DTU Health Technology, Hearing Systems section
Building 352, DTU, 2800 Lyngby
Phone: 45 25 39 30, Fax: 45 88 05 77 www.hea.healthtech.dtu.dk/

Course 22051: Signals and Linear Systems in Discrete Time

22051 E21

Cloudy...

Bastian Epp: 45 25 39 53 (office), e-mail: bepp@dtu.dk

Filters, filters, filters...

1 Which one...?

There exist a number of different filter approximations - each having it's pros and cons. The question might arise: Which one should I use now given a certain problem? The answer to that is not simple. Depending on the requirements, it is on the DSP engineer to decide which filter is to be used given the filter specifications and, potentially, the hardware boundary conditions. The order required to meet certain specifications varies between the filter approximations. Higher order means a higher need of "components" to implement the filter and hence longer impulse responses and higher cost. In this part of the exercise a direct comparison will be made for different filter design methods, all aimed to meet the given filter specifications. The filter design does not need to be done completely with pen-and-paper, but it is extremely important that you have a feeling about if what comes out of the Matlab functions make any sense.

This comparison of the filter realizations will give you some feeling on what happens when you design these filters and you can see which particular properties the approximations inherently have.

1.1 Given the specifications

You get the task to design a filter that meets a couple of specifications given below. Since your boss wants to save some money, he/she tells you to provide him/her with a filter that is as economical as possible. What you have is a device that samples at frequency of 20 kHz - and you are supposed to use it! The statement of your boss is: "How hard can it be to cut off a couple of frequencies? Just cut them off and get going!". You remember from your DSP course that there was a big fuzz about how to design a filter - but maybe your boss is right? Why not just cut off the higher frequencies? So you get started:

Design a filter that meets the specifications below[1]. Use different approaches (Butterworth, Chebychev I/II, Elliptic, simple "cutting" in the frequency domain) and decide which filter you would present to your boss. Collect and discuss arguments for your decision:

- Gain of no less than -2 dB in the frequency range [0, 1000] Hz
- Gain of not more than -40 dB in the frequency range [2000, 10.000] Hz
- Ripples might be accepted if they don't exceed 1 dB

```
useful commands:  buttord, cheb1ord, cheb2ord, ellipord, butter, cheby1,
cheby2, ellip, zplane, freqz, fft, filter
```

For each filter, provide (if applicable) the order n, the frequency response, the phase and the impulse response. Compare these across the filters you designed.

[1]Once done, you might also swap the frequency ranges to design a high-pass filter having similar specs - not to annoy your boss, but this might be interesting to see what that does to the specifications

1.2 Given the specifications - and here comes the boss again!

Sometimes you boss is a bit overworked - and forgets certain things. So after having solved the problem stated above, he comes again into your office, throws the same device on your table (that you just returned yesterday with your beautiful solution) and provides you with the following task:

Design a filter that meets the specifications below. Use different approaches (Butterworth, Chebychev I/II, Elliptic, simple "cutting" in the frequency domain) and decide which filter you would present to your boss. Collect and discuss arguments for your decision:

- Gain of no less than -2 dB in the frequency range [0, 1000] Hz
- *Gain of not more than -40 dB in the frequency range [2000, 20.000] Hz*
- Ripples might be accepted if they don't exceed 1 dB

  `useful commands: buttord, cheb1ord, cheb2ord, ellipord, butter, cheby1, cheby2, ellip, zplane, freqz, fft, filter`

What do you tell him? And why?

1.3 Listening to the filters

After having designed all those filters you also want to see how they perform. So pass a signal through your filter with which you can perceptually evaluate what is going on (you might also evaluate a running sum or others). A good signal for such a purpose is a white noise:

```
>> n = randn(fs*T,1); signal = n / (max(n))*.99;
```

with sampling frequency `fs` and a duration in seconds `T` or a sweep:

```
>> s = chirp(t, F0, T1, F1); signal = s*.99;
```

with a time vector `t`, a starting frequency `F0` and a frequency `F1` reached at time `T1`.

2 Hummmmmmmmm

Do you know that situation that there is this annoying 50 Hz humming on your favourite piece of music? This is probably line noise that made it somehow into your signal. Sources of line noise are everywhere - power plugs, light sources.... EVERYWHERE!

Import and mix the two sound files "hum.wav" and "scrambled_eggs_5s.wav" into one sound file that is "corrupted" by line noise. Then design a filter the removes the 50 Hz component of the signal. Provide the transfer function and listen to the result. Would you be happy with it?