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

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)

Impulse response from data

let $H(0) := D$, $H(t) := CA^{t-1}B$, and $H := \begin{bmatrix} H(0) \\ H(1) \\ \vdots \\ H(2\Delta-1) \end{bmatrix}$

given $\tilde{w} = (\tilde{u}, \tilde{y})$, find H

let $\mathcal{H}_\Delta(\tilde{w})$ be the block-Hankel matrix with Δ block-rows, composed of the elements $\tilde{w}(1), \tilde{w}(2), \ldots$

col span$\big(\mathcal{H}_{2\Delta}(\tilde{w})\big) = \mathfrak{B}|_{[0,2\Delta-1]} \implies \exists\, G$ s.t. $H = \mathcal{H}_{2\Delta}(\tilde{y})G$

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

define $\mathcal{H}_{\mathtt{n}_{\max}+2\Delta}(\tilde{u}) := \begin{bmatrix} U_\mathsf{p} \\ U_\mathsf{f} \end{bmatrix}$ row dim$(U_\mathsf{p}) = \mathtt{n}_{\max}\mathtt{m}$, row dim$(U_\mathsf{f}) = 2\Delta\mathtt{m}$

Impulse response from data (cont.)

similarly $\mathcal{H}_{n_{max}+2\Delta}(\tilde{y}) := \begin{bmatrix} Y_p \\ Y_f \end{bmatrix}$ $\quad$ row dim$(Y_p) = n_{max}p$, row dim$(Y_f) = 2\Delta p$

with G a solution of the system

$$\begin{bmatrix} U_p \\ U_f \\ Y_p \end{bmatrix} G = \begin{bmatrix} 0_{n_{max} \times m} \\ \begin{bmatrix} I_m \\ 0_{(2\Delta - m) \times m} \end{bmatrix} \\ 0_{n_{max} \times m} \end{bmatrix}$$

$\rightarrow$ zero initial conditions
$\rightarrow$ impulse inputs
$\rightarrow$ zero initial conditions

$$\boxed{H = Y_f G}$$

note: a solution G exists whenever $\tilde{u}$ is persistently exciting of order at least $2\Delta + n_{max}$

More samples of the impulse response

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

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 $L < \frac{1}{2\mathtt{m}}T - \mathrm{n}_{\mathrm{max}}$ 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 $$F_{\mathrm{u}}^{(1)} := \begin{bmatrix} 0_{\mathtt{n}_{\max}\times\mathtt{m}} \\ \begin{bmatrix} I_{\mathtt{m}} \\ 0_{(L-\mathtt{m})\times\mathtt{m}} \end{bmatrix} \end{bmatrix} \quad \text{and} \quad F_{\mathrm{y}}^{(1)} := \begin{bmatrix} 0_{\mathtt{n}_{\max}\times\mathtt{m}} \\ * \end{bmatrix}$$

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

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

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

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

More samples of the impulse response (cont.)

σM is the matrix obtained from M by deleting its first 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

Zero input response

let $\tilde{y}_0 : [0, 1, \ldots, \Delta] \to \mathbb{R}^p$ be a zero input response (due to an initial condition x_0)

given $\tilde{w} = (\tilde{u}, \tilde{y})$, find a zero input response $\tilde{y}_0$

let $\mathcal{T}_\Delta(H)$ be the lower triangular block-Toeplitz matrix with Δ block-rows and Δ block-columns, composed of $H(1), H(2), \ldots$

with a computed impulse response H of length Δ

$$\boxed{\tilde{y}_0 = \tilde{y}(1{:}\Delta) - \mathcal{T}_\Delta(H)\tilde{u}(1{:}\Delta)}$$

in particular $Y_0 = \mathcal{H}_\Delta(\tilde{y}) - \mathcal{T}_\Delta(H)\mathcal{H}_\Delta(\tilde{u})$ is a sequential sequence of zero input responses

Zero input response (cont.)

another approach: with g a solution of the system

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

$$\boxed{\tilde{y}_0 = Y_{\mathsf{f}}\, g}$$

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

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

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

More samples of the free response

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

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

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

Balanced state sequence

with $H = \begin{bmatrix} H(0) \\ H(1) \\ \cdots \end{bmatrix}$, σH denotes the shift-and-cut seq. $\begin{bmatrix} H(1) \\ H(2) \\ \cdots \end{bmatrix}$

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

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

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

matrix of sequential zero input responses: Y_0

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

Balanced model estimation by LS

$$X_{\text{bal}} = \begin{bmatrix} x_{\text{n}_{\max}+1} & x_{\text{n}_{\max}+2} & \cdots & x_{\text{n}_{\max}+T+1-L} \end{bmatrix}$$

$$\begin{bmatrix} x_{\text{n}_{\max}+2} & x_{\text{n}_{\max}+3} & \cdots & x_{\text{n}_{\max}+T+1-L} \\ y_{\text{n}_{\max}+1} & y_{\text{n}_{\max}+2} & \cdots & y_{\text{n}_{\max}+T-L} \end{bmatrix} =$$

$$\begin{bmatrix} \hat{A} & \hat{B} \\ \hat{C} & \hat{D} \end{bmatrix} \begin{bmatrix} x_{\text{n}_{\max}+1} & x_{\text{n}_{\max}+2} & \cdots & x_{\text{n}_{\max}+T-L} \\ u_{\text{n}_{\max}+1} & u_{\text{n}_{\max}+2} & \cdots & u_{\text{n}_{\max}+T-L} \end{bmatrix} \quad \text{(LS)}$$

A new algorithm

input: $\tilde{u}(1), \dots, \tilde{u}(T), \quad \tilde{y}(1), \dots, \tilde{y}(T)$

an upper bound $\mathrm{n}_{\mathrm{max}}$ for the system order

1. **zero input response:** $Y_0 = Y_{\mathrm{f}} G$, 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 $\begin{bmatrix} U_{\mathrm{p}} \\ U_{\mathrm{f}} \\ Y_{\mathrm{p}} \end{bmatrix} G = \begin{bmatrix} 0 \\ \left[\begin{smallmatrix} I \\ 0 \end{smallmatrix}\right] \\ 0 \end{bmatrix}$

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: $\tilde{u}_0, \ldots, \tilde{u}_T \quad \tilde{y}_0, \ldots, \tilde{y}_T$ **and** $i, \quad i \geq \mathtt{n}_{\max}$

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

row dim$(U_{\mathsf{p}}) = i\mathtt{m}$

row dim$(U_{\mathsf{f}}) = i\mathtt{m}$

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

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

Comments

- the oblique proj. $Y_{\mathsf{f}}/_{U_{\mathsf{f}}}\begin{bmatrix} U_{\mathsf{p}} \\ Y_{\mathsf{p}} \end{bmatrix}$ contains seq. 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 $\mathtt{n}_{\max}$ 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

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

$Y_\mathsf{f}/_{U_\mathsf{f}} \left[\begin{smallmatrix} U_\mathsf{p} \\ Y_\mathsf{p} \end{smallmatrix}\right]$ 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

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

$$\text{row dim}(U_\mathsf{p}) = \mathrm{n}_{\max}\mathrm{m} \qquad \text{row dim}(Y_\mathsf{p}) = \mathrm{n}_{\max}\mathrm{p}$$

$$\text{row dim}(U_\mathsf{f}) = \mathrm{n}_{\max}\mathrm{m} \qquad \text{row dim}(Y_\mathsf{f}) = \mathrm{n}_{\max}\mathrm{p}$$

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

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

Algorithm Moonen–Ramos

input: $\tilde{u}_0, \ldots, \tilde{u}_T, \quad \tilde{y}_0, \ldots, \tilde{y}_T$ **and** $\mathrm{n}_{\max}$

$$\begin{bmatrix} U_\mathrm{p} \\ U_\mathrm{f} \end{bmatrix} := \mathcal{H}_{2\mathrm{n}_{\max}}(\tilde{u}), \quad \begin{bmatrix} Y_\mathrm{p} \\ Y_\mathrm{f} \end{bmatrix} := \mathcal{H}_{2\mathrm{n}_{\max}}(\tilde{y})$$

row dim$(U_\mathrm{p}) = \mathrm{n}_{\max}\mathrm{m}$

row dim$(U_\mathrm{f}) = \mathrm{n}_{\max}\mathrm{m}$

0. **annihilators:** $[T_1 \; T_2 \; T_3 \; T_4]$
1. **free response:** $Y_0 = T_4^{+}[T_1 \; T_2] \begin{bmatrix} U_\mathrm{p} \\ Y_\mathrm{p} \end{bmatrix}$
2. **impulse response:** $T_4^{+}(T_2 T_4^{+} T_3 - T_1)$
3. **SVD:** $\mathfrak{H} = T_4^{+}(T_2 T_4^{+} T_3 - T_1) = U \Sigma V^\top$
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_i
  efficient implementation should exploit the Hankel structure
- we have a "dual" algorithm, to the one discussed, 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 T_i
- 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 we do not discuss numerical efficiency
(depends heavily on the implementation)

example used in all experiments:

third order random stable SISO system

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

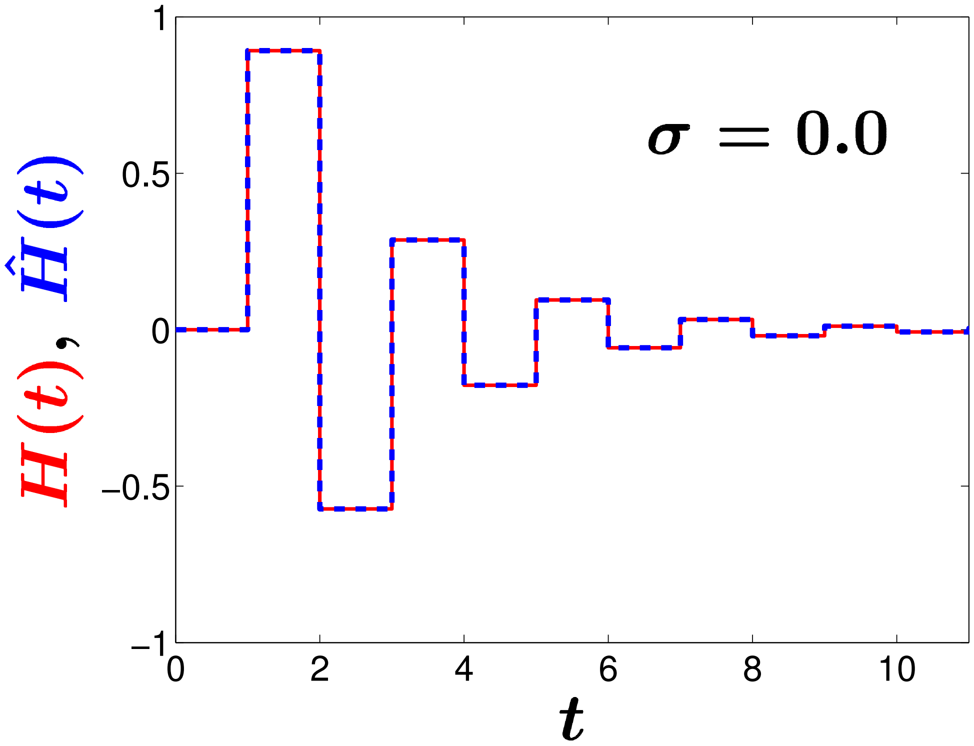
$\tilde{w}$ **is corrupted** by white noise with standard deviation σ

if not stated otherwise: $\mathrm{n}_{\max} = \mathrm{n}$ and $L = \mathrm{n}$

Impulse response estimation

solid red — exact impulse response H

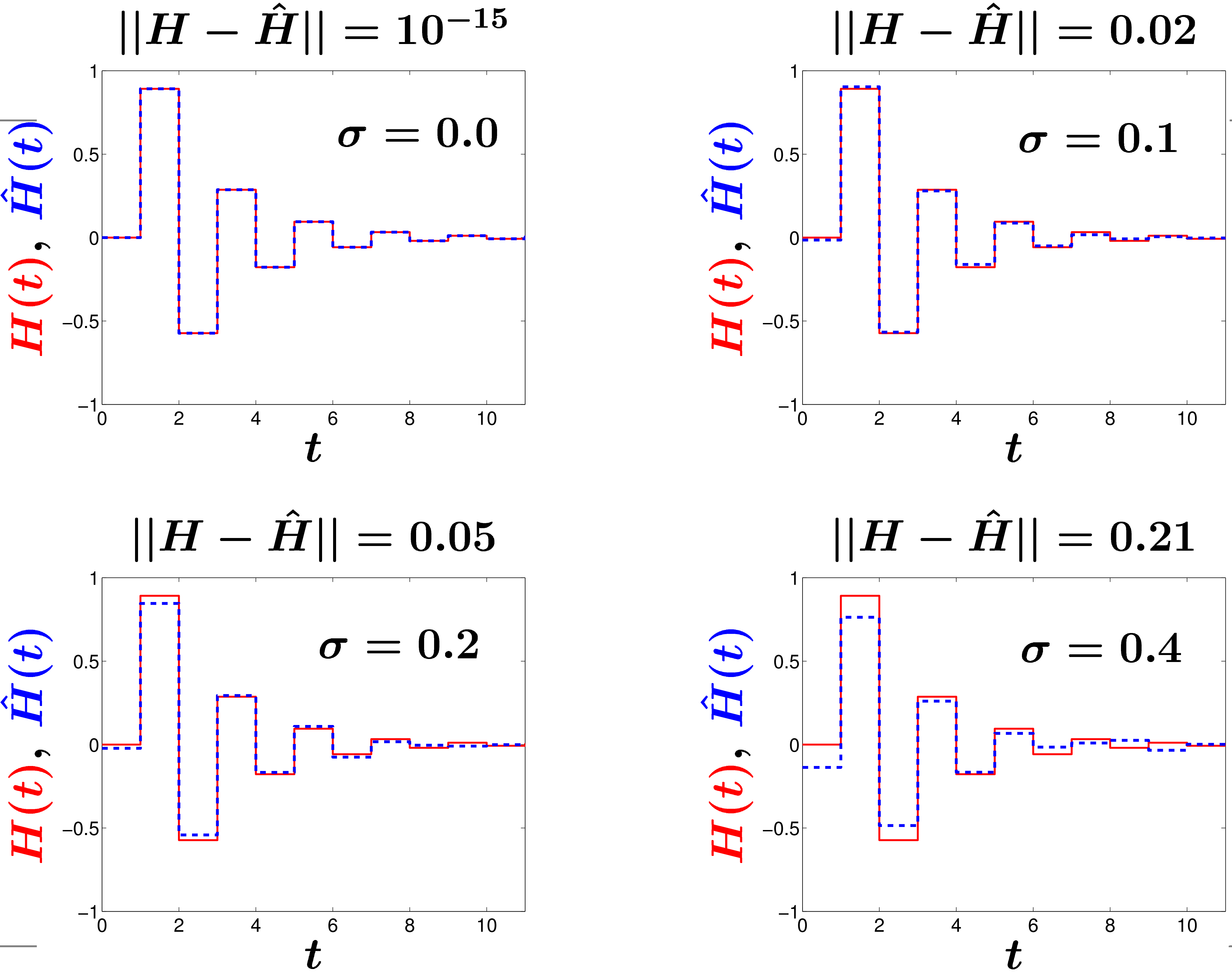
dashed blue — impulse response computed from data $\hat{H}$



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

up to the numerical precision

exact match


||H − Ĥ|| = 10^−15
σ = 0.0
||H − Ĥ|| = 0.02
σ = 0.1
||H − Ĥ|| = 0.05
σ = 0.2
||H − Ĥ|| = 0.21
σ = 0.4
H(t), Ĥ(t)
t
1
0.5
0
−0.5
−1
0
2
4
6
8
10

Free response estimation

$Y_0 = \Gamma X$ — exact sequence of free responses

$\hat{Y}_0$ — 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

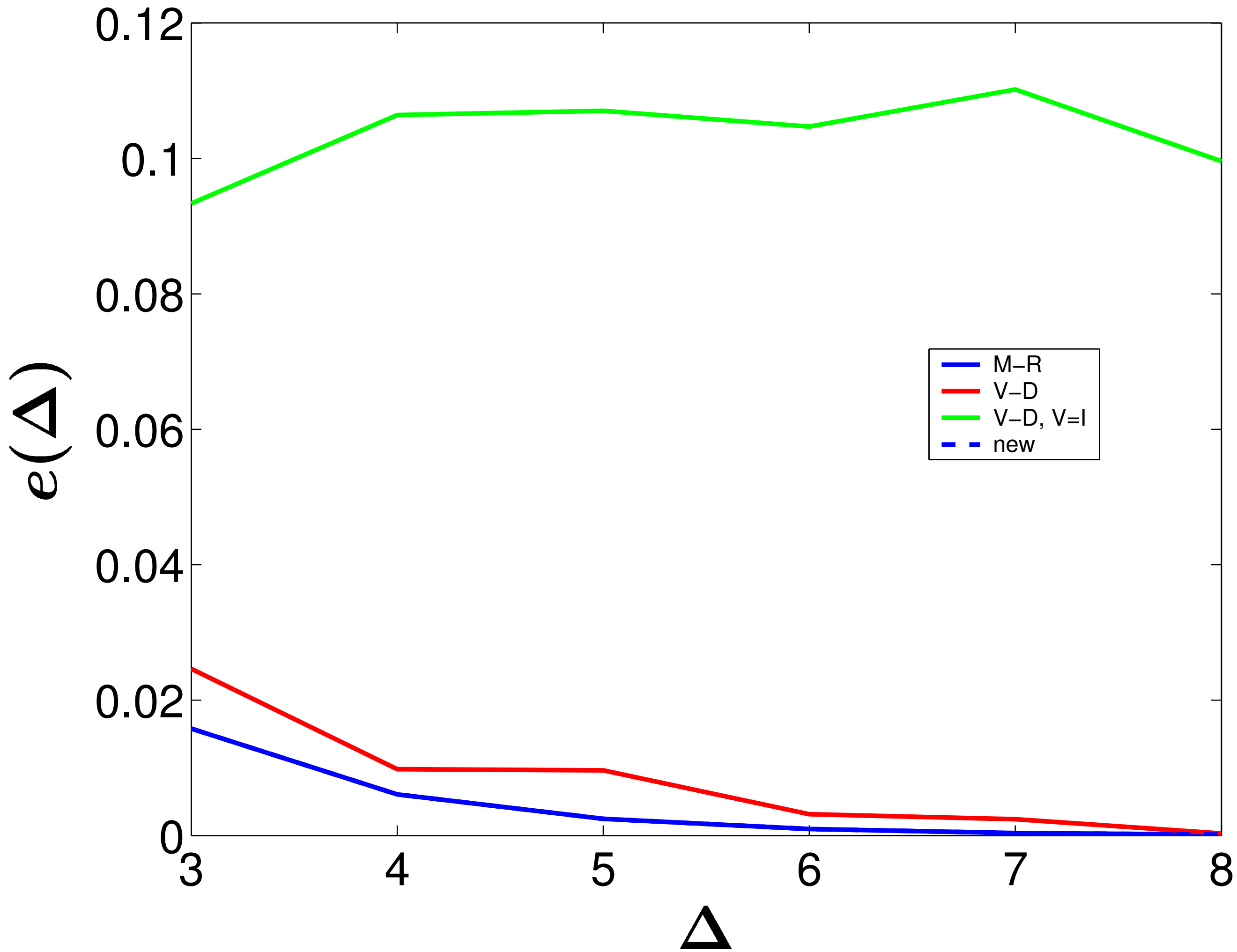
closeness to exact balancing

$\mathcal{C}/\mathcal{O}$ — **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.)

green — VO–DM, $V = I$ red — VO–DM blue — M–R
new ≡ M–R



Conclusions and discussion

Conclusions

- impulse response and sequential sequence of zero input responses are the **main tools for balanced model identification**
- they are classically computed via the **oblique projection**
- we showed **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: **efficient numerical implementation**