# Hybrid Control and Switched Systems

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

João P. Hespanha

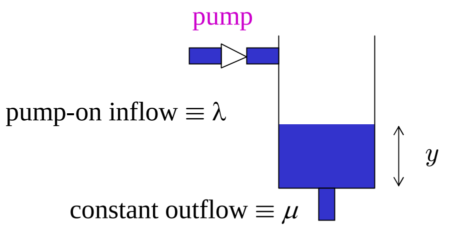
University of California
at Santa Barbara

---

# Summary

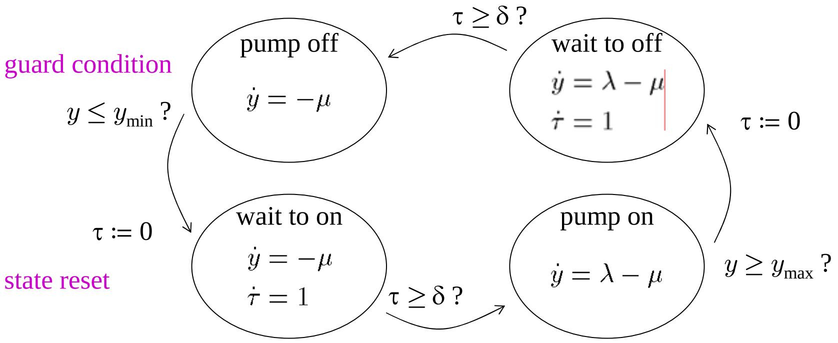
1. **Formal models for hybrid systems**:
    - Finite automata
    - Differential equations
    - Hybrid automata
    - Open hybrid automaton
2. Nondeterministic vs. stochastic systems
    - Non-deterministic hybrid automata
    - Stochastic hybrid automata

# 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
$M$

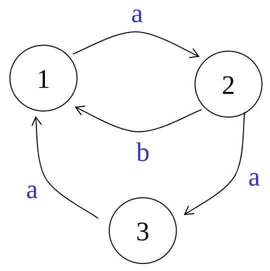
$\mathcal{Q} := \{q_1, q_2, \ldots, q_n\}$ ≡ finite set of states
$\Sigma := \{\text{a, b, c,} \ldots\}$ ≡ finite set of input symbols (alphabet)
$\Phi : \mathcal{Q} \times \Sigma \to \mathcal{Q}$ ≡ transition function

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 | ⭮ |

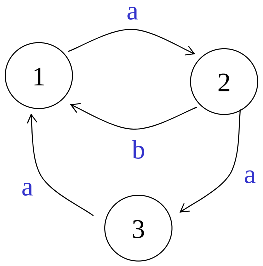
blocking state

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 ≡ finite sequence of symbols

$\in$ ≡ empty string

$\mathcal{A}^*$ ≡ 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)

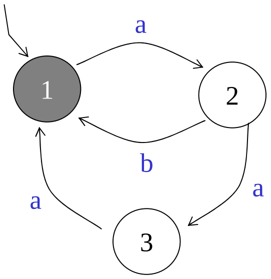
Definition: ***language*** accepted by automaton *M*

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 $\Sigma$

$\mathbb{R}^n$ ≡ state space
$\mathbb{R}^m$ ≡ input space
$f : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n$ ≡ vector field

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

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 $\Sigma$ (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 $\Sigma$

$\mathbb{R}^n$ ≡ state space
$f : \mathbb{R}^n \to \mathbb{R}^n$ ≡ vector field

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

Definition:

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

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

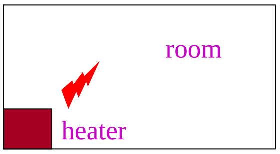
If $x$ is a solution then

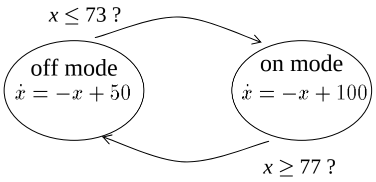
$$\frac{\mathrm{d}x}{\mathrm{d}t}(t) = f\big(x(t)\big)$$

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

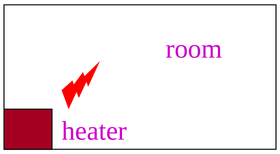
$$f(q,x) := \begin{cases} -x+50 & q = \text{off} \\ -x+100 & q = \text{on} \end{cases} \qquad \varphi(q,x) := \begin{cases} \text{on}, & q = \text{off}, x \le 73 \\ \text{off} & q = \text{off}, x > 73 \\ \text{off}, & q = \text{on}, x \ge 77 \\ \text{on}, & q = \text{on}, x < 77 \end{cases}$$

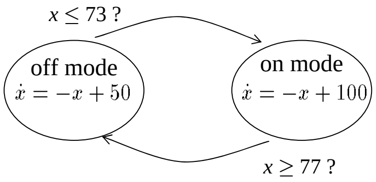
*note "closed" inequalities associated with jump and "open" inequalities with flow*

---

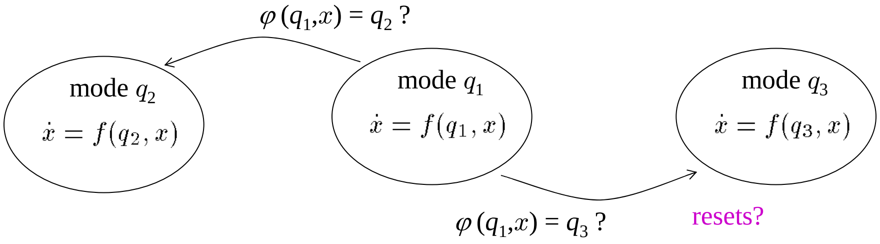
## Hybrid Automaton
### (Example #2: Thermostat)

$x \equiv$ mean temperature



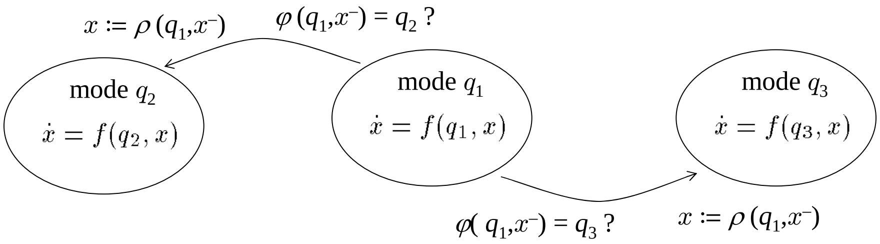


$\mathcal{Q}$ $\equiv$ set of discrete states
$\mathbb{R}^n$ $\equiv$ continuous state-space
$f : \mathcal{Q} \times \mathbb{R}^n \to \mathbb{R}^n \equiv$ vector field
$\varphi : \mathcal{Q} \times \mathbb{R}^n \to \mathcal{Q} \equiv$ discrete transition



# Hybrid Automaton

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



$x := \rho(q_1, x^-)$ $\varphi(q_1, x^-) = q_2$ ?

mode $q_2$ $\dot{x} = f(q_2, x)$

mode $q_1$ $\dot{x} = f(q_1, x)$

mode $q_3$ $\dot{x} = f(q_3, x)$

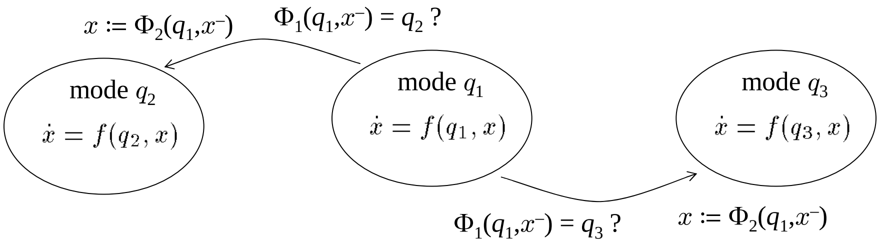
$\varphi(q_1, x^-) = q_3$ ? $x := \rho(q_1, x^-)$

---

# 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
$\Phi : \mathcal{Q} \times \mathbb{R}^n \to \mathcal{Q} \times \mathbb{R}^n$ ≡ discrete transition (& reset map)

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



$x := \Phi_2(q_1, x^-)$ $\Phi_1(q_1, x^-) = q_2$ ?

mode $q_2$ $\dot{x} = f(q_2, x)$

mode $q_1$ $\dot{x} = f(q_1, x)$

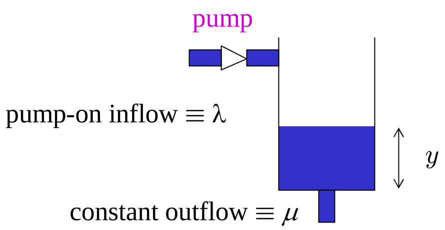
mode $q_3$ $\dot{x} = f(q_3, x)$

$\Phi_1(q_1, x^-) = q_3$ ? $x := \Phi_2(q_1, x^-)$

Compact representation of a hybrid automaton

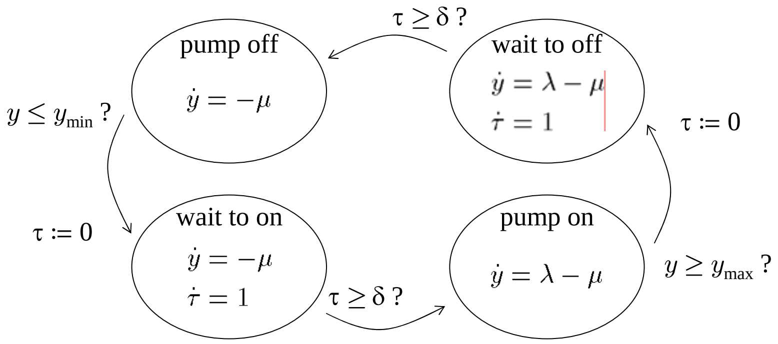
$$\dot{x} = f(q, x) \qquad (q, x) = \Phi(q^-, x^-) \qquad q \in \mathcal{Q},\ x \in \mathbb{R}^n$$

# Example #5: Multiple-tank system



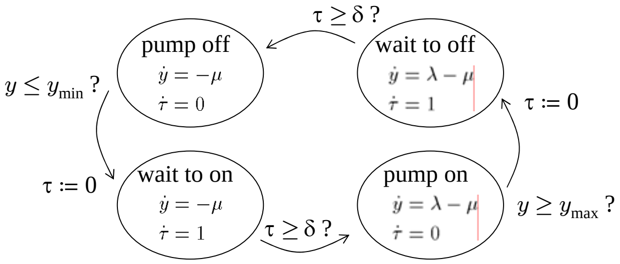
goal ≡ prevent the tank from emptying or filling up

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

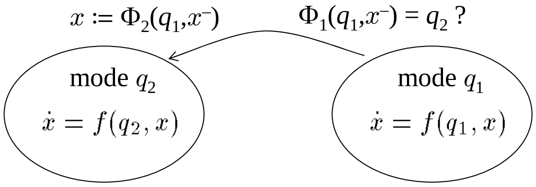
$$f(q,x) := \begin{cases} \begin{bmatrix} -\mu \\ 0 \end{bmatrix} & q = \text{off} \\ \begin{bmatrix} -\mu \\ 1 \end{bmatrix} & q = \text{won} \\ \begin{bmatrix} \lambda-\mu \\ 0 \end{bmatrix} & q = \text{on} \\ \begin{bmatrix} \lambda-\mu \\ 1 \end{bmatrix} & q = \text{woff} \end{cases}$$

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

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

# 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

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

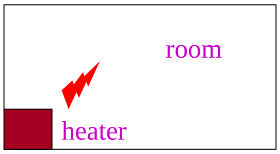
3. $(q(t),x(t)) = \Phi(q^-(t),x^-(t)) \quad \forall t \geq 0$

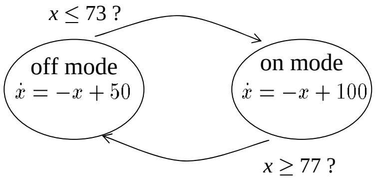
discrete transitions

# Hybrid Automaton

## (Example #2: Thermostat)

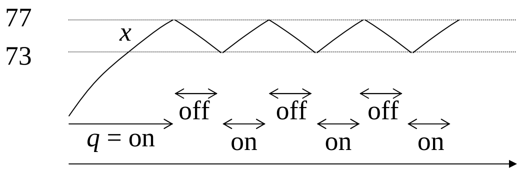
$x \equiv$ mean temperature





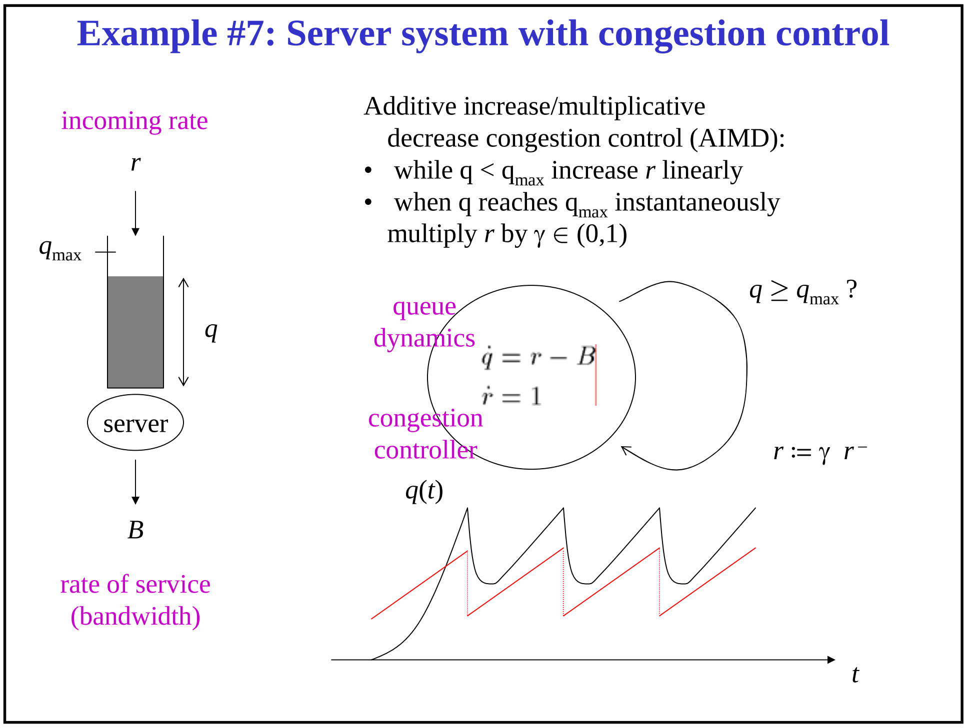
$$f(q,x) := \begin{cases} -x+50 & q = \text{off} \\ -x+100 & q = \text{on} \end{cases} \qquad \varphi(q,x) := \begin{cases} \text{on}, & q = \text{off}, x \leq 73 \\ \text{off} & q = \text{off}, x > 73 \\ \text{off}, & q = \text{on}, x \geq 77 \\ \text{on}, & q = \text{on}, x < 77 \end{cases}$$

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



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

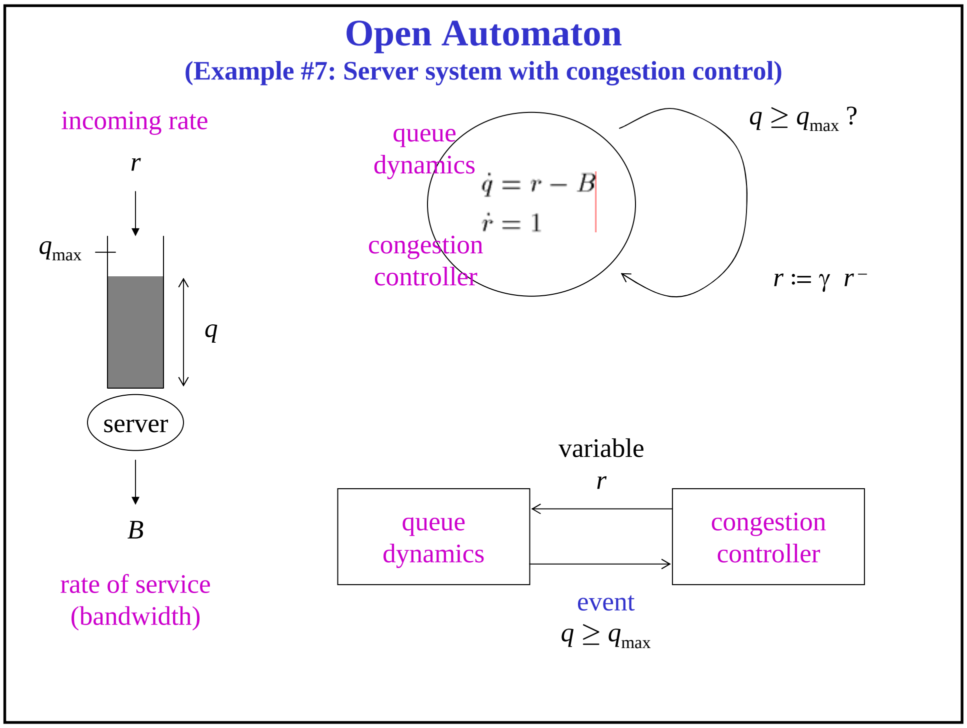
*note “closed” inequalities associated with jumps and “open” inequalities with flows*



## 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 $\gamma \in (0,1)$

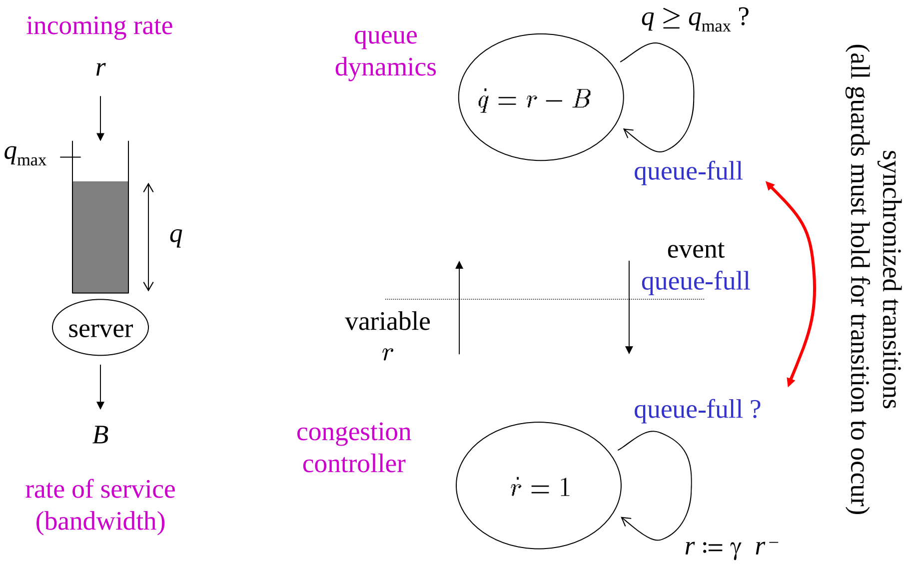


## Open Automaton

### (Example #7: Server system with congestion control)

# Open Automaton

(Example #7: Server system with congestion control)



incoming rate

$r$

$q_{\max}$

$q$

server

$B$

rate of service (bandwidth)

queue dynamics

$q \geq q_{\max}$ ?

$\dot{q} = r - B$

queue-full

event queue-full

variable $r$

congestion controller

queue-full ?

$\dot{r} = 1$

$r := \gamma\ r^{-}$

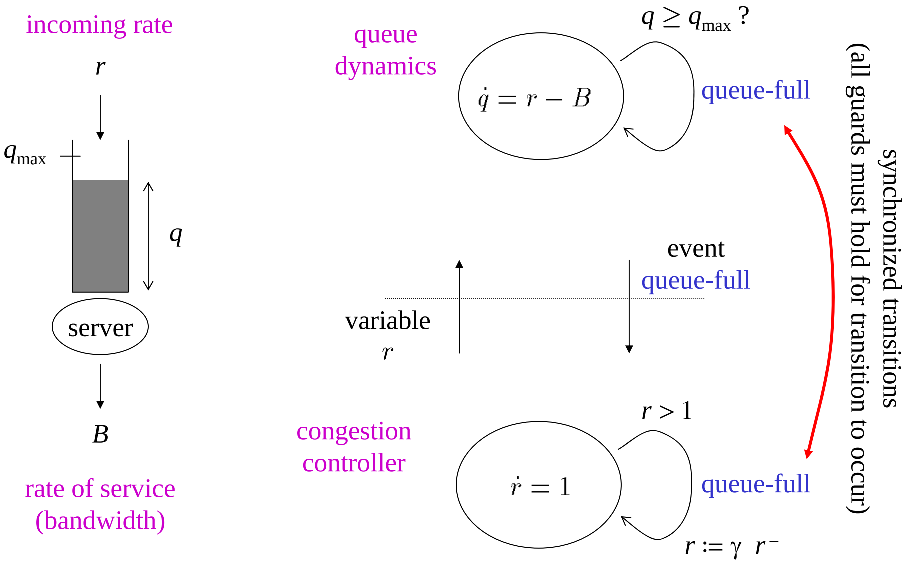
synchronized transitions (all guards must hold for transition to occur)

events are nothing more than symbolic labels for transitions

---

# Open Automaton

(Example #7: Server system with congestion control)



incoming rate

$r$

$q_{\max}$

$q$

server

$B$

rate of service (bandwidth)

queue dynamics

$q \geq q_{\max}$ ?

$\dot{q} = r - B$

queue-full

event queue-full

variable $r$

congestion controller

$r > 1$

$\dot{r} = 1$

queue-full

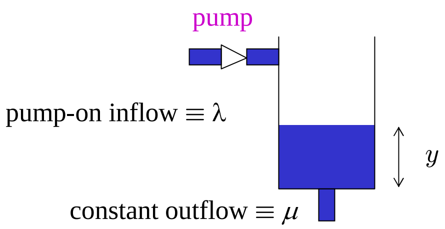
$r := \gamma\ r^{-}$

synchronized transitions (all guards must hold for transition to occur)

events are nothing more than symbolic labels for transitions

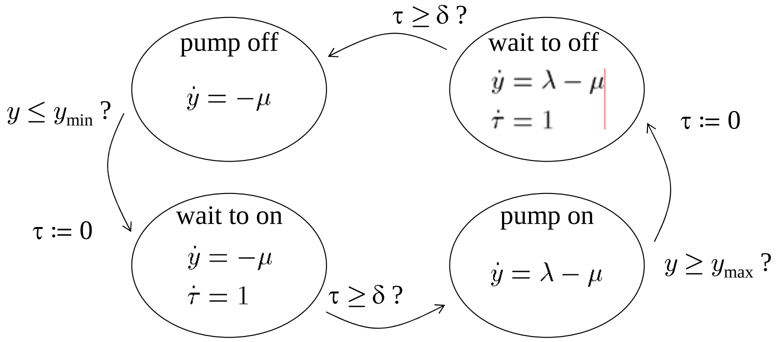
# Open Automaton

## (Example #5: Multiple-tank system)


pump
pump-on inflow ≡ λ
$y$
constant outflow ≡ $\mu$

goal ≡ prevent the tank from emptying or filling up

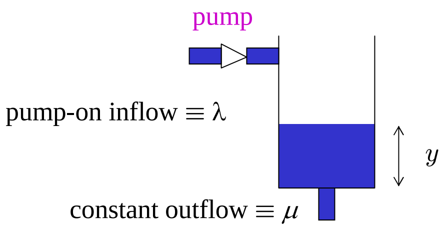
δ ≡ delay between command is sent to pump and the time it is executed


$\tau \geq \delta$ ?
pump off
$\dot{y} = -\mu$
wait to off
$\dot{y} = \lambda - \mu$
$\dot{\tau} = 1$
$y \leq y_{\text{min}}$ ?
$\tau := 0$
$\tau := 0$
wait to on
$\dot{y} = -\mu$
$\dot{\tau} = 1$
$\tau \geq \delta$ ?
pump on
$\dot{y} = \lambda - \mu$
$y \geq y_{\text{max}}$ ?

---

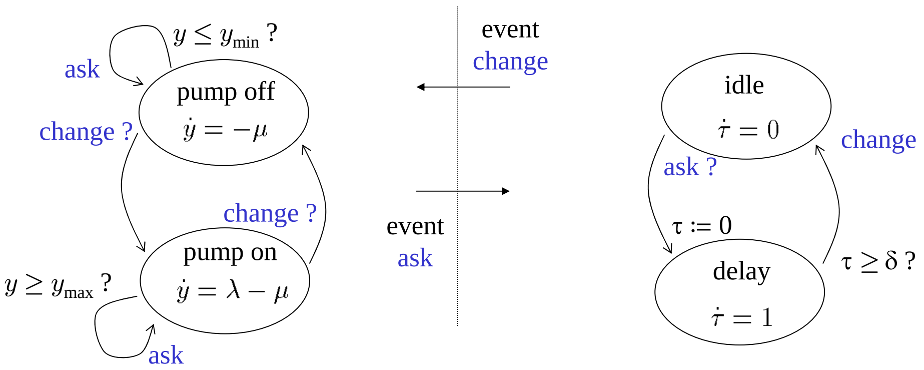
# Open Automaton

## (Example #5: Multiple-tank system)


pump
pump-on inflow ≡ λ
$y$
constant outflow ≡ $\mu$

goal ≡ prevent the tank from emptying or filling up

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


$y \leq y_{\text{min}}$ ?
ask
pump off
$\dot{y} = -\mu$
change ?
change ?
pump on
$\dot{y} = \lambda - \mu$
$y \geq y_{\text{max}}$ ?
ask
event
change
event
ask
idle
$\dot{\tau} = 0$
change
ask ?
$\tau := 0$
delay
$\dot{\tau} = 1$
$\tau \geq \delta$ ?

## Deterministic finite automaton

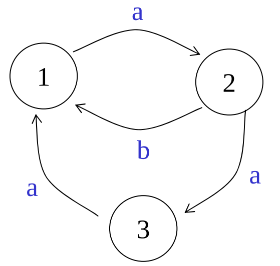
automata $M$
$$\begin{cases} \mathcal{Q} := \{q_1, q_2, \dots, q_n\} & \equiv \text{finite set of states} \\ \Sigma := \{\text{a, b, c,} \dots \} & \equiv \text{finite set of input symbols (alphabet)} \\ \Phi : \mathcal{Q} \times \Sigma \to \mathcal{Q} & \equiv \text{transition function} \end{cases}$$

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

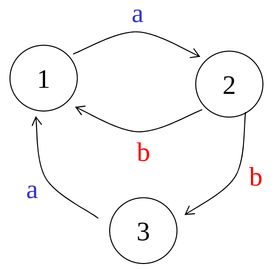
automata $M$
$$\begin{cases} \mathcal{Q} := \{q_1, q_2, \dots, q_n\} & \equiv \text{finite set of states} \\ \Sigma := \{\text{a, b, c,} \dots \} & \equiv \text{finite set of input symbols (alphabet)} \\ \Phi : \mathcal{Q} \times \Sigma \to 2^{\mathcal{Q}} & \equiv \text{transition set-valued function} \end{cases}$$

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 | {↻} |

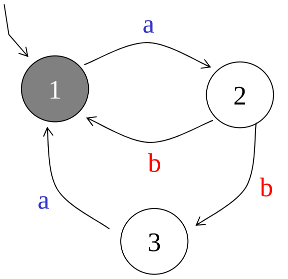
blocking
state

Graph representation:



**Notation:** Given a set $\mathcal{A}$,
$2^{\mathcal{A}} \equiv$ ***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:

$q_1 := 1$

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

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

$= ( (ab)^* (aba)^* )^*$

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 $M$ }

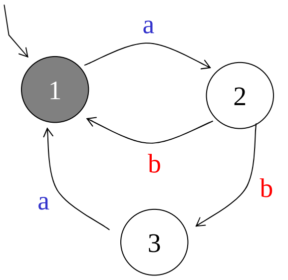
---

## Determinization

From formal language theory:

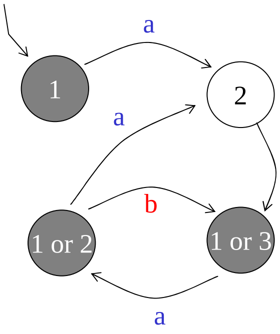
For every nondeterministic finite automaton there is a deterministic one that accepts the same language (but generally the deterministic one needs more states)

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 $N$

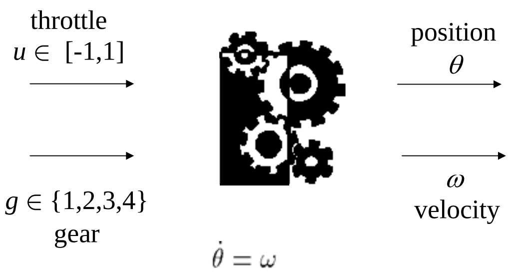


Same language:

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

$M$ provides more compact representation

# Example #3: Transmission



throttle
$u \in [-1,1]$

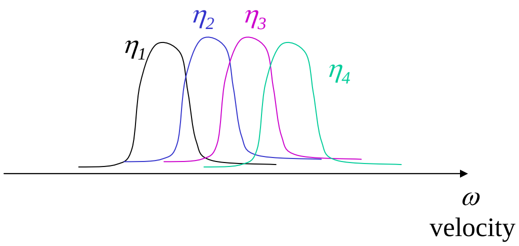
$g \in \{1,2,3,4\}$
gear

position
$\theta$

$\omega$
velocity

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

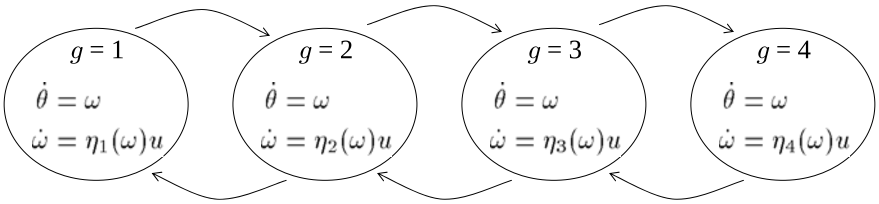
$\eta_k \equiv$ efficiency of the $k^{th}$ gear



[Hedlund, Rantzer 1999]

# Example #3: Semi-automatic transmission

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



$v$ = up or $\omega \geq \omega_2$ ?  $v$ = up or $\omega \geq \omega_3$ ?  $v$ = up or $\omega \geq \omega_4$ ?

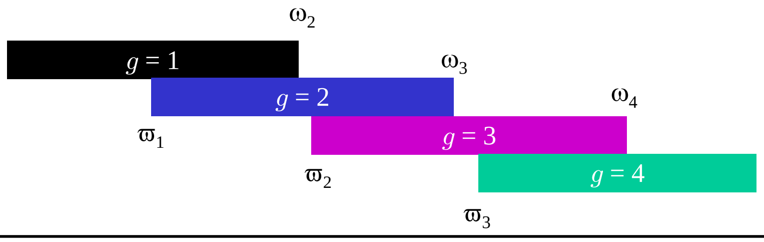
$g = 1$: $\dot{\theta} = \omega$, $\dot{\omega} = \eta_1(\omega)u$

$g = 2$: $\dot{\theta} = \omega$, $\dot{\omega} = \eta_2(\omega)u$

$g = 3$: $\dot{\theta} = \omega$, $\dot{\omega} = \eta_3(\omega)u$

$g = 4$: $\dot{\theta} = \omega$, $\dot{\omega} = \eta_4(\omega)u$

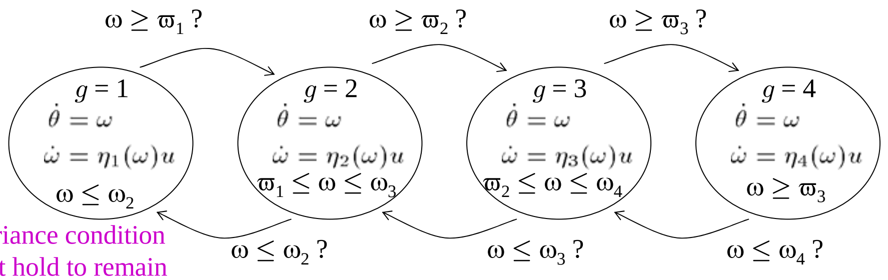
$v$ = down or $\omega \leq \varpi_1$ ?  $v$ = down or $\omega \leq \varpi_2$ ?  $v$ = down or $\omega \leq \varpi_3$ ?



# Nondeterministic Hybrid Automaton

### (Example #3: Semi-automatic transmission)

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



$\omega \geq \varpi_1$ ?

$\omega \geq \varpi_2$ ?

$\omega \geq \varpi_3$ ?

$g = 1$
$\dot{\theta} = \omega$
$\dot{\omega} = \eta_1(\omega)u$
$\omega \leq \omega_2$

$g = 2$
$\dot{\theta} = \omega$
$\dot{\omega} = \eta_2(\omega)u$
$\varpi_1 \leq \omega \leq \omega_3$

$g = 3$
$\dot{\theta} = \omega$
$\dot{\omega} = \eta_3(\omega)u$
$\varpi_2 \leq \omega \leq \omega_4$

$g = 4$
$\dot{\theta} = \omega$
$\dot{\omega} = \eta_4(\omega)u$
$\omega \geq \varpi_3$

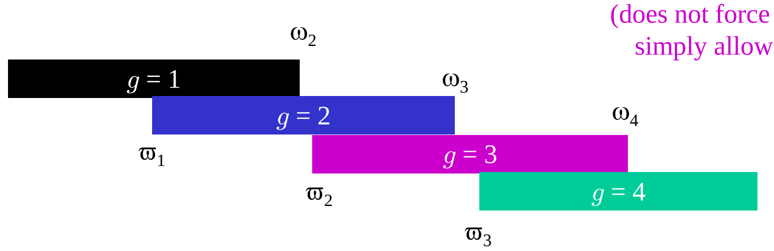
$\omega \leq \omega_2$ ?

$\omega \leq \omega_3$ ?

$\omega \leq \omega_4$ ?

invariance condition
(must hold to remain
in discrete state)

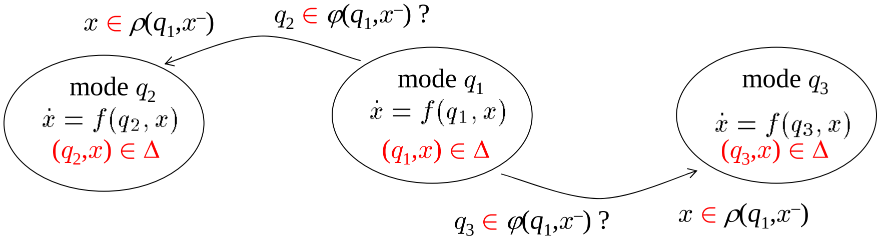
guard condition
(does not force jump,
simply allows it)



---

# 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 |



$x \in \rho(q_1, x^-)$

$q_2 \in \varphi(q_1, x^-)$ ?

mode $q_2$
$\dot{x} = f(q_2, x)$
$(q_2, x) \in \Delta$

mode $q_1$
$\dot{x} = f(q_1, x)$
$(q_1, x) \in \Delta$

mode $q_3$
$\dot{x} = f(q_3, x)$
$(q_3, x) \in \Delta$

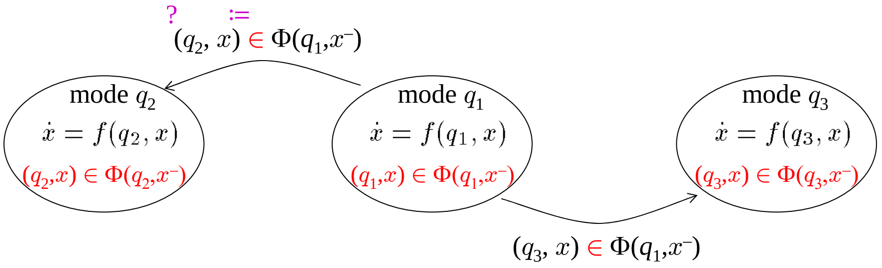
$q_3 \in \varphi(q_1, x^-)$ ?

$x \in \rho(q_1, x^-)$

# 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
$\Phi : \mathcal{Q} \times \mathbb{R}^n \to 2^{\mathcal{Q} \times \mathbb{R}^n}$ ≡ set-valued discrete transition (& reset & domain)

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



?   :=
$(q_2, x) \in \Phi(q_1, x^-)$

mode $q_2$
$\dot{x} = f(q_2, x)$
$(q_2, x) \in \Phi(q_2, x^-)$

mode $q_1$
$\dot{x} = f(q_1, x)$
$(q_1, x) \in \Phi(q_1, x^-)$

mode $q_3$
$\dot{x} = f(q_3, x)$
$(q_3, x) \in \Phi(q_3, x^-)$

$(q_3, x) \in \Phi(q_1, x^-)$

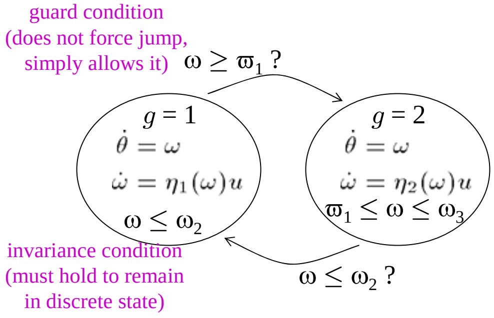
Compact representation of a nondeterministic hybrid automaton

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

---

# Nondeterministic Hybrid Automaton

## (Example #3: Semi-automatic transmission)



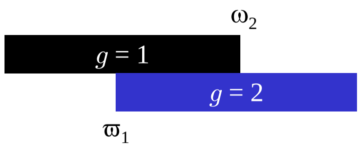
guard condition
(does not force jump,
simply allows it)
$\omega \geq \varpi_1$ ?

$g = 1$
$\dot{\theta} = \omega$
$\dot{\omega} = \eta_1(\omega) u$
$\omega \leq \omega_2$

$g = 2$
$\dot{\theta} = \omega$
$\dot{\omega} = \eta_2(\omega) u$
$\varpi_1 \leq \omega \leq \omega_3$

invariance condition
(must hold to remain
in discrete state)

$\omega \leq \omega_2$ ?

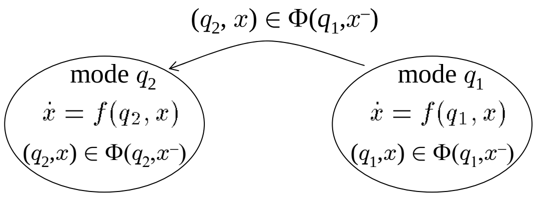


$\mathcal{Q} := \{ 1, 2 \}$
$\mathbb{R}^2$ ≡ continuous state-space

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

## Solution to a nondeterministic hybrid automaton

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



$(q_2, x) \in \Phi(q_1,x^-)$

mode $q_2$
$\dot{x} = f(q_2,x)$
$(q_2,x) \in \Phi(q_2,x^-)$

mode $q_1$
$\dot{x} = f(q_1,x)$
$(q_1,x) \in \Phi(q_1,x^-)$

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

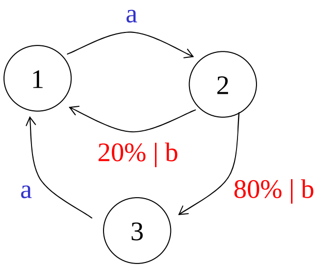
continuous evolution

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

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 $M$

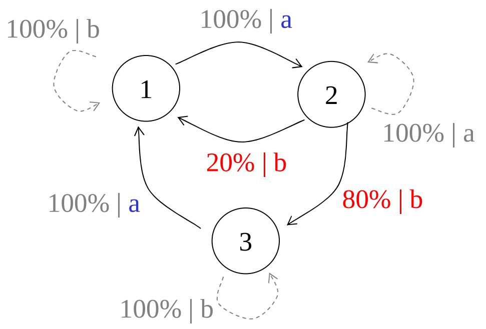
$\mathcal{Q} := \{q_1, q_2, \ldots, q_n\}$ ≡ finite set of states
$\Sigma := \{a, b, c, \ldots\}$ ≡ finite set of input symbols
$\Phi : \mathcal{Q} \times \mathcal{Q} \times \Sigma \to [0,1]$ ≡ transition probability function

$\Phi(q_1, q_2, s) \equiv$ 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

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

$\Phi(q_1, q_2, s) \equiv$ 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

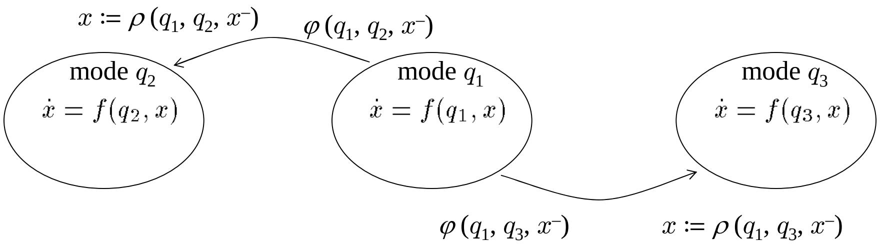
$f : \mathcal{Q} \times \mathbb{R}^n \to \mathbb{R}^n$ $\equiv$ vector field

$\varphi : \mathcal{Q} \times \mathcal{Q} \times \mathbb{R}^n \to [0,\infty]$ $\equiv$ 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)

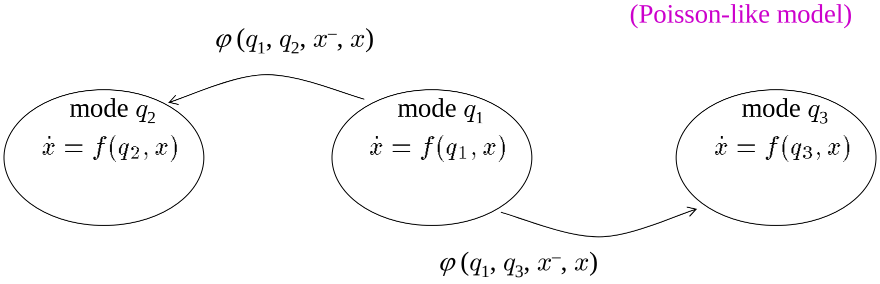


## Stochastic Hybrid Automaton

$\mathcal{Q}$ ≡ 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

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

(Poisson-like model)



$\varphi\,(q_1, q_2, x^-, x)$

mode $q_2$
$\dot{x} = f(q_2, x)$

mode $q_1$
$\dot{x} = f(q_1, x)$

mode $q_3$
$\dot{x} = f(q_3, x)$

$\varphi\,(q_1, q_3, x^-, x)$

*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