On the importance of low-dimensional structures for data-driven modeling

Jean-Christophe Loiseau

May, 13th 2022

- Maître de Conférences in Fluid Dynamics and Applied Math.
- Machine-learning enthusiast with application to engineering systems.
- Data-efficient models with guarantees of optimality or interpretability.

Given training data $\boldsymbol{y}_i = \boldsymbol{f}(\boldsymbol{x}_i)$ where $\boldsymbol{f}$ is unknown, we want to **learn** a function $\hat{\boldsymbol{f}}$ such that

$$\underset{\hat{\boldsymbol{f}} \in \mathcal{F}}{\text{minimize}} \quad \|\boldsymbol{f}(\boldsymbol{x}) - \hat{\boldsymbol{f}}(\boldsymbol{x})\|$$

where $\mathcal{F}$ is a particular *class* of models. Here, we'll restrict our attention to the class of *linear* models.

A (very) brief overview of SVD

$$\boldsymbol{A} = \boldsymbol{U} \; \boldsymbol{\Sigma} \; \boldsymbol{V}^{T}$$

Basis for colspan($\boldsymbol{A}$)

Basis for rowspan($\boldsymbol{A}$)

$$\boldsymbol{A} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T$$

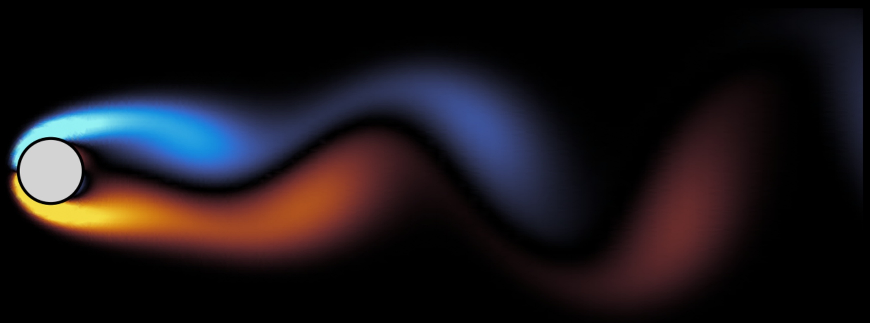
Diagonal matrix

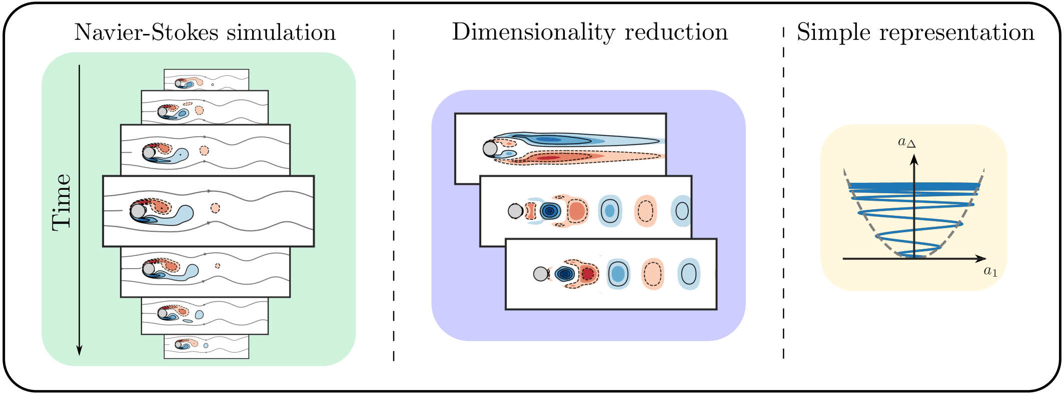
Relation to spectral decomposition

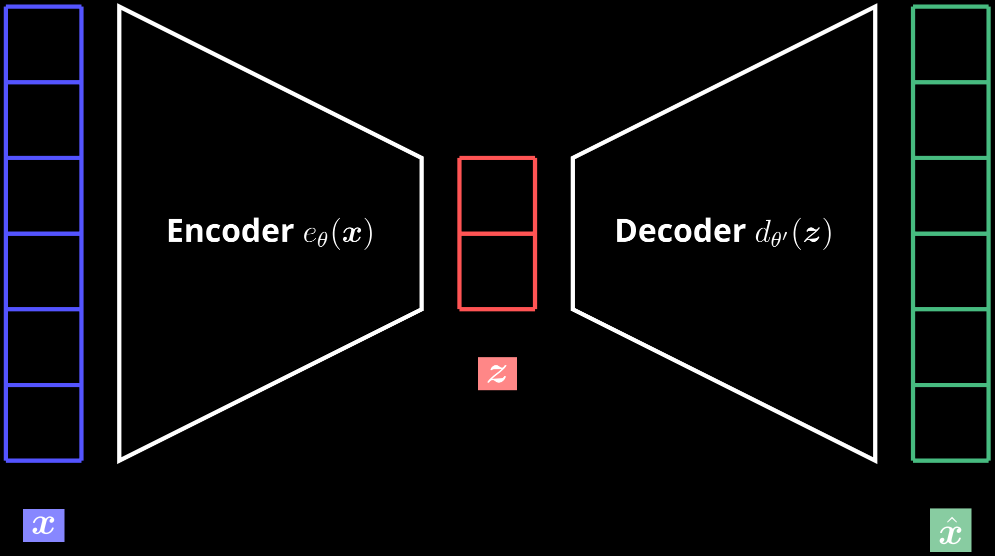
$$\begin{bmatrix} \mathbf{0} & \boldsymbol{A} \\ \boldsymbol{A}^T & \mathbf{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{u}_i \\ \boldsymbol{v}_i \end{bmatrix} = \sigma_i \begin{bmatrix} \boldsymbol{u}_i \\ \boldsymbol{v}_i \end{bmatrix}$$

Generalization of the *eigenvalue decomposition* to **non-square matrices** by E. Beltrami (1873) and C. Jordan (1874). The first efficient numerical algorithm was developed by G. Golub *et al.* in the late 1960s.

Extracting coherent patterns




Navier-Stokes simulation
Time
Dimensionality reduction
Simple representation
a_Δ
a_1


Encoder $e_\theta(x)$
Decoder $d_{\theta'}(z)$
z
x
$\hat{x}$

Estimate

$$\min_{\theta,\theta'} \sum_{i=1}^{N} \| \boldsymbol{x}_i - (d_{\theta'} \circ e_\theta)(\boldsymbol{x}_i) \|_2^2$$

Ground truth

$$
\begin{aligned}
\underset{\boldsymbol{P},\boldsymbol{Q}}{\text{minimize}} \quad & \sum_{i=1}^{N} \|\boldsymbol{x}_i - \boldsymbol{P}\boldsymbol{Q}^T\boldsymbol{x}_i\|_2^2 \\
\text{subject to} \quad & \operatorname{rank} \boldsymbol{P} = \operatorname{rank} \boldsymbol{Q} = r
\end{aligned}
$$

$$
\begin{aligned}
\underset{\boldsymbol{P}}{\text{minimize}} \quad & \sum_{i=1}^{N} \|\boldsymbol{x}_i - \boldsymbol{P}\boldsymbol{P}^T \boldsymbol{x}_i\|_2^2 \\
\text{subject to} \quad & \operatorname{rank} \boldsymbol{P} = r
\end{aligned}
$$

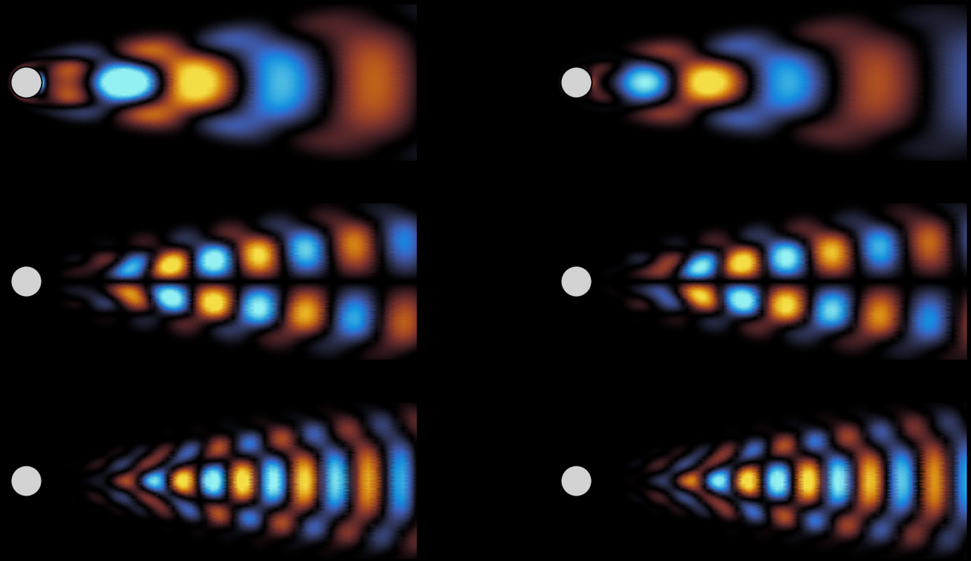
$$
\begin{aligned}
\underset{\boldsymbol{P}}{\text{minimize}} \quad & \|\boldsymbol{X} - \boldsymbol{P}\boldsymbol{P}^T\boldsymbol{X}\|_F^2 \\
\text{subject to} \quad & \boldsymbol{P}^T\boldsymbol{P} = \boldsymbol{I}_r
\end{aligned}
$$

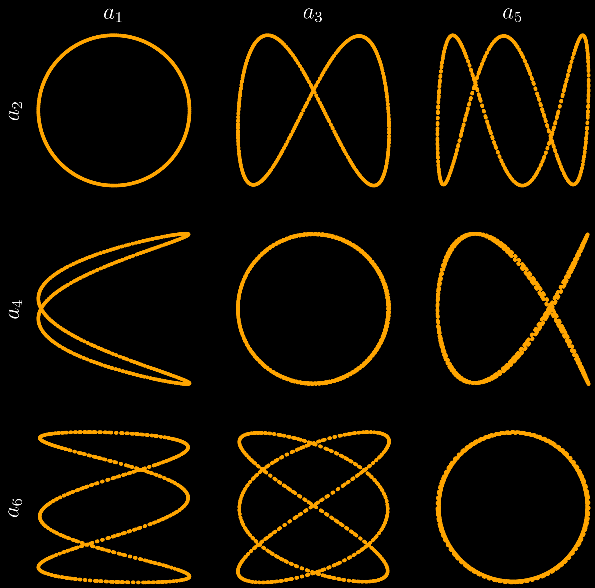
Proper Orthogonal Decomposition

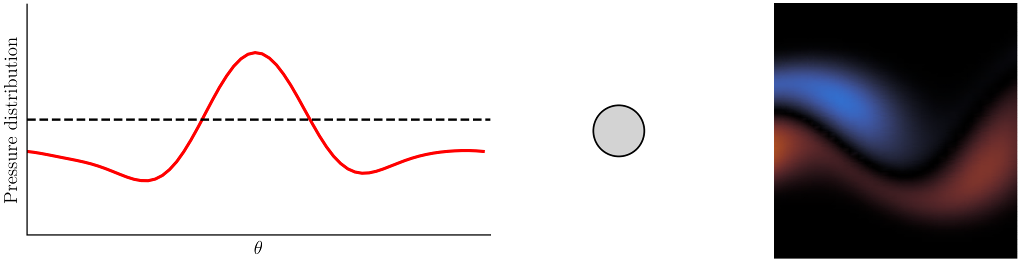
$$\boldsymbol{P}\boldsymbol{\Lambda} = \boldsymbol{C}_{\boldsymbol{xx}}\boldsymbol{P}$$

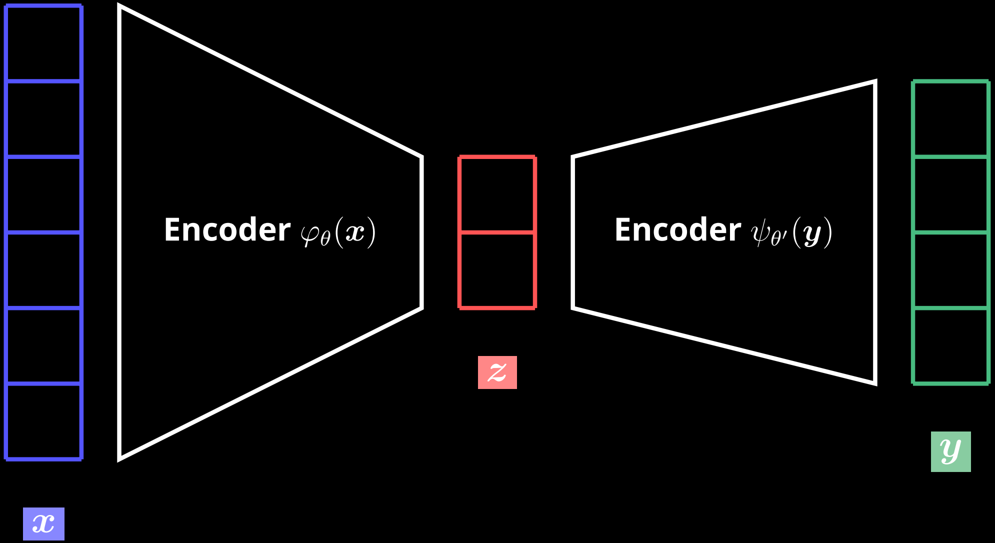
$\boldsymbol{P}$ corresponds to the left singular vectors of $\boldsymbol{X}$. The latent representation is given by $\boldsymbol{z}_i = \boldsymbol{P}^T\boldsymbol{x}_i$. The optimal rank of the model can be inferred from the distribution of the PCA eigenvalues $\boldsymbol{\Lambda} = \boldsymbol{\Sigma}^2$.

The whole dataset can be correctly approximated using only 500 so-called eigen-faces.




a_1
a_3
a_5
a_2
a_4
a_6


Pressure distribution
θ


Encoder $\varphi_\theta(x)$
Encoder $\psi_{\theta'}(y)$
z
x
y

$$\min_{\theta,\theta'} \quad \sum_{i=1}^{N} \|\varphi_\theta(\boldsymbol{x}_i) - \psi_{\theta'}(\boldsymbol{y}_i)\|_2^2$$

$$
\begin{aligned}
\underset{\boldsymbol{P},\boldsymbol{Q}}{\text{minimize}} \quad & \sum_{i=1}^{N} \|\boldsymbol{P}^T \boldsymbol{y}_i - \boldsymbol{Q}^T \boldsymbol{x}_i\|_2^2 \\
\text{subject to} \quad & \operatorname{rank} \boldsymbol{P} = \operatorname{rank} \boldsymbol{Q} = r
\end{aligned}
$$

$$\begin{aligned} \underset{\boldsymbol{P},\boldsymbol{Q}}{\text{minimize}} \quad & \|\boldsymbol{P}^T\boldsymbol{Y} - \boldsymbol{Q}^T\boldsymbol{X}\|_F^2 \\ \text{subject to} \quad & \boldsymbol{P}^T\boldsymbol{C}_{yy}\boldsymbol{P} = \boldsymbol{Q}^T\boldsymbol{C}_{xx}\boldsymbol{Q} = \boldsymbol{I}_r \end{aligned}$$

Canonical Correlation Analysis

$$\begin{bmatrix} C_{yy} & 0 \\ 0 & C_{xx} \end{bmatrix} \begin{bmatrix} P \\ Q \end{bmatrix} \Sigma = \begin{bmatrix} 0 & C_{yx} \\ C_{xy} & 0 \end{bmatrix} \begin{bmatrix} P \\ Q \end{bmatrix}$$

CCA relies on a *generalized eigenproblem*. P and Q describe the encoders such that the latent representations $z = Q^T x$ and $z' = P^T Y$ are as similar as possible. It is closely related to the concept of *mutual information*.

CCA-Encoding

$$\boldsymbol{z} = \boldsymbol{P}^T \boldsymbol{x}$$
$$\boldsymbol{z}' = \boldsymbol{Q}^T \boldsymbol{y}$$

CCA-Encoding

$$z = P^T x$$
$$z' = Q^T y$$

CCA-Latent space

$$z \simeq z'$$

CCA-Encoding

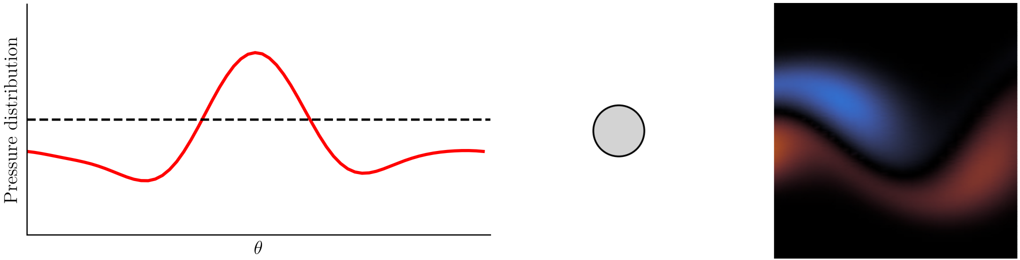
$$z = P^T x$$
$$z' = Q^T y$$

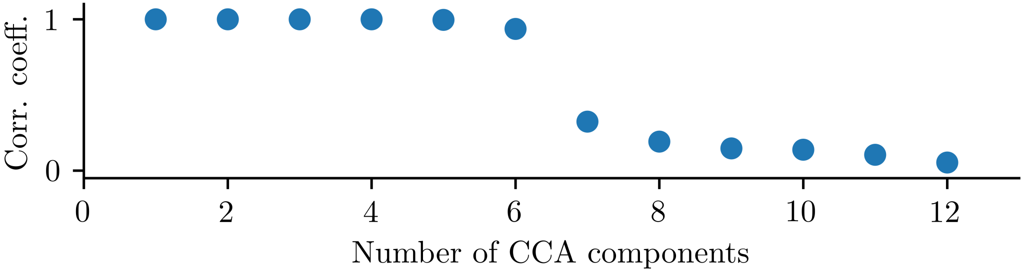
CCA-Latent space

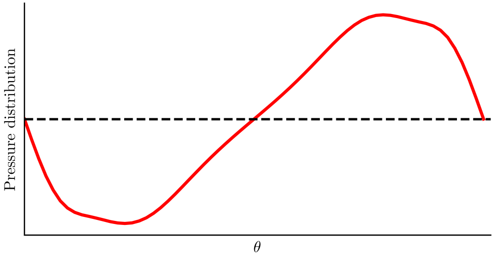
$$z \simeq z'$$

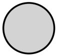
CCA-Decoding

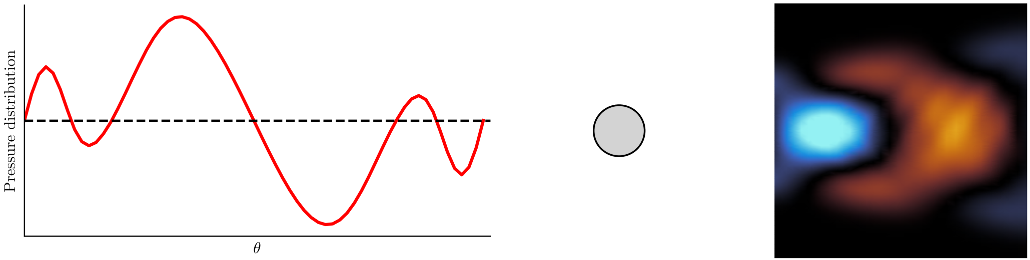
$$\hat{x} = P\left(P^T P\right)^{-1} z$$
$$\hat{y} = Q\left(Q^T Q\right)^{-1} z'$$

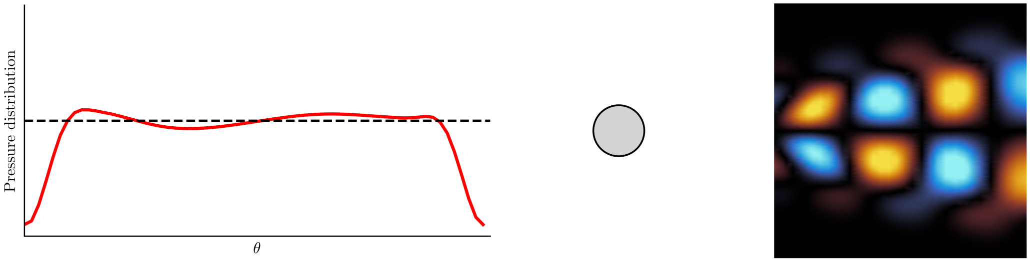

Pressure distribution
θ

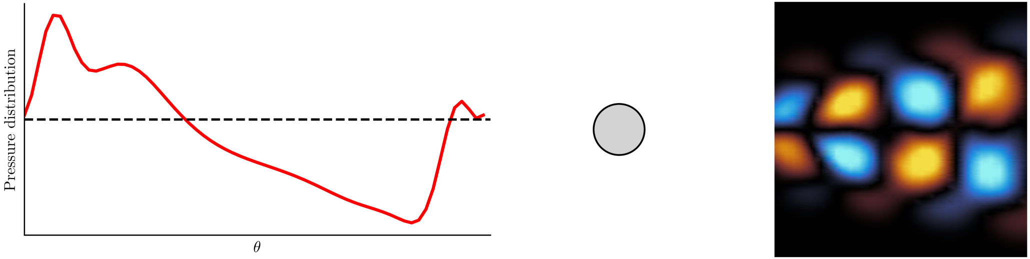

Corr. coeff.
1
0
0
2
4
6
8
10
12
Number of CCA components

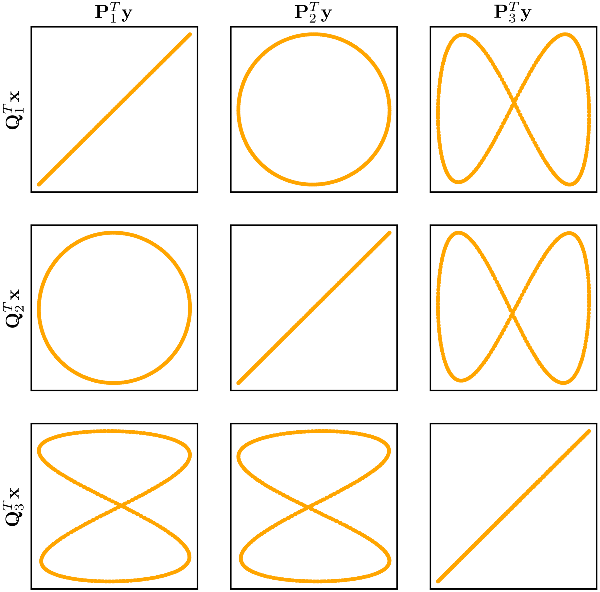

Pressure distribution
θ



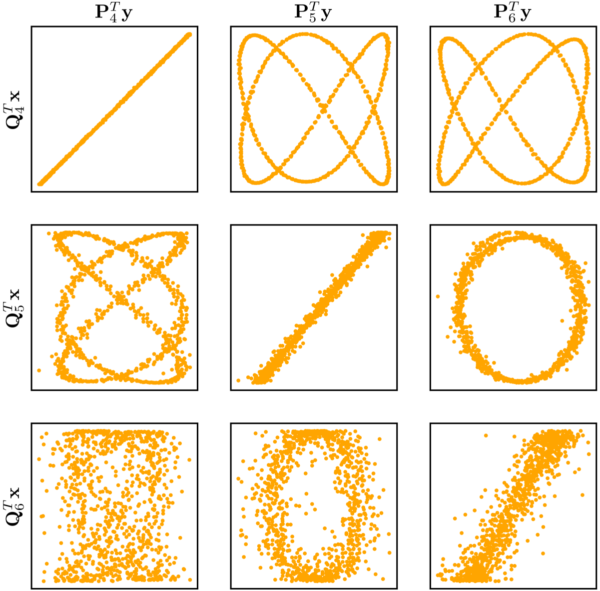

Pressure distribution
θ


Pressure distribution
θ


Pressure distribution
θ



$\mathbf{P}_1^T\mathbf{y}$
$\mathbf{P}_2^T\mathbf{y}$
$\mathbf{P}_3^T\mathbf{y}$
$\mathbf{Q}_1^T\mathbf{x}$
$\mathbf{Q}_2^T\mathbf{x}$
$\mathbf{Q}_3^T\mathbf{x}$



$\mathbf{P}_4^T\mathbf{y}$
$\mathbf{P}_5^T\mathbf{y}$
$\mathbf{P}_6^T\mathbf{y}$
$\mathbf{Q}_4^T\mathbf{x}$
$\mathbf{Q}_5^T\mathbf{x}$
$\mathbf{Q}_6^T\mathbf{x}$

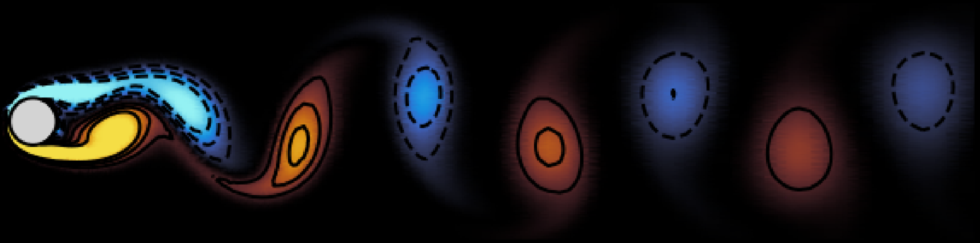
- Relies on extremely efficient numerical linear algebra techniques.
- The singular value distribution provides a simple diagnostic to estimate the dimensional of the embedding Euclidean space.
- Component analysis is well understood from a statistical point of view and benefits from good properties.

+

- The dimension inferred from the singular value distribution tends to over-estimate the intrinsic dimension of the underlying low-dimensional manifold.

- This is directly related to the fact the linear dimensionality reduction techniques do not account for *nonlinear correlations*, only linear ones..

Optimal sensor placement

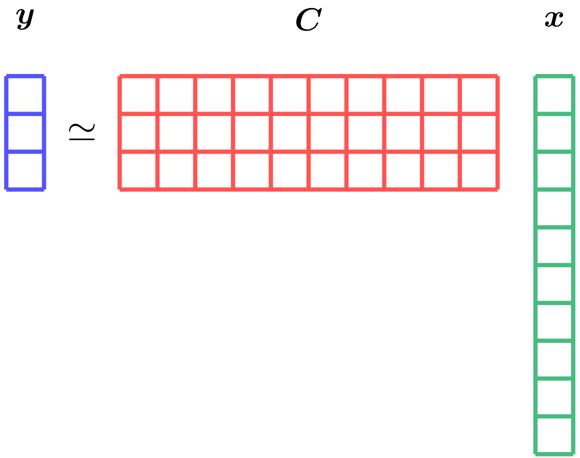


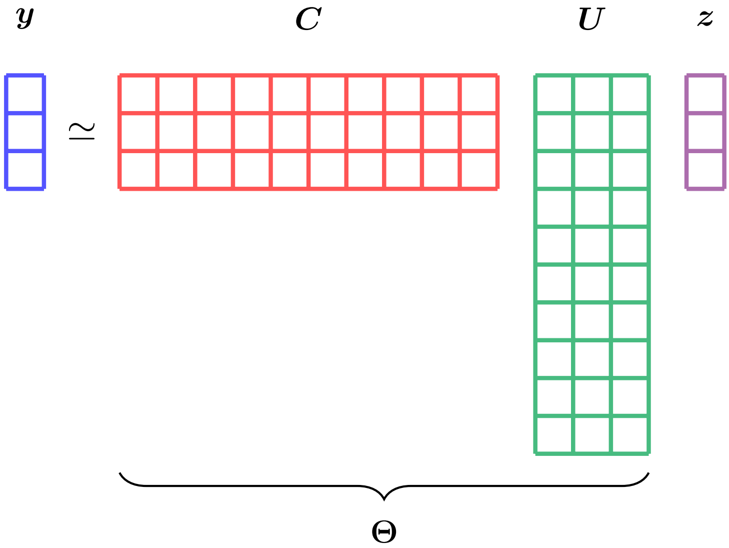
Measurement operator

$$y = Cx$$

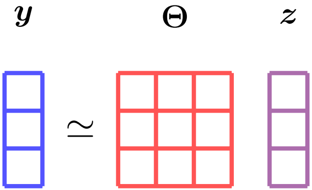
Full state

Observations


y
≃
C
x



$\boldsymbol{y}$
$\boldsymbol{C}$
$\boldsymbol{U}$
$\boldsymbol{z}$
$\simeq$
$\boldsymbol{\Theta}$


$\boldsymbol{y}$
$\boldsymbol{\Theta}$
$\boldsymbol{z}$
$\simeq$

$$\underset{\boldsymbol{z}}{\text{minimize}} \quad \|\boldsymbol{y} - \boldsymbol{\Theta}\boldsymbol{z}\|_2$$

$$\boldsymbol{z} = \boldsymbol{\Theta}^{-1}\boldsymbol{y}$$

$$\underset{\boldsymbol{C}}{\text{maximize}} \quad |\det(\boldsymbol{C}\boldsymbol{U})|$$

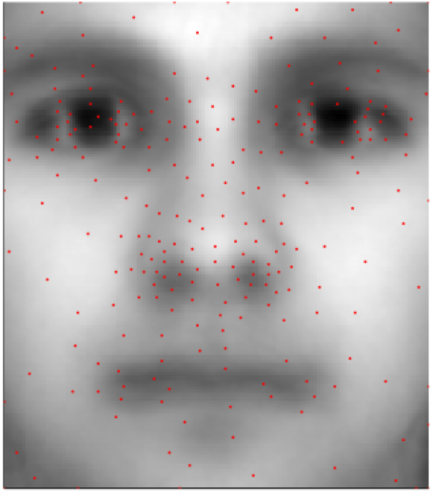
$$\begin{aligned} \underset{\boldsymbol{C}}{\text{maximize}} \quad & |\det(\boldsymbol{C}\boldsymbol{U})| \\ \text{subject to} \quad & \boldsymbol{C}_i \in \{\boldsymbol{e}_j\}_{j=1,n} \end{aligned}$$

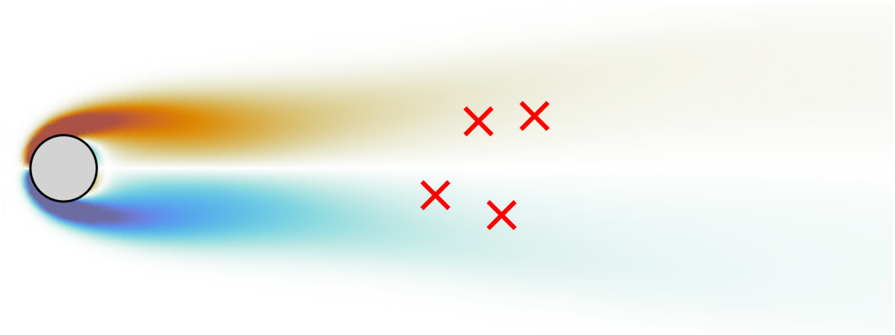
Permutation matrix

$$\boldsymbol{U}^T \boldsymbol{P} = \boldsymbol{Q}\boldsymbol{R}$$

Low-rank basis

Upper triangular matrix with $|r_{i-1}| \geq |r_i|$





- Relies on extremely efficient numerical linear algebra techniques.
- Greedy solution to an otherwise intractable combinatorial problem.
- Relevant in many situations and benefits from good empirical performances.

+

- No easy way to estimate how far from the optimum the greedy solution is.

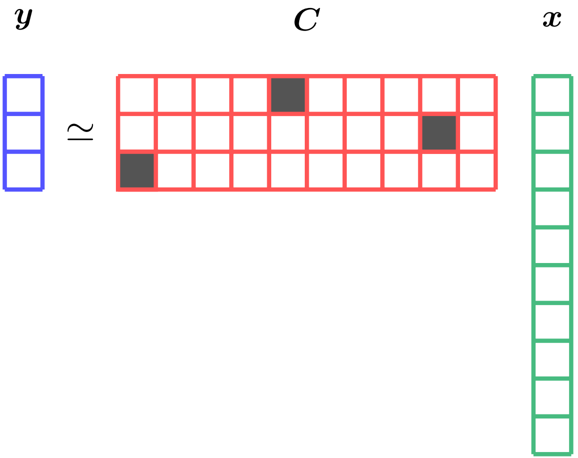
State estimation and low-rank sensing

Measurement operator

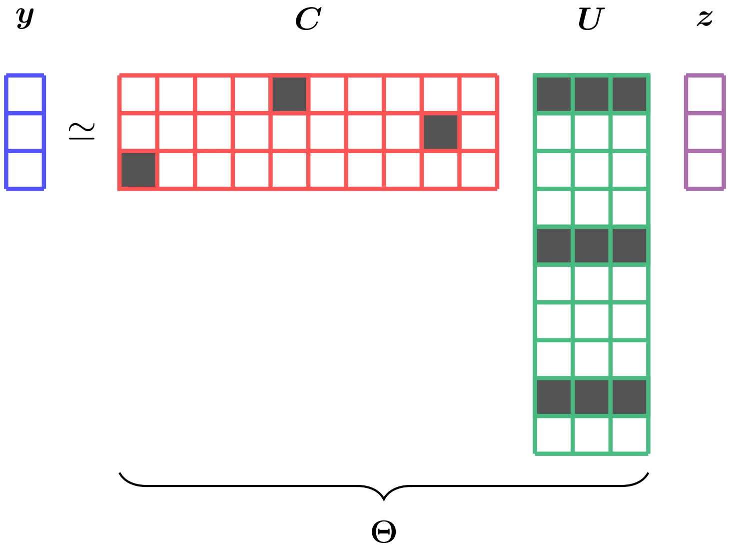
$$y = Cx$$

Full state

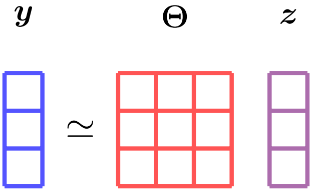
Observations



y
$\simeq$
C
x



$\boldsymbol{y}$
$\simeq$
$\boldsymbol{C}$
$\boldsymbol{U}$
$\boldsymbol{z}$
$\boldsymbol{\Theta}$


$\boldsymbol{y}$
$\simeq$
$\boldsymbol{\Theta}$
$\boldsymbol{z}$

Underdetermined problem

$$\begin{aligned} \underset{\boldsymbol{z}}{\text{minimize}} \quad & \|\boldsymbol{z}\|_2 \\ \text{subject to} \quad & \boldsymbol{y} = \boldsymbol{\Theta} \boldsymbol{z} \end{aligned}$$

Overdetermined problem

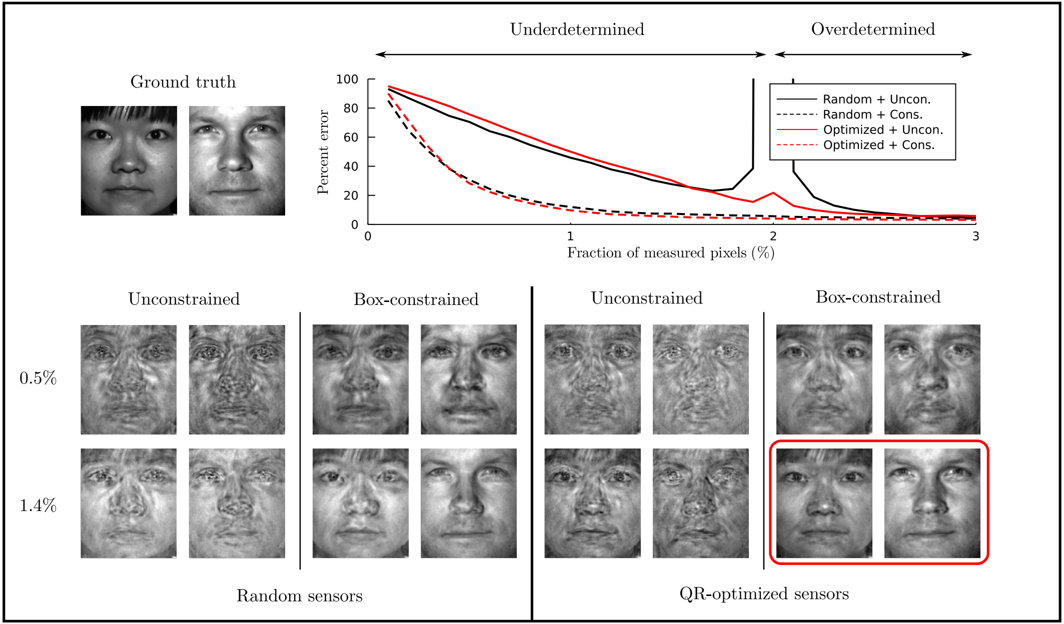
$$\underset{\boldsymbol{z}}{\text{minimize}} \quad \|\boldsymbol{y} - \boldsymbol{\Theta} \boldsymbol{z}\|_2^2$$

Regularized problem

$$\underset{\boldsymbol{z}}{\text{minimize}} \quad \|\boldsymbol{y} - \boldsymbol{\Theta}\boldsymbol{z}\|_2^2 + \lambda\|\boldsymbol{z}\|_2^2$$

Regularized and constrained problem

$$\begin{aligned} \underset{\boldsymbol{z}}{\text{minimize}} \quad & \|\boldsymbol{y} - \boldsymbol{\Theta}\boldsymbol{z}\|_2^2 + \lambda\|\boldsymbol{z}\|_2^2 \\ \text{subject to} \quad & |z_i| \leq 2\sigma_i \quad \forall\, i \end{aligned}$$



Underdetermined
Overdetermined
Ground truth
Percent error
Fraction of measured pixels (%)
Random + Uncon.
Random + Cons.
Optimized + Uncon.
Optimized + Cons.
Unconstrained
Box-constrained
Unconstrained
Box-constrained
0.5%
1.4%
Random sensors
QR-optimized sensors

- Low-rank sensing benefits from strong theoretical guarantees.
- Relies on simple yet efficient numerical procedures, both during the training and deployment stages.
- Requires much less data than naïve deep learning alternatives.

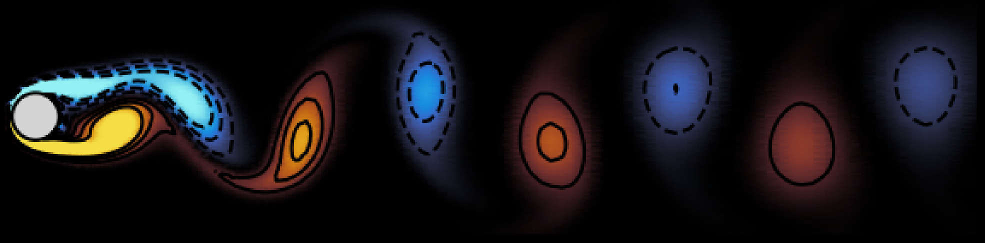
+

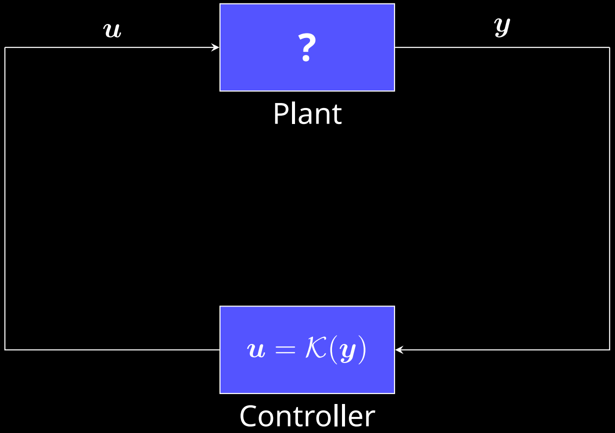
- Data needs to be standardized and characterized by an underlying low-rank structure.

- The state estimator is a **static map**. It does not account for the dynamics (if they exist) of the generating process.

- The measurement equation needs to be (approximately) linear.

System identification




u
?
Plant
y
$u = \mathcal{K}(y)$
Controller

Convolution model

$$\boldsymbol{y}_i = \boldsymbol{f}\left(\boldsymbol{u}_i, \boldsymbol{u}_{i-1}, \boldsymbol{u}_{i-2}, \cdots\right)$$

Convolution model

$$\boldsymbol{y}(t) = \boldsymbol{h}(t) * \boldsymbol{u}(t)$$

Output

Input

Unknown conv. kernel

Natural dynamics

Actuators

$$\boldsymbol{x}_{i+1} = \boxed{\boldsymbol{A}}\,\boldsymbol{x}_i + \boxed{\boldsymbol{B}}\,\boldsymbol{u}_i$$
$$\boldsymbol{y}_i = \boxed{\boldsymbol{C}}\,\boldsymbol{x}_i + \boxed{\boldsymbol{D}}\,\boldsymbol{u}_i$$

Measurements

Feedthrough

Input $\boldsymbol{u}_i$	State $\boldsymbol{x}_i$	Output $\boldsymbol{y}_i$
$\boldsymbol{u}_0$	$\boldsymbol{0}$	$\boldsymbol{D}\boldsymbol{u}_0$
$\boldsymbol{u}_1$	$\boldsymbol{B}\boldsymbol{u}_0$	$\boldsymbol{C}\boldsymbol{B}\boldsymbol{u}_0 + \boldsymbol{D}\boldsymbol{u}_1$
$\boldsymbol{u}_2$	$\boldsymbol{A}\boldsymbol{B}\boldsymbol{u}_0 + \boldsymbol{B}\boldsymbol{u}_1$	$\boldsymbol{C}\boldsymbol{A}\boldsymbol{B}\boldsymbol{u}_0 + \boldsymbol{C}\boldsymbol{B}\boldsymbol{u}_1 + \boldsymbol{D}\boldsymbol{u}_2$
$\boldsymbol{u}_3$	$\boldsymbol{A}^2\boldsymbol{B}\boldsymbol{u}_0 + \boldsymbol{A}\boldsymbol{B}\boldsymbol{u}_1 + \boldsymbol{B}\boldsymbol{u}_2$	$\boldsymbol{C}\boldsymbol{A}^2\boldsymbol{B}\boldsymbol{u}_0 + \boldsymbol{C}\boldsymbol{A}\boldsymbol{B}\boldsymbol{u}_1 + \boldsymbol{C}\boldsymbol{B}\boldsymbol{u}_2 + \boldsymbol{D}\boldsymbol{u}_3$
$\vdots$	$\vdots$	$\vdots$
$\boldsymbol{u}_k$	$\displaystyle\sum_{i=1}^{k} \boldsymbol{A}^{i-1}\boldsymbol{B}\boldsymbol{u}_{k-i}$	$\displaystyle\sum_{i=1}^{k} \boldsymbol{C}\boldsymbol{A}^{i-1}\boldsymbol{B}\boldsymbol{u}_{k-i} + \boldsymbol{D}\boldsymbol{u}_k$

$$\boldsymbol{y}_k = \boldsymbol{D}\boldsymbol{u}_k + \sum_{i=1}^{k} \boldsymbol{C}\boldsymbol{A}^{i-1}\boldsymbol{B}\boldsymbol{u}_{k-i}$$

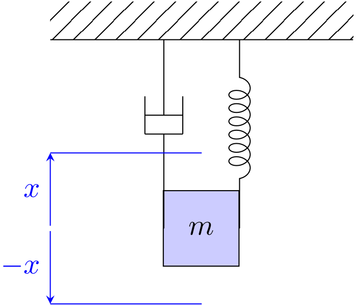
$$
\begin{bmatrix} \boldsymbol{y}_0 \\ \boldsymbol{y}_1 \\ \boldsymbol{y}_2 \\ \boldsymbol{y}_3 \\ \vdots \end{bmatrix} = \begin{bmatrix} \boldsymbol{u}_0 & \cdots & & & \\ \boldsymbol{u}_1 & \boldsymbol{u}_0 & \cdots & & \\ \boldsymbol{u}_2 & \boldsymbol{u}_1 & \boldsymbol{u}_0 & \cdots & \\ \boldsymbol{u}_3 & \boldsymbol{u}_2 & \boldsymbol{u}_1 & \boldsymbol{u}_0 & \cdots \\ \vdots & \ddots & \ddots & \ddots & \ddots \end{bmatrix} \begin{bmatrix} \boldsymbol{D} \\ \boldsymbol{C}\boldsymbol{B} \\ \boldsymbol{C}\boldsymbol{A}\boldsymbol{B} \\ \boldsymbol{C}\boldsymbol{A}^2\boldsymbol{B} \\ \boldsymbol{C}\boldsymbol{A}^3\boldsymbol{B} \\ \vdots \end{bmatrix}
$$

Output sequence

Unknown conv. kernel

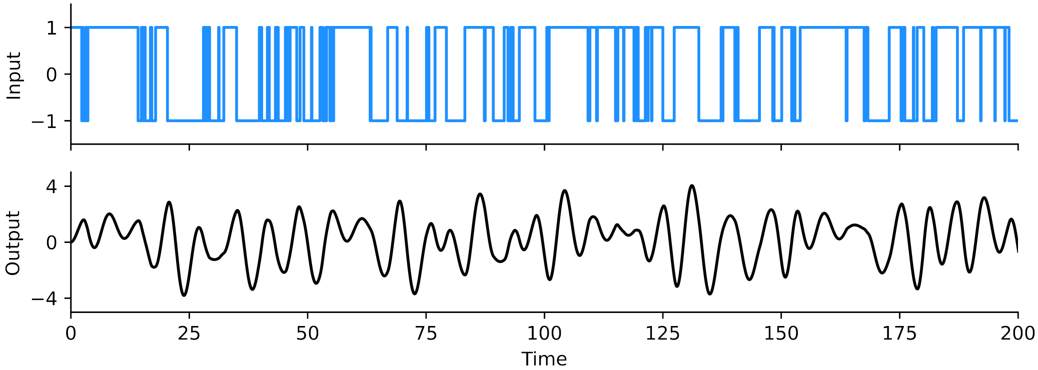
$$\underset{\boldsymbol{h}}{\text{minimize}} \quad \|\boldsymbol{y} - \mathcal{T}(\boldsymbol{u})\,\boldsymbol{h}\|_p + \mathcal{R}(\boldsymbol{h})$$

Toeplitz matrix built
using the input sequence



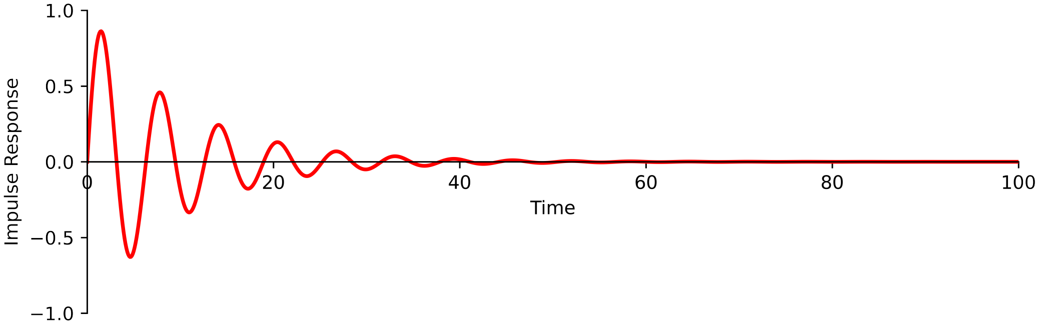
$$\frac{\mathrm{d}}{\mathrm{d}t}\begin{bmatrix} x \\ v \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -1 & -2\zeta \end{bmatrix}\begin{bmatrix} x \\ v \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u$$

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



Input/Output data for system identification

$$\underset{\boldsymbol{h}}{\text{minimize}} \quad \|\boldsymbol{y} - \boldsymbol{h} * \boldsymbol{u}\|_2^2$$



- Relies on simple linear regression.
- Easy to extend to nonlinear systems (e.g. LSTM)
- Prior knowledge can be included through regularization or constraints.

+

- The length of the convolution kernel is *a priori* unknown.
- The longer the kernel, the more data is needed to obtain converged estimates.
- It is *a priori* unrelated to the number of degrees of freedom in the system.

Natural dynamics

Actuators

$$
\begin{aligned}
\boldsymbol{x}_{i+1} &= \boldsymbol{A}\boldsymbol{x}_i + \boldsymbol{B}\boldsymbol{u}_i \\
\boldsymbol{y}_i &= \boldsymbol{C}\boldsymbol{x}_i + \boldsymbol{D}\boldsymbol{u}_i
\end{aligned}
$$

Measurements

Feedthrough

$$\mathcal{O}_k = \begin{bmatrix} \boldsymbol{C} \\ \boldsymbol{CA} \\ \boldsymbol{CA}^2 \\ \boldsymbol{CA}^3 \\ \vdots \\ \boldsymbol{CA}^{k-1} \end{bmatrix} \qquad \mathcal{C}_k = \begin{bmatrix} \boldsymbol{B} & \boldsymbol{AB} & \boldsymbol{A}^2\boldsymbol{B} & \boldsymbol{A}^3\boldsymbol{B} & \cdots & \boldsymbol{A}^{k-1}\boldsymbol{B} \end{bmatrix}$$

Observability

Controlability

$$\boldsymbol{h}_k = \begin{bmatrix} \boldsymbol{D} & \boldsymbol{C}\boldsymbol{B} & \boldsymbol{C}\boldsymbol{A}\boldsymbol{B} & \boldsymbol{C}\boldsymbol{A}^2\boldsymbol{B} & \boldsymbol{C}\boldsymbol{A}^3\boldsymbol{B} & \cdots & \boldsymbol{C}\boldsymbol{A}^{k-1}\boldsymbol{B} \end{bmatrix}$$

Markov parameters of the system

EigenRealization Algorithm

$$\boldsymbol{H}_1 = \begin{bmatrix} h_1 & h_2 & h_3 & h_4 & h_5 \\ h_2 & h_3 & h_4 & h_5 & h_6 \\ h_3 & h_4 & h_5 & h_6 & h_7 \\ h_4 & h_5 & h_6 & h_7 & h_8 \\ h_5 & h_6 & h_7 & h_8 & h_9 \end{bmatrix}$$

EigenRealization Algorithm

$$H_1 = \begin{bmatrix} CB & CAB & CA^2B & CA^3B & CA^4B \\ CAB & CA^2B & CA^3B & CA^4B & CA^5B \\ CA^2B & CA^3B & CA^4B & CA^5B & CA^6B \\ CA^3B & CA^4B & CA^5B & CA^6B & CA^7B \\ CA^4B & CA^5B & CA^6B & CA^7B & CA^8B \end{bmatrix}$$

EigenRealization Algorithm

$$H_1 = \begin{bmatrix} C \\ CA \\ CA^2 \\ CA^3 \\ CA^4 \end{bmatrix} \begin{bmatrix} B & AB & A^2B & A^3B & A^4B \end{bmatrix}$$

EigenRealization Algorithm

Observability : $\mathcal{O} = \boldsymbol{U}\boldsymbol{\Sigma}^{\frac{1}{2}}$

Controlability : $\mathcal{C} = \boldsymbol{\Sigma}^{\frac{1}{2}}\boldsymbol{V}^T$

EigenRealization Algorithm

$$\boldsymbol{H}_2 = \begin{bmatrix} h_2 & h_3 & h_4 & h_5 & h_6 \\ h_3 & h_4 & h_5 & h_6 & h_7 \\ h_4 & h_5 & h_6 & h_7 & h_8 \\ h_5 & h_6 & h_7 & h_8 & h_9 \\ h_6 & h_7 & h_8 & h_9 & h_{10} \end{bmatrix}$$

EigenRealization Algorithm

$$H_2 = \begin{bmatrix} C \\ CA \\ CA^2 \\ CA^3 \\ CA^4 \end{bmatrix} A \begin{bmatrix} B & AB & A^2B & A^3B & A^4B \end{bmatrix}$$

EigenRealization Algorithm

Natural dynamics

$$A = \mathcal{O}^\dagger \boldsymbol{H}_2 \mathcal{C}^\dagger$$

Actuators

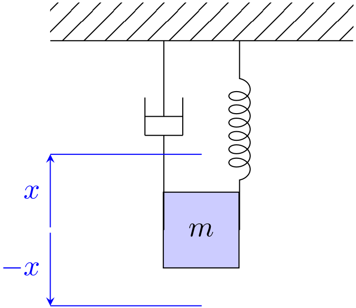
$$\boldsymbol{B} = \left[\boldsymbol{\Sigma}^{\frac{1}{2}} \boldsymbol{V}^T\right]_{:,1:p}$$

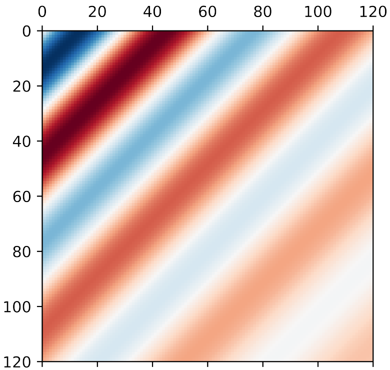
Measurements

$$\boldsymbol{C} = \left[\boldsymbol{U} \boldsymbol{\Sigma}^{\frac{1}{2}}\right]_{1:q,:}$$

Feedthrough

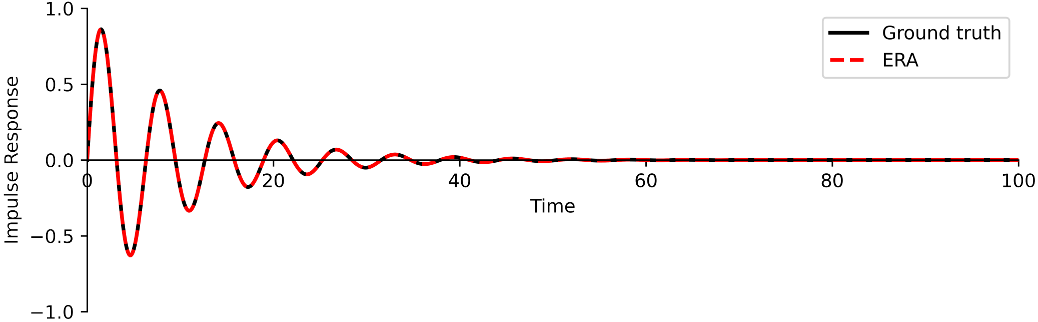
$$\boldsymbol{D} = \boldsymbol{h}_0$$





$$\mathbf{\Sigma} = \begin{bmatrix} 27.68 & 22.62 & 0 & 0 & \cdots \end{bmatrix}$$

$$\boldsymbol{A} = \begin{bmatrix} 0.986 & -0.098 \\ 0.098 & 0.984 \end{bmatrix} \quad \boldsymbol{B} = \begin{bmatrix} -0.232 \\ 0.210 \end{bmatrix} \quad \boldsymbol{C} = \begin{bmatrix} -0.232 & -0.210 \end{bmatrix}$$



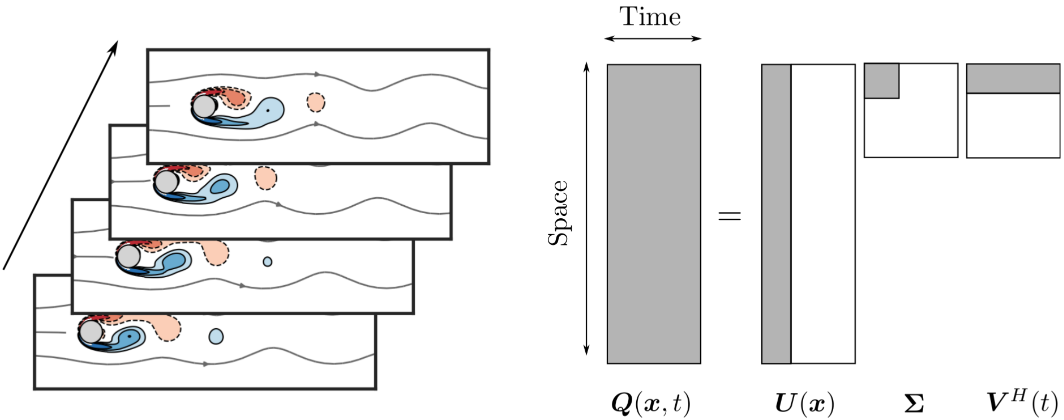
Cylinder flow example

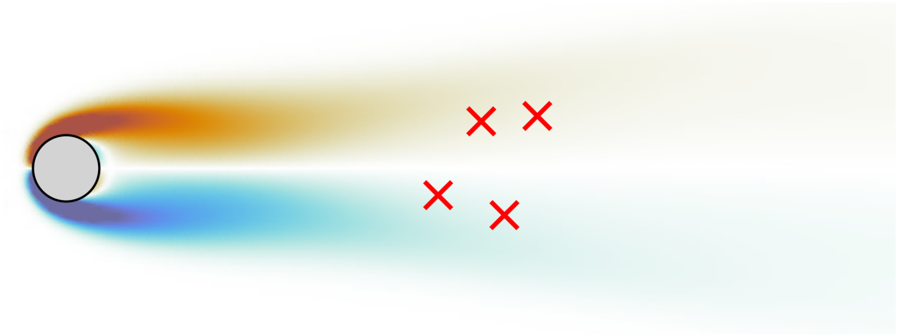
- Relies solely on the impulse response of the system and efficient numerical linear algebra.
- Identifies the minimal number of degrees of freedom needed to represent the dynamics.
- Benefits from strong theoretical guarantees and is well-grounded in control theory.

+

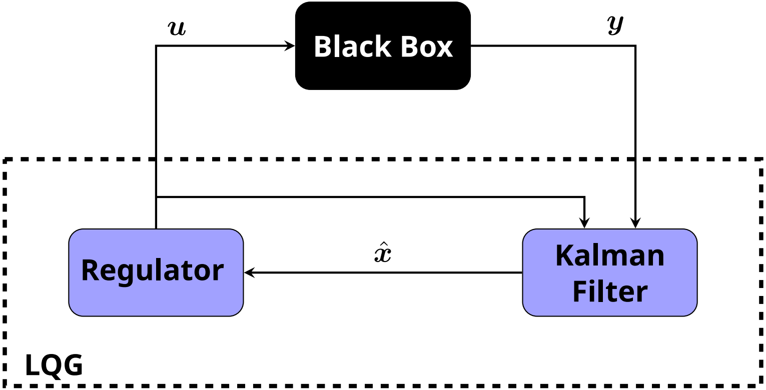
- Need a really good estimate of the impulse reponse.
- Extension to **interpretable** nonlinear systems is not straightforward.
- Parametric variations cannot be easily included into this framework.

Conclusion


Time
Space
=
$\boldsymbol{Q}(\boldsymbol{x}, t)$
$\boldsymbol{U}(\boldsymbol{x})$
$\boldsymbol{\Sigma}$
$\boldsymbol{V}^H(t)$



$$
\begin{aligned}
\boldsymbol{x}_{k+1} &= \boldsymbol{A}\boldsymbol{x}_k + \boldsymbol{B}\boldsymbol{u}_k \\
\boldsymbol{y}_k &= \boldsymbol{C}\boldsymbol{x}_k + \boldsymbol{D}\boldsymbol{u}_k
\end{aligned}
$$


u
Black Box
y
Regulator
$\hat{x}$
Kalman Filter
LQG

Thank you for your attention!

Any question?