HYBRID SYSTEMS

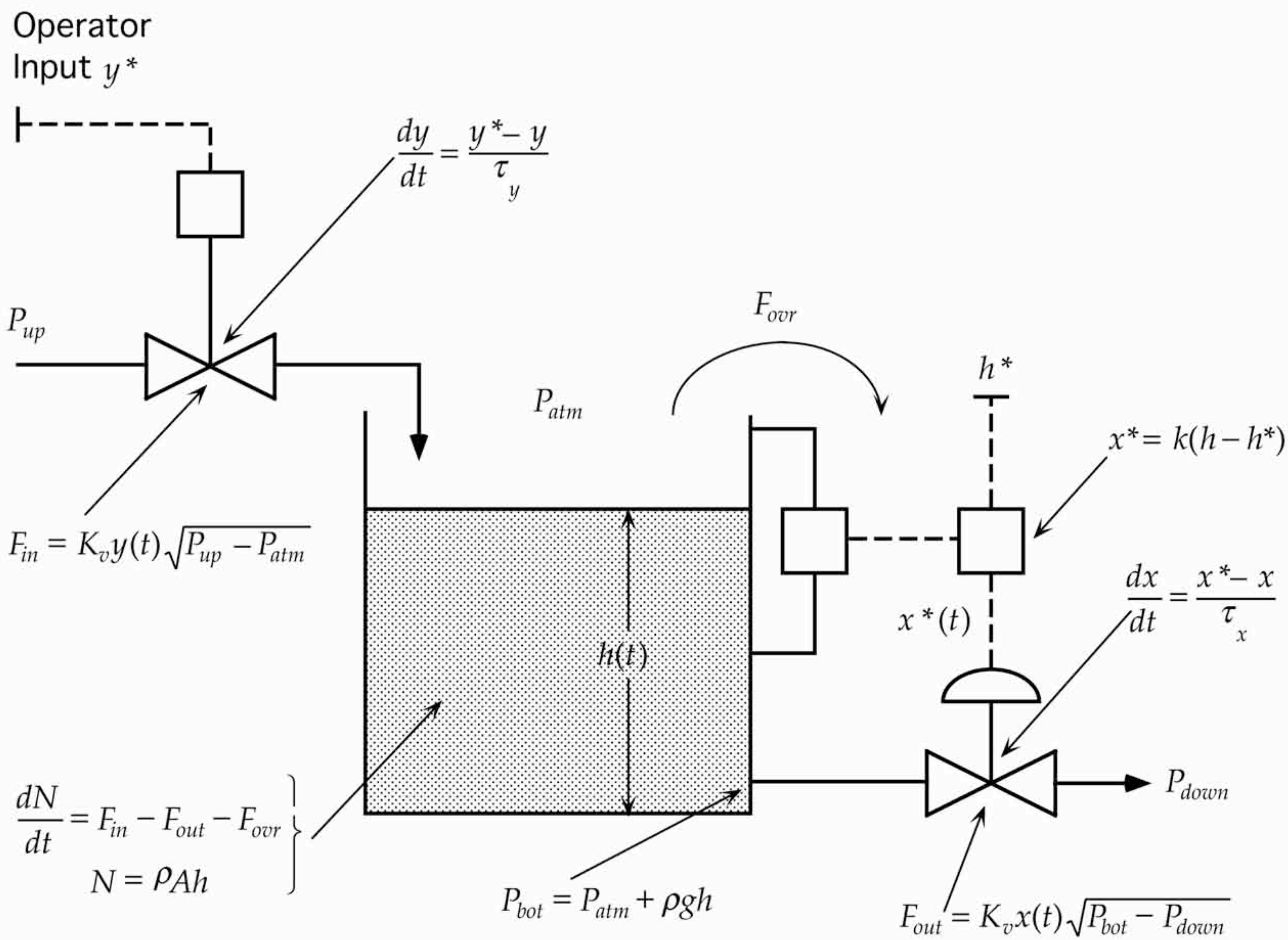
Paul I. Barton
Process Systems Engineering Laboratory
Department of Chemical Engineering
Massachusetts Institute of Technology
Cambridge MA 02139

Outline

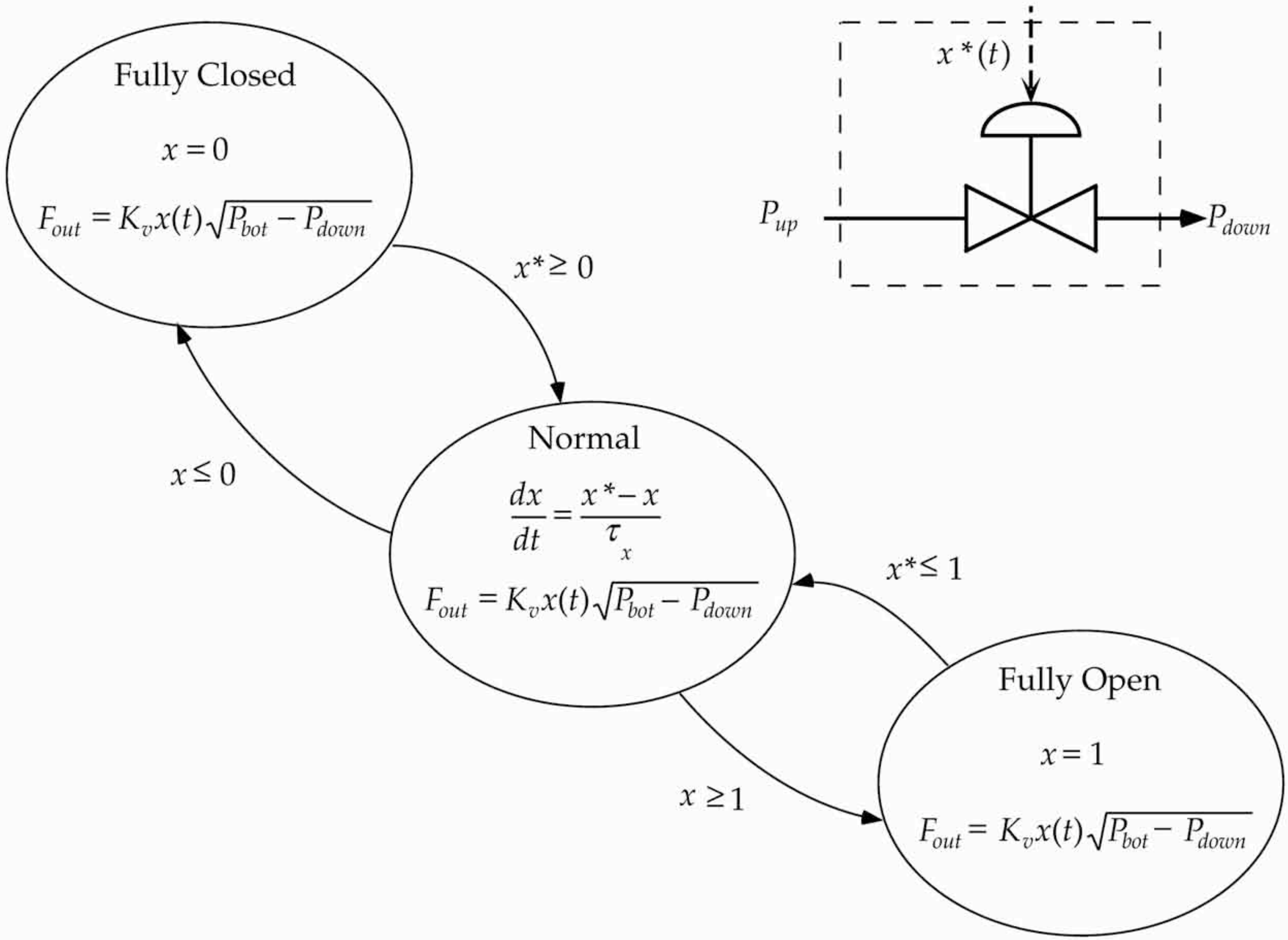
- Modeling
- Simulation
- Sensitivity Analysis
- Optimization

MODELING

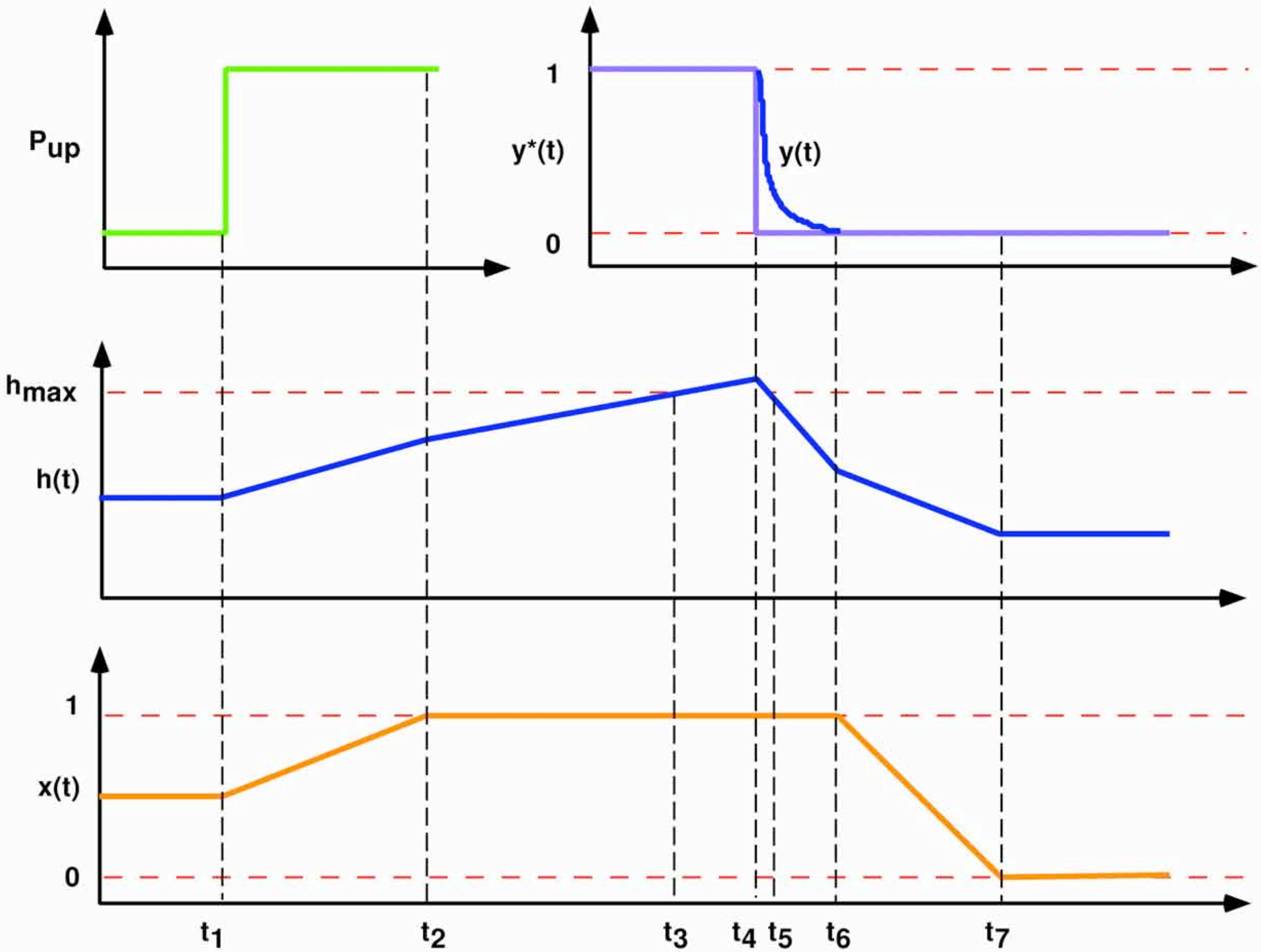
Discrete/Continuous Dynamics Tank Illustration



Discrete/Continuous Dynamics Valve Automaton



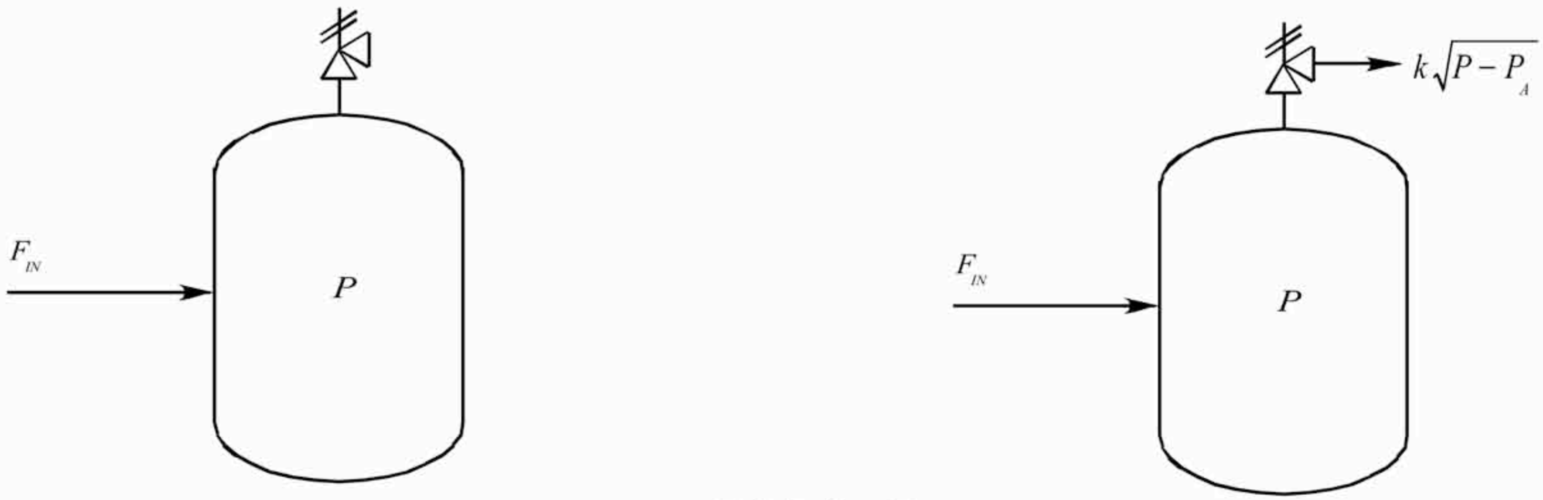
Discrete/Continuous Dynamics Simulation Results

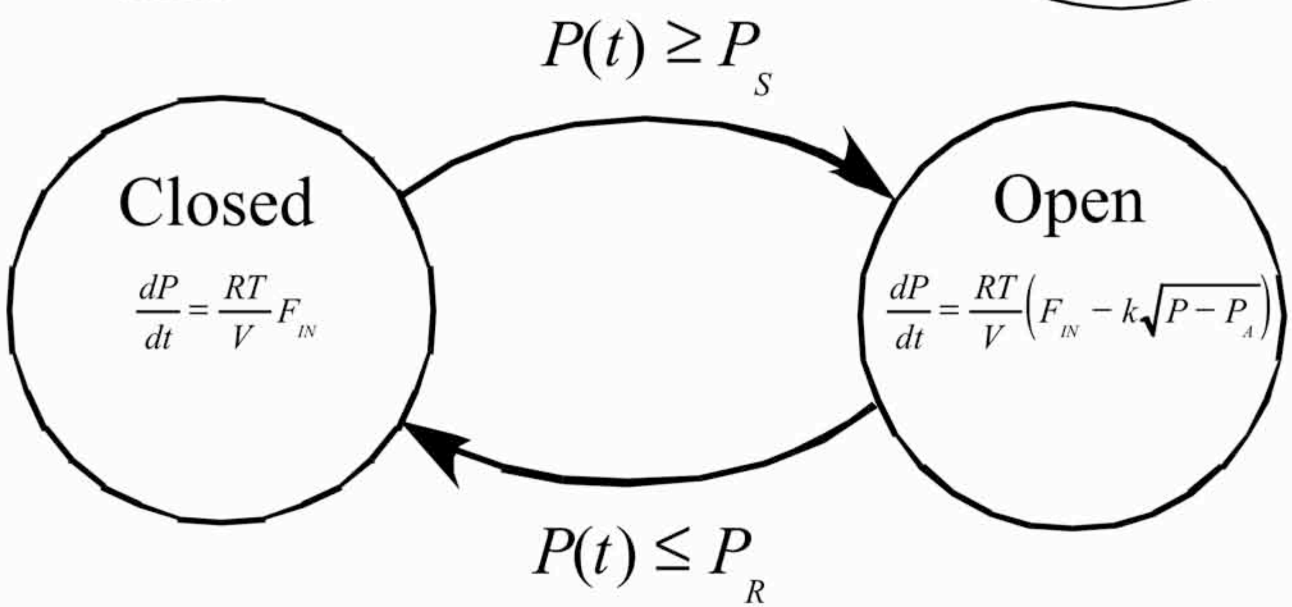


Discrete/Continuous Dynamics Summary

- Continuous time system
- Continuous state *and* discrete state dynamics
- Equations governing continuous dynamics change discretely - *switching*
- Periods of continuous evolution interrupted by *events*
  - *Time Events* - time of occurrence known explicitly in advance
    - Exogenous - timing known a priori
    - Endogeneous - timing computed from earlier event
  - *State Events* - time of occurrence determined implicitly by continuous state satisfying some condition
- At events a *transition* occurs:
  - Continuous state influences discrete state, if state event
  - Discrete state influences continuous state, e.g., by switching
- Between events:
  - Continuous state evolves according to current equations
  - Discrete state remains constant
- Sequence of events during *execution* of hybrid system implied by initial condition and parameter values

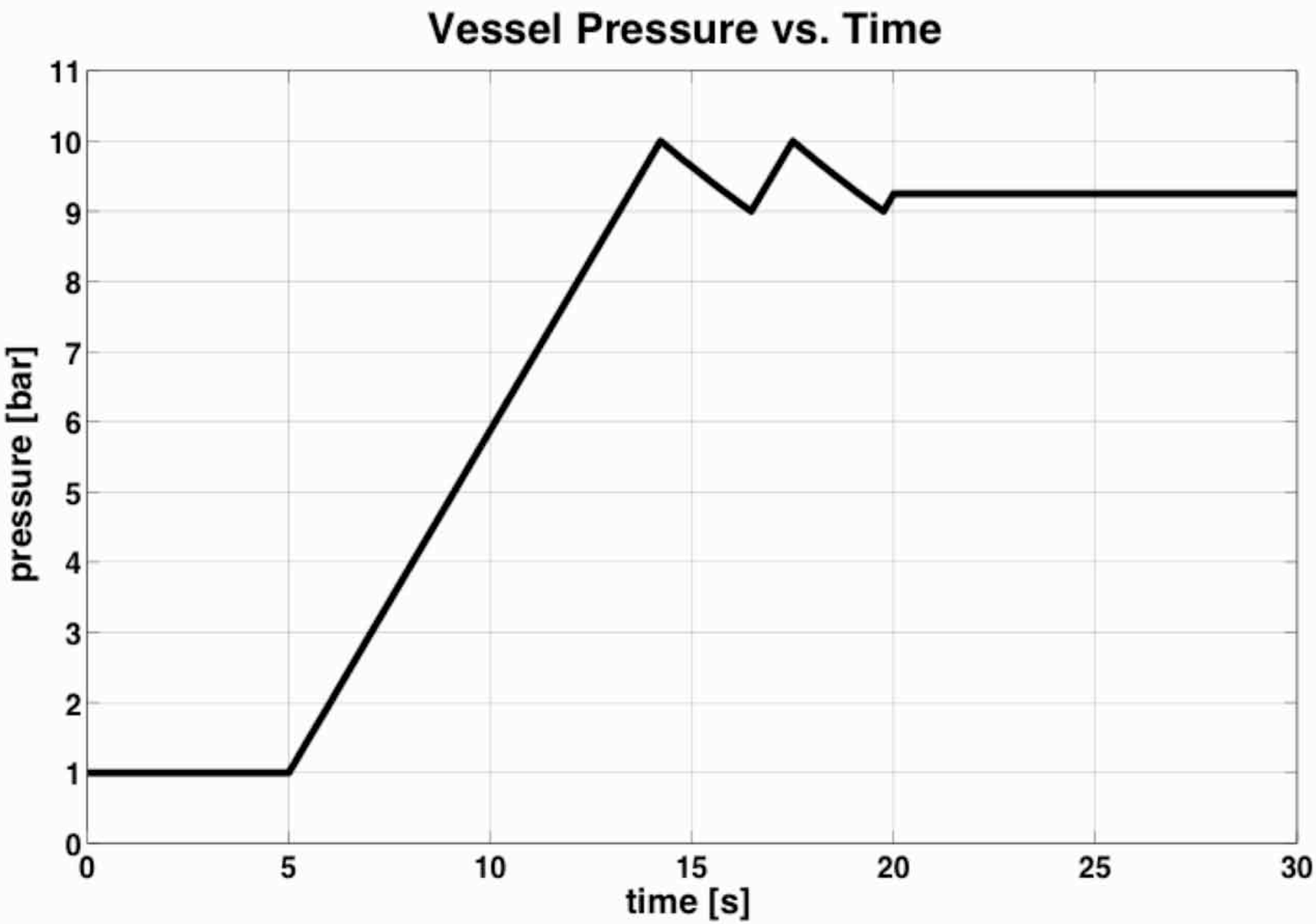
Asymmetric Transitions Vessel with Safety Relief Valve





$$P_R < P_S$$

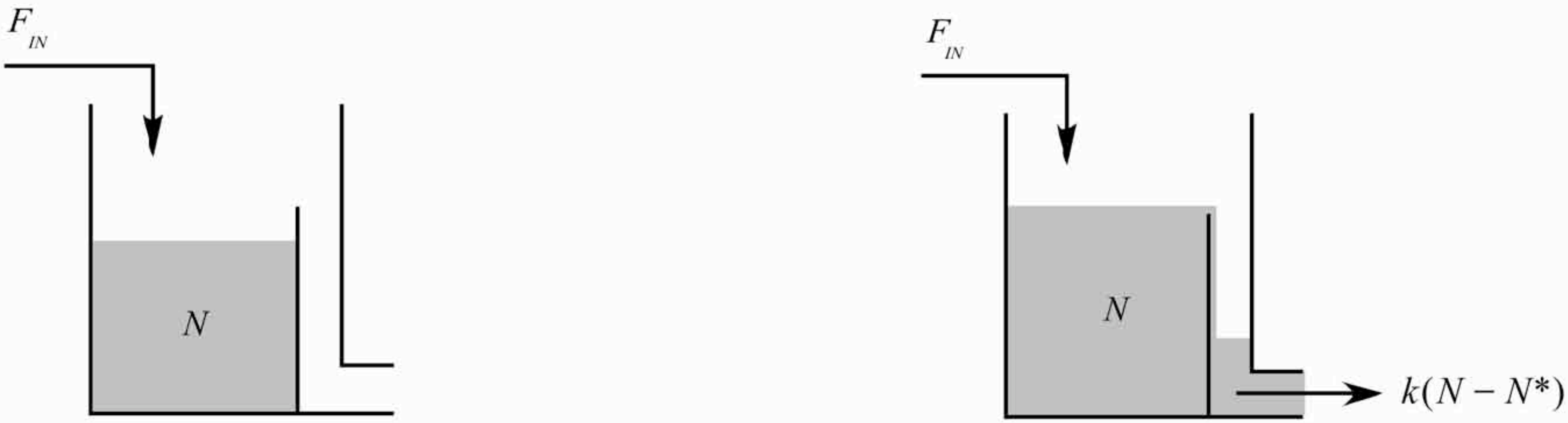
Asymmetric Transitions Vessel with Safety Relief Valve

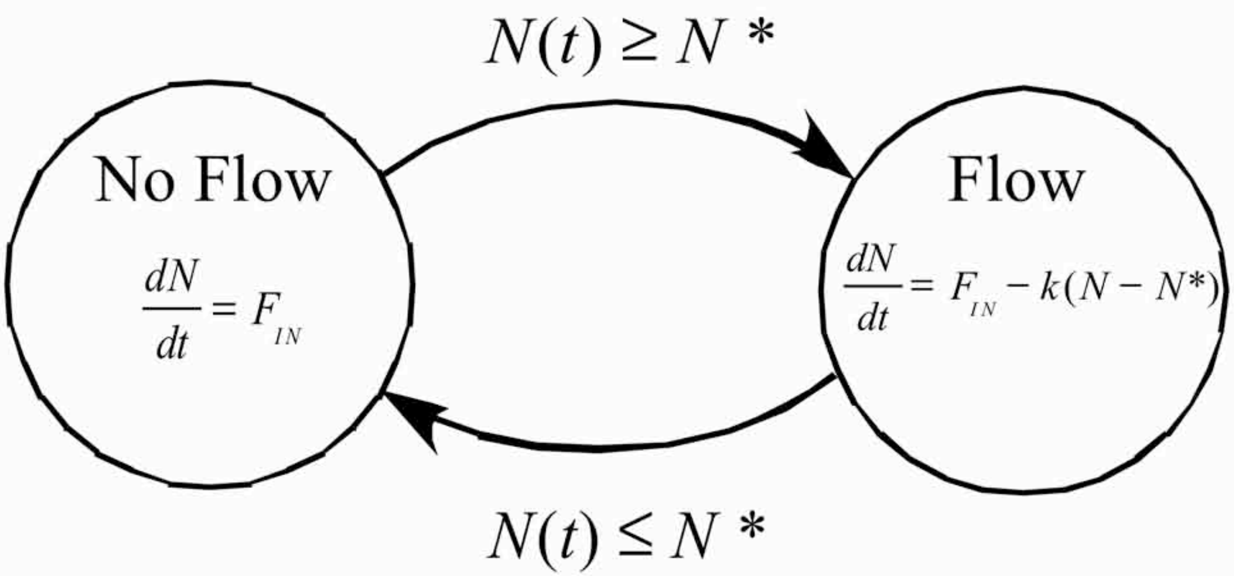


Asymmetric Transitions Vessel with Safety Relief Valve

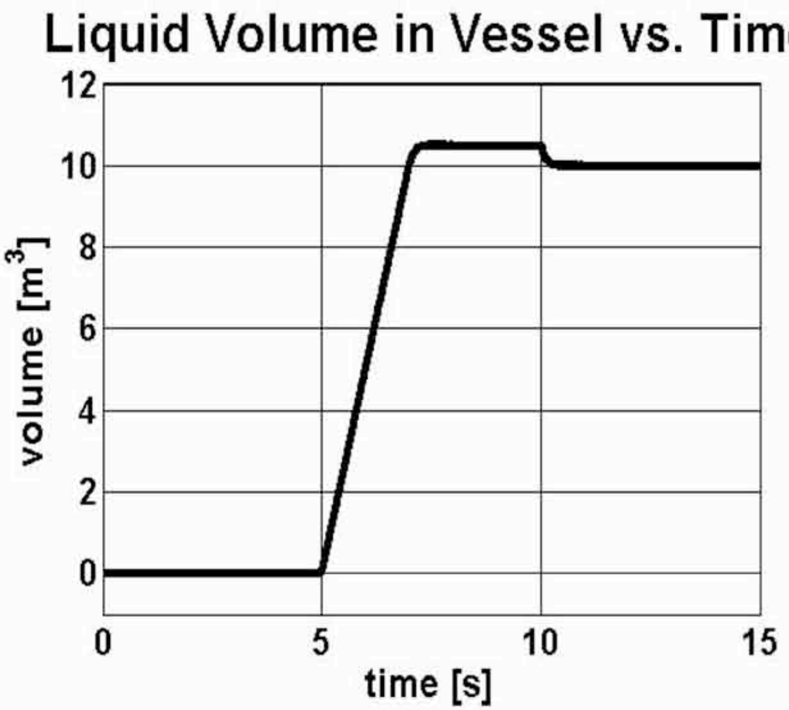
- Notion of “discrete state” distinct from “continuous state”
  - Discrete memory of previous states
- Initial condition must be specified for the discrete state
  - Discrete state cannot be inferred from continuous state
- Discrete state could be under explicit control
  - *Controlled transitions*

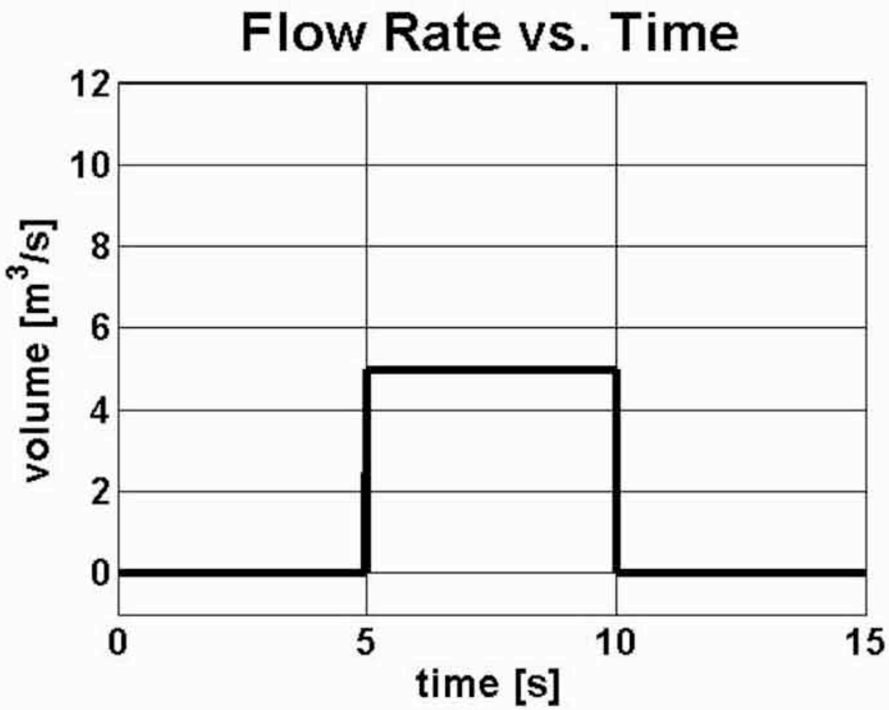
Reversible Transitions
Vessel with Overflow Weir





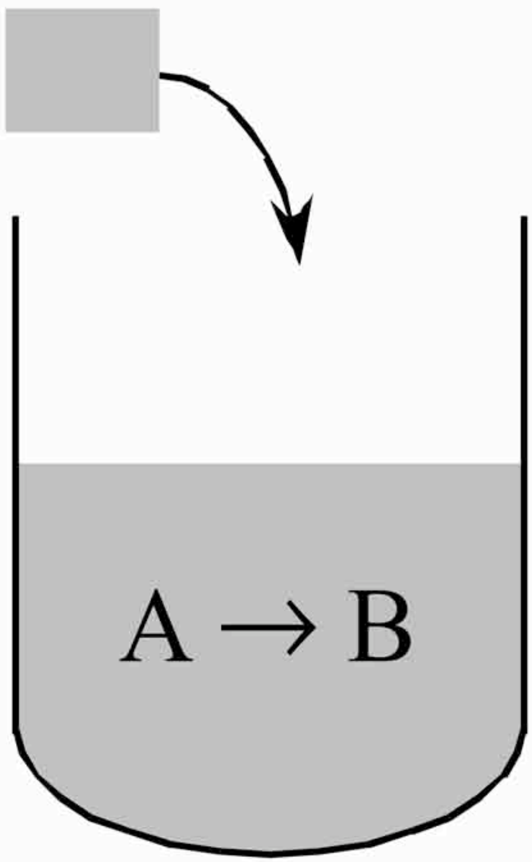
Reversible Transitions Vessel with Overflow Weir





- Now discrete state fully determined by current continuous state
- Initial discrete state can be inferred from initial condition for continuous state

Jumps
Discrete Charge to Reactor

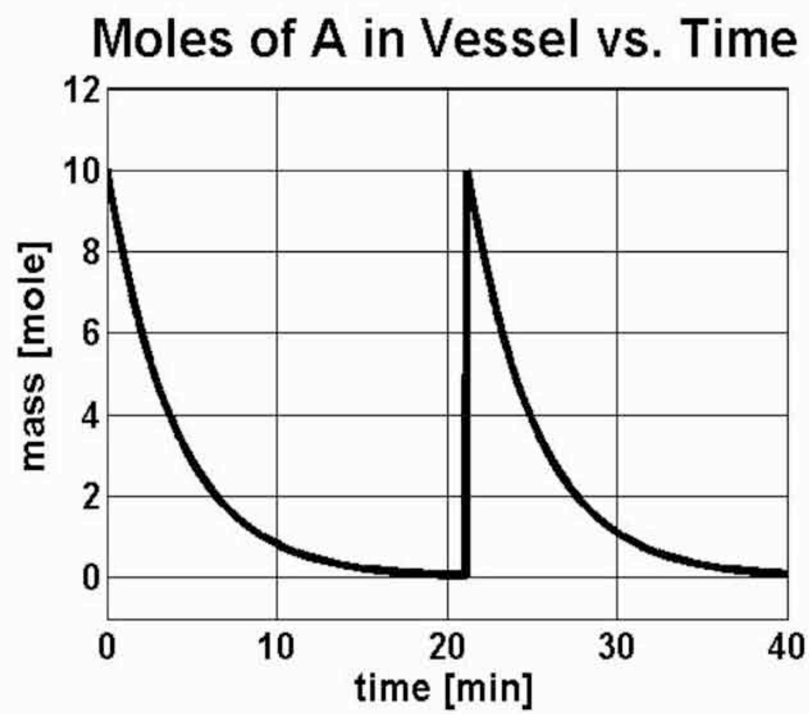


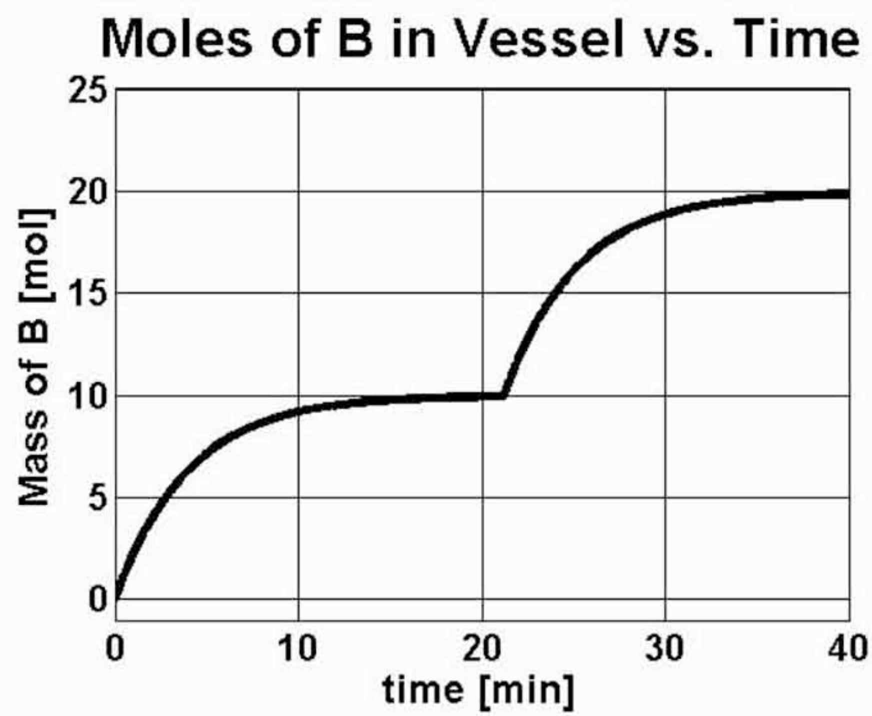
$$\frac{dN_A}{dt} = -k_A N_A$$

$$\frac{dN_B}{dt} = k_A N_A$$

$$N_A(\sigma_{i+1}) = N_A(\tau_i) + \Delta N_A$$

Jumps
Discrete Charge to Reactor



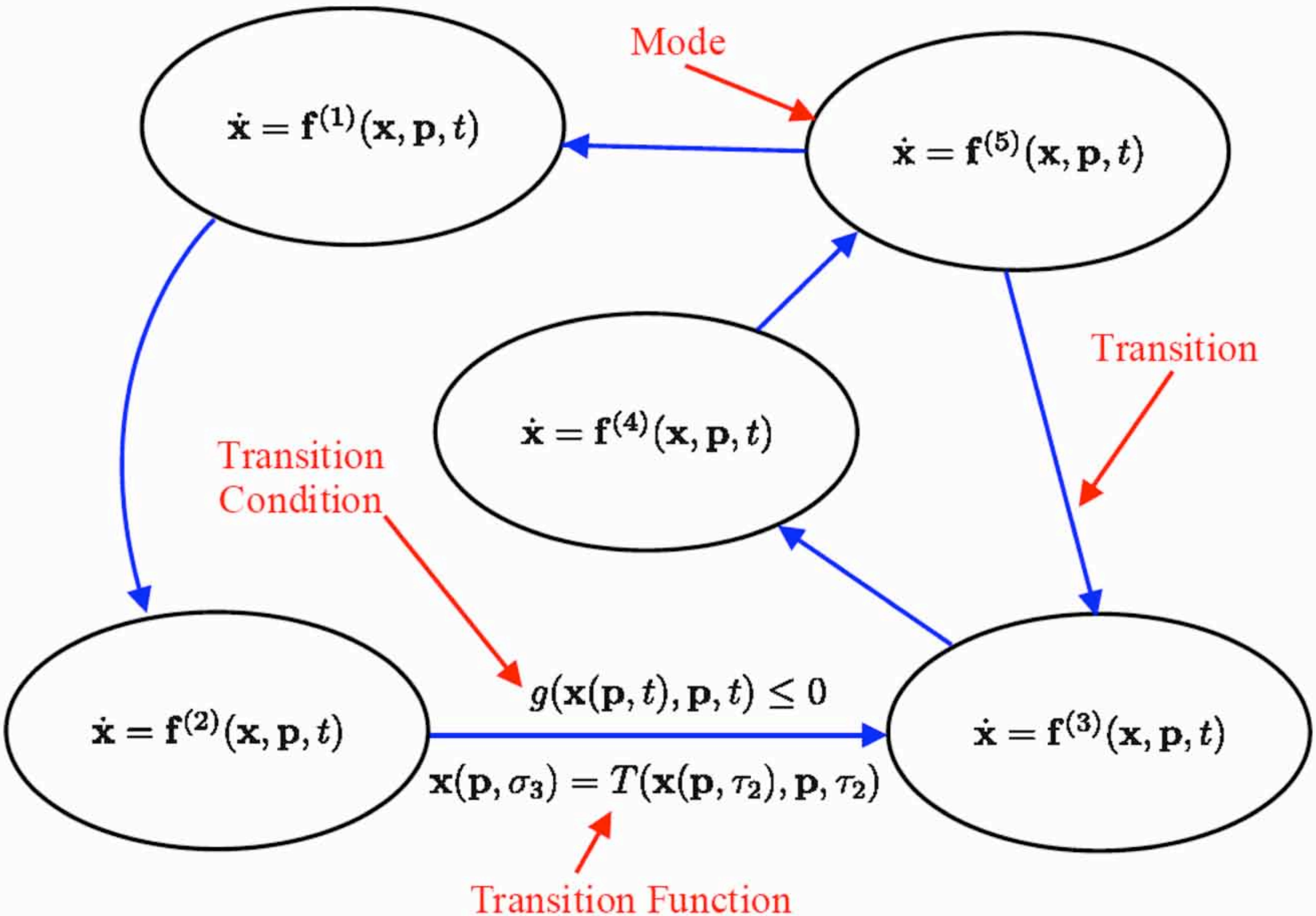


- Continuous state may not be continuous function at events
- May be due to explicit specification, or implicit property of model (“high-index phenomena”)

Hybrid Phenomena Summary

- Continuous state and discrete state
  - Discrete state may or may not be distinct from continuous state
- Implicit events as a result of continuous state reaching some condition
- Equation switching and/or continuous state jumps at events

Hybrid Automaton



Hybrid Automaton

- Set of discrete states:

$$\mathcal{M} = \{1, \ldots, n_m\}$$

Each element of this set is called a *mode* of the hybrid automaton.

- Parameters of the hybrid automaton:

$$\mathbf{p} \in P \subset \mathbb{R}^{n_p}$$

- Continuous state of the hybrid automaton:

$$\mathbf{x}(t, \mathbf{p}) \in \mathbb{R}^{n_x}$$

- Discrete state of the hybrid automaton:

$$s(t, \mathbf{p}) \in \mathcal{M}$$

Hybrid Automaton

- Each mode has system of differential equations that govern the continuous states while discrete state is in that mode, e.g.:

$$\dot{\mathbf{x}}(t,\mathbf{p}) = \mathbf{f}^{(j)}(t,\mathbf{x}(t,\mathbf{p}),\mathbf{p}), \quad \forall j \in \mathcal{M}$$

  RHS of ODEs called the *vector field* of the mode.
- Each mode has a collection pending *transitions* to other modes
  - Which modes the discrete state can switch to
  - Each transition has a *transition condition* and a *transition function*.
- Transition condition determines the time at which transition to the new mode is taken
  - Defines state or time event
  - Actual transition taken is the earliest event
- Transition function maps final state in this mode to initial state in new mode, e.g.:

$$\mathbf{x}(\sigma_{i+1},\mathbf{p}) = \mathbf{T}(\tau_i,\mathbf{x}(\tau_i,\mathbf{p}),\mathbf{p})$$

  Jumps implemented via transition functions

Extensions

- Each mode has different set of continuous state variables
- Multi-agent systems
  - Vehicles in a traffic management system
- Nondeterminism
  - Transition can be taken any time between an enabling condition and an exception condition
- Etc.

Execution of Hybrid Automaton

- Time horizon divided into contiguous *epochs*
- Each epoch *i* a closed time interval:

$$I_i = [\sigma_i, \tau_i], \quad \sigma_{i+1} = \tau_i, \quad \tau_{i+1} \geq \tau_i$$

- *Hybrid time trajectory* sequence of epochs:

$$T_\tau = \left\{ I_i \right\}_{i=1}^{n_e}$$

  Number of epochs n_e finite or infinite.
- Evolves continuously in epoch *i* if if $\tau_i > \sigma_i$, discretely otherwise

Execution of Hybrid Automaton

- Transitions taken at epoch boundaries
- Mode associated with each epoch, implying a *hybrid mode trajectory*:

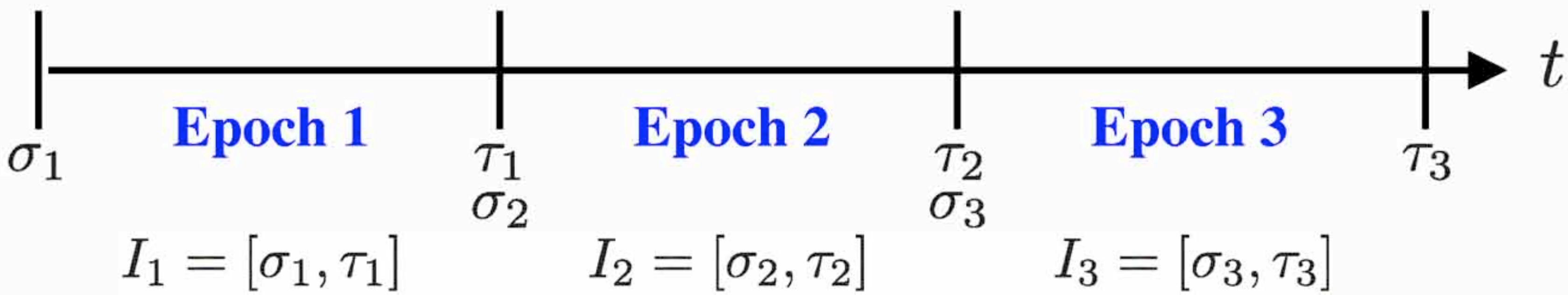
$$T_\mu = \left\{ m_i \right\}_{i=1}^{n_e}, \quad m_i \in \mathcal{M}$$

- Continuous state differentiable function of time on interior of epoch *i*, given by solution of differential equations for mode m_i
- Continuous state can be multi-valued at epoch boundaries, e.g., at jumps

$$\mathbf{x}(\sigma_{i+1}, \mathbf{p}) \neq \mathbf{x}(\tau_i, \mathbf{p})$$

  Epoch index allows to distinguish multiple values.
- An execution is *accepted* by the hybrid automaton if the transitions that are triggered coincide with the epoch boundaries

Example Execution



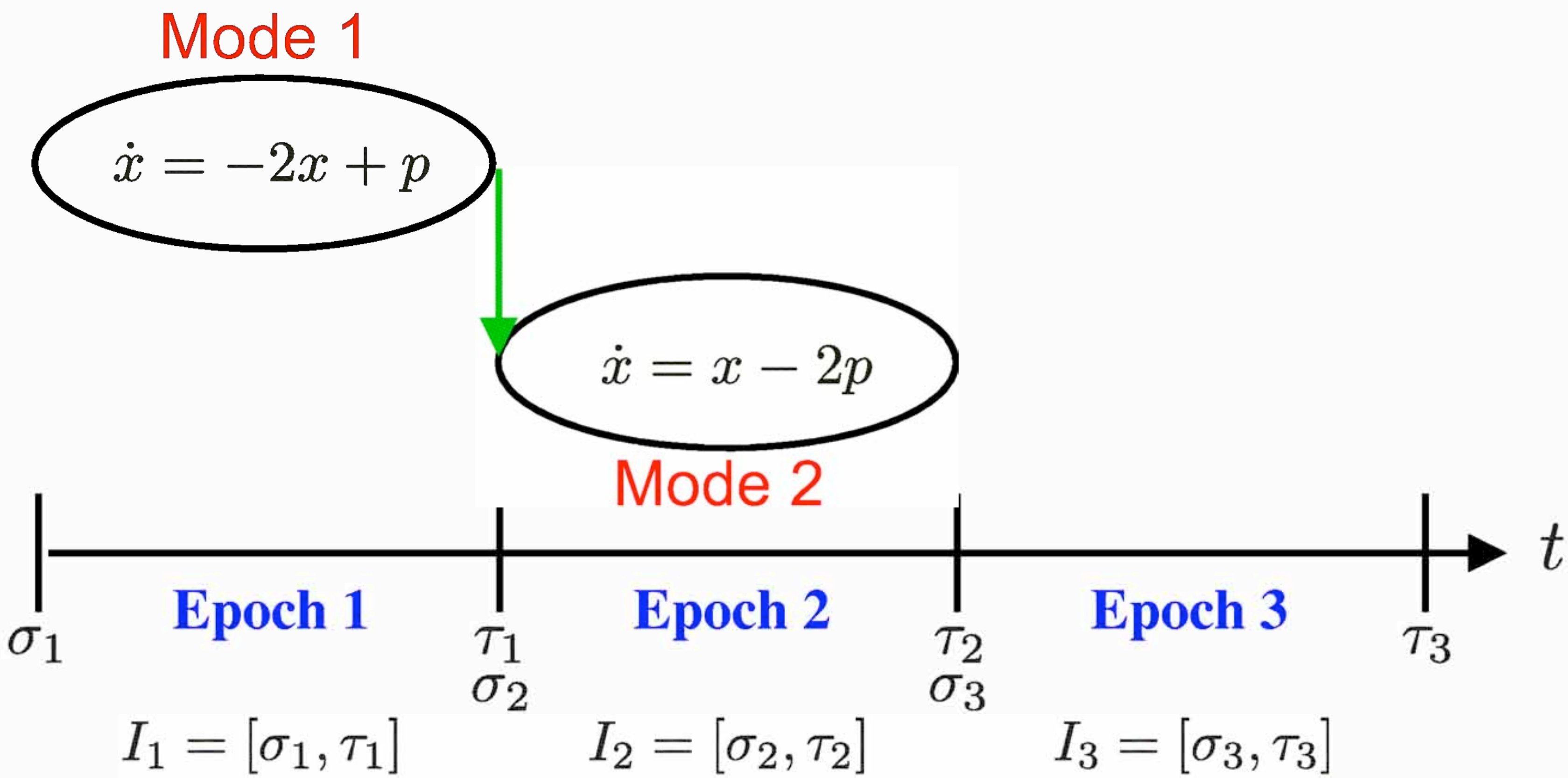
Example Execution

Mode 1

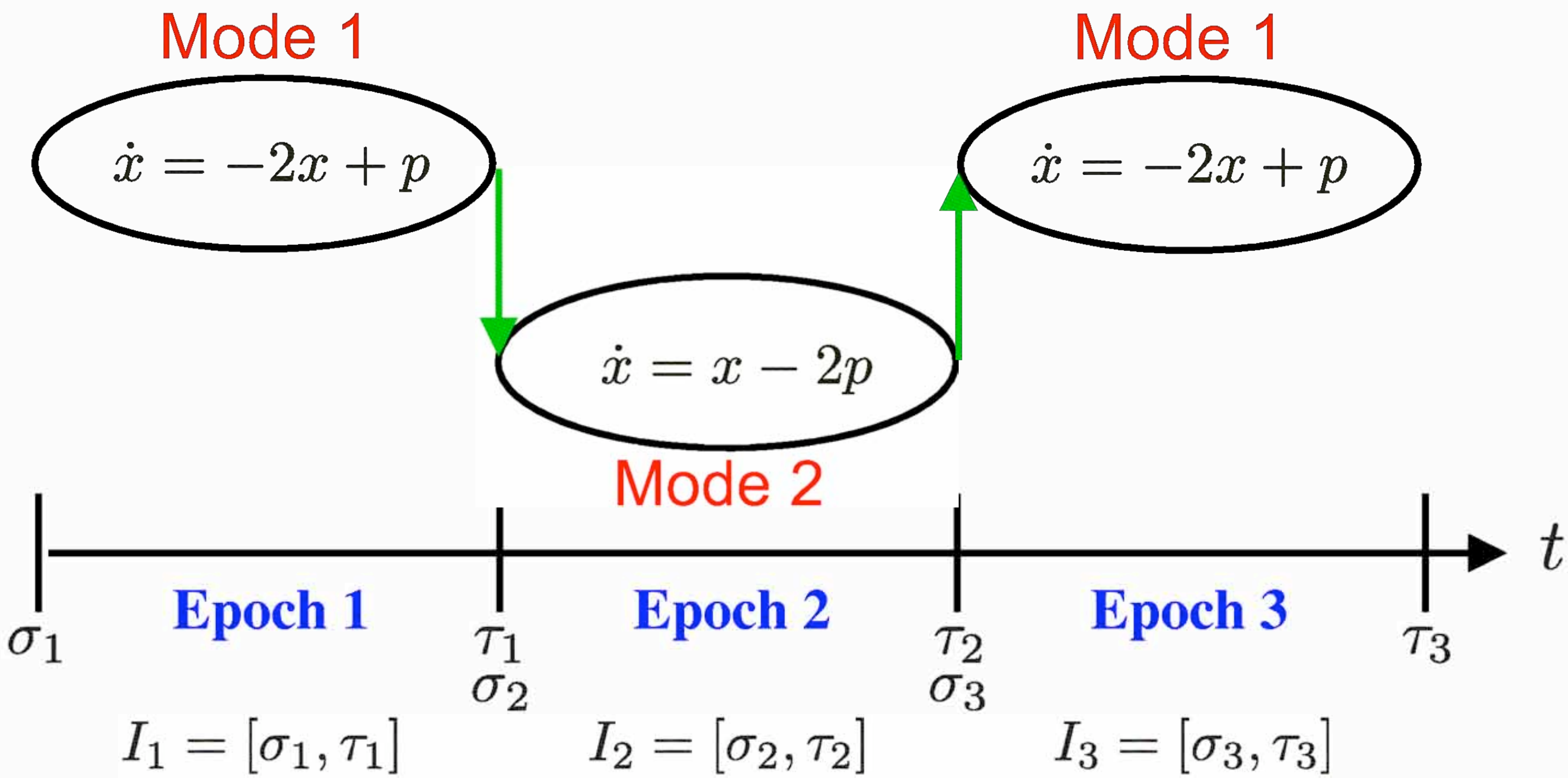
$$\dot{x} = -2x + p$$

$I_1 = [\sigma_1, \tau_1]$ $I_2 = [\sigma_2, \tau_2]$ $I_3 = [\sigma_3, \tau_3]$

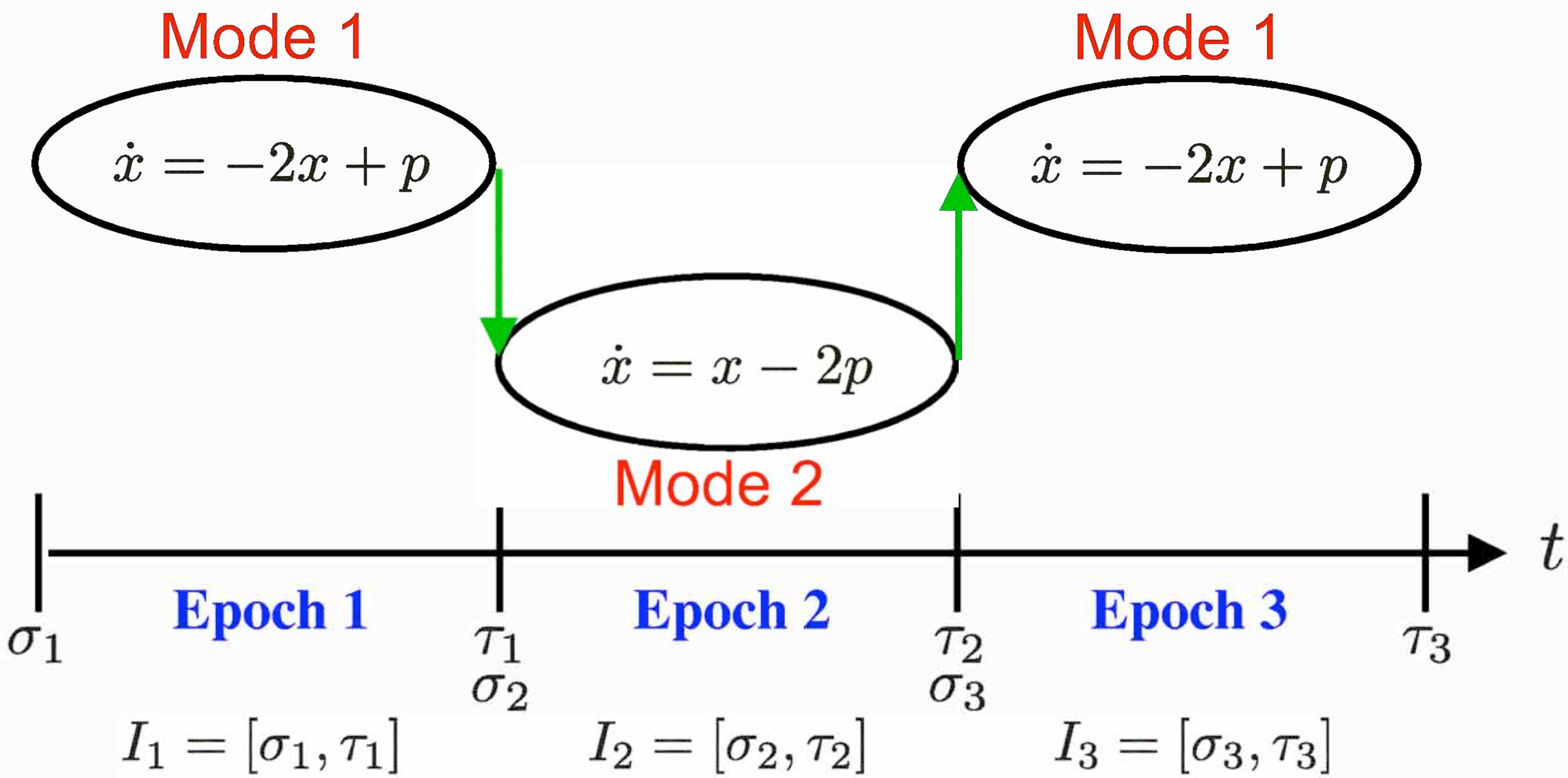
Example Execution



Example Execution

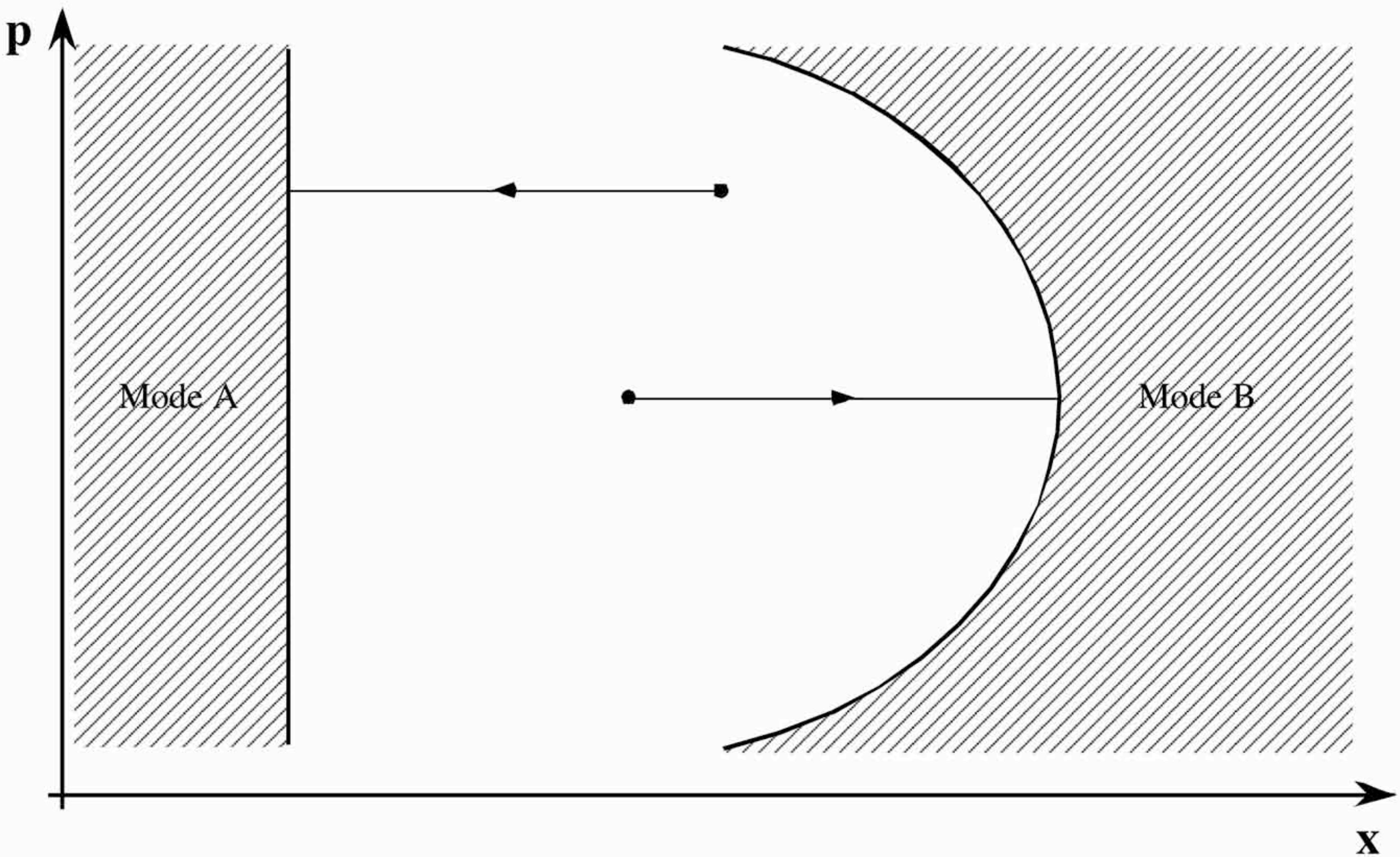


Example Execution



$$T_\mu = 1, 2, 1 \qquad T_\tau = I_1, I_2, I_3$$

Transition Conditions Departure Set



Departure Set

- If the continuous state enters the *departure set* of a mode, the corresponding transition is taken
- Next mode depends on what part of the departure set has been entered
- The departure set $G^{(j)}$ of a mode *must* be a **closed set** to have well defined event times. For example:

$$G^{(j)} \equiv \left\{ \mathbf{z} \in \mathbb{R}^{n_x} : g(\mathbf{z}) < 0 \right\}$$

  Now suppose that

$$g(\mathbf{x}(t^*)) = 0, \quad \left. \frac{\partial g}{\partial \mathbf{x}} \right|_{\mathbf{x}(t^*)} \dot{\mathbf{x}}(t^*) < 0$$

  so that the transition is not taken at t*. But,

$$g(\mathbf{x}(t^* + \varepsilon)) < 0, \quad \forall \varepsilon > 0, \text{ sufficiently small}$$

  When to take the transition, $t^*+\varepsilon$, $t^*+\varepsilon/2$, $t^*+\varepsilon/4$, … ????
- If initial continuous state in departure set of a mode, an *instantaneous transition* is taken

Discontinuity Functions

- Sets of the form:

$$G^{(j)} \equiv \left\{ (t, \mathbf{z}, \mathbf{w}) \in \mathbb{R} \times \mathbb{R}^{n_x} \times \mathbb{R}^{n_x} : g(t, \mathbf{z}, \mathbf{w}, \mathbf{p}) \le 0 \right\}$$

  and

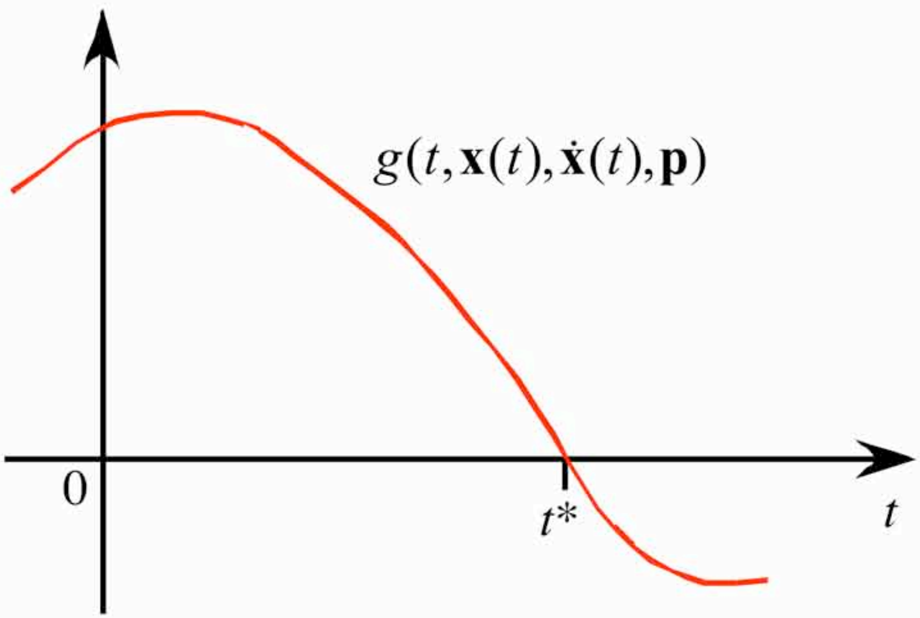
$$G^{(j)} \equiv \left\{ (t, \mathbf{z}, \mathbf{w}) \in \mathbb{R} \times \mathbb{R}^{n_x} \times \mathbb{R}^{n_x} : g(t, \mathbf{z}, \mathbf{w}, \mathbf{p}) \ge 0 \right\}$$

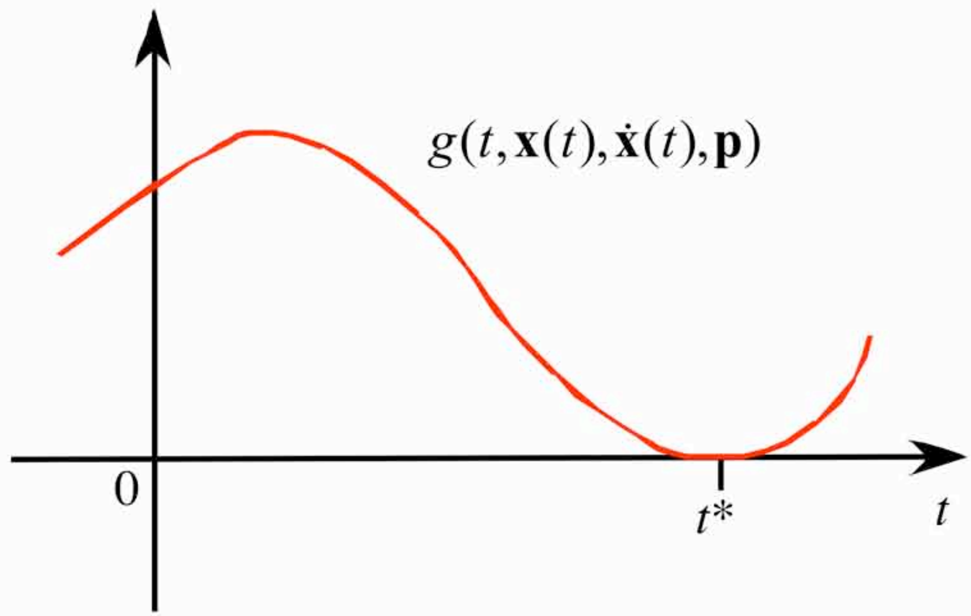
  are closed (*g* continuous, **p** fixed)
- Leads to the notion of a *discontinuity function*

$$g(t, \mathbf{x}(t), \dot{\mathbf{x}}(t), \mathbf{p})$$

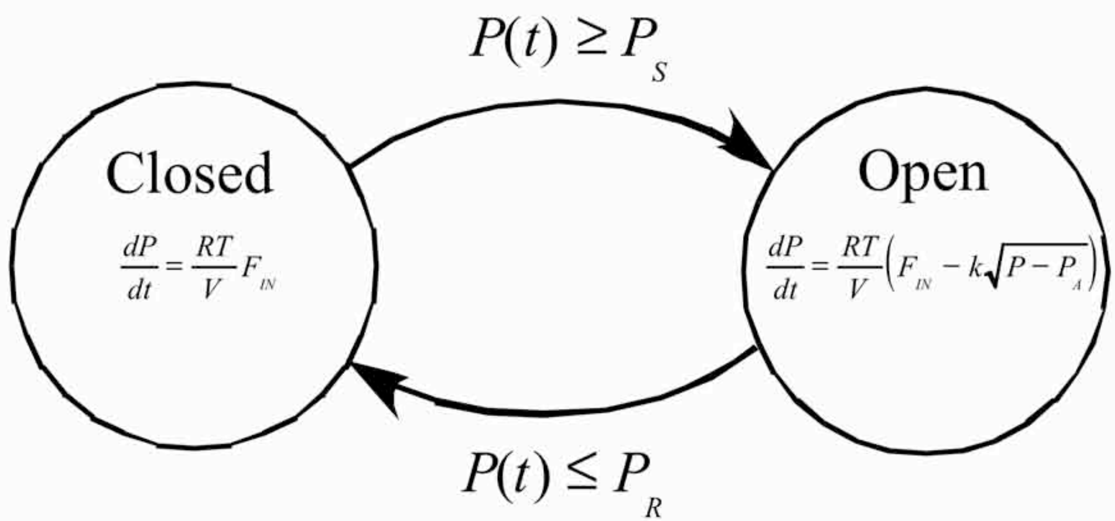
  that *reaches zero* at a transition
- Or, the discontinuity function *crosses zero* at the transition (???)

Discontinuity Functions





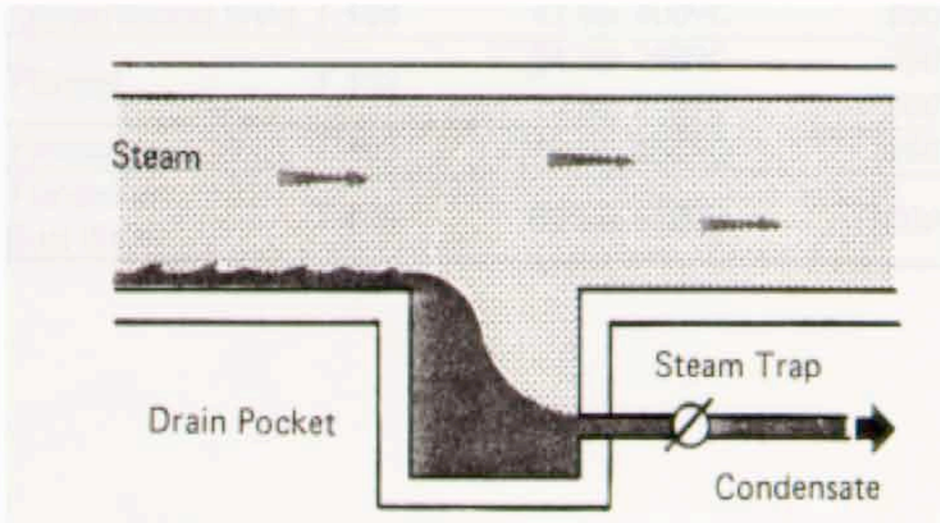
Discontinuity Functions Vessel with Safety Relief Valve

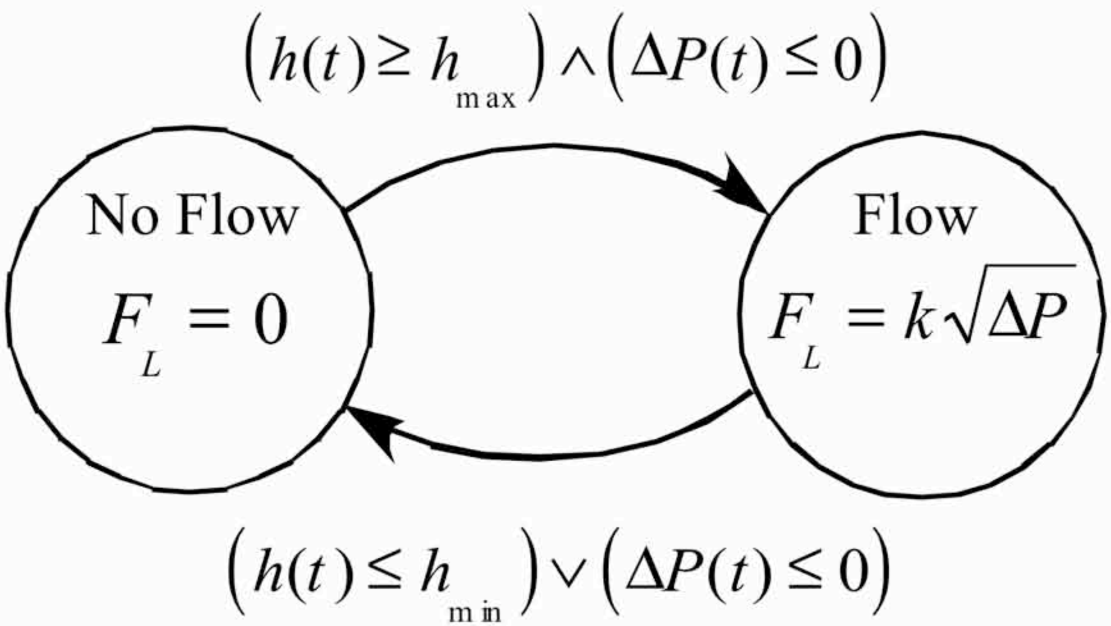


$$G^{(j)} \equiv \left\{(t, P(t)) \in \mathbb{R} \times \mathbb{R} : P_S - P(t) \leq 0\right\}$$

$$\left\{t^* : P_S - P(t^*) = 0\right\}$$

Logical Combinations of Discontinuity Functions

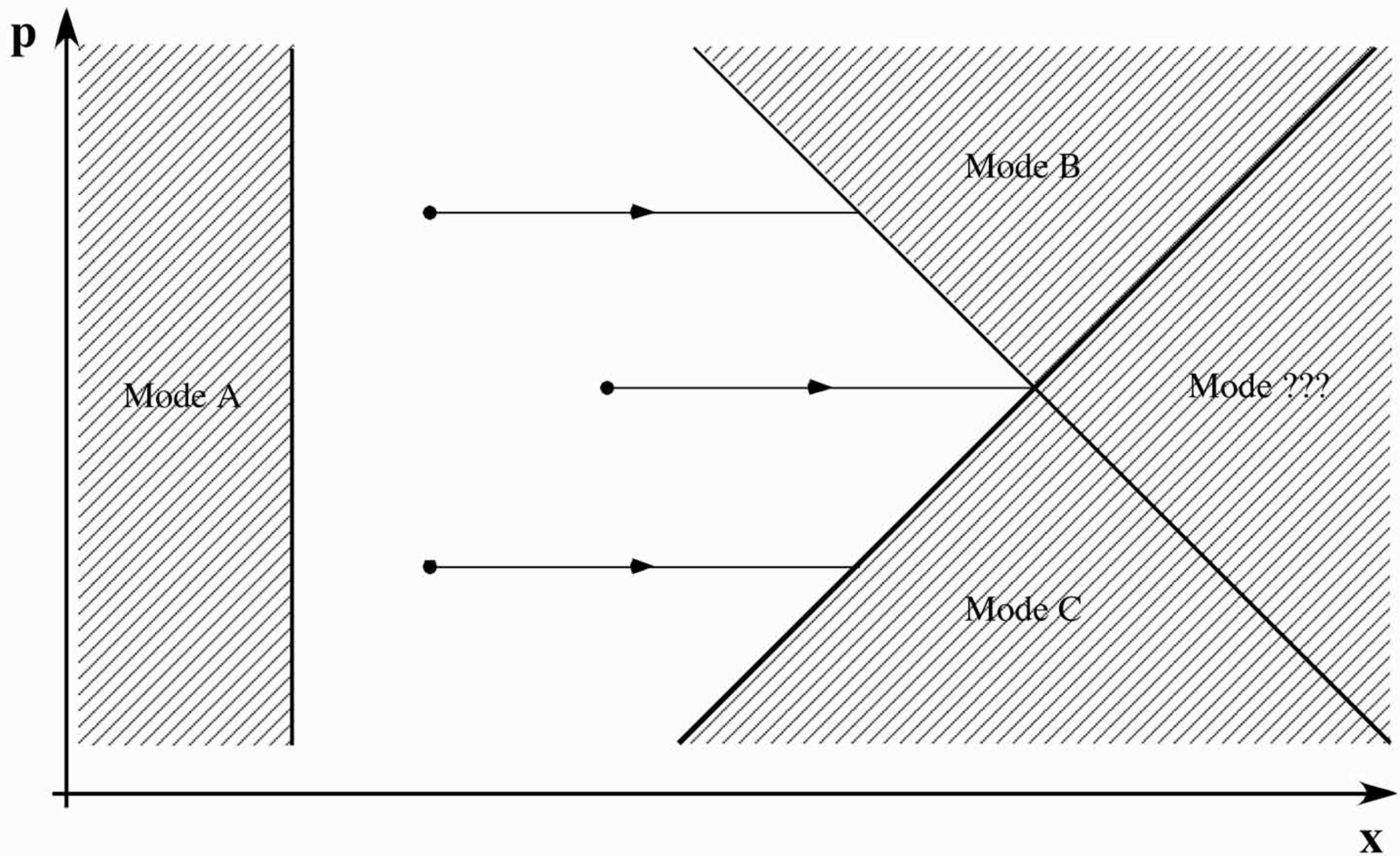




Permissible Transition Conditions

- **Basic rule**: a transition is taken earliest at the time at which a transition condition becomes true
- Transition conditions:
  - Atomic proposition involving discontinuity function and the relational operators $\leq$ and $\geq$
  - Arbitrary combinations of atomic proposition using the logical operations AND ($\wedge$) and OR ($\vee$)
  - Closed set because intersections ($\wedge$) and unions ($\vee$) of closed sets are closed sets
- Departure set closed because comprises union of all transition conditions

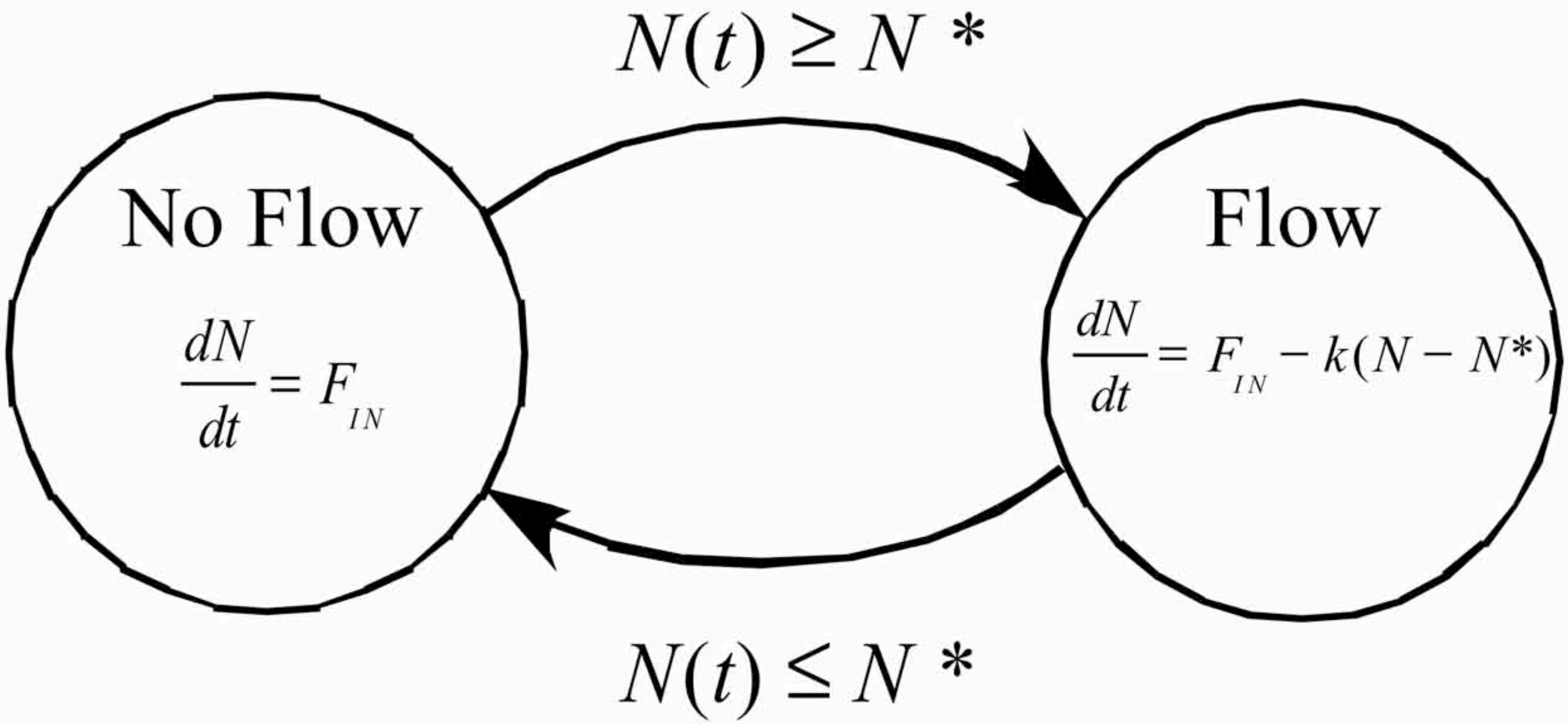
Precedence



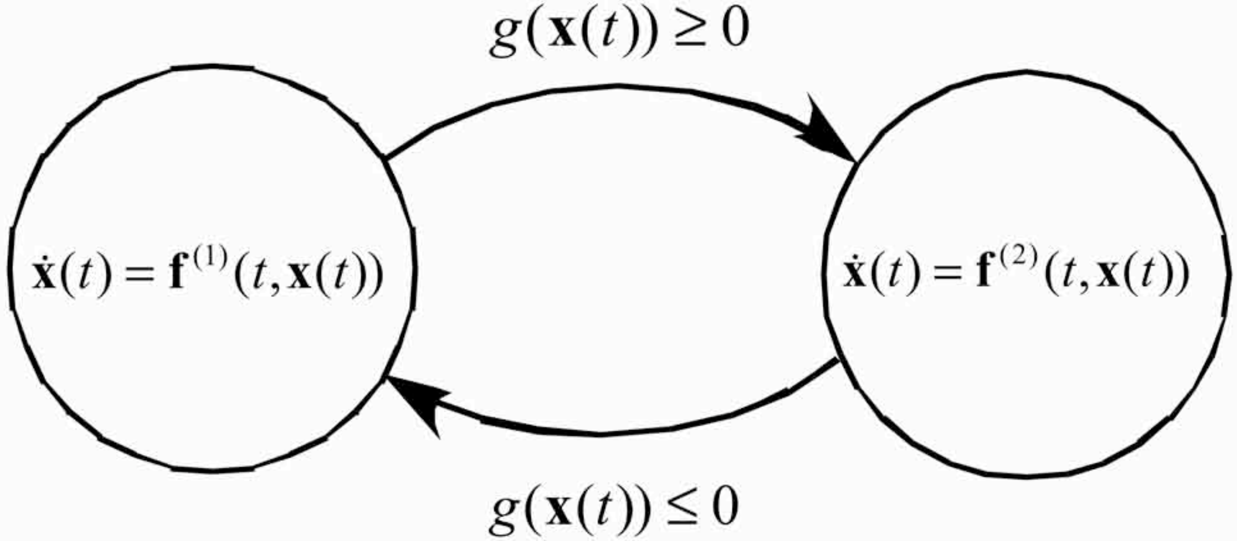
Precedence

- Example shows that departure sets for individual transition conditions can intersect
- Too restrictive to forbid intersections without relaxing rules for closed departure set
- When transition conditions intersect, explicitly define which transition takes precedence
  - For example, transition to mode B takes precedence over transition to mode C

Reversible Transitions Deadlock?



Reversible Transitions Transversality Condition

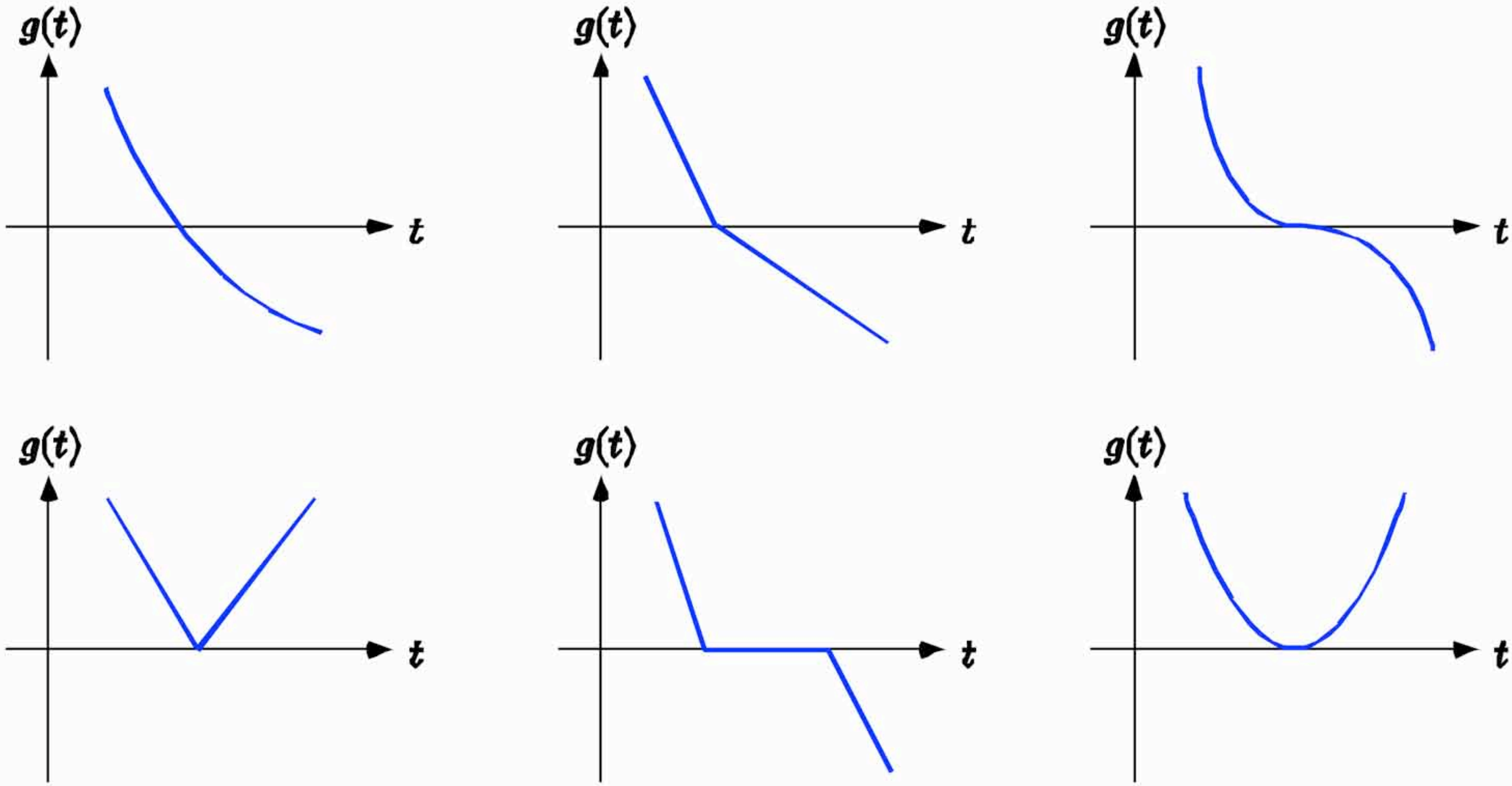


At $g(\mathbf{x}(\tau)) = 0 \quad \text{if} \quad \dot{g}(\tau^-) < 0 \wedge \dot{g}(\tau^+) < 0$

$\Rightarrow$ transition $1 \rightarrow 2$

$$\dot{g}(\tau^-) = \frac{\partial g}{\partial \mathbf{x}}(\mathbf{x}(\tau))\mathbf{f}^{(1)}(\tau, \mathbf{x}(\tau)), \quad \dot{g}(\tau^+) = \frac{\partial g}{\partial \mathbf{x}}(\mathbf{x}(\tau))\mathbf{f}^{(2)}(\tau, \mathbf{x}(\tau))$$

Reversible Transitions Exceptions



Default Transition Functions

- Is there natural or default transition function that can be inferred from differential equations of a mode?
  - Specification of a transition function overrides this default
- For ODEs natural to consider **x** a continuous function at transitions unless otherwise specified, i.e. default transition function:

$$\mathbf{x}(\sigma_{i+1},\mathbf{p}) = \mathbf{x}(\tau_i,\mathbf{p})$$

- For linear DAEs can use canonical form to derive default transition function, but **x** not necessarily continuous a default transitions
- Some arguments for nonlinear DAEs of special structure (e.g., semi-explicit index 1, etc.), but no general result for nonlinear DAEs

Complementarity Systems

$$\dot{\mathbf{x}}(t) = \mathbf{f}(t, \mathbf{x}(t), \mathbf{u}(t))$$

$$\mathbf{y}(t) = \mathbf{h}(t, \mathbf{x}(t), \mathbf{u}(t))$$

$$y_i(t)u_i(t) = 0, \quad \forall i = 1, \ldots, n_y$$

$$\mathbf{y}(t) \geq \mathbf{0}, \mathbf{u}(t) \geq \mathbf{0}$$

Schumacher, J.M., "Complementarity Systems in Optimization". Math. Prog., 101:263-295, 2004.

Dynamic Complementarity Problem (DCP)

- Behavior of complementarity system governed by existence and uniqueness of the solution of the DCP
- At each time, the DCP is a nonlinear complementarity problem parametric in the value of the continuous states $\mathbf{x}(t)$:

$$\mathbf{y}(t) = \mathbf{h}(t, \mathbf{x}(t), \mathbf{u}(t))$$

$$y_i(t)u_i(t) = 0, \quad \forall i = 1,\ldots,n_y$$

$$\mathbf{y}(t) \geq \mathbf{0}, \mathbf{u}(t) \geq \mathbf{0}$$

  Need for a solution of this complementarity problem to exist at all times imposes *inequality path constraints* on the continuous states $\mathbf{x}(t)$.

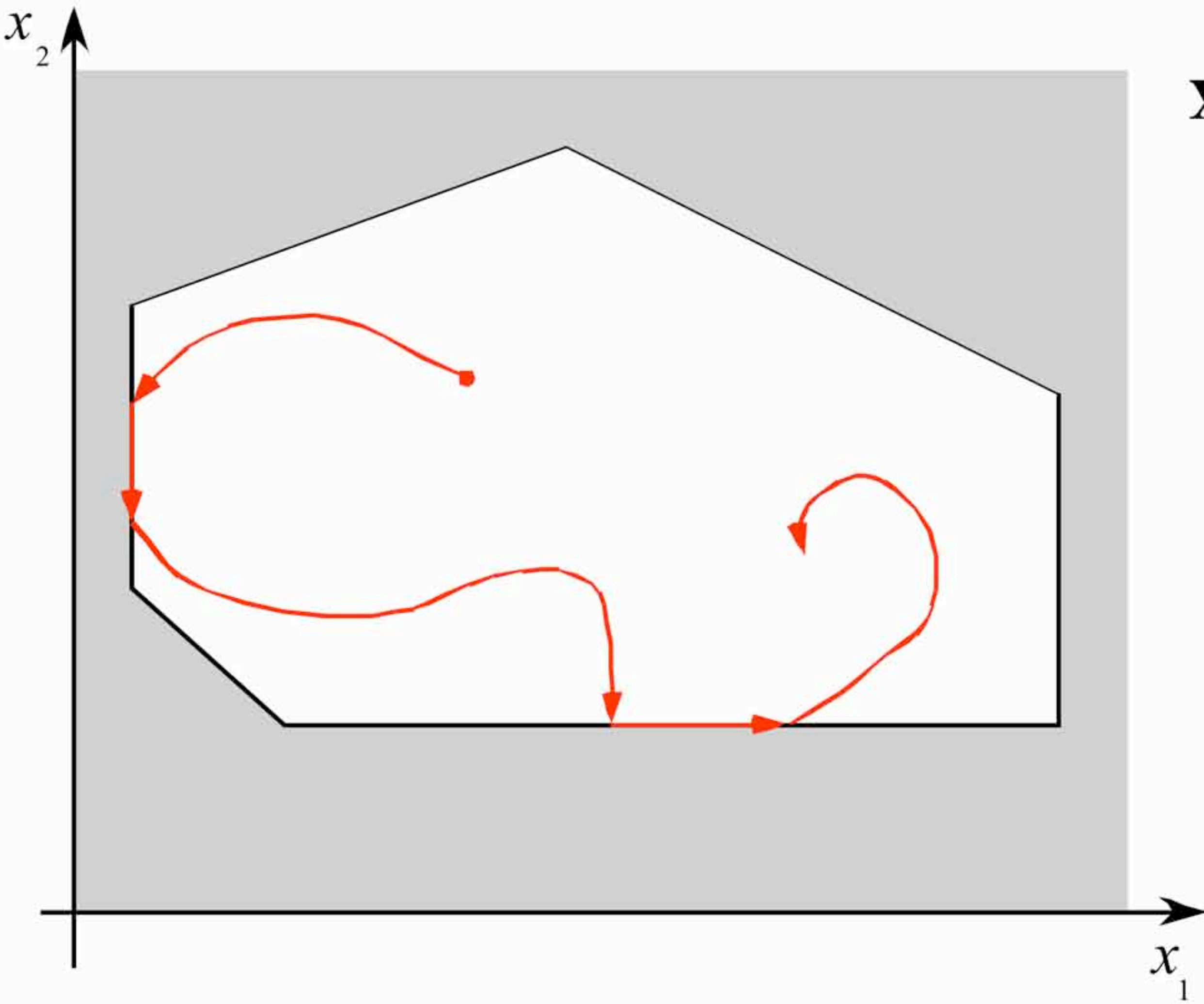
- Also question of uniqueness of solution of the complementarity problem at a given time, e.g. when $h(x(t))=0$ in:

$$y(t) = h(x(t))$$

$$y(t)u(t) = 0$$

$$y(t) \geq 0, u(t) \geq 0$$

Complementarity Systems



$$\dot{\mathbf{x}}(t) = \mathbf{f}(t, \mathbf{x}(t), \mathbf{u}(t))$$

$$\mathbf{h}(\mathbf{x}(t)) \geq \mathbf{0}$$

$$\mathbf{u}(t) \geq \mathbf{0}$$

- Each path constraint associated with a unique "control" u_i
- "Control" u_i zero when continuous state not on path constraint
- Need to stay on path constraint prescribes unique value for control

Complementarity Systems

- Notion of index set for active constraints:

$$\alpha(t) = \{ i : y_i(t) = 0 \}$$

$$u_i(t) = 0, \quad \forall i \notin \alpha(t)$$

- "Mode" of complementarity system for each possible set of active constraints:

$$\dot{\mathbf{x}}(t) = \mathbf{f}(t, \mathbf{x}(t), \mathbf{u}(t))$$

$$\mathbf{y}(t) = \mathbf{h}(t, \mathbf{x}(t), \mathbf{u}(t))$$

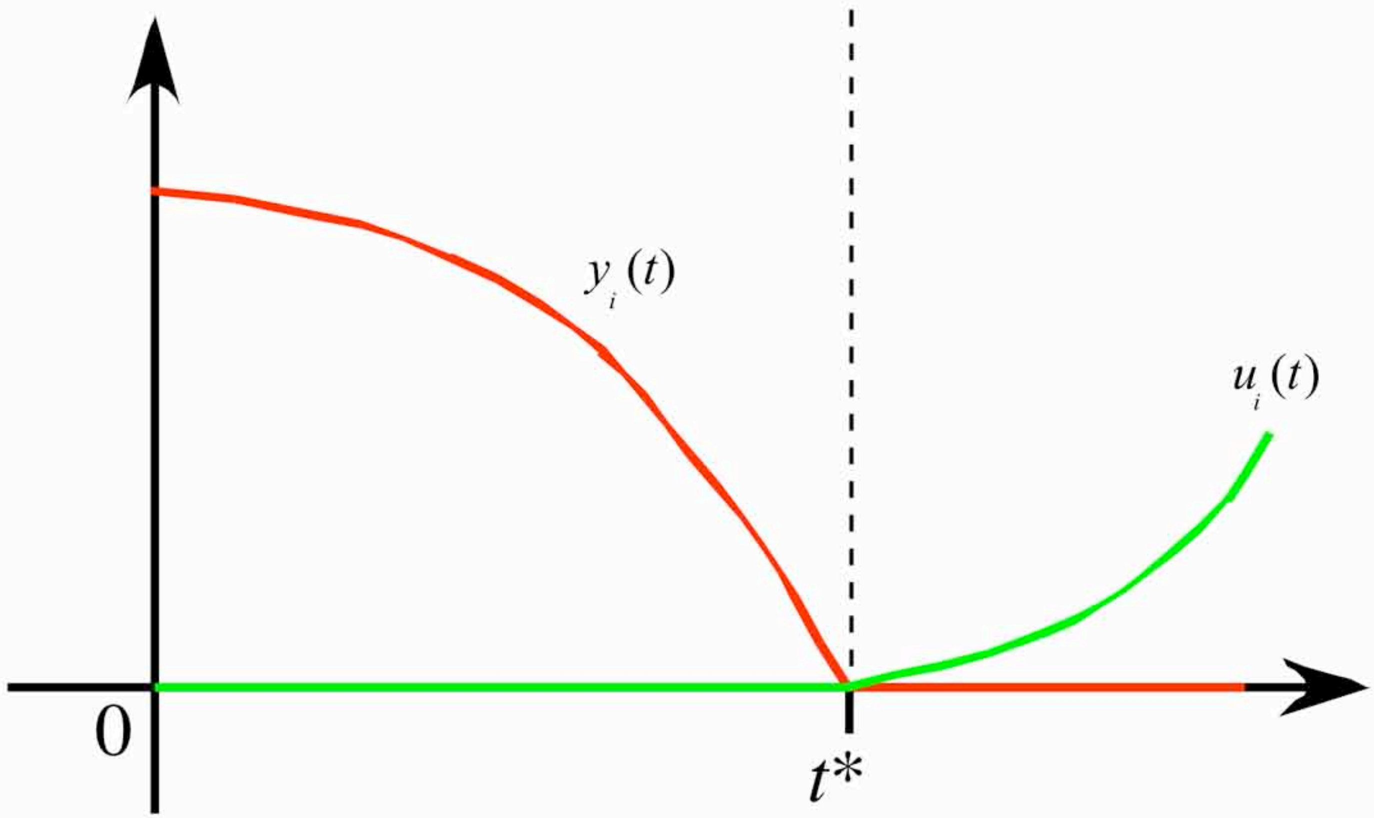
$$y_i(t) = 0, \quad \forall i \in \alpha$$

$$u_i(t) = 0, \quad \forall i \notin \alpha$$

- Discrete state fully defined by continuous state $\mathbf{x}(t)$
  - Discrete state = what path constraints currently active

Complementarity Systems

- Switch of mode (active index set) occurs when continuation within a given mode would violate nonnegativity constraints on free $y_i(t)$ and $u_i(t)$
  - Detect positive variable reaching zero



Complementarity Systems High-Index Phenomena

$$
\begin{aligned}
\dot{x}(t) &= f(x(t)) + g(x(t))u(t) \\
y(t) &= h(x(t)) \\
y(t)u(t) &= 0 \\
y(t) \geq 0&, u(t) \geq 0
\end{aligned}
$$

$$
\begin{aligned}
\dot{x}(t) &= f(x(t)) + g(x(t))u(t) \\
y(t) &= h(x(t)) \\
u(t) &= 0
\end{aligned}
$$

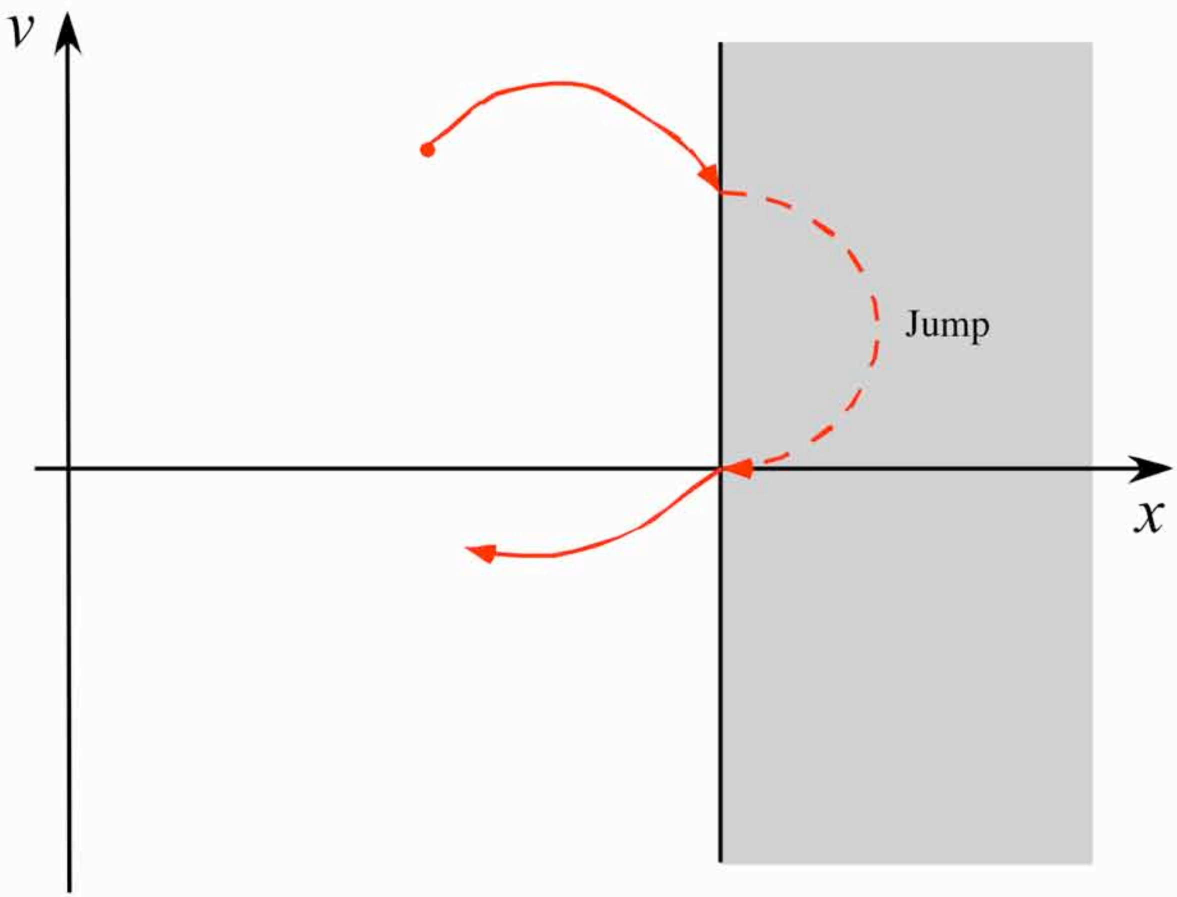
Mode 1: index = 1

$$
\begin{aligned}
\dot{x}(t) &= f(x(t)) + g(x(t))u(t) \\
0 &= h(x(t)) \\
y(t) &= 0
\end{aligned}
$$

Mode 2: index = 2

Complementarity Systems High-Index Phenomena

- No explicit representation of jumps, *but* jumps may occur due to "high-index" phenomena



Complementarity Systems Existence and Uniqueness

- Sufficient existence and uniqueness conditions known for solutions **x**(t) that are continuous in time for

$$\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t)) + \sum_{i=1}^{n_y} g_i(\mathbf{x}(t))u_i(t)$$

$$\mathbf{y}(t) = \mathbf{h}(\mathbf{x}(t))$$

$$y_i(t)u_i(t) = 0, \quad \forall i = 1,\ldots,n_y$$

$$\mathbf{y}(t) \geq \mathbf{0}, \mathbf{u}(t) \geq \mathbf{0}$$

  Index of modes can be arbitrary
- In general, if index 3+ modes, then jumps can occur and solutions **x**(t) not continuous in time
- Again, a general rule for such jumps is elusive
  - For linear complementarity systems general rule from linear DAE theory
  - Natural rules for inelastic and frictionless collisions in multi-body dynamics
  - Or, interpret as lack of uniqueness of solutions even in linear case - e.g., bouncing ball

Complementarity Systems Generalization

$$\dot{\mathbf{x}}(t) = \mathbf{f}(t, \mathbf{x}(t), \mathbf{u}(t))$$

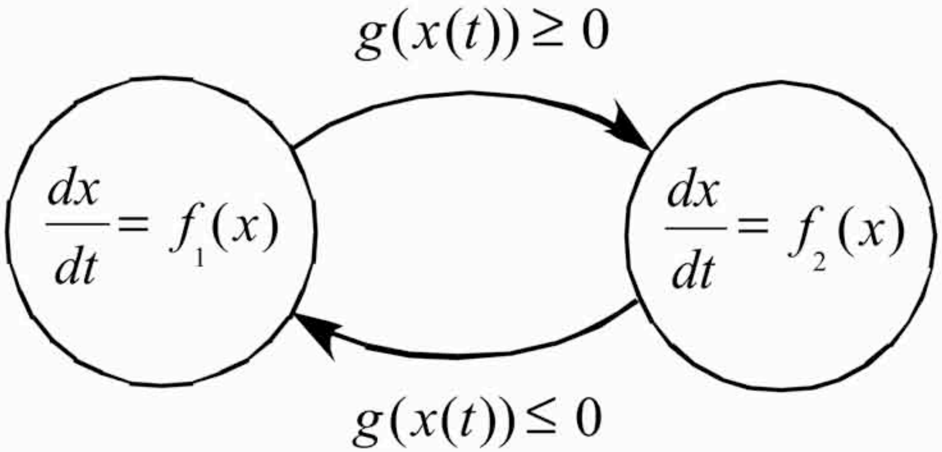
$$\mathbf{h}(t, \mathbf{x}(t), \mathbf{u}(t), \mathbf{y}(t), \mathbf{z}(t)) = \mathbf{0}$$

$$y_i(t) u_i(t) = 0, \quad \forall i = 1, \ldots, n_y$$

$$\mathbf{y}(t) \geq \mathbf{0}, \mathbf{u}(t) \geq \mathbf{0}$$

- **z** additional algebraic variables not required to complementary

Complementarity Systems Reversible Transitions



$$
\begin{aligned}
\dot{x}(t) &= u_1(t) f_1(x(t)) + u_2(t) f_2(x(t)) \\
u_1(t) + u_2(t) &= 1 \\
y_1(t) - y_2(t) &= g(x(t)) \\
y_1(t) u_1(t) &= 0 \\
y_2(t) u_2(t) &= 0 \\
y_1(t) \geq 0, u_1(t) \geq 0,\ & y_2(t) \geq 0, u_2(t) \geq 0
\end{aligned}
$$

$$
g(x(t)) < 0 \quad \Rightarrow \quad
\begin{aligned}
\dot{x}(t) &= f_1(x(t)) \\
u_1(t) &= 1 \\
-y_2(t) &= g(x(t)) \\
u_2(t) &= 0 \\
y_1(t) &= 0
\end{aligned}
$$

$$
g(x(t)) > 0 \quad \Rightarrow \quad
\begin{aligned}
\dot{x}(t) &= f_2(x(t)) \\
u_2(t) &= 1 \\
y_1(t) &= g(x(t)) \\
u_1(t) &= 0 \\
y_2(t) &= 0
\end{aligned}
$$

$$
g(x(t)) = 0 \quad \Rightarrow \quad
\begin{aligned}
\dot{x}(t) &= u_1(t) f_1(x(t)) + u_2(t) f_2(x(t)) \\
u_1(t) + u_2(t) &= 1 \\
y_1(t) &= y_2(t) \\
y_1(t) &= 0 \\
u_1(t) \geq 0,\ & u_2(t) \geq 0
\end{aligned}
$$

Complementarity Systems Reversible Transitions

- Question of *uniqueness* of solution of DCP arises when $g(x(t))=0$ but no path constraint active…
- If vector fields continuous on boundary between modes:

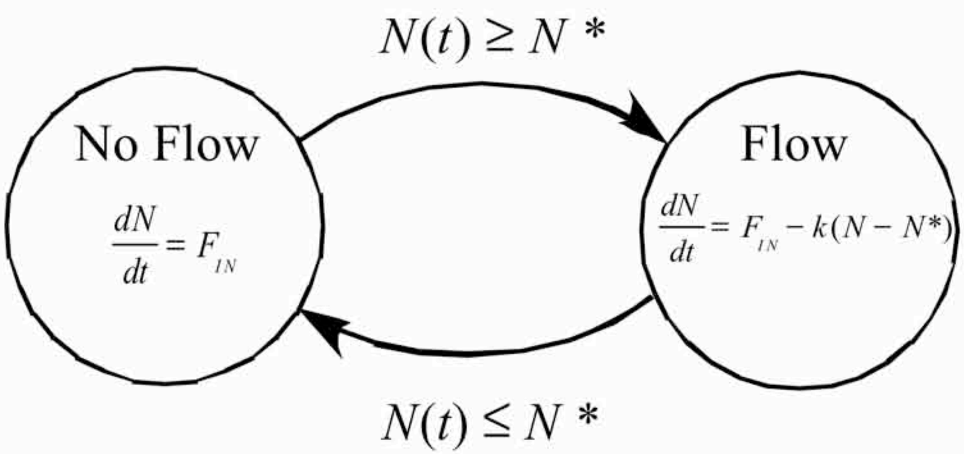
$$\forall z \in \mathbb{R} : g(z) = 0,\ f_1(z) = f_2(z)$$

Then

$$u_1(t) f_1(x(t)) + u_2(t) f_2(x(t))$$

Takes the same value for all feasible values for $u_1(t)$ and $u_2(t)$, so that $\mathbf{x}(t)$ is still unique
  - Manifestation of classical existence and uniqueness result for ODEs
- Even without continuity on boundaries, existence and uniqueness of solution $\mathbf{x}(t)$ can often still be argued
- Even for so-called *sliding mode*
- In general, does a unique $\mathbf{x}(t)$ exist for a given complementarity system?

Complementarity Systems Vessel with Overflow Weir



$$\dot{N}(t) = F_{IN}(t) - u_2(t)k(N(t) - N^*)$$
$$u_1(t) + u_2(t) = 1$$
$$y_1(t) - y_2(t) = N(t) - N^*$$
$$y_1(t)u_1(t) = 0$$
$$y_2(t)u_2(t) = 0$$
$$y_1(t) \geq 0,\ u_1(t) \geq 0,\ y_2(t) \geq 0,\ u_2(t) \geq 0$$

$N(t) - N^* < 0 \quad \Rightarrow$

$$\dot{N}(t) = F_{IN}(t)$$
$$u_1(t) = 1$$
$$-y_2(t) = N(t) - N^*$$
$$u_2(t) = 0$$
$$y_1(t) = 0$$

$N(t) - N^* > 0 \quad \Rightarrow$

$$\dot{N}(t) = F_{IN}(t) - k(N(t) - N^*)$$
$$u_2(t) = 1$$
$$y_1(t) = N(t) - N^*$$
$$u_1(t) = 0$$
$$y_2(t) = 0$$

$N(t) - N^* = 0 \quad \Rightarrow$

$$\dot{N}(t) = F_{IN}(t)$$
$$u_1(t) + u_2(t) = 1$$
$$y_1(t) = y_2(t)$$
$$y_1(t) = 0$$
$$u_1(t) \geq 0,\ u_2(t) \geq 0$$

Complementarity Systems Partition of State Space

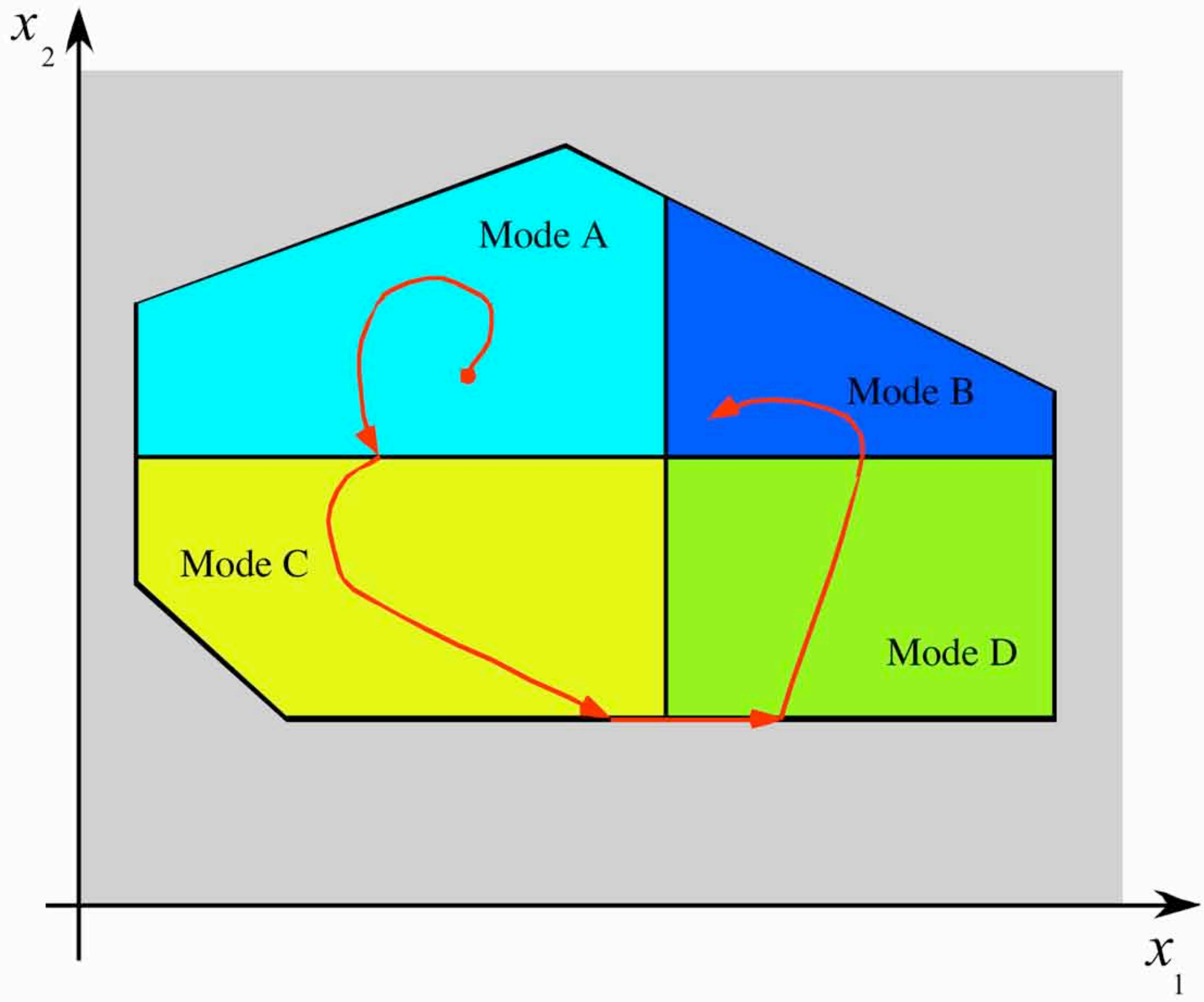
- Let X be the feasible subset of state space
  - As prescribed by existence of solution to DCP
- A *partition* of this feasible subset into K subsets is defined as:

$$X = \bigcup_{i=1}^{K} X_i$$

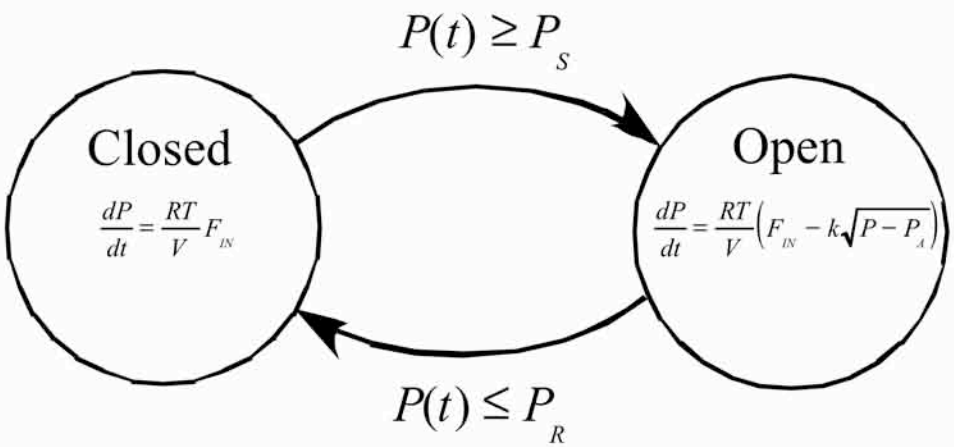
$$\operatorname{int} X_i \cap \operatorname{int} X_j, \quad \forall i \neq j$$

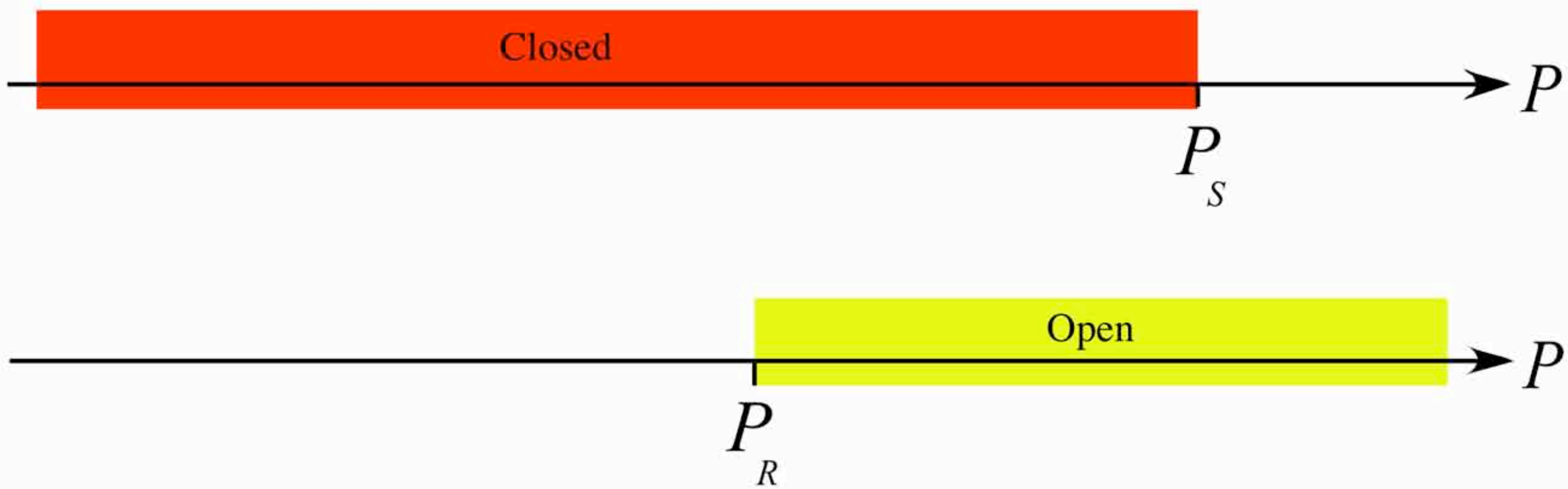
- Complementarity systems can be used to define a partition of X with a mode associated with the interior of each subset
- Only complication is to define dynamics on the boundaries between the modes where solution of DCP is nonunique
- Continuous vector fields on boundaries simple case

Complementarity Systems Partition of State Space



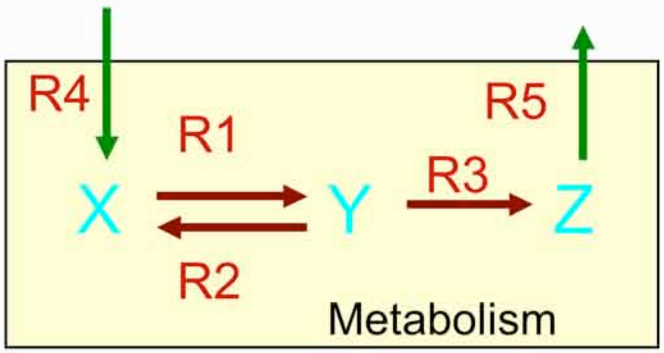
Complementarity Systems Partition of State Space





- Behavior not described by partition of state space - distinct discrete state

Flux Balance Analysis Models I



	Internal		External
R1	X → Y	R4	X
R2	Y → X	R5	-Z
R3	Y → Z		

	R1	R2	R3	R4	R5
X	-1	1	0	1	0
Y	1	-1	-1	0	0
Z	0	0	1	0	-1

$$\dot{\mathbf{c}} = \mathbf{A}\mathbf{v}$$

- **c** : Concentrations of X,Y,Z
- **v** : Reaction fluxes

Flux Balance Analysis Models II

- Reactions inside the organism occur at a very fast time scale: quasi steady-state approximation
- Steady-state does not determine all fluxes uniquely
- The metabolism can be formulated as a linear program where the metabolism is assumed to optimize an objective (e.g., growth of biomass)
- The linear program is solved to determine the fluxes

$$\begin{aligned} \min_{\mathbf{v},\mathbf{b}} \quad & \mathbf{w}^{\mathrm{T}}\mathbf{v} \\ \text{s.t.} \quad & \mathbf{A}\mathbf{v} = \mathbf{b} \\ & \mathbf{v}^{L} \leq \mathbf{v} \leq \mathbf{v}^{U} \\ & \mathbf{b}^{L} \leq \mathbf{b} \leq \mathbf{b}^{U} \end{aligned}$$

$\mathbf{w}$: objective vector

$\mathbf{A}$: stoichiometry matrix

$\mathbf{b}$: exchange and accumulation rates

$\mathbf{v}$: reaction fluxes

Flux Balance Analysis Models III

$$\dot{\mathbf{x}}(t) = \mathbf{f}(t, \mathbf{x}(t), \mathbf{u}(t))$$

$$\begin{aligned}
\mathbf{u}(t) \in \quad & \arg\min_{\mathbf{v},\mathbf{b}} \quad \mathbf{w}^{\mathrm{T}}\mathbf{v} \\
& \text{s.t.} \quad \mathbf{A}\mathbf{v} = \mathbf{b} \\
& \qquad \mathbf{v}^{L}(\mathbf{x}(t),t) \le \mathbf{v} \le \mathbf{v}^{U}(\mathbf{x}(t),t) \\
& \qquad \mathbf{b}^{L}(\mathbf{x}(t),t) \le \mathbf{b} \le \mathbf{b}^{U}(\mathbf{x}(t),t)
\end{aligned}$$

$\mathbf{x}$: state variables for bioreactor

$\mathbf{w}$: objective vector

$\mathbf{A}$: stoichiometry matrix

$\mathbf{b}$: exchange and accumulation rates

$\mathbf{v}$: reaction fluxes

Flux Balance Analysis Models IV

- Flux Balance Models can be analyzed as hybrid systems
- The discrete state is determined by the embedded linear program
  - Substitute with equivalent KKT conditions to yield a complementarity system (see next slide)
- The solution of the linear program is determined by a selection of columns of the stoichiometry matrix (basis)
- Each selection represents a potential discrete state
- There are exponentially many discrete states, hence enumeration of all modes not practical

Dynamic System with LP Embedded

$$\dot{\mathbf{x}}(t) = \mathbf{f}(t, \mathbf{x}(t), \mathbf{u}(t))$$

$$\mathbf{u}(t) \in \quad \arg\min_{\mathbf{v} \in \mathbb{R}^{n_v}} \quad \mathbf{c}(\mathbf{x}(t))^{\mathrm{T}} \mathbf{v}$$

$$\text{s.t.} \quad \mathbf{A}(\mathbf{x}(t))\mathbf{v} = \mathbf{b}(\mathbf{x}(t))$$

$$\mathbf{v} \geq \mathbf{0}$$

$$\Leftrightarrow$$

$$\dot{\mathbf{x}}(t) = \mathbf{f}(t, \mathbf{x}(t), \mathbf{u}(t))$$

$$\mathbf{c}(\mathbf{x}(t)) + \mathbf{A}(\mathbf{x}(t))^{\mathrm{T}} \mathbf{z}(t) - \mathbf{y}(t) = \mathbf{0}$$

$$\mathbf{A}(\mathbf{x}(t))\mathbf{u}(t) = \mathbf{b}(\mathbf{x}(t))$$

$$y_i(t) u_i(t) = 0, \quad \forall i = 1, \ldots, n_u$$

$$\mathbf{y}(t) \geq \mathbf{0}, \ \mathbf{u}(t) \geq \mathbf{0}$$

Flux Balance Analysis Models V

$$\dot{V} = F$$

$$\dot{X} = \mu X$$

$$\dot{G} = FG_f / V - u_g X$$

$$\dot{E} = u_e X$$

$$\begin{aligned} \mathbf{u}(t) \in \quad & \arg\max_{\mathbf{v},\mathbf{b}} \quad \mu = \mathbf{w}^{\mathrm{T}}\mathbf{v} \\ & \text{s.t.} \quad \mathbf{A}\mathbf{v} = \mathbf{b} \\ & \qquad v_g^L(t) \le v_g \le v_g^U(G(t), E(t)) \\ & \qquad v_o^L(t) \le v_o \le v_o^U(O(t)) \\ & \qquad \mathbf{v}^L \le \mathbf{v} \le \mathbf{v}^U \\ & \qquad \mathbf{b}^L \le \mathbf{b} \le \mathbf{b}^U \end{aligned}$$

F : Glucose feed rate

O : Oxygen concentration

G_f : Glucose feed concentration

V : Volume

X : Biomass concentration

E : Ethanol concentration

G : Glucose concentration

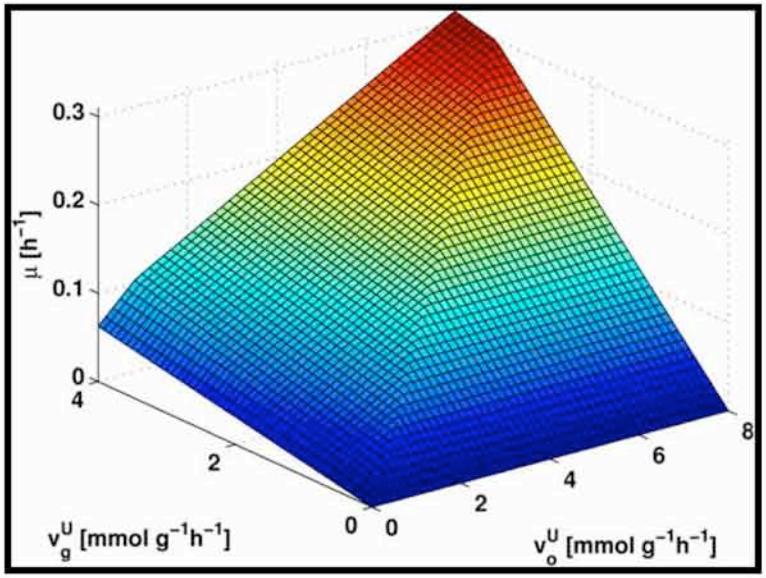
v_g, u_g : Glucose membrane exchange flux

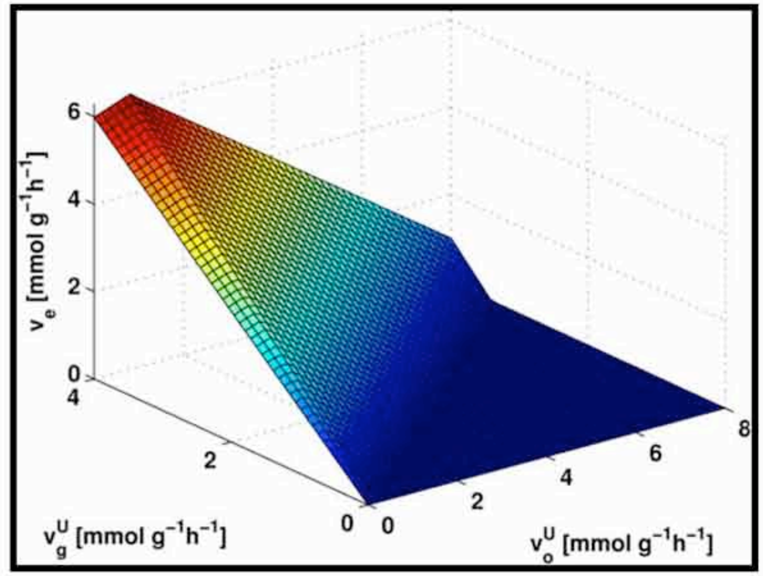
v_e, u_e : Ethanol membrane exchange flux

v_o : Oxygen membrane exchange flux

μ : Cellular growth rate

Flux Balance Analysis Models VI*
S. Cerevisiae

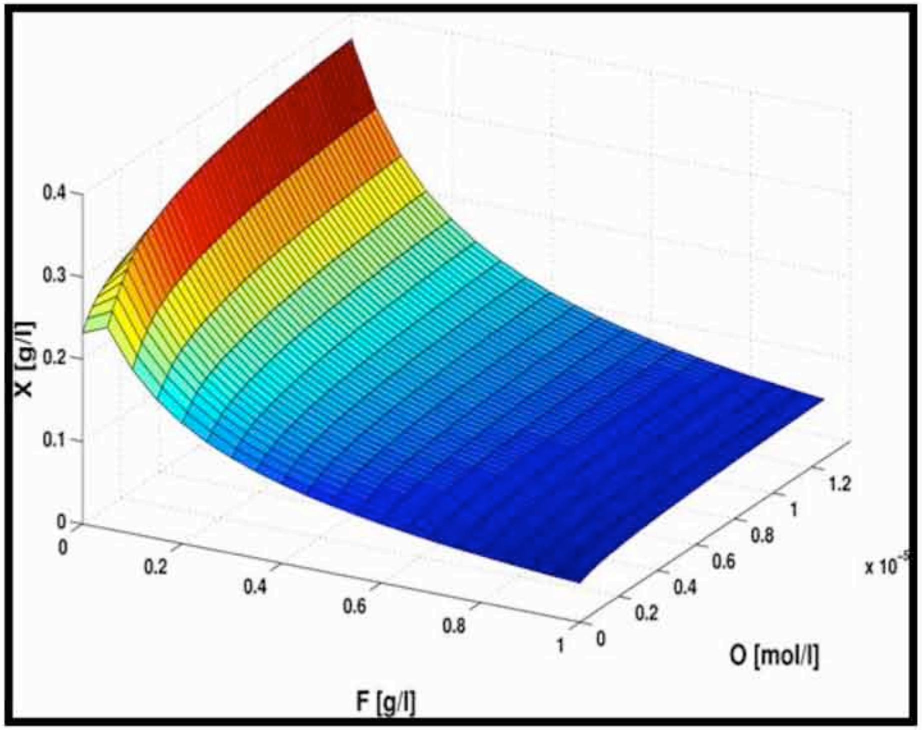


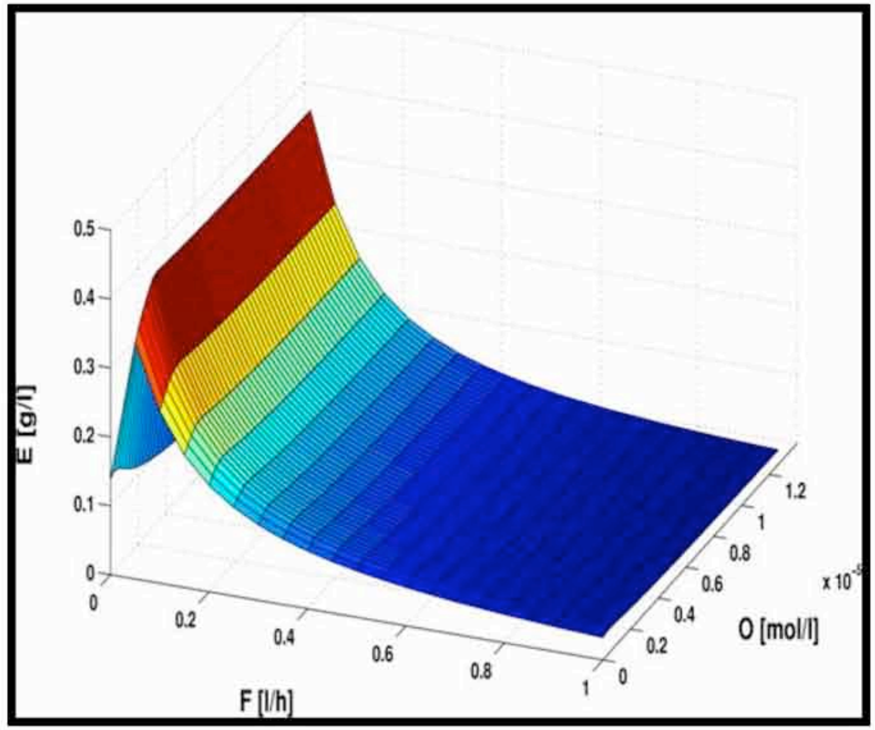


- Ethanol membrane exchange rate and biomass production rate as functions of the glucose and oxygen membrane exchange bounds

* FBA model courtesy of Jared Hjersted and Professor Michael A. Henson, University of Massachusetts, Amherst

Flux Balance Analysis Models VII
S. Cerevisiae





- The biomass and ethanol concentrations after 1 hour for different feed rates and oxygen concentrations.

Differential Variational Inequalities (DVI)

$$\dot{\mathbf{x}}(t) = \mathbf{f}(t, \mathbf{x}(t), \mathbf{u}(t))$$

$$\mathbf{u}(t) \in \arg\min_{\mathbf{v} \in C} g(\mathbf{x}(t), \mathbf{v})$$

C : convex set

$g(\mathbf{z}, \cdot)$: convex function

$$\Leftrightarrow$$

$$\dot{\mathbf{x}}(t) = \mathbf{f}(t, \mathbf{x}(t), \mathbf{u}(t))$$

$$\nabla g(\mathbf{x}(t), \mathbf{u}(t))^{\mathrm{T}} (\mathbf{w} - \mathbf{u}(t)) \geq 0, \quad \forall \mathbf{w} \in C \qquad \textbf{(DVI)}$$

$$\mathbf{u}(t) \in C$$

$$\Leftrightarrow$$

$$\dot{\mathbf{x}}(t) = \mathbf{f}(t, \mathbf{x}(t), \mathbf{u}(t))$$

KKT conditions for $\mathbf{u}(t)$

Basic Embedded LP for Complementarity Problems

$$
\begin{aligned}
\dot{x}(t) &= f(t, x(t), u(t)) \\
u(t) &\in \arg\min_{v} \quad h(x(t))v \\
& \qquad \text{s.t.} \qquad v \geq 0
\end{aligned}
\quad \Leftrightarrow \quad
\begin{aligned}
\dot{x}(t) &= f(t, x(t), u(t)) \\
y(t) &= h(x(t)) \\
y(t)u(t) &= 0 \\
y(t) \geq 0&, u(t) \geq 0
\end{aligned}
$$

$$
\begin{aligned}
h(x(t)) > 0 &\quad \Rightarrow \quad u(t) = 0 \\
h(x(t)) = 0 &\quad \Rightarrow \quad u(t) \geq 0 \\
h(x(t)) < 0 &\quad \Rightarrow \quad \text{unbounded infimum, no solution}
\end{aligned}
$$

Embedded LP for each path constraint enforced

Differential Variational Inequalities - Generalization?

$$\dot{\mathbf{x}}(t) = \mathbf{f}(t, \mathbf{x}(t), \mathbf{u}(t))$$
$$\mathbf{u}(t) \in \arg \min_{\mathbf{v} \in C(\mathbf{x}(t))} g(\mathbf{x}(t), \mathbf{v})$$

$C(\mathbf{z})$: convex set

$g(\mathbf{z}, \cdot)$: convex function

$$\Leftrightarrow$$

$$\dot{\mathbf{x}}(t) = \mathbf{f}(t, \mathbf{x}(t), \mathbf{u}(t))$$
$$\nabla g(\mathbf{x}(t), \mathbf{u}(t))^{\mathrm{T}} (\mathbf{w} - \mathbf{u}(t)) \geq 0, \quad \forall \mathbf{w} \in C(\mathbf{x}(t)) \qquad \textbf{(gDVI)}$$
$$\mathbf{u}(t) \in C(\mathbf{x}(t))$$

$$\Leftrightarrow$$

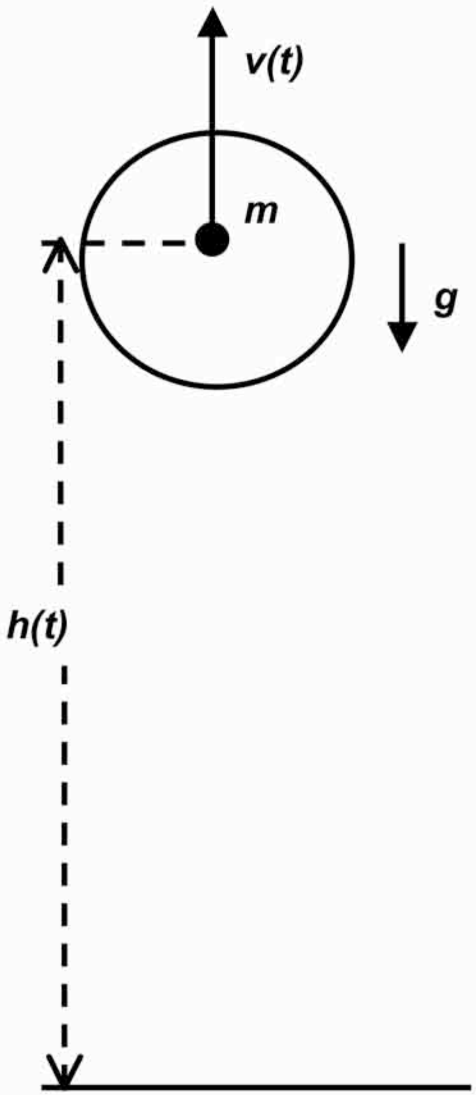
$$\dot{\mathbf{x}}(t) = \mathbf{f}(t, \mathbf{x}(t), \mathbf{u}(t))$$

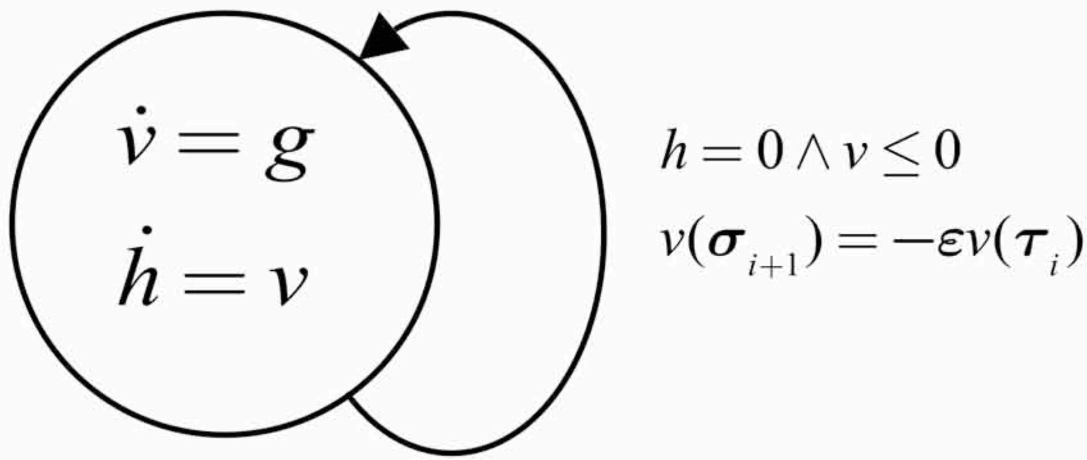
KKT conditions for $\mathbf{u}(t)$

Pathological Behaviors

- Deadlock
- Zeno behavior
- Sliding mode

Bouncing Ball

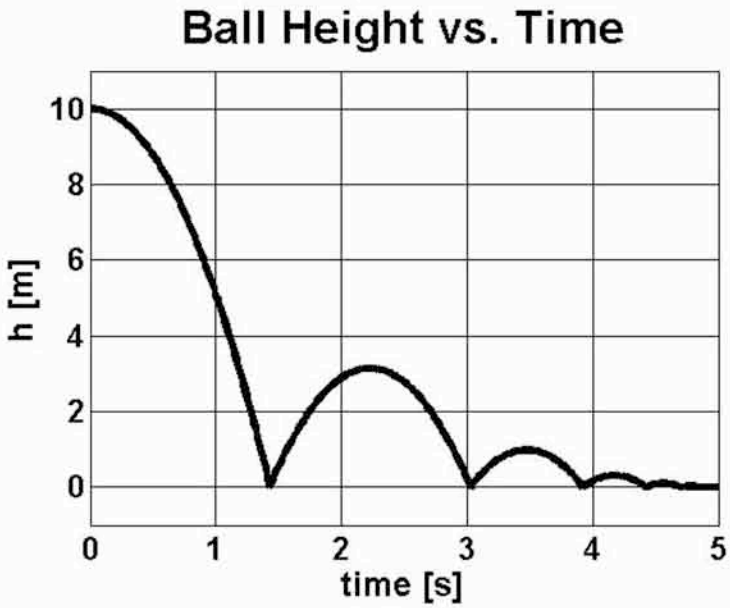


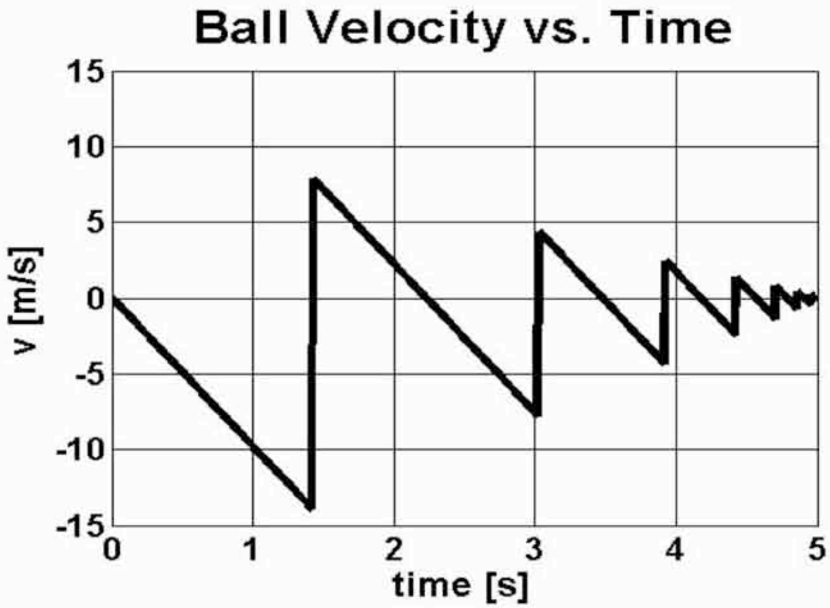


Bouncing Ball

Zeno Execution

$$h(t_0) > 0 \Rightarrow$$ **Accumulation of event times**

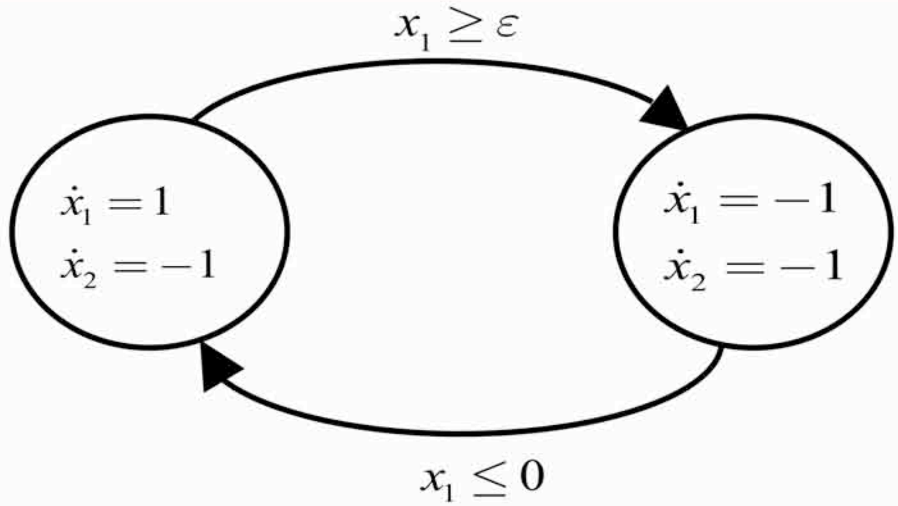




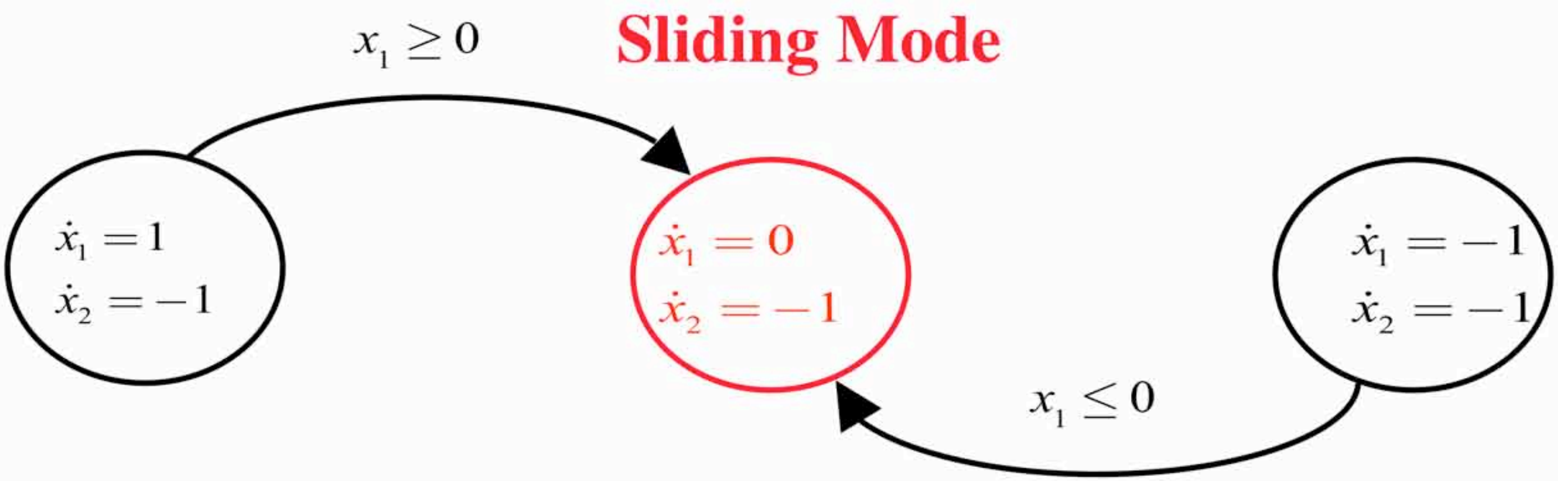
Deadlock

$$h(t_0) = 0 \wedge v(t_0) = 0 \Rightarrow$$ **Absence of continuous evolution**

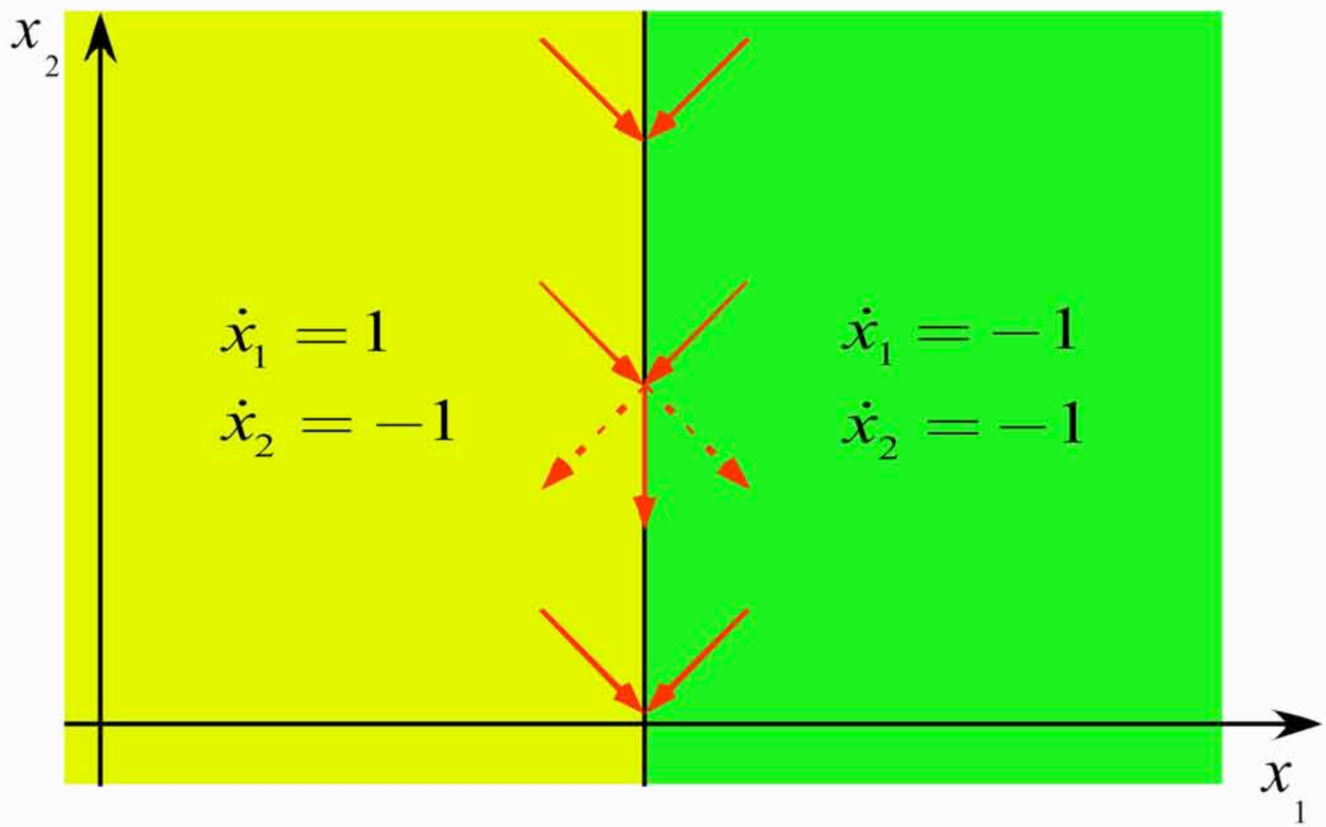
Sliding Mode



- if $\varepsilon = 0$, redefine dynamics to "slide" on $x_1 = 0$



Sliding Mode

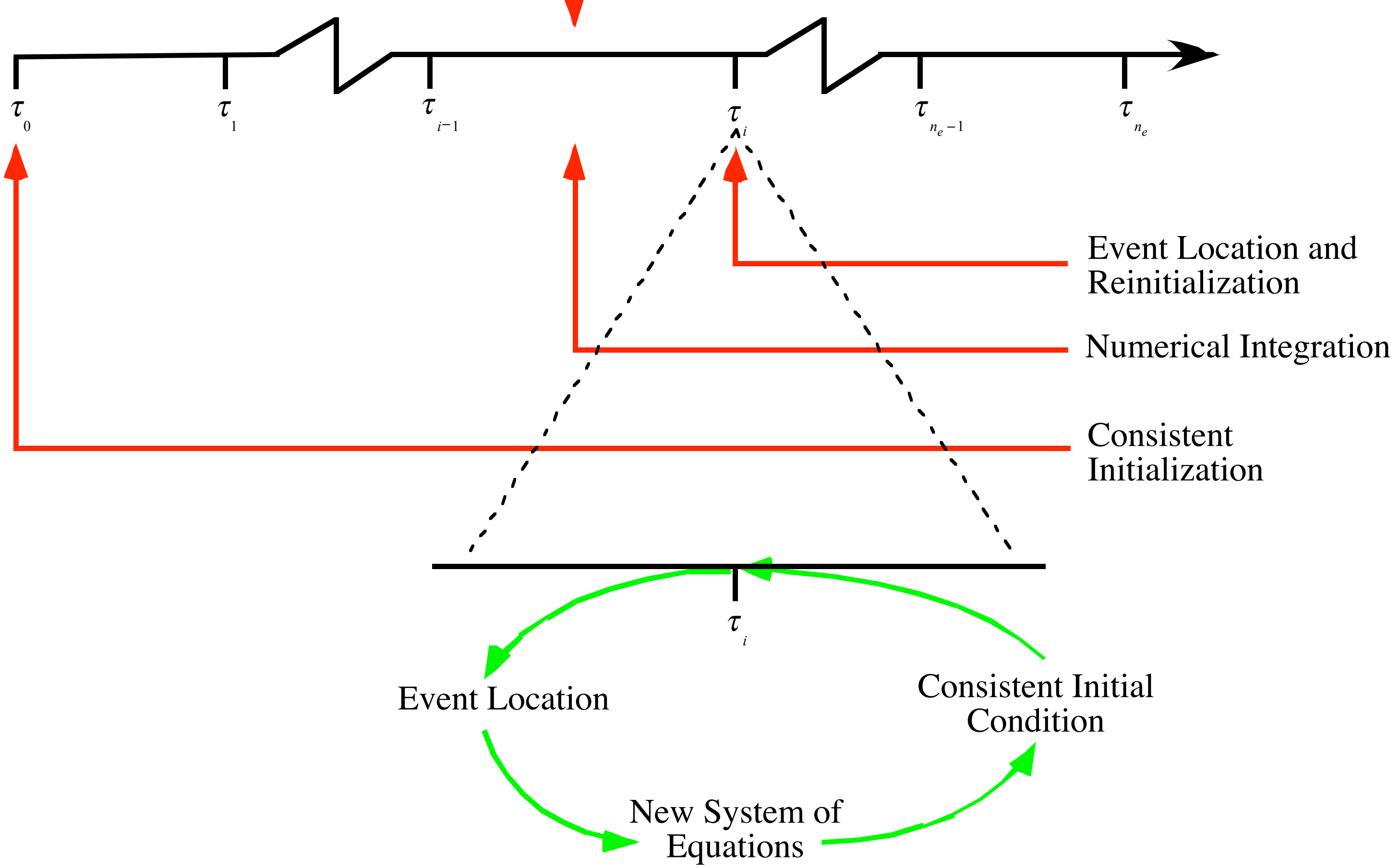


- DCP solution and vector field nonunique on boundary
- One of the possible vector fields allows sliding along the boundary

SIMULATION

Simulation of Hybrid Systems

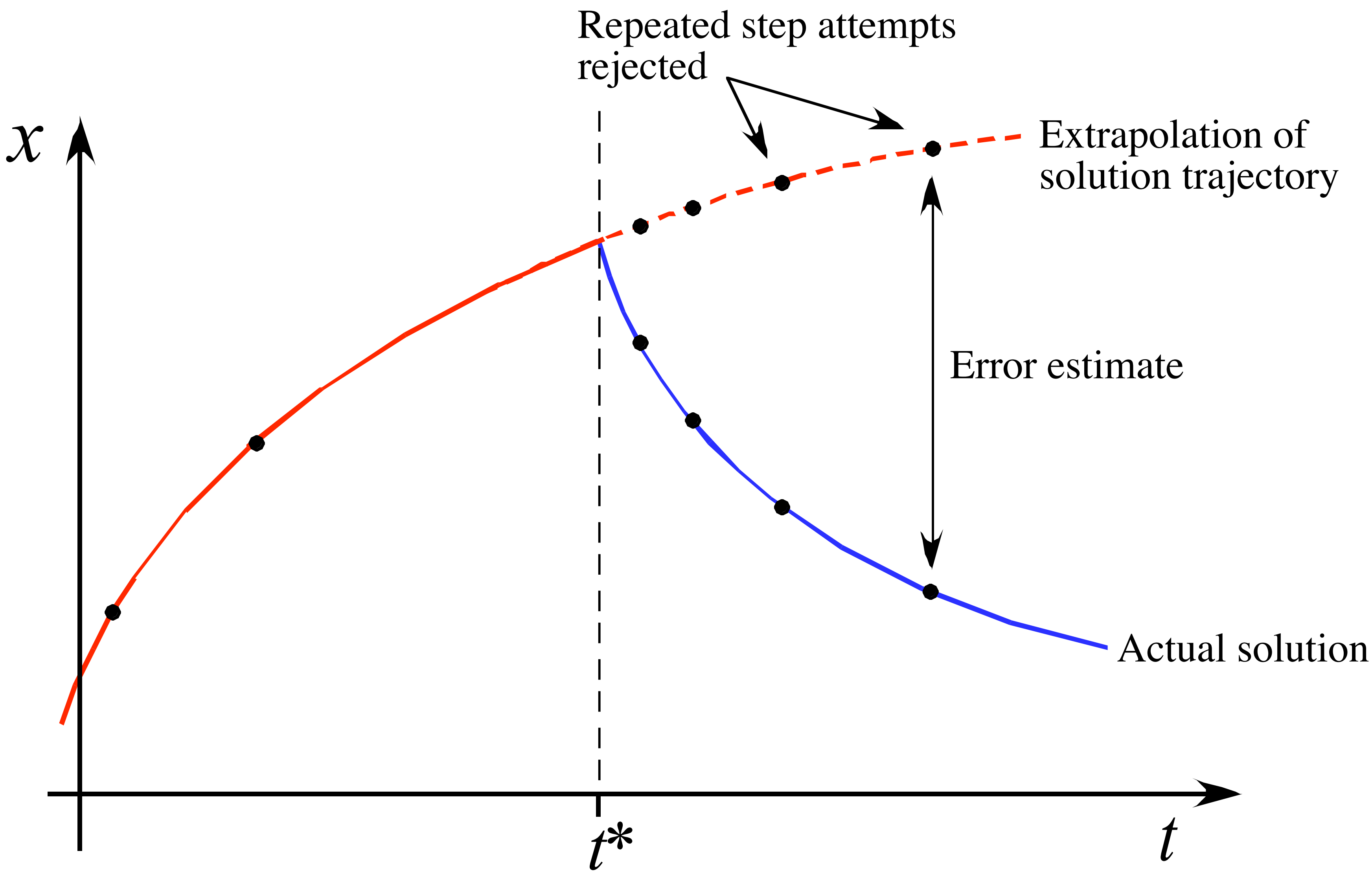
$$\mathbf{f}^{(m_i)}(t, \dot{\mathbf{x}}(t,\mathbf{p}), \mathbf{x}(t,\mathbf{p}), \mathbf{p}) = \mathbf{0}$$



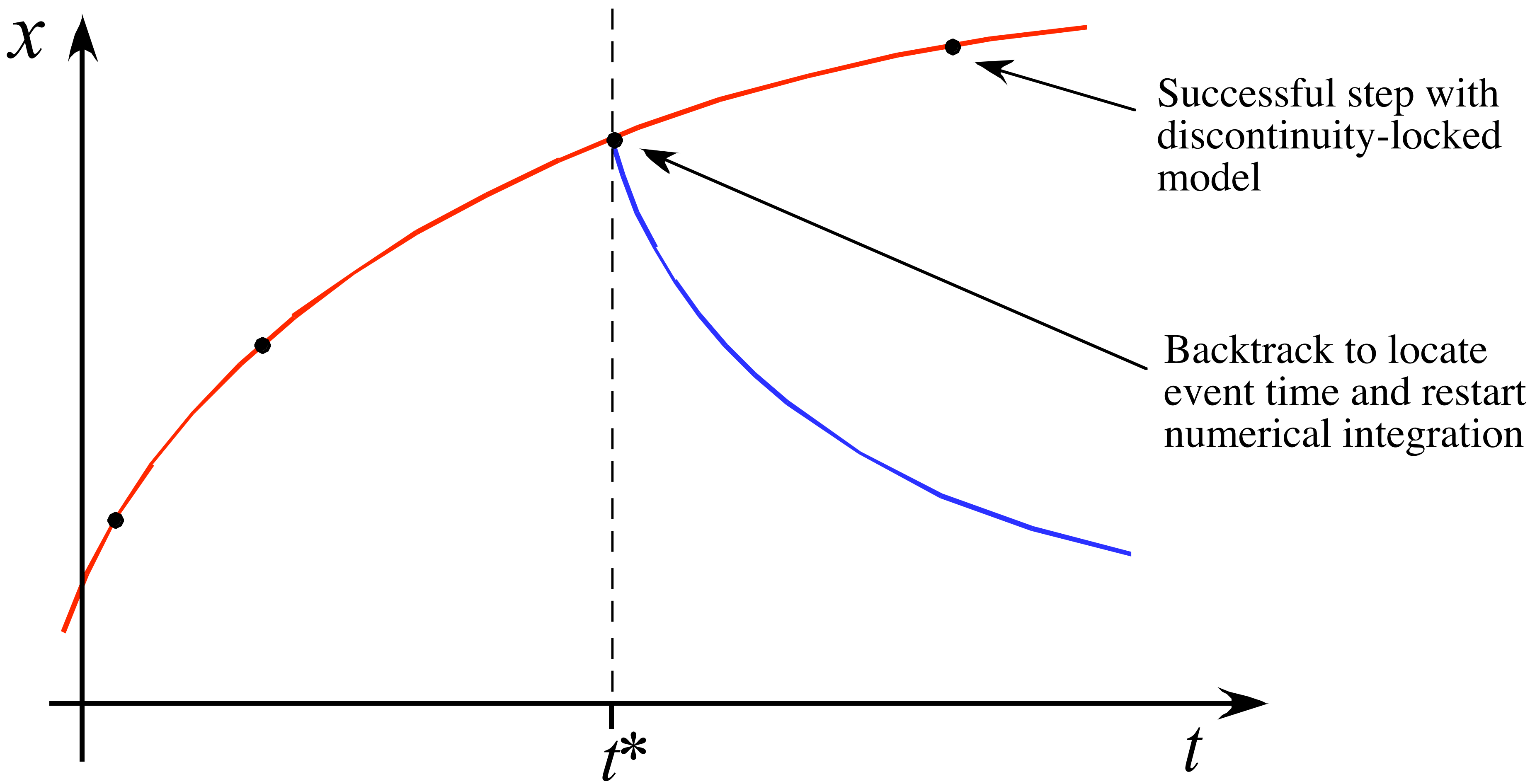
Numerical Integration of ODEs and DAEs

- Large, sparse, stiff ODEs and DAEs
  - State-of-the-art dynamic simulation technology
- Variable step size methods that control numerical integration error
- *Predictor* step - extrapolates past time trajectory
- *Corrector* step - Newton iteration to converge discretized differential equations
- Error control based on difference between predictor and corrector steps
- *Discontinuity locking* - do not allow equation switching while converging integration step

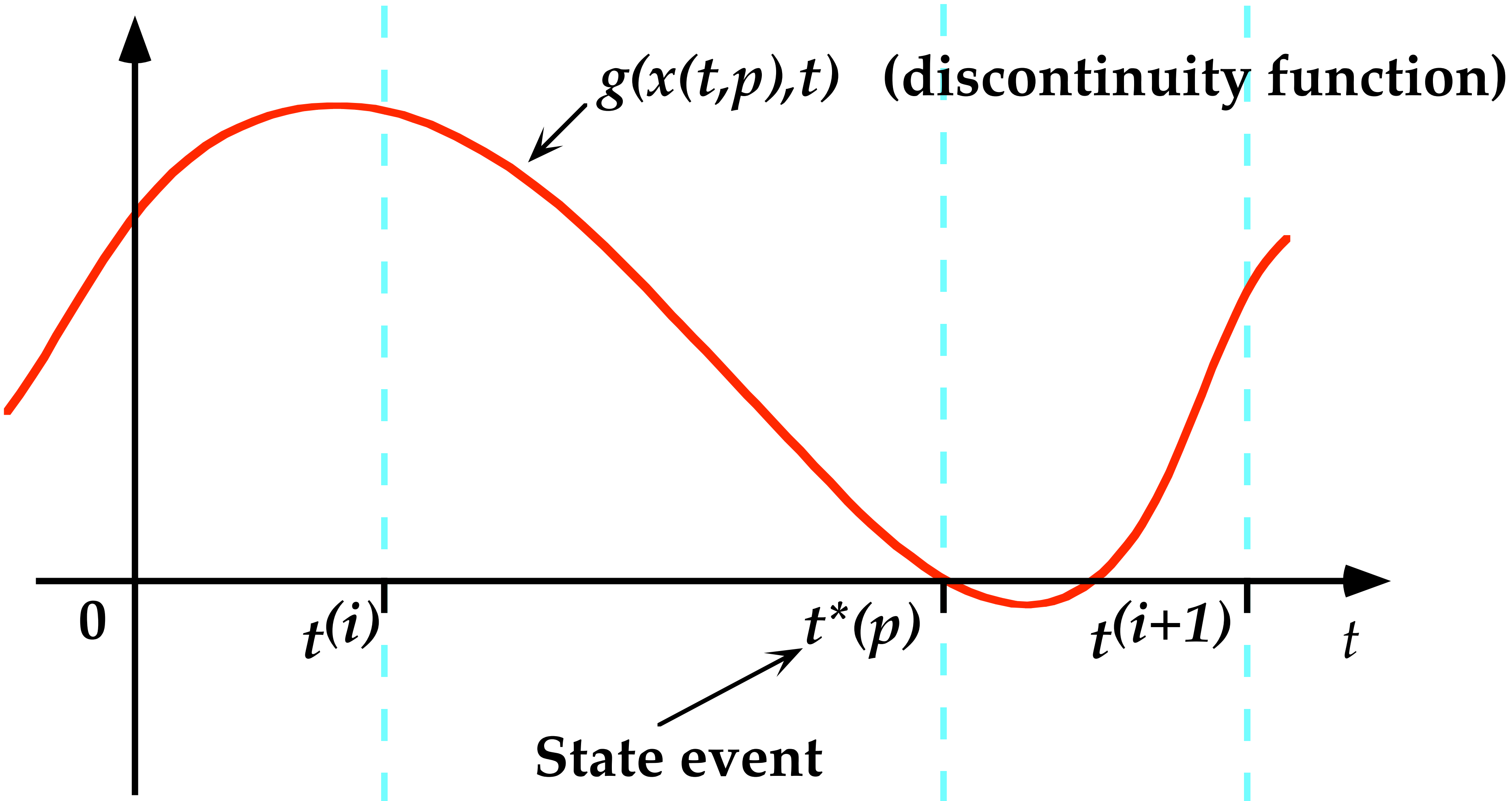
Event Location Without Discontinuity Locking



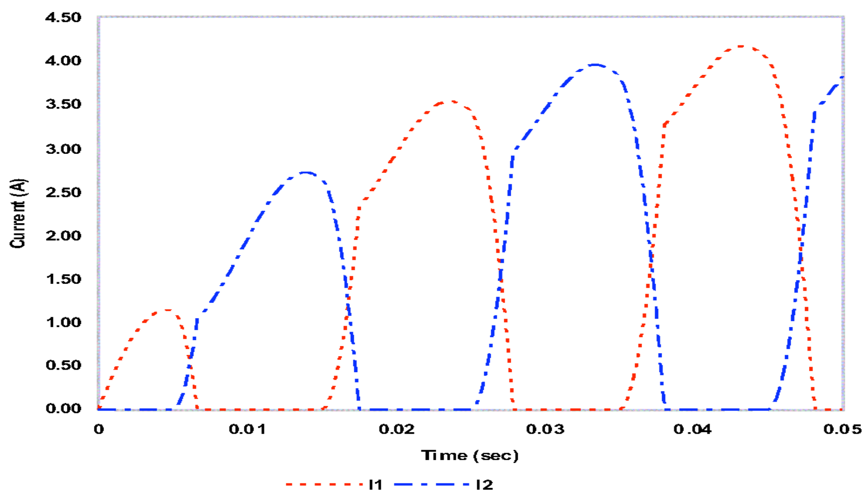
Event Location With Discontinuity Locking

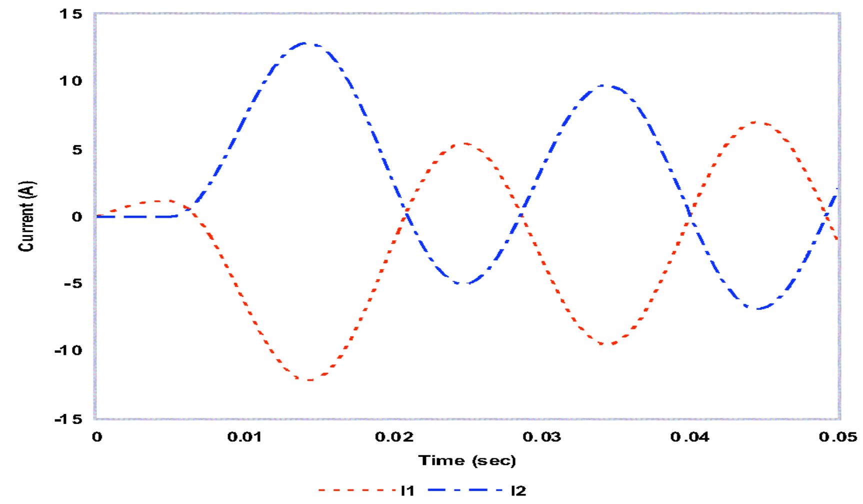


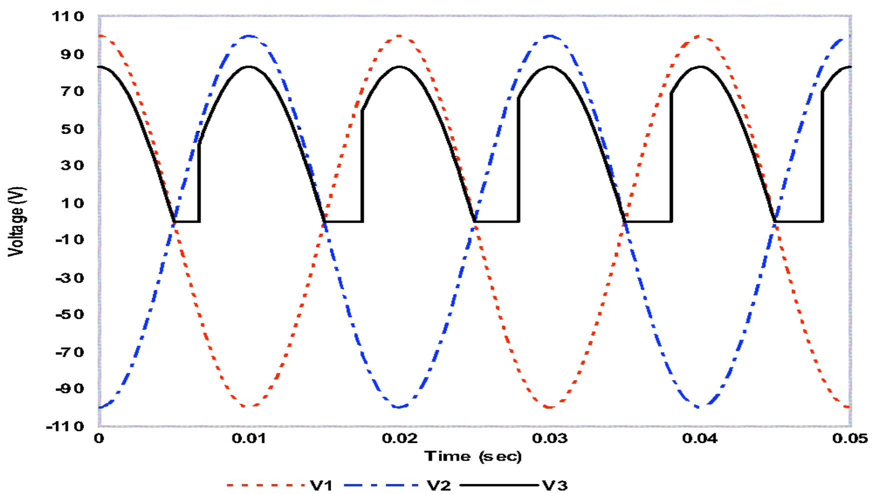
State Events
Discontinuity Function

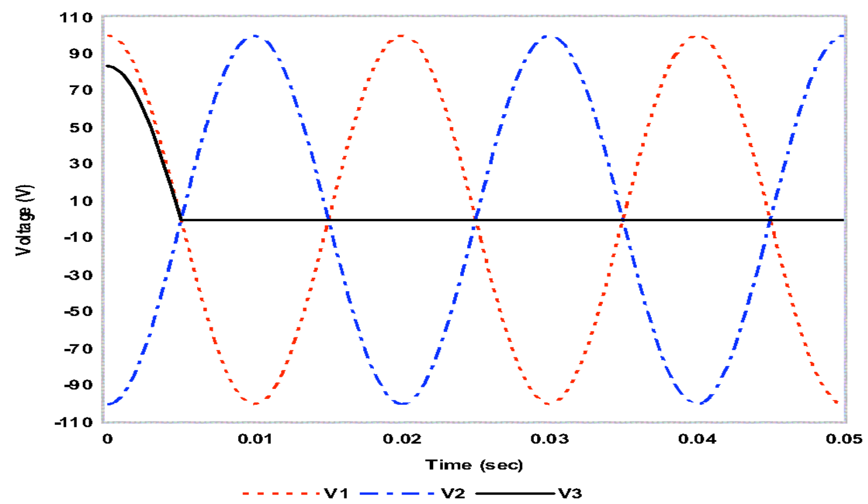


Getting State Event Location Right...

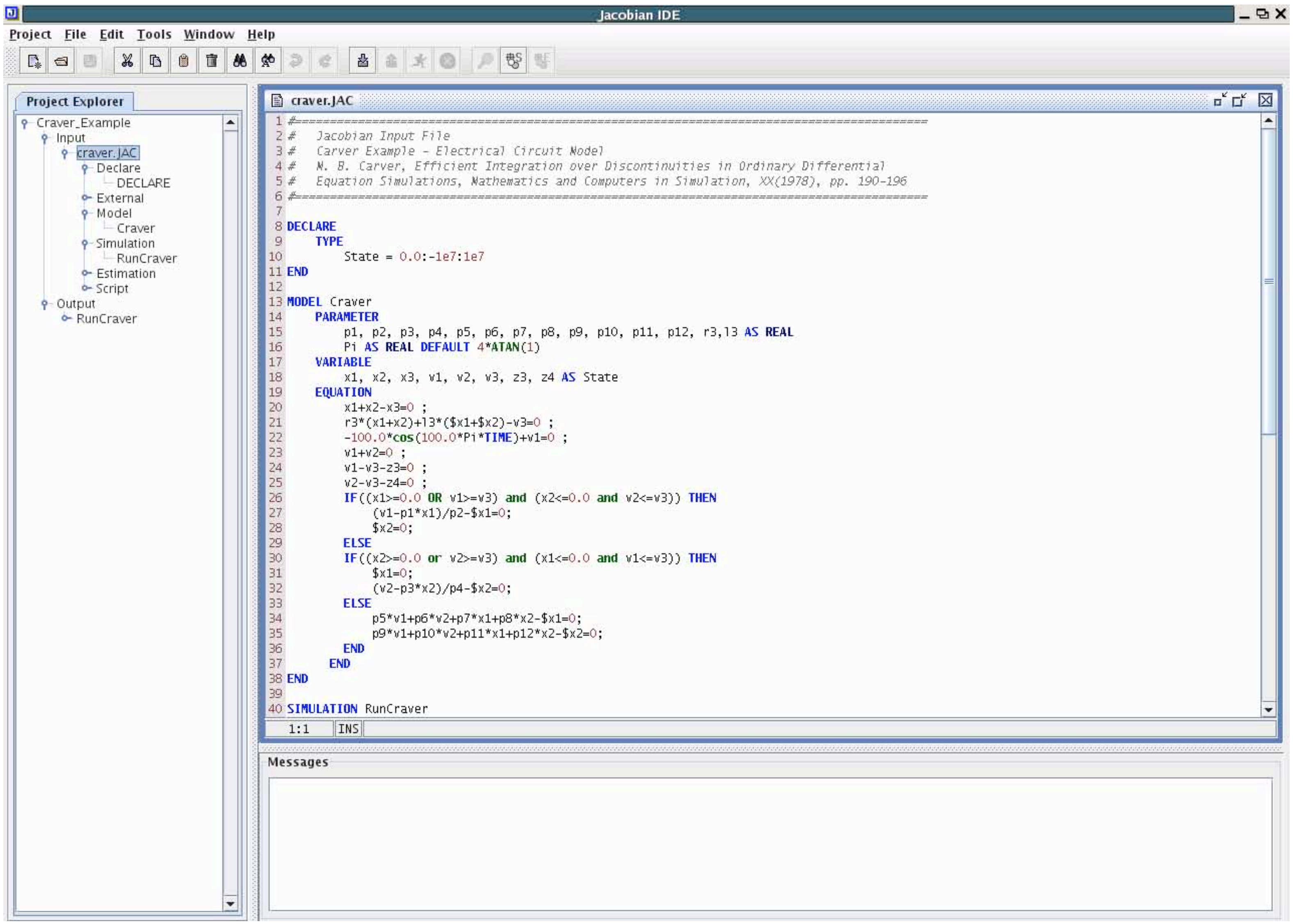




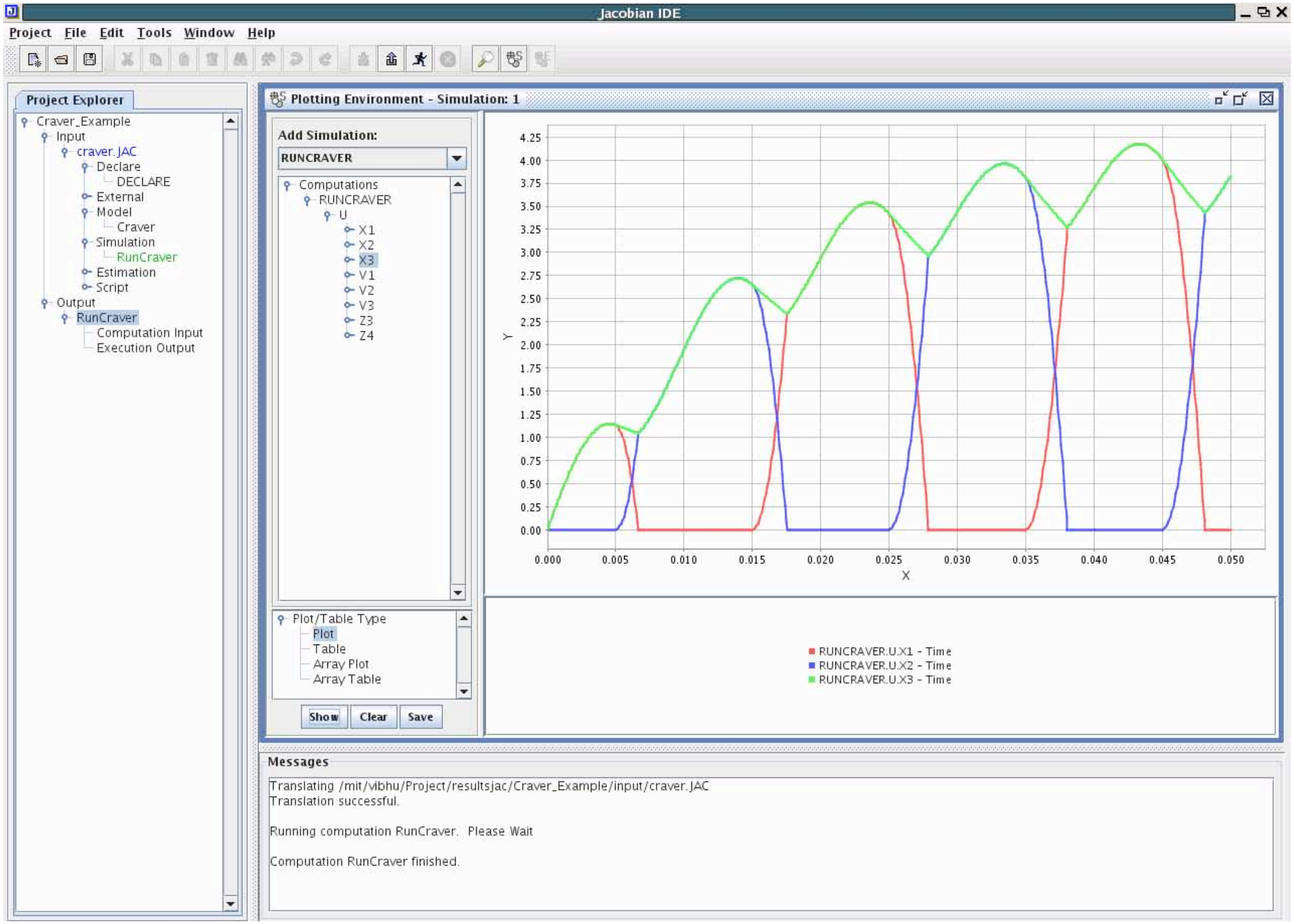




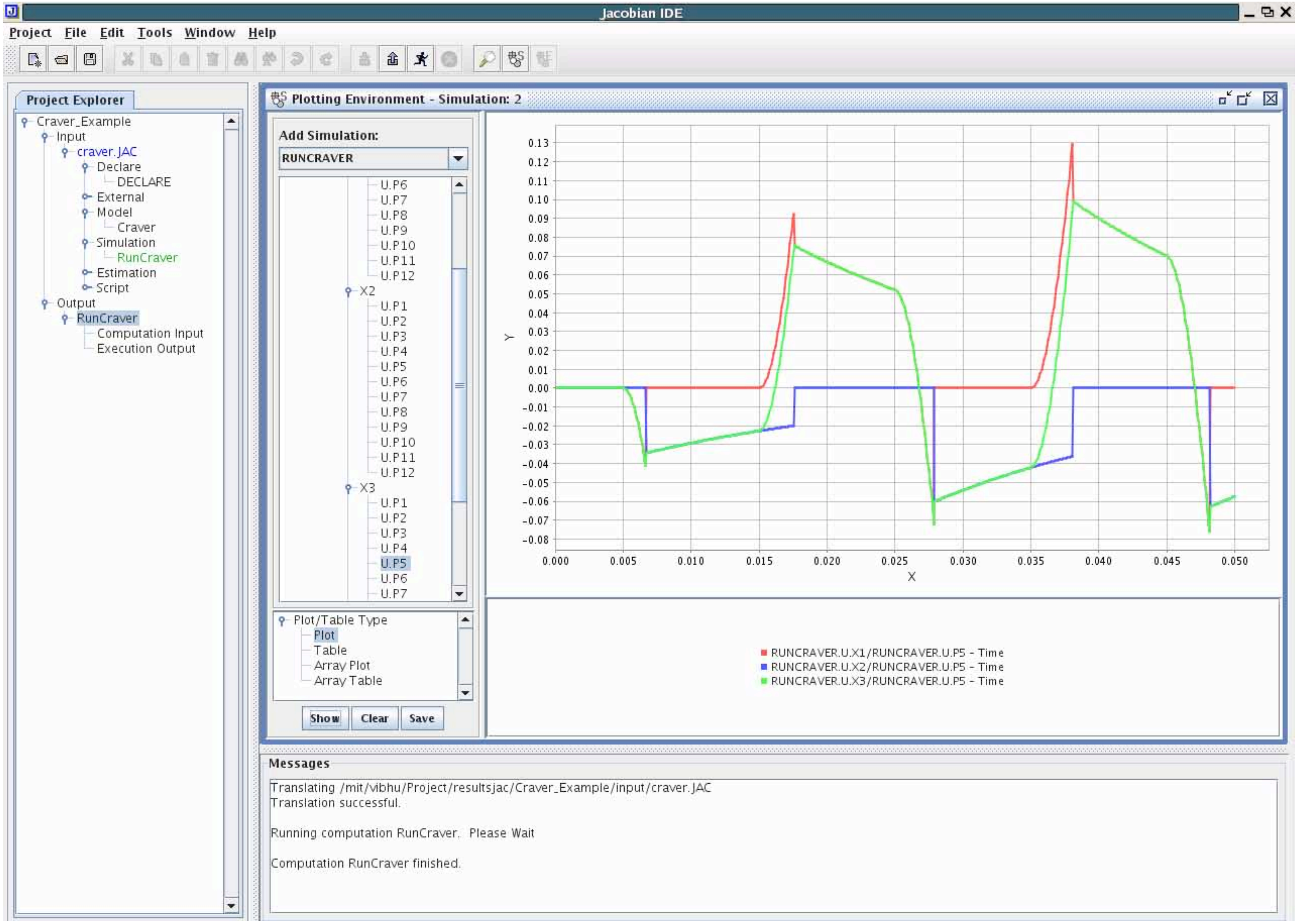
Jacobian®
http://numericatech.com



Jacobian®
http://numericatech.com



Jacobian®
http://numericatech.com



SENSITIVITY ANALYSIS

Parametric Sensitivities of a Dynamic System

- Given a dynamic system expressed in terms of a vector of time-invariant parameters **p**:

$$\dot{\mathbf{x}}(t,\mathbf{p}) = \mathbf{f}(t,\mathbf{x}(t,\mathbf{p}),\mathbf{p})$$

- Then, there exists a n_x by n_p matrix of partial derivatives $\partial\mathbf{x}/\partial\mathbf{p}$ determined by the related matrix differential equations:

$$\frac{d}{dt}\frac{\partial\mathbf{x}}{\partial\mathbf{p}}(t,\mathbf{p}) = \frac{\partial\mathbf{f}}{\partial\mathbf{x}}(t,\mathbf{x}(t,\mathbf{p}),\mathbf{p})\frac{\partial\mathbf{x}}{\partial\mathbf{p}}(t,\mathbf{p}) + \frac{\partial\mathbf{f}}{\partial\mathbf{p}}(t,\mathbf{x}(t,\mathbf{p}),\mathbf{p})$$

- Extension to DAEs straightforward

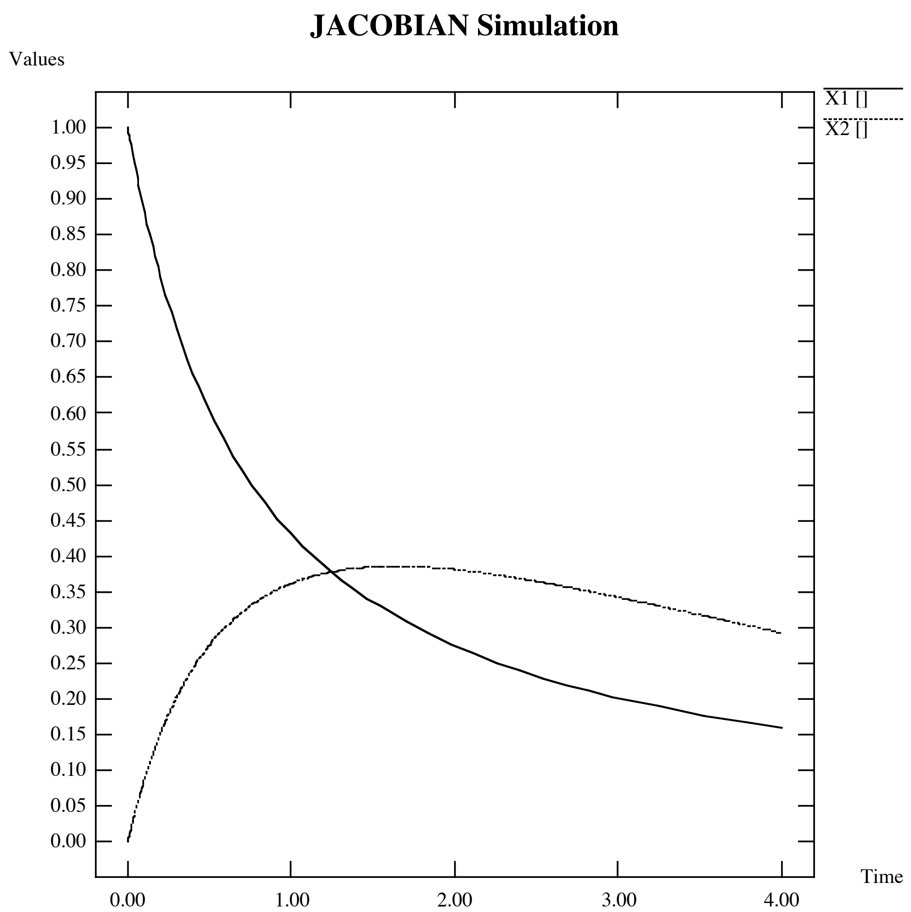
Dynamic Sensitivities Gas Oil Cracking Example

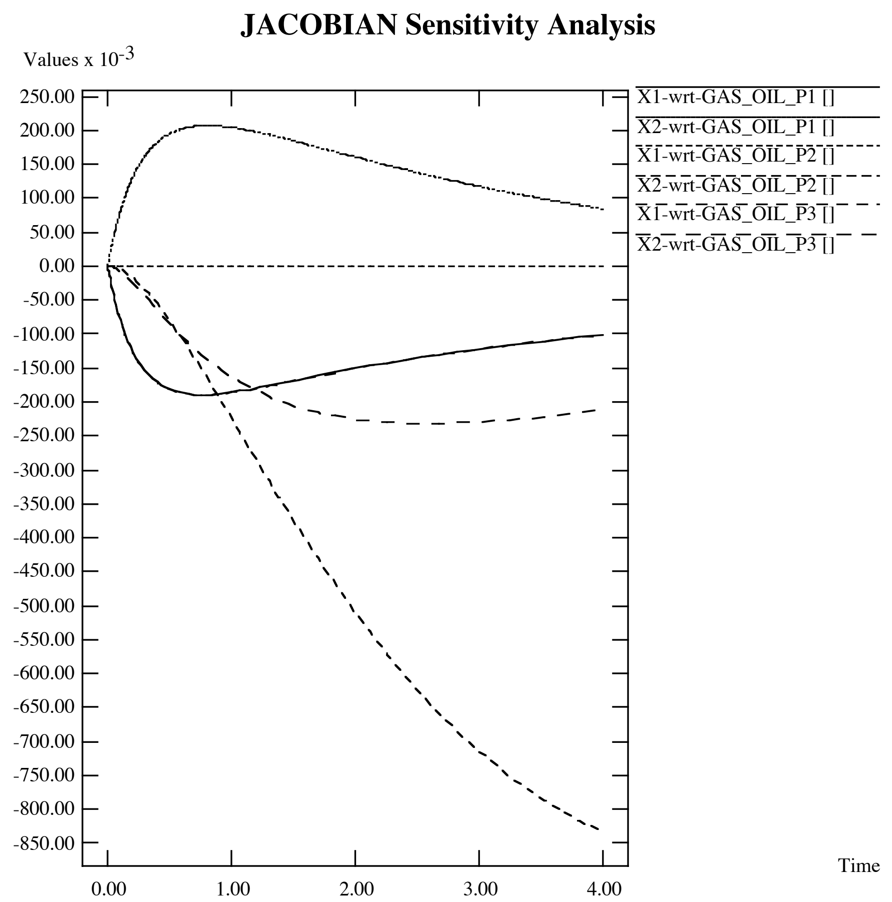
- Gas oil cracking kinetics described by the differential equations:

$$\boxed{\begin{aligned} \frac{dx_1}{dt} &= -(p_1 + p_3)x_1^2 \\ \frac{dx_2}{dt} &= p_1 x_1^2 - p_2 x_2 \end{aligned}} \quad \Rightarrow \quad \boxed{\begin{aligned} \frac{d}{dt}\frac{\partial x_1}{\partial p_1} &= -2(p_1 + p_3)x_1 \frac{\partial x_1}{\partial p_1} - x_1^2 \\ \frac{d}{dt}\frac{\partial x_2}{\partial p_1} &= 2p_1 x_1 \frac{\partial x_1}{\partial p_1} + x_1^2 - p_2 \frac{\partial x_2}{\partial p_1} \end{aligned}}$$

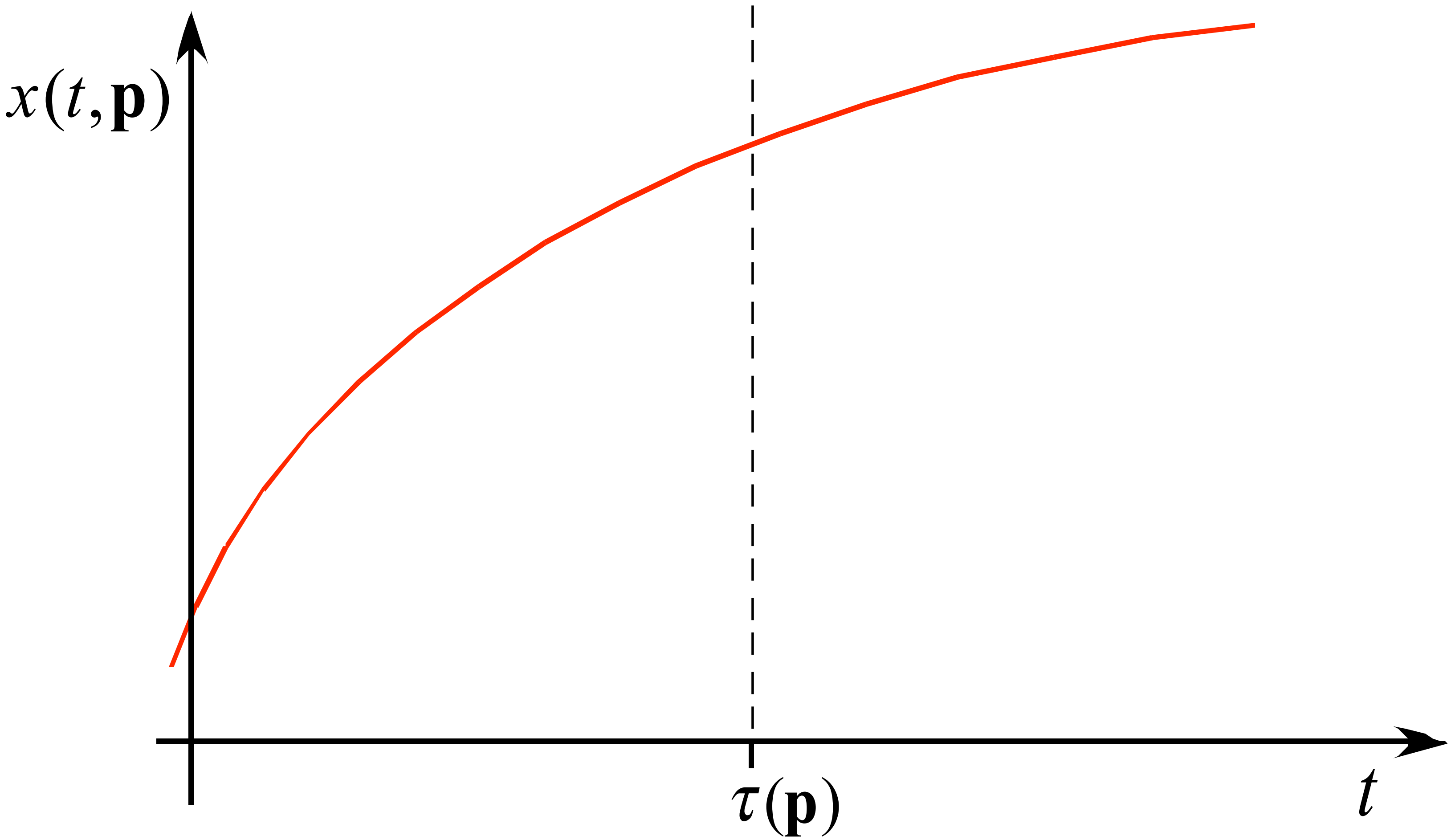
- Examine sensitivity of solution to kinetic parameters: p_1, p_2, p_3.
- Magnitude and time scale of influence of parameter

Dynamic Sensitivities Gas Oil Cracking Example





Sensitivity at Parameter Dependent Time



Sensitivity at Parameter Dependent Time

$\frac{\partial \mathbf{x}}{\partial \mathbf{p}}(t,\mathbf{p})$ Sensitivity of ODE solution without parametric dependence of time (deriv. of $\mathbf{x}(t,\mathbf{p})$ w.r.t. 2nd argument)

Total derivative of $\mathbf{x}(\tau(\mathbf{p}),\mathbf{p})$ includes contribution from parametric dependence of time:

$$\frac{D\mathbf{x}}{D\mathbf{p}}(\tau(\mathbf{p}),\mathbf{p}) = \frac{d\mathbf{x}}{dt}(\tau(\mathbf{p}),\mathbf{p})\frac{\partial \tau}{\partial \mathbf{p}}(\mathbf{p}) + \frac{\partial \mathbf{x}}{\partial \mathbf{p}}(\tau(\mathbf{p}),\mathbf{p})$$

$$\Leftrightarrow$$

$$\frac{D\mathbf{x}}{D\mathbf{p}}(\tau(\mathbf{p}),\mathbf{p}) = \mathbf{f}(\tau(\mathbf{p}),\mathbf{x}(\tau(\mathbf{p}),\mathbf{p}),\mathbf{p})\frac{\partial \tau}{\partial \mathbf{p}}(\mathbf{p}) + \frac{\partial \mathbf{x}}{\partial \mathbf{p}}(\tau(\mathbf{p}),\mathbf{p})$$

Sensitivity at Parameter Dependent Time

If $\tau = p$ then $\dfrac{\partial \tau}{\partial p} = 1$ and:

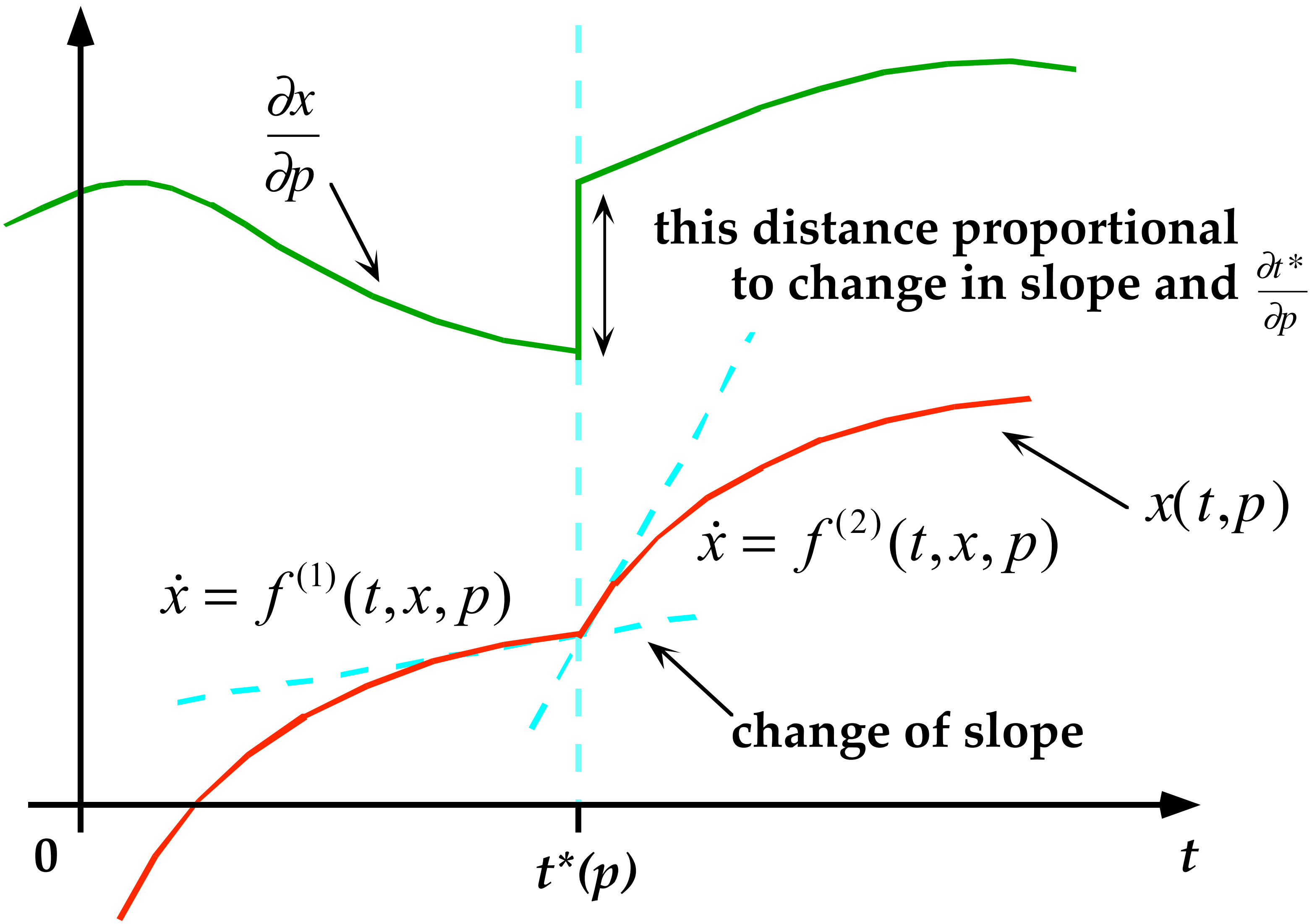
$$\frac{D\mathbf{x}}{Dp}(\tau(p),p) = \mathbf{f}(\tau(p),\mathbf{x}(t,p),p) + \frac{\partial \mathbf{x}}{\partial p}(\tau(p),p)$$

If $\tau(\mathbf{p})$ given by $g(\mathbf{x}(\tau(\mathbf{p}),\mathbf{p}),\mathbf{p}) = 0$ then solve:

$$\frac{\partial g}{\partial \mathbf{x}}(\mathbf{x}(\tau(\mathbf{p}),\mathbf{p}),\mathbf{p})\left(\frac{d\mathbf{x}}{dt}(\tau(\mathbf{p}),\mathbf{p})\frac{\partial \tau}{\partial \mathbf{p}}(\mathbf{p}) + \frac{\partial \mathbf{x}}{\partial \mathbf{p}}(\tau(\mathbf{p}),\mathbf{p})\right) + \frac{\partial g}{\partial \mathbf{p}}(\mathbf{x}(\tau(\mathbf{p}),\mathbf{p}),\mathbf{p}) = \mathbf{0}$$

for $\dfrac{\partial \tau}{\partial \mathbf{p}}(\mathbf{p})$ (if solution exists and is unique: transversality condition).

Sensitivities at Events Characteristic Jump



Sensitivities at Events

Transition function: $\mathbf{x}(\sigma_{i+1}(\mathbf{p}),\mathbf{p}) = \mathbf{x}(\tau_i(\mathbf{p}),\mathbf{p}).$

Differentiate w.r.t. **p**:

$$\frac{d\mathbf{x}}{dt}(\sigma_{i+1}(\mathbf{p}),\mathbf{p})\frac{\partial \sigma_{i+1}}{\partial \mathbf{p}}(\mathbf{p}) + \frac{\partial \mathbf{x}}{\partial \mathbf{p}}(\sigma_{i+1}(\mathbf{p}),\mathbf{p}) = \frac{d\mathbf{x}}{dt}(\tau_i(\mathbf{p}),\mathbf{p})\frac{\partial \tau_i}{\partial \mathbf{p}}(\mathbf{p}) + \frac{\partial \mathbf{x}}{\partial \mathbf{p}}(\tau_i(\mathbf{p}),\mathbf{p}),$$

or because $\sigma_{i+1}(\mathbf{p}) = \tau_i(\mathbf{p}) \Rightarrow \dfrac{\partial \sigma_{i+1}}{\partial \mathbf{p}}(\mathbf{p}) = \dfrac{\partial \tau_i}{\partial \mathbf{p}}(\mathbf{p}):$

$$\frac{\partial \mathbf{x}}{\partial \mathbf{p}}(\sigma_{i+1}(\mathbf{p}),\mathbf{p}) = \frac{\partial \mathbf{x}}{\partial \mathbf{p}}(\tau_i(\mathbf{p}),\mathbf{p}) +$$

$$\left(\mathbf{f}^{(m_i)}(\tau_i(\mathbf{p}),\mathbf{x}(\tau_i(\mathbf{p}),\mathbf{p}),\mathbf{p}) - \mathbf{f}^{(m_{i+1})}(\sigma_{i+1}(\mathbf{p}),\mathbf{x}(\sigma_{i+1}(\mathbf{p}),\mathbf{p}),\mathbf{p})\right)\frac{\partial \tau_i}{\partial \mathbf{p}}(\mathbf{p}).$$

Sensitivities at Events

$$\frac{\partial \tau_i}{\partial \mathbf{p}}(\mathbf{p}) \text{ from differentiation of transition condition as before}$$

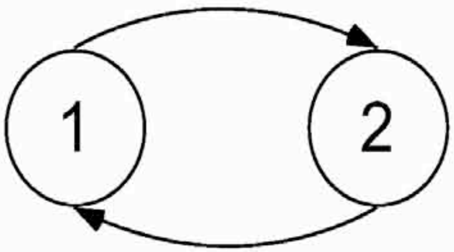
- Can also differentiate more general transition function
- Transversality condition implies unique sensitivity of event time
- Result assumes given sequence of events that does not change in a neighborhood of parameter value

Existence and Uniqueness Theorems

- Existence and uniqueness for sensitivities of hybrid systems embedded with:
  - Nonlinear ODEs
  - Linear time invariant DAEs
  - Index 1 semi-explicit DAEs
- Sensitivities of hybrid systems exist *almost everywhere*
- Critical parameter values at which (for example) sequence of events changes qualitatively
  - Hypotheses of theorems do not hold
  - Typically associated with discontinuity or nonsmoothness of the objective function in optimization

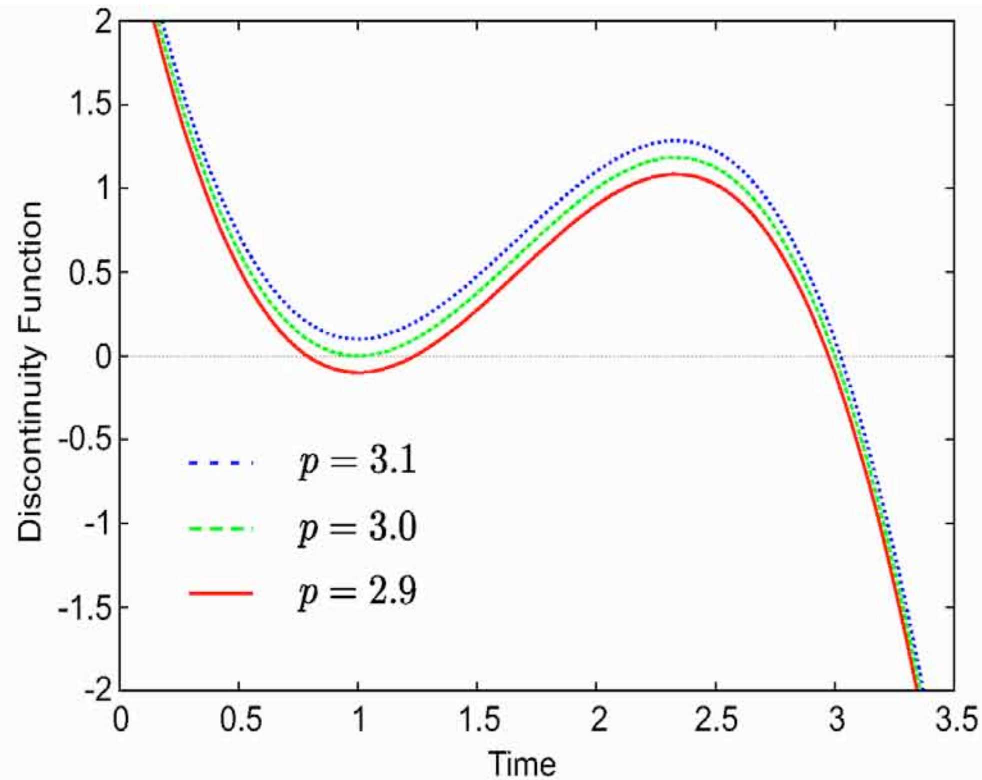
[1] S. Galán, W. F. Feehery and P. I. Barton. Parametric sensitivity functions for hybrid discrete/continuous systems. *Applied Numerical Mathematics*, **31**(1): 17-48, 1999.

Hybrid Sensitivities: Example

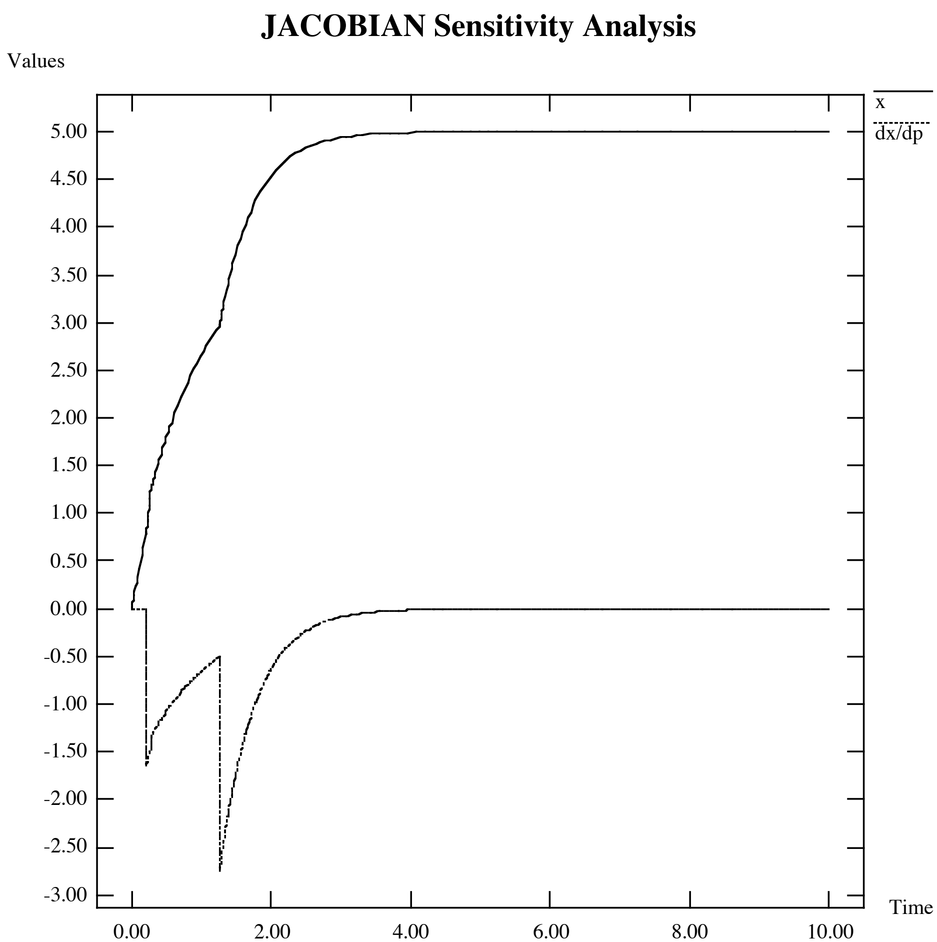


$$S_1 : \begin{cases} \frac{\mathrm{d}x^{(1)}}{\mathrm{d}t} = 4 - x^{(1)} \\ L_2^{(1)} : -(x^{(1)})^3 + 5(x^{(1)})^2 - 7x^{(1)} + p \leq 0 \\ T_2^{(1)} = x^{(2)} - x^{(1)} = 0 \end{cases}$$

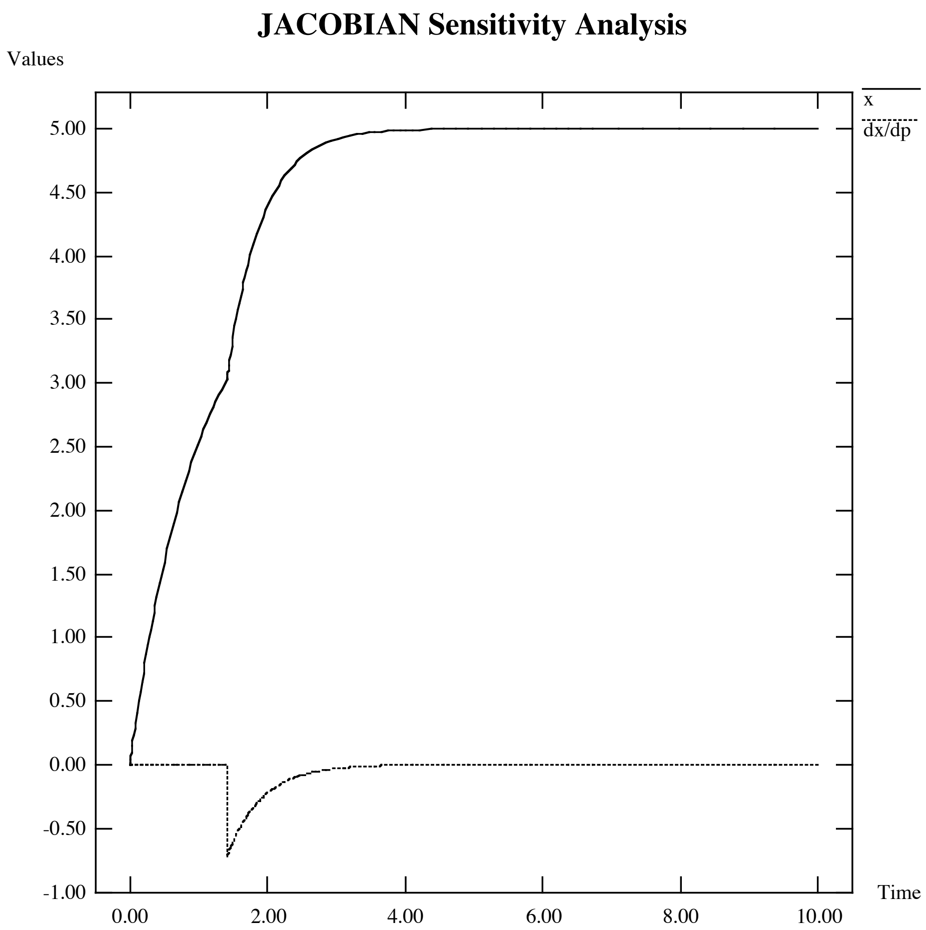
$$S_2 : \begin{cases} \frac{\mathrm{d}x^{(2)}}{\mathrm{d}t} = 10 - 2x^{(2)} \\ L_1^{(2)} : -(x^{(2)})^3 + 5(x^{(2)})^2 - 7x^{(2)} + p > 0 \\ T_1^{(2)} = x^{(1)} - x^{(2)} = 0 \end{cases}$$



Hybrid Sensitivities: Example



p=2.9

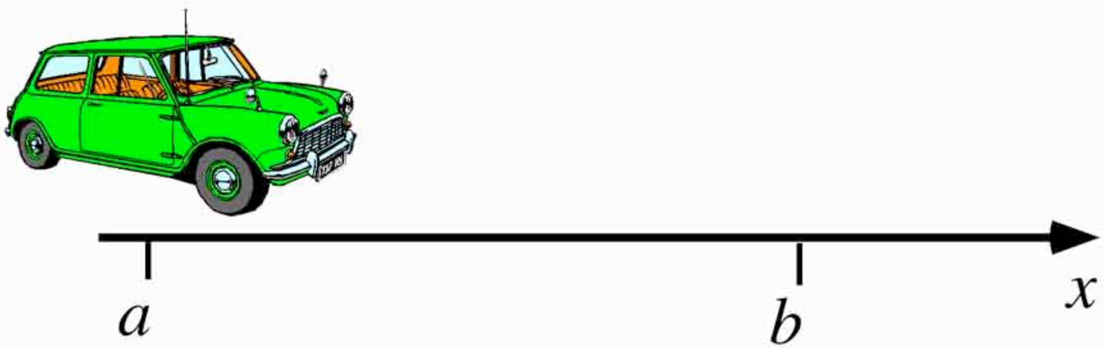


p=3.1

OPTIMIZATION

Dynamic Optimization
Car Example

- find acceleration profile that will get car from *a* to *b* in minimum time:



$$\min_{u(t)} \quad t_f$$

$$\dot{v} = u(t)$$

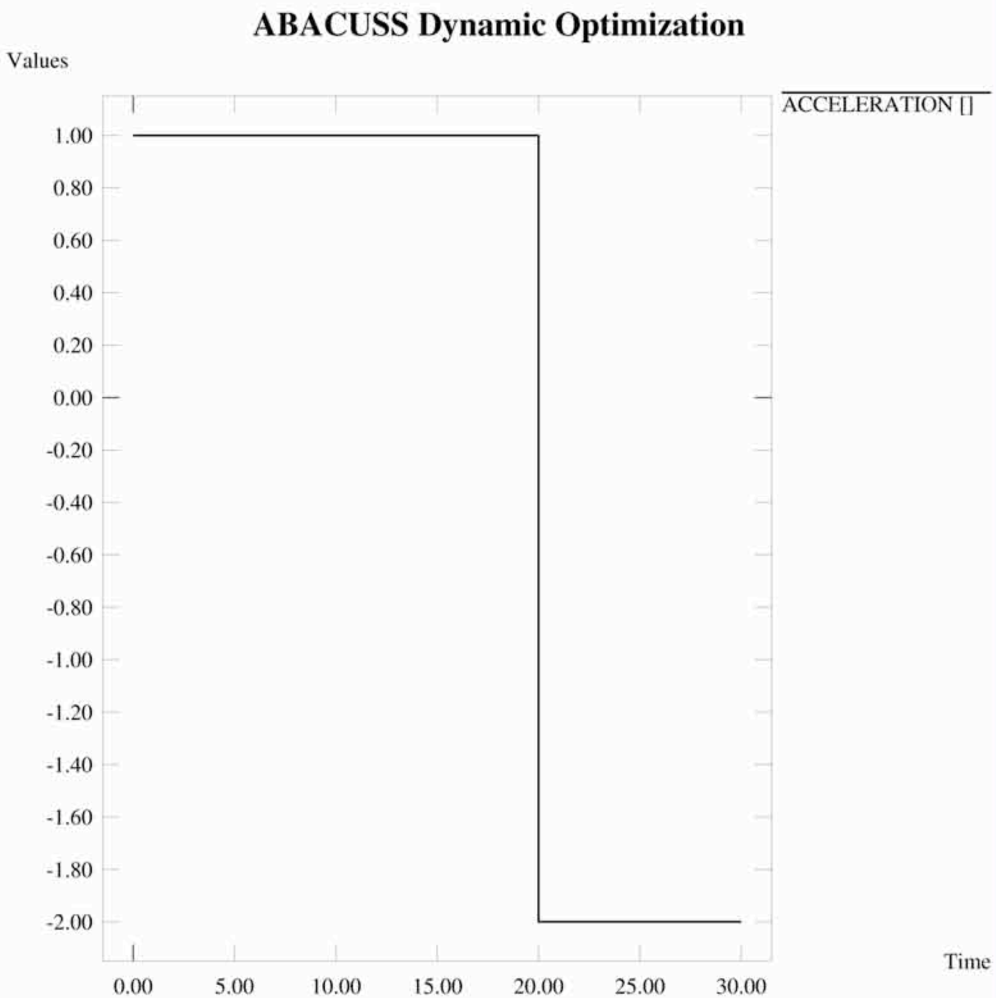
$$\dot{x} = v$$

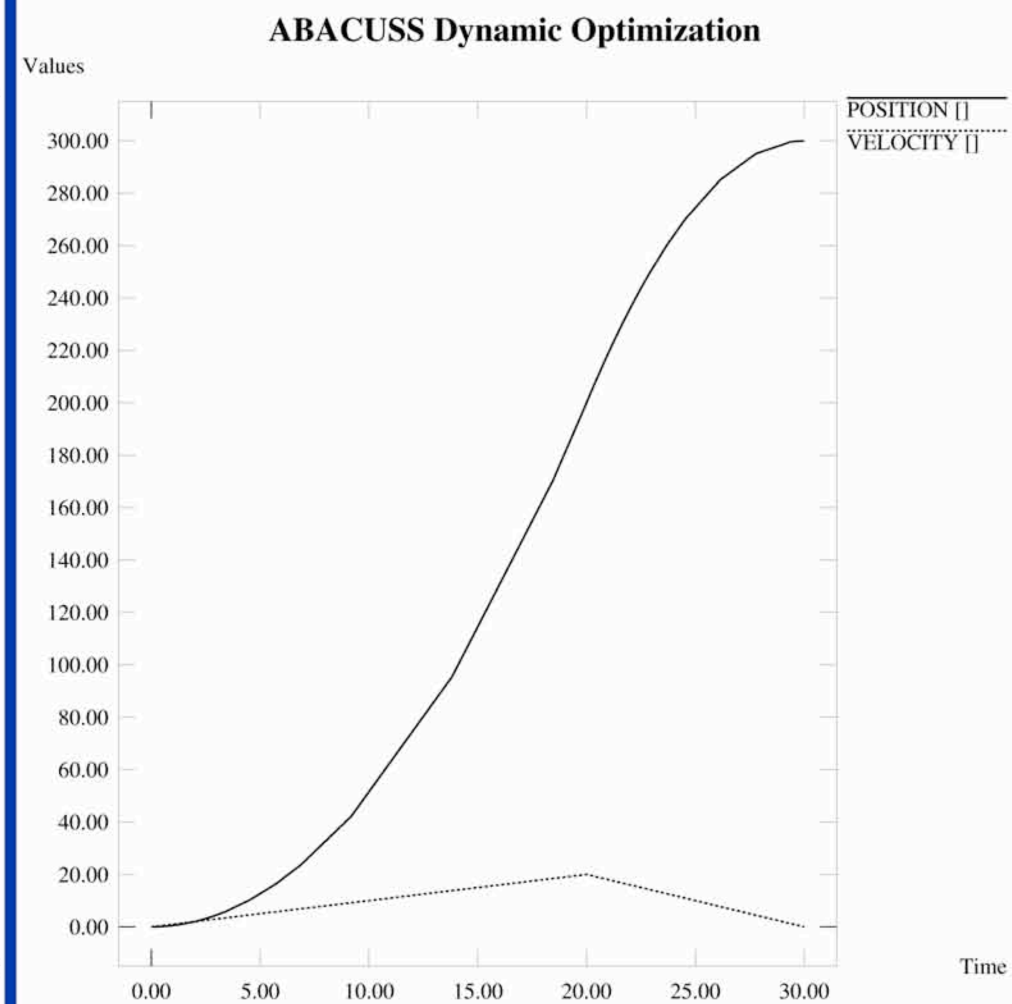
$$x(t_0) = a, v(t_0) = 0$$

$$x(t_f) = b, v(t_f) = 0$$

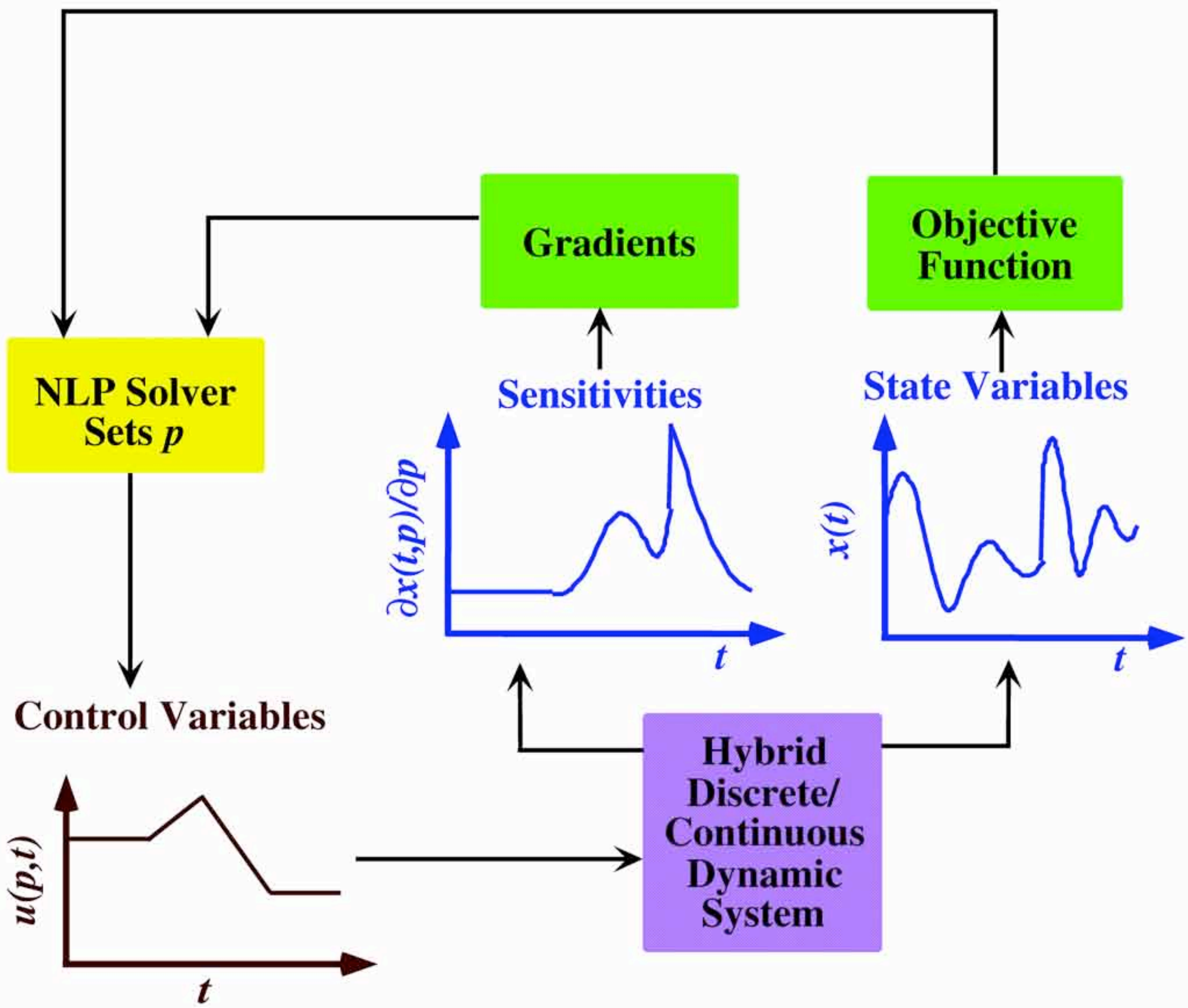
$$-d \leq u(t) \leq c$$

Car Example: Results





Control Parameterization



OPTIMAL MODE SEQUENCES

Epoch and Mode Sequences

$$x' = 2x + p$$

Mode 1

$$x' = x - 2p$$

Mode 2

Epoch and Mode Sequences

$$x' = 2x + p$$

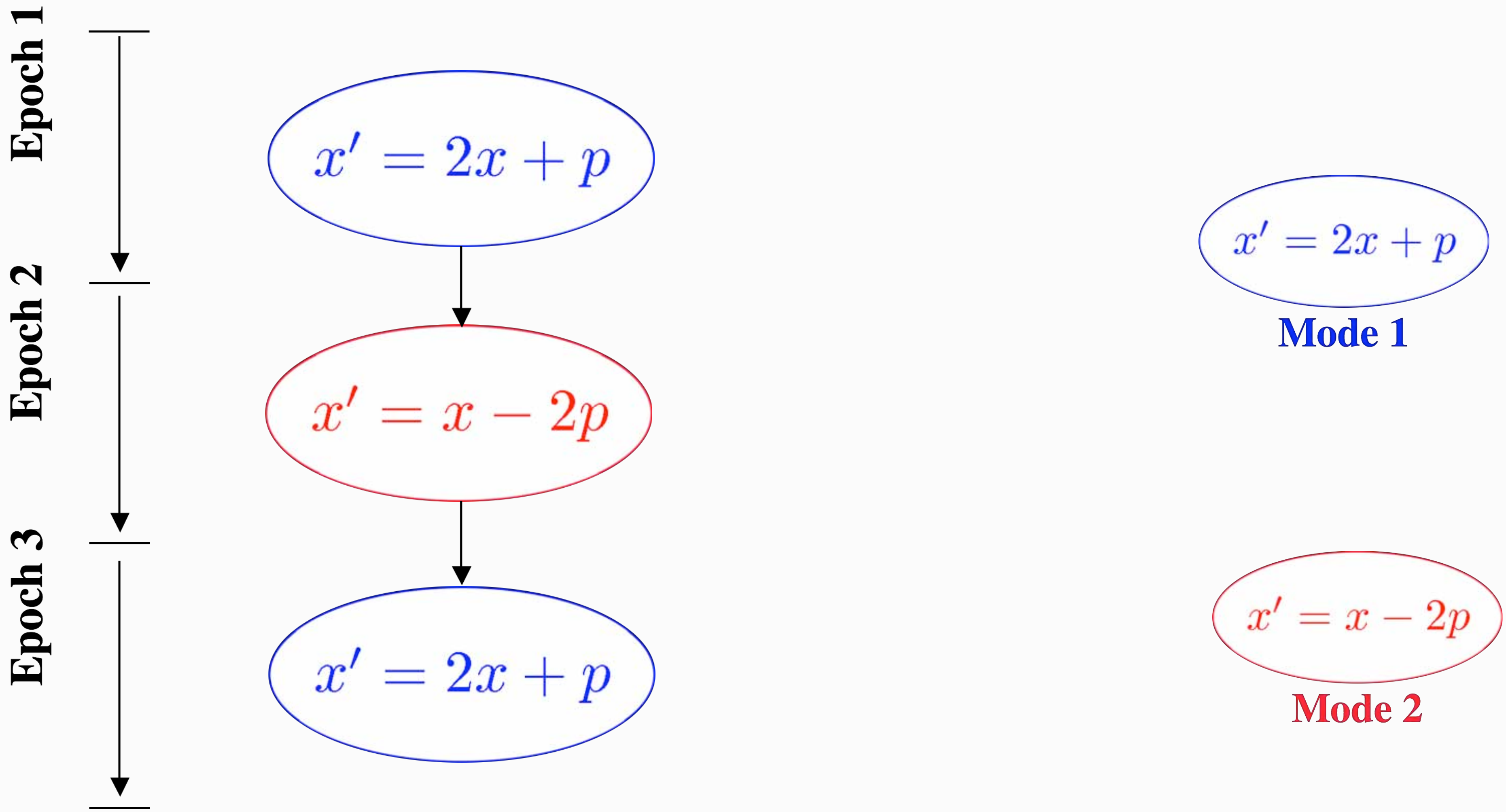
Mode 1

$$x' = x - 2p$$

Mode 2

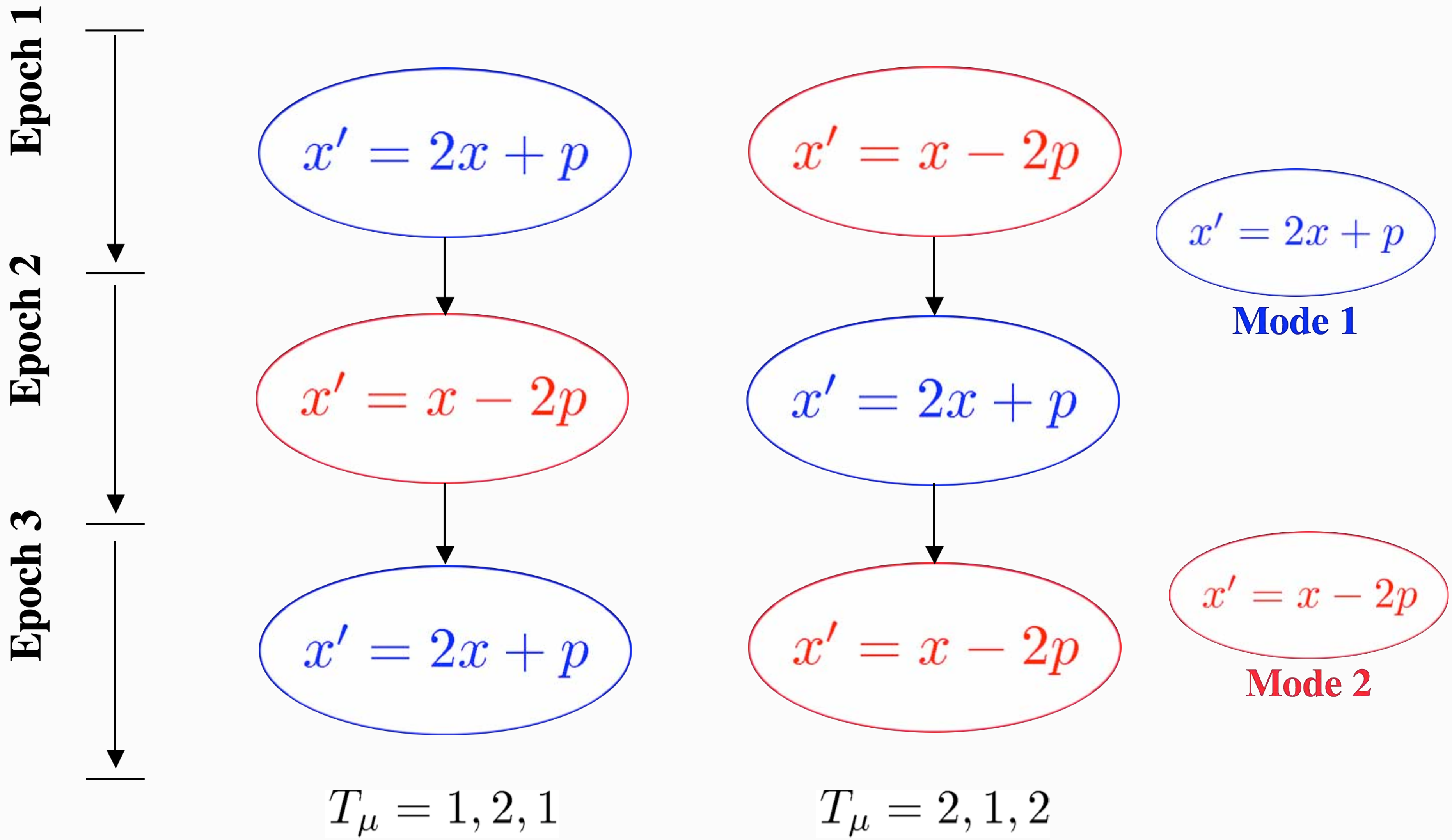
Epoch and Mode Sequences

Epoch and Mode Sequences

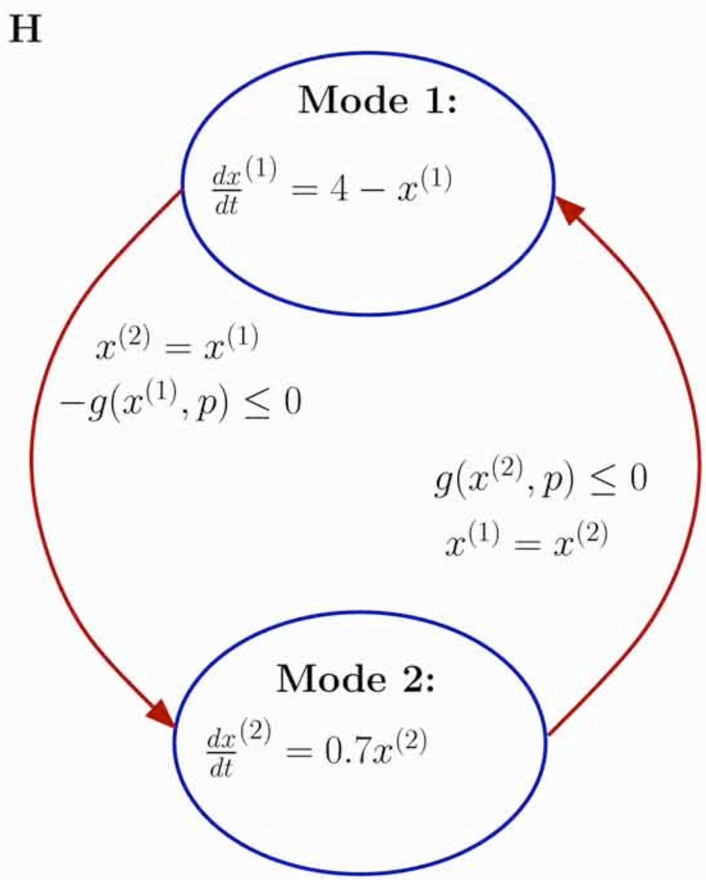


$$T_\mu = 1, 2, 1$$

Epoch and Mode Sequences



Nonsmoothness Example



$$g(x, p) = x^3 - 5x^2 + 7x - p$$

$$\min_p -x(p, t_f)$$

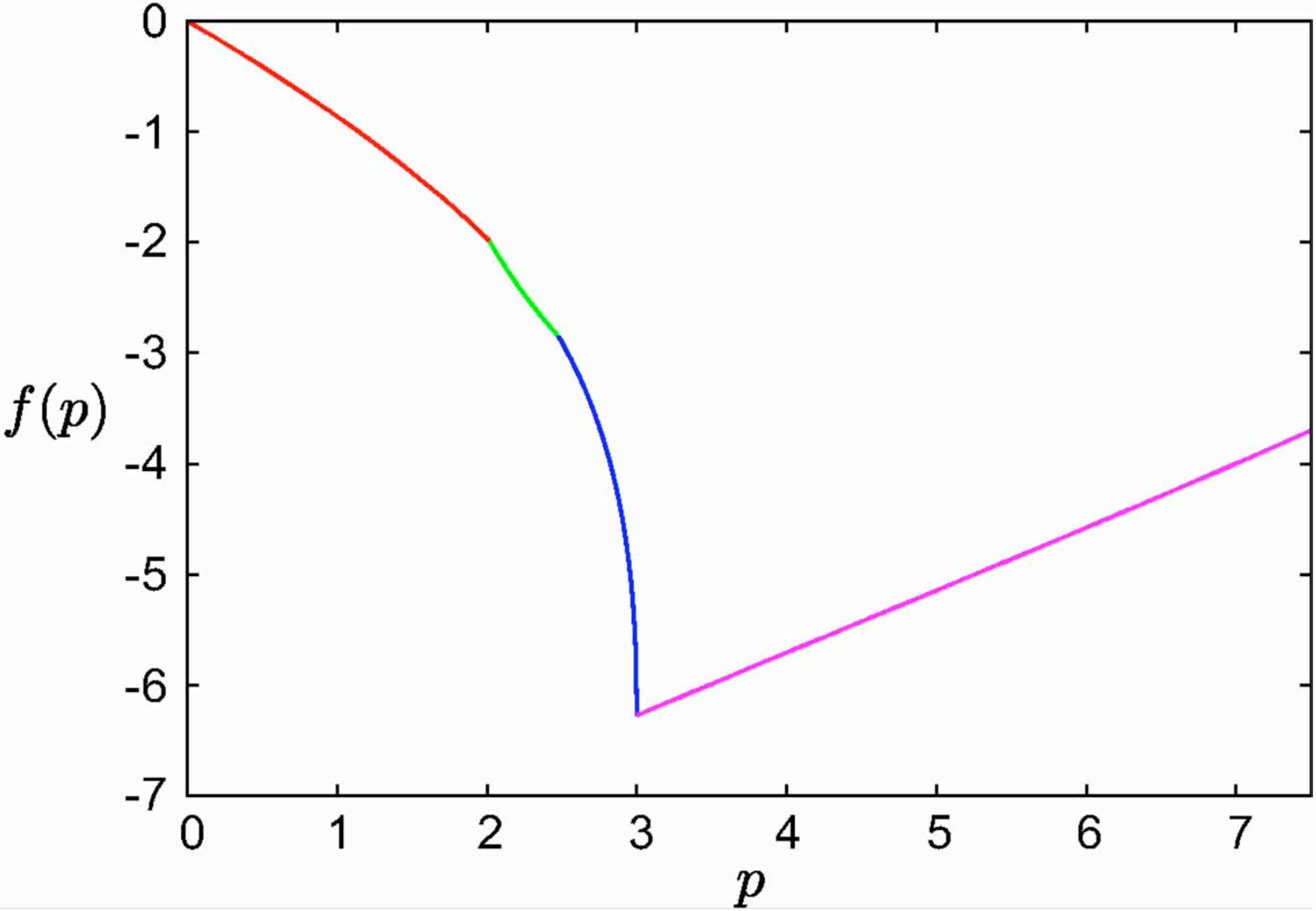
s. t.

$$p \in [0, 7.5]$$

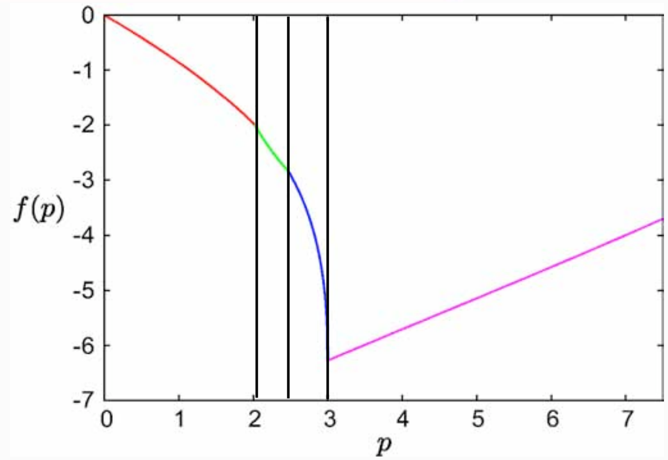
$$t_f = 3.0$$

$$x_0 = -4.0$$

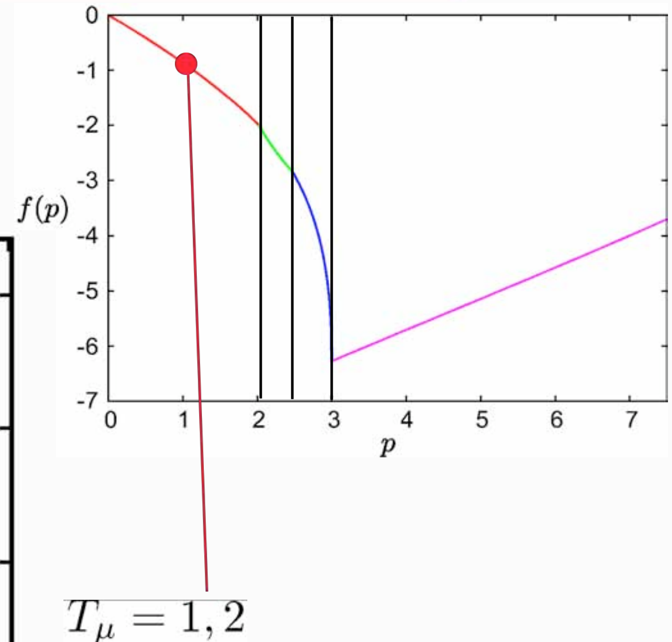
Nonsmoothness Example

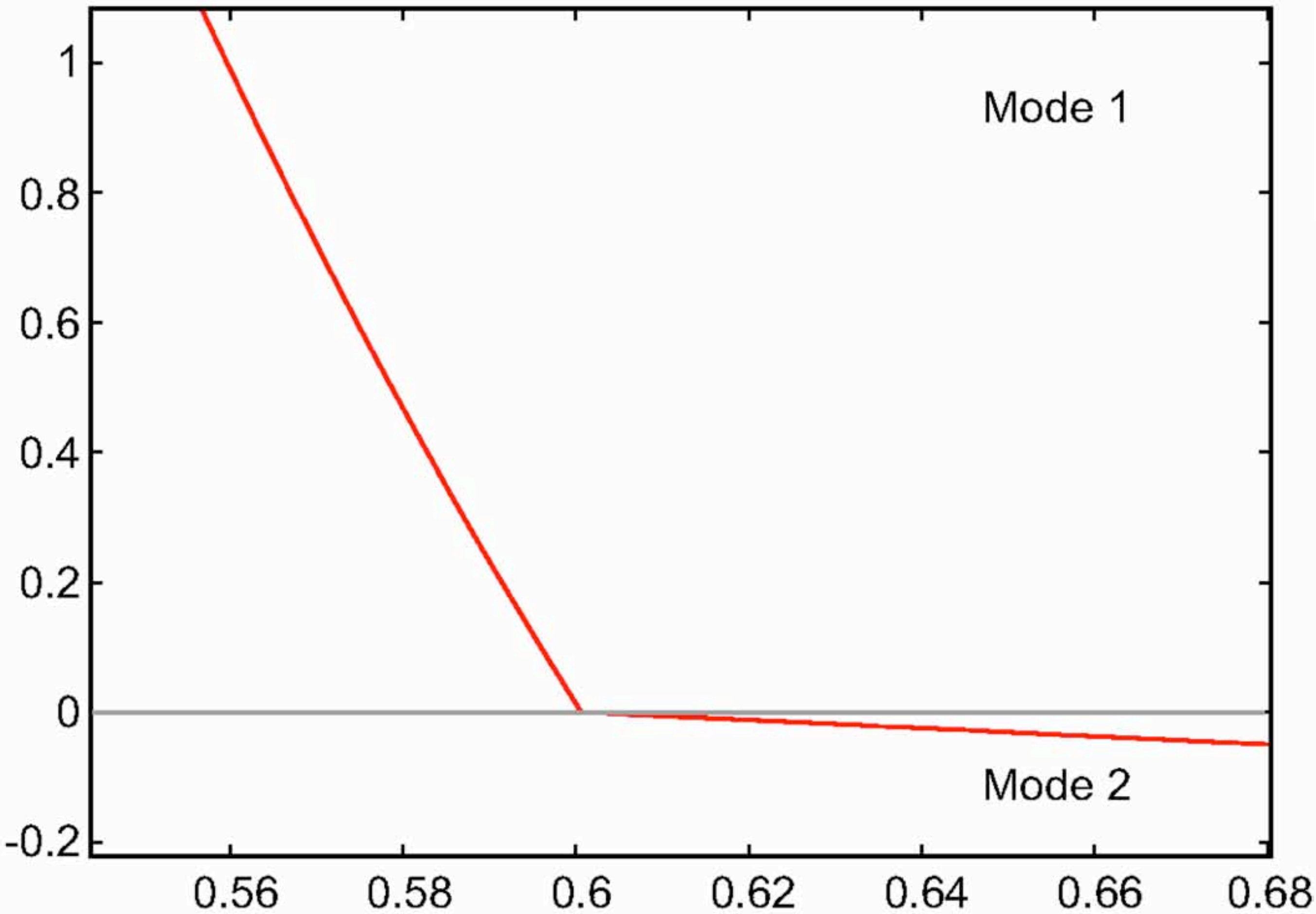


Nonsmoothness Example



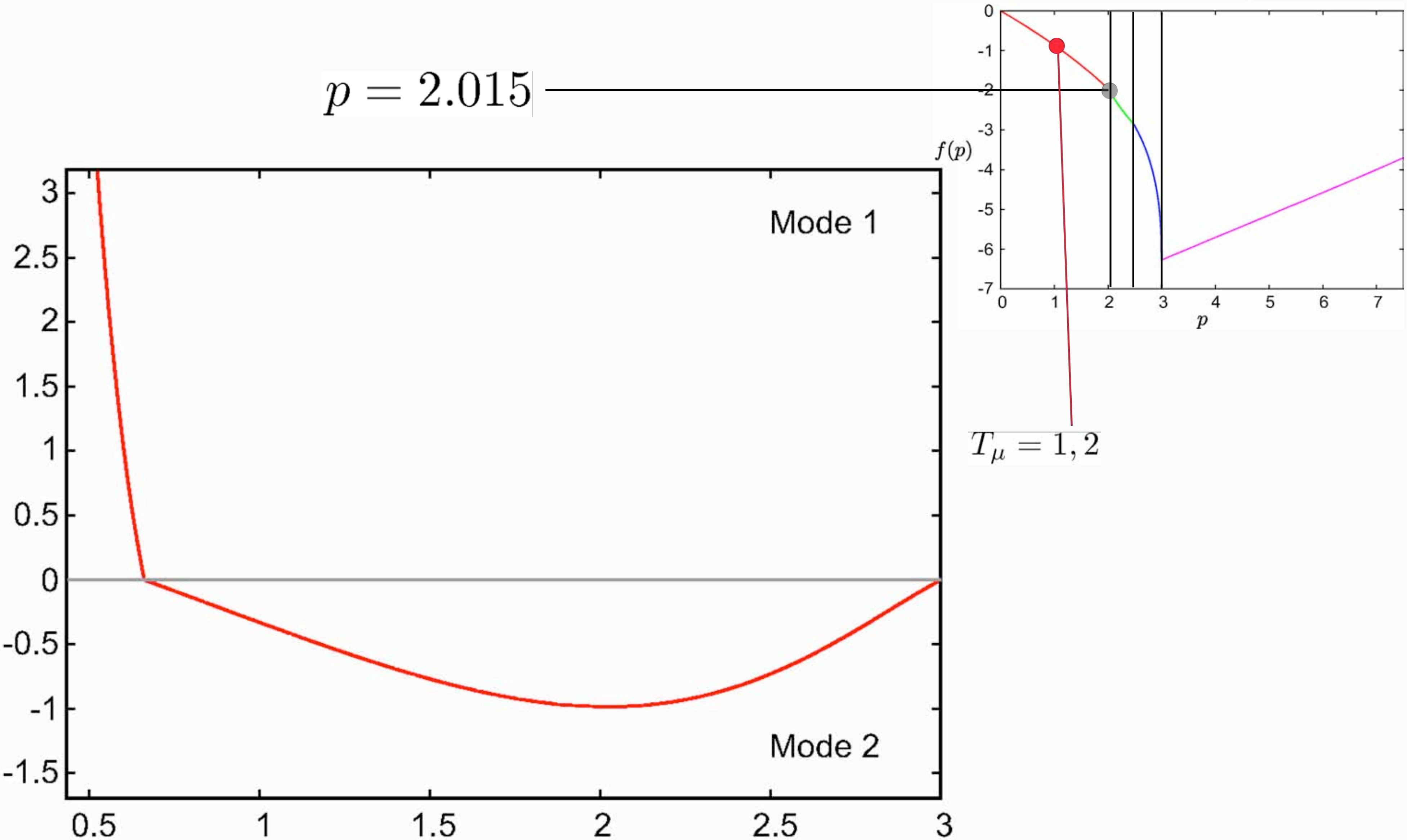
Nonsmoothness Example





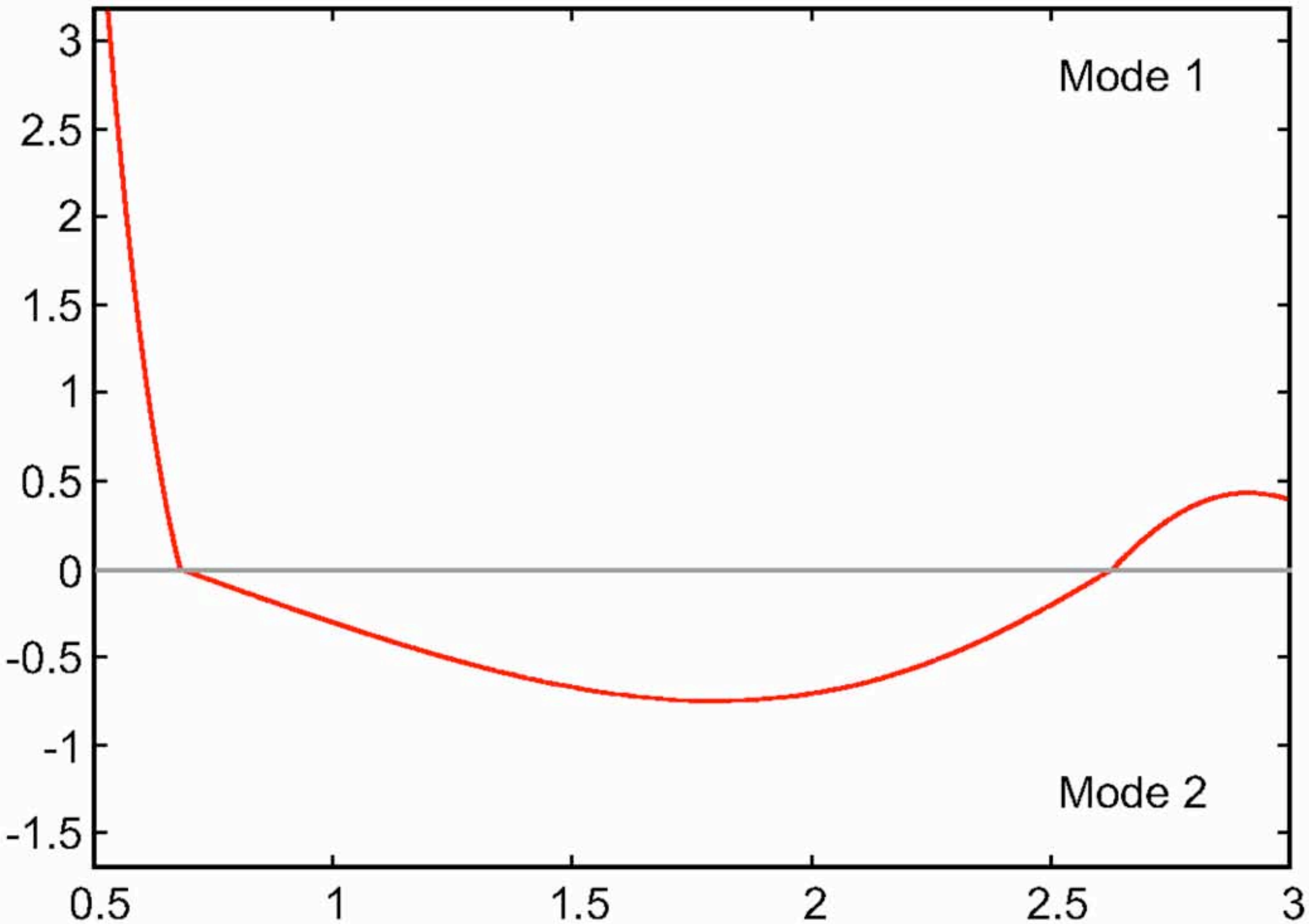
Discontinuity function $-x^3 + 5x^2 - 7x + p$

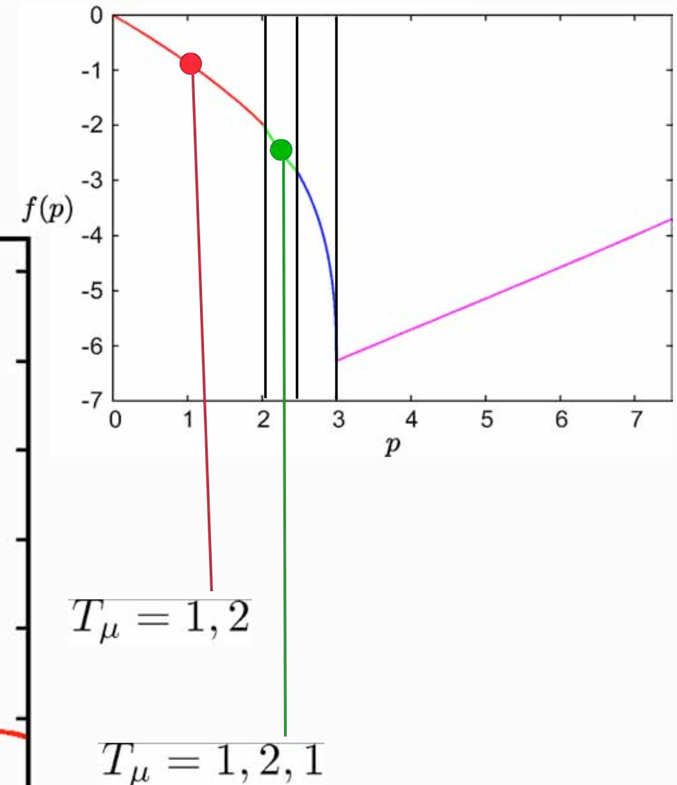
Nonsmoothness Example



Discontinuity function $-x^3 + 5x^2 - 7x + p$

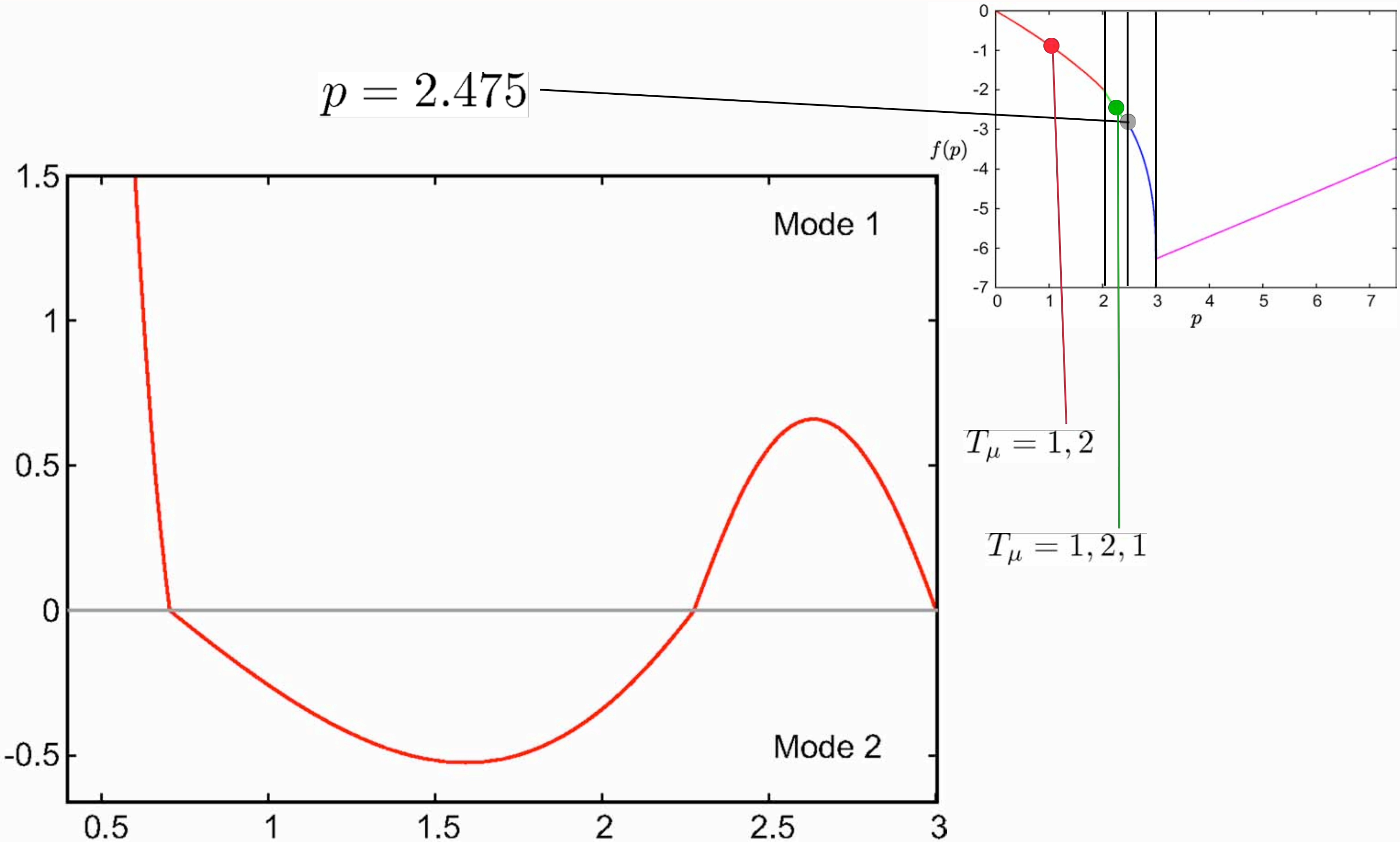
Nonsmoothness Example





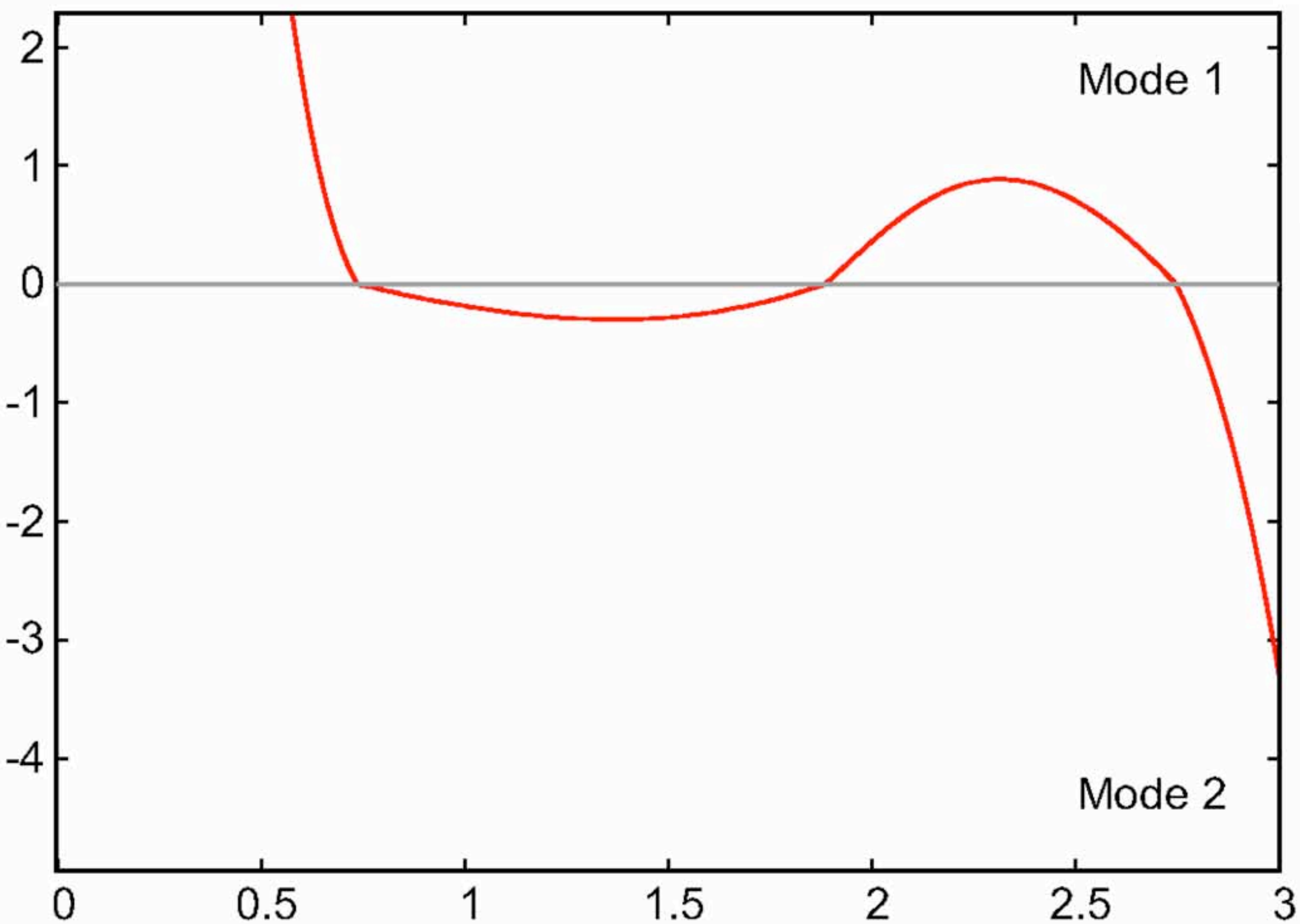
Discontinuity function $-x^3 + 5x^2 - 7x + p$

Nonsmoothness Example

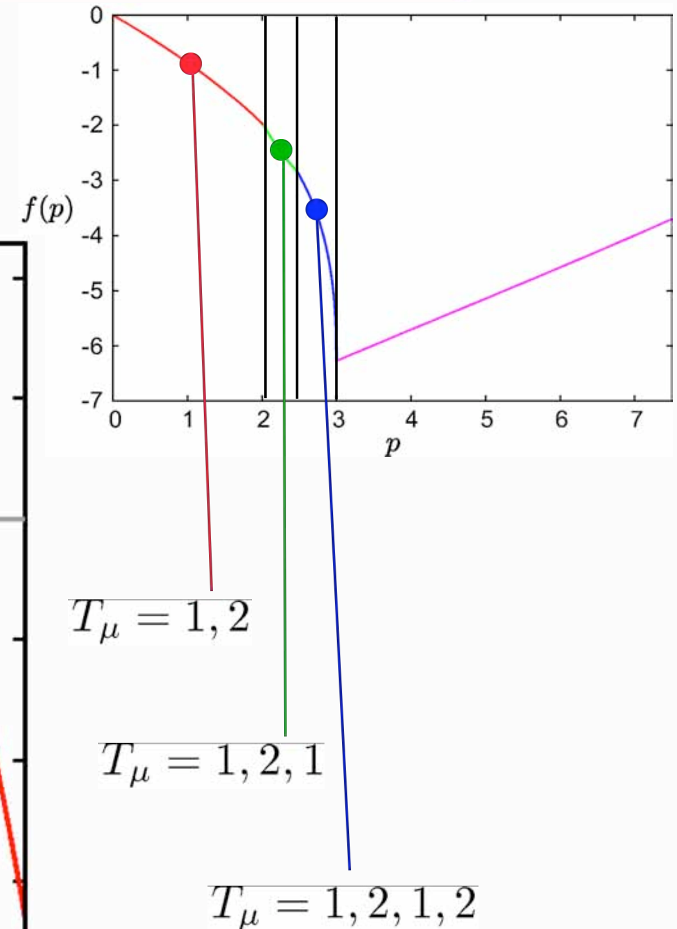


Discontinuity function $-x^3 + 5x^2 - 7x + p$

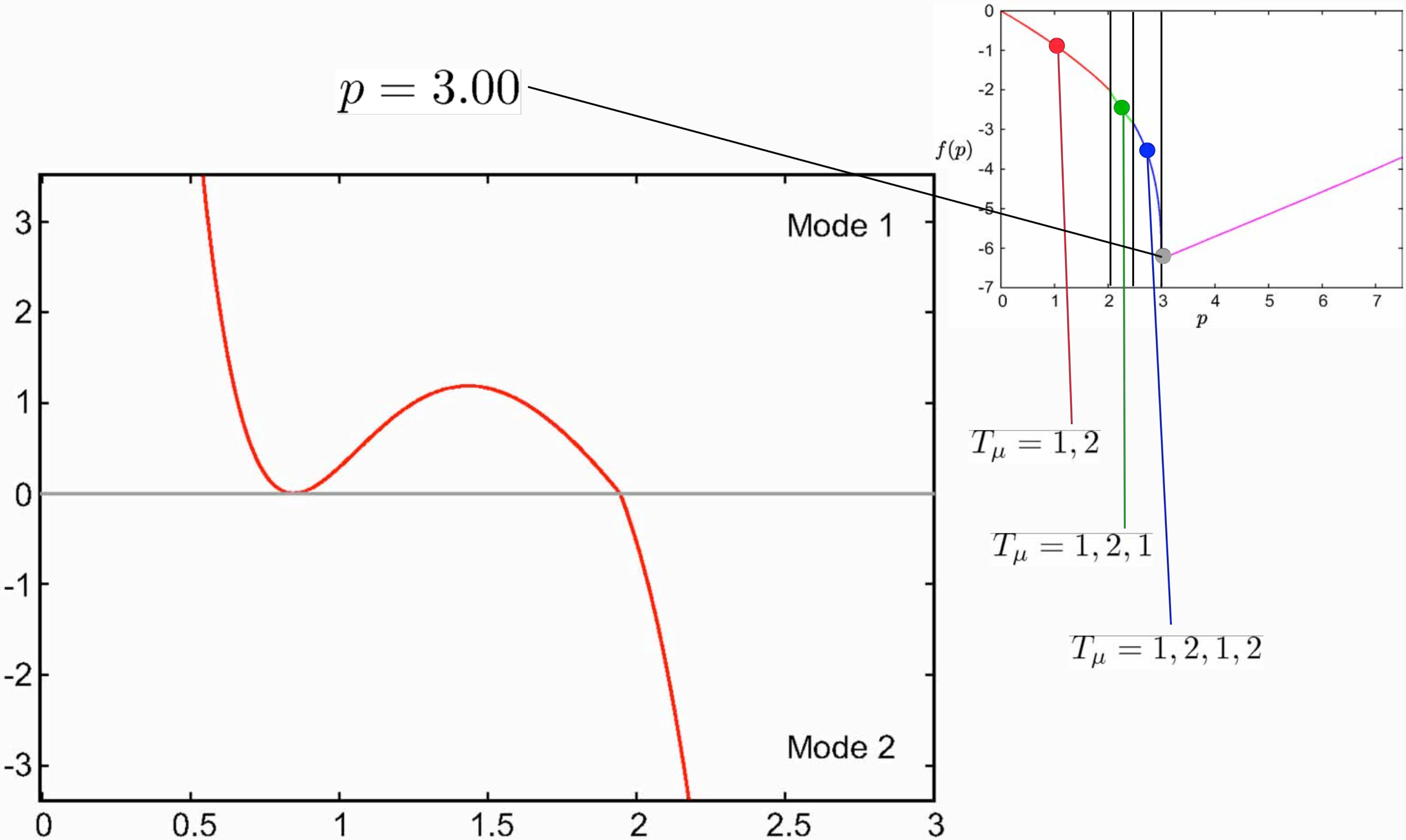
Nonsmoothness Example



Discontinuity function $-x^3 + 5x^2 - 7x + p$

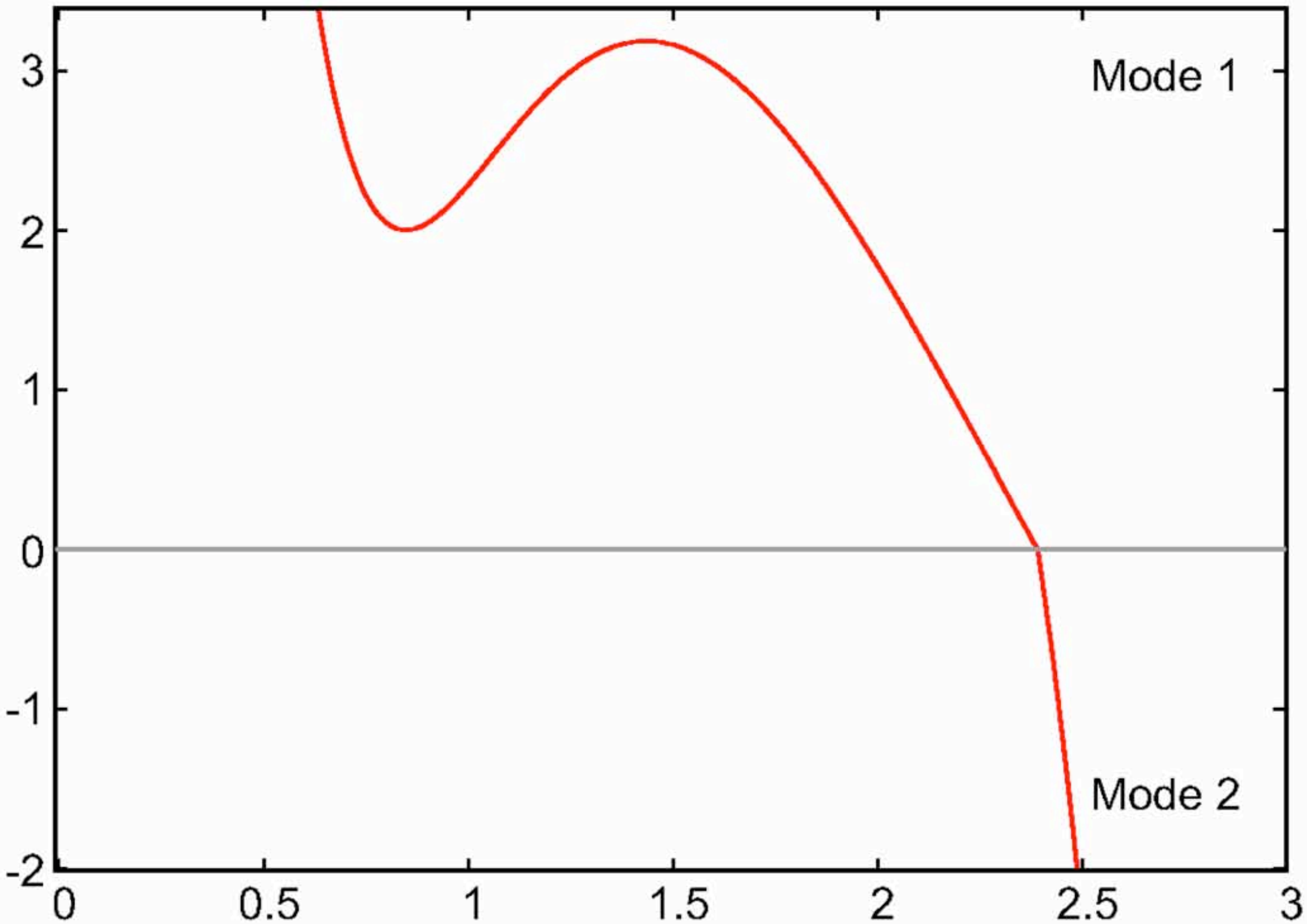


Nonsmoothness Example

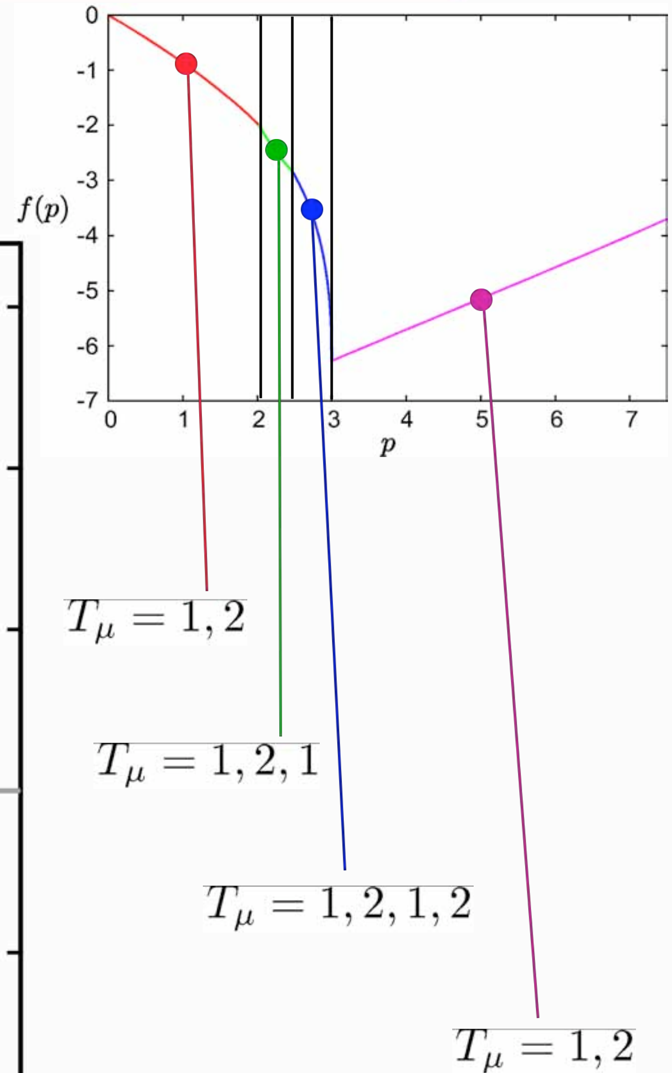


Discontinuity function $-x^3 + 5x^2 - 7x + p$

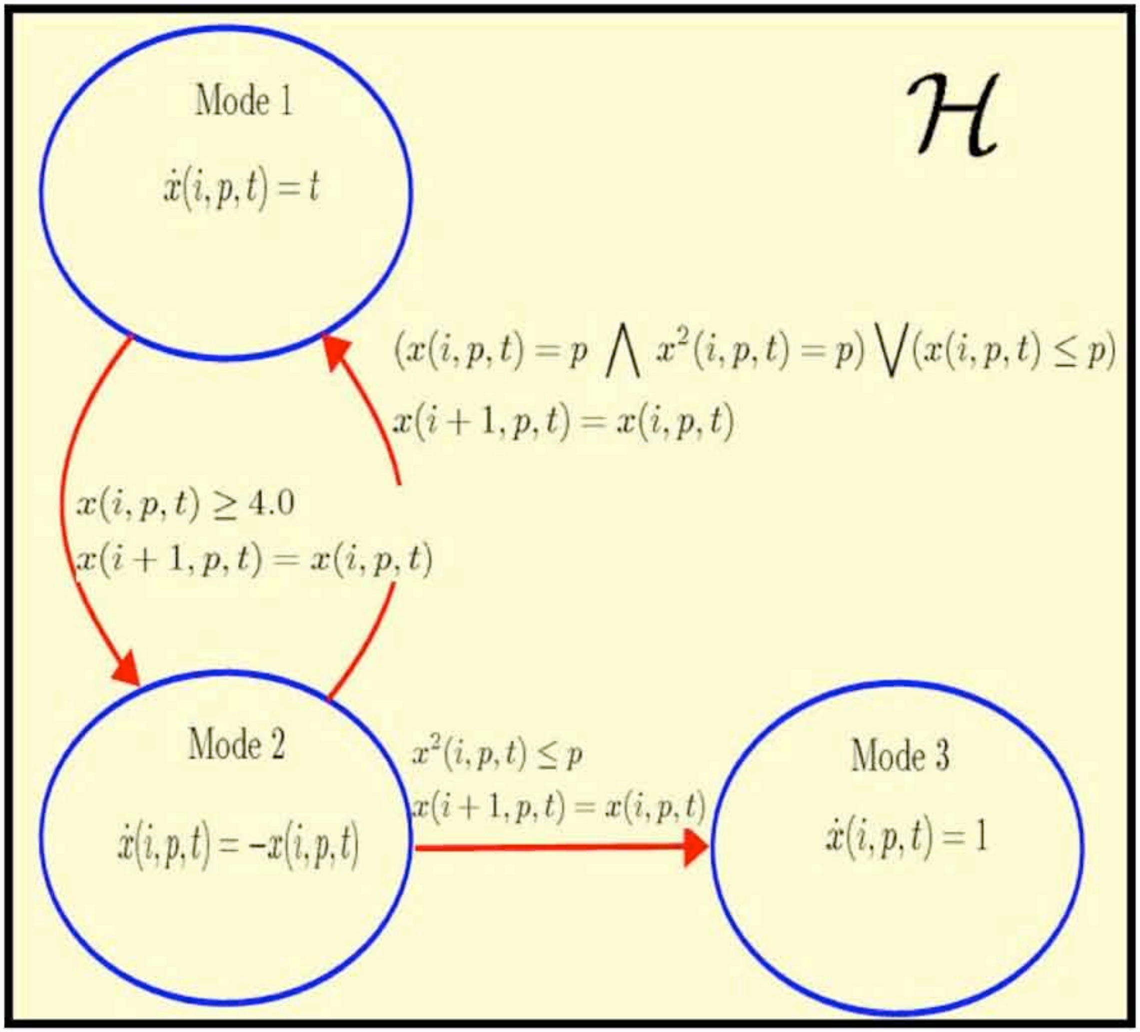
Nonsmoothness Example



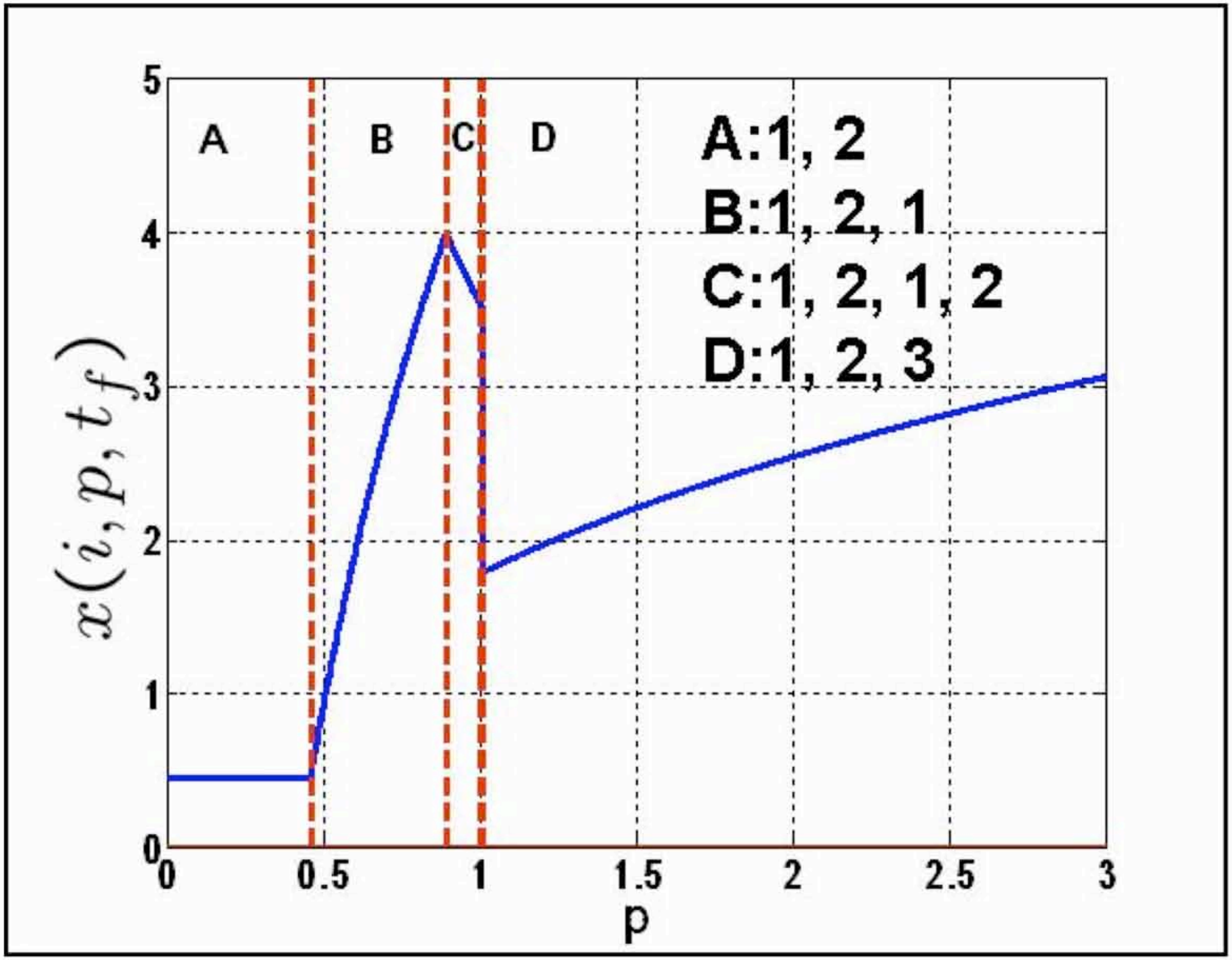
Discontinuity function $-x^3 + 5x^2 - 7x + p$



Discontinuous Example



Discontinuous Example



Classification of Optimization Problems with Hybrid Systems Embedded

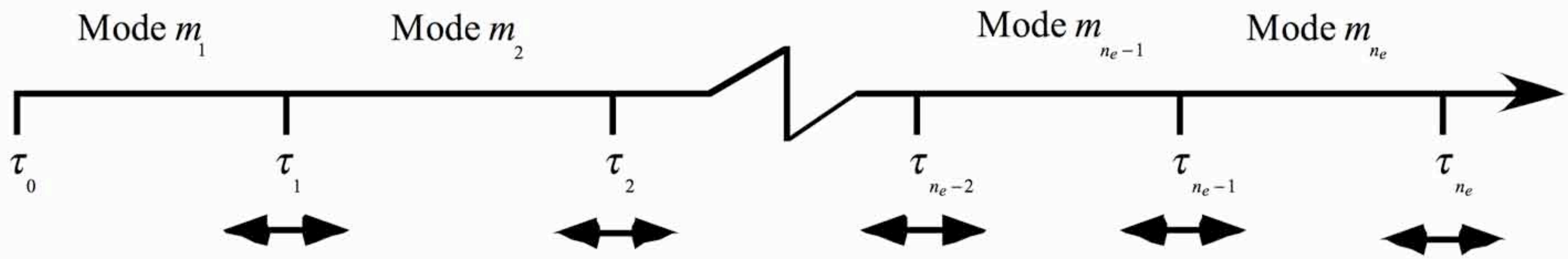
- Mode sequence does not vary - usually smooth
  - multistage dynamic optimization
- Continuous but nonsmooth hybrid systems
  - nonsmooth, mode sequence can vary
  - occurs even in classical ODE case with Lipschitz vector fields
- Discontinuous hybrid systems

Finding the Optimal Mode Sequence

- Gradient-based control parameterization for hybrid systems with autonomous transitions will often fail
  - Nonsmoothness and/or discontinuities when mode sequence varies
- Finding the optimal mode sequence
  - Designing process operations such as start-up, shut-down, etc.
  - Design and formal verification of embedded systems
  - Global optimization needed for formal verification
- Alternatives:
  - Nonsmooth approaches
    - Stochastic search
    - Deterministic methods
  - Mixed-integer optimization reformulation

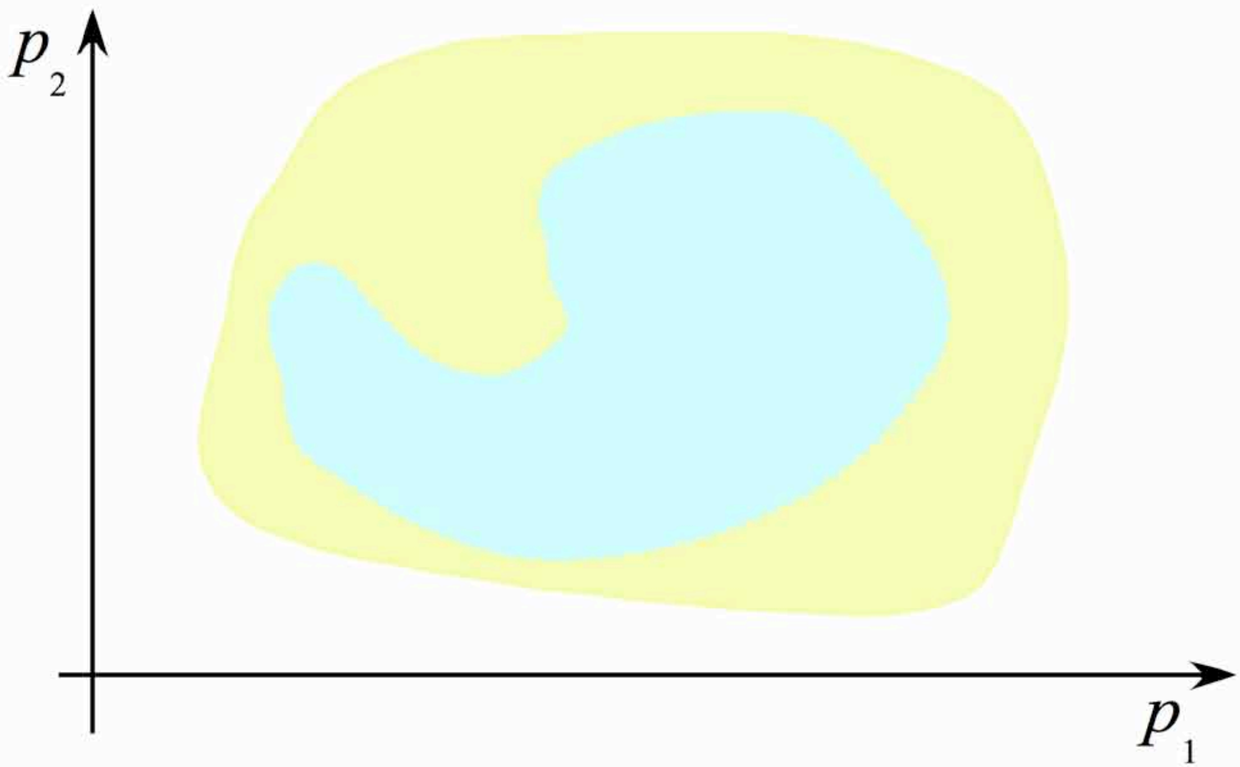
MULTISTAGE OPTIMIZATION

Multistage Optimization Lock the Mode Sequence



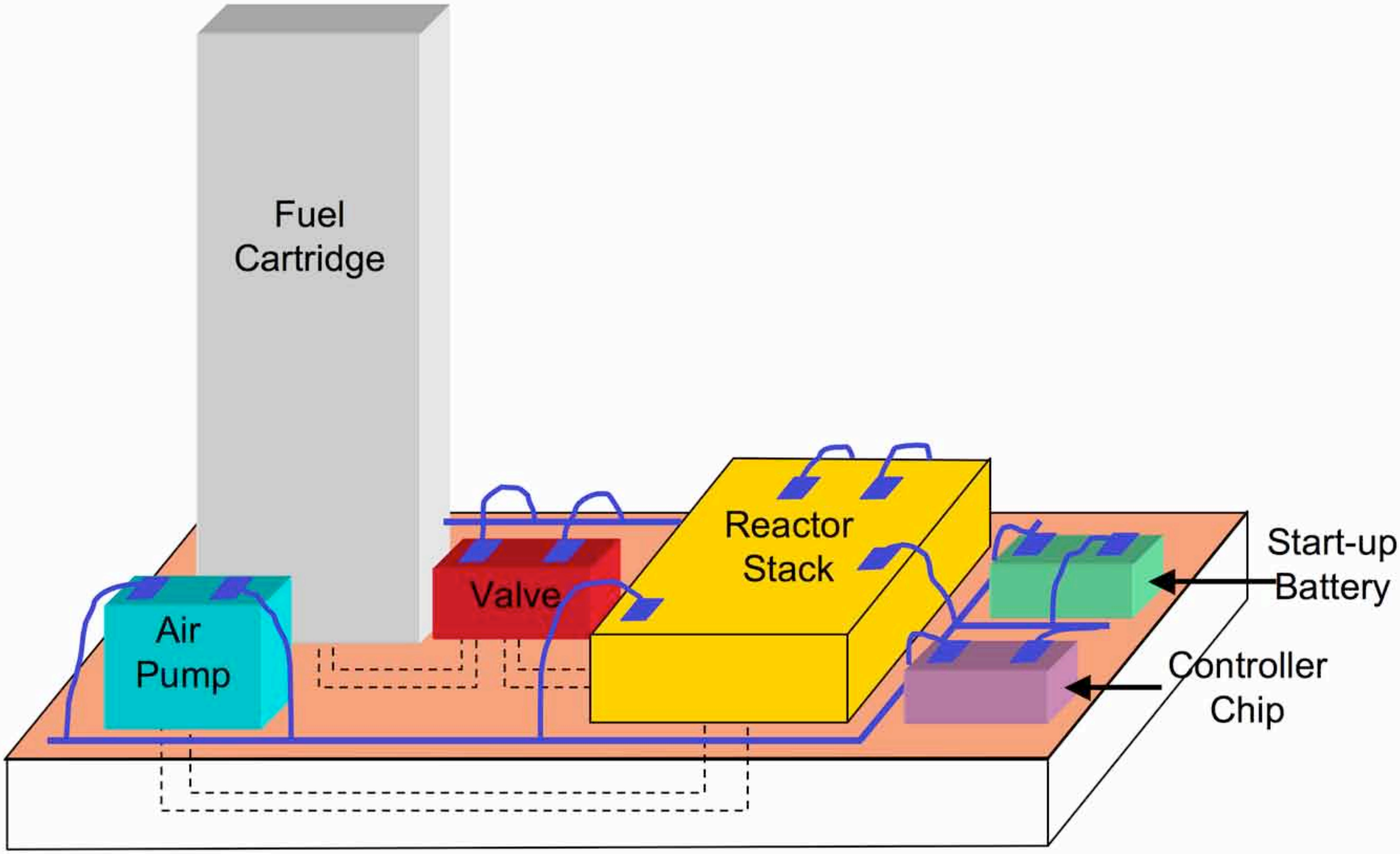
- Known number of epochs, known mode sequence
- Lock the mode sequence, event times as explicit decision variables in the optimization
- Point and path constraints enforce correct transition conditions at event times
- Smooth optimization problem, even when jumps in continuous states
- Care must be taken to compute correct jumps in sensitivities at event times

Relaxation Notion

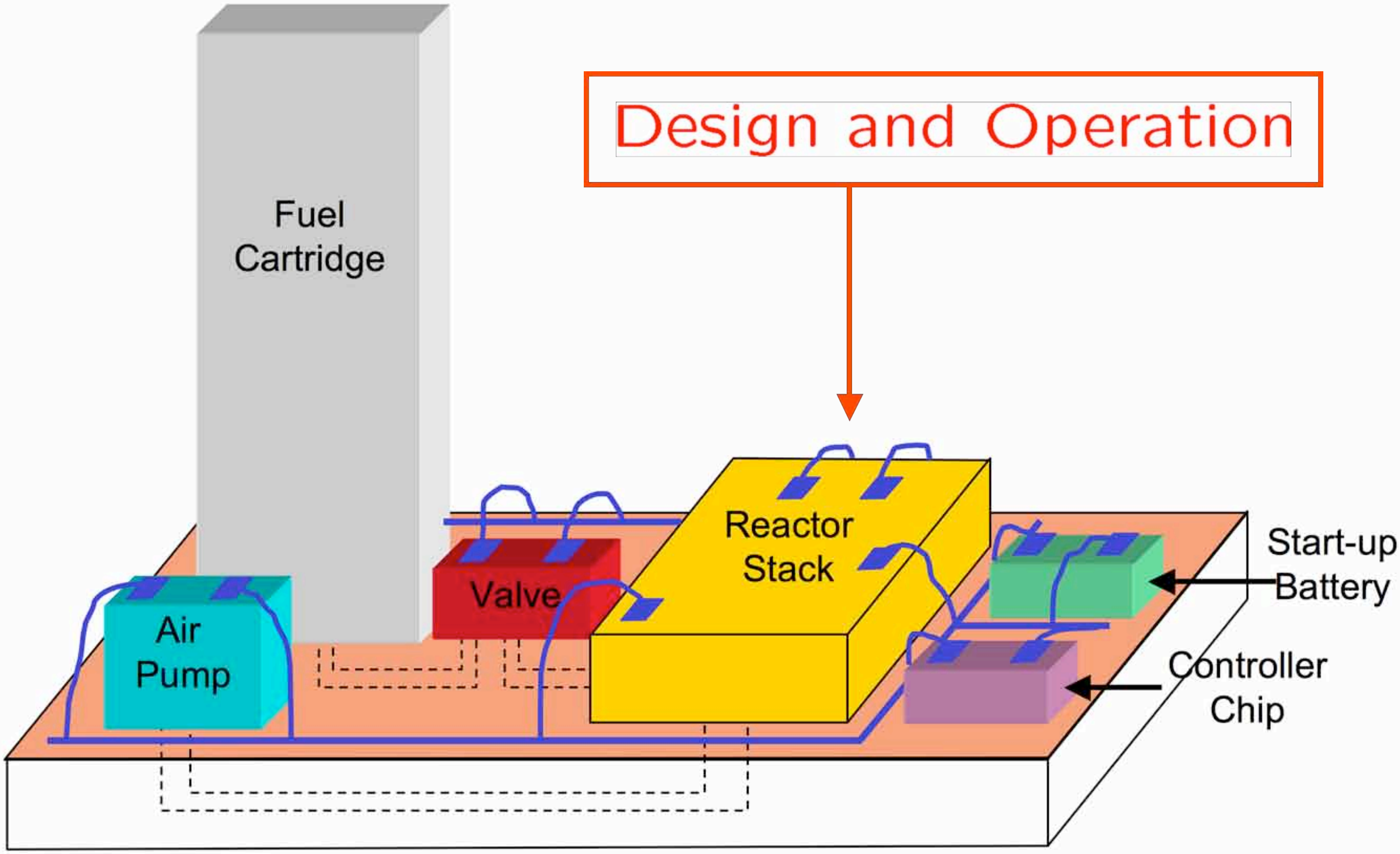


- Problem with locked mode sequence a relaxation of feasible set for given mode sequence
- Need to check feasibility w.r.t. all pending transitions a posteriori

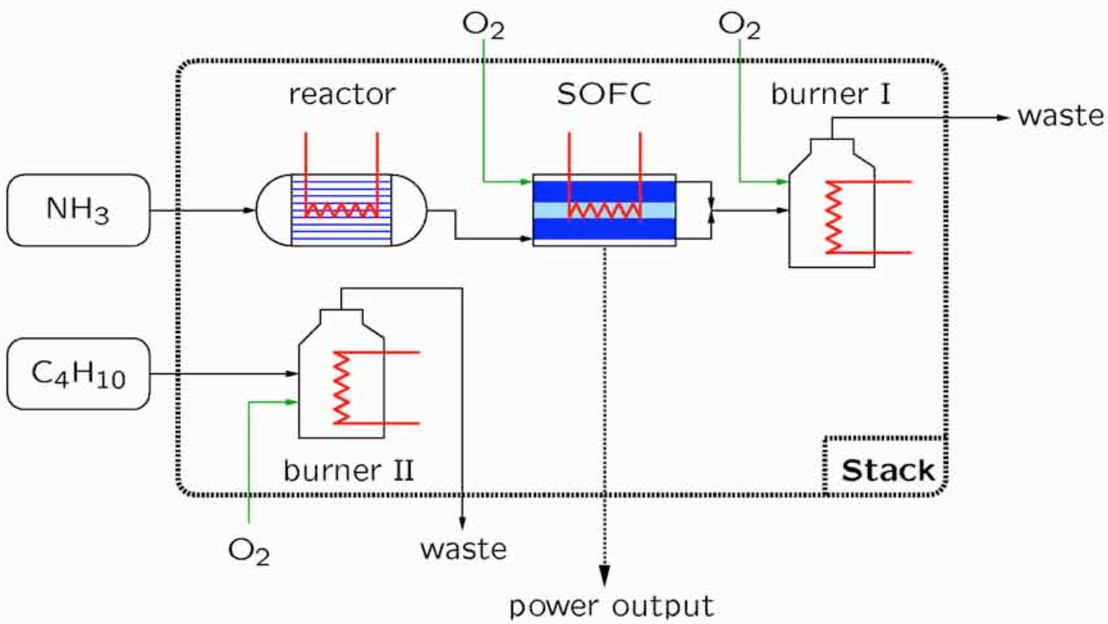
Micro Power Generation Concept



Micro Power Generation Concept



Case Study



Ammonia line:

1. **Reactor:** Production of H_2 from NH_3 catalytic decomposition
2. **SOFC:** Production of electric power from the electrochemical conversion H_2
3. **Burner I:** Heat generation from catalytic oxidation of residual H_2 and NH_3

Butane line:

4. **Burner II:** Heat generation from catalytic oxidation of Butane

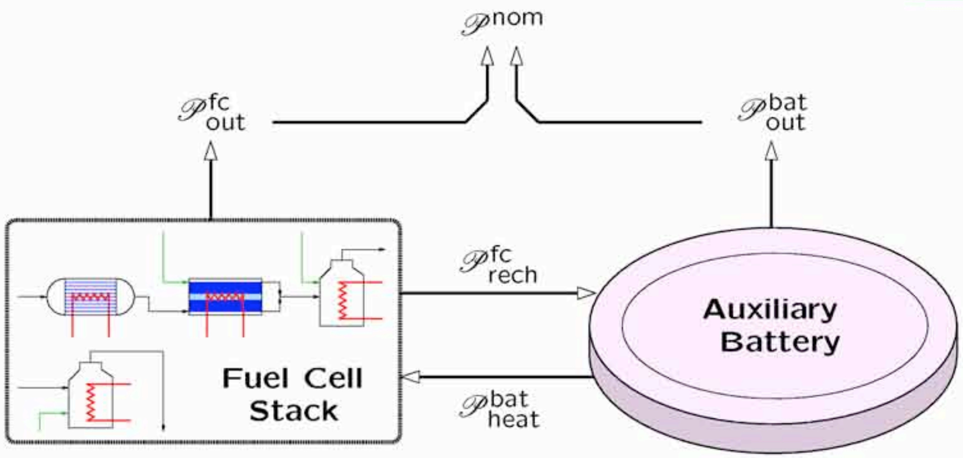
Optimal Start-up Problem

Transient Model:

- partial differential-algebraic equations
- multiple time scales: QSS approximation

Objective:

- Minimize the masses of fuels (ammonia, butane) and battery needed to complete the start-up phase



Control variables:

- fuel cell and battery voltages
- utilities flow rates
  e.g., NH_3, C_4H_{10}, air, *etc.*

Time horizon

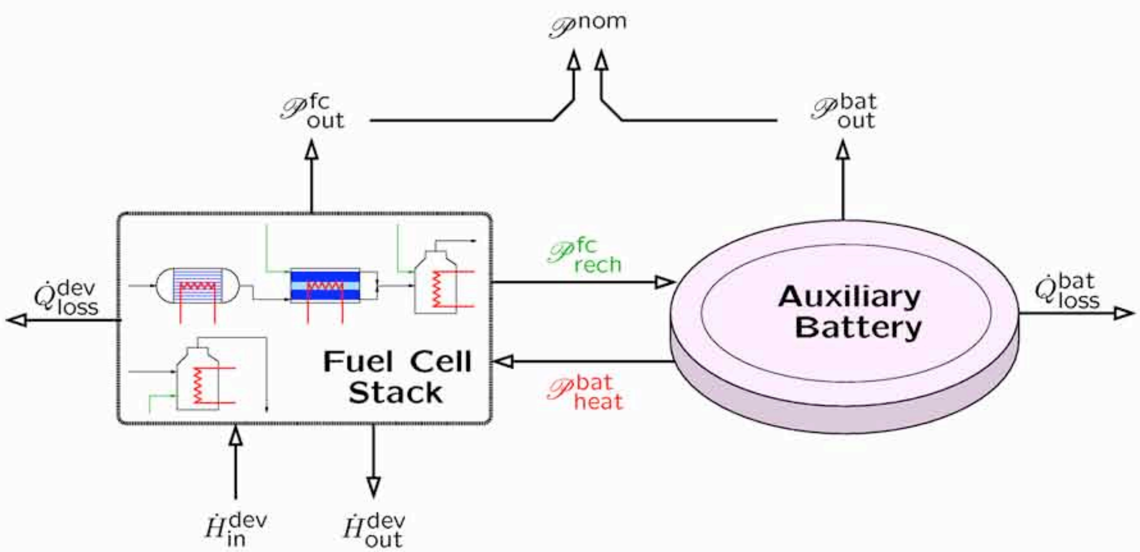
Design decision variables:

- initial mass of fuels
  e.g., NH_3, C_4H_{10}
- battery capacity/mass
- unit volumes
  e.g., reactor, fuel cell, burner, *etc.*

Constraints:

- battery fully recharged at terminal time
- meet nominal power demand at any time,
  e.g., 0.1 W–50 W
- satisfy maximal temperature and rate of heating (material constraints)

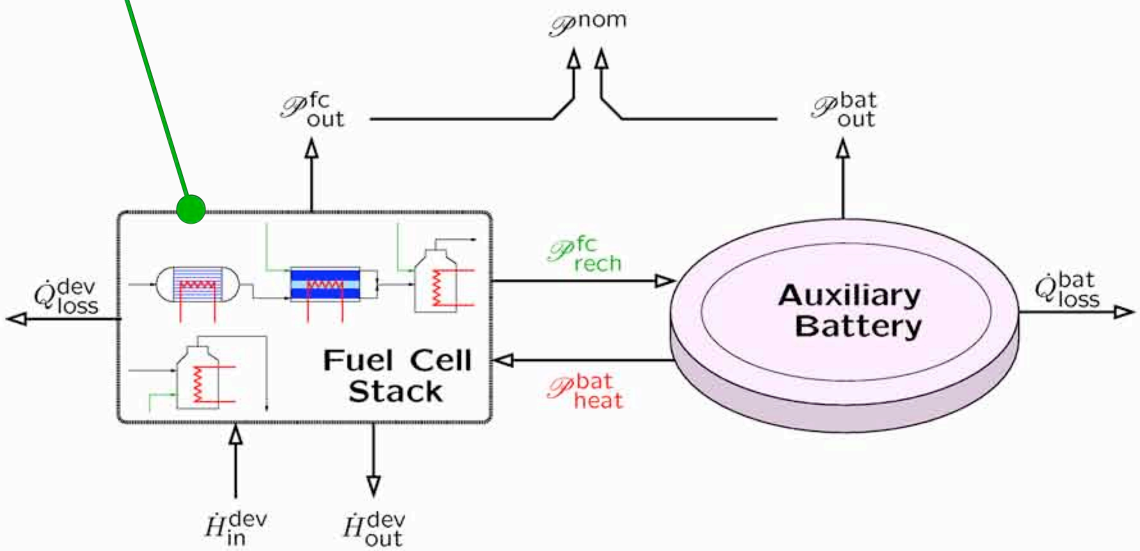
Transient Model Formulation



Transient Model Formulation

Mass and Species Conservation:

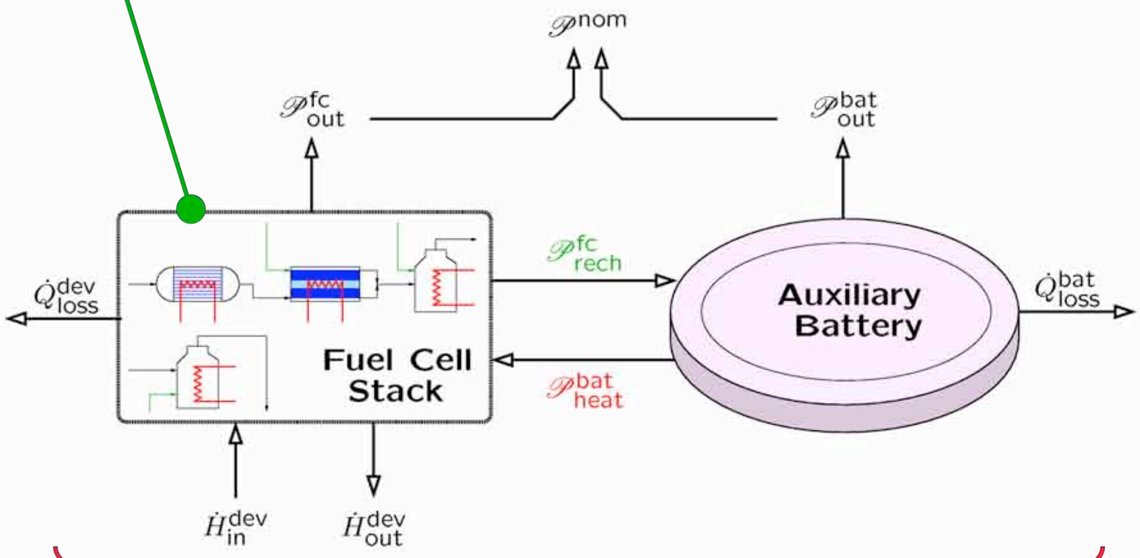
$$
\begin{aligned}
0 &= \frac{1}{V}\frac{\partial F}{\partial \xi} - \frac{1}{T}\frac{\partial T}{\partial t} - \frac{1}{\rho}\sum_{j=1}^{n_r}\sum_{i=1}^{n_c}\nu_{i,j}r_j \\
\frac{\partial y_i}{\partial t} &= -\frac{F}{V}\frac{\partial y_i}{\partial \xi} + \frac{1}{\rho}\sum_{j=1}^{n_r}\left[\nu_{i,j}r_j - y_i\sum_{k=1}^{n_c}\nu_{k,j}r_j\right],\ i = 1\ldots n_c \\
0 &= \rho - \frac{P}{\mathscr{R}T}
\end{aligned}
$$



Transient Model Formulation

Mass and Species Conservation:

$$
\begin{aligned}
0 &= \frac{1}{V}\frac{\partial F}{\partial \xi} - \frac{1}{T}\frac{\partial T}{\partial t} - \frac{1}{\rho}\sum_{j=1}^{n_r}\sum_{i=1}^{n_c} \nu_{i,j} r_j \\
\frac{\partial y_i}{\partial t} &= -\frac{F}{V}\frac{\partial y_i}{\partial \xi} + \frac{1}{\rho}\sum_{j=1}^{n_r}\left[\nu_{i,j} r_j - y_i \sum_{k=1}^{n_c} \nu_{k,j} r_j\right], \ i = 1 \ldots n_c \\
0 &= \rho - \frac{P}{\mathscr{R} T}
\end{aligned}
$$



Energy Conservation:

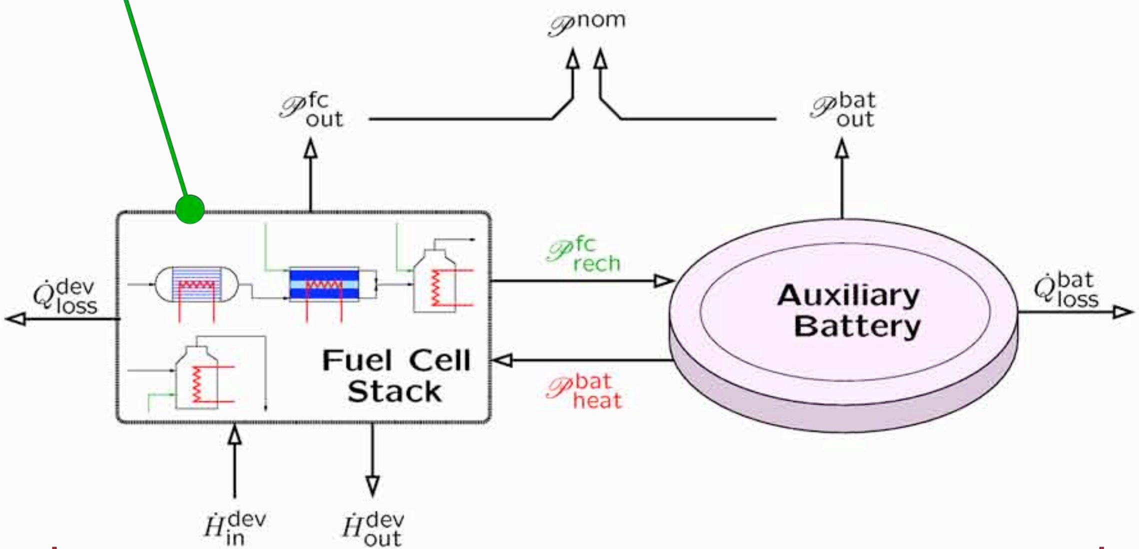
Battery: $\dot{\mathscr{E}}^{\text{bat}}(t) = -\mathscr{P}^{\text{bat}}_{\text{out}}(t) - \mathscr{P}^{\text{bat}}_{\text{heat}}(t) + \mathscr{P}^{\text{fc}}_{\text{rech}}(t) - \dot{Q}^{\text{bat}}_{\text{loss}}(t)$

Fuel Cell Stack: $V^{\text{dev}} c_p^{\text{dev}} \dot{T}(t) = \left[\dot{H}^{\text{dev}}_{\text{in}}(t) - \dot{H}^{\text{dev}}_{\text{out}}(t)\right] - \mathscr{P}^{\text{fc}}_{\text{out}}(t) - \mathscr{P}^{\text{fc}}_{\text{rech}}(t) + \mathscr{P}^{\text{bat}}_{\text{heat}}(t) - \dot{Q}^{\text{dev}}_{\text{loss}}(t)$

Transient Model Formulation

Mass and Species Conservation:

$$
\begin{aligned}
0 &= \frac{1}{V}\frac{\partial F}{\partial \xi} - \frac{1}{T}\frac{\partial T}{\partial t} - \frac{1}{\rho}\sum_{j=1}^{n_r}\sum_{i=1}^{n_c}\nu_{i,j} r_j \\
\frac{\partial y_i}{\partial t} &= -\frac{F}{V}\frac{\partial y_i}{\partial \xi} + \frac{1}{\rho}\sum_{j=1}^{n_r}\left[\nu_{i,j} r_j - y_i \sum_{k=1}^{n_c}\nu_{k,j} r_j\right], \; i = 1 \ldots n_c \\
0 &= \rho - \frac{P}{\mathscr{R}T}
\end{aligned}
$$



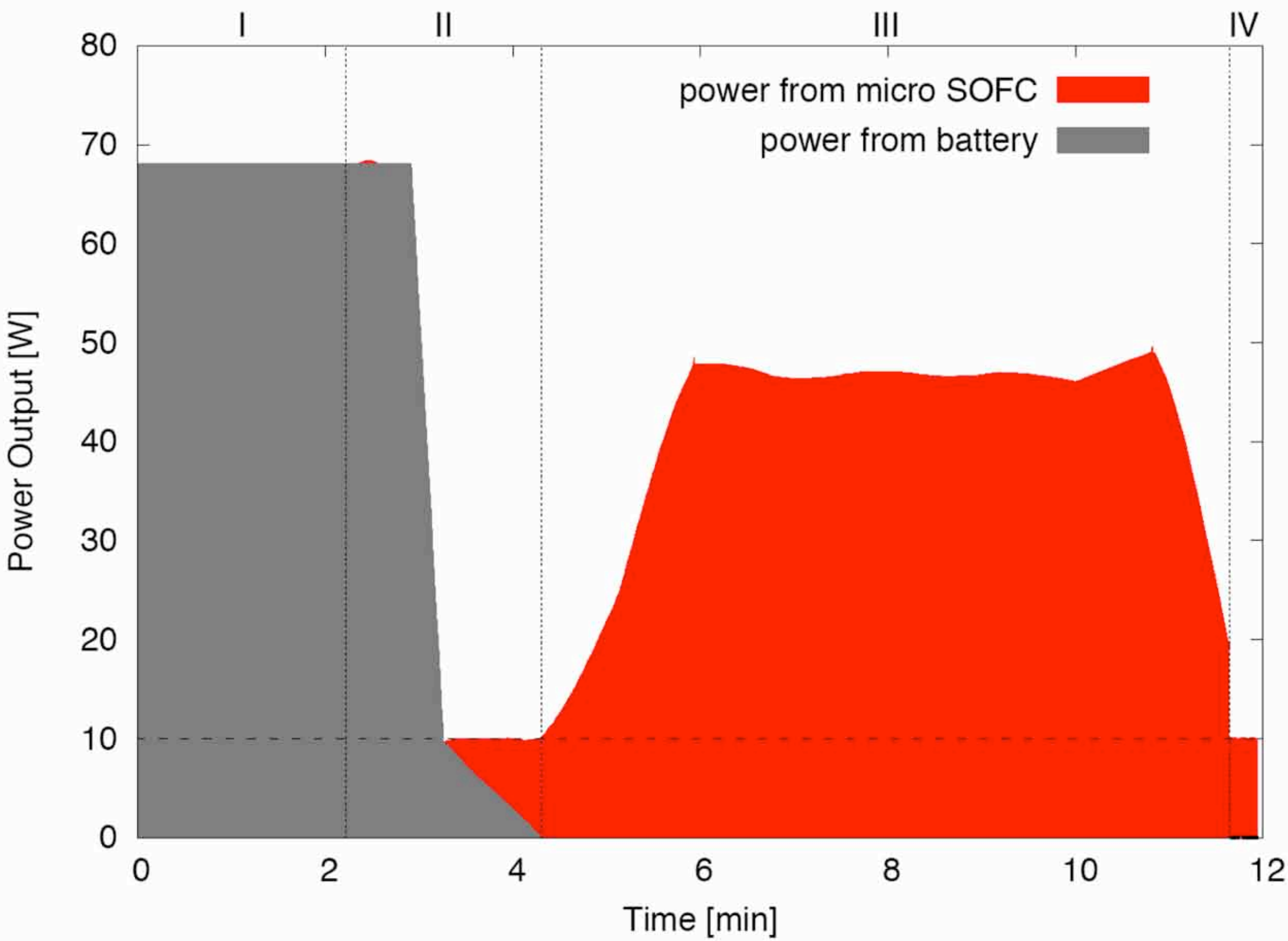
- **Hybrid Discrete/Continuous 1D-PDAEs:**
  - Discharge mode: $\mathscr{P}^{\text{bat}}_{\text{heat}} \geq 0$ and $\mathscr{P}^{\text{fc}}_{\text{rech}} = 0$
  - Recharge mode: $\mathscr{P}^{\text{fc}}_{\text{rech}} \geq 0$ and $\mathscr{P}^{\text{bat}}_{\text{heat}} = 0$
- **Index Analysis:**
  - Time index 1
  - Spatial index 1

Energy Conservation:

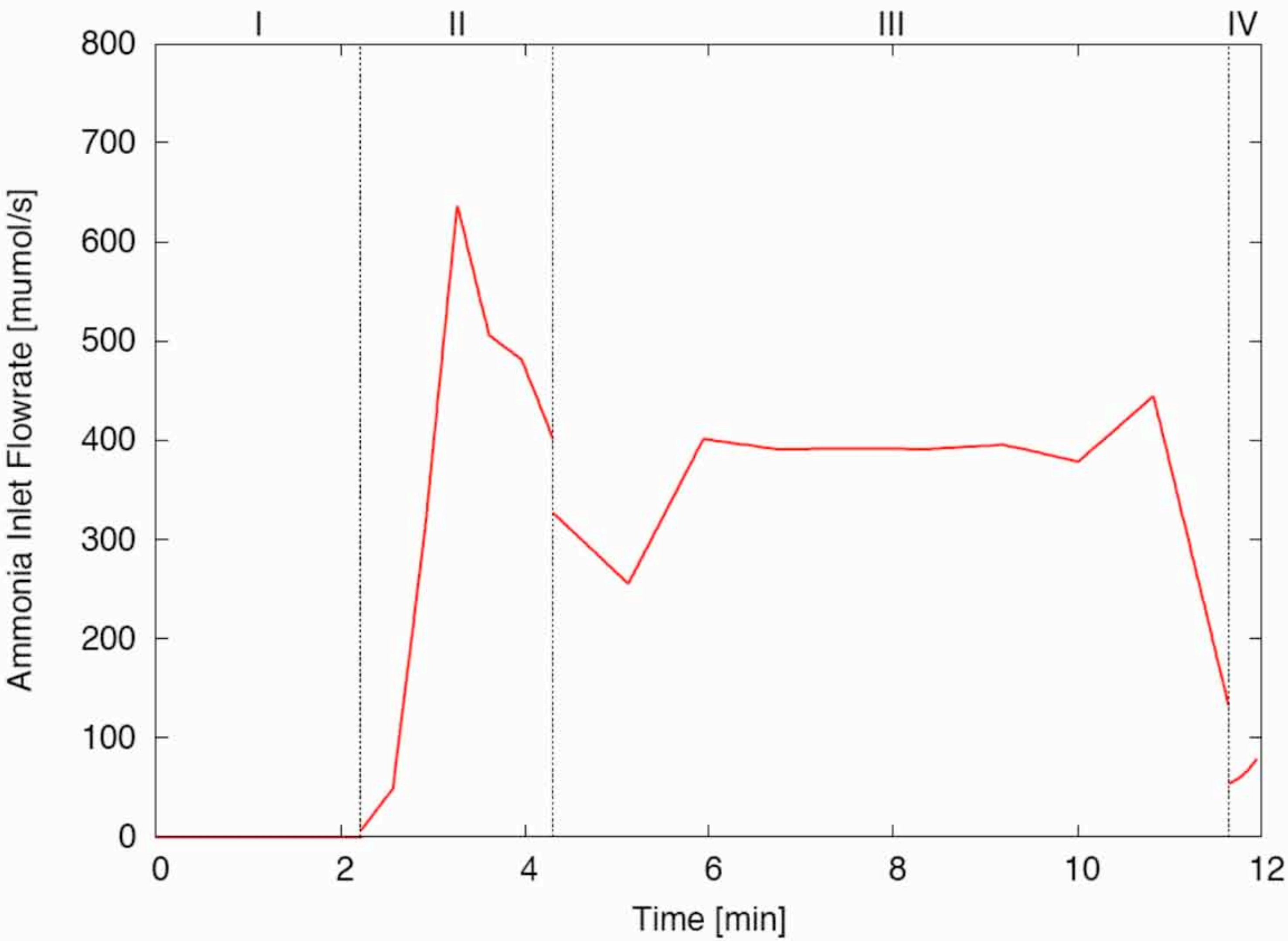
Battery: $\dot{\mathscr{E}}^{\text{bat}}(t) = -\mathscr{P}^{\text{bat}}_{\text{out}}(t) - \mathscr{P}^{\text{bat}}_{\text{heat}}(t) + \mathscr{P}^{\text{fc}}_{\text{rech}}(t) - \dot{Q}^{\text{bat}}_{\text{loss}}(t)$

Fuel Cell Stack: $V^{\text{dev}} c_p^{\text{dev}} \dot{T}(t) = \left[\dot{H}^{\text{dev}}_{\text{in}}(t) - \dot{H}^{\text{dev}}_{\text{out}}(t)\right] - \mathscr{P}^{\text{fc}}_{\text{out}}(t) - \mathscr{P}^{\text{fc}}_{\text{rech}}(t) + \mathscr{P}^{\text{bat}}_{\text{heat}}(t) - \dot{Q}^{\text{dev}}_{\text{loss}}(t)$

Optimal Start-up Results



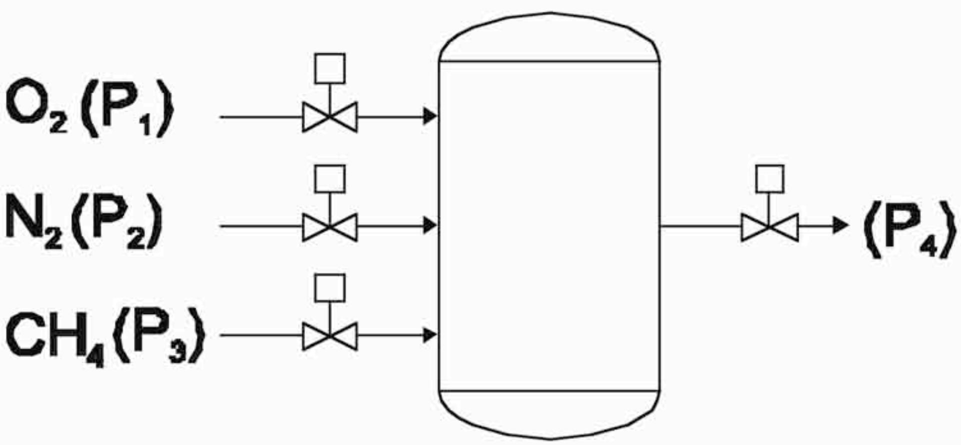
Optimal Start-up Results

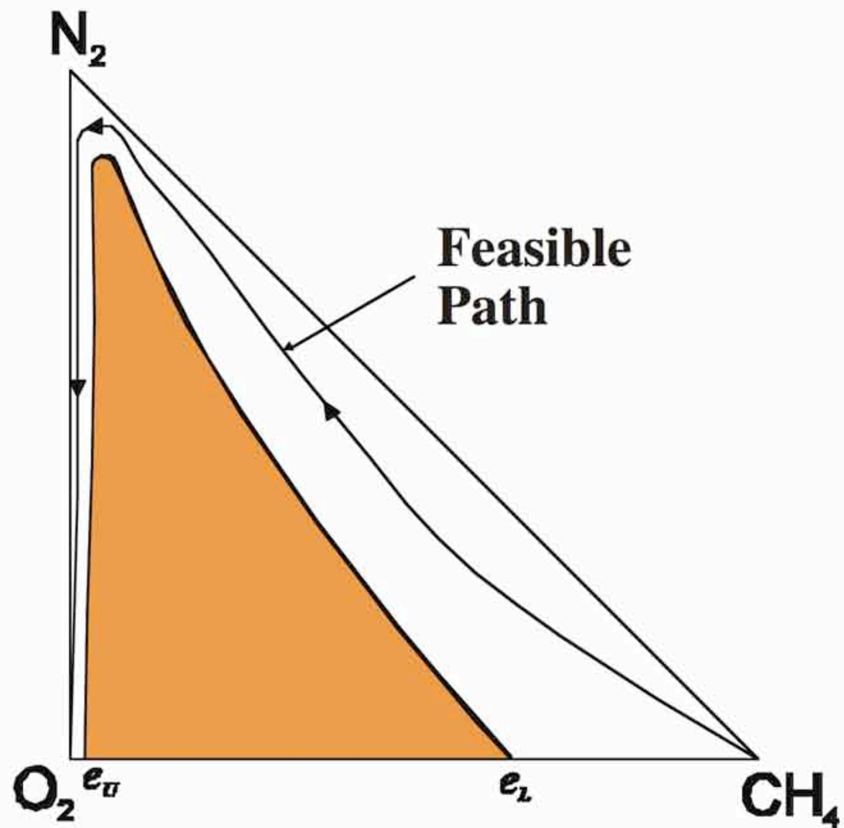


LOCAL NONSMOOTH OPTIMIZATION

Tank Changeover Example

$$\min_{u(t),t_f} \quad t_f$$





Model Formulation

$$\left.\begin{aligned}\dot{M}_j &= N_1 y_{1j} + N_2 y_{2j} + N_3 y_{3j} - N_4 y_{4j}\\ M_j &= M_T y_j\\ y_{4j} &= y_j\end{aligned}\right\} \forall\, j \in J$$

$$\sum_{j \in J} y_j = 1 \qquad\qquad PV = M_T RT$$

Model:

$$N_i = \left\{\begin{array}{ll} 0 & \text{if } \dfrac{P}{P_i} \geq 1\\ u_i C_v A_i \sqrt{\dfrac{P_i + P}{2}} \dfrac{P_i - P}{b + \sqrt{P_i - P}} & \text{if } 0.53 < \dfrac{P}{P_i} < 1\\ u_i C_v A_i \dfrac{P_i}{\sqrt{2}} 0.85 & \text{if } \dfrac{P}{P_i} \leq 0.53 \end{array}\right\} \forall\, i = 1, \ldots, 3$$

$$N_4 = \left\{\begin{array}{ll} 0 & \text{if } \dfrac{P_4}{P} \geq 1\\ u_4 C_v A_4 \sqrt{\dfrac{P_4 + P}{2}} \dfrac{P - P_4}{b + \sqrt{P - P_4}} & \text{if } 0.53 < \dfrac{P_4}{P} < 1\\ u_4 C_v A_4 \dfrac{P}{\sqrt{2}} 0.85 & \text{if } \dfrac{P_4}{P} \leq 0.53 \end{array}\right.$$

Point Constraints:

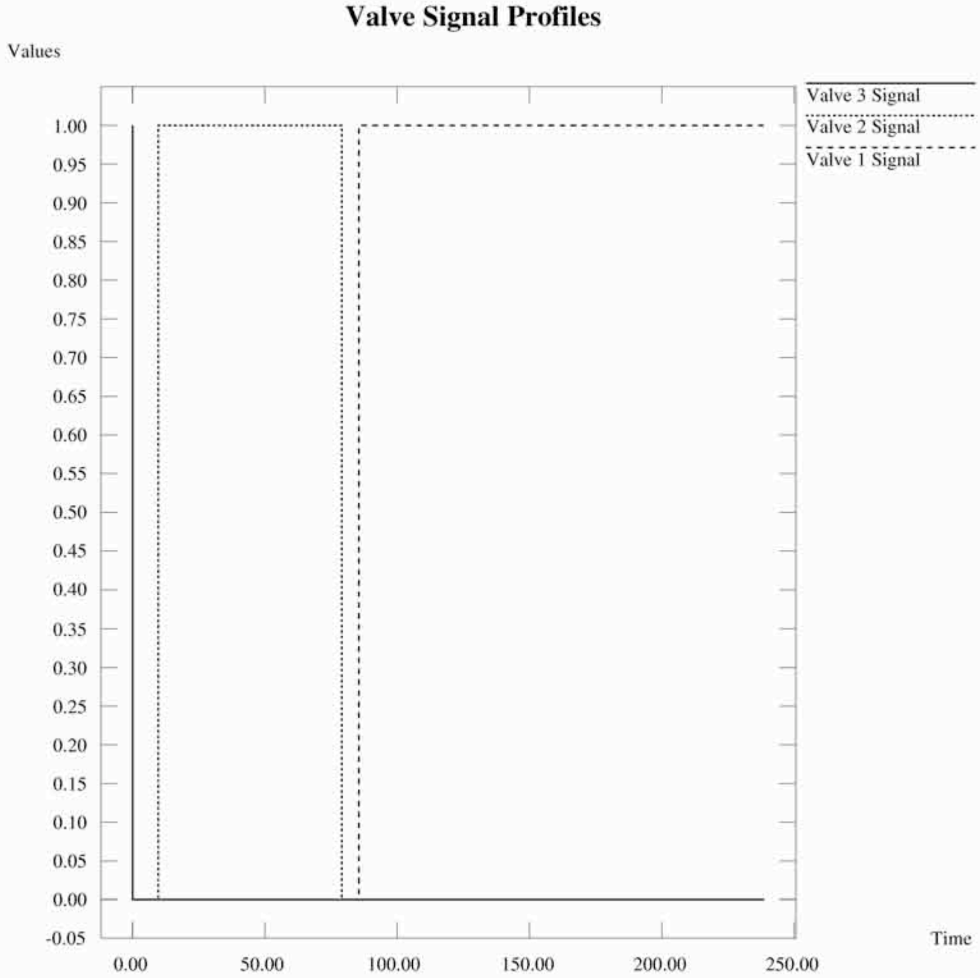
$$y_{O_2}(0) = 0 \qquad y_{CH_4}(0) = 1 \qquad P(0) = P_0$$

$$y_{O_2}(t_f) \geq 0.999$$

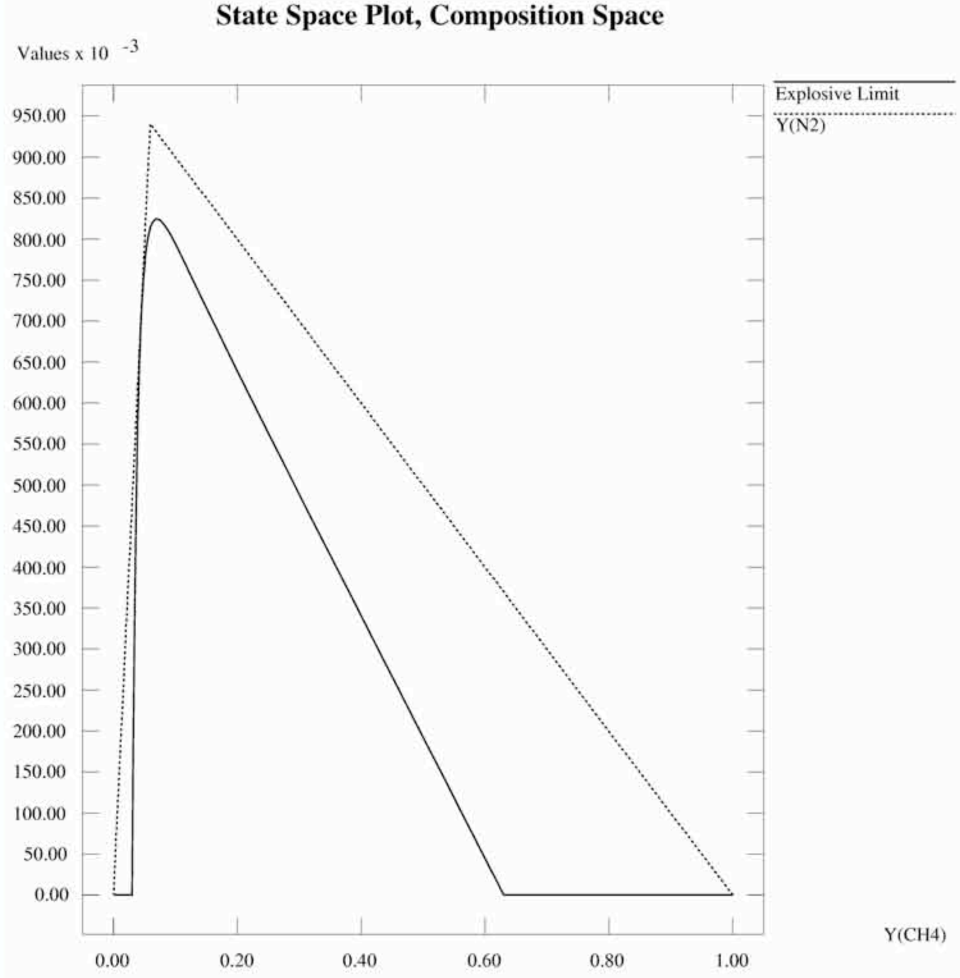
Path Constraints:

$$h(y_{O_2}(t), y_{CH_4}(t)) \leq 0$$

Results

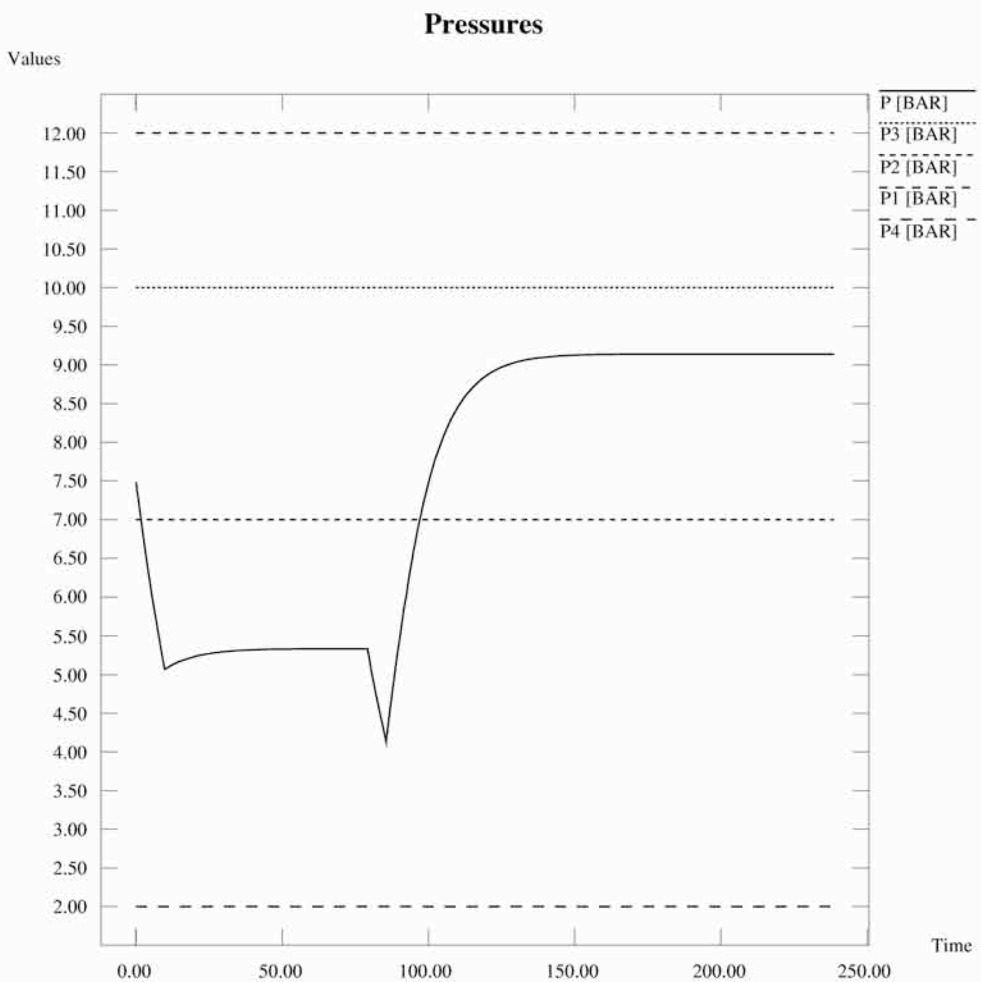


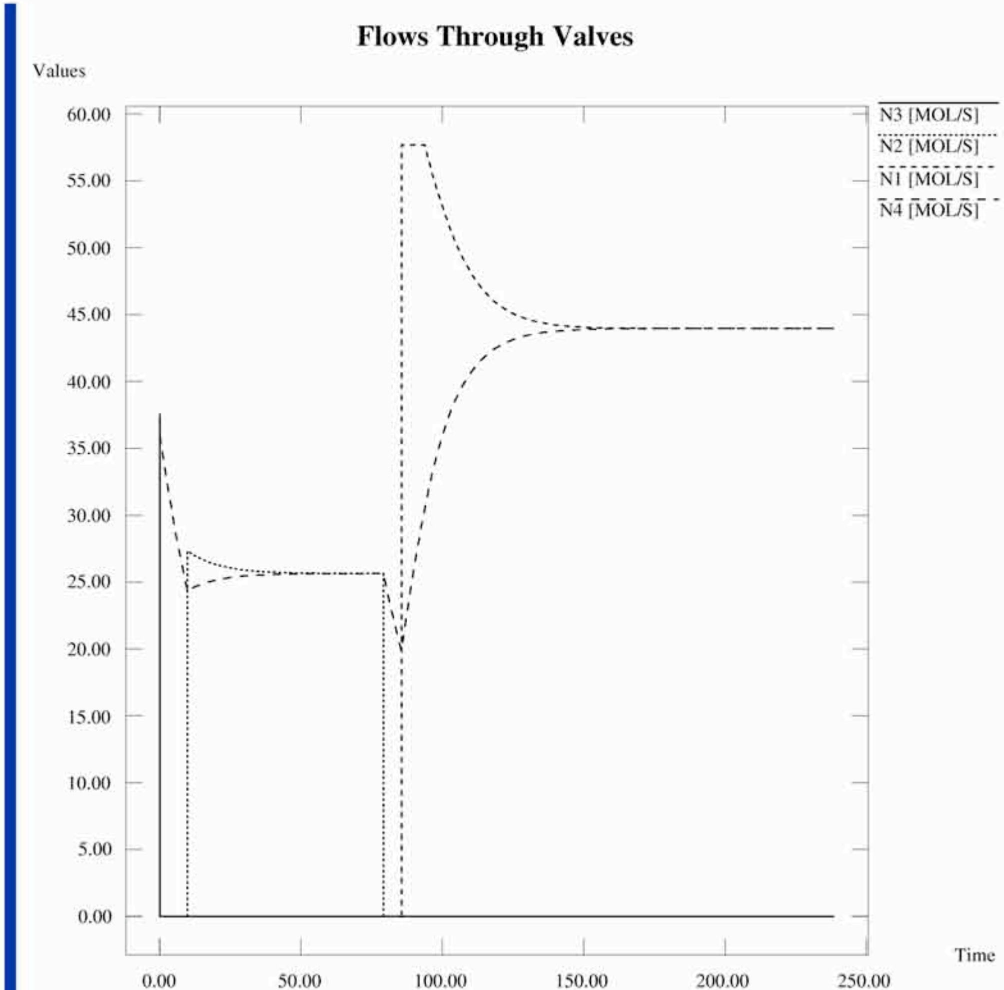
Valve Signal Profiles
Values
Valve 3 Signal
Valve 2 Signal
Valve 1 Signal
1.00
0.95
0.90
0.85
0.80
0.75
0.70
0.65
0.60
0.55
0.50
0.45
0.40
0.35
0.30
0.25
0.20
0.15
0.10
0.05
0.00
-0.05
0.00
50.00
100.00
150.00
200.00
250.00
Time



State Space Plot, Composition Space
Values x 10 -3
Explosive Limit
Y(N2)
950.00
900.00
850.00
800.00
750.00
700.00
650.00
600.00
550.00
500.00
450.00
400.00
350.00
300.00
250.00
200.00
150.00
100.00
50.00
0.00
0.00
0.20
0.40
0.60
0.80
1.00
Y(CH4)

Results


Pressures
Values
12.00
11.50
11.00
10.50
10.00
9.50
9.00
8.50
8.00
7.50
7.00
6.50
6.00
5.50
5.00
4.50
4.00
3.50
3.00
2.50
2.00
0.00
50.00
100.00
150.00
200.00
250.00
Time
P [BAR]
P3 [BAR]
P2 [BAR]
P1 [BAR]
P4 [BAR]


Flows Through Valves
Values
60.00
55.00
50.00
45.00
40.00
35.00
30.00
25.00
20.00
15.00
10.00
5.00
0.00
0.00
50.00
100.00
150.00
200.00
250.00
Time
N3 [MOL/S]
N2 [MOL/S]
N1 [MOL/S]
N4 [MOL/S]

Watch This Space...

CONTINUOUS GLOBAL DYNAMIC OPTIMIZATION

Problem Formulation

$$\min_{\mathbf{p}} \; J(\mathbf{p}) = \phi(\mathbf{x}(t_f, \mathbf{p}), \mathbf{p}) + \int_{t_0}^{t_f} \ell(t, \mathbf{x}(t, \mathbf{p}), \mathbf{p}) \; dt$$

$$\text{s.t.} \quad \mathbf{g}(\mathbf{x}(t_f, \mathbf{p}), \mathbf{p}) + \int_{t_0}^{t_f} \mathbf{h}(t, \mathbf{x}(t, \mathbf{p}), \mathbf{p}) \; dt \leq \mathbf{0}$$

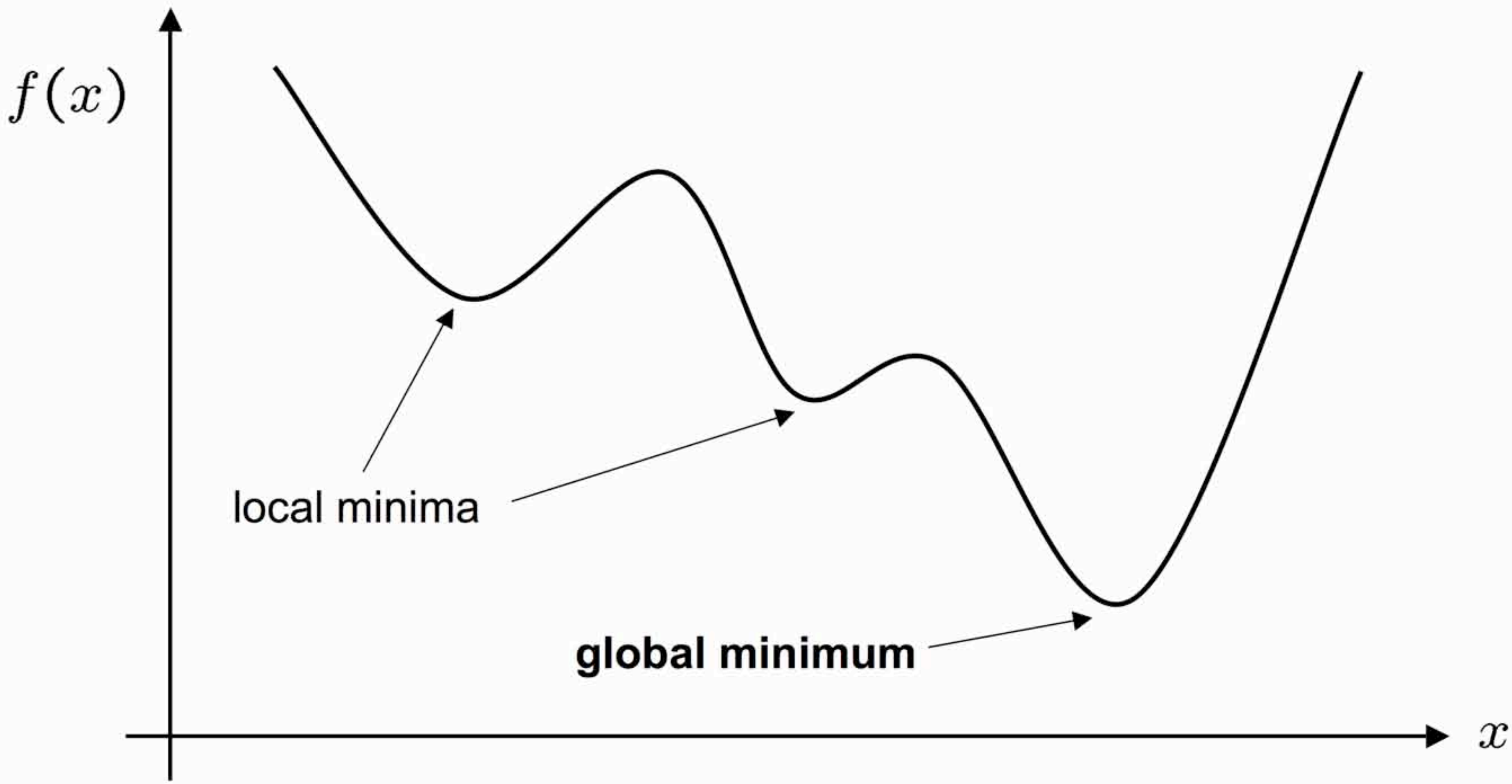
$$\mathbf{p} \in P \subset \mathbb{R}^{n_p}$$

where $\mathbf{x}(t, \mathbf{p})$ is given by

$$\dot{\mathbf{x}} = \mathbf{f}(t, \mathbf{x}, \mathbf{p})$$

$$\mathbf{x}(t_0, \mathbf{p}) = \mathbf{x}_0(\mathbf{p})$$

Nonconvex Optimization



Standard optimization techniques cannot distinguish between local minima.

Branch-and-Bound

Step 1 – Bound the solution

- Find *upper* and *lower* bounds such that:

$$\text{LBD} \le \min_{\mathbf{p}\in P\subset\mathbb{R}^{np}} f(\mathbf{p}) \le \text{UBD}$$

- Upper bound:

  Any feasible point
  e.g., a local minimum

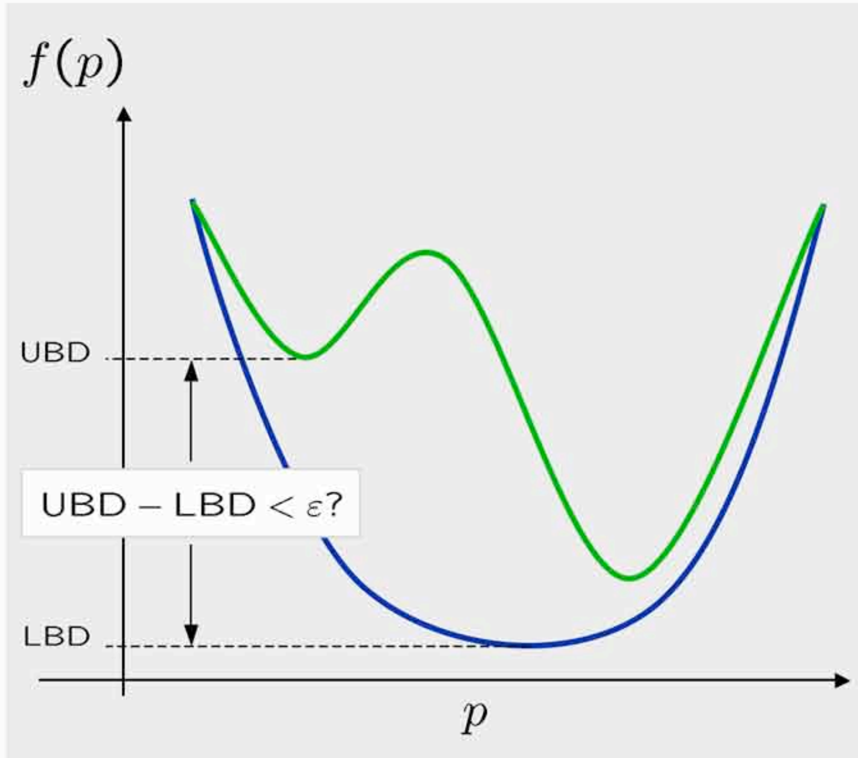
- Lower bound:

  Solution of a *convex relaxation*

$$\begin{cases} u(\mathbf{p}) \le f(\mathbf{p}) & \forall \mathbf{p} \in P \\ u \text{ convex on } P \end{cases}$$

  Local optimizer:

$$\text{LBD} = \min_{\mathbf{p}\in P} u(\mathbf{p})$$



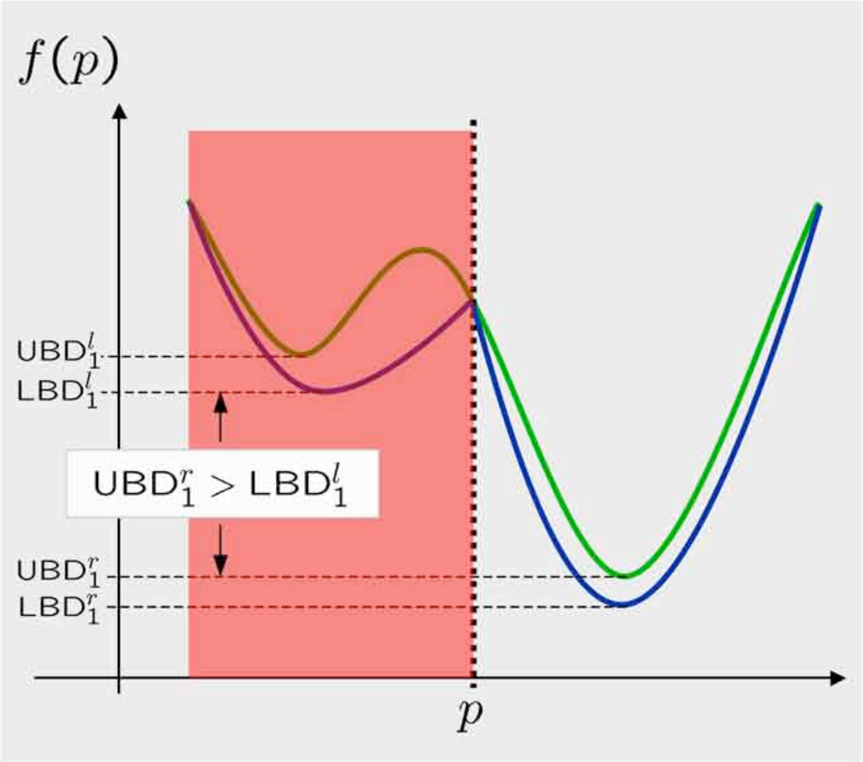
Branch-and-Bound

Step 2a – Branch the decision space

- Seek sequences $\{\text{UBD}_n\}$ and $\{\text{LBD}_n\}$ such that:

$$\lim_{n\to\infty} \text{LBD}_n = \min_{\mathbf{p}\in P\subset\mathbb{R}^{n_p}} f(\mathbf{p}) = \lim_{n\to\infty} \text{UBD}_n$$

- Exhaustive partitioning of the search space



Step 2b – Fathom

- Remove any partition wherein the lower bound exceeds the best upper bound (incumbent)

Convex Relaxations for Dynamic Optimization

$$J(\mathbf{p}) = \phi(\mathbf{x}(t_f, \mathbf{p}), \mathbf{p}) + \int_{t_0}^{t_f} \ell(t, \mathbf{x}(t, \mathbf{p}), \mathbf{p})\ dt$$

where $\mathbf{x}(t, \mathbf{p})$ is given by

$$\dot{\mathbf{x}} = \mathbf{f}(t, \mathbf{x}, \mathbf{p})$$
$$\mathbf{x}(t_0, \mathbf{p}) = \mathbf{x}_0(\mathbf{p})$$

The Question:

How to construct a convex relaxation of $J(\mathbf{p})$ when $\mathbf{x}(t, \mathbf{p})$ can only be evaluated by numerical solution of ODEs?

McCormick's Composition Theorem

Theorem. Let $P \subset \mathbb{R}^{n_p}$ be a nonempty convex set. Consider the function $f\left[x\left(\mathbf{p}\right)\right]$ and let

$$P \subset \left\{\mathbf{p} : x\left(\mathbf{p}\right) \in [x^L, x^U]\right\}.$$

Suppose we know a convex function c and a concave function C such that

$$c(\mathbf{p}) \leq x(\mathbf{p}) \leq C(\mathbf{p}) \quad \forall\ \mathbf{p} \in P,$$

and let e be a convex relaxation of f on $[x^L, x^U]$.

Then,

$$u(\mathbf{p}) = e\left[\text{mid}\left\{c(\mathbf{p}), C(\mathbf{p}), z_{\text{min}}\right\}\right]$$

is a convex relaxation of $f\left[x(\mathbf{p})\right]$ on P.

(McCormick factorization extends this to a vector $\mathbf{x}(\mathbf{p})$).

Relaxations of Functionals with ODEs Embedded

For both point and integral form functionals apply McCormick's theorem **pointwise in time**.

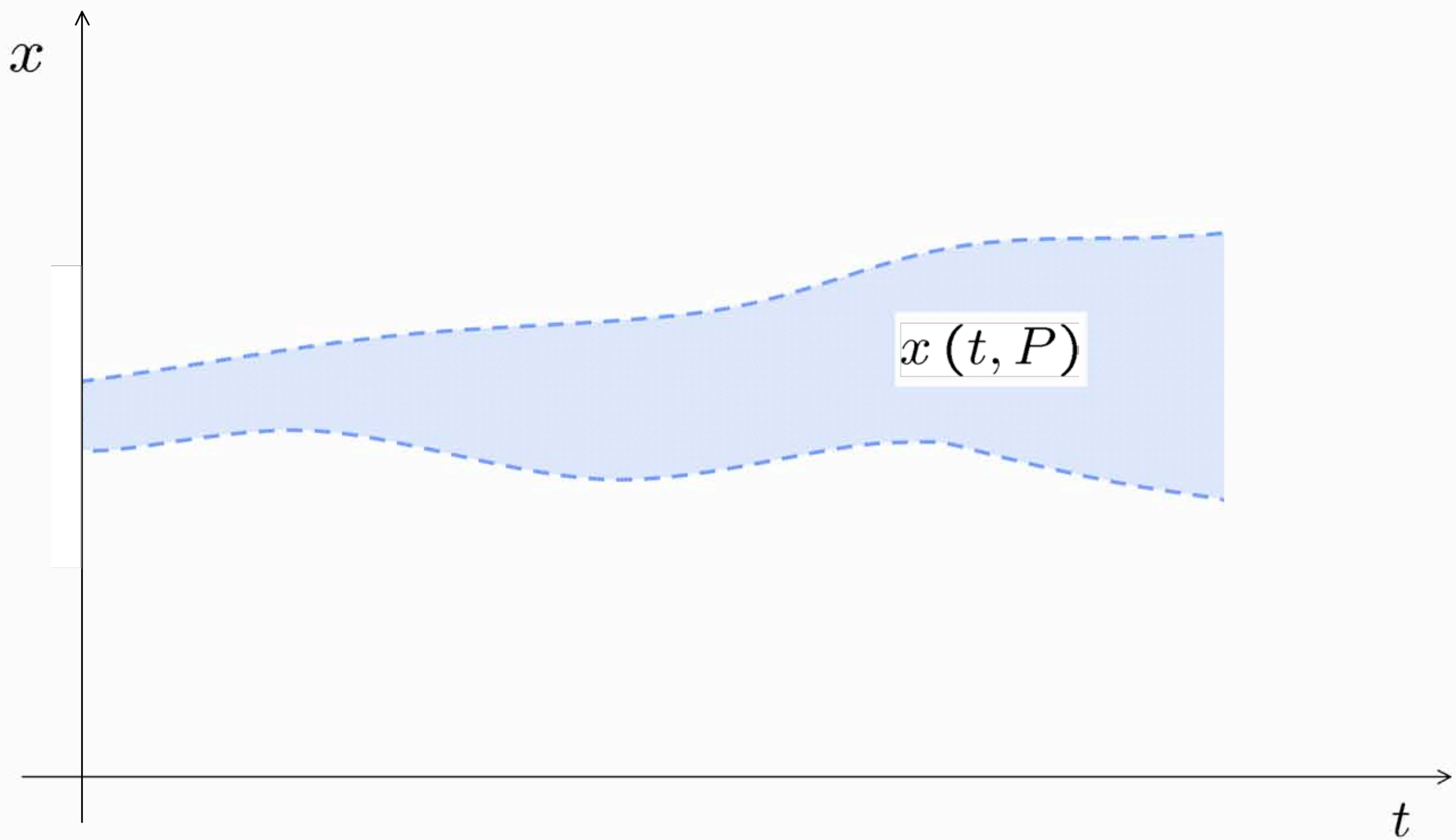
- Need **implied state bounds**:

$$\mathbf{x}^L(t), \mathbf{x}^U(t) : \mathbf{x}^L(t) \leq \mathbf{x}(t, \mathbf{p}) \leq \mathbf{x}^U(t)\,,\ \forall \mathbf{p} \in P$$

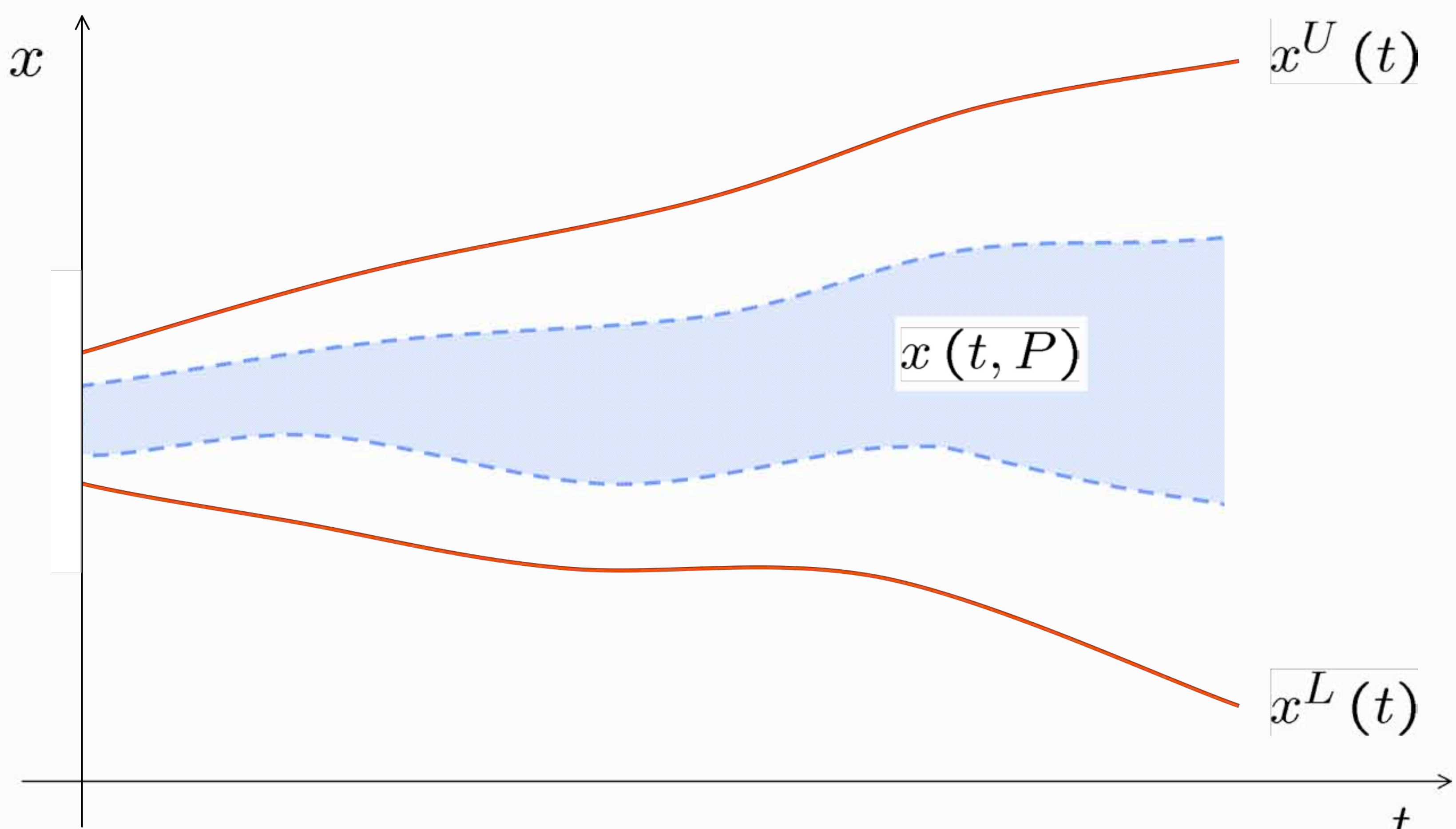
- And pointwise in time **convex** and **concave relaxations** for each $\underline{t} \in [t_0, t_f]$:

$$\mathbf{c}(t, \cdot), \mathbf{C}(t, \cdot) : \mathbf{c}(t, \mathbf{p}) \leq \mathbf{x}(t, \mathbf{p}) \leq \mathbf{C}(t, \mathbf{p})\,,\ \forall \mathbf{p} \in P$$

Implied State Bounds



Implied State Bounds



Linear ODEs

$$\dot{\mathbf{x}} = \mathbf{A}(t)\mathbf{x} + \mathbf{B}(t)\mathbf{p} + \mathbf{r}(t)$$
$$\mathbf{E}_0\mathbf{x}(t_0) + \mathbf{E}_f\mathbf{x}(t_f) = \mathbf{D}\mathbf{p} + \mathbf{q}$$

- Easy to show solution is **affine** in $\mathbf{p}$ for fixed t:

$$\mathbf{x}(t, \mathbf{p}) = \mathbf{M}(t)\mathbf{p} + \mathbf{n}(t)$$

- Affine function is both convex and concave:

$$\mathbf{c}(t, \mathbf{p}) = \mathbf{x}(t, \mathbf{p}) = \mathbf{C}(t, \mathbf{p})$$

- **Exact** implied state bounds from interval analysis:

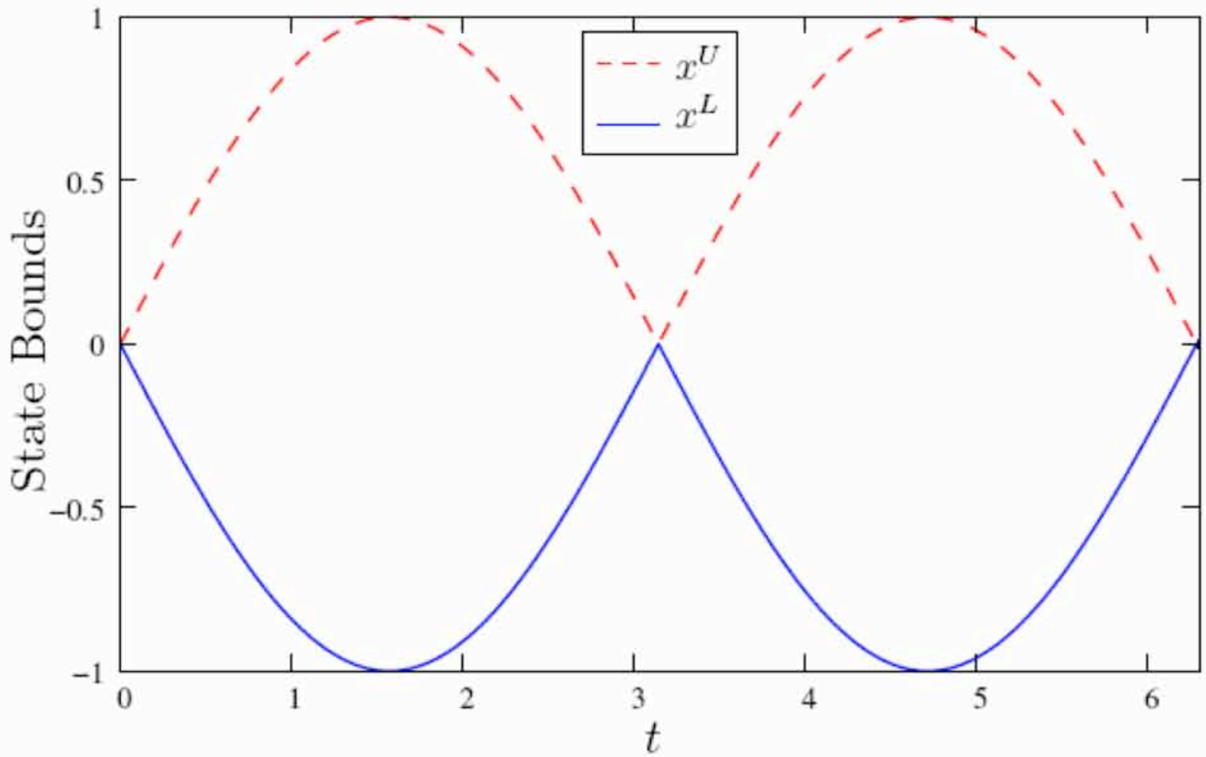
$$[\mathbf{x}](\underline{t}, [\mathbf{p}]) = \mathbf{M}(\underline{t})[\mathbf{p}] + \mathbf{n}(\underline{t})$$

Linear ODEs Implied State Bounds

$$\dot{x}_1 = x_2$$
$$\dot{x}_2 = -x_1$$
$$x_1(0) = 0, x_2(0) = p$$
$$p \in [-1, 1]$$

$$\xrightarrow[\text{bounds}]{\text{exact}}$$

$$x_1^L(t) = \min\{-1\sin(t), 1\sin(t)\}$$
$$x_1^U(t) = \max\{-1\sin(t), 1\sin(t)\}$$



Nonlinear ODEs
Harrison's Theorem

Suppose that:

$$\mathbf{x}^L(t_0) \le \inf_{\mathbf{p}\in P} \mathbf{x}(t_0,\mathbf{p}) \quad \text{and} \quad \mathbf{x}^U(t_0) \ge \sup_{\mathbf{p}\in P} \mathbf{x}(t_0,\mathbf{p})$$

and if $\forall \mathbf{x}^L(t), \mathbf{x}^U(t) \in G(t) = \left\{\mathbf{z} \in \mathbb{R}^{n_x} : \mathbf{x}^L(t) \le \mathbf{z} \le \mathbf{x}^U(t)\right\}$,

$$\left.\begin{array}{lcl} \dot{x}_i^L &=& g_i^L\left(t, \mathbf{x}^L, \mathbf{x}^U\right) \le \inf\limits_{\substack{\mathbf{z}\in G(t),\, \mathbf{p}\in P \\ z_i = x_i^L(t)}} f_i(t,\mathbf{z},\mathbf{p}) \\ \dot{x}_i^U &=& g_i^U\left(t, \mathbf{x}^L, \mathbf{x}^U\right) \ge \sup\limits_{\substack{\mathbf{z}\in G(t),\, \mathbf{p}\in P \\ z_i = x_i^U(t)}} f_i(t,\mathbf{z},\mathbf{p}) \end{array}\right\} \forall i = 1,\ldots,n_x$$

then,

$$\mathbf{x}^L(t) \le \mathbf{x}(t,\mathbf{p}) \le \mathbf{x}^U(t)\,,\ \forall(\mathbf{p},t) \in P \times [t_0, t_f]$$

Incorporating Natural Bounds and Solution Invariants

Suppose that it is known by an independent **physical** argument:

$$\mathbf{x}(t,\mathbf{p}) \in \overline{\mathcal{X}}(t) \quad \forall \mathbf{p} \in P.$$

If

$$\mathbf{x}^L(t_0) \leq \inf_{\mathbf{p}\in P} \mathbf{x}(t_0,\mathbf{p}) \;\text{ and }\; \mathbf{x}^U(t_0) \geq \sup_{\mathbf{p}\in P} \mathbf{x}(t_0,\mathbf{p})$$

and if $\forall \mathbf{x}^L(t), \mathbf{x}^U(t) \in G(t)$,

$$\left.\begin{aligned} \dot{x}_i^L &= g_i^L\left(t,\mathbf{x}^L,\mathbf{x}^U\right) \leq \inf_{\substack{\mathbf{z}\in\overline{\mathcal{X}}(t)\cap G(t),\, \mathbf{p}\in P \\ z_i = x_i^L(t)}} f_i(t,\mathbf{z},\mathbf{p}) \\ \dot{x}_i^U &= g_i^U\left(t,\mathbf{x}^L,\mathbf{x}^U\right) \geq \sup_{\substack{\mathbf{z}\in\overline{\mathcal{X}}(t)\cap G(t),\, \mathbf{p}\in P \\ z_i = x_i^U(t)}} f_i(t,\mathbf{z},\mathbf{p}) \end{aligned}\right\} \forall i = 1,\ldots,n_x$$

then,

$$\mathbf{x}^L(t) \leq \mathbf{x}(t,\mathbf{p}) \leq \mathbf{x}^U(t)\,,\; \forall(\mathbf{p},t) \in P \times [t_0,t_f]$$

Example

$$A + B \underset{k_r}{\overset{k_f}{\rightleftharpoons}} C$$

$$\begin{aligned}
\dot{x}_A &= -k_f x_A x_B + k_r x_C \\
\dot{x}_B &= -k_f x_A x_B + k_r x_C \\
\dot{x}_C &= k_f x_A x_B - k_r x_C
\end{aligned}$$

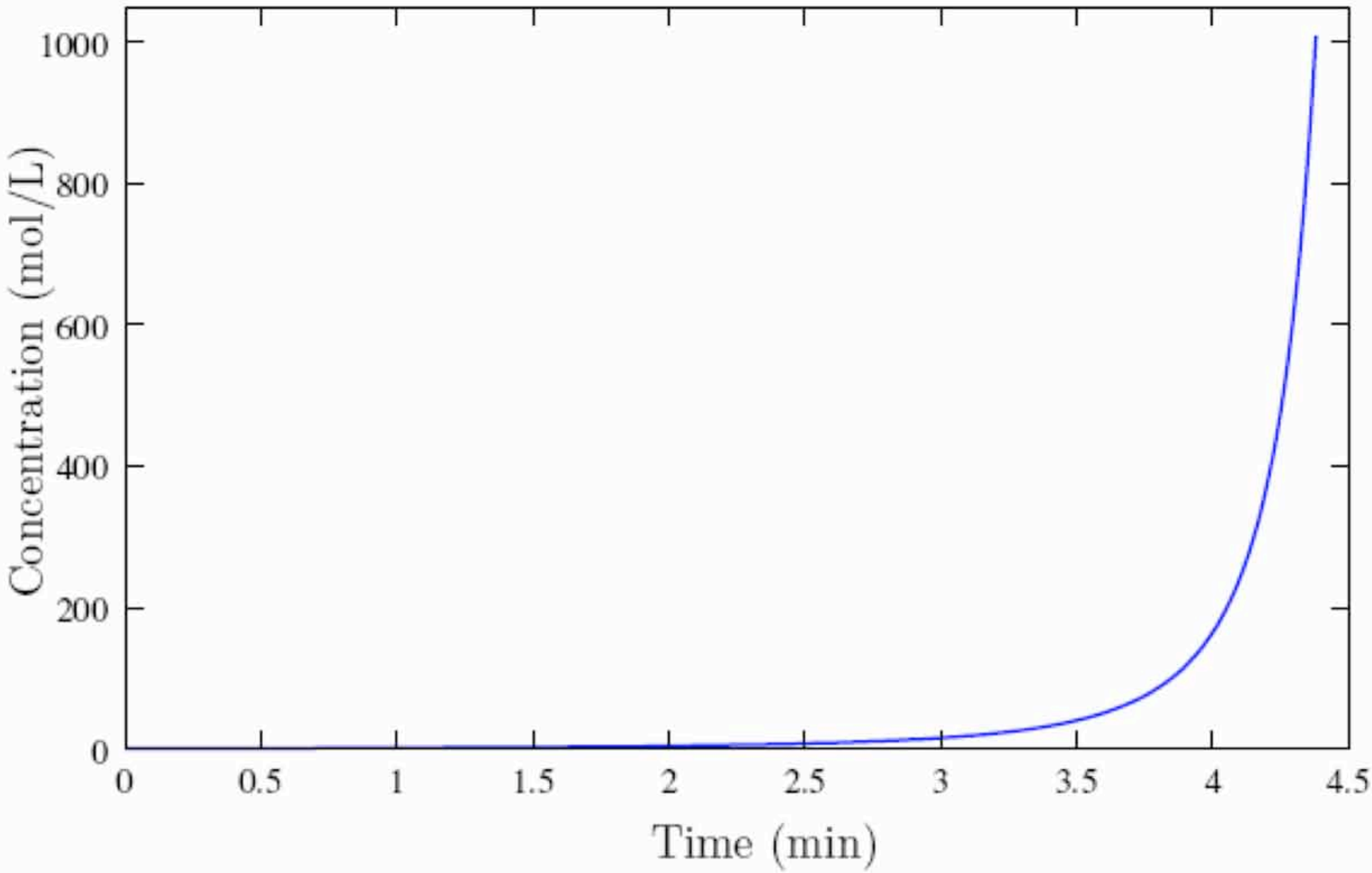
$$k_f \in [100, 500]$$
$$k_r \in [0.001, 0.01]$$

$$x_A(0) = x_{A0},\ x_B(0) = 0,\ x_C(0) = 0$$

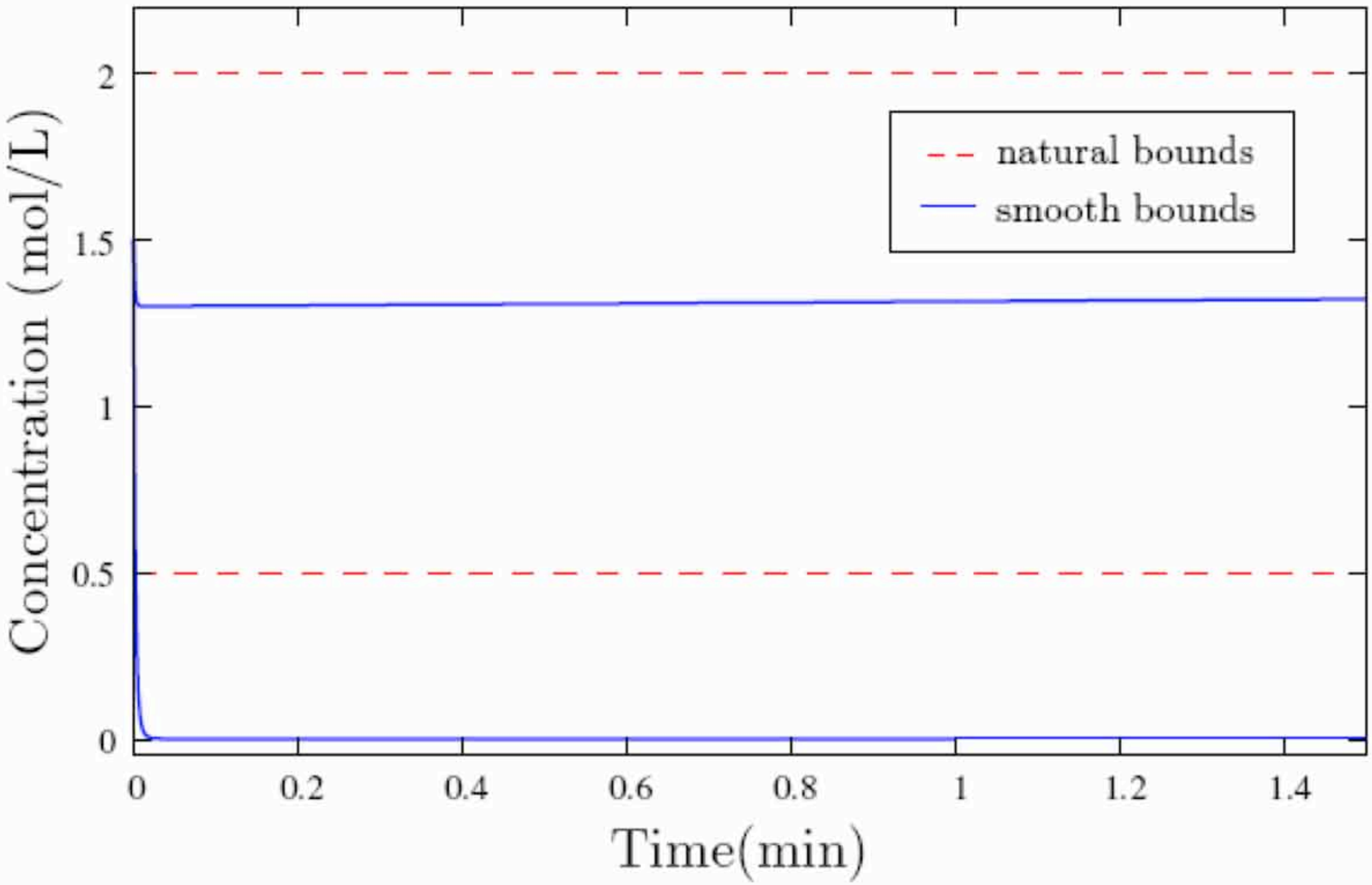
W.L.O.G assume $x_{B0} > x_{A0}$ so that by overall mass balance:

$$\begin{array}{rcccl}
0 & \leq & x_A(t) & \leq & x_{A0} + x_{C0} \\
x_{B0} - x_{A0} & \leq & x_B(t) & \leq & x_{B0} + x_{C0} \\
0 & \leq & x_C(t) & \leq & x_{A0} + x_{C0}
\end{array}$$

Bounds from Harrison's Theorem



Incorporating Natural Bounds



Convex and Concave Relaxations of ODE Solution

Find ODEs:

$$\begin{aligned}\dot{\mathbf{c}} &= \mathbf{f}^u\left(t, \mathbf{c}, \mathbf{C}, \mathbf{x}^L, \mathbf{x}^U, \mathbf{p}\right)\\ \dot{\mathbf{C}} &= \mathbf{f}^o\left(t, \mathbf{c}, \mathbf{C}, \mathbf{x}^L, \mathbf{x}^U, \mathbf{p}\right)\end{aligned}$$

such that:

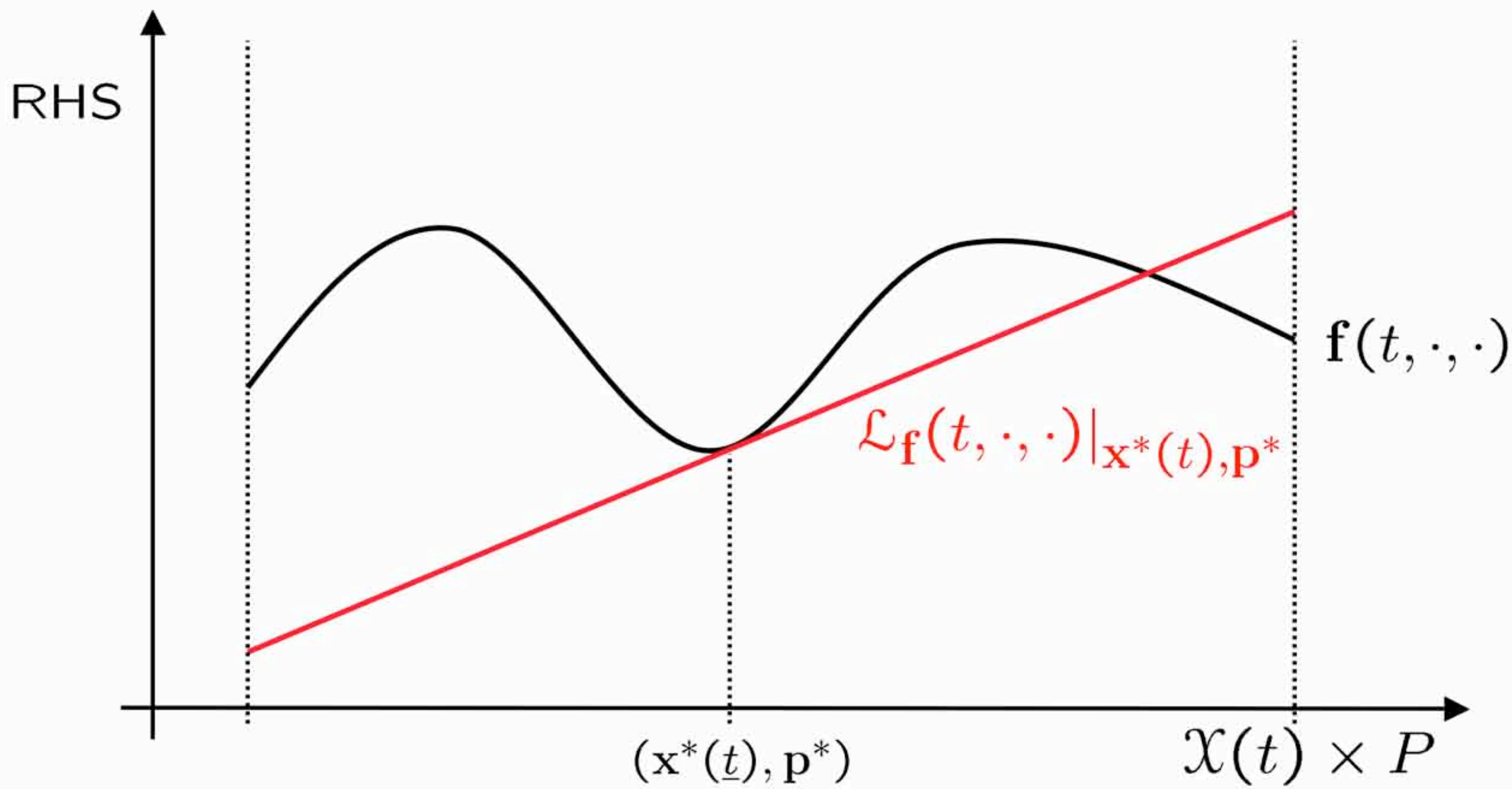
$$\mathbf{c}(t, \mathbf{p}) \leq \mathbf{x}(t, \mathbf{p}) \leq \mathbf{C}(t, \mathbf{p}) \quad \forall (\mathbf{p}, t) \in P \times [t_0, t_f]$$

and $\mathbf{c}(t, \cdot)$ is convex on P and $\mathbf{C}(t, \cdot)$ is concave on P.

If $\mathbf{f}^u$ and $\mathbf{f}^o$ define **linear systems**, then $\mathbf{c}(t, \cdot)$ and $\mathbf{C}(t, \cdot)$ will be **affine** and therefore convex and concave.

Outer Approximation Theorem

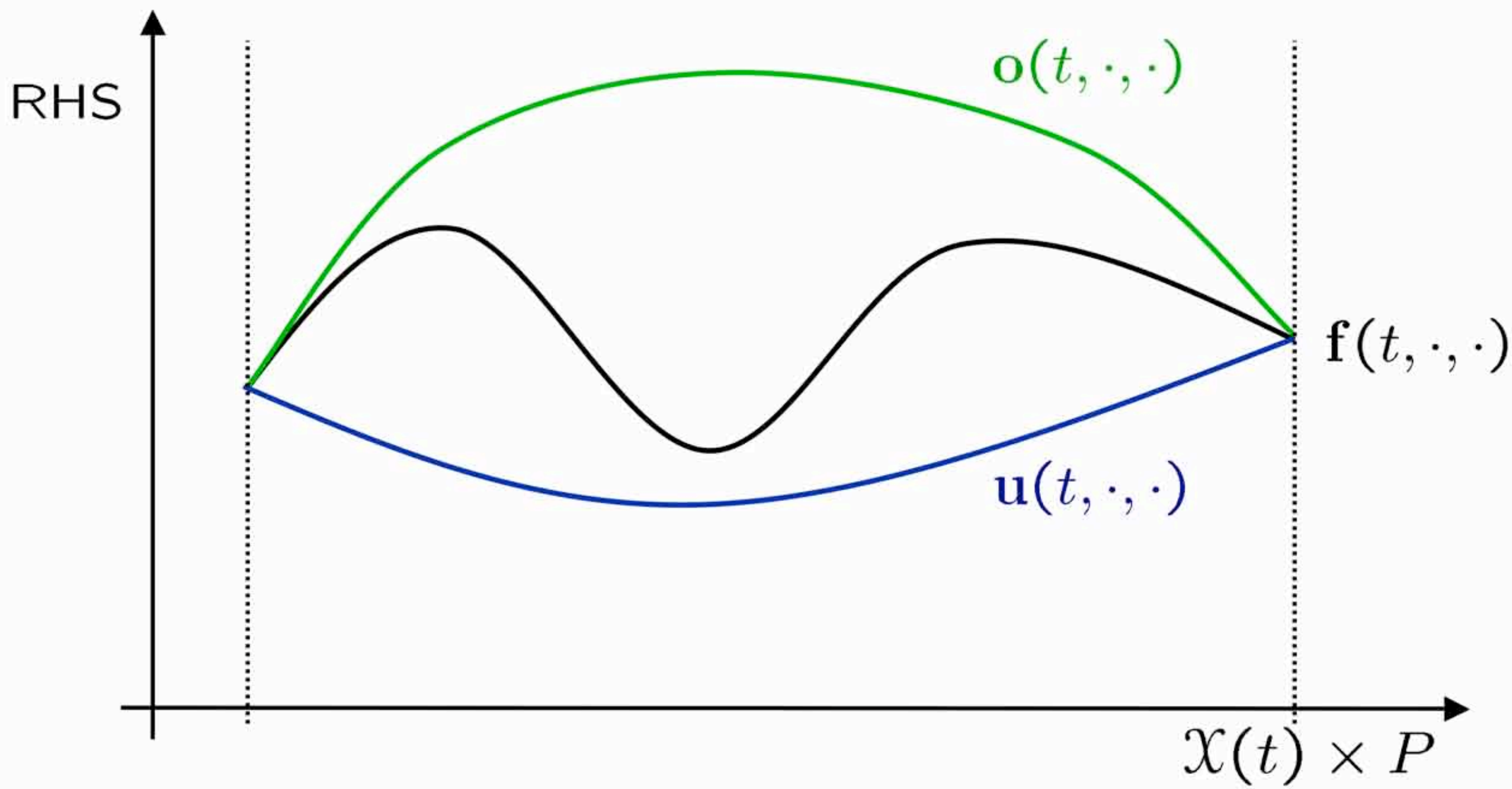
Pointwise in time:



Linearization along reference trajectory **invalid**!

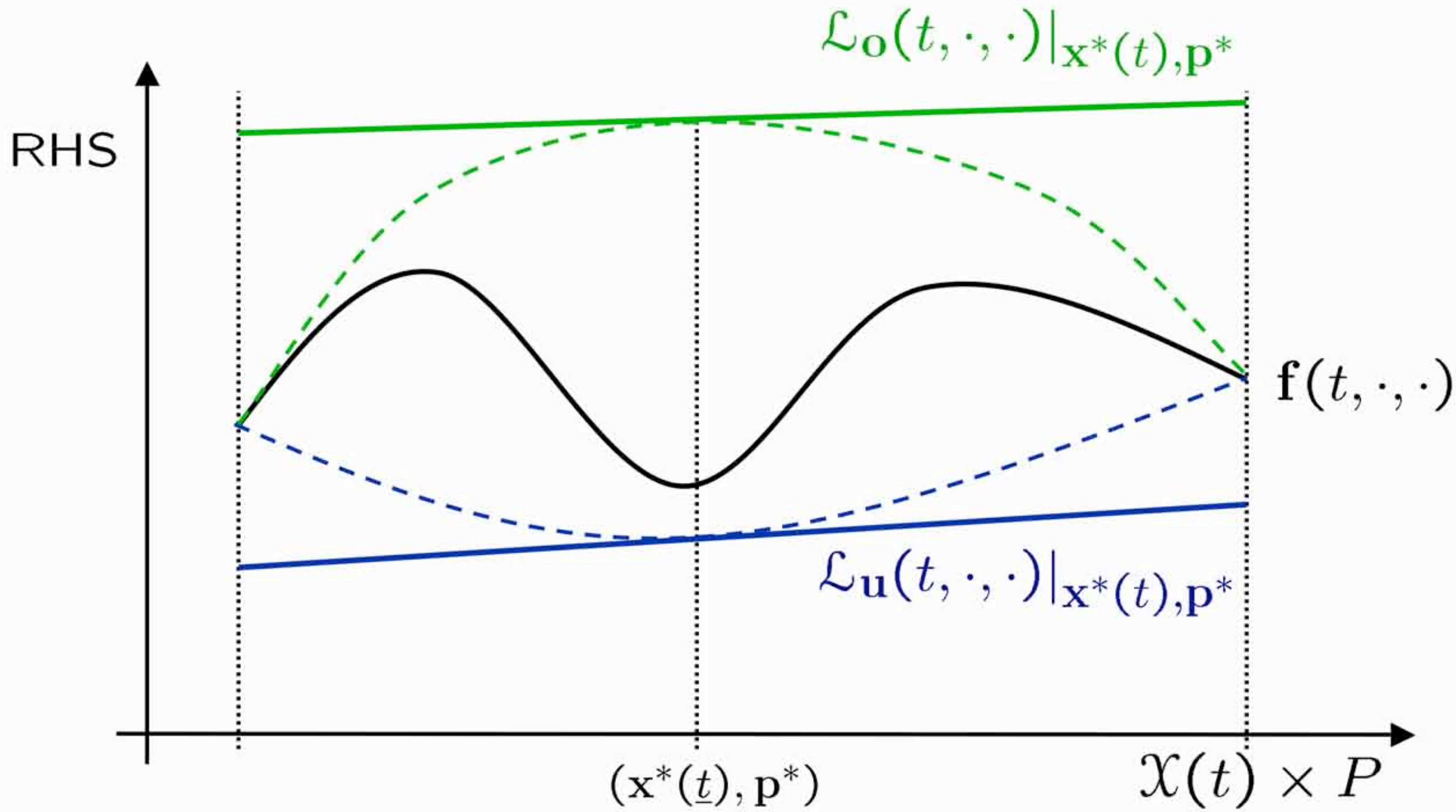
Outer Approximation Theorem

Pointwise in time:



Outer Approximation Theorem

Pointwise in time:



Outer Approximation Theorem

Suppose that:

$$\left.\begin{aligned}\dot{c}_i &= \inf_{\substack{\mathbf{z}\in\mathcal{C}(t,\mathbf{p})\\ z_i=c_i(t,\mathbf{p})}} \mathcal{L}_{u_i}(t,\mathbf{z},\mathbf{p})\Big|_{\mathbf{x}^*(t),\mathbf{p}^*}\\ \dot{C}_i &= \sup_{\substack{\mathbf{z}\in\mathcal{C}(t,\mathbf{p})\\ z_i=C_i(t,\mathbf{p})}} \mathcal{L}_{o_i}(t,\mathbf{z},\mathbf{p})\Big|_{\mathbf{x}^*(t),\mathbf{p}^*}\end{aligned}\right\}\ \forall i = 1,\ldots,n_x$$

and

$$\mathbf{c}(t_0,\mathbf{p}) \le \mathbf{x}(t_0,\mathbf{p}) \le \mathbf{C}(t_0,\mathbf{p})\ ,\ \forall\ \mathbf{p}\in P$$

where

$$\mathcal{C}(t,\mathbf{p}) = \{\mathbf{z} : \mathbf{c}(t,\mathbf{p}) \le \mathbf{z} \le \mathbf{C}(t,\mathbf{p})\}$$

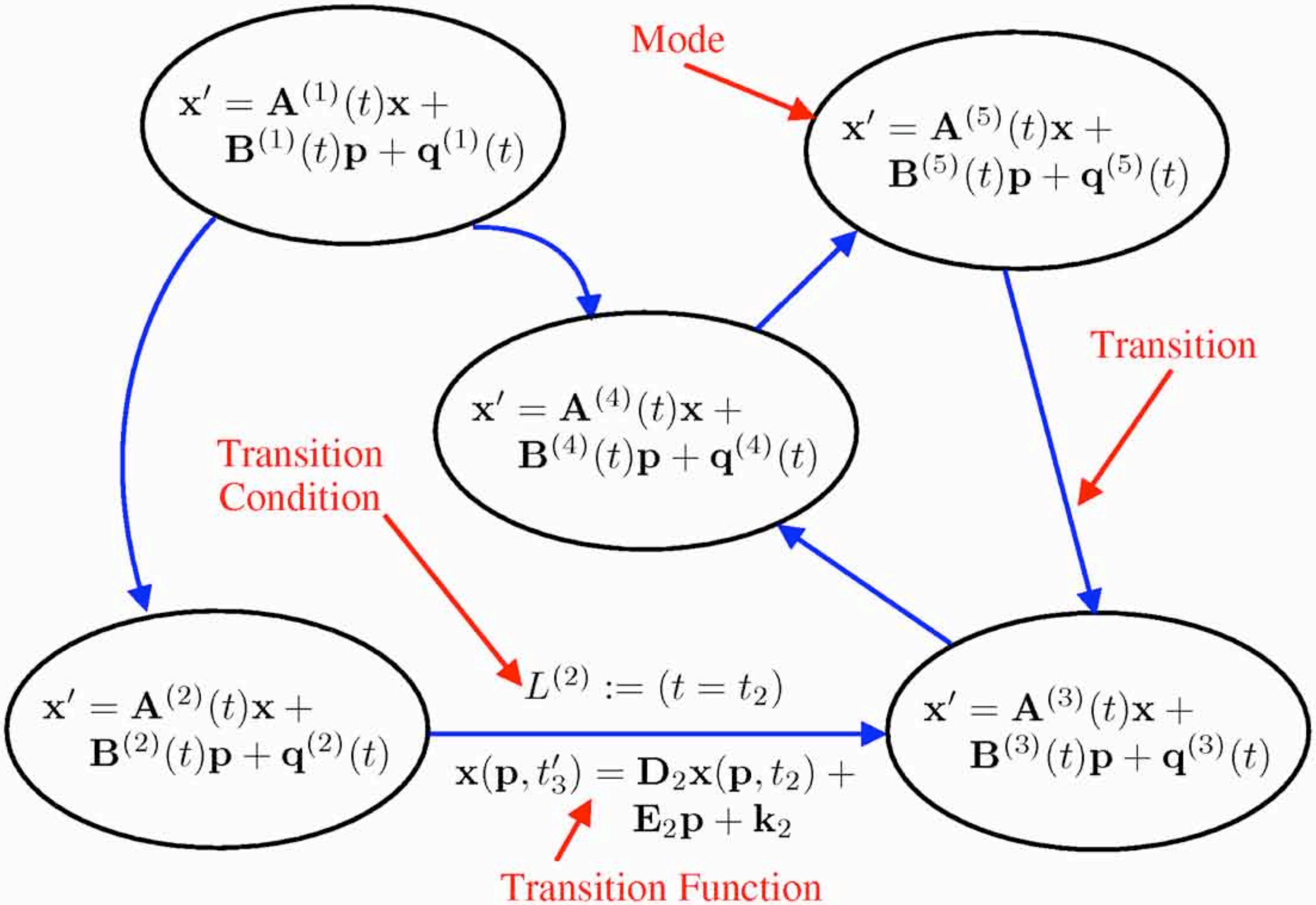
$\mathbf{c}(t,\cdot)$ and $\mathbf{C}(t,\cdot)$ have the desired properties

GLOBAL MIXED-INTEGER DYNAMIC OPTIMIZATION

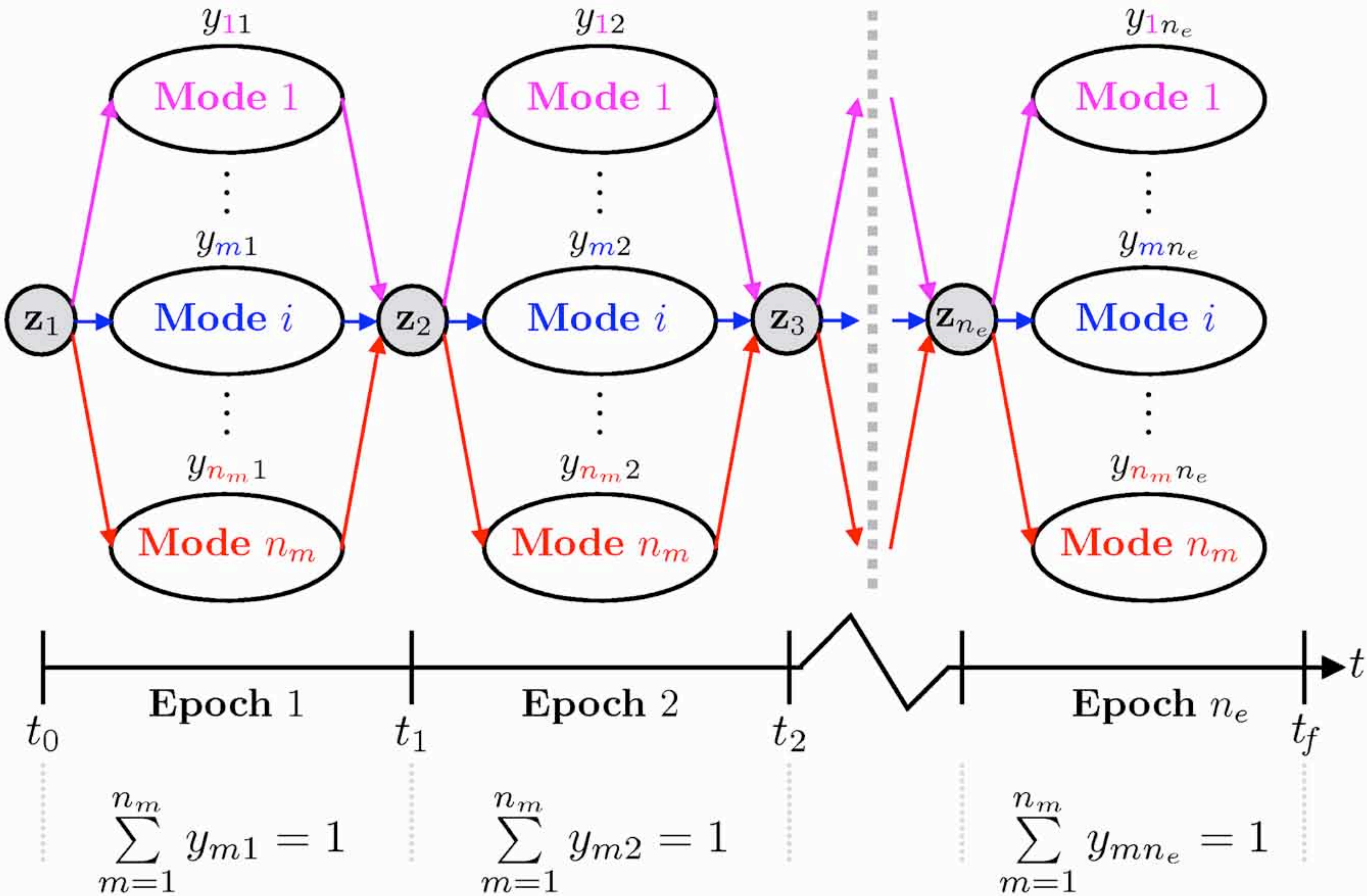
LTV Hybrid Automaton

Objective Function

$$\boxed{\sum_{i=1}^{n_\phi} \phi_i\Big(\mathbf{x}'(\mathbf{p}, t_i), \mathbf{x}(\mathbf{p}, t_i), \mathbf{p}\Big) + \int_{t_0}^{t_f} f\Big(\mathbf{x}'(\mathbf{p}, t), \mathbf{x}(\mathbf{p}, t), \mathbf{p}, t\Big)\ \mathrm{d}t}$$



Hybrid Superstructure



Problem Formulation

$$\min_{\mathbf{p},T_\mu} \quad \phi(\mathbf{x}(t_f,\mathbf{p},T_\mu),\mathbf{p}) + \sum_{i=1}^{n_e}\int_{\sigma_i}^{\tau_i} f_i(t,\mathbf{x}(t,\mathbf{p},T_\mu),\mathbf{p})\mathrm{d}t$$

$$\text{s.t.} \quad \mathbf{h}(\mathbf{x}(t_f,\mathbf{p},T_\mu),\mathbf{p}) + \sum_{i=1}^{n_e}\int_{\sigma_i}^{\tau_i} \mathbf{g}_i(t,\mathbf{x}(t,\mathbf{p},T_\mu),\mathbf{p})\mathrm{d}t \le \mathbf{0}$$

$$\mathbf{p} \in P \subset \mathbb{R}^{n_p}, T_\mu \in M^{n_e}$$

$\mathbf{x}(t,\mathbf{p},T_\mu)$ given by solution of embedded LTV hybrid system

Logical constraints can enforce permissible transitions in given hybrid automaton.

Varying Mode Sequence

- Binary decision variables are introduced to represent the mode sequence

$$y_{mi},\ i = 1, \ldots, n_e,\ m = 1, \ldots, n_m.$$

- If m_i^* is the active mode in epoch I_i then $y_{m_i^* i} = 1$ and $y_{m \neq m_i^*, i} = 0$:

$$\sum_{m=1}^{n_m} y_{mi} = 1, \quad \forall\ i = 1, \ldots, n_e.$$

- It is possible to incorporate these variables by introducing nonlinearity into the system - undesirable

$$\dot{\mathbf{x}}(\mathbf{p}, \mathbf{Y}, t) = \sum_{m=1}^{n_m} y_{mi}\Big(\mathbf{A}^{(m)}(t)\mathbf{x}(\mathbf{p}, t) + \mathbf{B}^{(m)}(t)\mathbf{p} + \mathbf{q}^{(m)}(t)\Big),\ \forall\, t \in I_i,\ i = 1, \ldots, n_e.$$

Exploit Linearity

- Introduce $n_m \times n_e$ dynamic LTV systems:

$$\dot{\mathbf{x}}_{mi}(\mathbf{p}, \mathbf{Z}, t) = \mathbf{A}^{(m)}(t)\mathbf{x}_{mi}(\mathbf{p}, \mathbf{Z}, t) + \mathbf{B}^{(m)}(t)\mathbf{p} + \mathbf{q}^{(m)}(t).$$

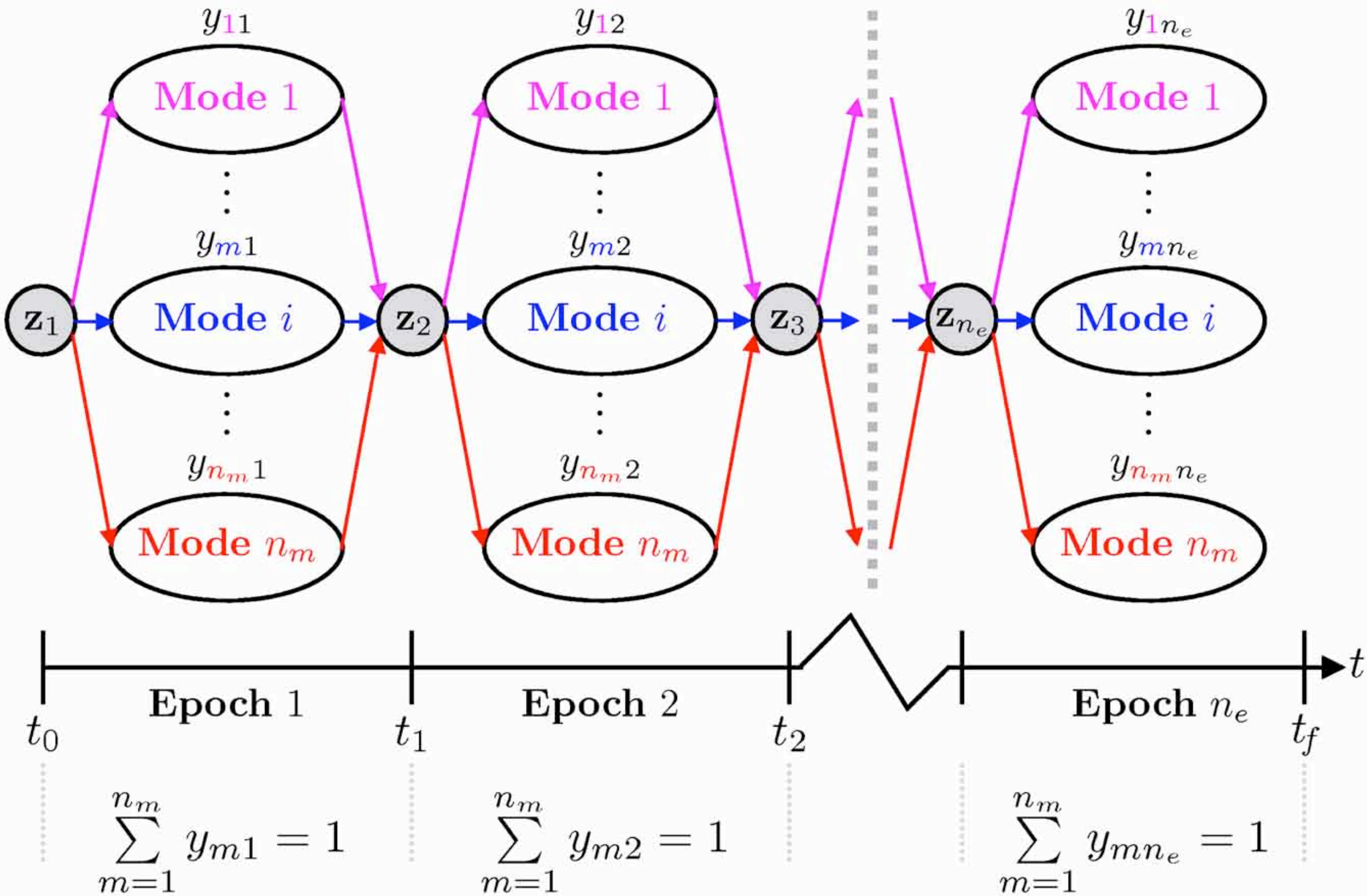
- Introduce $n_x \times n_e$ auxiliary parameters, $\mathbf{Z}$, to represent initial conditions

$$\begin{aligned} \mathbf{z}_1 &= \mathbf{E}_0\mathbf{p} + \mathbf{k}_0, \\ \mathbf{z}_i &= \mathbf{x}_{mi}(\mathbf{p}, \mathbf{Z}, t_i'), \quad \forall\, m \in M,\ i = 1, \ldots, n_e. \end{aligned}$$

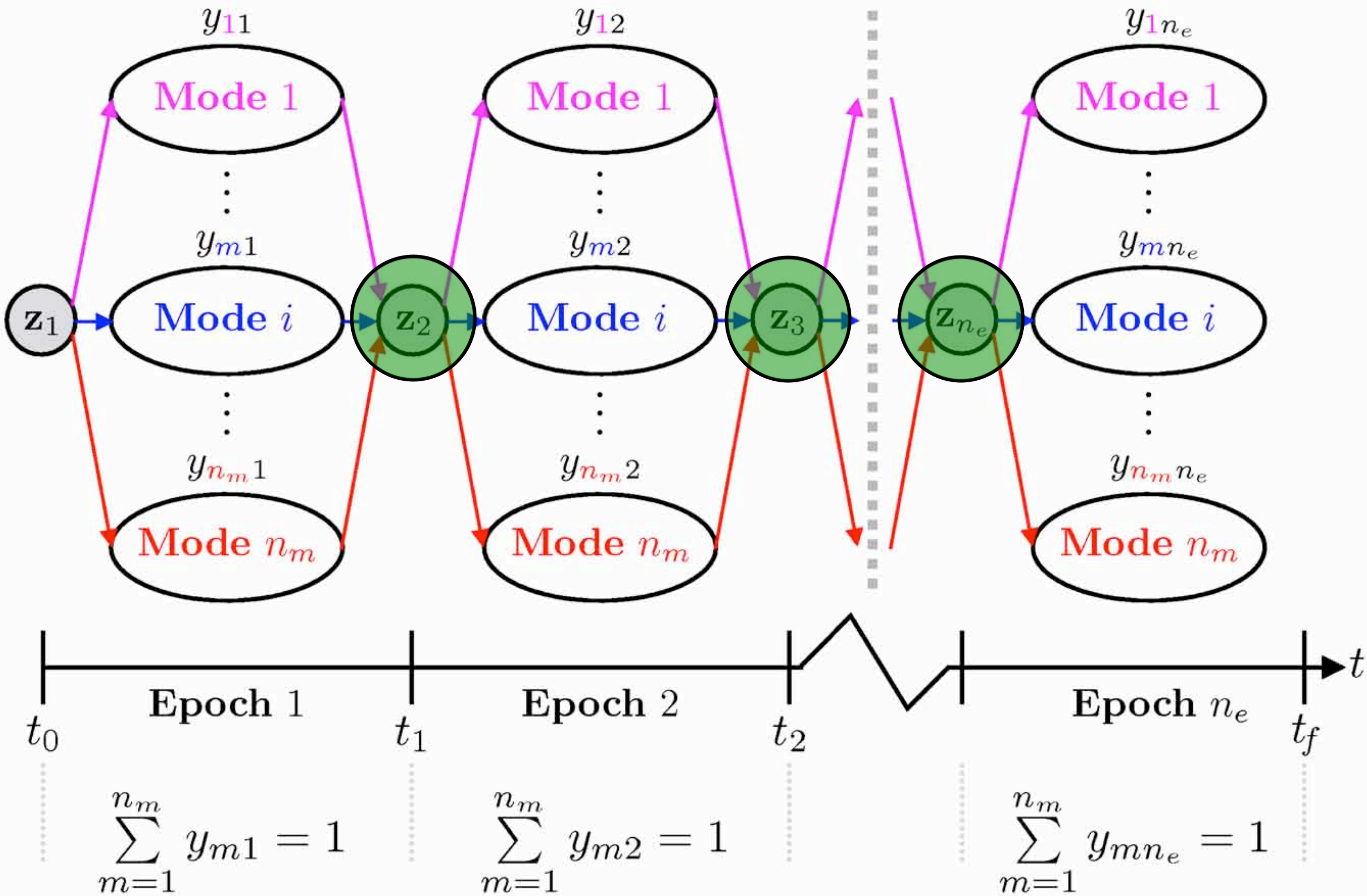
- Shift bilinearities to constraints:

$$\mathbf{z}_{i+1} = \mathbf{D}_i \left(\sum_{m=1}^{n_m} y_{mi}\mathbf{x}_{mi}(\mathbf{p}, \mathbf{Z}, t_i) \right) + \mathbf{E}_i\mathbf{p} + \mathbf{k}_i,\ \forall\, i = 1, \ldots, n_e - 1.$$

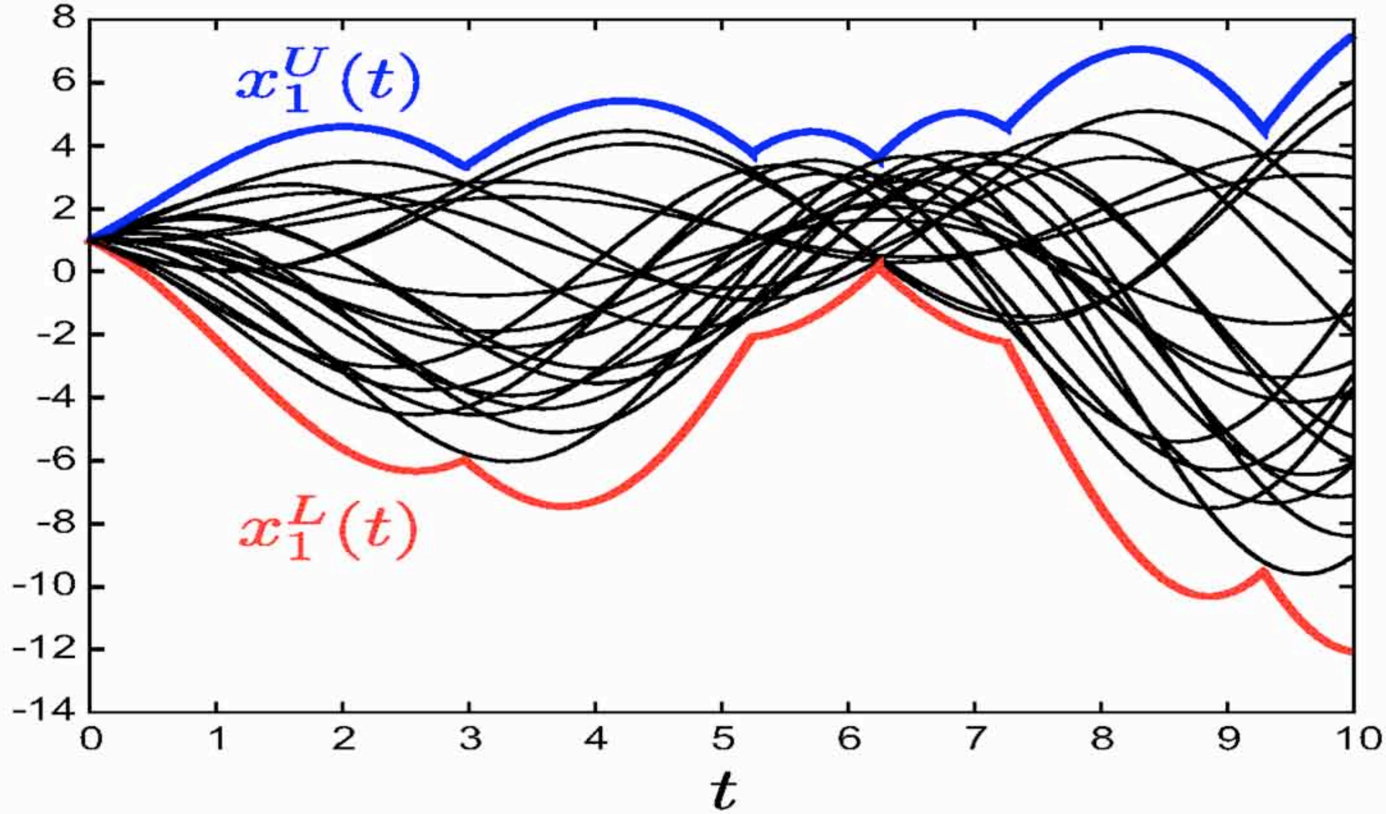
Hybrid Superstructure



Hybrid Superstructure



Bounding the set *Z* - (CLB)



$$\dot{x}_1 = 0.1x_1 - x_2 + p_1,$$

$$\dot{x}_2 = x_1 + 0.1x_2 - p2,$$

$$\mathbf{x}(\mathbf{p}, 0) = (1, 0),$$

$$\mathbf{p} \in [-2, 2]^2, t \in [0, 10].$$

Bounding *Z*
Cheap Linear Bounds (CLB)

Bounding Z
Cheap Linear Bounds (CLB)

Set of active modes:

$$\mathcal{A} \equiv \{\ (m, i) \mid \text{ mode } m \text{ is active in epoch } i\ \}$$

Bounding Z
Cheap Linear Bounds (CLB)

Set of active modes:

$$\mathcal{A} \equiv \{\ (m, i) \mid \text{ mode } m \text{ is active in epoch } i\ \}$$

1. Start at epoch 1.
2. Obtain exact bounds[1] for each individual mode, or the active mode in $\mathcal{A}$.
3. Obtain exact bounds for hybrid system at the end of epoch 1.

[1] A. B. Singer and P. I. Barton. Global Solution of Optimization Problems with Parameter-Embedded Linear Dynamic Systems. *Journal of Optimization Theory and Applications*, 121(3): 613–646, 2004.

Bounding Z
Cheap Linear Bounds (CLB)

Set of active modes:

$$\mathcal{A} \equiv \{\ (m,i) \mid \ \text{mode } m \text{ is active in epoch } i\ \}$$

1. Start at epoch 1.
2. Obtain exact bounds[1] for each individual mode, or the active mode in $\mathcal{A}$.
3. Obtain exact bounds for hybrid system at the end of epoch 1.
4. Start at epoch 2, with the initial conditions as parameters, with parameter bounds from Step 3.

[1] A. B. Singer and P. I. Barton. Global Solution of Optimization Problems with Parameter-Embedded Linear Dynamic Systems. *Journal of Optimization Theory and Applications*, 121(3): 613–646, 2004.

Bounding Z
Cheap Linear Bounds (CLB)

Set of active modes:

$$\mathcal{A} \equiv \{\ (m, i) \mid \text{ mode } m \text{ is active in epoch } i\ \}$$

1. Start at epoch 1.
2. Obtain exact bounds[1] for each individual mode, or the active mode in $\mathcal{A}$.
3. Obtain exact bounds for hybrid system at the end of epoch 1.
4. Start at epoch 2, with the initial conditions as parameters, with parameter bounds from Step 3.
5. Repeat Steps 2 and 3 for epoch 2.

[1] A. B. Singer and P. I. Barton. Global Solution of Optimization Problems with Parameter-Embedded Linear Dynamic Systems. *Journal of Optimization Theory and Applications*, 121(3): 613–646, 2004.

Bounding Z
Cheap Linear Bounds (CLB)

Set of active modes:

$$\mathcal{A} \equiv \{\ (m, i)\ |\ \text{mode } m \text{ is active in epoch } i\ \}$$

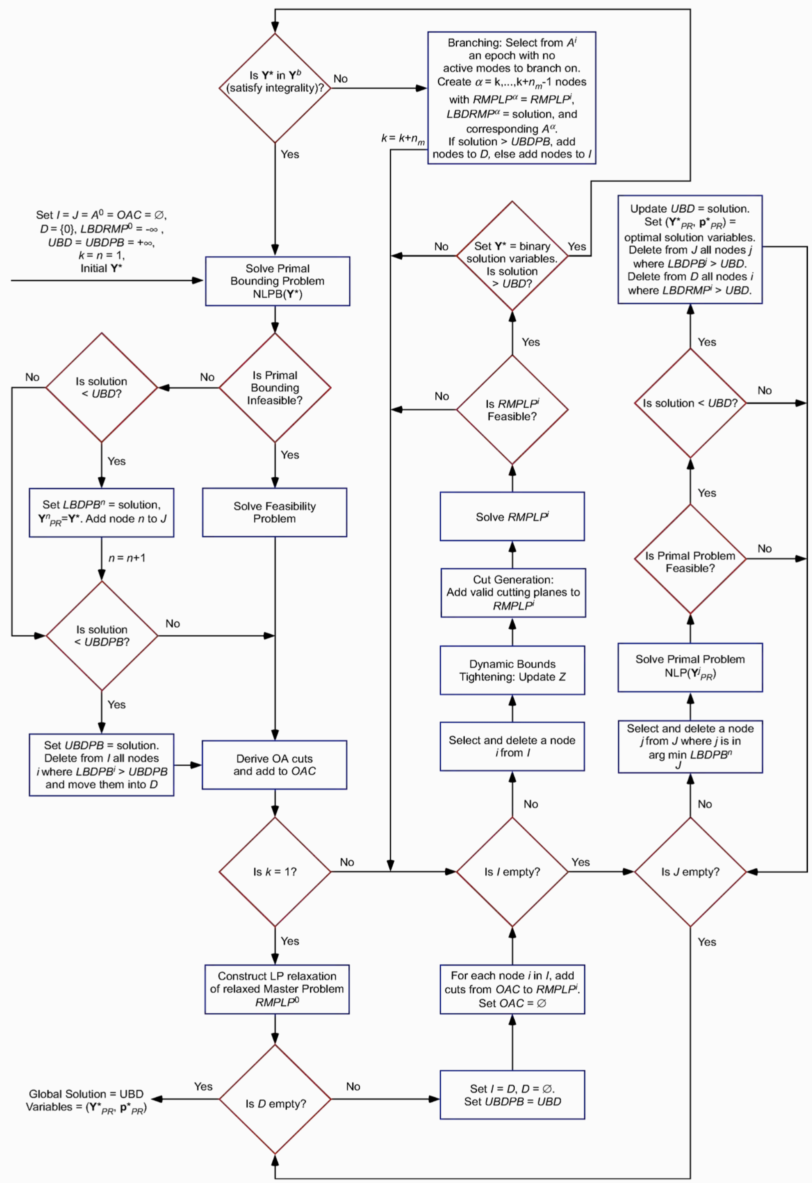
1. Start at epoch 1.
2. Obtain exact bounds[1] for each individual mode, or the active mode in $\mathcal{A}$.
3. Obtain exact bounds for hybrid system at the end of epoch 1.
4. Start at epoch 2, with the initial conditions as parameters, with parameter bounds from Step 3.
5. Repeat Steps 2 and 3 for epoch 2.
6. Repeat Steps 4 and 5 for each sequential epoch.

[1] A. B. Singer and P. I. Barton. Global Solution of Optimization Problems with Parameter-Embedded Linear Dynamic Systems. *Journal of Optimization Theory and Applications*, 121(3): 613–646, 2004.

Bounding *Z* LP Relaxation Bounds

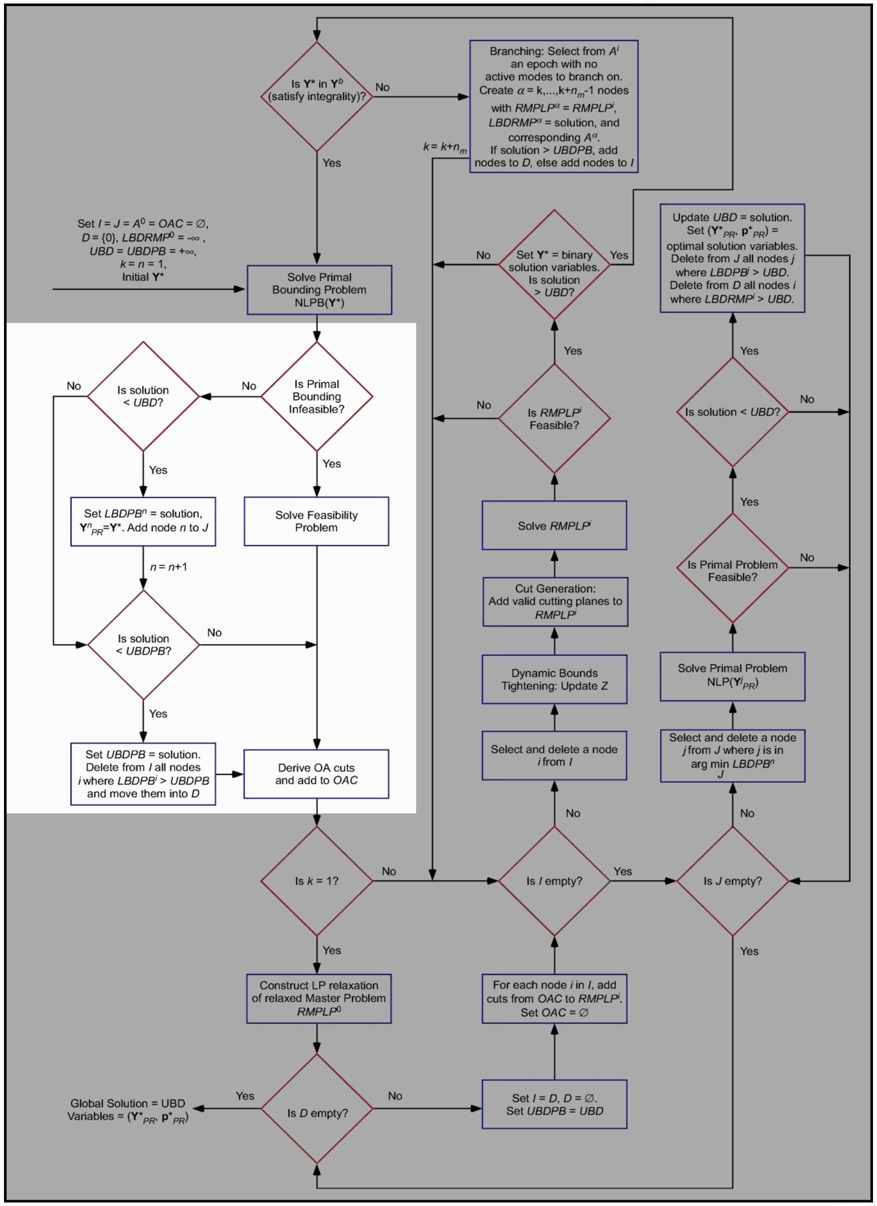
- For each epoch:
  - Formulate problem of obtaining upper and lower bounds for each state variable as an MILP.
  - Obtain bounds by solving the LP relaxation of the MILP.
- Able to incorporate physical insight as constraints
- Note: solving the MILP exactly gives the exact bounds

Generalized Branch-and-Cut



Generalized Branch-and-Cut

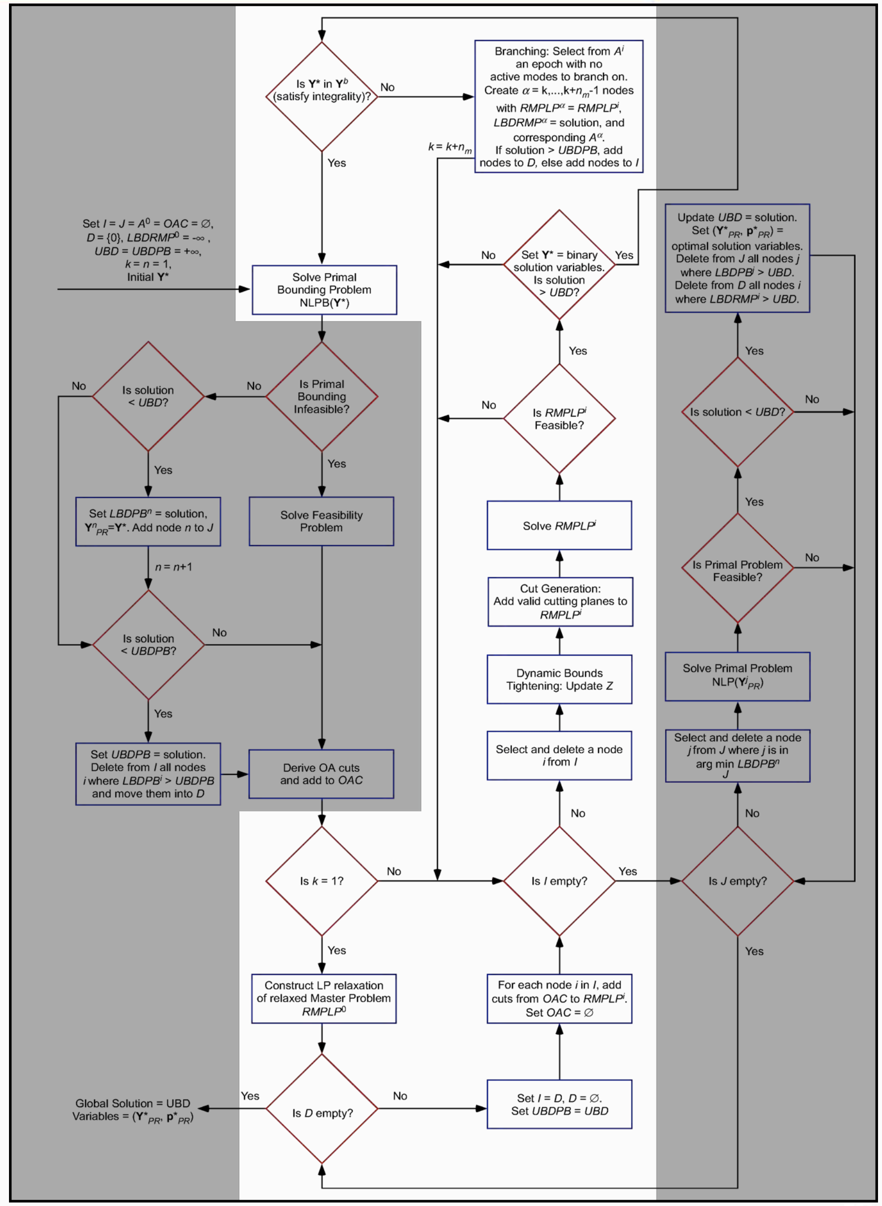
- **Primal Bounding Loop**
  - provides a valid lower bound for that particular integer realization



Generalized Branch-and-Cut

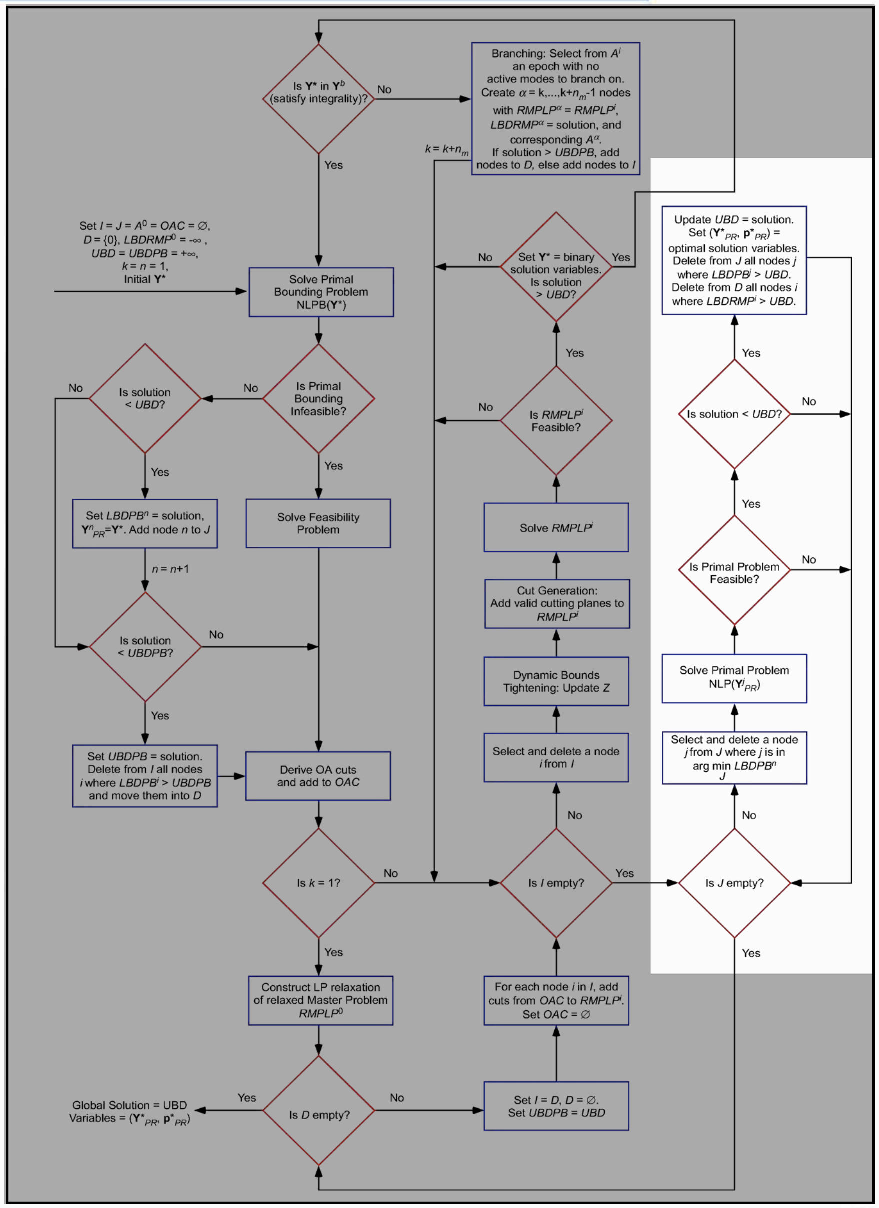
- **Primal Bounding Loop**
  - provides a valid lower bound for that particular integer realization

- **LP Relaxation Loop**
  - provides a valid lower bound for unexplored children nodes



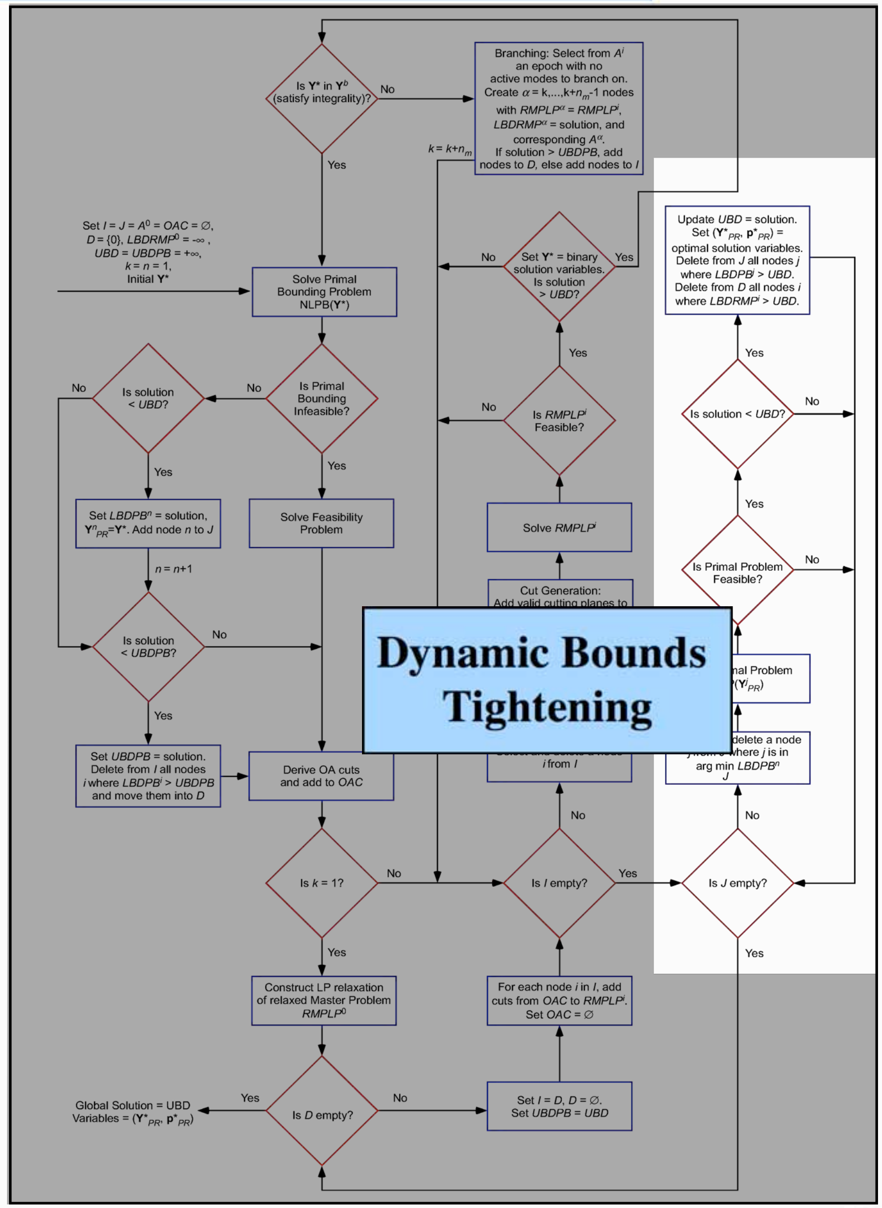
Generalized Branch-and-Cut

- **Primal Bounding Loop**
  - provides a valid lower bound for that particular integer realization
- **LP Relaxation Loop**
  - provides a valid lower bound for unexplored children nodes
- **Primal Loop**
  - provides a valid upper bound

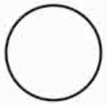


Generalized Branch-and-Cut

- **Primal Bounding Loop**
  - provides a valid lower bound for that particular integer realization
- **LP Relaxation Loop**
  - provides a valid lower bound for unexplored children nodes
- **Primal Loop**
  - provides a valid upper bound

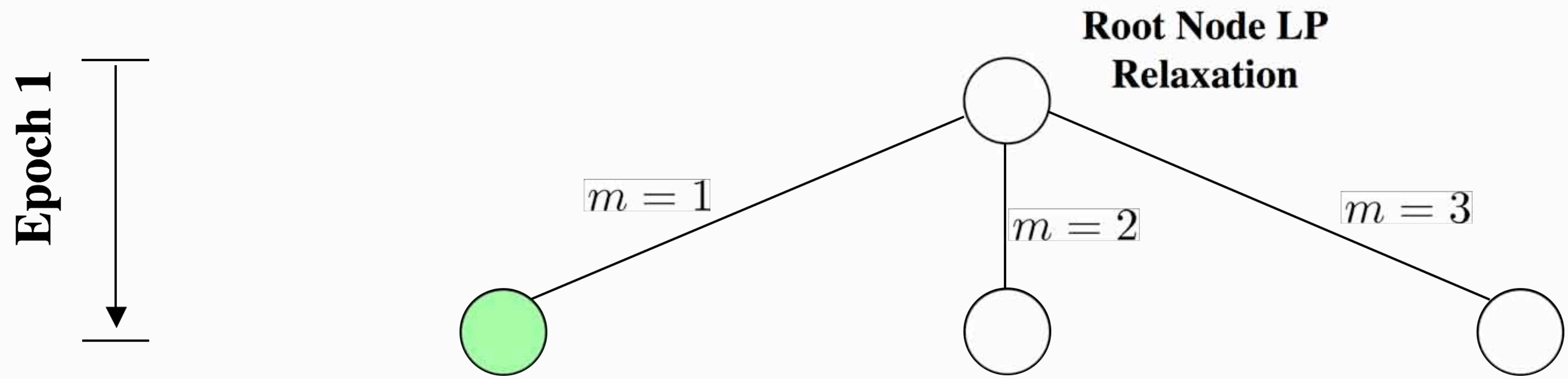


Dynamic Bounds Tightening

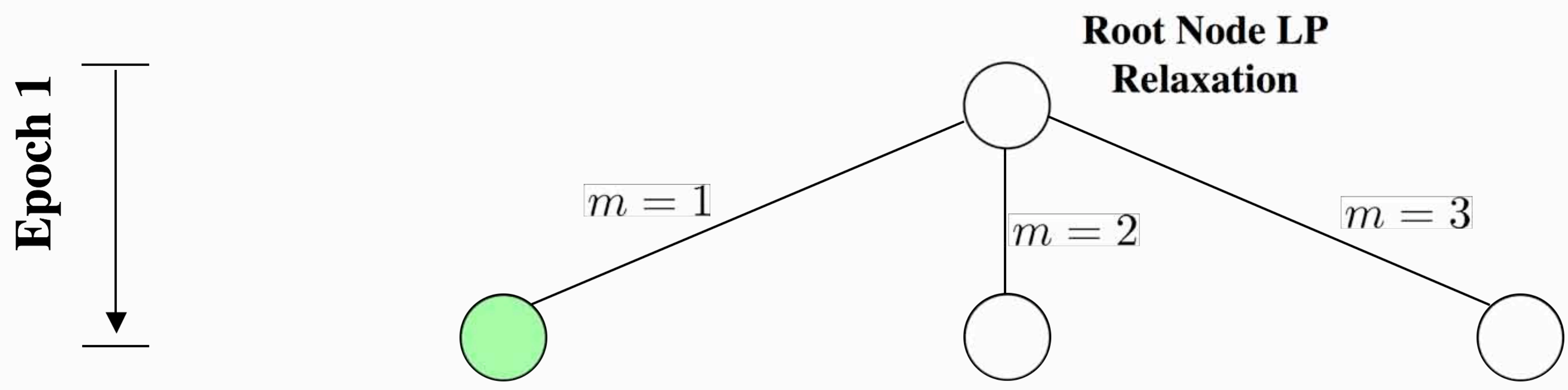


Root Node LP Relaxation

Dynamic Bounds Tightening



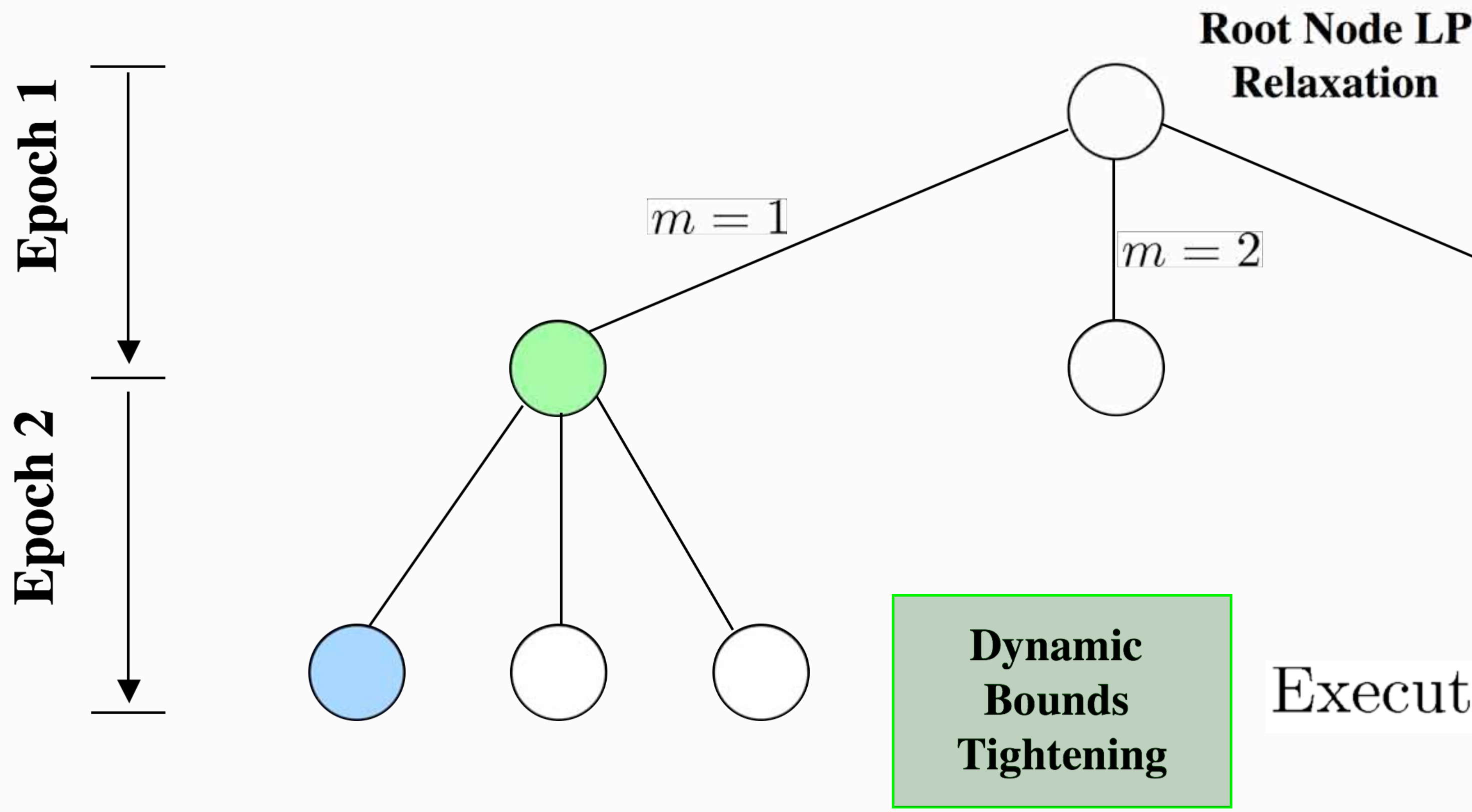
Dynamic Bounds Tightening



Dynamic Bounds Tightening

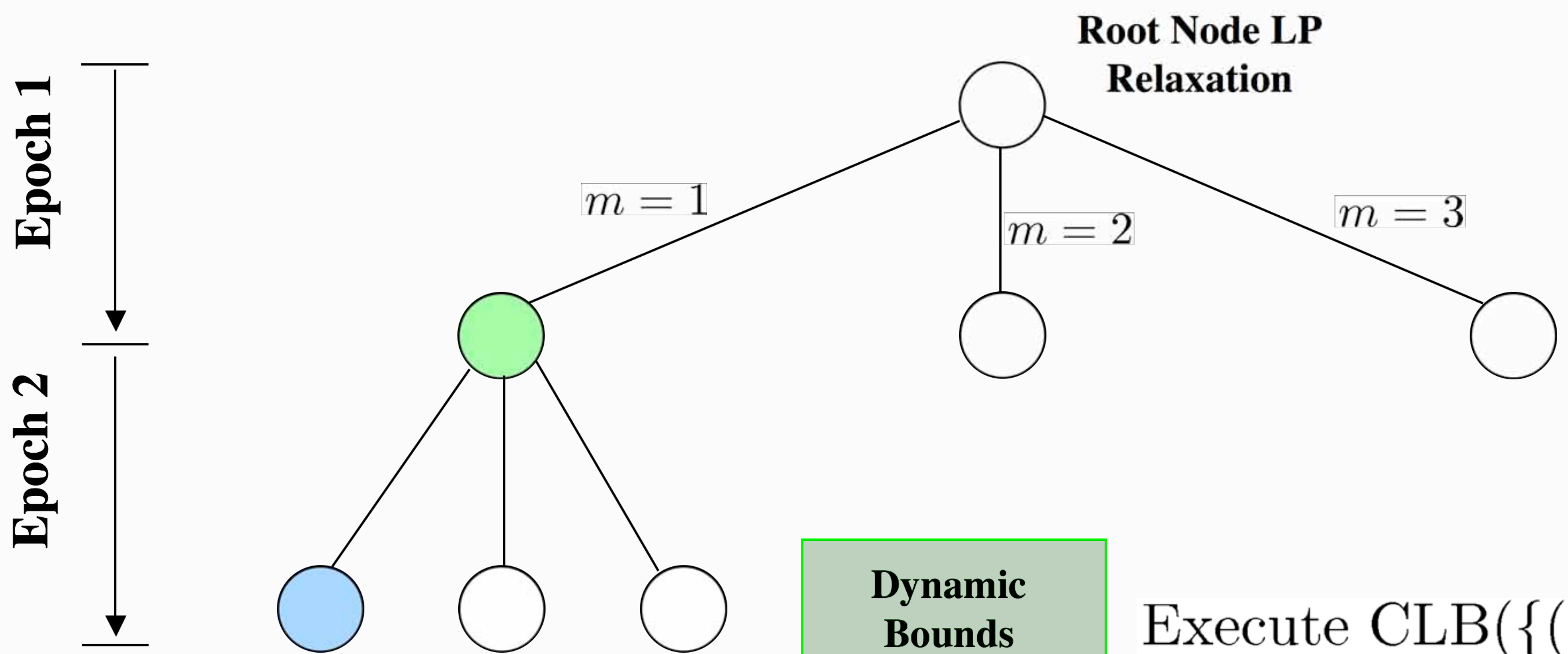
$\text{Execute CLB}(\{(1,1)\})$

Dynamic Bounds Tightening



Execute $\mathrm{CLB}(\{(1,1)\})$

Dynamic Bounds Tightening



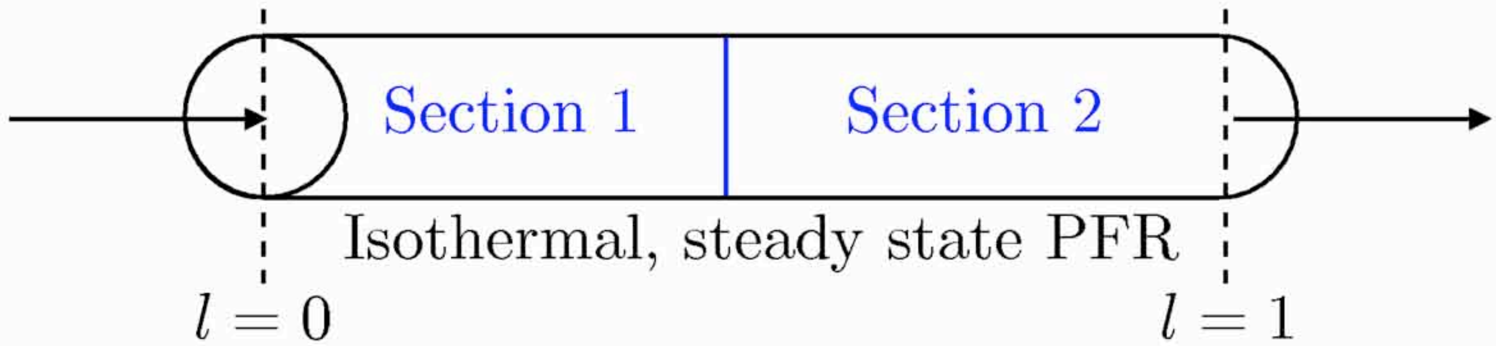
Dynamic Bounds Tightening

Execute $\mathrm{CLB}(\{(1,1)\})$

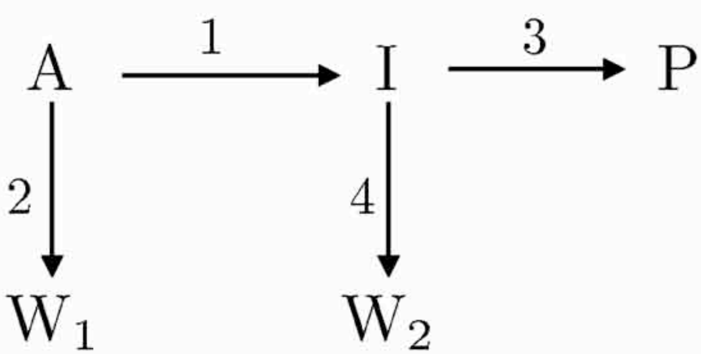
Dynamic Bounds Tightening

Execute $\mathrm{CLB}(\{(1,1),(1,2)\})$

Example: Catalyst Loading



Reaction scheme



Kinetics

$$
\begin{aligned}
\dot{x}_{\mathrm{A}} &= -(k_1 + k_2)x_{\mathrm{A}} \\
\dot{x}_{\mathrm{W}_1} &= k_2 x_{\mathrm{A}} \\
\dot{x}_{\mathrm{I}} &= k_1 x_{\mathrm{A}} - (k_3 + k_4)x_{\mathrm{I}} \\
\dot{x}_{\mathrm{W}_2} &= k_4 x_{\mathrm{I}} \\
\dot{x}_{\mathrm{P}} &= k_3 x_{\mathrm{I}}
\end{aligned}
$$

Example: Catalyst Loading

$x_{\mathrm{A}}(0) = 1000$ $\quad\quad$ $x_{\mathrm{P}}(1) - 0.01 x_{\mathrm{W}_1}(1) - 0.1 x_{\mathrm{W}_2}(1)$

Catalyst

	k_1	k_2	k_3	k_4
1	2.098	1.317	0.021	0.033
2	29.53	110.2	0.295	0.079
3	182.6	2325	1.826	0.143

	k_1/k_2	k_3/k_4
1	1.594	0.627
2	0.268	3.728
3	0.079	12.729

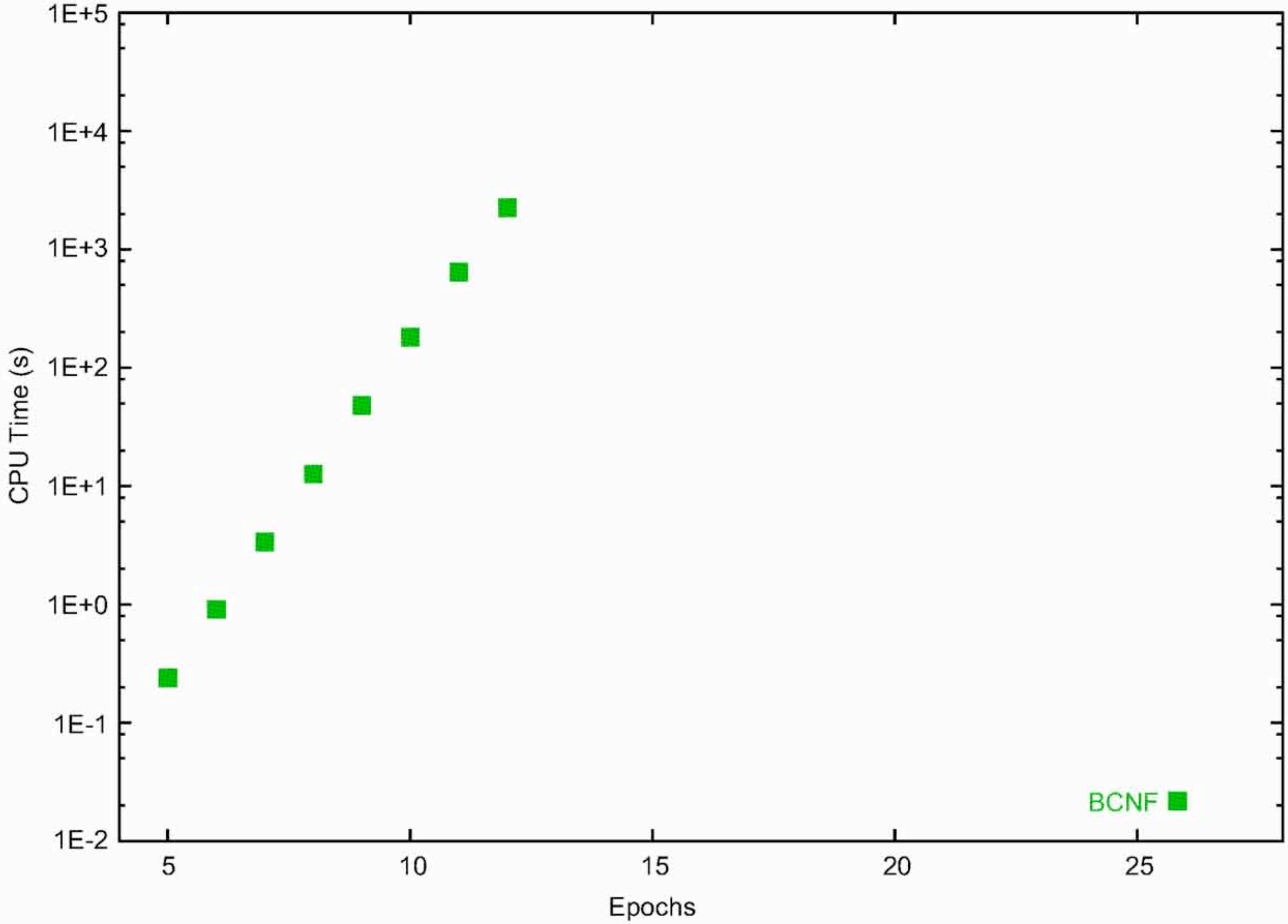
$$\max_{T_\mu} \quad x_{\mathrm{P}}(1) - 0.01 x_{\mathrm{W}_1}(1) - 0.1 x_{\mathrm{W}_2}(1)$$

$\longrightarrow$ $T_\mu = 1, 3 \qquad F = 290.4$

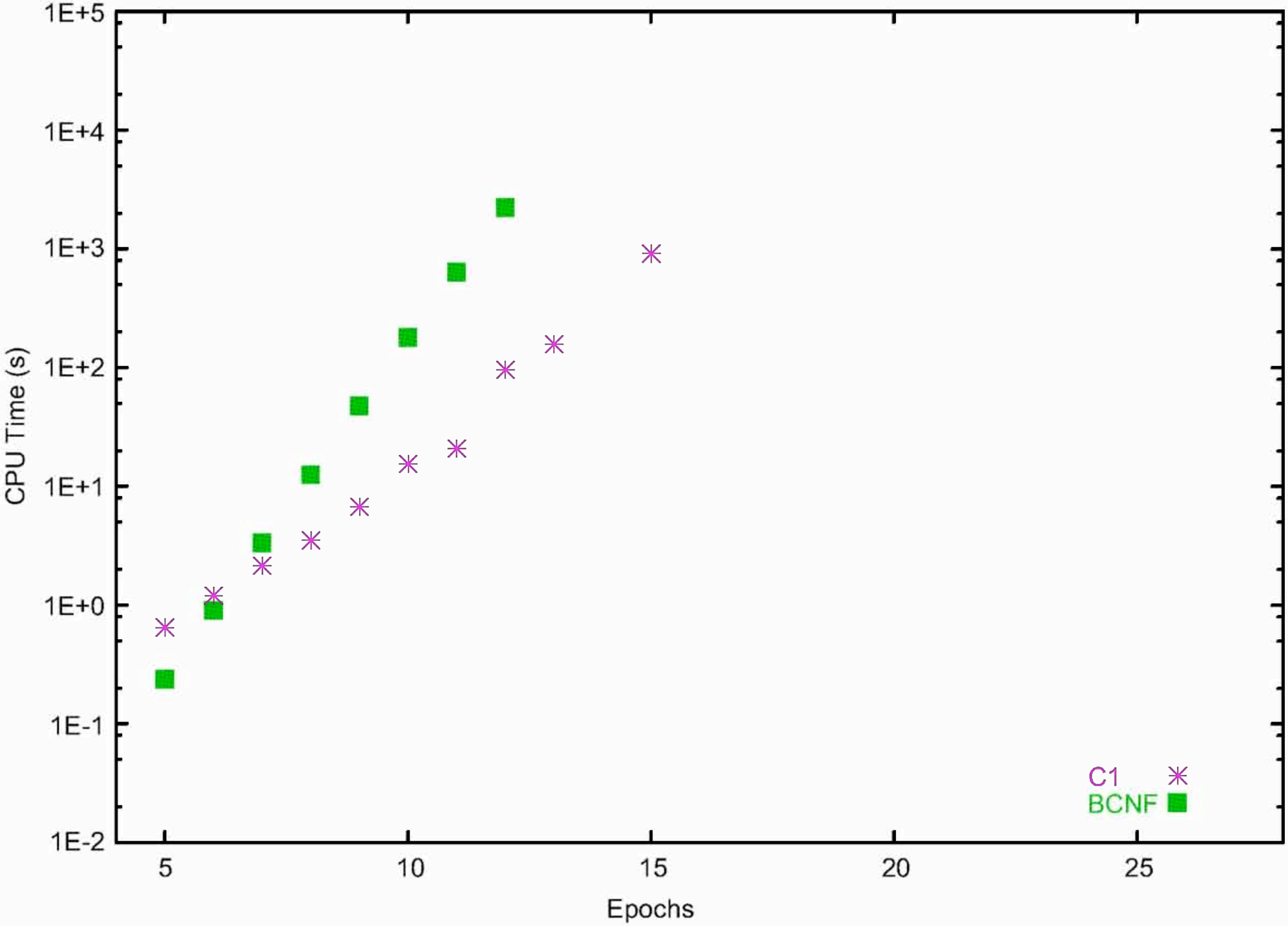
Algorithms Used

- EE – Explicit enumeration of all possible mode sequences
- BCDF – BC algorithm with dynamic bounds tightening, forward chronological branching
- BCNF – BC algorithm without dynamic bounds tightening, forward chronological branching
- C1 – CPLEX MILP solver with *Z* calculated with relaxed LP algorithm

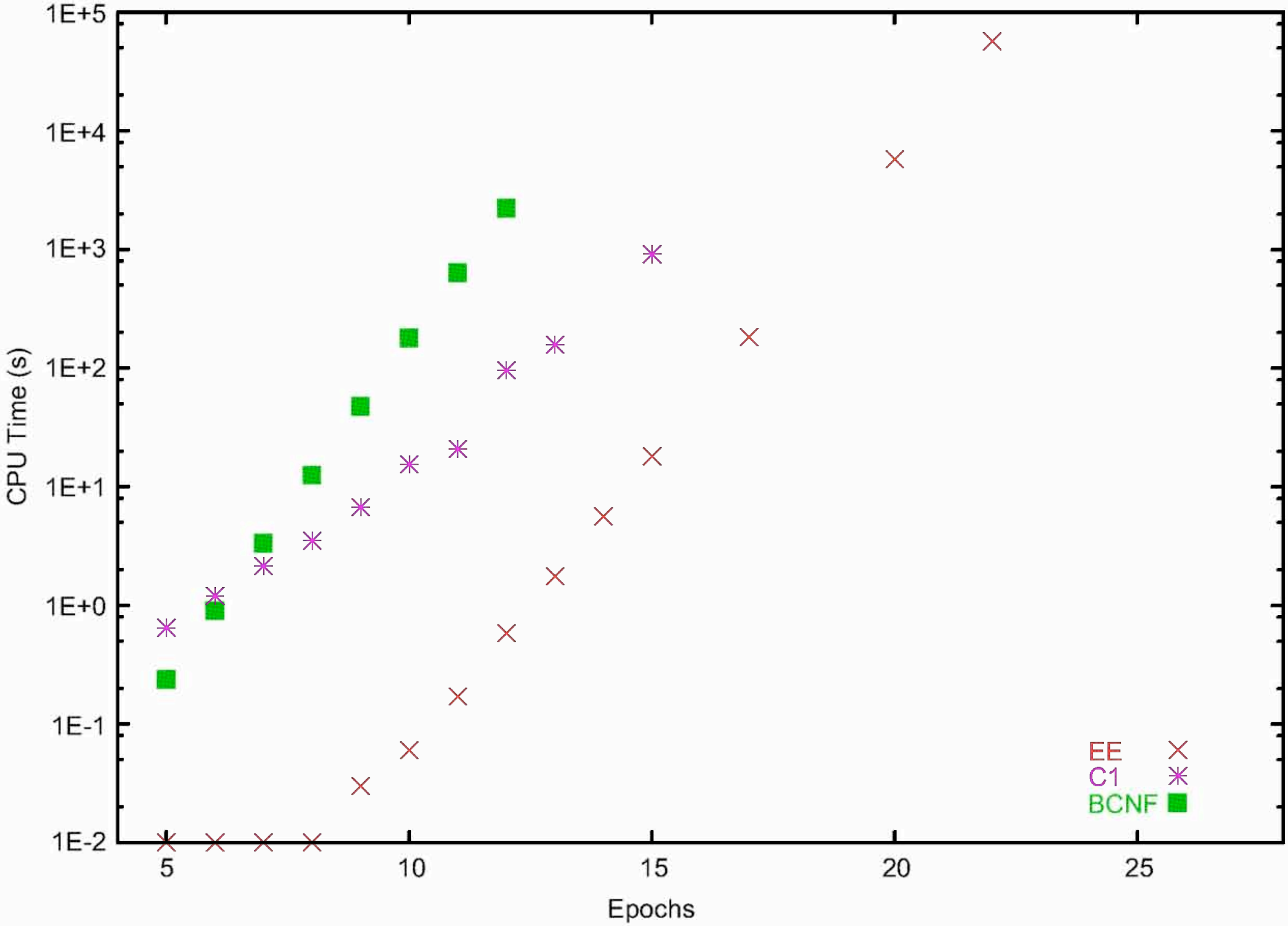
Results and Discussion



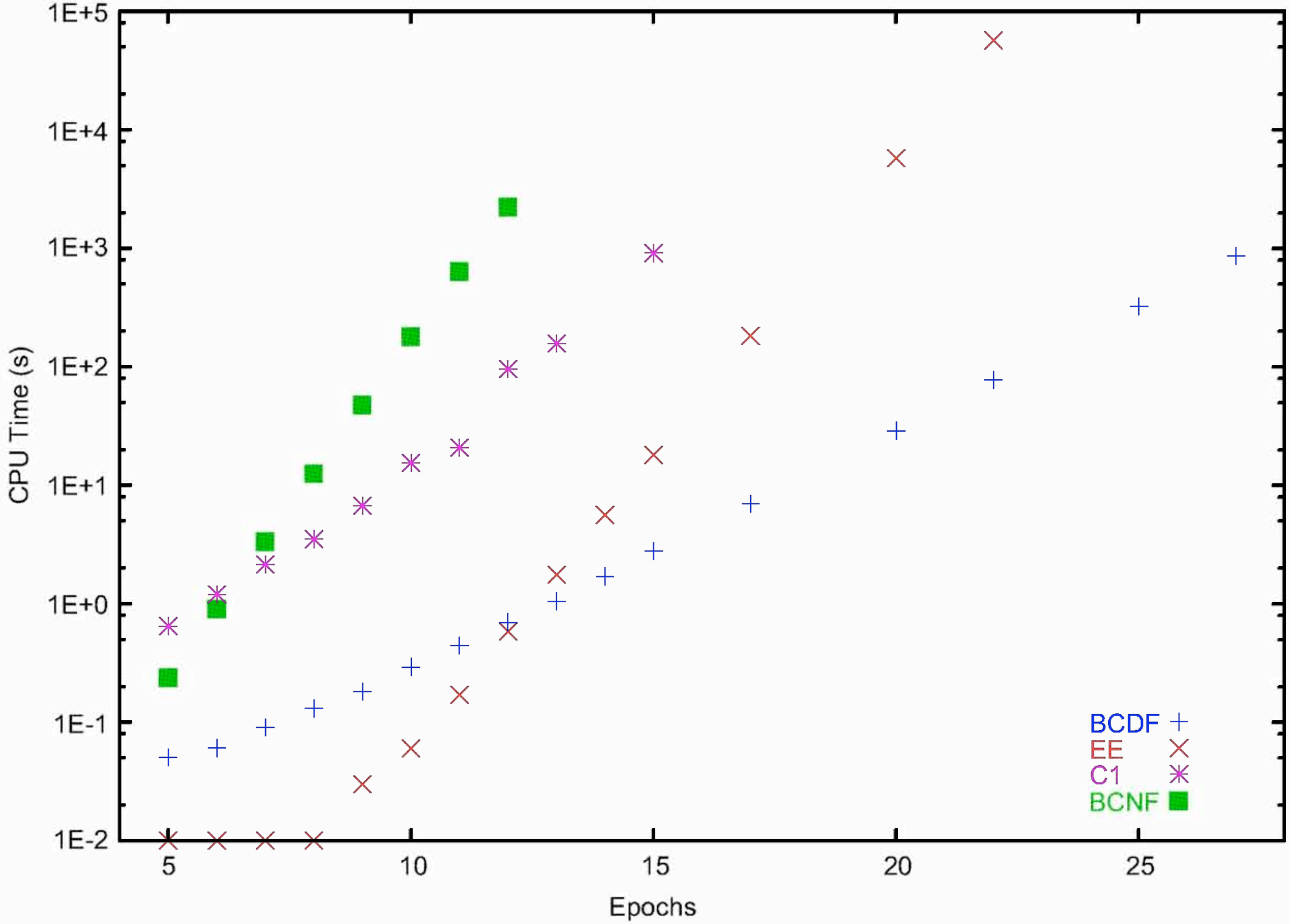
Results and Discussion



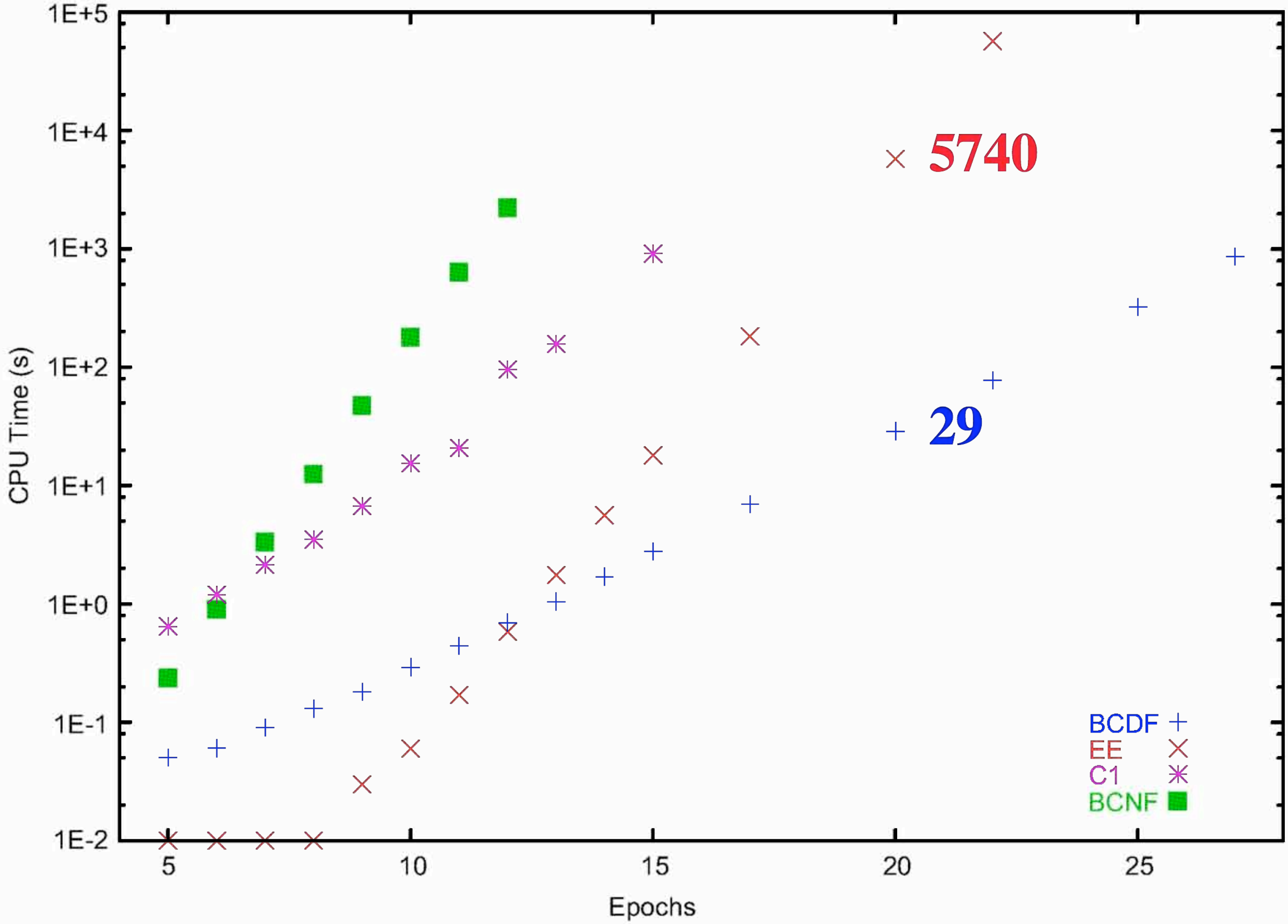
Results and Discussion



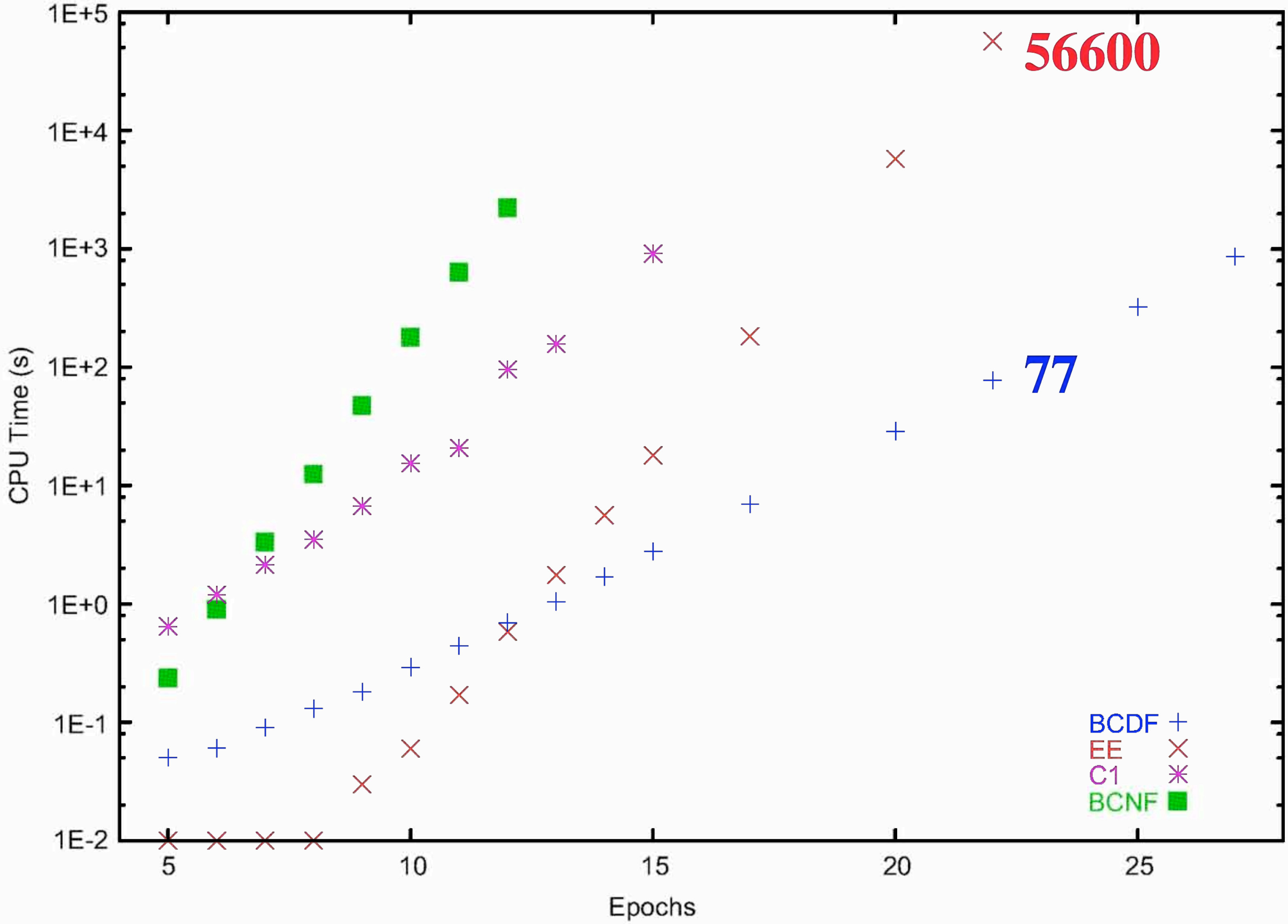
Results and Discussion



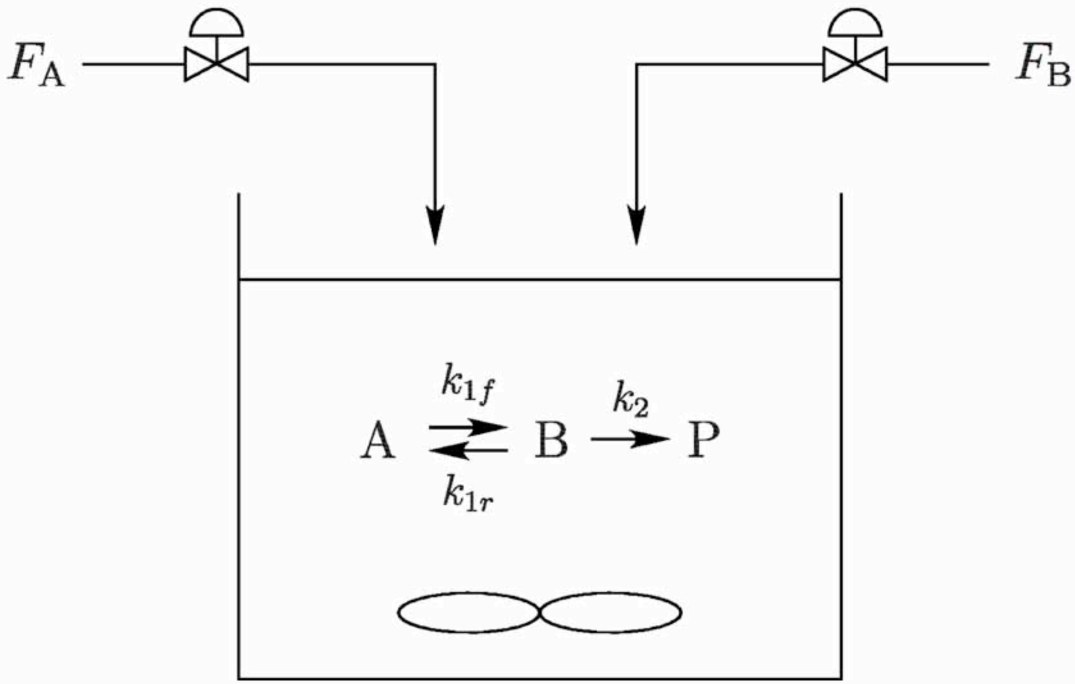
Results and Discussion



Results and Discussion



Determine the Optimal Feed Policy



Mode 1

F_A on, F_B off

$$\dot{N}_A = 4N_B - 2N_A + F_A$$
$$\dot{N}_B = 2N_A - 5N_B$$
$$\dot{N}_P = N_B$$

Mode 2

F_A off, F_B on

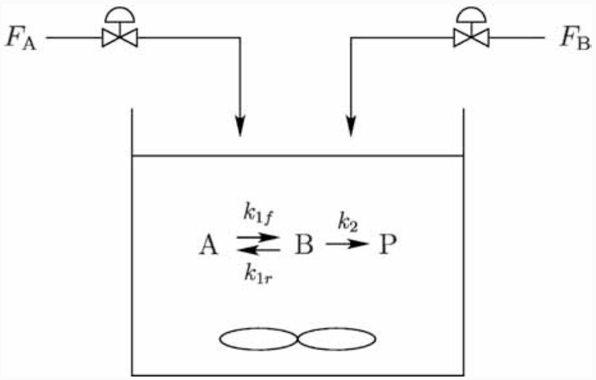
$$\dot{N}_A = 4N_B - 2N_A$$
$$\dot{N}_B = 2N_A - 5N_B + F_B$$
$$\dot{N}_P = N_B$$

Mode 3

F_A off, F_B off

$$\dot{N}_A = 4N_B - 2N_A$$
$$\dot{N}_B = 2N_A - 5N_B$$
$$\dot{N}_P = N_B$$

Determine the Optimal Feed Policy



Mode 1

F_A on, F_B off

$$\dot{N}_A = 4N_B - 2N_A + F_A$$
$$\dot{N}_B = 2N_A - 5N_B$$
$$\dot{N}_P = N_B$$

Mode 2

F_A off, F_B on

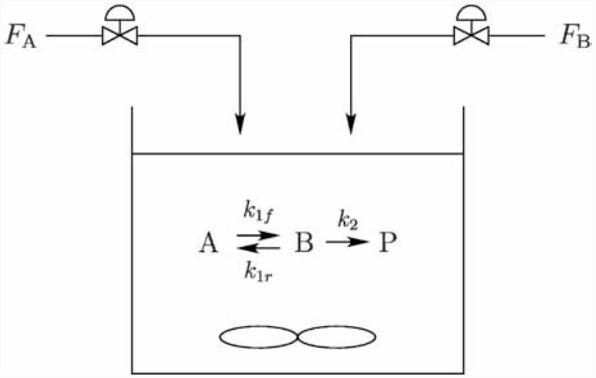
$$\dot{N}_A = 4N_B - 2N_A$$
$$\dot{N}_B = 2N_A - 5N_B + F_B$$
$$\dot{N}_P = N_B$$

Mode 3

F_A off, F_B off

$$\dot{N}_A = 4N_B - 2N_A$$
$$\dot{N}_B = 2N_A - 5N_B$$
$$\dot{N}_P = N_B$$

Determine the Optimal Feed Policy



Process Constraints:

1. **Inventory restriction on B**
2. **Minimum amount of product must be produced**
3. **Downstream processing constraint**

Mode 1

F_A on, F_B off

$$\dot{N}_A = 4N_B - 2N_A + F_A$$
$$\dot{N}_B = 2N_A - 5N_B$$
$$\dot{N}_P = N_B$$

Mode 2

F_A off, F_B on

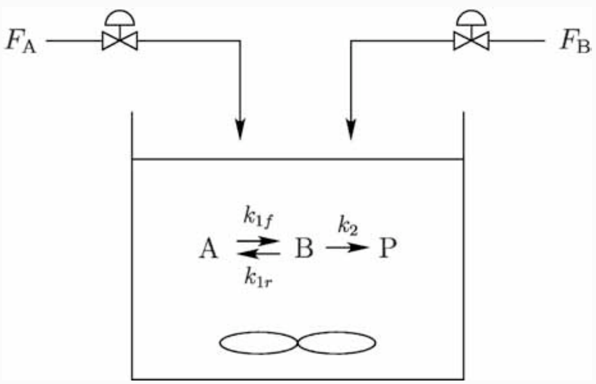
$$\dot{N}_A = 4N_B - 2N_A$$
$$\dot{N}_B = 2N_A - 5N_B + F_B$$
$$\dot{N}_P = N_B$$

Mode 3

F_A off, F_B off

$$\dot{N}_A = 4N_B - 2N_A$$
$$\dot{N}_B = 2N_A - 5N_B$$
$$\dot{N}_P = N_B$$

Determine the Optimal Feed Policy



Objective: Maximize selectivity

$$\max_{T_\mu, F_A, F_B} S = \frac{N_P(1)}{N_A(1) + N_B(1) + 1}$$

where F_A and F_B are control parametrized by piecewise constant controls.

Mode 1

F_A on, F_B off

$$\dot{N}_A = 4N_B - 2N_A + F_A$$
$$\dot{N}_B = 2N_A - 5N_B$$
$$\dot{N}_P = N_B$$

Mode 2

F_A off, F_B on

$$\dot{N}_A = 4N_B - 2N_A$$
$$\dot{N}_B = 2N_A - 5N_B + F_B$$
$$\dot{N}_P = N_B$$

Mode 3

F_A off, F_B off

$$\dot{N}_A = 4N_B - 2N_A$$
$$\dot{N}_B = 2N_A - 5N_B$$
$$\dot{N}_P = N_B$$

Results

n_e	(BCDR)	(BCNR)	(BCDF)	(BCNF)	(EEB)
5	1.41	1.30	1.32	1.43	2.93
6	5.10	4.60	4.67	5.58	12.1
7	17.5	15.6	16.4	19.3	43.8
8	59.0	59.0	56.8	74.8	167
9	193	201	195	268	619
10	665	677	671	905	2200
11	2180	2300	2240	3220	8050
12	7390	7700	8160	11300	27200
Exp. Coef.	1.2182	1.2431	1.2423	1.2788	1.3038
R^2 value	0.9999	0.9999	1.0000	0.9998	0.9997

Varying Transition Times Fixed Mode Sequence

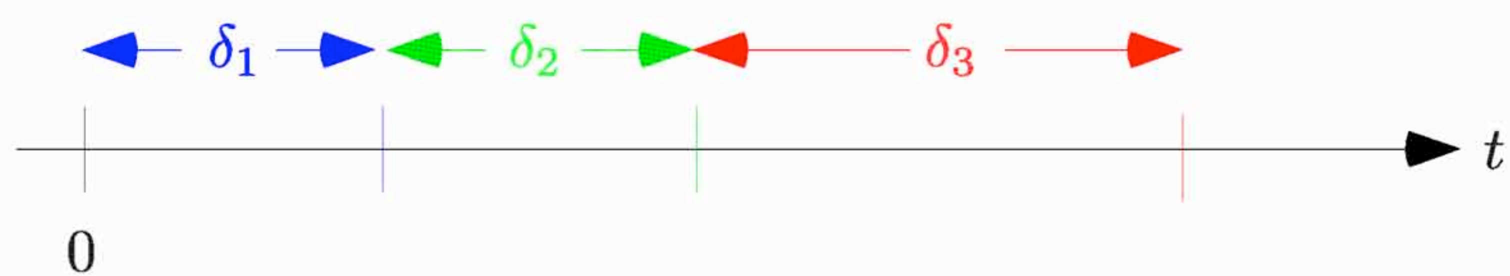
Consider the following problem

$$\min_{\mathbf{p}\in P,\boldsymbol{\delta}\in\Delta} \; F(\mathbf{p},\boldsymbol{\delta}) = \sum_{i=1}^{n_e} \left\{ \phi_i\Big(\mathbf{x}(\mathbf{p},\boldsymbol{\delta},\alpha_i),\mathbf{p},\boldsymbol{\delta}\Big) + \int_{\sigma_i(\boldsymbol{\delta})}^{\tau_i(\boldsymbol{\delta})} f_i\big(\mathbf{x},\mathbf{p},\boldsymbol{\delta},t\big)\,\mathrm{d}t \right\},$$

subject to the following point and isoperimetric constraints,

$$\mathbf{G}(\mathbf{p},\boldsymbol{\delta}) = \sum_{i=1}^{n_e} \left\{ \boldsymbol{\eta}_i\Big(\mathbf{x}(\mathbf{p},\boldsymbol{\delta},\beta_i),\mathbf{p},\boldsymbol{\delta}\Big) + \int_{\sigma_i(\boldsymbol{\delta})}^{\tau_i(\boldsymbol{\delta})} \mathbf{g}_i\big(\mathbf{x},\mathbf{p},\boldsymbol{\delta},t\big)\,\mathrm{d}t \right\} \;\leq\; \mathbf{0},$$

where $\mathbf{x}(\mathbf{p},\boldsymbol{\delta},t)$ is given by the solution of the embedded linear hybrid system.



CPET Transformation

- Introduce a new control v where

$$\frac{\mathrm{d}t}{\mathrm{d}s} = v$$

$$\Rightarrow \frac{\mathrm{d}x}{\mathrm{d}s} = \frac{\mathrm{d}x}{\mathrm{d}t}\frac{\mathrm{d}t}{\mathrm{d}s} = v\dot{x}$$

- The transformed problem becomes

$$\min_{p,v} \tilde{F}(p,v) = x(p,v,1)$$

$$\text{s.t. } \dot{x}(p,v,s) = v(-2x(p,v,s) + p)$$

$$x(p,v,0) = 1$$

[1] H. W. J. Lee, K. L. Teo, V. Rehbock, and L. S. Jennings. Control parametrization enhancing technique for time optimal control problems, *Dynamic Systems & Applications*, 6:243 – 262, 1997.

CPET Transformation

- Introduce a new control v where

$$\frac{\mathrm{d}t}{\mathrm{d}s} = v$$

$$\Rightarrow \frac{\mathrm{d}x}{\mathrm{d}s} = \frac{\mathrm{d}x}{\mathrm{d}t}\frac{\mathrm{d}t}{\mathrm{d}s} = v\dot{x}$$

- The transformed problem becomes

$$\min_{p,v} \tilde{F}(p,v) = x(p,v,1)$$

$$\text{s.t. } \dot{x}(p,v,s) = v(-2x(p,v,s)+p)$$

$$x(p,v,0) = 1$$

[1] H. W. J. Lee, K. L. Teo, V. Rehbock, and L. S. Jennings. Control parametrization enhancing technique for time optimal control problems, *Dynamic Systems & Applications*, 6:243 – 262, 1997.

CPET Transformation

- Introduce a new control v where

$$\frac{\mathrm{d}t}{\mathrm{d}s} = v$$

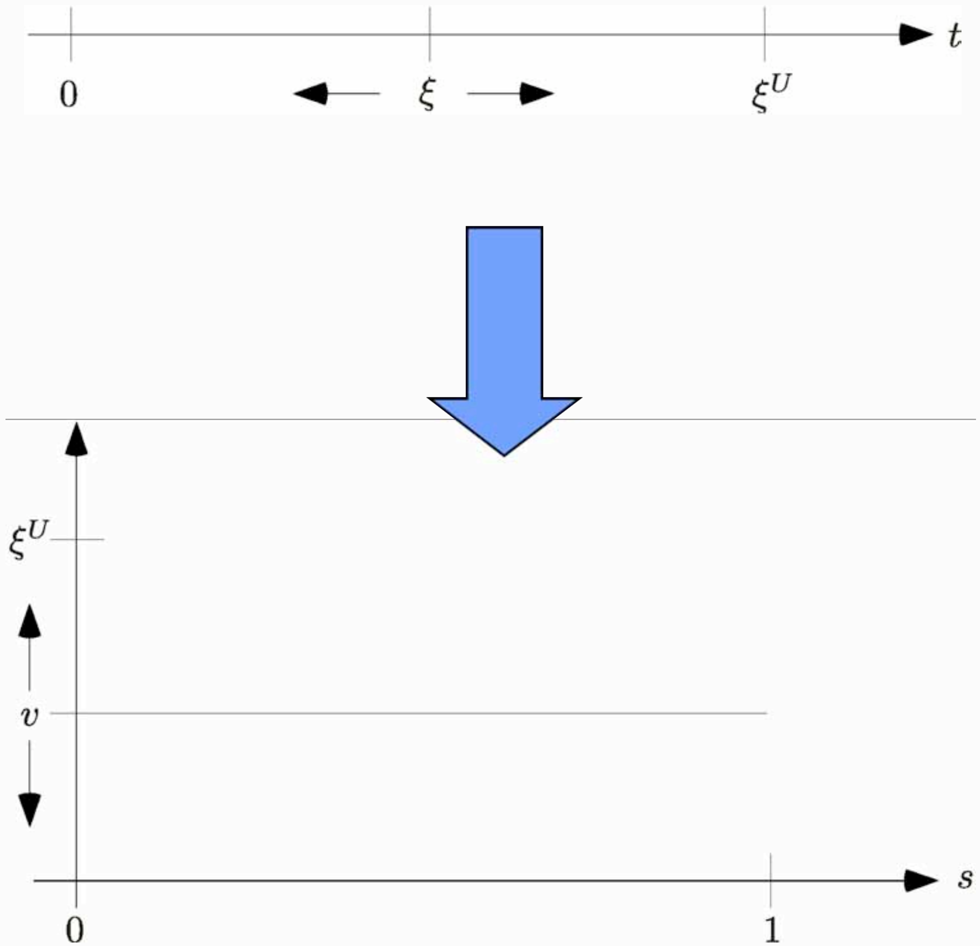
$$\Rightarrow \frac{\mathrm{d}x}{\mathrm{d}s} = \frac{\mathrm{d}x}{\mathrm{d}t}\frac{\mathrm{d}t}{\mathrm{d}s} = v\dot{x}$$

- The transformed problem becomes

$$\min_{p,v} \tilde{F}(p,v) = x(p,v,1)$$

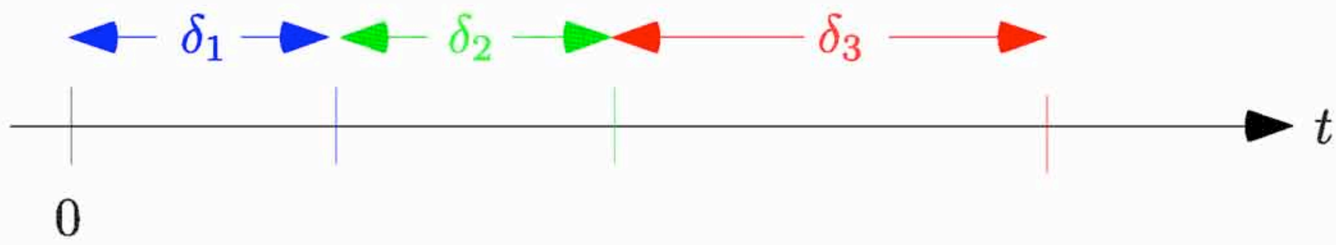
$$\text{s.t. } \dot{x}(p,v,s) = v(-2x(p,v,s)+p)$$

$$x(p,v,0) = 1$$

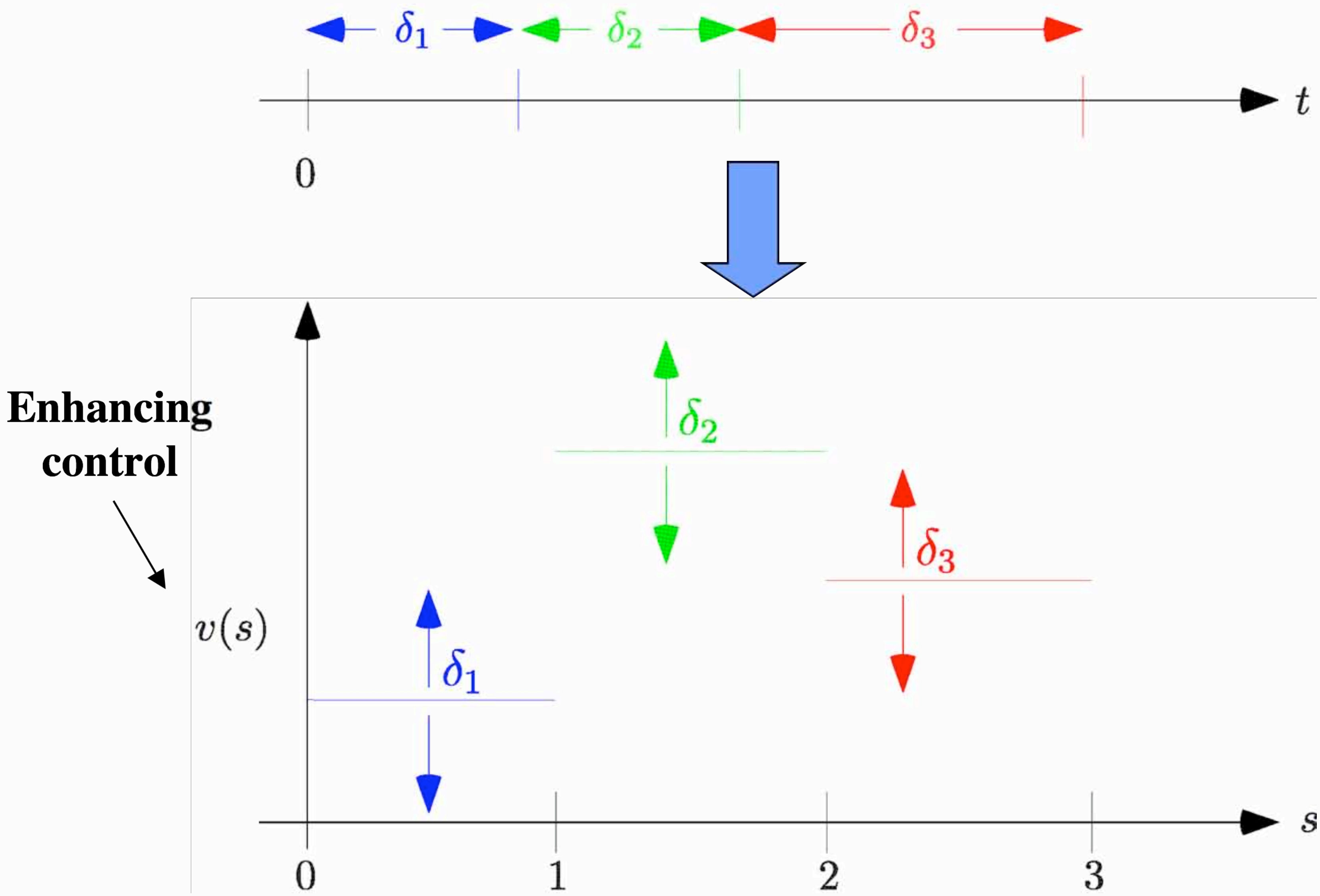


[1] H. W. J. Lee, K. L. Teo, V. Rehbock, and L. S. Jennings. Control parametrization enhancing technique for time optimal control problems, *Dynamic Systems & Applications*, 6:243 – 262, 1997.

For Multistage Systems



For Multistage Systems



Relaxation Theory

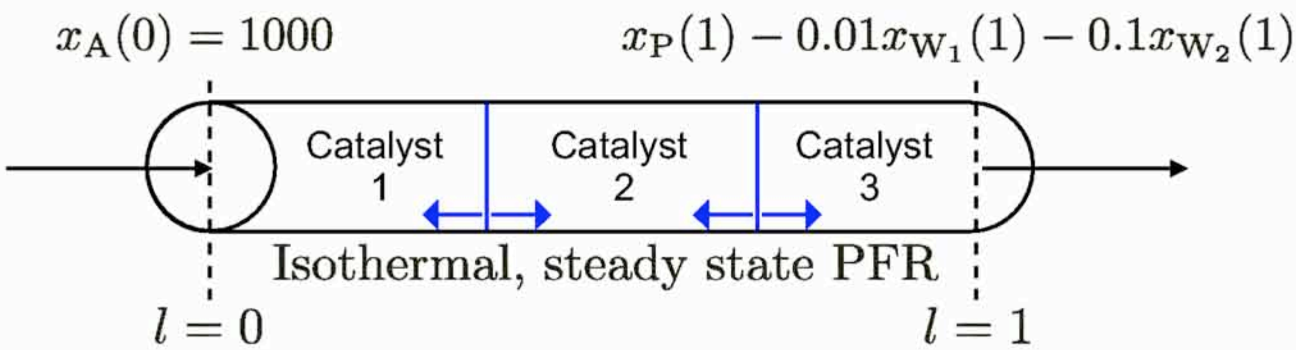
- CPET transforms linear hybrid system with variable epochs into a nonlinear hybrid system with fixed epochs

$$\hat{\mathbf{x}}'(\mathbf{p},\boldsymbol{\delta},s) = v(\boldsymbol{\delta},s)\Big(\hat{\mathbf{A}}^{(m_i^*)}(\boldsymbol{\delta},s)\hat{\mathbf{x}}(\mathbf{p},\boldsymbol{\delta},s) + \hat{\mathbf{B}}^{(m_i^*)}(\boldsymbol{\delta},s)\mathbf{p} + \hat{\mathbf{q}}^{(m_i^*)}(\boldsymbol{\delta},s)\Big)$$

- To construct convex relaxations of the Bolza-type functionals:
  - Need state bounds
  - Need convex and concave relaxations for the states and derivatives
  - Apply previous theory to fixed sequence of modes

Catalyst Loading Problem Variable Width Sections

- Catalyst loading problem:



$$\delta_1 + \delta_2 + \delta_3 = 1$$

- The optimal solution is 314.2, at the point $\delta^* =$ (0.363,0.020,0.617), solution time 560 s, 481 nodes, using $\zeta = (\hat{\mathrm{x}}(\boldsymbol{\delta}^L, s), \boldsymbol{\delta}^L)$

- For comparison, solution for 20 epochs is 308.6, best solution time 7070 s

Classification of Problems

Dynamics	Function(al)s	Event Timings	Mode Sequence	Approaches	Demonstrated
LTV	Affine	Fixed	Fixed	LP	yes
LTV	Nonlinear	Fixed	Fixed	Convex NLP Local NLP Nonconvex NLP	yes yes yes
LTV	Affine	Fixed	Both	MILP	yes
LTV	Nonlinear	Fixed	Both	Convex MINLP Nonconvex MINLP	yes yes
LTV	Nonlinear	Varying	Fixed	Local NLP Nonconvex NLP	yes yes
LTV	Nonlinear	Varying	Controlled	Nonconvex MINLP	no
LTV	Nonlinear	Varying	Autonomous	Nonsmooth NLP Nonconvex MINLP	stochastic no
LTV	Nonlinear	Varying	Both	Nonconvex MINLP	no
Nonlinear	Nonlinear	Varying	Fixed	Local NLP Nonconvex NLP	yes yes
Nonlinear	Nonlinear	Varying	Controlled	Nonconvex MINLP	no
Nonlinear	Nonlinear	Varying	Autonomous	Nonsmooth NLP Nonconvex MINLP	stochastic no
Nonlinear	Nonlinear	Varying	Both	Nonconvex MINLP	no

Conclusions

- Models of hybrid systems
  - Hybrid phenomena
  - Hybrid automaton
  - Complementarity systems and generalizations
  - Existence and uniqueness
- Simulation and sensitivity analysis
  - Dealing with implicit transitions
  - Existence and uniqueness
- Optimization
  - Nonsmooth and even discontinuous
  - Finding the optimal mode sequence is a challenging problem
  - Mixed-integer reformulation => global optimization techniques
  - Nonsmooth optimization techniques (local)

Acknowledgements

- Taeshin Park, William Feehery, Santos Galán, Julio Banga, Adam Singer, Cha Kun Lee, Benoit Chachuat, Alexander Mitsos, Mehmet Yunt, Vibhu Saxena
- U.S. Department of Energy, National Science Foundation and Army Research Office for financial support