Filtering and Identification

Day 4 - Lecture 1:
Subspace Identification
The Deterministic case

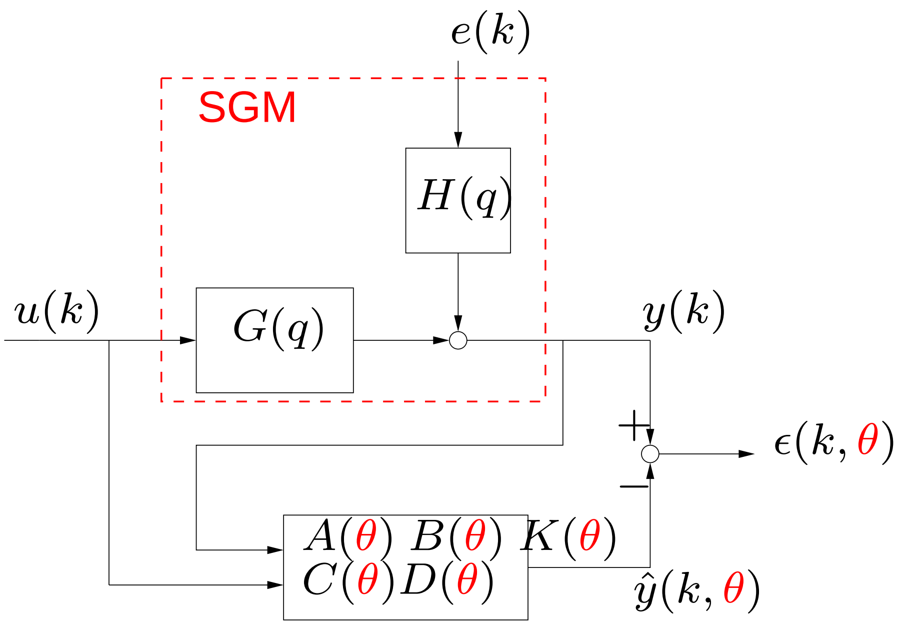
Michel Verhaegen

Division of Automatic Control - EE

Overview

- **Recap “classical” system identification**
- A different approach: Subspace Identification
  - A deterministic Identification problem
  - Subspace Identification of an autonomous system
  - Subspace Identification with inputs

Recap “Classical” (PEM) Identification methods

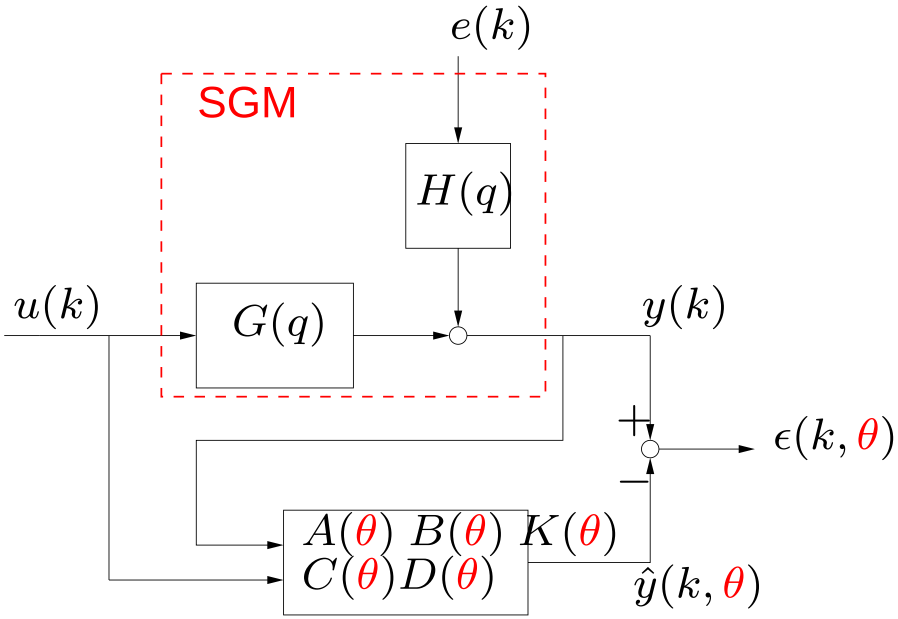


1. Assume $\widehat{\mathrm{SGM}}(\theta)$
2. Derive the “optimal” predictor for $\hat{y}(k,\theta)$.
3. Find the “best” estimate

   $\hat{\theta}_N = \text{argmin} J_N(\theta)$ with

with $J_N(\theta) = \frac{1}{N}\sum_{k=1}^{N} \|y(k) - \hat{y}(k,\theta)\|_2^2$

Recap “Classical” (PEM) Identification methods

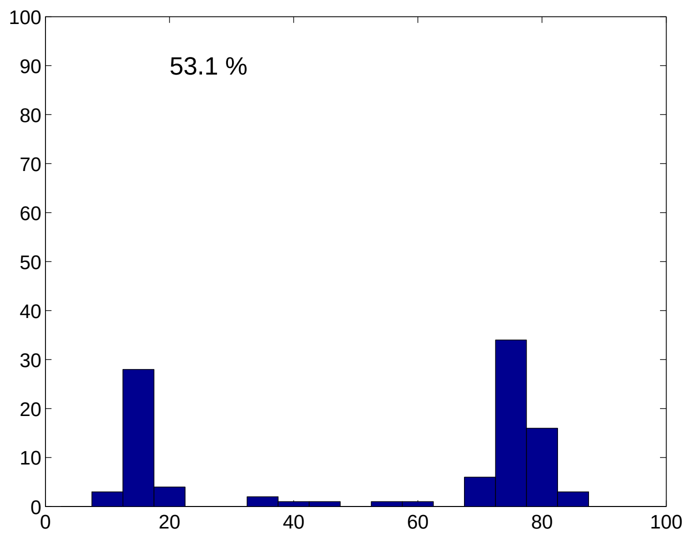


1. Assume $\widehat{\mathrm{SGM}}(\theta)$
2. Derive the “optimal” predictor for $\hat{y}(k, \theta)$.
3. Find the “best” estimate
   $\hat{\theta}_N = \text{argmin} J_N(\theta)$ with

with $J_N(\theta) = \frac{1}{N} \sum_{k=1}^{N} \|y(k) - \hat{y}(k, \theta)\|_2^2$

Challenges:

1. You need to fix the order of the SS model a priori!
2. (MIMO) parametrization is a difficult?
3. non-convex optimization



Overview

- Recap “classical” system identification
- A different approach: Subspace Identification
  - **A deterministic Identification problem**
  - Subspace Identification of an autonomous system
  - Subspace Identification with inputs

Deterministic Identification problem

SGM:

$$\begin{cases} x(k+1) &= Ax(k) + Bu(k) \\ y(k) &= Cx(k) + Du(k) \end{cases} \qquad x(k) \in \mathbb{R}^n$$

Determine n AND the system matrices A, B, C, and D from a finite number of measurements of $u(k)$ and $y(k)$ without parametrizing the system matrices, etc.

Deterministic Identification problem

SGM:

$$\begin{cases} x(k+1) &= Ax(k) + Bu(k) \\ y(k) &= Cx(k) + Du(k) \end{cases} \qquad x(k) \in \mathbb{R}^n$$

Determine n AND the system matrices A, B, C, and D from a finite number of measurements of $u(k)$ and $y(k)$ without parametrizing the system matrices, etc.

In general no unique solution!

Overview

- Recap “classical” system identification
- A different approach: Subspace Identification
  - A deterministic Identification problem
  - **Subspace Identification of an autonomous system**
  - Subspace Identification with inputs

Identification of an autonomous system

SGM: Autonomous system

$$\begin{cases} x(k+1) &= Ax(k) \\ y(k) &= Cx(k) \end{cases} \qquad x(k) \in \mathbb{R}^n$$

Output: $y(j) = CA^j x(0) \qquad j = 0 : s-1$

SIDemo.m

For a second order system ($n = 2$), we display:

$$\begin{bmatrix} y(0) \\ y(1) \\ y(2) \end{bmatrix} \quad \begin{bmatrix} y(1) \\ y(2) \\ y(3) \end{bmatrix} \quad \begin{bmatrix} y(2) \\ y(3) \\ y(4) \end{bmatrix}$$

in $\mathbb{R}^3$. What do we see?

Identification of an autonomous system

SGM: Autonomous system

$$\begin{cases} x(k+1) = Ax(k) \\ y(k) = Cx(k) \end{cases} \qquad x(k) \in \mathbb{R}^n$$

Output: $y(k+j) = CA^j x(k) \qquad j = 0 : s-1$

$$\begin{bmatrix} y(k) \\ y(k+1) \\ y(k+2) \\ \vdots \\ y(k+s-1) \end{bmatrix} = \begin{bmatrix} C \\ CA \\ CA^2 \\ \vdots \\ CA^{s-1} \end{bmatrix} x(k) = \mathcal{O}_s x(k)$$

SubidPlot.m

For the 2nd order system:

$$\begin{aligned} x(k+1) &= \begin{bmatrix} 0.9 & 0 \\ -0.5 & 0.7 \end{bmatrix} x(k) \\ y(k) &= \begin{bmatrix} 1 & -1 \end{bmatrix} x(k) \end{aligned}$$

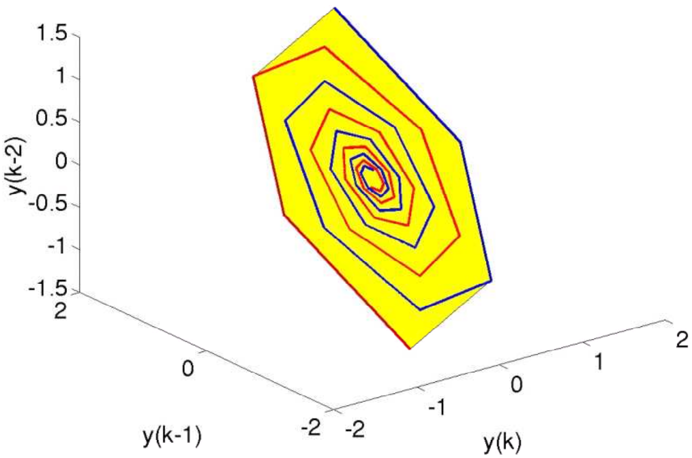
we plot the vector $\begin{bmatrix} y(k) \\ y(k+1) \\ y(k+2) \end{bmatrix}$ for $k = 1:10$ in $\mathbb{R}^3$.

What do we see?

Revealing information

We take $y(k) \in \mathbb{R}$, $n = 2$, $s = 3$.

$$\begin{bmatrix} y(k) \\ y(k+1) \\ y(k+2) \end{bmatrix} = \mathcal{O}_s x(k)$$

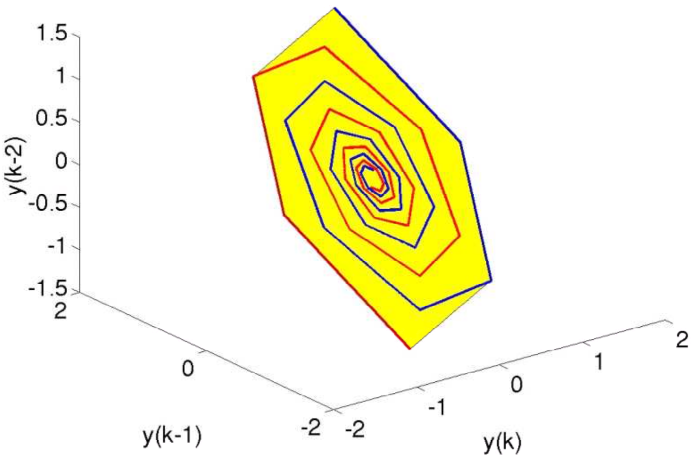


Plot for $k = 0, 1, 2, \ldots$

Revealing information

We take $y(k) \in \mathbb{R}$, $n = 2$, $s = 3$.

$$\begin{bmatrix} y(k) \\ y(k+1) \\ y(k+2) \end{bmatrix} = \mathcal{O}_s x(k)$$



Plot for $k = 0, 1, 2, \ldots$

$$\begin{bmatrix} y(0) & y(1) & \cdots & y(N-1) \\ y(1) & y(2) & \cdots & y(N) \\ y(2) & y(3) & \cdots & y(N+1) \end{bmatrix} = \begin{bmatrix} C \\ CA \\ CA^2 \end{bmatrix} \begin{bmatrix} x(0) & x(1) \cdots x(N-1) \end{bmatrix}$$

Data equation: 2nd-order example

$$\underbrace{\begin{bmatrix} y(0) & y(1) & \cdots & y(N-1) \\ y(1) & y(2) & \cdots & y(N) \\ y(2) & y(3) & \cdots & y(N+1) \end{bmatrix}}_{Y_{0,3,N}} = \underbrace{\begin{bmatrix} C \\ CA \\ CA^2 \end{bmatrix}}_{\mathcal{O}_3} \underbrace{\begin{bmatrix} x(0) & x(1) \cdots x(N-1) \end{bmatrix}}_{X_{0,N}}$$

Use Sylvester's inequality

$$\text{rank}(\mathcal{O}_3) + \text{rank}(X_{0,N}) - 2 \leq \text{rank}(Y_{0,3,N}) \leq \min\Big(\text{rank}(\mathcal{O}_3), \text{rank}(X_{0,N})\Big)$$

Data equation: 2nd-order example

$$\underbrace{\begin{bmatrix} y(0) & y(1) & \cdots & y(N-1) \\ y(1) & y(2) & \cdots & y(N) \\ y(2) & y(3) & \cdots & y(N+1) \end{bmatrix}}_{Y_{0,3,N}} = \underbrace{\begin{bmatrix} C \\ CA \\ CA^2 \end{bmatrix}}_{\mathcal{O}_3} \underbrace{\begin{bmatrix} x(0) & x(1) \cdots x(N-1) \end{bmatrix}}_{X_{0,N}}$$

Use Sylvester's inequality

$$\text{rank}(\mathcal{O}_3) + \text{rank}(X_{0,N}) - 2 \leq \text{rank}(Y_{0,3,N}) \leq \min\Big(\text{rank}(\mathcal{O}_3), \text{rank}(X_{0,N})\Big)$$

If (A, C) is observable and $(A, x(0))$ is controllable:

$$\text{rank}(Y_{0,3,N}) = 2 = n$$

Data equation: general case

Block Hankel matrix

$$Y_{i,s,N} = \begin{bmatrix} y(i) & y(i+1) & \cdots & y(i+N-1) \\ y(i+1) & y(i+2) & \cdots & y(i+N) \\ \vdots & \vdots & \ddots & \vdots \\ y(i+s-1) & y(i+s) & \cdots & y(i+N+s-2) \end{bmatrix}$$

State $X_{i,N} = \begin{bmatrix} x(i) & x(i+1) & \cdots & x(i+N-1) \end{bmatrix}$

Data equation $\boxed{Y_{0,s,N} = \mathcal{O}_s X_{0,N}}$

Determination column space $\mathcal{O}_s$

Objective: retrieve the matrices A and C from the measurements stored in $Y_{0,s,N}$

How to exploit the data equation?

$$Y_{0,s,N} = \mathcal{O}_s X_{0,N}$$

Determination column space $\mathcal{O}_s$

Objective: retrieve the matrices A and C from the measurements stored in $Y_{0,s,N}$

How to exploit the data equation?

$$Y_{0,s,N} = \mathcal{O}_s X_{0,N}$$

Columns of $Y_{0,s,N}$ are linear combinations of the n columns of $\mathcal{O}_s$.

$\Rightarrow$ range$(Y_{0,s,N}) \subseteq$ range$(\mathcal{O}_s)$

Determination column space $\mathcal{O}_s$

Objective: retrieve the matrices A and C from the measurements stored in $Y_{0,s,N}$

How to exploit the data equation?

$$Y_{0,s,N} = \mathcal{O}_s X_{0,N}$$

Columns of $Y_{0,s,N}$ are linear combinations of the n columns of $\mathcal{O}_s$.

$\Rightarrow$ range$(Y_{0,s,N}) \subseteq$ range$(\mathcal{O}_s)$

If observable and controllable: rank$(Y_{0,s,N}) = n$

$\Rightarrow$ range$(Y_{0,s,N}) =$ range$(\mathcal{O}_s)$

Compute the column space $\mathcal{O}_s$

Assume rank$(Y_{0,s,N}) = n$.

Singular value decomposition (SVD)

$$Y_{0,s,N} = \begin{bmatrix} U_n & U_n^{\perp} \end{bmatrix} \begin{bmatrix} \Sigma_n & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} (V_n)^T & (V_n^{\perp})^T \end{bmatrix}$$

with $\Sigma_n \in \mathbb{R}^{n \times n}$ and rank$(\Sigma_n) = n$.

Compute the column space $\mathcal{O}_s$

Assume $\text{rank}(Y_{0,s,N}) = n$.

Singular value decomposition (SVD)

$$Y_{0,s,N} = \begin{bmatrix} U_n & U_n^\perp \end{bmatrix} \begin{bmatrix} \Sigma_n & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} (V_n)^T & (V_n^\perp)^T \end{bmatrix}$$

with $\Sigma_n \in \mathbb{R}^{n \times n}$ and $\text{rank}(\Sigma_n) = n$.

$\text{range}(U_n) = \text{range}(Y_{0,s,N}) = \text{range}(\mathcal{O}_s)$

Compute the column space $\mathcal{O}_s$

Assume rank$(Y_{0,s,N}) = n$.

Singular value decomposition (SVD)

$$Y_{0,s,N} = \begin{bmatrix} U_n & U_n^{\perp} \end{bmatrix} \begin{bmatrix} \Sigma_n & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} (V_n)^T & (V_n^{\perp})^T \end{bmatrix}$$

with $\Sigma_n \in \mathbb{R}^{n \times n}$ and rank$(\Sigma_n) = n$.

range(U_n) = range$(Y_{0,s,N})$ = range$(\mathcal{O}_s)$

Thus: $\boxed{U_n = \mathcal{O}_s T}$ T non-singular!

Compute the matrices A and C

$$U_n = \mathcal{O}_s T = \begin{bmatrix} CT \\ CTT^{-1}AT \\ \vdots \\ CT\left(T^{-1}AT\right)^{s-1} \end{bmatrix} = \begin{bmatrix} C_T \\ C_T A_T \\ \vdots \\ C_T A_T^{s-1} \end{bmatrix}$$

Compute the matrices A and C

$$U_n = \mathcal{O}_s T = \begin{bmatrix} CT \\ CTT^{-1}AT \\ \vdots \\ CT\left(T^{-1}AT\right)^{s-1} \end{bmatrix} = \begin{bmatrix} C_T \\ C_T A_T \\ \vdots \\ C_T A_T^{s-1} \end{bmatrix}$$

We can determine A and C up to an unknown similarity transformation T.

Compute the matrices A and C

$$U_n = \mathcal{O}_s T = \begin{bmatrix} CT \\ CTT^{-1}AT \\ \vdots \\ CT\left(T^{-1}AT\right)^{s-1} \end{bmatrix} = \begin{bmatrix} C_T \\ C_T A_T \\ \vdots \\ C_T A_T^{s-1} \end{bmatrix}$$

We can determine A and C up to an unknown similarity transformation T.

C_T equals the first ℓ rows of U_n: $\quad C_T = U_n(1:\ell,:)$

A_T is computed by solving:

$$U_n(1:(s-1)\ell,:)A_T = U_n(\ell+1:s\ell,:)$$

Overview

- Recap “classical” system identification
- A different approach: Subspace Identification
  - A deterministic Identification problem
  - Subspace Identification of an autonomous system
  - **Subspace Identification with inputs**

Identification with inputs

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

Identification with inputs

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

$$\begin{aligned} x(1) &= Ax(0) + Bu(0) \\ x(2) &= Ax(1) + Bu(1) \\ &= A^2x(0) + ABu(0) + Bu(1) \\ x(3) &= A^3x(0) + A^2Bu(0) + ABu(1) + Bu(2) \end{aligned}$$

Identification with inputs

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

$$\begin{aligned} x(1) &= Ax(0) + Bu(0) \\ x(2) &= Ax(1) + Bu(1) \\ &= A^2x(0) + ABu(0) + Bu(1) \\ x(3) &= A^3x(0) + A^2Bu(0) + ABu(1) + Bu(2) \end{aligned}$$

State at time k: $x(k) = A^k x(0) + \sum_{i=0}^{k-1} A^{k-i-1} B u(i)$

Data equation with inputs

Output at time k:

$$y(k) = CA^k x(0) + \sum_{i=0}^{k-1} CA^{k-i-1} B u(i) + D u(k)$$

$$\begin{bmatrix} y(0) \\ y(1) \\ y(2) \\ \vdots \\ y(s-1) \end{bmatrix} = \underbrace{\begin{bmatrix} C \\ CA \\ CA^2 \\ \vdots \\ CA^{s-1} \end{bmatrix}}_{\mathcal{O}_s} x(0) + \underbrace{\begin{bmatrix} D & 0 & 0 & \cdots & 0 \\ CB & D & 0 & \cdots & 0 \\ CAB & CB & D & & 0 \\ \vdots & & \ddots & \ddots & \\ CA^{s-2}B & CA^{s-3}B & \cdots & CB & D \end{bmatrix}}_{\mathcal{T}_s} \begin{bmatrix} u(0) \\ u(1) \\ u(2) \\ \vdots \\ u(s-1) \end{bmatrix}$$

Data equation with inputs

Data equation $$Y_{0,s,N} = \mathcal{O}_s X_{0,N} + \mathcal{T}_s U_{0,s,N}$$

Strategy:

- Determine A and C (up to a similarity transformation) from the column space of $\mathcal{O}_s$.
- Given A and C determine B and D from the input and output measurements.

Data equation with inputs

Data equation $$Y_{0,s,N} = \mathcal{O}_s X_{0,N} + \mathcal{T}_s U_{0,s,N}$$

Strategy:

- Determine A and C (up to a similarity transformation) from the column space of $\mathcal{O}_s$.
- Given A and C determine B and D from the input and output measurements.

How to estimate the column space of $\mathcal{O}_s$?

Removing the input part in the data equation

Let $\Pi^{\perp}_{U_{0,s,N}}$ be the **Projection matrix** onto the orthogonal complement of the row space of $U_{0,s,N}$. This matrix is given as:

$$\Pi^{\perp}_{U_{0,s,N}} = \left(I_N - U^T_{0,s,N} (U_{0,s,N} U^T_{0,s,N})^{-1} U_{0,s,N} \right) \in \mathbb{R}^{N \times N}$$

Removing the input part in the data equation

Let $\Pi^{\perp}_{U_{0,s,N}}$ be the **Projection matrix** onto the orthogonal complement of the row space of $U_{0,s,N}$. This matrix is given as:

$$\Pi^{\perp}_{U_{0,s,N}} = \left(I_N - U^T_{0,s,N} (U_{0,s,N} U^T_{0,s,N})^{-1} U_{0,s,N} \right) \in \mathbb{R}^{N \times N}$$

Then: $U_{0,s,N} \Pi^{\perp}_{U_{0,s,N}} = 0$.

Removing the input part in the data equation

Let $\Pi^{\perp}_{U_{0,s,N}}$ be the **Projection matrix** onto the orthogonal complement of the row space of $U_{0,s,N}$. This matrix is given as:

$$\Pi^{\perp}_{U_{0,s,N}} = \left(I_N - U^T_{0,s,N} (U_{0,s,N} U^T_{0,s,N})^{-1} U_{0,s,N} \right) \in \mathbb{R}^{N \times N}$$

Then: $U_{0,s,N} \Pi^{\perp}_{U_{0,s,N}} = 0$.

$$\begin{aligned} Y_{0,s,N} \Pi^{\perp}_{U_{0,s,N}} &= \left(\mathcal{O}_s X_{0,N} + \mathcal{T}_s U_{0,s,N} \right) \Pi^{\perp}_{U_{0,s,N}} \\ &= \mathcal{O}_s X_{0,N} \Pi^{\perp}_{U_{0,s,N}} \end{aligned}$$

Dealing with inputs

$$Y_{0,s,N}\Pi^{\perp}_{U_{0,s,N}} = \mathcal{O}_s X_{0,N}\Pi^{\perp}_{U_{0,s,N}}$$

$$\text{range}(Y_{0,s,N}\Pi^{\perp}_{U_{0,s,N}}) \subseteq \text{range}(\mathcal{O}_s)$$

Dealing with inputs

$$Y_{0,s,N}\Pi^{\perp}_{U_{0,s,N}} = \mathcal{O}_s X_{0,N}\Pi^{\perp}_{U_{0,s,N}}$$

$$\text{range}(Y_{0,s,N}\Pi^{\perp}_{U_{0,s,N}}) \subseteq \text{range}(\mathcal{O}_s)$$

Lemma:

If $u(k)$ is such that rank $\left(\begin{bmatrix} X_{0,N} \\ U_{0,s,N} \end{bmatrix}\right) = n + sm$

and (A, C) observable $\Rightarrow$ rank $\left(Y_{0,s,N}\Pi^{\perp}_{U_{0,s,N}}\right) = n$

$$\boxed{\text{range}(Y_{0,s,N}\Pi^{\perp}_{U_{0,s,N}}) = \text{range}(\mathcal{O}_s)} \Rightarrow A_T, C_T$$

RQ for efficient implementation

For efficient implementation it is desirable **not to compute** the product $Y_{0,s,N}\Pi^{\perp}_{U_{0,s,N}}$ explicitly.

$$\begin{bmatrix} U_{0,s,N} \\ Y_{0,s,N} \end{bmatrix} = \begin{bmatrix} R_{11} & 0 & 0 \\ R_{21} & R_{22} & 0 \end{bmatrix} \begin{bmatrix} Q_1 \\ Q_2 \\ Q_3 \end{bmatrix} \Rightarrow \Pi^{\perp}_{U_{0,s,N}} = \left(I_N - Q_1^T Q_1\right)$$

Lemma: Given the **RQ factorization**, we have:
$Y_{0,s,N}\Pi^{\perp}_{U_{0,s,N}} = R_{22}Q_2$.

RQ for efficient implementation

For efficient implementation it is desirable **not to compute** the product $Y_{0,s,N}\Pi^{\perp}_{U_{0,s,N}}$ explicitly.

$$\begin{bmatrix} U_{0,s,N} \\ Y_{0,s,N} \end{bmatrix} = \begin{bmatrix} R_{11} & 0 & 0 \\ R_{21} & R_{22} & 0 \end{bmatrix} \begin{bmatrix} Q_1 \\ Q_2 \\ Q_3 \end{bmatrix} \Rightarrow \Pi^{\perp}_{U_{0,s,N}} = \left(I_N - Q_1^T Q_1\right)$$

Lemma: Given the **RQ factorization**, we have:
$Y_{0,s,N}\Pi^{\perp}_{U_{0,s,N}} = R_{22}Q_2$.

range$(\mathcal{O}_s)$ = range$(Y_{0,s,N}\Pi^{\perp}_{U_{0,s,N}})$ = range(R_{22})

Determine B and D

$$y(k) = C_T A_T^k x(0) + \sum_{i=0}^{k-1} C_T A_T^{k-i-1} B_T u(i) + D_T u(k)$$

Given A_T and C_T, **the output depends affinely** on B_T, D_T and $x(0)$.

Determine B and D

$$y(k) = C_T A_T^k x(0) + \sum_{i=0}^{k-1} C_T A_T^{k-i-1} B_T u(i) + D_T u(k)$$

Given A_T and C_T, **the output depends affinely** on B_T, D_T and $x(0)$.

Using property (§2.3) $\boxed{\text{vec}(XYZ) = (Z^T \otimes X)\text{vec}(Y)}$,

$$\begin{aligned} y(k) &= C_T A_T^k x(0) + \sum_{i=0}^{k-1} (u(i)^T \otimes C_T A_T^{k-i-1})\text{vec}(B_T) + (u(k)^T \otimes I)\text{vec}(D_T) \\ &= F(k, A_T, C_T)\theta \qquad \theta = \begin{bmatrix} x(0)^T & \text{vec}(B_T)^T & \text{vec}(D_T)^T \end{bmatrix}^T \end{aligned}$$

Find θ by solving a *Linear least squares problem!*

$$\min_{\theta} \frac{1}{N} \sum_{k=0}^{N-1} ||y(k) - F(k, A_T, C_T)^T \theta||^2$$

Next lecture: Consistency analysis