Multivariable fractional system identification

Abir MAYOUFI, Stéphane VICTOR, Rachid MALTI, Manel CHETOUI, Mohamed AOUN

25/11/2021

Outline

1 Problem formulation

2 Two methods for MISO fractional system identification

- Two-stage algorithm
- Output error method

3 Simulation examples

4 Conclusions and prospects

Outline

1 Problem formulation

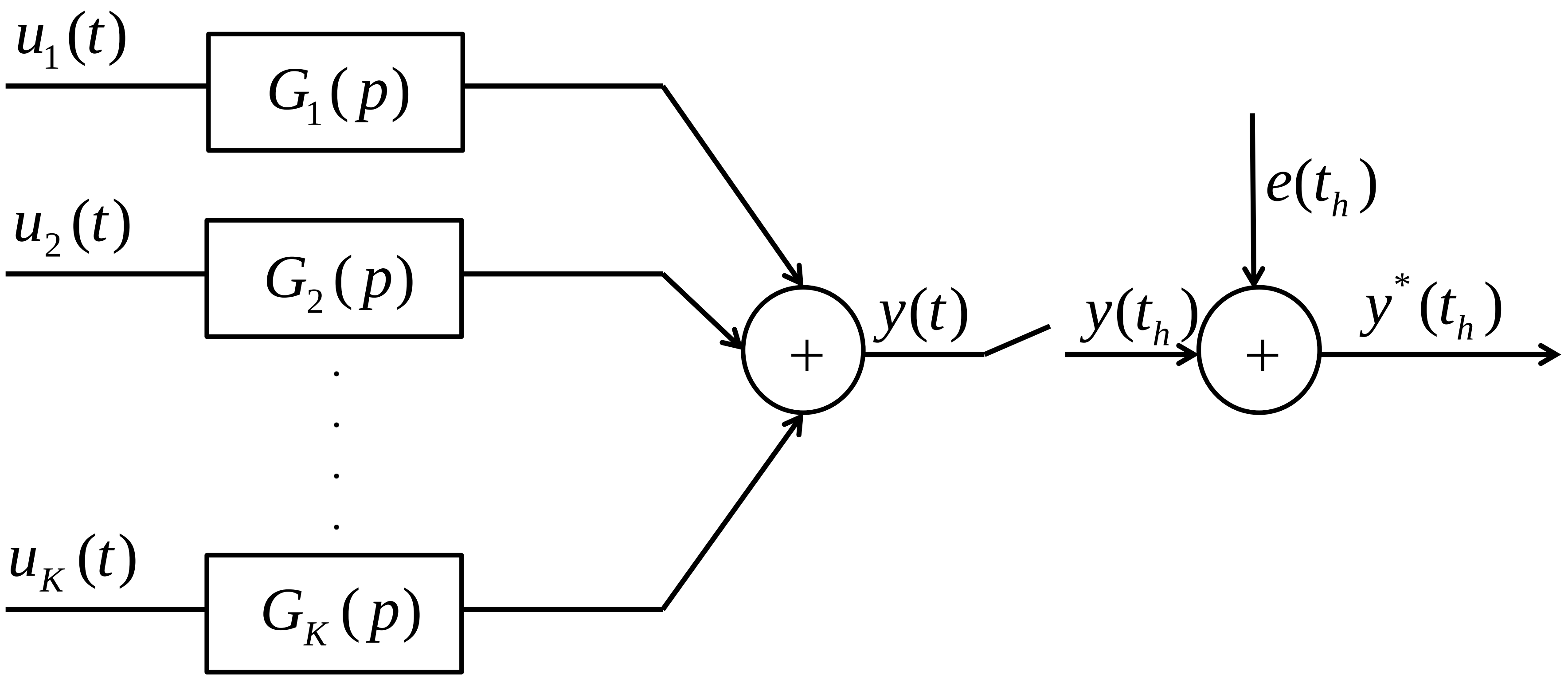
2 Two methods for MISO fractional system identification

3 Simulation examples

4 Conclusions and prospects

Problem formulation

- Consider a MISO fractional system



Problem formulation

- The MISO fractional system is represented by:

Number of inputs

The fractional transfer function

An additive white noise

$$\begin{cases} y_k(t) = G_k(p) u_k(t) \\ y(t) = \sum_{k=1}^{K} y_k(t) \\ y^*(t_h) = y(t_h) + e(t_h) \end{cases}$$

Differentiation orders

$$G_k(p) = \frac{B_k(p, \theta_k)}{A_k(p, \theta_k)} = \frac{\sum_{j=0}^{M_k} b_{j,k} p^{\beta_{j,k}}}{1 + \sum_{i=1}^{N_k} a_{i,k} p^{\alpha_{i,k}}}$$

$$(p = \frac{d}{dt})$$

Problem formulation

Structured-commensurability (S-commensurability)

The fractional transfer function S-commensurable

$$G_k(p) = \frac{B_k(p,\theta_k)}{A_k(p,\theta_k)} = \frac{\sum_{j=0}^{M_k} b_{j,k} p^{\beta_{j,k}}}{1 + \sum_{i=1}^{N_k} a_{i,k} p^{\alpha_{i,k}}}$$

The fractional global S-commensurate order

$$G_k(p) = \frac{\sum_{j=0}^{M_k} b_{j,k} p^{j\nu}}{1 + \sum_{i=1}^{N_k} a_{i,k} p^{i\nu}}, \nu \in \mathbb{R}^+$$

The fractional local S-commensurate order

$$G_k(p) = \frac{\sum_{j=0}^{M_k} b_{j,k} p^{j\nu_k}}{1 + \sum_{i=1}^{N_k} a_{i,k} p^{i\nu_k}}, \nu_k \in \mathbb{R}^+$$

Problem formulation

Objective: estimate the parameter vector θ of the MISO model system

The parameter vector θ is defined as :

$$\theta = [\rho \ \mu]^T$$

- ρ gathers all the MISO transfer function coefficients : $\rho = [\rho_1, \ldots, \rho_K]^T$

  with $$\rho_k = \left[b_{0,k}, b_{1,k}, \ldots, b_{M_k,k}, a_{1,k}, \ldots, a_{N_k,k}\right], \quad k = 1, \ldots, K$$

- μ gathers all the MISO transfer function differentiation orders :
  - Case1: if a global S-commensurate order is sought :

    $$\mu = \nu$$

  - Case2: if local S-commensurate order is sought :

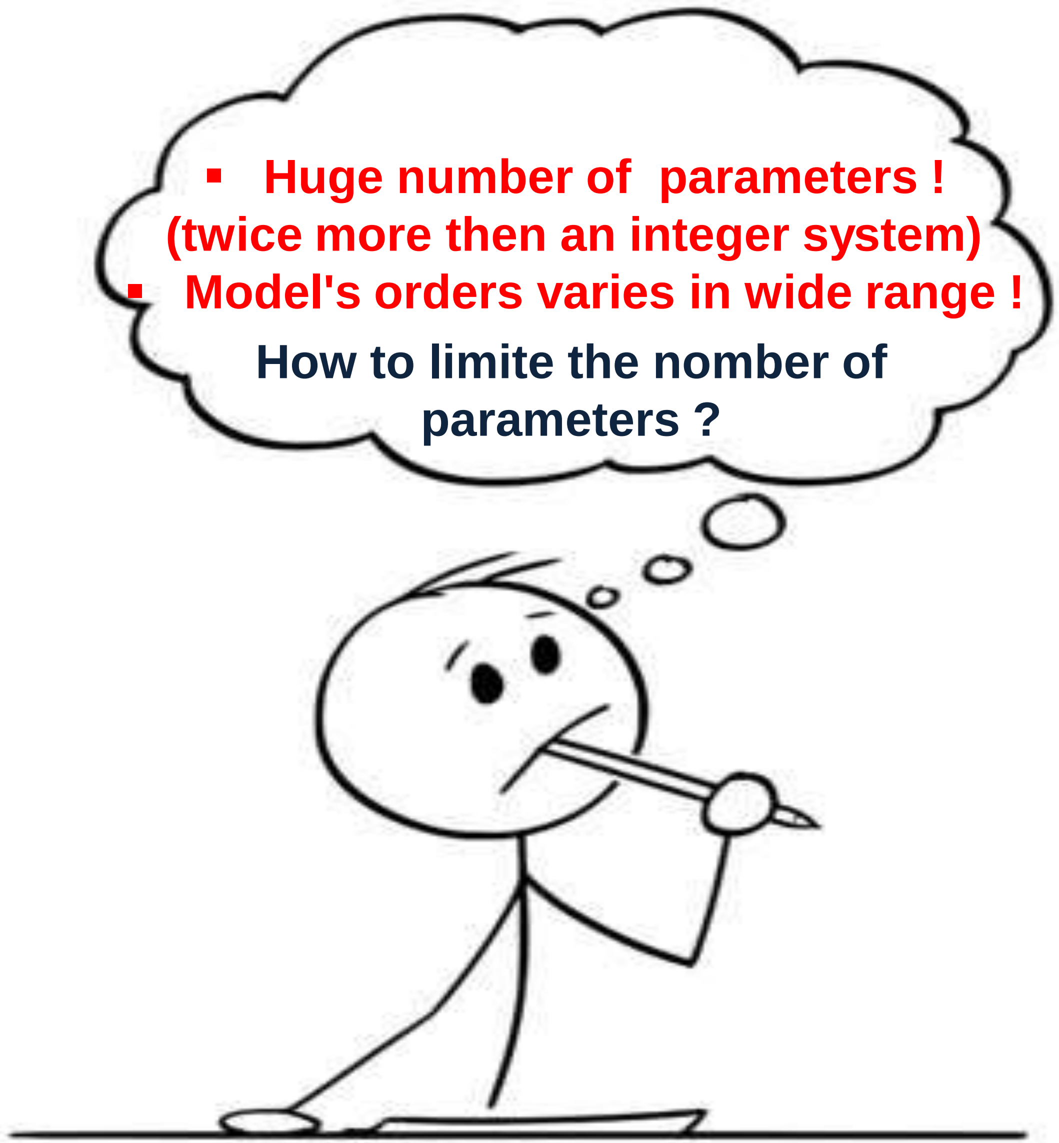
    $$\mu = [\nu_1, \ldots, \nu_K]^T$$

  - Case3: if the MISO model is non commensurate, μ gathers all the differentiation orders :

    $$\mu = [\mu_1, \ldots, \mu_K]^T$$

    with $$\mu_k = \left[\beta_{0,k}, \beta_{1,k}, \ldots, \beta_{M_k,k}, \alpha_{1,k}, \ldots, \alpha_{N_k,k}\right], \quad k = 1, \ldots, K$$

Problem formulation



Initializiation of differentiation orders

Global S-commensurate order

↓

Local S-commensurate order

↓

All differentiation orders

Outline

1 Problem formulation

2 Two methods for MISO fractional system identification

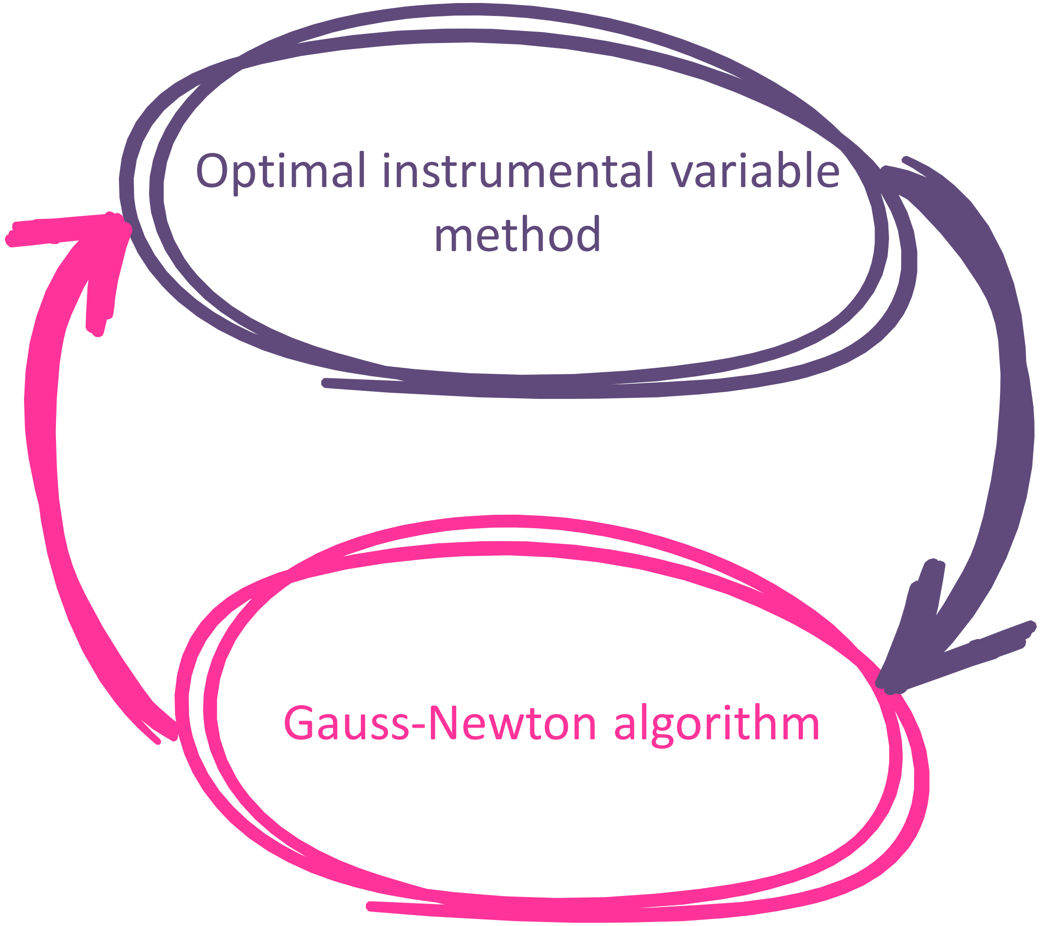
- Two-stage algorithm
- Output error method

3 Simulation examples

4 Conclusions and prospects

Two-stage algorithm

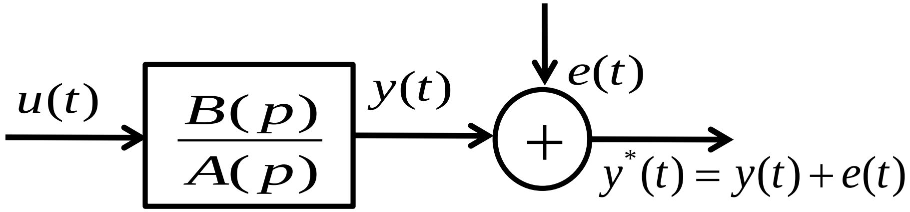
Two-stage algorithm



Optimal instrumental variable method for coefficient estimations

SISO model

The differentiation orders are assumed known



Equation error :

$$\varepsilon(t) = y^*(t) - \varphi^*(t)^T \rho$$

where

$$\varphi^*(t)^T = \begin{bmatrix} p^{\beta_0}u(t) \; \dots \; p^{\beta_N}u(t) \\ \text{-}p^{\alpha_1}y^*(t) \; \dots \; \text{-}p^{\alpha_M}y^*(t) \end{bmatrix}$$

with

$$\rho^T = \left[b_0 \;\; b_1 \;\; \dots \;\; b_N \;\; a_1 \;\; \dots \;\; a_M \right]$$

Approximation Least Squares estimates

$$\hat{\rho}_{LS} = \left(\phi^{*T} \phi^* \right)^{-1} \phi^{*T} Y^*$$

with

$$\phi^* = \left[\varphi^*(t_1) \dots \varphi^*(t_H) \right]$$

Limitation :

1. Computation of time domain derivatives of inputs/output signal
2. Amplification of noise in high frequencies

Filtered derivatives (ex: Poisson's filter)

$$^{P}D_{\gamma} = \frac{p^{\gamma}}{1 + \left(\frac{p}{\omega_c} \right)^N}$$

$$\varepsilon_f(t) = y_f^* - \varphi_f^* \rho$$

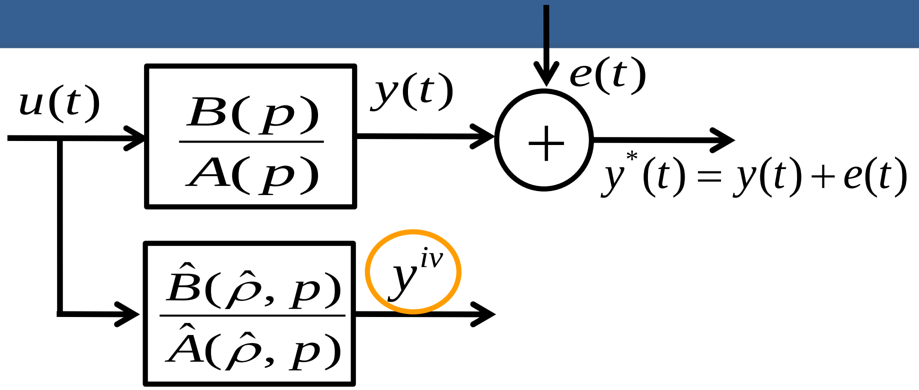
$$\hat{\rho} = \left(\phi_f^{*T} \phi_f^* \right)^{-1} \phi_f^{*T} Y^*$$

Limitation :

1. biased estimation in presence of output noise

Optimal instrumental variable method for coefficient estimations

SISO model



- Instrumental variable estimator

$$\hat{\rho}_{srivcf} = \left(\phi_f^{ivT}\phi_f^{*}\right)^{-1}\phi_f^{ivT}Y_f^{*}$$

- with the constraints (Ljung, 1999) :

$$\begin{cases}\phi_f^{ivT}\phi_f^{*} & : \text{non-singular}\\ \phi_f^{ivT}Y_f^{*} & : \text{uncorrelated}\end{cases}$$

- Young showed that the optimal IV estimator is :

$$F^{opt}(p) = \frac{1}{A(p)} \quad \begin{cases}\text{unbaised estimation}\\ \text{minimum variance}\end{cases}$$

However $A(p)$ is usually unknown

$$\varphi_f^{*}(t)^T = F(p)\varphi^{*}(t)^T = \begin{bmatrix} p^{\beta_0}u_f(t) & \dots\ p^{\beta_N}u_f(t) \\ \text{-}p^{\alpha_1}y_f^{*}(t) & \dots\ \text{-}p^{\alpha_M}y_f^{*}(t)\end{bmatrix}$$

$$\varphi_f^{iv}(t)^T = F(p)\varphi^{iv}(t)^T = \begin{bmatrix} p^{\beta_0}u_f(t) & \dots\ p^{\beta_N}u_f(t) \\ \text{-}p^{\alpha_1}y_f^{iv}(t) & \dots\ \text{-}p^{\alpha_M}y_f^{iv}(t)\end{bmatrix}$$

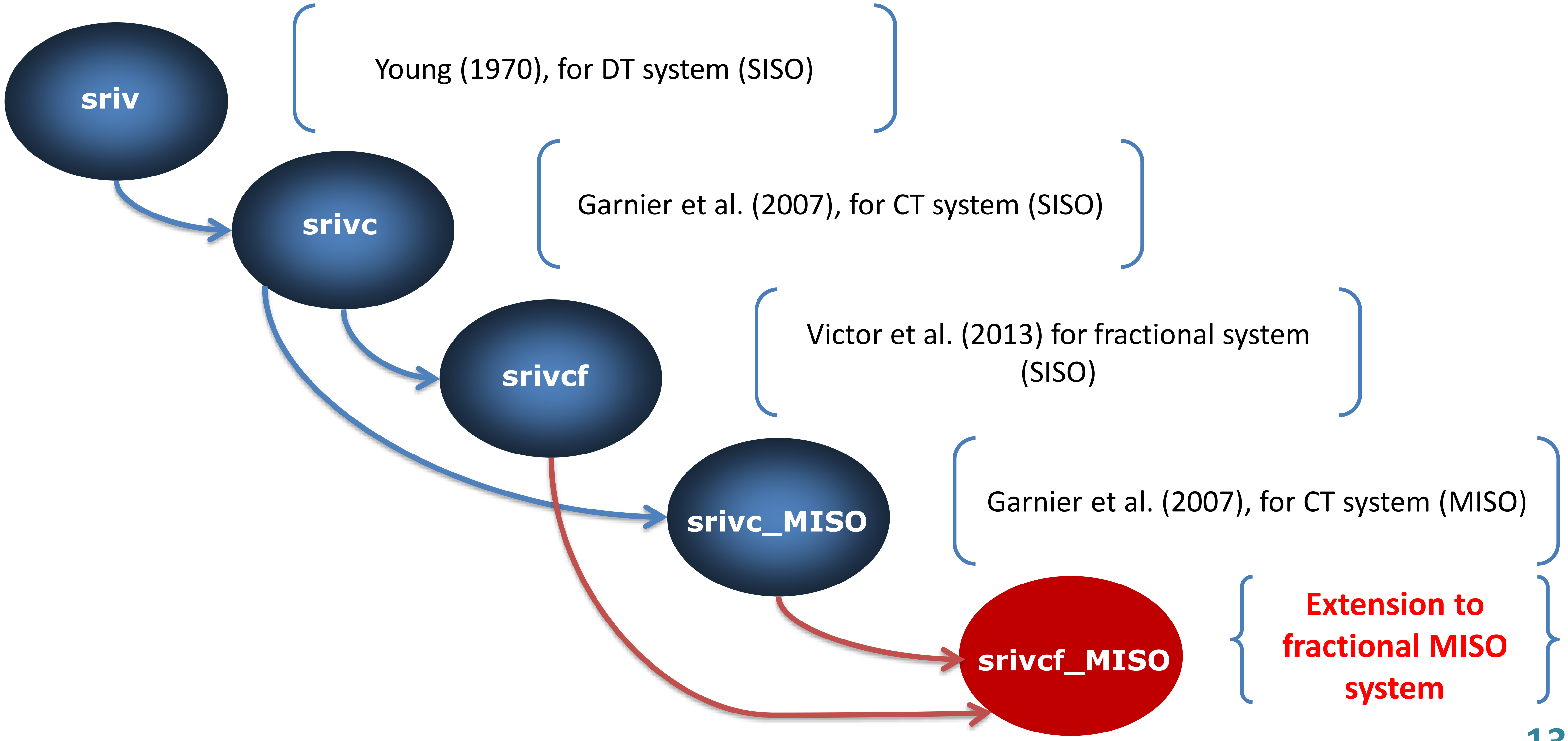
where the instrumental matrix is :

$$\Phi_f^{iv}(t) = \left[\varphi_f^{iv}(t_1)^T, \dots, \varphi_f^{iv}(t_H)^T\right]^T$$

and the regression matrix is :

$$\Phi_f^{*}(t) = \left[\varphi_f^{*}(t_1)^T, \dots, \varphi_f^{*}(t_H)^T\right]^T$$

Optimal instrumental variable method for coefficient estimation



Optimal instrumental variable method for coefficient estimations

MISO model

In this case, the error function takes the following form:

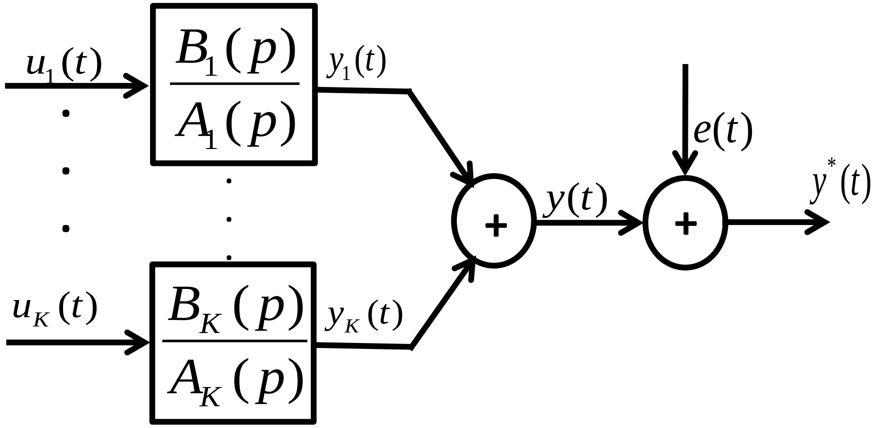
$$\varepsilon_k(t) = x_k(t) - y_k(t),\; k = 1,\dots,K$$

where y_k is the output of the k-th-subsystem:

and

$$x_k(t) = y^*(t) - \sum_{\substack{m=1\\ m\neq k}}^{K} y_m(t),\; k = 1,\dots,K$$

The main idea is to isolate the output of the k-th-subsystem and applicate the SISO algorithm.



The error function can also be written as:

$$\varepsilon_k(t) = x_k(t) - \varphi_k(t)\rho_k,\; k = 1,\dots,K$$

with $\varphi_k(t) = \begin{bmatrix} p^{\beta_{0,k}} u_k(t) \dots p^{\beta_{M_k,k}} u_k(t) \\ -p^{\alpha_{1,k}} x_k(t) \dots -p^{\alpha_{N_k,k}} x_k(t) \end{bmatrix}^T$

Optimal instrumental variable method for coefficient estimations

MISO model

Implement an iterative algorithm to optimize the instruments

Step 1: Initialize the parameter vector $\rho = \left[\rho_1, \ldots, \rho_K\right]^T$

Step 2: for iter = 1 : convergence (Iterative iv estimations)

for each subsystem k

i. Generate the instrumental variables $\hat{y}_k$ from the auxiliary model with the estimated polynomials based on the estimated parameter vector ρ_k^{iter-1}

ii. Update the filter $F_{k,\gamma}^{iter}(p, \hat{\rho}_k)$ with the new estimate

$$F_{k,\gamma}^{iter}(p, \hat{\rho}_k) = \frac{p^\gamma}{\hat{A}_k(p, \hat{\rho}_k)}$$

iii. Then evaluate the prefiltered derivatives of $u_k(t)$, $x_k(t)$ and $\hat{y}_k(t)$

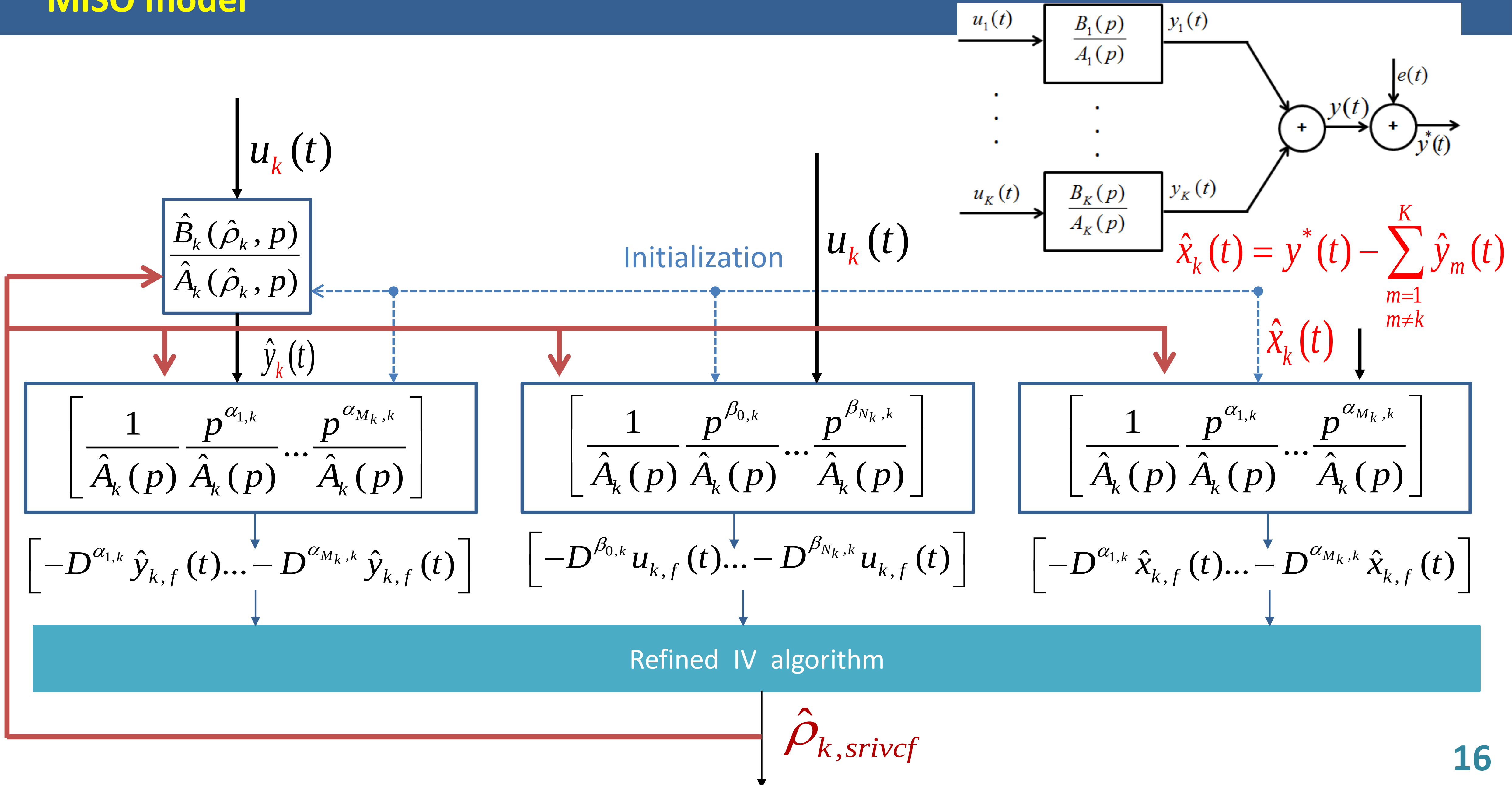
$$\begin{cases} D^{\beta_i} u_{k,f}(t) = F_{\beta_i}^{iter}(p) u_k(t) \\ D^{\alpha_j} y_{k,f}(t) = F_{\alpha_j}^{iter}(p) y_k(t) \\ D^{\alpha_j} x_{k,f}(t) = F_{\alpha_j}^{iter}(p) x_k(t) \end{cases}$$

iv. Based on these prefiltered data, compute the new estimates

$$\hat{\rho}_{k,srivcf}^{iter} = \left(\phi_{k,f}^{ivT} \phi_{k,f}\right)^{-1} \phi_{k,f}^{ivT} X_{k,f}$$

Optimal instrumental variable method for coefficient estimations

MISO model



Gauss-Newton algorithm for differentiation order estimation

- The estimation problem is formulated as a minimization problem of the ℓ_2-norm:

$$J(\hat{\theta}) = \frac{1}{2}\left\| \varepsilon(t,\hat{\theta}) \right\|_2$$

The output error

where the output error is defined as : $\varepsilon(t,\hat{\theta}) = y^*(t) - \hat{y}(t,\hat{\theta})$

and the estimated output is defined as : $\hat{y}(t) = \sum_{k=1}^{K} y_k(t)$

- A Gauss-Newton algorithm is used to compute, iteratively, the differentiation order vector μ^{iter} :

$$\mu^{iter} = \mu^{iter-1} - \lambda\left[\mathcal{H}^{-1}\frac{\partial J}{\partial \mu}\right]$$

The error sensitivity function

$\frac{\partial J}{\partial \mu}$ is the gradient defined as : $\frac{\partial J}{\partial \mu} = \frac{\partial \varepsilon(t,\hat{\theta})}{\partial \mu}^T \varepsilon(t,\hat{\theta})$

$\mathcal{H}$ is the approximated Hessian given by

$$\mathcal{H} = \frac{\partial \varepsilon(t,\hat{\theta})}{\partial \mu}^T \frac{\partial \varepsilon(t,\hat{\theta})}{\partial \mu}$$

Gauss-Newton algorithm for differentiation order estimation

The error sensitivity function is computed accordingly:

$k = 1, \ldots, K$

- Case1: if a global S-commensurate order is sought :

$$\frac{\partial \varepsilon}{\partial \mu} = \frac{\partial \varepsilon}{\partial \nu} = -\sum_{k=1}^{K} \frac{\partial \hat{y}_k}{\partial \nu} \Longrightarrow \frac{\partial \hat{y}_k}{\partial \nu} = \left[\sum_{j=0}^{M_k} j \hat{b}_{j,k} p^{j\nu} + \sum_{j=0}^{M_k} \sum_{i=1}^{N_k} (j-i) \hat{b}_{j,k} \hat{a}_{i,k} p^{(i+j)\nu} \right] \times \frac{\ln(p)}{\left(1 + \sum_{i=1}^{N_k} \hat{a}_{i,k} p^{i\nu}\right)^2} u_k(t)$$

???

- Case2: if local S-commensurate order is sought :

$$\frac{\partial \varepsilon}{\partial \mu} = \left[\frac{\partial \varepsilon}{\partial \nu_1}, \ldots, \frac{\partial \varepsilon}{\partial \nu_K} \right] = \left[\frac{-\partial \hat{y}_1}{\partial \nu_1}, \ldots, \frac{-\partial \hat{y}_K}{\partial \nu_K} \right]$$

- Case3: if the MISO model is non commensurate :

$$\frac{\partial \varepsilon}{\partial \mu} = \left[\frac{\partial \varepsilon}{\partial \mu_1}, \ldots, \frac{\partial \varepsilon}{\partial \mu_K} \right]$$

$$\frac{\partial \varepsilon}{\partial \mu_k} = \left[\frac{-\partial \hat{y}_k}{\partial \beta_{0.k}}, \ldots, \frac{-\partial \hat{y}_k}{\partial \beta_{M_k.k}}, \frac{-\partial \hat{y}_k}{\partial \alpha_{1,k}}, \ldots, \frac{-\partial \hat{y}_k}{\partial \alpha_{N_k.k}} \right]$$

A numerical solution was computed

Two-stage algorithm for coefficient and differentiation order estimation

Implement an iterative algorithm to optimize the parameter vector

Step 1: Initialize the parameter vector $\hat{\theta}^0 = \left[\hat{\rho}^0, \hat{\mu}^0\right]$

Step 2: Iterative all parameter estimation

do

i. Compute the coefficient vector $\hat{\rho}^{iter}$ with MISO-srivcf

ii. Differentiation order estimation

Initialize λ (usually to 1)

do

a. Evaluate the cost function

b. Refine the order estimate

$$\mu^{iter} = \mu^{iter-1} - \lambda \left[\mathcal{H}^{-1} \frac{\partial J}{\partial \mu} \right]_{\mu = \hat{\mu}^{iter-1}}$$

c. Evaluate the cost function

d. Set $\lambda = \lambda / 2$

while $J\left(\left[\hat{\rho}^{iter}, \hat{\mu}^{iter}\right]\right) > J\left(\left[\hat{\rho}^{iter-1}, \hat{\mu}^{iter-1}\right]\right)$

iii. Form the new estimate parameter vector

$$\hat{\theta}^{iter} = \left[\hat{\rho}^{iter}, \hat{\mu}^{iter}\right]$$

while $\sum_{l=1}^{\dim \hat{\rho}_k} \left| \frac{\hat{\theta}_l^{iter} - \hat{\theta}_l^{iter-1}}{\hat{\theta}_l^{iter-1}} \right| > \varepsilon$

Outline

1 Problem formulation

2 Two methods for MISO fractional system identification

- Two-stage algorithm
- Output error method

3 Simulation examples

4 Conclusions and prospects

Output Error method for coefficient and differentiation order estimations

- The estimation problem is formulated as a minimization problem of the ℓ_2-norm:

$$J(\hat{\theta}) = \frac{1}{2}\left\| \varepsilon(t,\hat{\theta}) \right\|_2$$

The output error

where the output error is defined as :

$$\varepsilon(t,\hat{\theta}) = y^*(t) - \hat{y}(t,\hat{\theta})$$

and the estimated output is defined as :

$$\hat{y}(t) = \sum_{k=1}^{K} y_k(t)$$

Output Error method for coefficient and differentiation order estimations

Implement an iterative algorithm to optimize the parameter vector

Step 1: Initialize the parameter vector $\hat{\theta}^0 = \left[\hat{\rho}^0, \hat{\mu}^0\right]$

Step 2: for iter = 1 : convergence

(Iterative Levenberg-Marquardt for all parameter estimation)

Compute the parameter vector $\hat{\theta}^{iter}$ as

$$\hat{\theta}^{iter+1} = \hat{\theta}^{iter} - \left\{ \left[\mathcal{H} + \xi I \right]^{-1} \frac{\partial J}{\partial \hat{\theta}} \right\} \Bigg|_{\hat{\theta}^{iter}}$$

with

$$\begin{cases} \dfrac{\partial J}{\partial \hat{\theta}} = \displaystyle\sum_{h=1}^{H} \frac{\partial \varepsilon(t_h)}{\partial \hat{\theta}}^T \varepsilon(t_h) & \text{: the gradient} \\ \mathcal{H} \approx \displaystyle\sum_{h=1}^{H} \frac{\partial \varepsilon(t_h)}{\partial \hat{\theta}}^T \frac{\partial \varepsilon(t_h)}{\partial \hat{\theta}} & \text{: pseudo\_Hessien} \\ \xi & \text{: Marquardt parameter} \end{cases}$$

Output Error method for coefficient and differentiation order estimations

The error sensitivity function is

$$\frac{\partial \varepsilon}{\partial \theta} = \frac{\partial \varepsilon}{\partial \left[\rho^T \ \mu^T \right]^T}$$

The coefficient error sensitivity function is

$$\frac{\partial \varepsilon}{\partial \rho} = \left[\frac{\partial \varepsilon}{\partial \rho_1}, \dots, \frac{\partial \varepsilon}{\partial \rho_K} \right]$$

where

$$\frac{\partial \varepsilon}{\partial \rho_k} = -\frac{\partial \hat{y}_k}{\partial \rho_k} = -\left[\frac{\partial \hat{y}_k}{\partial b_{0,k}}, \dots, \frac{\partial \hat{y}_k}{\partial b_{M_k,k}}, \frac{\partial \hat{y}_k}{\partial a_{1,k}}, \dots, \frac{\partial \hat{y}_k}{\partial a_{N_k,k}} \right]$$

with

$$\frac{\partial \hat{y}_k}{\partial b_{j.k}} = \frac{p^{\hat{\beta}_{j.k}}}{1 + \sum_{i=1}^{N_k} \hat{a}_{i,k} p^{\hat{\alpha}_{i,k}}} u_k(t), \quad j = 0, \dots, M_k$$

$$\frac{\partial \hat{y}_k}{\partial a_{i,k}} = -\frac{\sum_{j=0}^{M_k} \hat{b}_{j,k} p^{\hat{\beta}_{j.k} + \hat{\alpha}_{i,k}}}{\left(1 + \sum_{i=1}^{N_k} \hat{a}_{i,k} p^{\hat{\alpha}_{i,k}} \right)^2} u_k(t), \quad i = 1, \dots, N_k$$

The **differentiation order** error **sensitivity** function is **computed numerically**

Outline

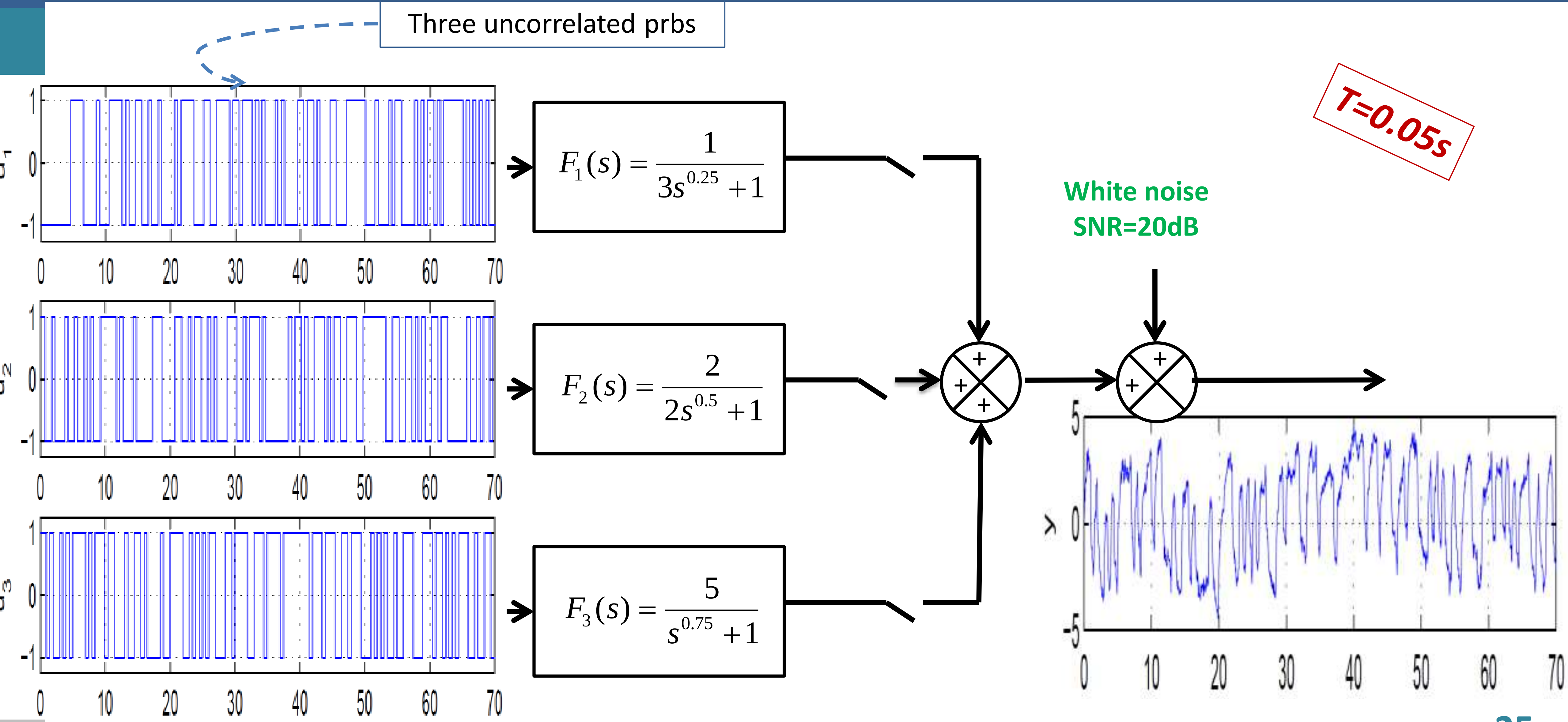
1 Problem formulation

2 Two methods for MISO fractional system identification

3 Simulation examples

4 Conclusions and prospects

Simulation example 1 – Coefficient estimation with known differentiation orders



Simulation example 1 – Coefficient estimation with known differentiation order

Hypotheses:

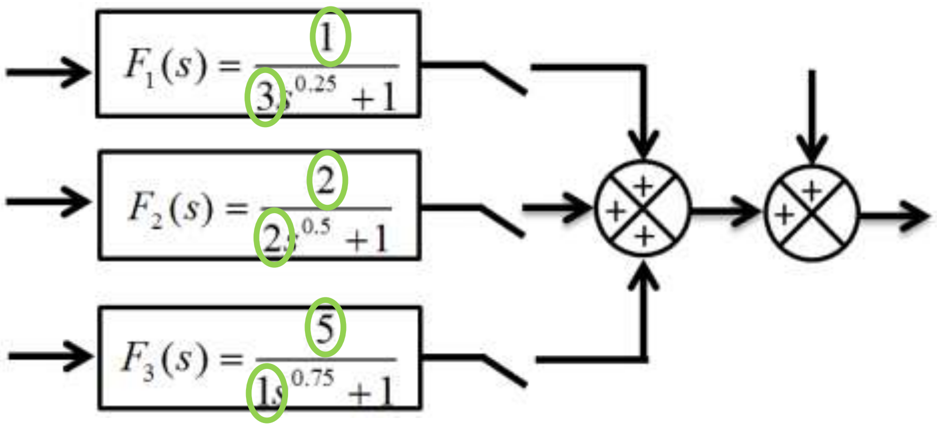
- ✓ Model structures are known.
- ✓ The output signal is corrupted by white noise.
- ✓ The differentiation orders are known

Objectives:

- ✓ Comparison between MISO-srivcf method and MISO-oe method:
  - ✓ Estimate the unknown coefficients.
  - ✓ Performance analysis with a Monte Carlo for 75 runs.
  - ✓ Study the influence of the commensurate order.

Simulation example 1 – Coefficient estimation with known differentiation order

	True	MISO-oe		MISO-srivcf	
	ρ	$\bar{\rho}$	$\hat{\sigma_\rho}$	$\bar{\rho}$	$\hat{\sigma_\rho}$
$b_{0,1}$	1	1.0225	0.1006	1.0096	0.0919
$a_{1,1}$	3	3.0802	0.4373	3.0161	0.3623
$b_{0,2}$	2	2.0082	0.0423	1.9996	0.0349
$a_{1,2}$	2	2.0107	0.0598	1.9983	0.0481
$b_{0,3}$	5	4.9998	0.0186	5.0017	0.0122
$a_{1,3}$	1	1.0005	0.0057	1.0009	0.0046

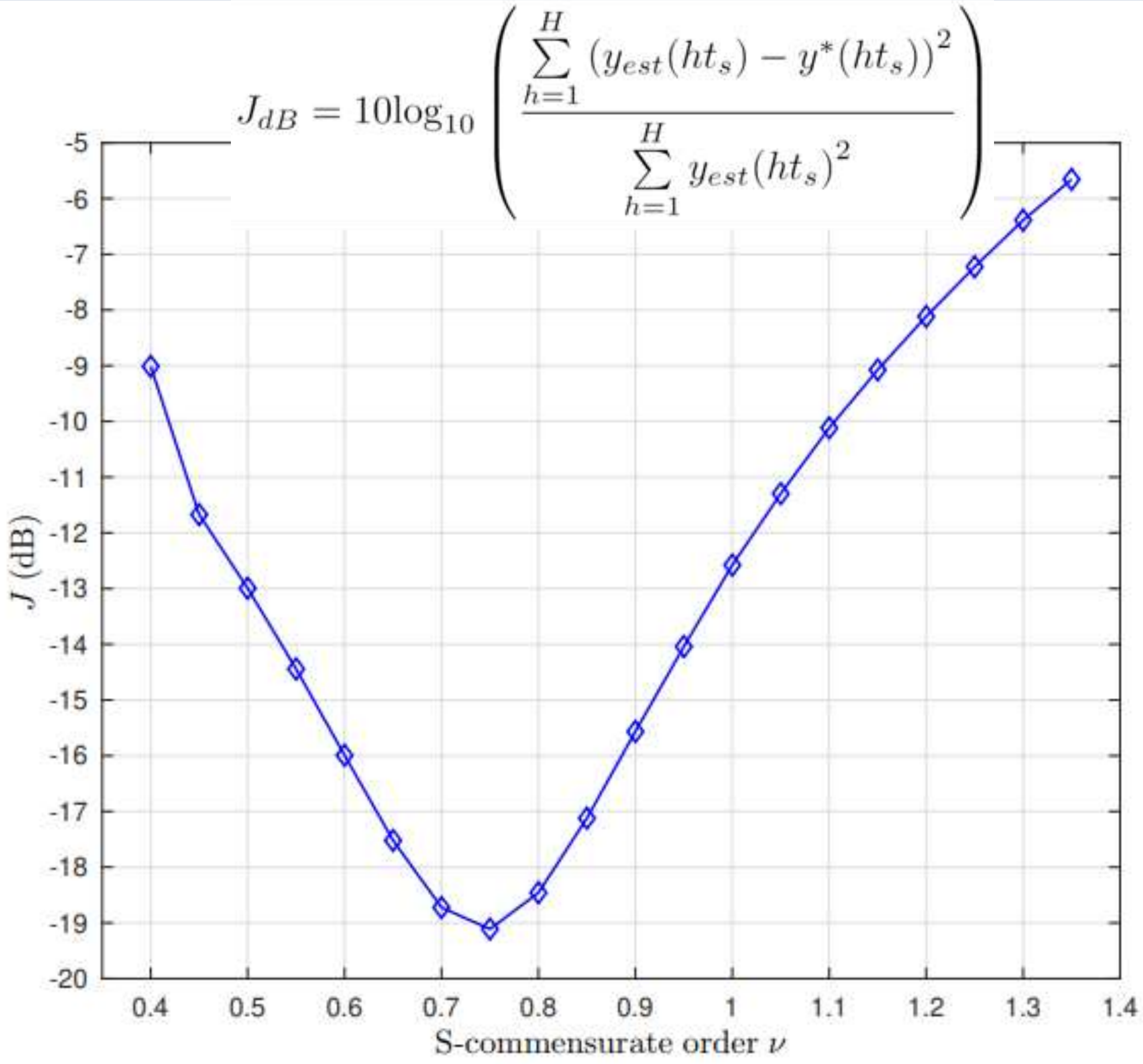


The table illustrates the numerical results of Monte Carlo simulation and the performance of MISO-srivcf and MISO-oe methods.

Simulation example 1 – Coefficient estimation with known differentiation order

❖ Unknown differentiation order

- The cost function is computed for different values of the commensurate order ν, in the stability range (0.4, 1.35), and plotted versus the commensurate order.
- The minimum of the cost function is found at ν=0.75. The minimum value of the criterion equals -19dB. Note that as a NSR of – 20dB is applied the modeling error is around 1dB,
- The smoothness of the cost function allows to implement gradient-based optimization algorithms.



Simulation example 2 – Coefficient and differentiation order estimation

Hypotheses:

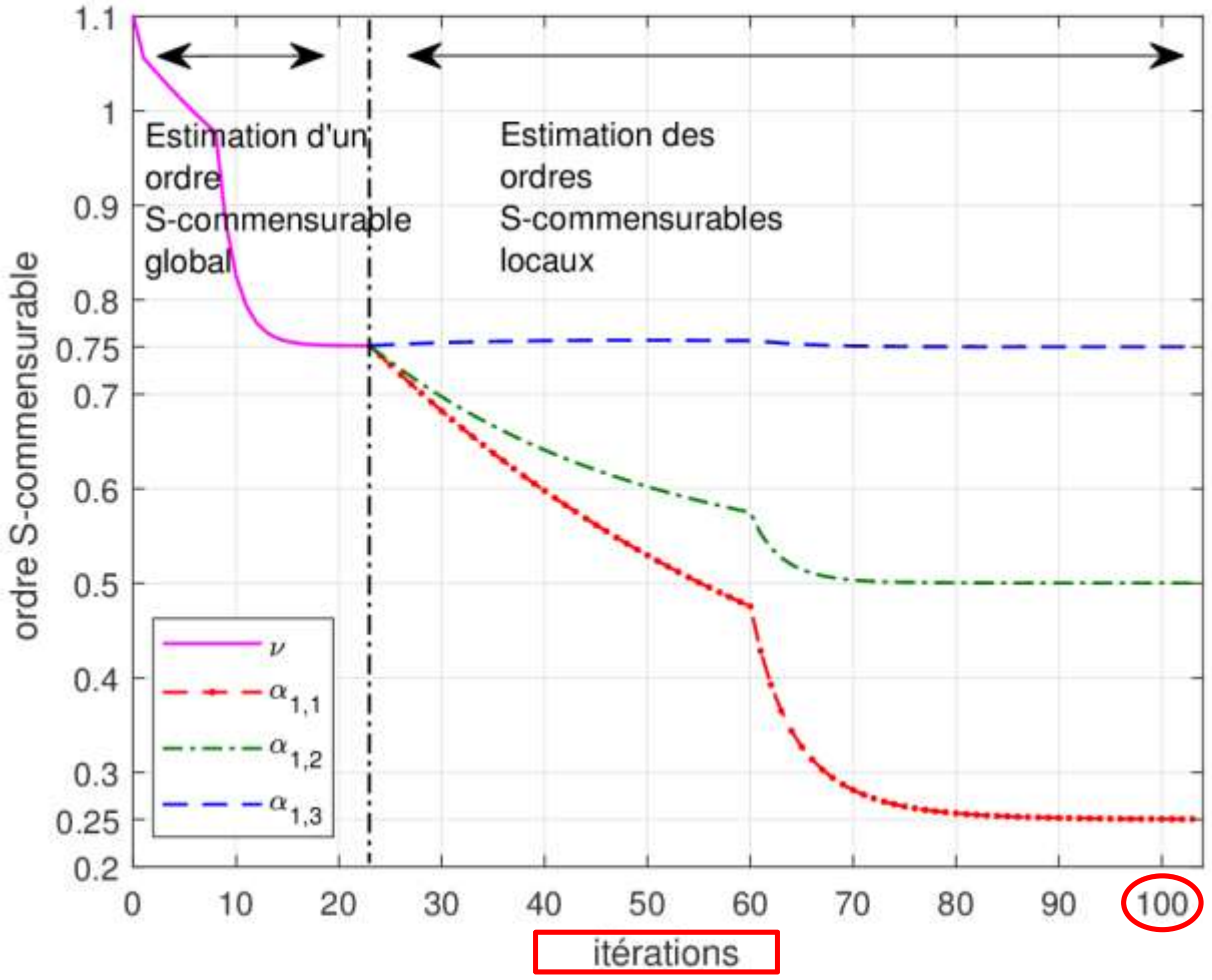
- ✓ Model structures are known.
- ✓ The output signal is corrupted by white noise.
- ✓ The S-commensurate order are unknown

Objectives:

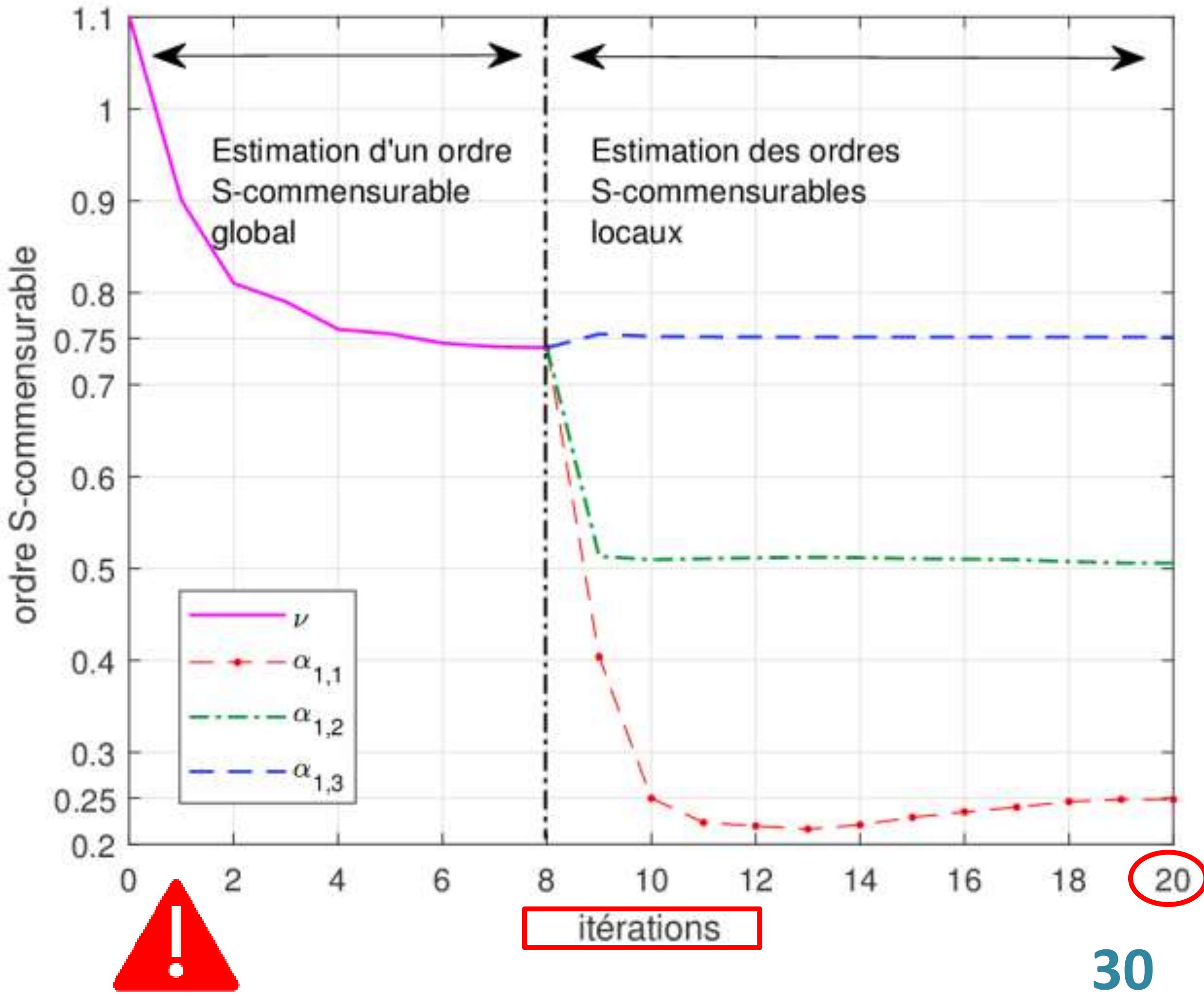
- ✓ Comparison between the MISO-oosrivcf method and the MISO-oe method:
  - ✓Estimate the unknown coefficients and the S-commensurate order.
  - ✓Performance analysis with a Monte Carlo for 75 runs. .

Simulation example 2 – Coefficient and differentiation order estimation

S-commensurate order estimation versus number of iterations
"MISO-oosrivcf method"



S-commensurate order estimation versus number of iterations
"MISO-oe method"



Simulation example 2 – Coefficient and differentiation order estimation

Hessian matrix

$$\mu^{iter} = \mu^{iter-1} - \lambda\left[\mathcal{H}^{-1}\frac{\partial J}{\partial \mu}\right]$$

$$\dim(\mathcal{H}) = 3$$

$$\hat{\theta}^{iter+1} = \hat{\theta}^{iter} - \left\{\left[\mathcal{H} + \xi I\right]^{-1}\frac{\partial J}{\partial \hat{\theta}}\right\}\Bigg|_{\hat{\theta}^{iter}}$$

$$\dim(\mathcal{H}) = 9$$

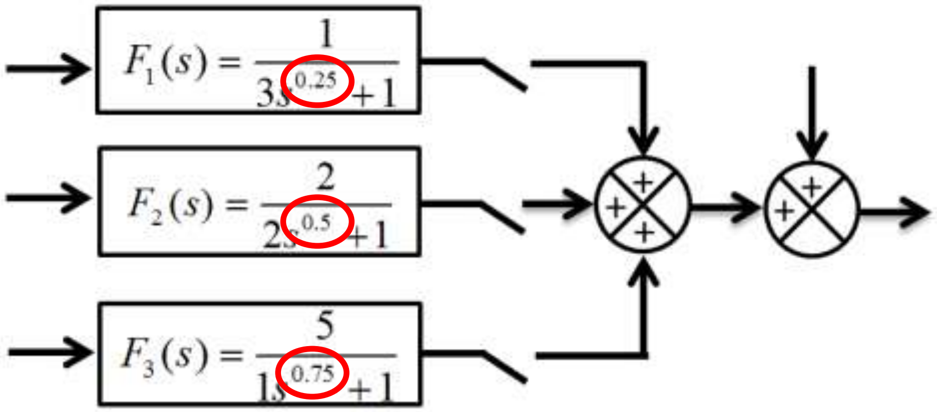
In MISO-oosrivcf algorithm, the hessian matrix contains only differentiation ordres sensitivity functions

lack of information make the MISO-oosrivcf algorithm slower to converge

In MISO-oe algorithm, the hessian matrix contains coefficients sensitivity functions and differentiation ordres sensitivity functions plus the cross sensitivity functions between the coefficients/differentiation orders.

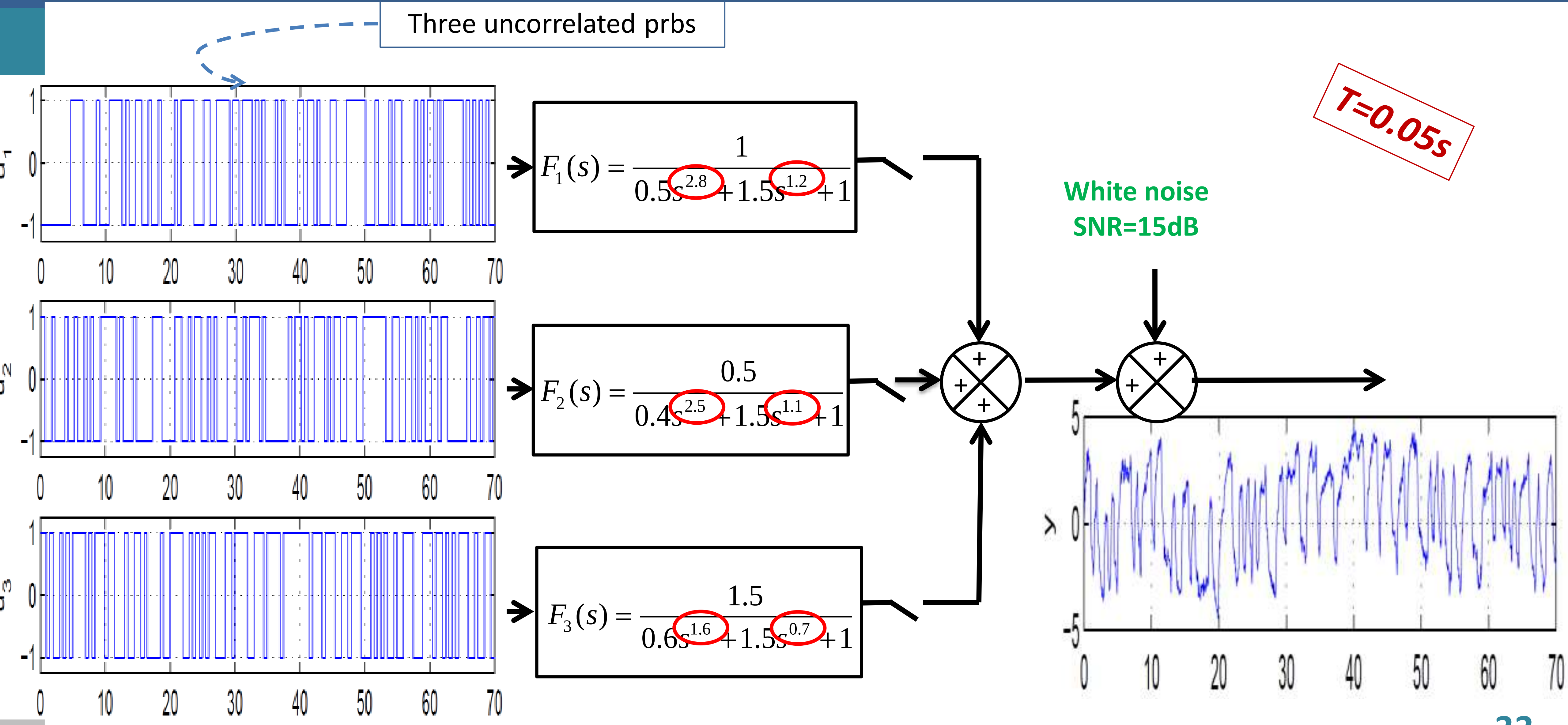
Simulation example 2 – Coefficient and differentiation order estimation

	True	MISO-oe		MISO-oosrivcf	
	θ	$\bar{\theta}$	$\hat{\sigma}_\theta$	$\bar{\theta}$	$\hat{\sigma}_\theta$
$b_{0,1}$	1	1.0079	0.1010	1.0047	0.0919
$a_{1,1}$	3	3.0366	0.4073	3.0125	0.3623
$\alpha_{1,1}$	0.25	0.2491	0.0135	0.2498	0.0105
$b_{0,2}$	2	2.0082	0.0523	1.9989	0.0439
$a_{1,2}$	2	2.0107	0.0707	1.9988	0.0625
$\alpha_{1,2}$	0.5	0.4987	0.0068	0.5002	0.0056
$b_{0,3}$	5	4.9998	0.0116	5.0117	0.0098
$a_{1,3}$	1	1.0005	0.0050	1.0009	0.0034
$\alpha_{1,3}$	0.75	0.7504	0.0025	0.7499	0.0012



The table illustrates the numerical results of Monte Carlo simulation and the performance of MISO-oosrivcf and MISO-oe methods.

Simulation example 3 – Coefficient and differentiation order estimation



Simulation example 3 – Coefficient and differentiation order estimation

Hypotheses:

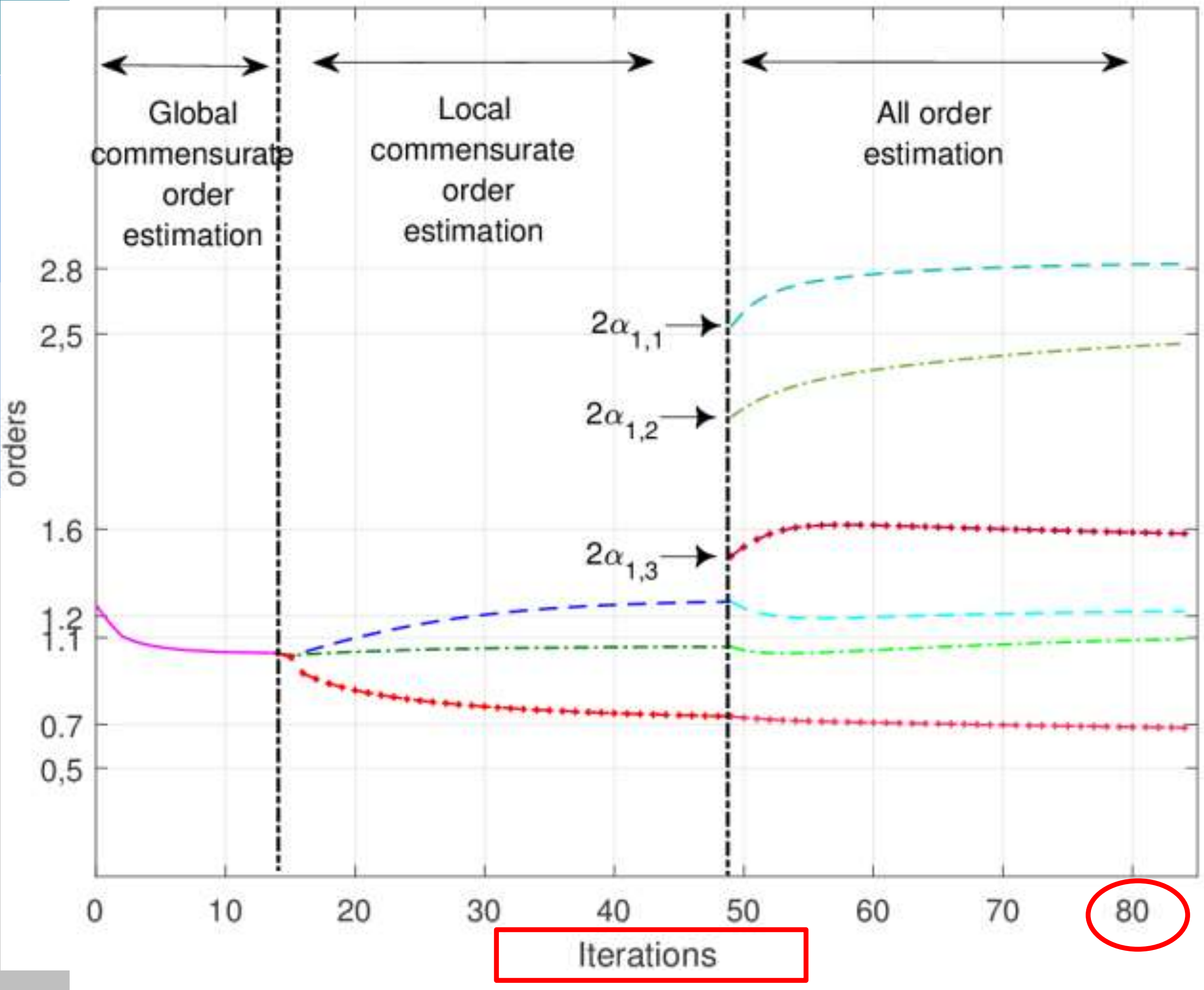
- ✓ Model structures are known.
- ✓ The output signal is corrupted by white noise.
- ✓ Differentiation orders are unknown

Objectives:

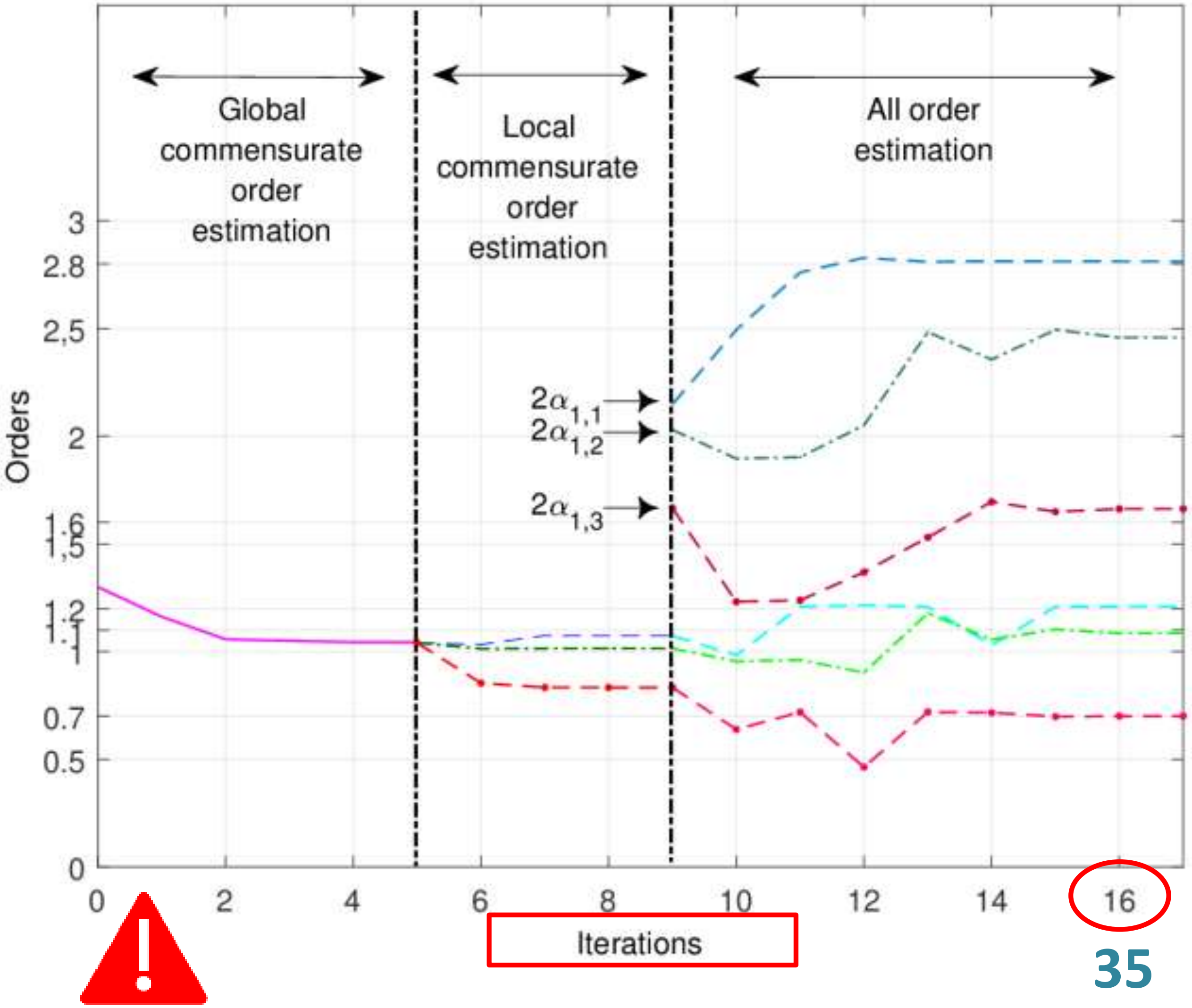
- ✓ Comparison between the MISO-oosrivcf method and the MISO-oe method:
  - ✓Estimate the unknown coefficients and the differentiation order.
  - ✓Performance analysis with a Monte Carlo for 50 runs. .

Simulation example 3 – Coefficient and differentiation order estimation

Differentiation order estimation versus number of iterations
“MISO-oosrivcf method”

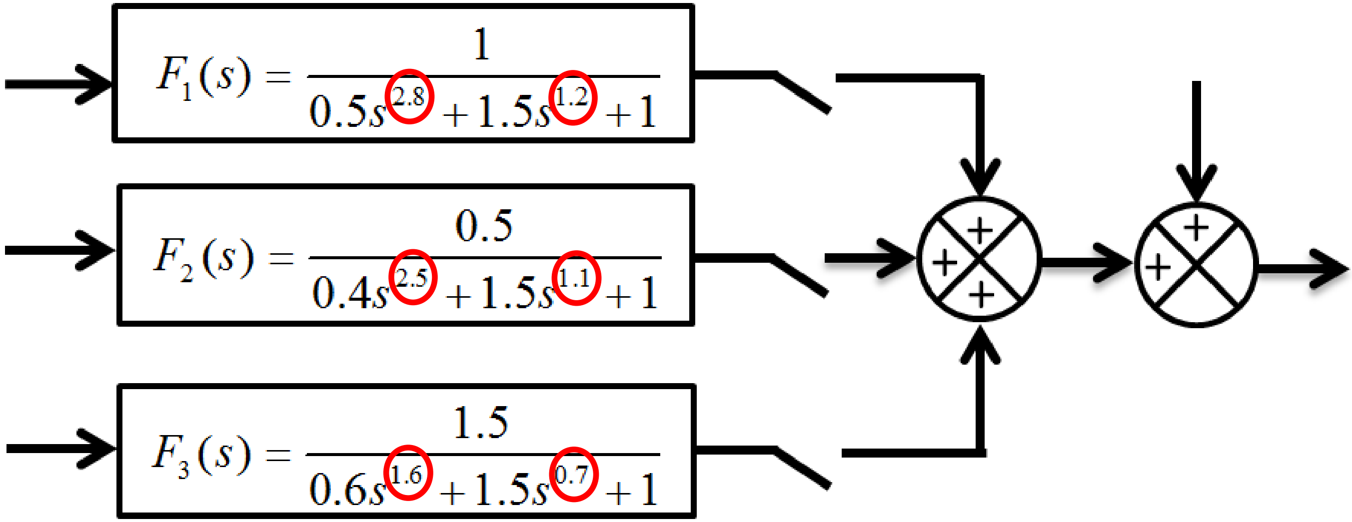


Differentiation order estimation versus number of iterations
“MISO-oe method”



Simulation example 3 – Coefficient and differentiation order estimation

	True	MISO-OE		MISO-oosrivcf	
	θ	$\bar{\theta}$	$\hat{\sigma}_\theta$	$\bar{\theta}$	$\hat{\sigma}_\theta$
$b_{0,1}$	1	1.0018	0.0110	1.0001	0.0098
$a_{1,1}$	0.5	0.5028	0.0192	0.5022	0.0106
$a_{2,1}$	1.5	1.5021	0.0112	1.5023	0.0152
$\mu_{1,1}$	2.8	2.7988	0.0145	2.7992	0.0161
$\mu_{2,1}$	1.2	1.1985	0.0164	1.1981	0.0108
$b_{0,2}$	0.5	0.4995	0.0158	0.5014	0.0145
$a_{1,2}$	0.4	0.4031	0.0571	0.4075	0.0452
$a_{2,2}$	1.5	1.4997	0.0466	1.5047	0.0479
$\mu_{1,2}$	2.5	2.5068	0.0759	2.4958	0.0685
$\mu_{2,2}$	1.1	1.1009	0.0415	1.0968	0.0356
$b_{0,3}$	1.5	1.5021	0.0336	1.4998	0.0298
$a_{1,3}$	0.6	0.6121	0.1253	0.6099	0.1234
$a_{2,3}$	1.5	1.4891	0.1088	1.4892	0.1023
$\mu_{1,3}$	1.6	1.5958	0.1109	1.6023	0.1065
$\mu_{2,3}$	0.7	0.6959	0.0538	0.6990	0.0501



The table illustrates the numerical results of Monte Carlo simulation and the performance of MISO-oosrivcf and MISO-oe methods.

Outline

Conclusion

Conclusions:

- Optimal instrumental method for MISO fractional systems.
  - ✓ Coefficient estimation
  - ✓ Cofficient estimation combined withdifferentiation ordre estimation

- Output error method for MISO fractional systems.

- Initialize procuss (three stage-initialization of differentiation order to reduce the number of parameter

- Comparison between both algorithm (the oe-MISO method converge faster)

Prospects:

- Application to climate modeling