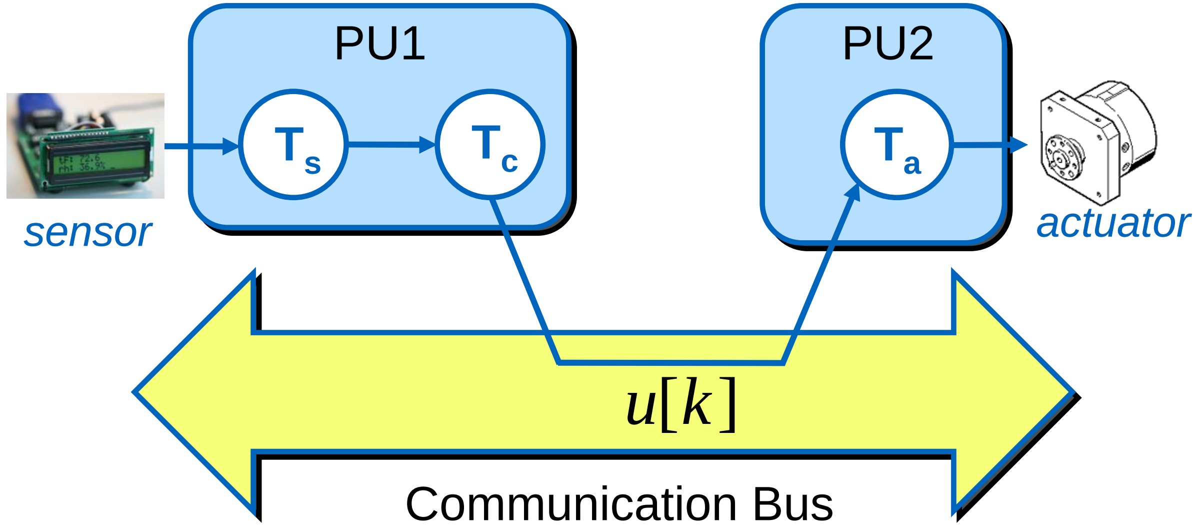
Co-Design of Cyber-Physical Systems via Controllers with Flexible Delay Constraints

Dip Goswami, Reinhard Schneider and Samarjit Chakraborty

TU Munich, Germany

Cyber-Physical Control Application



Actuator Task T_a: $x[k+1] = Ax[k] + Bu[k]$

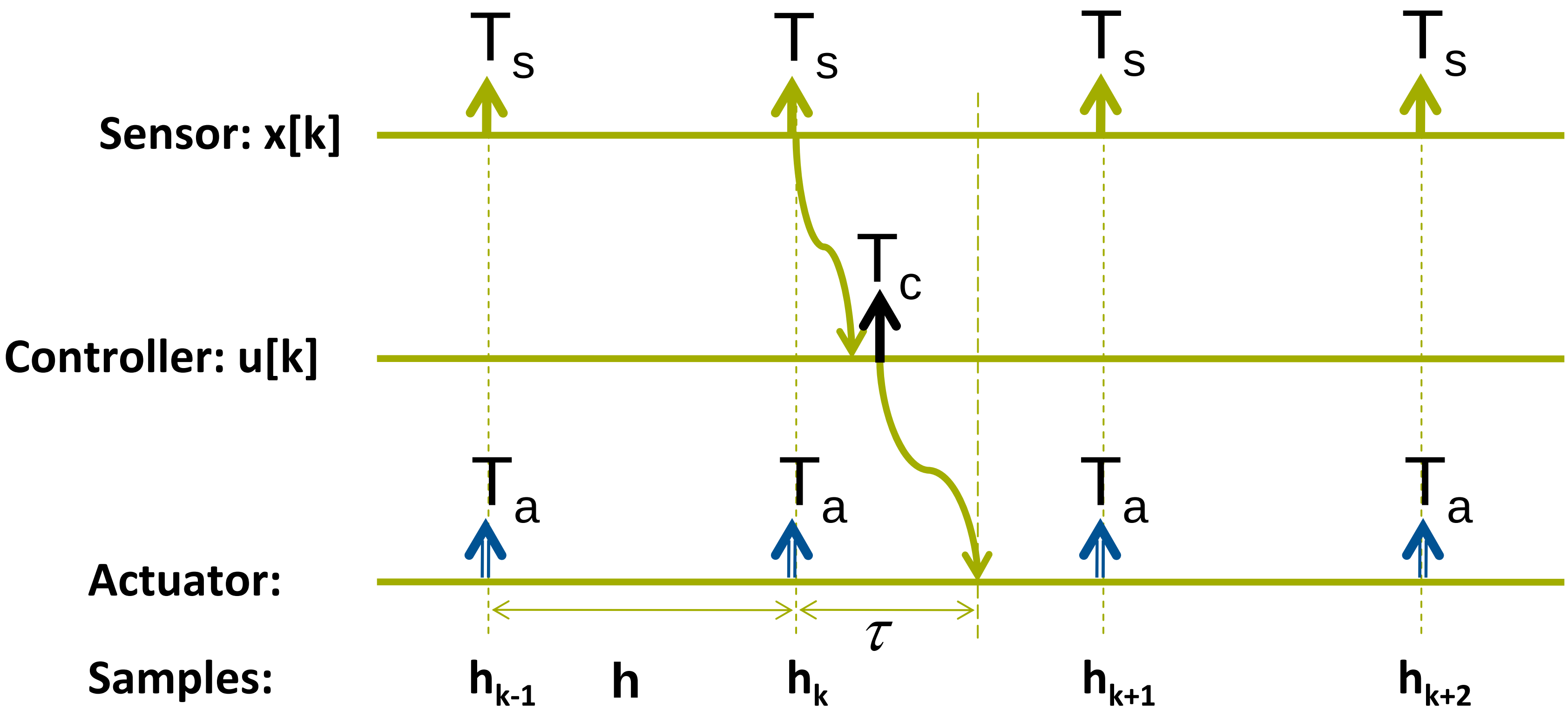
Controller task T_c: $u[k] = Kx[k-\delta]$

K= State feedback gains

Sensor Task T_s: $x[k]$

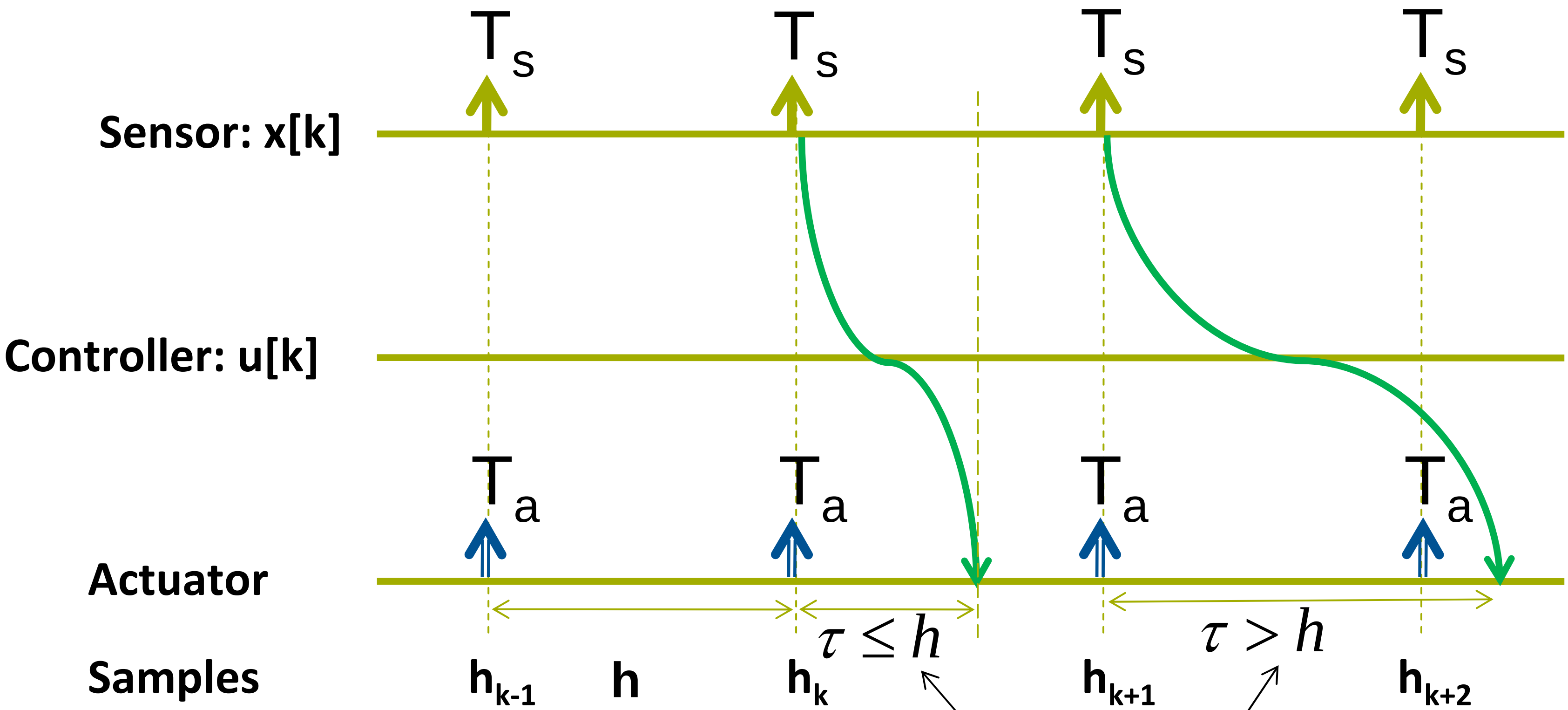
Communication delay reflected in the feedback signal

Task Triggering: Timing Diagram



Communication delay (samples): $$\delta = \left\lceil \frac{\tau}{h} \right\rceil$$

Stable and Unstable Samples



Stable Sample: $\delta = 1$

Unstable Sample: $\delta > 1$

Overall Control Scheme

$$\text{Proposed Control Scheme:}$$

$$u[k] = \begin{cases} Kx[k-1] \quad if \quad \left\lceil \frac{\tau}{h} \right\rceil = 1 \quad or \quad stable\ samples \\ 0 \quad if \quad \left\lceil \frac{\tau}{h} \right\rceil \neq 1, \ or \quad unstable \quad samples \end{cases}$$

- The controller is designed for one sample feedback delay
- When the feedback delay is more than one sample, no control input is provided.

Main result: Asymptotic stability

Condition for Asymptotic Stability:

$$\frac{\text{number of stable samples}}{\text{number of unstable samples}} \geq \mu$$

where μ is a positive integer.

Therefore, we allow a fraction $\frac{1}{\mu + 1}$ of all samples to be *unstable* or violate the deadlines with guaranteed asymptotic stability!

Co-design guidelines

1. Choose controller gains K such that $u[k] = Kx[k-1]$ achieves asymptotic stability, i.e, $x[k] \rightarrow 0$ as $k \rightarrow \infty$

2. Choose the communication bus parameters such that

$$\frac{\text{number of stable samples}}{\text{number of unstable samples}} \geq \mu$$

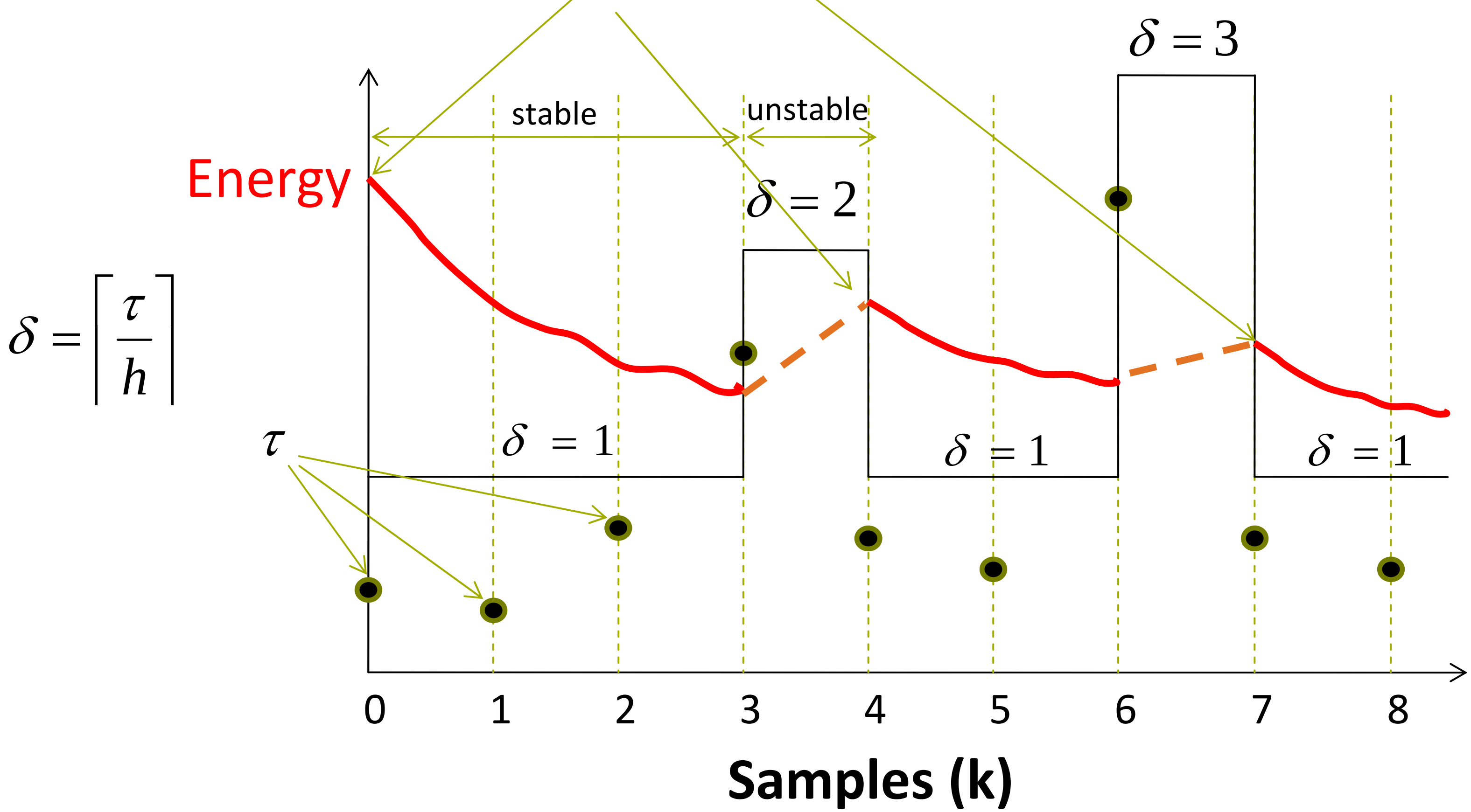
Significance of our results

- There is no need any worst-case timing analysis of the communication bus. Which simplifies the communication schedule synthesis.
- Applicable to any plant-controller setup. Depending on various factors μ might change.
- The presented approach can be utilized on top of the traditional packet dropout approaches. It will certainly increase the design robustness.
- No oversampling – communication bandwidth utilization is not compromised.

Intuitive Proof

Stability of the control system is based on the fact that the system ***energy-like function*** decreases with time

Energy decreases even if there is local increase in energy because of unstable samples



Discussions: Existence of μ

- Does μ exist for any discrete-time system?
  - Yes.

- Can μ be computed analytically?
  - Yes.

Proof Outline

Stable samples: $x[k+1] = A_{cl}x[k]$

Unstable samples: $x[k+1] = A_a x[k]$

$x[k+1] = A_{cl}x[k]$

$\equiv x[k] \rightarrow 0$ as $k \rightarrow \infty$

$\equiv \lim_{k\rightarrow\infty} A_{cl}^k = 0$

Hence, $A_{cl}^{\mu} \approx 0$ for large μ.

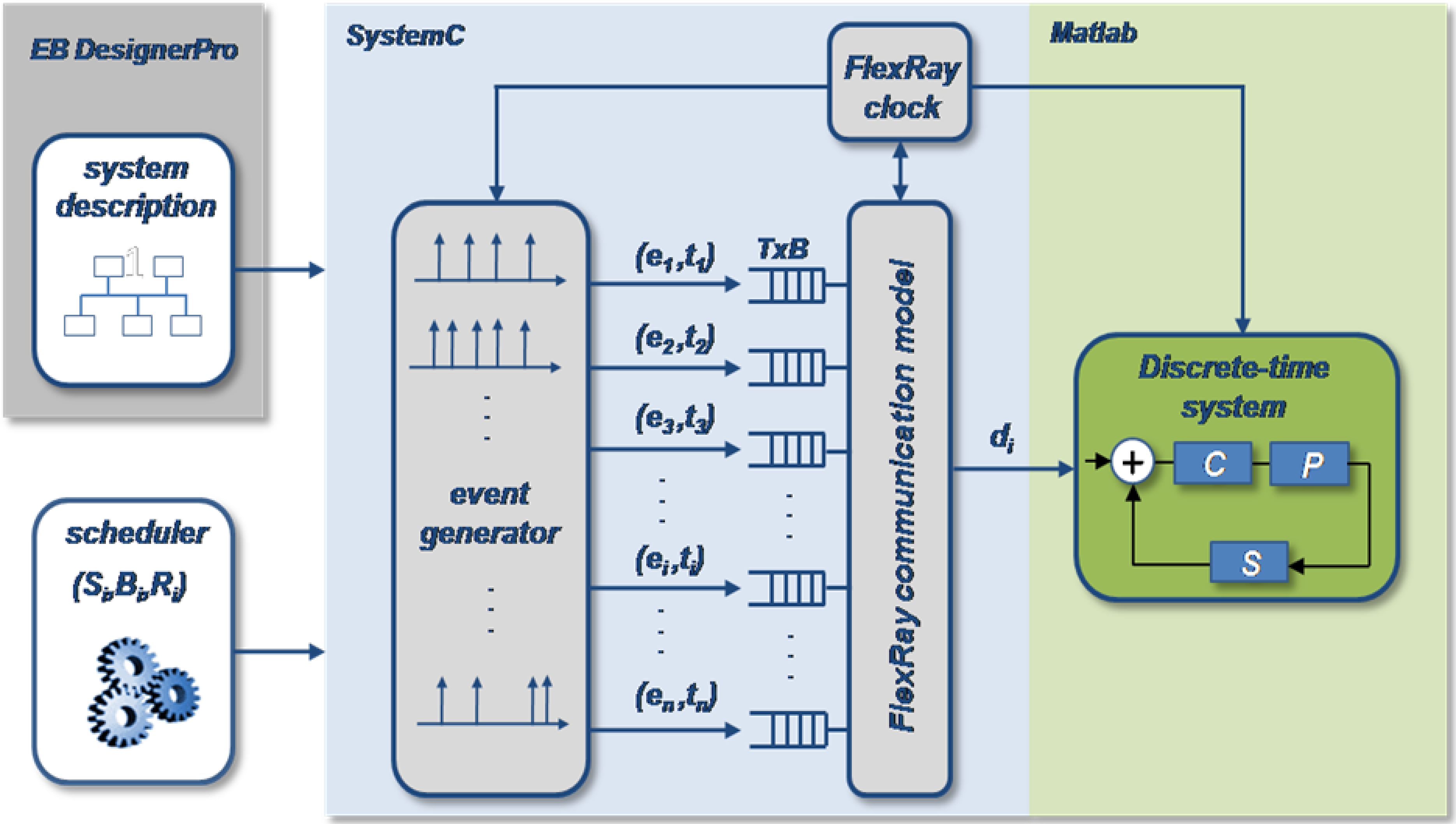
What happens if one unstable sample occurs after μ stable samples?

Resultant Dynamics: $x[\mu+2+k] = A_{cl}^{\mu}A_a x[k]$

If $A_{cl}^{\mu} \approx 0$, $A_{cl}^{\mu}A_a \approx 0$. $\Rightarrow \quad x[k] \rightarrow 0$

Find μ for which $A_{cl}^{\mu} \approx 0$ or $||A_{cl}^{\mu}|| < \epsilon$

Control/FlexRay Co-Simulator



Examples

$$Plant: x[k+1]=\begin{bmatrix} 0.4 & 0.6 & 0.7 \\ -0.56 & -0.9 & -0.6 \\ -3.6 & -1.2 & -2.8 \end{bmatrix} x[k]+\begin{bmatrix} 0.1 \\ 0.7 \\ 0.5 \end{bmatrix} u[k]$$

$$Open-loop\ \ Poles: [-1.57,\ -1.4,\ -0.3283],\ \text{Highly Unstable System Plant}$$

$$Controller: u[k]=Kx[k-1]=[-1.8622\quad -0.2858\quad -1.0355]x[k-1]$$

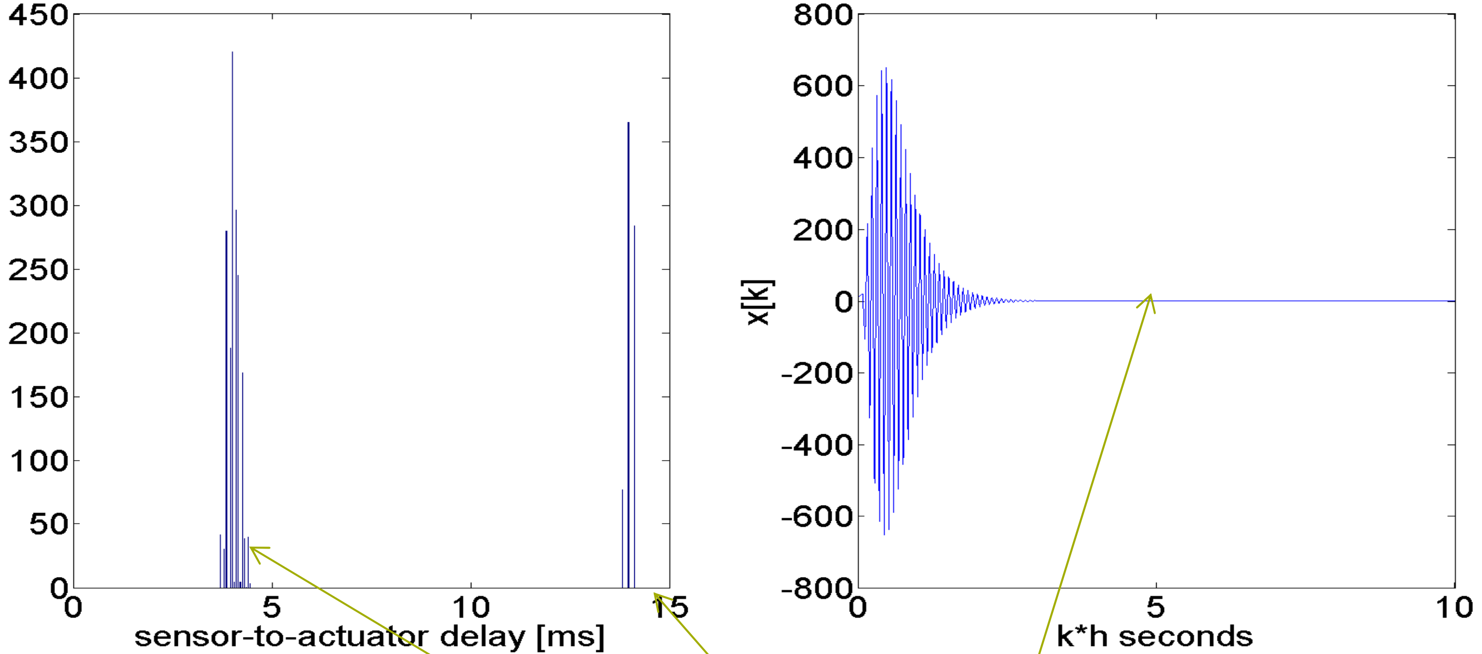
$$Design\ Goal\ (stability): x[k]\rightarrow 0\ as\ \ k\rightarrow\infty$$

$$Sampling\ \ Interval: 40ms$$

We have found using the simulator that the closed-loop system with the above Controller is stable as long as the ratio between number of stable and unstable samples $\mu \geq 52$

Scheduling: Example 1

Stable Example: We choose the communication bus parameters such that $\mu = \infty$

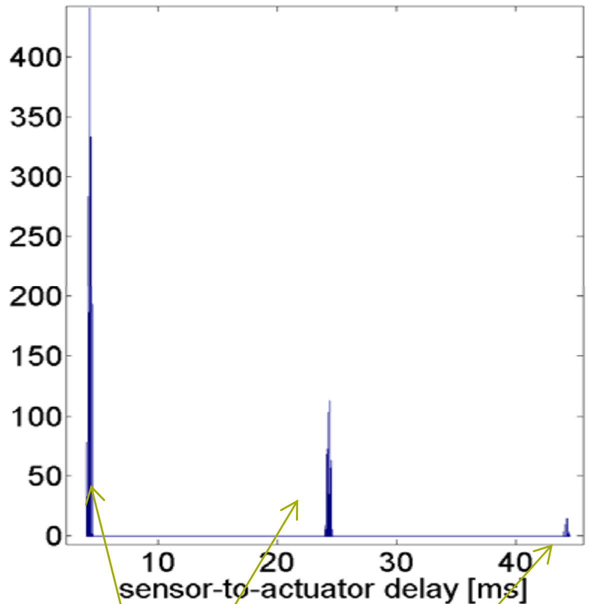


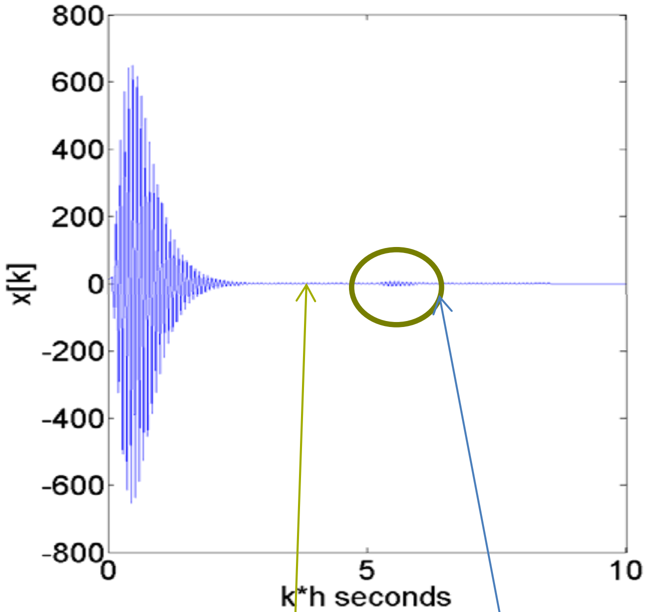
All samples experience delay less than 40 ms, i.e., delay in terms of samples is 1

System is stable as $x[k] \to 0 \;\; as \;\; k \to \infty$

Scheduling: Example 2

Stable Example: We choose the communication bus parameters such that $\mu = 70$





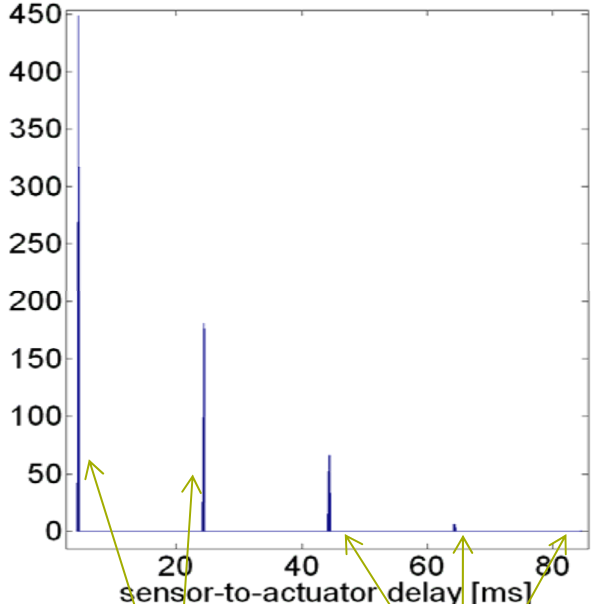
Stable samples Unstable samples

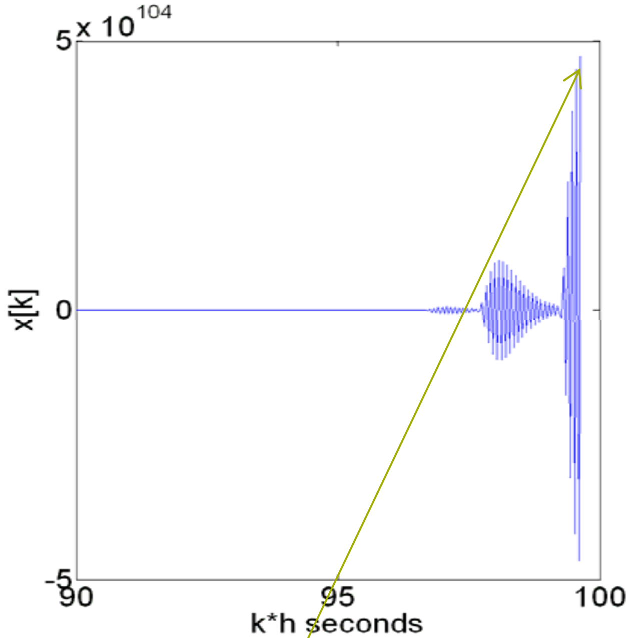
System is stable as $x[k] \rightarrow 0 \ as \ k \rightarrow \infty$

Certain oscillations due to the presence of unstable samples

Scheduling: Example 3

Unstable Example: We choose the communication bus parameters such that $\mu = 11$





Stable samples Unstable samples System is unstable as $x[k] \rightarrow \infty \; as \;\; k \rightarrow \infty$

Concluding remarks

- Given a distributed control/communication setup, we found that any controller design does have certain amount of robustness against the packet drop or unstable samples.
- The degree of robustness depends both on the choice of the control and the open-loop system dynamics.
- The synthesis of communication schedules can be relaxed utilizing such robustness, i.e., instead of hard deadlines, the deadline restrictions become soft. Therefore, some control messages can miss their deadlines of one sampling time $0 \leq \tau \leq h$ provided the number of such deadline miss is upper-bounded by the criteria imposed due to μ.