Software for approximate linear system identification

I. Markovsky, J. C. Willems, S. Van Huffel, and B. De Moor

K.U.Leuven, ESAT/SCD (SISTA)

44th IEEE Conference on Decision and Control, 12 December 2005, Seville, Spain

Outline

- Discrete-time linear time-invariant (LTI) systems
- Approximate LTI system identification
- Examples
  - approximate realization
  - identification from step response data
  - autonomous system identification
  - model reduction

Discrete-time LTI systems

LTI system: $\mathscr{B} \subset (\mathbb{R}^q)^{\mathbb{Z}}$, *i.e.*, a set of q-variables time-series $w : \mathbb{Z} \to \mathbb{R}^q$

$\mathscr{L}^q$ — class of finite dimensional LTI systems with q variables

input/state/output representation of $\mathscr{B} \in \mathscr{L}^q$

$$w = \begin{bmatrix} u \\ y \end{bmatrix}, \qquad \begin{aligned} x(t+1) &= Ax(t) + Bu(t) \\ y(t) &= Cx(t) + Du(t) \end{aligned} \qquad \text{(I/S/O)}$$

m — number of inputs
p — number of outputs (invariant of the representation)
n — minimal state dimension

Discrete-time LTI systems

difference equation representation of $\mathscr{B} \in \mathscr{L}^q$

$$R_0 w(t) + R_1 w(t+1) + \cdots + R_l w(t+l) = 0 \qquad \text{(DE)}$$

the smallest possible l is called the lag of $\mathscr{B}$ (also invariant)

$\dim(\mathscr{B})$ — complexity of the system, specified by (m,n) or (m,l)

$\mathscr{L}^q_{m,l}$ — model class of LTI systems of bounded complexity
(# inputs $\leq m$, lag $\leq l$)

Approximate LTI system identification

the quality of fit of $w_d \in (\mathbb{R}^q)^T$ ("d" for data) by the model $\mathscr{B} \in \mathscr{L}^q$ is measured by the ℓ_2-distance from w_d to $\mathscr{B}$

$$M(w_d, \mathscr{B}) := \min_{\hat{w} \in \mathscr{B}} \|w_d - \hat{w}\|_{\ell_2}$$

$M(w_d, \mathscr{B})$ is called misfit (lack of fit) between w_d and $\mathscr{B}$

global total least squares problem

$$\hat{\mathscr{B}} := \arg \min_{\mathscr{B} \in \mathscr{L}^q_{m,l}} M(w_d, \mathscr{B}) \qquad \text{(GTLS)}$$

misfit minimization over LTI models of bounded complexity

Solution approach

block-Hankel matrix constructed from $w = \big(w(1), \ldots, w(T)\big)$

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

generically equivalent formulation of (GTLS):

$$\hat{X} = \arg\min_X \left(\min_{\hat{w}} \|w_d - \hat{w}\|^2_{\ell_2} \quad \text{s.t.} \quad \mathscr{H}^\top_{l+1}(\hat{w}) \begin{bmatrix} X \\ -I_p \end{bmatrix} = 0 \right) \qquad \text{(STLS)}$$

known as structured total least squares problem

$$\begin{bmatrix} \hat{X}^\top & -I \end{bmatrix} = \begin{bmatrix} \hat{R}_0 & \hat{R}_1 & \cdots & \hat{R}_l \end{bmatrix}, \qquad \text{diff. eqn. repr. of } \hat{\mathscr{B}}$$

Software

`ident:` $(w_d, m, l) \mapsto (\hat{A}, \hat{B}, \hat{C}, \hat{D})$ solves (GTLS)

`misfit:` $\big(w_d, (A,B,C,D)\big) \mapsto (M, \hat{w}_d)$ misfit computation

calling syntax:

```
[ sysh, info, wh, xini ] = ident( w, m, l, options )
[ M, wh, xini ] = misfit( w, sys, options )
```

- multiple given time series $w_k = \big(w_k(1), \ldots, w_k(T)\big)$ can be treated
- elements of w can be specified as "exact", in which case they are not modified in the approximation $\hat{w}$

Example $l = 0$: Approximation by a static model

with $l = 0$, GTLS $\equiv$ classical total least squares (TLS)

```
>> l = 0; m = 2; p = 3; T = 20;
>>
>> % Generate data
>> n    = l*p;
>> sys0 = drss_(n,p,m);
>> u0   = randn(T,m);
>> y0   = lsim(sys0,u0);
>> w0   = [u0 y0];
>> w    = w0;
>>
>> % Identify the system
>> [sysh,info,wh]=ident(w,m,l);
```

```
>> info
info =
     iter: 1
     time: 0
        M: 1.5838e-16
>>
>> % Verify the results
>> err_sys = norm(sys0 - sysh)
err_sys =    0
>> err_w   = norm(w0 - wh,'fro')
err_w =      0
```

Example: Exact data

if the data is exact, $\hat{\mathscr{B}} \equiv$ data generating system $\bar{\mathscr{B}}$

```
>> l = 2; m = 2; p = 2; T = 100;
>>
>> % Generate data
>> n   = l*p;
>> sys0 = drss_(n,p,m);
>> u0 = randn(T,m);
>> y0 = lsim(sys0,u0);
>> w0 = [u0 y0];
>> w  = w0;
>>
>> % Identify the system
>> [sysh,info,wh]=ident(w,m,l);
```

```
>> info
info =
    iter: 1
    time: 0.0200
       M: 1.8817e-15
>>
>> % Verify the results
>> err_sys = norm(sys0 - sysh)
err_sys = 2.4896e-15
>> err_w   = norm(w0 - wh, 'fro')
err_w =   2.0814e-15
```


Example: Errors-in-variables system identification

in the EIV setup

$$w_{\mathrm{d}} = \bar{w} + \tilde{w}, \quad \text{where} \quad \bar{w} \in \bar{\mathscr{B}} \in \mathscr{L}^{q}_{m,l}, \quad \tilde{w} \sim \mathrm{N}(0, \bar{\sigma}^2 I) \qquad \text{(EIV)}$$

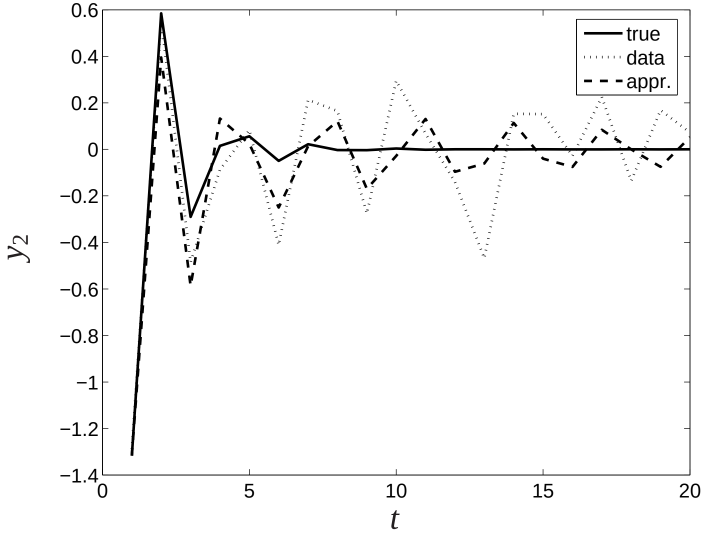
the GTLS solution is a maximum likelihood estimator

```
>> % Perturb the "true" data w0 with noise
>> w = w0 + 0.5 * randn(T,m+p);
>> % Identify the system
>> [sysh,info,wh] = ident(w,m,l);
>> info = iter: 100 time: 1.0100 M: 6.7320
>> % Verify the results
>> err_data = norm(w0-w,'fro')/norm(w0,'fro')
err_data =    0.2605
>> err_appr = norm(w0-wh,'fro')/norm(w0,'fro')
err_appr =    0.1963
```


Example $m = 0$: Output-only identification

```
>> % Generate data
>> T     = 20; % length of the data sequence
>> xini0 = randn(n,1);
>> y0    = initial(sys0,xini0,T);
>> w0    = y0;
>> % Perturb the "true" data w0 with noise
>> w = w0 + 0.25 * randn(T,p); % given data
>>
>> % Identify the system
>> [sysh,info,wh] = ident(w,0,l);
>> info iter: 100 time: 0.1500 M: 1.0719
>>
>> % Verify the results
>> err_data = norm(w0-w,'fro')/norm(w0,'fro')
err_data =    0.7269
>> err_appr = norm(w0-wh,'fro')/norm(w0,'fro')
err_appr =    0.4184
```


Example $m = 0$: Output-only identification



Example: Identification from step response data $s_{\rm d}$

a priori known zero initial conditions and pulse inputs

$$M(s_{\rm d},\mathscr{B}) = \min \|s_{\rm d} - \hat{s}\|_{\ell_2} \quad \text{subject to} \quad \hat{s} \text{ is a step response of } \mathscr{B}$$

multi-input systems $\rightsquigarrow$ multiple (equal length) time series

```
>> % Generate data
>> T   = 150;  % length of the data sequence
>> y0  = step(sys0,T);
>> % Perturb the "true" data y0 with noise
>> y   = y0 + 0.25 * randn(T,p,m);
>>
>> % Construct the inputs
>> u0   = zeros(T,m,m);
>> for i = 1:m, u0(:,i,i) = ones(T,1); end
>> w = [u0 y]; % input/output data
```



```
>> % Precede it with l zeros
>> wext = [zeros(l,m+p,m); w];
>>
>> % Identify the system from the ext. data
>> opt.exct = [1:m]; % exact inputs
>> [sysh,info,whext]=ident(wext,m,l,opt);
>> info
info =
    iter: 100
    time: 3.2200
       M: 5.9504
>> wh = whext(l+1:end,:,:); % Remove the trailing part
>>
>> % Verify the results
>> w0 = [u0 y0];
>> err_data = norm(w0(:)-w(:))/norm(w0(:))
err_data =    0.1711
>> err_appr = norm(w0(:)-wh(:))/norm(w0(:))
err_appr =    0.0384
```


Example: Identification from impulse response data $h_{\rm d}$

identification from exact impulse response — (partial) realization theory

approximation — Kung's algorithm, we will consider misfit minimization

$$M(h_{\rm d},\mathscr{B}) = \min \|h_{\rm d} - \hat{h}\|_{\ell_2} \quad \text{subject to} \quad \hat{h} \text{ is an impulse response of } \mathscr{B}$$

direct approach: I/O ident. with exactly known ini. cond. and input

indirect approach: — an equivalent output only-identification problem

the impulse response of (I/S/O) is equal to the free responses of

$$x(t+1) = Ax(t), \qquad y(t) = Cx(t) \tag{AUT}$$

under initial conditions—the columns of B


```
>> % Generate data
>> T   = 50; % length of the data sequence
>> y0 = impulse(sys0,T);
>> % Perturb the "true" data y0 with noise
>> y   = y0 + 0.25 * randn(T,p,m);
>>
>> % Solution 1: I/O identification.
>> u0 = zeros(T,m,m); u0(1,:,:) = eye(m); % Construct the inputs
>> w = [u0 y]; % given input/output data
>> wext = [zeros(l,m+p,m); w]; % Precede w with l zeros
>>
>> % Identify the system from the ext. data
>> opt.exct = [1:m]; % exact input
>> [sysh1,info1,whext] = ident(wext,m,l,opt);
>> info1 = iter: 100 time: 1.1300 M: 3.4258
>> wh1 = whext(l+1:end,:,:); % Remove the trailing part
>>
>> % Solution 2: output-only identification.
>>
```

```
>> % Identify an autonomous system
>> [sysh2,info2,yh2,xini2] = ident(y(2:end,:,:),0,l);
>> yh2 = [y(1,:,:); yh2];
>> info2 = iter: 100 time: 0.6300 M: 3.3443
>>
>> % Recover the I/O system
>> sysh2 = ss(sysh2.a,xini2,sysh2.c, reshape(yh2(1,:,:),p,m),-1);
>> wh2 = [u0 yh2];
>>
>> % Verify the results
>> w0 = [u0 y0];
>> err_data  = norm(w0(:)-w(:))/norm(w0(:))
err_data =     0.9219
>> err_appr1 = norm(w0(:)-wh1(:))/norm(w0(:))
err_appr1 =    0.2716
>> err_appr2 = norm(w0(:)-wh2(:))/norm(w0(:))
err_appr2 =    0.2608
```


Example: Finite time ℓ_2 model reduction

finite time T, ℓ_2 norm of $\mathscr{B}$

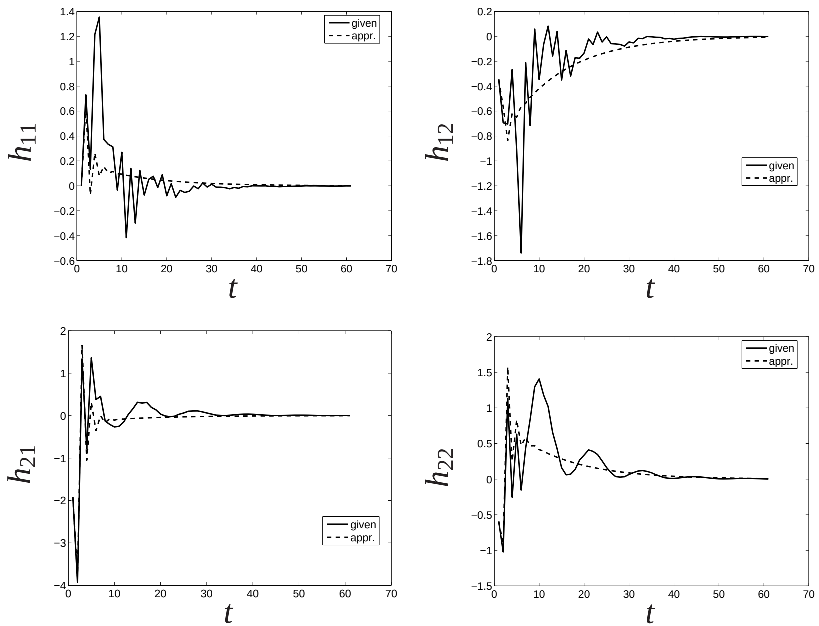
$$\|\mathscr{B}\|_{\ell_2,T} := \|h|_{[0,T]}\|_{\ell_2} = \sqrt{\textstyle\sum_{t=0}^{T} \|h(t)\|_{\mathrm{F}}^2}.$$

for a strictly stable system $\mathscr{B}$, $\|\mathscr{B}\|_{\ell_2,\infty} = \|\mathscr{B}\|_{\mathscr{H}_2}$

problem:

Given a system $\bar{\mathscr{B}} \in \mathscr{L}^q_{m,l}$, a natural number $l_{\text{red}} < l$, and a time horizon T, find a system $\hat{\mathscr{B}} \in \mathscr{L}^q_{m,l_{\text{red}}}$, that minimizes the finite time T, ℓ_2 norm $\|\bar{\mathscr{B}} - \hat{\mathscr{B}}\|_{\ell_2,T}$ of the error system.


```
>> l  = 10; m  = 2; p  = 2; lr = 1;  % lag of the reduced system
>>
>> % High order system
>> sys = drss_(p*l,p,m);
>>
>> % Simulate impulse response (determine automatically T)
>> h = impulse(sys); T = size(h,1);
>>
>> % Find reduced model
>> [sysr,info,hr,xini] = ident(h(2:end,:,:),0,lr);
>> hr = [h(1,:,:); hr];
>> sysr = ss(sysr.a,xini,sysr.c, reshape(hr(1,:,:),p,m),-1);
>> info = iter: 18 time: 0.1100 M: 3.5908
>>
>> % Relative Hinf and H2 norms of the error system
>> h2_err=norm(sys-sysr,2)/norm(sys,2)
h2_err = 0.5253
>> hi_err=norm(sys-sysr,'inf')/norm(sys,'inf')
hi_err = 0.5853
```

Example: The optimal misfit is generically independent of the input/output partitioning

for an arbitrary permutation matrix $\Pi \in \mathbb{R}^{q \times q}$

$$\min_{\mathscr{B} \in \mathscr{L}^q_{m,l}} M(w_\mathrm{d}, \mathscr{B}) = \min_{\mathscr{B} \in \mathscr{L}^q_{m,l}} M(\Pi w_\mathrm{d}, \mathscr{B})$$

```
>> % Identify the system from the original exact data
>> [sysh1,info1,wh1,xini1]=ident(w0,m,l);
>> info1 = iter: 1 time: 0.0200 M: 3.5617e-32
>>
>> % Identify the system from data with randomly permuted variables
>> perm = randperm(m+p)    =    3     4     1     2
>> w2   = w0(:,perm);
>> [sysh2,info2,wh2,xini2] = ident(w2,m,l);
>> info2 = iter: 1 time: 0 M: 0
>> % Compare the results
>> norm(wh1(:,perm) - wh2) =   2.0822e-32
```


Conclusions

- the STLS problem allows to treat a variety of problems
  - approximate realization
  - identification from step response data
  - autonomous system identification
  - model reduction
- the common framework is misfit minimization
- current research: relation to the classical prediction error methods