Lab 2: The $\mathcal{Z}$-transform and filter design

The $\mathcal{Z}$-transform

The $\mathcal{Z}$ transform is a slightly different way of understanding the Fourier transform. It is a complexification, given a signal $x[n]$ its $\mathcal{Z}$-transform is given by a complex meromorphic function:

$$\mathcal{Z}\{x[n]\} = X(z) = \sum_{k\in\mathbb{Z}} x[k]z^{-k}.$$

We use the same notation as in the Fourier transform because if we restrict the $\mathcal{Z}$-transform to the unit circle, i.e. we replace z by $e^{i\omega}$, we get exactly the Fourier transform: $X(e^{i\omega}) = \sum_{k\in\mathbb{Z}} x[k]e^{-ik\omega}$. Thus, we observe that we can recover as before a signal $x[n]$ from its Z-transform.

We know that any LTI system is determined by its impulse response function $h[n]$. The $\mathcal{Z}$-transform of the impulse response function $H(z)$ is called the system function (or transfer function) and it also characterizes the LTI system. At the level of system functions we have that $y = T(x)$ if and only if:

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

because of the convolution theorem. Most of the LTI systems T that are implemented in practice have a system function H of the form

$$H(z) = \frac{P(z)}{Q(z)} = \frac{\sum_{k=0}^{M} b_k z^{-k}}{\sum_{k=0}^{N} a_k z^{-k}}.$$

This is not a very serious restriction because any continuous function in the unit circle can be arbitrarily well approximated with the uniform norm by a rational function by Mergelyan theorem or Stone-Weirstrass theorem.

If we consider a system with rational system function the equation:

$$Y(z) = X(z)\frac{\sum_{k=0}^{M} b_k z^{-k}}{\sum_{k=0}^{N} a_k z^{-k}},$$

or equivalently

$$\left(\sum_{k=0}^{N} a_k z^{-k}\right) Y(z) = \left(\sum_{k=0}^{M} b_k z^{-k}\right) X(z),$$

Multiplication by z^{-1} corresponds to a shift of the sequence therefore this is equivalent to which implies that

$$\begin{aligned}a[0]y[n] =& b[0]x[n] + b[1]x[n-1] + \cdots + b[M]x[n-M] \\ &- a[1]y[n-1] - \cdots - a[N]y[n-N]\end{aligned}$$

This is a nice recursion formula that allows to compute $y[n]$ from the values of $x[n]$. This is what the Matlab/Octave function filter does. The cost of filtering the signal is $length(x) \times max(N+M)$.

Observe that applying an LTI system with rational transfer function is equivalent to solving a non-homogeneous recurrence equation with constant coefficients, and viceversa. We are using a "brute force" approach to solving the recurrence equation as a way to apply T to x.

Moreover any rational function can be factorized over the complex numbers as a product of more "elementary" factors (for instance the degree can be chosen to be at most one). This allows to express $H(z) = H_1(z) \cdots H_k(z)$ and it induces a decomposition of the corresponding LTI system T as a composition of more elementary systems $T = T_1 \circ \cdots \circ T_k$. To facilitate this decomposition, sometimes it is more convenient to provide the zeros and poles of H instead of the coefficients of P and Q.

This is called the zeros-poles representation of the LTI system and it is considered as one of the more convenient way to represent an LTI system.

Many properties of the convolution filters can be described in terms of the system function. For instance, its stability:

Definition 1. A system T is BIBO (bounded input, bounded output) stable if for any bounded initial signal, i.e $\sup_n |x[n]| < \infty$, we get that its image $y = T(x)$ is also bounded. That is, it maps $\ell^\infty(\mathbb{Z})$ to itself.

This is achieved in particular if the impulse response $h[n]$ is in ℓ^1, it is an easy exercise to check that an ℓ^1 sequence convolved with an ℓ^∞ sequence is bounded. For LTI systems with system function a rational function, if the system has no poles in the unit circle, then we can develop H in its Laurent series around the origin and check that $h[n]$ is in ℓ^1. Thus the system is BIBO-stable if its system function $H(z)$ has no poles on the unit circle. One has to be a bit careful here because there are different Laurent series associated to a rational function, depending on the region of convergence that we choose. If H has no poles in the unit circle, then we should take the Laurent series representation that converges in the unit circle. This would provide an impulse response function $h[n]$ that is BIBO stable.

The Paley Wiener theorem in the discrete context asserts that

$$\frac{1}{2\pi} \int_{-\pi}^{\pi} |X(\omega)|^2 \, d\omega = \sum_{n \in \mathbb{Z}} |x(n)|^2.$$

Thus if the transfer function H is bounded on the unit circle (there are no poles in it) then the operator $L^2([0, 2\pi]) \to L^2([0, 2\pi])$ that sends $F \to HF$ is bounded in $L^2([0, 2\pi])$ and thanks to the Paley-Wiener theorem it is also bounded from ℓ^2 to ℓ^2.

1. Filter design

The design of filters with desired characteristics is an important problem in signal processing and electric engineering.

The filters are LTI systems, that can be described by the impulse response function $h[n]$ or, more typically, its frequency response $H(\omega)$.

In practice there are two big families of filters: the FIR filters and the IIR filters.

1.1. The FIR filters and windowing. This are Finite Impulse Response Filters. By definition they are provided by a system with impulse response h.

In the case of FIR filters the filters have an impulse response that is a polynomial in z^{-1} $H(z) = \sum_{k=0}^{M} h[k]z^{-k}$. They are easily implemented as convolution operators.

The ideal low pass filter $h[n]$ would have frequency response

$$H_{ideal}(\omega) = \begin{cases} 1, & |\omega| < \omega_c \\ 0, & \omega_c < |\omega| \leq \pi. \end{cases}$$

The corresponding impulse response function $h[n]$ is

$$h_{ideal}[n] = \frac{\sin \omega_c n}{\pi n}, \qquad \forall n \in \mathbb{Z}.$$

This filter is not easily implemented (the transfer function is not rational). So the standard way to construct a pracical filter is to multiply it by a window of finite duration $w[n]$, $h[n] = h_{ideal}[n]w[n]$. The frequency response of the practical filter provided by $h[n]$ is

$$H(e^{i\omega}) = \frac{1}{2\pi} \int_{-\pi}^{\pi} H_{ideal}(e^{ix}) W(e^{i(\omega - x)})\, dx.$$

(Observe that I abuse the notation and I sometimes I denote the frequency response as $H(\omega)$ and other times as $H(e^{i\omega})$ depending of the convenience.) We usually have two (contradicting) requirements for a window $w[n]$. We want it to be "short" (small support, so that the filter is not expensive to apply) and we want it to have small support for its Fourier transform, so that H does not differ much from H_{ideal}. There are many popular windows: rectangular, Hamming, Bartlett, Hann, Blackman, etc... They look as in the illustration of Figure 1.

They are implemented in Octave/Matlab. You can produce a fir filter by windowing using the function fir1. You can produce more sophisticated filters with the funcition fir2 that uses the FFT (Fast Fourier Transform) to prescribe the filter. There are also interesting

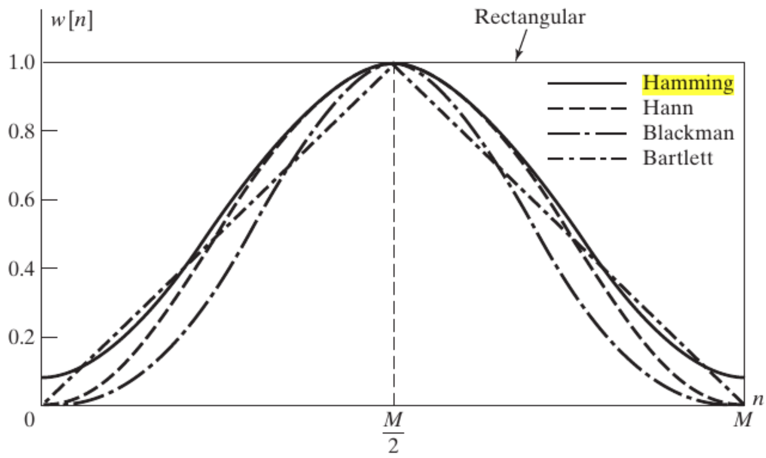

FIGURE 1. Several possible windows

mathematical problems to find polynomials of small degree that approximate optimally an ideal filter, these are the so called equiripple filters.

To examine the frequency effect that a given filter has on a signal we can use the function freqz to analyze it.

I propose to do the following exercise (inspired from an engineering course at Purdue University)

EXERCISE 1. *You will design a simple second order FIR filter with the two zeros on the unit circle. In order for the filter's impulse response to be real-valued, the two zeros must be complex conjugates of one another:* $z_1 = e^{i\theta}$, $z_2 = e^{-i\theta}$. *The transfer function for this filter is given by*

$$H(z) = 1 - 2\cos\theta z^{-1} + z^{-2}.$$

Use this transfer function to determine the recursion formula (also known as the difference equation) for this filter and compute the filter's impulse response $h[n]$*. This filter is a FIR filter because it has impulse response* $h[n]$ *of finite duration. Any filter with only zeros and no poles other than those at* 0 *and* ∞ *is a FIR filter. Zeros in the transfer function represent frequencies that are not passed through the filter. This can be useful for removing unwanted frequencies in a signal. The fact that* $H(z)$ *has zeros at* $e^{i\theta}$ *means that the filter will not pass pure sine waves at frequency* $\omega = \theta$*. Use Matlab to compute and plot the magnitude of the filter's frequency response* $|H(e^{i\omega})|$ *as a function of* ω *on the interval* $-\pi < \omega < \pi$*, when* $\theta = \pi/6$.

In the next experiment, we will use the filter $H(z)$ *to remove an undesirable sinusoidal interference from a speech signal. To run the experiment, first read the file "easy.wav" that is provided with the instruction:*

```
[x,FS] = audioread('easy.wav');
```

The vector x represents the audio signal and FS is an integer that denotes the frequency at which it has been sampled. You can play the sound with the instruction

```
player = audioplayer (0.8*x, FS);
play (player);
```

We add some sinusoidal noise to the signal

```
t = (1:length(x))/FS;
y = 0.8*x + 0.1*sin(35000* t')
```

We hear the noisy signal with

```
player = audioplayer (y, FS);
play (player);
```

Design a filter H with two zeros, that atenuates the noise, filter the signal and hear it again.

How can we choose θ?

Bibliography

[1] Ronald W. Schafer. Alan V. Oppenheim, *Discrete-time signal processing.*, Prentice Hall, 2010.