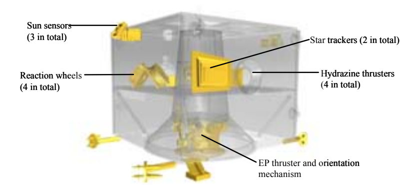
Modeling and Analysis of Hybrid Control Systems

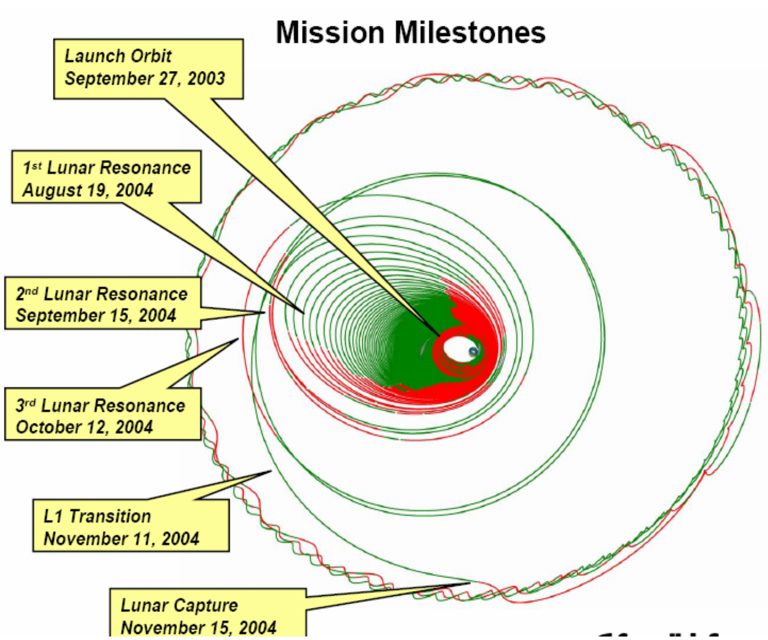
Karl Henrik Johansson
School of Electrical Engineering
Royal Institute of Technology
Stockholm, Sweden
www.s3.kth.se/~kallej

MOVEP 2006, Bordeaux, France

Control of the SMART-1 spacecraft

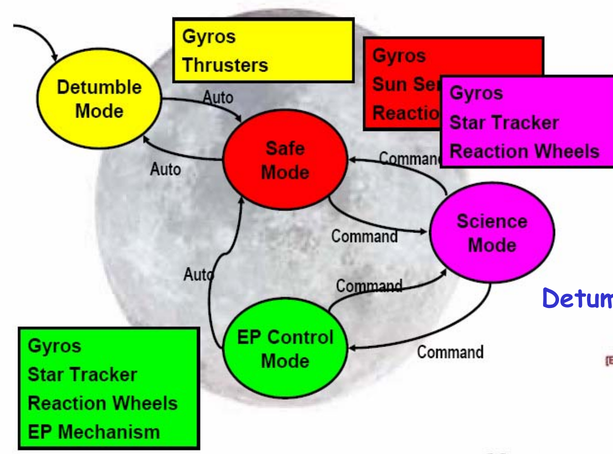
- First European lunar mission, launched Sep 2003
- Go to the Moon using Electric Primary Propulsion
- Advanced orbit and attitude control system





Bodin, Swedish Space Agency, 2005

Control modes for orbit and attitude



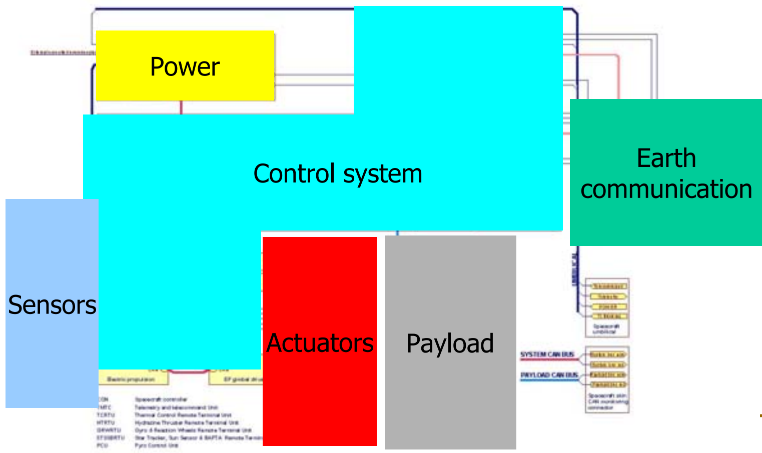
Detumble Mode → Safe Mode

Switch between different control laws
depending on commands and autonomous actions

Bodin, Swedish Space Agency, 2005

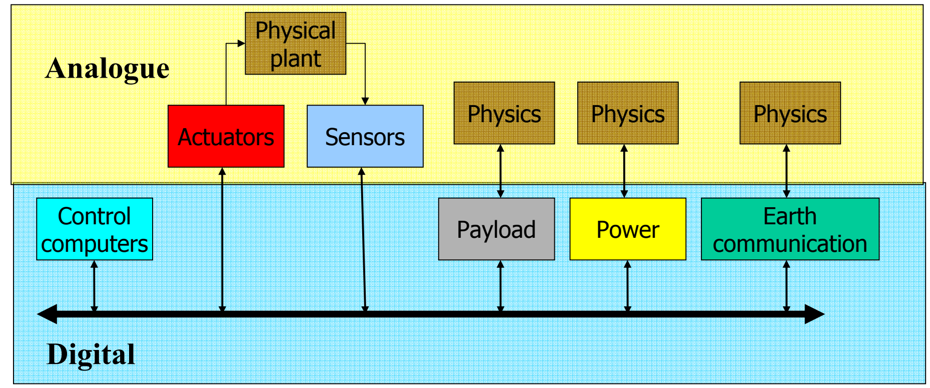
Architecture of SMART-1 spacecraft

- *A networked control system with tight interactions between computation, communication and control*



Architecture of SMART-1 spacecraft

- Coupling between **analogue physics** and **digital computations**



Outline: Hybrid control systems

- Introduction
  - What is a hybrid system
  - Motivating examples
- Models
  - Hybrid automata

} Lecture I

- Control
  - Stability and stabilization
- Application
  - Control of network traffic
- Summary
  - Outlook
  - Further reading

} Lecture II

What is a hybrid system?

- *A hybrid system is a dynamical system with interacting time-triggered and event-triggered dynamics*
- E.g., differential equations and finite automata

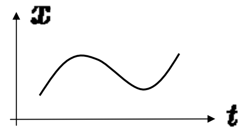
$$\dot{x} = f(x,u) \quad \text{and} \quad q^+ = g(q,v)$$

- Mixture of analogue and digital models of computations

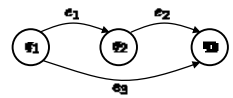
Control systems

- Time-triggered $\dot{x} = f(x, u)$

$$x : [0, \infty) \to \mathbf{R}^n, u : [0, \infty) \to \mathbf{R}$$



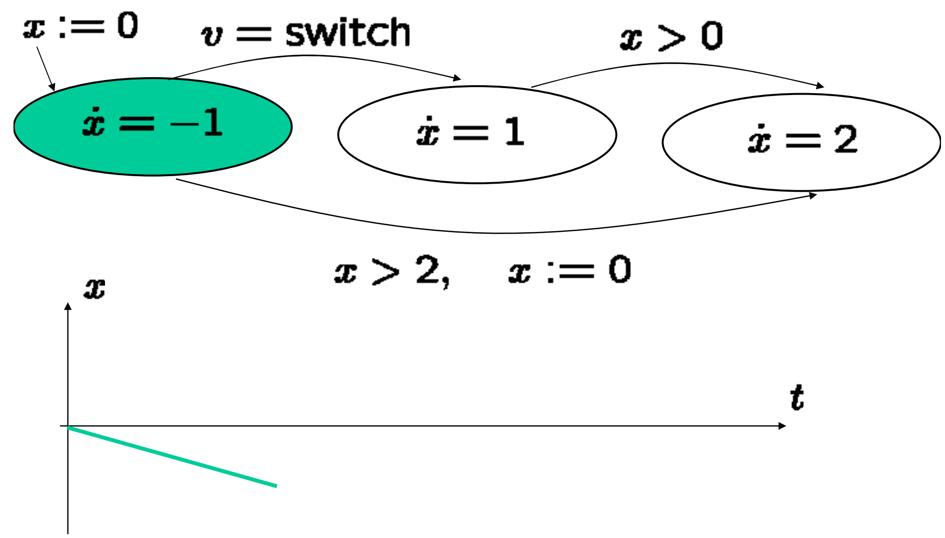
- Event-triggered $q^+ = g(q, v)$



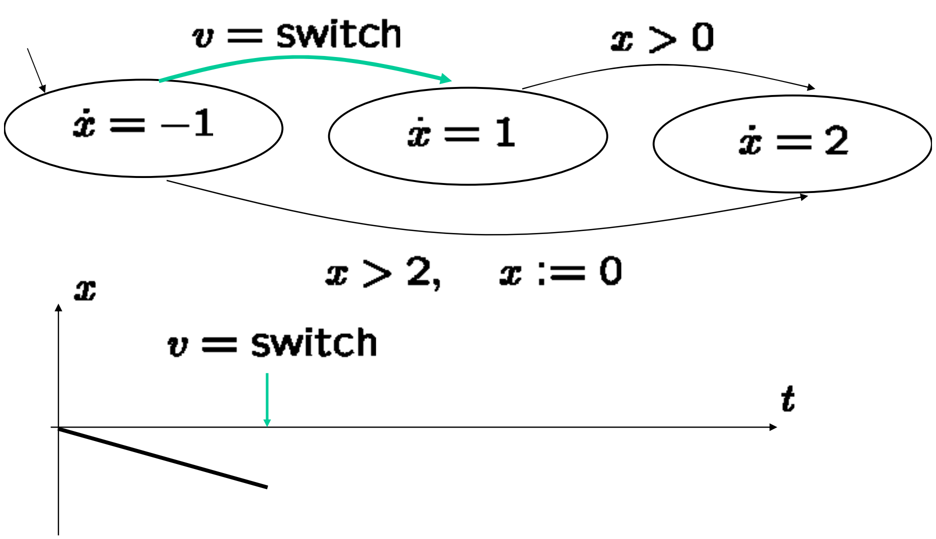
q_1, e_1, q_2, e_2, q_3

$$q : \mathbf{Z}^+ \to \{q_1, \ldots, q_N\}, v : \mathbf{Z}^+ \to \{e_1, \ldots, e_K\}$$

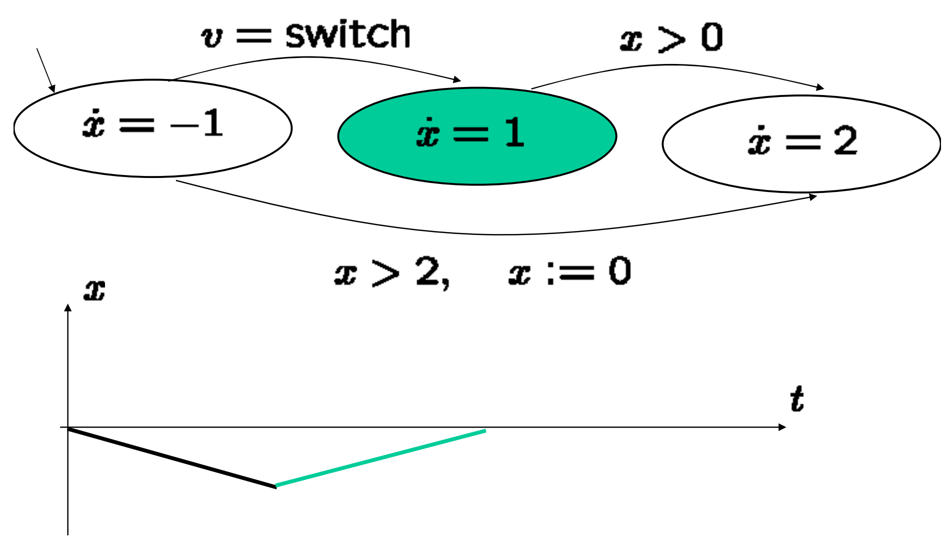
Example of a hybrid system



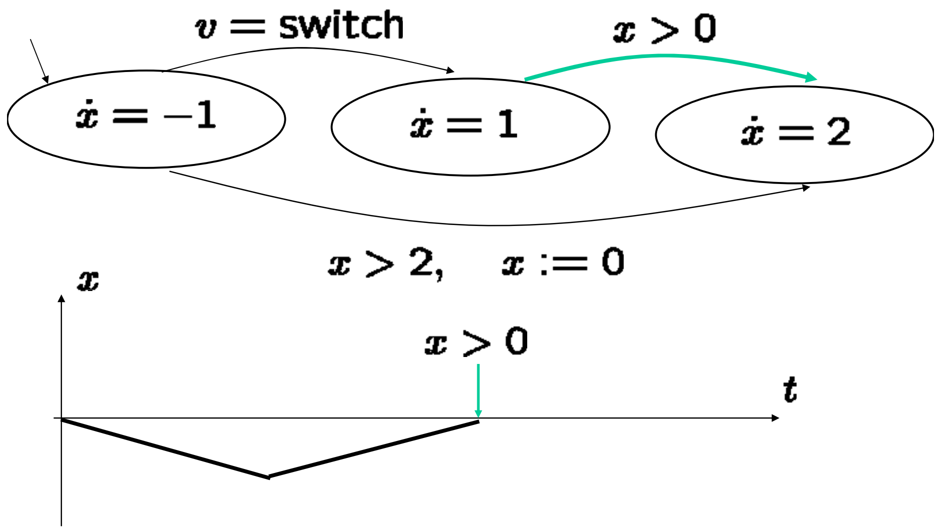
Example of a hybrid system



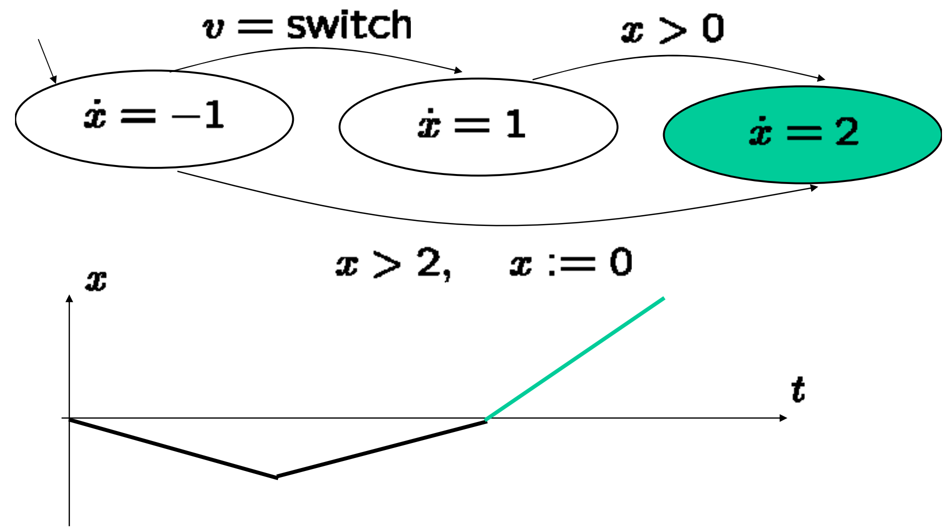
Example of a hybrid system



Example of a hybrid system



Example of a hybrid system

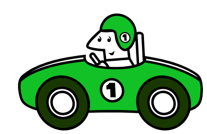


Motivating examples

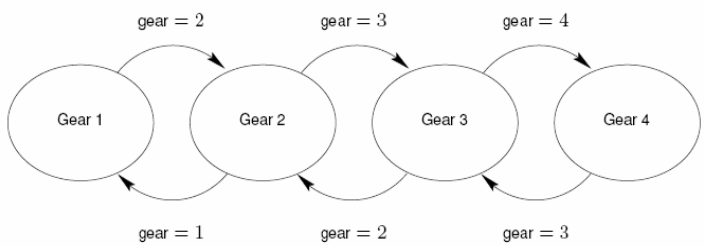
- Automatic gear box
- Rocking block
- Vacuum cleaning
- Network congestion control
- Networked embedded systems

Automatic gear box

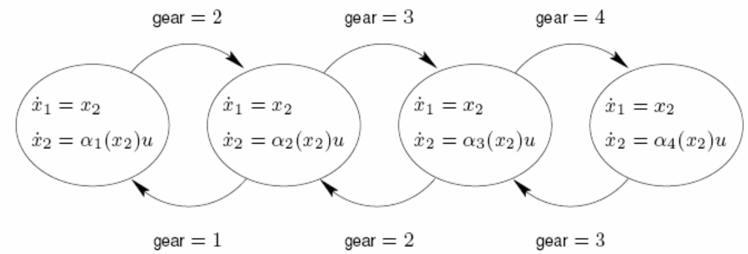
Task: Design the control system for an automatic gearbox



Discrete event system



Hybrid system



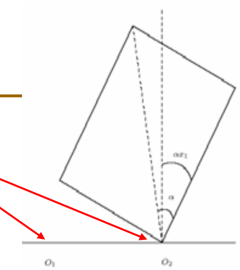
Rocking block

Hybrid models capture mechanical impacts and other discontinuous dynamics

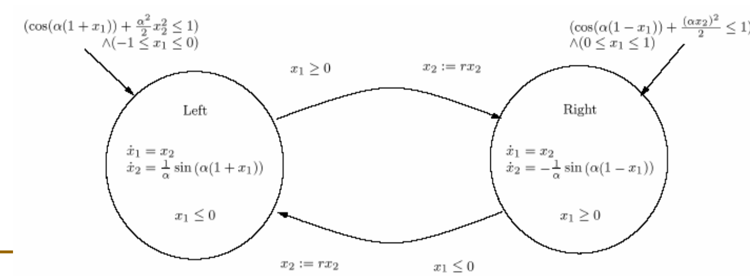
www.animats.com

Rocking block

- Rocking block rotates around one of two pivot points
- Impacts represented as discrete transitions
- System may show complex dynamics

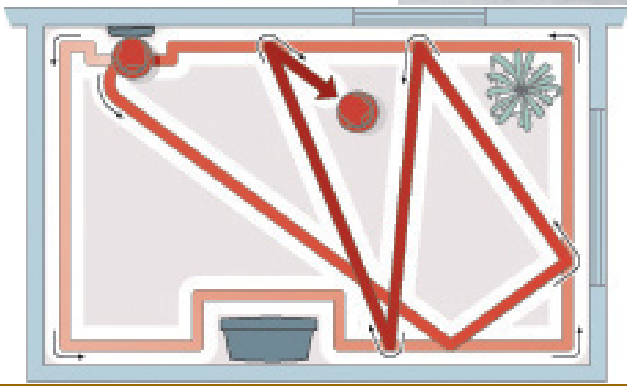


- Extensively studied as model for nuclear reactors, electrical transformers and tombstones



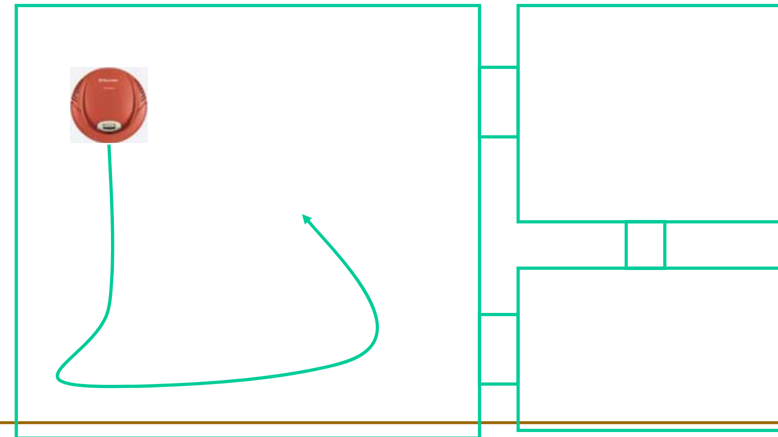
Vacuum Cleaning

- Find an efficient strategy for autonomous cleaning of an apartment

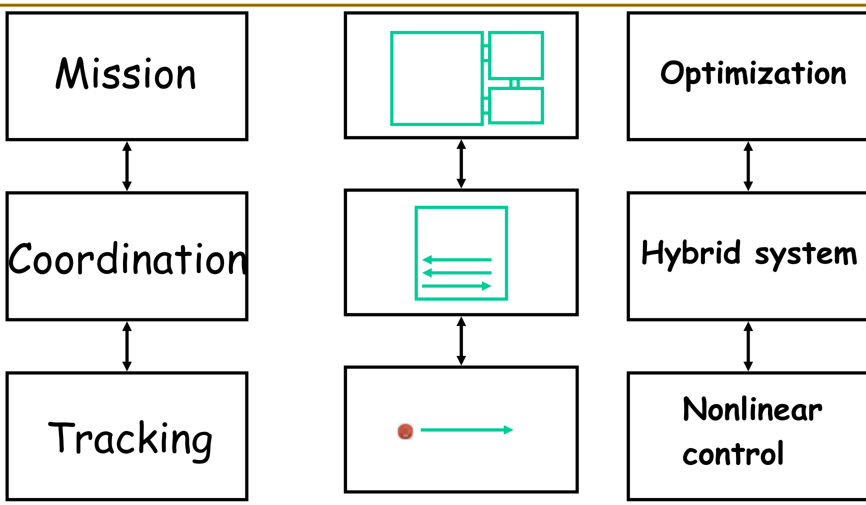


Efficient Area Coverage

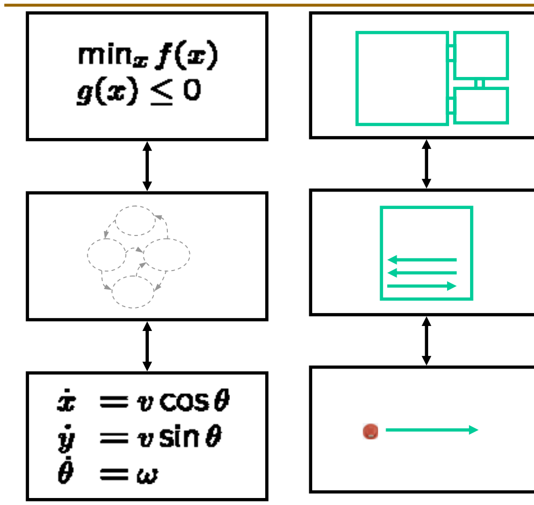
Constrained by nonlinear and uncertain dynamics, sensor noise, actuator limitations, unknown obstacles in environment etc.



Hierarchical Problem Structure

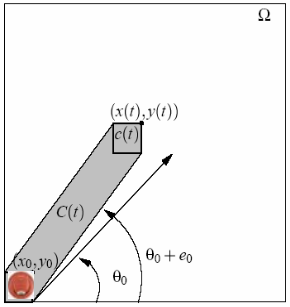


Hierarchical Control



- Natural to organize large systems into hierarchies
- Manageable layers
- Widely adopted approach in engineering: communication, robotics, transportation, etc.
- Cross-layer interaction results in hybrid systems

Area coverage with uncertain heading



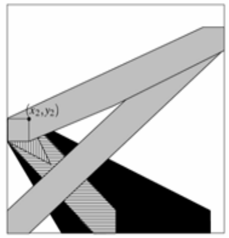
Robot motion governed by

$$\dot{x} = \cos(\theta + e)$$
$$\dot{y} = \sin(\theta + e)$$

where θ is heading, controlled when $c(t) \cap \partial\Omega \neq \emptyset$

$|e| < \varepsilon$ represents uncertainty in control

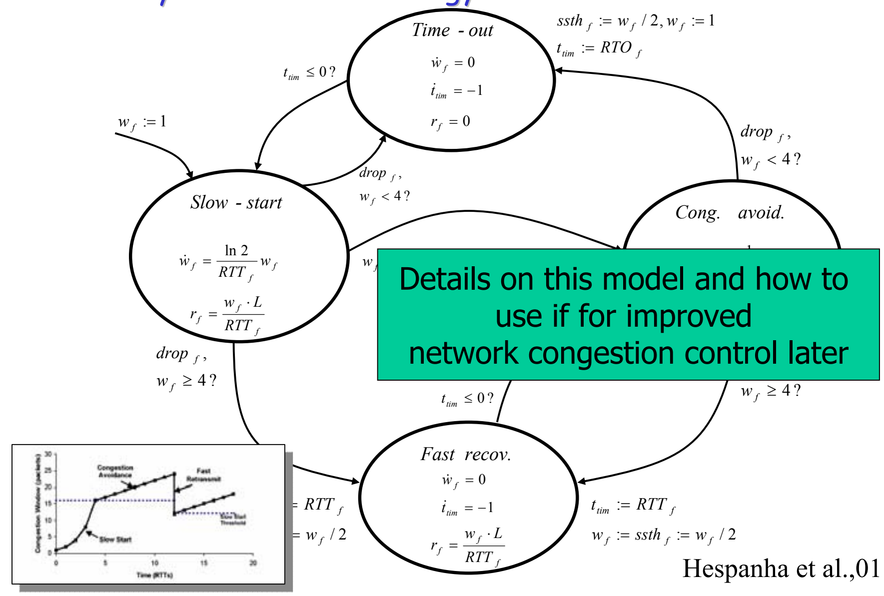
Problem: Given $\varepsilon > 0$, minimize number of turns N to cover Ω



Mazo & J, 04

Network congestion control

TCP is a hybrid control strategy

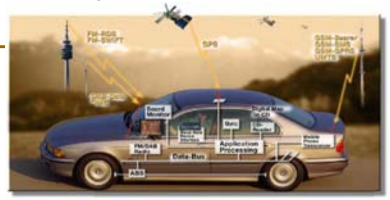


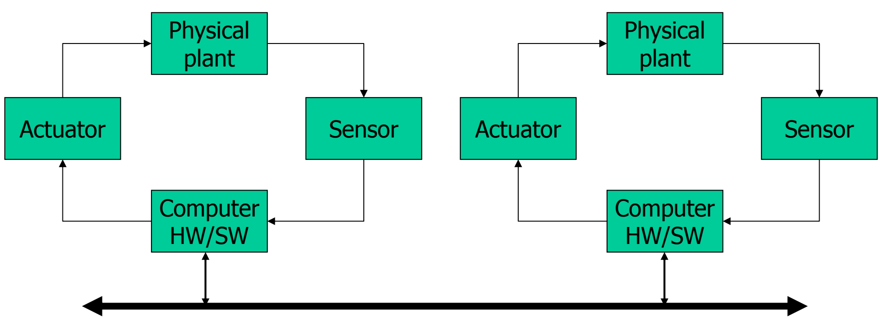
Details on this model and how to use if for improved network congestion control later

Hespanha et al.,01

Networked embedded systems

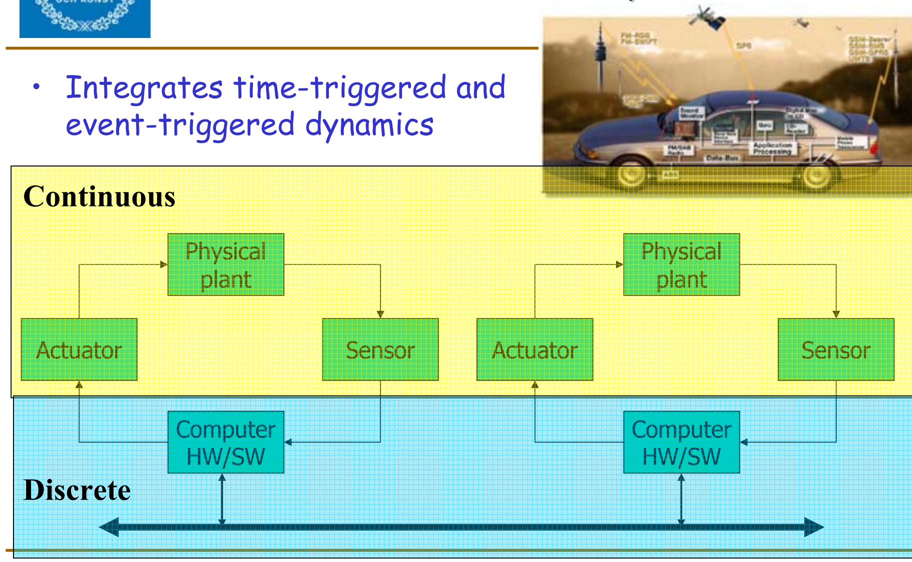
- *Most networked embedded systems are control systems*





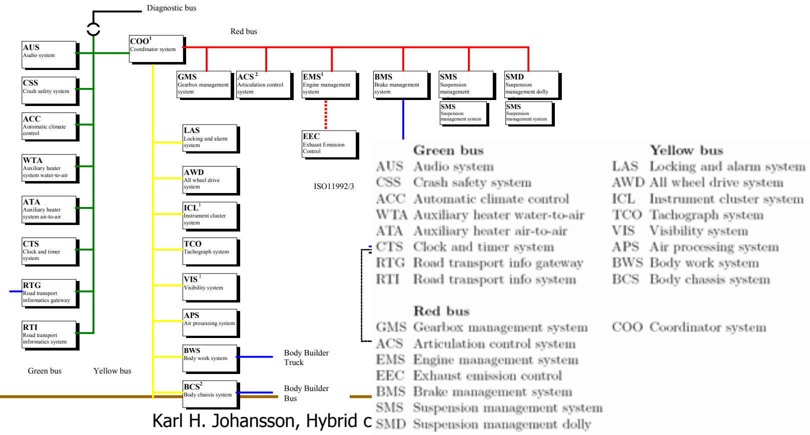
Networked embedded systems

- *Integrates time-triggered and event-triggered dynamics*



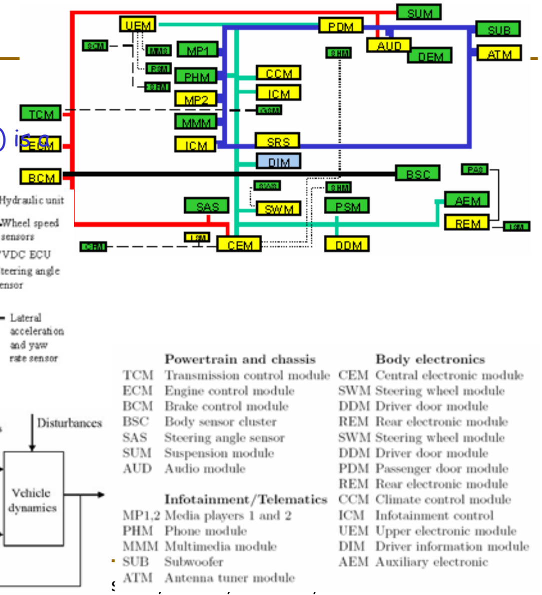
Control architecture of a Scania truck

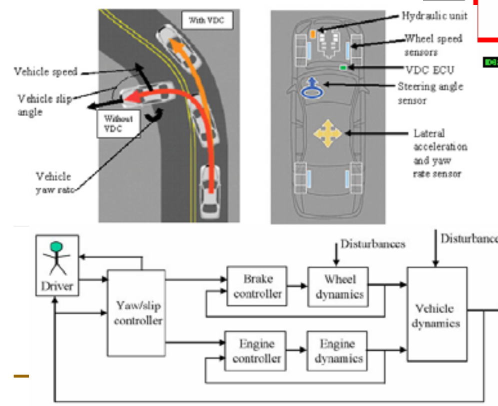
- Control units connected through 3 controller area networks (CANs) coloured by criticality
- CAN is a standard introduced by Bosch 1986



Control architecture of a Volvo XC90

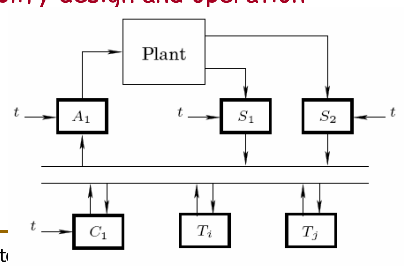
- 3 CAN networks connect up to 40 control units
- Vehicle dynamics control (VDC) is a control system using CAN





What is hybrid in networked embedded control systems?

- Networked control systems are inherently hybrid, because interaction of physical plant and computer control, but also because they have
  - mixture of event- and time-triggered communication protocols
  - asynchronous network nodes (if no global clock)
  - quantized sensor data to limit network traffic
  - symbolic control commands to simplify design and operation



Why hybrid systems?

- Abstractions in design lead to hybrid dynamics
  - Time-scale separation, large scale systems, hierarchical control
- Embedded computer systems are hybrid
  - Real-time software interacting with physical environment
- Many control strategies are hybrid
  - On-off, optimal control, batch control, supervisory control

Hybrid systems integrate control theory and computer science

Control theory
Continuous systems, stability, feedback, robustness

Computer science
Discrete systems, composition, abstraction concurrency

Hybrid systems
Hybrid dynamics, software-controlled systems, networked embedded systems

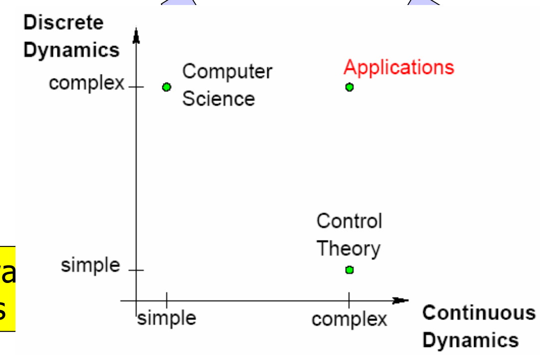
Software from a physics perspective

Physics from a software perspective

Hybrid systems integrate control theory and computer science

Control theory
Continuous systems, stability, feedback, robustness

Computer science
Discrete systems, composition, abstraction concurrency



Softwa
physics

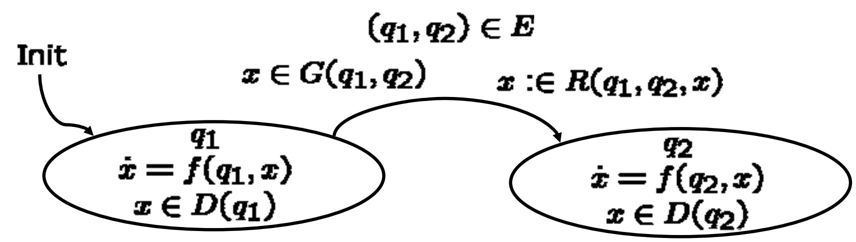
sics from a
re perspective

Outline: Hybrid control systems

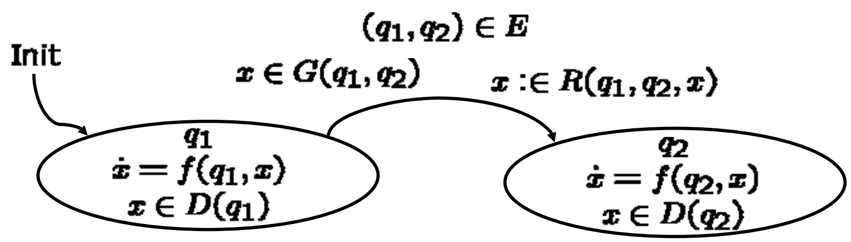
- Introduction
  - What is a hybrid system
  - Motivating examples
- **Models**
  - **Hybrid automata**
- Control
  - Stability and stabilization
- Application
  - Control of network traffic
- Summary
  - Outlook
  - Further reading

Hybrid automaton

- *A hybrid automaton is a formal model of a hybrid system*
- *Captures essential interaction between time-triggered and event-triggered dynamics*



Hybrid automaton $H = (Q, X, \text{Init}, f, D, E, G, R)$



Q discrete variables, X continuous variables

$\mathbf{Q} \times \mathbf{X}$ state space

$\text{Init} \subseteq \mathbf{Q} \times \mathbf{X}$ initial states

$f : \mathbf{Q} \times \mathbf{X} \to T\mathbf{X}$ vector fields

$D : \mathbf{Q} \to P(\mathbf{X})$ domains

$E \subset \mathbf{Q} \times \mathbf{Q}$ edges

$G : E \to P(\mathbf{X})$ guards

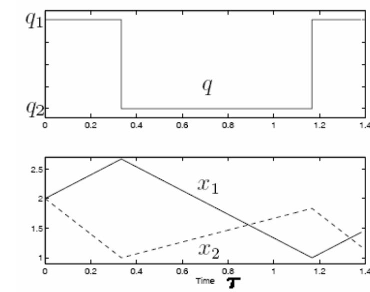
$R : E \times \mathbf{X} \to P(\mathbf{X})$ resets

Solution of a hybrid automaton

A solution $\chi = (\tau, q, x)$ of a hybrid automaton H consists of

Time trajectory τ: Time line on which the solution is defined

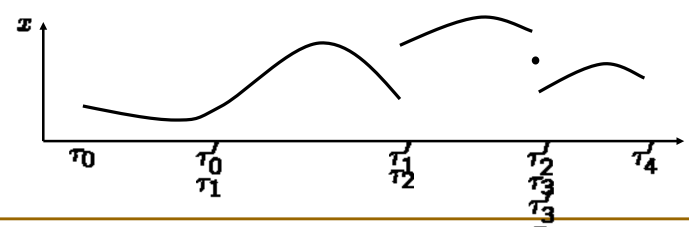
State trajectory (q, x): State evolution of H defined over τ



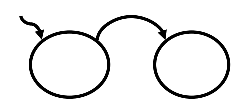
Hybrid time trajectory $\tau = \{I_i\}_{i=0}^N$

A hybrid time trajectory $\tau = \{I_i\}_{i=0}^N$ is a finite or infinite sequence of intervals such that

- $I_i = [\tau_i, \tau_i']$ for all $i < N$
- if $N < \infty$, then either $I_N = [\tau_N, \tau_N']$, or $I_N = [\tau_N, \tau_N')$
- $\tau_i \leq \tau_i' = \tau_{i+1}$ for all i



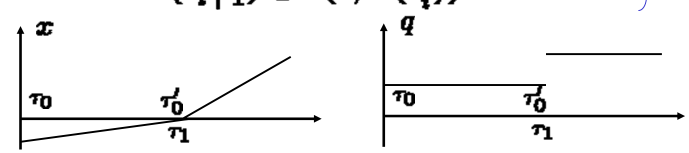
Solution $\chi = (\tau, q, x)$



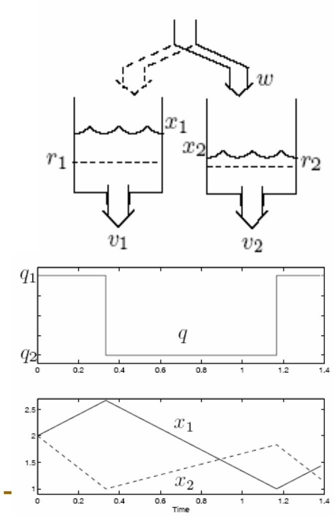
$$\tau = \{I_i\}_{i=0}^N$$
$$q : \langle\tau\rangle = \{1, \ldots, N\} \to \mathbf{Q}$$
$$x = \{x^i : i \in \langle\tau\rangle\}, \; x^i : I_i \to D(q(i))$$

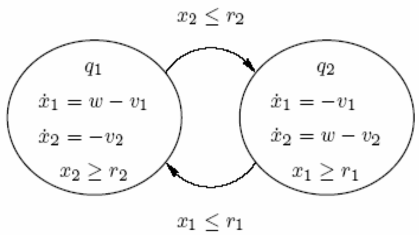
such that

- $(q(0), x^0(0)) \in \text{Init}$ — Initialization
- for all $t \in I_i$, $\dot{x}^i(t) = f(q(i), x^i(t))$ and $x^i(t) \in D(q(i))$ — Time-driven
- for all $i \in \langle\tau\rangle$,
  $e = (q(i), q(i+1)) \in E$
  $x^i(\tau_i') \in G(e)$
  $x^{i+1}(\tau_{i+1}) \in R(e, x^i(\tau_i'))$ — Event-driven



Example: Water tank system





- $Q = \{q\}$ and $\mathrm{Q} = \{q_1, q_2\}$;
- Keep water volumes above r_1 and r_2 by switching the inflow
- $\mathrm{Init} = \mathrm{Q} \times \{x \in \mathbb{R}^2 : x_1 \geq r_1 \wedge x_2 \geq r_2\}$;
- $D(q_1) = \{x \in \mathbb{R}^2 : \ x_2 \geq r_2\}$ and $D(q_2) = \{x \in \mathbb{R}^2 : \ x_1 \geq r_1\}$;
- $E = \{(q_1, q_2), (q_2, q_1)\}$;
- $G(q_1, q_2) = \{x \in \mathbb{R}^2 : x_2 \leq r_2\}$ and $G(q_2, q_1) = \{x \in \mathbb{R}^2 : x_1 \leq r_1\}$;
- $R(q_1, q_2, x) = R(q_2, q_1, x) = \{x\}$.

Example: Timed automaton

A timed automaton models n clocks (e.g., computation time)

A timed automaton is a hybrid automaton $H = (Q, X, \mathrm{Init}, f, D, E, G, R)$ with

- $Q = \{q_1, \ldots, q_m\}$;
- $X = \mathbb{R}^n_+$;
- $\mathrm{Init} \subseteq Q \times X$
- $f(q, x) = (1, \ldots, 1)^T$
- $D(q) = [a_1, b_1] \times \ldots [a_m, b_m]$
- $E \subset Q \times Q$
- $G(e) = [c_1, d_1] \times \ldots [c_m, d_m]$
- $R(e, x) = 0$

Properties of hybrid automata

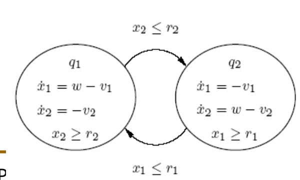
Liveness: There exists at least one infinite χ for all $(q_0, x_0) \in \mathrm{Init}$

Determinism: There exists at most one inf χ for all $(q_0, x_0) \in \mathrm{Init}$

Zenoness: There exists χ with $\tau_\infty = \sum_{i=1}^{\infty} (\tau_i' - \tau_i) < \infty$

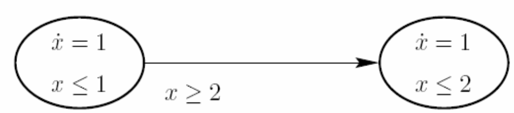
Stability: Solutions converge to equilibria

Reachability: Solutions visit certain sets

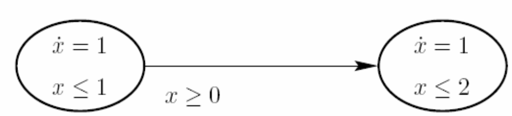


Examples of non-live and non-deterministic hybrid automata

Blocking hybrid automaton (non-live) if $(q_0, x_0) = (q_1, 0) \in \mathrm{Init}$



Non-deterministic hybrid automaton if $(q_0, x_0) = (q_1, 0) \in \mathrm{Init}$



Liveness

Fact

A hybrid automaton is live if for all reachable states for which continuous evolution is impossible, a discrete transition is possible, i.e.,

$$\forall (q,x) \in \mathrm{Reach}(H) \text{ with } x \in \mathrm{Out}(q):$$
$$\exists (q,q') \in E \text{ such that}$$
$$x \in G(q,q') \text{ and } R(q,q',x) \neq \emptyset$$

$$\mathrm{Out}(q) := \{x : \forall \epsilon > 0,\ \exists t \in [0,\epsilon),\ \phi(t,q,x) \notin D(q)\}$$

Determinism

Fact

A hybrid automaton is deterministic if (and only if) there is

- no choice between continuous and discrete evolution, and
- no discrete transition that can lead to multiple states

i.e., $\forall (q,x) \in \mathrm{Reach}(H)$:

- $x \in \bigcup_{(q,q')\in E} G(q,q')$ implies $x \in \mathrm{Out}(q)$
- $(q,q') \in E$ and $(q,q'') \in E$ with $q' \neq q''$ imply $x \notin G(q,q') \cap G(q,q'')$
- $(q,q') \in E$ and $x \in G(q,q')$ imply $|R(q,q',x)| \leq 1$

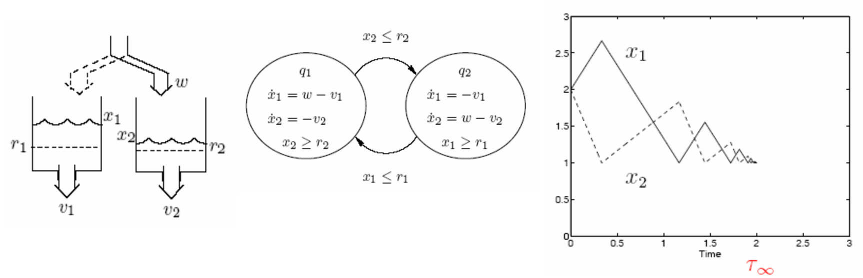
Karl H. Johansson, Hybrid control systems, MOVEP, Bordeaux, 2006

Zeno solution of hybrid automaton

A solution $\chi = (\tau, q, x)$ is Zeno if $\tau_\infty = \sum_{i=1}^{\infty} (\tau_i' - \tau_i) < \infty$

Example—Water tank system: If $\max\{v_1, v_2\} < w < v_1 + v_2$ then

$\tau_\infty = (x_1(0) + x_2(0) - 2r)/(v_1 + v_2 - w) < \infty$



Karl H. Johansson, Hybrid control systems, MOVEP, Bordeaux, 2006

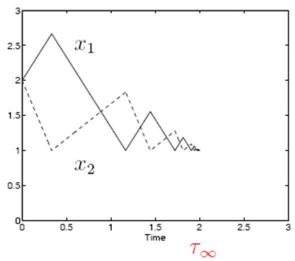
Zeno of Elea (490–430 B.C.)

- Born in southern Italy
- Met Socrates in Athens 449 B.C.
- Went back to Elea and into politics
- Tortured to death
- Paradoxes "proved" that motion and time are illusions
- Led to mathematical problems not solved until 19th century

Karl H. Johansson, Hybrid control systems, MOVEP, Bordeaux, 2006

Zeno hybrid systems

- A solution is Zeno if it exhibits infinitely many jumps in finite time
- A truly hybrid phenomenon: requires at least continuous time and discrete states

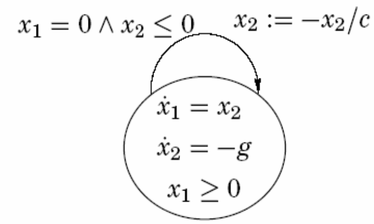


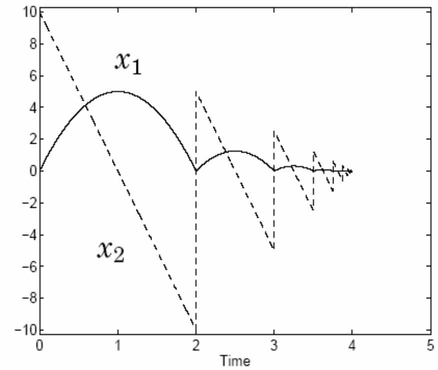
τ_∞

Execution is not defined for $t \geq \tau_\infty$

- Consequence of over-simplified or ill-posed model
- Potential problems for computer simulation, algorithmic verification, stability analysis, etc.
- Few methodologies to detect Zeno

Bouncing ball is Zeno



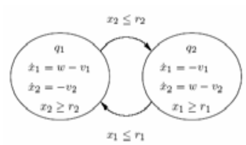


τ_∞

If $c > 1$ then

$$\tau_\infty = \sum_{i=1}^{\infty} (\tau_i' - \tau_i) = \frac{x_2(0)}{g} + \frac{(c+1)\sqrt{x_2(0)^2 + 2g x_1(0)}}{g(c-1)} < \infty$$

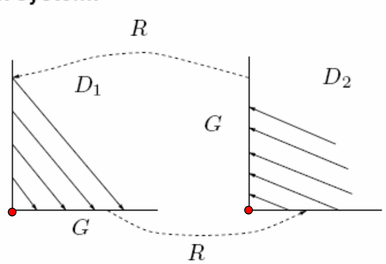
Zeno states



Fact

Zeno states (convergence points of Zeno solution) lie on the intersection of guards

Example—Water tank system:



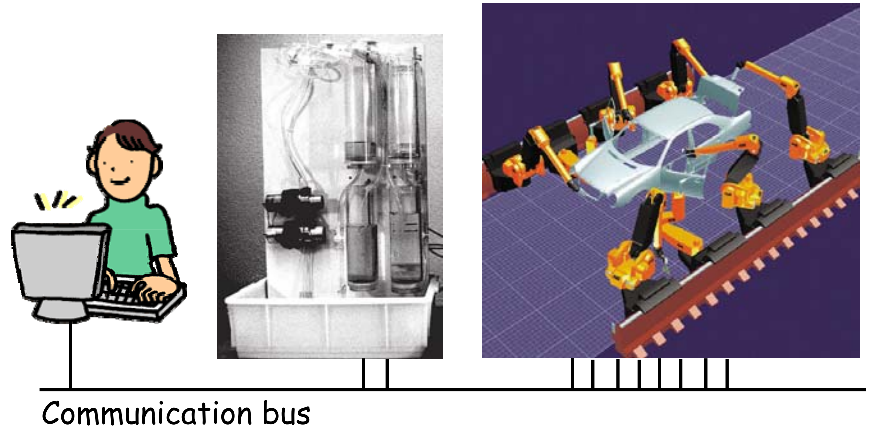
Outline

- Introduction
  - What is a hybrid system
  - Motivating examples
- Models
  - Hybrid automata
- **Control**
  - **Stability and stabilization**
- Application
  - Control of network traffic
- Summary
  - Outlook
  - Further reading

Control of hybrid systems

- Analysis and design problems for traditional continuous control systems can be reformulated for hybrid systems:
  - Stability
  - Optimality
  - Robustness
  - Etc
- Here we focus on **stability**

A hybrid stability problem for a networked control system



Stabilization of networked systems

- Consider joint state feedback stabilization of a set plants, when only one plant can utilize the bus at a time:

$u_i = K_i x_i$	$\dot{x}_1 = A_1 x_1 + B_1 u_1$	$\dot{x}_2 = A_2 x_2 + B_2 u_2$	$\dot{x}_3 = A_3 x_3 + B_3 u_3$

Communication bus

- Determine a control and communication policy that stabilizes all systems?

Switched systems

- For simplicity, we limit the discussion to switched systems, which is a subclass of hybrid automata

A switched system is defined as

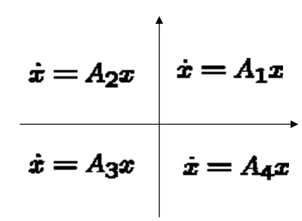
$$\dot{x} = f_q(x), \qquad x \in \Omega_q$$

Ω_q, $q = 1, \dots, m$ denotes a partition of $X = \mathbb{R}^n$

Example

$x \in \mathbb{R}^2$, Ω_q quadrant q, $q = 1, \dots, 4$, and

$$\dot{x} = A_q(x)$$
$$x \in \Omega_q$$



Switched system as hybrid automaton

$$\dot{x} = f_q(x), \qquad x \in \Omega_q$$

corresponds to the hybrid automaton

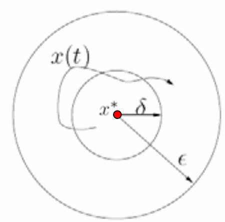
- $Q = \{1, \ldots, m\}$, $X = \mathbb{R}^n$, $\mathrm{Init} \subset \{q\} \times \Omega_q$
- $f(q, x) = f_q(x)$
- $D(q) = \Omega_q$
- $(q, q') \in E$ if $D(q)$ to $D(q')$ are "neighbors" (i.e., $\overline{D(q)} \cap \overline{D(q')} \neq \emptyset$) and there are solutions that go from $D(q)$ to $D(q')$
- $G(q, q') = \overline{D(q)} \cap \overline{D(q')}$
- $R(q, q', x) = x$

Stability for switched systems

A solution x^* of a switched system is stable if for all $\epsilon > 0$, there exists $\delta = \delta(\epsilon) > 0$ such that for all solutions x

$$\|x(0) - x^*(0)\| < \delta \quad \Rightarrow \quad \|x(t) - x^*(t)\| < \epsilon, \quad \forall t > 0$$

- The "usual" stability definition, cf., continuous systems
- How extend to hybrid automata?



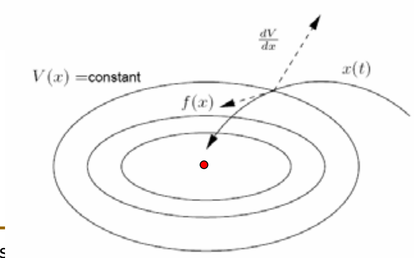
Lyapunov's second method

Let $x^* = 0$ be an equilibrium point of $\dot{x} = f(x)$. If there exists a function $V : \mathbb{R}^n \to \mathbb{R}$ such that

$$V(0) = 0$$
$$V(x) > 0, \quad \forall x \in \mathbb{R}^n \setminus \{0\}$$
$$\dot{V}(x) \leq 0, \quad \forall x \in \mathbb{R}^n,$$

then x^* is stable

***V* is a Lyapunov function**



Lyapunov function for linear systems

$\mathrm{Real}\,\lambda_i(A) < 0$ for all i if and only if for every positive definite $Q = Q^T$ there exists a positive definite $P = P^T$ such that

$$PA + A^T P = -Q$$

Example

$$\dot{x} = A_1 x = \begin{pmatrix} -1 & 10 \\ -100 & -1 \end{pmatrix} x$$

Then,

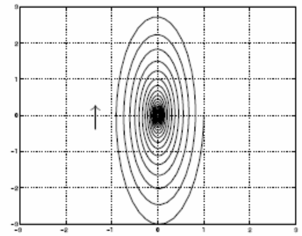
$$P = [\texttt{lyap} \text{ in Matlab}] = \begin{pmatrix} 0.2752 & -0.0225 \\ -0.0225 & 2.7478 \end{pmatrix}$$

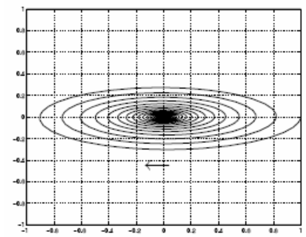
solves the Lyapunov equation $A_1 P + P A_1^T = -I$. Then, $V = x^T P x$ fulfills the three conditions in the Lyapunov theorem (check!). Hence, $x^* = 0$ is stable.

Note that $\lambda(A_1) = -1 \pm i10\sqrt{10}$

Phase portraits

$$\dot{x} = A_1 x = \begin{pmatrix} -1 & 10 \\ -100 & -1 \end{pmatrix} x \qquad \dot{x} = A_2 x = \begin{pmatrix} -1 & 100 \\ -10 & -1 \end{pmatrix} x$$



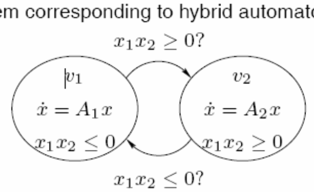


- Both systems ... us slide
- Lyapunov func... solutions:

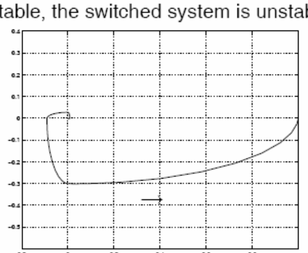
Is a hybrid system that switch between system 1 and system 2 stable?

Stable + Stable = Unstable

Consider switched system corresponding to hybrid automaton:

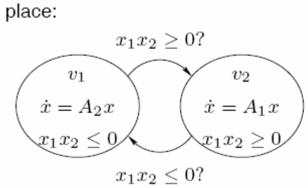


Even if A_1 and A_2 are stable, the switched system is unstable:

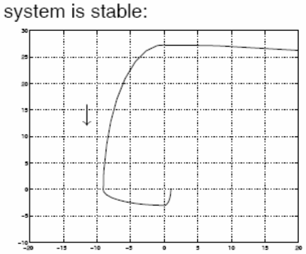


Stable + Stable = Stable

Let A_1 and A_2 change place:



Then, also the switched system is stable:



Multiple Lyapunov functions

Suppose $x^* = 0$ is an equilibrium of each mode $q = 1, \dots, m$ of

$$\dot{x} = f_q(x), \qquad x \in \Omega_q$$

If there exist functions $V_1, \dots, V_m$ such that

$$V_q(0) = 0, \quad V_q(x) > 0, \quad \forall x \in \mathbb{R}^n \setminus \{0\}$$

$$\dot{V}_q(x(t)) \leq 0, \quad \text{whenever } x(t) \in \Omega_q$$

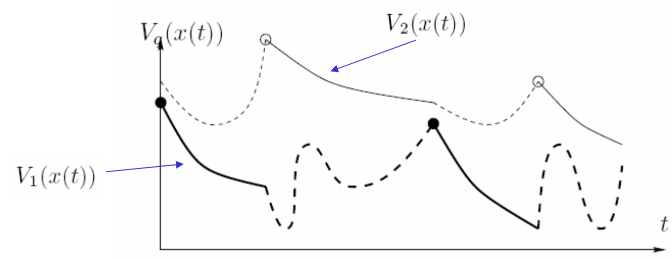
and the sequences $\{V_q(x(\tau_{i_q}))\}$, $q = 1, \dots, m$ are non-increasing, where τ_{i_q} are the time instances when mode q becomes active, then x^* is stable.

- Which of the conditions was violated in previous stable+stable=unstable example?

Branicky

Example with two discrete modes

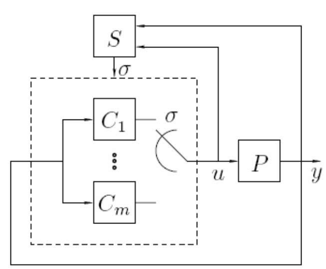
$$\dot{x} = f_q(x), \qquad x \in \Omega_q, \qquad q = 1, 2$$



- Active parts are solid and inactive parts dashed

Supervisory control

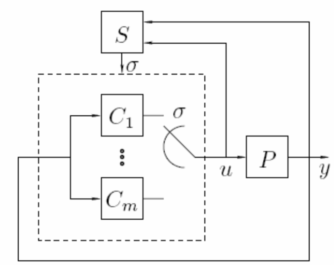
- How choose switching $\sigma = \sigma(t)$ such that $\dot{x} = f_\sigma(x)$ has desired property?
- Let a **supervisor** decide on which controller should be active through a switching signal $\sigma : [0, \infty) \to \{1, \dots, m\}$



- Resulting closed-loop system is a hybrid system

Common Lyapunov function

- If plant and controllers are linear, then closed-loop system is a switched linear system



Consider the system

$$\dot{x} = A_\sigma x$$

where $\sigma : [0, \infty) \to \{1, \dots, m\}$ is an arbitrary switching sequence. If there exists $P > 0$, such that

$$PA_q + A_q^T P = -I, \qquad q = 1, \dots, m$$

then the origin is stable

$V(x) = x^T P x$ is a common Lyapunov function for all systems $\dot{x} = A_q x$

Commuting system matrices

Consider the system $\dot{x} = A_\sigma x$, where $\sigma : [0,\infty) \to \{1,\dots,m\}$ is an arbitrary switching sequence. If all A_q are stable and

$$A_k A_\ell = A_\ell A_k, \qquad k,\ell \in \{1,\dots,m\}$$

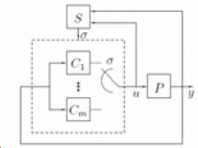
then the origin is stable.

Proof for $m=2$: If $A_1A_2 = A_2A_1$ then $\exp A_1 \exp A_2 = \exp A_2 \exp A_1$ (why?). Then, for time trajectory τ and $t \in [\tau_i, \tau_i']$,

$$\begin{aligned} x(t) &= \exp[A_1(t-\tau_i)]\exp[A_2(\tau_{i-1}'-\tau_{i-1})]\cdots\exp[A_1(\tau_0'-\tau_0)]x_0 \\ &= \exp\left[A_1[(t-\tau_i)+\cdots+(\tau_0'-\tau_0)]\right] \\ &\quad \times \exp\left[A_2[(\tau_{i-1}'-\tau_{i-1})+\cdots+(\tau_1'-\tau_1)]\right]x_0 \end{aligned}$$

Stability follows from that A_1 and A_2 are stable.

How choose stabilizing switching sequence



Suppose there exist $\mu_q \geq 0$, $q \in Q$ and $\sum_{q=1}^m \mu_k = 1$, such that $A = \sum_{q=1}^m \mu_k A_k$ is stable. Then, a stabilizing switching sequence $\sigma : [0,\infty) \to Q := \{1,\dots,m\}$ for

$$\dot{x} = A_\sigma x,$$

is given by

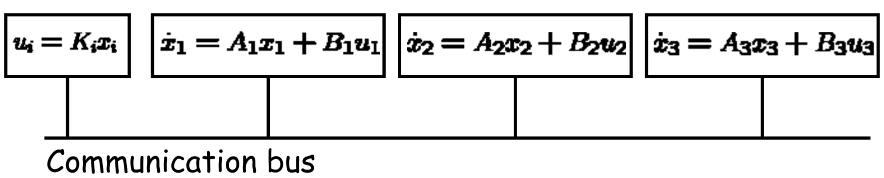
$$\sigma(x) = \arg\min_{q\in Q} x^T(A_q^T P + PA_q)x$$

where $P > 0$ is the solution to $A^TP + PA = -I$.

Proof: Follows from that $\sum_{q=1}^m \mu_q z^T(A_q^TP + PA_q)z < 0$ and $\mu_q \geq 0$, which gives $x^T(A_q^TP+PA_q)x < 0$ for any x.

Stabilization of networked systems revisited

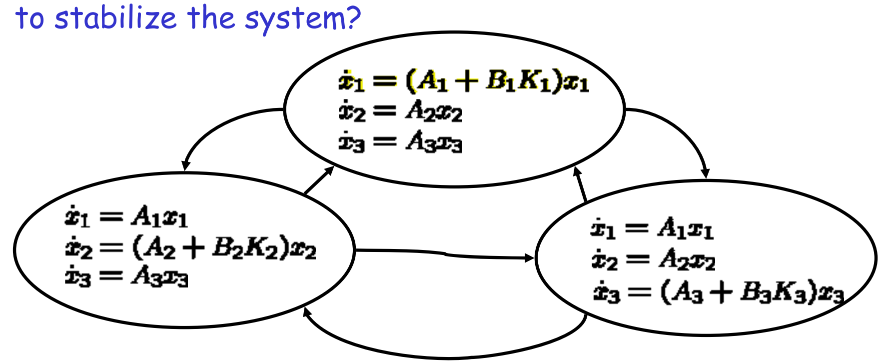
- Consider joint state feedback stabilization of a set plants, when only one plant can utilize the bus at a time:

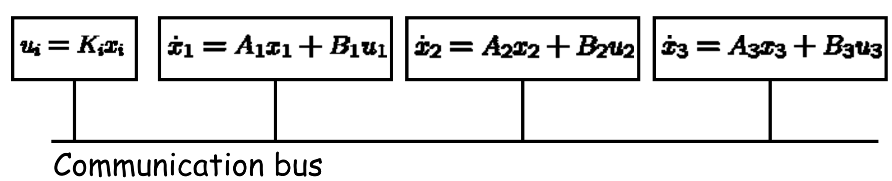


- Determine a control and communication policy that stabilizes all systems?

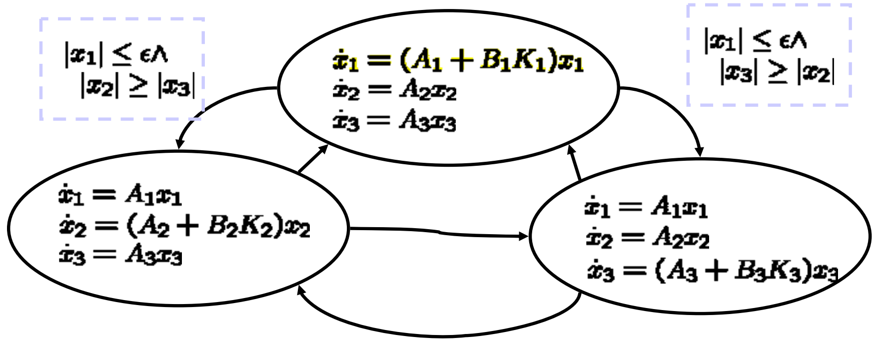
Hybrid system representation

- How choose the guard conditions of the hybrid automaton to stabilize the system?





"Largest state first"-policy



Theorem [Hristu-V. and Kumar]

For scalar unstable systems:

$$|x_i| \to \epsilon \text{ if and only if } -\sum_{i=1}^{3} \frac{A_i}{B_iK_i} < 1.$$

Outline

- Introduction
  - What is a hybrid system
  - Motivating examples
- Models
  - Hybrid automata
- Control
  - Stability and stabilization
- **Application**
  - **Control of network traffic**
- Summary
  - Outlook
  - Further reading

Traffic control in packet-switched communication network

Objective is to

- Give each user suitable service
- Utilize network resources efficiently

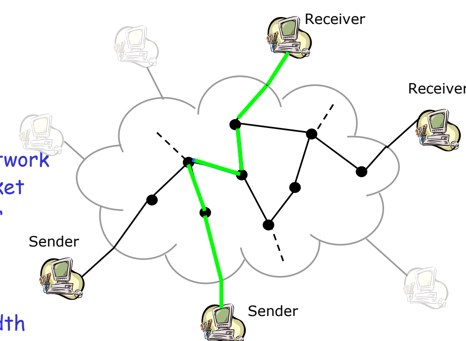
Obtained through two control mechanisms:

Spatial control

- route traffic short way through the network
- receiver address in header of each packet
- shortest-distance matrix in each router
- updated on a slow time scale

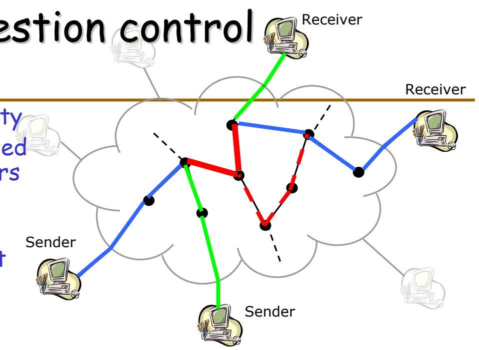
Temporal control

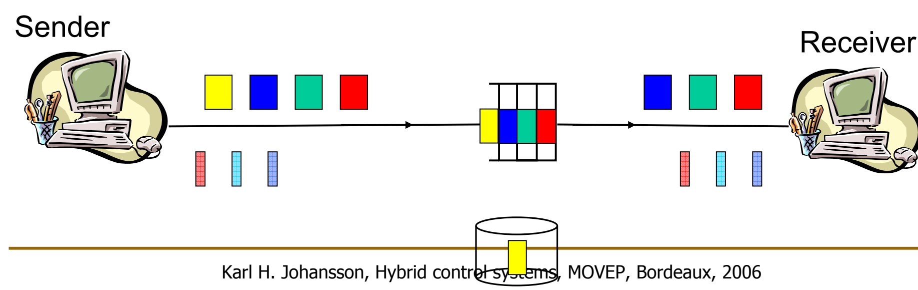
- adjust sending rate to available bandwidth
- base on info available in sender (end-to-end)
- implicit bandwidth estimate through ack's
- updated on a faster time scale



Temporal congestion control

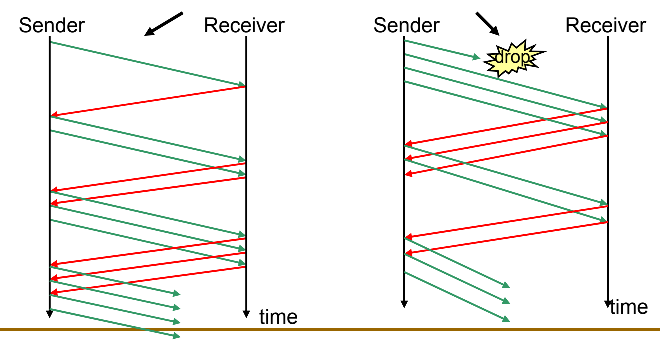
- Each network link has a limited capacity
- Variations in traffic is primarily handled by temporary storage in router buffers
- If a buffer gets full, it simply throws away incoming packets
- Hence, congestion leads to lost packet
- Acknowledgements (ack's) indicate sender, when to take action





Transmission control protocol (TCP)

- TCP implements a congestion controller that regulates the sending rate
- Control variable is the congestion window w, which represents number of outstanding (not-yet-acknowledged) packets
- Control is based on implicit feedback information from ack's
- TCP follows additive increase multiplicative decrease (AIMD) strategy

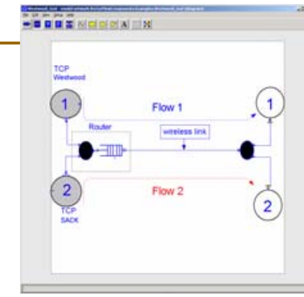


TCP congestion avoidance

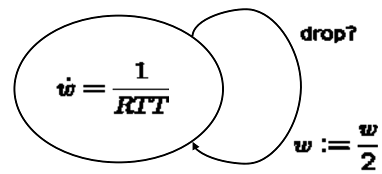
Window w is updated each round-trip time RTT

If no drops occur, then $w := w + 1$

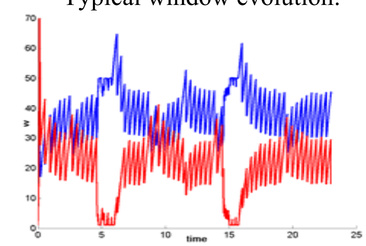
If drop occurs, then $w := w/2$



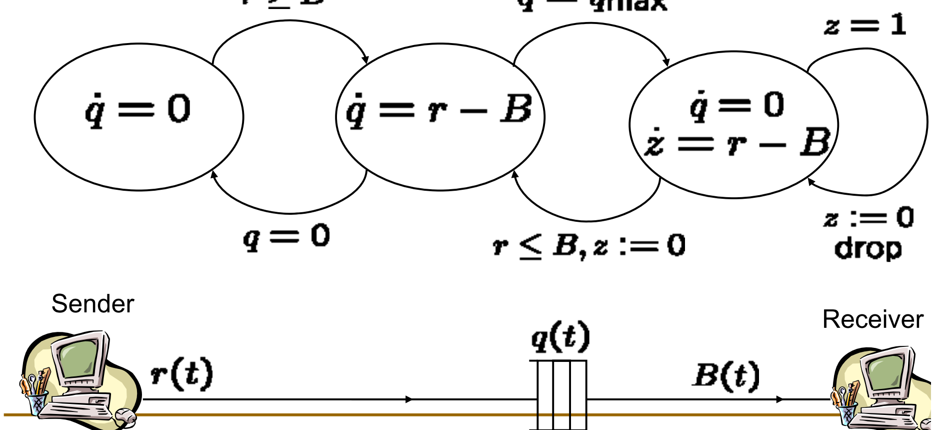
Hybrid system is obtained by interpreting w as a continuous-time real variable:



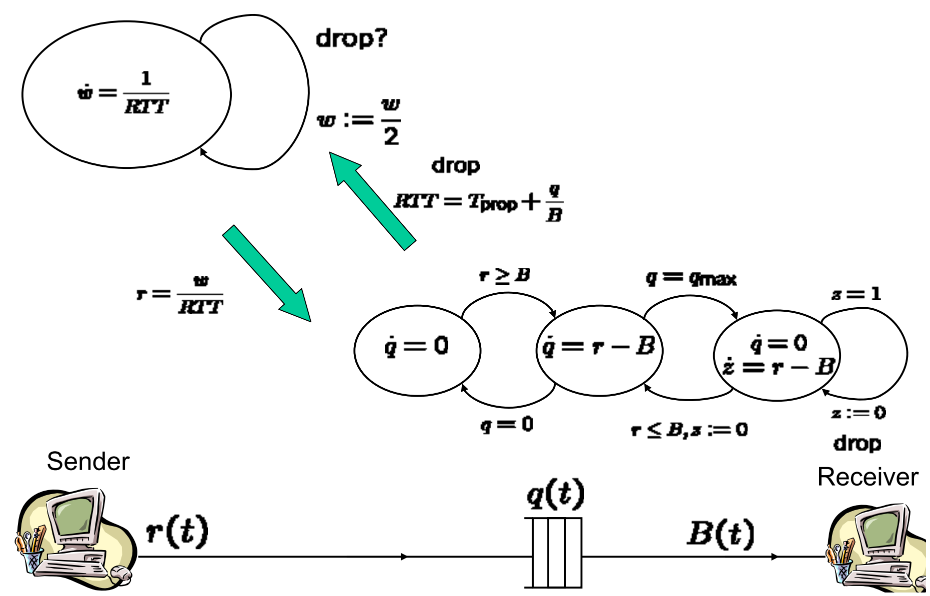
Typical window evolution:



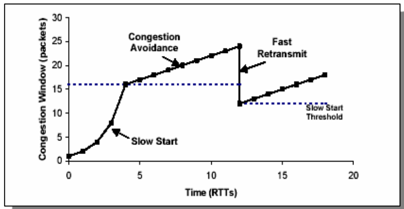
Hybrid dynamics of a queue

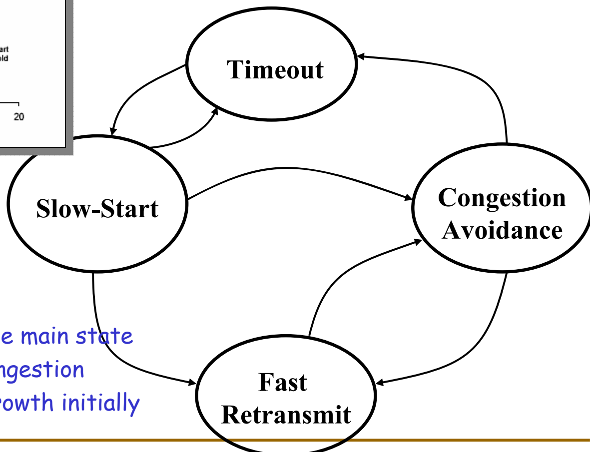


Hybrid model of TCP over single link



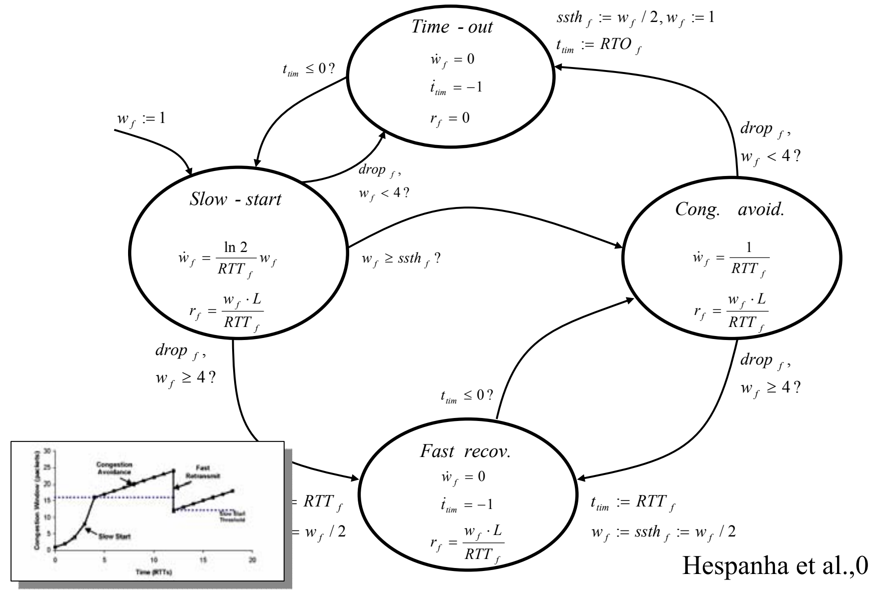
The discrete states of TCP





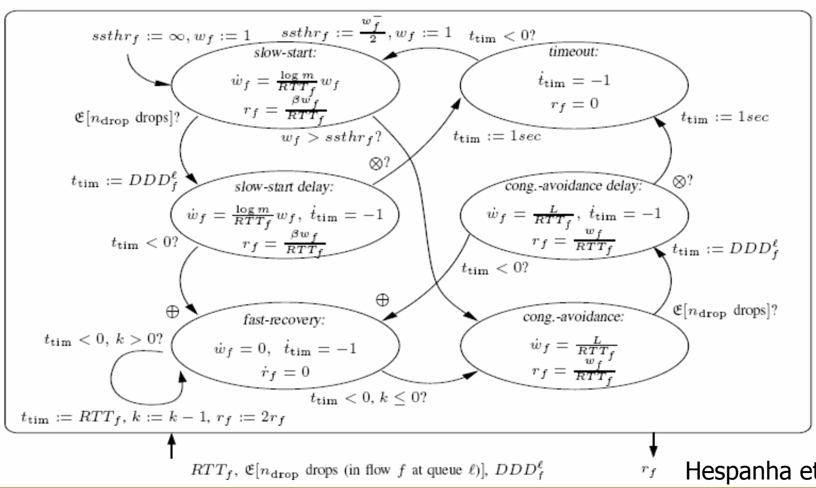
- **Congestion Avoidance** is the main state
- **Timeout** handles severe congestion
- **Slow-Start** gives faster growth initially

TCP is a hybrid control strategy



Hespanha et al.,01

A more accurate hybrid model of TCP



Hespanha et al.

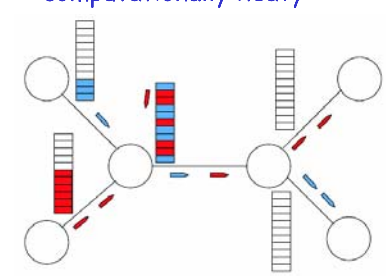
Alternative models of network traffic: Packet and fluid models

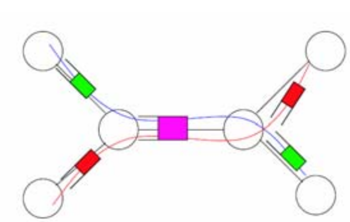
Packet models

- Model each individual packet (event-driven)
- Accurate but computationally heavy

Fluid models

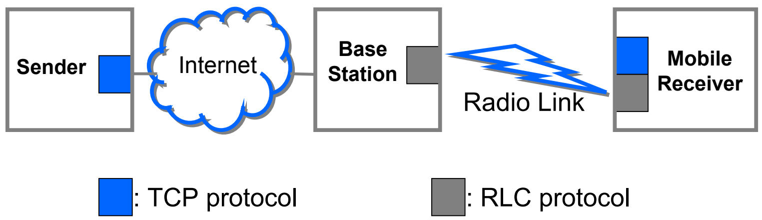
- Averaged fluid quantities (time-driven)
- Capture only steady-state and slow behaviors





- Hybrid model combines features of these traditional network models

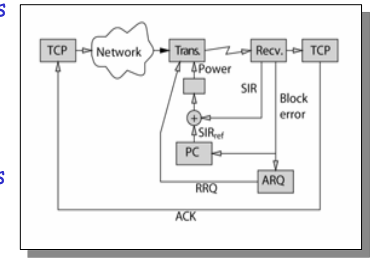
TCP over wireless links

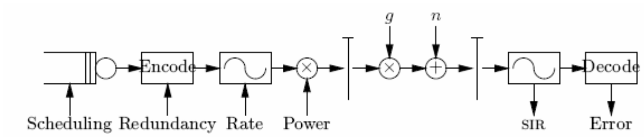


- Integration of Internet and cellular networks hard due to radio link variations
- When used over wireless links, TCP cannot ensure a high link utilization
- Packet drops, bandwidth and delay variations in radio link erroneously indicate network congestion to TCP
- How do radio links affect TCP throughput?
- Can we make the radio link and the cellular system "TCP friendly"?

Cascade control

- Radio link transforms losses into random delays
- Cascaded feedback control loops
  - Inner and outer power controls
  - Link-layer retransmission
  - TCP
- Increased probability of spurious timeout gives reduced TCP throughput
- Adjust link layer properties to optimize TCP throughput

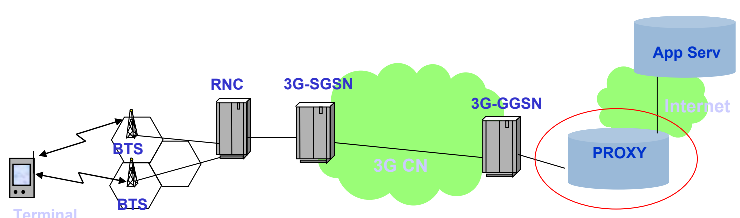




Karl H. Johansson, Hybrid control systems, MOVEP, Bordeaux, 2006

New feedback protocols for wireless Internet

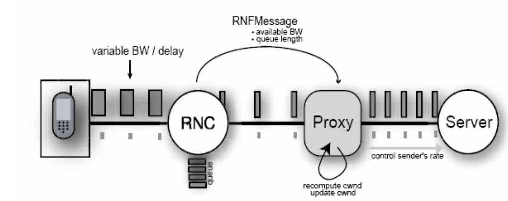
- Improved TCP throughput through new radio network feedback protocol
- Proxy between cellular system and Internet adapt sending rate to radio bandwidth variations obtained from radio network controller (RNC)

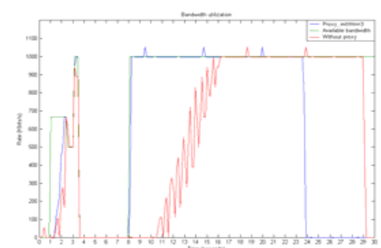


Karl H. Johansson, Hybrid control systems, MOVEP, Bordeaux, 2006

New feedback protocols for wireless Internet

- **Hybrid controller** in proxy regulates sending rate based on
  - Events generated by radio bandwidth changes obtained from RNC
  - Sampled measurements of queue length in RNC
- Improved time-to-serve-user and utilization compared to traditional end-to-end TCP



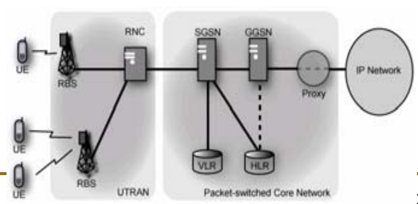


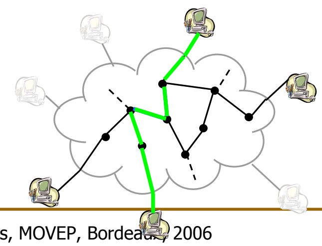
Karl H. Johansson, Hybrid control systems, MOVEP, Bordeaux, 2006

Summary of the application hybrid control of network traffic

- Hybrid model of congestion control in packet-switched networks
- Combine event-driven packet models with time-driven fluid models
- Accurate on time-scale of the round-trip time
- Enables analysis and efficient simulations of congestion control

- Interactions between wireless links and TCP lead to performance loss
- Hybrid controller gives improved user experience and network utilization





Outline

Examples of what was **not covered** in the presentation

- Models
  - Other hybrid models classes, e.g., stochastic hybrid systems
  - How to obtain models: system identification for hybrid systems
  - Computer simulation
- Estimation and observers
  - How to estimate the state of a hybrid system
- Control
  - Optimal control of hybrid systems
  - Non-smooth control
- **Verification**
  - Model checking
  - Reachability analysis
  - Reach set computations
- Implementation

Summary

- Hybrid systems arise naturally in the design of embedded computer system, where **real-time software** is **interacting** with a **physical environment**
- Integrate problem formulations and mathematical tools from **control theory** and **computer science**
- Emerging theories and computational tools for modeling, control, verification, simulation and implementation

- Area with a lot of activities, including major European projects

ARTIST2 HYCON

References

- A. van der Schaft and H. Schumacher. An Introduction to Hybrid Dynamical Systems. Lecture Notes in Control and Information Sciences 251, Springer-Verlag, 2000.
- D. Hristu-Varsakelis and W. S. Levine, Ed.'s, Handbook of Networked and Embedded Control Systems, Birkhäuser, 2005
- D. Liberzon. Switching in Systems and Control. Systems & Control: Foundations and Applications series. Birkhauser, Boston, 2003.

- Alur, Courcoubetis, Henzinger, Ho. Hybrid Automata: an Algorithmic Approach to Specification and Verification of Hybrid Systems. Hybrid Systems I (LNCS 736).
- J. Lygeros, K. H. Johansson, S. Simic, J. Zhang, and S. Sastry. Dynamical properties of hybrid automata, IEEE Transactions on Automatic Control, 48:1, 2-17, 2003.
- K. H. Johansson, M. Egerstedt, J. Lygeros, and S. Sastry, On the regularization of Zeno hybrid automata, System & Control Letters, 38, 141-150, 1999.
- M. S. Branicky, Multiple Lyapunov functions and other analysis tools for switched and hybrid systems, IEEE Transactions on Automatic Control, 43(4), 475-482 1998.
- A. Puri and P. Varaiya. Decidable Hybrid Systems, Computer and Mathematical Modeling,23(11/12):191-202, 1996.

- J. Hespanha, S. Bohacek, K. Obraczka, and J. Lee, "Hybrid Modeling of TCP Congestion Control". In M. D. Di Benedetto, A. Sangiovanni-Vincentelli, Hybrid Systems: Computation and Control, number 2034 in Lect. Notes in Comput. Science, 2001
- N. Möller, I. Cabrera Molero, K. H. Johansson, J. Petersson, R. Skog, and Å. Arvidsson, "Using radio network feedback to improve TCP performance over cellular networks", IEEE CDC-ECC, 2005

- 2E1245 Hybrid and Embedded Control Systems, http://www.s3.kth.se/control/kurser/2E1245

http://www.s3.kth.se/~kallej