Some results on optimal estimation and control for lossy NCS

Luca Schenato

Joint work w/ Bruno Sinopoli (CMU), Massimo Franceschetti (UCSD), Kameshwar Poolla and Shankar Sastry (UCB)

KTH, 14/09/2007
NCSs: physically distributed dynamical systems interconnected by a communication network
NCSs: what’s new for control?

Classical architecture: Centralized structure

- Actuators
- Plant
- Sensors
- Controller
NCSs: what’s new for control?

NCSs: Large scale distributed structure

Connectivity
Limited capacity

Interference
COMMUNICATION NETWORK
Congestion

Packet loss
Random delay
Quantization

A
A
A

S
S
S

C
C
C

C
C
C

C
C
C

C
C
C

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

COMPUTER SCIENCE
Interdisciplinary research needed

COMMUNICATIONS ENGINEERING
- Communication 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
Wireless Sensor Actuator Networks (WSANs)

- Small devices
  - µController, Memory
  - Wireless radio
  - Sensors & Actuators
  - Batteries
- Inexpensive
- Multi-hop communication
- Programmable (micro-PC)
NCS example: Smart Greenhouse

- Distributed estimation
- Distributed control
- Control under packet loss & random delay
- Sensor fusion
- Distributed time synchronization
Building thermodynamics model identification
- Sensor selection for identification
- Optimal sensor placement
NCS example:
Distributed Localization & Tracking

- Indoor radio signal modeling
- Real-time localization
- Distributed tracking
- Coordination
NCS example: Coordinated Robotics & WSNs

- Coordination & consensus algorithms
- Integration mobile nodes w/ static nodes
- WSN-based localization & navigation
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

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

Sensors and actuators are co-located, i.e. no delay nor loss
1. **Separation principle holds**: Optimal controller = Optimal estimator design + Optimal state feedback design
2. **Closed Loop system always stable** (under standard reach./det. hypotheses)
3. **Gains** $K,L$ are constant solution of Algebraic Riccati Equations
Optimal LQG control over DCN

Random delay or drop

Controller?

DIGITAL COMMUNICATION NETWORK

Controller

Random delay or drop

Controller?

ACK?

Digital Communication Network

$u_t^c = \begin{cases} u_{t-\tau}^c \\ 0 \end{cases}$

$u_t^a$

Plant

$y_t = Cx_t + v_t$

Controller

Controller?

ACK?

$y_{t-\tau_s}$
Some consideration on the separation principle

\[ u_t^c = u_t \]

**Actuators**

**Plant**

\[ x_{t+1} = Ax_t + Bu_t + w_t \]

**Sensors**

\[ y_t = Cx_t + v_t \]

**State feedback**

\[ u_t^c = L\hat{x}_t \]

**Kalman filter**

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

\[ \hat{x}_t = E[x_t | y_t, y_{t-1}, \ldots, y_0, u_{t-1}^a, \ldots, u_1^a] \]

If \((u_{t-1}^a, \ldots, u_1^a)\) known \(\implies e_t = x_t - \hat{x}_t = f(y_t, y_{t-1}, \ldots, y_1, y_0)\)
Modeling of Digital Communication Network (DCN)

Assumptions:
(1) Quantization noise $<<$ sensor noise
(2) Packet-rate limited ($\neq$ bit-rate)
(3) No transmission noise (data corrupted=dropped packet)
(4) Packets are time-stamped
Estimation modeling
Minimum variance estimation

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

\[ \begin{align*}
  x_{t+1} &= Ax_t + w_t \\
  y_t &= Cx_t + v_t
\end{align*} \]

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

\[ \tilde{y}_k = \gamma_k^t(Cx_k + v_k) = C_k^tx_k + u^t \]

Kalman time-varying linear system

\[ \hat{x}_t = \mathbb{E}[x_t \mid \tilde{y}_1, \ldots, \tilde{y}_t, \gamma_1^t, \ldots, \gamma_t^t] \]
Minimum variance estimation

\[ t = 3 \]
\[
\begin{array}{c}
y_2 \\
y_3 \\
\end{array}
\]
\[
\hat{x}_0
\]

\[ \gamma = 0 \]
\[
\begin{align*}
\hat{x}^+ &= A\hat{x} \\
P^+ &= APA^T + Q
\end{align*}
\]
Lyapunov Equation (unstable)

\[ \gamma = 1 \]
\[
\begin{align*}
\hat{x}^+ &= A\hat{x} + K^T_k(y_k - CA\hat{x}) \\
P^+ &= APA^T + Q - APC^T(CPC^T + R)^{-1}CPA^T
\end{align*}
\]
Riccati Equation (stable)
Minimum variance estimation

\[ t = 3 \]

\[ \begin{array}{cc}
\gamma = 0 \\
\gamma = 1 \\
\end{array} \]

Lyapunov Equation (unstable)

\[
\begin{align*}
\hat{x}^+ &= A\hat{x} \\
P^+ &= APA^T + Q \\
\end{align*}
\]

Riccati Equation (stable)

\[
\begin{align*}
\hat{x}^+ &= A\hat{x} + K_k^t(\bar{y}_k^t - CA\hat{x}) \\
P^+ &= APA^T + Q - APC^T(CPC^T + R)^{-1}CPA^T \\
\end{align*}
\]
Minimum variance estimation

\[ t = 3 \]

\[ \hat{x}_2 \]

\[ y_2, y_3 \]

\[ \gamma = 0 \]

\[
\begin{align*}
\hat{x}^+ &= A\hat{x} \\
P^+ &= APA^T + Q
\end{align*}
\]

Lyapunov Equation (unstable)

\[ \gamma = 1 \]

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

Riccati Equation (stable)
Minimum variance estimation

\[ t = 3 \]

\[ \begin{array}{c}
  y_2 \\
  y_3 \\
  \hat{x}_3 \\
\end{array} \]

\[ \begin{align*}
  \gamma &= 0 \\
  \hat{x}^+ &= A\hat{x} \\
  P^+ &= APA^T + Q
\end{align*} \]

Lyapunov Equation (unstable)

\[ \begin{align*}
  \gamma &= 1 \\
  \hat{x}^+ &= A\hat{x} + K_k^T(y_k^T - CA\hat{x}) \\
  P^+ &= APA^T + Q - APC^T(CPC^T + R)^{-1}CPA^T
\end{align*} \]

Riccati Equation (stable)
Minimum variance estimation

Lyapunov Equation (unstable)

\[
\begin{align*}
\dot{x}^+ &= A\hat{x} \\
\gamma &= 0 \\
P^+ &= APA^T + Q
\end{align*}
\]

Riccati Equation (stable)

\[
\begin{align*}
\dot{x}^+ &= A\hat{x} + K_k^T(\gamma_k - CA\hat{x}) \\
\gamma &= 1 \\
P^+ &= APA^T + Q - APC^T(CPC^T + R)^{-1}CPA^T
\end{align*}
\]
Minimum variance estimation

\[ t = 4 \]
\[ \begin{bmatrix} y_2 \ y_3 \end{bmatrix} \]
\[ \hat{x}_1 \]

\[ \gamma = 0 \]
\[ \begin{align*}
\dot{\hat{x}}^+ &= A\hat{x} \\
P^+ &= APA^T + Q
\end{align*} \]
Lyapunov Equation (unstable)

\[ \gamma = 1 \]
\[ \begin{align*}
\dot{\hat{x}}^+ &= A\hat{x} + K_k^T(y_k^T - CA\hat{x}) \\
P^+ &= APA^T + Q - APC^T(CPC^T + R)^{-1}CPA^T
\end{align*} \]
Riccati Equation (stable)
Minimum variance estimation

\[ t = 4 \]

\[
\begin{array}{c|c}
  y_2 & y_3 \\
  \hline
  \hat{x}_2
\end{array}
\]

**Lyapunov Equation (unstable)**

\[
\begin{align*}
  \hat{x}^+ &= A\hat{x} \\
  P^+ &= APA^T + Q
\end{align*}
\]

**Riccati Equation (stable)**

\[
\begin{align*}
  \hat{x}^+ &= A\hat{x} + \frac{1}{\gamma}K_k^T(\breve{y}_k^T - CA\hat{x}) \\
  P^+ &= APA^T + Q - APC^T(CPC^T + R)^{-1}CPA^T
\end{align*}
\]
Minimum variance estimation

\[
\begin{align*}
t &= 4 \\
\begin{array}{c}
y_2 \\
\ldots \\
\ldots \\
\ldots \\
y_3
\end{array} \\
\hat{x}_3
\end{align*}
\]

\[\begin{align*}
\gamma = 0 \\
\hat{x}^+ &= A\hat{x} \\
P^+ &= APA^T + Q
\end{align*}\]

Lyapunov Equation (unstable)

\[\begin{align*}
\gamma = 1 \\
\hat{x}^+ &= A\hat{x} + K_k^T(y_k^T - CA\hat{x}) \\
P^+ &= APA^T + Q - APC^T(CPC^T + R)^{-1}CPA^T
\end{align*}\]

Riccati Equation (stable)
Minimum variance estimation

\[
\begin{align*}
\hat{x}_4 &= y_2 y_3 \\
t &= 4
\end{align*}
\]

Lyapunov Equation (unstable)

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

Riccati Equation (stable)

$$
\begin{align*}
\hat{x}^+ &= A\hat{x} + K^T_k(y_k^T - CA\hat{x}) \\
P^+ &= APA^T + Q - APC^T(CPC^T + R)^{-1}CPA^T
\end{align*}
$$

\(\gamma = 0\)

\(\gamma = 1\)
Minimum variance estimation

\[ t = 5 \]

\[ y_1 \rightarrow y_1 y_2 y_3 \]

\[ \hat{x}_4 \]

\[ \gamma = 0 \]

\[ \dot{x}^+ = A\hat{x} \]
\[ P^+ = APA^T + Q \]

Lyapunov Equation (unstable)

\[ \gamma = 1 \]

\[ \dot{x}^+ = A\hat{x} + K_k^t (\hat{y}_k^t - CA\hat{x}) \]
\[ P^+ = APA^T + Q - APC^T (CPC^T + R)^{-1} CPA^T \]

Riccati Equation (stable)
Properties of Optimal Estimator

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

ESTIMATOR

\[ y_{t-h} \rightarrow P_0 \square \rightarrow [x_0] \rightarrow y_1 * y_3 y_4 * y_6 \rightarrow \cdots \rightarrow t \rightarrow \hat{x}_t \]

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

\[ \hat{x}^+ = A\hat{x} + \gamma_{t-N}^t PC^T (CPC^T + R)^{-1}(\bar{y}_{t-N}^t - CA\hat{x}), \]
\[ P^+ = APA^T + Q - \gamma_{t-N}^t APC^T (CPC^T + R)^{-1}CPA^T \]
Minimum variance estimation

\[ t = 4 \]

\[ y_2 \quad y_3 \]

\[ \hat{x}_2 \]

\[ \gamma = 0 \]

\[ \begin{align*}
\dot{x}^+ & = A\hat{x} \\
P^+ & = APA^T + Q
\end{align*} \]

Lyapunov Equation (unstable)

\[ \gamma = 1 \]

\[ \begin{align*}
\dot{x}^+ & = A\hat{x} + K_k^T (y_k^T - CA\hat{x}) \\
P^+ & = APA^T + Q - APC^T (CPC^T + R)^{-1} CPA^T
\end{align*} \]

Riccati Equation (stable)
Minimum variance estimation

\[ t = 5 \]

\[ \hat{x}_3 \]

\[ y_2 \ y_3 \]

Lyapunov Equation (unstable)

\[ \dot{x}^+ = A\hat{x} \]
\[ P^+ = APA^T + Q \]

Riccati Equation (stable)

\[ \dot{x}^+ = A\hat{x} + K_k^t(y_k^t - CA\hat{x}) \]
\[ P^+ = APA^T + Q - APC^T(CPC^T + R)^{-1}CPA^T \]

\[ \gamma = 0 \]

\[ \gamma = 1 \]
What about stability and performance?

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

\( \tau_k \): delay of packet \( y_k \), \( \tau_k = \infty \) if \( y_k \) never arrives.

\[
\begin{align*}
\lambda_h & \triangleq \mathbb{P}[\tau_k \leq h], \\
\lambda_{\text{loss}} & \triangleq \mathbb{P}[\tau_k = \infty]
\end{align*}
\]
Optimal estimation with constant gains and buffer finite memory

\[
\{K_h\}_{h=0}^{N-1}, \quad N \text{ static gains}
\]

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

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

GOAL: compute \( \bar{P}_{t|t} \)
Suboptimal minimum variance estimation

\[ t = 3 \]

\[ K_3 \quad K_2 \quad K_1 \]

\[ \hat{x}_1 \]

\[ y_2 \quad y_3 \]

\[ \gamma = 0 \]

\[ \hat{x}^+ = A\hat{x} \]

\[ \gamma = 1 \]

\[ \hat{x}^+ = A\hat{x} + K_l(y_k^T - CA\hat{x}) \]

Open loop

Closed loop
Suboptimal minimum variance estimation

\[ t = 4 \]

\[ \begin{bmatrix} K_3 & K_2 & K_1 \\ y_2 & y_3 \end{bmatrix} \]

\[ \hat{x}_2 \]

\[ \gamma = 0 \]

\[ \hat{x}^+ = A\hat{x} \]

Lyapunov Equation (unstable)

\[ \gamma = 1 \]

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

Riccati Equation (stable)
Steady state estimation error

Fixed gains:

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

\[ \overline{P} = \mathcal{L}_{\lambda_{N-1}}(K_{N-1}, \overline{P}) \]

\[ \overline{P}_k = \mathcal{L}_{\lambda_k}(K_k, \overline{P}), \quad k = N - 2, \ldots, 0 \]

\[ \lim_{t \to \infty} \overline{P}_{t|t} = \overline{P} \]

Optimal fixed gains:

\[ \Phi_\lambda(P) = APA^T + Q - \lambda APCT(CPC^T + R)^{-1} CPA^T \]

\[ \min_{K_0, \ldots, K_{N-1}} \overline{P} \]

\[ \overline{P}_{N-1} = \Phi_{\lambda_{N-1}}(\overline{P}_{N-1}) \]

\[ \overline{P}_k = \Phi_{\lambda_k}(\overline{P}_{k+1}), \quad k = N - 2, \ldots, 0 \]

\[ K_k = \overline{P}_k C^T (C \overline{P}_k C^T + R)^{-1} \]

(off-line computation)

Modified Algebraic Riccati Equation (MARE)

(\(\Phi_1(P) = \text{ARE}\))
Stability issues

Static estimator is stable iff there exists $P \geq 0$ such that:

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

- If $\lambda = 0$ then standard ARE
- Modified Riccati Algebraic Equation known since [Nahi TIF’69]
- If $A$ is unstable then there exist critical probability: if $\lambda < \lambda_c$ stable, if $\lambda > \lambda_c$ unstable
- Upper bound $\lambda_c \leq \frac{1}{\max |\text{eig}(A)|^2}$. Equality if $C$ invertible [Katayama TAC”76]
- Lower bound $\lambda_c \geq \prod_{\text{unstable}} \frac{1}{|\text{eig}(A)|^2}$. Equality if rank$(C') = 1$ [Elia TAC’01, SCL’05]
- Closed form expression for $\lambda_c$ not known for general $(A, C)$
Numerical example (I)

Discrete time linearized inverted pendulum:

\[
A = \begin{bmatrix} 1.01 & 0.05 \\ 0.05 & 1.01 \end{bmatrix}, \quad C = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad R = 1, \quad Q = \begin{bmatrix} 0.01 & 0.01 \\ 0.01 & 1 \end{bmatrix}
\]
Numerical example (II)

Time-varying arrival probability distribution

\[ \lambda^1 \quad 0 \leq t \leq 50 \]

\[ \lambda^2 \quad t > 50 \]
Multiple sensors

PLANT
\[ x_{t+1} = Ax_t + w_t \]

SENSOR
\[ y_t^1 = C^1 x_t + v_t^1 \]

SENSOR
\[ y_t^2 = C^2 x_t + v_t^2 \]

\[ \ldots \ldots \]

SENSOR
\[ y_t^M = C^M x_t + v_t^M \]

Digital Communication Network

Estimator
\[ y_{t-\tau h} \]
\[ \hat{x}_t \]
Back to the control problem

Plant:
\[ x_{t+1} = Ax_t + Bu_t + w_t \]

Sensors:
\[ y_t = Cx_t + v_t \]

Actuators:
\[ u_t^a = u_t^c \]

State feedback:
\[ u_t^c = L\hat{x}_t \]

Static Kalman filter:
\[ \hat{x}_t = A\hat{x}_{t-1} + Bu_{t-1}^c + K(y_t - C\hat{x}_t) \]
Back to the control problem

\[ u_t^c = u_t^a \]

\[
\begin{align*}
\dot{x}_{t+1} &= Ax_t + Bu_t^a + w_t \\
 y_t &= Cx_t + v_t
\end{align*}
\]

State feedback

\[ u_t^c = L\hat{x}_t \]

Time-varying Kalman filter w/ memory

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

Estimation error coupled with control action \(\Rightarrow\) no separation principle
LQG over TCP-like (ACK-based) protocols

- Separation principle hold (I know exactly $u_{t-1}^a$)
- $\nu_t$ Bernoulli rand. var and independent of observation arrival process
- Static state feedback, $L_\nu$ solution of dual MARE
LQG over UDP-like (no-ACK) protocols

- 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 ( $u_{t-1}^a$ NOT known exactly)
- ... but still have some statistical information about $u_{t-1}^a$
LQG over UDP-like (no-ACK) protocols

- Bernoulli arrival process \( P[\nu_t = 1] = \bar{\nu}, P[\gamma_t = 1] = \bar{\gamma} \)
- \( \bar{\nu}u_{t-1}^c = E[u_{t-1}^\alpha] \)
- 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

\[ \begin{align*}
\min_{K,L} \quad & \operatorname{Trace} \left( \begin{bmatrix} W & 0 \\
0 & \bar{v}LTUL \end{bmatrix} P \right) \\
\text{s.t.} \quad & P = \mathbb{E} \left[ \begin{bmatrix} A & -\nu_kBL \\
\gamma_kKC & A - \bar{v}BL - \gamma_kKC \end{bmatrix} P \begin{bmatrix} A & -\nu_kBL \\
\gamma_kKC & A - \bar{v}BL - \gamma_kKC \end{bmatrix}^T \right] + \begin{bmatrix} Q & 0 \\
0 & \bar{\gamma}KRK^T \end{bmatrix} \\
P \geq 0
\end{align*} \]

- Non convex problem even for \( \nu=\gamma=1 \), i.e. classic LQG
- Classic and TCP-based LQG become convex when exploiting optimality conditions like uncorrelation 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, Bernstein) (maybe using Sum-of-Squares?)
- Stability on \( \nu \) and \( \gamma \) is coupled
Side note: Kalman filter is not always optimal!

- Kalman filter always gives smallest estimate error regardless of controller $L$
- If controller $L \neq L_{LQ}$, then performance improves if my estimate is “bad”!
Numerical example: TCP vs UDP
To hold or to zero control input?

Most common strategy:

\[ g(u_{t-1}^a) = 0 \quad \text{zero-input strategy (mathematically appealing)} \]
\[ g(u_{t-1}^a) = u_{t-1}^a \quad \text{hold-input strategy (most natural)} \]
To hold or to zero control input: no noise (jump linear systems)

**Zero-input Strategy**

\[ u_k^P = \nu_k u_k^C \]

\[ x_{k+1} = Ax_k + Bu_k^P \]

\[ u_k^C = L_z x_k \]

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

**Hold-input Strategy**

\[ u_k^P = \nu_k u_k^C + (1 - \nu_k) u_{k-1}^P \]

\[ x_{k+1} = Ax_k + Bu_k^P \]

\[ u_k^d = L_h x_k \]

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

Using cost-to-go function (dynamic programming)

\[ J^*_z = E[x_0^T S_z x_0] \]

\[ S_z = \Phi_z(S_z) \]

\[ L_z^* = f_z(S_z) \]

\[ J^*_h = E[x_0^T S_h x_0] \]

\[ S_h = \Phi_h(S_h) \]

\[ L_h^* = f_h(S_h) \]
Example: unstable scalar system

$A=1.2, U=0$ (fastest convergence)

$A=1.2, U=10$ (small input)
LQG over TCP-like protocols revised

Separation principle hold

Optimal function \( g(u^a_{t-1}) = \rho u_{t-1} \)

Design parameter \( L, l, \rho \) obtained via LQ-like optimal state feedback
Smart sensors & smart actuators

\[ \begin{align*}
\mathbf{u}_t^c &\rightarrow \mathbf{u}_t^a \\
\text{Actuators} &\quad x_{t+1} = A x_t + B u_t^a + w_t \\
\text{Plant} &\quad y_t = C x_t + v_t \\
\text{Sensors} &\quad \mathbf{y}_t \\
\text{classic LQ controller} &\rightarrow \text{Time-varying kalman} \\
\hat{x}_{t-\tau} &\rightarrow \hat{x}_t \\
\end{align*} \]

no input packet loss

\[ \begin{align*}
\mathbf{u}_t^c &\rightarrow \hat{x}_{t-\tau} \\
\text{controller} &\quad \hat{x}_7 = E[x_t | y_6, y_5, \ldots, y_1] = E[x_7 | \hat{x}_6] \\
\end{align*} \]

"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

Arrival packet probability $\gamma$
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. $y_1, \ldots, y_m$ arrive at different times
  - distributed estimation & consensus $E[x|y_1, \ldots, y_N] \approx E[x|\hat{x}_{s1}, \hat{x}_{sN}]$

- **Multiple actuators**
  - trade-off between distributed control & centralized coordination
Distributed sensor fusion & Consensus-based estimation

Delay & packet loss prob.