Mixed-radix approach to incremental DFT refinement

Joseph M. Winograd and S. Hamid Nawab

ECS Department, Boston University
44 Cummington St., Boston, MA 02215

ABSTRACT

Incremental refinement algorithms can quickly produce approximate results and may then improve the quality of those results in subsequent stages of computation. They offer promise for the development of real-time systems whose performance degrades gracefully under diminishing hard deadlines. We present a new class of incremental refinement algorithms which employ mixed-radix signal representations for the calculation of successive approximations to the DFT. This class includes algorithms with a wide range of cost/quality tradeoff characteristics. This work generalizes a previously reported class of algorithms which employ binary signal representations only. The mixed-radix formulation allows solutions of a given level of quality to be achieved using significantly fewer arithmetic operations in many instances. Under certain restrictions, these algorithms can also be implemented with no computational overhead using fixed-point binary hardware.

Keywords: approximate processing, incremental refinement algorithms, successive approximations, discrete Fourier transform, short-time Fourier transform, mixed-radix numbering systems, real-time systems

1 INTRODUCTION

The next generation of real-time signal processors will be called upon to perform increasingly demanding tasks within complex and dynamically evolving environments. The success with which these systems are deployed will depend upon their ability to respond to changing deadline times, variations in the complexity of computational tasks, and fluctuations in the availability of computing resources. Established design techniques simply will not scale to meet the demands of such systems, and new approaches[1] are required.

In the Systems and Artificial Intelligence communities there has been an increasing interest in the realization of real-time systems through the use of approximate processing that allows systematic tradeoffs to be made between resource usage and output quality. The goal of this approach is to enable systems to adapt their performance to problem complexity, current deadlines, and resource availability. Such systems can continuously maximize their performance within the constraints imposed by the currently available resources and, in this way, offer graceful degradation of performance in adverse circumstances as an alternative to system failure. Much of this work is reviewed in two recent publications.[2,3] Of particular importance in this approach is that algorithms be available that can quickly produce a usable approximation and can then improve its quality incrementally.

We are currently investigating incremental refinement algorithms for approximate digital signal processing and have recently reported a family of algorithms[4] for computing successive approximations to the DFT. Based upon a two's complement binary encoding of the signal under analysis, these algorithms allow a wide variety of tradeoffs between output quality and computational cost to be achieved in successive refinement stages. We have

 0-8194-1922-2/95/$6.00

now extended these to a larger class of incremental refinement algorithms by considering a larger set of signal representations. We show that this enables equivalent sequences of successive approximation to be obtained using fewer arithmetic operations, and we describe a simple technique for efficiently implementing these algorithms using fixed-point two's complement binary arithmetic.

We begin by introducing the mixed radix complement representation, a class of numbering systems based on the use of the radix complement convention in a mixed radix setting. In section 3 this numbering system is used to develop a class of algorithms for incremental DFT refinement. A method for selecting the signal representation which minimizes the number of arithmetic operations needed to produce a given level of output quality is derived in section 4. Section 5 describes techniques for supporting mixed radix complement representations efficiently in standard binary hardware. This is followed by a brief performance example.

2 MIXED RADIX COMPLEMENT REPRESENTATIONS

Consider the class of nonredundant, positional, and weighted mixed-radix numbering systems which employ a fixed number of unsigned digits.[5,6] Numbering systems in this class can be uniquely identified by the number of digits used for each number, say D, and a D-tuple of radices associated with the digit positions, denoted $(m_{D-1}, m_{D-2}, \ldots, m_1, m_0)$. Without loss of generality, we shall assume that the radix point is fixed directly to the right of the least significant digit of this representation.* In any such system, a total of

$$Q = \prod_{d=0}^{D-1} m_d \tag{1}$$

different numbers can be constructed from unique sequences of digits, which we denote $x_{D-1}x_{D-2}\ldots, x_1x_0$, with $x_i \in \{0, 1, 2, \ldots, m_i - 1\}$ for $0 \leq i \leq D-1$. We refer to a digit sequence of this kind as a *number* and the quantity that it represents as its *value*.

The conventional method for assigning a positive numerical value, which we denote as x^+, to each of these digit sequences is according to the relation

$$\begin{aligned} x^+ &= x_0 + x_1 \cdot m_0 + x_2(m_1 \cdot m_0) + \ldots + x_{D-1}(m_{D-2} \cdot m_{D-3} \cdot \ldots \cdot m_1 \cdot m_0) \\ &= \sum_{d=0}^{D-1} x_d \beta_d \end{aligned} \tag{2}$$

where

$$\beta_d = \begin{cases} 1, & d = 0, \\ \prod_{j=0}^{d-1} m_j, & 1 \leq d \leq D-1, \end{cases} \tag{3}$$

This interpretation allows only unsigned quantities to be represented, and covers the range of integers $[0, Q-1]$.

Complement representations[5] can also be used within the context of mixed-radix systems. In particular, we can employ a *mixed radix complement* interpretation that is analogous to radix complement in the fixed-radix case.† Following radix complement, we define the value x of a mixed radix complement representation to be

$$x = \begin{cases} x^+, & 0 \leq x^+ \leq (Q/2) - 1, \\ x^+ - Q, & Q/2 \leq x^+ \leq Q - 1, \end{cases} \tag{4}$$

This enables the representation of values in the range $[-Q/2, Q/2 - 1]$, generalizing the asymmetry of radix complement representations for fixed radices. If we restrict our discussion to numbering systems for which m_{D-1}

*This is because the value of a number whose radix point lies to the immediate left of its rth digit is always related to its value with the radix point to the right of its least significant digit by a constant multiple.

†We use the term "radix complement" to refer explicitly to fixed-radices, and prefix it with "mixed" when discussing the more general class of representations.

$x_2x_1x_0$	000	001	010	011	100	101	110	111	200	201	210	211	300	301	310	311
x^+	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
x	0	1	2	3	4	5	6	7	-8	-7	-6	-5	-4	-3	-2	-1

Table 1: The mapping of numbers to values in the $(4, 2, 2)$ mixed radix system. Each number (a digit sequence $x_2x_1x_0$) is shown with its unsigned value (x^+) and its mixed radix complement value (x).

is even, another expression relating numbers to their mixed radix complement value is:

$$x = \sum_{d=0}^{D-1} \alpha(x_d, d)\beta_d \tag{5}$$

where

$$\alpha(y, d) = \begin{cases} y, & (d \neq D-1) \vee (0 \leq y \leq (m_{D-1}/2) - 1), \\ y - m_{D-1}, & (d = D-1) \wedge (m_{D-1}/2 \leq y \leq m_{D-1} - 1), \end{cases} \tag{6}$$

and β_d is as defined in equation (3). The equivalence of (4) and (5) for even m_{D-1} is easily shown. Table 1 illustrates the mapping of numbers to values for the $(4, 2, 2)$ numbering system using both unsigned and mixed radix complement interpretations.

3 INCREMENTAL REFINEMENT OF DFT APPROXIMATIONS

The mixed radix complement representations described in the previous section offer a generalization of the more familiar radix complement methods. As such, one can consider their use within the context of derivations for which radix complement is suitable, but the broader class may offer some advantage. We propose to do just this in the context of a class of methods for producing successive approximations to the DFT.

3.1 Successive approximations of the DFT

Suppose that the N-point signal frame $x(n)$ is real valued, windowed to length $N_w \leq N$, and encoded using a D-digit mixed radix complement representation with radices $(m_{D-1}, m_{D-2}, \ldots, m_1, m_0)$. We denote by $x_d(n)$ the dth digit of the nth sample of $x(n)$. For each value of $0 \leq d \leq D-1$, $x_d(n)$ can be considered an N-point *digit vector*, indexed by n. With the requirement that m_{D-1} be even, $x(n)$ can be related to the digit vectors $x_d(n)$, through equation (5), by

$$x(n) = \sum_{d=0}^{D-1} \alpha(x_d(n), n)\beta_d, \qquad 0 \leq n \leq N-1 \tag{7}$$

The DFT of $x(n)$ can then be expressed as

$$X(k) = \sum_{n=0}^{N-1} \left(\sum_{d=0}^{D-1} \alpha(x_d(n), d)\beta_d \right) e^{-j\frac{2\pi}{N}kn}, \qquad 0 \leq k \leq N-1 \tag{8}$$

Using a backward differencing approach,[7,8] we can produce an alternative expression for $X(k)$:

$$X(k) = \sum_{d=0}^{D-1} \sum_{n=0}^{N-1} g_d(n) G_{n,d}(k), \qquad 1 \leq k \leq N-1 \tag{9}$$

where

$$g_d(n) = \begin{cases} \alpha(x_d(0), d) - \alpha(x_d(N-1), d), & n = 0, \\ \alpha(x_d(n), d) - \alpha(x_d(n-1), d), & 1 \le n \le N-1, \end{cases} \tag{10}$$

and

$$G_{n,d}(k) = \beta_d \frac{e^{-j2\pi kn/N}}{1 - e^{-j2\pi k/N}} \tag{11}$$

The values of $X(k)$ in equations (8) and (9) differ only in that the DC component of (9) has been lost in the differencing operation of (10).

Following a derivation[4] which has been applied to binary encoded signals, we can use equation (9) as the basis for defining a class of successive approximations to the DFT of $x(n)$. The ith successive approximation, $\hat{X}_i(k)$, is defined to be

$$\hat{X}_i(k) = \sum_{d=D-v_i}^{D-1} \sum_{n=0}^{r_i-1} g_d(n) G_{n,d}(k), \qquad 1 \le k \le c_i \tag{12}$$

where the indexing bounds c_i, r_i, and v_i are control variable sequences which characterize the successive approximations. The control variables are naturally constrained by $1 \le c_i \le N/2$, $1 \le r_i \le \min(N_w + 1, N)$, and $1 \le v_i \le D$. In order to achieve approximations of monotonically increasing quality, we also require that they be nondecreasing with i and that $c_{i+1} + r_{i+1} + v_{i+1} > c_i + r_i + v_i$.

The quality of each approximation obtained in this way is a function of the corresponding values of the control variables. As with binary representations,[9] the frequency coverage of the ith successive approximation, denoted $q_{c,i}$ and measured in radians, is

$$q_{c,i} = \frac{2\pi c_i}{N} \tag{13}$$

and its frequency resolution may be shown to be approximately

$$q_{r,i} = r_i \tag{14}$$

with $q_{r,i}$ being the number of resolvable frequency components in $\hat{X}_i(k)$. Assuming $x(n)$ to be well-scaled with respect to the uniformly divided quantization range $[-Q/2, Q/2-1]$, the noise introduced by reducing the effective number of quantization levels can be shown to produce a SNR (in dB) of about

$$q_{v,i} = 20 \sum_{d=D-v_i}^{D-1} \log m_d \tag{15}$$

after the ith approximation.

3.2 Incremental DFT refinement algorithms

A proposed approach to the calculation of a sequence of successive approximations is the use of an incremental refinement algorithm which at each stage of computation improves the previous approximation via update equations.[4] Each successive approximation of (12) is related to the previous one by:

$$\hat{X}_i = \begin{cases} \hat{X}_{i-1}(k) + C_i(k), & c_{i-1} < k \le c_i, \\ \hat{X}_{i-1}(k) + R_i(k) + V_i(k), & 1 \le k \le c_{i-1}, \end{cases} \tag{16}$$

where $C_i(k)$ is the *coverage update*, which is defined as

$$C_i(k) = \sum_{d=D-v_i}^{D-1} \sum_{n=0}^{r_i} g_d(n) G_{n,d}(k), \tag{17}$$

$R_i(k)$ is the *resolution update*, which is defined as

$$R_i(k) = \sum_{d=D-v_i}^{D-1} \sum_{n=r_{i-1}+1}^{r_i} g_d(n) G_{n,d}(k), \tag{18}$$

and $V_i(k)$ is the *SNR update*, which is defined as

$$V_i(k) = \sum_{d=D-v_i}^{D-v_{i-1}-1} \sum_{n=0}^{r_{i-1}} g_d(n) G_{n,d}(k). \tag{19}$$

Here, $c_0 = r_0 = v_0 = 0$ and $\hat{X}_0(k) = 0$ for all k. Using stored pre-computed values for the terms of the above summation, and omitting those terms for which $g_d(n) = 0$, the update equations through stage i may be directly evaluated at a computational cost of

$$k_i = 2c_i r_i v_i \hat{\gamma}_i \tag{20}$$

where

$$\gamma_i = \frac{1}{r_i v_i} \sum_{d=D-v_i}^{D-1} \sum_{n=0}^{r_i-1} (1 - \delta(g_d(n))) \tag{21}$$

k_i represents the total computational cost of generating $\hat{X}_i(k)$ from $\hat{X}_0(k)$ and has the units of real additions.* It is a function of γ_i, the fraction of non-zero elements in the portion of $g_d(n)$ over which computation is performed, which is itself defined using the Dirac unit impulse function. This algorithm requires that $\sum_{d=0}^{D-1}(m_d - 1)N_w N$ real values be pre-computed and stored in memory. By replacing each addition operation with a scaling (by shifting the radix point) and a real addition, the storage requirement can be reduced to $\max_{0 \le d \le D-1}(m_d - 1)N_w N$ real values.

3.3 Frequency reversal for mixed radix complement representations

A primary characteristic of these incremental refinement algorithms is the dependence of their quality/cost tradeoffs on the signal data. Their computational cost has been observed to vary according to the frequency content of the signal under analysis, and a *frequency reversal* technique[10,8] has been proposed which reduces the cost of analyzing signals with significant high-frequency energy. This technique can be applied when using mixed radix complement representations, however some additional issues should be addressed.

The frequency reversal technique was originated in the context of single-digit signed digit signal representations. In it, the signal under analysis is multiplied by the signal $(-1)^n$, causing the frequency spectrum from 0 to π rads to be effectively flipped about $\pi/2$ rads. When computationally advantageous, spectral analysis is then performed on the modulated signal instead of the original, and the results reordered to correct for the frequency reversal.

Direct application of this approach when using mixed radix complement representations can be problematic because the individual digit vectors $x_d(n)$ are comprised of unsigned digits only. Thus, multiplication of each digit vector by $(-1)^n$ results in a signal that can no longer be represented in mixed radix complement form. An alternative method of performing frequency reversal is available, however, which does not require multiplication by -1. We define the mixed radix complement frequency reversed signal $r_d(n)$, for fixed $0 \le d \le D-1$, as:

$$r_d(n) = \begin{cases} x_d(n), & n \text{ even}, \\ \overline{x_d(n)}, & n \text{ odd}, \end{cases} \tag{22}$$

*The computation required for producing $g_d(n)$ from $x_d(n)$ is omitted from this metric for simplicity.

where $\overline{x}$ is the complement of the digit x (i.e. $\overline{x_d(n)} = (m_d - 1) - x_d(n)$). The signal $r_d(n)$ can be equivalently expressed as

$$r_d(n) = x_d(n)(-1)^n + m_d \sum_{l=0}^{\frac{N}{2}-1} \delta(n - 2l - 1) \tag{23}$$

Restricting N to be even, we can derive the relationship between $X_d(k)$, the DFT of $x_d(n)$, and $R_d(k)$, the DFT of $r_d(k)$ from equation (23):

$$\begin{aligned} R_d(k) &= \sum_{n=0}^{N-1} x_d(n)(-1)^n e^{-j\frac{2\pi}{N}kn} + m_d \sum_{n=0}^{N-1} \sum_{l=0}^{N/2-1} \delta(n - 2l - 1) e^{-j\frac{2\pi}{N}kn} \\ &= X_d\left((k + N/2) \bmod N\right) + m_d \sum_{n=0}^{N-1} e^{-j\frac{2\pi}{N}(2l+1)k} \\ &= X_d\left((k + N/2) \bmod N\right) + m_d \frac{N}{2}\left(\delta(k) - \delta(k - N/2)\right) \end{aligned} \tag{24}$$

We see that the frequency spectrum obtained from $r_d(n)$, is a frequency reversed version of $X_d(k)$, with a constant factor added to the highest and lowest frequency measurements. Thus, when N is even, by using the complement operation as in equation (22) we can obtain the full benefits of the frequency reversal technique while maintaining the unsigned digit representation for $x_d(n)$.

4 EFFICIENCY ANALYSIS FOR RADIX SELECTION

The numbering system used for signal representation with this approach to incremental DFT refinement is of fundamental importance to the cost/quality tradeoff achieved in successive refinement stages. To motivate our analysis of their relationship, we begin with a brief example. Consider an initial approximation to a DFT with $N = 256$ and $N_w = 128$, for which $q_{c,1} = \pi/8$ rads, $q_{r,1} = 32$ components, and $q_{v,1} = 12$ dB. Using two's complement binary format signal representation, (i.e. $\forall d : m_d = 2$), we derive from equations (13)-(15) the control values $c_1 = 16$, $r_1 = 32$, and $v_1 = 2$. Assuming the signal under analysis to be comprised of independent and uniformly distributed values, we put $\gamma_1 = 0.5$. The computational cost of performing this approximation is, by equation (20), $k_1 = 1024$ additions. A mixed radix complement representation, radix $(4, 2, 4)$ say, can also be used to represent the signal frame, and an identical first approximation can be computed from it using the control values $c_1 = 16$, $r_1 = 32$, and $v_1 = 1$. Again assuming the signal frame to possess an independent and uniform distribution of quantization levels, we let $\gamma_1 = 0.75$. This approximation can be computed for only $k_1 = 768$ additions, a reduction in cost of 25%.

This example hints at the importance of proper radix selection for efficient computation and leads us to ask: what signal representation minimizes the total number of arithmetic operation needed to produce an approximation of a given quality? Under assumptions similar to those made in our example above, this question is answered definitively by the following theorem.

THEOREM 1. *Fix i as any positive integer. Put $q_{c,i}$, $q_{r,i}$, and $q_{v,i}$ as the approximation quality achieved in frequency coverage, resolution, and SNR after i successive approximations to the DFT have been performed using the algorithm described in section 3. Let Q be the total number of signal quantization levels incorporated through stage i so that $q_{v,i} = 20 \log Q$. Assuming the signal under analysis to be independent and uniformly distributed across quantization levels, the total computational cost of achieving a solution of the quality given by $q_{c,i}$, $q_{r,i}$, and $q_{v,i}$ is minimized by refining over a single digit vector of the mixed radix complement signal representation with $m_0 = Q$.*

PROOF: We begin by remarking that each mixed radix numbering system that represents a signal uniformly quantized to Q levels is defined by a tuple of radices which form an integer factorization of Q. So, let us denote

by $\mathfrak{M}_Q$ the set of all factorizations of Q over $\mathbb{N}\backslash\{1\}$, where each $\mathbf{m} \in \mathfrak{M}_Q$ is a D-tuple (for some $D > 0$) of integer factors $\mathbf{m} = (m_{D-1}, m_{D-2}, \ldots, m_0)$ so that $Q = \prod_{d=0}^{D-1} m_d$. Since we are considering the computation only through stage i, no generality is lost by assuming that all digit vectors are used through stage i. Thus, we put $v_i = D$.*

Under the assumptions for signal value distribution, the fraction of non-zero elements in $g_d(n)$ is

$$\gamma_i = \frac{1}{D} \sum_{d=0}^{D-1} \frac{m_d - 1}{m_d} \tag{25}$$

Using equation (20) with fixed $c_i = (N/2\pi)q_{c,i}$, $r_i = q_{r,i}$, and $v_i = D$, we see that

$$k_i = C \sum_{d=0}^{D-1} \frac{m_d - 1}{m_d} \tag{26}$$

for some constant C dependent upon $q_{c,i}$, $q_{r,i}$, and $q_{v,i}$. Our challenge is to minimize equation (26) uniquely. We will identify its minima by demonstrating, equivalently, that for all Q, the function

$$k(\mathbf{m}) = \sum_{d=0}^{D-1} \frac{m_d - 1}{m_d} \tag{27}$$

is minimized over $\mathfrak{M}_Q$ by $\mathbf{m} = (Q)$.

Since k has no dependence on the ordering of factors in $\mathbf{m}$, we will consider all factorizations which are reorderings of the same factors to be equivalent and, without loss of generality, consider as canonical those D-tuples for which $m_d \geq m_{d-1}$ and restrict our discussion to them.

Let us denote by $\mathfrak{M}_Q^D$ the subset of $\mathfrak{M}_Q$ which contains all factorizations of Q with D factors, and denote by $\mathbf{m}_D$ an element of $\mathfrak{M}_Q^D$. We will first establish the form of $\mathbf{m}_D^*$, the factorization that minimizes k over each $\mathfrak{M}_Q^D$. Put p_n for $0 \leq n \leq N-1$ as prime factors of Q (with repeats) so that $p_n \geq p_{n-1}$ and $Q = \prod_{n=0}^{N-1} p_n$. Obviously, $\mathfrak{M}_Q^D \neq \emptyset$ for $1 \leq D \leq N$. We claim that

$$\min_{\mathbf{m}_D \in \mathfrak{M}_Q^D} k(\mathbf{m}_D) = k(\mathbf{m}_D^*) \text{ with } \mathbf{m}_D^* = \left(\left(\prod_{n=D-1}^{N-1} p_n\right), p_{D-2}, p_{D-3}, \ldots, p_0\right) \tag{28}$$

This will be shown by comparison of $k(\mathbf{m}_D^*)$ with $k(\mathbf{m}'_D)$, where $\mathbf{m}_D^* \neq \mathbf{m}'_D = (m'_{D-1}, m'_{D-2}, \ldots, m'_0) \in \mathfrak{M}_Q^D$. Now,

$$\begin{aligned} k(\mathbf{m}'_D) - k(\mathbf{m}_D^*) &= \left(\sum_{d=0}^{D-1} \frac{m'_d - 1}{m'_d}\right) - \left(\frac{\left(\prod_{n=D-1}^{N-1} p_n\right) - 1}{\prod_{n=D-1}^{N-1} p_n} + \sum_{n=0}^{D-2} \frac{p_n - 1}{p_n}\right) \\ &= \left(D - \sum_{d=0}^{D-1} \frac{1}{m'_d}\right) - \left(D - \frac{1}{\prod_{n=D-1}^{N-1} p_n} - \sum_{n=0}^{D-2} \frac{1}{p_n}\right) \\ &= \frac{1}{\prod_{n=D-1}^{N-1} p_n} + \left(\sum_{n=0}^{D-2} \frac{1}{p_n}\right) - \left(\sum_{d=D-1}^{N-1} \frac{1}{m'_d}\right) \end{aligned}$$

Since $p_{D-2}, \ldots, p_0$ are the smallest prime factors of Q, then $p_i \leq m'_i$ for $0 \leq i \leq D-2$. Further, the strict inequality must hold for at least one of these, because $\mathbf{m}_D^* \neq \mathbf{m}'_D$, Consequently.

$$\begin{aligned} &> \frac{1}{\prod_{n=D-1}^{N-1} p_n} + \frac{1}{m'_0} \\ &> 0 \end{aligned} \tag{29}$$

*Obviously, if further improvement in SNR is subsequently performed, additional digit vectors may be present in the signal representation.

The last step is based on the observation that $m_0' < \prod_{n=D-1}^{N-1} p_n$, and finalizes our proof of the claim (28).

Having established the member of $\mathfrak{M}_Q^D$ which minimizes k, we will now demonstrate that $\mathbf{m}_D^*$ are ordered in D, and specifically that

$$k(\mathbf{m}_D^*) < k(\mathbf{m}_{D+1}^*) \text{ when } \mathfrak{M}_Q^D \neq \emptyset,\ \mathfrak{M}_Q^{D+1} \neq \emptyset \tag{30}$$

This can be shown directly as follows:

$$\begin{aligned} k(\mathbf{m}_{D+1}^*) - k(\mathbf{m}_D^*) &= \left((D+1) - \frac{1}{\prod_{n=D}^{N-1} p_n} - \sum_{d=0}^{D-1} \frac{1}{p_n}\right) - \left(D - \frac{1}{\prod_{n=D-1}^{N-1} p_n} - \sum_{n=0}^{D-2} \frac{1}{p_n}\right) \\ &= 1 - \frac{1 - p_{D-1}}{\prod_{n=D-1}^{N-1} p_n} - \frac{1}{p_{D-1}} \\ &= \frac{\left(1 + \prod_{n=D-1}^{N-1} p_n\right)(p_{D-1} - 1)}{\prod_{n=D-1}^{N-1} p_n} \\ &> 0 \end{aligned} \tag{31}$$

This assertion relies only upon the fact that $p_n > 1$ for $0 \leq n \leq N-1$, and proves claim (30). Clearly, from (28), $\forall Q : \exists \mathbf{m}_1^* = (Q) \in \mathfrak{M}_Q^1$. This $\mathbf{m}_1^*$ is shown in (30) to minimize k over all $\mathfrak{M}_Q$. This radix also minimizes k_i, and the theorem is proved. ▌

The implications of this theorem for radix selection are quite clear. To minimize computational cost, one should always use the largest radix representation which provides the desired quality in SNR. Thus, when a refinement stage with an improvement of SNR of greater than 6 dB* is desired, a higher radix representation will always offer an improvement over the use of a binary signal representation. How much of an improvement can we expect? A simple efficiency analysis can be performed by restricting our consideration only to fixed-radix representations. Suppose we are computing an approximation with the quality $q_{c,i}$, $q_{r,i}$, and $q_{v,i}$ for some i, and let Q be defined as in the theorem. Suppose we are using a radix-p representation, so $Q = p^{v_i}$. Again assuming the signal frame to be comprised of an independent and uniform distribution of quantization levels, we let $\gamma_i = (p-1)/p$. It can then be easily shown that

$$k_i = C' \frac{p-1}{p \log p} \tag{32}$$

for some constant C' that depends on $q_{c,i}$, $q_{r,i}$, and $q_{v,i}$. This expression allows numeric comparisons of the relative cost of various (fixed) radices to be made. These costs are shown in Fig. 1 for radices 2 through 32, with the cost incurred using binary encoding normalized to unity.

5 MIXED RADIX COMPLEMENT ARITHMETIC USING FIXED-POINT BINARY HARDWARE

Donald Knuth has observed[6] that the number $\ldots a_3 a_2 a_1 a_0$ in unsigned radix p is equivalent to the number $\ldots x_3 x_2 x_1 x_0$ in unsigned radix p^u where each x_d has the same value as the number $a_{ud+u-1} \ldots a_{ud+1} a_{ud}$ has in radix p. A similar equivalence exists between mixed radix complement representations with power-of-two radices and two's complement binary. This relationship can be exploited to achieve the flexibility and efficiency of higher radix representations when using general purpose two's complement fixed-point binary hardware.

Consider the D-digit mixed radix complement representation of the value x in a numbering system with $m_d = 2^{u_d}$ for $0 \leq d \leq D-1$ and $u_d \in \mathbb{N}$, and put Q as in equation (1). This number $x_{D-1} x_{D-2} \ldots x_0$ can

*$20 \log 2 \approx 6$ dB is the increase in SNR when an additional binary digit vector in the approximation.

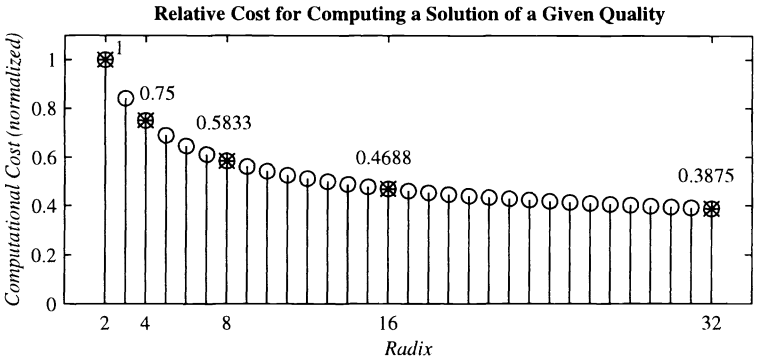

Figure 1: The predicted relative costs of obtaining a solution of a given quality using various fixed-radix representations. These costs were obtained using the relation $(p-1)/p\log p$ and normalized. The numerical values obtained for radices 2, 4, 8, 16, and 32 are shown.

be encoded as a binary number where each of the digits x_d is represented by a group of u_d bits as follows. Let $a_{B-1}a_{B-2}\ldots a_0$ be a binary number containing of $B = \log_2 Q$ bits. Using the mapping suggested by Knuth, we formally define the values of these bits so that

$$x_d = \sum_{b=s_d}^{s_d+u_d-1} a_b 2^{b-s_d} \tag{33}$$

where $s_d = \log_2 \beta_d$. The bits $a_{s_d+u_d-1}\ldots a_{s_d+1}a_{s_d}$ are the unsigned binary representation of the digit x_d. A diagram of the relationship between the mixed radix complement representation, the binary representation, and m_d, u_d, and s_d is given in Fig. 2.

To establish the equivalence of this mapping with respect to the mixed radix complement and two's complement binary representations, we must show that the mixed radix complement value, x, of $x_{D-1}x_{D-2}\ldots x_0$ is equal to the two's complement binary value, a, of $a_{B-1}a_{B-2}\ldots a_0$. This can be done using the relation mapping digits to value given in equation (5):

$$\begin{aligned} a &= \sum_{b=0}^{B-1} \alpha(a_b, b)2^b \\ &= \alpha(a_{B-1}, B-1)2^{B-1} + \sum_{b=s_{B-1}}^{B-2} a_b 2^b + \sum_{d=0}^{D-2} \sum_{b=s_{d-1}}^{s_d+u_d-1} a_b 2^b \\ &= \alpha(a_{B-1}, B-1)2^{B-1} + \sum_{b=s_{B-1}}^{B-2} a_b 2^b + \sum_{d=0}^{D-2} \alpha(x_d, d)\beta_d \end{aligned} \tag{34}$$

The function β_d used here is the one associated with the numbering system for $x_{D-1}x_{D-2}\ldots x_0$. Now, if $a_{B-1} = 0$, then $x_{D-1} < m_{D-1}/2$, and consequently

$$\begin{aligned} \alpha(a_{B-1}, B-1)2^{B-1} + \sum_{b=s_{B-1}}^{B-2} a_b 2^b &= \sum_{b=s_{B-1}}^{B-1} a_b 2^b \\ &= \alpha(x_{D-1}, D-1)\beta_{D-1} \end{aligned} \tag{35}$$

Else, $a_{B-1} = 1$, so $x_{D-1} \geq m_{D-1}/2$ and

$$\alpha(a_{B-1}, B-1)2^{B-1} + \sum_{b=s_{B-1}}^{B-2} a_b 2^b = -2^{B-1} + \sum_{b=s_{B-1}}^{B-2} a_b 2^b$$

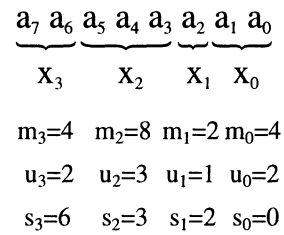

Figure 2: The relationship between the digits $x_3 \ldots x_0$ of a $(4, 8, 1, 4)$ mixed radix complement number, its mapping into a binary number $a_7 \ldots a_1 a_0$, and the variables used in the text to describe the mapping.

$$\begin{aligned} &= \left(2^{B-1} + \sum_{b=s_{B-1}}^{B-2} a_b 2^b\right) - 2^B \\ &= (x_{D-1} - m_{D-1})\,\beta_{D-1} \\ &= \alpha(x_{D-1}, D-1)\beta_{D-1} \end{aligned} \tag{36}$$

Considering equations (34)-(36) in conjunction with equation (5), it is apparent that $a = \sum_{d=0}^{D-1} \alpha(x_d, d)\beta_d = x$.

The equivalence of value maintained through the mapping of equation (33) establishes an isomorphism between arithmetic operations on mixed radix complement and two's complement binary representations. As long as the conventions of two's complement binary arithmetic are observed, such as extension of the sign bit to the MSB of the word, the two numerical representations can be considered equivalent, and the results from any mathematical operation can be interpreted in either. In this way, mixed radix complement numbers can be stored in binary encoded form and used directly in binary calculations by grouping bits together. The results may be interpreted in mixed radix complement by selecting groups of bits in a similar manner.

6 EXAMPLE

The computational efficiency of higher radix representations has been verified by applying our algorithms for incremental DFT refinement to the approximation of the discrete STFT.[11] A recording of a flute playing two successive notes (sampled at 8 kHz and quantized to 256 levels) was represented in radix 4 and analyzed by applying a sequence of four DFT approximations to each STFT signal frame. The quality of these successive approximations was $q_{c,i} = (65\pi/128, 75\pi/128, 85\pi/128, 95\pi/128)$, $q_{r,i} = (45, 85, 85, 120)$, and $q_{v,i} = (12, 12, 12, 12)$. The STFT parameters used were $N = 256$, $L = 64$, and $N_w = 128$ (rectangular windowing was applied). The results of these approximations, shown in Fig. 3(a)-(d), each required about 55% of the arithmetic operations required to produce results of the same quality using a binary signal representation.*

We have previously reported[4] an analysis of the same signal using a binary signal representation. In comparison with those results, the last two approximation stages here produce identical results at a significantly reduced total cost. The first and second stages generate results of higher quality but require more computation. The use of radix 4 makes the quality increments of 6 dB used in the previous example unrealizable, highlighting an important tradeoff inherent in the mixed radix framework. While higher radix representations are more efficient overall, using them requires that a sacrifice be made in the granularity of the quality increments. In order to achieve fine gradations in SNR, less efficient smaller radices must be used.

*This percentage is predicted by equation (32) to be 75%. The difference between the predicted and observed values is due to the pessimistic assumption of an uncorrelated signal in equation (32)'s derivation.

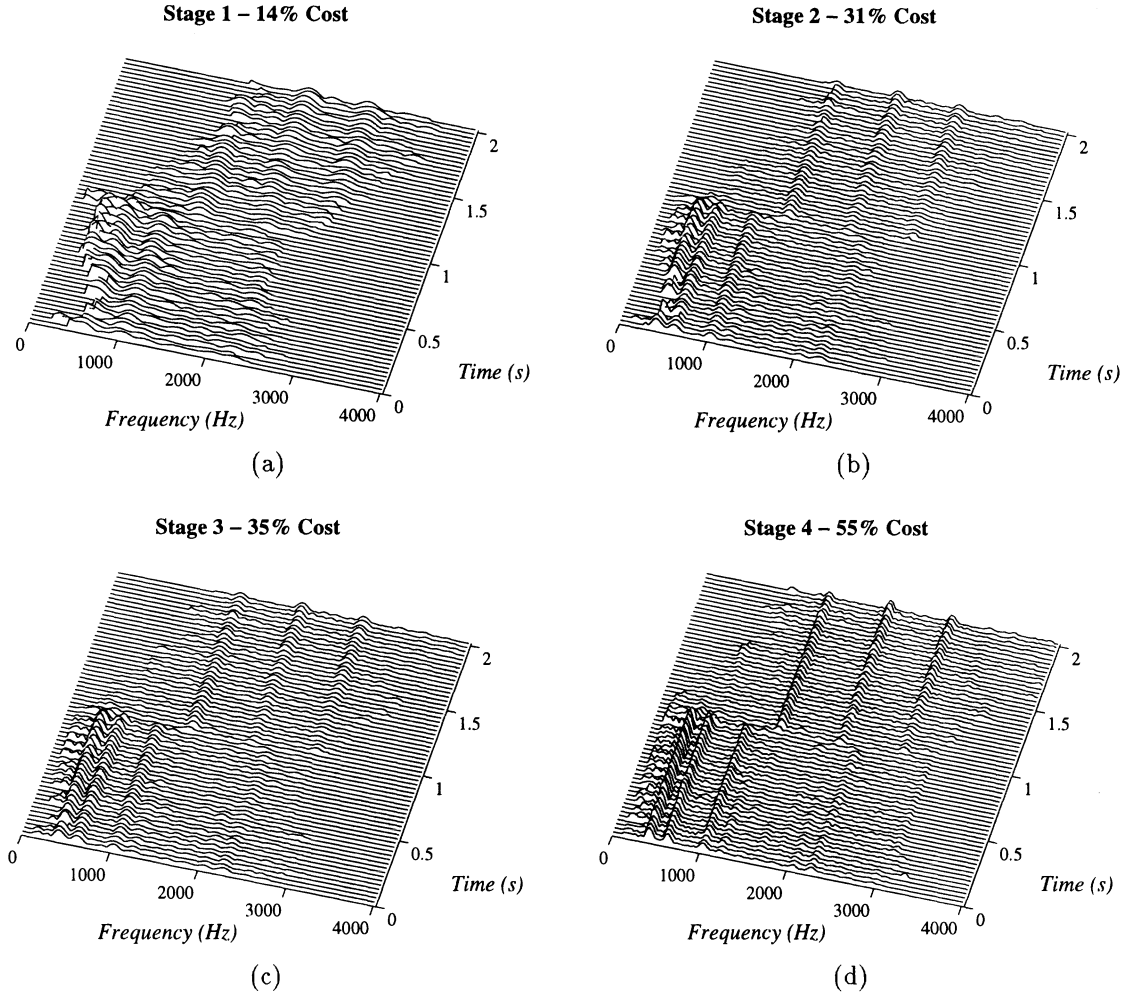

Figure 3: Incremental refinement of STFT approximations based on a radix-4 signal representation. STFT magnitude is shown and net computational cost is given as a percentage of the number of arithmetic operations required for FFT-based exact STFT analysis. (a) Result of stage 1: initial approximation. (b) Result of stage 2: refinement in coverage and resolution. (c) Result of stage 3: refinement in coverage. (d) Result of stage 4: refinement in coverage and resolution.

7 CONCLUSION

We have presented a new class of algorithms for computing successive approximations to the DFT. These algorithms utilize mixed-radix signal representations as opposed to the binary signal representation of our previously reported algorithms for incremenetal DFT refinement. We have shown that the mixed-radix signal representations leads to greater computational efficiency. Techniques for implementing these algorithms using general purpose computing hardware were also described. The work was motivated by the growing need for approximation algorithms with incremental refinement properties for the development of real-time signal processors that perform demanding tasks in complex and dynamically changing environments.

8 ACKNOWLEDGMENTS

This work was sponsored in part by the Department of the Navy, Office of the Chief of Naval Research, contract number N00014-93-1-0686 as part of the Advanced Research Projects Agency's RASSP program.

9 REFERENCES

[1] J. A. Stankovic. Real-time computing systems: The next generation. In J. A. Stankovic and K. Ramamritham, editors, *Hard Real-Time Systems*, pages 14–38. IEEE Computer Society Press, Washington D. C., 1988.

[2] J. W. S. Liu, W. K. Shih, K. J. Lin, R. Bettati, and J. Y. Chung. Imprecise computations. *Proc. IEEE*, 82(1):83–93, January 1994.

[3] A. Garvey and V. Lesser. A survey of research in deliberative real-time artificial intelligence. *Real-Time Systems*, 6(3):317–347, May 1994.

[4] J. M. Winograd and S. H. Nawab. Incremental refinement of DFT and STFT approximations. *IEEE Signal Processing Letters*, 2(2):25–28, February 1995.

[5] I. Koren. *Computer Arithmetic Algorithms.* Prentice-Hall, Englewood Cliffs, NJ, 1993.

[6] D. E. Knuth. *The Art of Computer Programming, Volume 2: Seminumerical Algorithms.* Addison-Wesley, Reading, MA, second edition, 1981.

[7] S. H. Nawab and E. Dorken. Efficient STFT approximation using a quantization and differencing method. In *Proc. IEEE ICASSP '93*, Minneapolis, April 1993.

[8] S. H. Nawab and E. Dorken. A framework for quality versus efficiency tradeoffs in STFT analysis. To appear in *IEEE Trans. Signal Processing*, April 1995.

[9] S. H. Nawab and J. M. Winograd. Approximate signal processing using incremental refinement and deadline-based algorithms. In *Proc. IEEE ICASSP '95*, Detroit, May 1995.

[10] E. Dorken and S. H. Nawab. Frame–adaptive techniques for quality versus efficiency tradeoffs in STFT analysis. In *Proc. IEEE ICASSP '94*, Adelaide, April 1994.

[11] S. H. Nawab and T. Quatieri. Short-time Fourier transform. In J. S. Lim and A. V. Oppenheim, editors, *Advanced Topics in Signal Processing.* Prentice-Hall, Englewood Cliffs, NJ, 1988.