FROM TIME SERIES to BALANCED REPRESENTATION

Part II: Algorithms

Ivan Markovsky (University of Leuven)
Jan C. Willems (University of Leuven)

Paolo Rapisarda (University of Maastricht)
Bart L.M. De Moor (University of Leuven)

ERNSI meeting, Noorwijkerhout, NL

October 7, 2003

Outline

- A new algorithm for balanced subspace identification
- Comparison with Van Overschee–De Moor algorithm
- Comparison with Moonen–Ramos algorithm
- Simulations
- Conclusions and discussion

A new algorithm for balanced subspace identification

The problem and an outline of the basic algorithm

problem: **given:** $u, y : [1, T] \to \mathbb{R}^{\mathtt{m}} \times \mathbb{R}^{\mathtt{p}}$

satisfying the conditions of the fundamental lemma

determine: an associated balanced state model

basic algorithm (with finite matrices):

1. find **sequential zero input responses** Y_0, row dim$(Y_0) = \Delta\mathtt{p}$
2. find the **impulse response** $H : [0, 2\Delta - 1] \to \mathbb{R}^{\mathtt{p}\times\mathtt{m}}$
3. compute the **SVD** of the Hankel matrix of Markov parameters $\mathfrak{H}$

$$\mathfrak{H} = U\Sigma V^\top, \qquad \text{where} \qquad \mathfrak{H} \in \mathbb{R}^{\Delta\mathtt{p}\times\Delta\mathtt{m}}$$

4. find a **balanced state sequence** $X := \sqrt{\Sigma^{-1}}U^\top Y_0$
5. find a **balanced realization** A, B, C, D (by LS)

Notation

part I distinguishes $\begin{array}{ll} w = (u,y) & \text{— general trajectory} \\ \tilde w = (\tilde u, \tilde y) & \text{— particular one} \end{array}$

later on: $w = (u,y)$ is a particular (measured) trajectory

$\mathcal{H}_L(\cdot)$ is a block-Hankel matrix with L block-rows

e.g., with $u : [1,T] \to \mathbb{R}^{\mathtt{m}}$

$$\mathcal{H}_t(u) = \begin{bmatrix} u(1) & u(2) & u(3) & \cdots & u(T-L+1) \\ u(2) & u(3) & u(4) & \cdots & u(T-L+2) \\ u(3) & u(4) & u(5) & \cdots & u(T-L+3) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ u(L) & u(L+1) & u(L+2) & \cdots & u(T) \end{bmatrix}$$

Notation

$\mathcal{T}_L(\cdot)$ is a lower triangular $L \times L$ block-Toeplitz matrix

e.g., with $H(0) := D,\ H(t) := CA^{t-1}B,\ t = 1, \ldots, L-1$

$$\mathcal{T}_L(H) = \begin{bmatrix} H(0) & 0 & 0 & \cdots & 0 \\ H(1) & H(0) & 0 & \ddots & \vdots \\ \vdots & H(1) & H(0) & \ddots & 0 \\ H(L-2) & \vdots & \ddots & \ddots & 0 \\ H(L-1) & H(L-2) & \cdots & H(1) & H(0) \end{bmatrix}$$

Impulse response from data

let $\quad H := \begin{bmatrix} H^\top(0) & H^\top(1) & \cdots & H^\top(2\Delta - 1) \end{bmatrix}^\top$

given $w = (u,y)$, $\quad$ find $\quad H$

col span$\big(\mathcal{H}_{2\Delta}(w)\big) = \mathfrak{B}|_{[1,2\Delta]} \implies \exists\, G$ s.t. $H = \mathcal{H}_{2\Delta}(y)G$

how to do that?

let $\mathtt{n}_{\max}$ be an estimate of an upper bound on the system order $\mathtt{n}$

$$\mathcal{H}_{\mathtt{n}_{\max}+2\Delta}(u) := \begin{bmatrix} U_{\mathsf{p}} \\ U_{\mathsf{f}} \end{bmatrix} \begin{matrix} \}\mathtt{n}_{\max}\mathtt{m} \\ \}2\Delta\mathtt{m} \end{matrix}, \quad \mathcal{H}_{\mathtt{n}_{\max}+2\Delta}(y) := \begin{bmatrix} Y_{\mathsf{p}} \\ Y_{\mathsf{f}} \end{bmatrix} \begin{matrix} \}\mathtt{n}_{\max}\mathtt{p} \\ \}2\Delta\mathtt{p} \end{matrix}$$

Impulse response from data (cont.)

with G a solution of the system

$$\begin{bmatrix} U_{\mathsf{p}} \\ U_{\mathsf{f}} \\ Y_{\mathsf{p}} \end{bmatrix} G = \begin{bmatrix} 0_{\mathtt{n}_{\max}\times\mathtt{m}} \\ \begin{bmatrix} I_{\mathtt{m}} \\ 0_{(2\Delta-\mathtt{m})\times\mathtt{m}} \end{bmatrix} \\ 0_{\mathtt{n}_{\max}\times\mathtt{m}} \end{bmatrix} \begin{matrix} \longrightarrow \text{zero initial conditions} \\ \longrightarrow \text{impulse inputs} \\ \longrightarrow \text{zero initial conditions} \end{matrix}$$

$$\boxed{Y_{\mathsf{f}}G = H}$$

note: a solution G exists whenever u is persistently exciting of order at least $2(\Delta + \mathtt{n}_{\max})$

More samples of the impulse response

H computed above is with length Δ at most $\frac{1}{2\mathtt{m}}T - \mathtt{n}_{\max}$

moreover for efficiency and accuracy we want to keep Δ small

it is possible, however, to find an arbitrary long H

we will compute iteratively blocks of $L < \frac{1}{2\mathtt{m}}T - \mathtt{n}_{\max}$ consecutive samples of the impulse response

there are conflicting criteria in the choice of L, we want:

small L for efficiency and statistical accuracy (under noise) but large L for numerical stability

More samples of the impulse response (cont.)

let $F_{\mathsf{u}}^{(1)} := \begin{bmatrix} 0_{\mathtt{n}_{\max}\times\mathtt{m}} \\ \begin{bmatrix} I_{\mathtt{m}} \\ 0_{(L\mathtt{m}-\mathtt{m})\times\mathtt{m}} \end{bmatrix} \end{bmatrix}$ and $F_{\mathsf{y}}^{(1)} := \begin{bmatrix} 0_{\mathtt{n}_{\max}\times\mathtt{m}} \\ * \end{bmatrix}$

for $k = 1, 2, \ldots$ solve the system

$$\begin{bmatrix} U_{\mathsf{p}} \\ U_{\mathsf{f}} \\ Y_{\mathsf{p}} \end{bmatrix} G^{(k)} = \begin{bmatrix} F_{\mathsf{u}}^{(k)} \\ F_{\mathsf{y,p}}^{(k)} \end{bmatrix} \quad \text{where} \quad F_{\mathsf{y}}^{(k)} =: \begin{bmatrix} F_{\mathsf{y,p}}^{(k)} \\ F_{\mathsf{y,f}}^{(k)} \end{bmatrix}$$

define $H^{(k)} := Y_{\mathsf{f}} G^{(k)}$, $F_{\mathsf{y,f}}^{(k)} := H^{(k)}$, and shift $F_{\mathsf{u}}, F_{\mathsf{y}}$

$$F_{\mathsf{u}}^{(k+1)} := \begin{bmatrix} \sigma^L F_{\mathsf{u}}^{(k)} \\ 0_{L\mathtt{m}\times\mathtt{m}} \end{bmatrix}, \quad F_{\mathsf{y}}^{(k+1)} := \begin{bmatrix} \sigma^L F_{\mathsf{y}}^{(k)} \\ * \end{bmatrix}$$

More samples of the impulse response (cont.)

σM is the matrix obtained from M by deleting its 1st block-row

the result $H := \begin{bmatrix} H^{(1)} \\ H^{(2)} \\ \cdots \end{bmatrix}$ of the algorithm is the impulse response

monitor $||H^{(k)}||$ and stop when it is small enough

note: gives an automatic way to determine the "depth" constant Δ

this is our method for computing the impulse response

Sequential zero input responses

let $y_0 : [0, 1, \ldots, \Delta] \longrightarrow \mathbb{R}^{\mathtt{p}}$ be a zero input response (due to an initial condition $x(1)$)

given $w = (u, y)$, find a zero input response y_0

with a computed impulse response H of length Δ

$$y_0 = y(1{:}\Delta) - \mathcal{T}_\Delta(H) u(1{:}\Delta)$$

in particular

$$\boxed{Y_0 = \mathcal{H}_\Delta(y) - \mathcal{T}_\Delta(H)\mathcal{H}_\Delta(u)}$$

is a matrix of sequential zero input responses

Sequential zero input responses (cont.)

another approach: with g a solution of the system

$$\begin{bmatrix} U_{\mathsf{p}} \\ U_{\mathsf{f}} \\ Y_{\mathsf{p}} \end{bmatrix} g = \begin{bmatrix} * \\ 0 \\ * \end{bmatrix} \begin{array}{l} \longrightarrow \text{set initial conditions} \\ \longrightarrow \text{zero input} \\ \longrightarrow \text{set initial conditions} \end{array}$$

$$Y_{\mathsf{f}}\, g = y_0$$

in particular with G a solution of the system $\begin{bmatrix} U_{\mathsf{p}} \\ U_{\mathsf{f}} \\ Y_{\mathsf{p}} \end{bmatrix} G = \begin{bmatrix} U_{\mathsf{p}} \\ 0 \\ Y_{\mathsf{p}} \end{bmatrix}$

$\boxed{Y_{\mathsf{f}} G = Y_0}$ is a matrix of sequential zero input responses

i.e., the oblique projection in the classical subspace algorithms

More samples of the free response

let $F_{\mathsf{u}}^{(1)} := \begin{bmatrix} U_{\mathsf{p}} \\ 0 \end{bmatrix}$ and $F_{\mathsf{y}}^{(1)} := \begin{bmatrix} Y_{\mathsf{p}} \\ * \end{bmatrix}$

for $k = 1, 2, \ldots$ solve the system

$$\begin{bmatrix} U_{\mathsf{p}} \\ U_{\mathsf{f}} \\ Y_{\mathsf{p}} \end{bmatrix} G^{(k)} = \begin{bmatrix} F_{\mathsf{u}}^{(k)} \\ F_{\mathsf{y,p}}^{(k)} \end{bmatrix} \quad \text{where} \quad F_{\mathsf{y}}^{(k)} =: \begin{bmatrix} F_{\mathsf{y,p}}^{(k)} \\ F_{\mathsf{y,f}}^{(k)} \end{bmatrix}$$

define $Y_0^{(k)} := Y_{\mathsf{f}} G^{(k)}$, $F_{\mathsf{y,f}}^{(k)} := Y_0^{(k)}$, and shift F_{u}, F_{y}

$$F_{\mathsf{u}}^{(k+1)} := \begin{bmatrix} \sigma^L F_{\mathsf{u}}^{(k)} \\ 0 \end{bmatrix}, \quad F_{\mathsf{y}}^{(k+1)} := \begin{bmatrix} \sigma^L F_{\mathsf{y}}^{(k)} \\ * \end{bmatrix}$$

Balanced state sequence

Hankel matrix of the Markov parameters: $\mathfrak{H} = \mathcal{H}_{\Delta}(\sigma H)$

$$\mathfrak{H} = U\Sigma V^{\top} = \underbrace{U\sqrt{\Sigma}}_{\Gamma_{\mathsf{bal}}}\underbrace{\sqrt{\Sigma}V^{\top}}_{\Delta_{\mathsf{bal}}}$$

$$\Gamma_{\mathsf{bal}} = \begin{bmatrix} C_{\mathsf{bal}} \\ C_{\mathsf{bal}} A_{\mathsf{bal}} \\ \vdots \\ C_{\mathsf{bal}} A_{\mathsf{bal}}^{\Delta-1} \end{bmatrix}, \quad \Delta_{\mathsf{bal}} = \begin{bmatrix} B_{\mathsf{bal}} & A_{\mathsf{bal}} B_{\mathsf{bal}} & \cdots & A_{\mathsf{bal}}^{\Delta-1} B_{\mathsf{bal}} \end{bmatrix}$$

matrix of sequential zero input responses: Y_0

$$Y_0 = \Gamma X = \Gamma_{\mathsf{bal}} X_{\mathsf{bal}} \implies \boxed{X_{\mathsf{bal}} = \sqrt{\Sigma^{-1}}U^{\top} Y_0}$$

Balanced model estimation by LS

$$X_{\mathsf{bal}} = \begin{bmatrix} x(\mathrm{n}_{\max}+1) & x(\mathrm{n}_{\max}+2) & \cdots & x(\mathrm{n}_{\max}+T+1-L) \end{bmatrix}$$

$$\begin{bmatrix} x(\mathrm{n}_{\max}+2) & x(\mathrm{n}_{\max}+3) & \cdots & x(\mathrm{n}_{\max}+T+1-L) \\ y(\mathrm{n}_{\max}+1) & y(\mathrm{n}_{\max}+2) & \cdots & y(\mathrm{n}_{\max}+T-L) \end{bmatrix} =$$

$$\begin{bmatrix} \hat{A} & \hat{B} \\ \hat{C} & \hat{D} \end{bmatrix} \begin{bmatrix} x(\mathrm{n}_{\max}+1) & x(\mathrm{n}_{\max}+2) & \cdots & x(\mathrm{n}_{\max}+T-L) \\ u(\mathrm{n}_{\max}+1) & u(\mathrm{n}_{\max}+2) & \cdots & u(\mathrm{n}_{\max}+T-L) \end{bmatrix}$$

A new algorithm

input: $u(1), \dots, u(T), \quad y(1), \dots, y(T)$
an upper bound $\mathrm{n_{max}}$ for the system order

1. zero input responses: $Y_0 = Y_\mathrm{f} G$, where $\begin{bmatrix} U_\mathrm{p} \\ U_\mathrm{f} \\ Y_\mathrm{p} \end{bmatrix} G = \begin{bmatrix} U_\mathrm{p} \\ 0 \\ Y_\mathrm{p} \end{bmatrix}$
2. impulse response: $H = Y_\mathrm{f} G$, where $\begin{bmatrix} U_\mathrm{p} \\ U_\mathrm{f} \\ Y_\mathrm{p} \end{bmatrix} G = \begin{bmatrix} 0 \\ \begin{bmatrix} I \\ 0 \end{bmatrix} \\ 0 \end{bmatrix}$
3. SVD: $\mathfrak{H} = \mathcal{H}_\Delta(\sigma H) = U \Sigma V^\top$
4. balanced state sequence: $X = \sqrt{\Sigma^{-1}} U^\top Y_0$
5. balanced model: solve the LS problem (LS)

output: $\hat{A}, \hat{B}, \hat{C}, \hat{D}$

Comparison with the algorithm Van Overschee–De Moor

Algorithm Van Overschee–De Moor

input: $u_0, \dots, u_T \quad y_0, \dots, y_T$ and $i, \quad i \geq \mathrm{n_{max}}$

$\begin{bmatrix} U_\mathrm{p} \\ U_\mathrm{f} \end{bmatrix} := \mathcal{H}_{2i}(u), \quad \begin{bmatrix} Y_\mathrm{p} \\ Y_\mathrm{f} \end{bmatrix} := \mathcal{H}_{2i}(y)$ row dim$(U_\mathrm{p}) = i\,\mathrm{m}$, row dim$(U_\mathrm{f}) = i\,\mathrm{m}$

1. oblique projection: $Y_0 := Y_\mathrm{f} / _{U_\mathrm{f}} \begin{bmatrix} U_\mathrm{p} \\ Y_\mathrm{p} \end{bmatrix}$
2. weight matrix: $W = U_\mathrm{p}^\top (U_\mathrm{p} U_\mathrm{p}^\top)^{-1} J$
3. SVD: $Y_0 W = U \Sigma V^\top$
4. balanced state sequence: $X_\mathrm{f} = \sqrt{\Sigma^{-1}} U^\top Y_0$
5. balanced model: solve the LS problem (LS)

output: $\hat{A}, \hat{B}, \hat{C}, \hat{D}$

Comments

- the oblique proj. $Y_\mathrm{f} / _{U_\mathrm{f}} \begin{bmatrix} U_\mathrm{p} \\ Y_\mathrm{p} \end{bmatrix}$ contains sequential zero input responses
- $Y_0 W$ contains impulse responses $+$ initial condition responses
- $Y_0 W$ is only approximately a Hankel matrix of Markov param.
- for large i the initial conditions responses die out and the impulse responses dominate
- due to the Hankel structure most elements are recomputed many times
- in approximate case the matrix $Y_0 W$ is not Hankel

Comparison

- both VO–DM and the new algorithm match the basic outline
- steps 4 (balanced state seq.) and 5 (LS) are the same
- different are the methods for computing the impulse response and the zero input response
- algorithm VO–DM computes the Hankel matrix itself
- the new algorithm computes the impulse response (and constructs the Hankel matrix from the response)

The oblique projection

the oblique projection $A/_B C$ is closely related to the solution of the system $\left[\begin{smallmatrix} C \\ B \end{smallmatrix}\right] G = \left[\begin{smallmatrix} C \\ 0 \end{smallmatrix}\right]$ that we use

$A/_B C$ — project A obliquely onto C along B

$$A/_B C := A \begin{bmatrix} C^\top & B^\top \end{bmatrix} \begin{bmatrix} CC^\top & CB^\top \\ BC^\top & BB^\top \end{bmatrix}^+ \begin{bmatrix} C \\ 0 \end{bmatrix} \qquad \text{(OBL)}$$

$Y_\mathrm{f}/_{U_\mathrm{f}} \left[\begin{smallmatrix} U_\mathrm{p} \\ Y_\mathrm{p} \end{smallmatrix}\right]$ is the standard way of computing $Y_0 = \Gamma X$

let G be the least-norm, least-squares solution of the system

$$\begin{bmatrix} C \\ B \end{bmatrix} G = \begin{bmatrix} C \\ 0 \end{bmatrix} \qquad \text{then} \qquad A/_B C = AG$$

Comparison with Moonen–Ramos algorithm

Algorithm Moonen–Ramos

$$\begin{bmatrix} U_\mathrm{p} \\ U_\mathrm{f} \end{bmatrix} := \mathcal{H}_{2i}(u) \qquad \begin{bmatrix} Y_\mathrm{p} \\ Y_\mathrm{f} \end{bmatrix} := \mathcal{H}_{2i}(y)$$

row dim$(U_\mathrm{p}) = i\,\mathtt{m}$ row dim$(Y_\mathrm{p}) = i\,\mathtt{p}$
row dim$(U_\mathrm{f}) = i\,\mathtt{m}$ row dim$(Y_\mathrm{f}) = i\,\mathtt{p}$

let the rows of $\begin{bmatrix} T_1 & T_2 & T_3 & T_4 \end{bmatrix}$ form a basis for the left kernel of $\left[\begin{smallmatrix} U_\mathrm{p} \\ Y_\mathrm{p} \\ U_\mathrm{f} \\ Y_\mathrm{f} \end{smallmatrix}\right]$

$$\begin{bmatrix} T_1 & T_2 & T_3 & T_4 \end{bmatrix} \begin{bmatrix} U_\mathrm{p} \\ Y_\mathrm{p} \\ U_\mathrm{f} \\ Y_\mathrm{f} \end{bmatrix} = 0$$

Algorithm Moonen–Ramos

input: $u_0, \ldots, u_T, \quad y_0, \ldots, y_T$ **and** $i, \quad i \geq \mathrm{n}_{\max}$

$$\begin{bmatrix} U_\mathrm{p} \\ U_\mathrm{f} \end{bmatrix} := \mathcal{H}_{2i}(u), \quad \begin{bmatrix} Y_\mathrm{p} \\ Y_\mathrm{f} \end{bmatrix} := \mathcal{H}_{2i}(y) \qquad \text{row dim}(U_\mathrm{p}) = i\,\mathrm{m}, \quad \text{row dim}(U_\mathrm{f}) = i\,\mathrm{m}$$

0. **annihilators:** $[T_1 \ T_2 \ T_3 \ T_4]$
1. **zero input responses:** $Y_0 = T_4^+ [T_1 \ T_2] \begin{bmatrix} U_\mathrm{p} \\ Y_\mathrm{p} \end{bmatrix}$
2. **Hankel matrix:** $\mathfrak{H} = T_4^+ (T_2 T_4^+ T_3 - T_1)$
3. **SVD:** $\mathfrak{H} = U \Sigma V^\top$
4. **balanced state sequence:** $X_\mathrm{f} = \sqrt{\Sigma^{-1}} U^\top Y_0$
5. **balanced model:** solve the LS problem (LS)

output: $\hat{A}, \hat{B}, \hat{C}, \hat{D}$

Comments

- the main computation is to find the annihilators $[T_1 \cdots T_4]$ efficient implementation should **exploit the Hankel structure**
- we have a **"dual" algorithm**, to the one presented, that recursively computes the left kernel of the data matrix
- $[T_1 \ T_2] \begin{bmatrix} U_\mathrm{p} \\ Y_\mathrm{p} \end{bmatrix}$ is a **state sequence** (shift-and-cut operator)
- $T_4^+ [T_1 \ T_2] \begin{bmatrix} U_\mathrm{p} \\ Y_\mathrm{p} \end{bmatrix}$ is a matrix of **zero input responses**
- $T_4^+ (T_2 T_4^+ T_3 - T_1)$ is the **Hankel matrix**

Comparison

- Moonen–Ramos algorithm also **fits into the basic outline**
- steps 4 (balanced state seq.) and 5 (LS) are the same
- the impulse and a free responses are computed via the annihilators $\begin{bmatrix} T_1 & T_2 & T_3 & T_4 \end{bmatrix}$
- again most elements are **recomputed** many times

  therefore under noise $T_4^+ (T_2 T_4^+ T_3 - T_1)$ is **not Hankel**

Simulations

Simulation setup

aim: to show correctness and advantages of the new algorithm

but the algorithms are not optimized in efficiency

example used in all experiments:

third order random stable SISO system

$T = 100$, $\tilde{u}$ is unity variance white noise

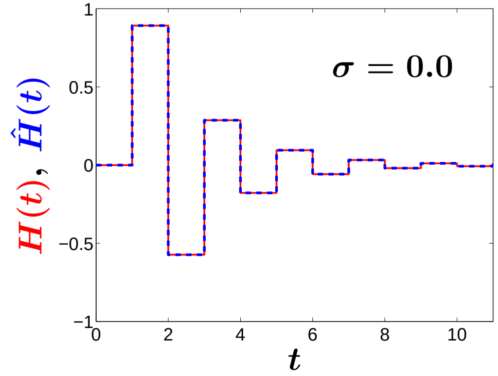
$\tilde{w}$ **is corrupted** by white noise with standard deviation σ

in all simulations: $\mathtt{n}_{\max} = \mathtt{n}$ and $L = \mathtt{n}$

Impulse response estimation

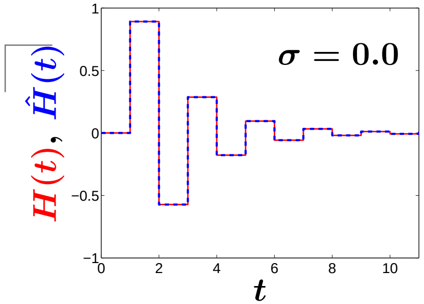
solid red — exact impulse response H

dashed blue — impulse response computed from data $\hat{H}$

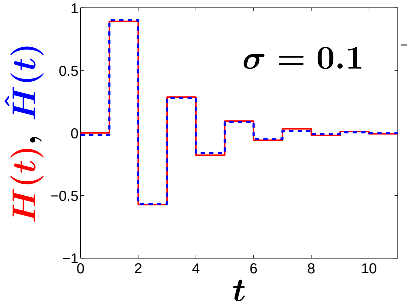


$||H - \hat{H}||_F = 10^{-15}$ $\Longrightarrow$ up to the numerical precision **exact match**

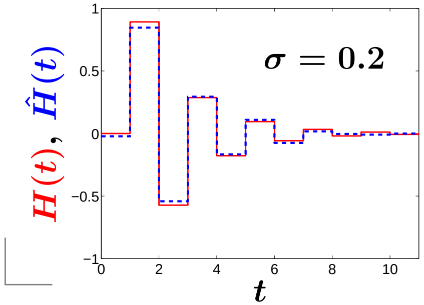
$||H - \hat{H}|| = 10^{-15}$



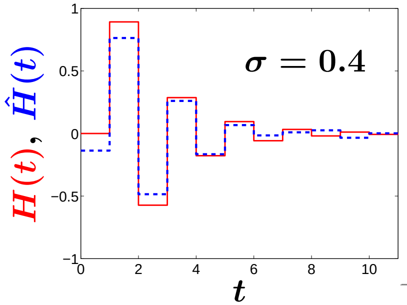
$||H - \hat{H}|| = 0.02$



$||H - \hat{H}|| = 0.05$



$||H - \hat{H}|| = 0.21$



Free response estimation

$Y_0 = \Gamma X$ — exact sequence of free responses ($\Delta = 10$)

$\hat{Y}_0$ — estimated sequence of free responses

error of estimation: $e = ||Y_0 - \hat{Y}_0||_F$

σ	0.0	0.1	0.2	0.4
new algorithm	10^{-14}	1.33	2.84	4.48
oblique proj.	10^{-11}	2.02	4.03	5.44

the oblique projection is computed by (OBL)

note: the new algorithm uses **more overdetermined system** of equations and **does not square the data**

Closeness to balancing

the algorithms return a finite time balanced model

we illustrate the effect of the depth parameter Δ on the balancing

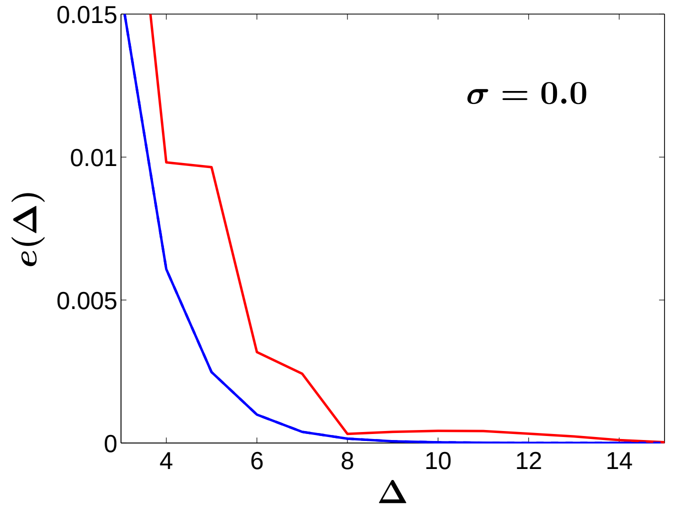
closeness to exact balancing

$\mathcal{C}/\mathcal{O}$ — contr./obsrv. Gramian of the exact balanced model
$\hat{\mathcal{C}}/\hat{\mathcal{O}}$ — contr./obsrv. Gramian of the identified model

$$e^2 := \frac{||\mathcal{C} - \hat{\mathcal{C}}||_F^2 + ||\mathcal{O} - \hat{\mathcal{O}}||_F^2}{||\mathcal{C}||_F^2 + ||\mathcal{O}||_F^2}$$

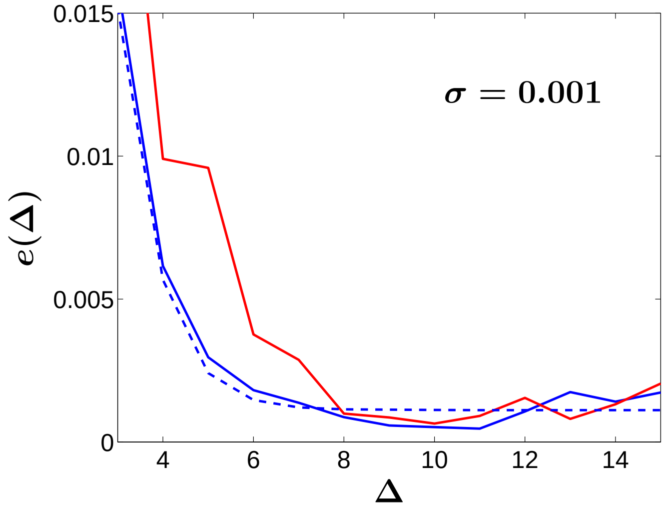
Closeness to balancing (cont.)

red — VO–DM blue — M–R new $\equiv$ M–R



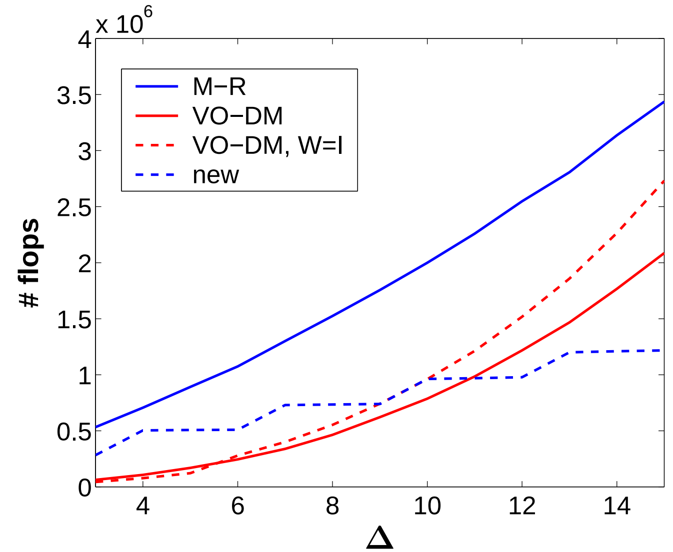
Closeness to balancing (cont.)

red — VO–DM blue — M–R dashed blue — new



Flops count

solid red — VO–DM
dashed red — VO–DM, $W = I$
solid blue — M–R
dashed blue — new



Conclusions and discussion

Conclusions

- impulse response and sequence of zero input responses are the **main tools for balanced model identification**
- classically they are computed via the **oblique projection**
- we gave **system theoretic interpretation** of the oblique proj.
- arbitrary **long responses** can be computed **from finite data set**
- computation of impulse response instead of Hankel matrix of Markov parameters can **improve efficiency and accuracy**
- next goal: **optimize efficiency and implement in C/FORTRAN**