Sistemi Embedded

A.A. 2012/2013

3CFU Automatica

Prof. Maria D. Di Benedetto

Università dell'Aquila – Centro di Eccellenza DEWS

Organizzazione corso

- *Prima parte* (6 CFU ING-INF/05)
  Sviluppo di hw/sw per sistemi embedded (dedicati)

- ***Seconda parte* (3 CFU ING-INF/04)**
  Controllo embedded mediante costruzione di **modelli simbolici**
  (Anche parte del corso di Analisi e Controllo di Sistemi Ibridi)

Docenti (3 CFU ING-INF/04)

- Prof. Maria Domenica Di Benedetto
  - UNIVAQ - Direttore del Centro di Eccellenza DEWS
  - Email: mariadomenica.dibenedetto@univaq.it
- Ing. Giordano Pola
  - UNIVAQ – Ricercatore Centro di Eccellenza DEWS
  - Email: giordano.pola@univaq.it
- Ing. Alessandro Borri
  - IASI-CNR Roma
  - Email: alessandro.borri@univaq.it

Struttura del corso e modalità d'esame

Struttura del corso:

- 24 ore
  - Costruzione di modelli simbolici per sistemi di controllo nonlineari ed ibridi
- 6 ore
  - Seminari tecnico/pratici e tesine

Modalita' d'esame:

- Prova scritta sul tema della progettazione controllori embedded "correct-by design" (10/30)

Materiale didattico

- Paulo Tabuada, Verification and Control of Hybrid Systems, A symbolic approach, Springer, 2009
- Articoli di ricerca
- Sito della didattica e-learning – Prof. M.D. Di Benedetto
- Sito del Prof. L. Pomante www.pomante.net

HYCON2-EECI Graduate School on Control 2013

"Symbolic control design of Cyber-Physical systems"

29/04/2013 – 03/05/2013

Istanbul (Turkey)

www.eeci-institute.eu

Embedded Systems
2012/13

Basilica di Santa Maria di Collemaggio, 1287, L'Aquila

Lecture 1
Introduction

Maria Domenica Di Benedetto
Giordano Pola
Alessandro Borri
University of L'Aquila

Embedded Systems

DEWS CENTER OF EXCELLENCE

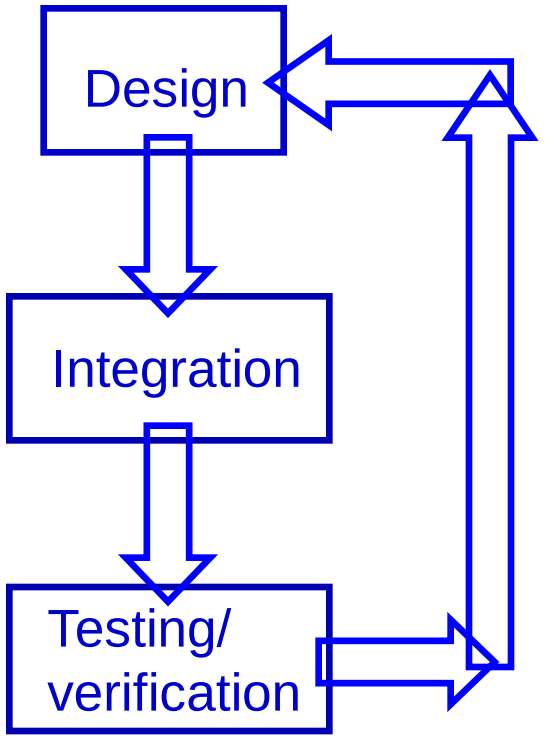
- Computational systems, but not stand alone computers
- Computation interfacing sensors and actuators
- Reactive to physical environment stimuli
- Networked and distributed information processing

Correct-by-design methodology for embedded systems design

Embedded Systems Design

Decomposition of concerns:

- Systems and control engineers design controllers assuming fast sampling rates for correct operation
- Software engineers design real-time operating systems and high level code assuming correct operation of controllers
- Different teams design different modules that coordinate through digital as well as physical communication
- The integrated product is then tested and/or verified to determine correctness



Embedded Systems Design

Decomposition of concerns:

- Systems and control engineers design controllers assuming fast sampling rates for correct operation
- Software engineers design real-time operating systems and high level code assuming correct operation of controllers
- Different teams design different modules that coordinate through digital as well as physical communication
- The integrated product is then tested and/or verified to determine correctness

Conservative design

Simulation/Verification is currently the only available method to “prove” correctness of embedded systems.
It is time and cost demanding

When a bug is found the system has to be redesigned, which may introduce new bugs

Formal verification is only possible for systems with very simple continuous dynamics, e.g. timed automata, rectangular hybrid systems, ...

Embedded Systems Design

Decomposition of concerns:

- Systems and control engineers design controllers assuming fast sampling rates for correct operation
- Software engineers design real-time operating systems and high level code assuming correct operation of controllers
- Different teams design different modules that coordinate through digital as well as physical communication
- The integrated product is then tested and/or verified to determine correctness

Powertrain Unit by Magneti-Marelli

Memory	256 Kb
Lines of C code	50 000
Productivity	6 Lines/Day
Changing rate	3 years
Development effort	40 man-year
Validation time	5 months
Time to market	24 months

*Fabio Romeo, Magneti-Marelli Design Automation Conference, Las Vegas, June 20th, 2001

Example: Powertrain Embedded Control

- Electronic device controlling an internal combustion engine and a gearbox
- The goal
  - offer appropriate driving performance (e.g. torque, comfort, safety)
  - minimize fuel consumption and emissions
- Relevant characteristics
  - strictly coupled with mechanical parts
  - hard real-time constraints
  - complex algorithms for controlling fuel injection, spark ignition, throttle position, gear shift …
  - Heuristic algorithms and simplistic average-value models used in the past

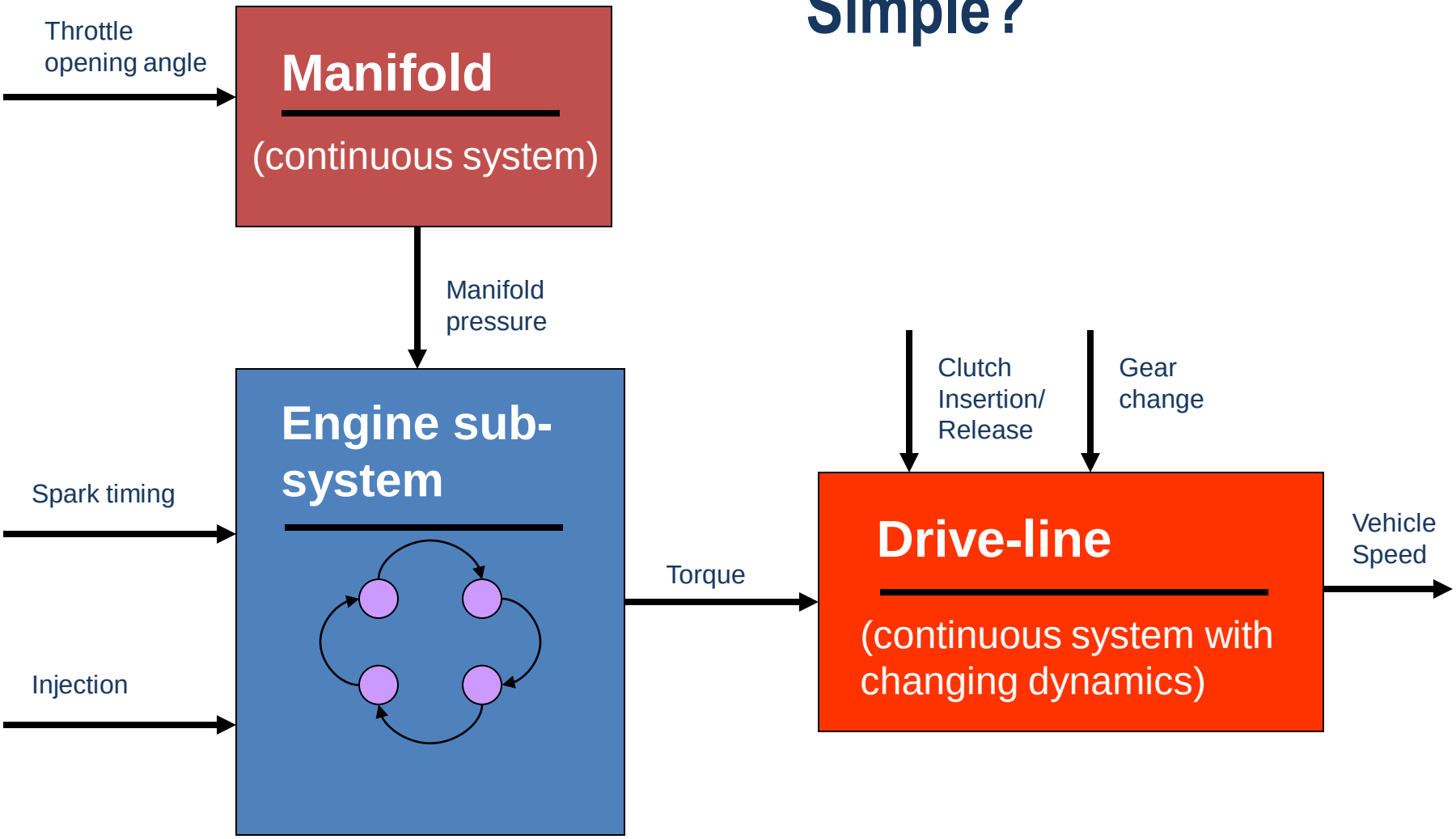
Challenges

- Increasing complexity
- New safety, quality and emission requirements
- New System-on-Chip architectures
- Shorter time to market

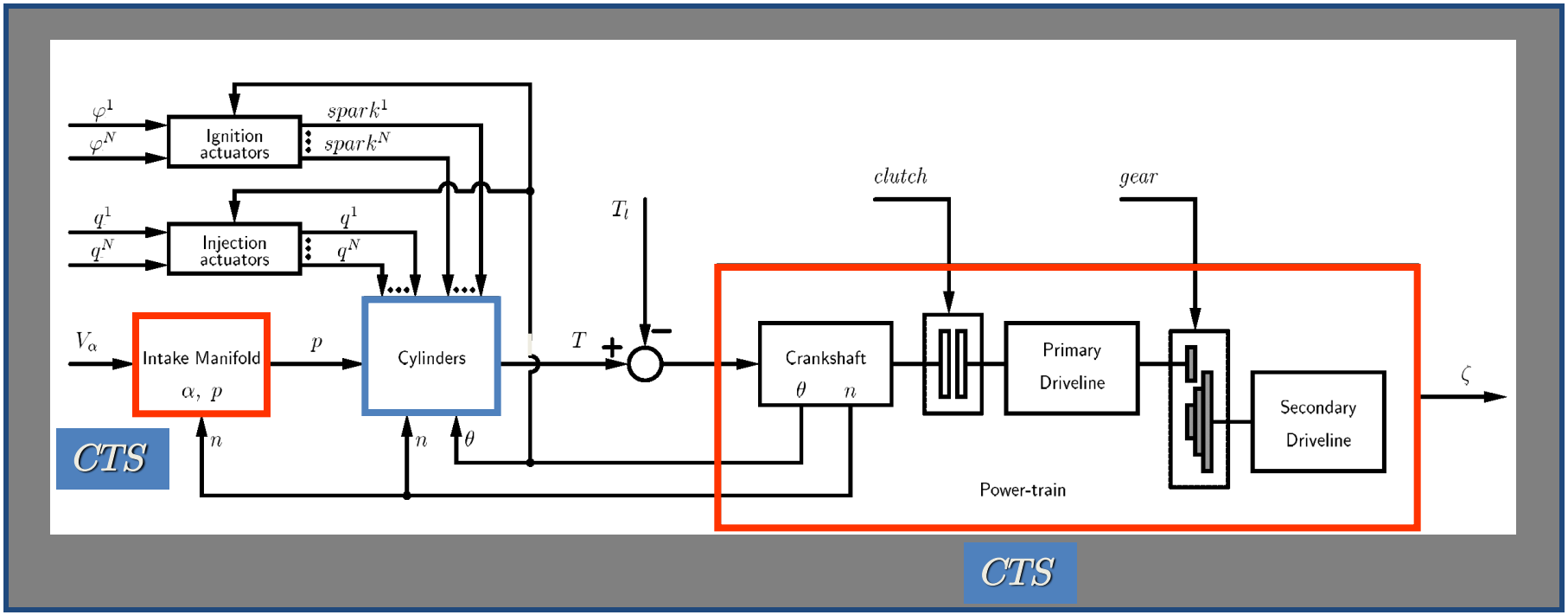
NEED more rigorous approach with

- More accurate models
- Clear partition between algorithms and implementation
- Formal proofs of correctness
- Successive refinement from specifications to implementation
- Executable Specifications

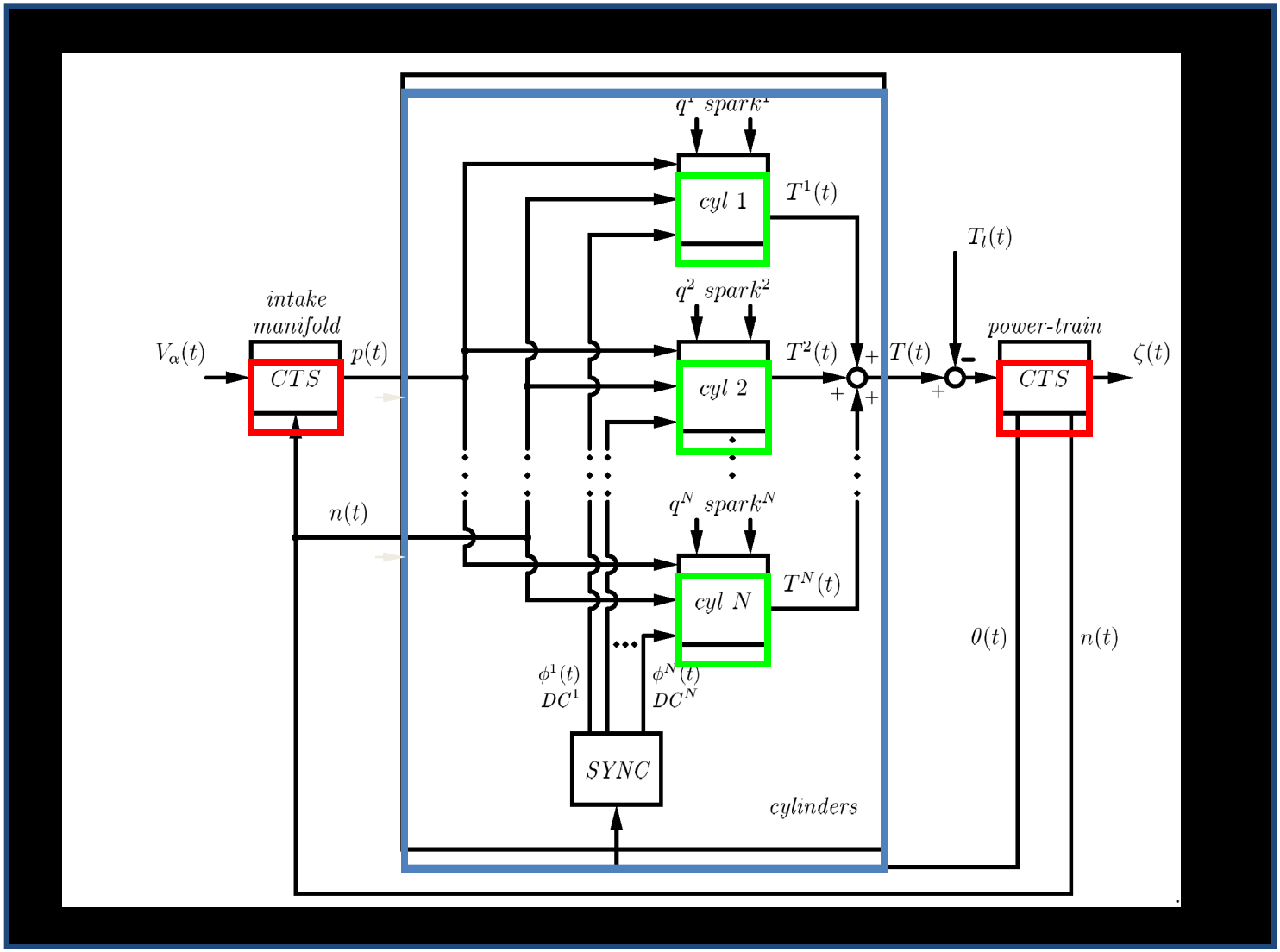
Simple?


Throttle opening angle
Manifold
(continuous system)
Manifold pressure
Engine sub-system
Spark timing
Injection
Torque
Clutch Insertion/ Release
Gear change
Drive-line
(continuous system with changing dynamics)
Vehicle Speed

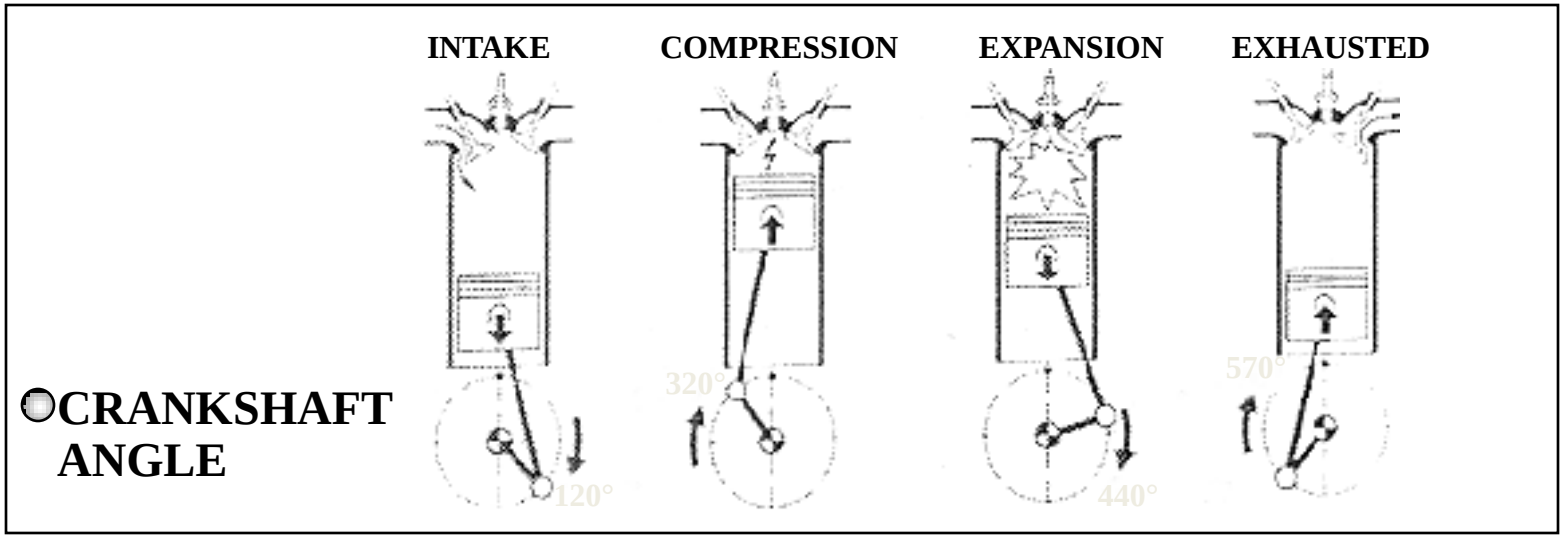
Engine and Driveline



Engine and Driveline



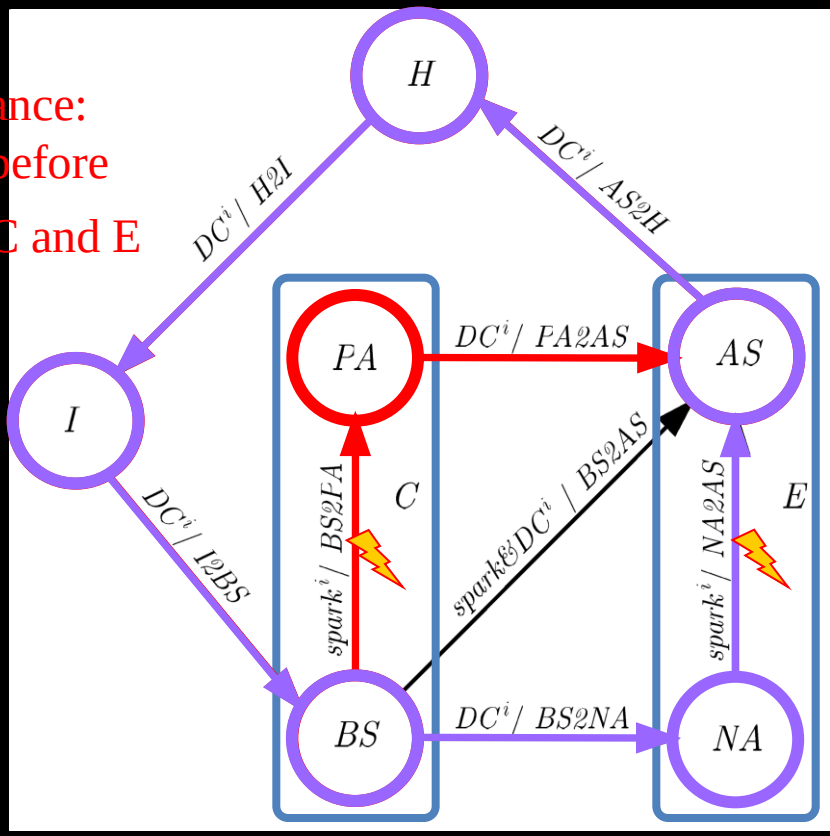
Combustion Process



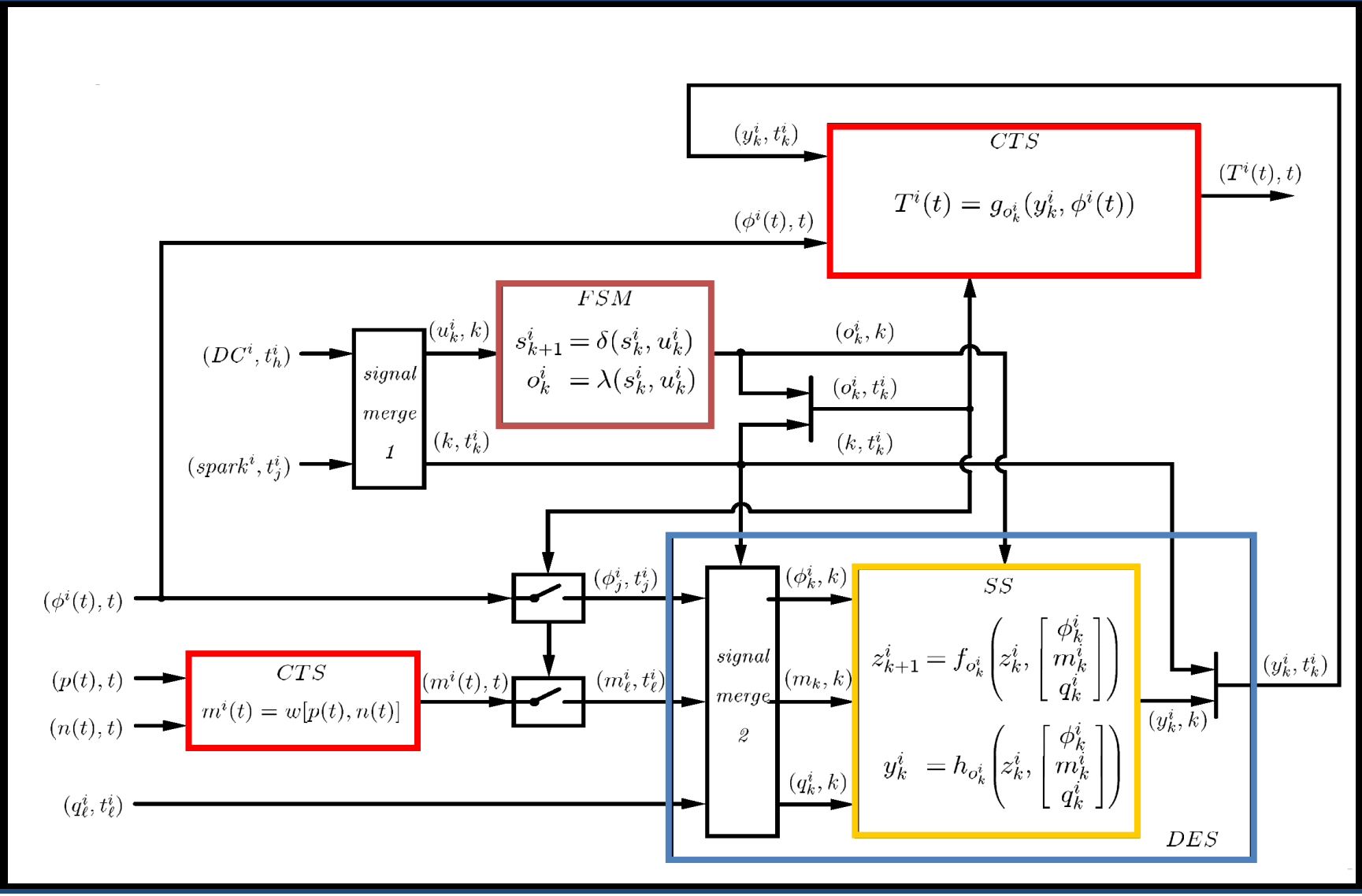
FSM for a single cylinder

negative spark advance:
the spark is given after
the TDC between C and E

positive spark advance:
the spark is given before
the TDC between C and E



Single-cylinder hybrid model



System - Software Interaction

- Ariane V launched on 4th June 1996. It exploded 37s after launch
- The program had been running for 10 years, costing $7 billions
- Software worked perfectly on Ariane IV, the same was used in Ariane V

What had changed,
was the physical
system around
the software …

A glimpse to the future…

- 7000 billions components will serve 7 billions people in 2017 !!!

(Source: WirelessWorldResearchForum (WWRF))

- 1000 wireless components per person?

ICT Emergent scenario

Infrastructural core

Sensory swarm

Mobile access

A relevant Industrial Application: Mining Ventilation Control

DEWS CENTER OF EXCELLENCE

BOLIDEN

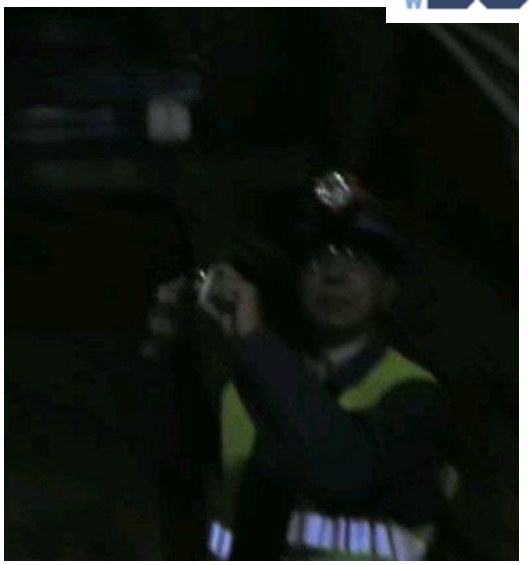
- Mining and smelting company
- 3 500 MEUR turnover
- 4 500 employees
- 3 Swedish and 1 Irish mine

Garpenberg mine

- Mining since 9th century
- 1000K ton ore/yr
  - 58K Zn, 21K Pb, 0.56K Cu, 0.1K Ag, 0.2 Au
- 1 100 m deep
- 280 employees

HYCON
AMING HETEROGENEITY AND COMPLEXITY

A relevant Industrial Application: Mining Ventilation Control



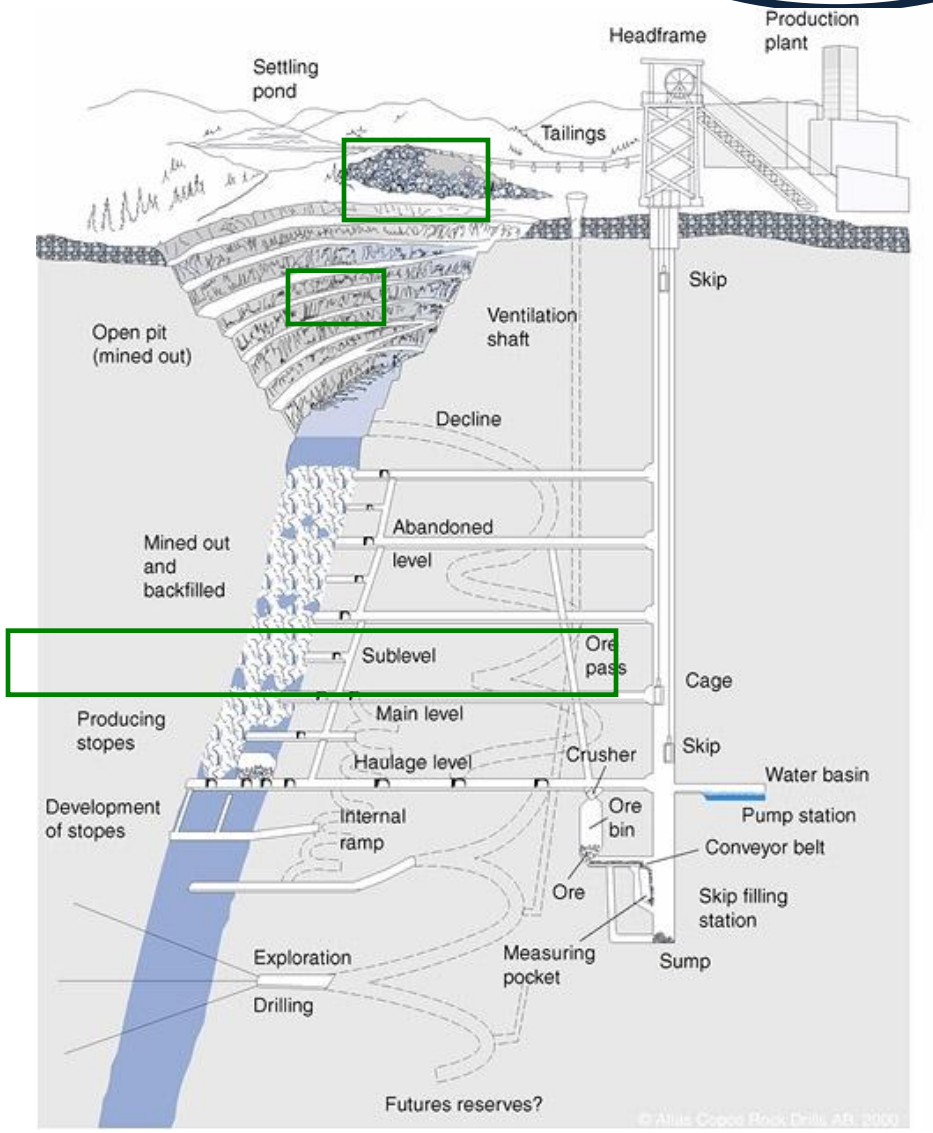
Boliden, 6 Mar 2007

Garpenberg, 14 Sep 2007

Mining Process

Mining phases:

- Drilling and blasting
- Ore transportation
- Ore crushing

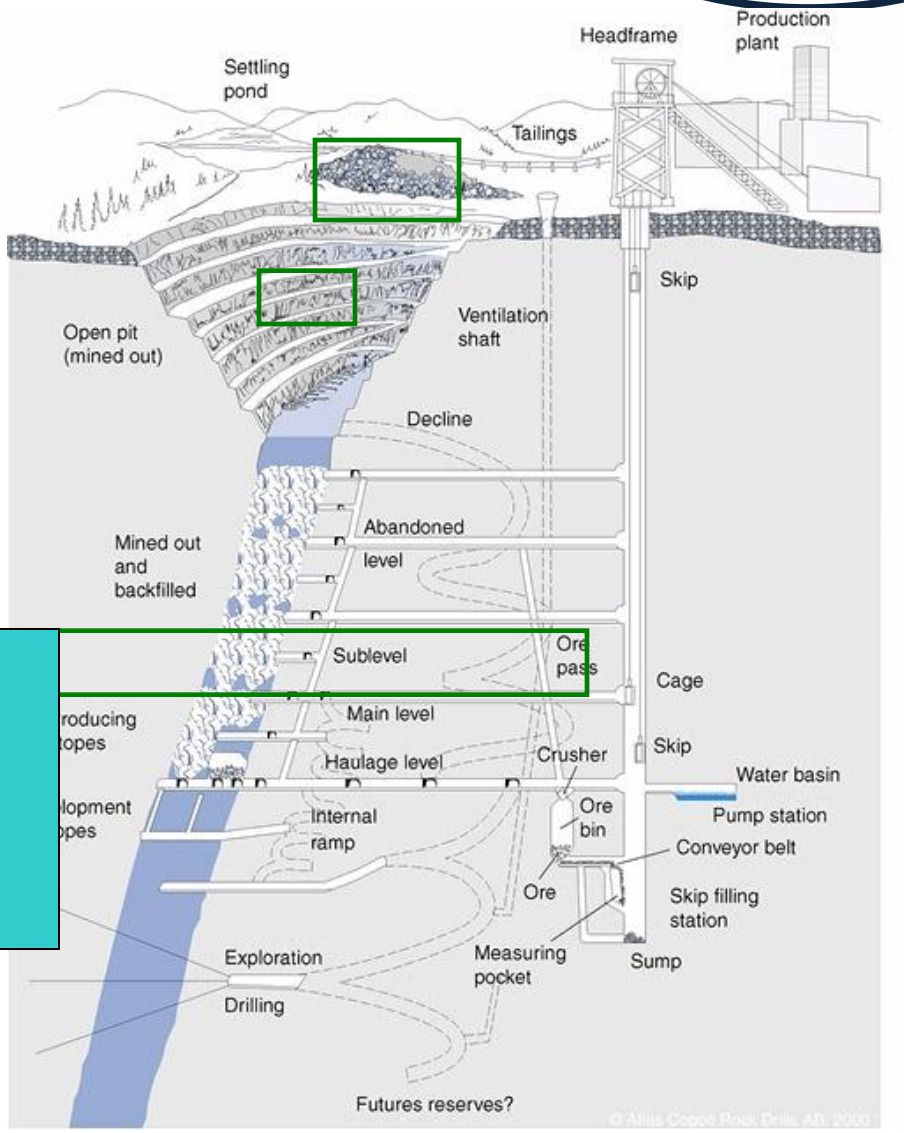


Mining Process

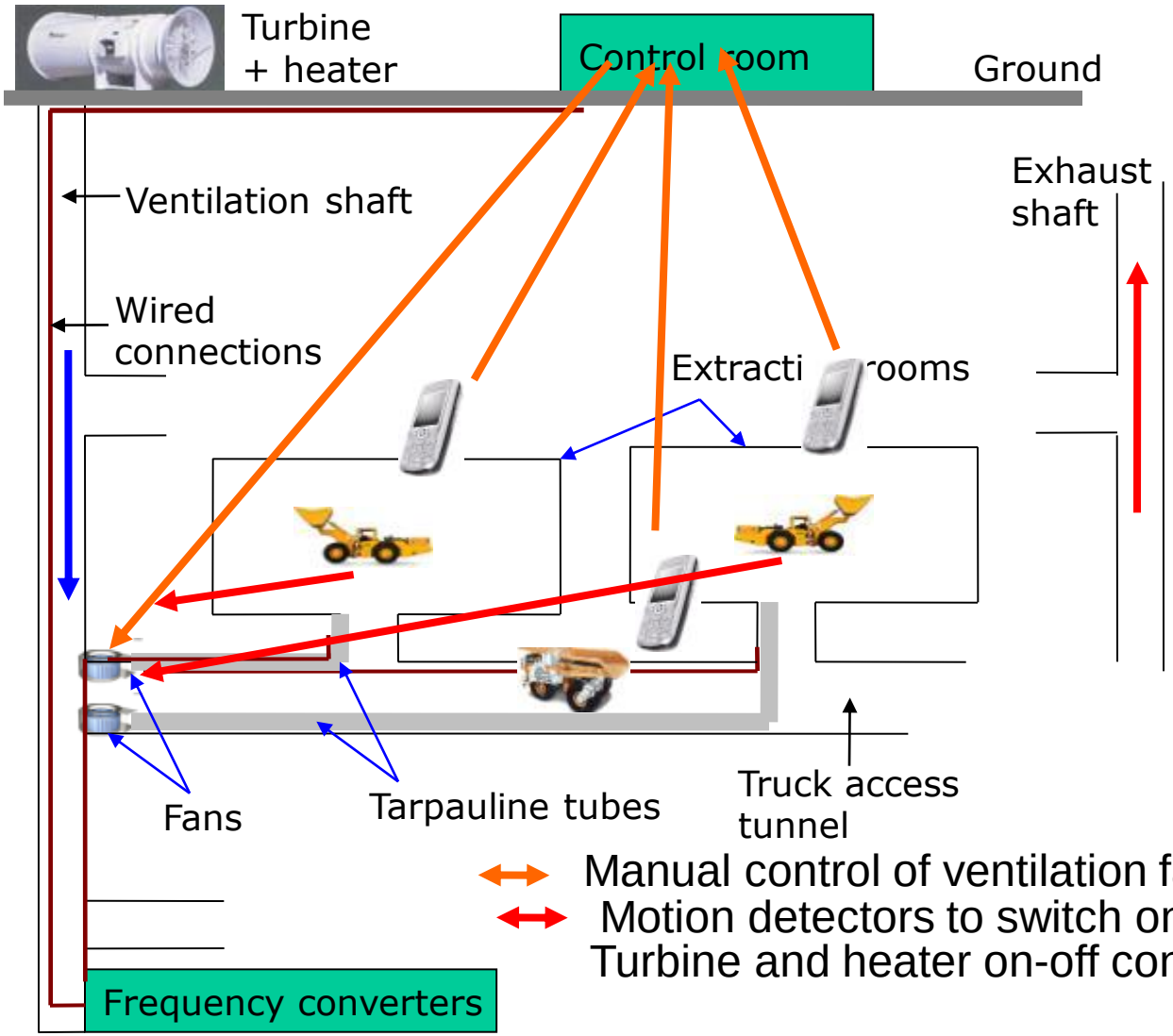
Mining phases:

- Drilling and blasting
- Ore transportation
- Ore crushing

- Ventilation represents >50% of power consumption in mining
- Ventilation control system is often poor or non-existing



Mining Ventilation Control: Today's Architecture



Manual control of ventilation fans over walkie-talkie

Motion detectors to switch on fans for safety

Turbine and heater on-off control based on timers

Mining Ventilation Control: Today's Architecture

Turbine
+ heater

Control room

Ground

Ventilation shaft

Wired
connections

Extracti rooms

Exhaust
shaft

- No automatic control, but max ventilation power during ore extraction
- No continuous monitoring of air quality
- No localization system

Fans

Tarpauline tubes

Truck access
tunnel

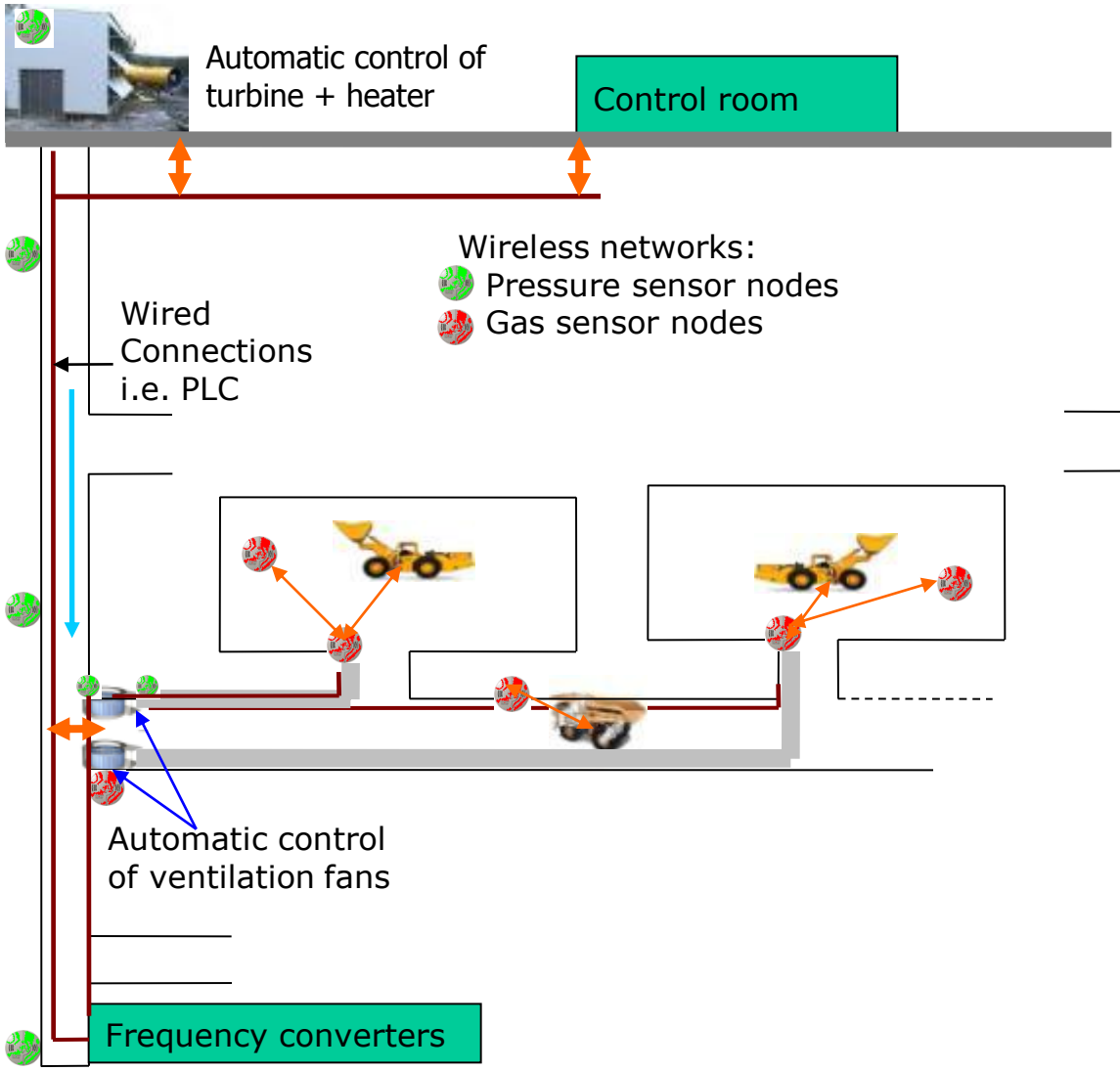
Frequency converters

↔ Manual control of ventilation fans over walkie-talkie

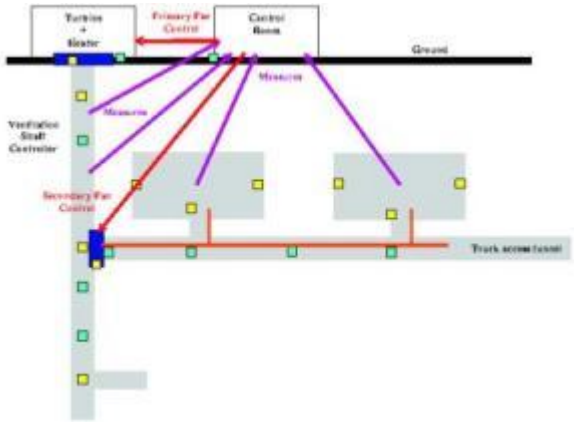
↔ Motion detectors to switch on fans for safety

Turbine and heater on-off control based on timers

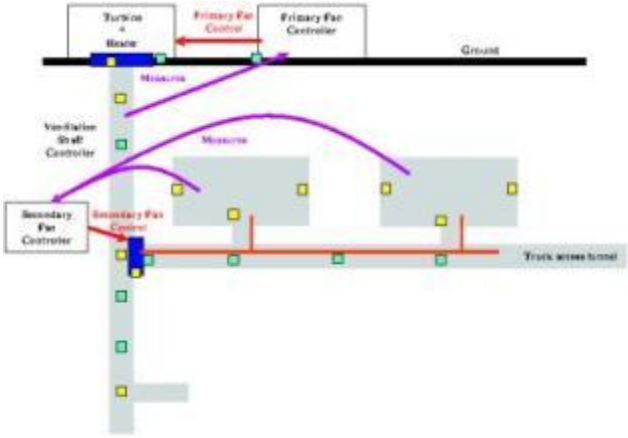
Wireless Control Architecture



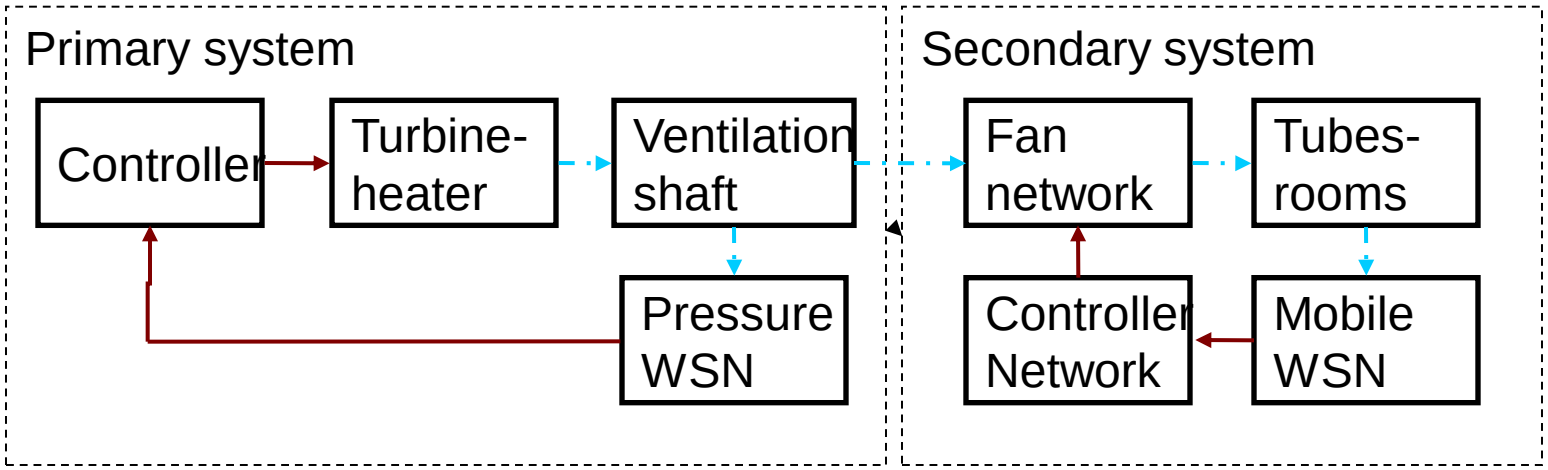
Centralized



Decentralized



Control Architecture and Objectives



Objectives:

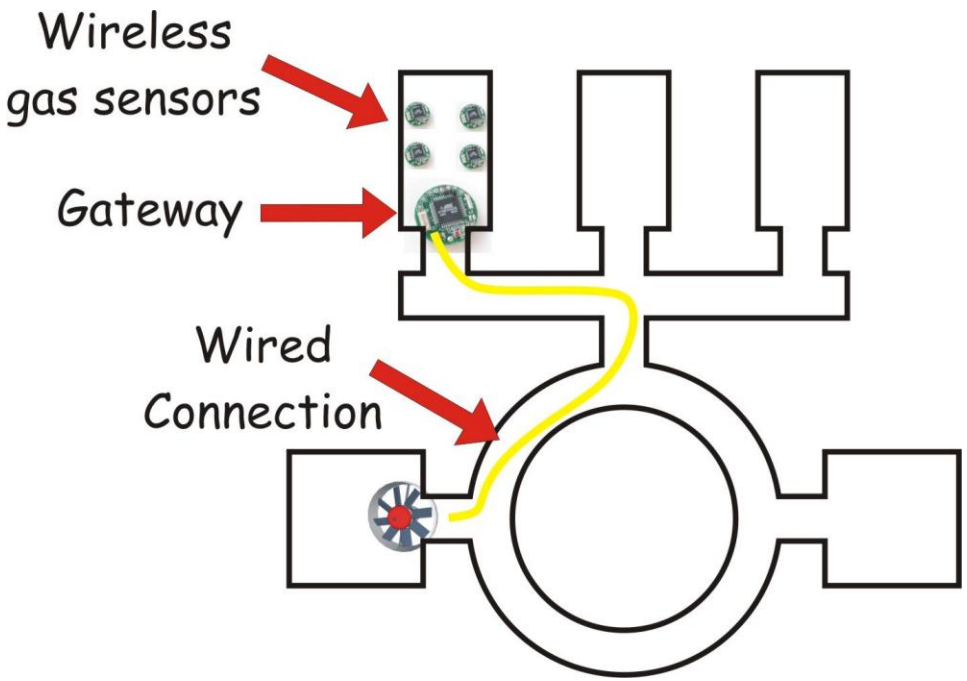
- Control air quality (O_2, NOx and CO) in extraction rooms at suitable level
  - Regulate turbine and heater to provide suitable airflow pressure at ventilation fans
  - Regulate ventilation fans to ensure air quality in extraction rooms
- Safety through wireless networking for personal communication and localization

Design constraints:

- Physical interconnections, actuators limitations and networking capabilities
- Sensing capabilites: O_2, NOx, COx, pressure and temperature

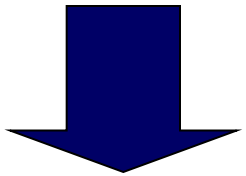
Secondary System

- Secondary fan: boundary condition with primary system, rotational speed
- Tarpaulin tube: size, curves, junctions
- Extraction room: size, number of working trucks



Secondary System: control challenges

- Secondary system may change even in the very same mine, complex geometry
- Safety must be guaranteed with good margins!



- Our solution needs to be simple
- Flexible, over-approximating model

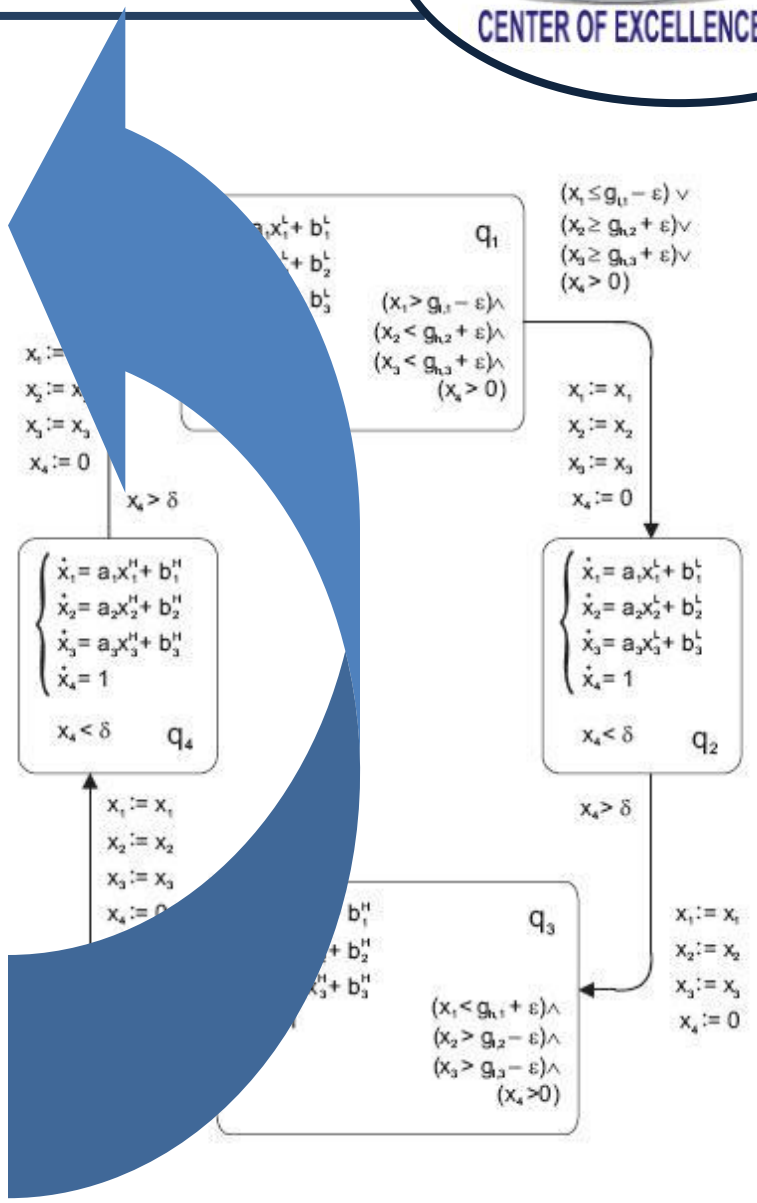
Closing the loop on wireless

Choose simple threshold control strategy

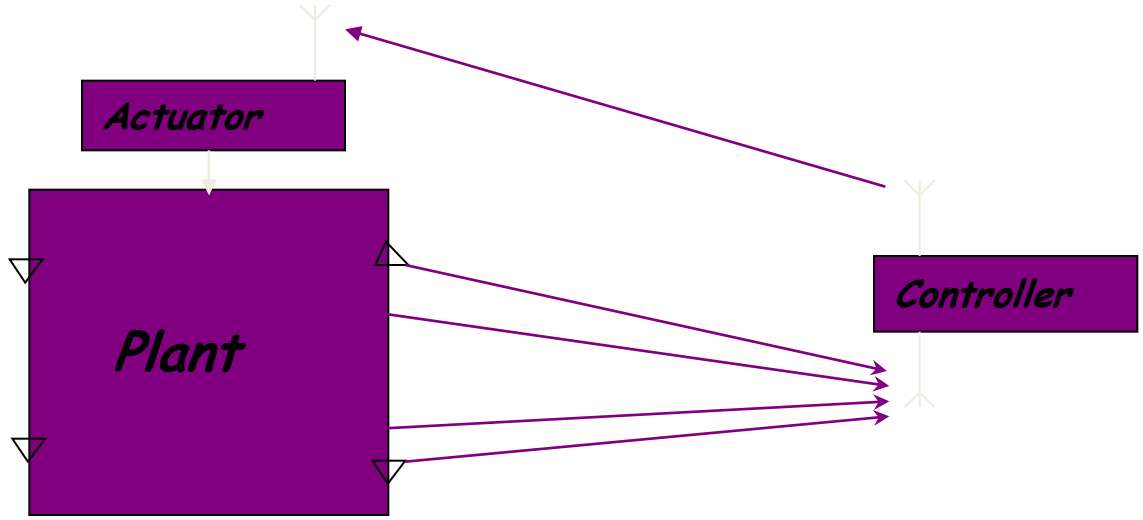
Derive autonomous affine hybrid model of the closed loop system

Formalize safety and comfort control specifications by means of temporal logic formulae (model checking)

Perform automatic verification of the closed loop system using abstraction techniques



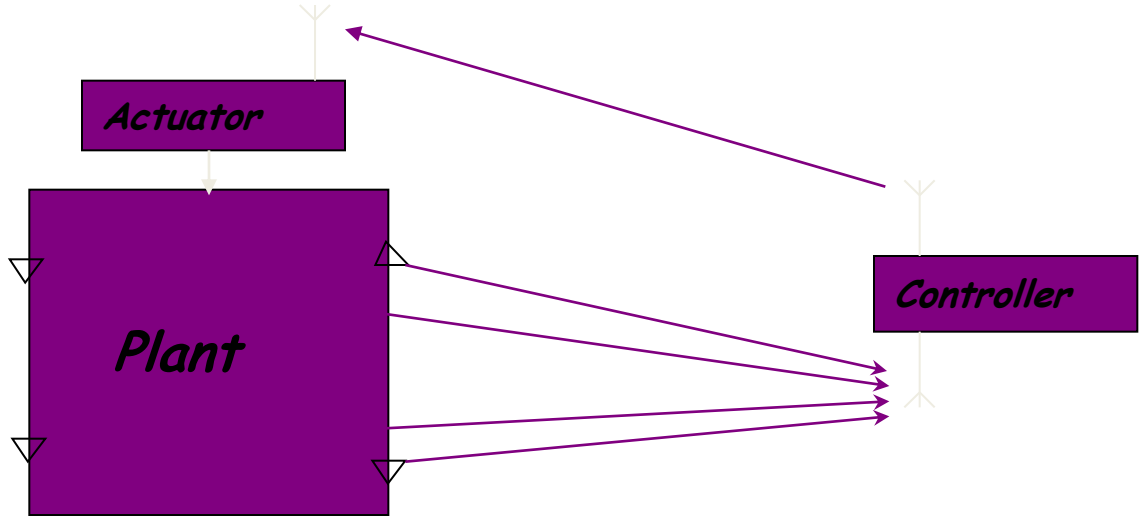
A communication or a control problem?



- **Control specifications:**
  - Safety or Liveness
  - Stability
  - Robustness

- **Wireless Network Issues:**
  - Packet delay
  - Packet loss
  - Limited spectrum
  - Variable channel gain
  - Interference

A communication or a control problem?



Joint Communication and Control design:

- Robustness of controller w.r.t. wireless network characteritics
- Communication scheme must satisfy constraints inferred by control specifications

Triple C Convergence

Communication Systems

Control Theory

Computer Science

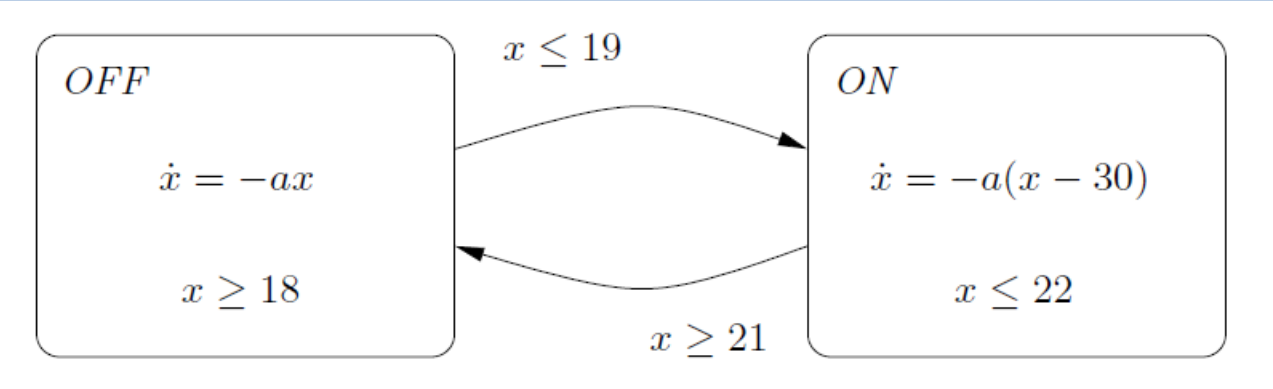
Networked hybrid control systems

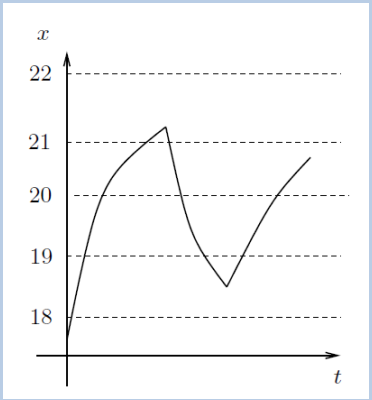
System Modeling

- Traditional approach: time-invariant systems, discrete-time and continuous-time models
- Importance of “software” and “computing systems” mixed with “physical systems”, e.g. electric circuits, mechanical control systems, etc.; these “components” form “Cyber-Physical Systems”
- A RLC circuit and a software are both “systems”
- In some cases, the state is a continuous variable in a differential equation, in other cases it corresponds to the values stored in computer memories and CPU registers
- Distributed and Networked Systems focus on communication

Interaction between continuous dynamical control systems and finite state machines

The Thermostat Model





From “Lecture Notes on Hybrid Systems” , J. Lygeros

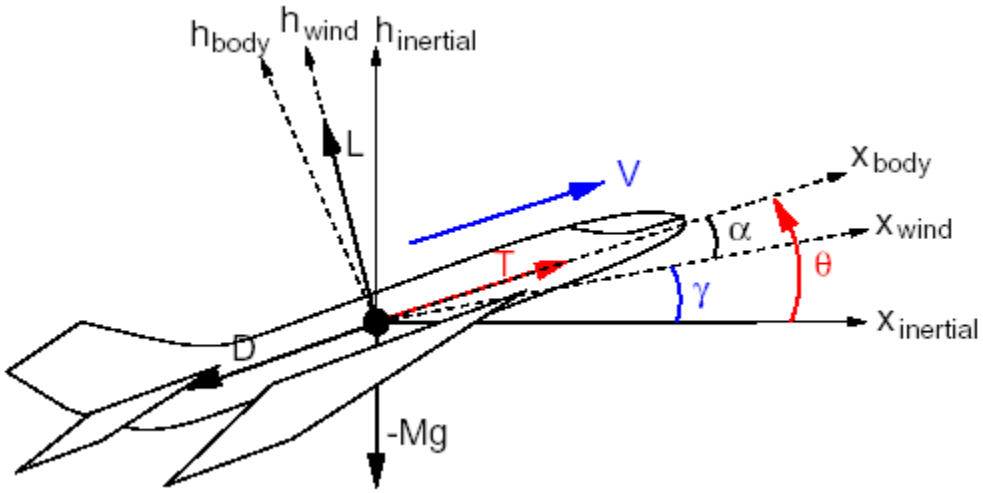
Why Hybrid Systems?

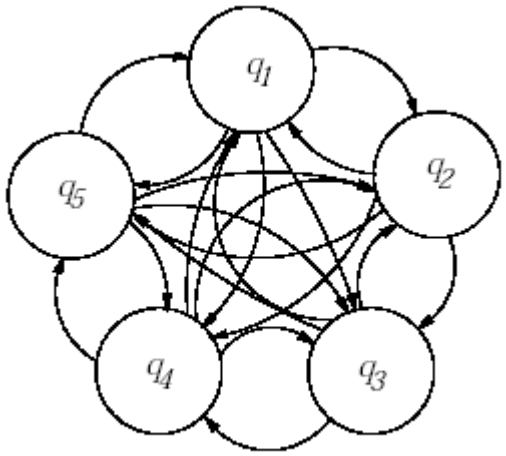
- **Continuous systems with a phased operation:**
  - bouncing ball
  - walking robots
  - biological cell growth and division

- **Continuous systems controlled by discrete logic:** need for design techniques that can guarantee safety and performance specifications of *embedded systems*, or systems that couple discrete logic with the analog physical environment.
  - thermostat
  - chemical plants with valves, pumps
  - control modes for complex systems, eg. intelligent cruise control in automobiles, aircraft autopilot modes

- **Coordinating processes:** many interacting subsystems (multi-agent systems) typically feature continuous controllers to optimize performance of individual agents, and coordination among agents to compete for scarce resources, resolve conflicts, etc.
  - air and ground transportation systems
  - swarms of micro-air vehicles

Auto-pilot of Boeing B737





5 flight modes

Thrust T, Pitch θ

Flight path γ, Speed V

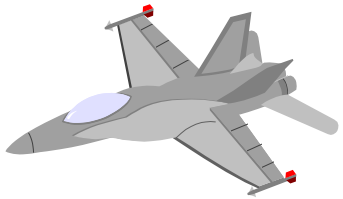
Why Hybrid Systems?

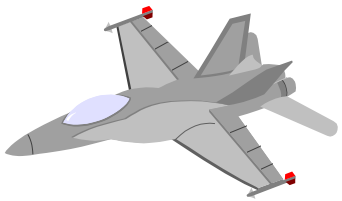
- **Continuous systems with a phased operation:**
  - bouncing ball
  - walking robots
  - biological cell growth and division

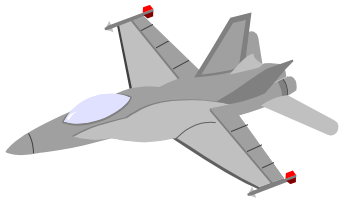
- **Continuous systems controlled by discrete logic:** need for design techniques that can guarantee safety and performance specifications of *embedded systems*, or systems that couple discrete logic with the analog physical environment.
  - thermostat
  - chemical plants with valves, pumps
  - control modes for complex systems, eg. intelligent cruise control in automobiles, aircraft autopilot modes

- **Coordinating processes:** many interacting subsystems (multi-agent systems) typically feature continuous controllers to optimize performance of individual agents, and coordination among agents to compete for scarce resources, resolve conflicts, etc.
  - air and ground transportation systems
  - swarms of micro-air vehicles

Multi-agent control







- Large number of semiautonomous agents
- Coordinate to
  - Make efficient use of common resource
  - Achieve a common goal

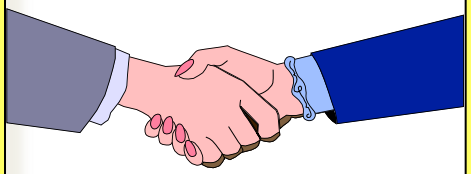
- Individual agents have various modes of operation
- Agents optimize locally, coordinate to resolve conflicts
- System architecture is hierarchical and distributed
- Safety critical systems

***Challenge*: Develop models, analysis, and synthesis tools for designing and verifying the safety of multi-agent systems**

Why Hybrid Systems?

Control Theory

Continuous systems
Stability, control
Feedback, robustness



Computer Science

Transitions systems
Concurrency models
Composition, abstraction

Hybrid Systems

Software controlled systems
Multi-modal systems
Embedded real-time systems
Multi-agent systems

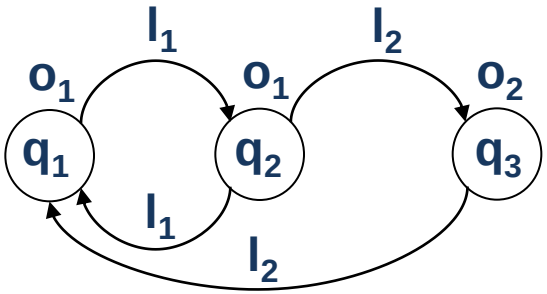
Dealing with heterogeneity

Definition A Labelled Transition System (for short also called LTS or System) is a tuple:

$$T = (Q, Q_0, L, \longrightarrow, O, H),$$

consisting of:

- a set of states Q
- a set of initial states Q_0
- a set of labels L
- a transition relation $\longrightarrow \subseteq Q \times L \times Q$
- a set of outputs O
- an output function $H: Q \rightarrow O$



We will follow standard practice and denote $(q, l, q') \in \longrightarrow$ by $q \longrightarrow q'$ l

When $Q_0 = Q$ we write the LTS in a more compact notation as $T = (Q, L, \longrightarrow, O, H)$

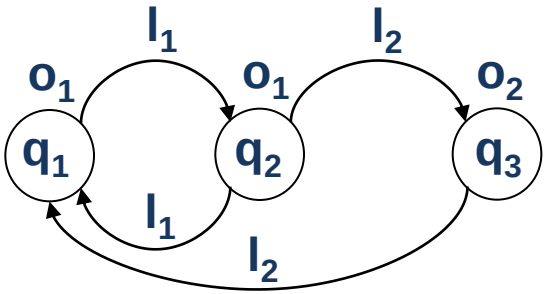
Dealing with heterogeneity

Definition A Labelled Transition System (LTS also called System) is a tuple:

$$T = (Q, Q_0, L, \longrightarrow, O, H),$$

consisting of:

- a set of states Q
- a set of initial states Q_0
- a set of labels L
- a transition relation $\longrightarrow \subseteq Q \times L \times Q$
- a set of outputs O
- an output function $H: Q \rightarrow O$

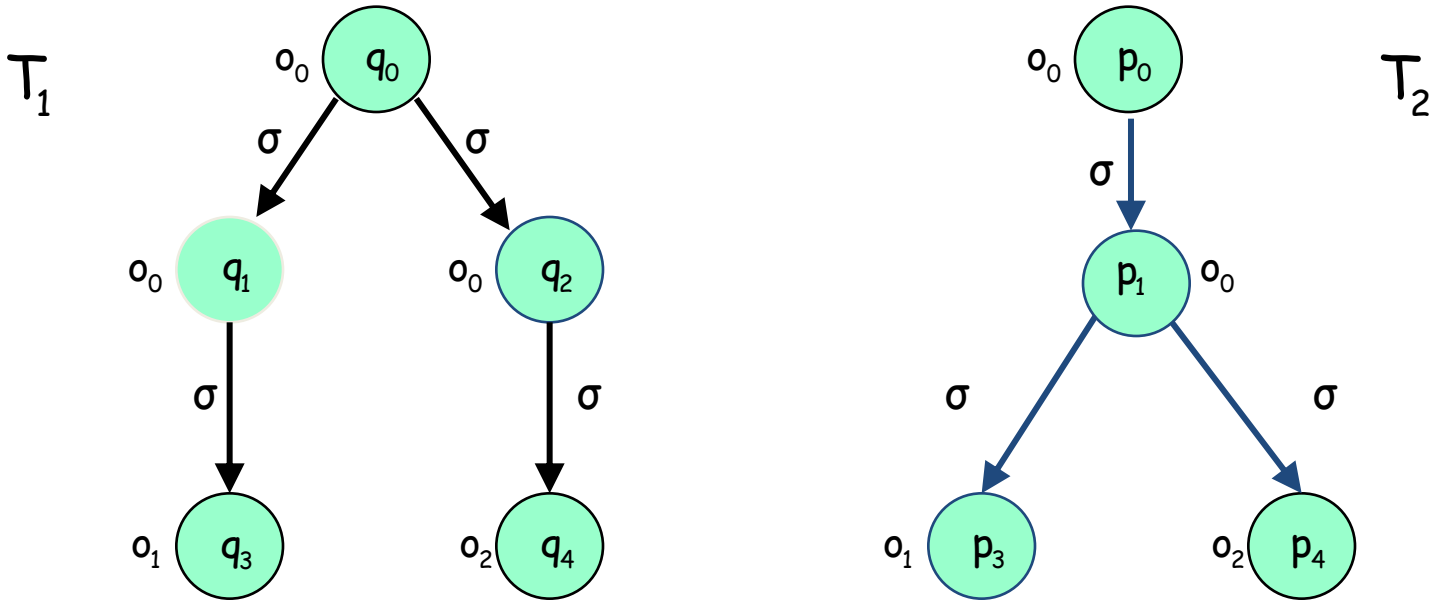


T is said:

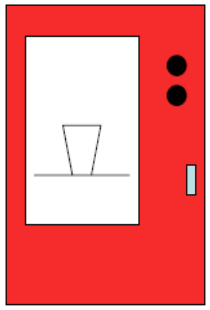
- countable if Q and L are countable sets
- symbolic/finite if Q and L are finite sets
- deterministic if for q and l there exists at most one p such that $q \xrightarrow{l} p$
- non-blocking if for any q there exist at least one l and one p such that $q \xrightarrow{l} p$
- metric if O is a metric space

Example

Are T_1 and/or T_2 deterministic? Symbolic?

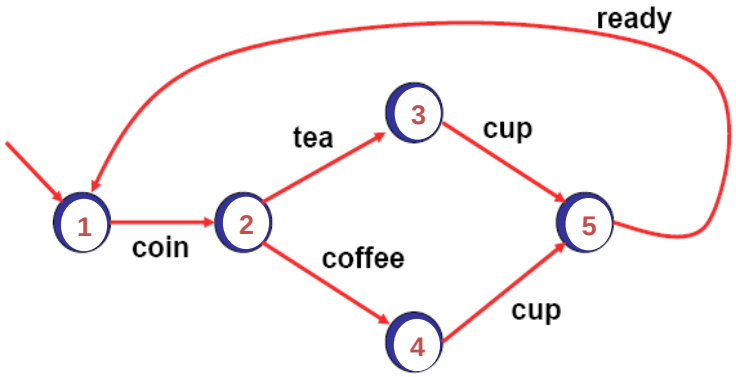


Vending machine

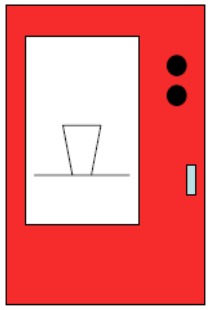


Vending machine

1. Insert coin(s)
2. Choose tea or coffee
3. Put the cup on the tray

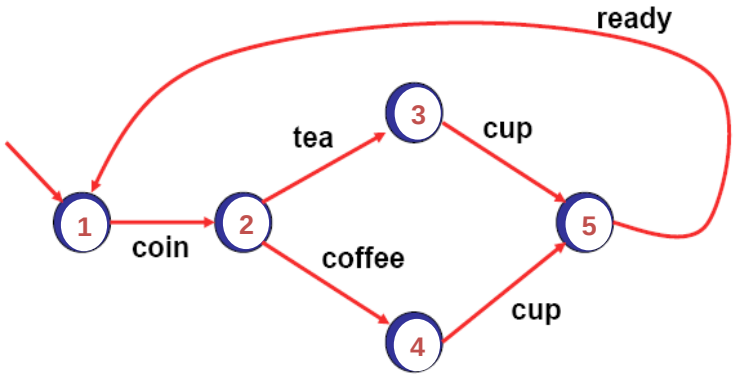


Vending machine



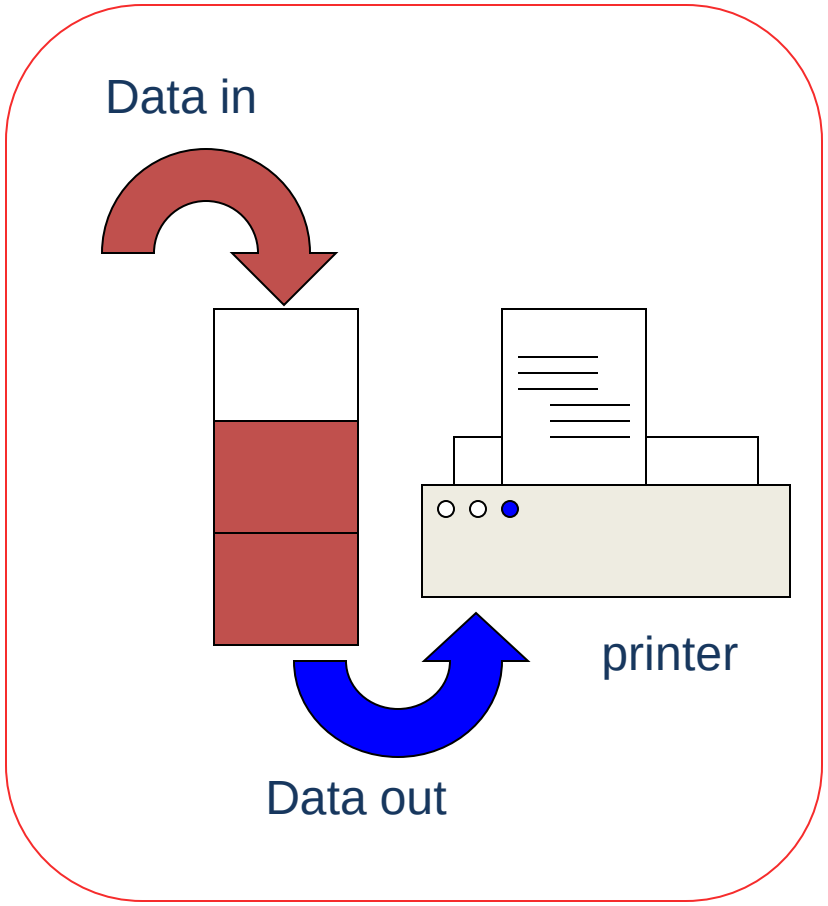
Vending machine

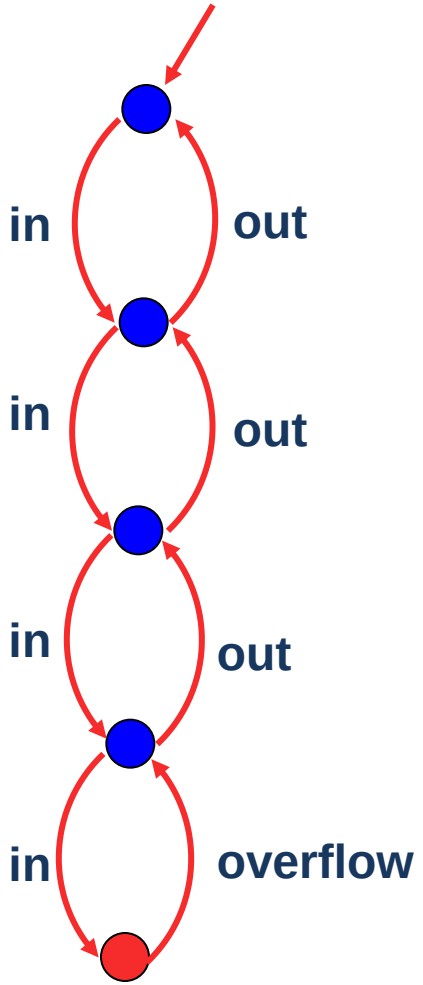
1. Insert coin(s)
2. Choose tea or coffee
3. Put the cup on the tray



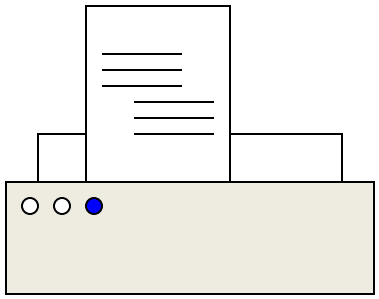
... Event driven versus time driven !

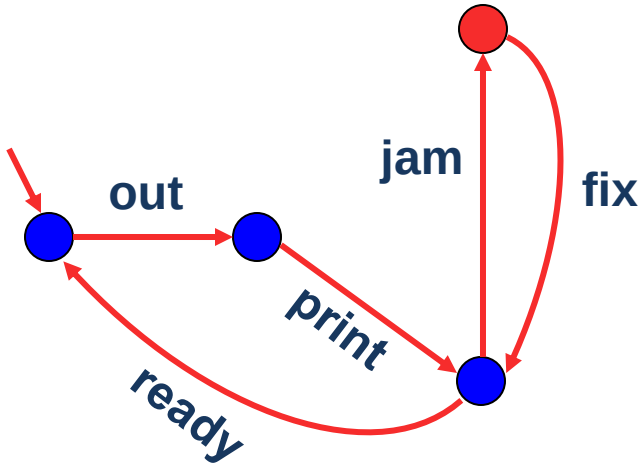
A printer data buffer





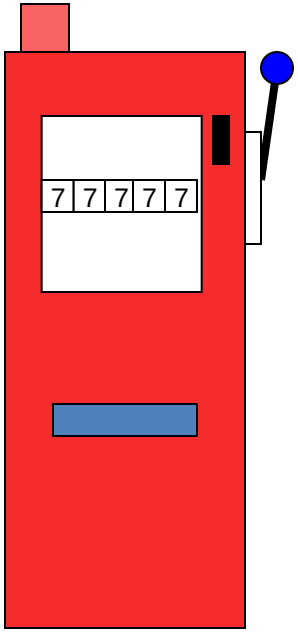
A printer



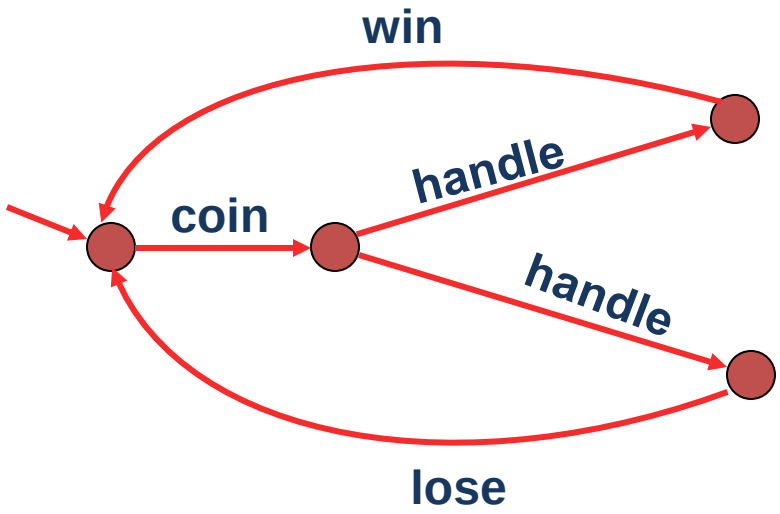


The printer receives data from the buffer, and print it out. Once the printout is ready, the printer is ready to receive new data. While printing, the paper can jam and need to be fixed before the printing process can resume.

A slot machine



1. Insert coin
2. Pull handle
3. Win if the combination is good, otherwise lose.



Modeling features

- Events are time-abstract.
- Just like modeling of continuous systems, the level of detail is 'modeler dependent'.
- Compositionality is possible (to be discussed later).
- There can be non-determinism.

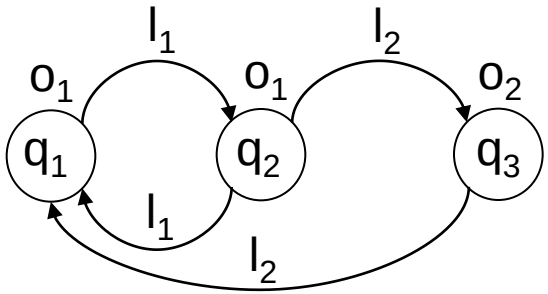
A unifying framework: discrete processes

Definition A Labelled Transition System (for short also called LTS or System) is a tuple:

$$T = (Q, Q_0, L, \longrightarrow, O, H),$$

consisting of:

- a set of states Q
- a set of initial states Q_0
- a set of labels L
- a transition relation $\longrightarrow \subseteq Q \times L \times Q$
- a set of outputs O
- an output function $H: Q \rightarrow O$



We can formally model software as a LTS. The states are all the possible memory configurations and the transition relation describes how the memory contents are changed by the execution of instructions.

Example: Software

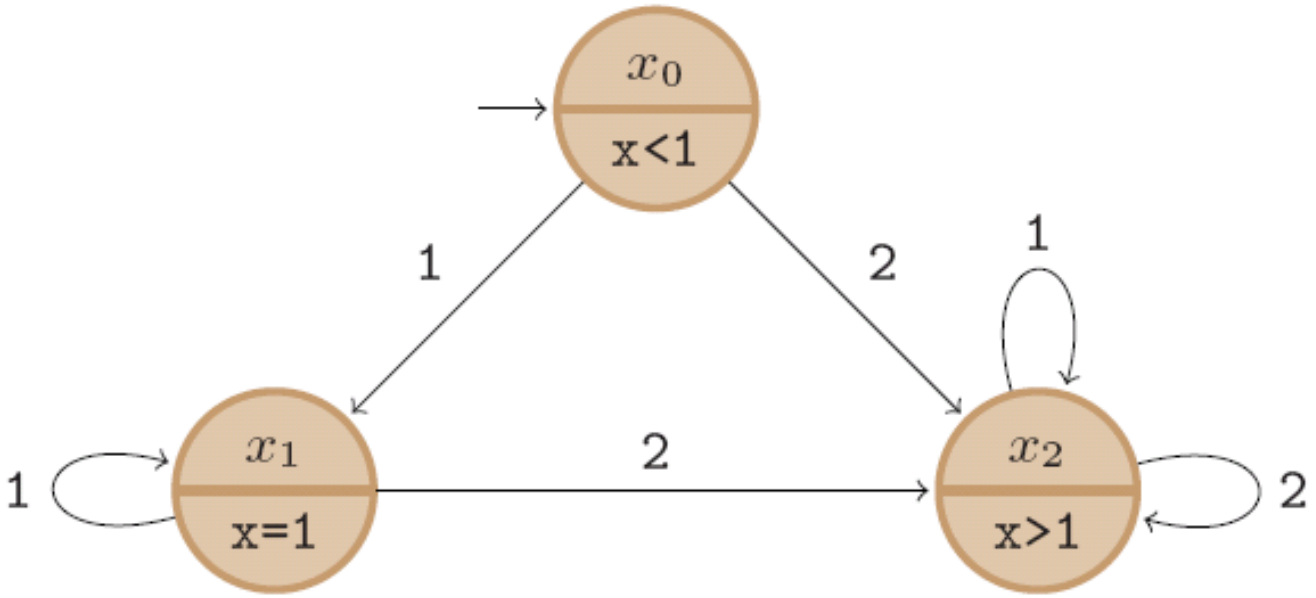
Suppose that we want to compute the average of a stream of numbers but we do not know a priori the length of the stream. One possible way to compute the average is to update the average upon the reception of a new number in the stream.

$x{:=}0;$
$n{:=}0;$
while *true* **do**
  $y{:=}\text{read}(\text{input});$
  $x := x\frac{n}{n+1} + y\frac{1}{n+1};$
  $n := n + 1;$
end

Taken from Tabuada, Verification and Control of Hybrid Systems, Springer Verlag, 2009.

Example: Software

Assume that we are interested in knowing if x is smaller, equal, or greater than 1 when y is restricted to assume values in the set {1,2}. One possible finite-state model capturing the dynamics of x is represented in the following:



Taken from Tabuada, Verification and Control of Hybrid Systems, Springer Verlag, 2009.

A unifying framework: continuous processes

Given a control system Σ:

$$\dot{x} = f(x,u) \qquad x \in R^n,\ u \in R^m$$

we can define the following LTS

$$T(\Sigma) = (Q, L, \longrightarrow, O, H),$$

where:

- $Q = R^n$
- L is the collection of control signals $u : R \rightarrow R^m$
- $p \xrightarrow{u} q$, if $x(\tau,p,u) = q$ for some $\tau \geq 0$
- $O = R^n$
- H is the identity function

$T(\Sigma)$ captures all information contained in Σ

… by using similar arguments I can associate a LTS to a hybrid system!

A unifying framework: continuous processes

Given a control system Σ:

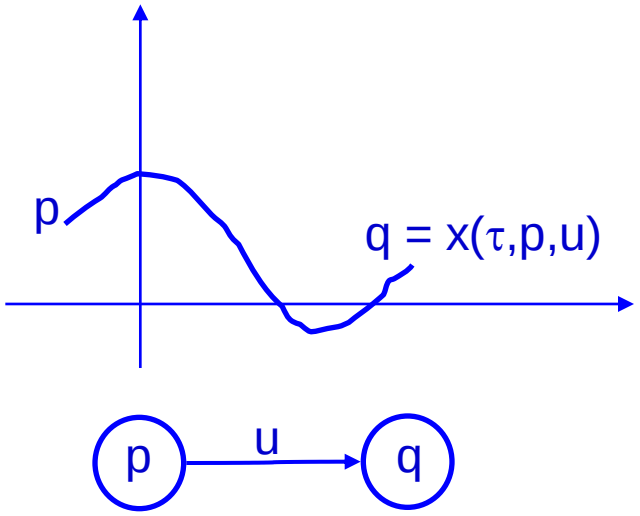
$$\dot{x} = f(x,u) \qquad x \in R^n,\ u \in R^m$$

we can define the following LTS

$$T(\Sigma) = (Q, L, \longrightarrow, O, H),$$

where:

- $Q = R^n$
- L is the collection of control signals $u : R \rightarrow R^m$
- $p \xrightarrow{u} q$, if $x(\tau,p,u) = q$ for some $\tau \geq 0$
- $O = R^n$
- H is the identity function



$T(\Sigma)$ captures all information contained in Σ

… by using similar arguments an LTS can be associated to a hybrid system!

A unifying framework: continuous processes

Given a control system Σ:

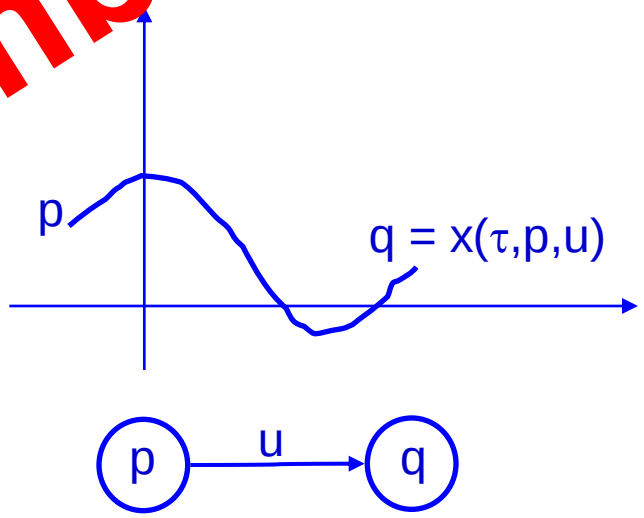
$$\dot{x} = f(x,u) \qquad x \in R^n,\ u \in R^m$$

we can define the following LTS

$$T(\Sigma) = (Q, L, \longrightarrow, O, H),$$

where:

- $Q = R^n$
- L is the collection of control signals $u: R \to R^m$
- $p \xrightarrow{u} q$, if $x(\tau,p,u) = q$ for some $\tau \geq 0$
- $O = R^n$
- H is the identity function



$T(\Sigma)$ captures all information contained in Σ

… by using similar arguments I can associate a LTS to a hybrid system!

T(Σ) is not symbolic !

Dealing with heterogeneity

Software modeled by LTS

States are all memory configurations and transition relation describes how memory contents change with execution of instructions

Nonlinear control systems modeled by LTS

Given the nonlinear control system

$P : dx/dt = f(x,u),\ x \in R^n,\ u \in R^m$

consider the LTS

$T(P) = (Q, L, \longrightarrow, O, H),$

where:

- $Q = R^n$
- L is the collection of control signals u
- $q \xrightarrow{u} p$, if $x(\tau,q,u) = p$ for some $\tau \geq 0$
- $O = R^n$
- H is the identity function