# Adaptive Signal Processing and Machine Intelligence

## Lecture 1 - Background


**Danilo Mandic**

room 813, ext: 46271


Department of Electrical and Electronic Engineering

Imperial College London, UK

d.mandic@imperial.ac.uk, URL: www.commsp.ee.ic.ac.uk/∼mandic

# Outline

## Background on:

- Linear algebra
- Estimation theory
- Gaussianity
- Bias-variance dilemma
- Sequential and block estimators
- Special matrices, matrix decompositions
- Phase space, attractor dynamics

# Is there such a thing as a nonlinear signal?

**Linear System**

System which obeys superposition and scaling principles:

$$f(ax + by) = a\ f(x) + b\ f(y)$$

**Nonlinear System**

System which does *not* obey these principles

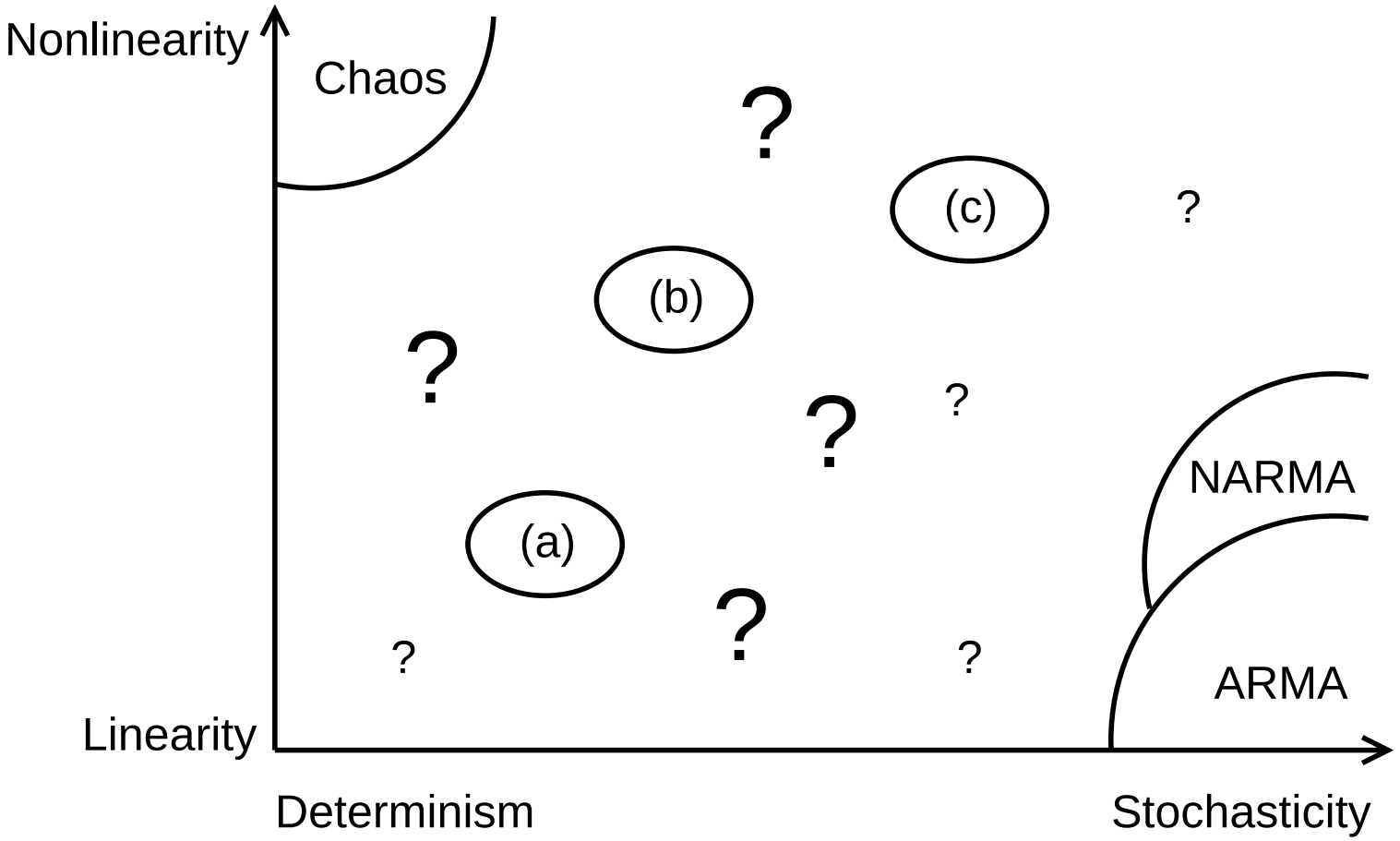
**Linear Signal**

Signal generated by a linear system driven by white (Gaussian) noise

**Nonlinear Signal**

**Any (!) signal which is not linear**

# Real world signals – 'Nonlinearity' and 'Stochasticity'

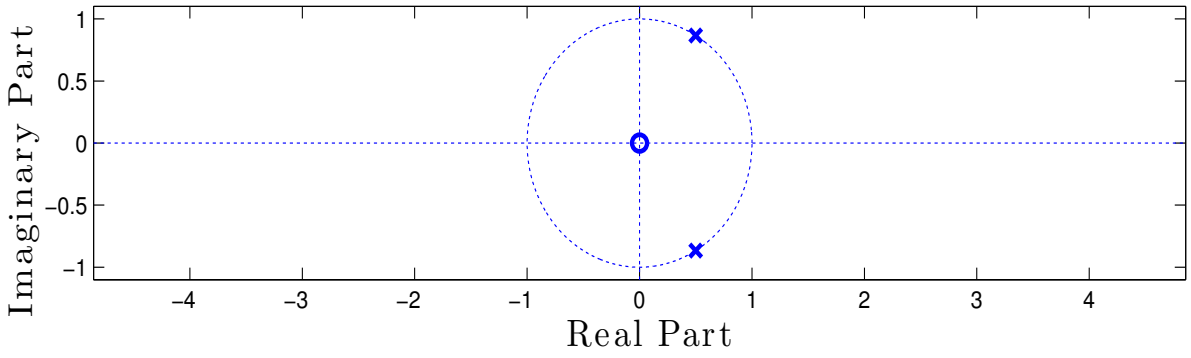


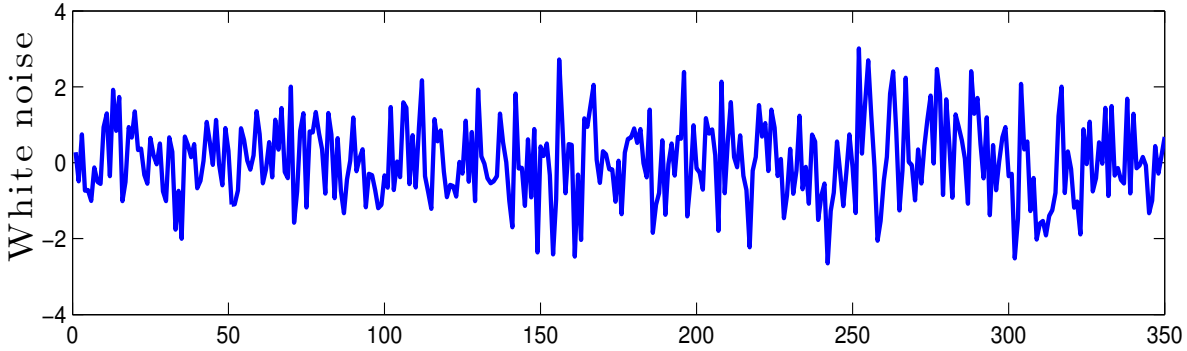

a) Periodic oscillations b) Small nonlinearity c) Route to chaos

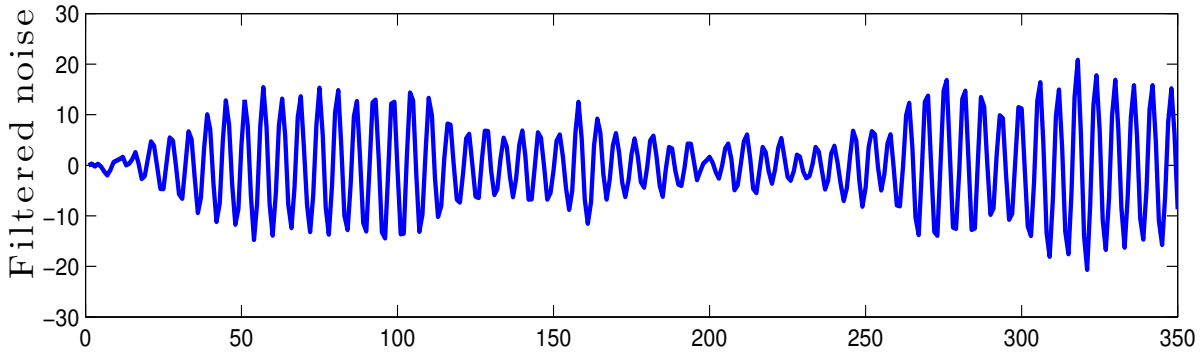
d) Route to chaos e) small noise f) HMM and others

# Example 1.1. So - how about a real-world sinewave?

## Spectral estimation or adaptive filtering to estimate it?




### Matlab code:

```
z1=0;
p1=[0.5+0.866i,0.5-0.866i];
[num1,den1]=zp2tf(z1,p1,1);
zplane(num1,den1);
s=randn(1,1000);
s1=filter(num1,den1,s);
figure;
subplot(311),plot(s),
subplot(313),plot(s1),
subplot(312),;
zplane(num1,den1)
```

### The AR model of a sinewave

```
x(k)=a1*x(k-1)+a2*x(k-2)+w(k)
a1=-1, a2=0.98, w~N(0,1)
```

# System nonlinearity detection

**Parametric System NL Testing**

- develop parametric nonlinear model
- drive unknown system with known input
- fit model parameters to match system output

**Nonparametric System NL Testing**

- drive unknown system with different stimuli
- test the superposition and scaling principles

1) apply short and long stimulus $\rightarrow R_{\mathrm{s}}$, $R_{\mathrm{l}}$

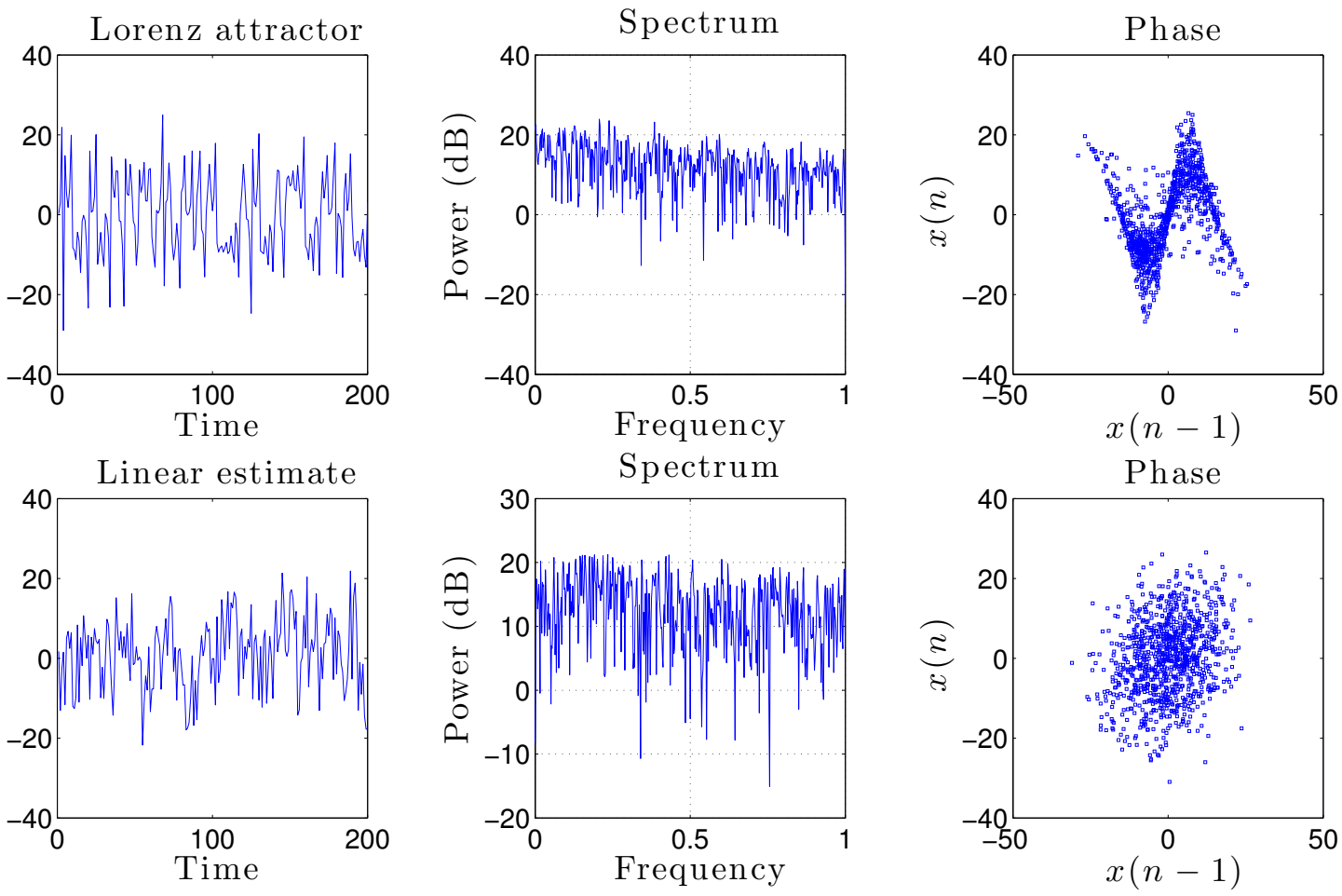
$\rightsquigarrow$ can $R_{\mathrm{l}}$ be predicted from time shifted $R_{\mathrm{s}}$

2) apply small and large amplitude stimulus $\rightarrow R_{\mathrm{sa}}$, $R_{\mathrm{la}}$

$\rightsquigarrow$ can $R_{\mathrm{la}}$ be predicted from superposition of $R_{\mathrm{sa}}$

# Example 1.2. Analysis in the phase space

## How do we differentiate between signals, second order statistics is not enough

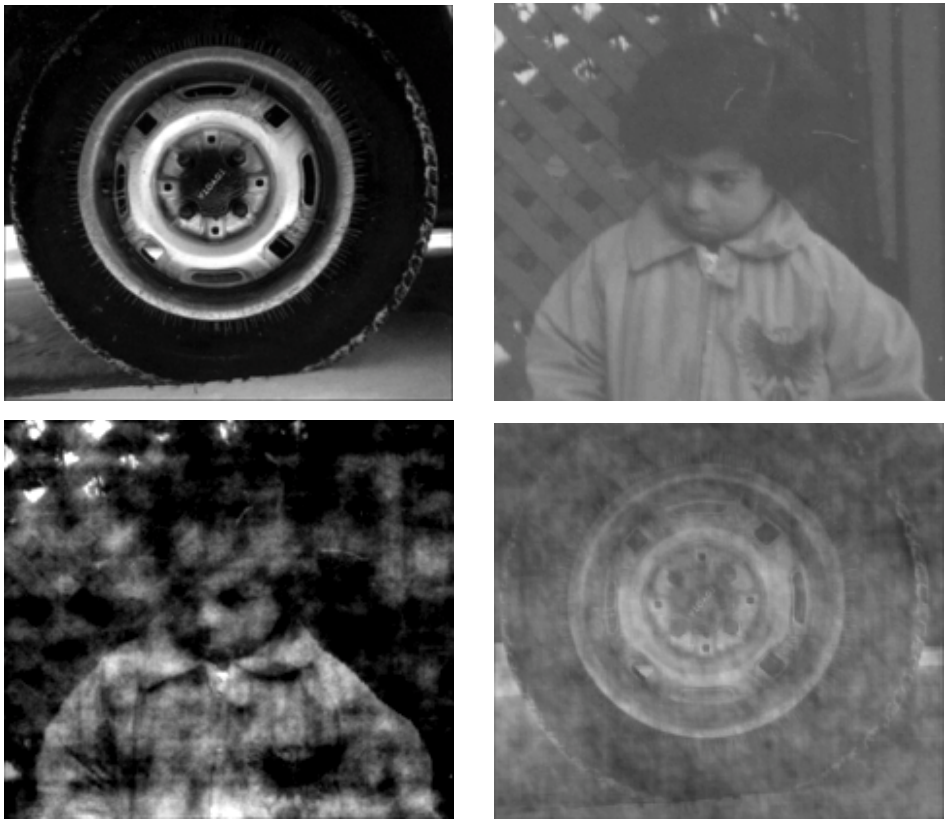



The chaotic Lorenz signal and its linear estimate have the same spectral properties.

The Lorenz signal has an attractor in phase space (higher order properties).

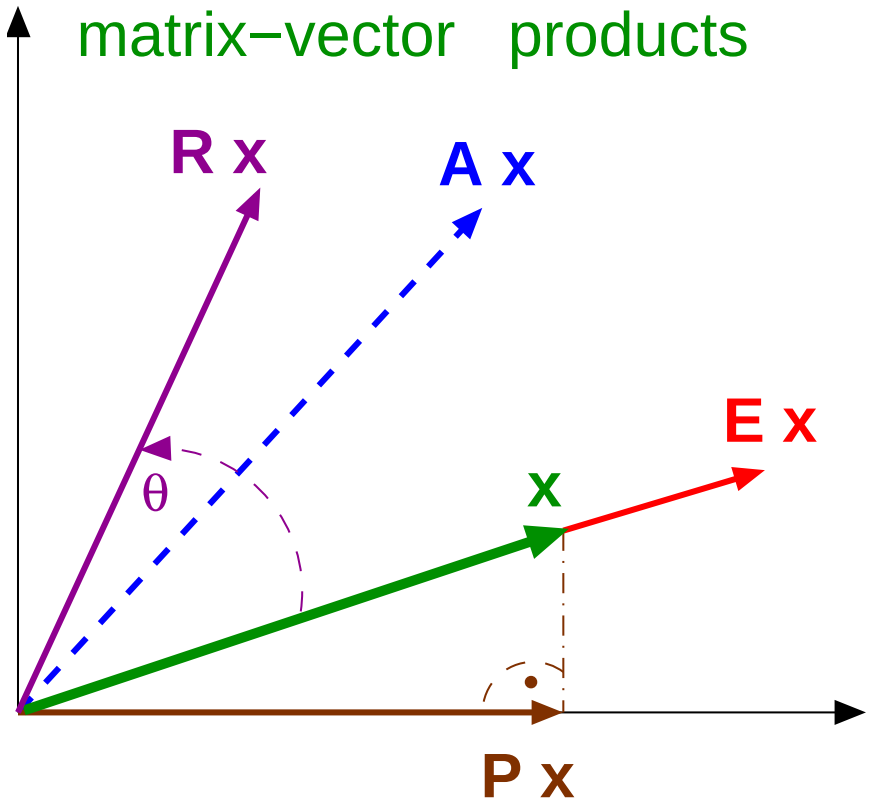
# Example: Human Visual System

## Importance of Phase Information



Surrogate images. *Top:* Original images $I_1$ and $I_2$; *Bottom:* Images $\hat{I}_1$ and $\hat{I}_2$ generated by exchanging the amplitude and phase spectra of the original images.

# What is that a matrix does to a vector?




**Ampli-twist:** a matrix $\mathbf{A}$ which multiplies a vector $\mathbf{x}$
(i) stretches or shortens the vector
(ii) rotates the vector

**A** $\rightsquigarrow$ any general matrix

**R** $\rightsquigarrow$ a rotation matrix ($\mathbf{R}^T = \mathbf{R}^{-1}$ and $\det \mathbf{R} = 1$)

$\mathbf{Ex} = \lambda \mathbf{x}$ $\rightsquigarrow$ eigenanalysis

**P** $\rightsquigarrow$ projection matrix

An example of a rotation matrix

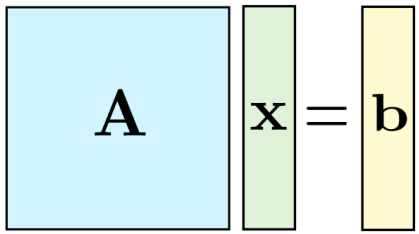
$$\mathbf{R} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}$$

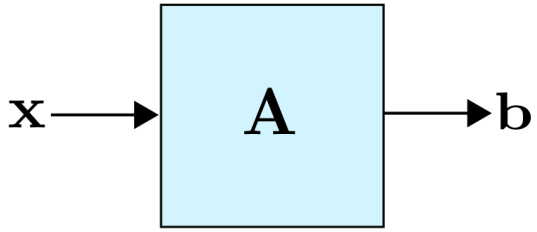
What can we say about the properties of the matrix $\mathbf{A}$, matrix $\mathbf{E}$ and the projection matrix $\mathbf{P}$ (rank, invertibility, ...)?

Is the projection matrix invertible?

# The meaning of eigenanalysis

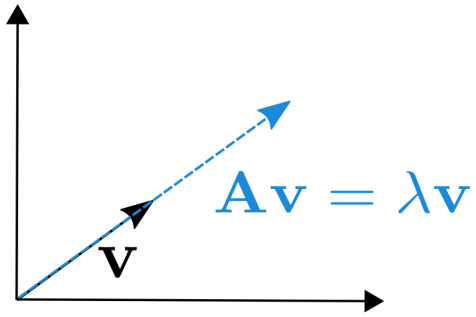
Let $\mathbf{A}$ be an $n \times n$ matrix, where $\mathbf{A}$ is a linear operator on vectors in $\mathbb{R}^n$, such that $\mathbf{A}\,\mathbf{x} = \mathbf{b}$

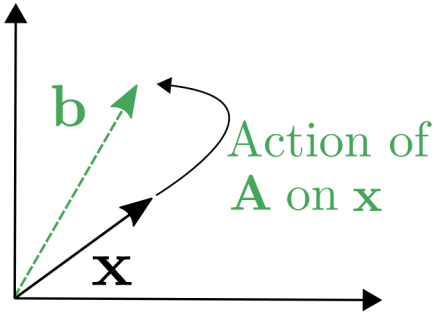



An **eigenvector** of $\mathbf{A}$ is a vector $\mathbf{v} \in \mathbb{R}^n$ such that $\mathbf{A}\,\mathbf{v} = \lambda\mathbf{v}$, where $\lambda$ is called the corresponding eigenvalue.

**Matrix $\mathbf{A}$ only changes the length of $\mathbf{v}$, not its direction!**




**Equation $\mathbf{A}\,\mathbf{v} = \lambda\mathbf{v}$** **Equation $\mathbf{A}\,\mathbf{x} = \mathbf{b}$.**

# Eigenvalues

For an $n \times n$ matrix $\mathbf{A}$, its **eigenvalues** are found from the $n$-th order polynomial in $\lambda$ defined by

$$\mathbf{A}\mathbf{x} = \lambda\mathbf{x} \quad \Rightarrow (\mathbf{A} - \lambda\mathbf{I})\,\mathbf{x} = \mathbf{0} \quad \Rightarrow \quad \text{nontrivial solution} \quad det\,(\mathbf{A} - \lambda\mathbf{I}) = 0$$

where $\mathbf{I}$ is the $n \times n$ identity matrix and $\lambda_1, \ldots, \lambda_n$ the eigenvalues.

The corresponding $n$ **eigenvectors**, $\mathbf{v}_1, \ldots, \mathbf{v}_n$, satisfy $\mathbf{A}\mathbf{v} = \lambda\mathbf{v}$ and are generally normalised to have unit norm $\| \mathbf{v} \|_2 = 1$.

For distinct eigenvalues, these eigenvectors are **linearly independent**.

A symmetric matrix is positive definite iff all its eigenvalues are positive

The **Spectral Theorem** allows for a symmetric matrix to be written as

$$\mathbf{A} = \sum_{i=1}^{n} \lambda_i \mathbf{v}_i \mathbf{v}_i^T$$

and the

$$\text{Trace}(\mathbf{A}) = \sum_{i=1}^{n} \lambda_i$$

Any connection with signal power?

## Example 1.4. More eigenanalysis

Let $\mathbf{A} = \begin{bmatrix} 2 & -4 \\ -1 & -1 \end{bmatrix}$. The characteristic polynomial $det(\mathbf{A} - \lambda\mathbf{I})$ is

$$\begin{aligned} p(\lambda) &= det\left(\begin{bmatrix} 2-\lambda & -4 \\ -1 & -1-\lambda \end{bmatrix}\right) \\ &= (2-\lambda)(-1-\lambda) - (-4)(-1) = \lambda^2 - \lambda - 6 = (\lambda - 3)(\lambda + 2) \end{aligned}$$

Thus the **eigenvalues** of $\mathbf{A}$ are $\lambda_1 = 3$ and $\lambda_2 = -2$.

To find **eigenvectors** $\mathbf{v} = [v_1, \ldots, v_n]^T$ corresponding to an eigenvalue $\lambda$

$$solve \qquad (\mathbf{A} - \lambda\mathbf{I})\,\mathbf{v} = 0 \qquad for \;\; \mathbf{v}$$

For $\lambda_1 = 3$ we thus have

$$\begin{bmatrix} 2-3 & -4 \\ -1 & -1-3 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad \Rightarrow \quad \mathbf{v}_1 = \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} = \begin{bmatrix} -4 \\ 1 \end{bmatrix}$$

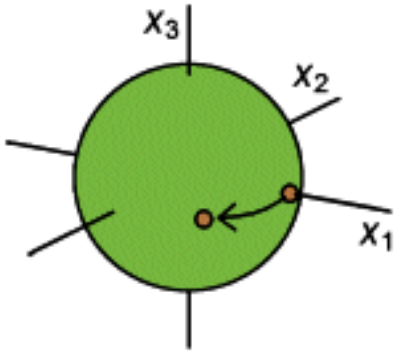
Similarly, for $\lambda_2 = -2$ we have $\mathbf{v}_2 = [1,\ 1]^T$, and thus $\mathbf{v}_1 \perp \mathbf{v}_2$.

$\mathbf{v}_1$ and $\mathbf{v}_2$ are **bases of the eigenspace spanned by these vectors**

## Example 1.5. Eigenanalysis ↬ An intuitive example

A sphere of unit radius is positioned at the centre of a three–dimensional coordinate system. It is rotating about the $x_3$ axis. This rotation[1] can be described by the matrix

$$\mathbf{C} = \begin{bmatrix} 0.707 & 0.707 & 0 \\ -0.707 & 0.707 & \\ 0 & 0 & 1 \end{bmatrix}$$



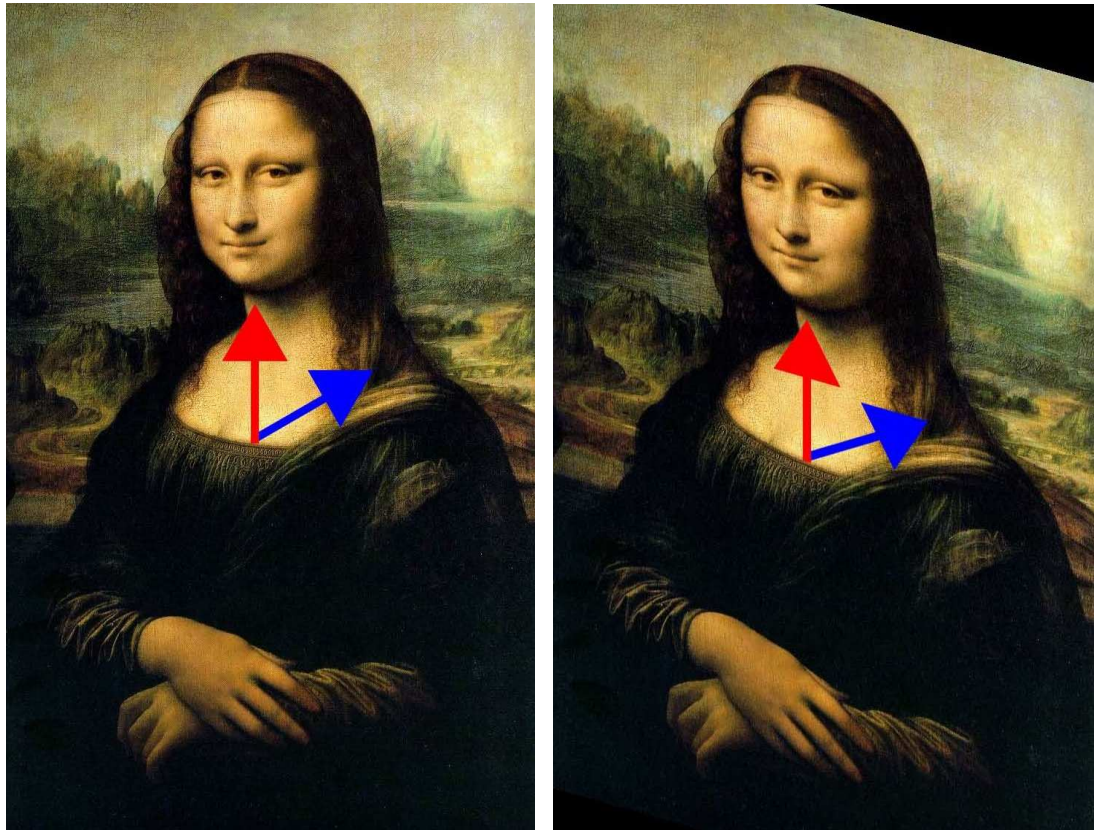

Intuitively, what is the eigenvector of $\mathbf{C}$? **Is there a point on the unit sphere that a $45^o$ rotation transforms into a multiple of itself?**

This is the north pole $[0, 0, 1]$ $\Rightarrow$ an eigenvector of $\mathbf{C}$ is the vector $[0, 0, 1]$.

[1] By $45^o$ in this case. For example, multiplying $\mathbf{C}$ by the vector $[1, 0, 0]$ yields the vector $[0.707, 0.707.0]$, which is rotated $45^o$.

# Example 1.6. Eigenanalysis – Image representation

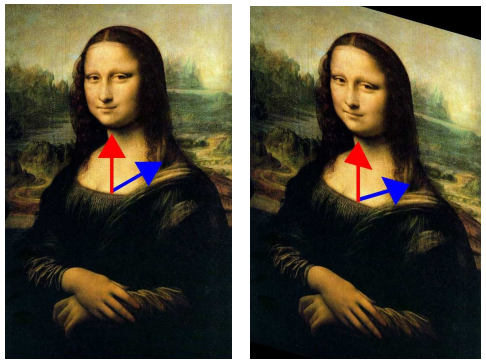
**Eigenvector** of a transformation $\Rightarrow$ a vector which, in the transformation, is multiplied by a constant factor, called the **eigenvalue of that vector**



The red vector is an eigenvector of the transformation, and the blue is not

# Example 1.6. Eigenanalysis of Mona Lisa

**The red vector was neither stretched nor compressed $\Rightarrow$ its eigenvalue is 1**



$$\text{Transformation} \qquad A = \begin{bmatrix} 1 & 0 \\ -\frac{1}{2} & 1 \end{bmatrix}$$

## Eigenvectors:

$$\mathbf{A}\mathbf{x} = \lambda\mathbf{x} \quad \Rightarrow \quad (\mathbf{A} - \lambda\mathbf{I})\,\mathbf{x} = \mathbf{0}$$

For non–trivial solutions $\rightarrow det(\mathbf{A} - \lambda\mathbf{I}) = 0 \quad \Rightarrow$

$$det\left(\begin{bmatrix} 1 & 0 \\ -\frac{1}{2} & 1 \end{bmatrix} - \lambda \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}\right) = det \begin{bmatrix} 1-\lambda & 0 \\ -\frac{1}{2} & 1-\lambda \end{bmatrix} = 0 \Rightarrow \lambda = 1$$

## Example 1.6. Eigenvectors for Mona Lisa

We have found $\lambda = 1$, the eigenvalue of matrix $\mathbf{A}$.

We can now solve for eigenvectors

$$\begin{bmatrix} 1-\lambda & 0 \\ -\frac{1}{2} & 1-\lambda \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} = \mathbf{0}$$

Substituting $\lambda = 1$ we have

$$\mathbf{v} = \begin{bmatrix} 0 \\ c \end{bmatrix}$$

where $c$ is an arbitrary constant.

**All vectors of this form, pointing straight up or down, are eigenvectors of $\mathbf{A}$**

In general $\mathbf{A}$ will have two distinct eigenvalues, and thus two distinct eigenvectors.

**Most vectors will have both their lengths and direction changed by $\mathbf{A}$ whereas eigenvectors will have only their lengths changed.**

# Example 1.7. Dynamical systems and eigenanalysis

**South Ken has two pizza places and $N \to \infty$ of pizza–loving students.**

◦ Suppose that 5000 people buy one pizza each every week.

Tony's Pizza place has the better pizza and $80\%$ of people who buy pizza each week at Tony's return the following week. Mike's Pizza does not have a good sauce and only $40\%$ of the customers return the following week.

We can represent this situation by a discrete dynamical system

$$\mathbf{x}_{n+1} = \mathbf{A}\,\mathbf{x}_n \quad where \qquad \mathbf{A} = \begin{bmatrix} 0.8 & 0.6 \\ 0.2 & 0.4 \end{bmatrix}$$

Let us start from $\mathbf{x}_0 = [2500,\ 2500]^T$, then we have

$$\mathbf{x}_1 = \begin{bmatrix} 3500 \\ 1500 \end{bmatrix}, \mathbf{x}_2 = \begin{bmatrix} 3700 \\ 1300 \end{bmatrix}, \mathbf{x}_3 = \begin{bmatrix} 3740 \\ 1260 \end{bmatrix}, \mathbf{x}_4 = \begin{bmatrix} 3748 \\ 1252 \end{bmatrix}, \mathbf{x}_5 = \begin{bmatrix} 3750 \\ 1250 \end{bmatrix}$$

**Also $\mathbf{x}_6 = \cdots = \mathbf{x}^* = \cdots = x_\infty = [3750, 1250]^T$**

Clearly, $\mathbf{x}^* = [3750, 1250]^T$ is the **eigenvector of $\mathbf{A}$, and $\mathbf{A}\,\mathbf{x}^* = \mathbf{x}^*$**

# Example 1.7. Will Mike have to close down?

## In Matlab

```
[v,lambda] = eig(A)          %also look at the demo 'eigshow(A)'

     0.9487   -0.7071
v =
     0.3162    0.7071

          1.0000         0
lambda =
          0              0.2000
```

$$\looparrowright \mathbf{v}_1 = [0.9487, 0.3162]^T, \ \mathbf{v}_2 = [-0.7071, 0.7071]^T, \quad \lambda_1 = 1, \ \lambda_2 = 0.2.$$

Notice that the elements of $\mathbf{v}_1$ are related as $3 \div 1$, the same as the ratio of Tony's and Mike's customers

**Since $\mathbf{x}_n = \mathbf{A}^n \mathbf{x}_0 \rightsquigarrow \mathbf{A}^n_{n\to\infty} = [\mathbf{v}_1 \vdots \mathbf{v}_2]$ $\looparrowright$ equilibrium!**

**This is closely related to fixed point theory since $\mathbf{A}$ is a Markov matrix**

# Most signal transforms can be represented as a matrix-vector multiplication

Consider a signal $\mathbf{x} = \begin{bmatrix} x_1 & x_2 & \dots & x_M \end{bmatrix}^T \in \mathbb{C}^{M\times 1}$. It can be represented with a different basis using

$$\mathbf{x} = \mathbf{F}\mathbf{w}$$

where $\mathbf{w} \in \mathbb{C}^{M\times 1}$The matrix $\mathbf{F} \in \mathbb{C}^{M\times M}$ can represent many different representations

- **Fourier basis:** Columns of $\mathbf{F}$ are sinusoids with different frequencies.
- **Wavelet basis:** Columns of $\mathbf{F}$ are wavelets.
- **Principal Component Analysis (PCA):** Rows of $\mathbf{F}$ are the eigenvectors of the covariance matrix of $\mathbf{F}$.

There are many other bases to represent signals which reside in a certain subspace of e.g. $\mathbb{R}^N$ or $\mathbb{C}^N$.

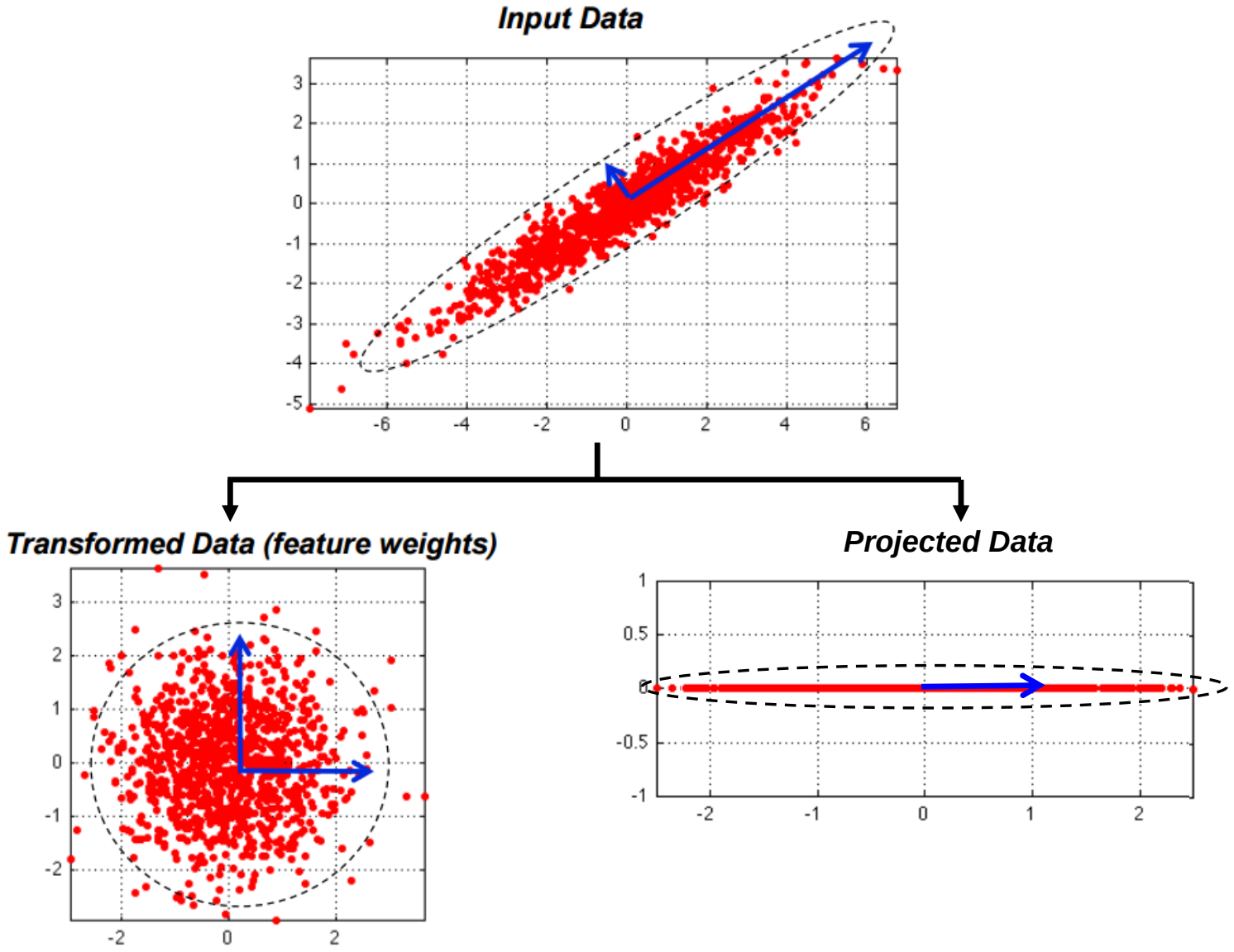
# Principal Component Analysis (PCA)

## Also known as the Karhunen-Loeve transform

- Many signal processing, control and machine learning tasks employ multivariate data which often exhibit dependencies and redundancies.
- For example, it is often useful to reduce the dimensionality of a signal while maintaining the useful information.
- This reduces the computational complexity of any algorithm while preserving the physical meaning of the data.
- Besides dimensionality reduction, we often would like to transform the multi-channel data such each channel is orthogonal to each other (the data covariance matrix is diagonal)
- We use the PCA to accomlish this goal $\rightarrow$ The PCA has been called one of the most valuable results from applied linear algebra.

# Principal Component Analysis (PCA)

## Geometric View



# Principal Component Analysis (PCA)

## Derivation

- Consider a general data vector, $\mathbf{x}_k \in \mathbb{C}^{M\times 1}$, with the empirical (sample) covariance matrix defined as

$$\text{cov}(\mathbf{x}_k) \overset{\text{def}}{=} \mathbf{R}_{\mathbf{x}} = \frac{1}{N}\sum_{i=0}^{N-1} \mathbf{x}_k \mathbf{x}_k^H .$$

- Also, if we define:

$$\mathbf{X} = \begin{bmatrix} \mathbf{x}_1 & \mathbf{x}_2 & \dots & \mathbf{x}_N \end{bmatrix} \in \mathbb{C}^{M\times N} \implies \mathbf{R}_{\mathbf{x}} = \tfrac{1}{N}\mathbf{X}\mathbf{X}^H$$

- The symmetric covariance matrix $\mathbf{R}_{\mathbf{x}}$ admits the following eigenvalue decomposition: $\mathbf{Q}^H \mathbf{R}_{\mathbf{x}} \mathbf{Q} = \mathbf{\Lambda}$

- The diagonal eigenvalue matrix, $\mathbf{\Lambda} = \text{diag}\{\lambda_1, \lambda_2, \dots, \lambda_M\}$, indicates the power of each component of $\mathbf{x}_k$.

- The matrix of eigenvectors, $\mathbf{Q}_r = [\mathbf{q}_1, \mathbf{q}_2, \dots, \mathbf{q}_M]$, designates the principal directions of the data.

# Principal Component Analysis (PCA)

## Derivation

- Suppose $\mathbf{x}_k$ is to be transformed into a vector, $\mathbf{u}_k \in \mathbb{C}^{M\times 1}$, using a linear transformation matrix $\mathbf{W}$, so that

$$\mathbf{u}_k = \mathbf{W}\mathbf{x}_k, \quad \text{where} \quad \text{cov}(\mathbf{u}_k) = \mathbf{I}.$$

- The PCA states that $\mathbf{W} = \mathbf{\Lambda}^{-\frac{1}{2}}\mathbf{Q}^H$ can be obtained from the eigenvector and eigenvalue matrices.

- Proof:

$$\begin{aligned}
\text{cov}(\mathbf{u}_k) &= \frac{1}{N}\sum_{i=0}^{N-1}\mathbf{u}_k\mathbf{u}_k^H \\
&= \mathbf{\Lambda}^{-\frac{1}{2}}\mathbf{Q}^H\left(\frac{1}{N}\sum_{i=0}^{N-1}\mathbf{x}_k\mathbf{x}_k^H\right)\mathbf{Q}\mathbf{\Lambda}^{-\frac{1}{2}} \\
&= \mathbf{\Lambda}^{-\frac{1}{2}}\mathbf{Q}^H\mathbf{R}_\mathbf{x}\mathbf{Q}\mathbf{\Lambda}^{-\frac{1}{2}} = \mathbf{I}.
\end{aligned}$$

# Principal Component Analysis (PCA)

## Dimensionality Reduction

- To perform dimensionality reduction, the PCA can be applied to obtain a transformed data vector $\mathbf{u}_{r,k} \in \mathbb{C}^{r\times 1}$ with the dimension $r < M$ as

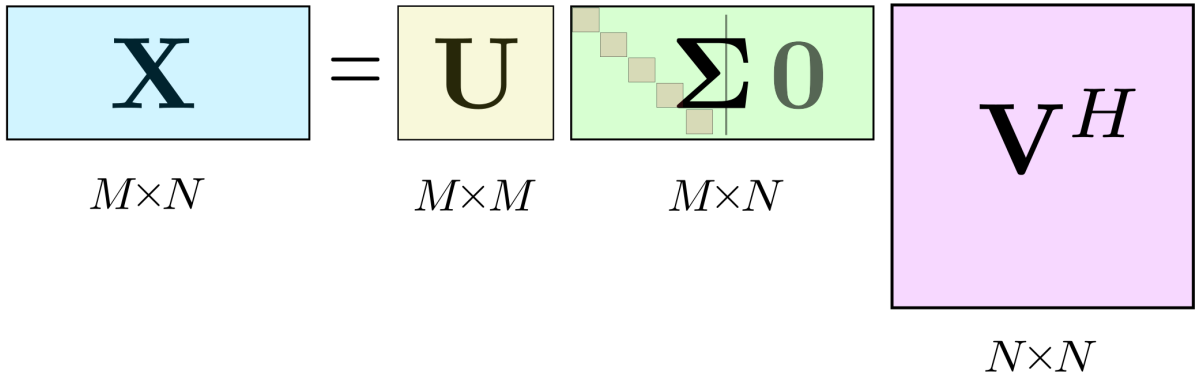
$$\mathbf{u}_{r,k} = \mathbf{W}_r \mathbf{x}_k = \mathbf{\Lambda}_{1:r}^{-\frac{1}{2}} \mathbf{Q}_{1:r}^H \mathbf{x}_k$$

- $\mathbf{\Lambda}_{1:r} = \text{diag}\{\lambda_1, \lambda_2, \ldots, \lambda_r\}$ and $\mathbf{Q}_{1:r} = [\mathbf{q}_1, \mathbf{q}_2, \ldots, \mathbf{q}_r]$
- Note: $r$ corresponds to the $r$ largest eigenvalues in $\mathbf{\Lambda}$.
- The PCA selects the **directions in which the data expresses maximal variance**, that is, the directions of the principal eigenvectors of the data matrix.
- The PCA matrix $\mathbf{W}_r$ can be interpreted as a projection matrix as we are unable to recover $x_k$ from the "reduced" data vector $\mathbf{u}_{r,k}$.
- The PCA projects the data onto the axes which exhibits the $r$-largest variances.

# Principal Component Analysis (PCA)

## Connections with Singular Value Decomposition (SVD)

- Consider the SVD of the data matrix $\mathbf{X} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^H$.




- The covariance matrix $\mathbf{R}_\mathbf{x} = \frac{1}{N}\mathbf{X}\mathbf{X}^H = \mathbf{U}\mathbf{\Sigma}^2\mathbf{U}^H$.

- Related to the eigenvalue decomposition: $\mathbf{Q}\mathbf{\Lambda}\mathbf{Q}^H = \mathbf{U}\mathbf{\Sigma}^2\mathbf{U}^H$

- So, PCA matrix $\mathbf{W} = \mathbf{\Lambda}^{-\frac{1}{2}}\mathbf{Q}^H$ can be obtained from the SVD of $\mathbf{X}$ as $\mathbf{W} = \mathbf{\Sigma}^{-1}\mathbf{U}^H \rightarrow$ **No need to compute $\mathbf{R}_\mathbf{x}$.**

# Linear Systems in Matrix Theory

**Solving a set of linear equations is necessary in the formulation of many algorithms for spectral estimation and adaptive signal processing!**

A set of linear equations is most conveniently represented in matrix form:

$$\mathbf{A}\mathbf{x} = \mathbf{b}$$

where the dimensions are

- $\mathbf{A}: \ m \times n \qquad \mathbf{x}: \ n \times 1 \qquad \mathbf{b}: \ m \times 1$

Depending on the values of $m$ and $n$ we have three special cases:

$$m = n \quad \looparrowright \quad \textbf{exactly determined case}$$

$$m < n \quad \looparrowright \quad \textbf{over-determined case}$$

$$m > n \quad \looparrowright \quad \textbf{under-determined case}$$

# The three cases: Possible solutions

$\circ$ $m = n \rightarrowtail$ **Exactly determined set of equations** If $\mathbf{A}$ is invertible or nonsingular, i.e. it has full column rank, with $m$ linearly independent columns, then

$$\mathbf{x} = \mathbf{A}^{-1}\mathbf{b}$$

If $\mathbf{A}$ is singular, then there may be either no solution or many solutions.

$\circ$ $m < n \rightarrowtail$ **There are more equations than unknowns (over-determined)** Therefore, the additional degrees of freedom can be used for enhanced accuracy in the presence of noise. A **least squares solution** that minimises $\| \mathbf{Ax} - \mathbf{b} \|_2^2$ yields

$$\mathbf{A}^T\mathbf{Ax} = \mathbf{A}^T\mathbf{b}$$

If $\mathbf{A^TA}$ is invertible, then

$$\mathbf{x} = (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\mathbf{b} \qquad \text{The } \textbf{least squares solution}$$

with minimum least squares error $\mathbf{b}^T\mathbf{b} - \mathbf{b}^T\mathbf{Ax}$.

$\circ$ $m > n \rightarrowtail$ **The under-determined case** (under-constrained system)
A solution: the **minimum norm solution which finds** $\min \| \mathbf{x} \|_2^2$ such that $\mathbf{Ax} = \mathbf{b}$. Provided that the rows of $\mathbf{A}$ are linearly independent (and $\mathbf{AA}^T$ is invertible), we have

$$\mathbf{x} = \mathbf{A}^T \left(\mathbf{AA}^T\right)^{-1} \mathbf{b}$$

where $\mathbf{A}^T \left(\mathbf{AA}^T\right)^{-1}$ is the pseudo-inverse of $\mathbf{A}$. (we can also use Lagrange multipliers)

# Special matrix forms

**The exchange matrix** (or reflection matrix)

$$\mathbf{J} = \begin{bmatrix} 0 & \cdots & 0 & 1 \\ 0 & \cdots & 1 & 0 \\ \vdots & \ddots & \vdots & \vdots \\ 1 & \cdots & 0 & 0 \end{bmatrix} \text{ - notice } \mathbf{J} \times \mathbf{J} = \mathbf{I}$$

$\Rightarrow$ $\mathbf{J}$ is its own inverse; and $\mathbf{J}^T\mathbf{A}\mathbf{J}$ reflects each element of the square matrix $\mathbf{A}$ about its central element.

**Toeplitz matrix** $\looparrowright$tremendous amount of structure (constant diagonals, $a_{i,j} = a_{i+1,j+1}$)

**Examples:** Toeplitz $\mathbf{T} = \begin{bmatrix} 1 & 3 & 5 & 7 \\ 2 & 1 & 3 & 5 \\ 4 & 2 & 1 & 3 \\ 6 & 4 & 2 & 1 \end{bmatrix}$ Hankel $\mathbf{H} = \begin{bmatrix} 1 & 3 & 5 & 7 \\ 3 & 5 & 7 & 4 \\ 5 & 7 & 4 & 2 \\ 7 & 4 & 2 & 1 \end{bmatrix}$

Notice that all entries are defined once the first column and first row have been specified. A **convolution** matrix is an example of a Toeplitz matrix.

**Hankel matrix** $\looparrowright$ similar to Toeplitz for which $a_{i,j} = a_{i-1,j+1}$ (upside-down Toeplitz). An example of a Hankel matrix is the exchange matrix. Also, $\mathbf{H} = \mathbf{T}\mathbf{J}$.

# Inverse matrix

## We need to calculate those inverses for our main models

**The structure of an inverse matrix is also important!**

- A symmetric matrix has a symmetric inverse
- A **Toeplitz** matrix has a persymmetric inverse (symmetric about the cross diagonal)
- A Hankel matrix has a symmetric inverse (useful in harmonic retrieval)

A useful formula for matrix inversion is **Woodbury's identity** (matrix inversion Lemma)

$$\left(\mathbf{A}+\mathbf{U}\mathbf{V}^{\mathbf{T}}\right)^{-\mathbf{1}}=\mathbf{A}^{-\mathbf{1}}-\left[\mathbf{A}^{-\mathbf{1}}\mathbf{U}\left(\mathbf{I}+\mathbf{V}^{\mathbf{T}}\mathbf{A}^{-\mathbf{1}}\mathbf{U}\right)^{-\mathbf{1}}\mathbf{V}^{\mathbf{T}}\mathbf{A}^{-\mathbf{1}}\right]$$

If $\mathbf{u}$ and $\mathbf{v}$ are vectors, then this simplifies into

$$(\mathbf{A}+\mathbf{u}\mathbf{v}^{T})^{-1}=\mathbf{A}^{-1}-\frac{\mathbf{A}^{-1}\mathbf{u}\mathbf{v}^{T}\mathbf{A}^{-1}}{1+\mathbf{v}^{T}\mathbf{A}^{-1}\mathbf{u}}$$

which will be important in the derivation of adaptive algorithms.

In a special case $\mathbf{A}=\mathbf{I}$ and we have

$$(\mathbf{I}+\mathbf{u}\mathbf{v}^{T})^{-1}=\mathbf{I}-\frac{\mathbf{u}\mathbf{v}^{T}}{1+\mathbf{v}^{T}\mathbf{u}}$$

## Example 1.3. Matrix inversion lemma ↬ another approach

Construct an 'augmented' matrix $[A, B; C, D]$ and its inverse $[E, F; G, H]$, so that

$$\begin{bmatrix} A & B \\ C & D \end{bmatrix}^{-1} = \begin{bmatrix} E & F \\ G & H \end{bmatrix}$$

**Consider the following two products**

$$\begin{bmatrix} A & B \\ C & D \end{bmatrix}\begin{bmatrix} E & F \\ G & H \end{bmatrix} = \begin{bmatrix} AE+BG & AF+BH \\ CE+HC & CF+DH \end{bmatrix} = \begin{bmatrix} I & 0 \\ 0 & I \end{bmatrix}$$

and

$$\begin{bmatrix} E & F \\ G & H \end{bmatrix}\begin{bmatrix} A & B \\ C & D \end{bmatrix} = \begin{bmatrix} EA+FC & EB+FD \\ GA+HC & GB+HD \end{bmatrix} = \begin{bmatrix} I & 0 \\ 0 & I \end{bmatrix}$$

Combine the corresponding terms to get another form

$$\left(\mathbf{A}-\mathbf{B}\mathbf{D}^{-1}\mathbf{C}\right)^{-1} = \mathbf{A}^{-1} + \mathbf{A}^{-1}\mathbf{B}\left(\mathbf{D}-\mathbf{C}\mathbf{A}^{-1}\mathbf{B}\right)^{-1}\mathbf{C}\mathbf{A}^{-1}$$

# Estimation theory: Recap of basic notions

**Problem:** To estimate one or more parameters from some given discrete-time signal $\{\mathbf{x}[n]\}$.

**Mathematical Context:**

Given an $N$-point data set which depends upon an unknown parameter $\theta$ (scalar), define an "estimator" as some function $g$ of the dataset, that is

$$\hat{\theta} = g\big(x[0], x[1], \ldots, x[N-1]\big)$$

Note: difference between estimator and estimate

- Estimate = particular value of $\hat{\theta}$ for one observation data set
- Estimator = rule that assigns a value of $\hat{\theta}$ for a given realisation of

$$\mathbf{x} = \big[x[0], x[1], \ldots, x[N-1]\big]$$

There are two types of estimators

- Classical: the unknown parameter(s) is(are) fixed
- Bayesian: the unknown parameter(s) is(are) random

# The bias–variance dilemma

The mean square error (MSE) of an estimate $\hat{\theta}$ of a parameter $\theta$ is given by

$$MSE(\hat{\theta}) = E\{(\hat{\theta} - \theta)^2\} \qquad \text{average deviation from the true value}$$

For every estimator: **Bias:** $B = E\{\hat{\theta}\} - \theta$ **Variance:** $\text{var} = E\{(\hat{\theta} - E\{\hat{\theta}\})^2\}$

**Therefore:**

$$\begin{aligned}
\text{MSE} &= E\{(\hat{\theta} - \theta)^2\} = E\{[\hat{\theta} - E\{\hat{\theta}\} + \underbrace{E\{\hat{\theta}\} - \theta}_{\text{bias } B(\hat{\theta})}]^2\} \\
&= E\{[\hat{\theta} - E\{\hat{\theta}\}]^2\} + E\{B^2(\hat{\theta})\} + 2E\{[\hat{\theta} - E\{\hat{\theta}\}]B(\hat{\theta})\} \\
&\quad \text{due to the linearity of the } E\{\cdot\} \text{ operator and that } E\{B(\hat{\theta})\} = B(\hat{\theta}) \\
&= E\{[\hat{\theta} - E\{\hat{\theta}\}]^2\} + B^2(\hat{\theta}) + \underbrace{2E\{[\hat{\theta} - E\{\hat{\theta}\}]\}}_{=0,\ \text{the } E\{\hat{\theta}\} \text{ are equal}} B(\hat{\theta}) \\
&= \text{var}(\hat{\theta}) + B^2(\hat{\theta}) \qquad \text{MSE depends both on bias and variance}
\end{aligned}$$

# Example 1.8. Types of estimators

## Window length matters in sequential estimation too!

**Block based and sequential:**

Example: block based - **sample mean:** $\hat{\theta}_{N-1} = \frac{1}{N}\sum_{k=0}^{N-1} x[k]$

It is block based since all the data are required to form an estimate.

**Sequential estimator:** A new estimate can be calculated as each new sample arrives.

$$\text{we want to express } \hat{\theta}(N) = f\left(\hat{\theta}(N-1), x(N)\right)$$

$$\hat{\theta}[N] = \frac{1}{N+1}\sum_{k=0}^{N} x[k] = \frac{1}{N+1}\left[\sum_{k=0}^{N-1} x[k] + x[N]\right]$$

$$= \frac{N}{N+1}\hat{\theta}[N-1] + \frac{1}{N+1}x[n]$$

$$= \left(1 - \frac{1}{N+1}\right)\hat{\theta}[N-1] + \frac{1}{N+1}x[n]$$

$$= \hat{\theta}[N-1] + \frac{1}{N+1}\left[x[N] - \hat{\theta}[N-1]\right]$$

**New quantity = old quantity + Gain × Error**

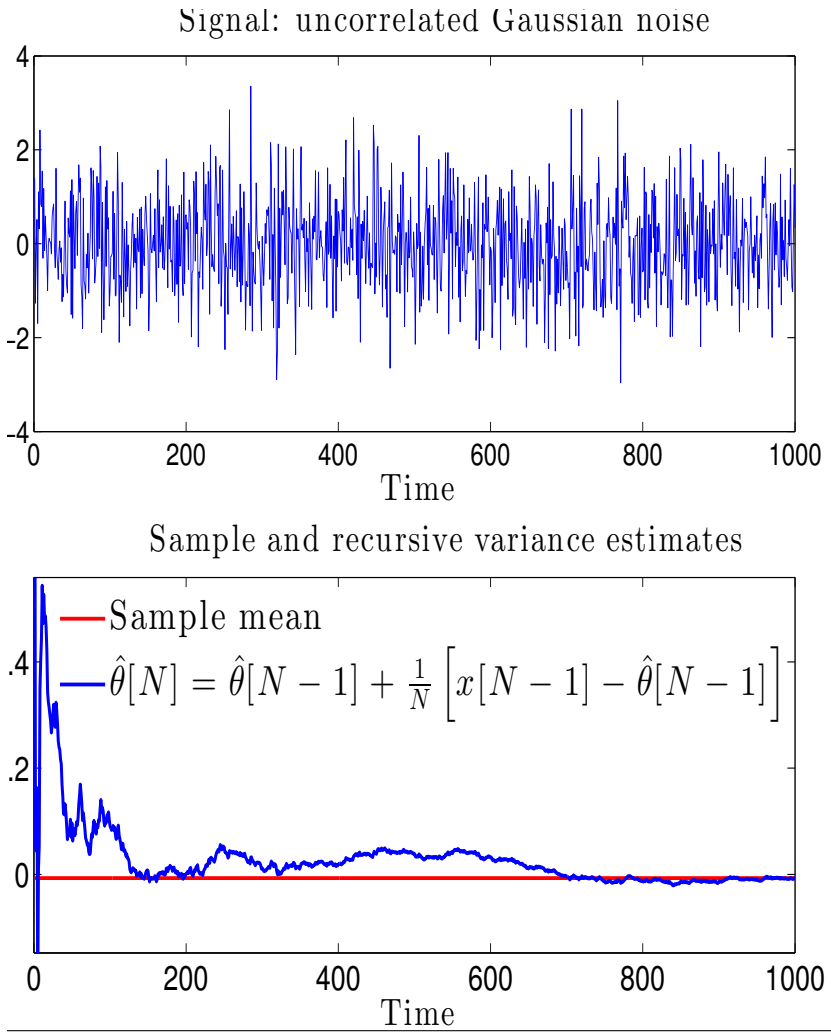
$\frac{1}{N+1}$ = gain, $x[N] - \hat{\theta}[N-1]$ = error

**This update equation is the common form of all adaptive algorithms:** the LMS (Least mean square), NLMS (Normalised LMS), Kalman filtering.

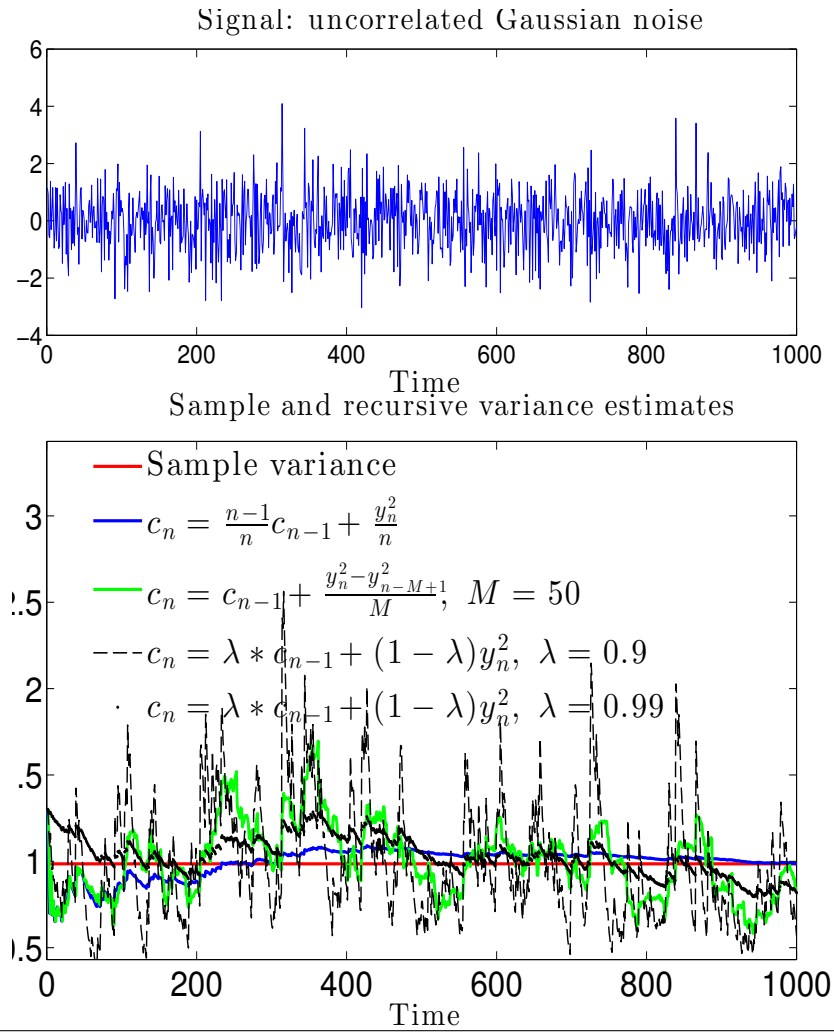
# Example 1.9. Convergence of sequential estimators

## For simplicity we consider a Gaussian $\sim \mathcal{N}(0,1)$

### Sequential DC level estimation.
### *(mean of a Gaussian signal)*




### Sequential variance estimation.
### *(variance of a Gaussian)*



# Data model: Gaussianity

## Starting From Real-valued Data

Why Gaussian? **Justification:** **Central Limit Theorem**

If we form a sum of independent measurements

$\Rightarrow$ the distribution of the sum tends to a Gaussian distribution

$$p(x) = \frac{1}{\sqrt{2\pi\sigma_x^2}} e^{-\frac{(x-\mu_x)^2}{2\sigma_x^2}} \qquad x \sim \mathcal{N}(\mu_x, \sigma_x^2)$$

$\Rightarrow$ **distribution defined by its mean and variance!!! SOS**

If $x \sim \mathcal{N}(0, \sigma_x^2)$ then $E\{x^{2n-1}\} = 1, 3, \ldots, (2n-1)\sigma_x^{2n}, \quad \forall n$

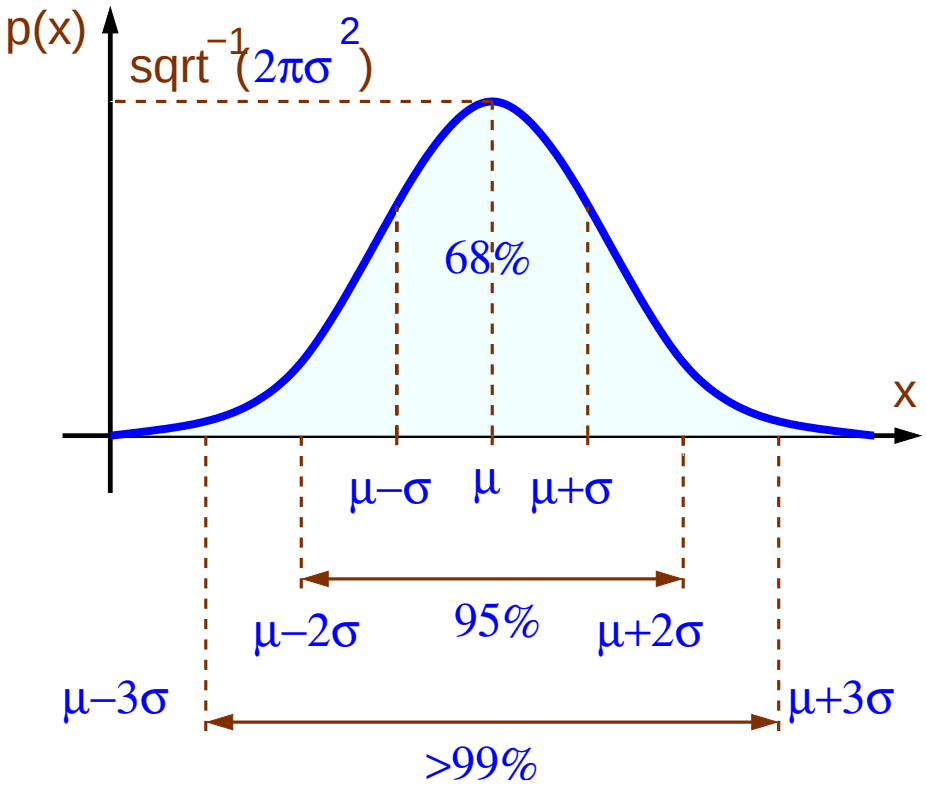
In the vector case ($N$ Gaussian random variables)

$$p\big(x[0], x[1], \ldots, x[N-1]\big) = \frac{1}{(2\pi)^{N/2} det(\mathbf{C}_{xx})^{1/2}} e^{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_x)^T \mathbf{C}_{xx}^{-1}(\mathbf{x}-\boldsymbol{\mu}_x)}$$

where $\mathbf{C}_{xx} = E\{(\mathbf{x}-\boldsymbol{\mu}_x)(\mathbf{x}-\boldsymbol{\mu}_x)^T\}$ is the **covariance matrix.**

# Data model: Gaussianity

## Properties of a Gaussian




1) If x and y are jointly Gaussian, then for any constants $a$ and $b$ the random variable

$$z = ax + by$$

is Gaussian with mean

$$m_z = am_x + bm_y$$

and variance

$$\sigma_z^2 = a^2\sigma_x^2 + b^2\sigma_y^2 + 2ab\sigma_x\sigma_y\rho_{xy}$$

2) If two jointly Gaussian random variables are *uncorrelated* ($\rho_{xy} = 0$) then they are also statistically independent, that is

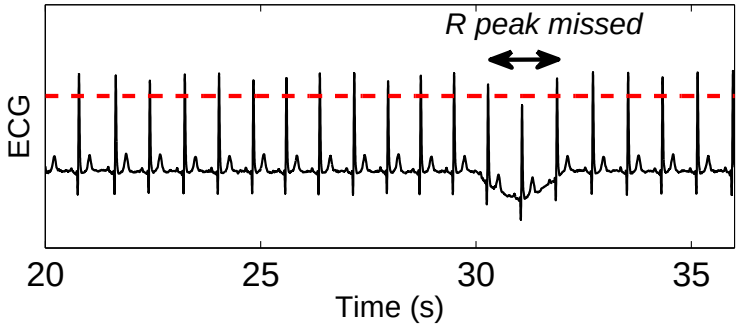
$$f_{x,y} = f(x)f(y)$$

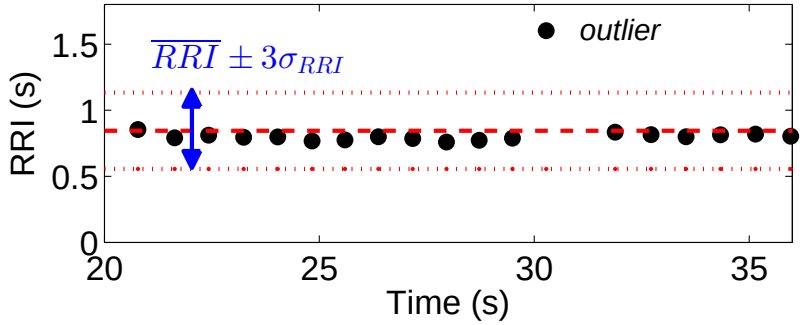
**For $\mu = 0$, $\sigma = 1$, the inflection points are $\pm 1$.**

# Data model: Connection with your Coursework

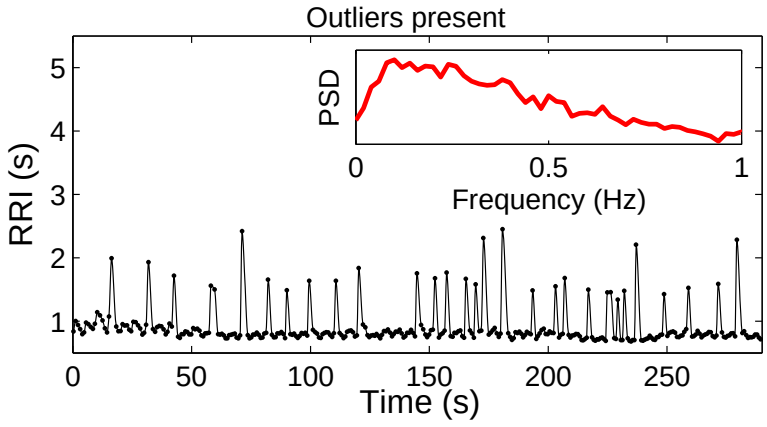
## Rejecting outliers from cardiac data using Properties of a Gaussian

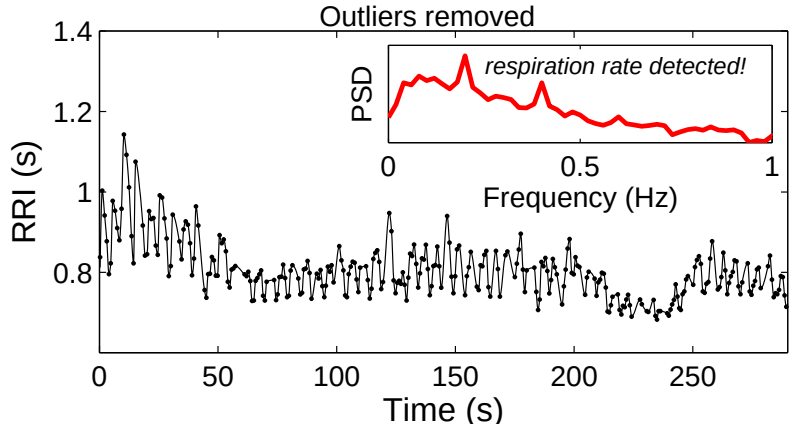
- Failed detection of R peaks in ECG [left] causes outliers in R-R interval (RRI, time difference between consecutive R peaks) [right]



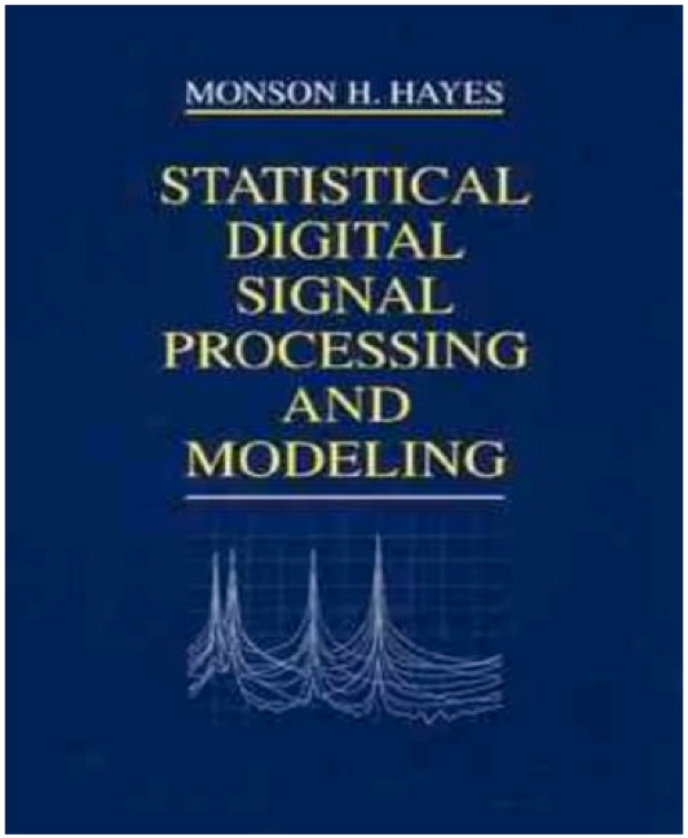

- No clear outcome from PSD analysis of outlier-compromised RRI [left], but PSD of RRI with outliers removed reveals respiration rate [right]



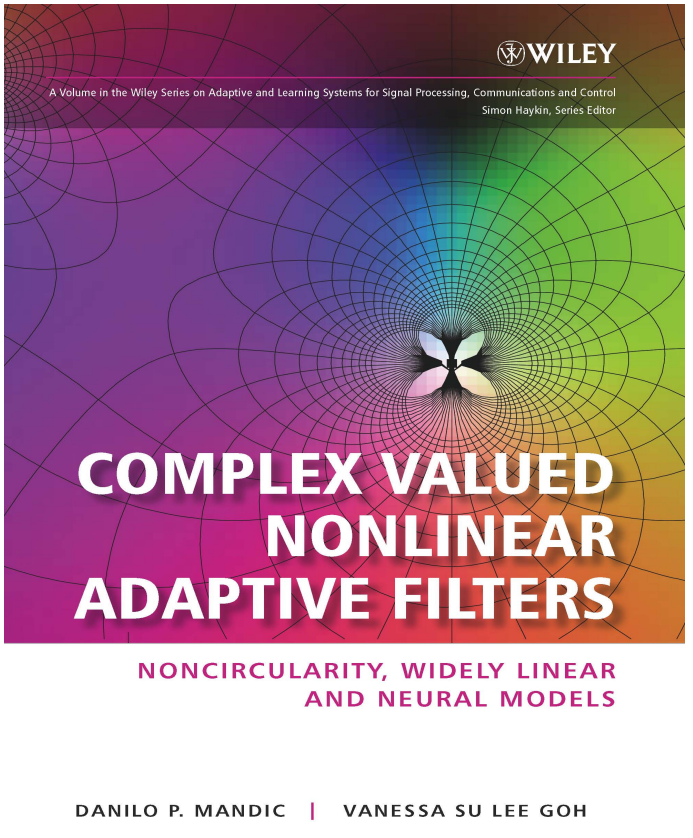
# Textbooks: Recommended

**M. Hayes (*Statistical Signal Processing*, several editions)**




spectral estimation and part of adaptive filters

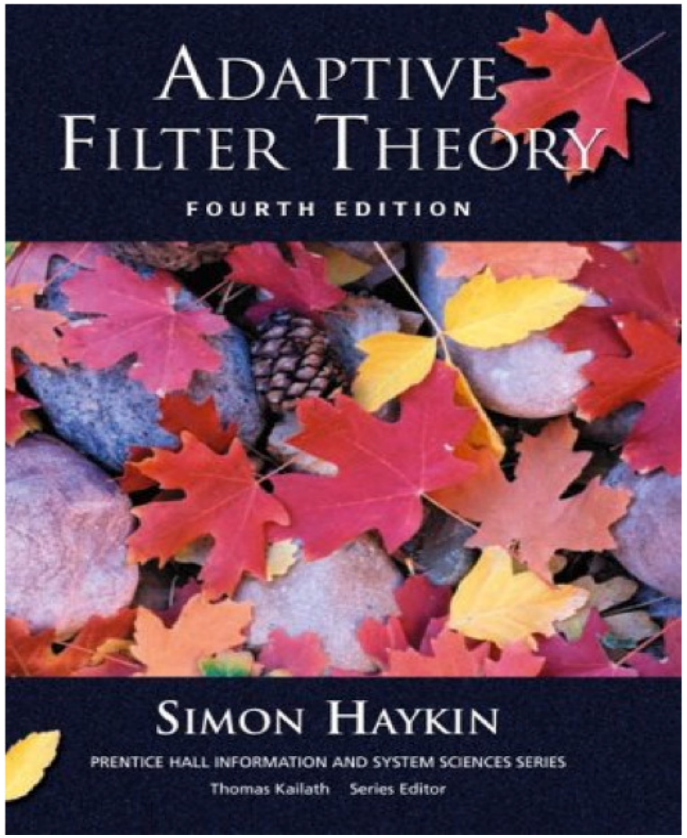
**D. Mandic and S. Goh (*Complex Adaptive Filters*, Wiley 2009).**




real, complex, and neural adaptive filters

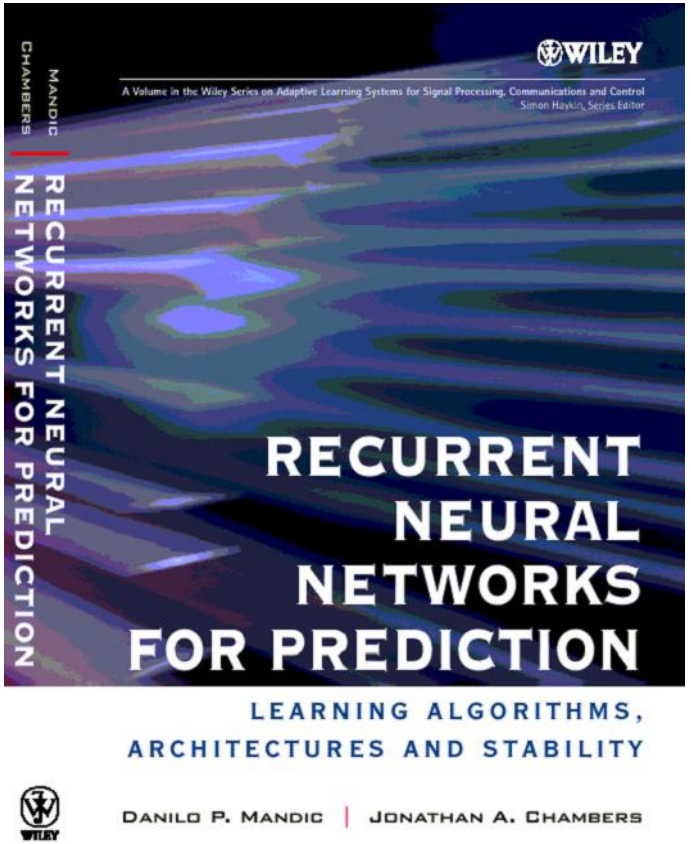
# Textbooks: Additional reading

**S. Haykin (*Adaptive Filtering Theory*, several editions)**




a comprehensive account of adaptive filters

**D. Mandic & J. Chambers (*RNNs for Prediction*, Wiley 2001).**




feedback and neural network architectures

## Summary

- Several background concepts in one place
- The course is self–contained, most of the background should be found in this Lecture
- For more detail about spectrum estimation, refer to Hayes' book
- For more detail about adaptive filtering, refer to Haykin's book
- We will frequently come back to this lecture, when using these techniques in spectral estimation and adaptive signal processing
- Towards the end of the course, we will combine spectrum estimation and adaptive signal processing towards an application in brain computer interface (BCI)

**We will next look into ideas in Complex-valued Signal Processing**

# Appendix: Taylor series expansion (TSE)

Most 'smooth' functions can be expanded into their Taylor Series Expansion (TSE)

$$f(x) = f(x_0) + \frac{f'(x_0)}{1}(x - x_0) + \frac{f''(x_0)}{2!}(x - x_0)^2 + \cdots = \sum_{n=1}^{\infty} \frac{f^{(n)}(x_0)}{n!}$$

To show this consider the polynomial

$$f(x) = a_0 + a_1(x - x_0) + a_2(x - x_0)^2 + a_3(x - x_0)^3 + \cdots$$

1. To get $a_0$ $\quad\rightarrowtail\quad$ choose $x = x_0$ $\quad\Rightarrow\quad$ $a_0 = f(x_0)$
2. To get $a_1$ $\quad\rightarrowtail\quad$ take derivative of the polynomial above to have

$$\frac{d}{dx}f(x) = a_1 + 2a_2(x - x_0) + 3a_3(x - x_0)^2 + 4a_4(x - x_0)^4 + \cdots$$

choose $x = x_0$ $\quad\Rightarrow\quad$ $a_1 = \frac{df(x)}{dx}_{|x=x_0}$ and so on ... $a_k = \frac{1}{k!}\frac{d^k f(x)}{dx^k}_{|x=x_0}$

# Appendix: Power series - contd.

Consider

$$f(x)=\sum_{n=0}^{\infty}a_n x^n \;\Rightarrow\; f'(x)=\sum_{n=1}^{\infty}n a_n x^{n-1} \text{ and } \int_0^x f(t)dt=\sum_{n=0}^{\infty}\frac{a_n}{n+1}x^{n+1}$$

1. Exponential function, cosh, sinh, sin, cos, ...

$$e^x=\sum_{n=0}^{\infty}\frac{x^n}{n!} \quad \text{and} \quad e^{-x}=\sum_{n=0}^{\infty}(-1)^n\frac{x^n}{n!} \quad \Rightarrow \quad \frac{e^x-e^{-x}}{2}=\sum_{n=0}^{\infty}\frac{x^{2n}}{(2n)!}$$

2. other useful formulas

$$\sum_{n=0}^{\infty}x^n=\frac{1}{1-x} \quad \Rightarrow \quad \sum_{n=1}^{\infty}nx^{n-1}=\frac{1}{(1-x)^2} \quad \text{and} \quad \frac{1}{1+x^2}=\sum_{n=0}^{\infty}(-1)^n nx^2$$

Integrate to obtain $atan(x)=\sum_{n=0}^{\infty}(-1)^n\frac{x^{2n+1}}{2n+1}$. For $x=1$ we have $\frac{\pi}{4}=1=1/3+1/5-1/7+\cdots$

# Appendix: Numerical differentiation $\hookrightarrow$ Some examples

- $f' = \frac{f(1)-f(0)}{h}$
- $f' = \frac{f(0)-f(1)}{h}$
- $f' = \frac{f(1)-2f(0)+f(-1)}{2h}$
- $f'' = \frac{f(1)-2f(0)+f(-1)}{h^2}$
- $f' = \frac{f(-2)-8f(-1)+8f(1)-f(2)}{12h}$
- $f'' = \frac{-f(-2)+16f(-1)-30f(0)+16f(1)-f(2)}{12h^2}$

# Appendix: Second order data modelling

**AR(p) model:** $x(n) = a_1 x(n-1) + \cdots + a_p x(n-p) = \mathbf{a}^T\mathbf{x} =< \mathbf{a}, \mathbf{x} >$

**We start from:** $y(n) = a_1(n)x(n-1) + a_2(n)x(n-2) + \cdots + a_p(n)x(n-p)$

**Teaching signal:** $d(n)$ **Output error:** $e(n) = d(n) - y(n)$

**Fixed coeff.** $\mathbf{a}$ & $x(n) = y(n)$

**Autoregressive modelling**

$$\begin{aligned} r_{xx}(1) &= a_1 r_{xx}(0) + \cdots + a_p r_{xx}(p-1) \\ r_{xx}(2) &= a_1 r_{xx}(1) + \cdots + a_p r_{xx}(p-2) \\ \vdots &= \vdots \\ r_{xx}(p) &= a_1 r_{xx}(p-1) + \cdots + a_p r_{xx}(0) \\ \cdots & \cdots \\ \mathbf{r}_{xx} &= \mathbf{R}_{xx}\mathbf{a} \end{aligned}$$

**Solution:** $\mathbf{a} = \mathbf{R}_{xx}^{-1}\mathbf{r}_{xx}$

**Yule–Walker equation**

**Fixed optimal coeff.** $\mathbf{w}_o = \mathbf{a}_{opt}$

$$J = E\{\frac{1}{2}e^2(n)\} = \sigma_d^2 - 2\mathbf{w}^T\mathbf{p} + \mathbf{w}^T\mathbf{R}\mathbf{w}$$

is quadratic in $\mathbf{w}$ and for a full rank $\mathbf{R}$, it has **one unique minimum.**

**Now:**

$$\frac{\partial J}{\partial \mathbf{w}} = -\mathbf{p} + \mathbf{R}\cdot\mathbf{w} = \mathbf{0}$$

**Solution:** $\mathbf{w}_o = \mathbf{R}^{-1}\mathbf{p}$

**Wiener–Hopf equation**

# Matrix inversion lemma ↬ A handy derivation

**General form:**

$$\begin{aligned}(\mathbf{A}+\mathbf{BCD})^{-1} &= (\mathbf{A}[\mathbf{I}+\mathbf{A}^{-1}\mathbf{BCD}])^{-1} = [\mathbf{I}+\mathbf{A}^{-1}\mathbf{BCD}]^{-1}\mathbf{A}^{-1} \\ &= [\mathbf{I}-(\mathbf{I}+\mathbf{A}^{-1}\mathbf{BCD})^{-1}\mathbf{A}^{-1}\mathbf{BCD}]\mathbf{A}^{-1} \qquad \text{(using P1)} \\ &= \mathbf{A}^{-1}-(\mathbf{I}+\mathbf{A}^{-1}\mathbf{BCD})^{-1}\mathbf{A}^{-1}\mathbf{BCDA}^{-1}\end{aligned}$$

where the identity P1:

$$(\mathbf{I}+\mathbf{P})^{-1} = (\mathbf{I}+\mathbf{P})^{-1}(\mathbf{I}+\mathbf{P}-\mathbf{P}) = \mathbf{I}-(\mathbf{I}+\mathbf{P})^{-1}\mathbf{P}$$

We can also use identity P2:

$$\mathbf{P}+\mathbf{PQP} = \mathbf{P}(\mathbf{I}+\mathbf{QP}) = (\mathbf{I}+\mathbf{PQ})\mathbf{P} \quad \Rightarrow \quad (\mathbf{I}+\mathbf{PQ})^{-1}\mathbf{P} = \mathbf{P}(\mathbf{I}+\mathbf{QP})^{-1}$$

to have another matrix inverse:

$$(\mathbf{A}+\mathbf{BCD})^{-1} = \mathbf{A}^{-1}-\mathbf{A}^{-1}\mathbf{BCDA}^{-1}(\mathbf{I}+\mathbf{BCDA}^{-1})^{-1}$$

**There are many other equivalent forms - you can derive them easily when needed**

# Notes:

○

# Notes:

○

# Notes:

○