Identification of Output-Error (OE) Models using Generalized Spectral Decomposition

Deepak Maurya, Arun K. Tangirala and Shankar Narasimhan

Systems and Controls Group

IIT Madras

January 10, 2019

Problem statement and setting

Objective

Identify the **dynamic (difference equation) model** from input data and output **measurements** using **generalized spectral decomposition** (GSD)

Problem statement and setting

Objective

Identify the **dynamic (difference equation) model** from input data and output **measurements** using **generalized spectral decomposition** (GSD)

Framework / Assumptions:

- Linear time-invariant **(LTI)** processes.

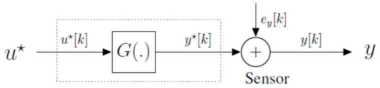


Figure 1: OE Model Setup

$$y^\star[k] + \sum_{i=1}^{n_a} a_i y^\star[k-i] = \sum_{j=D}^{n_b} b_j u^\star[k-j], \qquad \eta = \max(n_a, n_b)$$

Problem statement and setting

Objective

Identify the **dynamic (difference equation) model** from input data and output **measurements** using **generalized spectral decomposition** (GSD)

Framework / Assumptions:

- Linear time-invariant **(LTI)** processes.
- **Output** is known only with **errors**.

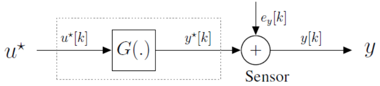


Figure 1: OE Model Setup

$$y^\star[k] + \sum_{i=1}^{n_a} a_i y^\star[k-i] = \sum_{j=D}^{n_b} b_j u^\star[k-j], \qquad \eta = \max(n_a, n_b)$$

Problem statement and setting

Objective

Identify the **dynamic (difference equation) model** from input data and output **measurements** using **generalized spectral decomposition** (GSD)

Framework / Assumptions:

- Linear time-invariant **(LTI)** processes.
- **Output** is known only with **errors**.
- **Order** and **delay** are unknown.

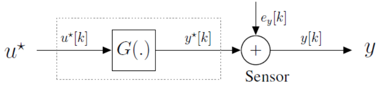


Figure 1: OE Model Setup

$$y^\star[k] + \sum_{i=1}^{n_a} a_i y^\star[k-i] = \sum_{j=D}^{n_b} b_j u^\star[k-j], \qquad \eta = \max(n_a, n_b)$$

Problem statement and setting

Objective

Identify the **dynamic (difference equation) model** from input data and output **measurements** using **generalized spectral decomposition** (GSD)

Framework / Assumptions:

- Linear time-invariant **(LTI)** processes.
- **Output** is known only with **errors**.
- **Order** and **delay** are unknown.
- Output noise variance is unknown.

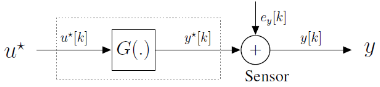


Figure 1: OE Model Setup

$$y^\star[k] + \sum_{i=1}^{n_a} a_i y^\star[k-i] = \sum_{j=D}^{n_b} b_j u^\star[k-j], \qquad \eta = \max(n_a, n_b)$$

Existing Methods (Ljung, 1998)

- **Prediction Error Minimization (PEM)** : minimize n-step ahead prediction error
- **Steiglitz-McBride (SM)** : minimizes the mean-square error between system and model outputs
- **Instrumental Variable (IV)** : Utilizes combination of projection & linear algebra based approach

Existing Methods (Ljung, 1998)

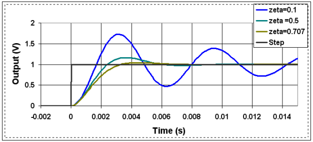
- **Prediction Error Minimization (PEM)** : minimize n-step ahead prediction error
- **Steiglitz-McBride (SM)** : minimizes the mean-square error between system and model outputs
- **Instrumental Variable (IV)** : Utilizes combination of projection & linear algebra based approach

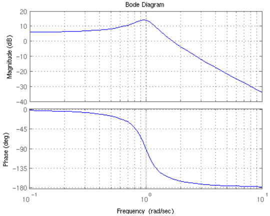
Shortcomings

All these approaches deliver unbiased and consistent parameter estimates but for right choice of model structure - delay, order

Model Structure Determination

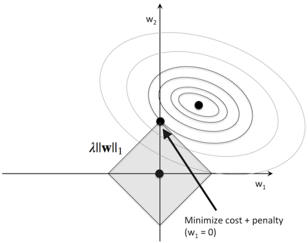
- Pre-Estimation
  - User specified using domain / process knowledge
  - Nonparametric analysis : step response, frequency domain methods





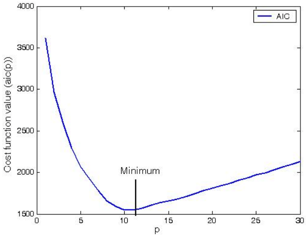
Model Structure Determination

- Pre-Estimation
  - User specified using domain / process knowledge
  - Nonparametric analysis : step response, frequency domain methods
- During Estimation
  - Regularization / Sparsification / Compressed sensing

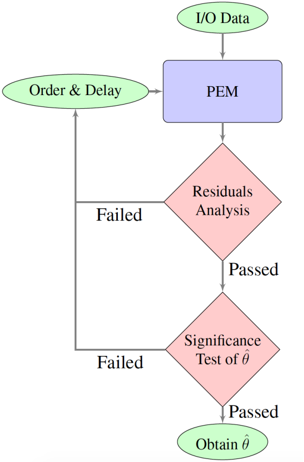


Model Structure Determination

- Pre-Estimation
  - User specified using domain / process knowledge
  - Nonparametric analysis : step response, frequency domain methods
- During Estimation
  - Regularization / Sparsification / Compressed sensing
- Post Estimation
  - Residual Analysis
  - Information criteria such as AIC or BIC



Comparison of Methodology



Comparison of Methodology

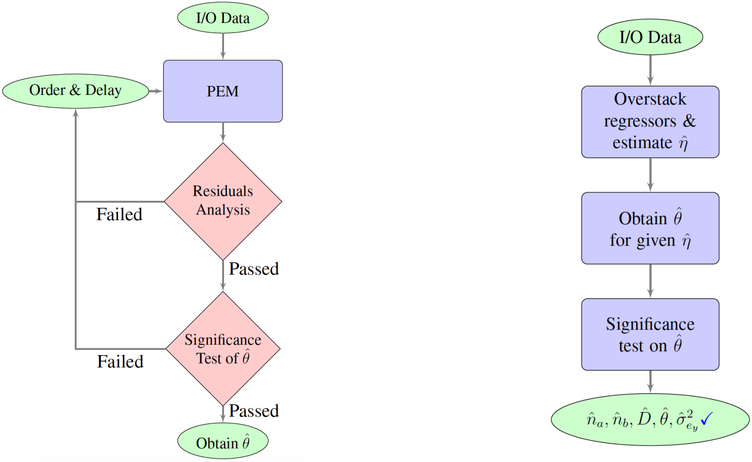


Figure 2: Comparison of Existing & Proposed Method

Dynamic Iterative PCA (Maurya et al., 2018)

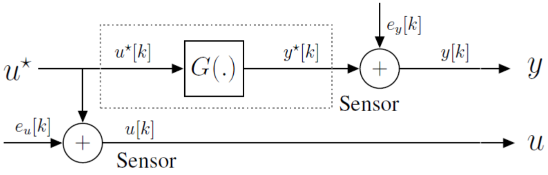


Figure 3: EIV Setup

DIPCA iteratively estimates

- noise covariance matrix

Dynamic Iterative PCA (Maurya et al., 2018)

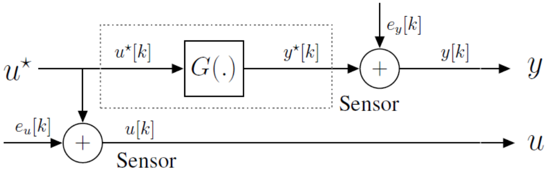


Figure 3: EIV Setup

DIPCA iteratively estimates

- noise covariance matrix
- equation order, $\hat{\eta}$

Dynamic Iterative PCA (Maurya et al., 2018)

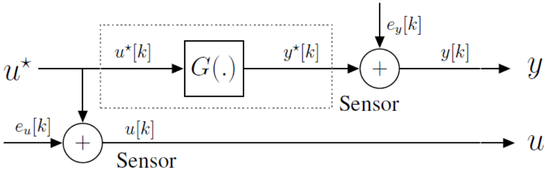


Figure 3: EIV Setup

DIPCA iteratively estimates

- noise covariance matrix
- equation order, $\hat{\eta}$
- parameter vector, $\hat{\theta}$

PCA for identification: Quick review (Jollife, 2002)

Suppose $\mathbf{z}^\star[k] \in \mathbb{R}^{M\times 1}$ are **instantaneously** related through d **linear** constraints

$$\mathbf{A}\mathbf{z}^\star[k] = 0, \qquad \mathbf{A} \in \mathbb{R}^{d\times M}, \ \text{rank}(\mathbf{A}) = d \tag{1}$$

Then, given N noise-free observations of $\mathbf{z}^\star[k]$ in $\mathbf{Z}^\star = \{\mathbf{z}^\star[k]\}_{k=0}^{N-1} \in \mathbb{R}^{N\times M}$, the following results fall out from the SVD (PCA) of $\mathbf{Z}^\star$ under $d < M$

1. rank$(\mathbf{Z}^\star) = M - d$, i.e., the last d singular values, $\sigma_{M-d+1} = \cdots = \sigma_M = 0$.
2. The **right singular vectors** corresponding to the d zero singular values provide **a basis for the null space** of $\mathbf{Z}^\star$, i.e.,

$$\bar{\mathbf{A}} \triangleq \bar{\mathbf{V}} = \begin{bmatrix} \mathbf{v}_{M-d+1} & \mathbf{v}_{M-d+2} & \cdots & \mathbf{v}_M \end{bmatrix}^T = \mathbf{T}\mathbf{A}, \qquad \det(\mathbf{T}) \neq 0 \tag{2}$$

PCA & Variants

$$\mathbf{Z} = \mathbf{Z}^\star + \mathbf{E}, \qquad \mathbf{Z}^\star \mathbf{A}^T = \mathbf{0}$$

PCA produces **unbiased** estimates only in **homoskedastic** case, meaning

$$\boldsymbol{\Sigma}_\mathbf{e} = \sigma_e^2 \mathbf{I}$$

PCA & Variants

$$\mathbf{Z} = \mathbf{Z}^\star + \mathbf{E}, \qquad \mathbf{Z}^\star \mathbf{A}^T = \mathbf{0}$$

PCA produces **unbiased** estimates only in **homoskedastic** case, meaning

$$\boldsymbol{\Sigma}_\mathbf{e} = \sigma_e^2 \mathbf{I}$$

Iterative PCA (Narasimhan and Shah, 2008)

- **Core idea** : Transform **heteroskedastic** problem to **homoskedastic** case by **scaling** with $\Sigma_\mathbf{e}^{-1/2}$, i.e., $\mathbf{z} \to \tilde{\mathbf{z}} \triangleq \Sigma_\mathbf{e}^{-1/2}\mathbf{z}$.

PCA & Variants

$$\mathbf{Z} = \mathbf{Z}^{\star} + \mathbf{E}, \qquad \mathbf{Z}^{\star}\mathbf{A}^{T} = \mathbf{0}$$

PCA produces **unbiased** estimates only in **homoskedastic** case, meaning

$$\mathbf{\Sigma_e} = \sigma_e^2 \mathbf{I}$$

Iterative PCA (Narasimhan and Shah, 2008)

- **Core idea** : Transform **heteroskedastic** problem to **homoskedastic** case by **scaling** with $\Sigma_{\mathbf{e}}^{-1/2}$, i.e., $\mathbf{z} \rightarrow \tilde{\mathbf{z}} \triangleq \Sigma_{\mathbf{e}}^{-1/2}\mathbf{z}$.
- **Result** : Eigenvalues of the (sample) covariance matrices are **shifted by unity**!

$$\lambda(\mathbf{S}_{\tilde{\mathbf{z}}}) = \lambda(\mathbf{S}_{\mathbf{z}}) + 1, \qquad \mathbf{S}_{\tilde{\mathbf{z}}} = \frac{1}{N}\mathbf{Z}^{T}\mathbf{Z}$$

Dynamic PCA (Ku et al., 1995)

Idea: Apply *static* PCA to the matrix of lagged measurements $\mathbf{Z}$, i.e., treat the dynamic relation as a **static constraint on lagged variables**.

$$\mathbf{Z} = \begin{bmatrix} y[k] & \dots & y[k-L] & u[k] & \dots & u[k-L] \\ y[k+1] & \dots & y[k-L+1] & u[k+1] & \dots & u[k-L+1] \\ \vdots & \dots & \vdots & \vdots & \dots & \vdots \\ y[N-L] & \dots & y[N-L] & u[N-L] & \dots & u[N] \end{bmatrix}$$

Underlying relation: $\boxed{\mathbf{Z}^\star \mathbf{A}^T = 0}$

DPCA requires **exact stacking** i.e. $L = \eta \implies$ Order has to be known

Optimal only for homoskedastic cases i.e $\sigma^2_{e_u} = \sigma^2_{e_y}$

Dynamic Iterative PCA (Maurya et al., 2018)

Noise Covariance Matrix (Σ_e) is available

- Sufficiently **over-stack** the lagged input-output variables.
- **Scale** the data $\mathbf{z} \rightarrow \tilde{\mathbf{z}} \triangleq \Sigma_{\mathbf{e}}^{-1/2}\mathbf{z}$.
- Identify number of linear relations (d) by applying **PCA on scaled measurements**

Dynamic Iterative PCA (Maurya et al., 2018)

Noise Covariance Matrix (Σ_e) is available

- Sufficiently **over-stack** the lagged input-output variables.
- **Scale** the data $\mathbf{z} \rightarrow \tilde{\mathbf{z}} \triangleq \Sigma_{\mathbf{e}}^{-1/2}\mathbf{z}$.
- Identify number of linear relations (d) by applying **PCA on scaled measurements**
- Estimate equation order using $\hat{\eta} = L - \hat{d} + 1$
- Reconfigure the data matrix with $\hat{\eta}$ and estimate model coefficients **from last eigenvector.**

Dynamic Iterative PCA (Maurya et al., 2018)

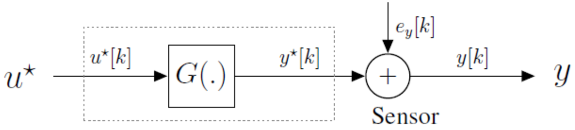
Noise Covariance Matrix (Σ_e) is available

- Sufficiently **over-stack** the lagged input-output variables.
- **Scale** the data $\mathbf{z} \rightarrow \tilde{\mathbf{z}} \triangleq \Sigma_{\mathbf{e}}^{-1/2}\mathbf{z}$.
- Identify number of linear relations (d) by applying **PCA on scaled measurements**
- Estimate equation order using $\hat{\eta} = L - \hat{d} + 1$
- Reconfigure the data matrix with $\hat{\eta}$ and estimate model coefficients **from last eigenvector.**

Noise Covariance Matrix (Σ_e) is NOT available

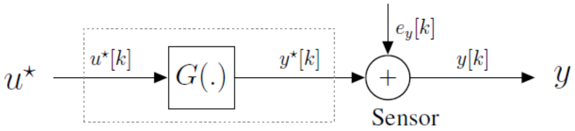
Estimate $\boldsymbol{\Sigma}_e$ by **maximizing the likelihood of residuals** from the assumed model.

Revisiting Classical Problem



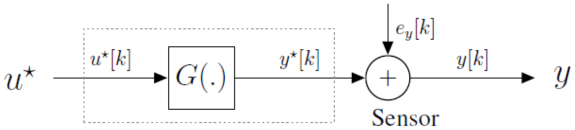
- **Scaling** ?

Revisiting Classical Problem



- **Scaling** ?
- **Over-stacking** of lagged variables ?

Revisiting Classical Problem



- **Scaling** ?
- **Over-stacking** of lagged variables ?
- **Eigen-value shift** ?

For known order & noise variance

Proposition

DIPCA algorithm minimizes the **weighted TLS** cost function which could be derived from **generalized spectral decomposition**

$$\mathbf{S_{Z_s} V_s = V_s \Lambda_s}$$

For known order & noise variance

Proposition

DIPCA algorithm minimizes the **weighted TLS** cost function which could be derived from **generalized spectral decomposition**

$$\mathbf{S}_{\mathbf{Z}_\mathbf{s}}\mathbf{V}_\mathbf{s} = \mathbf{V}_\mathbf{s}\mathbf{\Lambda}_\mathbf{s}$$

$$\frac{1}{N-\eta}\mathbf{L}^{-T}\mathbf{Z}^T\mathbf{Z}\mathbf{L}^{-1}\mathbf{V}_\mathbf{s} = \mathbf{V}_\mathbf{s}\mathbf{\Lambda}_\mathbf{s}$$

For known order & noise variance

Proposition

DIPCA algorithm minimizes the **weighted TLS** cost function which could be derived from **generalized spectral decomposition**

$$\mathbf{S_{Z_s} V_s} = \mathbf{V_s \Lambda_s}$$

$$\frac{1}{N-\eta}\mathbf{L}^{-T}\mathbf{Z}^T\mathbf{Z}\mathbf{L}^{-1}\mathbf{V_s} = \mathbf{V_s \Lambda_s}$$

$$\mathbf{L}^{-T}\mathbf{S_Z}\mathbf{L}^{-1}\mathbf{V_s} = \mathbf{V_s \Lambda_s}$$

For known order & noise variance

Proposition

DIPCA algorithm minimizes the **weighted TLS** cost function which could be derived from **generalized spectral decomposition**

$$\mathbf{S}_{\mathbf{Z}_s}\mathbf{V}_s = \mathbf{V}_s\mathbf{\Lambda}_s$$

$$\frac{1}{N-\eta}\mathbf{L}^{-T}\mathbf{Z}^T\mathbf{Z}\mathbf{L}^{-1}\mathbf{V}_s = \mathbf{V}_s\mathbf{\Lambda}_s$$

$$\mathbf{L}^{-T}\mathbf{S}_\mathbf{Z}\mathbf{L}^{-1}\mathbf{V}_s = \mathbf{V}_s\mathbf{\Lambda}_s$$

Pre-multiplying both sides with $\mathbf{L}^{-1}$

$$\mathbf{L}^{-1}\mathbf{L}^{-T}\mathbf{S}_\mathbf{Z}\mathbf{L}^{-1}\mathbf{V}_s = \mathbf{L}^{-1}\mathbf{V}_s\mathbf{\Lambda}_s$$

For known order & noise variance

Proposition

DIPCA algorithm minimizes the **weighted TLS** cost function which could be derived from **generalized spectral decomposition**

$$\mathbf{S_{Z_s}V_s} = \mathbf{V_s\Lambda_s}$$

$$\frac{1}{N-\eta}\mathbf{L}^{-T}\mathbf{Z}^T\mathbf{Z}\mathbf{L}^{-1}\mathbf{V_s} = \mathbf{V_s\Lambda_s}$$

$$\mathbf{L}^{-T}\mathbf{S_Z}\mathbf{L}^{-1}\mathbf{V_s} = \mathbf{V_s\Lambda_s}$$

Pre-multiplying both sides with $\mathbf{L}^{-1}$

$$\mathbf{L}^{-1}\mathbf{L}^{-T}\mathbf{S_Z}\mathbf{L}^{-1}\mathbf{V_s} = \mathbf{L}^{-1}\mathbf{V_s\Lambda_s}$$

$$\mathbf{S_Z}\mathbf{L}^{-1}\mathbf{V_s} = \mathbf{\Sigma_e}\mathbf{L}^{-1}\mathbf{V_s\Lambda_s}$$

For known order & noise variance

Proposition

DIPCA algorithm minimizes the **weighted TLS** cost function which could be derived from **generalized spectral decomposition**

$$\mathbf{S}_{\mathbf{Z}_\mathbf{s}}\mathbf{V}_\mathbf{s} = \mathbf{V}_\mathbf{s}\mathbf{\Lambda}_\mathbf{s}$$

$$\frac{1}{N-\eta}\mathbf{L}^{-T}\mathbf{Z}^T\mathbf{Z}\mathbf{L}^{-1}\mathbf{V}_\mathbf{s} = \mathbf{V}_\mathbf{s}\mathbf{\Lambda}_\mathbf{s}$$

$$\mathbf{L}^{-T}\mathbf{S}_\mathbf{Z}\mathbf{L}^{-1}\mathbf{V}_\mathbf{s} = \mathbf{V}_\mathbf{s}\mathbf{\Lambda}_\mathbf{s}$$

Pre-multiplying both sides with $\mathbf{L}^{-1}$

$$\mathbf{L}^{-1}\mathbf{L}^{-T}\mathbf{S}_\mathbf{Z}\mathbf{L}^{-1}\mathbf{V}_\mathbf{s} = \mathbf{L}^{-1}\mathbf{V}_\mathbf{s}\mathbf{\Lambda}_\mathbf{s}$$

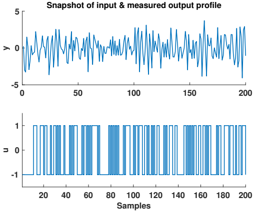
$$\mathbf{S}_\mathbf{Z}\mathbf{L}^{-1}\mathbf{V}_\mathbf{s} = \mathbf{\Sigma}_\mathbf{e}\mathbf{L}^{-1}\mathbf{V}_\mathbf{s}\mathbf{\Lambda}_\mathbf{s}$$

$$\mathbf{S}_\mathbf{Z}\mathbf{V} = \mathbf{\Sigma}_\mathbf{e}\mathbf{V}\mathbf{\Lambda}_\mathbf{s}$$

For known order & noise variance

$$y^\star[k] + 0.4y^\star[k-1] + 0.6y^\star[k-2] = 1.2u^\star[k-1]$$

$$y[k] = y^\star[k] + e_y[k], \quad \text{var}(e_y) = 0.23 \quad \text{s.t SNR} = 10$$



For known order & noise variance

$$y^\star[k] + 0.4y^\star[k-1] + 0.6y^\star[k-2] = 1.2u^\star[k-1]$$

$$y[k] = y^\star[k] + e_y[k], \quad \text{var}(e_y) = 0.23 \quad \text{s.t SNR} = 10$$

$$\begin{bmatrix} 2.63 & -0.6 & -1.23 & 1.2 \\ -0.60 & 2.63 & -0.6 & 0 \\ -1.23 & -0.6 & 2.63 & -0.02 \\ 1.2 & 0 & -0.02 & 1 \end{bmatrix} \mathbf{V} = \begin{bmatrix} 0.23 & 0 & 0 & 0 \\ 0 & 0.23 & 0 & 0 \\ 0 & 0 & 0.23 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \mathbf{V\Lambda}$$

For known order & noise variance

$$y^{\star}[k] + 0.4y^{\star}[k-1] + 0.6y^{\star}[k-2] = 1.2u^{\star}[k-1]$$

$$y[k] = y^{\star}[k] + e_y[k], \quad \text{var}(e_y) = 0.23 \quad \text{s.t SNR} = 10$$

$$\begin{bmatrix} 2.63 & -0.6 & -1.23 & 1.2 \\ -0.60 & 2.63 & -0.6 & 0 \\ -1.23 & -0.6 & 2.63 & -0.02 \\ 1.2 & 0 & -0.02 & 1 \end{bmatrix} \mathbf{V} = \begin{bmatrix} 0.23 & 0 & 0 & 0 \\ 0 & 0.23 & 0 & 0 \\ 0 & 0 & 0.23 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \mathbf{V\Lambda}$$

Spectral Decomposition using QZ algorithm

$$\lambda_{min} = \mathbf{0.97} \qquad \hat{\theta} = \begin{bmatrix} 1 & \mathbf{0.4015} & \mathbf{0.5994} & \mathbf{-1.1907} \end{bmatrix}^T \quad \checkmark$$

For known noise variance

Can use the idea of **over-stacking lagged variables** (regressors)

For known noise variance

Can use the idea of **over-stacking lagged variables** (regressors)

$$\mathbf{S_z v} = \lambda \mathbf{\Sigma v}, \qquad \mathbf{\Sigma} = \begin{bmatrix} \mathbf{\Sigma_{ey}} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}$$

For known noise variance

Can use the idea of **over-stacking lagged variables** (regressors)

$$\mathbf{S_z v} = \lambda \boldsymbol{\Sigma} \mathbf{v}, \qquad \boldsymbol{\Sigma} = \begin{bmatrix} \boldsymbol{\Sigma}_{\mathbf{ey}} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}$$

$$\lim_{N \to \infty} E[\mathbf{S_z}] = \mathbf{S_{z^\star}} + \boldsymbol{\Sigma}$$

For known noise variance

Can use the idea of **over-stacking lagged variables** (regressors)

$$\mathbf{S_z v} = \lambda \mathbf{\Sigma v}, \qquad \mathbf{\Sigma} = \begin{bmatrix} \mathbf{\Sigma_{ey}} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}$$

$$\lim_{N \to \infty} E[\mathbf{S_z}] = \mathbf{S_{z^\star}} + \mathbf{\Sigma}$$

$$(\mathbf{S_{z^\star}} + \mathbf{\Sigma})\mathbf{v} = \lambda \mathbf{\Sigma v}$$

For known noise variance

Can use the idea of **over-stacking lagged variables** (regressors)

$$\mathbf{S_z v} = \lambda \boldsymbol{\Sigma} \mathbf{v}, \qquad \boldsymbol{\Sigma} = \begin{bmatrix} \boldsymbol{\Sigma_{ey}} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}$$

$$\lim_{N \to \infty} E[\mathbf{S_z}] = \mathbf{S_{z^\star}} + \boldsymbol{\Sigma}$$

$$(\mathbf{S_{z^\star}} + \boldsymbol{\Sigma})\mathbf{v} = \lambda \boldsymbol{\Sigma} \mathbf{v}$$

$$\mathbf{S_{z^\star} v} = (\lambda - 1)\boldsymbol{\Sigma} \mathbf{v}$$

For known noise variance

Can use the idea of **over-stacking lagged variables** (regressors)

$$\mathbf{S_z v} = \lambda \boldsymbol{\Sigma} \mathbf{v}, \qquad \boldsymbol{\Sigma} = \begin{bmatrix} \boldsymbol{\Sigma}_{\mathbf{e_y}} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}$$

$$\lim_{N \to \infty} E[\mathbf{S_z}] = \mathbf{S_{z^\star}} + \boldsymbol{\Sigma}$$

$$(\mathbf{S_{z^\star}} + \boldsymbol{\Sigma})\mathbf{v} = \lambda \boldsymbol{\Sigma} \mathbf{v}$$

$$\mathbf{S_{z^\star} v} = (\lambda - 1) \boldsymbol{\Sigma} \mathbf{v}$$

Eigenvalue shift theorem doesn't hold but zero eigenvalues of *noise-free* data maps to unity generalized eigenvalues.

Generalized Case

Order, delay and output noise variance are **unknown**

$$\mathbf{S_z v} = \lambda \mathbf{\Sigma v}$$

where, $\mathbf{\Sigma} = \begin{bmatrix} \mathbf{\Sigma_{e_y}} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}$

Generalized Case

Order, delay and output noise variance are **unknown**

$$\mathbf{S_z v} = \lambda \mathbf{\Sigma v}$$

where, $$\mathbf{\Sigma} = \begin{bmatrix} \mathbf{\Sigma_{e_y}} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} = \sigma_{e_y}^2 \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}$$

Generalized Case

Order, delay and output noise variance are **unknown**

$$\mathbf{S_z v} = \lambda \boldsymbol{\Sigma} \mathbf{v}$$

where, $$\boldsymbol{\Sigma} = \begin{bmatrix} \boldsymbol{\Sigma_{e_y}} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} = \sigma_{e_y}^2 \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} = \sigma_{e_y}^2 \boldsymbol{\Sigma}'$$

Generalized Case

Order, delay and output noise variance are **unknown**

$$\mathbf{S_z v} = \lambda \boldsymbol{\Sigma} \mathbf{v}$$

where, $$\boldsymbol{\Sigma} = \begin{bmatrix} \boldsymbol{\Sigma}_{\mathbf{e_y}} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} = \sigma_{e_y}^2 \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} = \sigma_{e_y}^2 \boldsymbol{\Sigma}'$$

$$\mathbf{S_{z^\star} v} = (\lambda - \sigma_{e_y}^2) \boldsymbol{\Sigma}' \mathbf{v}$$

1. $\boldsymbol{\Sigma}'$ is a **constant**
2. Instead of **unity**, look for **equal eigenvalues**

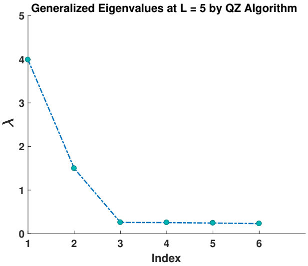
Generalized Case

Order, delay and output noise variance are **unknown**

$$\mathbf{S_z v} = \lambda \boldsymbol{\Sigma} \mathbf{v}$$

where, $$\boldsymbol{\Sigma} = \begin{bmatrix} \boldsymbol{\Sigma}_{\mathbf{e_y}} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} = \sigma_{e_y}^2 \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} = \sigma_{e_y}^2 \boldsymbol{\Sigma}'$$

$$\mathbf{S_{z^\star} v} = (\lambda - \sigma_{e_y}^2)\boldsymbol{\Sigma}' \mathbf{v}$$



1. $\boldsymbol{\Sigma}'$ is a **constant**
2. Instead of **unity**, look for **equal eigenvalues**

Order, $\hat{\eta} = L - \hat{d} + 1 = 5 - 4 + 1 = 2$ ✓

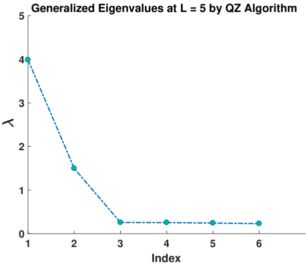
Generalized Case

Order, delay and output noise variance are **unknown**

$$\mathbf{S_z v} = \lambda \mathbf{\Sigma v}$$

where, $\mathbf{\Sigma} = \begin{bmatrix} \mathbf{\Sigma_{e_y}} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} = \sigma^2_{e_y} \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} = \sigma^2_{e_y} \mathbf{\Sigma}'$

$$\mathbf{S_{z^\star} v} = (\lambda - \sigma^2_{e_y})\mathbf{\Sigma}' \mathbf{v}$$

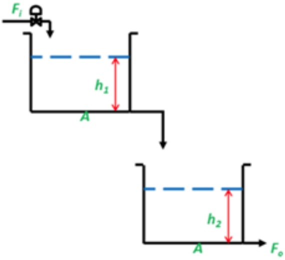


1. $\mathbf{\Sigma}'$ is a **constant**
2. Instead of **unity**, look for **equal eigenvalues**

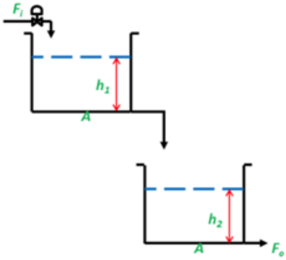
Order, $\hat{\eta} = L - \hat{d} + 1 = 5 - 4 + 1 = 2$ ✓

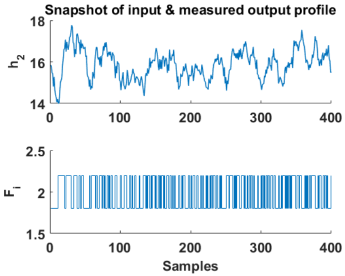
Model, $\hat{y}[k] + \underset{\pm(0.019)}{0.4005}\,\hat{y}[k-1] + \underset{\pm(0.018)}{0.6018}\,\hat{y}[k-2] = \underset{\pm(0.036)}{1.1976}\,u[k-1]$ ✓

Two non-interacting tank system

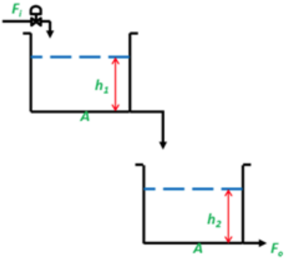


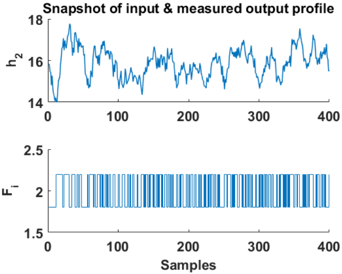
Two non-interacting tank system





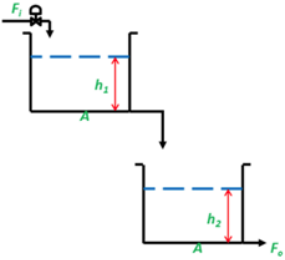
Two non-interacting tank system

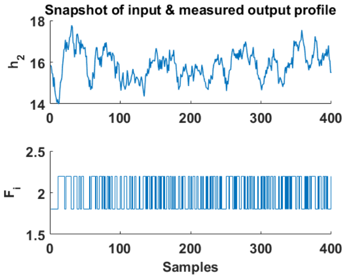




$$\frac{dh_1(t)}{dt} + \frac{Cv_1}{A_1}\sqrt{h_1(t)} = \frac{1}{A_1}F_i(t)$$
$$\frac{dh_2(t)}{dt} + \frac{Cv_2}{A_1}\sqrt{h_2(t)} = \frac{Cv_1}{A_2}\sqrt{h_1(t)}$$

Two non-interacting tank system





$$\frac{dh_1(t)}{dt} + \frac{Cv_1}{A_1}\sqrt{h_1(t)} = \frac{1}{A_1}F_i(t)$$

$$\frac{dh_2(t)}{dt} + \frac{Cv_2}{A_1}\sqrt{h_2(t)} = \frac{Cv_1}{A_2}\sqrt{h_1(t)}$$

Cv_1	Cv_2	A_1	A_2	h_{1ss}	h_{2ss}
1.8	0.5	2.4	1.2	1.23	16

Two non-interacting tank system

Step 1: Order Determination

Last 6 eigenvalues for L = 4

$$\Lambda = \begin{bmatrix} 2.03 & 0.0795 & 0.0457 & 0.044 & 0.0432 \end{bmatrix}$$

$$\hat{\eta} = L - \hat{d} + 1 = 2$$

Two non-interacting tank system

Step 1: Order Determination

Last 6 eigenvalues for L = 4

$$\Lambda = \begin{bmatrix} 2.03 & 0.0795 & 0.0457 & 0.044 & 0.0432 \end{bmatrix}$$

$$\hat{\eta} = L - \hat{d} + 1 = 2$$

Step2 : Model Estimation

$$\hat{G}(z^{-1}) = \frac{0.4208z^{-1} + 0.352z^{-2}}{1 - 1.4437z^{-1} + 0.493z^{-2}}$$

Two non-interacting tank system

Step 1: Order Determination

Last 6 eigenvalues for L = 4

$$\Lambda = \begin{bmatrix} 2.03 & 0.0795 & 0.0457 & 0.044 & 0.0432 \end{bmatrix}$$

$$\hat{\eta} = L - \hat{d} + 1 = 2$$

Step2 : Model Estimation

$$\hat{G}(z^{-1}) = \frac{0.4208z^{-1} + 0.352z^{-2}}{1 - 1.4437z^{-1} + 0.493z^{-2}}$$

$$G(s) = \frac{0.2813}{s^2 + 0.3896s + 0.0176}$$

Discretization under ZOH assumption

$$G(z^{-1}) = \frac{0.44z^{-1} + 0.33z^{-2}}{1 - 1.41z^{-1} + 0.46z^{-2}}$$

Concluding Remarks

- A systematic, rigorous method to accurately estimate the dynamic model using generalized spectral decomposition

Concluding Remarks

- A systematic, rigorous method to accurately estimate the dynamic model using generalized spectral decomposition
  - **Minimal user intervention** (maximum stacking lag to be supplied).

Concluding Remarks

- A systematic, rigorous method to accurately estimate the dynamic model using generalized spectral decomposition
  - **Minimal user intervention** (maximum stacking lag to be supplied).
  - Determines **the order**

Concluding Remarks

- A systematic, rigorous method to accurately estimate the dynamic model using generalized spectral decomposition
  - **Minimal user intervention** (maximum stacking lag to be supplied).
  - Determines **the order**
  - Delivers **output noise variance**

Concluding Remarks

- A systematic, rigorous method to accurately estimate the dynamic model using generalized spectral decomposition
  - **Minimal user intervention** (maximum stacking lag to be supplied).
  - Determines **the order**
  - Delivers **output noise variance**
  - Estimator is observed to be **consistent**

Concluding Remarks

- A systematic, rigorous method to accurately estimate the dynamic model using generalized spectral decomposition
  - **Minimal user intervention** (maximum stacking lag to be supplied).
  - Determines **the order**
  - Delivers **output noise variance**
  - Estimator is observed to be **consistent**

Concluding Remarks

- A systematic, rigorous method to accurately estimate the dynamic model using generalized spectral decomposition
  - **Minimal user intervention** (maximum stacking lag to be supplied).
  - Determines **the order**
  - Delivers **output noise variance**
  - Estimator is observed to be **consistent**

- Proposed algorithm **non-iterative** and consists of **two steps only**.

Concluding Remarks

- A systematic, rigorous method to accurately estimate the dynamic model using generalized spectral decomposition
  - **Minimal user intervention** (maximum stacking lag to be supplied).
  - Determines **the order**
  - Delivers **output noise variance**
  - Estimator is observed to be **consistent**

- Proposed algorithm **non-iterative** and consists of **two steps only**.
- Generalized framework which could handle **both EIV and classical case** with different model structures.

Future Work

- Extension to closed loop systems

Future Work

- Extension to closed loop systems
- Extension for colored noise structure such as

Future Work

- Extension to closed loop systems
- Extension for colored noise structure such as
  - Autoregressive with exogenous input (under review in ICASSP 2019)

Future Work

- Extension to closed loop systems
- Extension for colored noise structure such as
  - Autoregressive with exogenous input (under review in ICASSP 2019)
  - ARMAX, Box-Jenkins model

Future Work

- Extension to closed loop systems
- Extension for colored noise structure such as
  - Autoregressive with exogenous input (under review in ICASSP 2019)
  - ARMAX, Box-Jenkins model
- Extensions to MISO and MIMO case:

Future Work

- Extension to closed loop systems
- Extension for colored noise structure such as
  - Autoregressive with exogenous input (under review in ICASSP 2019)
  - ARMAX, Box-Jenkins model
- Extensions to MISO and MIMO case:
  - Break up the given system into individual SISO systems using signal conditioning, i.e., work with *partial* covariance matrices.

Bibliography

- Jollife IT. Principal component analysis. 1986.
- Ljung, Lennart. "System identification." Signal analysis and prediction. Birkhäuser, Boston, MA, 1998. 163-173.
- Narasimhan, Shankar, and Sirish L. Shah. "Model identification and error covariance matrix estimation from noisy data using PCA." Control Engineering Practice 16.1 (2008): 146-155.
- Ku, Wenfu, Robert H. Storer, and Christos Georgakis. "Disturbance detection and isolation by dynamic principal component analysis." Chemometrics and intelligent laboratory systems 30.1 (1995): 179-196.
- Maurya, Deepak, Arun K. Tangirala, and Shankar Narasimhan. "Identification of Errors-in-Variables Models Using Dynamic Iterative Principal Component Analysis." Industrial & Engineering Chemistry Research 57.35 (2018): 11939-11954.

Acknowledgment

Various images in the PPT were borrowed from

- https://en.wikipedia.org/wiki/
- https://emojiterra.com/thinking-face/
- https://knowyourmeme.com/
- https://www.slideshare.net/

QZ Algorithm

The QZ algorithm is numerical method for solving generalized eigenvalue problem

$$\mathbf{Av} = \mathbf{\Lambda Bv} \tag{3}$$

without performing matrix inversion $\mathbf{B}$.

1. The idea is to transform (3) to
$$\mathbf{QAZy} = \mathbf{\Lambda QBZy}, \qquad \text{under} \quad \mathbf{v} = \mathbf{Zy} \tag{4}$$
where $\mathbf{Q}$ and $\mathbf{Z}$ are unitary matrices such that $\mathbf{QAZ}$ and $\mathbf{QBZ}$ are upper triangular
2. Eigenvalues can be computed from the diagonals of the triangular form. Eigenvectors can be computed by transforming back the eigenvectors of triangular problem with $\mathbf{Z}$