A NEW TIME-DOMAIN DECONVOLUTION ALGORITHM AND ITS APPLICATIONS

T. Engin TUNCER
Middle East Technical Unv., EE Eng. Dept., Ankara, TURKEY
egn@rorqual.cc.metu.edu.tr

ABSTRACT

Recently a new time-domain method has been presented for deconvolution [1]. This multidimensional method completely eliminates the problems of the previous methods in one dimension and covers a reasonable part of the solutions in multidimensions. In this paper, we present some of the properties of this method. We will especially focus on the frequency domain behaviour of the algorithm as well as the performance under numerical errors and errors due to noise. In addition we will present examples of the applications including blind deconvolution with a modified NAS-RIF algorithm.

I. INTRODUCTION

Several researchers are interested on the deconvolution problem which is one of the most important problem of linear system theory. Deconvolution can be seen as an inverse filtering for removing the effects of linear convolution operator. There are several different approaches developed for deconvolution throughout the years. Wiener filtering [2], Fourier domain techniques [3], homomorphic deconvolution [4], and iterative approaches [5] are only a subset of currently available methods. Unfortunately almost all of these methods suffer from the problems of stability, division by zero, approximation and computational complexity. We have proposed a new time-domain deconvolution (TDD) algorithm [1] where all of these problems are solved.

This multidimensional TDD method results exact solution for one dimensional case without any of the problems associated with the previous methods. In multidimensional case, TDD method has always solutions when the convolution kernel satisfies det(h)=0 condition. Even when this condition is not satisfied, one can still solve the problem depending on the value of det(h) and the split factor which is related to the length of the input signal.

In this paper, we discuss several properties of the TDD method. We show that what TDD method does in time-domain corresponds to the pole-zero cancellation in Fourier domain with some appropriate spectral shaping. In addition, we will discuss how the spread of convolution kernel zeros and the split factor affect the noise behaviour of the TDD method. One interesting property of the TDD method is that there are several copies of the input signal at the output of deconvolution. The number of input images depends on the number of zeros of the convolution filter and the distance between images is equal to the split factor. It turns out that each input image has different noise sensitivity and it is always better to choose the input image, which shows up the first.

TDD method has been employed for both one and multidimensional signals. In addition to direct deconvolution when the output and convolution kernel are known, TDD method is also used for blind deconvolution problem. A modified non-negativity and support recursive inverse filter (NAS-RIF) structure is used with TDD block [6]. In blind deconvolution, input signal is required to be found without any knowledge on the convolution kernel.

Another interesting problem is the circular deconvolution. It turns out that circular deconvolution is possible if the circular Toeplitz matrix, H, is nonsingular. Note that singularity is not an issue for linear convolution and deconvolution is always possible for one-dimensional case.

There are several applications of the deconvolution operator [7]. Image restoration for medicine [8] and astronomy [6] employs different algorithms, all of which use a deconvolution operation. Blurring effects due to moving objects and focusing of the camera can be eliminated by deconvolution methods. Seismic signal processing also requires inverse filtering for the localisation of natural resources.

II. TDD METHOD

Time domain deconvolution method is based on simple algebraic identities, which permit exact deconvolution for first order kernels or filters. Deconvolution of higher order kernels is done by the combination of first order factors. In this respect, TDD method requires the factorisation of a convolution kernel into its first order terms. Ideally a deconvolution filter should satisfy $h(\mathbf{n}) \overset{D}{*} h_{c\_inv}(\mathbf{n})=\delta(\mathbf{n})$ where $\overset{D}{*}$ is D dimensional convolution. The solution for the above equation is ill conditioned since there are more equations than the unknowns. TDD method tries to find a deconvolution filter which satisfies,

$$h(\mathbf{n}) \overset{D}{*} h_{c\_inv}(\mathbf{n}) = \sum_{p_0=0}^{I} \cdots \sum_{p_{D-1}=0}^{I} d_{p_0,\dots,p_{D-1}} \delta(n_0-p_0, n_1-p_1, \dots, n_{D-1}-p_{D-1})$$

$$\mathrm{mod}_I\left(\sum_{i=0}^{D-1} p_i\right)=0 \qquad [1]$$

Additional terms on right hand side of equation (1) allow us to find exact solutions for deconvolution operation. As long as the sample splitting factor, I, is greater than the input length, exact copies of the input signal can be obtained. Note that finding a solution to a conditioned equation like (1) is not an easy task. Instead we will use an operation which is called as sample splitting for finding a solution for (1). Let $h_0(\mathbf{n})$ be a D dimensional first order convolution kernel. The deconvolution kernel, $h_{c\_inv,0}(\mathbf{n})$, can be found by a successive sample splitting operation defined as below.

$$\tilde{h}_0(\mathbf{n}) = \frac{(-1)^{(n_0+n_1+\dots+n_{D-1})}}{\left[\max\{Abs(h_0(\mathbf{n}))\}\right]^2} h_0(\mathbf{n}) \qquad [2]$$

$$\tilde{h}_k(\mathbf{n}) = (-1)^{na_k} h_k(\mathbf{n}), \qquad [3]$$

where $na_k = \begin{cases} \frac{n_0+n_1+\dots+n_{D-1}}{2^k} & n_0+n_1+\dots+n_{D-1}=m2^k \\ 0 & otherwise \end{cases}$

$$h_1(\mathbf{n}) = h_0(\mathbf{n}) \overset{D}{*} \tilde{h}_0(\mathbf{n})$$
$$h_k(\mathbf{n}) = h_{k-1}(\mathbf{n}) \overset{D}{*} \tilde{h}_{k-1}(\mathbf{n}) \qquad [4]$$

Sample splitting is continued until k satisfies,

$$k > Log_2(N_x D)-1 \qquad [5]$$

where N_x is the largest length of the input sequence. Then the deconvolution filter is obtained as,

$$h_{c\_inv,0}(\mathbf{n}) = \tilde{h}_0(\mathbf{n}) \overset{D}{*} \tilde{h}_1(\mathbf{n}) \overset{D}{*} \dots \overset{D}{*} \tilde{h}_K(\mathbf{n}) \qquad [6]$$

Here $h_{c\_inv,0}(\mathbf{n})$, is the deconvolution kernel for the first order filter $h(\mathbf{n})$. In order to choose a copy of the input sequence from the output of deconvolution, which has several images of the scaled input signal, we use the windowing operation,

$$x(\mathbf{n}) = \left[y(\mathbf{n}) \overset{D}{*} h_{c\_inv,0}(\mathbf{n})\right] \mathrm{Re}\,ct_{N_x}(\mathbf{n}) \qquad [7]$$

Above TDD method can be used for any type of convolution kernel (including the ones which have zeros on the unit circle) in one-dimensional case. If the first order convolution kernel satisfies the det(h)=0 condition, then it is possible to obtain exact deconvolution in multidimensions. It may still be possible to obtain exact deconvolution even when det(h)≠0 depending on the size of the input signal and the value of det(h).

III. PROPERTIES OF THE TDD METHOD

We will first show that deconvolution filter in TDD method can be obtained by a pole-zero cancellation. In this sense deconvolution is a pole-zero cancellation followed by some spectral shaping. For one dimensional case, convolution and deconvolution in z-domain can be written as,

$$Y(z) = X(z)H(z) \qquad [8]$$
$$D(z^I) = Y(z)H_{c\_inv}(z)$$

where $D(z^I)$ is a polynomial of z^I. Let X(z)=1. Then deconvolution filter, $H_{c\_inv}(z)$ will be

$$H_{c\_inv}(z) = \frac{\prod_{k\in M_1}(1-d_k{}^I z^{-I}) \prod_{k\in M_2} d_k{}^{-I}(1-d_k{}^I z^{-I})}{\prod_{k\in G_1}(1-c_k z^{-1}) \prod_{k\in G_2}(1-c_k z^{-1})} \qquad [9]$$

where M_1 and G_1 are the sets of minimum phase zeros and M_2 and G_2 are the sets of maximum phase zeros. If we choose $d_k=c_k$ ($M_1= G_1$, $M_2= G_2$) and apply the pole-zero cancellation, we have

$$H_{c\_inv}(z) = \prod_{k\in M_1}(1+d_k z^{-1}+\dots+d_k{}^{(I-1)} z^{-(I-1)}) \qquad [10]$$
$$\prod_{k\in M_2} d_k{}^{-I}(1+d_k z^{-1}+\dots+d_k{}^{(I-1)} z^{-(I-1)})$$

Above deconvolution filter is always stable and causal.

Noise sensitivity of the TDD method depends on several properties. The numerical error noise sensitivity is the noise sensitivity when there is no noise on both output y(n) and convolution kernel h(n) sequences. Noise sensitivity of the TDD method in this case depends on:

a) The location of the zeros on the unit circle.
b) The value of the split factor, I.
c) Which copy of the input sequence, x(n), will be chosen.
d) How accurate the zeros of h(n) are found.

a) *Location of the zeros*: If the zeros on the unit circle get close to each other, noise sensitivity increases. The zeros of the convolution kernel outside the unit circle have little effect on the noise sensitivity.
b) *Split factor, I*: Noise sensitivity increases with the split factor, I. In addition noise sensitivity shows a cyclic behaviour. The period, P, of this cyclic behaviour is approximately equal to,

$$P = \frac{\pi}{\pi - \theta} N_p$$

where θ is the angle of the zero on unit circle and N_p is the smallest integer that makes P as an integer value.

c) *Copy of the input sequence*: Noise sensitivity depends on which copy of the input sequence is chosen from the deconvolution output. The copy of x(n) that corresponds to the first d(n) coefficient has the best noise quality. This copy of x(n) at the deconvolution output requires the least number of arithmetical operations, where some of them should exactly add up to zero. This is why the first copy of x(n) has the smallest numerical noise.

d) *Zero finding accuracy*: Noise sensitivity of the TDD method depends on how accurately we find the zeros of the convolution kernel. Pole-zero cancellation is affected by this accuracy.

IV. APPLICATIONS OF THE TDD METHOD

Applications of deconvolution operator can be divided into two main groups, namely direct and blind deconvolution. Direct deconvolution is related to the problem of finding the input signal when the system output and convolution kernel are known. Blind deconvolution is used when the convolution kernel is unknown. This is an iterative procedure in contrast to one step operation of direct deconvolution. In this paper, we will give examples for both direct and blind deconvolution.

Example 1:

In this example, we will show the performance of the TDD method for noisy observations. The convolution kernel is chosen with zeros at z_0=1.0, $z_{1,2}$=0.27±j0.51, z_3=-1.72, and $z_{1,2}$=0.9±j2.17. Figure 1.a shows the input signal. Convolution output with a SNR of 34.6 dB after a zero mean white noise is added is shown in Figure 1.b. Deconvolution filter, $h_{c\_inv}(n)$, is given in Figure 1.c. The result of the deconvolution is presented in Figure 1.d. SNR for the reconstructed input sequence is 24.72 dB on the average for 30 trial runs.

Example 2:

In this example, we used a modified version of the NAS-RIF structure for blind deconvolution. Figure 2 shows the modified NAS-RIF structure. In this modified structure, deconvolution filter length depends on the input image size. But optimisation is done on the convolution filter coefficients rather than the deconvolution filter. Input is a 32 × 32 8 bit image of heart and lungs obtained from electrical impedance tomography. Blurring point spread function is assumed to be a first order nonseparable filter. Figure 3.a. is the original image. Figure 3.b is obtained after several iterations by manual intervention to the filter coefficient optimisation routine. Figure 3.c shows the result of the modified NAS-RIF algorithm after 20 iterations. Note that there are some structures, which are not clearly seen, in the original image in both b and c.

V. CONCLUSION

In this paper, properties of the recently proposed multidimensional deconvolution method have been investigated. This new method gives exact solutions for any convolution kernel in one dimensional case and covers a set of possible solutions in multidimensions. It turns out that this time-domain method is closely related to pole-zero cancellation with an additional spectral shaping. Noise sensitivity of the TDD method increases when the zeros on the unit circle get closer. It also depends on the accuracy of the factorisation of the convolution kernel and the value of the split factor, I. In addition to these, the first copy of the input from the several other copies at the deconvolution output is the best in terms of noise sensitivity. We have given different examples of the applications of the TDD method. TDD method is robust to additive noise and it can be successfully used in blind deconvolution.

VI. REFERENCES

[1] T.Engin Tuncer, " A new method for D-dimensional exact deconvolution", to be published in IEEE Trans. on Signal Proc.

[2] J.Lim, A.V.Oppenheim, ''Advanced Topics in Signal Processing'', Prentice Hall, 1988.

[3] D.Zazula, L.Gyergyek, ''Direct frequency domain deconvolution when the signals have no spectral inverse'', IEEE Trans. on Signal Proc., Vol. 41, No.2, pp.977-981, Feb. 1993.

[4] Jae Lim, ''Spectral root homomorphic deconvolution system'', IEEE Trans. on ASSP-27, No.3, pp.223-233, June 1979.

[5] C. Sanchez-Avila,''An adaptive regularized method for deconvolution of signals with edges by convex projections'', IEEE Trans. on Signal Proc., Vol.42, pp.1849-1851, July 1994.

[6] D. Kundur, D. Hatzinakos, " Blind image deconvolution", IEEE Signal Proc. Mag., pp.43-64, May 1996.

[7] M.R. Banham, A.K.Katsaggelos, "Digital Image Restoration", IEEE Signal Proc. Mag., pp. 24-41, March 1997.

[8] J.A. Goyette, G.D. Lapin, M.G.Kang, A.K. Katsaggelos, ''Improving autoradiography resolution using image restoration techniques'', IEEE Engin. Medicine Biology, pp. 571-574, Aug. 1994.

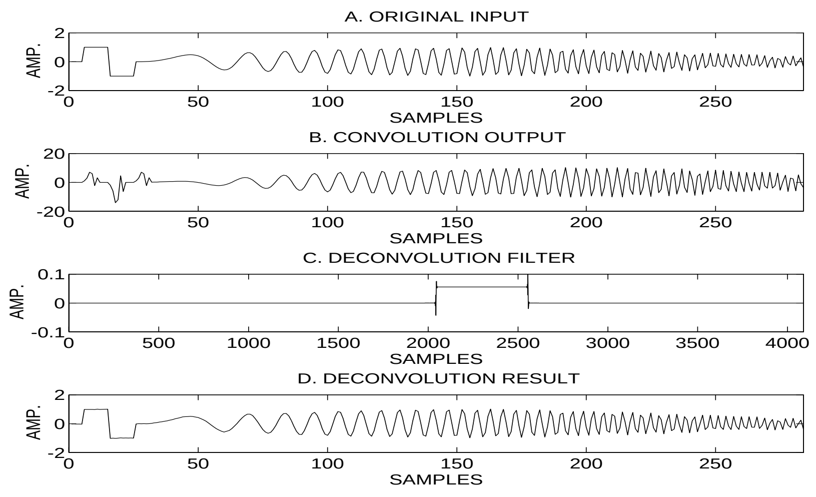

Figure 1. One dimensional deconvolution with additive noise.

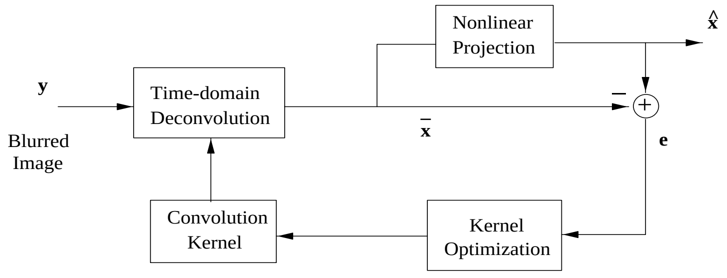

Figure 2. Modified NAS-RIF structure for blind deconvolution.

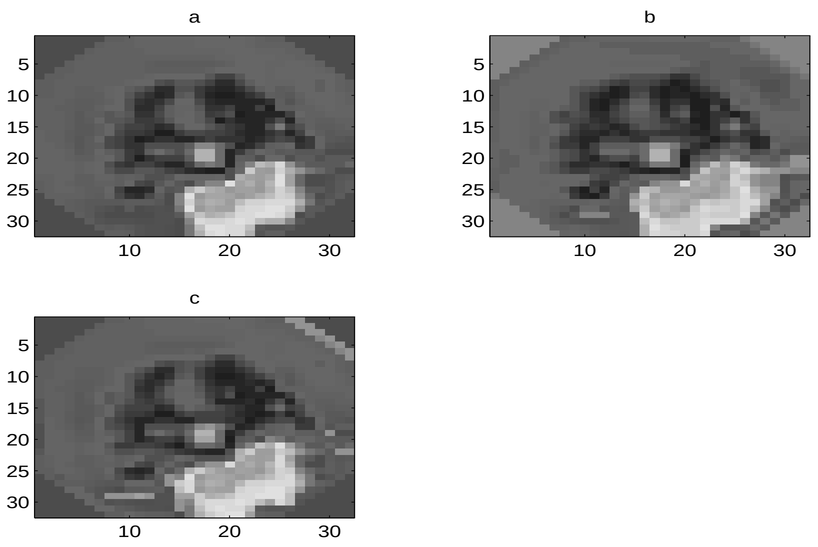

Figure 3. Blind deconvolution with modified NAS-RIF structure. a. Original image, b.image optimized subjectively, c. image obtained with NAS-RIF algorithm after 20 iterations.