12. Subspace methods

- main idea
- notation
- geometric tools
- deterministic subspace identification
- stochastic subspace identification
- combination of deterministic-stochastic identifications
- MATLAB examples

Introduction

consider a stochastic discrete-time linear system

$$x(t+1) = Ax(t) + Bu(t) + w(t), \quad y(t) = Cx(t) + Du(t) + v(t)$$

where $x \in \mathbf{R}^n, u \in \mathbf{R}^m, y \in \mathbf{R}^l$ and $\mathbf{E} \begin{bmatrix} w(t) \\ v(t) \end{bmatrix} \begin{bmatrix} w(t) \\ v(t) \end{bmatrix}^T = \begin{bmatrix} Q & S \\ S^T & R \end{bmatrix} \delta_{t,s}$

problem statement: given input/output data $(u(t), y(t))$ for $t = 0, \ldots, N$

- find an appropriate order n
- estimate the system matrices (A, B, C, D)
- estimate the noice covariances: Q, R, S

Basic idea

the algorithm involves two steps:

1. estimation of state sequence:
   - obtained from input-output data
   - based on linear algebra tools (QR, SVD)

2. least-squares estimation of state-space matrices (once states $\hat{x}$ are known)

$$\begin{bmatrix} \hat{A} & \hat{B} \\ \hat{C} & \hat{D} \end{bmatrix} = \underset{A,B,C,D}{\text{minimize}} \left\| \begin{bmatrix} \hat{x}(t+1) & \hat{x}(t+2) & \cdots & \hat{x}(t+j) \\ y(t) & y(t+1) & \cdots & y(t+j-1) \end{bmatrix} - \begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} \hat{x}(t) & \hat{x}(t+1) & \cdots & \hat{x}(t+j-1) \\ u(t) & u(t+1) & \cdots & u(t+j-1) \end{bmatrix} \right\|_F^2$$

and $\hat{Q}, \hat{S}, \hat{R}$ are estimated from the least-squares residuals

Geometric tools

- notation and system related matrices
- row and column spaces
- orthogonal projections
- oblique projections

System related matrices

extended observability matrix

$$\Gamma_i = \begin{bmatrix} C \\ CA \\ \vdots \\ CA^{i-1} \end{bmatrix} \in \mathbf{R}^{li \times n}, \quad i > n$$

extended controllability matrix

$$\Delta_i = \begin{bmatrix} A^{i-1}B & A^{i-2}B & \cdots & AB & B \end{bmatrix} \in \mathbf{R}^{n \times mi}$$

a block Toeplitz

$$H_i = \begin{bmatrix} D & 0 & 0 & \cdots & 0 \\ CB & D & 0 & \cdots & 0 \\ CAB & CB & D & \cdots & 0 \\ \vdots & \vdots & \vdots & & \vdots \\ CA^{i-2}B & CA^{i-3}B & CA^{i-4}B & & D \end{bmatrix} \in \mathbf{R}^{li \times mi}$$

Notation and indexing

we use subscript i for time indexing

$$X_i = \begin{bmatrix} x_i & x_{i+1} & \cdots & x_{i+j-2} & x_{i+j-1} \end{bmatrix} \in \mathbf{R}^{n\times j}, \quad \text{usually } j \text{ is large}$$

$$U_{0|2i-1} \triangleq \left[\begin{array}{ccccc} u_0 & u_1 & u_2 & \cdots & u_{j-1} \\ u_1 & u_2 & u_3 & \cdots & u_j \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ u_{i-1} & u_i & u_{i+1} & \cdots & u_{i+j-2} \\ \hline u_i & u_{i+1} & u_{i+2} & \cdots & u_{i+j-1} \\ u_{i+1} & u_{i+2} & u_{i+3} & \cdots & u_{i+j} \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ u_{2i-1} & u_{2i} & u_{2i+1} & \cdots & u_{2i+j-2} \end{array}\right] = \left[\begin{array}{c} U_{0|i-1} \\ \hline U_{i|2i-1} \end{array}\right] = \left[\begin{array}{c} U_p \\ \hline U_f \end{array}\right]$$

- $U_{0|2i-1}$ has $2i$ blocks and j columns and usually j is large
- U_p contains the past inputs and U_f contains the future inputs

we can shift the index so that the top block contain the row of u_i

$$U_{0|2i-1} \triangleq \left[\begin{array}{ccccc} u_0 & u_1 & u_2 & \cdots & u_{j-1} \\ u_1 & u_2 & u_3 & \cdots & u_j \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ u_{i-1} & u_i & u_{i+1} & \cdots & u_{i+j-2} \\ u_i & u_{i+1} & u_{i+2} & \cdots & u_{i+j-1} \\ \hline u_{i+1} & u_{i+2} & u_{i+3} & \cdots & u_{i+j} \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ u_{2i-1} & u_{2i} & u_{2i+1} & \cdots & u_{2i+j-2} \end{array}\right] = \left[\begin{array}{c} U_{0|i} \\ \hline U_{i+1|2i-1} \end{array}\right] = \left[\begin{array}{c} U_p^+ \\ \hline U_f^- \end{array}\right]$$

- $+/-$ can be used to shift the border between the past and the future block
- $U_p^+ = U_{0|i}$ and $U_f^- = U_{i+1|2i-1}$
- the output matrix $Y_{0|2i-1}$ is defined in the same way
- $U_{0|2i-1}$ and $Y_{0|2i-1}$ are **block Hankel** matrices (same block along anti-diagonal)

Row and Column spaces

let $A \in \mathbf{R}^{m \times n}$

row space	column space
$\mathbf{row}(A) = \left\{ y \in \mathbf{R}^n \mid y = A^T x, \; x \in \mathbf{R}^m \right\}$	$\mathcal{R}(A) = \{ y \in \mathbf{R}^m \mid y = Ax, \; x \in \mathbf{R}^n \}$
$z^T = u^T A$	$z = Au$
z^T is in $\mathbf{row}(A)$	z is in $\mathcal{R}(A)$
$Z = BA$	$Z = AB$
rows of Z are in $\mathbf{row}(A)$	columns of Z are in $\mathcal{R}(A)$

it's obvious from the definition that

$$\mathbf{row}(A) = \mathcal{R}(A^T)$$

Orthogonal projections

denote P the projections on the row or the column space of B

$\mathbf{row}(B)$		$\mathcal{R}(B)$	
$P(y^T)$	$= y^T B^T (BB^T)^{-1} B$	$P(y)$	$= B(B^T B)^{-1} B^T y$
B	$= \begin{bmatrix} L & 0 \end{bmatrix} \begin{bmatrix} Q_1^T \\ Q_2^T \end{bmatrix}$	B	$= \begin{bmatrix} Q_1 & Q_2 \end{bmatrix} \begin{bmatrix} R \\ 0 \end{bmatrix}$
$P(y^T)$	$= y^T Q_1 Q_1^T$	$P(y)$	$= Q_1 Q_1^T y$
$(I-P)(y^T)$	$= y^T Q_2 Q_2^T$	$(I-P)(y)$	$= Q_2 Q_2^T y$
A/B	$= AB^T (BB^T)^{-1} B$	A/B	$= B(B^T B)^{-1} B^T A$

- result for row space is obtained from column space by replacing B with B^T
- A/B is the projection of the $\mathbf{row}(A)$ onto $\mathbf{row}(B)$ (or projection of $\mathcal{R}(A)$ onto $\mathcal{R}(B)$)

Projection onto a row space

denote the projection matrices onto $\mathbf{row}(B)$ and $\mathbf{row}(B)^\perp$

$\mathbf{row}(B)$	$\mathbf{row}(B)^\perp$
$\Pi_B = B^T(BB^T)^{-1}B$	$\Pi_B^\perp = I - B^T(BB^T)^{-1}B$
$A/B = AB^T(BB^T)^{-1}B$	$A/B^\perp = A(I - B^T(BB^T)^{-1}B)$

get projections of $\mathbf{row}(A)$ onto $\mathbf{row}(B)$ or $\mathbf{row}(B)^\perp$ from LQ factorization

$$\begin{aligned} \begin{bmatrix} B \\ A \end{bmatrix} &= \begin{bmatrix} L_{11} & 0 \\ L_{21} & L_{22} \end{bmatrix} \begin{bmatrix} Q_1^T \\ Q_2^T \end{bmatrix} = \begin{bmatrix} L_{11}Q_1^T \\ L_{21}Q_1^T + L_{22}Q_2^T \end{bmatrix} \\ A/B &= (L_{21}Q_1^T + L_{22}Q_2^T)Q_2Q_1^T = L_{21}Q_1^T \\ A/B^\perp &= (L_{21}Q_1^T + L_{22}Q_2^T)Q_2Q_2^T = L_{22}Q_2^T \end{aligned}$$

Oblique projection

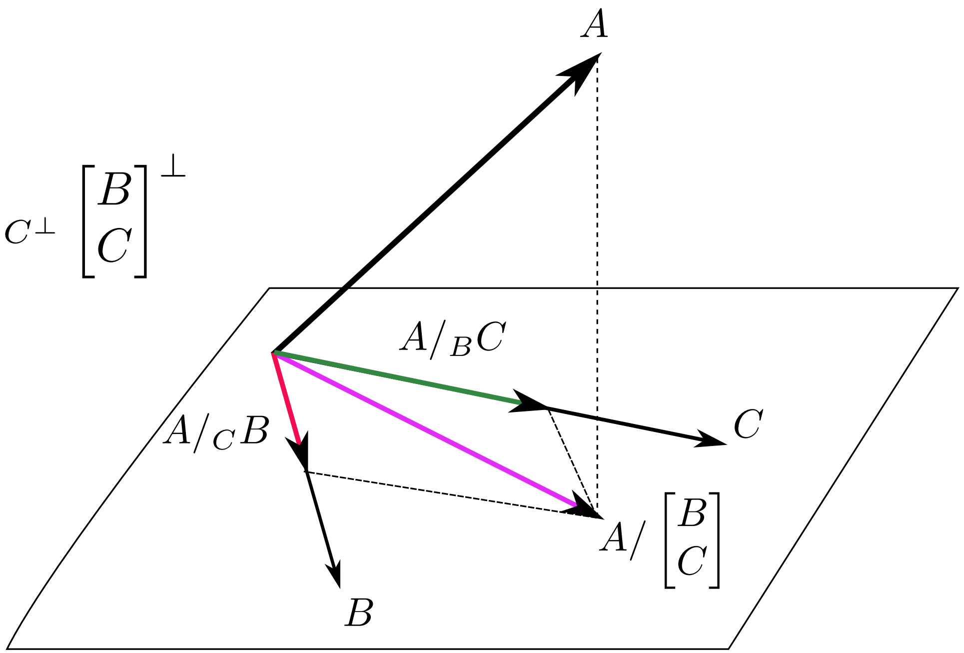
instead of an *orthogonal* decomposition $A = A\Pi_B + A\Pi_{B^\perp}$,
we represent $\mathbf{row}(A)$ as a linear combination of

the rows of two *non-orthogonal* matrices B and C and
of the orthogonal complement of B and C

$$A = L_B B + L_C C + L_{B^\perp, C^\perp} \begin{bmatrix} B \\ C \end{bmatrix}^\perp$$

$$L_C C \triangleq A/_B C$$

$$L_B B \triangleq A/_C B$$



$A/_B C$ is called the **oblique projection** of $\mathbf{row}(A)$ along $\mathbf{row}(B)$ into $\mathbf{row}(C)$

the oblique projection can be interpreted as follows

1. project $\mathbf{row}(A)$ orthogonally into the *joint* row of B and C that is $A / \begin{bmatrix} B \\ C \end{bmatrix}$
2. decompose the result in part 1) along $\mathbf{row}(B)$, denoted as $L_B B$
3. decompose the result in part 1) along $\mathbf{row}(C)$, denoted as $L_C C$
4. the orthogonal complement of the result in part 1) is denoted as
   $L_{B^\perp, C^\perp} \begin{bmatrix} B \\ C \end{bmatrix}^\perp$

the **oblique projection** of $\mathbf{row}(A)$ along $\mathbf{row}(B)$ into $\mathbf{row}(C)$ can be computed as

$$A/_B C = L_C C = L_{32} L_{22}^{-1} \begin{bmatrix} L_{21} & L_{22} \end{bmatrix} \begin{bmatrix} Q_1^T \\ Q_2^T \end{bmatrix}$$

where

$$\begin{bmatrix} B \\ C \\ A \end{bmatrix} = \begin{bmatrix} L_{11} & 0 & 0 \\ L_{21} & L_{22} & 0 \\ L_{31} & L_{32} & L_{33} \end{bmatrix} \begin{bmatrix} Q_1^T \\ Q_2^T \\ Q_3^T \end{bmatrix}$$

the computation of the oblique projection can be derived as follows

- the projection of $\mathbf{row}(A)$ into the joint row space of B and C is

$$A/\begin{bmatrix} B \\ C \end{bmatrix} = \begin{bmatrix} L_{31} & L_{32} \end{bmatrix} \begin{bmatrix} Q_1^T \\ Q_2^T \end{bmatrix} \qquad (1)$$

- this can also written as linear combination of the rows of $B$and C

$$A/\begin{bmatrix} B \\ C \end{bmatrix} = L_B B + L_C C = \begin{bmatrix} L_B & L_C \end{bmatrix} \begin{bmatrix} L_{11} & 0 \\ L_{21} & L_{22} \end{bmatrix} \begin{bmatrix} Q_1^T \\ Q_2^T \end{bmatrix} \qquad (2)$$

- equating (1) and (2) gives

$$\begin{bmatrix} L_B & L_C \end{bmatrix} = \begin{bmatrix} L_{11} & 0 \\ L_{21} & L_{22} \end{bmatrix}^{-1} \begin{bmatrix} L_{31} & L_{32} \end{bmatrix}$$

Equivalent form of oblique projection

the oblique projection of $\mathbf{row}(A)$ along $\mathbf{row}(B)$ into $\mathbf{row}(C)$ can also be defined as

$$A/_B C = A \begin{bmatrix} B^T & C^T \end{bmatrix} \left(\begin{bmatrix} BB^T & BC^T \\ CB^T & CC^T \end{bmatrix}^{\dagger} \right)_{\text{last } r \text{ columns}} \cdot C$$

where C has r rows

using the properties: $B/_B C = 0$ and $C/_B C = 0$, we have

corollary: oblique projection can also be defined

$$A/_B C = (A/B^{\perp}) \cdot (C/B^{\perp})^{\dagger} C$$

see detail in P.V. Overschee page 22

Subspace method

- main idea
- notation
- geometric tools
- **deterministic subspace identification**
- stochastic subspace identification
- combination of deterministic-stochastic identification
- MATLAB examples

Deterministic subspace identification

problem statement: estimate A, B, C, D in **noiseless** case from y, u

$$x(t+1) = Ax(t) + Bu(t), \quad y(t) = Cx(t) + Du(t)$$

method outline:

1. calculate the state sequence (x)
2. compute the system matrices (A, B, C, D)

it is based on the input-output equation

$$Y_{0|i-1} = \Gamma_i X_0 + H_i U_{0|i-1} \qquad (1)$$

$$Y_{i|2i-1} = \Gamma_i X_i + H_i U_{i|2i-1} \qquad (2)$$

Calculating the state sequence

derive future outputs

from state equations we have input/output equations

$$\text{past:} \quad Y_{0|i-1} = \Gamma_i X_0 + H_i U_{0|i-1}, \quad \text{future:} \quad Y_{i|2i-1} = \Gamma_i X_i + H_i U_{i|2i-1}$$

from state equations, we can write X_i (future) as

$$\begin{aligned} X_i &= A^i X_0 + \Delta_i U_{0|i-1} = A^i(-\Gamma_i^\dagger H_i U_{0|i-1} + \Gamma_i^\dagger Y_{0|i-1}) + \Delta_i U_{0|i-1} \\ &= \begin{bmatrix} \Delta_i - A^i \Gamma_i^\dagger H_i & A^i \Gamma_i^\dagger \end{bmatrix} \begin{bmatrix} U_{0|i-1} \\ Y_{0|i-1} \end{bmatrix} \triangleq L_p W_p \end{aligned}$$

future states = in the row space of past inputs and past outputs

$$Y_{i|2i-1} = \Gamma_i L_p W_p + H_i U_{i|2i-1}$$

find oblique projection of future outputs: onto past data and along the future inputs

$$A/_B C = (A/B^\perp)\cdot(C/B^\perp)^\dagger C \quad \Longrightarrow \quad Y_f/_{U_f} W_p = (Y_{i|2i-1}/U^\perp_{i|2i-1})(W_p/U^\perp_{i|2i-1})^\dagger W_p$$

the oblique projection is defined as $\mathcal{O}_i$ and can be derived as

$$\begin{aligned}
Y_{i|2i-1} &= \Gamma_i L_p W_p + H_i U_{i|2i-1} \\
Y_{i|2i-1}/U^\perp_{i|2i-1} &= \Gamma_i L_p W_p/U^\perp_{i|2i-1} + 0 \\
(Y_{i|2i-1}/U^\perp_{i|2i-1})(W_p/U^\perp_{i|2i-1})^\dagger W_p &= \Gamma_i L_p \underbrace{(W_p/U^\perp_{i|2i-1})(W_p/U^\perp_{i|2i-1})^\dagger W_p}_{W_p} \\
\mathcal{O}_i &= \Gamma_i L_p W_p = \Gamma_i X_i
\end{aligned}$$

projection = extended observability matrix · future states

we have applied the result of $FF^\dagger W_p = W_p$ which is NOT obvious

see Overschee page 41 (up to some assumptions on excitation in u)

compute the states: from SVD factorization

since Γ_i has n columns and X_i has n rows, so $\mathbf{rank}(\mathcal{O}_i) = n$

$$\mathcal{O}_i = \begin{bmatrix} U_1 & U_2 \end{bmatrix} \begin{bmatrix} \Sigma_n & 0 \\ 0 & 0 \end{bmatrix} V^T \begin{bmatrix} V_1^T \\ V_2^T \end{bmatrix} = U_1 \Sigma_n V_1^T$$
$$= U_1 \Sigma_n^{1/2} T \cdot T^{-1} \Sigma_n^{1/2} V_1^T, \quad \text{for some non-singular } T$$

the extended observability is equal to

$$\Gamma_i = U_1 \Sigma_n^{1/2} T$$

the future states is equal to

$$X_i = \Gamma_i^{\dagger} \mathcal{O}_i = \Gamma_i^{\dagger} \cdot Y_{i|2i-1} / _{U_{i|2i-1}} W_p$$

future states $=$ inverse of extended observability matrix $\cdot$ projection of future outputs

note that in Overschee use SVD of $W_1 \mathcal{O}_i W_2$ for some weight matrices

Computing the system matrices

from the definition of $\mathcal{O}_i$, we can obtain

$$\mathcal{O}_{i-1} = \Gamma_{i-1} X_{i+1} \quad \Longrightarrow \quad X_{i+1} = \Gamma_{i-1}^{\dagger} \mathcal{O}_{i-1}$$

(X_i and X_{i+1} are calculated using only input-output data)

the system matrices can be solved from

$$\begin{bmatrix} X_{i+1} \\ Y_{i|i} \end{bmatrix} = \begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} X_i \\ U_{i|i} \end{bmatrix}$$

in a linear least-squares sense

- options to solve in a single or two steps (solve A, C first then B, D)
- for two-step approach, there are many options: using LS, total LS, stable A

Subspace method

- main idea
- notation
- geometric tools
- deterministic subspace identification
- **stochastic subspace identification**
- combination of deterministic-stochastic identification
- MATLAB examples

Stochastic subspace identification

problem statement: estimate A, C, Q, S, R from the system without input:

$$x(t+1) = Ax(t) + w(t), \quad y(t) = Cx(t) + v(t)$$

where Q, S, R are noise covariances (see page 12-2)

method outline:

1. calculate the state sequence (x) from input/output data
2. compute the system matrices (A, C, Q, S, R)

note that classical identification would use Kalman filter that requires *system matrices* to estimate the state sequence

Bank of non-steady state Kalman filter

if the system matrices *would be known*, $\hat{x}_{i+q}$ would be obtained as follows

$$
\begin{array}{lll}
\hat{X}_0 & = & \begin{bmatrix} 0 & \cdots & 0 & \cdots & 0 \end{bmatrix} \\
P_0 & = & 0 \\
Y_p & & \begin{bmatrix} y_0 & \cdots & y_q & \cdots & y_{j-1} \\ \vdots & & \vdots & & \vdots \\ y_{i-1} & \cdots & y_{i+q-1} & \cdots & y_{i+j-2} \end{bmatrix} \\
\hat{X}_i & & \begin{bmatrix} \hat{x}_i & \cdots & \hat{x}_{i+q} & \cdots & \hat{x}_{i+j-1} \end{bmatrix}
\end{array}
\quad \begin{array}{c} \text{Kalman filter} \\ \downarrow \end{array}
$$

- start the filter at time q with the initial 0
- iterate the non-steady state Kalman filte over i time steps (vertical arrow down)
- note that to get $\hat{x}_{i+q}$ it uses only partial i outputs
- repeat for each of the j columns to obtain a *bank* of non-steady state KF

Calculation of a state sequence

project the future outputs: onto the past output space

$$\mathcal{O}_i \triangleq Y_{i|2i-1}/Y_{0|i-1} = Y_f/Y_p$$

it is shown in Overschee (THM 8, page 74) that

$$\mathcal{O}_i = \Gamma_i \hat{X}_i$$

(product of extended observability matrix and the vector of KF states)

define another projection and we then also obtain

$$\begin{aligned}\mathcal{O}_{i-1} &\triangleq Y_{i+1|2i-1}/Y_{0|i} = Y_f^-/Y_p^+ \\ &= \Gamma_{i-1}\hat{X}_{i+1}\end{aligned}$$

(proof on page 82 in Overschee)

compute the state: from SVD factorization

- the system order (n) is the rank of $\mathcal{O}_i$

$$\mathcal{O}_i = \begin{bmatrix} U_1 & U_2 \end{bmatrix} \begin{bmatrix} \Sigma_n & 0 \\ 0 & 0 \end{bmatrix} V^T \begin{bmatrix} V_1^T \\ V_2^T \end{bmatrix} = U_1 \Sigma_n V_1^T$$

- for some non-singular T, and from $\mathcal{O}_i = \Gamma_i \hat{X}_i$, we can obtain

$$\Gamma_i = U_1 \Sigma_n^{1/2} T, \quad \hat{X}_i = \Gamma_i^\dagger \mathcal{O}_i$$

- the shifted state $\hat{X}_{i+1}$ can be obtained as

$$\hat{X}_{i+1} = \Gamma_{i-1}^\dagger \mathcal{O}_{i-1} = (\underline{\Gamma_i})^\dagger \mathcal{O}_{i-1}$$

  where $\underline{\Gamma_i}$ denotes Γ_i without the last l rows

- $\hat{X}_i$ and $\hat{X}_{i+1}$ are obtained directly from output data (do not need to know system matrices)

Computing the system matrices

system matrices: once $\hat{X}_i$ and $\hat{X}_{i+1}$ are known, we form the equation

$$\underbrace{\begin{bmatrix} \hat{X}_{i+1} \\ Y_{i|i} \end{bmatrix}}_{\text{known}} = \begin{bmatrix} A \\ C \end{bmatrix} \underbrace{\hat{X}_i}_{\text{known}} + \underbrace{\begin{bmatrix} \rho_w \\ \rho_v \end{bmatrix}}_{\text{residual}}$$

- $Y_{i|i}$ is a block Hankel matrix with only one row of outputs
- the residuals (innovation) are uncorrelated with $\hat{X}_i$ (regressors) then solving this equation in the LS sense yields an asymptotically unbiased estimate:

$$\begin{bmatrix} \hat{A} \\ \hat{C} \end{bmatrix} = \begin{bmatrix} \hat{X}_{i+1} \\ Y_{i|i} \end{bmatrix} \hat{X}_i^{\dagger}$$

noise covariances

- the estimated noise covariances are obtained from the residuals

$$\begin{bmatrix} \hat{Q}_i & \hat{S}_i \\ \hat{S}_i^T & \hat{R}_i \end{bmatrix} = (1/j) \begin{bmatrix} \rho_w \\ \rho_v \end{bmatrix} \begin{bmatrix} \rho_w \\ \rho_v \end{bmatrix}^T$$

- the index i indicates that these are the *non-steady* state covariance of the non-steady state KF
- as $i \to \infty$, which is upon convergence of KF, we have convergence in Q, S, R

Subspace identification

- main idea
- notation
- geometric tools
- deterministic subspace identification
- stochastic subspace identification
- **combination of deterministic-stochastic identifications**
- MATLAB examples

Combined deterministic-stochastic identification

problem statement: estimate A, C, B, D, Q, S, R from the system:

$$x(t+1) = Ax(t) + Bu(t) + w(t), \quad y(t) = Cx(t) + Du(t) + v(t)$$

(system with **both** input and noise)

assumptions: (A, C) observable and see page 98 in Overschee

method outline:

1. calculate the state sequence (x) using oblique projection
2. compute the system matrices using least-squares

Calculating a state sequence

project future outputs: into the joint rows of past input/output along future inputs

define the two oblique projections

$$\mathcal{O}_i = Y_f /_{U_f} \begin{bmatrix} U_p \\ Y_p \end{bmatrix}, \quad \mathcal{O}_{i-1} = Y_f^- /_{U_f^-} \begin{bmatrix} U_p^+ \\ Y_p^+ \end{bmatrix}$$

important results: the oblique projections are the product of extended observability matrix and the KF sequences

$$\mathcal{O}_i = \Gamma_i \tilde{X}_i, \quad \mathcal{O}_{i-1} = \Gamma_{i-1} \tilde{X}_{i+1}$$

where $\tilde{X}_i$ is initialized by a particular $\hat{X}_0$ and run the same way as on page 12-23

(see detail and proof on page 108-109 in Overschee)

compute the state: from SVD factorization

- the system order (n) is the rank of $\mathcal{O}_i$

$$\mathcal{O}_i = \begin{bmatrix} U_1 & U_2 \end{bmatrix} \begin{bmatrix} \Sigma_n & 0 \\ 0 & 0 \end{bmatrix} V^T \begin{bmatrix} V_1^T \\ V_2^T \end{bmatrix} = U_1 \Sigma_n V_1^T$$

- for some non-singular T, and from $\mathcal{O}_i = \Gamma_i \hat{X}_i$, we can compute

$$\Gamma_i = U_1 \Sigma_n^{1/2} T, \quad \tilde{X}_i = \Gamma_i^\dagger \mathcal{O}_i$$

- the shifted state $\tilde{X}_{i+1}$ can be obtained as

$$\tilde{X}_{i+1} = \Gamma_{i-1}^\dagger \mathcal{O}_{i-1} = (\underline{\Gamma_i})^\dagger \mathcal{O}_{i-1}$$

  where $\underline{\Gamma_i}$ denotes Γ_i without the last l rows

- $\hat{X}_i$ (stochastic) and $\tilde{X}_i$ (combined) are different by the initial conditions

Computing the system matrices

system matrices: once $\tilde{X}_i$ and $\tilde{X}_{i+1}$ are known, we form the equation

$$\underbrace{\begin{bmatrix} \tilde{X}_{i+1} \\ Y_{i|i} \end{bmatrix}}_{\text{known}} = \begin{bmatrix} A & B \\ C & D \end{bmatrix} \underbrace{\begin{bmatrix} \tilde{X}_i \\ U_{i|i} \end{bmatrix}}_{\text{known}} + \underbrace{\begin{bmatrix} \rho_w \\ \rho_v \end{bmatrix}}_{\text{residual}}$$

- solve for A, B, C, D in LS sense and the estimated covariances are

$$\begin{bmatrix} \hat{Q}_i & \hat{S}_i \\ \hat{S}_i^T & \hat{R}_i \end{bmatrix} = (1/j) \begin{bmatrix} \rho_w \\ \rho_v \end{bmatrix} \begin{bmatrix} \rho_w \\ \rho_v \end{bmatrix}^T$$

(this approach is summarized in a combined algorithm 2 on page 124 of Overschee)

properties:

- $\tilde{X}_i$ and $\hat{X}_i$ are different by initial conditions but their difference goes to zero if either of the followings holds: (page 122 in Overschee)
  1. as $i \to \infty$
  2. the system if purely deterministic, $i.e.$, no noise in the state equation
  3. the deterministic input $u(t)$ is white noise
- the estimated system matrices are hence **biased** in many practical settings, $e.g.$, using steps, impulse input
- when at least one of the three conditions is satisfied, the estimate is asymptotically unbiased

Summary of combined identification

deterministic (no noise)	stochastic (no input)	combined
$\mathcal{O}_i = Y_f/_{U_f}\begin{bmatrix} U_p \\ Y_p \end{bmatrix}$	$\mathcal{O}_i = Y_f/Y_p$	$\mathcal{O}_i = Y_f/_{U_f}\begin{bmatrix} U_p \\ Y_p \end{bmatrix}$
$\mathcal{O}_i = \Gamma_i X_i$	$\mathcal{O}_i = \Gamma_i \hat{X}_i$	$\mathcal{O}_i = \Gamma_i \tilde{X}_i$
states are determined	state are estimated	state are estimated
	$\hat{X}_0 = 0$	$\tilde{X}_0 = X_0/_{U_f}U_p$

- without input, $\mathcal{O}_i$ is the projection of future outputs into past outputs
- with input, $\mathcal{O}_i$ should be explained jointly from past input/output data using the knowledge of inputs that will be presented to the system in the future
- with noise, the state estimates are initialized by the projection of the deterministic states

Complexity reduction

goal: to find as low-order model as possible that can predict the future

- reduce the complexity of the amount of information of the past that we need to keep track of to predict future
- thus we reduce the complexity of $\mathcal{O}_i$ (reduce the subspace dimension to n)

$$\underset{R}{\text{minimize}} \quad \|W_1(\mathcal{O}_i - \mathcal{R})W_2\|_F^2, \quad \text{subject to} \quad \mathbf{rank}(\mathcal{R}) = n$$

  W_1, W_2 are chosen to determine which part of info in $\mathcal{O}_i$ is important to retain

- then the solution is

$$\mathcal{R} = W_1^{-1} U_1 \Sigma_n V_1^T W_2^\dagger$$

  and in existing algorithms, $\mathcal{R}$ is used (instead of $\mathcal{O}_i$) to factorize for Γ_i

Algorithm variations

many algorithms in the literature start from SVD of $W_1 \mathcal{O}_i W_2$

$$W_1 \mathcal{O}_i W_2 = U_1 \Sigma_n^{1/2} T T^{-1} \Sigma_n^{1/2} V_1^T$$

and can be arranged into two classes:

1. obtain the right factor of SVD as the state estimates $\tilde{X}_i$ to find the system matrices

2. obtain the left factor of SVD as Γ_i to determine A, C and B, D, Q, S, R subsequently

algorithms: `n4sid, CVA, MOESP` they all use different choices of W_1, W_2

Conclusions

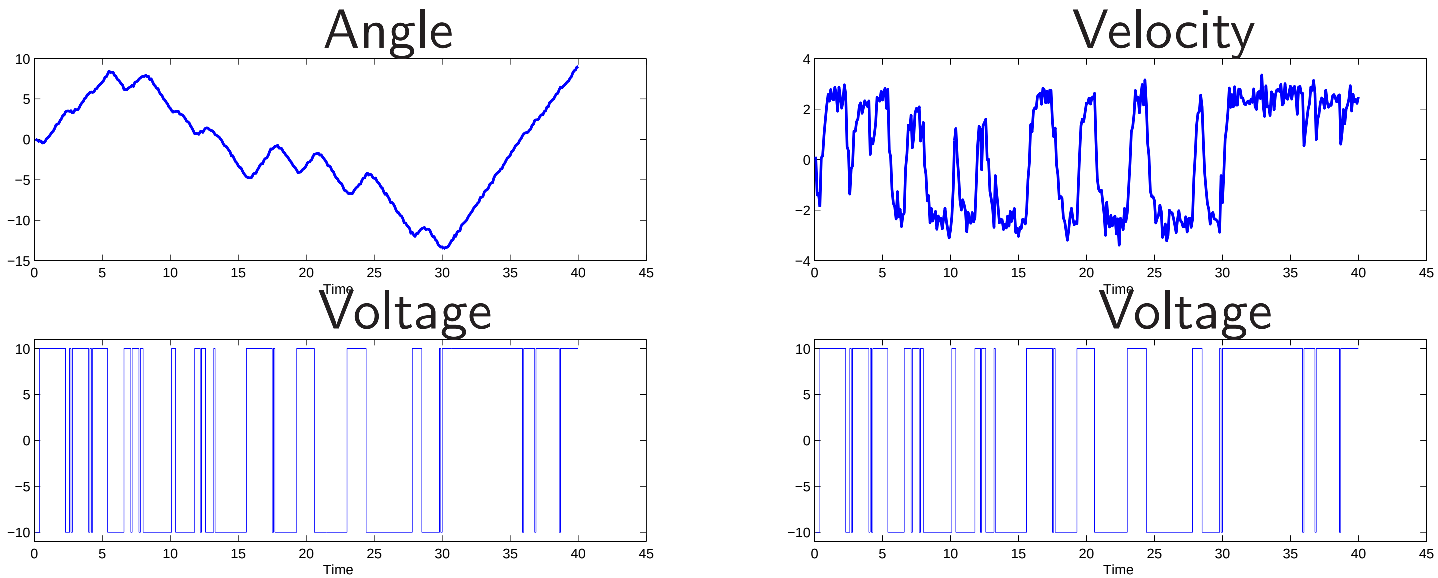
- the subspace identification consists of two main steps:
  1. estimate the state sequence *without knowing the system matrices*
  2. determine the system matrices once the state estimates are obtained
- the state sequences are estimated based on the oblique projection of future input
- the projection can be shown to be related with the extended observability matrix and the state estimates, allowing us to retrieve the states via SVD factorization
- once the states are estimated, the system matrices are obtained using LS

Example: DC motor

time response of the second-order DC motor system

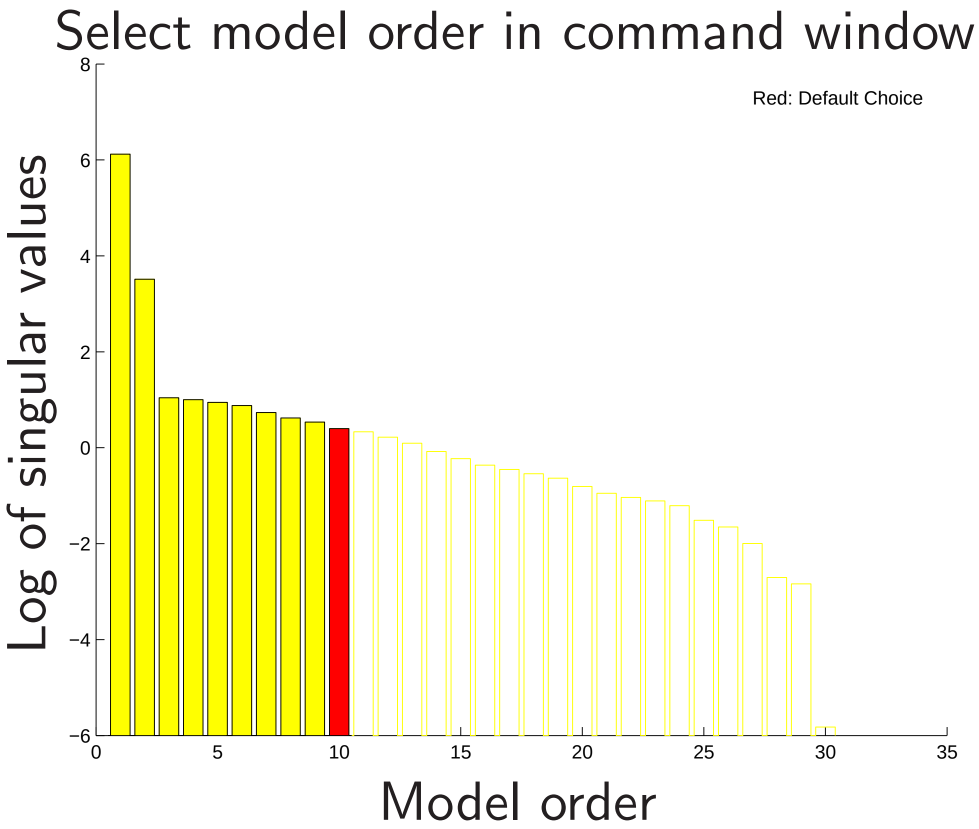
$$\dot{x}(t) = \begin{bmatrix} 0 & 1 \\ 0 & 1/\tau \end{bmatrix} x(t) + \begin{bmatrix} 0 \\ \beta/\tau \end{bmatrix} u(t) + \begin{bmatrix} 0 \\ \gamma/\tau \end{bmatrix} T_l(t)$$
$$y(t) = \begin{bmatrix} 1 & 0 \end{bmatrix} x(t)$$

where τ, β, γ are parameters to be estimated



use `n4sid` command in MATLAB

```
z = iddata(y,u,0.1);
m1 = n4sid(z,[1:10],'ssp','free','ts',0);
```



the software let the user choose the model order select $n = 2$ and the result from free parametrization is

$$A = \begin{bmatrix} 0.010476 & -0.056076 \\ 0.76664 & -4.0871 \end{bmatrix}, \quad B = \begin{bmatrix} 0.0015657 \\ -0.040694 \end{bmatrix}$$

$$C = \begin{bmatrix} 116.37 & 4.6234 \\ 4.766 & -24.799 \end{bmatrix}, \quad D = 0$$

the structure of A, B, C, D matrices can be specified

```
As = [0 1; 0 NaN]; Bs = [0; NaN];
Cs = [1 0; 0 1]; Ds = [0; 0];
Ks = [0 0;0 0]; X0s = [0;0];
```

where `NaN` is free parameter and we assign this structure to `ms` model

```
A = [0 1; 0 -1]; B = [0; 0.28];
C = eye(2); D = zeros(2,1);
ms = idss(A,B,C,D); % nominal model (or initial guess)
setstruc(ms,As,Bs,Cs,Ds,Ks,X0s);
set(ms,'Ts',0);  % Continuous model
```

the structured parametrization can be used with `pem` command

```
m2 = pem(z,ms,'display','on');
```

the estimate now has a desired structure

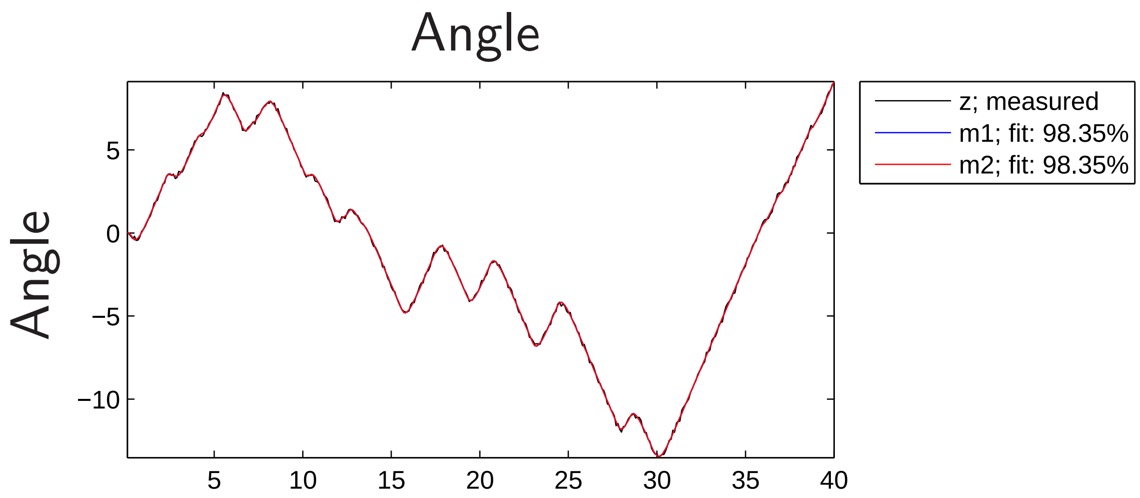
$$A = \begin{bmatrix} 0 & 1 \\ 0 & -4.0131 \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ 1.0023 \end{bmatrix}$$
$$C = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad D = 0$$

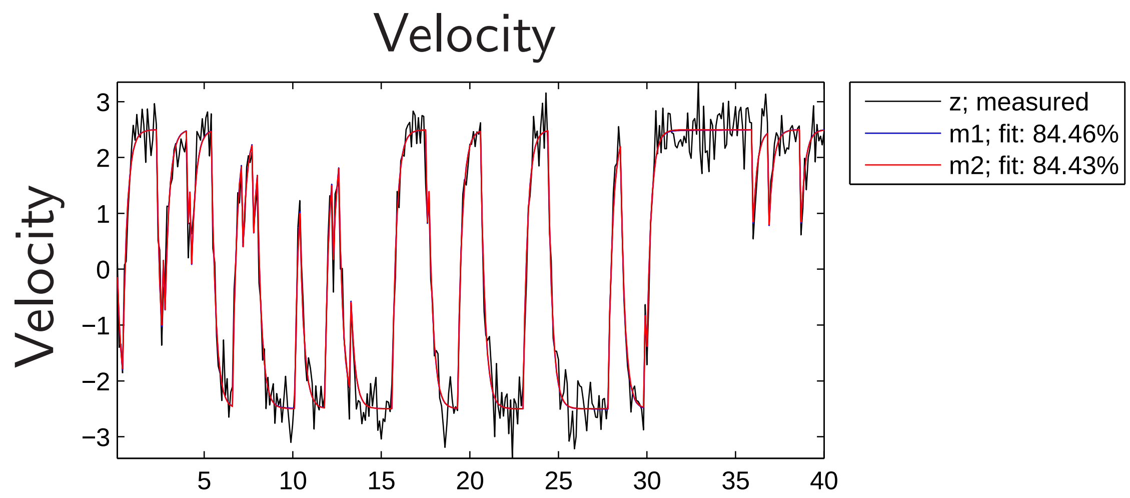
choosing model order is included in `pem` command as well

```
m3 = pem(z,'nx',1:5,'ssp','free');
```

`pem` use the `n4sid` estimate as an initial guess compare the fitting from the two models

```
compare(z,m1,m2);
```





References

Chapter 7 in
L. Ljung, *System Identification: Theory for the User*, 2nd edition, Prentice Hall, 1999

System Identification Toolbox demo
Building Structured and User-Defined Models Using System Identification Toolbox

P. Van Overschee and B. De Moor, *Subspace Identification for Linear Systems*, KLUWER Academic Publishers, 1996

K. De Cock and B. De Moor, *Subspace identification methods*, 2003