Adaptive and Array Signal Processing
Lecture Notes WiSe2020

Guillermo Ventura

November 11, 2020

Contents

1 Lecture 2. Nov 2020

1.1 Introduction:

Complex numbers -> [TODO] all forms +cycle

[copy Equations]
all solutions of j-something to the power of j is R (real). It's a 90deg turn. On whichever direction.

The real question is...

–> take the derivative of a f(x) where a the variable is C (complex).

1.2 Notion of Analytics Functions:

-> A name that starts with H...
–> these we can analyse, d->0 or d->oo

There exist both real analytic functions and complex analytic functions, categories that are similar in some ways, but different in others. Functions of each type are infinitely differentiable, but complex analytic functions exhibit properties that do not hold generally for real analytic functions. A function is analytic if and only if its Taylor series about x0 converges to the function in some neighborhood for every x0 in its domain.
there are other functions which are not analytic.

1.3 We will learn...

1.3.1 to optimize functions:

-> constraints
-> inconstraints

–> linear constraints are easy and have real life applications.

1.3.2 We will also cover Linear Algebra:

- Pseudoinverse usage
- Least Squares usage

After the Math -> Sig Processing - adaptions to real life problems

–> Procesing Signals finds a relationship between Space and Time which can be understood and used. Electromagnetism allows this but things like a Wall can produce Delay in an air signal.

1.4 What is actually a Filter?

–> Brought term
at least 1 input at least 1 output
Dig. / Analog
Discrete / Const.
Fixed / Adaptive

1.5 Analog Filters:

The following circuit is an Analog and fixed Filter (linear). This is actually the first filter on a Smartphone to get the right band frequency.

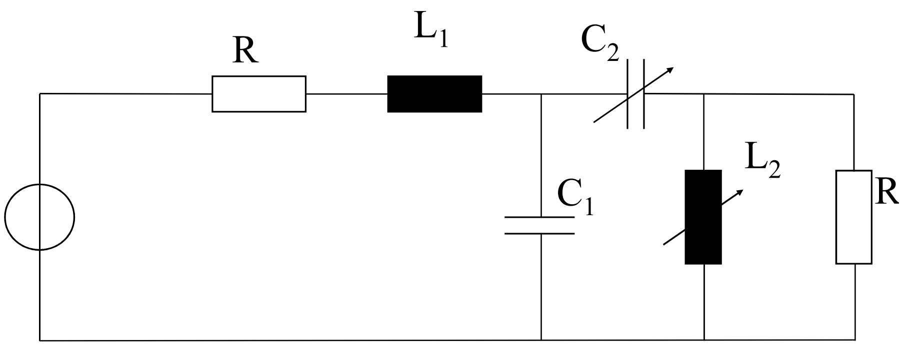


Figure 1: Linear Filter, fixed parameters

C_2 in graphic 1 can be tuned → programmable filter (due to changeable parameters). Note that this is not an Adaptive Filter yet, just a programmable one.
E.g. Bandpass with changeable frequency but same bandwidth (you have to change all: C, R and L!). Changing all parameters would make this filter not fixed.
Continuous –> Digital –> easier

1.6 Digital Filter:

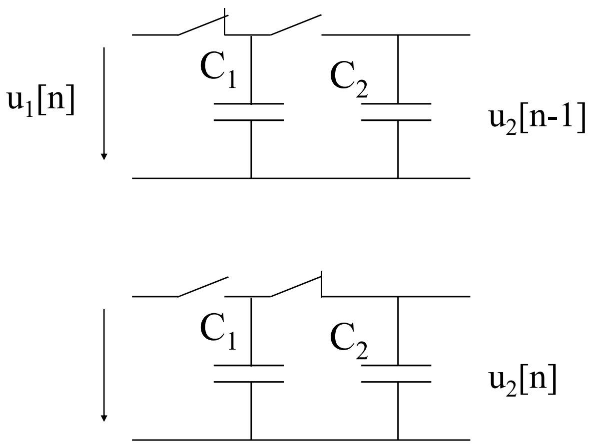


Figure 2: One open, one closed, discrete time filter

Step 1:
Trans. 1 opened
Trans. 2 closed

Step 2:
Trans. 1 closed
Trans. 2 opened

The steps correspond to the discrete clock cycles.

1.6.1 IIR-Filter

Infinite impulse response (IIR) is a property applying to many linear time-invariant systems. Common examples of linear time-invariant systems are most electronic and digital filters. Systems with this property are known as IIR systems or IIR filters, and are distinguished by having an impulse response which does not become exactly zero past a certain point, but continues indefinitely. This is in contrast to a finite impulse response (FIR) in which the impulse response h(t) does become exactly zero at times $t > T$ for some finite T, thus being of finite duration. The discrete filter in picture 2 is defined by:

$$C_1 \cdot u_1[n] + C_2 \cdot u_2[n-1] = (C_1 + C_2)u_2[n]$$

$$u_2[n] = \underbrace{\frac{C_2}{C_1 + C_2}}_{a} \cdot u_2[n-1] + \underbrace{\frac{C_1}{C_1 + C_2}}_{1-a} \cdot u_1[n]$$

It's impulse response is given by table 1. As can be seen coefficient a allows us to change the filter's impulse response and therefore to program the filter. Figure 3 shows a block diagram for such an easy, programmable discrete filter. From this block diagram it can be seen immediately that this filter is an IIR-Filter and can therefore be unstable.

Table 1: discrete filter - Impulse Response

n	≤ 0	0	1	2	...	k
$u_1[n]$	0	1	0	0	...	0
$u_2[n]$	0	$1-a$	$a(1-a)$	$a^2(1-a)$	...	$a^k(1-a)$

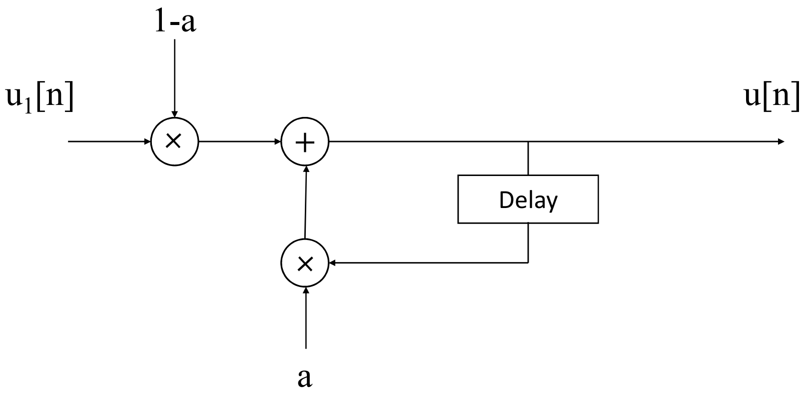


Figure 3: Discrete, easy to software implement filter

The beauty of going from Analog to Digital is that it opens the option to input anything as a, even Complex numbers/variables.
$\rightarrow$ Stability Issue:
$y[n] = \sum_{m=-\inf}^{+\inf} h[m]x[n-m]$
digital filters may be

- bounded $(-\infty, +\infty)$
- unstable

$\rightarrow$ we will introduce that $x[n] \leq A < \infty$
This is the condition for stability.
which means...

$|y[n]| \leq A \sum_{m=-\infty}^{+\infty} |h[m]| < \infty$
In case we have a *Bounded In Bounded Out (**BIBO**)* Filter: $x[m] = h^*[n-m]/|h[n-m]|$
$\rightarrow$ for BIBO (stable) $|y[n]| = \sum_{m=-\inf}^{+\inf} |h[m]| < \infty$
We can manipulate a in 3. From Table 1 we get:

$$h[n] = \begin{cases} (1-a)^n & \text{for x} \leq 0 \\ 0 & \text{else} \end{cases} \tag{1}$$

Stability Requirement:

$$\boxed{|a| \leq 1} \tag{2}$$

Must be in the unit circle!
IIR are not used for adaptive Filters

1.6.2 FIR-Filter

To overcome the stability problems an FIR-filter can be used instead. FIR-Filters are always stable. Suitable for adaptive filters but it's an approximation $\Rightarrow$ finite. Math and informatics can get good aproximations for $\to \infty$.
Structure: tapped delay line
linear: neither delay nor coefficient are dependent on input signal. $\to$ linear adaptive filter, see picture 4.

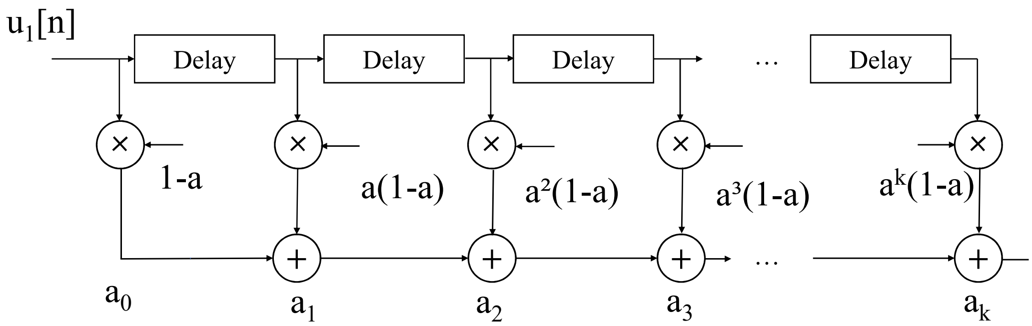


Figure 4: Linear adaptive filter (tapped delay line)

Delays of 1 clock cycle:
$D \cdot x[n] = x[n-1]$

Delay of 2 clock cycles:
$\underbrace{DD}_{D^2} \cdot x[n] = D \cdot x[n-1] = x[n-2] = D^2 \cdot x[n]$

Fractional delay:
$F \cdot x(n) = x(n - \frac{1}{2})$
$FF \cdot x(n) = F \cdot x(n - \frac{1}{2}) = x[n-1] = D \cdot x[n]$
$F^2 = FF \equiv D; F = \sqrt{D} = D^{\frac{1}{2}}$
$D^{\frac{1}{2}} \cdot D^{\frac{1}{2}} = D^{\frac{1}{2}+\frac{1}{2}} = D' = D$
Developed using a Taylor series: $D^{\frac{1}{2}} \approx 1 + \frac{1}{2}(D-1) + \frac{1}{8}(D-1)^2 = \frac{3}{8} + \frac{3}{4} \cdot D + \frac{1}{8}D^2$

Second order Taylor expansion $\to$ Second degree Filter
The filter in graphic 5 shows the shifting of the signal by $\frac{1}{2}$.

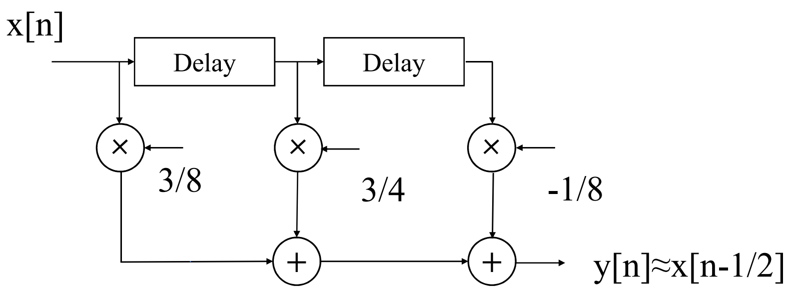


Figure 5: Used for synchronization of receiver and transmitter

A fractional delay $y[n] \approx u[n-a]$ for $0 \leq a \leq 1$
$y[n] = a_0 \cdot u[n] + a_1 \cdot u[n-1] + a_2 \cdot u[n-2]$

- $a_0 = \frac{1}{2}(a^2 - 3a + 2)$
- $a_1 = (2-a)a$
- $a_2 = \frac{1}{2}(a-1)a$

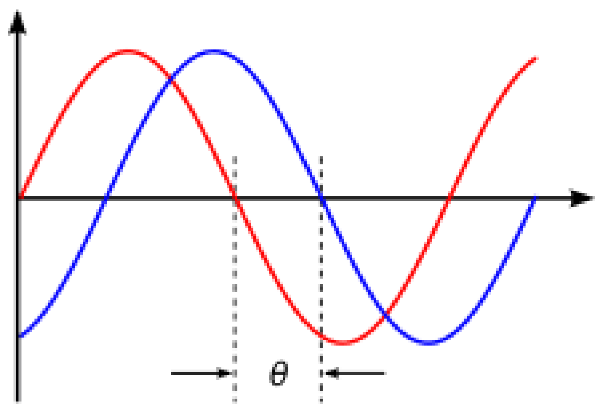


Figure 6: Phase-shift for θ

Consider a discrete-time signal $s[n]$ which propagates through a multi-path channel and arrives at the receiving end as the signal

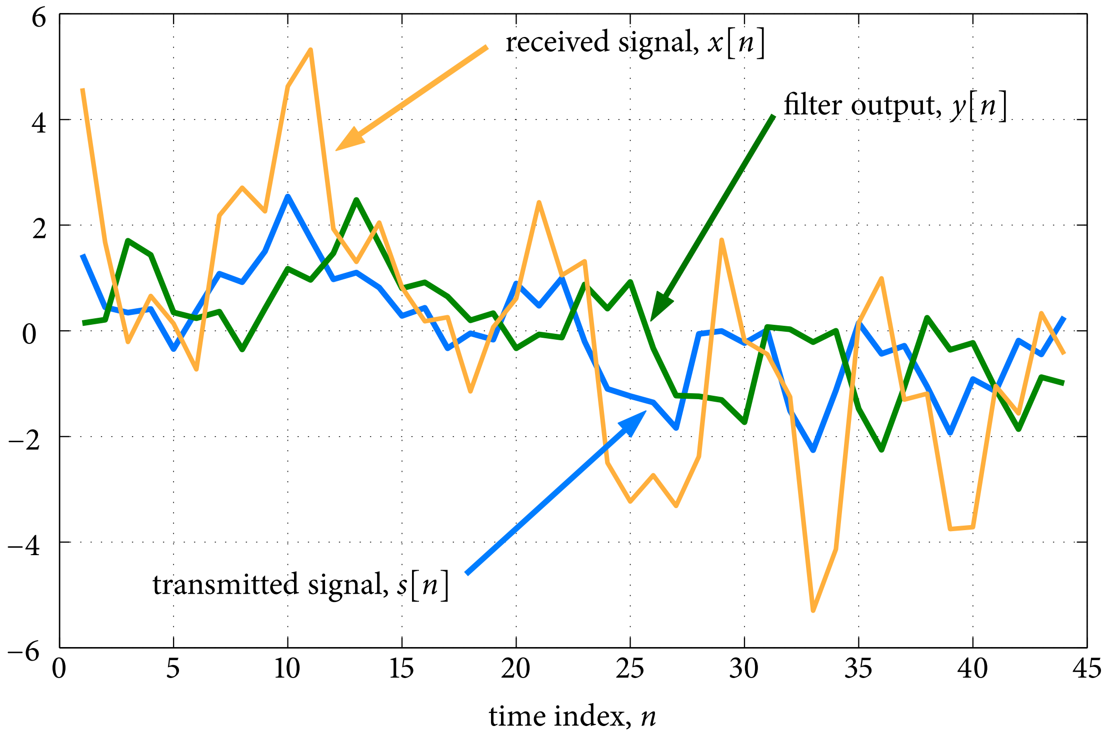
$$x[n] = s[n] + 2s[n-1] - s[n-2].$$

This signal is passed through a linear receive-side filter at which output the signal

$$y[n] = \frac{1}{17}x[n] - \frac{1}{7}x[n-1] + \frac{6}{17}x[n-2] + \frac{1}{7}x[n-3] + \frac{1}{17}x[n-4]$$

appears. It can serve as an estimate of the *delayed* transmitted signal $s[n-3]$:

$$y[n] = \underbrace{\frac{118}{119}s[n-3]}_{\substack{\text{delayed}\\ \text{\& scaled signal}}} + \underbrace{\frac{7s[n] - 3s[n-1] + s[n-2] - s[n-4] - 3s[n-5] - 7s[n-6]}{119}}_{\text{intersymbol interference}}.$$



The filter used here is a *least-squares*[1] filter. In the example from the Figure above, the signal has got a zero mean and an auto-correlation of $\mathrm{E}\big[s[n]s[n-m]\big] = 0.7^{|m|}$. From this fact follows that the signal to intersymbol-interference ratio is about 119, or about 20.8 dB. This is

1 It tries its best to equalize the channel perfectly, that is, to make the combined impulse response of the channel and the filter equal to $\delta[n-3]$, but it simply hasn't got enough degrees of freedom to achieve this perfectly. So it makes the sum of the squares of the difference of the actual combined impulse response and the ideal $\delta[n-3]$ to its absolutue minimum.

not bad, but it could be improved if the knowledge about the signal's auto-correlation was actually used in the design of the filter. But the least-squares filter does not care about it! Its design solely depends on the coefficients of the multi-path channel. However, the *linear minimum mean square error* filter does care for the signal correlation and thereby may produce better results. In our example case, the output of such a filter equals

$$y[n] = \frac{39570}{855703}x[n] - \frac{1}{7}x[n-1] + \frac{551}{1553}x[n-2] + \frac{1}{7}x[n-3] + \frac{70630}{855703}x[n-4],$$

and you shall learn in this course how and why this is so. If this $y[n]$ is taken now as an estimation of $s[n-3]$ (scaled down by the factor 0.99531), the signal to intersymbol-interference ratio becomes about 181, or some 22.6 dB. Comparing to the least-squares filter from before, this is about 1.8 dB better! This gain in performance comes about because the linear minimum mean square error filter uses information about the auto-correlation function of the signal,[1] while the least-squares filter does not.[2]

1 It also uses the auto-correlation of the noise and the cross-correlation between the noise and the signal. In the example studied here, there is no noise, though.

2 However, one could also make the point that the least-squares filter has the *advantage* that it does not *need* to know the auto-correlation of the signal!

2 2^{nd} Lecture: 9. Nov 2020

2.1 Adaptive Filters

2.1.1 Principle Structure of an Adaptive Filter

- No matter what type of parameters we put into the filter, this will always be stable.

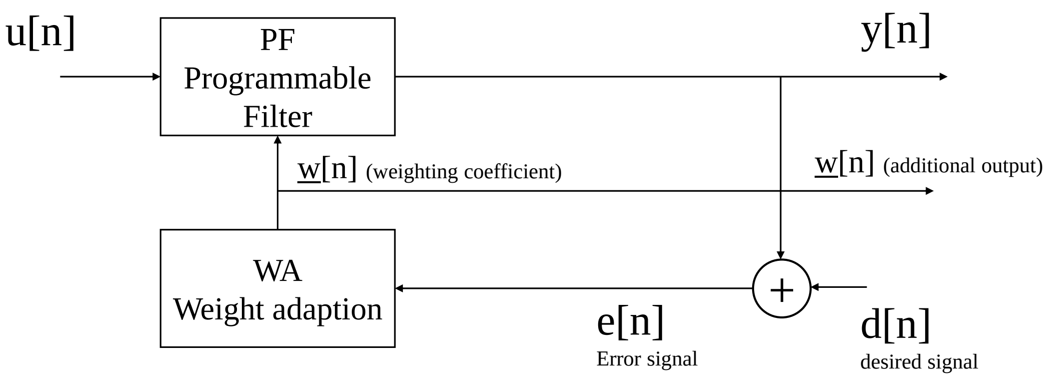


Figure 7: Principle structure of an Adaptive Filter

$e[n] = 0 \quad \Rightarrow \quad \underline{w}[n+1] = \underline{w}[n]$

- An adaptive filter is composed by two things:
    1.- Programmable Filter
    2.- Weight adaptation

2.2 Classes of Adaptive Filters

...or: What are the applications of Adaptive Filters? Adaptive Filters are used for...

2.2.1 Filter Identification

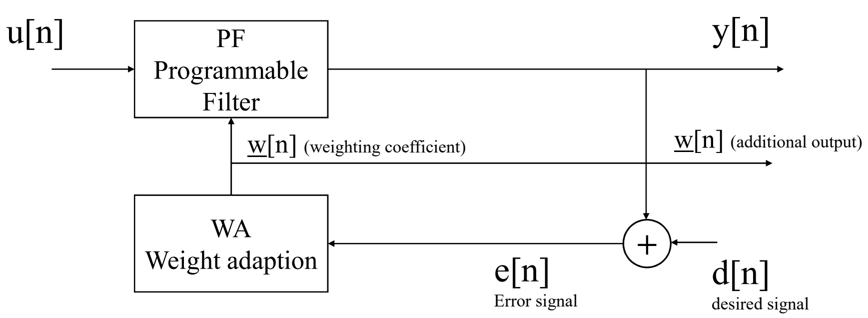


Figure 8: System identification

-> The weight w is here what we really care about.

2.2.2 System Inversion

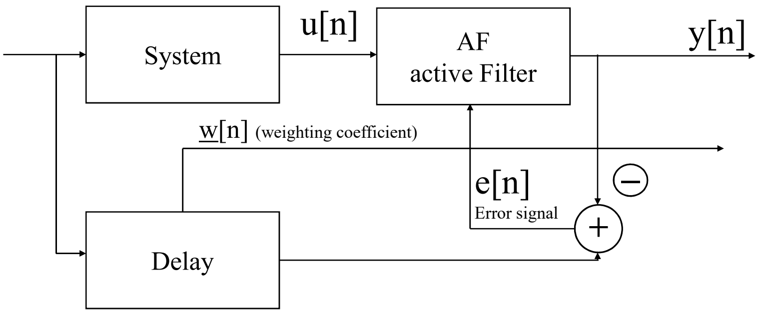


Figure 9: Inverse Modelling

- The delay is there to give more time to the AF to predict the output.

2.2.3 Prediction

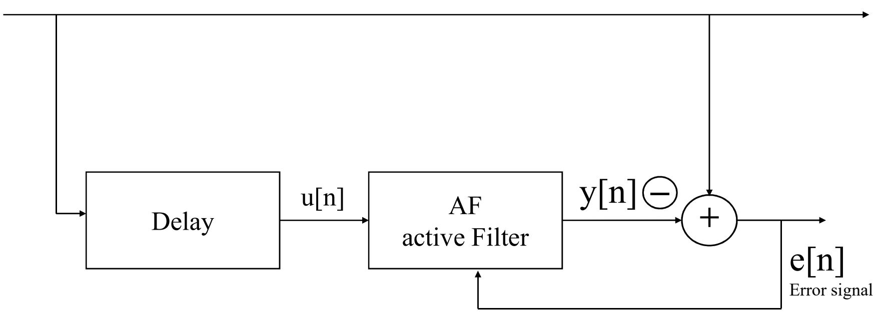


Figure 10: Prediction

Out of the past - we can predict the *present*.
We run into trouble if the error signal $e[n]$ is quantized. Therefore, we introduce a compressor and a decompressor.

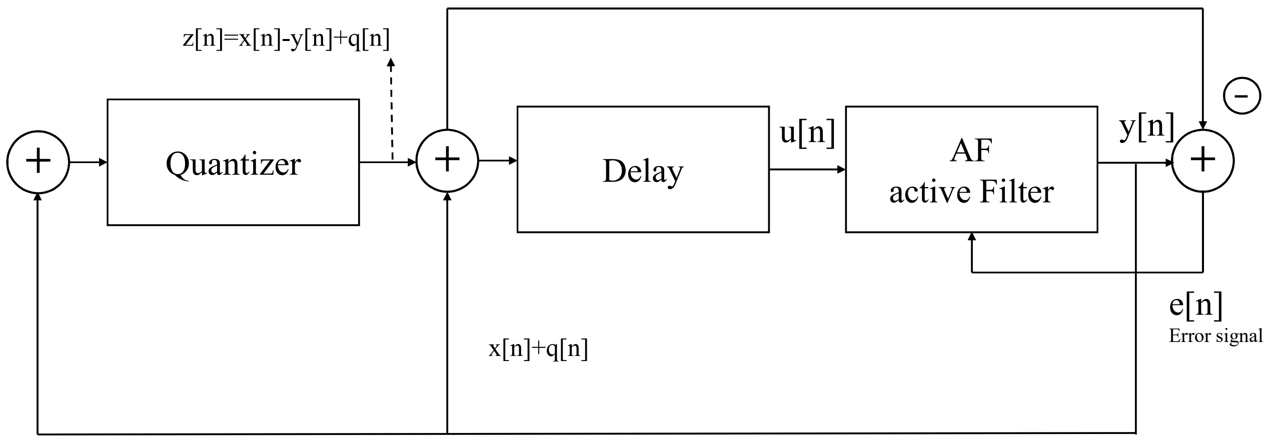


Figure 11: Compressor

Predictive filter works well $\Rightarrow$ if error is small

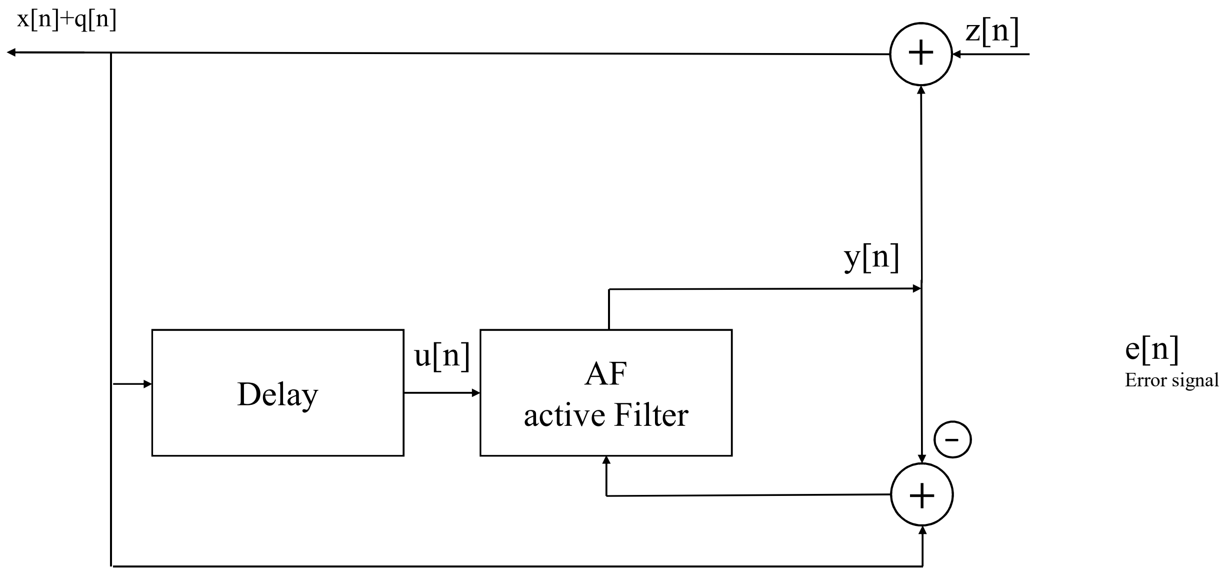


Figure 12: Decompressor

Thanks to this type Adaptive Filter, signals can be compressed to certain bit-amount (*e.g.: how many bits do we need to compress a voice signal in order for it to still be audible?*)

$\Rightarrow$ plot y-Axis:$=e[n]$

$\Rightarrow$ plot x-Axis:$=num_bits$

-> You search for compromiss. The smaller the error, the better the predictive filter.

2.2.4 Interference Cancellation

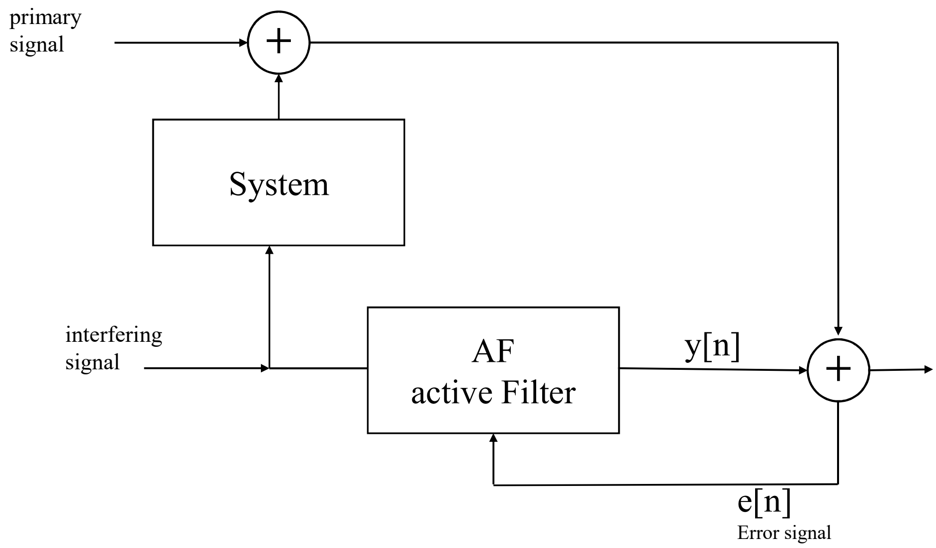


Figure 13: Interference Cancellation

Assume primary signal and interfering signal are statistically independent. A linear filter can in this case separate the primary signal from the sumated signal. A good adaptive filter can also crack the signals if the signals are statisctically dependent.
This can be ilustrated in the example of a person listening to music (noise) over speakers and speaking to a microphone at the same time.

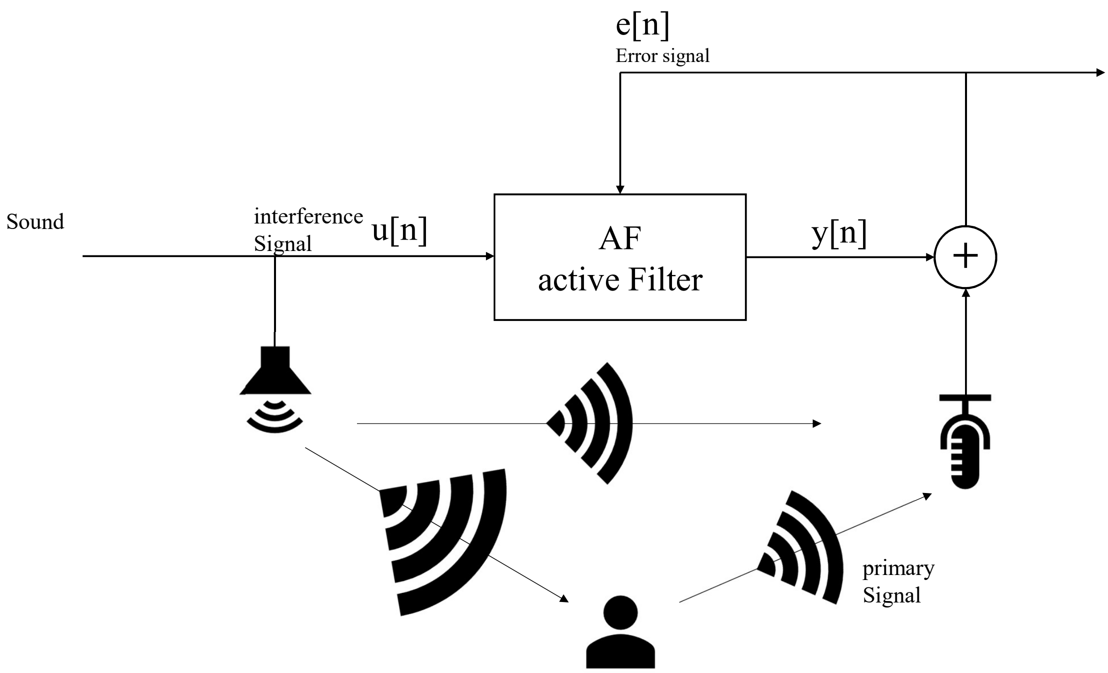


Figure 14: Example for an application for an interference-cancelling-filter

In Interference Cancellation - in the case of audio - the interference signal is captured and sent to the Adaptive Filter at the soundcard. Linear filters are better than nonlinear ones for this case. Filter tries to "'kill"' the interference because it knows, what has been played on the speaker.

2.3 A Time Signal

$$x(t) = A(t)cos(2\pi f_0 t + \varphi(t))$$

$$\varphi := \text{Phase Modulation}$$
$$A(t) := \text{Amplitude Modulation}$$

- Frequency modulation is just another way of Phase Modulation. (f<->t)

$$x(t) = \text{Re}\left\{u(t)e^{j2\pi f_0 t}\right\}$$
with

- $x(t) \in \mathbb{R}$: Real Signal
- $u(t) \in \mathbb{R}$: Complex Envelope of $x(t)$
- f_0: Center of Frequency; $[f_0] = 1Hz$

Notation with Magnitude and phase:

$$u(t) = |u(t)| \cdot e^{j\varphi_u(t)}$$

$x(t) \iff u(t) \quad \Rightarrow$ It's not a fourier transform, but related. Are different representations of the same thing.

$$x(t-T) \iff u(t-T)e^{-j2\pi\varphi T}$$
$$a \in R \quad \Rightarrow \quad a \cdot x(t-T) = u(t-T) \cdot e^{-j2\pi\varphi_0 T} \cdot a = w^* \cdot u(t-T)$$
$$w* \in \mathbb{C} \quad \Rightarrow \quad w^* = e^{-j2\pi\varphi_0 T} a$$

2.4 Constraints for Bandwidth

$$-2f_0 + \frac{B}{2} \leq -\frac{B}{2}$$
$$\boxed{\Rightarrow \quad B \leq 2f_0 \qquad \text{or} \qquad f_0 \geq \frac{B}{2}}$$

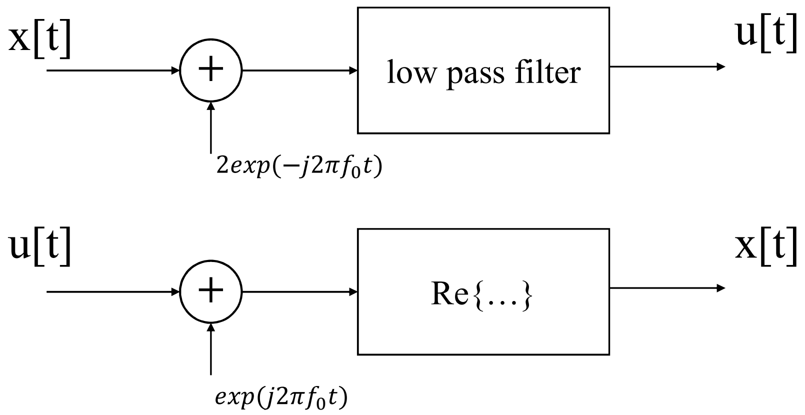


Figure 15: Transformation from complex envelope and signal as block diagram and vice versa

2.5 Narrow Band

Assumption:

$$u(t - 1/f_0) \approx u(t)$$
$$f_0 \triangle t = ((\varphi w)/(2\pi) - f_0 \tau) mod(1)$$
$$\triangle t = ((\varphi w)/(2\pi f_0) - \tau) mod(1/f_0)$$

then...

$$\tilde{\mathrm{t}} = ((\varphi w)/(2\pi f_0) - \tau) mod(1/f_0)$$
$c = |w|$:= Complex envelope (only narrow bands)

f_0 := Carrier Frequency

$$abs(w)x(t - T - \triangle t) \iff w^* u(t - T) \qquad \triangle t = (avg(w)/(2\pi f_0) - T) mod(1/f_0)$$

2.6 Planar Waves

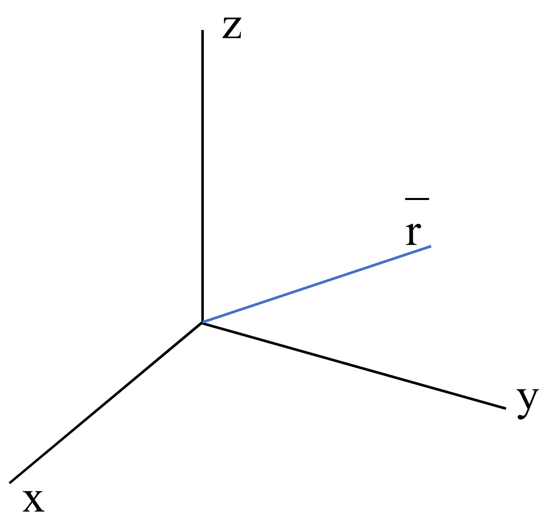


Figure 16: Direction of planar wave

$\Rightarrow$ A wave is relationship of time and space

$P(t, \vec{\boldsymbol{r}}) = F(t - \vec{\boldsymbol{n}} * \vec{\boldsymbol{r}}/c)$

$\Rightarrow$ $\vec{\boldsymbol{n}}$:= unit vector $\vec{\boldsymbol{n}}$*$\vec{\boldsymbol{n}}$=1

$\Rightarrow$ c > 0 := Wave Propagation Speed

$\Rightarrow$ $\triangle\vec{\boldsymbol{r}} * \vec{\boldsymbol{n}} = 0$

$P(t, \vec{\boldsymbol{r}}) = p(t, \vec{\boldsymbol{r}} + \triangle\vec{\boldsymbol{r}})$

Property 1

$\forall\Delta\vec{\boldsymbol{r}} : \Delta\vec{\boldsymbol{r}} \cdot \vec{\boldsymbol{n}} = 0 \qquad \Rightarrow \quad q(t, \vec{\boldsymbol{r}} + \Delta\vec{\boldsymbol{r}}) = q(t, \vec{\boldsymbol{r}})$

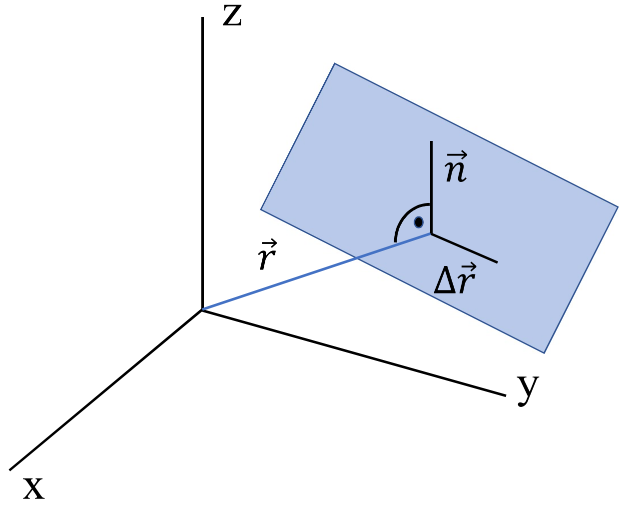


Figure 17: plane in space

Property 2
$q(t + \Delta t, \overrightarrow{\boldsymbol{r}} + \Delta t \cdot c\overrightarrow{\boldsymbol{n}}) \equiv q(t, \overrightarrow{\boldsymbol{r}})$
Moving of $q(t, \overrightarrow{\boldsymbol{r}})$ into direction of $\overrightarrow{\boldsymbol{n}}$ with speed c.

2.6.1 Harmonic Waves

Harmonic planar waves have a third property. A harmonic wave has
$F(t) = Acos(2\pi f_0 t + \varphi) \quad \Rightarrow \quad$ all const apart of t
$P(t, \overrightarrow{\boldsymbol{r}}) = Acos(2\pi f_0(t - (\overrightarrow{\boldsymbol{n}} * \overrightarrow{\boldsymbol{r}})/c) + \varphi)$
$\Rightarrow \quad$ complex envelope
$= \mathrm{Re}\left\{u(\overrightarrow{\boldsymbol{r}}) \cdot e^{j2\pi f_0 t}\right\}$

$u(\overrightarrow{\boldsymbol{r}}) = Ae^{-j2\pi f_0(\overrightarrow{\boldsymbol{n}} * \overrightarrow{\boldsymbol{r}})/c)} + e^{j\varphi}$
$u(\overrightarrow{\boldsymbol{r}}) = Ae^{j\varphi} \cdot e^{-j2\pi f_0(\overrightarrow{\boldsymbol{n}} * \overrightarrow{\boldsymbol{r}})/c)}$
$\Rightarrow \quad Ae^{j\varphi} \in \mathbb{C}$ we call it s
Property 3 This property is only for harmonic plane waves

$\Rightarrow \quad \mathbb{C}$-Envelope: $u(\overrightarrow{\boldsymbol{r}} + \frac{c}{f_0}\overrightarrow{\boldsymbol{n}}) \equiv u(\overrightarrow{\boldsymbol{r}}) \qquad \lambda$ wave length; $\lambda = \frac{c}{f_0}$

$\Rightarrow \quad u(\overrightarrow{\boldsymbol{r}}) = s \cdot e^{-j2\pi \frac{\overrightarrow{\boldsymbol{n}} \cdot \overrightarrow{\boldsymbol{r}}}{\lambda}}$

2.7 Spatial Sampling

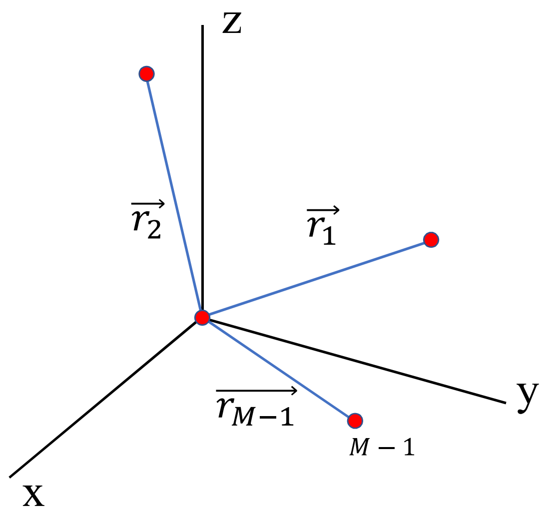


Figure 18: Spatial Sampling with M sensors

$$\underline{\boldsymbol{u}}[t] = \begin{bmatrix} s(t) \\ s(t - \frac{\vec{n}\cdot\vec{r}_1}{c})e^{-j2\pi\frac{\vec{n}\cdot\vec{r}_1}{\lambda}} \\ \vdots \\ s(t - \frac{\vec{n}\cdot\vec{r}_{M-1}}{c})e^{-j2\pi\frac{\vec{n}\cdot\vec{r}_{M-1}}{\lambda}} \end{bmatrix} \quad \Rightarrow \quad \text{Sensor array (receive vector)}$$

Definition coherence: Coherence (physics), an ideal property of waves that enables stationary (i.e. temporally and spatially constant) interference.