Hybrid Control and Switched Systems

Lecture #2
How to describe a hybrid system?
Formal models for hybrid system

João P. Hespanha

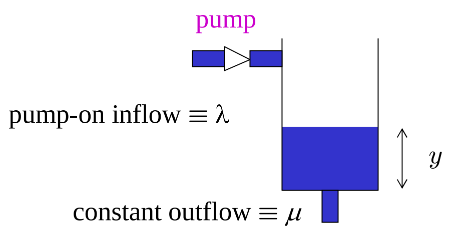
University of California
at Santa Barbara

UCSB

Summary

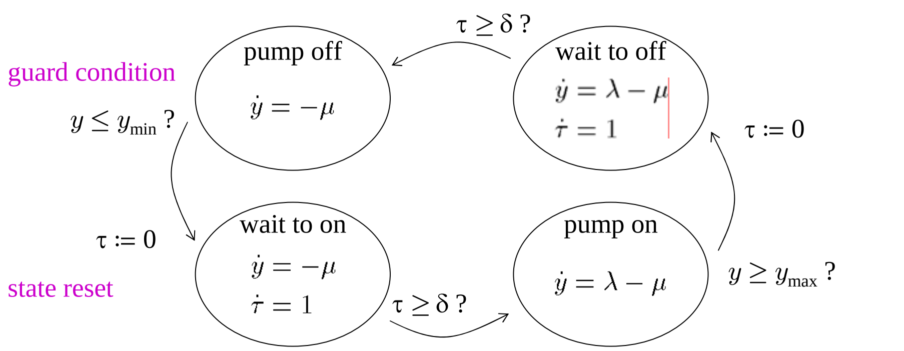
1. **Formal models for hybrid systems**:
   - Finite automata
   - Differential equations
   - Hybrid automata
   - Open hybrid automaton
2. Nondeterministic vs. stochastic systems
   - Non-deterministic hybrid automata
   - Stochastic hybrid automata

Example #5: Multiple-tank system



goal ≡ prevent the tank from emptying or filling up

δ ≡ delay between command is sent to pump and the time it is executed



How to formally describe this hybrid system?

Deterministic finite automaton

automata M:

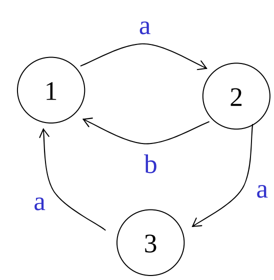
- $\mathcal{Q} := \{q_1, q_2, \ldots, q_n\}$ ≡ finite set of states
- $\Sigma := \{\text{a, b, c,} \ldots\}$ ≡ finite set of input symbols (alphabet)
- $\Phi : \mathcal{Q} \times \Sigma \to \mathcal{Q}$ ≡ transition function

Example:

$q \in \mathcal{Q}$	$s \in \Sigma$	$\Phi(q,s)$
1	a	2
1	b	↻
2	a	3
2	b	1
3	a	1
3	b	↻
↻	a/b	↻

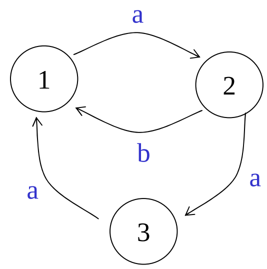
blocking state

Graph representation:



- one node per state (except for blocking state ↻)
- one directed edge (arrow) from q to $\Phi(q, s)$ with label s for each pair (q, s) for which $\Phi(q, s) \neq$ ↻

Deterministic finite automaton



Notation: Given set $\mathcal{A}$

string $\equiv$ finite sequence of symbols

$\in \equiv$ empty string

$\mathcal{A}^*$ $\equiv$ set of all strings of symbols in set $\mathcal{A}$

e.g., $\mathcal{A}$ = {a, b}

s = abbbbaab $\in \mathcal{A}^*$

s[3] = b (3rd element)

$|\, s \,|$ = 8 (length of string)

Definition: Given • initial state $q_1 \in \mathcal{Q}$

• set of final states $\mathcal{F} \subset \mathcal{Q}$

M ***accepts*** a string $s \in \Sigma^*$ with length $n := |\, s \,|$ if

there exists a sequence of states $q \in \mathcal{Q}^*$ with length $|\, q \,| = n+1$ (***execution***) such that

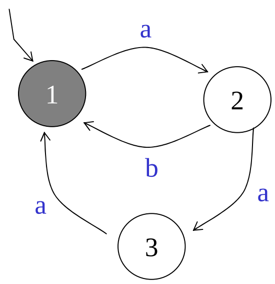
1. $q[1] = q_1$ (starts at initial state)
2. $q[i+1] = \Phi(q[i], s[i])$, $i \in \{1,2,\ldots,n\}$ (follows arrows with correct label)
3. $q[n+1] \in \mathcal{F}$ (ends in set of final states)

Definition: ***language*** accepted by automaton *M*

L(*M*) := { set of all strings accepted by *M* }

There is no concept of time– the whole string is accepted "instantaneously"

Deterministic finite automaton



Example:

q_1 := 1

$\mathcal{F}$:= {1}

L(*M*) = {$\in$, ab, aaa, abab, abaaa, aaaab, … }

= ((ab)* (aaa)*)*

Questions in formal language theory:

Is there a finite automaton that accepts a given language?

Do two given automata accept the same language?

What is the smallest automaton that accepts a given language? etc.

Definition: Given • initial state $q_1 \in \mathcal{Q}$

• set of final states $\mathcal{F} \subset \mathcal{Q}$

M ***accepts*** a string $s \in \Sigma^*$ with length $n := |\, s \,|$ if

there exists a sequence of states $q \in \mathcal{Q}^*$ with length $|\, q \,| = n+1$ (***execution***) such that

1. $q[1] = q_1$ (starts at initial state)
2. $q[i+1] = \Phi(q[i], s[i])$, $i \in \{1,2,\ldots,n\}$ (follows arrows with correct label)
3. $q[n+1] = \mathcal{F}$ (ends in set of final states)

Definition: ***language*** accepted by automaton *M*

L(*M*) := { set of all strings accepted by *M* }

Differential equation

ordinary differential equation with input Σ

$$\begin{cases} \mathbb{R}^n & \equiv \text{state space} \\ \mathbb{R}^m & \equiv \text{input space} \\ f : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n & \equiv \text{vector field} \end{cases}$$

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

Definition: Given an input signal $u : [0,\infty) \to \mathbb{R}^m$

A signal $x : [0,\infty) \to \mathbb{R}^n$ is a ***solution*** to Σ (in the sense of Caratheodory) if

1. x is piecewise differentiable
2. $x(t) = x_0 + \int_0^t f\big(x(\tau), u(\tau)\big) d\tau \qquad \forall t \geq 0$

If x is a solution then

$$\frac{\mathrm{d}x}{\mathrm{d}t}(t) = f\big(x(t), u(t)\big)$$

at any time *t* for which the derivative exists

Differential equation (no inputs)

ordinary differential equation without input Σ

$$\begin{cases} \mathbb{R}^n & \equiv \text{state space} \\ f : \mathbb{R}^n \to \mathbb{R}^n & \equiv \text{vector field} \end{cases}$$

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

Definition:

A signal $x : [0,\infty) \to \mathbb{R}^n$ is a ***solution*** to Σ (in the sense of Caratheodory) if

1. x is piecewise differentiable
2. $x(t) = x_0 + \int_0^t f\big(x(\tau)\big) d\tau \qquad \forall t \geq 0$

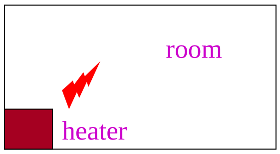
If x is a solution then

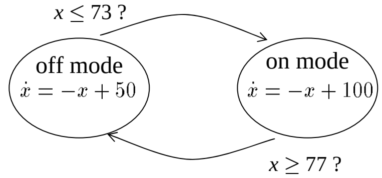
$$\frac{\mathrm{d}x}{\mathrm{d}t}(t) = f\big(x(t)\big)$$

at any time *t* for which the derivative exists

Hybrid Automaton
(Example #2: Thermostat)

$x \equiv$ mean temperature





$\mathcal{Q}$ $\equiv$ set of discrete states
$\mathbb{R}^n$ $\equiv$ continuous state-space
$f : \mathcal{Q} \times \mathbb{R}^n \to \mathbb{R}^n \equiv$ vector field
$\varphi : \mathcal{Q} \times \mathbb{R}^n \to \mathcal{Q} \equiv$ discrete transition

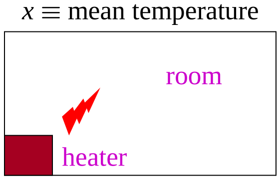
Example: $\mathcal{Q} := \{$ off, on $\}$ $n := 1$

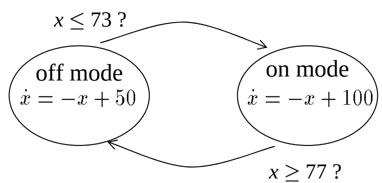
$$f(q,x) := \begin{cases} -x + 50 & q = \text{off} \\ -x + 100 & q = \text{on} \end{cases} \qquad \varphi(q,x) := \begin{cases} \text{on}, & q = \text{off}, x \leq 73 \\ \text{off} & q = \text{off}, x > 73 \\ \text{off}, & q = \text{on}, x \geq 77 \\ \text{on}, & q = \text{on}, x < 77 \end{cases}$$

note "closed" inequalities associated with jump and "open" inequalities with flow

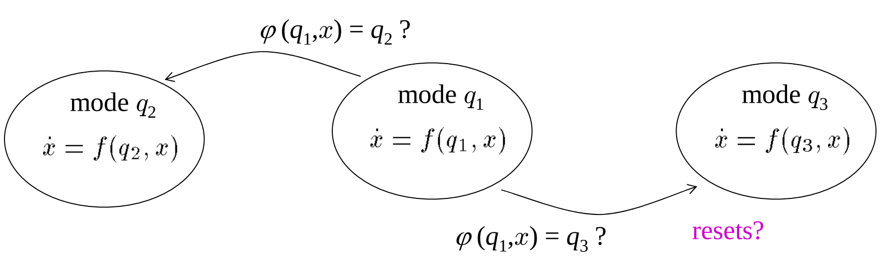
Hybrid Automaton
(Example #2: Thermostat)

$x \equiv$ mean temperature



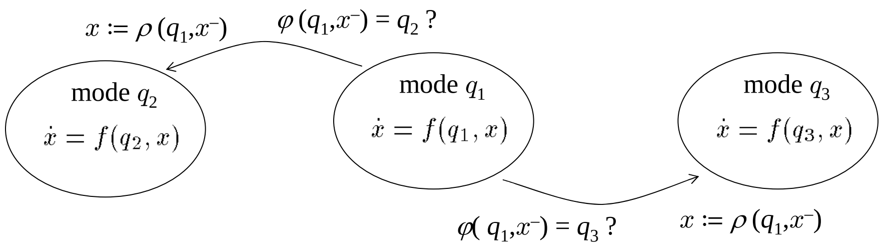


$\mathcal{Q}$ $\equiv$ set of discrete states
$\mathbb{R}^n$ $\equiv$ continuous state-space
$f : \mathcal{Q} \times \mathbb{R}^n \to \mathbb{R}^n \equiv$ vector field
$\varphi : \mathcal{Q} \times \mathbb{R}^n \to \mathcal{Q} \equiv$ discrete transition



Hybrid Automaton

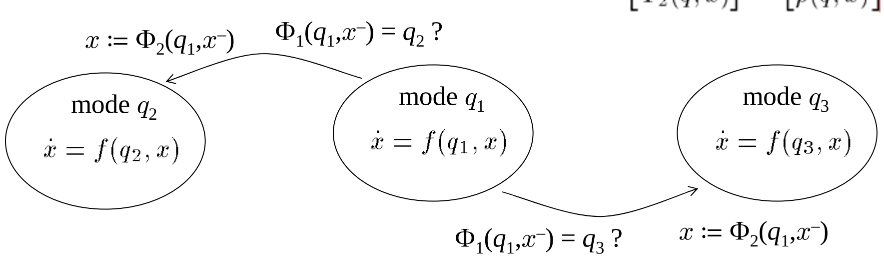
$\mathcal{Q}$ $\equiv$ set of discrete states
$\mathbb{R}^n$ $\equiv$ continuous state-space
$f : \mathcal{Q} \times \mathbb{R}^n \to \mathbb{R}^n$ $\equiv$ vector field
$\varphi : \mathcal{Q} \times \mathbb{R}^n \to \mathcal{Q}$ $\equiv$ discrete transition
$\rho : \mathcal{Q} \times \mathbb{R}^n \to \mathbb{R}^n$ $\equiv$ reset map



Hybrid Automaton

$\mathcal{Q}$ $\equiv$ set of discrete states
$\mathbb{R}^n$ $\equiv$ continuous state-space
$f : \mathcal{Q} \times \mathbb{R}^n \to \mathbb{R}^n$ $\equiv$ vector field
$\Phi : \mathcal{Q} \times \mathbb{R}^n \to \mathcal{Q} \times \mathbb{R}^n$ $\equiv$ discrete transition (& reset map)

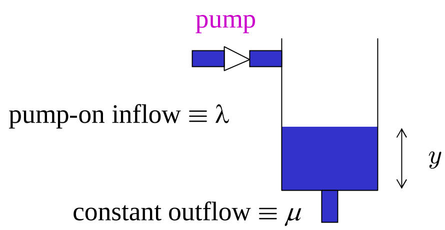
$$\Phi(q,x) = \begin{bmatrix} \Phi_1(q,x) \\ \Phi_2(q,x) \end{bmatrix} = \begin{bmatrix} \varphi(q,x) \\ \rho(q,x) \end{bmatrix}$$



Compact representation of a hybrid automaton

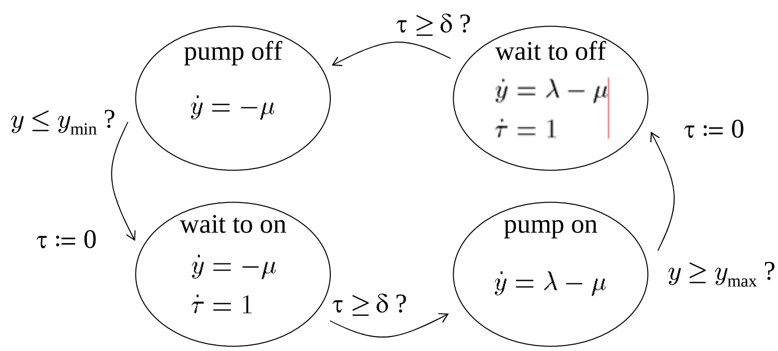
$$\dot{x} = f(q,x) \qquad (q,x) = \Phi(q^-, x^-) \qquad q \in \mathcal{Q},\ x \in \mathbb{R}^n$$

Example #5: Multiple-tank system

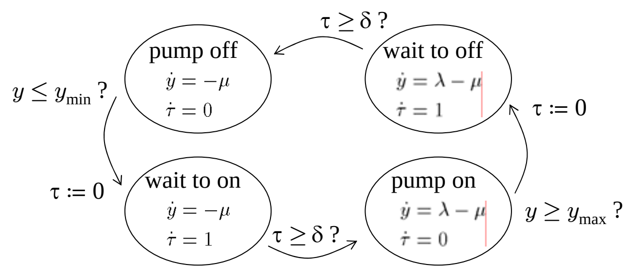


goal ≡ prevent the tank from emptying or filling up

δ ≡ delay between command is sent to pump and the time it is executed



Example #5: Multiple-tank system



$\mathcal{Q}$:= { off, won, on, woff }

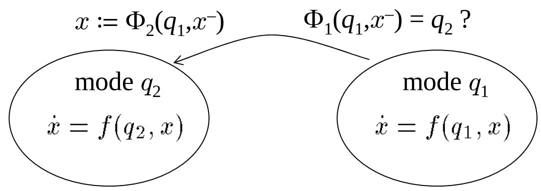
$\mathbb{R}^2$ ≡ continuous state-space

$$f(q,x) := \begin{cases} \begin{bmatrix} -\mu \\ 0 \end{bmatrix} & q = \text{off} \\ \begin{bmatrix} -\mu \\ 1 \end{bmatrix} & q = \text{won} \\ \begin{bmatrix} \lambda-\mu \\ 0 \end{bmatrix} & q = \text{on} \\ \begin{bmatrix} \lambda-\mu \\ 1 \end{bmatrix} & q = \text{woff} \end{cases}$$

$$\varphi(q,x) := \begin{cases} \text{off} & q = \text{woff}, \tau \geq \delta \\ \text{off} & q = \text{off}, y > y_{\min} \\ \text{won} & q = \text{off}, y \leq y_{\min} \\ \text{won} & q = \text{won}, \tau < \delta \\ \vdots & \end{cases} \qquad \rho(q,x) := \begin{cases} x & q = \text{woff}, \tau \geq \delta \\ x & q = \text{off}, y > y_{\min} \\ \begin{bmatrix} y \\ 0 \end{bmatrix} & q = \text{off}, y \leq y_{\min} \\ x & q = \text{won}, \tau < \delta \\ \vdots & \end{cases}$$

Solution to a hybrid automaton

$$\dot{x} = f(q, x) \qquad (q, x) = \Phi(q^-, x^-) \qquad q \in \mathcal{Q},\ x \in \mathbb{R}^n$$



Definition: A ***solution*** to the hybrid automaton is a pair of right-continuous signals

$$x : [0,\infty) \to \mathbb{R}^n \qquad q : [0,\infty) \to \mathcal{Q}$$

such that

1. x is piecewise differentiable & q is piecewise constant
2. on any interval (t_1,t_2) on which q is constant and x continuous

continuous evolution

$$x(t) = x(t_1) + \int_{t_1}^{t} f\big(q(t_1), x(\tau)\big)d\tau \qquad \forall t \in [t_1, t_2)$$

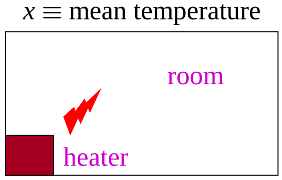
3. $\big(q(t), x(t)\big) = \Phi\big(q^-(t), x^-(t)\big) \quad \forall t \geq 0$

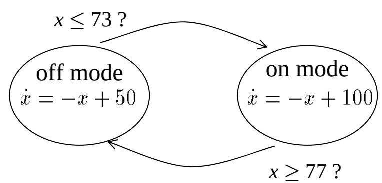
discrete transitions

Hybrid Automaton

(Example #2: Thermostat)

$x \equiv$ mean temperature

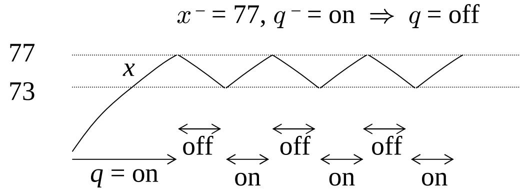




$$f(q,x) := \begin{cases} -x + 50 & q = \text{off} \\ -x + 100 & q = \text{on} \end{cases}$$

$$\varphi(q,x) := \begin{cases} \text{on}, & q = \text{off}, x \leq 73 \\ \text{off} & q = \text{off}, x > 73 \\ \text{off}, & q = \text{on}, x \geq 77 \\ \text{on}, & q = \text{on}, x < 77 \end{cases}$$

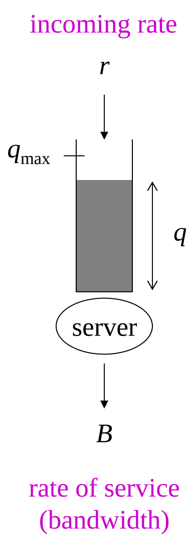
$x^- = 77,\ q^- = \text{on} \ \Rightarrow\ q = \text{off}$



no transition would occur if the "jump branch" had a strict inequality $x > 77$

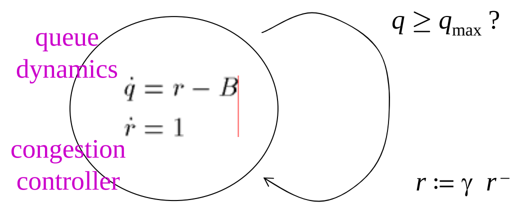
note "closed" inequalities associated with jumps and "open" inequalities with flows

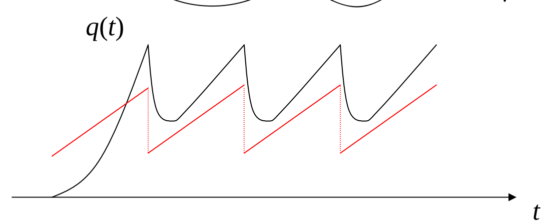
Example #7: Server system with congestion control



Additive increase/multiplicative decrease congestion control (AIMD):

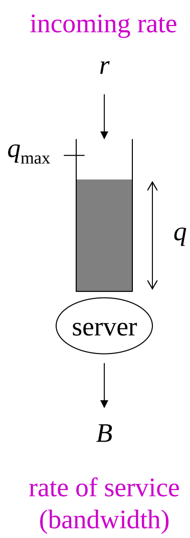
- while $q < q_{max}$ increase r linearly
- when q reaches q_{max} instantaneously multiply r by $\gamma \in (0,1)$

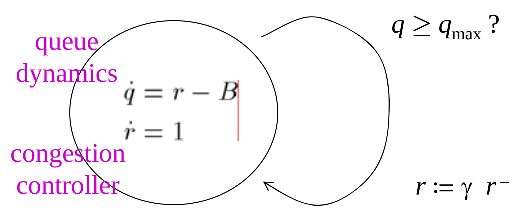


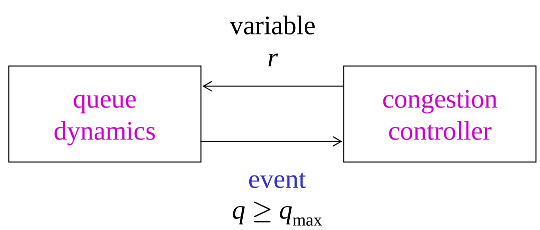


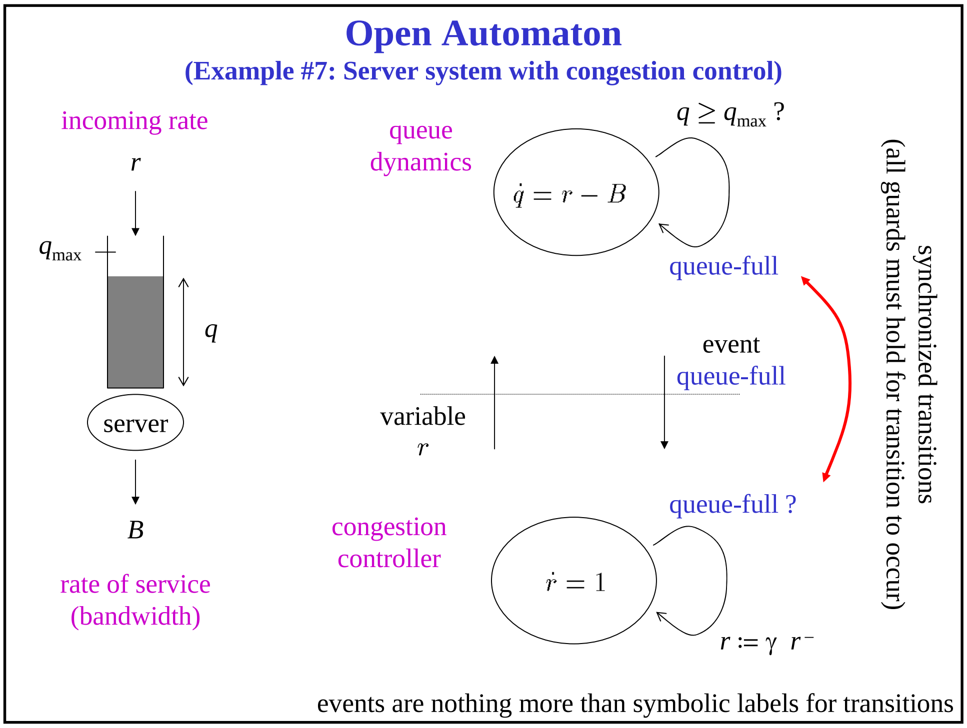
Open Automaton

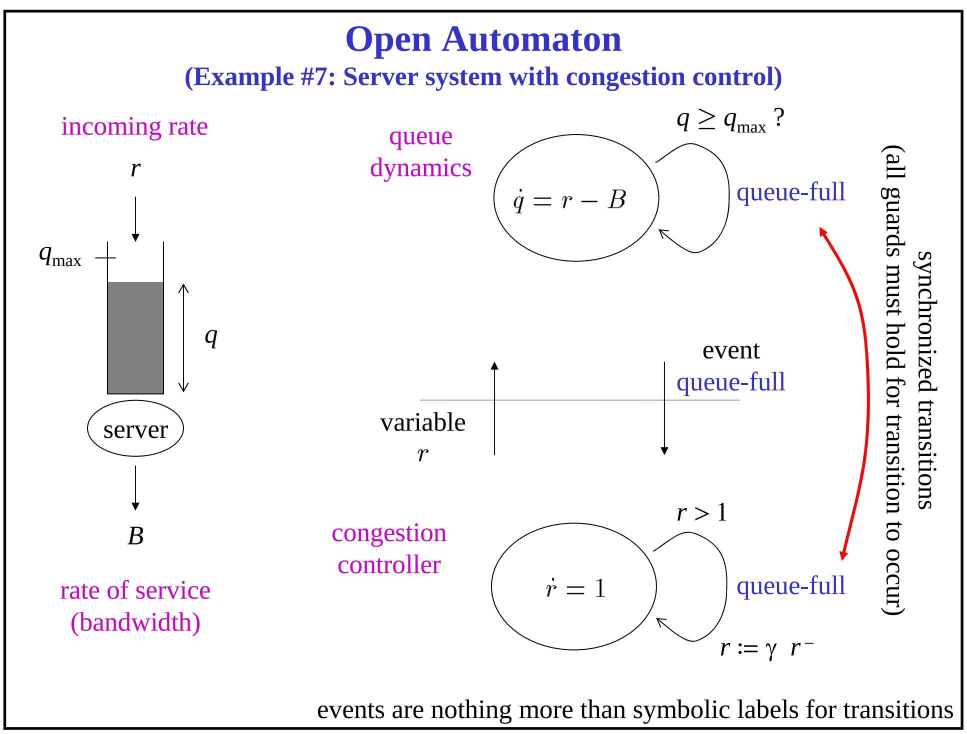
(Example #7: Server system with congestion control)





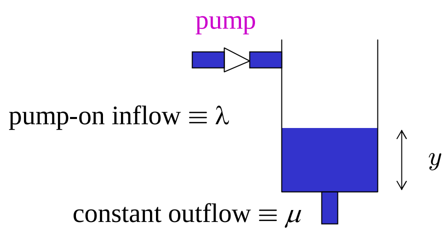



Open Automaton
(Example #7: Server system with congestion control)
incoming rate
r
qmax
q
server
B
rate of service
(bandwidth)
queue
dynamics
q̇ = r − B
q ≥ qmax ?
queue-full
event
queue-full
variable
r
congestion
controller
ṙ = 1
queue-full ?
r := γ r⁻
synchronized transitions
(all guards must hold for transition to occur)
events are nothing more than symbolic labels for transitions


Open Automaton
(Example #7: Server system with congestion control)
incoming rate
r
qmax
q
server
B
rate of service
(bandwidth)
queue
dynamics
q̇ = r − B
q ≥ qmax ?
queue-full
event
queue-full
variable
r
congestion
controller
ṙ = 1
r > 1
queue-full
r := γ r⁻
synchronized transitions
(all guards must hold for transition to occur)
events are nothing more than symbolic labels for transitions

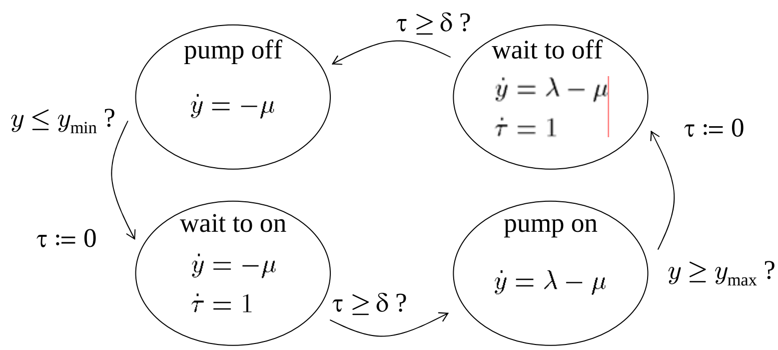
Open Automaton

(Example #5: Multiple-tank system)



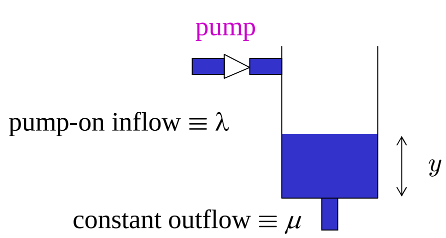
goal ≡ prevent the tank from emptying or filling up

δ ≡ delay between command is sent to pump and the time it is executed



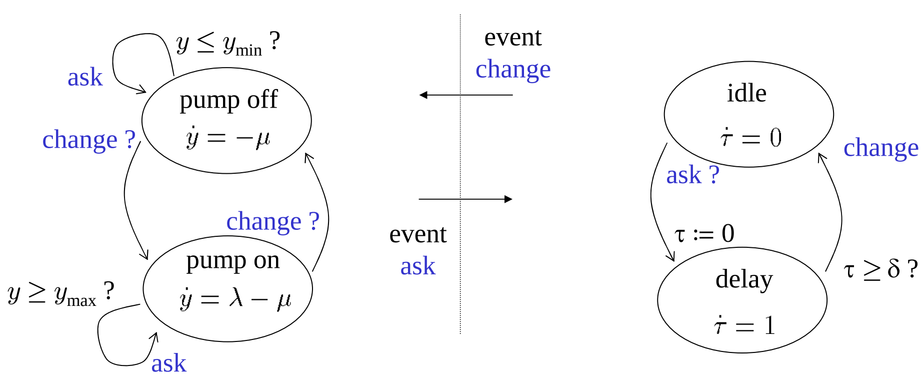
Open Automaton

(Example #5: Multiple-tank system)



goal ≡ prevent the tank from emptying or filling up

δ ≡ delay between command is sent to pump and the time it is executed



Deterministic finite automaton

automata M:

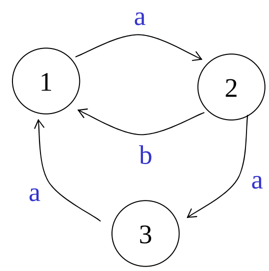
- $\mathcal{Q} := \{q_1, q_2, \ldots, q_n\}$ ≡ finite set of states
- $\Sigma := \{a, b, c, \ldots\}$ ≡ finite set of input symbols (alphabet)
- $\Phi : \mathcal{Q} \times \Sigma \to \mathcal{Q}$ ≡ transition function

Example:

$q \in \mathcal{Q}$	$s \in \Sigma$	$\Phi(q,s)$
1	a	2
1	b	⭮
2	a	3
2	b	1
3	a	1
3	b	⭮
⭮	a/b	⭮

blocking state

Graph representation:



- one node per state (except for blocking state ⭮)
- one directed edge (arrow) from q to $\Phi(q, s)$ with label s for each pair (q, s) for which $\Phi(q, s) \neq$ ⭮

Nondeterministic finite automaton

automata M:

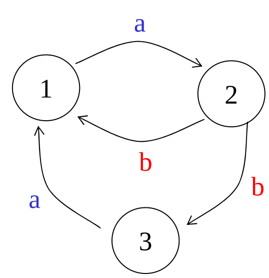
- $\mathcal{Q} := \{q_1, q_2, \ldots, q_n\}$ ≡ finite set of states
- $\Sigma := \{a, b, c, \ldots\}$ ≡ finite set of input symbols (alphabet)
- $\Phi : \mathcal{Q} \times \Sigma \to 2^{\mathcal{Q}}$ ≡ transition set-valued function

Example:

$q \in \mathcal{Q}$	$s \in \Sigma$	$\Phi(q,s)$
1	a	{2}
1	b	{⭮}
2	a	{⭮}
2	b	{1,3}
3	a	{1}
3	b	{⭮}
⭮	a/b	{⭮}

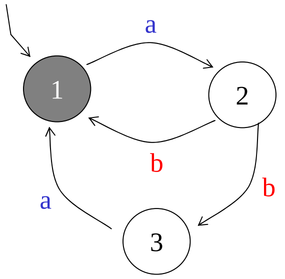
blocking state

Graph representation:



Notation: Given a set $\mathcal{A}$,
$2^{\mathcal{A}}$ ≡ ***power-set*** of $\mathcal{A}$, i.e., the set of all subsets of $\mathcal{A}$
e.g., $\mathcal{A} = \{1,2\} \Rightarrow 2^{\mathcal{A}} = \{\in, \{1\}, \{2\}, \{1,2\}\}$
When $\mathcal{A}$ has $n < \infty$ elements then $2^{\mathcal{A}}$ has 2^n elements

Nondeterministic finite automaton



Example:

q_1 := 1

$\mathcal{F}$:= {1}

L(M) = { $\in$, ab, aba, abab, ababa, abaab, … }

= ((ab)* (aba)*)*

Definition: Given
- initial state $q_1 \in \mathcal{Q}$
- set of final states $\mathcal{F} \subset \mathcal{Q}$

M ***accepts*** a string $s \in \Sigma^*$ with length $n := |\, s \,|$ if there exists a sequence of states $q \in \mathcal{Q}^*$ with length $|\, q \,| = n+1$ (***execution***) such that

1. $q[1] = q_1$ (starts at initial state)
2. $q[i+1] \in \Phi(q[i], s[i])$, $i \in \{1,2,\ldots,n\}$ (follows arrows with correct label)
3. $q[n+1] \in \mathcal{F}$ (ends in set of final states)

Definition: ***language*** accepted by automaton M

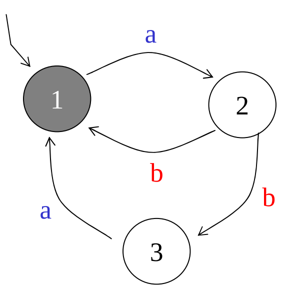
L(M) := { set of all strings accepted by M }

Determinization

From formal language theory:

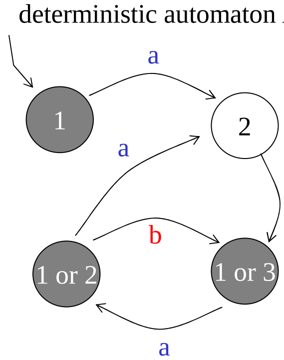
For every nondeterministic finite automaton there is a deterministic one that accepts the same language (but generally the deterministic one needs more states)

nondeterministic automaton M



- from 1 only accepts a and goes to 2
- from 2 only accepts b and can go to either 1 or 3
- from 1 or 3 only accepts a and goes to 2 or 1 resp.
- from 1 (or 2) can accepts a and go to 2 from (1 or) 2 can accept b and go to 1 or 3

deterministic automaton N

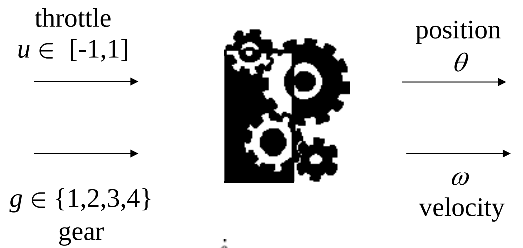


Same language:

L(M) = L(N) = ((ab)* (aba)*)*

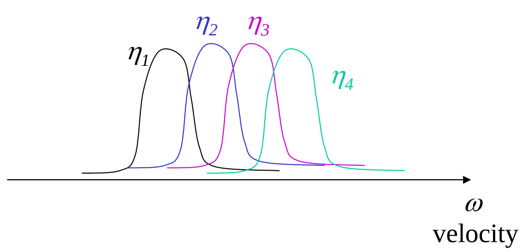
M provides more compact representation

Example #3: Transmission



$$\dot{\theta} = \omega$$
$$\dot{\omega} = \eta_g(\omega)u$$

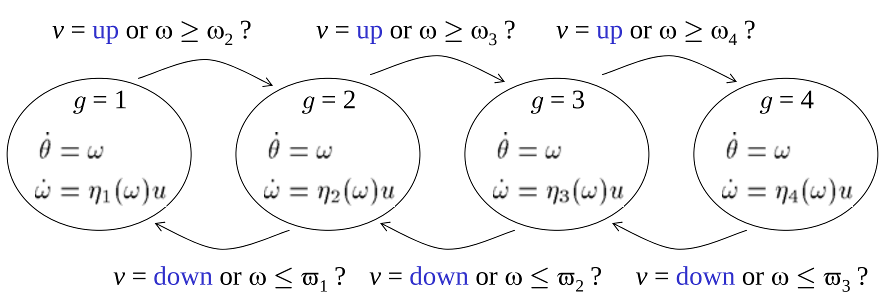
$\eta_k \equiv$ efficiency of the k^{th} gear

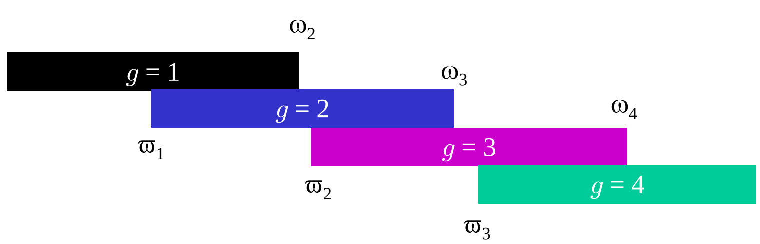


[Hedlund, Rantzer 1999]

Example #3: Semi-automatic transmission

$v(t) \in \{$ up, down, keep $\} \equiv$ drivers input (discrete)

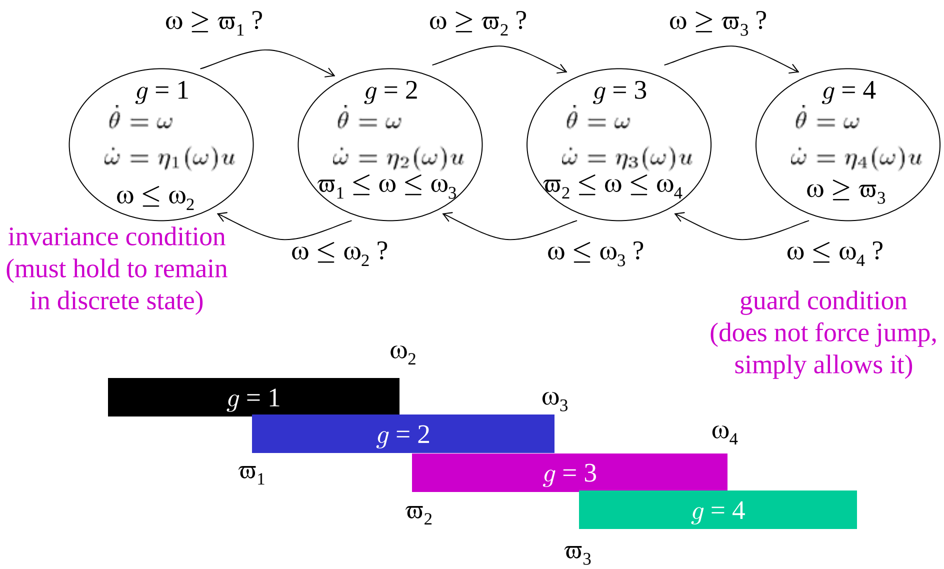




Nondeterministic Hybrid Automaton

(Example #3: Semi-automatic transmission)

Suppose we want to consider *all possible driver inputs*:



Nondeterministic Hybrid Automaton

$\mathcal{Q}$ ≡ set of discrete states

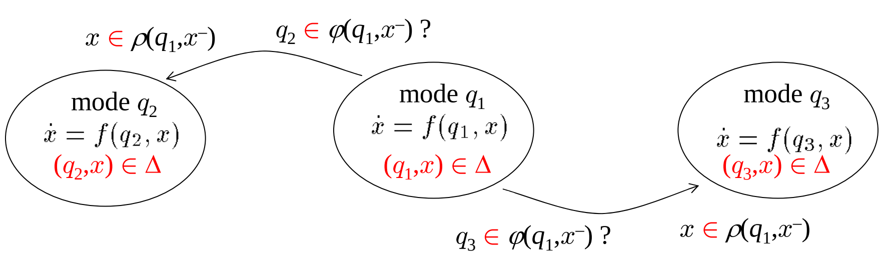
$\mathbb{R}^n$ ≡ continuous state-space

$f : \mathcal{Q} \times \mathbb{R}^n \to \mathbb{R}^n$ ≡ vector field

$\varphi : \mathcal{Q} \times \mathbb{R}^n \to 2^{\mathcal{Q}}$ ≡ set-valued discrete transition

$\rho : \mathcal{Q} \times \mathbb{R}^n \to 2^{\mathbb{R}^n}$ ≡ set-valued reset map

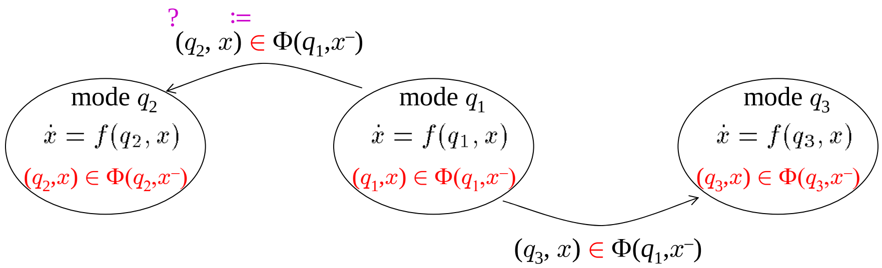
$\Delta \subset \mathcal{Q} \times \mathbb{R}^n$ ≡ domain or invariant set



Nondeterministic Hybrid Automaton

$\mathcal{Q}$	$\equiv$ set of discrete states
$\mathbb{R}^n$	$\equiv$ continuous state-space
$f : \mathcal{Q} \times \mathbb{R}^n \to \mathbb{R}^n$	$\equiv$ vector field
$\Phi: \mathcal{Q} \times \mathbb{R}^n \to 2^{\mathcal{Q}\times\mathbb{R}^n}$	$\equiv$ set-valued discrete transition (& reset & domain)

$$\Phi(q,x) = \big(\varphi(q,x) \times \rho(q,x)\big) \cap \Delta$$

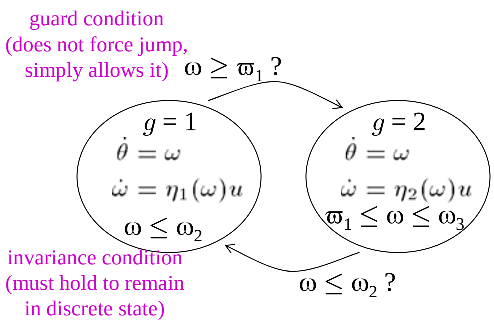


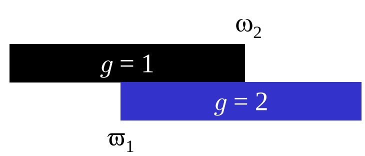
Compact representation of a nondeterministic hybrid automaton

$$\dot{x} = f(q,x) \qquad (q,x) \in \Phi(q^-, x^-) \qquad q \in \mathcal{Q},\ x \in \mathbb{R}^n$$

Nondeterministic Hybrid Automaton

(Example #3: Semi-automatic transmission)





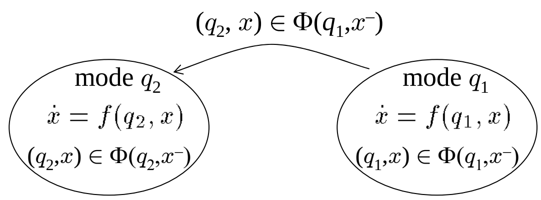
$\mathcal{Q} := \{ 1, 2 \}$

$\mathbb{R}^2 \equiv$ continuous state-space

$$f(q,x) = \begin{bmatrix} \omega \\ \eta_q(\omega)u \end{bmatrix} \qquad \Phi(q,x) = \begin{cases} \{(1,x)\} & q = 1, \omega < \varpi_1 \\ \{(1,x),(2,x)\} & q = 1, \omega \in [\varpi_1, \omega_2] \\ \{(2,x)\} & q = 1, \omega > \omega_2 \\ \{(2,x)\} & q = 2, \omega > \omega_2 \\ \{(1,x),(2,x)\} & q = 2, \omega \in [\varpi_1, \omega_2] \\ \{(1,x)\} & q = 2, \omega < \varpi_1 \end{cases}$$

Solution to a nondeterministic hybrid automaton

$$\dot{x} = f(q, x) \qquad (q, x) \in \Phi(q, x^-) \qquad q \in \mathcal{Q},\ x \in \mathbb{R}^n$$



Definition: A ***solution*** to the hybrid automaton is a pair of right-continuous signals

$$x : [0,\infty) \to \mathbb{R}^n \qquad q : [0,\infty) \to \mathcal{Q}$$

such that

1. x is piecewise differentiable & q is piecewise constant
2. on any interval (t_1,t_2) on which q is constant and x continuous

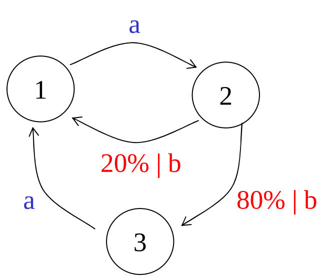
continuous evolution

$$x(t) = x(t_1) + \int_{t_1}^{t} f\big(q(t_1), x(\tau)\big)d\tau \qquad \forall t \in [t_1, t_2)$$

3. $\big(q(t), x(t)\big) \in \Phi\big(q^-(t), x^-(t)\big) \quad \forall t \geq 0$

discrete transition & resets & domain

Stochastic finite automaton: controlled Markov chain



controlled Markov chain M

$\mathcal{Q} := \{q_1, q_2, \ldots, q_n\}$ $\equiv$ finite set of states

$\Sigma := \{\text{a, b, c,} \ldots\}$ $\equiv$ finite set of input symbols

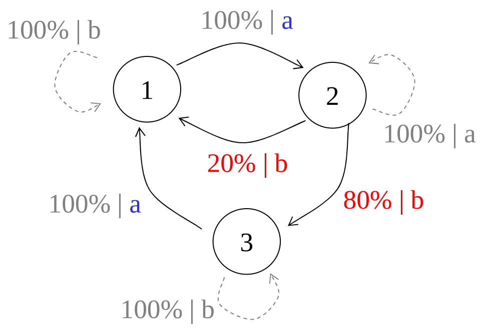
$\Phi : \mathcal{Q} \times \mathcal{Q} \times \Sigma \to [0,1]$ $\equiv$ transition probability function

$\Phi(q_1, q_2, s) \equiv$ probability of transitioning to state q_2, when in state q_1 and symbol s is selected

By convention, typically

- edges drawn without probabilities correspond to transitions that occur with probability 1
- self-loops may be omitted

Stochastic finite automaton: controlled Markov chain



By convention, typically

- edges drawn without probabilities correspond to transitions that occur with probability 1
- self loops may be omitted

$\mathcal{Q} := \{1, 2, 3\}$ $\Sigma := \{a, b\}$

$\Phi(q_1, q_2, s) \equiv$ probability of transitioning to state q_2, when in state q_1 and symbol s is selected

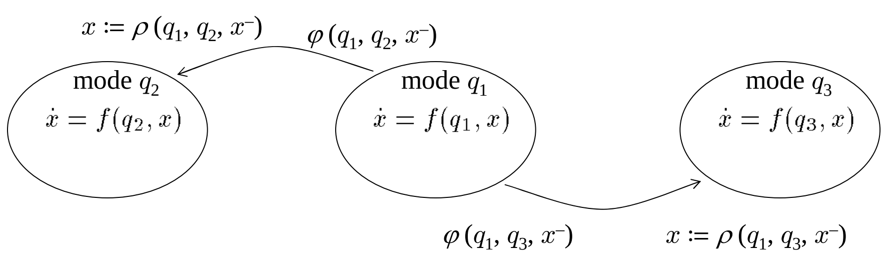
$$\sum_{q_2 \in \mathcal{Q}} \Phi(q_1, q_2, s) = 1 \qquad \forall q_1 \in Q,\ s \in \Sigma$$

Stochastic Hybrid Automaton

$\mathcal{Q}$ $\equiv$ set of discrete states
$\mathbb{R}^n$ $\equiv$ continuous state-space
$f : \mathcal{Q} \times \mathbb{R}^n \to \mathbb{R}^n$ $\equiv$ vector field
$\varphi : \mathcal{Q} \times \mathcal{Q} \times \mathbb{R}^n \to [0,\infty]$ $\equiv$ discrete transition probability
$\rho : \mathcal{Q} \times \mathcal{Q} \times \mathbb{R}^n \to \mathbb{R}^n$ $\equiv$ reset map (deterministic)

$$\varphi(q_1, q_2, x) = \lim_{dt \downarrow 0} \frac{\mathrm{P}\left(q(t+dt) = q_2 \mid q^-(t) = q_1, x^-(t) = x\right)}{dt}$$

(Poisson-like model)

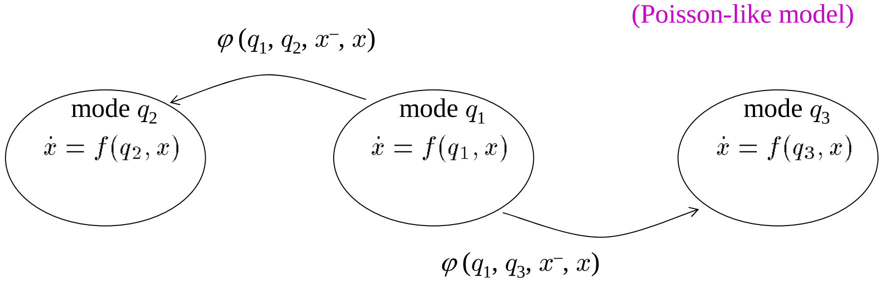


Stochastic Hybrid Automaton

$\mathcal{Q}$ ≡ set of discrete states
$\mathbb{R}^n$ ≡ continuous state-space
$f : \mathcal{Q} \times \mathbb{R}^n \to \mathbb{R}^n$ ≡ vector field
$\Phi : \mathcal{Q} \times \mathcal{Q} \times \mathbb{R}^n \times \mathbb{R}^n \to [0,\infty]$ ≡ discrete transition probability & reset

$$\Phi(q_1, q_2, x_1, x_2) = \lim_{dt \downarrow 0} \frac{\mathrm{P}\left(q(t+dt) = q_2, x(t+dt) = x_2\right) \mid q^-(t) = q_1, x^-(t) = x_1)}{dt}$$

(Poisson-like model)



More as special topic later…

Next class…

1. **Trajectories of hybrid systems**:
   - Solution to a hybrid system
   - Execution of a hybrid system
2. Degeneracies
   - Finite escape time
   - Chattering
   - Zeno trajectories
   - Non-continuous dependency on initial conditions