ISTITUTO
DI TECNOLOGIE DELLA
COMUNICAZIONE,
DELL'INFORMAZIONE
E DELLA
PERCEZIONE

Scuola Superiore
Sant'Anna

Course Introduction to Matlab and Simulink - Stateflow

Emanuele Ruffaldi

June 08, 2017

http://www.eruffaldi.com/wp/introduction-to-matlab-and-simulink/

Scuola Superiore Sant'Anna, Pisa

Event-driven Systems

- Real world discrete Systems
  - Something happens
  - A decision has to be taken
  - Pump, Sensors …
- How can we represent such systems?
  - Theory: Finite-State Machines
  - Implementation
    - Programmatically
    - Visually

STATE MACHINES

Finite-State Machines

- Decompose the problem in states and transitions between them
- Transitions are controlled by conditions
- Transition (or non transition) can be associated to operation
- Finite State Machines (FSM) use a finite number of states

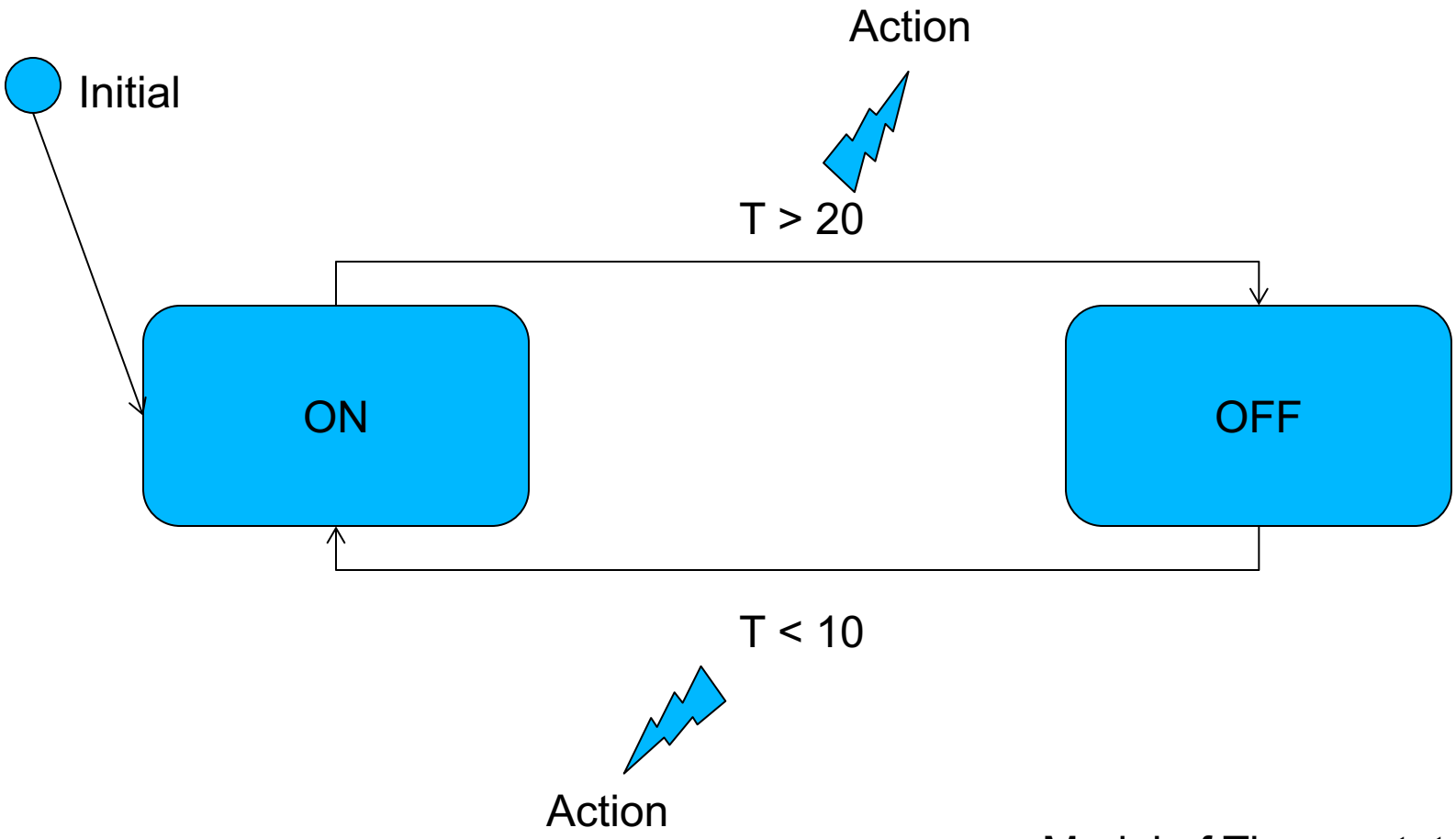
Extensions

- Hierarchical FSM
- Parallel FSM
- Temporal Logic
- Petri Net (multiple states, multiple active conditions)

FSM Mathematics

- A FSM can be described as (A,S,s0,d,F)
  - A is the input Alphabet
  - S is finite state
  - s0 is the initial state
  - d is the transition function
  - F is the set of final states
- In the case of generic conditions for state transitions we associate each expression to a symbol in the input Alphabet
- If the FSM generates also an output we have an output alphabet triggered by state transitions
  - Optionally self state transitions when idle

FSM Graphical Representation



Model of Thermostat with Hysteresis

SIMULINK STATEFLOW

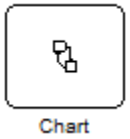
Simulink Stateflow

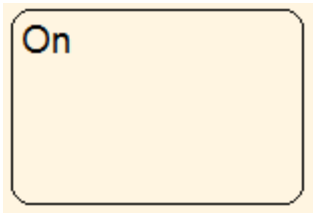
- Event-Driven modeling
- Serial or Parallel State Machines
- Visual Editing
- Code generation for Real Time Workshop

Entities

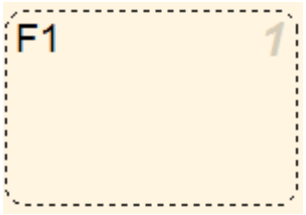
1. States
2. Events (input,output,local) – Naming is Capitalize
3. Variables (input,output,local) – Naming is lowercase
4. Transitions by Events or by Conditions
5. Actions on State's step or on Transition
6. Integration in Simulink as Discrete Subsystem

Graphical Notation





State



Parallel State

Initial State

>> Example sf_simple

Action on States

- Actions can be associated to States on Phase
- Phases:
  - entry (en)
    - when the State becomes active
  - during (du)
    - a step in which the state is maintained
  - exit (ex)
    - when a state is left
  - on “event”
    - available while State is in During phase
- Stateflow action language has a syntax similar to M but actually different
  - MATLAB accessible as ml.matfunc. E.g. pause(seconds)
- Naming an Event triggers it

>> Example sf_simplemat

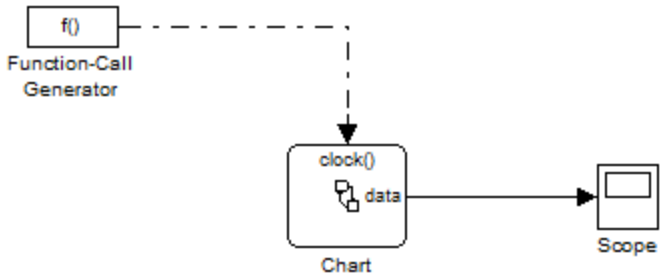
Transitions

- Transitions among States are triggered by conditions or events
- Syntax:
  - **event_trigger[condition]{condition_action}/transition_action**
  - event_trigger is the name of an event
  - condition is an expression
  - condition_action is a statement
  - transition_action

- Condition Expression
  - boolean operators on variables
  - in(state_name)

Stateflow Chart Activation

- Possible Activations
  - Sampling
  - Signal or Function Call for Trigger
  - Connected to other Chart
  - Default **tick** or wakeup



Use Function-Call generator for triggering the Chart at specified sampletime

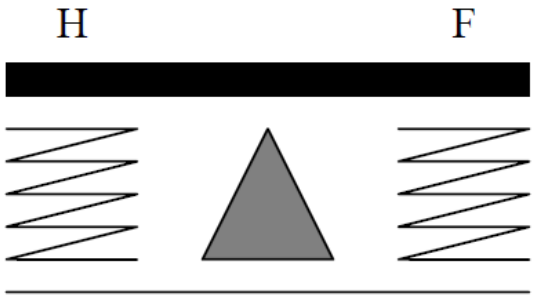
>> Example sf_clock

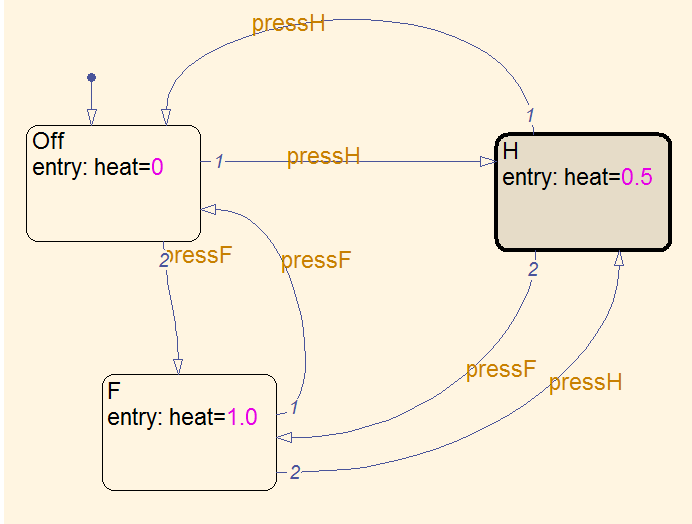
Workflow

1. Define interface with Simulink
2. Define States
3. Define State Actions
4. Define Transition
5. Simulation outside Simulink
6. Integration

Example

- Problem
  - Pushbutton for controlling heating in a Car
  - Pressing H gives 50% heating
  - Pressing F gives 100% heating
  - Pressing again an active H or F gives stop
- Actions
  - Create States
  - Define inputs
  - Define output
  - Define transitions
- Notes
  - Set Chart as Discrete
  - Events vs Conditions

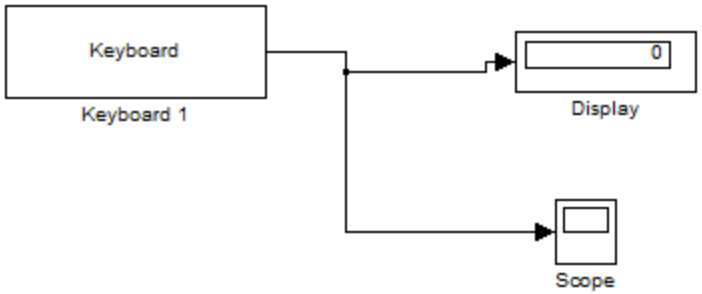


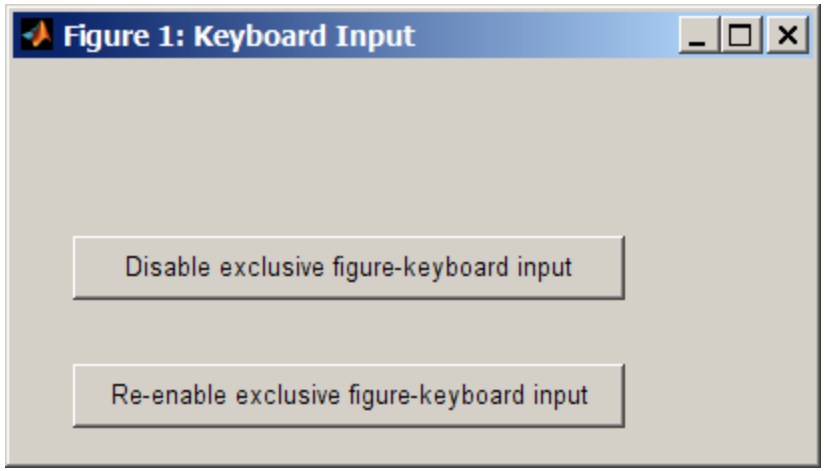


>> Example sf_heat

Keyboard input

- Getting input from the Keyboard using SIMULINK
- Modal Figure

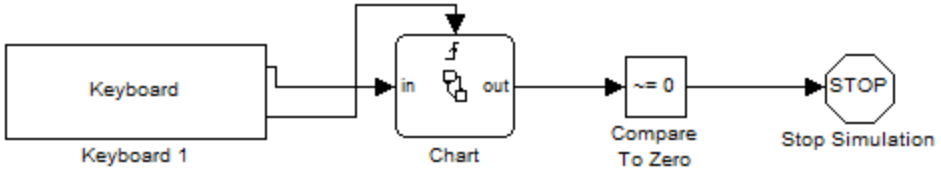


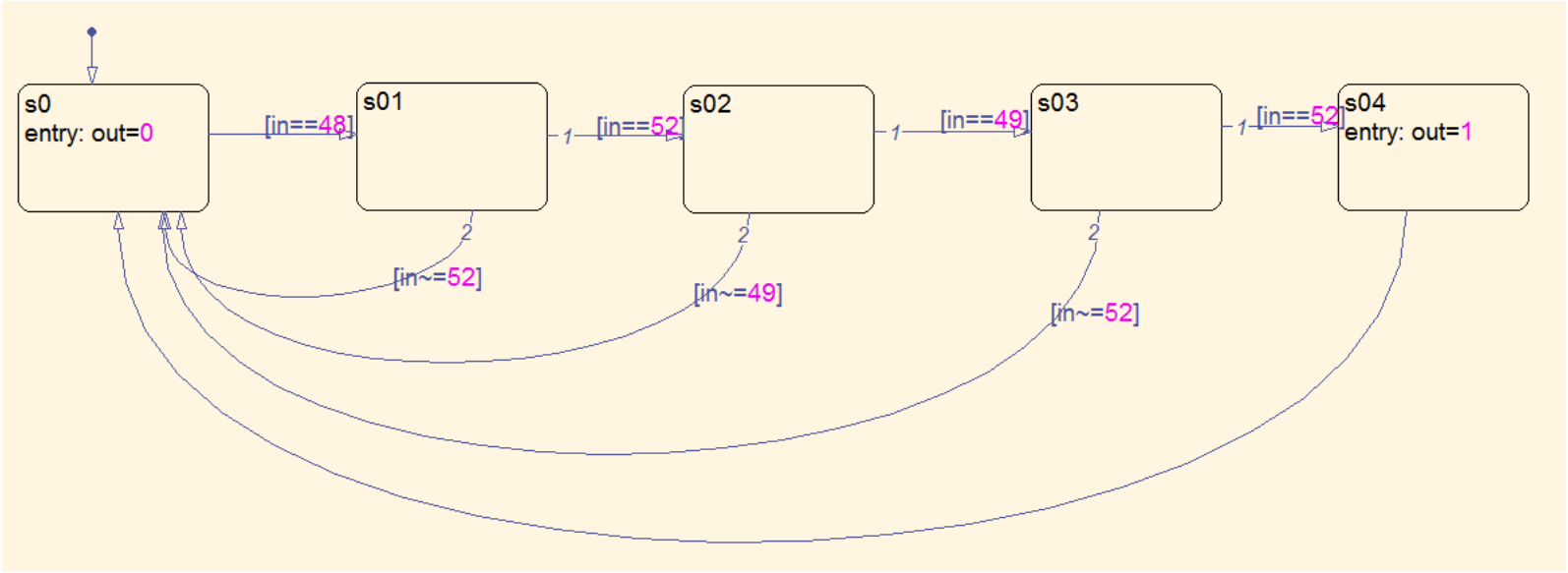


>> Example sf_keyboard

Password Recognition

- Use keyboard: 0414



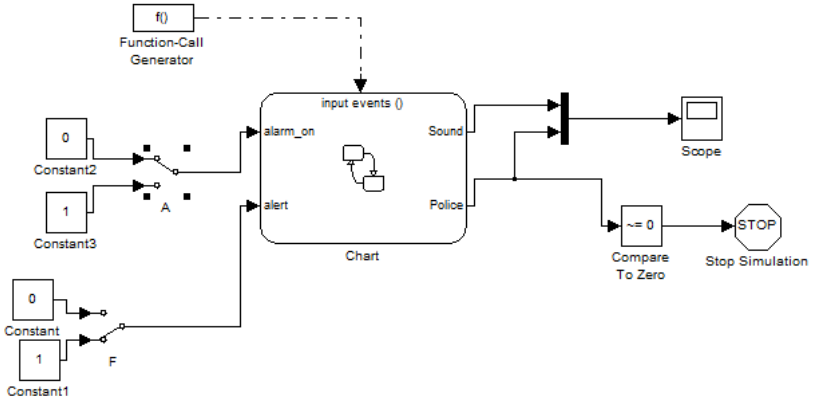


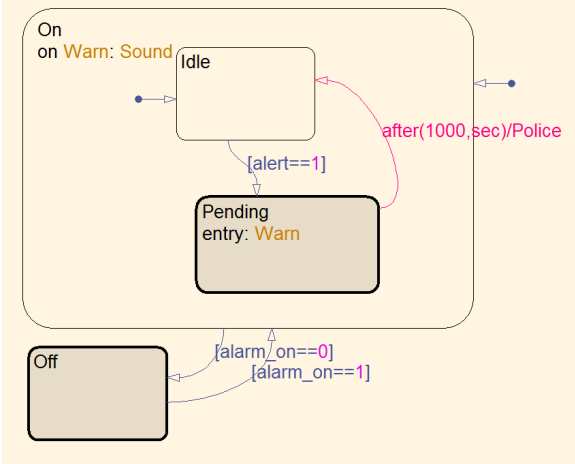
Hierarchical States

- Transition to a Hierarchical State
  - If Parallel
    - Enter every contained State
  - If Serial
    - Select the default State
    - Enter the default State
- Transition from a Hierarchical State
  - Exits all the active States

Alarm System

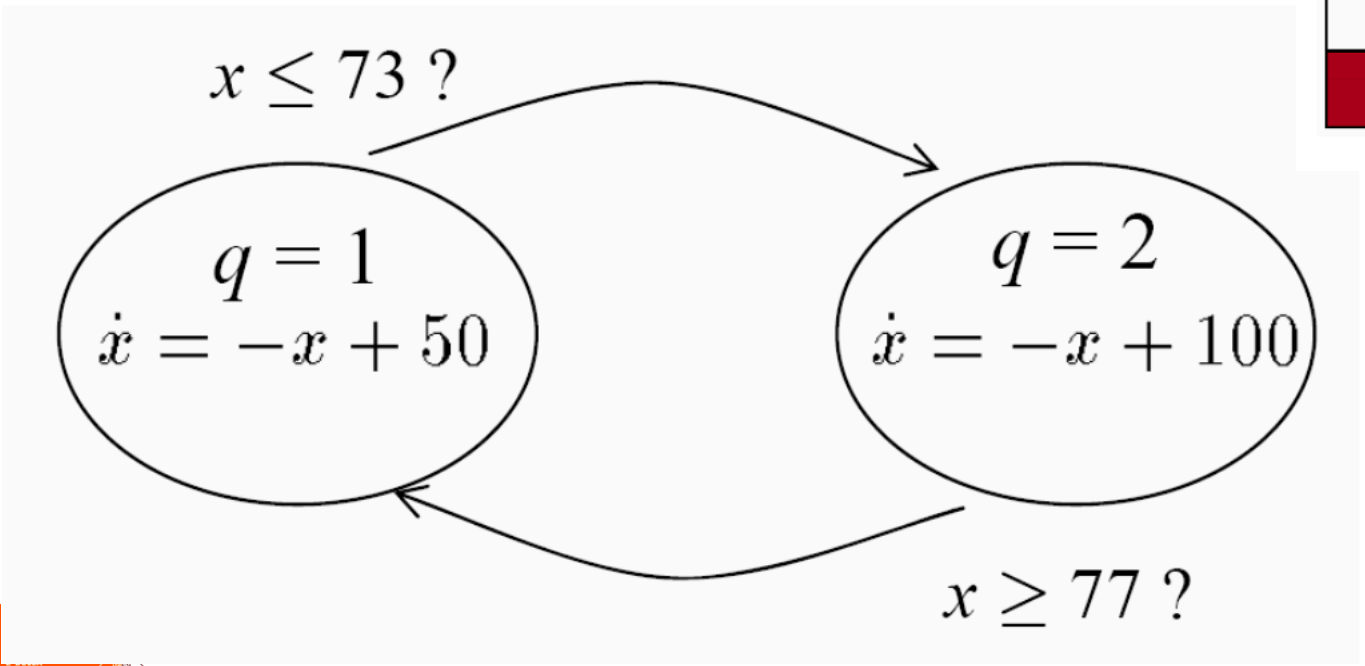
- Alarm
  - Can be Disabled
  - If triggered and not deactivated by given time calls police
- Input Values
  - alarm_on
  - alert
- Output Values
  - Sound
  - Police
- Temporal Logic
  - Condition based on time
  - e.g. after(10,myevent)
  - tlo(n,E)[C]
    - tlo is an operator
    - n is a time
    - E is an event
    - C a condition
  - tlo(n,sec)[C]
    - n is seconds
    - sec is a keyword
- Comples States can be “SubCharted”
  - Context Menu, Make Content, “SubCharted”

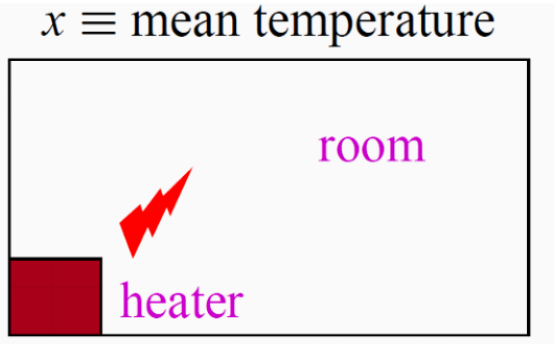




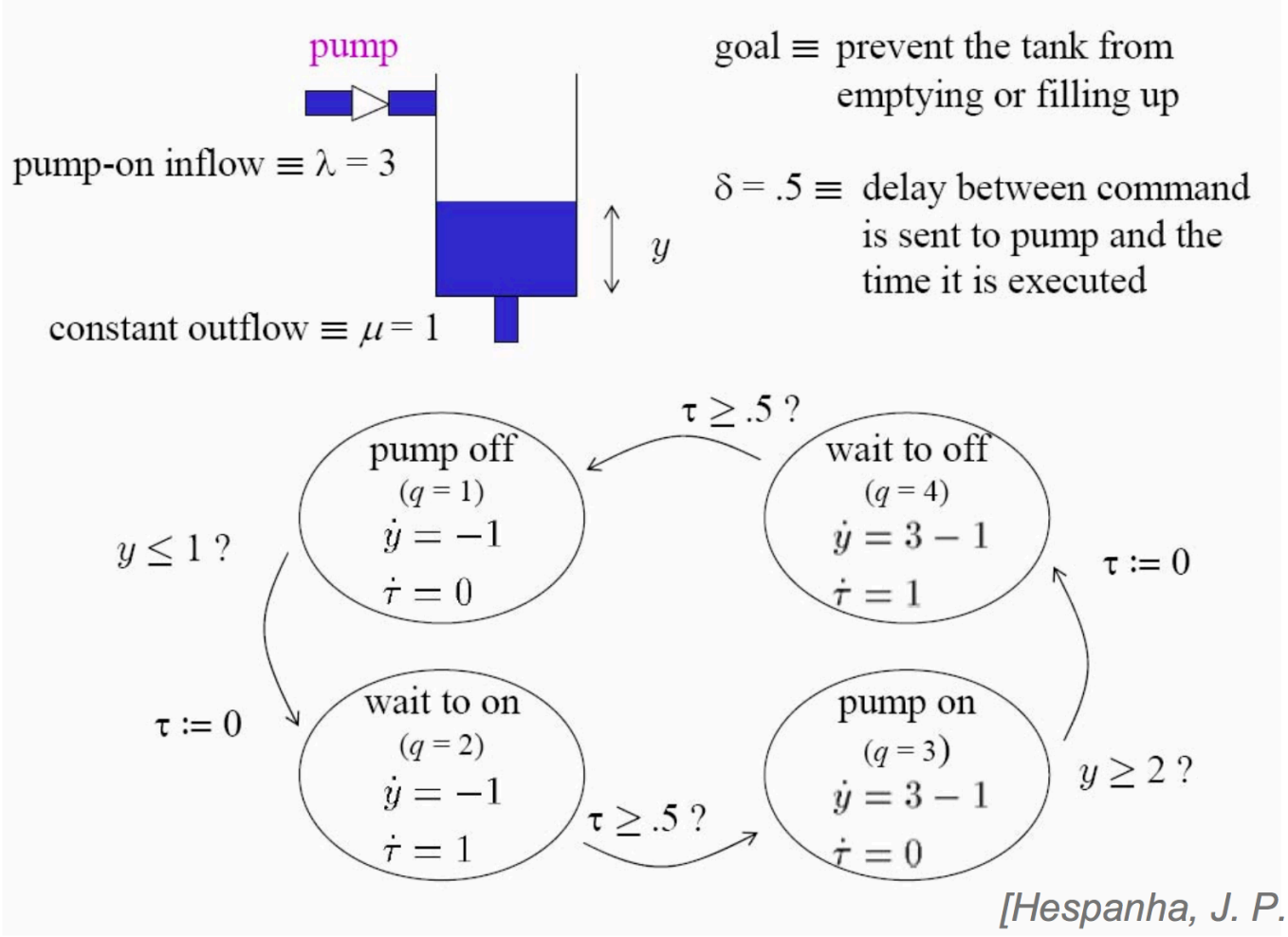
Hybrid Automata

- Real simulations involve a combination of continuous and discrete elements



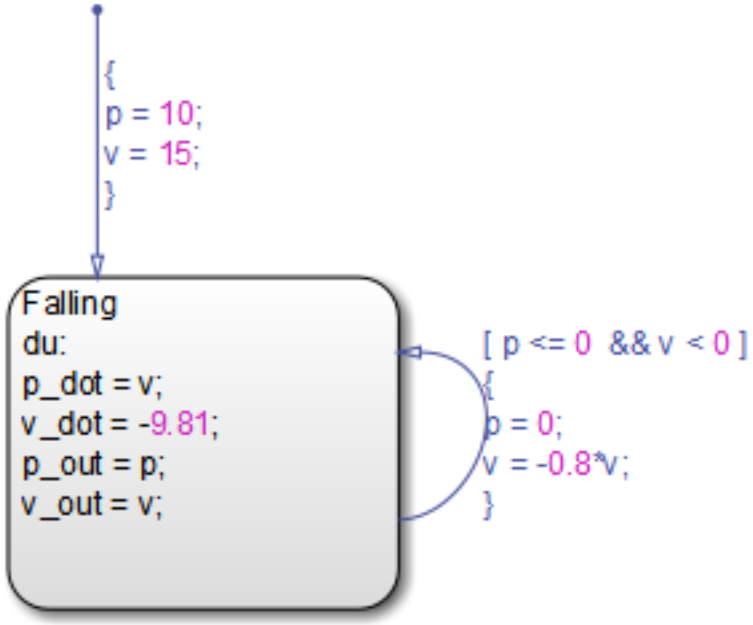


Hybrid Other



[Hespanha, J. P. 05]

Falling Ball from Matlab




```
>> sf_ball
```