CSE 591: Theoretical Aspects of CPS

Control Systems:
Tabuada Ch 8.1

Instructor: Georgios E. Fainekos

School of Computing, Informatics and Decision System Engineering

Arizona State University

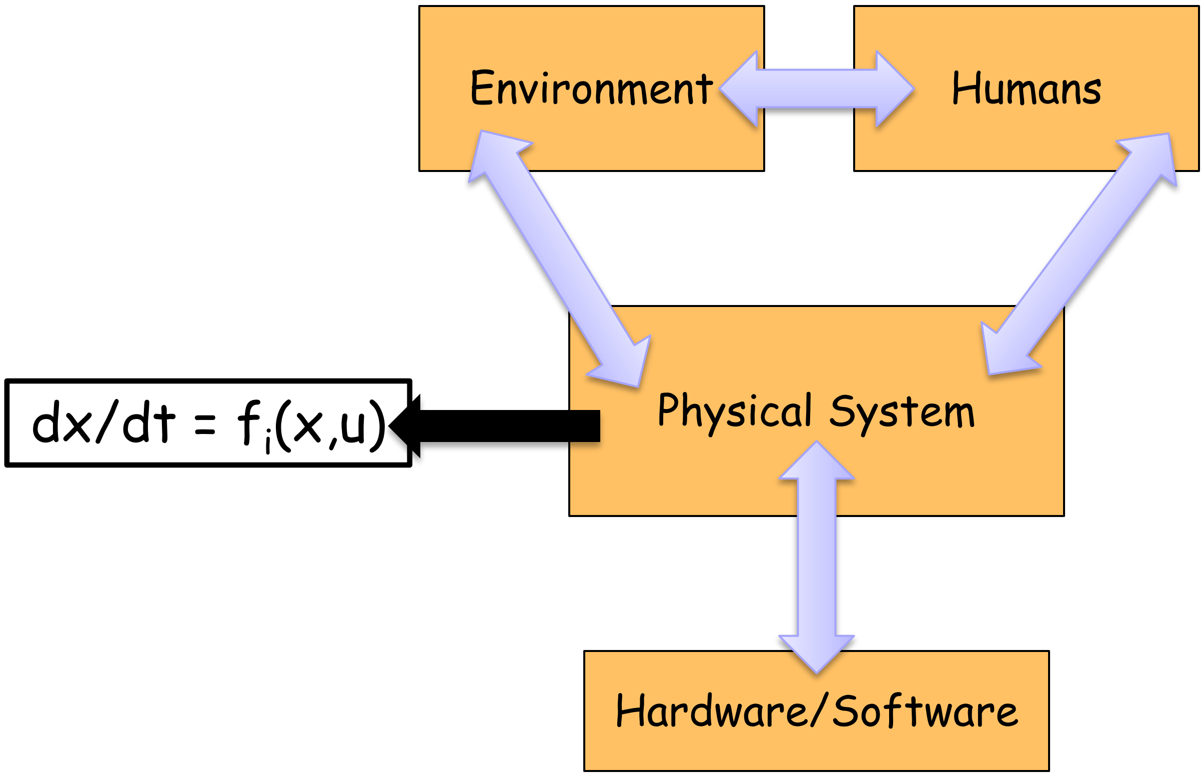
fainekos at asu edu

http://www.public.asu.edu/~gfaineko

Results of Survey

- Textbook
  - The notation is challenging
  - Not enough examples
- More high level lectures
- Homeworks
  - Challenging, long and time consuming
- Practical applications and review of literature
- What is specific to CPS?

Models of systems

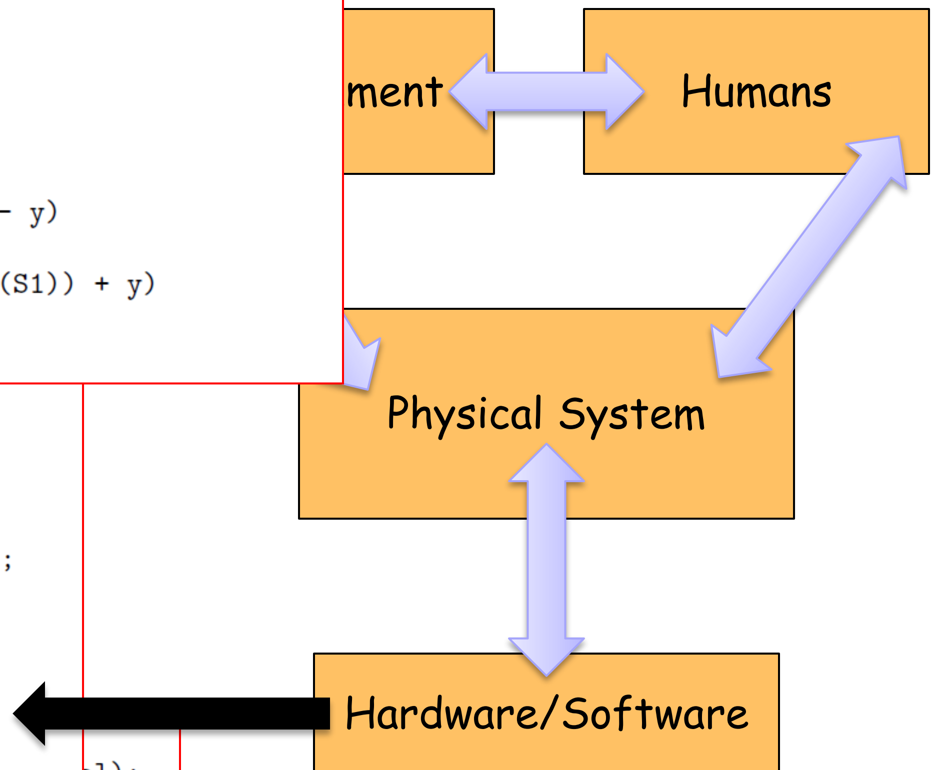


Models of systems

```
node Rate_Lim(E1: real; Lim, cycle_duration: real;)
          returns (S1 : real) ;
var x, y : real;
let
   x = E1 - (0.0 -> pre(S1)) ;
   y = Lim * cycle_duration ;
   S1 = if (x < (-y))
          then ((0.0 -> pre(S1)) - y)
          else (if (x > y)
                then ((0.0 -> pre(S1)) + y)
                else E1) ;
tel.
```

```
nc
            returns (s1 : real) ;
var xmin:real, xmax : real;
let
  (xmax , xmin) = if (Max >= Min)
                     then (Max , Min)
                     else (Min , Max) ;
   s1 = if (xmax <= e1)
          then xmax
          else (if (e1 > xmin)
                 then e1
                 else xmin) ;
tel.
```

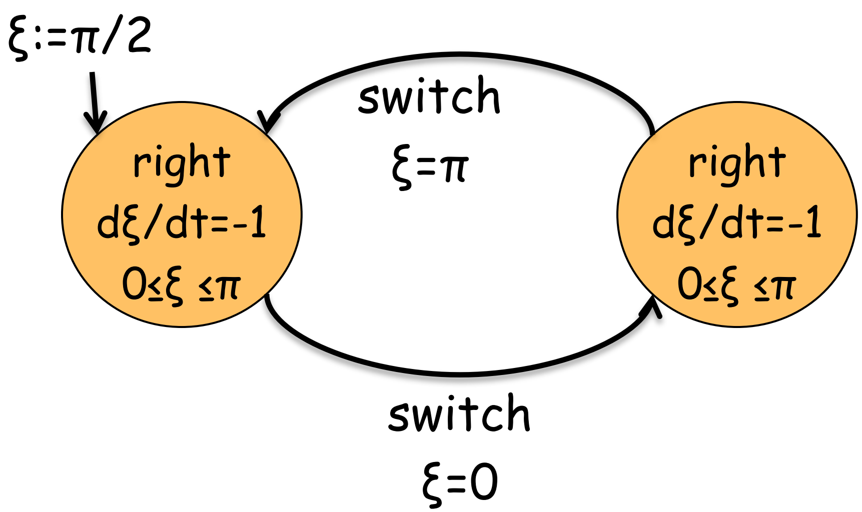
```
                                                    al);
var x, y : real; let
 Current = Val_Lim(x*(-3.2), -8.0, 8.0);
 Order = (y*0.75) +
    ((0.0->pre(y))*0.5) -
        ((0.0 -> pre(0.0 -> pre(y)))*0.25)
   x = Order - (Sensor*130.);
   y = Rate_Lim(Xgouv, 2.0, 0.01) ;
tel.
```

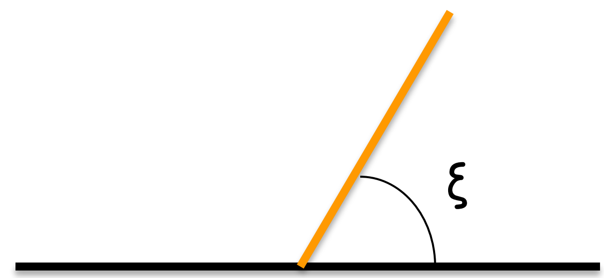


Code snippets from: Robustness analysis of avionics embedded systems by Ait-Ameur et al

Ideal model: Hybrid automaton

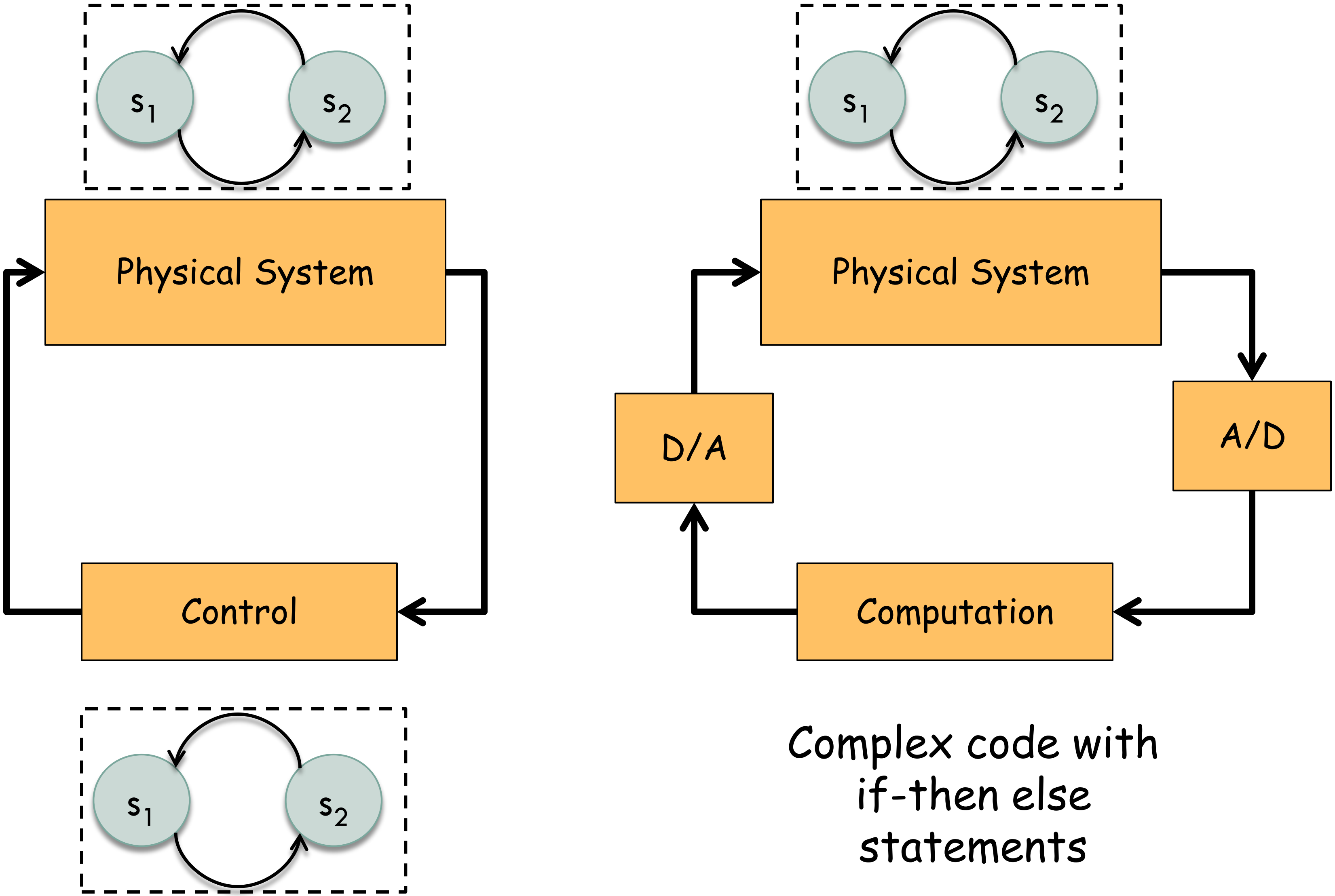
What can you say about the behavior of the system in the ideal case?





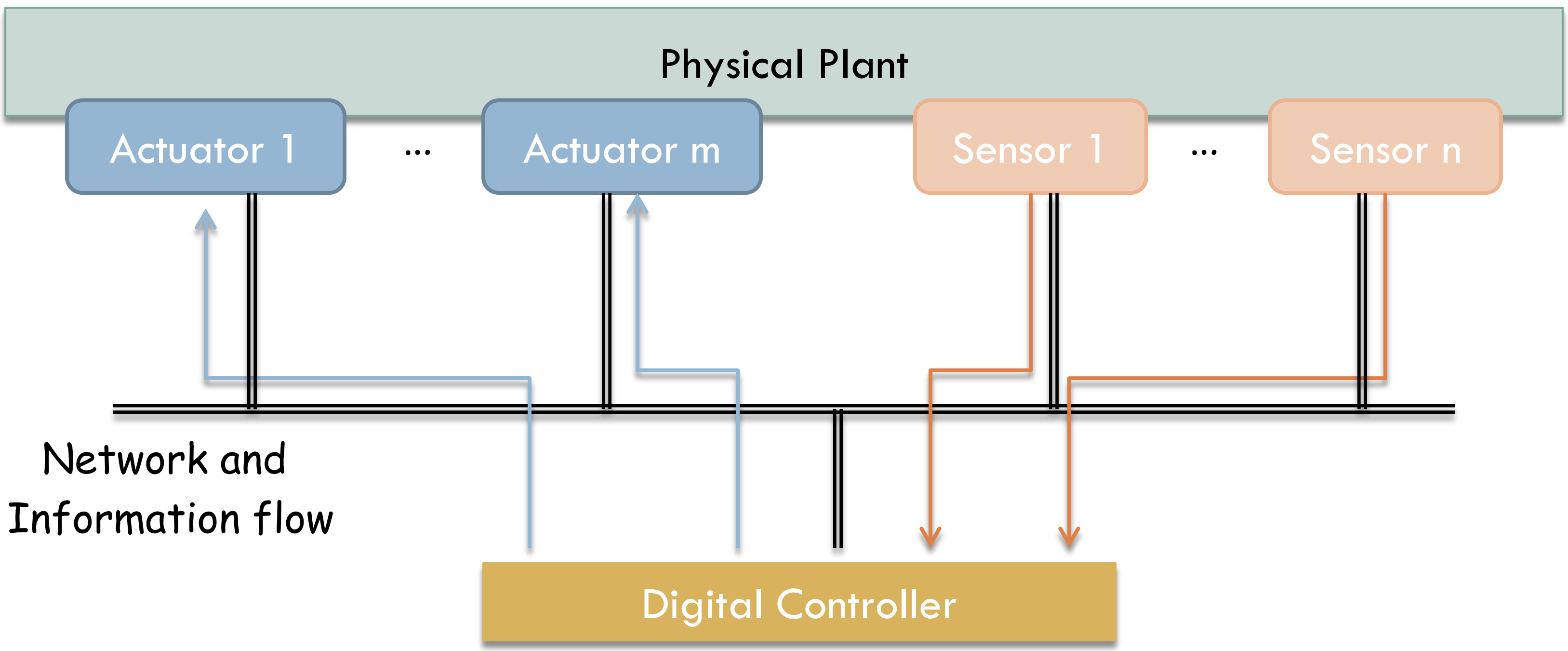
So far we have seen the limitations and the success stories on the verification of ideal models!

What is the distance between implementation and ideal behavior?



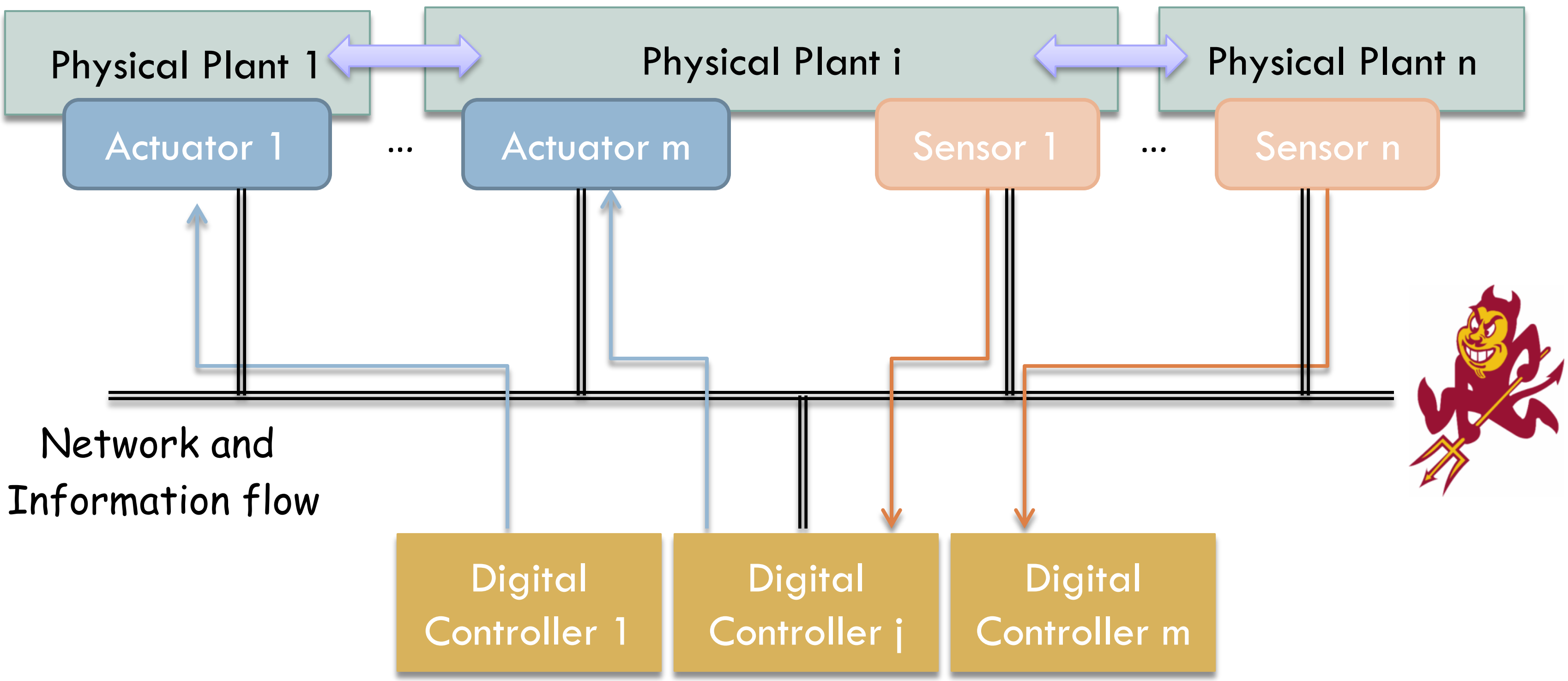
Complex code with if-then else statements

More complex …



More and more complex …

Controllers and plants appear and disappear



Hybrid behavior arises in

Hybrid dynamics

- Hybrid model is a simplification of a larger nonlinear model
- Some phenomena in nature can only be modeled with hybrid models

Quantized control of continuous systems

- Input and observation sets are finite

Logic based switching

- Software is designed to supervise various dynamics/controllers

Partial synchronization of many continuous systems

- Resource allocation for competing multi-agent systems

Hybrid specifications of continuous systems

- Plant is continuous, but specification is discrete or hybrid…

And many more …

Research Issues

Modeling Issues

- Well posedness, robustness, zenoness

Analysis

- Stability issues, qualitative theory, parametric analysis

Verification

- Algorithmic methods that verify system performance

Controller Synthesis

- Algorithmic methods that design hybrid controllers

Simulation

- Mixed signal simulation, event detection, modularity

Code generation

- From hybrid models to embedded code

Complexity

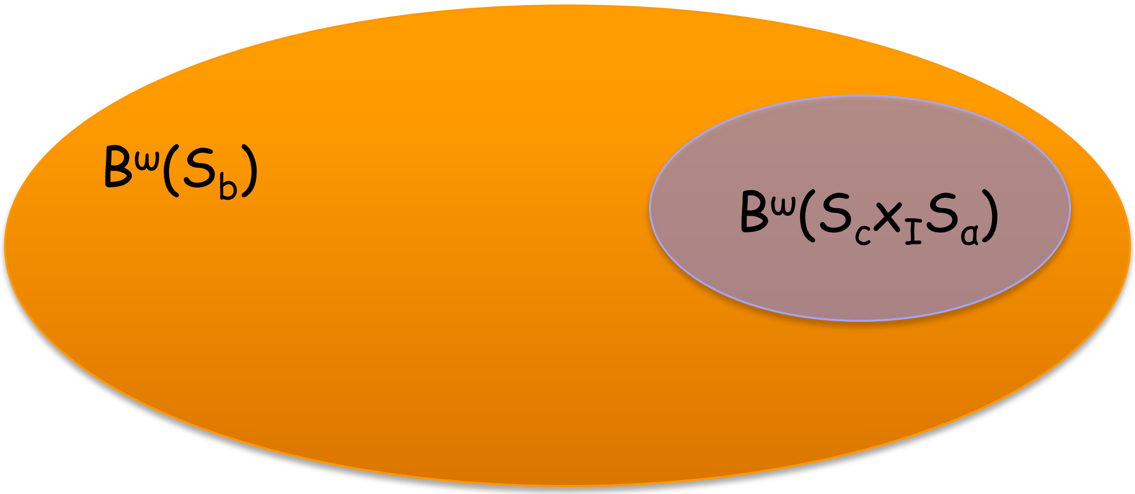
- Compositionality and hierarchies

Back to homework

- HW0
  - State spaces
- HW1
  - Partial orders, monotonicity and continuity
  - Homework problem is not arbitrary: used in language theory
  - Applications of partial-orders and lattices in hybrid systems:
    - Interesting applications on vehicle collision avoidance by Domitilla Del Vecchio
    - See tutorial paper in CDC 07

Control

- Can we find a controller S_c such that $S_c x_I S_a \preceq S_b$



Discrete-Time Control Systems

- $\Sigma = (\mathbb{R}^n, \mathbb{R}^m, f)$
  - $\mathbb{R}^n$ State space
  - $\mathbb{R}^m$ Input space
  - $f : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n$ smooth map
- If Q is an equivalence on $\mathbb{R}^n$, then the system $S_Q(\Sigma)$ associated with Σ and Q is
  - $X = \mathbb{R}^n$
  - $U = \mathbb{R}^m$
  - $x \xrightarrow{u} x'$ if $x' = f(x,u)$
  - $Y = X/Q$
  - $H = \pi_Q$

Discrete-time Controllability

- A discrete time system Σ is controllable if for any two states x, x' there exists a finite internal behavior of $S_Q(\Sigma)$

$$\mathbf{x} \xrightarrow{\mathbf{u}_0} \mathbf{x_1} \xrightarrow{\mathbf{u}_1} \mathbf{x_2} \xrightarrow{\mathbf{u}_2} \ldots \xrightarrow{\mathbf{u}_{k-1}} \mathbf{x}'$$

- Linear system $\Sigma = (\mathbb{R}^n, \mathbb{R}^m, f)$ if $f(x,u) = Ax + Bu$

- Theorem: A discrete-time linear system $\Sigma=(\mathbb{R}^n, \mathbb{R}^m, A, B)$ is controllable iff rank $C = n$ where C is the controllability matrix:
- $C = [B \mid AB \mid A^2B \mid \ldots \mid A^{n-1}B]$

Continuous-Time Control Systems

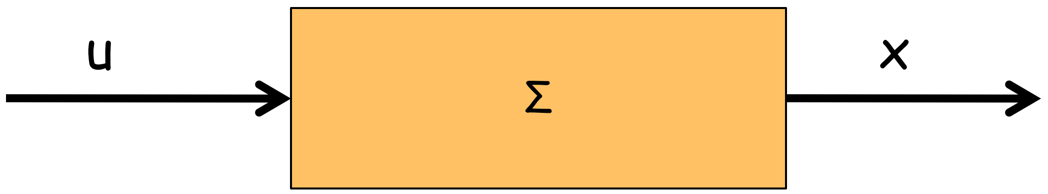
- $\Sigma = (\mathbb{R}^n, U, f)$
  - $\mathbb{R}^n$ is the state space
  - U is a family of piecewise continuous functions that map to $\mathbb{R}^m$
  - $f : \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}^n$ is a smooth map

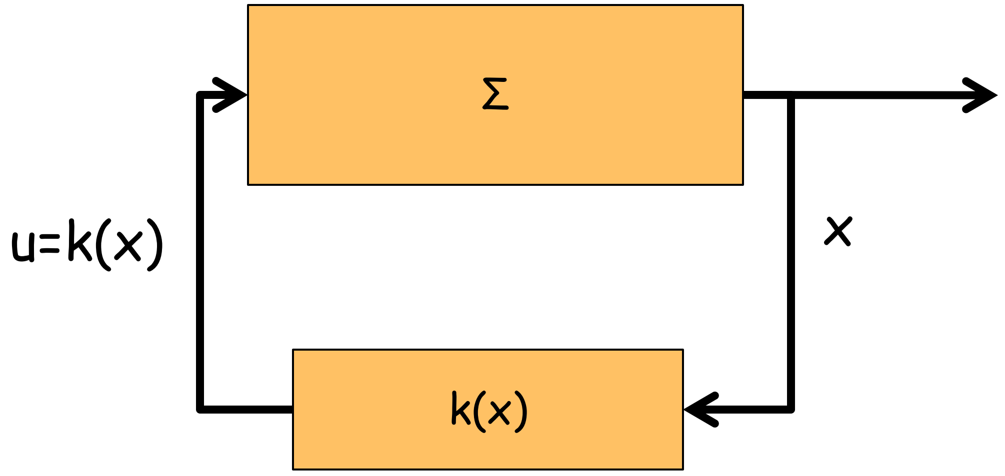
- A trajectory ξ is a solution of Σ if there exists $u \in U$ such that $d\xi/dt = f(\xi, u)$

Continuous-Time Control Systems

- If Q is an equivalence on $\mathbb{R}^n$, then the system $S_Q(\Sigma)$ associated with Σ and Q is
  - $X = \mathbb{R}^n$
  - U
  - $x \xrightarrow{u} x'$ if one of the following holds
    1. $\pi_Q(x) \neq \pi_Q(x')$, $\xi : [0,\tau] \to \mathbb{R}^n$ is a solution satisfying $\xi(0)=x$ and $\xi(\tau)=x'$ and there exists $\varepsilon \in [0,\tau]$ such that
       1. $\pi_Q(\xi(t))=\pi_Q(x)$ for $t \in [0,\varepsilon)$ and $\pi_Q(\xi(t))=\pi_Q(x')$ for $t \in [\varepsilon,\tau]$, or
       2. $\pi_Q(\xi(t))=\pi_Q(x)$ for $t \in [0,\varepsilon]$ and $\pi_Q(\xi(t))=\pi_Q(x')$ for $t \in (\varepsilon,\tau]$
    2. $\pi_Q(x)=\pi_Q(x')$, and $\xi : \mathbb{R}^+ \to \mathbb{R}^n$ is a solution satisfying $\xi(0)=x$, $\xi(\tau)=x'$ and $\pi_Q(\xi(t))=\pi_Q(x)$ for all $t \in \mathbb{R}^+$
  - $Y = X/Q$
  - $H = \pi_Q$

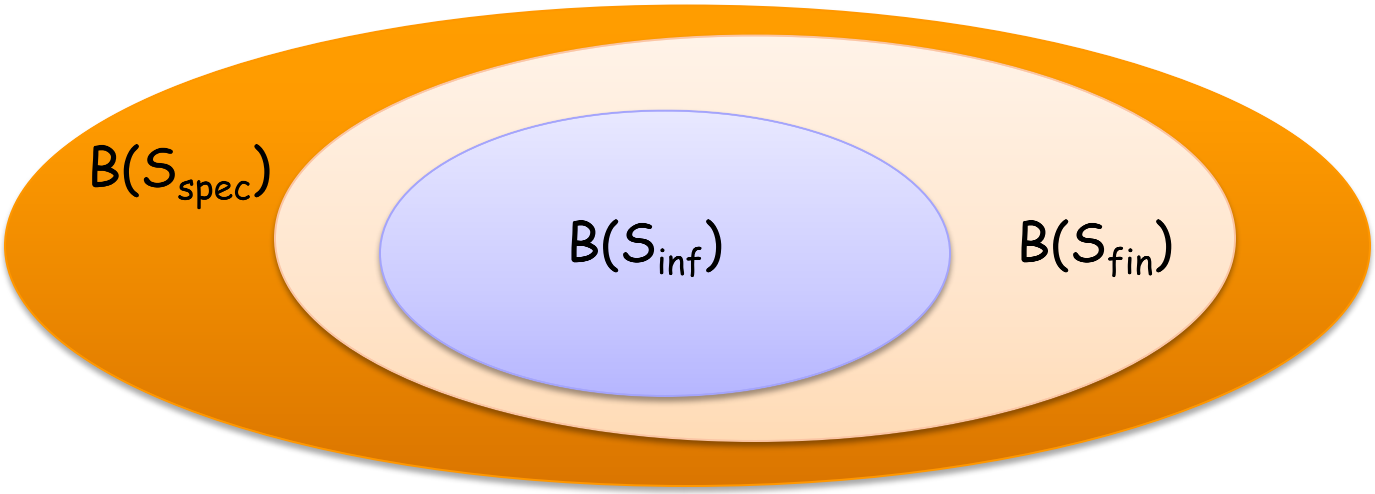
Open-loop VS Closed-loop





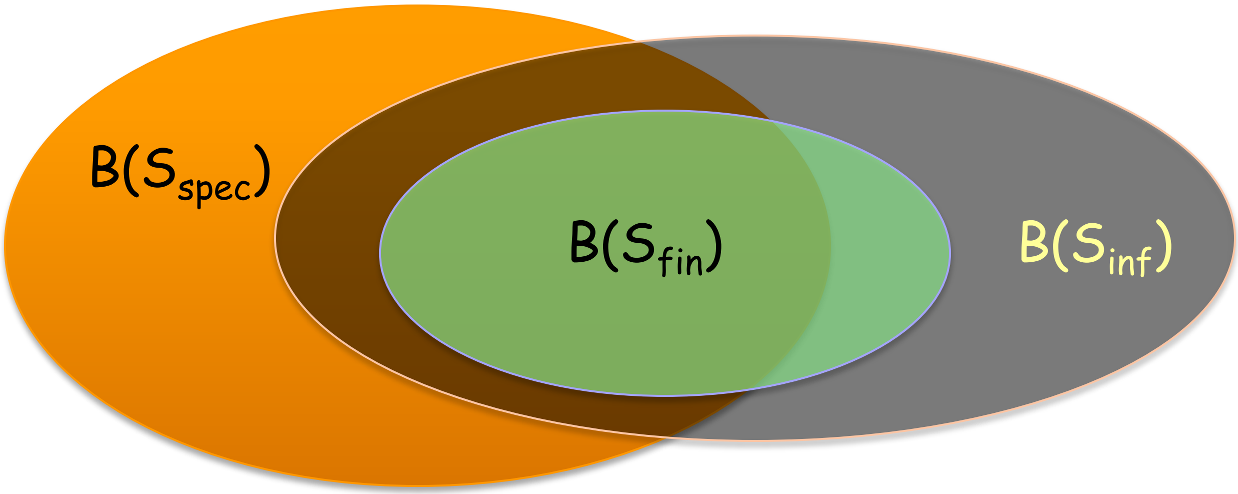
Verification Problem

- Given a infinite-state system S_{inf}, find whether S_{inf} satisfies the specification S_{spec}
- In some cases, we can find a finite-state system S_{fin} such that $S_{inf} \preceq_S S_{fin}$
- Then we can solve the verification problem for S_{fin}, i.e., is $S_{fin} \preceq_S S_{spec}$, which implies $S_{inf} \preceq_S S_{spec}$

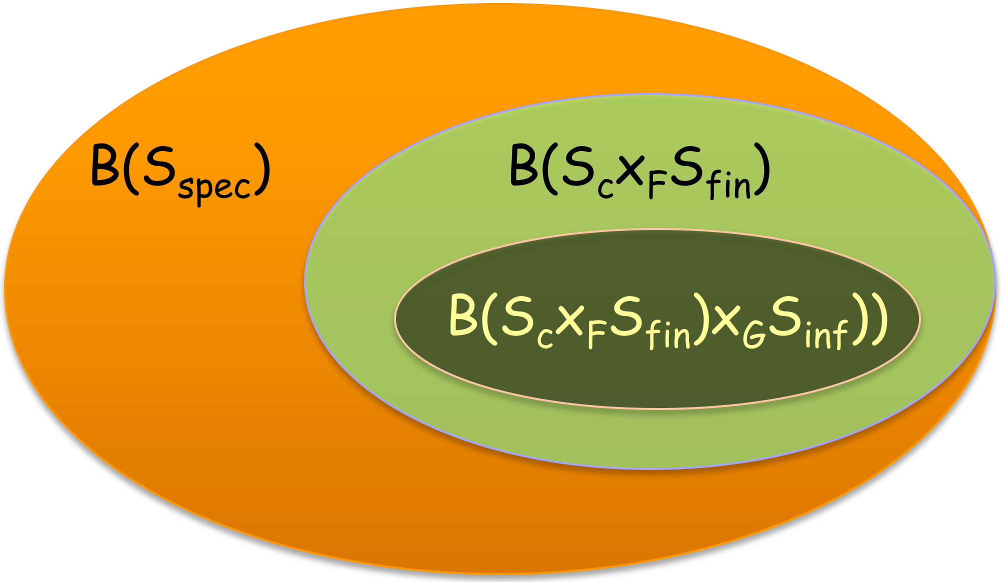


Control Problem

- Given a infinite-state system S_{inf}, find a controller $S_{c'}$ such that $S_{c'} \times_F S_{inf}$ satisfies a specification S_{spec}
- In some cases, we can find a finite-state system S_{fin} such that $S_{fin} \preceq_{AS} S_{inf}$
- We know how to design a controller S_c for S_{fin}. How can we use S_c to control S_{inf}?



Control Problem



Controller Refinement

- Proposition 8.7
- $S_c \preceq_{AS} S_a$ and $S_a \preceq_{AS} S_b$ then $S_c \times_F S_a \preceq_{AS} S_b$

- Proof consider the alternating bisimulation relation
  - ${}_{ca}R_b = \{((x_c, x_a), x_b) \mid (x_c, x_a) \in {}_cR_a \text{ and } (x_a, x_b) \in {}_aR_b\}$

Controller Refinement

- We design S_c for S_{fin} so that it satisfies S_{spec}
  - $S_c \preceq_{AS} S_{fin}$
  - $S_c x_F S_{fin} \preceq_{AS} S_{spec}$

- Assume $S_{fin} \preceq_{AS} S_{inf}$
  - then $S_c x_F S_{fin} \preceq_{AS} S_{inf}$
  - I.e. $S_c x_F S_{fin}$ is feedback composable with S_{inf} and

- Let $S_{c'} = S_c x_F S_{fin}$
  - then $S_{c'} x_F S_{inf} \preceq_{AS} (S_c x_F S_{fin}) x_F S_{inf} \preceq_{AS} (S_c x_F S_{fin}) \preceq_{AS} S_{spec}$

Controller Refinement

- If S_c does not exist for S_{fin}, does not imply that $S_{c'}$ does not exist for S_{inf}
- Assume $S_{fin} \cong_{AS} S_{inf}$, then a controller exists for S_{fin} iff a controller exists for S_{inf}
  - If a controller exists for S_{inf}, then $S_{c'} x_F S_{inf} \preceq_{AS} S_{spec}$
  - $S_{c'} x_F S_{inf} \preceq_{AS} S_{fin}$
  - $S_c x_F S_{fin} \preceq_{AS} (S_{c'} x_F S_{inf}) x_F S_{fin} \preceq_{AS} (S_{c'} x_F S_{inf}) \preceq_{AS} S_{spec}$