research supported by NSF, ARO

Networked Control Systems: Protocols and Algorithms

João P. Hespanha

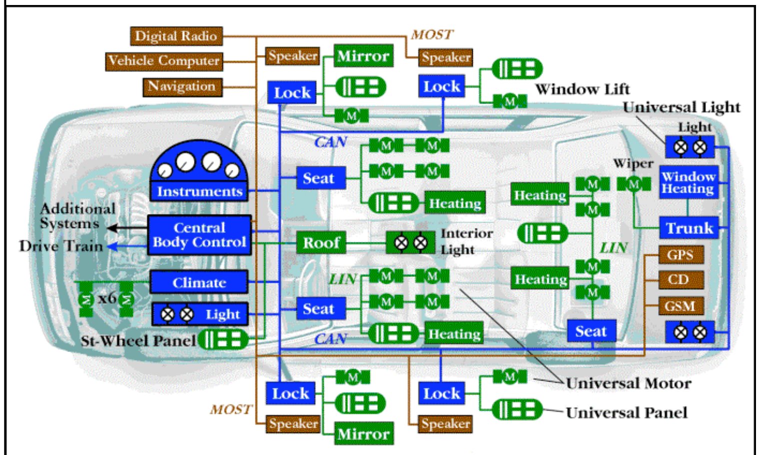
Center for Control
Dynamical Systems and Computation

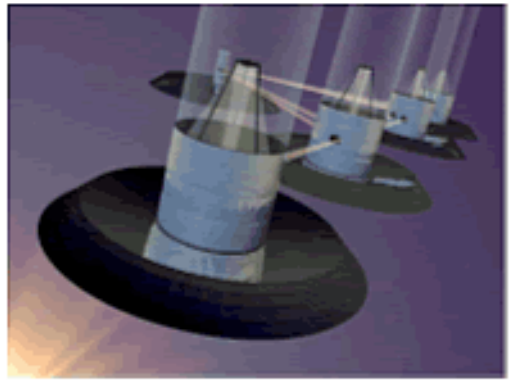
UC SANTA BARBARA
engineering

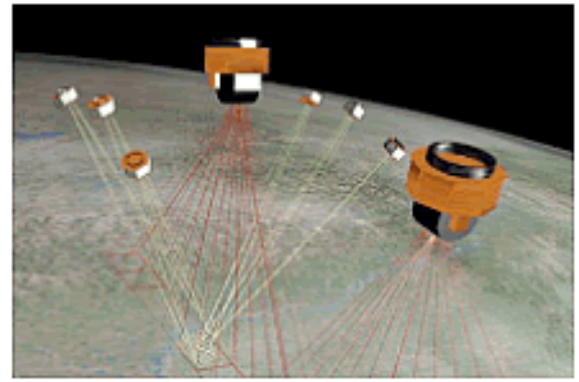
CCDC
Center for Control, Dynamical Systems and Computation

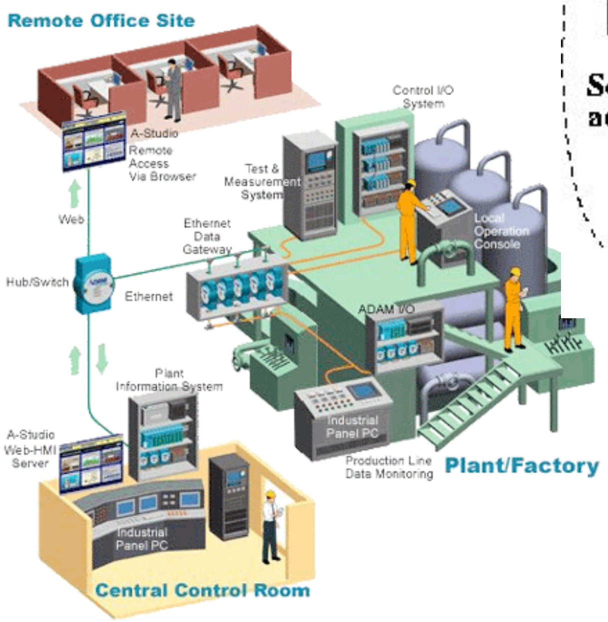
In collaboration with
A. Mesquita (PhD student @ UCSB)
P. Naghshtabrizi (Ford, formerly at UCSB)
Girish Nair (U. Melbourne)

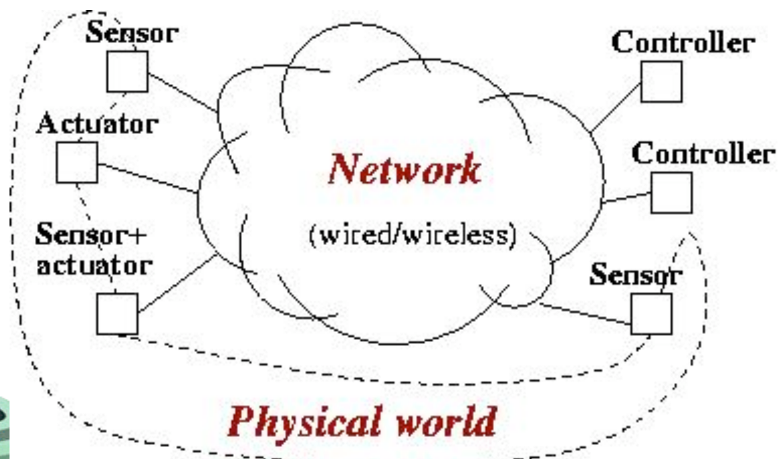
Networked control systems







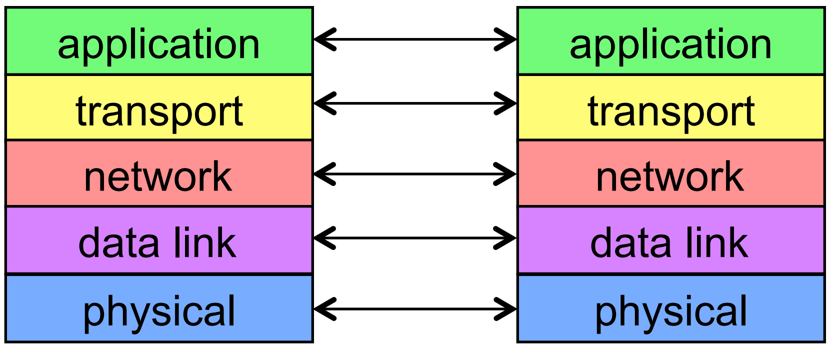




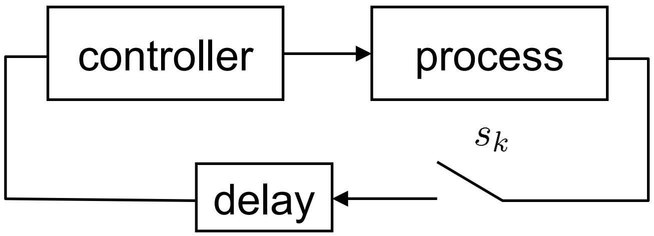
Network protocols & Control laws

UC SANTA BARBARA engineering

end2end network view:



control view:



This talk: ***Co-design of network protocols and control algorithms***

1. Characterize *key parameters* that determine the stability/performance of a networked controls system
2. Construct *protocols* that directly take these parameters into considerations

Illustrative examples:

- transport layer: error correction (& flow control) [ACC'09]
- data link layer: medium access control [Trans. Inst. Meas. Control, 2008]

Transport layer protocols

UC SANTA BARBARA engineering

Most common (general purpose) protocols:

UDP

- no attempt at error correction
- no attempt to control data rate

TCP

- error correction
  - all packets sent should be acknowledged by receiver
  - lack of acknowledgement of packet n leads to retransmission of same packet after packet $n + 3$ (triple duplicate ack mechanism)
- congestion control
  - packet drops are taken as a sign of congestion and lead to send rate decrease

Transport layer protocols

UC SANTA BARBARA engineering

Most common (general purpose) protocols:

high drop rates can lead to poor performance and eventually instability

UDP

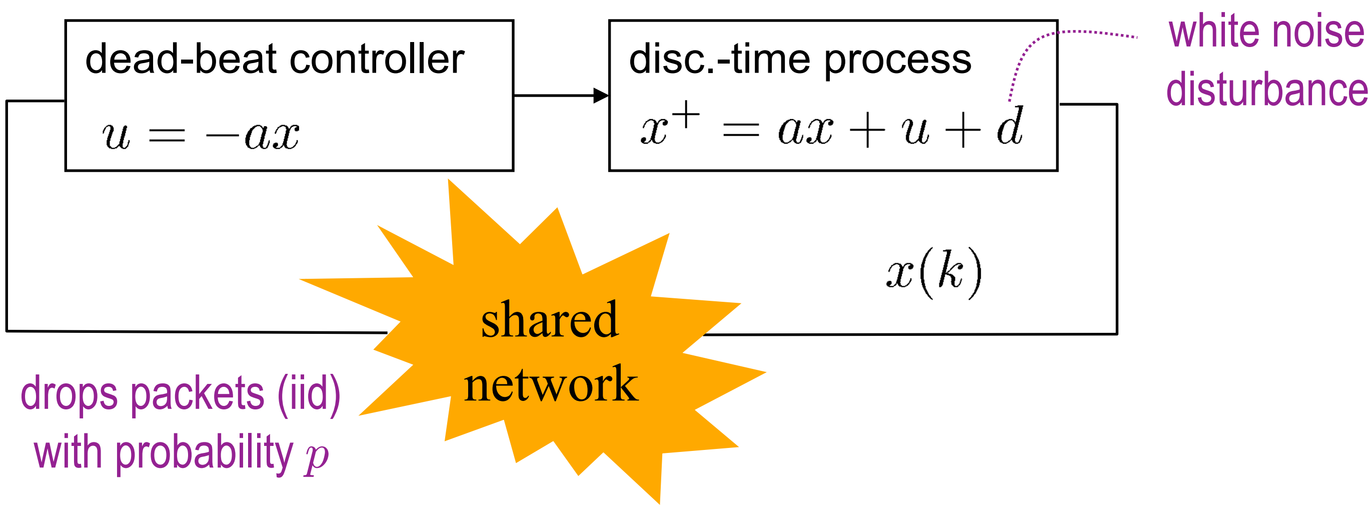
- no attempt at error correction
- no attempt to control data rate

TCP

- error correction
  - all packets sent should be acknowledged by receiver
  - lack of acknowledgement of packet *n* leads to retransmission of same packet after packet *n* + 3 (triple duplicate ack mechanism)
- congestion control
  - packet drops are taken as a sign of c decrease

some drops may be okay; delayed retransmissions are essentially useless

Illustrative 1-D problem



The closed-loop is *mean-square stable* (i.e., $E[\ x(k)^2\ ] < \infty$) if and only if

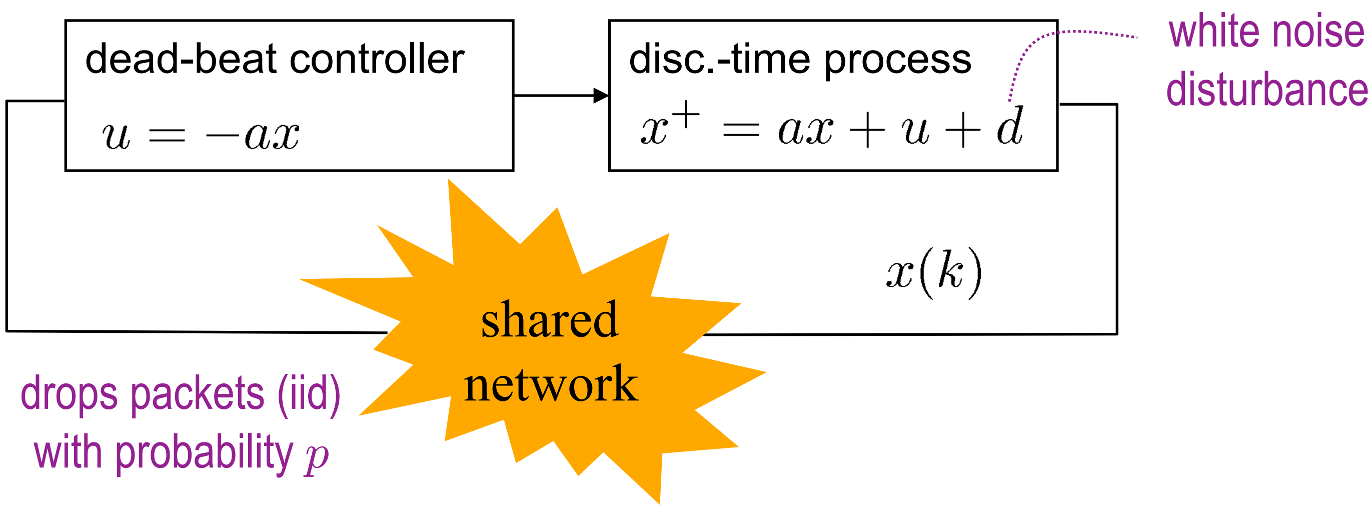
$$p < \frac{1}{|a|^2}$$

(it is also straightforward to compute a tight asymptotic bound on $E[x(k)^2]$)

Intuition: ignoring the disturbance d

$$x(k+1)^2 = \begin{cases} 0 & \text{with probability } 1-p \\ |a|^2 x(k)^2 & \text{with probability } p \end{cases} \quad \Rightarrow \quad E[x(k+1)^2] = p\,|a|^2\, E[x(k)^2]$$

Illustrative 1-D problem



The closed-loop is *mean-square stable* (i.e., E[$x(k)^2$] <∞) if and only if

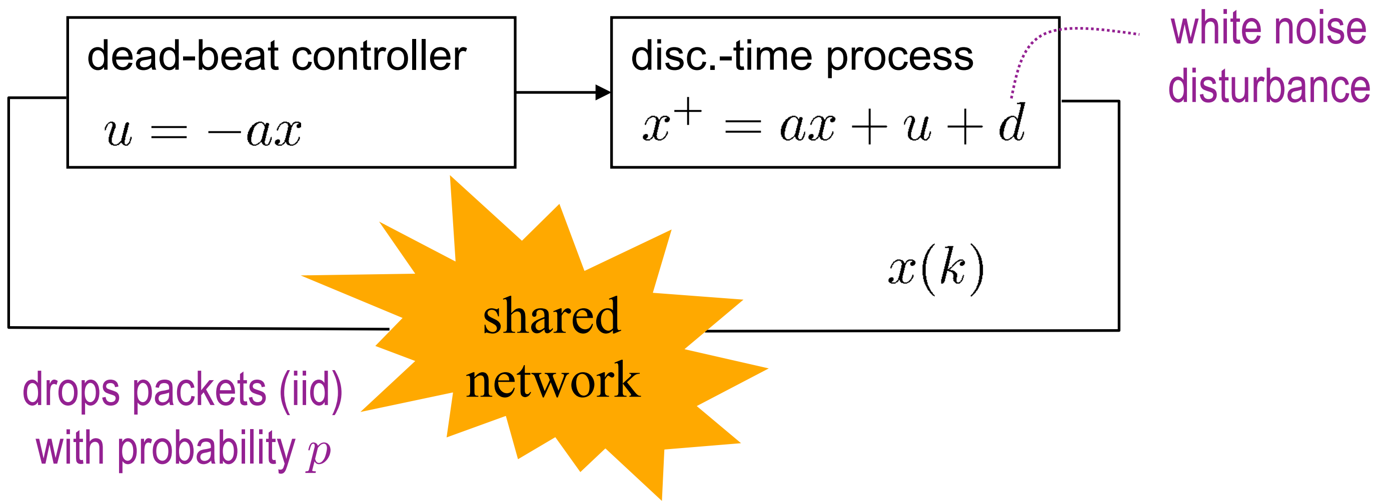
$$p < \frac{1}{|a|^2}$$

(it is also straightforward to compute a tight asymptotic bound on E[$x(k)^2$])

But what if |a|>1 *and the probability of drop is larger than this bound?*

Redundant transmissions

UC SANTA BARBARA engineering



redundant transmissions ≡ at each time step one sends N copies of $x(k)$ through independent channels (time, frequency, or spatial diversity), each with drop probability p

The closed-loop is *mean-square stable* (i.e., E[$x(k)^2$] <∞) if and only if

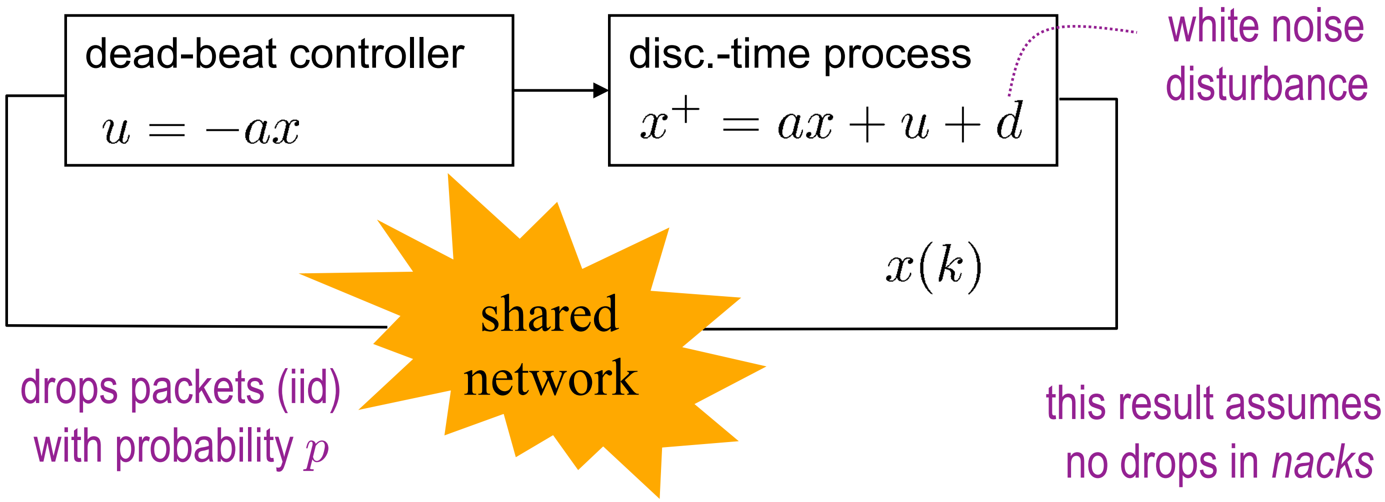
$$p < \frac{1}{|a|^{\frac{2}{N}}}$$

but transmission rate is N *times larger*

any drop probability can be accommodated by choosing N sufficiently large

A simple “error-correction” protocol

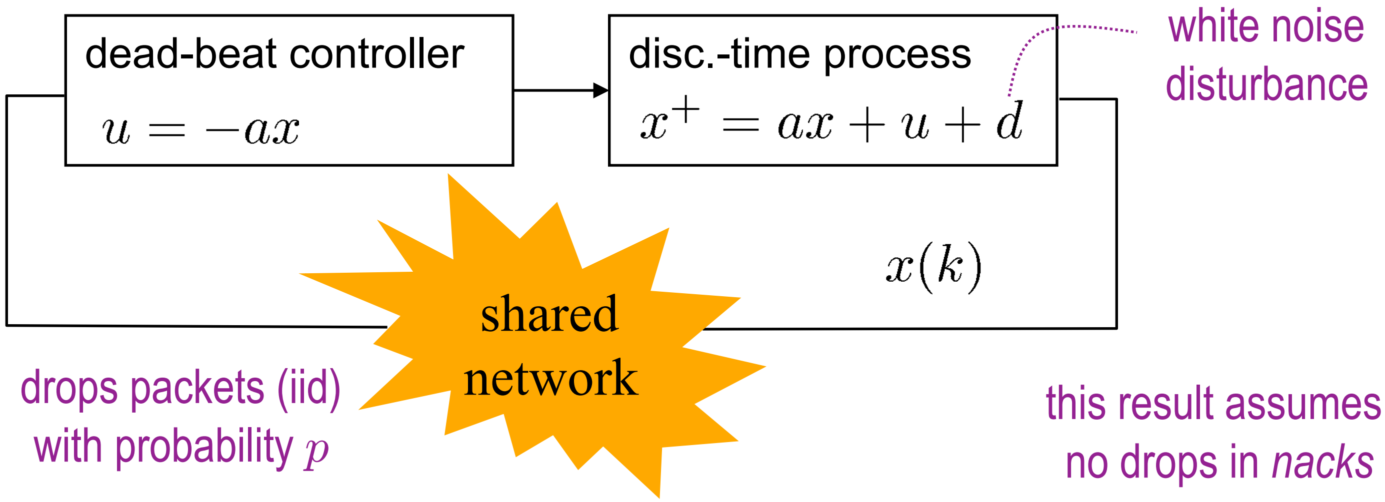
UC SANTA BARBARA engineering



1. when a packet is lost, receiver sends a “negative acknowledgement” (*nack*)
2. transmitter generally sends *one* packet at each sampling time, however…
3. upon reception of *nack*, transmitter sends *two* copies of the following packet

A simple “error-correction” protocol

UC SANTA BARBARA engineering



1. when a packet is lost, receiver sends a “negative acknowledgement” (*nack*)
2. transmitter generally sends *one* packet at each sampling time, however…
3. upon reception of *nack*, transmitter sends *two* copies of the following packet

The closed-loop is *mean-square stable* (i.e., E[$x(k)^2$] $<\infty$) if and only if

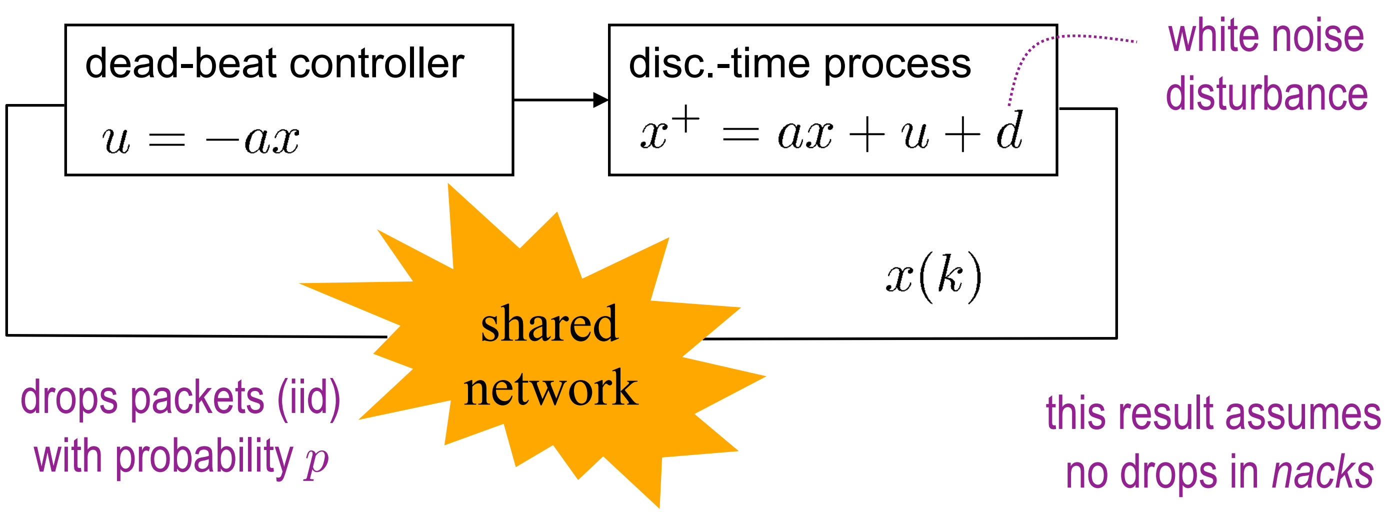
$$p < \frac{1}{|a|}$$

similar bound as if always sending **two packets**

but average transmission rate is only *1+O(p) times larger*

[ACC’09]

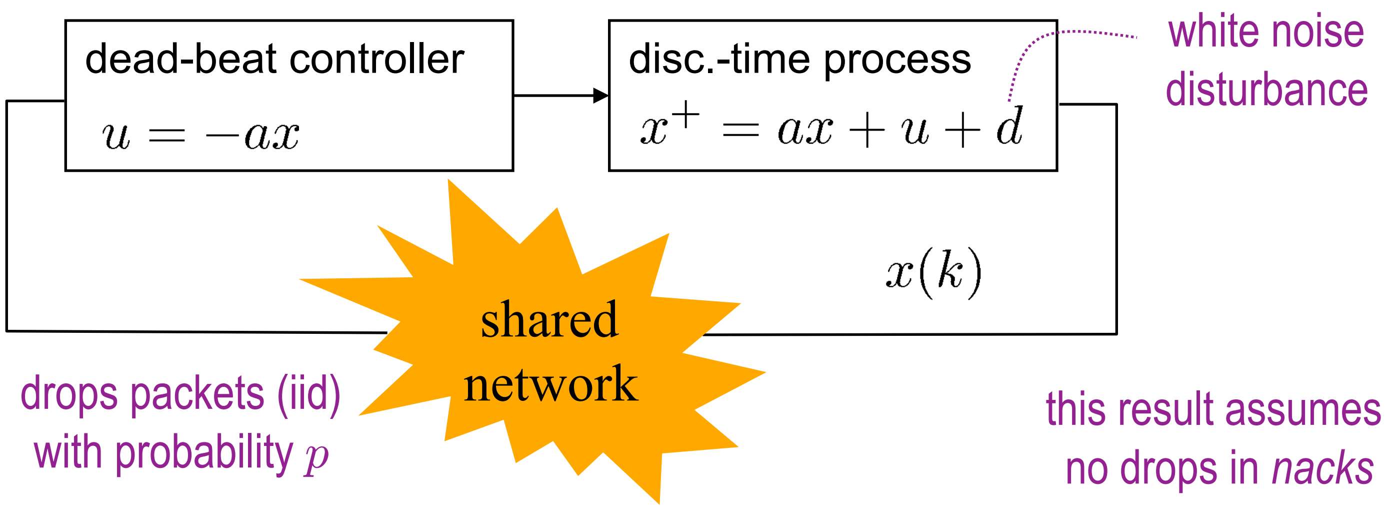
Even better…



Pick a function $v : \mathbb{N} \to \mathbb{N}$, with $v(0) = 1$

1. when a packet is lost, receiver sends a "negative acknowledgement" (*nack*)
2. transmitter keeps track of number $\ell(k)$ of consecutive drops prior to time k
3. transmitter sends $v\big(\ell(k)\big)$ copies of each packet

Even better…



Pick a function $v : \mathbb{N} \to \mathbb{N}$, with $v(0) = 1$

1. when a packet is lost, receiver sends a "negative acknowledgement" (*nack*)
2. transmitter keeps track of number $\ell(k)$ of consecutive drops prior to time k
3. transmitter sends $v\big(\ell(k)\big)$ copies of each packet

For every p, a, and N, one can find a function $v : \mathbb{N} \to \mathbb{N}$ such that

- closed-loop is *mean-square stable* (i.e., E[$x(k)^2$] $< \infty$)
- average transmission rate is only *1+O(p^N) times larger*
- requires at least N independent channels

stabilizes any system

arbitrarily small increase in the transmission rate

all but one channel are rarely utilized

Even better…

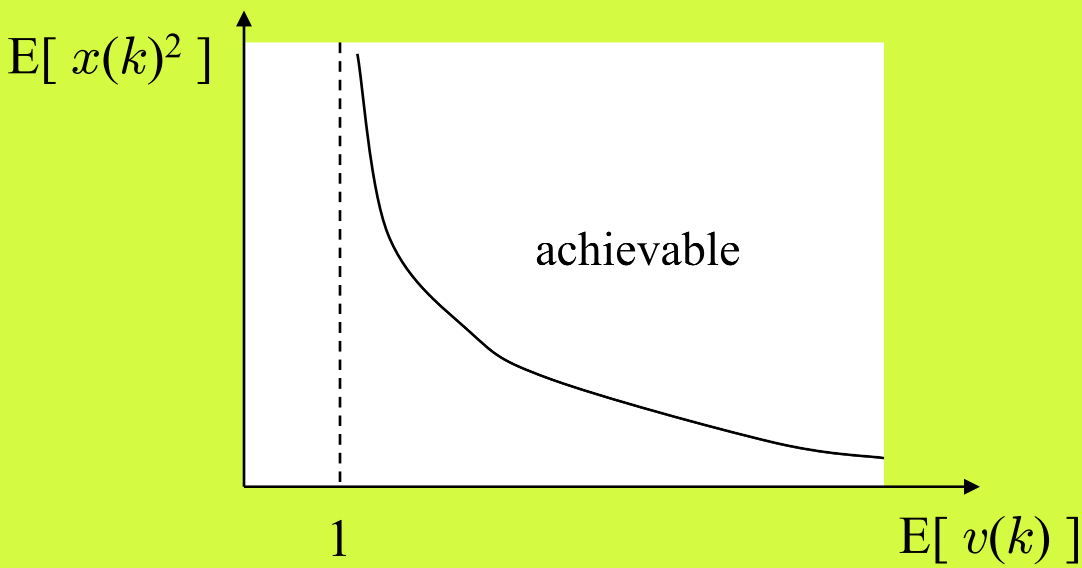
dead-beat controller

disc.-time process

white noise disturbance

- can stabilize any system for any drop probability
- with arbitrarily small increase in the transmission rate

no (completely) free lunch… E[$x(k)^2$] will be large



…sult assumes …ops in *nacks*

Pick a f…
1. when… ’ (*nack*)
2. trans… time k
3. trans…

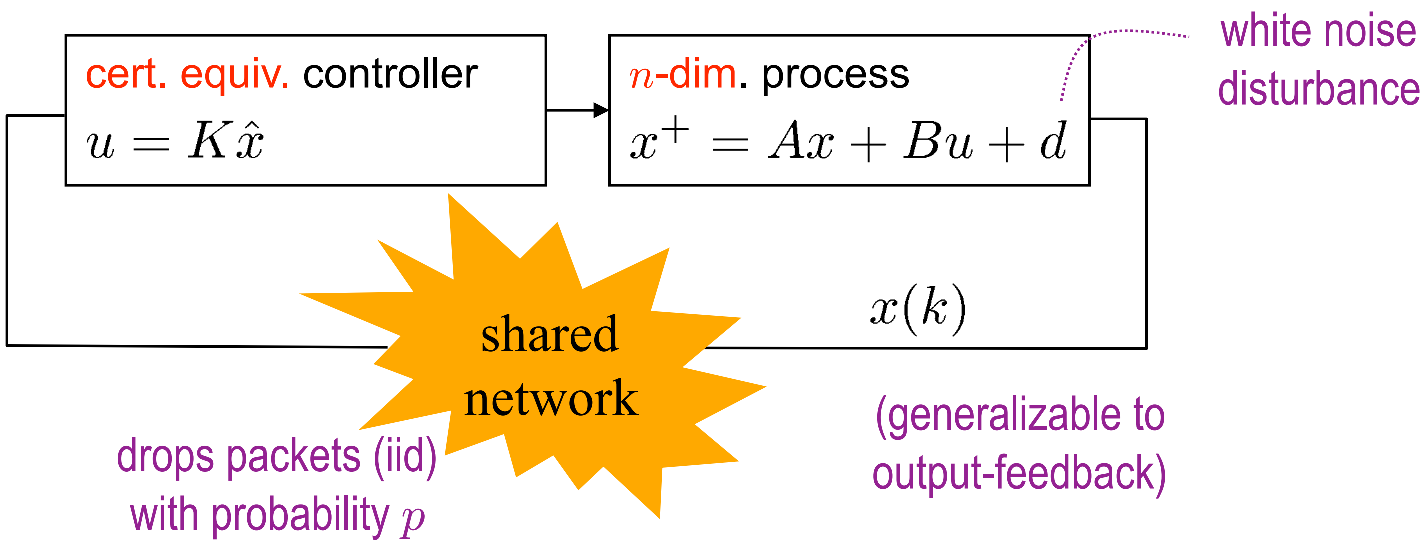
For ever…
- close… …ny system
- average transmission rate is only $1+O(p^N)$ *times larger*
- requires at least N independent channels

arbitrarily small increase in the transmission rate

all but one channel are rarely utilized

Optimal "error-correction" protocols

UC SANTA BARBARA engineering



choose $v(k) \equiv$ number of copies of $x(k)$ to send at time instant k

to minimize

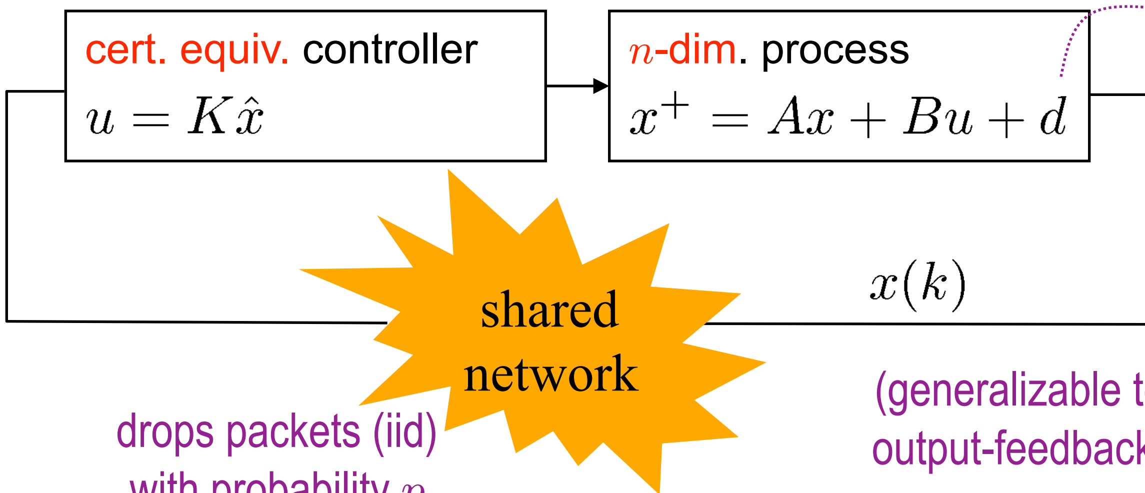
$$\lim_{N\to\infty} \underbrace{\left(\frac{1}{N}\,\mathrm{E}\left[\sum_{k=0}^{N-1} \|x(k)-\hat{x}(k)\|^2\right]\right)}_{\substack{\text{state estimation error}\\ \text{(performance)}}} + \lambda \underbrace{\left(\frac{1}{N}\,\mathrm{E}\left[\sum_{k=0}^{N-1} v(k)\right]\right)}_{\substack{\text{transmission rate}\\ \text{(communication)}}}$$

average-cost optimal control of a Markov process on $\mathbb{R}^n$

Optimal "error-correction" protocols

UC SANTA BARBARA engineering

white noise disturbance



drops packets (iid) with probability p

(generalizable to output-feedback)

$$\lim_{N\to\infty} \left(\frac{1}{N} \mathrm{E}\left[\sum_{k=0}^{N-1} \|x(k) - \hat{x}(k)\|^2 \right] \right) + \lambda \left(\frac{1}{N} \mathrm{E}\left[\sum_{k=0}^{N-1} v(k) \right] \right)$$

Theorem:

- optimal $v(k)$ is generated by a memoryless policy of the form

$$v(k) = \pi^*\big(x(k) - \hat{x}(k)\big)$$

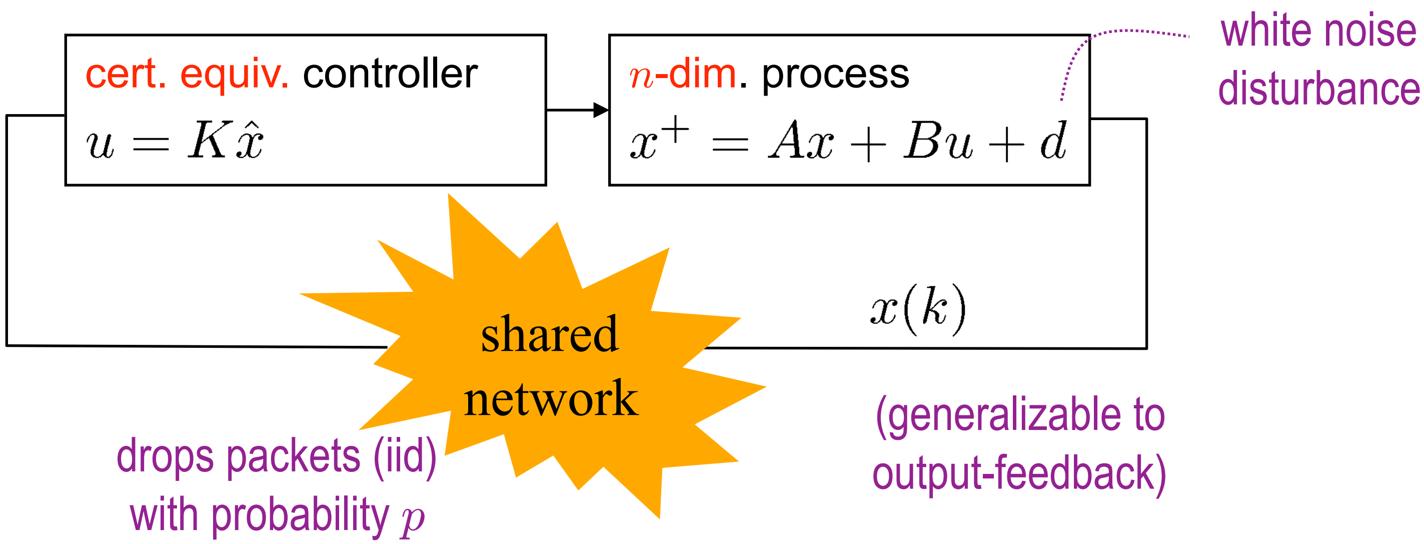
transmitter must construct a state estimate to determine optimal $v(k)$

- optimal policy π^* can be computed using dynamic programming and value-iteration

computationally difficult for large n

Optimal “simplified” protocols

UC SANTA BARBARA engineering



choose $v(k) \equiv$ number of copies of $x(k)$ to send at time instant k

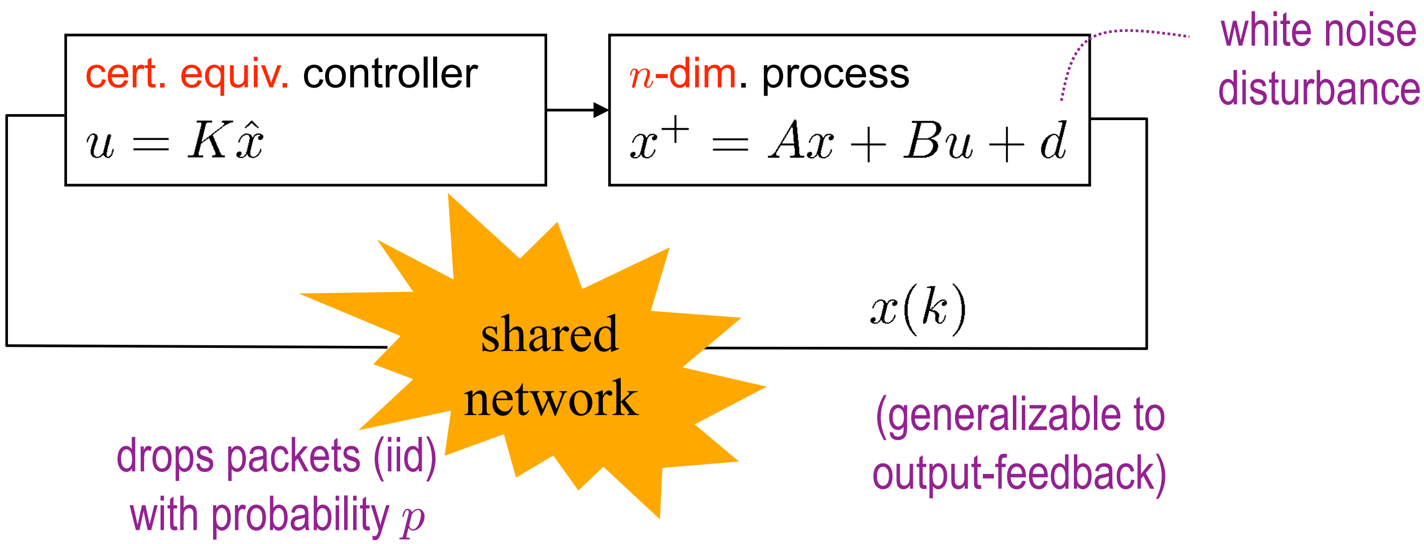
to minimize

$$\lim_{N\to\infty} \underbrace{\left(\frac{1}{N}\,\mathrm{E}\left[\sum_{k=0}^{N-1} \|x(k) - \hat{x}(k)\|^2\right]\right)}_{\substack{\text{state estimation error}\\ \text{(performance)}}} + \lambda \underbrace{\left(\frac{1}{N}\,\mathrm{E}\left[\sum_{k=0}^{N-1} v(k)\right]\right)}_{\substack{\text{transmission rate}\\ \text{(communication)}}}$$

but transmitter must choose $v(k)$ based only on # of consecutive drops (from nacks)

Optimal "simplified" protocols

UC SANTA BARBARA engineering



$$\lim_{N\to\infty} \left(\frac{1}{N} \mathrm{E} \left[\sum_{k=0}^{N-1} \|x(k) - \hat{x}(k)\|^2 \right] \right) + \lambda \left(\frac{1}{N} \mathrm{E} \left[\sum_{k=0}^{N-1} v(k) \right] \right)$$

Theorem:

- optimal $v(k)$ is generated by a memoryless policy of the form

$$v(k) = \pi^* \big(\ell(k) \big)$$

transmitter only needs to keep track of $\ell(k) \equiv$ # of consecutive drops (from nacks)

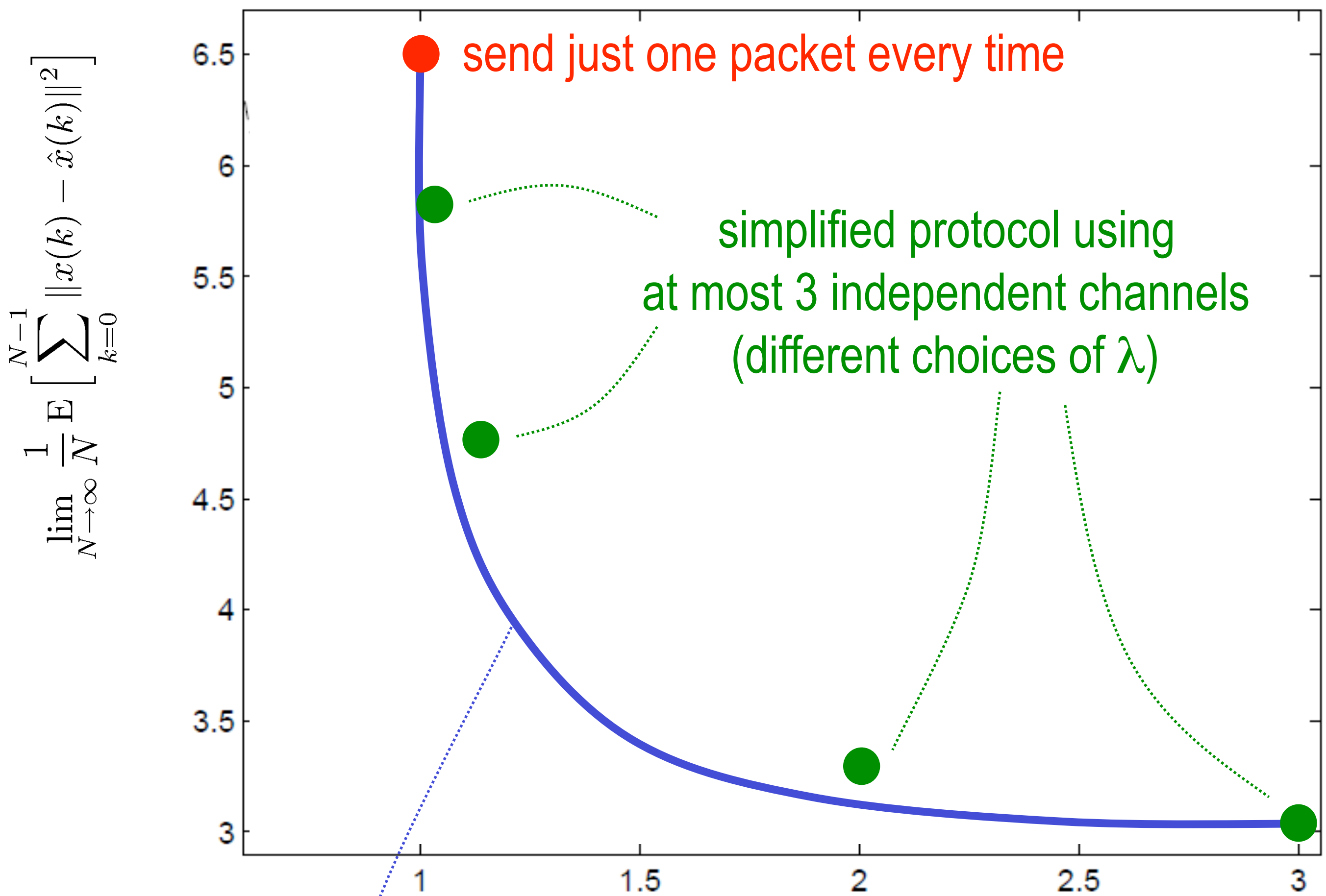
- optimal policy π^* can be computed using dynamic programming and value-iteration

computationally much easier (optimization on countable-state MDP with size independent of n)

Example

$$x^+ = 2x + u + w$$
$$w \sim N(0,3)$$
$$p = .15$$

state estimation error
(performance)



$$\lim_{N\to\infty} \frac{1}{N} \mathrm{E}\left[\sum_{k=0}^{N-1} v(k)\right]$$

optimal protocol using
at most 3 independent channels
(different choices of λ)

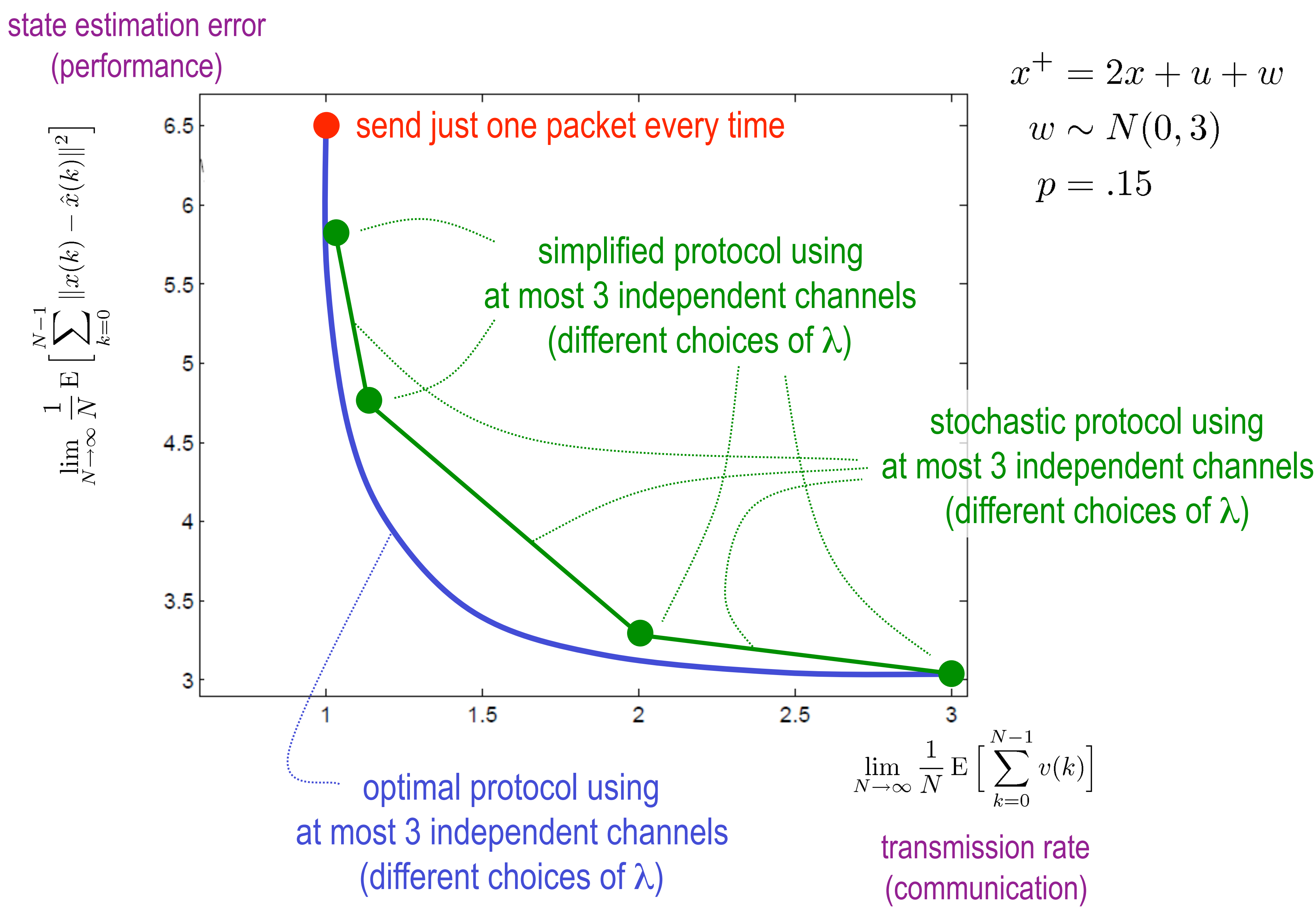
transmission rate
(communication)

Example

$$x^+ = 2x + u + w$$

$$w \sim N(0,3)$$

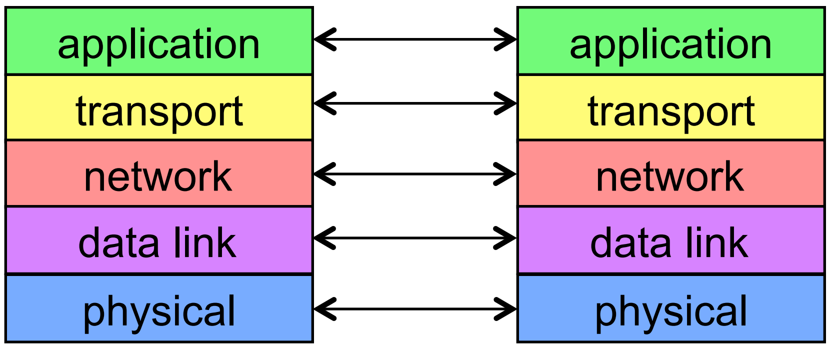
$$p = .15$$



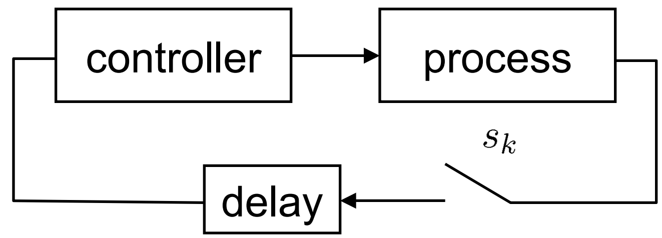
Network protocols & Control laws

UC SANTA BARBARA engineering

network view:



control view:



This talk: ***Co-design of network protocols and control algorithms***

1. Characterize *key parameters* that determine the stability/performance of a networked controls system
2. Construct *protocols* that directly take these parameters into considerations

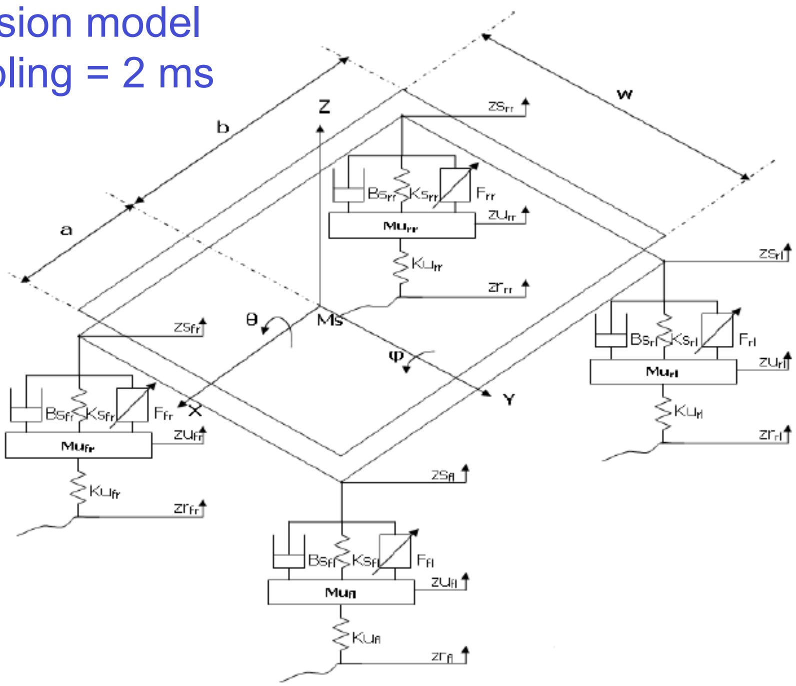
Illustrative examples:

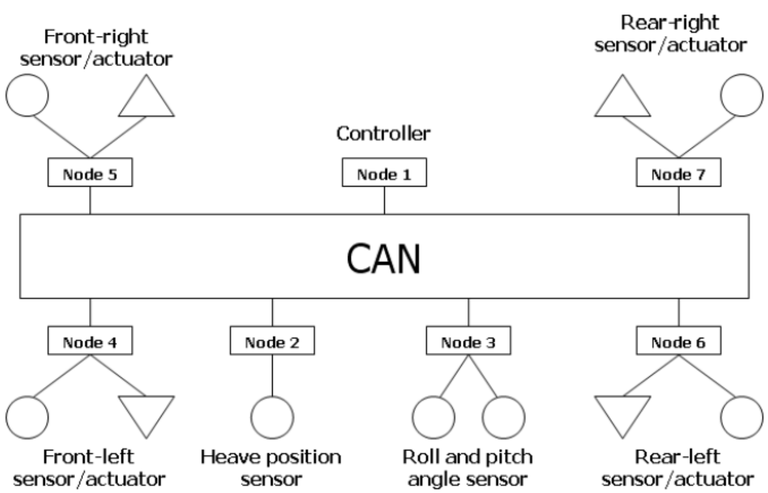
- transport layer: error correction (& flow control) [ACC'09]
- data link layer: medium access control [Trans. Inst. Meas. Control, 2008]

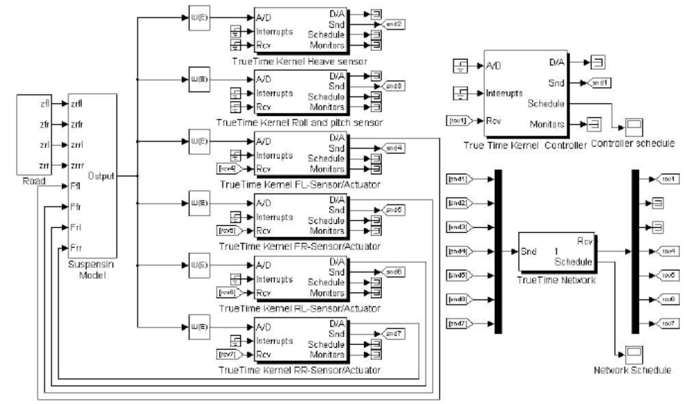
Network access - does priority matter?

UC SANTA BARBARA engineering

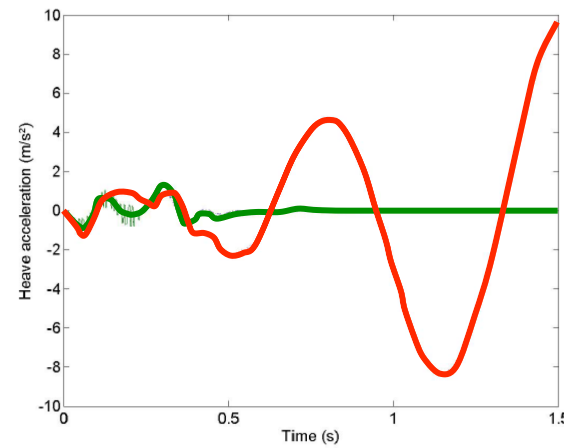
Active suspension model
constant sampling = 2 ms







Simulated with TrueTime



node	1	2	3	4	5	6	7
case 1	1	2	3	4	5	6	7
case 2	7	1	2	3	4	5	6

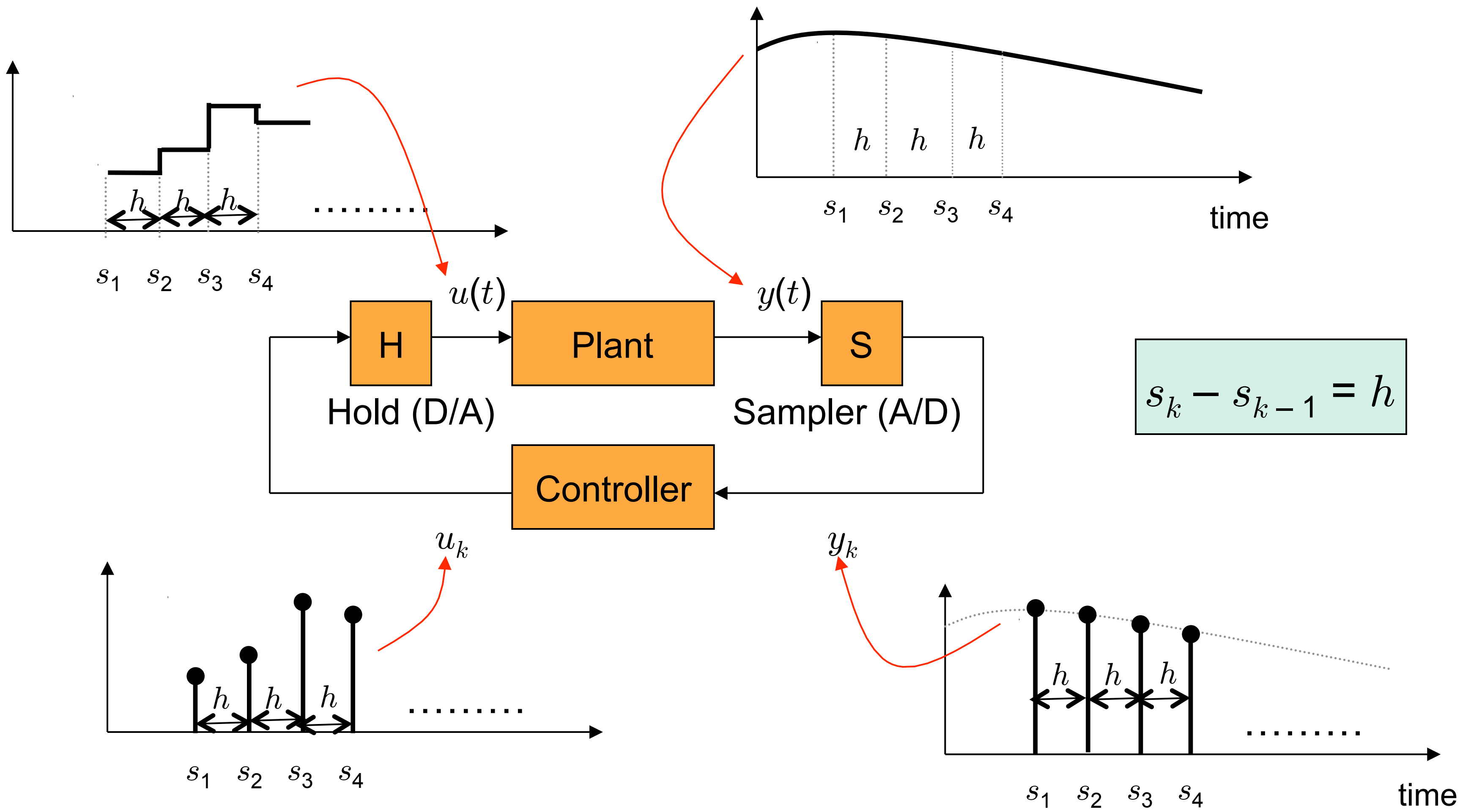
CAN bus priority

Ben Gaid, Cela,Kocik

Digital control systems

UC SANTA BARBARA engineering

Digital control systems usually exhibit uniform sampling intervals and small delays

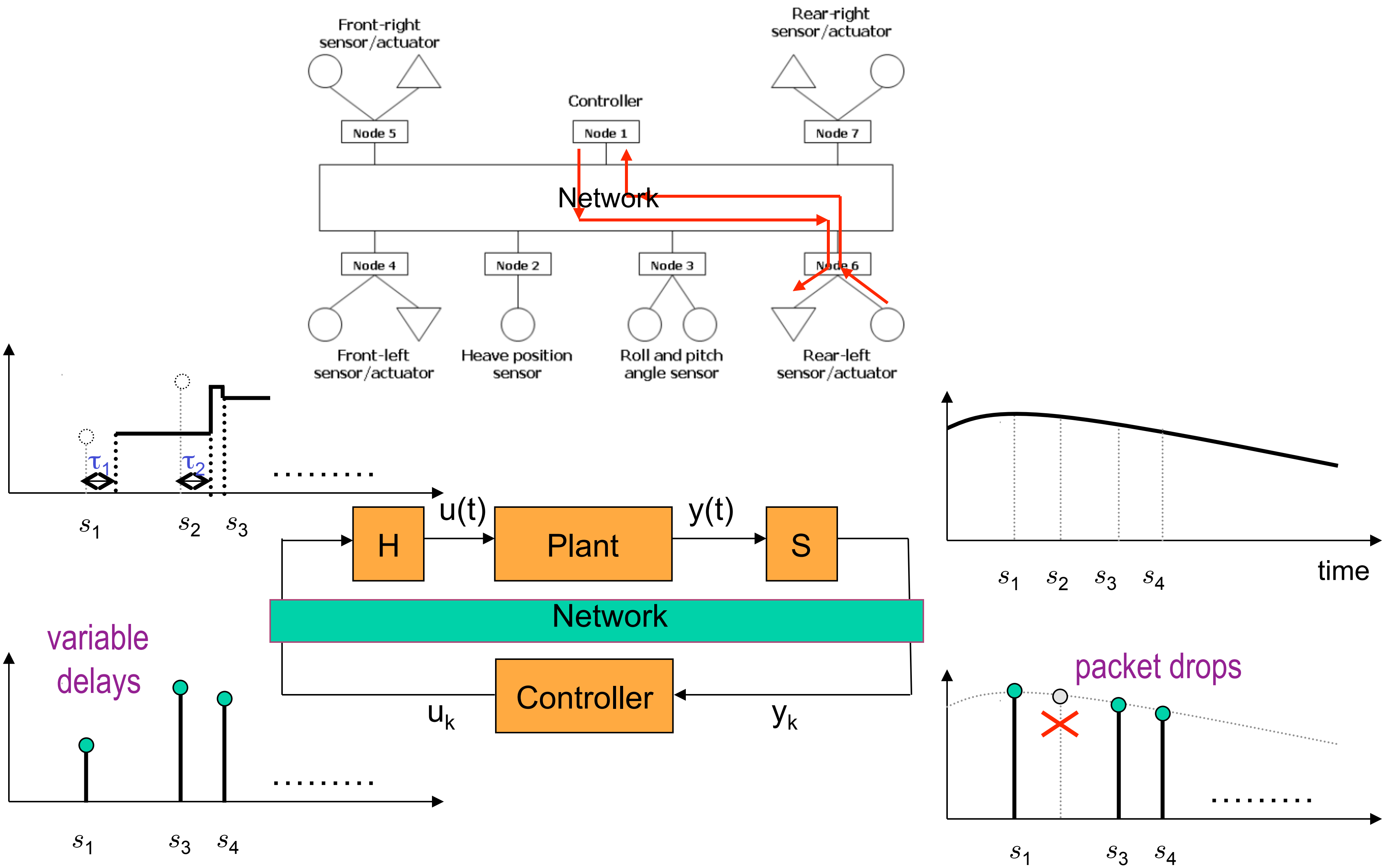


$$s_k - s_{k-1} = h$$

Non-uniform sample/delay

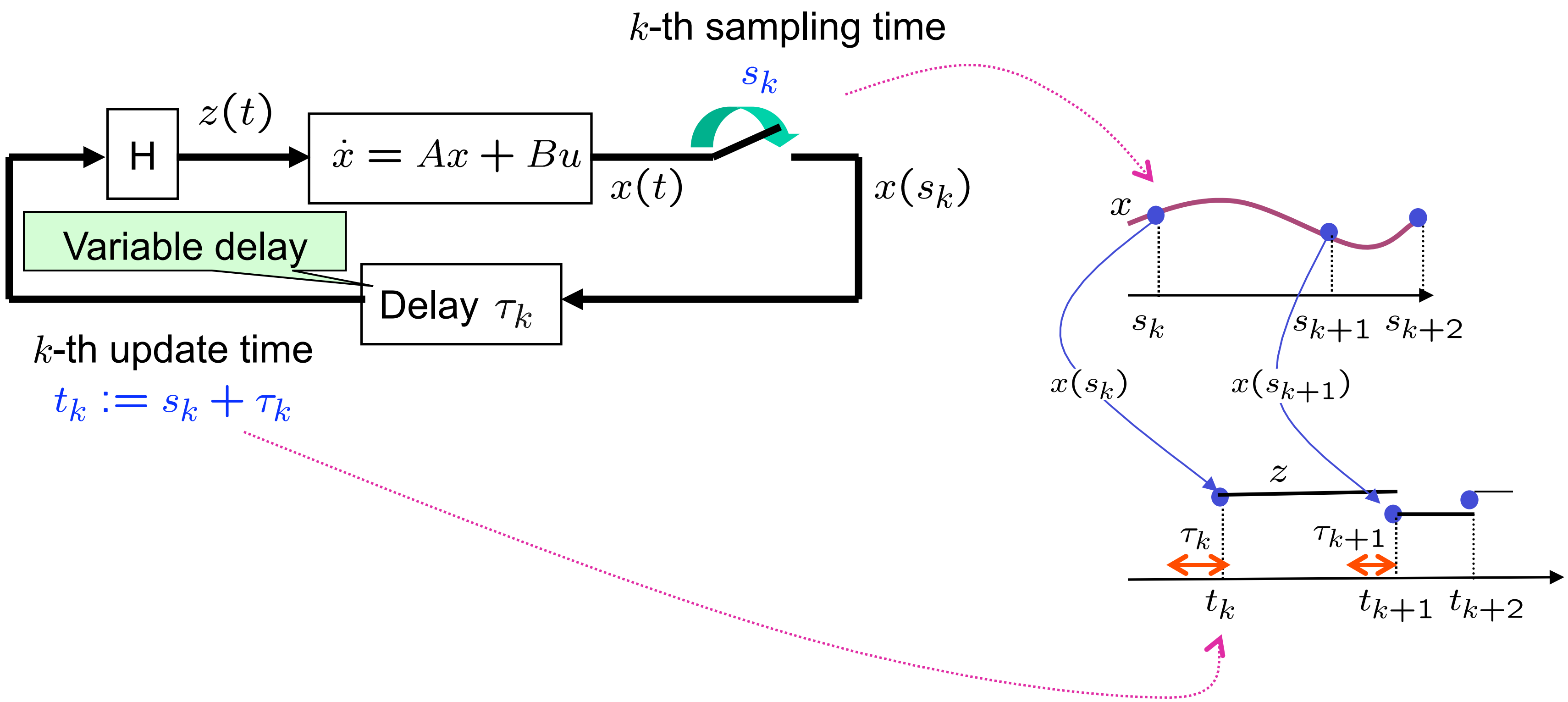
UC SANTA BARBARA engineering

- Uniform sampling cannot be guaranteed in NCSs (packet drops, clock synchronization …)
- Different samples may experience different delays.



Delay impulsive systems (SISO)

UC SANTA BARBARA engineering



Flow: $$\begin{bmatrix} \dot{x} \\ \dot{z} \end{bmatrix} = \begin{bmatrix} Ax + Bz \\ 0 \end{bmatrix}, \qquad t \neq t_k, \forall k \in \mathbb{N}$$

Jump or impulse: $$\begin{bmatrix} x(t_k) \\ z(t_k) \end{bmatrix} = \begin{bmatrix} x^-(t_k) \\ x(\underbrace{t_k - \tau_k}_{s_k}) \end{bmatrix}, \qquad t = t_k, \forall k \in \mathbb{N}$$

$$x^-(t) := \lim_{\tau \uparrow t} x(\tau)$$

Stability of delay impulsive systems

Consider delay impulsive system

$$\begin{aligned} \dot{x} &= f_k(x,t), && t \neq t_k, \forall k \in \mathbb{N}, \\ x(t_{k+1}) &= g_k(x^-(t_{k+1}), x(t_{k+1}-\tau_k)) && t = t_k, \forall k \in \mathbb{N}. \end{aligned}$$

System is GUES if there exists a Lyapunov functional

$$V : C\big([-r,0],\mathbb{R}^n\big) \times \mathbb{R}^+ \to \mathbb{R}^+$$

such that

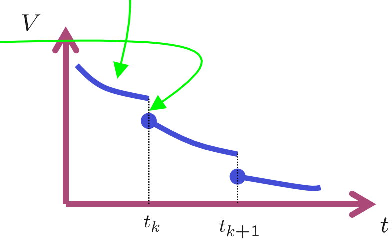
state x truncated to the last r time units

(a) $d_1|\phi(0)|^b \leq V(\phi,t) \leq d_2|\phi(0)|^b + \bar{d}_2 \int_{t-r}^{t} |\phi(s)|^b ds \quad \forall \phi \in C\big([-r,0]\big), t \in \mathbb{R}^+$

(b) $\dfrac{dV(x_t,t)}{dt} \leq -d_3|x(t)|^b \qquad t \neq t_k,\ \forall k \in \mathbb{N}$

(c) $V(x_{t_k},t_k) \leq \lim_{t \uparrow t_k} V(x_t,t) \qquad t = t_k, \forall k \in \mathbb{N}$

for $d_1, d_2, \bar{d}_2, d_3, b > 0,$

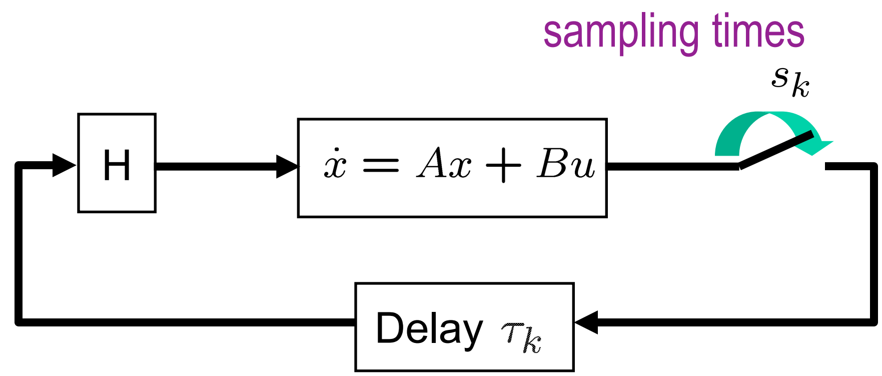


state x truncated to the last r time units

- Extended version of Lyapunov-Krasovskii Theorem for delayed systems with jumps.
- Lead to LMI for linear systems as opposed to conditions in *Liu ITAC 01*, *Sun-Michel ITAC 05*

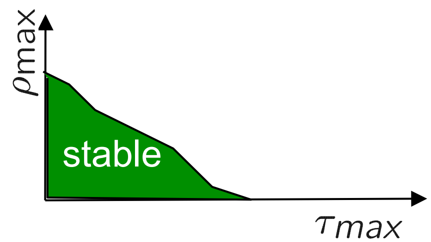
Stability of SISO NCSs

UC SANTA BARBARA engineering



- Based on previous theorem and an appropriate choice of functional:

- There exists a set of pairs (ρ_{max}, τ_{max})



such that

$$\left.\begin{array}{l} s_{k+1} - s_k \leq \rho_{max} \\ 0 \leq \tau_k \leq \tau_{max} \end{array}\right. \quad \forall k \in \mathbb{N} \quad \Rightarrow$$

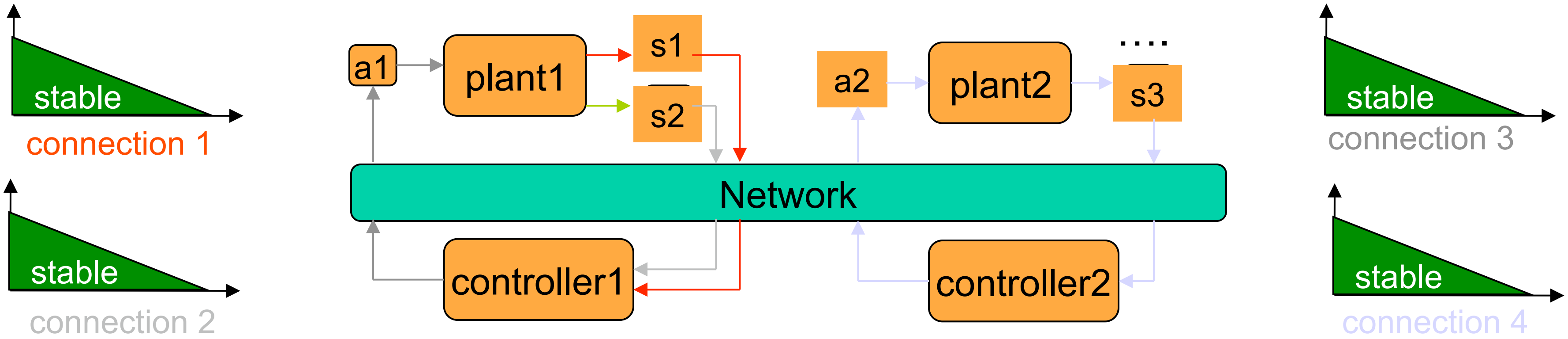
exponential stability of the closed loop

- We find the stability region by solving Linear Matrix Inequalities (LMIs)

$$V := x'Px + \int_{t-\rho}^{t} (\rho_{max} - t + s)\dot{x}'(s)R_1\dot{x}(s)ds + \int_{t-\sigma}^{t} (\sigma_{max} - t + s)\dot{x}'(s)R_2\dot{x}(s)ds + \cdots$$

$$\rho(t) := t - s_k,\ \sigma(t) := t - t_k \quad t_k \leq t < t_{k+1} \cdots$$

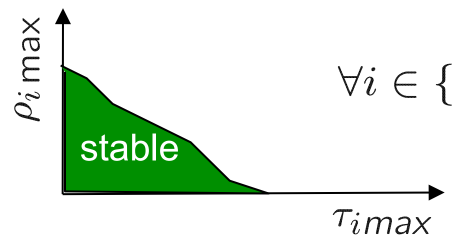
Stability of MIMO NCSs



kth sampling time of channel $i \equiv s_k^i$

kth update time of channel $i \equiv t_k^i := s_k^i + \tau_k^i$

There exist sets of pairs $(\rho_{i\,\max}, \tau_{i\max})$



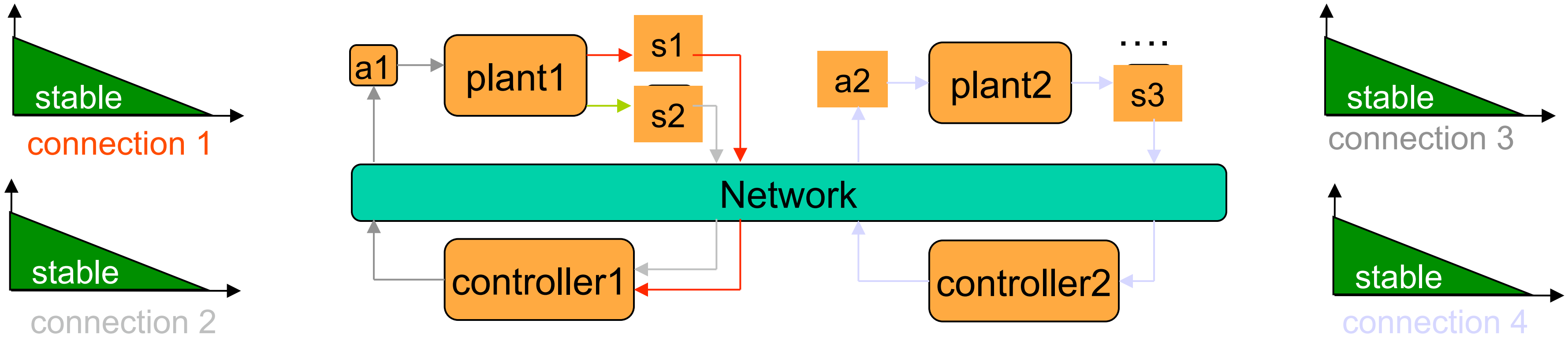
$\forall i \in \{1, \cdots, m\}$

such that

$$\left.\begin{array}{l} s_{k+1}^i - s_k^i \leq \rho_{i\,\max}, \\ \text{delay} = \tau_k^i \leq \tau_{i\max} \end{array}\right\} \Rightarrow \text{exponential stability of the all closed loops}$$

Based on previous theorem and an appropriate choice of functional…

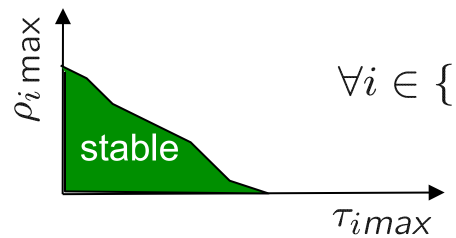
Stability of MIMO NCSs



k*th* sampling time of channel $i \equiv s_k^i$

k*th* update time of channel $i \equiv t_k^i := s_k^i + \tau_k^i$

There exist sets of pairs $(\rho_{i\,\max}, \tau_{i\max})$



$\forall i \in \{1, \cdots, m\}$

such that

$$s_{k+1}^i - s_k^i \le \rho_{i\,\max},$$

$$\text{delay} = \tau_k^i = b_k^i + C_k^i \le \tau_{i\max}$$

blocking delay transmission + prop. delay

$\Rightarrow$ exponential stability of the all closed loops

- Blocking delay depends on **priority assignment**
- Inequalities above define **deadlines for transmission delivery** (to be used, e.g., by Earliest Deadline First – EDF – scheduling)

Stable EDF scheduling

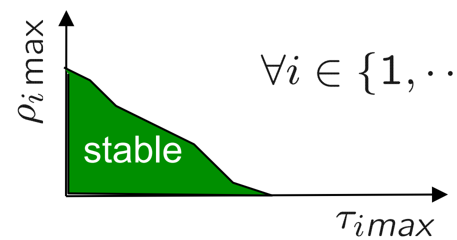
UC SANTA BARBARA engineering

Suppose:

do not sample too fast

- $\rho_{i\,\min} \le s^i_{k+1} - s^i_k \le \rho_{i\,\max}, \forall k \in \mathbb{N}, i \in \{1, \cdots, n\}$
- $\displaystyle\sum_{i=1}^{n} \frac{C_i}{\rho_{i\,\min}} \le 1$ fastest sampling does not exceed capacity
- $\displaystyle\sum_{i=1}^{n} \lfloor \frac{t - \tau_{i\,\max}}{\rho_{i\,\min}} \rfloor^{+} C_i + \max_i C_i \le t \quad \forall t \in S$

and every $(\rho_{i\,\max}, \tau_{i\,max})$ belongs to shaded region



$\forall i \in \{1, \cdots, m\}$

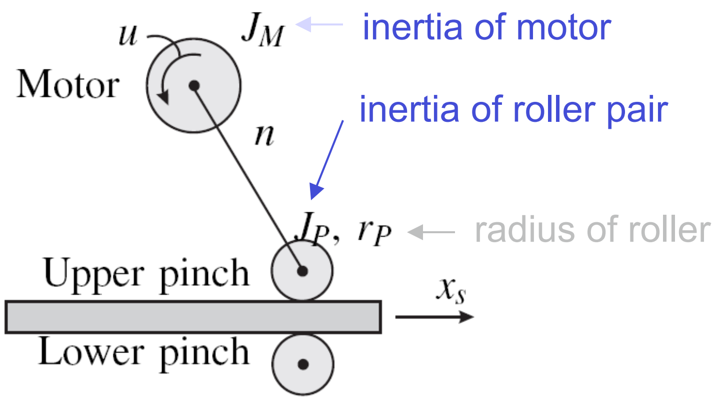
can be implemented, e.g., using CAN priorities

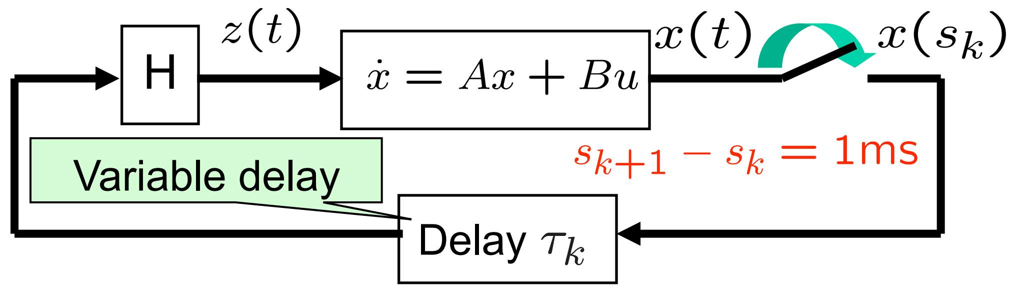
Then the following holds for EDF scheduling

$\tau^i_k = b^i_k + C^i_k \le \tau_{i\,max}$ $\Rightarrow$ exponential stability of the all closed loops

$$S = \bigcup_{i=1}^{n} \{\tau_{i\,\max} + h\rho_{i\,\min}, h = 0, 1, \cdots, \lfloor \frac{d - \tau_{i\,\max}}{\rho_{i\,\min}} \rfloor\}, d := \ldots. \quad \lfloor x \rfloor^{+} := \ldots.$$

Example: motion control system for sheet control





$$x = \begin{bmatrix} x_s \\ \dot{x}_s \end{bmatrix}, \quad A = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}, \quad B = -\begin{bmatrix} 0 \\ b \end{bmatrix} \times \begin{bmatrix} 50 & 11.8 \end{bmatrix}$$

$$b := \frac{n r_R}{J_M + n^2 J_R}$$

Controller gain

n: trans. ratio between motor and roller
x_s: sheet position
u: motor torque

- Position and velocity measurements are sent to an ECU through a CAN network
- ECU computes control commands and applies to motors directly, which takes *0.1ms*
- Transmission time is C_i= *1 ms* (*8* bytes, *64* kbit/s)
- Closed-loop system remains stable for any constant sampling smaller than *48 ms* when delay=0

(by solving $eig(\begin{bmatrix} I & 0 \\ I & 0 \end{bmatrix} e^{\begin{bmatrix} A & B \\ 0 & 0 \end{bmatrix} h}) \leq 1$)

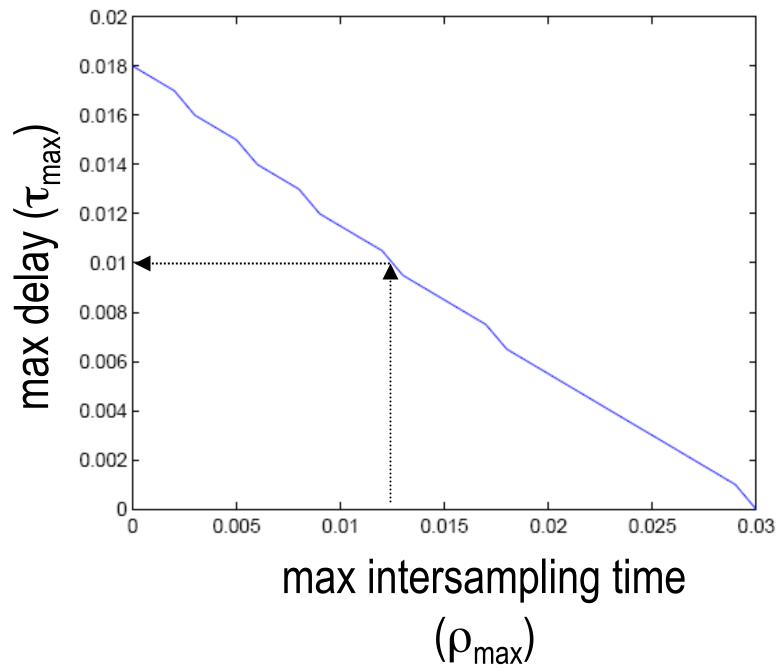
$\Rightarrow$ we choose sampling interval =*12 ms*

Example (continued)

- **How many motors can be controlled?**
- Traditional approach: a conservative designer $n=6$ so bus load *50%*
  an aggressive designer $n=11$ so bus load *<100% (91.7%)*

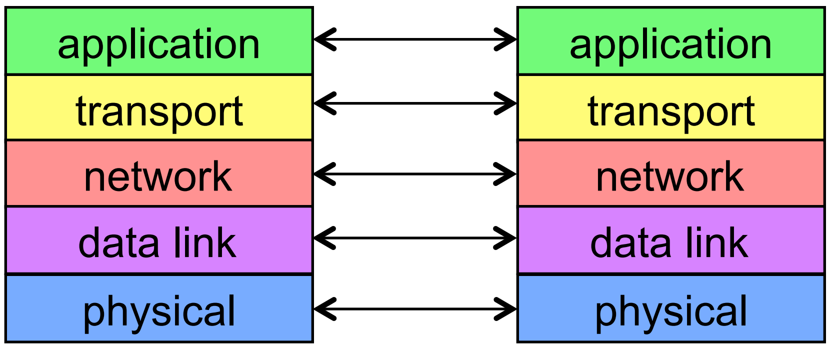
Our approach:

- By solving the LMIs we find admissible set of sampling-delays. For sampling=*12 ms*, max variable delay=*10ms*
- By testing scheduling condition with $T_i=12ms$, $D_i=10-0.1=9.9ms$, $C_i=1ms$ we conclude $n=9$ (bus load *75%)*
- By following the proposed method we avoid conservative choices and 'unsafe' choices

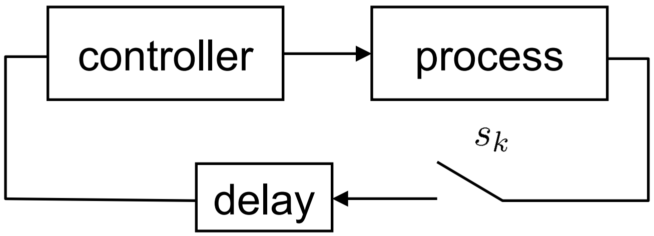


Network protocols & Control laws

network view:



control view:



This talk: ***Co-design of network protocols and control algorithms***

1. Characterize *key parameters* that determine the stability/performance of a networked controls system
2. Construct *protocols* that directly take these parameters into considerations

Illustrative examples:

- transport layer: error correction (& flow control) [ACC'09]
- data link layer: medium access control [Trans. Inst. Meas. Control, 2008]