Spring 2021

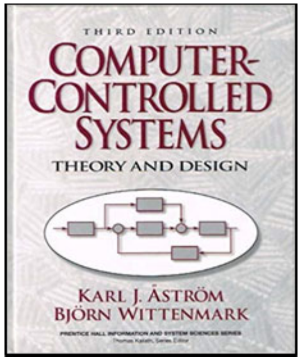
數位控制系統
Digital Control Systems

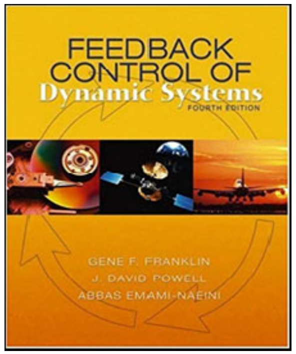
DCS-15
Timing Analysis

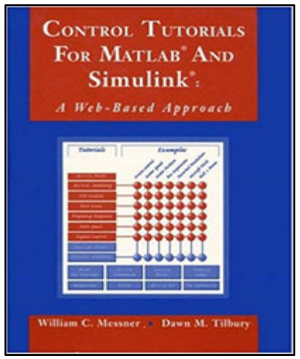
Feng-Li Lian

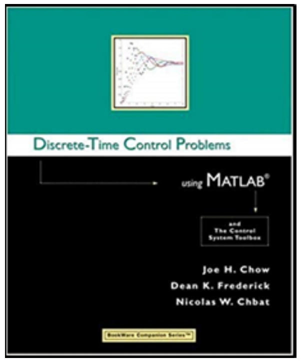
NTU-EE

Feb – Jun, 2021





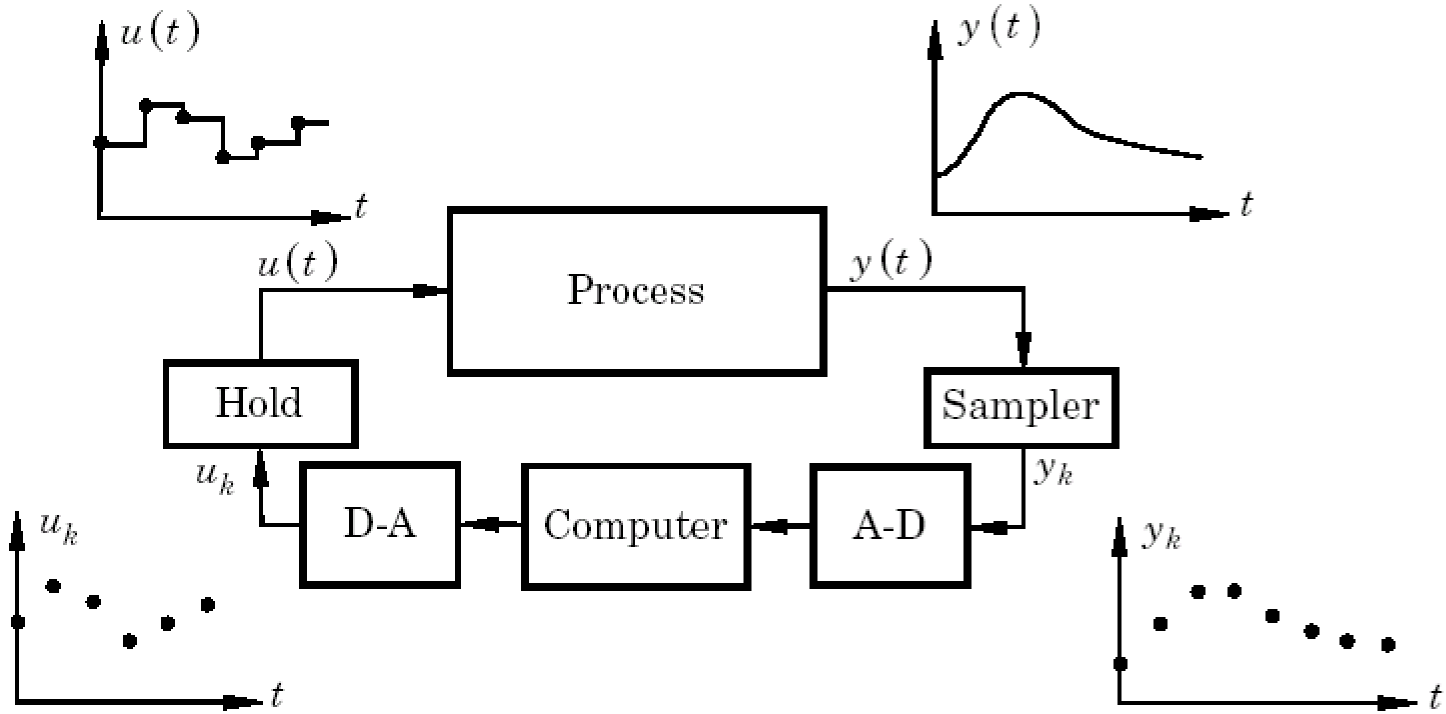




Outline

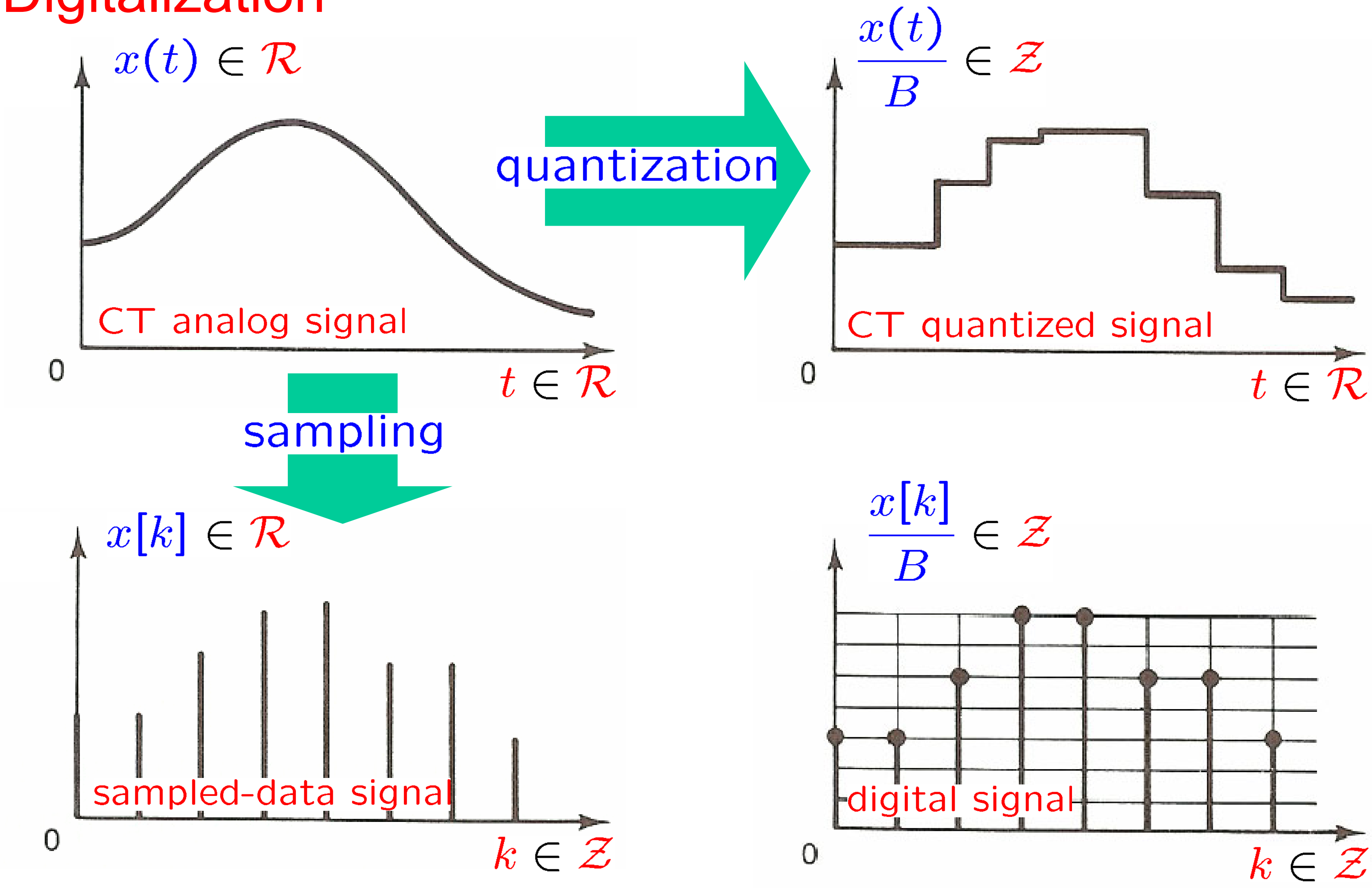
- Digitalization
- Timing Analysis in a Control System
- Some Related Issues:
  - Multiple or Random Time-Delay Systems
  - Types of Jitters
  - Timing Requirements & Control Attributes
  - Multiprocessor Implementation of Digital Engine Control
  - End-to-End Delay of Videoconferencing
  - Timing Analysis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

▪ Control System Block Diagram



Astrom & Wittenmark 1997

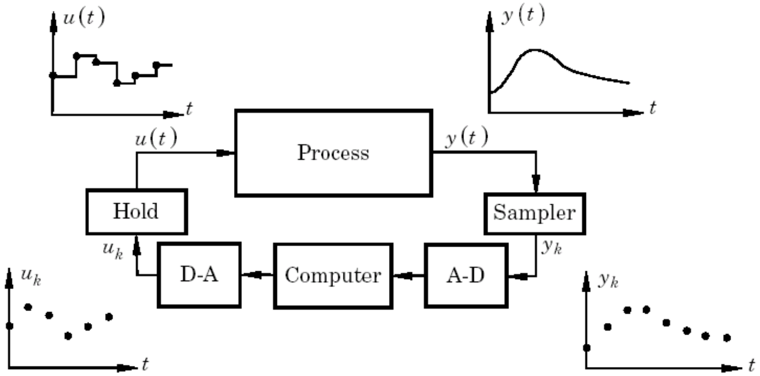
- Digitalization

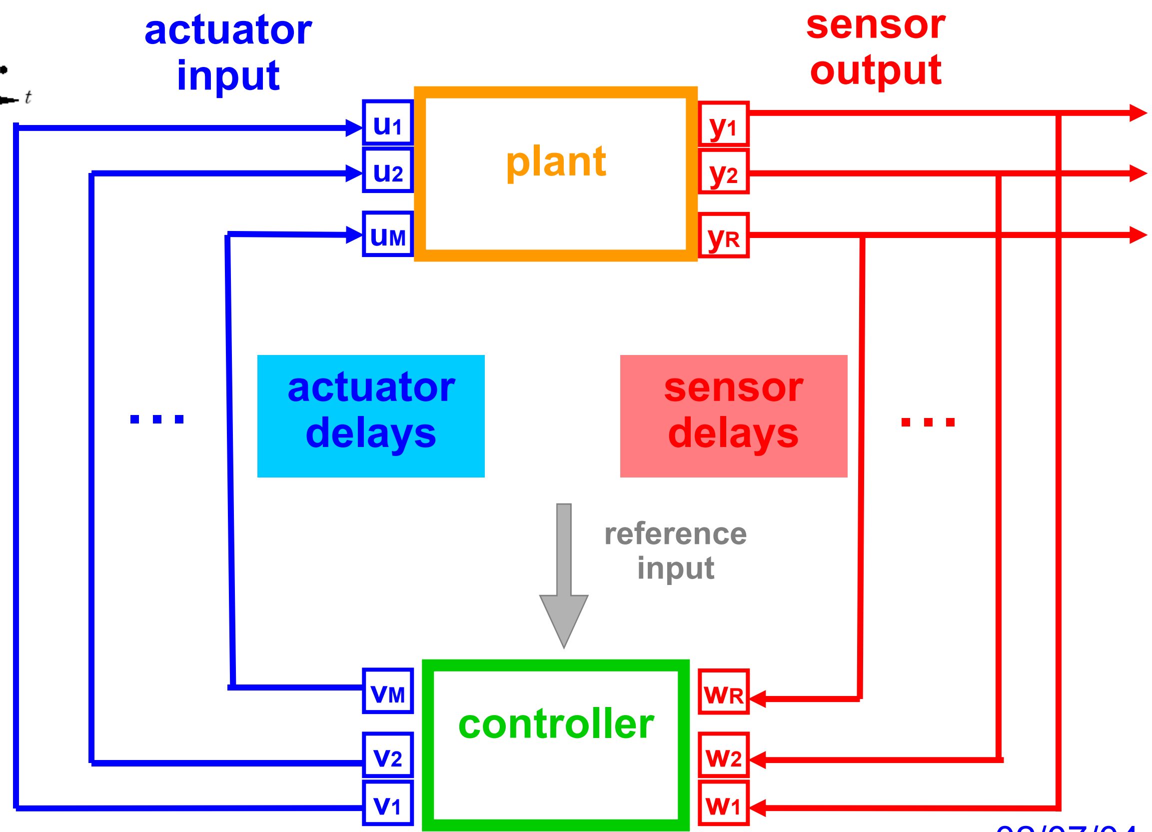


Ogata 1995

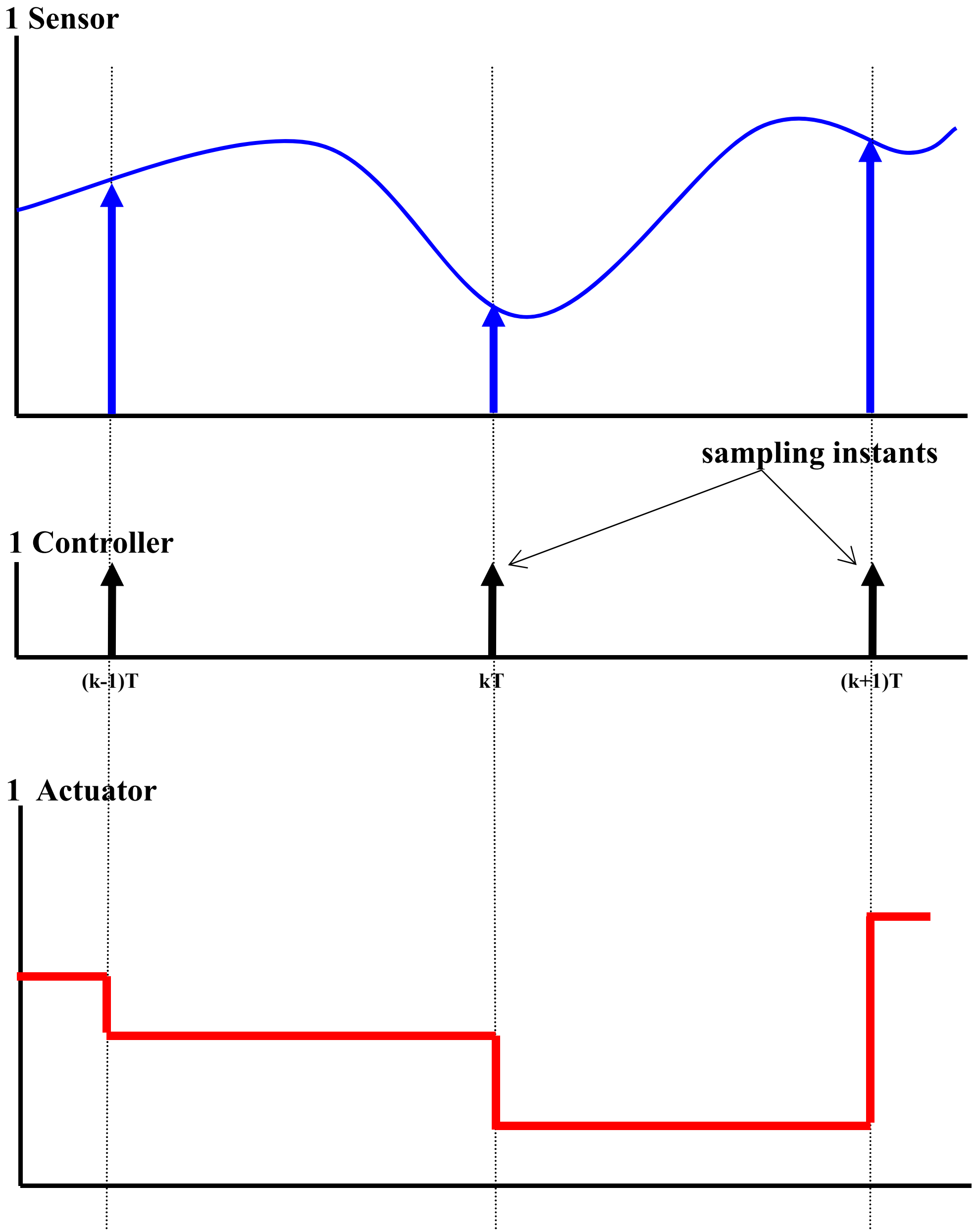
Timing Analysis: SISO and MIMO and Delays

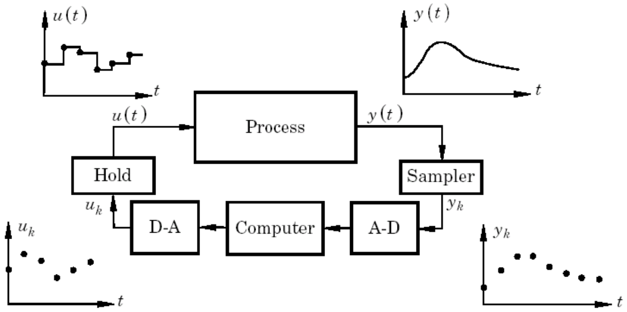
- Single-Input-Single-Output and Multiple-Input-Multiple-Output
- Single Delay and Multiple Delays



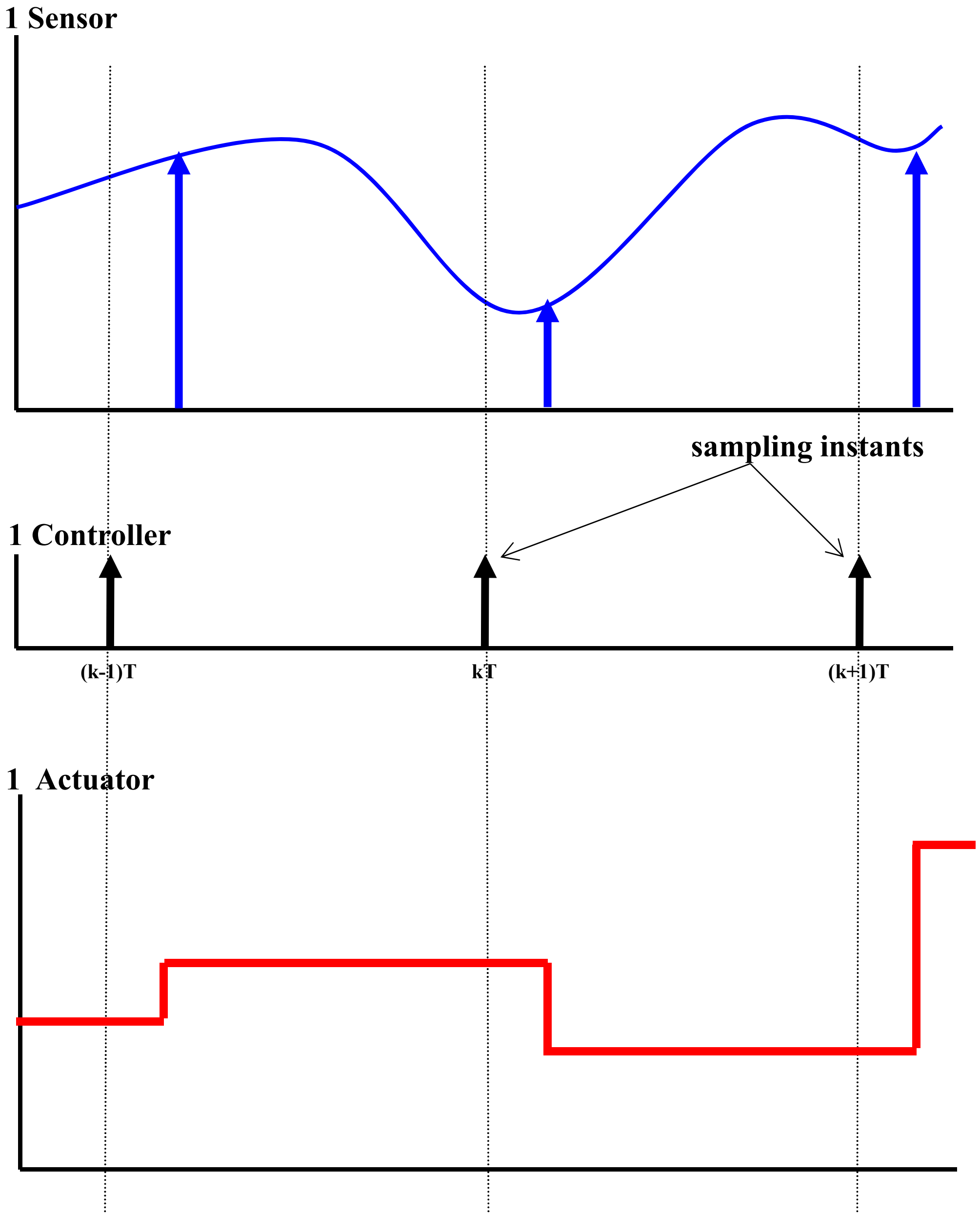


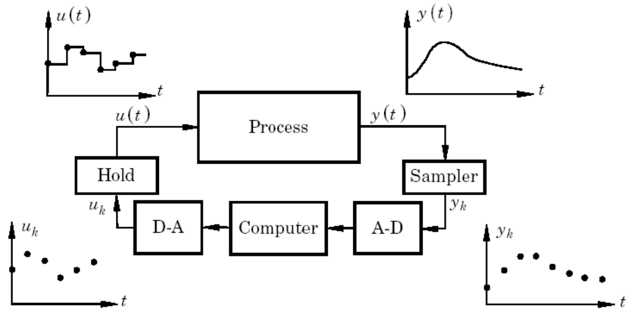
Timing Analysis: Single-Input-Single-Output



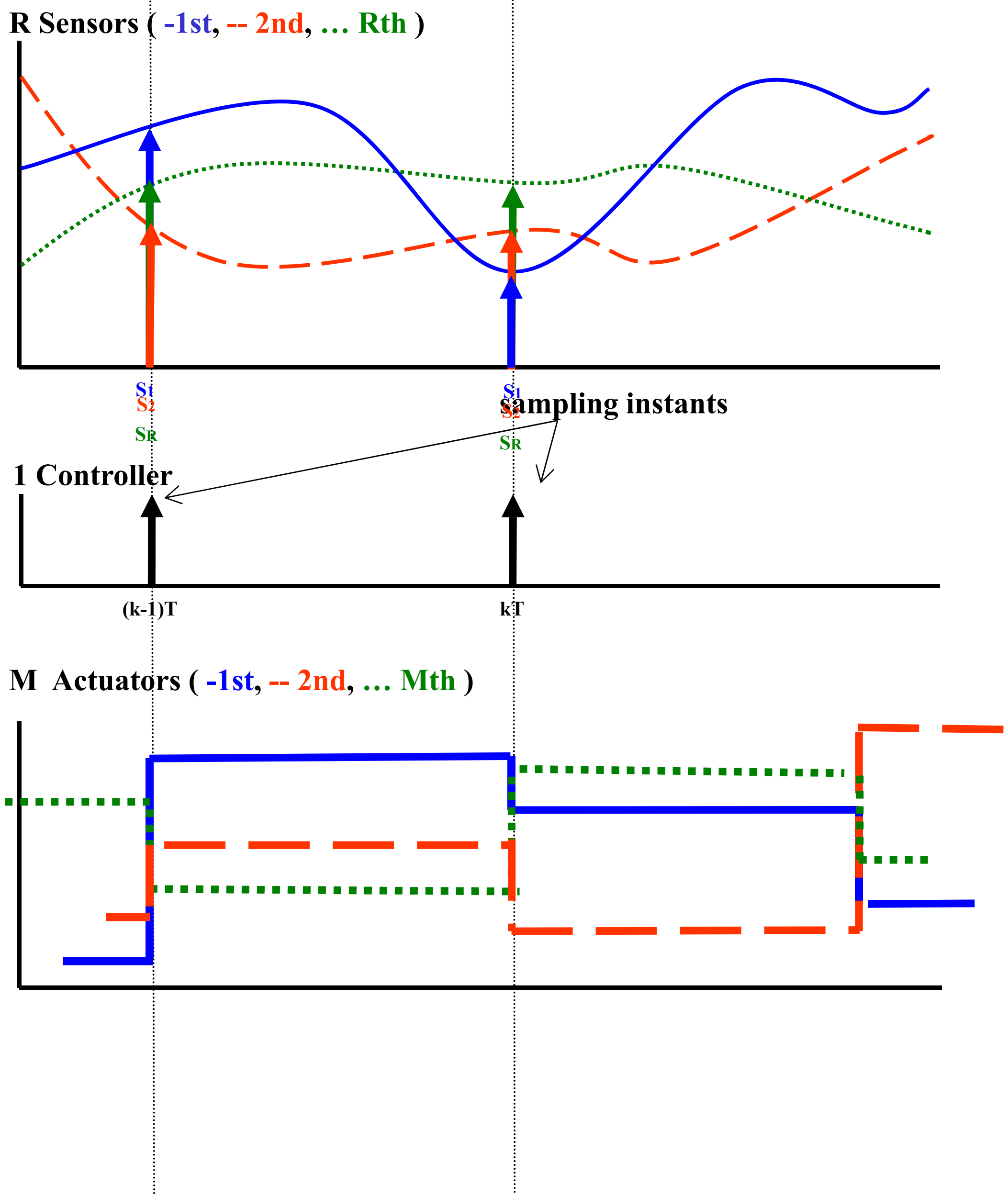


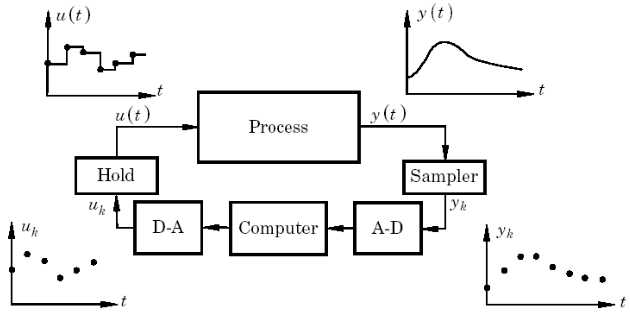
Timing Analysis: Single-Input-Single-Output with Single Delay

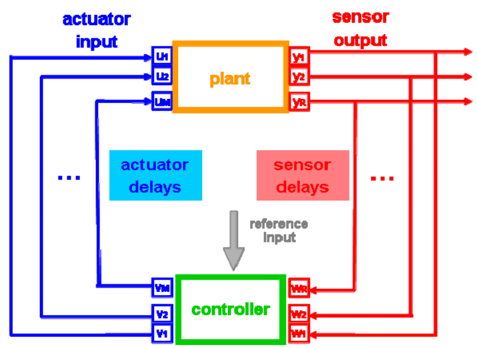




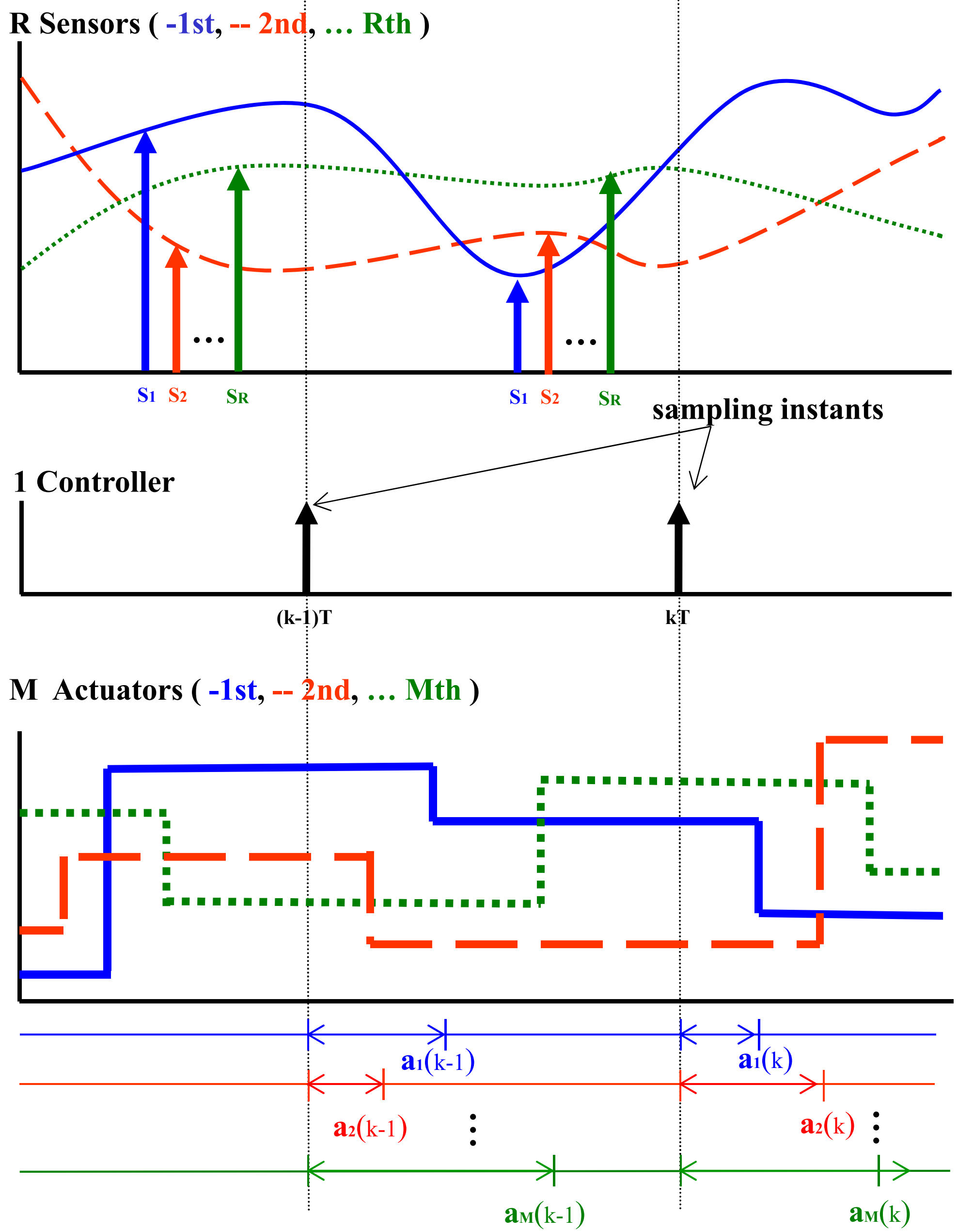
Timing Analysis: Multiple-Input-Multiple-Output

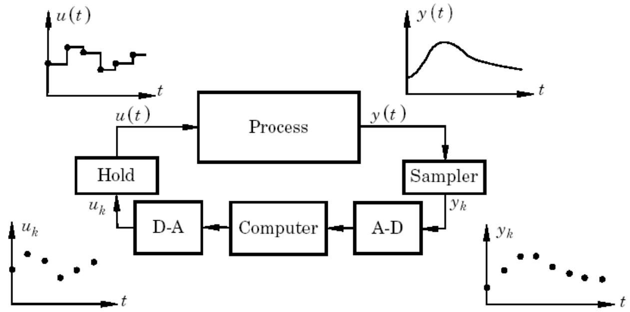


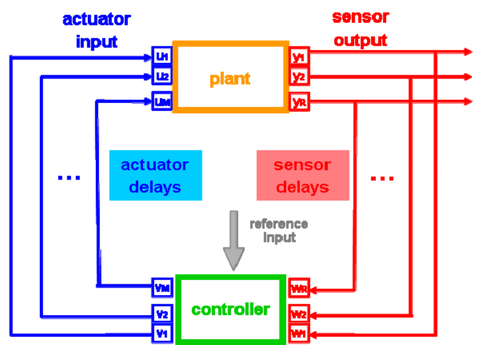


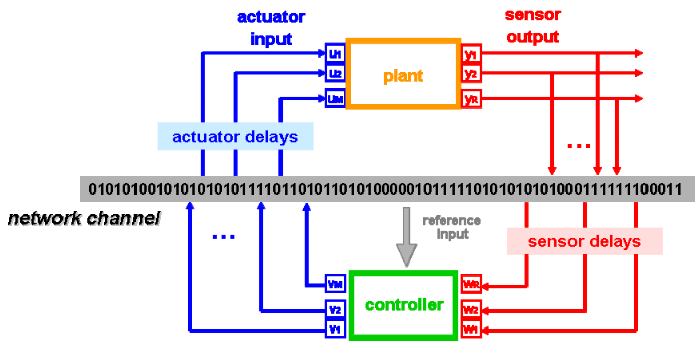


Timing Analysis: Multiple-Input-Multiple-Output with Delays









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.

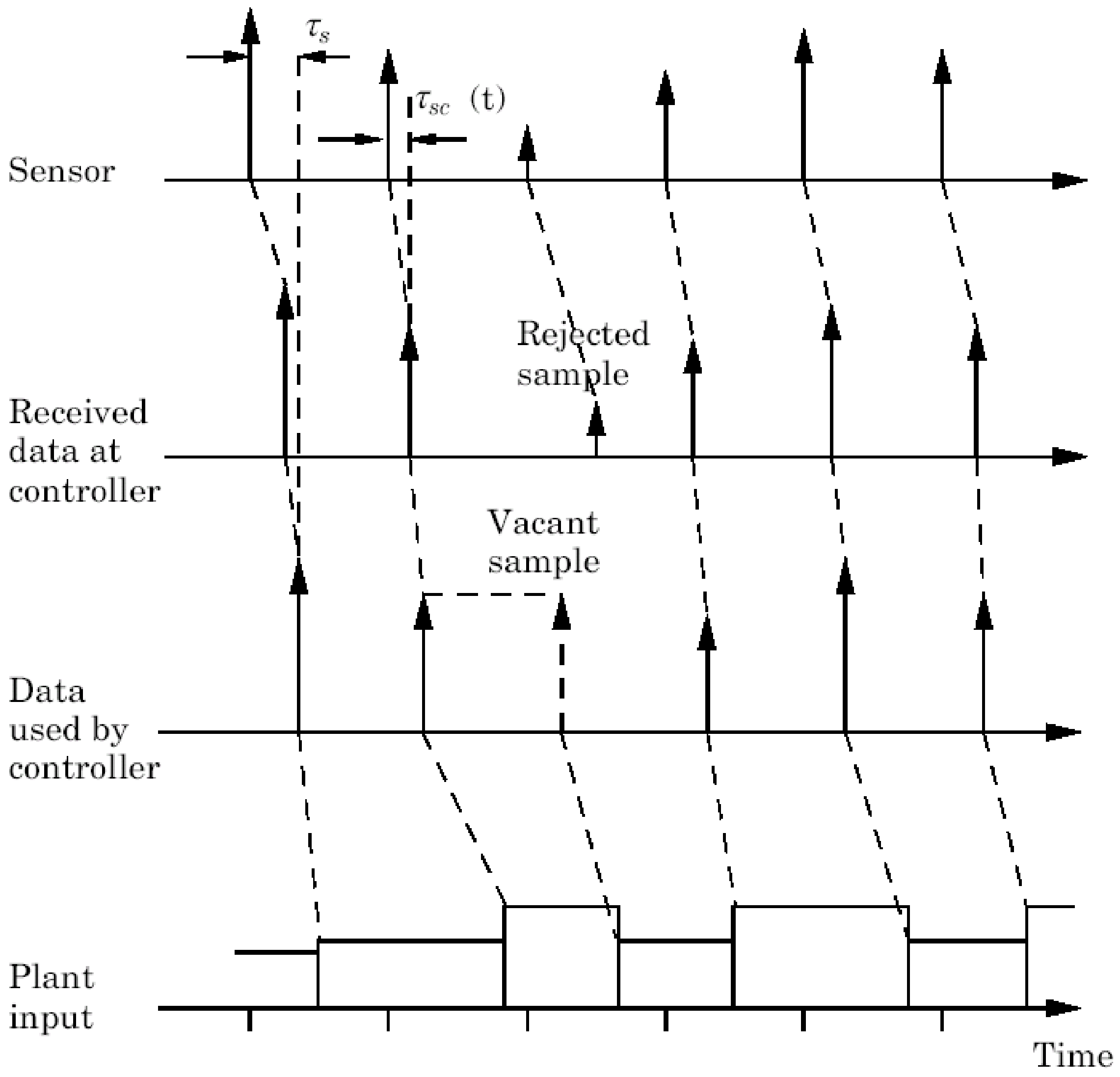
- Abstract:
  - In this paper we have discussed some of the timing problems in real-time control systems. The influence of the scheduling on the models is discussed together with different interesting problem formulations. The effect of the timing problems are exemplified through some simulated examples. 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.

Multiple or Random Time-Delay Systems

- 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.

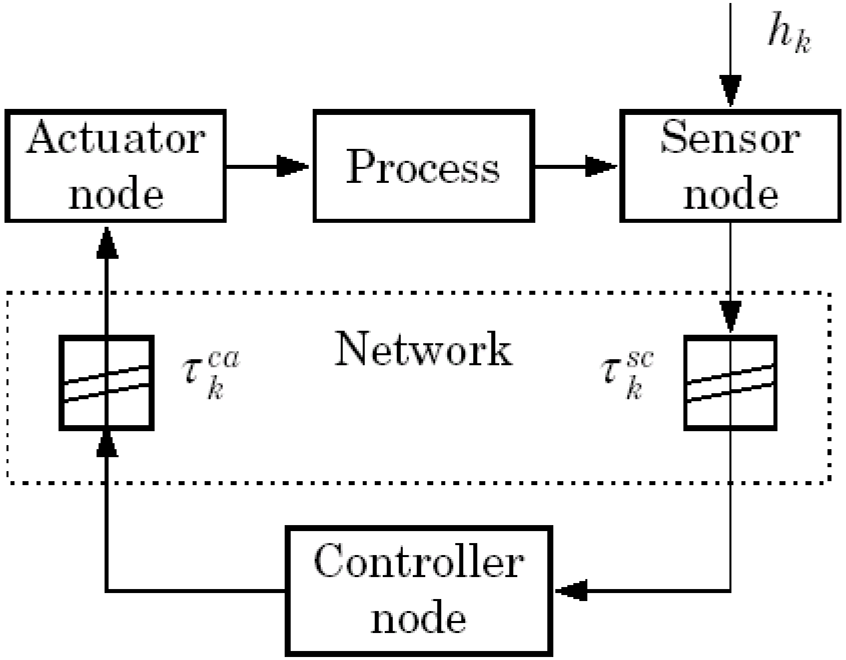
- Abstract:
  - The paper discusses modeling and analysis of real-time systems subject to random time delays in the communication network. A new method for analysis of different control schemes is presented. The method is used to evaluate different suggested schemes from the literature. A new scheme, 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.

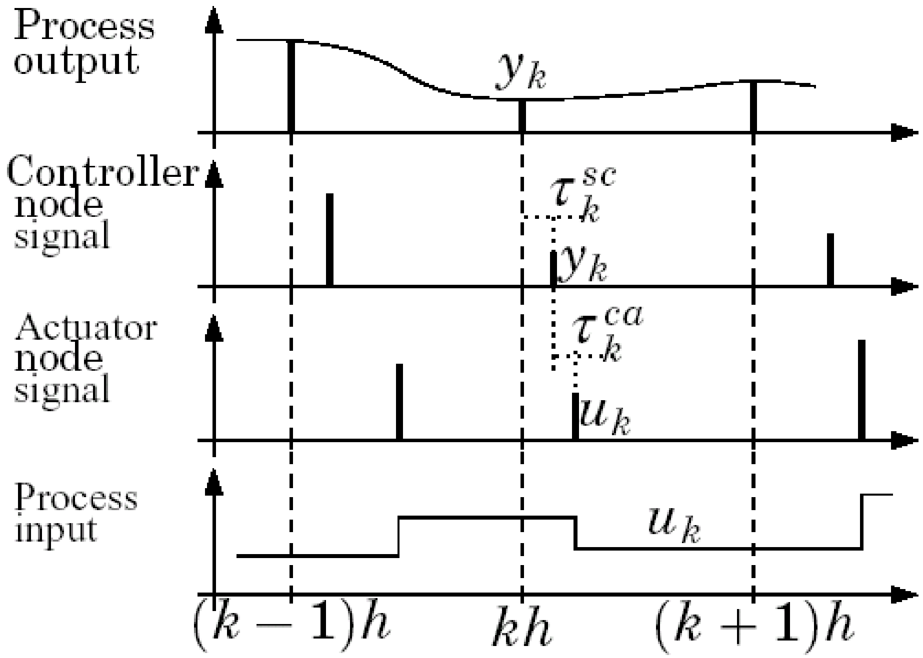
Multiple or Random Time-Delay Systems



Wittenmark, Nilsson, Torngren 1995

Multiple or Random Time-Delay Systems





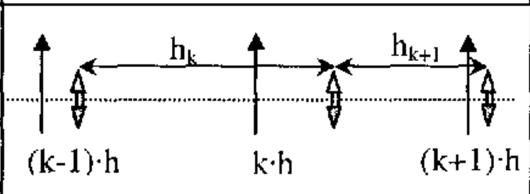
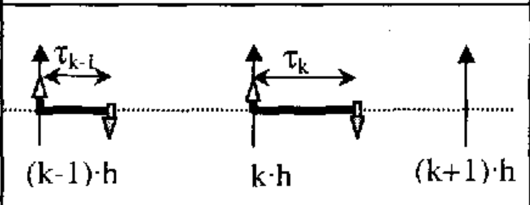
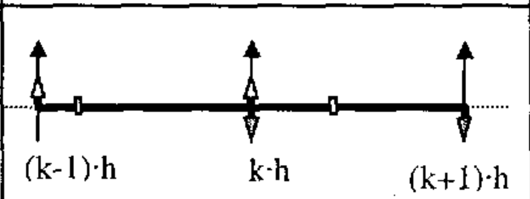
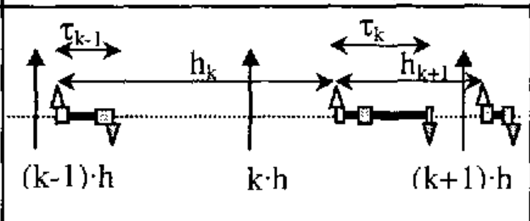
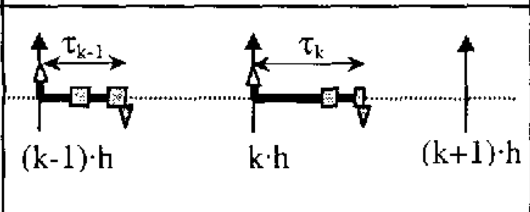
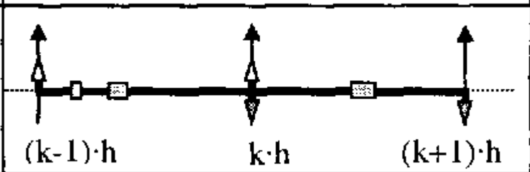
$$x_{k+1} = \Phi x_k + \Gamma_0(\tau_k^{sc}, \tau_k^{ca}) u_k + \Gamma_1(\tau_k^{sc}, \tau_k^{ca}) u_{k-1} + v_k$$

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[1]	S[2]	SA[3]	SP[4]
1		x	-	h_k
2		-	x	τ_k
3		-	-	-
4		x	x	h_k, τ_k
5		-	x	τ_k
6		-	-	-

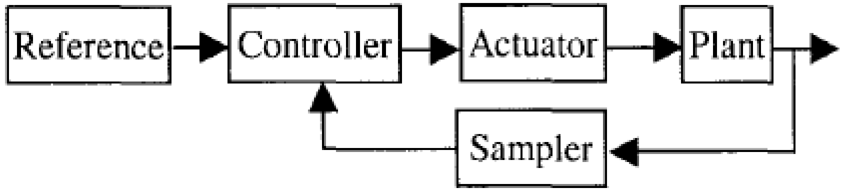
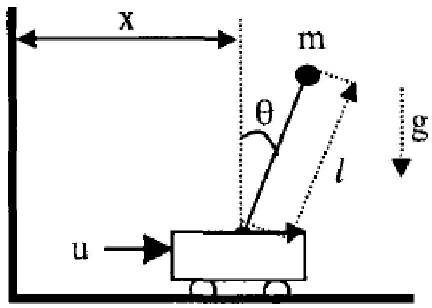


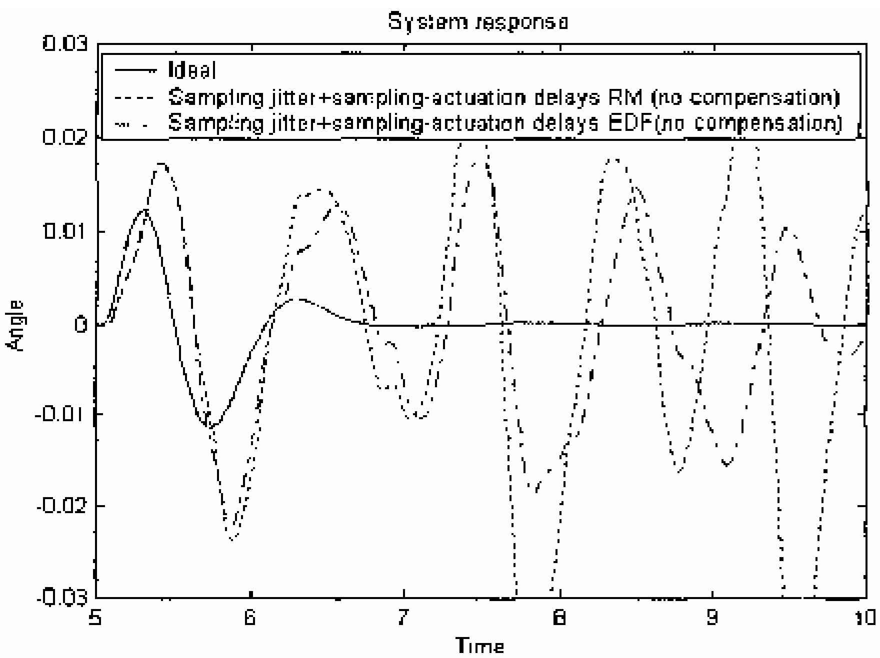
Figure 1. Control loop

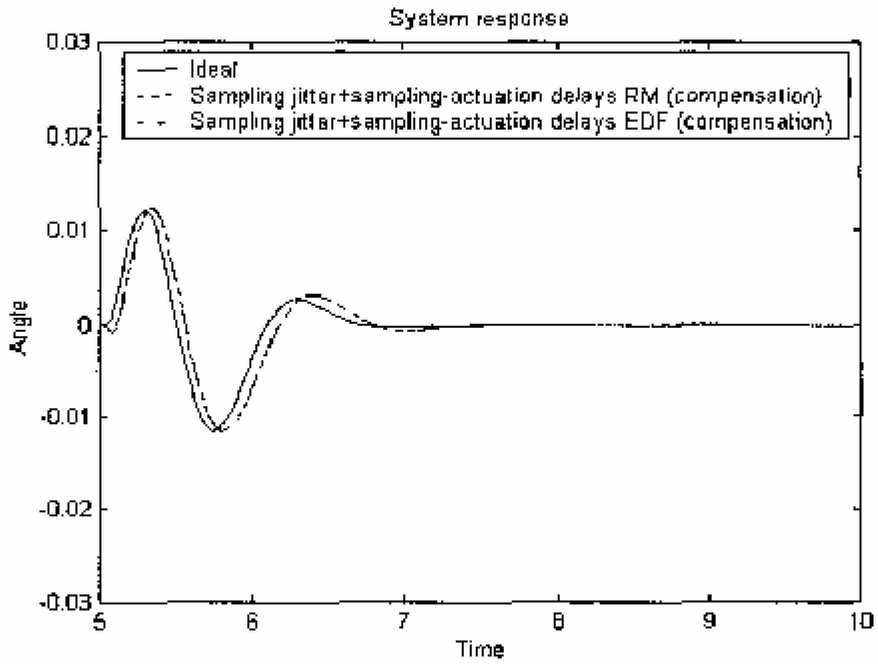
[1] Δ, ⊡ and ∇ denote sampling time, controller execution and actuation time
[2] S stands for sampling jitter
[3] SA stands for sampling-actuation delays
[4] SP stands for source of problems

Types of Jitters: Fixed Timing Constraints



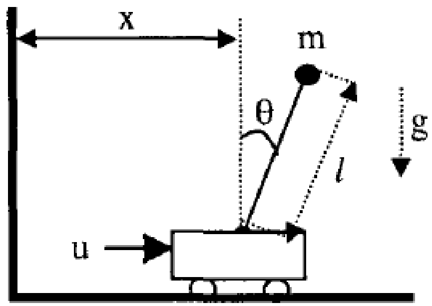
	Task1	Task2	Control task
T	60ms	70ms	80ms
C	10ms	10ms	1ms





Marti et al. 2001

Types of Jitters: Flexible Timing Constraints



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

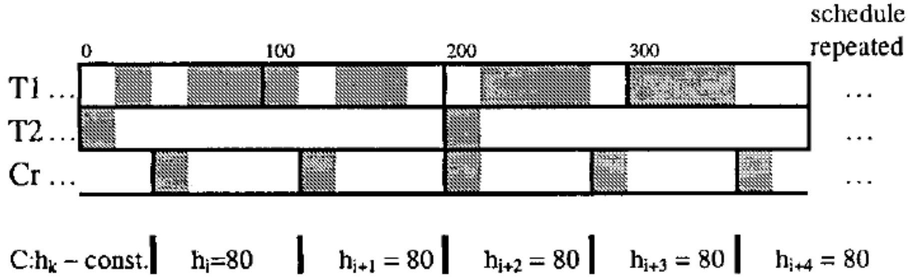


Figure 6. Not feasible schedule

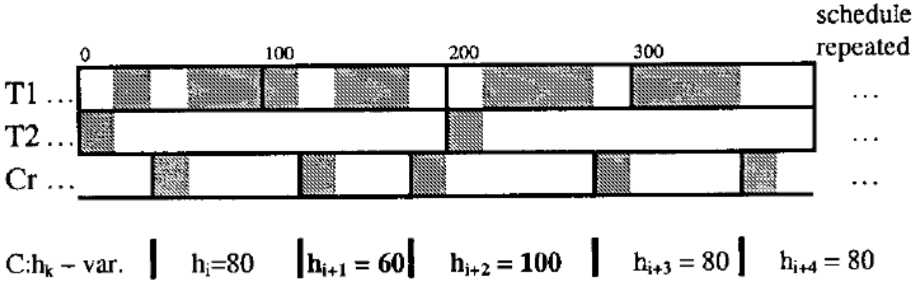
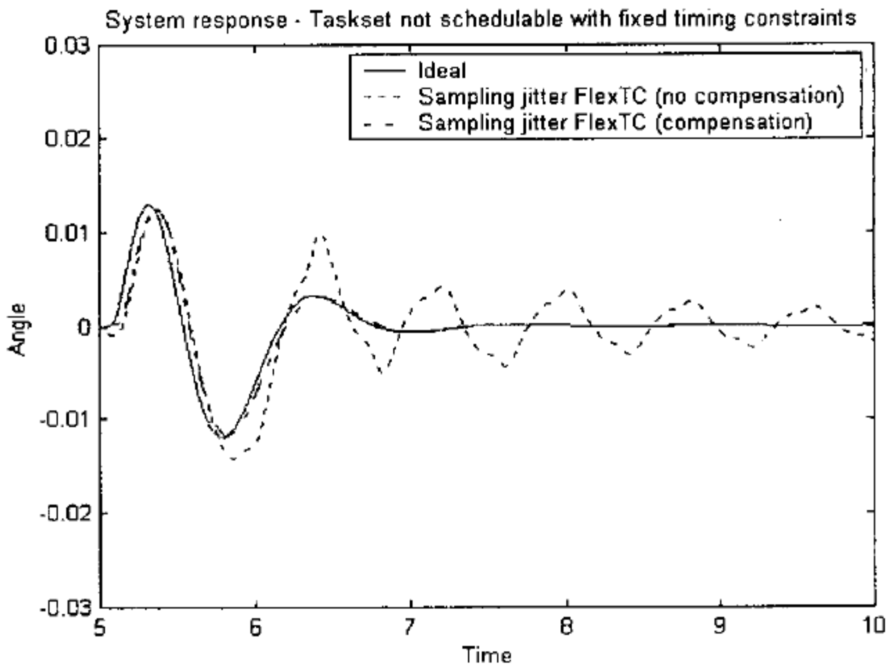


Figure 7. Feasible schedule



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.

Timing Requirements & Control Attributes

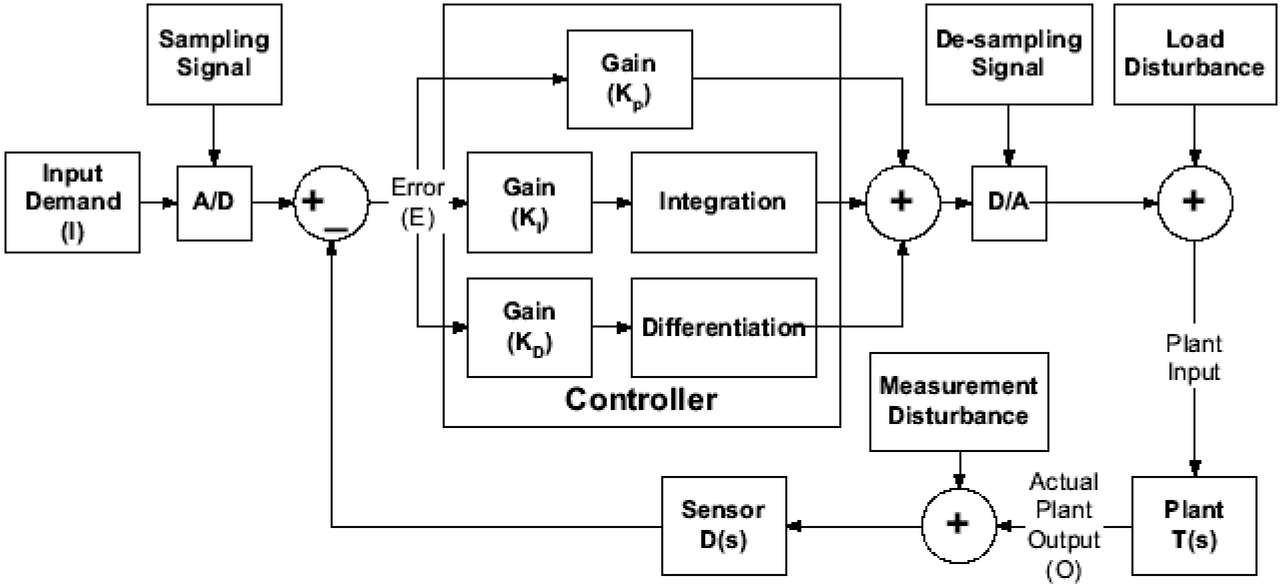
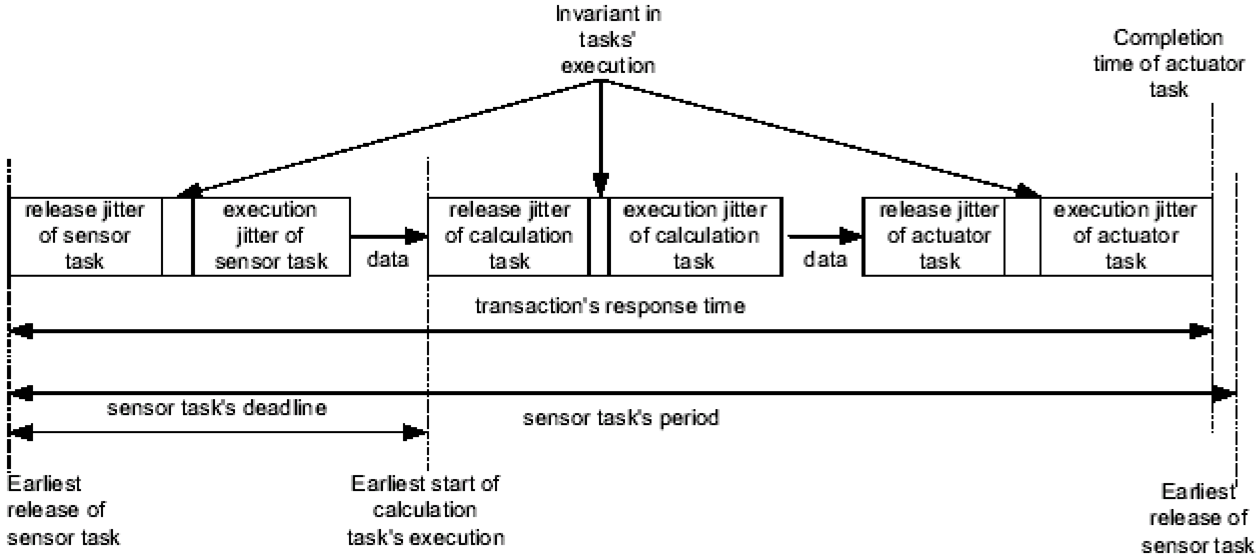


Figure 1 – Typical PID Loop



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.

End-to-End Delay of Videoconferencing

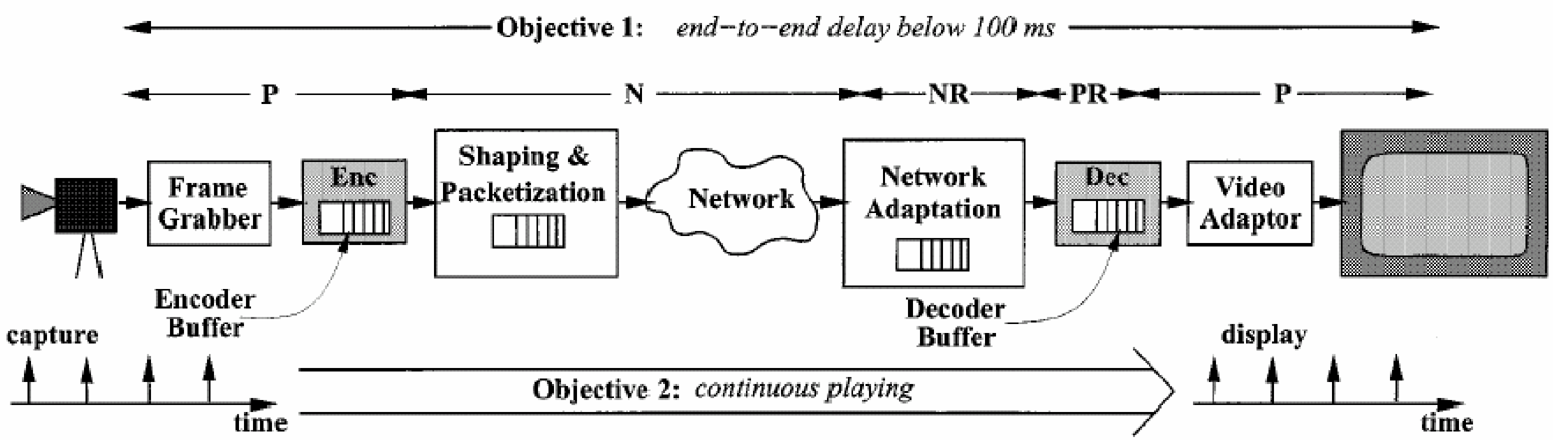
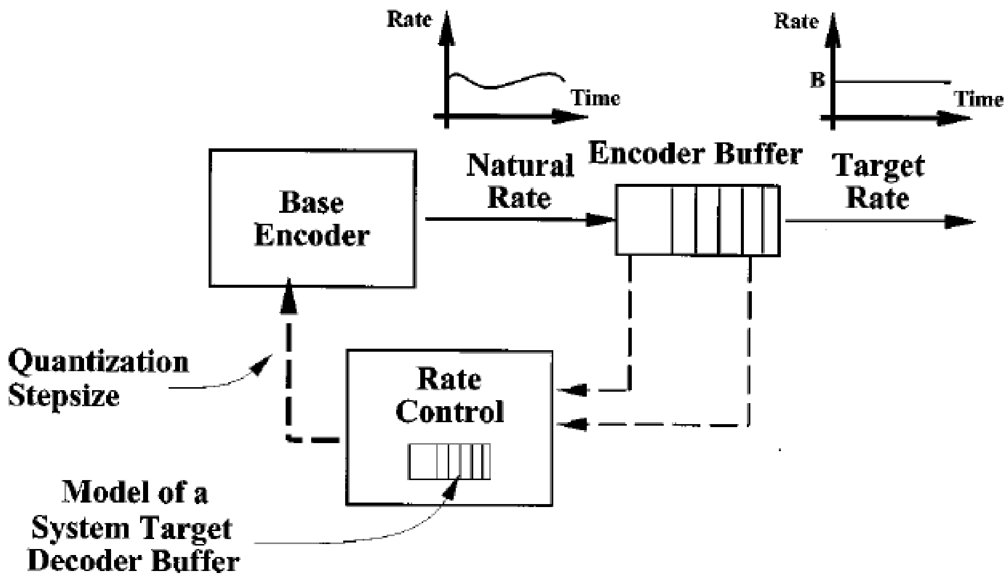


Fig. 1. Model of a videoconferencing system. **P**: processing delay. **PR**: processing resynchronization delay. **N**: network delay. **NR**: network resynchronization delay.

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



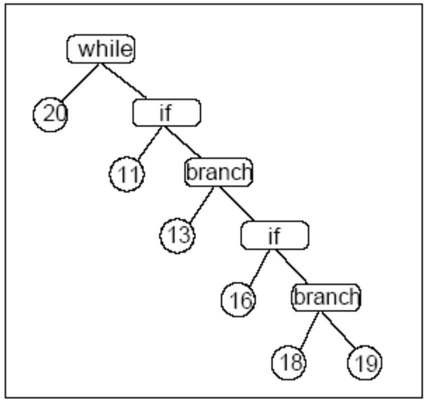
Timing Analysis for Programs

- 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.

Timing Analysis for Programs

```
while (low <= up) {
   mid = (low + up) >> 1;
   if (data[mid].key == x) {
       up = low - 1;
       fvalue = data[mid].value;
   }
  else
     if (data[mid].key > x)
        up = mid - 1;
    else
        low = mid + 1;
}
```

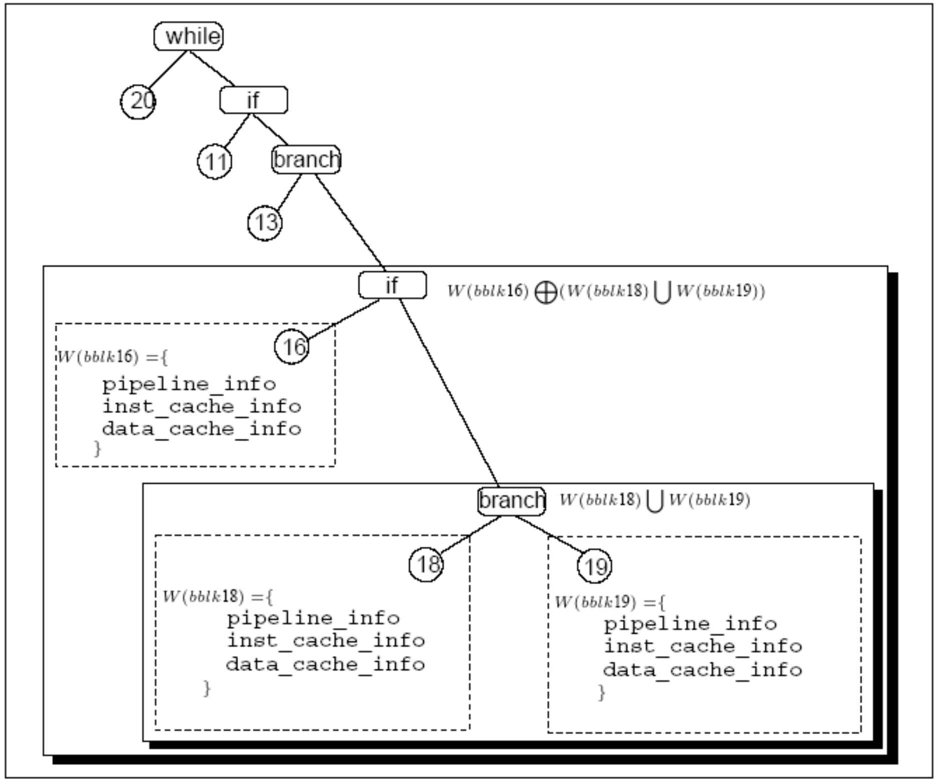
(a)



(b)

```
bblk11:
    addu    $25,$15,$14
    sra     $13,$25,0x1
    sll     $25,$13,0x3
    lui     $1,0x1000
    addu    $1,$1,$25
    lw      $25,0($1)
    nop
    bne     $25,$4,0x40010c
bblk13:
    sll     $25,$13,0x3
    lui     $24,0x1000
    addu    $24,$24,$25
    lw      $24,4($24)
    b       0x400138
    addiu   $14,$15,0xffff
bblk16:
    sll     $25,$13,0x3
    lui     $1,0x1000
    addu    $1,$1,$25
    lw      $25,0($1)
    nop
    slt     $1,$4,$25
    beq     $1,$0,0x400134
    nop
bblk18:
    b       0x400138
    addiu   $14,$13,0xffff
bblk19:
    addiu   $15,$13,0x1
bblk20:
    slt     $1,$14,$15
    beq     $1,$0,0x4000d0
    nop
```

(c)



(d)

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.

Timing Analysis for Programs

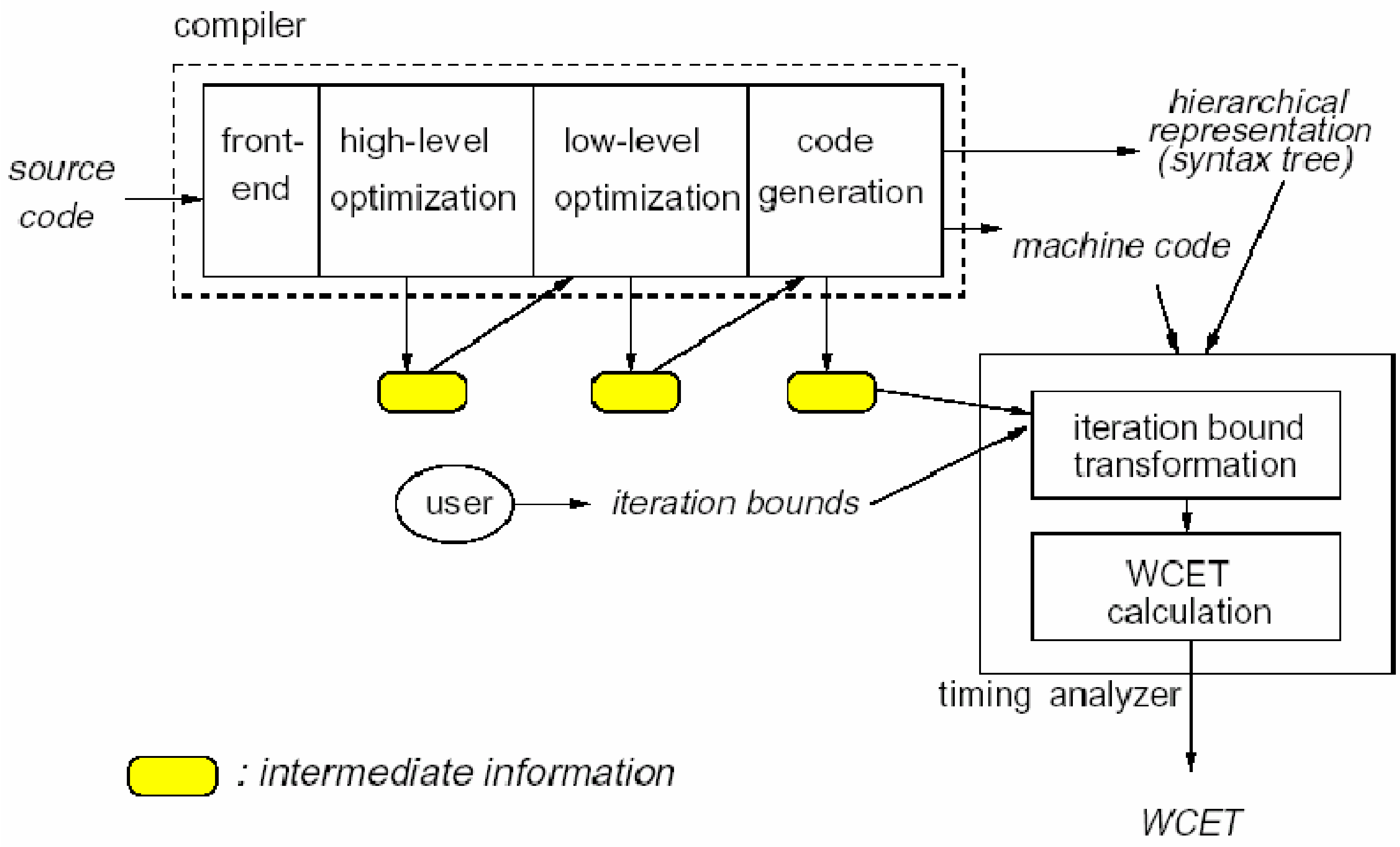


Figure 2. Overview of a compiler-assisted timing analyzer.

Timing Analysis for Programs

Benchmark Programs	Description
MatMul	multiplies two 5 × 5 integer matrices
JFDCTINT	performs the forward Discrete Cosine Transform used in JPEG
FIR	performs a 32-taps Finite Impulse Response (FIR) filtering operation
FFT	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
MatMul	1673	1739
JFDCTINT	4456	4780
FIR	30940	32218
FFT	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.

Multiprocessor Implementation of Digital Engine Control

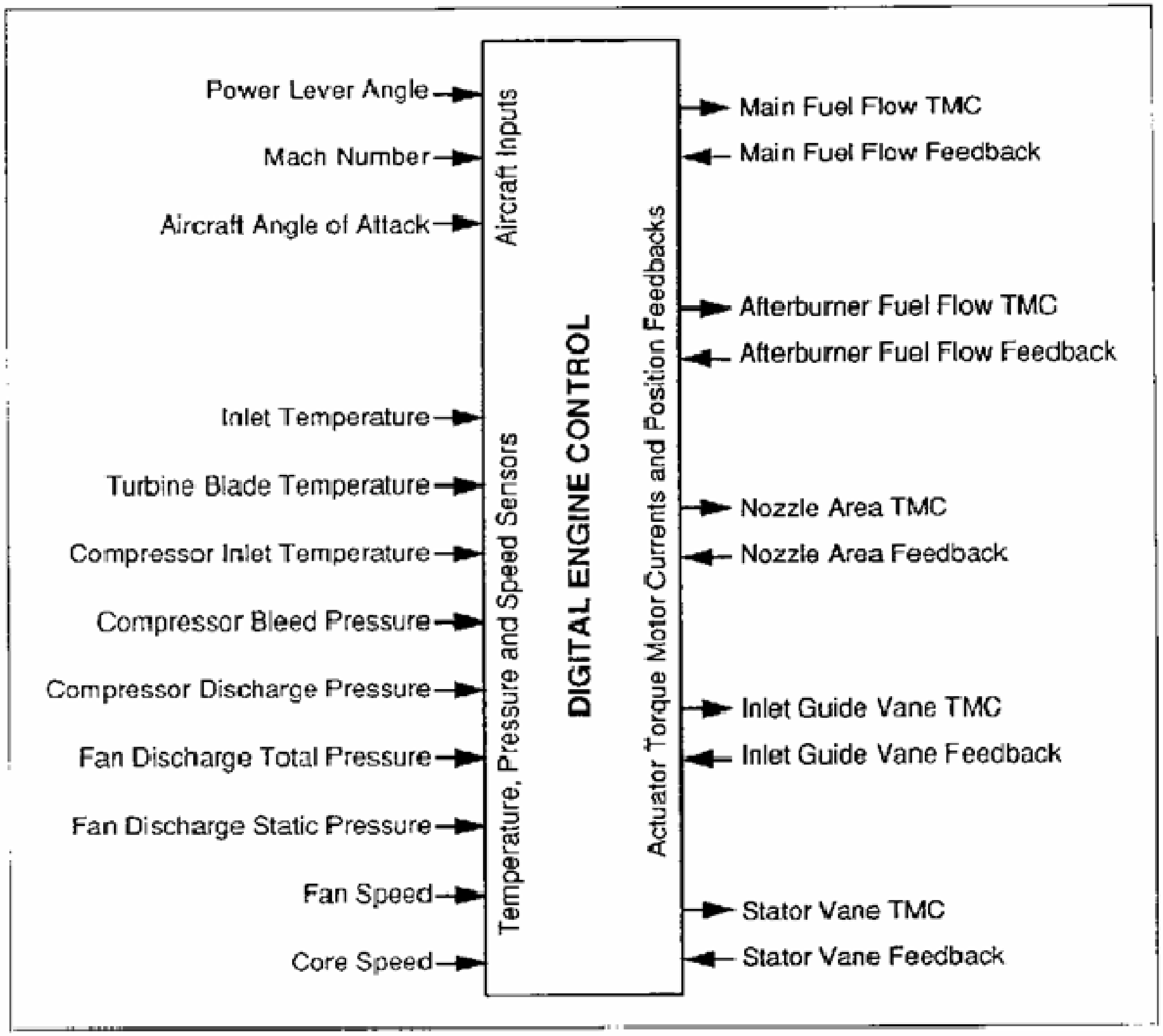


Fig. 1. Digital engine control for a turbojet engine, showing inputs from aircraft, sensor inputs from engine, and actuator torque motor concurrent (TMC) outputs and position feedbacks.

Shaffer 1990

Multiprocessor Implementation of Digital Engine Control

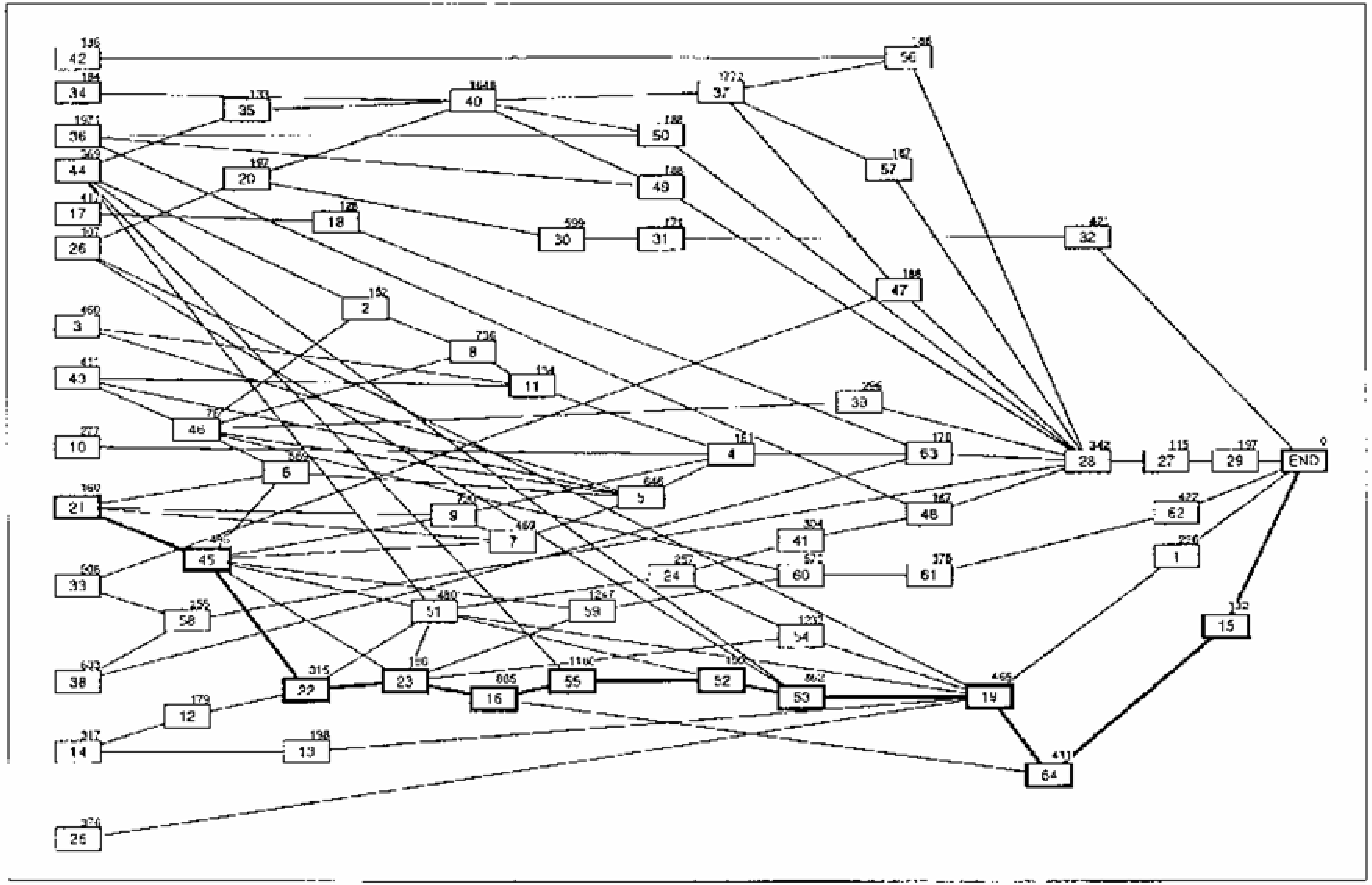


Fig. 2. Computation graph for control program. Each node represents a procedure, and is labeled (on the upper right corner) with the maximum execution time (μs). Graph edges represent data dependencies between routines (data flow is from left to right). External inputs and outputs are not shown.

Multiprocessor Implementation of Digital Engine Control

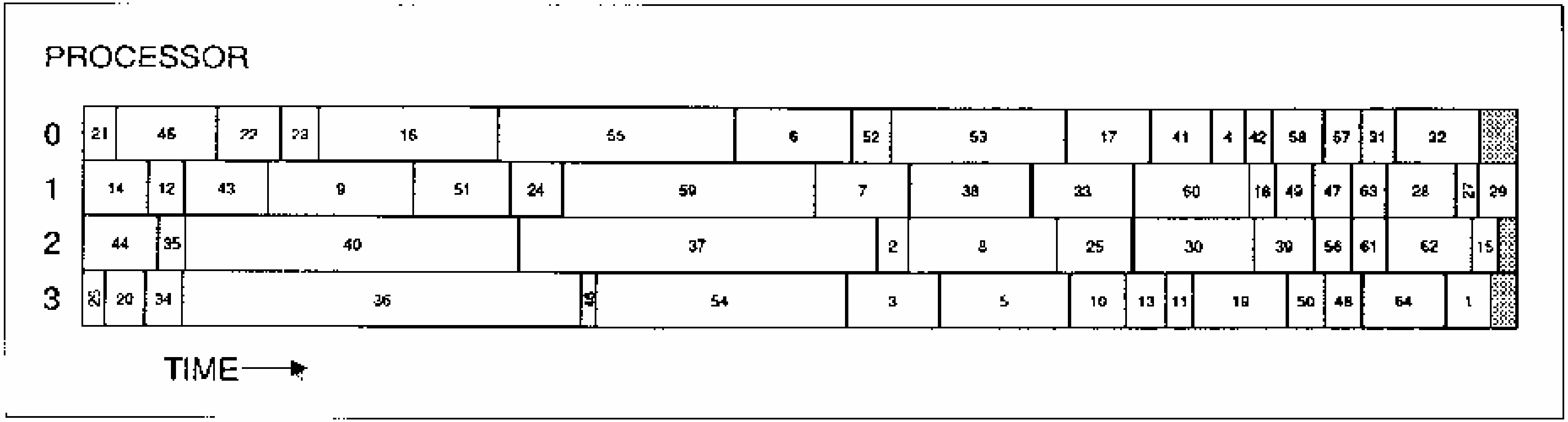


Fig. 3. A possible schedule for the control program on four processors, representing a speedup of 3.94. The shaded areas represent idle processor time.

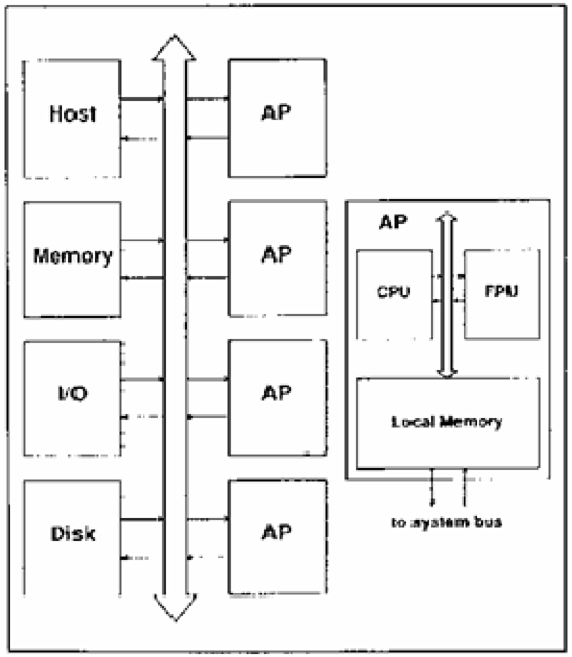


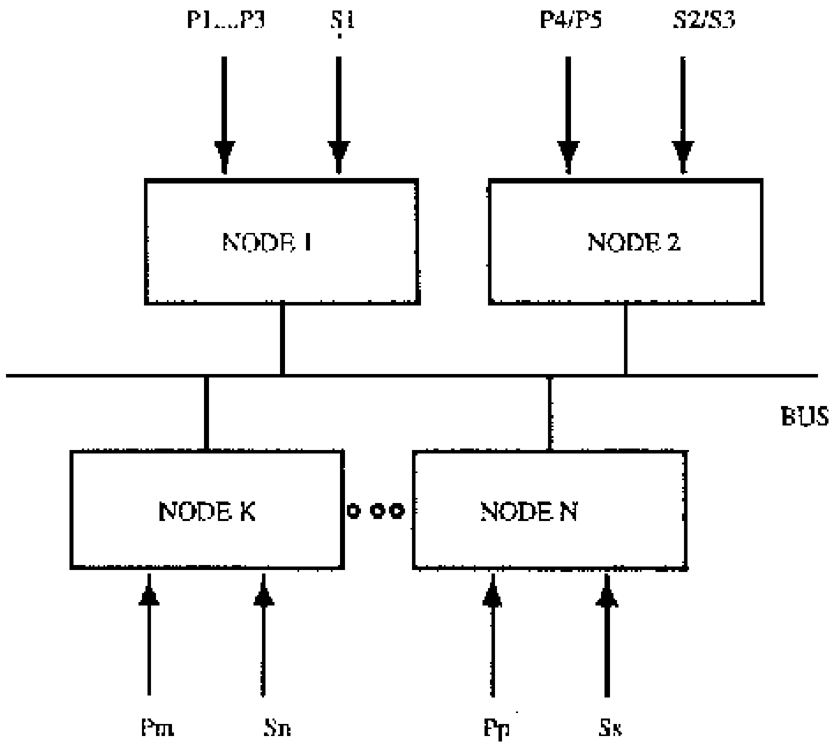
Fig. 4. The multiprocessor architecture used to implement the turbojet engine control program. Each application processor (AP) contains a 68020 central processor unit (CPU) and 68881 floating-point unit (FPU), as well as local RAM. The I/O (input/output) board is an Ethernet controller.

Shaffer 1990

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

Temporal Characteristics of Task Transmission



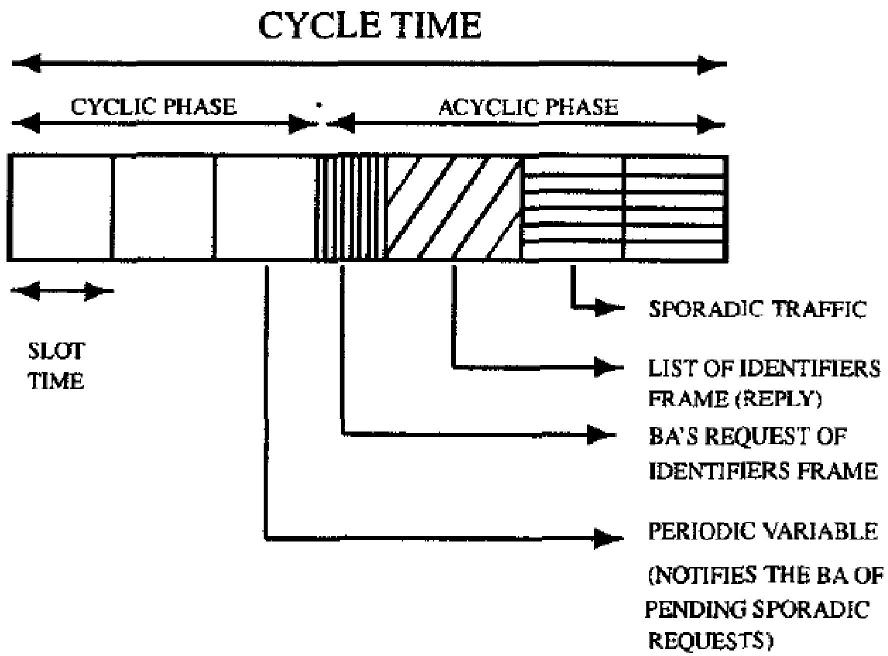


Figure 3. Slot time in a micro-cycle

Temporal Characteristics of Task Transmission

- 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.

Temporal Characteristics of Task Transmission

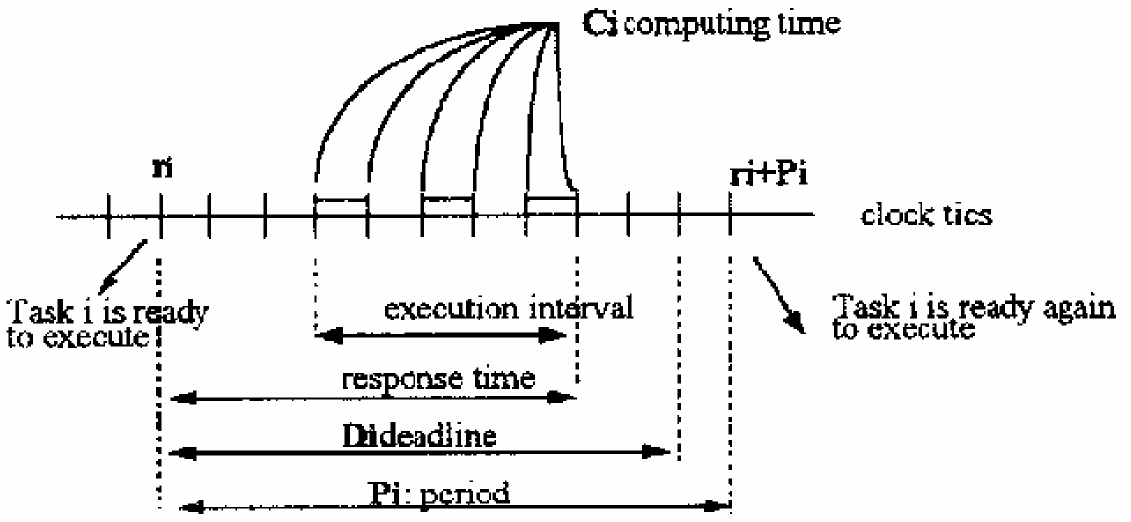


Figure 1: *Temporal characteristics of a task.*

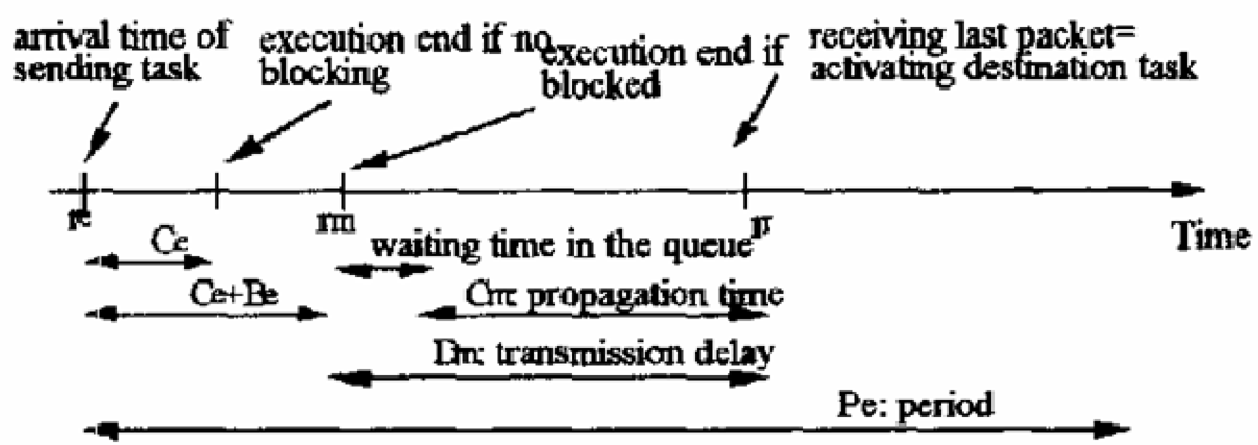
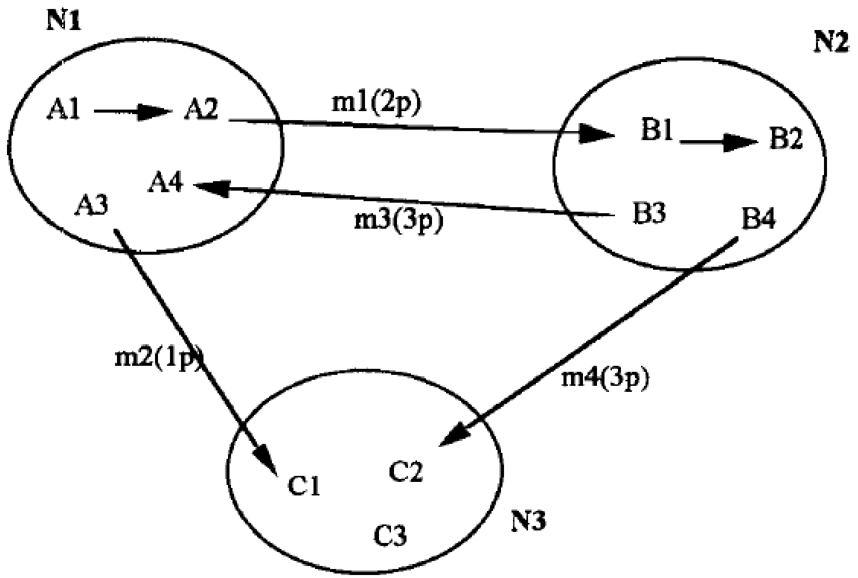


Figure 5: *end to end communication*

Temporal Characteristics of Task Transmission

Table1: Timing characteristics of the application tasks

Task	node	C	P=D	P_r
A_1	N_1	200	10000	3
A_2	N_1	300	10000	2
A_3	N_1	100	20000	1
A_4	N_1	200	5000	4
B_1	N_2	300	10000	2
B_2	N_2	400	10000	1
B_3	N_2	200	5000	3
B_4	N_2	100	15000	1
C_1	N_3	300	20000	2
C_2	N_3	150	15000	3
C_3	N_3	200	10000	4



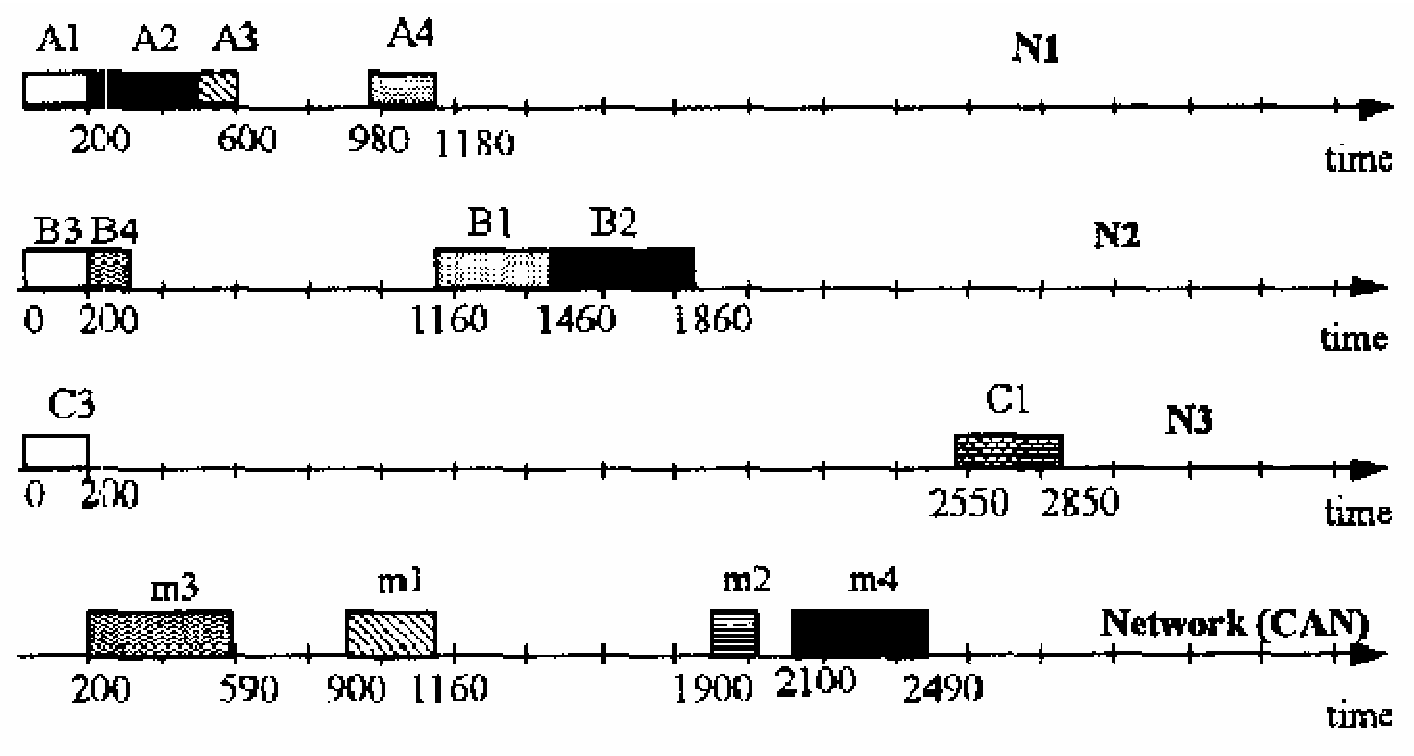
message	m_1	m_2	m_3	m_4
r_m	900	1900	200	2100

Table2: The timing attributes of messages

message	CAN			FDDI		
	C	D	P	C	D	P
m_1	26 0	260	10000	726	3726	10000
m_2	13 0	650	20000	363	6726	20000
m_3	39 0	780	5000	1089	3726	5000
m_4	39 0	1820	15000	1089	12726	15000

Saad-Bouzefrane & Cottet 1997

Temporal Characteristics of Task Transmission



Saad-Bouzefrane & Cottet 1997