The Riemannian Singular Value Decomposition

Philippe Dreesen   Bart De Moor

Katholieke Universiteit Leuven
Department of Electrical Engineering – ESAT
Research Division SCD

23 June 2008

Outline

Least Squares

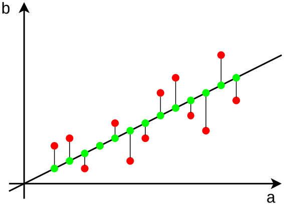
- Measurement:

$$Av \approx b$$

- Correction:

$$Av = b + \Delta b$$

- C.F. Gauss (± 1794): Predict future location of asteroid Ceres



$$\begin{array}{ll} \min & ||\Delta b||_2^2 , \\ \text{s.t.} & Av = b + \Delta b, \\ & v^T v = 1 \end{array}$$

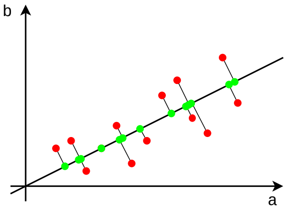
Total Least Squares

- Measurement:

$$Av \approx b$$

- Correction:

$$(A + \Delta A)v = (b + \Delta b)$$



$$\begin{array}{ll} \min & \left|\left|\begin{bmatrix}\Delta A & \Delta b\end{bmatrix}\right|\right|_F^2, \\ \text{s.t.} & (A + \Delta A)v = b + \Delta b, \\ & v^T v = 1 \end{array}$$

Structured Total Least Squares

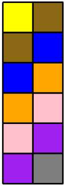
- Measurement:

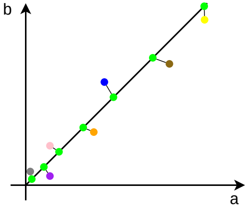
$$Av \approx b$$

- Correction:

$$(A + \Delta A)v = (b + \Delta b)$$

$$\begin{bmatrix} \Delta A & \Delta b \end{bmatrix} \text{ structured}$$

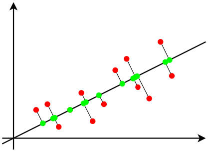




$$\begin{array}{ll} \min & \left|\left| \begin{bmatrix} \Delta A & \Delta b \end{bmatrix} \right|\right|_F^2 , \\ \text{s.t.} & (A + \Delta A)v = b + \Delta b, \\ & v^T v = 1 \\ & \begin{bmatrix} \Delta A & \Delta b \end{bmatrix} \text{ structured} \end{array}$$

TLS vs. STLS

Static Linear Modeling



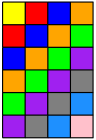
- Rank Deficiency
- minimization problem:

$$\begin{array}{ll} \min & \|[\Delta A \quad \Delta b]\|_F^2, \\ \text{s.t.} & (A + \Delta A)v = b + \Delta b, \\ & v^T v = 1 \end{array}$$

- Singular Value Decomposition:
  find (u, σ, v) which minimizes σ^2

$$\left\{ \begin{array}{rcl} A^T u & = & v\sigma \\ Av & = & u\sigma \\ v^T v & = & 1 \\ u^T u & = & 1 \end{array} \right.$$

Dynamic Linear Modeling



- Rank deficiency
- minimization problem:

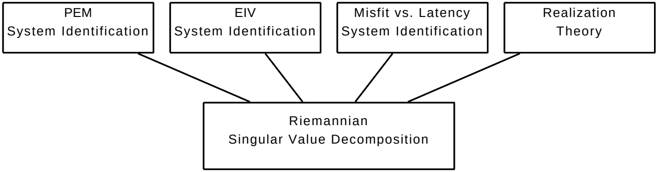
$$\begin{array}{ll} \min & \|[\Delta A \quad \Delta b]\|_F^2, \\ \text{s.t.} & (A + \Delta A)v = b + \Delta b, \\ & v^T v = 1 \\ & [\Delta A \quad \Delta b] \text{ structured} \end{array}$$

- Riemannian SVD:
  find (u, τ, v) which minimizes τ^2

$$\left\{ \begin{array}{rcl} A^T u & = & D_u v\tau \\ Av & = & D_v u\tau \\ v^T v & = & 1 \\ u^T D_v u & = & 1 \; (= v^T D_u v) \end{array} \right.$$

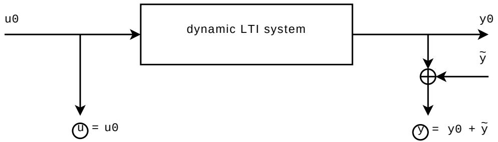
Outline

Applications



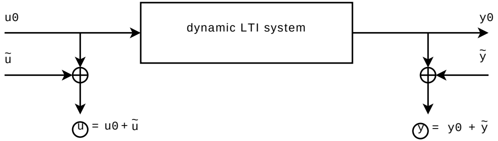
- System identification
- Realization theory
- Model reduction
- Pole placement low-order controllers
- Harmonic retrieval
- Information retrieval

Prediction Error Methods



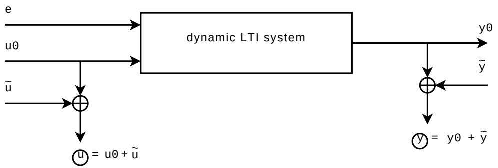
- Measurement error at the output
- Minimize error on model prediction
- Widely used (e.g. MATLAB System Identification Toolbox)

EIV System Identification



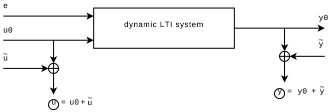
- Both input and output are subject to measurement error
- Considered a more difficult problem

Misfit vs. Latency



- **Misfit**: distance between observed trajectory and model
- **Latency**: unobserved error signal e
- Unification of System Identification methods?

Misfit vs. Latency



- Models in this framework are of the form

$$A(q)z(t) = B(q)w(t) + C(q)e(t)$$

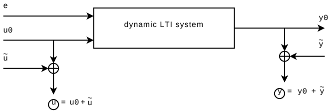
  with $z(t) = y_0(t) + \tilde{y}(t), w(t) = u_0(t) + \tilde{u}(t)$
  $A(q)$, $B(q)$ and $C(q)$ polynomials of appropriate degree

- Minimize the following cost function:

$$\min J = \alpha J_{\text{output}} + \beta J_{\text{input}} + \gamma J_{\text{latency}}$$

  with $J_{\text{output}} = \sum(\tilde{y}(t))^2$, $J_{\text{input}} = \sum(\tilde{u}(t))^2$ and $J_{\text{latency}} = \sum(e(t))^2$

Misfit vs. Latency



- In matrix format, the model selection procedure can be rephrased as:

$$\begin{array}{|ll|}\hline \min & \alpha J_{\text{output}} + \beta J_{\text{input}} + \gamma J_{\text{latency}} \\ \text{s.t.} & Za - Wb - Ec = 0 \\ \hline \end{array}$$

with a, b and c containing the model parameters, and Z, W and E Hankel matrices constructed from data

Misfit vs. Latency

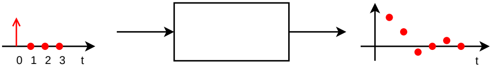
$$\begin{array}{ll} \min & \alpha J_{\text{output}} + \beta J_{\text{input}} + \gamma J_{\text{latency}} \\ \text{s.t.} & Za - Wb - Ec = 0 \end{array}$$

- Choice of the specific values of (α, β, γ) results in different LTI dynamic systems

(α, β, γ)	Case	Data misfit	Model	M/L
$(*, 1, 1)$	ARMA with noisy inputs	$U \rightarrow W$	$Wb = 0$	M+L
$(1, *, *)$	Noisy output realization	$Y \rightarrow Z$	$Za = 0$	M
$(1, 1, *)$	Dynamic EIV	$Y \rightarrow Z$, $U \rightarrow W$	$Za + Wb = 0$	M
$(1, 1, 1)$	ARMAX with noisy in/outputs	$Y \rightarrow Z$, $U \rightarrow W$	$Za + Wb + Ec = 0$	M+L
$(1, \infty, *)$	Output error	$Y \rightarrow Z$	$Za + Ub = 0$	M
$(1, \infty, 1)$	ARMAX with noisy output	$Y \rightarrow Z$	$Za + Ub + Ec = 0$	M+L
$(\infty, *, *)$	Output is impulse response		$Ya = 0$	E
$(\infty, *, 1)$	ARMA		$Ya + Ec = 0$	L
$(\infty, \infty, 1)$	ARMAX		$Ya + Ub + Ec = 0$	L

- Optimization problem only depends on the model parameters a, b and c
- Can be solved as a Riemannian SVD

Linear Realization Theory



- Impulse response experiment: measure output data $h(k)$
- Construct Hankel matrix from data:

$$H = \begin{bmatrix} h(1) & h(2) & h(3) & h(4) & \dots \\ h(2) & h(3) & h(4) & & \\ h(3) & h(4) & & & \\ h(4) & & & & \\ \vdots & & & & \end{bmatrix}$$

- $\mathrm{rank}(H) = \text{system order}$
- Direction-of-arrival estimation, chemometrics,. . .

Bilinear Realization Theory (1)

- State Space formulation:

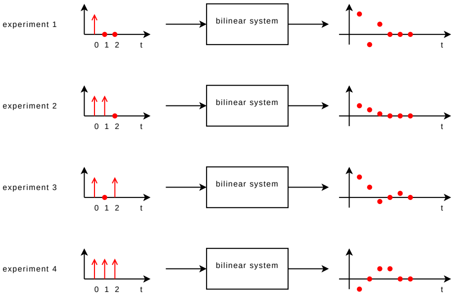
$$\begin{aligned} x_{k+1} &= Ax_k + Bu_k + Nx_k \otimes u_k \\ y_k &= Cx_k + Du_k \end{aligned}$$

- Volterra kernels (SISO):

$$\begin{array}{cccccc} CB & & & & & \\ CAB & CNB & & & & \\ CA^2B & CANB & CNAB & CN^2B & & \\ CA^3B & CA^2NB & CNA^2B & CNANB & CN^2AB & CN^3B \end{array}$$

Bilinear Realization Theory (2)

- How can we find these kernels experimentally?



- **experiment 1:**
  $D, CB, CAB, CA^2B, \ldots$
- **experiment 2:**
  $D, CB + D, CAB + CNB + CB,$
  $CA^2B + CANB + CAB, \ldots$
- **experiment 3:**
  $D, CB, CAB, CA^2B,$
  $CA^3B + CNA^2B + CB, \ldots$
- **experiment 4:**
  $D, CB + D,$
  $CAB + CNB + CB + D,$
  $CA^2B + CANB + CAB +$
  $CNAB + CN^2B + CNB + CB, \ldots$

Bilinear Realization Theory (3)

- Generalized block Hankel matrix H:

$$\left[\begin{array}{c|cc|cccc}
CB & CAB & CNB & CA^2B & CANB & CNAB & CN^2B \\
\hline
CAB & CA^2B & CANB & CA^3B & CA^2NB & CANAB & CAN^2B \\
CNB & CNAB & CN^2B & CNA^2B & CNANB & CN^2AB & CN^3B \\
\hline
CA^2B & CA^3B & CA^2NB & CA^4B & CA^3NB & CA^2NAB & CA^2N^2B \\
CNAB & CNA^2B & CNANB & CNA^3B & CNA^2NB & CNANAB & CNAN^2B \\
CANB & CANAB & CAN^2B & CANA^2B & CANANB & CAN^2AB & CAN^3B \\
CN^2B & CN^2AB & CN^3B & CN^2A^2B & CN^2ANB & CN^3AB & CN^4B
\end{array}\right]$$

Bilinear Realization Theory (4)

- Factorize:

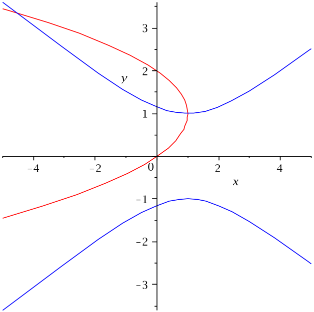
$$H = \begin{bmatrix} C \\ CA \\ CN \\ CA^2 \\ CNA \\ CAN \\ CN^2 \\ \vdots \end{bmatrix} \begin{bmatrix} B & AB & NB & A^2B & ANB & NAB & N^2B & \ldots \end{bmatrix}$$

Outline

From Polynomials to Eigenproblems

Solve the following set of equations:

$$\begin{cases} f(x,y) = y^2 - 2y + x = 0 & \text{(red)} \\ g(x,y) = 3y^2 - x^2 + 2x - 4 & \text{(blue)} \end{cases}$$



Example

Solve the following set of equations:

$$\begin{cases} f(x,y) &= y^2 - 2y + x = 0 \quad \text{(red)} \\ g(x,y) &= 3y^2 - x^2 + 2x - 4 \quad \text{(blue)} \end{cases}$$

Let

$$b = \begin{bmatrix} 1 & x & y & xy \end{bmatrix}^T$$

Find corresponding A_x and A_y such that:

$$A_x b = xb \quad \text{and} \quad A_y b = yb$$

Solving a polynomial system reduces to solving an Eigenvalue Problem!

$$\boxed{\mathbf{A}\mathbf{v} = \lambda\mathbf{v}}$$

Example

Find A_x:

$$\begin{bmatrix} 0 & 1 & 0 & 0 \\ \times & \times & \times & \times \\ 0 & 0 & 0 & 1 \\ \times & \times & \times & \times \end{bmatrix} \begin{bmatrix} 1 \\ x \\ y \\ xy \end{bmatrix} = x \begin{bmatrix} 1 \\ x \\ y \\ xy \end{bmatrix}$$

and A_y:

$$\begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ \times & \times & \times & \times \\ \times & \times & \times & \times \end{bmatrix} \begin{bmatrix} 1 \\ x \\ y \\ xy \end{bmatrix} = y \begin{bmatrix} 1 \\ x \\ y \\ xy \end{bmatrix}$$

Find expressions for x^2, x^2y, y^2 and xy^2 in terms of b...

Constructing A_x and A_y

$$f(x,y) = y^2 - 2y + x = 0 \qquad (1)$$
$$g(x,y) = 3y^2 - x^2 + 2x - 4 \qquad (2)$$

From (1) and (2):

$$\begin{aligned} x^2 &= 3y^2 + 2x - 4 \\ &= 3(2y - x) + 2x - 4 \\ &= 6y - x - 4 \qquad (3) \end{aligned}$$

From y times (3) and (1):

$$\begin{aligned} x^2 y &= 6y^2 - xy - 4y \\ &= 6(2y - x) - xy - 4y \\ &= 8y - 6 - xy \qquad (4) \end{aligned}$$

From (1):

$$y^2 = 2y - x \qquad (5)$$

From x times (1) and (3):

$$\begin{aligned} xy^2 &= 2xy - x^2 \\ &= 2xy - 6y + x + 4 \qquad (6) \end{aligned}$$

Constructing A_x and A_y

Solving

$$\begin{cases} f(x,y) &= y^2 - 2y + x = 0 \\ g(x,y) &= 3y^2 - x^2 + 2x - 4 \end{cases}$$

reduces to solving Eigenproblems

$$\begin{bmatrix} 0 & 1 & 0 & 0 \\ -4 & -1 & 6 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & -6 & 8 & -1 \end{bmatrix} \begin{bmatrix} 1 \\ x \\ y \\ xy \end{bmatrix} = x \begin{bmatrix} 1 \\ x \\ y \\ xy \end{bmatrix}$$

and

$$\begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & -1 & 2 & 0 \\ 4 & 1 & -6 & 2 \end{bmatrix} \begin{bmatrix} 1 \\ x \\ y \\ xy \end{bmatrix} = y \begin{bmatrix} 1 \\ x \\ y \\ xy \end{bmatrix}$$

Results (1)

Eigenvalue Decomposition of A_x:

$$A_x = V_x \Sigma_x V_x^{-1}$$

with

$$\begin{aligned} V_x &= \begin{bmatrix} 1 & 1 & 1 & 1 \\ -4.55 & 0.77 + 2.54i & 0.77 - 2.54i & 1 \\ 3.36 & -0.18 + 1.08i & -0.18 - 1.08i & 1 \\ -15.26 & -2.87 + 0.38i & -2.87 - 0.38i & 1 \end{bmatrix} \\ \Sigma_x &= \mathrm{diag}(-4.55, 0.77 + 2.54i, 0.77 - 2.54i, 1) \end{aligned}$$

Results (2)

Eigenvalue Decomposition of A_y:

$$A_y = V_y \Sigma_y V_y^{-1}$$

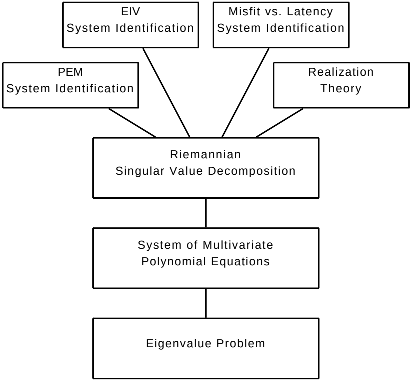
with

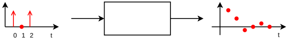
$$\begin{aligned} V_y &= \begin{bmatrix} 1 & 1 & 1 & 1 \\ -4.55 & 0.77 + 2.54i & 0.77 - 2.54i & 1 \\ 3.36 & -0.18 + 1.08i & -0.18 - 1.08i & 1 \\ -15.26 & -2.87 + 0.38i & -2.87 - 0.38i & 1 \end{bmatrix} \\ \Sigma_y &= \mathrm{diag}(3.36, -0.18 + 1.08i, -0.18 - 1.08i, 1) \end{aligned}$$

Remarks

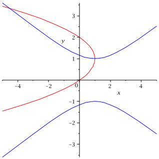
- All roots are extracted from V_x and/or V_y
- A_x and A_y commute; common eigenspaces
- How to solve for one specific root (e.g. in optimization problems)
- Direct construction of the multiplicative structure A_x, b?

From STLS to Eigenproblems





$$\begin{cases} A^T u &= D_u v \tau \\ A v &= D_v u \tau \\ v^T v &= 1 \\ u^T D_v u &= 1 \; (= v^T D_u v) \end{cases}$$



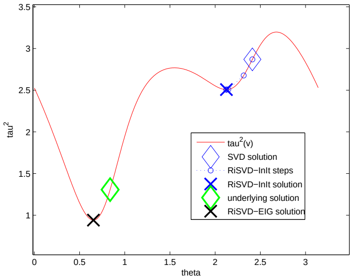
$$\mathbf{A}\mathbf{v} = \lambda \mathbf{v}$$

Preliminary Results

The Riemannian SVD is equivalent to solving the following minimization problem:

$$\boxed{\begin{array}{ll} \min\limits_{v} & \tau^2 = v^T A^T D_v^{-1} A v \\ \text{s.t.} & v^T v = 1. \end{array}}$$

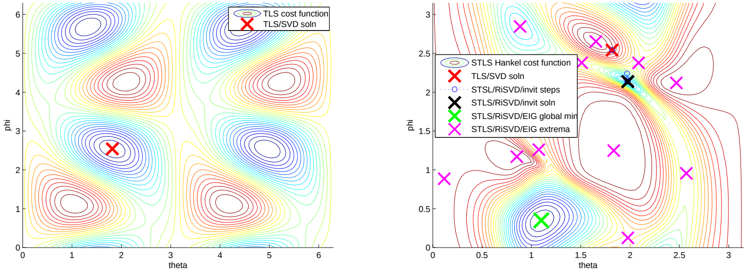
3×2 Hankel STLS



	1	**2**	3	4
v_1	.8941	**.7939**	.5259	.0215
v_2	-.4478	**.6081**	-.8505	.9998
τ^2	3.1975	**.9419**	2.5074	2.7684
global solution?	no	**yes**	no	no

Eigenvalue decomposition on 20×20 matrix

6×3 Hankel STLS



method	TLS/SVD	STLS/RiSVD inverse iteration	STLS/RiSVD eigenproblem
v_1	.8003	.4922	**.8372**
v_2	-.5479	-.7757	**.3053**
v_3	.2434	.3948	**.4535**
τ^2	4.8438	3.0518	**2.3822**
global solution?	no	no	**yes**

Eigenvalue decomposition on 437×437 matrix

Outline

Conclusions

- Riemannian SVD as a tool for solving STLS problems
- Broad application field
- Riemannian SVD can be solved as an Eigenproblem
- Applicable to short data records

Challenges

- Choice of appropriate multiplication structure
- Direct construction of (minimal) Eigenproblem from polynomial system
- Solving for specific roots
- Recursive formulation of Riemannian SVD solver

The End.