Number Theoretic Transfoms in Audio Processing

S. Gudvangen, Høgskulen i Buskerud, AUI, EMR, Kongsberg, Norway
Email: Sigmund.Gudvangen@hibu.no.

Abstract— **This paper is concerned with application of Number Theoretic Transforms (NTTs) to audio processing. The problem of dynamic range is of particular interest for this application and is therefore treated in some detail.**

Keywords— **Audio processing, NTTs, trandforms.**

I. Introduction

Number theoretic transforms are Discrete Fourier Transforms (DFTs) defined over finite rings or fields and can, for certain applications, be a viable alternatives to the DFT defined over the (infinite) complex field $\mathbb{C}$. Introduction to finite rings and fields can be found in e.g. [1], here we shall have to be content with a few brief remarks.

Roughly speaking, a finite ring is an algebraic structure that supports addition and multiplication, while a finite field in addition also supports division. Note that subtraction is equivalent to addition with an additive inverse element and that division is equivalent to multiplication with a multiplicative inverse element. When all elements in a finite ring have inverse elements the ring becomes a finite field, also called a Galois Field (GF). Finite fields and rings are closed; that is, given a ring, or field, R and two elements $a, b \in R$, then $a \diamond b \in R$, where the operation $\diamond$ is either addition or multiplication. The fact that finite fields and rings are closed allows exact arithmetic, i.e. arithmetic without round-off errors, to be carried out. In the sequel a finite ring with M elements will be denoted R_M. All arithmetic in such a finite structure must be carried out modulo M, which will be indicated by the notation $\langle \cdot \rangle_M$.

A generic NTT is defined as

$$X(k) = \langle \sum_{n=0}^{N-1} x(n)\alpha^{kn} \rangle_M, \qquad (1)$$

$$k = 0, 1, \ldots, N-1,$$

where N is the transform length, M is the modulus and α is a root of unity of order N in the ring R_M. Note the close relationships with the DFT; the root of unity α plays the same role as $e^{-j(2\pi/N)}$(a root of unity of order N in $\mathbb{C}$) in the DFT. Unlike the DFT defined over $\mathbb{C}$, however, the NTT-domain sequence $X(k)$ has no amplitude or phase associated with it.

The inverse NTT (INNT) is likewise defined as

$$x(n) = \langle N^{-1} \sum_{k=0}^{N-1} X(k)\alpha^{-nk} \rangle_M, \qquad (2)$$

$$n = 0, 1, \ldots, N-1.$$

NTTs share several properties with the DFT defined over $\mathbb{C}$. In particular, they possess the cyclic convolution property (CCP):

$$y = T^{-1}(T(h(n) \cdot T(x(n))), \qquad (3)$$

where $x(n)$ and $h(n)$ are the two sequences to be convolved, $\cdot$ denote element by element multiplication and T and T^{-1} are the forward and inverse transforms, respectively.

The modulus M can be a prime $M = p$, a power of a prime $M = p^m$ or a composite number $M = \prod_{i=1}^{K} M_i = \prod_{i=1}^{K} p_i^{m_i}$, resulting in NTT pairs defined over a Galois Field (GF), an extension field $GF(p^m)$ or a finite ring R_M. Generally, a NTT of length N and with modulus $M = \prod_{i=1}^{K} p_i^{m_i}$ and possessing the Cyclic Convolution Property (CCP), exists in a finite ring R_M, provided that [2], [?], [4]:

1. a root of unity α of order N exists in R_M, with $\gcd(\alpha, M) = \gcd(N, M) = 1$.
2. $N \mid \gcd(p_i - 1, p_j - 1)\ \forall\, i, j \in [1, K],\ i \neq j$.
3. $\gcd((\alpha^i - 1), M) = 1\ \ \forall\, i \in [1, N-1]$.

The simplest case is when the modulus M is a prime p, as the finite ring R_M then becomes a Galois field $GF(p)$, so condition 2 above simplifies to $N|(p-1)$.

The incentive for using NTTs for audio processing steam from the promise of error-free computation, as well as the potential for reduced computational complexity. For the latter to be achieved, it is required that: i) the transform length is highly composite, so that fast divide and conquer FFT-type algorithms can be employed, ii) that a simple root of unity α is used, so that multiplications with powers of α may be carried out with barrel-shifters, which are much less area-demanding than general multipliers. The element by element multiplications in the NTT-domain still calls for general multiplication though.

Booth those properties can be achieved by choosing a Fermat number, i.e. $M = F_t = 2^{2^t} + 1$, as modulus, which results in a so-called Fermat Number Transform (FNT). FNTs have for this reason been among the most frequently employed NTTs and numerous implementations have been reported [5], [6], [7], [8], [9], [10]. Table 1 lists the possible combinations of N, M and α for M equal to F_4, F_5 and F_6 and root of unity $\alpha = 2$ and $\alpha = \sqrt{2}$. From a complexity point of view $\alpha = 2$ is the most attractive, as it allows multiplications with powers of α may be carried out with barrel-shifters. The latter case, i.e. $\alpha = \sqrt{2}$ simply means the element $\alpha \in R_{F_t}$, such that $\alpha^2 = 2$, which turns out to be equal to $2^{2^{t-2}}(2^{2^{t-1}} - 1) = 2^{3 \cdot 2^{t-2}} - 2^{2^{t-2}}$. Multiplication with $\alpha = \sqrt{2}$ can therefore be carried out by two shifts and one addition. Addition mod F_t is only slightly more expensive than normal 2's-complement addition [11]. Table 1 shows that for a given modulus M only two transforms lengths are available when $\alpha \in [2, \sqrt{2}]$. Although longer transforms do exist, they do not have a simple α, and hence multiplication with powers of α calls for general multiplications. Note that when the filter coefficient vector $h(n)$ is fixed, $H(k) = NTT(h(n))$ can be precomputed and pre-multiplied with N^{-1}. Explicit multiplication with N^{-1} in the INTT can thus be avoided.

$\mathrm{M} = 2^{2^t}+1$	t	B	B_{eff}	N	
				$\alpha = 2$	$\alpha = \sqrt{2}$
F_4	4	17	16	16	32
F_5	5	33	32	32	64
F_6	6	65	64	64	128

Table 1:Some combinations of α, N and M for FNTs.

II. Dynamic Range

The input sequence $\widehat{x}(n)$ will often consist of bipolar elements belonging to a sub-set S of the ring $\mathbb{Z}$ of integers, i.e. $S \subset \mathbb{Z}$. It is therefore generally necessary to map the elements $\widehat{x} \in S$ into elements $x \in R_M$, i.e. $\chi : \widehat{x} \rightarrow x$:

$$x = \begin{cases} \widehat{x} \text{ for } \widehat{x} \in [0, M/2) \\ \widehat{x} + M \quad \text{otherwise.} \end{cases} \tag{4}$$

At the end of the computation the inverse mapping $\chi^{-1} : y \rightarrow \widehat{y}$ must be carried out. When the convolution is carried out in another ring, or field R_M, this ring/field is often referred to as a *surrogate* ring/field.

It is imperative that the dynamic ranges of the input and output sequences are bounded, that is $|x(n)|_{\max}, |h(n)|_{\max}$ and $|y(n)|_{\max} \leqslant M$. When both input sequences are unknown, which generally is the case for e.g. correlation, the modulus M must be large enough to ensure that

$$|y(n)|_{\max} \leqslant N|x(n)|_{\max}|h(n)|_{\max}. \tag{5}$$

In cases where one of the input sequences is knows, such as for example when it consists of a fixed filter coefficient vector, this bound is too pessimistic and should be replaced with

$$|y(n| \leqslant |x(n|_{\max} \sum_{k=0}^{L_h-1} |h(k)|, \tag{6}$$

where L_h is the length of the filter coefficient vector. Since the amplitude of the output sequence $|y(n)|$, must be limited to the interval $[0, M)$, the corresponding effective wordlengths B_{eff} become

$$B_{eff} = \log_2 N + \log_2 |h(n)|_{\max} + \log_2 |x(n)|_{\max}. \tag{7}$$

and

$$B_{eff} = \log_2 \left(\sum_{k=0}^{L-1} |h(k)| \right) + \log_2 |x(n)|_{\max}, \tag{8}$$

respectively. Note from (4) that for bipolar sequences $|x(n)|_{\max} = 2|\widehat{x}(n)|_{\max}$; likewise for $h(n)$ and $y(n)$.

Failure to satisfy these bounds will result in $y(n)$ wrapping around and rendering the result meaningless. The minimum size of the modulus M therefore becomes proportional to the dynamic range of the two sequences to be convolved, as well as to the length L_h of the filter coefficient vector.

For high-quality audio the wordlengths of the input sequence $x(n)$ will lie between 16 and 24 bits. As seen from (6), $\log_2(\sum_{k=0}^{L_h-1} |h(k)|)$ bits must be added, to account for the dynamic range increase during the convolution. It should be appreciated, however, that relation (6) might be very different from relation (5). A typical example occurs when $h(n)$ consists of a long room impulse response sequence, in which case the tail of the impulse response consists of very small values.

III. Segmented Convolution

High-quality audio processing calls for a large dynamic range, which means that a large modulus M must be employed. In order to achieve a low computational complexity, however, a NTT with a simple root of unity α and a highly composite length N must be used. Although the structure of finite fields and rings allows a large number of NTTs to be defined, only a handful come close to satisfying all these requirements. Fermat Number Transforms (FNTs) [?], i.e. NTTs with M a Fermat number, are among the more interesting candidates. However, the large dynamic range required for high-quality audio processing means that the only Fermat moduli of interest (for single-modulus NTTs) are F_5 and F_6, with effective wordlength of 32 and 64 bits, respectively. As seen from Table 1, the corresponding transform lengths with simple roots of unity, i.e. $\alpha = 2$ or $\alpha = \sqrt{2}$, are limited to 32 and 64 for $M = F_5$ and 64 and 128 for $M = F_6$, respectively.

However, many applications in audio, such as e.g. convolution with room impulse responses, result in very long filter coefficient vectors; much longer than the available transform lengths. In order to be able to employ NTTs with relatively short lengths, therefore, the filter coefficient vector $h(n)$, of length L_h, must be partitioned into $Q = \lceil L_h/K \rceil$ short blocks, each of length K, where $K < N$ (the transform length), thus $h(n) = h_0(n)|h_1(n)| \ldots |h_{Q-1}(n)$, where $|$ denotes concatenation. Each segment $h_j(n)$, of length K, is then padded with $N-$ K zeros and convolved separately with a delayed version of the input sequence $x(n)$, thus $h_{jK}(n) * x_{jK}(n)$, where the sub-scripts $_{jK}$ on $h(n)$ and $x(n)$ denotes the delay relative to the first input sample in the current block. The output sequence is then obtained as

$$y_{jK}(n) = \sum_{j=0}^{Q-1} h_{jK}(n) * x_{jK}(n). \tag{9}$$

Although the Q short convolutions can now be carried out by any efficient convolution algorithm, here we shall only be concerned with the use of NTTs, according to (3).

As is well known, when the FFT is used for fast convolution, the arithmetic complexity decreases as the length of the transform increases. However, when NTTs are used the transform length N is dictated by the choice of modulus M and root of unity α and will usually be shorter than if FFTs were employed. This apparent loss of efficiency, however, is recuperated by the absence of general multiplications in the transforms (provided a simple root of unity is employed). The minimum transform length for transform domain processing to be more effective than direct form implementations will therefore be less than when FFTs (implementing the DFT defined over $\mathbb{C}$) are used.

For real-time applications such partitioning will usually be necessary in any case, in order to reduce the input-output delay. A scheme proposed in [12] employs transforms (FFTs) of varying lengths, in order to reduce the arithmetic complexity as much as possible. The use of remainder arithmetic and unusual wordlengths (such as 33 and 65 bits), means that NTTs are only attractive when implemented as an Application Specific Integrated Circuits (ASICs), or Field Programmable Gate Arrays (FPGAs). Hence, it will usually be expedient to standardize on a single transform length, even though at least two different transform lengths might be available. In cases where the tail of the coefficient sequence $h(n)$ is very low in amplitude, the dynamic range requirements will decrease towards the end of $h(n)$. Dynamic range might then be traded for longer segments, by varying the blocl-length K.

Coefficient vector partitioning and processing of short blocks will directly lead to a significant reduction of the input-output delay. The delay can be further reduced all the way down to one sample-interval if the first block is processed separately in the form of a direct form FIR filter, as suggested in [12]. The running FIR filter for the first block can then be computed with fixed-point arithmetic, in which case round-off errors will be incurred.

If it is deemed important that also the first block be computet without errors, the direct form FIR filter may be implemented by means of the RNS [13]. The drawback of that approach is that if arithmetic units different from those used in the NTTs must be incorporated in an ASIC, those units might be under-utilised. If, on the other hand, the first section is convolved by means of so-called sliding NTTs [14] this expense can be avoided, since in this case no additional arithmetic units are required. The price to be paid for eliminating the delay of the first block is in any case increased computational complexity.

Even though the short convolutions of (9) are carried out in a surrogate ring, or field, R_M the additions in (9) may be computed either in R_M or in the set $S \subset \mathbb{Z}$, with e.g. 2's complement addition. Since the dynamic range of the output sequence $y(n)$ eventually has to be reduced back to that of the input sequence $x(n)$, performing exact addition in a finite field/ring does not bring any benefits compared to the use of e.g. 2's complement addition with rounding. Although such a hybrid scheme relay of lossy addition for the recombination of the output sequence, the short block convolutions in (9) are error-free. As the round-off noise is largely caused be multiplication round-off errors [15], the bulk of the round-off errors are still avoided.

IV. Residue Number System Implementation

Single modulus NTTs with the required dynamic range for high quality audio processing are not abundant. One way to increase theflexibility with respect to dynamic range is to map the input sequences $x(n)$ and $h(n)$ into r smaller fields, $GF(p_1)$, $GF(p_2), \ldots, GF(p_r)$. Thus, given a sub-moduli set $\{p_1, p_2, \ldots, p_r\}$, such that $M = \prod_{j=1}^{r} p_j$, $\gcd(p_i, p_j) = 1 \ \forall \ i \neq j \in [1, r]$, the two input sequences $x(n)$ and $h(n)$ can be mapped into r separate *channels* by means of the Chinese Remainder Theorem (CRT).

The CRT [16] states that given a composite integer $M = \prod_{j=1}^{r} p_j$, where p_j are primes and $\gcd(p_i, p_j) = 1 \ \forall \ i \neq j$, the system of linear congruencies

$$x \equiv x_j \bmod p_j, \ j = 1, 2, \ldots, r \tag{10}$$

has the unique simultaneous solution mod M:

$$x = \left\langle \sum_{j=1}^{r} x_j (M p_j^{-1})^{\varphi(p_j)} \right\rangle_M , \tag{11}$$

where $\varphi(x)$ is Euler's totient function [16], i.e. the number of positive integers not exceeding x that are relative prime to x.

When the CRT is used to map the two input sequences $x(n)$ and $h(n)$ into r independent channels the scheme is usually referred to as the Residue Number System (RNS). We now proceed as follows [17]:

1. Input mapping: $x_j(n) = x(n) \bmod p_j$ and $h_j(n) = h(n) \bmod p_j$.
2. Sub-channel convolution:

$$X_j(k) = \left\langle \sum_{n=0}^{N_j-1} x_j(n) \alpha_j^{nk} \right\rangle_{M_j} , \ j = 1, 2, \ldots, r, \tag{12}$$

$$H_j(k) = \left\langle \sum_{n=0}^{N_j-1} h_j(n) \alpha_j^{nk} \right\rangle_{Mj} , \ j = 1, 2, \ldots, r. \tag{13}$$

$$Y_j = \langle X_j(k) H_j(k) \rangle_{M_j} , \ k = 0, \ldots, \ N_j - 1, \ \ j = 1, \ldots, r, \tag{14}$$

$$y_j(k) = \left\langle N_j^{-1} \sum_{n=0}^{N_j-1} Y_j(k) \alpha_j^{-nk} \right\rangle_{M_j} , \ j = 1, 2, \ldots, r, \tag{15}$$

3 Output mapping by the CRT:

$$y(n) = \left\langle \sum_{j=1}^{r} y_j(n) (M p_j^{-1})^{\varphi(p_j)} \right\rangle_M , \ n = 0, 1, \ldots, N_E, \tag{16}$$

where N_E is the effective block-length (see below).

Since the transform lengths N_j will generally be different for each channel, condition 2 in the introduction seems to require that $N|\gcd(p_i-1, p_j-1)$. However, there is really no need to insist on a NTT of length N defined over a composite ring R_M. It suffices that the r NTTs defined over $GF(p_j)$, $j = 1, 2, \ldots, r$, exists in their respective fields. [17]. The input map then merely dissects the input sequence $x(n)$ into sub-sequences $x_j(n), j = 1, 2, \ldots, r$, and the output map (12) in turn reassembles the sequence $y(n)$ from the r independent channel-sequences $y_j(n)$, $j = 1, 2, \ldots, r$. The effective block-length N_E is therefore given by $\min(N_j)$, $j = 1, 2, \ldots, r$. As a consequence, all channel blocks, apart from the block in the channel with the smallest N_j, must be zero-padded. The individual transform lengths should therefore be as similar as possible, in order to maximize efficiency.

One of the advantages of this scheme is that the dynamic range may be tailored to a particular requirement by adjusting the number of sub-moduli. Note that the convolutions in the r individual channels are completely independent of each other. The r convolutions may therefore be computed with any efficient NTT pair. There are nevertheless several factors to be considered when closing the sub-moduli p_j :
1. if short carry-propagation paths are deemed important, the individual sub-moduli should be relatively small
2. they should preferably not be too different, as the clocking rate will be limited by the slowest channel
3. efficient NTTs should exist in the r fields $GF(p_j), .j = 1, 2, \ldots, r$.

Requirement 3 above means that *preferably* all r NTTs should have booth simple roots of unity α_j and highly composite lengths N_j. However, this is not always easy to satisfy. In practise, therefore, a compromise might have to be struck. Since only NTTs defines over finite fields are used, the largest Fermat number moduli that can be used is F_4 (larger Fermat numbers are composite). Choosing $\alpha = \sqrt{2}$, results in a FNT of length $N = 32$, which is the most promising candidate in such a multi-moduli scheme. For the $r-1$ remaining channels other suitable NTTs must be found. Promising candidates with lengths a power of two have been listed in e.g. [18], [19]. It is generally not possible to find simple (i.e. powers of 2) roots of unity. However, other methods are available for avoiding general multiplications. On of these methods consists of various combinations of look-up tables [17], [18], [19]. This method is only suited for small moduli, though, as the tables tend to consume too much area for large moduli. An alternative method is provided by performing a basis-conversion [20]. That is, the arithmetic is carried out in a number representation which admit simple multiplication by powers of α.

V. Conclusion

The question of whether or not NTTs represent a viable alternative for audio processing cannot be answered with a simple yes or no. It all depends on the type of processing to be performed, the number of units one wants to produce, etc. The need for residue arithmetic and the unusual wordlengths encountered means that NTTs are not competitive in software implementations. For ASIC implementations, however, NTTs deserve serious consideration, as the potential for reduced area and absence of, or reduced, round-off noise might for certain applications be turned into commercial advantages.

References

[1] I. Herstein, *Topics in algebra.* John Wiley & Sons, 1975. 2nd. ed.

[2] J. Pollard, "The fast Fourier transform in a finite field," *Math. of Comp.*, vol. 25, pp. 365–374, April 1971.

[3] R. C. Agarwal and C. Burrus, "Fast convolution using Fermat number transforms with applications to digital filtering," *IEEE Trans. on Acoustics, Speech, and Signal Processing*, vol. ASSP-22, pp. 87–97, April 1974.

[4] I. Reed, "The use of finite fields and rings to compute convolutions," Tech. Rep. 1975-50, Mass. Int. of Techn., Lexington, USA, 6th June 1975.

[5] T. Troung, I. Reed, C.-S. Yeh, and H. Shao, "A parallel VLSI architecture for a digital filter of arbitrary length using a Fermat number transform," in *Proc. IEEE Int. Conf. on Circuits and Computers (ICCC'82)*, pp. 574–578, 1982.

[6] N. Yamane, Y. Morikawa, and H. Hamada, "A fast image filtering processor -FIFP-," in *Proc. Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP'86), Tokyo, Japan*, pp. 565–568, 1986.

[7] A. Bouridane, A. Pajayakrit, S. Dlay, and A. Holt, "CMOS VLSI circuits of pipeline sections for 32 and 64-point Fermat number transformers," *Integration*, vol. 8, pp. 51–64, 1989.

[8] P. Towers, A. Pajayakrit, and A. Holt, "Cascadable NMOS VLSI circuit for implementing a fast convolver using the Fermat number transform," *IEE Proc.*, vol. 134, Pt. G, pp. 57–66, April 1987.

[9] S. Gudvangen and A. Patel, "Rapid synthesis of a macro-pipelined CMOS ASIC for the Fermat number transform," in *Proc. Norwegian Signal Processing Symposium (NORSIG), Stavanger, Norway*, pp. 143–148, 1-2 Sept. 1995.

[10] A. Patel and S. Gudvangen, "Implementation of the Fermat number transform on FPGAs," in *Proc. 7th. Microcomputer School: VLSI and ASIC Design, Baligród-Bystre, Poland*, pp. 243–249, 9-13, Oct. 1995.

[11] L. Leibowitz, "A simplified binary arithmetic for the Fermat number transform," *IEEE Trans. on Acoustics, Speech, and Signal Processing*, vol. ASSP-24, pp. 356–359, Oct. 1976.

[12] W. Gardner, "Efficient convolution without input-output delay," *J. Audio Engineering Society*, vol. 43, pp. 127–136, March 1995.

[13] W. Jenkins and B. Leon, "The use of residue number systems in the design of finite impulse response digital filters," *IEEE Trans. on Circuits and Systems*, vol. CAS-24, pp. 191–201, April 1977.

[14] S. Gudvangen, "A class of sliding Fermat number transforms that admit a tradeoff between complexity and input-output delay," *IEEE Trans. on Signal Processings*, vol. 45, pp. 3094–3096, Dec. 1997.

[15] P. Chevillat, "Transform-domain digital filtering with number theoretic transforms and limited word lengths," *IEEE Trans. on Acousics, Speech, and Signal Processing*, vol. ASSP-26, pp. 284–290, Aug. 1978.

[16] D. Burton, *Elementary Number Theory.* Allyn and Bacon, 1980.

[17] W. Jenkins, "Composite number theoretic transforms for digital filtering," pp. 421–425, 9th Asilomar Conf. on Circuits, Systems and Computers, 1975.

[18] R. Saleh, *Algorithms and architectures using the number theoretic transform for digital signal processing.* PhD thesis, Electronic Laboratories, Univ. of Kent, Canterbury, Kent, UK, 1985.

[19] O. Hinten, "Implementation and application of generalised number theoretic transforms," in *Digest of Papers, IEE Colloquium on Signal Processing Applications of Finite Field Mathematics*, pp. 6/1–6/4, 1st. June 1989.

[20] M. Parker and M. Benaissa, "Bit-serial, VLSI architecture for the implementation of maximum-length number-theoretic transforms using mixed radix basis representations," in *Proc. Int. Conf. on Signal Processing (ICASSP)*, pp. I.341–I.344, 1993.