**Department of Electrical Engineering**

|  |  |
| --- | --- |
| **Faculty Member:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_** | **Dated: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_** |
|  |  |
| **Course/Section:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_** | **Semester: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_** |
|  |  |

**EE-330 Digital Signal Processing**

**Lab#10 IIR Filter Design using Analog Prototypes**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  | **PLO4-CLO4** | | **PLO5-CLO5** | **PLO8-CLO6** | **PLO9-CLO7** |
| **Name** | **Reg. No** | **Viva / Quiz / Lab Performance** | **Analysis of data in Lab Report** | **Modern Tool Usage** | **Ethics and Safety** | **Individual and Team Work** |
|  |  | **5 Marks** | **5 Marks** | **5 Marks** | **5 Marks** | **5 Marks** |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |

**Lab10: IIR Filter Design using Analog Prototypes**

**Objectives**

The objective is to design digital filters with an infinite impulse response (IIR). IIR filters can be designed either directly, by means of a clever placement of poles and zeros in the z-plane, or “indirectly” by first designing a corresponding analog filter, which is then transformed to the digital domain. In this lab we will only consider the later design procedure. Note that the described principle of using analog prototypes for IIR filter design originally stems from the idea to use digital filters as a substitute for analog systems. This allows us to use the well-constructed theory of analog filter design and simply transform it to the digital domain.

* Familiarization with IIR filter design procedure.
* Familiarization with frequency transformation.
* Bilinear frequency transformation
* Impulse invariance frequency transformation

**Lab Instructions**

* This lab activity comprises of three parts: Pre-lab, Lab Exercises, and Post-Lab Viva session.
* The lab report shall be uploaded on LMS.
* Only those tasks that completed during the allocated lab time will be credited to the students. Students are however encouraged to practice on their own in spare time for enhancing their skills.

**Lab Report Instructions**

All questions should be answered precisely to get maximum credit. Lab report must ensure following items:

* Lab objectives
* MATLAB/C codes
* Results (graphs/tables) duly commented and discussed
* Conclusion

# IIR Filter Design using Analog Prototypes

## Introduction

IIR filters can be designed either directly, by means of a clever placement of poles and zeros in the z-plane, or “indirectly” by first designing a corresponding analog filter, which is then transformed to the digital domain. Within this practical we will only consider the latter design procedure. Note that the described principle of using analog prototypes for IIR filter design originally stems from the idea to use digital filters as a substitute for analog systems.This allows us to use the well-constructed theory of analog filter design and simply transfer it to the digital domain.

## Pre lab:

### IIR filter design problem

Ideally, we are interested in a digital filter H(ejω) that simulates a continuous time system Hc(j), i.e. the following identity is fullfilled



Here ω denotes the dimensionless discrete-time frequency in radians per sample, and the analog frequency in radians per second. Td is a design parameter for the analog to digital conversion, and does not need to be the same as T = 1/fs, where fs is the sampling frequency.

### Design procedure

Given a digital filter specifications, the design of digital IIR filters can be summarized by a three-step procedure:

1. Translate the discrete-time filter specifications to the analog design domain,so that after the continuous to discrete-time transform is applied, they will be proper.

2. Select an analog prototype (Butterworth, Chebyshev, etc.) and design your analog filter Hc(s) using the translated specifications. Analog filter design it treated in next section .

3. Apply the continuous to discrete-time transform, and convert the analog transfer function Hc(s) into a discrete-time transfer function H(z). Details on the continuous to discrete-time conversion are given in Section 3. So before designing an appropriate analog filter, we have to translate the discrete-time filter specifications, which may be given either in real frequency f in Hz or normalized frequency ω = 2πf/fs in radians per sample, to analog design frequency . The translation depends on which continuous to discrete time conversion scheme is used. An overview for the frequency translation is depicted in Figure 1. The digital IIR filter design steps, described above can be done automatically in the MATLAB design programs. However, to understand the procedure we will go through the individual steps in one of the experiments.

### Analog filter design

We briefly describe methods for analog lowpass filter design and then show how to easily obtain highpass, bandpass or bandstop from analog lowpass prototypes by using frequency transformations.

(1) Bilinear transformation

(2) Impulse invariance method.



Figure 1: Translation of filter specifications (f or ω) to analog frequency Ω.

#### Analog lowpass filter design

There are four established techniques for the design of analog filters, including Butterworth, Chebyshev Type I and II, and Elliptic analog prototypes. They represent four different combinations of two error approximation measures. One is based on Taylor series, and results in a maximally flat frequency response. The other approximation minimizes the maximum difference between the desired and actual response, and results in an equiripple frequency response. The four lowpass prototypes are:

1. The analog Butterworth filter is based on a Taylor series with expansions at Ω = 0 and Ω= ∞, with desired values 1 and 0 respectively, and first up to (N-1)-th derivative equal to zero. This results in a smooth transition between at Ω = 0 and Ω= ∞. The squared magnitude of the normalized frequency response is given by



where N is the filter order and Ωc is 3 dB cutoff frequency. It can be seen that |Hc(jΩ)|2 = 0.5 at Ω= Ωc.

2. The analog Chebyshev Type I filter has a rippled passband with equally sized ripples and a smooth behavior in the stopband, as can be seen in Figure 2. The squared magnitude of the normalized frequency response



Figure 2: Magnitude specification for a Chebyshev Type I lowpass filter of a Chebyshev Type I filter is given by



where parameter ε determines the passband ripple amplitude, given by , and VN (x) is the Chebyshev polynomial of order N, defined recursively as



3. The analog Chebyshev Type II is the reverse counterpart of the Type I, i.e. it has a smooth passband and a rippled stopband. Here the design parameter specifies the stopband ripple amplitude.

4. The analog Elliptic filter uses a Chebyshev approximation in both the passband and the stopband. Two design parameters are used to specify the passband and stopband ripple amplitude. Corresponding MATLAB functions for analog prototype lowpass filter design are buttap, cheb1ap, cheb2ap and ellipap. They provide a zero pole representation, which can be easily converted to differential coefficient parameters by zp2tf.

#### Frequency transformation

Sometimes it is simpler to calculate filter coefficients of an analog lowpass prototype only once, or to use tables. This prototype is then converted by means of frequency transforms, according to the desired characteristics. In the following, let Hc(s) be an analog lowpass prototype of order N with normalized band edge Ωc = 1.

• For instance, a lowpass with an arbitrary band edge c can be obtained from the analog prototype Hc(s) by simply replacing

• Using this principle, we can also convert Hc(s) to a highpass filter with band edge Ωc,by changing this essentially replaces Ω = 0 and Ω= ∞.

• A bandpass filter with lower and upper band edge Ω1 and Ω2, can be obtained from the lowpass prototype Hc(s) by changing



Where BP is the geometric mean of the two band edges. This mapping doubles the order of the filter and is a one-to-one mapping of the frequency domain.In MATLAB you can use the functions lp2lp, lp2hp, lp2bp and lp2bs for the analog frequency transformations. For more detail we refer to the corresponding help files.

### Continuous to discrete-time conversion

Two methods for discrete-time conversion are considered: the impulse invariance and the bilinear transformation.

#### Impulse invariance method

The impulse invariance method is based on an intuitive concept. We directly transfer the discrete specifications into the analog domain, using Ω= ω/Td, and design a corresponding continuous-time filter Hc(j). We then determine the impulse response of Hc(jΩ), denoted hc(t) and obtain the impulse response of our discrete-time filter by sampling hc(t), i.e.



Here Td is the design sampling interval. Due to the fact that the FT of hc(nTd) is a periodic extension of Hc(jΩ), which is not perfectly bandlimited in practice, aliasing may occur so that (1) does not hold exactly. The continuous to discrete-time conversion by means of the impulse invariance can be done in MATLAB using the function impinvar.

#### The bilinear transformation

The bilinear transformation is a one-to-one mapping of the entire analog frequency domain −1 < < 1 onto the discrete-time frequency interval −π < ω < π. Using complex Laplace and Z-transform variables s and z, the mapping is given by



In the discrete-time domain we are interested in z variables on the unit circle whereas in the continuous-time domain, s variables on the imaginary axis s = jΩ are of importance.

By evaluating (3) at and s = jΩ, we obtain the relation between analog and digital frequency =



We notice that ω = 0 implies Ω= 0 and for increasing ω the continuous frequency also increases and we obtain . Hence, positive continuous-time frequencies are mapped to the upper half of the unit circle and negative frequencies to the lower half. Two important properties are:

• The bilinear transform preserves stability. A stable continuous-time system Hc(s) with poles on the left half of the s-plane, is mapped to a stable discrete-time system H(z) with poles inside the unit circle.

• Identity (1) holds exactly at Ω = 2/Td·tan(ω/2) when using the bilinear transform. Finally, we can perform the bilinear transformation and convert Hc(s) to H(z) using the relation



In MATLAB, the corresponding function bilinear does this job.

## Lab Tasks

### Bilinear transformation and impulse invariance

The N-th order analog Butterworth lowpass filter with cutoff frequency Ωc is given by 

1. Design a Butterworth filter with N = 4, ωc = 0.3π and Td = 2 using the impulse invariance method. Proceed as follows:

(a) Determine Ωc and use (5) to calculate the analog filtering coefficients

(b) In the analog domain, plot hc(t) and |Hc(jΩ)|. For the analog domain plots in MATLAB use the commands zp2tf, tf ,impulse and freqs.

(c) Transform your filter coefficients from the analog to the discrete time domain.

(d) In the discrete-time domain, plot h(n) and |H(ejω)| . Compare your results with (b).

2. Repeat 1. but with the bilinear transformation.

3 Further, compare both methods: what are their strengths and weaknesses?

### Lab Task 2:

#### Characteristics of IIR filters

For simplicity, we now use MATLAB functions which automatically perform all digital IIR filter design steps, mentioned in the introduction. The commands are e.g. butter, cheby1 or ellip. Use the help command for more details. Say the filtering specifications for a particular job are:

*Sampling frequency 200 Hz*

*Passband ripple \_ 1 dB*

*Passband edge 32 Hz*

*Stopband edge 38 Hz*

*Stopband attenuation > 25 dB*

1. Determine the order of a Butterworth, Chebyshev Type I and Elliptic filters that meet these specifications? Use the buttord, cheb1ord and ellipord commands.

2. Use butter, cheby1 and ellip to design the specified filters. For each, plot the magnitude of the frequency response, the pole-zero location diagram and the significant part of the impulse response.

3. For comparison, design a FIR filter using fir1 that meets the specifications and compare results.

|  |  |  |  |
| --- | --- | --- | --- |
| Window  Name | Transition Approximate | Width  Exact Values | Min. Stopband Attenuation |
| Rectangular |  |  | 21 dB |
| Bartlett |  |  | 25 dB |
| Hanning |  |  | 44 dB |
| Hamming |  |  | 53 dB |
| Blackman |  |  | 74 dB |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |