Software package for structured total least squares problems

Sabine Van Huffel
K.U.Leuven, ESAT/SCD (SISTA)

Joint work with
Ivan Markovsky

5th ERCIM workshop, 27–29 August 2004, Prague

Outline

- Motivation
- Solution method
- Implementation
- Six standard examples
- Application for system identification
- Conclusions

Structured total least squares

$$\min_{\Delta A, \Delta B, X} \| \begin{bmatrix} \Delta A & \Delta B \end{bmatrix} \|_{\mathrm{F}}^2 \quad \text{s.t.} \quad (A - \Delta A)X = B - \Delta B \quad \text{and}$$

$$\begin{bmatrix} \Delta A & \Delta B \end{bmatrix} \text{ has the same structure as } \begin{bmatrix} A & B \end{bmatrix}$$

a typical example how structure in the data matrix $\begin{bmatrix} A & B \end{bmatrix}$ occurs

$$R_0 w(t) + R_1 w(t+1) + \cdots + R_l w(t+l) = 0, \quad \text{for } t = 1, \ldots, T-l$$

is equivalent to the structured system of equations $R\mathcal{H}_{l+1}(w) = 0$, where $R := \begin{bmatrix} R_0 & R_1 & \cdots & R_l \end{bmatrix}$ and $\mathcal{H}_{l+1}(w)$ is the block-Hankel matrix

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

History of the problem

- [Aoki and Yue, 1970] — errors-in-variables identification $\rightsquigarrow$ STLS
- [Cadzow, 1988], [Bresler and Macovski, 1986] — suboptimal
- [Abatzoglou et al., 1991]—often cited as the original paper on STLS
- [De Moor, 1993] — Riemannian singular value decomposition
- [Rosen et al., 1996] — structured total least norm approach
- [Lemmerling et al., 2000, Mastronardi et al., 2000] — fast algorithms

Issues

- structure: varies from general affine to specific Hankel/Toeplitz
- rank reduction: for all methods, except [Van Huffel et al., 1996], one
- efficiency: varies from $O(m^3)$ to $O(m)$, where $m := \operatorname{row\,dim}(A)$

no efficient algorithms for multivariate problems

no robust software implementation

$\Longrightarrow$ up to now STLS was not used in real-life applications

Motivation

make the STLS method practically useful by:

1. algorithms that are general enough for various applications and efficient enough for non-toy examples
2. robust software implementation
3. practical problems where the STLS method and software give advantage over the alternative methods

Problem formulation

$$\min_{X,\ \Delta p} \|\Delta p\|_2^2 \quad \text{s.t.} \quad \mathcal{S}(p - \Delta p) \begin{bmatrix} X \\ -I_d \end{bmatrix} = 0$$

$X \in \mathbb{R}^{n \times d}$ — parameter, $p \in \mathbb{R}^{n_p}$ — data vector, Δp — correction

$$\mathcal{S}(p) = \begin{bmatrix} C^{(1)} & \cdots & C^{(q)} \end{bmatrix}, \text{ with } C^{(l)} \begin{cases} \texttt{T} & \text{block-Toeplitz,} \\ \texttt{H} & \text{block-Hankel,} \\ \texttt{U} & \text{unstructured, or} \\ \texttt{F} & \text{exact.} \end{cases}$$

all block-Toeplitz/Hankel $C^{(l)}$ have blocks of the same row dimension K and of column dimensions t_l

Structure specification

$\mathcal{S}$ is specified by the array $\mathcal{D} \in \{(\texttt{T}, \texttt{H}, \texttt{U}, \texttt{F}) \times \mathbb{N} \times \mathbb{N}\}^q$

$\mathcal{D}_l$ describes the block $C^{(l)}$ as follows:

- $\mathcal{D}_l(1)$ is the type of structure,
- $\mathcal{D}_l(2) = n_l := \operatorname{col\,dim}(C^{(l)})$, and
- $\mathcal{D}_l(3) = t_l$

example: $\mathcal{S}(p) = \begin{bmatrix} A & b \end{bmatrix}$, $A \in \mathbb{R}^{m \times n}$ Toeplitz, $b \in \mathbb{R}^m$ unstructured

$$\mathcal{D} = \begin{bmatrix} \begin{bmatrix} \texttt{T} & n & 1 \end{bmatrix} & \begin{bmatrix} \texttt{U} & 1 & 1 \end{bmatrix} \end{bmatrix}$$

Equivalent problem

$$\min_X r^\top(X)\Gamma^{-1}(X)r(X), \qquad r(X) := \text{vec}\Big(\big(\mathcal{S}(p)\left[\begin{smallmatrix} X \\ -I_d \end{smallmatrix}\right]\big)^\top\Big)$$

$\Gamma(X)$ has block-Toeplitz and block-banded structure,

$$\Gamma(X) = \begin{bmatrix} \Gamma_0 & \Gamma_{-1} & \cdots & \Gamma_{-s} & & 0 \\ \Gamma_1 & \ddots & \ddots & \ddots & \ddots & \\ \vdots & \ddots & \ddots & \ddots & \ddots & \Gamma_{-s} \\ \Gamma_s & \ddots & \ddots & \ddots & \ddots & \vdots \\ & \ddots & \ddots & \ddots & \ddots & \Gamma_{-1} \\ 0 & & \Gamma_s & \cdots & \Gamma_1 & \Gamma_0 \end{bmatrix} \in \mathbb{R}^{md \times md}$$

block size dK, $s = \max_{l=1,\dots,q} n_l/t_l$

this structure is exploited for efficient $O(m)$ computation of the cost function and its first derivative

Implementation

the software is written in ANSI C with calls to BLAS and LAPACK

we use the GNU scientific library (GSL) and a C version of MINPACK's Levenberg–Marquardt algorithm

the Cholesky decomposition of $\Gamma(X)$ is done via the subroutine `MB02GD` from the SLICOT library

the software is callable from MATLAB via a mex file interface:

```
>> [xh, info, v] = stls(a, b, s, x0, opt);
```

`a` $\leftrightarrow A$, `b` $\leftrightarrow B$, `s` $\leftrightarrow \mathcal{D}$, `xh` $\leftrightarrow \hat{X}$, `v` $\leftrightarrow \text{cov}\big(\text{vec}(\hat{X})\big)$

`x0`, `opt`, `info` — initial approximation, convergence options, and convergence information for the optimization algorithm

Simulation examples

illustrate application of the package on standard estimation problems

the problems listed below are special cases of the block-Toeplitz/Hankel STLS problem, for particular choices of the structure specification $K, \mathcal{D}$

if not given, K and the third element of $\mathcal{D}_l$ are by default equal to ones

all examples are performed in MATLAB 6.0, running on a Linux i686 PC

the MATLAB scripts are included in the package as a demo file `demo.m`

Least squares

$AX \approx B$, $A \in \mathbb{R}^{m\times n}$ exact unstr., $B \in \mathbb{R}^{m \times d}$ perturbed unstr.

STLS problem with structure specification $\mathcal{D} = [[\texttt{F}\ n], [\texttt{U}\ d]]$

```
>> % Define dimensions and generate random data
>> m = 100; n = 5; d = 2; a = rand(m,n); b = rand(m,d);
>> % Find the LS estimate by Matlab's \
>> tic, x_ls = a; t_ls = toc
t_ls =   0.00157700000000
>> disp(x_ls(1,1:d))
   0.05737144079627   0.22486444701677
>> % Define and solve the LS problem as an STLS problem
>> s_ls = [4 n 1; 3 d 1];
>> tic, x_stls = stls(a,b,s_ls); t_stls = toc
t_stls = 0.00730900000000
>> disp(x_stls(1,1:d))
   0.05737144079627   0.22486444701677
```

Total least squares

$AX \approx B$, $C := \begin{bmatrix} A & B \end{bmatrix} \in \mathbb{R}^{m\times(n+d)}$ perturbed and unstructured

STLS problem with structure specification $\mathcal{D} = [\texttt{U}\ n+d]$

```
>> % The data is a,b used above.
>> % Solve the TLS problem via SVD
>> tic, x_tls = tls(a,b); t_tls = toc
t_tls =    0.00375500000000
>> disp(x_tls(1,1:d))
  -0.49791037472338   0.03277992784515
>> % Define and solve the TLS problem as an STLS problem
>> s_tls = [3 n+d 1];
>> tic, i_stls = stls(a,b,s_tls); t_stls = toc
t_stls =  0.00504600000000
>> disp(x_stls(1,1:d))
  -0.49791037472338   0.03277992784515
```


Mixed least squares total least squares

$AX \approx B$, $A = [A_{\mathsf{f}}\ \ A_{\mathsf{p}}]$, $A_{\mathsf{p}} \in \mathbb{R}^{m\times n_1}$, $B \in \mathbb{R}^{m\times d}$ perturbed unstructured, and $A_{\mathsf{f}} \in \mathbb{R}^{m\times n_2}$ exact unstructured

STLS problem with structure specification $\mathcal{D} = [[\texttt{U}\ n_1], [\texttt{F}\ n_2], [\texttt{U}\ d]]$

`lstls.m` — MATLAB implementation of an (exact) SVD based method

```
>> n1 = 5; % The data is a,b used above.
>> % Solve the mixed LS-TLS problem via exact algorithm
>> tic, x_lstls = ls_tls(a(:,1:n1),a(:,n1+1:end),b); t_lstls = toc
t_lstls =  0.03171700000000
>> disp(x_lstls(1,1:d))   0.05737144079627   0.22486444701677
>> % Define and solve the mixed LS-TLS problem as an STLS problem
>> s_lstls = [4 n1 1; 3 n+d-n1 1];
>> tic, [x_stls,i_stls] = stls(a,b,s_lstls); t_stls = toc
t_stls =    0.00724800000000
>> disp(x_stls(1,1:d))   0.05737144079627   0.22486444701677
```


Hankel low-rank approximation

$$\min_{\Delta p} \|\Delta p\|_2^2 \quad \text{s.t.} \quad \mathcal{H}(p - \Delta p) \text{ has given rank } n.$$

$\mathcal{H} : \mathbb{R}^{n_p} \to \{\, m \times (n+d)$ block-Hankel matrices, block size $n_y \times n_u$

STLS problem with $K = n_y$ and $\mathcal{D} = [\texttt{H}\ n+d\ n_u]$

the case $n_y = n_u = 1$ is treated in [Lemmerling et al., 2000]

`faststln2` — MATLAB implementation

```
>> % Generate data
>> np = 12; p0 = (1:np)'; p  = p0 + [5; zeros(np-1,1)];
>> c = hankel(p(1:10),p(10:np)); a = c(:,1:2); b = c(:,3);
>> % Define the structure and solve the problem via STLS
>> s  = [2 3 1];
>> tic, [xh_stls,i_stls] = stls(a,b,s); t_stls = toc
t_stls =    0.00395000000000
```



```
>> disp(xh_stls(1:2)')   0.30331872971326   0.87809000348994
>> disp(i_stls.fmin) % value of the cost function at xh_stls
   2.88924164814028
>> % Solve via an alternative STLS method
>> ct = fliplr(c); % ct is Toeplitz structured
>> tic, xh_stln = faststln2(c(:,1:2),c(:,3)); t_stln = toc
t_stln =    0.19313500000000
>> % recover the solution of the Hankel structured problem
>> x_ext = [xh_stln; -1]; x_ext = flipud(x_ext);
>> xh_stln = -x_ext(1:2)/x_ext(3);
>> disp(xh_stln(1:2)')   0.30320645782842   0.87819047149399
>> disp(cost(xh_stln,a,b,s)) % value of the cost function at xh_stln
   2.88924181032173
```

Deconvolution problem $b_i = \sum_{j=-\infty}^{\infty} x_j a_{i-j}$

if $x_j = 0$ for all $j < 1$ and $j > n$, then b_i, $i = 1, \ldots, m$ is given by

$$\underbrace{\begin{bmatrix} a_0 & a_{-1} & \cdots & a_{1-n} \\ a_1 & a_0 & \cdots & a_{2-n} \\ \vdots & \vdots & & \vdots \\ a_{m-1} & a_{m+n-2} & \cdots & a_{m-n} \end{bmatrix}}_{A} \underbrace{\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}}_{x} = \underbrace{\begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{bmatrix}}_{b}$$

A Toeplitz structured, parameterized by $a = \mathrm{col}(a_{1-n}, \ldots, a_{m-1})$

deconvolution problem: find x, given a and b

with a, b perturbed, an STLS problem with $\mathcal{D} = [[\mathtt{T}\ n], [\mathtt{U}\ 1]]$

this STLS problem is studied in [Mastronardi et al., 2000]

`faststln1` — MATLAB implementation


```
>> m = 200;  n = 2; % m = length(x), n = length(b)
>> % Generate true data: a0, b0, and x0
>> a0 = rand(n+m-1); A0 = toeplitz(a0(n:n+m-1),a0(n:-1:1));
>> x0 = rand(n,1); b0 = A0*x0;
>> % Add noise: a = a0 + noise, b = b0 + noise
>> a = a0 + 0.25 * randn(n+m-1); b = b0 + 0.25 * randn(m,1);
>> A = toeplitz(a(n:n+m-1),a(n:-1:1));
>> % Define the structure and solve the problem via STLS
>> s = [1 n 1; 3 1 1]; tic, [xh_stls,i_stls] = stls(A,b,s); t_stls = toc
t_stls =   0.08655400000000
>> disp(xh_stls(1:2)')   0.26594446871296   0.31420369470136
>> disp(i_stls.fmin) % value of the cost function at xh_stls
  15.23654290180259
>> % Solve via an alternative STLS method
>> tic, xh_stln = faststln1(A,b); t_stln = toc
t_stls =  16.09171600000000
>> disp(xh_stln(1:2)')   0.26594792311858   0.31420021871824
>> disp(cost1(xh_stln,a,b,s)) % value of the cost function at xh_stln
  15.23654290217436
```


Transfer function estimation

consider a dynamical system described by the difference equation

$$y_t + \sum_{\tau=1}^{n} a_\tau y_{t+\tau} = \sum_{\tau=0}^{n} b_\tau u_{t+\tau}$$

parameter vector $x := \mathrm{col}(b_0, \ldots, b_n, -a_0, \ldots, -a_{n-1}) \in \mathbb{R}^{2n+1}$

problem: find x, given $(u_t, y_t)_{t=1}^{T}$ and the order n

for $t = 1, \ldots, T$, () can be written as

$$\left[\begin{array}{cccc|cccc} u_1 & u_2 & \cdots & u_{n+1} & y_1 & y_2 & \cdots & y_n \\ u_2 & u_3 & \cdots & u_{n+2} & y_2 & y_3 & \cdots & y_{n+1} \\ \vdots & \vdots & & \vdots & \vdots & \vdots & & \vdots \\ u_m & u_{m+1} & \cdots & u_T & y_m & y_{m+1} & \cdots & y_{T-1} \end{array}\right] x = \begin{bmatrix} y_{n+1} \\ y_{n+2} \\ \vdots \\ y_T \end{bmatrix}$$

STLS problem with structure specification $\mathcal{D} = [[\mathtt{H}\ n+1], [\mathtt{H}\ n+1]]$


```
>> n   = 3; num = 0.151*[1 0.9 0.49 0.145]; den = [1 -1.2 0.81 -0.27];
>> % True data
>> T   = 1000; u0 = randn(T,1); [y0,x0] = dlsim(num,den,u0);
>> % Noisy data
>> v_n = .1; y = y0 + v_n * randn(T,1); u = u0 + v_n * randn(T,1);
>> % Define the system of equations
>> m = length(y) - n;
>> a = [ hankel(u(1:m),u(m:end)) hankel(y(1:m),y(m:end)) ];
>> b = a(:,end); a(:,end) = [];
>> % Ignore the structure and solve the problem via LS and TLS
>> tic, xh_ls  = a; t_ls = toc
t_ls =   0.00329800000000
>> tic, xh_tls = tls(a,b); t_tls = toc
t_tls =  0.00638900000001
>> % Define the structure and solve the problem via STLS
>> s = [2 n+1 1; 2 n+1 1];
>> tic, [xh_stls,i_stls] = stls(a,b,s); t_stls = toc
t_stls = 1.27913800000000
>> % Extract the estimates
```

```
>> num_ls = fliplr(xh_ls(1:n+1)');
>> den_ls = [1 fliplr(-xh_ls(n+2:end)')];
>> num_tls = fliplr(xh_tls(1:n+1)');
>> den_tls = [1 fliplr(-xh_tls(n+2:end)')];
>> num_stls = fliplr(xh_stls(1:n+1)');
>> den_stls = [1 fliplr(-xh_stls(n+2:end)')];
>> % Compare the relative errors of estimation
>> e_ls = norm([num-num_ls,den-den_ls]) / norm([num,den]);
   0.74534028390667
>> e_tls = norm([num-num_tls,den-den_tls]) / norm([num,den]);
   0.02825246589733
>> e_stls = norm([num-num_stls,den-den_stls]) / norm([num,den]);
   0.02381490382674
```

the example shows an application of STLS for system identification

the method is extended for multivariable systems

next we shows simulation results with more realistic problems

Results on data sets from DAISY

DAISY — data base for system identification, available from

http://www.esat.kuleuven.ac.be/~tokka/daisydata.html

real-life and simulated data for verification and comparison of ident. alg.

the estimates obtained by the following methods are compared:

- `subid` — robust combined subspace algorithm
- `detss` — deterministic balanced subspace algorithm
- `pem` — the prediction error method of the Identification Toolbox
- `stls` — an identification method based on STLS

comparison is in the misfit $M(w, \hat{\mathcal{B}})$ scaled by $M(w, \hat{\mathcal{B}}_{\mathtt{stls}})$

#	Data set name	parameters T	m	p	l	scaled misfit `subid`	`detss`	`pem`
1	Destillation column	90	5	3	1	2.8	9.6	15.9
2	Destillation column n10	90	5	3	1	2.8	9.6	15.9
3	Destillation column n20	90	5	3	1	8.3	2.3	36.1
4	Destillation column n30	90	5	3	1	7.8	3.3	132
5	Glass furnace (Philips)	1247	3	6	1	2.9	2.5	2.7
6	120 MW power plant	200	5	3	2	7.2	3.4	28
7	pH process	2001	2	1	6	1.3	1.3	3.0
8	Hair dryer	1000	1	1	5	1.2	1.2	1.0
9	Winding process	2500	5	2	2	1.5	1.4	2.8
10	Ball-and-beam setup	1000	1	1	2	1.0	10	1.0
11	Industrial dryer	867	3	3	1	1.2	1.1	1.1

#	Data set name	parameters T	m	p	l	scaled misfit `subid`	`detss`	`pem`
12	CD-player arm	2048	2	2	1	1.2	1.1	1.4
13	Wing flutter	1024	1	1	5	1.6	1.7	2.8
14	Robot arm	1024	1	1	4	2.7	18.7	26.0
15	Lake Erie	57	5	2	1	1.5	2.3	23.1
16	Lake Erie n10	57	5	2	1	2.1	2.2	8.4
17	Lake Erie n20	57	5	2	1	2.2	2.4	9.8
18	Lake Erie n30	57	5	2	1	2.4	1.6	5.6
19	Heat flow density	1680	2	1	2	1.8	1.3	9.8
20	Heating system	801	1	1	2	1.3	1.2	1.3
21	Steam heat exchanger	4000	1	1	2	1.8	1.8	8.1
22	Industrial evaporator	6305	3	3	1	1.5	1.1	1.6
23	Tank reactor	7500	1	2	1	2.3	2.1	52.9
24	Steam generator	9600	4	4	1	2.4	3.1	3.3

comparison in the execution time scaled by $M(w, \hat{\mathcal{B}}_{\texttt{subid}})$

#	Data set name	parameters T	m	p	l	scaled exec. time detss	stls	pem
1	Destillation column	90	5	3	1	3.3	6.4	11.1
2	Destillation column n10	90	5	3	1	7.3	12.5	23.1
3	Destillation column n20	90	5	3	1	7.2	12.8	7.2
4	Destillation column n30	90	5	3	1	7.0	12.1	7.2
5	Glass furnace (Philips)	1247	3	6	1	13	361	373
6	120 MW power plant	200	5	3	2	6.3	15.5	27.3
7	pH process	2001	2	1	6	2.9	7.4	32.3
8	Hair dryer	1000	1	1	5	1.5	5.8	36.4
9	Winding process	2500	5	2	2	4.4	37.1	74.8
10	Ball-and-beam setup	1000	1	1	2	1.9	4.1	7.2
11	Industrial dryer	867	3	3	1	6.6	25.5	27.3

#	Data set name	parameters T	m	p	l	scaled exec. time subid	stls	pem
12	CD-player arm	2048	2	2	1	6.4	19.5	49.4
13	Wing flutter	1024	1	1	5	1.7	4.7	33.5
14	Robot arm	1024	1	1	4	1.8	3.8	30.7
15	Lake Erie	57	5	2	1	1.4	4.6	7.0
16	Lake Erie n10	57	5	2	1	1.4	4.6	11.4
17	Lake Erie n20	57	5	2	1	1.6	4.8	9.1
18	Lake Erie n30	57	5	2	1	1.7	4.8	7.0
19	Heat flow density	1680	2	1	2	2.6	6.3	39.7
20	Heating system	801	1	1	2	1.7	3.7	12.4
21	Steam heat exchanger	4000	1	1	2	4.3	8.4	31.1
22	Industrial evaporator	6305	3	3	1	10.5	59.9	134.4
23	Tank reactor	7500	1	2	1	11.0	25.2	146.0
24	Steam generator	9600	4	4	1	13.6	192.0	220.1

Conclusions

- we reviewed the origin and development of the STLS problem
- flexible formulation was proposed that covers various applications *e.g.*, Hankel low rank approx., deconvolution, transfer function estim.
- an efficient computational method for the new formulation was described
- a software implementation of the new method is developed
- the STLS problem is used for MIMO system identification
- simulation results with real and simulated data sets show that the STLS software is fast and reliable

References

[Abatzoglou et al., 1991] Abatzoglou, T., Mendel, J., and Harada, G. (1991). The constrained total least squares technique and its application to harmonic superresolution. *IEEE Trans. on Signal Proc.*, 39:1070–1087.

[Aoki and Yue, 1970] Aoki, M. and Yue, P. C. (1970). On a priori error estimates of some identification methods. *IEEE Trans. on Aut. Control*, 15(5):541–548.

[Bresler and Macovski, 1986] Bresler, Y. and Macovski, A. (1986). Exact maximum liklihood parameter estimation of superimposed exponential signals in noise. *IEEE Trans. on Acoustics, speech, and Signal Proc.*, 34:1081–1089.

[Cadzow, 1988] Cadzow, J. (1988). Signal enhancement—a composite property mapping algorithm. *IEEE Trans. on Signal Proc.*, 36:49–62.

[De Moor, 1993] De Moor, B. (1993). Structured total least squares and L_2 approximation problems. *Lin. Alg. and Its Appl.*, 188–189:163–207.

[Lemmerling et al., 2000] Lemmerling, P., Mastronardi, N., and Huffel, S. V. (2000). Fast algorithm for solving the Hankel/Toeplitz structured total least squares problem. *Numerical Algorithms*, 23:371–392.

[Mastronardi et al., 2000] Mastronardi, N., Lemmerling, P., and Huffel, S. V. (2000). Fast structured total least squares algorithm for solving the basic deconvolution problem. *SIAM J. Matrix Anal.*, 22:533–553.

[Rosen et al., 1996] Rosen, J. B., Park, H., and Glick, J. (1996). Total least norm formulation and solution of structured problems. *SIAM J. Matrix Anal.*, 17:110–128.

[Van Huffel et al., 1996] Van Huffel, S., Park, H., and Rosen, J. B. (1996). Formulation and solution of structured total least norm problems for parameter estimation. *IEEE Trans. on Signal Proc.*, 44(10):2464–2474.