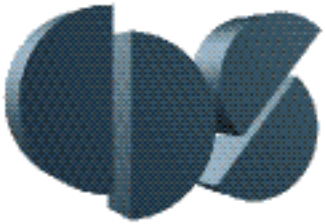
Lecture 10
Summary and Open Questions



Richard M. Murray
Nok Wongpiromsarn Ufuk Topcu
California Institute of Technology

EECI, 22 Mar 2013

Outline:

- Review key concepts from the course
- Discussion open issues, technical challenges and risks

"TS $\models \Box(\neg b \rightarrow \Box(a \wedge \neg b)$"

Control in an Information Rich World (2001-03)

1. Executive Summary

2. Overview of the Field

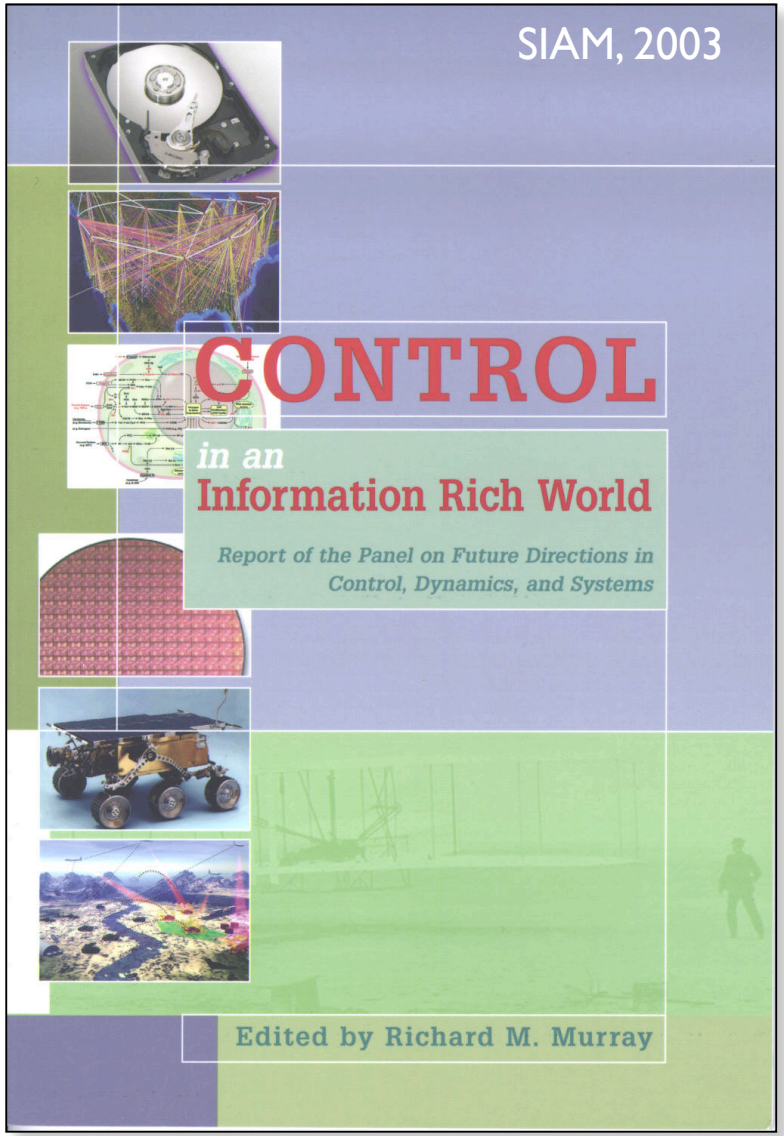
- What is Control?
- Control System Examples
- Increasing Role of Information-Based Systems
- Opportunities and Challenges

3. Applications, Opportunities & Challenges

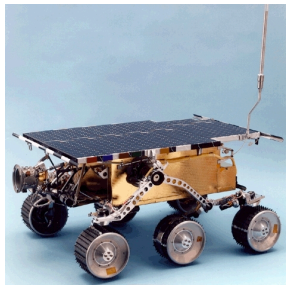
- Aerospace and Transportation
- Information and Networks
- Robotics and Intelligent Machines
- Biology and Medicine
- Materials and Processing
- Other Applications

4. Education and Outreach

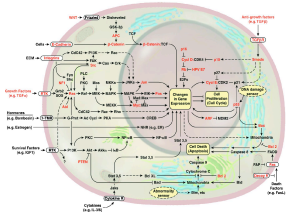
5. Recommendations



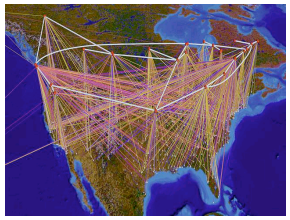
CDS Panel Recommendations



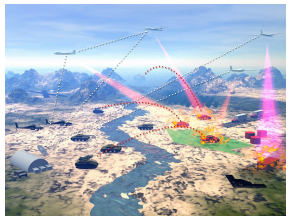
1. Substantially increase research aimed at the integration of control, computer science, communications, and networking.



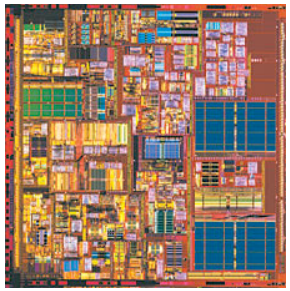
2. Substantially increase research in control at higher levels of decision making, moving toward enterprise level systems.



3. Explore high-risk, long-range applications of control to areas such as nanotechnology, quantum mechanics, electromagnetics, biology, and environmental science.



4. Maintain support for theory and interaction with mathematics, broadly interpreted.



5. Invest in new approaches to education and outreach for the dissemination of control concepts and tools to non-traditional audiences.

Some Important Trends in Control in the Last Decade

(Online) Optimization-based control

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

Layering and architectures

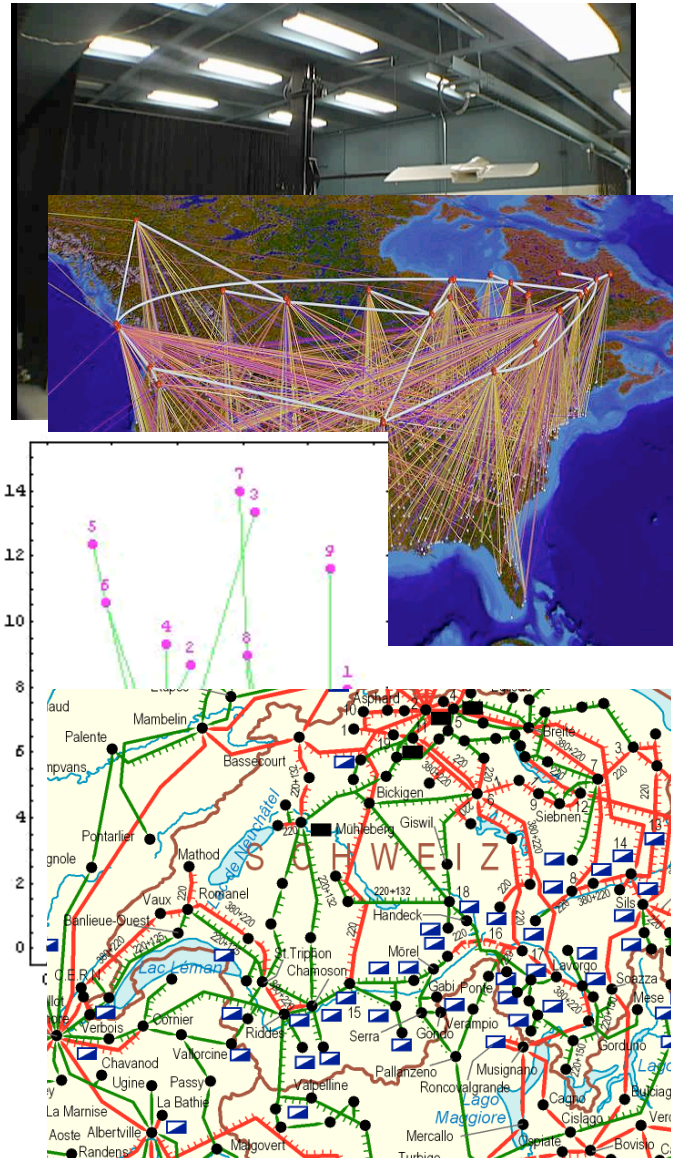
- Command & control at multiple levels of abstraction
- Modularity in product families via layers

Formal methods for analysis, design and synthesis

- Combinations of continuous and discrete systems
- Formal methods from computer science, adapted for hybrid systems (mixed continuous & discrete states)

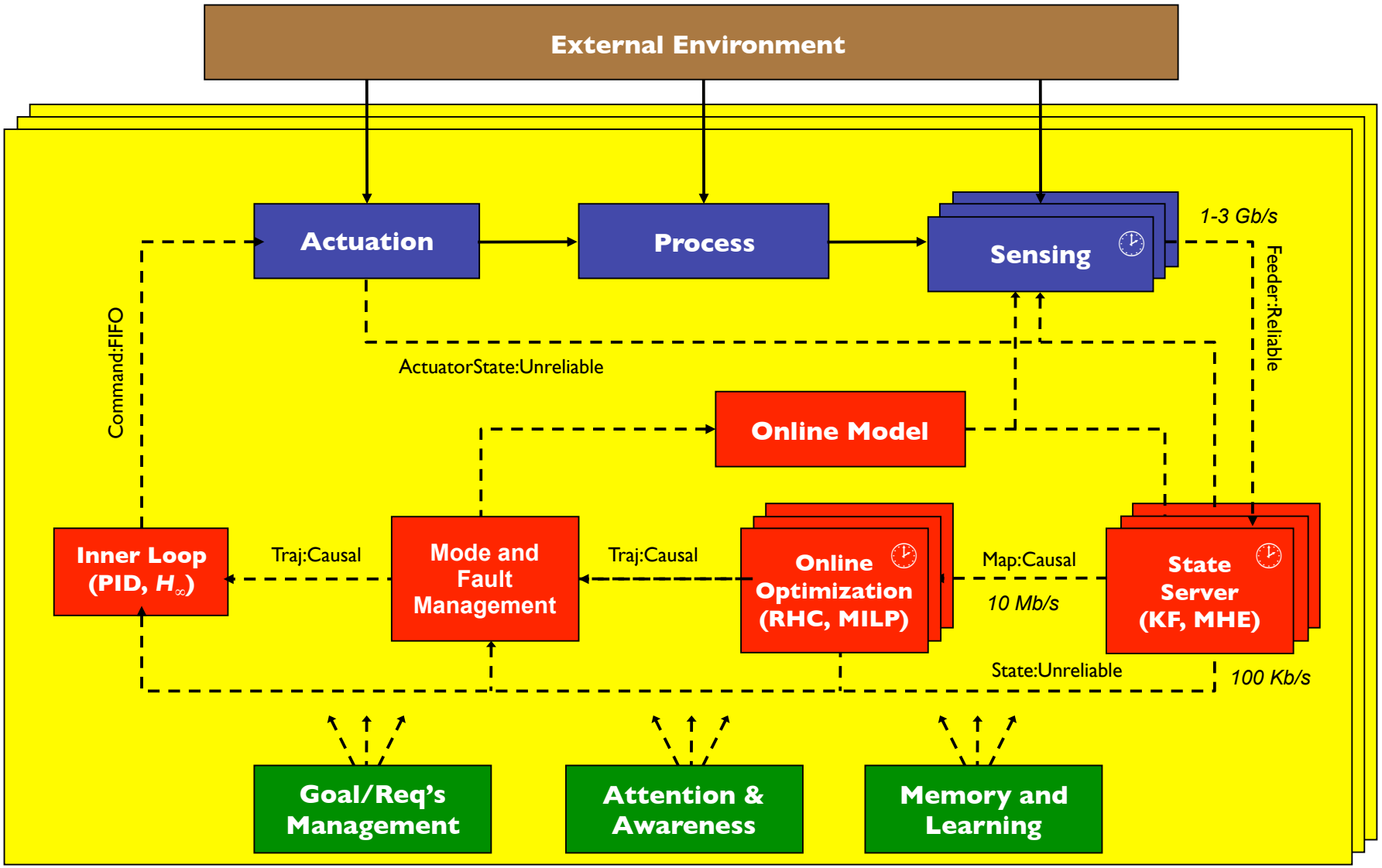
Components → Systems → Enterprise

- Movement of control techniques from “inner loop” to “outer loop” to entire enterprise (eg, supply chains)
- Use of *systematic* modeling, analysis and synthesis techniques at all levels
- Integration of “software” with “controls” (Internet of things, cyber-physical systems, etc)

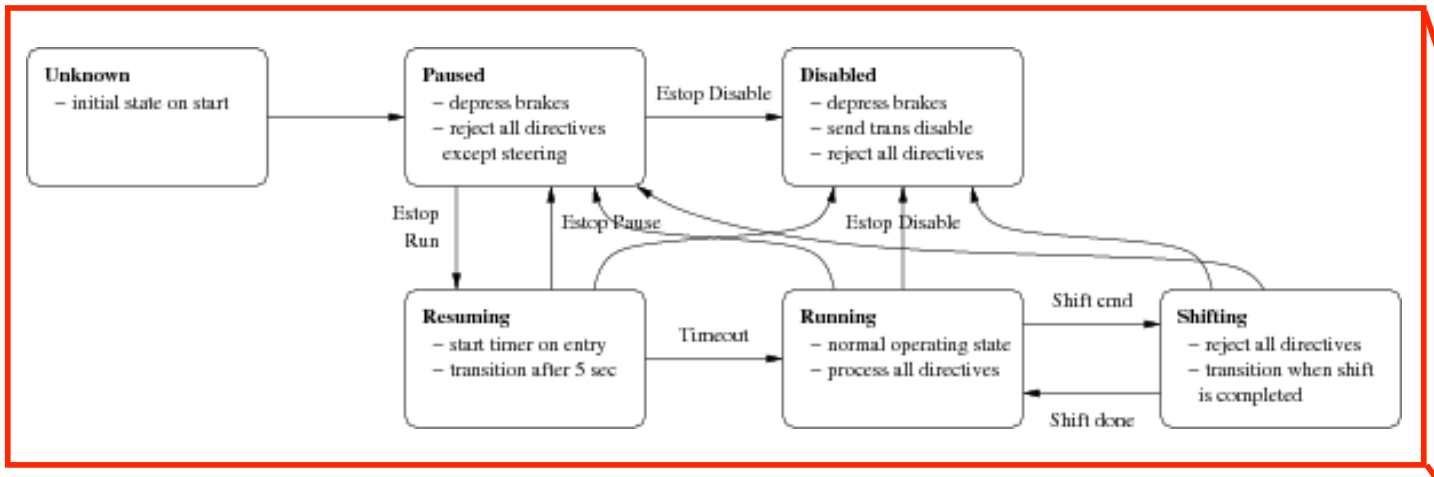


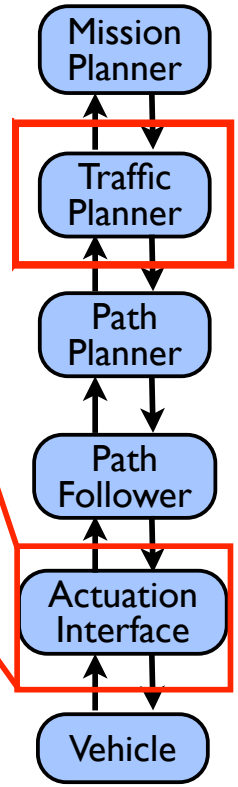
Modern Networked Control System Architecture

(following P. R. Kumar)



Challenge: Verification of Control Logic





Function: respond to control commands + DARPA pause/emergency stop

Verification using temporal logic (Lamport's TLC, TLA+)

- Model follower, Actuation Interface, DARPA, accModule, transModule in TLC
- Shared variables: *state, estop, acc, acc_command, trans, trans_command*

Verify the following properties

- $\Box((estop = \text{DISABLE}) \Rightarrow \Diamond\Box(state = \text{DISABLED} \wedge acc = -1))$
- $\Box((estop = \text{PAUSE}) \Rightarrow \Diamond(state = \text{PAUSED} \vee estop = \text{DISABLE}))$
- $\Box((estop = \text{RUN}) \Rightarrow \Diamond(state = \text{RUNNING} \vee state = \text{RESUMING}))$
- $\Box((state = \text{RESUMING}) \Rightarrow \Diamond(state = \text{RUNNING} \vee estop = \text{DISABLE} \vee estop = \text{PAUSE}))$
- $\Box((state \in \{\text{DISABLE, PAUSED, RESUMING, SHIFTING}\} \Rightarrow acc = -1)$

Lessons Learned from Alice

Online optimization solves nonlinear control problems

- Modern computation allows constrained optimization problems to be solved online
- Solutions exist for situations with more limited computation (multi-parametric optimization)

Layered control architectures allow more efficient design

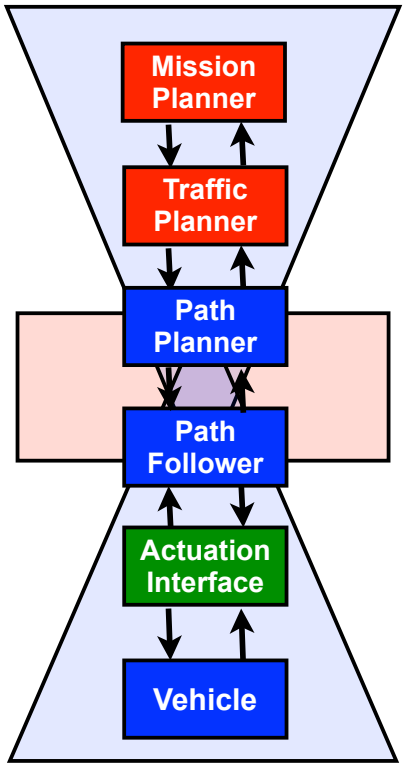
- Allows for "separation of concerns" between subsystems
- Provided a very modular design, capable of rapid (human-controlled) adaptation

Verification of control protocols is necessary, but hard

- Traditional methods of simulation and testing not sufficient
- Formal methods not easily applied to "hand designed" control protocols

New tools for "correct by construction" design are needed

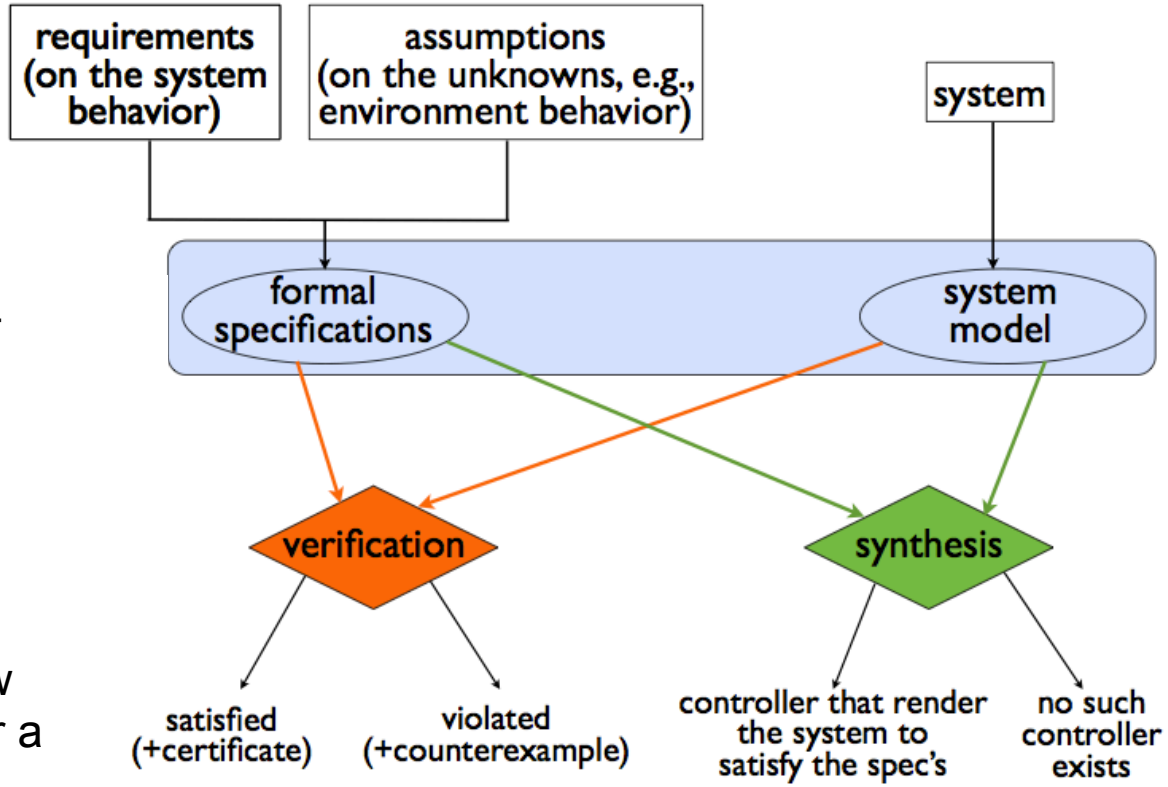
- Temporal logic(s) are powerful language for specifying desired behavior (combined with traditional measures)
- New tools are becoming available, but not yet ready for prime time



Formal Methods for System Verification & Synthesis

Specification using LTL

- Linear temporal logic (LTL) is a math'l language for describing linear-time prop's
- Provides a particularly useful set of operators for construc-ting LT properties without specifying sets



Methods for verifying an LTL specification

- *Theorem proving*: use formal logical manipulations to show that a property is satisfied for a given system model
- *Model checking*: explicitly check all possible executions of a system model and verify that each of them satisfies the formal specification

Methods for *synthesis* of correct-by-construction control protocols

- Build on results in logic synthesis and (recent) results in GR(1) synthesis
- Key challenges: dynamics, uncertainty, complexity

Hybrid, Multi-Agent System Description

Subsystem/agent dynamics - continuous

$$\dot{x}^i = f^i(x^i, \alpha^i, y^{\sim i}, u^i) \quad x^i \in \mathbb{R}^n, u^i \in \mathbb{R}^m$$
$$y^i = h^i(x^i, \alpha^i) \qquad\qquad y^i \in \mathbb{R}^q$$

Agent mode (or "role") - discrete

- $\alpha \in \mathcal{A}$ encodes internal state + relationship to current task
- Transition $\alpha' = r(x, \alpha)$

Communications graph $\mathcal{G}$

- Encodes the system information flow
- Neighbor set $\mathcal{N}^i(x, \alpha)$

Communications channel

- Communicated information can be lost, delayed, reordered; rate constraints

$$y^i_j[k] = \gamma y^i(t_k - \tau_j) \quad t_{k+1} - t_k > T_r$$

- γ = binary random process (packet loss)

Task

- Encode task as finite horizon optimal control + temporal logic (assume coupled)

$$J = \int_0^T L(x, \alpha, u)\, dt + V(x(T), \alpha(T)),$$

$$(\varphi_{init} \wedge \Box\varphi_e) \implies (\Box\varphi_s \wedge \Diamond\varphi_g)$$

Strategy

- Control action for individual agents

$$u^i = \gamma(x, \alpha) \qquad \{g^i_j(x, \alpha) : r^i_j(x, \alpha)\}$$

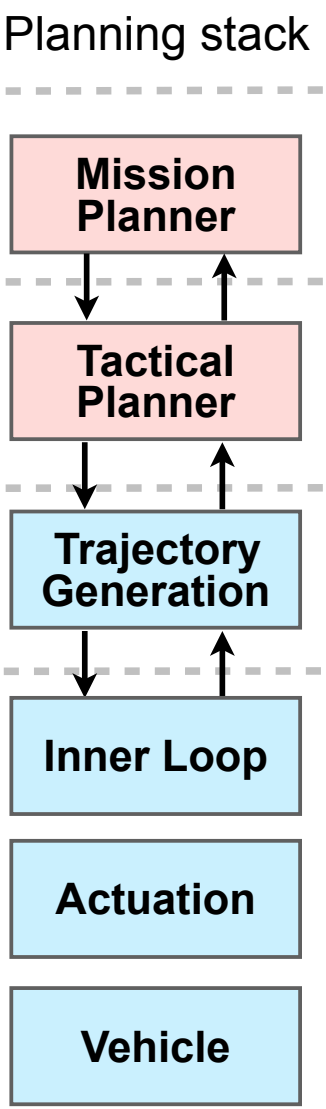
$$\alpha^{i\,\prime} = \begin{cases} r^i_j(x, \alpha) & g(x, \alpha) = \text{true} \\ \text{unchanged} & \text{otherwise.} \end{cases}$$

***Decentralized* strategy**

$$u^i(x, \alpha) = u^i(x^i, \alpha^i, y^{-i}, \alpha^{-i})$$
$$y^{-i} = \{y^{j_1}, \ldots, y^{j_{m_i}}\}$$
$$j_k \in \mathcal{N}^i \quad m_i = |\mathcal{N}^i|$$

- Similar structure for role update

Hierarchical, Networked Control Systems



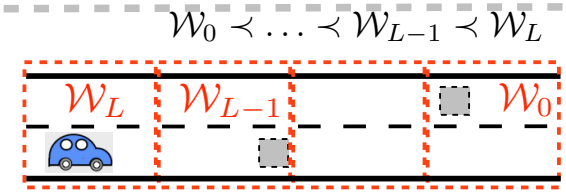
Different views

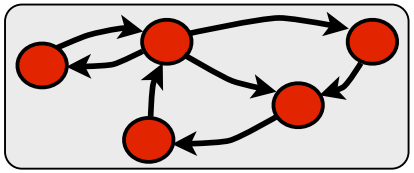
"long-horizon specification"

"short-horizon specification"

continuous dynamics& constraints

Multi-scale models





$$\min \int_{t_0}^{T} L(x,u)dt$$
$$\text{s.t.} \quad \dot{x} = f(x,u)$$
$$g(x,u) \leq 0$$

Implementation

- High level "mission" abstraction (iterative graph search)
- Reactive planner: decision-making based on events in environment
- Continuous control action based on "mode" and cost

Multi-layer approach

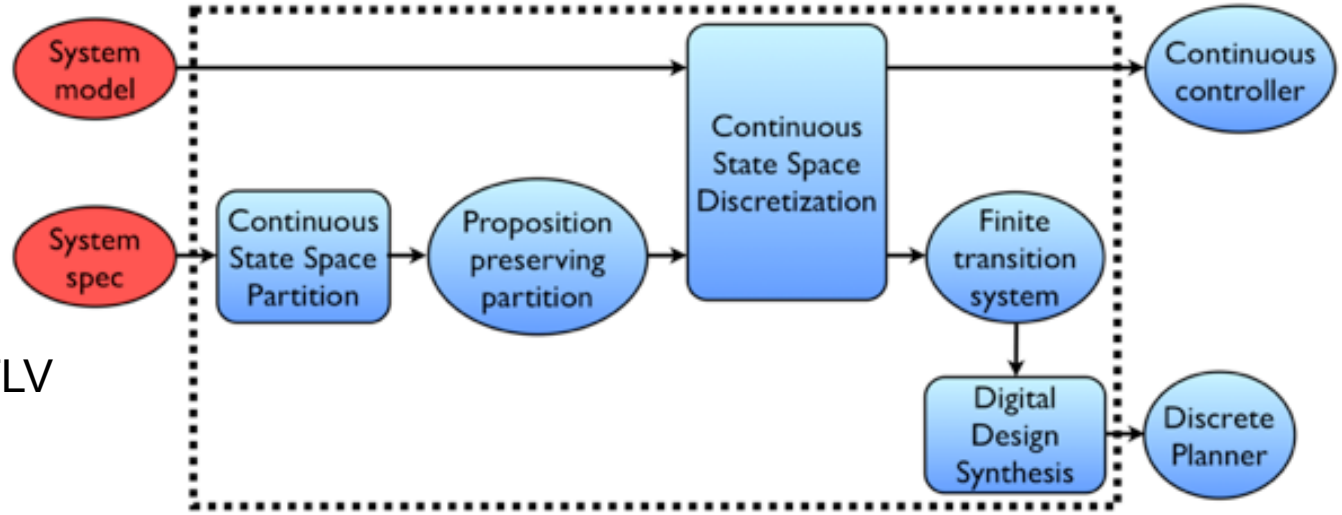
- Use optimal trajectory generation to create a discrete abstraction that captures the dynamics at a simplified level
- Reactive planner based on GR(1) synthesis (possibly RHC)
- High level planner sends specifications to reactive planer
- Online versus offline decisions at each level

Temporal Logic Planning (TuLiP) toolbox

http://tulip-control.sourceforge.net

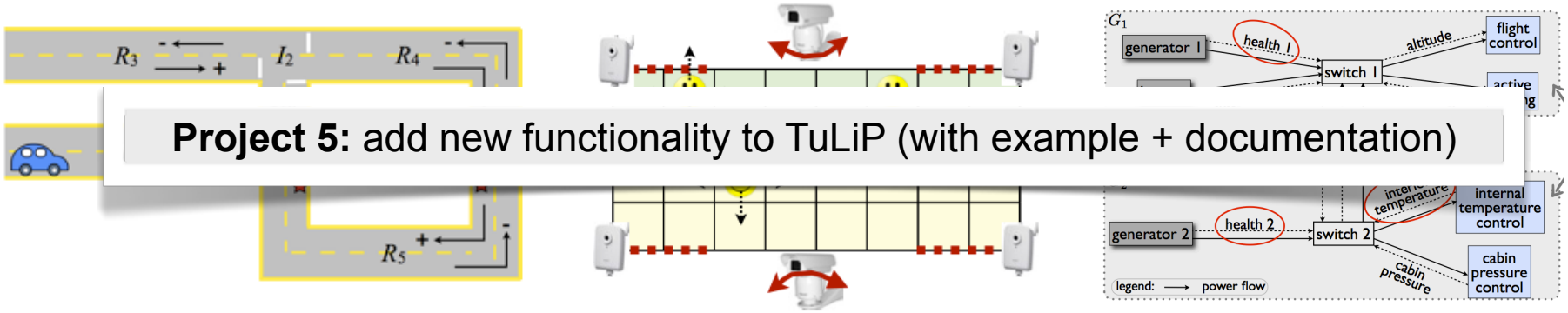
Python Toolbox

- GR(1), LTL specs
- Nonlin dynamics
- Supports discret-ization via MPT
- Control protocol designed using JTLV
- Receding horizon compatible



Applications of TuLiP in the last few years

- Autonomous vehicles - traffic planner (intersections and roads, with other vehicles)
- Distributed camera networks - cooperating cameras to track people in region
- Electric power transfer - fault-tolerant control of generator + switches + loads



Project 5: add new functionality to TuLiP (with example + documentation)

Open (Research) Issues

Optimality: "language-constrained, optimal trajectory generation"

$$(\varphi_{init} \wedge \Box\varphi_e) \implies (\Box\varphi_s \wedge \Diamond\varphi_g) \qquad J = \int_0^T L(x,\alpha,u)\,dt + V(x(T),\alpha(T)),$$

Partial order computation and hierarchical structure

- How do we determine the partial order for RHTLP and link to "supervisory" levels?

Verification and synthesis with (hard) real-time constraints

- How do we incorporate time in our specifications, verification and synthesis tools?
- Note: time automata and timed temporal logic formulas available...

Contract-based design: automate search interfaces for distributed synthesis

- How do we decompose a larger problem into smaller pieces?
- Especially important for large scale projects with multiple teams/companies

Uncertainty and robustness

- How to specify uncertainty for transition systems, robustness for controllers, specs
- New methods for describing robustness by Tabuada et al: look at how much the specifications must be enlarged to capture new behaviors based on uncertainty

Many other directions: incremental, probabilistic, performance metrics, ...

- Identify problems where knowledge of dynamics, uncertainty and feedback matter

Distributed Synthesis of Control Protocols

Problem statement

- Given a global spec $\varphi_e \rightarrow \varphi_s$ and an inter-connection structure, find local controllers to satisfy the spec.

Approach

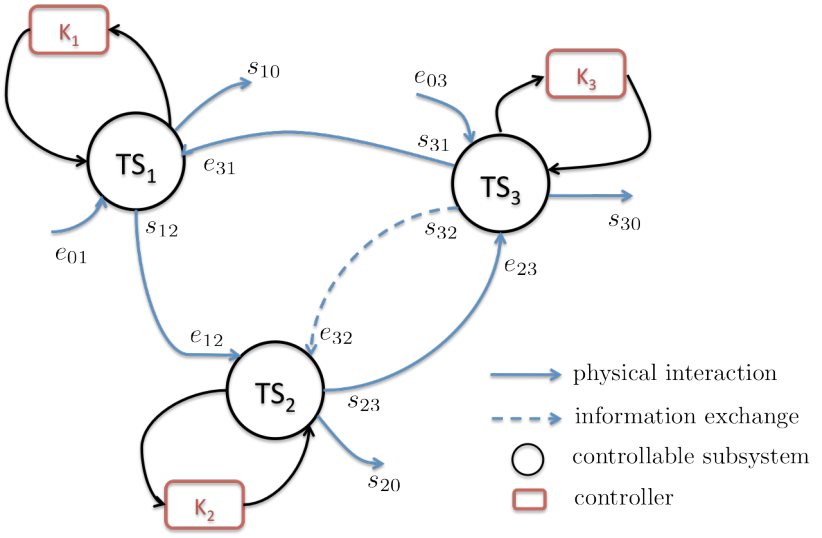
- Decompose the global spec into local ones $\varphi_{e_i} \rightarrow \varphi_{s_i}$ for each control unit such that

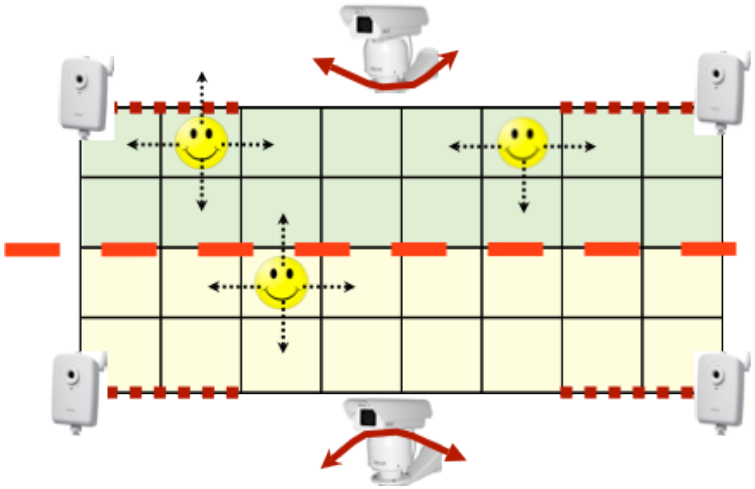
$$\bigwedge_i \varphi_{e_i} \rightarrow \varphi_e \rightarrow \varphi_s \rightarrow \bigwedge_i \varphi_{s_i}$$

- If the local specs satisfy certain conditions and there exist local controllers to satisfy the local specs, local implementations satisfy global spec!
- If the local specs are unrealizable
  - Refine the local specs:

$$(\phi_2' \wedge \varphi_{e_1}) \rightarrow (\varphi_{s_1} \wedge \phi_1)$$
$$(\phi_1' \wedge \varphi_{e_2}) \rightarrow (\varphi_{s_2} \wedge \phi_2)$$

  - Under certain conditions (e.g., to avoid circularity), realizability of the refined specs is sufficient for satisfying the global spec





Example: smart camera networks

Optimal Synthesis with Weighted Average Costs

Problem Setting

- Deterministic weighted transition system TS
- LTL specification ϕ
- $J(\sigma) := \limsup_{n\to\infty} \sum_{i=0}^{n} c(\sigma_i, \sigma_{i+1})$
- Problem: Compute run σ that minimizes J over all σ and satisfies ϕ.

Main Results

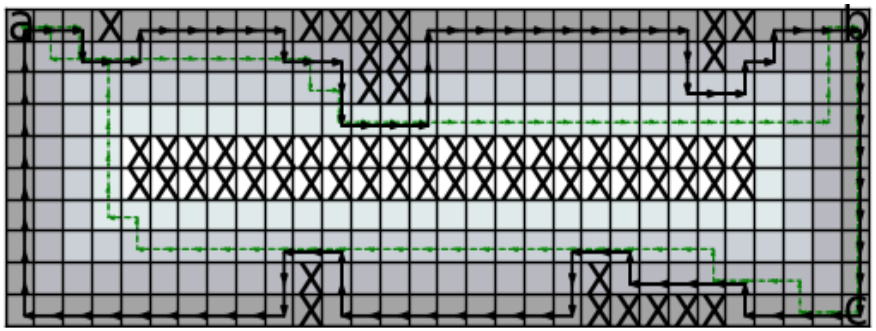
- Reduce problem to finding optimal cycle in product automaton P.
- Dynamic programming recurrence computes optimal cost cycle on P = (V,E). $F_k(v)$ is minimum cost walk of length k between vertices s, v in V.

$$F_k(v) = \min_{(u,v)\in E} [F_{k-1}(u) + c(u,v)]$$

- Complexity: $O(n_a(mn + n^2 \log(n))$ for 0-1 weights, where n_a is the number of accepting states.

Example

- Costs lower near boundary
- ϕ = □◊a ∧ □◊b ∧ □¬x
- Optimal (black) and feasible using DFS (green)



(shading represents cost)

Questions

- Nondeterministic transition system?
- Reactive environments?
- Multi-objective?
- Discounted cost function?

Markov Decision Processes with LTL Specifications

Problem Setting

- Markov decision process (MDP) system model, with uncertainty in transitions (disturbances, failures)
- LTL specification ϕ (probably GR(1))
- Problem: Maximize probability of MDP satisfying ϕ over uncertainty set:

$$\max_{\pi \in \Pi} \min_{\tau \in \mathcal{T}} \mathbb{P}^{\pi,\tau}(s_0 \models \varphi)$$

Main Results

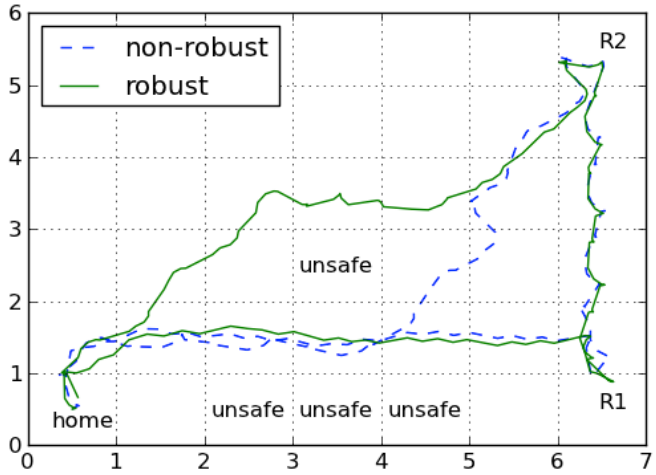
- Transform P = MDP x LTL to stochastic shortest path (SSP) form
- Compute satisfaction probabilities on SSP with robust dynamic program'g

$$(TJ)(s) := \min_{a \in A(s)} [c(s,a) + \max_{p \in \mathcal{P}_s^a} p^T J]$$

- Project policy π back to MDP
- Complexity: O(n2m log(1/ε)2) for ε-suboptimal policy

Example

- Differential drive robot (x,y,theta)
- Transition probabilities estimated (MC)
- ϕ = ◊(R1 ∧ ◊R2) ∧ □¬unsafe ∧ ◊□ home



Questions

- Simpler fragments of temporal logic?
- Tradeoffs between costs and probability of success?
- Principled abstraction of MDPs from continuous systems?

Livingston, Burdick, Murray
ICRA 2013

"Patching" Reactive Control Protocols

Reduce complexity via "patching"

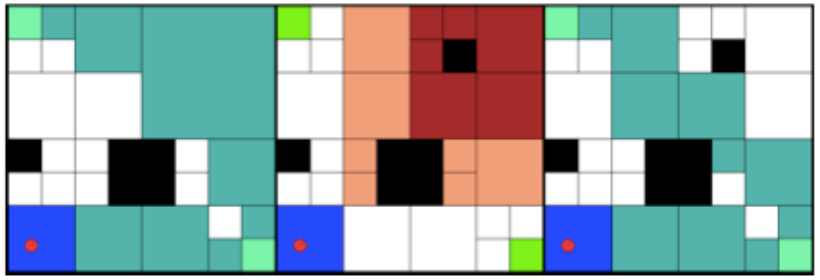
- Given an existing controller, create local "patch" when specs changes
- Avoids costly re-synthesis of full controller
- Can be used for "on-the-fly" synthesis and/or hierarchical synthesis

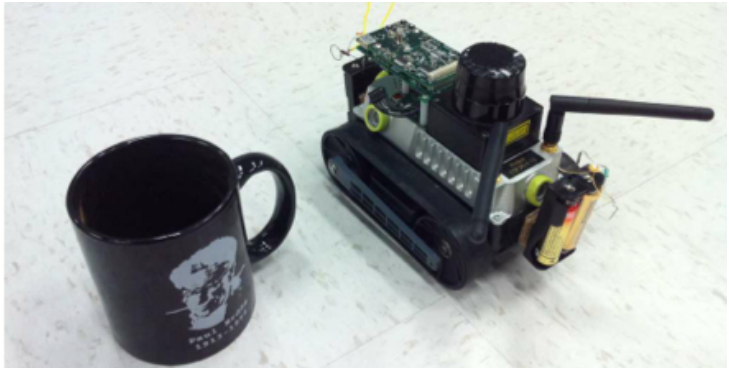
Technical approach: local μ calculus

- Abstract system dynamics as a discrete transition system (finite state automaton)
- Formulate synthesis problem as a μ calculus formula with GR(1) specification
- Given a change in specification, establish entry and exit goals for modified DTS
- Synthesize a "local" controller that connects entry and exit nodes and satisfies spec

Work to date

- Algorithms written and available in TuLiP
- Prototype implementation using Landroids

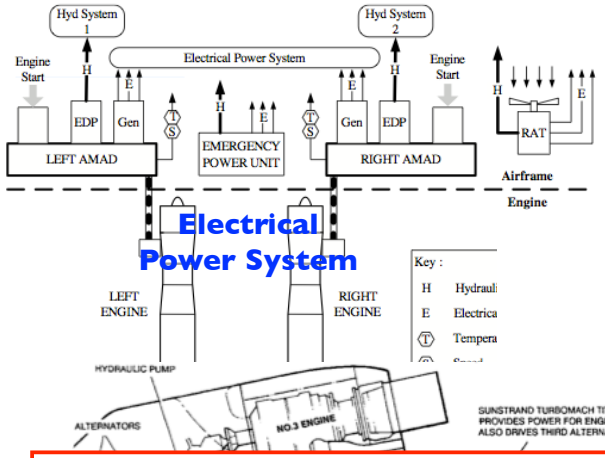




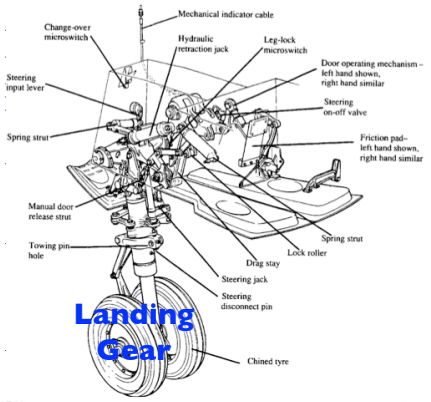
Technical Challenges and Risks

1. **Writing LTL (or other temporal logic) specifications is not a job for mortals**
   - Easy to make mistakes when writing LTL and hard to interpret complex formulas
   - Possible approach: domain specific tools that provide engineer-friendly interface
2. **Model checking and logic synthesis tools won't work on large problems**
   - Combinatorial explosion in discrete states for modest engineering systems will make it impossible to apply model checking/synthesis to "raw" problem
   - Approaches: abstraction layers and modularity via interfaces
     - Vertical layering: apply tools to different layers and enforce bisimulation
     - Horizontal contracts: define formal subsystem interfaces & reason about them
3. **Expertise in modeling and specification not yet developed**
   - Engineers in domains in which these tools are needed don't yet have experience developing models that ignore the right sets of things
     - Compare to reduced order models for aircraft (aerodynamic, aeroelastic) and agreed on specifications (bandwidth, response time, stability margins, etc)
     - Particularly worried about dynamics, uncertainty, interconnection
     - How do we convince FAA to allow use of these tools?
   - Approach (?): explore application domains, moving from modest to complex problems, and develop expertise, tools, tool chains, processes, ...

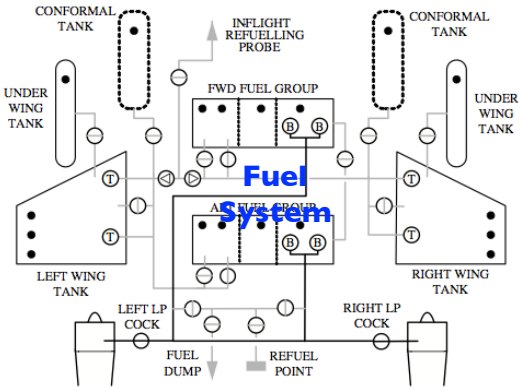
Aircraft Vehicle Management Systems



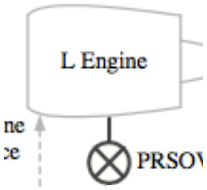
Electrical Power System



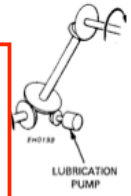
Landing Gear

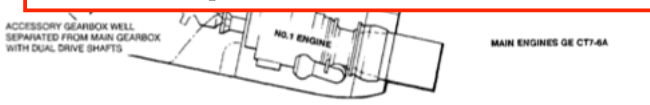


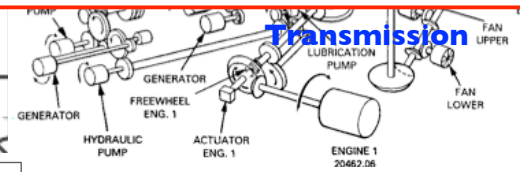
Fuel System



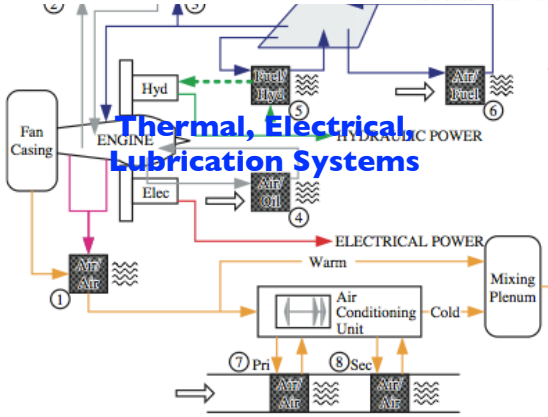
How do we design control logic for subsystems to insure safe operation across all operating conditions (w/ failures)



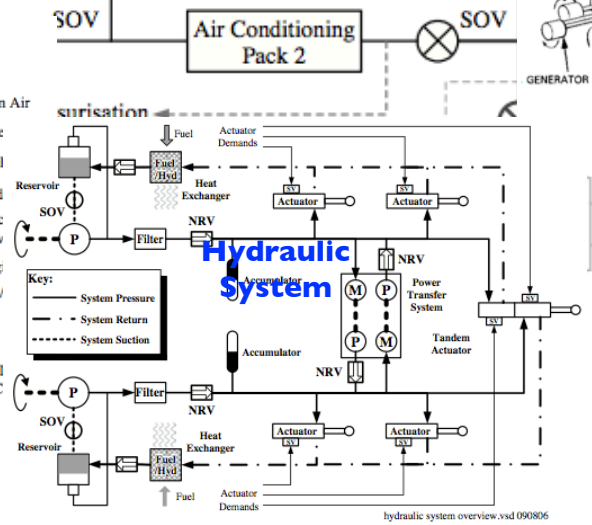




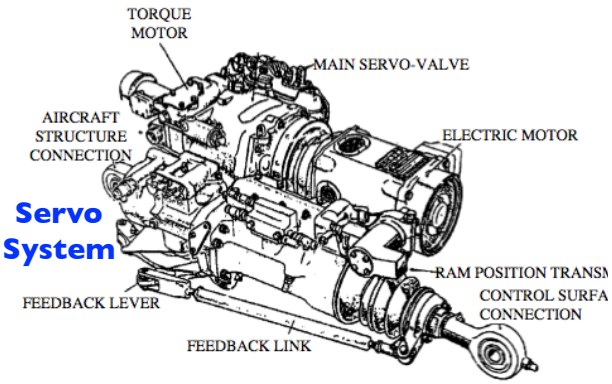
Transmission



Thermal, Electrical, Lubrication Systems

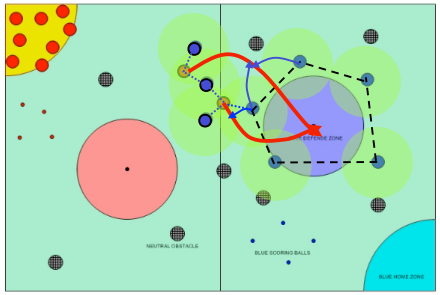


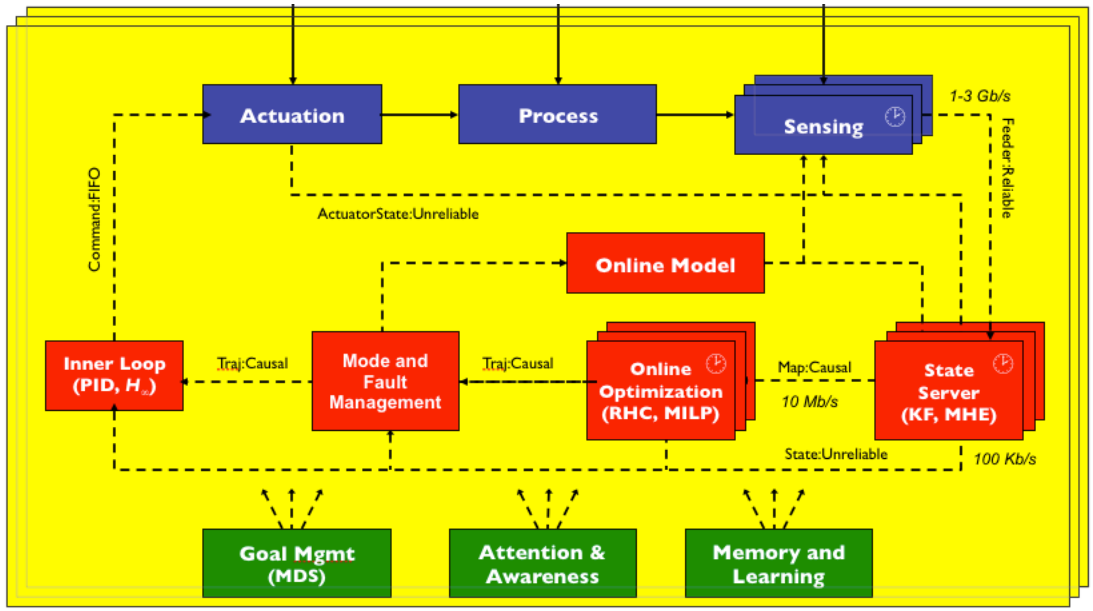
Hydraulic System



Servo System

Specification, Design and Verification of NCS





Specification

- How do we describe correct behavior?

Design

- What tools can we use to design protocols to implement that behavior?

Verification

- How do we know if it is actually correct?

$$J = \int_0^T L(x, \alpha, u)\, dt + V(x(T), \alpha(T)),$$

$$(\varphi_{init} \wedge \Box\varphi_e) \implies (\Box\varphi_s \wedge \Diamond\varphi_g)$$