Approximate system identification

Ivan Markovsky

University of Southampton

Outline

- From exact to approximate identification
- Misfit vs latency
- Misfit minimization
- Misfit computation
  - kernel
  - image
  - input/state/output

From exact to approximate identification

Exact system identification and the MPUM

Exact identification problem:

Given a vector time series

$$w_\mathrm{d} = \big(w_\mathrm{d}(1), \ldots, w_\mathrm{d}(T)\big) \in (\mathbb{R}^{\mathtt{w}})^T$$

find the smallest $\mathtt{m} \in \mathbb{N}$ and $\ell \in \mathbb{N}$ and LTI system $\mathscr{B}_\mathrm{mpum} \in \mathscr{L}^{\mathtt{w}}_{\mathtt{m},\ell}$, s.t.

$$w_\mathrm{d} \in \mathscr{B}_\mathrm{mpum}.$$

The model $\mathscr{B}_\mathrm{mpum} = \mathscr{B}_\mathrm{mpum}(w_\mathrm{d})$ is unique and is called the MPUM of w_d in the model class $\mathscr{L}^{\mathtt{w}}_{\mathtt{m},\ell}$.

There are effective algorithms for computing representations of $\mathscr{B}_\mathrm{mpum}(w_\mathrm{d})$ from given $w_\mathrm{d} = (u_\mathrm{d}, y_\mathrm{d})$ and an upper bound ℓ_max on ℓ.

Identifiability

Provided that $w_{\mathrm{d}} \in \overline{\mathscr{B}} \in \mathscr{L}^{\mathtt{w}}_{\mathtt{m},\ell_{\max}}$, find conditions under which

$$\mathscr{B}_{\mathrm{mpum}}(w_{\mathrm{d}}) = \overline{\mathscr{B}}.$$

Main theoretical result in the exact identification setting:

$\overline{\mathscr{B}}$ is identifiable from $w_{\mathrm{d}} = (u_{\mathrm{d}}, y_{\mathrm{d}})$ in $\mathscr{L}^{\mathtt{w}}_{\mathtt{m},\ell_{\max}}$ if

1. w_{d} is exact, *i.e.*, $w_{\mathrm{d}} \in \overline{\mathscr{B}}$,
2. the model class is correct, *i.e.*, $\overline{\mathscr{B}} \in \mathscr{L}^{\mathtt{w}}_{\mathtt{m},\ell_{\max}}$,
3. u_{d} is persistently exciting of order $\ell_{\max} + \mathtt{n}$, and
4. $\overline{\mathscr{B}}$ is controllable

Notes

- the conditions are only sufficient
- conditions 1, 2, and 4 are not verifiable from the given data and should therefore be postulated
- a known input/output partitioning of the data is assumed
- from a practical point of view, condition 1 is a strong assumption that limits the applicability of the exact SYSID methods

approaches for relaxing condition 1 are described next

MPUM in the case of "noisy" data

Q: What is the MPUM of a noisy trajectory

$$w_{\rm d} = \overline{w} + \widetilde{w} \qquad \text{where} \quad \overline{w} \in \overline{\mathscr{B}} \in \mathscr{L}^{\tt w}_{{\tt m},\ell_{\max}} \qquad \text{(EIV)}$$
$$\text{and} \quad \widetilde{w} \text{ is random and zero mean?}$$

A: With probability 1,

$$\mathscr{B}_{\rm mpum}(w_{\rm d}) = (\mathbb{R}^{\tt w})^{\mathbb{Z}_+} \qquad \text{(all variables are inputs)}$$

This is a trivial model because it fits every trajectory.

Alternatively, $\mathscr{B}_{\rm mpum}(w_{\rm d})$ does not exist in a model class $\mathscr{M} = \mathscr{L}^{\tt w}_{{\tt m},\ell_{\max}}$ of bounded (${\tt m} < {\tt w}$, $\ell_{\max} \ll T$) complexity.

In what follows we assume a given bounded complexity model class $\mathscr{M} = \mathscr{L}^{\tt w}_{{\tt m},\ell_{\max}}$, so we refer to lack of existence rather than trivial MPUM.

In practice, $w_{\rm d}$ is often generated by a

nonlinear, infinite dimensional, time-varying system $\overline{\mathscr{B}}$

possibly with process and measurement noises, *i.e.*, $\overline{\mathscr{B}} \notin \mathscr{L}^{\tt w}_{\tt m,\ell_{max}}$

$\Longrightarrow$ even without noise, identifying exact model is often not possible

It can be argued that in practice the approximation aspect ($\widehat{\mathscr{B}} \approx \overline{\mathscr{B}}$) is often more important than the stochastic estimation ($\widehat{\mathscr{B}} \to \overline{\mathscr{B}}$ as $T \to \infty$)

An approximate $\widehat{\mathscr{B}} \in \mathscr{L}^{\tt w}_{\tt m,\ell_{max}}$ is what is anyway needed:

Many prediction and control methods are based on LTI models

$\Longrightarrow$ even if it was possible to identify $\overline{\mathscr{B}}$, it would be necessary to approximate it by $\widehat{\mathscr{B}} \in \mathscr{L}^{\tt w}_{\tt m,\ell_{max}}$

Misfit vs latency

Modifications of the MPUM concept

Unless the model class $\mathscr{M} = \mathscr{L}^{\mathtt{w}}_{\mathtt{m},\ell_{\max}}$ is enlarged, *i.e.*, $(\mathtt{m}, \ell_{\max})$ is increased, until the MPUM exists in $\mathscr{M}$

we have to accept falsified models in $\mathscr{M}$
$\rightsquigarrow$ approximate SYSID

The main question in approximate SYSID is:

Q: Which (falsified) model in $\mathscr{M}$ to choose?

A: In some sense the "least falsified" ("approximately unfalsified") one.

Two major notions of "least falsified" are small misfit and small latency.

They quantify the discrepancy between the model and the data.

Misfit approach for approximate SYSID

Consider given data w_{d} and model class $\mathscr{M} = \mathscr{L}^{\mathtt{w}}_{\mathtt{m},\ell_{\max}}$.

If the MPUM does not exists in $\mathscr{M}$, *i.e.*,

$$\mathscr{B}_{\text{mpum}}(w_{\text{d}}) \notin \mathscr{M}$$

we aim to find an approximate model $\widehat{\mathscr{B}}$ for w_{d} in $\mathscr{M}$.

The misfit approach modifies w_{d} as little as possible, so that the modified data, say $\widehat{w}$, has MPUM in $\mathscr{M}$, *i.e.*,

$$\mathscr{B}_{\text{mpum}}(\widehat{w}) \in \mathscr{M}$$

The approximate model for w_{d} in $\mathscr{M}$ is defined as $\widehat{\mathscr{B}}_{\text{misfit}} := \mathscr{B}_{\text{mpum}}(\widehat{w})$.

The modification of the data is measured by the misfit $\|w_{\text{d}} - \widehat{w}\|$

Latency

The latency approach augments w_{d} by, as small as possible, variable e, so that the augmented data $w_{\text{ext}} := \mathrm{col}(e, w_{\mathrm{d}})$ has MPUM in the augmented model class, *i.e.*,

$$\mathscr{B}_{\text{mpum}}\big(\mathrm{col}(e, w_{\mathrm{d}})\big) \in \mathscr{M}_{\text{ext}} := \mathscr{L}^{\mathtt{w}+\mathtt{e}}_{\mathtt{m}+\mathtt{e},\ell_{\max}}$$

Let Π_w be the projection of $\mathrm{col}(e, w)$ on w. The approximate model for w_{d} in $\mathscr{M}$ is defined as

$$\widehat{\mathscr{B}}_{\text{latency}} := \Pi_w \, \mathscr{B}_{\text{mpum}}\big(\mathrm{col}(e, w_{\mathrm{d}})\big)$$

The size of e is measured by the latency $\|e\|$

Notes

- Both the misfit and latency approaches reduce the approximate SYSID problem to (different) exact SYSID problems:

  - $\widehat{\mathscr{B}}_{\text{misfit}}$ is exact for the modified data $\widehat{w}$
  - $\widehat{\mathscr{B}}_{\text{latency}}$ is obtained from an exact model for $\text{col}(e, w_{\text{d}})$

- If $\mathscr{B}_{\text{mpum}}(w_{\text{d}}) \in \mathscr{M}$ (the data is exact),

$$\widehat{\mathscr{B}}_{\text{misfit}} = \widehat{\mathscr{B}}_{\text{latency}} = \mathscr{B}_{\text{mpum}}(w_{\text{d}}) \qquad (\widehat{w} = w_{\text{d}} \text{ and } e = 0)$$

  So, misfit and latency are indeed extensions of the MPUM.

- The misfit approach modifies w_{d} but does not change $\mathscr{M}$.

- The latency approach modifies $\mathscr{M}$ but does not change w_{d}.

Static case: latency $\leftrightarrow$ LS misfit $\leftrightarrow$ TLS

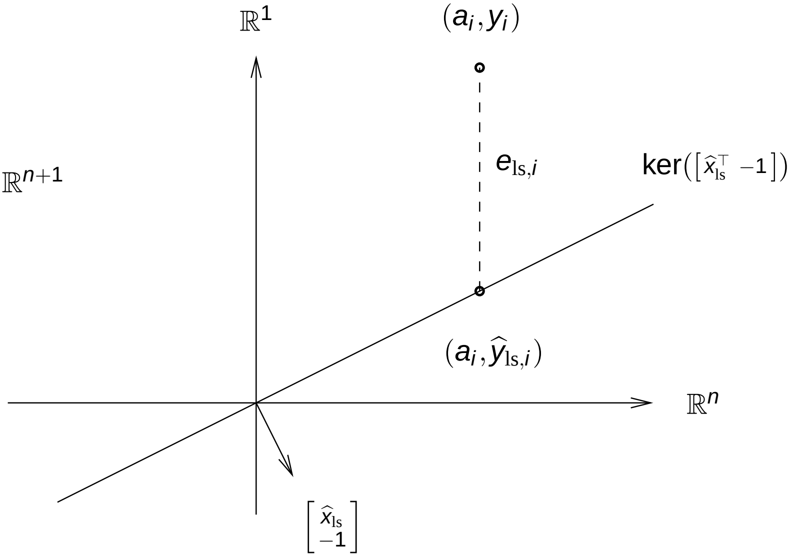
LS: minimize$_{e,x}$ $\|e\|_2$ subject to $Ax = y + e$ e latent variable

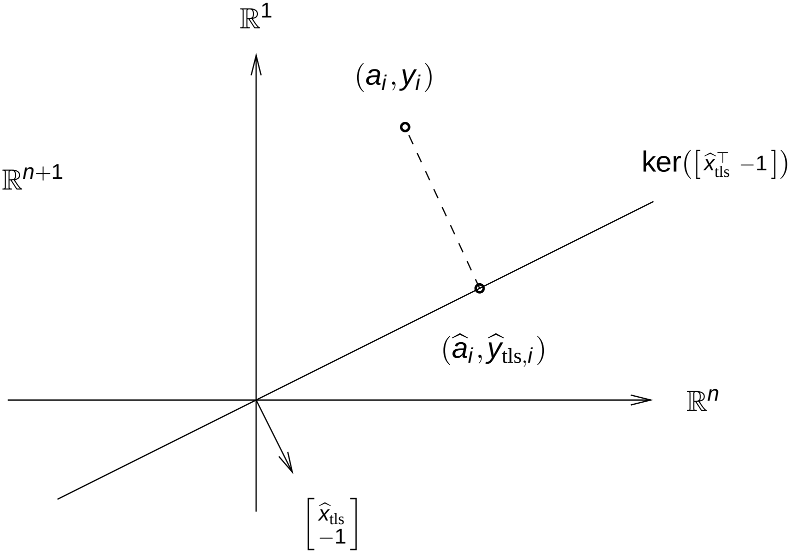
$$\text{latency}\big((A,y),x\big) := \Big(\min_e \|e\|_2^2 \text{ s.t. } Ax = y + e\Big) = \|Ax - y\|_2^2$$

TLS: minimize$_{\Delta A, \Delta y, x}$ $\big\| \begin{bmatrix} \Delta A & \Delta y \end{bmatrix} \big\|_{\mathrm{F}}$
subject to $(A + \Delta A)x = y + \Delta y$ ΔA, Δy data corrections

$$\text{misfit}\big((A,b),x\big) := \min_{\Delta A, \Delta b} \big\| \begin{bmatrix} \Delta A & \Delta b \end{bmatrix} \big\|_{\mathrm{F}} \text{ s.t. } (A + \Delta A)x = b + \Delta b$$

$$= \frac{\|Ax - b\|_2}{\sqrt{1 + \|x\|_2^2}}$$

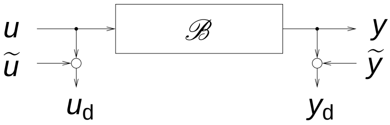

$\mathbb{R}^1$
(a_i, y_i)
$\mathbb{R}^{n+1}$
$e_{\mathrm{ls},i}$
$\ker([\widehat{x}_{\mathrm{ls}}^\top \ -1])$
$(a_i, \widehat{y}_{\mathrm{ls},i})$
$\mathbb{R}^n$
$\begin{bmatrix} \widehat{x}_{\mathrm{ls}} \\ -1 \end{bmatrix}$


$\mathbb{R}^1$
(a_i, y_i)
$\mathbb{R}^{n+1}$
$\ker\left(\begin{bmatrix}\widehat{x}_{\text{tls}}^\top & -1\end{bmatrix}\right)$
$(\widehat{a}_i, \widehat{y}_{\text{tls},i})$
$\mathbb{R}^n$
$\begin{bmatrix}\widehat{x}_{\text{tls}} \\ -1\end{bmatrix}$

Statistical interpretation of misfit and latency

misfit	$\leftrightarrow$	errors-in-variables (EIV) model
latency	$\leftrightarrow$	ARMAX model

EIV model: $\widetilde{w} = (\widetilde{u}, \widetilde{y})$ — measurement errors



ARMAX model: e — process noise

Assumptions: $\widetilde{w}$, e — zero mean, stationary, white, ergodic, Gaussian

- $\Pi_w \mathscr{B}_{\text{ext}}$ — deterministic part of the model
- $\Pi_e \mathscr{B}_{\text{ext}}$ — stochastic part of the model

Notes:

- The Kalman filter and LQG control are based on the latency model
- $\implies$ stochastic part is used by the KF and LQG controller

Misfit minimization

Identification problems

Misfit minimization (GTLS): given $w_\mathrm{d} \in (\mathbb{R}^\mathtt{w})^T$ and $\ell_\mathrm{max} \in \mathbb{N}$, find

$$\widehat{\mathscr{B}}^*_\mathrm{gtls} := \arg\min_{\widehat{\mathscr{B}},\widehat{w}} \quad \|w_\mathrm{d} - \widehat{w}\| \quad \text{subject to} \quad \widehat{w} \in \widehat{\mathscr{B}} \in \mathscr{L}_{\mathtt{m},\ell_\mathrm{max}}$$

Latency minimization (PEM): given $w_\mathrm{d} \in (\mathbb{R}^\mathtt{w})^T$ and $\ell_\mathrm{max} \in \mathbb{N}$, find

$$\widehat{\mathscr{B}}^*_\mathrm{pem} := \arg\min_{\widehat{\mathscr{B}}_\mathrm{ext},e} \quad \|e\| \quad \text{subject to} \quad (e,\widehat{w}) \in \widehat{\mathscr{B}}_\mathrm{ext} \in \mathscr{L}_{\mathtt{m}+\mathtt{e},\ell_\mathrm{max}}$$

Notes:

- nonconvex optimization problems
- solution methods based on local optimization
- initial approximation obtained from subspace methods

Misfit minimization

Define the misfit between w_d and $\mathscr{B}$ as follows

$$\mathrm{misfit}(w_\mathrm{d}, \mathscr{B}) := \text{minimize}_{\widehat{w}} \quad \|w_\mathrm{d} - \widehat{w}\|_{\ell_2} \quad \text{subject to} \quad \widehat{w} \in \mathscr{B}$$

the minimizer $\widehat{w}^*$ is projection of w_d on $\mathscr{B}$ (best ℓ_2 approx. of w_d in $\mathscr{B}$)

alternatively, $\widehat{w}^*$ is the smoothed estimate of w_d, given $\mathscr{B}$

our goal is to find the model $\widehat{\mathscr{B}}$ that minimizes $\mathrm{misfit}(w_\mathrm{d}, \mathscr{B})$, *i.e.*,

$$\widehat{\mathscr{B}} := \arg\min_{\widehat{w}} \ \mathrm{misfit}(w_\mathrm{d}, \mathscr{B}) \quad \text{subject to} \quad \mathscr{B} \in \mathscr{M}$$

a double minimization problem: inner minimization is projection on a subspace (easy), outer minimization is a nonconvex problem (difficult)

Maximum likelihood estimator in the EIV setup

Assuming that the data is generated according to the model

$$w_\text{d} = \overline{w} + \widetilde{w}, \qquad \text{where} \quad \overline{w} \in \overline{\mathscr{B}} \in \mathscr{M} \quad \text{and} \quad \widetilde{w} \sim \mathrm{N}(0, s^2 I)$$

$\widehat{\mathscr{B}}$ is the maximum likelihood estimator of the true model $\overline{\mathscr{B}}$

$\widehat{\mathscr{B}}$ is a consistent estimator of the true model $\overline{\mathscr{B}}$, *i.e.*, $\widehat{\mathscr{B}} \to \overline{\mathscr{B}}$ as $T \to \infty$

The log-likelihood function is (“const” does not depend on $\widehat{w}$ and $\widehat{\mathscr{B}}$)

$$\ell(\widehat{\mathscr{B}}, \widehat{w}) = \begin{cases} \text{const} - \frac{1}{2s^2} \|w_\text{d} - \widehat{w}\|_{\ell_2}^2, & \text{if } \widehat{w} \in \widehat{\mathscr{B}} \in \mathscr{M} \\ -\infty, & \text{otherwise}, \end{cases}$$

likelihood evaluation $\iff$ misfit computation

Misfit computation

Computation of the misfit

Given w_{d} and $\mathscr{B} \in \mathscr{L}^{\mathtt{w}}_{\mathtt{m},\ell_{\max}}$, find $\operatorname{misfit}(w_{\mathrm{d}}, \mathscr{B}) := \min_{\widehat{w} \in \mathscr{B}} \|w_{\mathrm{d}} - \widehat{w}\|_{\ell_2}$

$$\begin{aligned} \mathscr{B} \text{ is subspace} \quad &\Longrightarrow \quad \text{the constraint is linear} \\ &\Longrightarrow \quad \text{ordinary LS problem} \end{aligned}$$

Using general purpose LS solvers, the comput. complexity is $O(T^3)$.

Time-invariance of $\mathscr{B}$, however, implies Toeplitz structure of the LS prob.

Structure exploiting misfit computation methods have complexity $O(T)$.

They are based on:

1. structured matrix computations
   (*e.g.*, using displacement rank theory and the gen. Schur alg.)
2. Riccati recursions (Kalman smoother)

Misfit computation using image representation

$$\text{minimize}_{\widehat{w}} \quad \|w_\text{d} - \widehat{w}\|_{\ell_2} \quad \text{subject to} \quad \widehat{w} \in \mathscr{B} := \text{image}\big(M(\sigma)\big) \qquad \text{(M)}$$

Recall from Lecture 5 that

$$w \in \mathscr{B} \iff w = \underbrace{\begin{bmatrix} M_0 & M_1 & \cdots & M_\ell & & & \\ & M_0 & M_1 & \cdots & M_\ell & & \\ & & \ddots & \ddots & & \ddots & \\ & & & M_0 & M_1 & \cdots & M_\ell \end{bmatrix}}_{\mathscr{T}_M} \begin{bmatrix} v(1) \\ v(2) \\ \vdots \\ v(T+\ell) \end{bmatrix}$$

So that (M) is an ordinary least squares problem

$$\text{minimize}_{v} \quad \|w_\text{d} - \mathscr{T}_M v\| \qquad \text{(M')}$$

and

$$\text{misfit}\big(w_\text{d}, \text{image}\,(M(\sigma))\big) = w_\text{d}^\top \mathscr{T}_M \big(\mathscr{T}_M^\top \mathscr{T}_M\big)^{-1} \mathscr{T}_M^\top w_\text{d}$$

Misfit computation using kernel representation

$$\text{minimize}_{\widehat{w}} \quad \|w_\text{d} - \widehat{w}\|_{\ell_2} \quad \text{subject to} \quad \widehat{w} \in \mathscr{B} := \ker\big(R(\sigma)\big) \qquad \text{(R)}$$

Recall from Lecture 5 that

$$w_\text{d} \in \mathscr{B} \iff \underbrace{\begin{bmatrix} R_0 & R_1 & \cdots & R_\ell & & & \\ & R_0 & R_1 & \cdots & R_\ell & & \\ & & \ddots & \ddots & & \ddots & \\ & & & R_0 & R_1 & \cdots & R_\ell \end{bmatrix}}_{\mathscr{T}_T(R)} \begin{bmatrix} w_\text{d}(1) \\ w_\text{d}(2) \\ \vdots \\ w_\text{d}(T) \end{bmatrix} = 0$$

So that (R) is an equality constrained least squares problem

$$\text{minimize}_{\widehat{w}} \quad \|w_\text{d} - \widehat{w}\|_{\ell_2} \quad \text{subject to} \quad \mathscr{T}_R \widehat{w} = 0 \qquad \text{(R')}$$

In order to solve (R') explicitly, we need a basis for $\text{null}\left(\mathscr{T}_R\right)$. Let

$$N \text{ be such that} \quad \mathscr{T}_R N = 0 \quad \text{and} \quad N \text{ is full column rank}$$

Then, $\mathscr{T}_R \widehat{w} = 0 \iff \exists\, z \in \mathbb{R}^{\text{coldim}(N)}$ s.t. $\widehat{w} = Nz$, and

$$\text{misfit}\left(w_{\text{d}}, \ker\left(R(\sigma)\right)\right) = w_{\text{d}}{}^{\top} N \left(N^{\top} N\right)^{-1} N^{\top} w_{\text{d}}.$$

Note that the columns of $\mathscr{T}_M$ form a particular basis for the null space of $\mathscr{T}_R$. This can be seen algebraically from

$$\mathscr{T}_R \mathscr{T}_M = 0 \qquad \text{and} \qquad \mathscr{T}_M \text{ is full column rank}$$

or (better) from a system theoretic point of view:

$$\text{colspan}\left(\mathscr{T}_R\right) = \text{null}\left(\mathscr{T}_R\right) = \mathscr{B}_T.$$

Efficient reduction of (R) to (M)

Computing N, reduces (R) to (M), however, N need not be Toeplitz.

Moreover, computing N by general purpose methods is expensive, while the path $R \mapsto M \mapsto \mathscr{T}_M$ is cheap.

Let $R(z) =: \begin{bmatrix} Q(z) & P(z) \end{bmatrix}$ with $P(z) \in \mathbb{R}^{\mathrm{p} \times \mathrm{p}}[z]$ nonsingular
(this is equivalent to assuming existence of I/O partition $w = \operatorname{col}(u, y)$)

Compute the right matrix fraction of $G(z) := P^{-1}(z)Q(z)$

$$G(z) = Q_\mathrm{l}(z)P_\mathrm{l}^{-1}(z)$$

Then

$$M(z) = \begin{bmatrix} P_\mathrm{l}(z) \\ Q_\mathrm{l}(z) \end{bmatrix}$$

which also reduces (R) to (M).

Misfit computation using I/S/O representation

$$\text{minimize}_{\widehat{w}} \quad \|w_\text{d} - \widehat{w}\|_{\ell_2} \quad \text{subject to} \quad \widehat{w} \in \mathscr{B} := \mathscr{B}_\text{i/s/o}(A,B,C,D) \quad \text{(SS)}$$

Recall from Lecture 5 that

$w = (u,y) \in \mathscr{B}_\text{i/s/o}(A,B,C,D) \quad \iff \quad$ there exists $x_\text{ini} \in \mathbb{R}^\text{n}$, such that

$$y = \underbrace{\begin{bmatrix} C \\ CA \\ CA^2 \\ \vdots \\ CA^{T-1} \end{bmatrix}}_{\mathscr{O}} x_\text{ini} + \underbrace{\begin{bmatrix} H(0) & & & & \\ H(1) & H(0) & & & \\ H(2) & H(1) & H(0) & & \\ \vdots & \ddots & \ddots & \ddots & \\ H(T-1) & \cdots & H(2) & H(1) & H(0) \end{bmatrix}}_{\mathscr{T}_H} u$$

where $H(0) = D$ and $H(t) = CA^{t-1}B$, for $t = 1,2,\dots$

Then (SS) is equivalent to the ordinary LS problem:

$$\text{minimize}_{x_{\text{ini}},\widehat{u}} \quad \left\| \begin{bmatrix} u_{\text{d}} \\ y_{\text{d}} \end{bmatrix} - \begin{bmatrix} I & 0 \\ \mathscr{O} & \mathscr{T}_H \end{bmatrix} \begin{bmatrix} x_{\text{ini}} \\ \widehat{u} \end{bmatrix} \right\| \tag{SS'}$$

Efficient solution via Riccati recursion $\quad\rightsquigarrow\quad$ Kalman smoother

References

1. B. Roorda and C. Heij,
   Global total least squares modeling of multivariate time series,
   IEEE-AC, 40(1):50–63, 1995

2. I. Markovsky et al.,
   Application of structured total least squares for system identification and model reduction,
   IEEE-AC, 50(10):1490–1500, 2005

3. P. Lemmerling and B. De Moor,
   Misfit versus latency,
   Automatica, 37:2057–2067, 2001.

4. I. Markovsky, J. C. Willems, S. Van Huffel, and B. De Moor.
   Exact and Approximate Modeling of Linear Systems
   SIAM, 2006

Software

A Matlab toolbox:

```
ftp.esat.kuleuven.be/pub/SISTA/markovsky/
                              abstracts/04-221a.html
```

Exercises 3 applies a simple GTLS algorithm on benchmark problem and compares the results with the ones of the latency minimization approach, implemented in the System Identification Toolbox.