POLITECNICO DI MILANO

LPV/LFT modelling and identification: overview, applications and perspectives

Marco Lovera
Dipartimento di Elettronica e Informazione, Politecnico di Milano

Outline

- Introduction and motivation
- LPV model classes:
  - state-space vs input-output form
  - focus on state-space form
- Overview of LPV state space model identification
- Application: LPV Identification in Virtualized Service Center Environments
- Integrated LFT modelling and identification of physical systems
- Perspectives and conclusions

- Modern methods for robust and gain scheduled controller design call for advanced modelling and identification techniques

- Critical issue: deriving models in which the dependence from operating point information and/or uncertain parameters is explicit

- Linear Parametrically Varying (LPV) models: a useful modelling approach to bridge the gap between identification and controller design

- Major trade-off: black vs grey vs white box

LPV models

"Time varying systems, the dynamics of which are functions of a measurable, time-varying parameter vector δ."

Common assumptions on parameter vector δ:

- Component-wise bounded
- Component-wise rate-bounded

- The equations

$$\boxed{\begin{aligned} x(t+1) &= A(\delta(t))x(t) + B(\delta(t))u(t) \\ y(t) &= C(\delta(t))x(t) + D(\delta(t))u(t) \end{aligned}}$$

$$\begin{aligned} \dot{x} &= A(\delta(t))x + B(\delta(t))u \\ y &= C(\delta(t))x + D(\delta(t))u \end{aligned}$$

describe a whole family of time-varying systems.

- A specific time-varying system is defined once a realisation $\delta(t)$ is chosen.

- A given LPV system can give rise to very different behaviours!

- Affine parameter dependence (LPV-A):

$$A(t) = A_0 + A_1\delta_1(t) + \ldots + A_s\delta_s(t)$$

- Input-affine parameter dependence (LPV-IA):
  - only B and D are function of δ
  - A and C are constant

- Rational parameter dependence (LPV-R):

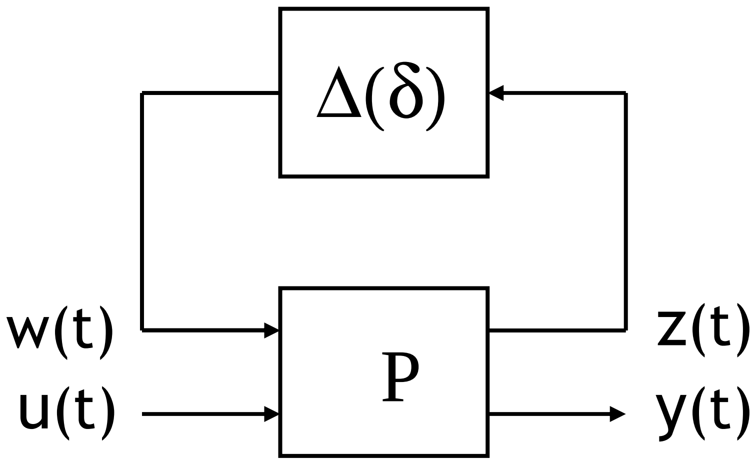
$$\begin{aligned} A(t) = &[A_{n0} + A_{n1}\delta_1(t) + \ldots + A_{ns}\delta_s(t)] \\ &[I + A_{d1}\delta_1(t) + \ldots + A_{ds}\delta_s(t)]^{-1} \end{aligned}$$

Structure of LPV models – state-space form

- Linear fractional representation parameter dependence (LPV-LFR):

$$
\begin{aligned}
x(t+1) &= Ax(t) + B_0 w(t) + B_1 u(t) \\
z(t) &= C_0 x(t) + D_{00} w(t) + D_{01} u(t) \\
y(t) &= C_1 x(t) + D_{10} w(t) + D_{11} u(t)
\end{aligned}
$$

$$
\begin{aligned}
w(t) &= \Delta z(t) \\
\Delta &= \mathrm{diag}\begin{pmatrix} \delta_1 I_{r_1} & \dots & \delta_s I_{r_s} \end{pmatrix}
\end{aligned}
$$



Of course the LFR is potentially much more general.
We will come back to this.

Discrete-time LPV models in input-output form

In the literature, input-output models of the type

$$y(t) = -\sum_{i=1}^{n_a} a_i(\delta(t))y(t-i) + \sum_{j=1}^{n_b} b_i(\delta(t))u(t-j)$$

have been considered, which are parameter-dependent extensions of discrete-time LTI input-output models.

As for state-space models, the a_i's and b_j's can be

- Affine
- Rational
- Linear Fractional functions of the parameter vector δ.

Issues in input-output to state-space conversion

In the LTI case the equivalence between $\left[TAT^{-1}, TB, CT^{-1}, D\right]$

and $G(z) = C(zI - A)^{-1}B + D$
where T is square and non singular, is well known.

This motivates a number of identification methods which

1. Perform an initial (e.g., nonparametric) estimation of the input-output behaviour

2. Refine the initial estimate
   - either directly in state-space form
   - or in input-output form, followed by state-space conversion

LPV models, however, are *time-varying*. In discrete-time:

$$\begin{aligned} x(t+1) &= A(t)x(t) + B(t)u(t) \\ y(t) &= C(t)x(t) + D(t)u(t) \end{aligned} \qquad \tilde{x}(t) = T(t)x(t)$$

$$\begin{aligned} \tilde{x}(t+1) &= \boxed{T(t+1)A(t)T^{-1}(t)}\tilde{x}(t) + \boxed{T(t)B(t)}u(t) \\ y(t) &= \boxed{C(t)T^{-1}(t)}\tilde{x}(t) + \boxed{D(t)}u(t) \end{aligned}$$

- T will be parameter dependent) LTI input-output and state-space equivalence should be used very carefully!
- In particular:
  - The change of state-space basis will depend *locally* on the value of the parameters and on their rate of change
  - The choice of input-output vs state-space models should be based on the eventual goal of the identification exercise, as conversion is not trivial (see Toth 2008).

Two broad classes of methods can be defined:

- Global approaches
  - a single experiment) the parameter is also excited
  - a parameter-dependent model is directly obtained

- Local approaches
  - multiple experiments) constant parameter values
  - many LTI models are obtained, which have to be interpolated

Overview of the literature: global approach

- Input/output models
  - (Bamieh and Giarre',1999 & 2002)
  - (Previdi and Lovera, 2003 & 2004)
  - (Toth *et al.*, 2007 & 2008)
- State space models
  - (Nemani *et al.*, 1995)
  - (Lee and Poolla, 1997 & 1999)
  - (Lovera *et al.*, 1998)
  - (Sznaier and Mazzaro, 2001 & 2003)
  - (Verdult and Verhaegen, 2002)
  - (Felici *et al.*, 2007)
  - (van Wingerden and Verhaegen, 2009)

(Nemani *et al.*, 1995)

- Identification of single input LPV-LFT models with a scalar parameter

- State vector assumed available for measurement

- Both cases of noise free and noisy state measurement are taken into account, together with process noise in the state equation.

- The problem is solved using RLS; the use of IV-RLS is also proposed to deal with non-white measurement noise.

(Lovera *et al.* 1998)

- Identification of MIMO LPV-A models
- No restrictive assumptions on the number of parameters
- Possibly noisy state vector measurement available

$$x_{t+1} = (A_0 + A_1\delta_{1,k} + \ldots + A_s\delta_{s,k})x_t + (B_0 + B_1\delta_{1,k} + \ldots + B_s\delta_{s,k})u_t$$

$$y_t = x_t - e_t$$

- Batch solution using IV least squares

$$\begin{aligned} y_{t+1} + e_{t+1} = {} & A_0(y_t + e_t) + A_1\delta_{1,k}y_t + A_1\delta_{1,k}e_t + \\ & \ldots + A_s\delta_{s,k}y_t + A_s\delta_{s,k}e_t + (B_0 + B_1\delta_{1,k} + \ldots + B_s\delta_{s,k})u_t \end{aligned}$$

(Lee and Poolla, 1997 & 1999)

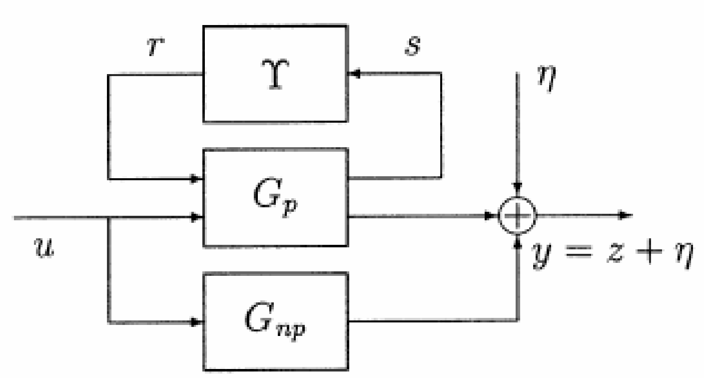
- A maximum likelihood (ML) algorithm for the identification of MIMO LPV-LFT models is proposed
- The algorithm is based on PEM and is strongly related to classical methods for the ML identification of ARMA and ARMAX models
- The computation of the gradient and of the hessian is performed by means of (LPV) filtering operations
- Major issue related to this algorithm: initialisation

(Sznaier and Mazzaro, 2001 & 2003)

Consider a model class of the form

with

$$\mathcal{F}_u(G_p, \Upsilon) = \sum_{i=1}^{N_p} p_i \mathcal{F}_u(G_i, \Upsilon)$$



where

- the N_p $G_i(z)$ transfer functions are known,
- $G_{np}(z)$ is a stable, norm-bounded operator
- η is a bounded measurement noise.

An approach is provided which allows to test consistency (in the form of LMIs) of the *a priori* modelling information with the results (measured y and parameters) of experiments.

(Verdult and Verhaegen, 2002), (Felici *et al.*, 2007), (van Wingerden and Verhaegen, 2009)

- Extensions to LPV systems of classical subspace model identification algorithms for LTI models

- Model class

$$
\begin{aligned}
x_{k+1} &= \left(A_0 + \sum_{i=1}^{s} [p_k]_i A_i\right) x_k + \left(B_0 + \sum_{i=1}^{s} [p_k]_i B_i\right) u_k + w_k \\
y_k &= \left(C_0 + \sum_{i=1}^{s} [p_k]_i C_i\right) x_k + \left(D_0 + \sum_{i=1}^{s} [p_k]_i D_i\right) u_k + v_k
\end{aligned}
$$

Subspace-based methods

- General approach:
  - Construct a data equation relating inputs, outputs and states
  - Estimate the state sequence
  - Reconstruct the state space matrices

- Main issues:
  - Persistency of excitation conditions only partially understood
  - The number of rows in the data matrices grows exponentially with the system order

- Solutions available in the literature:
  - use the RQ factorisation to select the dominant rows in the data matrices and discard the rest
  - use kernel methods to compress the row spaces of the data matrices

- Available methods are reliablebut still limited in terms of problem size.

Two broad classes of methods can be defined:

- Global approaches
  - a single experiment) the parameter is also excited
  - a parameter-dependent model is directly obtained
- Local approaches
  - multiple experiments) constant parameter values
  - many LTI models are obtained, which have to be interpolated

Overview of the literature: local approach

Local approach

- (Steinbuch *et al.* 2003)
- (Paymans *et al.* 2006 & 2008)
- (Toth *et al.* 2007)
- (Lovera and Mercere 2007)

Issues with local approaches:

- Numerical accuracy: poorly conditioned canonical forms used in local problems) ill-conditioning in the interpolation
- Consistency of the interpolation procedure:
  - Input/output form) interpolating transfer function coefficients
  - State space form) consistency of state space basis

The method of Steinbuch et al.

The algorithm (applicable to SISO or MISO models) can be summarised as follows:

1. Local experiments) nonparametric estimates $\widehat{G}(j\omega)$ of the local frequency response
2. Parametric TFs $G(s) = \dfrac{\beta_{n-1}s^{n-1} + \beta_{n-2}s^{n-2} + \cdots + \beta_1 s + \beta_0}{s^n + \alpha_{n-1}s^{n-1} + \cdots + \alpha_1 s + \alpha_0} + \beta_n$ are fitted to the local frequency responses
3. Each TF is converted to Canonical Controllability Form
4. The parameters of the local models are interpolated
5. The model is converted to LFT form

The method of Steinbuch et al.: discussion

- Numerical issues: the CCF is ill conditioned, so the interpolation step will be numerically very sensitive.
- Method restricted to
  - low order models
  - without sensitive poles/zeros (lightly damped complex conjugate poles)
- State space interpolation: no guarantee that all local models are in the same state-space basis.

The method of Paijmans et al.

- Local models are parameterised using poles, zeros and gain and factored into first and/or second order subsystems

$$G_i(s) = \gamma_i \prod_{\tau=1}^{\tau_1} F_i^\tau(s) \prod_{\tau=1}^{\tau_1} S_i^\tau(s), \quad i = 1, \ldots, m$$

- Each local model is decomposed using the following rules:
  - A second order system is created for each pair of c.c.poles
  - All pairs of c.c. zeros are added to existing second order systems
  - For each remaining real pole a first order system is created
  - The remaining real zeros are added to the first and second order subsystems

- Parameter-dependent poles and zeros loci are optimised in order to fit the pole/zero maps of the local models.

The method of Paijmans et al.: discussion

- Limited to SISO systems
- Interpolation step very critical – requires manual intervention
- Constraints introduced to preserve affine parameter dependence: B^{τ} and D^{τ} matrices of local models must be constant.

The method of Lovera and Mercere

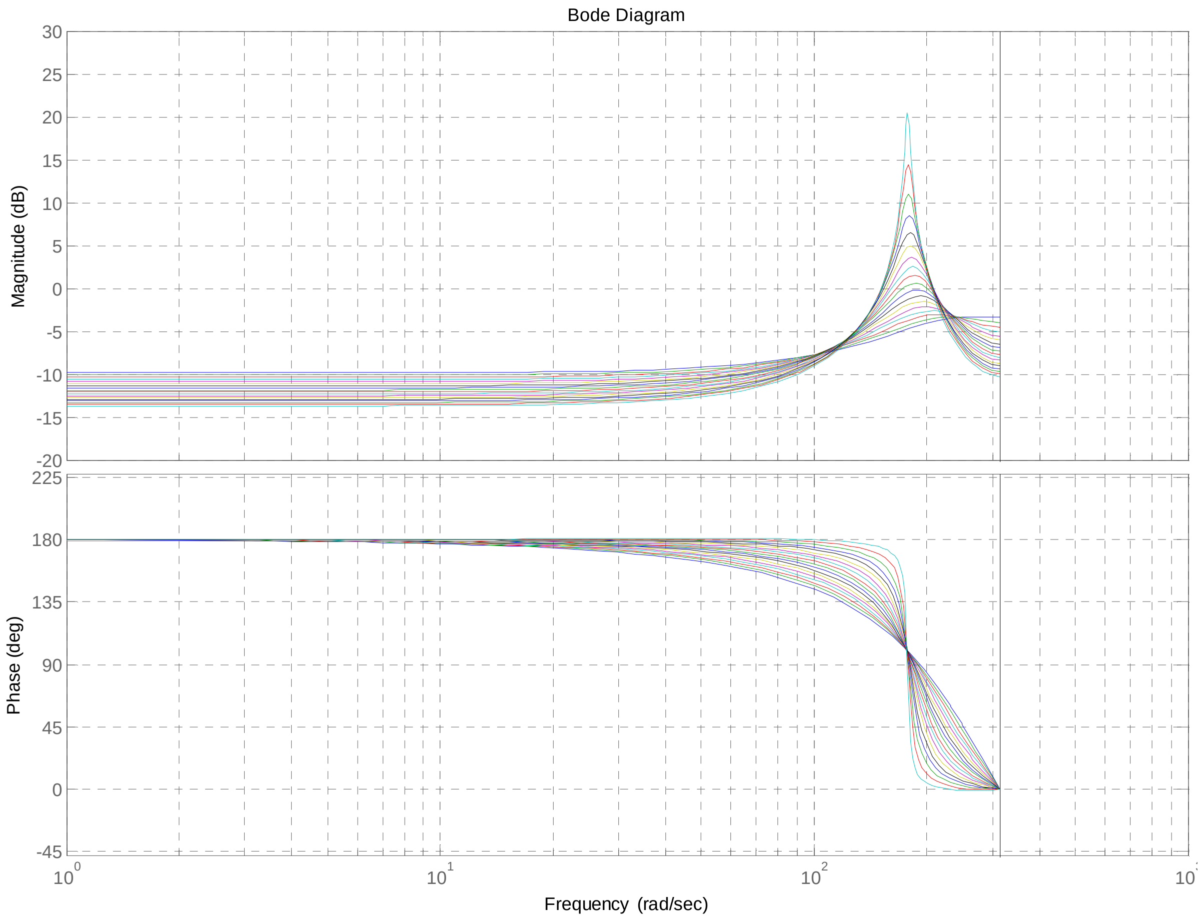
The proposed method addresses numerical issues, as follows.

1. Linear discrete-time state space models are estimated for each operating point, using time- or frequency-domain subspace algorithms
2. The identified models are balanced using the numerical algorithm of (Laub *et al.* 1987)
3. If necessary, the balanced models are converted to continuous-time using a bilinear transformation
4. The p-dependent model is obtained by interpolation of the state-space matrices of the local models, made possible by the properties of balanced realisations
5. The model can be eventually converted to LFT form.

A numerical example

$$A = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0.01p & -(p-0.004) & -0.39 \end{bmatrix}, \quad B = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}^T, C = \begin{bmatrix} 0.02 & 0 & -0.5 \end{bmatrix}$$

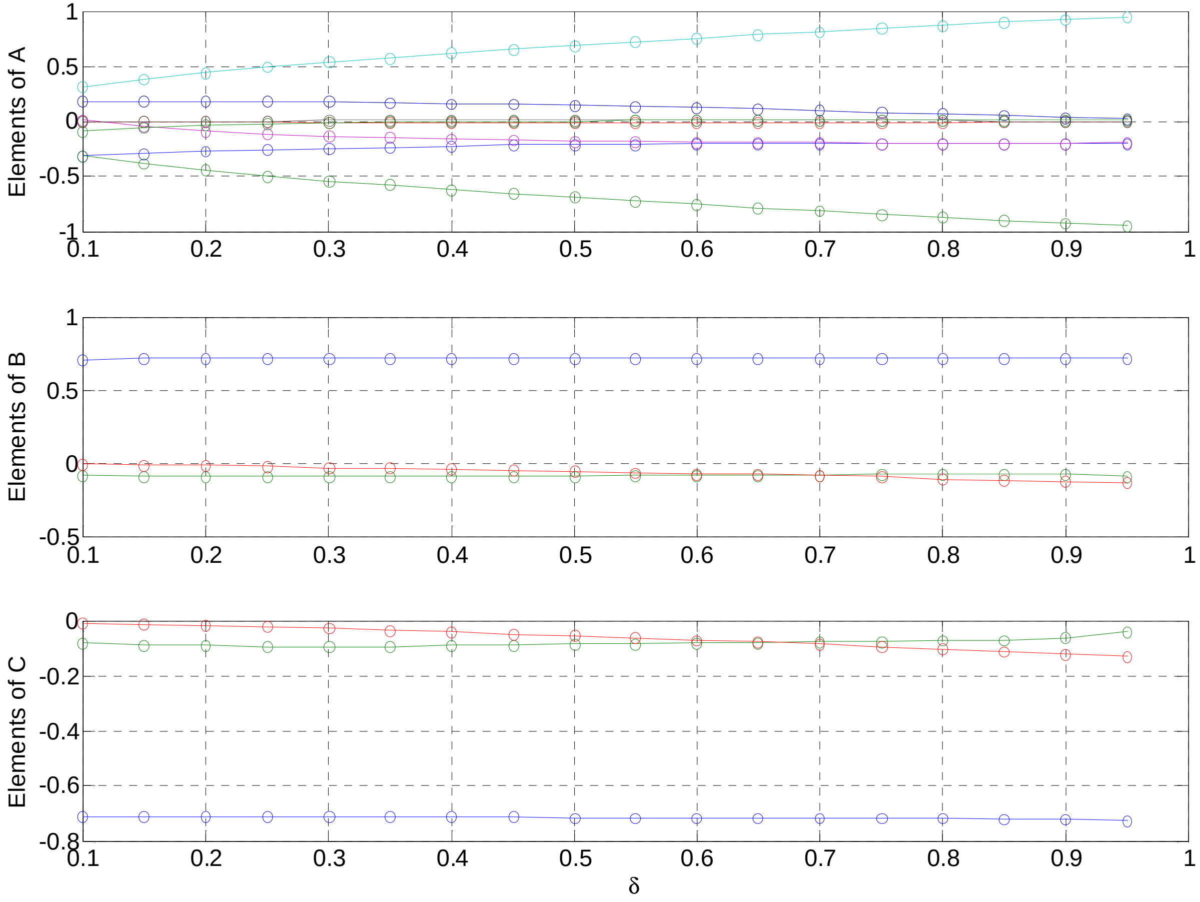
$$p \in \begin{bmatrix} 0.1 & 0.95 \end{bmatrix}$$



A numerical example

$$A = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0.01p & -(p-0.004) & -0.39 \end{bmatrix}, \quad B = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}^T, C = \begin{bmatrix} 0.02 & 0 & -0.5 \end{bmatrix}$$

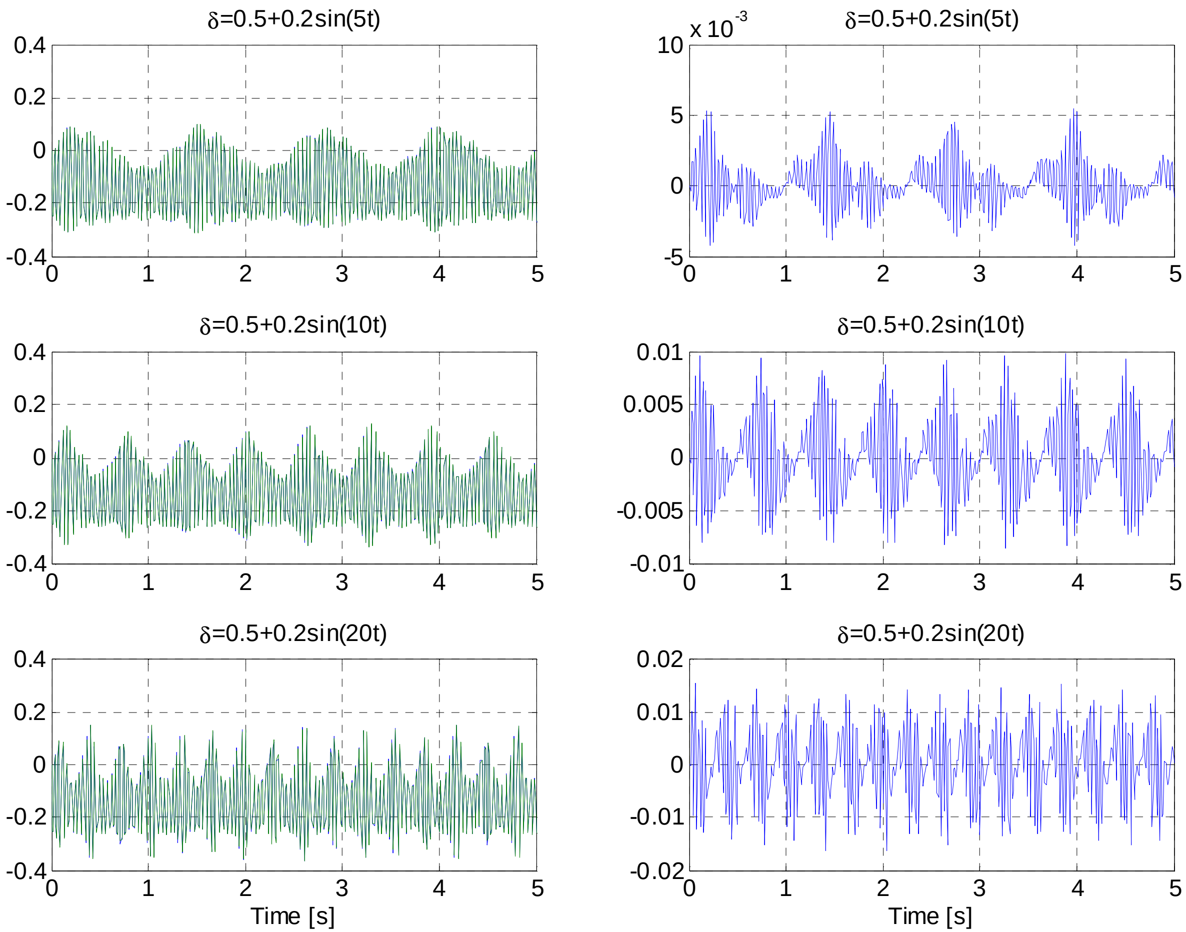
$$p \in \begin{bmatrix} 0.1 & 0.95 \end{bmatrix}$$



A numerical example

$$A = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0.01p & -(p-0.004) & -0.39 \end{bmatrix}, \quad B = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}^T, C = \begin{bmatrix} 0.02 & 0 & -0.5 \end{bmatrix}$$

$$p \in \begin{bmatrix} 0.1 & 0.95 \end{bmatrix}$$



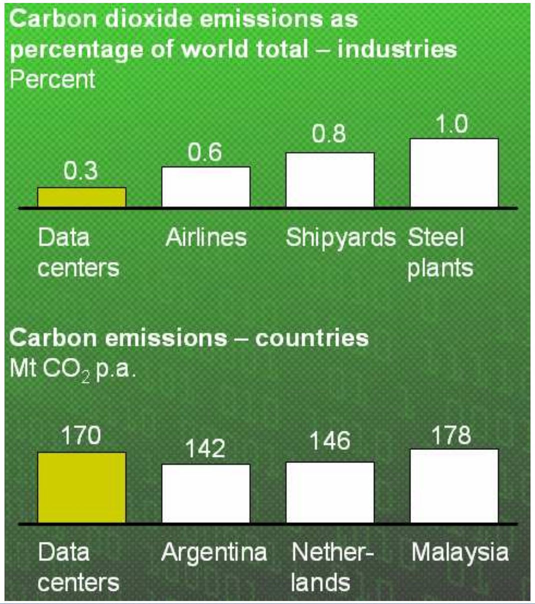
Outline

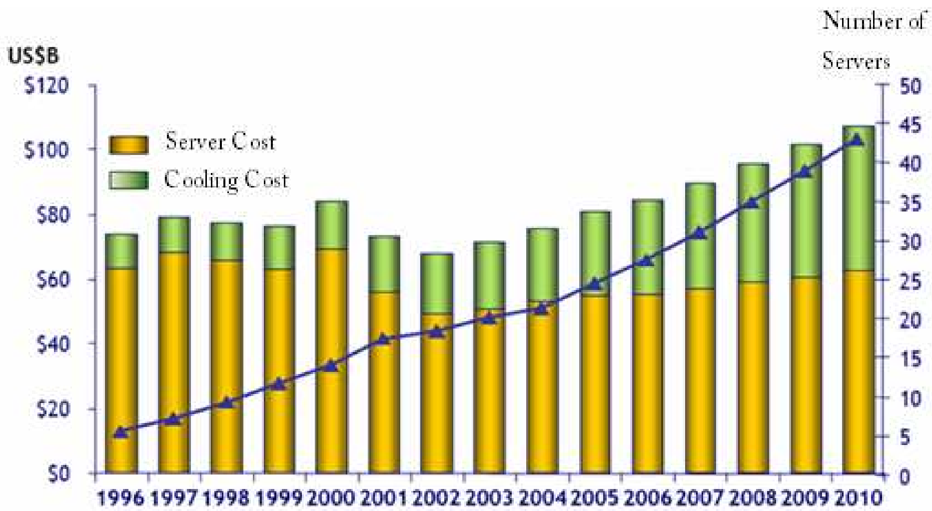
- Introduction and motivation
- LPV model classes:
  - state-space form
  - input-output form
- Overview of LPV model identification
- Application:
  LPV Identification in Virtualized Service Center Environments
- Integrated LFT modelling and identification of physical systems
- Perspectives and conclusions

Energy Management in Service Centers

Data Center issues

- Energy consumption
  - 2% of CO2 emission
  - By 2012 energy costs will be 40% of TCO
    - Related costs: cooling, UPS, …
- QoS guarantees and workload variability
- Dynamic resource managment

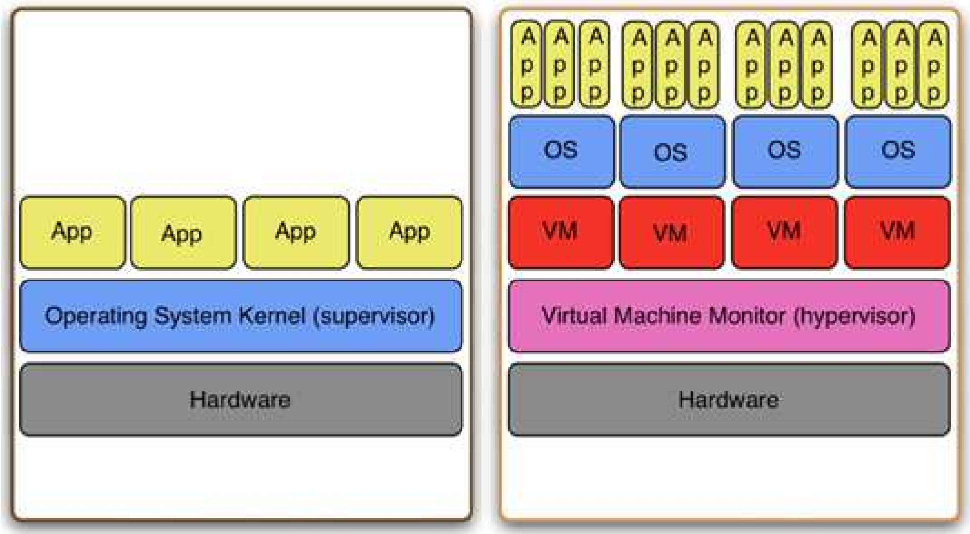




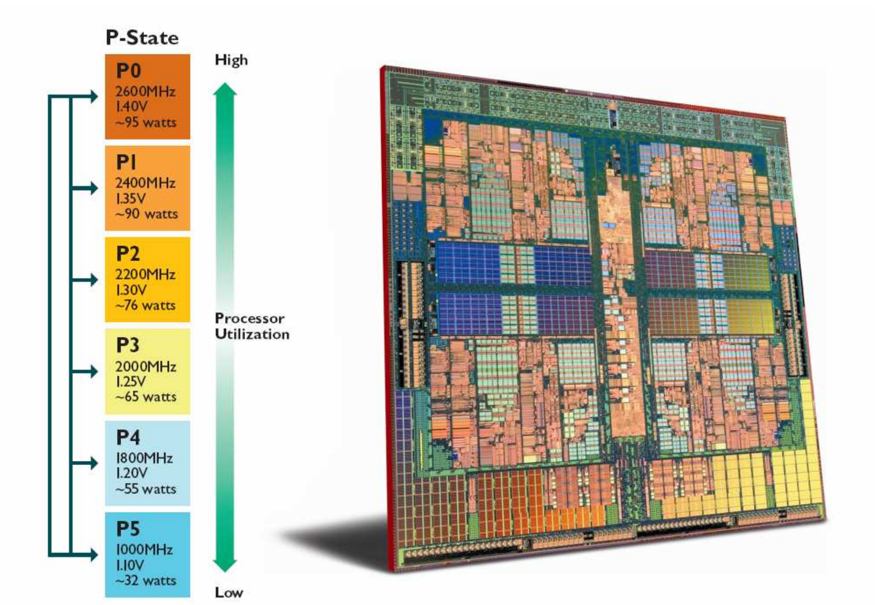
Virtualization

- Hardware resources (CPU, RAM, ecc...) are partitioned and shared among multiple **virtual machines** (VMs)
- The virtual machine monitor (VMM) governs the access to the physical resources among running VMs
- Performance isolation and security

A single OS



VMs run possibly different OSs



Dynamic Frequency Scaling (DFS)

- Modern CPUs can work in multiple **p-states** (performance-state) characterized by a given value of voltage and clock frequency
- A p-state transition implies a CPU clock update and, hence, different cost and performance
- Reduced overheads

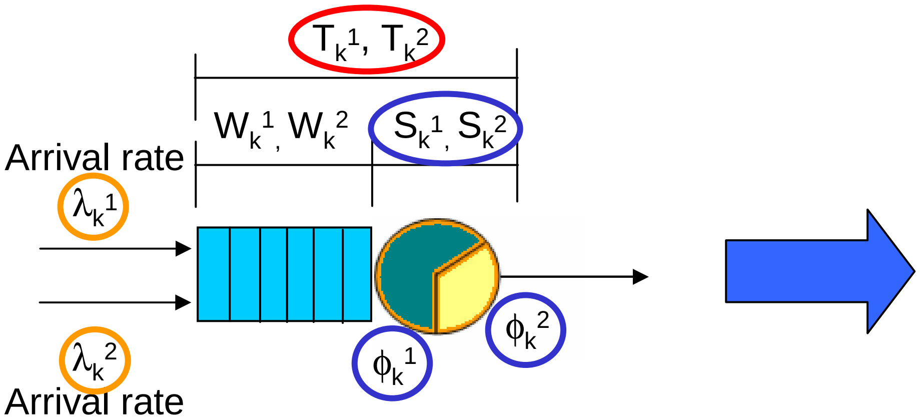
- Utility Based Approach: Queueing Network model + Optimization framework (e.g., IBM's Tivoli)
  - Multiple decision variables
  - Long term time horizon (several minutes)
  - Steady state assumption

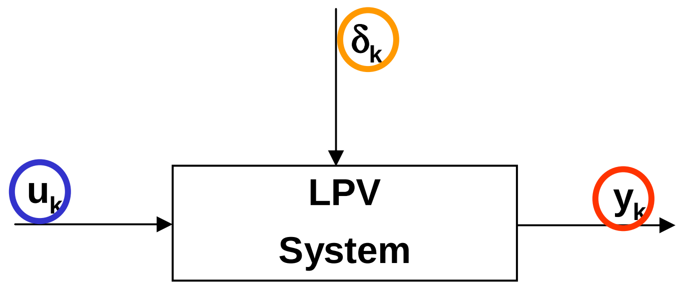
- Control Theory Approach
  - Short time frame (minutes, seconds)
  - System identification used to develop models for:
    - Capturing system transients
    - Taking into account workload variability
  - Advanced control design techniques used to:
    - Ensure closed-loop stability
    - Guarantee performance (QoS) levels *a priori*

System Identification: models for a multi-class virtualised environment

- Multi-class system
  - λ_k^i: requests arrival rate
  - s_k^i: service time, CPU time required to serve a single request
  - T_k^i: response time, overall time a request stays in the system
  - ϕ_k^i: VMM scheduling parameters

- Example: two VMs environment



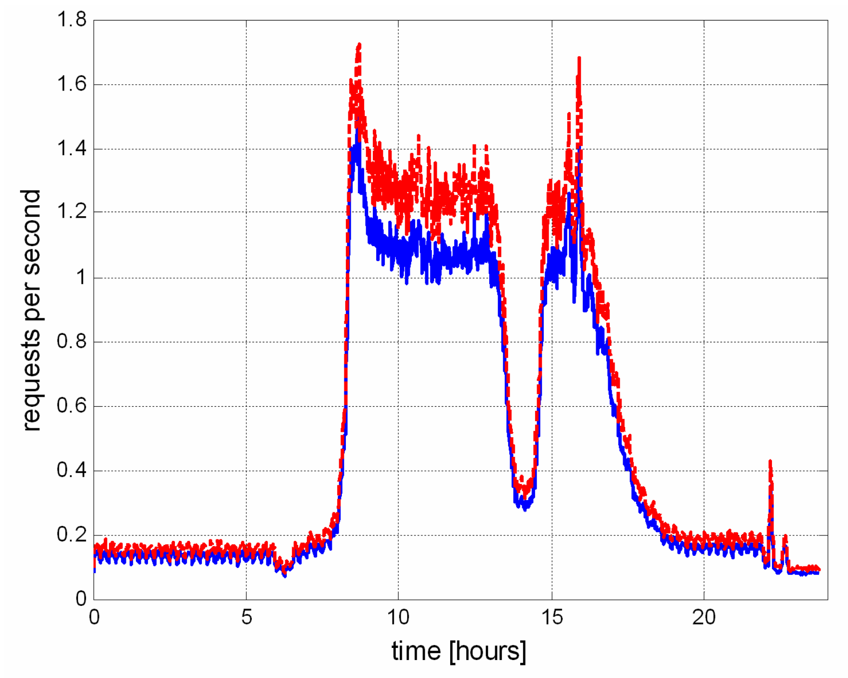


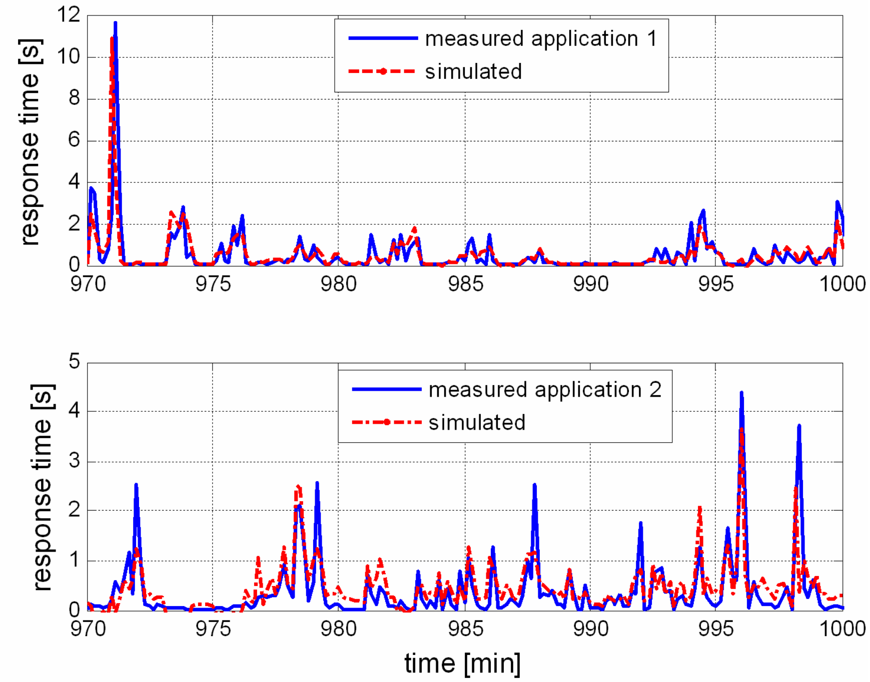
$$x_{k+1} = A(\delta_k)x_k + B(\delta_k)u_k$$
$$y_k = C(\delta_k)x_k + D(\delta_k)u_k$$

Experimental setting

- A workload generator
  - Apache JMeter custom extension
- Micro benchmarking web application
  - CPU service time generated according to deterministic (identification), exponential, lognormal, Pareto (validation) distributions
- Application instrumentation (otherwise, ARM API or kernel-based measurement)
- VMM monitor: Xen 3.0. Two instances of the micro-benchmarking Web service applications hosted in two Linux Fedora VMs
- Validation: synthetic workload inspired by a real-world usage (Politecnico di Milano Web site and a large financial system, 24 hours)

Multi-class virtualised environment: workload and performance metrics





$$VAF = 100\left(1 - \frac{Var[y_k - y_{sim,k}]}{Var[y(k)]}\right)$$

$$e_{avg} = 100\left|\frac{E_t[y_k - y_{sim,k}]}{E_t[y_k]}\right|$$

		VAF on 24h	e_{avg} on 24h
2 VMs	VM1	82.21%	5.46%
	VM2	73.41%	2.63%
3 VMs	VM1	81.59%	4.61%
	VM2	81.1%	1.16%
	VM3	72.97%	11.07%
4 VMs	VM1	78.66%	8.12%
	VM2	73.12%	11.36%
	VM3	73.57%	7.74%
	VM4	56.27%	15.06%

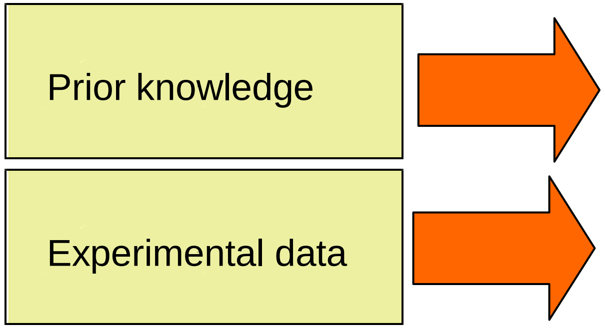
Preliminary conclusions and future work

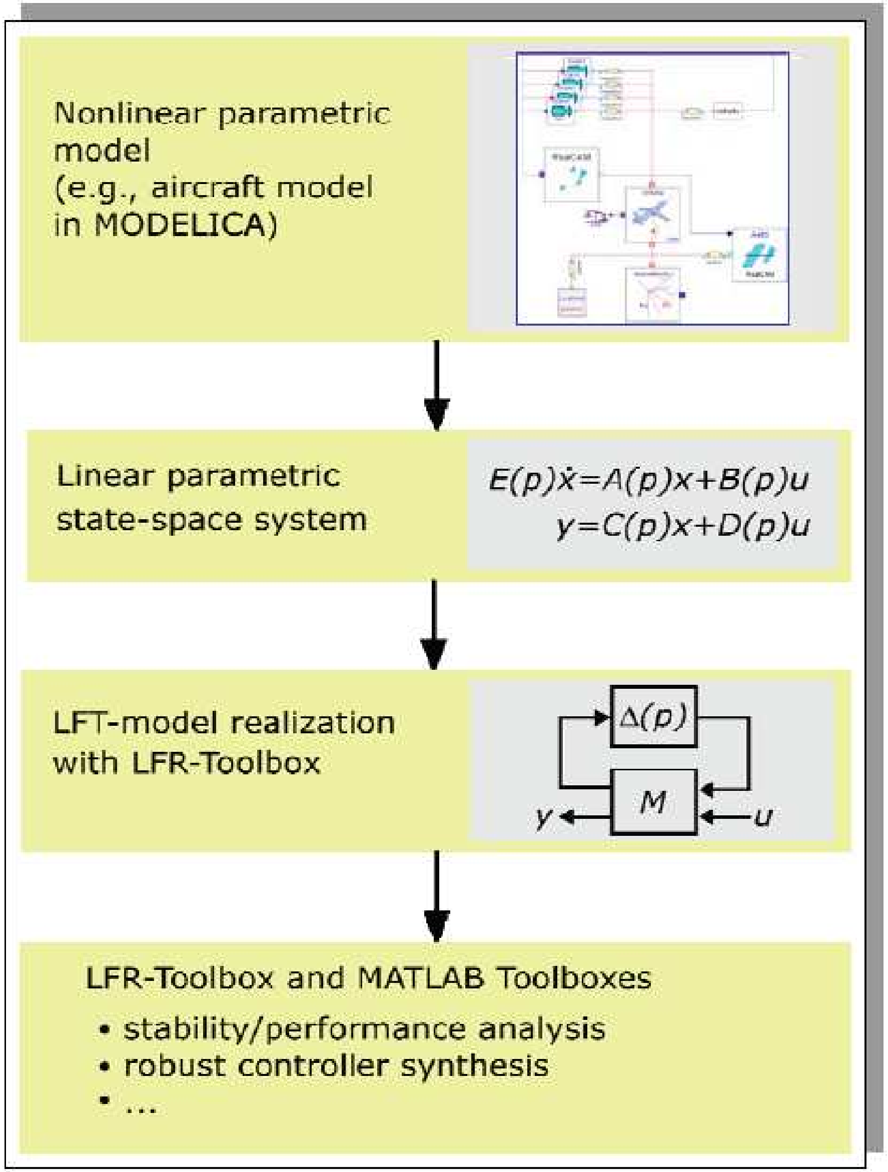
- LPV model identification seems suitable to model Web-servers dynamics
- Also tested recursive identification algorithms (i.e., for on-line applications), with promising results
- Current work aims at:
  - Identification of dynamic models for:
    - Admission control
    - Large scale virtualised set-up
  - Control design for:
    - Single-class systems (admission and DVS control)
    - MIMO virtualised environments

Outline

- Introduction and motivation
- LPV model classes:
  - state-space form
  - input-output form
- Overview of LPV model identification
- Application:
  LPV Identification in Virtualized Service Center Environments
- Integrated LFT modelling and identification of physical systems
- Perspectives and conclusions

The control-oriented modelling process





It is often suggested to use data at the simulator level: “model calibration”

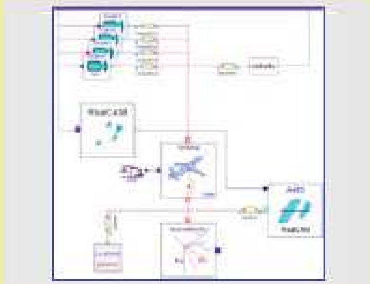
- Simulink
- Dymola
- “MoCaVa” (T. Bohlin)

…is this sensible?

The control-oriented modelling process (revisited)

Prior knowledge

Nonlinear parametric model
(e.g., aircraft model in MODELICA)

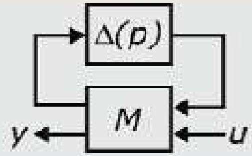


Linear parametric state-space system

$$E(p)\dot{x}=A(p)x+B(p)u$$
$$y=C(p)x+D(p)u$$

Experimental data

LFT-model realization with LFR-Toolbox



LFR-Toolbox and MATLAB Toolboxes

- stability/performance analysis
- robust controller synthesis
- ...

The present work

From L. Ljung's IFAC'08 plenary paper…

6.2 An Efficient Integration of Modeling and Parameter Estimation

Models and simulation are playing a more and more important role in industrial product and process development. Modeling and simulation tools like SIMULINK, DYMOLA, NI-MATRIXX, MODELICA, etc are ubiquitous for engineering work. Mostly, these models are derived from first principles, physical insight and sometimes they are black-box models obtained by system identification. The System Identification framework certainly offers grey-box techniques for mixing physical insights with information from measured data. But for these to be more used, they should be integrated in the engineer's daily modeling tool. An environment based on modern modeling from first principles, allowing Differential Algebraic Equations and model libraries, such as MODELICA, should be integrated with efficient parameter fitting to observed signals, and serious statistical model validation techniques. A study of how DAE modeling that includes stochastic disturbances can be adapted to system identification is given in Gerdin (2006).

This is precisely our goal!

Main advantages

- Physical modelling is performed in the appropriate language, with the appropriate tools (Modelica and, e.g., Dymola or OpenModelica)

- Support for the necessary symbolic manipulation already provided by:
  - Existing Modelica compilers
  - Tools such as the Matlab/Scilab LFR Toolbox

- Parameter estimation is performed in the appropriate framework, i.e., analysed is possible

- Modelica is an open, equation-based, object-oriented (O-O) modelling language, defined from 1997 by a non-profit organization.

- Models are written as differential-algebraic equations (DAEs):
  - easy model development (as on paper) and customization
  - easy documentation (models are self-documenting)
  - native multi-physics modelling

Main features of the language:

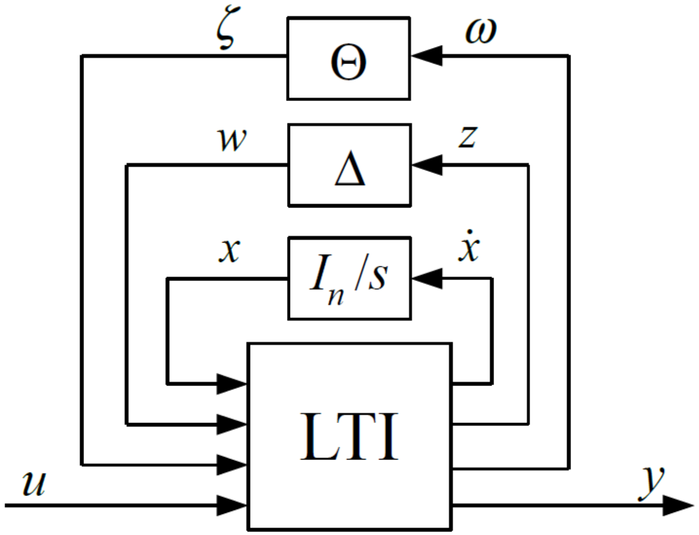
- Complex models built by connections through a-causal physical ports

- O-O features such as abstract interfaces, inheritance and replaceable models support model re-use and flexibility

- Many applications in automotive, aerospace, mechatronics, robotics, electrical machinery, hydraulics, thermodynamics, energy systems…

- Both commercial and open-source compilers are available

Introduction: LFR modelling and tools



$$
\begin{aligned}
\dot{x} &= Ax + B_1 w + B_2 \zeta + B_3 u \\
z &= C_1 x + D_{11} w + D_{12} \zeta + D_{13} u \\
\omega &= C_2 x + D_{21} w + D_{22} \zeta + D_{23} u \\
y &= C_3 x + D_{31} w + D_{32} \zeta + D_{33} u \\
w &= diag\left\{\delta_1 I_{r_1}, \ldots, \delta_q I_{r_q}\right\} z \\
\zeta &= \Theta(\omega)
\end{aligned}
$$

- Tools exist to build and manipulate LFRs via the block-diagram metaphor: **LFR Toolbox** (Onera/DLR) for Matlab and Scilab
- Cannot be used directly for O-O a-causal physical models
- Proposed solution: based on the integration of multiple tools
  - OpenModelica compiler
  - LFR Toolbox
  - Symbolic Toolbox

Bridge the gap between O-O models and LPV/LFT sysid & control

Preliminary DAE manipulation

Modelica code of the model

```
model XYZ
  Real x, y;
  input Real u;
  parameter Real T;
equation
  T*der(x) = -x +u;
  y = u + x;
end XYZ;
```

Flattening

DAEs of the O-O model

$$F_0(x_0, \dot{x}_0, v_0, u, p_0) = 0$$

Index reduction and change of state variables

Index-1 DAE system

$$F(x, \dot{x}, v, u, p_0) = 0$$

BLT transformation

Re-ordered sets of implicit eq.s

$$\Phi_1(x, u, \Xi_1, p_0) = 0$$

$$\Phi_2(x, u, \Xi_1, \Xi_2, p_0) = 0$$

$$\ldots$$

$$\Phi_q(x, u, \Xi_1, \ldots, \Xi_{q-1}, \Xi_q, p_0) = 0$$

Ξ_j contains the re-ordered unknowns $\dot{x}, v$

Getting rid of redundant parameters

The original Modelica model contains binding equations for all parameters (e.g., $p_1 = p_2 - p_3, \quad p_2 = 10, \quad p_3 = 20$)

- Binding equations for uncertain parameters p (a subset of the parameters p_0) are ignored
- The remaining binding equations are solved *symbolically* → symbolic expressions for p_0 as functions of p
- Such expressions are substituted *symbolically* in the reordered equations:

$$
\begin{aligned}
&\Phi_1(x, u, \Xi_1, p) = 0 \\
&\Phi_2(x, u, \Xi_1, \Xi_2, p) = 0 \\
&\ldots \\
&\Phi_q(x, u, \Xi_1, \ldots, \Xi_{q-1}, \Xi_q, p) = 0
\end{aligned}
$$

Recursive formulation

Assumption: the index-1 system is well-posed.

Then

$$
\begin{aligned}
&\Phi_1(x,u,\Xi_1,p)=0\\
&\Phi_2(x,u,\Xi_1,\Xi_2,p)=0\\
&\dots\\
&\Phi_q(x,u,\Xi_1,\dots,\Xi_{q-1},\Xi_q,p)=0
\end{aligned}
\quad\Rightarrow\quad
\begin{aligned}
&\Xi_1=\Psi_1(x,u,p)\\
&\Xi_2=\Psi_2(x,u,\Xi_1,p)\\
&\Xi_3=\Psi_3(x,u,\Xi_1,\Xi_2,p)\\
&\dots\\
&\Xi_q=\Psi_q(x,u,\Xi_1,\dots,\Xi_{q-1},p)
\end{aligned}
$$

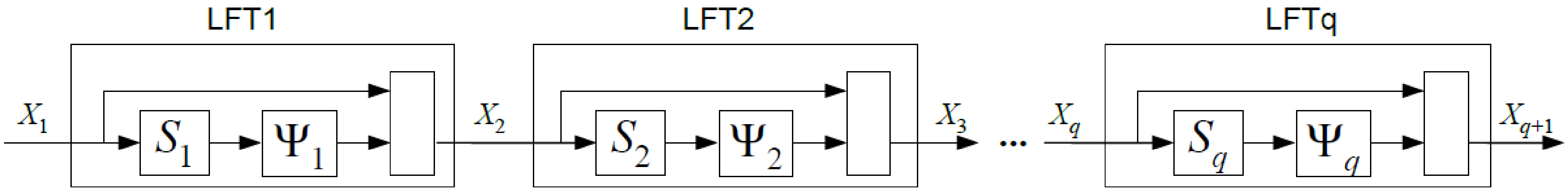
letting

$$
X_1=\begin{bmatrix} x\\ u\end{bmatrix}\quad X_j=\begin{bmatrix} x\\ u\\ \Xi_1\\ \dots\\ \Xi_{j-1}\end{bmatrix}
\quad\Rightarrow\quad \Xi_j=\Psi_j(\Sigma_j,p)
$$

$$
\Sigma_j=S_jX_j \qquad X_{j+1}=\begin{bmatrix} X_j\\ \Psi_j(S_jX_j,p)\end{bmatrix}
$$

Building the LFR

Easy interpretation in terms of a block diagram from the inputs x, u to all



- Each Ψ_j block is represented as an elementary LFR

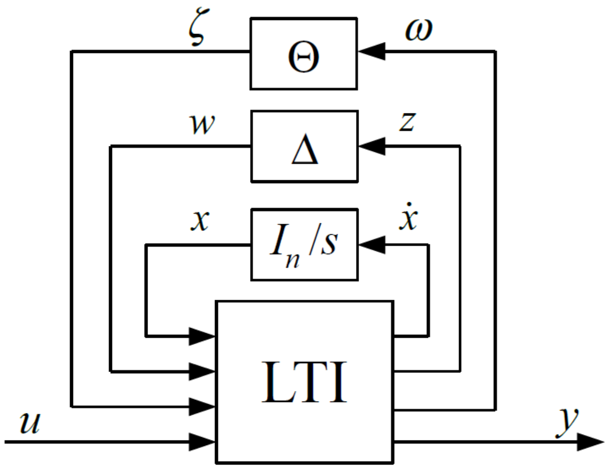
- $\Phi_j = 0$ linear (possibly depending on the uncertain parameters!)
  $\rightarrow$ LFR with uncertain parameters in Δ

Building the LFR - II

- By selecting the rows in the output equations, it is possible to obtain the LFR to compute **all derivatives** and the **required output variables only**

- By suitably re-arranging the corresponding matrices, the LFR can then be brought in the standard form

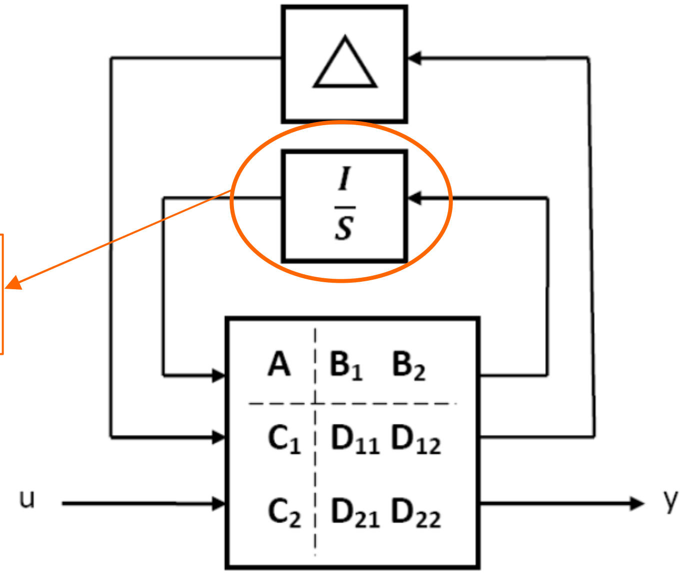
$$
\begin{aligned}
\dot{x} &= D_{3311}x + D_{311}w + D_{321}\zeta + D_{3312}u \\
z &= D_{131}x + D_{11}w + D_{12}\zeta + D_{132}u \\
\omega &= D_{231}x + D_{21}w + D_{22}\zeta + D_{232}u \\
y &= D_{3321}x + D_{312}w + D_{322}\zeta + D_{3322}u \\
w &= diag\left\{\delta_1 I_{r_1}, \ldots, \delta_q I_{r_q}\right\} z \\
\zeta &= \Theta(\omega).
\end{aligned}
$$



Discretisation and parameter estimation (focus on the linear case)

- Recall that the bilinear transformation is actually an LFT

$$\boxed{\frac{1}{s} = \frac{T}{2}\frac{z+1}{z-1} = T + \sqrt{T}z^{-1}(1-z^{-1})^{-1}\sqrt{T}}$$



- Discretisation of LPV/LFT models: interesting, partially open problem, see (Apkarian 1997), (Imbert 2001),…

- The output error algorithm of (Hsu et al. 2008): M-Δ interconnection, with

$$M_d = \begin{bmatrix} A_d & B_{1d} & B_{2d} \\ C_{1d} & D_{11d} & D_{12d} \\ C_{2d} & D_{21d} & D_{22d} \end{bmatrix}, \qquad \Delta = diag\left\{\delta_1 I_{n1}, ..., \delta_q I_{nq}\right\}$$

- Cost function:

$$J(\delta) = \frac{1}{2}\langle e(\delta), e(\delta)\rangle \quad \Rightarrow \quad \begin{aligned} g_k(\delta) &= \langle \frac{\partial e(\delta)}{\partial \delta_k}, e(\delta)\rangle \quad k = 1, \ldots, q \\ H_{is}(\delta) &= \langle \frac{\partial e(\delta)}{\partial \delta_i}, \frac{\partial e(\delta)}{\partial \delta_s}\rangle \quad i, s = 1, \ldots, q \end{aligned}$$

Conclusions

- LFRs are useful for system identification, control system analysis and design BUT
- existing LFR tools only deal with *causal* model specifications

- An algorithm has been proposed to translate generic a-causal, O-O nonlinear models with uncertain parameters into LFRs

- This bridges the gap between tools for physical system modelling and identification/control techniques based on LFRs

- The algorithm has been implemented within the OpenModelica compiler and MATLAB's Symbolic Toolbox, using Onera/DLR's LFR Toolbox.

- The resulting tool will be available as open source under GPL

Outline

- Introduction and motivation
- LPV model classes:
  - state-space form
  - input-output form
- Overview of LPV model identification
- Application:
  LPV Identification in Virtualized Service Center Environments
- Integrated LFT modelling and identification of physical systems
- Perspectives, ads, acks and conclusions

A fact: LPV model identification hardly used at all in practice.

There might be many reasons for this...

- Most tools are not available in public domain
- Most methods require unrealistic assumptions
- The obtained models do not match the existing design methods and tools: the “square peg-round hole” problem of system identification!

IEEE TCST Special Issue on "Applied LPV Modeling and Identification"

Call for Papers

Special Issue on Applied LPV Modelling and Identification

The IEEE CSS Technical Committee on System Identification and Adaptive Control is presently seeking papers for a special issue on "Applied LPV Modelling and Identification" for the IEEE Transactions on Control Systems Technology. The special issue aims at the following objectives:

- To present *significant case studies of relevant applications* (including, among others, fields such as aeronautics, space, automotive, robotics, mechatronics, computing systems, manufacturing) which have been successfully solved using methods and tools from the fields of Linear Parameter-Varying (LPV) modelling and identification.
- To promote LPV modelling and identification as an effective approach to deal with some important real world applications.
- To consolidate and solidify the several branches of LPV modelling and identification, highlighting promising application-oriented research directions arising from *the application to real, challenging problems of methods and tools for the derivation of high accuracy control-oriented models both on the basis of prior physical knowledge and input-output data.*

Deadline for submissions: April 30 2009!

Acknowledgments

- LPV model identification – local approach
  - Guillaume Mercere (LAII, Poitiers)

- LPV modelling of computing systems
  - Danilo Ardagna, Nicola Schiavoni and Mara Tanelli (POLIMI)
  - Alessandro Busi, Alessandro Caio, Marco Caldirola, Luca Destito, Paolo Sala and Mauro Speroni (POLIMI students)
  - Li Zhang (IBM T.J. Watson Research Center)

- Integrated LFT modelling and identification of physical systems:
  - Francesco Casella, Filippo Donida and Carlo Romani (POLIMI)

Concluding remarks

- A (quick) overview of the field of black-box LPV model identification has been provided
- A discussion of the pros and cons of each approach has been offered
- A case study of black-box LPV identification in the area of computing systems has been illustrated
- Current results in the integrated modelling and identification of LFT models have been presented.