Networked and Embedded Control Systems

Jerome Le Ny

ESE 680, Spring 2011

Upenn

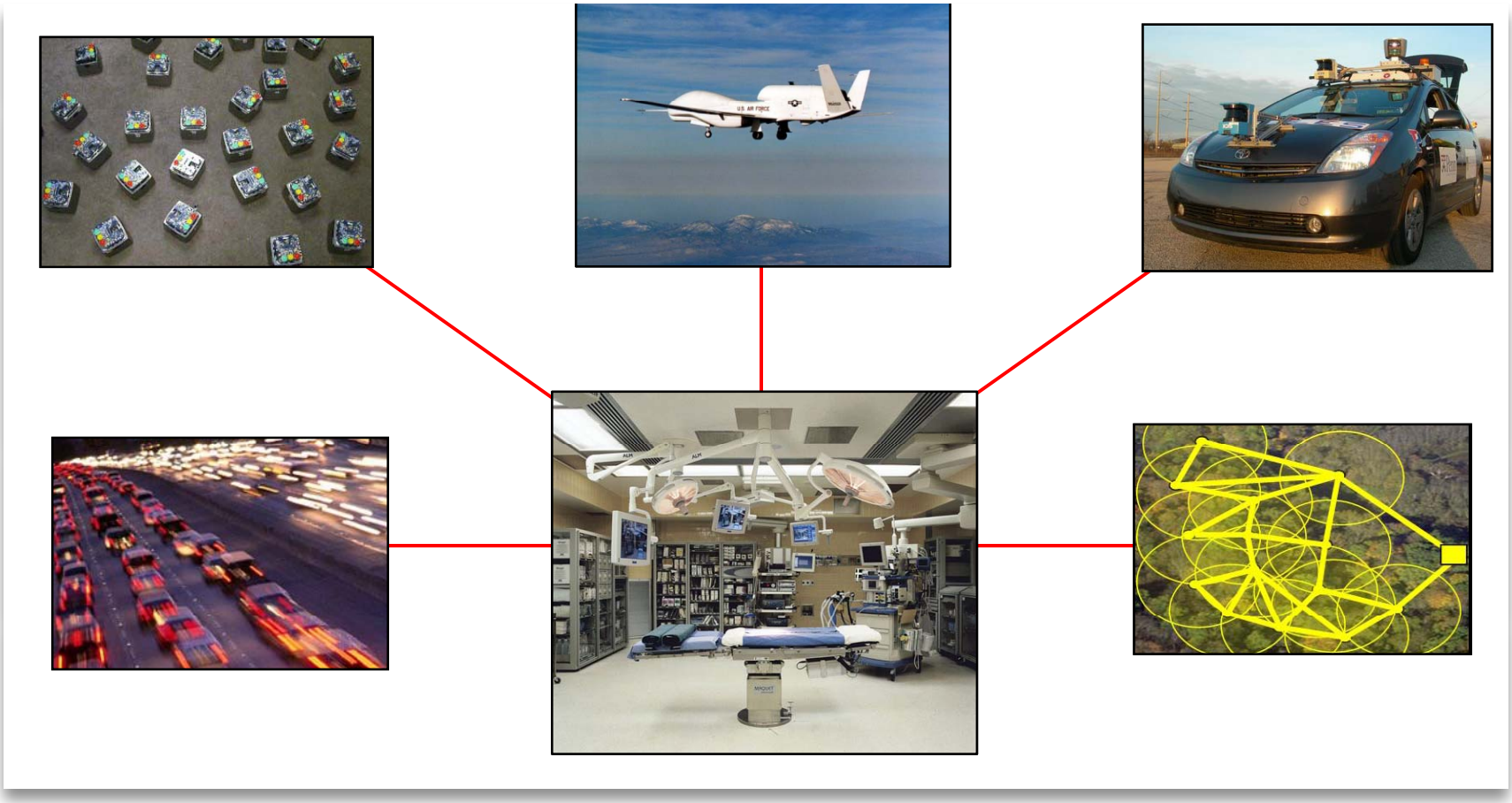
Outline

- Motivation and Examples
- Introduction to NECS Issues
- Administrative Stuff

Motivation and Examples

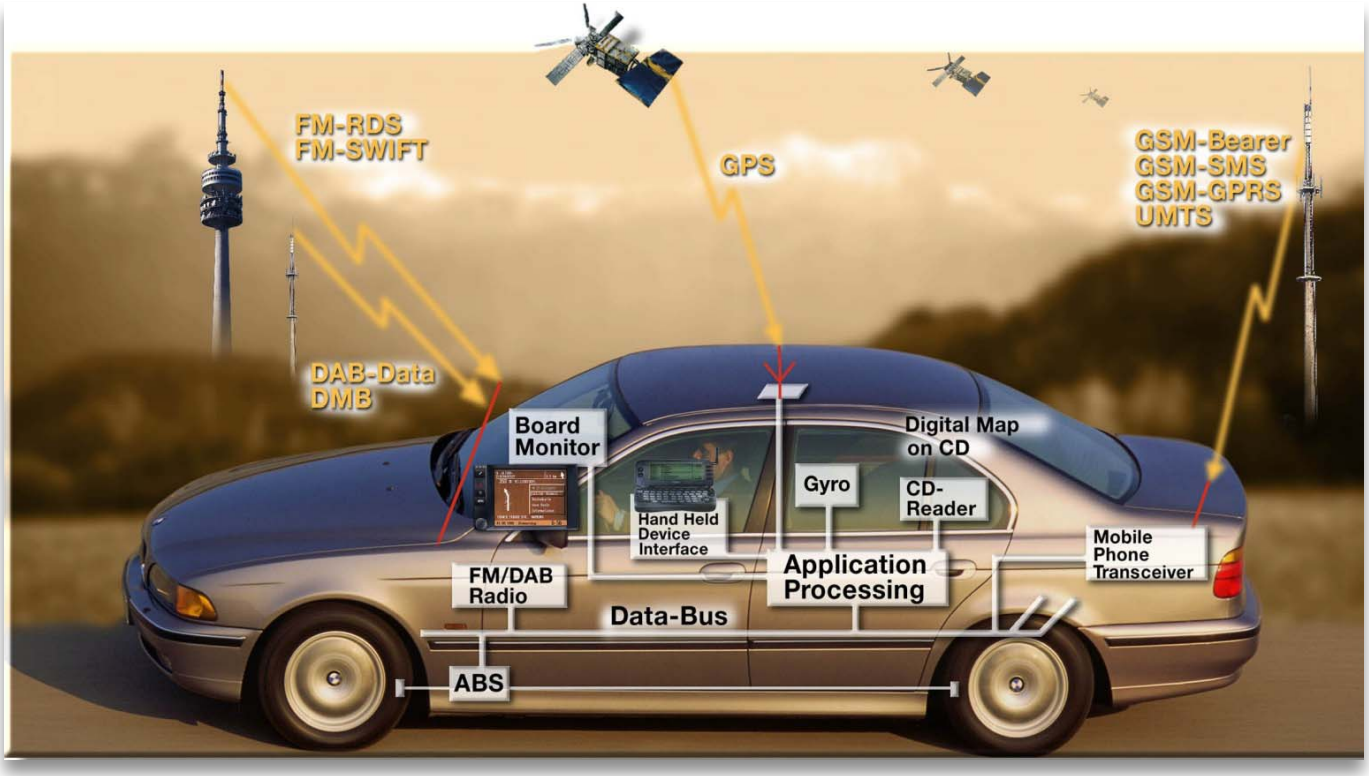
Cyber-Physical Systems (CPS)

Information/Computing Systems Interacting with physical systems, broadly



Increasingly integrated systems: sharing of sensing, actuation, computation and communication resources across distributed systems

Networked and Embedded Control Systems (NECS, NCS)



Modern cars: 40-100 networked microprocessors (brakes, ESC, transmission, engine, safety, climate, emissions, multimedia, …)

Several CAN and other buses

Boeing 777: 1280 networked microprocessors

2-5 million lines of code

good design Principles?

Embedded Control Characteristics

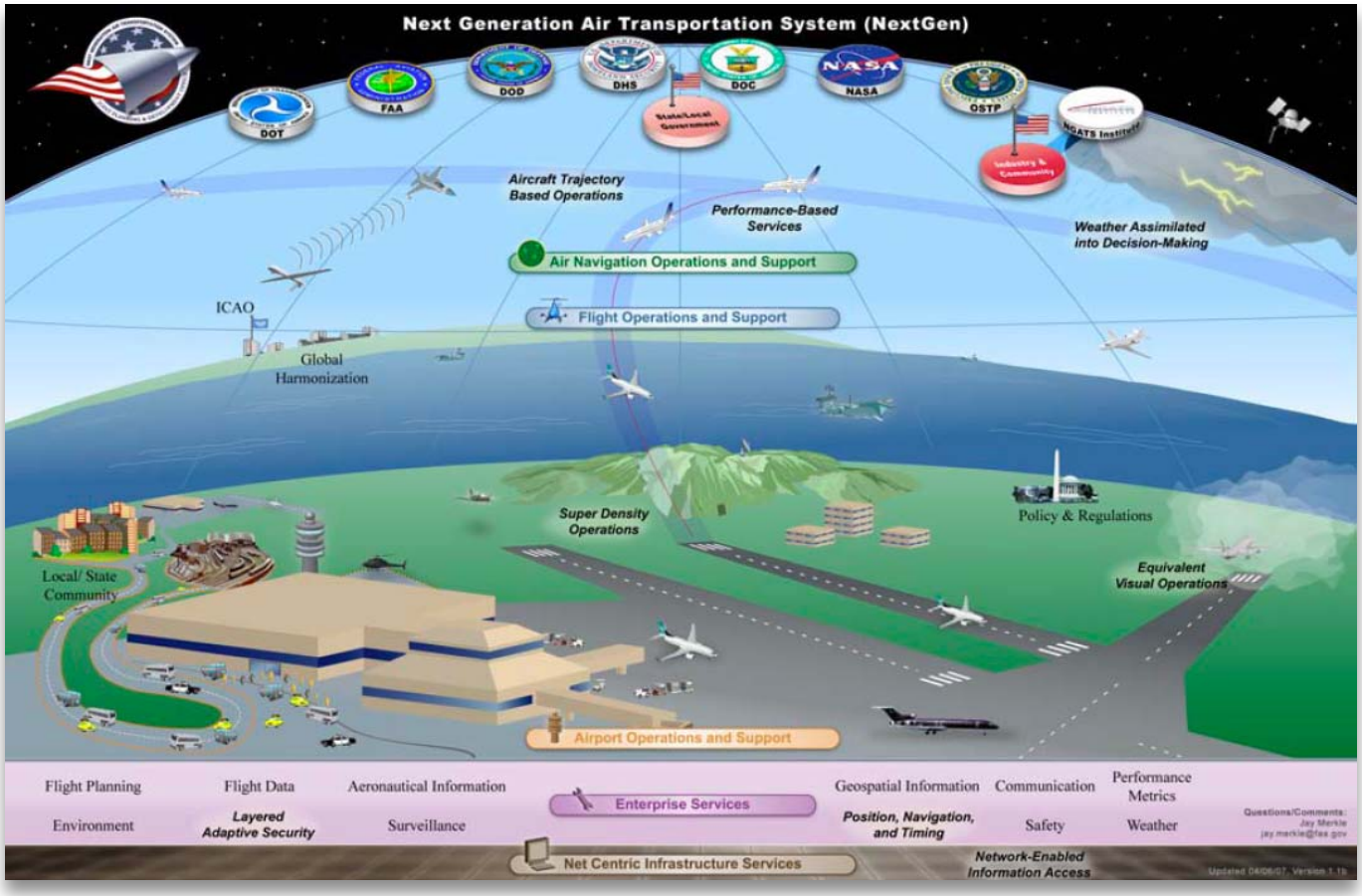
- Dependability
  - Safety (stability), reliability, security
  - E.g.: flight control or cruise control in cars
  - Certification issues, e.g. DO 178-B in aerospace. Formal guarantees?
    - currently most certification standards are process based
- Performance guarantees (e.g. standard control specifications)

vs

- Cost constraints
  - Often mass-market products, e.g. cars
  - Associated computation, communication, memory, energy constraints…
  - Flexibility, ease and speed of development (COTS components, code reuse). Better system integration methods critical for future complex system development
  - Maintainability is very important, might need to handle evolutions and upgrades easily and safely

Similar Issues at all Scales...

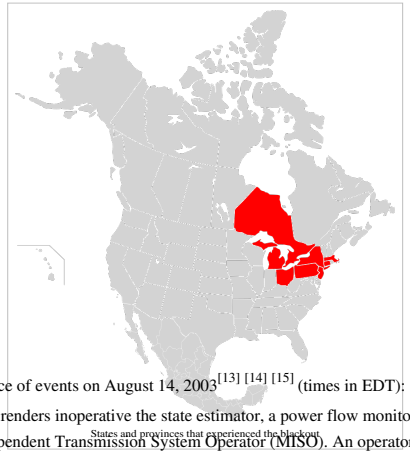
-Safety
-Performance
-Sensing and Control
-Network effects
-Distributed information
-Humans in the loop
-Heterogeneous components
-Certification issues, system of systems
-gradual evolution



Some Mishaps…

Northeast Blackout of 2003

The **Northeast Blackout of 2003** was a massive widespread power outage that occurred throughout parts of the Northeastern and Midwestern United States and Ontario, Canada on Thursday, August 14, 2003, at approximately 4:11 p.m. EDT (UTC-04) . At the time, it was the second most widespread electrical blackout in history, after the 1999 Southern Brazil blackout.[1] [2] The blackout affected an estimated 10 million people in Ontario and 45 million people in eight U.S. states.



States and provinces that experienced the blackout

Sequence of events

The following is the blackout's sequence of events on August 14, 2003[13] [14] [15] (times in EDT):

- 12:15 p.m. Incorrect telemetry data renders inoperative the state estimator, a power flow monitoring tool operated by the Indiana-based Midwest Independent Transmission System Operator (MISO). An operator corrects the telemetry problem but forgets to restart the monitoring tool.
- 1:31 p.m. The Eastlake, Ohio generating plant shuts down. The plant is owned by FirstEnergy, an Akron, Ohio-based company that had experienced extensive recent maintenance problems.
- 2:02 p.m. The first of several 345 kV overhead transmission lines in northeast Ohio fails due to contact with a tree in Walton Hills, Ohio.[16] [17]
- 2:14 p.m. An alarm system fails at FirstEnergy's control room and is not repaired.
- 3:05 p.m. A 345 kV transmission line known as the Chamberlain-Harding line fails in Parma, south of Cleveland, due to a tree.
- 3:17 p.m. Voltage dips temporarily on the Ohio portion of the grid. Controllers take no action.
- 3:32 p.m. Power shifted by the first failure onto another 345 kV power line, the Hanna-Juniper interconnection, causes it to sag into a tree, bringing it offline as well. While MISO and FirstEnergy controllers concentrate on understanding the failures, they fail to inform system controllers in nearby states.
- 3:39 p.m. A FirstEnergy 138 kV line fails in northern Ohio.[18]

Toyota Announces Voluntary Recall on 2010 Model-Year Prius and 2010 Lexus HS 250h Vehicles to Update ABS Software

Click here for FAQs About the 2010 Prius/2010 Lexus HS 250h/Camry Voluntary Recalls

Inspection of Power Steering Hose Position on Certain 2010 Camry Also Announced

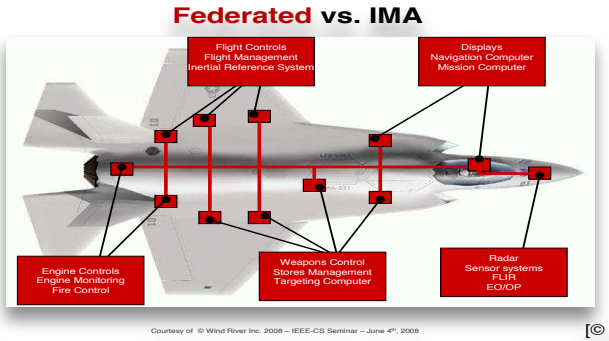
Recalls Underscore Toyota's Commitment to Address All Vehicle Quality and Safety Issues Promptly and Effectively

TORRANCE, Calif., February 8, 2010 – Toyota Motor Sales (TMS), U.S.A., Inc, today announced it will conduct a voluntary safety recall on approximately 133,000 2010 Model Year Prius vehicles and 14,500 Lexus Division 2010 HS 250h vehicles to update software in the vehicle's anti-lock brake system (ABS). No other Toyota, Lexus, or Scion vehicles are involved in this recall.

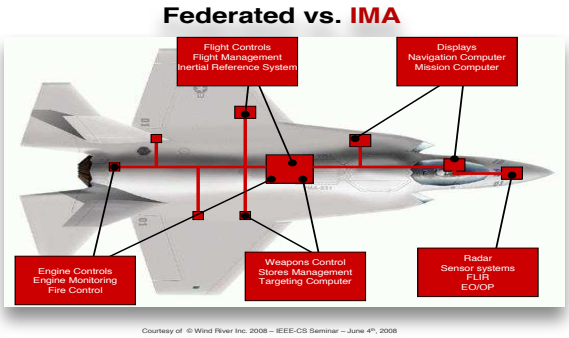
The ABS, in normal operation, engages and disengages rapidly (many times per second) as the control system senses and reacts to tire slippage. Some 2010 model year Prius and 2010 HS 250h owners have reported experiencing inconsistent brake feel during slow and steady application of brakes on rough or slick road surfaces when the ABS is activated in an effort to maintain tire traction.

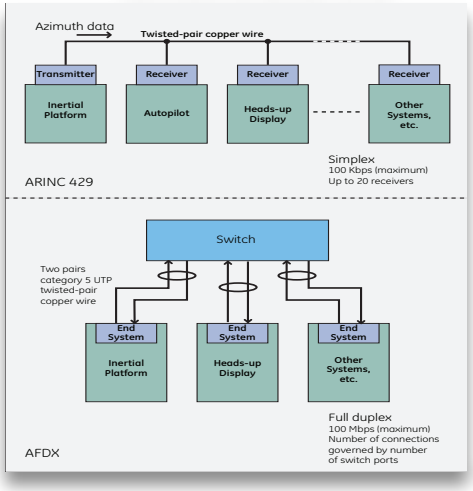
Bugs in software, but also in specifications! Particularly problematic because many CPS are safety-critical

Trends in NECS

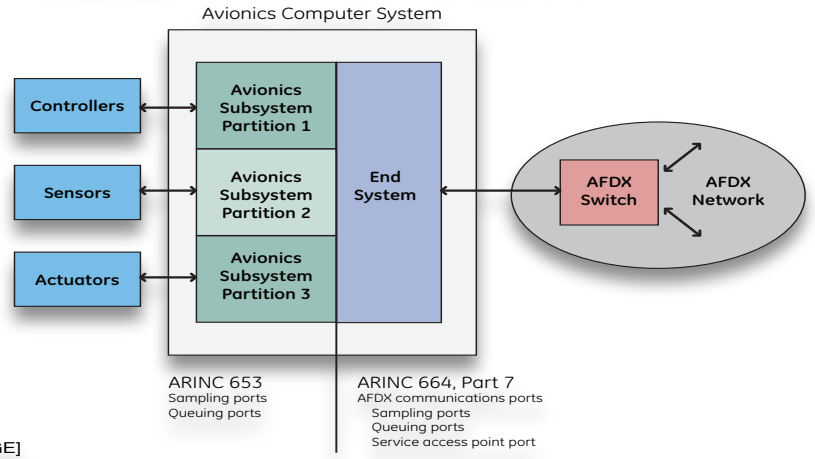


[© Wind River]





[© GE]



Attempt at Controlling Growing System Complexity, while
- Properly managing resources (computational, communication)
- Guaranteeing timings: important for control

Trends in NECS II

- Toward rigorous Model Based Engineering vs. process based certification
  - e.g. DO 178B replacement
  - ultimate goal: want to reason about high-level models, then generate correct controllers and code from these models
  - introduce more formal methods

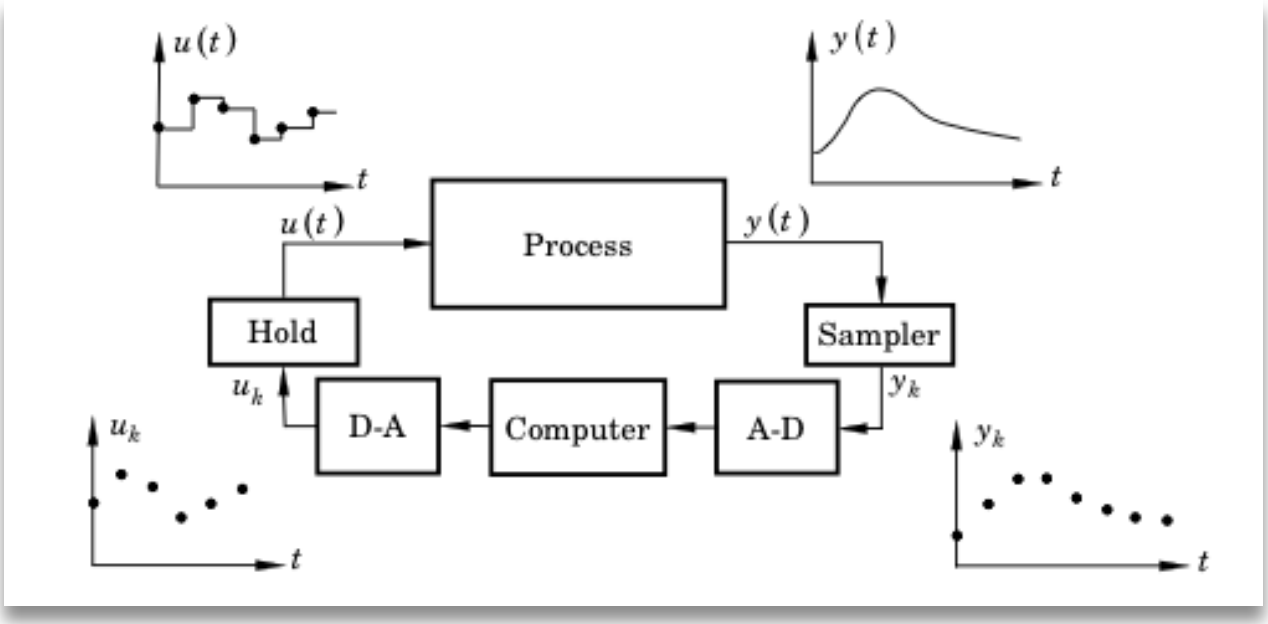
- Compositionality?
  - e.g. needed to add and remove small generators in electric grid
  - component reusability to reduce development costs
  - modular design with isolated components to avoid recertification during incremental changes

Introduction to NECS Issues

Sampled-Data Systems

- Classical theory of sampled-data systems is a good starting point
- Three controller design approaches:
  - discretize a continuous-time design
  - discrete-time design, neglect intersample behavior in synthesis
  - direct sampled-data design (lifting) - most rigorous but involved theory

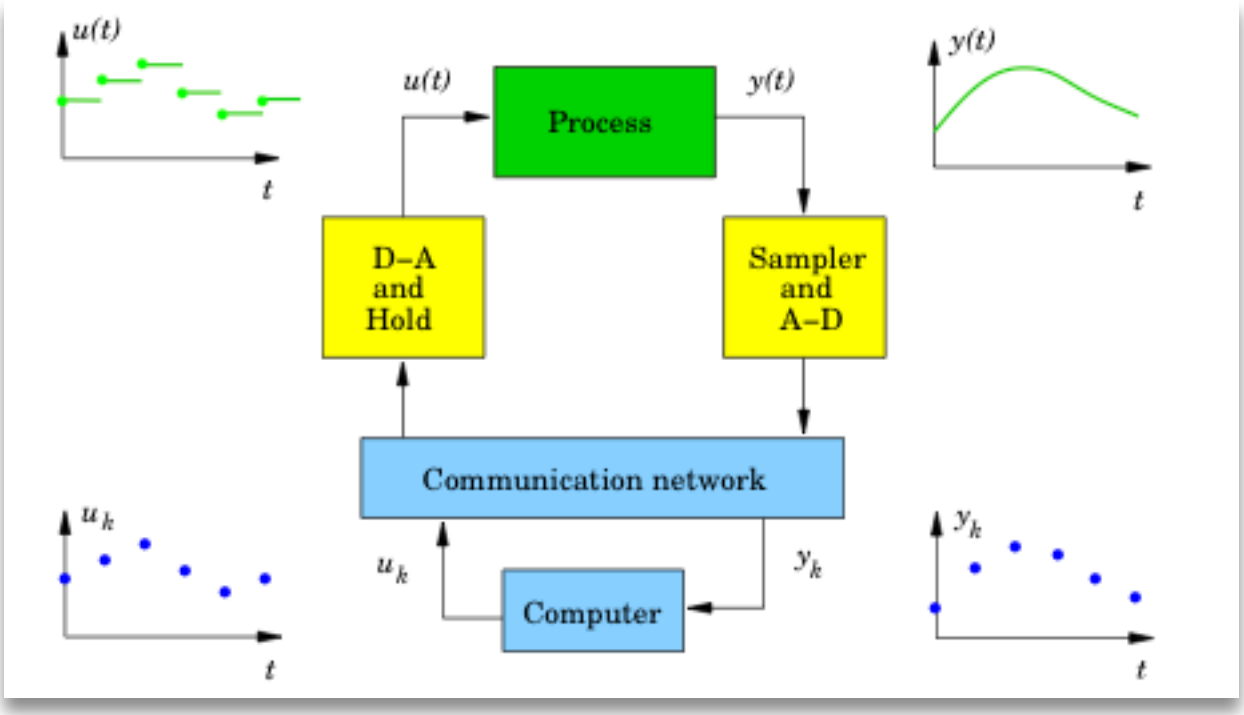
then simulate design for potential issues



[©K.-E. Arzen, Lund]

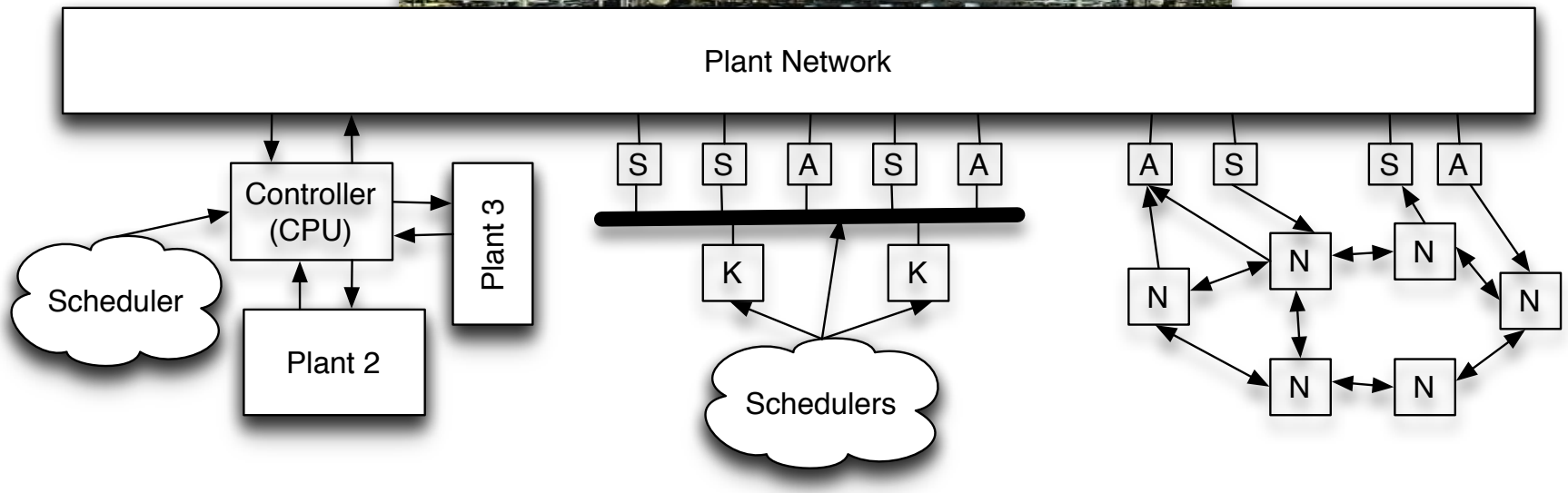
Networked Control Systems

- Adds delay, jitter and possibly packets losses
- Managing access to the communication network
- Focus of much of the current research literature in NCS
- Most models still very simplistic



[©K.-E. Arzen, Lund]

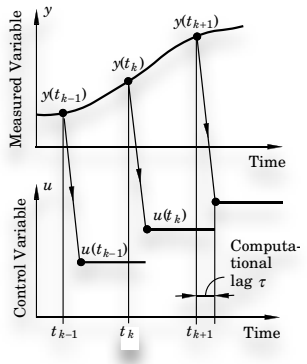
Many Implementation Choices



Real-Time Embedded Control Systems

classical controller timing

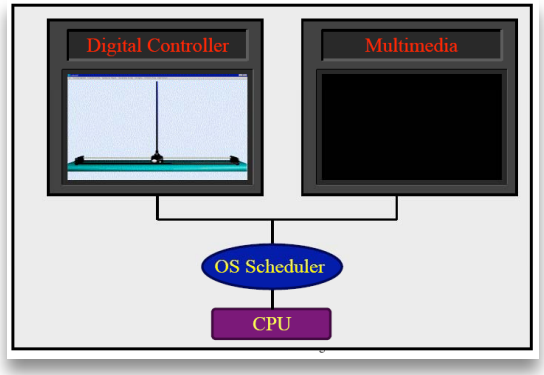
[©K.-E. Arzen, Lund]



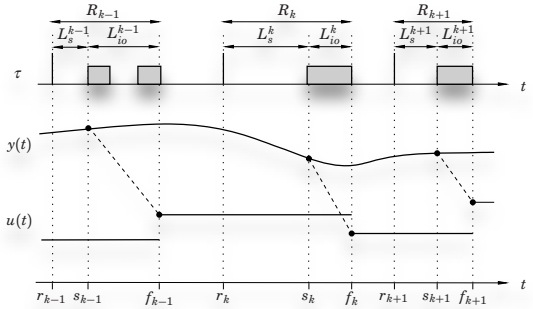
- Output $y(t)$ sampled periodically at time instants $t_k = kh$
- Control $u(t)$ generated after short and constant time delay τ

- system dedicated to control task
- timing very predictable
- costly, little flexibility
- software and hardware becomes unmanageable as system becomes more complex

CPU time management using a Real-Time Operating System (RTOS)



Controller timing with RTOS scheduler



[©K.-E. Arzen, Lund]

- Control task τ released periodically at time instances $r_k = kh$
- Output $y(t)$ sampled after time-varying **sampling latency** L_s
- Control $u(t)$ generated after time-varying **input-output latency** L_{io}

Increased flexibility, but delays and non-determinism introduced by RTOS and network. Impact on control performance?

Control System Specifications

- Stability
- Analog Performance: transients and steady-state
- Safety, absence of deadlocks
- Liveness
- Derive Specifications
  - Which correspond to what is intuitively expected from the system
  - Which are consistent (object of this course), backed by formal analysis, formal specification language
- Develop system and software from them

for systems including both analog signals and switching logic: hybrid systems

About this course

- We will explore some of these issues
  - model based analysis and synthesis of digital controllers
  - understand modern implementation issues: computation and communication resource management, impact on control performance
  - discuss complex specifications, certification and verification issues: formal methods (model checking, deductive methods)
- This is a topics course: I'm here to help you learn, but what you will get out of it will mostly depend on your personal involvement in the subject
  - currently active research topics. We are far from a mature theory of system design
  - You should critique and challenge the models we will discuss
  - complexity and variety of issues make this subject interesting to researchers with diverse backgrounds and interests: control engineering, real-time systems, logic and formal methods, etc.

Course Outline

- Review of classical theory
  - System modeling: continuous-time and discrete-time systems, automata including hybrid automata (2 lectures)
  - Sampling and Sampled-Data Systems (2 lectures) --> discretizations of CT controllers
  - Overview of 2 other classical digital control design: discrete methods and direct SD design (3-4 lectures)
- NECS issues:
  - Examples of communication standards for control (FlexRay, ARINC 664). Intro to real-time scheduling (1-2 lectures)
  - Control for systems with delays and sampling jitter via Lyapunov methods (2+ lectures)
  - Input-output methods for NECS (2+ lectures). Passivity and wave variables for networked systems, relevant integral quadratic constraints
- Formal methods for verification and design (after Spring break):
  - Discrete system abstractions: exact and approximate bisimulations
  - Model checking
  - Intro to deductive methods: Hoare logic for controller verification, hybrid dynamic logic
- Additional lectures
  - external speakers: AADL (Oleg Sokolsky), possibly more.
  - Student lectures.

Administrative Stuff

Grading

- Prepare a lecture (40%)
  - Review e.g. 2-3 related papers (choice guided by me).
  - Prepare lecture notes.
  - Lecture format free: slides or blackboard.
- Project (40-50%)
  - Design a control system, and do an analysis/simulation as realistic as possible (including effect of communication protocol, real-time scheduler, etc.). Most realistic: implement control algorithm on a microcontroller, w/ physical process possibly simulated (if have access to hardware and know how to program a RT system. Physical implementation not required).
  - or explore a new design/analysis technique. Aim for a new contribution to the field of NECS.
  - No literature review accepted for this part (that's the previous point).
- Homework (0-10%)
  - I might give a few exercises during the term to check your understanding of the material.
- Participation ~10%
  - Influences grade subjectively in any case…
  - The course will be boring if I'm the only one speaking. Again, constructive criticism is strongly encouraged. View this as a research seminar.

Example of Topics for Student Lectures

- Decentralized Control: decentralized fixed modes, quadratic invariance, …
- Decentralized Estimation: distributed LMS, RMS, distributed Kalman filtering, …
- Switched systems and applications to NECS
- Event triggered sampling for control
- More advanced real-time scheduling for control
- Synchronous languages or other programming paradigms
- Other choices possible. To determine with me during first couple of weeks