A software package for exact linear system identification

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

K.U.Leuven, ESAT-SISTA

Outline

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 $w_\mathrm{d} \in \overline{\mathscr{B}} \in \mathscr{L}$
- find back $\overline{\mathscr{B}}$ from w_d (and an upper bound of the order)
- this is possible provided $\overline{\mathscr{B}}$ 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. $w_\mathrm{d} \in \widehat{\mathscr{B}}$
- $\widehat{\mathscr{B}}$—most powerful unfalsified model (MPUM) for w_d in $\mathscr{L}$

Exact identification: not a practical SYSID problem

w_d can always be fitted exactly

- take all variables as inputs
- for finite $w_d \in (\mathbb{R}^w)^T$, take the order sufficiently large

$\rightsquigarrow$ trivial solution

Of interest are a nontrivial solutions, i.e., we want

$\mathscr{M}$ to be a set of bounded complexity LTI systems $\mathscr{L}_{\mathtt{m},\mathtt{n}_{\max}}$,
# of inputs $\leq \mathtt{m}$ and order $\leq \mathtt{n}_{\max}$.

However,

- $w_d \in \bar{\mathscr{B}} \in \mathscr{L}_{\mathtt{m},\mathtt{n}_{\max}}$ is a too restrictive assumption

alternatively

- the MPUM generically does not exist in $\mathscr{L}_{\mathtt{m},\mathtt{n}_{\max}}$

Approximate identification: optimization point of view

- the model need not fit the data exactly
- choose a distance measure $M(w_d, \mathscr{B})$ between w_d and $\mathscr{B}$
- minimize $M(w_d, \mathscr{B})$ over all models in $\mathscr{M}$

Computing $M(w_d, \mathscr{B})$ is equivalent to

- finding the “best” approximation of w_d in $\mathscr{B}$,
- smoothing or filtering (if causality is imposed) w_d by $\mathscr{B}$,
- projecting w_d on $\mathscr{B}$.

$M(w_d, \mathscr{B})$ can be computed in various ways:
smoothing, spectral factorization, Cholesky factorization, . . .

Approximate identification: suboptimal methods

exact identification is more than an academic problem

it leads to suboptimal approximate identification methods

an exact ID method can be used for approximate SYSID by

- rank computation $\leftrightarrow$ numerical rank computation
- solution of equations $\leftrightarrow$ least squares

MATLAB does this substitution automatically where necessary

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 $h : \mathbb{Z} \to \mathbb{R}^{p \times m}$)

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

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

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

$\mathtt{p} := \dim(y) = \mathrm{row\,dim}(R)$ is the # of outputs
$\mathtt{m} := \dim(u)$ is the # of inputs, $\mathtt{l} := \mathrm{degree}(R)$ is the lag

Algorithms for exact identification

1. $w_{\rm d} \mapsto R(z)$
2. $w_{\rm d} \mapsto$ impulse response H
3. $w_{\rm d} \mapsto (A,B,C,D)$ (possibly balanced)

   3.1 $w_{\rm d} \mapsto R(\xi) \mapsto (A,B,C,D)$ or $w_{\rm d} \mapsto H \mapsto (A,B,C,D)$

   3.2 $w_{\rm d} \mapsto \mathscr{O}_{1_{\max}+1}(A,C) \mapsto (A,B,C,D)$

   3.3 $w_{\rm d} \mapsto \big(x_{\rm d}(1),\dots,x_{\rm d}(n_{\max}+m+1)\big) \mapsto (A,B,C,D)$

Various ways to implement the mapping $w_{\rm d} \mapsto (A,B,C,D)$.

$w_{\rm d} \mapsto R(z)$

The difference equation representation

$$R_0 w_{\rm d}(t) + R_1 w_{\rm d}(t+1) + \cdots + R_l w_{\rm d}(t+l) = 0, \quad \text{for } t = 1,\dots,T-l$$

is equivalent to the linear system of equations

$$\underbrace{\begin{bmatrix} R_0 & R_1 & \cdots & R_l \end{bmatrix}}_{R} \underbrace{\begin{bmatrix} w_{\rm d}(1) & w_{\rm d}(2) & \cdots & w_{\rm d}(T-l) \\ w_{\rm d}(2) & w_{\rm d}(3) & \cdots & w_{\rm d}(T-l+1) \\ \vdots & \vdots & & \vdots \\ w_{\rm d}(l+1) & w_{\rm d}(l+2) & \cdots & w_{\rm d}(T) \end{bmatrix}}_{\mathscr{H}_{l+1}(w_{\rm d})} = 0.$$

Finding R, requires to compute the left kernel of $\mathscr{H}_{l+1}(w_{\rm d})$.

$w_{\rm d} \mapsto H$

Assuming $(u_{\rm d}, y_{\rm d}) \in \mathscr{B}$, $\mathscr{B}$ controllable, and $u_{\rm d}$ persist. exciting, there is G, such that $H = \mathscr{H}_t(y_{\rm d})G$.

$w_{\rm d} \mapsto H$ reduces to the problem of finding a particular G.

$$\begin{bmatrix} \mathscr{H}_{l+t}(u_{\rm d}) \\ \hline \mathscr{H}_{l+t}(y_{\rm d}) \end{bmatrix} G = \begin{bmatrix} 0 \\ \begin{bmatrix} I \\ 0 \end{bmatrix} \\ \hline 0 \\ H \end{bmatrix} \begin{array}{l} \leftarrow \ l \text{ zero samples} \\ \leftarrow \ t \text{ samples long impulse} \\ \hline \leftarrow \ l \text{ zero samples} \\ \leftarrow \ t \text{ samples impulse response} \end{array}$$

Block algorithm $w_{\rm d} \mapsto H$

1. Solve the system of equations in blue for G.
2. Substitute G in the equations in red $\rightsquigarrow$ H.

Simulation example $w_{\rm d} \mapsto H$

Simulation setup

- $\mathscr{B}$ is of order $n = 4$, lag $l = 2$, with $m = 2$ inputs, and $p = 2$ outputs
- $w_{\rm d}$ is a trajectory of $\mathscr{B}$ with length $T = 500$

Compared algorithms

- the block algorithm
- an iterative refinement of the block algorithm
- the function `impulse` from the Identification Toolbox

Approximation error $e = ||H - \hat{H}||_{\rm F}$ and execution time

method	error, e	time, sec.
block algorithm	10^{-14}	0.293

$w_\mathrm{d} \mapsto (A,B,C,D)$

Possible paths to go

- $w_\mathrm{d} \mapsto H(0:2\mathtt{l}_\mathrm{max})$ or $R(z) \xrightarrow{\text{realization}} (A,B,C,D)$
- $w_\mathrm{d} \mapsto \mathscr{O}_{\mathtt{l}_\mathrm{max}+1}(A,C) \xrightarrow{(1)} (A,B,C,D)$
- $w_\mathrm{d} \mapsto \big(x_\mathrm{d}(1),\dots,x_\mathrm{d}(\mathtt{n}_\mathrm{max}+\mathtt{m}+1)\big) \xrightarrow{(2)} (A,B,C,D)$

(1) and (2) are easy

$$\mathscr{O}_{\mathtt{l}_\mathrm{max}+1}(A,C) \mapsto (A,C) \quad \text{and} \quad (u_\mathrm{d},y_\mathrm{d},A,C) \mapsto (B,C,x_\mathrm{ini}) \qquad (1)$$

$$\begin{bmatrix} x_\mathrm{d}(2) & \cdots & x_\mathrm{d}(\mathtt{n}_\mathrm{max}+\mathtt{m}+1) \\ y_\mathrm{d}(1) & \cdots & y_\mathrm{d}(\mathtt{n}_\mathrm{max}+\mathtt{m}) \end{bmatrix} = \begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} x_\mathrm{d}(1) & \cdots & x_\mathrm{d}(\mathtt{n}_\mathrm{max}+\mathtt{m}) \\ u_\mathrm{d}(1) & \cdots & u_\mathrm{d}(\mathtt{n}_\mathrm{max}+\mathtt{m}) \end{bmatrix} \qquad (2)$$

SISTA

$\mathscr{O}_{\mathtt{l}_\mathrm{max}+1}(A,C) \mapsto (A,B,C,D)$

First C and A

C is the first block entry of $\mathscr{O}_{\mathtt{l}_\mathrm{max}+1}(A,C)$ and A is given by

$$\big(\sigma^*\mathscr{O}_{\mathtt{l}_\mathrm{max}+1}(A,C)\big)A = \big(\sigma\mathscr{O}_{\mathtt{l}_\mathrm{max}+1}(A,C)\big) \qquad \text{shift equation}$$

(σ^* removes the last block entry and σ removes the first block entry)

Then D, B, and $x_\mathrm{d}(1)$

Once C and A are known, the system of equations

$$y_\mathrm{d}(t) = CA^t x_\mathrm{d}(1) + \sum_{\tau=1}^{t-1} CA^{t-1-\tau} B u_\mathrm{d}(\tau) + D\delta(t+1),$$

is linear in D, B, $x_\mathrm{d}(1)$ and can be solved explicitly.

SISTA

$w_\mathrm{d} \mapsto \mathscr{O}_{\mathtt{l}_\mathrm{max}+1}(A,C)$

- the columns of $\mathscr{O}_{\mathtt{l}_\mathrm{max}+1}(A,C)$ are lin. indep. free resp. of $\mathscr{B}$
- under the conditions of FL, such resp. can be computed

$$\begin{bmatrix} \mathscr{H}_t(u_\mathrm{d}) \\ \mathscr{H}_t(y_\mathrm{d}) \end{bmatrix} G = \begin{bmatrix} 0 \\ Y_0 \end{bmatrix} \quad \begin{matrix} \leftarrow & \text{zero inputs} \\ \leftarrow & \text{free responses} \end{matrix}$$

  (G should be maximal rank)

- once we have a maximal rank matrix of free responses Y_0

$$Y_0 = \mathscr{O}_{\mathtt{l}_\mathrm{max}+1}(A,C) \underbrace{\begin{bmatrix} x_{\mathrm{ini},1} & \cdots & x_{\mathrm{ini},j} \end{bmatrix}}_{X_\mathrm{ini}} \qquad \text{rank revealing factorization}$$

- the factorization fixes the basis for $\mathscr{O}_{\mathtt{l}_\mathrm{max}+1}(A,C)$ and X_ini

SISTA

$w_\mathrm{d} \mapsto \big(x_\mathrm{d}(1),\dots,x_\mathrm{d}(\mathtt{n}_\mathrm{max}+\mathtt{m}+1)\big)$

Main idea

If the free resp. are sequential, i.e., if Y_0 is block-Hankel, then X_ini is a state sequence of $\mathscr{B}$.

Computation of sequential free responses

$$\begin{bmatrix} U_\mathrm{p} \\ Y_\mathrm{p} \\ U_\mathrm{f} \end{bmatrix} G = \begin{bmatrix} U_\mathrm{p} \\ Y_\mathrm{p} \\ 0 \end{bmatrix} \quad \begin{matrix} \left.\begin{matrix} \\ \\ \end{matrix}\right\} \text{sequential ini. conditions} \\ \leftarrow \quad \text{zero inputs} \end{matrix}$$

$$Y_\mathrm{f} \quad G \quad = \quad Y_0$$

$$Y_0 = \mathscr{O}_{\mathtt{l}_\mathrm{max}+1}(A,C) \begin{bmatrix} x_\mathrm{d}(1) & \cdots & x_\mathrm{d}(\mathtt{n}_\mathrm{max}+\mathtt{m}+1) \end{bmatrix} \qquad \text{rank revealing factorization}$$

SISTA

Building blocks for the algorithms

Function	Description
`w2r`	from data (time series w) to a kernel repr.
`r2pq`	from a kernel repr. to an LMF representation
`pq2ss`	from an LMF repr. to an I/S/O representation
`uy2h`	from data to the impulse response
`h2ss`	from the impulse resp. to an I/S/O repr.
`uy2y0`	from data to sequential free responses
`y02ox`	from free responses to an observability matrix and a state sequence
`h2ox`	from the impulse response to an observability matrix and a state sequence
`uy02ss`	from data and an observability matrix to an I/S/O representation
`uyx2ss`	from data and a state seq. to an I/S/O repr.
`hy02xbal`	from the impulse response and sequential free responses to a balanced state sequence

Conclusions

- choice of representation
- decomposition of the identification problem into standard easy to solve subproblems
- various ways to achieve the mapping $w_{\mathrm{d}} \mapsto \mathscr{B}$
- can be used as suboptimal approximate ID methods
- open question:

> when w_{d} is not exact, which choice of representation and computational algorithm gives best approximate system?

Thank you