Project 2: Design a digital filter to recover a masterpiece

DEADLINE: 2015-02-23, 23:59

EMBEDDED SIGNAL PROCESSING SYSTEMS VT 2015

Instructions

- This project (the second "inlämningsuppgift") is aimed at a DSP application: *Design a digital filter to recover a masterpiece, Air on the String by J. S. Bach,* which is noisy and interfered with an unwanted annoying tone. To this purpose, you are supposed to design a digital filter and implement it on the AVR UC3A3 Xplained evaluation board by using the theoretical and practical knowledge you have learnt about digital filters.
- *The project is to be made in groups of two students.*
- Hand in the report before the deadline.

Equipment, devices, hardware, software and signal/data files:

- a computer installed with MATLAB, Atmel Studio 6, Flip and Putty
- an AVR UC3A3 Xplained evaluation board
- the DIY circuit
- Atmel AVR32 DSPLib and Atmel software framework and (ASF)
- A software framework (in C), *DSP framework.c*, that includes ADC, DAC, external trigger and USART is available for implementing digital systems and performing real time DSP. Note that Atmel Software Framework AVR32 DSPLib is good DSP resource.
- 2 WAV-files, `Noisy_interfered_Music.wav` and `Backgroud_noise.wav`, are packed into a compressed file *Proejct_2.rar*. Each file contains a 20 seconds long audio signal.

Contact me if you have any questions on the project: Ping.Wu@angstrom.uu.se or 070 167 9407

---

1. Introduction

Noise (e.g., thermal noise, ambient noise, *etc.*) are pervasive, and unwanted interference (e.g., 50 Hz power line interference, manufacturing noise from cutting machines, *etc.*) are often present and unnoticed. To eliminate the influence of noise and interference, digital filters of appropriate types can be used. Noise and interference are of different types. It would be impossible to have a filter that can manage all types of noise and interference. Therefore, a filter is often application oriented.

In the present project, the application of DSP is to eliminate noise and interference that came into a masterpiece, *Air on the String by J. S. Bach*. To this end you are going to design a digital filter and implement it on an embedded DSP system based on the AVR UC3A3 Xplained evaluation board. In other words, the DSP system is designed and implemented as a filter that is aimed to eliminate (filter out) the noise and interference.

2. Tasks and specifications

This project *Design a digital filter to recover a masterpiece* is aimed at designing digital filters, which include

1) an FIR filter that is suggested to reduce the background noise
2) a filter that can eliminate both the noise and the interference (a filter for reducing interference is suggested in Appendix 1)

The project report needs to include

- A short introduction to digital filters and design methods
- A detailed description of the filters (e.g., FIR or IIR filters of the different types - low pass, high pass, band pass, or the combination of them, etc.) chosen for the tasks
- The frequency responses of the designed filters and their plots made in MATLAB
- Implementation of the filters on the embedded system
- Results and discussions
- Conclusions
- C codes (**only your own part**!) in the appendix

3. Implementation issues

Different from a block of data that present just once within a short time like DTMF touch tone signals, an audio signal (e.g., songs or music played with any media player, and talks and presentations given in lectures and conferences) consists of bit stream data that last in real time over a certain period from minutes to hours). When a DSP system processes such signals, any time gaps between processed data blocks that are caused by the processing would create disturbing sound.

To design a filter that can fulfill the given specifications, it is necessary to analyze the signal thoroughly in the beginning. In the time domain analysis, you need to figure out when the signal, the noise and the interference are presented, respectively, and in the frequency domain, you need to sort out which frequency ranges the signal, the noise and the interference are respectively distributed. Only after such signal analysis is correctly done, you can design a filter that can fulfill the predefined specifications.

Noise can be of different kinds (http://en.wikipedia.org/wiki/Colors_of_noise). It is good to know it before conducting any noise reduction. Sometimes noise can be helpful (e.g., http://www.tinnitus.org.uk/products/14). But much more often noise is disturbing and annoying, e.g., in this project, a masterpiece music which becomes noisy and interfered with some unwanted tone. Noise may come from various sources. The background noise that has got into the music is given in the WAV-file: `Backgroud_noise.wav`, which is packed into a compressed file *Proejct_2.rar* together with `Noisy_interfered_Music.wav`.

4. Requirements “för att bli godkänd”

The project needs to be finished in time and the report handed in latest on the deadline. The complete source codes for the project are required to be handed in together with the report but in a separate file, e.g., with the name *main.c*.

A demo day will be arranged on which all groups have chance to demonstrate their projects. The performance of the works is going to be evaluated. The project that you have done will be graded in two levels U and G.

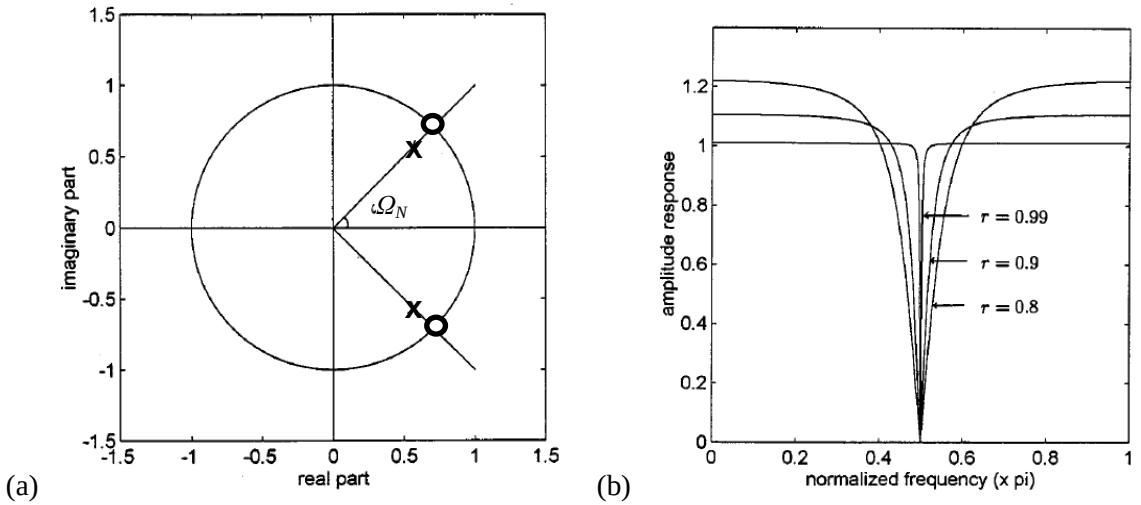
“För att bli godkänd”, the implemented DSP system intended for the specified purpose needs to be shown to work properly, showing good noise reduction; the report needs to clearly state basics about digital filters (e.g., FIR and IIR filters of different types – LP, HP, BP and BS), explain how to choose filters and design them for the given applications, present how to implement the algorithm on the embedded system, and give a clear illustration of results together with a proper discussion.

APPENDIX 1. DESIGN OF IIR NOTCH FILTER – Based on the zero-pole placement

A notch filter is very useful when a certain frequency component in a signal needs to be removed. A digital notch filter can be designed in different orders. An often seen notch filter is the second order

$$H(z)=\frac{(z-z_1)(z-z_2)}{(z-p_1)(z-p_2)}=\frac{(z-e^{-j\Omega_N})(z-e^{j\Omega_N})}{(z-re^{-j\Omega_N})(z-re^{j\Omega_N})}=\frac{z^2-2z\cos\Omega_N+1}{z^2-2rz\cos\Omega_N+r^2}$$

Zeros and poles in the transfer function $H(z)$ yield zeros and peaks in the frequency response $H(\Omega)$, respectively. The closer the poles to the unit circle (i.e., the closer the radius r to 1), the narrower the notch.



(a) (b)

Fig. A-1. The notch filter: (a) placement of zeros o and poles x and (b) the frequency response

The procedure for designing the above notch filter is

1. Place a pair of zeros (complex conjugate) on the unit circle exactly at the notch frequency f_N
2. Place a pair of poles (complex conjugate) close to the unit circle ($r < 1$) exactly at f_N where r is determined by

$$r=1-\frac{BW}{F_s}\pi$$

Where BW is the bandwidth of the notch, and F_s is the sampling frequency.

3. Calculate Ω that corresponds to the notch frequency f_N

$$\Omega_N=2\pi\frac{f_N}{F_s}$$

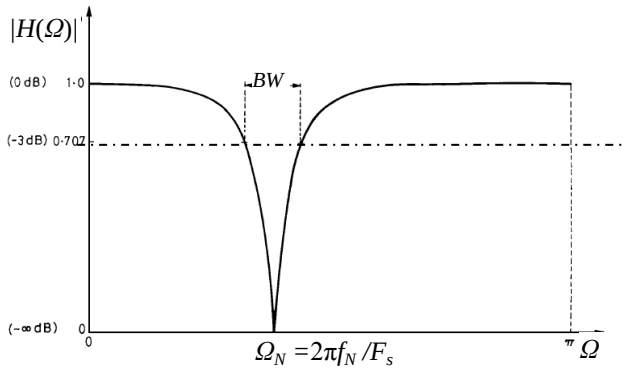


Fig. A-2. Notations for the notch filter

4. Implement $H(z)$

$$H(z)=\frac{(z-e^{-j\Omega_N})(z-e^{j\Omega_N})}{(z-re^{-j\Omega_N})(z-re^{j\Omega_N})}=\frac{z^2-2z\cos\Omega_N+1}{z^2-2rz\cos\Omega_N+r^2}$$