Feedback and Control in Networked Systems

Richard M. Murray
Control & Dynamical Systems and Bioengineering
California Institute of Technology

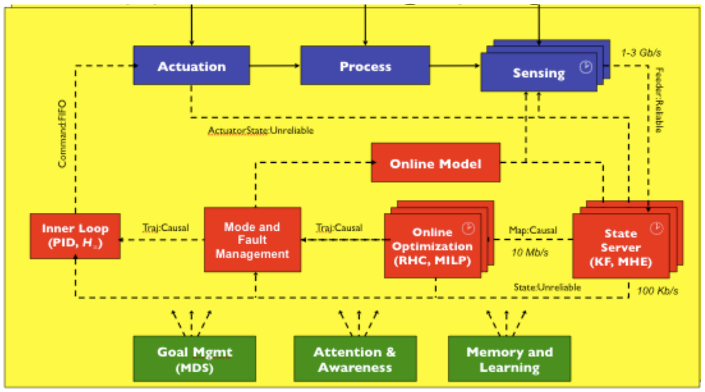
KITP Mini-Program: Network Architecture of Brain Structures and Functions
29 July 2011

Outline

I. Brief overview of what I work on
II. Feedback Control Theory: what is it good for... (tutorial: ~30 min)
III. (option 1) Bio-plausible architectures for insect-inspired control systems
IV. (option 2) Modern architectures for autonomous systems
V. (option 3) Early weekend

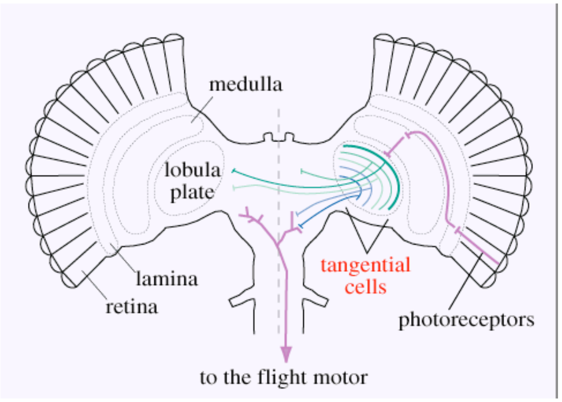
Control of Complex Systems

Alice

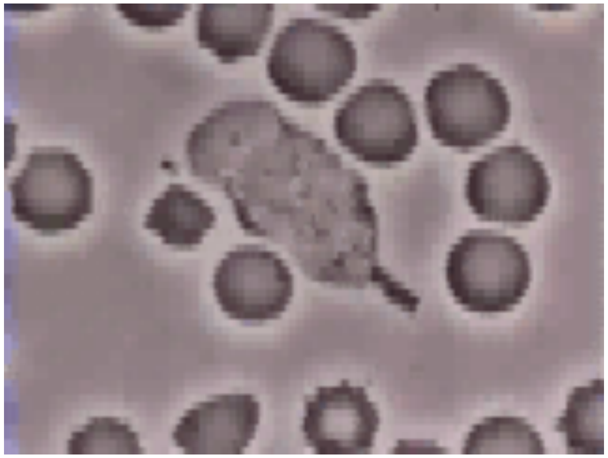


- Dominant challenges: (design for) verification and robustness

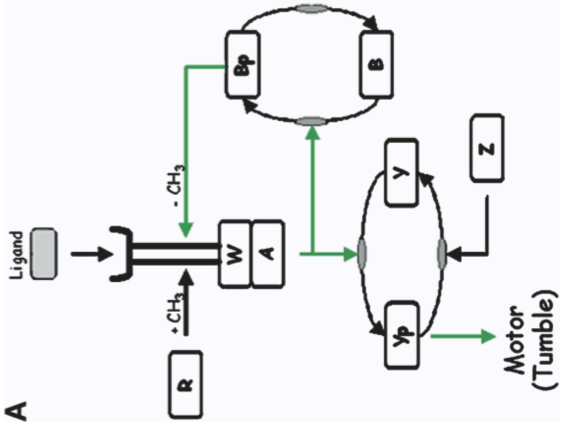
Drosophila



- Dominant challenges: decoding organization and architecture



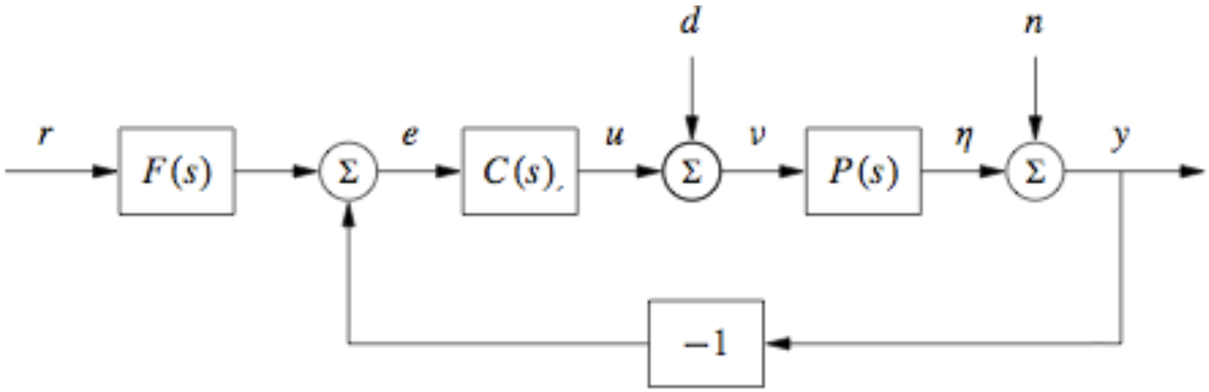
Neutrophil



- Dominant challenges: (lack of) modularity, stochastic program’g

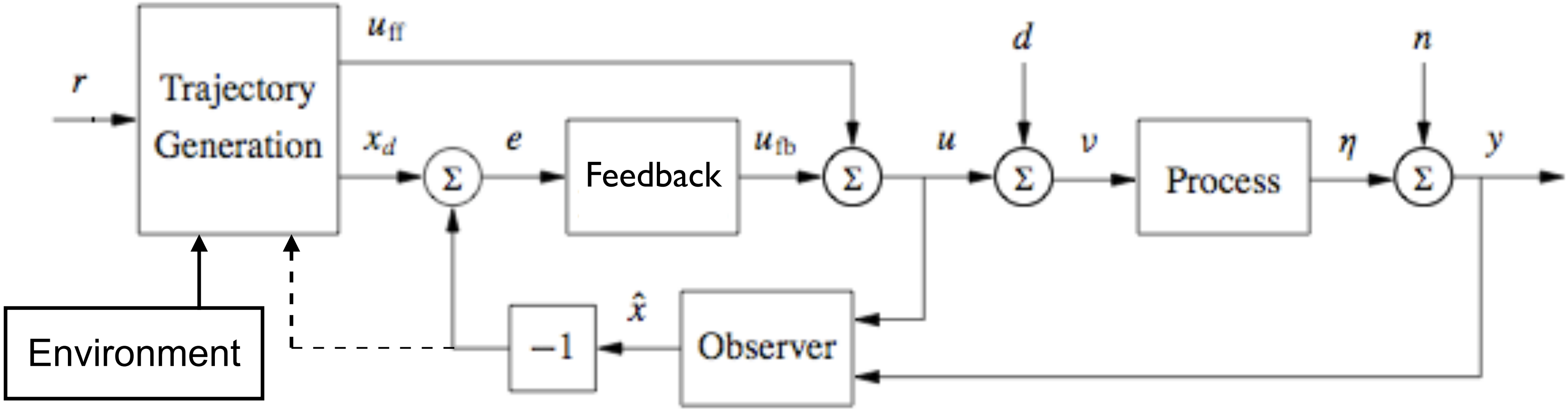
Design Patterns for (Engineered) Control Systems

Reactive compensation



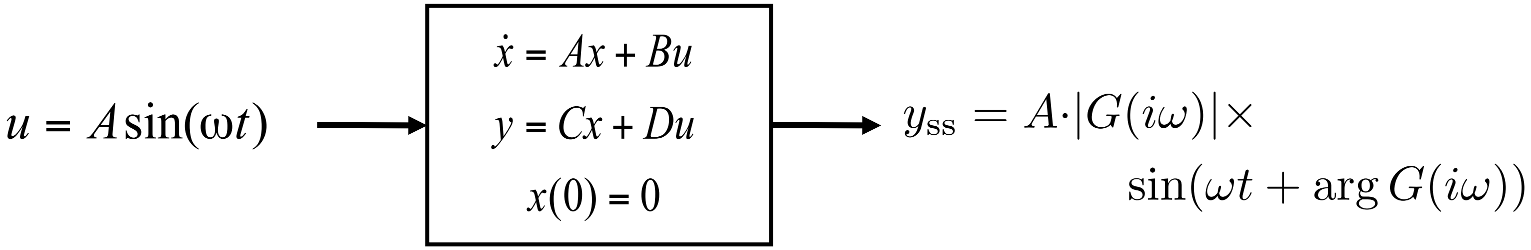
- Reference input shaping
- Feedback on output error
- Compensator dynamics shape closed loop response
- *Uncertainty* in process dynamics + external disturbances and noise
- Goals: stability, performance (tracking), robustness

Predictive compensation

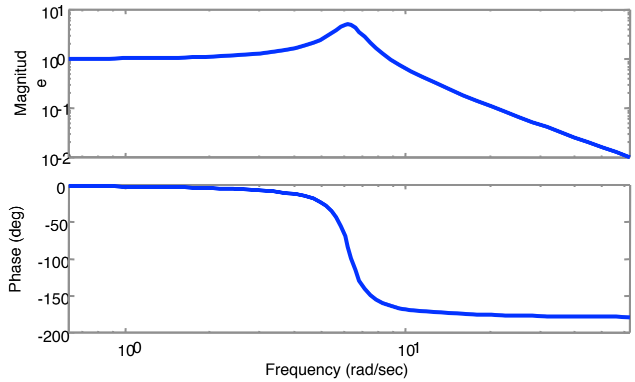


- Explicit computation of trajectories given a model of the process and environment

Frequency Response, Transfer Functions, Block Diagrams

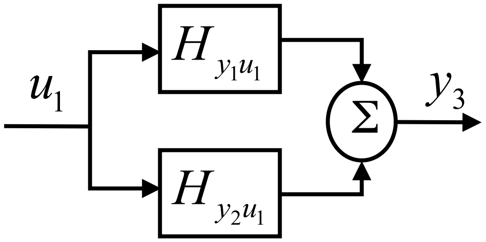
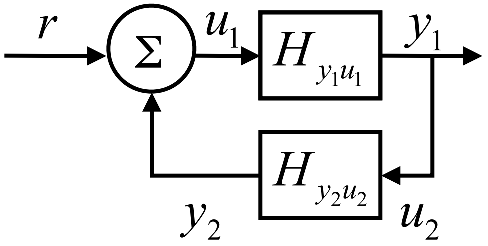


$$u = A\sin(\omega t) \longrightarrow \boxed{\begin{array}{c}\dot x = Ax + Bu \\ y = Cx + Du \\ x(0) = 0\end{array}} \longrightarrow y_{\rm ss} = A\cdot|G(i\omega)|\times \sin(\omega t + \arg G(i\omega))$$



$$G(s) = C(sI - A)^{-1}B + D$$

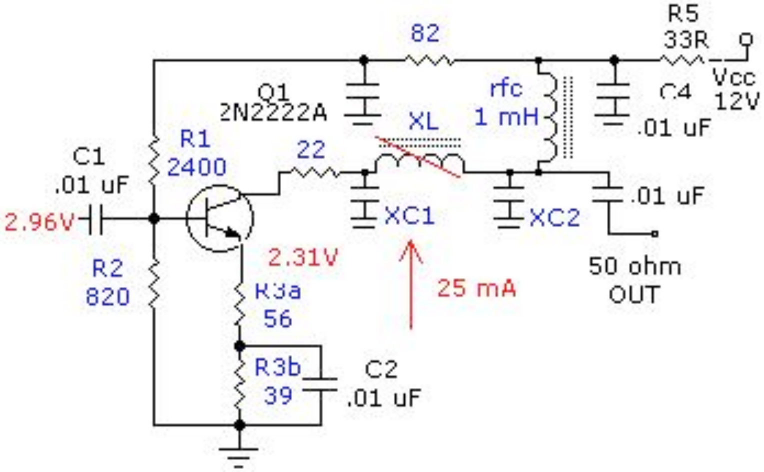
$$G_{y_2u_1} = G_{y_2u_2}G_{y_1u_1} = \frac{n_1n_2}{d_1d_2}$$

Parallel		$H_{y_3u_1} = H_{y_2u_1} + H_{y_1u_1} = \dfrac{n_1d_2 + n_2d_1}{d_1d_2}$
Feedback		$H_{y_1r} = \dfrac{H_{y_1u_1}}{1 + H_{y_1u_1}H_{y_2u_2}} = \dfrac{n_1d_2}{n_1n_2 + d_1d_2}$

Two Main Principles of Control

Robustness to Uncertainty thru Feedback

- Feedback allows high performance in the presence of uncertainty
- Example: repeatable performance of amplifiers with 5X component variation
- Key idea: accurate *sensing* to compare actual to desired, correction through *computation* and *actuation*



Design of Dynamics through Feedback

- Feedback allows the dynamics of a system to be modified
- Example: stability augmentation for highly agile, unstable aircraft
- Key idea: interconnection gives *closed loop* that modifies natural behavior

Control involves (computable) *tradeoffs* between robustness and performance

Control Tools: 1940-2000

Modeling

- Input/output representations for subsystems + interconnection rules
- System identification theory and algorithms
- Theory and algorithms for reduced order modeling + model reduction

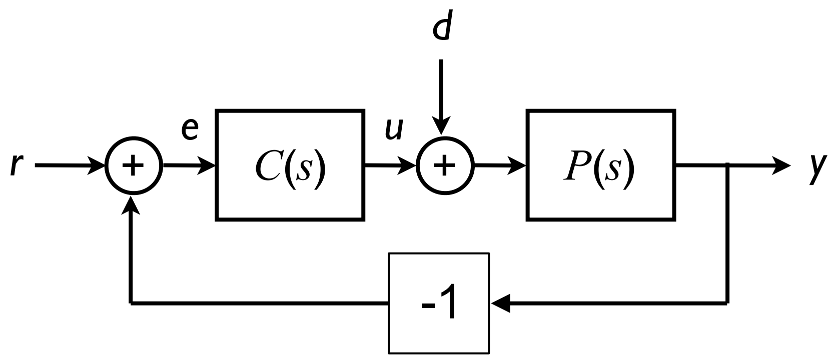
Analysis

- Stability of feedback systems, including robustness "margins"
- Performance of input/output systems (disturbance rejection, robustness)

Synthesis

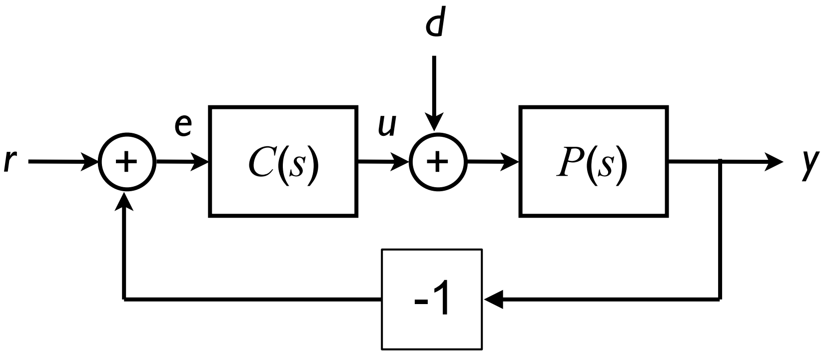
- Constructive tools for design of feedback systems
- Constructive tools for signal processing and estimation

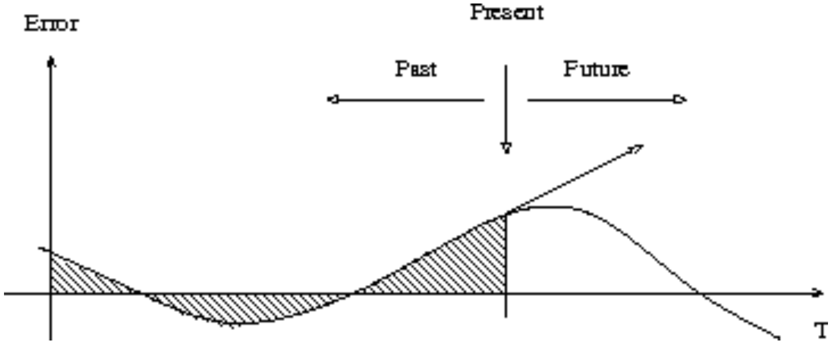
Basic feedback loop



- Plant, *P* = process being regulated
- Reference, *r* = external input (often encodes the desired setpoint)
- Disturbances, *d* = external environment
- Error, *e* = reference - actual
- Input, *u* = actuation command
- Feedback, *C* = closed loop correction
- Uncertainty: plant dynamics, sensor noise, environmental disturbances

Canonical Feedback Example: PID Control





$$\dot{x} = Ax + Bu$$
$$y = Cx + Du$$

$$u(t) = ke(t) + k_i \int_0^t e(\tau)\,d\tau + k_d \frac{de(t)}{dt}$$

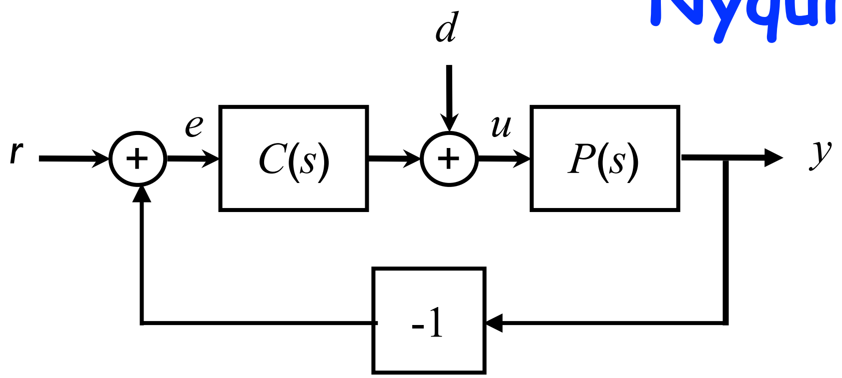
Three term controller

- Present: feedback proportional to current error
- Past: feedback proportional to *integral* of past error
  - Insures that error eventual goes to 0
  - Automatically adjusts setpoint of input
- Future: derivative of the error
  - *Anticipate* where we are going

PID design

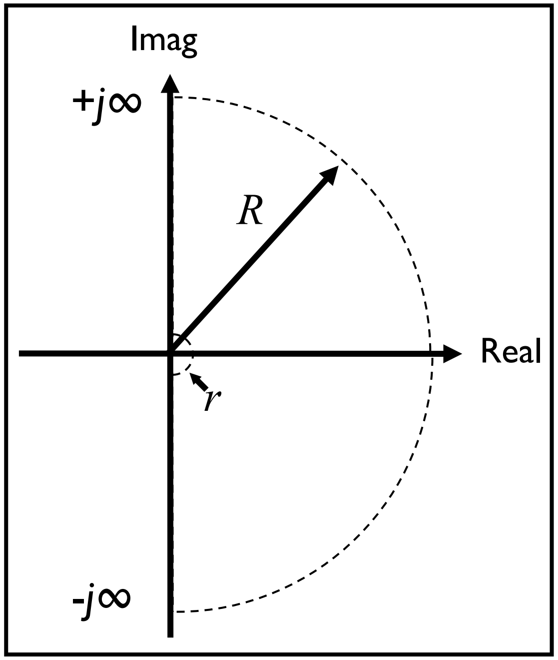
- Choose *gains* k, k_i, k_d to obtain the desired behavior
- *Stability*: solutions of the closed loop dynamics should converge to eq pt
- *Performance*: output of system, y, should track reference
- *Robustness*: stability & performance properties should hold in face of disturbances and plant uncertainty

Nyquist Criterion



Determine stability from (open) loop transfer function, $L(s) = P(s)C(s)$.

- Use "principle of the argument" from complex variable theory (see reading)



- Nyquist "D" contour
- Take limit as $r \to 0$, $R \to \infty$
- Trace from $-\infty$ to $+\infty$ along imaginary axis

Thm (Nyquist). Consider the Nyquist plot for loop transfer function $L(s)$. Let

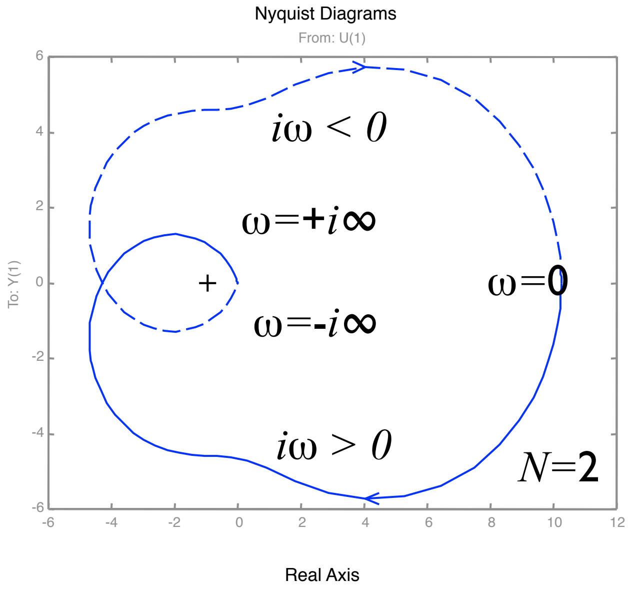
P # RHP poles of $L(s)$

N # clockwise encirclements of -1

Z # RHP zeros of $1 + L(s)$

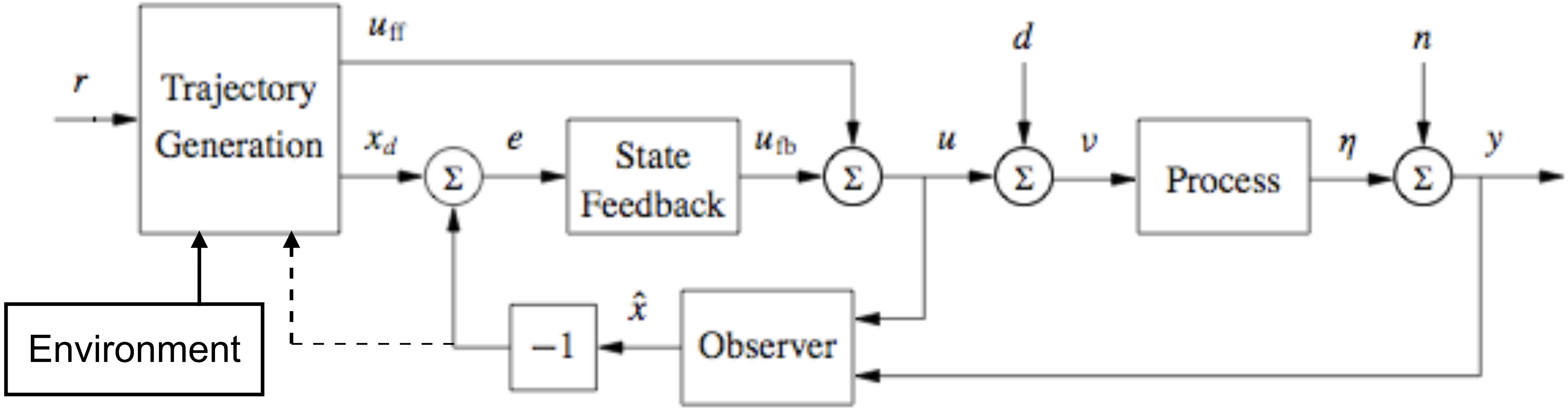
Then

$$Z = N + P$$



- Trace frequency response for $L(s)$ along the Nyquist "D" contour
- Count net # of clockwise encirclements of the -1 point

Feedforward and Feedback

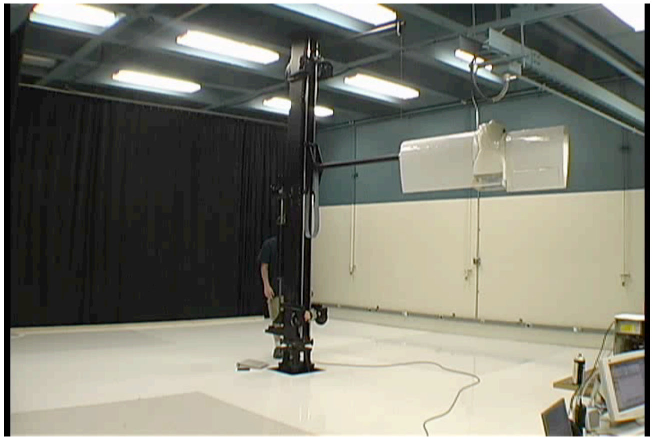


Benefits of feedforward compensation

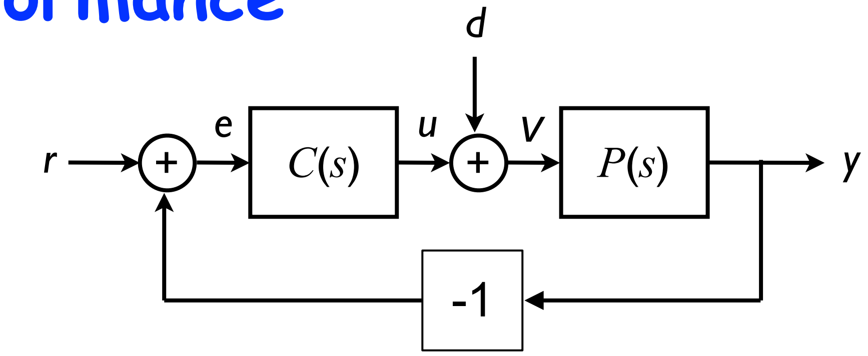
- Allows online generation of trajectories based on current situation/environment
- Optimization-based approaches can handle constraints, tradeoffs, uncertainty
- Trajectories can be pre-stored and used when certain conditions are met

Replanning using receding horizon

- Idea: regenerate trajectory based on new states, environment, constraints, etc
- Provides "outer loop" feedback at slower timescale
- Stability results available



Limits of Performance



Q: How well can you reject a disturbance?

- Would like *v* to be as small as possible
- Assume that we have signals *v*(*t*), *d*(*t*) that satisfy the loop dynamics
- Take Fourier transforms *V*(*ω*), *D*(*ω*)
- *Sensitivity function*: *S*(*ω*) = *V*(*ω*)/*D*(*ω*); want *S*(*ω*) « 1 for good performance

Thm (Bode) Under appropriate conditions (causality, non-passivity)

$$\int_0^\infty \log|S(\omega)|d\omega \geq 0$$

Consequences: achievable performance is bounded

- Better tracking in some frequency band ⇒ other bands get worse
- For linear systems, formula is known as the *Bode integral formula* (get equality)
- “Passive” (positive real) systems can beat this bound

Extensions

- Discrete time nonlinear systems: similar formula holds (Doyle)
- Incorporate Shannon limits for communication of disturbances (Martins et al)

Example: Magnetic Levitation System

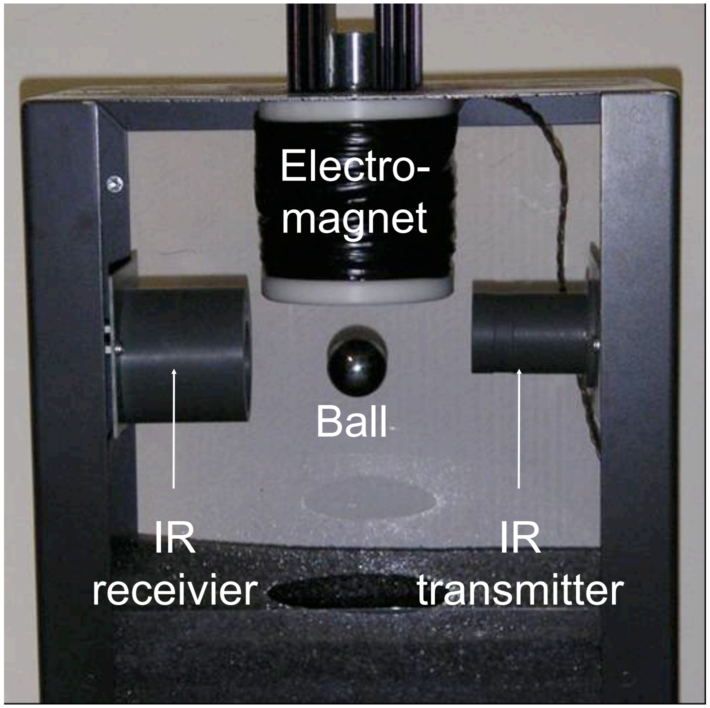
Nominal design gives low perf

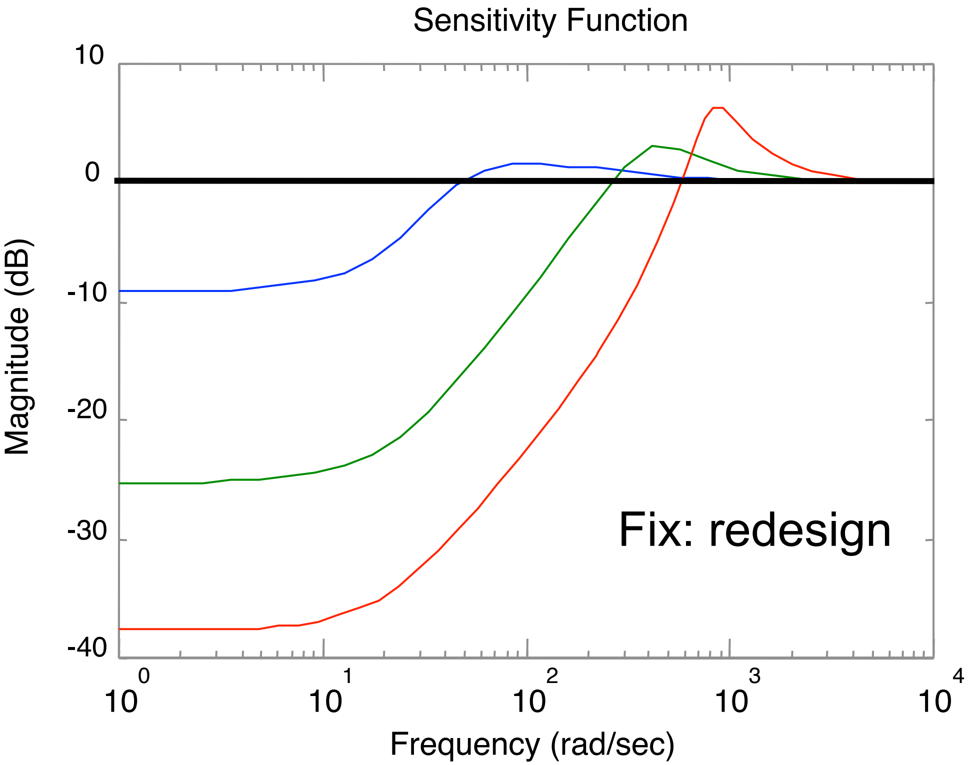
- Not enough gain at low frequency
- Try to adjust overall gain to improve low frequency response
- Works well at moderate gain, but notice waterbed effect

Bode integral limits improvement

$$\int_0^\infty \log |S(j\omega)| d\omega = \pi r$$

- Must increase sensitivity at some point





Fix: redesign

Stability in the Presence of Uncertainty

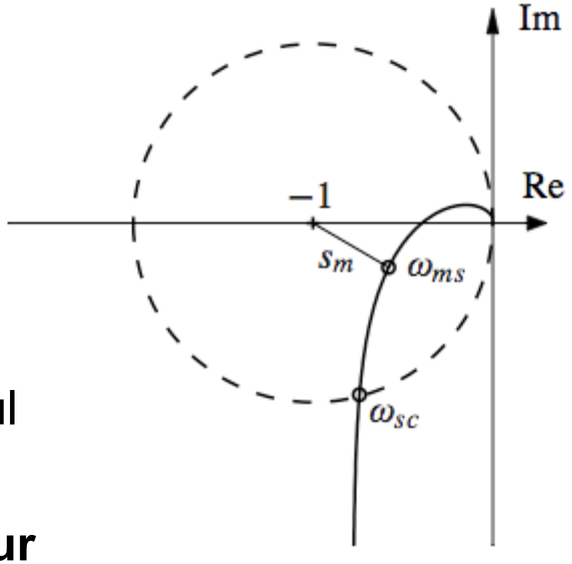
Characterize stability in terms of stability margin s_m

- Stability margin = distance on Nyquist plot to -1 point
- Stability margin = $1/M_s$ (M_s = maximum sensitivity)

$$M_s = \max |S(i\omega)| = \max \left| \frac{1}{1+L} \right|,$$

$$s_m = \min |(-1) - L| = \min |1 + L| = 1/M_s$$

- For robustness analysis, stability margin is more useful than classical gain and phase margins

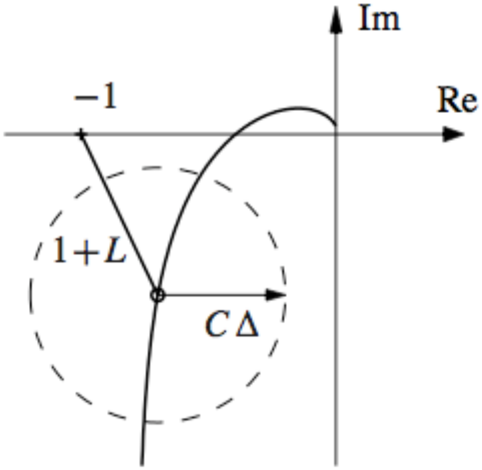


Robust stability: verify no new net encirclements occur

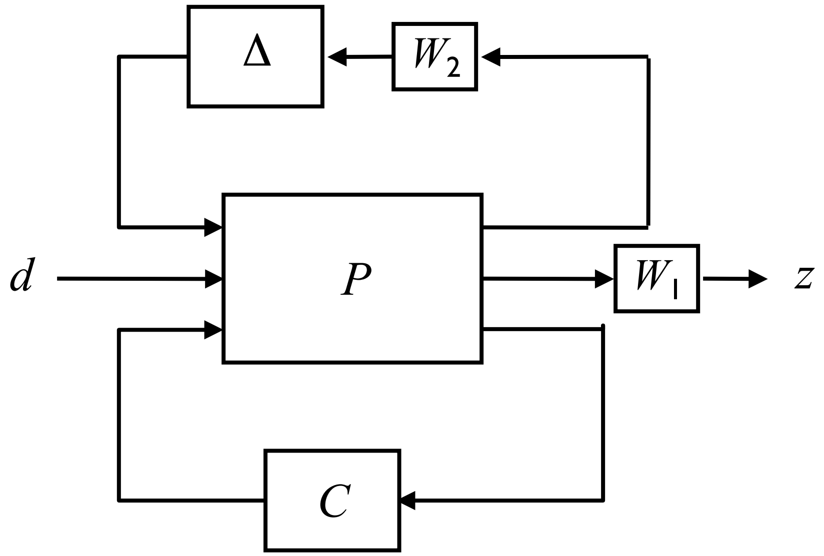
- Assume that nominal system is stable
- New loop transfer function: $\tilde{L} = (P + \Delta)C = L + C\Delta$
- No net encirclements as long as $|C\Delta| < |1 + L|$
- Can rewrite as bound on allowable perturbation

$$|\Delta| < \left| \frac{1 + PC}{C} \right| = \left| \frac{P}{T} \right| \quad \text{or} \quad |\delta| = \left| \frac{\Delta}{P} \right| < \frac{1}{|T|}$$

- If condition is satisfied, then sm will never cross to zero => no new net encirclements



Robust Control Theory



Model components as I/O operators

y(·) = P(u(·), d(·), w(·))

d	disturbance signal
z	output signal
Δ	uncertainty block
W_1	performance weight
W_2	uncertainty weight

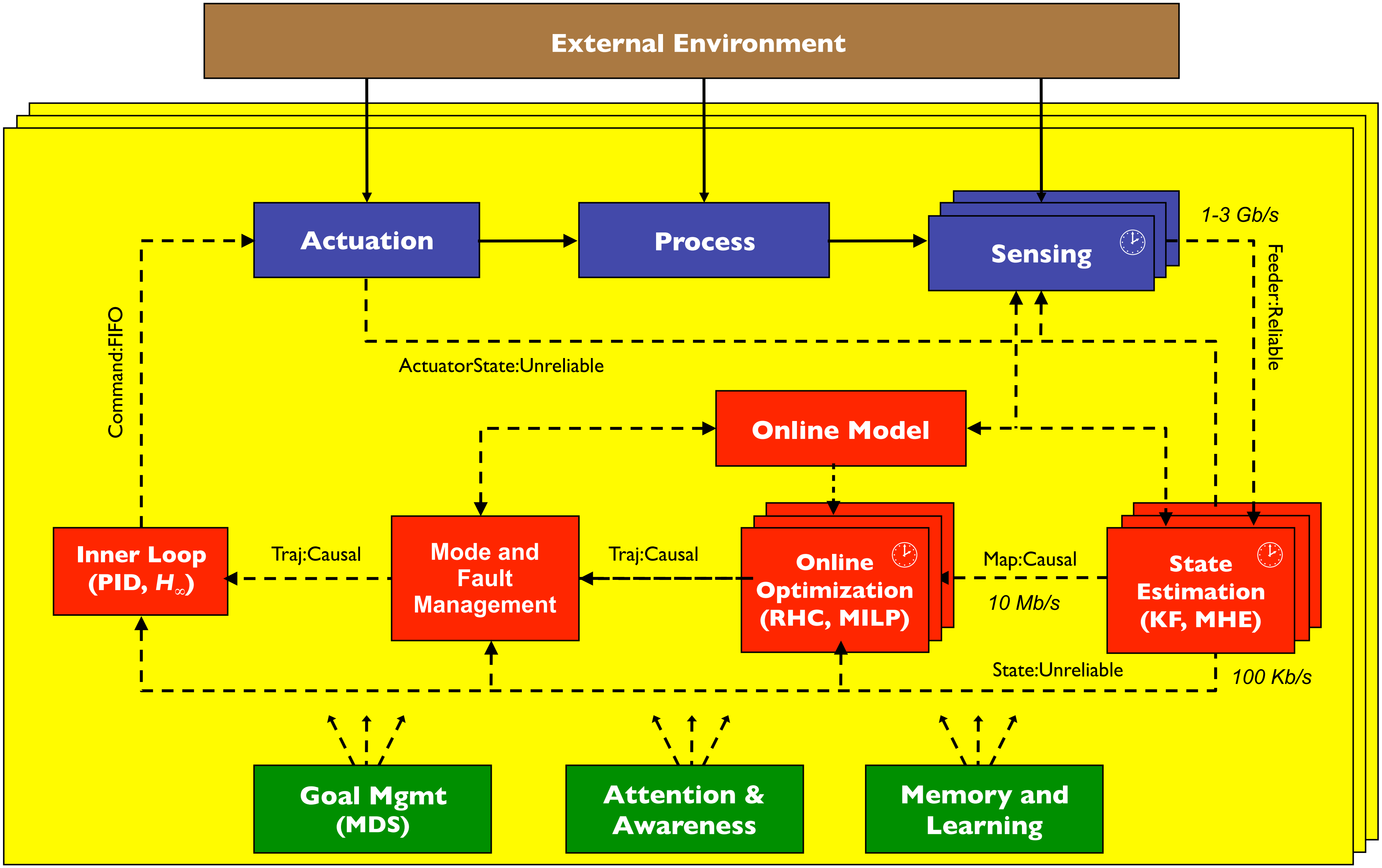
Goal: guaranteed performance in presence of uncertainty

$$\|z\|_2 \leq \gamma \, \|d\|_2 \quad \text{for all} \quad \|\Delta\| \leq 1$$

- Compare energy in disturbances to energy in outputs
- Use frequency weights to change performance/uncertainty descriptions
- “Can I get X level of performance even with Y level of uncertainty?”
- Generalizations to nonlinear systems (along trajectories) available [Tierno et al]

Networked Control Systems

(following P. R. Kumar)

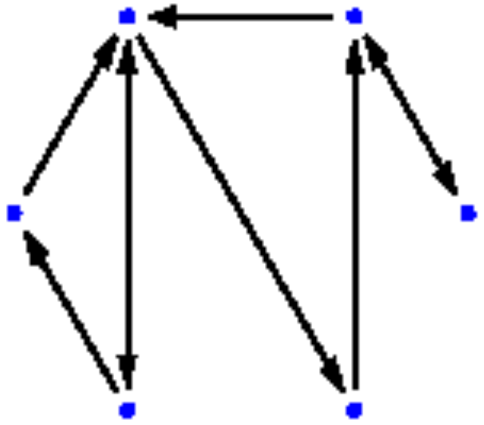


Fax and M
TAC, 2004

Stability of Interconnected Systems

Goal: maintain state/outputs relative to neighbors

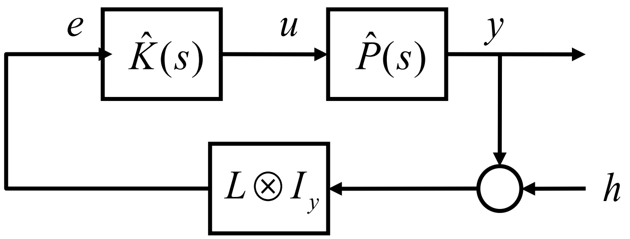
- “Neighbors” defined by a *directed* graph
- Assume only sensed data (system outputs) for now
- Assume identical dynamics, identical controllers



Example: hexagon formation

- Maintain fixed relative spacing between left & right neighbors

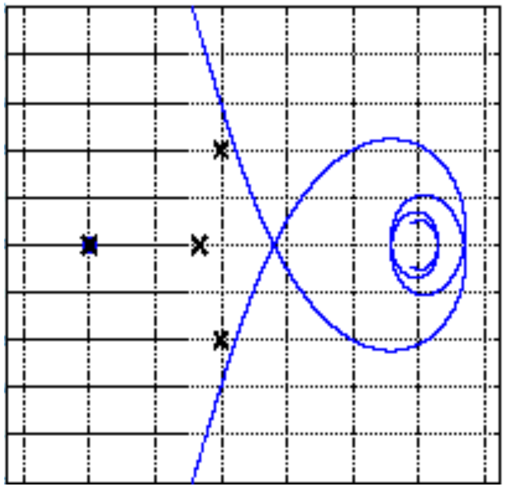
$$e_i = \sum_{j \in N_i} \underset{\substack{\uparrow \\ \text{weighting} \\ \text{factor}}}{w_j} (\underbrace{y_i - y_j}_{\substack{\text{relative} \\ \text{position}}} - \underset{\substack{\uparrow \\ \text{offset}}}{h_{ij}})$$



$$P(s) = \frac{e^{-s\tau}}{s^2}$$

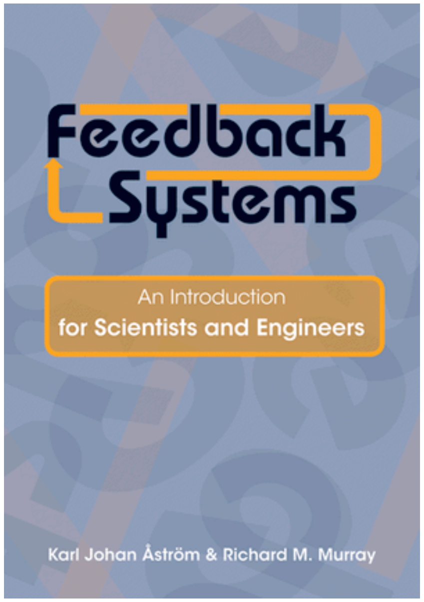
$$K(s) = K_d s + K_p$$

Theorem (Fax & M, 2004) *The closed loop system is (neutrally) stable iff the Nyquist plot of the open loop system does not encircle -1/λi(L), where λi(L) are the nonzero eigenvalues of L*



- Links topology of graph (via Laplacian) to dynamics of agents
- Can extend to discrete time, MIMO, nonlinear

Summary: Control Theory



Two main principles of (feedback) control theory

- Feedback is a tool to provide robustness to uncertainty
  - Uncertainty = noise, disturbances, unmodeled dynamics
  - Useful for modularity: consistent behavior of subsystems
- Feedback is a tool to design the dynamics of a system
  - Convert unstable systems to stable systems
  - Tune the performance of a system to meet specifications

Control theory is (primarily) a design-oriented theory

- Tools were developed to help engineers design control systems
- Analysis tools and fundamental limits can be used for natural systems

More information

- *Feedback Systems*: http://www.cds.caltech.edu/~murray/FBSwiki
- *Optimization-Based Control*: http://www.cds.caltech.edu/~murray/FBSwiki/OBC
- Additional references posted on the mini-program wiki

Next

- (option 1) Bio-plausible architectures for insect-inspired control systems
- (option 2) Modern architectures for autonomous systems
- (option 3) Early weekend

Bioplausible Approaches to Control using Distributed, Slow Computing

Richard M. Murray
Control & Dynamical Systems and Bioengineering
California Institute of Technology

Andrea Censi Sawyer Fuller Shuo Han Andrew Straw
Javad Lavaei Somayeh Sojoudi

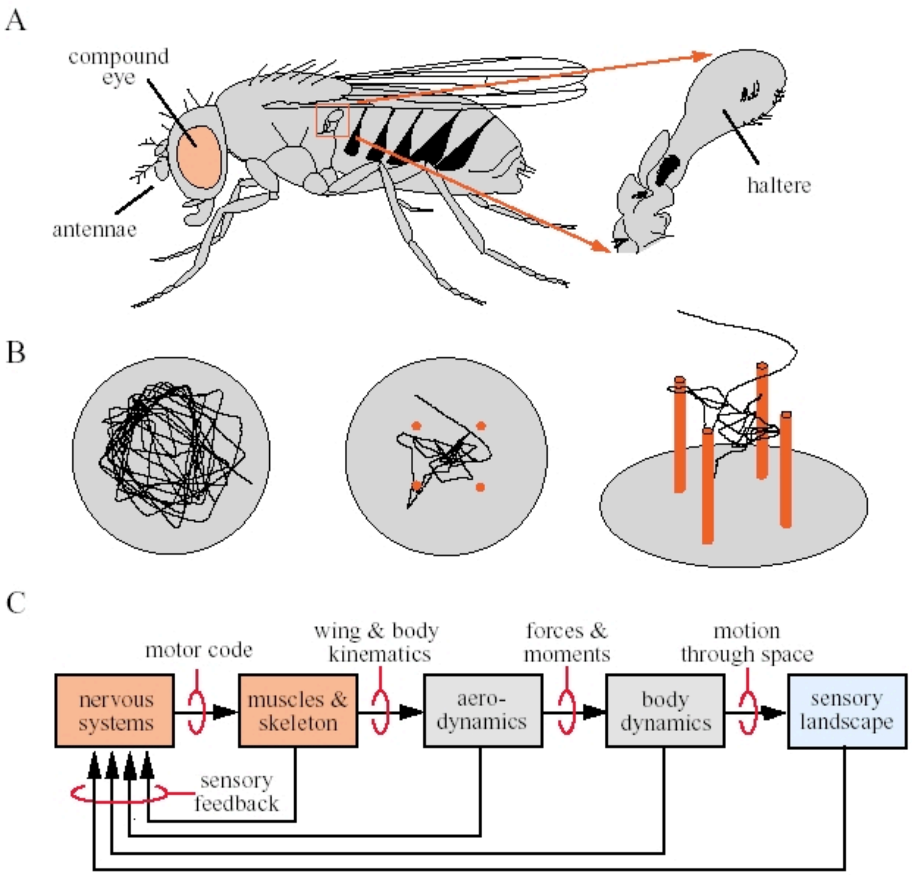
NSF CPS Workshop
10 August 2010

Outline

I. Motivation (and definition of bioplausible + slow computing)
II. Vision-based stabilization using bio-plausible control laws
III. Design of interconnections and time-delays for feedback control
IV. Future directions and next steps

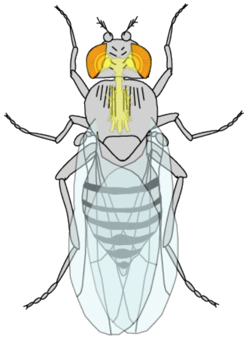
Humbert, Dickinson & M
CDC05

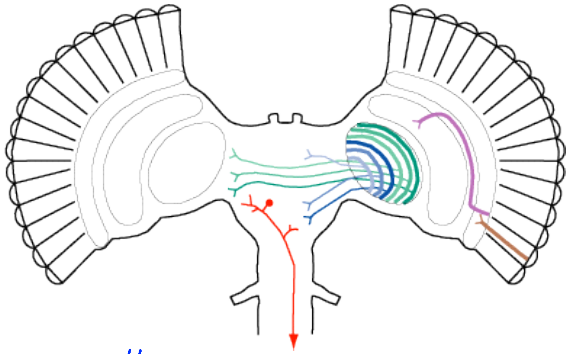
Networked Feedback Systems in Biology - Insects

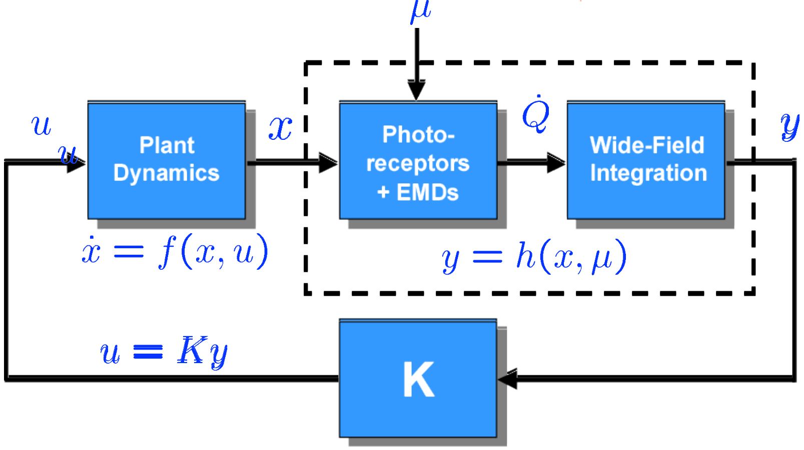


Different architecture than engineering

- Large collection of diverse sensors (many more than required)
- Very slow computation with lots of parallel pathways







μ

u

u

Plant Dynamics

x

Photo-receptors + EMDs

$\dot{Q}$

Wide-Field Integration

y

$\dot{x} = f(x, u)$

$y = h(x, \mu)$

$u = Ky$

K

Stabilization of a goal image/optical flow

- Compute forces and torques based on spatial integration of optimal flow

$$z_i(\boldsymbol{x}) \;=\; \langle \dot{Q}, F_i \rangle_w \;=\; \frac{1}{\pi}\int_0^{2\pi} \dot{Q}(\gamma, \boldsymbol{x}) \cdot F_i(\gamma)\, d\beta.$$

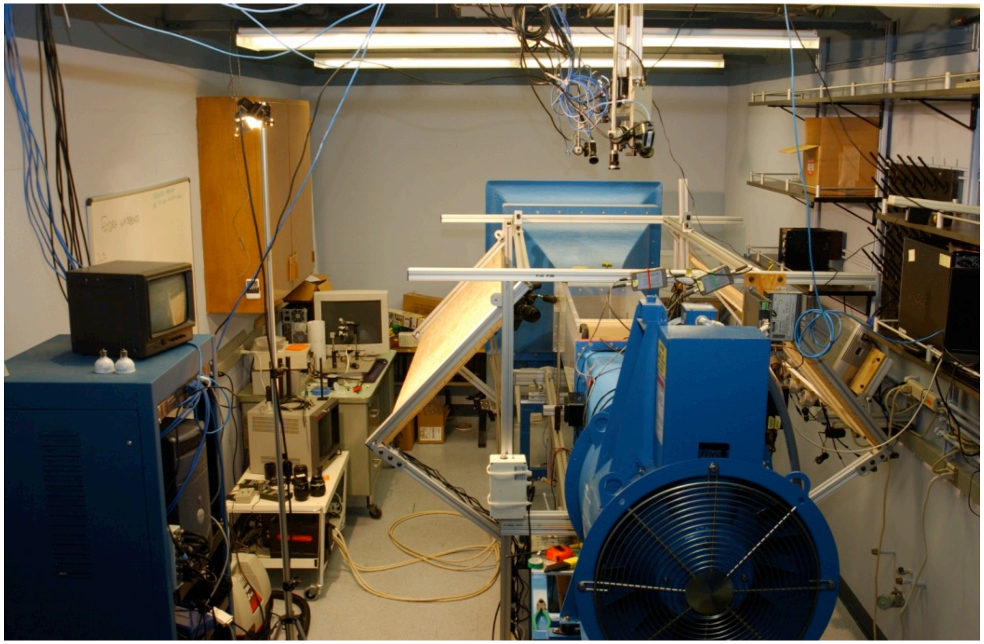
- Integration kernel F encodes controller

Fuller, Peek, Straw, Dickinson & M
(in progress)

Decoding the Architecture of Insect Flight Control

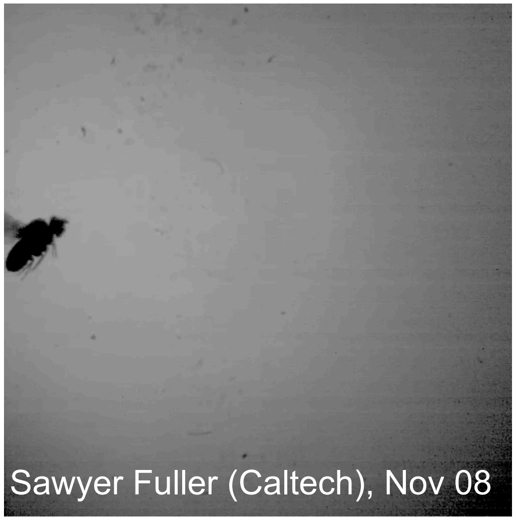
Fly wind tunnel/flight simulator

- Track flies in real-time, controlling wind + visual environ. (open/closed loop)
- High speed camera with location trigger
- Capture *lots* of data, then filter

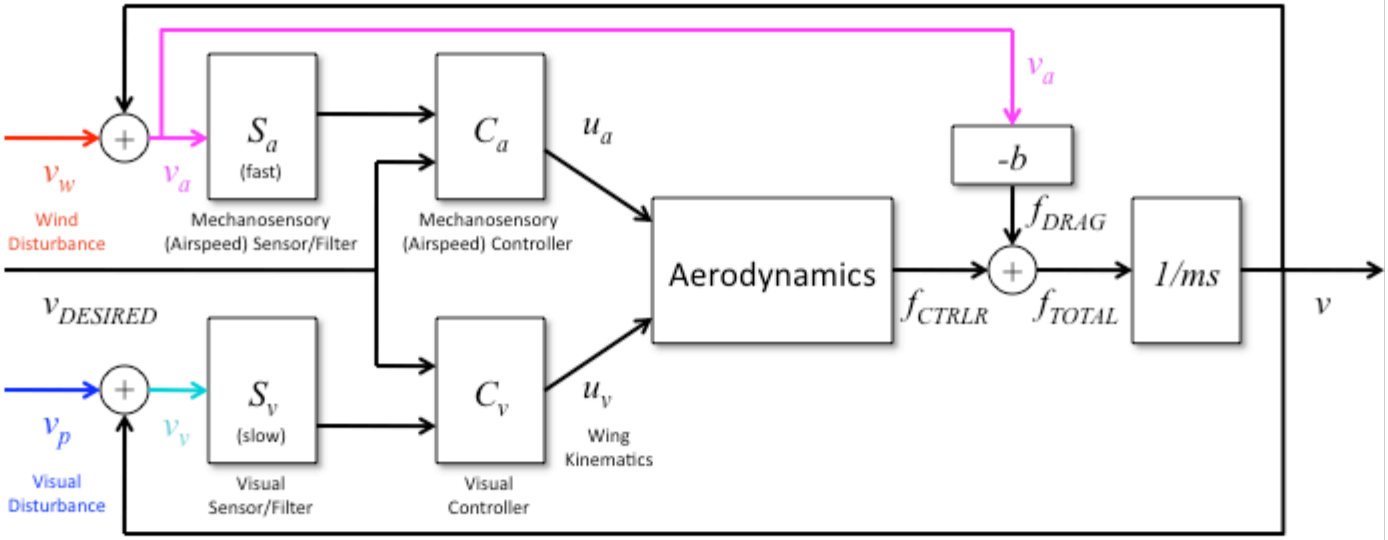


Current focus: gust response

- Sensing systems: vision, halteres (~gyros), aristae (antennae)
- How are different sensors integrated? Serial, parallel, inner/outer, etc...

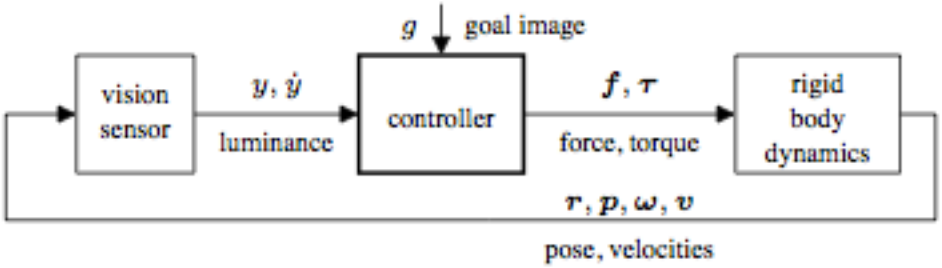


Sawyer Fuller (Caltech), Nov 08

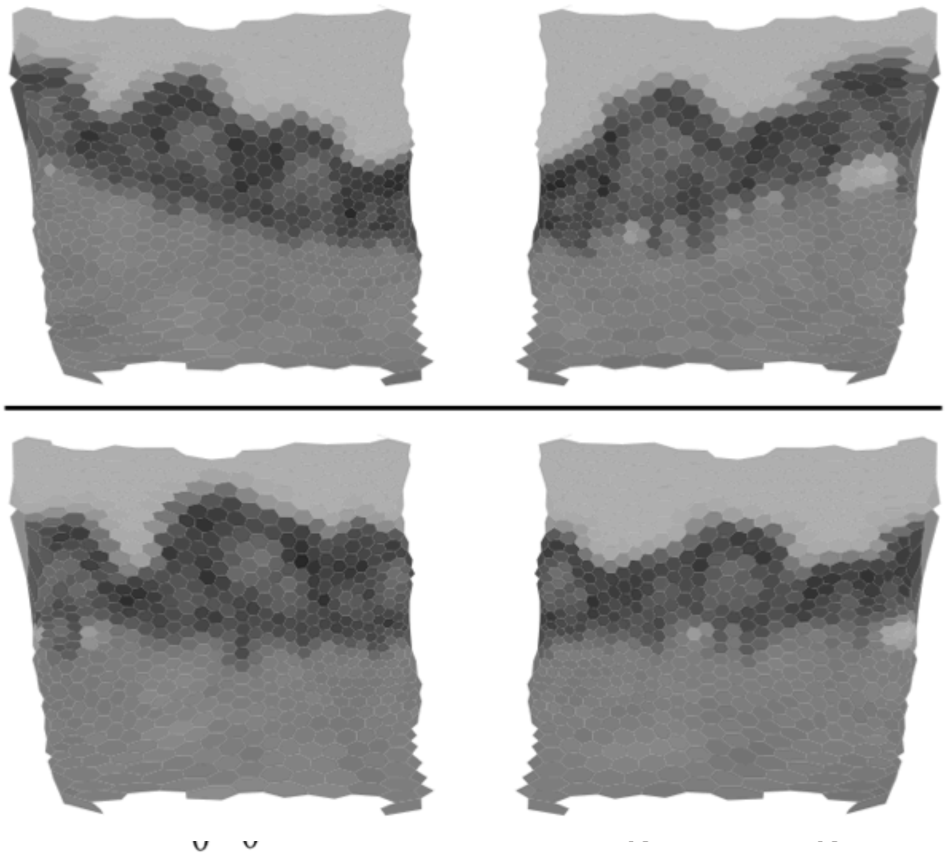


Censi, Han, Fuller, Straw and M
CDC09

Bootstrappable Design for Visual Pose Stabilization



(a) Model for purely visual pose stabilization.



Approach: minimize image error

$$J(q) = \frac{1}{2}\langle (y-g)^2 \rangle = \frac{1}{2}\int_{S^2} (y(s)-g(s))^2 ds$$

- Can write the resulting controller in terms of spatial gradients + products

$$\boldsymbol{\tau} = \|\mathbb{I}\| \langle (\mathrm{S}y)(g-y) \rangle - k_d \|\mathbb{I}\| \langle (\mathrm{S}y)\, \dot{y} \rangle ,$$
$$\boldsymbol{f} = \alpha m \langle (\nabla y)\,(g-y) \rangle - \alpha m k_d \langle (\nabla y)\, \dot{y} \rangle ,$$

$$\mathrm{S}y \triangleq s \times \nabla_s y$$

Lagged or delayed y also works

- *S* can be *learned* (up to a positive definite factor) by watching how the environment moves given f, τ

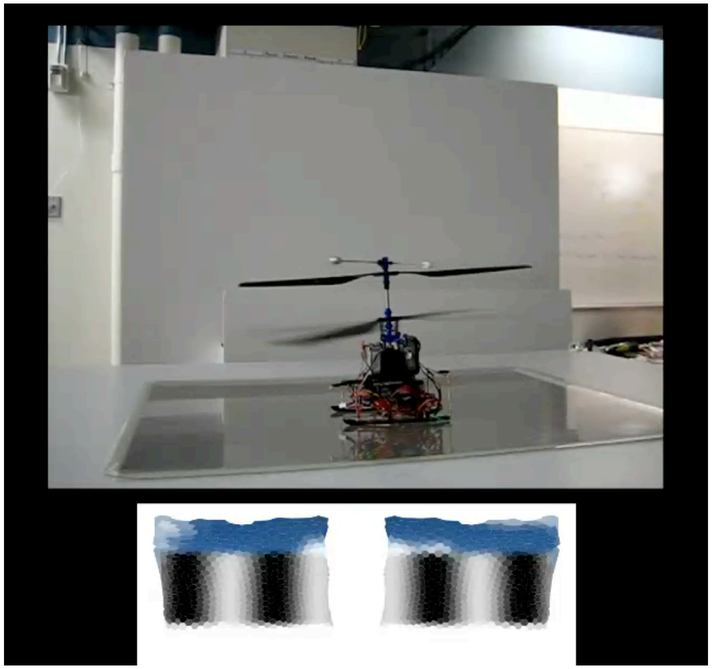
Structure of the resulting controller

- Weights have a sparse structure: take positive and negative products of nearby sensors
- Controller with delayed y has similar structure to Reichardt correlator (rough measurement of optic flow)

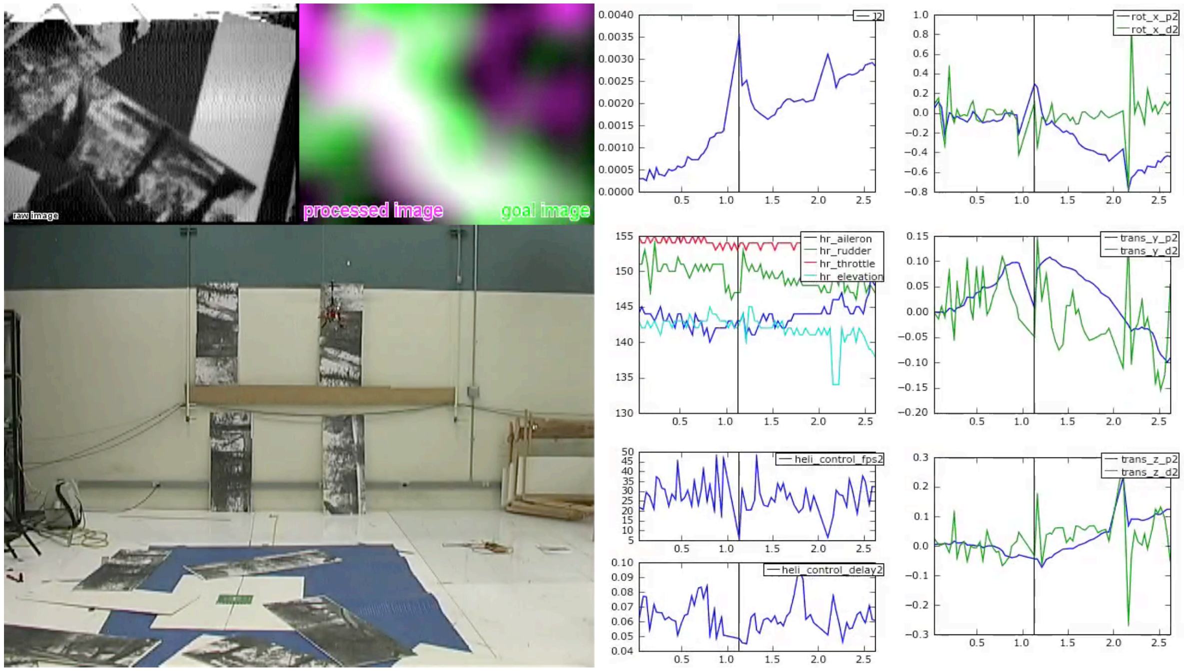
Censi, Han, Fuller, Straw and M
CDC09, IROS10 (s)

Controller Implementation

Stripe fixation (rotation only):

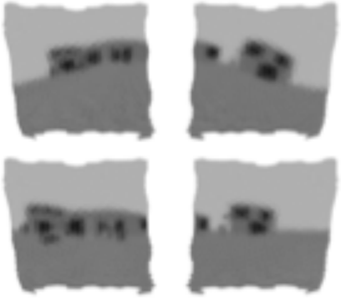


3D pose experiments

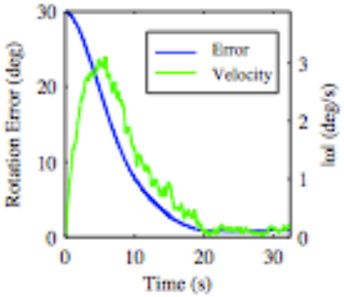


3D pose stabilization (simulations):

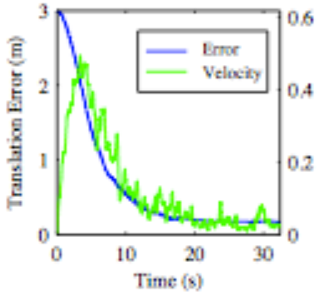
(f) Artificial environment



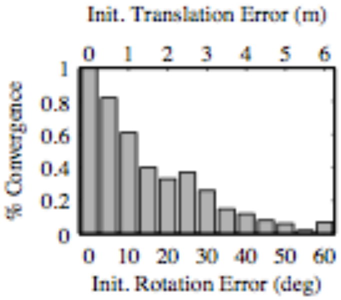
(g) Start/end images



(h) Rotation error and $|\omega|$



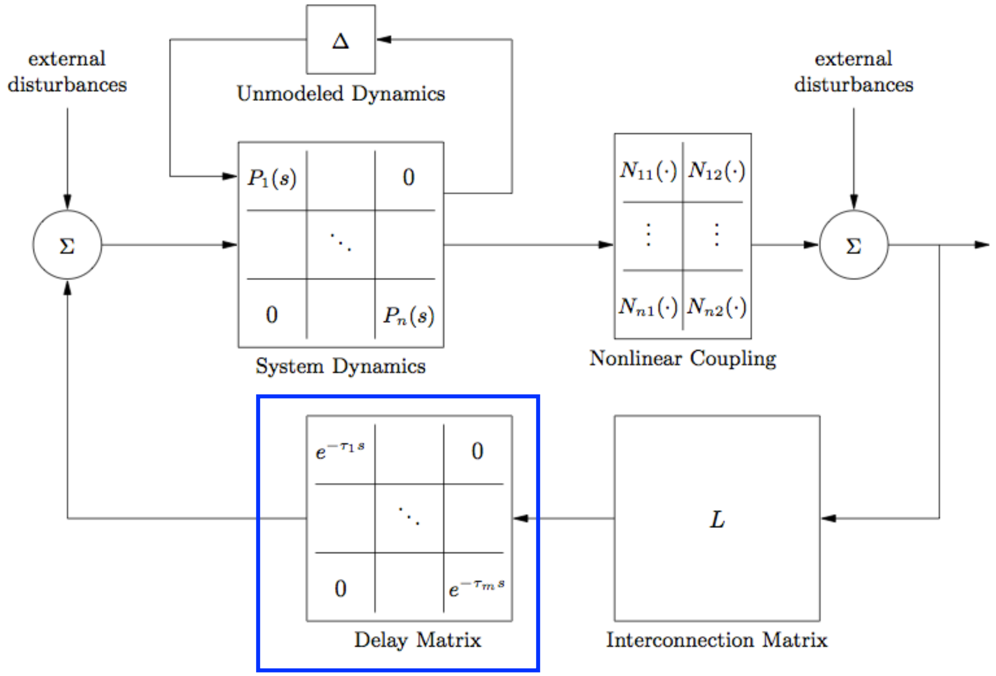
(i) Translation error and $|v|$



(j) Convergence test

Lavaei, Sojoudi and M
ACC, 2010

Control Using Time Delay



Can we design control laws using (possibly variable) time delay?

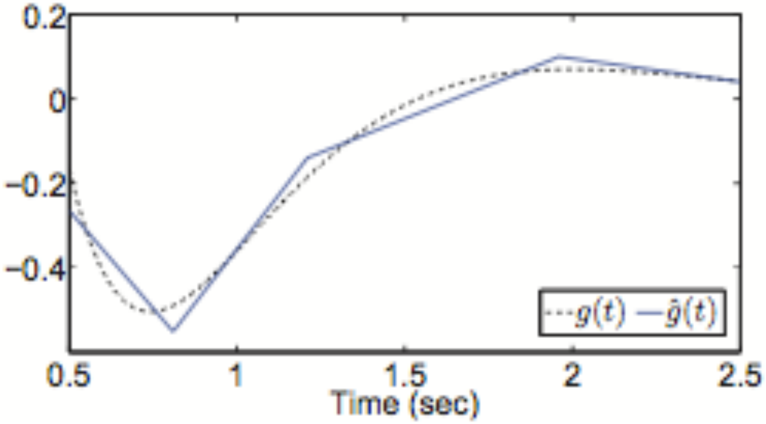
- Easy to obtain in many bio systems
- Idea: combine signals with different amounts of delay to get control $\hat{G}(s)$
- Approach: given G(s), implement the impulse response using delay + 1-2 integrators (or lags)

$$\hat{G}(s): \quad \ddot{u} = \sum \alpha_i y(t - \tau_i)$$

Preliminary results

Theorem 3: The approximation error $\|G(j\omega) - \hat{G}(j\omega)\|_\infty$ satisfies the following inequality:

$$\|G(s) - \hat{G}(s)\|_\infty \leq \sqrt{2}\int_0^{\tau_1} |g(t)|dt + \sqrt{2}\int_{\tau_k}^{\infty} |g(t)|dt$$
$$+ \sum_{i=1}^{k-1} \max_{\tau \in [\tau_i, \tau_{i+1}]} |g''(\tau)| \frac{\sqrt{2}(\tau_{i+1} - \tau_i)^3}{12}.$$

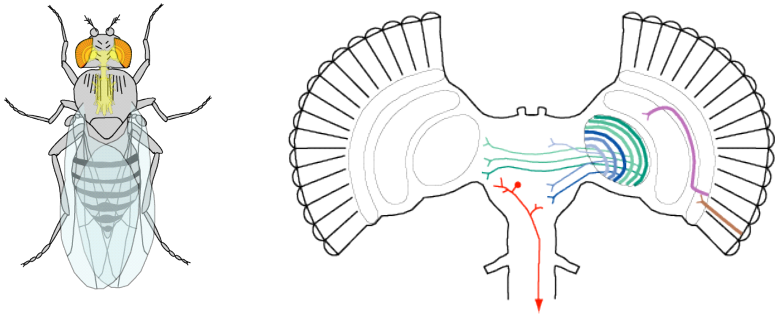


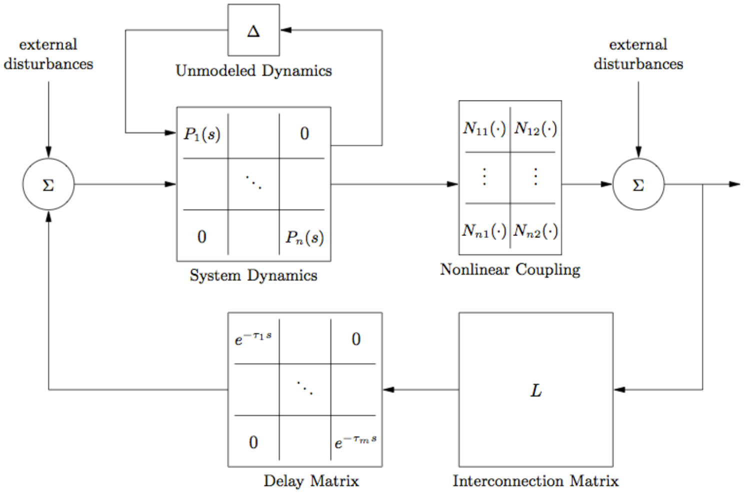
- Can also get bounds on error when delays vary (possibly useful for jitter?)

Summary and Conclusions

Bioplausible approaches using slow computing

- Look at the opposite extreme to high speed computing; what can you do with a few watts...
- Approach #1: highly parallel computation using lots of sensors and simple linear + nonlinear ops
  - Control design via interconnect + nonlinear
  - Bootstrap algorithms thru online learning
- Approach #2: control using time delay
  - Treat time-delay as programmable element
  - Fast compensation becomes trickier...





Next steps:

- Highly agile control of dynamic vehicles using slow computing (DGC IV - beat the fly)
- Biomolecular implementations: motion control in nanosystems (DGC V - beat the neutrophil)
- Design approaches that exploit data-rich environments (learning, evolvability, ...)