



TDDC47: Real-Time Systems and Concurrent Programming

Block 3 b): Scheduling of control processes

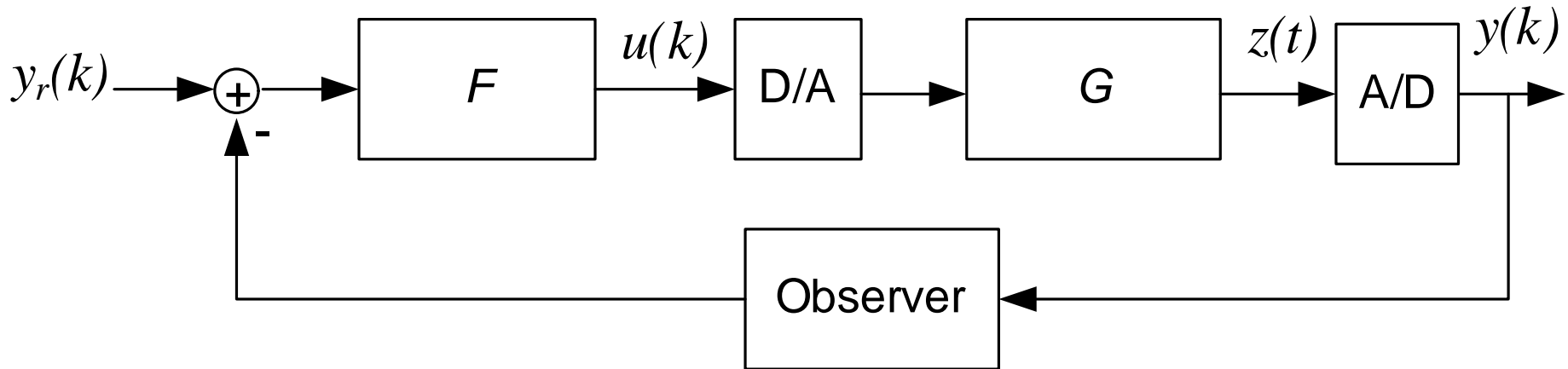
URL: <http://www.ida.liu.se/~TDDC47/>



Jonas Elmqvist
Mehdi Amirijoo
Jörgen Hansson

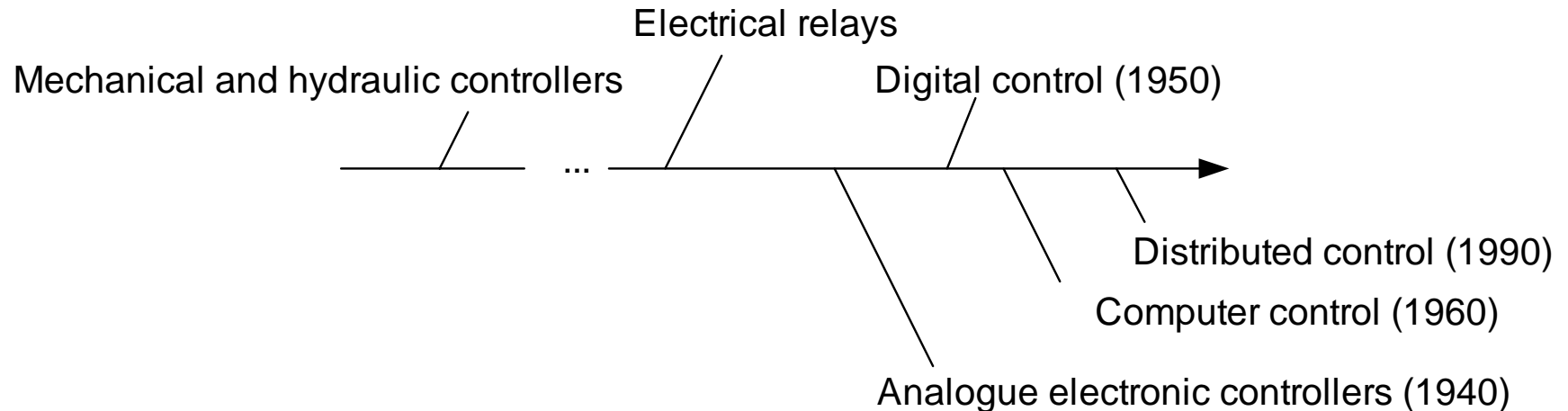
Real-Time Systems Laboratory
Dept. Of Computer Science
Linköping university

- The goal of this lecture is to bring forward:
 - the nature of modern control systems
 - how temporal properties in the real-time domain are mapped to temporal properties in the control domain
 - how temporal properties in the control domain affect the control performance
 - how the control performance is improved by adopting a
 - control engineering and a
 - real-time schedulingapproach



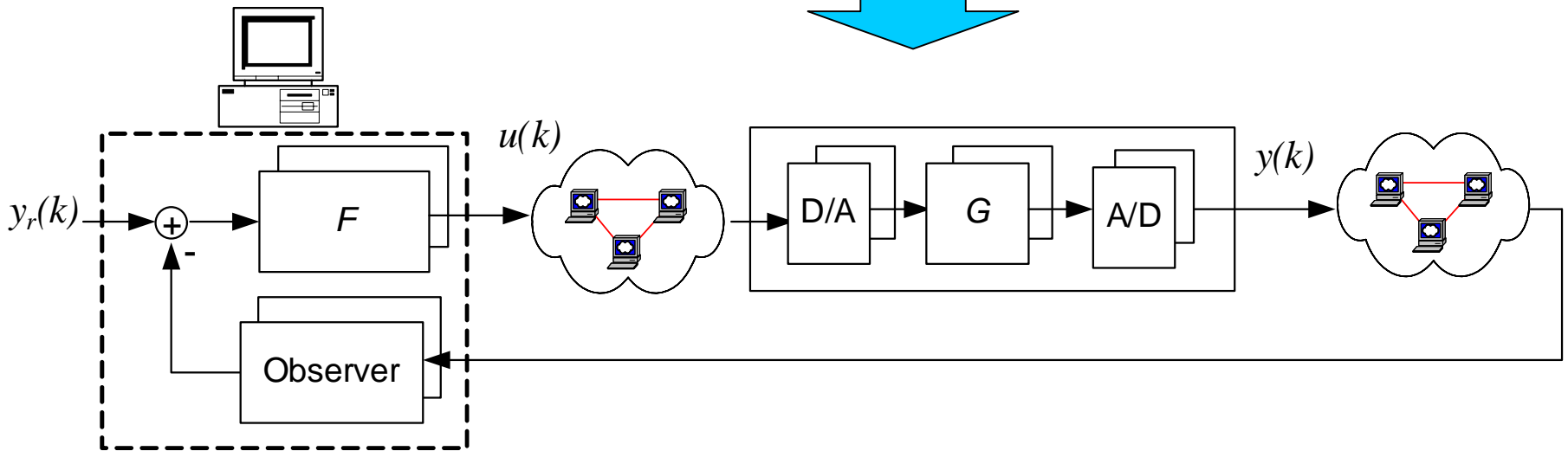
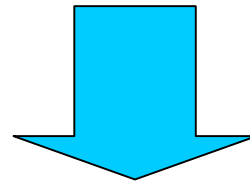
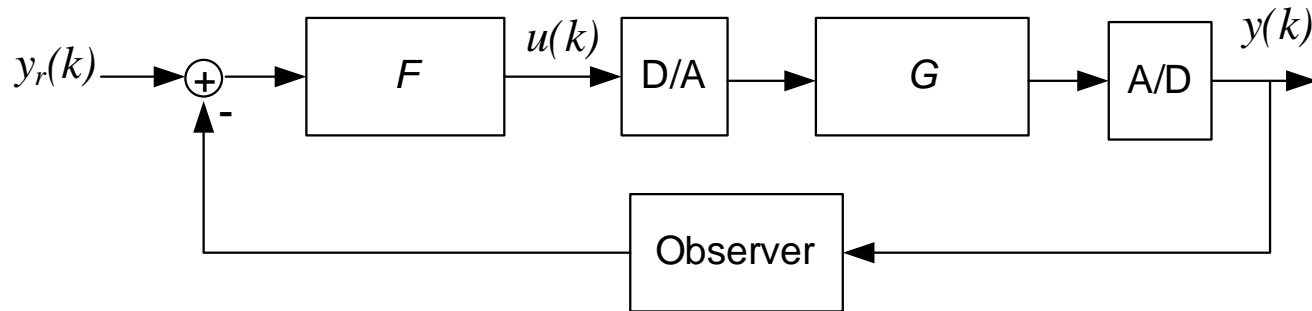
- $y_r(k)$: Reference variable (börvärdet)
- $u(k)$: The manipulated variable (styrsignal)
- $z(k)$: The controlled variable (reglerstorhet)
- $y(k)$ The measured variable (mätsignalen)
- G : Controlled system (reglerobjektet)
- F : Controller (regulator)

- How are control systems implemented?
A historical perspective.



Current status:

- Multiple control algorithms on the same computer
Each control algorithm implemented using a process
- Distributed control
Sensors, control algorithms, actuators on different nodes





The Complexity of Control Systems

- Control engineering is a multi-disciplinary subject
 - Mechanics
 - Control design
 - Computer science and engineering
 - Numerical representation, fixed point calculus
 - Scheduling
 - Networking
 - ...

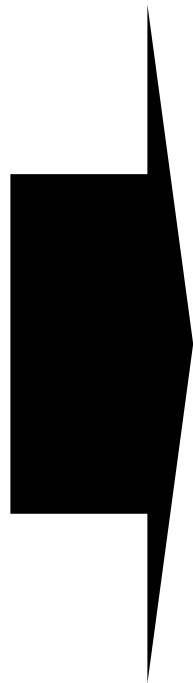
We focus on how *process scheduling + networking* affects the control performance

Question: How can scheduling and networking affect the control performance?

Mapping of Temporal Properties

How are temporal properties mapped?

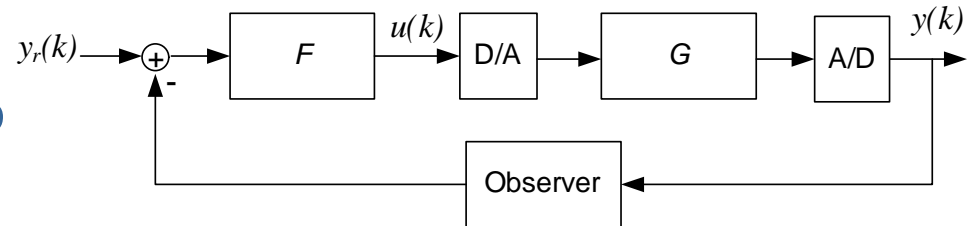
Response time
Execution time
Blocking
Transmission time
Output jitter
Offset
Utilization
Vacant sampling
...



Control period (reglerperiod)
Control delay (reglerfördröjning)
Jitter
Transient error (transient fel)

- Consider the DC-servo model:

$$G(s) = \frac{1000}{s(s+1)}$$



- The controlled system is sampled with the period T
- A discrete PD (Proportional Derivative) controller

$$F(z) = -K \left(1 + \frac{T_d}{T} \frac{z-1}{z} \right), \quad K = 1.5, \quad T_d = 0.035$$

is used.

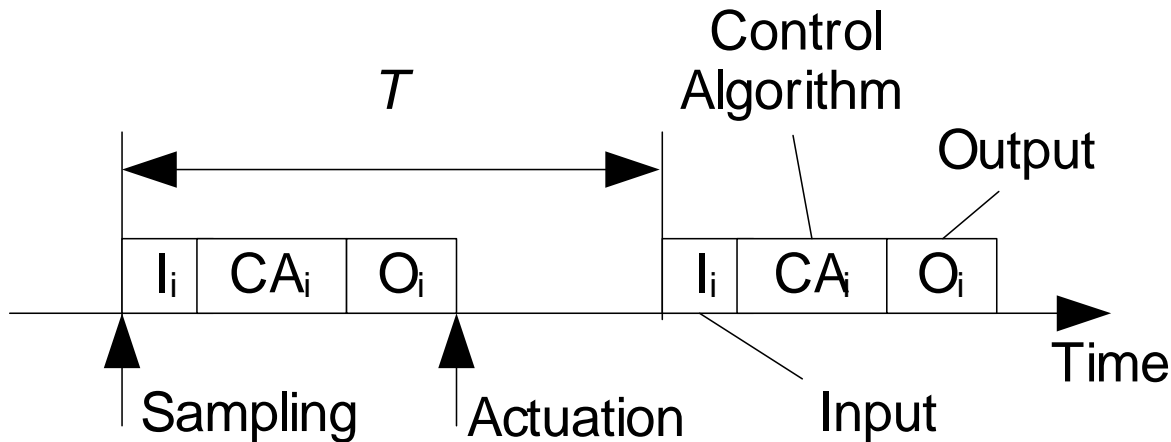
- Use the linear-quadratic cost function

$$J = \lim_{T \rightarrow \infty} \frac{1}{T} \int_0^T (y^2(t) + u^2(t)) dt$$

to measure the performance of the controller.

Definition:

The *control period* T determines the rate of sampling, control algorithm computation, and actuation



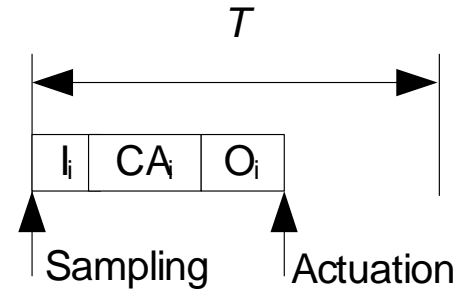
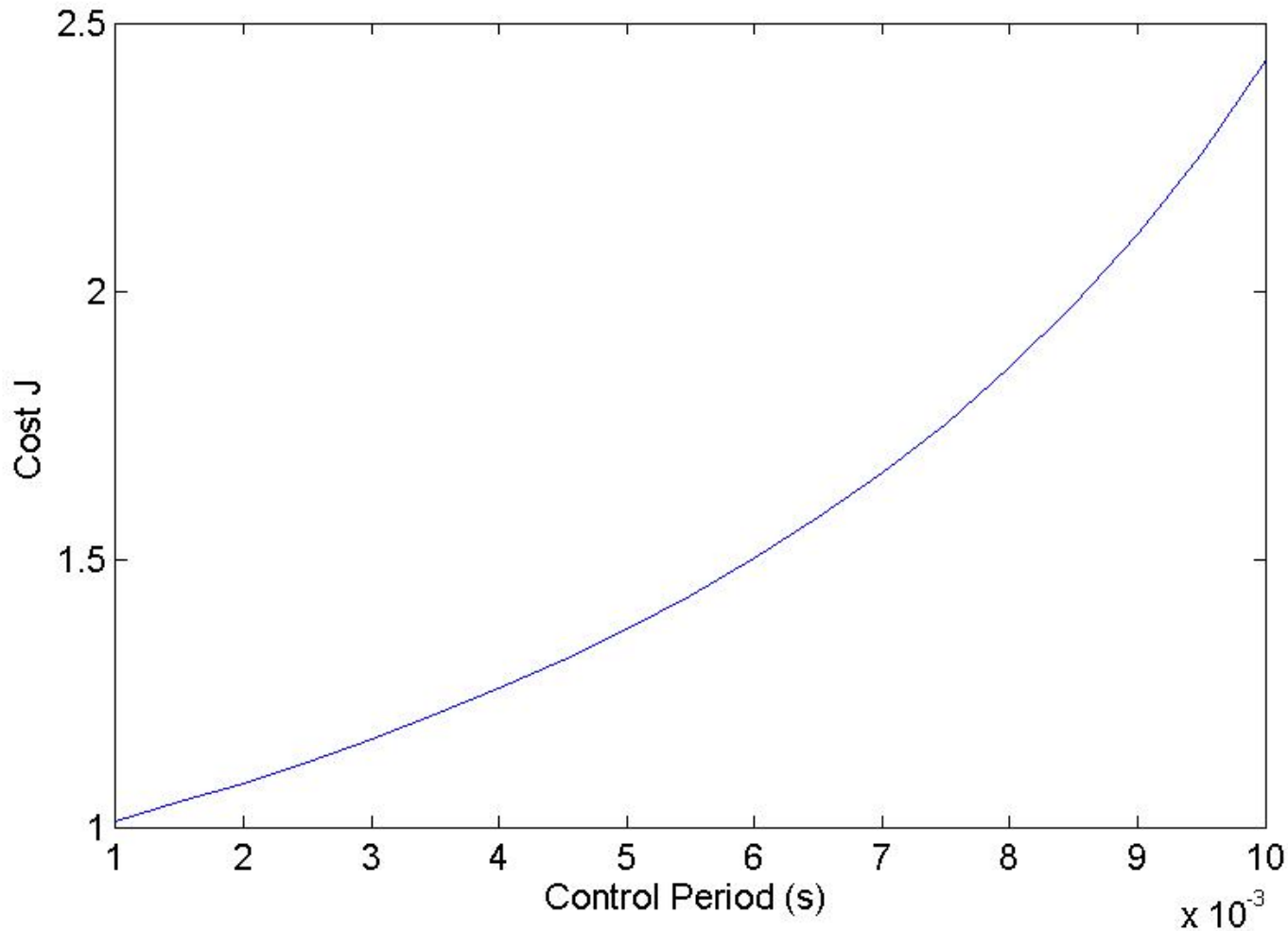
–“The shorter control period, the better performance”

- Conflicts:

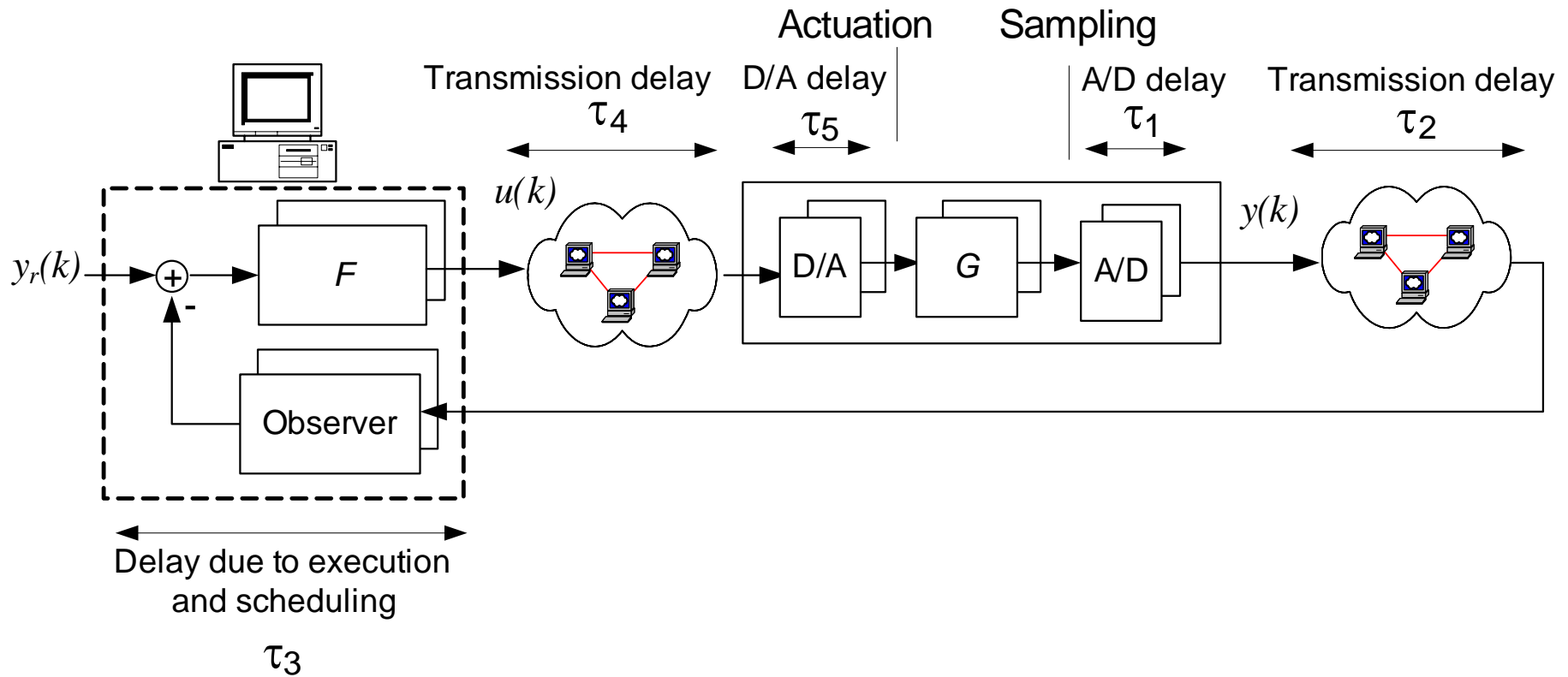
- The period is based on the dynamics of the closed-loop system, $0.2 < \omega_b T < 0.6$ ω_b = bandwidth of closed-loop system
- Based on the available resources; remember the RM utilization test:

$$\sum_{i=1}^n \frac{C_i}{T_i} \leq n \left(2^{\frac{1}{n}} - 1 \right)$$

How does control period affect performance?

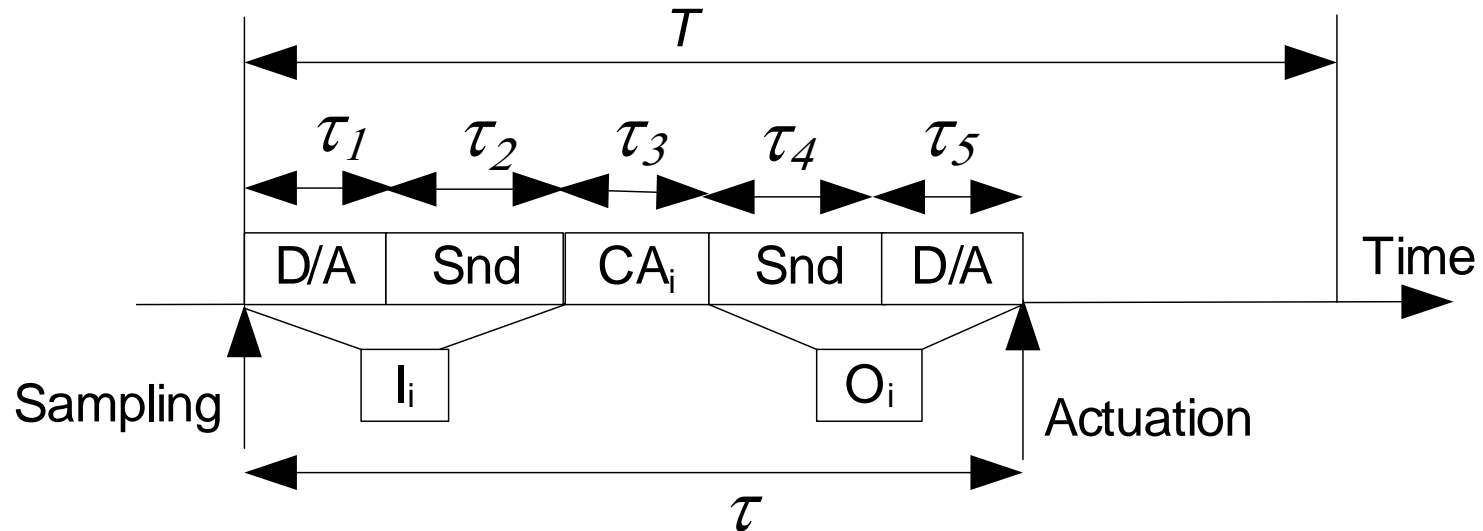


Problems – Control Delay



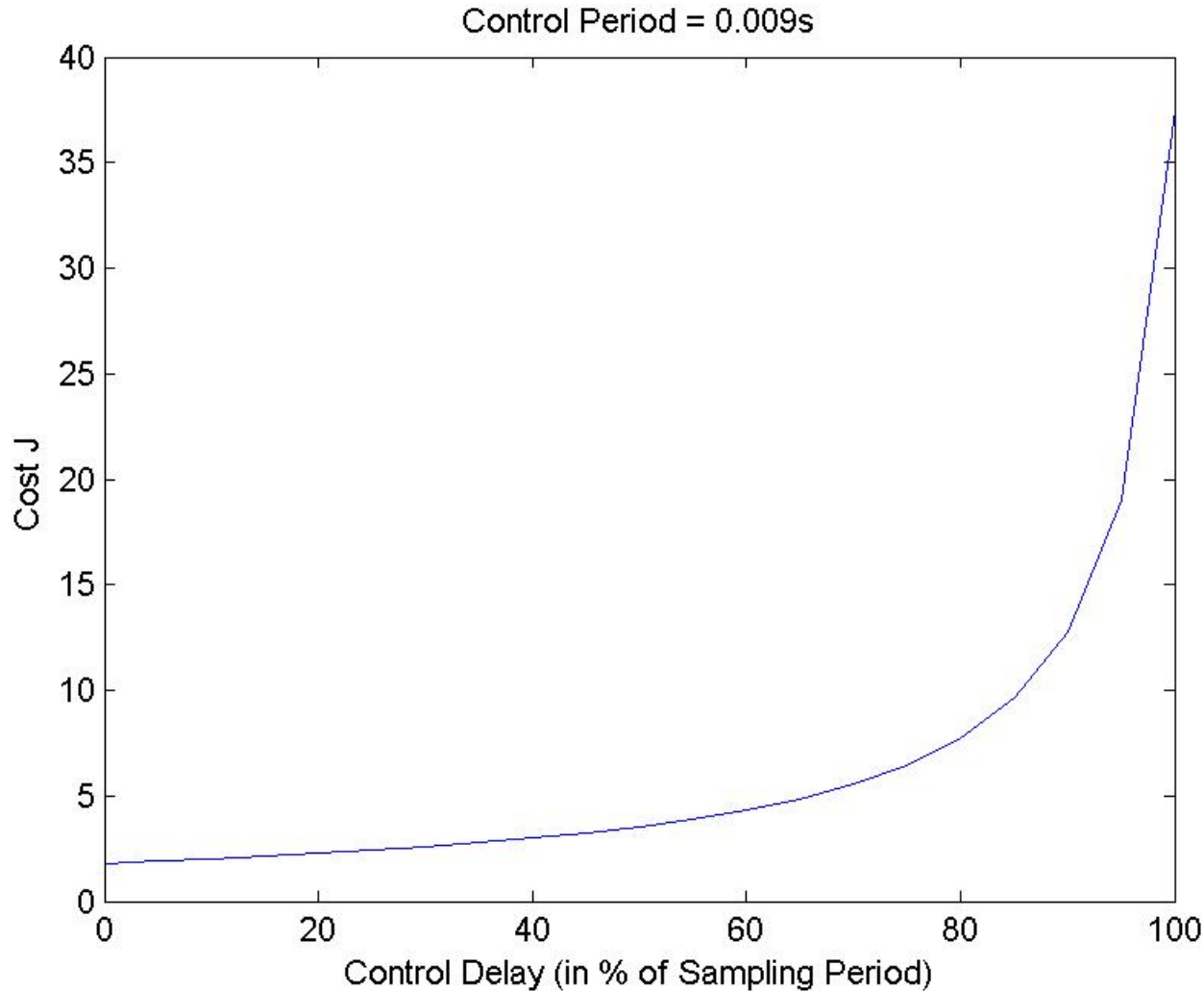
Definition:

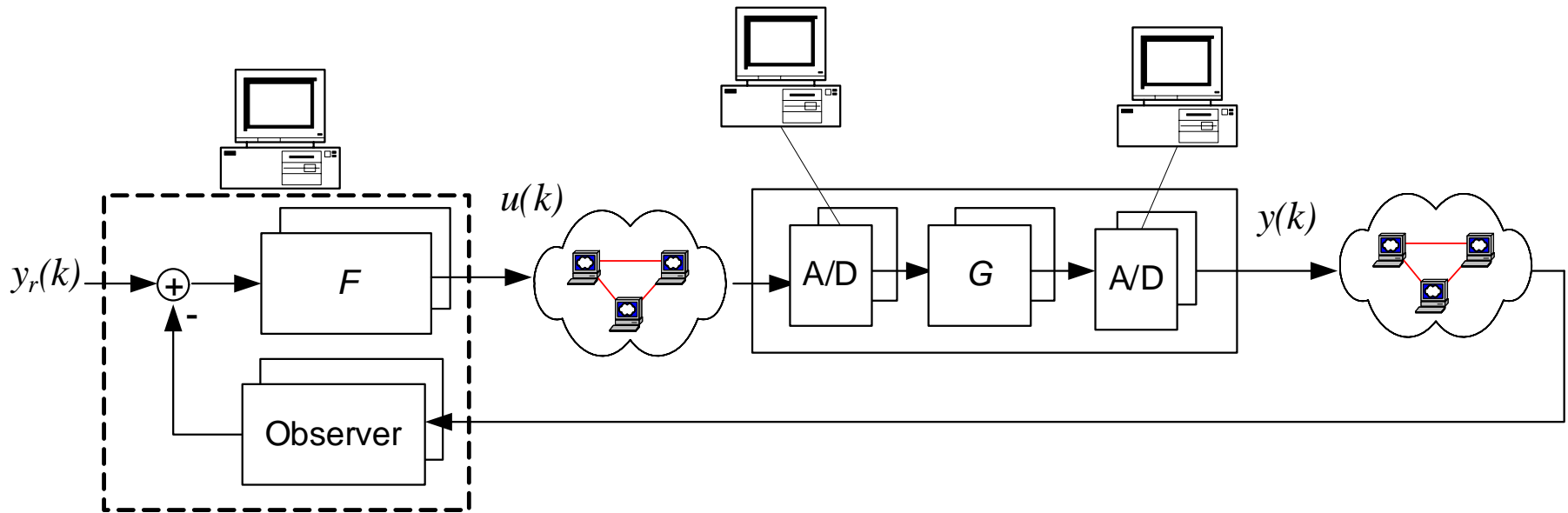
A control delay (τ) is the time between sampling and actuation.



- Caused by:
 - sample & hold (SH), D/A, A/D,
 - network communication,
 - execution of control algorithms, synchronization, and interference and blocking.
- A constant delay decreases the phase margin (can lead to instability)

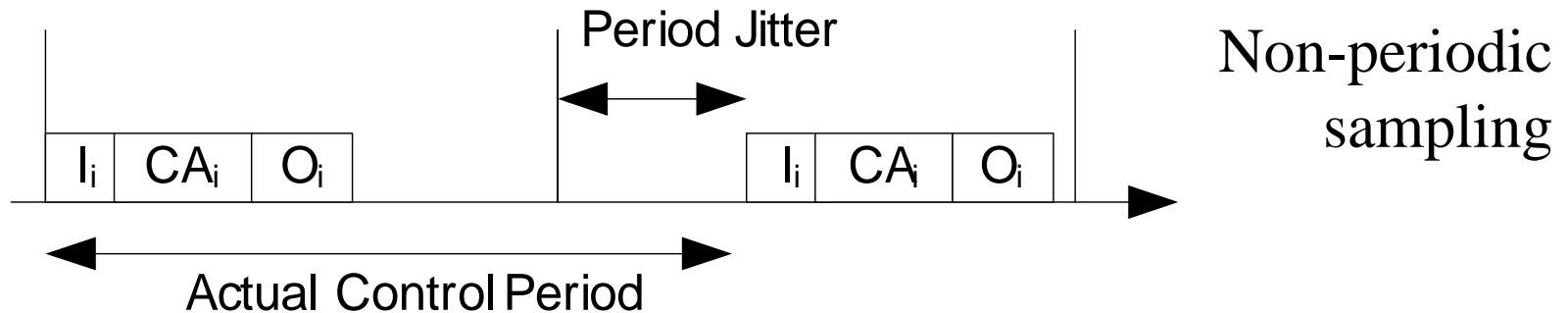
How does control delay affect performance?



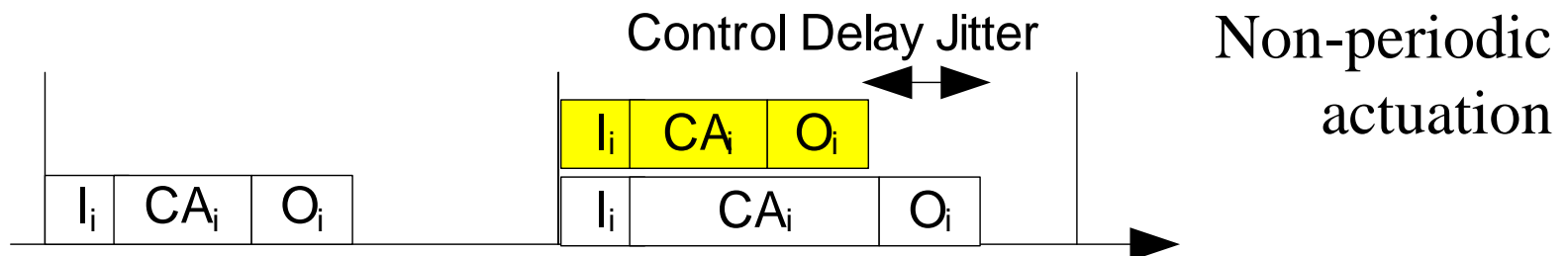


Definition:

(i) Period jitter – variations in control period



• (ii) Delay jitter – variations in control delay



• Caused by:

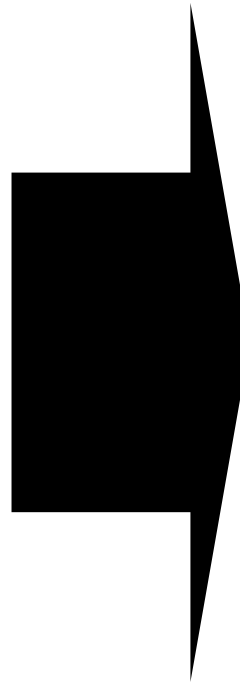
- Hardware: cache memory
- Software: branching in code
- Process scheduling: interference and blocking

• In general, jitter degrades the control performance

- Definition:
A transient error is caused by a fault with a finite duration
- Results in:
 - Loss or corruption of control related variable
 - E.g. $u(2)$ is lost
 - Temporary suspension of control action
- Caused by:
 - Error in software
 - Failures in message delivery
 - Packets dropped
 - Packets arrive too late
- Effects performance and stability

We have the following temporal mappings:

Response time
Execution time
Blocking
Transmission time
Output jitter
Offset
Utilization
Vacant sampling
...

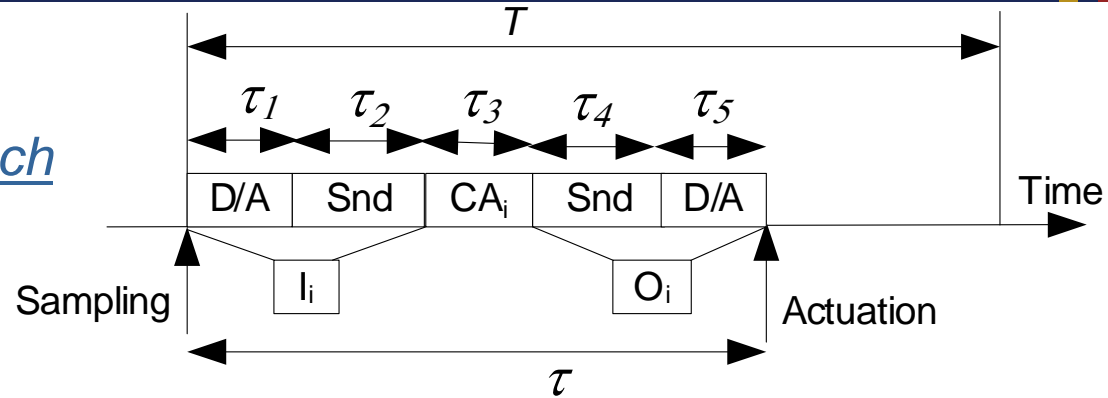


Control period
Control delay
Jitter
Transient error

A co-design approach is needed!

Solutions – Control Delay

Control engineering approach

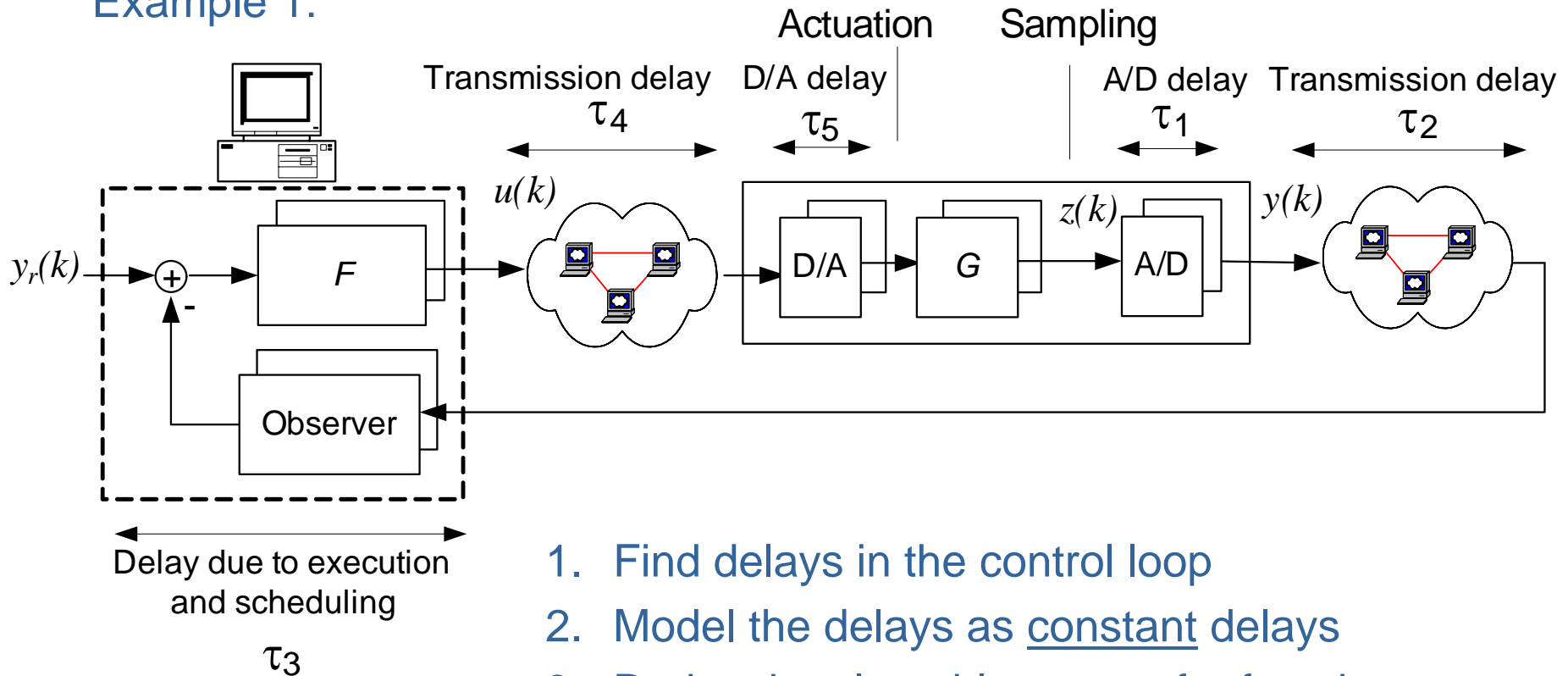


Use delay compensation. This involves:

1. Find delays in the control loop
2. Model the delays as constant delays
3. Derive the closed-loop transfer function
4. Design a controller with that compensates for the delay

Works well if we have no or small delay jitter

Example 1:



1. Find delays in the control loop
2. Model the delays as constant delays
3. Derive the closed-loop transfer function
4. Design a controller with that compensates for the delay

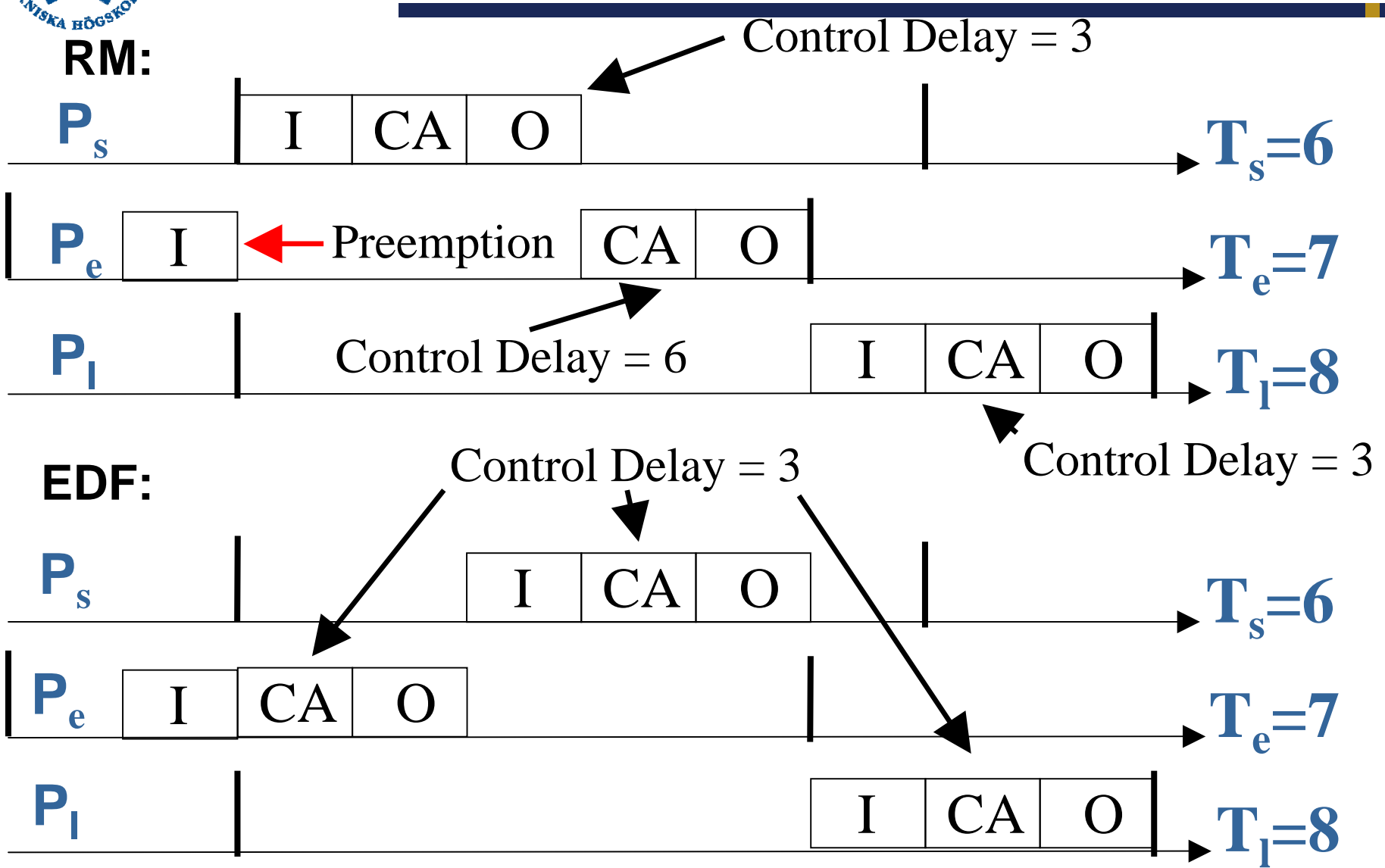
Real-time scheduling approach

Theorem: *The maximum control delay of each process under EDF is shorter than or equal to the corresponding maximum control delay under RM.*

Proof sketch:

- Minimum control delay is achieved if the control process is not interrupted. Assume that process P_e has the earliest absolute deadline. The process P_l has longer relative deadline (period) than P_e . The process P_s has shorter relative deadline (period) than P_e . Assume that P_e is about to start executing CA, and that P_l and P_s arrives.
- RM:
 P_e is not preempted by P_l . However, P_e is preempted by P_s .
- EDF:
 P_e is not preempted by P_l , since P_l has a later absolute deadline.
If P_s has a later absolute deadline than P_e , then P_e cannot be preempted by P_s
 $\Rightarrow P_e$ can continue executing (CA and O).

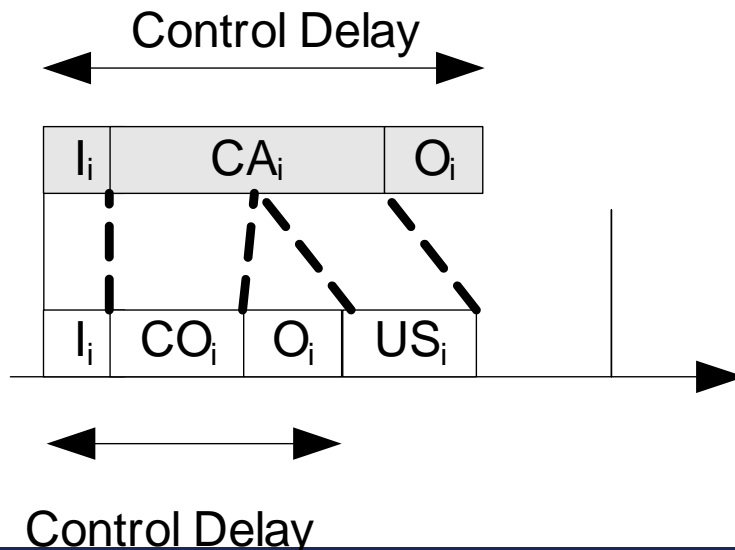
Solutions – Control Delay



Real-time scheduling approach 2

The control algorithm is divided into two subprocesses:

1. Minimum computation to produce output
2. The rest of the computation:
 - Updating the state of controller or filter
 - Update the integral part of an (P)I(D) controller (forward difference)
 - Update the state of a Kalman filter



```

LOOP
  ReadInput;           // I
  CalculateOutput;    // CO
  WriteOutput;        // O
  UpdateState;        // US
  WaitForNextPeriod;
END
    
```

Real-time scheduling approach 2

Subprocess scheduling:

Let $P_{i,C}$ denote the subprocess of P_i consisting of I_i , CO_i , and O_i

Let $P_{i,U}$ denote the subprocess of P_i consisting of US_i

Algorithm:

1. Start by setting the relative deadline (D) of each subprocess

$$D_{i,U} = T_{i,U} = T_i, \quad D_{i,C} = T_{i,U} - C_{i,U}$$

2. Assign priorities (π) according to deadline monotonic (DM), where the priority increases as D decreases.
3. Calculate worst-case response times $R_{i,C}$ of each $P_{i,C}$
4. Assign new relative deadlines to each $P_{i,C}$ according to

$$D_{i,C} = R_{i,C}$$

5. Repeat from 2 until no further improvements in response times is obtained.

Example 2: subprocess scheduling

1st iteration, steps 1-3

Subprocess	T	C	D	π	R
P _{1,C}	20	3	16	6	3
P _{1,U}	20	4	20	5	7
P _{2,C}	29	3	25	4	10
P _{2,U}	29	4	29	3	14
P _{3,C}	35	3	31	2	17
P _{3,U}	35	4	35	1	28

3rd iteration

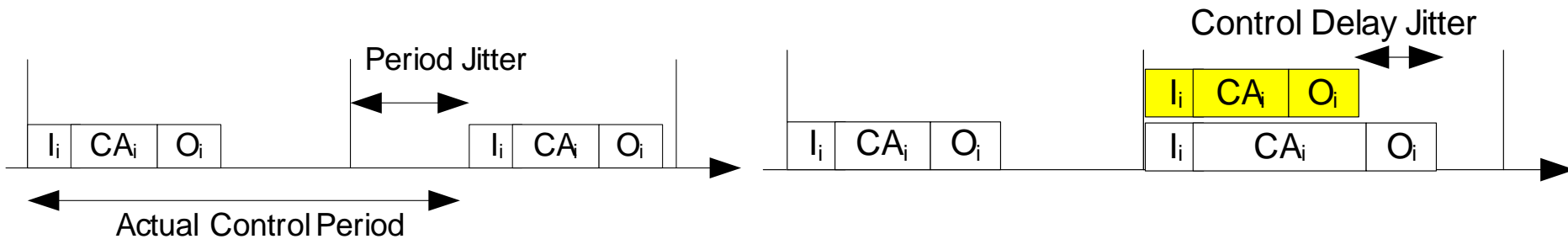
Subprocess	T	C	D	π	R
P _{1,C}	20	3	3	6	3
P _{1,U}	20	4	20	3	13
P _{2,C}	29	3	6	5	6
P _{2,U}	29	4	29	2	17
P _{3,C}	35	3	9	4	9
P _{3,U}	35	4	35	1	28

$$R_i = C_i + \sum_{\forall P_j \in hp(P_i)} \left\lceil \frac{R_i}{T_j} \right\rceil C_j$$

Control engineering approach

- Period jitter
 - Using jitter compensating controllers
 - Measure the actual control period and re-compute controller parameter

Not part of this course!



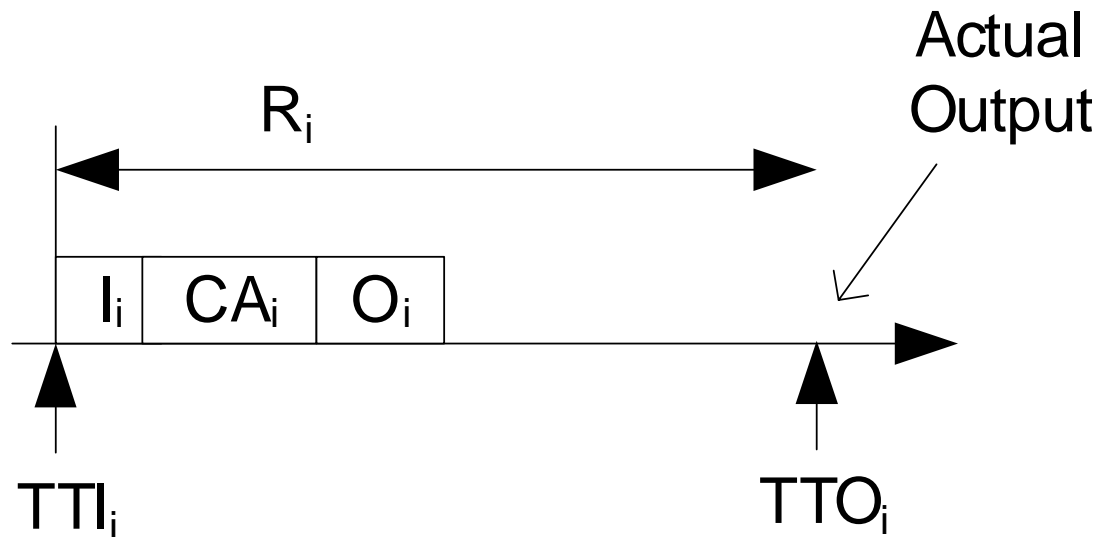
- Control delay jitter
 - Varying control delay \Rightarrow time-varying system!
 - Use robust control design

Provides robustness against delay uncertainties

Real-time scheduling approach

Statically scheduled I/O

- Time-triggered I/O:
Time-triggered input (TTI), Time-triggered Output (TTO)
- I/O taken care of by kernel (operating system)

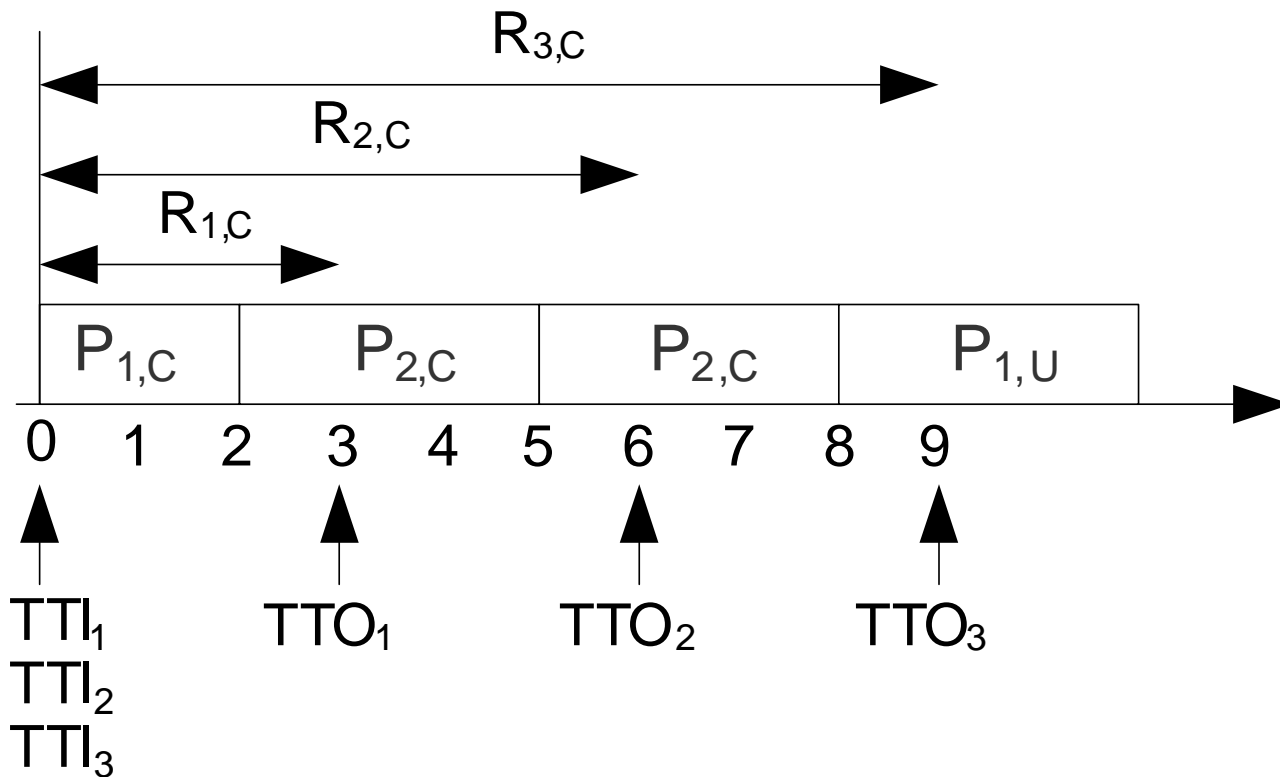


- Minimal period jitter and control delay jitter
- Constant control delay \Rightarrow use delay compensation (see solution on control delay)

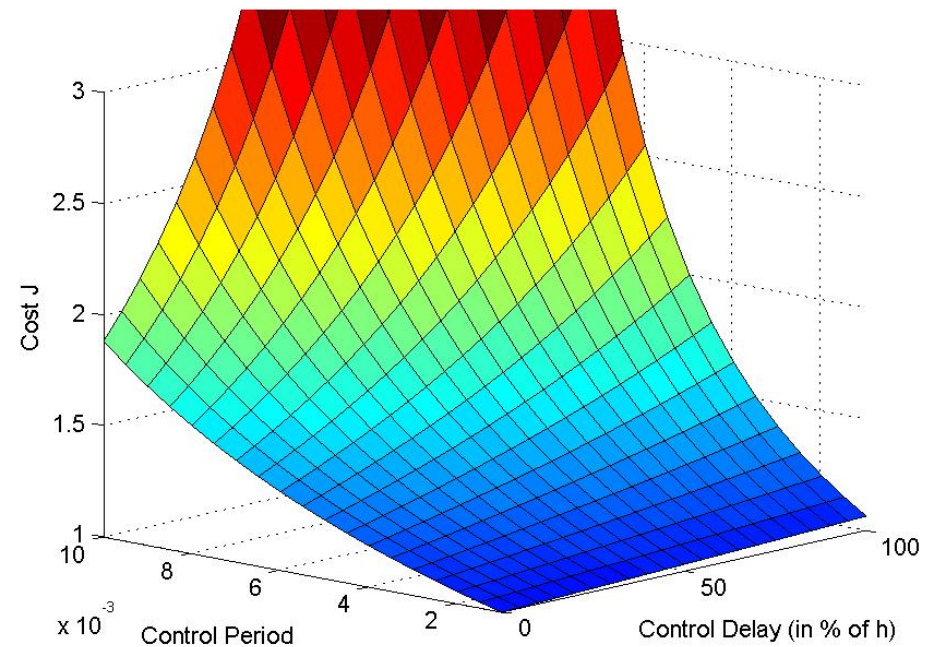
Real-time scheduling approach

Statically scheduled I/O (cont')

- Combine with subprocess scheduling to get low control delay
- Compare to example 2:



- Recent advances:
 - Multiple control algorithms on the same computer
 - Distributed control
- The temporal properties
 - control period,
 - control delay,
 - jitter, and
 - transient errorall degrade the control performance
- A co-design approach is needed:
 - Control engineering
 - Real-time scheduling



- Real-time scheduling:
 - Try to minimize the control delay
 - The control algorithm is divided into two subprocesses
 - Subprocess scheduling reduces the control delay
 - Using statically scheduled I/O reduces jitter
 - Combined approach results in improved control performance
- Readings (in the course compendium):
 - A. Cervin, D. Henriksson, B. Lincoln, J. Eker, and K.-E. Årzen. *How does control timing affect performance? analysis and simulation of timing using jitterbug and truetime*. In IEEE Control Systems Magazine, 23:16-30, 2003.
 - B. Wittenmark, J. Nilsson, and M. Törngren. *Timing problems in real-time control systems*. In Proceedings of the American Control Conference, vol 3, pages 2000-2004, 1995.