ELE6953E : Cyber-Physical Systems and the Internet of Things

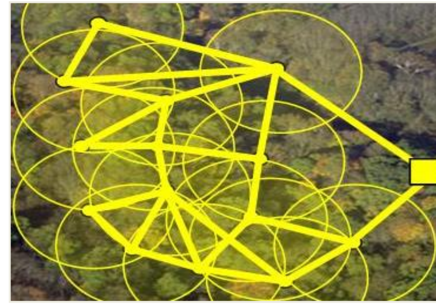
Lecture 1-1: Introduction and Motivation

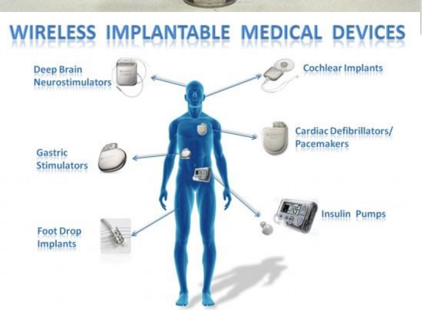
Jérôme Le Ny
Department of Electrical Engineering, Polytechnique Montreal

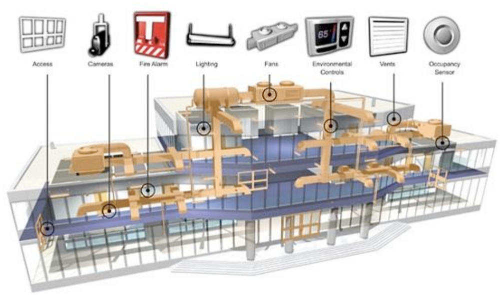
GERAD
GROUP FOR RESEARCH IN DECISION ANALYSIS

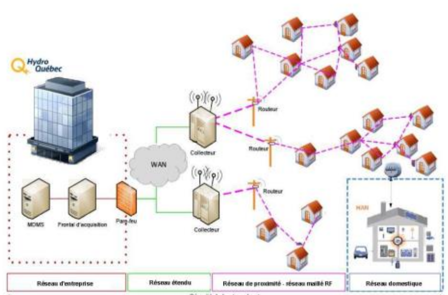
POLYTECHNIQUE MONTRÉAL

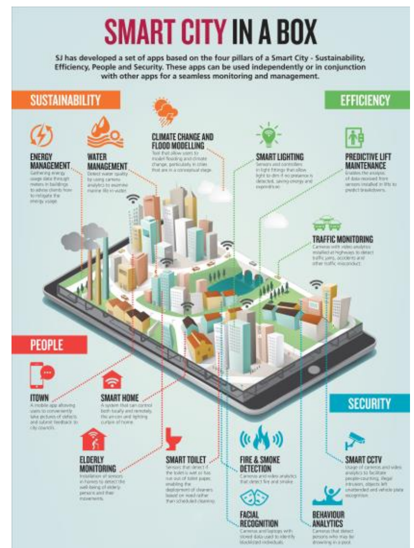
CYBER-PHYSICAL SYSTEMS











Robotics, health, smart buildings / grid / transportation / cities, …

Information systems interacting with physical systems

OUTLINE

- **Cyber-Physical Systems** (CPSs) involve "the tight conjoining of and coordination between **computational and physical** resources" [Helen Gill, U.S. National Science Foundation]
- Potentially very broad. We'll adopt a **control perspective** to study CPS, and emphasize **information systems** more broadly than computations
- How do we start thinking about/analyzing/designing such systems?

Themes of the course:

1. Modern Networked and Embedded Control Systems
2. Decentralized Control of Multi-Agent Systems
3. Next generation SCADA*/Distributed Monitoring & Control Systems
    1. IoT, Cloud Computing, Big Data, Stream Processing, etc. Software Tools
    2. Fault-detection, security
    3. Cyber-physical **human** systems: privacy, individual incentives, etc.

* Supervisory control and data acquisition

COURSE OBJECTIVES

- Some understanding of the current trends in CPS and IoT and their potential impact on the development of new automation (control networks, industrial DCS, robotic networks, smart X…)
- Discuss some illustrative examples of networked control systems
- Introduce some methods for the analysis and design of networked dynamical systems: NECS, decentralized estimation and control, optimization…
- Understand how cloud computing could be used to implement (parts of) large-scale control systems, in particular processing large volumes of streaming data
- Understand issues with large-scale CPS related to reliability, security & privacy

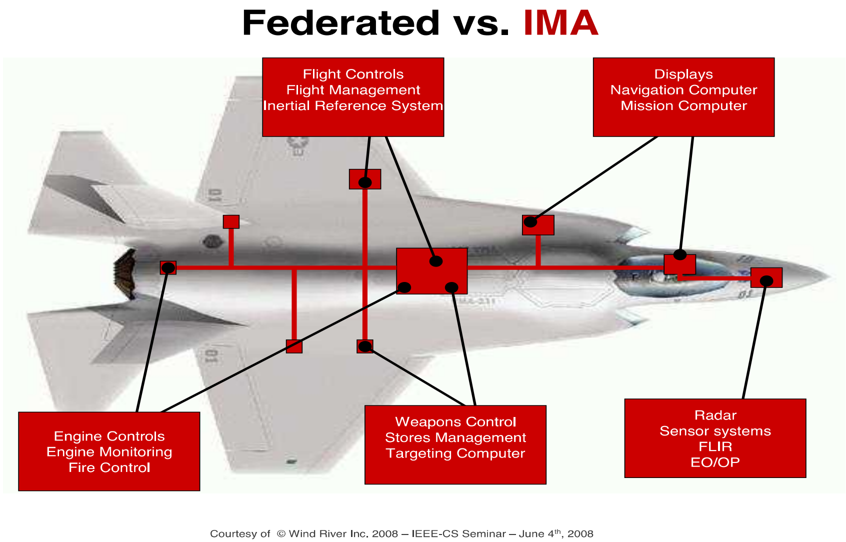
INSTRUCTOR AND STUDENTS

POLYTECHNIQUE MONTRÉAL

- Jerome Le Ny, Associate Professor, EE Department
- Contact: jerome.le-ny@polymtl.ca, office A.429-13
  http://www.professeurs.polymtl.ca/jerome.le-ny
- Technical expertise and research activities
  - Networked and embedded control systems
  - (Mobile) Robotics and autonomous systems
  - Decentralized control of multi-agent systems
  - Verification, certification, security, privacy issues associated with complex, large-scale monitoring and control systems

- About yourself:
  - Name, program, year, department
  - Background in control systems / embedded systems / maths, etc.?
  - Remarks/interest/experience related to this course?

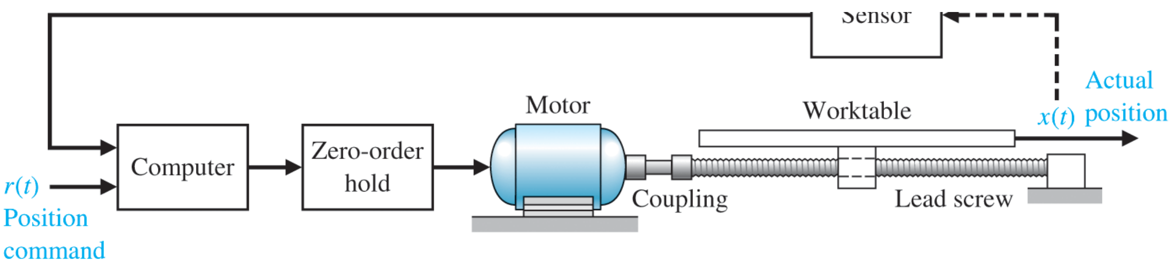
Networked and Embedded Control Systems (NECS)



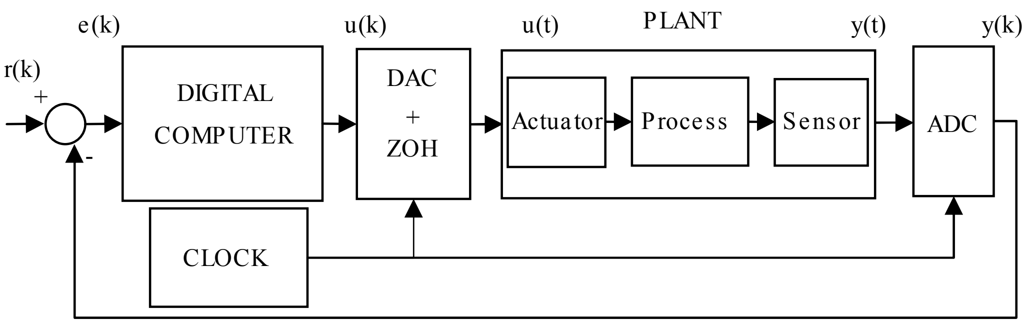
[Wind River]

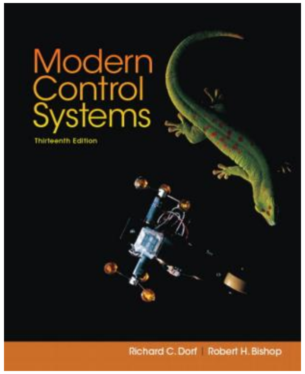
CLASSICAL THEORY FOR DIGITAL CONTROL SYSTEMS

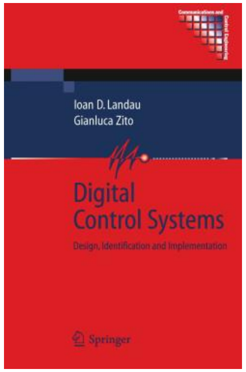
POLYTECHNIQUE MONTRÉAL

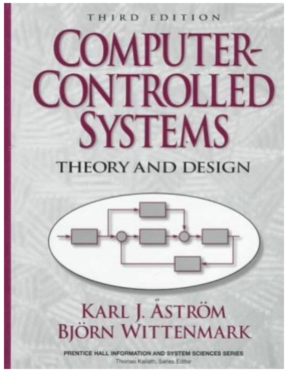


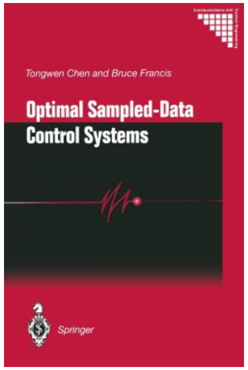
(a)





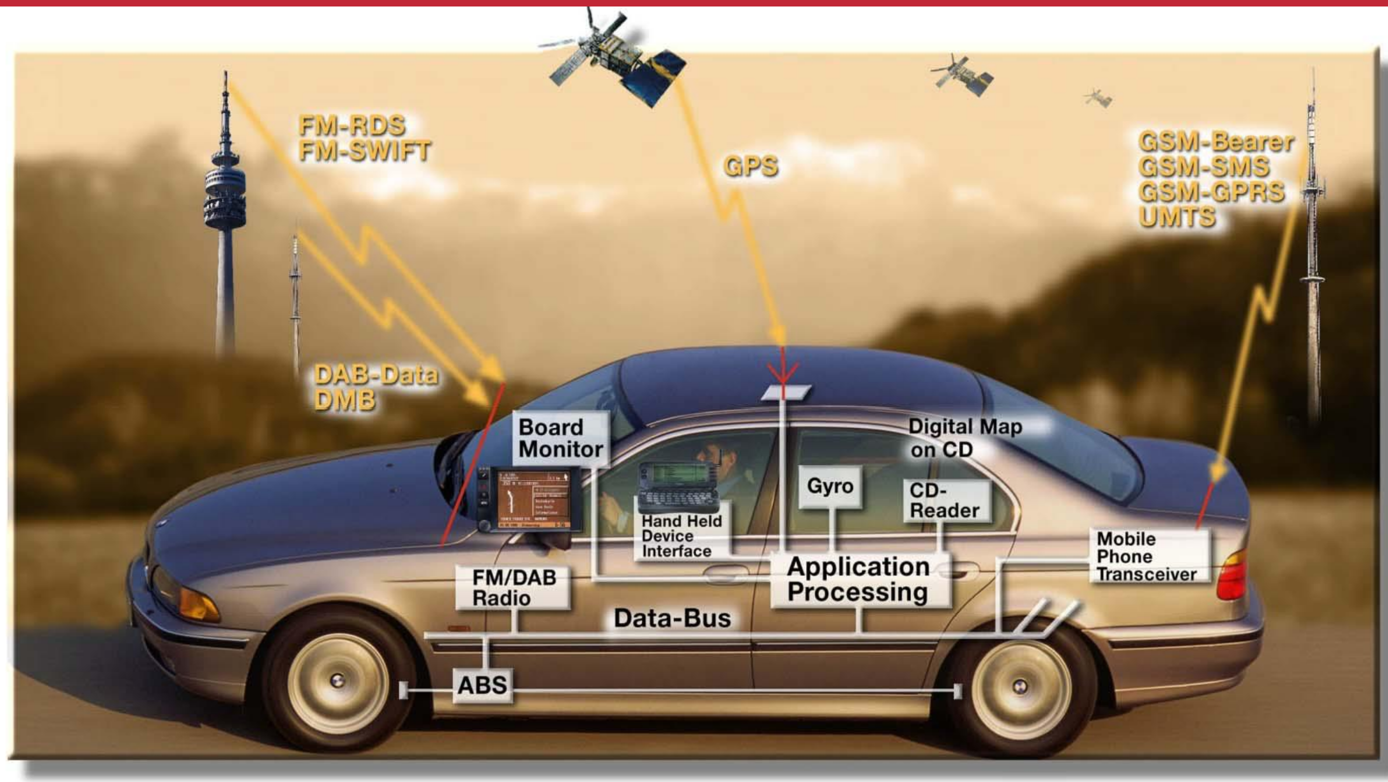






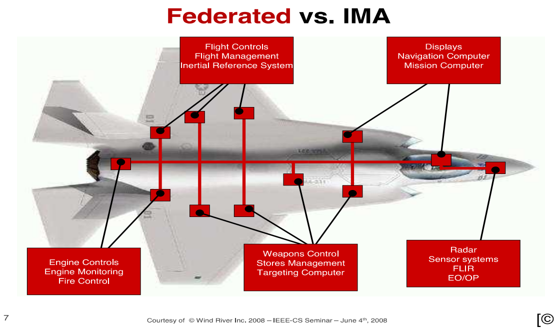
- Most control systems now implemented on digital computers (µproc./µmicrocontrollers): flexibility, maintenance, low-cost, etc.
- Classical abstractions and tools: periodic sampling, synchronized D/A & A/D => discretizations of continuous-time (CT) systems, z-transforms, etc. (ex: ELE8200 course)
- **Very simple models of the computing platform**, computational & communication resources dedicated to control task

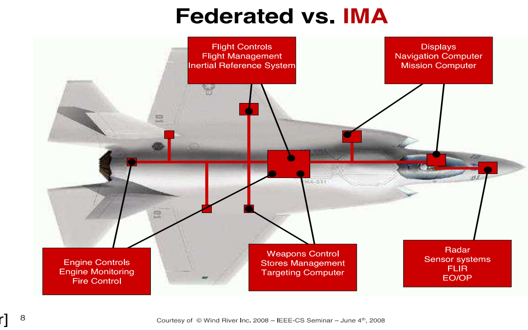
REALITY OF EMBEDDED CONTROL SYSTEMS: NETWORKED, COMPLEX, SHARED RESOURCES



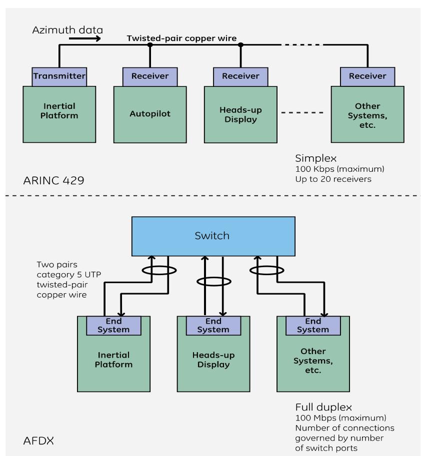
- Modern car: 40-100 networked microprocessors
  - Average modern high-end car: 100 million LOC: brakes, transmission, engine, safety, climate, emissions, multimedia, cloud connectivity, etc.
  - Multitasking computers
  - Several CAN and other communication buses
- Boeing 777: 1280 networked microprocessors, 787: 6.5 million LOC just for avionics & support systems

AVIONICS EXAMPLE

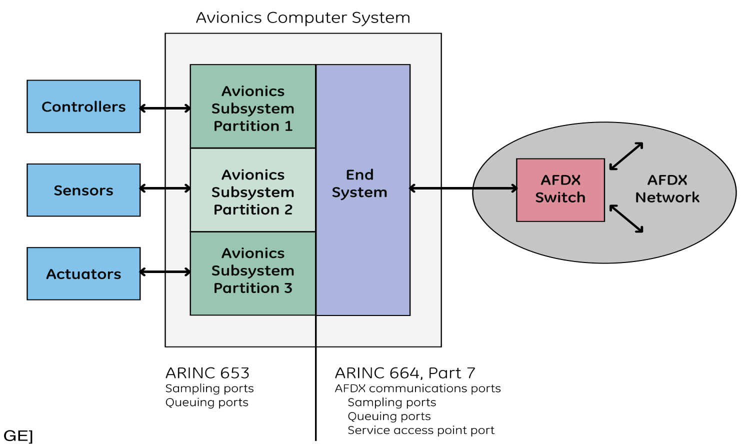




[© Wind River]

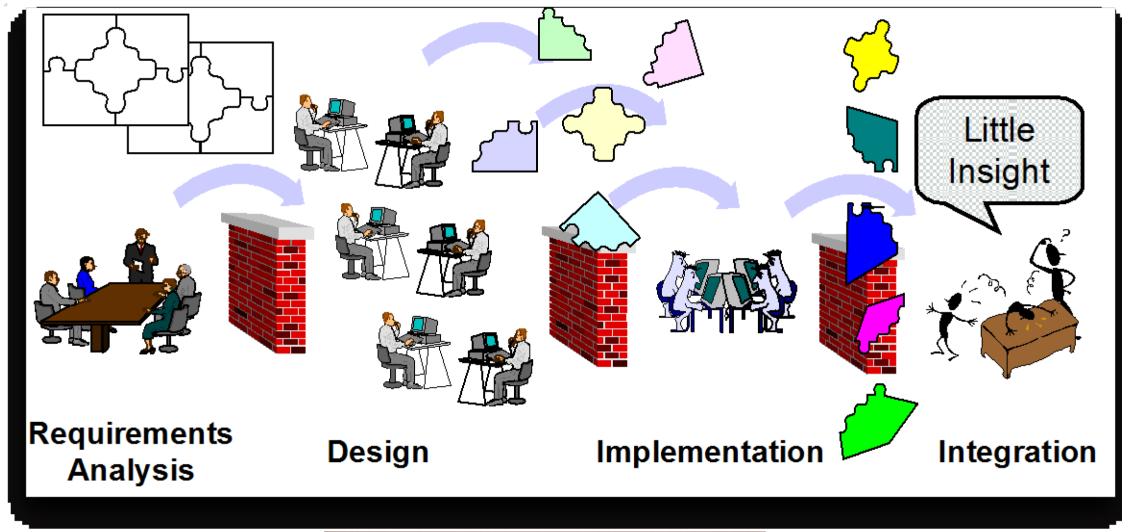


[© GE]

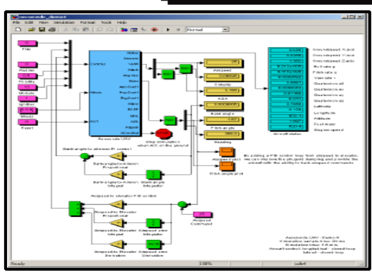


- System architecture choices driven by: cost, weight, modularity, technological evolution (ex: multicore proc., cach), maintenance, etc., not by control theory!
- Result: sharing of computing and communication resources, essentially nondeterministic implementation platforms, etc.
- Bad for digital control abstractions!

RETHINKING THE INTERFACE BETWEEN CONTROL AND EMBEDDED SYSTEM DESIGN



[©B. Lewis and P. Feiler, AADL]



Interface?

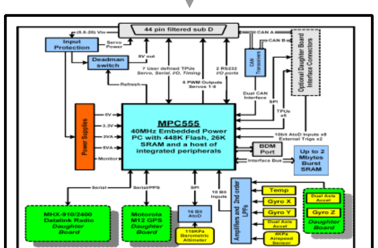
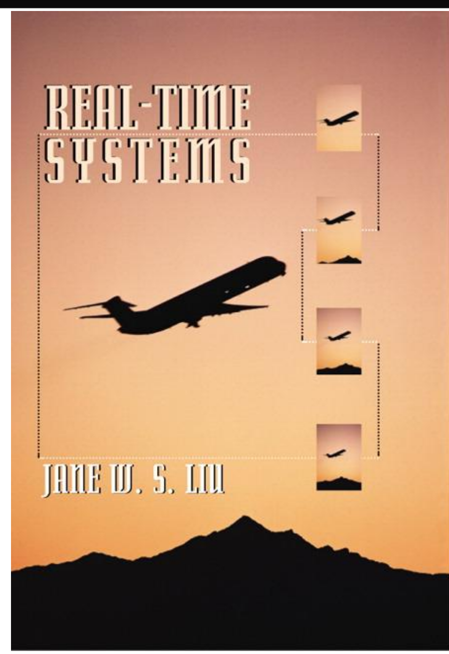


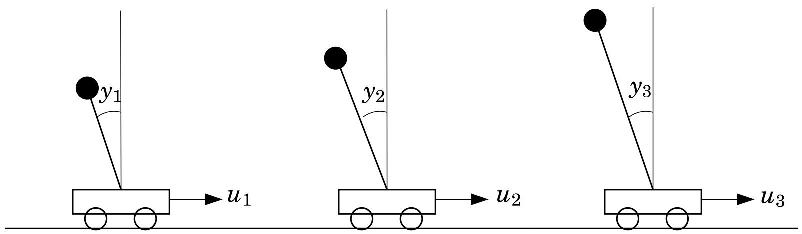
Figure 2. Piccolo block diagram



- Ex: RT system designers tend to treat control task as hard real-time (missing a timing constraint = catastrophic)
- But control engineers choose sampling period typically by “rule-of-thumb” (ex: 10-20 periods during rise time)!
- Moreover RT abstraction of period is different (has jitter)

EX [CERVIN '02]: CONTROL UNDER RATE-MONOTONIC SCHEDULING

3 control loops running on a single CPU

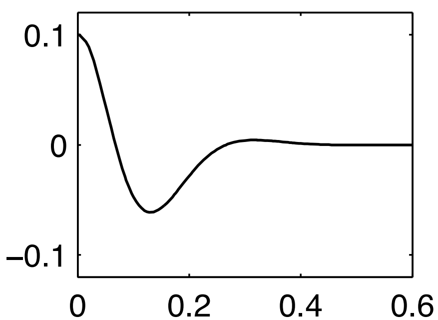


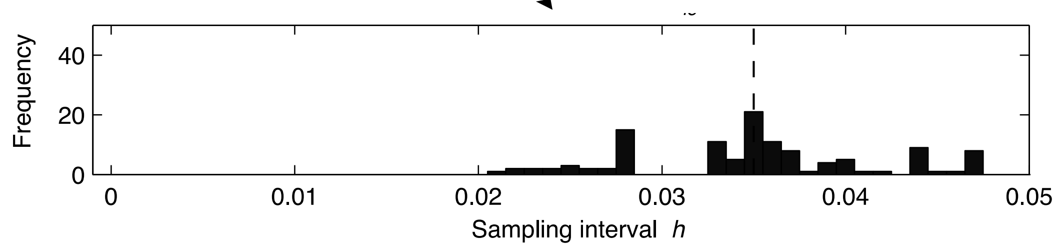
h_1=20ms h_1=29ms h_1=35ms

Control design

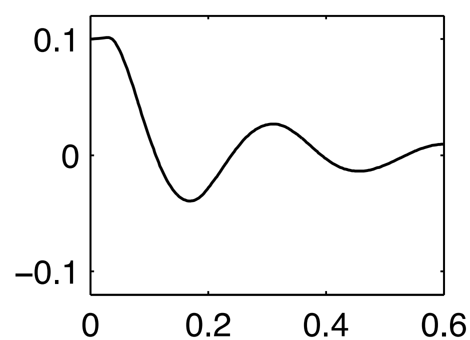
Fixed priority RM scheduling (1>2>3) **with preemption**

Pendulum 3

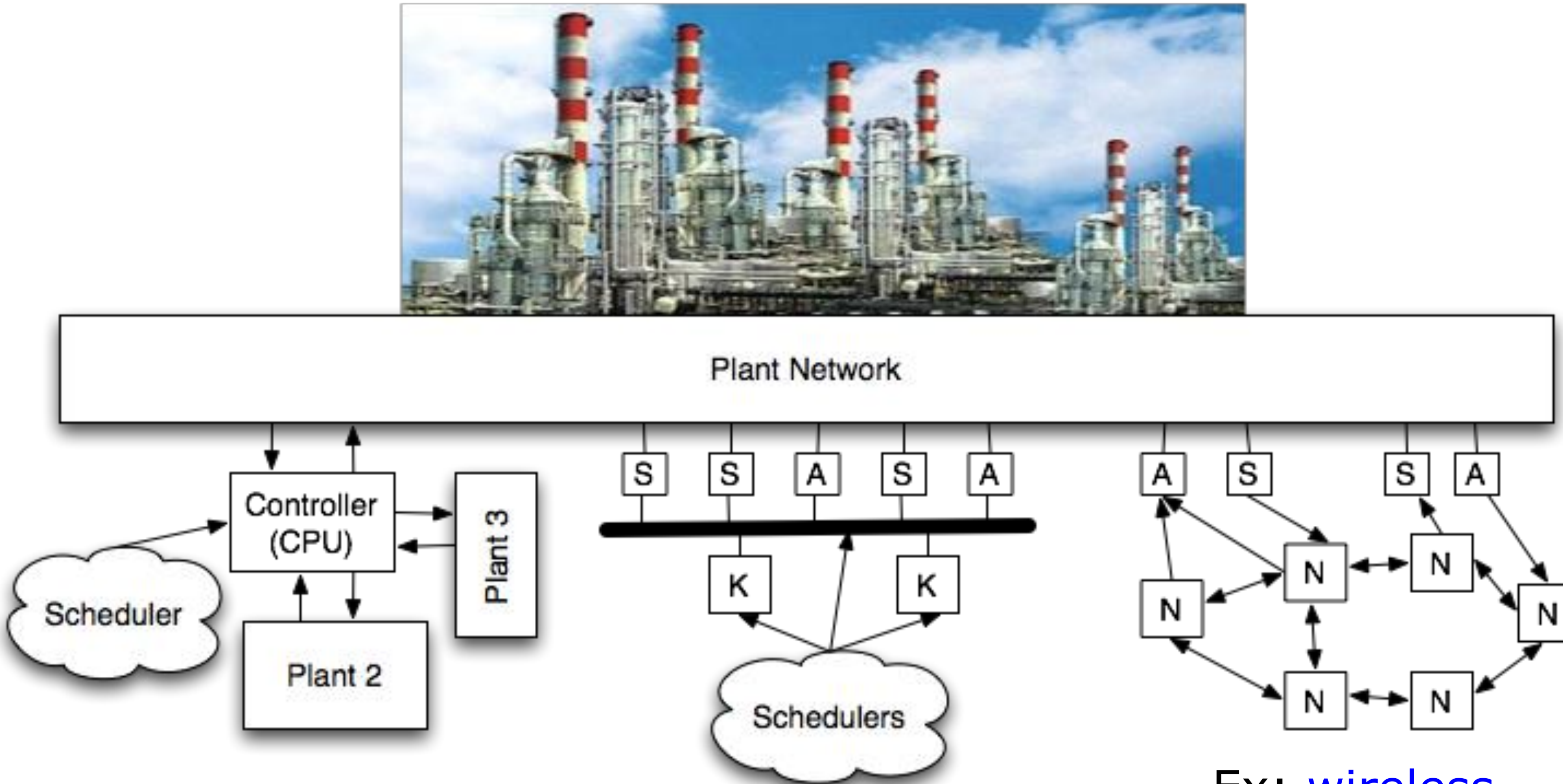




Pendulum 3



VARIETY OF CONTROL SYSTEM CONFIGURATIONS TO CONSIDER



Ex: [wireless control networks]

NECS SIMULATION WITH TRUETIME

POLYTECHNIQUE MONTRÉAL

FEATURE

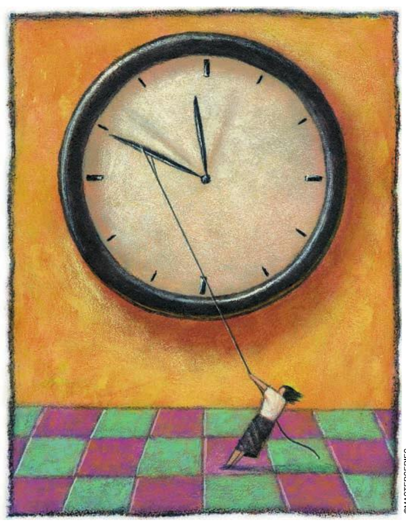
How Does Control Timing Affect Performance?

Analysis and Simulation of Timing Using Jitterbug and TrueTime

Control systems are becoming increasingly complex from both the control and computer science perspectives. Today, even seemingly simple embedded control systems often contain a multitasking real-time kernel and support networking. At the same time, the market demands that the cost of the system be kept at a minimum. For optimal use of computing resources, the control algorithm and the control software designs need to be considered at the same time. For this reason, new computer-based tools for real-time and control codesign are needed.

Many computer-controlled systems are distributed systems consisting of computer nodes and a communication network connecting the various systems. It is not uncommon for the sensor, actuator, and control calculations to reside on different nodes, as in vehicle systems, for example. This gives rise to networked control loops (see [1]). Within the individual nodes, the controllers are often implemented as one or several tasks on a microprocessor with a real-time operating system. Often the microprocessor also contains tasks for other functions (e.g., communication and user interfaces). The operating system typically uses multiprogramming to multiplex the execution of the various tasks. The CPU time and the communication bandwidth can hence be viewed as shared resources for which the tasks compete.

Digital control theory normally assumes equidistant sampling intervals and a negligible or constant control delay from sampling to actuation. However,



©MASTERSERIES

By Anton Cervin, Dan Henriksson, Bo Lincoln, Johan Eker, and Karl-Erik Årzén

Cervin (anton@control.lth.se), Henriksson, Lincoln, Eker, and Årzén are with the Department of Automatic Control, Lund Institute of Technology, Box 118, SE-221 00 Lund, Sweden.

[Cervin et al. IEEE CSM, 2003]

http://www.control.lth.se/truetime/

NECS simulation tool, see HW1

CURRENT GAP BETWEEN CONTROL THEORY AND IMPLEMENTATION PRACTICE

POLYTECHNIQUE MONTRÉAL

- **Some consequences**
  - Overdesign, impose very stringent implementation constraints (ex: synchronization, periodicity with no jitter, etc.) => cost increase
    - Requires comput./communication resources available as soon as needed
  - Potential loss of performance/stability if requirements ignored
    - Quality impact
    - If redesign needed at later state of the design cycles: cost, loss of productivity
  - Loss of flexibility / modularity: hard to add new functions because rescheduling a system requires new simulations / recertifying, with unpredictable results
- **Solution:** develop better interfaces between control design and implementation on computation/communication infrastructure
  - Better abstractions of implementation platforms, useful at design stage for more accurate predictions,
  - Better control design techniques that take implementation constraints into account, mitigate their impact on performance, increase flexibility, etc.
  - Work with CS & communications researchers to develop programming abstractions, communication protocols, hardware, etc., that support rigorous CPS development and the transfer of formal certification/proofs from control design stage to the final implementation stage
  - Review student training curriculum in digital control, broaden scope

SOME MISHAPS…

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

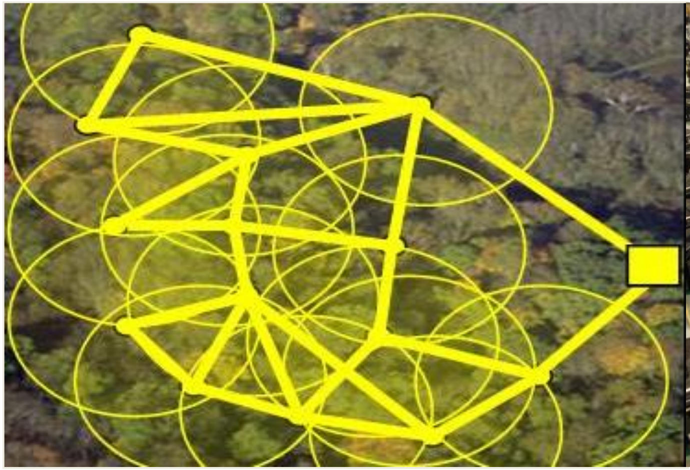
Recalls Underscore Toyota's Commitment to Address All Vehicle Quality and Safety Issues Promptly and Effectively

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.

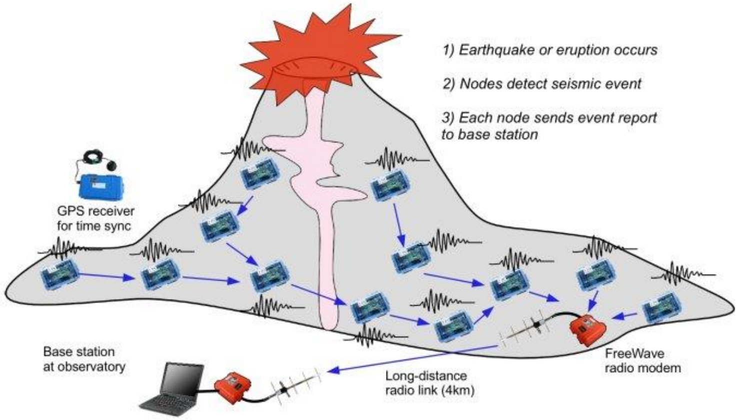
The ABS, in normal operation, engages and disengages rapidly (many times per second) as the control system senses and reacts to tire slippage. Some 2010 model year Prius and 2010 HS 250h owners have reported experiencing inconsistent brake feel during slow and steady application of brakes on rough or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

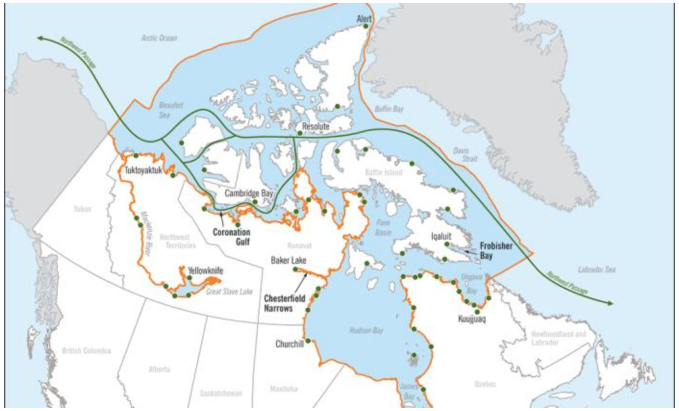
- Bugs in software, but also in specifications! (Particularly problematic because many CPS are safety-critical)
- Formal CPS verification is a topic related to the course, but not covered (can be project topic)

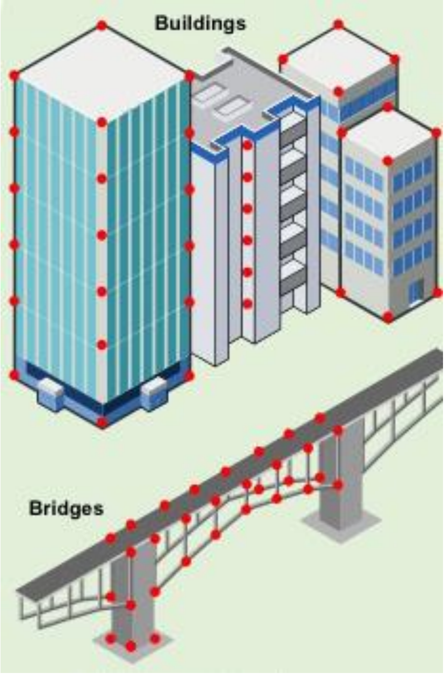
Decentralized Control of Multi-Agent Systems

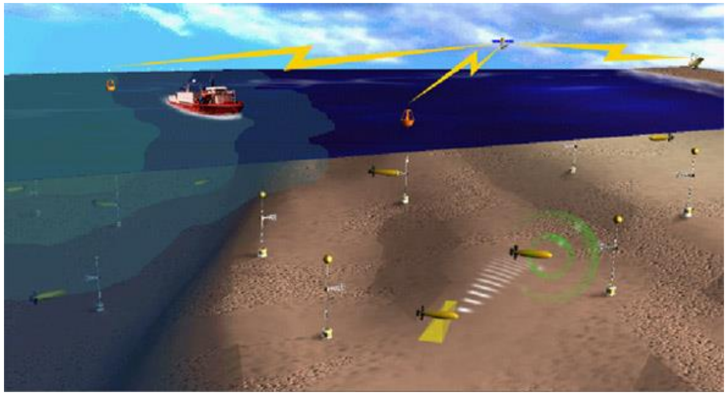


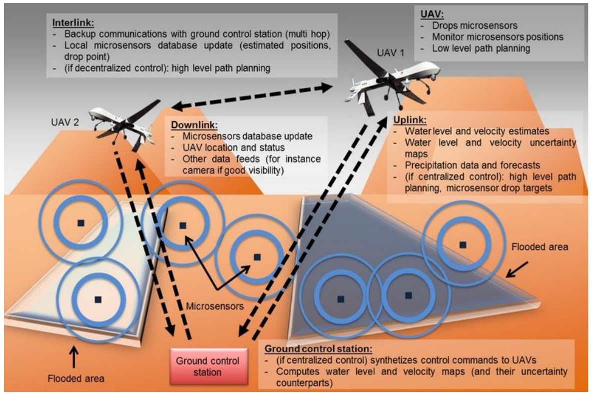
SENSOR NETWORKS







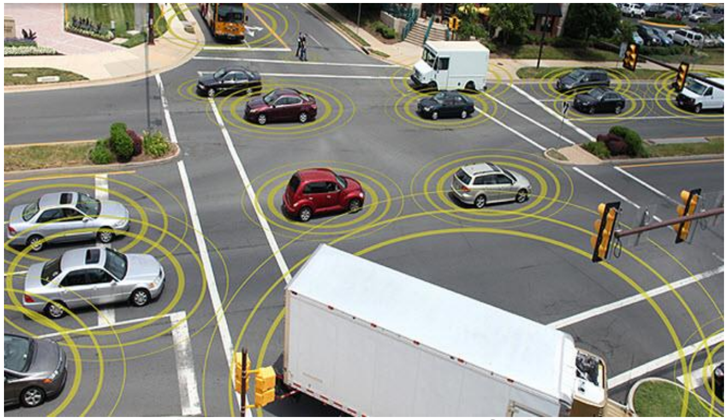




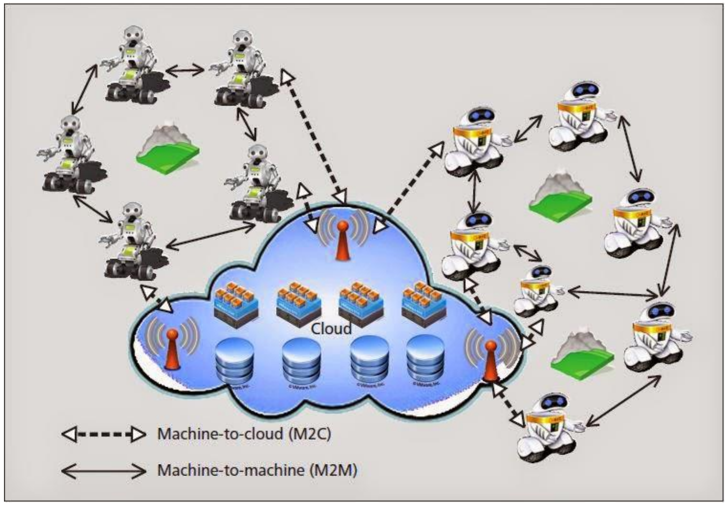
[Claudel et al., 2013]
Ground+airborne predictive flood warning

- Surveillance, environmental monitoring, intelligent infrastructures, etc.
- Static, mobile, hybrid

MOBILE ROBOTIC NETWORKS



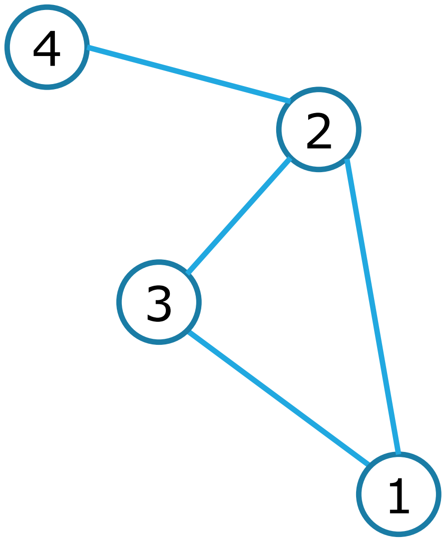
Connected cooperating self-driving cars



- Comms: M2M, Cloud, hybrid
- Fully or partly decentralized architectures

FUNDAMENTAL EXAMPLE: AVERAGE CONSENSUS

- n agents, agent i starts with x_{0i}
- Want to compute the average value
- Only local communication with neighbors
- Possible **distributed** algorithm: average your and your neighbors' values
- Leads to study the dynamical system



$$\mathbf{x}_{k+1} = \begin{bmatrix} 1/3 & 1/3 & 1/3 & 0 \\ 1/4 & 1/4 & 1/4 & 1/4 \\ 1/3 & 1/3 & 1/3 & 0 \\ 0 & 1/2 & 0 & 1/2 \end{bmatrix} \mathbf{x}_k$$

- Properties? Convergence? To mean? Speed? Conditions on communication graph?

- Examples of other tasks: distributed estimation, detection, tracking, decision, localization, synchronization, area coverage, etc.

CENTRALIZED VS. DECENTRALIZED CONTROL

- In this part, the individual nodes are more independent, capable of some level of decision making
- Fundamental questions:
  - Right information sharing architectures?
  - Decentralized estimation, detection, decision-making, etc., ?
  - How these two issues relate
- Some pros/cons of centralized systems (ex: cloud)
  - + All available information necessary at central node
  - + Conceptually simpler algorithms, standard computation model
  - - potentially single point of failure (but not nec. true for cloud comp.)
  - - communication bandwidth
  - - latency due to round-trip delays
- Some pros/cons of decentralized systems
  - + Potentially simpler implementation, maintenance; plug-and-play for new nodes, no need to maintain global network view)
  - + Resilient, no single point of failure
  - + Sometimes no real choice (ex: human teams)
  - - Conceptually more complicated algorithms; bandwidth spared?
  - - Potential loss of optimality in decisions
- Hybrid? (ex: cloud+edge computing)

ROBOTARIUM TO TEST IDEAS

Georgia Tech

INSTITUTE FOR ROBOTICS AND INTELLIGENT MACHINES

ROBOTARIUM

Opening August 22, 2017, the 725-square-foot Robotarium facility houses more than 100 rolling and flying swarm robots that are accessible to anyone. Researchers from around the globe can write their own computer programs, upload them, then get the results as the Georgia Tech machines carry out the commands.

Overview

No other university has a facility comparable to the Robotarium. Located in the Van Leer Building in the heart of Georgia Tech's campus, motion capture cameras mounted on the ceiling peer down at the lab's centerpiece: a white, bowl-shaped arena.

Up to 100 palm-sized, rolling robots move around the surface. They automatically activate when given a program from someone in the room or a remote coder in a different state or country. Once it finishes the experiment, the swarm autonomously returns to wireless charging slots on the edge of the rink and waits to be activated for its next mission.

Quick Facts

- $2.5 million lab funded by the National Science Foundation (NSF) and Office of Naval Research
- Democratizes robotics research by providing access to resources that otherwise are cost-prohibitive
- Autonomous quadcopters will soon be added to the 725-square-foot lab for remote flying experiments
- More than 100 research groups have logged on and used the prototype, mini version of the Robotarium

FOR MORE INFORMATION, VISIT ROBOTARIUM.ORG

Georgia Tech

ENVISIONED AND DEVELOPED BY MAGNUS EGERSTEDT

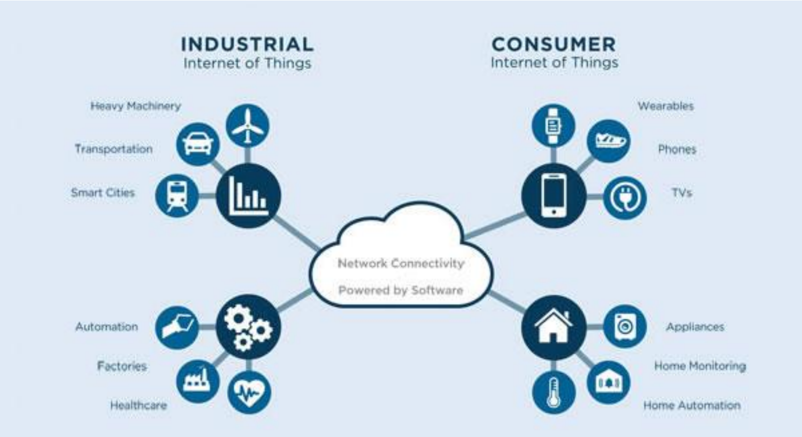
3-D PRINTED SWARM OF GROUND ROBOTS

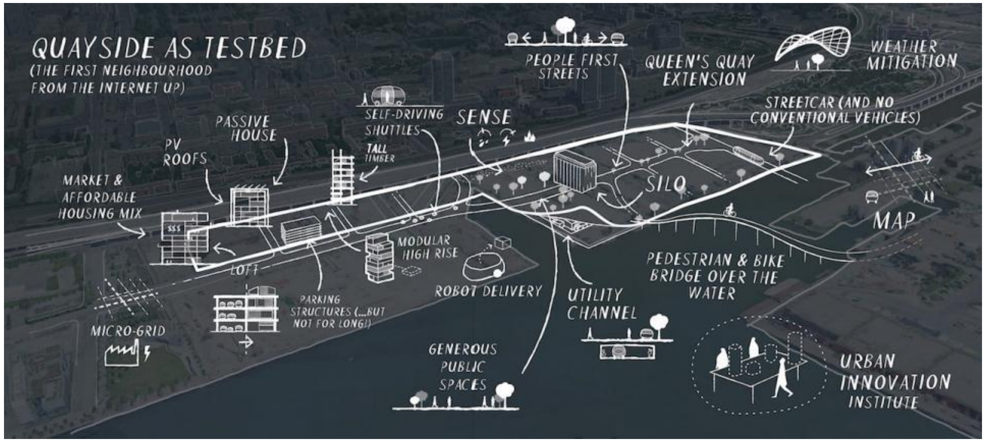
MEMBERS OF THE ROBOTARIUM TEAM

ROBOTARIUM by GeorgiaTech

Expect to use for HW3. Start with simulator:
https://github.com/robotarium/robotarium-matlab-simulator
https://github.com/robotarium/robotarium_python_simulator

Next-Generation Distributed Monitoring and Control Systems

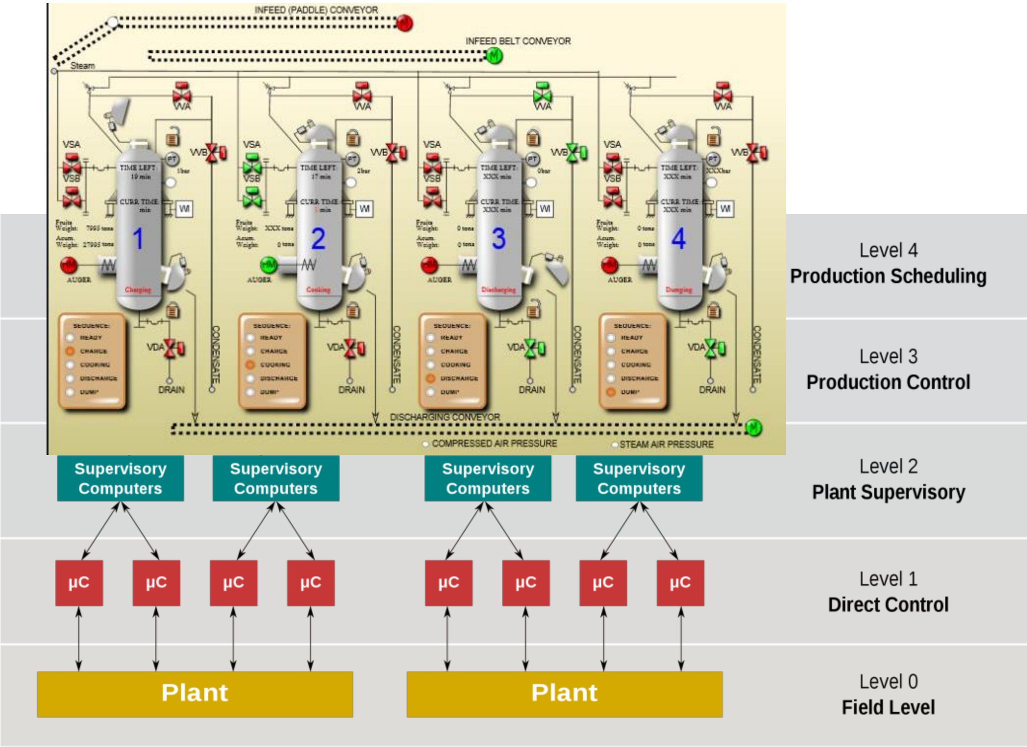




[Sidewalk Toronto]

SCADA SYSTEMS

POLYTECHNIQUE MONTRÉAL



SCADA / DCS: supervisory control, alert monitoring, change setpoints

PIDs, PLCs, RTUs: hard real-time feedback control, simple logic

- Monitoring and control of typically large-scale (industrial) processes
- Applications: process control, oil & gas, water distribution, sewage treatment, power grids, building HVACs, assembly lines, etc.

CORRESPONDENCE WITH CONTROL AND OPTIMIZATION TECHNOLOGY

POLYTECHNIQUE MONTRÉAL

Level	*Description*	*Goal*	*Time frame*	*Typical design tool*
4	Plant wide optimization	Meeting customer orders and scheduling supply of materials	Everyday (say)	Static optimization
3	Steady state optimization at unit operational level	Efficient operation of a single unit (e.g. distillation column)	Every hour (say)	Static optimization
2	Dynamic control at unit operation level	Achieving set-points specified at level 3 and achieving rapid recovery from disturbances	Every minute (say)	Multivariable control, e.g. Model Predictive Control
1	Dynamic control at single actuator level	Achieving liquid flow rates etc as specified at level 2 by manipulation of available actuators (e.g. valves)	Every second (say)	Single variable control, e.g. PID

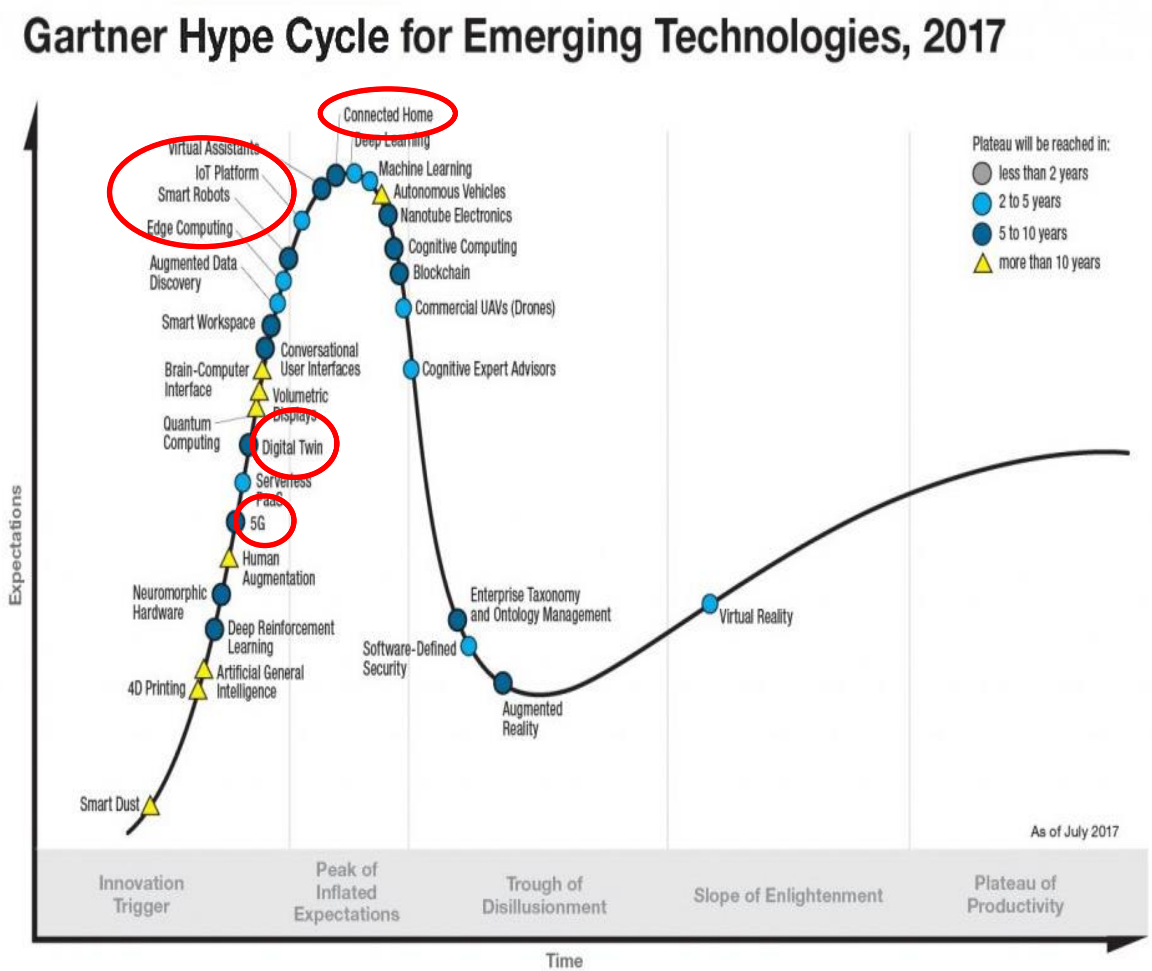
[Goodwin, Graebe and Salgado, 2000]

THE INTERNET OF THINGS (IOT)

POLYTECHNIQUE MONTRÉAL

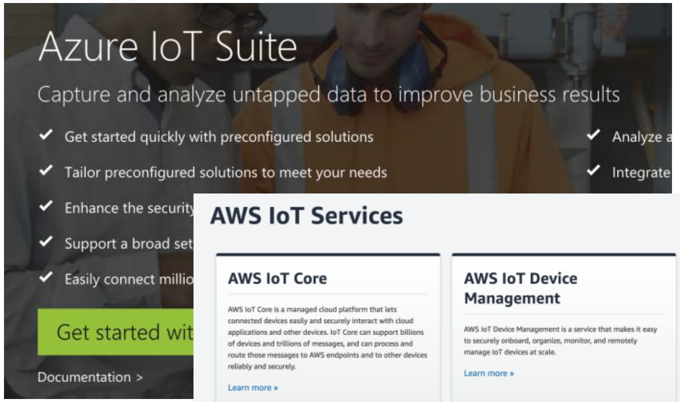
- IoT: term coined by technologist [Kevin Ashton]
- Could be viewed as an extension of SCADA and sensor network concepts outside of traditional industrial applications:
  - Smart everything: infrastructure, power grid, manufacturing, agriculture, transportation, buildings, cities, environmental monitoring, etc.
  - Sensors everywhere **around us** => collect data => inform decisions (real-time + long-term planning)
  - Will most likely involve loops at different spatial and time scales (edge computing)
  - Leads to Cyber-Physical **Human** Systems; implications of interactions with self-interested human agents, social factors
- Triggered by progress/developments in several technological areas:
  - Embedded computing (raw power, form factors, etc.)
  - Miniaturization of sensors (ex: MEMS)
  - Networking (in particular wireless)
  - Cloud computing, etc.
- Similar ideas branded by different companies / groups:
  - Fog Computing, Swarm Computing, Edge Computing, Industrial Internet, Industry 4.0
  - Embraced by cloud computing providers, telecom companies (M2M driving some 5G requirements), etc., as potentially important source of business

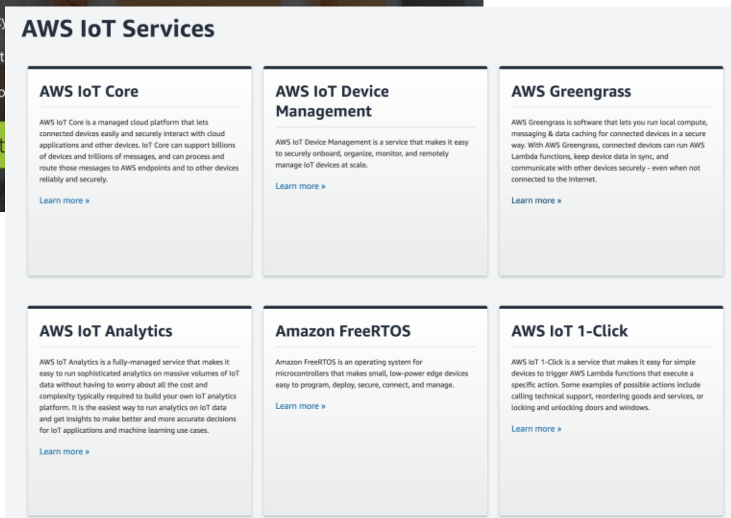
POSITION ON THE HYPE CYCLE

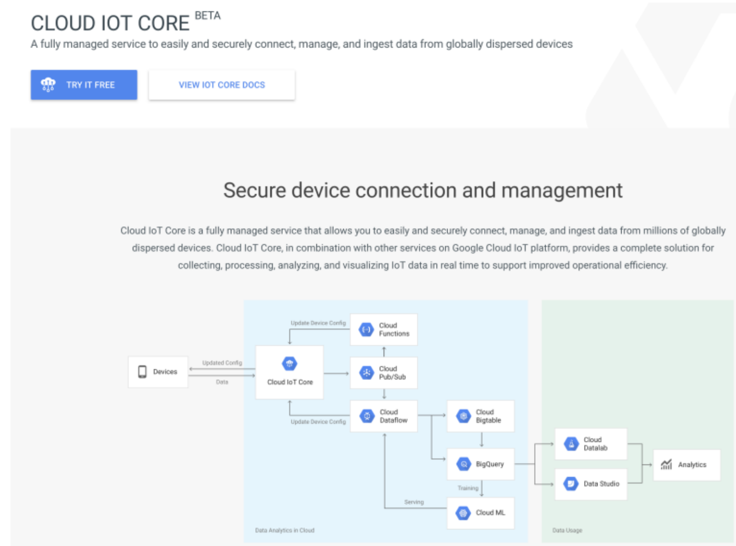


- Potential impact of these technologies still to determine, but probably useful tools for automation nonetheless

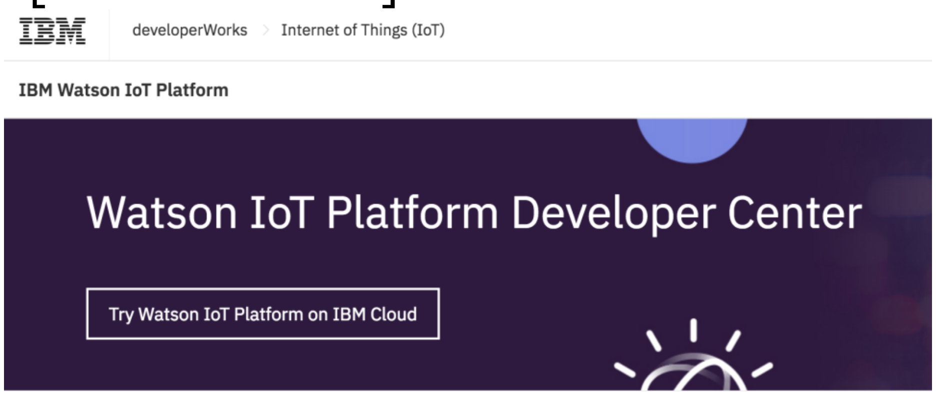
SOME IOT PLATFORMS & HARDWARE KITS

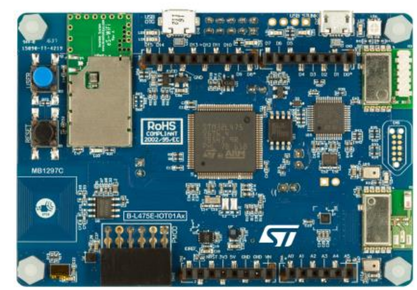






[Focus of HW4]





arm MBED Enabled

STM32 L4

SOFTWARE ECOSYSTEM FOR BIG DATA, CLUSTER COMPUTING AND STREAM ANALYTICS

POLYTECHNIQUE MONTRÉAL

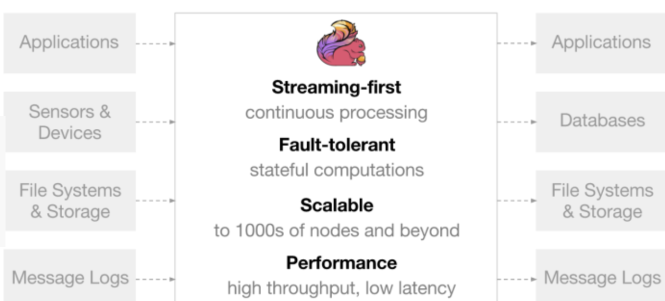
https://db-engines.com/en/

cassandra

redis

APACHE HBASE

Flink



APACHE hadoop

APACHE kafka
A distributed streaming platform

APACHE Spark

Spark Streaming

Amazon Kinesis

Easily collect, process, and analyze video and data streams in real time

Azure Stream Analytics

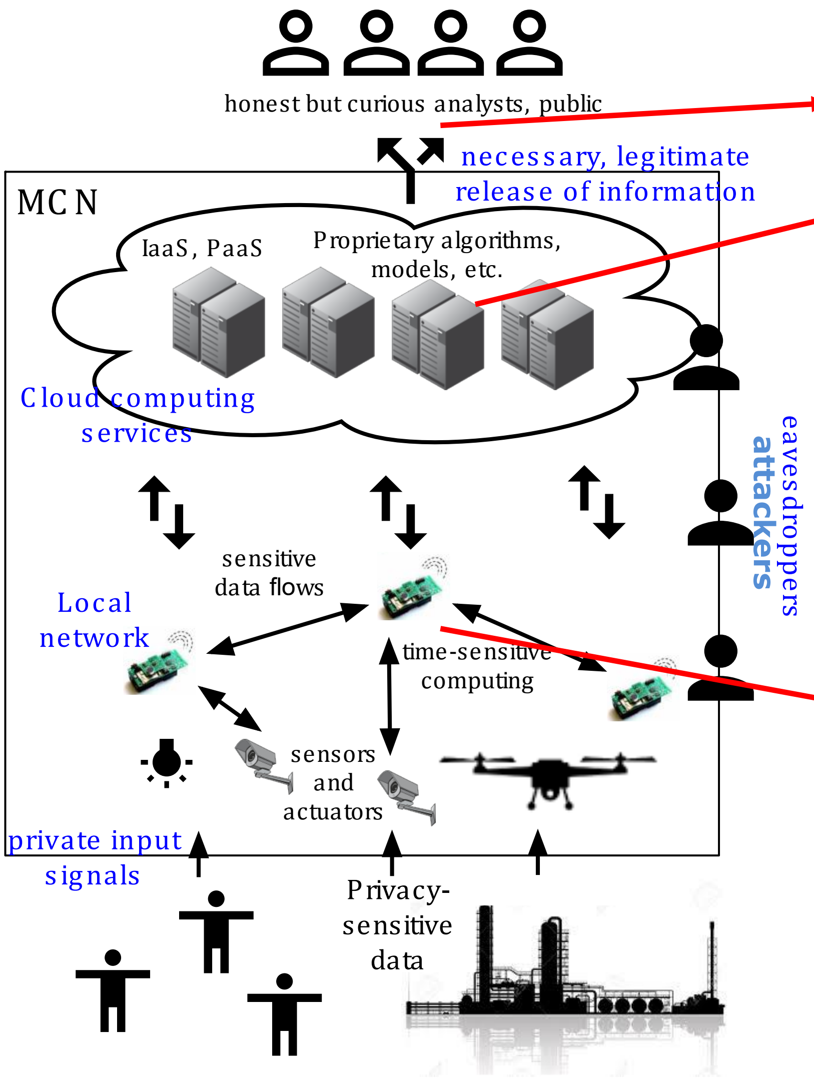
IBM Data Science Experience

IBM Streams

A complete real-time analytics solution with development environment, runtime and analytic toolkits.

Zoo with many other alternative options, open-source or not…

SECURITY & PRIVACY



- Privacy: (unintentional) disclosure of sensitive information to third party
- Crypto tools for communication between trusted parties, computation on semi-trusted infrastructures
- Need more than crypto for privacy-preserving data analysis; even aggregate data publication leaks info
- Issue magnified by proliferation of sensors, close to individual users of "smart infrastructures"
- Ex: location data, electricity markets, etc.

- Security: traditionally neglected in SCADA systems, IoT offers many more attack opportunities
- Critical data theft, false data injection, change setpoints, etc.
- Potentially catastrophic outcomes due to physical consequences, safety-criticality
- Beyond fault-detection: malicious intent of attacker vs random / predictable

EXAMPLES OF ATTACKS ON SCADA SYSTEMS

Maroochy Waste Water

Event: More than 750,000 gallons of untreated sewage intentionally released into parks, rivers, and hotel grounds

Impact: Loss of marine life, public health jeopardized, $200,000 in cleanup and monitoring costs

Specifics: SCADA system had 300 nodes (142 pumping stations) governing sewage and drinking water

- Used OPC ActiveX controls, DNP3, and ModBus protocols
- Used packet radio communications to RTUs
- Used commercially available radios and stolen SCADA software to make laptop appear as a pumping station
- Caused as many as 46 different incidents over a 3-month period (Feb 9 to April 23)

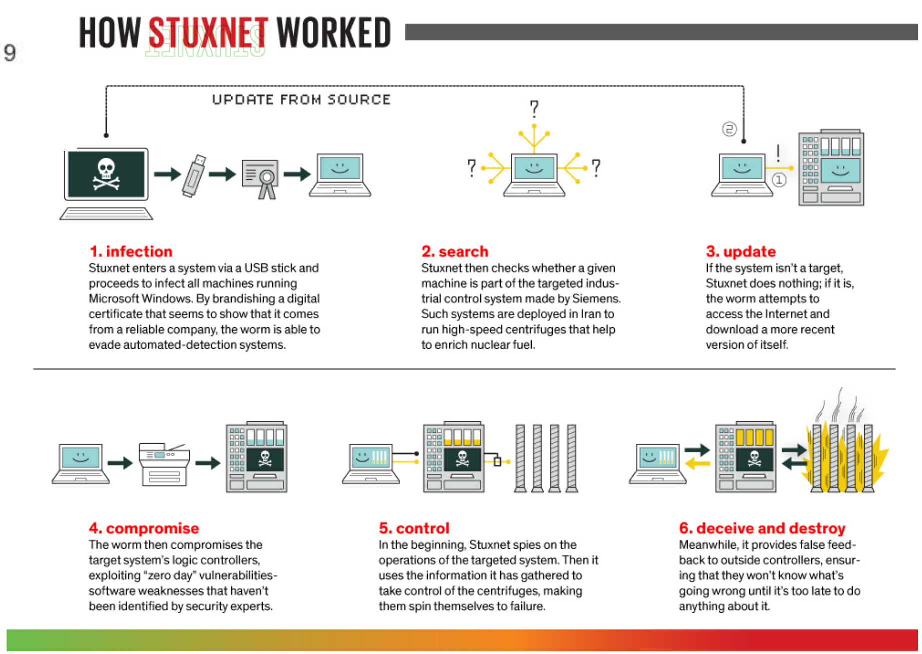
Lessons learned:

- Suspend all access after terminations
- Investigate anomalous system behavior
- Secure radio and wireless transmissions

Homeland Security

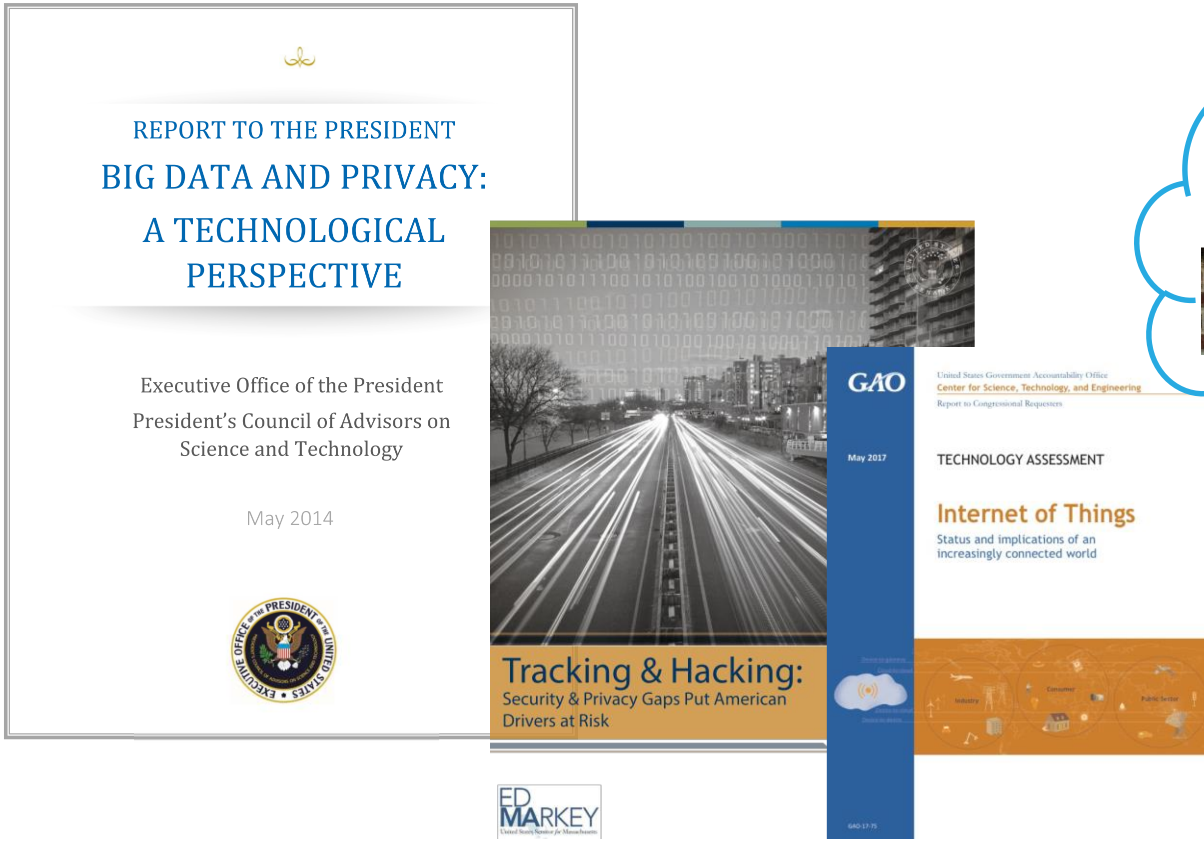
http://goo.gl/3b9U9s

https://www.youtube.com/watch?v=C_PRhTXp6VQ



PRIVACY CONCERNS WITH IOT, CONNECTED VEHICLES & BIG DATA

POLYTECHNIQUE MONTRÉAL



"I strongly urge the commission to recommend that privacy enhancing technologies (PETs), such as secure multi-party computation (MPC) and differential privacy, must be utilized by agencies and organizations that seek to draw public policy related insights from the private data of Americans." -- Sen. Ron Wyden (Oregon), May 2017

PRIVACY / UTILITY TRADEOFFS

- Fundamental challenge: perform/publish accurate **data analysis** at the aggregate (population) level while providing formal privacy protection guarantees to the individual data providers

Course Organization

SPIRIT OF THE COURSE & EVALUATION

POLYTECHNIQUE MONTRÉAL

- Semi-open discussion about topics of current interest in CPS
  - Biased by my own interests and research activities
  - Roughly 1/3 NECS, 1/3 MA, 1/3 DCS (IoT, fault detection, security, privacy)
  - Not mature like topics from standard curriculum: needs your active participation to explore the material, tools, reading, debugging, etc.
  - Your suggestions on how to do things better are appreciated
- Opportunity to introduce new material as needed
  - Theory (state-space and frequency-domain analysis for NECS, basic algebraic graph theory, fault detection methods, differential privacy…)
  - Computational methods (LMIs and SDP, …)
  - Software (simulation, IoT platforms, cluster computing frameworks, …)
- Evaluation
  - Homework (4 or 5 problem sets, 40%)
  - Giving a mini-lecture based on a paper (20%, 10-15 min talk+notes)
  - Project on a topic related to the class (40%)

PROJECT

POLYTECHNIQUE MONTRÉAL

- Needs to be a bit more active than a paper summary/literature review (purpose of mini-lecture already)
- I want to see some personal input
  - Try to implement a method from a paper on a different, reasonably complicated problem
  - Design and simulate a CPS with a software tool, analyze it
  - Experiment with hardware and IoT platforms
  - Extend the available theory (can be related to your research, but no recycling of other courses/projects)
- Short presentation, final report
- Work in pairs (preferred)
  - 3 if particularly ambitious project that can be demonstrably split (need permission)
- Not restricted to the exact topics covered in the course, but should be related