Application of momentary Fourier transform to SAR processing

S.Albrecht and I.Cumming

Abstract: A common technique in signal and image processing is to extract a portion of the signal by windowing, and then perform the DFT on the window contents. The momentary Fourier transform (MFT) applies to the particular case where the window is moved one data sample along the signal between successive transforms. An alternative derivation of the recursive form of the MFT using general matrix transforms is given. How DFTs and IDFTs are used in the SPECAN and SIFFT methods of synthetic aperture radar (SAR) processing is described. The MFT and inverse MFT are applied to these methods and the advantages and disadvantages they have compared to the FFT/IFFT algorithms are shown.

1 Introduction

The discrete Fourier transform (DFT) is a widely used tool in signal or image processing and its efficiency is important. There are applications where it is desirable to use relatively small, successive, overlapped DFTs to obtain the spectrum coefficients. The momentary Fourier transform (MFT) computes the DFT of a discrete-time sequence for every new sample in an efficient recursive form. In this paper we give an alternative derivation of the MFT using the momentary matrix transform (MMT). Recursive and nonrecursive forms of the inverse MFT are also given, which can provide efficient frequency domain manipulation (e.g. filtering).

Uses of the incremental DFT were introduced by Papoulis in 1977 [1], and by Bitmead and Anderson in 1981 [2]. A detailed derivation of the momentary Fourier transform was given by Dudás in 1986 [3]. In 1991, Lilly gives a similar derivation, using the term 'moving Fourier transform', and uses the MFT for updating the model of a time-varying system [4].

After discussing the properties and arithmetic requirements of the MFT we investigate the applicability of the MFT to synthetic aperture radar (SAR) signal processing. In particular, we show what advantages the MFT algorithm offers to the SPECtral ANalysis (SPECAN) method and burst-mode SAR processing. In the SPECAN algorithm, received signals are multiplied by a reference function in the time domain, and overlapped short-length DFTs are used to focus the data. The azimuth FM rate of the signal varies in each range cell, which leads to the issue of keeping the azimuth resolution and output sampling rate constant. After the introduction to SPECAN, we show what advantages and disadvantages the MFT has compared to FFT algorithms.

When a SAR system is operated in burst mode, its azimuth received signal has a segmented frequency–time energy in its Doppler history. It requires that inverse DFTs (IDFTs) be located at specific points in the frequency domain to perform the azimuth signal compression. After the introduction of the burst-mode data properties, we show why the short IFFT (SIFFT) algorithm has the requirement of arbitrary-length, highly-overlapped IDFTs to process burst-mode data, in which case the IMFT is shown to have computational advantages.

2 Theory of momentary matrix transformation

2.1 Recursive momentary matrix transformation

Let x_i be a sample of an arbitrary complex-valued sequence of one variable. The sequence will be analysed through an N-point window, ending at the current sample i. In subsequent analyses the window will be advanced one sample at a time. At time i, sample x_i enters the window, while x_{i-N} leaves the window. At samples $i-1$ and i, the windowed function can be represented by the following two column vectors:

$$\boldsymbol{x}_{i-1} = \begin{bmatrix} x_{i-N} \\ \cdot \\ \cdot \\ x_{i-2} \\ x_{i-1} \end{bmatrix}, \qquad \boldsymbol{x}_i = \begin{bmatrix} x_{i-(N-1)} \\ \cdot \\ \cdot \\ x_{i-1} \\ x_i \end{bmatrix} \tag{1}$$

Let $\boldsymbol{T}$ be an $N \times N$ nonsingular transformation matrix, which has the inverse $\boldsymbol{T}^{-1}$. The sequence of windowed vectors can be *transformed* by $\boldsymbol{T}$ at each sample

$$\ldots\ \boldsymbol{y}_{i-1} = \boldsymbol{T}\boldsymbol{x}_{i-1}, \quad \boldsymbol{y}_i = \boldsymbol{T}\boldsymbol{x}_i, \ \ldots \tag{2}$$

Let $\boldsymbol{P}$ be the $N \times N$ elementary cyclic permutational matrix

$$\boldsymbol{P} = \begin{bmatrix} 0 & 1 & 0 & \cdot & 0 \\ \cdot & 0 & 1 & 0 & \cdot \\ \cdot & \cdot & 0 & 1 & 0 \\ 0 & \cdot & \cdot & 0 & 1 \\ 1 & 0 & \cdot & \cdot & 0 \end{bmatrix} \tag{3}$$

IEE Proceedings online no. 19990777

DOI: 10.1049/ip-rsn:19990777

Paper first received 9th December 1998 and in revised form 11th August 1999

The authors are with the Department of Electrical and Computer Engineering, The University of British Columbia, Vancouver, BC, Canada V6T 1Z4

S. Albrecht is currently with the Traffic Analysis and Network Performance Laboratory, Ericsson Hungary, 1037 Budapest, Hungary

I. Cumming is on sabbatical at DLR/DFD, Postfach 1116, D-82234 Wessling, Germany until 31st August 2000

When the vector $\boldsymbol{x}_{i-1}$ is premultiplying by $\boldsymbol{P}$, a one-element circular shift is performed, such that the index of each element is increased by one, and the first element becomes the last one

$$\boldsymbol{P}\boldsymbol{x}_{i-1} = \begin{bmatrix} x_{i-(N-1)} \\ \cdot \\ \cdot \\ x_{i-1} \\ x_{i-N} \end{bmatrix} \tag{4}$$

Then the $\boldsymbol{x}_i$ vector can be expressed by the shifted $\boldsymbol{x}_{i-1}$ vector, with an adjustment $\boldsymbol{\Delta x}_i$ made in the last row for the difference between the samples entering and leaving the window

$$\boldsymbol{x}_i = \begin{bmatrix} x_{i-(N-1)} \\ \cdot \\ \cdot \\ x_{i-1} \\ x_{i-N} \end{bmatrix} + \begin{bmatrix} 0 \\ \cdot \\ \cdot \\ 0 \\ x_i - x_{i-N} \end{bmatrix} = \boldsymbol{P}\boldsymbol{x}_{i-1} + \boldsymbol{\Delta x}_i \tag{5}$$

Substituting eqn. 5 into the transformation associated with the ith window in eqn. 2 and using the inverse transform $\boldsymbol{x}_{i-1} = \boldsymbol{T}^{-1}\boldsymbol{y}_{i-1}$, the following relationships are obtained:

$$\boldsymbol{y}_i = \boldsymbol{T}\boldsymbol{x}_i = \boldsymbol{T}[\boldsymbol{P}\boldsymbol{x}_{i-1} + \boldsymbol{\Delta x}_i] = \boldsymbol{T}\boldsymbol{P}\boldsymbol{T}^{-1}\boldsymbol{y}_{i-1} + \boldsymbol{T}\boldsymbol{\Delta x}_i \tag{6}$$

Eqn. 6 expresses the recursivity of the momentary matrix transform (MMT), since calculation of the newly transformed vector $\boldsymbol{y}_i$ is obtained from the previously transformed vector $\boldsymbol{y}_{i-1}$ and the difference between the samples entering and leaving the window.

2.2 Diagonal form of MMT

The momentary matrix transform is particularly efficient and the elements of $\boldsymbol{y}$ can be calculated separately only if the similarity transform $\boldsymbol{T}\boldsymbol{P}\boldsymbol{T}^{-1}$ in eqn. 6 is diagonal. The $\boldsymbol{P}$ matrix has $\boldsymbol{N}$ distinct eigenvalues $(\lambda_0, \ldots, \lambda_{N-1})$ which are the nth complex unit roots, $\lambda_k = w^{-k} = e^{j2\pi k/N}$. There are $\boldsymbol{N}$ linearly independent eigenvectors that correspond to each eigenvalue

$$\lambda_0 \Leftrightarrow s_0 = \begin{bmatrix} 1 \\ 1 \\ \cdot \\ \cdot \\ 1 \end{bmatrix}, \ \lambda_k \Leftrightarrow s_k = \begin{bmatrix} 1 \\ w^{-k} \\ \cdot \\ \cdot \\ w^{-(N-1)k} \end{bmatrix}, \ldots,$$

$$\lambda_{N-1} \Leftrightarrow s_{N-1} = \begin{bmatrix} 1 \\ w^{-(N-1)} \\ \cdot \\ \cdot \\ w^{-(N-1)(N-1)} \end{bmatrix} \tag{7}$$

If the eigenvectors are chosen to be the columns of the inverse of matrix $\boldsymbol{T}$, then $\boldsymbol{T}\boldsymbol{P}\boldsymbol{T}^{-1}$ is a diagonal matrix, with the eigenvalues of $\boldsymbol{P}$ along its diagonal

$$\boldsymbol{T}\boldsymbol{P}\boldsymbol{T}^{-1} = \boldsymbol{S}^{-1}\boldsymbol{P}\boldsymbol{S} = [s_0^{-1} s_1^{-1} \cdots s_{N-1}^{-1}]\boldsymbol{P}[s_0 s_1 \cdots s_{N-1}]$$

$$= \begin{bmatrix} \lambda_0 & 0 & \cdot & \cdot & 0 \\ \cdot & \lambda_1 & 0 & \cdot & 0 \\ \cdot & \cdot & \lambda_2 & 0 & \cdot \\ 0 & \cdot & \cdot & \cdot & 0 \\ 0 & 0 & \cdot & \cdot & \lambda_{N-1} \end{bmatrix} \tag{8}$$

where $\boldsymbol{S}$ is the eigenvector matrix of $\boldsymbol{P}$ made up of the indicated column vectors, as given in eqn 7. The diagonalising matrix $\boldsymbol{S}$ is not unique. An eigenvector $\mathbf{s}_k$ can be multiplied by a constant, and will remain an eigenvector [5]. Therefore the columns of $\boldsymbol{S}$ can be multiplied by any nonzero constants and produce a new diagonalising $\boldsymbol{S}$. There is also no preferred order of the columns of $\boldsymbol{S}$. The order of the eigenvectors in $\boldsymbol{S}$ and the eigenvalues in the diagonal matrix is automatically the same. Therefore all $\boldsymbol{T}$ matrices which satisfy these properties will diagonalise the momentary matrix transform

$$y_i = \begin{bmatrix} \lambda_k & 0 & \cdot & \cdot & 0 \\ \cdot & \lambda_l & 0 & \cdot & 0 \\ \cdot & \cdot & \cdot & 0 & \cdot \\ 0 & \cdot & \cdot & \cdot & 0 \\ 0 & 0 & \cdot & \cdot & \lambda_m \end{bmatrix} y_{i-1} + \boldsymbol{T}_{N-1}(x_i - x_{i-N}) \tag{9}$$

where $k, l, m \in \{0, 1, \ldots, N-1)\}$ and $\boldsymbol{T}_{N-1}$ is the last column of the matrix $\boldsymbol{T}$.

2.3 Inverse of diagonalised MMT

If y_i is available at each sample and the columns of $\boldsymbol{T}$ are the eigenvectors of $\boldsymbol{P}$, an efficient implementation of the inverse of the MMT can be obtained. The inverse MMT (IMMT) at time i is

$$\boldsymbol{x}_i = \boldsymbol{T}^{-1} \boldsymbol{y}_i \tag{10}$$

$$\begin{bmatrix} x_{i-(N-1)} \\ \cdot \\ \cdot \\ x_{i-1} \\ x_i \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & w^{-1} & w^{-2} & \cdot & w^{-(N-1)} \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ 1 & w^{-(N-1)} & w^{-2(N-1)} & \cdot & w^{-(N-1)(N-1)} \end{bmatrix} \times \begin{bmatrix} y_{i,0} \\ y_{i,1} \\ \cdot \\ \cdot \\ y_{i,N-1} \end{bmatrix} \tag{11}$$

The first row of $\boldsymbol{T}^{-1}$ contains only ones so the oldest element of $\boldsymbol{x}_i$ can be computed using adds only

$$x_{i-(N-1)} = \sum_{k=0}^{N-1} y_{i,k} \tag{12}$$

from which the elements of the input sequence $(x_{i-(N-1)} \dots x_i)$ can be computed from the transform domain sequence y_i with an N-1 sample delay.

In summary, the recursive form of the MMT is general. The following Section shows that the DFT/IDFT is the only transform which has the efficient diagonal form eqn. 8, as a result of its column vectors being the eigenvectors in eqn. 7 in a specific order.

3 Momentary Fourier transform

The matrix of the discrete Fourier transform (DFT) and the inverse discrete Fourier transform (IDFT) have the properties described in Section 2.2, thus their columns are the eigenvectors of the matrix $\boldsymbol{P}$. Choosing a specific order of the eigenvectors of $\boldsymbol{P}$ (columns of $\boldsymbol{S}$),

$$\mathrm{DFT} = \boldsymbol{F} = \boldsymbol{S}^{-1}$$

$$= \begin{bmatrix} 1 & 1 & 1 & \cdot & 1 \\ 1 & w & w^2 & \cdot & w^{N-1} \\ 1 & w^2 & w^4 & \cdot & w^{2(N-1)} \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ 1 & w^{N-1} & w^{2(N-1)} & \cdot & w^{(N-1)(N-1)} \end{bmatrix} \quad (13)$$

$$\mathrm{IDFT} = \boldsymbol{F}^{-1} = \boldsymbol{S}$$

$$= \frac{1}{N} \begin{bmatrix} 1 & 1 & 1 & \cdot & 1 \\ 1 & w^{-1} & w^{-2} & \cdot & w^{-(N-1)} \\ 1 & w^{-2} & w^{-4} & \cdot & w^{-2(N-1)} \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ 1 & w^{-(N-1)} & w^{-2(N-1)} & \cdot & w^{-(N-1)(N-1)} \end{bmatrix}$$

$$= \frac{1}{N} \begin{bmatrix} 1 & 1 & \cdot & 1 & 1 \\ 1 & w^{N-1} & \cdot & w^2 & w \\ 1 & w^{2(N-1)} & \cdot & w^4 & w^2 \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ 1 & w^{(N-1)(N-1)} & \cdot & w^{2(N-1)} & w^{N-1} \end{bmatrix} \quad (14)$$

Using the fact that w is the Nth complex root of unity (i.e. $w^{-k} = w^{N-k}$), it can be seen that the columns of the IDFT matrix are the same as the DFT matrix, but they are in reverse order from the second column onwards eqn. 14. Therefore if $\boldsymbol{T}$ performs the DFT eqn. 15 or the IDFT eqn. 16, diagonal forms of the MMT can be obtained

$$\boldsymbol{y}_i = \boldsymbol{F}\boldsymbol{P}\boldsymbol{F}^{-1}\boldsymbol{y}_{i-1} + \boldsymbol{F}\boldsymbol{\Delta}\boldsymbol{x}_i = \begin{bmatrix} 1 & 0 & \cdot & \cdot & 0 \\ \cdot & w^{-1} & 0 & \cdot & 0 \\ \cdot & \cdot & w^{-2} & 0 & \cdot \\ 0 & \cdot & \cdot & \cdot & 0 \\ 0 & 0 & \cdot & \cdot & w^{-(N-1)} \end{bmatrix}$$

$$\boldsymbol{y}_{i-1} + \begin{bmatrix} 1 \\ w^{-1} \\ w^{-2} \\ \cdot \\ w^{-(N-1)} \end{bmatrix} (x_i - x_{i-N}) \quad (15)$$

$$\boldsymbol{x}_i = \boldsymbol{F}^{-1}\boldsymbol{P}\boldsymbol{F}\boldsymbol{x}_{i-1} + \boldsymbol{F}^{-1}\boldsymbol{\Delta}\boldsymbol{y}_i$$

$$= \begin{bmatrix} 1 & 0 & \cdot & \cdot & 0 \\ \cdot & w^{-(N-1)} & 0 & \cdot & 0 \\ \cdot & \cdot & w^{-(N-2)} & 0 & \cdot \\ 0 & \cdot & \cdot & \cdot & 0 \\ 0 & 0 & \cdot & \cdot & w^{-1} \end{bmatrix} \boldsymbol{x}_{i-1}$$

$$+ \begin{bmatrix} 1 \\ w^{-(N-1)} \\ w^{-(N-2)} \\ \cdot \\ w^{-1} \end{bmatrix} (y_i - y_{i-N}) \quad (16)$$

Eqn. 15 expresses the recursive equation of the momentary Fourier transform (MFT) [3, 4, 6, 7]. The N-element vector $\boldsymbol{y}_i$ contains the Fourier coefficients of the N-point sequence $\boldsymbol{x}_i$ ending at sample i. Note that each spectral component $\boldsymbol{y}_{i,k}$ can be calculated independently,

$$y_{i,k} = w^{-k}(y_{i-1,k} + x_i - x_{i-N}) \quad (17)$$

which increases efficiency if only a few frequency components need to be computed, as in the zoom transform.

On the other hand, eqn. 16 is the dual of the MFT, the recursive inverse momentary Fourier transform (IMFT), where the N-element vector $\boldsymbol{x}_i$ contains the N-point data sequence and $\boldsymbol{y}_i$ contains N Fourier coefficients ending at frequency bin i. Note that the each sample in $\boldsymbol{x}_i$ can also be obtained independently and that the same twiddle factors, but in a different order, can be used to calculate both the MFT and IMFT.

Thus it has been shown that if the DFT or the IDFT performs the momentary matrix transform of a sequence, the elements of the transformed sequence can be computed recursively and independently using N complex multiplies and $N+1$ complex adds (additional computational savings are available if the input sequence is real-valued).

3.1 Nonrecursive inverse MFT

The *nonrecursive* inverse momentary Fourier transform (IMFT) can be expressed using eqns. 12 and 16 as follows:

$$x_{i-(N-1)} = \frac{1}{N}\sum_{k=0}^{N-1} \boldsymbol{y}_{i,k} \quad (18)$$

from which each sample of the input sequence $\boldsymbol{x}_i$ can be computed using adds only from the two-dimensional time-dependent spectrum $\boldsymbol{y}_i$ with an $N-1$ sample delay. In this way the MFT–nonrecursive [MFT] transform pair eqns 15, 18 can provide an efficient frequency-domain manipulation method (e.g. filtering), especially if many of the DFT coefficients are not needed.

If the elements of $\boldsymbol{x}_i$ are real, one can take advantage of the conjugate symmetry of the spectrum, and the oldest element can be computed using only the real part $\Re$ of the spectrum:

$$\boldsymbol{x}_i - (N-1) = \frac{1}{N}\sum_{k=0}^{N-1} \Re\{\boldsymbol{y}_{i,k}\} \quad (19)$$

It has been shown [3] that if $\boldsymbol{x}_i$ is real, the Hilbert transform $\mathscr{H}$ of $x_{i-(N-1)}$ can be obtained by summing only the imaginary part $\Im$ of the spectrum coefficients

$$\mathscr{H}\{\boldsymbol{x}_{i-(N-1)}\} = \frac{1}{N}\sum_{k=0}^{N-1}\Im\{\boldsymbol{y}_{i,k}\} \tag{20}$$

In this case, the MFT–nonrecursive IMFT pair can be useful for different signal processing applications where the in-phase and quadrature components of the signal are needed (i.e. communications and radar systems).

4 Implementation of MFT

As shown in Section 3, the spectrum components in the MFT algorithm can be calculated independently of each other. Thus, the MFT can be implemented using a sequence of identical blocks, where a block computes the spectrum at a single frequency using eqn. 17. The software implementation of one MFT block can be obtained using the trigonometric form of the equation

$$y_{i,k} = w^{-k}(y_{i-1,k} + x_i - x_{i-N}) \tag{21}$$

where $w^{-k} = e^{j2\pi k/N} = cos(\Phi_k) + j\sin(\Phi_k)$, $\Phi_k = 2\pi k/N$, and equivalently,

$$\begin{aligned}\Re\{y_{i,k}\} &= \cos(\Phi_k)(\Re\{y_{i-1,k}\} + \Re\{x_i - x_{i-N}\}) \\ &\quad - \sin(\Phi_k)(\Im\{y_{i-1,k}\} + \Im\{x_i - x_{i-N}\}) \\ \Im\{y_{i,k}\} &= \cos(\Phi_k)(\Im\{y_{i-1,k}\} + \Im\{x_i - x_{i-N}\}) \\ &\quad + \sin(\Phi_k)(\Re\{y_{i-1,k}\} + \Re\{x_i - x_{i-N}\})\end{aligned} \tag{22}$$

Eqns. 21 and 22 correspond to the kth MFT block for the complex sequence x_i, and $y_{i,k}$ is the kth spectrum component at sample i. The MFT blocks can be organised in a for loop to calculate the needed DFT coefficients. The following pseudocode segment illustrates the computer coding of the MFT algorithm, assuming the sine and cosine arrays (twiddle factors) have been precomputed:

```
calculate $(x_i - x_{i-N})$;
for $k$ = start to start + $N_c - 1$ do
    MFTblock(k);    %    computed as in eqn. (22)
endfor
```

where N_c is the number of DFT coefficients to be computed, $1 \leq N_c \leq n$. The index of the 'for' loop indicates that it is possible to compute only a small group of the DFT coefficients. If $N_c < n$ or if another subset of the coefficients are to be calculated, we refer to the MFT as a 'reduced-MFT', where significant computation savings can be realised if not all of the spectral coefficients are needed. If the calculation of the spectrum coefficients is off-line, the difference of the entering and leaving samples of the window can be calculated for the whole data set and stored in a file or an array in memory. If it is on-line, a modulo-N array is needed to calculate $x_i - x_{i-N}$.

Table 1 gives the memory requirements of the MFT algorithm. The memory requirements depend on the number of calculated spectrum coefficients. If the whole spectrum is computed, an $8N$ word memory is needed for the computation. Assuming that the input signal has 16-bits of precision, and 16-bit accuracy is required at the output, the MFT arithmetic should be done with a word length of at least 24 bits, as the multiple stages of the MFT algorithm will create more roundoff noise than the $\log_2 N$ stages of the FFT.

Table 1: Memory requirement of MFT (words)

Array type	Size
Twiddle factors (sine and cosine array)	$2N_c$
Modulo-N FIFO for complex $x_i - x_{i-N}$	$2N$
Spectrum coefficients at time $i-1$ ($y_{i-1,k}$)	$2N_c$
Spectrum coefficients at time i ($y_{i,k}$)	$2N_c$

4.1 Example of MFT usage

To illustrate the usage of the incremental form of the MFT, a frequency shift key (FSK) modulated sinusoidal signal of length $4N$ samples is used. Using an analysis window length $N = 100$, and two frequencies of five cycles per window and 29 cycles per window, the magnitude of the evolving spectrum is shown in Fig. 1, when the MFT is incremented by one sample at each analysis stage.

The MFT begins with the initial conditions of $y_0 = 0$. This is equivalent to having N zeros precede the data vector. In Fig. 1 note how the energy in the spectrum rises from zero to a maximum in the first N samples. Also note how spectral leakage is observed in the first $N-1$ time samples, because the sinusoidal signal does not have an integer number of cycles per window over this time. At time N, there is an integer number of cycles per window, so all the energy in the spectrum lies in one bin. For the next $N-1$ samples, leakage occurs again as the window slides towards to the next frequency component of the signal. The spectral energy of the 5th frequency bin decays to zero while the spectral energy of the 29th bin rises to its maximum. This spectrum energy swapping between the two frequency bins is repeated as the window is moving through the two frequency components.

In Fig. 2, the same FSK signal is analysed in the presence of noise (SNR = 0 dB). The spectrum energy swapping between the two frequency bins is also noticeable, which shows how the MFT can be useful for signal detection in a noise environment. The MFT can be useful here if the FSK switching times are not known.

5 Computing efficiency of MFT

This section examines the computing efficiency of the MFT compared to traditional DFT and FFT implementations. Although computing efficiency has many ramifications, we restrict attention to the number of real-valued 'signal processing' operations (multiplies and adds) required to implement the algorithms.

Consider the case where N-point DFTs are used to analyse an M-point complex-valued data record. If the window is shifted by q samples between each DFT application, where $1 \leq q \leq N$, then $(M-N)/q + 1$ DFTs are needed to spectrally analyse the record in the case of the FFT. If the MFT is applied, M MFTs are needed, because the spectrum coefficients have to be calculated at each time sample, irrespective of the value of q. Then, when radix-2 FFTs are used,

$$OPS_{FFT} = ((M-N)/q + 1)5N\log_2(N) \tag{23}$$

real operations are needed, while in the case of the MFT

$$OPS_{MFT} = M(8N_c + 2) \tag{24}$$

real operations are needed to analyse the whole record. In this way, the MFT becomes efficient relative to the FFT when the shift q is small. From eqns. 23 and 24, the

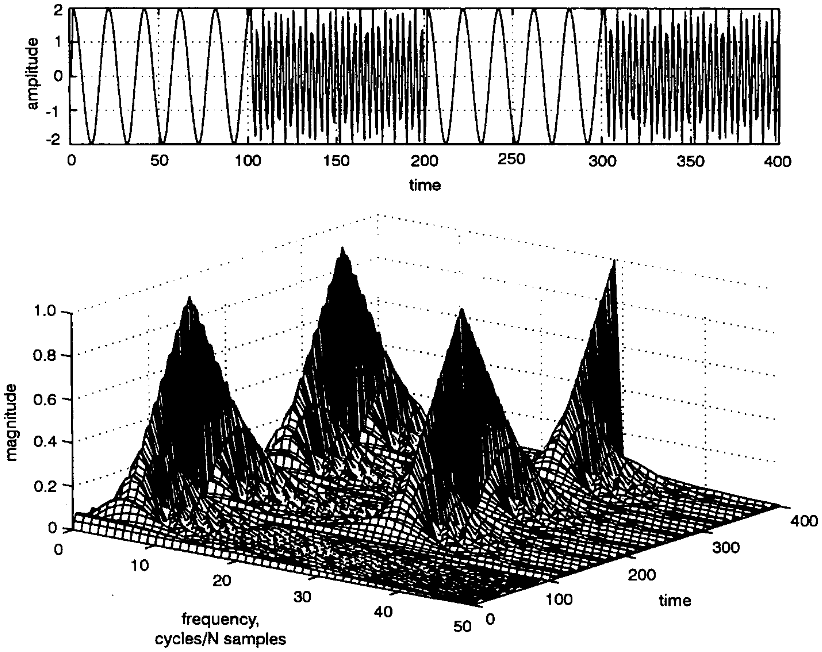

Fig. 1 *FSK signal analysis using the MFT: FSK signal (top) and time-varying spectrum (bottom)*

number of shifts q_{MFT} between DFTs when the MFT is more efficient than the radix-2 FFT is less than

$$q_{MFT} < \frac{(M - N)5N \log_2(N)}{M(8N_c - 1) - 5N \log_2(N)} \tag{25}$$

q_{MFT} is a function of the length of the data record M, the size of the window N and the number of calculated MFT coefficients N_c. In Fig. 3, the shift between DFTs when the MFT is more efficient is shown as a function of the window length, for two values of N_c.

The full MFT is more efficient compared with the radix-2 FFT if the shift between DFTs is very small ($q_{MFT} < 5$), while the reduced MFT ($N_c = N/4$) is more efficient even for larger values of shift. Note, if the data record is longer, the values of q_{MFT} are larger for all window sizes. The computational load for $q_{MFT} = 1$ is illustrated in Fig. 4.

The arithmetic of the MFT is linear with the computed spectrum coefficients N_c and the length of the data record M. For a given record size, the MFT arithmetic remains the same with varying shifts, while the FFT arithmetic drops down considerably as the value of shift gets larger.

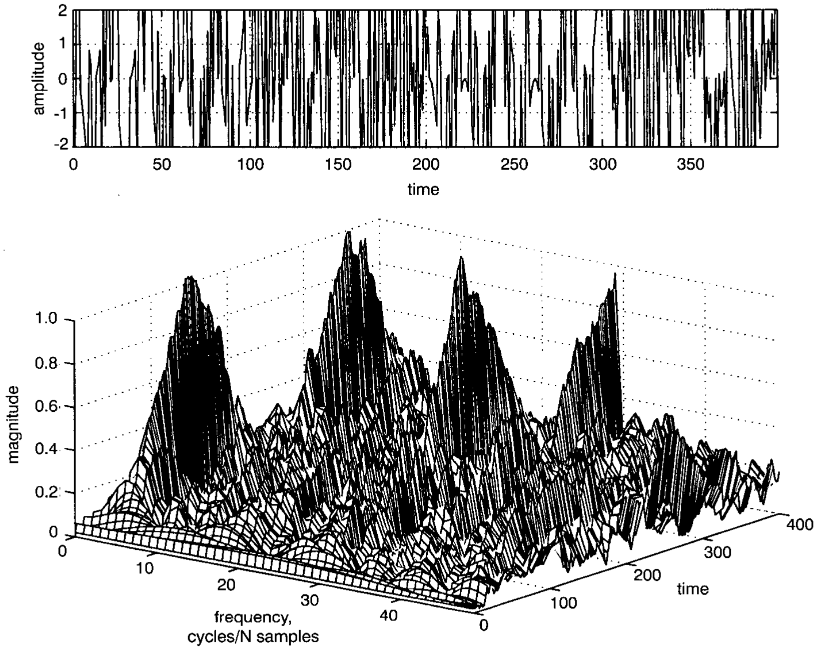

Fig. 2 *Signal detection using the MFT: FSK signal with noise (top) and time-varying spectrum (bottom)*

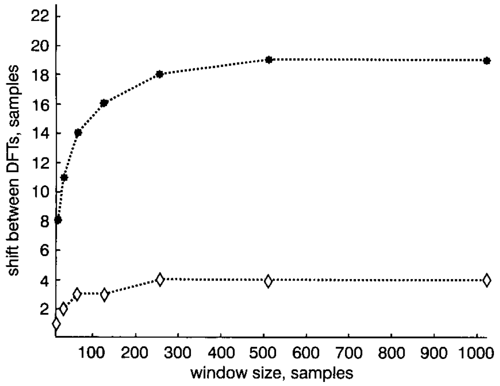

Fig. 3 *Shift between DFTs for which the MFT is more efficient than the radix-2 FFT*

Total samples analysed = 5000
* 1/4N MFT
◇ full MFT

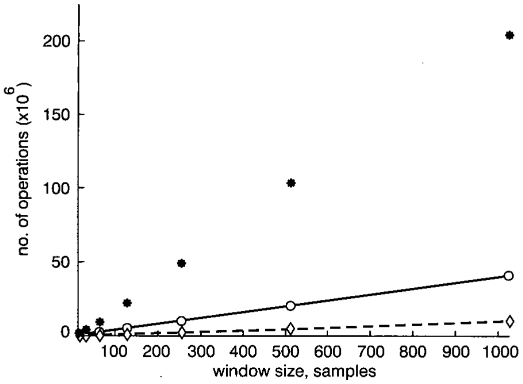

Fig. 4 *Comparison of MFT and FFT arithmetic when* $q_{MFT} = 1$

Total samples analysed = 5000
* radix-2 FFT
○ full MFT
◇ 1/4N MFT

The computational order of the MFT to recursively calculate the coefficients of an N-point DFT is N, a factor of $\log_2 N$ improvement over the FFT. If only a subset of the spectrum components are needed, the computing load of the MFT can be further reduced, calculating only the frequency coefficients of interest. The MFT does not rely on on N being a power of two to obtain its efficiency, in contrast to standard FFT algorithms. In this way the MFT can provide more efficient computation of the DFT when any or all of the following conditions apply:

- DFTs are highly overlapped,
- only a subset of the Fourier coefficients are needed and/or
- a specific, noncomposite DFT length is needed.

Considering these MFT properties, we see that it can be useful in different applications of signal processing such as

- on-line computations in real-time spectral analysis,
- on-line signal identification and detection
- speech processing and
- radar and sonar processing.

6 Application to SPECAN algorithm

The SPECtral ANalysis (SPECAN) algorithm [8, 9] is an alternative to conventional processing algorithms for SAR azimuth compression. It consists of two major computational steps

- deramping, and
- weighted DFT.

Deramping is the operation of multiplying the received linear FM signals with a complex conjugate reference signal having the same FM rate K_a, but opposite FM slope. In radar processing, deramping can be done in range or in azimuth, but in our context we deal with the azimuth signal. The deramping operation turns the linear FM target signals into constant-frequency sine waves, with frequency proportional to azimuth position. When viewed in a frequency/time diagram, the deramped target energy exists in parallelogram-shaped regions, as illustrated by the six targets in Fig. 5. Each deramped target in the parallelogram has a unique time span and a unique frequency ranging from 0 to F_a, where F_a is the azimuth sampling rate or pulse repetition frequency. The base of the parallelogram is $M = F_a^2/K_a$ samples long and the DFTs are N samples long, in input time space.

The next step in the processing is to separate the target energy into different output cells, corresponding to their azimuth position. This is done by performing short-length DFTs across the deramped data. The placement of the DFTs for the four-look case is shown in Fig. 5, where the DFT length is N samples. Weighting is used in the DFTs for sidelobe control, which means that the looks should be overlapped a small amount to obtain uniform energy utilisation.

6.1 SPECAN SAR processing using MFT

The azimuth FM rate of the received SAR signal is inversely proportional to range, so it changes as the range varies in each range cell. To keep the resolution and output sample rate constant across the range swath there is a need to choose different DFT lengths, with the DFT length increasing one sample at a time as range increases. The effect of the varying range on the azimuth FM rate and on the desired DFT length for a typical airborne radar case is shown in Fig. 6. The radar parameters used are given in Table 2. Note that there is a need for a wide range of DFT lengths to keep the resolution constant through the whole swath.

The radix-2 FFT can only be used when the DFT length is power of two. In other cases of window length, mixed-radix FFT algorithms are used to achieve efficiency when N is a highly composite number. However, this makes the architecture of the SPECAN processor rather complex when many different FFT lengths are needed. In contrast

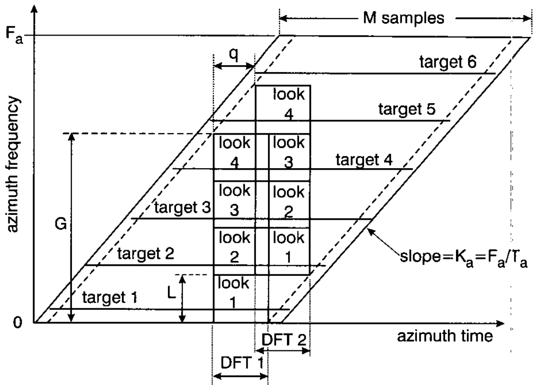

Fig. 5 *Processing regions of SPECAN algorithm*

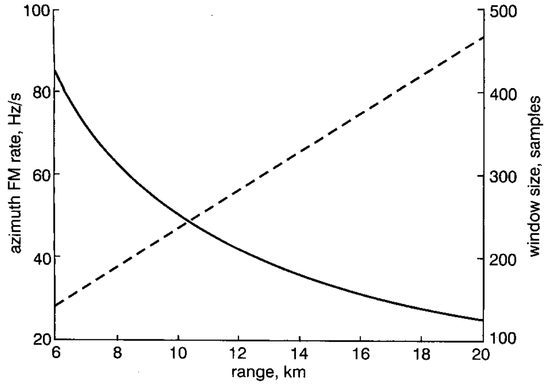

Fig. 6 *Azimuth FM rate and DFT length with varying range, airborne SAR case*

—— azimuth FM rate
---- DFT length

Table 2: Airborne SAR parameters assumed for SPECAN arithmetic calculation

Radar parameter	Value	Units
Platform velocity V_r	120	m/s
Wavelength λ	0.057	m
Weighting parameter σ	0.68	-
Guard band β	0.15	-
Slant range R	6–20	km
Sampling frequency F_a	300	Hz
Number of looks N_l	10	-
Azimuth resolution σ	4	m

to FFT algorithms, the structure and the efficiency of the MFT does not depend on the size of the DFT.

In the DFT operation only a portion of the spectrum coefficients (the good output samples G) are used to obtain the compressed output data from each DFT. The number of these spectrum components are the same from one DFT to the next, but their position changes with each DFT. So, a simple reduced-MFT algorithm cannot be used. The arithmetic of the required reduced-MFT algorithm is introduced subsequently.

For the first MFT (e.g. DFT 1 in Fig. 5) all of the good output samples are computed for the first time, so using eqn. 24 the arithmetic of the first MFT is

$$MFTOPS_{DFT1} = N(8\bar{G} + 2) \tag{26}$$

For the next MFT (DFT 2 in Fig. 5) the position of the subband of the good output points is shifted towards to the higher frequencies by L samples, the number of output samples per look. Thus, there are L new frequency components to calculate in addition to the $G - L$ recalculated ones. The number of real operations needed for the L new coefficients is

$$MFTOPS_{DFT2\ New} = N(8L + 2) \tag{27}$$

while the arithmetic of the previously computed $G - L$ spectrum coefficients is

$$MFTOPS_{DFT2\ old} = q(8(G - L) + 2) \tag{28}$$

where q is the input sample shift between successive DFTs. The 'new' and the 'old' spectrum coefficients have to be computed $(M - N)/q - 1$ times through the whole region. Using eqns. 26 to 28

$$\mathrm{MFTOPS} = N(8G + 2) + \left(\frac{M - N}{q} - 1\right)(N(8L + 2) + q(8(G - L) + 2)) \tag{29}$$

real operations are needed to process the whole processing region with the reduced MFT. Although eqn. 29 looks rather complex, the implementation of this reduced-MFT algorithm is the same as the full MFT algorithm, except for the timing and synchronisation of the sub-band of the spectrum coefficients. The weighting can be implemented using a short-length convolution after the MFT. The arithmetic of the weighting is not included in the present calculations.

Fig. 7 shows the number of operations of the SPECAN azimuth compression for the airborne radar case, covering the range of DFT lengths defined in Fig. 6. In this figure we compare the arithmetic of the direct DFT algorithm, the full and reduced MFT, the mixed-radix and the radix-2 FFT. In Fig. 7 the FFT arithmetic is generally smaller than the MFT arithmetic, although it is quite variable as the radix changes throughout the range swath.

In the SPECAN algorithm the resolution is inversely related to the DFT length, thus larger DFTs are needed to obtain finer resolution. As the DFT length gets longer the number of good output points per DFT will shorten, therefore more DFT blocks with higher overlap ratios will be needed to cover the processing region (Fig. 5). Thus as the processed resolution increases, the MFT will become more efficient relative to the mixed-radix FFT.

6.2 Output sample rate

Besides the complexity and computational efficiency, another important issue in the SPECAN algorithm is to keep the output sampling rate constant. In other words, targets which are T seconds apart in azimuth input time must appear T seconds apart in the output data. It was shown in [8] that the azimuth output sample rate is

$$F_{out} = \frac{K_a N}{F_a}\ [\mathrm{Hz}] \quad \text{where} \quad K_a = \frac{2V_r^2}{\lambda R}\ [\mathrm{Hz/s}] \tag{30}$$

The output sampling rate strongly depends on the azimuth FM rate K_a, so when it changes with R throughout the

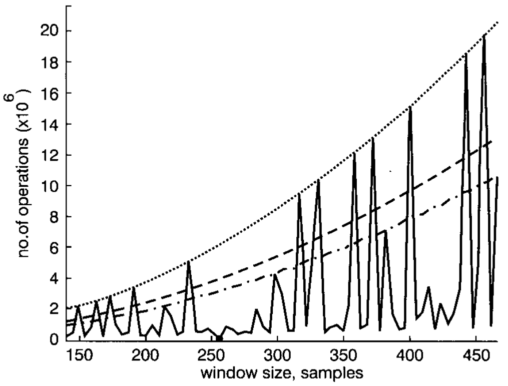

Fig. 7 *Arithmetic of SPECAN azimuth compression with different DFT algorithms, airborne SAR case*

······ DFT
—— mixed-radix FFT
---- full MFT
–·–· reduced MFT
● radix-2 MFT

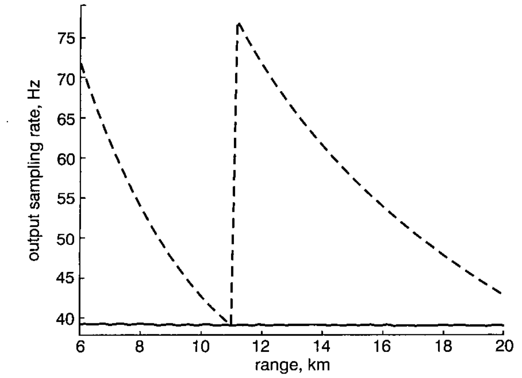

Fig. 8 *Output sampling rate of SPECAN algorithm*

--- radix-2 FFT, $N = 256$ and 512
—— MFT

swath, there is a need for a slowly varying DFT length to keep the output sampling rate constant. Fig. 8 shows F_{out} as the function of range, when the MFT and the radix-2 FFT is applied in the SPECAN algorithm to the airborne system. Note that when the MFT algorithm is used the output sampling rate is more uniform. During the application of the radix-2 FFT, only two transformation lengths 128 and 256 can be used, which is the reason for the large change of the output sampling rate in this case.

7 Application to burst-mode SAR processing

Burst-mode operation is used in SAR systems such as RADARSAT or ENVISAT to image wide swaths, to save power or to reduce data link bandwidth [10]. In this operational mode the received data is windowed in a periodic fashion in the azimuth time variable which results in a segmented frequency–time structure of its Doppler energy. This frequency–time pattern requires special processing to maintain accurate focusing, consistent phase and efficient computing.

7.1 Properties of targets in burst-mode data

A typical two-beam burst-mode data collection pattern is shown in Fig. 9, where 16 targets are depicted. The targets are all in the same range cell, and are evenly spaced in azimuth time. The azimuth time variable runs down the page, and the staggered vertical lines in the top left panel show the exposure time of each target. In this example, the burst length is chosen to be 20% of the aperture length, and the solid parts of each line show that part of each target actually exposed in burst mode. Each burst of data is followed by an equal-length gap.

Note that the part of the target exposure captured in burst mode varies with each target, which is illustrated in the frequency-time diagram in the top right panel of Fig. 9. Each successive target is received at a lower Doppler frequency within a given burst, but is later captured at a higher frequency in the next burst as long as it stays within the beam. The slope of the lines in this part of the Figure is given by the azimuth FM rate K_a

$$K_a = F_a / T_a \quad \text{[Hz/s]} \tag{31}$$

where F_a is the radar pulse repetition frequency (PRF) and T_a is the time taken for a target to generate a Doppler frequency span of F_a (which we refer to as the synthetic aperture time or the total beam exposure time).

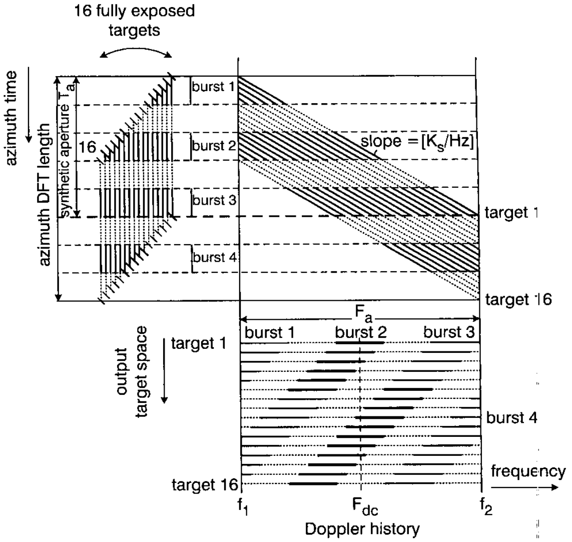

Fig. 9 *Burst-mode SAR operation showing time and frequency properties of 16 fully-exposed, evenly-spaced targets in one range cell*

······ continuous signal
—— burst signal

The Doppler history of the 16 targets is shown in the bottom panel of Fig. 9. As T_a equals five burst lengths in this case, F_a in the frequency domain consists of five burst bandwidths. This lower plot shows the distribution of target spectral energy when an azimuth DFT is taken over four bursts plus four gaps, as shown in the top left of Fig. 9. Note that some targets appear in two full bursts (targets 6 and 16), some appear in three full bursts (targets 1 and 11), while others appear in two full and one partial burst. In this case the average number of target exposures or bursts per aperture is 2.5.

If single-look complex processing is to be done there is a choice of which bursts to use for each target. Normally, the target exposures closest to the Doppler centroid F_{dc} will be selected, as shown by the heavier lines in the lower part of Fig. 9. However, other bursts may also be chosen, e.g. when the data is processed for InSAR purposes, or for speckle reduction purposes.

7.2 SIFFT burst-mode processing algorithm

Most SAR processing algorithms are based on the fast convolution principle where a matched filter is applied in the azimuth or Doppler frequency domain. When this method is applied to burst-mode data, the interburst gaps are filled with zeros and all the bursts are compressed at once using a full length matched filter followed by an IFFT. However, the compressed targets are then left with a burst-induced modulation [11].

The SIFFT algorithm differs from the conventional fast convolution algorithm in that short, overlapped IFFTs are taken after the matched filter multiply in the Doppler domain [12, 13]. The IFFT lengths are chosen so that when one burst of a target is fully captured by the IFFT, little or no energy from adjacent bursts of the same target is present in the same IFFT. In this way each IFFT compresses a group of targets without interference (modulation) from other bursts, and an accurate impulse response is obtained.

To capture a target fully, the length of the IFFT must be at least as long as the bandwidth of one burst of that target, which is

$$BW_{Hz} = \frac{N_b K_a}{F_a} \quad \text{[Hz]} \quad \text{or}$$

$$BW_{bin} = \frac{N_b K_b N_{FFT}}{F_a^2} \quad \text{[frequency bins]}$$

where N_b is the burst length in samples and N_{FFT} is the length of the azimuth FFT in samples. Then the minimum IFFT length is

$$N_{IFFT\ min} = BW_{bin} = \frac{N_b K_a N_{FFT}}{F_a^2} \quad \text{[samples]} \qquad (33)$$

The IFFT cannot be longer than the bandwidth of one burst plus one gap, so that a fully-exposed target is not contaminated by a partial exposure of the same target at a different frequency. In the N_m-beam scanSAR case the length of the gap is often equal to $N_m - 1$ times the burst length, in which case the maximum-length IFFT is

$$N_{IFFT\ max} = N_m BW_{bin} = N_m \frac{N_b K_a N_{FFT}}{F_a^2} \quad \text{[samples]} \quad (34)$$

Note that $N_{IFFTmax}$ and $N_{IFFTmin}$ vary with range and with N_{FFT}. The effect of this property is discussed in the next subsection, where the arithmetic of the SIFFT is given. The length and locations of IFFT$_{max}$ and IFFT$_{min}$ that can be used to extract targets from that part of the spectrum with the highest energy are shown in Fig. 10. In the case where the maximum possible DFT lengths are selected, two IFFTs are shown, such that IFFT$_{max1}$ and IFFT$_{max2}$ together compress the complete set of targets shown in the Figure. Then the outputs of these two IDFTs are stitched together to form a contiguous set of compressed targets. The targets compressed are shown by the heavier black horizontal lines.

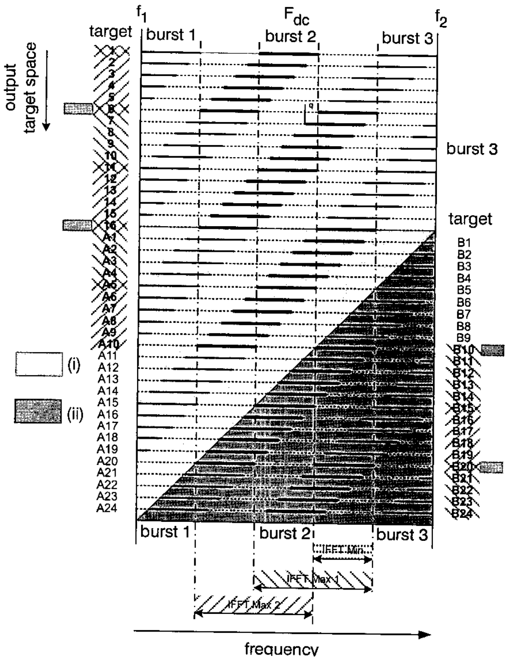

Fig. 10 *How IDFTs are placed to compress groups of targets from each burst*

(i) targets after this synthetic aperture
(ii) targets previous to this synthetic aperture

Fig. 10 is a more detailed version of the lower panel of Fig. 9 with one main addition. We have included partially exposed targets which preceed and follow the fully-exposed targets 1–16, and which are at least partially captured by the 4 + 4 burst FFT. These partially-exposed targets are located before and after targets 1–16 and are denoted by B1–B24 (dark shaded region) and A1–A24 (light shaded region), respectively. FFT wraparound makes the targets B1–B24 appear at the end of the spectrum. Left- and right-sloping cross-hatching is used to indicate which targets come from each of the two IDFTs.

It can be seen that IFFT$_{max1}$ captures the complete energy of a single burst of targets 1, 6–11, 16–A5, B10–B15 and B20–B24. For these targets IFFT$_{max1}$ does not extract any energy from other bursts so their impulse response is not corrupted by modulation. Similarly, IFFT$_{max2}$ captures the complete energy of a single burst of targets 1–6, 11–16, A5–A10 and B15–B20 and between the two IFFTs, all the targets are correctly compressed.

Fig. 10 also shows the size and starting location of the minimum length IFFT, which is used in the efficiency calculations of the Appendix. The next IFFT$_{min}$ is shifted q samples to the left of the one shown.

To form a continuous output image the results of successive IFFTs are stitched together. If only bursts with the highest energy are used to compress targets, each output target gets placed in a different output cell. Note that not all of the partially exposed targets shown in Fig. 10 can be compressed with the best SNR in this FFT, but they can be in preceding and following FFTs. Targets A16–A24 cannot be compressed at all with this FFT because none of the bursts covers them completely. Also note that targets before the full synthetic aperture (B10–B24) get compressed at the end of the output array, but their position can be correctly identified and rearranged in the output array.

7.3 Efficiency of SIFFT algorithm using IMFT

7.3.1 ENVISAT parameters: To show the efficiency of the SIFFT algorithm using the IMFT against the IFFT implementation we use the parameters of the alternating polarisation (AP) mode of the ESA ASAR on the ENVISAT satellite. The AP mode provides medium resolution products (approximately 30 m) in any of the seven swaths located over a range of incidence angles from 15 to 45° with polarisation changing from subaperture to subaperture within the synthetic aperture. Effectively, a two-beam case ScanSAR technique is used but without varying the beam elevation angle. The effective radar velocity of the satellite is $V_r = 7000$ m/s, the radar wavelength is $\lambda = 0.0567$ m, and the azimuth FFT length is set to 2048 and 4096 for the efficiency evaluation. Other parameters of the seven swaths are given in Table 3.

Table 3: ENVISAT swath parameters

Swath	PRF [Hz]	Burst/gap length [samples]	Range [km]
IS 1	1678	194	825–864
IS 2	1645	196	843–891
IS 3	2096	257	887–934
IS 4	1680	218	929–990
IS 5	2082	277	983–1032
IS 6	1698	238	1027–1087
IS 7	2070	297	1080–1133

7.3.2 Effect of varying SAR parameters and SNR/efficiency tradeoffs:

As the azimuth FM rate of the received SAR signal is inversely proportional to range, the bandwidth of the bursts eqn. 32 varies with range because the burst length is constant in the time domain. At near range the bandwidth is a maximum while at far range it is a minimum, as shown in Fig. 11 for swath IS1. In a swath the minimum IFFT length $N_{IFFTmin}$ should be at least as long as the maximum BW_{bin} to compress all the targets in each range cell correctly. Because of different burst bandwidths, this minimum IFFT length is different for each ENVISAT swath. The SNR of the SIFFT algorithm depends on the ratio of the IFFT length and the burst bandwidth, both expressed in bins [13]. The SNR is maximum when the IFFT length is equal to the burst bandwidth ($N_{IFFT} = N_{IFFTmin}$), while the SNR is 3 dB lower when the IFFT window is one burst plus one equal-sized gap long ($N_{IFFT} = N_{IFFTmax}$). So to keep the SNR loss below a certain value across the swath there is a need to choose specific azimuth IFFT lengths. The change in SNR for a given IFFT length is

$$\mathrm{dSNR} = 10\log_{10}\left(\frac{N_{IFFT}}{BW_{bin}}\right) \tag{35}$$

The IFFT length should be kept constant with range to keep the compressed sample spacing uniform. Then, as the burst bandwidth decreases with range, the minimum IFFT length is set at near range, and the SNR will decrease slowly with range. Fig. 11 shows how dSNR changes with range in the case of IS1 swath and $N_{FFT} = 2048$. The decrease in SNR is zero at near range and rises to about 0.2 dB at far range. Note that although BW_{bin} in eqn. 35 depends on N_{FFT}, the slope and maximum of dSNR is the same for different azimuth FFT lengths.

7.3.3 Arithmetic of SIFFT using IMFT and IFFT algorithms:

During the efficiency evaluation of the SIFFT algorithm, the full-IMFT, the reduced-IMFT and the mixed-radix IFFT algorithms are considered. A formula for the arithmetic of each algorithm is developed in the Appendix. During the efficiency evaluation, we choose the IDFT lengths on the principles

- maximum SNR at near range
- minimum sampling rate at near range, and
- the sampling rate constant with range

and consider two possible forward azimuth FFT lengths of 2048 and 4096 samples.

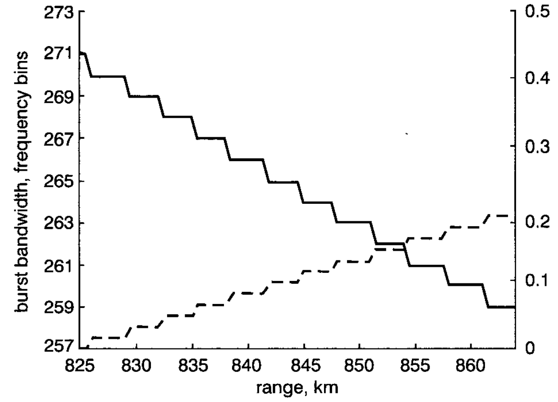

Fig. 11 *Burst bandwidth and dSNR of IS1 swath*

Number of windows to choose from = 1; $N_b = 194$; $N_{FFT} = 2048$

—— burst bandwidth

---- dSNR

First we make the IDFT as small as possible at near range i.e. $N_{IDFT} = MaxBW_{bin}$, and have it stay the same with range, even though the burst bandwidth decreases with increasing range. Thus there is only one IDFT window length to choose from in the IFFT or the IMFT algorithms. Secondly, we consider the case where the IDFT is allowed to be up to four samples longer than the minimum i.e. $MaxBW_{bin} \leq N_{IDFT} \leq MaxBW_{bin} + 4$. This allows some flexibility in choosing a favourable IFFT length from five different window sizes, at the expense of a small decrease in SNR.

The burst bandwidth in bins is directly proportional to N_{FFT} eqn. 32, so it changes in the same ratio as N_{FFT}. If the value of BW_{bin} is bigger because of a bigger N_{FFT} it is easier to find a highly composite number in its neighbourhood, thus it is easier to pick an efficient length for the IFFT. The change in dSNR when a more suitable window is used for the IDFTs is less than 0.1 dB, thus the SNR decrease is quite small.

Fig. 12 shows the number of real operations of the SIFFT algorithm per forward DFT, when it is used to compress the data of IS1 swath. The algorithms used to obtain the azimuth compression are the mixed-radix IFFT and the full- and reduced-IMFT. Fig. 12*a* shows that the IMFT algorithms are more efficient through the whole swath when only the minimum window length can be used.

The IFFT window begins to cover the bandwidth of more than one target as BW_{bin} decreases with range, so there is a change in the number of targets compressed per IFFT. The IMFT arithmetic decreases slowly with range while the arithmetic of the IFFT drops by half when the number of

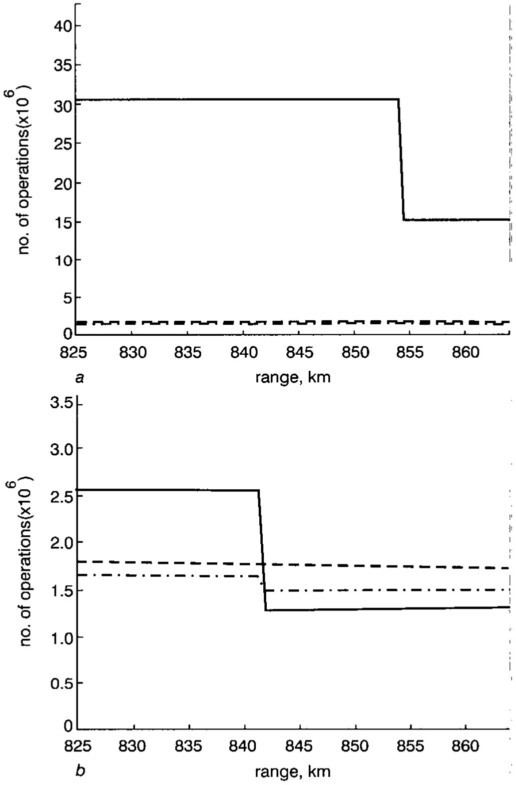

Fig. 12 *Arithmetic of SIFFT algorithm when applied to IS1 swath*

—— IFFT

--- full IMFT

-·-· reduced IMFT

a Maximum SNR, $N_{IFFT} = 271$, $N_{FFT} = 2048$

b 0.1 dB SNR loss allowed, $N_{IFFT} = 275$, $N_{FFT} = 2048$

targets compressed per IFFT doubles from 1 to 2. The IFFT arithmetic is constant on both sides of the down-step. There is more than a factor of ten difference between the arithmetic of IMFT and IFFT, because the window length is prime (271) in this case.

In Fig. 12*b* the mixed-radix IFFT arithmetic dramatically drops when the IFFT window length is a composite number (275). The arithmetic of the IMFT algorithms did not change significantly and they are more efficient only in a part of the swath, where only one target can be fully compressed in each group. The down-step of the IFFT arithmetic happens at closer range, because the IFFT window is larger, so it starts to fully cover two targets earlier. The reduced-IMFT arithmetic also drops down when G_{group} doubles, but this change is not significant compared to the change in the IFFT arithmetic. The arithmetic of the IMFT and IFFT algorithms of the other swaths follows the same pattern as the arithmetic of swath IS1.

In Figs. 13 and 14, the average millions of operations (MOPS) are shown for all the ENVISAT swaths when the azimuth FFT is 2048 and 4096 samples long (the results are given per forward DFT, each point on the horizontal axis represents another ENVISAT swath). The trend of the arithmetic of the full- and reduced-IMFT is similar in all cases, while the IFFT arithmetic is quite variable depending on the composition of the window length. When $N_{FFT} = 2048$ (Fig. 13) the IMFT is more efficient for most of the swaths even if there is an option to choose a suitable window length for the IFFT algorithm.

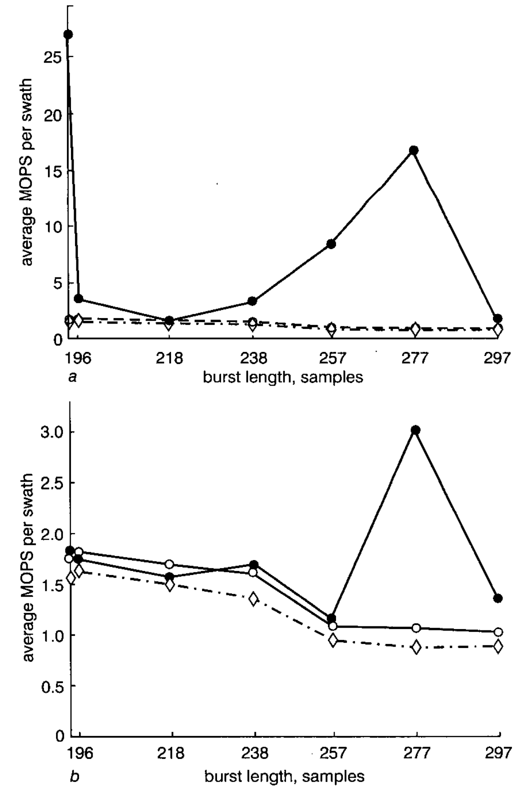

Fig. 13 *Arithmetic of SIFFT when applied to ENVISAT AP burst-mode*
—●— IFFT
—○— full IMFT
—◇— reduced IMFT
a Maximum SNR; IDFT windows to choose from = 1; $N_{FFT} = 2048$
b 0.1 dB SNR loss allowed; IDFT windows to choose from = 5; $N_{FFT} = 2048$

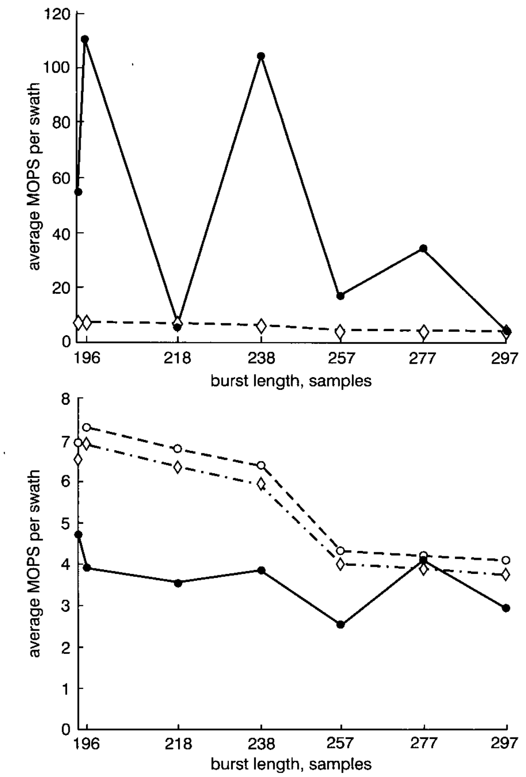

Fig. 14 *Arithmetic of SIFFT when applied to ENVISAT AP burst mode*
—●— IFFT
—○— full IMFT
—◇— reduced IMFT
a Max. SNR; IDFT windows to choose from = 1; $N_{FFT} = 4096$
b 0.1 dB SNR loss allowed; IDFT windows to choose from = 5; $N_{FFT} = 4096$

When $N_{FFT} = 4096$ (Fig. 14) the IMFT is more efficient than the IFFT if the maximum SNR is to be obtained (Fig. 14*a*). When there is a possibility to choose a favourable IFFT length, the IFFT is more efficient for all the swaths except one. There is a higher possibility of finding a highly composite number in the neighbourhood of the smallest window size value, and more groups of good targets can be extracted, when the azimuth FFT is larger.

7.3.4 *Efficiency:*

From this arithmetic survey one can see that the IMFT algorithm can improve the computational efficiency of the SIFFT algorithm when

- the azimuth FFT is relatively small,
- the maximum SNR is required, and
- the IDFT window length is a noncomposite number.

Besides its efficiency, the IMFT has the following advantages when applied to the SIFFT algorithm:

- the IMFT has more consistent computing load as the burst bandwidth changes, and
- it is easier to implement the IMFT algorithm for different burst and N_{FFT} lengths, because the same IMFT algorithm can be used for the different IDFT window lengths.

8 Conclusions

The momentary matrix transform has been introduced and it has been shown that when it takes the form of the DFT or the IDFT the resulting MFT/IMFT have an efficient recursive computational structure. The spectrum coefficients of the MFT/IMFT can be calculated independently and only one complex multiplication and two complex additions are needed to update each spectrum component. This is a factor of $\log_2(N)$ improvement over the radix-2 FFT algorithm *if all incremental DFT results are needed.* The efficiency of the MFT/IMFT does not rely on the transform length being a power of two, in contrast to standard FFT algorithms.

The applicability of the MFT to the SPECAN SAR processing algorithm has been investigated. Although the MFT does not improve the computational efficiency of the SPECAN algorithm, except at the finest resolutions, it has several advantages over the FFT implementation, notably in keeping the sample rate constant.

In burst-mode SAR processing, the time-varying spectral properties of the azimuth received data requires that highly-overlapped inverse DFTs be used at specific locations in the frequency domain to obtain accurate azimuth compression. It was shown that the IMFT can be more efficient than the IFFT when it is applied to the SIFFT burst-mode data processing algorithm, especially when the highest possible SNR is desired.

9 Acknowledgments

The authors express their appreciation to MacDonald Dettwiler (MDA), the Natural Sciences and Engineering Research Council (NSERC) of Canada, the British Columbia Advanced Systems Institute (ASI) and the Science Council of British Columbia for funding this research.

10 References

1 PAPOULIS, A.: 'Signal analysis' (McGraw-Hill, 1977)

2 BITMEAD, R.R., and ANDERSON, B.D.O.: 'Adaptive frequency sampling filters', *IEEE Trans. Circuits Syst.*, June 1981, **CAS-28**, pp. 524–534

3 DUDAS, J.: 'The momentary Fourier transform'. PhD thesis, Technical University of Budapest, 1986

4 LILLY, J.H.: 'Efficient DFT-based model reduction for continuous systems', *IEEE Trans. Autom. Control*, Oct. 1991, **36**, pp. 1188–1193

5 STRANG, J.: 'Linear algebra and its applications' (Saunders HBJ, 1988)

6 ALBRECHT, S., CUMMING, I., and DUDAS, J.: 'The momentary Fourier transformation derived from recursive matrix transformations'. Proceedings of the 13th international conference on *Digital signal processing*, 2–4 July 1997, Santorini, Greece, pp. 337–340

7 SHERLOCK, B.G., and MONRO, D.M.: 'Moving discrete Fourier transform', *IEE Proc. F*, Aug. 1992, **139**, pp. 279–282

8 CUMMING, I., and LIM, J.: 'The design of a digital breadboard processor for the ESA remote sensing satellite synthetic aperture radar'. Final report for ESA contract 3998/79/NL/HP(SC) Technical July 1981, MacDonald Dettwiler, Richmond, BC, Canada

9 SACK, M., ITO, M., and CUMMING, I.G.: 'Application of efficient linear FM matched filtering algorithms to SAR processing', *IEE Proc. F*, 1985, **132**, (1), pp. 45–57

10 RANEY, R.K., LUSCOMBE, A.P., LANGHAM, E.J., and AHMED, S.: 'RADARSAT', *Proc. IEEE*, June 1991, **79**, pp. 839–849

11 BAMLER, R., and EINEDER, M.: 'ScanSAR processing using standard high precision SAR algorithms', *IEEE Trans. Geosci. Remote Sens.*, Jan. 1996, **34**, p. 212–218

12 WONG, F., STEVENS, D., and CUMMING, I.: 'Phase-preserving processing of scanSAR data with a modified range doppler algorithm'. Proceedings of the International symposium on *Geoscience and remote sensing* IGARSS'97, August 1997, Singapore pp. 725–727

13 CUMMING, I., WONG, F., GUO, Y.: 'Modifying the RD algorithm for burst-mode SAR processing'. Proceedings of the European conference on *Synthetic aperture radar*, EUSAR'98 May 1998, Friedrichshafen, Germany pp. 477–480

11 Appendix: SIFFT efficiency calculations

This Appendix provides the detailed arithmetic computation calculations of the IMFT used in the SIFFT algorithm as reported in Section 7.3.3. The number of IFFTs needed to compress bursts with the highest energy is

$$M_{IFFT} = 2\frac{N_{IFFT}}{N_{FFT}} N_b \frac{1}{G_{group}} \tag{36}$$

where N_{FFT} and N_{IFFT} are the forward and inverse transform lengths, N_b is the number of samples per burst, and G_{group} is the number of correctly compressed targets in each group (burst) by each IFFT. From eqn. 36 M_{IFFT} is inversely proportional to G_{group} as, if more good targets are compressed from each group, fewer IFFTs are needed to form a contiguous set of output points. Then the number of operations needed to compress all the targets using the IFFT algorithm is

$$OP_{IFFT} = M_{IFFT}\ OP_{N_{IFFT}} \tag{37}$$

where $OP_{N_{IFFT}}$ is the number of operations needed for one N-sample mixed-radix IFFT. If N is power of 2 then $OP_{N_{IFFT}} = 5N\log_2 N$. As seen in Section 5, in case of the full-IMFT algorithm, $OP_{IMFT} = M\,(8\,N_{IFFT} + 2)$ real operations are needed to analyse an M-point complex data record. In the case of two-beam burst processing, $M = 3\ BW_{bin}$ so the arithmetic of the full-IMFT algorithm is

$$OP_{IMFT} = 3BW_{bin}(8N_{IMFT} + 2) \tag{38}$$

During the IDFT extraction in the SIFFT algorithm only a portion of the output target space ('good' output targets G_{IFFT}) is compressed correctly. Although the number of the good targets remains the same through the processing of a range cell, their position changes with the position of the IDFTs. So, a simple reduced-IMFT algorithm cannot be used for the target extraction. The position of the computed spectrum coefficients has to change in phase with position of the "not-compressed" good output samples, and the Doppler-frequency coefficients of the targets which are already compressed do not have to be computed during the rest of the processing. The arithmetic of the required reduced-MFT algorithm follows.

At first the reduced-IMFT algorithm has to be applied N_{IMFT} times to give the first valid compression result. This requires

$$OP_{IMFT_{Reduced1}} = N_{IMFT}(8N_{IMFT} + 2) \tag{39}$$

real operations. When the first IMFT is done G_{IFFT} targets are compressed correctly, so in the next IMFT $N_{Reduced1} = N_{IMFT} - G_{IFFT}$ targets have to be compressed. Then the IMFT window is shifted q times a sample at a time, till it fully covers the next target in the group (see target 7 in Fig. 10). Now, G_{group} number of targets can be extracted correctly, so during the next shift of q samples, $N_{reduced} - G_{group}$ targets need to be extracted. The calculation of the spectrum coefficients from the previously reduced whole output target cell repeats $(3BW_{bin} - N_{IFFT})/q$ times through the processing region until all the targets get compressed. It can be shown that the arithmetic of this procedure is

$$\begin{aligned} OP_{IMFT_{reduced2}} &= 2(3BW_{bin} - N_{IMFT}) \\ &\quad \times \left[4(N_{IMFT} - G_{IFFT}) \right. \\ &\quad \left. - 2G_{IFFT}\left(\frac{3BW_{bin} - N_{IMFT}}{q} - 1\right) + 1\right] \end{aligned} \tag{40}$$

Then

$$\begin{aligned}OP_{IMFT_{reduced}} &= OP_{IMFT_{reduced1}} + OP_{IMFT_{reduced2}}\\ &= N_{IMFT}(8N_{IMFT}+2) + 2[3BW_{bin} - N_{IMFT}]\\ &\quad \times \left[4(N_{IMFT} - G_{IFFT})\right.\\ &\quad \left. - 2G_{IFFT}\left(\frac{3BW_{bin} - N_{IMFT}}{q} - 1\right) + 1\right] \qquad (41)\end{aligned}$$

real operations are needed to compress all targets using the reduced-IMFT algorithm. The arithmetic of the reduced-IMFT depends on G_{IFFT} and G_{group}, thus if more targets get extracted by an IMFT, less computation is needed. The implementation of the reduced IMFT is the same as the full-IMFT algorithm, except that you must recognise which targets have already been computed by the IMFT.