MECHANICAL ENGINEERING

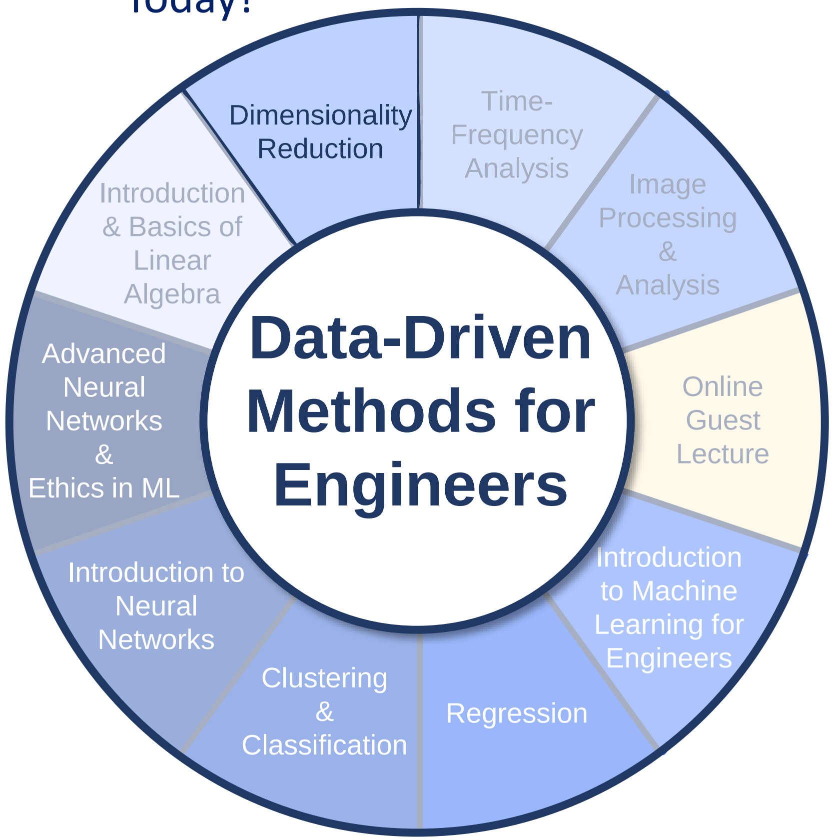
Data-Driven Methods for Engineers (MECH0107) 2025 - 2026

Dr Lama Hamadeh
Module Lead
Office 429 | Roberts Building
Mechanical Engineering Department
l.hamadeh@ucl.ac.uk

Dr Llewellyn Morse
Module Tutor
Office 503D| Roberts Building
Mechanical Engineering Department
l.morse@ucl.ac.uk

Module Lectures

Today!



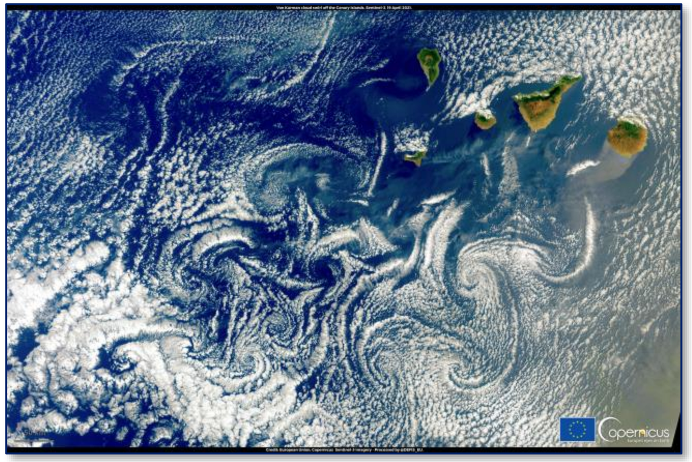
Introduction

- High-dimensional data refers to datasets with a large number of variables, features, or attributes describing each observation.

- While high-dimensional data can capture rich and detailed information, it also introduces unique challenges (often called the “curse of dimensionality”) where the sheer number of features makes patterns harder to detect and increases computational demands.

- In many naturally occurring systems, it is observed that data exhibit dominant patterns, which may be characterized by a low-dimensional attractor or manifold. Analyzing high-dimensional data effectively requires techniques such as dimensionality reduction, to identify the most relevant information while preserving the underlying structure of the data.

Complex Dynamical System:
Von Kármán vortex cloud swirls off the Canary Islands, Spain



NETFLIX prime video

Introduction

- The pile of trash in the foreground represents the raw, high-dimensional data—messy, complex, and seemingly without clear meaning when viewed directly.

Artwork by Tim Noble and Sue Webster

Introduction

- The pile of trash in the foreground represents the raw, high-dimensional data—messy, complex, and seemingly without clear meaning when viewed directly.

- However, when illuminated from the right perspective, its shadow on the wall reveals a clear, simplified image of two people, analogous to how dimensionality reduction techniques project complex, high-dimensional datasets into a lower-dimensional space that preserves the essential structure or meaning.

- In the same way the shadow filters out irrelevant details and organizes the chaos into a recognizable form, dimensionality reduction extracts the most important features from data, making patterns and relationships easier to interpret while discarding noise.

Artwork by Tim Noble and Sue Webster

Outline

- Singular Value Decomposition (SVD)
- Principal Component Analysis (PCA)

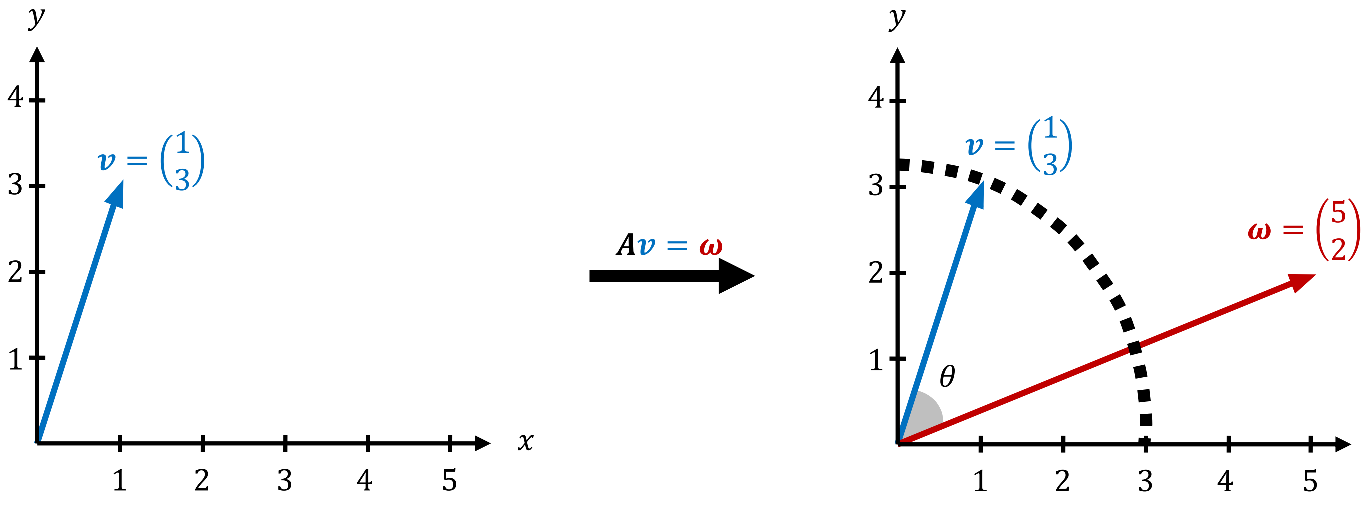
Singular Value Decomposition (SVD)

Graphical Representation

$$\mathbf{v} = \begin{bmatrix} 1 \\ 3 \end{bmatrix} \qquad \mathbf{A} = \begin{bmatrix} 2 & 1 \\ -1 & 1 \end{bmatrix}$$

$$\omega = \mathbf{A}\mathbf{v} = \begin{bmatrix} 5 \\ 2 \end{bmatrix}$$

Stretched & Rotated



Singular Value Decomposition (SVD)

Graphical Representation

$$\mathbf{A} = \begin{bmatrix} 2 & 1 \\ -1 & 1 \end{bmatrix}$$

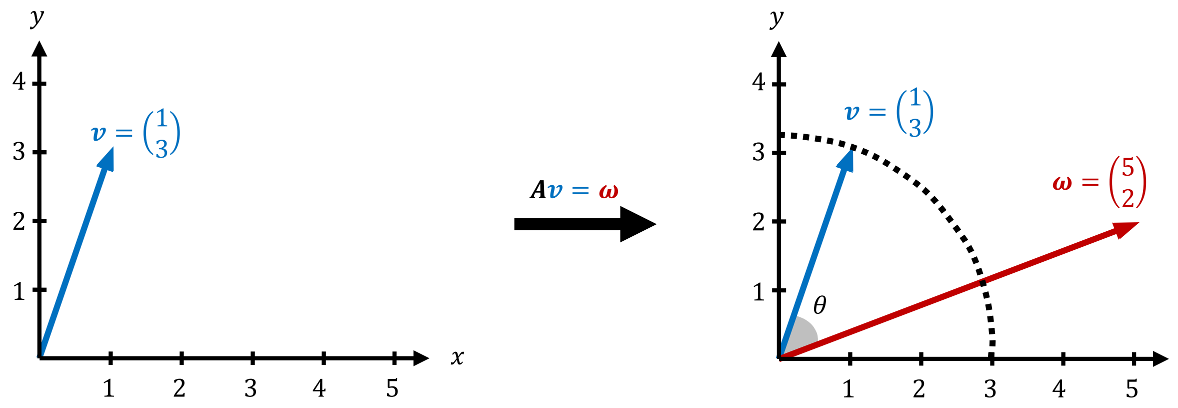
Rotation

$$\mathbf{A}_{\text{Rotation}} = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix}$$

Compressing/Stretching

$$\mathbf{A}_{\text{Scaling}} = \begin{bmatrix} \alpha & 0 \\ 0 & \alpha \end{bmatrix}$$

Defines the amount of stretching/compressing applied on the vector.
$\alpha > 0$ → Stretching
$\alpha < 0$ → Compressing
$\alpha = 0$ → Vector with a magnitude $= 0$



$$\mathbf{v} = \begin{bmatrix} 1 \\ 3 \end{bmatrix} \quad \mathbf{A} = \begin{bmatrix} 2 & 1 \\ -1 & 1 \end{bmatrix} \quad \Longrightarrow \quad \omega = \mathbf{A}\mathbf{v} = \begin{bmatrix} 5 \\ 2 \end{bmatrix}$$

Singular Value Decomposition (SVD)

Graphical Representation

(2×2) (2×2) (2×2) (2×2)

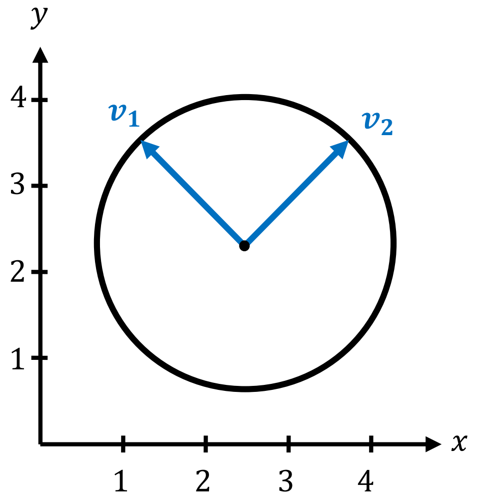
$$\mathbf{A}\begin{bmatrix} | & | \\ \mathbf{v_1} & \mathbf{v_2} \\ | & | \end{bmatrix} = \begin{bmatrix} | & | \\ \mathbf{u_1} & \mathbf{u_2} \\ | & | \end{bmatrix}\begin{bmatrix} \sigma_1 & 0 \\ 0 & \sigma_2 \end{bmatrix}$$

The *singular values* are simply the length and width of the transformed circle. If one of the singular values is 0, this means that our transformation flattens our circle. And the larger of the two singular values tells you about the maximum "action" of the transformation.

Or:

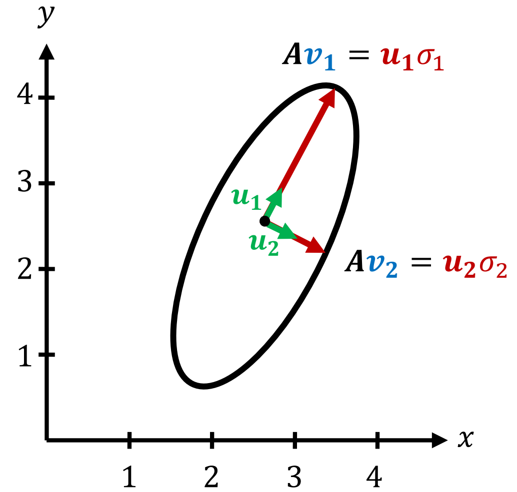
$$\mathbf{A} = \begin{bmatrix} | & | \\ \mathbf{u_1} & \mathbf{u_2} \\ | & | \end{bmatrix}\begin{bmatrix} \sigma_1 & 0 \\ 0 & \sigma_2 \end{bmatrix}\begin{bmatrix} - & \mathbf{v_1^*} & - \\ - & \mathbf{v_2^*} & - \end{bmatrix}$$

$\boldsymbol{v_1}$ and $\boldsymbol{v_2}$ are **orthonormal** vectors; unit vectors



$\boldsymbol{Av} = \boldsymbol{u}\,\sigma$

$\boldsymbol{u_1}$ and $\boldsymbol{u_2}$ are **orthonormal** vectors; unit vectors



Singular Value Decomposition (SVD)

Graphical Representation

$$A = U\Sigma V^*$$

V^*

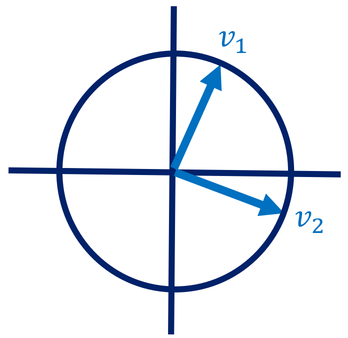
Rotates all input vectors v_1 and v_2. The new vectors e_1 and e_2 are the new vectors in a rotated coordinate system.

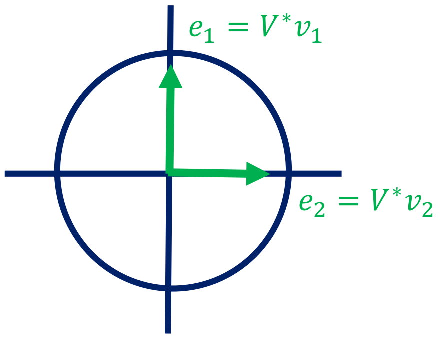
Σ

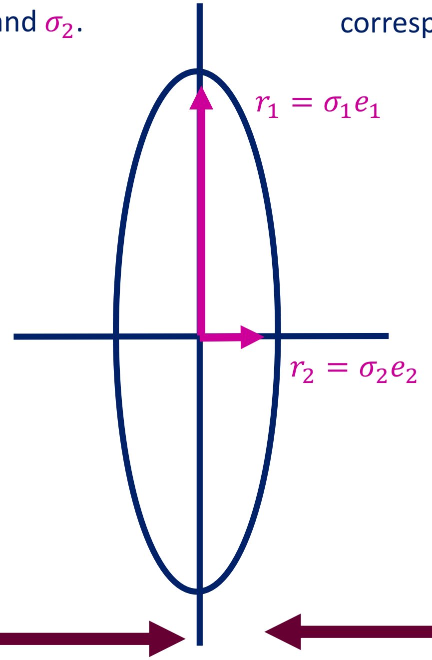
Stretches/compresses the rotated vectors. The unit circle becomes an ellipse stretched by σ_1 and σ_2.

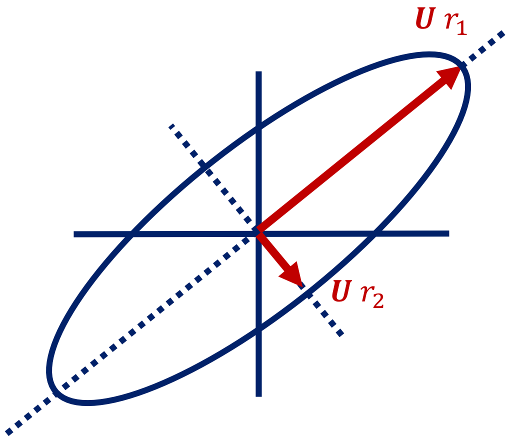
U

Rotates the scaled result into its final orientation in output space - the one that corresponds to the full transformation A.









Rotate

Scale

Rotate

MECHANICAL ENGINEERING

Singular Value Decomposition (SVD)

Graphical Representation

Generalizing the SVD expression for a linear transformation $\boldsymbol{A}$ that maps vectors from an n-dimensional space to an n -dimensional space gives:

$$\boldsymbol{A} \quad = \quad \boldsymbol{U} \quad \boldsymbol{\Sigma} \quad \boldsymbol{V}^*$$

$$\underset{n \times n}{\begin{bmatrix} \end{bmatrix}} = \underset{n \times n}{\begin{bmatrix} | & | & | \\ \boldsymbol{u_1} & \cdots & \boldsymbol{u_n} \\ | & | & | \end{bmatrix}} \underset{n \times n}{\begin{bmatrix} \sigma_1 & & 0 \\ & \ddots & \\ 0 & & \sigma_n \end{bmatrix}} \underset{n \times n}{\begin{bmatrix} - & \boldsymbol{v_1^*} & - \\ - & \vdots & - \\ - & \boldsymbol{v_n^*} & - \end{bmatrix}}$$

Rotate Scale Rotate

Singular Value Decomposition (SVD)

Mathematical Representation

$$A = U\Sigma V^{*}$$

U → Left Singular Vectors

Σ → Singular Values

V^{*} → Right Singular Vectors

SVD works for all!

A

$(n \times n)$

A

$(m \times n)$
$m \gg n$

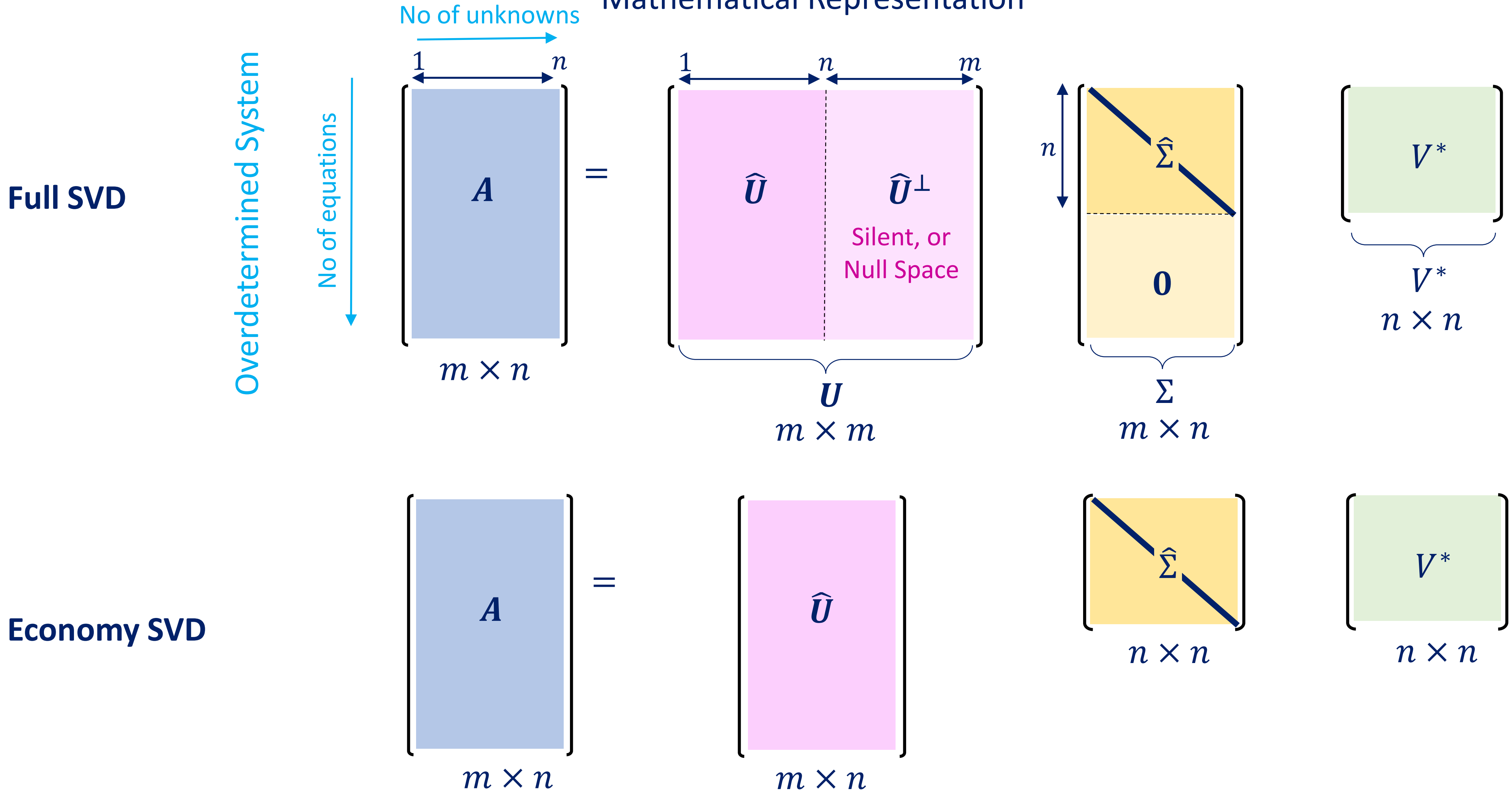
Common case in most of the datasets

A

$(m \times n)$
$m \ll n$

Singular Value Decomposition (SVD)

Mathematical Representation



Singular Value Decomposition (SVD)

Mathematical Representation

$$A = U\Sigma V^{*}$$

Conjugate Transpose

$(\mathbf{V}^*)^* = \mathbf{V}$

$$\begin{aligned}\mathbf{A}^*\mathbf{A} &= (\mathbf{U\Sigma V}^*)^*(\mathbf{U\Sigma V}^*)\\ &= \mathbf{V\Sigma}\underbrace{\mathbf{U}^*\mathbf{U}}_{\mathbf{I}}\mathbf{\Sigma V}^*\\ &= \mathbf{V\Sigma}^2\mathbf{V}^*\end{aligned}$$

Eigenvalues for A^*A

Normalized Eigenvectors for A^*A

$$\mathbf{A}^*\mathbf{A}\mathbf{V} = \mathbf{V\Sigma}^2$$

$$\begin{aligned}\mathbf{A}\mathbf{A}^* &= (\mathbf{U\Sigma V}^*)(\mathbf{U\Sigma V}^*)^*\\ &= \mathbf{U\Sigma V}^*\mathbf{V\Sigma U}^*\\ &= \mathbf{U\Sigma}^2\mathbf{U}^*\end{aligned}$$

Eigenvalues for AA^*

Normalized Eigenvectors for AA^*

$$\mathbf{A}\mathbf{A}^*\mathbf{U} = \mathbf{U\Sigma}^2$$

Singular Value Decomposition (SVD)

Mathematical Representation

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

$$\mathbf{A} = \begin{bmatrix} 3 & 0 \\ 4 & 5 \end{bmatrix}$$

$$\mathbf{A}^*\mathbf{A} = \begin{bmatrix} 3 & 4 \\ 0 & 5 \end{bmatrix} \begin{bmatrix} 3 & 0 \\ 4 & 5 \end{bmatrix} = \begin{bmatrix} 25 & 20 \\ 20 & 25 \end{bmatrix}$$

$$\mathbf{A}^*\mathbf{A}\mathbf{V} = \mathbf{V}\mathbf{\Sigma}^2$$

$$\mathbf{A}\mathbf{A}^* = \begin{bmatrix} 3 & 0 \\ 4 & 5 \end{bmatrix} \begin{bmatrix} 3 & 4 \\ 0 & 5 \end{bmatrix} = \begin{bmatrix} 9 & 12 \\ 12 & 41 \end{bmatrix}$$

$$\mathbf{A}\mathbf{A}^*\mathbf{U} = \mathbf{U}\mathbf{\Sigma}^2$$

Singular Value Decomposition (SVD)

Mathematical Representation

$$\mathbf{A} = \begin{bmatrix} 3 & 0 \\ 4 & 5 \end{bmatrix}$$

$$\mathbf{A}^*\mathbf{A} = \begin{bmatrix} 3 & 4 \\ 0 & 5 \end{bmatrix} \begin{bmatrix} 3 & 0 \\ 4 & 5 \end{bmatrix} = \begin{bmatrix} 25 & 20 \\ 20 & 25 \end{bmatrix}$$

Calculating eigenvalues:

$$\begin{aligned} |\mathbf{A}^*\mathbf{A} - \overset{\sigma^2}{\lambda} I| &= \begin{bmatrix} 25 & 20 \\ 20 & 25 \end{bmatrix} - \lambda \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \\ &= \begin{bmatrix} 25 & 20 \\ 20 & 25 \end{bmatrix} - \begin{bmatrix} \lambda & 0 \\ 0 & \lambda \end{bmatrix} \\ &= \begin{bmatrix} 25 - \lambda & 20 \\ 20 & 25 - \lambda \end{bmatrix} \\ &= 0 \end{aligned}$$

$$\lambda_1 = \sigma_1^2 = 45 \longrightarrow \sigma_1 = \sqrt{45}$$

$$\lambda_2 = \sigma_2^2 = 5 \longrightarrow \sigma_2 = \sqrt{5}$$

Singular Value Decomposition (SVD)

Mathematical Representation

$$\mathbf{A} = \begin{bmatrix} 3 & 0 \\ 4 & 5 \end{bmatrix}$$

$$\mathbf{A}^*\mathbf{A} = \begin{bmatrix} 3 & 4 \\ 0 & 5 \end{bmatrix} \begin{bmatrix} 3 & 0 \\ 4 & 5 \end{bmatrix} = \begin{bmatrix} 25 & 20 \\ 20 & 25 \end{bmatrix}$$

Calculating eigenvectors :

For $\sigma_1 = \sqrt{45}$: $\mathbf{A}^*\mathbf{A}\mathbf{V}_1 = \sigma^2 \mathbf{V}_1$

$$\begin{bmatrix} 25 & 20 \\ 20 & 25 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \end{bmatrix} = 45 \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

$$\boxed{\mathbf{V}_1 = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ 1 \end{bmatrix}}$$

Normalization Factor

For $\sigma_2 = \sqrt{5}$: $\mathbf{A}^*\mathbf{A}\mathbf{V}_2 = \sigma^2 \mathbf{V}_2$

$$\begin{bmatrix} 25 & 20 \\ 20 & 25 \end{bmatrix} \begin{bmatrix} -1 \\ 1 \end{bmatrix} = 5 \begin{bmatrix} -1 \\ 1 \end{bmatrix}$$

$$\boxed{\mathbf{V}_2 = \frac{1}{\sqrt{2}} \begin{bmatrix} -1 \\ 1 \end{bmatrix}}$$

Normalization Factor

Singular Value Decomposition (SVD)

Mathematical Representation

$$\mathbf{A} = \begin{bmatrix} 3 & 0 \\ 4 & 5 \end{bmatrix}$$

$$\mathbf{A}\mathbf{A}^* = \begin{bmatrix} 3 & 0 \\ 4 & 5 \end{bmatrix} \begin{bmatrix} 3 & 4 \\ 0 & 5 \end{bmatrix} = \begin{bmatrix} 9 & 12 \\ 12 & 41 \end{bmatrix}$$

Same eigenvalues

$$\sigma_1 = \sqrt{45}$$

$$\sigma_2 = \sqrt{5}$$

Calculating eigenvectors:

For $\sigma_1 = \sqrt{45}$:

$$\mathbf{A}\mathbf{A}^*\mathbf{U}_1 = \sigma^2\mathbf{U}_1$$

$$\begin{bmatrix} 9 & 12 \\ 12 & 41 \end{bmatrix} \begin{bmatrix} 1 \\ 3 \end{bmatrix} = 45 \begin{bmatrix} 1 \\ 3 \end{bmatrix}$$

$$\boxed{\mathbf{U}_1 = \frac{1}{\sqrt{10}} \begin{bmatrix} 1 \\ 3 \end{bmatrix}}$$

Normalization Factor

For $\sigma_2 = \sqrt{5}$:

$$\mathbf{A}\mathbf{A}^*\mathbf{U}_2 = \sigma^2\mathbf{U}_2$$

$$\begin{bmatrix} 9 & 12 \\ 12 & 41 \end{bmatrix} \begin{bmatrix} -3 \\ 1 \end{bmatrix} = 5 \begin{bmatrix} -3 \\ 1 \end{bmatrix}$$

$$\boxed{\mathbf{U}_1 = \frac{1}{\sqrt{10}} \begin{bmatrix} -3 \\ 1 \end{bmatrix}}$$

Normalization Factor

Singular Value Decomposition (SVD)

Mathematical Representation

$$\mathbf{A} = \begin{bmatrix} 3 & 0 \\ 4 & 5 \end{bmatrix}$$

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

$$\boldsymbol{U} = \frac{1}{\sqrt{10}}\begin{bmatrix} 1 & -3 \\ 3 & 1 \end{bmatrix} \qquad \boldsymbol{\Sigma} = \begin{bmatrix} \sqrt{45} & 0 \\ 0 & \sqrt{5} \end{bmatrix} \qquad \boldsymbol{V} = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & -1 \\ 1 & 1 \end{bmatrix}$$

Singular Value Decomposition (SVD)

Physical Understanding – The Big Picture

$$\underset{m \times n}{\boldsymbol{A}} = \underset{m \times n}{\boldsymbol{U}} \; \underset{n \times n}{\Sigma} \; \underset{n \times n}{V^*}$$

Time, Features

$$\mathbf{A} = \begin{bmatrix} - & \text{Snapshot } 1 & - \\ - & \text{Snapshot } 2 & - \\ & \vdots & \\ - & \text{Snapshot } n & - \end{bmatrix}$$

The Engineering System

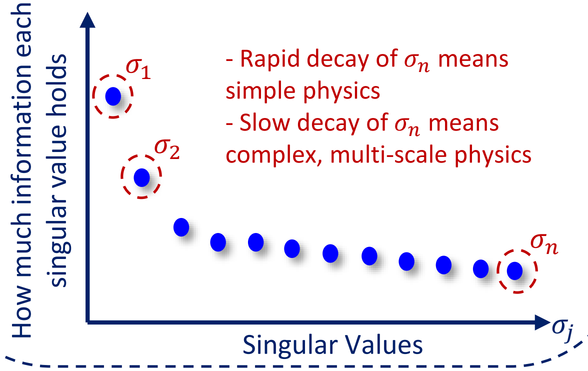
$$\mathbf{U} = \begin{bmatrix} | & | & \cdots & | \\ \mathbf{u_1} & \mathbf{u_2} & \cdots & \mathbf{u_n} \\ | & | & \cdots & | \end{bmatrix}$$

describes the time evolution of each pattern?

$$\Sigma = \begin{bmatrix} \sigma_1 & 0 & 0 & \dots & 0 \\ 0 & \sigma_2 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & \sigma_n \end{bmatrix}$$

$$\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_n \geq 0$$

Measure how strongly the system reacts and determines the **model order**.



- Rapid decay of σ_n means simple physics
- Slow decay of σ_n means complex, multi-scale physics

$$\mathbf{V}^* = \begin{bmatrix} - & \mathbf{v_1^*} & - \\ - & \mathbf{v_2^*} & - \\ - & \vdots & - \\ - & \mathbf{v_n^*} & - \end{bmatrix}$$

Gives dominant spatial patterns present in the data.

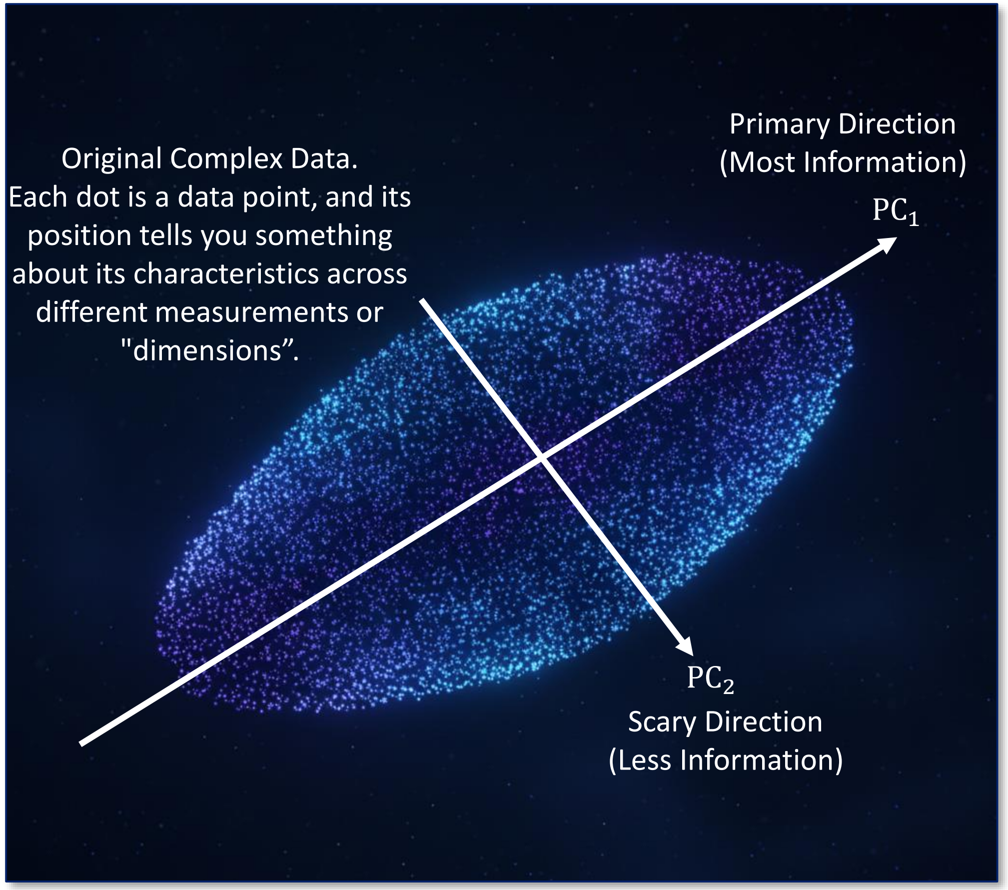
Outline

- Singular Value Decomposition (SVD)
- Principal Component Analysis (PCA)

Principal Component Analysis (PCA)

- **Principal Component Analysis (PCA)** is a statistical technique used to reduce the dimensionality of a dataset while preserving as much variability as possible. It works by transforming the original correlated variables into a new set of uncorrelated variables called *principal components*, which are ordered so that the first few capture most of the variation in the data.

- The main aims of PCA are to simplify data interpretation, reduce computational complexity, and eliminate noise or redundancy. By focusing on the most significant components, PCA helps reveal underlying patterns, visualize high-dimensional data, and improve the performance of subsequent analyses or machine learning models.



Principal Component Analysis (PCA)

Hooke's force

$$\mathbf{F} = m\mathbf{a}$$

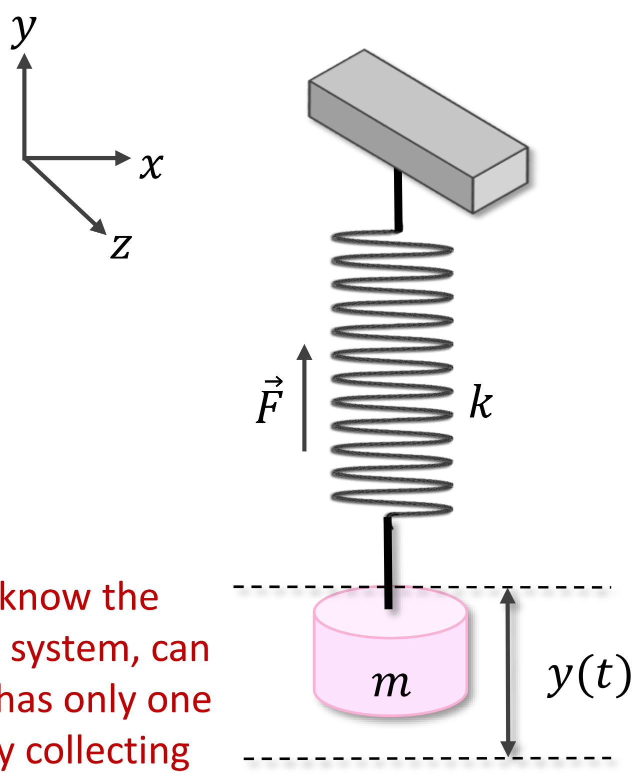
Hooke's constant

$$-ky(t) = m\frac{d^2y(t)}{dt^2}$$

Displacement

The mass is suspended by a spring

$$\frac{d^2y(t)}{dt^2} = -\frac{k}{m}y(t) = -\omega^2 y(t)$$



$$\frac{d^2y(t)}{dt^2} + \omega^2 y(t) = 0$$

Solving the ODE

$$y(t) = A\cos(\omega t) + B\sin(\omega t)$$

Suppose that we don't know the governing equation of this system, can we know that this system has only one degree of freedom just by collecting measurements of its motion?

The state of the system can be described as a one-degree-of-freedom system

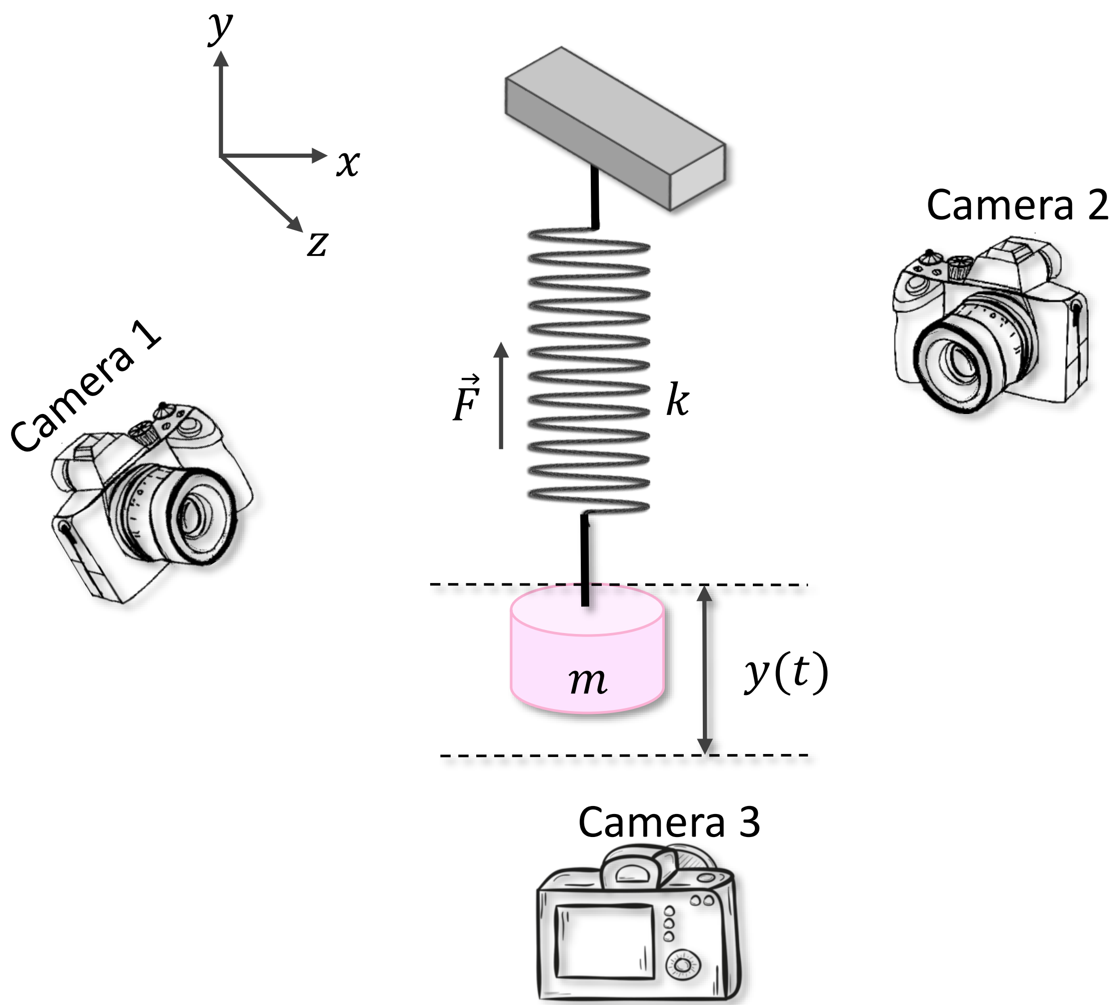
Principal Component Analysis (PCA)

Step 1: Data Collection

$$\text{Camera 1} : (\mathbf{x}_a, \mathbf{y}_a, \mathbf{z}_a) \qquad \text{Camera 2} : (\mathbf{x}_b, \mathbf{y}_b, , \mathbf{z}_b)$$

$$\text{Camera 3} : (\mathbf{x}_c, \mathbf{y}_c, \mathbf{z}_c)$$

$$\mathbf{X} = \begin{array}{c} \xrightarrow{\quad n \quad} \\ \begin{bmatrix} - & \mathbf{x}_a & - \\ - & \mathbf{y}_a & - \\ - & \mathbf{z}_a & - \\ - & \mathbf{x}_b & - \\ - & \mathbf{y}_b & - \\ - & \mathbf{z}_b & - \\ - & \mathbf{x}_c & - \\ - & \mathbf{y}_c & - \\ - & \mathbf{z}_c & - \end{bmatrix} \end{array} \Bigg\downarrow m$$



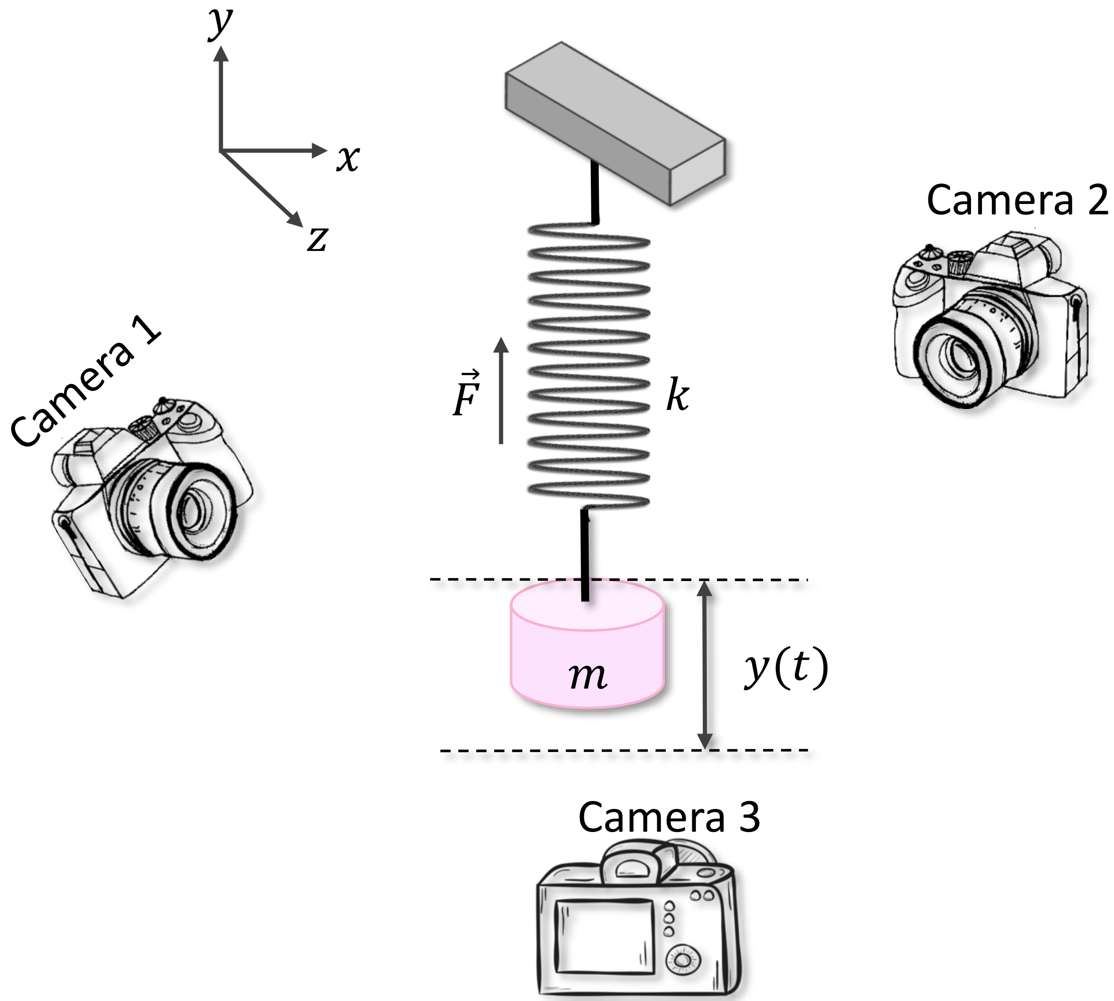
MECHANICAL ENGINEERING

Principal Component Analysis (PCA)

Step 2: Compute the mean of X

$$\overline{\mathbf{x}}_i = \frac{1}{n}\sum_{j=1}^{n} \mathbf{X}_{ij} \quad \Longrightarrow \quad \overline{\mathbf{X}} = \begin{array}{cc} 1 & \\ \begin{bmatrix} \overline{\mathbf{x}}_1 \\ \overline{\mathbf{x}}_2 \\ \vdots \\ \overline{\mathbf{x}}_m \end{bmatrix} & m \end{array}$$

The mean of each row

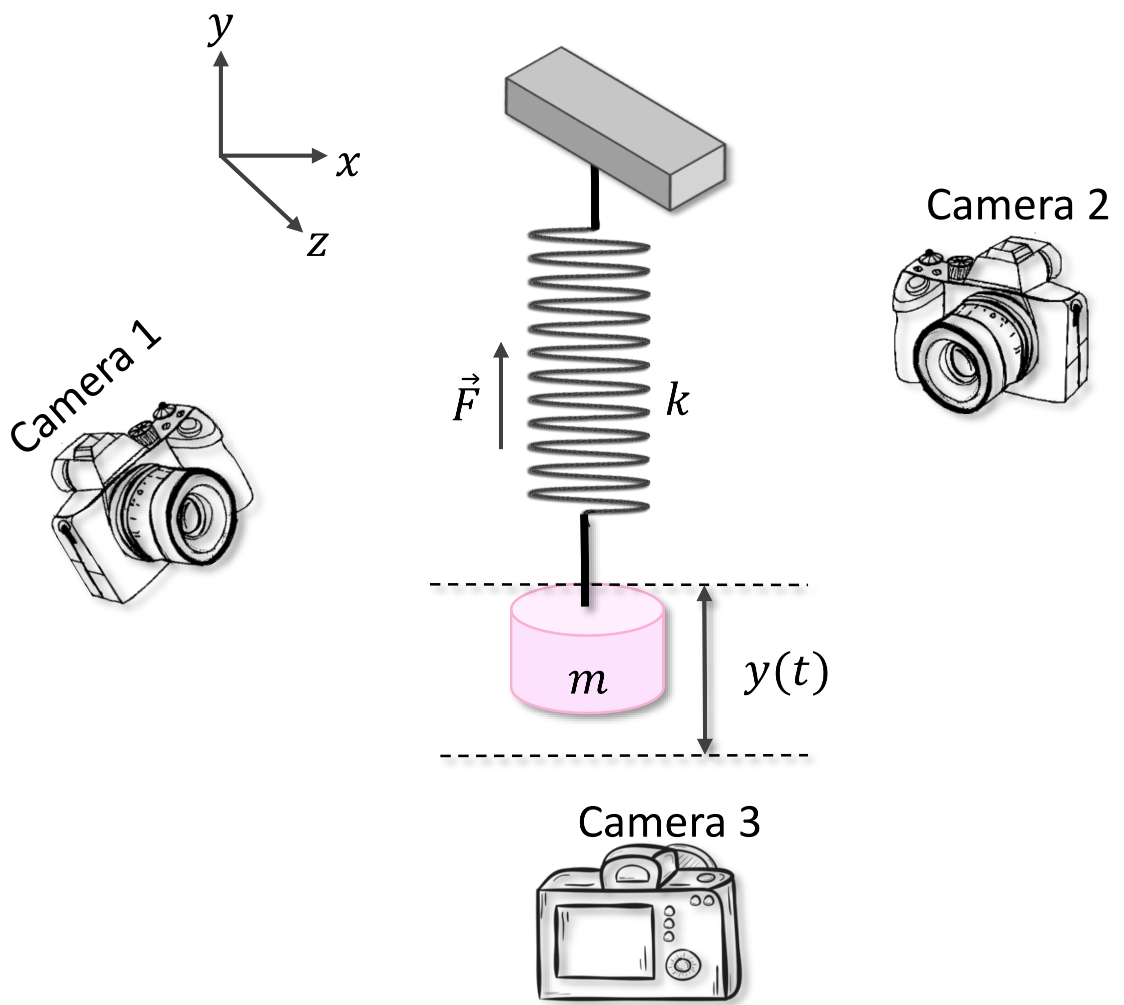


Principal Component Analysis (PCA)

Step 3: Subtract $\overline{\mathbf{X}}$ from $\mathbf{X}$

$$\underset{m \times n}{\mathbf{B}} = \mathbf{X} - \overline{\mathbf{X}}$$

$$= \underbrace{\begin{bmatrix} - & \mathbf{x}_a & - \\ - & \mathbf{y}_a & - \\ - & \mathbf{z}_a & - \\ - & \mathbf{x}_b & - \\ - & \mathbf{y}_b & - \\ - & \mathbf{z}_b & - \\ - & \mathbf{x}_c & - \\ - & \mathbf{y}_c & - \\ - & \mathbf{z}_c & - \end{bmatrix}}_{m \times n} - \underbrace{\begin{bmatrix} \overline{\mathbf{x}}_1 \\ \overline{\mathbf{x}}_2 \\ \vdots \\ \overline{\mathbf{x}}_m \end{bmatrix}}_{m \times 1} = \underbrace{\begin{bmatrix} - & \tilde{\mathbf{x}}_a & - \\ - & \tilde{\mathbf{y}}_a & - \\ - & \tilde{\mathbf{z}}_a & - \\ - & \tilde{\mathbf{x}}_b & - \\ - & \tilde{\mathbf{y}}_b & - \\ - & \tilde{\mathbf{z}}_b & - \\ - & \tilde{\mathbf{x}}_c & - \\ - & \tilde{\mathbf{y}}_c & - \\ - & \tilde{\mathbf{z}}_c & - \end{bmatrix}}_{m \times n}$$

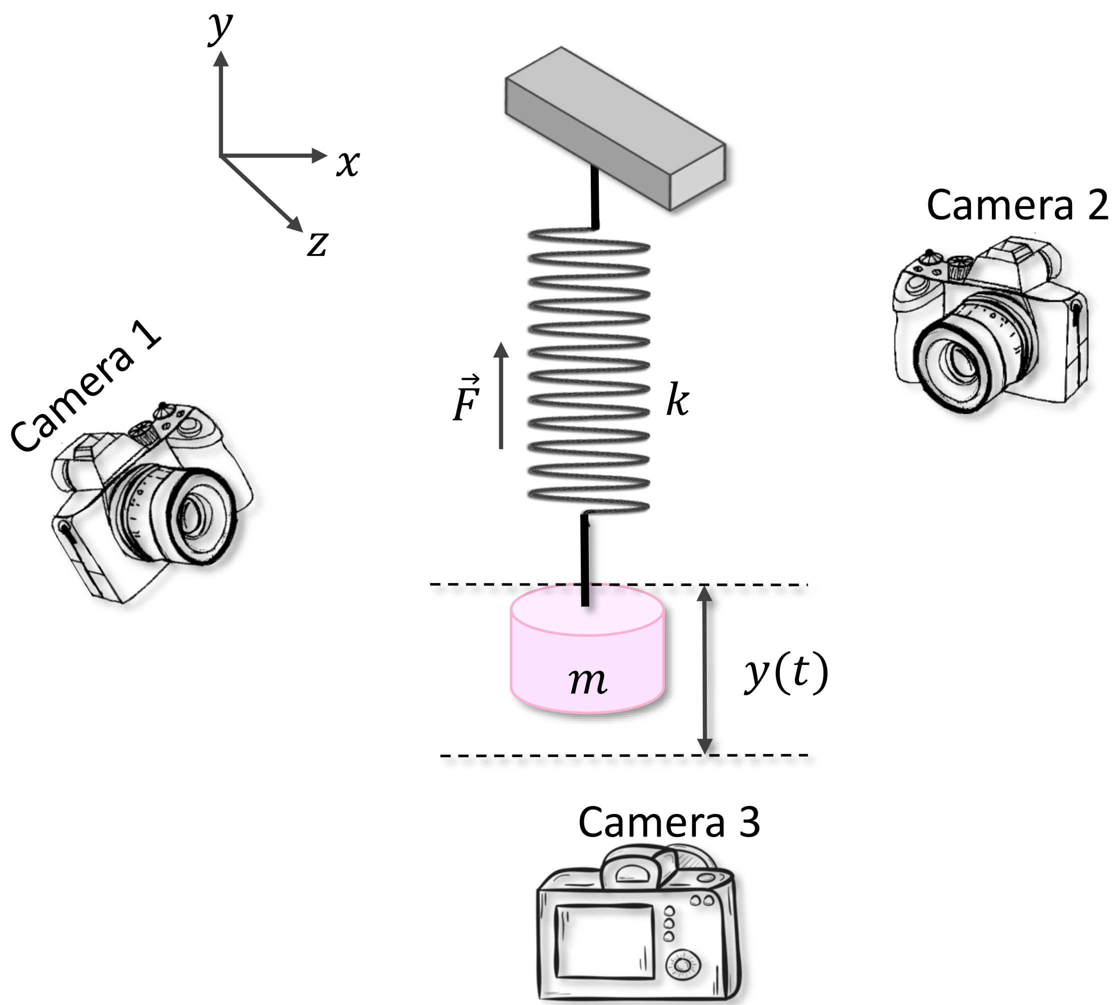


Principal Component Analysis (PCA)

Step 4: Construct the covariance matrix *C*

$$m \times m \quad \mathbf{C} = \frac{1}{n-1}\mathbf{B}\mathbf{B}^*$$

$$\mathbf{C} = \frac{1}{n-1}\begin{bmatrix} - & \tilde{\mathbf{x}}_a & - \\ - & \tilde{\mathbf{y}}_a & - \\ - & \tilde{\mathbf{z}}_a & - \\ - & \tilde{\mathbf{x}}_b & - \\ - & \tilde{\mathbf{y}}_b & - \\ - & \tilde{\mathbf{z}}_b & - \\ - & \tilde{\mathbf{x}}_c & - \\ - & \tilde{\mathbf{y}}_c & - \\ - & \tilde{\mathbf{z}}_c & - \end{bmatrix}\begin{bmatrix} | & | & | & & | & | \\ \tilde{\mathbf{x}}_a & \tilde{\mathbf{y}}_a & \tilde{\mathbf{z}}_a & \cdots & \tilde{\mathbf{y}}_c & \tilde{\mathbf{z}}_c \\ | & | & | & & | & | \end{bmatrix}$$



Principal Component Analysis (PCA)

Step 4: Construct the covariance matrix C

Diagonalise it!

$$m \times m \quad \mathbf{C} = \frac{1}{n-1}\mathbf{B}\mathbf{B}^*$$

Variance

Covariance
Correlated measurements
= redundancy

$$\mathbf{C} = \frac{1}{n-1}\begin{bmatrix} - & \tilde{\mathbf{x}}_a & - \\ - & \tilde{\mathbf{y}}_a & - \\ - & \tilde{\mathbf{z}}_a & - \\ - & \tilde{\mathbf{x}}_b & - \\ - & \tilde{\mathbf{y}}_b & - \\ - & \tilde{\mathbf{z}}_b & - \\ - & \tilde{\mathbf{x}}_c & - \\ - & \tilde{\mathbf{y}}_c & - \\ - & \tilde{\mathbf{z}}_c & - \end{bmatrix}\begin{bmatrix} | & | & | & & | & | \\ \tilde{\mathbf{x}}_a & \tilde{\mathbf{y}}_a & \tilde{\mathbf{z}}_a & \dots & \tilde{\mathbf{y}}_c & \tilde{\mathbf{z}}_c \\ | & | & | & & | & | \end{bmatrix} = \frac{1}{n-1}\begin{bmatrix} \tilde{\mathbf{x}}_a\tilde{\mathbf{x}}_a & \tilde{\mathbf{x}}_a\tilde{\mathbf{y}}_a & \tilde{\mathbf{x}}_a\tilde{\mathbf{z}}_a & \dots & \tilde{\mathbf{x}}_a\tilde{\mathbf{z}}_c \\ \tilde{\mathbf{y}}_a\tilde{\mathbf{x}}_a & \tilde{\mathbf{y}}_a\tilde{\mathbf{y}}_a & \tilde{\mathbf{y}}_a\tilde{\mathbf{z}}_a & \dots & \tilde{\mathbf{y}}_a\tilde{\mathbf{z}}_c \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \tilde{\mathbf{z}}_c\tilde{\mathbf{x}}_a & \dots & \dots & \dots & \tilde{\mathbf{z}}_c\tilde{\mathbf{z}}_c \end{bmatrix}$$

- Remove redundancy, and hence reducing the dimensionality of the data by keeping only the important dimensions.
- Identify those signals with maximal variance that hold the most important information about the studied system.

Principal Component Analysis (PCA)

Step 5: Eigendecompostion of matrix C

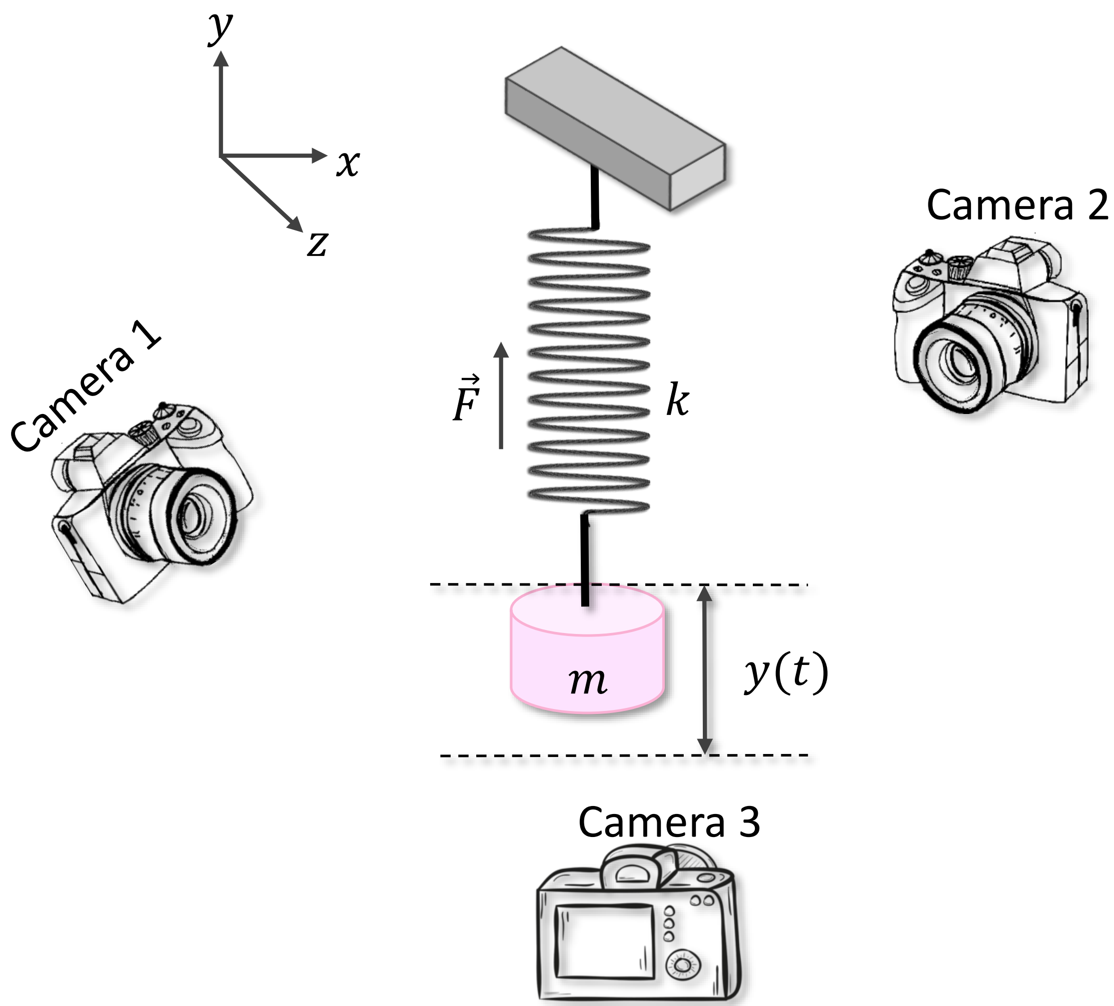
$$\mathbf{C}\mathbf{V} = \mathbf{V}\mathbf{\Lambda}$$

Eigenvalues

Eigenvectors

$$\underset{m \times m}{\mathbf{V}} = \begin{bmatrix} | & | & \cdots & | & \cdots & | \\ v_1 & v_2 & \cdots & v_k & \cdots & v_m \\ | & | & \cdots & | & \cdots & | \end{bmatrix}$$

$$\underset{m \times m}{\mathbf{\Lambda}} = \begin{bmatrix} \lambda_1 & 0 & 0 & 0 & \cdots & 0 \\ 0 & \lambda_2 & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \cdots & \vdots \\ 0 & 0 & 0 & \lambda_k & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & 0 \\ 0 & 0 & 0 & 0 & 0 & \lambda_m \end{bmatrix}$$



Principal Component Analysis (PCA)

Step 6: Sort & Select

Most information → Less information

$m \times k$

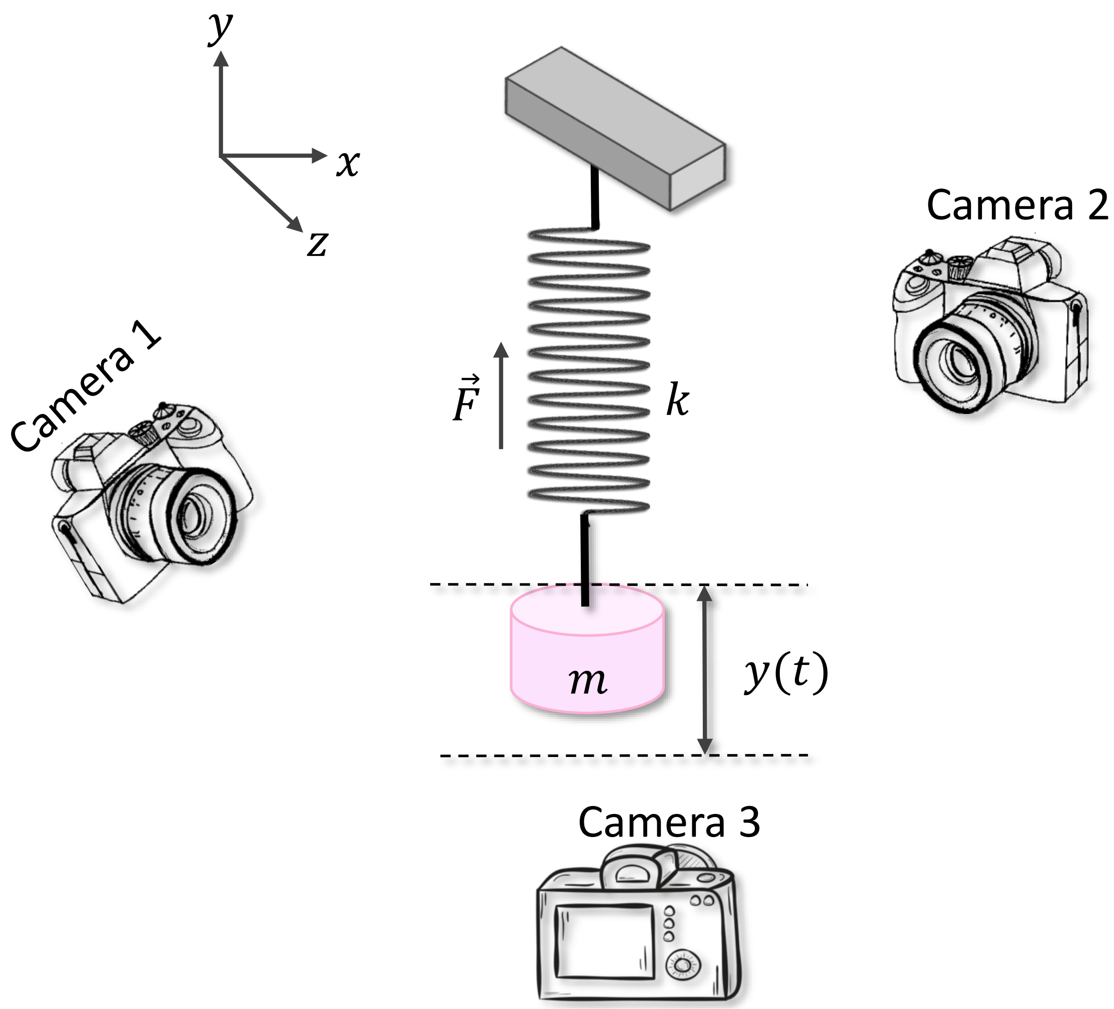
$$\underset{m \times m}{\mathbf{V}} = \begin{bmatrix} | & | & \cdots & | & \cdots & | \\ v_1 & v_2 & \cdots & v_k & \cdots & v_m \\ | & | & \cdots & | & \cdots & | \end{bmatrix}$$

Feature Space

$m \times k$

$$\underset{m \times m}{\boldsymbol{\Lambda}} = \begin{bmatrix} \lambda_1 & 0 & 0 & 0 & \cdots & 0 \\ 0 & \lambda_2 & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \cdots & \vdots \\ 0 & 0 & 0 & \lambda_k & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & 0 \\ 0 & 0 & 0 & 0 & 0 & \lambda_m \end{bmatrix}$$

$$\lambda_1 > \lambda_2 > \cdots \lambda_k > \cdots \lambda_m$$



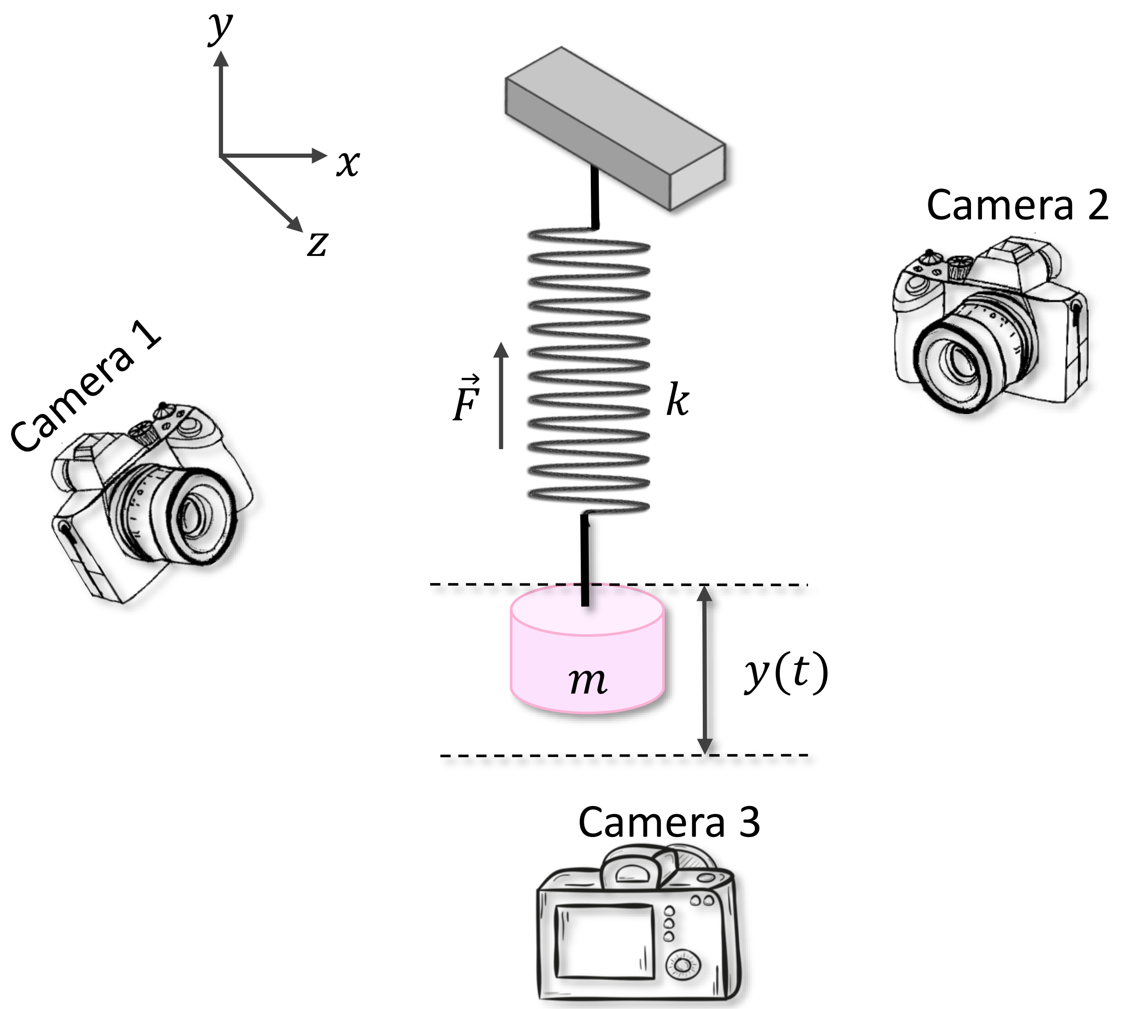
Principal Component Analysis (PCA)

Step 7: Project the data along the principal components

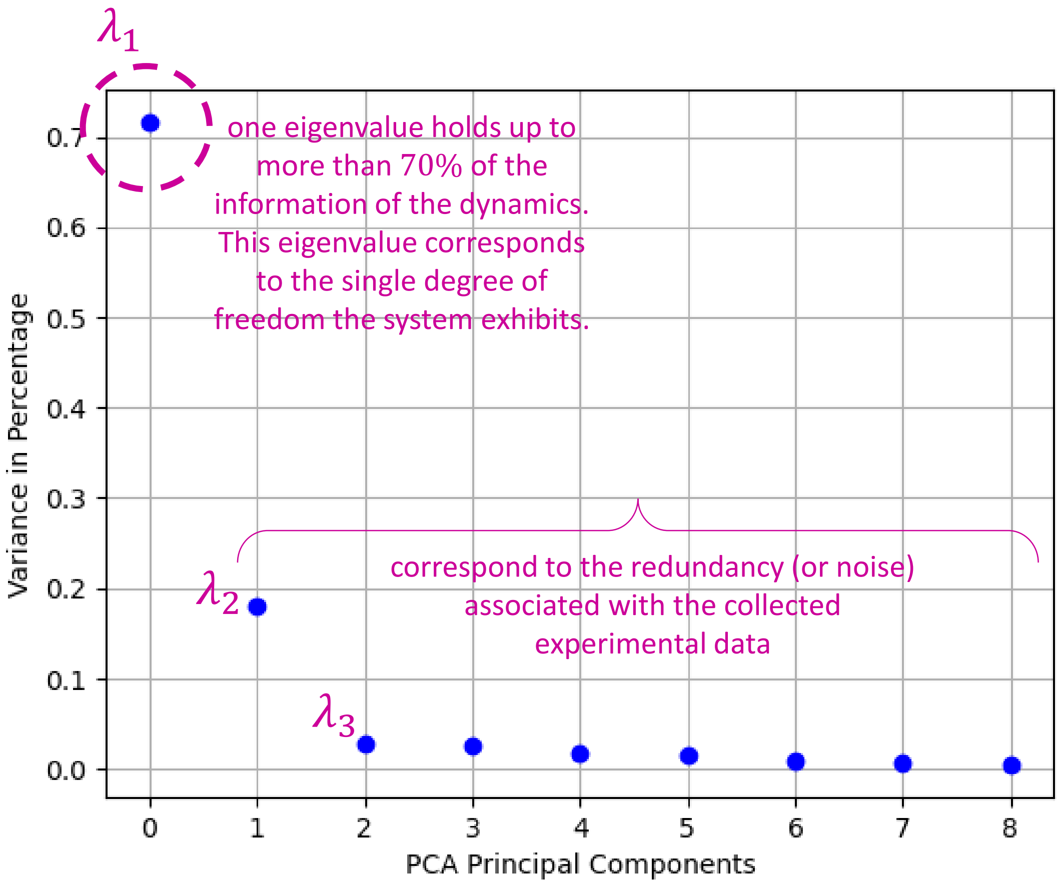
- The end goal here is to transform the original, noisy, redundant, and high-dimensional dataset $\mathbf{X}$ onto the new orthogonal and independent subspace, i.e., the subspace of principal components or PCA scores (or modes).
- This suggests that instead of working directly with the matrix $\mathbf{X}$, we consider working with its transformed low-dimensional version, or the principal component basis.

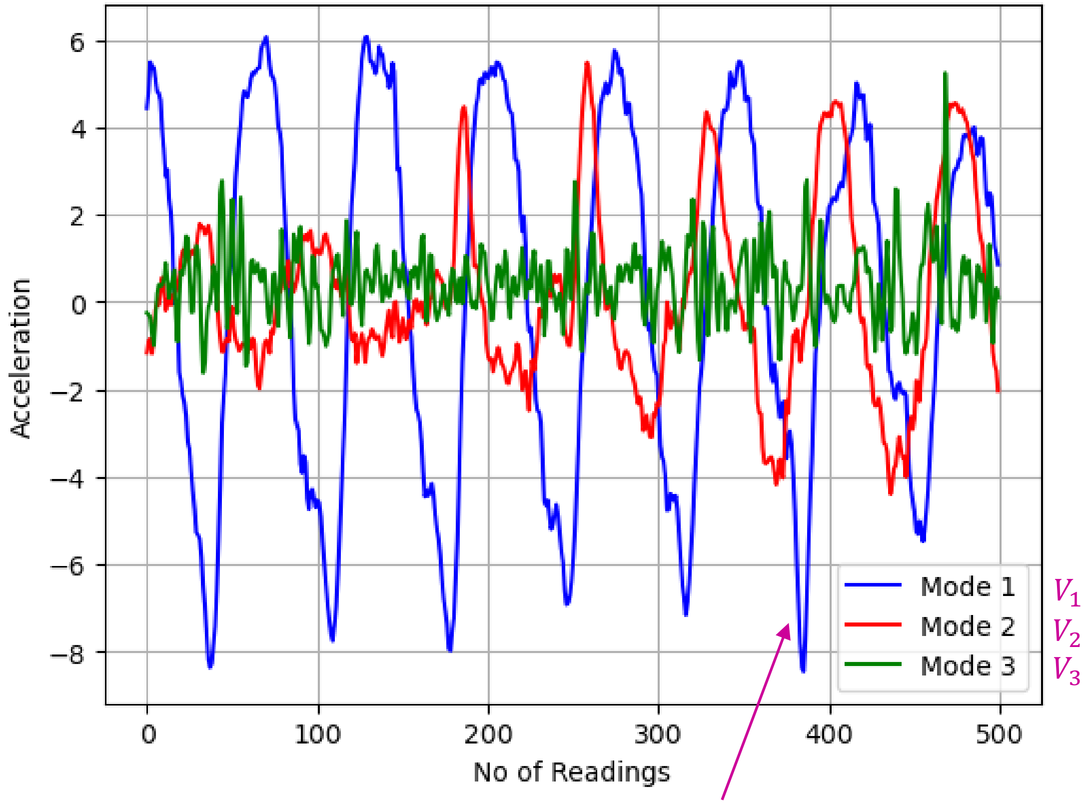
$$\mathbf{Y} = \mathbf{V}_k^T \ \mathbf{B}$$

This matrix contains k principal components (or columns) that hold rich information about the behaviour and the dynamics of the given physical system.



Principal Component Analysis (PCA)





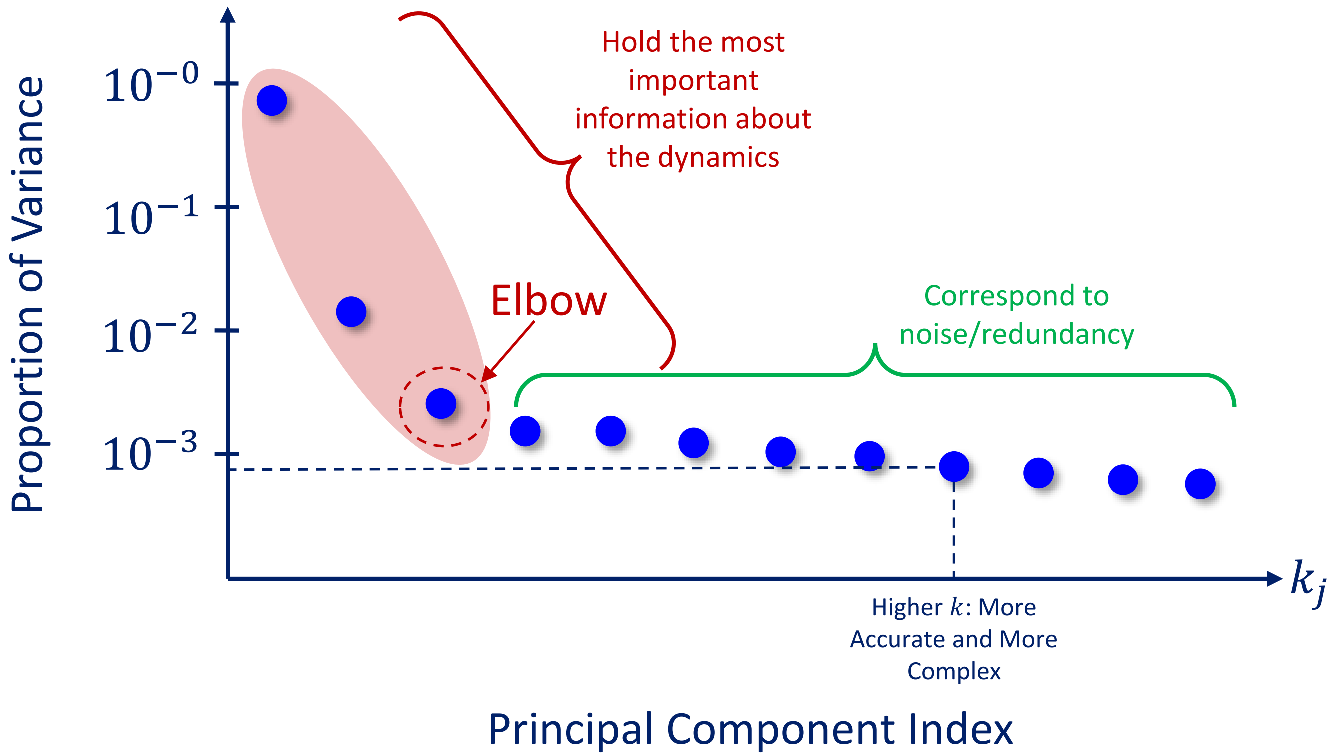
The first score reflects the oscillatory behaviour of the harmonic system and thus corresponds to the one degree of freedom of the dynamics

Choosing the Optimal Truncation Order

- The **elbow method** is a simple visual technique used to determine the optimal number of principal components (or eigenvalues) to keep in **PCA** or **SVD** for dimensionality reduction.

- It involves plotting the **explained variance** (or cumulative variance) against the **number of components**. As more components are added, the explained variance increases—but after a certain point, the rate of improvement sharply decreases, forming an "elbow" shape on the graph.

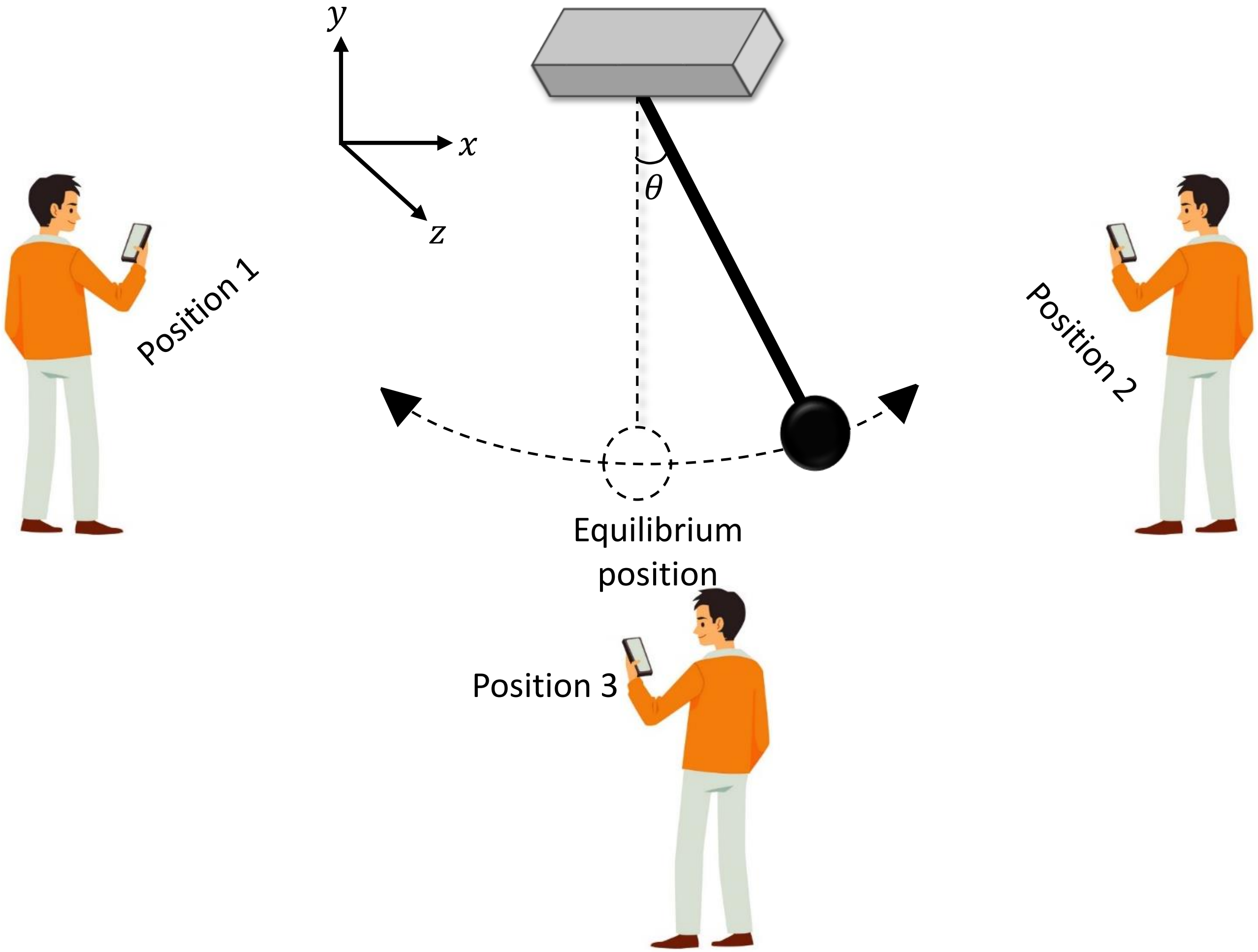
- The **elbow point** marks where adding more components yields little additional benefit, indicating the right balance between simplicity (fewer components) and accuracy (captured variance).



Let’s move on to MATLAB!

Tutorial on Tuesday 27th

Phyphox: Physical Phone Experiment Software.
Phyphox allows you to use the sensors in your phone for your experiments



To wrap up with a few reminders…

- **Your second tutorial will be on Tuesday 27th** . Four PGTAs will be with you during the session. Don’t spare any question! Ask them and they will be happy to help.
- All Lecture Material will be uploaded to Moodle later this day, along with the questions of the first tutorial .
- See you next Thursday 29th !