Modeling and Control of Hybrid Systems

Optimal control of Switched Systems

Daniele Corona

d.corona@tudelft.nl

February 14, 2007

Outline

Background

– Properties of linear autonomous vector fields
– Properties of the LQ cost
– Basic concepts of dynamic programming
– Polar coordinates in $\mathbb{R}^n$

Background

Properties of linear autonomous vector fields

Affine systems: $\dot{\xi} = \mathcal{A}\xi + \mathcal{F}$, $\xi \in \mathbb{R}^n$

Equivalent to $\dot{x} = Ax$ where $A = \begin{bmatrix} \mathcal{A} & \mathcal{F} \\ 0 & 0 \end{bmatrix}$ and $x = \begin{bmatrix} \xi \\ x_{n+1} \end{bmatrix}$, $x_{n+1}(0) = 1$, $x \in \mathbb{R}^{n+1}$

For systems of the class

$$\dot{x} = Ax$$

we have

State Transition Matrix:

$$x(t) = \Phi(t, t_0)x(t_0) = \int_{t_0}^{t} Ax(\tau)d\tau = e^{A(t-t_0)}x_0$$

where $x_0 = x(t_0)$ is the initial condition

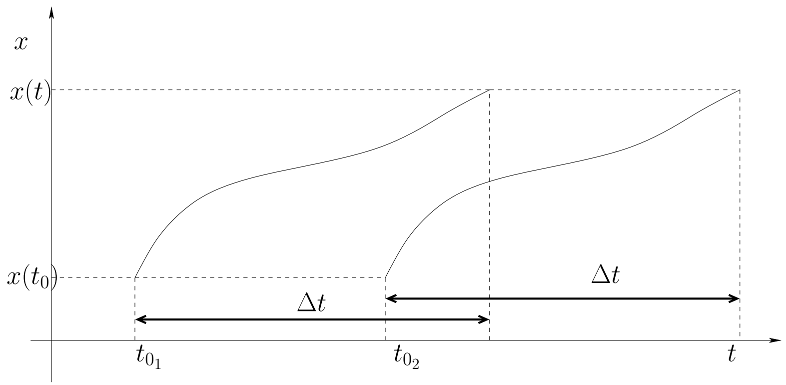
proof: Verifies initial condition and the DE

Background

Properties of linear autonomous vector fields

Time invariant:

$$x(t) = \Phi(t, t_0)x(t_0) = e^{A(t-t_0)}x(t_0) = e^{A\Delta t}x(t_0)$$



Final $x(t)$ depends **ONLY** on $x(t_0)$ and Δt, but **NOT** on t, t_0

D. Corona Optimal control of Switched Systems

Background

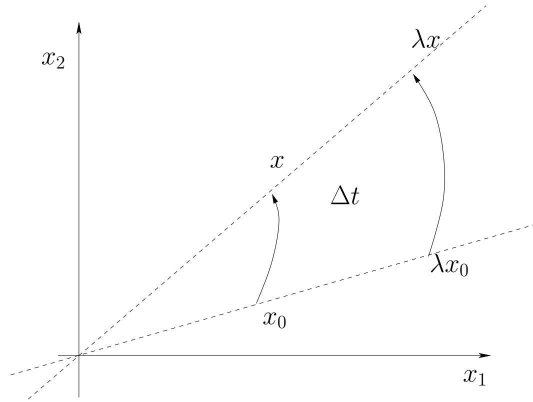
Properties of linear autonomous vector fields

Homogeneity: given a point $x_0 \in \mathbb{R}^n$ and a non zero scalar $\lambda \in \mathbb{R}$, for all t

$$x(t) = e^{At}x_0$$

and

$$\lambda x(t) = e^{At}\lambda x_0$$



D. Corona Optimal control of Switched Systems

Background

Properties of linear autonomous vector fields

Stability arguments:
Lyapunov: the vector field $\dot{x} = Ax$ is stable if $\forall \varepsilon > 0\ \exists\, \delta_\varepsilon$ such that $\forall\, \|x(t_0)\| < \delta_\varepsilon$ and $\forall\, t > t_0\ \|x(t) = \Phi(t, t_0, x(t_0))\| < \varepsilon$

Asymptotic stability :

$$\lim_{t \to +\infty} \|x\| = 0$$

(Convergence of the norm)

For linear autonomous vector fields (Matrix A is *Hurwitz*):

$$\text{Asymptotically stable} \iff Re(\lambda_i) < 0\ i = 1, \dots, n$$

λ_i: zeros of the n-order polynomial

$$det(A - \lambda I) = 0$$

For linear case:

$$\text{Asymptotically stable} \iff \text{Exponentially stable}$$

D. Corona Optimal control of Switched Systems

Background

Properties of the cost

Cost of the evolution until time $t > t_0$ from the initial point x_0, with the dynamics A
Weight matrix $Q \geq 0$, initial point x_0 and a matrix A

$$J(x_0, t_0, t) \triangleq \int_{t_0}^{t} x'(\tau)Qx(\tau)d\tau,$$

subject to

$$\dot{x} = Ax, \quad x(t_0) = x_0$$

Basic properties

- $J(x_0, t_0, t)$ is finite if $t < +\infty$ and positive if $Q > 0, t > t_0$
- $J(x_0, t_0, t_0) = 0$
- if $Q > 0$, $J(x_0, t_0, +\infty) = \begin{cases} < +\infty & \text{if } A \text{ is strictly Hurwitz} \\ +\infty & \text{else} \end{cases}$
- $J(x_0, t_0, t) = J(x_0, t_0, +\infty) - J(x_0, t, +\infty)$

Background

Properties of the cost

$$J(x_0, t_0, t) = \int_{t_0}^{t} x'(\tau) Q x(\tau) d\tau,$$

1. *Quadratically homogeneous* and *even* over x: for all $\lambda \neq 0$

$$J(x_0, t_0, t) = \frac{1}{\lambda^2} J(\lambda x_0, t_0, t)$$

2. There exists a matrix $M(t - t_0)$ s.t.

$$J(x_0, t_0, t) = x_0' M(t - t_0) x_0$$

Proofs-

Point 1: TRIVIAL - Exercise

Point 2: $J(x_0, t_0, t) = \int_{t_0}^{t} x'(\tau) Q x(\tau) d\tau = \int_{t_0}^{t} x_0' (e^{A(\tau - t_0)})' Q e^{A(\tau - t_0)} x_0 d\tau$, hence

$$J(x_0, t_0, t) = x_0' \int_0^{t - t_0} e^{A'\vartheta} Q e^{A\vartheta} d\vartheta \; x_0 = x_0' M(t - t_0) x_0$$

Background

Properties of the cost

Particular (important) case:
When A is Hurwitz then the Lyapunov equation

$$A'Z + ZA = -Q$$

admits a **unique** solution and

$$J(x_0, t_0, t) = x_0' \big(Z - e^{A'(t - t_0)} Z e^{A(t - t_0)} \big) x_0$$

Proof: $J(x_0, t_0, t) = -\int_{t_0}^{t} x'(\tau)(A'Z + ZA) x(\tau) d\tau = -\Big(\int_{t_0}^{t} \dot{x}' Z x d\tau + x' Z \dot{x} d\tau \Big) =$

$$-\int_{t_0}^{t} \frac{dx' Z x}{d\tau} d\tau = -\int_{t_0}^{t} dx' Z x = x'(t_0) Z x(t_0) - x'(t) Z x(t) = x_0' \big(Z - e^{A'(t - t_0)} Z e^{A(t - t_0)} \big) x_0$$

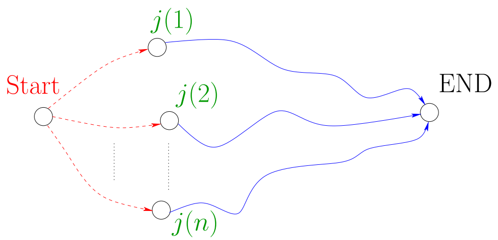
NB: In Matlab use commands
`>>Z=lyap(A',Q)` to compute Z and
`>>eAt=expm(A*t)` to compute e^{At}
If Z does not exist then the cost $J(x_0, t_0, t)$ must be computed numerically

Background

Basic concepts of dynamic programming

Principle of optimality [Bellman, 1957]

"An optimal policy has the property that
whatever the initial state and the initial decisions are,
the remaining decisions must constitute an optimal policy
with regard to the state resulting from the first decision"



Background

Basic concepts of dynamic programming

A formulation of the *principle of optimality*

$$J_k(x(t)) = \min_{u(t)} \Big(J_{k+1}(x(t+T)) + \int_t^{t+T} L(x(\tau), u(\tau)) d\tau \Big)$$

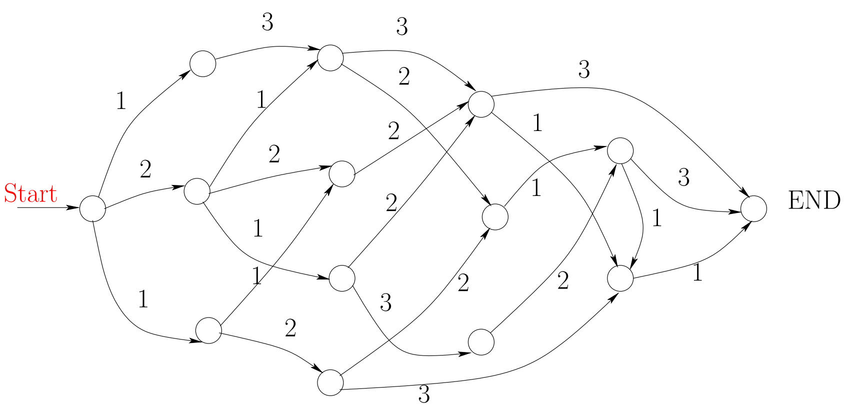
$$k = 1, \dots, N$$

and $J_N(x)$ given as a boundary condition

Background

Basic concepts of dynamic programming

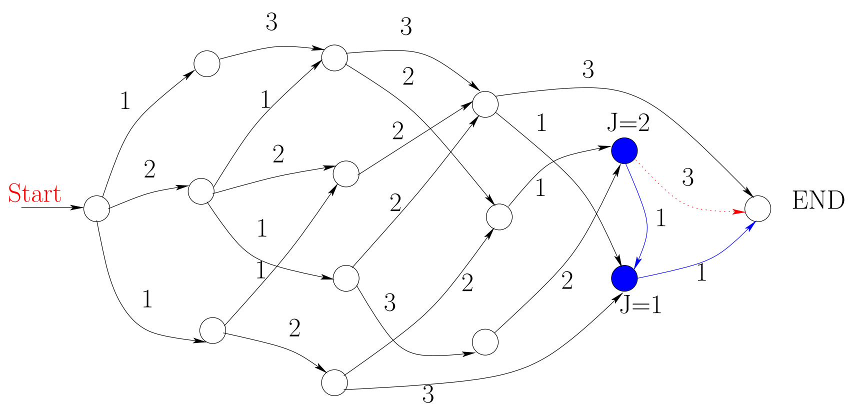
Example: Minimize the toll of the path from *Start* to *END*



Background

Basic concepts of dynamic programming

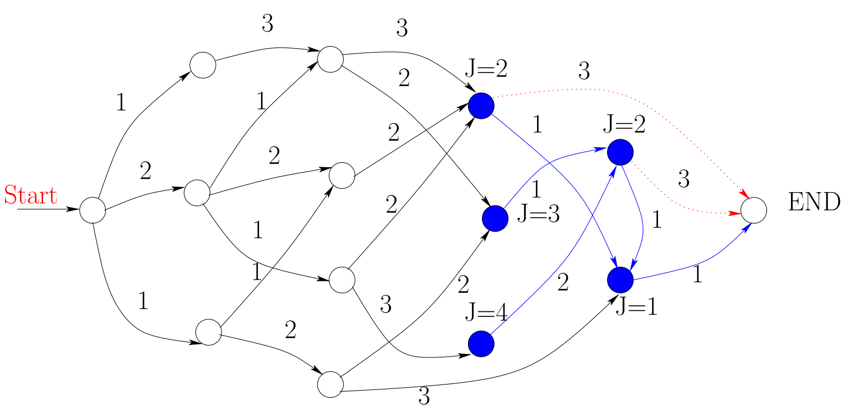
Example: Stage 1



Background

Basic concepts of dynamic programming

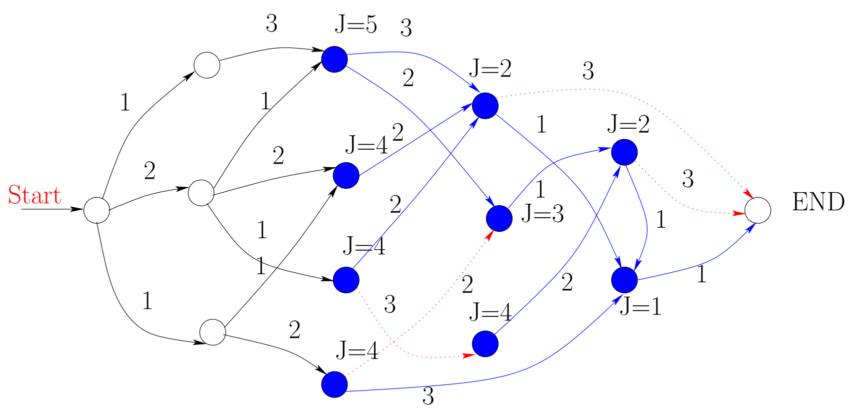
Example: Stage 2



Background

Basic concepts of dynamic programming

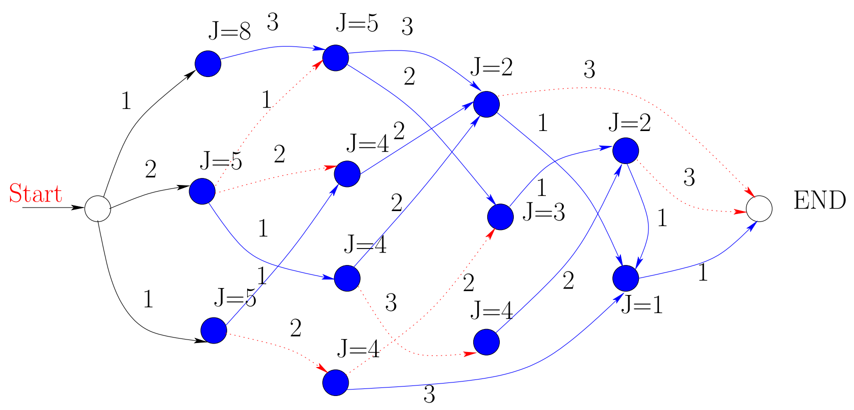
Example: Stage 3



Background

Basic concepts of dynamic programming

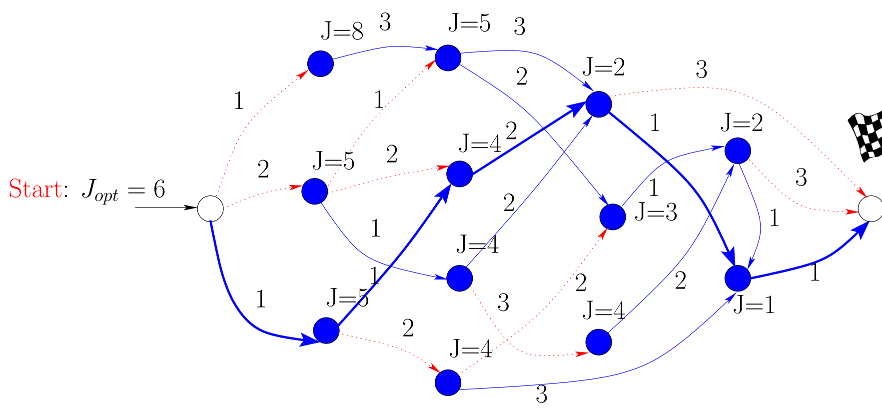
Example: Stage 4



Background

Basic concepts of dynamic programming

Example: Stage 5

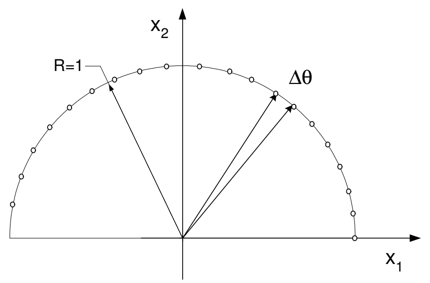


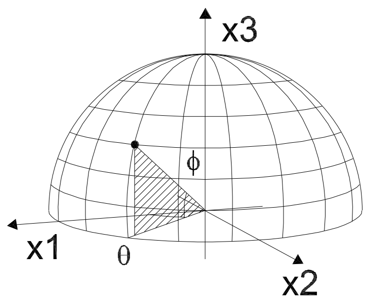
NOTE: State feedback, if disturbance occur...

Background

Polar coordinates in $\mathbb{R}^n$

The properties of homogeneity of the autonomous vector field $\dot{x} = Ax$ and the quadratic form of the cost function allow to keep the attention on the *unitary semi sphere*





This geometry advises to reconsider the formulation in polar coordinates

Background

Polar coordinates in $\mathbb{R}^n$

Given a vector $x = [x_1, x_2, \ldots, x_n] \in \mathbb{R}^n$, its *polar* representation will be given by a radius ρ_n and $n-1$ angles θ_i, $i = 2, \ldots, n$ related as

$$\begin{cases} x_n = \rho_n \sin(\theta_n) \\ x_{n-1} = \rho_{n-1} \sin(\theta_{n-1}) \\ \vdots \\ x_3 = \rho_3 \sin(\theta_3) \\ x_2 = \rho_2 \sin(\theta_2) \\ x_1 = \rho_2 \cos(\theta_2) \end{cases}$$

where $\rho_n = \|x\|$, $\rho_i = \rho_{i+1} \cos(\theta_i)$ for $i = n-1, \ldots, 2$.

Ranges

$$\rho_n \in [0, +\infty)$$
$$\theta_2 \in [0, 2\pi)$$
$$\theta_3, \ \ldots, \ \theta_n \in [-\frac{\pi}{2}, \frac{\pi}{2}]$$

The unitary semi sphere Σ_n:

$\rho_n = 1$, $\theta_2 \in [0, 2\pi)$, $\theta_3, \ \ldots, \ \theta_{n-1} \in [-\frac{\pi}{2}, \frac{\pi}{2})$, and $\theta_n \in [0, \frac{\pi}{2}]$.

Outline

1. Background
2. Switched systems
3. Problem definition
4. Switching table procedure
5. Computational complexity

Switched Systems

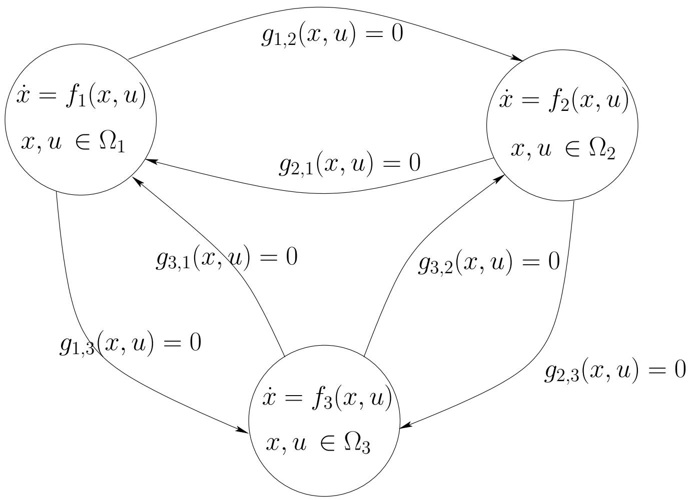
– Description

– A paradox

– Example: buck-boost converter

Switched Systems

Description



Switched Systems

Description

Locations ℓ_i, labels: indicate the current operating mode

Vector fields f_i, functions: govern the evolution in mode ℓ_i

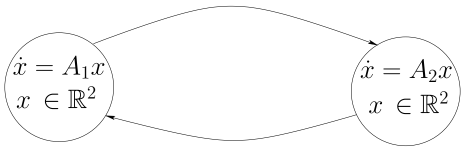
Constraint sets Ω_i, domain: restrict the area where mode ℓ_i is *allowed*

Guards $g_{i,j}$, switching manifold: trigger the switch from $\ell_i \to \ell_j$

Continuous state: $x \in \mathbb{R}^n$
Continuous control signal: $u \in \mathbb{R}^m$
Discrete control signal: $i \in \mathcal{S}$

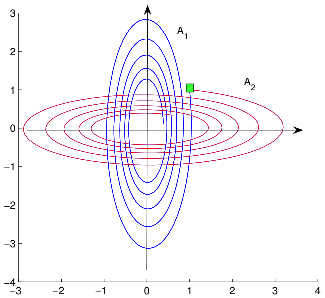
Switched Systems

Paradox [Branicky, 1998]



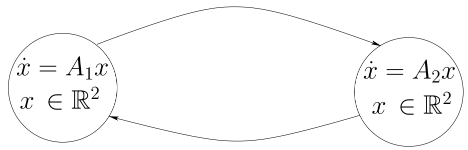
$$A_1 = \begin{bmatrix} -1 & 10 \\ -100 & -1 \end{bmatrix} \quad \text{Stable}$$

$$A_2 = \begin{bmatrix} -1 & 100 \\ -10 & -1 \end{bmatrix} \quad \text{Stable}$$



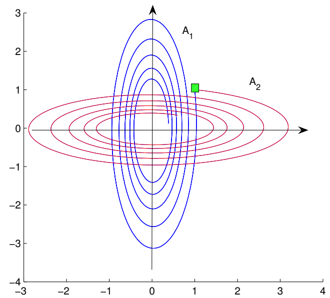
Switched Systems

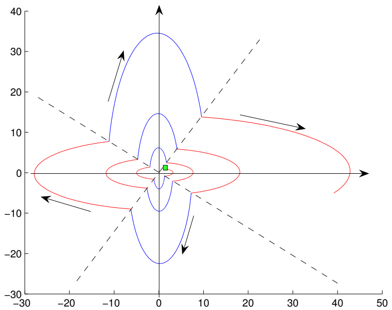
Paradox [Branicky, 1998]



$$A_1 = \begin{bmatrix} -1 & 10 \\ -100 & -1 \end{bmatrix} \quad \text{Stable}$$

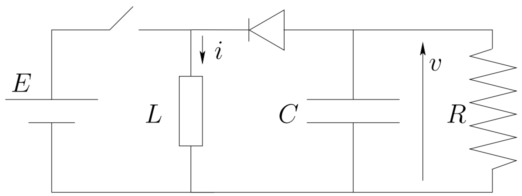
$$A_2 = \begin{bmatrix} -1 & 100 \\ -10 & -1 \end{bmatrix} \quad \text{Stable}$$





Switched Systems

Example: the buck-boost converter



Four possible modes: $x = \begin{bmatrix} i \\ v \end{bmatrix}$

1. Switch closed, diode blocked
2. Switch open, diode conducting
3. Switch open, diode blocked
4. Switch closed, diode conducting

$$\text{I:} \quad \dot{x} = \begin{bmatrix} 0 & 0 \\ 0 & -\frac{1}{RC} \end{bmatrix} x + \begin{bmatrix} \frac{E}{L} \\ 0 \end{bmatrix}$$

$$\text{II:} \quad \dot{x} = \begin{bmatrix} 0 & \frac{1}{L} \\ -\frac{1}{C} & -\frac{1}{RC} \end{bmatrix} x$$

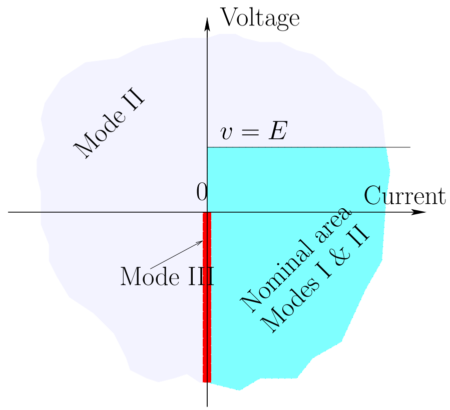
$$\text{III:} \quad \dot{x} = \begin{bmatrix} 0 & 0 \\ 0 & -\frac{1}{RC} \end{bmatrix} x$$

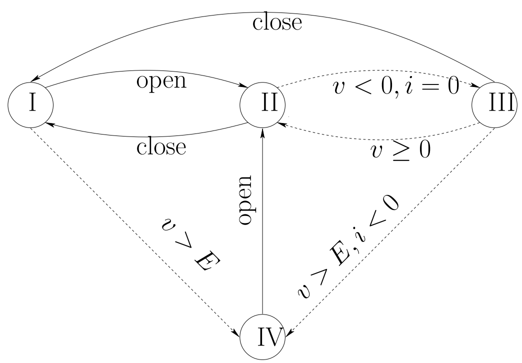
IV: Fault mode

Switched Systems

Example: the buck-boost converter

Switched system representation





Switched systems

Formal definition of switched systems

$S = (\mathcal{L}, act, \mathcal{E}, \mathcal{M})$, S
— $\mathcal{L}$: finite set of locations, $i = 1, \ldots, s$
— $act : \mathcal{L} \to (\mathbb{R}^n \times \mathbb{R}^n)$ **Affine differential equation**

$$\dot{x} = A_i x + f_i$$

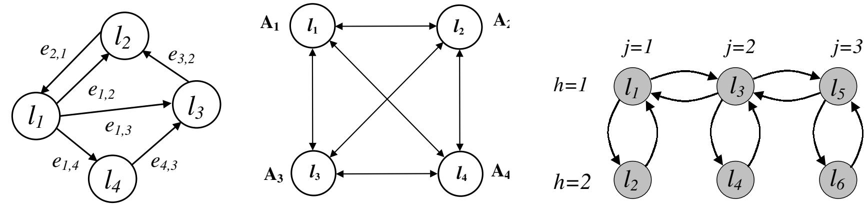
— $\mathcal{E} \subset \mathcal{L} \times \mathcal{L}$ **set of edges**
$e_{i,j} = (i, j)$: **arc** from i to j, $i \neq j$
— $\mathcal{M} : \mathcal{E} \to \mathbb{R}^{n \times n}$ **Switching reset**
Switch from i to j occurs $\Rightarrow x^+ = M_{i,j} x^-$.

Set of indexes: $\mathcal{S}$ associated to each location

Minimum permanence time $\delta_{\min}(i) \geq 0$ associated to each location i.

Switched Systems

Oriented graph: examples



Hybrid state: couple (x, i)
$x \in \mathbb{R}^n$: continuous state $\quad i \in \mathcal{S}$: discrete location

Hybrid evolution: $(x(t), i(t))$

Set of successors: $succ(i) \subset \mathcal{S}$ set of locations reachable from i

Outline

1. Background
2. Switched systems
3. Problem definition
4. Switching table procedure
5. Computational complexity

Problem definition

optimal control problem $\mathrm{OP}_N(S)$
N: upper bound (**finite**) on the number of switches
S: switched system as above

Formulation:

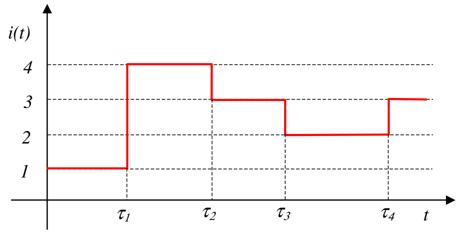
$$J_N^* \triangleq \min_{\mathcal{I},\mathcal{T}} \left\{ F(\mathcal{I}, \mathcal{T}) \triangleq \int_0^\infty x'(t) Q_{i(t)} x(t) dt + \sum_{k=1}^{N} H_{i_{k-1}, i_k} \right\}$$

$$\begin{array}{lll} \text{s.t.} & \dot{x} = A_{i(t)} x + f_{i(t)} & \\ & x(0) = x_0 & \\ & i(t) = i_k \quad \text{for } \tau_k \leq t < \tau_{k+1} & k = 0, \ldots, N \\ & i_{k+1} \in succ(i_k) & k = 0, \ldots, N \\ & \tau_0 = 0, \ \tau_{N+1} = +\infty & \\ & \tau_{k+1} \geq \tau_k + \delta_{\min}(i_k) & k = 0, \ldots, N \\ & x(\tau_k^+) = M_{i_{k-1}, i_k} x(\tau_k^-) & k = 1, \ldots, N \end{array}$$

Q_i: positive semi-definite matrices, weight continuous evolution
$H_{i,j}$: switching costs positive reals, weight discrete evolution
x_0: initial state of the system

Problem definition

Design the function $i(t)$



Two types of decision variables

$$\mathcal{T} \triangleq \{\tau_1, \ldots, \tau_N\}$$

Finite sequence of switching instants

$$\mathcal{I} \triangleq \{i_0, \ldots, i_N\}$$

Finite sequence of modes

Outline

1. Background
2. Switched systems
3. Problem definition
4. Switching table procedure
5. Computational complexity

Switching table procedure

– Assumptions
– Objectives
– Definitions
– Fundamental theorem
– Definition of the switching table
– Algorithm
– Application example

Switching table procedure

Assumptions

Finite solution of $\mathrm{OP}_N(S)$: it must hold

Assumption 1

(i) there exists at least one location $i \in \mathcal{S}$ such that A_i is strictly Hurwitz, $f_i = \mathbf{0}$;

(ii) if the initial location i_0 is imposed, than the number N of available switches is such that the location i must be reachable from i_0 in $k \leq N$ steps.

Comments

(i) At least one location i is characterized by a stable dynamics

(ii) This (these) location(s) i must be *reachable* within N switches

If i_0 is not assigned then (ii) can be relaxed

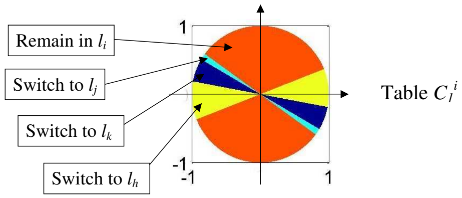
Switching table procedure

Objectives

Construct **offline** a partition of the state space

State feedback optimal control law: *checks* system state (x, i) and *decide* if a switch from location i should occur

Switching table: $\mathcal{C}_k^i$ of location i and k available switches



Recursively:
First construct tables $\mathcal{C}_1^i$ **Then** construct tables $\mathcal{C}_k^i$ using $\mathcal{C}_{k-1}^i$, $k = 1, \ldots, N$

Switching table procedure

Definitions

Assume

- Current state: (x, j_k)
- Missing switches: $k \geq 0$ out of N

Schedule of future *evolution* from (x, j_k):

Time intervals	ϱ_k	ϱ_{k-1}	$\ldots$	ϱ_0
Indexes $j(t)$	j_k	j_{k-1}	$\ldots$	j_0

Cost of **remaining** evolution:

$$J = \int_{\tau_{N-k}}^{\infty} x'(t) Q_{j(t)} x(t) dt + \sum_{i=k}^{1} H_{j_i, j_{i-1}}$$

Cost of *remaining evolution*: two parts

(i) *event driven* cost (SUM)

(ii) *time driven* cost (INTEGRAL)

Switching table procedure

Definitions

Time cost: recursive expression

At k available switches

$$\begin{aligned}\tilde{T}_k(x, j_k, \ldots, j_0, \varrho_k, \ldots, \varrho_0) &= x'\bar{Q}_{j_k}(\varrho_k)x + x'\bar{A}'_{j_k}(\varrho_k) M'_{j_k,j} \bar{Q}_j(\delta_{\min}(j)) M_{j_k,j} \bar{A}_{j_k}(\varrho_k) x \\ &+ \tilde{T}_{k-1}(z, j_{k-1}, \ldots, j_0, \varrho_{k-1}, \ldots, \varrho_0)\end{aligned}$$

At $k = 0$ available switches (initial stage of recursion)

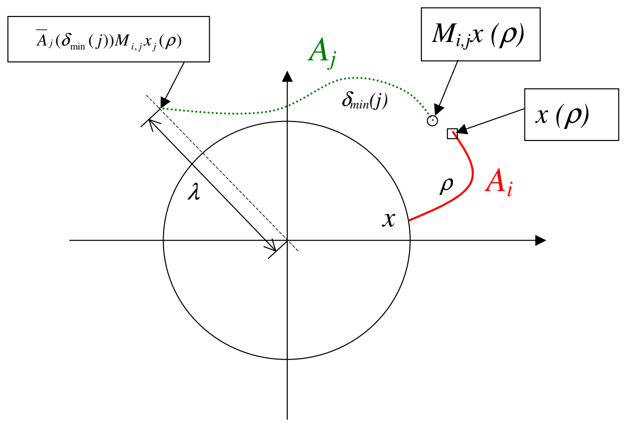
$$\tilde{T}_0(x, j_0, \varrho_0) = x'\bar{Q}_{j_0}(\varrho_0)x = x'\bar{Q}_{j_0}(+\infty)x$$

with $\bar{Q}(\varrho) = \int_0^{\varrho} x'(t) Q x(t) dt$ and $\bar{A}(\varrho) = e^{A\varrho}$

Switching table procedure

Definitions

The point $z = \bar{A}_{j_{k-1}}(\delta_{\min}(j_{k-1})) M_{j_k, j_{k-1}} \bar{A}_{j_k}(\varrho_k) x$ is reached after the evolution in the current dynamics j_k and the $\delta_{\min}(j_{k-1})$



Switching table procedure

Definitions

Event cost: recursive expression

At k available switches

$$E_k(j_k,\dots,j_0) = H_{j_k,j_{k-1}} + E_{k-1}(j_{k-1},\dots,j_0)$$

At $k=0$ available switches (initial stage of recursion)

$$E_0(j_0) = 0$$

Total residual cost

$$T_k(x,j_k,\dots,j_0,\varrho_k,\dots,\varrho_0) = \tilde{T}_k(x,j_k,\dots,j_0,\varrho_k,\dots,\varrho_0) + E_k(j_k,\dots,j_0)$$

Switching table procedure

Theorem

Minimize the function *residual cost* $T_k(x,j_k,\dots,j_0,\varrho_k,\dots,\varrho_0)$

$$T_k^*(x,j_k) = \min_{\substack{\{j_{k-1},\dots,j_0\}\\ \{\varrho_k,\dots,\varrho_1\}}} T_k(x,j_k,\dots,j_0,\varrho_k,\dots,\varrho_0)$$

constrained to

$$\begin{array}{l} \varrho_0 = +\infty \\ \varrho_h \geq \delta_{\min}(j_h) \\ \varrho_k \geq 0 \\ j_k = i \\ j_h \in succ(j_{h+1}) \cup \{j_h\} \\ h = 0,\dots,k-1. \end{array}$$

Mixed integer quadratic programming (MIQP) problem

Integer variables $\{j_{k-1},\dots,j_0\} \in 2^{\mathcal{S}}$
Continuous variables $\{\varrho_k,\dots,\varrho_1\}$

Switching table procedure

Theorem

Solution 1: "brute force" method

Advantages: appropriate numerical tools for MIQP (e.g., *CPLEX*)

Disadvantages: mixed integer (!!)

Solution 2: heuristic methods (Genetic algorithms, Simulated annealing,...)

Disadvantages: sub-optimal solutions

Advantages: velocity

Switching table procedure

Theorem

***Theorem*: Optimal remaining cost**
k switches available, current hybrid state x, j_k.

1. If $k=0$ then the remaining optimal cost starting from x is:

$$T_0^*(x,j_0,\varrho_0) = \tilde{T}_0(x,j_0,\varrho_0)$$

2. If $k \in \{1,\dots,N\}$ then:

(i) Remaining optimal cost starting from x is:

$$T_k^*(x,j_k) = \min_{j_{k-1} \in succ(j_k) \cup \{j_k\} \varrho_k \geq 0} F(x,j_k,j_{k-1},\varrho_k) + T_{k-1}^*(z(x,j_k,j_{k-1},\varrho_k),j_{k-1})$$

(ii) Next optimal dynamics

$$j_{k-1}^*(x,j_k) = \arg \min_{\substack{j_{k-1} \in succ(j_k) \cup \{j_k\}\\ \varrho_k \geq 0}} F(x,j_k,j_{k-1},\varrho_k) + T_{k-1}^*(z(x,j_k,j_{k-1},\varrho_k),j_{k-1})$$

(iii) Switching time $\varrho_k^*(x,j_k)$

$$\varrho_k^*(x,j_k) = \arg \min_{\substack{j_{k-1} \in succ(j_k) \cup \{j_k\}\\ \varrho_k \geq 0}} F(x,j_k,j_{k-1},\varrho_k) + T_{k-1}^*(z(x,j_k,j_{k-1},\varrho_k),j_{k-1})$$

Switching table procedure

Theorem

Proof
Case 1: $k = 0$. The systems is forced to evolve with dynamics A_{j_k} to infinity hence the remaining optimal is given
Case 2: $k > 0$.

$$T_k^*(x, j_k) = \min_{\substack{\{j_{k-1},\dots,j_0\} \\ \{\varrho_k,\dots,\varrho_1\}}} T_k(x, j_k, \dots, j_0, \varrho_k, \dots, \varrho_0)$$

but

$$T_k(x, j_k, \dots, j_0, \varrho_k, \dots, \varrho_0) = F(x, j_k, j_{k-1}, \varrho_k) + T_{k-1}(x, j_{k-1}, \dots, j_0, \varrho_{k-1}, \dots, \varrho_0)$$

and by the principle of optimality (choose optimal T_{k-1}^*)

$$T_k^*(x, j_k) = \min_{\substack{\{j_{k-1} \in succ(j_k) \cup \{j_k\}\} \\ \{\varrho_k \geq 0\}}} F(x, j_k, j_{k-1}, \varrho_k) + T_{k-1}^*(z(x, j_k, j_{k-1}, \varrho_k), j_{k-1})$$

Only one discrete and one continuous variable

$$(j_{k-1}, \varrho_k)$$

Switching table procedure

Definition of the switching table

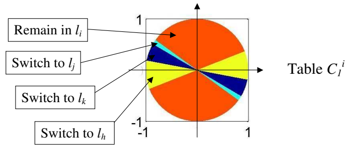
$\varrho_k^*(x, i) \geq 0$: time spent in the current location i

$\varrho_k^*(x, i) = 0 \Rightarrow$ **Switch immediately** to location j_{k-1}^*
$\varrho_k^*(x, i) > 0 \Rightarrow$ **Remain** in the current location i

$\mathcal{C}_k^i$ (switching table): partition of $\mathbb{R}^n$

(i) $\mathcal{R}_j$: it is optimal to **switch immediately** to j^*

(ii) $\mathcal{R}_i$: it is optimal to **remain** in i



If switching costs are null then the regions are **conic** $\longrightarrow$ Unitary semi sphere

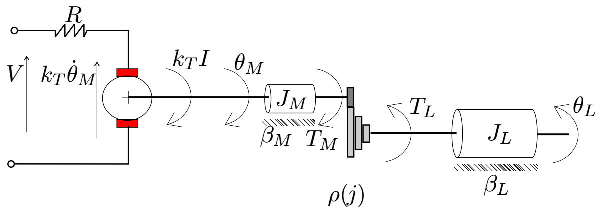
Switching table procedure

Algorithm/Pseudo-code

1. $k = 0$ `% remaining switches (initialize)`
   $\forall\ (y, i)\ \in\ \Sigma_n \times \mathcal{S}$
   $Z_i\ :\ A_i' Z_i + Z_i A_i = -Q_i.$
   $$T_0(y, i) = \begin{cases} y' Z_i y & \text{if} \quad \exists\ Z_i > 0 \\ +\infty & \text{else.} \end{cases}$$
   $C_0(y, i) = i$
2. For $k = 1 : N$
   $\forall\ (y, i)\ \in\ \Sigma_n \times \mathcal{S}$
   Compute the set $succ(i)$
   $T_k(y, i) = \min_{j \in succ(i) \cup \{i\}, t \geq 0} T(y, i, j, t)$ `% Defined above`
   $(j^*, t^*) = \arg\min_{j,t} T(y, i, j, t).$
   $$C_k(y, i) = \begin{cases} j^* & \text{if} \quad t^* = 0 \\ i & \text{if} \quad t^* > 0. \end{cases}$$
   end (k)

Switching table procedure

Application: Servomechanism system



System dynamics

$$V = RI + k_T\dot{\theta}_M,\ J_M\ddot{\theta}_M = k_T I - \beta_M\dot{\theta}_M - T_M$$
$$\dot{\theta}_M = \rho(j)\dot{\theta}_L,\ T_L = \rho(j)T_M,\ J_L\ddot{\theta}_L = -\beta_L\dot{\theta}_L + T_L$$

Linear differential equation

$$\left[J_L + \rho^2(j)J_M\right]\ddot{\theta}_L + \left[\beta_L + \rho^2(j)\left(\frac{k_T^2}{R} + \beta_M\right)\right]\dot{\theta}_L = \rho(j)\frac{k_T}{R}V.$$

V: PI controller of two levels

$$V = -k_1(h)\theta_L - k_2(h)\dot{\theta}_L, \quad h = 1, 2$$

Switching table procedure

Application: Servomechanism system

State space variables:

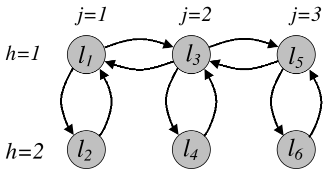
$$x \triangleq [\theta_L, \dot{\theta}_L]'$$

Autonomous switched linear system representation

$$\dot{x} = A(h,j)x = \begin{bmatrix} 0 & 1 \\ a_{21}(h,j) & a_{22}(h,j) \end{bmatrix} x$$

$a_{21}(h,j), a_{22}(h,j)$: gear/voltage configuration

Oriented graph



Switching table procedure

Application: Servomechanism system

Numerical values

Symbol	Value (IS)	Physical meaning
J_M	1	motor inertia
β_M	0.2	motor friction coefficient
R	50	resistance of armature
k_T	15	motor constant
J_L	50	nominal load inertia
β_L	10	load friction coefficient
ρ	1,2,3	gear ratios
$k_1(1)$	3.2	proportional action (smooth)
$k_1(2)$	31.6	proportional action (aggressive)
$k_2(1)$	3.5	integral action (smooth)
$k_2(2)$	32.1	integral action (aggressive)

$$A_1 = \begin{bmatrix} 0 & 1 \\ -0.019 & -0.31 \end{bmatrix} \quad A_3 = \begin{bmatrix} 0 & 1 \\ -0.036 & -0.57 \end{bmatrix} \quad A_5 = \begin{bmatrix} 0 & 1 \\ -0.049 & -0.94 \end{bmatrix}$$

$$A_2 = \begin{bmatrix} 0 & 1 \\ -0.186 & -0.47 \end{bmatrix} \quad A_4 = \begin{bmatrix} 0 & 1 \\ -0.351 & -0.89 \end{bmatrix} \quad A_6 = \begin{bmatrix} 0 & 1 \\ -0.482 & -1.38 \end{bmatrix}$$

Switching table procedure

Application: Servomechanism system

Numerical simulations

Additional values:

- Number of switches $N = 5$
- No switching costs
- Minimum permanence time $\delta_{min} = 0.2\ s$
- Initial state: $x_0 = [-1.4, 1.5]'$
- Initial location $i_0 = 1$
- Weight matrices

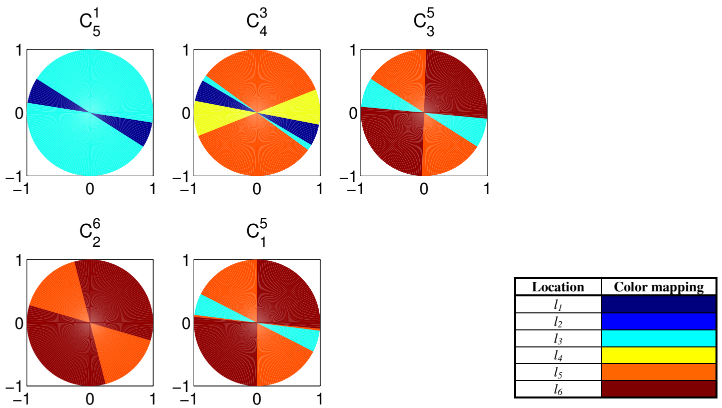
$$Q_1 = Q_3 = Q_5 = \begin{bmatrix} 1 & 0 \\ 0 & 2 \end{bmatrix} \quad Q_2 = Q_4 = Q_6 = \begin{bmatrix} 3 & 0 \\ 0 & 6 \end{bmatrix}$$

Switching table procedure

Application: Servomechanism system

Step 1:

Evaluation offline of $(N = 5 \times \mathcal{S} = 6) = 30$ switching tables

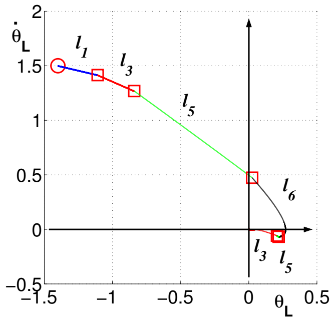


Note: Tables interested by the evolution from the given initial state

Switching table procedure

Application: Servomechanism system

Step 2:
Perform an evolution from the given initial state by using the 30 switching tables



Hybrid evolution that minimizes the LQ performance index
Circle: indicates the initial state
Squares: indicate the values of the state at the switching instants

Switching table procedure

Application: Servomechanism system

Relevant observation
The performance of the system is related to the number of available switches.

From the same initial state, the performance index J varies when $k = 0, \dots, 6$ switches are available

Available switches	Index Value
0	108.62
1	20.78
2	6.69
3	4.84
4	4.84
5	4.75
6	4.69

Outline

1. Background
2. Switched systems
3. Problem definition
4. Switching table procedure
5. Computational complexity

Computational complexity

Computational complexity of the procedure:

$$O(Ns(s-1)r^{n-1}N_t)$$

where

- N is the number of available switches
- s is the cardinality of the set $\mathcal{S}$
- r is the number of samples along each direction of $\mathbb{R}^n$
- N_t is the number of time samples used in the minimization over time

The complexity of the algorithm is **polynomial** over the number of switches and the number of modes

It is **exponential** over the space discretization (limits high dimensional applications)

Some conclusions

- STP for finite number of switches
- Feedback solution
- Offline
- Global optimum
- State space discretization (accuracy vs. time consumption)
- Studied extensions
  1. Switching constraints on the state space
  2. Infinite number of switches
  3. Design of a stabilizing switching signal

Thank you