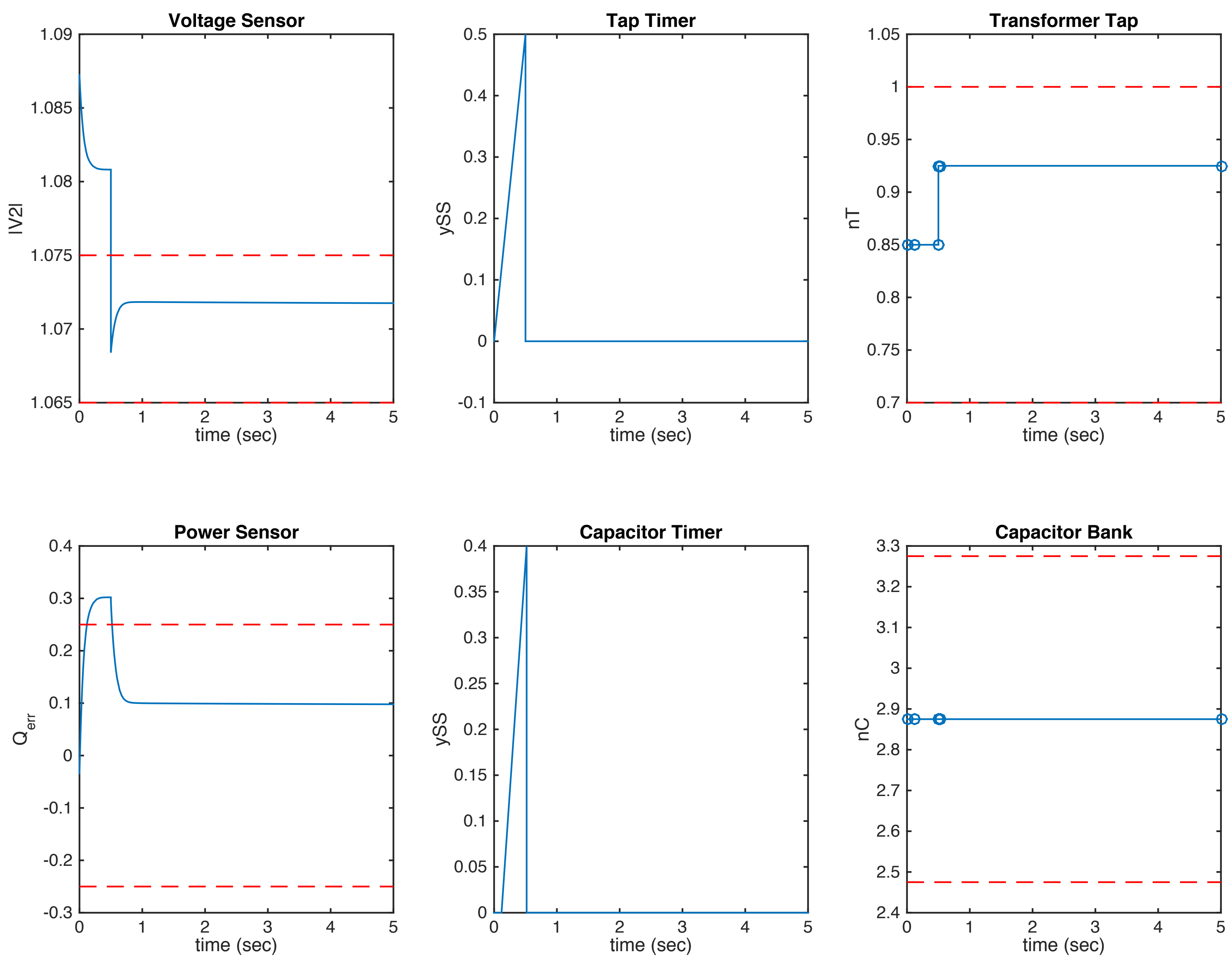
LECTURE 6: Hybrid Models

Modeling and Simulation 2

Daniel Georgiev

CASE STUDY HYBRID EXECUTION

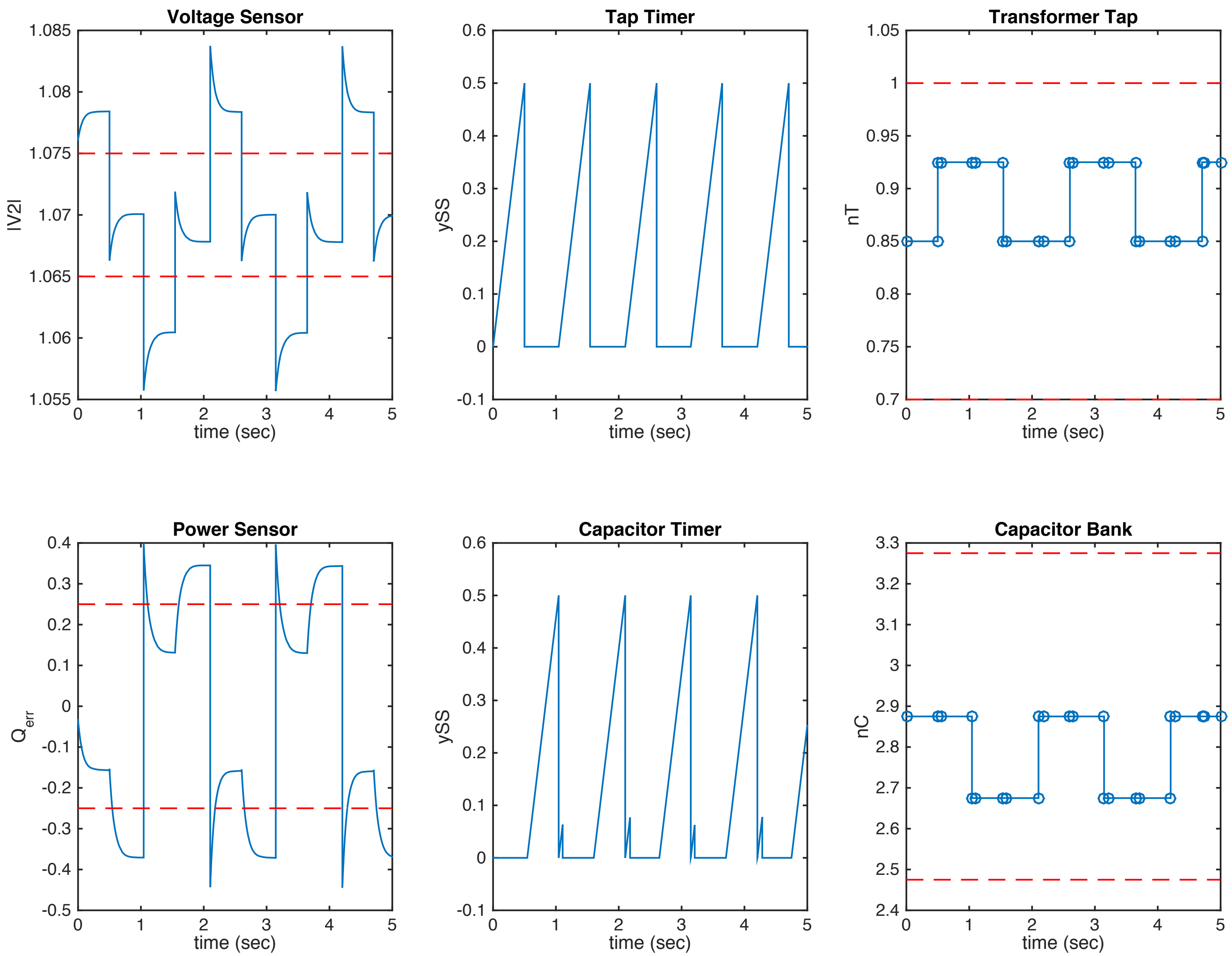


X2: .05 => 0.0515

PS: 3.0 => 2.9624

TM: 8.0 => 8.0205

CASE STUDY HYBRID EXECUTION

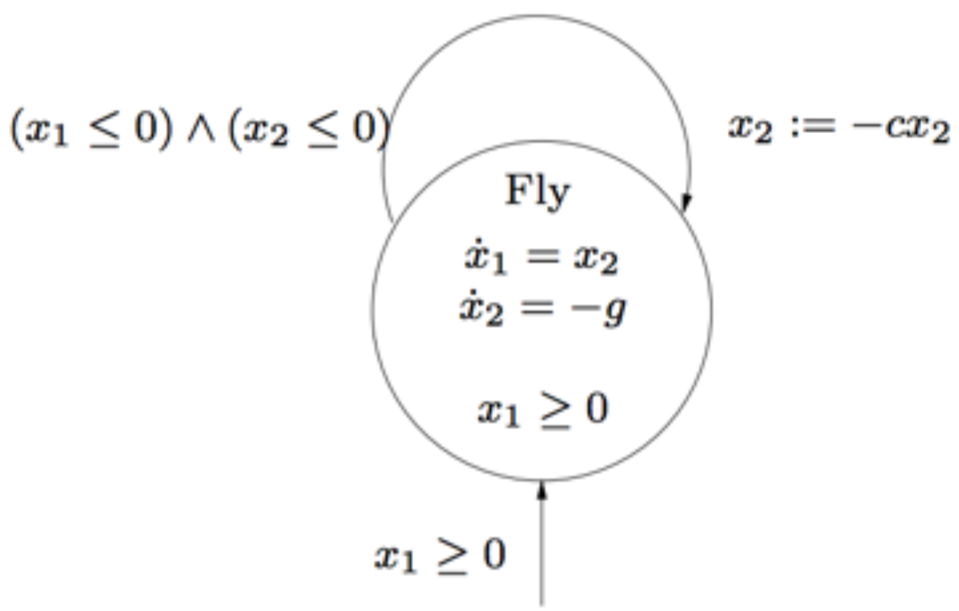


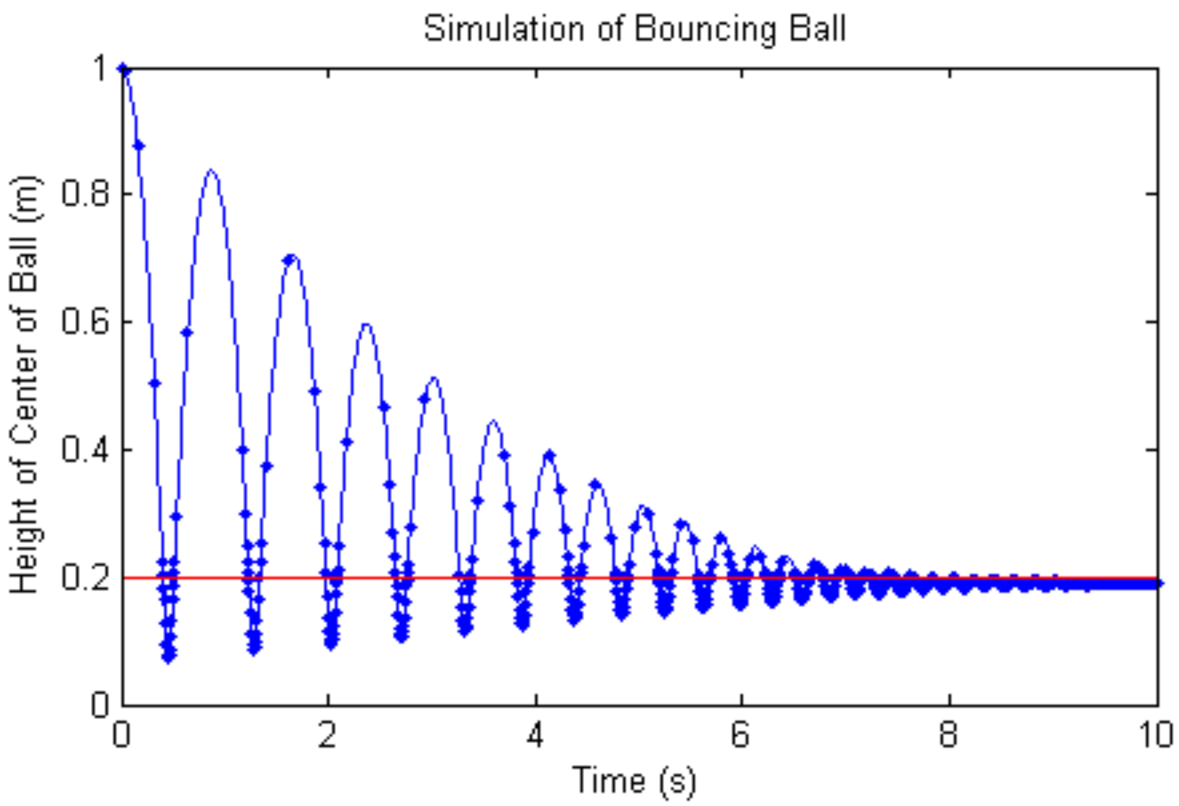
X2: .05 => 0.0503

PS: 3.0 => 2.9667

TM: 8.0 => 8.2820

SIMPLE EXAMPLE: BOUNCING BALL





SIMPLE EXAMPLE: WATER TANK

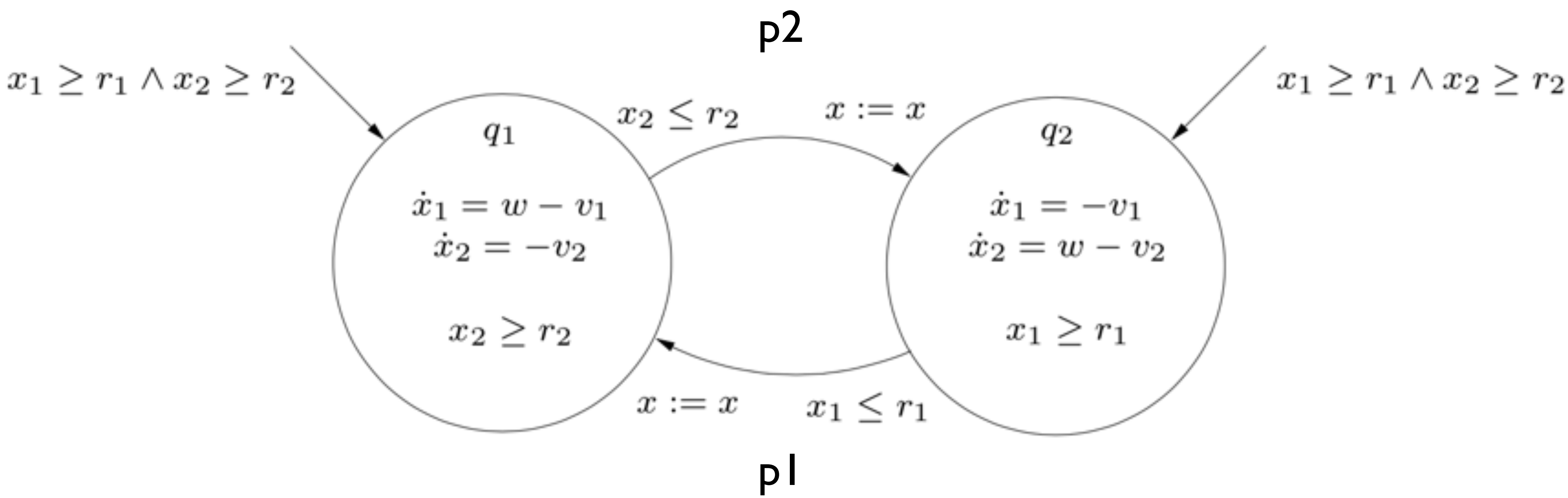


Figure 3.7: Graphical representation of the water tank hybrid automaton.

$$(Q, E, q(0), X, Z, x(0), z(0), f, g, R)$$

HYBRID AUTOMATA DEFINITION

$$Q = \{q_1, q_2, ...\} \tag{1}$$

$$E = \{e_1, e_2, ...\} \tag{2}$$

$$q(0) = \text{initial discrete state} \tag{3}$$

$$X = \mathbb{R}^n \tag{4}$$

$$Z = \mathbb{R}^p \tag{5}$$

$$x(0) = \text{initial continuous state} \tag{6}$$

$$z(0) = \text{initial dynamic parameter value} \tag{7}$$

$$g : Q \times X \to Q \tag{8}$$

$$f : X \times Z \to X \tag{9}$$

$$R : Q \times X \times Z \to X \times Z \tag{10}$$

CONTINUOUS BEHAVIOR

Theorem 2.1 (Existence & Uniqueness of Solutions) *If f is Lipschitz continuous, then the differential equation*

$$\dot{x} = f(x)$$
$$x(0) = x_0$$

has a unique solution $x(\cdot) : [0, T] \to \mathbb{R}^n$ for all $T \geq 0$ and all $x_0 \in \mathbb{R}^n$.

Theorem 2.2 (Continuity with Initial State) *Assume f is Lipschitz continuous with Lipschitz constant λ. Let $x(\cdot) : [0, T] \to \mathbb{R}^n$ and $\hat{x}(\cdot) : [0, T] \to \mathbb{R}^n$ be solutions to $\dot{x} = f(x)$ with $x(0) = x_0$ and $\hat{x}(0) = \hat{x}_0$ respectively. Then for all $t \in [0, T]$*

$$\|x(t) - \hat{x}(t)\| \leq \|x_0 - \hat{x}_0\| e^{\lambda t}$$

HYBRID TRAJECTORIES

Definition 3.2 (Hybrid Time Set) *A hybrid time set is a sequence of intervals* $\tau = \{I_0, I_1, \ldots, I_N\} = \{I_i\}_{i=0}^{N}$*, finite or infinite (i.e.* $N = \infty$ *is allowed) such that*

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

Definition 3.4 (Execution) *An execution of a hybrid automaton* H *is a hybrid trajectory,* (τ, q, x)*,*

HYBRID TRAJECTORIES

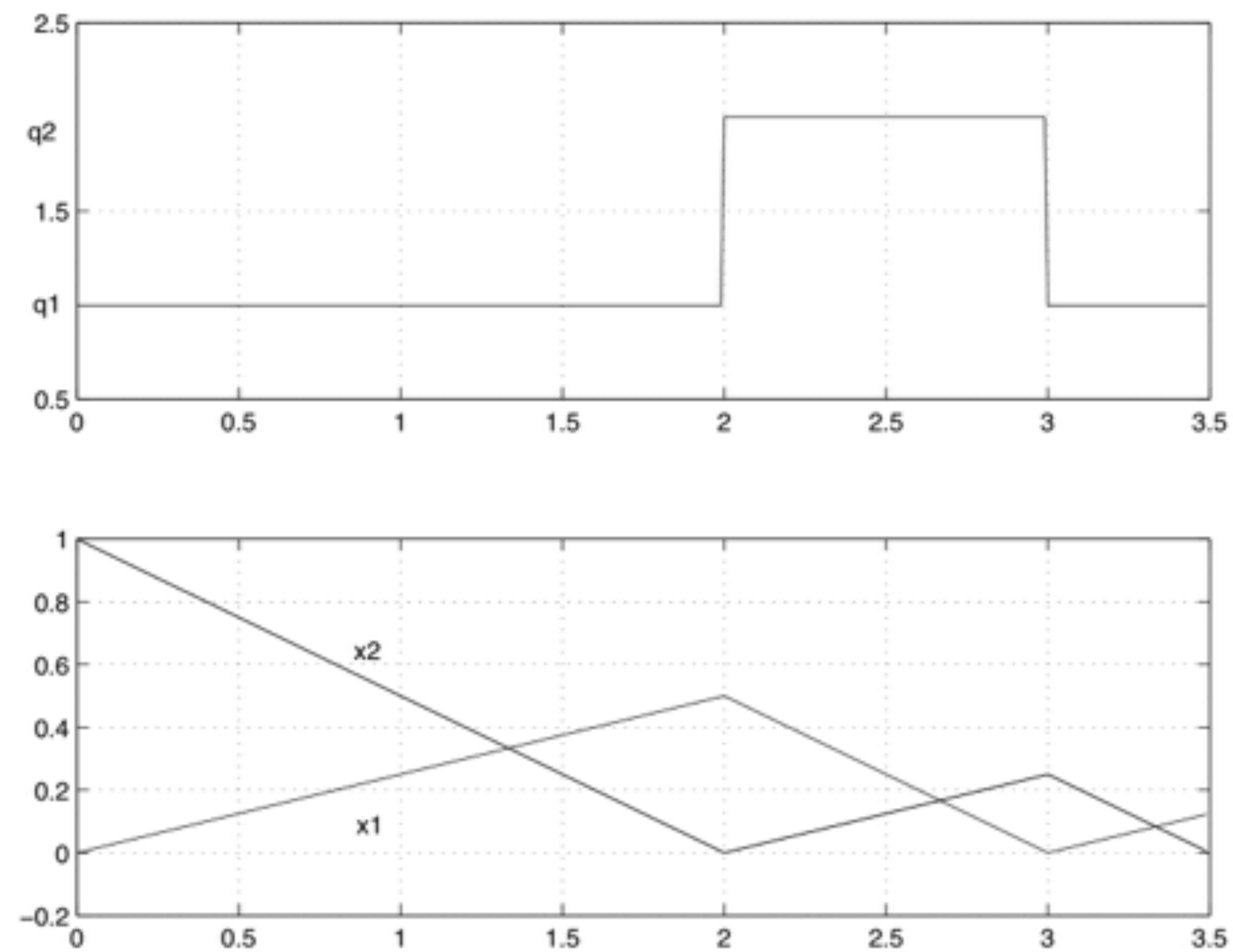


Figure 3.10: Example of an execution of the water tank hybrid automaton.

Example (Water Tank (cont.)) Figure 3.10 shows an execution of the water tank automaton. The hybrid time set τ of the execution consists of three intervals, $\tau = \{[0,2],[2,3],[3,3.5]\}$. The evolution of the discrete state is shown in the upper plot, and the evolution of the continuous state is shown in the lower plot. The values chosen for the constants are $r_1 = r_2 = 0$, $v_1 = v_2 = 1/2$ and $w = 3/4$. The initial state is $q = q_1$, $x_1 = 0$, $x_2 = 1$. ■

HYBRID BEHAVIOR

Definition 3.5 (Classification of executions) *An execution* (τ, q, x) *is called:*

- **Finite**, *if* τ *is a finite sequence and the last interval in* τ *is closed.*
- **Infinite**, *if* τ *is an infinite sequence, or if the sum of the time intervals in* τ *is infinite, i.e.*

$$\sum_{i=0}^{N} (\tau_i' - \tau_i) = \infty.$$

- **Zeno**, *if it is infinite but* $\sum_{i=0}^{\infty} (\tau_i' - \tau_i) < \infty$.
- **Maximal** *if it is not a strict prefix of any other execution of* H.

CLASSIFICATION OF EVENTS

Consider a pair of discrete states (q1,q2) and an event e1 in E. In the state q1, the following are classifications of the event.

- Instant event: g(q1,x) = q2, for all x in Dom(q1)
- Impossible events: g(q1,x) is not defined, for all x in Dom(q1)
- Overlapping events: for some q3, g(q1,x) = q2 and g(q1,x) = q3
- Toggle events: g(q1,x) = q2 and g(q2,R(q1,x)) = q1

WITHOUT CONTINUITY WEIRD THINGS ARE POSSIBLE

Depending on the types of events present, various behaviours may arise.

1) Nondeterminism: overlapping events

2) Zeno trajectories: overlapping events or toggle events

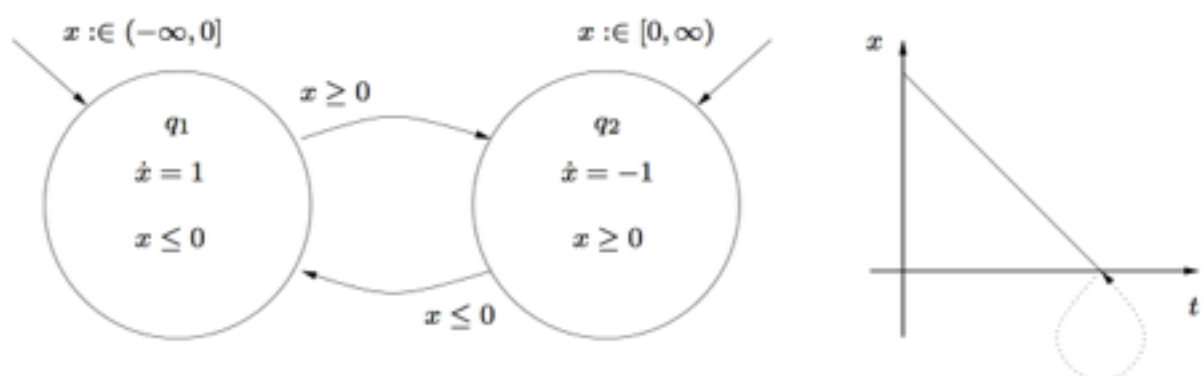


Figure 4.2: Chattering system.

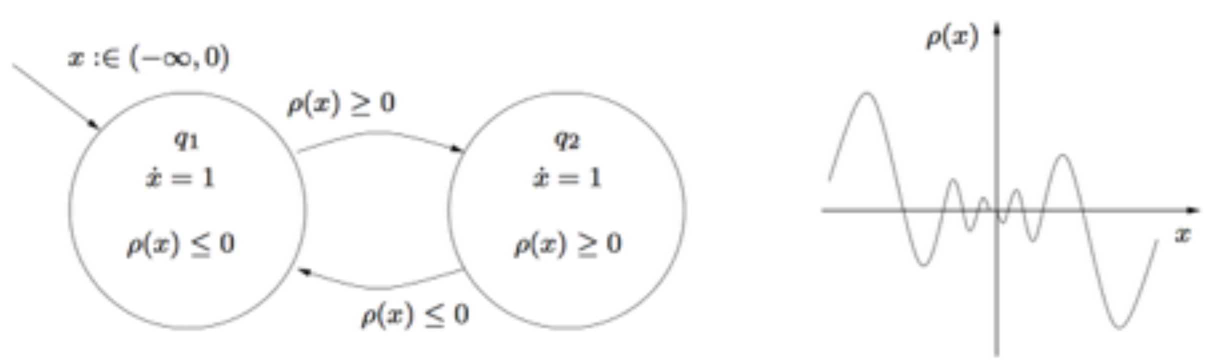


Figure 4.3: System with a smooth, non-analytic domain.

UNIQUENESS AND NON-BLOCKING CONDITIONS

[for those interested see Lygeros '95]

Denote states reachable from *Init* by the set *Reach* and transition states on the boundary of *Dom(q)*, q in Q, by the set *Trans*.

Theorem 4.1 (Existence and Uniqueness) *A hybrid automaton H accepts a unique infinite execution for each initial state if it satisfies all the conditions of Lemmas 4.1 and 4.2.*

Lemma 4.1 *A hybrid automaton, H, is non-blocking if for all $(\hat{q}, \hat{x}) \in \mathrm{Reach} \cap \mathrm{Trans}$, there exists $\hat{q}' \in Q$ such that $(\hat{q}, \hat{q}') \in E$ and $\hat{x} \in G(\hat{q}, \hat{q}')$. If H is deterministic, then it is non-blocking if and only if this condition holds.*

Lemma 4.2 *A hybrid automaton, H, is deterministic if and only if for all $(\hat{q}, \hat{x}) \in \mathrm{Reach}$*

1. *if $\hat{x} \in G(\hat{q}, \hat{q}')$ for some $(\hat{q}, \hat{q}') \in E$, then $(\hat{q}, \hat{x}) \in \mathrm{Trans}$;*
2. *if $(\hat{q}, \hat{q}') \in E$ and $(\hat{q}, \hat{q}'') \in E$ with $\hat{q}' \neq \hat{q}''$ then $\hat{x} \notin G(\hat{q}, \hat{q}') \cap G(\hat{q}, \hat{q}'')$; and,*
3. *if $(\hat{q}, \hat{q}') \in E$ and $x \in G(\hat{q}, \hat{q}')$ then $R(\hat{q}, \hat{q}', \hat{x}) = \{\hat{x}'\}$, i.e. the set contains a single element, $\hat{x}'$.*

CASE STUDY

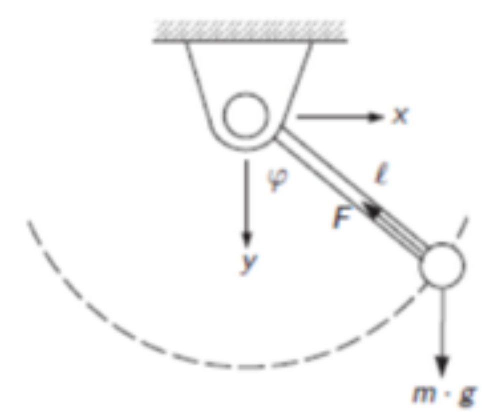
In order to apply hybrid system formalism to case study, first we'll look at Matlab's numerical solver.

CONTINUOUS STATE TRANSITIONS

- Ordinary differential equations (ode) represent the basic modelling formalism for continuous systems. ODE's do not lend themselves well to modular modelling.
- Many system have algebraic constraints.

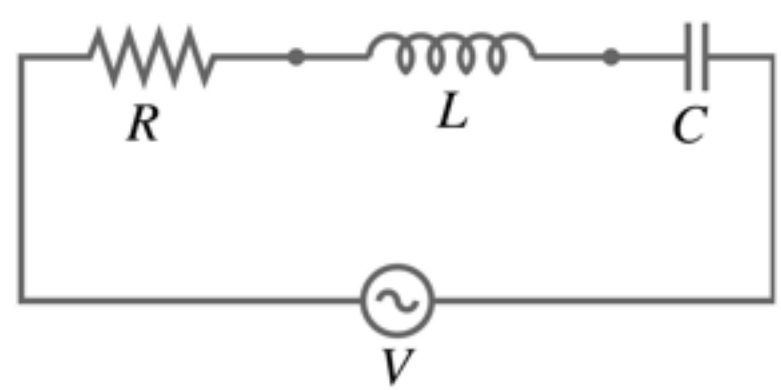
geometrical constraints

$$m\ddot{x} = -\frac{F}{l}x$$
$$m\ddot{y} = mg\frac{F}{l}y$$
$$l^2 = x^2 + y^2$$



conservation laws

$$\dot{V}_C = \frac{1}{C}i_L$$
$$\dot{V}_L = \frac{1}{L}i_L$$
$$0 = V_R + Ri_E$$
$$0 = V_E + V_R + V_C + V_L$$
$$0 = i_L - i_E$$



DIFFERENTIAL ALGEBRAIC EQUATIONS (DAEs)

A system of equations of the form

$$F(t, x, \dot{x}) = 0$$

is called a differential algebraic equation (DAE) if the Jacobian matrix

$$\frac{\partial F(t,x,\dot{x})}{\partial \dot{x}}$$

is singular.

In general, if the Jacobian matrix is non-singular, then the system can be transformed into an ODE.

The following system is a DAE.

$$x_1 - \dot{x}_1 + 1 = 0$$

$$\dot{x}_1 x_2 + 2 = 0$$

NETWORK

$$0 = \frac{n}{Z_1}(V_0 - nV_2) - I_S - I_C - \frac{1}{Z_2}(V_2 - V_3)$$

$$0 = S_D - \frac{1}{Z_2}(V_2 - V_3)V_3^*$$

$$\frac{dV_0}{dt} = 0$$

TURBINE

$$0 = -V_2 + I_S(R_S + X_S i) + I_R(X_M i)$$

$$0 = I_S(X_M i) + I_R X_R i - I_R \frac{R_R}{w_R/w_S - 1}$$

$$J_G \dot{w}_R = T_M + Imag(I_S(I_S X_S/W_S + I_R X_M/W_S)^*)$$

LOAD

$$\dot{x}_D = -\frac{1}{T_P}(P_D - P_S)$$
$$0 = P_D - x_D - P_T|V_3|_2^2$$
$$0 = Q_D$$

TIMERS

$$\dot{x}_{TC} = onC \qquad \dot{x}_{TT} = onT$$

$$y_{TC} = x_{TC} - tresC \qquad y_{TT} = x_{TT} - tresT$$

CAPACITOR BANK

$$0 = I_C - (V_2 C n_C) i$$

DAE CLASSIFICATION

- semi-explicit

$$\dot{x} = f(t, x, z)$$
$$0 = g(t, x, z)$$

- fully implicit - any fully-implicit DAE can be transformed into a semi-explicit DAE

  our systems are semi-explicit

- index of DAE

  if we differentiate the algebraic equation, we can get an ODE system

  what is the index of the following system?

$$x_1 - \dot{x}_1 + 1 = 0$$
$$\dot{x}_1 x_2 + 2 = 0$$

REMARK: The index of DAE is the number of differentiations required to get an ODE. Index 1 and 2 systems are easily solvable by off-the-shelf solvers (MATLAB). Hence it is useful to formulate the system as a low index system.

SOLUTION: LINEAR DAEs (index = 1)

Consider the semi-explicit DAE

$$\dot{x} = f(t, x, z)$$
$$0 = g(t, x, z)$$

differentiating the algebraic equation yields

$$0 = \frac{\partial g(t,x,z)}{\partial t} = \frac{\partial g}{\partial t} + \frac{\partial g}{\partial x}\dot{x} + \frac{\partial g}{\partial z}\dot{z}$$

Index 1 implies the Jacobian with respect to z is non-singular. Hence

$$\dot{z} = -\frac{\partial g}{\partial z}^{-1}\left(\frac{\partial g}{\partial t} + \frac{\partial g}{\partial x}f(t, x, z)\right)$$

REMARK: for higher indices, when the Jacobian is singular, this method is not efficient.