L2. IIR filter design with the Pade method

Statistical Data Processing - Lab 2

Filter design

The Pade method is a way to design an IIR filter such that its impulse reponse $h[n]$ meets certain requirements.

Designing a filter means finding the values of the numerator coefficients $b_0, b_1, ...b_M$ and of the denominator coefficients $a_1, a_2, ...a_N$ of a system function:

$$H(z) = \frac{b_0 + b_1 z^{-1} + b_2 z^{-2} + ... + b_M z^{-M}}{1 + a_1 z^{-1} + a_2 z^{-2} + ... + b_N z^{-N}},$$

such that the filter does what it is required to do.

The Pade method

In the Pade method, we want to design the filter $H(z)$ such that the impulse response $h[n]$ is similar to a desired impulse response $h_d[n]$.

The desired impulse response $h_d[n]$ can be, for example, the impulse reponse of an ideal low-pass, band-pass or high-pass filter (which needs a filter of infinite order so can't be implemented exactly).

The idea of the Pade method is the following: let's select $b_0, b_1, ...b_M$ and $a_1, a_2, ...a_N$ in order to make the **first samples of $h[n]$ identical to the ones of $h_d[n]$**. Why? Because in many cases the first samples are the most important, and if we manage to make the first equal, we can reasonably assume that the following ones will not matter so much, and therefore our filter $h[n]$ will be reasonably similar to the desired $h_d[n]$.

Mathematical procedure

Start from the **difference equation** of the system:

$$y[n] = -a_1 y[n-1] - ...a_N y[n-N] + b_0 x[n] + b_1 x[n-1] + ...b_M x[n-m]$$

If the input is $\delta[n]$, the output will be $h[n]$, therefore:

$$h[n] = -a_1 h[n-1] - ...a_N h[n-N] + b_0 \delta[n] + b_1 \delta[n-1] + ...b_M \delta[n-m]$$

Make first $M + N + 1$ samples equal to those of $h_d[n]$. The number of samples is the same as the total number of coefficients. Assume the initial conditions $h[-1] = h[-2] = ... = 0$ (if not specified otherwise).

We have the following system:

$$h[0] = b_0 = h_d[0]$$

$$h[1] = -a_1 h_d[0] + b_1 = h_d[1]$$

$$h[M] = -a_1 h_d[M-1] - \dots - a_M h_d[0] + b_M = h_d[M]$$

$$h[M+1] = -a_1 h_d[M] - \dots - a_M h_d[M+1-N]$$

$$h[M+N] = -a_1 h_d[M+N-1] - \dots - a_M h_d[N]$$

There should be a total of $M+N+1$ equations, sufficient for a total of $M+N+1$ unknowns a_i and b_i to be found.

Solve the system. When solving by hand, note that:

- The last N equations do not depend on b_i. Solve them first, and obtain the values of the coefficients a_i
- Then move to the first $M+1$ equations. Replace a_i with the values found above, and find the b_i. Each equation should give one value of a b_i.

Exercise / Example at blackboard

1. Use the Pade method to find out the parameters of the system with the following system function of order 2:

$$H(z) = \frac{b_0 + b_1 z^{-1} + b_2 z^{-2}}{1 + a_1 z^{-1} + a_2 z^{-2}}$$

considering the desired impulse response:

$$h_d[n] = \left(\frac{1}{3}\right)^n \cos\left(\frac{n\pi}{4}\right).$$

Matlab exercise

1. Implement in Matlab a function for creating and then solving the equation system resulting from the Pade method.

The functions should be called as:

```
[b,a] = pade(order, hd)
```

The function shall have the following arguments:

- `order`: the order of the designed filter
- `hd`: a vector holding the first samples of the desired impulse response

The function shall return the coefficients of the system function for the resulting filter:

- `b`: a vector with the numerator coefficients

- `a`: a vector with the denominator coefficients

2. Use the function above to design a second order filter with the Pade method, for approximating the desired impulse response given below:

$$h_d[n] = \left(\frac{1}{3}\right)^n \cdot \cos\left(\frac{\pi}{4} n\right) \cdot u[n]$$

3. Load a sample audio file in Matlab and filter it with the filter found above.

Play the filtered signal. How does it sounde like? Compare it with the original signal.

Final questions

1. TBD