TOPIC between 5 and 6
Digital filters
PART 1 (and no more)

Linear difference equations with constant coefficients

$$\sum_{i=0}^{L} b_i y[n-i] = \sum_{r=0}^{R} c_r x[n-r]$$

$n \geq 0$

$n = 0, 1, 2, 3...$

With L-INITIAL CONDITIONS (they are required)

$$y[-1], y[-2], ..., y[-L]$$

We need to know these L values !

Example

$$y[n] - 0.2y[n-1] - 0.2y[n-2] = x[n] + x[n-1]$$

$$L = 2 \qquad R = 1$$

$$b_0 = 1, b_1 = -0.2, b_2 = -0.2 \qquad c_0 = 1, x_1 = 1$$

Additionally consider the L- INITIAL CONDITIONS:

$$y[-1] = 8, y[-2] = -5$$

and the following input signal:

$$x[-1] = 6, x[0] = -2, x[1] = -3,$$

$$\text{the rest of values are zero for } n \neq -1, 0, 1$$

Example

Obtain the first 3 values of the output signal y[n], i.e., y[0], y[1] and y[2]:

$$y[n] - 0.2y[n-1] - 0.2y[n-2] = x[n] + x[n-1]$$

$$y[-1] = 8, y[-2] = -5$$

$$x[-1] = 6, x[0] = -2, x[1] = -3,$$

$$\text{the rest of values are zero for } n \neq -1, 0, 1$$

Example

$$y[n] = 0.2y[n-1] + 0.2y[n-2] + x[n] + x[n-1]$$

$$n = 0$$

$$y[0] = 0.2y[0-1] + 0.2y[0-2] + x[0] + x[0-1]$$

$$y[0] = 0.2 \cdot 8 + 0.2 \cdot (-5) - 2 + 6$$

$$y[0] = 4.6$$

$$n = 1$$

$$y[1] = 0.2y[1-1] + 0.2y[1-2] + x[1] + x[1-1]$$

$$y[1] = 0.2y[0] + 0.2y[-1] + x[1] + x[0]$$

$$y[1] = 0.2 \cdot 4.6 + 0.2 \cdot 8 - 3 - 2$$

$$y[1] = -2.48$$

Example

$$y[0] = 4.6 \quad y[1] = -2.48$$

$$n = 2$$

$$y[2] = 0.2y[2-1] + 0.2y[2-2] + x[2] + x[2-1]$$

$$y[2] = 0.2y[1] + 0.2y[0] + x[2] + x[1]$$

$$y[2] = 0.2 \cdot (-2.48) + 0.2 \cdot 4.6 + 0 - 3$$

$$y[2] = -2.576$$

Example

Solution:

$$y[0] = 4.6 \quad y[1] = -2.48 \quad y[2] = -2.576$$

Linear difference equations with constant coefficients

Different names:

- ✓ Linear difference equations with constant coefficients
- ✓ ARMA filters (special cases: AR filters - MA filters)
- ✓ IIR filters (FIR filters)
- ✓ Digital filters

SOLUTION: y[n]

We have solved the difference equation for 3 times steps by applying the recursion 3 times….

For linear difference equations with constant coefficients general analytical solutions can be obtained (as for the linear differential equations with constant coefficient).

GENERAL SOLUTION: y[n]

$$y[n] = y_o[n] + y_f[n]$$

Solution of the homogeneous system:
With x[n]=0 and generic initial conditions (free-response; "transitory", if stable…)

"Forced" solution:
Generic x[n] but null initial conditions.

Regarding y_o[n]....

$$y[n] = y_o[n] + y_f[n]$$

Solution of the homogeneous system: with x[n]=0 and generic initial conditions (free-response; "transitory", if stable…)

✓ We have to study the ``characteristic'' polynomial: related to the denominator of H(z) and the poles are roots of the ``characteristic'' polynomial…

✓ There are courses only for this goal: Dynamical systems etc… (as with the differential equations)

Regarding y_o[n]....

$$y[n] = y_o[n] + y_f[n]$$

✓ **IF THE INITIAL CONDITIONS ARE NULL, THEN:**

$$y_o[n] = 0$$

Regarding y_o[n]....

✓ **IF THE INITIAL CONDITIONS ARE NULL, THEN:**

$$y_o[n] = 0$$

And so:

$$y[n] = y_f[n]$$

Regarding y_f[n]....

$$y_f[n] = x[n] * h[n]$$

$$Y_f(z) = X(z)H(z)$$

In all this course, we have always considered null initial conditions:

$$y[n] = y_f[n] = x[n] * h[n]$$

$$Y(z) = Y_f(z) = X(z)H(z)$$

How can we find the corresponding h[n]?

$$y[-1] = ... = y[-L] = 0$$

$$x[n] = \delta[n]$$

Recall that h[n] is the "impulse response":
It can be done "by hand" following the recursion…

There is also a general procedure to obtain the anylitical form of h[n]
(we do not considered it now…)

Classification

$$y[-1] = \ldots = y[-L] = 0$$

$$\sum_{i=0}^{L} b_i y[n-i] = \sum_{r=0}^{R} c_r x[n-r]$$

Auto-regressive part (AR)

Moving-average part (MA)

AR+MA = ARMA filter

General consideration

- ✓ A filter with an AR part is "more powerful" in general (than an MA filter, without an AR part…)

- ✓ BUT it can be STABILITY ISSUES !
WE HAVE TO BE CAREFUL !

Moving average (MA) filter

$$y[n] = \sum_{r=0}^{R} c_r x[n-r]$$

Examples:

$$y[n] = x[n] - x[n-1] + 10x[n-10]$$

$$y[n] = \frac{1}{3}x[n] + \frac{1}{3}x[n-1] + \frac{1}{3}x[n-3]$$

Arithmetic mean of the last 3 samples of x[n]

MA filter = FIR filter

✓ **MA filters are also called-known as FINITE IMPULSE RESPONSE (FIR) filters.**

WHY? Because the corresponding h[n] has finite-length !

MA filter = “all-zeros” filter

- ✓ **An MA filter is also called as “all-zeros” filter**

- ✓ **WHY? The H(z) of h[n] correspoding to an MA filter has “only” zeros, except one pole (generally multiple) at |z|=r=0.**

MA filter = “all-zeros” filter

$$y[n] = \sum_{r=0}^{R} c_r x[n-r] \quad \Rightarrow \quad Y(z) = \sum_{r=0}^{R} c_r z^{-r} X(z)$$

$$H(z) = \frac{Y(z)}{X(z)} = \sum_{r=0}^{R} c_r z^{-r}$$

$$H(z) = \frac{\sum_{r=0}^{R} c_r z^{R-r}}{z^R}$$

MA filter = FIR filter

$$H(z) = \frac{Y(z)}{X(z)} = \sum_{r=0}^{R} c_r z^{-r}$$

$$h[n] = \sum_{r=0}^{R} c_r \delta[n-r]$$

✓ **FINITE IMPULSE RESPONSE (FIR) filter**

MA filter:
advantanges and drawbacks

- ✓ **Good point:** Easy to implement

- ✓ **Very good point:** **WE HAVE NOT STABILITY ISSUES !!!**
If x[n] is stable (i.e., bound) no problem.

- ✓ **Drawback:** **to reach the performance of an AR filter (i.e., to have the same features in frequency) we need a huge value of R (several coefficients)…. We will see that an AR filter can be expressed as an MA filter with R=infinity.**

Auto-regressive (AR) filter

$$\sum_{i=0}^{L} b_i y[n-i] = x[n]$$

$$y[n] = -\sum_{i=1}^{L} \frac{b_i}{b_0} y[n-i] + \frac{1}{b_0} x[n]$$

Auto-regressive (AR) filter

Examples:

$$y[n] = y[n-1] + x[n]$$

$$y[n] = 0.5y[n-1] - 0.2y[n-2] + x[n]$$

$$y[n] = 0.8y[n-3] - 5y[n-10] + x[n]$$

AR filter = IIR filter

✓ **AR filters are also called-known as INFINITE IMPULSE RESPONSE (IIR) filters.**

WHY? Because the corresponding h[n] has Infinite-length !

AR filter = “all-poles” filter

- ✓ An AR filter is also called as “all-poles” filter

- ✓WHY? The H(z) of h[n] corresponding to an AR filter has “only” poles, except one zero (generally multiple) at |z|=r=0.

AR filter = "all-poles" filter

$$y[n] = -\sum_{i=1}^{L} \frac{b_i}{b_0} y[n-i] + \frac{1}{b_0} x[n]$$

$$Y(z) = -\sum_{i=1}^{L} \frac{b_i}{b_0} z^{-i} Y(z) + \frac{1}{b_0} X(z)$$

$$Y(z)\left(b_0 + \sum_{i=1}^{L} b_i z^{-i}\right) = X(z)$$

$$H(z) = \frac{Y(z)}{X(z)} = \frac{1}{b_0 + \sum_{i=1}^{L} b_i z^{-i}} = \frac{1}{\sum_{i=0}^{L} b_i z^{-i}}$$

AR filter = "all-poles" filter

$$H(z) = \frac{Y(z)}{X(z)} = \frac{1}{b_0 + \sum_{i=1}^{L} b_i z^{-i}} = \frac{1}{\sum_{i=0}^{L} b_i z^{-i}}$$

$$H(z) = \frac{Y(z)}{X(z)} = \frac{z^L}{\sum_{i=0}^{L} b_i z^{L-i}}$$

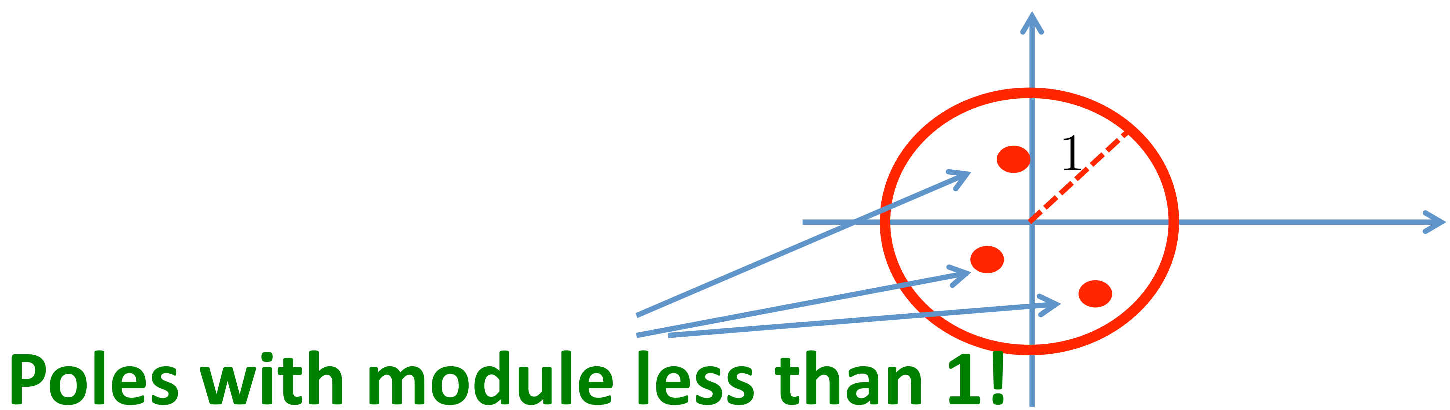
✓**The H(z) has "only" poles, except one zero (generally multiple) at |z|=r=0.**

AR filter: advantanges and drawbacks

- ✓ Good point: we can obtain very good performance.
- ✓We will see that an AR filter can be expressed as an MA filter with R=infinity.
- ✓ Drawback: WE HAVE STABILITY ISSUES !!!
- ✓ Drawback: the implementation is not "easy" also for the problem of the stability.

AR filter: STABILITY

✓ **ALL POLES WITH MODULE LESS THAN 1, i.e., ALL POLES WITHIN THE CIRCLE WITH RADIUS 1.**



AR+MA = ARMA filter

= linear difference equation with constant coefficients and null initial conditions

= LTI system

AR+MA = ARMA filter

$$y[-1] = \ldots = y[-L] = 0$$

$$\sum_{i=0}^{L} b_i y[n-i] = \sum_{r=0}^{R} c_r x[n-r]$$

Auto-regressive part (AR)

Moving-average part (MA)

AR+MA = ARMA filter

ARMA filter - Zeta Transform

$$\mathcal{Z}\left\{\sum_{i=0}^{L} b_i y[n-i]\right\} = \mathcal{Z}\left\{\sum_{r=0}^{R} c_r x[n-r]\right\}$$

$$\sum_{i=0}^{L} b_i \mathcal{Z}\left\{y[n-i]\right\} = \sum_{r=0}^{R} c_r \mathcal{Z}\left\{x[n-r]\right\}$$

$$\sum_{i=0}^{L} b_i z^{-i} Y(z) = \sum_{r=0}^{R} c_r z^{-r} X(z)$$

ARMA filter -Zeta transform

$$\sum_{i=0}^{L} b_i z^{-i} Y(z) = \sum_{r=0}^{R} c_r z^{-r} X(z)$$

$$H(z) = \frac{Y(z)}{X(z)} = \frac{\sum_{r=0}^{R} c_r z^{-r}}{\sum_{i=0}^{L} b_i z^{-i}}$$

ARMA filter - Zeta Transform of the impulse response of a LTI system

$$H(z) = \frac{Y(z)}{X(z)} = \frac{\sum_{r=0}^{R} c_r z^{-r}}{\sum_{i=0}^{L} b_i z^{-i}}$$

$$H(z) = \frac{Y(z)}{X(z)} = z^{L-R} \frac{\sum_{r=0}^{R} c_r z^{R-r}}{\sum_{i=0}^{L} b_i z^{L-i}}$$

Clearly

ARMA filter = IIR filter,

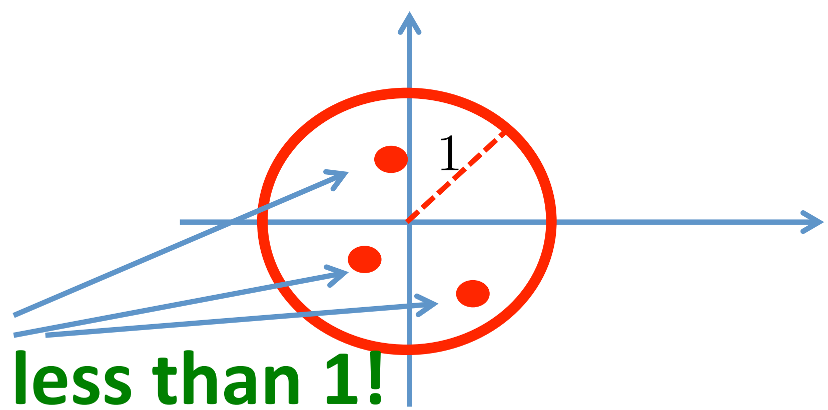
Since it has an AR part.

ARMA filter

- ✓ **MORE flexibility than AR and MA filters.**
- ✓ **ARMA is an IIR filter due to its AR part.**

ARMA filter: STABILITY

✓ **ALL POLES WITH MODULE LESS THAN 1, i.e., ALL POLES WITHIN THE CIRCLE WITH RADIUS 1.**



Poles with module less than 1!

Expressing an AR filter as a MA filter of "order Infinity"

We are going to show it, starting with the simplest AR filter: AR(1), of order 1

$$AR(1) \quad y[n] = \alpha y[n-1] + x[n]$$

$$y[n-1] = \alpha y[n-2] + x[n-1]$$

$$y[n] = \alpha(\alpha y[n-2] + x[n-1]) + x[n]$$

$$y[n] = \alpha^2 y[n-2] + \alpha x[n-1] + x[n]$$

$$y[n] = \alpha^m y[n-m] + \sum_{k=0}^{m-1} \alpha^k x[n-k],$$

Si: $|\alpha| < 1$ y $m \to \infty$

In that case the AR(1) Is stable !

$$y[n] = \sum_{k=0}^{\infty} \alpha^k x[n-k], \qquad MA(\infty)$$

MA filtering
(an example)

Input signal x[n] (to be filtered)

```
T=0.1;
%%%ws=2*pi/T;
W=2;
tc=0:0.0001:10; %%% simulating the continous domain
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
xc=sin(W*tc);
%xs=sin(W*ts)+j*cos(W*ts);
ts=0:T:10;
sig=0.5;
f=sin(W*ts);
noise=sig*randn(1,length(ts));
xs=f+noise;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

$$x[n] = \sin(WnT) + \text{Gaussian noise}$$

ts

MA filter of order 9 – MA(9)

$$y[n] = \frac{1}{10}\sum_{r=0}^{9} x[n-r]$$

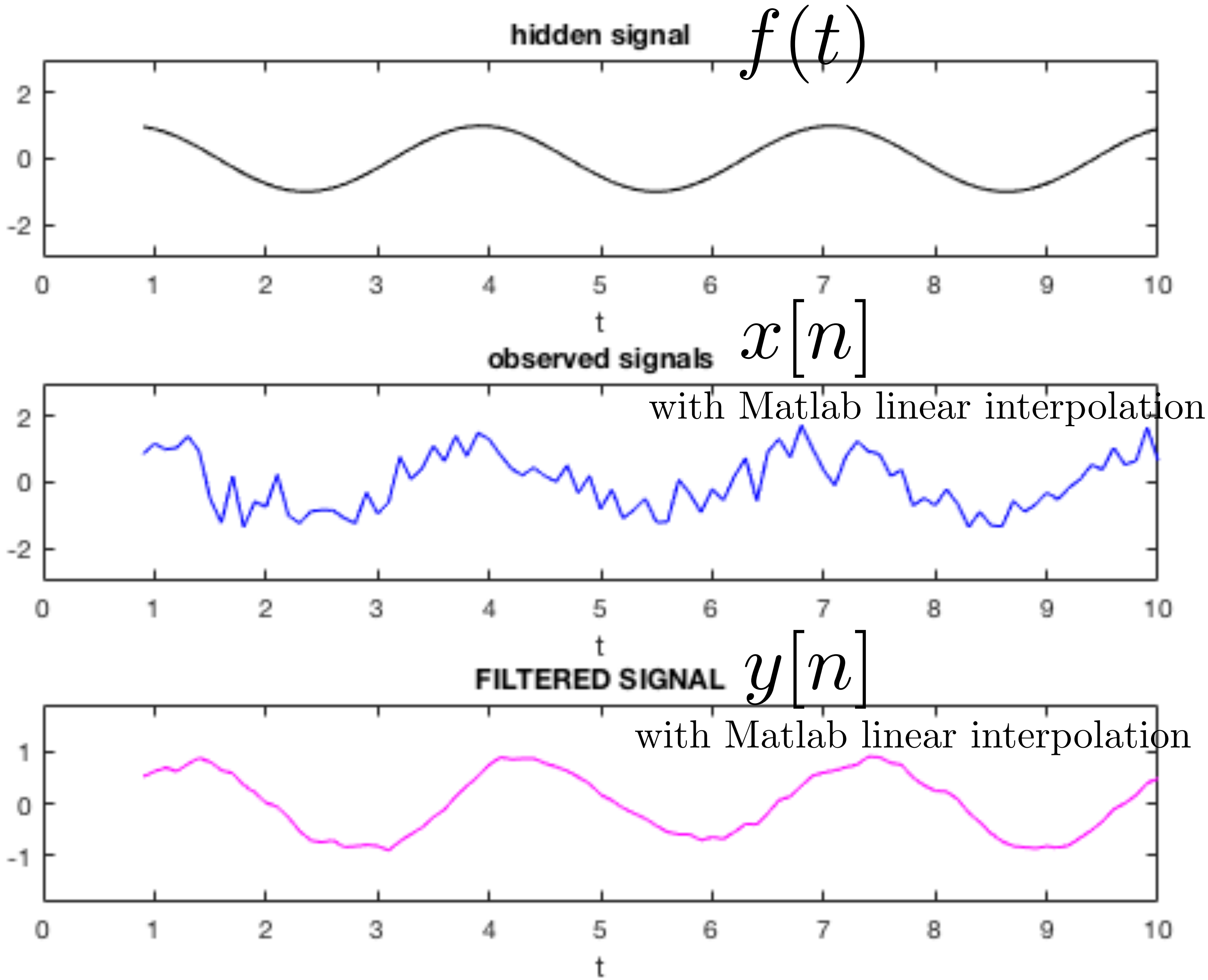
ARITHMETIC MEAN OF 10 CONSECUTIVE VALUES:

Which kind of filter can be?

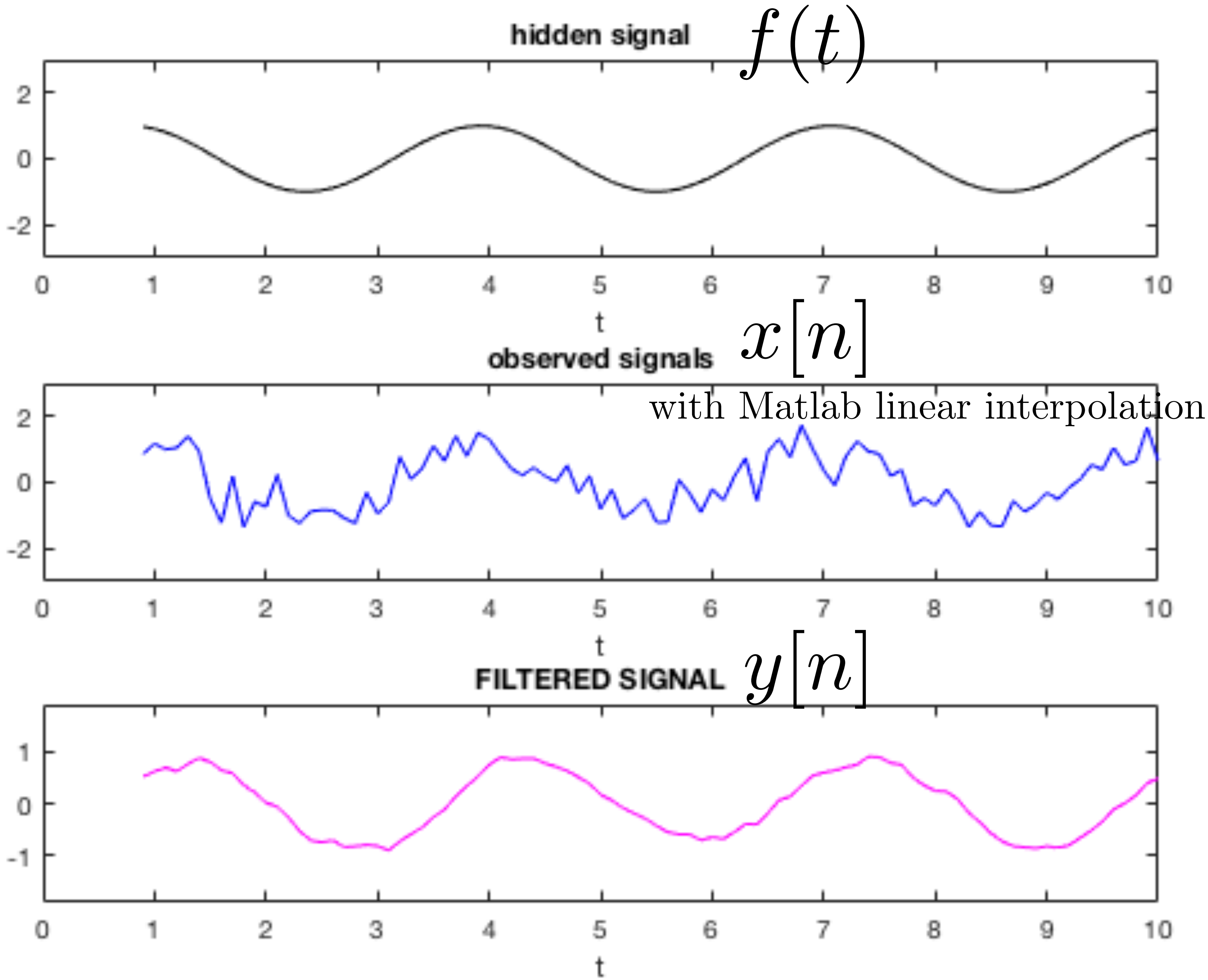
…it is doing a smoothing… making the average…

MA filter of order 9 – MA(9) impulse response

$$h[n] = \frac{1}{10} \sum_{r=0}^{9} \delta[n - r]$$



basically, we are doing a first-order interpolation/reconstruction as in Topic 3...



We are removing "high frequencies" keeping the local trends...

Indeed, this MA(9) is a low-pass filter, as expected !!!