Analysis of FIR Notch Filters from second order IIR Prototypes with Numerical Instability

B. Henckell E. Reißner

July 1, 2024

Contents

List of Figures

Listings

List of Tables

Abstract

Although notch filters, e.g. used as net filters, are classically implemented as 2nd order IIR filters, there are reasons to use FIR filters instead, most important potential instability, ringing and nonlinear phase. Among the approaches for FIR notch filters collected in [RKJ01], there is one, originally suggested in [RJK97], to derive a FIR filter from an IIR filter. That way the derived FIR filters are comparable to their IIR filter counterpart. In particular, the FIR filters inherit angle and radius of the poles which determine frequency and tightness of the notch. For IIR filter, a tight notch comes at cost of stability and extensive ringing. Also, FIR filters add another parameter, its order.

In this paper we extend the analysis given by [RJK97] in that we include time domain response of a sine in notch frequency to analyze extinction and a synthetic ECG signal with notch frequency and peak reminding a unit impulse to consider ringing and degradation. Additionally, we include in our considerations radii of the pole close to 1, which correspond to tight notches, and find numerical instability in the coefficients of the FIR filter which is the price for stability of the FIR filter itself. In other words, instability is shifted from runtime to design.

1 Introduction

Potential instability, ringing and nonlinear phase of classical IIR notch filters, are among the motivations to consider FIR notch filters instead of IIR filters. This paper focuses on two approaches deriving a FIR filter from an according second order IIR filter, both presented in [RKJ01], Section 2.3. The derived FIR filter inherits good properties like low delay but also bad ones like nonlinear phase both close to the corresponding IIR.

At first sight, one gets rid of numerical instability and ringing. At second sight, numerical instability is just shifted from filter execution to filter design (where it may be acceptable). Still the design algorithm as such keeps simple, simpler at least as compared with other approaches as e.g. [RJK94], but they must be carried out with special arithmetic and is thus still quite time and memory consuming.

Also ringing is not really removed, but its length is bounded and tied to the order of the FIR filter.

For fixed notch frequency, the IIR filter essentially has a single parameter, the radius of the poles which must be less than 1. Depending on the field of application, the acceptable width of the notch is bounded and so the radius must be close to 1. This is exactly what causes numerical instability and ringing with little damping. So motivation to replace the IIR filter by a FIR filter is for the critical parameter values. To keep up quality, the FIR filter must approximate its according IIR filter well enough, which requires some minimum order and this in turn is the length of its "ringing". Besides the width of the notch, the field of application may restrict the length of ringing. For example in an ECG application the QRS complex ("heart beat") may cause ringing and this must at least be damped enough until the next beat comes. Since the segment immediately after the QRS complex is particularly significant for diagnosis, much quicker damping is vital. This may cause an unsolvable conflict for the derived FIR filter as much as for the original IIR filter: the radius must be smaller than one for quick damping of ringing but close to 1 to obtain a tight notch.

The idea to replace an IIR notch filter by a derived FIR filter has been introduced in [RJK97]. The approach advocated there is to decompose the filter into a FIR part which eliminates the notch frequency and into the IIR part which compensates outside the notch so that the complete IIR filter leaves frequencies apart from the notch frequency essentially unaltered. Then the IIR part is replaced by a FIR filter approximation, just computing the impulse response and cutting off somewhere. The composition of the two FIR filters, which is again a FIR filter, eliminates the notch frequency still completely, as does the original IIR filter, but compensation outside the notch is degraded so that it perturbs frequencies remote from the notch frequency. Worse than that, increasing the filter order cannot resolve that problem. This effect is the stronger, the tighter the notch is, i.e. when numerical instability begins to be an issue. Since [RJK97] presents punctual performance results in terms of the frequency domain with relatively broad notch, this effect is not observed in that paper.

Section 2.3 in [RKJ01] takes up the approach suggested in [RJK97], restates the results and adds a second approach, just taking the IIR filter as a whole, computing the impulse response and cutting off somewhere directly. Of course, the resulting FIR filter may not be expected to eliminate the original notch frequency completely, but this filter turns out to be a "weak" kind of notch filter in the sense that the amplitude at the notch frequency is reduced by a factor tied to the filter order. In contrast to the first approach, the filter preserves frequencies apart from the notch frequency quite well and fitting can be improved increasing filter order whereas there seems to remain ripple close to the notch frequency. The tighter the notch, the less damping on the notch frequency. So as for the first approach, the deficiencies of the filters are apparent in a parameter domain not observed in [RKJ01].

This paper discusses the two approaches in a parameter domain only touched by [RKJ01] and by [RJK97], but relevant in application, namely where the original IIR filter starts to be numerical instable and shows long ringing. In addition, for sake of comparability, we include parameters discussed in [RKJ01] and in [RJK97]. We restrict ourselves to a single notch angle, 0.2π which comes from field of application, filtering out 50Hz net frequency at sample frequency of 500Hz. For any IIR filter we find an order of the according FIR filter approximating the original IIR filter appropriately.

Whereas [RKJ01] and [RJK97] both analyze the filters in the frequency domain only, we also consider the time domain and apply the filters to a synthetic calibration ECG referred to as `CAL20500` in [EN615], 201.12.4.107.1.2, and to a sine with notch frequency.

The ECG signal `CAL20500` which is depicted in Figure 1 shows that `CAL20500` has a significant peak representing a heartbeat. This makes it easy to determine the delay the filter causes when applied to `CAL20500`. Moreover, `CAL20500` has frequency components around 50Hz and is thus degraded by a 50Hz notch filter and causes ringing. Passing a sine to the filter is like passing a zero signal followed by a sine which is to show damping of the notch frequency as a dynamical process.

2 The Approach: FIR from IIR Notch Filter

In Subsection 2.1 we present the original IIR filter to start with. It eliminates the notch frequency and the notch can be made arbitrarily tight, choosing the radius of the poles appropriately. From that in Subsection 2.3 a FIR notch filter is deduced eliminating the notch frequency also but not strictly approximating the original IIR filter. In contrast, Subsection 2.2 presents a FIR filter just approximating the underlying IIR filter but not strictly eliminating the notch frequency.

2.1 The starting point: an IIR notch filter

The original IIR filter is given by [LRO94] and has transfer function

$$H(z) = g \cdot \frac{1 - 2\cos(\omega)z^{-1} + z^{-2}}{1 - 2r\cos(\omega)z^{-1} + r^2 z^{-2}} \quad \text{for } r \in (0,1), \omega \in [0,\pi]. \tag{1}$$

There are more ways to design an IIR notch filter, but this one is quite natural: First, to make the filter eliminate frequency ω, the transfer function must have zero $z_n = e^{i\omega}$. We consider more general $z_n = re^{i\omega}$ although only $r = 1$ is relevant. To obtain real coefficients, also $\overline{z_n}$ must be a zero. So the numerator must have at least a factor

$$(z - z_n)(z - \overline{z_n}) = z^2 - (z_n + \overline{z_n})z + z_n\overline{z_n} = z^2 - 2r\cos(\omega)z + r^2 \tag{2}$$

For $r = 1$ this is up to division by z^{-2} what is given in the numerator of Equation (1). The transfer function must have conjugate poles z_p close to z_n but inside the unit circle to make the filter stable. If z_p is at fixed distance from z_n, the best stability is for least absolute value $|z_p|$ and this is realized by using the same angle. So the approach is $z_p = re^{\omega}$ with radius $r \in [0,1)$ but close to 1. This yields analogous to the numerator of the transfer function the denominator $(z - z_p)(z - \overline{z_p}) = z^2 - 2\cos(\omega)rz + r^2$. Now we can see that numerator and denominator are just as in Equation (1), just factored out z^{-2}.

Finally, the gain g is no free parameter but must be chosen so that $1 = H(z)$ for $z = e^0$ and so

$$g = \frac{1 - 2r\cos(\omega) + r^2}{1 - 2\cos(\omega) + 1}.$$

If we fix ω, the only parameter is $r \in [0, 1)$ which shall be close to 1.

Note that [Orf96], Section 11.3 presents a slightly different kind of 2nd order IIR digital notch filter derived from an analog filter. The advantage there is that the radius can be computed from the $-3db$ bandwidth of the notch at the expense of z_p not being optimally stable. More variants and a special variant considered optimal also are given in [TP01]. All these approaches can be used as a starting point for the kind of FIR filter design presented here.

2.2 A FIR filter approximating an IIR notch filter

Transfer functions $H(z)$ of IIR filters are rational functions, whereas those of FIR filters are polynomials. Writing $H(z)$ as a product

$$\begin{aligned} H(z) &= gA(z)B(z) \quad \text{with} && (3)\\ A(z) &= 1 - 2\cos(\omega)z^{-1} + z^{-2} \quad \text{and} \\ B(z) &= \frac{1}{1 - 2r\cos(\omega)z^{-1} + r^2 z^{-2}} = \frac{1}{1 - az^{-1} + bz^{-2}}, \end{aligned}$$

the latter equation for $B(z)$ using the abbreviations

$$a = 2r\cos(\omega) \text{ and } b = r^2,$$

we can interpret the original filter as the composition of a filter with transfer function $A(z)$ and another one with transfer function $B(z)$. Since $A(z)$ is a polynomial in z^{-1}, the according filter is a FIR filter already. As a first step towards replacing $B(z)$ by a polynomial in z^{-1} as well, rewriting $B(z)$ as a Taylor series in z^{-1} yields

$$B(z) = \sum_{i=0}^{\infty} d_i z^{-i} \quad \text{with} \tag{4}$$

$$d_i = \begin{cases} \sum_{j=0}^{\lfloor \frac{i}{2} \rfloor} (-1)^j \binom{i-j}{j} a^{i-2j} b^j & \text{for } i \in \mathbb{N}_0 \\ 0 & \text{for } i = -1, -2 \end{cases}. \tag{5}$$

Note that the latter branch is required not until the following steps.
Next we write the whole transfer function $H(z)$ as Taylor series.

$$\begin{aligned} H(z) &= gA(z)B(z) && (6)\\ &= g\left(1 - 2\cos(\omega)z^{-1} + z^{-2}\right) \sum_{i=0}^{\infty} d_i z^{-i} \\ &= g \sum_{i=0}^{\infty} \left(d_i z^{-i} - 2\cos(\omega) d_i z^{-(i+1)} + d_i z^{-(i+2)} \right) \\ &\overset{d_{-1}, d_{-2}=0}{=} g \sum_{i=0}^{\infty} \left(d_i z^{-i} - 2\cos(\omega) d_{i-1} z^{-i} + d_{i-2} z^{-i} \right) \\ &= g \sum_{i=0}^{\infty} (d_i - 2\cos(\omega) d_{i-1} + d_{i-2}) z^{-i} \\ &= g \sum_{i=0}^{\infty} D_i z^{-i} \quad \text{with} \\ D_i &= d_i - 2d_{i-1}\cos(\omega) + d_{i-2} \quad \text{for } i \in \mathbb{N}_0. && (7) \end{aligned}$$

Cutting off that series after the Nth summand yields the transition function of an Nth order FIR filter

$$H_N(z) = g \sum_{i=0}^{N} D_i z^{-i}. \tag{8}$$

approximating the original IIR filter with transition function $H(z)$. On the one hand, this sounds good, but on the other hand, there is no reason to assume that this filter is still a notch filter eliminating the notch frequency. This motivates us to try another approach in Section 2.3.

The D_i's can be rewritten in terms of binomial coefficients in the style as the d_i's in Equation 5, which may make a difference numerically. Unlike [RKJ01], Section 2.3.3, we refrain from doing so for sake of a simple unifying algorithm for the variants of this section and the following one.

2.3 A FIR notch filter eliminating the notch frequency

Again we start our considerations with the transfer function of the IIR filter given in the form $H(z) = gA(z)B(z)$ in (3) and observe that the IIR filter eliminates the notch frequency because of an according zero in $A(z)$. To obtain a FIR filter which, unlike in the approach in Section 2.2, still eliminates the notch frequency, we preserve the product form and approximate $B(z)$ only, just cutting off the series expansion given in (4). Since $A(z)$ has order 2, to obtain a FIR filter with order N, we have to cut off $B(z)$ after $M = N - 2$ summands which yields

$$B_{N-2}(z) = \sum_{i=0}^{N-2} d_i z^{-i}. \tag{9}$$

With

$$d_{i,N} = \begin{cases} d_i & i \in \{0, \ldots, N-2\} \\ 0 & \text{otherwise} \end{cases} \quad \text{for } i \in \mathbb{Z}, m \in \mathbb{N}_0, \tag{10}$$

we can write

$$B_{N-2}(z) = \sum_{i=0}^{N} d_{i,N} z^{-i}. \tag{11}$$

Then replacing $B(z)$ in (3) by $B_{N-2}(z)$ yields a FIR filter with order N eliminating the notch frequency with transfer function

$$
\begin{aligned}
H^{\circ}_N(z) &= gA(z)B_{N-2}(z) \qquad (12)\\
&= g\,(1-2\cos(\omega)z^{-1}+z^{-2})\sum_{i=0}^{N} d_{i,N}z^{-i}\\
&= g\sum_{i=0}^{N}\left(d_{i,N}z^{-i}-2\cos(\omega)d_{i,N}z^{-(i+1)}+d_{i,N}z^{-(i+2)}\right)\\
&= g\sum_{i=0}^{N}(d_{i,N}z^{-i}-2\cos(\omega)d_{i-1,N}z^{-i}+d_{i-2,N}z^{-i})\\
&= g\sum_{i=0}^{N}(d_{i,N}-2\cos(\omega)d_{i-1,N}+d_{i-2,N})z^{-i}\\
&= g\sum_{i=0}^{N}D_{i,N}z^{-i} \quad \text{with}\\
D_{i,N} &= d_{i,N}-2d_{i-1,N}\cos(\omega)+d_{i-2,N}. \qquad (13)
\end{aligned}
$$

Note that in [RJK97] computes B_N with ordering N ending up with a FIR filter with ordering $N+2$ whereas here, the index N in $H_N(z)$ refers to the order of the resulting filter, in view of uniformity with the second approach we present in Subsection 2.2.

The transfer function $H^{\circ}(z) = gA(z)B_M(z)$ of that FIR filter is the product of the functions $gA(x)$ and $B_M(z)$ both being polynomials in z^{-1}. Correspondingly, it is the composition of the FIR filters with transfer functions $gA(z)$ and $B_M(z)$. The first filter eliminates the notch frequency and the second one approximates a filter inverting the first filter outside the notch frequency.

2.4 Implementation of both types of FIR filters

In this section we present an implementation of the two kinds of FIR filters derived from an according IIR filter in Sections 2.3 and 2.2 in the MATLAB language.

Comparing the formulae for the two approaches (12) and (8), we find that just the coefficients D_i given by (13) are replaced by the $D_{i,N}$'s given in (7). Comparing the formulae shows, that the d_i's given in (10) are replaced by their modifications $d_{i,N}$ given by (5). This allows a quite unified implementation of the two types of filters under consideration.

Both variants are implemented in a single MATLAB function `firNotchFromIir` given in Listing 1.

Let us discuss the parameters:

`angle` is the notch angle. It is also the angle of the zero and of the pole.

`rPole` is the radius of the pole of the transfer function, whereas the radius of its zero is 1.

`alt` determines whether the FIR filter is designed as described in Section 2.3; else it is as designed in Section 2.2.

`DegN` determines the degree of the FIR filter approximating the original IIR filter.

The return value `dMaxArr` holds the coefficients of the FIR filter, which is equivalent to the impulse response. The impulse response comes in the ordering given by `dMaxArr` but, as usual in the MATLAB language, indices start with 1, not with 0 as in theory. The length of `dMaxArr` is the order of the FIR filter.

Note that in this paper we only consider angles 0.2π. This is motivated by the fact that in our application the sample rate is 500Hz, and we evaluated the FIR filters for singling out net frequency of 50Hz which maps to $\frac{50}{500}2\pi = 0.2\pi$. Nevertheless, the MATLAB function is given with parameter `angle`.

3 Filter Coefficients, Impulse Responses and Numerical Instability

Table 1 shows the filter coefficients of FIR filters with order 50 approximating an IIR notch filter with pole radii $r = 0.85$ and 0.992. The computations are executed with various kinds of arithmetic, namely single precision and double precision arithmetic, both implemented in java, `octave`'s double precision arithmetic and finally adaptive l2r-arithmetic with verified results. The filter coefficients of the FIR filters are at the same time the nontrivial section of the impulse responses of FIR filters. By construction, we would expect that they coincide with the impulse responses of the original IIR filters which are given also.

For $r = 0.85$ we obtain the FIR filter we expect, except for single precision arithmetic. The impulse response shows some damped ripple. The results do not depend on the arithmetic and the response for FIR filter and for according IIR filter coincide. The IIR filter behaves as expected, even if computed with single precision. Solely the FIR filter shows artifacts at the end of the impulse response.

This behavior changes dramatically for $r = 0.992$. Whereas the IIR filter shows an impulse response as expected even for the worst arithmetic, the FIR filter shows those artifacts which are for $r = 0.85$ only observed for single precision. Solely the l2r-arithmetic, which performs computations with adaptive precision, shows good results, indistinguishable from the results, the IIR filter shows for any arithmetic.

Besides artifacts, Table 1 also shows reality: increasing r leads to a lower ripple but with less damping. In parallel, numerical stability decreases. This effect becomes even more apparent, if we need higher order filters, which is true in the sequel (starting with Section 4.2).

The left-hand side of Table 2 shows the filter coefficients of FIR filters with order 50 derived from an IIR notch filter with radii $r = 0.85$ and 0.992 and eliminating the same notch frequency. The computations are executed with various kinds of arithmetic. The filter coefficients of the FIR filters are at the same time the nontrivial section of the impulse responses of FIR filters. As this kind of FIR filter does not approximate the IIR filter it is derived from, we draw no comparison with the IIR filter. Instead, we compare with the FIR filters of Table 1 for order 50.

For $r = 0.85$ we obtain what we expect, except for single precision arithmetic. The impulse response shows some damped ripple and everything looks very much like the FIR filters approximating its IIR filter. The results don't depend on the arithmetic and the response for FIR filter. Solely for single precision arithmetic the impulse response shows artifacts at the end of the impulse response, whereas the peak at the beginning seems to be missing.

This behavior changes somehow but not dramatically for $r = 0.992$. Compared with the l2r arithmetic which is trustable, both, java double arithmetic and native octave arithmetic show some artifacts, although not the same at the end of the signal. Be careful to distinguish: not the negative peak at the end of the impulse response is the artifact, it is the oscillation shortly before. That even the l2r-arithmetic shows a negative peak at the end of the impulse response, shows that the peak is an undesirable property of the filter and not an artifact. For ECG applications this leads to a negative echo of the QRS complex which is highly undesirable.

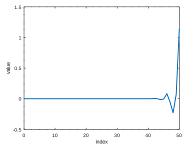
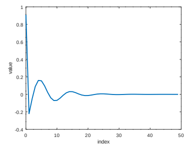
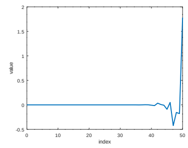
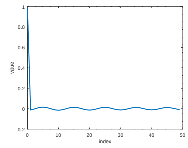
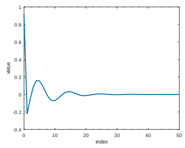
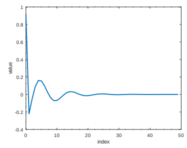
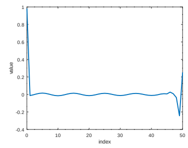
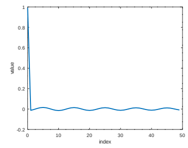
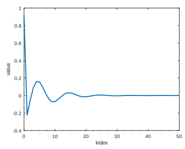
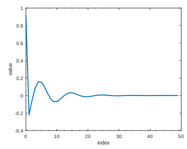
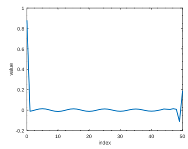
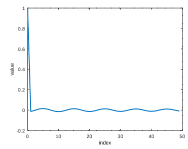
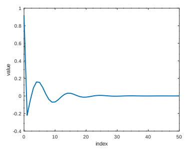
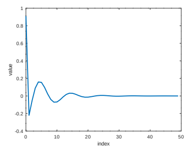
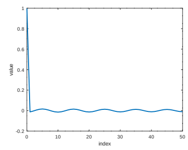
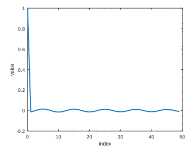
Arithmetic	FIR $r = 0.85$	IIR $r = 0.85$	FIR $r = 0.992$	IIR $r = 0.992$
single				
double				
octave				
l2r				

Table 1: Impulse responses of FIR filters, approximating its IIR notch filter with various kinds of arithmetic, same parameters: pole radii 0.85 and 0.992; filter order 50

For filter order 50 the eliminating notch filters seem to be numerically more stable than the approximating notch filters. Section 7 states that for pole radius close to one as e.g. $r = 0.992$, the filter order must be high and order 110 is still quite low to lead to a usable filter The right-hand side of Table 2 shows the results of the various kinds of arithmetic for that ordering. Remember that only l2r arithmetic is trustable. All the other kinds of arithmetic neither show the ringing at the beginning of the signal nor the subtle waves in the middle part, but instead show dominant artifact preceding the negative echo which was present only rudimentary for filter order 50.

Although the two kinds of double arithmetic are not trustable in general, they at least show the negative echo for pole radius $r = 0.85$ for filter order 50.

Part of an explanation for the numerical instability is the occurrence of high order binomial coefficients in Equation (5) on page 5. In MATLAB language, binomial coefficients are treated as double floating point numbers. They are as if computed with full precision and then rounded to the nearest double.

We observe that for pole close to (but inside) the unit circle, the summands for computation of the filter coefficients are essentially the binomial coefficients and as a result of this, vary in a broad range of magnitude. For small radius in contrast, powers of certain constants equalize the summands somehow.

Note that numerical instability may be worsened, because for poles close to the unit circle higher filter orders are required, but instability increases with the radius, even if the filter order is kept constant.

If using adaptive precision arithmetic but still `matlab`'s implementation of binomial coefficients

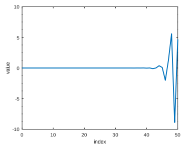
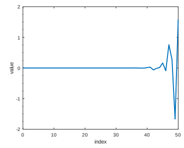
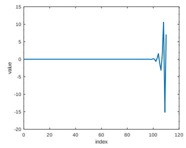
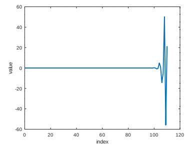
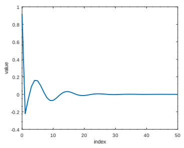
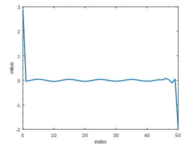
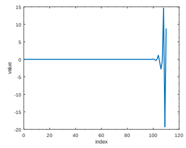
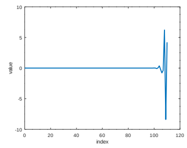
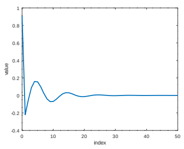
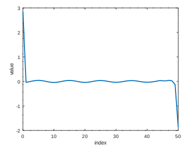
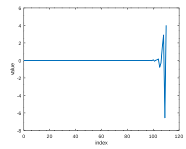
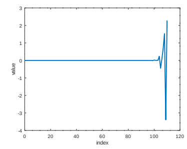
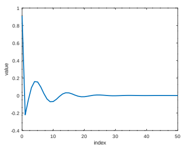
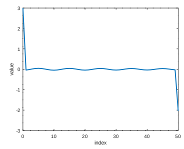
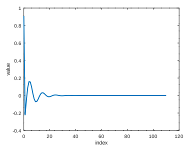
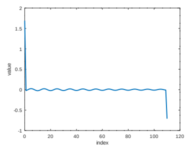
Arithmetic	FIR $r = 0.85$	FIR $r = 0.992$	FIR $r = 0.85$	FIR $r = 0.992$
	order 50		order 110	
single				
double				
octave				
l2r				

Table 2: Impulse responses of FIR filters, eliminating the notch frequency with various kinds of arithmetic, same parameters: pole radii 0.85 and 0.992; filter orders 50 and 110

yields much different results but not with fewer artifacts but with different ones. This leads us to conjecture, that the binomial coefficients play a role in instability, but not alone.

Here, we could also insert a table with dependency of impulse response/transfer function of FIR filters on the order, if computed with traditional double arithmetic. We refrain from doing so because this dependency disappears if computations are carried out with l2r-arithmetic.

For $r = 0.9987$, one can see already considerable differences between java double and octave double. This is another indication of numerical instability.

The main conclusion from this section is, that the proper analysis shall be based solely on l2r-arithmetic; else it may not be significant.

4 The Transfer Functions

The transfer functions for the IIR notch filters are quite simple: They have a zero at the notch frequency, are close to one apart from the notch. The larger the radius $r < 1$ of the pole, the tighter is the notch and the tighter the graph outside the notch to the value one, i.e. the better the filter from the point of view of frequency domain.

For the derived FIR filters, this is much more involved. First, in Section 4.1, we show that the numerical instability shown in Section 3 for the impulse response, more precisely in Tables 1 and 2, are inherited by the according transfer functions. Thus, the built-in types of arithmetic of octave/MATLAB are inappropriate. Then we analyze the dependency of the transfer function on pole radius and filter order based on l2r-arithmetic in Section 4.2.

4.1 Dependency of the transfer function on arithmetic

Note that in this section the transfer functions are represented in linear scaling to show more details. That only l2r arithmetic is appropriate finally is also clearly observable in the usual logarithmic scaling.

For FIR filters approximating the original IIR notch filter, take a view on Table 3 showing the transfer functions for pole radii 0.85 and 0.992, the FIR filters with order 50. The transfer function of the IIR filters are insensitive towards arithmetic as are the impulse responses given in Table 1. Since the pictures are indistinguishable, they are given for `octave` arithmetic only. So any kind of arithmetic is appropriate although only l2r-arithmetic adapts precision and is thus reliable.

For FIR filters however, sensitivity towards the arithmetic type depends on the parameter: Whereas for pole radius $r = 0.85$ the transfer function computed with single precision is a pure artifact which has absolute no similarity with the true solution and does not even allow to locate the notch, the other types of arithmetic yield similar and correct results.

In contrast, for pole radius $r = 0.992$, only l2r-arithmetic shows the correct location and shape of the notch and also the shape of the ripple which is no artifact and which is larger close to the notch. The two types of double precision arithmetic overestimate the ripple and underestimate the depth of the notch so much, that the notch appears like a perturbation. Nevertheless, its location seems quite correct. The fact that java double arithmetic and `octave` double arithmetic are almost identical, whereas its results differ considerably, indicates the sensitivity on the kind of arithmetic, maybe on details in the implementations of functions.

Comparing the transfer functions of FIR filters computed with l2r-arithmetic for the given pole radii shows, that for $r = 0.85$ the transfer function of the FIR filter qualitatively has the same shape as that of the according IIR filter, whereas for $r = 0.992$, significant ripple appears, in particular close to the notch, which does not occur for the according IIR filter. Partially this leads to amplification and partially to damping of the frequency.

For FIR filters eliminating the notch frequency, take a view on Table 4, and compare with Table 2. Let us discuss filter order 50 first. For pole radius $r = 0.85$, all kinds of arithmetic indicate the notch, but single precision arithmetic shows amplification on one side of the notch, which is an artifact. All other kinds of arithmetic yield a quite correct transfer function for the FIR filter and this transfer function is close to the transfer function of the according IIR filter.

Single precision arithmetic is also inappropriate for $r = 0.992$, whereas the other kinds of arithmetic are close to the correct results of l2r-arithmetic all showing the ripple specific for larger pole radii like $r = 0.992$. A property visible only for l2r-arithmetic is, that only in discrete equally distributed points the amplitudes are preserved.

This changes for FIR filter with higher order like 110. For both pole radii still all kinds of arithmetic show the correct location of the notch. Even left-hand side of the notch the transfer function is computed quite correctly, but the right-hand side is only correct in l2r arithmetic. As for lower order 50 also for order 110 ripple occurs only for $r = 0.992$, but unlike for order 50 for order 110 the ripple is detected only by l2r arithmetic.

4.2 Dependency of the transfer function on parameters

Now that we understood that significant results require appropriate kind of arithmetic, we shall confine ourselves to l2r-arithmetic. The problem is, that at time of this writing, l2r-arithmetic

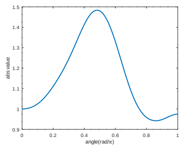
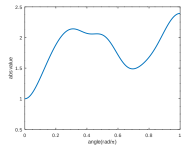
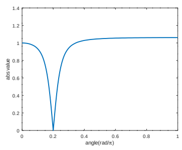
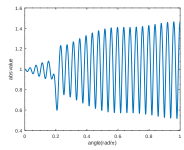
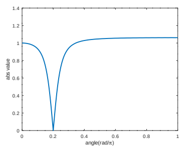
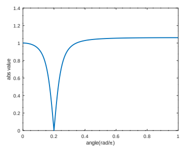
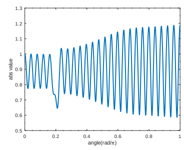
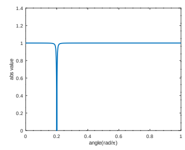
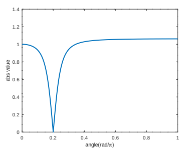
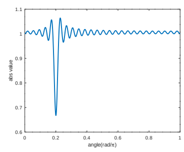
Arithmetic	FIR $r = 0.85$	IIR $r = 0.85$	FIR $r = 0.992$	IIR $r = 0.992$
single				
double				
octave				
l2r				

Table 3: Transfer functions of FIR filters, approximating its IIR notch filter with various kinds of arithmetic, same parameters: pole radii 0.85 and 0.992; filter orders 50 and 110, represented in **linear y scale**

does not support complex numbers. Thus, we make the compromise to compute impulse responses, which seem the critical part, in l2r-arithmetic, whereas transformation of this into transfer function is done in traditional double arithmetic. In the long run, this shall change.

In the present paper, the notch frequency $\frac{\pi}{5}$ is special in that it is a rational fraction of π with small denominator. Thus, the notch fits easily into the equidistant ripple of the FIR filters. This may make the notch frequency a special case and the following observations may require generalization for arbitrary notch frequency.

The last line of Table 5 shows the transfer functions for IIR notch filters with notch frequency 0.2π for various pole radii. One can see, that the transfer functions have a notch at 0.2π and are quite constant at 0db apart from the notch frequency. The more the radius approaches 1, the tighter the notch and the tighter the transfer function at 0db apart from the notch. Note that independent of the pole radius, the IIR filters eliminate the notch frequency completely, i.e. the notch is “infinitely deep”.

The rest of Table 5 shows the transfer functions for FIR filters approximating the given IIR filters, the better, the higher the filter order, i.e. the higher the row index in the table. That is why the transfer functions of the IIR filters are placed in the last line.

Unlike the IIR filters, the FIR filters show some ripple, in particular for small filter order. This is true already for relatively small pole radius if the filter order is small also, but the larger the pole radius, the more apparent the ripple. The ripple looks like a sine centered around 0db, damped with distance to the notch frequency. The 0db are reached at angle of 0 and the 0db is

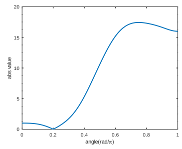
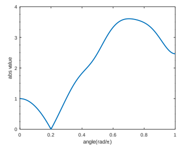
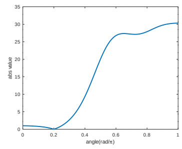
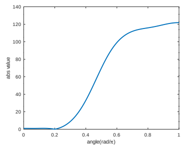
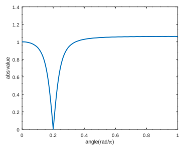
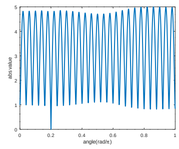
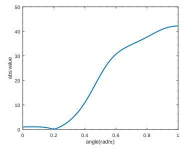
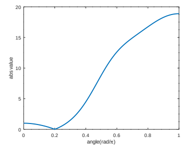
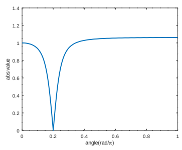
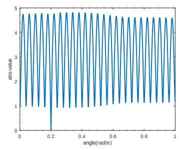
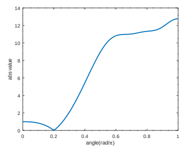
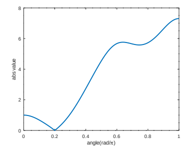
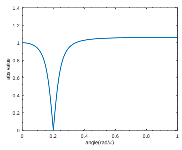
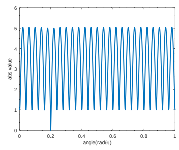
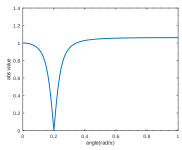
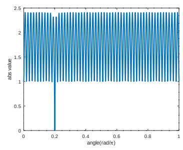
Arithmetic	FIR $r = 0.85$	FIR $r = 0.992$	FIR $r = 0.85$	FIR $r = 0.992$
	order 50		order 110	
single				
double				
octave				
l2r				

Table 4: Transfer functions of FIR filters, eliminating the notch frequency with various kinds of arithmetic, same parameters: pole radii 0.85 and 0.992; filter order 50 and 110, represented in **linear y scale**

approached at angle π for large pole radius. Effectively, the notch itself looks like part of the ripple.

With filter order, also ripple frequency increases which fits the observation that the notch becomes tighter with increasing filter order. Presumably, in addition, its tightness is bound by the tightness of notch of the according IIR filter. In the examples, the notch of the IIR filter is tighter than of any derived FIR filter.

Unlike for the IIR filter, for the derived FIR filters the depth of the notch is finite. Details on this are deferred until Section 5. On the other hand, increasing the filter order the notch can be made as deep as required but as a consequence, also the amplitude of the ripple around the notch increases.

Since at the same time, the amplitude of the ripple decreases with the filter order apart from the ripple, the ripple becomes more accentuated around the notch frequency with increasing filter order. Whether the ripple converges to zero even close to the notch frequency, is difficult to say. In Table 7 the same transfer functions are depicted, this time with linear scaling which makes it easier to judge whether it makes sense to conjecture convergence. One can see that the ripple becomes visible for smaller pole radius already. Convergence seems plausible although it is not monotone. Final judgement would require a mathematical proof. Observe that even in linear scaling IIR filters show absolutely no ripple.

The last line of Table 6 shows the transfer functions for IIR notch filters for the same notch frequency and pole radii as in the last line of Table 6.

The rest of Table 6 shows the transfer functions for FIR filters derived from the given IIR filters and thus eliminating the same notch frequency. Strictly speaking, there is no indication that they are approximating their according IIR filter but nevertheless their transfer functions compare in a way we make more explicit below. Since the transfer functions of FIR filters approach those of the according IIR filters with increasing order, i.e. with increasing row index of the table, it is natural to place the IIR filters in the last line.

Unlike the IIR filters, the FIR filters show some ripple, in particular for small filter order. This is true already for relatively small pole radius if the filter order is small also, but the larger the pole radius, the more apparent the ripple. The ripple looks like a wave with local minima at 0db, except one minimum which is exactly at the notch frequency in our case and which is at $-\infty$. In that sense, the notch itself is integrated in the ripple. The 0db are reached at angle of 0 and the 0db is slowly approached at angle π for large pole radius. The local maxima seem all at the same level and above the 0db.

With filter order, also ripple frequency increases which fits the observation that the notch becomes tighter with increasing filter order. The higher the pole radius the more the ripple assumes the shape of a lobe, with flat maxima and peak minima, one of them at the notch frequency. Also, with higher pole radius, the ripple becomes larger which allows the notch to be tighter. Presumably, the tightness of the notch is bound by the tightness of the according IIR filter.

Unlike the IIR filter, the derived FIR filters show considerable ripple uniformly distributed over the whole frequency domain. On the other hand, increasing the filter order the amplitude of the ripple seems to decrease very slowly, but it is not clear whether it converges to 0 point-wise or uniformly. Since at the same time, the ripple frequency increases linearly with the filter order, the ripple becomes more tense all over the frequency domain with increasing filter order.

In Table 8 the same transfer functions are depicted, this time with linear scaling which makes it easier to judge whether it makes sense to conjecture convergence of the amplitude of the ripple. One can see that the ripple becomes visible for smaller pole radius already. Convergence seems very likely, and it seems to be monotone even. Final judgement would require a mathematical proof.

In both Tables, 5 and 6 one can see that, whereas IIR filters evolve smoothly when increasing the pole radius, for the according FIR filter this is not really the case: For low pole radii, the FIR filters resemble the according IIR filter even for small filter order, and as a consequence there is not much difference between the two kinds of FIR filters (approximating its IIR filter or eliminating its notch frequency). In contrast, for larger pole radii, for the two kinds of FIR filter specific features become apparent which differentiate them from one another and also from the according IIR filters. These additional properties are disadvantageous from the point of view of application. The FIR filters approximating their IIR filters are superior to the FIR filters eliminating the notch frequency.

5 Applying the Filters to a Sine-Step Signal with Notch Frequency

In this section, we apply the filters under consideration to a sine signal with unit amplitude and notch frequency 0.2π. As for all linear and time invariant filters, the sine is converted in another sine with the same frequency. Only the amplitude and the phase may change. For the ideal notch filter the amplitude is zero and the phase is undefined.

To be more precise, the signal we use is a finite section of a sine signal. As our signal must "start somewhere" and is not infinite in backward time, it is in fact a signal with two sections: for negative indices it is constant zero, else it is a sine. Although it is clear, at least for the IIR filters and for one kind of FIR filters under consideration, that this frequency is eliminated, it is interesting to illustrate the dynamical aspect of elimination: Applying the filter to a signal which is zero initially and then resembles a sine, yields a signal which is zero initially, then a sine with initial amplitude of the input sine, and then it is damped in some way depending on the kind of filter.

As one can see from the last line in Table 9, IIR filters do not just eliminate a sine instantly switched on; rather than it is damped with damping factor given by the pole radius. This illustrates why the radius must be less than 1. Note that for exact computations, the output never really disappears; it eventually becomes negligible and only rounding in a computer makes it vanish.

For the FIR filters approximating the IIR filters one can see the same damping rate as for the initial IIR filter, but damping is performed only for as many samples as given by the filter order. After that, the output of the filter is a sine with amplitude kept constant.

This explains why FIR filters approximating the IIR filters don't eliminate the notch frequency completely but reduce its amplitude, the more, the higher the filter order. It also shows that for small pole radius, i.e. quick damping, low filter order suffices to reach a given reduction, whereas for pole radius close to 1 higher filter order are required.

Table 10 shows the effect of a FIR filter eliminating the sine completely. As for the FIR filters approximating the according IIR filter, the sine is just damped with the same damping rate as for the initial IIR filter. As for approximating FIR filters, damping is performed only for as many samples as given by the filter order. After that, the output is abruptly and precisely zero. This holds independently of the filter order. This is better than the behavior of the FIR filter approximating the IIR filter which just leaves the amplitude unchanged, and it is even better than the behavior of the original IIF filter which just goes on damping. On the other hand, this beautiful behavior comes at the expense of the negative echo described in Sections 3 and 6.

For filtering a sine, we come to conclusions similar to those of the sections above:

For low pole radii, both kinds of FIR filters resemble the according IIR filter even for small filter order because of strong damping the output signal becomes small within the number of samples given by the lowest filter order of FIR filter we consider. Thus, the overall damping by FIR filter is quite the same as for IIR filter. As a consequence, there is not much difference between the two kinds of FIR filters.

In contrast, for pole radius approaching 1, damping is low so that at the end of the filter response of the FIR filters, a quite big portion of the initial amplitude still persists. Thus, one can see the difference between further damping performed by the IIR filter, abrupt elimination done by the eliminating FIR filter and leaving the amplitude reached so far forever as in the FIR filter approximating the IIR filter. The three kinds of filters become similar again for FIR filters with high filter order.

6 Applying the Filters to a synthetic ECG Signal

Figure 1 shows the shape of the synthetic calibration signal `CAL20500` referred to at the norm [EN615], 201.12.4.107.1.2 for electrocardiographs. The original signal is defined by the file `CAL20500.CYC` in the CTS[1] database. According to documentation and metadata in `CAL20500.CYC`, the sample values are understood in μV and its length is 1000ms. Since it has 1000 samples, the sample rate is 1000Hz. For Figure 1, this signal is downsampled to 500Hz. It

[1] Clinical Trial Subject

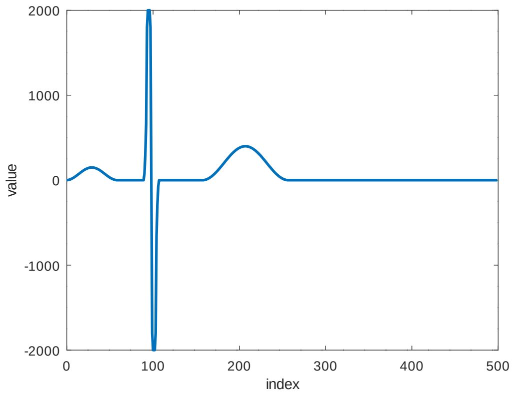


Figure 1: The synthetic signal `CAL20500`

must be interpreted as the period of a periodic signal and represents a single human heartbeat with heartbeat frequency of 1Hz. One can see the P wave, the QRS complex and the T wave; the rest of the signal is zero.

Table 11 shows the output of IIR filters and FIR filters approximating them, both with various parameters applied to `CAL20500`.

For the IIR filters we observe that the P wave is preserved, whereas the QRS complex introduces ringing. Also, the T wave is preserved as such but for pole radius near 1, the ringing of the QRS complex is modulated on the T wave. In other words, we see the superposition of the T wave and ringing from the QRS complex.

The explanation is, that the QRS complex of `CAL20500` has considerable 50Hz components, whereas the P and T waves have not.

For low pole radius, the initial amplitude of ringing is high because the notch is broad and so frequencies in a broad interval around the notch frequency contribute. On the other hand, low pole radius quickly damp these frequencies. In contrast, for pole radius approaching 1, ringing is with low amplitude, but it is also slowly damped. As a consequence, it is not completely extinct after the whole complex and may influence the next heart beat.

The FIR filters approximating the IIR notch filters show the same ringing as the IIR filters they approximate, both in initial amplitude and in damping. The only difference is that after the number of samples given by the order of the FIR filter, ringing ends abruptly, which is what we expect. After the ringing, the output signal is indistinguishable from the input signal.

For FIR filters derived from IIR filters and eliminating the notch frequency, Table 12 shows the according output of the filters. Those FIR filters show ringing with initial amplitude and damping like the according IIR filters, except that ringing ends abruptly after the number of samples given by the filter order as for the approximating FIR filters considered above. Nevertheless, there are two differences: Ringing does not end just, it is ended after a kind of “negative echo” already observed for the impulse responses in Section 3. This is most apparent for the QRS complex but occurs for the T wave also and even for the P wave. The delay of the echo is the filter order. The relative amplitude of the echo increases with pole radius and approaches one for pole radius converging to one from below. The amplitude of the echo converges to zero with the filter order.

The explanation is quite simple: The relative height of the echo reflects how much of the amplitude of the notch frequency is left after a number of samples given by the order of the FIR filter, because after the last sample the notch frequency is eliminated completely As shown in

Section 5, this is the more, the less damping, i.e. closer the pole radius to one and the lower the filter order is.

Viewing both, Table 11 and Table 12 in common, we come to a conclusion analogous to that in Section 4: For low pole radii, the FIR filters resemble the according IIR filter even for small filter order, As a consequence, there is not much difference between the two kinds of FIR filters. In contrast, for larger pole radii, for the two kinds of FIR filter specific features become apparent which differentiate them from one another and also from the according IIR filters. These additional properties are disadvantageous from the point of view of application. Again, the FIR filters approximating their IIR filters are superior to the FIR filters eliminating the notch frequency.

7 Conclusion, Deficiencies and Prospects

In [RKJ01], Section 2.3 two approaches are suggested to derive FIR filters from an IIR notch filter. We extended the analysis given there in that we approach the boundaries of the parameter range, both in pole radii and in filter order. Moreover, we take additional aspects into account: In the frequency domain we give representations not only in db, but also with linear scaling. In time domain we analyze filtering of a synthetic ECG-signal and of a sine in notch frequency. Last not least, we demonstrate the presence of numerical instability and solve that problem using an arithmetic with dynamical precision.

Extending the parameter range under consideration we obtain more differentiation between the original IIR filter, the derived FIR filter approximating the IIR filter and the derived FIR filter eliminating the notch frequency. The differences comprise the transfer function, damping of a sine with notch frequency and filtering an artificial ECG signal.

Whereas for low pole radius, i.e. wide notch, both kinds of FIR filters are a good replacement for the according IIR filter, and are almost not distinguishable, for pole radius approaching 1, i.e. if tight notch is requested, it becomes apparent that the approximating FIR filter does not eliminate the notch frequency completely, but to a fraction converging to zero in its filter order. Accordingly, for eliminating FIR filters the echo becomes significant which converges to zero again in the filter order. So for different reasons the two kinds of FIR filters need high filter order which make them similar to the original IIR filter and also diminish their advantage compared to the IIR filter. In fact, both FIR filters are computationally expensive, both in the design step to determine the coefficients and at runtime evaluating the filters which is in contrast to the IIR filter. For both kind of FIR filters, the transfer function outside the notch deviates considerably from the 0db line. The deviations have a hot spot at the notch frequency for approximating FIR filters whereas for eliminating FIR filters high deviations occur also far from the notch frequency. This is another disadvantage compared to the IIR filter.

In [RKJ01], Section 2.3.3 another computational path for approach II, approximation of IIR filter is presented, which is indeed mathematically equivalent, but may behave considerably different from a numerical point of view.

The content of this paper is widely phenomenological. From a mathematical point of view many statements are conjectures and must be proved or refuted. Above all this is true for limit transitions: what happens if the pole radius approaches 1 and what if the filter order goes to infinity.

In frequency domain we only consider the absolute value, not the delay. This is compensated to some extent by the analyses in time domain.

We computed just the filter coefficients of the FIR filters with adaptive precision so called l2r-arithmetic; the rest, i.e. the computation of the transfer function and application of the filters, including filtering a calibration signal and a signal switching on a sine with notch frequency, are computed still in MATLAB with traditional arithmetic. Although there is no indication that a special kind of arithmetic is required as it is the case in computation of the filter coefficients of the FIR filters, it would be better to compute all with one arithmetic. Currently, this is not possible, because l2r-arithmetic is restricted to real numbers and is bad in evaluating a filter. In the long run, the l2r-arithmetic shall be extended accordingly.

One of the caveats is also that the computation is slow and memory intensive. Since on the other hand, we found potential in improving performance, we should do that because the bad performance shrinks the possible field of application for the methods we used.

Another problem is, that we only partially understand where the numerical instability for poles near the unit circle in the complex plain comes from. One source are definitely the binomial coefficients, but the mechanism the binomial coefficients lead to instability is not clear, neither we know whether binomial coefficients are the sole source of instability.

Finally, it seems quite interesting to know how precise the radius and the input signals must be known to obtain output with resolution given by the DA converter. This proves whether the AD converter fits the DA converter. Precision of the filter coefficients are also vital to substantiate hardware design, in particular if fixed point arithmetic is used, think of small DSP and FPGAs.

Currently, `ArithIntOctave` has methods `fir` and `iir` which are quite slow. In contrast, `octave` and also `matlab`, the latter maybe also some toolbox, has a function `filter` which even generalizes `iir` and is much faster. In the long run, `ArithIntOctave` shall provide a comparable implementation. We don't expect numerical problems in executing the filters but on the other hand, it is a good idea to find out the precision required for the signals and for the coefficients. As a consequence, one could find out dependencies on the parameters. For the notch frequency this gives a measure on the sharpness of the filter. For the pole radius it shows boundaries in which changes really make a difference in stability of algorithm and damping of ringing.

References

[EN615] *Medical electrical equipment – Part 2-25: Particular requirements for the basic safety and essential performance of electrocardiographs (EN 60601-2-25:2015).* 3 rue de Varembé, PO Box 131 1211 Geneva 20, Switzerland, 10 2015.

[LRO94] T.I. Laakso, J. Ranta, and S.J. Ovaska. Design and implementation of efficient IIR notch filters with quantization error feedback. *IEEE Trans. Instrum. Meas.*, 43:449–456, 6 1994.

[Orf96] S.J. Orfanidis. *Introduction to Signal Processing.* Prentice Hall, Englewood Cliffs, New Jersey, 1996.

[RJK94] S.C. Dutta Roy, Shail B. Jain, and B. Kumar. Design of digital FIR notch filters. *IEE Proc. on Vision, Image and Signal Processing*, pages 334–338, 10 1994.

[RJK97] S.C. Dutta Roy, Shail B. Jain, and B. Kumar. Design of digital FIR notch filters from second order IIR prototype. *IETE Journal of Research*, 43(4):275–279, 8 1997.

[RKJ01] S. C. Dutta Roy, B. Kumar, and Shail B. Jain. FIR notch filter design – a review. *FACTA UNIVERSITATIS (NIS), Series: Electronics and Energetics*, 14(3.):295–327, 12 2001.

[TP01] Chien-Chien Tseng and Soo-Chang Pei. Stable IIR notch filter design with optimal pole placement. *IEEE Transactions on Signal Processing*, 49(11):334–338, 11 2001.

```
%%
%% @seealso{fir}
%% @seealso{iir}
%% @seealso{weakIirNotch}
%% @end deftypefn

function dMaxArr = firNotchFromIir(angle, rPole, DegN, alt)
    cosAngle = cos(angle);
    bCoeff = rPole^2;
    aCoeff = 2. * rPole * cosAngle;
    DegN1 = DegN;
    if alt
        DegN1 = DegN1 + 2;
    end
    dMinArr = initCoeffMinD(aCoeff, bCoeff, DegN1);
    [dMaxArr, scale] = initCoeffMaxD(DegN, DegN1, cosAngle, dMinArr);
    %assert(scale > 0);
end % function firNotchFromIir

function dMinArr = initCoeffMinD(aCoeff, bCoeff, DegN)
for idx = DegN - 2:-1:0 % top down for performance reason
    % here, add the d_{idx} needed.
    d = 0.;
    for m = 0:floor(idx/2)
        d = d+(-1.)^m*nchoosek(idx-m,m)*aCoeff^(idx-2*m)*bCoeff^m;
    end
    dMinArr(idx+1) = d;
end % for
%assert(dMinArr(1) == 1);
end % function initCoeffMinD

function d = coeffMinD(DegN, dMinArr, i)
assert(i >= -2 && i <= DegN);
if i < 0 || i > DegN - 2
    d = 0.;
    return;
end
% obj.dMinArr(numel...) contains d_{numel(...)-1}
d = dMinArr(i+1);
end % function coeffMinD


function [dMaxArr, scale] = initCoeffMaxD(DegN, DegN1, cosAngle, dMinArr)
sum = 0.;
for idx = DegN:-1:0
  d0 = coeffMinD(DegN1, dMinArr, idx);
  d1 = coeffMinD(DegN1, dMinArr, idx-1);
  d2 = coeffMinD(DegN1, dMinArr, idx-2);
  d = d0 - 2. * d1 * cosAngle + d2;
    dMaxArr(idx+1) = d;
    sum = sum + d;
end % for
scale = 1. / sum;
dMaxArr = dMaxArr * scale;
end % function coeffMaxD

% function [dMaxArr, scale] = initCoeffMaxD2(DegN, cosAngle, dMinArr)
% sum = 0;
% for idx = DegN:-1:0
%     d =        coeffMinD(DegN, dMinArr, idx) ...
%           -2 * coeffMinD(DegN, dMinArr, idx-1) * cosAngle ...
%           +    coeffMinD(DegN, dMinArr, idx-2);
%     dMaxArr(idx+1) = d;
%     sum = sum + d;
% end % for
% scale = 1 / sum;
% dMaxArr = dMaxArr * scale;
% end % function coeffMaxD2
```

Listing 1: Matlab without comment: FIR notch from IIR notch

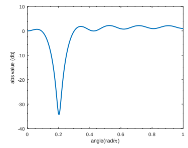
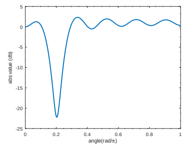
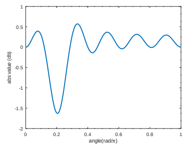
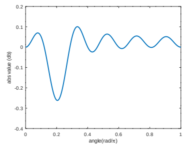
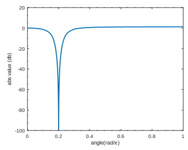
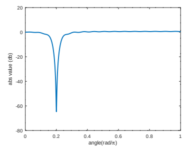
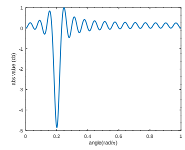
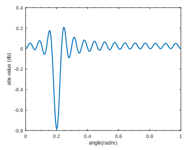
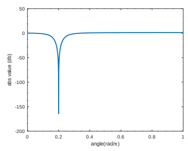
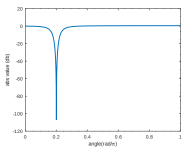
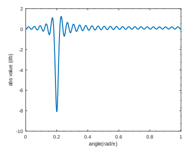
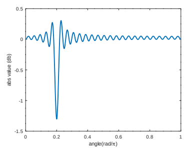
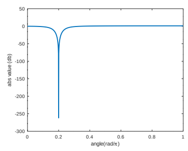
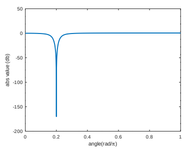
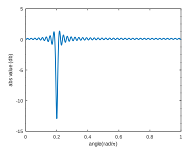
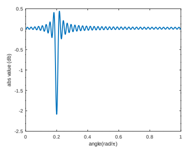
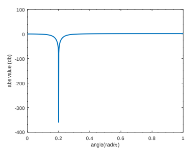
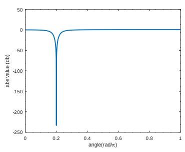
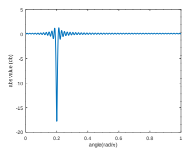
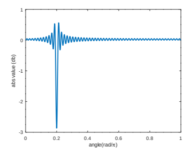
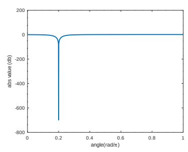
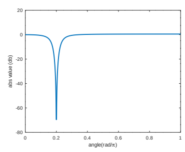
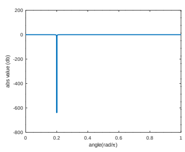
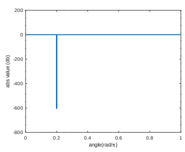
Order	$r = 0.85$	$r = 0.9$	$r = 0.992$	$r = 0.9987$
10				
30				
50				
80				
110				
IIR				

Table 5: Transfer functions of FIR filters approximating the according IIR filter (last row) with various combinations of pole radii and filter orders

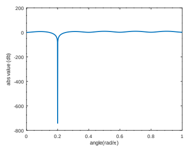
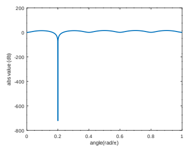
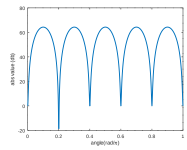
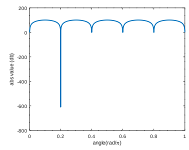
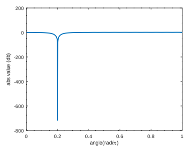
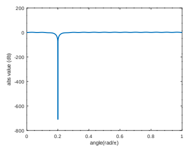
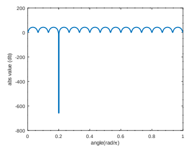
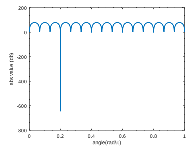
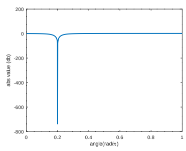
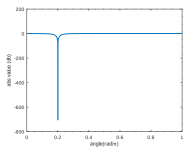
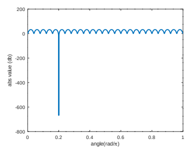
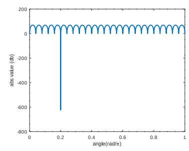
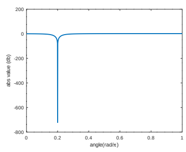
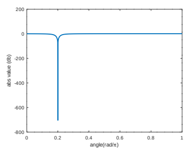
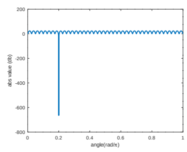
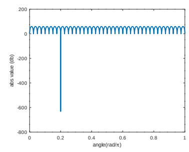
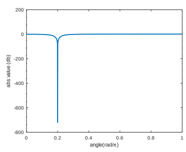
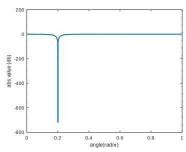
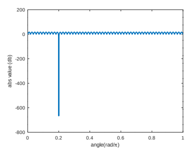
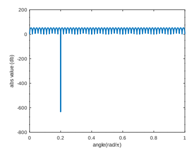
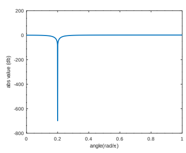
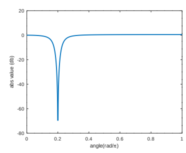
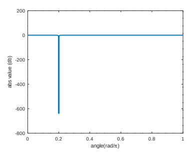
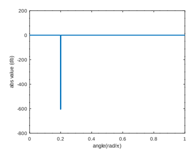
Order	$r = 0.85$	$r = 0.9$	$r = 0.992$	$r = 0.9987$
10				
30				
50				
80				
110				
IIR				

Table 6: Transfer functions of FIR filters, eliminating the notch frequency with various combinations of pole radii and filter orders, together with the transfer functions of their according IIR filters (last row)

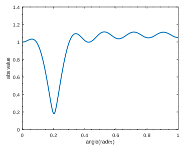
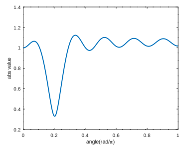
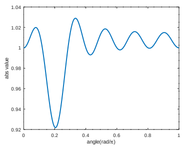
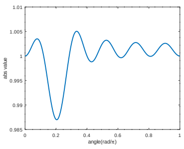
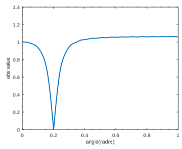
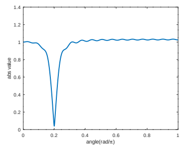
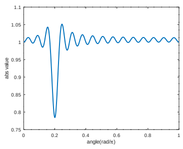
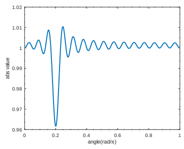
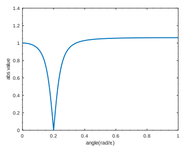
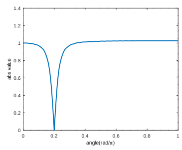
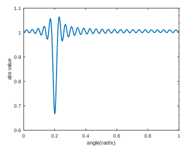
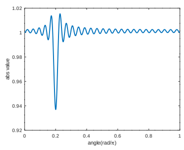
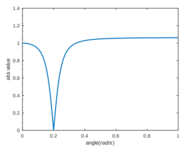
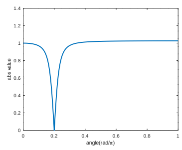
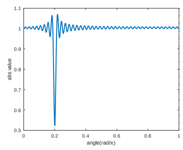
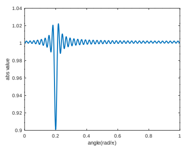
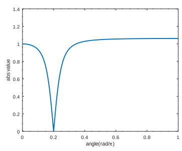
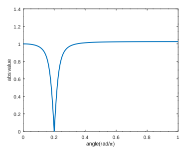
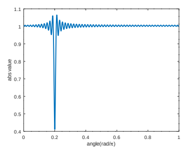
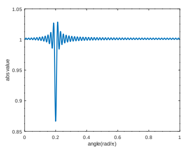
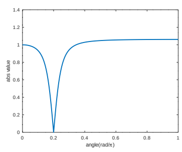
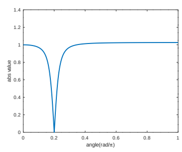
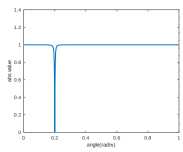
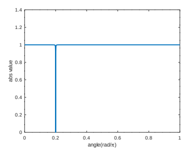
Order	$r = 0.85$	$r = 0.9$	$r = 0.992$	$r = 0.9987$
10				
30				
50				
80				
110				
IIR				

Table 7: Transfer functions of FIR filters approximating the according IIR filter with various combinations of pole radii and filter orders represented in **linear y scale**

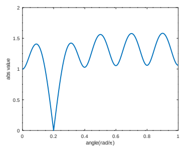
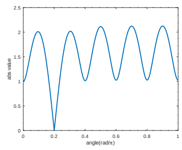
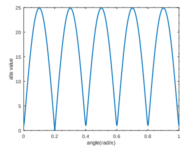
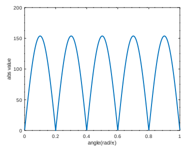
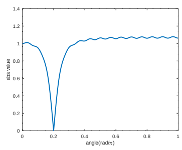
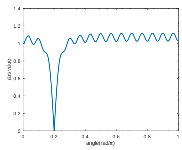
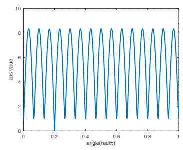
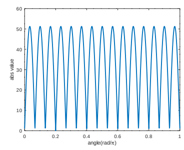
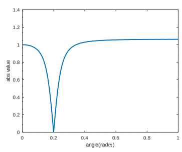
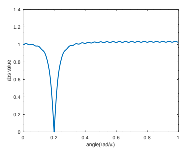
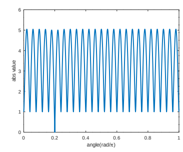
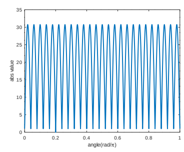
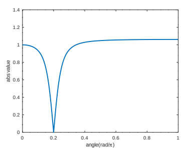
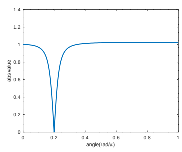
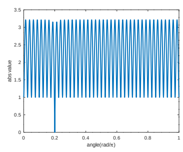
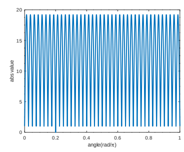
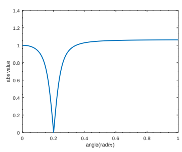
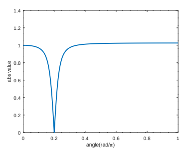
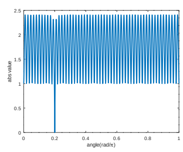
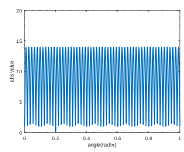
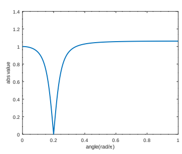
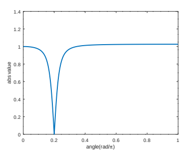
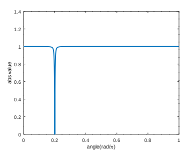
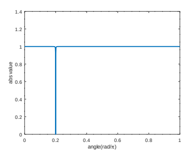
Order	$r = 0.85$	$r = 0.9$	$r = 0.992$	$r = 0.9987$
10				
30				
50				
80				
110				
IIR				

Table 8: Transfer functions of FIR filters, eliminating the notch frequency with various combinations of pole radii and filter orders, together with the transfer functions of their according IIR filters represented in **linear y scale**

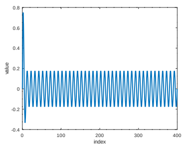
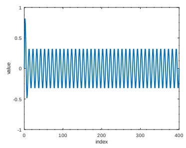
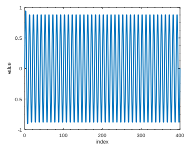
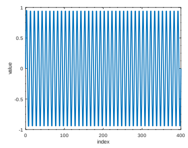
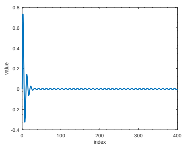
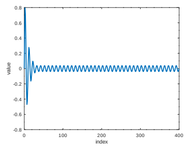
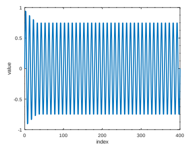
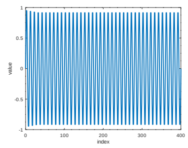
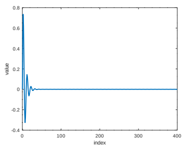
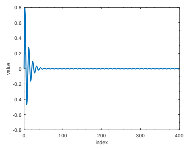
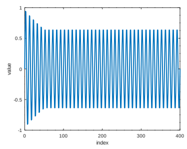
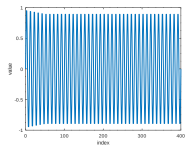
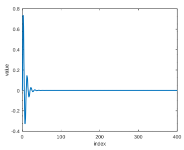
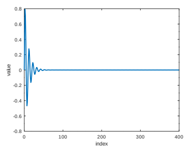
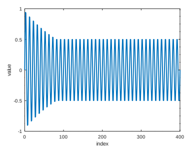
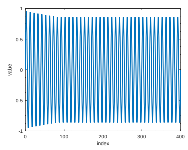
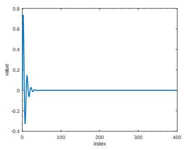
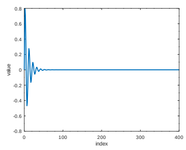
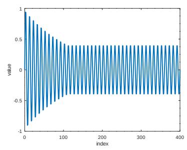
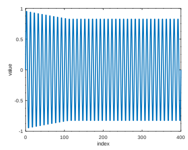
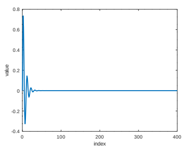
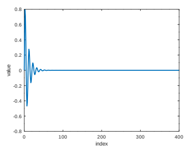
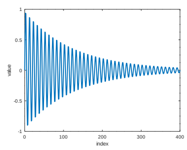
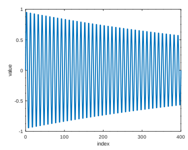
Order	$r = 0.85$	$r = 0.9$	$r = 0.992$	$r = 0.9987$
10				
30				
50				
80				
110				
IIR				

Table 9: Sine 50Hz filtered by FIR filters, approximating the according IIR filter (last row) with various combinations of pole radii and filter orders

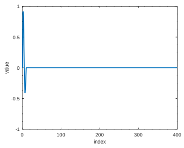
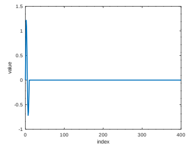
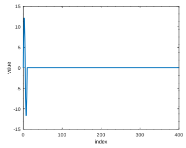
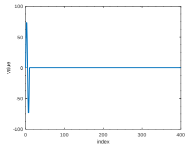
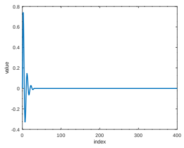
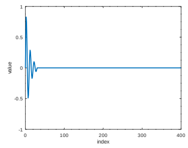
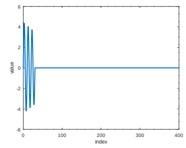
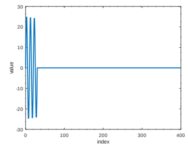
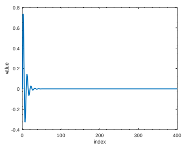
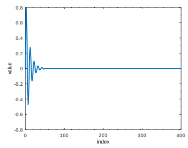
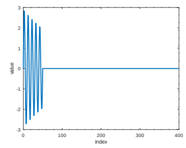
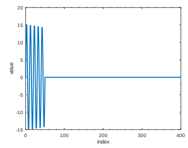
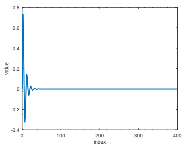
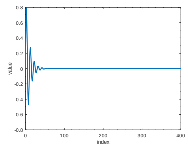
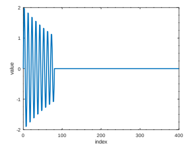
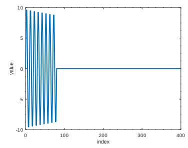
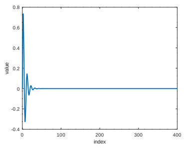
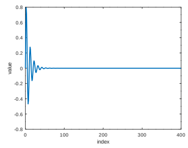
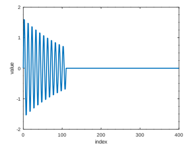
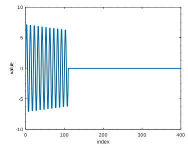
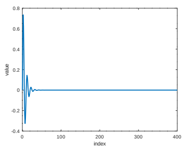
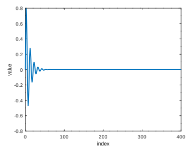
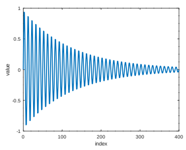
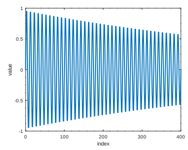
Order	$r = 0.85$	$r = 0.9$	$r = 0.992$	$r = 0.9987$
10				
30				
50				
80				
110				
IIR				

Table 10: Sine 50Hz filtered by FIR filters, eliminating the notch frequency with various combinations of pole radii and filter orders, and filtered by their according IIR filters (last row)

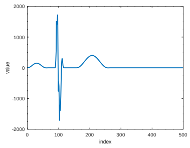
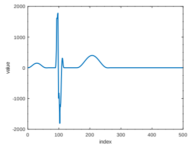
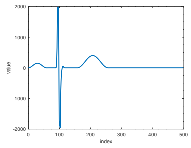
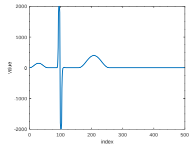
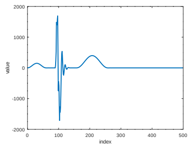
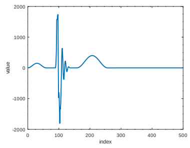
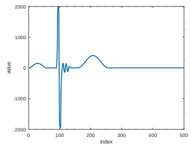
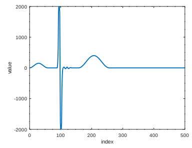
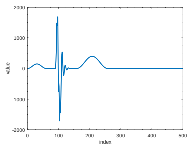
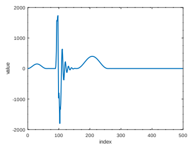
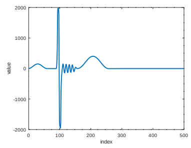
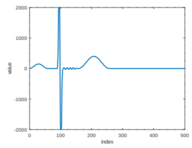
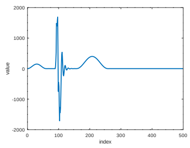
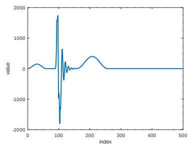
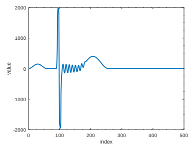
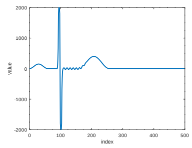
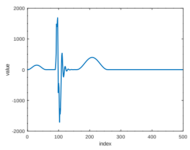
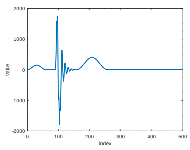
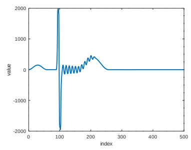
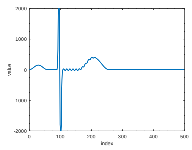
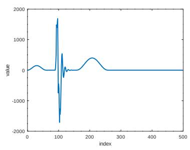
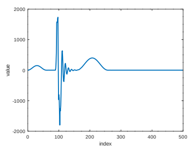
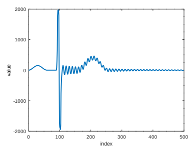
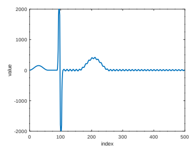
Order	$r = 0.85$	$r = 0.9$	$r = 0.992$	$r = 0.9987$
10				
30				
50				
80				
110				
IIR				

Table 11: Calibration signal `CAL20500` filtered by FIR filters, approximating the according IIR filter (last row) with various combinations of pole radii and filter orders

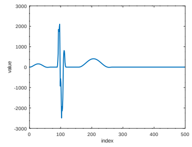
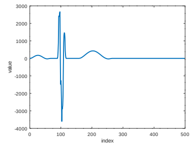
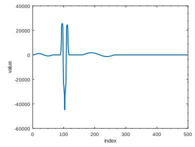
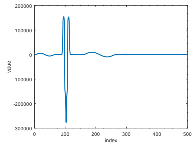
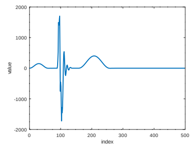
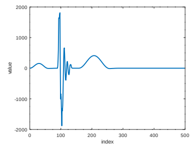
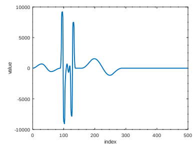
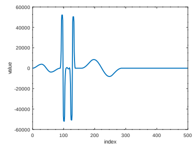
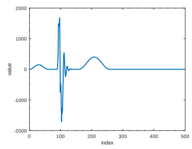
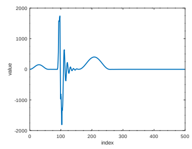
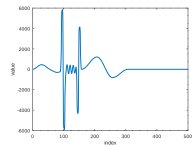
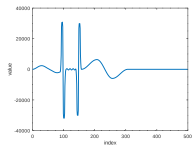
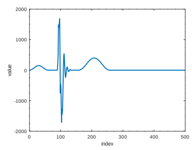
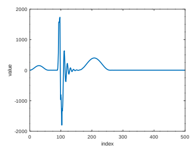
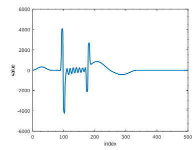
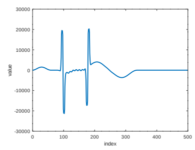
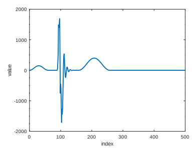
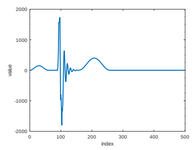
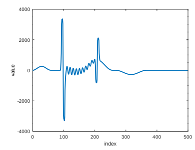
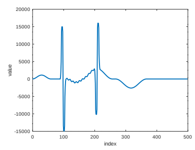
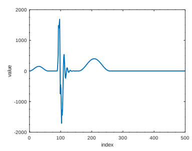
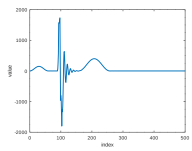
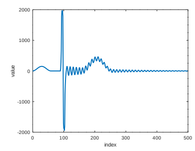
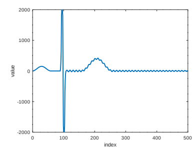
Order	$r = 0.85$	$r = 0.9$	$r = 0.992$	$r = 0.9987$
10				
30				
50				
80				
110				
IIR				

Table 12: Calibration signal `CAL20500` filtered by FIR filters, eliminating the notch frequency with various combinations of pole radii and filter orders, and filtered by their according IIR filters (last row)