Some results on optimal estimation and control for lossy NCS

DEPARTMENT OF INFORMATION ENGINEERING

UNIVERSITY OF PADOVA

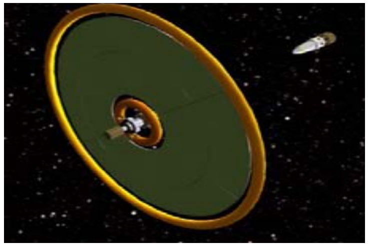
Luca Schenato

Networked Control Systems

Drive-by-wire systems

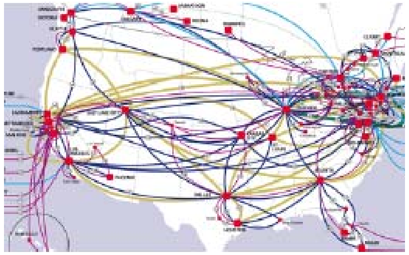
Swarm robotics

Smart structures:
adaptive space telescope

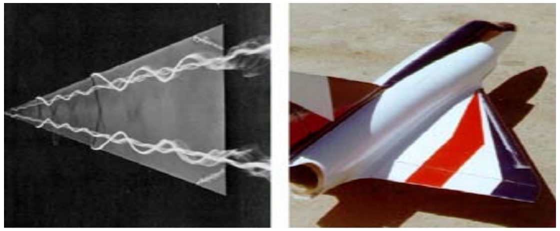


Wireless Sensor Networks

Traffic Control:
Internet and transportation



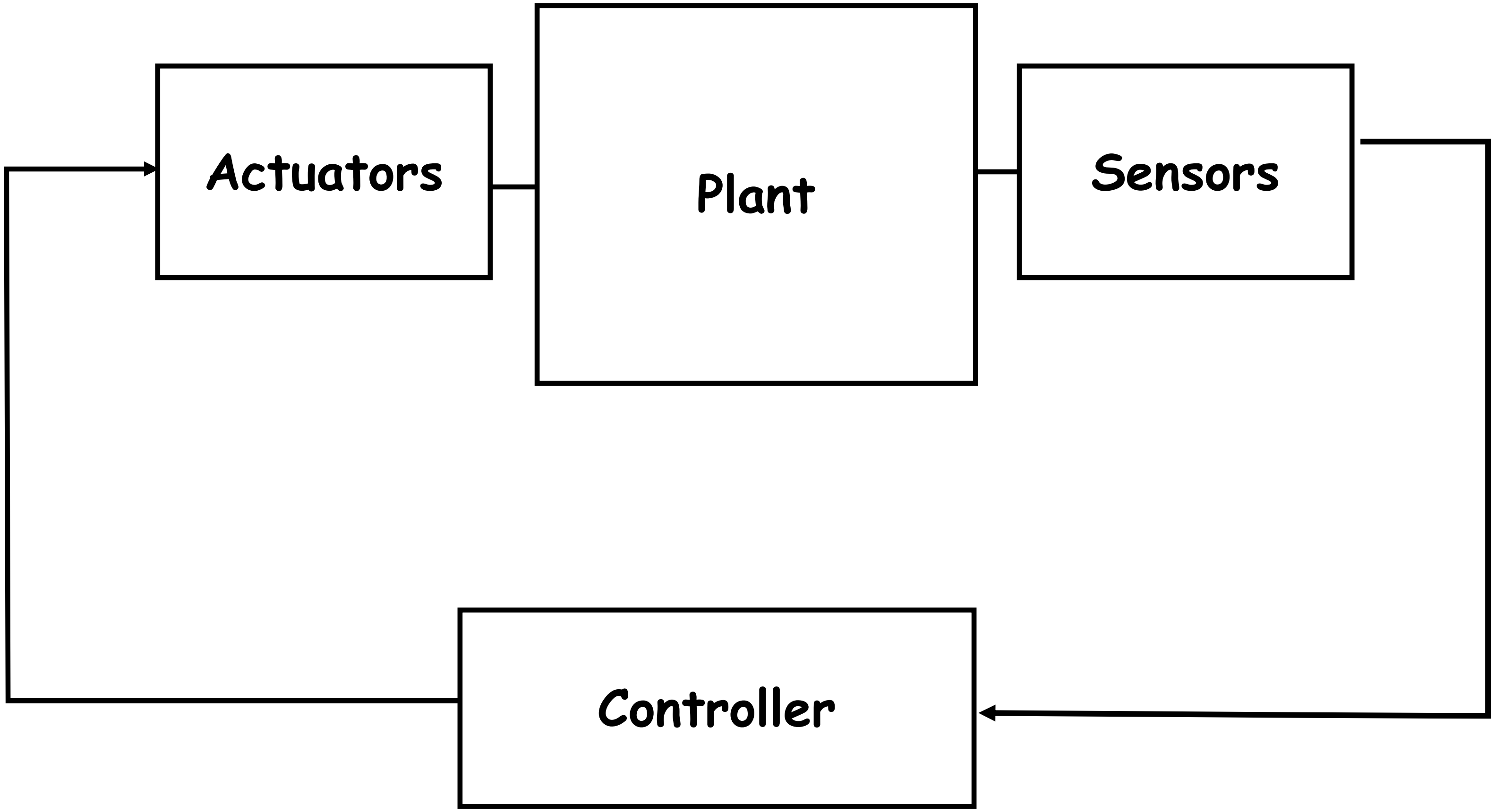
Smart materials:
sheets of MEMS
sensors and actuators



NCSs: physically distributed dynamical systems interconnected by a communication network

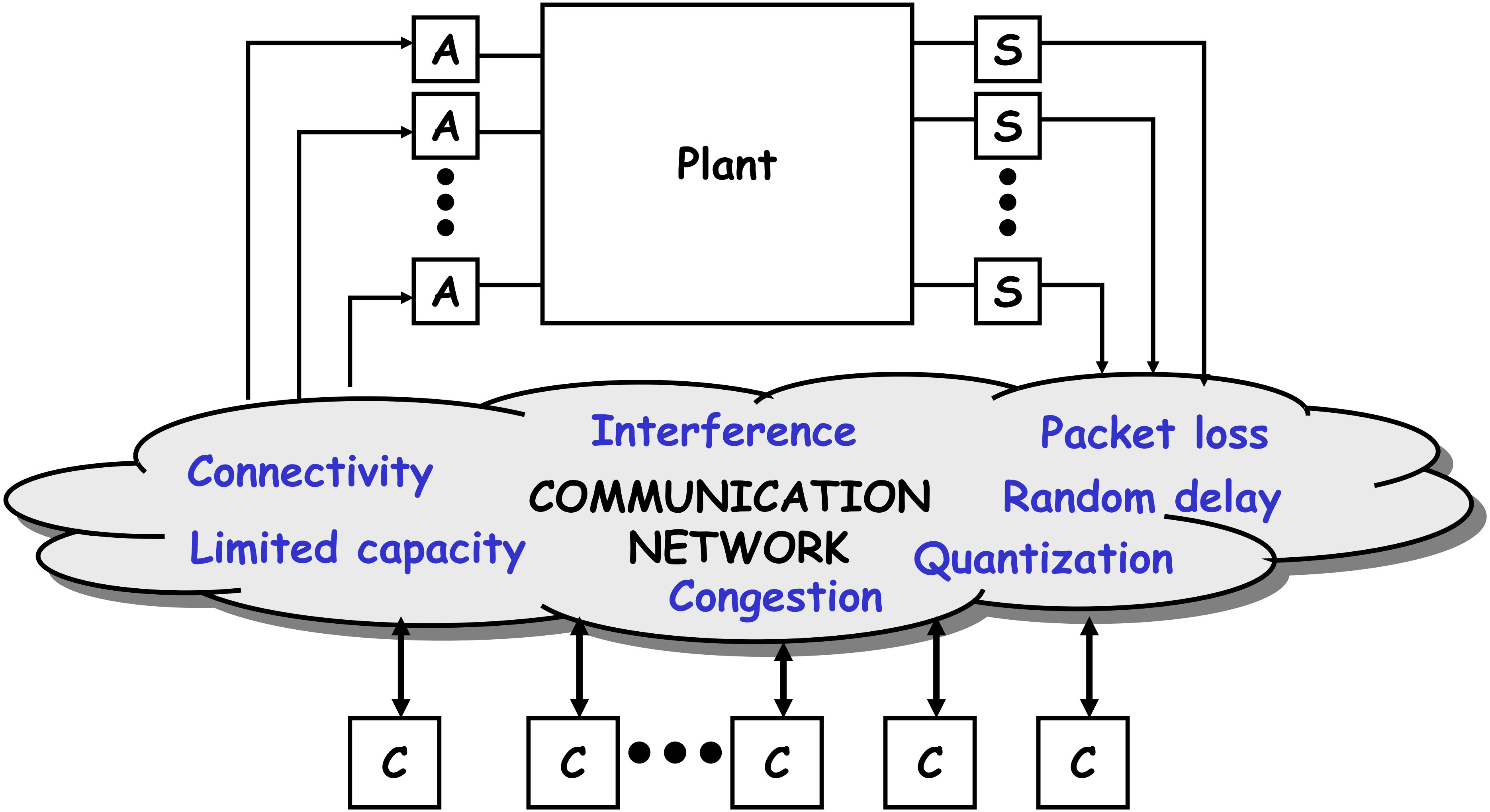
NCSs: what’s new for control?

Classical architecture: Centralized structure

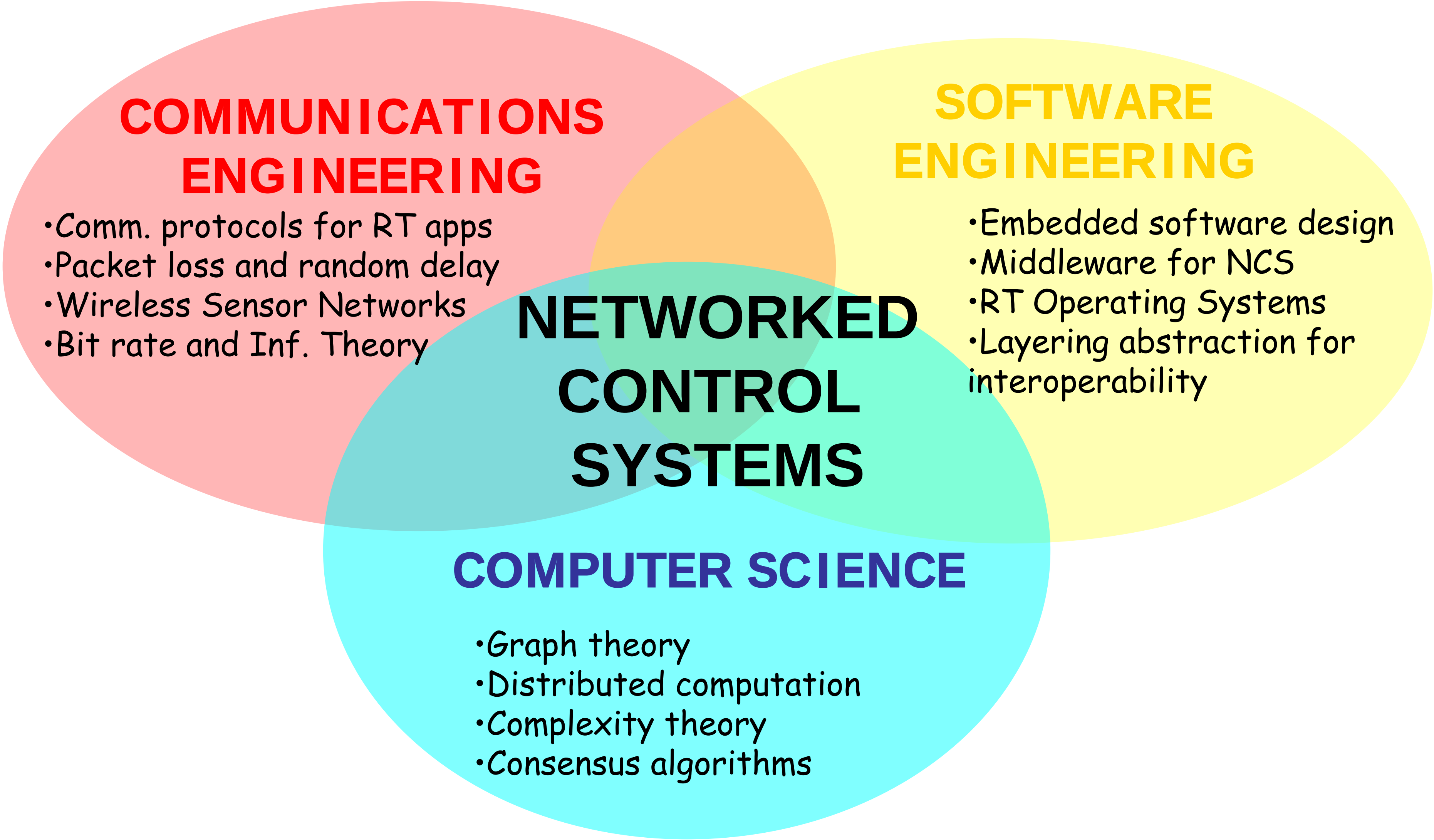


NCSs: what's new for control?

NCSs: Large scale distributed structure



Interdisciplinary research needed



COMMUNICATIONS ENGINEERING

- Comm. protocols for RT apps
- Packet loss and random delay
- Wireless Sensor Networks
- Bit rate and Inf. Theory

SOFTWARE ENGINEERING

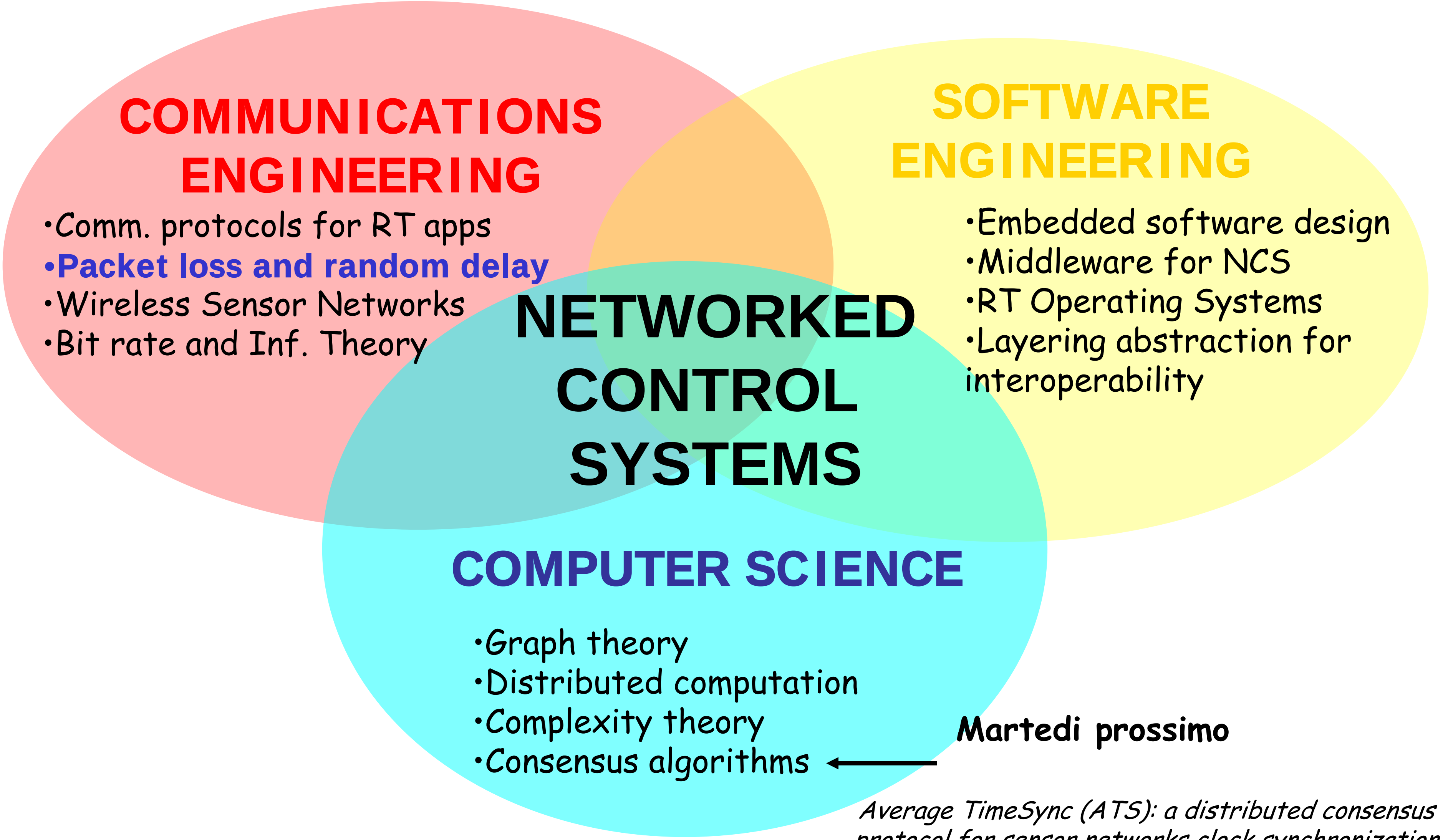
- Embedded software design
- Middleware for NCS
- RT Operating Systems
- Layering abstraction for interoperability

NETWORKED CONTROL SYSTEMS

COMPUTER SCIENCE

- Graph theory
- Distributed computation
- Complexity theory
- Consensus algorithms

Interdisciplinary research needed



COMMUNICATIONS ENGINEERING

- Comm. protocols for RT apps
- **Packet loss and random delay**
- Wireless Sensor Networks
- Bit rate and Inf. Theory

SOFTWARE ENGINEERING

- Embedded software design
- Middleware for NCS
- RT Operating Systems
- Layering abstraction for interoperability

NETWORKED CONTROL SYSTEMS

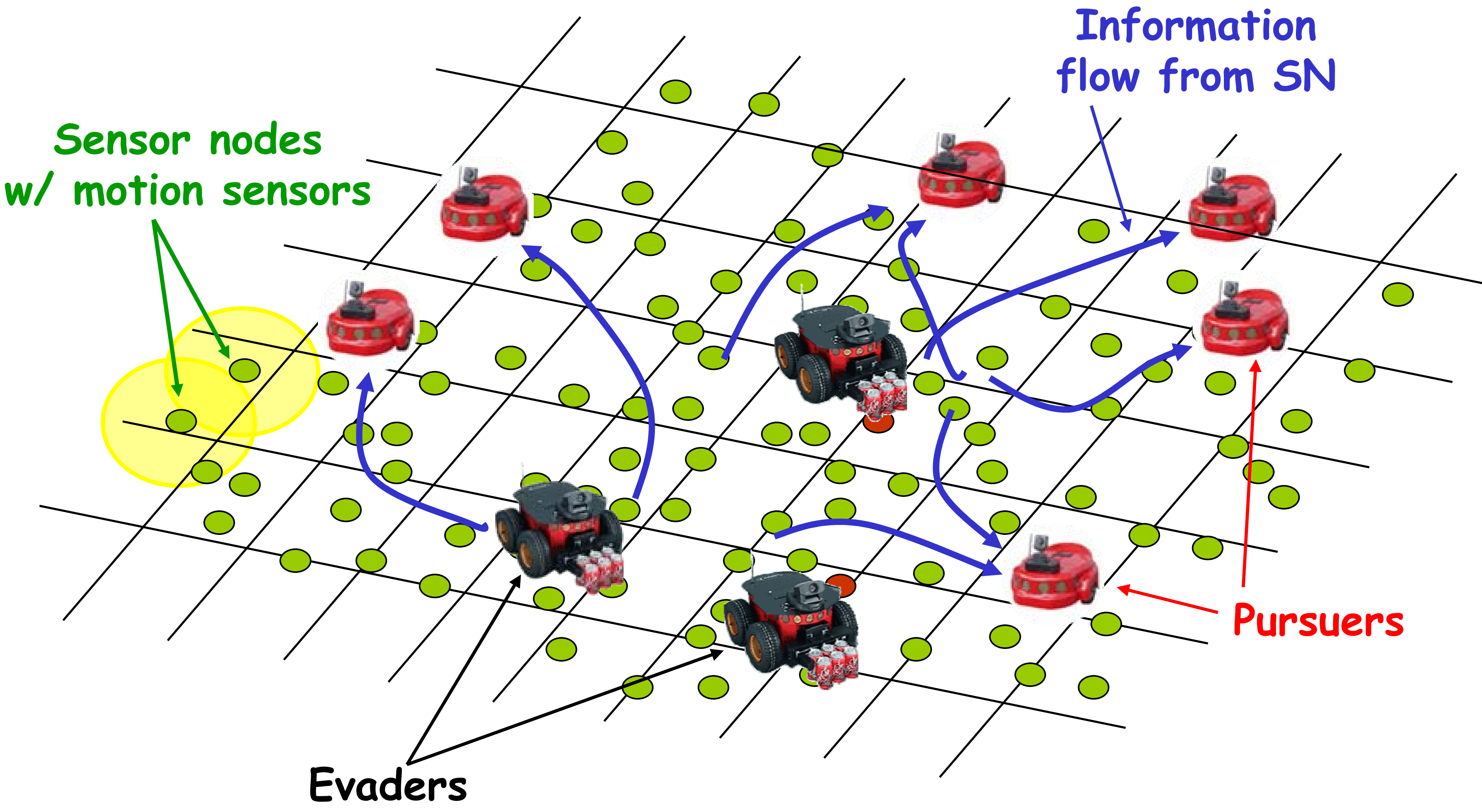
COMPUTER SCIENCE

- Graph theory
- Distributed computation
- Complexity theory
- Consensus algorithms ←

Martedi prossimo

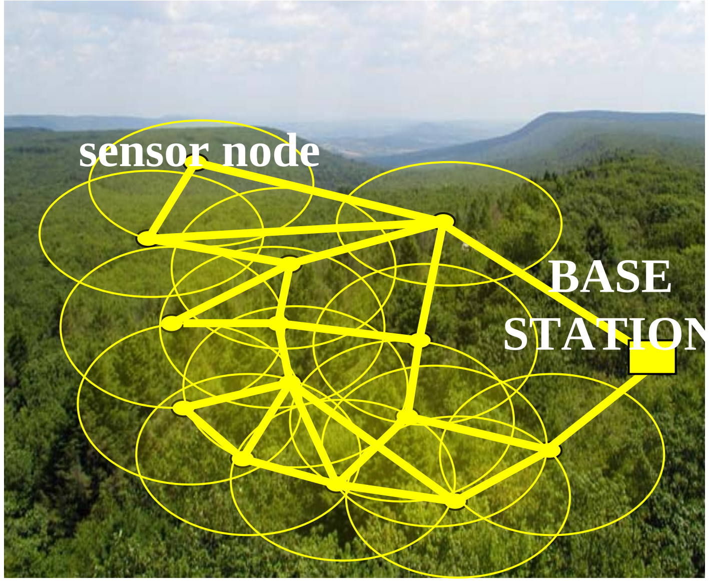
Average TimeSync (ATS): a distributed consensus protocol for sensor networks clock synchronization

NCS example: Pursuit Evasion Games w Sensor Networks

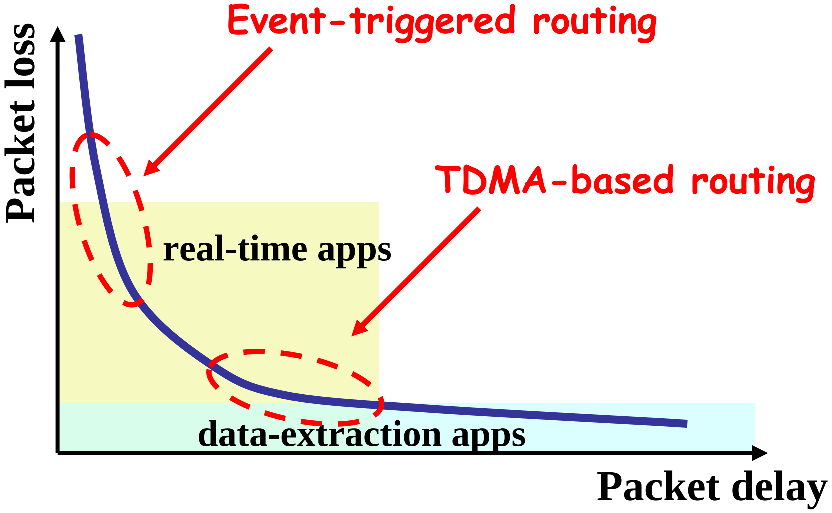


Motivating example: wireless sensor networks

Forest Temperature Monitoring (data-extraction application)

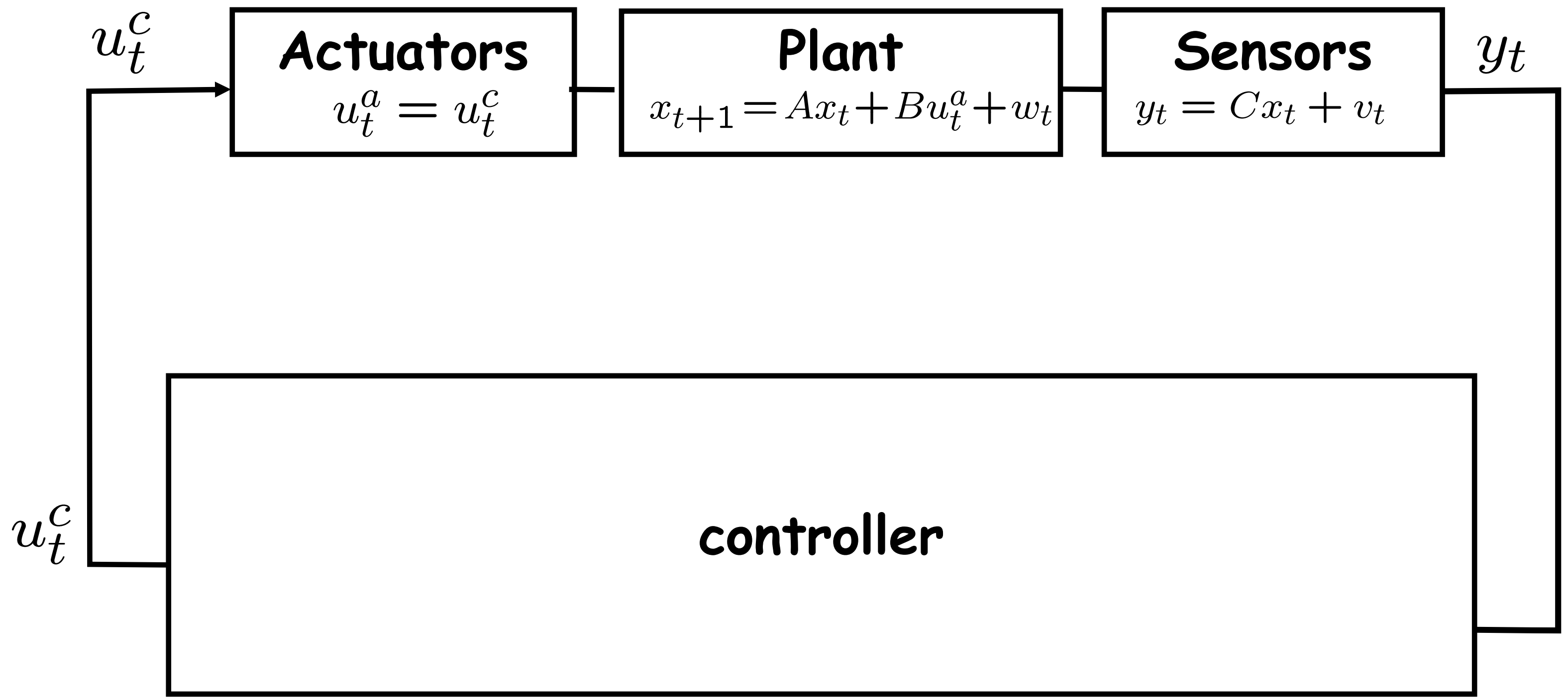


Wildfire detection & tracking (real-time application)



- Can we design optimal estimators that compensate for random delay and packet loss ?
- What is the performance if we have packet arrival statistics ?
- How can we compare different communication/routing protocols in terms of estimation performance ?

Optimal LQG



u_t^c

Actuators

$u_t^a = u_t^c$

Plant

$x_{t+1} = Ax_t + Bu_t^a + w_t$

Sensors

$y_t = Cx_t + v_t$

y_t

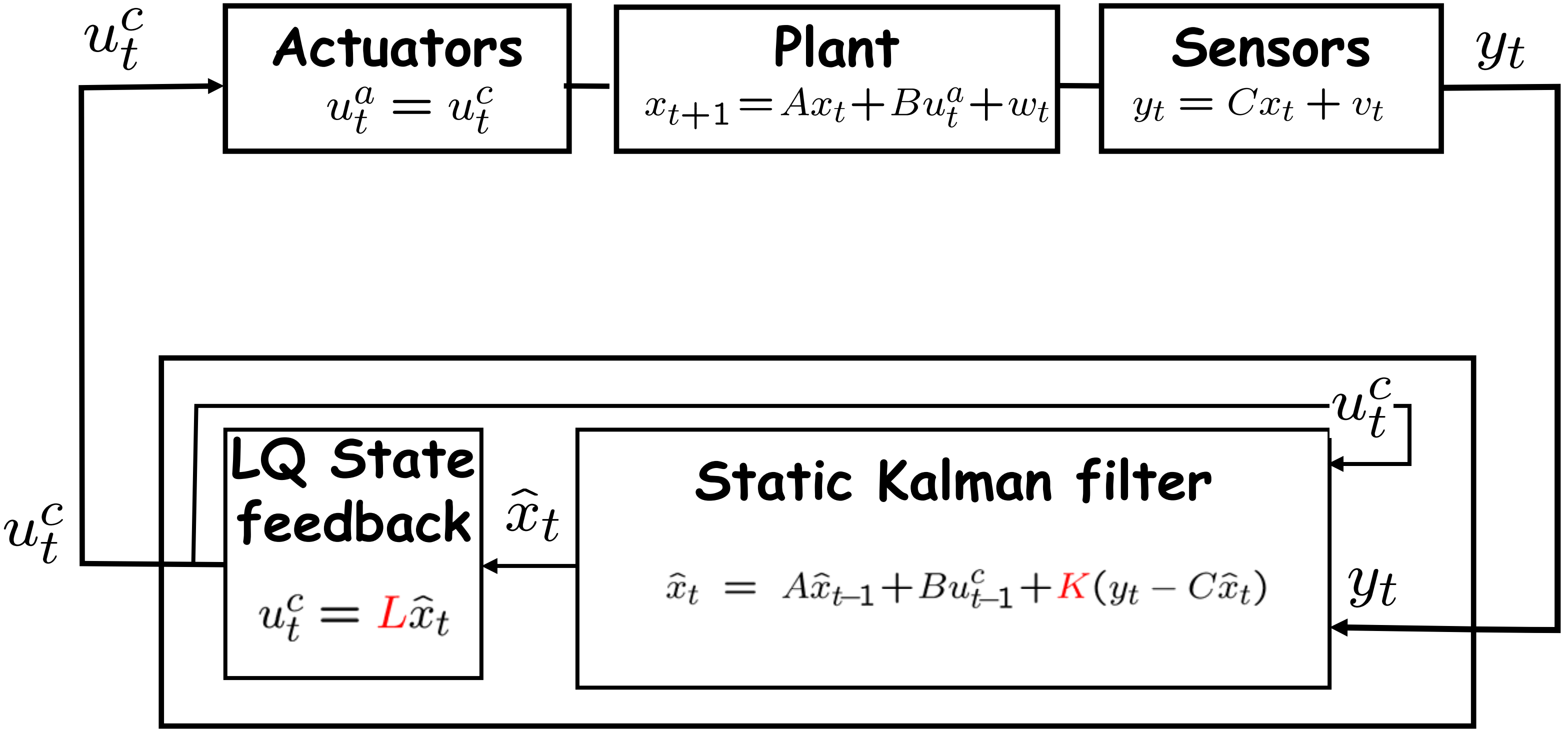
controller

u_t^c

$$\min_{u_1^c,\dots u_T^c} J = \sum_{t=1}^{T} E[x_t^T W x_t + u_t^T U u_t], \quad T \to \infty$$

Sensors and actuators are co-located, i.e. no delay nor loss

Optimal LQG



u_t^c

Actuators

$u_t^a = u_t^c$

Plant

$x_{t+1} = Ax_t + Bu_t^a + w_t$

Sensors

$y_t = Cx_t + v_t$

y_t

LQ State feedback

$u_t^c = L\hat{x}_t$

$\hat{x}_t$

Static Kalman filter

$\hat{x}_t = A\hat{x}_{t-1} + Bu_{t-1}^c + K(y_t - C\hat{x}_t)$

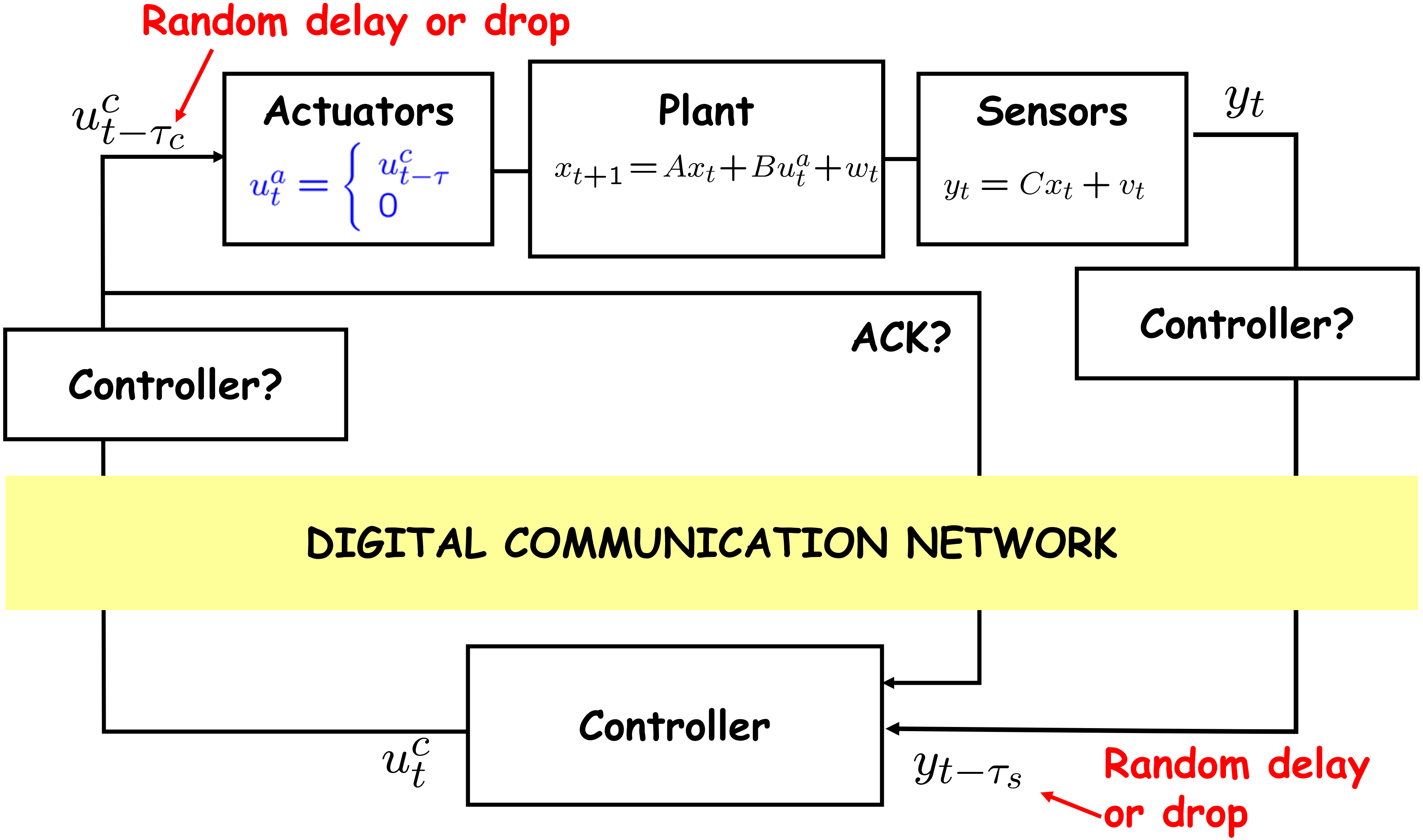
u_t^c

y_t

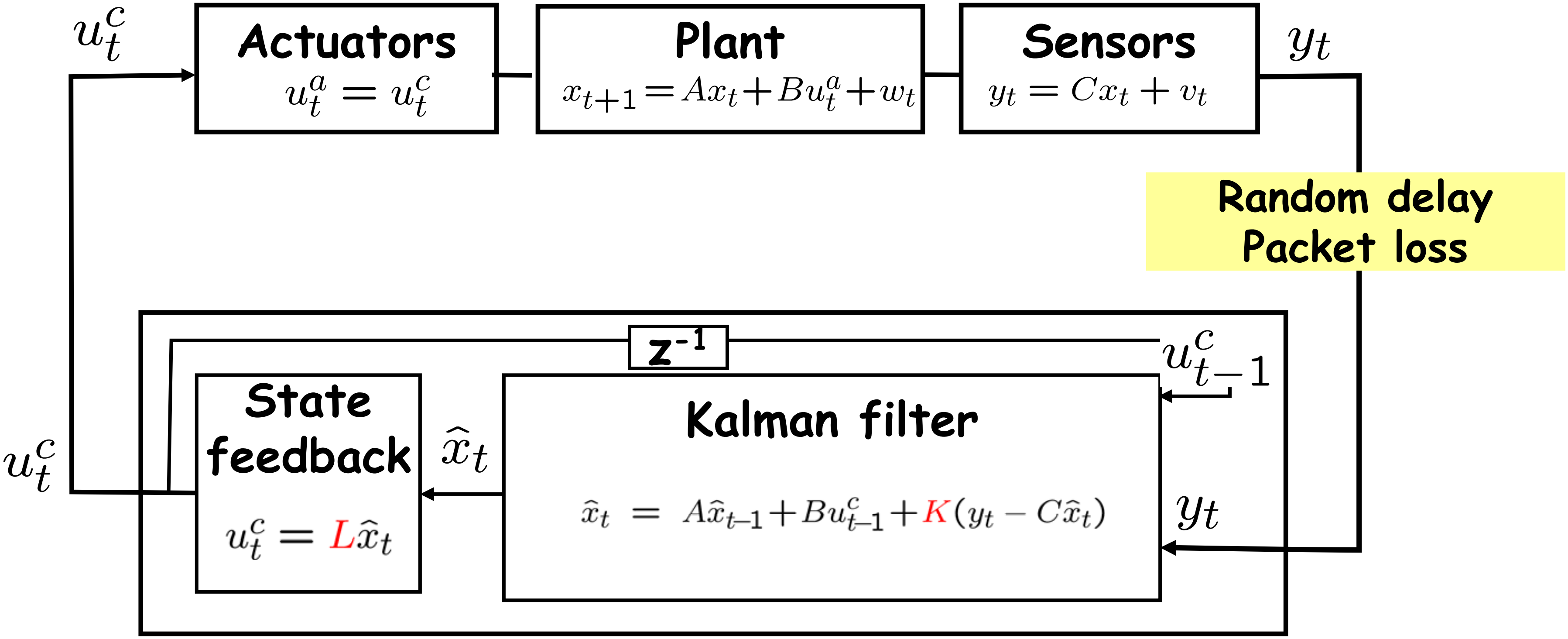
u_t^c

1. Separation principle holds: Optimal controller = Optimal estimator design + Optimal state feedback design
2. Closed Loop system always stable (under standard cont/obs. hypotheses)
3. Gains K,L are constant solution of Algebraic Riccati Equations

Optimal LQG control over DCN



Some consideration on the separation principle

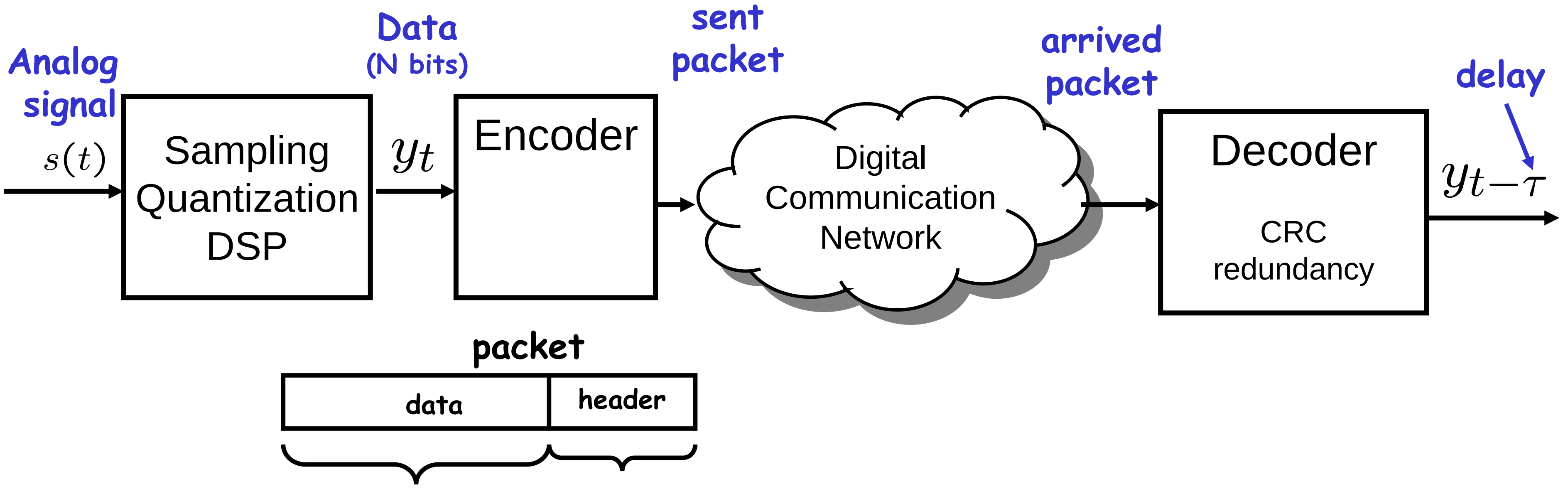


$$
\begin{aligned}
\hat{x}_t &= E[x_t | y_t, y_{t-1}, .., y_0, u_{t-1}^a, .., u_1^a] \\
&= f(y_t, y_{t-1}, .., y_0)
\end{aligned}
$$

if $(u_{t-1}^a, \ldots u_1^a)$ known $\Longrightarrow e_t = x_t - \hat{x}_t = f(y_t, \cancel{y_{t-1}}, .., \cancel{y_1}, y_0)$

DEPARTMENT OF INFORMATION ENGINEERING
UNIVERSITY OF PADOVA

Modeling of Digital Communication Network (DCN)



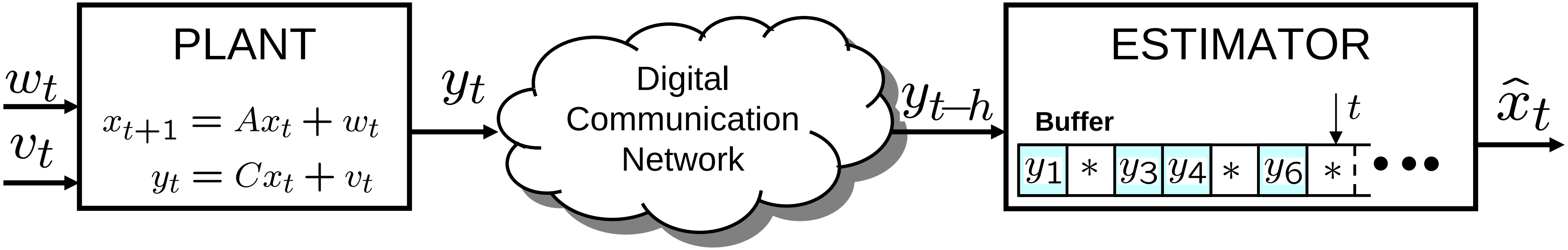
ATM	384 bits	40 bits
Ethernet	>368 bits	112 bits
Bluetooth	>499 bits	~100 bits
Zigbee	<1000 bits	128 bits

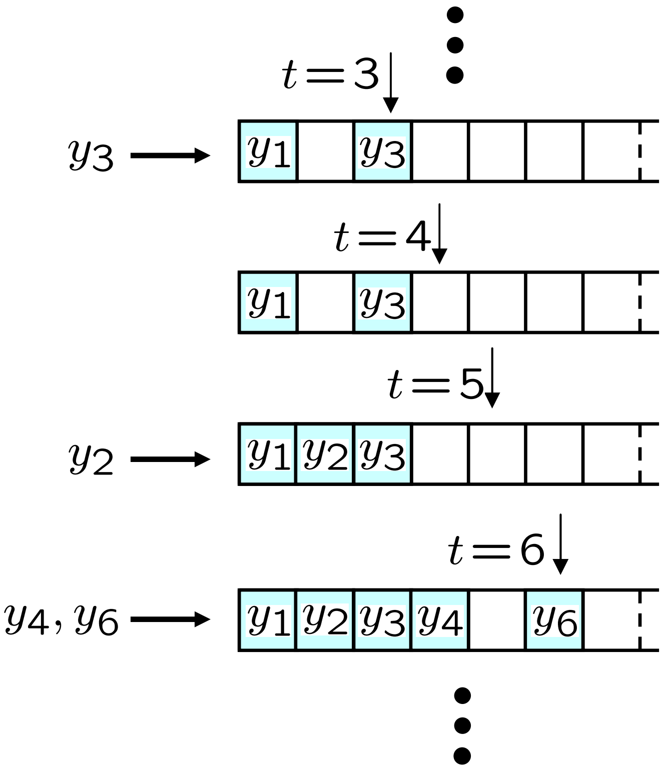
Assumptions:

(1) Quantization noise<<sensor noise
(2) Packet-rate limited (□ bit-rate)
(3) No transmission noise (data corrupted=dropped packet)
(4) Packets are time-stamped

Random delay
&
Packet loss (□=1)
at receiver

Estimation modeling





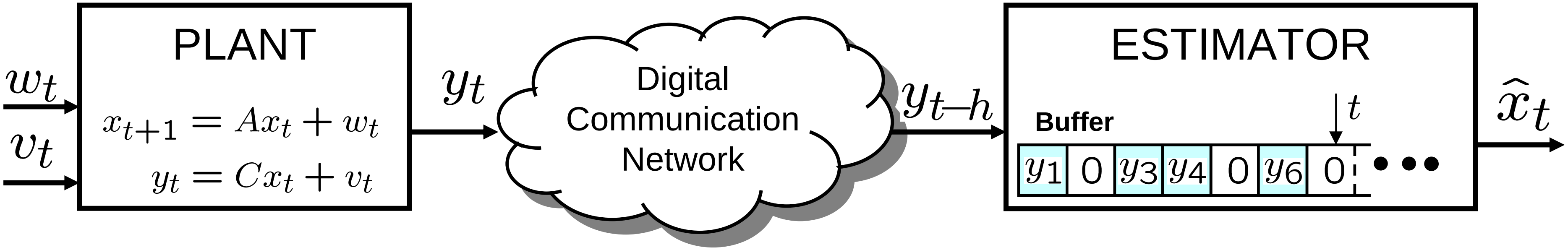
No packet arrives

Packet out of order

Multiple packets arrive

Minimum variance estimation

$$\hat{x}_t = \mathbb{E}[x_t | \{y_k\} \text{ available at estimator at time } t]$$



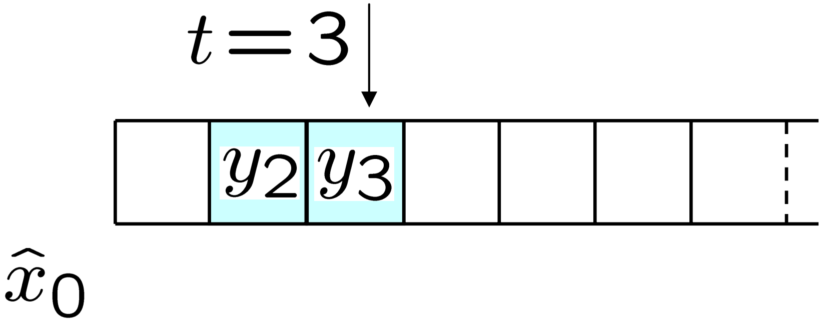
$$\gamma_k^t = \begin{cases} 1 & \textit{if } y_k \text{ arrived before or at time } t, \; t \geq k \\ 0 & \textit{otherwise} \end{cases}$$

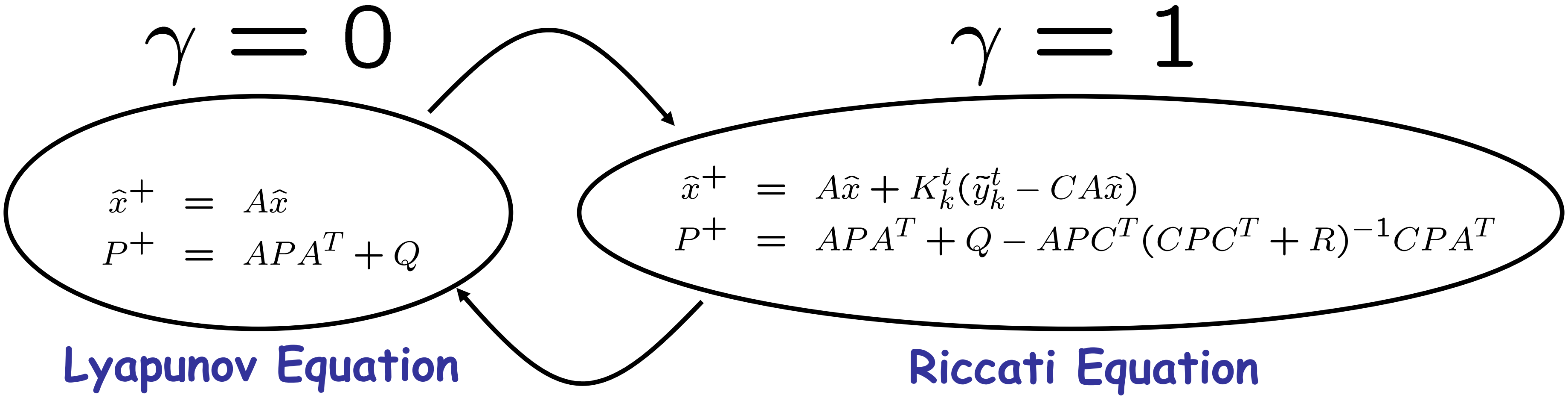
$$\tilde{y}_k = \gamma_k^t (Cx_k + v_k) = C_k^t x_k + u^t$$

Kalman time-varying linear system

$$\hat{x}_t = \mathbb{E}[x_t \,|\, \tilde{y}_1, \ldots, \tilde{y}_t, \gamma_1^t, \ldots, \gamma_t^t]$$

Minimum variance estimation





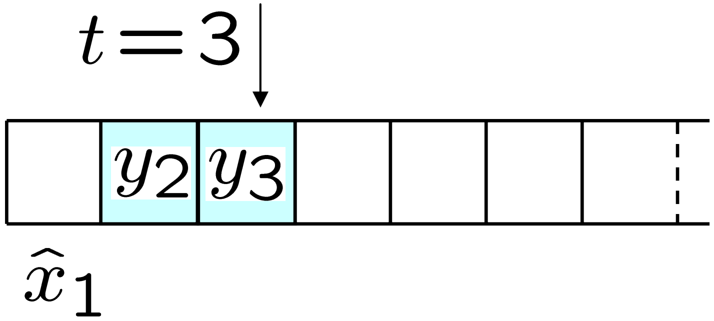
$$\begin{aligned} \hat{x}^+ &= A\hat{x} \\ P^+ &= APA^T + Q \end{aligned}$$

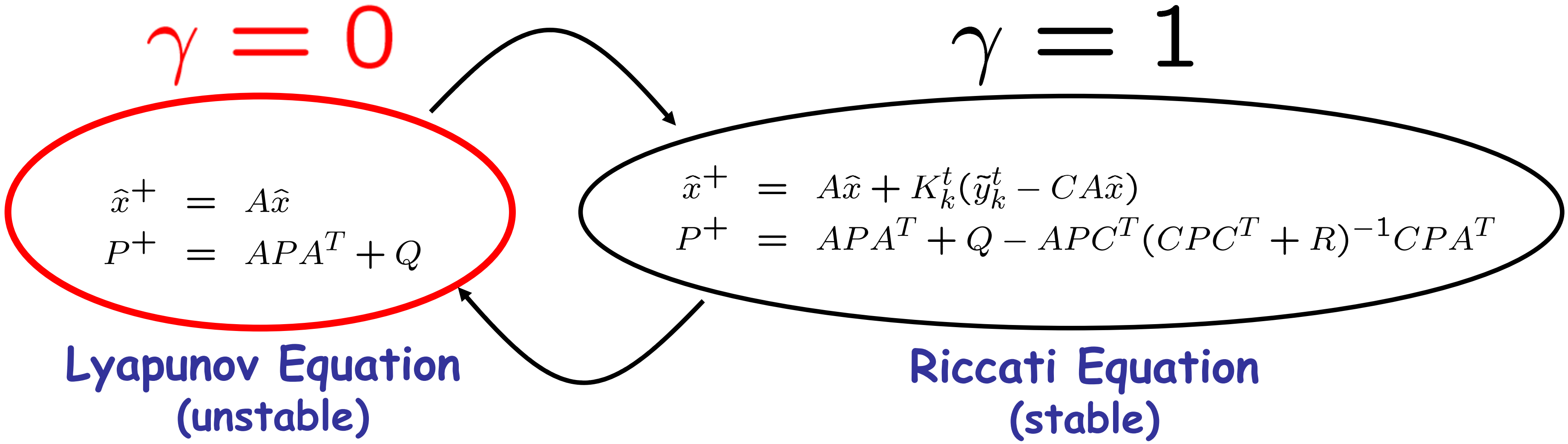
$$\begin{aligned} \hat{x}^+ &= A\hat{x} + K_k^t(\tilde{y}_k^t - CA\hat{x}) \\ P^+ &= APA^T + Q - APC^T(CPC^T + R)^{-1}CPA^T \end{aligned}$$

Lyapunov Equation
(unstable)

Riccati Equation
(stable)

Minimum variance estimation





$\gamma = 0$

$$\begin{aligned} \hat{x}^+ &= A\hat{x} \\ P^+ &= APA^T + Q \end{aligned}$$

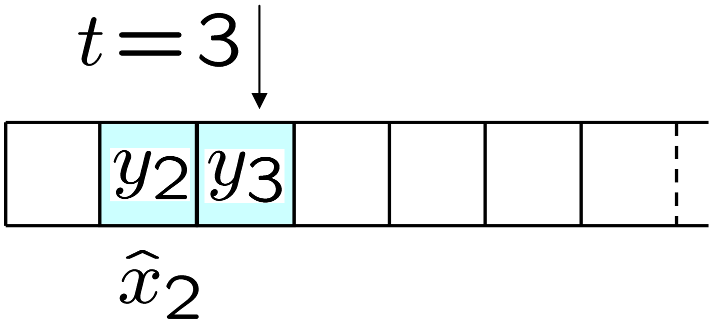
Lyapunov Equation
(unstable)

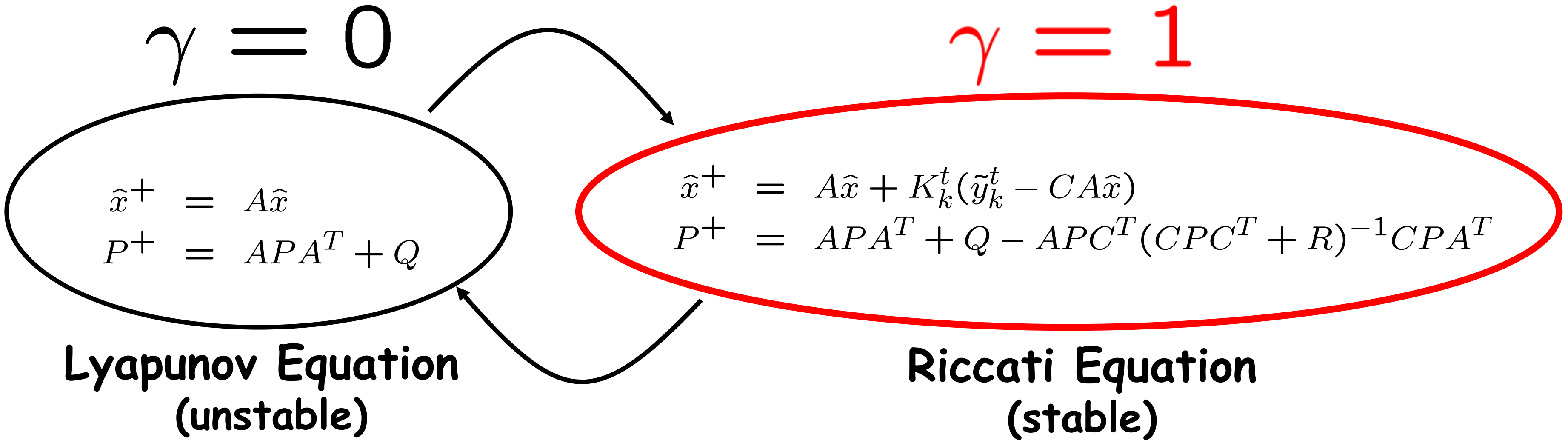
$\gamma = 1$

$$\begin{aligned} \hat{x}^+ &= A\hat{x} + K_k^t(\tilde{y}_k^t - CA\hat{x}) \\ P^+ &= APA^T + Q - APC^T(CPC^T + R)^{-1}CPA^T \end{aligned}$$

Riccati Equation
(stable)

Minimum variance estimation





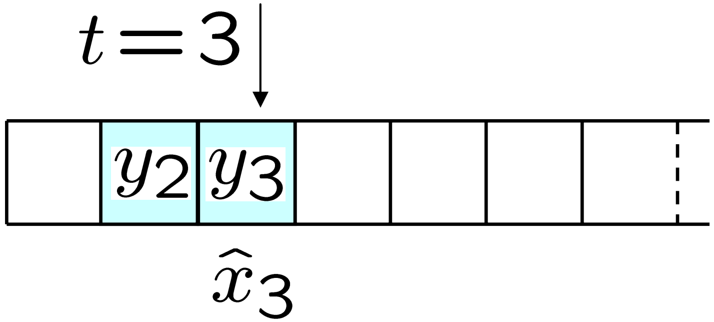
$$\begin{aligned} \hat{x}^+ &= A\hat{x} \\ P^+ &= APA^T + Q \end{aligned}$$

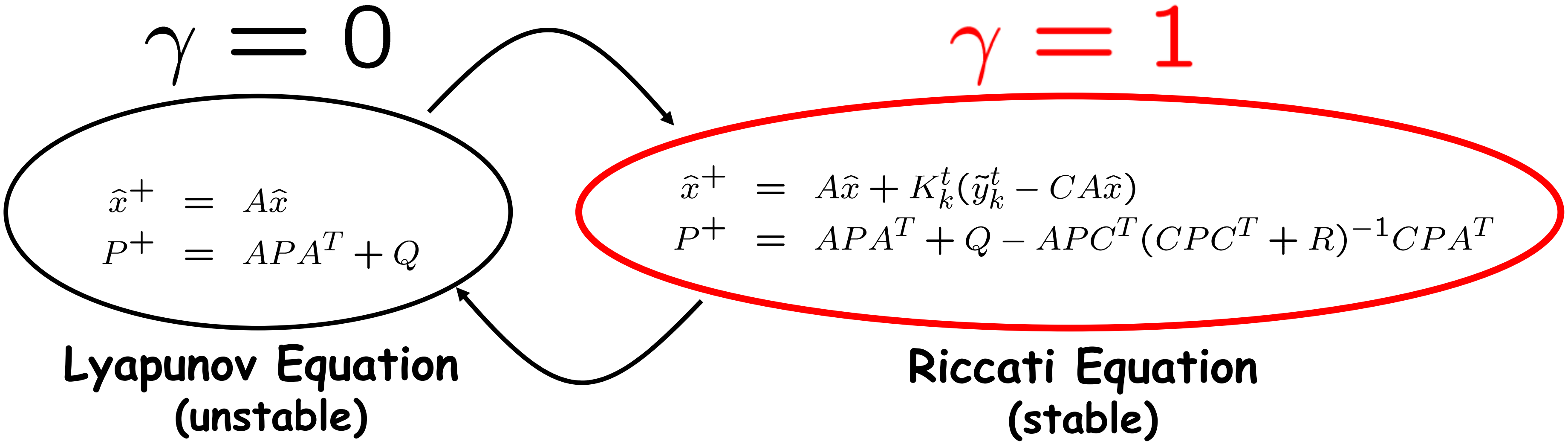
Lyapunov Equation (unstable)

$$\begin{aligned} \hat{x}^+ &= A\hat{x} + K_k^t(\tilde{y}_k^t - CA\hat{x}) \\ P^+ &= APA^T + Q - APC^T(CPC^T + R)^{-1}CPA^T \end{aligned}$$

Riccati Equation (stable)

Minimum variance estimation





$\gamma = 0$

$$\begin{aligned} \hat{x}^+ &= A\hat{x} \\ P^+ &= APA^T + Q \end{aligned}$$

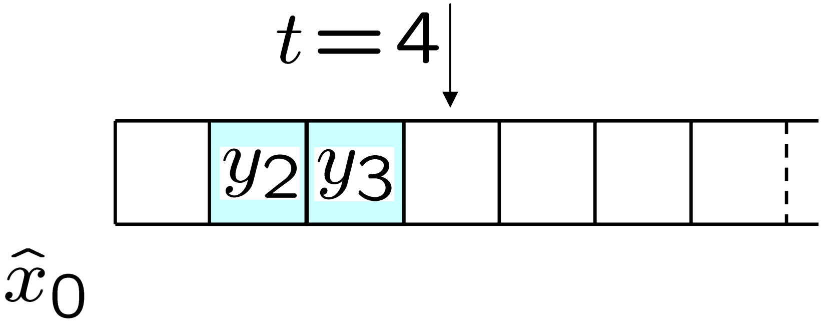
Lyapunov Equation
(unstable)

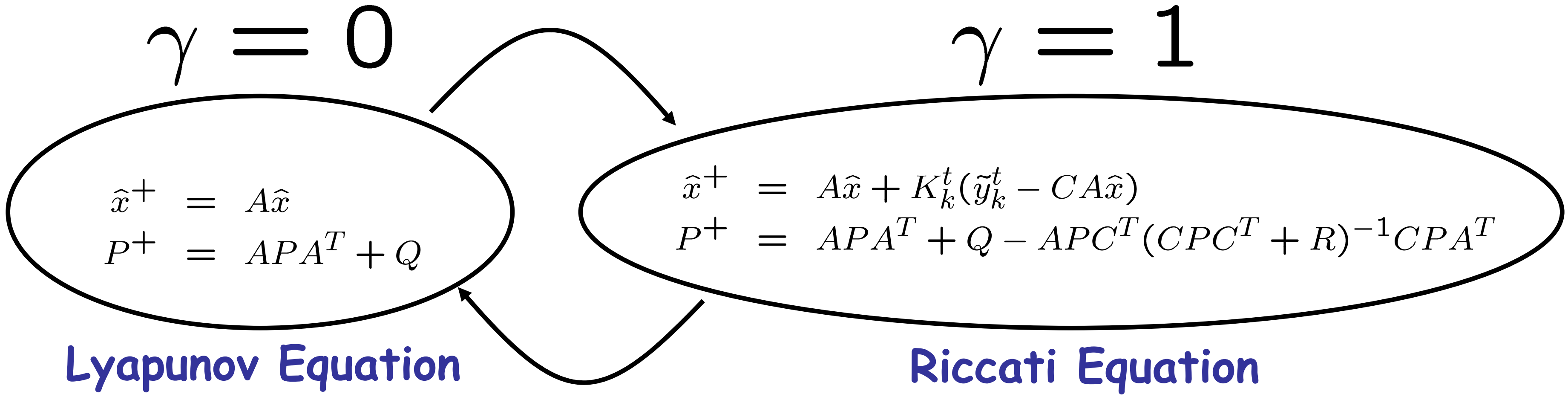
$\gamma = 1$

$$\begin{aligned} \hat{x}^+ &= A\hat{x} + K_k^t(\tilde{y}_k^t - CA\hat{x}) \\ P^+ &= APA^T + Q - APC^T(CPC^T + R)^{-1}CPA^T \end{aligned}$$

Riccati Equation
(stable)

Minimum variance estimation





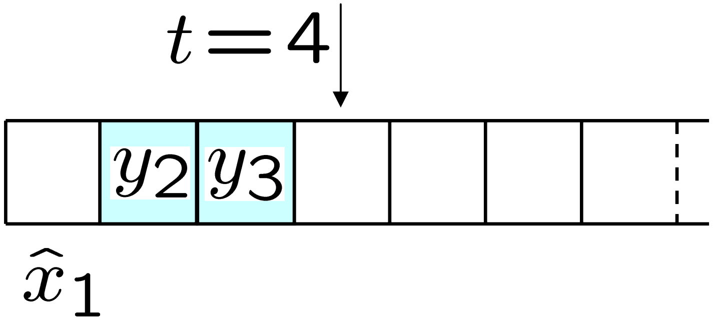
$$\begin{aligned}\hat{x}^+ &= A\hat{x}\\ P^+ &= APA^T + Q\end{aligned}$$

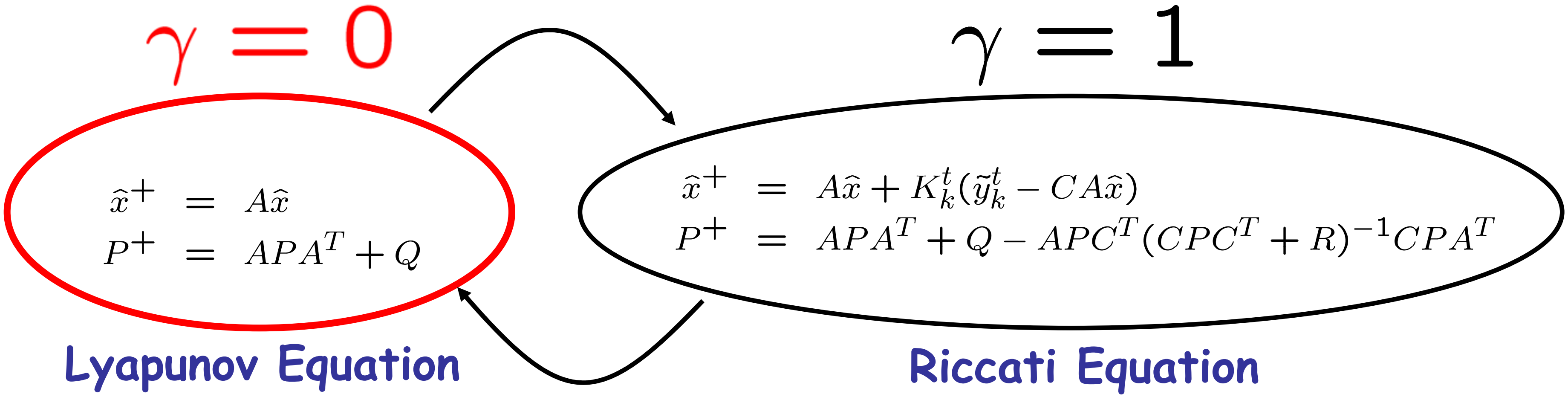
$$\begin{aligned}\hat{x}^+ &= A\hat{x} + K_k^t(\tilde{y}_k^t - CA\hat{x})\\ P^+ &= APA^T + Q - APC^T(CPC^T + R)^{-1}CPA^T\end{aligned}$$

Lyapunov Equation
(unstable)

Riccati Equation
(stable)

Minimum variance estimation





$\gamma = 0$

$$\begin{aligned} \hat{x}^+ &= A\hat{x} \\ P^+ &= APA^T + Q \end{aligned}$$

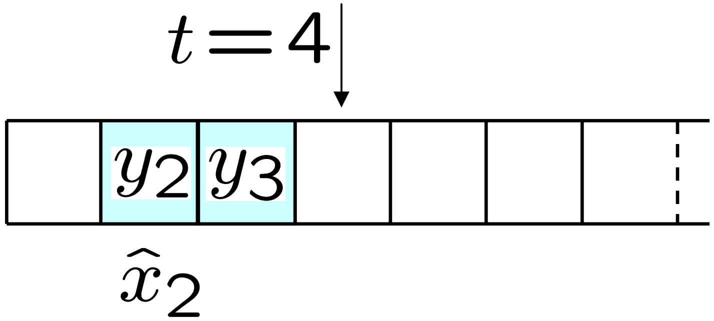
$\gamma = 1$

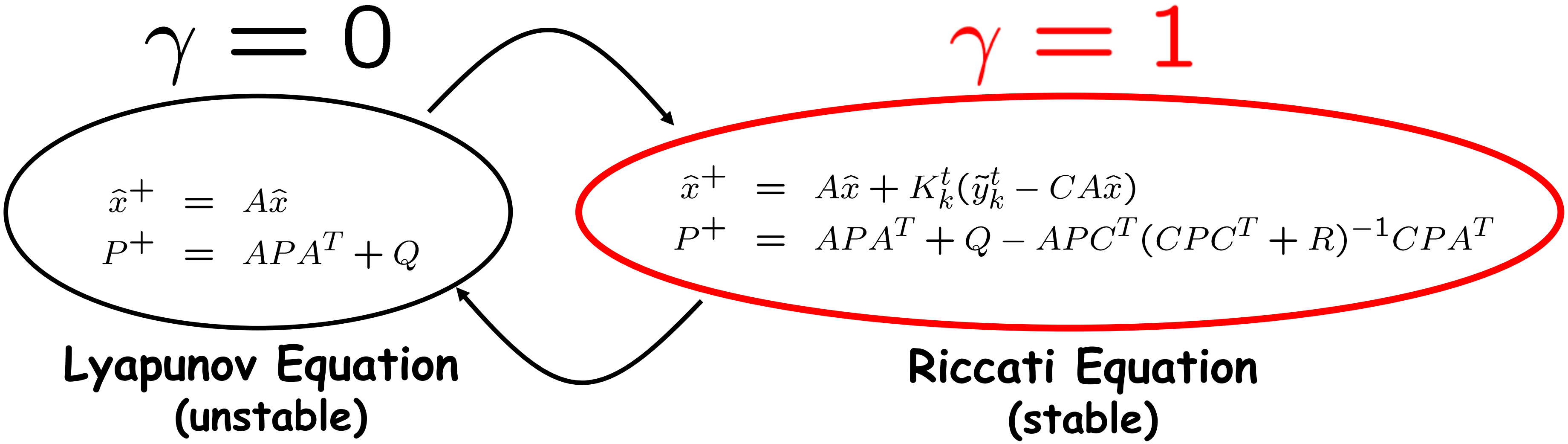
$$\begin{aligned} \hat{x}^+ &= A\hat{x} + K_k^t(\tilde{y}_k^t - CA\hat{x}) \\ P^+ &= APA^T + Q - APC^T(CPC^T + R)^{-1}CPA^T \end{aligned}$$

Lyapunov Equation
(unstable)

Riccati Equation
(stable)

Minimum variance estimation





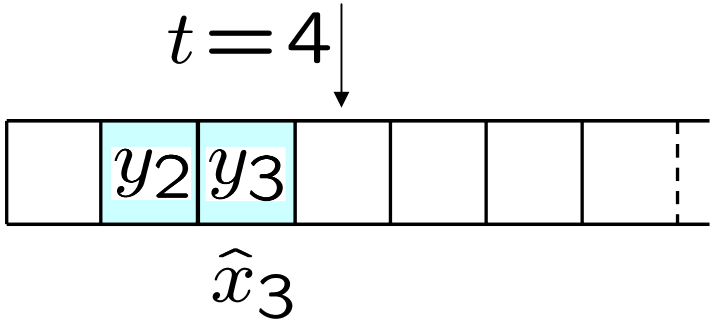
$$\begin{aligned}\hat{x}^+ &= A\hat{x}\\ P^+ &= APA^T + Q\end{aligned}$$

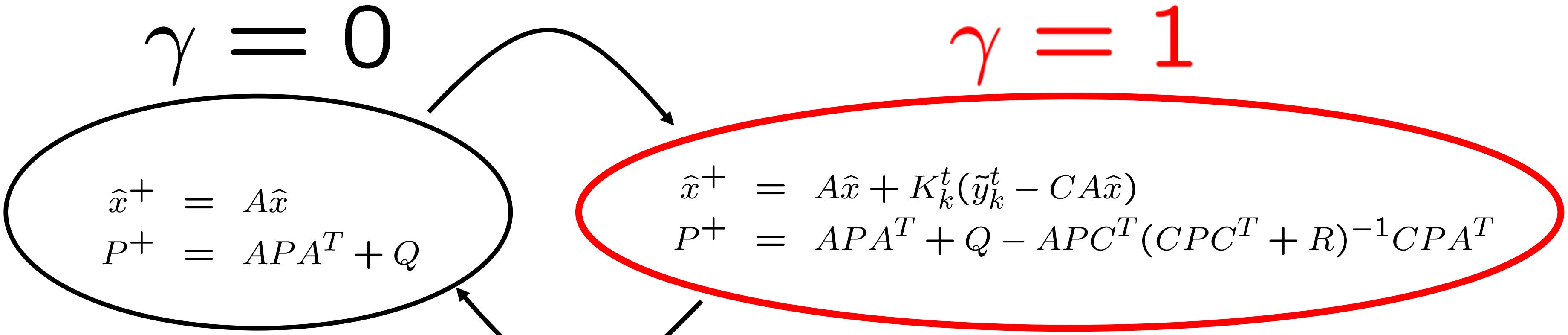
Lyapunov Equation
(unstable)

$$\begin{aligned}\hat{x}^+ &= A\hat{x} + K_k^t(\tilde{y}_k^t - CA\hat{x})\\ P^+ &= APA^T + Q - APC^T(CPC^T + R)^{-1}CPA^T\end{aligned}$$

Riccati Equation
(stable)

Minimum variance estimation





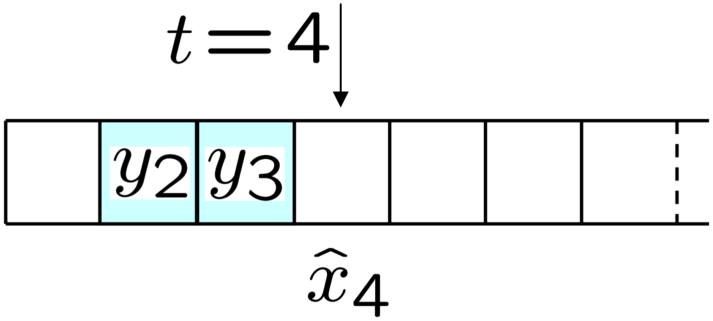
$$\begin{aligned}\hat{x}^+ &= A\hat{x}\\ P^+ &= APA^T + Q\end{aligned}$$

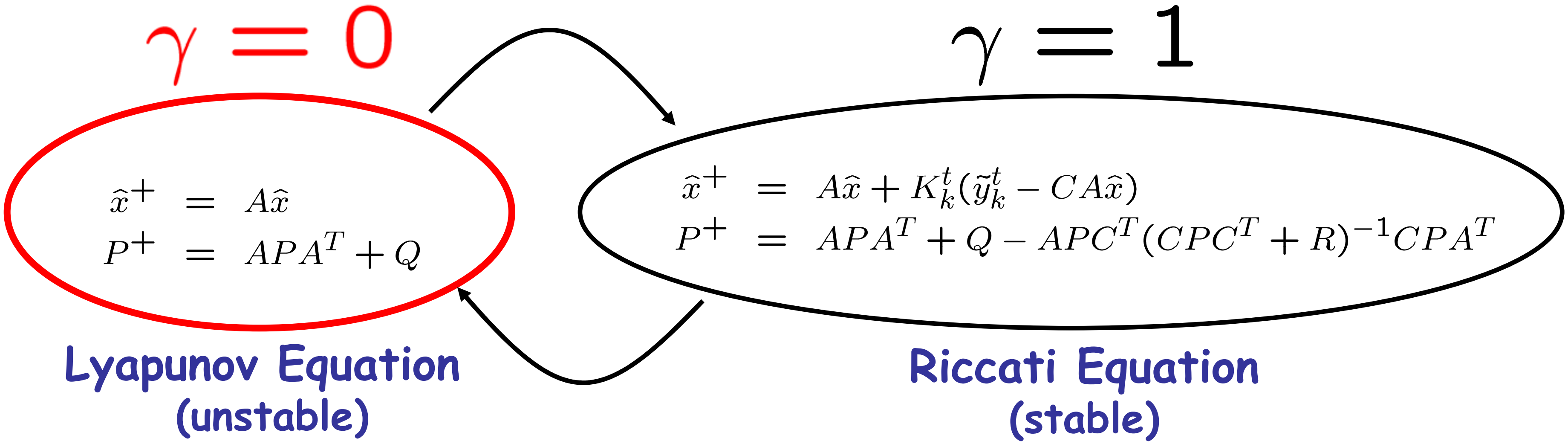
$$\begin{aligned}\hat{x}^+ &= A\hat{x} + K_k^t(\tilde{y}_k^t - CA\hat{x})\\ P^+ &= APA^T + Q - APC^T(CPC^T + R)^{-1}CPA^T\end{aligned}$$

Lyapunov Equation
(unstable)

Riccati Equation
(stable)

Minimum variance estimation





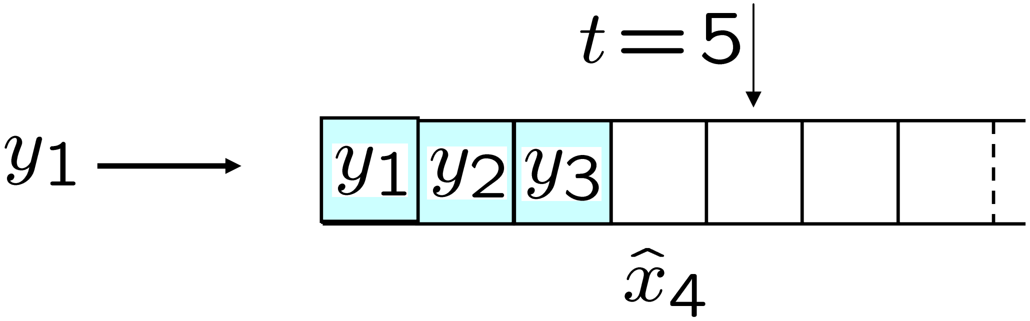
$$\begin{aligned} \hat{x}^+ &= A\hat{x} \\ P^+ &= APA^T + Q \end{aligned}$$

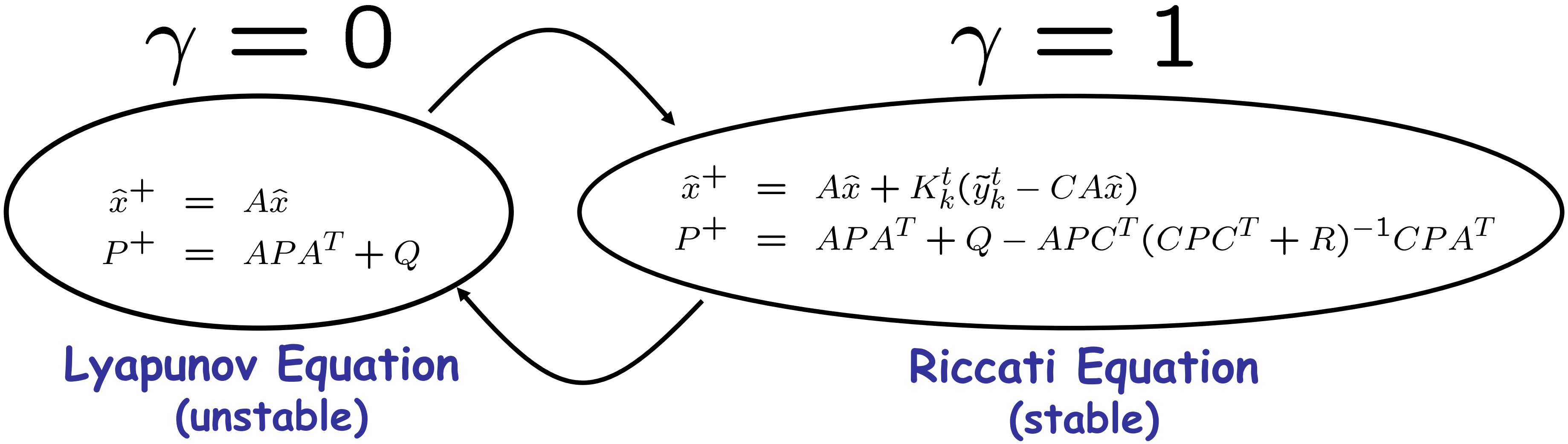
$$\begin{aligned} \hat{x}^+ &= A\hat{x} + K_k^t(\tilde{y}_k^t - CA\hat{x}) \\ P^+ &= APA^T + Q - APC^T(CPC^T + R)^{-1}CPA^T \end{aligned}$$

Lyapunov Equation
(unstable)

Riccati Equation
(stable)

Minimum variance estimation





$\gamma = 0$

$$\begin{aligned}\hat{x}^+ &= A\hat{x}\\ P^+ &= APA^T + Q\end{aligned}$$

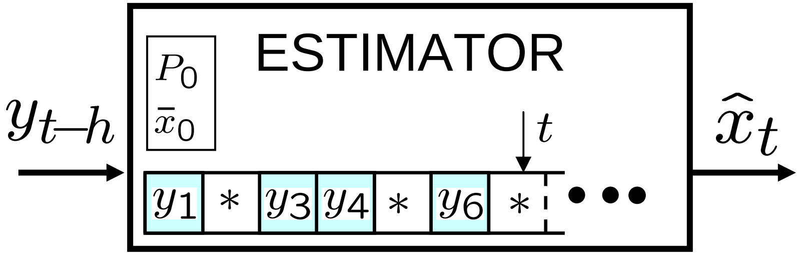
Lyapunov Equation
(unstable)

$\gamma = 1$

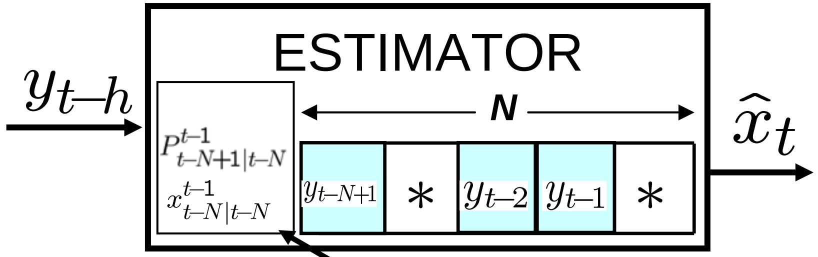
$$\begin{aligned}\hat{x}^+ &= A\hat{x} + K_k^t(\tilde{y}_k^t - CA\hat{x})\\ P^+ &= APA^T + Q - APC^T(CPC^T + R)^{-1}CPA^T\end{aligned}$$

Riccati Equation
(stable)

Properties of Optimal Estimator



- Optimal for any arrival process
- Stochastic time-varying gain $K_t=K(\gamma_1,..,\gamma_t)$
- Possibly infinite memory buffer
- Inversion of up to *t* matrices at any time *t*



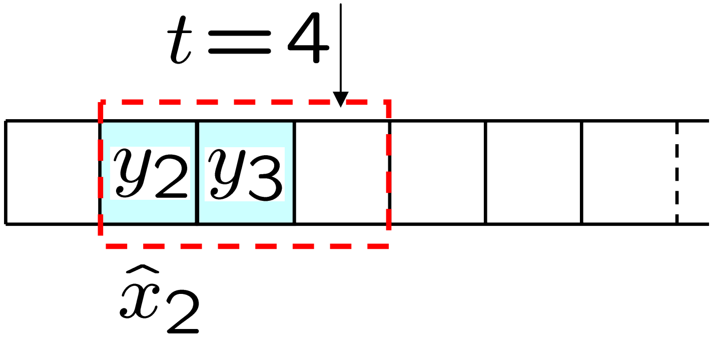
$\Longleftrightarrow$

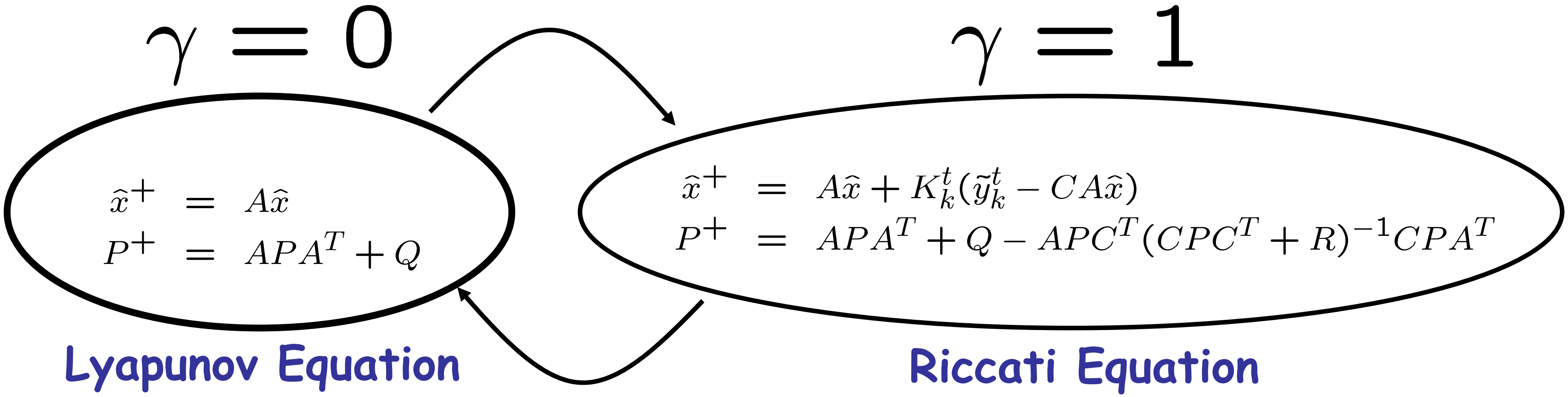
$$\gamma_k^t = \text{cost}, \;\; t \geq k+\tau_{max}$$
$$\tau_{max} = N, \quad \text{delay}$$

$$\hat{x}^{t-1}_{t-N|t-N} \triangleq \hat{x}$$
$$P^{t-1}_{t-N+1|t-N} \triangleq P$$

$$\boxed{\begin{aligned} \hat{x}^+ &= A\hat{x} + \gamma^t_{t-N} PC^T(CPC^T+R)^{-1}(\tilde{y}^t_{t-N} - CA\hat{x}), \\ P^+ &= APA^T + Q - \gamma^t_{t-N} APC^T(CPC^T+R)^{-1}CPA^T \end{aligned}}$$

Minimum variance estimation





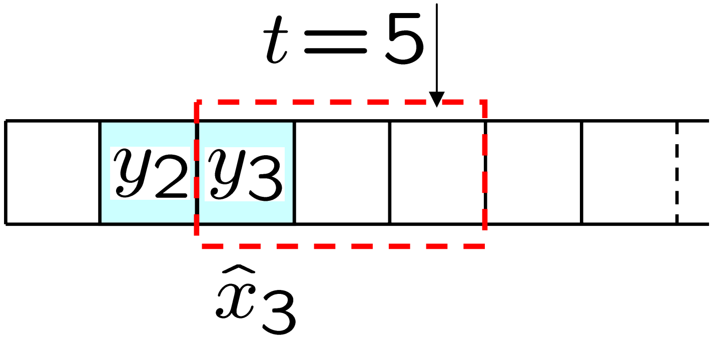
$$\begin{aligned} \hat{x}^+ &= A\hat{x} \\ P^+ &= APA^T + Q \end{aligned}$$

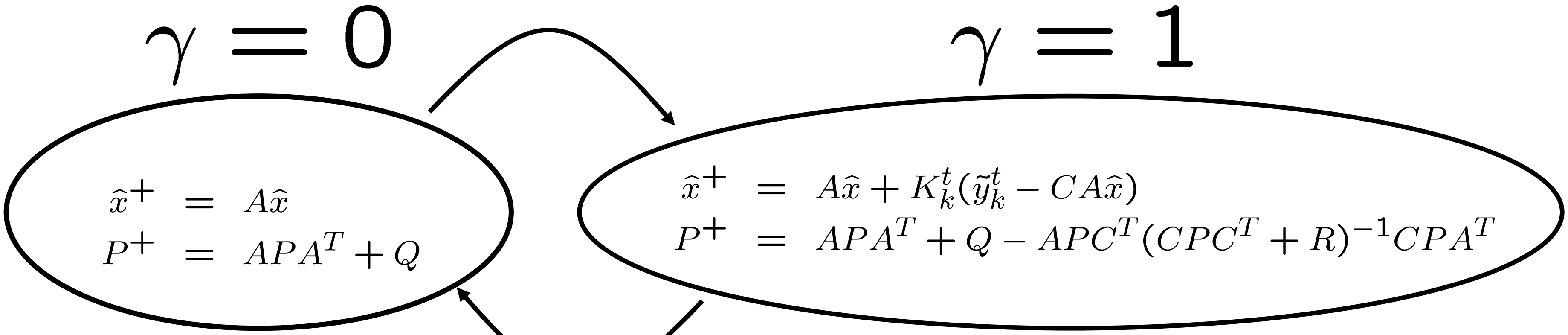
$$\begin{aligned} \hat{x}^+ &= A\hat{x} + K_k^t(\tilde{y}_k^t - CA\hat{x}) \\ P^+ &= APA^T + Q - APC^T(CPC^T + R)^{-1}CPA^T \end{aligned}$$

Lyapunov Equation
(unstable)

Riccati Equation
(stable)

Minimum variance estimation





$$
\begin{aligned}
\hat{x}^+ &= A\hat{x} \\
P^+ &= APA^T + Q
\end{aligned}
$$

$$
\begin{aligned}
\hat{x}^+ &= A\hat{x} + K_k^t(\tilde{y}_k^t - CA\hat{x}) \\
P^+ &= APA^T + Q - APC^T(CPC^T + R)^{-1}CPA^T
\end{aligned}
$$

Lyapunov Equation
(unstable)

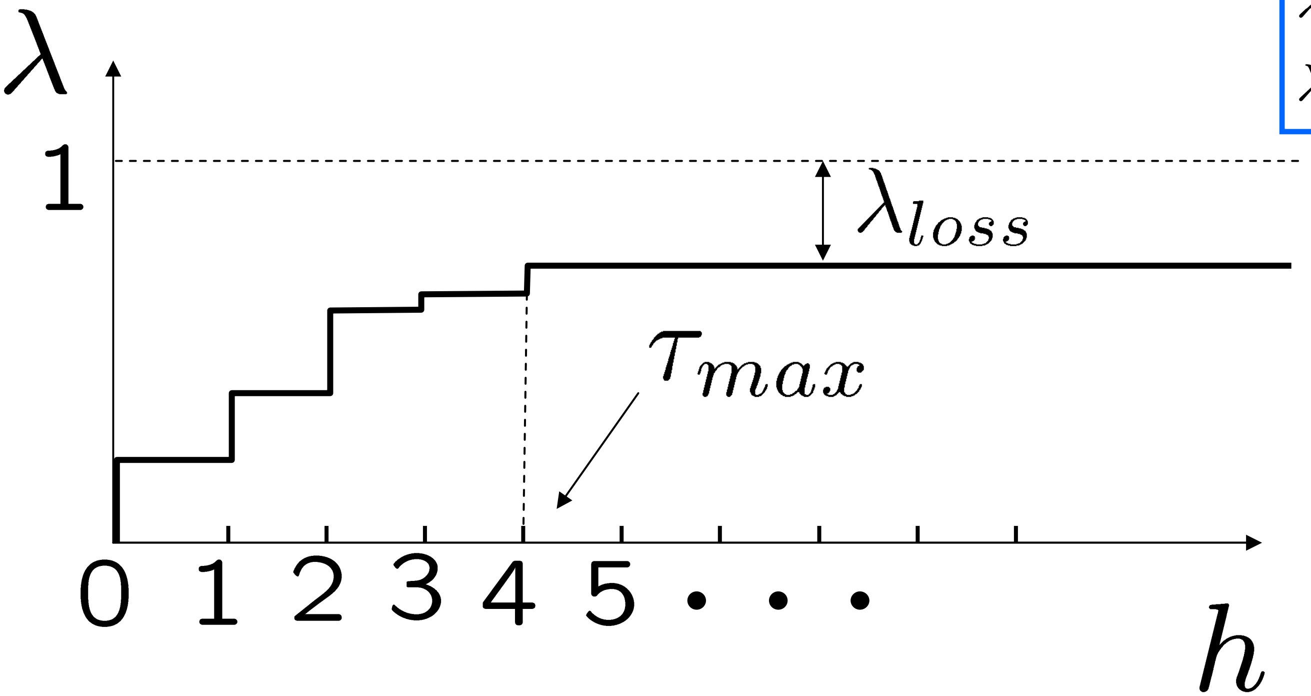
Riccati Equation
(stable)

What about stability and performance?

Additional assumption on arrival sequence necessary:
i.i.d. arrival with stationary distribution

τ_k : delay of packet y_k, $\tau_k = \infty$ if y_k never arrives

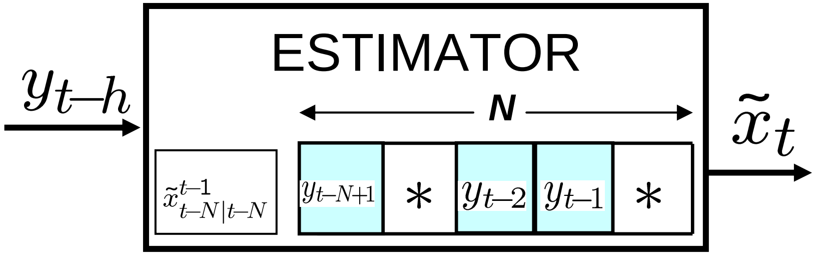
$$\lambda_h \triangleq \mathbb{P}[\tau_k \leq h],$$
$$\lambda_{loss} \triangleq \mathbb{P}[\tau_k = \infty]$$



Optimal estimation with constant gains and buffer finite memory

$\{K_h\}_{h=0}^{N-1}$, N static gains

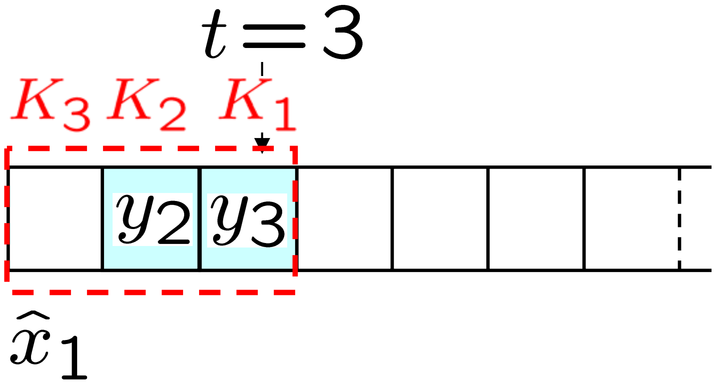
$$\tilde{x}^+ = A\tilde{x} + \gamma_{t-h}^{t} K_h(\tilde{y}_{t-h}^{t} - CA\tilde{x}), \quad h = N-1, \ldots, 0$$

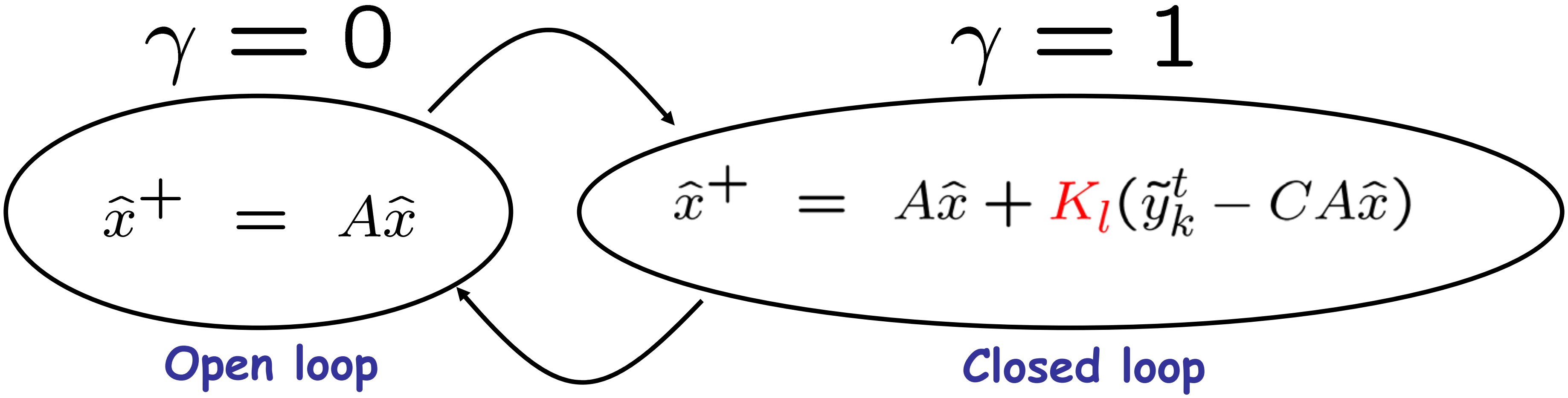


- Does not require any matrix inversion
- Simple to implement
- Upper bound for optimal estimator: $P_t \leq \tilde{P}_{t|t} \Longrightarrow \mathbb{E}_\gamma[P_{t|t}] \leq \mathbb{E}_\gamma[\tilde{P}_{t|t}] = \overline{P}_{t|t}$
- N is design parameter

GOAL: compute $\overline{P}_{t|t}$

Suboptimal minimum variance estimation

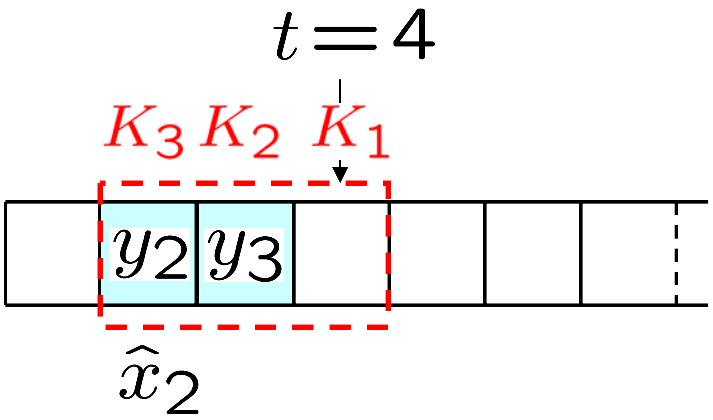


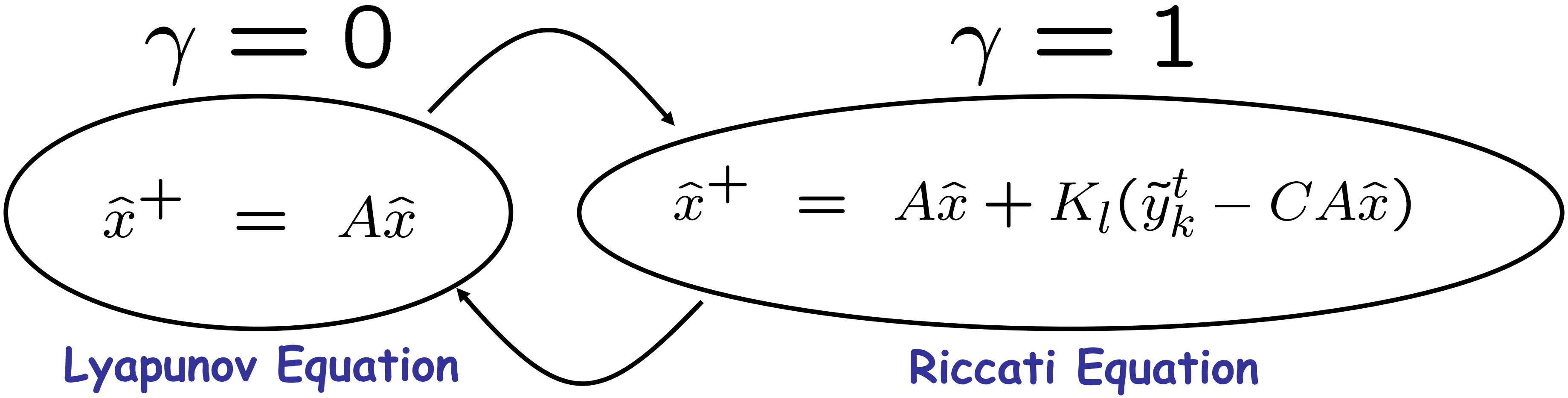


$$\hat{x}^+ = A\hat{x}$$

$$\hat{x}^+ = A\hat{x} + K_l(\tilde{y}_k^t - CA\hat{x})$$

Suboptimal minimum variance estimation





$\gamma = 0$

$\gamma = 1$

$$\hat{x}^+ = A\hat{x}$$

$$\hat{x}^+ = A\hat{x} + K_l(\tilde{y}_k^t - CA\hat{x})$$

Lyapunov Equation
(unstable)

Riccati Equation
(stable)

Steady state estimation error

Fixed gains:

$$\mathcal{L}_\lambda(K,P) = \lambda A(I-KC)P(I-KC)^T A^T + (1-\lambda)APA^T + Q + \lambda AKRK^T A^T$$

$$\boxed{\begin{array}{l} \begin{aligned} \overline{P} &= \mathcal{L}_{\lambda_{N-1}}(K_{N-1},\overline{P}) \\ \overline{P}^+ &= \mathcal{L}_{\lambda_k}(K_k,\overline{P}), \qquad k=N-2,\ldots,0 \end{aligned} \\ \lim_{t\to\infty} \overline{P}_{t|t} = \overline{P} \end{array}}$$

Optimal fixed gains:

$$\Phi_\lambda(P) = APA^T + Q - \lambda\, APC^T(CPC^T+R)^{-1}CPA^T$$

Modified Algebraic Riccati Equation (MARE)
($_1$(P)=ARE)

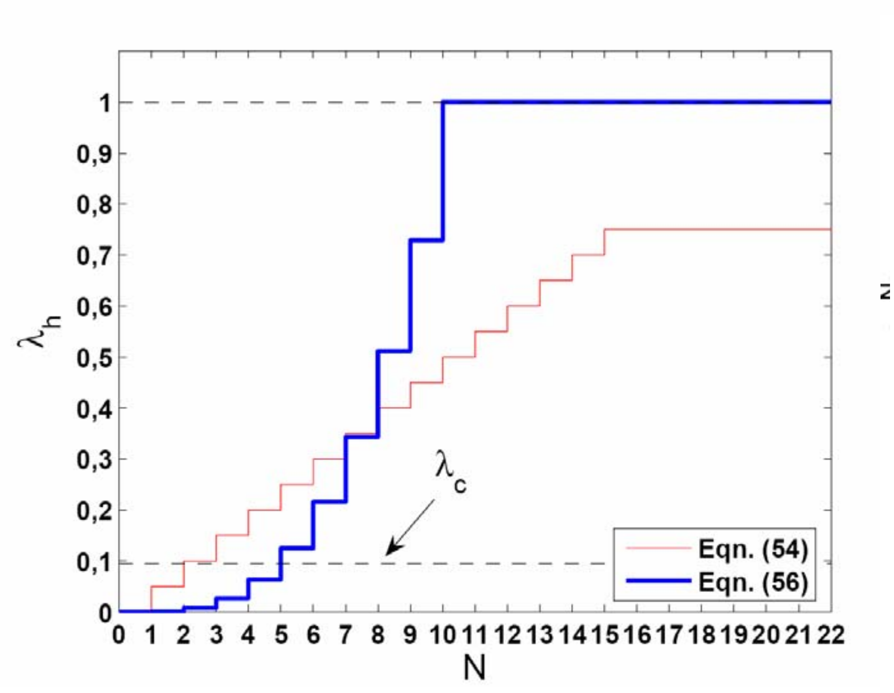
$$\min_{K_0,\ldots,K_{N-1}} \overline{P} \quad \Longrightarrow \quad \begin{aligned} \overline{P}_{N-1} &= \Phi_{\lambda_{N-1}}(\overline{P}_{N-1}) \\ \overline{P}_k &= \Phi_{\lambda_k}(\overline{P}_{k+1}), \qquad k=N-2,\ldots,0 \\ K_k &= \overline{P}_k C^T(C\overline{P}_k C^T + R)^{-1} \end{aligned}$$

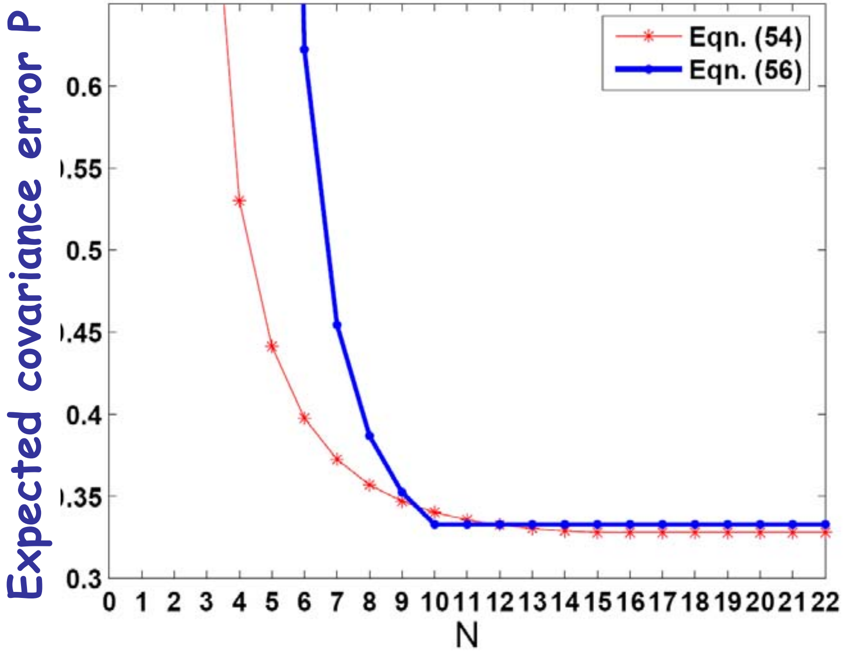
(off-line computation)

Numerical example (I)

Discrete time linearized inverted pendulum:

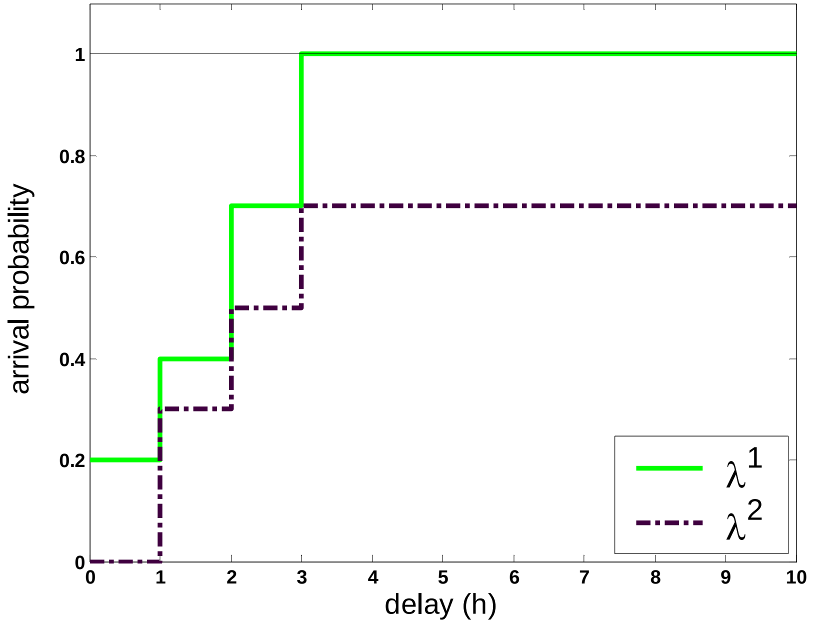
$$A = \begin{bmatrix} 1.01 & 0.05 \\ 0.05 & 1.01 \end{bmatrix}, \;\; C = [\,1 \quad 0\,], \;\; R = 1, \;\; Q = \begin{bmatrix} 0.01 & 0.01 \\ 0.01 & 1 \end{bmatrix}$$

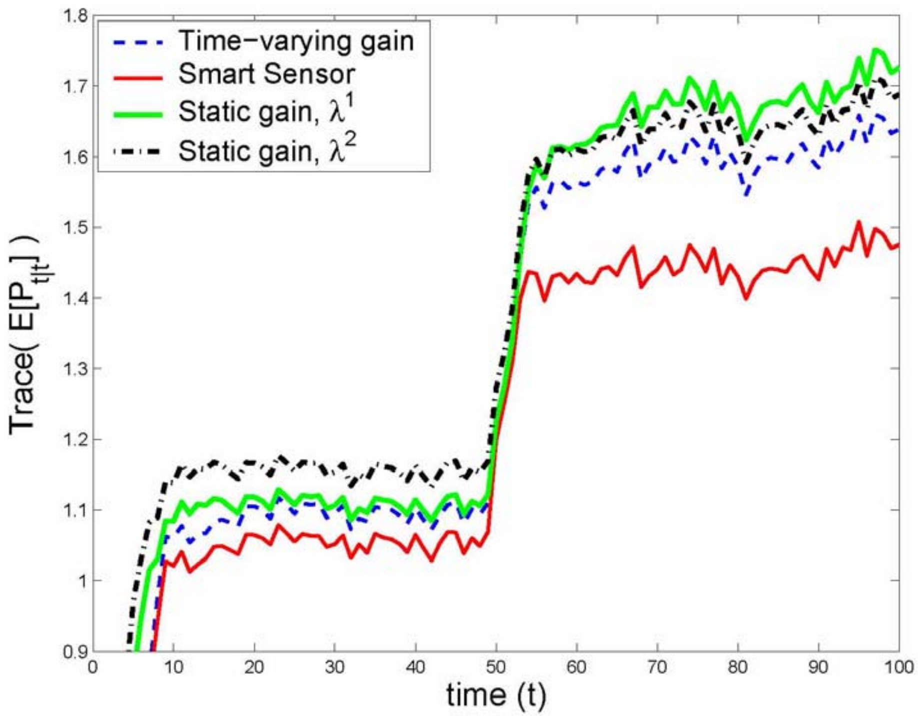




Numerical example (II)

Time-varying arrival probability distribution

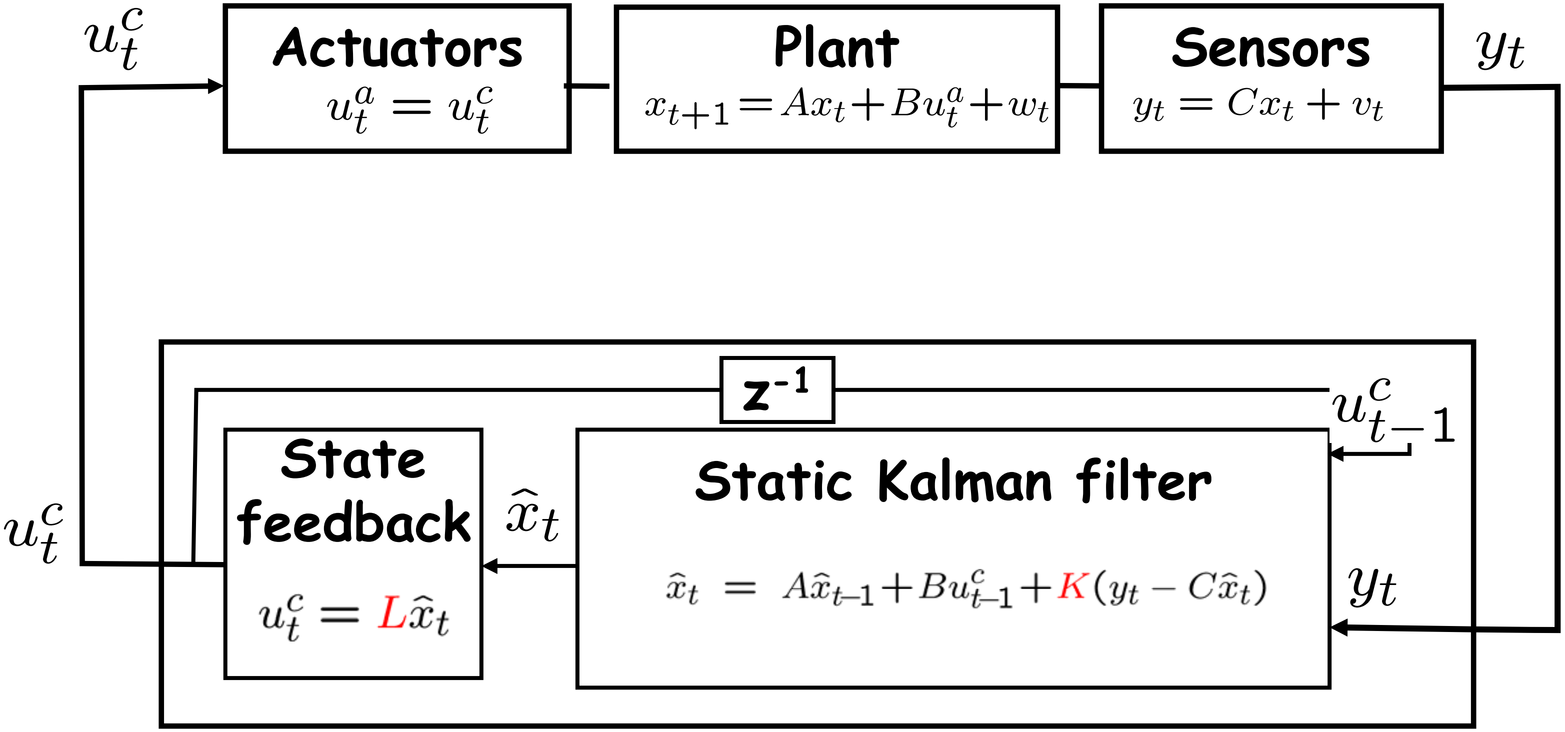




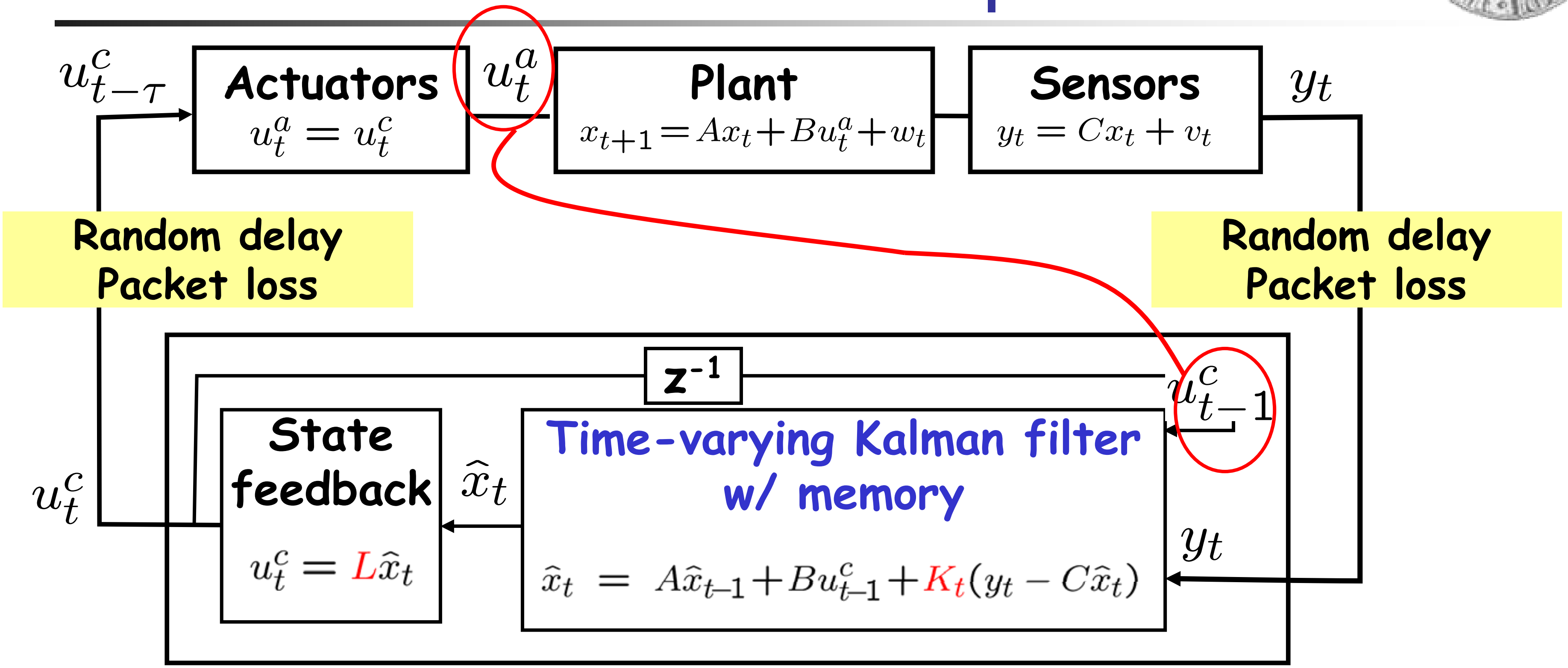
$$\lambda^1 \qquad 0 \leq t \leq 50$$

$$\lambda^2 \qquad t > 50$$

Back to the control problem



Back to the control problem



$u^c_{t-\tau}$

Actuators
$u^a_t = u^c_t$

u^a_t

Plant
$x_{t+1} = Ax_t + Bu^a_t + w_t$

Sensors
$y_t = Cx_t + v_t$

y_t

Random delay
Packet loss

Random delay
Packet loss

z^{-1}

u^c_{t-1}

u^c_t

State feedback
$u^c_t = L\hat{x}_t$

$\hat{x}_t$

Time-varying Kalman filter w/ memory
$\hat{x}_t = A\hat{x}_{t-1} + Bu^c_{t-1} + K_t(y_t - C\hat{x}_t)$

y_t

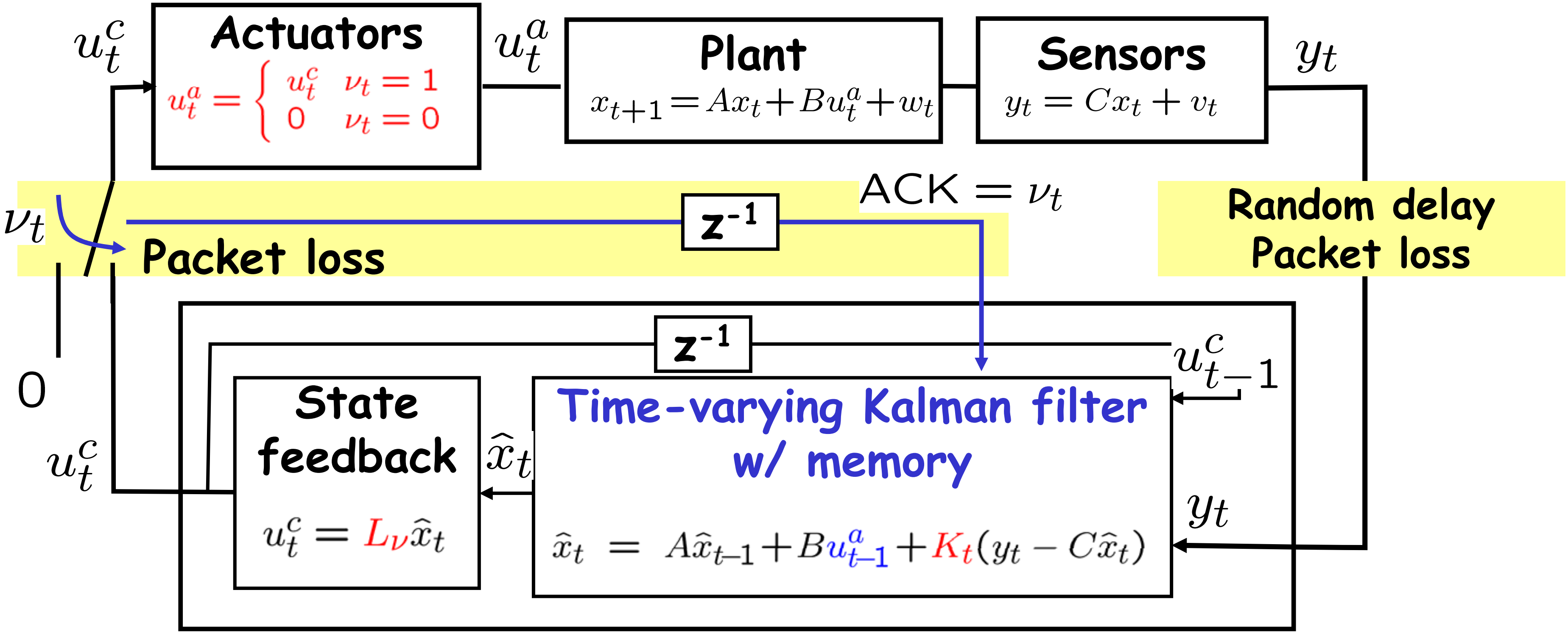
$$\hat{x}_t = E[x_t | y_t, y_{t-1}, .., y_0, u^a_{t-1}, .., u^a_1]$$

if $u^c_{t-1} \neq u^a_{t-1} \implies e_t = x_t - \hat{x}_t = f(y_t, .., y_0, u^c_t, .., u^c_0, u^a_t, .., u^a_0)$

$$P_{t|t-1} = AP_{t-1|t-1}A^T + Q + B(u^a_{t-1} - u^c_{t-1})(u^a_{t-1} - u^c_{t-1})^T B^T$$

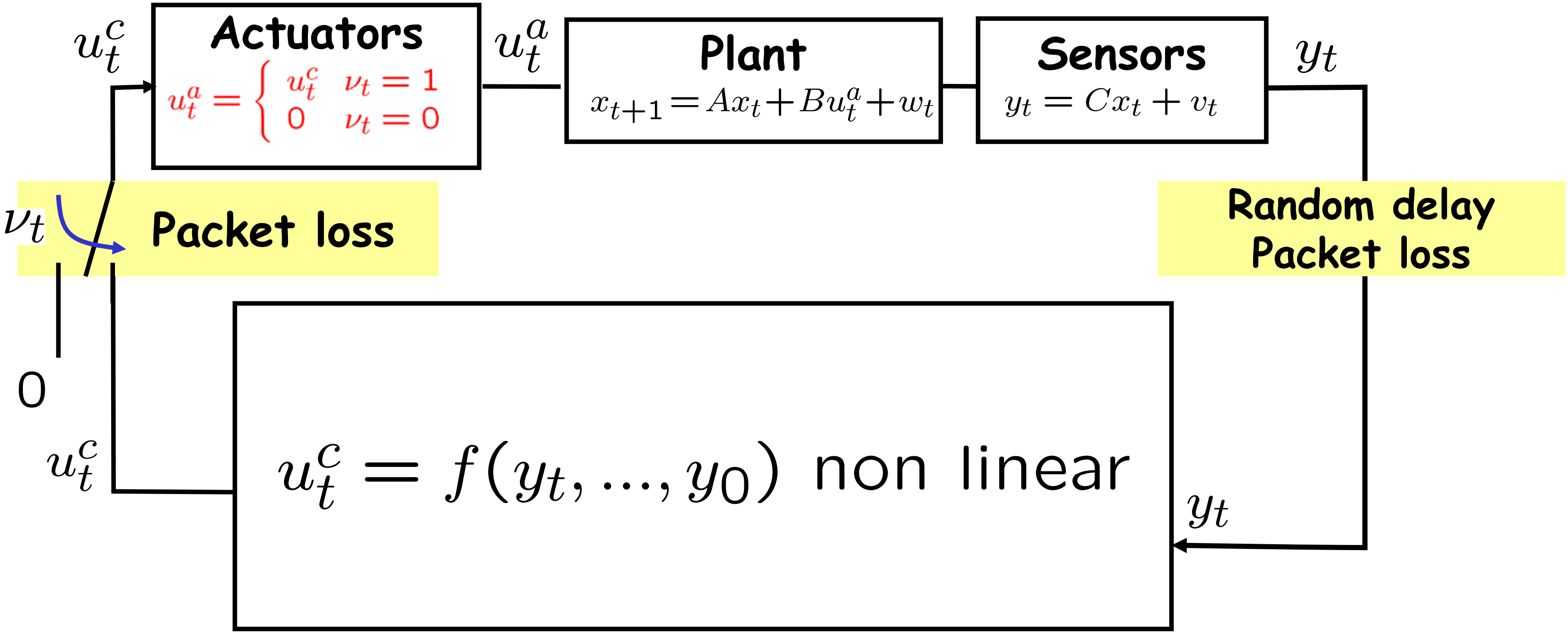
Estimation error coupled with control action → no separation principle

LQG over TCP-like (ACK-based) protocols


u_t^c
Actuators
$u_t^a = \begin{cases} u_t^c & \nu_t = 1 \\ 0 & \nu_t = 0 \end{cases}$
u_t^a
Plant
$x_{t+1} = Ax_t + Bu_t^a + w_t$
Sensors
$y_t = Cx_t + v_t$
y_t
ν_t
Packet loss
z-1
ACK $= \nu_t$
Random delay
Packet loss
0
z-1
u_{t-1}^c
u_t^c
State feedback
$u_t^c = L_\nu \hat{x}_t$
$\hat{x}_t$
Time-varying Kalman filter w/ memory
y_t
$\hat{x}_t = A\hat{x}_{t-1} + Bu_{t-1}^a + K_t(y_t - C\hat{x}_t)$

- Separation principle hold (I know exactly u^a_{t-1})
- $_t$ Bernoulli rand. var and independent of observation arrival process
- Static state feedback, L solution of dual MARE

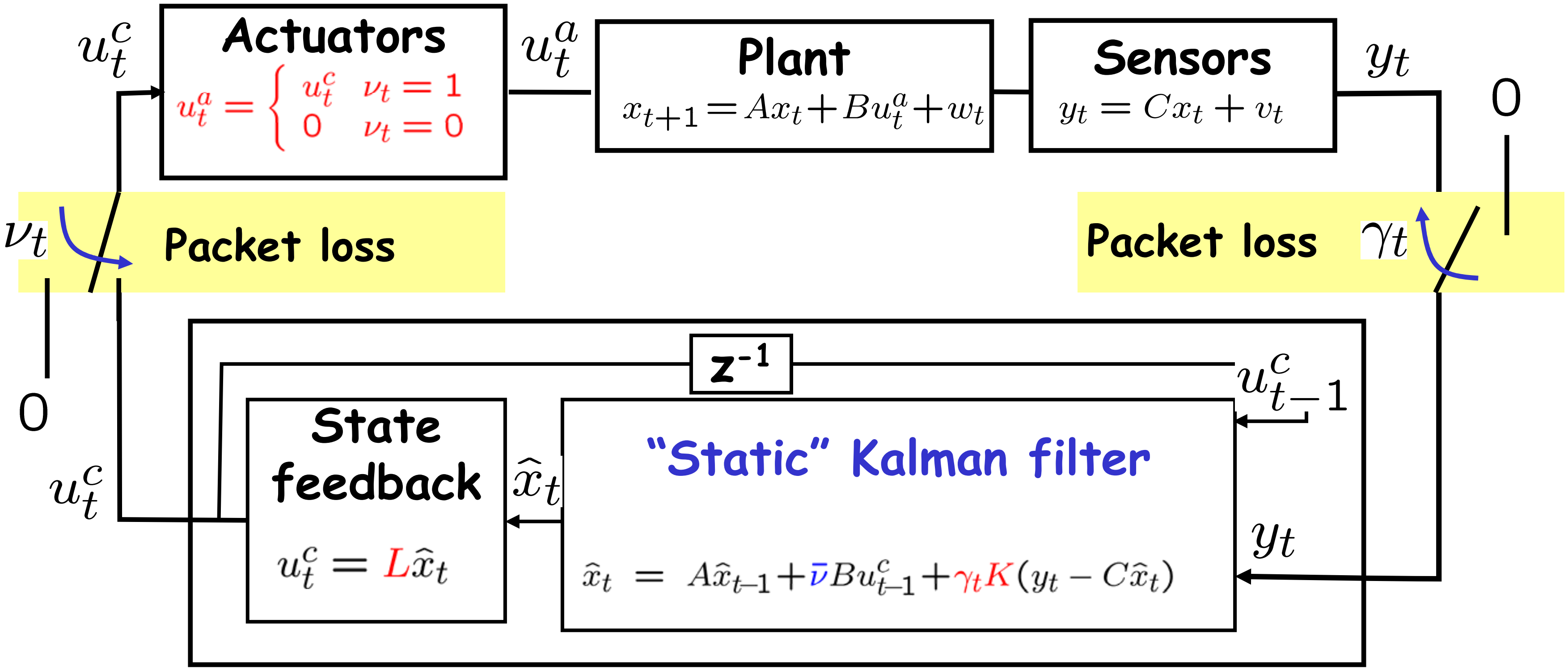
LQG over UDP-like (no-ACK) protocols



u_t^c
Actuators
$u_t^a = \begin{cases} u_t^c & \nu_t = 1 \\ 0 & \nu_t = 0 \end{cases}$
u_t^a
Plant
$x_{t+1} = Ax_t + Bu_t^a + w_t$
Sensors
$y_t = Cx_t + v_t$
y_t
ν_t
Packet loss
Random delay
Packet loss
0
u_t^c
$u_t^c = f(y_t, \ldots, y_0)$ non linear
y_t

- LQG problem still well defined: $\min_{u_t^c, \ldots, u_1^c} E[\sum_{h=1}^t x_t^T W x_t + (u_t^a)^T U u_t^a]$
- No separation principle hold (u^a_{t-1} NOT known exactly)
- ... but still have some statistical information about u^a_{t-1}

LQG over UDP-like (no-ACK) protocols



- Bernoulli arrival process $P[\nu_t = 1] = \bar{\nu}, P[\gamma_t = 1] = \bar{\gamma}$
- $\bar{\nu}u_{t-1}^c = E[u_{t-1}^a]$
- Sub-optimal controller forced to be state estimator+state feedback
- Optimal choice of K,L is unique solution of 4 coupled Riccati-like equations

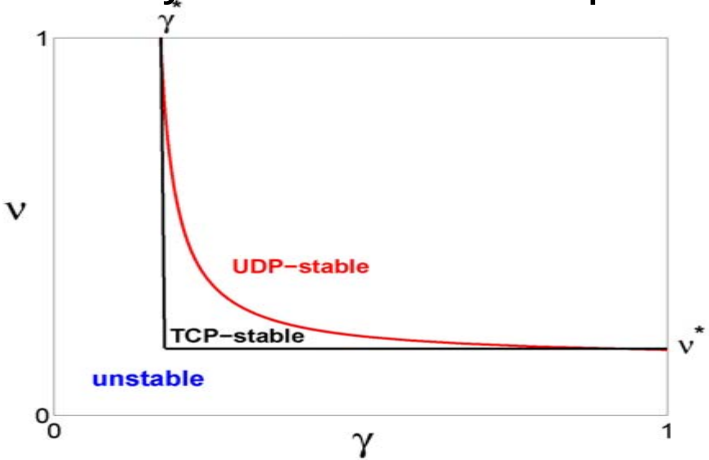
"Compensability and Optimal Compensation of systems with white parameters", De Koning, TAC'92

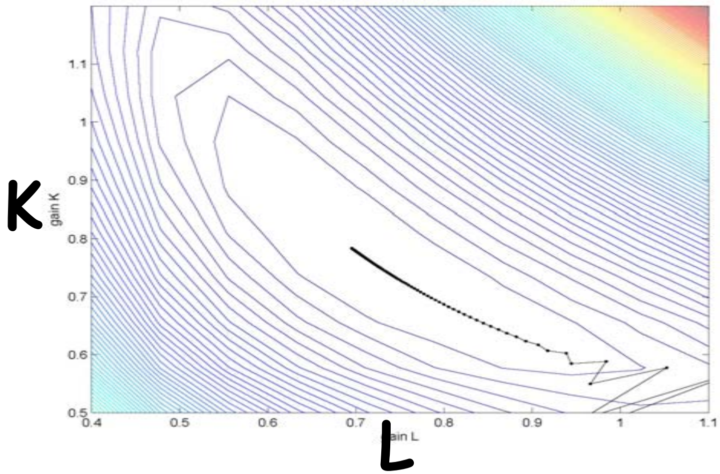
LQG as optimization problem

$$P \triangleq \mathbb{E}\left[\begin{bmatrix} x \\ \hat{x} \end{bmatrix}\begin{bmatrix} x^T & \hat{x}^T \end{bmatrix}\right] = \begin{bmatrix} P_{11} & P_{12} \\ {P_{12}}^T & P_{22} \end{bmatrix}$$

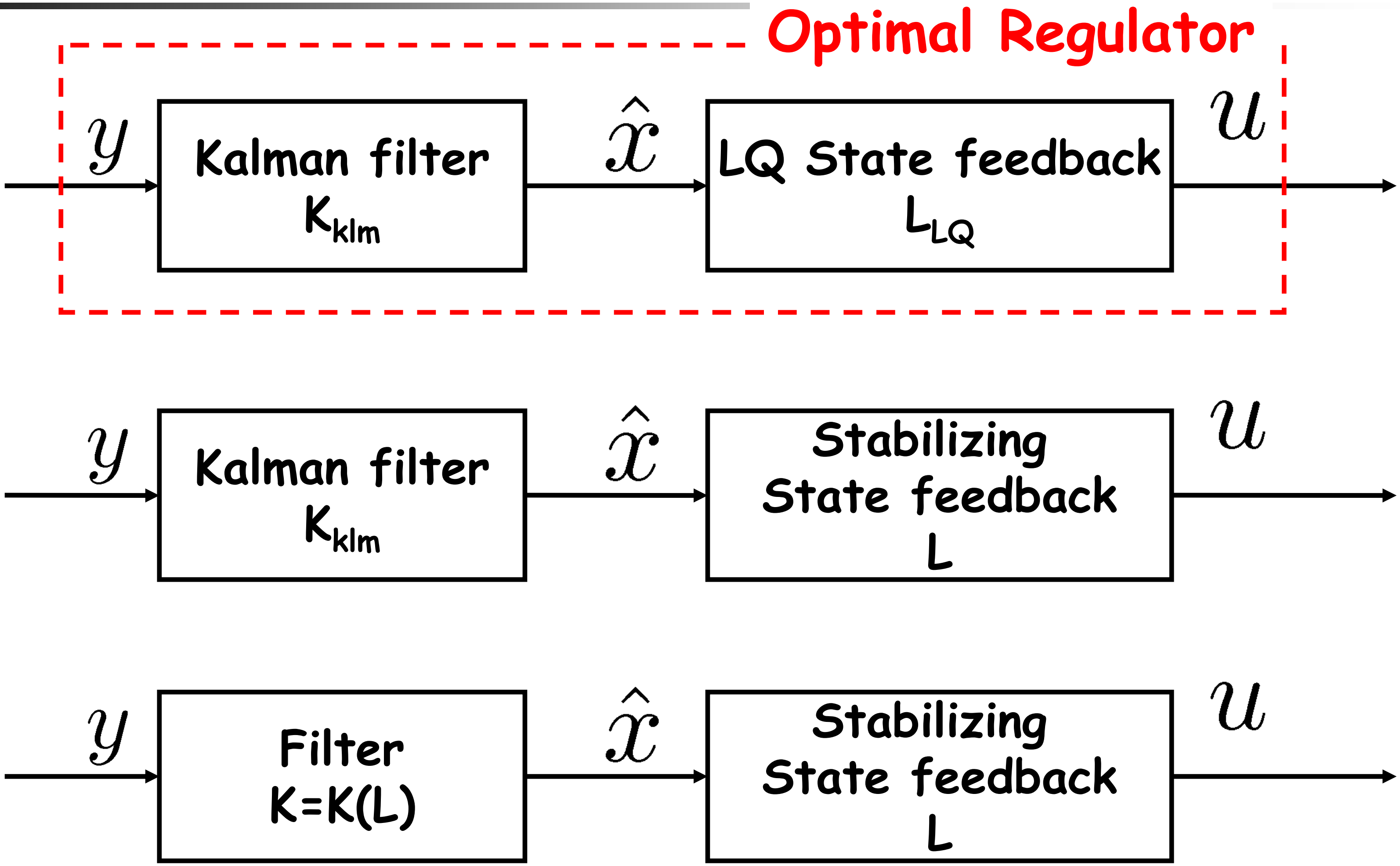
$$\begin{aligned} \text{Min}_{K,L} \quad & \text{Trace}\left(\begin{bmatrix} W & 0 \\ 0 & \bar{\nu} L^T U L \end{bmatrix} P\right) \\ s.t. \quad & P = \mathbb{E}\left[\begin{bmatrix} A & -\nu_k BL \\ \gamma_k KC & A - \bar{\nu} BL - \gamma_k KC \end{bmatrix} P \begin{bmatrix} A & -\nu_k BL \\ \gamma_k KC & A - \bar{\nu} BL - \gamma_k KC \end{bmatrix}^T\right] + \begin{bmatrix} Q & 0 \\ 0 & \bar{\gamma} K R K^T \end{bmatrix} \\ & P \geq 0 \end{aligned}$$

- Non convex problem even for = =1, i.e. classic LQG
- Classic and TCP-based LQG become convex when exploiting optimality conditions like uncorralation between estimate and error estimate $\mathbb{E}[x(x-\hat{x})^T] = 0$
- For UDP-like problem non convex but unique solution using Homotopy and Degree Theory (DeKoning,Athans,Bernstain) (maybe using Sum-of-Squares?)
- Stability on and is coupled



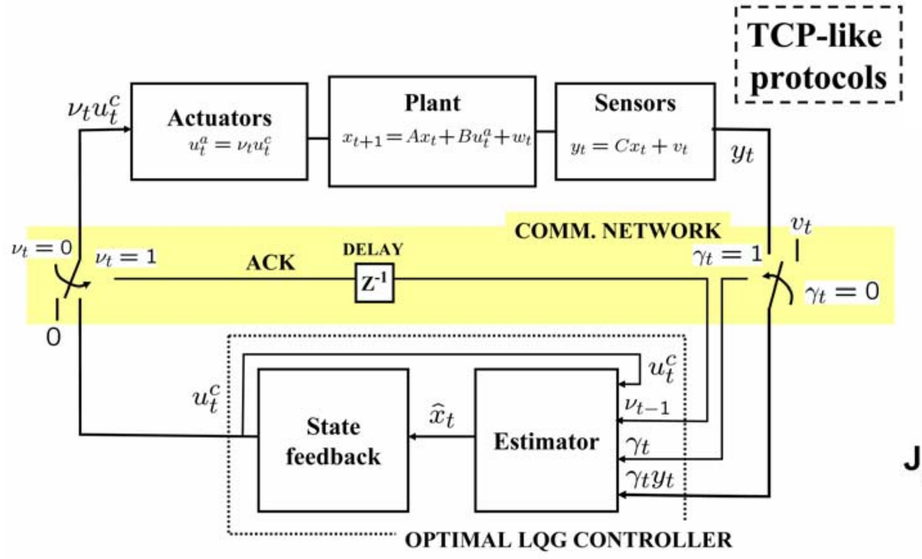


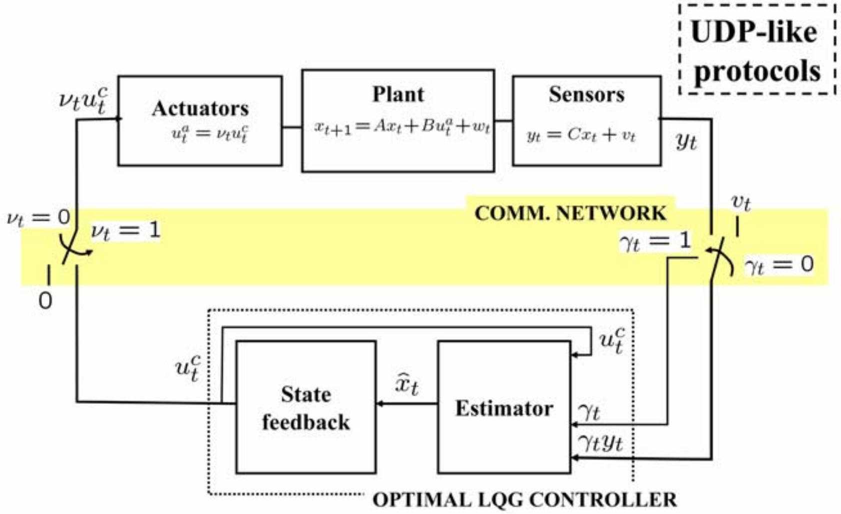
Side note: Kalman filter is not always optimal !

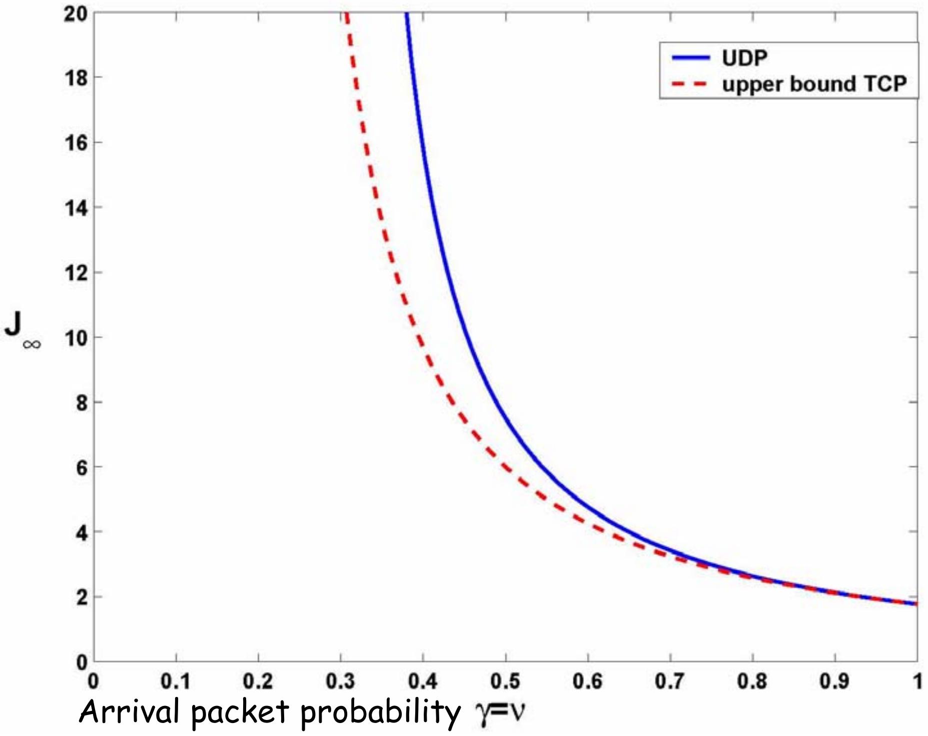


- Kalman filter always gives smallest estimate error **regardless** of controller L
- If controller L L_{LQ} , then performance improves if my estimate is “bad” !

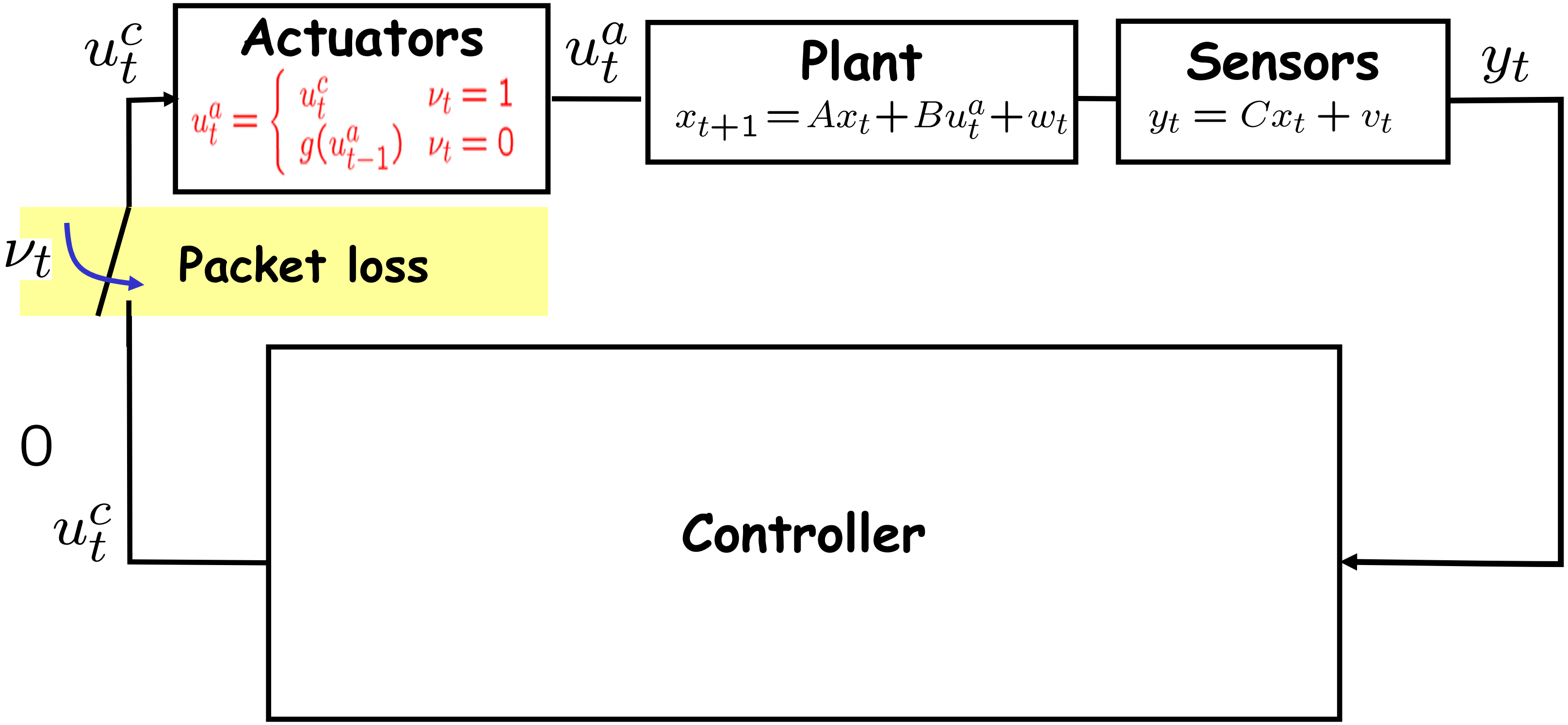
Numerical example: TCP vs UDP







To hold or to zero control input?


u_t^c
Actuators
$u_t^a = \begin{cases} u_t^c & \nu_t = 1 \\ g(u_{t-1}^a) & \nu_t = 0 \end{cases}$
u_t^a
Plant
$x_{t+1} = Ax_t + Bu_t^a + w_t$
Sensors
$y_t = Cx_t + v_t$
y_t
ν_t
Packet loss
0
u_t^c
Controller

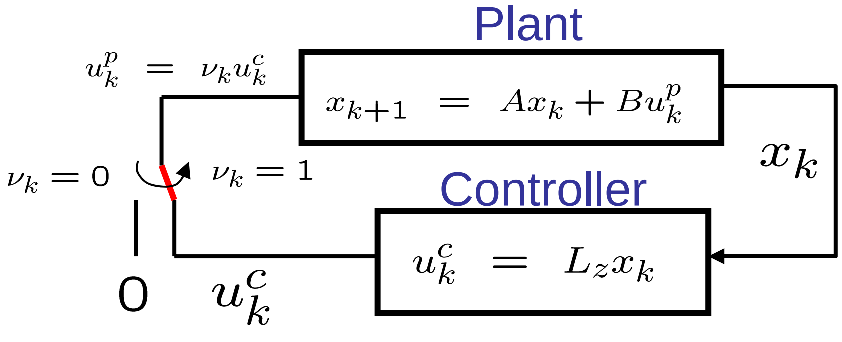
Most common strategy:

$g(u_{t-1}^a) = 0$ zero-input strategy **(mathematically appealing)**

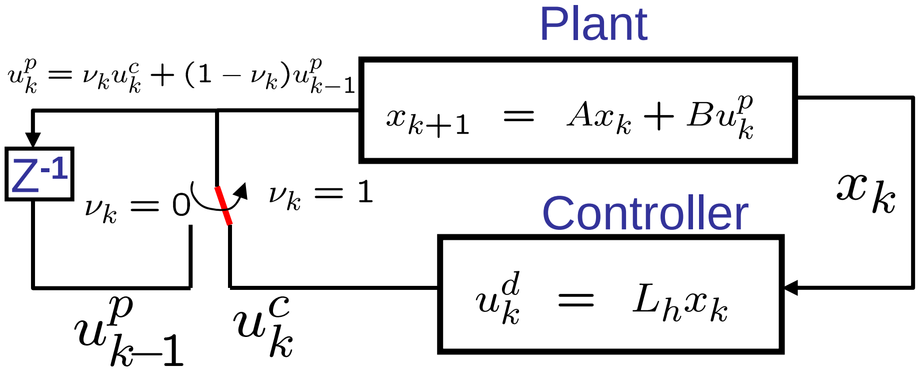
$g(u_{t-1}^a) = u_{t-1}^a$ hold-input strategy **(most natural)**

To hold or to zero control input: no noise (jump linear systems)

Zero-input Strategy



Hold-input Strategy



$$J_z^* = \min_{L_z} E[\sum_{t=1}^{\infty} x_t^T W x_t + (u_t^a)^T U u_t^a]$$

$$J_h^* = \min_{L_h} E[\sum_{t=1}^{\infty} x_t^T W x_t + (u_t^a)^T U u_t^a]$$

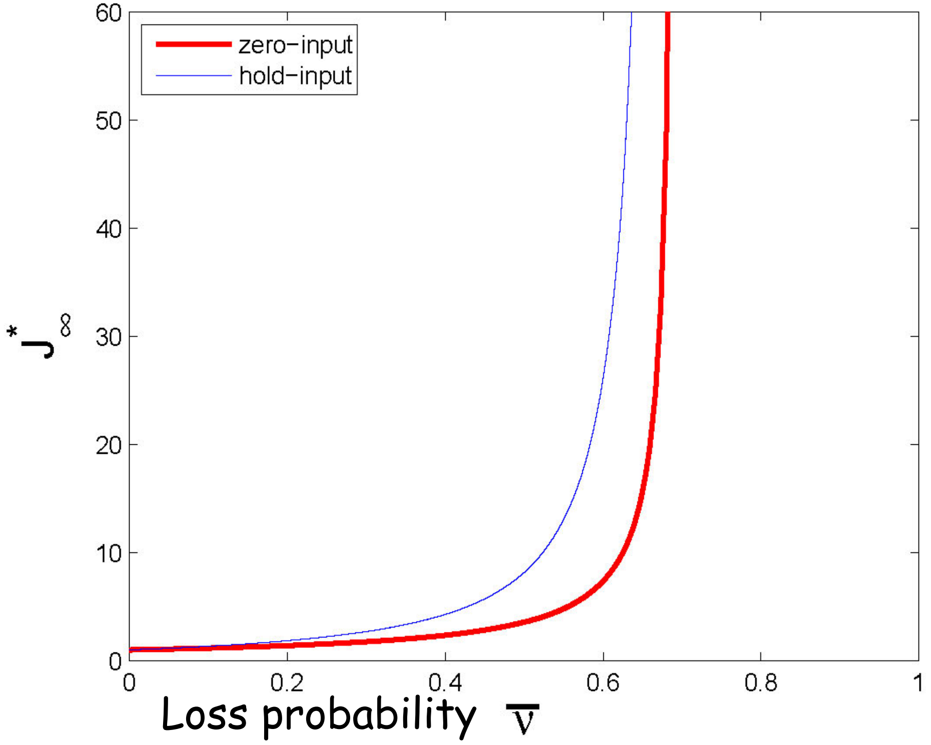
Using cost-to-go function (dynamic programming)

$$J_z^* = E[x_0^T S_z x_0] \qquad J_h^* = E[x_0^T S_h x_0]$$

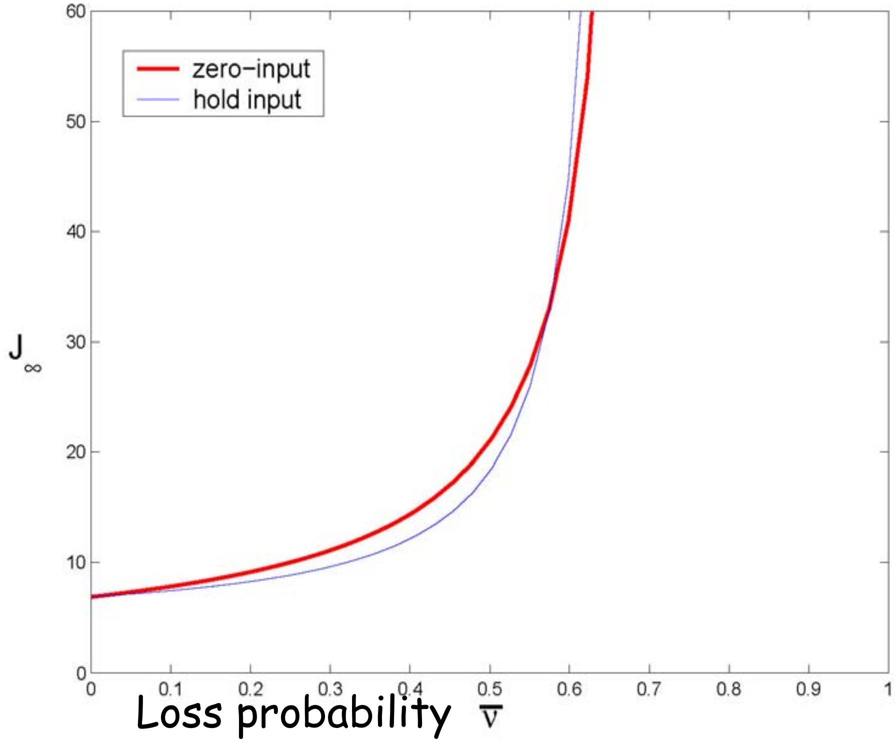
$$S_z = \Phi_z(S_z) \longleftarrow \text{Riccati-like equation} \longrightarrow S_h = \Phi_h(S_h)$$

$$L_z^* = f_z(S_z) \qquad L_h^* = f_h(S_h)$$

Example: unstable scalar system

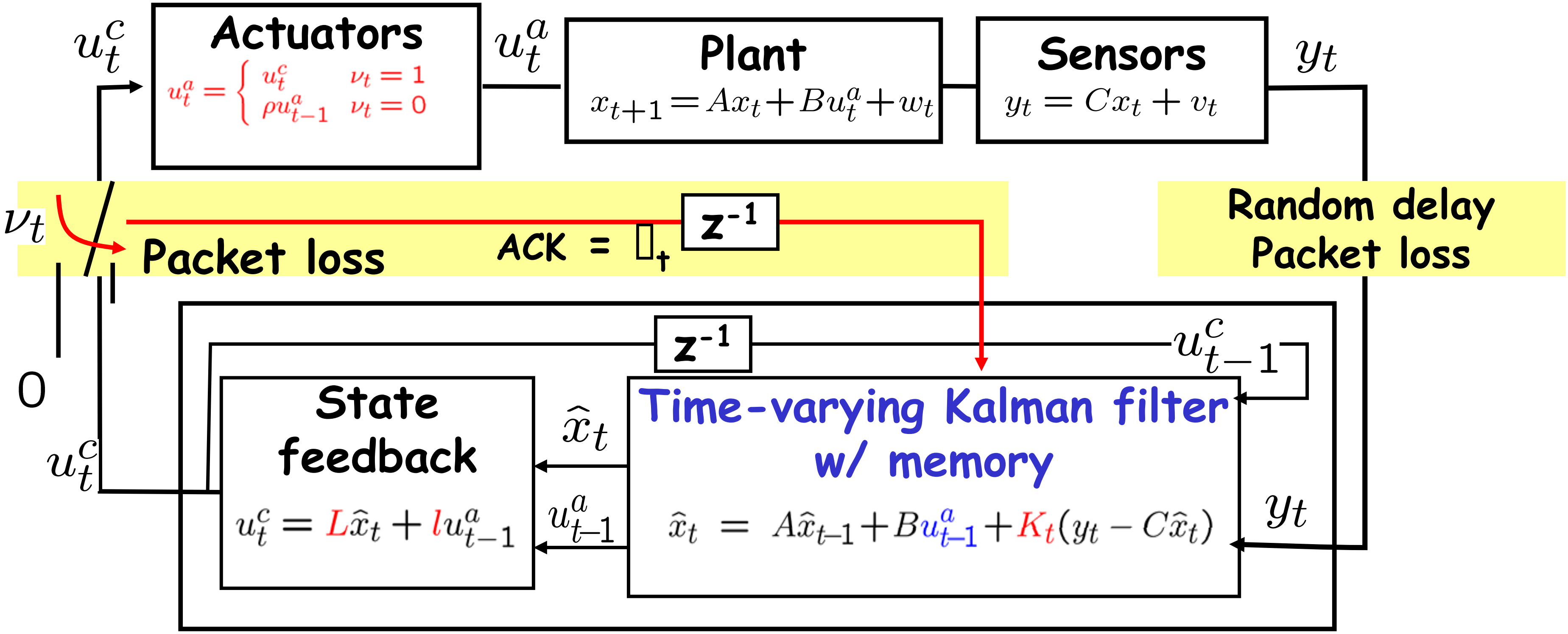


A=1.2, U=0 (fastest convergence)



A=1.2, U=10 (small input)

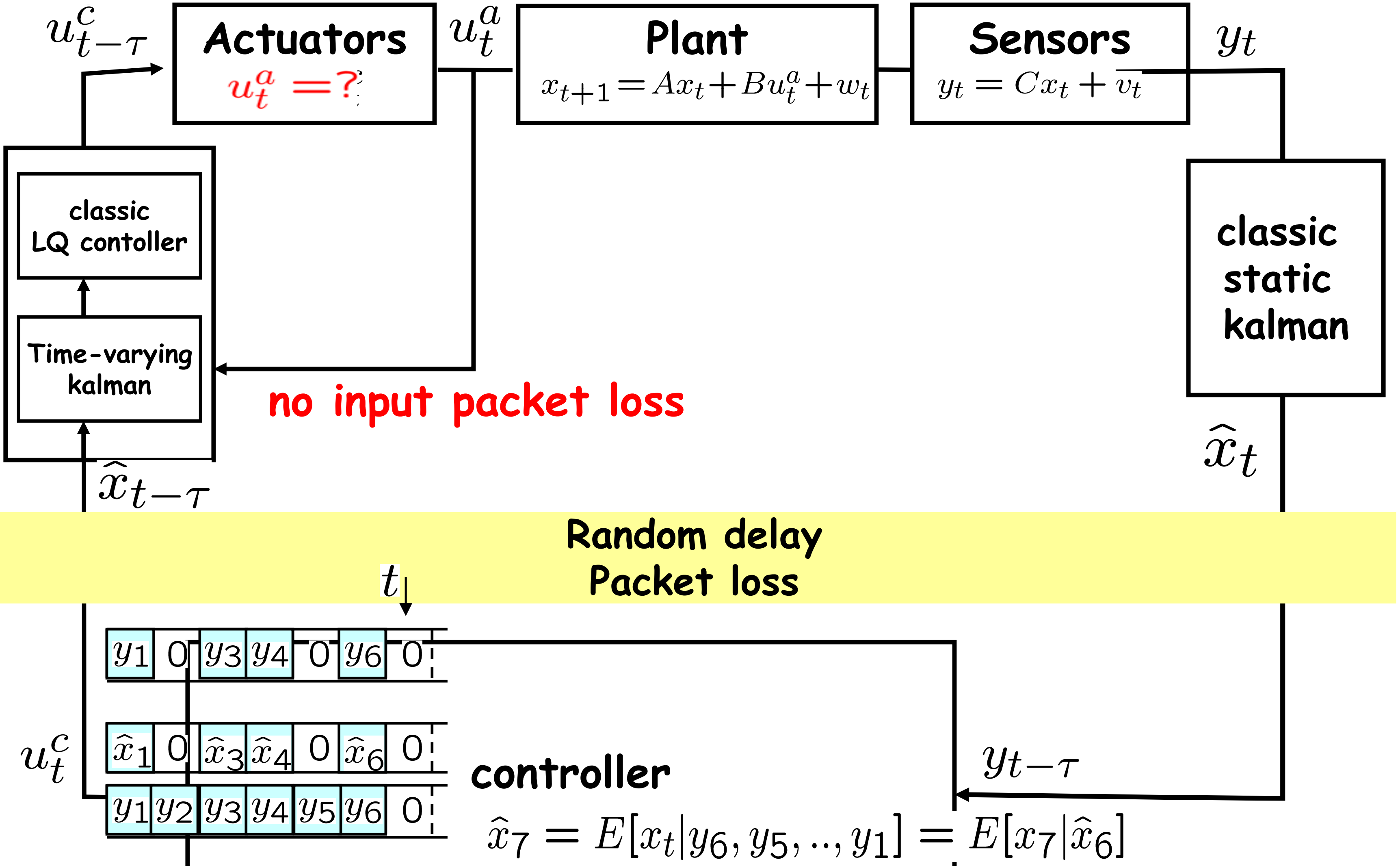
LQG over TCP-like protocols revised


u_t^c
Actuators
$u_t^a = \begin{cases} u_t^c & \nu_t = 1 \\ \rho u_{t-1}^a & \nu_t = 0 \end{cases}$
u_t^a
Plant
$x_{t+1} = Ax_t + Bu_t^a + w_t$
Sensors
$y_t = Cx_t + v_t$
y_t
ν_t
Packet loss
ACK = $\square_t$
z-1
Random delay
Packet loss
0
z-1
u_{t-1}^c
u_t^c
State feedback
$u_t^c = L\hat{x}_t + l u_{t-1}^a$
$\hat{x}_t$
u_{t-1}^a
Time-varying Kalman filter w/ memory
$\hat{x}_t = A\hat{x}_{t-1} + Bu_{t-1}^a + K_t(y_t - C\hat{x}_t)$
y_t

Conjecture:

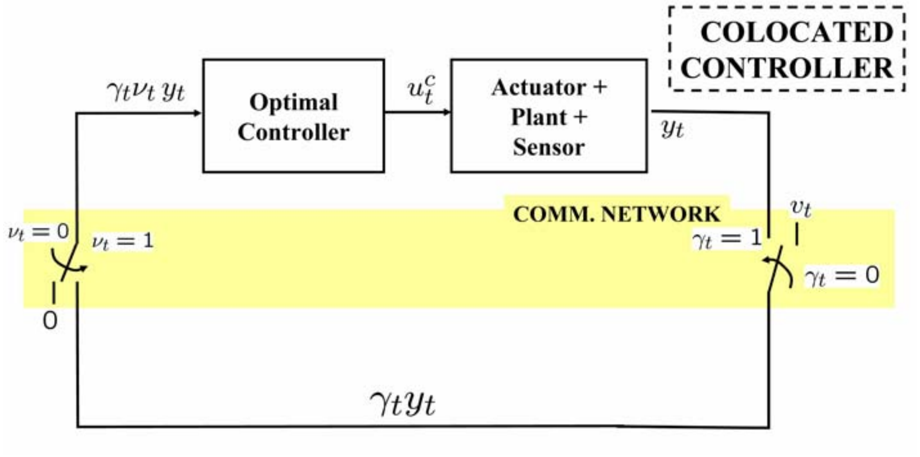
- Separation principle hold
- Optimal function $g(u_{t-1}^a) = \rho u_{t-1}$
- Design parameter L, l, ρ obtained via LQ-like optimal state feedback

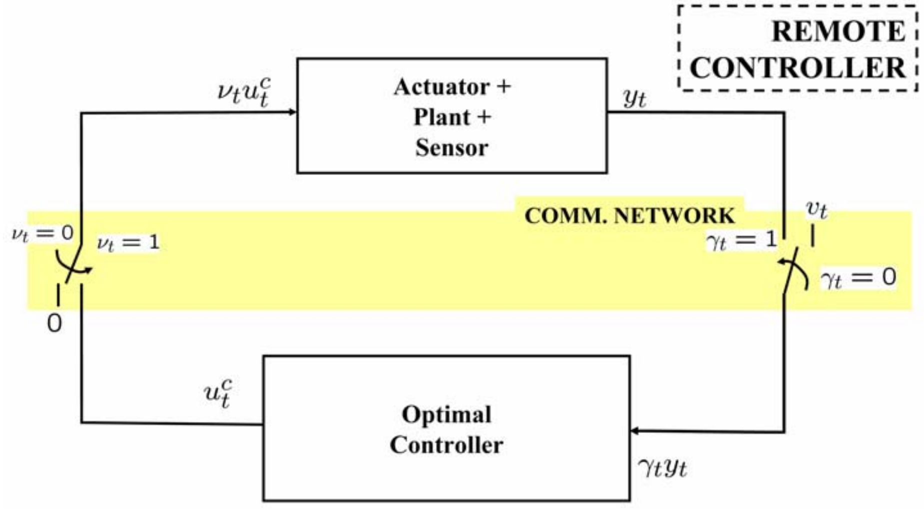
Smart sensors & smart actuators

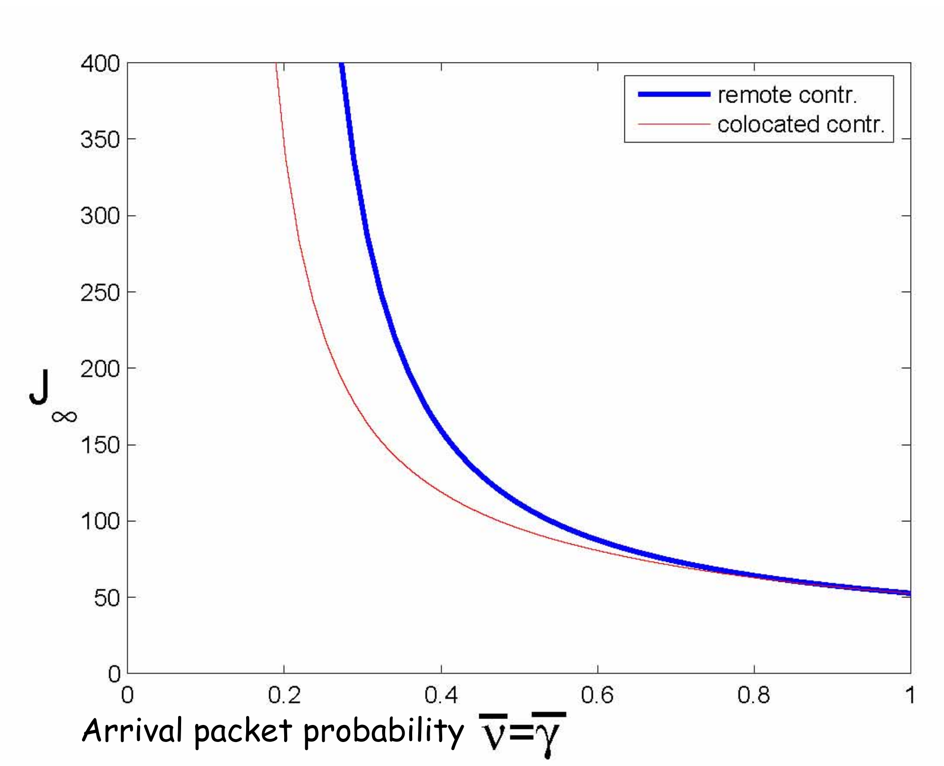


"Optimal LQG control across a packet-dropping link", Gupta, Spanos, Murray, Submitted to Sys.Cont.Lett. 05
"Estimation under controlled and uncontrolled communications in networked control systems", Xu, Hespanha, CDC 05

Numerical example: remote vs co-located controller



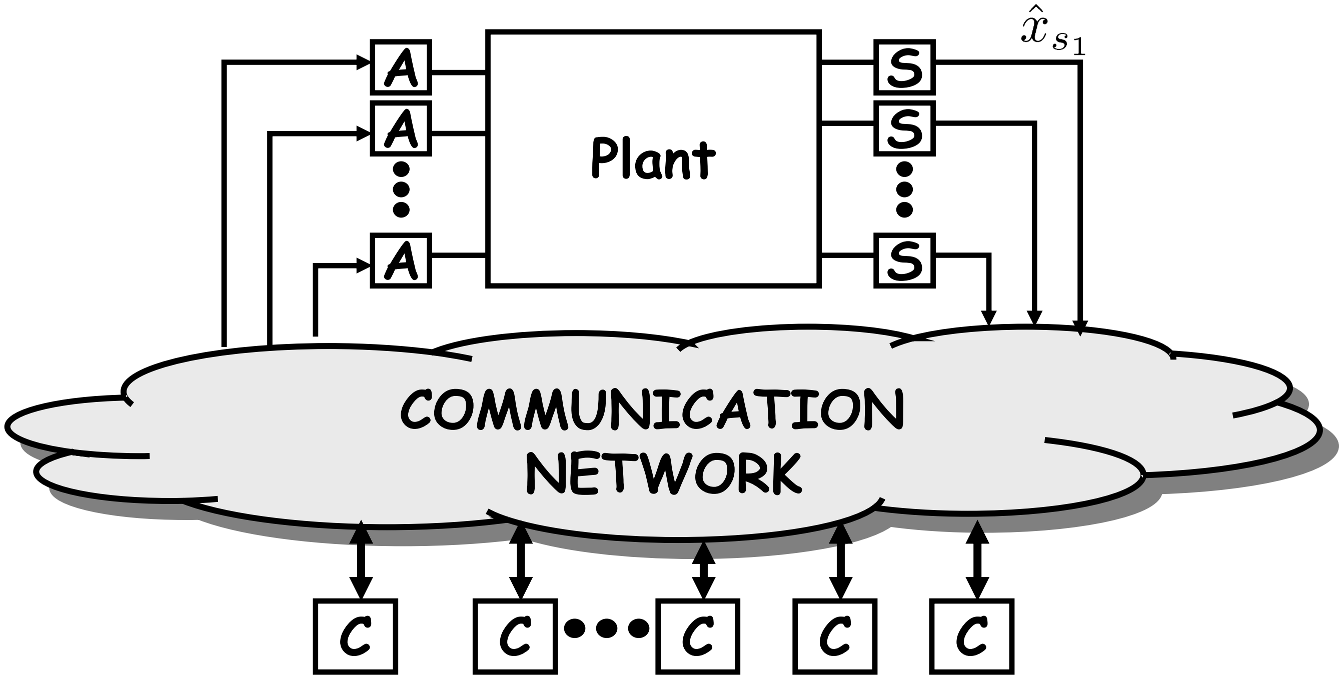




Takeaway points

- Input packet loss more dangerous than measurement packet loss
- TCP-like protocols help controller design as compared to UDP-like (but harder for communication designer)
- If you can, place controller near actuator
- If you can, send estimate rather than raw measurement
- Zero-input control seems to give smaller closed loop state error ($||x_t||$) than hold-input (but higher input)
- Trade-off in terms of performance, buffer length, computational resources (matrix inversion) when random delay
- Can help comparing different communication protocols from a real-time application performance

Future work



- Multiple sensors:
  - data fusion, i.e. $y_1,..,y_m$ arrive at different times
  - distributed estimation & consensus $E[x|y_1,...,y_N] \overset{?}{=} E[x|\hat{x}_{s_1},\hat{x}_{s_N}]$
- Multiple actuators
  - trade-off between distributed control & centralized coordination