# Some Further Directions

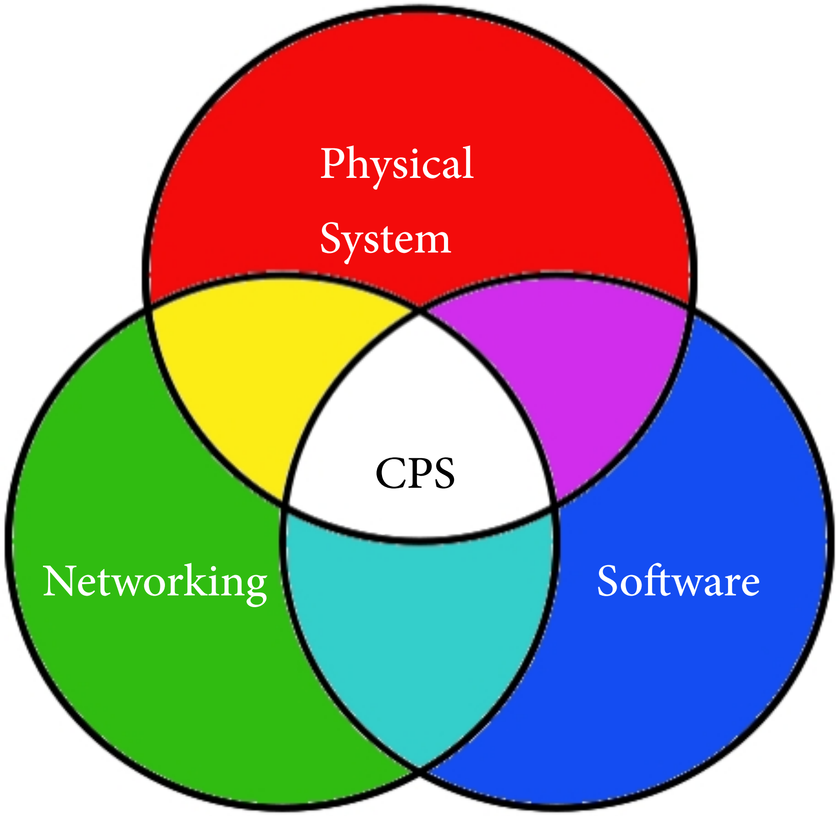
Panos J. Antsaklis, Bill Goodwine, and Vijay Gupta

UNIVERSITY OF NOTRE DAME

Joint work with Y. Wang, M. Xia, Y. Zhao

NSF CPS Large Project Meeting, 2013

# Overview



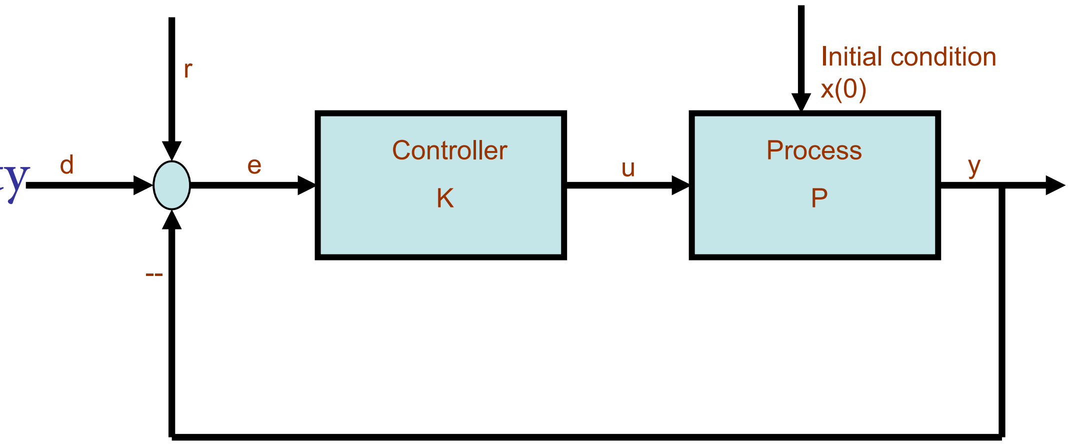

Need:

- Compositionality
- Composability
- Scalability
- Robustness to uncertainty

Technical tools:

- Passivity
- Symmetry
- Event-triggered schemes
- Anytime schemes

# Scalability of performance

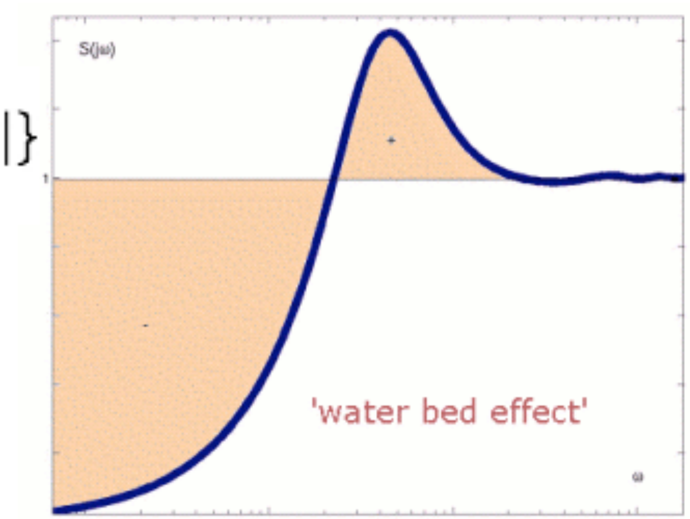



- Beyond stability, need performance

- Choose classical metric of sensitivity to disturbance

- Fundamental performance limitation: Sensitivity (from random disturbance to error) cannot be reduced at all frequencies

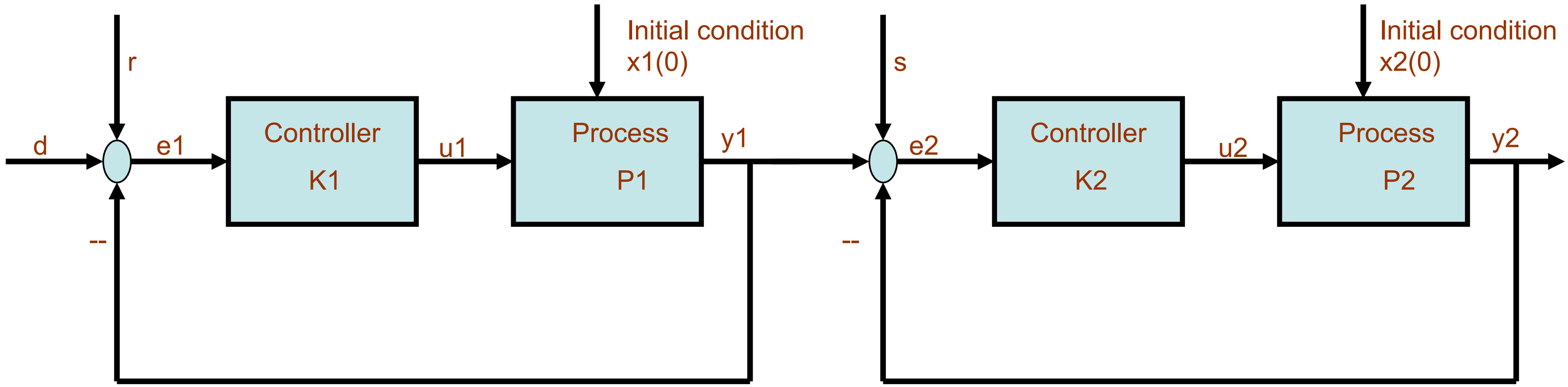
$$\frac{1}{2\pi}\int_{-\pi}^{\pi} \log \mid S_{d,e}(\omega) \mid d\omega = \sum_{j=1}^{n} \max\{0, \log \mid \lambda_j(A) \mid\}$$



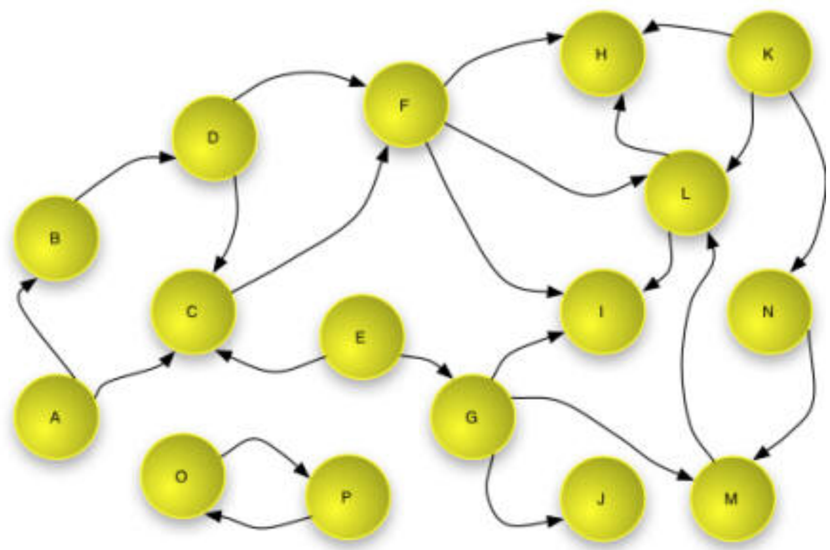

- Holds for any second moment stabilizing controller

- Many extensions in the literature

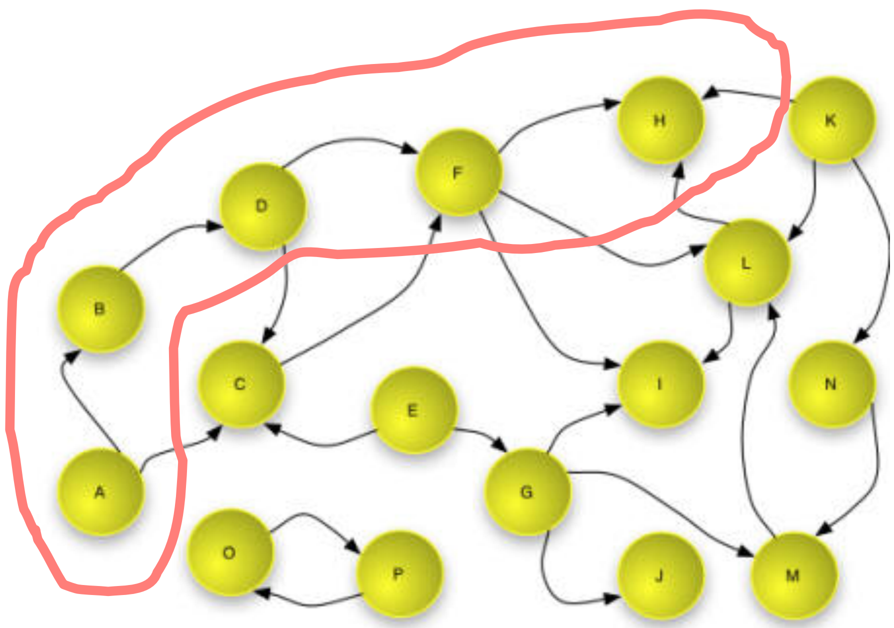
# Performance in large scale systems




- In general, a hard problem (related to distributed control)
- Interesting issues related to information transfer and usage
- How to design controllers?
- How to design information flow topologies?
- Focus on fundamental performance limitations

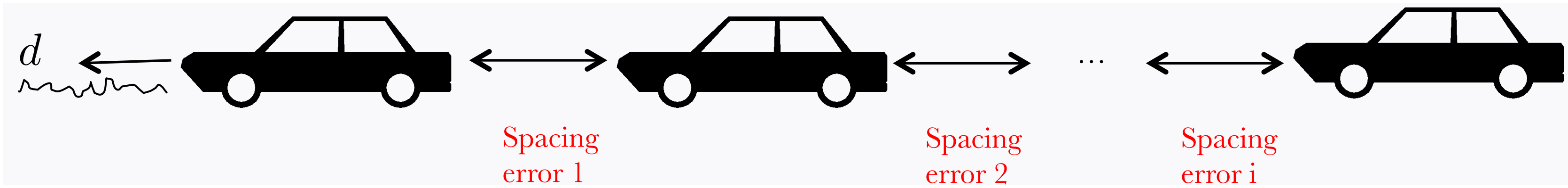


# 1-D Formations




- Maintain constant spacing wrt predecessor
- Independent interest in vehicle platooning applications
- Cost (or error) graph and information flow graph identical
- Error propagation due to coupled cost


# Prior Work




- String stability (Peppard (1974), Swaroop and Hedrick (1996))

- Disturbance propagation performance (Seiler et al (2004), Middleton and Braslavasky (2010))

Predecessor following strategy

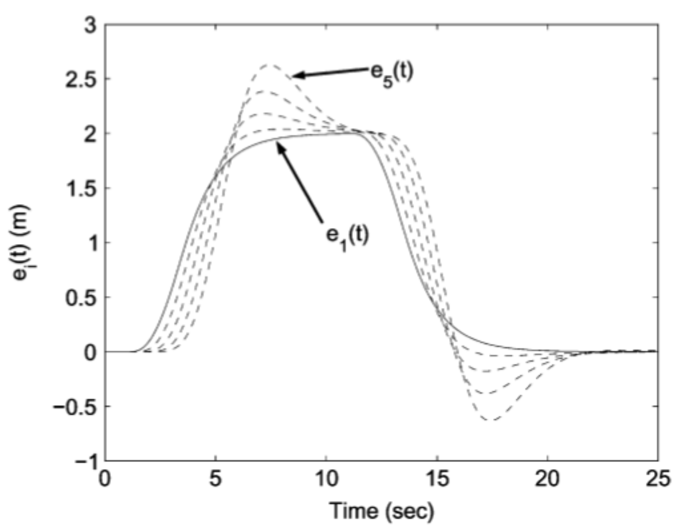



Fig. 5. Time domain plots of spacing errors with the predecessor following strategy.

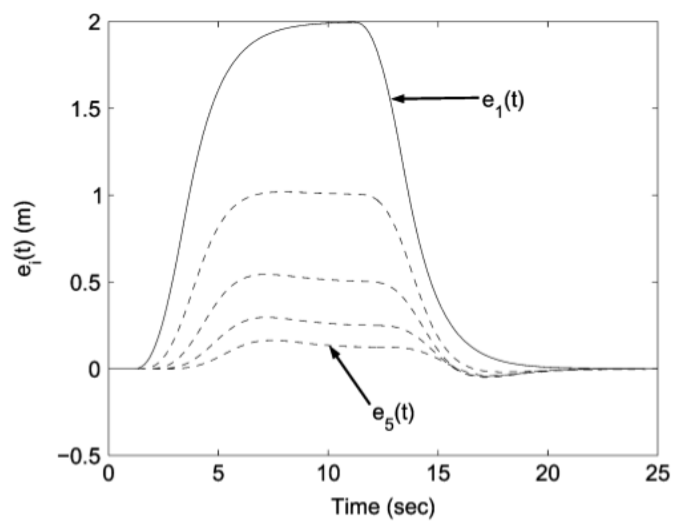



Fig. 6. Time domain plots of spacing errors with the predecessor and leader following strategy.

Predecessor and leader following strategy

## Summary of Results

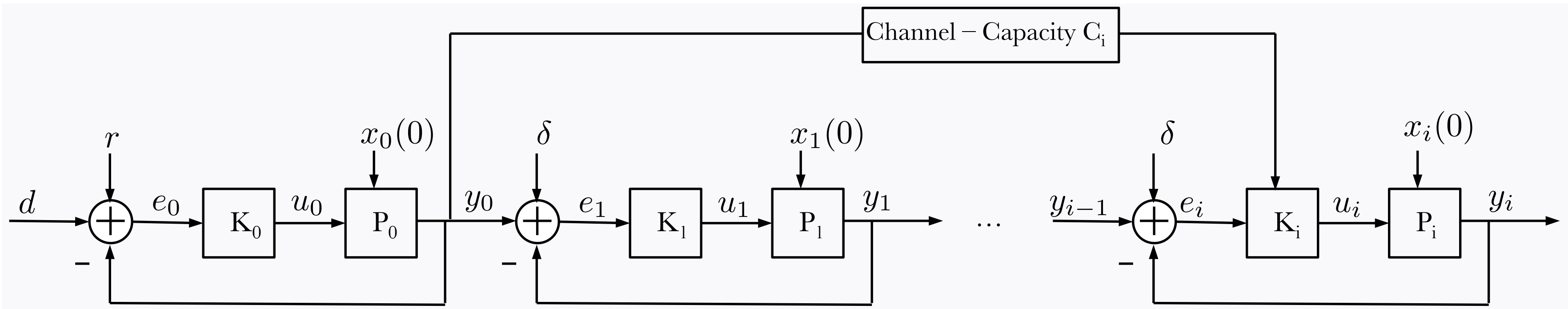
- We consider the sensitivity of the agents' position (error) with respect to an external disturbance affecting the leader

- Obtain a generalization of Bode's integral formula to distributed systems in this setting

- Fundamental limitation that holds for any plant, non-linear controllers, information flow across finite capacity channels

- Use information theoretic tools to obtain a result in distributed control

# Problem Framework




For i-th SISO process

$$x_i(k) = A_i x_i(k-1) + B_i u_i(k-1)$$
$$y_i(k) = C_i x_i(k)$$

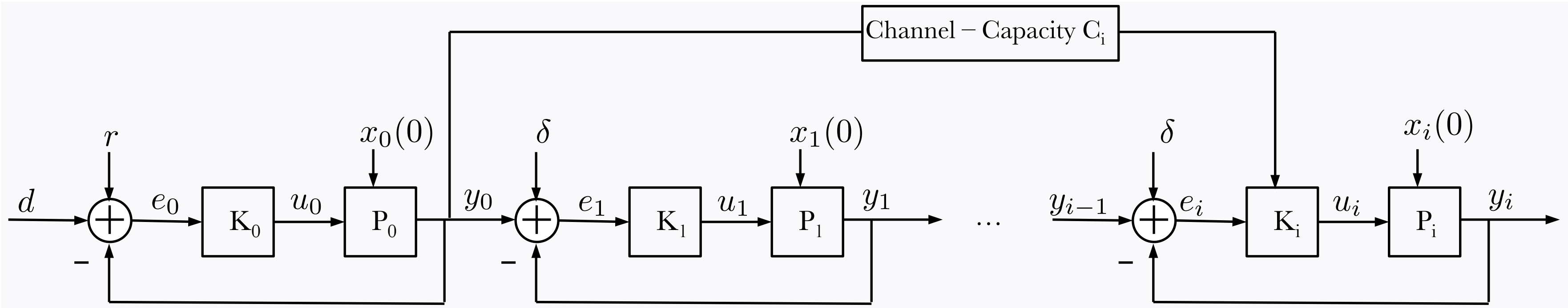
Errors

$$e_0(k) = r_0(k) + d(k) - y_0(k)$$
$$e_i(k) = y_{i-1}(k) - y_i(k) - s \qquad 1 \leq i \leq M$$

Cost function

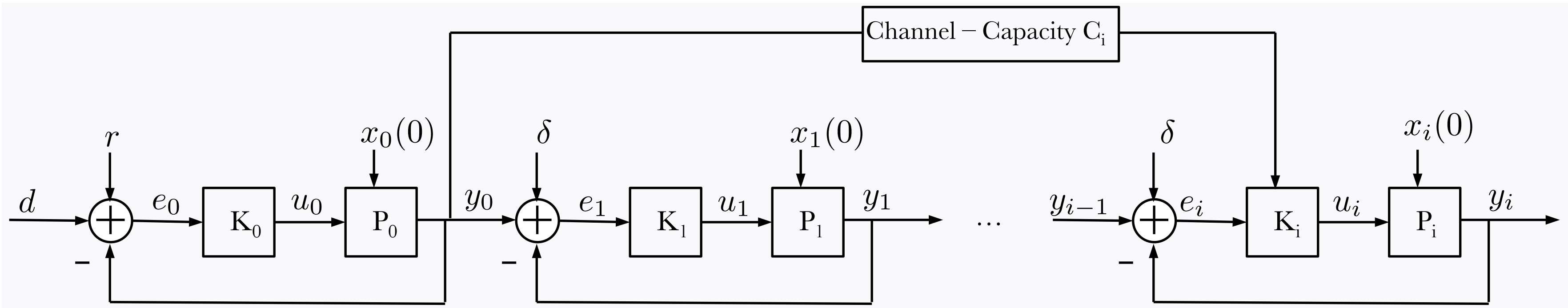
$$S_{d,e_i}(\omega) = \sqrt{\frac{\phi_{e_i}(\omega)}{\phi_d(\omega)}} \qquad \phi(\omega)\text{: power spectral density}$$

# Technical Assumptions




- Disturbance is AWGN, independent of initial conditions
- Initial conditions define a Markov chain
- All subsystems are closed loop mean squared stable, observable
- All processes are strictly proper
- Control is deterministic, piecewise continuous, bijective function of own error; can be time-varying, non-linear and with internal state

# Main Result




For any i = 0, 1, …

$$\frac{1}{2\pi}\int_{-\pi}^{\pi} \log S_{\mathbf{d},\mathbf{e}_i}(\omega)d\omega \geq \sum_{j=0}^{i-1}\Big(\sum_{\beta\in\mathcal{UZ}_j} \log|\beta| + P_j + K_j - C_j\Big)^+ + \sum_{\lambda\in\mathcal{UP}_i} \log|\lambda| - C_i,$$

where

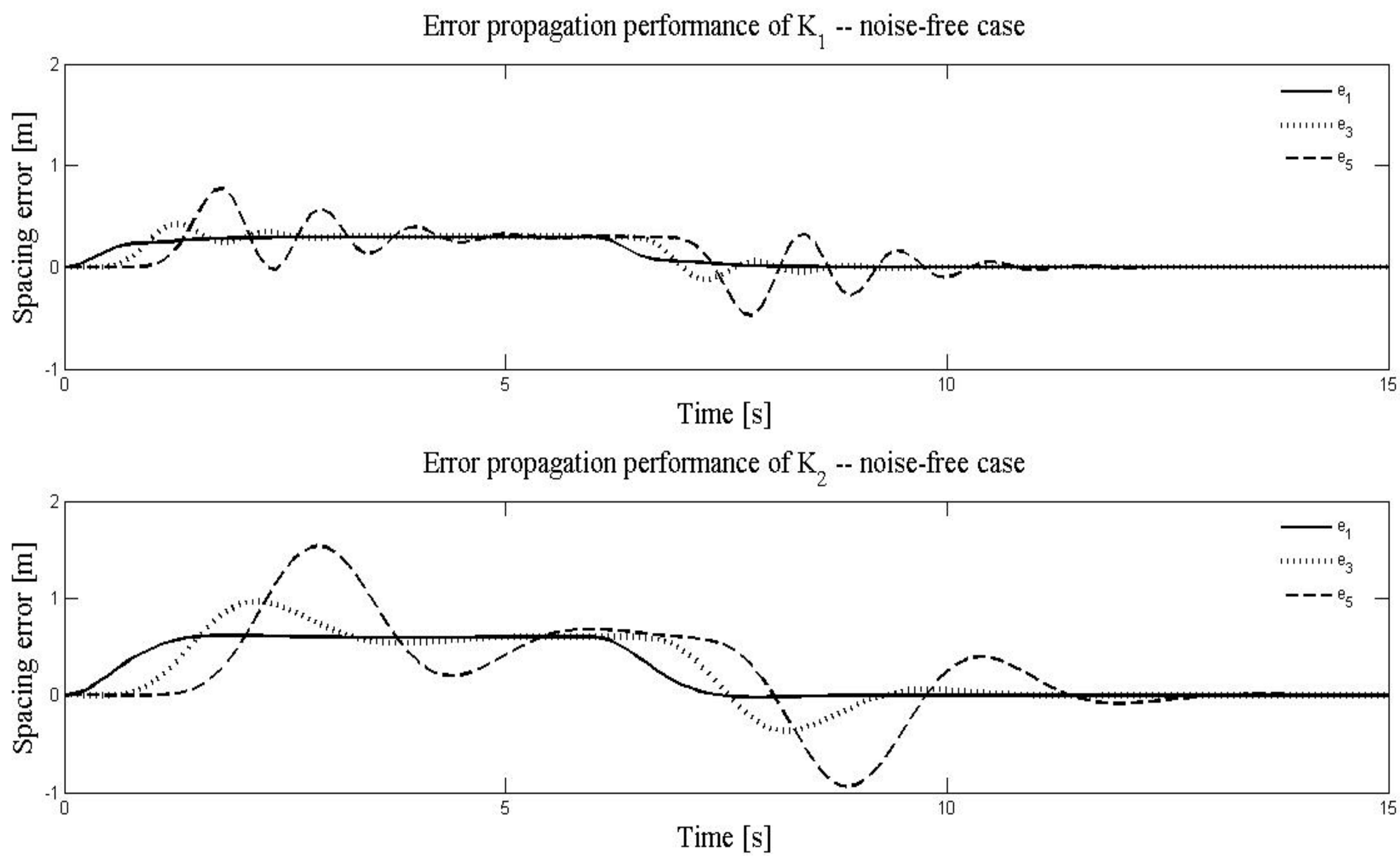
$$P_i = \log|\lim_{z\to\infty} z^{\alpha_i} P_i(z)|, \quad K_i = \liminf_{N\to\infty} \tfrac{1}{N}\textstyle\sum_{k=0}^{N} \mathbb{E}\big(\log|u'_{i,k}(\hat{y}_i^k, \mathbf{e}_i^k)|\big)$$

- Special case i = 0 recovers traditional Bode integral formula
- Channels reduce RHS, but there is a saturation effect
- Achievable in special (although non-trivial) cases

# Illustration

Consider the process $H(s) = \dfrac{1}{s^2(0.1s+1)}$ discretized with a step size 0.25

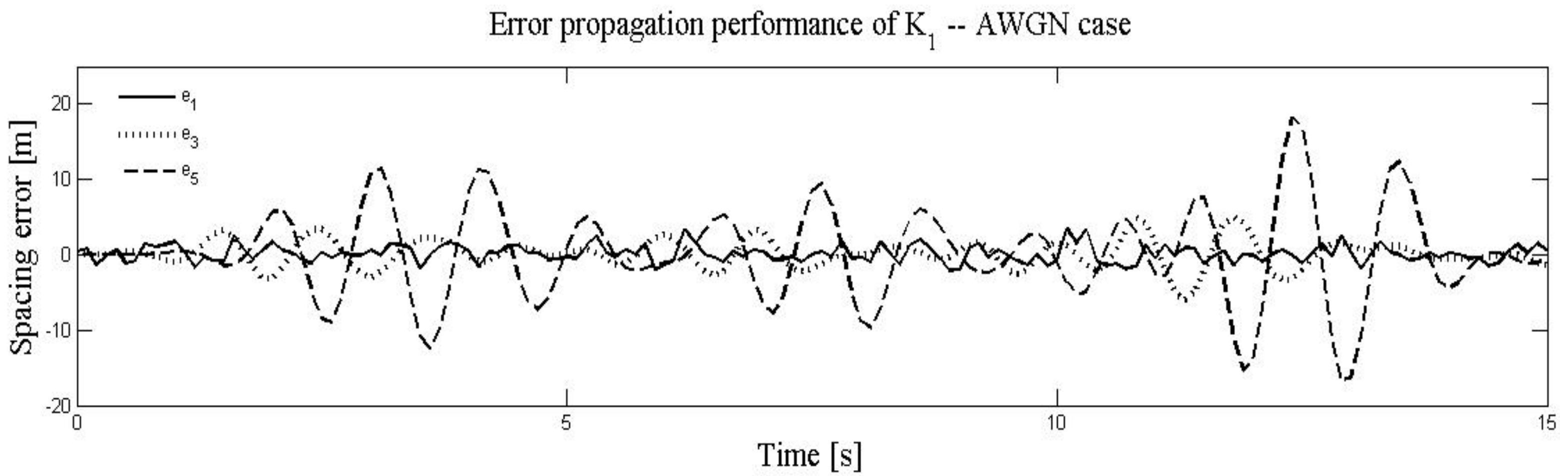
$$K(s) = K_i \frac{2s+1}{0.05s+1}, \qquad K_1 = 1, K_2 = 0.5$$

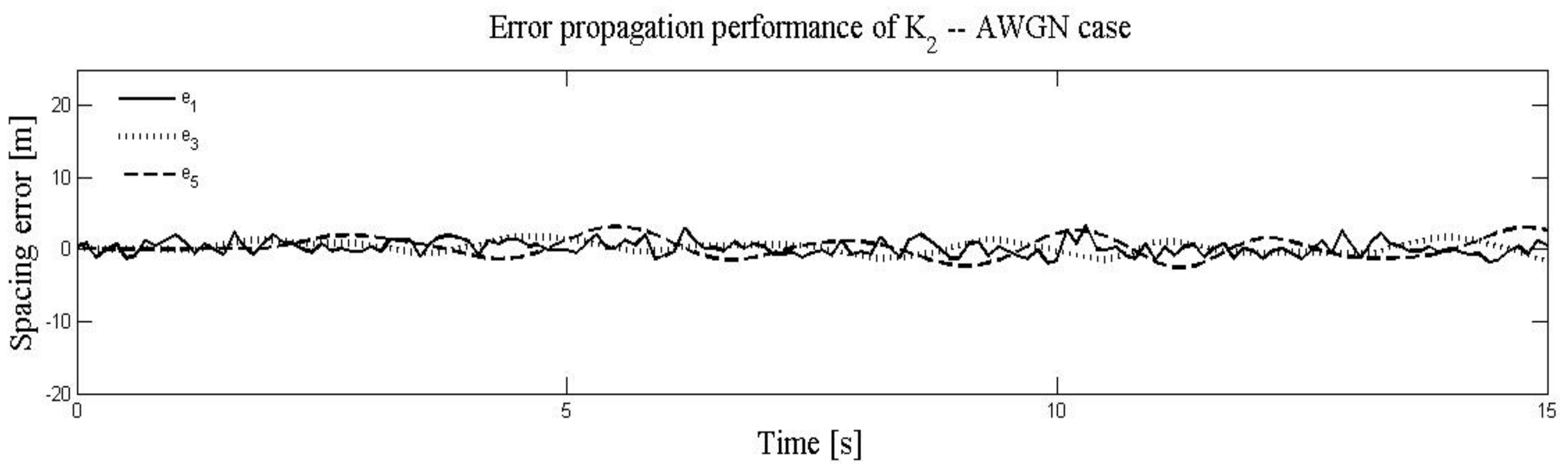


# Illustration

Consider the process $H(s) = \dfrac{1}{s^2(0.1s+1)}$ discretized with a step size 0.25

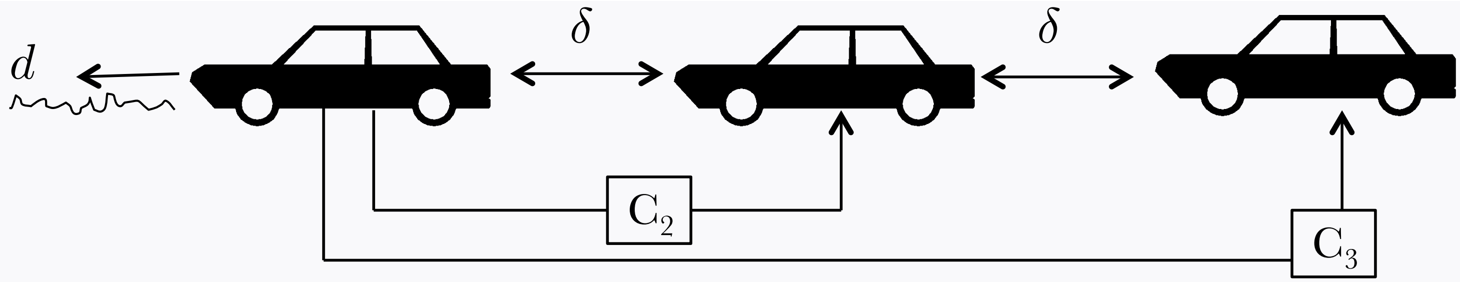
$$K(s) = K_i \frac{2s+1}{0.05s+1}, \qquad K_1 = 1, K_2 = 0.5$$



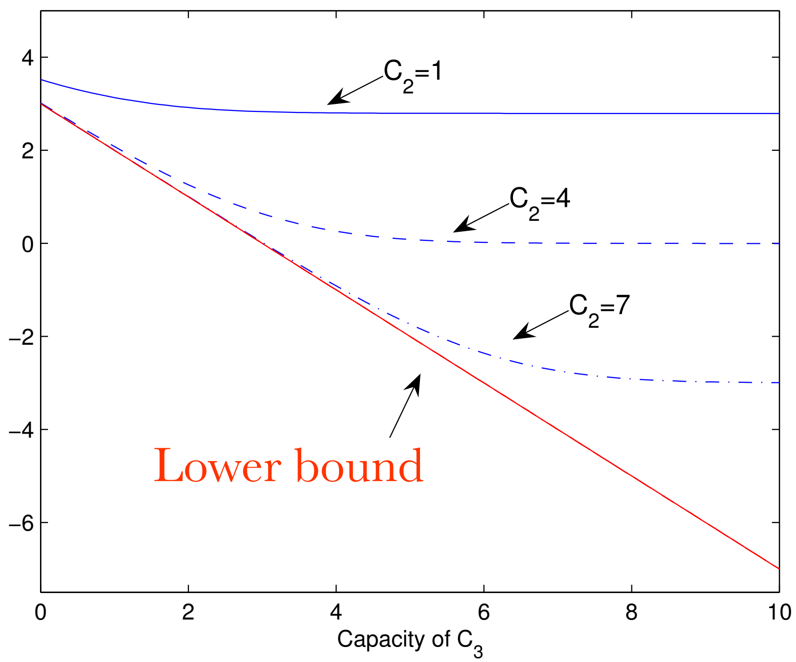



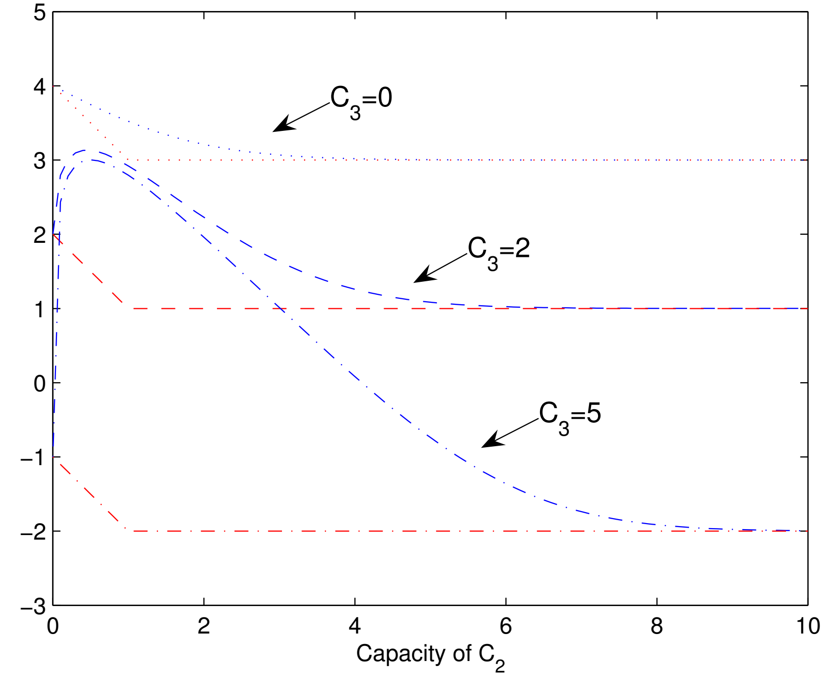
# Illustration

Specific Gaussian setting in which sensitivities can be calculated analytically

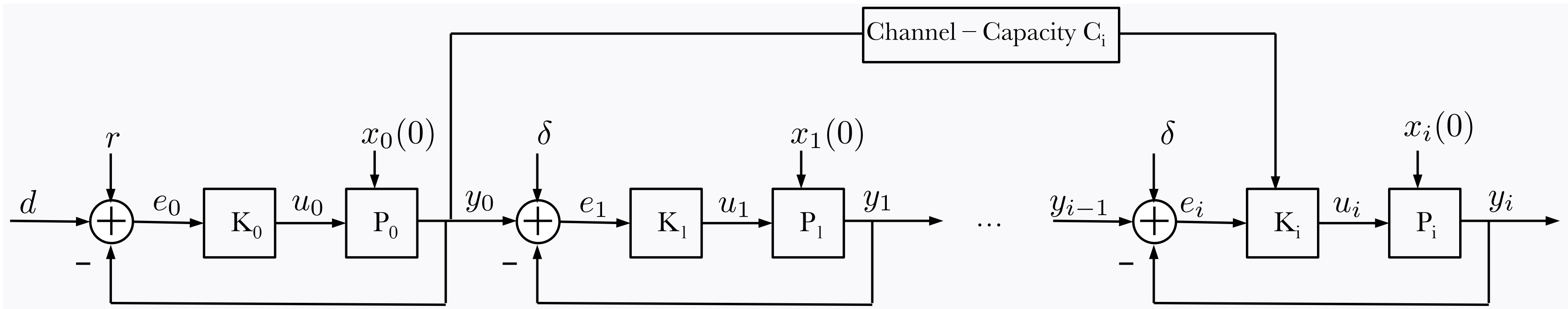



Lower bound vs Integral log sensitivity



# Performance Scalability

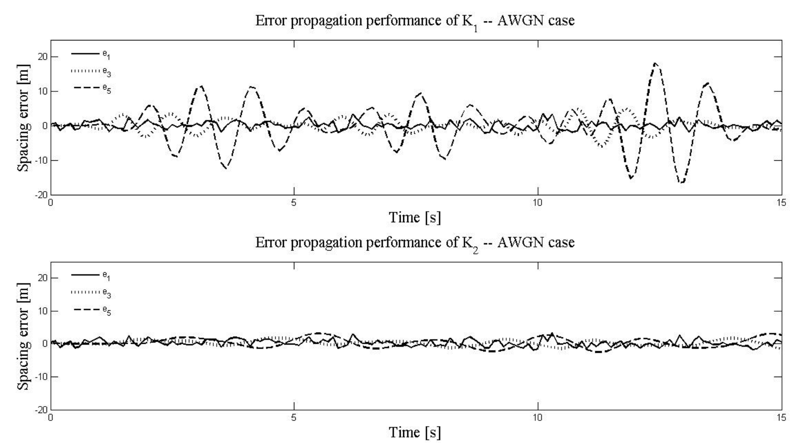



$$\frac{1}{2\pi}\int_{-\pi}^{\pi} \log S_{\mathbf{d},\mathbf{e}_i}(\omega)d\omega \geq \sum_{j=0}^{i-1}\Big(\sum_{\beta\in\mathcal{UZ}_j} \log|\beta| + P_j + K_j - C_j\Big)^+ + \sum_{\lambda\in\mathcal{UP}_i} \log|\lambda| - C_i,$$

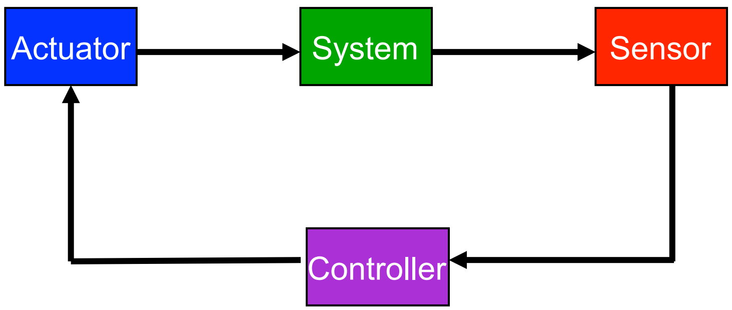
where

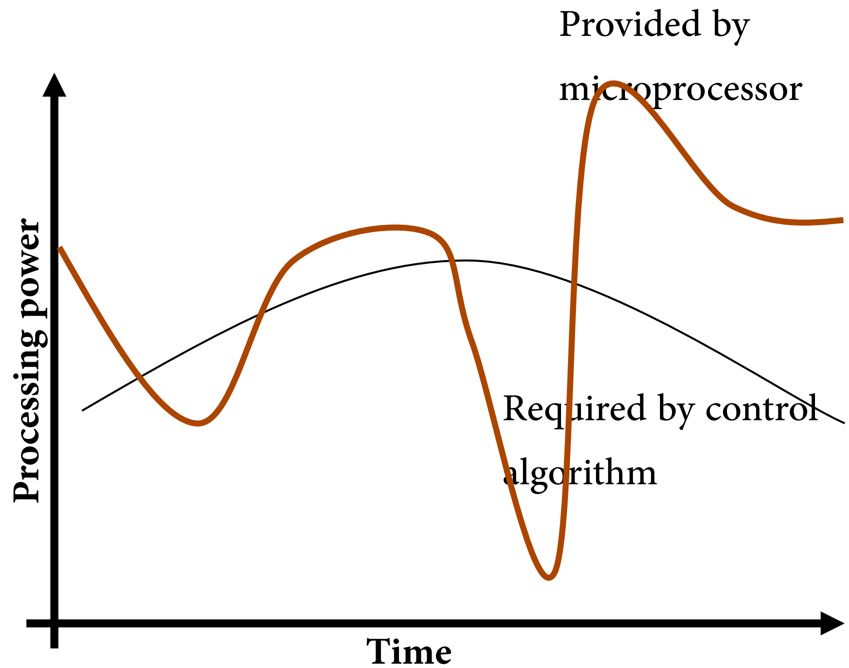
$$P_i = \log|\lim_{z\to\infty} z^{\alpha_i} P_i(z)|, \quad K_i = \liminf_{N\to\infty} \tfrac{1}{N}\textstyle\sum_{k=0}^{N} \mathbb{E}\big(\log|u'_{i,k}(\hat{y}_i^k, \mathbf{e}_i^k)|\big)$$

- Obtained information theoretic bounds on performance in strings
- Current work: process noise, cyclic graphs
- Passivity, LQG notions



# Algorithms for Limited Processor Availability



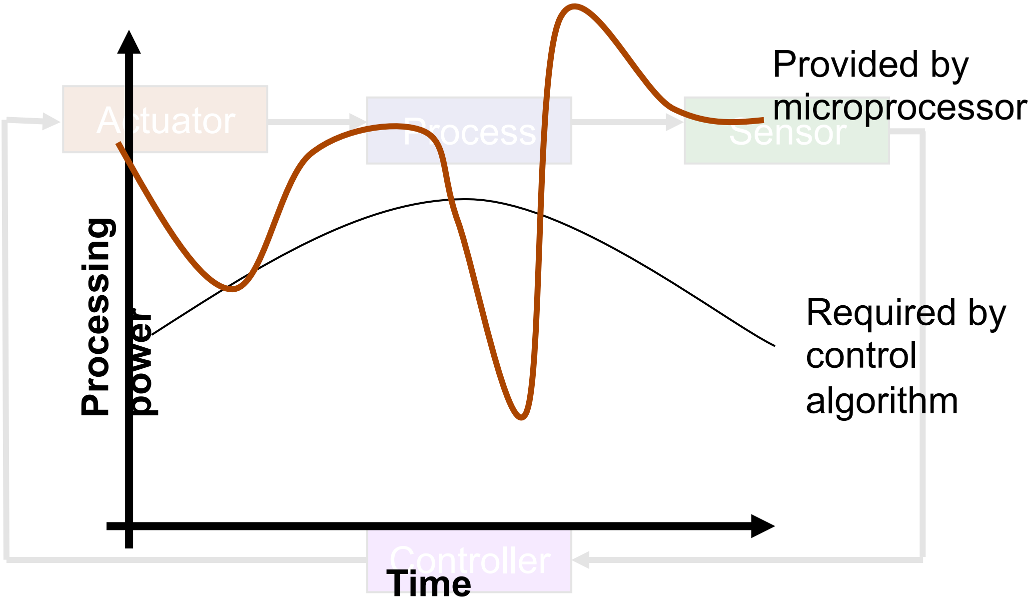



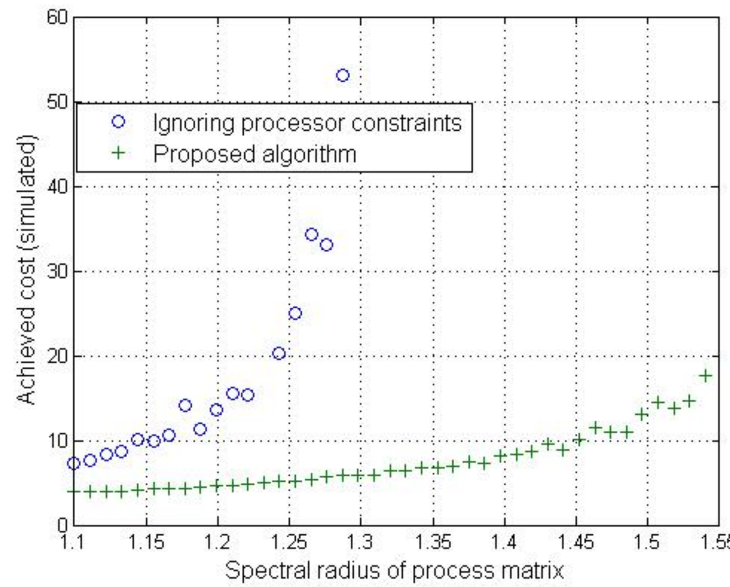

- Anytime control algorithms
- Event triggered control

# Last Year

Quevedo and Gupta, TAC 2013

Gupta and Luo, TAC 2013




- Anytime algorithms for control (a priori unknown execution time)

- Basic idea: Use coarse model of the process to generate a control input; then progressively use more complicated models to refine the input

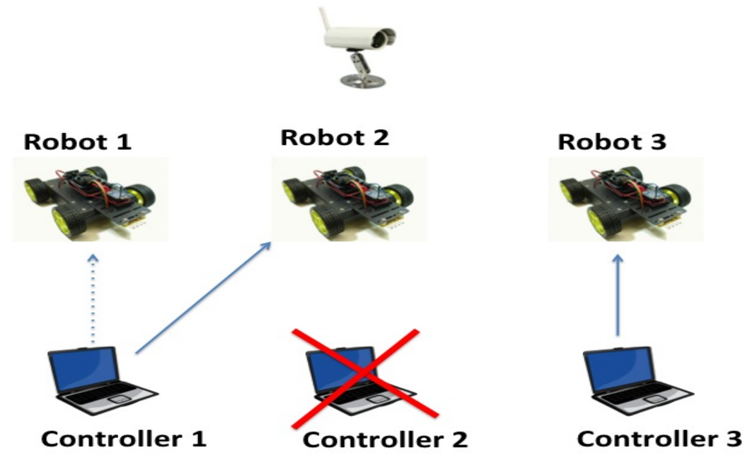
- Linear processes, RHC based extension, non-linear processes

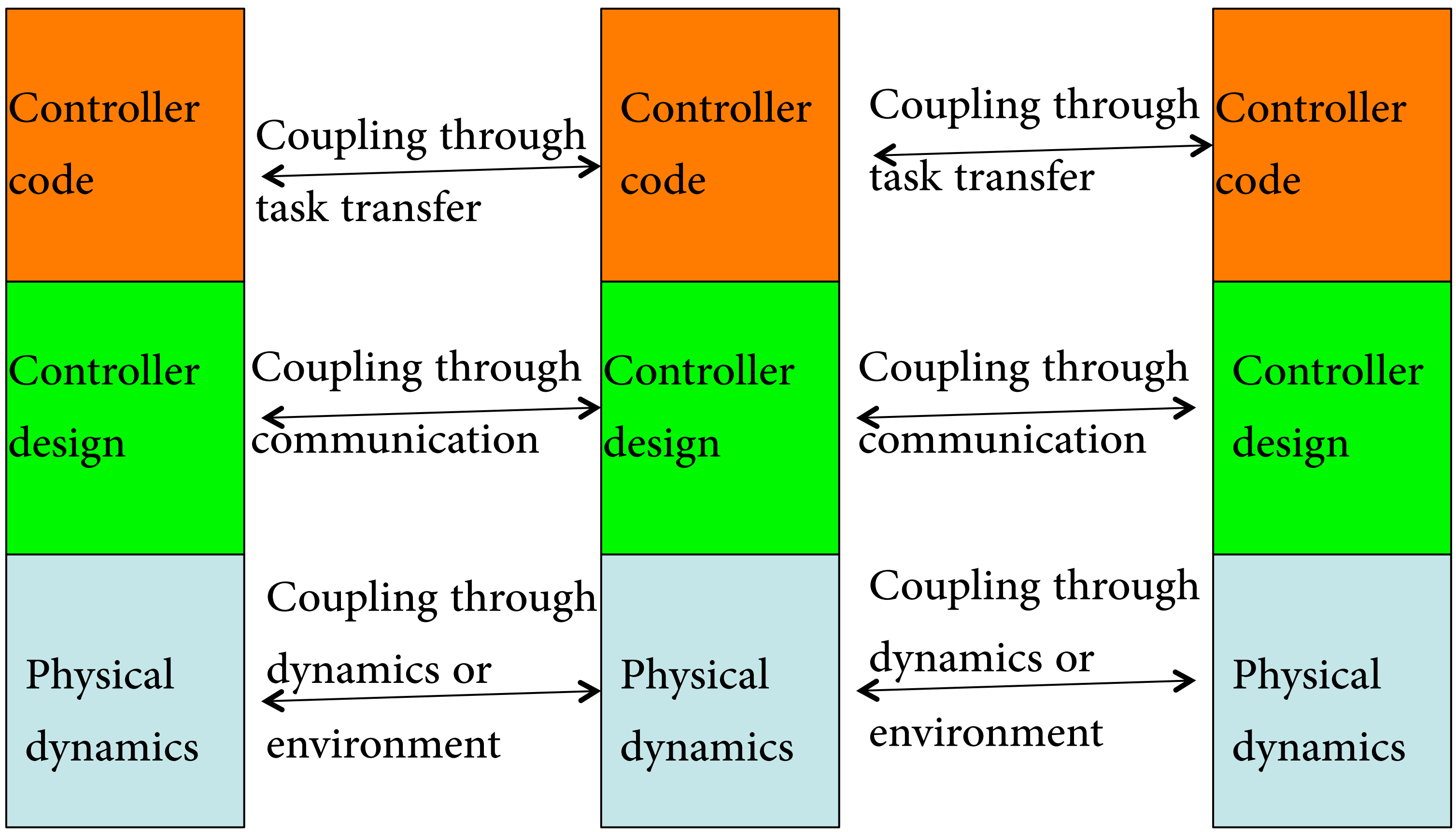
- Provable stability and performance guarantees

## This Year

- Distributed systems
- Event-triggered priority among control inputs

# Basic Idea


Robot 1
Robot 2
Robot 3
Controller 1
Controller 2
Controller 3


Controller code
Coupling through task transfer
Controller code
Coupling through task transfer
Controller code
Controller design
Coupling through communication
Controller design
Coupling through communication
Controller design
Physical dynamics
Coupling through dynamics or environment
Physical dynamics
Coupling through dynamics or environment
Physical dynamics

# How to transfer tasks?

- Push mechanism v/s Pull mechanism
- Need distributed policies, but global guarantees
- We study stochastic pull strategies

## A Stochastic Pull Algorithm

- We define a random sequence $\{h_i\}_{\mathbb{N}_0}$ (referred as strategy sequence) to indicate if agent $i$ helps agent $j$ or receives help from $j$ at time $k$, for all $i,\ j \in \{1, ..., N\}$ as:

$$h_i(k) \triangleq (i,j), \quad \text{if controller } i \text{ helps } j \text{ at time } k \text{ ,}$$

$$h_i(k) \triangleq (j,i), \quad \text{if controller } i \text{ receives help from } j \text{ at time } k \text{ .}$$

$$h_i(k) \triangleq 0, \quad \text{if controller } i \text{ neither provides nor receives help.}$$

- Assume these three events are mutually exclusive and exhaustive.
- Moreover this sequence is i.i.d. with defined probability distribution.
- Controller can calculate a buffer of inputs if extra time available.

# Model and Analysis

- The number of control inputs in the buffer of agent i follows a Markov chain.

- The agent then evolves as a Markovian jump system

$$x_i(k+1) = \begin{cases} f_i(x_i(k), \mathbf{0}_p), & \text{if } Z_i(k) = 0, \\ f_i(x_i(k), \kappa_i(x_i(k))), & \text{if } Z_i(k) \in \{1, 2, ..., \Lambda\}. \end{cases}$$

- Stability can be assessed using standard tools

$$\mathbf{E}\{V_i(x_i(k+1)) \,|\, x_i(k) = \chi_i, Z_i(k-1) = \bar{s}\} \leq \Omega_i(\bar{s}) V_i(\chi_i)$$
$$\forall \chi_i \in \mathbb{R}^n,\ k \in \mathbb{N}_0,\ i \in \{1, 2, ..., N\},\ \bar{s} \in \{0, 1, 2, ..., \Lambda\}$$

*where*

$$\Omega_i(\bar{s}) \triangleq q^i_{\bar{s}0}\alpha_i + \sum_{\tilde{s}\neq 0} q^i_{\bar{s}\tilde{s}}\rho_i.$$

- Coupled dynamics require a relaxation

# Optimizing the Pull Algorithm

- Can now optimize convergence rate with respect to the task transfer probabilities
  - Given $\alpha_i, \rho_i, p_l^i, i \in \{1, ..., N\}, l \in \{0, 1, ..., \Lambda\}$,

$$
\begin{aligned}
\text{maximize} \quad & \textstyle\sum_{i \in \{1,\ldots,N\}} U_i(1 + \Lambda - \sum_{\bar{s} \in \{0,\ldots,\Lambda\}} \Omega_i(\bar{s})) \\
\text{subject to} \quad & 0 < q^i_{\bar{s}\tilde{s}} < 1, \quad i \in \{1, ..., N\},\ (\tilde{s}, \bar{s}) \in \mathbb{S} \times \mathbb{S}, \\
& 0 < p_{ij} < 1, \quad i, j \in \{1, ..., N\}, \\
& \Omega_i(\bar{s}) < 1, \quad i \in \{1, ..., N\},\ \bar{s} \in \{0, 1, 2, ..., \Lambda\}, \\
& 0 < \textstyle\sum_{a \in N_b} p_{ab} < 1, \quad b \in N_r, \\
& 0 < \textstyle\sum_{c \in N_b} p_{bc} < 1, \quad b \in N_h, \\
& \textstyle\sum_{a \in N_b} p_{ab} + \sum_{c \in N_b} p_{bc} + p_w^b = 1, \quad b \in N_u,
\end{aligned}
\tag{7}
$$

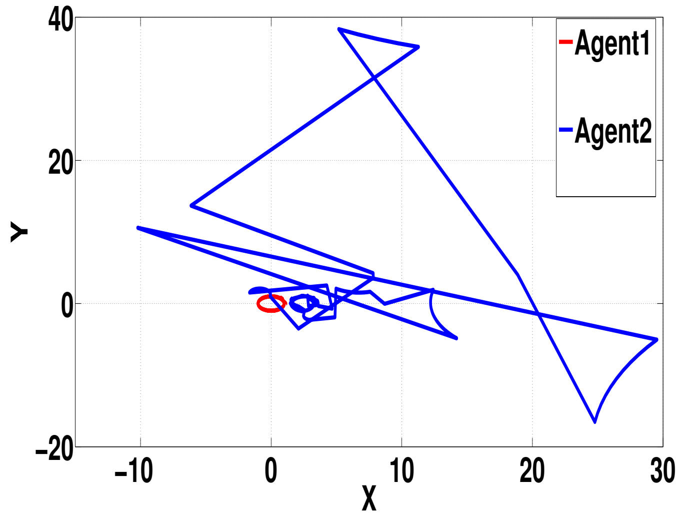
  where the utility function $U_i(\cdot)$ is a concave and increasing function, e.g., $\sqrt{(\cdot)}$ and $\log(\cdot)$.

- Can optimize in a distributed online fashion using dual decomposition with subgradient method

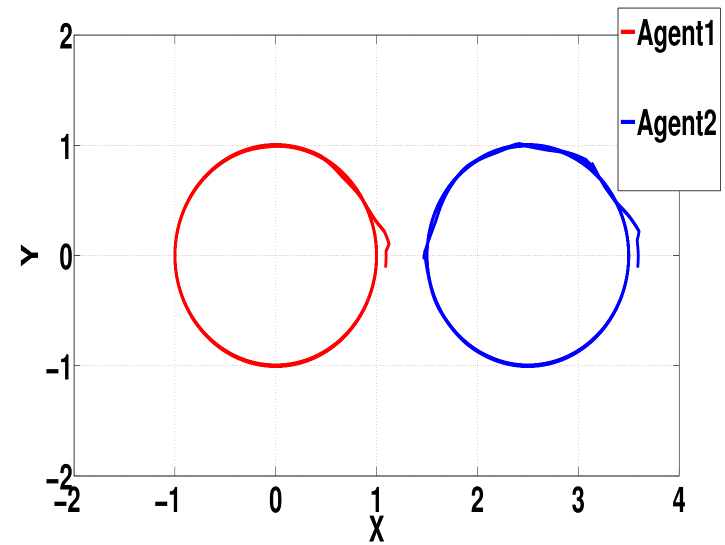
# Example

- TrueTime simulation for two agent trajectory following when agent 2's processor is overloaded

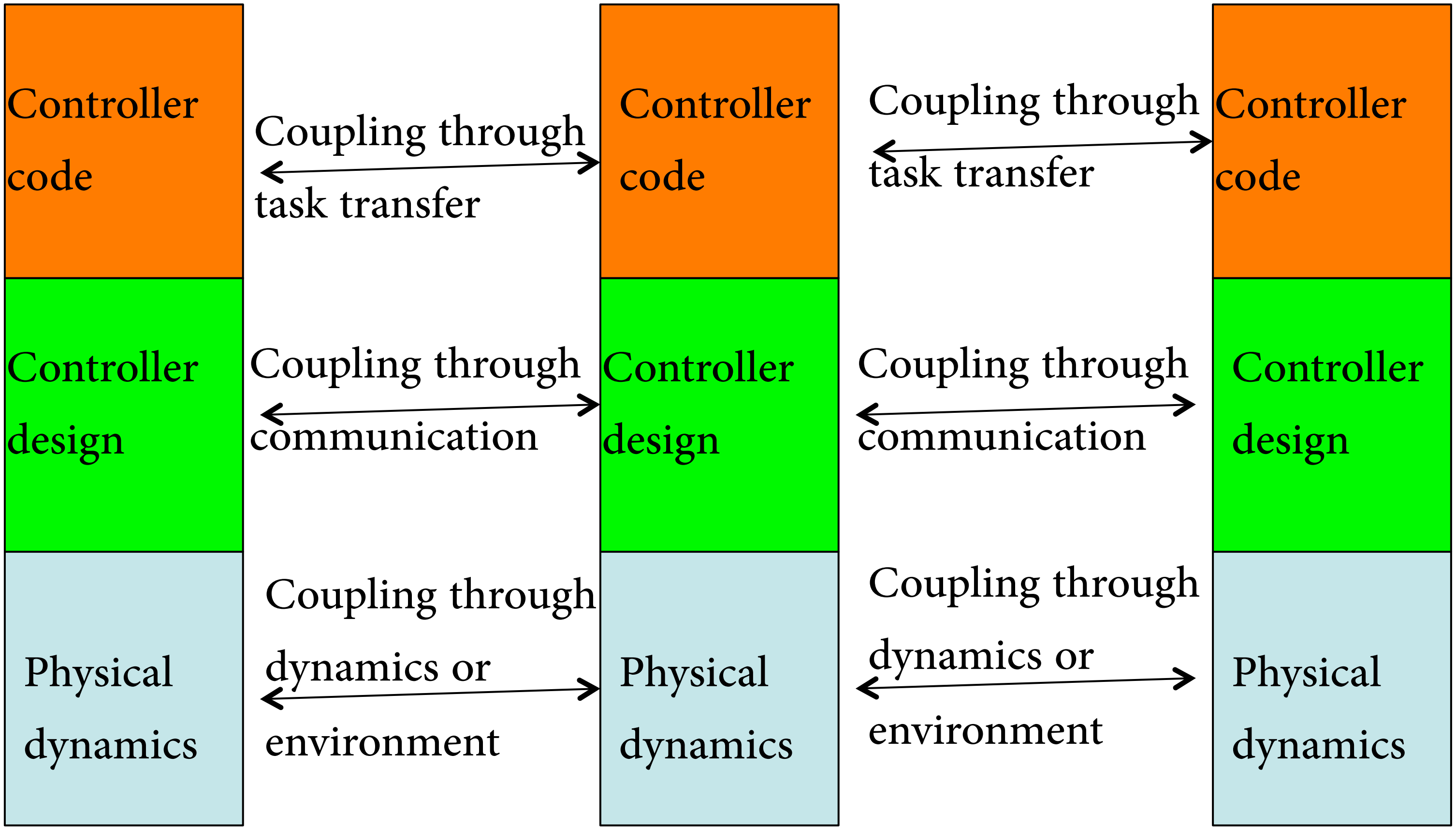
- Agent 1 does not help Agent 2 ($p_{12} = 0$).
- Agent 2 is not stable.




- Agent 1 helps Agent 2 ($p_{12} = 0.5$).

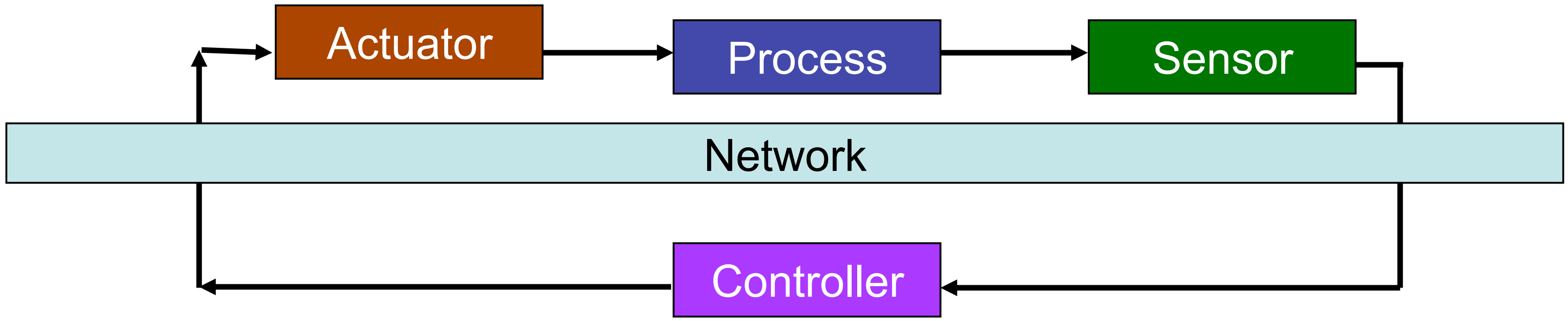


# Algorithms for Limited Processor Availability




- Middleware
- Timing and synchronization

# Passivity in the presence of uncertainty




- Packet loss (through TDMA protocols or network effects)
- Basic problem: system may not be passive in open loop

$$x(k+1) = \begin{cases} f(x(k), u(k)) & \text{if packet received} \\ f(x(k), 0) & \text{otherwise} \end{cases}$$

- Contributions:
  - Extended passivity definition
  - Conditions for retaining passivity
  - A more general theory of passivity of switched systems

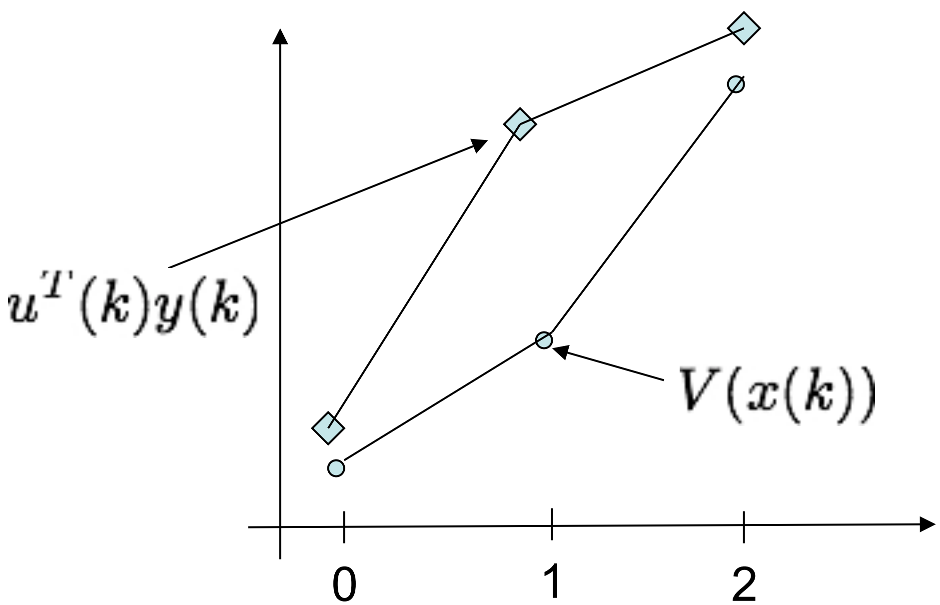
# Basic Idea

$$x(k+1) = f(x(k), u(k))$$
$$y(k) = h(x(k), u(k))$$

- Traditional definition does not allow open loop non passive system

$$V(x(k+1)) - V(x(k)) \le u^T(k)y(k)$$

- Extensions to switched systems (Zhao and Hill) assume every mode to be passive




- Our idea: require

$$V(x(k)) - V(x(0)) \le \sum_{j=0}^{k} u^T(j)y(j)$$

- Energy stored can increase temporarily

# With the New Definition

- Definition reduces to traditional definition for all modes passive
- Preserves nice properties of passivity (L-2 stability, feedback etc)

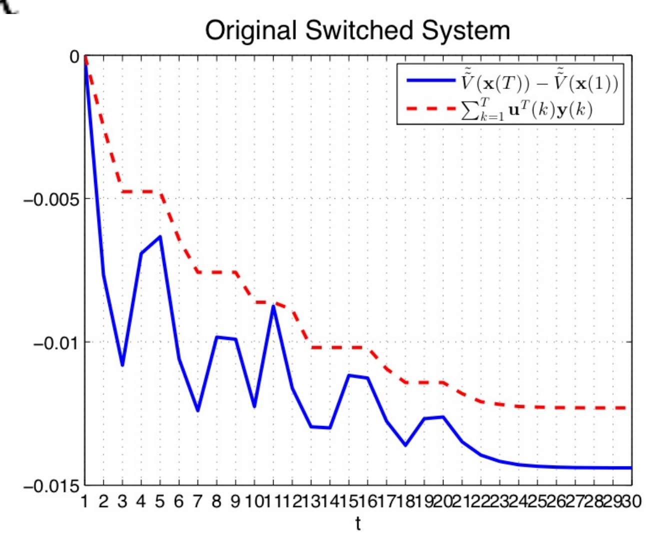
**Theorem:** Let

$$V(f(x(k),0)) \leq \chi V(x(k)$$
$$V(f(x(k),u(k)) \leq \sigma V(x(k)).$$

If the ratio of closed loop to open loop instances at any time $T$ satisfies

$$r(T) > \frac{(T-1)\ln\chi}{\ln\chi - T\ln}$$

then system is locally passive around origin.



## This Year

- Extended definition to multiple passive, feedback passive, and non-passive modes

- Modeled network as a stochastic system, possibly with memory: need a theory of stochastic passivity

- Using stochastic switching among controllers, improving performance while remaining passive

# Generalized Feedback Passivity

- If modes can be passive, feedback passive, or non-passive, then the definition can be generalized
- Use only one storage function for all modes
- A switched nonlinear system is locally feedback passive if and only if its zero dynamics are locally passive

*Theorem 6.1:* Design the switching signal such that

$$\frac{K^{-}(0,T)}{K^{+}(0,T)} \geq \frac{\ln L_2 - \ln L_0}{\ln L_0 - \ln L_1}, \tag{41}$$

where $L_0 \in (L_1, 1)$, $K^{-}(0,T)$ is the total activation time of the passive and feedback passive modes, and $K^{+}(0,T)$ is the total activation time of the non-feedback passive modes during time interval $[0,T)$, $\forall T \in \{0\} \bigcup \mathbb{Z}_+$. The zero dynamics (12) are passive under the switching signal (41).

- Offline vs online switching

# Passivity of Markovian Jump Nonlinear Systems

- For the system

$$\mathbf{x}(k+1) = f_{\boldsymbol{\sigma}(k)}(\mathbf{x}(k), u(k)),$$

$$\mathbf{y}(k) = h_{\boldsymbol{\sigma}(k)}(\mathbf{x}(k), u(k)),$$

define passivity as existence of suitable functions such that

$$\mathbb{E}[V(\mathbf{x}(k+1), \boldsymbol{\sigma}(k+1))|x(k), \sigma(k)] - V(x(k), \sigma(k)) \leq y^T(k)u(k).$$

- Passivity indices can be defined similarly

$$\mathbb{E}[V(\mathbf{x}(k+1), \boldsymbol{\sigma}(k+1))|x(k), \sigma(k)] - V(x(k), \sigma(k))$$

$$\leq (1 + \rho_{\sigma(k)}\nu_{\sigma(k)})y^T(k)u(k) - \rho_{\sigma(k)}y^T(k)y(k) - \nu_{\sigma(k)}u^T(k)u(k).$$

# Developments with this definition

- Consistent with usual definitions for non-stochastic systems
- Can develop usual stability and interconnection results

*Theorem 1:* If the discrete-time Markovian jump nonlinear system (1) is locally passive, then it is locally Lyapunov stable in probability.

*Theorem 3:* If system (1) is locally state strictly passive with $\underline{\alpha}_i||x(k)||^2 \leq V(x(k), i) \leq \bar{\alpha}_i||x(k)||^2$ and $S(x(k), i) \geq c_i||x(k)||^2$ for all $x(k) \in \mathbb{D}$ and some $\underline{\alpha}_i > \bar{\alpha}_i > 0$, $c_i > 0$, then it is locally stochastically stable.

*Theorem 4:* Let system (1) be locally passive and zero-state detectable. Then, the output feedback control law $u = -\varphi(y)$ with $\varphi : \mathbb{R}^m \to \mathbb{R}^m$ being any first/third sector function (i.e., $y^T\varphi(y) > 0 \; \forall y \neq 0$ and $\varphi(0) = 0$), renders the equilibrium locally asymptotically stable in probability.

*Corollary 1:* If system (1) is IFP($\nu$) and OFP($\rho$) and $\rho > 0$, then it is input-output $L_2$ stable with $L_2$ gain $\gamma \leq \max(1/|\rho|, |\nu|)$.

*Theorem 2:* The parallel and feedback interconnections of two locally passive Markovian jump nonlinear systems (as shown in Fig. 1), with the stochastic processes $(\mathbf{x}_1(0), \boldsymbol{\sigma}_1)$ and $(\mathbf{x}_2(0), \boldsymbol{\sigma}_2)$ being mutually independent, remain locally passive.

## Feedback Passivation

- Passifying using feedback control requires minimum phase for linear systems

- For MJLS, allow control to be mode specific

$$u(k) = E_{\boldsymbol{\sigma}(k)}\mathbf{x}(k) + F_{\boldsymbol{\sigma}(k)}v(k),$$

- Main result: feedback passivation requires the system to be strictly minimum phase (i.e. the zero dynamics should be mean squared stable; not merely in probability)

# Conditions involving Stochastic Conic Systems

- Can also generalize to conic systems

*Definition 10:* An operator $H(\boldsymbol{\omega}, u)$ with $\boldsymbol{\omega}$ representing a zero-mean stochastic process with mutually independent random variables $\boldsymbol{\omega}(k)$ and $\mathbb{E}\,||\boldsymbol{\omega}||^2 < \infty$, is *stochastically* interior conic with respect to $f(\boldsymbol{\omega})$ if there exist real constants $r \geq 0$ and $c$ such that for $\forall u \in \mathbb{U}$, the inequality

$$\mathbb{E}\,||H(\boldsymbol{\omega}, u) - cu||^2 \leq r^2||u||^2 \tag{33}$$

is satisfied. Similarly, $H(\boldsymbol{\omega}, u)$ is *stochastically* exterior conic if $\mathbb{E}\,||H(\boldsymbol{\omega}, u) - cu||^2 \geq r^2||u||^2$.

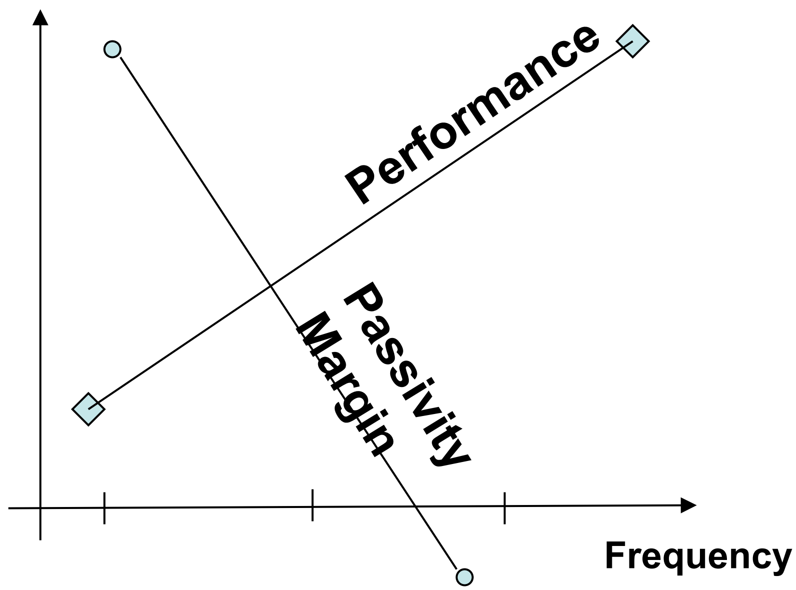
- Probabilistic interpretation

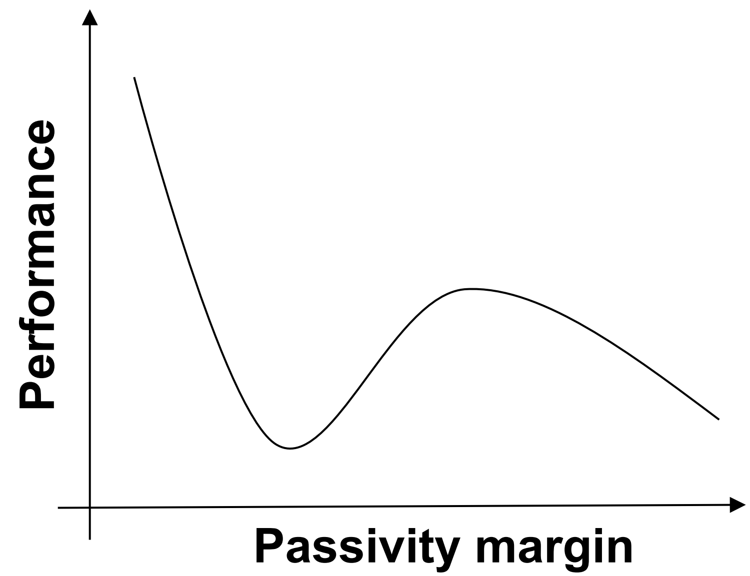
$$\Pr(||H(\boldsymbol{\omega}, u) - cu|| \geq nr||u||)) \leq \frac{\mathbb{E}\,||H(\boldsymbol{\omega}, u) - cu||^2}{n^2r^2||u||^2} \leq \frac{1}{n^2}.$$

- Consistent with the passivity indices definition proposed earlier

# Controller Switching for Performance

- Finding a controller that is passifying while optimizing a performance criterion is known to be hard
- We bypass this problem by switching between two controllers: a passifying controller and an optimizing controller
- The frequency of usage of the two controllers is the design parameter



## Illustration

$$x(k+1) = \begin{bmatrix} 1 & -0.5 \\ -0.1 & 0.6 \end{bmatrix} x(k) + \begin{bmatrix} 0.5 \\ 0 \end{bmatrix} u(k),$$

$$y(k) = \begin{bmatrix} 0.5 & -3 \end{bmatrix} x(k) + 0.2u(k).$$

- LQG optimal controller:

$$u_1(k) = \begin{bmatrix} 0.8864 & -0.696 \end{bmatrix} x(k) + v(k).$$

- A passifying controller:

$$u_2(k) = \begin{bmatrix} 2.5252 & -15.3605 \end{bmatrix} x(k) + v(k).$$

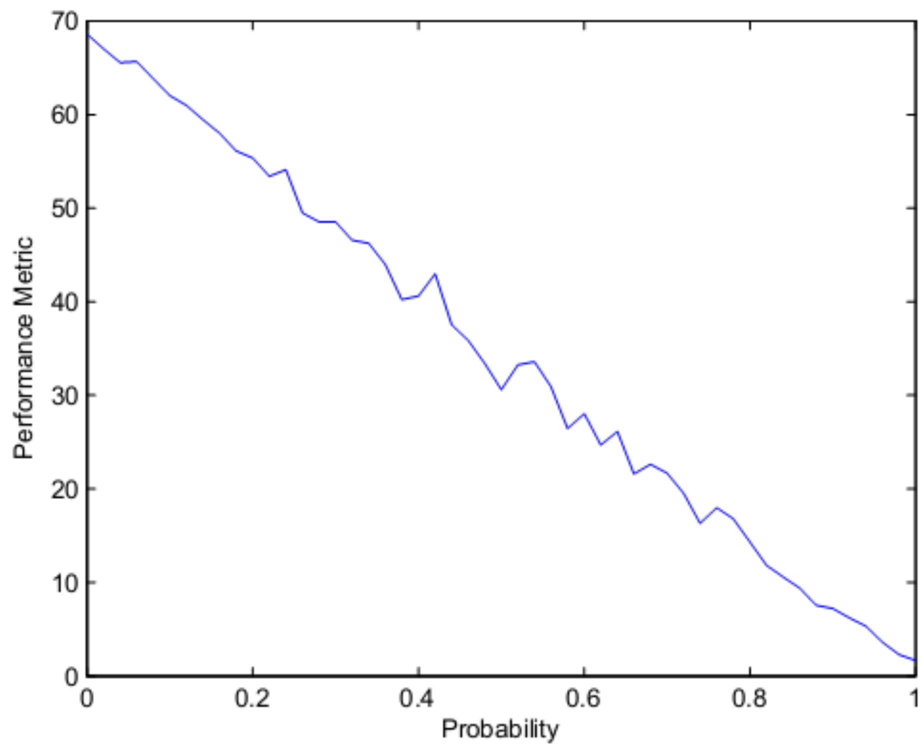
Suppose at every time, we apply $u_1$ with prob. $p$ and $u_2$ with prob. $1 - p$.

# Illustration

- Stochastic passivity requires small p (p<0.1)

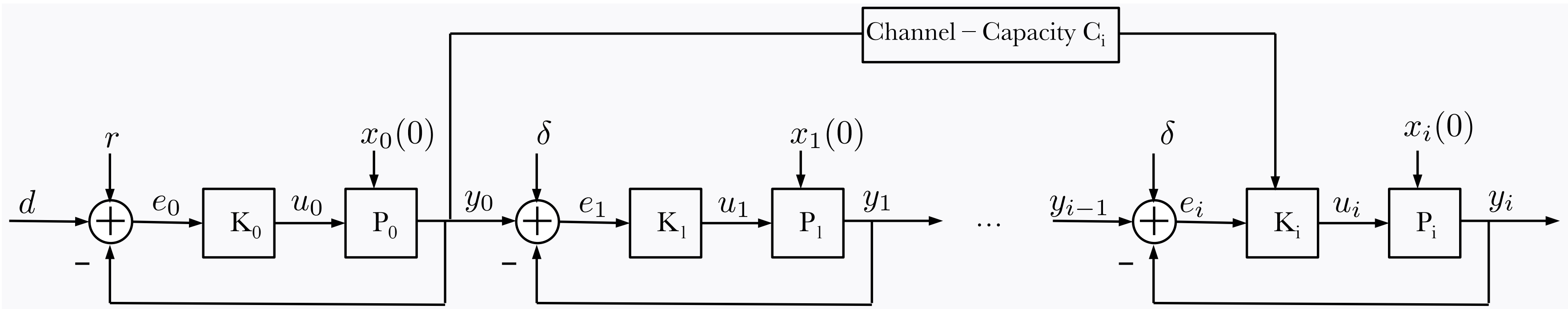
- 'Good' performance requires large p:

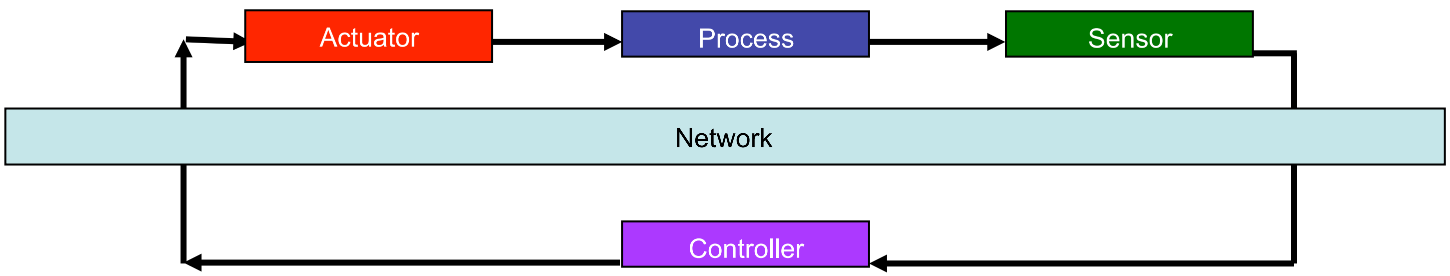
- Trade off?



vgupta2@nd.edu http://ee.nd.edu/faculty/vgupta/

# Next Steps




- Passivity of software
- Performance guarantees
- Robustness
- Implementations