Workshop

Future Directions in Systems and Control Theory

Thursday , June 24

Viewgraphs - Volume 3

A Framework for Control, State Estimation, and Verification of Hybrid Systems

Manfred Morari

Alberto Bemporad
Giancarlo Ferrari-Trecate
Domenico Mignone
Fabio Danilo Torrisi

Automatic Control Laboratory
Swiss Federal Institute of Technology (ETH)
morari@aut.ee.ethz.ch

Technical Reports: http://www.aut.ee.ethz.ch

ETH Eidgenössische Technische Hochschule Zürich

Premise

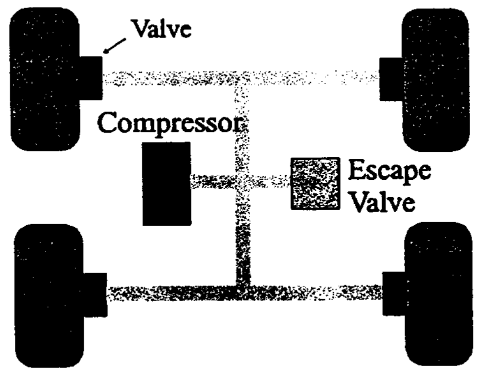
- The discussed problems are important but inherently difficult
- All useful techniques must involve significant off-line and/or on-line computation

Results

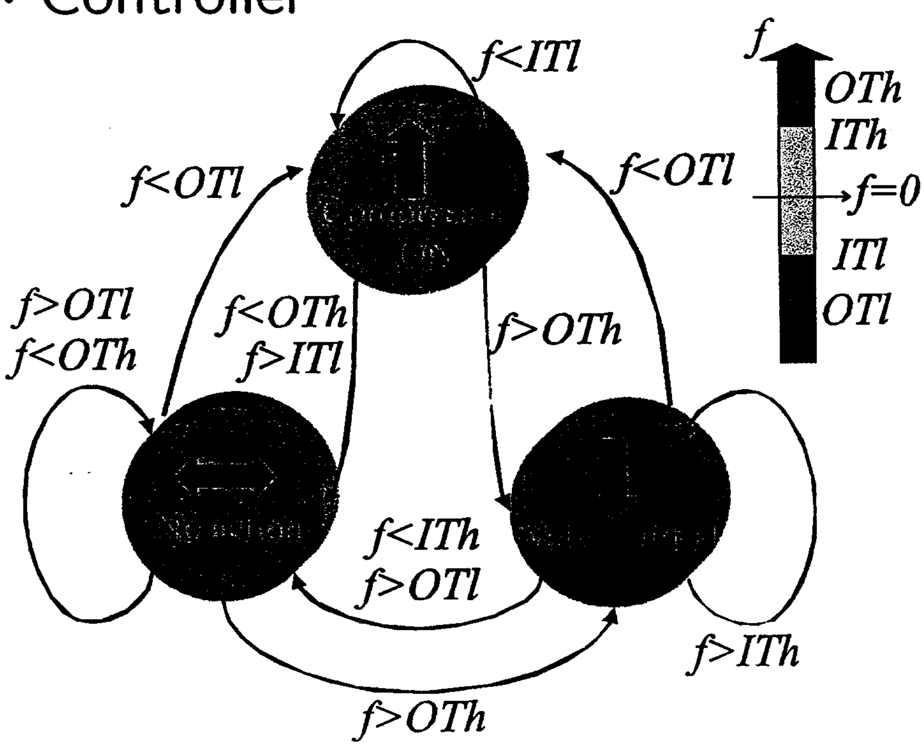
- New System Type:
  Mixed Logical Dynamical (MLD) System
- Many practical problems can be represented in MLD form
- Control, estimation, and verification require solution of Mixed-Integer Linear (or Quadratic) Programs (MILP, MIQP) for which efficient techniques are becoming available

Example: Verification of an Automotive Active Leveler

- System



- Controller



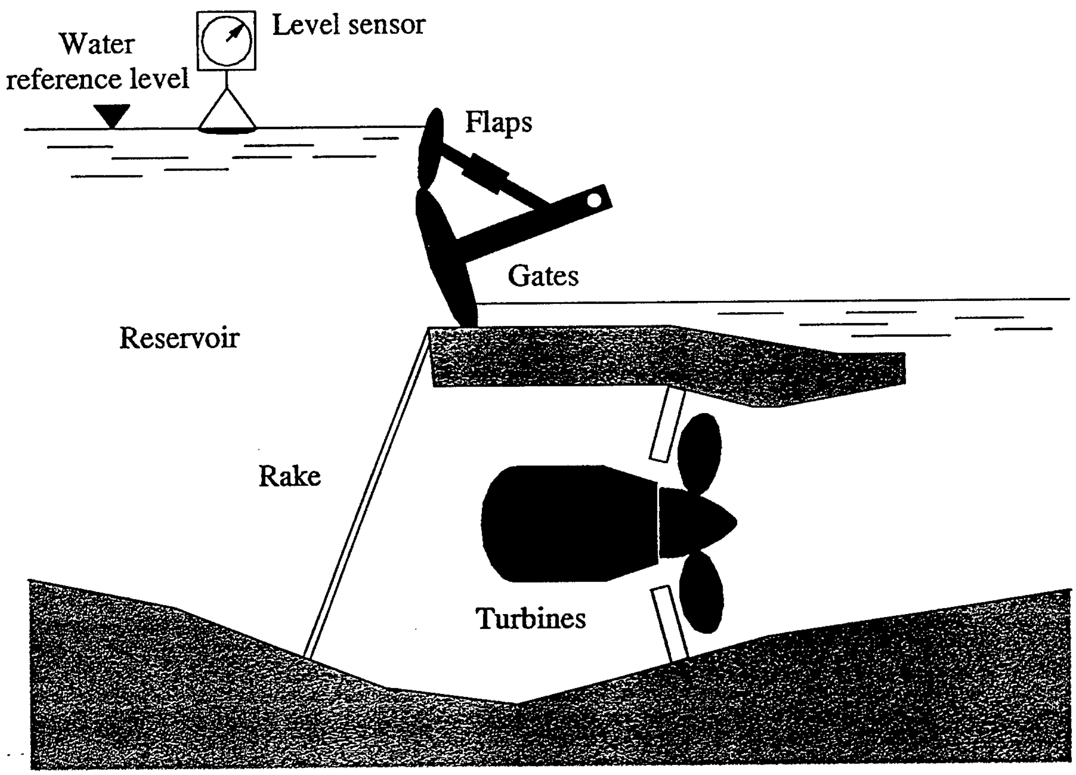
$$f = \frac{1}{1+as}h$$

- For a given set of initial conditions, variation of parameters, and road conditions:

(VP#1) Find worst performance

(VP#2) Verify that height is within desired limits

Example: Control of a Hydroelectric Power Plant



Objective:

- Maximize power generation

Manipulated variables:

- Turbine ON / OFF
- Stepper motors $(-1, 0, 1)$

Hydroelectric Power Plant

Constraints

- on actuators (flaps, gates)
- on level

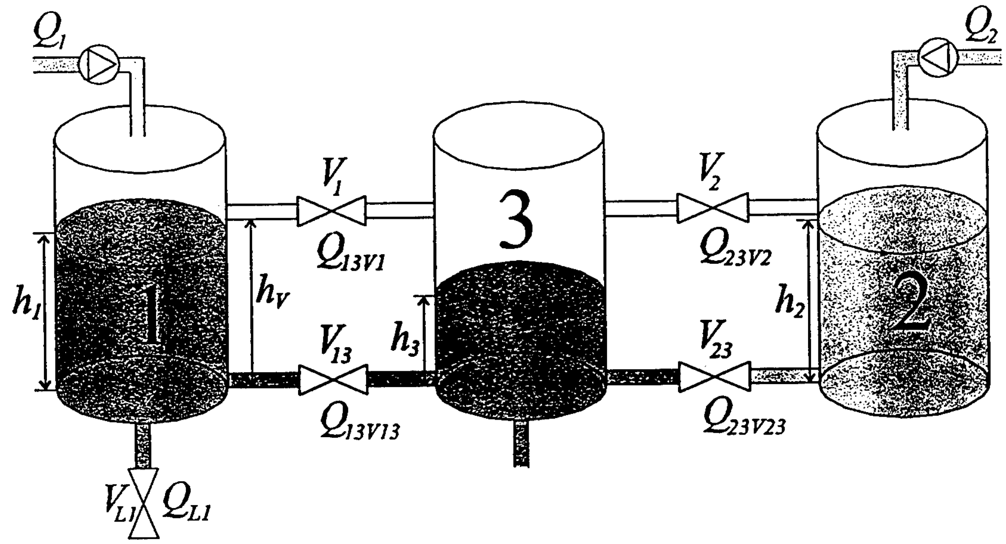
Targets (= end point constraints)

- level
- actuators (flaps, gates)

Preference hierarchy / heuristics

1. Flaps rather than gates
2. Turbine ON $> T_{min}$
3. Stepper motor ON $> T'_{min}$
4. Transitions: turbines / flaps / gates
5. Stepper motor OFF $< T_{max}$

Example: Three Tank System



Nominal behaviour:

- Liquid level in tank 1 controlled by pump 1
- Liquid level in tank 3 controlled by switching valve V_1

Faults:

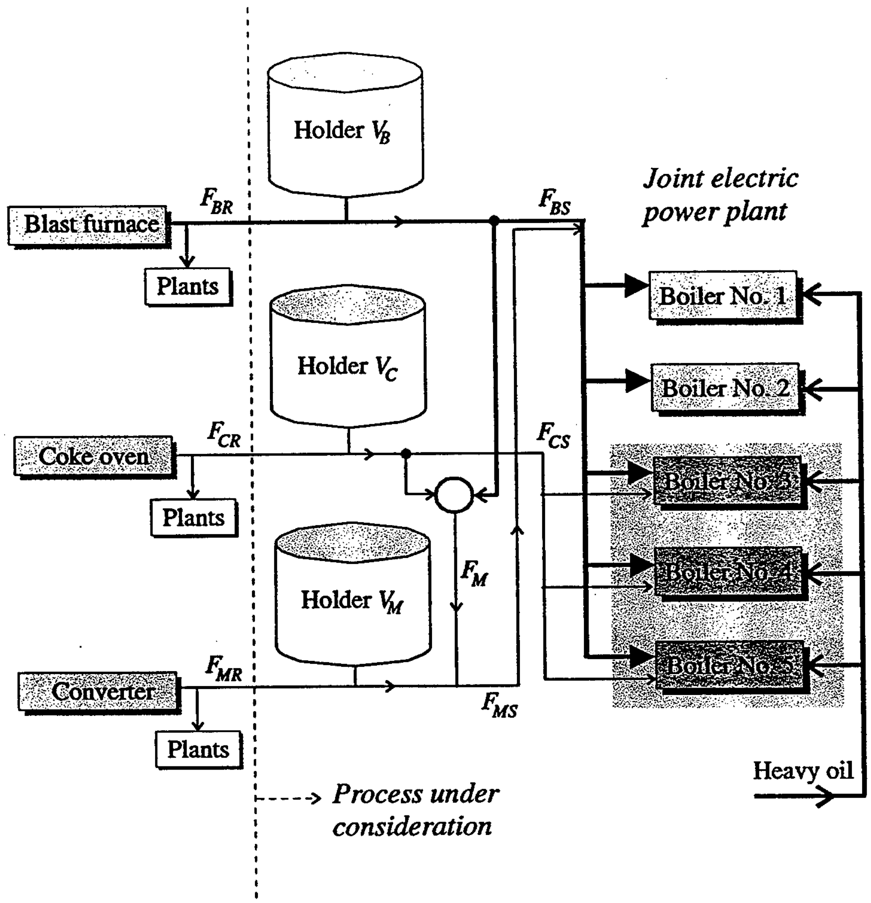
ϕ_1: Leak in tank 1

ϕ_2: Valve V_1 blocked closed

Maintaining nominal operating condition requires fault detection & control reconfiguration

Example: Gas Supply System

(Kawasaki Steel)



- Maintain hold-up
- Maximize use of gas for combustion
- Limitations on gas flows, heat produced, gas volume in holders
- Complex rules for boiler switching

Premise

- The discussed problems are important but inherently difficult
- All useful techniques must involve significant off-line and/or on-line computation

Results

- New System Type:
  Mixed Logical Dynamical (MLD) System
- Many practical problems can be represented in MLD form
- Control, estimation, and verification require solution of Mixed-Integer Linear (or Quadratic) Programs (MILP, MIQP) for which efficient techniques are becoming available

General Class: MLD Systems

General formulation of Mixed Logical-Dynamic (MLD) systems

$$
\begin{aligned}
x(t+1) &= Ax(t) + B_1u(t) + B_2\delta(t) + B_3z(t) \\
y(t) &= Cx(t) + D_1u(t) + D_2\delta(t) + D_3z(t) \\
E_2\delta(t) &+ E_3z(t) \leq E_1u(t) + E_4x(t) + E_5
\end{aligned}
$$

- State :

$$x = \begin{bmatrix} x_c \\ x_\ell \end{bmatrix}, \; x_c \in \mathbb{R}^{n_c}, \; x_\ell \in \{0,1\}^{n_\ell}$$

- Output :

$$y = \begin{bmatrix} y_c \\ y_\ell \end{bmatrix}, \; y_c \in \mathbb{R}^{p_c}, \; y_\ell \in \{0,1\}^{p_\ell}$$

- Input :

$$u = \begin{bmatrix} u_c \\ u_\ell \end{bmatrix}, \; u_c \in \mathbb{R}^{m_c}, \; u_\ell \in \{0,1\}^{m_\ell}$$

- Auxiliary binary variables : $\delta \in \{0,1\}^{r_\ell}$
- Auxiliary continuous variables : $z \in \mathbb{R}^{r_c}$

Assumptions

Recall:

$$\begin{aligned} x(t+1) &= Ax(t) + B_1u(t) + B_2\delta(t) + B_3z(t) \\ y(t) &= Cx(t) + D_1u(t) + D_2\delta(t) + D_3z(t) \\ E_2\delta(t) &+ E_3z(t) \le E_1u(t) + E_4x(t) + E_5 \end{aligned}$$

- Well-Posedness :

$$\{x(t),\ u(t)\} \rightarrow \{x(t+1)\} \text{ single valued}$$

$$\{x(t),\ u(t)\} \rightarrow \{y(t)\} \quad \text{ single valued}$$

This allows to define trajectories in x-space

- Physical Constraints :

$$\mathcal{C} \triangleq \left\{ \begin{bmatrix} x_c \\ u_c \end{bmatrix} \in \mathbb{R}^{n_c+m_c} :\ Fx_c + Gu_c \le H \right\}$$

$\mathcal{C}$ is a bounded polyhedral set (not restrictive in practice)

This allows to define upper and lower bounds M, m

Note: Physical constraints are included in the inequality

Propositional Calculus and Linear Integer Programming

Truth table

X_1	X_2	$\sim X_1$	$X_1 \vee X_2$	$X_1 \wedge X_2$	$X_1 \rightarrow X_2$	$X_1 \leftrightarrow X_2$
F	F	T	F	F	T	T
F	T	T	T	F	T	F
T	F	F	T	F	F	F
T	T	F	T	T	T	T

Propositional logic $\Leftrightarrow$ Integer linear inequalities

$X \in \{F, T\}$ $\Leftrightarrow$ $\delta \in \{ 0, 1 \}$

$X_1 \vee X_2$ is equivalent to $\delta_1 + \delta_2 \geq 1$

$X_1 \wedge X_2$ is equivalent to $\delta_1 + \delta_2 \geq 2$

$\sim X_1$ is equivalent to $\delta_1 \leq 0$

$X_1 \rightarrow X_2$ is equivalent to $\delta_1 - \delta_2 \leq 0$

$X_1 \leftrightarrow X_2$ is equivalent to $\delta_1 - \delta_2 = 0$

Logic Facts Involving Continuous Variables

$[f(x) \leq 0] \;\wedge\; X$	$f(x) - \delta \leq -1 + m(1-\delta)$
$[f(x) \leq 0] \;\vee\; X$	$f(x) \leq M\delta$
$\sim [f(x) \leq 0]$	$f(x) \geq \epsilon$
$[f(x) \leq 0] \;\rightarrow\; X$	$-f(x) + \epsilon \leq (\epsilon - m)\delta$
$[f(x) \leq 0] \;\leftrightarrow\; X$	$\begin{cases} f(x) \leq M(1-\delta) \\ f(x) \geq \epsilon + (m-\epsilon)\delta \end{cases}$

- M, m = upper-, lower-bounds on $f(x)$
- ϵ = small tolerance (e.g. machine precision)
- $X \in \{$F, T$\}$, $x \in \mathbb{R}^n$, $\delta \in \{0,1\}$

if $f(x)$ linear $\Rightarrow$ Mixed-Integer Linear Inequalities

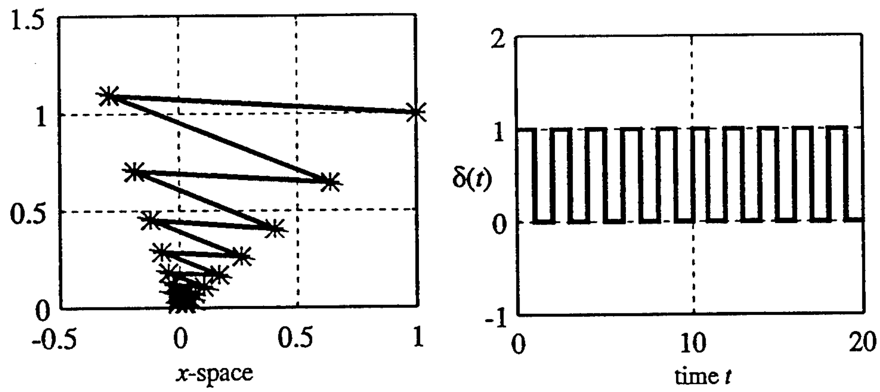
Product of variables

$\delta_3 = \delta_1\delta_2$	$\begin{cases} -\delta_1 + \delta_3 \leq 0 \\ -\delta_2 + \delta_3 \leq 0 \\ \delta_1 + \delta_2 - \delta_3 \leq 1 \end{cases}$
$y = \delta f(x)$	$\begin{cases} y \leq M\delta \\ y \geq m\delta \\ y \leq f(x) - m(1-\delta) \\ y \geq f(x) - M(1-\delta) \end{cases}$

- M, m = upper-, lower-bounds on $f(x)$
- ϵ = small tolerance (e.g. machine precision)

if $f(x)$ linear $\Rightarrow$ Mixed-Integer Linear Inequalities

An Example



- System:

$$\begin{cases} x(t+1) &= 0.8 \begin{bmatrix} \cos\alpha(t) & -\sin\alpha(t) \\ \sin\alpha(t) & \cos\alpha(t) \end{bmatrix} x(t) + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u(t) \\ y(t) &= [1\ 0]x(t) \\ \alpha(t) &= \begin{cases} \frac{\pi}{3} & \text{if} \quad [1\ 0]x(t) \geq 0 \\ -\frac{\pi}{3} & \text{if} \quad [1\ 0]x(t) < 0 \end{cases} \end{cases}$$

- Rewrite as

$$x(t+1) = [I \quad I]\, z(t)$$

$$\begin{bmatrix} 10 \\ -10-\epsilon \\ -M \\ -M \\ M \\ M \\ M \\ M \\ -M \\ -M \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \delta(t) + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ I & 0 \\ -I & 0 \\ 0 & I \\ 0 & -I \\ I & 0 \\ -I & 0 \\ 0 & I \\ 0 & -I \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} z(t) \leq \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ [0\ 1]' \\ -[0\ 1]' \\ [0\ 1]' \\ -[0\ 1]' \\ 0 \\ 0 \\ 1 \\ -1 \end{bmatrix} u(t) + \begin{bmatrix} 1 & 0 \\ -1 & 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ A_1 \\ -A_1 \\ A_2 \\ -A_2 \\ I \\ -I \\ 0 \\ 0 \end{bmatrix} x(t) + \begin{bmatrix} 10 \\ -\epsilon \\ 0 \\ 0 \\ M \\ M \\ M \\ M \\ 0 \\ 0 \\ N \\ N \\ 1 \\ 1 \end{bmatrix}$$

- The origin is globally asymptotically stable

System Theory for MLD Systems

- Well-Posedness
- Stability
- Reachability/Controllability
- Observability/Reconstructibility
- Optimal Control
- State Estimation

Observability Test for Autonomous MLD Systems

- DEFINITION of *incremental observability*:

  $\forall x_1,\ x_2 \in \mathcal{X}(0)$

$$\sum_{t=0}^{T-1} \|y(t,x_1) - y(t,x_2)\|_\infty \geq w\|x_1 - x_2\|_1$$

- GOAL : compute minimum number of output measurements necessary to distinguish two initial states $x_1, x_2 \in \mathcal{X}(0)$

- TEST :

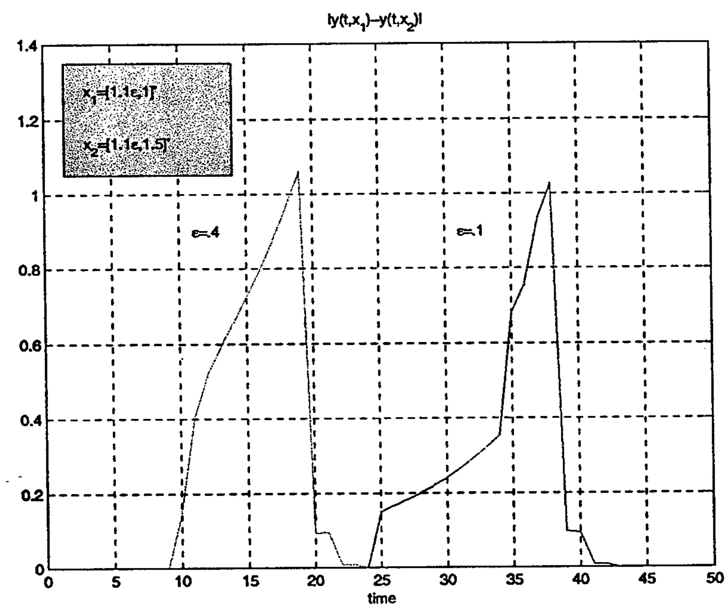
  0. Fix w_{min} as small as desired
     ($w < w_{\text{min}}$=practical indistinguishability)
  1. Fix T_{max}=maximum number of measurements
     ($T > T_{\text{max}}$=practical indistinguishability)
  2. $T \leftarrow 1$
  3. Minimize $\sum_{t=0}^{T-1} \|y(t,x_1)-y(t,x_2)\|_\infty - w_{\text{min}}\|x_1-x_2\|_1$
  4. If minimum < 0, increase T.
  5. If $T > T_{\text{max}}$, STOP (system unobservable)
  6. Go to 3

$\Rightarrow$ Mixed-Integer Linear Program

Observability Index: Example

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}(t+1) = \begin{cases} \begin{bmatrix} 1.1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}(t) & \text{if } \epsilon \le x_1(t) < 1, \ \epsilon > 0 \\ \begin{bmatrix} 0 & 0.9 \\ 0.9 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}(t) & \text{otherwise} \end{cases}$$

$$y(t) = x_1(t), \qquad \epsilon < x_1(0) < 1$$



$$T \triangleq \left\lceil \frac{\log \frac{1}{\epsilon}}{\log 1.1} \right\rceil + 1$$

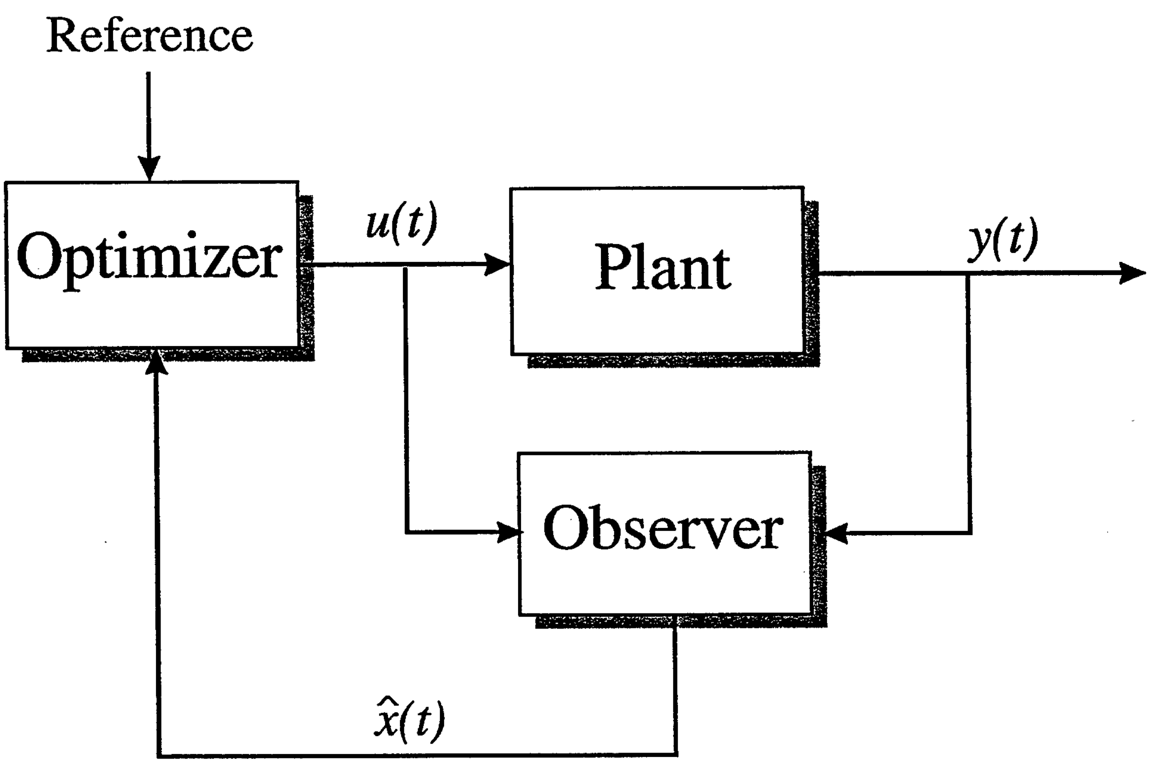
ϵ	T
0.9	3
0.4	11
0.1	26
0.01	50
0.001	74
0	∞

- Observability index T does not depend on state dimension n
- T can be arbitrarily large (also $T = \infty$)

System Theory for MLD Systems

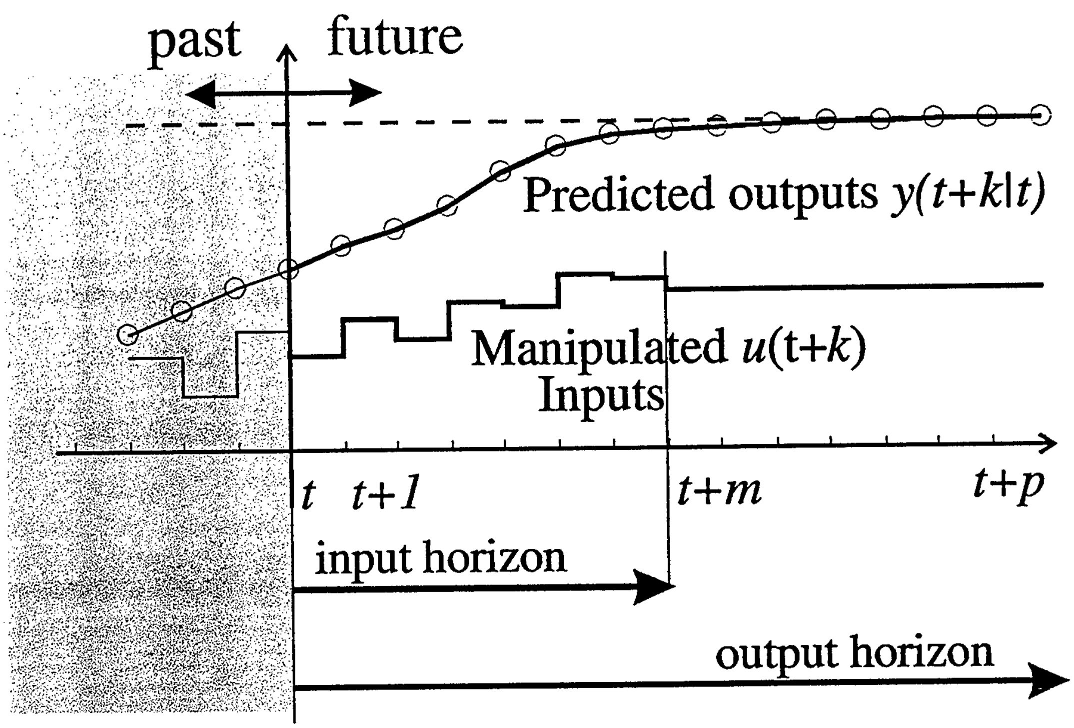
- Well-Posedness
- Stability
- Reachability/Controllability
- Observability/Reconstructibility
- Optimal Control
- State Estimation

Model Predictive Control



- MODEL : a model of the plant is needed to predict the future behaviour of the plant
- PREDICTIVE : optimization is based on the predicted future evolution of the plant
- CONTROL : control complex constrained multivariable plants

Optimization Problem

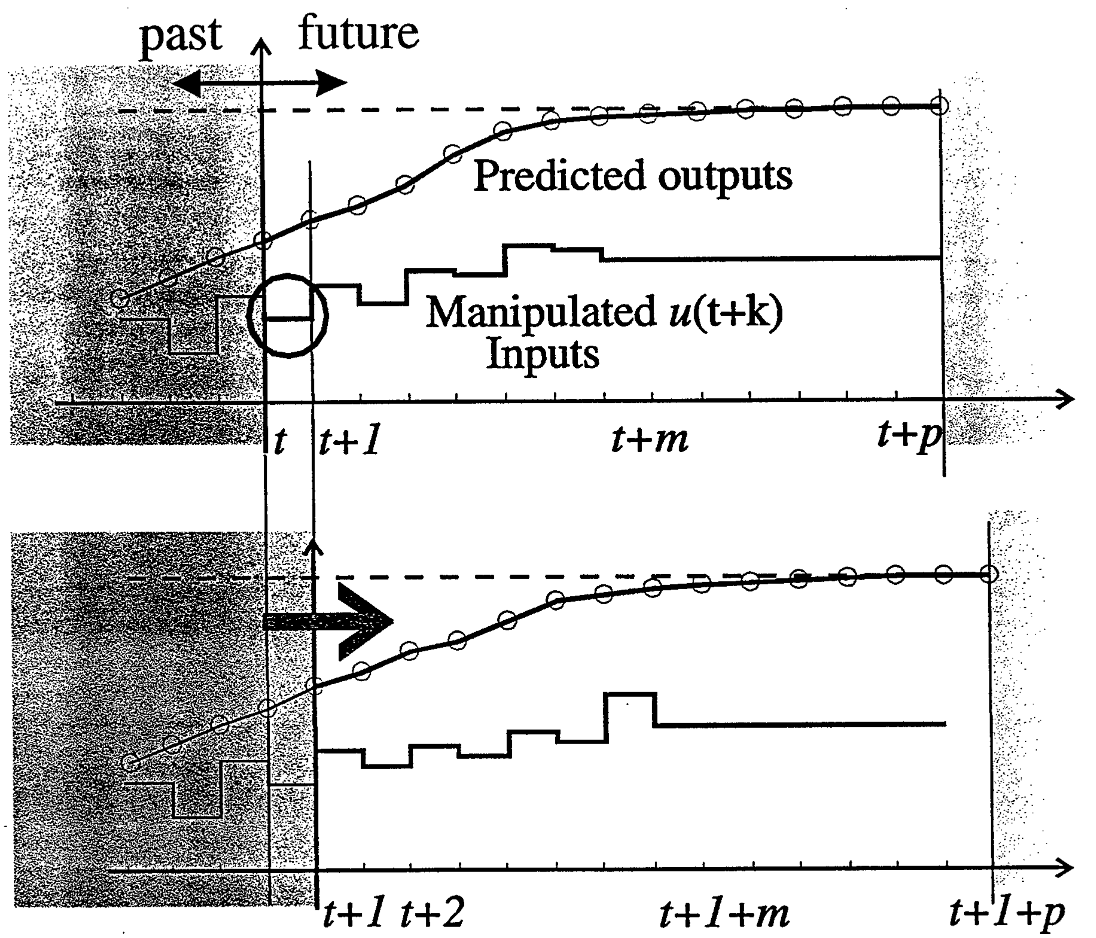


Compute the optimal sequence of manipulated inputs which minimizes

- tracking error = output - reference
- subject to constraints on inputs and outputs

On-line optimization ⇒ Receding Horizon

Receding Horizon



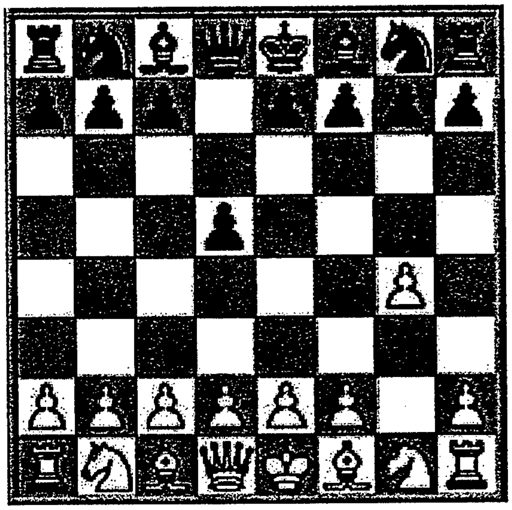
- Optimize at time t (new measurements)
- Only apply the first optimal move $u(t)$
- Repeat the whole optimization at time $t+1$

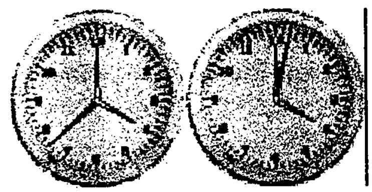
Advantage of on-line optimization:

FEEDBACK!

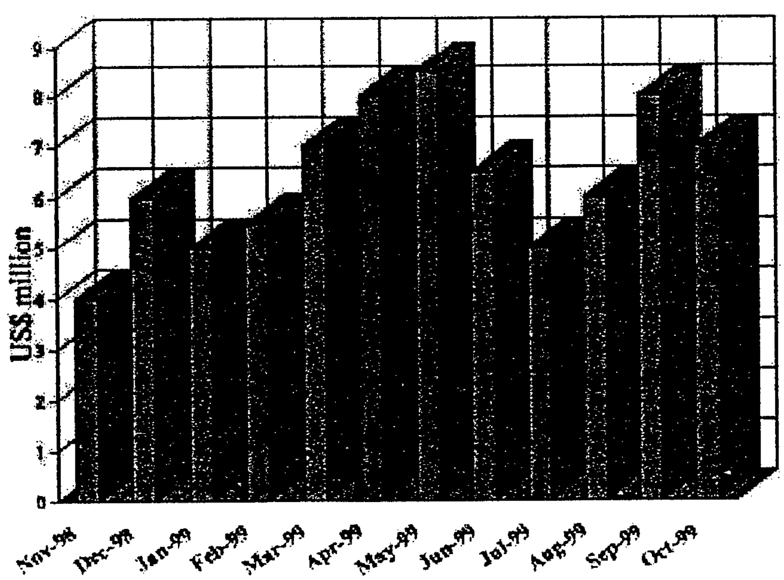
Receding Horizon - Examples

Chess Game





Investment plans



Predictive Controller

- On-line optimization problem (MIQP)

$$\min_{\{v_0^{T-1}\}} J(v_0^{T-1}, x(t)) \triangleq \sum_{k=0}^{T-1} \|v(k) - u_e\|_{Q_1}^2 + \|\delta(k|t) - \delta_e\|_{Q_2}^2$$

$$\|z(k|t) - z_e\|_{Q_3}^2 + \|x(k|t) - x_e\|_{Q_4}^2 + \|y(k|t) - y_e\|_{Q_5}^2$$

$$\text{subj. to} \begin{cases} x(T|t) = x_e \\ x(k+1|t) = Ax(k|t) + B_1 v(k) + \\ \qquad B_2\delta(k|t) + B_3 z(k|t) \\ y(k|t) = Cx(k|t) + D_1 v(k) + D_2\delta(k|t) + D_3 z(k|t) \\ E_2\delta(k|t) + E_3 z(k|t) \leq E_1 v(k) + E_4 x(k|t) + E_5 \end{cases}$$

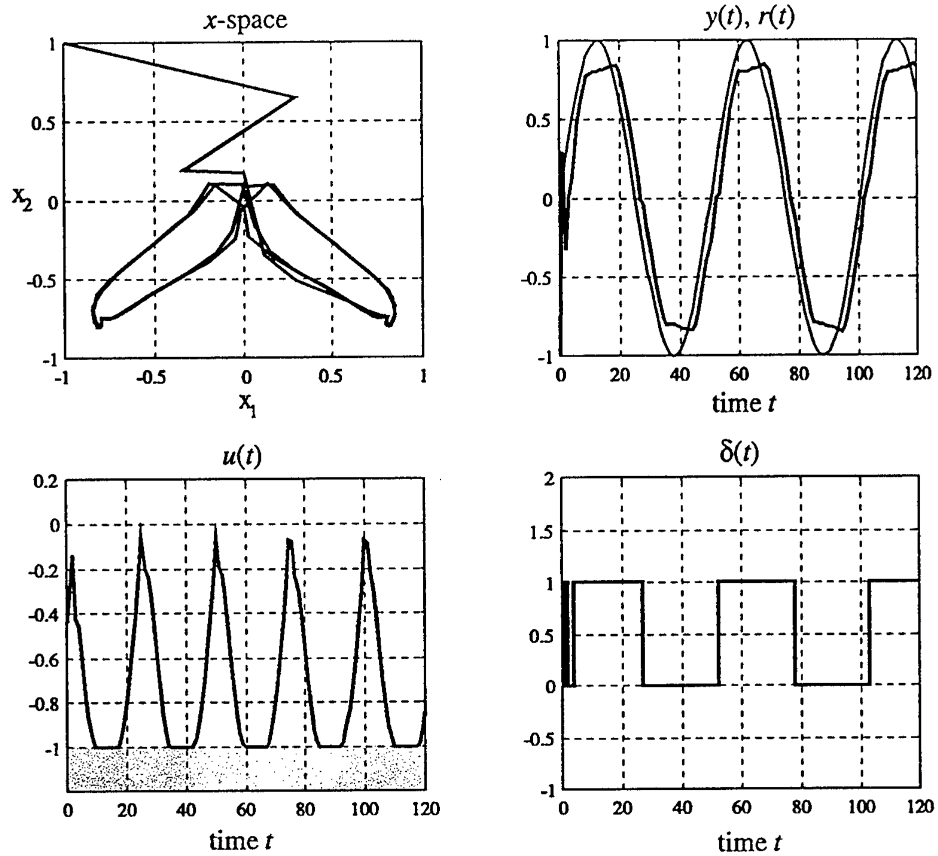
- According to a receding horizon philosophy, set

$$u(t) = v_t^*(0)$$

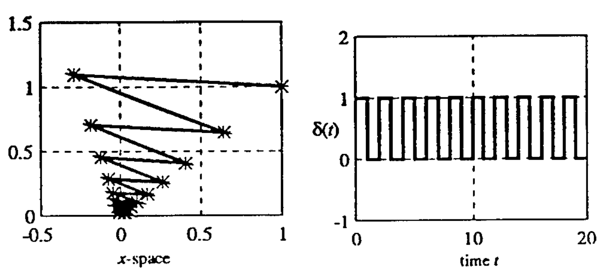
- Repeat everything at time $t+1$

- Feasibility of MIQP @$t = 0$ $\Rightarrow$ closed-loop stability
- Global optimum not needed !

Closed-Loop Example



$$
\begin{cases}
x(t+1) &= 0.8\begin{bmatrix}\cos\alpha(t) & -\sin\alpha(t)\\ \sin\alpha(t) & \cos\alpha(t)\end{bmatrix}x(t)+\begin{bmatrix}0\\1\end{bmatrix}u(t)\\
y(t) &= [1\ 0]x(t)\\
\alpha(t) &= \begin{cases}\frac{\pi}{3} & \text{if} \quad [1\ 0]x(t)\geq 0\\ -\frac{\pi}{3} & \text{if} \quad [1\ 0]x(t)<0\end{cases}
\end{cases}
$$



Computational Aspects

- The on-line optimization problem is a Mixed-Integer Quadratic Program (MIQP)
- No need to reach global optimum (see proof of the theorem)
- Available methods: Generalized Benders' Decomposition, Outer Approximation, LP/QP based branch and bound, and Branch and Bound.
- Branch & Bound is the most effective
- General purpose B&B MIQP solvers available, for both dense and sparse problems (e.g. Fletcher-Leyffer's)

System Theory for MLD Systems

- Well-Posedness
- Stability
- Reachability/Controllability
- Observability/Reconstructibility
- Optimal Control
- State Estimation

Fault Detection and State Estimation for Hybrid Systems

- Use Moving Horizon Estimation for MLD Systems (dual of MPC)
- Mixed logic dynamic fault (MLDF) form (Bemporad, Mignone, Morari, 1998):

$$
\begin{aligned}
x(t+1) =& Ax(t) + B_1u(t) + B_2\delta(t) + B_3z(t) \\
&+ B_6\phi(t) + \xi(t) \\
y(t) =& Cx(t) + D_1u(t) + D_2\delta(t) + D_3z(t) + \\
&+ D_6\phi(t) + \zeta(t) \\
E_2\delta(t) + E_3z(t) \leq& E_1u(t) + E_4x(t) + E_5 + E_6\phi(t)
\end{aligned}
$$

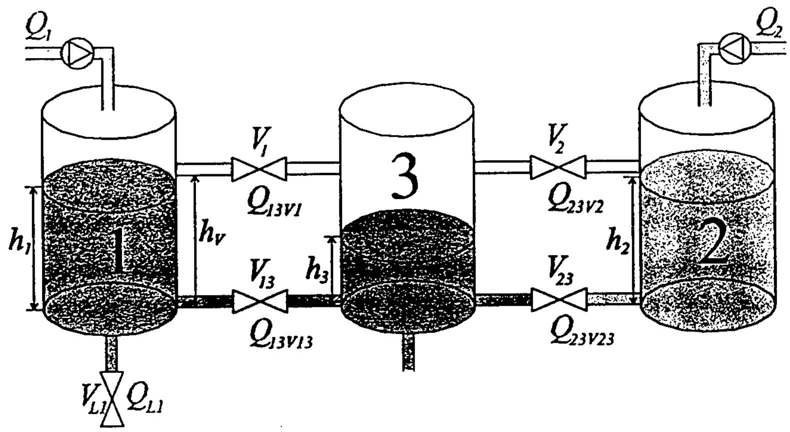
Faults: $\phi(t) \in \{0,1\}^f$

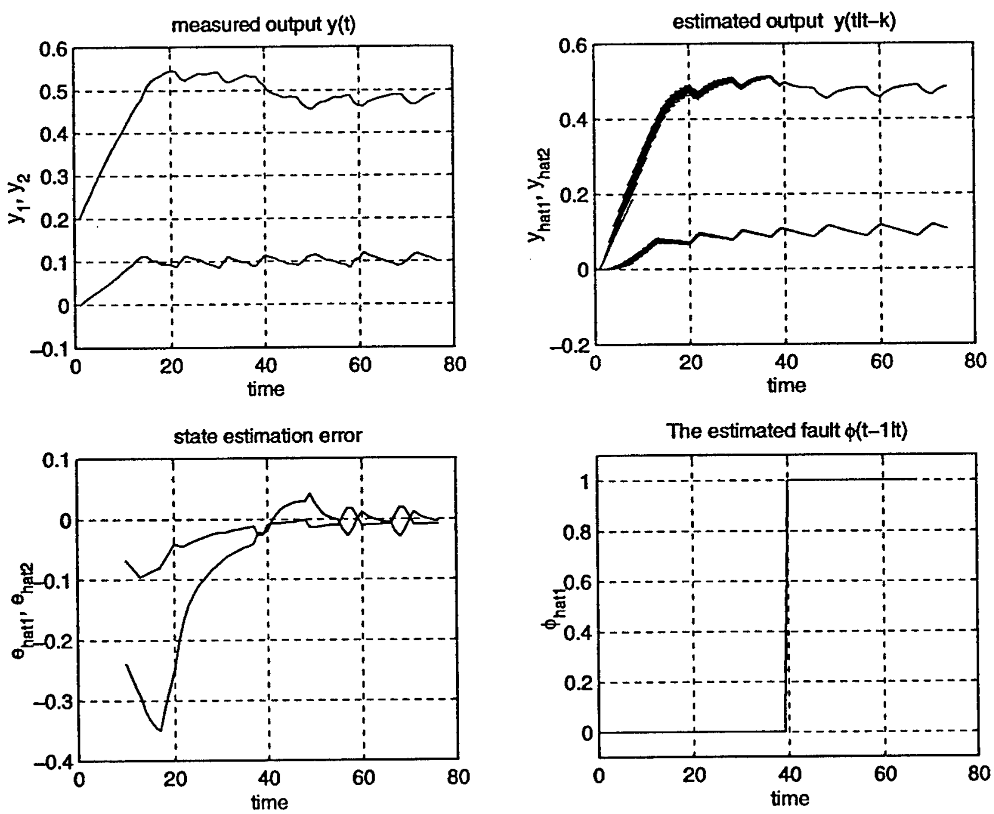
Disturbances: $\xi(t) \in R^n$, $\zeta(t) \in R^p$

- At each time t the estimates $\hat{\phi}(t)$, $\hat{x}(t)$ are obtained by minimizing an MIQP over a horizon extending backwards in time

$$
\sum_{k=t-T+1}^{0} ||\hat{y}(k|t) - y(k)||^2_{Q_y} + L\Big(\zeta(k|t), \hat{\phi}(k|t), \hat{x}(k|t), \xi(k|t), \hat{\delta}(k|t), \hat{z}(k|t)\Big)
$$

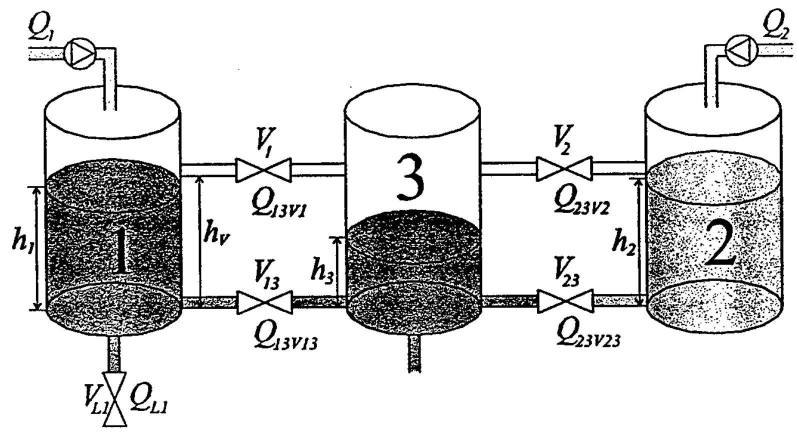
Three Tank System: Fault Detection - I

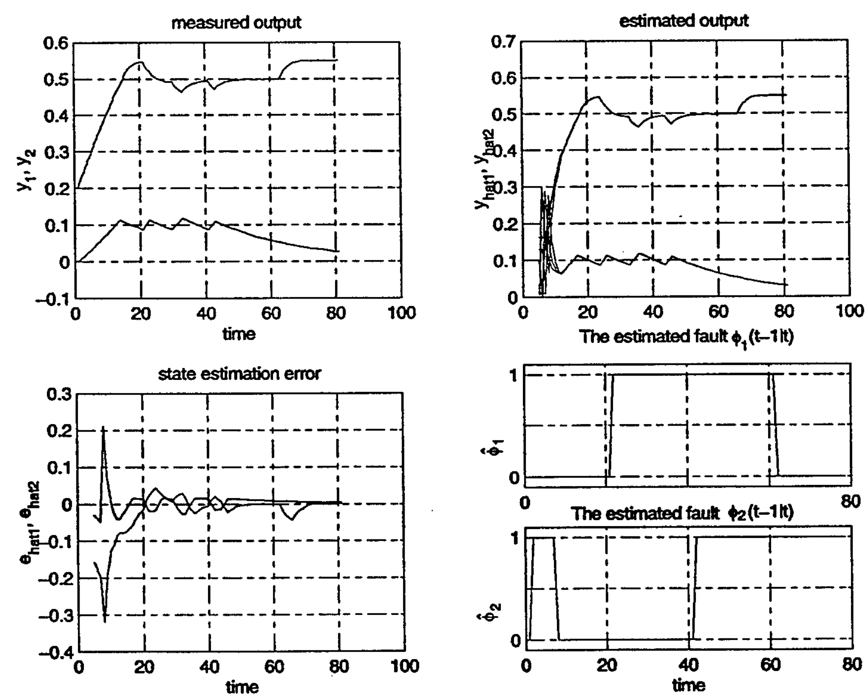




- ϕ_1: leak in tank 1 at time $t = 38$

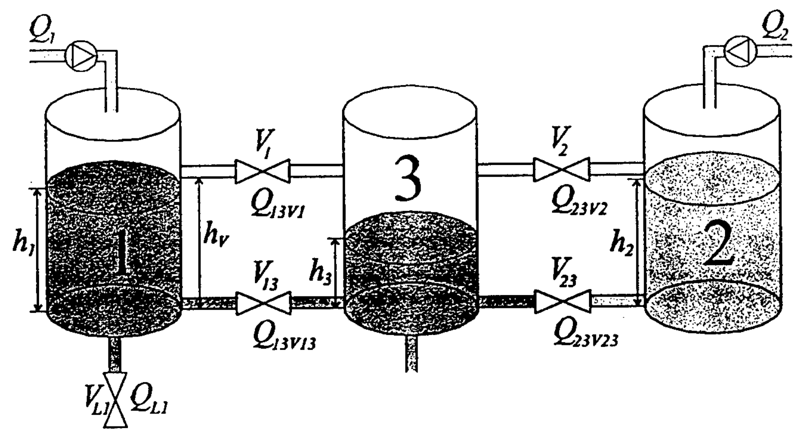
Three Tank System: Fault Detection - II

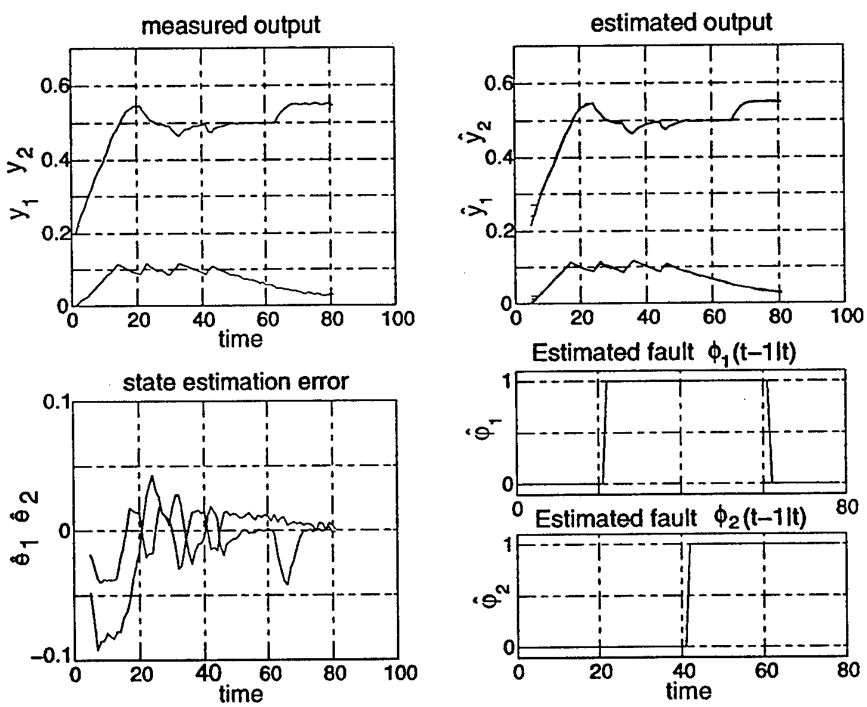




- ϕ_1: leak in tank 1 for $20s \leq t \leq 60s$
- ϕ_2: valve V_1 blocked for $t \geq 40s$

Three Tank System: Fault Detection - III





- ϕ_1 leak in tank 1 for $20s \leq t \leq 60s$
- ϕ_2 valve V_1 blocked for $t \geq 40s$
- $+$ logic constraint $[h_1 \leq h_v] \Rightarrow \phi_2 = 0$

Research Program

- Modeling Language and Compiler
  - Express models and specs with "standard" vocabulary
  - Translate into MILP/MIQP
- Verification/Controllability/Reachability
  - Use tools from Polyhedral Computation
  - New case studies
- State Estimation/Fault detection
  - Dual of Model Predictive Control
  - Convergence properties of estimator
- Optimization Algorithms
  - Sparse QP solver
  - New Branch-and-Bound strategies
- Model Reduction/Approximation

HYSDEL

(HYbrid Systems DEscription Language)

- Describe Hybrid Systems in a compact way
  - Propositional Logic
  - Dynamics
  - Constraints
- Automatically generate MLD models
- MLD model is not unique in terms of the number of auxiliary variables $\Rightarrow$ optimize model: # binary variables = min.!
- Reduction of model complexity for models containing purely logic relations (Truth Tables $\Rightarrow$ Convex Hull)

HYSDEL Example

Automotive Active Leveler

```
% Description of variables and constants

state f,h,xl1,xl2;
input d,dc,dev;

const OTh, OTl, ITh, ITl;
const M1,M2,M3,M4,m1,m2,m3,m4;
const e;
const Ts,cbar,evbar,eats,cmax,cmin,evmax,evmin;

% Variable types

real f,h,z1,z2,z3,d,dc,dev;
logic d1,d2,d3,d4,d5,d6,d7,d8,d9,d10,d11,d12,d13,d14;


% Relations

d1 = {f-ITh <= 0, M1, m1, e};
d2 = {f-ITl <= 0, M2, m2, e};
d3 = {f-OTh >= 0, M3, m3, e};
d4 = {f-OTl >= 0, M4, m4, e};

d5 = ~xl1 & d3;   % Should be accepted also: d5=(1-xl1)&d3, d5=(1-xl1)*d3
d6 = xl1 & ~d1;
d7 = xl2 & d2;
d8 = ~xl2 & ~d4;
d9 = d5 | d6;
d10 = d7 | d8;
d11 = ~xl1 & xl2;
d12 = xl1 & ~xl2;
d13 = ~(~d9 & ~d10 & d14);
d14 = xl1 | xl2;

z1 = d11 * (cbar + dc) {cmax,cmin,e};
z2 = d12 * (evbar + dev) {evmax,evmin,e};
z3 = (eats * f + (1 - eats) * h)*d13 {10*OTh,10*OTl,e};

% Other constraints

must xl1 + xl2 <= 1;
must ~(d9 & d10);
must ~(d11 & d12);

% Update


update f = z3;
update h = h + Ts * (d + z1 + z2);
update xl1 = d9;
update xl2 = d10;
```

General Transformation of Truth Tables into Linear Integer Inequalities

T:

x_1	x_2	$\dots$	x_{n-1}	$x_n = F(x_1, \dots, x_{n-1})$
0	0		0	1
0	0		1	0
$\vdots$	$\vdots$		$\vdots$	
1	1		1	1

$$\Downarrow$$

$$A\delta \leq B, \ \ \delta \in \{0,1\}^n$$

Theorem:

The polytope $P \triangleq \{\delta : A\delta \leq B\}$ *is the convex hull of the rows of the truth table* T

$$P = \text{conv}(T)$$

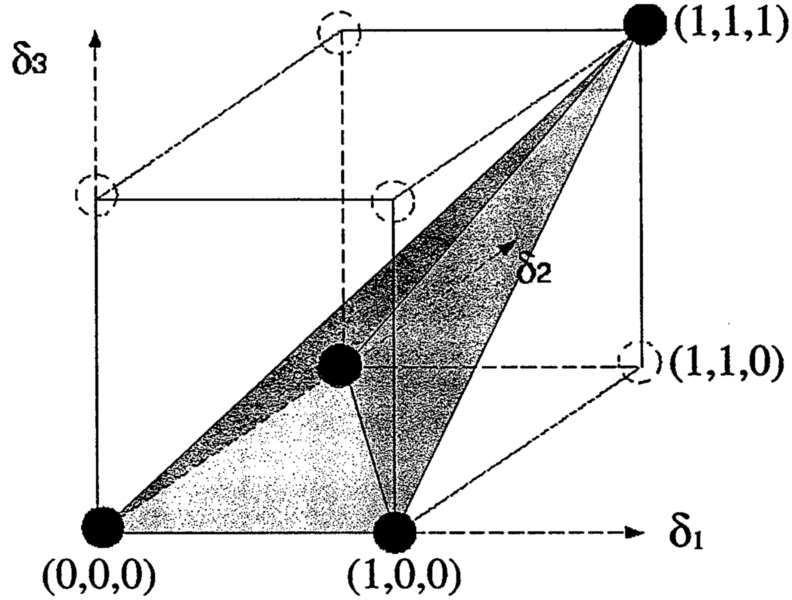
(Bemporad and Mignone) □

Every logic proposition can be translated into linear integer inequalities

Truth Tables ⇒ Linear Integer Inequalities

Example: "AND"

δ_1	δ_2	δ_3
0	0	0
0	1	0
1	0	0
1	1	1



$$\text{conv}\left(\begin{bmatrix}0\\0\\0\end{bmatrix},\begin{bmatrix}0\\1\\0\end{bmatrix},\begin{bmatrix}1\\0\\0\end{bmatrix},\begin{bmatrix}1\\1\\1\end{bmatrix}\right) = \left\{\delta : \begin{array}{c} -\delta_1+\delta_3 \le 0 \\ -\delta_2+\delta_3 \le 0 \\ \delta_1+\delta_2-\delta_3 \le 1 \end{array}\right\}$$

Algorithm to compute convex hull: `lrs`
(David Avis, McGill Univ.) `ftp://mutt.cs.mcgill.cs/pub/c`

Other algorithms: `qhull`, `chD`, `Hull`, `Porto`, `cdd`.

Research Program

- Modeling Language and Compiler
  - Express models and specs with "standard" vocabulary
  - Translate into MILP/MIQP
- Verification/Controllability/Reachability
  - Use tools from Polyhedral Computation
  - New case studies
- State Estimation/Fault detection
  - Dual of Model Predictive Control
  - Convergence properties of estimator
- Optimization Algorithms
  - Sparse QP solver
  - New Branch-and-Bound strategies
- Model Reduction/Approximation

Formal Verification of Hybrid Systems

- MLD Model:

$$
\begin{aligned}
x(t+1) &= Ax(t) + B_1 w(t) + B_2 \delta(t) + B_3 z(t) \\
y(t) &= Cx(t) + D_1 w(t) + D_2 \delta(t) + D_3 z(t) \\
E_2 \delta(t) + E_3 z(t) &\leq E_1 w(t) + E_4 x(t) + E_5
\end{aligned}
$$

- *Verification Problem VP #1*: Find max range for $y(t)$ $\forall t \geq 0$, $w(t) \in W$, $x(0) \in \mathcal{X}(0)$

- *Verification Problem VP #2*: $\forall w \in \mathcal{W}$ and $\forall x(0) \in \mathcal{X}(0)$ verify that $x(t) \in \mathcal{X}_s$ (set of *safe* states)

- Simple solution (VP #1): Solve $\forall T \geq 0$

$$
\max_{x(0),\{w(t),\delta(t),z(t)\}_{t=0}^{T}} y(T) = Cx(T) + D_1 w(T) + D_2 \delta(T) + D_3 z(T)
$$

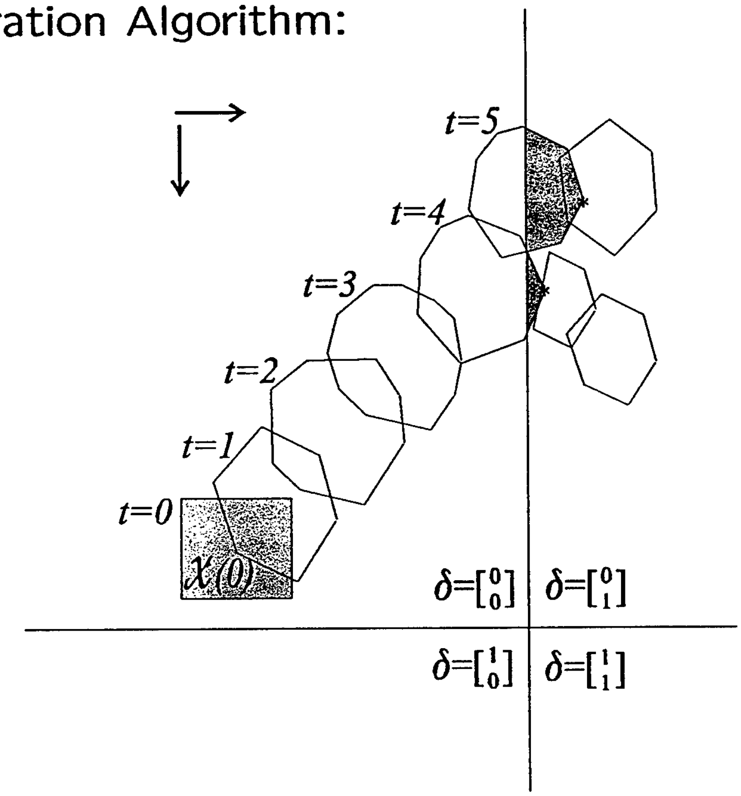
$$
\text{subj. to} \begin{cases}
x(0) \in \mathcal{X}(0) \\
w(t) \in \mathcal{W}, \ 0 \leq t \leq T \\
x(t+1) = Ax(t) + B_1 w(t) + B_2 \delta(t) + B_3 z(t) \\
E_2 \delta(t) + E_3 z(t) \leq E_1 u(t) + E_4 x(t) + E_5
\end{cases}
$$

- Similarly for VP #2: feasibility test with $x(T) \notin \mathcal{X}_s$

- IMPRACTICAL ! (even for polyhedral sets $\mathcal{W}$, $\mathcal{X}_s$, $\mathcal{X}(0)$, because of integer constraints)

Verification Algorithm

- RMK: when $\delta(t) \equiv$ const , system behaves linearly.
- IDEA: hypothetically partition $\mathbb{R}^{n_c} = \bigcup_{i=1}^{N} \mathcal{C}_i$ $(N \leq 2^{r_\ell})$ where $x_c(t) \in \mathcal{C}_i \;\Leftrightarrow\; \delta(t) = \delta_i$
- Exploration Algorithm:



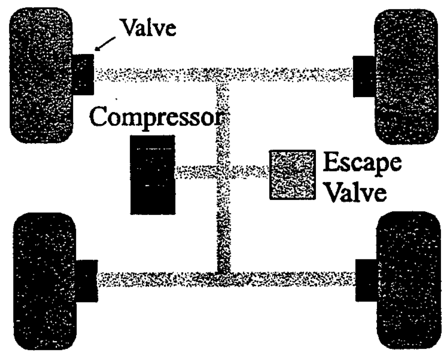
$$\max_{\{w(k),z(k)\}_{k=0}^{t},x_c(0),\delta(t)} \begin{bmatrix} x(t) \\ -x(t) \end{bmatrix}$$

$$\text{subj. to} \begin{cases} x(k+1) = Ax(k) + B_1 w(k) + B_2 \delta(k) + B_3 z(k) \\ E_2 \delta(k) + E_3 z(k) \leq E_1 w(k) + E_4 x(k) + E_5 \\ w(k) \in W, \; k = 0, \ldots, t \\ \delta(k) = \delta_i, \; k = 0, \ldots, t-1, \; \delta(t) \in \{0,1\}^{r_\ell} \\ x(0) \in \mathcal{X}(0) \end{cases}$$

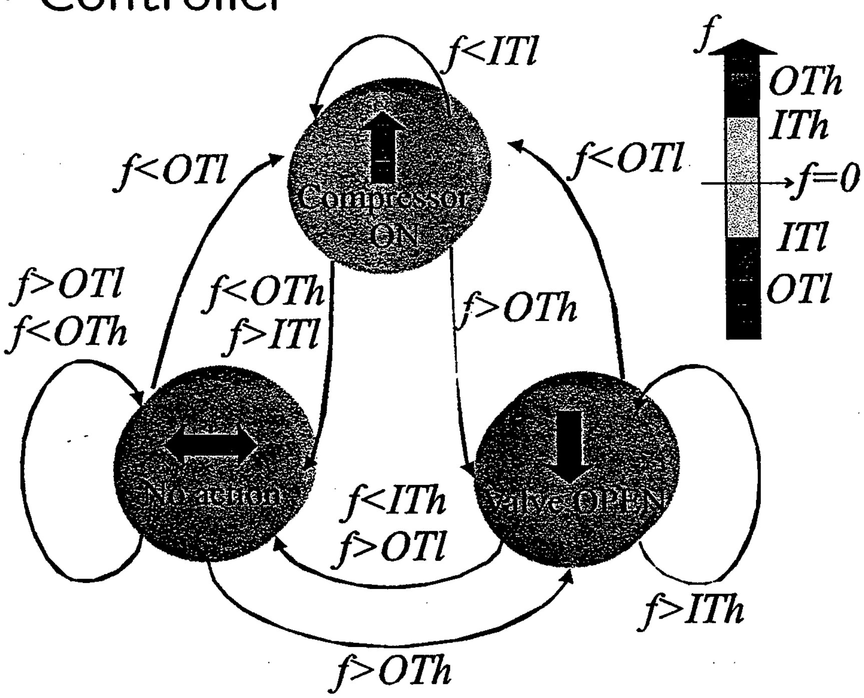
- Reachable set implicitly defined by linear inequalities (Hp: $\mathcal{W}$, $\mathcal{X}(0)$, $\mathcal{X}_s$ polytopes)

Example: Verification of an Automotive Active Leveler

- System



- Controller



$$f = \frac{1}{1+as} h$$

- For a given set of initial conditions, variation of parameters, and road conditions:

(VP#1) Find worst performance

(VP#2) Verify that height is within desired limits

Verification of an Automotive Active Leveler - II

- System can be transformed into MLD form

Continuous states	2
Logic states	2
Disturbance inputs	3
Boolean auxiliary variables	14
Continuous auxiliary variables	3

- Results and computational times:

$-44.54 \leq h(t) \leq 25.00$	4 m 25 m	Pentium II 400 MHz SPARCstation 20 (M-code)

  - No use of fast iterative reach-set projection
  - Use of rectangular approximation of initial sets

- Using HyTech (Stauner *et al.*, 1997):

$-47 \leq h(t) \leq 27$	62 m	SPARCstation 20

- Analytical solution (Elia and Brandin, 1998):

$-43 \leq h(t) \leq 23$	— — —	by-hand

Note: exact limits in discrete-time $\neq$ continuous-time

Research Program

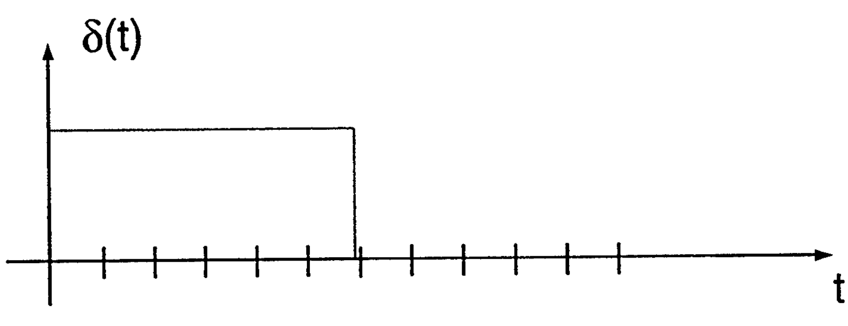
- Modeling Language and Compiler
  - Express models and specs with "standard" vocabulary
  - Translate into MILP/MIQP
- Verification/Controllability/Reachability
  - Use tools from Polyhedral Computation
  - New case studies
- State Estimation/Fault detection
  - Dual of Model Predictive Control
  - Convergence properties of estimator
- Optimization Algorithms
  - Sparse QP solver
  - New Branch-and-Bound strategies
- Model Reduction/Approximation

Branch & Bound for Optimal Control (and Fault Detection) of Hybrid Systems

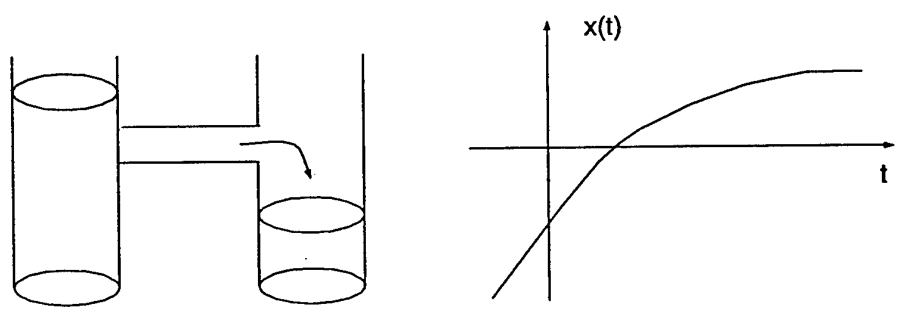
(A.Bemporad, D. Mignone, M.Morari)

Key Idea:

1. Optimal sequences $\{ \delta(0), \delta(1), \ldots, \delta(T) \}$ have FEW switches

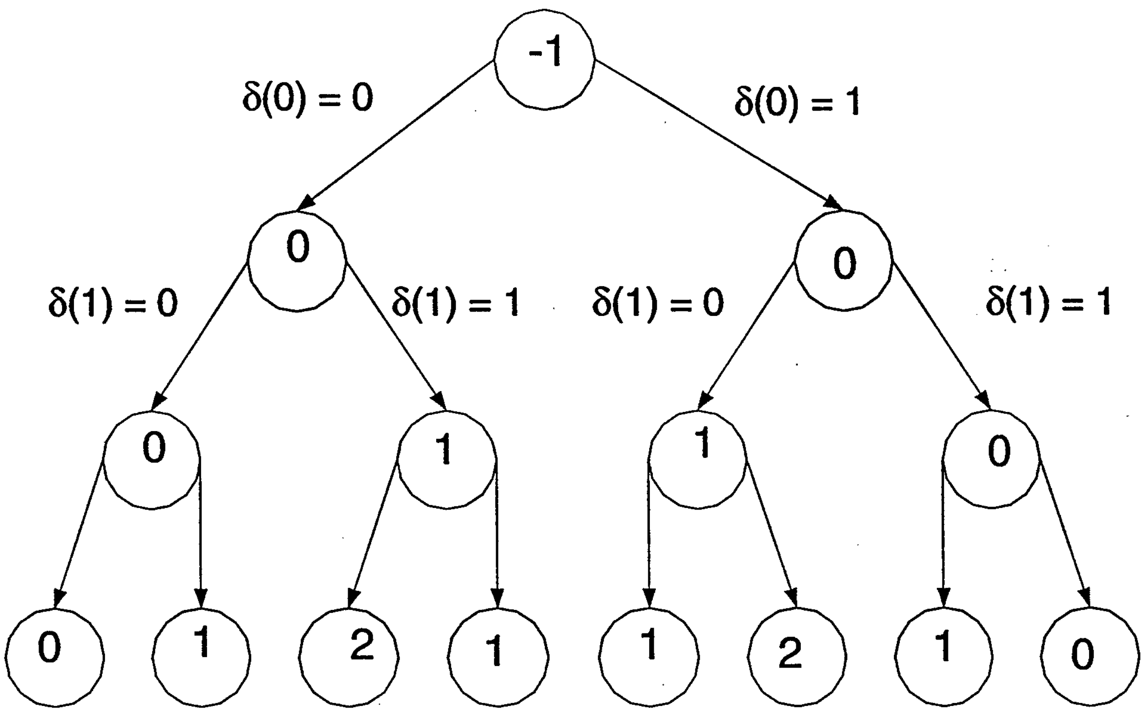


example: $[\delta(t) = 1] \leftrightarrow [x(t) \geq 0]$



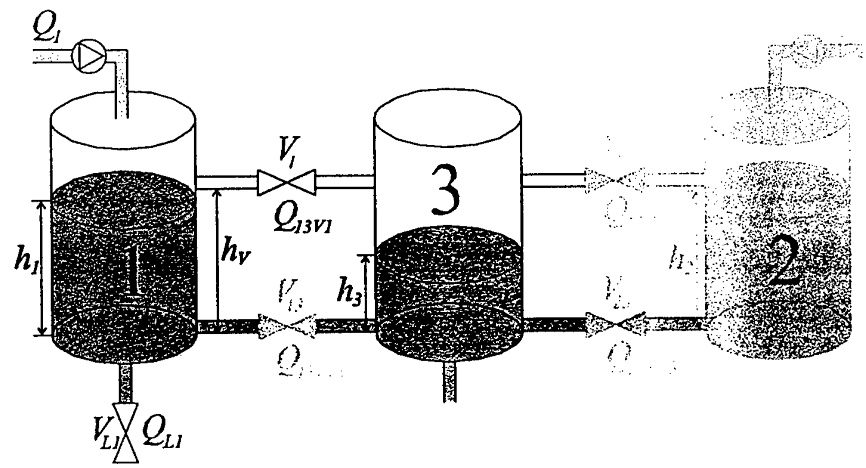
2. Order the sequences by number of switches: 0000, 1111; 0001, 0011; 0010, 0110; ...

Branch & Bound for Hybrid Systems

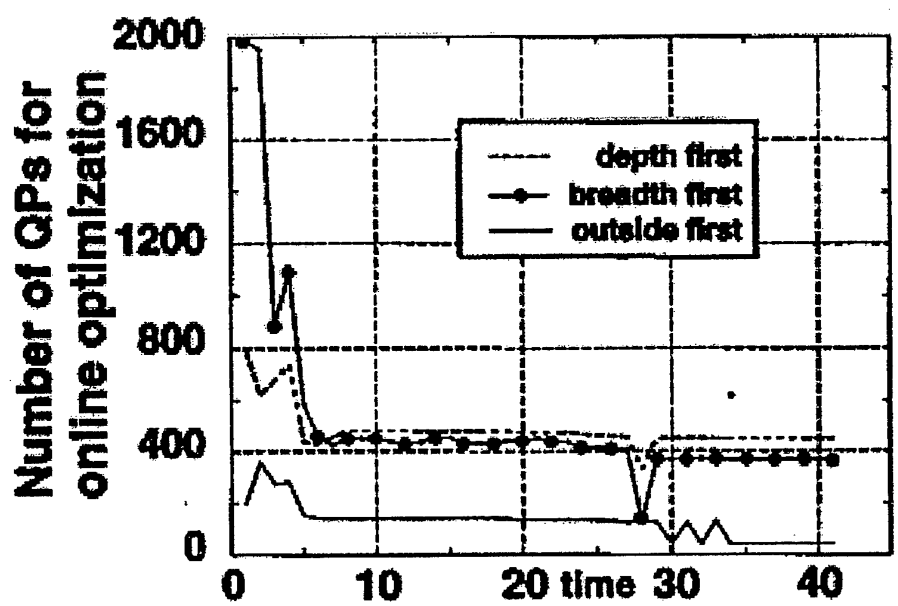


- First explore combinations of δ with low number of switches ("outside first" strategy)
- Assumption: global optimum is reached early $\Rightarrow$ large subtrees are fathomed.
- If not enough available computation time look only for sequences having a number of switches $\leq K_{max}$

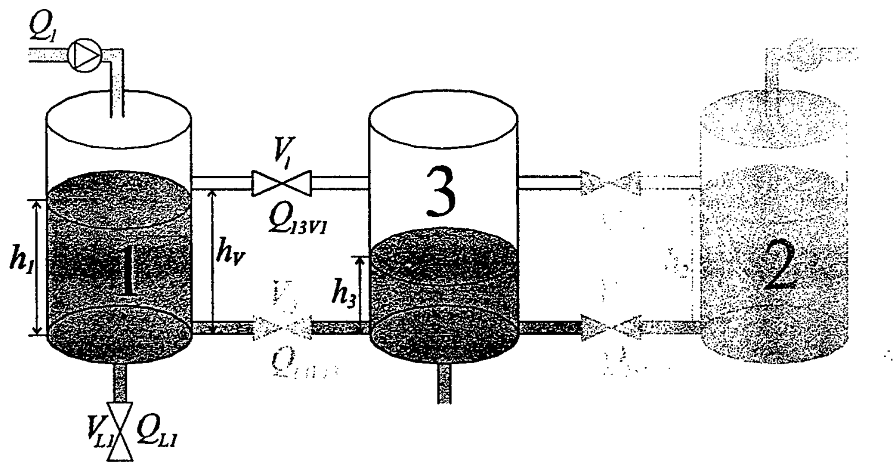
Three Tank System: Control -I



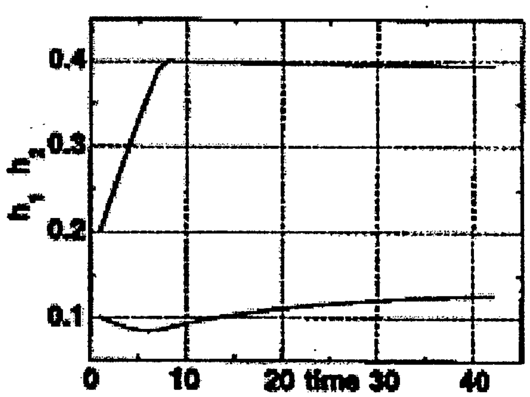
Number of QPs in receding horizon control using different tree exploring startegies



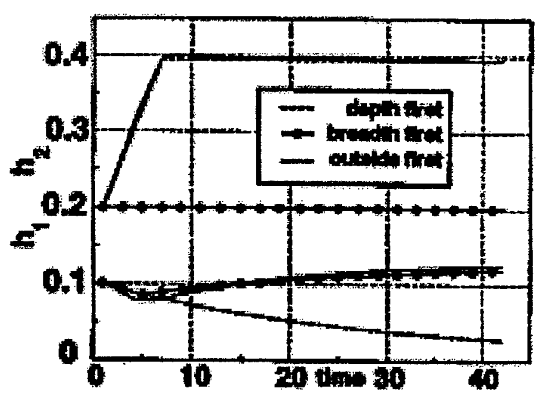
Three Tank System: Control -II



Trajectories using suboptimal MIQP solutions:
Limiting the number of QPs at each time step



global optimum



limiting # QPs

Premise

- The discussed problems are important but inherently difficult
- All useful techniques must involve significant off-line and/or on-line computation

Results

- New System Type:
  Mixed Logical Dynamical (MLD) System
- Many practical problems can be represented in MLD form
- Control, estimation, and verification require solution of Mixed-Integer Linear (or Quadratic) Programs (MILP, MIQP) for which efficient techniques are becoming available

ROLLOUT ALGORITHMS: PERFORMANCE ANALYSIS

Dimitri Bertsekas
Dept. of Electrical Engineering
and Computer Science
M.I.T.

(Joint Work w/ David Castanon and John Tsitsiklis)

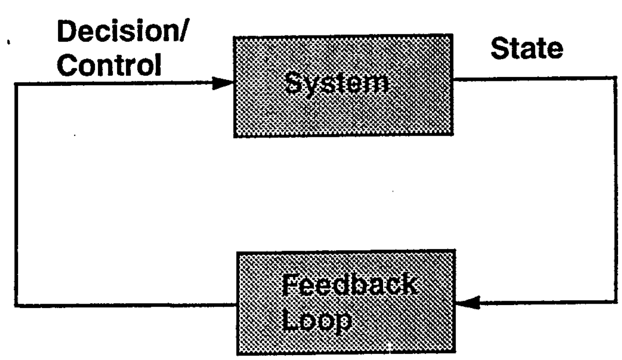
June 1999

OUTLINE

- Neuro-Dynamic Programming
- Rollout policies
- Use of rollout policies in deterministic combinatorial optimization
- The breakthrough problem
- Average performance analysis

DYNAMIC PROGRAMMING / DECISION AND CONTROL

- Main ingredients:
  - State evolving over time (e.g., a Markov chain)
  - Decision/control applied at each time
  - Reward is obtained at each time period
  - There may be noise & model uncertainty
  - There is state feedback used to determine the control



DP and NDP

- DP: Optimal decision at the current state maximizes the expected value of

  Current stage reward + Future stages reward starting from the next state (using opt. policy)

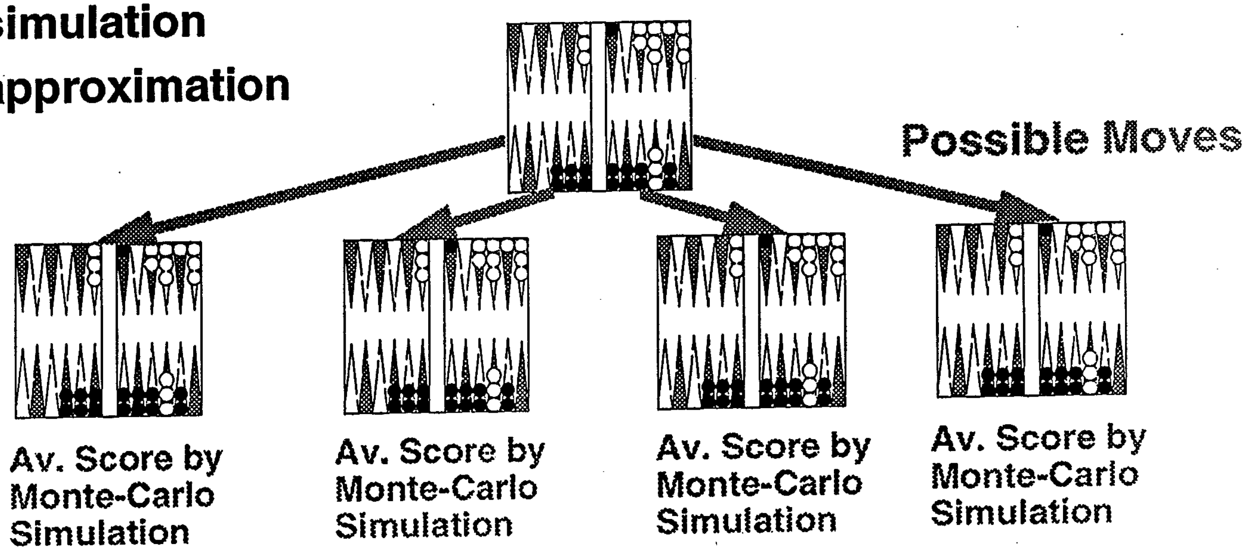
- NDP: Instead select decision that maximizes expected value of

  Current stage reward + Approximate future stages reward starting from the next state

- Approximate future reward function chosen from a parametric class with few parameters

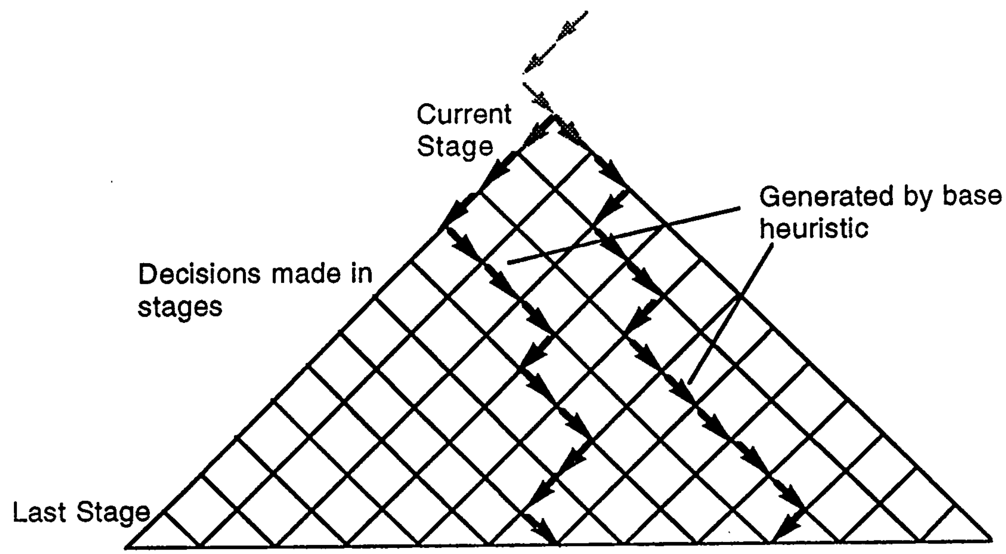
ROLLOUT POLICIES (Tesauro 1996)

- Use as approximate reward the reward of some suboptimal base policy (one-step policy iteration)
- Evaluation of base policy by
  - simulation
  - approximation



DETERMINISTIC PROBLEMS

- Use a heuristic as a base policy
- At each state, consider all possible next states, and run the heuristic (once) from each
- Select the next state with best heuristic reward



ROLLOUT POLICY PROPERTIES

- Forward looking (the heuristic runs forward)
- Self-correcting (the heuristic is reapplied at each time step)
- Suitable for on-line use, replanning
- Policy improvement result: Rollout policy improves on the base heuristic (Bertsekas, Tsitsiklis, Wu, J. Heuristics, 1997)
- Substantial positive experience, e.g., for scheduling problems
  - Bertsekas and Castanon
  - Donohue, Bertsekas, and Tsitsiklis

PERFORMANCE QUESTIONS

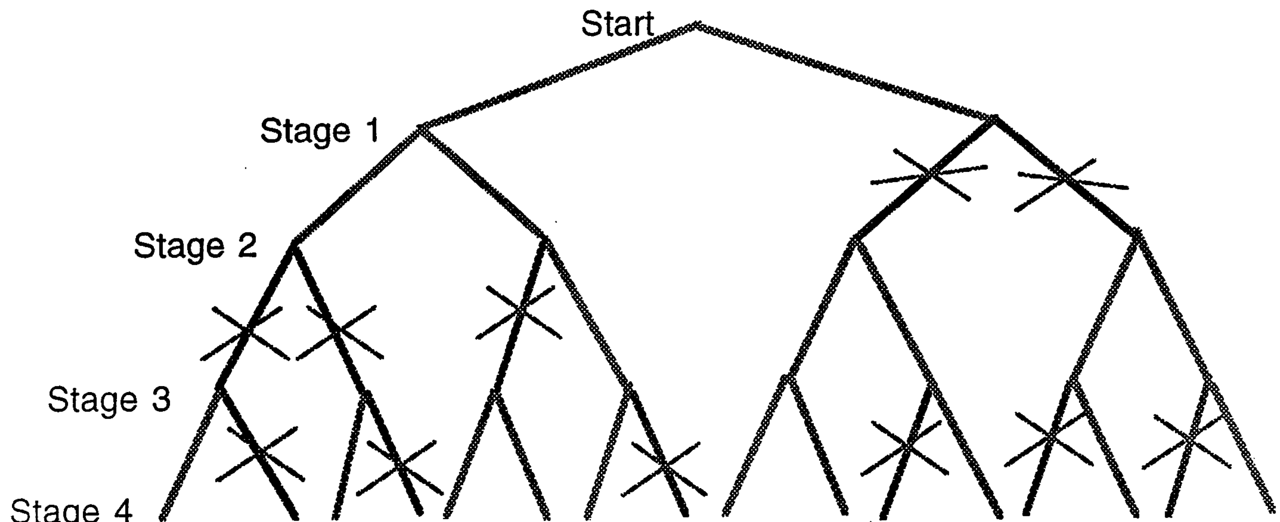
- How much better than the base heuristic?
- How close to optimal?
- Proper size of lookahead?

The art of doing mathematics is to find that special case which contains all the germs of generality

Hilbert

THE BREAKTHROUGH PROBLEM

- **Consider:**
  - **Binary tree with N stages**
  - **Each arc is FREE or BLOCKED**
- **Find a FREE path**



AVERAGE PERFORMANCE ANALYSIS

- Randomization over problem class: Each arc is FREE with prob. p, independently of others
- Calculate Probability(p,N) of finding a free path for:
  - The optimal DP algorithm [complexity $O(2^N)$]
  - The greedy algorithm: use the first available free arc [complexity $O(N)$]
  - The rollout algorithm using the greedy as base heuristic [complexity $O(N^2)$]
- For a rollout with limited lookahead, what is the impact of the size of lookahead?

OPTIMAL BREAKTHROUGH PROBABILITY

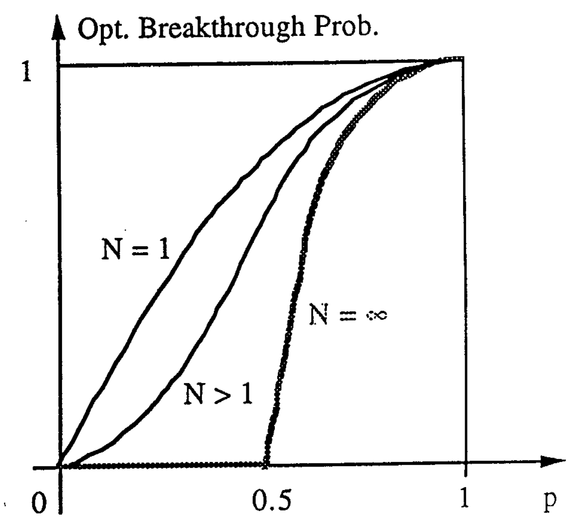
- **The optimal breakthrough probability, $B^*(p,N)$, for an N-stage problem is given by:**

$$B^*(p,N) = p\big(2 - pB^*(p,N-1)\big)B^*(p,N-1)$$

$$B(p,0) = 1$$

$$B^*(p,N) \to 0 \quad \textbf{if } p \leq 0.5$$

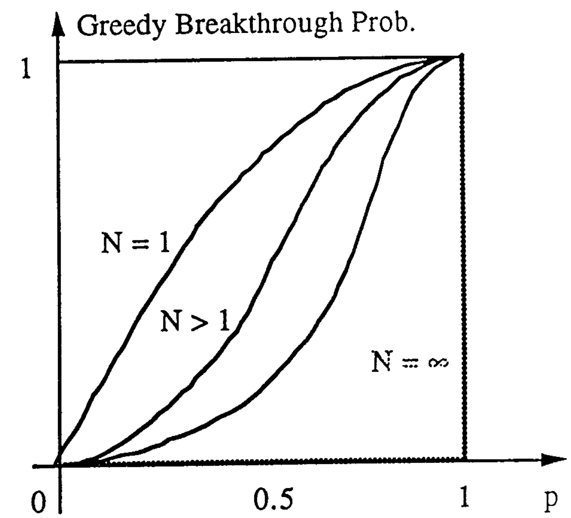
$$B^*(p,N) \to \tfrac{2p-1}{p^2} \quad \textbf{if } p > 0.5$$



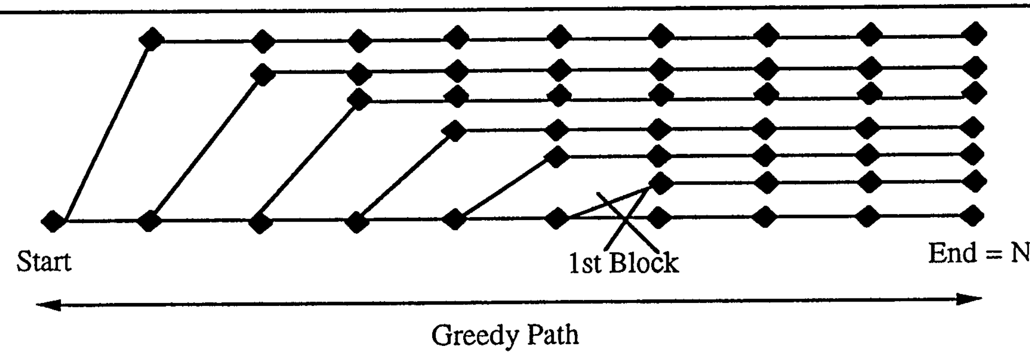
GREEDY BREAKTHROUGH PROBABILITY

- The greedy breakthrough probability, G(p,N), for an N-stage problem is given by:

$$G(p,N) = \big(p(2-p)\big)^N$$



ROLLOUT BREAKTHROUGH PROBABILITY



$$R(p,N) = G(p,N) + \sum_{i=2}^{N} P(\textbf{1st block of greedy path is in position } i)$$

$$P(\textbf{Alternate free path is found in stages 1 to } i)$$

Recursion:

$$R(p,N) = \big(2p(1-p) - p^2 G(p,N-1)\big) R(p,N-1) + p^2 G(p,N-1)$$

$$R(p,0) = 1$$

ROLLOUT TO GREEDY RATIO

Recursion yields

$$\frac{R(p,N)}{G(p,N)} = \frac{R(p,N-1)}{G(p,N-1)} + \frac{p}{2-p}\left(1 - R(p,N-1)\right)$$

so

$$\frac{R(p,N)}{G(p,N)} = O\left(N\frac{p}{2-p}\right)$$

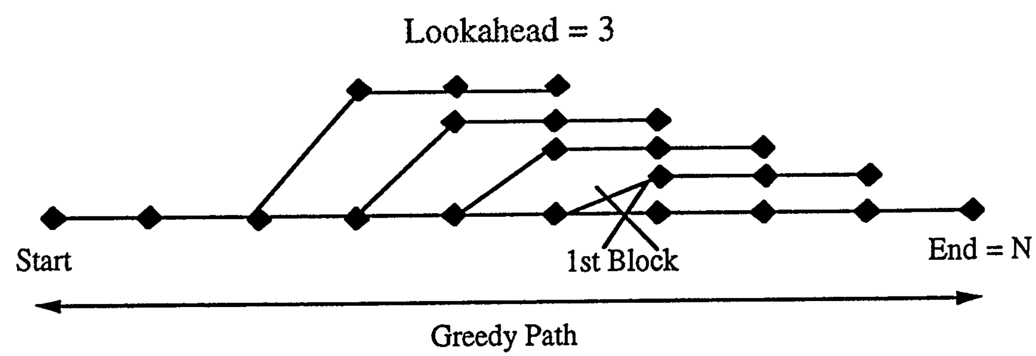
Rollout:

Does N times more work

Achieves N times better performance

ROLLOUT WITH LIMITED LOOKAHEAD m

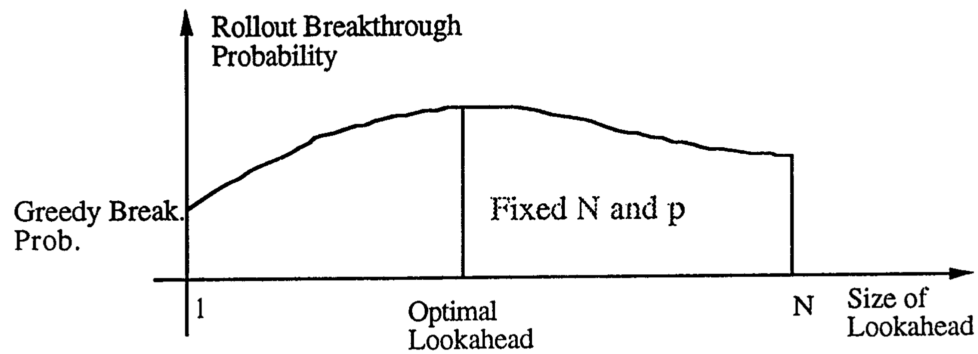
- The length of the greedy paths is reduced to m
- With lookahead of m, the complexity of rollout is reduced to O(mN)
- Limited lookahead does not guarantee improvement of rollout over greedy (for a single problem instance)

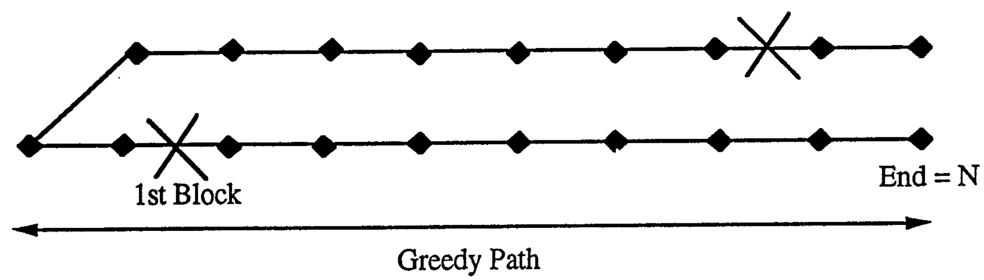


LITERATURE ON LIMITED LOOKAHEAD

- Rolling horizon results (Alden and Smith, 1992, Hernandez-Lerma and Lasserre, 1990)
- Receding horizon results (Mayne and Michalska, 1990, Keerthi and Gilbert, 1986)
- Game/computer program literature (Chess, Backgammon, etc)
- General story: As the length of lookahead increases, the performance improves

EXPERIMENTAL OBSERVATION

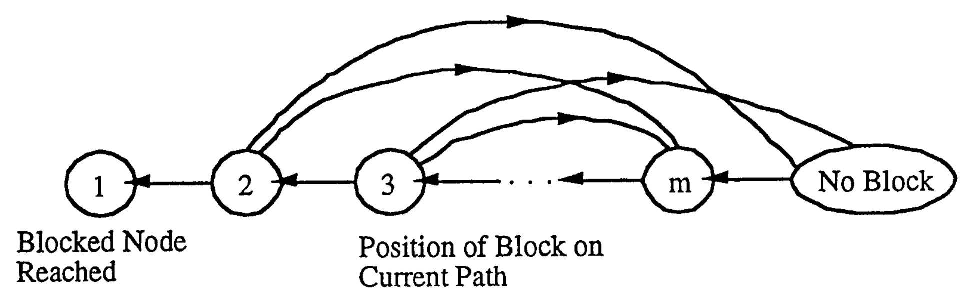




Long lookahead is more susceptible to extraneous info

MARKOV ANALYSIS OF LOOKAHEAD

- State is position of 1st block on the default path



- Rollout breakthrough prob. with lookahead m
  = Prob. of NOT reaching state 1 in N steps
  = O((max eigenvalue of transition matrix)^N)

ROLLOUT TO GREEDY RATIO

- **Rollout to greedy ratio increases exponentially**

$$\frac{R(p,N,m)}{G(p,N,m)} = O\left(\lambda(p,m)^N\right)$$

- **Optimal lookahead maximizes λ(p,m) over m, and is independent of N (for large N)**

SHORTEST PATHS WITH 0-1 LENGTHS

- Qualitatively similar story
- Interesting shortest path analysis (Karp and Pearl, 1983)
- With unlimited lookahead

$$\lim_{N\to\infty} \frac{\text{Av. Rollout Cost per Stage(p,N)}}{\text{Av. Greedy Cost per Stage(p,N)}} = 1$$

- With limited lookahead m

$$\lim_{N\to\infty} \frac{\text{Av. Rollout Cost per Stage(p,N,m)}}{\text{Av. Greedy Cost per Stage(p,N,m)}} < 1$$

The Behavioral Approach to Systems Modelling and Control

Jan C. Willems
University of Groningen
The Netherlands

J.C.Willems@math.rug.nl

RuG

Aim :

to provide a mathematical language for

- modelling
- analysis
- synthesis / control filtering

of dynamical systems

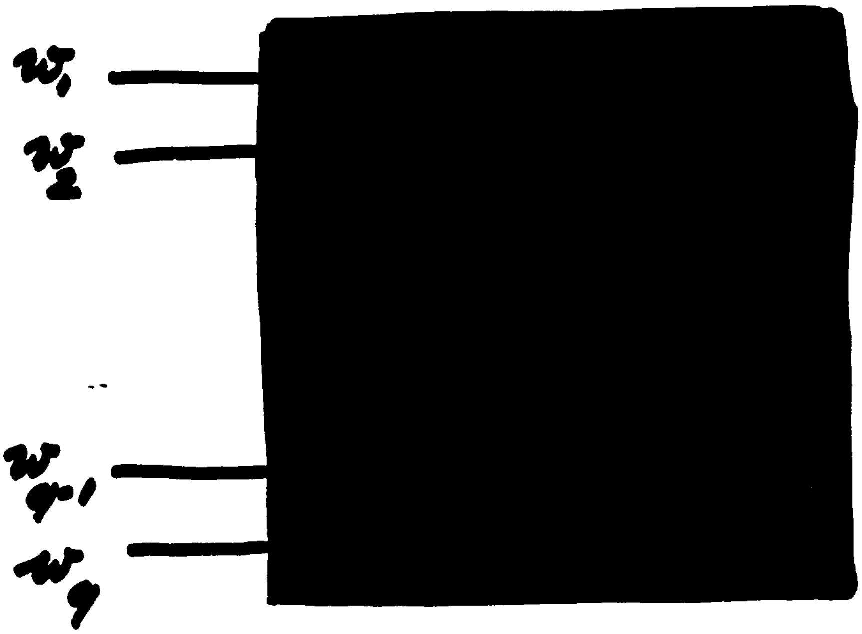
Topics to be covered

- Modelling ideas
- Latent variables
- Linear time-invariant systems
- Elimination
- Controllability
- Observability
- Control in a behavioral setting

First principles modelling

Tearing
&
Zooming

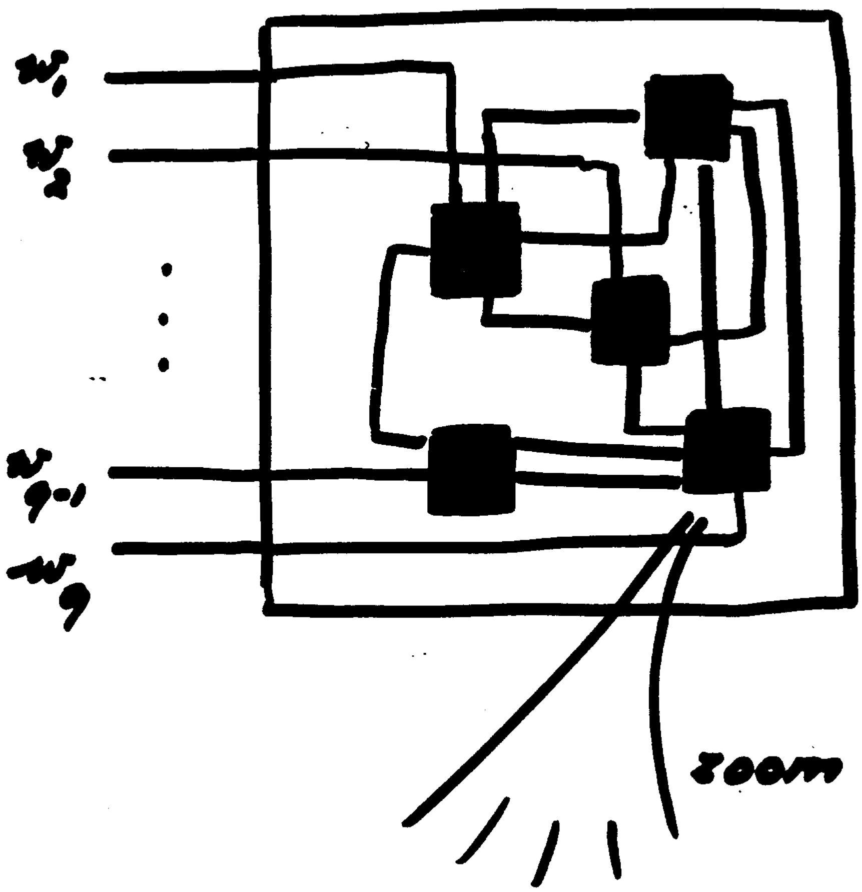
Modelling problem:
??????? ??????

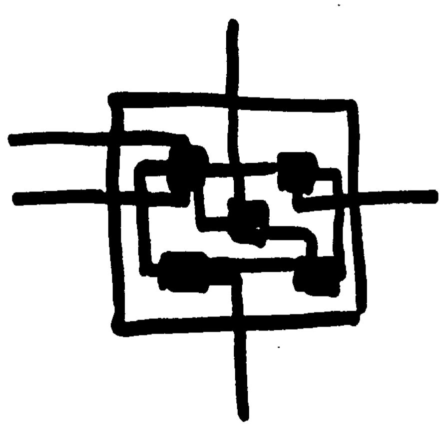


Model (dynamical) relation among variables

$w_1, w_2, \dots, w_q$

Teasing + Framing





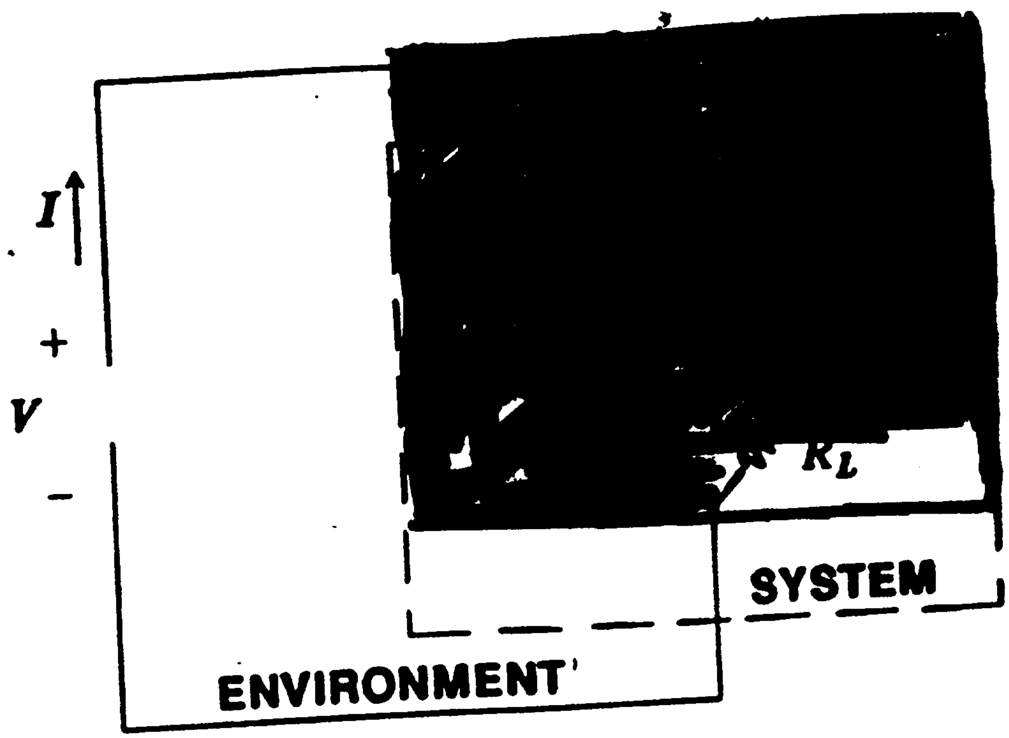
etc. etc.

Example:

an RLC - circuit

Introduce

V_{R_C}, I_{R_C} ; V_L, I_L ; V_C, I_C ; V_{R_L}, I_{R_L}



Model the relation between port voltage $\underline{V}$ and port current $\underline{I}$

Constitutive equations:

$$V_{R_C}=R_C I_{R_C}\ ;\ V_{R_L}=R_L I_{R_L}\ ;\ C\frac{dV_C}{dt}=I_C\ ;\ L\frac{dI_L}{dt}=V_L \qquad \textbf{(CE)}$$

Kirchhoff's current laws:

$$I=I_{R_C}+I_L\ ;\ I_{R_C}=I_C\ ;\ I_L=I_{R_L}\ ;\ I_C+I_{R_L}=I \qquad \textbf{(KCL)}$$

Kirchhoff's voltage laws:

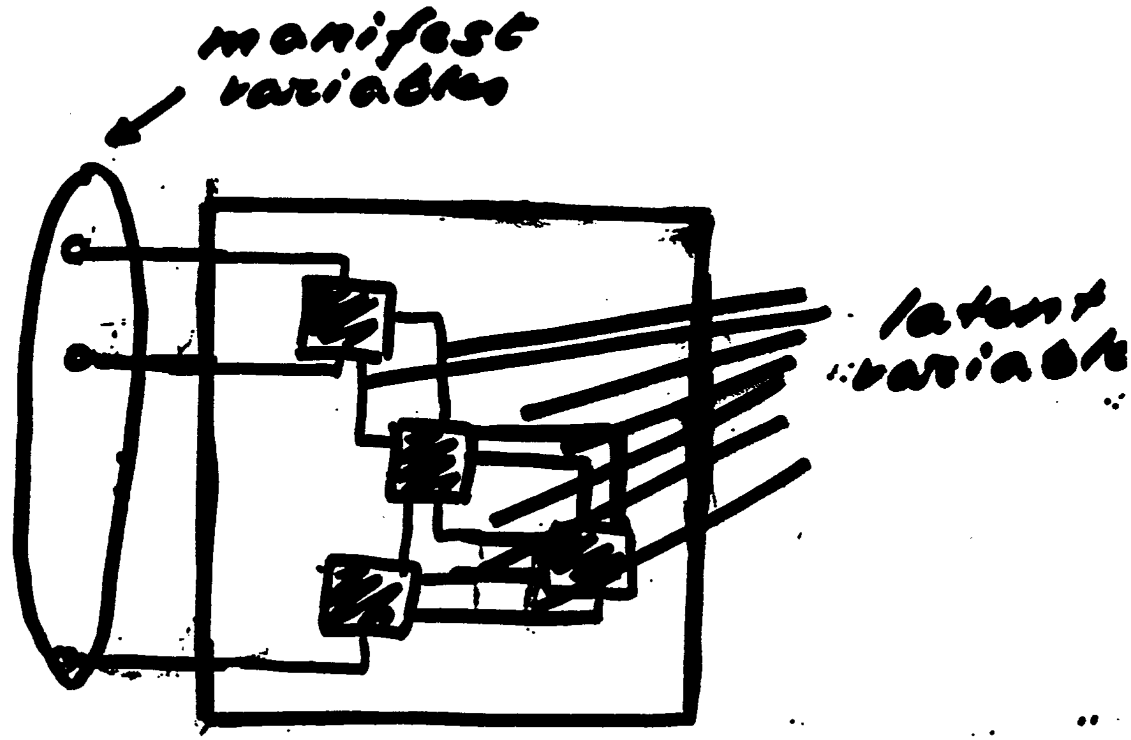
$$V=V_{R_C}+V_C\ ;\ V=V_L+V_{R_L}\ ;\ V_{R_C}+V_C=V_L+V_{R_L} \qquad \textbf{(KVL)}$$

Far distance from

$\frac{d}{dt}x = f(x,u)$ or transfer function

We want a theory
that can take such
a set of equations
as its
starting point

TEARING & ZOOMING



$$w = \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_q \end{bmatrix} \quad \text{manifest}$$

$$l = \begin{bmatrix} l_1 \\ l_2 \\ \vdots \\ l_d \end{bmatrix} \quad \text{latent}$$

Resulting model (case of diff. eqns):

manifest var.

$$f_1\left(w, \frac{d}{dt}w, \dots, \frac{d^N}{dt^N}w, \ell, \frac{d}{dt}\ell, \dots, \frac{d^N}{dt^N}\ell\right)$$

latent var.

$$= f_2\left(w, \dots, \frac{d^N}{dt^N}w, \ell, \dots, \frac{d^N}{dt^N}\ell\right)$$

system equations

This is, in principle, our <u>starting point</u> for analysis and synthesis of dynamical systems!

End point of modelling process

→ initial point of
analysis
synthesis algorithms

"behavioral approach"

Abstract
framework

Axiomatics

Formalism: DYNAMICAL SYSTEMS

Phenomena are <u>time</u> functions

A dynamical system is a triple

$$\Sigma = (T, \mathbb{W}, \mathfrak{B})$$

$T \subseteq \mathbb{R}$	time-axis
$\mathbb{W}$	signal space
$\mathfrak{B} \subseteq \mathbb{W}^T$	behavior

$T = \mathbb{R}$ (or $\mathbb{R}_+$) continuous-time s.

$= \mathbb{Z}$ (or $\mathbb{Z}_+$, $\mathbb{N}$) discrete —

$\mathbb{W} = \mathbb{R}^q$ lumped systems

$=$ finite set DES

$=$ function space DPS

The behavior expresses the <u>laws</u>

elements of $\mathfrak{B}$: possible

A dyn. system with latent var.

$$\Sigma_L = (T, \mathbb{W}, \mathbb{L}, \mathfrak{B}_f)$$

$T \subseteq \mathbb{R}$ — time-axis

$\mathbb{W}$ — space of manifest va

$\mathbb{L}$ — space of latent var.

$\mathfrak{B}_f \subseteq (\mathbb{W} \times \mathbb{L})^T$ — full behavior

$\Downarrow$

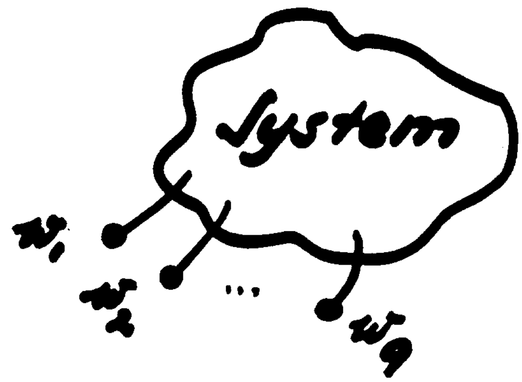
$$\Sigma = (T, \mathbb{W}, \mathfrak{B})$$

$$\mathfrak{B} := \{ w : T \to \mathbb{W} \mid \exists\, \ell : T \to \mathbb{L} \text{ such that } (w, \ell) \in \mathfrak{B}_f \}$$

manifest behavior — object of model

Linear time-invariant systems

Dynamical System



$R(\frac{d}{dt})w =$

dynamic relation among variables $(w_1, w_2, \ldots, w_q)$: system ≈ behavior

---

- finite number of real variables

- relation is linear
  time-invariant

- laws can be expressed as differential equations

→ (linear time-invariant) Differential systems **LTI**

$w = \begin{bmatrix} w_1 \\ \vdots \\ w_q \end{bmatrix}$

model variables

$$R_0 w + R_1 \frac{dw}{dt} + \cdots + R_L \frac{d^L w}{dt^L} = 0$$

$$\underbrace{R_0, R_1, \ldots, R_L}_{\text{model parameters}} \in \mathbb{R}^{\bullet \times q}$$

$$R(\xi) := R_0 + R_1 \xi + \cdots + R_L \xi^L \in \mathbb{R}^{\bullet \times q}[\xi]$$

$\bullet$ = # of eqns = rowdim (R)

$$\boxed{R\left(\frac{d}{dt}\right) w = 0}$$

$$\Sigma = (\mathbb{R}, \mathbb{R}^q, \mathfrak{B})$$

$\mathfrak{B}$: solution set of kernel representation

Polynomial matrices

$$R \in \mathbb{R}^{\bullet \times \bullet}[\xi]$$

- ξ: "indeterminate"
- $\mathbb{R}$: real coefficients
- first $\bullet$: number of rows
- second $\bullet$: number of columns

$$\xi \to x \in \mathbb{R} \quad \leadsto \quad R(x) \in \mathbb{R}^{\bullet \times \bullet}$$

$$\xi \to \lambda \in \mathbb{C} \quad \leadsto \quad R(\lambda) \in \mathbb{C}^{\bullet \times \bullet}$$

$$\xi \to \frac{d}{dt} \quad \leadsto \quad R\left(\frac{d}{dt}\right) \qquad \text{differential operator}$$

$$\xi \to A \in \mathbb{R}^{n \times n} \quad \leadsto \quad R(A) \qquad \text{matrix}$$

etc.

$$R(\xi) = R_0 + R_1 \xi + \cdots + R_{L-1} \xi^{L-1} + R_L \xi$$

Linear time-invariant case
with latent variables

$$R_0 w + R_1 \frac{d}{dt} w + \cdots + R_N \frac{d^N}{dt^N} w$$
$$= M_0 \ell + M_1 \frac{d}{dt} \ell + \cdots + M_N \frac{d^N}{dt^N} \ell$$

may be written more compactly as

$$\boxed{R(\frac{d}{dt}) w = M(\frac{d}{dt}) \ell}$$

LTI starting point

$$R(\xi) = R_0 + R_1 \xi + \cdots + R_N \xi^N$$
$$M(\xi) = M_0 + M_1 \xi + \cdots + M_N \xi^N$$

polynomial matrices

Ex.: the RLC-circuit

$$\underbrace{\begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 1 \\ 0 & 0 \\ 0 & 0 \\ 0 & 1 \\ 1 & 0 \\ 0 & 0 \end{bmatrix}}_{R(\frac{d}{dt})} \cdot \underbrace{\begin{bmatrix} V \\ I \end{bmatrix}}_{w} = \underbrace{\begin{bmatrix} 1 & -R_c & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & -R_L \\ 0 & 0 & C\frac{d}{dt} & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & L\frac{d}{dt} & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 1 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & -1 \\ 0 & 0 & 0 & -1 & 0 & 0 & 0 & -1 \\ -1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & -1 & 0 & -1 & 0 \end{bmatrix}}_{M(\frac{d}{dt})} \underbrace{\begin{bmatrix} V_{R_c} \\ I_{R_c} \\ V_c \\ I_c \\ V_L \\ I_L \\ V_{R_L} \\ I_{R_L} \end{bmatrix}}_{\ell}$$

"Elimination"
theorem

Elimination

first principles model

$$R\left(\frac{d}{dt}\right)w = M\left(\frac{d}{dt}\right)\ell$$

in this case (LTI!) there exists another polynomial matri[x] such that

$$\boxed{\tilde{R}\left(\frac{d}{dt}\right)w = 0}$$

kernel representat[ion]

is equivalent

meaning: same trajectories w

$$(R, M) \longrightarrow \tilde{R}$$

Gröbner basis type algorithms

Ex. : RLC - [illegible]

case 1 : $\frac{R_L}{L} \neq \frac{1}{CR_c}$

solve for v_c -- obtain

(CE) + (*) + (**) + RED +

$$\frac{R_c}{R_L} + \left(1+\frac{R_c}{R_L}\right) CR_c \frac{d}{dt} + CR_c \frac{L}{R_L} \frac{d^2}{dt^2} \Big) v$$

$$= \left(1 + CR_c \frac{d}{dt}\right)\left(1 + \frac{L}{R_L} \frac{d}{dt}\right) R_c I$$

case 2 : $\frac{R_L}{L} = \frac{1}{CR_c}$ obtain

(CE) + (*) + (**) $\Rightarrow$ $v_c + CR_c \frac{dv_c}{dt} = v$

$$\left(\frac{R_c}{R_L} + CR_c \frac{d}{dt}\right) V = \left(1 + CR_c \frac{d}{dt}\right) R_c I$$

$$f_1\left(w, \frac{dw}{dt}, \ldots, \frac{d^N w}{dt^N}, \ell, \frac{d\ell}{dt}, \ldots, \frac{d^N \ell}{dt^N}\right)$$

$$= f_2 (\text{———})$$

Are the laws implied by this system of diff. eqns. on the variables $w = (w_1, w_2, \ldots, w_9)$ also differential equations?

In general : <u>NO</u>!

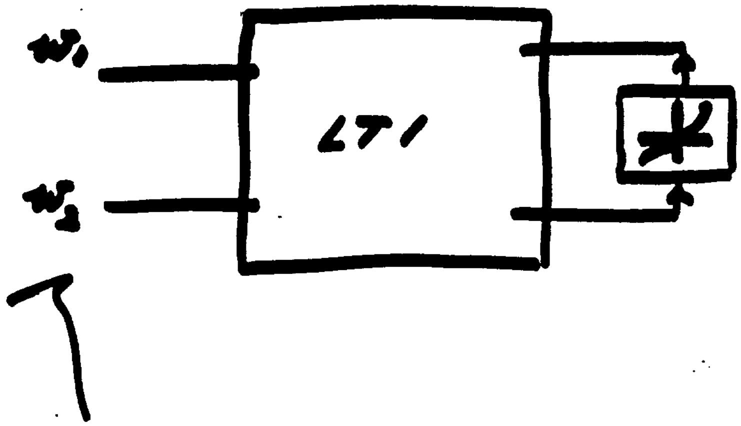
(not a matter of smoothness.

$$\tilde{f}_1\left(w, \frac{d}{dt} w, \ldots, \frac{d^{\tilde{N}}}{dt^{\tilde{N}}} w\right) = \tilde{f}_2(\text{—})$$

$$\frac{d}{dt} x = f(x, w_1) \qquad w_2 = h(x, w_2)$$

$$\frac{d}{dt} x = f(x)$$

doubtfull! starting pts



no reason why relation
between w_1 and w_2
is a differential equation!

not a matter of smoothess ...

Controllability

and

Observability

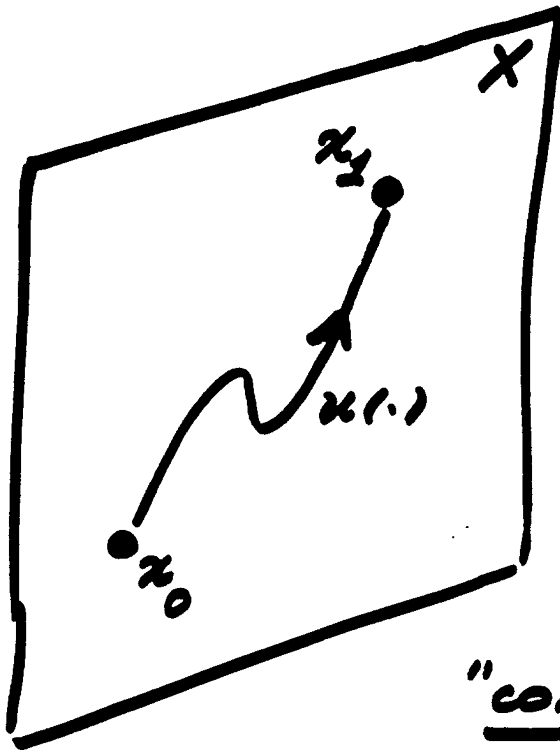
in a

Behavioral

Setting

$$\frac{d}{dt}x = f(x, u) \qquad (y = h(x, u))$$

lassical def:



"controllability"

'Thm': Every input/output system (transfer fn, convolution $\int$) can be represented by a st. sp. system that is contr. & observ.

Not a system property? Representation dependent?

Since depends on state representation!

$\Sigma = (T, W, \mathcal{B})$ is said to be controllable if

$\forall\, w_1, w_2 \in \mathcal{B}$

$\exists\, w \in \mathcal{B}$ and $T \geq 0$

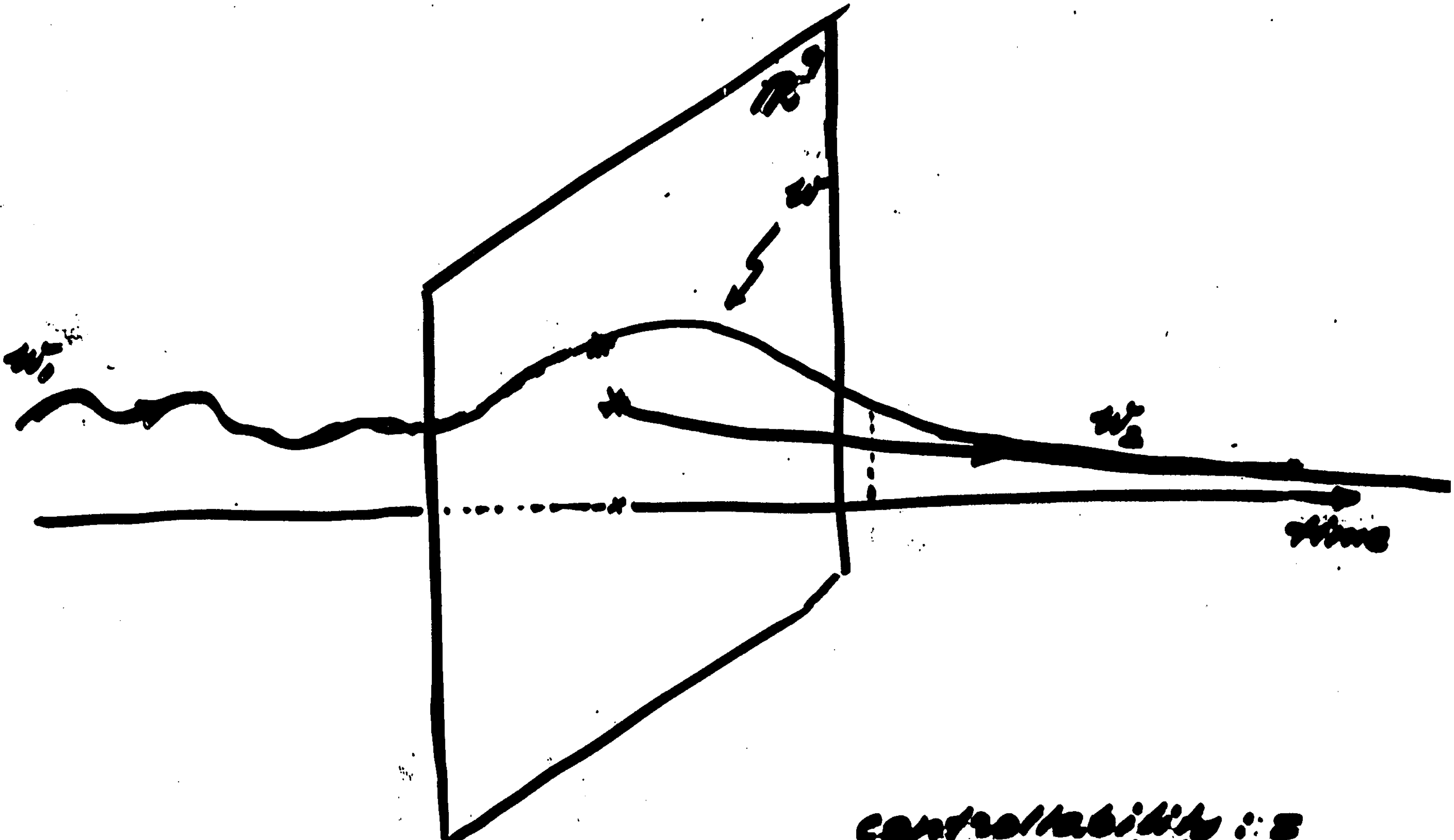
$\ni$

$$w(t) = \begin{cases} w_1(t) & t < 0 \\ w_2(t-T) & t \geq T \end{cases}$$

Def. is representation independent

nicely generalizable to n-D systems e.g. PDE's

w_1, w_2 : [illegible]
behavioral eq
w : idem



controllability :⇔
w exists
for all $w_1, w_2 \in \mathfrak{B}$

Result : $R_0 + R_1\xi + \cdots + R_N\xi$
is a polynomial matrix

$R(\frac{d}{dt})w = 0$ is

controllable if and only if

$\Leftrightarrow$ $R(\lambda)$ has the same rank for all complex $\lambda \in \mathbb{C}$

real matrices

$= R_0 + R_1\lambda + \cdots + R_N\lambda^N$

complex numbers

$R(\lambda)$ complex matrix

cfr PBH-te

---

1. there exists algorithms deciding controllability from $R_0, R_1, \ldots, R_N$.
2. this result has been generalized to nonlinear (diff. algebraic) syste

$$f(w, \frac{d}{dt}w, \ldots, \frac{d^N}{dt^N}w) = 0$$

Ore rings

equivalence :

$$R(\tfrac{d}{dt})\,w = 0$$

defines a controllable (LTI) system iff and only iff it admits a representation as

$$w = M(\tfrac{d}{dt})\,\ell$$

"flat" systems

also studied in nonlinear case

- $R(\tfrac{d}{dt})\,w = M(\tfrac{d}{dt})\,\ell$ — first principles model
- $R(\tfrac{d}{dt})\,w = 0$ — kernel repr. : after elimination
- $w = M(\tfrac{d}{dt})\,\ell$ — image repr. : $\Leftrightarrow$ controllable

... for LTI case ...

generalization to differential- delay systems:

$$R\left(\frac{d}{dt}, \Delta\right) w = 0$$

diff. operator ($\frac{d}{dt}$) — delay (Δ)

is controllable iff

$$\boxed{\text{rank } R(\lambda, e^{-\lambda}) \text{ constant}}$$

for $\forall \lambda \in \mathbb{C}$

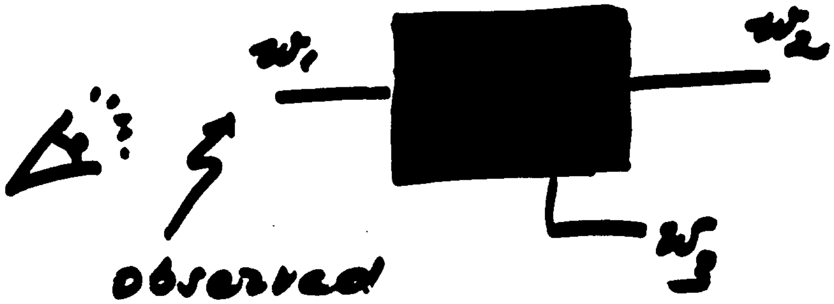
---

generalized in many directions:

- delay - differential systems
- PDE's
- time - varying systems
- nonlinear systems 'flat'

very active area of research

$$\Sigma = (\mathbb{R}, \mathbb{W}_1 \times \mathbb{W}_2, \mathcal{B})$$



<u>Def.</u> : w_2 is <u>observable</u> from w_1

if

$$(w_1, w_2') \in \mathcal{B}$$
$$(w_1, w_2'') \in \mathcal{B}$$
$$\Rightarrow \; w_2' = w_2''$$

<u>Rem</u>: 1. $\exists \; F : \mathbb{W}_1^{\mathbb{R}} \longmapsto \mathbb{W}_2^{\mathbb{R}}$

$$(w_1, Fw_1) \in \mathcal{B}$$
$$\forall \, w_1 \in \mathcal{B}_1 := \cdots$$

2. Classical defn. : $\quad w_2 = x$

3. Special interest : $\mathbb{W}_1 = \mathbb{W}, \; \mathbb{W}_2 = L$

manifest → latent

$$R_1\left(\frac{d}{dt}\right) w_1 = R_2\left(\frac{d}{dt}\right) w_2$$

w_2 observable from w_1 ?

Test :

$$\text{rank}\,(R_2(\lambda)) = \dim(w_2)$$

$$\forall \lambda \in \mathbb{C}$$

Representation :

$$R_1'\left(\frac{d}{dt}\right) w_1 = 0$$

$$w_2 = R_2'\left(\frac{d}{dt}\right) w_1$$

Generalization :

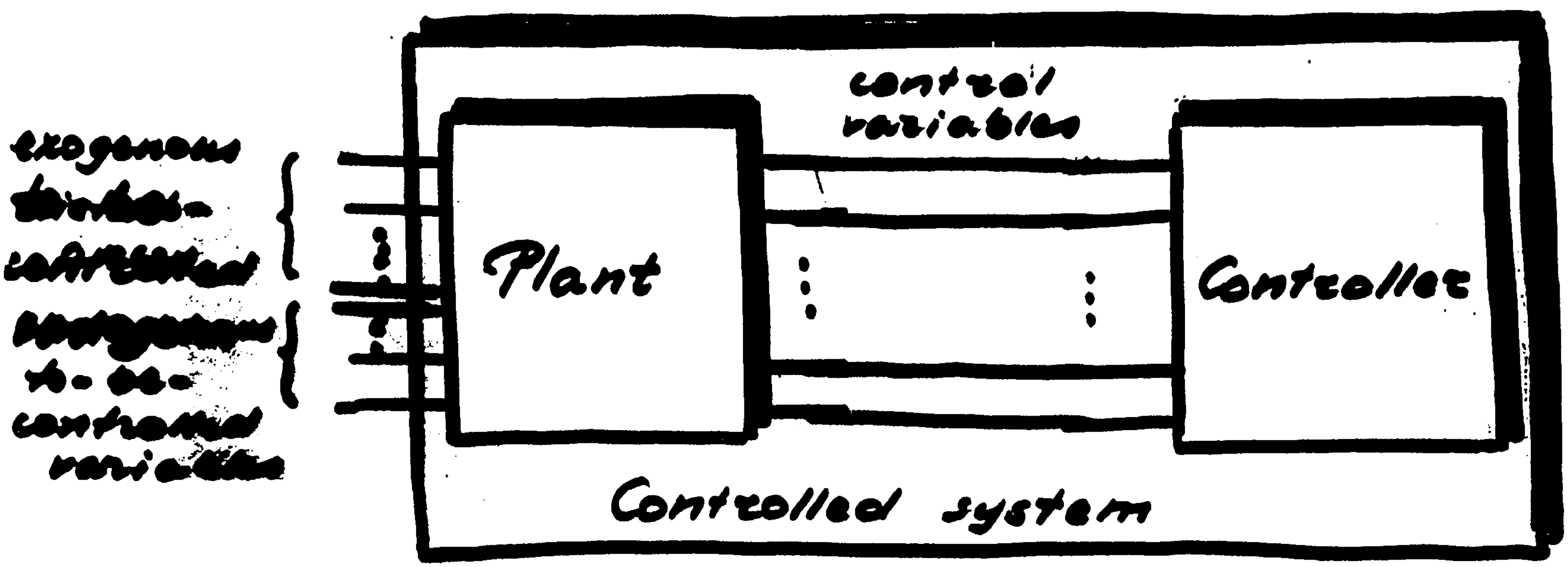
A-D systems

PDE's

nonlinear systems ...

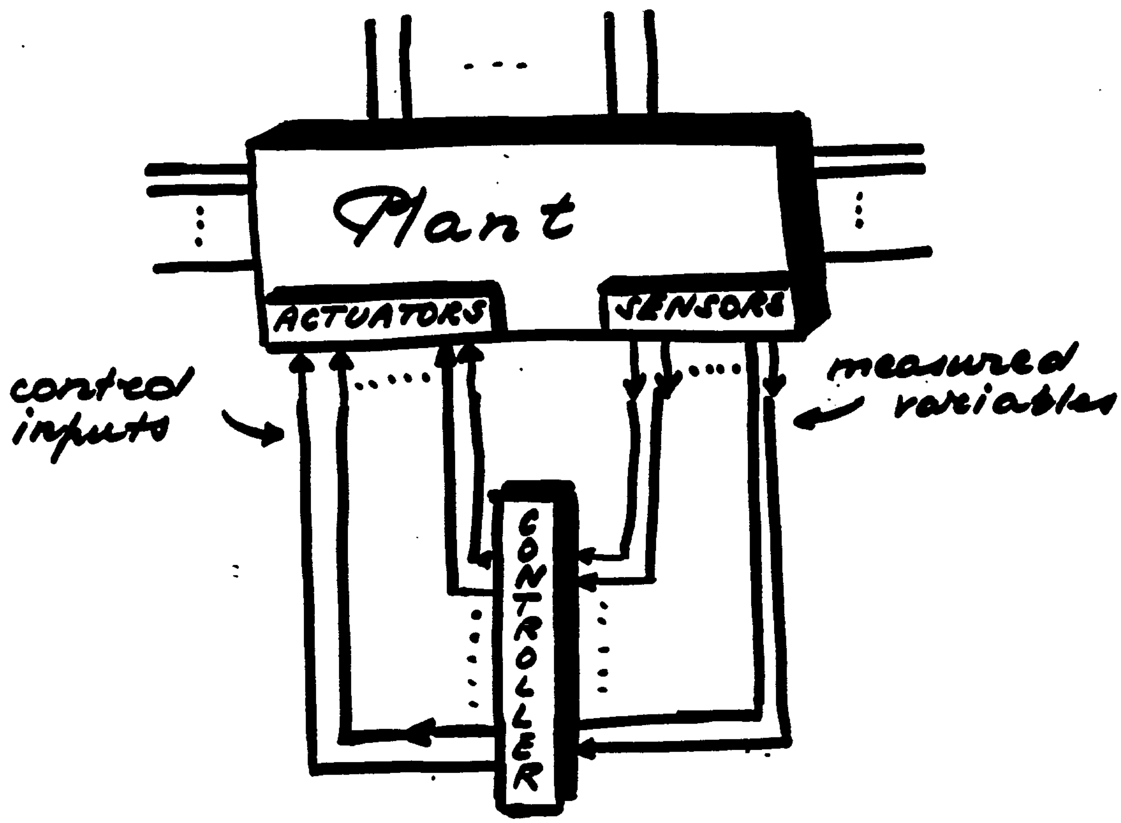
Control as Interconnection

Control



Given plant, design controller such that controlled system meets certain specifications

Special case:



"Intelligent" control

Control = designing a **subsystem**

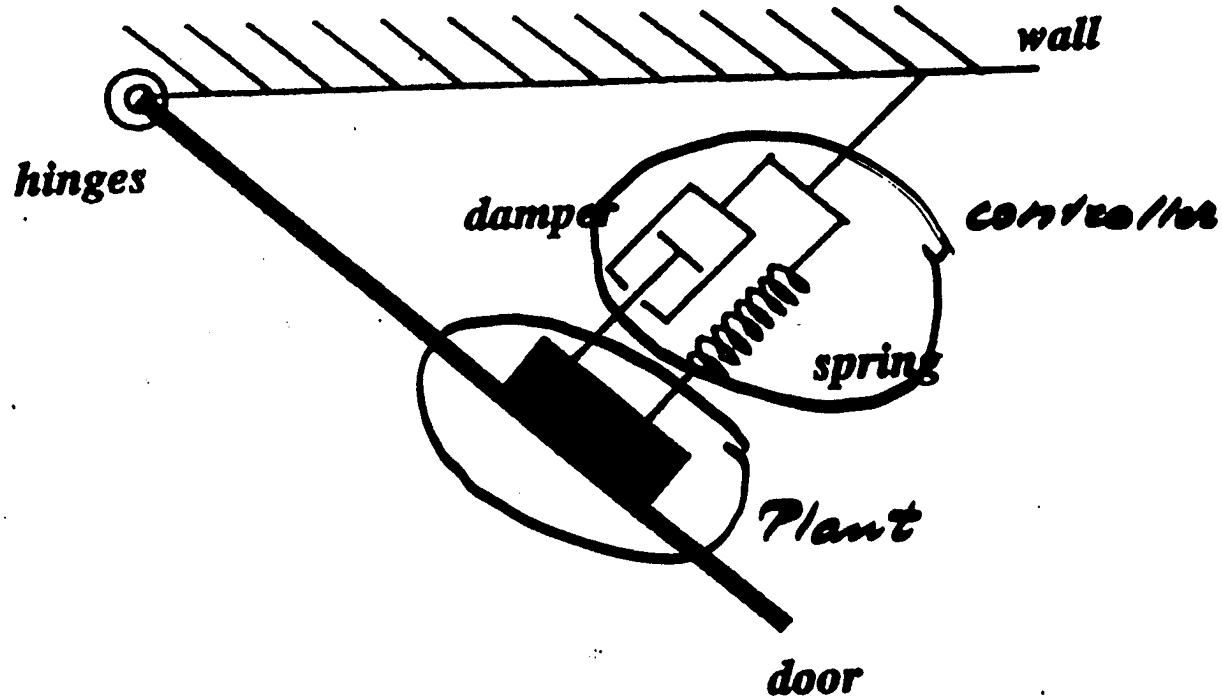
= adding new laws to system variables

System = a family of trajectories

Control = selecting a suitable subfamily

- class of admissible controller
- objectives, optimality
- specification of controlled traj.
- implementation

Motivational Examples



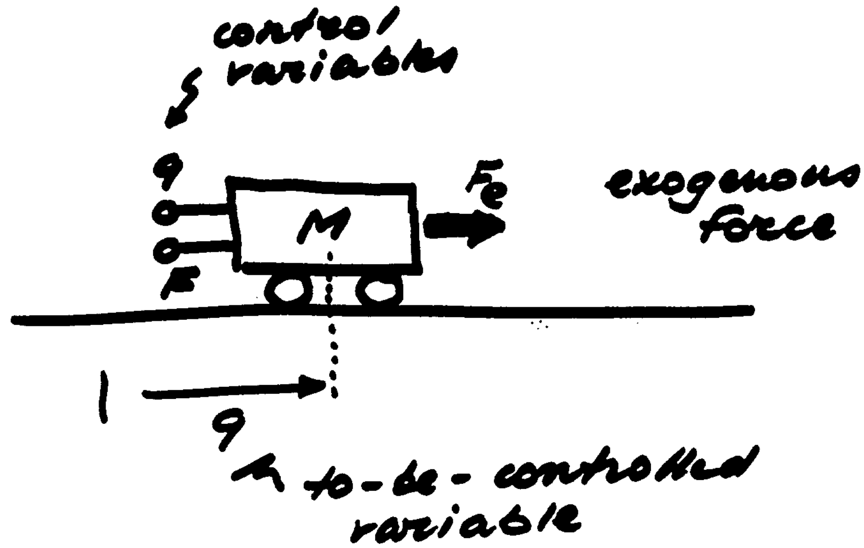
A door-closing mechanism

other similar examples: shock absorbers in cars

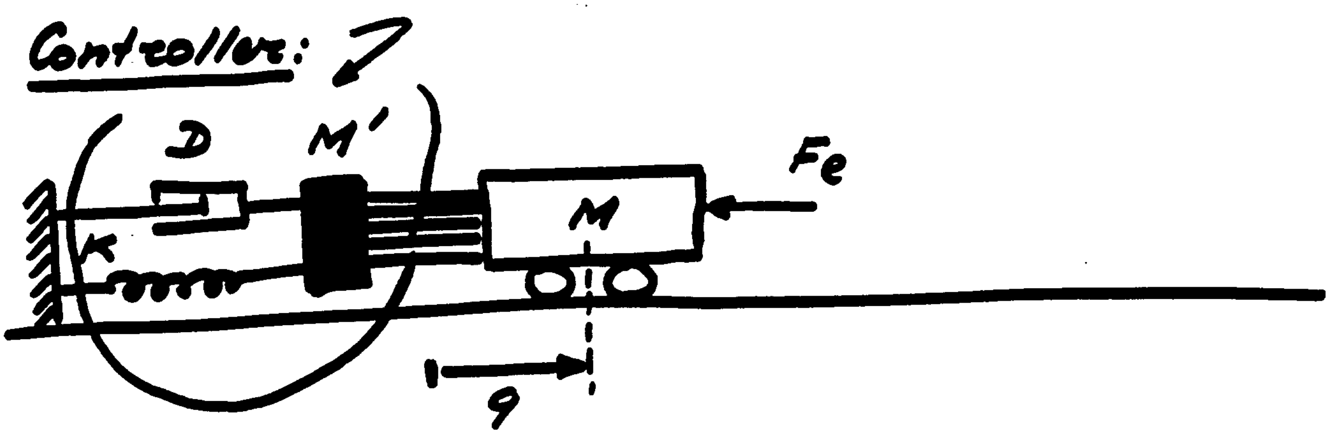
SIMPLE EXAMPLE

Hold a mass at a particular position

Plant:



Controller:



K, D, M' : to-be-designed

- fast settling time
- small overshoot
- low steady state gain $F_e \rightarrow q$

Want :

$$M \frac{d^2}{dt^2} q = F_e + F$$

Control law:

$$F = - M' \frac{d^2}{dt^2} q - D \frac{d}{dt} q - K q$$

Controlled system:

$$(M + M') \frac{d^2}{dt^2} q + D \frac{d}{dt} q + K q = F_e$$

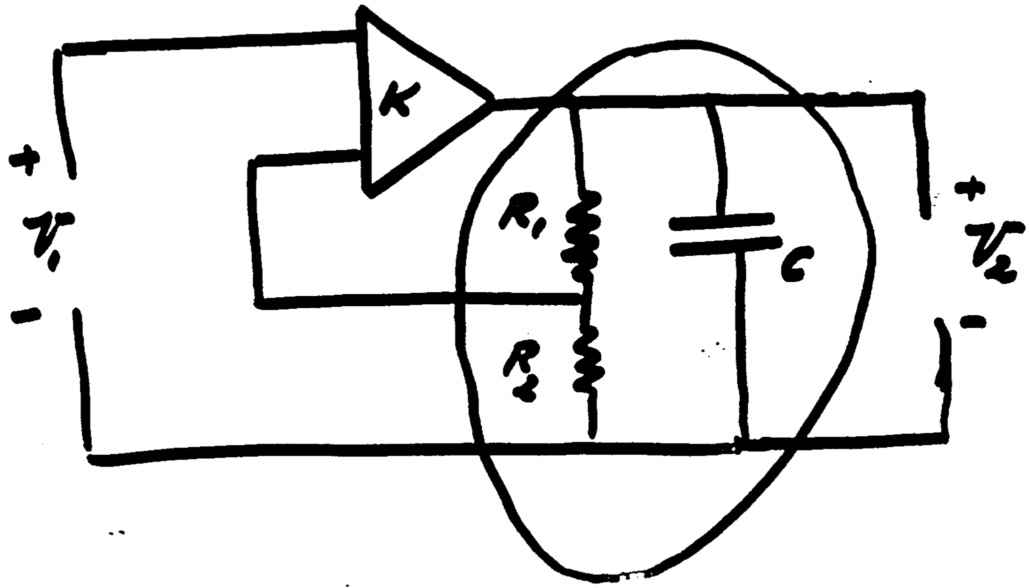
... design M', D, F

Practical example:

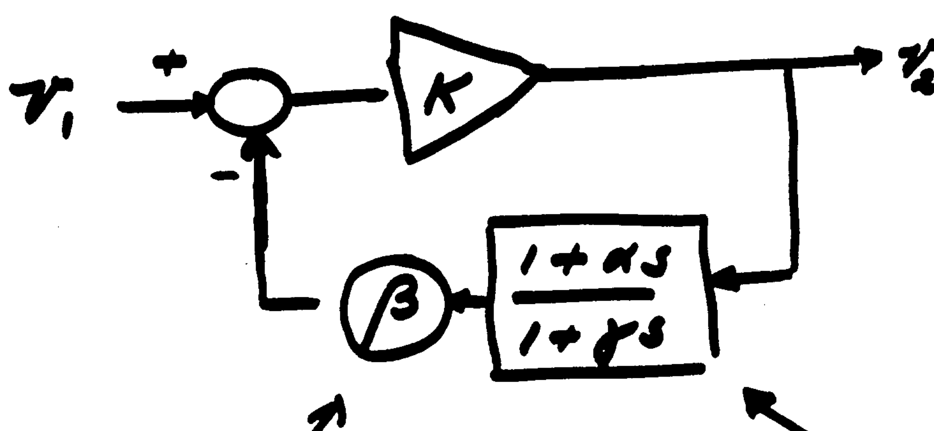
car damper

Operational amplifier

Black, Nyquist, Bode



↑ **controller** for robustness loop-shaping



limits bandwidth ≅ stability

$$\frac{K}{1+\beta K} = \frac{1}{\beta + \frac{1}{K}} \cong \frac{1}{\beta}$$

insensitive to K! ---- stability

Implementability

of a

controlled

behavior

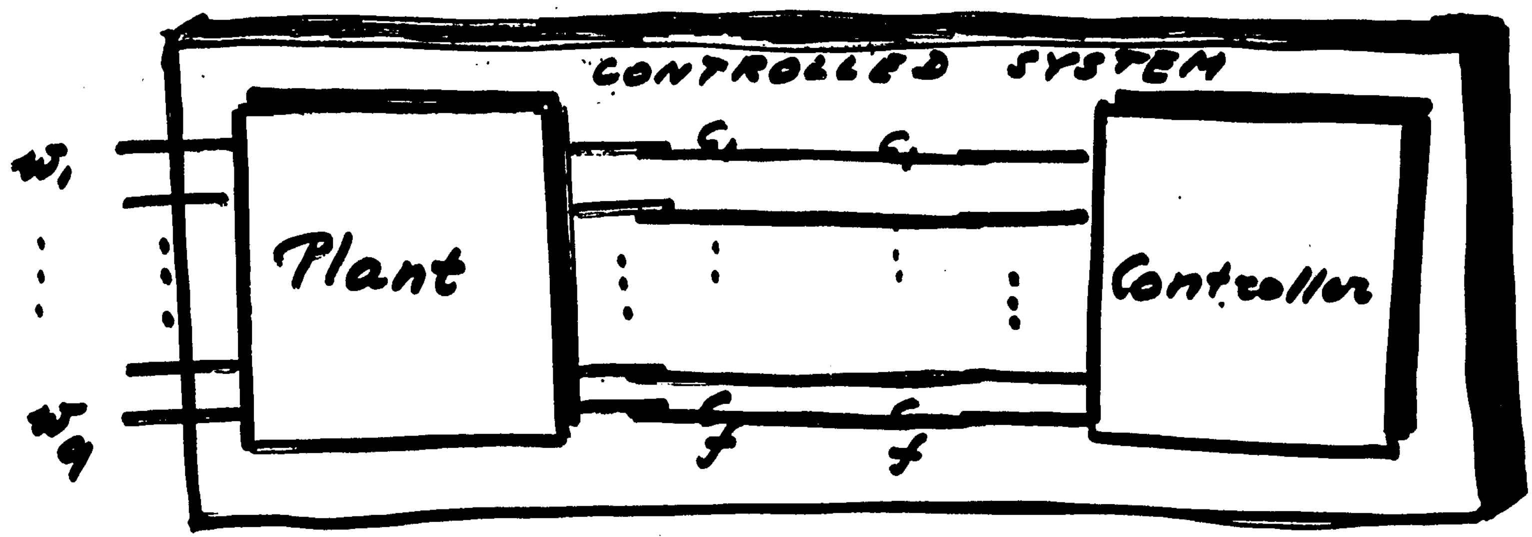
—

Linear

time - invariant

case

Controlled systems



relevant behaviors: $\mathcal{P}_{full}$, $\mathcal{P}$, $\mathcal{C}$, $\mathcal{K}$, $\mathcal{N}$

assume all of them: LTI differential systems

$\mathcal{P}_{full}$	behavior of variables (w, c) before control
$\mathcal{P}$	behavior of variables w before control
$\mathcal{C}$	behavior of variables c as imposed by controlled
$\mathcal{K}$	behavior of variables w after control
$\mathcal{N}$	behavior of variables w compatible with $c = ($

$\in \mathcal{L}^{q+f}$

$$\| \; \{(w,c) \in C^\infty(\mathbb{R},\mathbb{R}^{q+f}) \mid \text{laws of plant } \checkmark\}$$

$\in \mathcal{L}^{q}$

$$\| \; \{w \in C^\infty(\mathbb{R},\mathbb{R}^{q}) \mid \exists c \ldots \}$$

$\in \mathcal{L}^{f}$

$$\| \; \{c \in C^\infty(\mathbb{R},\mathbb{R}^{f}) \mid \text{laws of controller } \checkmark\}$$

$\in \mathcal{L}^{q}$

$$\| \; \{w \in C^\infty(\mathbb{R},\mathbb{R}^{q}) \mid \exists c \in \mathcal{C} \ni (w,c) \in \mathcal{P}_{full}\}$$

$\in \mathcal{L}^{q}$

$$\| \; \{w \in C^\infty(\mathbb{R},\mathbb{R}^{q}) \mid (w,0) \in \mathcal{P}_{full}\}$$

Obviously:

$$\mathcal{N} \subset \mathcal{K} \subset \mathcal{P}$$

$$\pi = \{ w \mid (w, 0) \in \mathcal{P}_{full} \}$$

π = 'hidden' behavior

on π controller is bound to be inactive: no information is obtained on what is happening in the plant

Def: We say that

C implements $\mathcal{K}$

if $\quad \mathcal{K} = \{ w \mid \exists\ c \in C$ such that $(w, c) \in \mathcal{P}_{full} \}$

Which $\mathcal{K} \in \mathcal{L}^q$ can be implemented?

? to what extent can the system behavior be modified by controller?

Basic **lemma**:

hidden behavior

plant behavior

Given $\pi, \mathcal{P} \in \mathcal{L}^q$; $\pi \subset \mathcal{P}$

$\mathcal{K} \in \mathcal{L}^q$ is implementable

if and only if

$$\pi \subset \mathcal{K} \subset \mathcal{P}$$

Full information

$\pi = \{0\}$

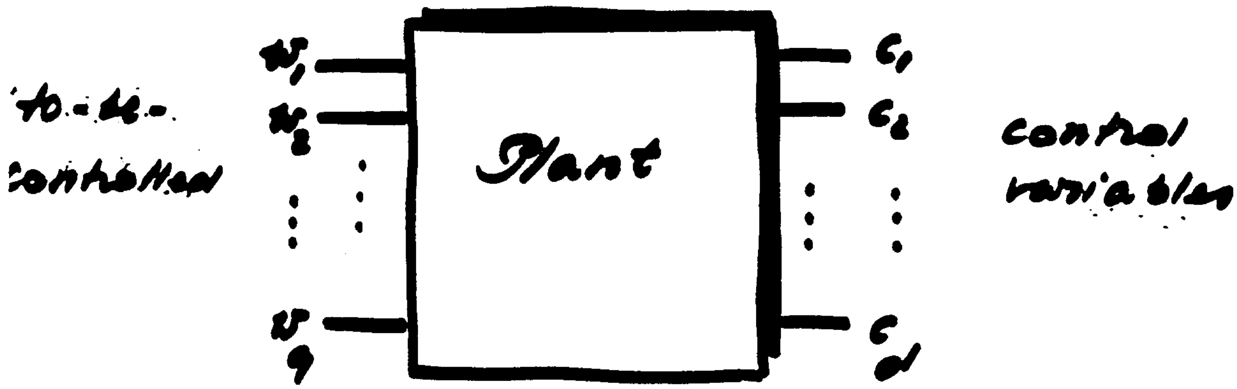
$\iff$ w observable from c

In full information case,
<u>every subbehavior</u> is
implementable

---

more restrictive notion of
implementability

$\Rightarrow$ more realistic
results



$$R\left(\frac{d}{dt}\right) w = C\left(\frac{d}{dt}\right) c$$

If w variables are ~~controllable~~ stabilizable

& w variables are ~~observable~~ detectable from c variables

then there exists a regular controller such that controlled system is stable

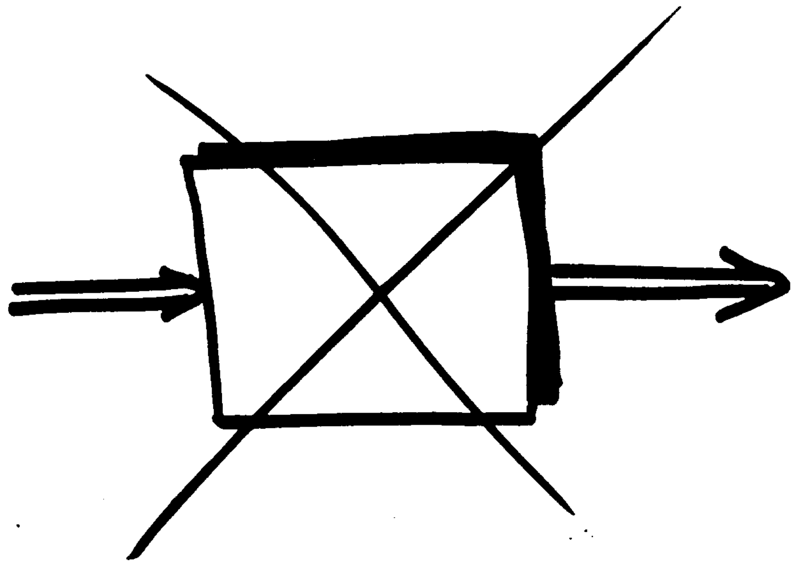
This framework has been applied to

- Stabilization
- Pole placement with memoryless feedback

  $n < mp \Rightarrow$ generic pole placement

- $\mathcal{H}_\infty$ - control
- $\mathcal{H}_2$ - control

Main idea:



A physical system
is not a
signal processor

Conclusions

- The behavioral approach gives a very cogent framework for discussing math. models dynamical systems

- convincing treatment of
  - controllability
  - observability
  - (stability)
  - (dissipative systems)

- has been applied to a number of synthesis problems:
  - controllers
  - (observers)
  - (system identification)

- Generalizable to
  PDE's, non-linear systems
  (stochastics ?)

REFERENCES

BOOK

1. J.W. Polderman and JCW, *Introduction to Mathematical Systems Theory: A Behavioral Approach*, Springer-Verlag, New York, 1998.

GENERAL ARTICLES

2. JCW, Paradigms and puzzles in the theory of dynamical systems, *IEEE Transactions on Automatic Control*, Volume 36, pages 259-294, 1991.

3. JCW, Models for dynamics, *Dynamics Reported*, Volume 2, pages 171-269, 1989.

4. JCW, From time series to linear system , Part I. Finite dimensional linear time invariant systems, Part II. Exact modelling, Part III. Approximate modelling, *Automatica*, Volume 22, pages 561-580, pages 675-694, 1986; Volume 23, pages 87-115, 1987.

5. JCW, The behavioral approach to systems and control, *European Journal of Control*, Volume 2, pages 250-259, 1996.

CONTROL

6. JCW and H.L. Trentelman, H_∞ control in a behavioral context: The full information case , *IEEE Transactions on Automatic Control*, Volume 44, pages 521-536, 1999.

7. JCW, On interconnections, control, and feedback, *IEEE Transactions on Automatic Control*, Volume 42, pages 326-339, 1997.

8. JCW, Generic eigenvalue assignability by real memoryless output feedback made simple, *Communications, Computation, Control and Signal Processing*, (Eds. A. Paulraj, V. Roychowdhury, and C.D. Schaper), Kluwer, pages 343-354, 1997.

9. JCW and H.L. Trentelman, On quadratic differential forms, *SIAM Journal on Control and Optimization*, Volume 36, pages 1702-1749, 1998.

10. JCW, Path integrals and stability, *Mathematical Control Theory* (Eds. J. Baillieul and J.C. Willems), Springer Verlag, pages 1-32, 1998.

RECENT PROGRESS IN FEEDBACK PASSIVATION DESIGNS

Petar V. Kokotović

Center for Control Engineering and Computation
University of California, Santa Barbara

ACTIVATION OF CONCEPTS

Over the last ten years several **descriptive** concepts have been **activated** into constructive **design tools** such as

- Lyapunov functions → Control Lyapunov functions (CLF)
- Total stability → Input/state stability (ISS)
- Passivity → Passivation
- Optimality → Inverse optimality

leading to **design procedures** like

- Cascade passivation
- Backstepping: robust, adaptive, stochastic
- Forwarding
- Passivation using LMI's

FOUNDATIONS:

absolute stability: Lurie (1951),

passivity: Popov (1960),

PR Lemma: Yakubovich (1962), Kalman (1963),

passivity and small gain: Zames (1966),

inverse optimality: Kalman (1964), Moylan and Anderson (1973),

dissipative systems: Willems (1972), Hill and Moylan (1976),

conicity and margins: Safonov and Athans (1977), Safonov (1980).

DESIGN TOOLS AND PROCEDURES:

Lyapunov design: Meilakhs (1976,1978), Jurdjević and Quinn (1978), Corless and Leitmann (1981), Qu (1998),

control Lyapunov functions: Artstein (1983), Sontag (1989),

backstepping: Tsinias (1989), Byrnes and Isidori (1989),

ISS and ISS small gain: Sontag (1989), Jiang, Teel and Praly (1994), Teel (1996),

passivation: PK and Sussmann (1989), Byrnes, Isidori and Willems (1991),

adaptive backstepping: Kanellakopoulos, PK and Morse (1991), Krstić, Kanellakopoulos and PK (1992),

robust backstepping: Freeman and PK (1992), Marino and Tomei (1993),

ISS and H_∞ backstepping: Jiang, Teel and Praly (1994), Pan and Başar (1998), Krstić and Li (1998),

stochastic backstepping: Krstić and Deng (1998), Pan and Başar (1999),

forwarding: Teel (1992), Mazenc and Praly (1996), Janković, Sepulchre and PK (1996).

In this lecture

- **Locally optimal backstepping design**

  Ezal, Pan and PK (1997)

- **Passivation redesign**

  Arcak, Seron, Braslavsky and PK (1998)

- **Observer-based feedback passivation**

  Larsen and PK (1999)

- **Passive observer design**

  Arcak and PK (1999)

- **Example: Ship steering**

  Arcak, Fossen and PK (1999)

CLF Construction

Backstepping Lemma If $V(X)$ is a CLF for

$$\dot{X} = F(X) + G(X)u$$

and

$$L_{(F+G\alpha)}V = L_F V(X) + L_G V(X)\,\alpha(X) \leq -\sigma(X)$$

then,

$$V_+(X,x) = V(X) + (x - \alpha(X))^2$$

is a CLF for

$$\begin{aligned} \dot{X} &= F(X) + G(X)x \\ \dot{x} &= f(X,x) + g(X,x)u \qquad g(X,x) \neq 0. \end{aligned}$$

Example $$\begin{aligned} \dot{X} &= X^3 + x \\ \dot{x} &= u\,. \end{aligned}$$

$$V(X) = X^2\,, \quad \alpha(X) = -X^3 - X$$

$$V_+(X,x) = X^2 + (x + X + X^3)^2.$$

$$
\begin{aligned}
\dot{X}_0 &= F_0(X_0) + G_0(X_0)x_1 \\
\dot{x}_1 &= f_1(X, x_1) + g_1(X, x_1)x_2 \\
\dot{x}_2 &= f_2(X, x_1, x_2) + g_2(X, x_1, x_2)x_3 \qquad \text{(SF)} \\
\vdots &= \vdots \\
\dot{x}_n &= f_n(X, x) + g_n(X, x)u\,.
\end{aligned}
$$

$$g_i(X, \cdots, x_i) \neq 0$$

$$L_{(F_0+G_0\alpha_0)}V_0(X_0) \leq -\sigma_0(X_0)$$

Step 1. Set $X = X_0$, $x = x_1$, $V(X) = V_0(X_0)$, apply backstepping lemma: $\quad V_+ =: V_1(X, x_1)$.

Step i. $(i = 2, \cdots, n)$

$$X = \begin{bmatrix} X_0 \\ \vdots \\ x_{i-1} \end{bmatrix}, \; x = x_i \;\rightarrow\; V_+ =: V_i(X, x_1, \cdots, x_i).$$

Step n $\quad \rightarrow \quad \boxed{V_n(X, x_1, \cdots, x_n)}$

Locally Optimal Backstepping Design

K. Ezal et al. – CDC 97 Best Student Paper

$$\dot{x} = Ax + f(x) + G_1(x)w + G_2(x)u \qquad \text{(strict-feedback class)}$$

Robust backstepping:

$$\begin{aligned} V_i &:= V_{i-1} + \delta_i z_i^2 \\ z_{i+1} &:= x_{i+1} - \alpha_i(x_1, \ldots, x_i) \end{aligned} \qquad \frac{\partial \alpha_i}{\partial x_j}(0) = \alpha_{ij}$$

Globally Inverse Optimal: $u = -r^{-1}(x)\delta_n z_n$

$$J = \int_0^\infty \left[q(x) + r(x)u^2 - \gamma^2 w'w \right] dt$$

$r(x)$, $q(x)$ – **a posteriori constructed!**

Desire Local $\mathcal{H}_\infty$-Optimality: $\dot{x} = Ax + B_1 w + B_2 u$

Requires $r(0) = R$, $q_{xx}(0) = Q$, $B_i = G_i(0)$

$R > 0$, $Q = Q' > 0$ – **a priori specified**

Solution: Cholesky factorization

$$PA + A'P + P\left(\gamma^{-2}B_1B_1' - B_2R^{-1}B_2'\right)P + Q = 0$$

$$P = L' \begin{bmatrix} \delta_1 & & 0 \\ & \ddots & \\ 0 & & \delta_n \end{bmatrix} L, \qquad L = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ -\alpha_{11} & 1 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ -\alpha_{n-1,1} & \cdots & -\alpha_{n-1,n-1} & 1 \end{bmatrix}$$

Result: **Locally $\mathcal{H}_\infty$-optimal, globally inverse optimal**

Stability region of an H$_\infty$ design

$$\dot{x}_1 = x_1^2 + x_2 + w$$
$$\dot{x}_2 = u$$

Linearized system:

$$\dot{x}_1 = x_2 + w$$
$$\dot{x}_2 = u$$

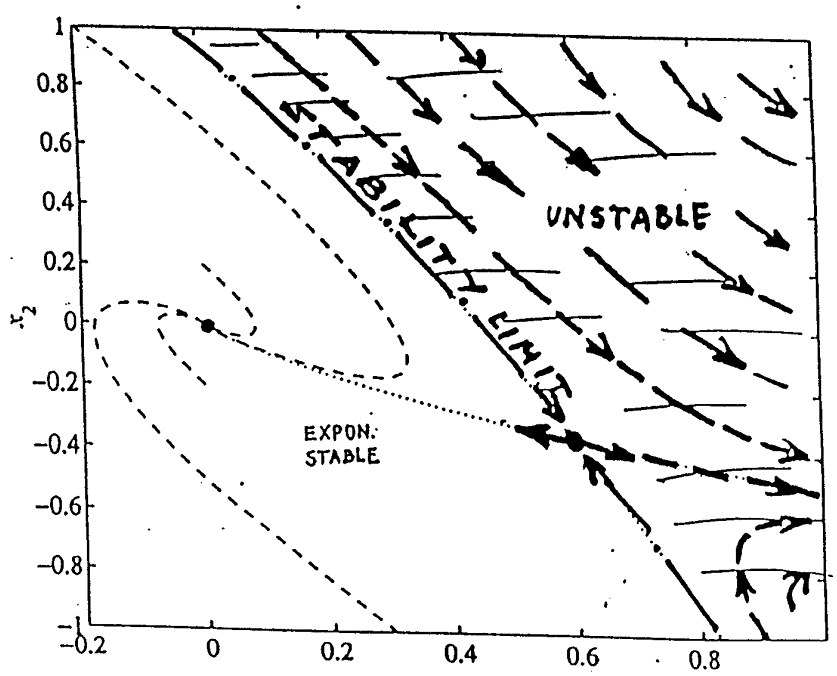
Cost functional:

$$J = \int_0^\infty \left[x_1^2 + x_2^2 + u^2 - \gamma^2 w^2\right] dt$$

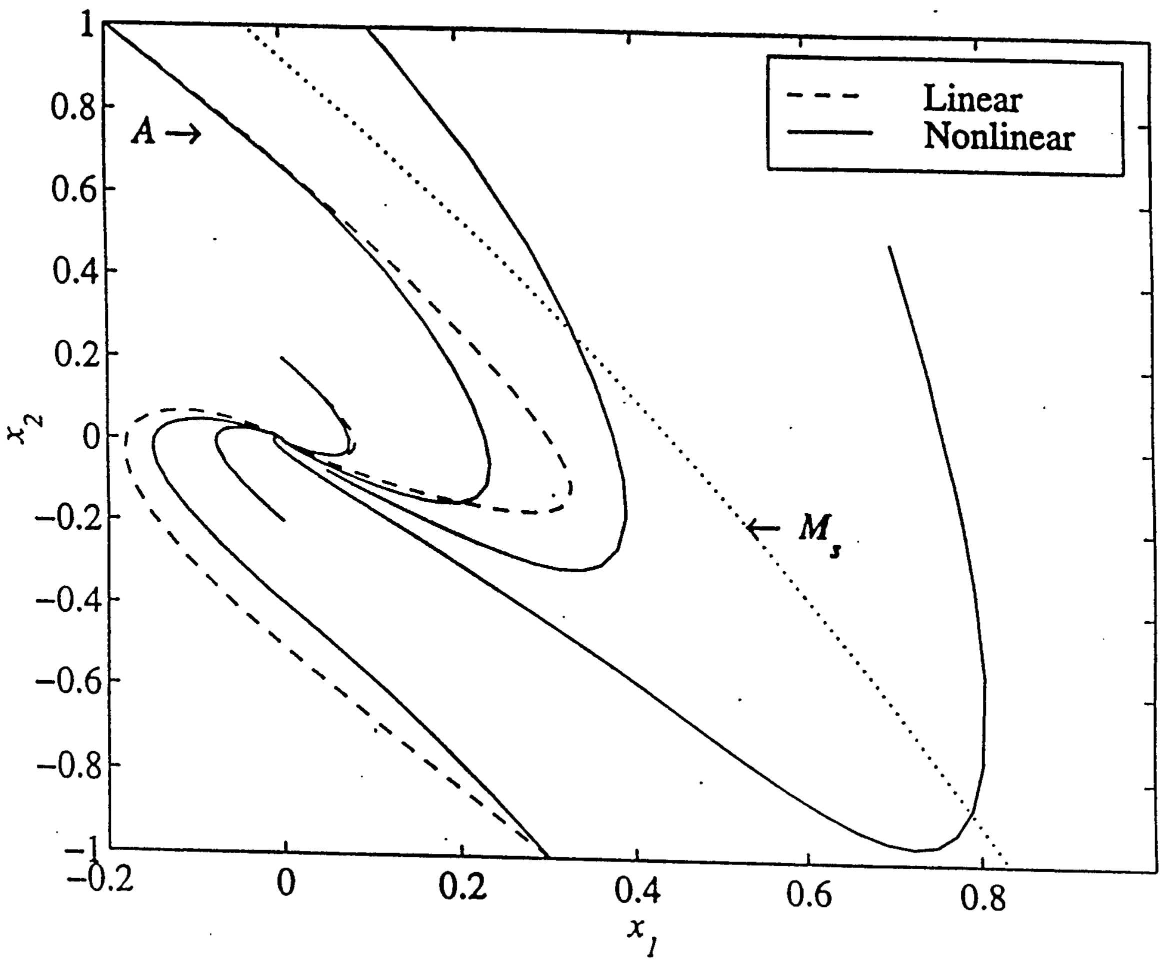
Desired attenuation: $\gamma = 5 > \gamma^* = 1.27$

Optimal linear control:

$$u = -B'Px = -1.06x_1 - 1.78x_2$$

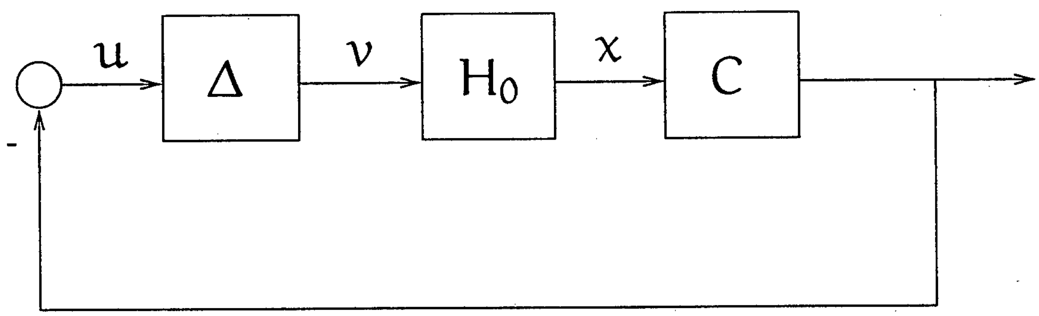


Global Asymptotic Stability of Nonlinear Locally Optimal Design ($w \equiv 0$)



Passivation Redesign

Arcak, Seron, Braslavsky and PK (1998)



$\boxed{H_0}$ Nominal plant

$$\dot{\chi} = \Phi(\chi) + \Gamma(\chi)\nu$$

$\boxed{\Delta}$ Input unmodeled dynamics

$$\begin{aligned} \dot{\xi} &= f_\Delta(\xi, u) \\ \nu &= h_\Delta(\xi, u)\,. \end{aligned}$$

Δ is 0-GAS and $\exists$ pos.def. $\Pi(\xi)$:

$\dot{\Pi}(\xi) \leq -\lambda u^2 + \nu u, \quad \lambda > 0$ (excess of passivity).

This includes stable linear $\Delta(s)$ with

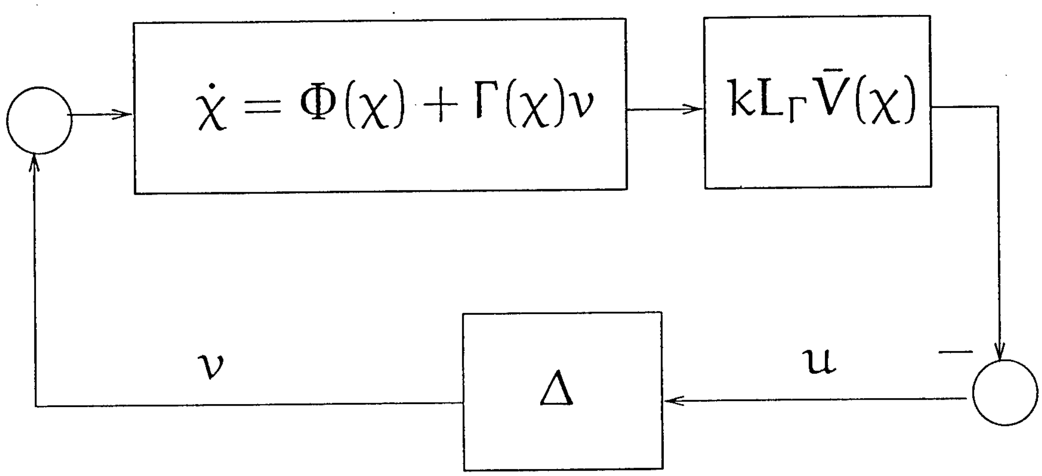
$$\mathrm{Re}\{\Delta(j\omega)\} \geq \lambda.$$

Find pos.def. $\bar{V}(\chi)$ such that

$$\boxed{L_\Phi \bar{V}(\chi) < (L_\Gamma \bar{V}(\chi))^2 \quad \forall \chi \neq 0.}$$

Then, $\boxed{u = -k L_\Gamma \bar{V}(\chi) \quad k \geq \frac{1}{\lambda}} \Rightarrow$ GAS.

Proof: Shortage vs. excess of passivity



Shortage of passivity $1/k$:

$$k\dot{\bar{V}} = kL_\Phi \bar{V} + kL_\Gamma \bar{V}\nu < \underbrace{k(L_\Gamma \bar{V})^2}_{u^2/k} + \underbrace{kL_\Gamma \bar{V}\nu}_{-u\nu}$$

Excess of passivity λ: $\quad \dot{\Pi} \leq -\lambda u^2 + \nu u$

$$U(\chi, \xi) = k\bar{V}(\chi) + \Pi(\xi) \;\Rightarrow\; \dot{U} < 0 \quad \forall \chi \neq 0.$$

$\chi \equiv 0 \Rightarrow u \equiv 0 \Rightarrow \xi = 0$ is the largest invariant set by 0-GAS of Δ.

When does such $\bar{V}(\chi)$ exist?

Iff $\exists$ CLF $V(\chi)$ such that

$$\limsup_{\chi \to 0} \frac{L_\Phi V(\chi)}{(L_\Gamma V(\chi))^2} < +\infty. \qquad \text{(L)}$$

Construction of $\bar{V}(\chi)$:

Find positive $\theta(\cdot)$ such that

(i) $\frac{L_\Phi V(\chi)}{(L_\Gamma V(\chi))^2} < \theta(V(\chi))$,

(ii) $\lim_{t \to \infty} \int_0^t \theta(s) ds = +\infty$.

Then, $$\boxed{\bar{V}(\chi) = \int_0^{V(\chi)} \theta(s) ds}$$

Lemma When H_0 is strict feedback (SF), our backstepping CLF $V_n(X, x_1, \cdots, x_n)$ constructed from $V_0(X_0)$ with

$$L_{F_0 + G_0 \alpha_0} V_0(X_0) \leq -\sigma_0(X_0)$$

satisfies (L) if V_0 and σ_0 have pos.def. Hessians.

Passivation Redesign Example

$$\begin{aligned}
\dot{X} &= X^3 + x \\
\dot{x} &= \nu \\
\nu &= \frac{12(s+35)(s+20)}{7(s+40)(s+30)} u = \Delta(s)u.
\end{aligned}$$

Excess of passivity $\quad \mathrm{Re}\{\Delta(j\omega)\} \geq \lambda = 1.$

$$V_0(X) = X^2, \quad \alpha_0(X) = -X - X^3, \quad \sigma_0(X) = 2X^2$$

$$V_1(X, x) = X^2 + (x + X + X^3)^2$$

$$\frac{L_\Phi V_1(X, x)}{(L_\Gamma V_1(X, x))^2} < \frac{3}{2} + \frac{9}{4} V_1^2(X, x)$$

Choose $\quad \theta(V_1) = \frac{3}{2} + \frac{9}{4} V_1^2, \quad k = \frac{1}{\lambda} = 1,$

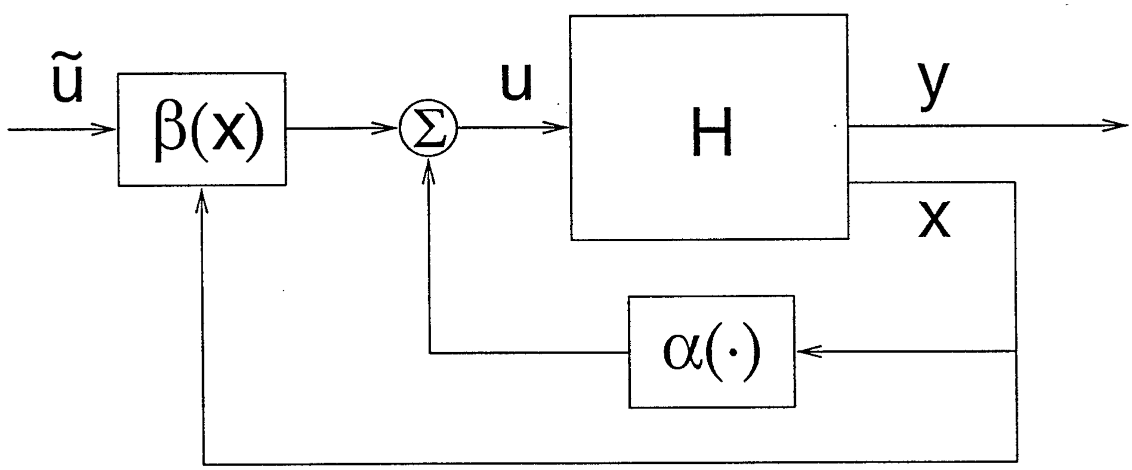
$$u = -k L_\Gamma \bar{V} = -\theta(V_1) L_\Gamma V_1 = -2\theta(V_1)(x + X + X^3).$$

Feedback Passivation (FPR) Design

In design we use state feedback

$$u(x) = \alpha(x) + \beta(x)\tilde{u}$$

to make H passive for the input-output pair $(\tilde{u}, y)$:



This is possible if

- H is **relative degree one**:

u appears in the $\dot{y}$-equation,

- H is (weakly) **minimum phase**:

the "**zero dynamics**" which remain when the output is kept at zero, $y(t) \equiv 0$, are **stable**.

Observer-based FPR Design

Larsen and PK (1999)

$$\begin{aligned}\dot{x} &= f_0(x,0) + g(x,y_2)y_1\\ \dot{\xi} &= A\xi + Bu\\ y_1 &= C_1\xi \qquad y_2 = C_2\xi\end{aligned}$$

- Design an asymptotically stable observer

$$\begin{aligned}\dot{\hat{\xi}} &= A\hat{\xi} + Bu + L(y_2 - \hat{y}_2)\\ \hat{y}_2 &= C_2\hat{\xi}\end{aligned}$$

$A_o^T M + M A_o = -I$, $A_o := A + LC_2$, $M > 0$.

- Design a state feedback gain K to stabilize the linear part of the system with $A_k := A - BK$ Hurwitz and (A_k, B, C_1) passive by solving

$$\begin{aligned}A_k^T P + P A_k &= -Q \leq 0 \quad P > 0\\ PB &= C_1^T.\end{aligned}$$

- Design a stabilizing u from the CLF

$$W(x,\xi,e) = V(x) + \frac{1}{2}\xi^T P\xi + \frac{1}{\mu}e^T Me.$$

Observer-based FPR design

With the feedback $u = -K\hat{\xi} + w$ and observer error $e = \xi - \hat{\xi}$ the system is

$$
\begin{aligned}
\dot{x} &= f_0(x,0) + g(x,y_2)y_1 \\
\dot{\xi} &= A_k\xi - BKe + Bw \\
\dot{e} &= A_o e
\end{aligned}
$$

$$
\dot{W} = L_{f_0}V - \frac{1}{2}\xi^T Q\xi - \frac{1}{\mu}e^T e - \xi^T PBKe + \xi^T C_1^T (L_gV)^T + \xi^T PBw.
$$

Choose $w = -(L_gV)^T - B^T P\hat{\xi}$ and let $N := B^T P - K$:

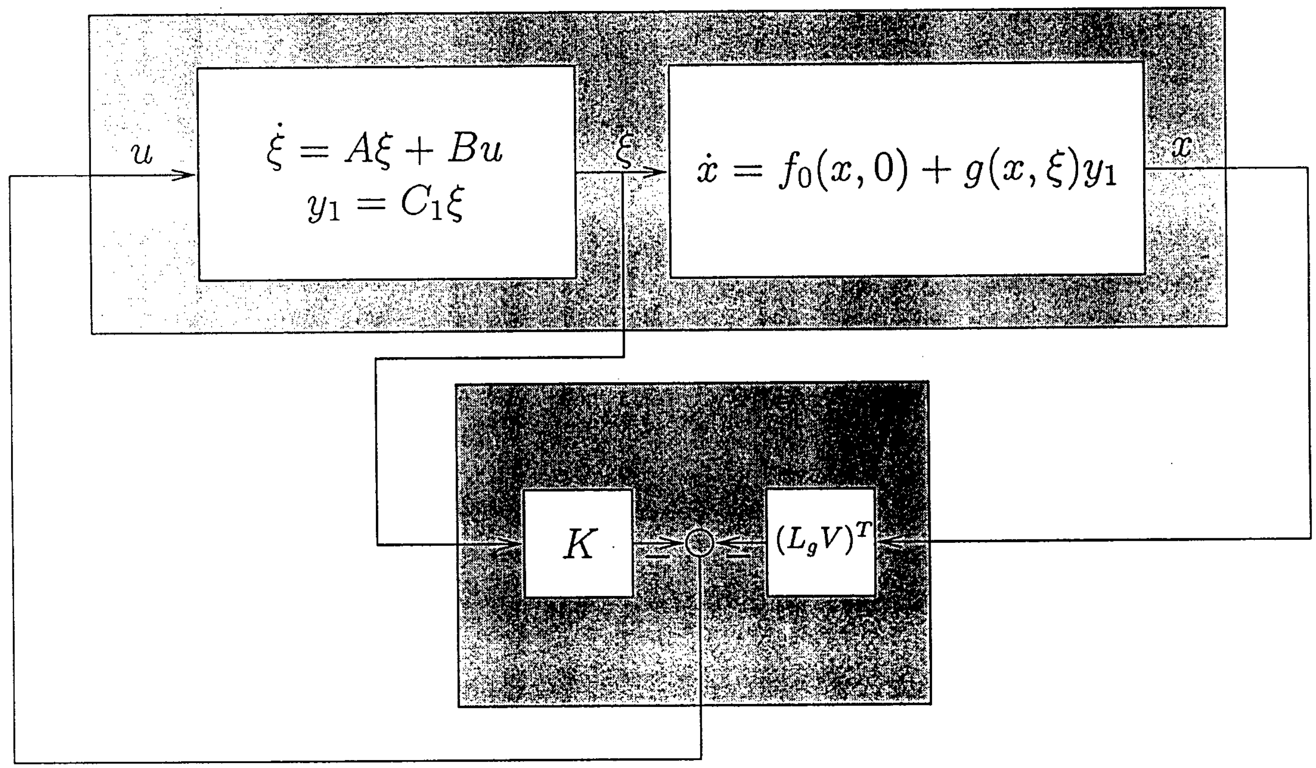
$$
\begin{aligned}
\dot{W} &= L_{f_0}V - \frac{1}{2}\xi^T Q\xi - \frac{1}{\mu}e^T e + \xi^T PBNe - \xi^T PBB^T P\xi \\
&= L_{f_0}V - \frac{1}{2}\xi^T Q\xi - (B^T P\xi - \frac{1}{2}Ne)^T(B^T P\xi - \frac{1}{2}Ne) - (\frac{1}{4}N^T N + \frac{1}{\mu}I)e^T e \\
&\le -\xi^T Q\xi \le 0.
\end{aligned}
$$

GAS with

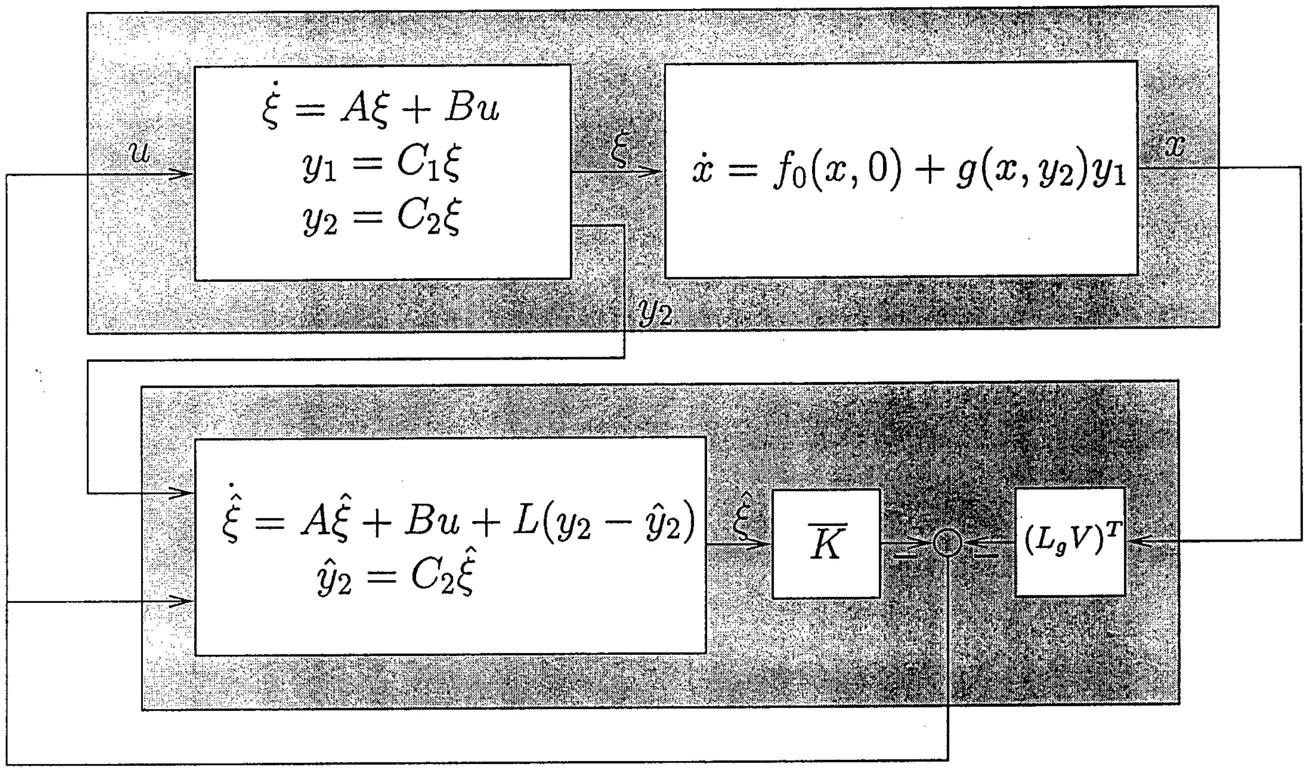
$$
u = -\overline{K}\hat{\xi} - (L_gV)^T \qquad \overline{K} = K + B^T P
$$

follows from the LaSalle Invariance Principle using the detectability of (A_k, Q).

State feedback FPR design



Observer feedback FPR design



Passive Observer Design

Arcak and PK (1999)

$$\dot{x} = Ax - \psi(x) + \gamma(y,u)\,, \quad y = Cx\,,$$

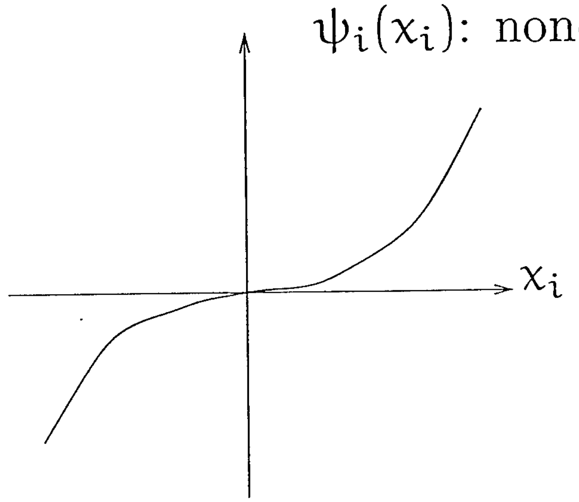
Observer:

$$\dot{\hat{x}} = A\hat{x} - \psi(\hat{x}) + \gamma(y,u) + L(y - C\hat{x})$$

Observer gain matrix L to be designed for

$$\psi(x) = \begin{bmatrix} \psi_1(x_1) \\ \psi_2(x_2) \\ \vdots \\ \psi_n(x_n) \end{bmatrix}$$

$\psi_i(x_i)$: nondecreasing (ND)



(ND) $\Rightarrow$ $\quad \sigma\,[\psi_i(x_i) - \psi_i(x_i - \sigma)] \geq 0$ $\quad$ (*sector*)

Error dynamics:

$$\dot{e} = (A - LC)e - (\psi(x) - \psi(\hat{x}))$$

Indicator matrix K locates the nonlinearities:

$$K = \mathrm{diag}(k_1, \cdots, k_n)$$

$$k_i = \begin{cases} 0 & \text{if } \psi_i(x_i) \equiv 0 \\ 1 & \text{otherwise} \end{cases}$$

Note: $PK = K$ *does not imply* $P = I$.

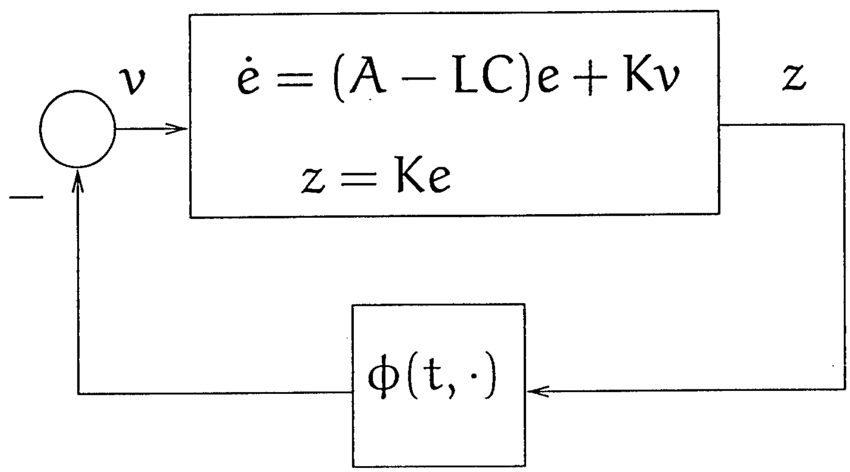
$$z := Ke$$

$$\phi(t, z) := \psi(x) - \psi(\hat{x})$$

$$z^T \phi(t, z) \geq 0 \qquad \text{(sector)}$$

$$\begin{aligned} \dot{e} &= (A - LC)e - K\phi(t, z) \\ z &= Ke\,. \end{aligned}$$

Observer Design



Find $P = P^T > 0$ and L such that

$$(A{-}LC)^T P + P(A{-}LC) \leq 0 \quad \text{and} \quad PK = K\,. \qquad \text{(PR)}$$

Then, whenever $e(t)$ exists, it satisfies

$$|e(t)| \leq k|e(0)|\,.$$

Proof: PR and sector properties.

Control Law Design

$$\begin{aligned} \dot{x} &= Ax - \psi(x) + \gamma(y,u)\,, \quad y = Cx\,, \\ \dot{\hat{x}} &= A\hat{x} - \psi(\hat{x}) + \gamma(y,u) + L(y - C\hat{x})\,. \end{aligned} \qquad \text{(CL)}$$

Find $u = \alpha(y, \hat{x})$ to guarantee

$$|x(t)| \leq \max\left\{ \beta(|x(0)|, t),\ \theta\left(\sup_{\tau\in[0,t)} |e(\tau)|\right) \right\}. \qquad \text{(ISS)}$$

Main Result

If (PR) and (ISS) are satisfied, then (CL) is GS.

If, in addition, $(K, A - LC)$ is detectable and, ψ_i's are either strictly increasing or identically zero, then (CL) is GAS.

Proof:

(PR) implies $|e(t)| \leq k|e(0)|$ and, with (ISS), proves GS. From (PR) and Barbalat's lemma

$$z(t) = Ke(t) \to 0.$$

(CL) is autonomous and $z \equiv 0$ dynamics are $\dot{e} = (A - LC)e$. Detectability implies $e(t) \to 0$ and GAS follows from ISS.

Feasibility

(PR) is an LMI in $L^T P$ and $P = P^T > 0$:

$$\begin{bmatrix} (A-LC)^T P + P(A-LC) & PK - K \\ KP - K & 0 \end{bmatrix} \leq 0 .$$

Feasibility depends on K, the location of $\psi_i(x_i)$.

Example

$$\begin{aligned} y &= x_1 \\ \dot{x}_1 &= x_2 \\ \dot{x}_2 &= -\psi_2(x_2) + \gamma(y) + u . \end{aligned}$$

$$\psi_1(x_1) \equiv 0 \;\rightarrow K = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}$$

(PR) satisfied with $P = I$, $L = [\,1 \;\; 1\,]$.

Evaluate $\dot{V}$ for $V = \frac{1}{2}e_1^2 + \frac{1}{2}e_2^2$ along

$$\begin{aligned} \dot{e}_1 &= -e_1 + e_2 \\ \dot{e}_2 &= -e_1 - (\psi_2(x_2) - \psi_2(\hat{x}_2)). \end{aligned}$$

$$\dot{V} = -e_1^2 - e_2(\psi_2(x_2) - \psi_2(x_2 - e_2)) \leq -e_1^2 .$$

Example: Ship Steering

Arcak, Fossen and PK (1999)

$$\begin{aligned} \dot{\eta} &= J(\eta)\nu \\ M\dot{\nu} &= -D(\nu)\nu - g(\eta) + \tau . \end{aligned}$$

$$D(\nu)\nu = D_0\nu + \begin{bmatrix} \delta_1(\nu_1) \\ \vdots \\ \delta_6(\nu_6) \end{bmatrix}$$

$$D_0 = D_0^T \geq 0 , \quad \delta_i(\cdot)\text{'s are ND.}$$

Observer:

$$\begin{aligned} \dot{\hat{\eta}} &= J(\eta)\hat{\nu} + (\eta - \hat{\eta}) , \\ M\dot{\hat{\nu}} &= -D(\hat{\nu})\hat{\nu} - g(\eta) + \tau + J^T(\eta)(\eta - \hat{\eta}) \end{aligned}$$

$$V(\tilde{\eta}, \tilde{\nu}) = \frac{1}{2}\tilde{\eta}^T\tilde{\eta} + \frac{1}{2}\tilde{\nu}^T M\tilde{\nu} ,$$

$$\begin{aligned} \dot{V} &= -\frac{1}{2}\tilde{\eta}^T\tilde{\eta} - \frac{1}{2}\tilde{\nu}^T D_0\tilde{\nu} - \tilde{\nu}^T[\delta(\nu) - \delta(\nu - \tilde{\nu})] \\ &\leq -\frac{1}{2}\tilde{\eta}^T\tilde{\eta} - \frac{1}{2}\tilde{\nu}^T D_0\tilde{\nu} . \end{aligned}$$

Observer Based Control Design

$$\begin{aligned} y &= x_1 \\ \dot{x}_1 &= x_2 + x_1^2 \\ \dot{x}_2 &= x_3 - x_2(1 + x_2^2) + u \\ \dot{x}_3 &= 2u. \end{aligned}$$

$$A = \begin{bmatrix} 0 & 1 & 0 \\ 0 & -1 & 1 \\ 0 & 0 & 0 \end{bmatrix} \qquad C = [\,1 \;\; 0 \;\; 0\,]$$

$$\psi(x) = \begin{bmatrix} 0 \\ x_2^3 \\ 0 \end{bmatrix} \qquad \gamma(y, u) = \begin{bmatrix} y^2 \\ u \\ 2u \end{bmatrix}$$

$$\rightarrow K = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

(PR) satisfied with $L = [\,3 \;\; 3 \;\; 1\,]^T$.

ISS Control Law Design by Backstepping:

$$\begin{aligned} y &= x_1 \\ \dot{x}_1 &= x_2 + x_1^2 \\ \dot{x}_2 &= x_3 - x_2(1 + x_2^2) + u \\ \dot{x}_3 &= 2u\,. \end{aligned}$$

Relative degree two + ISS zero dynamics

$\longrightarrow$ two steps of observer backstepping:

Step 1. $\quad \dot{x}_1 = x_2 + x_1^2$

$$\alpha_1(x_1) = -x_1^2 - x_1 - n_1 x_1\,, \quad n_1 > 0.$$

$$\dot{x}_1 = -x_1 - n_1 x_1 + e_2 + (\hat{x}_2 - \alpha_1)$$

Step 2. $\quad z_2 := \hat{x}_2 - \alpha_1(x_1) = x_2 - \alpha_1(x_1) + e_2$

$$\dot{z}_2 = u + q(x_1, \hat{x}) - \frac{\partial \alpha_1}{\partial x_1} e_2$$

$$\boxed{u = \alpha_2(x_1, \hat{x}) = -q(x_1, \hat{x}) - x_1 - z_2 - n_2 \left(\frac{\partial \alpha_1}{\partial x_1}\right)^2 z_2}$$

Proof of ISS:

$$
\begin{aligned}
\dot{x}_1 &= -x_1 - n_1 x_1 + z_2 + e_2 \\
\dot{z}_2 &= -x_1 - z_2 - n_2 \left(\frac{\partial \alpha_1}{\partial x_1}\right)^2 z_2 - \frac{\partial \alpha_1}{\partial x_1} e_2
\end{aligned}
$$

ISS Lyapunov function: $V(x_1, z_2) = x_1^2 + z_2^2$

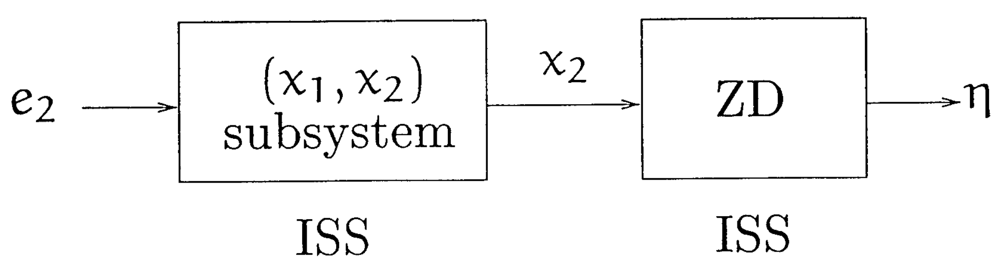
$\rightarrow (x_1, x_2)$ subsystem is ISS with input e_2.

ISS of zero dynamics:

$$\eta := x_3 - 2x_2$$

$$\dot{\eta} = -2\eta - 2x_2 + 2x_2^3 \qquad \text{(ZD)}$$

Cascade of ISS systems is ISS:



Splines and Optimal Control

Fátima Silva Leite

Departamento de Matemática - Universidade de Coimbra, Portugal

and

Instituto de Sistemas e Robótica - Pólo de Coimbra, Portugal

Visiting the Systems Science and Engineering Research Center

Arizona State University, USA

Collaborators:

- P. Crouch - Arizona State University, USA
- M. Camarinha - Instituto de Sistemas e Robótica-Pólo de Coimbra, Portugal
- R. Rodrigues - Instituto de Sistemas e Robótica-Pólo de Coimbra, Portugal
- H. Maurer - Munster University, Germany
- G. Kun - University of Aachen, Germany

- Summary:
  - Motivation/applications
  - Generalized splines on Euclidean spaces
  - Relations with linear optimal control
  - Splines on curved spaces
  - The variational approach
  - Connections with nonlinear optimal control
  - The Hamiltonian approach
  - A geometric algorithm to construct Euclidean and non-Euclidean splines
  - Conclusion

The theory of splines has been a useful mathematical technology in such areas as approximation theory, numerical analysis and, more recently, in computer-aided geometric design.

- A spline function (curve) interpolates a set of points in space, has the same expression between those points and is required to be as smooth as possible.

- Spline functions are widely used for practical approximation of functions or more commonly for fitting smooth curves through preassigned points.

- Spline techniques have the advantage over most approximation and interpolation techniques in that they are computationally feasible.

- Polynomial splines, in particular cubic splines, are the most well known.

In recent years there have been significant efforts to combine ideas of splines and control theory

- The tracking problem in which certain dynamic variables of a control system are forced to follow a desired path, is a major problem in theory and practice.

- Another way of attacking the problem is to specify the trajectory in terms of a discrete, ordered set of points through which the dynamic variables must pass. It is natural to impose smoothness constraints on the trajectories. The objective then becomes to determine suitable controls which give rise to such trajectories. This has been called the dynamic interpolation problem.

(Crouch, Jackson - 1990).

The relationship with optimal control arises when one also requires minimizing natural costs associated with the controls.

Motivations and Applications:

- Air traffic control
- Path-planning for mobile robots

  (car-like autonomous vehicles which navigate over planar surfaces)
- Path-planning for autonomous vehicles which navigate in $3D$

  (robotic air and ocean vehicles).
- Spherical motions.

Difficulties:

- Configuration spaces for mechanical systems are usually non-Euclidean

  Rotation group SO(3)

  Groups of Euclidean motions SE(2)/SE(3)

  SUP(n))

- Equations describing motion might be highly nonlinear.

- Nonholonomic constraints (the number of actuated degrees of freedom is less than the dimension of the configuration space).

- Notations:

For a real-valued functions s, defined on the time interval $[0, T]$,

$$D^m s = s^{(m)} = \frac{d^m s}{dt^m}$$

$$s \in K^m[0,T] \iff \begin{cases} s \in \mathcal{C}^{m-1}[0,T] \\ D^{m-1}s \quad \text{absolutely continuous on } [0,T] \\ D^m s \in \mathcal{L}_2[0,T] \end{cases}$$

$\Delta :\ 0 = t_0 < t_1 < \cdots < t_{n-1} < t_n = T$ a partition of the interval $[0, T]$

$L \equiv D^m + b_m D^{m-1} + \cdots + b_2 D + b_1$ a linear differential operator

$L^* \equiv (-1)^m D^m + (-1)^{m-1} b_m D^{m-1} + \cdots - b_2 D + b_1$ the adjoint of L

Generalized splines in Euclidean Space

A real function s is a <u>generalized spline</u> for the partition Δ of the time interval $[0, T]$, if the following holds simultaneously:

$$\left.\begin{array}{l} s \in K^{2m}[t_{k-1}, t_k], \ k = 1, 2, \ldots, n-1; \\ s \in \mathcal{C}^{2m-2}[0, T]; \end{array}\right\} \Longleftarrow \text{ smoothness conditions}$$

$$L^* L s(t) = 0, \quad \text{on each subinterval}$$

$$\left. D^i s(0) = \beta_{i,0}, \ D^i s(T) = \beta_{i,n}, \ i = 0, 1, \ldots, m-1 \right\} \Longleftarrow \text{ boundary conditions}$$

$$\left. s(t_k) = \alpha_k, \ k = 1, 2, \ldots, n-1 \right\} \Longleftarrow \text{ interpolation conditions}$$

(α_k, $\beta_{i,0}$, $\beta_{i,n}$, $\forall i, k$, are given real constants)

Theorem *[Ahlberg Nilson and Walsh, 1967]*

Given the differential operator L and the partition Δ of the time interval $[0, T]$, there exists a unique generalized spline, for each set of boundary and interpolation conditions.

This generalized spline also minimizes the functional

$$J(f) = \int_0^T (Lf(t))^2 \, dt,$$

among all functions belonging to $K^m[0, T]$ and satisfying the same boundary and interpolation conditions.

To find this unique spline it is enough to determine $2mn$ unknowns, corresponding to $2m$ arbitrary constants in the general solution of the differential equation $L^*Ls(t) = 0$, for each subinterval of the partition Δ.

The required interpolation conditions provide $n - 1$ equations,

the boundary conditions generate $2m$ equations,

the smoothness requirements give rise to $(n-1)(2m-1)$ equations,

leading to a total of $2mn$ linear algebraic equations in the $2mn$ unknowns.

Solving this system is all that one needs to determine the corresponding generalized spline.

An optimal control problem with interpolation conditions

Given:
$$\begin{cases} \text{distinct points } x_0, x_T \text{ in the state space;} \\ n-1 \text{ real constants } \alpha_k,\ k = 1, 2, \ldots, n-1; \\ \text{a partition of the interval } [0,T],\ \Delta:\ 0 = t_0 < t_1 < \cdots < t_n = T; \\ \mathcal{U} = \{u:\ u \in \mathcal{C}^{m-2}[0,T],\ \text{and } u_{|[t_{k-1},t_k]} \in \mathcal{C}^m[t_{k-1}, t_k]\}, \end{cases}$$

$$\min_{u \in \mathcal{U}} \ J(u) = \int_0^T u^2(t)\, dt$$

subject to:

$$\begin{cases} \dot{x}(t) = Ax(t) + bu(t) \\ y(t) = [\,1 \ \ 0 \ \cdots \ 0\,]x(t) \end{cases} \qquad (A, b) \textit{ in controllability canonical form}$$

$$x(0) = x_0, \quad x(T) = x_T;$$

$$y(t_k) = \alpha_k, \quad k = 1, 2, \ldots, n-1.$$

(*Martin, Enqvist, Tomlinson, Zhang - 1995*), (*Rodrigues, Silva Leite, Simões - 1999*)

- The output function corresponding to the previous optimal control problem is a generalized spline.

- All generalized splines may be obtained as optimal output functions.

- Splines and Optimal Control are manifestations of the same phenomena.

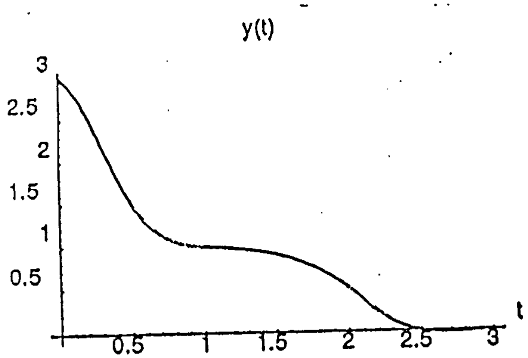
Examples of generalized splines - the bidimensional case

Time interval $[0, 3]$, partition $\Delta: \ t_0 = 0 < 1/2 < 1 < 2 < 9/4 < 3 = t_5$,

Interpolation conditions:

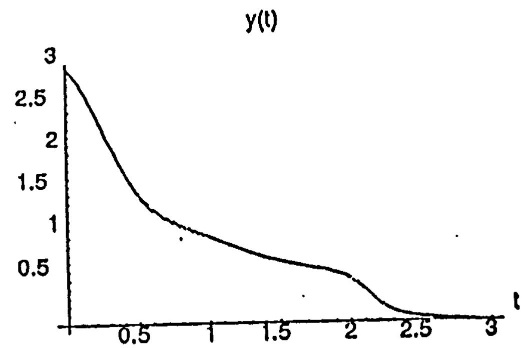
$$y(0) = 3, \quad y(1/2) = 3/2, \ y(1) = 1,$$
$$y(2) = 1/2, \ y(9/4) = 1/6, \ y(3) = 0$$

Boundary conditions: $\dot{y}(0) = -1, \ \dot{y}(3) = 0.$



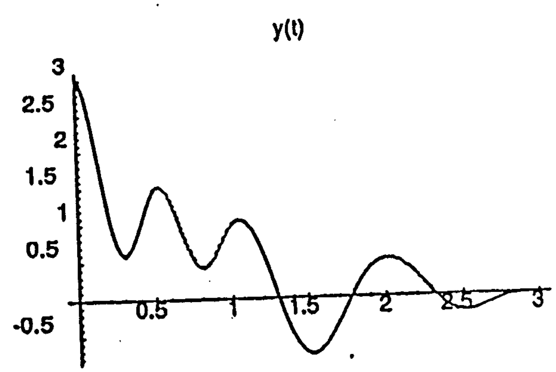
Cubic spline

$\sigma(A) = \{0\}$



Exponencial spline

$\sigma(A) = \{1, -10\}$



Trigonometric spline

$\sigma(A) = \{\pm 7i\}$

- The optimal control problem with arbitrary output function is similarly related to another class of splines, the L-splines. This case allows uneven smoothness conditions at the interpolating points.

- m-dimensional splines ($s : [0, T] \rightarrow \mathbb{R}^m$) are not well studied, but we expect the same kind of connection between them and the output function for MIMO optimal control problems with interpolation conditions.

- What about splines on non-Euclidean spaces? Can we relate nonlinear optimal control problems evolving on Lie groups or spheres, with splines on these non-Euclidean spaces?

- Cubic spline in $I\!R^m$ - a variational approach

$$\min_{x \in \mathcal{C}^2[0,T]} \quad J(x) = \int_0^T < \frac{d^2x}{dt^2}, \frac{d^2x}{dt^2} > dt$$

subject to:

$$x(0) = x_0, \quad x(T) = x_T, \quad \dot{x}(0) = v_0, \quad \dot{x}(T) = v_T, \qquad \textit{(boundary conditions)}$$

$$x(t_k) = x_k, \quad k = 1, 2, \ldots, n-1. \qquad \textit{(interpolation conditions)}$$

The Euler-Lagrange equation associated with this problem is

$$\frac{d^4x}{dt^4} = 0, \qquad \text{on each subinterval } [t_{k-1}, t_k].$$

and, indeed, the cubic spline minimizes the functional, among all the functions belonging to $\mathcal{C}^2[0,T]$ and satisfying the same boundary and interpolation conditions.

Splines on curved spaces - A variational approach

(Riemannian geometry is now the mathematical machinery)

M is a Riemannian manifold equipped with a Riemannian metric $< \cdot, \cdot >$.

$\frac{D}{dt}$ denotes covariant derivative along curves x on M.

- The cubic spline

$$\min_{x \in C^2[0,T]} J(x) = \int_0^T < \frac{DV}{dt}, \frac{DV}{dt} > dt, \qquad V = \dot{x},$$

subject to:

$$x(0) = x_0, \quad x(T) = x_T, \quad \dot{x}(0) = v_0, \quad \dot{x}(T) = v_T, \qquad \textit{(boundary conditions)}$$

$$x(t_k) = x_k, \quad k = 1, 2, \ldots, n-1. \qquad \textit{(interpolation conditions)}$$

The Euler-Lagrange equation associated with this problem is:

$$\frac{D^3V}{dt^3} + R(\frac{DV}{dt}, V)V = 0, \qquad \text{on each subinterval } [t_{k-1}, t_k].$$

(R is the curvature tensor and measures how curved M is.)

(Noakes, Heinzinger, Paden - 1989), (Crouch, Silva Leite - 1991/1995), (Camarinha, 1996)

- Particular cases:

  - M is a Lie group (For instance, the rotation group $SO(3)$)

$$\dddot{V} + [V, \ddot{V}] = 0, \qquad \text{on each subinterval } [t_{k-1}, t_k].$$

  - M is a symmetric space (For instance, a sphere)

$$\dddot{V} + [V, [\dot{V}, V]] = 0, \qquad \text{on each subinterval } [t_{k-1}, t_k].$$

(double-bracket equations)

Difficulties:

- It remains an open problem to describe solutions of these double-brackett equations.
- Unlike the Euclidean case, solutions of the Euler-Lagrange equation do not necessarily minimize the functional J.
- Addition of constraints on velocities does not create additional difficulties, but may give rise to abnormal extremals.
- Other choices for the Lagrangian give rise to generalized splines on Riemannian manifolds.

(Camarinha, Silva Leite, Crouch - 1995, 1998), (Brunnett, Crouch, Silva Leite - 1996)

- Where are the connections with optimal control?
- It is possible to formulate a number of optimal control problems which are equivalent to the variational problems associated with non-Euclidean splines.

We give examples of optimal control problems for systems evolving on the Lie group $SO(3)$

X_1, X_2 and X_3 are the skewsymmetric matrices defined by:

$$X_1 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix}, \quad X_2 = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ -1 & 0 & 0 \end{bmatrix}, \quad X_3 = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}.$$

The interpolation conditions, as well as the boundary conditions, are always assumed to hold and will be omitted hereafter.

Case I - full control

$$\min_{u(\cdot)} \frac{1}{2} \int_0^T (u_1^2 + u_2^2 + u_3^2)\, dt \quad \text{subject to}$$

$$\dot{x} = v_1 X_1(x) + v_2 X_2(x) + v_3 X_3(x), \quad \dot{v}_i = u_i, \quad i = 1, 2, 3.$$

The extremals satisfy

$$\frac{d^3 v}{dt^3} + v \times \frac{d^2 v}{dt^2} = 0 \text{ where } v = (v_1, v_2, v_3)^T.$$

Case II - nonholonomic constraint, driftless

$$\min_{u(\cdot)} \frac{1}{2} \int_0^T (u_1^2 + u_2^2)\, dt \quad \text{subject to}$$

$$\dot{x} = v_1 X_1(x) + v_2 X_2(x), \quad \dot{v}_1 = u_1, \quad \dot{v}_2 = u_2.$$

The extremals satisfy

$$\begin{cases} \ddot{v}_1 - \lambda_3 v_2 = 0 \\ \ddot{v}_2 + \lambda_3 v_1 = 0 \\ \dot{\lambda}_3 - v_1 \ddot{v}_2 + v_2 \ddot{v}_1 = 0 \end{cases}$$

Case III - nonholonomic constraint, with drift

$$\min_{u(\cdot)} \frac{1}{2}\int_0^T u_2^2 dt \quad \text{subject to}$$

$$\dot{x} = X_1(x) + v_2 X_2(x), \quad \dot{v}_2 = u_2.$$

The extremals satisfy

$$\begin{cases} \ddot{v}_2 + \lambda_3 = 0 \\ \dot{\lambda}_1 + \lambda_3 v_2 = 0 \\ \dot{\lambda}_3 - \ddot{v}_2 - \lambda_1 v_2 = 0 \end{cases}$$

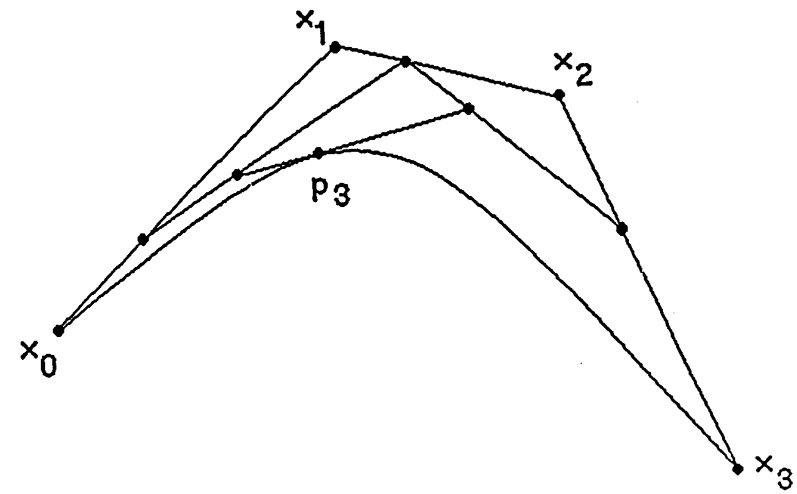
- Theoretically these optimal control problems may be solved alternatively using the maximum principle. But this Hamiltonian approach does not bring any light into the geometry of the extremals.

- The De Casteljau algorithm - a geometric construction of splines

(De Casteljau, Technical Report, Citroen/Paris -1959)

- The classical De Casteljau algorithm is a geometric construction, whereby two points in $\mathbb{R}^m$ are joined by a polynomial through an iterative linear interpolation process.

Illustration for the construction of the cubic polynomial:



First step

$$p_1(t, x_0, x_1) = tx_1 + (1-t)x_0,$$

$$p_1(t, x_1, x_2) = tx_2 + (1-t)x_1,$$

$$p_1(t, x_2, x_3) = tx_3 + (1-t)x_2$$

Second step

$$p_2(t, x_0, x_1, x_2) = tp_1(t, x_1, x_2) + (1-t)p_1(t, x_0, x_1) = t^2x_2 + 2t(1-t)x_1 + (1-t)^2x_0,$$

$$p_2(t, x_1, x_2, x_3) = tp_1(t, x_2, x_3) + (1-t)p_1(t, x_1, x_2) = t^2x_23 + 2t(1-t)x_2 + (1-t)^2x_1,$$

Third step

$$\begin{aligned} p_3(t, x_0, x_1, x_2, x_3) &= tp_2(t, x_1, x_2, x_3) + (1-t)p_2(t, x_0, x_1, x_2) \\ &= t^3x_3 + 4t^2(1-t)x_2 + 3t(1-t)^2x_1 + (1-t)^3x_0, \end{aligned}$$

- Relationship between control points and initial/final velocity/acceleration:

$$p_3(0) = x_0, \qquad p_3(1) = x_3,$$

$$\dot{p}_3(0) = 3(x_1 - x_0), \qquad \dot{p}_3(1) = 3(x_3 - x_2),$$

$$\ddot{p}_3(0) = 6\{(x_2 - x_1) - (x_2 - x_0)\}, \quad \ddot{p}_3(1) = 6\{(x_3 - x_2) - (x_2 - x_1)\}.$$

- Given Hermite conditions [2 points (x_0, x_3) and 2 vectors (v_0, v_3)] find the cubic polynomial $p_3(t)$ that satisfies:

$$p_3(0) = x_0, \quad p_3(1) = x_3, \quad \dot{p}_3(0) = v_0, \quad \dot{p}_3(1) = v_3.$$

- Given [2 points (x_0, x_3) and 2 vectors (v_0, w_0)] find the cubic polynomial $p_3(t)$ that satisfies:

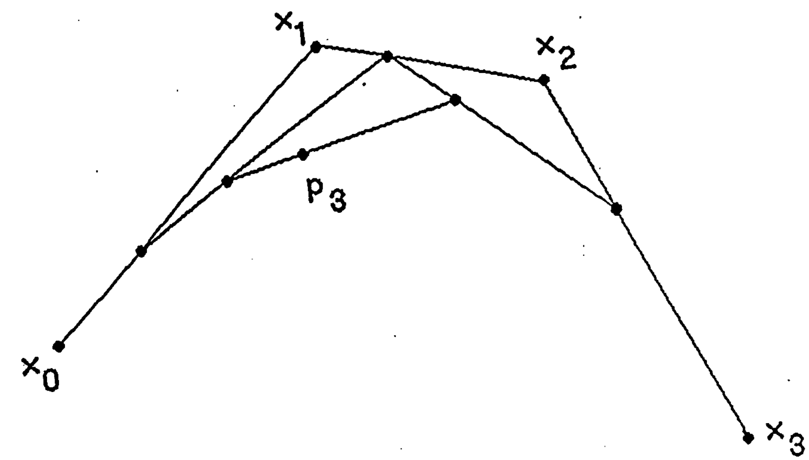
$$p_3(0) = x_0, \quad p_3(1) = x_3, \quad \dot{p}_3(0) = v, \quad \ddot{p}_3(0) = w_0.$$

- The last conditions are more appropriate to construct spline curves.

- The power of this algorithm lies in the fact that it can be easily generalized from $I\!R^m$ to other spaces, as long as the linear interpolation process is suitably redefined.

- Generalized De Casteljau algorithm on Lie groups

(Ge, Ravani -1994), (Park, Ravani -1995), (Crouch, Kun, Silva Leite -1996, 1999)



First step

$$p_1(t, x_0, x_1) = e^{tV_0^1} x_0, \quad \text{where } x_1 = e^{V_0^1} x_0,$$

$$p_1(t, x_1, x_2) = e^{tV_1^1} x_1, \quad \text{where } x_2 = e^{V_1^1} x_1,$$

$$p_1(t, x_2, x_3) = e^{tV_2^1} x_2, \quad \text{where } x_3 = e^{V_2^1} x_2.$$

Second step

$$p_2(t, x_0, x_1, x_2) = e^{tV_0^2} e^{tV_0^1} x_0, \quad \text{where } e^{V_0^2} = e^{tV_1^1} e^{(1-t)V_0^1},$$

$$p_2(t, x_1, x_2, x_3) = e^{tV_1^2} e^{tV_1^1} x_1, \quad \text{where } e^{V_1^2} = e^{tV_2^1} e^{(1-t)V_1^1},$$

Third step

$$p_3(t, x_0, x_1, x_2, x_3) = e^{tV_0^3} e^{tV_0^2} e^{tV_0^1} x_0, \quad \text{where } e^{V_0^3} = e^{tV_1^2} e^{(1-t)V_0^2}.$$

- Relationship between control points and initial/final velocity/cov. acceleration:

$$p_3(0) = x_0, \qquad p_3(1) = x_3,$$

$$\dot{p}_3(0) = 3V_0^1 x_0, \qquad \dot{p}_3(1) = 3V_2^1 x_3,$$

$$\frac{D\dot{p}_3}{dt}(0) = 6\chi_0^{-1}(V_1^1 - V_0^1)x_0, \qquad \frac{D\dot{p}_3}{dt}(1) = 6\chi_1^{-1}(V_2^1 - V_1^1)x_3.$$

where $\chi_0 = \int_0^1 e^{u\, adV_0^1} du, \quad \chi_1 = \int_0^1 e^{-u\, adV_2^1} du.$

(ad is the adjoint operator in the Lie algebra: $ad\, A(B) = [A, B]$.

This is the same as the classical case, except for χ_0 *and* χ_1*!*

- Main difficulties in implementing the De Casteljau algorithm:
  - Exponentiating matrices (in the Lie algebra)
  - Finding logarithms of matrices (in the Lie group)

- For the rotation group $SO(3)$ there are explicit formulas for the exponential and the logarithm: If $S_a \in so(3)$ denotes the skewsymmetric matrix defined by $S_a b = a \times b$, for a and b vectors in $\mathbb{R}^3$ and $\times$ the cross product in $\mathbb{R}^3$, we have

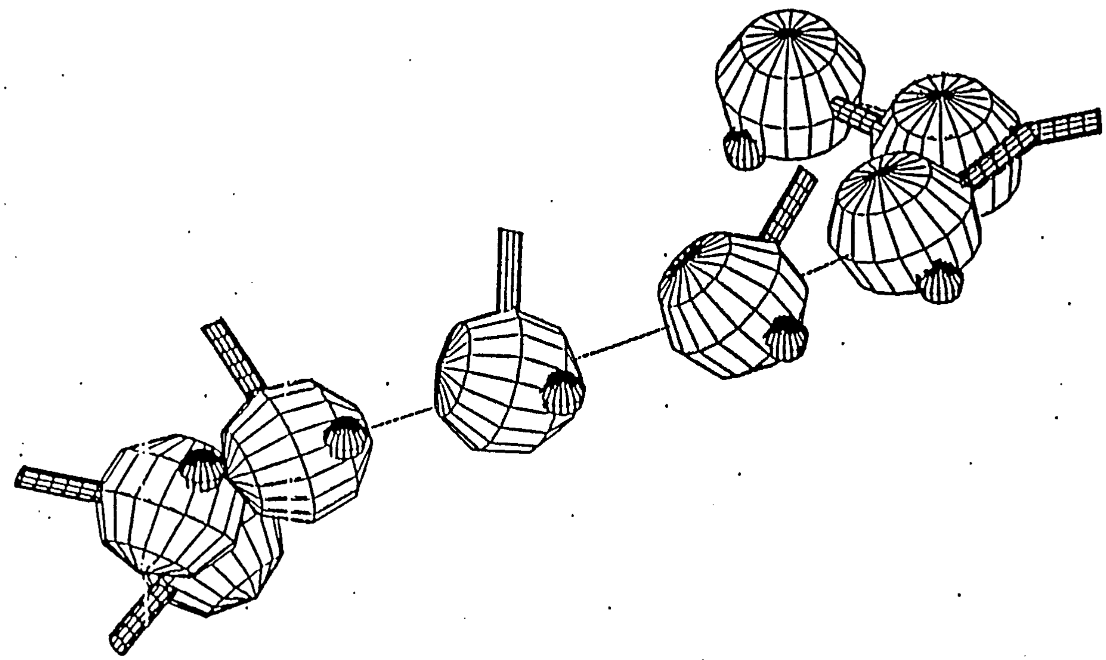
$$\exp(S_a) = I\cos(\| a) + \frac{\sin(\| a \|)}{\| a \|} S_a + \frac{1 - \cos(\| a \|)}{\| a \|^2} a a^T. \qquad \textit{(Rodrigues' formula)}$$

Also, if $x \in SO(3)$, then

$$\log x = \frac{\alpha}{2 \sin \alpha}(x - x^T),$$

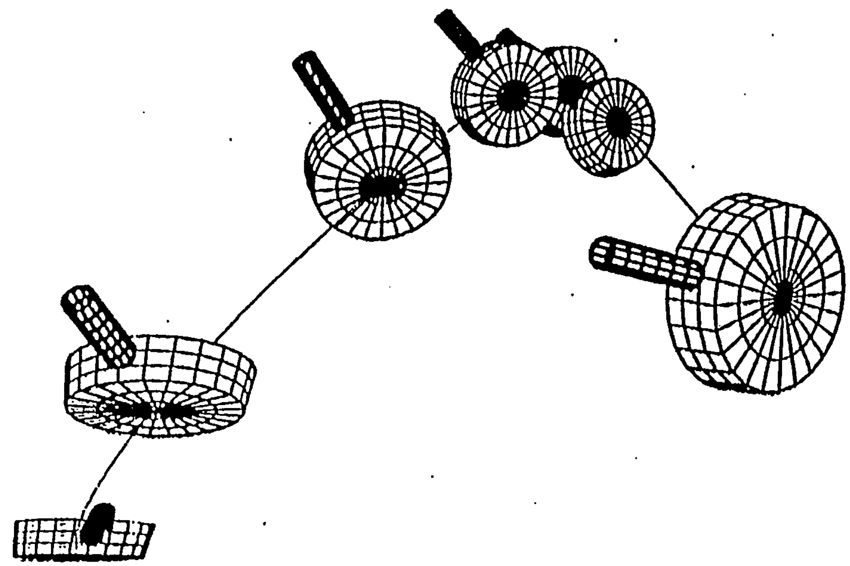
where $\cos \alpha = \dfrac{trace(x) - 1}{2}$.

- So, for SO(3), the De Casteljau algorithm to produce polynomial curves and spline curves is easy to implement.



Animation of a satellite

(*produced with a cubic spline on* $\boldsymbol{SO}(3)$)



The flying pan

(produced with a cubic polynomial on $SO(3)$*)*

- Generalized De Casteljau algorithm on spheres S^m

(Crouch, Kun, Silva Leite - 1999)

Again, the De Casteljau algorithm relies on the ability to compute geodesics on the sphere. The geodesic on S^m that joins two points, x_0 at $(t = 0)$ to x_1 at $(t = 1)$, is given by

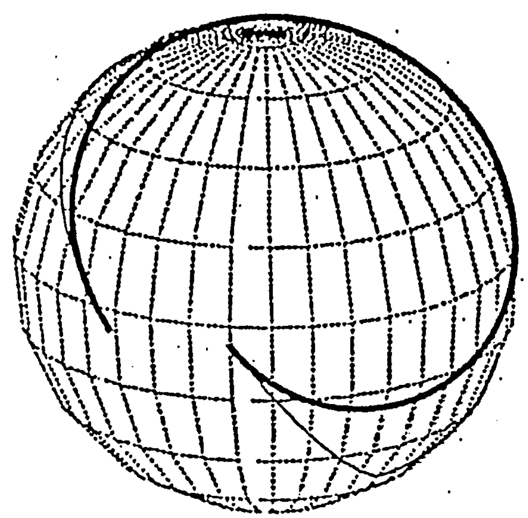
$$p_1(t, x_0, x_1) = \frac{\sin((1-t)\theta)}{\sin\theta} x_0 + \frac{\sin(t\theta)}{\sin\theta} x_1, \qquad \theta = \cos x_0^T x_1^{-1}.$$

where $\theta = \cos x_0^T x_1{}^{-1}$.

This geodesic is the projection onto the sphere of a geodesic on $SO(m+1)$. ($SO(m+1)$ acts transitively on S^m). It turns out however that, for a sphere of any dimension, only matrix exponentials and logarithms in $SO(3)$ need to be computed.

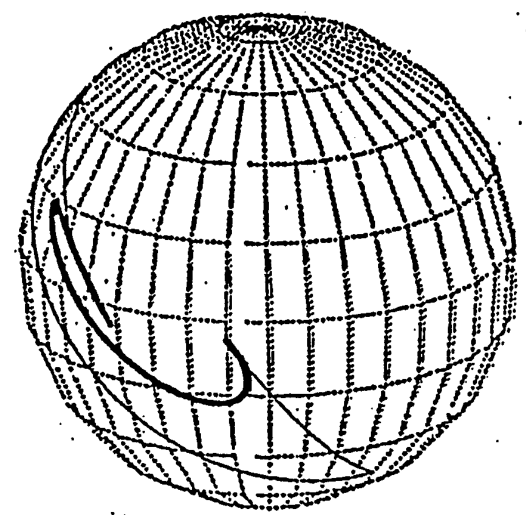
- So, the De Casteljau algorithm on spheres can be easily implemented.

The two dimensional case (S^2) reveals some interesting features, which do not show in Euclidean spaces.



length $\approx$ 5.2

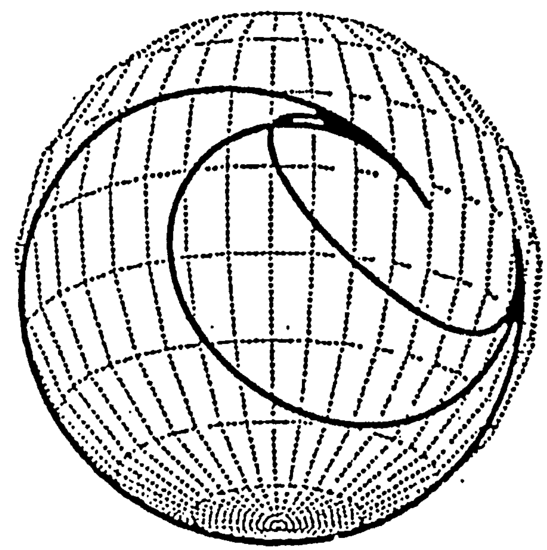
average acceleration $\approx$ 250



length $\approx$ 2.2

average acceleration $\approx$ 1150

Two cubics for the same boundary value problem



The control points are antipodal

- The class of interpolating curves defined by a variational principle or by a Hamiltonian approach does not seem so computationally tractable as those developed by the De Casteljau algorithm.
- Only for abelian Lie groups the curves produced by the variational principle are exactly those produced via the De Casteljau algorithm.

- Conclusion
  - Good reasons to study the geometry of splines:
    - The problem of synthesizing a smooth motion of a rigid body or groups of rigid bodies, such as robots, that interpolates a set of configurations in space has considerable importance in many engineering applications.
    - Splines and optimal control (both linear and nonlinear) seem to be manifestations of the same phenomena.