Feedback Control Theory: Architectures and Tools for Real-Time Decision Making

Richard M. Murray
California Institute of Technology

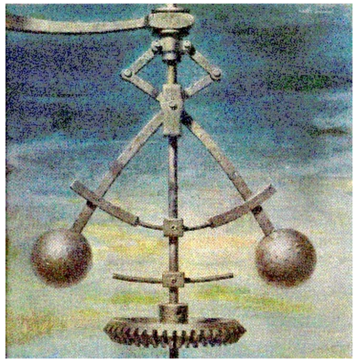
Real-Time Decision Making Bootcamp
Simons Institute for the Theory of Computing

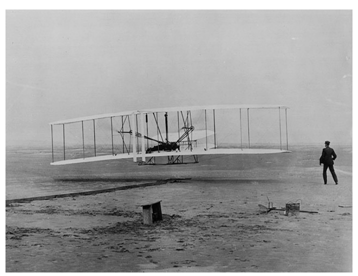
24 January 2018

Goals for this lecture

- Give an *overview* of key ideas from control theory that might be relevant for applications in real-time decision making
- Encourage you to come find me if you want to learn more or work on a joint project applying ideas from control theory
- RMM schedule: Tue-Thu most weeks from now to end of Mar

What is “Control”?





Traditional view

- Use of feedback to provide stability, performance, robustness

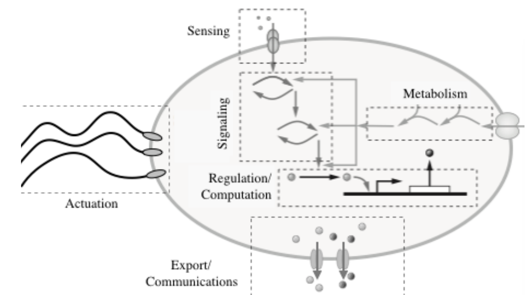
Emerging view

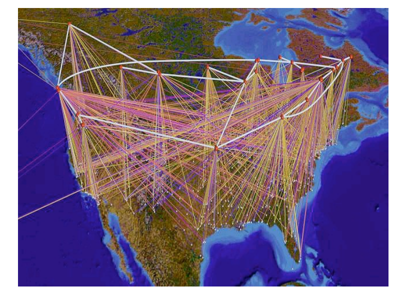
- Collection of tools and techniques for analyzing, designing, implementing complex systems in presence of uncertainty
- Combination of dynamics, interconnection (feedback/ feedforward), communications, computing and software

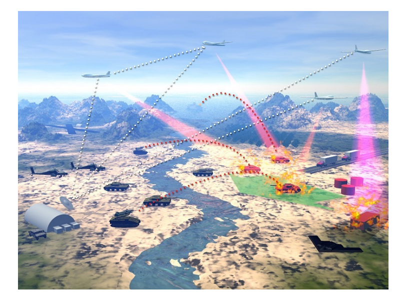
Control = dynamics ⊗ uncertainty ⊗ feedforward ⊗ feedback

Key principles for control systems

- Principle #1: Feedback is a tool for managing uncertainty (system and environment) [no uncertainty ⇒ don’t bother]
- Principle #2: Feedforward & feedback are tools for design of dynamics via integration of sensing, actuation & computation
- Corollary: Feedback enables subsystem modularity and interoperability ⇒ ability to manage complexity at scale







Important Trends in Control in the Last 15 Years

(Online) Optimization-based control

- Increased use of online optimization (MPC/RHC)
- Use knowledge of (current) constraints & environment to allow performance and adaptability

Layering, architectures, networked control systems

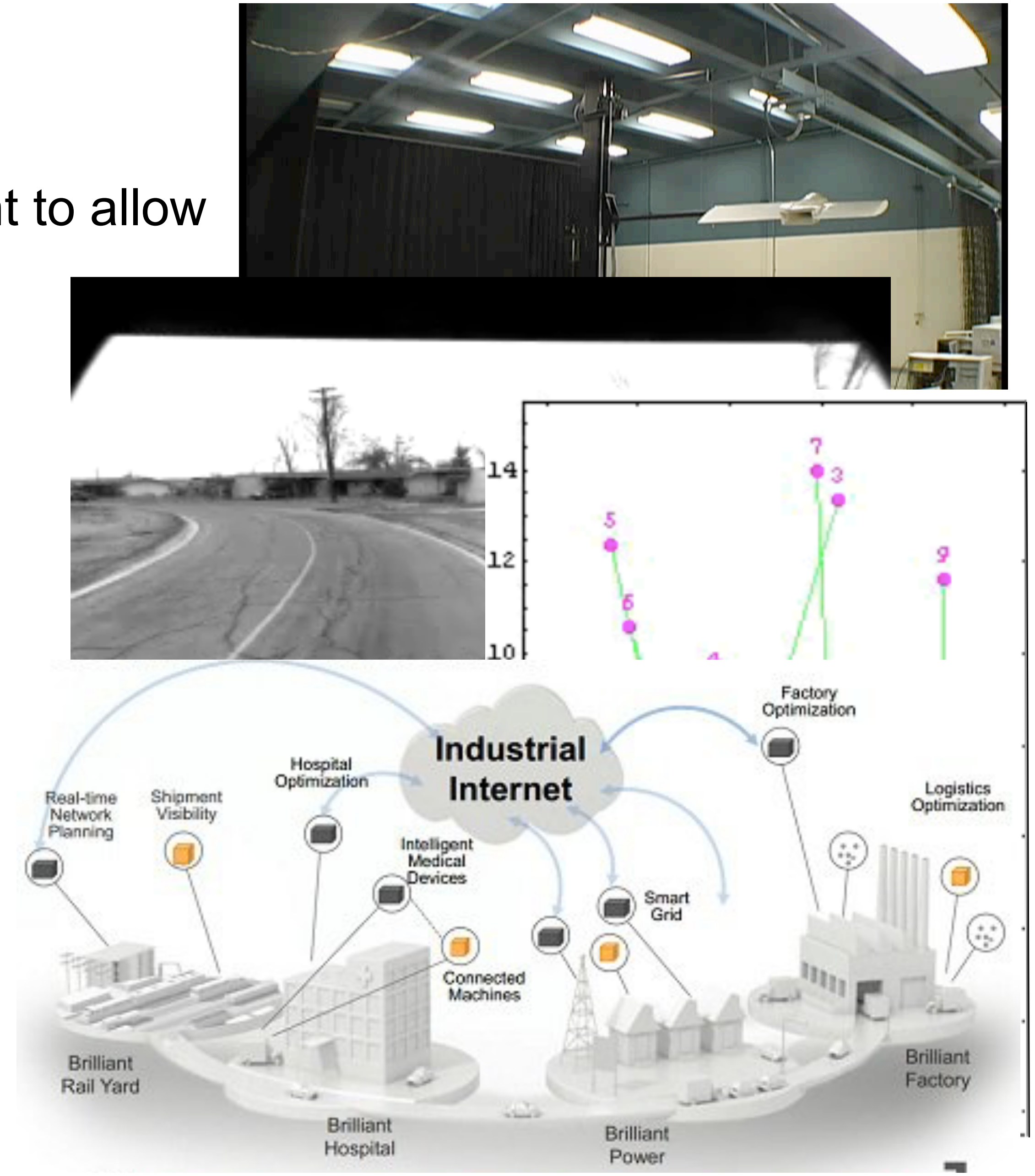
- Command & control at multiple levels of abstraction
- Modularity in product families via layers

Formal methods for analysis, design and synthesis

- Build on work in hybrid and discrete event systems
- Formal methods from computer science, adapted for “cyberphysical” (computing + control) systems

Components → Systems → Enterprise

- Increased scale: supply chains, smart grid, IoT
- Use of modeling, analysis and synthesis techniques at all levels. Integration of “software” with “controls”



Outline

Control Systems: Architectures and Examples
- "Standard model" (for control systems)
- Examples and relationship to real-time decision making

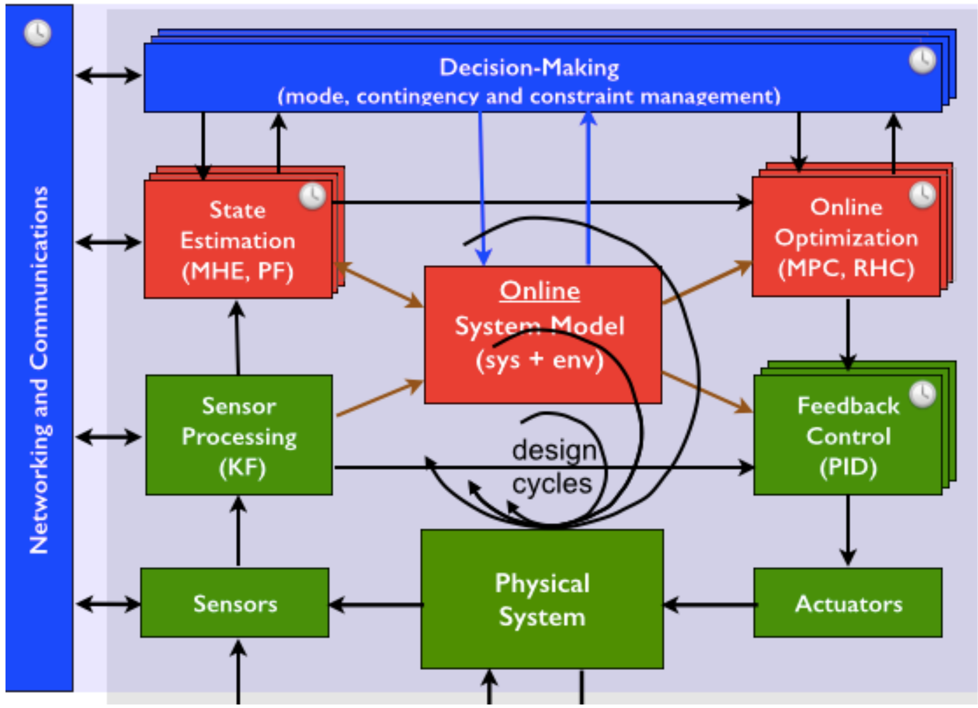
Control System Design Patterns

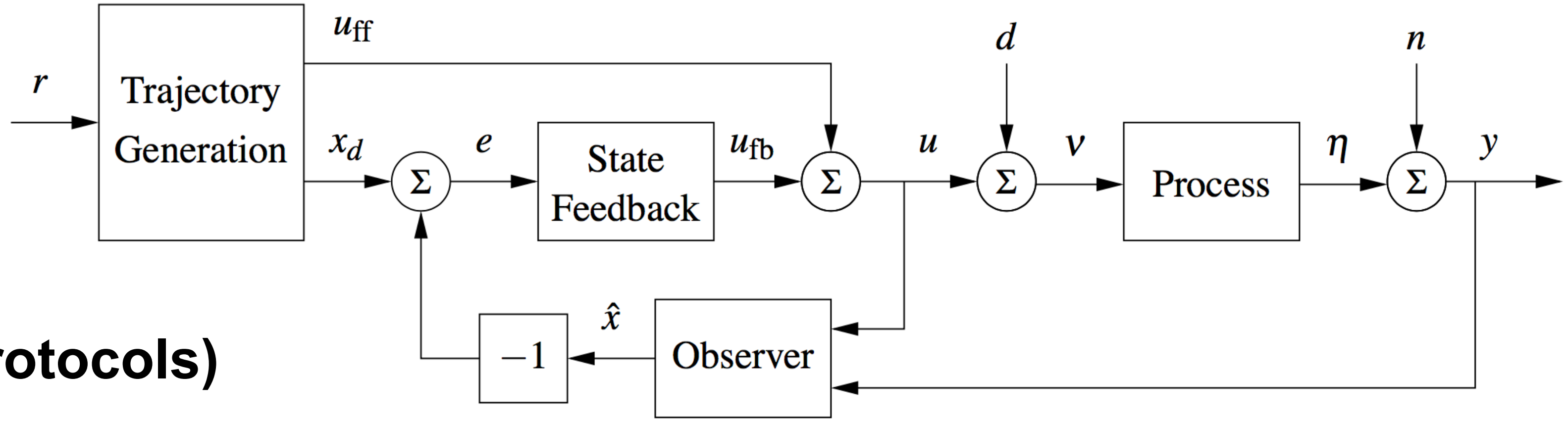
Design of Feedback Systems
- Specifications for control systems
- Integral feedback (and PID)
- State feedback

Design of Feedforward Systems
- Real-time optimization
- Receding horizon control

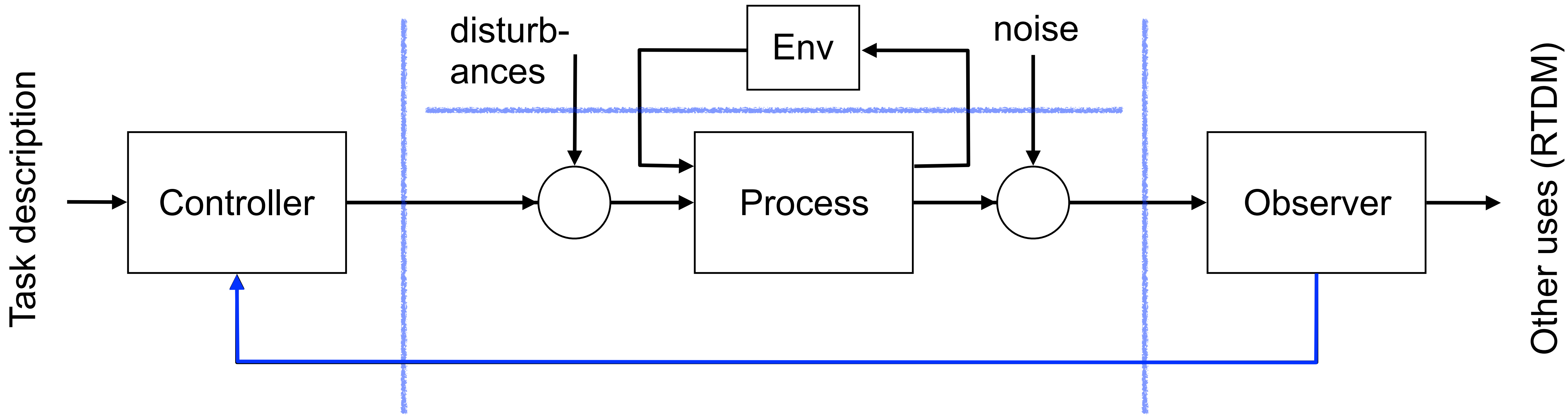
Layered architectures

Discrete state systems (reactive protocols)





Control System "Standard Model"



Key elements

- Process: input/output system w/ dynamics (memory)
- Environment: description of the uncertainty present in the system (bounded set of inputs/behaviors)
- Observer: real-time processing of process data
- Controller: achieve desired task via data, actions

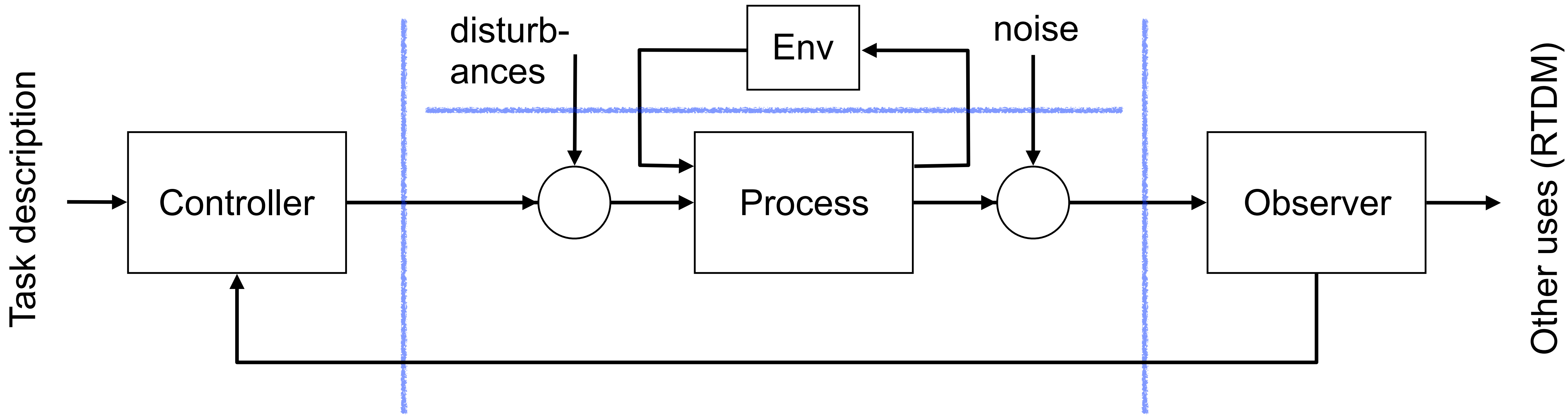
Disadvantages of feedback

- Increased complexity
- Potential for instability
- Amplification of noise

Advantages of feedback

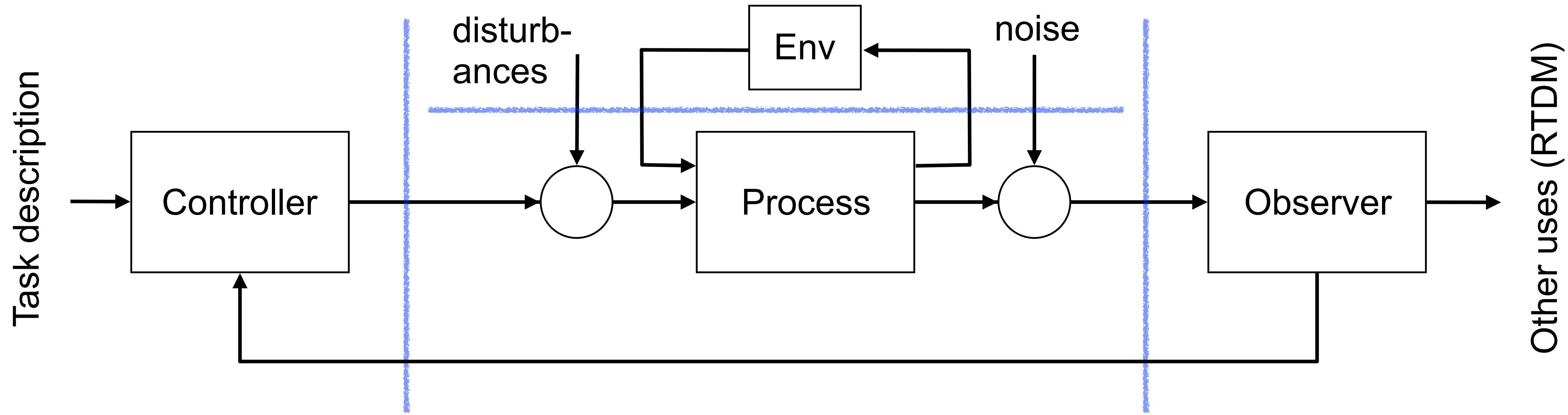
- Robustness to uncertainty
- Modularity and interoperability

Control System Examples



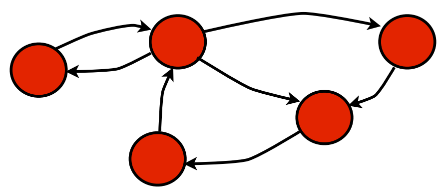
Application	Process description (input/output)	Specifications	Comments
Scientific detection	Actuator: N/A Sensor: instruments	Detect events	Observer only
Transportat'n networks	Actuator: schedules, incentives Sensor: demand, supply	Optimize utility function	Open-loop at fast time scale; closed-loop at slower rates
Autonomous vehicles	Actuator: gas, steer Sensor: cameras, radar, LIDAR, traffic	A to B w/out hitting anyone	Multiple decision-making loops; very complex environment

Different Types of Control Systems



System type	Modeling approaches	Specifications	Comments
Continuous states	Ordinary and partial differential equations; difference equations	Integrated cost over time/space	Well-studied; excellent tools avail (especially LTI systems)
Discrete state systems	Finite state automata, timed automata, Petri nets	Temporal logic formulae	Good tools for verification; design/synthesis is harder
Probabilistic systems	Stochastic ODEs, Kolmogorov equations, Markov chains	Expected values and moments	Well-studied; excellent tools avail (especially LTI, MDPs)

Control System Specifications

Level	Model	Specification
Regulation	$y = P_{yu}(s)\, u + P_{yd}(s)\, d$ $\lVert W(s) d(s) \rVert \leq 1$	$\lVert W_1 S + W_2 T \rVert_\infty < \gamma$
Optimization (planning)	$\dot{x} = f_\alpha(x, u)$ $g_\alpha(x, u, z) \leq 0$	$\min J = \int_0^T L_\alpha(x, u)\, dt + V(x(T))$
Decision-Making		$(\phi_{\text{init}} \wedge \Box \phi_{\text{env}}) \implies (\Box \phi_{\text{safe}} \wedge \Box \Diamond_{\leq T} \phi_{\text{live}})$

Transient: initial response to input

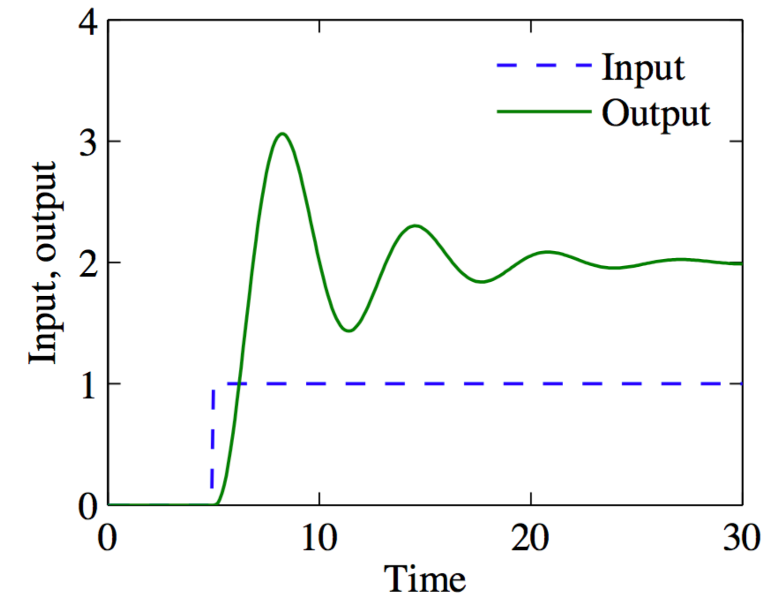
- Step response: rise time, overshoot, settling time, etc

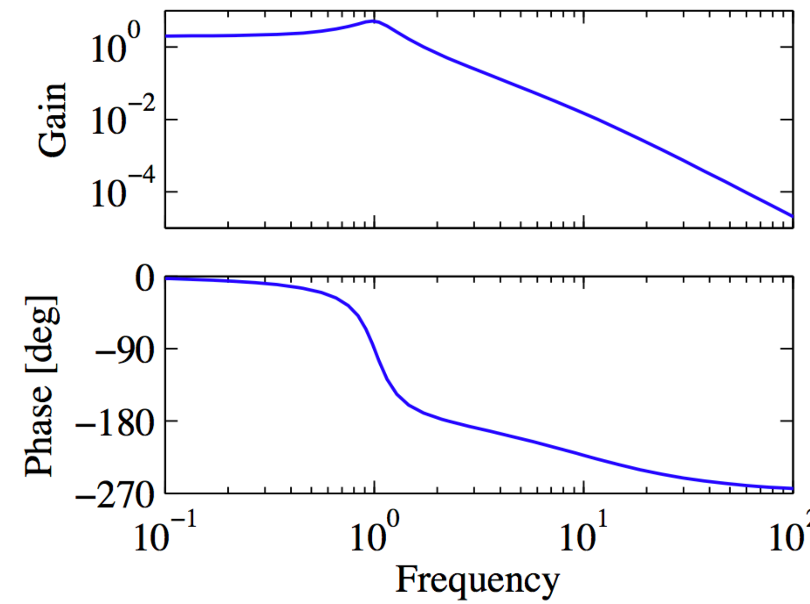
Steady state: response after the transients have died out

- Frequency response: magnitude and phase for sinusoids

Safety: constraints that the system should never violate

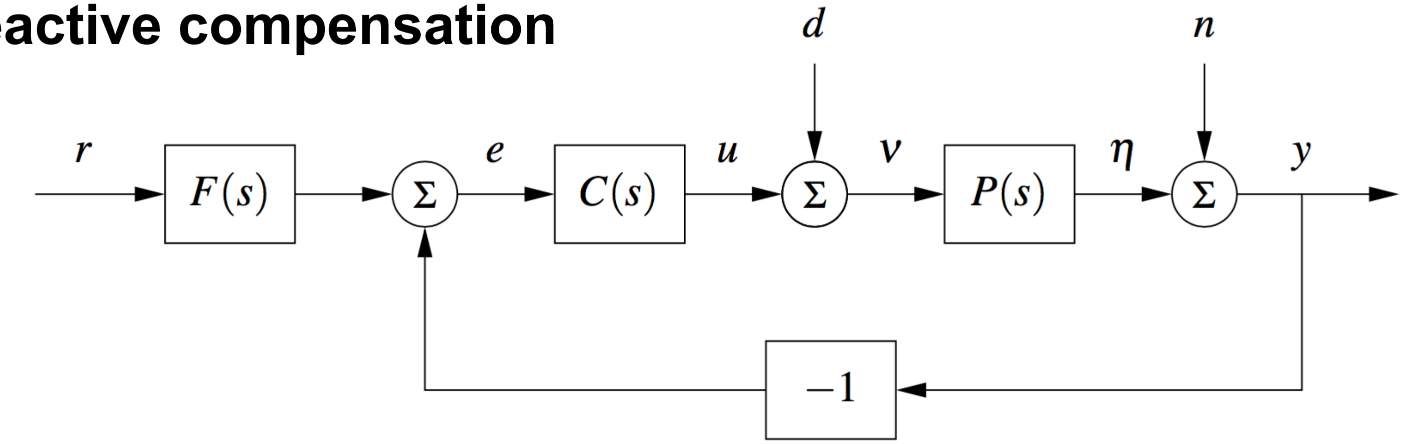
Liveness: conditions that system should satisfy repeatedly





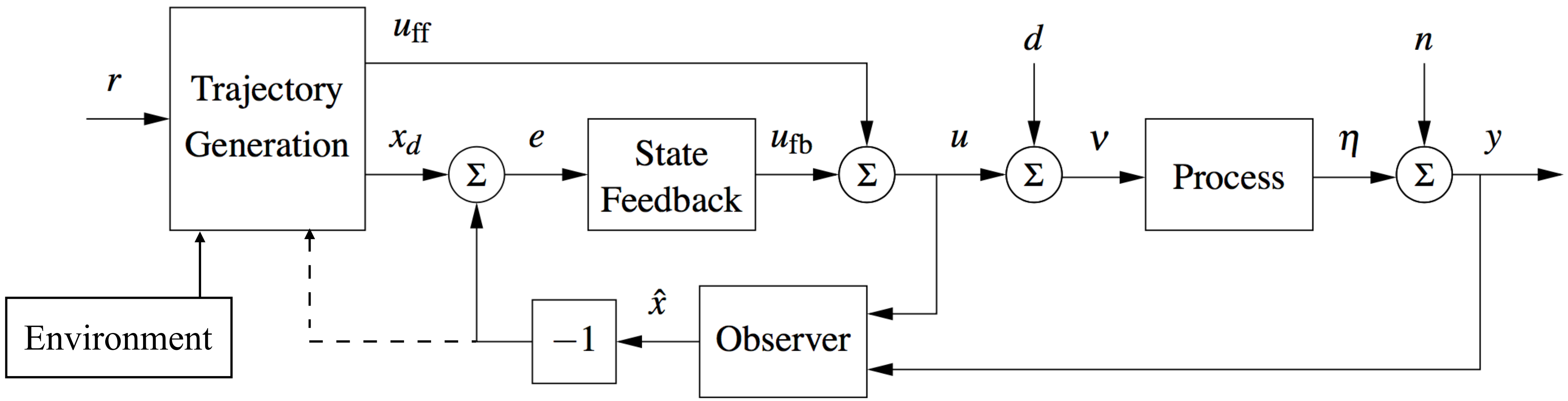
Design Patterns for Control Systems

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

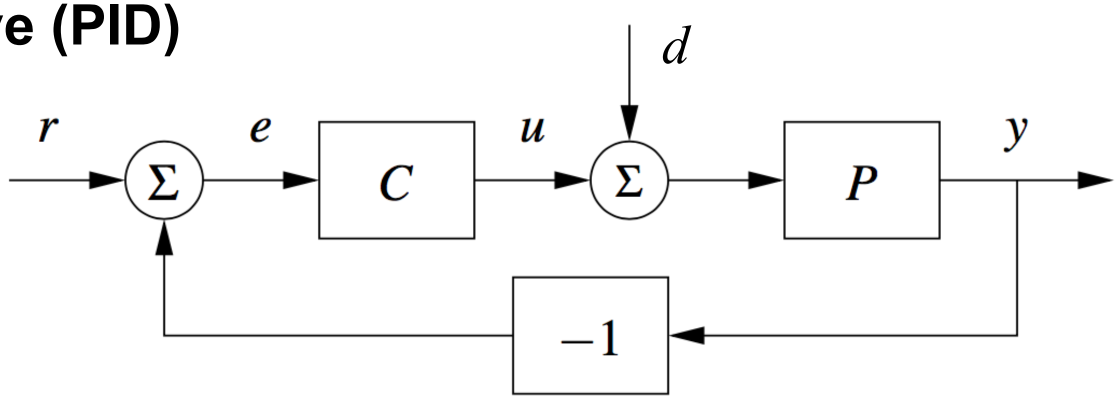


- Explicit computation of trajectories given a model of the process and environment

Feedback Design Tools: PID Control

Three term controller: proportional, integral, derivative (PID)

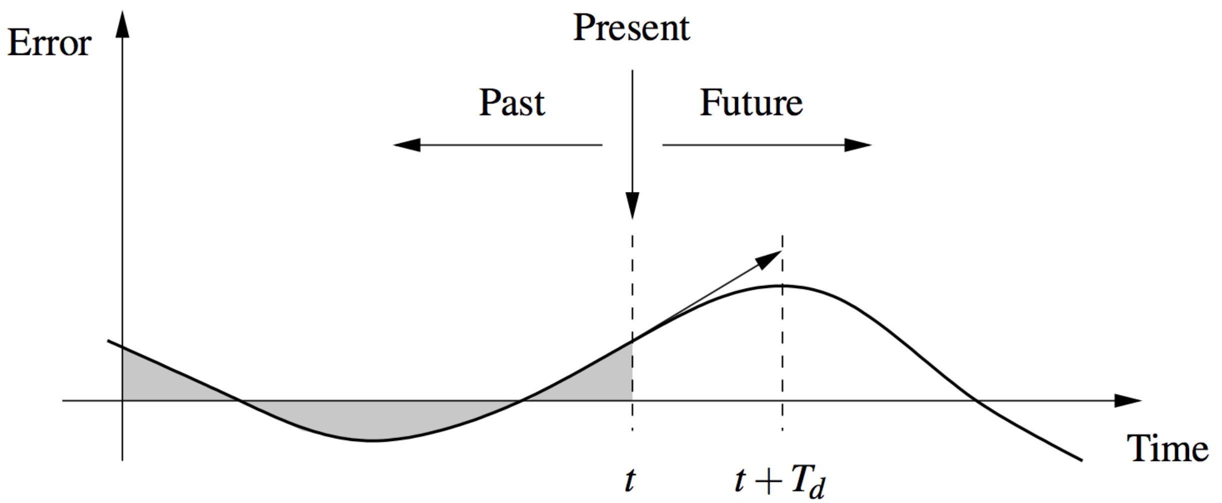
- Present: feedback proportional to current error
- Past: feedback proportional to integral of past error
  - Insures that error eventual goes to 0
  - Automatically adjusts setpoint of input
- Future: derivative of the error
  - Anticipate where we are going



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

PID design

- Choose gains k, ki, kd to obtain desired behavior
- Stability: solutions converge to equilibrium point
- Performance:
  - output of system, y, should track reference
  - disturbances d should be attenuated
- Robustness: stability and performance properties should hold in face of disturbances & process uncertainty



Feedback Design Tools: State Space Control

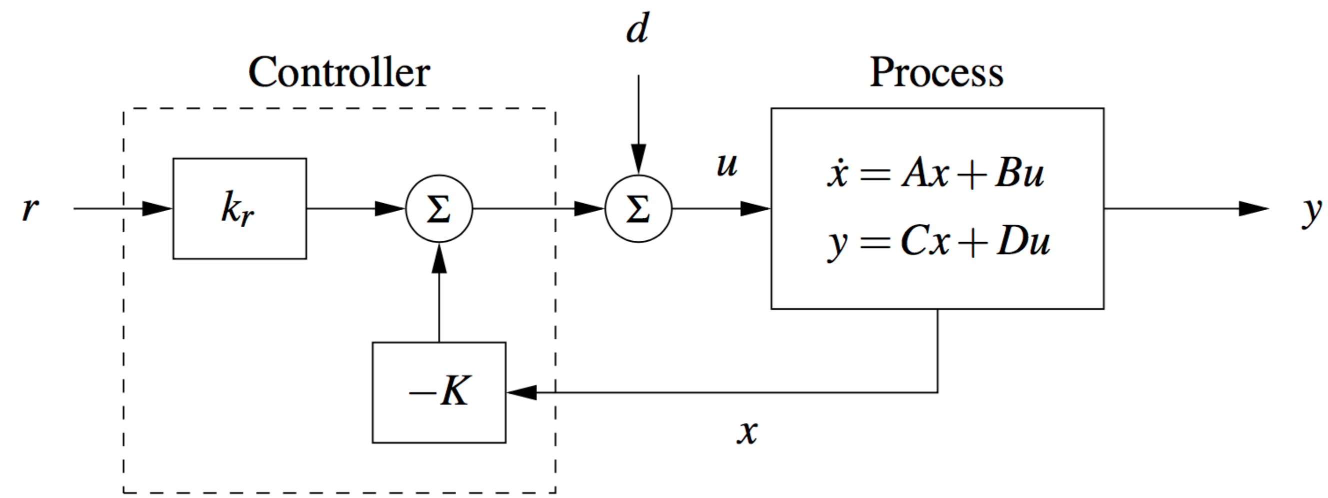
$$\dot{x} = Ax + Bu \qquad x \in \mathbb{R}^n,\ x(0) \text{ given}$$
$$y = Cx + Du \qquad u \in \mathbb{R},\ y \in \mathbb{R}$$

$$x(T) = e^{AT}x_0 + \int_{\tau=0}^{T} e^{A(T-\tau)}Bu(\tau)d\tau$$

Goal: find a linear control law $u = -Kx + k_r r$ such that the closed loop system

$$\dot{x} = Ax + Bu = (A - BK)x + Bk_r r$$

is stable at equilibrium point x_e with $y_e = r$.



Remarks

- If $r = 0$, control law simplifies to $u = -Kx$ and system becomes $\dot{x} = (A - BK)x$
- Stability based on eigenvalues ⇒ use K to make eigenvalues of $(A - BK)$ stable
- Can also link eigenvalues to *performance* (eg, initial condition response)
- Q: Can we place the eigenvalues anyplace that we want? A: Yes, if *reachable*

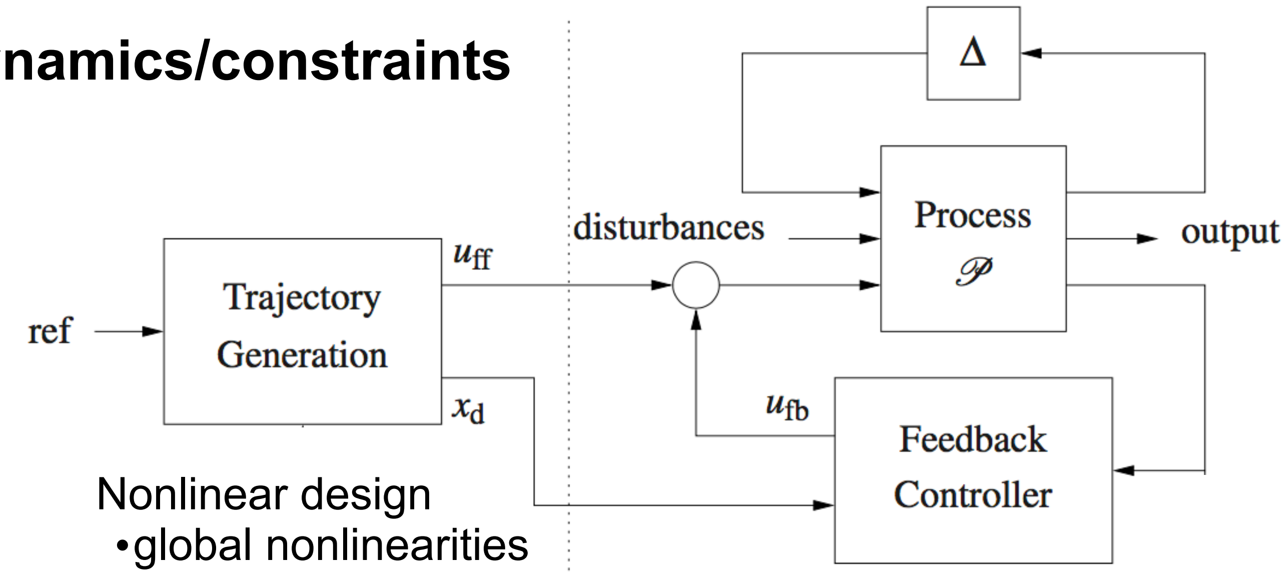
MATLAB/Python: K = place(A, B, eigs), K = lqr(A, B, Q, R), … Note: this is *design of dynamics*

Feedforward Design Tools: Real-Time Trajectory Generation

Goal: find a feasible trajectory that satisfies dynamics/constraints

$$\min J = \int_{t_0}^{T} q(x,u)\,dt + V(x(T),u(T))$$

$$\dot{x} = f(x,u) \qquad lb \leq g(x,u) \leq ub$$



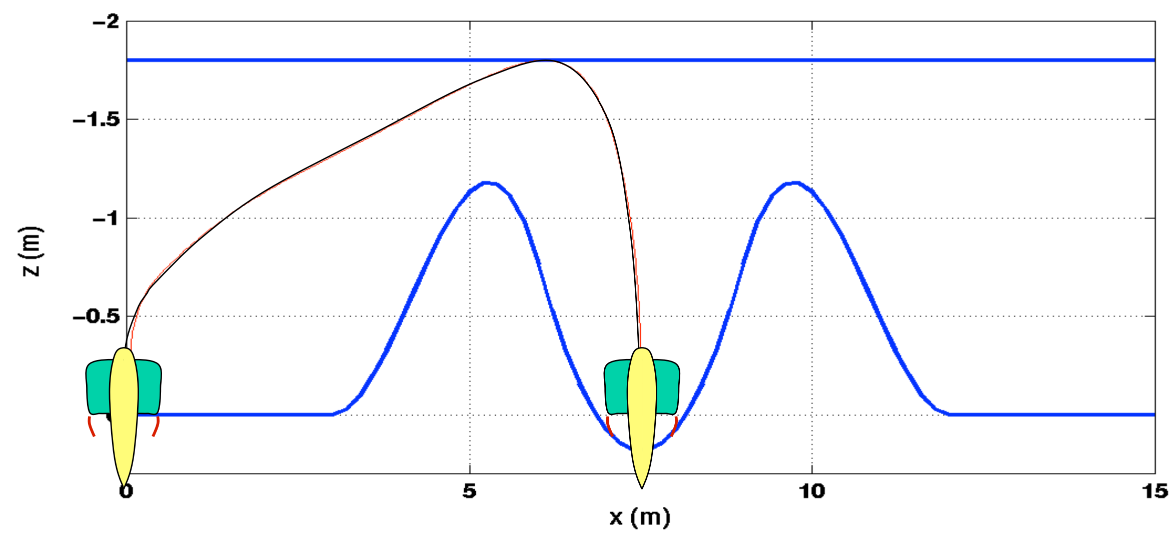
Nonlinear design
- global nonlinearities
- input saturation
- state space constraints

Solve as a constrained optimization problem

- Various tricks to get very fast calculations
- Need to update solutions at the rate at which the reference (task description) is modified

Use feedback ("inner loop") to track trajectory

- Trajectory generation provides feasible trajectory plus nominal input
- Feedback used to correct for disturbances and model uncertainties
- Example of "two degree of freedom" design



Feedforward Design Tools: Receding Horizon Control

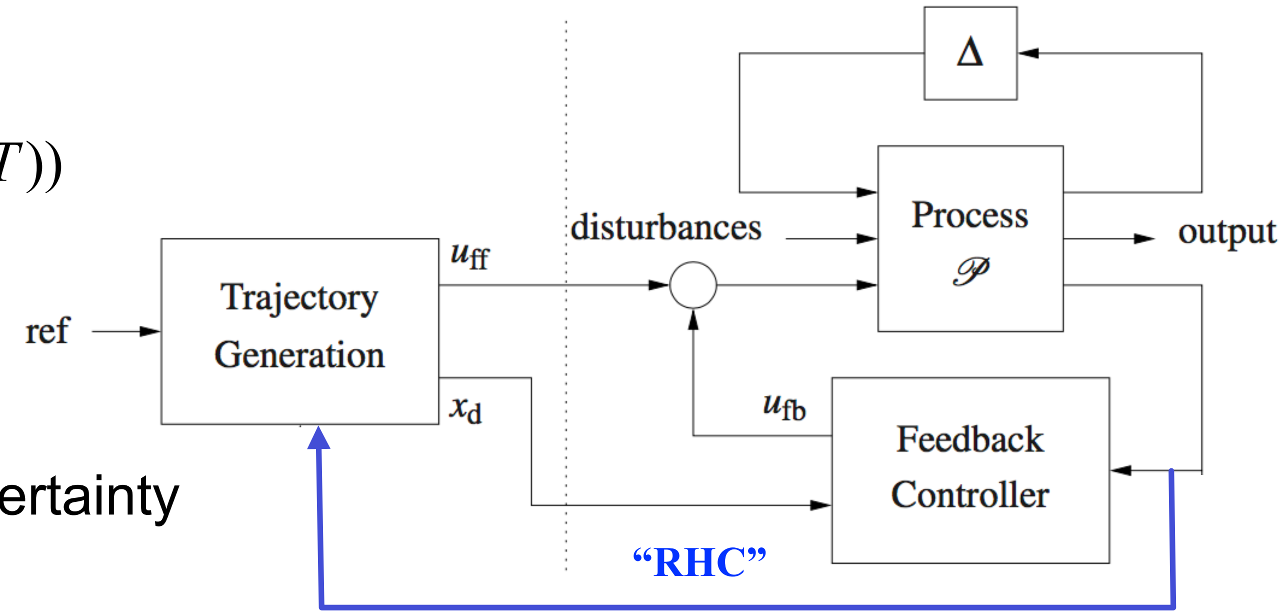
Basic idea: recompute solutions

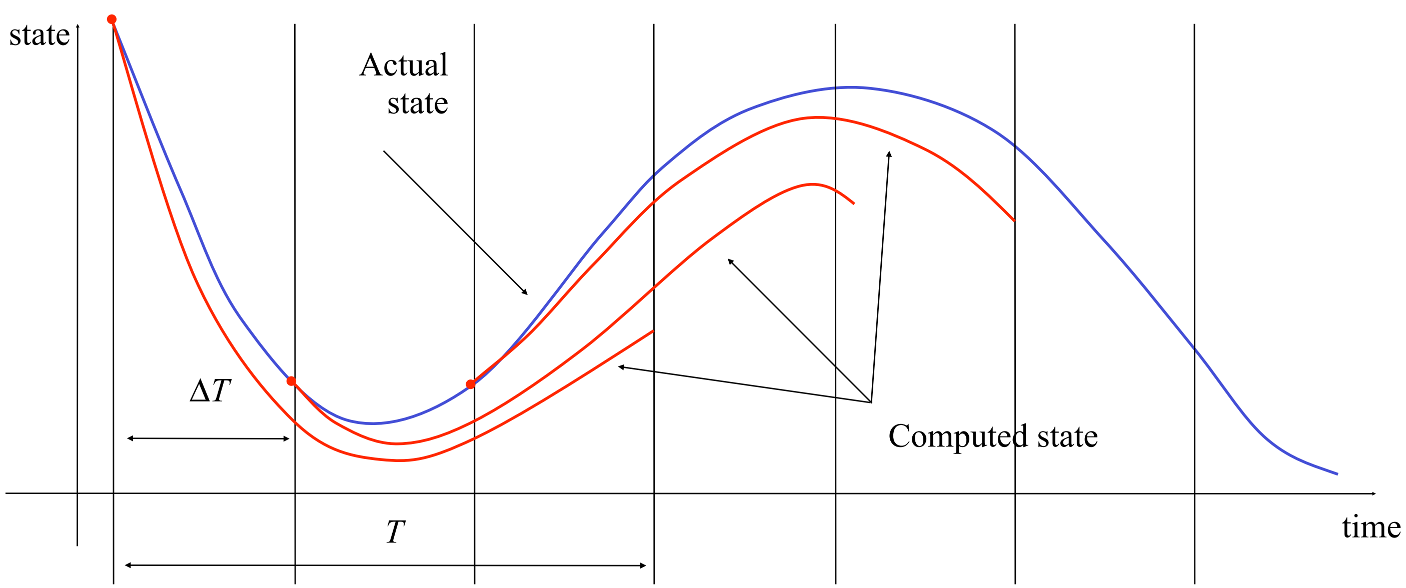
$$u_{[t,t+\Delta T]} = \arg\min \int_t^{t+T} L(x(\tau),u(\tau))\,d\tau + V(x(t+T))$$

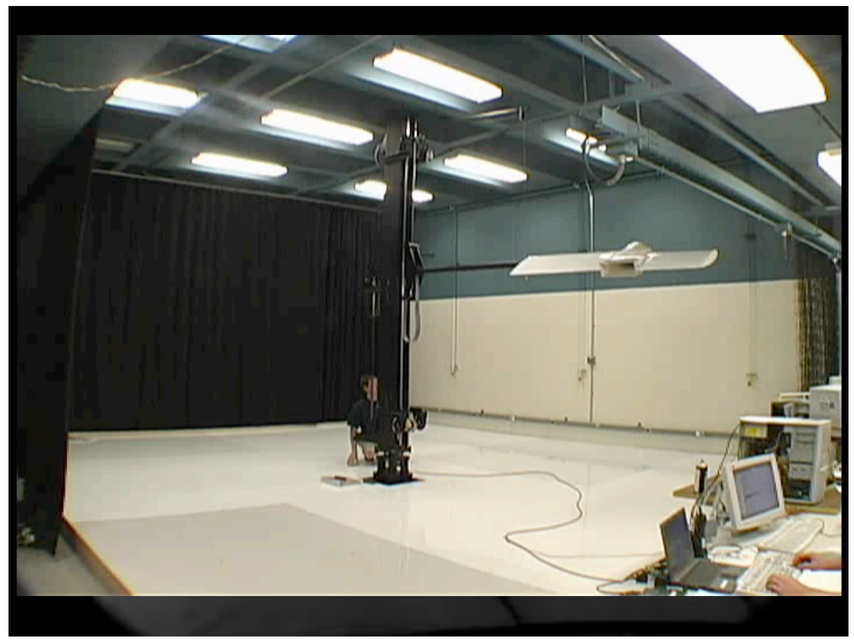
$$x_0 = x(t) \quad x_f = x_d(t+T)$$

$$\dot{x} = f(x,u) \quad g(x,u) \le 0$$

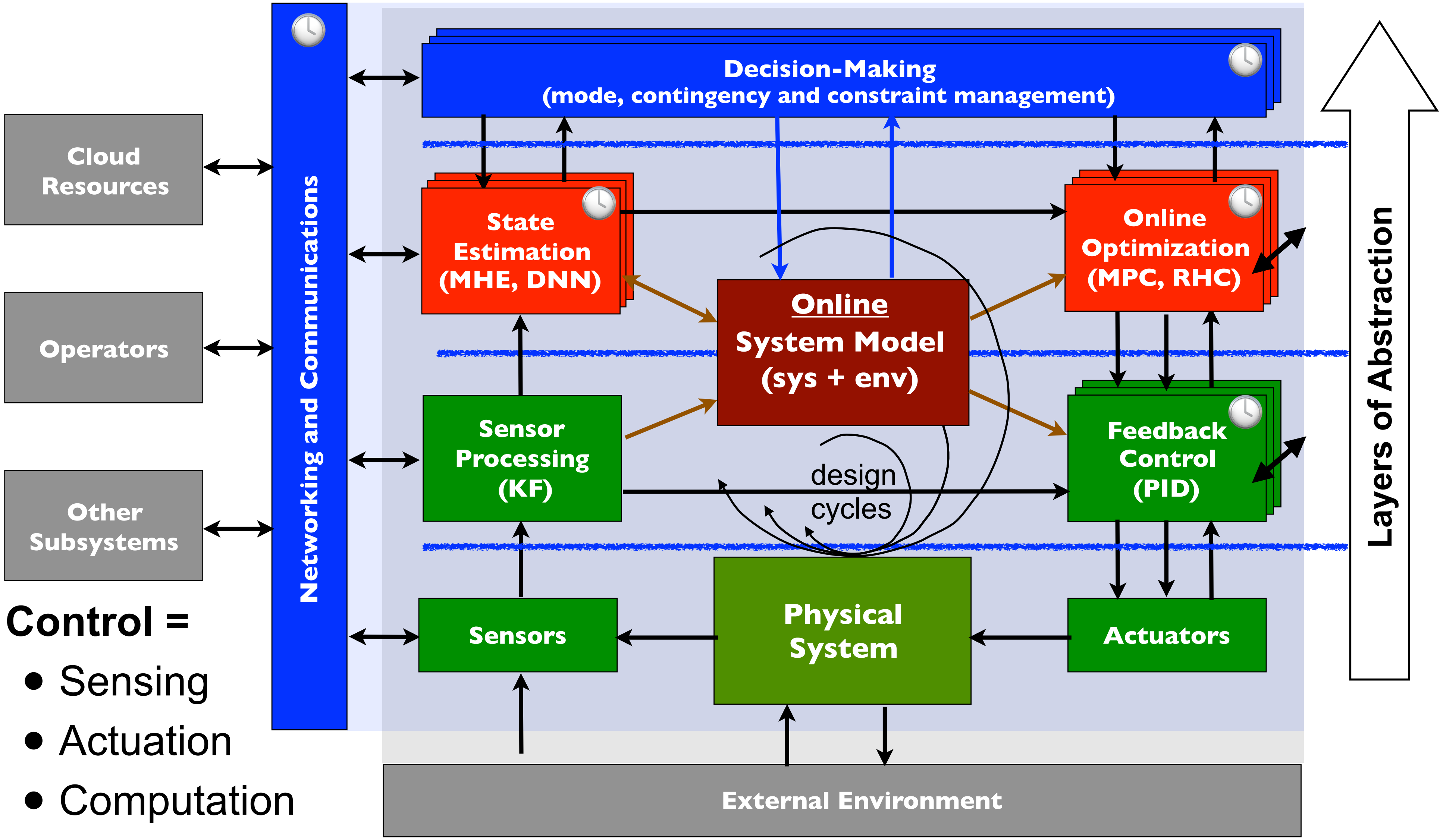
- Provides second feedback loop to manage uncertainty
- Need to be careful about terminal constraints







Design of Modern (Networked) Control Systems



Control =

- Sensing
- Actuation
- Computation

Control = dynamics, uncertainty, feedforward, feedback

Examples

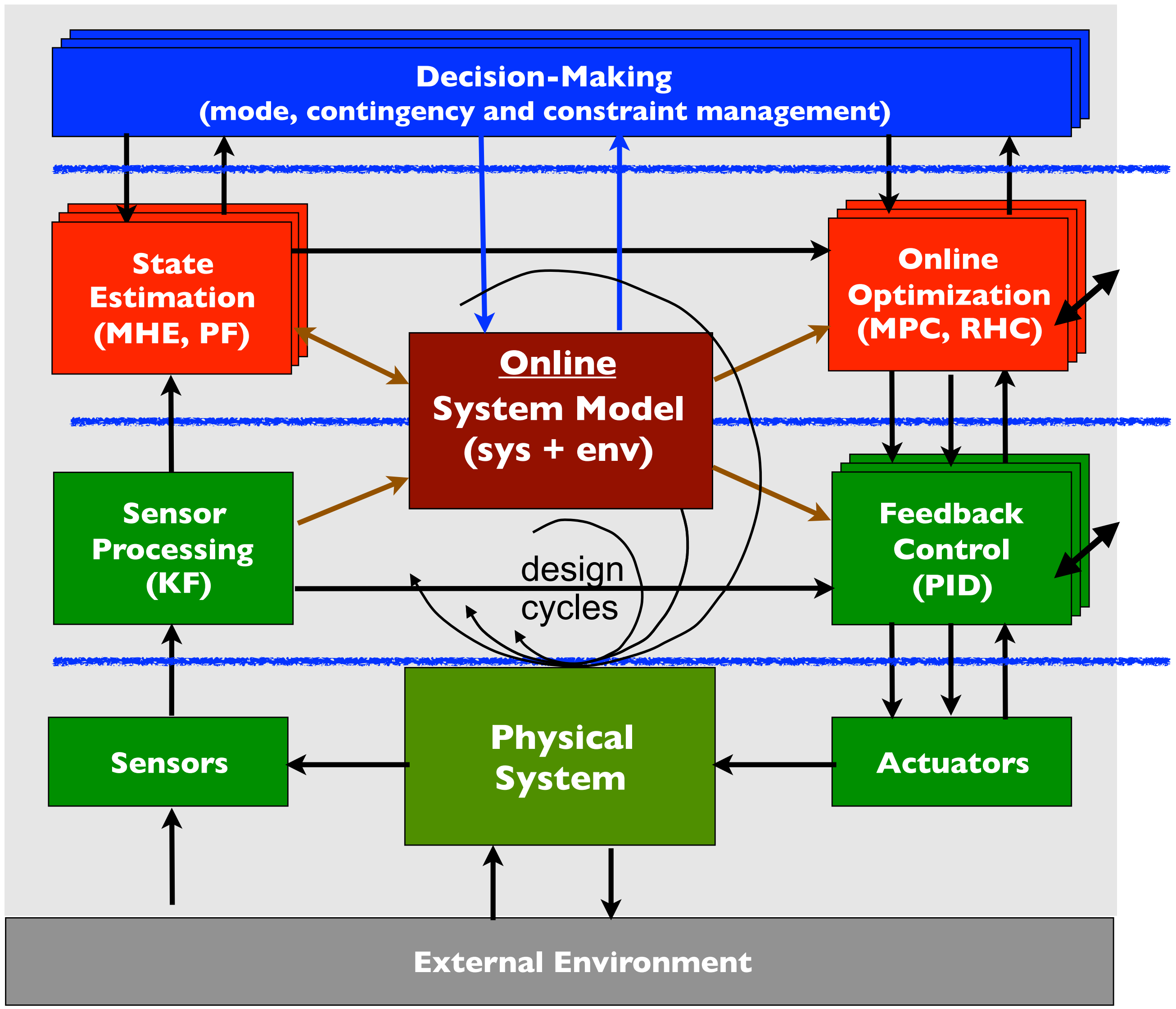
- Aerospace systems
- Self-driving cars
- Factory automation/ process control
- Smart buildings, grid, transportation

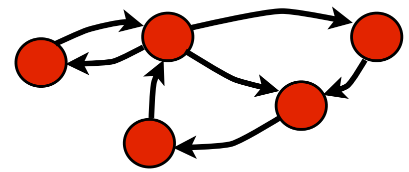
Challenges

- How do we define the layers/interfaces (vertical contracts)
- How do we scale to *many* devices (horizontal contracts)
- Stability, robustness, security, privacy

Layered Approaches to Design

Multi-layer Networked Control System



Model	Specs
	$(\phi_{\text{init}} \wedge \Box\phi_{\text{env}}) \implies (\Box\phi_{\text{safe}} \wedge \Box\Diamond_{\leq T}\phi_{\text{live}})$
$\dot{x} = f_\alpha(x,u)$ $g_\alpha(x,u,z) \leq 0$	$\min J = \int_0^T L_\alpha(x,u)\,dt + V(x(T))$
$y = P_{yu}(s)\,u + P_{yd}(s)\,d$ $\lVert W(s)d(s)\rVert \leq 1$	$\lVert W_1 S + W_2 T\rVert_\infty < \gamma$
$\dot{x}^i = f_\alpha(x^i, u^i, d^i)$ $x \in \mathcal{X}, u \in \mathcal{U}, d \in \mathcal{D}$	Operating Envelope Energy Efficiency Actuator Authority

Specifying Discrete Behavior Using Temporal Logic

Linear temporal logic (LTL)

◊ "eventually" - satisfied at some point in the future

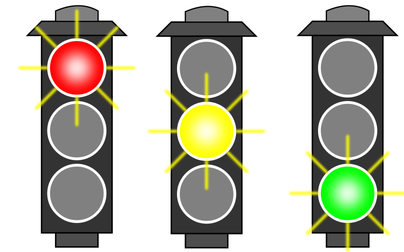
☐ "always" - satisfied now and forever into the future

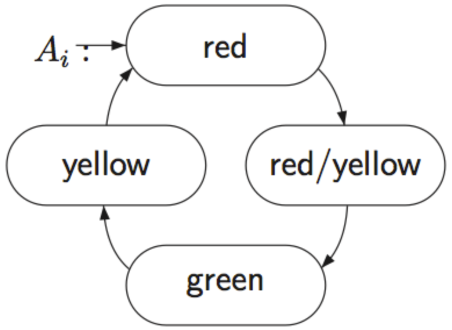
○ "next" - true at next step

Signal temporal logic (STL)

- Allow predicates that compare values
- Allow temporal bounds

• $p \rightarrow \Diamond q$	p implies eventually q (response)
• $p \rightarrow q \, U \, r$	p implies q until r (precedence)
• $\Box\Diamond p$	always eventually p (progress)
• $\Diamond\Box p$	eventually always p (stability)
• $\Diamond p \rightarrow \Diamond q$	eventually p implies eventually q (correlation)
• $V < V_{\max}$	$V(t)$ less than threshold ($V_{\max}$)
• $\Box_{[t1,t2]}\, p$	p true for all time in [t_1, t_2]
• $p \rightarrow \Diamond_{[0,t]}\, q$	if p occurs, q will occur w/in time t





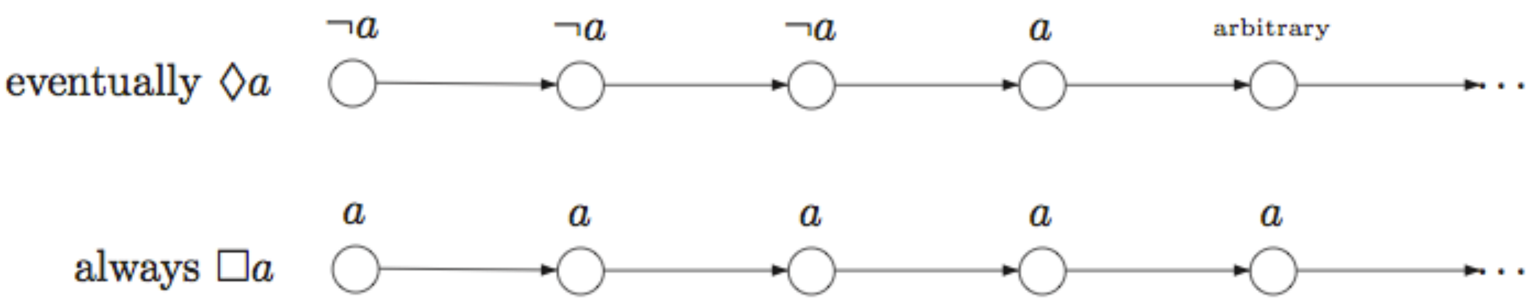
Baier and Katoen, *Principles of Model Checking*, 2007

$\Box\Diamond$green

$\Box$ (green → ¬ ○ red)

$\Box$(red → ($\Diamond$ green

∧ (¬ green U yellow)))



Baier and Katoen, *Principles of Model Checking*, 2007

Synthesis of Reactive Controllers

Reactive Protocol Synthesis

- Find control action that insures that specification is always satisfied
- For LTL, complexity is doubly exponential (!) in the size of system specification

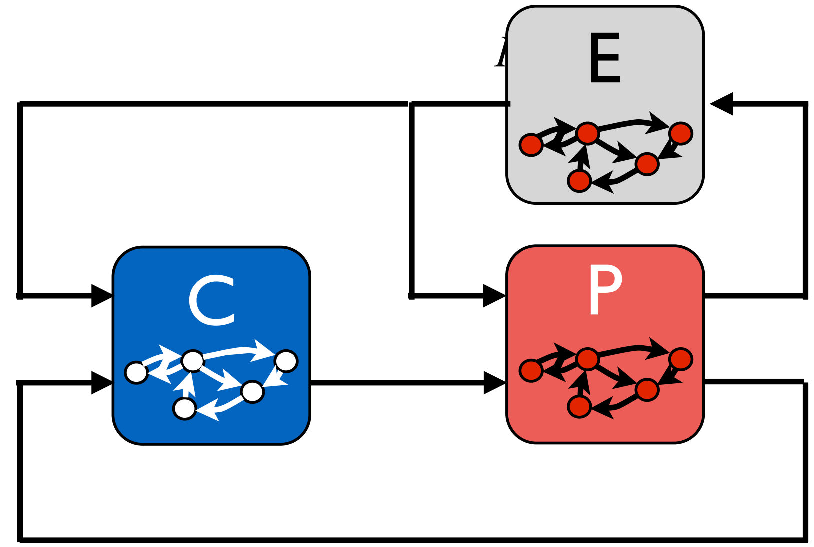
GR(1) synthesis for reactive protocols

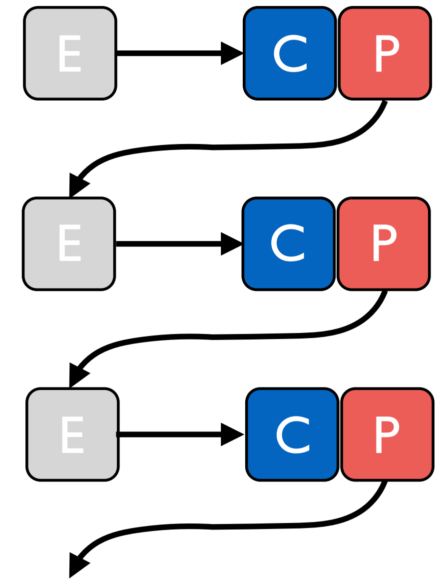
- Piterman, Pnueli and Sa'ar, 2006
- Assume environment fixes action before controller (breaks symmetry)
- For certain class of specifications, get complexity cubic in # of states (!)

$$(\phi^{\mathrm{e}}_{\mathrm{init}} \wedge \Box\phi^{\mathrm{e}}_{\mathrm{safe}} \wedge \Box\Diamond\phi^{\mathrm{e}}_{\mathrm{prog}}) \rightarrow (\phi^{\mathrm{s}}_{\mathrm{init}} \wedge \Box\phi^{\mathrm{s}}_{\mathrm{safe}} \wedge \Box\Diamond\phi^{\mathrm{s}}_{\mathrm{prog}})$$

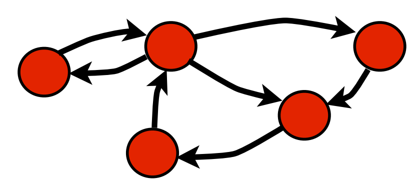
Environment assumption System guarantee

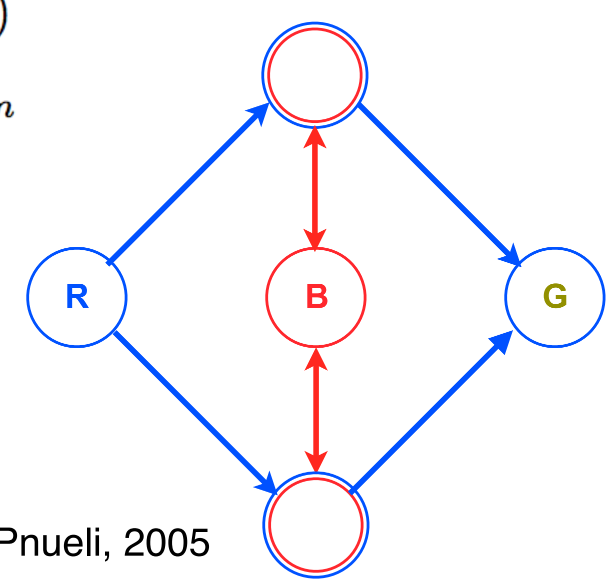
- GR(1) = general reactivity formula
- Assume/guarantee style specification





$$\mathcal{L}_m(S/G) = \mathcal{L}(S/G) \cap \mathcal{L}_m(G)$$
$$\subseteq \overline{L_{am}} \cap \mathcal{L}_m(G) = L_{am}$$



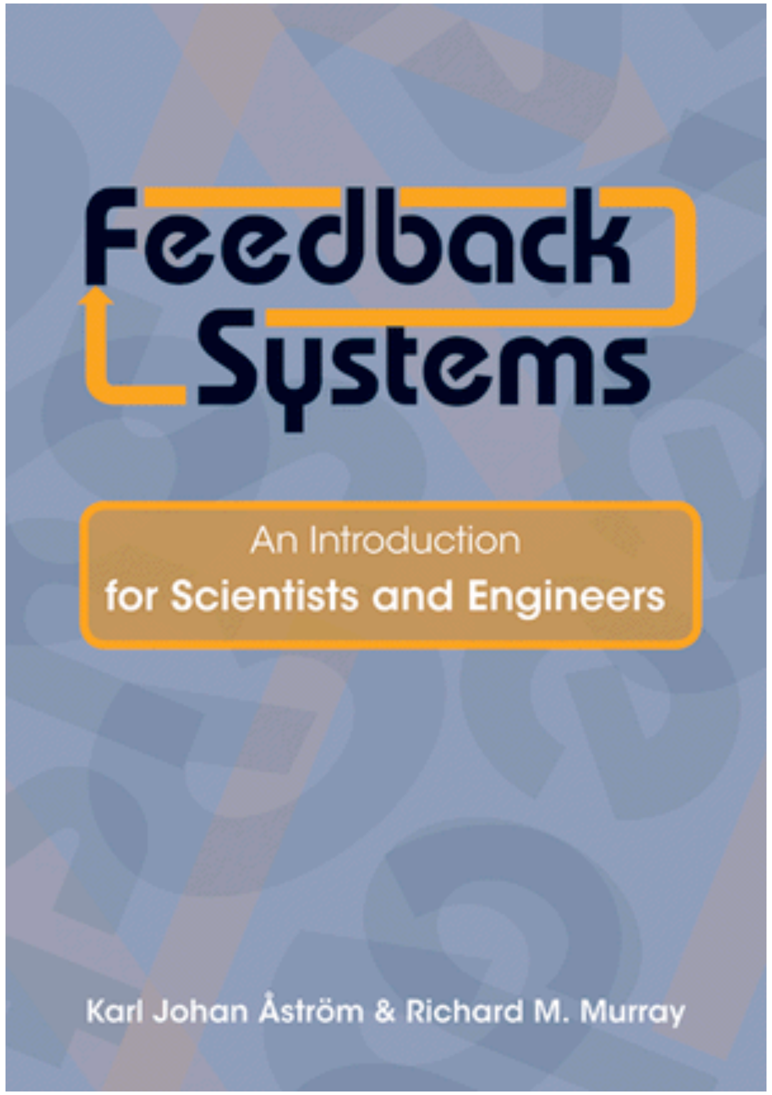


A. Pnueli, 2005

Summary: Feedback 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
- Combined, these principles enable **modularity** and **hierarchy**



Control theory: past, present and future

- Tools originally developed to design low-level control systems
- Increasing application to networked (hybrid) control systems
- New challenges: systematic design of layered architectures and control protocols, security and privacy, data-driven (AI/ML)

More information

- *Feedback Systems* (free download): *https://fbsbook.org*