Subspace Identification Methods
--- A Tutorial

S. Joe Qin

Texas-Wisconsin Modeling and Control Consortium
Department of Chemical Engineering
University of Wisconsin-Madison
On leave from
Department of Chemical Engineering
The University of Texas at Austin
Austin, Texas 78712
February 9, 2004

Outline of This Talk

- Why Subspace Identification Methods (SIM)
- Basic State Space Concepts
- Deterministic SIMs
- Stochastic SIMs
- Additional SIM Issues

History of SIMs

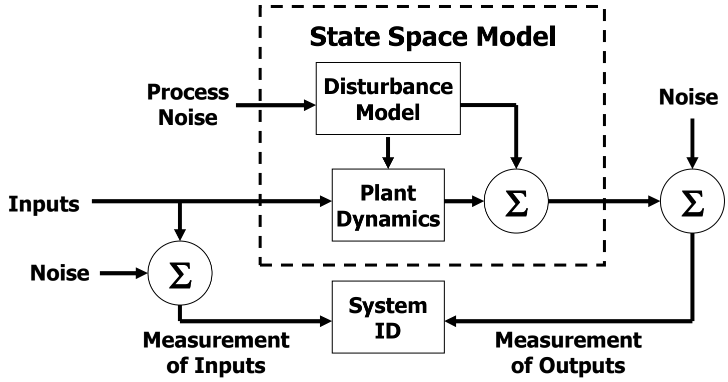
- Deterministic SIMs (Ho and Kalman, 1966)
- Stochastic Realization (Akaike, 1974)
- Canonical Variate Analysis (CVA, Larimore, 1983, 1990)
- Multivariable Output-Error State Space (MOESP, Verhaegen & Dewilde, 1992)
- Numerical algorithms for Subspace State Space System Identification (N4SID, Van Overschee & De Moor, 1994; Viberg, 1994)
- …

Why Subspace Methods?

- Simple in parameterization
  - No need for canonical forms for MIMO process models
  - Subspace first, parameterization later
  - Compact models in minimal realization
- Numerical property
  - No nonlinear optimization techniques required
- Statistical property
  - Simple Kalman filter framework

SIM Problem

From input-output measurements, estimate a **state space model** of a MIMO process given that there might be output noise, state noise, and input noise



Basic Concepts

- Linear Regression and Least Squares
- Orthogonal Projections
- Least Squares of more than One Regressor
- State Space Models
- Observability Matrix
- Extended State Space Representation

Linear Regression and Least Squares

- Given input vector x(k) and output vector y(k), build a linear relation between them

$$y(k) = \Theta x(k) + v(k)$$

- Collect data for input and output varaibles and fill the data matrices

$$\underbrace{\begin{bmatrix} y(1) & y(2) & \dots & y(N) \end{bmatrix}}_{Y} = \Theta \underbrace{\begin{bmatrix} x(1) & x(2) & \dots & x(N) \end{bmatrix}}_{X} + V$$

- The least squares solution is

$$\hat{\Theta} = YX^T (XX^T)^{-1}$$

- The model prediction is

$$\hat{Y} = \hat{\Theta} X = YX^T (XX^T)^{-1} X$$

Orthogonal Projections

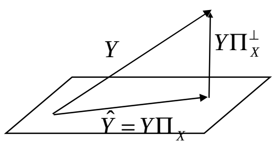
- Define

$$\Pi_X = X^T (XX^T)^{-1} X$$

as the projection matrix to the row space of X, then

$$\hat{Y} = YX^T (XX^T)^{-1} X = Y\Pi_X$$

is a projection of Y on X



- The least square residual is

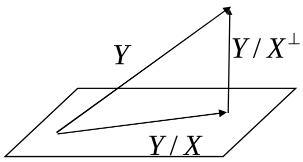
$$\tilde{Y} = Y - \hat{Y} = Y - Y\Pi_X = Y(I - \Pi_X)$$

where $\Pi_X^{\perp} = I - \Pi_X = I - X^T (XX^T)^{-1} X$

is the projection to the orthogonal complement of X

- The model $\hat{Y}$ and resiual $\tilde{Y}$ are orthogonal

Orthogonal Projection – Alternative Notation



- Sometimes we denote Y project on X as

$$Y / X = Y\Pi_X$$

and

$$Y / X^{\perp} = Y\Pi_X^{\perp}$$

which is a projection on the orthogonal complement of X

- It is easy to verify that

$$X / X = X\Pi_X = XX^T(XX^T)^{-1}X = X$$

$$X / X^{\perp} = X\Pi_X^{\perp} = X(I - X^T(XX^T)^{-1}X) = X - X = 0$$

Least Squares of more than One Regressor

- For a model with two sets of input X and U with noise V

$$Y = \Gamma X + HU + V = \begin{bmatrix} \Gamma & H \end{bmatrix} \begin{bmatrix} X \\ U \end{bmatrix} + V$$

we can find $\begin{bmatrix} \Gamma & H \end{bmatrix}$ by least squares.

- What if we are only interested in Γ?

First of all, since V is independent of U,

$$\frac{1}{N} VU^T = \frac{1}{N}\left[v(1), \ldots, v(N)\right]\left[u(1), \ldots, u(N)\right]^T \xrightarrow{N\to\infty} 0$$

$$V\Pi_U^{\perp} = V(I - U^T(UU^T)^{-1}U) = V - VU^T(UU^T)^{-1}U = V$$

Then, by 'projecting out' U

$$Y\Pi_U^{\perp} = (\Gamma X + HU + V)\Pi_U^{\perp} = \Gamma X \Pi_U^{\perp} + V$$

Γ can be found by regress $Y\Pi_U^{\perp}$ on $X\Pi_U^{\perp}$

State Space Models

- A determinist state space model is

$$\begin{cases} x(k+1) = Ax(k) + Bu(k) \\ y(k) = Cx(k) + Du(k) \end{cases}$$

- Extending the state space model into the future:

$$y(k+1) = Cx(k+1) + Du(k+1) = CAx(k) + CBu(k) + Du(k+1)$$
$$y(k+2) = Cx(k+2) + Du(k+2) = CAx(k+1) + CBu(k+1) + Du(k+2)$$
$$= CA^2x(k) + CABu(k) + CBu(k+1) + Du(k+2)$$

$\cdots$

$$y(k+j) = CA^j x(k) + \begin{bmatrix} CA^{j-1}B & \cdots & CB & D \end{bmatrix} \begin{bmatrix} u(k) \\ u(k+1) \\ \vdots \\ u(k+j) \end{bmatrix}$$

Observability Matrix

- Collect the future outputs into a vector,

$$\begin{bmatrix} y(k) \\ y(k+1) \\ \vdots \\ y(k+j) \end{bmatrix} = \underbrace{\begin{bmatrix} C \\ CA \\ \vdots \\ CA^j \end{bmatrix}}_{\Gamma_{j+1}} x(k) + \underbrace{\begin{bmatrix} D & & & \\ CB & D & & \\ \vdots & CB & \ddots & \\ CA^{j-1}B & \cdots & CB & D \end{bmatrix}}_{H_{j+1}} \begin{bmatrix} u(k) \\ u(k+1) \\ \vdots \\ u(k+j) \end{bmatrix}$$

- Γ_{j+1} is known as the observability matrix: if (C,A) is observable, Γ_{j+1} has full column rank for $j \geq n-1$, where n is the order of the system.
- H_{j+1} is known as a Toeplitz matrix referring to the special structure. H_{j+1} contains the impulse response coefficients of the model, also known as the Markov parameters
- Note that Γ_{j+1} and H_{j+1} contain all model parameters $(A,\ B,\ C,\ D)$

Define Data Matrices

Past data

$$\begin{bmatrix} y(k-p) \\ \vdots \\ y(k-2) \\ y(k-1) \end{bmatrix} \quad \begin{bmatrix} y(1) & y(2) & & y(N) \\ y(2) & y(3) & \vdots\vdots\vdots & y(N+1) \\ \vdots & \vdots & & \vdots \\ y(5) & y(6) & & y(N+4) \end{bmatrix} = Y_p;\ p \text{ block rows}$$

$$\begin{bmatrix} y(k) \\ y(k+1) \\ \vdots \\ y(k+f-1) \end{bmatrix} \quad \underbrace{\begin{bmatrix} y(6) & y(7) & & y(N+5) \\ y(7) & y(8) & \vdots & y(N+6) \\ \vdots & \vdots & & \vdots \\ y(10) & y(11) & & y(N+9) \end{bmatrix}}_{N \text{ columns}} = Y_f;\ f \text{ block rows}$$

Future data

Extended State Space Model

- From

$$\begin{bmatrix} y(k) \\ y(k+1) \\ \vdots \\ y(k+f-1) \end{bmatrix} = \underbrace{\begin{bmatrix} C \\ CA \\ \vdots \\ CA^{f-1} \end{bmatrix}}_{\Gamma_f} x(k) + \underbrace{\begin{bmatrix} D & & & \\ CB & D & & \\ \vdots & CB & \ddots & \\ CA^{f-2}B & \cdots & CB & D \end{bmatrix}}_{H_f} \begin{bmatrix} u(k) \\ u(k+1) \\ \vdots \\ u(k+f-1) \end{bmatrix}$$

Populate the data vectors with multiple columns

$$Y_f = \underbrace{\begin{bmatrix} C \\ CA \\ \vdots \\ CA^{f-1} \end{bmatrix}}_{\Gamma_f} \underbrace{\begin{bmatrix} x(1) & x(2) & \cdots & x(N) \end{bmatrix}}_{X(k)} + \underbrace{\begin{bmatrix} D & & & \\ CB & D & & \\ \vdots & CB & \ddots & \\ CA^{f-2}B & \cdots & CB & D \end{bmatrix}}_{H_f} U_f$$

or

$$\boxed{Y_f = \Gamma_f X(k) + H_f U_f}$$

which is known as the extended state space model.

Deterministic SIM

- We want to estimate Γ_f and H_f from input and output data

$$Y_f = \Gamma_f X(k) + H_f U_f$$

If only X(k) is known, this is a least squares problem.

- However, we know Γ_f is full rank (n) if we choose $f \geq n$

This is a problem of more than one regressor. Project out U_f by $\Pi_{U_f}^{\perp}$:

$$Y_f \Pi_{U_f}^{\perp} = \Gamma_f X(k) \Pi_{U_f}^{\perp} + H_f U_f \Pi_{U_f}^{\perp} = \Gamma_f X(k) \Pi_{U_f}^{\perp}$$

- From the right hand side, Γ_f has at most rank n if $f \geq n$. Therefore,

the data matrix on the left hand side is also rank n.

- Perform singular value decomposition on $Y_f \Pi_{U_f}^{\perp}$,

$$Y_f \Pi_{U_f}^{\perp} = USV^T = US^{1/2} S^{1/2} V^T$$

A balanced choice for Γ_f is: $\Gamma_f = US^{1/2}$

Estimate A and C

- To be 'smart' choose $f = n + 1$

$$\underbrace{\Gamma_2 = \begin{bmatrix} C \\ CA \\ CA^2 \\ \vdots \\ CA^{n-1} \\ CA^n \end{bmatrix}}_{\Gamma_{n+1}} \quad \begin{bmatrix} C \\ CA \\ CA^2 \\ \vdots \\ CA^{n-1} \end{bmatrix} = \Gamma_1$$

we have $\Gamma_2 = \Gamma_1 A$

C is the first row and A can be calculated using least squares

$$A = (\Gamma_1^T \Gamma_1)^{-1} \Gamma_1^T \Gamma_2$$

(A,B,C,D) and Similarity Transform

- Let $x = Tx'$ and T is invertable

$$\begin{cases} x(k+1) = Ax(k) + Bu(k) \\ y(k) = Cx(k) + Du(k) \end{cases}$$

we have

$$\begin{cases} Tx'(k+1) = ATx'(k) + Bu(k) \\ y(k) = CTx'(k) + Du(k) \end{cases} \Leftrightarrow \begin{cases} x'(k+1) = T^{-1}ATx'(k) + T^{-1}Bu(k) \\ y(k) = CTx'(k) + Du(k) \end{cases}$$

then

$$\Gamma_f' = \begin{bmatrix} CT \\ CT(T^{-1}AT) \\ \vdots \\ CT(T^{-1}AT)^{f-1} \end{bmatrix} = \begin{bmatrix} CT \\ CAT \\ \vdots \\ CA^{f-1}T \end{bmatrix} = \Gamma_f T$$

Therefore, (A,B,C,D) from SIM are not unique, but are unique up to a similarity transform. The transfer function is unique.

Representing a Stochastic System

- Process data contain state and measurement noise:

$$\begin{cases} x(k+1) = Ax(k) + Bu(k) + v(k) \\ y(k) = Cx(k) + Du(k) + w(k) \end{cases}$$

where the noise terms $v(k)$ and $w(k)$ are independent white noise

- This process has also a Kalman filter representation

$$\begin{cases} \hat{x}(k+1) = A\hat{x}(k) + Bu(k) + K(y(k) - C\hat{x}(k) + Du(k)) \\ \text{define innovation: } e(k) = y(k) - C\hat{x}(k) - Du(k) \end{cases}$$

or equivalently we have the innovation form Kalman filter:

$$\begin{cases} \hat{x}(k+1) = A\hat{x}(k) + Bu(k) + Ke(k) \\ y(k) = C\hat{x}(k) + Du(k) + e(k) \end{cases}$$

- If we look carefully, both the innovation form Kalman filter and the orignal process represent the input and output data $u(k)$ and $y(k)$ exactly
Therefore, both models can represent the input and output data, and both have the same $A,\ B, C, D$ matrices

Innovation Form State Space Model

- Similar to the deterministic model, we have

$$\boxed{Y_f = \Gamma_f X(k) + H_f U_f + G_f E_f}$$

where $G_f = \begin{bmatrix} I & & & \\ CK & I & & \\ \vdots & CK & \ddots & \\ CA^{f-2}K & \cdots & CK & I \end{bmatrix}$

- The Kalman state $X(k)$ is unknown, but we know that any Kalman state is estimated from past input and output data, i.e.,

$$X(k) = \begin{bmatrix} L_u & L_y \end{bmatrix} \begin{bmatrix} U_p \\ Y_p \end{bmatrix} = L_z Z_p$$

which is a finite impulse response (FIR) for the state. Hence,

$$\boxed{Y_f = \Gamma_f L_z Z_p + H_f U_f + G_f E_f}$$

System Identification: Battle Against Noise

- Under open loop tests, E_f is uncorrelated to U_f,

$$E_f U_f^T = 0$$

or $$E_f \Pi_{U_f}^{\perp} = E_f (I - U_f^T (U_f U_f^T)^{-1} U_f) = E_f$$

- Under open loop tests, E_f is uncorrelated to $Z_p = \begin{bmatrix} U_p \\ Y_p \end{bmatrix}$,

$$E_f Z_p^T = 0$$

The above two relations are very useful in SIMs.

SIM: An SVD Approach (N4SID, etc)

Step 0. Collect data under open loop test, Y_f, U_f, Z_p.

Step 1. Projecting out U_f by multiplying $\Pi_{U_f}^{\perp}$

$$Y_f \Pi_{U_f}^{\perp} = \Gamma_f L_z Z_p \Pi_{U_f}^{\perp} + H_f U_f \Pi_{U_f}^{\perp} + G_f E_f \Pi_{U_f}^{\perp}$$
$$= \Gamma_f L_z Z_p \Pi_{U_f}^{\perp} + G_f E_f$$

Step 2. Remove the noise term by multiplying $Z_p{}^T$,

$$Y_f \Pi_{U_f}^{\perp} Z_p{}^T = \Gamma_f L_z Z_p \Pi_{U_f}^{\perp} Z_p{}^T + G_f E_f Z_p{}^T$$
$$= \Gamma_f L_z Z_p \Pi_{U_f}^{\perp} Z_p{}^T$$

We have data on the left hand side and unknowns on the RHS

Step 3. Perform SVD,

$$Y_f \Pi_{U_f}^{\perp} Z_p{}^T = USV^T$$

and choose $\Gamma_f = US^{1/2}$ as a balanced realization.

SIM: A Regression Approach

Step 0. Collect data under open loop test, Y_f, U_f, Z_p.

Step 1. Projecting out U_f by multiplying $\Pi_{U_f}^{\perp}$

$$Y_f \Pi_{U_f}^{\perp} = \Gamma_f L_z Z_p \Pi_{U_f}^{\perp} + G_f E_f$$

Step 2. Perform least squares to find $\Gamma_f L_z$,

$$\hat{\Gamma}_f L_z = Y_f \Pi_{U_f}^{\perp} (\Pi_{U_f}^{\perp} Z_p{}^T)(Z_p \Pi_{U_f}^{\perp} \Pi_{U_f}^{\perp} Z_p{}^T)^{-1}$$
$$= Y_f \Pi_{U_f}^{\perp} Z_p{}^T (Z_p \Pi_{U_f}^{\perp} Z_p{}^T)^{-1}$$

Step 3. Perform SVD,

$$\hat{\Gamma}_f L_z = USV^T$$

and choose $\hat{\Gamma}_f = US^{1/2}$ as a balanced realization.

SIM: Reduced Rank Regression (CVA)

Step 0. Collect data under open loop test, Y_f, U_f, Z_p.

Step 1. Projecting out U_f by multiplying $\Pi_{U_f}^{\perp}$

$$Y_f \Pi_{U_f}^{\perp} = \Gamma_f L_z Z_p \Pi_{U_f}^{\perp} + G_f E_f$$

Observation: Notice that $\Gamma_f L_z$ is not full rank!

Therefore, the two step regression approach is not a good idea.

Canonical correlation analysis (CCA) is optimal for reduced rank.

Step 2. Perform CCA between $Y_f \Pi_{U_f}^{\perp}$ and $Z_p \Pi_{U_f}^{\perp}$.

The non-zero canonical correlations give the best estimate of Γ_f

Note: CCA between Y and X is an SVD of

$$(X^T X)^{-1/2}(X^T Y)(Y^T Y)^{-1/2}$$

It is, in fact, three SVDs

Additional Issues in SIMs

- SIM can estimate the optimal Kalman gain from data!
- With (C,A,K) estimates, B,D can be estimated similar to maximum likelihood
- QR factorization for numerical efficiency

- What about input noise? (See Jin Wang's talk)
- What about closed-loop data? (J. Wang and W. Lin)
- SIM Model formulation is actually not causal! And has extra terms. See W. Lin's talk on how to make it causal and parsimonious.