Recursive computation of the most powerful unfalsified model

Jan C. Willems, Ivan Markovsky, and Bart De Moor

K.U.Leuven, ESAT-SISTA

Outline

Recursive LTI modeling

Our aim:

exact recursive modeling of time series by LTI systems

The algorithms can work recursively in:

1. the number of time series,
2. the length of the time series (real-time),
3. the length of the system laws,

or a combination of these.

recursive $\approx$ computationally efficient algorithms

System identification: $w_{\mathrm{d}} \mapsto \widehat{\mathscr{B}} \in \mathscr{M}$

Notation:

- $w_{\mathrm{d}} = (u_{\mathrm{d}}, y_{\mathrm{d}})$ — given data, in this talk a vector time series
- $\widehat{\mathscr{B}}$ — to be found model for w_{d}, in this talk an LTI system
- $\mathscr{M}$ — model class, in this talk the set of LTI systems $\mathscr{L}$

System identification

- defines a mapping $w_{\mathrm{d}} \mapsto \mathscr{B}$
- derives effective algorithms that realize the mapping, and
- develops efficient software that implements the algorithms

Exact identification: two points of view

Find the true data generating system

- assume that $w_d \in \bar{\mathscr{B}} \in \mathscr{L}$
- find back $\bar{\mathscr{B}}$ from w_d (and an upper bound of the order)
- this is possible provided $\bar{\mathscr{B}}$ is controllable and an input component of w_d is persistently exciting

Find the least complex LTI system that fits w_d

- no assumption about w_d
- find $\widehat{\mathscr{B}} \in \mathscr{L}$ with minimal # of inputs and order, s.t. $w_d \in \widehat{\mathscr{B}}$
- $\widehat{\mathscr{B}}$—most powerful unfalsified model (MPUM) for w_d in $\mathscr{L}$

The most powerful unfalsified model (MPUM)

$\mathscr{L}^{\mathtt{w}}$ — LTI model class with $\mathtt{w}$ external variables

$\mathscr{B} \in \mathscr{L}^{\mathtt{w}}$ is a linear, shift-invariant, closed subspace of $(\mathbb{R}^{\mathtt{w}})^{\mathbb{N}}$

$\mathscr{B}$ is the MPUM of w_d in $\mathscr{L}^{\mathtt{w}}$ if

1. $\mathscr{B}$ is in the model class $\mathscr{L}^{\mathtt{w}}$, *i.e.*, $\mathscr{B} \in \mathscr{L}^{\mathtt{w}}$
2. $\mathscr{B}$ is unfalsified by w_d, *i.e.*, $w_d \subseteq \mathscr{B}$, and
3. any other unfalsified system in $\mathscr{L}^{\mathtt{w}}$ is less powerful, *i.e.*, $\mathscr{B}' \in \mathscr{L}^{\mathtt{w}}$ and $w_d \subseteq \mathscr{B}'$ imply $\mathscr{B} \subseteq \mathscr{B}'$.

LTI model representations

- Kernel representation (parameter $R(z) := \sum_{i=0}^{\mathtt{l}} R_i z^i$)

$$R_0 w(t) + R_1 w(t+1) + \cdots + R_{\mathtt{l}} w(t+\mathtt{l}) = 0$$

- Impulse response represent (parameter $h : \mathbb{Z} \to \mathbb{R}^{\mathtt{p}\times\mathtt{m}}$)

$$w = \operatorname{col}(u,y), \qquad y(t) = \sum_{\tau=-\infty}^{t} h(\tau) u(t-\tau)$$

- Input/state/output representation (parameter (A,B,C,D))

$$w = \operatorname{col}(u,y), \qquad \begin{aligned} x(t+1) &= Ax(t) + Bu(t) \\ y(t) &= Cx(t) + Du(t) \end{aligned}$$

$\mathtt{p} := \dim(y) = \operatorname{rowdim}(R)$ is the # of outputs

$\mathtt{m} := \dim(u)$ is the # of inputs, $\quad \mathtt{l} := \operatorname{degree}(R)$ is the lag

Laws of the system

Consider a system $\mathscr{B} \in \mathscr{L}^{\mathtt{w}}$ defined by

$$\mathscr{B} := \{ w \in (\mathbb{R}^{\mathtt{w}})^{\mathbb{Z}} \mid R_0 w(t) + R_1 w(t+1) + \cdots + R_{\mathtt{l}} w(t+\mathtt{l}) = 0 \}$$

and let $R(z) := \sum_{i=0}^{\mathtt{l}} R_i z^i \in \mathbb{R}^{\mathtt{p}\times\mathtt{w}}[z]$.

The rows $r_1(z), \ldots, r_{\mathtt{p}}(z)$ of $R(z)$ are annihilators or laws of $\mathscr{B}$.

Define the shift operator σ: $\sigma w(t) := w(t+1)$

$$r_i(\sigma)\mathscr{B} = 0, \qquad \text{for } i = 1, \ldots, \mathtt{p}$$

Hankel matrix of the data

$$\mathscr{H}_{l+1}(w_\mathrm{d}) := \begin{bmatrix} w_\mathrm{d}(1) & w_\mathrm{d}(2) & \cdots & w_\mathrm{d}(T-l) \\ w_\mathrm{d}(2) & w_\mathrm{d}(3) & \cdots & w_\mathrm{d}(T-l+1) \\ w_\mathrm{d}(3) & w_\mathrm{d}(4) & \cdots & w_\mathrm{d}(T-l+2) \\ \vdots & \vdots & & \vdots \\ w_\mathrm{d}(l+1) & w_\mathrm{d}(\Delta+1) & \cdots & w_\mathrm{d}(T) \end{bmatrix}$$

The left kernel of $\mathscr{H}_{l+1}(w_\mathrm{d})$

$$\mathscr{N}_{l+1}(w_\mathrm{d}) := \operatorname{left\,ker}\left(\mathscr{H}_{l+1}(w_\mathrm{d})\right)$$

contains laws of $\mathscr{B}_\mathrm{mpum}(w_\mathrm{d})$.

Moreover, if $\mathscr{B}_\mathrm{mpum}(w_\mathrm{d})$ is controllable, all laws of $\mathscr{B}_\mathrm{mpum}(w_\mathrm{d})$ are contained in a basis of $\mathscr{N}_{l+1}(w_\mathrm{d})$ for $l \geq \mathtt{l}$ (the lag of $\mathscr{B}_\mathrm{mpum}$).

Identification algorithm: $w_\mathrm{d} \to R(z)$

Let the rows of R form a basis for $\mathscr{N}_{l+1}(w_\mathrm{d})$, *i.e.*,

$$R\mathscr{H}_{l+1}(w_\mathrm{d}) = 0, \quad \operatorname{rank}(R) = \dim\left(\mathscr{N}_{l+1}(w_\mathrm{d})\right).$$

Define a polynomial matrix $R(z)$ constructed from R,

$$R(z) = \sum_{i=0}^{l} R_i z^i \in \mathbb{R}^{g\times \mathtt{w}}[z],$$

where $R =: \begin{bmatrix} R_0 & R_1 & \cdots & R_l \end{bmatrix}$, $R_i \in \mathbb{R}^{g\times\mathtt{w}}$. Then

$$\ker\left(R(\sigma)\right) = \mathscr{B}_\mathrm{mpum}(w_\mathrm{d}).$$

Inefficiency of the algorithm

Suppose that $\mathscr{H}_{l+1}(w_\mathrm{d})$ is rank deficient and let

$$\underbrace{\begin{bmatrix} r_0 & r_1 & \cdots & r_l \end{bmatrix}}_{r} \mathscr{H}_{l+1}(w_\mathrm{d}) = 0.$$

Then due to the Hankel structure for $L > l$

$$\begin{bmatrix} r_0 & r_1 & \cdots & r_l & & & 0 \\ & r_0 & r_1 & \cdots & r_l & & \\ & & \cdots & \cdots & & \cdots & \\ 0 & & & r_0 & r_1 & \cdots & r_l \end{bmatrix} \mathscr{H}_{L+1}(w_\mathrm{d}) = 0.$$

Moreover, extra work has to be done in order to "extract" from $\operatorname{left\,ker}\left(\mathscr{H}_{L+1}(w_\mathrm{d})\right)$ a short law r.

In polynomial language, $R(z)$ is not row reduced and an extra work is needed for its row reduction.

Lag-recursive algorithm

Assume that the k shortest laws $r^{(1)}, \ldots, r^{(k)}$ have been computed

$$r^{(i)}(\sigma)w_\mathrm{d} = 0, \quad \text{for } i = 1, \ldots, k.$$

Question: How to use this information in order to compute the remaining laws $r^{(k+1)}, \ldots, r^{(\mathrm{p})}$ of $\mathscr{B}_\mathrm{mpum}(w_\mathrm{d})$ efficiently?

Assuming that the system induced by the matrix

$$R' := \operatorname{col}\left(r^{(1)}, \ldots, r^{(k)}\right)$$

is controllable, this can be done as follows:

1. Compute D, such that $\operatorname{col}(R', D)$ is unimodular.
2. Define $e := D(\sigma)w_\mathrm{d}$.
3. Compute E, such that $\mathscr{B}_\mathrm{mpum}(e) = \ker\left(E(\sigma)\right)$.
4. Define $R := (R', ED)$.

Lag-recursive algorithm (cont.)

Lemma

- Assume that $\mathscr{B} := \ker\left(R(\sigma)\right)$ is controllable.
- Let $R'(\sigma)\mathscr{B} = 0$ and $\mathscr{B}' := \ker\left(R'(\sigma)\right)$ is controllable.
- Let D be such that $\mathrm{col}(R', D)$ is unimodular.

Then there exist U and F, such that $UR = \mathrm{col}(R', FD)$.

Proposition

- Assume that $\mathscr{B}_{\mathrm{mpum}}(w_{\mathrm{d}})$ is controllable.
- Let $R'(\sigma)w_{\mathrm{d}} = 0$ and $\ker\left(R'(\sigma)\right)$ is controllable.
- Let D be such that $\mathrm{col}(R', D)$ is unimodular.
- Define $e := D(\sigma)w_{\mathrm{d}}$ and let $\ker\left(E(\sigma)\right) = \mathscr{B}_{\mathrm{mpum}}(e)$.

Then $\ker\left(\mathrm{col}(R', ED)(\sigma)\right) = \mathscr{B}_{\mathrm{mpum}}(w_{\mathrm{d}})$.

Lag-recursive algorithm (cont.)

Computing a D, such that $\mathrm{col}(r, D)$ is unimodular is a Bézout-type problem.

Assuming that r_1 and r_2 of r are coprime, we solve the Bézout equation

$$r_1 b - r_2 a = 1,$$

and define

$$D = \begin{bmatrix} a & b & 0 \\ 0 & 0 & I_{\mathtt{w}-2} \end{bmatrix},$$

where $I_{\mathtt{w}-2}$ is the $(\mathtt{w}-2) \times (\mathtt{w}-2)$ identity matrix.

Applying the procedure recursively leads to a lag-recursive algorithm for computing a kernel representation of $\mathscr{B}_{\mathrm{mpum}}(w_{\mathrm{d}})$.

$R = \mathtt{lag\_recursive\_mpum}(w_{\mathrm{d}})$

```
if w_d = 0 then
   Let R := [].
else
   Find the smallest l ∈ {0,1,2,...}, for which H_{l+1}(w_d) is
   rank deficient, or define l := −1 if there is no such l.
   if l = −1 then
      Let R := [].
   else
      Compute r =: [r_0  r_1  ···  r_l], r_i ∈ ℝ^{1×w} in
      leftker(H_{l+1}(w_d)) and let r(z) := Σ_{i=0}^{l} r_i z^i.
      Find D, such that col(r, D) is unimodular.
      Compute e := D(σ)w_d.
      r' := lag_recursive_mpum(e).
      if r' = [], then R := r else R := col(r, r'D) end if
   end if
end if
```


Simulation example

Consider the 2 outputs 1 input system $\mathscr{B}$ induced by

$$R(z) = \begin{bmatrix} r_1(z) \\ r_2(z) \end{bmatrix} = \begin{bmatrix} -1.73 & 1.37 & -2.03 \\ 0.35 & -0.50 & -0.24 \end{bmatrix} z^0 + \begin{bmatrix} 5.97 & -8.36 & 12.25 \\ 0.16 & 0.65 & 0.19 \end{bmatrix} z^1 + \begin{bmatrix} 2.00 & 7.82 & -11.33 \\ 0.00 & 0.00 & 0.00 \end{bmatrix} z^2.$$

$\mathscr{B}$ has two laws: $r_1(z)$ is of degree 2 and $r_2(z)$ is of degree 1

Simulation example (cont.)

$$\dim\big(\underbrace{\operatorname{left}\ker(\mathscr{H}_2(w_\mathrm{d}))}_{\mathscr{N}_2(w_\mathrm{d})}\big) = 1 \text{ and } \mathscr{N}_2(w_\mathrm{d}) = \operatorname{span}(r_2(z))$$

$$\mathscr{N}_2(w_\mathrm{d}) = \operatorname{span}\left(\begin{bmatrix}0.35 & -0.50 & -0.24 & 0.16 & 0.65 & 0.19\end{bmatrix}^\top\right).$$

$$\dim\big(\mathscr{N}_3(w_\mathrm{d})\big) = 3, \text{ and } \mathscr{N}_3(w_\mathrm{d}) = \operatorname{span}(r_2(z), zr_2(z), r_1(z))$$

$$\operatorname{span}\left(\begin{bmatrix}0.35 & -0.5 & -0.24 & 0.16 & 0.65 & 0.19 & 0.00 & 0.00 & 0.00\\ 0.00 & 0.00 & 0.00 & 0.35 & -0.50 & -0.24 & 0.16 & 0.65 & 0.19\\ -1.7 & 1.37 & -2.03 & 5.97 & -8.36 & 12.25 & 2.00 & 7.82 & -11.3\end{bmatrix}^\top\right).$$

In general, $\dim\big(\mathscr{N}_{l+1}(w_\mathrm{d})\big) = 2l-1$, for $l \geq 1$.

Simulation example (cont.)

Let w_d be a random trajectory of $\mathscr{B}$.

Applied on w_d, the proposed algorithm returns the matrix

$$\widehat{R}(z) = \begin{bmatrix}0.36 & -0.52 & -0.24\\ 0.00 & 0.03 & 0.13\end{bmatrix} z^0 + \begin{bmatrix}0.17 & 0.67 & 0.20\\ -0.03 & -0.12 & -0.74\end{bmatrix} z^1 + \begin{bmatrix}0.00 & 0.00 & 0.00\\ 0.03 & 0.13 & 0.64\end{bmatrix} z^2$$

which is up to pre-multiplication by unimodular matrix equal to R

$$\mathscr{B} = \ker\big(\widehat{R}(\sigma)\big).$$

Conclusions

- Exact recursive in the lag of the laws modeling.
- Avoid the repeated calculation of one and the same law.
- Main idea: project the time series on the orthogonal complement of the currently found laws.
- This is achieved by solving a Bézout equation.

Thank you