Communication constraints and latency in Networked Control Systems

João P. Hespanha

Center for Control Engineering and Computation
University of California
Santa Barbara

UCSB

In collaboration with Antonio Ortega (USC) and Yonggang Xu (UCSB)
1. Feedback control over a digital channel with limited bit-rate:

2. Decentralized cooperative control with limited message-rate and delays
Control with finite bit-rate

Motivation: Control of systems with sensors and actuators far from each other, connected by a digital network.

E.g., control of an autonomous flying vehicle using measurements from a camera on the ground.
Questions:
1. What is the minimum bit-rate for which stabilization (boundedness) is possible?
2. How to divide the bits among the distinct components of the output?
3. How to choose quantization intervals?
Theorem: Stabilization is \textbf{not} possible with average bit-rate smaller than

\begin{align*}
    r_{\min} &= \frac{1}{\log 2} \sum_{\Re \lambda_i > 0} \lambda_i \\
    r_{\min} &= \frac{1}{\log 2} \sum_{|\lambda_i| > 1} \log \lambda_i
\end{align*}

\(\lambda_i\) \equiv \text{eigenvalues of } A
Proof outline

\[
u = Ky_q
\]

\[
\begin{align*}
\dot{x} &= Ax + Bu \\
y &= x
\end{align*}
\]

A stable \quad \Rightarrow \quad x \text{ decays to zero without control} \quad \Rightarrow \quad \text{minimum bit-rate is zero (just set } u := 0) \quad \Rightarrow \quad \text{can reduce } A \text{ to its unstable inv. subspace}

A \text{ has asympt. stable inv. subspace } S^- \quad \Rightarrow \quad \text{component of } x \text{ in } S^- \text{ goes to zero without control}

\therefore \text{ we will assume that all eigenvalues of } A \text{ have real part } \geq 0
Proof outline

\[ y_q(t) \]

encoder

decoder

\[ b_0 \quad b_1 \quad b_2 \quad b_3 \quad b_4 \]

\[ t_0 \quad t_1 \quad t_2 \quad t_3 \quad t_4 \quad t \]

\[ Y(t) = x(t) \]

\[ \mathcal{X}_0 \equiv \text{set to which } x(t_0) \text{ is known to belong after bit } b_0 \text{ is received} \]

\[ \mathcal{X}_1 \equiv \text{set to which } x(t_1) \text{ is known to belong after bit } b_1 \text{ is received} \]

...
Proof outline

Case 1: For some $k$, $\mathcal{X}_k$ has a single element
\[ \implies \text{could take } x \text{ to zero (even in finite time) after } t_k \]

Case 2: As $k \to \infty$, $\rho(\mathcal{X}_k) \to 0$
\[ \implies \text{could take } x \text{ to zero as } k \to \infty \]

Case 3: As $k \to \infty$, $\rho(\mathcal{X}_k)$ unbounded
\[ \implies \text{no matter what control we use, } x(t_k) \text{ is unbounded} \]
Case 1: For some $k$, $X_k$ has a single element
⇒ could take $x$ to zero (even in finite time) after $t_k$

Case 2: As $k \to \infty$, $\rho(X_k) \to 0$
⇒ could take $x$ to zero as $k \to \infty$

Case 3: As $k \to \infty$, $\rho(X_k)$ unbounded
⇒ no matter what control we use, $x(t_k)$ is unbounded

Main idea:
bit rate $\leq r_{\text{min}}$ ⇒ $\mu(X_k)$ unbounded ⇒ $\rho(X_k)$ unbounded ⇒ diameter of $X_k$

Stabilization not possible
Proof outline

\[ x(t_{k+1}) = e^{A(t_{k+1}-t_k)}x(t_k) + \int_{t_k}^{t_{k+1}} e^{A(\tau-t_k)}u(\tau) \, d\tau \]

before \( b_{k+1} \) is received, it is only known that

\[ x(t_{k+1}) \in \mathcal{X}_{k+1}^- := e^{A(t_{k+1}-t_k)}\mathcal{X}_k + u_k \]

\[ \mu(\mathcal{X}_{k+1}^-) = |\det e^{A(t_{k+1}-t_k)}| \mu(\mathcal{X}_k) \]

Volume of \( \mathcal{X}_{k+1}^- \)

Volume of \( \mathcal{X}_k \)
Proof outline

before $b_{k+1}$ is received, it is only known that

$$x(t_{k+1}) = e^{A(t_{k+1} - t_k)}x(t_k) + \int_{t_k}^{t_{k+1}} e^{A(t - t_k)}u(t)\,dt$$

$$\in \mathcal{X}_k$$

$$u_k$$

$$\mathcal{X}_{k+1}^- := e^{A(t_{k+1} - t_k)}\mathcal{X}_k + u_k$$

$$\mu(\mathcal{X}_{k+1}^-) = |\det e^{A(t_{k+1} - t_k)}| \mu(\mathcal{X}_k)$$

$$\text{volume of } \mathcal{X}_{k+1}^- \quad \text{volume of } \mathcal{X}_k$$

Q: Which coding would make $\mu(\mathcal{X}_{k+1})$ as small as possible?

A: Divide $\mathcal{X}_{k+1}^-$ into two sets of equal volume & use bit $b_{k+1}$ to locate $x(t_{k+1})$ in one of them
coding that minimizes volume

$$\mu(\mathcal{X}_{k+1}) = \frac{\mu(\mathcal{X}_{k+1}^-)}{2}$$
Proof outline

At best...

\[
\mu(\mathcal{X}_{k+1}) \geq \frac{\mu(\mathcal{X}_{k+1}^-)}{2} = \frac{\left| \det e^{A(t_{k+1} - t_k)} \right|}{2} \mu(\mathcal{X}_k) = e^{(t_{k+1} - t_k) \sum \lambda_i} \mu(\mathcal{X}_k)
\]

with = only for coding that minimizes volume

iterating...

\[
\mu(\mathcal{X}_{k+1}) \geq e^{t_k(\sum \lambda_i) - k \log 2} \mu(\mathcal{X}_k)
\]

explodes if average bit-rate is smaller than

\[
r_{\min} := \frac{1}{\log 2} \sum \lambda_i
\]
Minimum bit-rate

Theorem: Stabilization is \textbf{not} possible with average bit-rate smaller than

$$r_{\min} := \frac{1}{\log 2} \sum_{\Re \lambda_i > 0} \lambda_i$$

continuous-time process

$$r_{\min} := \frac{1}{\log 2} \sum_{|\lambda_i| > 1} \log \lambda_i$$

discrete-time process

[Tatikonda & Mitter]

\(\lambda_i \equiv \text{eigenvalues of } A\)

\(\text{average bit-rate } \equiv r := \lim_{k \to \infty} \frac{k}{t_k}\)
Minimum bit-rate

\[ y_q(t) \rightarrow u(t) \rightarrow y(t) \]

\[ u = Ky_q \]

\[ \dot{x} = Ax + Bu \]

\[ y = x \]

Theorem: Assume \( A \) is diagonalizable

1. It is possible to keep the state bounded with any average bit rate larger or equal to \( r_{\text{min}} \)

2. It is possible to make the state converge to zero with any average bit rate strictly larger than \( r_{\text{min}} \)

But … minimum volume coding may not work because

unbounded volume \( \Rightarrow \) unbounded diameter

bounded volume \( \nRightarrow \) bounded diameter

average bit-rate \( \equiv \lim_{k \to \infty} \frac{k}{t_k} \)
Assume: closed-loop is stable for “transparent” encoding/decoding, i.e., $A + B K$ asymptotically stable

Questions:
1. How to design the encoder/decoder pair to make the closed-loop stable?
2. How much larger than $r_{\text{min}}$ does the bit-rate need to be for stability?
Inspired by *Differential Pulse Code Modulation* (DPCM)…

1. Encoder and decoder maintain consistent estimates of the state, based only on the quantized information sent to the decoder.

2. At sampling times, the difference between the measured state and its estimate (based on previously transmitted data) is quantized and transmitted digitally. (hopefully state estimation error has smaller dynamic range than state itself)

3. Upon transmission, the state-estimates are corrected using the quantized error.
1. Encoder and decoder maintain consistent estimates of the state, based only on the quantized information sent to the decoder.

- Process:
  \[ \dot{x} = Ax + B(u + d) \]
  \[ y = x + n \]

- Estimate:
  \[ \hat{x} = A\hat{x} + Bu \]
  \[ \hat{y} = \hat{x} \]
1. Encoder and decoder maintain consistent estimates of the state, based only on the quantized information sent to the decoder.

2. At sampling times, the difference between the measured state and its estimate is quantized and transmitted digitally.
1. Encoder and decoder maintain consistent estimates of the state, based only on the quantized information sent to the decoder.

2. At sampling times, the difference between the measured state and its estimate is quantized and transmitted digitally.

3. Upon transmission, the state-estimates are corrected using the quantized error:

\[
\hat{x}(t_k) = \hat{x}(t_k^-) + Q \left( \hat{y}(t_k^-) \right) = \hat{x}(t_k^-) + Q \left( y(t_k) - \hat{x}(t_k^-) \right)
\]

without quantization ($Q = \text{identity}$) and noise, we would have

\[
\hat{x}(t_k) = y(t_k) = x(t_k)
\]
1. Encoder and decoder maintain consistent estimates of the state, based only on the quantized information sent to the decoder.

2. At sampling times, the difference between the measured state and its estimate is quantized and transmitted digitally

3. Upon transmission, the state-estimates are corrected using the quantized error.
Fixed-step quantization

For simplicity…
- assume \( A \) diagonal with real eigenvalues
- quantize the vector by using a scalar quantizer on each of its components

If \( A \) was not diagonal one would precede the component-wise quantization by a diagonalizing linear transformation (see [JH, Ortega, Vasudevan, 02] for case of \( A \) with complex eigenvalues)

\[ \ell_i \leftrightarrow _{\tilde{y}_i} \]

\( K_i \) levels (equidistributed)

\( \ell_i \equiv \) saturation level for the \( i \)th component quantizer

\( K_i \equiv \# \) of quantization levels used for the \( i \)th component of \( \tilde{y} \)

\# of words needed \( \equiv N = \prod_{i=1}^{n} K_i \)

\( \) bit-rate \( \equiv \frac{\log_2 N}{T_s} = \frac{\sum_i \log K_i}{T_s \log 2} \)
Fixed-step quantization

**Theorem:** The state of the closed-loop system will remain bounded provided that

\[ K_i \geq \frac{\ell_i e^{\lambda_i T_s}}{\ell_i - \eta_i - \delta_i} \]

and

\[ |x_i(0)| \leq \ell_i - \eta_i \]

\( \eta_i, \delta_i \equiv \text{constants that depend on upper bounds on the noise/disturbance} \)

# of quant. levels

\( \lambda_i \text{ large} \Rightarrow e^{\lambda_i T_s} \text{ large} \Rightarrow K_i \text{ large} \Rightarrow \text{many bits needed for } i\text{th eigenspace} \)

*bit allocation...*
Theorem: The state of the closed-loop system will remain bounded provided that

\[ K_i \geq \frac{\ell_i e^{\lambda_i T_s}}{\ell_i - \eta_i - \delta_i} \]

\[ |x_i(0)| \leq \ell_i - \eta_i \]

\( \eta_i, \delta_i \equiv \text{constants that depend on upper bounds on the noise/disturbance} \)

\[ \lambda_i \text{ large } \Rightarrow e^{\lambda_i T_s} \text{ large } \Rightarrow K_i \text{ large } \Rightarrow \text{many bits needed for } i\text{th eigenspace} \]

required bit-rate...

\[ \text{bit-rate} \equiv \frac{\log_2 N}{T_s} \]

\[ N = \prod_{i=1}^{n} K_i \geq \prod_{i=1}^{n} \max \left\{ 1, \frac{\ell_i e^{\lambda_i T_s}}{\ell_i - \eta_i - \delta_i} \right\} \]

\[ \text{bit-rate} \geq \sum_{K_i > 1} \left( \frac{\lambda_i}{\log 2} + \frac{1}{T_s} \log_2 \frac{\ell_i}{\ell_i - \eta_i - \delta_i} \right) \]

\( \lambda_i \) large \( \Rightarrow \) \( e^{\lambda_i T_s} \) large \( \Rightarrow \) \( K_i \) large \( \Rightarrow \) many bits needed for \( i \)th eigenspace

\[ \lambda_i \text{ large } \Rightarrow e^{\lambda_i T_s} \text{ large } \Rightarrow K_i \text{ large } \Rightarrow \text{many bits needed for } i\text{th eigenspace} \]

\[ \text{bit-rate} \equiv \frac{\log_2 N}{T_s} \]

\[ N = \prod_{i=1}^{n} K_i \geq \prod_{i=1}^{n} \max \left\{ 1, \frac{\ell_i e^{\lambda_i T_s}}{\ell_i - \eta_i - \delta_i} \right\} \]

\[ \text{bit-rate} \geq \sum_{K_i > 1} \left( \frac{\lambda_i}{\log 2} + \frac{1}{T_s} \log_2 \frac{\ell_i}{\ell_i - \eta_i - \delta_i} \right) \]

\( \lambda_i \text{ large } \Rightarrow e^{\lambda_i T_s} \text{ large } \Rightarrow K_i \text{ large } \Rightarrow \text{many bits needed for } i\text{th eigenspace} \]

\[ \text{bit-rate} \equiv \frac{\log_2 N}{T_s} \]

\[ N = \prod_{i=1}^{n} K_i \geq \prod_{i=1}^{n} \max \left\{ 1, \frac{\ell_i e^{\lambda_i T_s}}{\ell_i - \eta_i - \delta_i} \right\} \]

\[ \text{bit-rate} \geq \sum_{K_i > 1} \left( \frac{\lambda_i}{\log 2} + \frac{1}{T_s} \log_2 \frac{\ell_i}{\ell_i - \eta_i - \delta_i} \right) \]

\( \lambda_i \text{ large } \Rightarrow e^{\lambda_i T_s} \text{ large } \Rightarrow K_i \text{ large } \Rightarrow \text{many bits needed for } i\text{th eigenspace} \]

\[ \text{bit-rate} \equiv \frac{\log_2 N}{T_s} \]

\[ N = \prod_{i=1}^{n} K_i \geq \prod_{i=1}^{n} \max \left\{ 1, \frac{\ell_i e^{\lambda_i T_s}}{\ell_i - \eta_i - \delta_i} \right\} \]

\[ \text{bit-rate} \geq \sum_{K_i > 1} \left( \frac{\lambda_i}{\log 2} + \frac{1}{T_s} \log_2 \frac{\ell_i}{\ell_i - \eta_i - \delta_i} \right) \]

\( \lambda_i \text{ large } \Rightarrow e^{\lambda_i T_s} \text{ large } \Rightarrow K_i \text{ large } \Rightarrow \text{many bits needed for } i\text{th eigenspace} \]

\[ \text{bit-rate} \equiv \frac{\log_2 N}{T_s} \]

\[ N = \prod_{i=1}^{n} K_i \geq \prod_{i=1}^{n} \max \left\{ 1, \frac{\ell_i e^{\lambda_i T_s}}{\ell_i - \eta_i - \delta_i} \right\} \]

\[ \text{bit-rate} \geq \sum_{K_i > 1} \left( \frac{\lambda_i}{\log 2} + \frac{1}{T_s} \log_2 \frac{\ell_i}{\ell_i - \eta_i - \delta_i} \right) \]

\( \lambda_i \text{ large } \Rightarrow e^{\lambda_i T_s} \text{ large } \Rightarrow K_i \text{ large } \Rightarrow \text{many bits needed for } i\text{th eigenspace} \]

\[ \text{bit-rate} \equiv \frac{\log_2 N}{T_s} \]

\[ N = \prod_{i=1}^{n} K_i \geq \prod_{i=1}^{n} \max \left\{ 1, \frac{\ell_i e^{\lambda_i T_s}}{\ell_i - \eta_i - \delta_i} \right\} \]

\[ \text{bit-rate} \geq \sum_{K_i > 1} \left( \frac{\lambda_i}{\log 2} + \frac{1}{T_s} \log_2 \frac{\ell_i}{\ell_i - \eta_i - \delta_i} \right) \]

\( \lambda_i \text{ large } \Rightarrow e^{\lambda_i T_s} \text{ large } \Rightarrow K_i \text{ large } \Rightarrow \text{many bits needed for } i\text{th eigenspace} \]

\[ \text{bit-rate} \equiv \frac{\log_2 N}{T_s} \]

\[ N = \prod_{i=1}^{n} K_i \geq \prod_{i=1}^{n} \max \left\{ 1, \frac{\ell_i e^{\lambda_i T_s}}{\ell_i - \eta_i - \delta_i} \right\} \]

\[ \text{bit-rate} \geq \sum_{K_i > 1} \left( \frac{\lambda_i}{\log 2} + \frac{1}{T_s} \log_2 \frac{\ell_i}{\ell_i - \eta_i - \delta_i} \right) \]

\( \lambda_i \text{ large } \Rightarrow e^{\lambda_i T_s} \text{ large } \Rightarrow K_i \text{ large } \Rightarrow \text{many bits needed for } i\text{th eigenspace} \]
Variable-step quantization

For simplicity…
- assume $A$ diagonal with real eigenvalues
- quantize the vector by using a scalar quantizer on each of its components

If $A$ was not diagonal one would precede the component-wise quantization by a diagonalizing linear transformation
(see paper for case of $A$ with complex eigenvalues)

\[ \ell_i(k) \equiv \text{saturation level for the } i\text{th component quantizer at sampling time } t_k \]

$K_i \equiv \# \text{ of quantization levels used for the } i\text{th component of } \tilde{y}$

\[ \ell_i \uparrow \quad \underbrace{\quad \quad \quad \quad \quad \quad \quad \quad} \quad K_i \text{ levels} \]

\[ \ell_i(k) \quad (\text{equidistributed}) \]

\[ \text{bit-rate} \equiv \frac{\log_2 N}{T_s} = \frac{\sum_i \log K_i}{T_s \log 2} \]
Variable-step quantization

**Theorem:** The state of the closed-loop system will remain bounded provided that

\[ K_i \geq e^{\lambda_i T_s}, \]

\[ \ell_i(k + 1) = \frac{e^{\lambda_i T_s}}{K_i} \ell_i(k) + \eta_i + \delta_i \]

\[ |x_i(0)| \leq \ell_i(0) - \eta_i \]

\[ \lambda_i \text{ large } \Rightarrow e^{\lambda_i T_s} \text{ large } \Rightarrow K_i \text{ large } \Rightarrow \text{many bits needed for } i \text{th eigenspace} \]

**bit allocation...**

required bit-rate...

\[ \text{bit-rate} \geq \frac{1}{\log 2} \sum_{\lambda_i > 0} =: r_{\text{min}} \]

\[ \text{minimum rate can be achieved !!!} \]
There exists a minimum rate below which stabilization is not possible.

We proposed encoder/decoder pairs (inspired by DPCM) that can achieve rates arbitrarily close to the minimal.

Variable-step quantization allows one to achieve the minimum bit rate.

Performance/robustness vs. bit-rate tradeoffs are still poorly understood.

Need to investigate problem in stochastic setting (Will entropy-like coding work with lower rates?)
Distributed Control

- physically distributed process to be controlled (e.g., autonomous vehicles)
- controller couplings supported by a communication network

- minimize controller communication (stealth, bandwidth)
- study the effect of nonideal communication (delays, drops, blackouts)
Scenarios

Rendezvous in minimum-time or using minimum-energy (in spite of disturbances)

Group of autonomous agents cooperate in searching for a target (perhaps mobile—search & pursuit)
Communication minimization

controller node
controller node
controller node

controller couplings supported by a communication network

The “every bit-counts” paradigm...

Goal: Design each controller to minimize the number of bits/second that need to be exchanged between nodes (quantization, compression, …)
Domain: Media with little capacity and low-overhead protocols (bit at-a-time)
E.g., underwater acoustic comm. between a small number of nodes.

The “cost-per-message” paradigm...

Goal: Design each controller to minimize the number of message exchanges between nodes (scheduling, estimation, …)
Domain: Media shared by a large number of nodes with nontrivial media access control (MAC) protocol (packet at-a-time)
E.g., 802.11 wireless comm. between a large number of nodes.
In this talk:

★ **decoupled** linear processes (with stochastic disturbance \(d\))

\[
x^+ = \begin{bmatrix} * & 0 & 0 \\ 0 & * & 0 \\ 0 & 0 & * \end{bmatrix} x + \begin{bmatrix} * & 0 & 0 \\ 0 & * & 0 \\ 0 & 0 & * \end{bmatrix} u + d
\]

★ **coupled** quadratic control objective

\[
\sum_{k=0}^{\infty} x' \begin{bmatrix} * & * & 0 \\ * & * & * \\ 0 & * & * \end{bmatrix} x + \|u\|^2
\]

notation: \(x^+ \equiv x(k+1)\)
Prototype problem

In this talk:

- **decoupled** linear processes (with stochastic disturbance $d$)

$$x^+ = \begin{bmatrix} A_1 & 0 \\ 0 & A_2 \end{bmatrix} x + \begin{bmatrix} B_1 \\ 0 \\ B_2 \end{bmatrix} u + d$$

- **coupled** quadratic control objective

$$\sum_{k=0}^{\infty} x' \begin{bmatrix} C_1' & C_1 \\ -C_2'C_1 & C_2' \end{bmatrix} x + \| u \|^2$$

E.g., rendez-vous of two vehicles

notation: $x^+ \equiv x(k+1)$
Prototype problem

In this talk:

★ *decoupled* linear processes
(with stochastic disturbance $d$)

$$x^+ = \begin{bmatrix} * & 0 & 0 \\ 0 & * & 0 \\ 0 & 0 & * \end{bmatrix} x + \begin{bmatrix} * & 0 & 0 \\ 0 & * & 0 \\ 0 & 0 & * \end{bmatrix} u + d$$

Minimum-cost solution (centralized)

$$u = \begin{bmatrix} * & * & * \\ * & * & * \\ * & * & * \end{bmatrix} x$$

Completely decentralized solution

$$u = \begin{bmatrix} * & 0 & 0 \\ 0 & * & 0 \\ 0 & 0 & * \end{bmatrix} x$$
Communication performance trade-off

- Optimal communication
- Achievable cost/comm. pairs
- Non-achievable cost/comm. pairs
- Minimum comm. needed for solvability (zero, when decentralized solution yields finite cost)
- Minimum-cost (centralized)
Centralized architecture

\( i^{th} \) local process
\[
x_i^+ = A_i x_i + B_i u_i + d_i
\]

\( i^{th} \) local controller
\[
u_i = K_{ii} x_i + K_{ij} x_j
\]

Closed-loop system
\[
x_i^+ = (A_i + B_i K_{ii}) x_i + B_i K_{ij} x_j + d_i
\]
\[
x_j^+ = (A_j + B_j K_{jj}) x_j + B_j K_{ji} x_i + d_j
\]

constant communication between local and external process(es)

\( j^{th} \) external process
\[
x_j^+ = A_j x_j + B_j u_j + d_j
\]

for simplicity here we assume only two processes
Estimator-based distributed architecture

$i$th local process
\[ x_i^+ = A_i x_i + B_i u_i + d_i \]

$i$th local controller
\[ u_i = K_{ii} x_i + K_{ij} \hat{x}_j \]

$i$th local estimator for $j$th external process
\[ \hat{x}_j^+ = A_j \hat{x}_j + B_j \hat{u}_j \]
\[ \hat{u}_j = K_{jj} \hat{x}_j + K_{ji} \hat{x}_i \]

Closed-loop system
\[ x_i^+ = (A_i + B_i K_{ii}) x_i + B_i K_{ij} x_j + d_i - B_i K_{ij} e_j \]
\[ e_i := x_i - \hat{x}_i \]
\[ x_j^+ = (A_j + B_j K_{jj}) x_j + B_j K_{ji} x_i + d_j - B_j K_{ji} e_i \]
\[ e_j := x_j - \hat{x}_j \]

jth external process
\[ x_j^+ = A_j x_j + B_j u_j + d_j \]

additive perturbation w.r.t centralized equations

[Yook & Tilbury, Montestruque & Antsaklis, Xu & JH]
**Communication logic**

\[ \hat{x}_j^+ = A_j \hat{x}_j + B_j \hat{u}_j \]

**How to fuse data?**

With no noise & delay (for now…)

\[ x_j(k) \text{ received from network at time } k \]

\[ \hat{x}_j(k + 1) = A_j x_j(k) + B_j u_j(k) \]

"best-estimate" based on data received

\[ e_j(k + 1) = -d_j(k) \]

**When to send data?**

Scheduling logic action

\[ a_j(k) = \begin{cases} 1 & x_j(k) \text{ sent at time } k \\ 0 & \text{no transmission at } k \end{cases} \]

Options…

- periodically
  \[ a_j(k) = \{ k \text{ divisible by } T \}, \quad T \in \mathbb{N} \]
- feedback policy
  \[ a_j(k) = F(x_j(k), \ldots) \]
- “optimal” …
Optimal Scheduling Logic

**Goals:**
- minimize the estimation error $\Rightarrow$ minimize cost-penalty w.r.t. centralized
- minimize the number of transitions $\Rightarrow$ minimize communication bandwidth

\[
\limsup_{T \to \infty} E \left[ \frac{1}{T} \sum_{k=0}^{T-1} e_j(k)'Qe_j(k) \right]
\]
\[
\text{average L-2 norm}
\]
\[
\min_{a(k)} \limsup_{T \to \infty} E \left[ \frac{1}{T} \sum_{k=0}^{T-1} e_j(k)'Qe_j(k) + \lambda a_j(k) \right]
\]
\[
\text{relative weight of two criteria (will lead to Pareto-optimal solution)}
\]

- relative weight of two criteria (will lead to Pareto-optimal solution)

\[
x_j^+ = A_j x_j + B_j u_j + d_j
\]
Dynamic Programming solution

\[
\min_{a(k)} \lim_{T \to \infty} \text{sup} \ E \left[ \frac{1}{T} \sum_{k=0}^{T-1} e(k)'Qe(k) + \lambda a(k) \right]
\]

undiscounted average-cost problem

Theorem

\((TV)(e) := \min_a E \left[ e^+'Qe^+ + \lambda a + V(e^+) \mid e \right] \)

dynamic programming (DP) operator

1. There exists \( J^* \in \mathbb{R} \) and bounded \( h^* : \mathbb{R}^n \to \mathbb{R} \) such that
   \[
   h^*(0) = 0, \quad h^* + J = Th^*
   \]

2. \( J^* \) is the optimal cost and is achieved by the (deterministic) static policy
   \[
   a(k) = \pi^*(e(k)), \quad \pi^*(e) := \begin{cases} 
   1 & E[h^*(Ae + d)] + e'A'QAe \geq E[h^*(d)] + \lambda \\
   0 & \text{otherwise}
   \end{cases}
   \]

3. \( h \) can be found by value iteration
   \[
   h_{i+1} = Th_i - (Th_i)(0) \quad \overset{\text{exp.}}{\underset{i \to \infty}{\longrightarrow}} \quad h^*
   \]
Proof outline:
1. $e(k)$ is Markov and its transition distribution satisfies an Ergodic property (requires a mild restriction on the set of admissible policies omitted here)
2. $T$ is a span-contraction [Hernandez-Lerma 96]
3. Result follows using standard arguments based on Banach’s Fixed-Point Theorem for semi-norms.

Theorem

\[
(TV)(e) := \min_a \mathbb{E} \left[ e^+ Q e^+ + \lambda a + V(e^+) \mid e \right]
\]

1. There exists $J^* \in \mathbb{R}$ and bounded $h^* : \mathbb{R}^n \to \mathbb{R}$ such that

\[
h^*(0) = 0, \quad h^* + J = Th^*
\]

2. $J^*$ is the optimal cost and is achieved by the (deterministic) static policy

\[
a(k) = \pi^*(e(k)), \quad \pi^*(e) := \begin{cases} 1 & \mathbb{E}[h^*(Ae + d)] + e' A' Q A e \geq \mathbb{E}[h^*(d)] + \lambda \\ 0 & \text{otherwise} \end{cases}
\]

3. $h$ can be found by value iteration

\[
h_{i+1} = Th_i - (Th_i)(0) \xrightarrow{i \to \infty} h^*
\]
Example (2-dim)

$$\lim_{T \to \infty} \sup \mathbb{E} \left[ \frac{1}{T} \sum_{k=0}^{T-1} e(k)' \begin{bmatrix} 4 & 0 \\
0 & 1 \end{bmatrix} e(k) + \lambda a(k) \right]$$

The optimal scheduling is given by:

$$a(k) = \begin{cases} 0 & \text{for } e(1) < 0 \\ 1 & \text{for } e(1) \geq 0 \end{cases}$$

The local process is:

$$A := A_j + B_j K_j = \begin{bmatrix} 1 & 1 \\
.1 & .9 \end{bmatrix}$$

$$\mathbb{E} \left[ d(k)d(k)' \right] = \begin{bmatrix} 1 & 0 \\
0 & 1 \end{bmatrix}$$

Not ellipses!
Example (2-dim)

$$\lim_{T \to \infty} \sup T^{-1} \sum_{k=0}^{T-1} e(k)' \begin{bmatrix} 4 & 0 \\ 0 & 1 \end{bmatrix} e(k) + \lambda a(k)$$

$$a(k) = \begin{cases} 0 & \text{small weight in communication cost} \\ 1 & \text{large weight in communication cost} \end{cases}$$

$$\lambda = 100$$

$$a(k) = 1$$

$$\lambda = 10$$

$$A := A_j + B_j K_j = \begin{bmatrix} 1 & 1 \\ .1 & .9 \end{bmatrix}$$

$$E[d(k)d(k)'] = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

local process

large weight in comm. cost

$$\downarrow$$

large error threshold

$$\downarrow$$

only communicate when error is very large

$$e_2$$

$$e_1$$

-10

0

5

10

0

5

10

0

5

10
Communication with latency

\( \hat{x}^+_j = A_j \hat{x}_j + B_j \hat{u}_j \)

Sched. logic sends \( x_j(k) \)

Fusion logic recs. \( x_j(k) \)

\( x_j(k) \) is incorporated into estimate

\( \tau = 0 \) in previous case
Example (1-dim)

$$\lim \sup_{T \to \infty} E \left[ \frac{1}{T} \sum_{k=0}^{T-1} e(k)^2 + \lambda \alpha(k) \right]$$

\[ A := A_j + B_j K_j = 2 \]

\[ E[d(k)^2] = .01, \ \forall k \]

optimal scheduling

\[ \alpha(k) = \begin{cases} 1 & |\bar{e}(k)| \geq e_{\text{threshold}} \\ 0 & \text{otherwise} \end{cases} \]

with network latency
same error variance
requires more bandwidth
We constructed communications logics that minimize communication (measured in messages sending rate)

We considered networks with (fixed) latency

Study the effect of *packet losses* (especially important in wireless networks)

Coupled control/communication-logic design

Nonlinear processes

Papers available at [http://www.ece.ucsb.edu/~hespanha](http://www.ece.ucsb.edu/~hespanha)