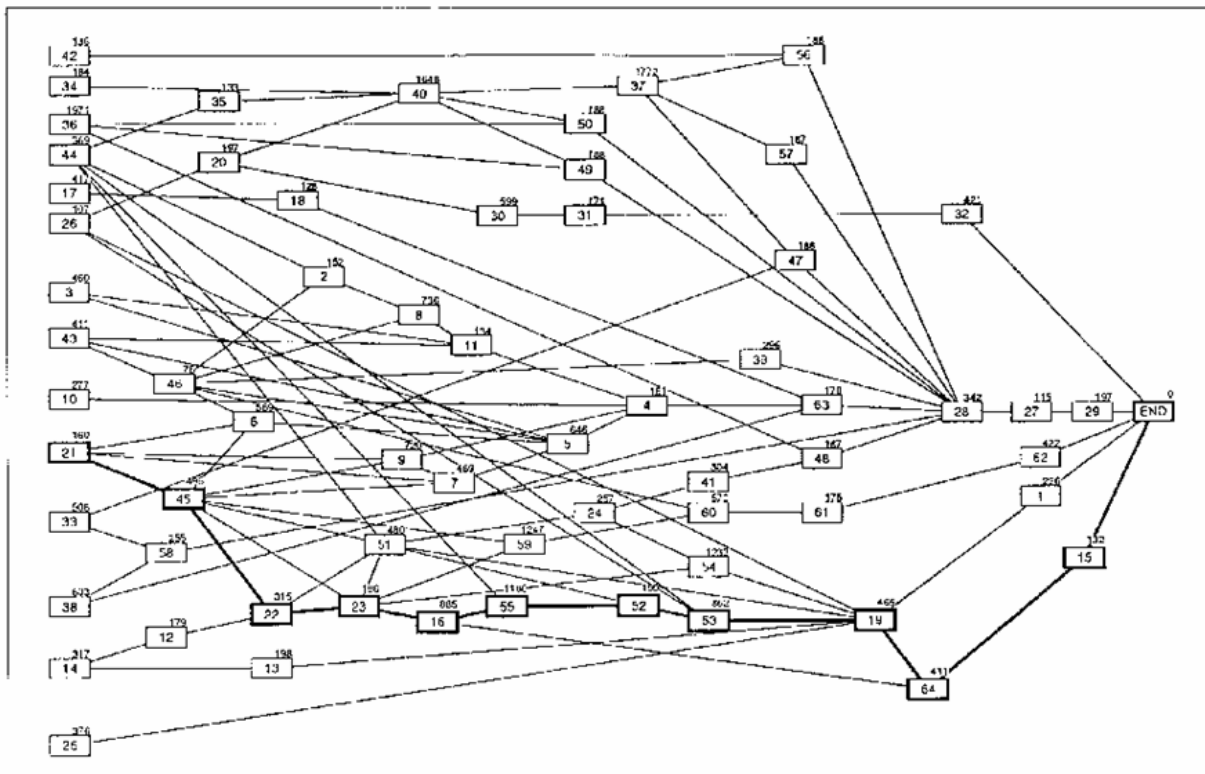


Fig. 2. Computation graph for control program. Each node represents a procedure, and is labeled (on the upper right corner) with the maximum execution time (µs). Graph edges represent data dependencies between routines (data flow is from left to right). External inputs and outputs are not shown.

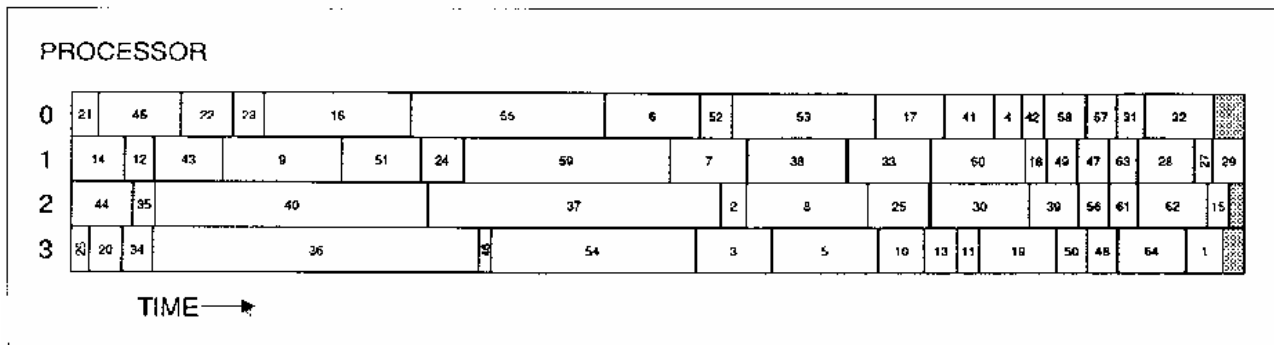


Fig. 3. A possible schedule for the control program on four processors, representing a speedup of 3.94. The shaded areas represent idle processor time.

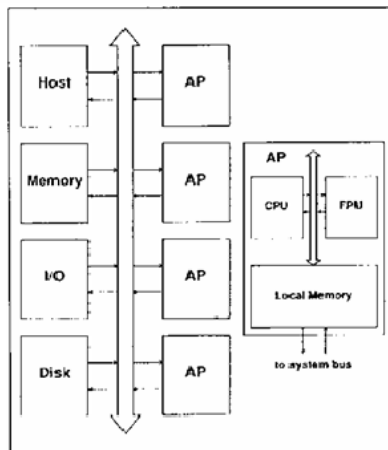


Fig. 4. The multiprocessor architecture used to implement the turbojet engine control program. Each application processor (AP) contains a 68020 central processor unit (CPU) and 68881 floating-point unit (FPU), as well as local RAM. The I/O (input/output) board is an Ethernet controller.

Shaffer 1990

03/16/04

## Temporal Characteristics of Task Transmission

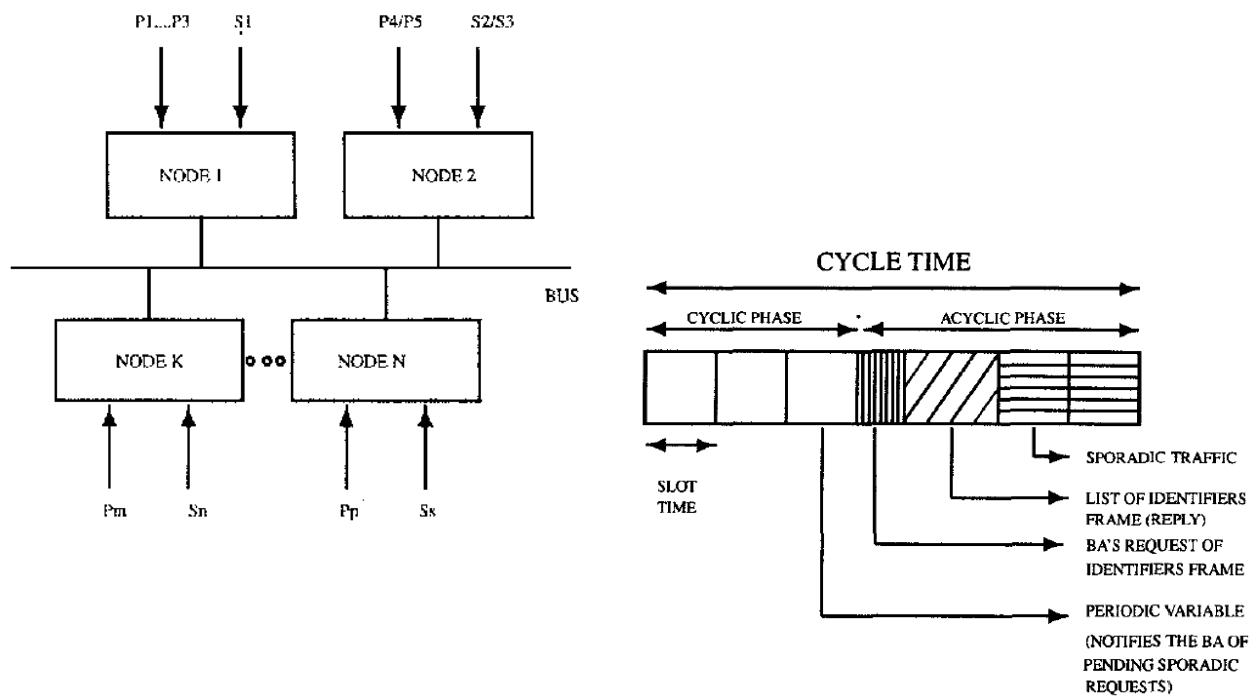
### ■ Paper:

- P. Pedro and A. Burns,
- "Worst case response time analysis of hard real-time sporadic traffic in FIP networks,"
- Proc. Ninth Euromicro Workshop on Real-Time Systems, pp. 3-10, June 1997

### ■ Abstract:

- Real-time fieldbuses are currently a significant issue in both process control and manufacturing areas. They constitute the base upon which real-time fault-tolerant distributed systems can be designed for these application areas. A potential large leap towards the use of Fieldbus in such time-critical applications lies in the evaluation of its temporal behaviour. In particular an important problem associated with the Fieldbus FIP is its inability to guarantee the timing performance of sporadic traffic. In this paper we develop the pre-run-time schedulability analysis of FIP bounding the worst case response time of the sporadic traffic

05/03/09



**Figure 3. Slot time in a micro-cycle**

▪ **Paper:**

- S. Saad-Bouzefrane, and F. Cottet,
- "A performance analysis of distributed hard-real time applications,"
- Proc. IEEE Int'l Workshop on Factory Communication Systems, pp. 167-176, Oct. 1997

▪ **Abstract:**

- In distributed hard real-time applications there is a need for temporal analysis to evaluate and optimise a design with respect to the deadlines. The key method is the scheduling analysis of such applications, which means the schedulability not only of its tasks but of its messages too. The authors present a schedulability analysis which allows one to determine the timing parameters of messages and to update those of tasks. Given an initial task set characterised by temporal attributes and network interactions, this methodology permits one to produce valid execution sequences and to evaluate the relevant timing factors. Simulation results done with CAN and FDDI protocols are deduced.



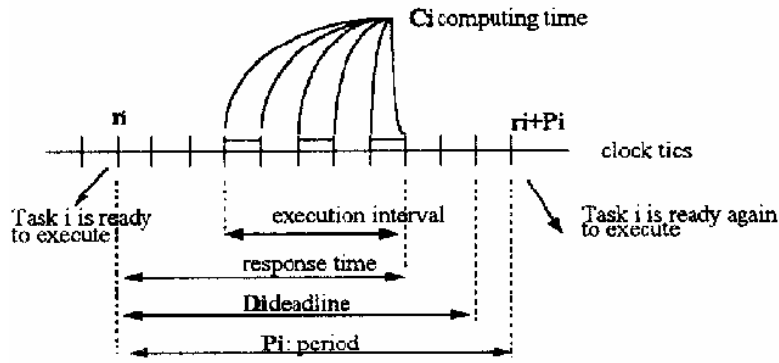


Figure 1: Temporal characteristics of a task.

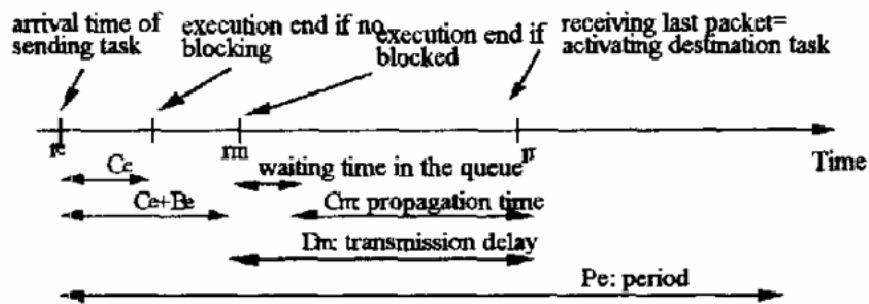


Figure 5: end to end communication

Table1: Timing characteristics of the application tasks

Task	node	C	P=D	P <sub>r</sub>
A <sub>1</sub>	N <sub>1</sub>	200	10000	3
A <sub>2</sub>	N <sub>1</sub>	300	10000	2
A <sub>3</sub>	N <sub>1</sub>	100	20000	1
A <sub>4</sub>	N <sub>1</sub>	200	5000	4
B <sub>1</sub>	N <sub>2</sub>	300	10000	2
B <sub>2</sub>	N <sub>2</sub>	400	10000	1
B <sub>3</sub>	N <sub>2</sub>	200	5000	3
B <sub>4</sub>	N <sub>2</sub>	100	15000	1
C <sub>1</sub>	N <sub>3</sub>	300	20000	2
C <sub>2</sub>	N <sub>3</sub>	150	15000	3
C <sub>3</sub>	N <sub>3</sub>	200	10000	4

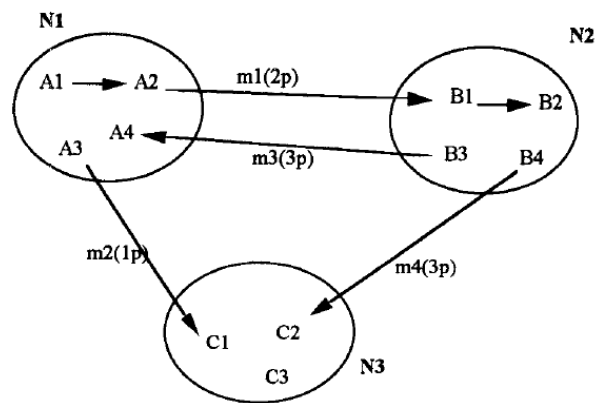


Table2: The timing attributes of messages

message	CAN			FDDI		
	C	D	P	C	D	P
m <sub>1</sub>	26	260	10000	726	3726	10000
0						
m <sub>2</sub>	13	650	20000	363	6726	20000
0						
m <sub>3</sub>	39	780	5000	1089	3726	5000
0						
m <sub>4</sub>	39	1820	15000	1089	12726	15000
0						

message	m <sub>1</sub>	m <sub>2</sub>	m <sub>3</sub>	m <sub>4</sub>
r <sub>m</sub>	900	1900	200	2100

