TÉCNICO LISBOA

Principles of Biosignals and Biomedical Imaging

3rd year, P_3 (ECTS: 3.0), LEBiom
2022/2023

João Sanches
jmrs@tecnico.ulisboa.pt
Department of Bioengineering
Instituto Superior Técnico
University of Lisbon

Outline

Space Vector of Signals

Match filter

Non-linear signal generation

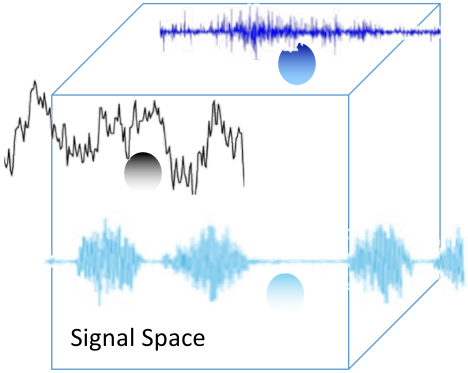
Space vector and signals

Vector Space of Signals

- A **linear space of vectors,** S, defined over a **scalar set R,** is a set of objects, called **vectors**, for which an addition operation between vectors is defined as well as a multiplication operation by scalars

- Basic operations are defined between elements of the set

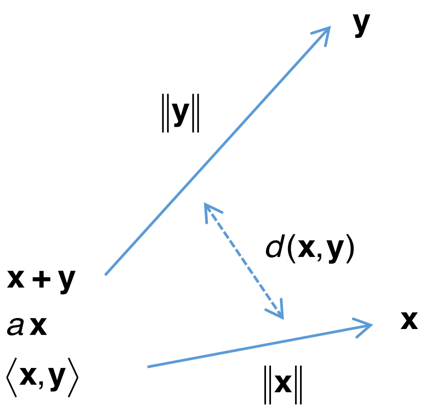
- Operation results always belong to the set (Closeness)



Vector operations

- Addition
- Multiplication by scalars

- Metric (Distance between vectors)
- Norm (Length)
- Inner-product (product/correlation)



Metric Function

- **Definition**

  A metric $d : X \times X \rightarrow R$ is a function that measures the distance between the elements in a set X. To be a metric the function d must satisfy the following properties for all $x, y \in X$

  1) $d(x,y) = d(y,x)$
  2) $d(x,y) \geq 0$
  3) $d(x,y) = 0$ if and only if $x = y$
  4) For all elements $x, y, z \in X : d(x,z) \leq d(x,y) + d(y,z)$

     Triangle Inequality

Definition:

A **metric space** is a space where a metric function is defined.

Metrics of discrete signals

1. l_1, called Manhattan distance, …….

$$d_1(x,y)=\sum_{i=1}^{n}\left|x_i-y_i\right|$$

2. l_2, called Euclidean distance, ………

$$d_2(x,y)=\sqrt{\sum_{i=1}^{n}\left|x_i-y_i\right|^2}$$

1. l_p, ………………………………………………

$$d_p(x,y)=\left(\sum_{i=1}^{n}\left|x_i-y_i\right|^p\right)^{1/p}$$

1. l_∞, ……………………………………………..

$$d_\infty(x,y)=\max_{i=1,2,..,n}\left|x_i-y_i\right|$$

1. l_0,………………………………

$$d_0(x,y)=\lim_{p\to 0}\left[d_p(x,y)\right]^p=\#\left[\left|x_i-y_i\right|>0\right]$$

Norm Function

- **Definition**
  Let S be a vector space and ||x|| be a real function where $\mathbf{x} \in S$. The operator ||.|| is called **norm** if the following properties hold

$$\|\mathbf{x}\| \geq 0 \text{ for any } \mathbf{x} \in S$$

$$\|\mathbf{x}\| = 0 \text{ if and only if } \mathbf{x} = \mathbf{0}$$

$$\|a\mathbf{x}\| = |a|\|\mathbf{x}\|, \text{where } a \text{ is a scalar}$$

$$\|\mathbf{x}+\mathbf{y}\| \leq \|\mathbf{x}\| + \|\mathbf{y}\|, \text{(triangle inequality)}$$

||x|| is called norm or length of x

Definition:

A **Banach** space is a space where a norm function is defined.

Inner Product

- **Definition**

Let S be a vector space defined over a scalar field R. An inner product is a function $\langle .,. \rangle : S \times S \rightarrow C$ with the following properties

$$1)\langle \mathbf{x},\mathbf{y} \rangle = \overline{\langle \mathbf{y},\mathbf{x} \rangle}$$

$$2)\langle a\mathbf{x},\mathbf{y} \rangle = a\langle \mathbf{x},\mathbf{y} \rangle$$

$$3)\langle \mathbf{x}+\mathbf{y},\mathbf{z} \rangle = \langle \mathbf{x},\mathbf{z} \rangle + \langle \mathbf{y},\mathbf{z} \rangle$$

$$4)\langle \mathbf{x},\mathbf{x} \rangle > 0 \text{ if } \mathbf{x} \neq \mathbf{0}, \text{and} \langle \mathbf{x},\mathbf{x} \rangle = \mathbf{0} \text{ if and only if } \mathbf{x} = \mathbf{0}$$

Definition:

A **Hilbert** space is a space where an **inner product** function is defined.

The inner product in C^n is defined as

$$\langle \mathbf{x},\mathbf{y} \rangle = \sum_{k=1}^{n} x_k \overline{y}_k = \mathbf{y}^H \mathbf{x}$$

Inner product and induced norm

- The inner product operation can be used to defined a norm function, called in this case, an induced norm.

$$\|\mathbf{x}\| = \langle \mathbf{x},\mathbf{x} \rangle^{1/2}$$

- Theorem (Cauchy-Schwartz inequality)
  In an inner product space S with induced norm

$$\left|\langle \mathbf{x},\mathbf{y} \rangle\right| \le \|\mathbf{x}\|.\|\mathbf{y}\|$$

  for any x,y $\in S$ with equality if and only if for some scalar a, y=ax

The inner product in C^n is defined as

$$\langle \mathbf{x},\mathbf{y} \rangle = \sum_{k=1}^{n} x_k \bar{y}_k = \mathbf{y}^H \mathbf{x}$$

- The vectors x and y from a vector space with inner product are said orthogonal if <x,y>=0
- The **null** vector is orthogonal to all other vectors

First and second order statistics

Let x(n) and y(n) be N length discrete signals:

Mean $$\mu = \langle \boldsymbol{x} \rangle = \frac{1}{N}\sum_{n=0}^{N-1} x(n)$$

Variance $$\sigma^2 = \frac{1}{N-1}\sum_{n=0}^{N-1} (x(n) - \mu)^2$$

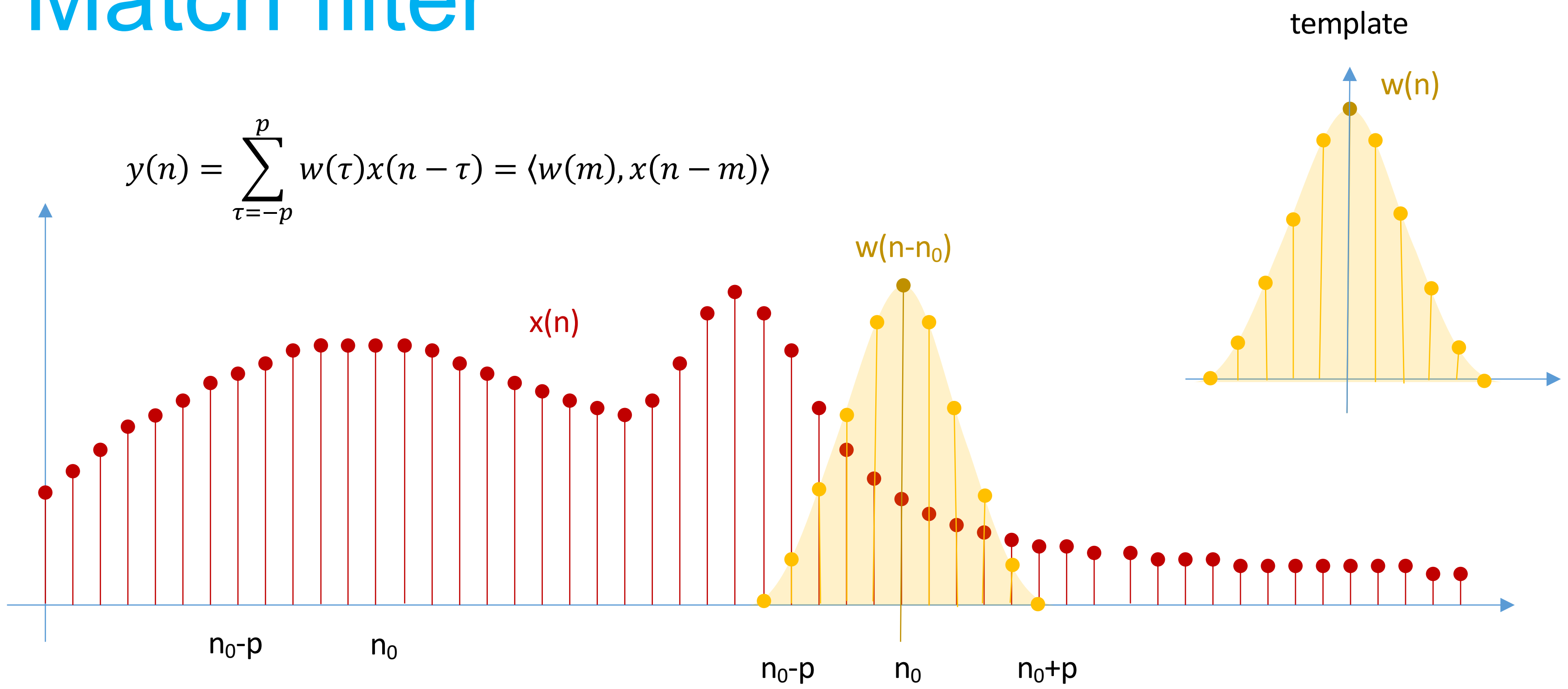
Autocorrelation $$\phi_{xx}(m) = \frac{1}{N}\sum_{n} x(n)x(n+m)$$

Cross-correlation $$\phi_{xy}(m) = \frac{1}{N}\sum_{n} x(n)y(n+m)$$

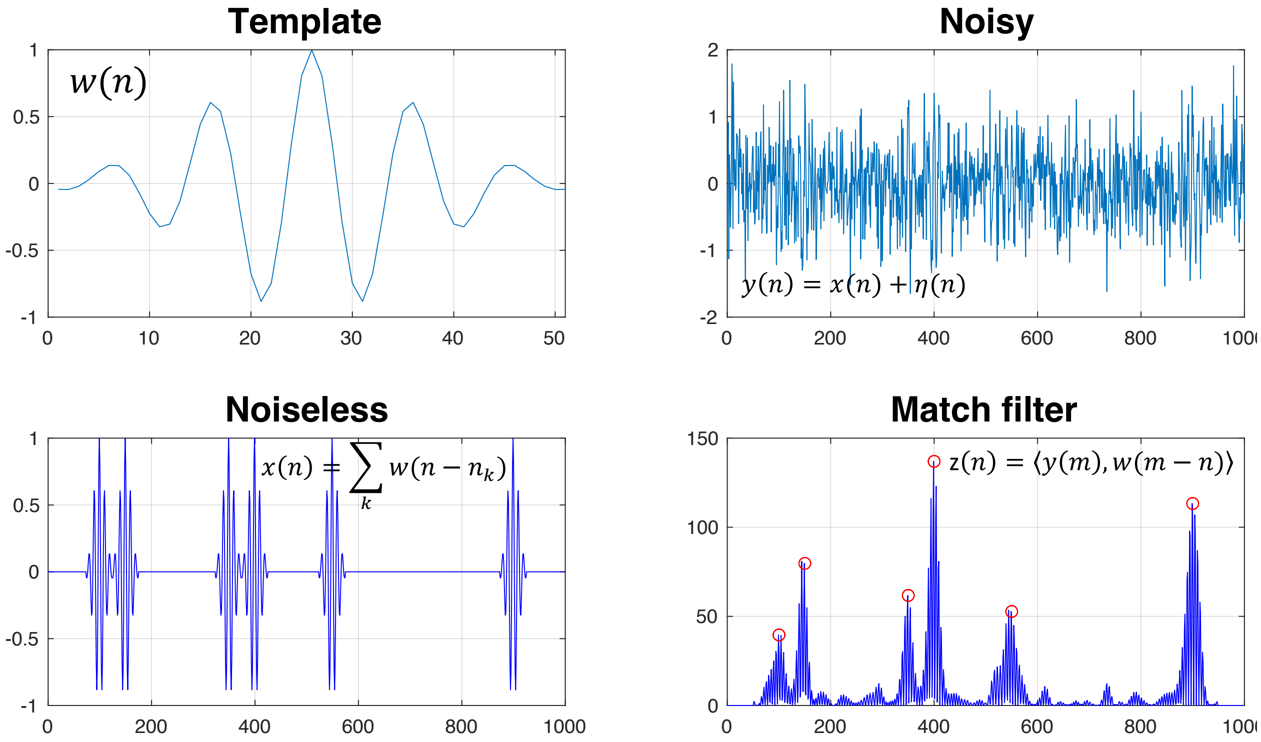
Match filter

Match filter

$$y(n) = \sum_{\tau=-p}^{p} w(\tau)x(n-\tau) = \langle w(m), x(n-m) \rangle$$



Match filter




```
%Template creation
T=50;
w=2*pi/10;
a=0.005;
n=(-T/2:T/2)';
h=cos(w*n).*exp(-a*n.^2);

% Building a set of noiseless and noisy replicas
N=1000;
x=zeros(N,1);
n=0:length(x)-1;

%Replica locations
n0=[2*T, 3*T, 7*T, 8*T, 11*T, 18*T];

%noiselss – adding template replicas
for l=1:length(n0)
   x(n0(l)-T/2: n0(l)+T/2)=h;
end

%add noise
eta=0.5;
y=x + eta*randn(size(x));

%Mach filter
z=zeros(size(x));
for l=T+1:N-T-1
   z(l)=(h'*y(l-T/2:l+T/2)).^2;
end

figure(1);
subplot(2,2,1); plot(h); title('Template'; grid; xlim([0 T+1]);
subplot(2,2,2); plot(n,x,'b'); title('Noiseless'); grid;
subplot(2,2,3); plot(y); title('Noisy'); grid;
subplot(2,2,4); plot(n, z, 'b', n0,z(n0)*ones(size(n0)),'or');
title('Match filter', 'fontsize',18); grid;
```

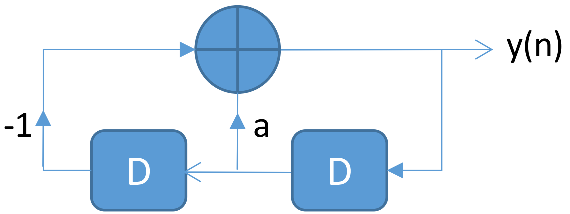
TÉCNICO LISBOA

Break

Non linear signal generation
Harmonic generation

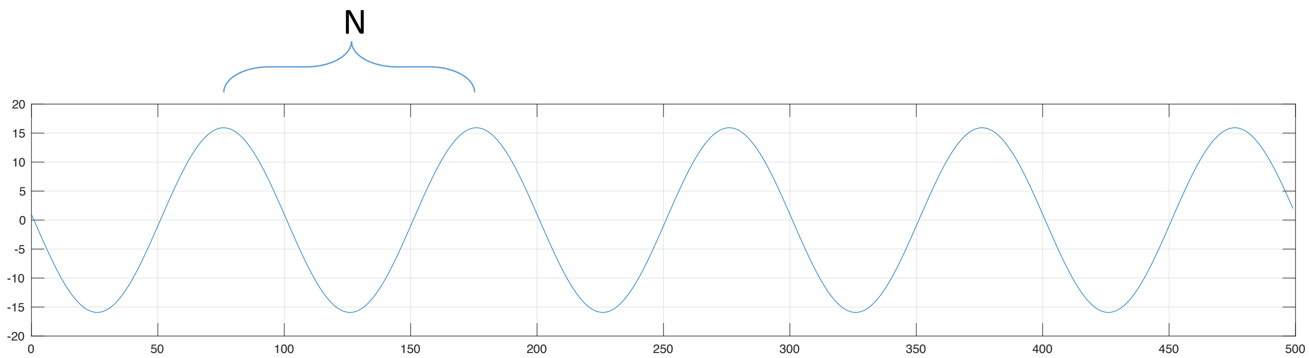
Discrete Oscillator

$$y(n) = ay(n-1) - y(n-2)$$



$$a = 2\cos(\omega)$$

$$\omega = \frac{2\pi}{N}$$




```
%Digital oscillator
N=1000;
n=(0:N-1)';
y=zeros(size(n)); y(1)=1;

w=pi/20;
for l=3:N
   y(l)=2*cos(w)*y(l-1)-y(l-2);
end

figure(1);
plot(n,y); grid;
```

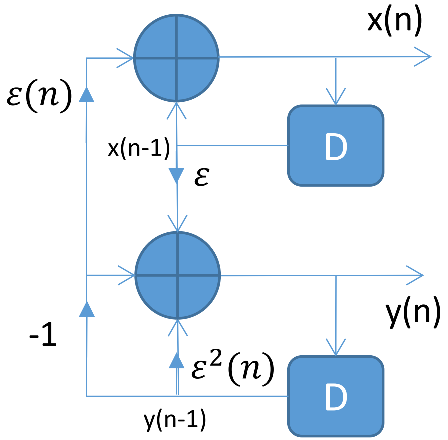
DCO

Digital Controlled Oscillator

$$x(n) = x(n-1) - \varepsilon(n)y(n-1)$$

$$y(n) = \varepsilon(n)x(n-1) - (1-\varepsilon^2(n))y(n-1)$$

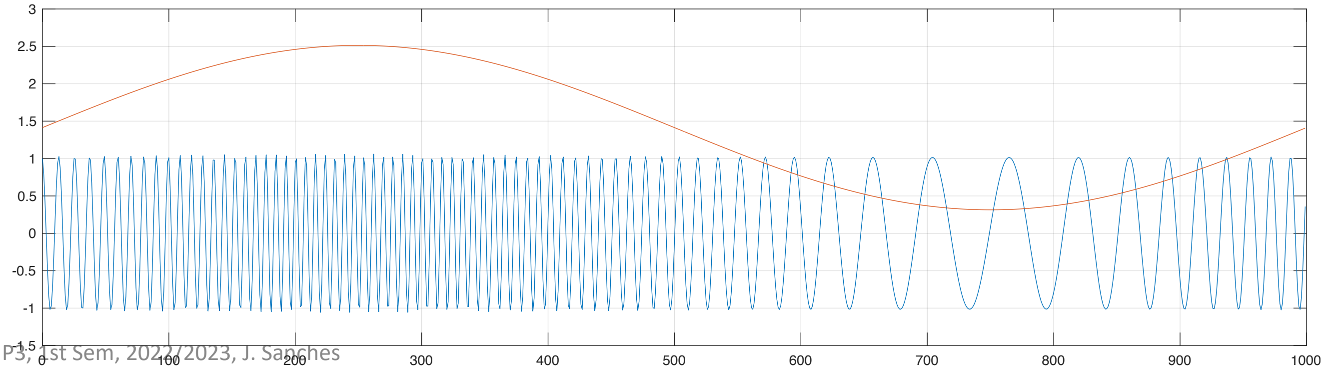
$$\varepsilon(n) = \omega(n)$$



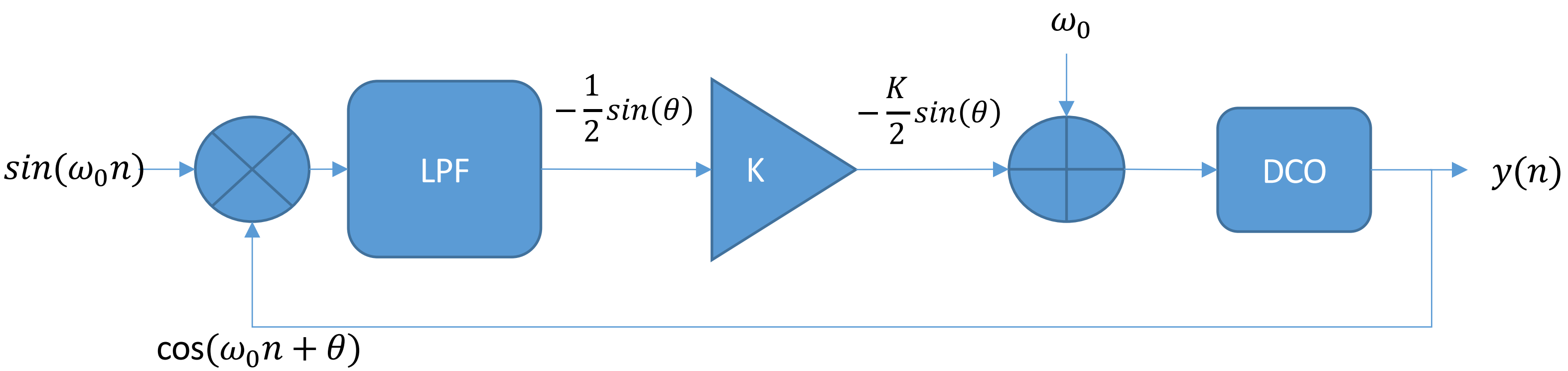

```
%% VCO – Voltage Controlled Oscillator
N=1000;
n=(0:N-1)';
x=zeros(size(n));
y=zeros(size(n)); y(1)=1;
w0=ones(size(n));        %Varying frequency in rad/sample

w0min=0.1*pi;            %Maximum frequency in rad/sample
w0max=0.8*pi;            %Minimum frequency in rad/sample
w0=w0min + (w0max-w0min)*(1+sin(2*pi*n/N))/2;

for l=2:N
   e=w0(l)/pi;
   x(l)=x(l-1)-e*y(l-1);
   y(l)=e*x(l-1)+(1-e^2)*y(l-1);
end
```



Digital Phase Lock Loop (PLL)



$$\sin(\mathrm{a})\cos(\mathrm{b}) = \frac{1}{2}(\sin(a+b) + \sin(a-b)) \Longrightarrow \sin(\omega_0 n)\, cos(\omega_0 n + \theta) = \frac{1}{2}[\sin(2\omega_0 n + \theta) - sin(\theta)]$$

Digital Phase Lock Loop (PLL)

```
%PLL filter

N=1000;
n=(0:N-1)';

%Initializations
x =zeros(size(n));        % Original
x0=zeros(size(n));
z =zeros(size(n));        % Noisy
p =zeros(size(n));        % Product block
f =zeros(size(n));        % Filtered
fi=zeros(size(n));        % Integrator
w =zeros(size(n));        % Varying frequency

vcox=zeros(size(n));
vcoy=zeros(size(n)); vcoy(1:2)=1;

%Input data generator
w0  = 2*pi/20;            % Central frequency
sig = 0.25;               % Noise power
a   = 0.9;                % Low-pass filter parameters
b   = 0.5;                % Integrator gain
e   = w0;                 % VCO Initialization
g   = 0.25;               % Input VCO gain
```

```
% Varying frequency of the input signal
u = ones(size(n)); u(N/2:N-1)=-1;
%w = w0 + (pi/100)*sin(pi*n/N);
w  = w0 + (pi/25)*u;

x =sin(w.*n);                 % Original
x0=cos(w.*n);                 % Quadrature
z =x+sig*randn(size(n));      % Noisy
```

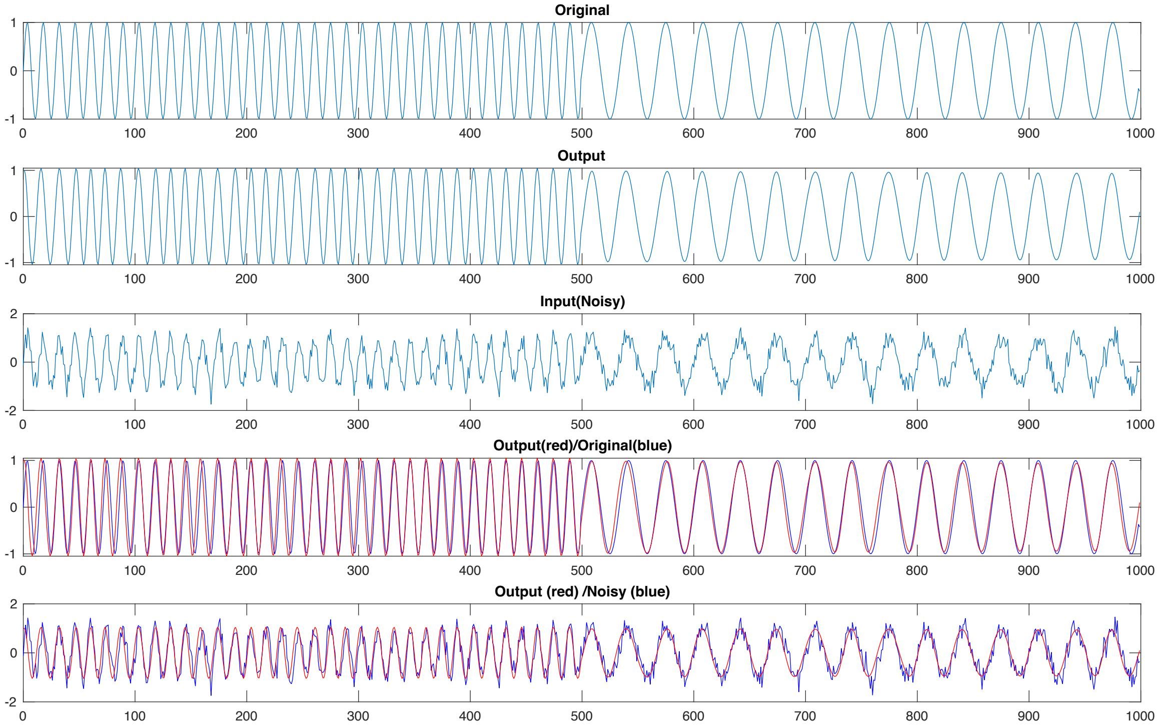
```
%Display the results
figure(1);
subplot(5,1,1); plot(n,x);              title('Original');
subplot(5,1,2); plot(n,vcoy);           title('Output');
subplot(5,1,3); plot(n,z);              title('Input(Noisy)');

subplot(5,1,4); plot(n,x,'b',n,vcoy,'r');   title('Output(red)/Original(blue)');
subplot(5,1,5); plot(n,z,'b',n,vcoy,'r');   title('Output (red) /Noisy (blue)');
```

```
%PLL implementaion
for l=3:N
   %VCO
      vcox(l)=vcox(l-1)-e*vcoy(l-1);
      vcoy(l)=e*vcox(l-1)+(1-e^2)*vcoy(l-1);

   %Phase Detector
      %Product: z*vco_x
      p(l)=vcox(l)*z(l);
      % Filtering
      f(l)=(1-a)^2*p(l)+2*a*f(l-1)-a^2*f(l-2);
      %Integrator
      fi(l)=(1-b)*f(l)+b*fi(l-1);
   % VCO input
      e=w(l)+g*fi(l);
end
```

Digital Phase Lock Loop (PLL)



TÉCNICO LISBOA