Fast Deconvolution Using Frequency-Dependent Regularization

Ole Kirkeby, Per Rubak, and Angelo Farina*

Department of Communication Techology, Fredrik Bajers Vej 7, Aalborg University, DK-9220 Aalborg Ø, Denmark

***Department of Industrial Engineering, University of Parma, Via delle Scienze, 43100 Parma, Italy**

A very fast deconvolution method, which is based on the Fast Fourier Transform, can be used to design a matrix of causal finite impulse response filters whose performance is optimized at a large number of discrete frequencies. The method is very efficient for both single-channel deconvolution, which can be used for loudspeaker equalisation, and multi-channel deconvolution, which can be used to design cross-talk cancellation networks.

The fast deconvolution algorithm essentially provides a quick way to solve, in the least squares sense, a linear equation system whose coefficients, right hand side, and unknowns are z-transforms of stable digital filters. Frequency-dependent regularisation is used to prevent sharp peaks in the magnitude response of the optimal filters. A z-domain analysis demonstrates that the regularisation works by pushing the poles of an ideal optimal solution away from the unit circle, and in doing so reduces the effective duration of its time response. However, the algorithm assumes that it is feasible to use long optimal filters, and it works well only when two regularisation parameters, a shape factor and a gain factor, are set appropriately. In practice, the values of the two regularisation parameters are most easily determined by trial-and-error experiments.

EDICS code: SA 2.3.3

I. INTRODUCTION

Deconvolution in its most basic form can be described as the task of calculating the input to a discrete-time system from its output [1, p.16]. It is usually assumed that the system is linear and that its input-output mapping is known with good accuracy. In acoustics and audio, single-channel deconvolution is particularly useful since it can compensate for the response of imperfect transducers such as headphones, loudspeakers, and amplifiers [2], [3]. Even the room response can be compensated for if desired [4], [5]. Multi-channel deconvolution is necessary in the design of cross-talk cancellation systems and virtual source imaging systems ,[6], [7], [8], [9], [10]. In practice, audio-related deconvolution problems usually involve long impulse responses, and since the methods that are traditionally used to design control systems for conventional engineering applications generally assume a relatively low filter order, those methods are not entirely suitable for our purpose. It is mainly for this reason that we are interested in developing computationally efficient filter design algorithms.

Deconvolution in the time domain tends to lead to complicated expressions for the optimal filters regardless of whether one chooses to use an adaptive method [6], [7] or a closed form method [9], [10]. This is because time domain filter design is global in the sense that all the unknown coefficients must be adjusted simultaneously — the problem cannot be decomposed into a smaller number of independent problems. Consequently, one has to solve a large number of coupled linear equations (although it is sometimes possible to take advantage of the specific structure of the equation system [5]). Nevertheless, the global optimisation scheme ensures that the optimal filters make very good use of their coefficients, and they are therefore very efficient from an implentation point of view. The trade-off between design- and implementation efficiency means that time domain methods are appropriate when the optimal filters are short (each contains less than a few hundred coefficients as a rule of thumb).

Deconvolution in the frequency domain tends to lead to simple expressions for the optimal filters since their responses can be calculated individually at a large number of discrete frequencies [11]. This means that the equations are uncoupled and can be solved independently (this is equivalent to solving a linear equation system whose coefficient matrix is of diagonal form). However, if the magnitude responses of the optimal filters are very large within a narrow range of frequencies, the impulse responses corresponding to the optimal frequency responses will ring for a long time and possibly even cause an unwanted wrap-around effect. Regularisation can prevent this from happening by reducing the longest time constant of the optimal filters, but even so the optimal filters still need to be long enough to ensure that the impulse responses have effectively died away well before the ends of the filters. Consequently, the optimal filters do not make very good use of their coefficients, and they are therefore not very efficient from an implementation point of view. Frequency domain methods are appropriate when the optimal filters are long (each contains more than a few hundred coefficients as a rule of thumb).

A modeling delay can be used to ensure that the optimal filters perform well not only in terms of amplitude, but also in terms of phase [1, Example 7.2.2]. By delaying the output by a small amount, typically a few milliseconds, it is possible to compensate for non-minimum phase components that would otherwise cause the optimal filters to be either unstable in forward time, or non-causal. Our method uses a modeling delay, and we will not consider minimum-phase implementations.

Regularisation is a method that is commonly used when one is faced with an ill-conditioned problem [12, Section 18.4]. The basic idea is to prevent the solution from having some undesirable feature by adding a “smoothness” term to the cost function that we wish to minimise. A suitable choice of the smoothness term can improve the conditioning of the problem substantially. In addition, the value of a regularisation parameter must be set appropriately. The regularisation parameter determines how much weight to assign to the smoothness term, and since regularisation by its very nature imposes a subjective constraint on the solution, it can be difficult to come up with a reliable objective method for determing this weight.

In the following, we extend the fast deconvolution method described in [11] to include frequency-dependent regularisation, and we show that the effect of regularisation can be conveniently explained by a pole-zero analysis of a matrix of idealised optimal filters.

II. SYSTEM DESCRIPTION

The discrete-time multichannel deconvolution problem is shown in block diagram form in Fig. 1. We will use z-transforms to denote discrete time filters and signals [1, Chapter 3], and sometimes these z-transforms will be referred to as polynomials even though strictly speaking their powers are negative. We define the following column vectors: $\mathbf{u}(z)$ is a vector of T observed signals, $\mathbf{v}(z)$ is a vector of S source input signals, $\mathbf{w}(z)$ is a vector of R reproduced signals, $\mathbf{d}(z)$ is a vector of R desired signals, and $\mathbf{e}(z)$ is a vector of R performance error signals. The matrices $\mathbf{A}(z)$, $\mathbf{C}(z)$, and $\mathbf{H}(z)$ represent multi-channel filters. $\mathbf{A}(z)$ is an $R\times T$ target matrix, $\mathbf{C}(z)$ is an $R\times S$ plant matrix, and $\mathbf{H}(z)$ is an $S\times T$ matrix of optimal filters ($\mathbf{H}$ is now denoted without the subscript $m,\mathbf{A}$ that was used in [11]). We assume that the elements of $\mathbf{A}(z)$, $\mathbf{C}(z)$, and $\mathbf{H}(z)$ are finite impulse response (FIR) filters. Note that the vectors read alphabetically **u**, **v**, **w** along the lower half of the block diagram, and that their dimensions read *T*, *S*, *R* (*R*, *S*, *T*, in reverse); this will make the notation easier to remember. The component z^{-m} implements a so-called modeling delay by shifting all the elements of **u** by an integer number of *m* samples [1, Example 7.2.2]. The problem is to determine $\mathbf{H}(z)$, and in order to achieve this it is necessary to invert $\mathbf{C}(z)$ in some sense.

III. EXACT LEAST SQUARES DECONVOLUTION

The idea central to our filter design algorithm is to minimise, in the frequency domain, a cost function of the type

$$J = E + \beta V \tag{1}$$

where *E* is a measure of the performance error **e** and *V* is a measure of the effort **v**. The positive real number β is a regularization parameter that determines how much weight to assign to the effort term. As β is increased from zero to infinity, the solution changes gradually from minimizing *E* only to minimizing *V* only. By making the regularization frequency-dependent, we can control the time response of the optimal filters in quite a profound way. However, instead of specifying β as a function of frequency it is advantegous to build the frequency-dependence into *V*.

It is convenient to consider the regularization to be the product of two components: a gain factor and a shape factor. The gain factor is the conventional regularization parameter β, and the shape factor $B(z)$ is the *z*-transform of a digital filter that amplifies the frequencies that we *do not* want to see boosted by $\mathbf{H}(z)$. Frequencies that are suppressed by $B(z)$ are not affected by the regularization. Although it is the frequency response, and not the time response, of $B(z)$

that is important, we prefer to design $B(z)$ in the time domain. The phase response of $B(z)$ is irrelevant since $H(z)$ is determined by minimizing an energy quantity.

The derivation of $\mathbf{H}(z)$ in the general multi-channel case is directly analogous to that presented in [11]. We find

$$\mathbf{H}(z) = \left[\mathbf{C}^{\mathrm{T}}(z^{-1})\mathbf{C}(z) + \beta\, B(z^{-1})B(z)\mathbf{I}\right]^{-1}\mathbf{C}^{\mathrm{T}}(z^{-1})\mathbf{A}(z)\, z^{-m} \tag{2}$$

In the single-channel case, this result simplifies to

$$H(z) = \frac{C(z^{-1})\,A(z)}{C(z^{-1})C(z) + \beta\, B(z^{-1})B(z)}\, z^{-m} \tag{3}$$

This special case is very important, not only because it is often encountered in practice, but also because it suggests a way to analyze the more complex multichannel result.

IV. POLE-ZERO ANALYSIS

It is seen from Eq. 3 that in the single-channel case $H(z)$ can be naturally expressed in rational form, just like a conventional infinite impulse response (IIR) filter. Consequently, we can learn about the properties of this filter by looking at its poles and zeros in the complex plane. The zeros of $H(z)$ are the zeros of the numerator of Eq. 3, and the poles of $H(z)$ are the zeros of the denominator of Eq. 3. The positions of the poles with respect to the unit circle are particularly important. Poles near the unit circle make the time response of the filter decay away very slowly [13, Section 8.2.4]. The time constant τ, in samples, associated with a single pole close to the unit cirle is approximately proportional to the reciprocal of the distance r between the two, so

$$\tau = \frac{1}{r} \qquad (4)$$

when r<<1 [14]. If the pole is just inside the unit circle, the filter's time response will be right-sided and decay away in forward time, if the pole is just outside the unit circle, its response will be left-sided and decay away in backward time [15, Chapter 2].

A. General results; the single-channel case

We start by observing that if Eq. 3 is written as a fraction,

$$H(z) = \frac{P(z)}{Q(z)} \qquad (5)$$

then the zeros of $H(z)$ are the zeros of the numerator polynomial $P(z)$,

$$P(z) = C(z^{-1})\,A(z)\,z^{-m} \qquad (6)$$

and the poles of $H(z)$ are the zeros of the denominator polynomial $Q(z)$,

$$Q(z) = C(z^{-1})C(z) + \beta\, B(z^{-1})B(z) \qquad (7)$$

B. Two single-channel examples

In order to illustrate how the regularisation modifies the pole-zero structure of $H(z)$, we consider the simple system

$$C(z) = 1 - z^{-1} \qquad (8)$$

$C(z)$ has a single zero on the unit circle at z = 1. If we set $A(z)$ = 1 (flat target response with zero phase), $m = 0$ (no modeling delay), and $B(z) = 1$ (frequency-independent regularisation), then from Eq. 3 we find

$$H(z) = \frac{1 - z^{-1}}{(1 - z)(1 - z^{-1}) + \beta} \tag{9}$$

$H(z)$ has a zero at $z = 1$, and two poles, also on the real axis, at

$$z = 1 \pm \frac{\sqrt{\beta^2 + 4\beta}}{2} + \frac{\beta}{2} \tag{10}$$

When β<<1, a series expansion of this expression gives

$$z = 1 \pm \sqrt{\beta} + O(\beta) \tag{11}$$

which shows that for small values of β, the distance from the two poles of $H(z)$ to the unit circle is proportional to the square root of β. For example, if β = 0.0001, then the two poles of $H(z)$ are on the real axis at 1.01 and 0.99. According to Eq. 4, this corresponds to a time constant τ of approximately 100 samples since both poles are a distance of approximately 0.01 away from the unit circle. We now consider how frequency-dependent regularization modifies the pole-zero structure of a slightly more complex system.

Fig. 2 shows the properties of the sequence $c(n) = \{1,0,0,0,0.96\}$. This filter has been constructed from its four zeros at 0.99, ±0.99i, and -0.99. Note that the zeros are evenly spaced around the unit at a distance of 0.01 away from it. Fig. 2a shows the modulii of the zeros of $C(z)$ plotted against their arguments in radians (note the scaling of the y-axis) , and Fig. 2b shows the magnitude response $|C|$ of $C(z)$.

Fig. 3 shows the pole-zero map of the filter $H(z)$ calculated from Eq. 3 when β = 0.003

and $A(z) = 1$. In Fig. 3a, the shape factor is constant as a function of frequency which means that B(z) = 1. In Fig. 3b, the shape factor is a first order high-pass FIR filter, $B(z) = 1\text{-}z^{-1}$ (see Eq. 8), whose single zero is on the unit circle at 1. The circles are the zeros, given by the zeros of $P(z)$ (see Eq. 6), and the crosses are the poles, given by the zeros of $Q(z)$ (see Eq. 7). If β was zero, such a plot would show half the poles being cancelled out exactly by zeros. The positions of the surviving poles would correspond exactly to the zeros of $P(z)$ (see Fig. 2a). When β is increased, however, the poles move away from the unit circle in the regions where $B(z)$ contains energy. In Fig. 3a, $B(z)$ is an all-pass filter, and so all the poles have moved. In Fig. 3b, $B(z)$ is a high-pass filter, and so the poles bend away from the unit circle at high frequencies (at arguments near $\pm\pi$) whereas the pole just outside the unit circle at zero radians is cancelled by a zero of $P(z)$ because the regularisation does not have any effect at low frequencies.

Fig. 4 shows the magnitude response $|H|$ of $H(z)$ calculated with frequency-dependent regularisation (solid line) and with no regularisation (dashed line). Thus, the solid line in Fig. 4a corresponds to the frequency response of the signal whose pole-zero map is plotted in Fig. 3a whereas the solid line in Fig. 4b corresponds to the frequency response of the signal whose pole-zero map is shown in Fig. 3b. It is seen that the frequency-dependent regularization has succeeded in attenuating high frequencies without affecting low frequencies.

C. General results; the multichannel case

Just as in the single-channel case, we start by writing the optimal filter matrix $\mathbf{H}(z)$ (see Eq. 2) in the form of a “fraction”,

$$\mathbf{H}(z) = \mathbf{Q}^{-1}(z)\mathbf{P}(z)\,. \tag{12}$$

Here, **Q**(z) is a square $S \times S$ matrix,

$$\mathbf{Q}(z) = \mathbf{C}^{\mathrm{T}}(z^{-1})\mathbf{C}(z) + \beta\, B(z^{-1})B(z)\mathbf{I}\,, \tag{13}$$

and **P**(z) is an $S \times T$ matrix,

$$\mathbf{P}(z) = \mathbf{C}^{\mathrm{T}}(z^{-1})\mathbf{A}(z)\, z^{-m}\,. \tag{14}$$

If we invert **Q**(z) by dividing its adjoint adj[**Q**(z)] by its determinant $Q(z)$ [16, Section 0.8.2], we can write

$$\mathbf{H}(z) = \frac{\mathrm{adj}\left[\mathbf{Q}(z)\right]}{Q(z)}\mathbf{P}(z) \tag{15}$$

It is seen that the determinant of **Q**(z), which is a scalar function of z as in the single-channel case, is a common denominator of all the elements of **H**(z). This is a strong result; it implies that the elements of **H**(z) share a common set of poles given by the zeros of the polynomial $Q(z)$, and in addition that those poles are not related in a simple way to the zeros of the elements of **Q**(z). Consider, for example, the two-by-two system

$$\mathbf{C}(z) = \begin{bmatrix} 1 & 0.5z^{-1} \\ 0.5z^{-1} & 1 \end{bmatrix}. \tag{16}$$

If $B(z) = 1$ and **A**(z) is an identity-matrix of order 2, then

$$\mathbf{Q}(z) = \begin{bmatrix} 1.25 & 0.5(z + z^{-1}) \\ 0.5(z + z^{-1}) & 1.25 \end{bmatrix}. \tag{17}$$

The two off-diagonal elements of $\mathbf{Q}(z)$ each have two zeros on the imaginary axis at +i and -i whereas the diagonal elements do not have any finite zeros at all. Nevertheless, $Q(z)$ has two zeros on the real axis at ±0.5 and another two at ±2, and none of these coincides with the zeros of any of the individual elements of $\mathbf{Q}(z)$, or $\mathbf{C}(z)$.

D. The roots of the denominator Q(z)

It is no trivial matter to find the roots of a polynomial of high order, and there is an overwhelming amount of literature available on the subject (see [17] for a very comprehensive bibliography). A very reliable numerical method finds a polynomial’s roots by calculating the eigenvalues of its related companion matrix, but this algorithm is of complexity $O(N^3)$ since it spends most of its time by transforming the polynomial's companion matrix into Hessenberg form (see [18, Chapter 13] for details). Consequently, it is quite expensive computationally. As an alternative, one can use an algorithm based on the principle of deflation and root polishing which is approximately of complexity $O(N^2)$, but this method can fail some cases (see [19] for details). As a rule of thumb, a polynomial whose degree is less than 500 can easily be factored on a fast PC by the eigenvalue method. A polynomial whose degree is greater than 5,000 is likely to be difficult to factor no matter which method is used. As an alternative to explicit factoring of a high-order polynomial, there exists efficient methods for counting the number of roots contained inside a circle of a given radius [20], [21].

Note that the “inverse” of the root-finding problem, which is the problem of calculating the coefficients of a polynomial from its zeros, is extremely sensitive to round-off errors (as demonstrated by Wilkinson’s famous example [18, p. 201]). Even polynomials of relatively low order, say, $N = 50$, are usually unrecognisable when reconstructed from their roots using double-precision arithmetic, and for this reason it is generally not practical directly to manipulate the poles and zeros of a digital filter. There are few general results that are useful to us but the following three rules give some idea about what to expect in practice. The rules concern symmetry, clustering, and attractors.

First, the roots of $Q(z)$ always appear in groups of two or four. Since $Q(z)$ is equal to $Q(z^{-1})$ it follows that if z_0 is a zero of $Q(z)$ then so is $1/z_0$. Furthermore, zeros off the real axis must appear in complex conjugate pairs since the coefficients of $Q(z)$ are real. Note that the symmetry in z and z^{-1} means that for each zero inside the unit circle, there is a corresponding zero outside, and this effectively spoils our chances of finding a stable optimal filter (apart from in a few special cases).

Secondly, the roots of a polynomial of high order are not scattered all over the complex plane, but rather "...the roots of a random polynomial tend to be evenly distributed in angle and tightly clustered near the unit circle as the degree of the polynomial increases" (quote from [19]). Since most signals we come across in practice, such as measured impulse responses, have a certain degree of randomness built into them, this asymptotic result accurately describes what is most often observed with experimental data.

Thirdly, the roots of $B(z^{-1})B(z)$ act as a kind of attractor set for the roots of $Q(z)$ for large values of β. It is easily verified that when β is very small, the roots of $Q(z)$ are those of $C(z^{-1})C(z)$ whereas when β is very large, the roots of $Q(z)$ are those of $B(z^{-1})B(z)$. Consequently, for some choices of $B(z)$ it is possible that excessive use of regularisation can cause some of the poles to be pulled back towards the unit circle. Even though this will happen only for very large values of β, it can be a real concern when one tries to implement “black box” regularisation routines. Note that when frequency-independent regularisation is used, $B(z)$ is a constant and consequently it has no roots. The implication of this is that for large β the attractors are zero (the origin) and complex infinity (points very far away from the origin), and so in this case the regularisation will push all the poles away from the unit circle.

Finally, it should be mentioned that the practical value of knowing the roots of $Q(z)$ is mainly limited to the cases where we can pick out isolated roots near the unit circle. This is usually not possible when we are dealing with polynomials of very high order, and so in this case the pole-zero analysis is not necessarily a useful design tool.

V. THE FAST DECONVOLUTION ALGORITHM

Although z-transforms are easy to manipulate formally, we cannot expect the time sequence corresponding to an arbitrarily choosen z-transform to be both causal and stable. In fact, if the z-transform has poles away from the origin and infinity, the corresponding time sequence is not even unique. However, if we decide that the sequence’s region of convergence has to include the unit circle then the filter will be stable as long as there are no poles on the unit circle. Each pole inside the unit circle will make a contribution that decays away in forward time whereas each pole outside the unit circle will make a contribution that decays away in backward time. The filter can, in principle, be made causal by shifting its impulse response very far to the right (delaying it by a large amount). In this section we show how to calculate a matrix of optimal causal FIR filters that each contain N_h coefficients. Since this method uses Fast Fourier Transforms (FFTs), N_h must be a power of two. The implementation of the inversion method is straightforward in practice. FFTs are used to get in and out of the frequency domain, and the system is inverted for each frequency in turn. Since using the FFT effectively means that we are operating with periodic sequences, a cyclic shift of the inverse FFTs of the optimal frequency responses is used to implement a modelling delay [??].

Eq. 2 gives an expression for the response of $\mathbf{H}(z)$ as a continuous function of frequency. If an FFT is used to sample the frequency response of $\mathbf{H}(z)$ at N_h points *without* including the phase contribution from the modeling delay, then the value of $\mathbf{H}(k)$ at those frequencies is given by

$$\mathbf{H}(k) = \left[\mathbf{C}^{\mathrm{H}}(k)\mathbf{C}(k) + \beta\, B^{*}(k)B(k)\mathbf{I}\right]^{-1}\mathbf{C}^{\mathrm{H}}(k)\mathbf{A}(k) \qquad (18)$$

where k denotes the k'th frequency line; that is, the frequency corresponding to the complex number exp(i2k/N_h). The superscript H denotes the Hermitian operator that transposes and conjugates its argument, the superscript * denotes complex conjugation of its scalar argument. In the single-channel case, $\mathbf{C}^{\mathrm{H}}(k)$ is equivalent to $C^{*}(k)$. In order to calculate the impulse responses of a matrix of causal filters the following steps are necessary.

1. Calculate $\mathbf{A}(k)$, $B(k)$, and $\mathbf{C}(k)$ by taking N_h-point FFTs of each of their elements

2. For each of the N_h values of k, calculate $\mathbf{H}(k)$ from Eq. 18

3. Calculate one period of $\mathbf{h}(n)$ by taking N_h-point inverse FFTs of the elements of $\mathbf{H}(k)$

4. Implement the modeling delay by a cyclic shift of m samples of each element of $\mathbf{h}(n)$

The exact value of m is not critical; a value of $N_h/2$ is likely to work well in all but a few cases.

VI. A MULTICHANNEL EXAMPLE: CROSS-TALK CANCELLATION

The system given by Eq. 16 is a very crude approximation to the matrix of transfer functions that one has to deal with when designing cross-talk cancellation networks. Fig. 5 shows the geometry of such a system. The two loudspeakers span only ten degrees as seen by the listener; we refer to such a loudspeaker arrangement as a stereo dipole [22]. In practice, it is necessary to take into account the influence of the listener's head on the incoming sound waves, and we will now use frequency-dependent regularisation to design a cross-talk cancellation network based on a matrix of modeled head-related transfer functions (HRTFs). The HRTFs are calculated from a analytical rigid sphere model [23], [24]. The sphere is assumed to have a radius of 7cm, and the ears not quite at opposite positions, but rather they are pushed back ten degrees so that they are at 100 degrees relative to straight front. This geometry ensures a good match to the true interaural time difference for near-frontal sources [25].

Since the two loudspeakers are placed symmetrically in front of the listener, the transfer function from the left loudspeaker to the left ear is the same as the transfer function from the right loudspeaker to the right ear. This direct path is denoted by $C_1(z)$. The symmetry of the geometry causes the transfer function from the left loudspeaker to the right ear to be the same as the transfer function from the right loudspeaker to the left ear, and this cross-talk path is denoted by $C_2(z)$. Thus, the matrix $\mathbf{C}(z)$ of plant transfer functions is a symmetric two-by-two matrix with $C_1(z)$ on the diagonal, and $C_2(z)$ off the diagonal,

$$\mathbf{C}(z) = \begin{bmatrix} C_1(z) & C_2(z) \\ C_2(z) & C_1(z) \end{bmatrix} \qquad (19)$$

Fig. 6 shows the impulse responses of a) $C_1(z)$, and b) $C_2(z)$ for a sampling frequency f_s of 44.1kHz. Since we do not have direct access to a time domain expression for the scattered field, the simulated time responses are calculated by an inverse Fourier transform of the sampled frequency response (see [23] for details). The frequency responses have been windowed in order to ensure that the time responses are of relatively short duration. The windowing in the frequency domain is equivalent to convolution with a so-called digital Hanning pulse given by the time sequence {0, 0.5, 1, 0.5, 0}. Thus, $C_1(z)$ and $C_2(z)$ are essentially low-pass filtered versions of the true transfer functions, and this must be compensated for by also setting the diagonal elements of the target matrix $\mathbf{A}(z)$ equal to the Hanning pulse. Note that $C_1(z)$ and $C_2(z)$ are quite similar because the two loudspeakers are very close together.

When $\mathbf{C}(z)$ and $\mathbf{A}(z)$ are symmetric, $\mathbf{H}(z)$ is also symmetric, and just as $\mathbf{C}(z)$ (see Eq. 19) it is made up of only two different elements, $H_1(z)$ on the diagonal and $H_2(z)$ off the diagonal,

$$\mathbf{H}(z) = \begin{bmatrix} H_1(z) & H_2(z) \\ H_2(z) & H_1(z) \end{bmatrix} \qquad (20)$$

Fig. 7 shows the magnitude response of a) $|H_1(z)|$ and b) $|H_2(z)|$ calculated with frequency-dependent regularisation (dashed lines) and with no regularisation (solid lines). The shape factor $B(z)$ contains 32 coefficients, and it is a “gradual” high-pass filter whose magnitude response increases from zero to one as the frequency increases from $0.3f_{\mathrm{Nyq}}$ to $0.4f_{\mathrm{Nyq}}$. The gain factor β is 0.2. It is seen that the regularisation has taken out the peak just below $0.6f_{\mathrm{Nyq}}$ ($\approx$12kHz), and that the response at high frequencies rolls of gently without causing a brick-wall type of low-pass filtering. Note that even though the magnitude responses of $H_1(z)$ and $H_2(z)$ are very similar, their phase responses are completely different [26].

As seen from Fig. 7, the cross-talk cancellation network requires a very powerful boost of the frequencies near DC in order to work properly. The reason why this boost is required is not that the two HRTFs $C_1(z)$ and $C_2(z)$ do not contain any significant energy at low frequencies, but rather it is because of the way the elements of $\mathbf{C}(z)$ interact (see Section IV.C). Since $C_1(z)$ and $C_2(z)$ are very similar at low frequencies, the two-by-two matrix $\mathbf{C}(z)$ will contain four almost identical numbers, and this clearly makes $\mathbf{C}(z)$ difficult to invert. It is important to realise that this is not an artifact produced by a particular mathematical model; efficient cross-talk cancellation is inherently difficult at low frequencies.

Fig. 8 shows the two different impulse responses, a) $H_1(z)$ and b) $H_2(z)$, that are necessary in order to implement a cross-talk cancellation network of the type shown in Fig. 6. The impulse responses correspond to the magnitude responses shown with the dashed line in Fig. 7, and they are calculated by the fast deconvolution method. It is seen that when the filters are long enough (N_h = 1024), the wrap-around effect is not a problem despite the very powerful low-frequency boost that characterizes both $H_1(z)$ and $H_2(z)$.

VII. DETERMINING THE REGULARIZATION GAIN- AND SHAPE FACTORS

Since the purpose of the regularization is to impose a subjective constraint on the solution, it is very difficult to come up with a reliable black box routine that can set the gain factor β and the shape factor $B(z)$ simultaneously. For audio-related problems, though, the generic function shown in Fig. 9 often works very well. As a function of frequency, the magnitude $|B|$ of $B(z)$ has a low-frequency asymptotic value B_L, and a high-frequency asymptotic value B_H. In the mid-frequency region, $|B|$ is one. B_L and B_H are usually much greater than one. The frequencies f_{L1}, f_{L2}, f_{H1}, and f_{H2} define the two transition bands. When the sampling frequency is high, 44.1kHz for example, it is sometimes advantegeous to design $|B|$ on a double-logarithmic scale since this is a good approximation to the way the ear perceives sound. For a sampling frequency of 44.1kHz, a suitable set of values for a loudspeaker equalization problem might be $B_L = 100$, $B_H = 100$, $f_{L1} = 40$Hz, $f_{L2} = 100$Hz, $f_{H1} = 12$kHz, and $f_{H1} = 16$kHz. Once $B(z)$ is known, there are plenty of methods one can use to determine β automatically. Since the main undesirable feature of the solution is likely to be sharp peaks in the magnitude response, one can try to adjust β such that a certain maximum value is not exceeded, or such that the peak-to-rms ratio is well-behaved within certain frequency bands. It is up to the user to specify a criterion that is appropriate for the application at hand.

VIII. CONCLUSIONS

The computational complexity of the fast deconvolution method is essentially that of the Fast Fourier Transform which is an $O(N \log N)$ algorithm where N is the number of coefficients in the optimal filters [27]. The method is easy to implement, numerically robust, and since the optimal respones at different frequencies are independent of each other it is possible to speed up the calculation even further by parallel execution.

The longest time constant of the optimal filters can be estimated by mapping out the zeros of the determinant of a matrix of z-transforms. Those zeros become the poles of a matrix of ideal optimal filters, and if any of those poles are too close to the unit circle, it can cause a wrap-around effect that corrupts the time response of the optimal filters. In order to avoid this, the optimal filters can be made longer, or alternatively their effective duration can be reduced by using regularisation.

The regularisation gain factor β and shape factor $B(z)$ are most reliably determined by trial-and-error experiments. It is possible to design black box routines that work well in most cases, but there is always an element of danger in using them.

Finally, it is worth mentioning that it is straightforward to relax the constraints on the target matrix **A**(z), the shape factor $B(z)$, and the plant transfer functions **C**(z) so that they can have infinite impulse responses as long as the optimal filters **H**(z) still have finite impulse responses. For example, if $B(z)$ is written $B_{\mathrm{FIR}}(z)/B_{\mathrm{IIR}}(z)$, fomal manipulations of the z-transforms still allow the optimal filters to be written in fractional form so that the pole-zero analysis and the fast deconvolution method can be applied.

REFERENCES

[1] R.A. Roberts and C.T. Mullis, *Digital Signal Processing*, Addison-Wesley, 1987

[2] M. Karjalainen, E. Piirilä, and J. Huopaniemi, “Comparison of loudspeaker equalization methods based on DSP techniques”, presented at the 102nd *Audio Eng. Soc. Convention* in Munich, March 22-25, 1997. AES preprint 4437 (F2)

[3] R. Greenfield and M.O. Hawksford, “Efficient filter design for loudspeaker equalization”, *J. Audio Eng. Soc.*, vol. 39, no. 10, pp. 739-751, 1991

[4] S.T. Neely and J.B. Allen, “Invertibility of a room impulse response”, *J. Acoust. Soc. Am.*, vol. 66, no. 1, pp. 165-169, 1979

[5] J. Mourjopoulos, “Digital equalization of room acoustics”, *J. Audio Eng. Soc.*, vol. 42, no. 11, pp. 884-900, 1994

[6] S.J. Elliot and P.A. Nelson, “Multiple point equalization in a room using adaptive digital filters”, *J. Audio Eng. Soc*., vol. 37, no. 11, pp. 899-907, 1989

[7] P.A. Nelson, H. Hamada, and S.J. Elliott, “Adaptive inverse filters for stereophonic sound reproduction”, *IEEE Trans. Signal Processing*, vol. 40, pp. 1621-1632, 1992

[8] J. Bauck and D.H. Cooper, “Generalized transaural stereo and applications”, *J. Audio Eng. Soc*., vol. 44, no. 9, pp. 683-705, 1996

[9] O. Kirkeby, P.A. Nelson, F. Orduna-Bustamante, and Hareo Hamada, “Local sound field reproduction using digital signal processing”, *J. Acoust. Soc. Am*., vol. 100, no. 3, pp. 1584-1593, 1996

[10] O. Kirkeby, P.A. Nelson, and H.Hamada, “Digital filter design for virtual source imaging systems”, presented at the 104th *Audio Eng. Soc. Convention* in Amsterdam, May 16-19, 1998. AES preprint 4688 (P1-3). Submitted to *J. Audio Eng. Soc*.

[11] O. Kirkeby, P.A. Nelson, H. Hamada, and F. Orduna-Bustamante, “Fast deconvolution of multichannel systems using regularization”, *IEEE Trans. Speech Audio Processing*, vol. 6, pp. 189-194, 1998

[12] W.H. Press, S.A. Teukolsky, W.T. Vetterling, and B.P. Flannery, *Numerical Recipes in C*, 2nd ed. Cambridge, U.K.: Cambridge Univ. Press, 1992

[13] P. Kraniauskas, *Transforms in Signals and Systems*. Reading, MA: Addison-Wesley, 1992

[14] M. Bellanger, *Digital Processing of Signals*, 2nd ed. John Wiley & Sons, 1989

[15] A.V. Oppenheim and R.W. Schafer, *Digital signal processing*. Englewood Cliffs, NJ: Prentice-Hall, 1975

[16] R.A. Horn and C.R. Johnson, *Matrix Analysis*. Cambridge University Press, 1985

[17] J.M. McNamee, “A bibliography on roots of polynomials”, *J. Comput. Appl. Math.*, vol. 47, pp. 391-394, 1993. New version available on the World Wide Web at http://www.elsevier.nl/locate/cam

[18] F.S. Acton, *Numerical Methods That Work*, 3rd printing, The Mathematical Association of America, 1990

[19] K. Steiglitz and B. Dickinson, “Phase unwrapping by factorization”, *IEEE Trans. Acoustics Speech and Signal Processing*., vol. ASSP-30, pp. 984-991, December 1982

[20] D.H. Lehmer, “A machine method for solving polynomial equations”, *J. Association Computing Machinery*, vol. 8, pp. 151-162, 1961

[21] Y. Bistritz, “Zero location with respect to the unit circle of discrete-time linear system polynomials”, *Proc. IEEE*, vol. 72, pp. 1131-1142, September 1984

[22] O. Kirkeby, P.A. Nelson, H. Hamada, “The “stereo dipole” - a virtual source imaging system using two closely spaced loudspeakers”, *J. Audio Eng. Soc*., vol. 46, no. 5, pp. 387-395, 1998

[23] O. Kirkeby, P.A. Nelson, and H. Hamada, “Local sound field reproduction using two closely spaced loudspeakers”, *J. Acoust. Soc. Am*., vol. 104, no. 4, pp. 1973-1981, 1998

[24] R.O. Duda and W.L. Martens, “Range dependence of the response of a spherical head model”, *J. Acoust. Soc. Am.*, vol. 104, no. 5, pp. 3048-3058, 1998

[25] K.B. Rasmussen and P.M. Juhl, “The effect of head shape on spectral stereo theory”, *J. Audio Eng. Soc.*, vol. 41, no. 3, pp. 135-141, 1993

[26] O. Kirkeby and P.A. Nelson, “Virtual source imaging using the stereo dipole”, presented at the 103rd *Audio Eng. Soc. Convention* in New York, September 26-29, 1997. AES preprint 4574 (J10)

[27] J. Gruska, *Foundations of Computing*, Thomson Computer Press, 1997

FIGURE CAPTIONS

Fig. 1 The discrete-time deconvolution problem in block diagram form

Fig. 2 The properties of the sequence $c(n)$ whose z-transform is $C(z)=1-0.96z^{-1}$. a) The zeros of $C(z)$ in the complex plane, and b) its magnitude response $|C|$. Fig. 2a is essentially a close-up of a thin strip that covers the unit circle

Fig. 3 The positions of the poles (crosses) and zeros (circles) in the complex plane of the ideal inverse $H(z)$ of $C(z)$ whose properties are shown in Fig. 2. $H(z)$ is calculated with a) frequency-independent regularisation, and b) frequency-dependent regularisation

Fig. 4 The magnitude response $|H|$ corresponding to the pole-zero maps plotted in Fig. 3. $H(z)$ is calculated with a) frequency-independent regularisation, and b) frequency-dependent regularisation. For reference, the dashed lines show $|H|$ calculated with no regularisation

Fig. 5 The design of a cross-talk cancellation network for the “stereo dipole” is an example of a multi-channel deconvolution problem

Fig. 6 The impulse responses of a) the direct path C_1 and b) the cross-talk path C_2 as defined in Fig. 6 when the listener’s head is modeled as a rigid sphere, and the sampling frequency is 44.1kHz

Fig. 7 The magnitude responses a) $|H_1|$ and b) $|H_2|$ of the two filters $H_1(z)$ and $H_2(z)$ necessary for implementing the cross-talk cancellation network shown in Fig. 6. The two frequency responses are calculated with frequency-dependent regularisation (solid lines) and no regularisation (dashed lines)

Fig. 8 The impulse responses of the two filters whose magnitude responses are shown with the dashed line in Fig. 7 calculated by the fast deconvolution method. Note that low-frequency boost required by the cross-talk cancellation network makes it necessary to use long filters (N_h = 1024) in order to prevent the wrap-around effect

Fig. 9 A suggested magnitude response function for the shape factor $|B|$ that often works well with audio-related deconvolution problems. When the sampling frequency is high, it is sometimes advantegeous to define $|B|$ to be piecewise linear on a double-logarithmic scale