ULTRA FAST FOURIER TRANSFORM: A MRS CASE OF STUDY

P. A. ANGELIDIS

School of Electrical Engineering, Aristotle University of Thessaloniki, Greece

INTRODUCTION

The Fourier transform (FT) plays a key role in magnetic resonance spectroscopy (MRS). The acquired sampled Free Induction Decays (FIDs) are usually subjected to a magnitude discrete Fourier transform (DFT). The DFT sum of a sequence x[n] of length N ($n \in Z$, $0 \le n < N$) is defined by

$$X_f(\omega) = \sum_{n=0}^{N-1} f(n\omega)\, x[n], \qquad 0 \le \omega < 1 \tag{1}$$

where with $f(\omega)$ we denote the complex exponential function (cef) $e^{j2\pi\omega}$. Usually, the DFT is computed at N equidistant frequency points, thus resulting to the following, more familiar definition

$$X[k] = \sum_{n=0}^{N-1} W_N^{nk} x[n], \qquad k \in Z,\ 0 \le k < N \tag{2}$$

where

$$W_N = e^{-j2\pi/N}$$

and

$$\omega = k/N \tag{3}$$

i.e. ω is discretised in N equidistant points.

The sequence X[k] constitutes a perfect sampling sequence of the FT of the continuous FID as long as the Nyquist criterion is satisfied during the acquisition.

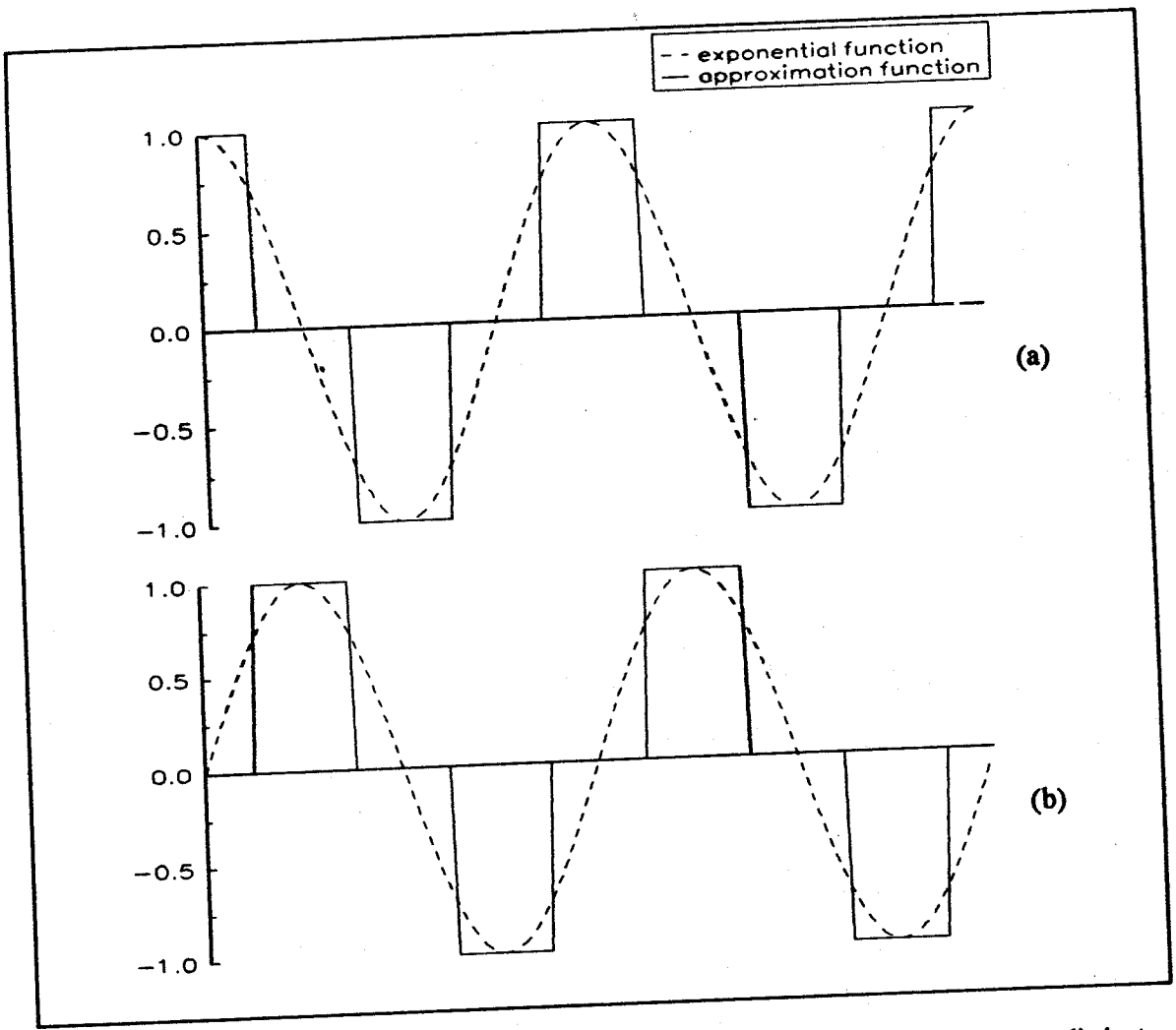
Almost always the DFT is computed using the well-known FFT algorithm, which reduces the computational load from $O(N^2)$ to $O(N\log_2 N)$. In order to further reduce the reconstruction time one can approximate the cef $f(\omega)$ by a more efficient, from a computational point of view, periodic function. This way the computational effort will be lessen, but in the cost of introducing artifacts in the final result. However, in cases where a very fast general overview of the spectrum is desired, such an approach could be useful.

We examine here the performance of the DFT when it is computed by approximating the cefs f(k/N) with a periodic function that takes only four values, namely 1, -1, j and -j. This approximation is shown schematically in fig.1. This is the crudest approximation one can think of. The error terms that it introduces are weighted aliased harmonic components, whose amplitude decreases as the number of

Address of correspondance: P. Angelidis Stefanou Nouka 20 Harilaou GR-54250 Thessaloniki, GREECE
Tel: +30-31-304305 Fax: +30-31-426667 E_mail: page@vergina.eng.auth.gr

the points of the approximation function increases. Naturally, an M-point approximation (M>4) gives a result closer to the actual FT. On the other hand, the four-values approximation function is the best as for the computational time it calls for, because it does not require any multiplication at all. The multiplications in the DFT defining expression (2) become trivial; they completely vanish or become a mere sign change or a real-to-imaginary swap.

Alternative time saving strategies, for example integer approximations, have also been proposed, but they offer lower speed advantages. This four point approximation transform appears to be the fastest solution and has been successfully implemented as a real-time DTMF tone detection scheme [1].



Approximation of a complex exponential function with a simpler function taking only four distinct values and having the same period: (a) real and (b) imaginary part. Two periods of the functions are shown.

Figure 1

ANALYSIS

Let

$$X_g(\omega) = \sum_{n=0}^{N-1} g(n\omega)\, x[n] \tag{4}$$

be the transform computed when approximating the cef f(ω) by a periodic function g(ω). The last, being periodic, it can be expanded as a Fourier series [2] and be written as

$$g(\omega) = \sum_m a_m e^{j2\pi m\omega} \tag{5}$$

where the parameters a_m are the Fourier series coefficients.

Inserting (5) to (4) results to

$$X_g(\omega) = \sum_{n=0}^{N-1} \sum_m a_m e^{j2\pi mn\omega}\, x[n] \tag{6}$$

Switching the order of summation and taking into account eq.(1) one finds

$$X_g(\omega) = \sum_m a_m X_f(m\omega) \tag{7}$$

Thus, G(ω) is a good approximation of the DFT of the sequence x[n], F(ω), as long as a_0 is close to one and all a_m for m≠0 are near zero. In fact for F(ω) holds exactly that

$$a_m = \begin{cases} 1 & \text{for } m = 0 \\ 0 & \text{otherwise} \end{cases}$$

The error terms $a_m F(m\omega)$ are easily identified as aliased harmonics of the basic frequency ω. Following (3), we set the four-values approximation function equal to

$$g_4(\omega) = e^{j2\pi k/4} = e^{j\pi k/2} \tag{8}$$

Since 0≤k‹4, it is obvious that the function $g_4(\omega)$ takes only the four values 1, -1, j and -j. Then, it turns out easily that

$$a_m = \begin{cases} \dfrac{(-1)^l}{m} \operatorname{sinc}(\pi/N) & \text{if } m = 4l + 1,\ l \in Z \\ 0 & \text{otherwise} \end{cases} \tag{9}$$

Thus, eq. (7) becomes

$$X_s(\omega) = \frac{2\sqrt{2}}{\pi}\left(X_f(\omega) + \frac{1}{3}X_f(-3\omega) - \frac{1}{5}X_f(5\omega) - \frac{1}{7}X_f(-7\omega) + \frac{1}{9}X_f(9\omega)\cdots \right) \quad (10)$$

A more general formula can be derived by replacing 4 in eq. (8) and (9) by M. Equation (10) shows that the transform in (7) for the approximation function $g_4(\omega)$ in (8), yields a scaled version of the FT of the sequence x[n], plus erroneous harmonics which become less important as their frequency increases, in the least possible time. Due to its speed, we call this transform ultra fast Fourier transform (UFFT).

Table 1 summarizes the differences in computing time between the FFT and the UFFT algorithms for sequences of various lengths and two different processing platforms. The measurements were carried out only for the main multiplication/addition body of the calculation. The additional time required for the cosine/sine calculation, which further lengthens the FFT total duration, was not included in the comparison. The results show that the time gained is very important, especially for two dimensional (2D) sequences. The total time consumed by the UFFT is less than half the time required by the FFT.

Sequence (samples)	FFT computing time (sec)		UFFT computing time (sec)		Difference (sec)	
	386*	486^	386*	486^	386*	486^
128	1.593	0	0.714	0	0.879	0
512	8.076	0.055	3.846	0	4.230	0.055
4096	86.868	0.165	41.593	0.055	45.275	0.110
128x128	232.143	0.439	109.780	0.165	122.363	0.274
256x256	1040.47	1.813	492.637	0.879	547.835	0.934

Comparison of the computing time between the FFT and the UFFT algorithms for two different processing platforms

* 80386-SX/20MHz processor without numerical co-processor

^ 80486-DX/66 MHz processor

Table 1

APPLICATION

A function in C which computes the ultra fast DFT (uf__dft) is given in fig.2. This function assumes that a complex identifier type exists consisting of two double reals, corresponding to the real and imaginary parts of the complex number. It also assumes access to two functions, namely comadd, which performs an addition of two complex variables and comjadd, which also performs an addition of two complex variables, having previously switch the real and imaginary parts of the second. The uf__dft function receives three arguments; a pointer to the complex array of the raw FID data (x), a pointer to the complex array where the result will be stored (y) and an integer denoting the length of the two sequences (N). It computes the UFFT using the definitions in (4) and (8). This is the simplest approach. Naturally, a butterfly

computational scheme, similar to the one utilized by the FFT [3], may be implemented.

We applied the algorithm to a phosphorous FID of a phantom solution of inorganic phosphate (Pi), pyrophosphate (PPi) and adenosine-triphosphate (ATP), obtained in an 4.7T OTSUKA VIVOSPEC spectrometer, located at Panum Instittutet, Copenhagen, Denmark. The data sequence is two thousand forty-eight samples long, acquired using a sampling period of 100 μs which results to a frequency resolution of 4.88 Hz (0.06 ppm).

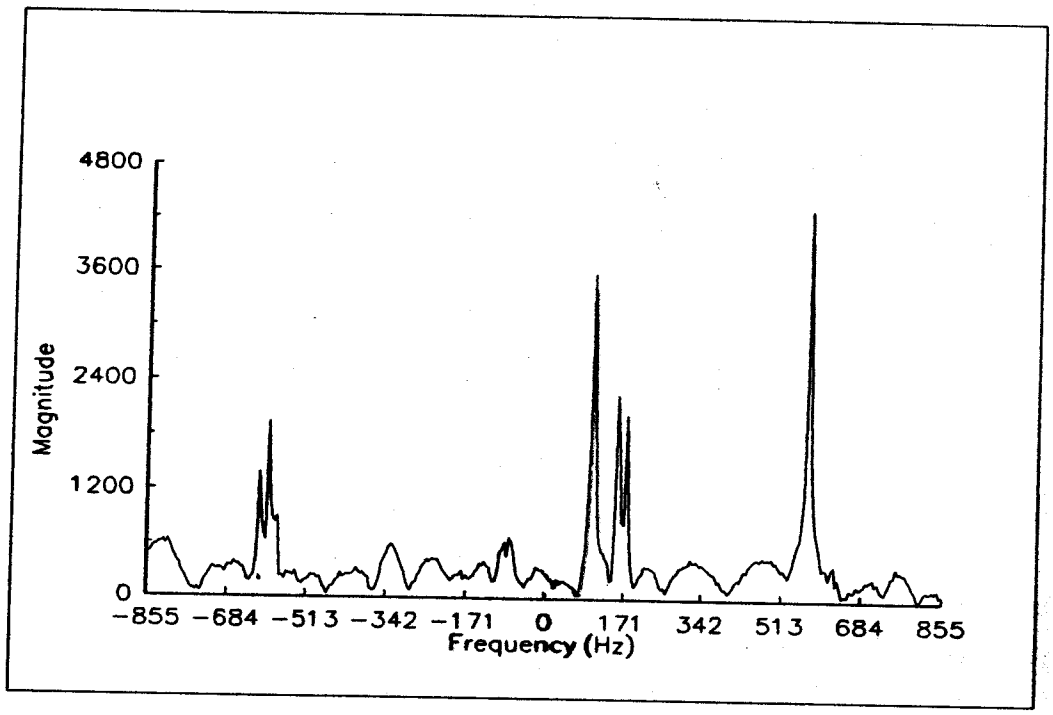
```
uf__dft(x, y, N)
        complex x,y;
        unsigned N;
{
        unsigned i,k;
        int f;
        double w,ws,ww;
        complex sum;

        ws=1./(double)N;
        w=0.;
        for(i=0;i<N;i++) {
                sum=compeq(0.,0.);
                ww=0;
                for(k=0;k<N;k++) {
                        f=(int)(ww*8.);
                        if(f>7) f-=8;
                        switch(f) {
                                case -1:
                                    0:  sum=comadd(sum,x[k]);
                                        break;
                                case 1:
                                    2:  sum=comjadd(sum,x[k]);
                                        break;
                                case 3:
                                    4:  sum=comadd(sum,-x[k]);
                                        break;
                                case 5:
                                    6:  sum=comjadd(sum,-x[k]);
                                        break;
                        }
                        ww+=w;
                }
                y[i]=sum;
                w+=ws;
        }
}
```

A C routine that computes the UFFT

Figure 2

Figure 3 shows the spectrum of the data calculated using a magnitude-FFT algorithm (it is shown only the part from -855 Hz to 855 Hz , which embodies all the useful information). The magnitude values of the nine peaks of interest (two singlets, two doublets and one triplet) are given in Table 2 (second column). Figure 4 shows the spectrum of the data calculated using the magnitude-UFFT algorithm of fig. 2 (same

region). Observe the similarity of the two plots. The original spectrum pattern profile has been revealed by the UFFT, with all nine peaks present at their correct locations. Their magnitude values are given in Table 2 (third column), normalized according to eq. (10).



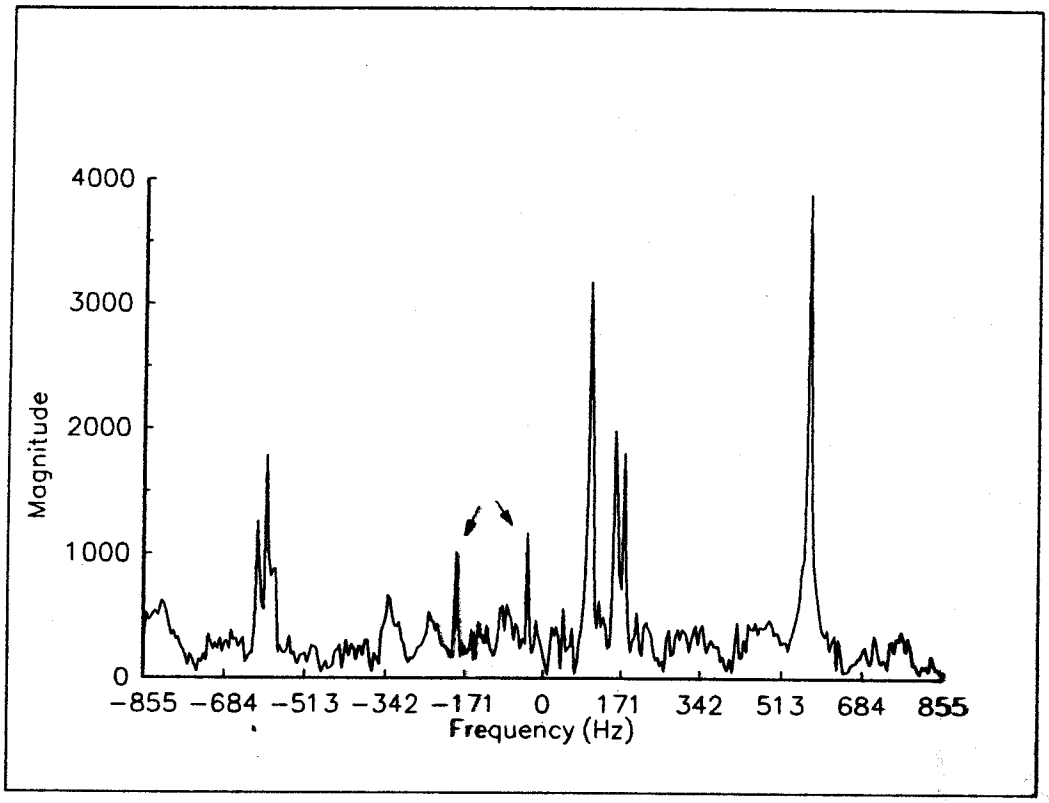
The magnitude-FFT of the FID data

Figure 3

Frequency (Hz)	Peak	Magnitude-FFT	Magnitude-UFFT	Estimation error (%)
-615.23	ATP_β	1389.55	1390.53	0.07
-595.70	ATP_β	1939.43	1973.76	-1.77
-576.17	ATP_β	896.39	971.06	-7.69
-87.89	ATP_α	621.99	646.30	-3.90
-78.12	ATP_α	674.36	654.38	2.96
102.54	PP_i	3577.06	3520.32	1.59
156.25	ATP_γ	2250.30	2193.46	2.53
175.78	ATP_γ	2032.68	1997.56	1.73
571.29	P_i	4315.06	4315.60	0.01

Comparison of the results between the magnitude values obtained by the FFT and the UFFT

Table 2



The magnitude-UFFT of the FID data. The arrows indicate the erroneous peaks. All true peaks appear at their correct positions and have magnitude values slightly deviating from those computed by the FFT (see text).

Figure 4

Note also the two small peaks, marked with arrows in fig. 4, which are aliased erroneous terms appearing at frequencies -190 Hz and -34 Hz. They are easily identified as false peaks because of their non-damping exponential nature and their position within the spectrum. A spectroscopist with some experience is able to directly point them out, while simply inspecting the spectrum. Furthermore, they can be numerically identified by calculating their width (i.e. their normal deviation). A true peak will show significantly less sharp decaying rhythm (i.e. it will be wider) than a false one. Table 3 summarizes such comparative results between the two false peaks and representative true ones. They confirm the above claim. In practice, such calculations would be necessary only if there is an uncertainty on the nature of a given peak. The rest of the erroneous aliased terms are hidden by the noise and they do not distort at all the pattern profile of the spectrum.

In the last column of Table 2 the errors in the estimation of the peak values, which are due to the UFFT approximations, are given. These demonstrate that the UFFT may also be used for an initial preliminary quantitative analysis of FID data. Observe that the estimation errors are quite small (less than 4% in all but one (7.7%) case). It is interesting to note that such peak estimation errors are less severe than the ones observed in the results of other non-FFT techniques, as are for example the parametric approaches in [4], [5], which indicates that the deviations encountered in the UFFT workspace are usually acceptable within a typical quantitative estimation frame.

Frequency (Hz)	Peak	Unit Amplitude Decay per Hz
-576.17	ATP_β	0.0514
-190	False	0.0839
-78.12	ATP_α	0.0201
-34	False	0.0831
102.54	PP_i	0.0263
156.25	ATP_γ	0.0586
571.29	P_i	0.0236

Comparison of the decaying rhythm between the false and true peaks in the UFFT spectrum

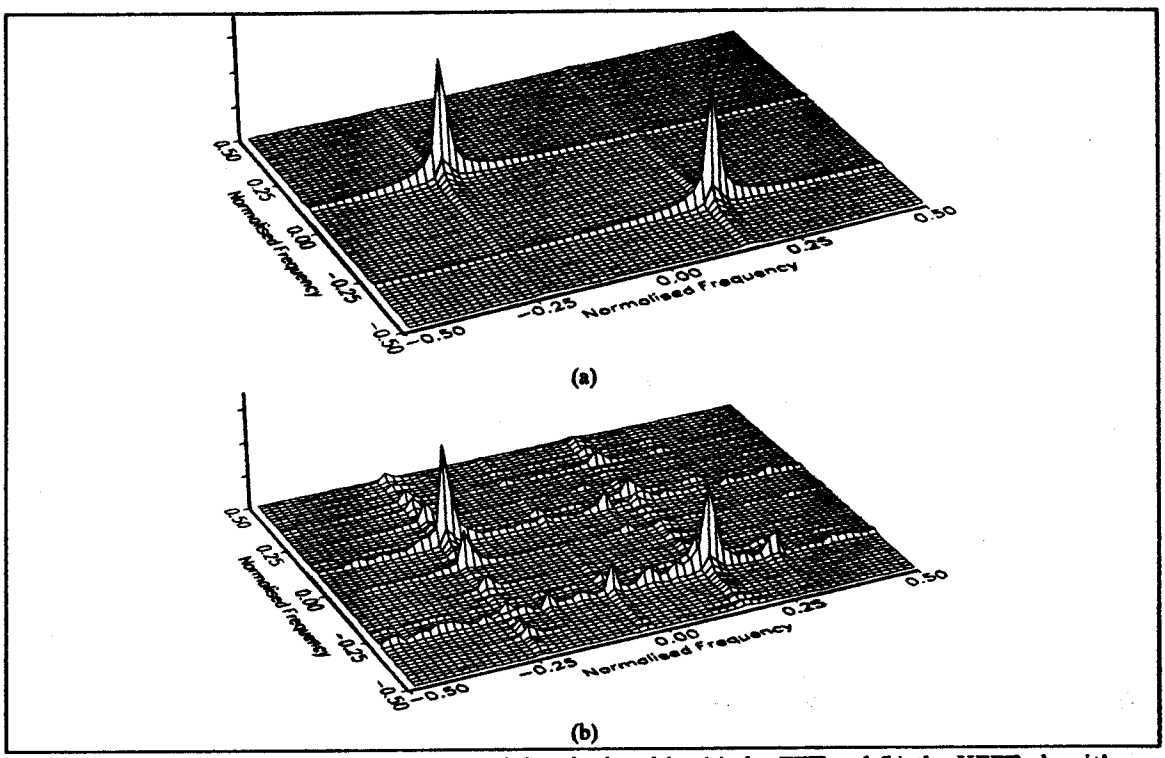
Table 3

We then applied the algorithm to a 2D simulation signal consisting of two damping exponentials plus additive white Gaussian noise, described mathematically by

$$f(t_x, t_y) = \sum_{i=1}^{2} A_i e^{j(\omega_{xi} t_x + \omega_{yi} t_y)} e^{-t/T_{2i}} + n(t)$$

and having a Signal-to-Noise ratio (SNR) of -10 db. Figure 5a shows the spectrum of the noiseless data calculated by the FFT. Observe the broadening of the peaks due to the decaying nature of the resonances. Figure 5b shows the spectrum that was calculated by the UFFT algorithm. We applied the routine in fig. 2 in both dimensions, utilizing a row-column decomposition scheme. The corresponding results for the noisy data are shown in fig. 6. Note in fig. 6b that the false harmonics in the UFFT spectrum appear to be masked by the noise. The FFT and UFFT patterns match almost perfectly above the noise level. This once more verifies the importance of the UFFT in qualitative spectrum information extraction. It is fast, reliable concerning the location of the resonances and also indicative of their amplitudes. The UFFT spectrum complies with the profile an expert expects to see in a practical situation.

The above results suggest that the UFFT may be successfully applied in MRI, where quantitative measurements are rarely carried out. There, the relative values of the magnitude spectrum of the data, i.e. the image values, are of importance, rather than the absolute ones. Under this view the UFFT is expected to provide a quick means to reconstruct MR images, thus allowing fast overviews of long series of images, which often are needed in diagnostic MR investigations. Efforts are under evaluation, to analyze the multidimensional UFFT and test its performance in MR image reconstruction.

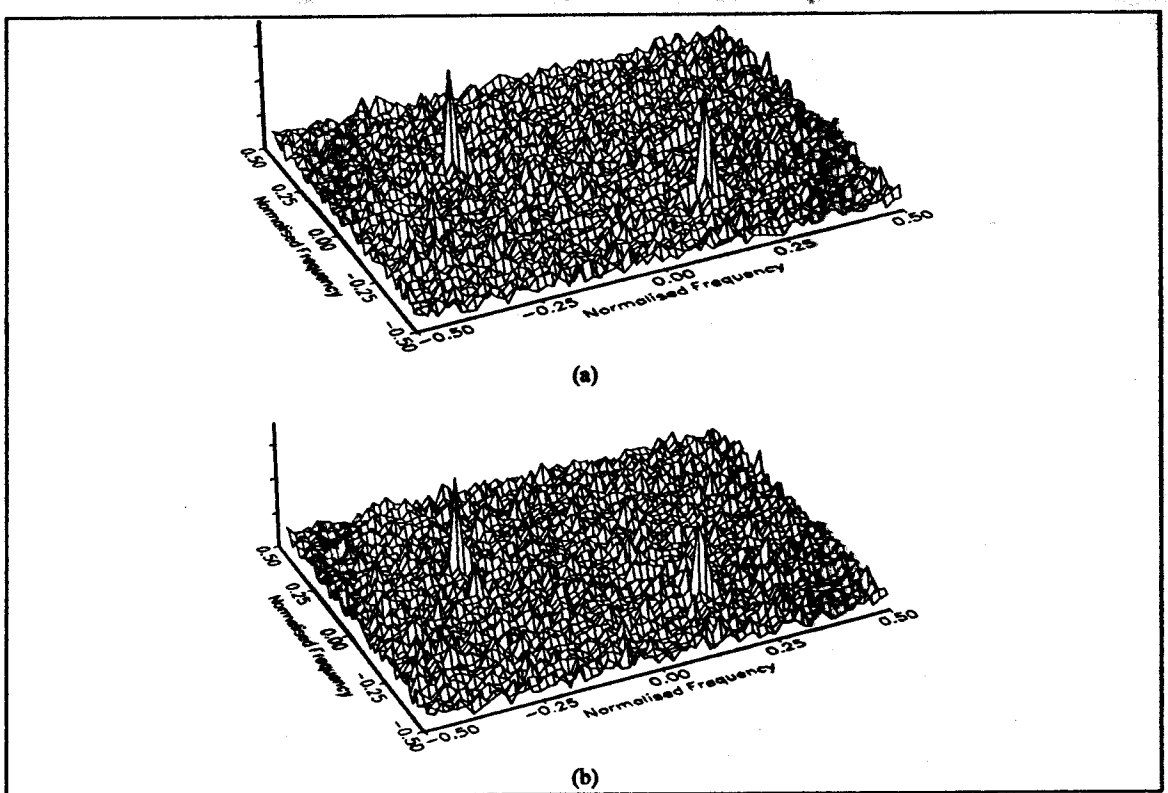


(a)

(b)

The 2D spectrum of two damping exponentials calculated by (a) the FFT and (b) the UFFT algorithms (Noiseless case)

Figure 5



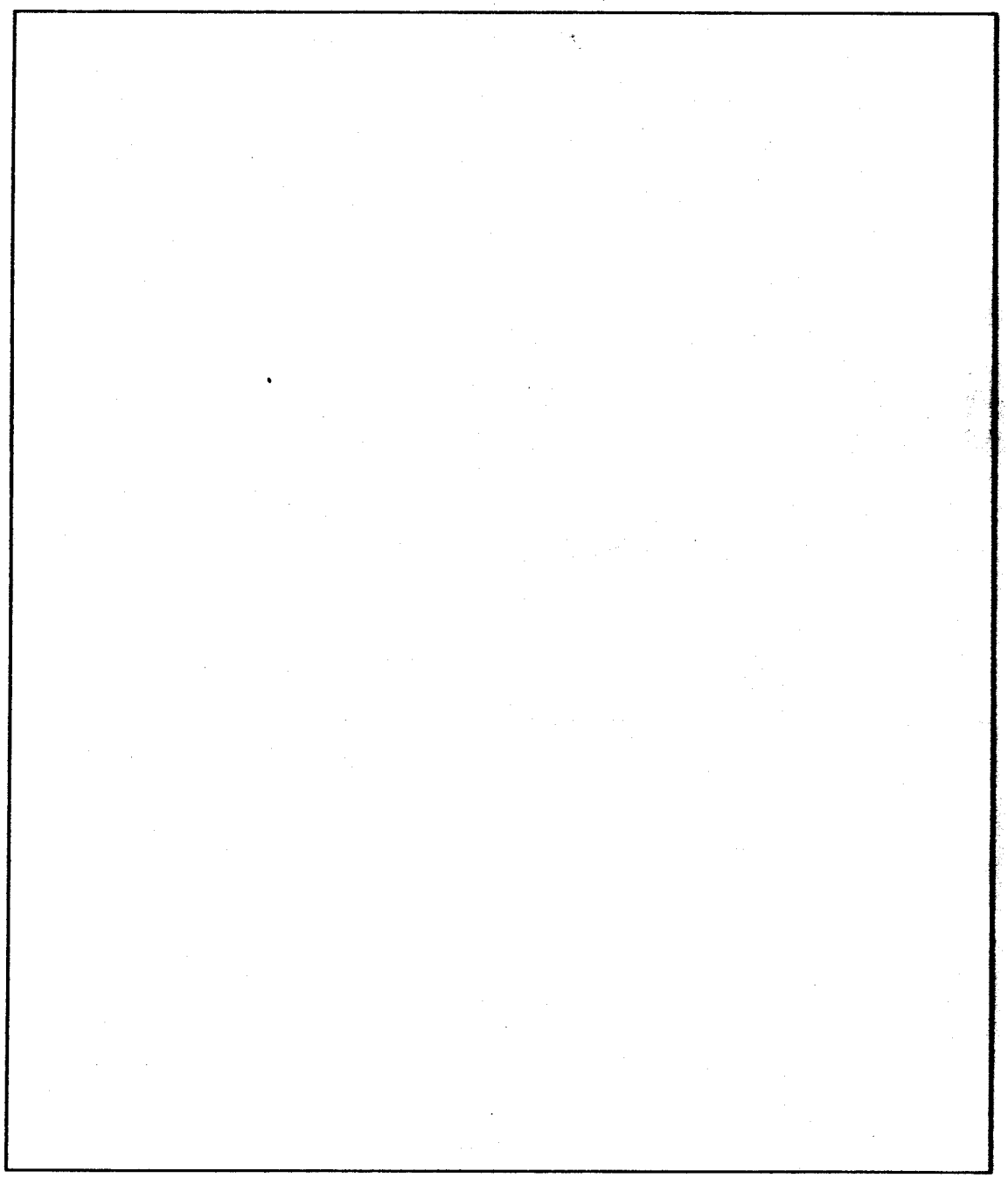
(a)

(b)

The 2D spectrum of two damping exponentials calculated by (a) the FFT and (b) the UFFT algorithms (Noisy data)

Figure 6

Figure 7 presents an initial example of the reconstruction of an MR image of two uniform tubes (standard spin-echo, 128x128 MRI obtained at 4.7T). One of the tubes was filled with tap water, while the other with olive oil. The FFT image reconstruction (fig. 7; top) shows two circles of uniform density, each at a different gray level. On the other hand, in the UFFT image reconstruction (fig. 7; bottom) the tubes do not appear to be uniform. Observe the gray level variation, which is the effect of the UFFT approximation. However, the structural information of the image is correctly revealed. This suggests that the UFFT will possibly find application in certain MR experiments, especially in 3D techniques, where the time savings are high.



Application of the UFFT to MR image reconstruction; top: the FFT image, bottom: the UFFT one

Figure 7

CONCLUSION

We developed an algorithm, which we call the ultra fast Fourier transform for computing the spectrum of a sequence very quickly, using only additions. This is achieved by approximating the complex exponential functions involved in the computation of the Fourier transform by periodic functions which take only the four values 1, -1, j and -j.

The analysis shows that this approximation introduces error terms, which are aliased harmonics of the actual frequency components present. The results demonstrate that this transform may be used in quantitative MRS as a fast replacement of the FFT, when an uncertainty in the order of 5% is tolerated. They also suggest that a multidimensional version of the transform may become useful for MR image reconstruction. An preliminary example is presented. The application of the UFFT to MRI will be discussed in a future paper.

ACKNOWLEDGMENTS

We thank Ib Therkelsen from Panum Institutet for providing the FID data.
We thank E. Kaldoudi and Dr. S.C.R. Williams from Queen Mary's College for providing the MRI data.

REFERENCES

1. M. P. Lamoureux, The Poorman's Transform, *IEEE Trans. Sign. Proc.*, vol.41, pp.1413-1415, March 1993.
2. R. N. Bracewell, "The Fourier Transform and its Applications", Mc-Graw Hill, New York, 1965.
3. A. V. Oppenheim and R. W. Schafer, "Digital Signal Processing", Prentice-Hall, Englewood Cliffs, New Jersey, 1975.
4. P. Barone et al., Prony-Burg Method for NMR Spectral Analysis, *J. Magn. Res.*, vol.73, pp.23-33, 1987.
5. K. P. Whittall, Quantitative Interpretation of NMR Relaxation Data, *J. Magn. Res.*, vol.84, pp.134-152, 1989.