Direct continuous-time approaches to system identification.
Overview and benefits for practical applications

Hugues GARNIER

University of Lorraine, CNRS
Research Centre for Automatic Control (CRAN)
Nancy, France

1. Data-driven Modelling
   • Motivation, goals and application areas
   • General system identification procedure
   • Overview of discrete and continuous-time approaches

2. Direct Continuous-time Methods
   • Basic issues
   • Traditional SVF-based estimators
   • Optimal instrumental variable estimators
   • Recent extensions to handle more complicated situations

3. Software Aspects and Benefits in Practical Applications
   • The CONTSID toolbox. A guided tour
   • Advantages of the continuous-time approaches
   • A selection of successful practical applications
Data is now everywhere

✓ From the engineer’s perspective

- How to use the information in the data **to build a model**?
- What kind of software is available?
- How to do it?

What is system identification?

✓ This is the field of modelling dynamic systems from input/output measured data (**also called data-driven modelling**)

\[
\begin{align*}
\dot{x}(t) &= Ax(t) + Bu(t) \\
y(t) &= Cx(t) + v(t)
\end{align*}
\]
System identification: a data compression approach

- Can be viewed as a **systematic data compression** method
  - Reduces a large number $N$ of noisy input/output observations to a small number $n_p$ of parameters ($n_p \ll N$) of a mathematical model

\[ u(t) \to G(q, \theta_0) \to y(t) \]

Find a model $G(q, \hat{\theta})$ based on $\{u(t), y(t)\}$ that models dynamics between $u(t)$ and $y(t)$

System

- **System**: define a part of the real world. The output is the response of the system to the input and disturbances

\[ u(t_k) \to \text{System} \to y(t_k) \]

- **Dynamic system**: a system with memory, *i.e.*, the input value at time $t$ will influence the output at future time-instants
  - Dynamic systems are everywhere!
✓ **Model**: a description of the system. It should capture the essential information about the system. 

\[
\begin{align*}
\text{Model} & \rightarrow \hat{y}(t_k) \\
u(t_k) & \rightarrow \text{System} \\
\text{System} & \rightarrow y(t_k) \\
u(t_k) & \rightarrow \text{Model} \\
\end{align*}
\]

✓ **Do not forget**! 
- Models are always *approximated versions* of the real systems. 
- But they can help to answer many questions about the system.

---

**A model. What for?**

*Models: used in virtually all disciplines of science and engineering*

✓ **Process design**: designing new cars, new airplanes, etc.

✓ **Control design**: simple controllers require simple models while optimal controllers need sophisticated models.

✓ **Simulation**: train nuclear plant operator, try new operation strategy.

✓ **Prediction**: forecasting the river flood, predicting the stock market, etc.

✓ **Signal processing**: communication, echo cancellation.

✓ **Decision making**: aiding in decision making by simulating ‘what if’ scenarios.

✓ **Fault detection**: use the model to detect the changes of the system behavior.
Wing flutter modelling: challenging problem in dynamic aeroelasticity
- Flutter phenomena can happen when vibrations occurring in an aircraft match the natural frequency of the structure.
- If not properly damped, the oscillations can increase in amplitude, leading to structural damage or even failure.

Effect of incorrect prediction of flutter
- [www.wired.com/2010/03/flutter-testing-aircraft/](http://www.wired.com/2010/03/flutter-testing-aircraft/)

Objective:
- Identified model should help to prevent dynamic instability of the wing
- Design changes are made to stiffen the wing structure

Typical testing data for wing flutter modelling

During a sweep sine test, a wide range of oscillations are introduced to a flight-control surface such as the ailerons, rudder or elevator
Rainfall-flow modelling: challenging problem in hydrology

Objective:
- identified model should represent the rain that is effective in causing variations in the river flows
- A flood forecasting and warning system can be designed

Typical data for rainfall-flow modelling

Daily data from the ephemeral Canning river, Western Australia
Applications of system identification

✓ More examples listed in the *IEEE Control Systems Magazine*, Oct. 2007

- In-Flight Vibration Monitoring of Aeronautical Structures
- Electric Load Forecasting
- Dynamic Model Identification for Industrial Robots
- High-Purity Distillation column
- Identification of Stellar Interferometer models from empirical data
- Space Weather Forecasting

✓ More examples listed in the *special issue of Int. Journal of Control*, July 2014

- Identification for a class of hybrid systems with application to power electronics
- Identification of aircraft structural modes from short-duration flight test
- Identification of the steering dynamics of a ship on a river
- Identification of a smoking cessation intervention
- Identification of blood glucose dynamics for type 1 diabetes
Types of models

- Mental or intuitive
  - Ex: driving your bike, your car

- Graphs and tables (also known as non-parametric model)
  - Ex: step responses or Bode plots

- Mathematical models (also known as parametric model)
  - differential and difference equations, state-space representation
  - are well-suited to model dynamic systems

Data-driven mathematical models

- Grey-box models
  - model constructed in continuous-time from basic physical principles
  - parameters have a direct physical interpretation
  - also known as physically-parameterised or tailor-made models

- Black-box models
  - are families of flexible models of general applicability
  - parameters have no direct physical interpretation
  - also known as ready-made models
  - can be continuous-time or discrete-time

Only linear black-box model identification considered in this lecture
Discrete-time (DT) models of linear systems

- A model that directly expresses the relationship between the values of the I/O signals at the time-instants is called a **discrete-time model**
  - **Difference equation / transfer (operator) function model**
    \[ y(t_k) + a_1 y(t_{k-1}) + \cdots + a_{n_a} y(t_{k-n_a}) = b_1 u(t_{k-1}) + \cdots + b_{n_b} u(t_{k-n_b}) \]
    
    \[ G(z) = \frac{Y(z)}{U(z)} = \frac{B(z)}{A(z)} \]

- **State-space model**
  \[
  \begin{align*}
  x(t_{k+1}) &= Ax(t_k) + Bu(t_k) \\
  y(t_k) &= Cx(t_k) + Du(t_k)
  \end{align*}
  \]
  \[ G(z) = C(zI - A)^{-1} B + D \]

Continuous-time (CT) models of linear systems

- A model that describes the relationship between time continuous I/O signals is called a **continuous-time model**
  - **Differential equation / transfer (operator) function model**
    \[ \frac{d^n y(t)}{dt^n} + a_1 \frac{d^{n-1} y(t)}{dt^{n-1}} + \cdots + a_n y(t) = b_0 \frac{d^m u(t)}{dt^m} + b_1 \frac{d^{m-1} u(t)}{dt^{m-1}} + \cdots + b_m u(t) \]
    
    \[ G(s) = \frac{Y(s)}{U(s)} = \frac{B(s)}{A(s)} \]

- **State-space model**
  \[
  \begin{align*}
  \dot{x}(t) &= Ax(t) + Bu(t) \\
  y(t) &= Cx(t) + Du(t)
  \end{align*}
  \]
  \[ G(s) = C(sI - A)^{-1} B + D \]
Stages in the iterative system identification procedure

1. Start
2. Experimental Design and Execution
   - Experimental Design
   - Data Collection
   - Data Preprocessing
   - Model Structure Determination
   - Parameter Estimation
3. Identification
   - Data Preprocessing
   - Model Structure Determination
   - Parameter Estimation
4. Model Validation
   - Simulation, Residual auto and cross-correlation, step-response
5. Does the model meet validation criteria?
   - Yes
   - No

System identification in practice

- User/practitioner
  - expected to have an extensive background in
    - statistics
    - estimation theory
    - signal processing
    - optimization
  - expected to understand
    - various model parameter estimation methods
    - associated decision variables in terms of **bias-variance** trade-offs
Principal sources of error in estimation theory

- Consider the estimate \( \hat{\theta}_N \) of the true parameter vector \( \theta_o \) based on \( N \) data

\[ \text{Error} = \text{Bias} + \text{Variance} \]

- Bias = systematic error
  - Unbiased estimate if \( E(\hat{\theta}_N) = \theta_o \)

- Variance = fluctuations of the estimates around its expected value
  \[
P_\theta = E\left((\hat{\theta}_N - \theta_o)(\hat{\theta}_N - \theta_o)^T\right)
  \]
  - Affected by number of data \( N \), noise, SNR

- Consistency
  - When \( N \to \infty \), if the bias and variance \( \to 0 \), then the estimator is said to be consistent

Illustration of bias-variance trade-off for estimators

Center of the target represents \( \theta_o \)
- Estimator A: biased (estimates not in the center of the target)
- Estimator B: unbiased but quite large fluctuations – large variance
- Estimator C: unbiased and minimum variance
Objectives when developing an estimator

✓ First focus: construct estimators to be unbiased

✓ Second focus: construct estimators with minimum variance for the parameter estimates, and in any case as small as possible
  - **Optimal** estimator: unbiased and minimum variance

✓ Further aspect: strive to get a robust estimator
  - robustness to experiment design \((T_s, N, \text{prefiltering}, \ldots)\)
  - robustness to implementation aspects
  - robustness to noise
  - robustness to undermodelling, …

---

System identification – a brief historical review

✓ Early research
  - focused on identification of **CT models from CT data**

✓ Rapid developments in digital data acquisition and computers in the seventies have led to the identification of **DT models from sampled data**
  - **DT model identification** has become **THE** approach used in system identification

✓ The **last decade** has witnessed a renewed interest in the techniques for the identification of direct **CT models from sampled data**
Main approaches to identify linear parametric models from time-domain sampled data?

Discrete-time model

Continuous-time model

Matlab SID toolbox

CONTSID toolbox

Simulation example to start with

✓ The Rao-Garnier benchmark: 4th order simulated system

\[
G_0(p) = \frac{K(1-Tp)}{p^2 + \frac{2\zeta_1\omega_{n,1}}{\omega_{n,1}} p + 1}\frac{p^2 + \frac{2\zeta_2\omega_{n,2}}{\omega_{n,2}} p + 1}{p^4 + 5p^3 + 408p^2 + 416p + 1600} = \frac{-6400p + 1600}{p^4 + 5p^3 + 408p^2 + 416p + 1600}
\]

\[
\omega_{n,1} = 2 \text{ rad} / \text{s}; \quad \omega_{n,2} = 20 \text{ rad} / \text{s}
\]

\[
\zeta_1 = 0.1; \quad \zeta_2 = 0.25
\]

✓ 2 separated oscillatory modes
✓ 1 unstable zero
Simulation setup

✓ Performance of DT and CT model identification evaluated by using Monte Carlo simulation (MCS) of 100 runs
  ▪ MCS: very useful tool to evaluate and compare estimators

✓ Simulation setup
  ▪ $T_s = 10 \text{ ms}$
    • fast sampling
    • but not unrealistic in view of today’s sampling situations
  ▪ $u(t)$: PRBS (kept the same during the MCS)
  ▪ $e(t_k)$: DT white noise, SNR=10dB

I/O data for one realization of the noise
CT and DT models and related methods used

**DT model and estimation methods tested**

\[
G(q, \theta) = \frac{\bar{b}_1 q^{-1} + \bar{b}_2 q^{-2} + \bar{b}_3 q^{-3} + \bar{b}_4 q^{-4}}{1 + \bar{a}_1 q^{-1} + \bar{a}_2 q^{-2} + \bar{a}_3 q^{-3} + \bar{a}_4 q^{-4}}
\]

- IV4 (multistep IV ≠ iterative IV)
- N4SID
- OE (pem)

4 parameters in the numerator
artificial zeros due to ZOH discretisation

**CT model and estimation methods tested**

\[
G(p, \theta) = \frac{b_0 + b_1 p}{a_0 + a_1 p + a_2 p^2 + a_3 p^3 + p^4}
\]

- LSSVF
- IVSVF (two-step IV)
- SRIVC (iterative IV)

A priori knowledge about the number of parameters
to be estimated for the numerator easy to accommodate

All routines from the SID and CONTSID toolboxes used in their default mode with the latest Matlab R2015a release

Frequency responses of the DT models estimated from the measured data (SID toolbox)

u(t) \quad G_o(p) \quad x(t) \quad T_s \quad u(t_k) \quad e(t_k) \quad y(t_k)

\[
G(q, \theta) = \frac{b_1 q^{-1} + b_2 q^{-2} + b_3 q^{-3} + b_4 q^{-4}}{1 + a_1 q^{-1} + a_2 q^{-2} + a_3 q^{-3} + a_4 q^{-4}}
\]

N4SID

IV4

OE
Frequency responses of the CT models estimated from the measured data (CONTSID toolbox)

\[ G(p, \theta) = \frac{b_0 + b_1 p}{a_0 + a_1 p + a_2 p^2 + a_3 p^3 + p^4} \]

Frequency responses of the DT methods estimated from prefILTERED data (SID toolbox)

\[ G(q, \theta) = \frac{b_0 q^{-1} + b_2 q^{-2} + b_3 q^{-3} + b_4 q^{-4}}{1 + a_1 q^{-1} + a_2 q^{-2} + a_3 q^{-3} + a_4 q^{-4}} \]
Frequency responses of the DT methods estimated from \textit{decimated} data (SID toolbox)

\[ L(q) \cdot \text{low-pass filter} \]

\[ \begin{align*} G_0(p) \quad & \quad x(t) \\ T_s \quad & \quad u(t_k) \\ L(q) \quad & \quad y(t_k) \\ r \quad & \quad e(t_k) \end{align*} \]

\[ \begin{align*} u(t_k) & \rightarrow L(q) \rightarrow y(t_k) \\ T_s & \rightarrow u(t_k) \rightarrow y(t_k) \end{align*} \]

\[ G(q,\theta) = \frac{b_1q^{-1} + b_2q^{-2} + b_3q^{-3} + b_4q^{-4}}{1 + a_1q^{-1} + a_2q^{-2} + a_3q^{-3} + a_4q^{-4}} \]

\[ \text{Decimation } T_{s1} = rT_s \quad R = 10 \]

---

What can be concluded from the benchmark results?

- **Traditional DT model techniques** encounter \textit{pbs}
  - Very sensitive to the initiation of the iterative estimators
  - Numerical issues for fast sampling

  - To get good results, the user need to have some expertise to apply:
    - \textit{data prefiltering}
    - \textit{data decimation}: reduction of the sampling frequency

- **Direct CT model identification techniques**
  - Free of these difficulties
  - Include \textit{inherent data prefiltering}
  - \textit{Seems worth to know more about the theory behind these approaches…}
Outline

1. Introduction to Data-based Modelling
   - Motivation, goals and application areas
   - General system identification procedure
   - Overview of discrete and continuous-time approaches

2. Direct Continuous-time Methods
   - Basic issues
   - Traditional SVF-based estimators
   - Optimal instrumental variable estimators
   - Recent extensions to handle more complicated situations

3. Software Aspects and Benefits in Practical Applications
   - The CONTSID toolbox. A guided tour
   - Advantages of the continuous-time approaches
   - A selection of successful practical applications

Aim of this part

✓ Provide an introduction
  - theory of some reliable methods for direct continuous-time model identification of dynamic systems

✓ The key computational methods we refer to are
  - State-Variable Filtering (SVF) approach
    - for estimating the I/O time-derivatives
  - Instrumental variable (IV) method
    - for identifying the parameters of the continuous-time model
Issue in CT model identification: time-derivative measurement problem

✓ **DT model identification** - *difference* equation model

\[ y(k) + a_1 y(k-1) + \cdots + a_n y(k - n_a) = b_1 u(k-1) + \cdots + b_n u(k - n_b - 1) \]

✓ **CT model identification** - *differential* equation model

Unlike the DT model, where only sampled input and output data appear, the CT differential equation model contains I/O time-derivatives

\[
\begin{align*}
\frac{d^n y(t)}{dt^n} + a_1 \frac{d^{n-1} y(t)}{dt^{n-1}} + \cdots + a_n y(t) &= b_0 \frac{d^m u(t)}{dt^m} + b_1 \frac{d^{m-1} u(t)}{dt^{m-1}} + \cdots + b_m u(t) \\
\end{align*}
\]

Well-known approach to handle the time-derivative problem:

**data prefiltering strategy** to generate *smoothed* time-derivative estimates

Traditional SVF method

\[ y^{(n)}(t) + a_1 y^{(n-1)}(t) + \cdots + a_n y^{(0)}(t) = b_0 u^{(m)}(t) + \cdots + b_m u^{(0)}(t) \]

Apply a stable filter with operator model \( L(p) = 1/E(p) \) on both sides, the prefiltered differential equation model becomes (except for a possible transient)

\[ y_f^{(n)}(t) + a_1 y_f^{(n-1)}(t) + \cdots + a_n y_f^{(0)}(t) = b_0 u_f^{(m)}(t) + \cdots + b_m u_f^{(0)}(t) \]

How to choose \( E(p) \)?

\[ E(p) = (p + \lambda)^n \]

The filtered time-derivatives can then be exploited to estimate the parameters of the differential equation model
Simple LS-based SVF estimator

✓ At $t = t_k$, the prefiltred DE model can be rewritten in linear regression form

$$y_f^{(n)}(t_k) = q^T_f(t_k) \theta + \varepsilon(t_k)$$

$$q^T_f(t_k) = \begin{bmatrix} -y_f^{(n-1)}(t_k) & \cdots & -y_f^{(0)}(t_k) & u_f^{(m)}(t_k) & \cdots & u_f^{(0)}(t_k) \end{bmatrix}$$

$$\theta = \begin{bmatrix} a_1 & \cdots & a_n & b_0 & \cdots & b_m \end{bmatrix}^T$$

✓ From $N$ samples observed at $t_1, \ldots, t_N$, the Least squares-based SVF estimate is the minimizing argument of the sum of squared errors

$$\hat{\theta}_{\text{LSSVF}} = \arg \min_{\theta} \left( \frac{1}{N} \sum_{k=1}^{N} \left( y_f^{(n)}(t_k) - q^T_f(t_k) \theta \right)^2 \right)$$

$$\hat{\theta}_{\text{LSSVF}} = \left[ \frac{1}{N} \sum_{k=1}^{N} q_f(t_k) q^T_f(t_k) \right]^{-1} \left[ \frac{1}{N} \sum_{k=1}^{N} q_f(t_k) y_f^{(n)}(t_k) \right]$$

This simple LS-based SVF estimator represents the simplest archetype of CT model identification from sampled data
Consider a second-order system
\[
\frac{d^2 y(t)}{dt^2} + a_1 \frac{dy(t)}{dt} + a_2 y(t) = b_0 u(t) + e(t)
\]
\[
(p^2 + a_1 p + a_2) y(t) = b_0 u(t) + e(t)
\]

Apply a second-order SVF filter \(L(p)=1/(p+\lambda)^2\)
\[
\left(\frac{p^2}{(p+\lambda)^2} + a_1 \frac{p}{(p+\lambda)^2} + a_2 \frac{1}{(p+\lambda)^2}\right) y(t) = \begin{bmatrix} b_0 & 1 \\ 1 & 1 \end{bmatrix} u(t) + \begin{bmatrix} 1 \\ (p+\lambda)^2 \end{bmatrix} e(t)
\]
\[
y_f^{(2)}(t) + a_1 y_f^{(1)}(t) + a_2 y_f^{(0)}(t) = b_0 u_f^{(0)}(t) + e_f(t)
\]
\[
y_f^{(2)}(t_k) = \begin{bmatrix} -y_f^{(1)}(t_k) & -y_f^{(0)}(t_k) & u_f^{(0)}(t_k) \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ b_0 \end{bmatrix} + e_f(t_k)
\]

For \(t=t_1, \ldots, t_N\), we have
\[
\begin{bmatrix}
y_f^{(2)}(t_1) \\
y_f^{(2)}(t_2) \\
\vdots \\
y_f^{(2)}(t_N)
\end{bmatrix} = \begin{bmatrix}
-y_f^{(1)}(t_1) & -y_f(t_1) & u_f(t_1) \\
-y_f^{(1)}(t_2) & -y_f(t_2) & u_f(t_2) \\
\vdots & \vdots & \vdots \\
-y_f^{(1)}(t_N) & -y_f(t_N) & u_f(t_N)
\end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ b_0 \end{bmatrix} + \begin{bmatrix} e_f(t_1) \\ e_f(t_2) \\ \vdots \\ e_f(t_N) \end{bmatrix}
\]
\[
Y_N = \Phi_N \theta + E_N
\]
\[
\hat{\theta}_{lssvf} = \left(\Phi_N^T \Phi_N\right)^{-1} \Phi_N^T Y_N
\]
LSSVF estimator - Matlab implementation – Noise-free case

B=2;
A=[1 4 3]; % A(p)=p^2+4p+3 – True system
Ts=0.01;
N=1000;
t=(0:N-1)*Ts;
u=square(2*pi*0.5*t);
x=lsim(B,A,u,t); % simulation of the noise-free output
data=iddata(x,u,Ts);idplot(data);

lambda=3; % λ : cut-off frequency of the SVF filter
den_L=[1 2*lambda lambda^2]; % denominator of the SVF filter
num_L0=1; % numerator of L0(p)=1/(p+λ)^2
num_L1=[1 0]; % numerator of L1(p) =p/(p+λ)^2
num_L2=[1 0 0]; % numerator of L2(p) =p^2/(p+λ)^2
xf0=lsim(num_L0,den_L,x,t); % Computation of the filtered output
xf1=lsim(num_L1,den_L,x,t); % Computation of the filtered output
xf2=lsim(num_L2,den_L,x,t); % Computation of the filtered output
uf0=lsim(num_L0,den_L,u,t); % Computation of the filtered input
phi_f=[-xf1 -xf0 uf0]; % Regression matrix
yf=xf2; % Output vector
theta_lssvf=phi_f'yf % LSSVF estimates

LSSVF estimator - Matlab implementation – Noisy case

B=2;
A=[1 4 3]; % A(p)=p^2+4p+3 – True system
Ts=0.01;
N=1000;
t=(0:N-1)*Ts;
u=square(2*pi*0.5*t);
x=lsim(B,A,u,t); % simulation of the noise-free output
data=iddata(x,u,Ts);idplot(data);

lambda=3; % λ : cut-off frequency of the SVF filter
den_L=[1 2*lambda lambda^2]; % denominator of the SVF filter
num_L0=1; % numerator of L0(p)=1/(p+λ)^2
num_L1=[1 0]; % numerator of L1(p) =p/(p+λ)^2
num_L2=[1 0 0]; % numerator of L2(p) =p^2/(p+λ)^2
yf0=lsim(num_L0,den_L,y,t); % Computation of the filtered output
yf1=lsim(num_L1,den_L,y,t); % Computation of the filtered output
yf2=lsim(num_L2,den_L,y,t); % Computation of the filtered output
uf0=lsim(num_L0,den_L,u,t); % Computation of the filtered input
phi_f=[-yf1 -yf0 uf0]; % Regression matrix
yf=yf2; % Output vector
theta_lssvf=phi_f'yf % LSSVF estimates

% see also the LSSVF routine in the CONTSID toolbox
Frequency responses of the LSSVF models (recap of the Rao-Garnier benchmark results)

\[ L(p) = \frac{1}{(p + \lambda)^4} \]
\[ \lambda = 13 \text{ rad / sec} \]

A small bias can be observed

Basic LSSVF estimator – Statistical analysis

✓ Assume the data-generating system is described as

\[ S: \quad y^{(n)}(t_k) = \varphi^T(t_k)\theta_o + v(t_k) \]

where \( \theta_o \) is the true parameter vector

✓ Assume that \( v(t_k) \) is a stationary stochastic process independent of \( u(t_k) \). After the SVF filtering, the data-generating system can be rewritten as

\[ y_f^{(n)}(t_k) = \varphi_f^T(t_k)\theta_o + v_f(t_k) \]

\[ \hat{\theta}_{lssvf} = \left[ \frac{1}{N} \sum_{k=1}^{N} \varphi_f(t_k)\varphi_f^T(t_k) \right]^{-1} \left[ \frac{1}{N} \sum_{k=1}^{N} \varphi_f(t_k)y_f^{(n)}(t_k) \right] \]

\[ \hat{\theta}_{lssvf} = \theta_o + \left[ \frac{1}{N} \sum_{k=1}^{N} \varphi_f(t_k)\varphi_f^T(t_k) \right]^{-1} \left[ \frac{1}{N} \sum_{k=1}^{N} \varphi_f(t_k)v_f(t_k) \right] \]
Basic LSSVF estimator – Statistical analysis

✓ Under weak conditions, the normalized sums tend to the corresponding expected values as N tends to infinity

✓ Hence

$$\hat{\theta}_{\text{LSSVF}} \xrightarrow{N \to \infty} \theta_0$$

if

$$\begin{cases} E \{ q_f(t_k)q_f^T(t_k) \} \text{ is nonsingular} \\ E \{ q_f(t_k)v_f(t_k) \} = 0 \end{cases}$$

- The first condition is satisfied in most cases
- The second condition is never satisfied
  - LSSVF estimates are always biased because of the correlation between the regression vector $q_f(t_k)$ and the noise $v_f(t_k)$
  - even if $v(t_k)$ is white noise, $v_f(t_k)$ becomes colored due to the SVF filtering

Simple LSSVF estimator – Conclusions

✓ Simple LSSVF method has some attractive properties
  - Simple, analytical solution easy to compute, low computational complexity

✓ Main shortcomings
  - **always biased** in noisy output measurement situations

$$\hat{\theta}_{\text{LSSVF}} \neq \theta_0 \text{ since } E \{ q_f(t_k)v_f(t_k) \} \neq 0$$

- **quite sensitive** to the SVF filter cut-off frequency

Motivation for studying more advanced methods

*We can do better!*
Traditional solutions to the LS bias problem

✓ Prediction Error Method (PEM)

✓ Instrumental Variable Method (IV)

Prediction Error Method (PEM)

✓ Main idea: model the noise!
✓ General approach applicable to a wide range of model structures
✓ Conditions to obtain optimal PEM estimates are well-established

\[
\hat{\theta}_{pem} = \arg \min_{\theta} \sum_{k=1}^{N} e^{2}(t_k, \theta) = \arg \min_{\theta} \sum_{k=1}^{N} \| y(t_k) - \hat{y}(t_k, \theta) \|^2
\]

✓ If assumptions about the noise valid: delivers optimal estimates
✓ Involves often solving a non-convex optimization problem
  - Relies on iterative nonlinear optimization (computationally quite demanding)
  - special care required for the initialization of the iterative search
  - may be trapped in false solutions that correspond to local minima
Instrumental Variable (IV) method

- Main idea: model the noise!
- General approach applicable to a wide range of model structures
- Conditions to obtain optimal IV estimates are well-established

\[
\hat{\delta}_{IV}^{opt} = \arg \min_{\delta} \sum_{k=1}^{N} \left\| z_{f}^{opt}(t_k)L^{opt}(p)\left[ y(t_k) - \varphi^{T}(t_k)\theta \right]\right\|_{Q}^2
\]

- Need to specify the instrument \( z_f \) and the prefilter \( L(p) \)
- If the assumptions about the noise are valid: delivers optimal estimates
- If assumptions about the noise are not valid: delivers unbiased estimates
- Based on (pseudo) linear regression (do not rely on nonlinear optimization)
  - low computational complexity (comparable to the LS method)

Surprisingly IV methods appear to be under-appreciated

Two main references for IV methods

- P.C. Young, Refined instrumental variable estimation: ML optimization of a unified BJ model, Automatica, 2015
Main idea: Instrumental Variable (IV) method

✓ Recap: LSSVF estimates always biased because of the correlation between the regression vector \( q_f(t_k) \) and the noise \( v_f(t_k) \)

✓ **Main idea of IV:** introduce a vector \( z_f(t_k) \) called instrument or instrumental variable which components are uncorrelated with \( v_f(t_k) \)

\[
E \{ z_f(t_k) v_f(t_k) \} = 0
\]

\[
\frac{1}{N} \sum_{k=1}^{N} z_f(t_k) v_f(t_k) = 0 \quad \text{with} \quad v_f(t_k) = y_f^{(n)}(t_k) - q_f^T(t_k) \theta
\]

\[
\hat{\theta}_{iv} = \text{sol}_{\theta} \left[ \frac{1}{N} \sum_{k=1}^{N} z_f(t_k) (y_f^{(n)}(t_k) - q_f^T(t_k) \theta) \right] = 0
\]

\[
\hat{\theta}_{iv} = \left[ \frac{1}{N} \sum_{k=1}^{N} z_f(t_k) q_f^T(t_k) \right]^{-1} \left[ \frac{1}{N} \sum_{k=1}^{N} z_f(t_k) y_f^{(n)}(t_k) \right]
\]

- How should the instrument \( z_f(t_k) \) be chosen?

Basic two step IV-based SVF estimator

✓ The instrument must be chosen so that it is:
  - not correlated with the measurement noise
  - sufficiently correlated with the filtered regression vector

\[
q_f^T(t_k) = L(p) \begin{bmatrix} -y_f^{(n-1)}(t_k) & \cdots & -y(t_k) & u^{(m)}(t_k) & \cdots & u(t_k) \end{bmatrix} \quad L(p) = \frac{1}{(p + \lambda)^{\lambda}}
\]

✓ In the basic IVSVF estimator, the instrument is built as

\[
z_f^T(t_k) = L(p) \begin{bmatrix} -\hat{x}^{(n-1)}(t_k) & \cdots & -\hat{x}(t_k) & u^{(m)}(t_k) & \cdots & u(t_k) \end{bmatrix}
\]

\[
\hat{x}(t_k) = G(p, \hat{\theta}_\text{lssvf}) u(t_k)
\]

is the estimated noise-free output calculated from an *a priori* LSSVF estimate

✓ The basic IV-based SVF estimator can then be computed from

\[
\hat{\theta}_{ivsf} = \left[ \frac{1}{N} \sum_{k=1}^{N} z_f(t_k) q_f^T(t_k) \right]^{-1} \left[ \frac{1}{N} \sum_{k=1}^{N} z_f(t_k) y_f^{(n)}(t_k) \right]
\]
### Two-step IV-based SVF estimator - Summary

**First step**
- \( B(p) \) and \( A(p) \)
- \( v(t_k) \)
- \( y(t_k) \)
- \( E(p) = (p + \lambda)^T \)
- LS algorithm
- \( \hat{\theta}_{lssvf} \)

**Second step**
- \( y(t_k) \)
- \( u(t_k) \)
- \( \hat{\theta}_{ivsvf} \)

**Bank of SVF filter**
- \( \hat{B}_{lssvf}(p) \)
- \( \hat{A}_{lssvf}(p) \)
- \( \hat{x}(t_k) \)

**Auxiliary model**
- \( \hat{\theta}_{lssvf} \)

### IVSVF estimator - Matlab implementation – Noise-free case

```
B=2;  % A(p)=p^2+4p+3 – True system
A=[1 4 3];
Ts=0.01;
N=1000;
t=(0:N-1)*Ts;
u=square(2*pi*0.5*t);
x=lsim(B,A,u,t);  % simulation of the noise-free output
y=x+0.2*randn(N,1);  % white noise added to the noise free output
data=iddata(y,u,Ts);idplot(data);
lambda=3;  % \( \lambda \) : cut-off frequency of the SVF filter
den_L=[1 2*lambda lambda^2];  % denominator of the SVF filter
num_L0=1;  % numerator of \( L_0(p)=1/(p+\lambda)^2 \)
num_L1=[1 0];  % numerator of \( L_1(p)=p/(p+\lambda)^2 \)
num_L2=[1 0 0];  % numerator of \( L_2(p)=p^2/(p+\lambda)^2 \)
yf0=lsim(num_L0,den_L,y,t);  % Computation of the filtered output
yf1=lsim(num_L1,den_L,y,t);  % Computation of the filtered output
yf2=lsim(num_L2,den_L,y,t);  % Computation of the filtered output
uf0=lsim(num_L0,den_L,u,t);  % Computation of the filtered input
phi_f=[-yf1 -yf0 uf0];  % Regression matrix
yf=yf2;  % Output vector
theta_lssvf=phi_f*yf  % LSSVF estimates
```

```
\[
\begin{array}{c}
\hat{x}(t_k) \\
\end{array}
\]
```

```
\[
\begin{array}{c}
\hat{\theta}_{lssvf} \\
\end{array}
\]
```

```
\[
\begin{array}{c}
\hat{\theta}_{ivsvf} \\
\end{array}
\]
```
Blssvf=theta_lssvf(3)';
Alssvf=[1 theta_lssvf(1:2)'];
xest=lsim(Blssvf,Alssvf,u,t); % simulation of the auxiliary model output
xestf0=lsim(num_L0,den_L,xest,t); % Computation of the filtered auxiliary model output
xestf1=lsim(num_L1,den_L,xest,t); % Computation of the filtered auxiliary model output
% time derivative

Z_f=[-xestf1 -xestf0 uf0]; % Instrumental variable
theta_ivsvf=(z_f*phi_f)'z_f*yf; % IVSVF estimates

3.8542  2.6055  1.9732 % see also the ivsvf routine in the CONTSID toolbox
% M_ivsvf=ivsvf(data,[2 1 0],lambda)

Frequency responses of the IVSVF models (Rao-Garnier benchmark)

\[ L(p) = \frac{1}{(p + \lambda)^2} \]
\[ \lambda = 13 \text{ rad / sec} \]
Basic IVSVF estimator – Conclusions

✓ Has some attractive properties
  ▪ simple
  ▪ analytical solution
  ▪ low computational complexity
  ▪ quite robust to the choice of the SVF filter
  ▪ unbiased estimates in output measurement noise situations

✓ BUT
  ▪ Parameter estimates are not minimum variance

✓ Motivation for studying more advanced IV methods

We still can do better!

How to choose the instrument to get the optimal estimates?

Optimal instrumental variable for COE models

✓ Consider a Continuous-time Output Error (COE) model

\[
\begin{align*}
  u(t_k) &\rightarrow G_o(p) & x(t_k) &\rightarrow e(t_k) & y(t_k) \\
  x(t_k) &= \frac{B_o(p)}{A_o(p)} u(t_k) \\
  y(t_k) &= x(t_k) + e(t_k)
\end{align*}
\]

✓ The optimal (unbiased and minimal variance) choice for the instrument is

\[
\left\{ \begin{array}{l}
  z^{opt}(t_k) = L^{opt}(p) q_o(t_k) \\
  L^{opt}(p) = \frac{1}{A_o(p)}
\end{array} \right.
\]

where \( q_o(t_k) \) represents the noise-free version of the regression vector

\[
q_o(t_k) = \begin{bmatrix} -x^{(n-1)}(t_k) & \cdots & -x^{(0)}(t_k) & u^{(m)}(t_k) & \cdots & u^{(0)}(t_k) \end{bmatrix}^T
\]

\( x(t_k) \): noise-free output
Successful implementation of the optimal IV: the Simplified Refined IV method (SRIVC)

There is one major problem with the optimal solution
- The optimal pre-filter and noise-free output are not known a priori!!

Solution
- use of an iterative procedure that adjusts an initial estimate of the parameter vector until it converges on an optimal estimate

\[
\hat{\theta}_{srivc}^{i+1} = \left[ \sum_{k=1}^{N} z(t_k, \hat{\theta}^i) \psi^T(t_k, \hat{\theta}^i) \right]^{-1} z(t_k, \hat{\theta}^i) y(t_k) ^{(n)}(t_k)
\]

\[
L(p) \text{ has two roles here: optimal IV role and generate the time-derivatives}
\]

\[
\psi(t_k, \hat{\theta}^i) = \left[ -x(t_k, \hat{\theta}^i) \cdots -x^{(n-1)}(t_k, \hat{\theta}^i) u^{(m)}(t_k) \cdots u^{(0)}(t_k) \right]^T
\]

\[
x(t_k, \hat{\theta}^i) = G(p, \hat{\theta}^i) u(t_k)
\]

Optimal SRIVC method for CT OE models - Summary
Statistical properties of SRIVC estimates

✓ Data-generating system

\[ y(t_k) = G_o(p)u(t_k) + e(t_k) \]

or

\[ y(t_k) = \varphi^T(t_k)\theta_o + \nu(t_k) \]

✓ When the system is in the model set, we have for the **SRIVC estimates**

✓ Mean

\[ E(\hat{\theta}_{srivc}) = \theta_o \]

✓ Covariance matrix

\[
P_{\hat{\theta}_{srivc}} = E\left\{ (\hat{\theta}_{srivc} - \theta_o) (\hat{\theta}_{srivc} - \theta_o)^T \right\} = \sigma_e^2 \left( \sum_{k=1}^{N} z_f(t_k, \hat{\theta}_{srivc}) z_f^T(t_k, \hat{\theta}_{srivc}) \right)^{-1}
\]

Frequency responses of the stable SRIVC models (Rao-Garnier benchmark - white noise)

Unbiased and minimum variance (optimal) estimates. **Cannot do better!”**
To sump up

- **Simple LSSVF**: always biased
- **Basic IVSF**: unbiased but not minimum variance
- **Optimal SRIVC**: the best!

SRIVC particularly reliable and robust method: recommended as a first choice in practice

---

Model structure/order determination

- **Start**
- **Experimental Design and Execution**
  - (Step, Pulse, or PRBS-Generated Data)
- **"Identification"**
  - Data Preprocessing
  - Model Structure Determination
  - Parameter Estimation
  - (Linear Plant and Disturbance Models)
- **Model Validation**
  - (Simulation, Residual auto and cross-correlation, step response)
  - Does the model meet validation criteria?
  - **Yes**
  - **No**

University of Lorraine, ESSTIN, January 2016 – H. Garnier
Model structure/order determination

✓ One of the most difficult stage of the system identification procedure

✓ Recommended procedure

1. Start with non-parametric estimates (impulse, step or frequency-responses)
   • Can provide useful information about the time-delay, system order and bandwidth

2. Begin with COE model structure
   \[
   \begin{align*}
   x(t) &= G(p)u(t - \tau) \\
   y(t_k) &= x(t_k) + e(t_k)
   \end{align*}
   \]

3. Select polynomial degrees and time-delay according to the flexibility/parsimony trade-off

4. Try more complex model structures if necessary (Box-Jenkins model)

Flexibility-parsimony trade-off

- Measure of good fit
  - the Mean Squared Error (MSE)
    \[
    V(\hat{\theta}_{n_p}, Z^N) = \frac{1}{N} \sum_{k=1}^{N} e^2(t_k) = \frac{1}{N} \sum_{k=1}^{N} [y(t_k) - \hat{y}(t_k, \hat{\theta}_{n_p})]^2
    \]

  Fig. 1. Typical relationship between total squared error, squared bias, variance and degrees of freedom in the model.

- MSE consists of a bias contribution and a variance contribution
  - more flexible model => reduce the bias but increase the variance
  - more parsimionous model => reduce the variance but increase the bias

- Trade-off between flexibility and parsimony
  - there is no ideal solution/method
In practice, it is easy to identify a model for a range of possible model orders and select the best order for which a criterion $J(n_p, Z^N)$ is minimal or does not decrease significantly anymore.

$$\hat{n}_{p}^{\text{best}} = \arg \min_{n_p} \left( J(n_p, Z^N) \right)$$

$$n_p = n + m : \text{number of parameters to be estimated}$$

### Model order selection – Practical considerations

- In practice, it is easy to identify a model for a range of possible model orders and select the best order for which a criterion $J(n_p, Z^N)$ is minimal or does not decrease significantly anymore.

### Traditional criteria for model order selection

- The criterion $J(n_p, Z^N)$ is formulated from two functions:
  - one term measuring the model fit based on the loss function
  - one term penalizing the model complexity

$$J(n_p, Z^N) = \log V(\hat{\theta}_{n_p}, Z^N) + \beta(n_p, Z^N)$$

- $\beta(n_p, Z^N)$ is a function which should increase with the model order but decrease to zero when $N \rightarrow \infty$

- Basic approach: pick the model that minimizes
  - Akaike’s Information Criteria (AIC)
  - Final Prediction Error (FPE)

- Both criteria tend to over-estimate the model order
Recommended SRIVC-based model order selection approach

✓ Computation of two criteria
  ▪ Coefficient of determination (measure of the model fit)
    \[ R_T^2 = 1 - \frac{\sigma_\varepsilon^2}{\sigma_y^2} \] 
    \( \varepsilon(t_k) = y(t_k) - \hat{y}(t_k, \hat{\theta}_{np}) \)
  ▪ Young Information Criteria (YIC) (parcimonious principle)
    \[ YIC = \log \left( \frac{\sigma_\varepsilon^2}{\sigma_y^2} \right) + \log \frac{1}{n_p} \sum_{j=1}^{n_p} \frac{\sigma_\varepsilon^2 \hat{P}_{jj}}{\hat{\theta}_j^2} \]
    \( \hat{P}_{jj} \): \( j \) – th diagonal element of \( \hat{P}_\theta \)
    \( \hat{\theta}_j^2 \): square of the \( j \) – th SRIVC parameter
    \( n_p \): number of parameters to be estimated

The covariance matrix of the SRIVC parameter estimates is taken into account

✓ Optimisation of \( R_T^2 \) only: over-parametrisation of the model
✓ Optimisation of YIC only: under-parametrisation of the model

Recommended approach

▪ Select the model order that has the most negative YIC with the highest associated \( R_T^2 \)

Example: Rao-Garnier benchmark

Different model structures in the range [nb nf nk] = [1 3 0] to [2 5 0] have been computed for a data set

<table>
<thead>
<tr>
<th>np</th>
<th>nb</th>
<th>nf</th>
<th>nk</th>
<th>RT2</th>
<th>YIC</th>
<th>Niter</th>
<th>FPE</th>
<th>AIC</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>2</td>
<td>3</td>
<td>0</td>
<td>0.83</td>
<td>-8.33</td>
<td>10</td>
<td>2.35</td>
<td>0.85</td>
</tr>
<tr>
<td>6</td>
<td>2</td>
<td>4</td>
<td>0</td>
<td>0.92</td>
<td>-8.19</td>
<td>5</td>
<td>1.13</td>
<td>0.12</td>
</tr>
<tr>
<td>7</td>
<td>1</td>
<td>5</td>
<td>0</td>
<td>0.53</td>
<td>-7.51</td>
<td>10</td>
<td>6.90</td>
<td>1.93</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>3</td>
<td>0</td>
<td>0.51</td>
<td>-3.72</td>
<td>10</td>
<td>8.40</td>
<td>2.12</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>4</td>
<td>0</td>
<td>0.03</td>
<td>-0.89</td>
<td>10</td>
<td>13.9</td>
<td>2.63</td>
</tr>
</tbody>
</table>

The second model with [nb nf nk] = [2 4 0] seems to be quite clear cut
It has the second most negative YIC = -8.19, with the highest associated RT2 = 0.92

See srivcstruc and selcstruc in the CONTSID toolbox
Principles of model validation

✓ Try to verify that observations behave according to modelling assumptions

✓ Recommended tests

1. Compare simulated model output with measured data

2. Compare step or frequency responses of the identified model with measured responses if available

3. Interpret the main features of the identified model in a physical sense

4. Perform statistical tests on prediction errors
   Residual auto and cross-correlation analysis

SRIVC - Extensions to more advanced situations

✓ SRIVC has proven to be particularly reliable in the day-to-day use

✓ SRIVC has been extended to handle wider practical applications

- Multiple input (MISO) systems
- Frequency domain
- Box-Jenkins models
- Simple process models with delay
- Systems operating in closed loop
- Linear Time-Varying (LTV) systems via recursive algorithms
- Block-oriented nonlinear systems
- Linear parameter varying (LPV) systems
- Fractional order systems
- Partial differential equations
- Non-uniformly sampled data
- …
Extension – Frequency domain

✓ CT model identification can also be done in the frequency domain
  ▪ Similar iterative IV procedure can be used

\[ \hat{\theta} = \arg \min_{\theta} \sum_{k=1}^{N} (G(\omega_k, \theta) - \hat{G}(\omega_k))^2 \]

\[ \hat{\theta} = \arg \min_{\theta} \sum_{k=1}^{N} \left( \frac{G(\omega_k, \theta) - Y(\omega_k)}{U(\omega_k)} \right)^2 \]

➢ SRIVC: naturally extended to work with frequency-domain data

---

M. Gilson, J. Welsh, H. Garnier, Frequency-domain IV based method for wide band system identification. ACC’2013

---

Extension – Simple process models

✓ In industrial practice
  ▪ identification for control: often reduced to the determination of simple process models of low order

\[ G(s) = \frac{K e^{-T_d s}}{1 + Ts} \]

\[ G(s) = \frac{K (1 + T_z s) e^{-T_p s}}{(1 + T_{p1}s)(1 + T_{p2}s)} \]

✓ Solutions
  ▪ Many ad hoc methods exist

➢ SRIVC naturally adapted to identify simple process models from step responses and similar

---

Extension – Colored noise situation

✓ In practice
  - Measurement noise is colored: does not have the nice white properties
  - SRIVC still yields unbiased estimates with low but not minimum variance
    • prefilters are not designed to account for the colour of the noise

✓ Solution
  - RIVC: optimal IV approach for CT hybrid Box-Jenkins models
    • CT model for the plant + DT model for the noise

\[
\begin{align*}
  y(t_k) &= G(p)u(t_k) + H(q)e(t_k) \\
  y(t_k) &= \frac{B(p)}{A(p)}u(t_k) + \frac{C(q)}{D(q)}e(t_k)
\end{align*}
\]


Extension – Closed-loop identification

✓ In industrial practice
  - Many systems have feedback that cannot be interrupted for an identification experiment

\[
\begin{align*}
  y(t) &= G(p)u(t) + v(t) \\
  u(t) &= r(t) + C(p)(y_c(t) - y(t))
\end{align*}
\]

- To get reliable estimates, IV methods have to be adapted:
  - input is contaminated by noise

✓ Solutions
  - **Optimal SRIVC/RIVC method for closed-loop identification**
  
  - **Two stage-based SRIVC approach**
Extension – Linear time-varying (LTV) systems

✓ In practice
  - The nature of the system is time-varying
  - A time-invariant model built on initial operating data would obviously be unable to capture the changing system characteristics

✓ First solution
  - Use LTI model with time-varying parameters
    - Recursive SRIVC version for tracking the varying parameters

Extension – Linear parameter varying (LPV) systems

✓ In industrial practice
  - The nature of the system is time-varying

✓ Second solution
  - Use a LPV model to parametrize the time-varying nature of the parameters

\[
\begin{align*}
\sum_{i=1}^{n} a_i(t) p(t) x(t) &= \sum_{i=0}^{m} b_i(t) p(t) u(t) \\
\frac{d}{dt} p(t) &= x(t_k) + v(t_k)
\end{align*}
\]

- SRIVC/RIVC methods for LPV input/output models


1. Introduction to Data-based Modelling
   • Motivation, goals and application areas
   • General system identification procedure
   • Overview of discrete and continuous-time approaches

2. Direct Continuous-time Methods
   • Basic issues
   • Traditional SVF-based estimators
   • Optimal instrumental variable estimators
   • Recent extensions to handle more complicated situations

3. Software Aspects and Benefits in Practical Applications
   • The CONTSID toolbox. A guided tour
   • Advantages of the continuous-time approaches
   • A selection of successful practical applications

Software aspects

✓ Several actively maintained toolboxes are available
  ▪ Comprehensive Mathworks SID toolbox (L. Ljung)
  ▪ FDIDENT toolbox (I. Kollar, J. Schoukens)
  ▪ UNIT toolbox (B. Ninness)
  ▪ CAPTAIN toolbox (P. Young)

✓ No software entirely dedicated to direct CT approaches
  ▪ first released in 1999
CONTSID toolbox - Key features

- Supports most of the time and frequency-domain methods for identifying CT parametric models of linear and nonlinear dynamic systems from regularly or irregularly sampled data.

- Provides transfer function model identification for SISO and MIMO systems:
  - mainly based on the iterative SRIVC algorithm
  - includes also a few PEM and subspace-based methods

- May be seen as an add-on to the Matlab SID toolbox
  - use of the same syntax, same data and same model objects
  - current version 7.0: compatible with the latest release 2015b of Matlab

- Is freely available for academics from: www.cran.univ-lorraine.fr/contsid

- Includes a flexible GUI and many demos to illustrate its use and the recent developments

CONTSID toolbox – Demonstration programs

>>idcdemo

Advantages of the CONTSID toolbox methods

- Identified Parameters Are Closer to the Physical Coefficients
- Can Cope with Non-uniformly Sampled Data
- Are Ideally Suited for Stiff Dynamic Systems
- Can Cope Easily with Fast Sampled Data
- Include Inherent Data Filtering
- Make the Identification Procedure Easier for the User
- Are Robust Against Measurement Setup Assumption

More Advanced Identification
Quit
CONTSID GUI

- Allows the user to easily apply the iterative process of system identification

Successful applications in different fields

CONTSID has proven successful in many practical applications

- **Robotics**
  - Complex robot arm modelling

- **System biology**
  - Uptake kinetics of a photosensitising agent into cancer cells

- **Electro-mechanical field**
  - Multivariable winding pilot plant

- **Chemical engineering**
  - Industrial binary distillation column

- **Biomedical systems**
  - Robot arm designed for heart beating movement compensation

- **Environmental sciences**
  - Rainfall/flow modelling
  - Pollutant transport in river systems
What have CT models to offer?

CT models have certain advantages in relation to their equivalent DT models

- Are more intuitive to control engineers in their every-day practice
- Many practical control design are still based on CT models
- CT models are often preferred for fault detection
  - reveal faults more directly than their DT counterparts
- Parameter values are independent of $T_s$

$$G_0(p) = \frac{1}{p^2 + p + 1}$$

$$T_s = 0.1s; \quad G_T(q) = \frac{0.0048q^{-1} + 0.0047q^{-2}}{1 - 1.8953q^{-1} + 0.9048q^{-2}}$$

$$T_s = 1s; \quad G_T(q) = \frac{0.3403q^{-1} + 0.2417q^{-2}}{1 - 0.7849q^{-1} + 0.3679q^{-2}}$$

What have direct CT methods to offer?

CT methods present many advantages in relation to their equivalent DT methods

- include inherent data prefiltering
- are well-suited to fast sampling situations
- are well-adapted to identify stiff systems
- can cope easily with irregularly sampled data
Irregularly sampled data

✓ Appears in many situations: in biomedical or environmental systems, in transport and traffic systems, ...
  - losses in data transmission
  - manual measurements
  - event-based sampling
    - signals are sampled only when they pass certain limits

CT identification works with irregularly sampled data

✓ A constant $T_s$ is assumed for DT models

$$
\begin{bmatrix}
y(t_3) \\
y(t_4) \\
\vdots \\
y(t_N)
\end{bmatrix}
= 
\begin{bmatrix}
-y(t_2) & -y(t_1) & u(t_2) & u(t_1) \\
-y(t_2) & -y(t_1) & u(t_3) & u(t_2) \\
\vdots & \vdots & \vdots & \vdots \\
-y(t_{N-2}) & -y(t_{N-1}) & u(t_{N-1}) & u(t_{N-2})
\end{bmatrix}
\begin{bmatrix}
a_1 \\
a_2 \\
b_1 \\
b_2
\end{bmatrix}
+ 
\begin{bmatrix}
e(t_3) \\
e(t_4) \\
\vdots \\
e(t_N)
\end{bmatrix}
$$

- when $T_s$ varies, DT model parameters become time-varying
  - Estimation still possible but much more difficult

✓ CT models represent the system at every $t_k$

$$
\begin{bmatrix}
y^{(2)}_1(t_1) \\
y^{(2)}_1(t_2) \\
\vdots \\
y^{(2)}_1(t_N)
\end{bmatrix}
= 
\begin{bmatrix}
-y^{(1)}_1(t_1) & -y^{(1)}_1(t_1) & u(r(t_1)) \\
y^{(1)}_1(t_2) & -y^{(1)}_1(t_2) & u(r(t_2)) \\
\vdots & \vdots & \vdots \\
y^{(1)}_1(t_N) & -y^{(1)}_1(t_N) & u(r(t_N))
\end{bmatrix}
\begin{bmatrix}
a_1 \\
a_2 \\
b_0
\end{bmatrix}
+ 
\begin{bmatrix}
e_1(t_1) \\
e_1(t_2) \\
\vdots \\
e_1(t_N)
\end{bmatrix}
$$

- The time-instants do not need to be equidistantly spaced
  - Appropriate ODE solver for the digital implementation of the CT filtering operations is only required
Example: tracer experiment identification from irregularly sampled data

✓ Data from a real-life tracer experiment in a wetland located in Florida

- Tracer experiment: traditional experiment in environmental science to *model transport and dispersion of pollutants* in rivers and wetlands
- Conservative potassium bromide (KBr): used as the tracer material
- Irregularly sampled data arise because
  - the sampling is rapid to start with, so that the peak of the event is captured well
  - the samples are then taken more slowly over the subsequent slow recession

Dispersion modelling of a pollutant in a wetland

*Irregular sampling of the data is clear, with $T_s$ varying from 2 to 40 h*
SRIVC model identification from irregularly sampled data

\[ R^2_T = 0.99 \]

- SRIVC model explains the data very well
- 99% of the downstream concentration is explained by the SRIVC model output

Identification of stiff systems

Stiff systems are challenging: combine both slow & fast dynamics

- makes the selection of \( T_s \) difficult
  - requires rapid sampling to capture the fast dynamics
  - requires slow sampling to identify the slow dynamics
  - these requirements are incompatible!
  - need for a compromise: normally, the selection of fast sampling
CT identification more easily identifies stiff systems

In case of **rapidly sampled data for stiff systems**, fitting an accurate model is not straightforward
- both algorithmic and numerical aspects play a role

✔ DT methods
  - poles of the estimated DT model will be very close to the unit circle
  - possible numerical issues might affect the estimated DT model

✔ CT methods
  - are particularly well-suited to **fast sampling** situations
  - work well with rapidly sampled data from **stiff** systems

✔ Typical example of stiff systems: Light-emitting diodes (LED)

Light-emitting diodes (LED) are everywhere

✔ Semi-conductors: more and more used in applications as diverse as
  - automotive lighting
  - general lighting
  - traffic signals,…

✔ LED lamps have a lifespan and electrical efficiency that is several times better than incandescent lamps
Example: thermal stiff response of LEDs

✓ Typical example of stiff response
  • thermal transient response of High-powered light-emitting-diodes (LEDs)
  • combine both slow and fast thermal phenomena

✓ Transient LED response can be used to detect possible thermal defects
  • LED driven by a current step source and the thermal stiff response is measured
  • Goal: to fit a model from the data that captures the stiff behavior

Thermal step response of the LED

Fast sampling: $f_s=50$ kHz, $T_s=0.02$ ms

Large number of data: $N=98000$
Step responses of the different estimated models

\[ G(s) = \sum_{i=1}^{6} \frac{K_i}{1 + \tau_i s} \]

LED temperature (°C)

Measured output
CT SRIVC model output
CT COE model output
CT TFEST model output
DT OE model output

Time (seconds)

Conclusion

✓ For more than 40 years, the general mainstream identification approach has been to identify DT models from sampled data

✓ To obtain satisfying results, **DT model identification**
  - requires the active participation of an experienced practitioner to pre-process the data (decimation or prefiltering)
  - accentuates the feeling that system identification is "more an art than a science"!

✓ We argue that **direct CT model identification** has been under-appreciated although it includes many advantages:
  - well adapted to **irregularly** and **rapidly** sampling situations
  - requires less participation from the user (**inherent pre-filtering**)
  - makes the **application** of the SYSID procedure much **easier**
Conclusion

Amongst the different CT methods, one is particularly recommended

- **SRIVC**: iterative IV method
  - robust to assumptions and algorithmic aspects
  - extended to handle wider practical applications: colored noise, closed loop, frequency-domain, TVP, LPV systems, ...

Direct CT methods now available as user-friendly algorithms in:

- **CONTSID** toolbox
- Matlab SID toolbox,…

*It is hoped that the use of these direct CT approaches will lead to a better appreciation of data-driven system identification in the general systems and control community*

To know more

- See for this lecture, in particular, Chapters 1, 4 and 9
  - H. Garnier and L. Wang (Eds.)
  - *Identification of CT models from sampled data*, Springer-Verlag, 2008
- H. Garnier, T. Söderström and J. Yuz (Guest editors)
  - Special issue on «CT model identification», *IET Control Theory & Applications*, May 2011
- H. Garnier, P.C. Young (Guest editors)