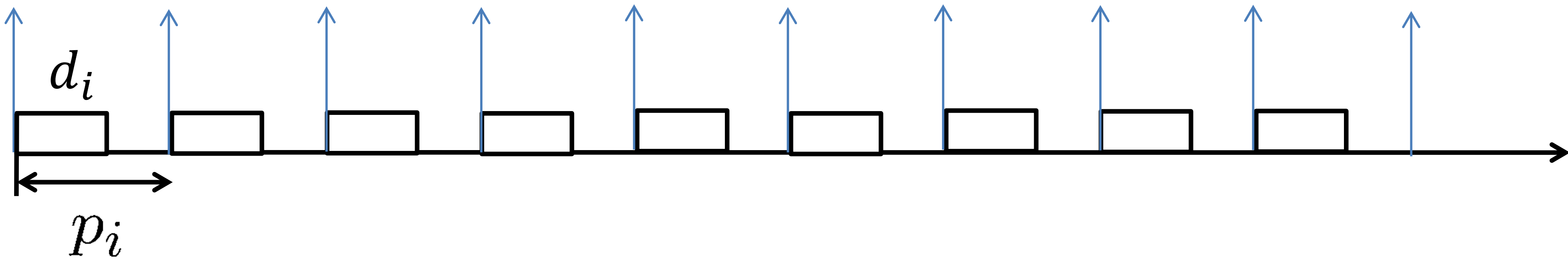
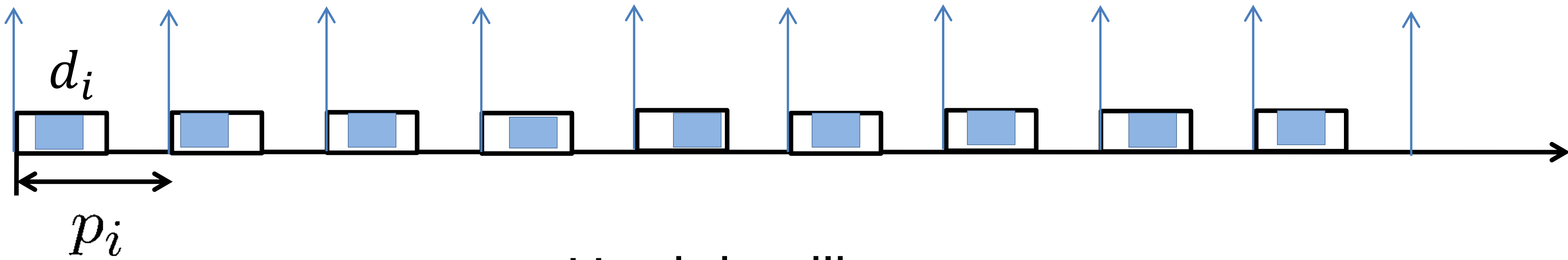
Deadlines misses and their Implication on Feedback Control Loops

Dip Goswami
Eindhoven University of Technology (TU/e)
The Netherlands

Periodic tasks

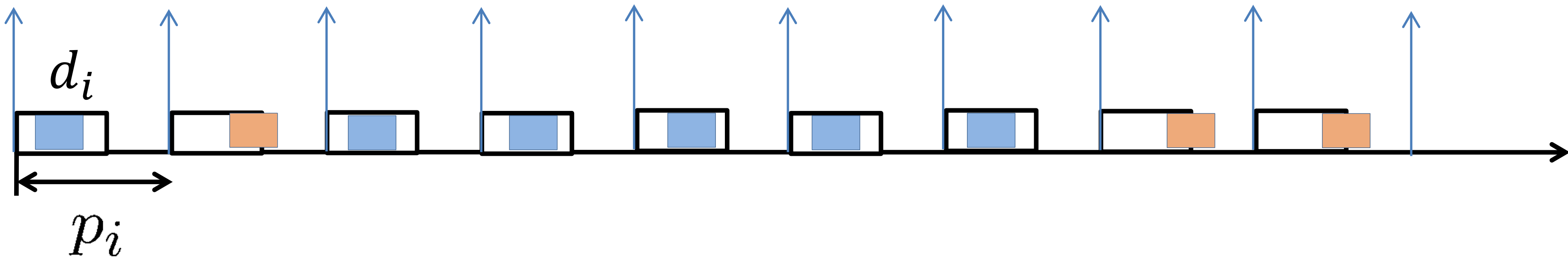


Periodic tasks



Hard deadlines

Periodic tasks



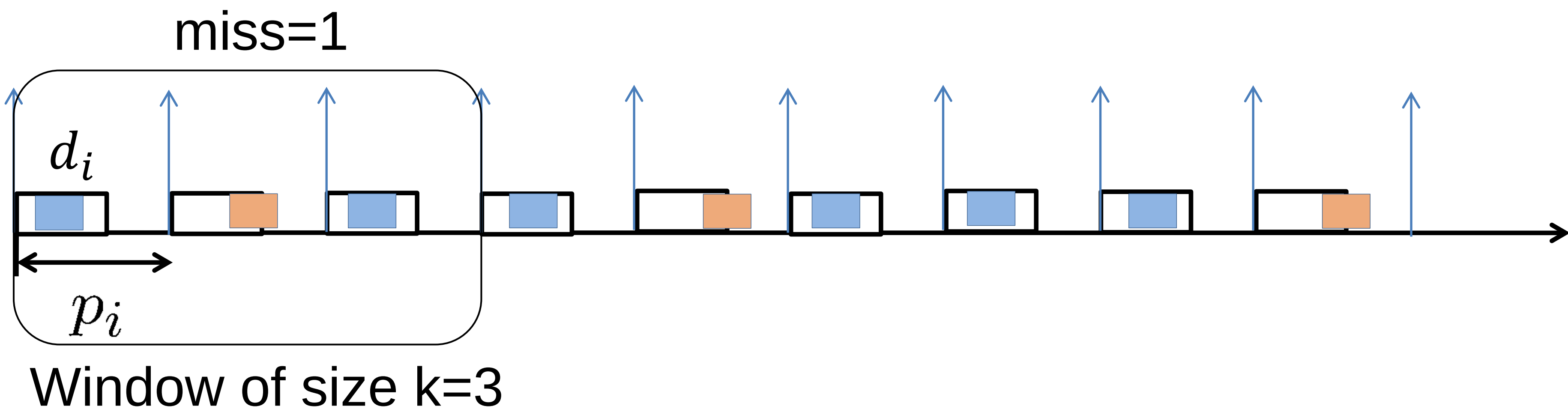
Some jobs miss deadlines -- no restriction on order of misses: soft deadline

Firm deadlines

(m,k)-firm deadline: A periodic task is said to have an (m,k)-firm guarantee if it is adequate to meet the deadlines of m out of k consecutive instances of the task (jobs), where $m \leq k$.

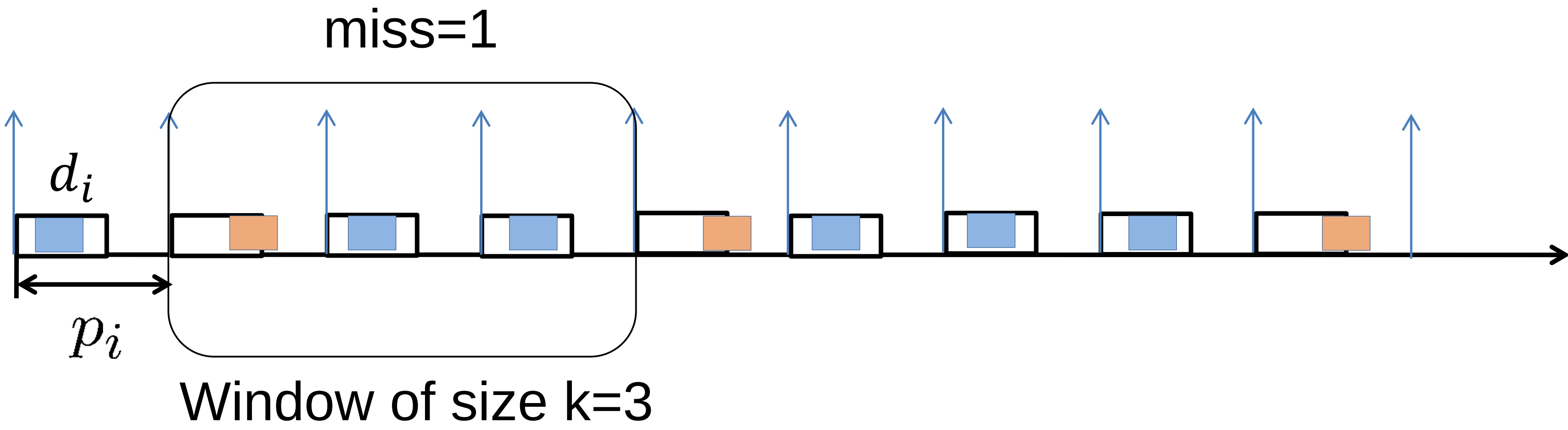
(m,k)-firm deadline

A periodic task is said to have an (m,k)-firm guarantee if it is adequate to meet the deadlines of m out of k consecutive instances of the task (jobs), where m ≤ k.



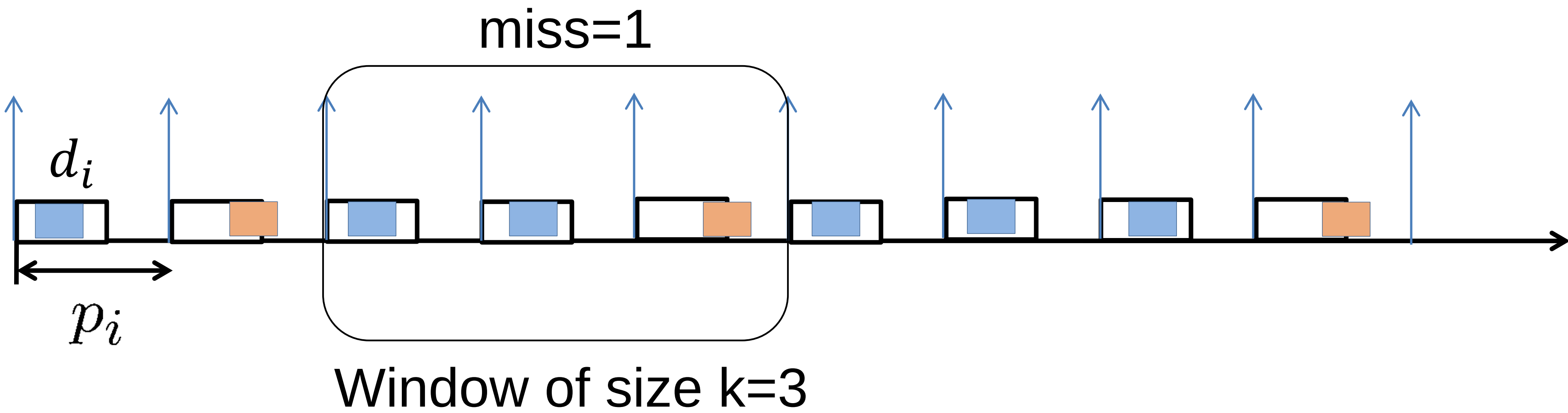
(m,k)-firm deadline

A periodic task is said to have an (m,k)-firm guarantee if it is adequate to meet the deadlines of m out of k consecutive instances of the task (jobs), where m ≤ k.



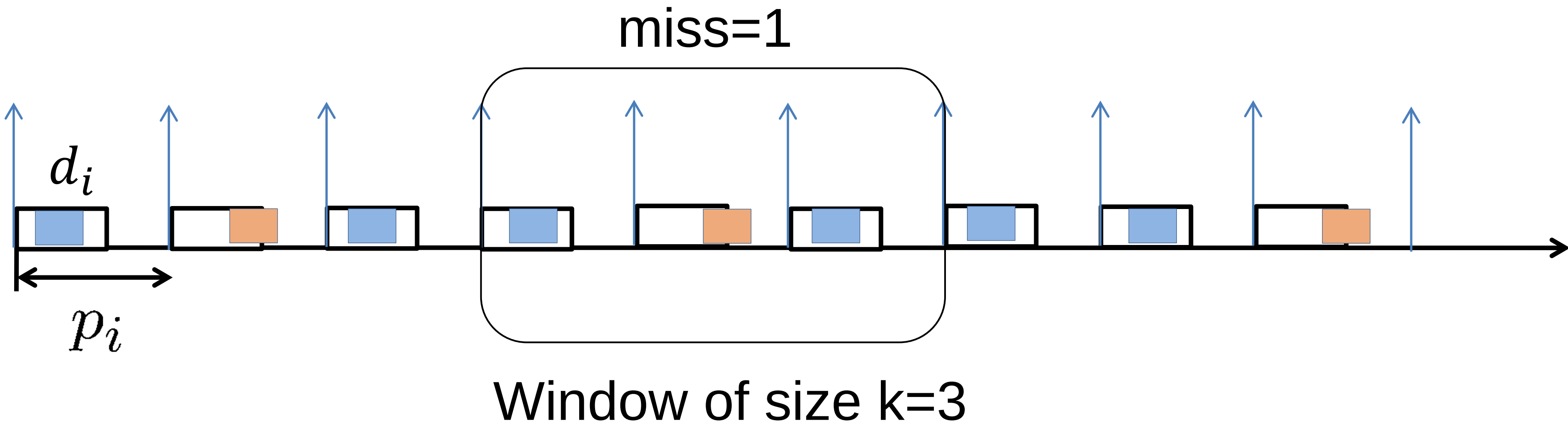
(m,k)-firm deadline

A periodic task is said to have an (m,k)-firm guarantee if it is adequate to meet the deadlines of m out of k consecutive instances of the task (jobs), where m ≤ k.



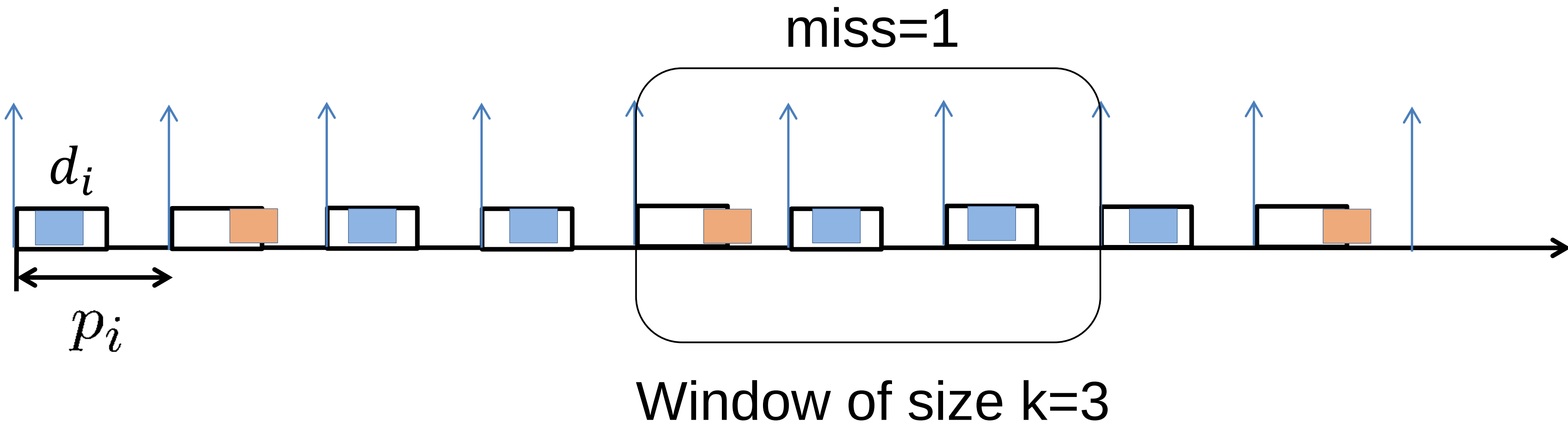
(m,k)-firm deadline

A periodic task is said to have an (m,k)-firm guarantee if it is adequate to meet the deadlines of m out of k consecutive instances of the task (jobs), where m ≤ k.



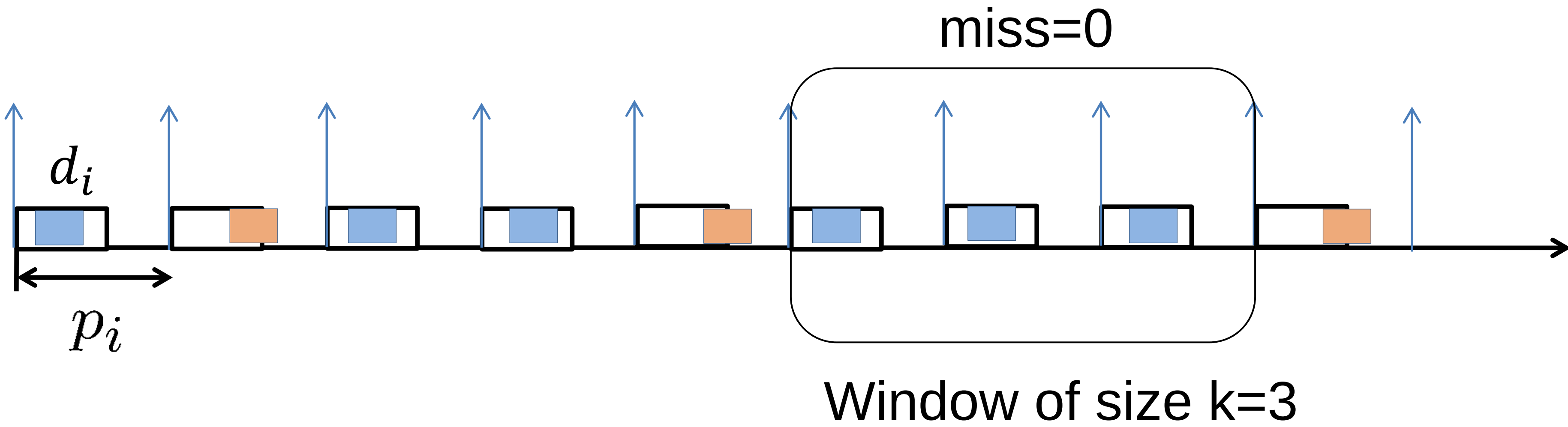
(m,k)-firm deadline

A periodic task is said to have an (m,k)-firm guarantee if it is adequate to meet the deadlines of m out of k consecutive instances of the task (jobs), where m ≤ k.



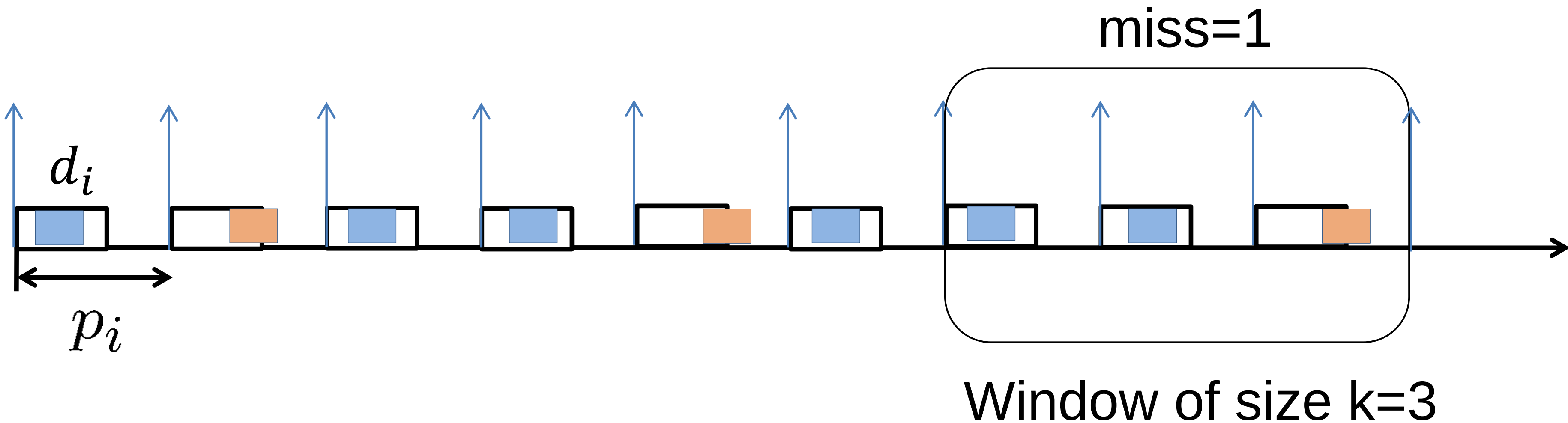
(m,k)-firm deadline

A periodic task is said to have an (m,k)-firm guarantee if it is adequate to meet the deadlines of m out of k consecutive instances of the task (jobs), where m ≤ k.



(m,k)-firm deadline

A periodic task is said to have an (m,k)-firm guarantee if it is adequate to meet the deadlines of m out of k consecutive instances of the task (jobs), where m ≤ k.



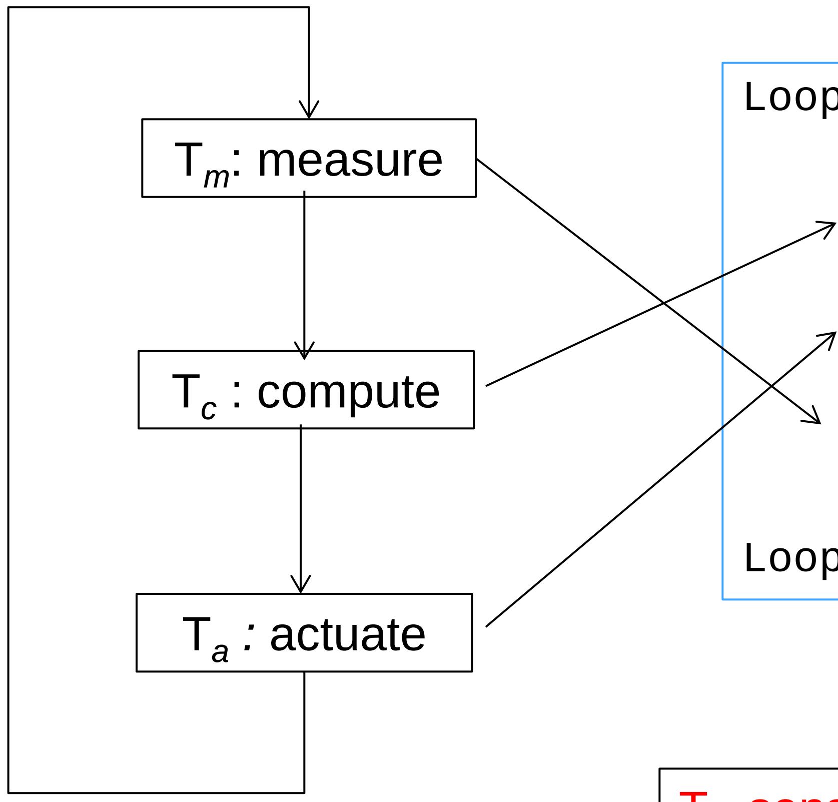
m=1 →(1,3)-firm deadline

Outline

- Task model and deadlines in feedback control
- Typical application scenario
- Implication of an firm-deadline
- Analytical outlook
- Some results

...control task model and deadlines...

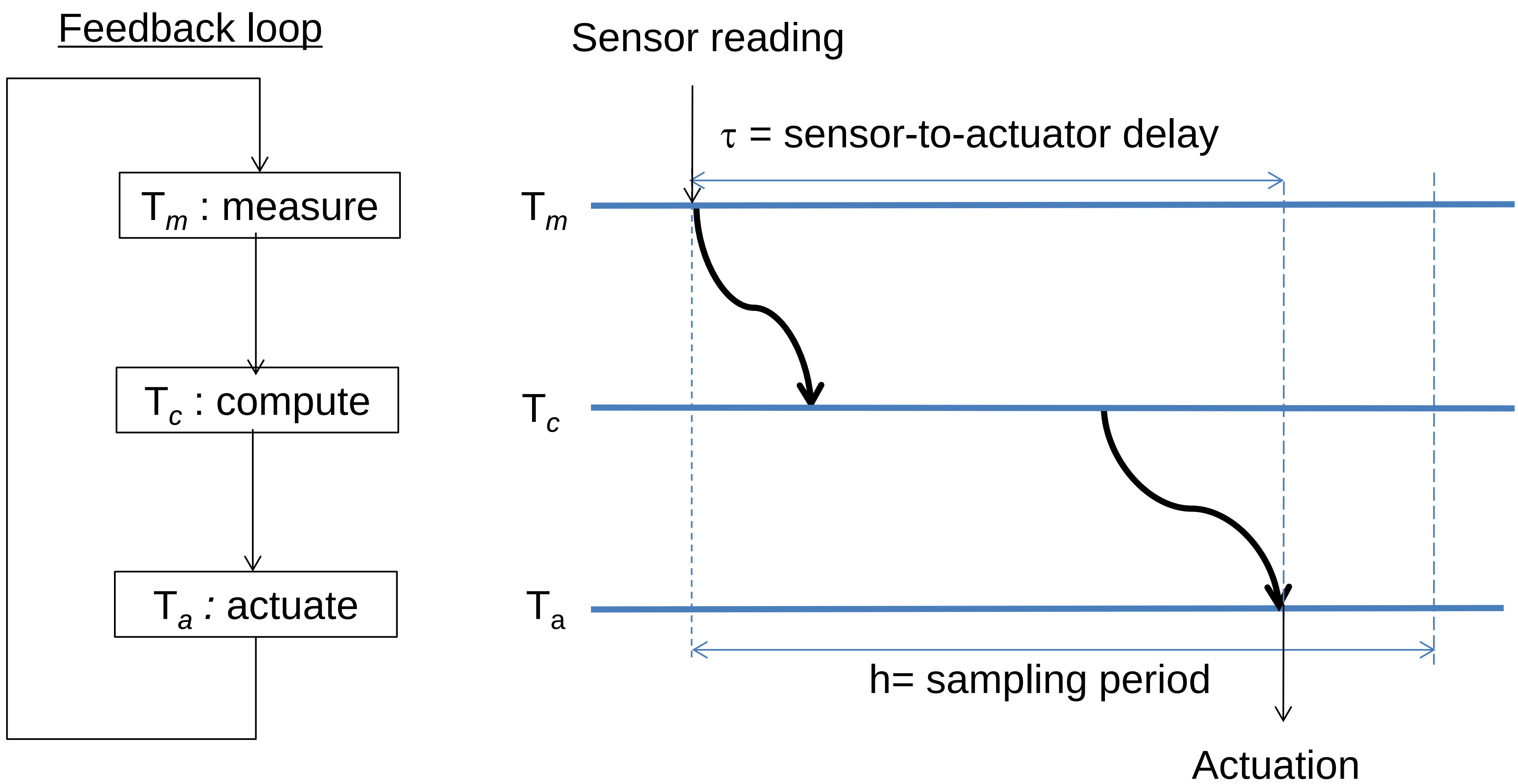
Feedback loop




```
Loop start

    u = K*[x1(i);x2(i)] + r*F;

    xkp1 = A*[x1(i);x2(i)]+  B*u;

    x1(i+1) = xkp1(1);
    x2(i+1) = xkp1(2);

Loop repeats
```

T_m sensor task
T_c controller task
T_a actuator task

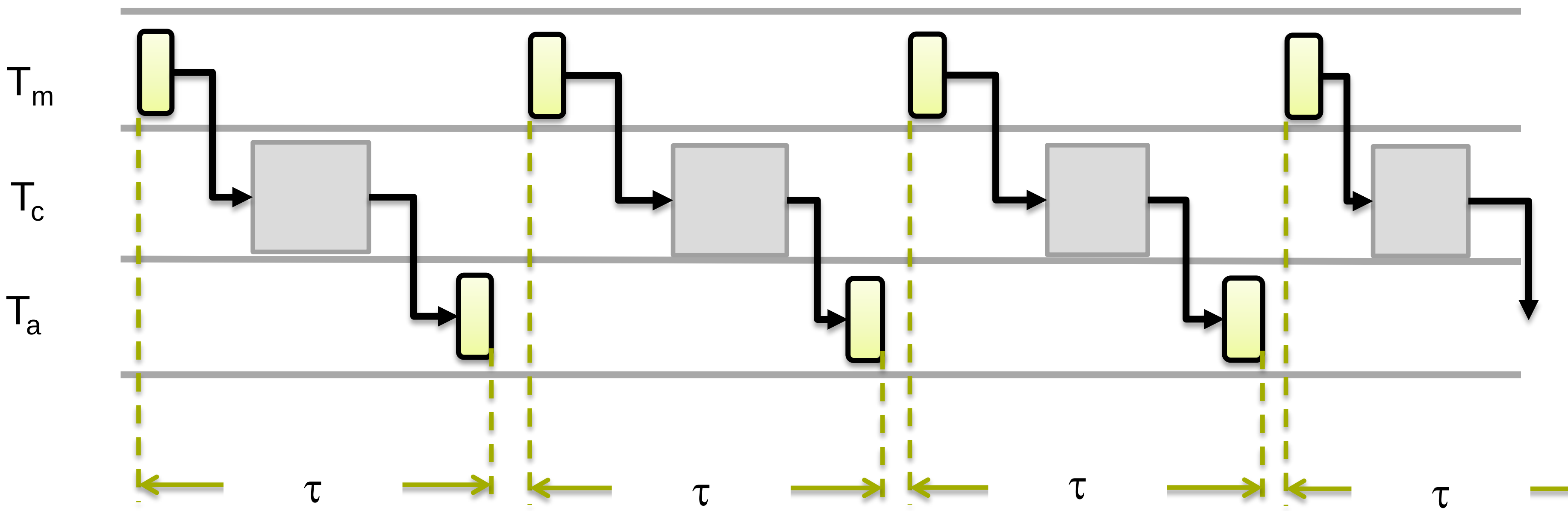
Control loop



Ideal design assumes: $\tau = 0$ or $\tau << h$

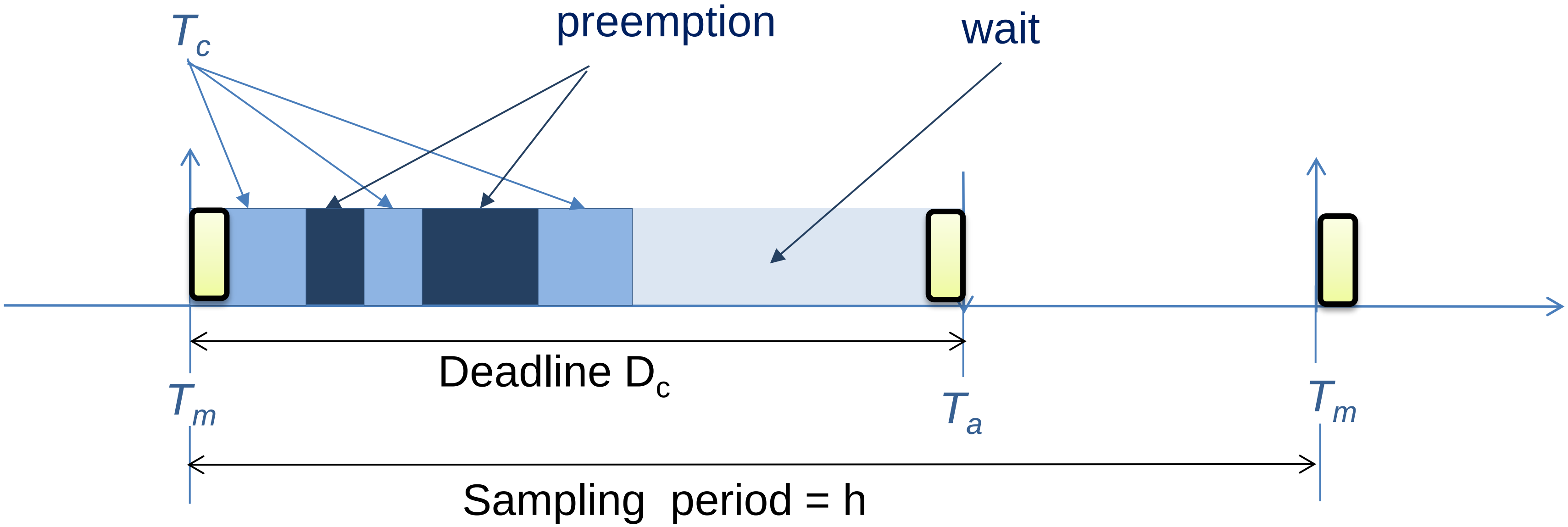
Control task triggering

- In general, T_m and T_a tasks consume negligible computational time and are time-triggered
- T_c needs finite computation time and can be preemptive
- When multiple tasks are running on a processor, T_c can be preempted

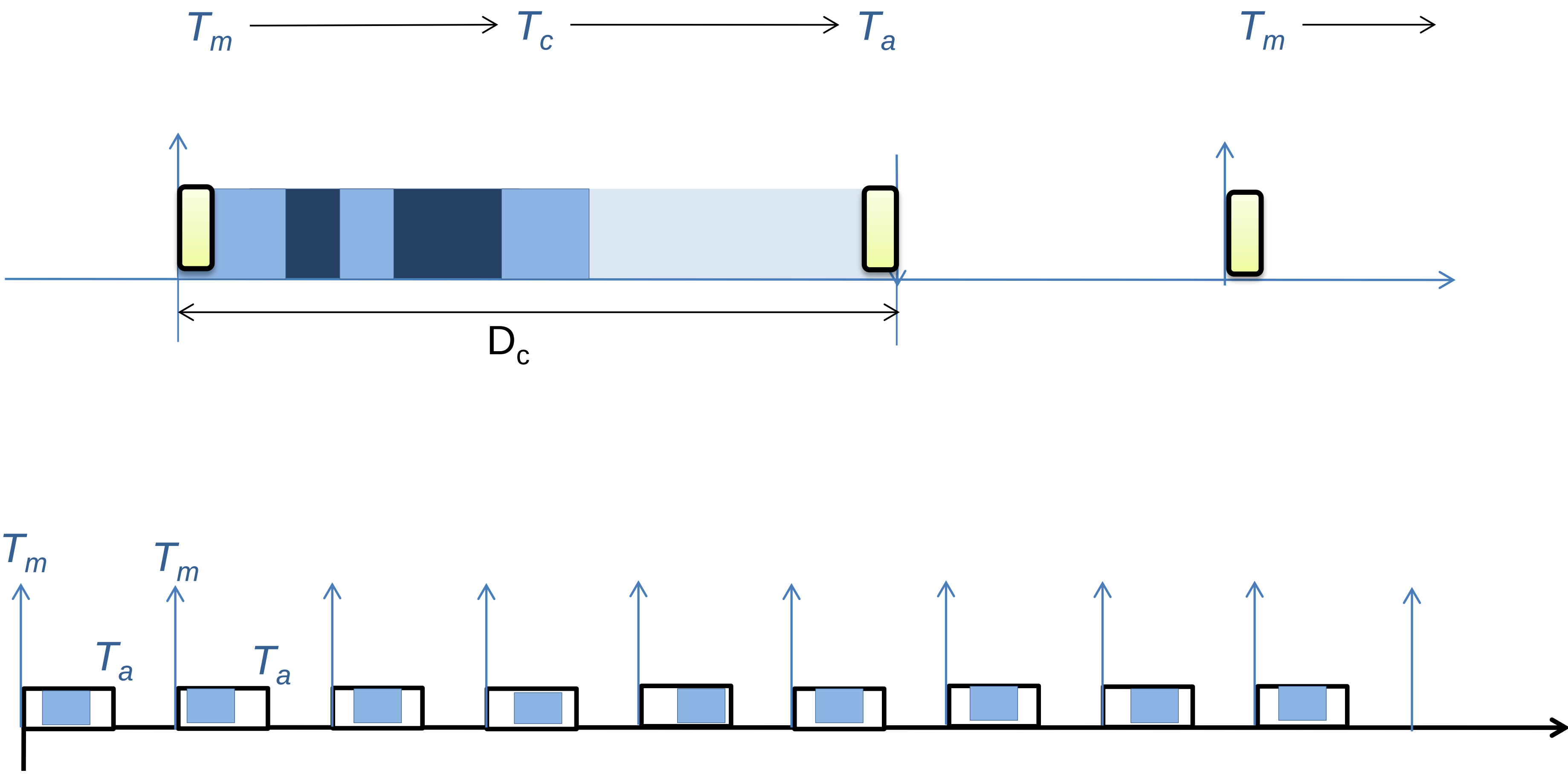


Sensor-to-actuator delay: τ

Deadline for control tasks

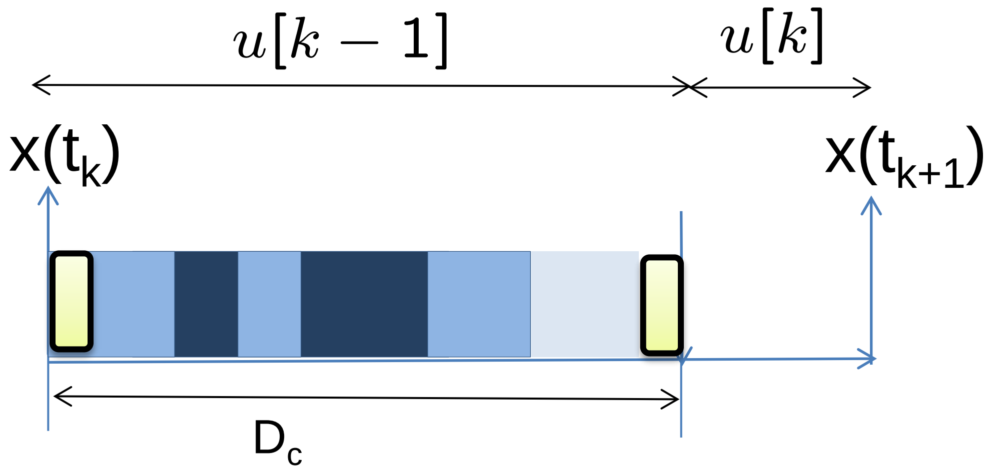


sensor-to-actuator delay $\tau = D_c$



p_i=h d_i=D_c

Constant delay: sampled-data model



$$x[k+1] = \emptyset x[k] + B_1(D_c)u[k-1] + B_0(D_c)u[k]$$
$$y[k] = Cx[k]$$

...will be illustrated in Part 4

...automotive application scenario...

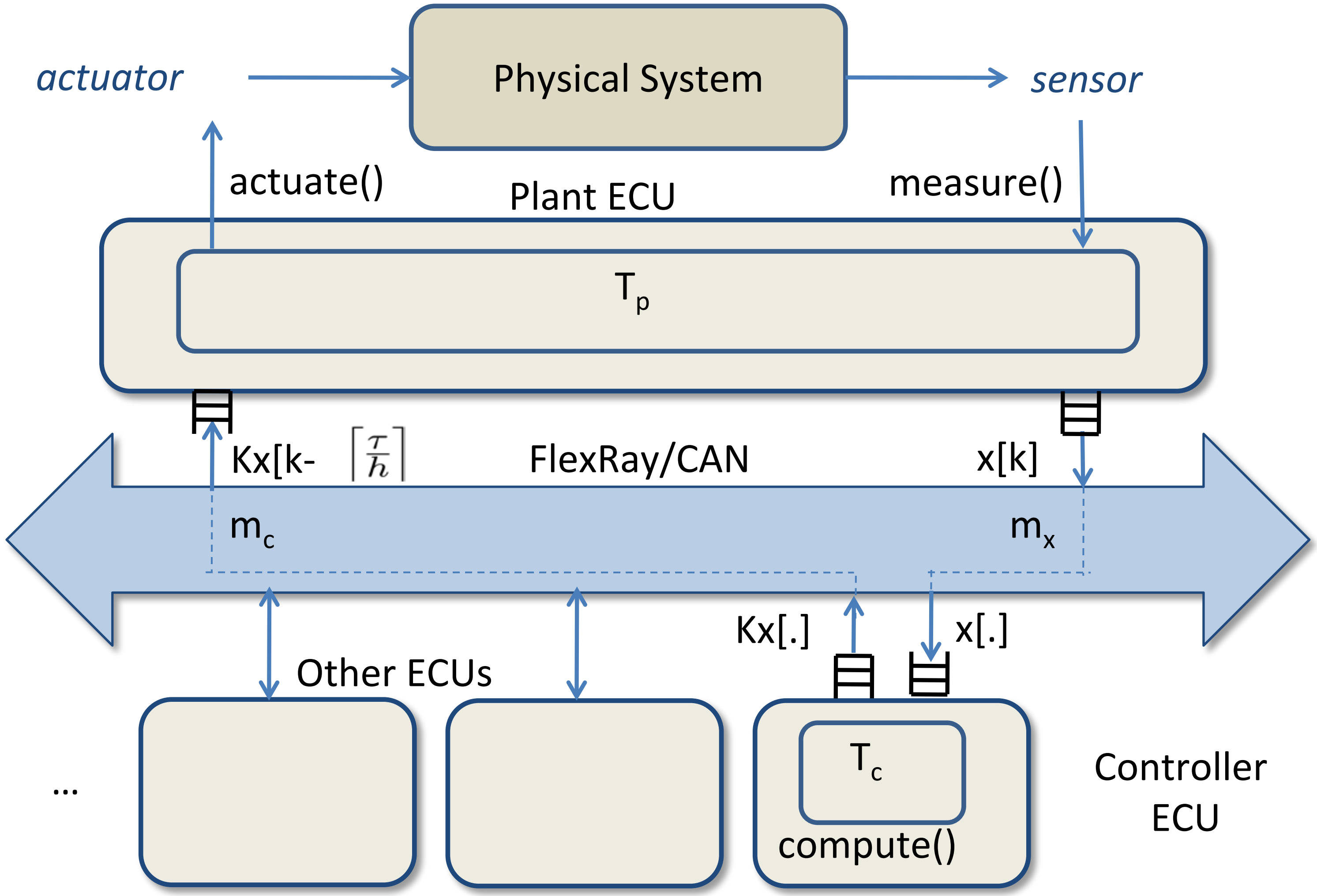
Based on the article:

Dip Goswami, Samarjit Chakraborty, Reinhard Schneider

Relaxing Signal Delay Constraints in Distributed Embedded Controllers

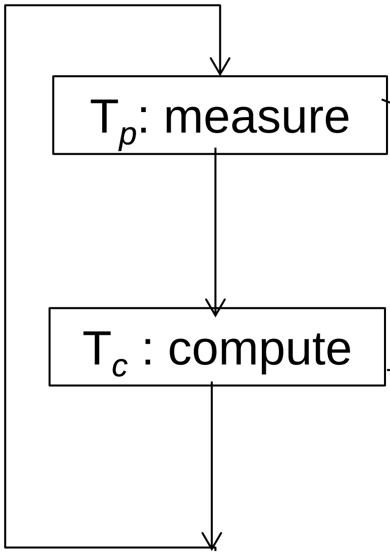
IEEE Trans. On Control Systems Technology, 2014

Distributed automotive control loops



Task partitioning

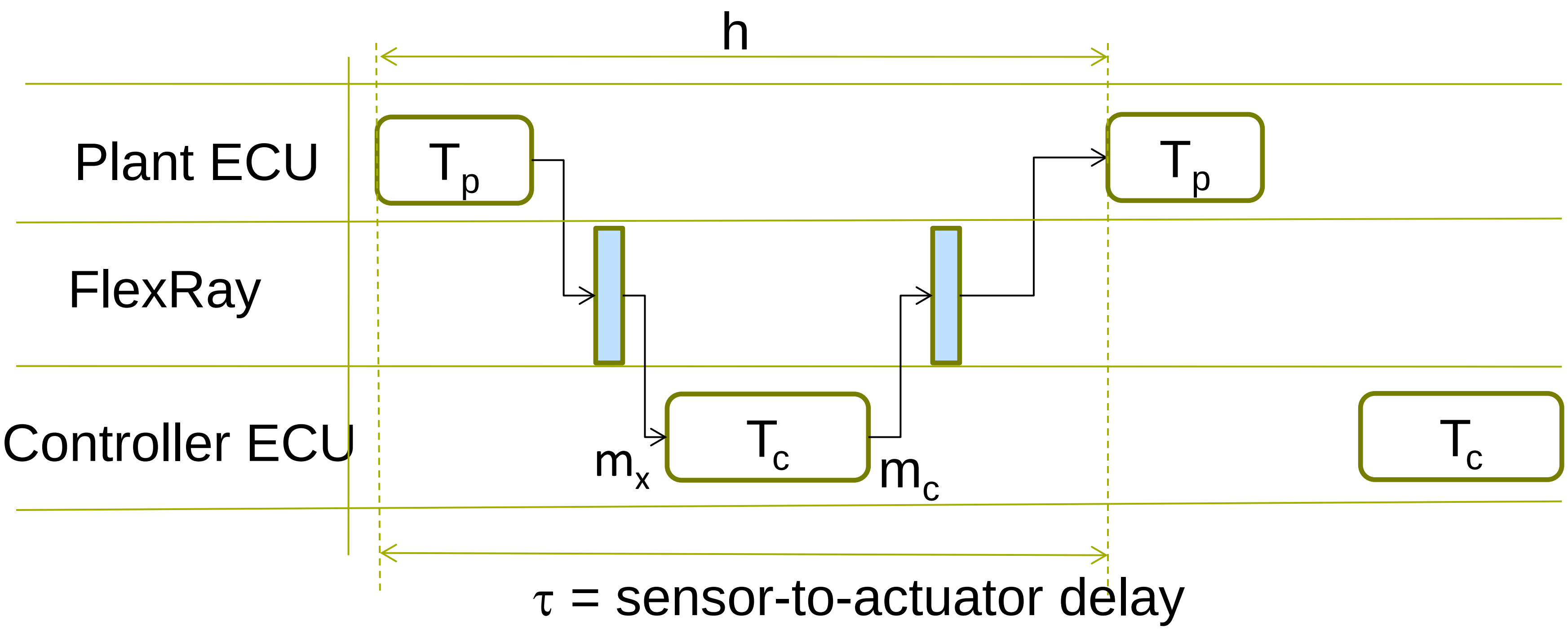
Feedback loop




```
Loop start

    xkp1 = A*[x1(i);x2(i)]+  B*u;
    x1(i+1) = xkp1(1);
    x2(i+1) = xkp1(2);

    u = K*[x1(i+1);x2(i+1)] + r*F;

Loop repeats
```

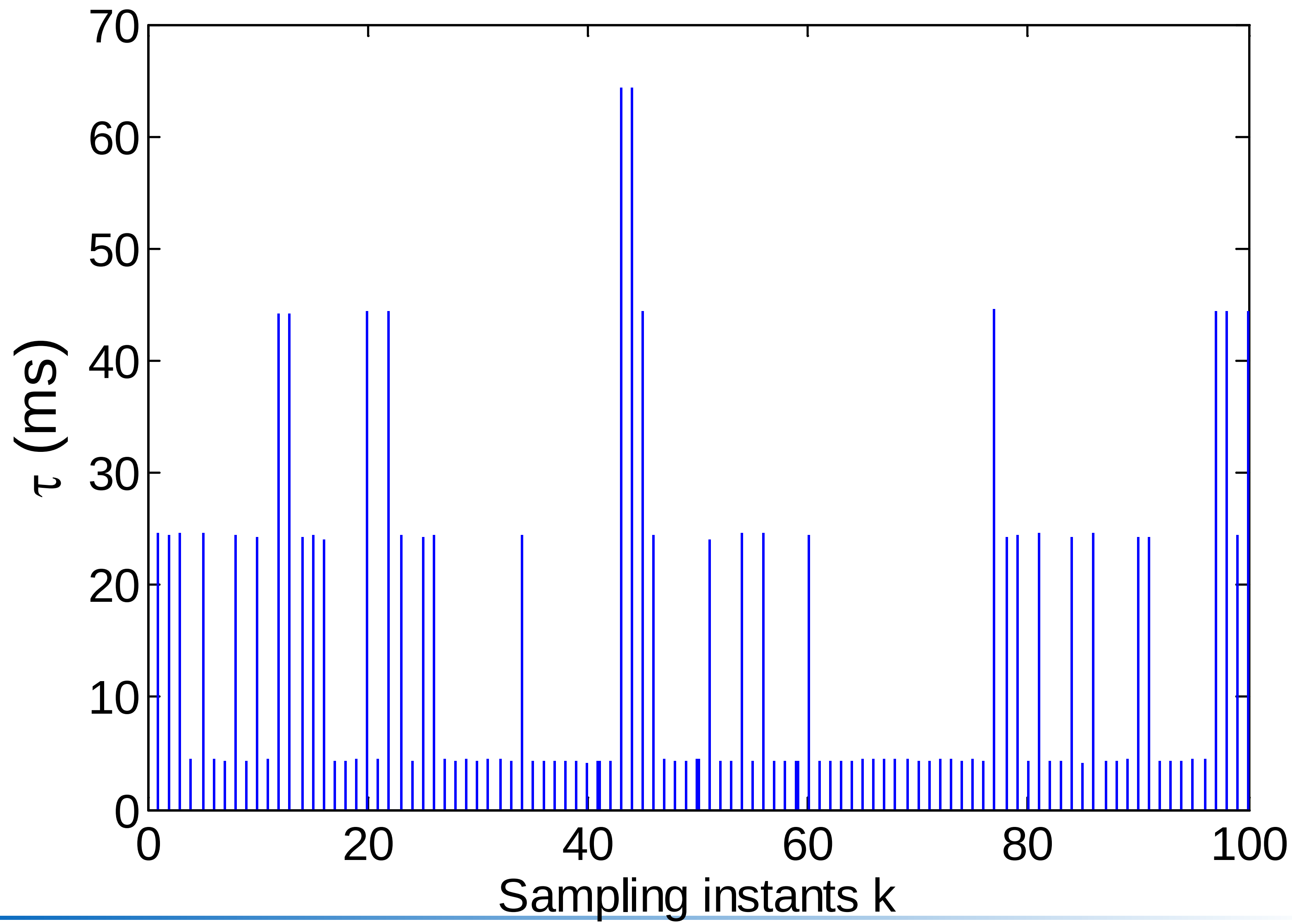
T_p plant task
T_c controller task

Timing diagram and deadline

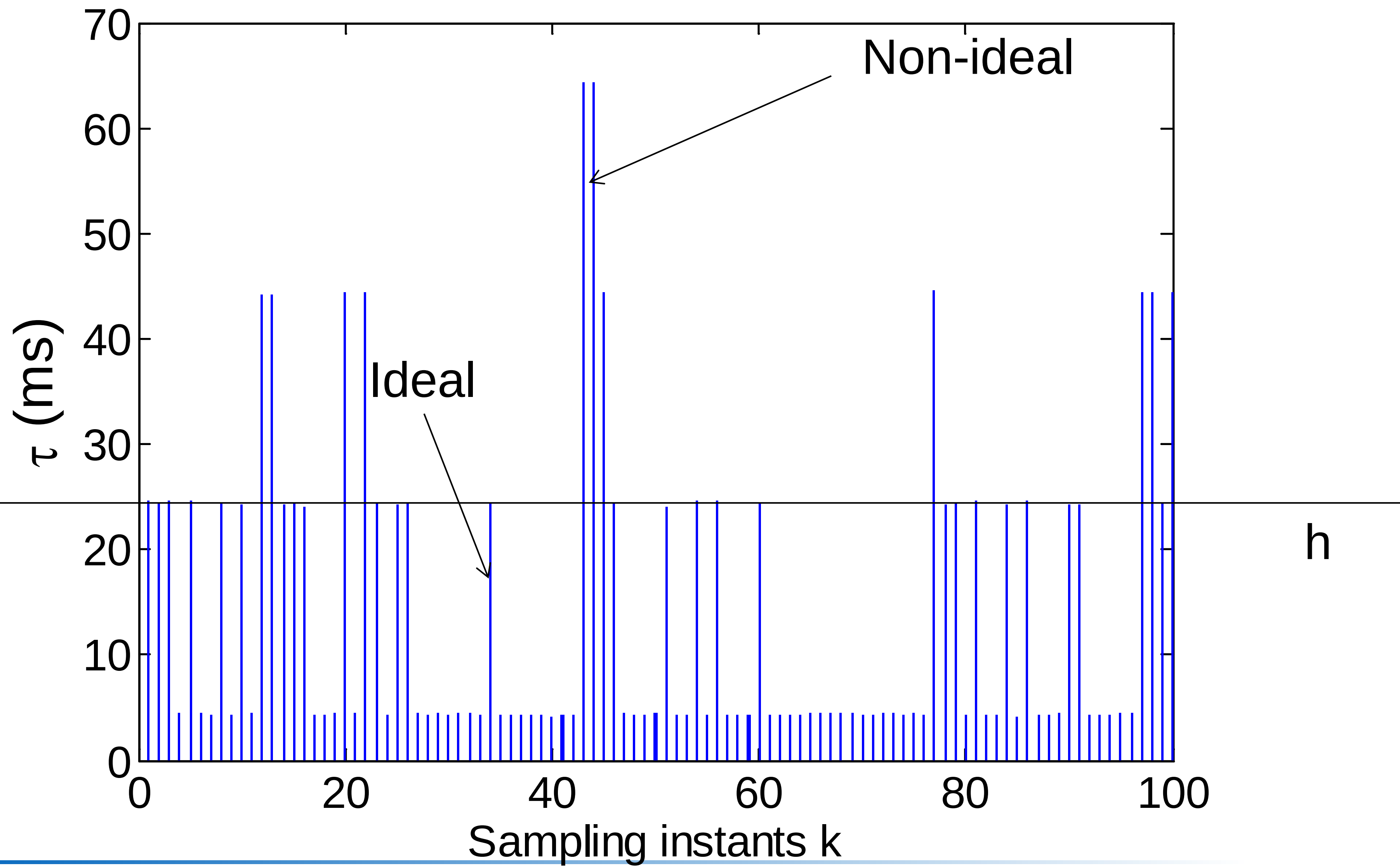


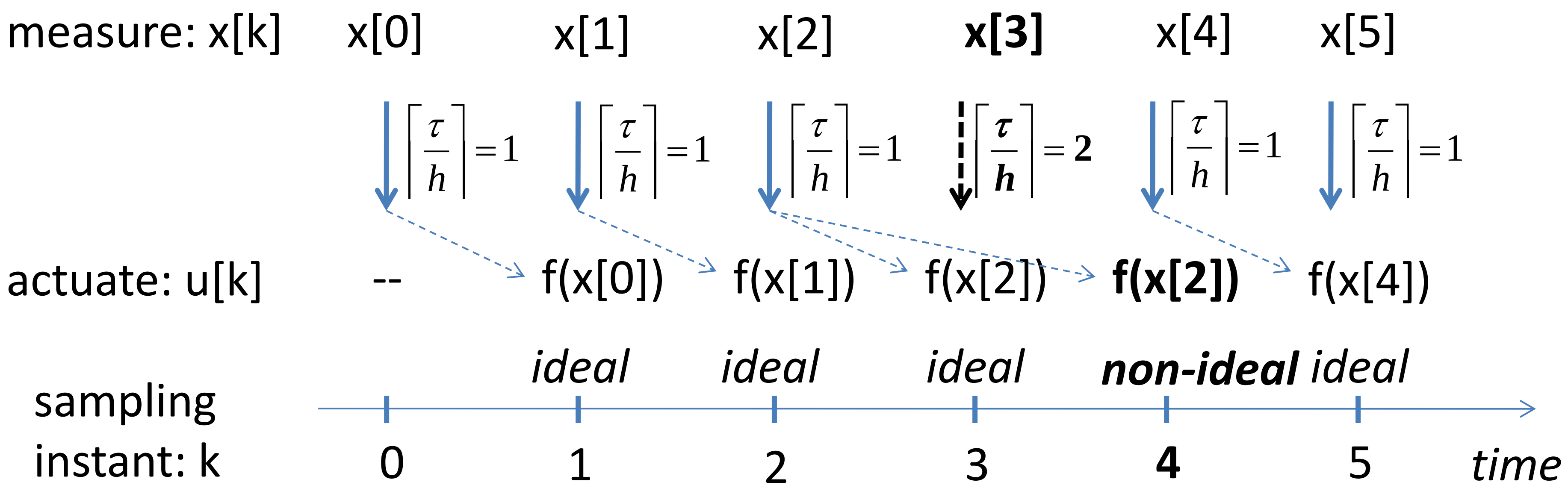
Deadline ≈ sampling period

FlexRay: typical delay profile



Delay based classification




measure: x[k]
x[0]
x[1]
x[2]
x[3]
x[4]
x[5]
⌈τ/h⌉ = 1
⌈τ/h⌉ = 1
⌈τ/h⌉ = 1
⌈τ/h⌉ = 2
⌈τ/h⌉ = 1
⌈τ/h⌉ = 1
actuate: u[k]
--
f(x[0])
f(x[1])
f(x[2])
f(x[2])
f(x[4])
ideal
ideal
ideal
non-ideal
ideal
sampling instant: k
0
1
2
3
4
5
time

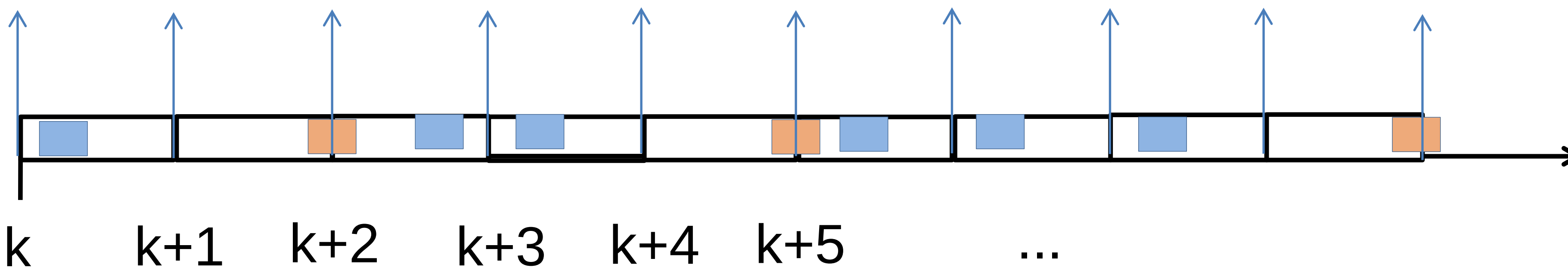
Control scheme

$$u[k] = Kx[k-1] + F_1 r \quad \forall \text{ideal samples}$$
$$= F_2 r \quad \forall \text{non-ideal samples}$$

$$x[k+1] = Ax[k] + Bu[k]$$

$$\text{Ideal: } x[k+1] = Ax[k] + BKx[k-1] + BF_1 r \quad \Rightarrow z[k+1] = A_{cl} z[k]$$
$$\text{Non-ideal: } x[k+1] = Ax[k] + BF_2 r \quad \Rightarrow z[k+1] = A_o z[k]$$

Back to firm-deadline setting



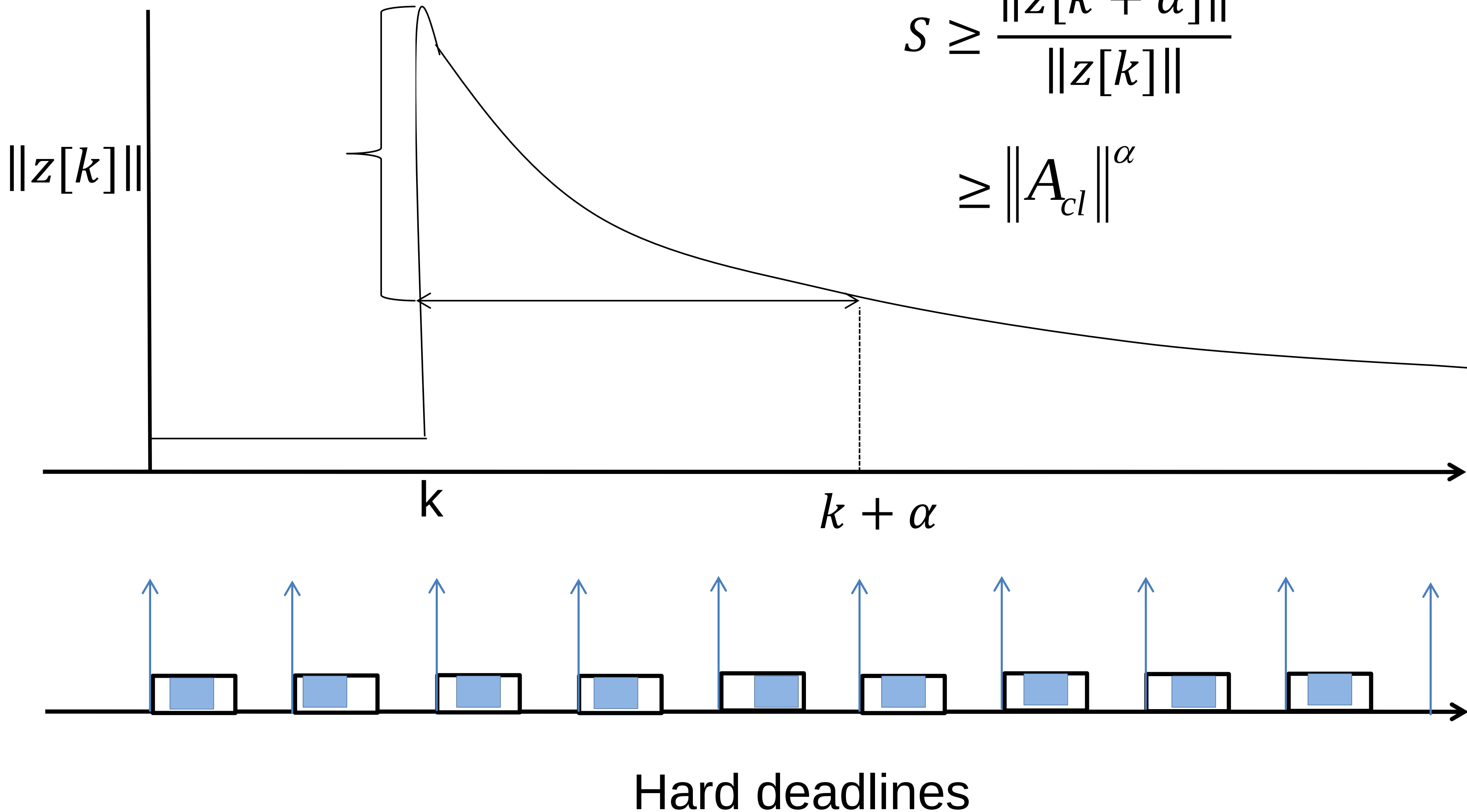
$$z[k+1] = A_{cl}\, z[k]$$
$$z[k+2] = A_o\, z[k+1] = A_o\ A_{cl} z[k]$$
$$z[k+3] = A_{cl}\, z[k+2] = A_{cl} A_o\ A_{cl} z[k]$$
$$z[k+4] = A_{cl}\, z[k+3] = A_{cl} A_{cl} A_o\, A_{cl} z[k]$$
$$z[k+5] = A_o\, z[k+4] = A_o\, A_{cl} A_{cl} A_o\, A_{cl} z[k]$$

...

Performance: fast disturbance rejection

$$S \geq \frac{\|z[k+\alpha]\|}{\|z[k]\|}$$

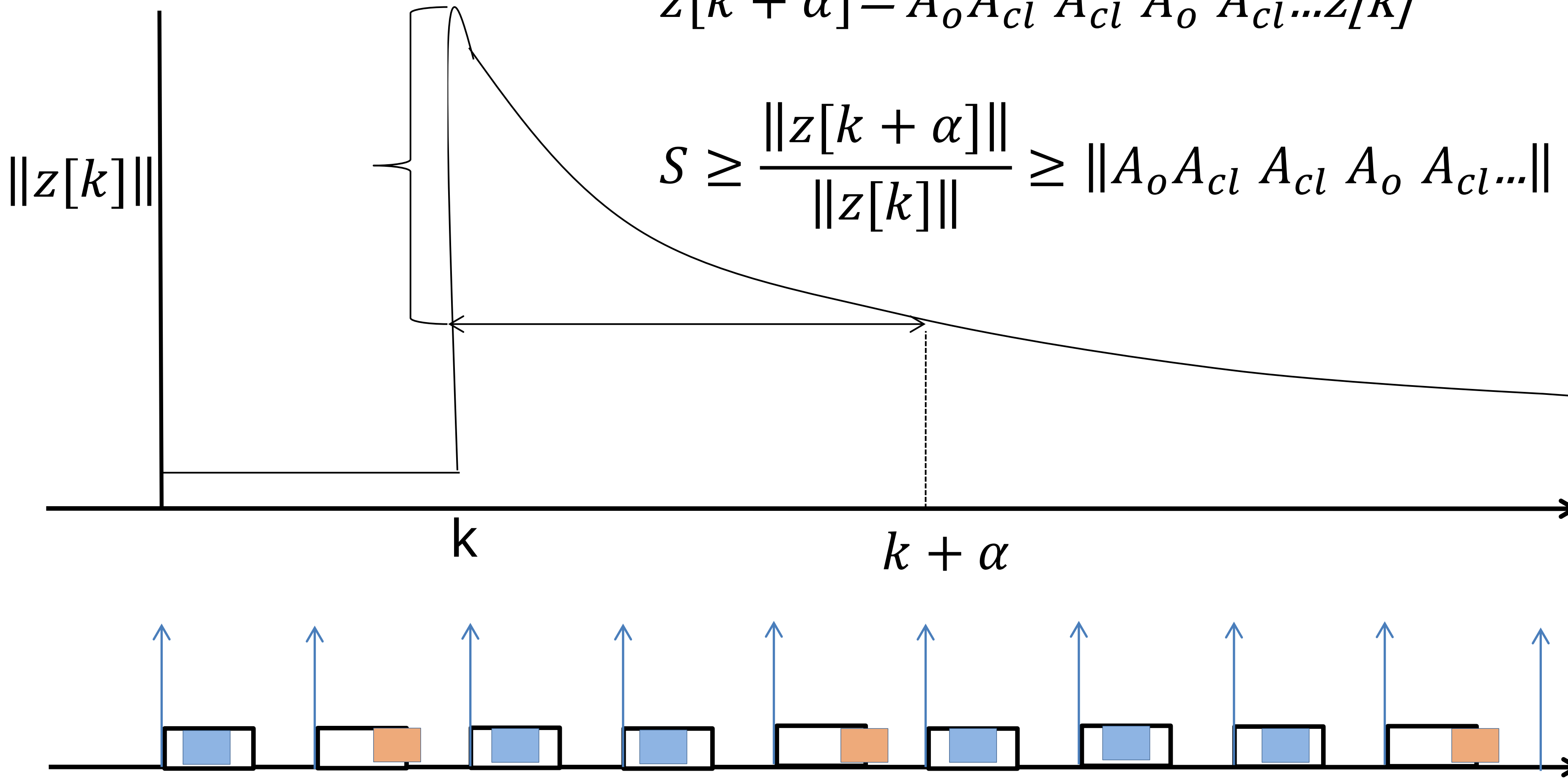
$$\geq \left\|A_{cl}\right\|^{\alpha}$$



Performance and deadline miss

$$z[k+\alpha] = A_o A_{cl}\ A_{cl}\ A_o\ A_{cl} \ldots z[k]$$

$$S \geq \frac{\|z[k+\alpha]\|}{\|z[k]\|} \geq \|A_o A_{cl}\ A_{cl}\ A_o\ A_{cl} \ldots\|$$



Firm deadlines

Simplified special case

- $S \geq \|A_o A_{cl}\ A_{cl}\ A_o\ A_{cl} \ldots\|$
- For any matrix A, the following property holds

$$\left\|A^k\right\| \leq c\gamma^k$$

where c and γ are constants

- $\|A_{cl}\|^k \leq c_1{\gamma_1}^k$
- $\|A_o\|^k \leq c_2{\gamma_2}^k$
- Special case of (m,k)-firmness: m=1

$\longrightarrow$ 1 deadline miss is allowed by within α samples

$$S \geq (c_1)^2{\gamma_1}^{\alpha-1}\ c_2\gamma_2$$

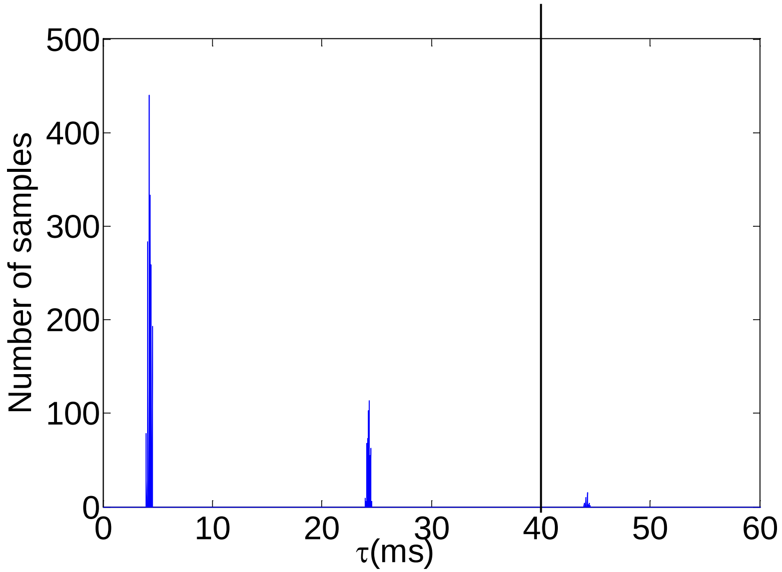
- Compute minimum α

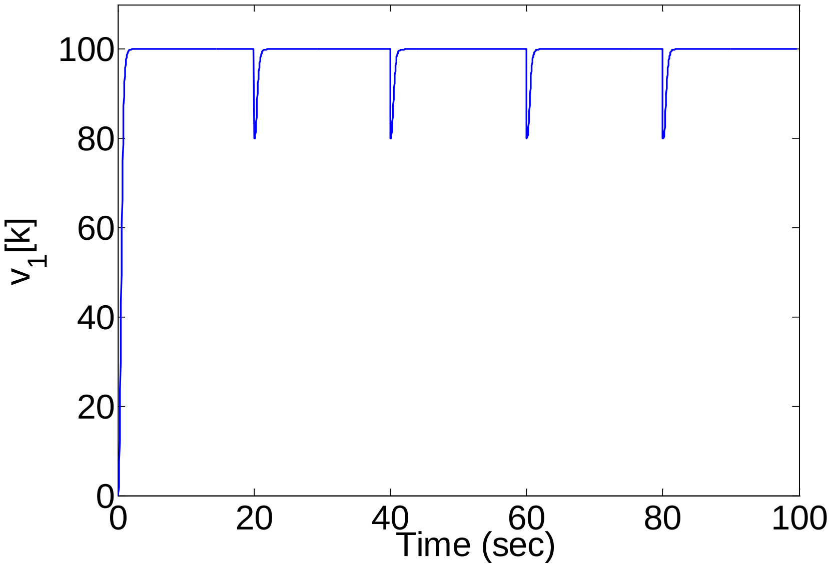
Some results

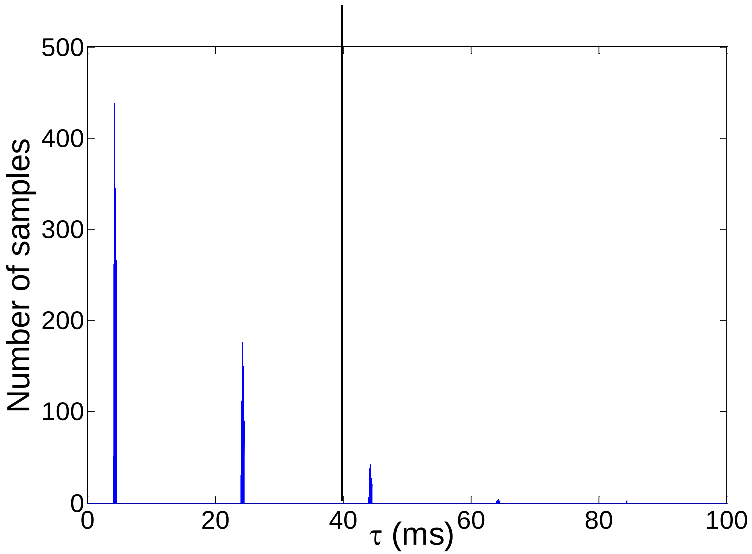
- Cruise control systems:
  - Sampling period = 40ms
  - Performance requirement:
    - Change in reference speed should be adapted asap
    - 95% of the disturbance must be rejected in 5 sec
    - Implies α=125

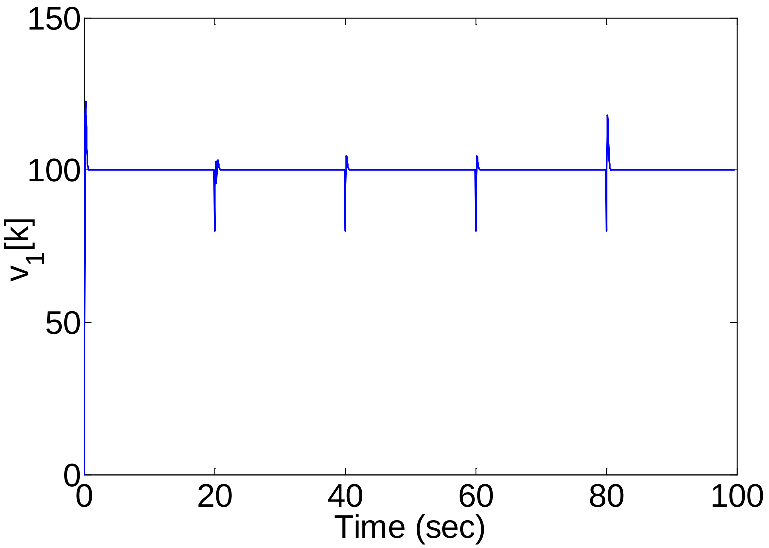
- Allowed deadline miss: 7 out of 125

Simulation based results









Summary

- ✓ Task model and deadlines in feedback control
- ✓ Typical application scenario
- ✓ Implication of an firm-deadline
- ✓ Analytical outlook
- ✓ Some results