Coherent optical implementations of the fast Fourier transform and their comparison to the optical implementation of the quantum Fourier transform

Rupert C. D. Young, Philip M. Birch, Chris R. Chatwin

Department of Engineering and Design,
School of Engineering and Informatics
University of Sussex,
Brighton, United Kingdom, BN1 9QT
E-mail: r.c.d.young@sussex.ac.uk

ABSTRACT

Optical structures to implement the discrete Fourier transform (DFT) and fast Fourier transform (FFT) algorithms for discretely sampled data sets are considered. In particular, the decomposition of the FFT algorithm into the basic Butterfly operations is described, as this allows the algorithm to be fully implemented by the successive coherent addition and subtraction of two wavefronts (the subtraction being performed after one has been appropriately phase shifted), so facilitating a simple and robust hardware implementation based on waveguided hybrid devices as employed in coherent optical detection modules. Further, a comparison is made to the optical structures proposed for the optical implementation of the quantum Fourier transform and they are shown to be very similar.

Keywords: coherent optical Fourier transform, quantum Fourier transform, discrete optical Fourier transform, fast Fourier transform

1. INTRODUCTION

The manipulation of the complex amplitude a coherent wave front forms the basis of coherent optical processing techniques. Central to these methods has been the ability of a converging lens to form in its back focal plane the complex two-dimensional Fourier transform of an aperture function placed in its front focal plane[1]. In addition, the recording of the Fourier domain complex field via carrier wave holographic methods has led to the development of the two lens 4-focal length coherent optical correlator which has been extensively investigated now for several decades[2]. In this paper, by contrast, we concentrate on the one dimensional discrete Fourier transform (DFT) of a discretely sampled data sequence (which, in the most general case, can be complex). We also review the scenario in which the data is in a quantum superposition state, as the DFT forms a central component of one of the most important quantum computing algorithms, namely the Shor factorisation algorithm[3]. Firstly, we briefly review implementation of the DFT by vector matrix multiplication approaches. However, a major component of the paper is to highlight that a more elegant realisation of the DFT can be accomplished by employing the superposition of coherent wavefronts to naturally implement the Butterfly operation which forms the basic computation involved in computing the DFT via the fast Fourier transform (FFT) algorithm[4,5,8]. Based on this, a waveguiding structure can be realised employing an optical hybrid device developed for combining the signal and local oscillator waves in optical communication coherent detection systems[6]. Combining these hybrids together with waveguides to control the relative paths lengths between the elements, thus allows a coherent implementation of the FFT by, effectively, embedding the algorithm directly into an optical hardware structure. Further, since this operation has been shown to involve only coherent additions of the wavefronts,

together with pure phase rotations, this computation can be performed on a superposition state without its collapse since it involves only unitary operations. This means that the coherent optical implementation of the FFT can be compared directly to the quantum FFT which forms a central component of the Shor algorithm. Thus very similar hardware structures can be used for the implementation of both the coherent optical and quantum optical FFTs, although the data entry and readout will be very different, as described below.

2. OPTICAL IMPLEMENTATION OF THE DISCRETE FOURIER TRANSFORM

The most direct way in which to implement a Fourier transform with coherent light is to directly employ a converging lens but since this is so well known no further discussion of this will be made here[1]. Slightly more inconvenient is the implementation of the Fourier transformation of a 1-D signal since this must in addition be imaged along the orthogonal axis to the 1-D Fourier transformation to avoid diffraction effects. The data input is by means of some form of spatial light modulator (SLM) which may be analogue or discrete. For instance, an acousto-optic cell has been employed to input high frequency time domain data from radar systems for real-time spectral analysis[7]. Alternatively, a 1-D SLM could be employed for a similar purpose, most commonly based on a nematic or ferro-electric liquid crystal[1,7]. If this device is capable of full complex modulation of the signal, real-only or full complex data can be introduced into the optical processor by this means. Generally, the frequency domain detection array will be an intensity sensitive device such as a charge coupled array (CCD) and the power spectrum of the input data sequence will be measured, although the complex frequency components could be detected if mixed with a reference beam which is mutually coherent with the signal beam i.e. the output could be holographically recorded, although this would require interferometric stability of the system and will be more complicated to arrange.

Alternatively, the Fourier transform operation may be implemented discretely on the data input. In this case the operation performed becomes the direct calculation of the discrete Fourier transform relation:

$$X(k) = \sum_{n=0}^{N-1} x(n) e^{-j\frac{2\pi nk}{N}} \tag{1}$$

where an n-point DFT is calculated for k discrete complex frequency components. The exponential is conveniently abbreviated to $e^{-j\frac{2\pi nk}{N}} = W_N^{nk}$ and can be represented as nk steps around the unit circle in the z-domain divided into N equal angle sectors. Note that any physical dimension of the time or spatial variable n and (temporal or spatial) frequency variable k is not present in the equation and the computation can computed purely numerically without reference to any physical dimensions. Equation (1) involves N^2 complex multiply operations since each of the N frequency components k requires, in general, n complex multiplications to evaluate. This makes direct evaluation numerically intensive for large N and of course was the motivation for the development of the various fast algorithms which are discussed in the next section. However, the structure of the computation is a matrix-vector multiply which can be seen by writing it in the following form (where we consider a 4-point DFT as this length is short but just long enough to illustrate the symmetries present in the computation):

$$\begin{bmatrix} X(0) \\ X(1) \\ X(2) \\ X(3) \end{bmatrix} = \begin{bmatrix} W_4^0 & W_4^0 & W_4^0 & W_4^0 \\ W_4^0 & W_4^1 & W_4^2 & W_4^3 \\ W_4^0 & W_4^2 & W_4^4 & W_4^6 \\ W_4^0 & W_4^3 & W_4^6 & W_4^9 \end{bmatrix} \begin{bmatrix} x(0) \\ x(1) \\ x(2) \\ x(3) \end{bmatrix} \tag{2}$$

Since the W_N^{nk} repeat once the nk product exceeds 3, i.e. the vector has passed more than once around the unit circle, the entries reduce to:

$$\begin{bmatrix} X(0) \\ X(1) \\ X(2) \\ X(3) \end{bmatrix} = \begin{bmatrix} W_4^0 & W_4^0 & W_4^0 & W_4^0 \\ W_4^0 & W_4^1 & W_4^2 & W_4^3 \\ W_4^0 & W_4^2 & W_4^0 & W_4^2 \\ W_4^0 & W_4^3 & W_4^2 & W_4^1 \end{bmatrix} \begin{bmatrix} x(0) \\ x(1) \\ x(2) \\ x(3) \end{bmatrix} \tag{3}$$

given which can be abbreviated for any *n, k* by using a vector-matrix notation:

$$\underline{X}^T(k) = \mathbf{W}(k,n)\underline{x}^T(n) \tag{4}$$

For the case illustrated, the matrix entries are the complex numbers:

$$\begin{bmatrix} X(0) \\ X(1) \\ X(2) \\ X(3) \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -j & -1 & j \\ 1 & -1 & 1 & -1 \\ 1 & j & -1 & -j \end{bmatrix} \begin{bmatrix} x(0) \\ x(1) \\ x(2) \\ x(3) \end{bmatrix} \tag{5}$$

and so, for instance, $X(1)$ is calculated as:

$$X(1) = x(0) - jx(1) - x(2) + jx(3) \tag{6}$$

Of importance is that the entries of the matrix $\mathbf{W}(k,n)$ can be expressed in terms of their phase-only retardations as:

$$\begin{bmatrix} e^{j0} & e^{j0} & e^{j0} & e^{j0} \\ e^{j0} & e^{-j\frac{\pi}{2}} & e^{-j\pi} & e^{-j\frac{3\pi}{2}} \\ e^{j0} & e^{-j\pi} & e^{-j0} & e^{-j\pi} \\ e^{j0} & e^{-j\frac{3\pi}{2}} & e^{-j\pi} & e^{-j\frac{\pi}{2}} \end{bmatrix} \tag{7}$$

since W_N^{nk} moves clockwise around the unit circle in steps of $\dfrac{\pi}{2}$ (for a 4-point forward DFT). The entries are thus all of unit modulus and consist only of a varying phase term.

The matrix-vector multiplication can be optically computed in parallel by a matrix-vector multiplier as illustrated in Figure 1 below, originally proposed by Goodman[1]. Although usually designed with incoherent LED sources, a coherent source illuminating an input 1-D SLM, provided appropriate phase correction is ensured for the fan-out and fan-in optics, would allow direct computation of a DFT on the device. Of significance for such a coherent optical implementation of the matrix-vector multiplier for the calculation of the DFT is that the 2-D SLM used to record the W_N^{nk} entries need only be capable of phase only modulation (over 2π radians). Again, direct intensity detection at the output will only allow the power spectral terms, $|X(k)|^2$, to be detected but a full complex spectrum will be computed and available for measurement if required and an interferometric method for achieving this is implemented.
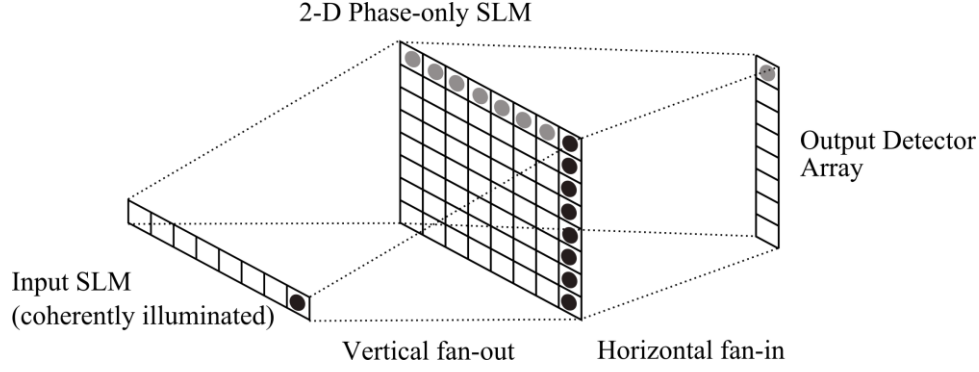
Figure 1. Coherent matrix-vector multiplier for calculation of the discrete Fourier transform

Importantly for later discussions in this paper, the phase only property of the DFT matrix, and hence the DFT operation, is a unitary i.e. the matrix has the property that $\mathbf{WW}^\dagger = \mathbf{I}$ where the dagger superscript indicates the conjugate transpose of $\mathbf{W}$. Physically this means that the physical realisation of the computational operation is reversible and non-dissipative[9]. Thus in the example of the DFT if the output detector was replaced by a mirror and the output reflected back through the matrix (changed to $\mathbf{W}^\dagger$ because the input is now horizontal and the output vertical), the input sequence would be reconstructed without loss of information. In quantum computing, the unitary transformation implies conservation of the overall probability of detection of the propagated wavefunction through the system, only the distribution of the probability of detection over the output detector being altered from that of the input, the overall probability of collapse remaining conserved. Importantly, the wavefunction must propagate through the system without collapse, a requirement fundamental to quantum computing hardware.

3. COHERENT OPTICAL IMPLEMENTATION OF THE DISCRETE FOURIER TRANSFORM VIA THE FAST FOURIER TRANSFORM

As emphasised in the previous section, direct evaluation of the DFT requires computations proportional to the square of the transform size N. Coherent optical computations can be made in parallel over a two-dimensional surface (the discrete values being recorded on a 2-D SLM) and so computations increasing as a second power law, such as the DFT, can be made efficiently.

However, symmetries in the structure of the DFT allow it to be calculated in a time proportional to $N\log_2 N$, as first described by the Cooley-Tukey fast Fourier transform algorithm[4]. The basic decimation in time FFT will be briefly reviewed here because its basic structure allows it to be implemented with coherent optical methods in a particularly simple and elegant manner as first realised by Siegman[8].

A brief summary of the principle of the FFT decomposition is summarised so the structure that provides the basis for a coherent optical implementation can be made clear. The basic operation of the decimation in time algorithm is to split the data sequence $x(n)$ into an even and odd sequence and to DFT each separately:

$$X(k) = \sum_{n=0}^{N-1} x(n) W_N^{nk} = \sum_{n=0}^{\frac{N}{2}-1} x(2n) W_N^{k.2n} + \sum_{n=0}^{\frac{N}{2}-1} x(2n+1) W_N^{k(2n+1)}$$

(8)

Noting the symmetry in W_N:

$$W_N^2 = \left(e^{-j\frac{2\pi}{N}} \right)^2 = e^{-j\frac{2\pi}{N/2}} = W_{N/2}$$

(9)

gives:

$$X(k) = \sum_{n=0}^{\frac{N}{2}-1} x_1(n) W_{N/2}^{kn} + W_N^k \sum_{n=0}^{\frac{N}{2}-1} x_2(n) W_{N/2}^{kn}$$

$$X(k) = X_1(k) + W_N^k X_2(k)$$

(10)

where $k = 0,1,\ldots.N/2-1$, that is k is only evaluated to the half-length of the sequence, and $X_1(k)$ and $X_1(k)$ are the DFTs of the half-length even and odd sequences $x_1(n)$ and $x_2(n)$, respectively.

For the values where $k > (N/2-1)$, and using the symmetry:

$$W_N^{k-\frac{N}{2}} = -W_N^k$$

(11)

we have:

$$X(k) = X_1(k-N/2) - W_N^k X_2(k-N/2)$$

(12)

which is evaluated for values $N/2 \le k \le (N-1)$.

Computation of $X_1(k)$ and $X_2(k)$ requires $(N/2)^2$ complex multiplications in each case and to form the sum of the two combined half sequences requires a further $N/2$ complex multiplications but, for large N, overall this approximates to $N^2/2$ and thus the number of complex multiplications required is approximately halved. Equations (10) and (12) indicate that computation of two transform points from each of the half-length sequences can be calculated together to form two points in the full length transform and since the rotation factor, W_N, is the same in both cases only one complex multiplication is required. This operation forms the basis of the FFT algorithm and is known as a Butterfly operation[5] due to its shape on a signal flow graph as illustrated in Figure 2.
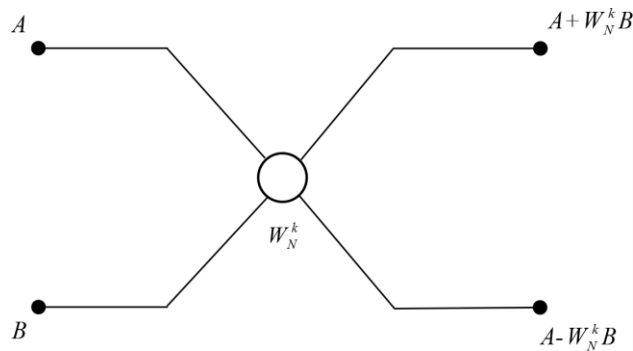

Figure 2. 2-point discrete Fourier transform signal flow graph

The algorithm proceeds by repeating the halving process which, provided N is a power of two, after $\log_2 N$ steps results $N/2$ 2-point DFTs remaining, each of which can be performed as a summation and difference of the two inputs since W^N becomes equal to π, i.e. -1, for a two point DFT. The overall structure is best visualised from the signal flow chart shown in Figure 3 for an 8-point FFT. Note that there are three stages in the computation i.e. $\log_2 8$ stages, each requiring four complex multiplications (one in each of the four Butterflies per stage). The successive division of the input sequence into even and odd sequences results in input values having to be re-ordered for input to the FFT as shown in Figure 3, this order being readily determined by reversing their bit addresses.
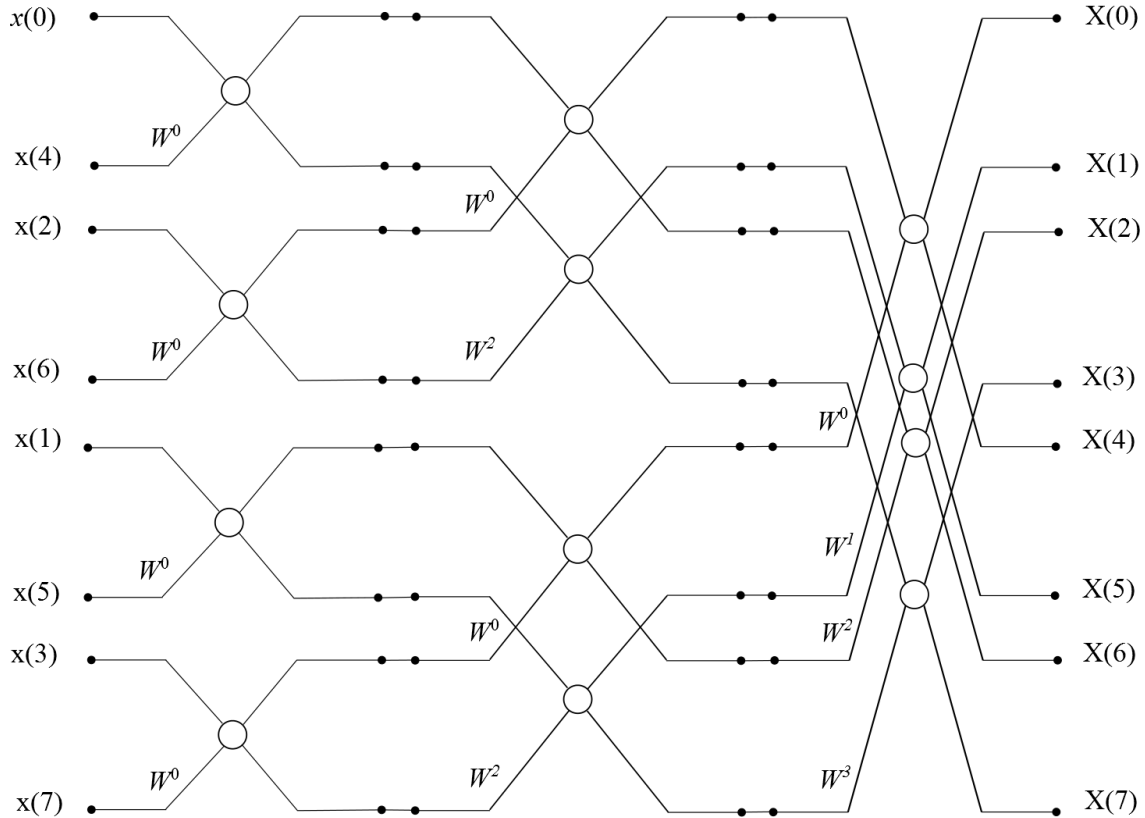

Figure 3. Signal flow graph for 8-point fast Fourier transform

As noted by Siegman[8], the breakdown of the DFT calculation into the Butterfly operations that the FFT algorithm creates results in only three simple operations being required at each Butterfly, namely: the addition of two complex numbers; the subtraction of two complex numbers; and a pure phase rotation of a complex number (performed before the subtraction). These operations can be readily performed by coherent optical methods and make an elegant mapping of the FFT algorithm into a coherent optical 'hardwired' structure possible. Siegman[8] suggested employing 3 dB fibre couplers at each Butterfly connected by carefully matched length fibre links but notes this may be susceptible to mechanical instability affecting the matched phase delays between fibres. An alternative would be to employ an etched waveguide structure in a suitable substrate. The Butterfly operation could then be realised by the waveguide structure shown in Figure 4. This device has been developed to act as a so-called hybrid device as employed in fibre optical communications coherent detection systems[6]. In this use, the weak signal and strong local oscillator beams are coherently added and subtracted at the two outputs of the device before being mixed onto two balanced photodetectors (to eliminate Relative Intensity Noise generated by the local oscillator). In addition, a precisely controlled phase delay can be applied to the lower beam path by means of an electro-optic modulator as shown in Figure 4. Its action can thus be written in matrix-vector form as[6]:

$$\begin{bmatrix} E_{o1} \\ E_{o2} \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 1 & e^{-j\phi} \end{bmatrix} \begin{bmatrix} E_{i1} \\ E_{i2} \end{bmatrix}$$

(13)

Thus the hybrid is capable of implementing the three coherent operations we require to realise an FFT Butterfly stage. In particular, a fine phase control will be required for large N FFTs e.g. a 1024 point FFT requiring $2\pi/1024$ radian increments of phase delay (also making accurate control of the path lengths connecting the Butterflies critical).
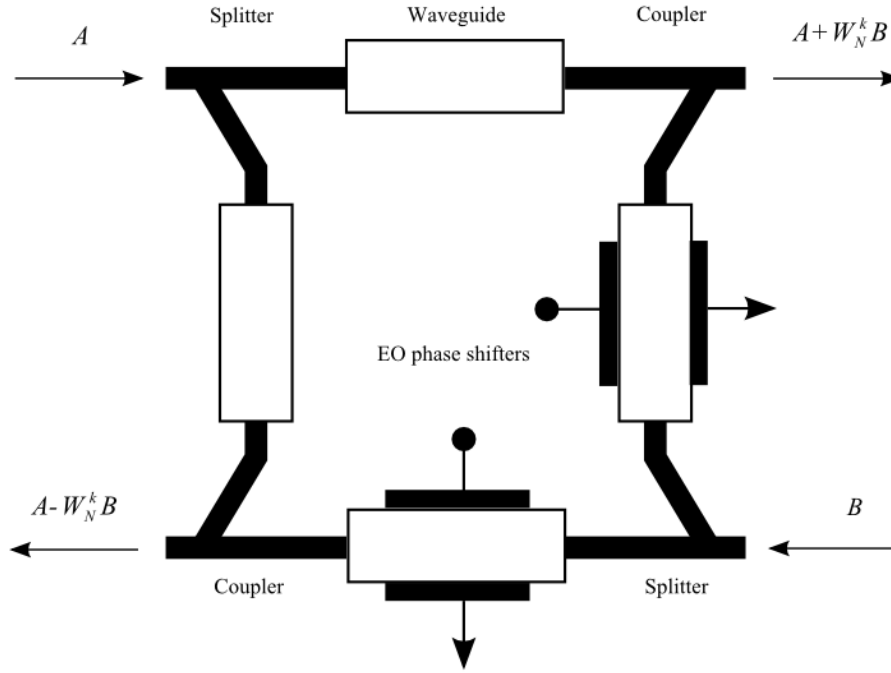

Figure 4. Hybrid device for coherent beam addition with controlled relative phase delay

4. COHERENT OPTICAL IMPLEMENTATION OF THE QUANTUM FOURIER TRANSFORM

The quantum Fourier transform has received considerable attention recently, mainly because it forms a central component of the Shor factorisation algorithm[3]. This algorithm is of central importance as its successful implementation would allow the prime factorisation of very large numbers. This is of central importance to the breaking of cryptographic codes which, due to the very large size of the numbers employed comprising 1024 bits or more, is very lengthy or even intractable by other computing means. Quantum algorithms offer the potential for an exponential speed up in processing time and so would make factorisation of such large numbers realistically possible[3].

The Shor algorithm factorises an integer $N = N_1 N_2$ by finding the period, r, of a function $f(k) = m^k$ mod N where $1<m<N$ and the greatest common divisor, gcd(m,N) = 1, where r must be even and $f(r/2) \neq N$-1. This gives factors N_1 = gcd($f(r/2)$+1, N) and N_2 = gcd($f(r/2)$-1, N) [3,14]. Given a randomly selected m such that gcd(m, N) = 1 the probability of finding a suitable r can be shown to be greater than 0.75 so a few selections of m only need be tested to find a factor of N.

However, employing conventional computing methods requires exponential time in N to determine the factors. The Shor algorithm proposes generating a series of superposition states of the modular functions followed by the Fourier decomposition of the state by a quantum Fourier transform (QFT) operation. Some of these states will comprise a

spatially periodic wavefunction, of period r, generated by a successful application of the modular arithmetic procedure for a good choice of m. The unitary transformation effected by the QFT will not result in the collapse of this wavefunction but a concentration of the probability distribution in the location of the detector cell corresponding the value of r. Repeated measurements on successive experiments will then result in a high probability that a detection at this location will be made, enabling the periodicity of the wavefunction r to be determined and hence N_1 and N_2 by determining the greatest common denominators of $f(r/2)+1$ and $f(r/2)-1$ with N which is possible to calculate efficiently using Euclid's algorithm[9].

Thus a QFT must be performed on the wavefunction generated by the modular arithmetic section of the algorithm. If we assume we have, for example, four entangled qubits (i.e. a one *and* a zero superposition at each bit location) described by the wavefunction (or state vector):

$$|\psi\rangle = \sum_{n=0}^{N-1} x_n |n\rangle = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} \tag{14}$$

that is, wavefunction can be considered a superposition of four qubits each weighted by a probability x_n representing the value of the input signal at that qubit location:

$$|\psi\rangle = x_{00}|00\rangle + x_{01}|01\rangle + x_{10}|10\rangle + x_{11}|11\rangle \tag{15}$$

The QFT of this state vector may then be computed:

$$|\Psi\rangle = \sum_{n=0}^{N-1} e^{-j\frac{2\pi nk}{N}} |\psi\rangle \tag{16}$$

which since this operation is unitary maintains the wavefunction superposition state but transforms it to the Fourier coefficients corresponding to input wavefunction, thus:

$$|\Psi\rangle = X_{00}|00\rangle + X_{01}|01\rangle + X_{10}|10\rangle + X_{11}|11\rangle \tag{17}$$

where the X_n are complex coefficients corresponding to the complex Fourier components at that qubit location in the output array. However, since they comprise the overall wavefunction they will not be directly accessible to measurement. Rather the probability of detection, by a single photodetection event, will be given by $|\Psi|^2$. If the input wavefunction is periodic, e.g. arises from the successful application of the *gcd* procedure, $|\Psi|^2$ will have a peak in its probability distribution at the output location corresponding to this. Thus repeated application of the QFT will yield more detection events at this location and hence allow determination of the periodicity, r, of the input as required to factor the initial input sequence, N. Thus the QTF is more powerful than the FFT in that it can process 2^N inputs in parallel with effectively the same complexity of hardware structure (and so is exponentially faster in computation), as described below. However, the FFT yields N complex frequency components at its output whereas the QFT produces a probability distribution only which collapses to a single photon detection event upon measurement. This, however, may be very useful and used to solve problems with exponential complexity if applied appropriately as, instance, in the Shor algorithm.

Since, as has been emphasised in Section 2, the optical implementation of the DFT by a coherent matrix-vector multiplier is a unitary operation, this hardware configuration could be utilised to realise a QFT provided the input sequence is in the required superposition state. Also, the waveguide structure described in Section 3 for the optical

implementation of the FFT could be employed in a similar way provided, again, that the inputs bits are in a quantum superposition state, thus in the optical case comprising entangled individual photons entering each input port to the FFT structure.

Barak and Ben-Aryeh[10] have described a QTF implementation with linear optics, following closely an Cooley-Tukey FFT based matrix decomposition to reduce the required number of optical components required. 50:50 beamsplitters and pure phase shifters are shown to be the only linear optical components required to implement the FFT structure, and an arrangement of these is shown for the proposed implementation of the 3 qubit QTF considered in detail in the paper.

The basic building block for the QFT is the Hadamard gate. This can be written:

$$\mathbf{H} = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

(18)

This is the basic operation of comprising the Hadamard transform and is optically implementable with a 50:50 beamsplitter together with an additional $\pi/2$ phase shift in one of the beams as shown in Figure 5 (this arrangement is also used in fibre optic communication coherent detection systems to realise a 180^0 hybrid).
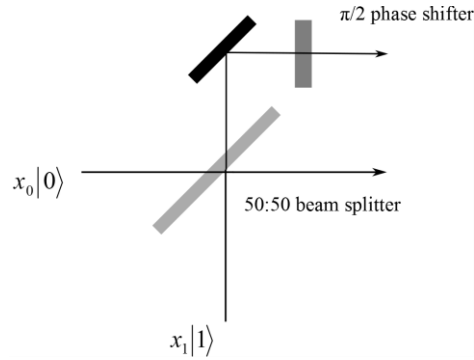

Figure 5. Optical implementation of Hadamard gate

This arrangement will act on a single qubit state to give:

$$\mathbf{H}\big[x_0|0\rangle + x_1|1\rangle\big] = \frac{1}{\sqrt{2}}(x_0 + x_1)|0\rangle + \frac{1}{\sqrt{2}}(x_0 - x_1)|1\rangle$$

(19)

Thus it can be seen that the Hadamard transform performs a 2-point QFT by implementing the basic FFT building block of the Butterfly operation i.e. the subtraction and addition of the two input signals, albeit in a superposition state. Larger count QTFs can thus be implemented by the same basic operation with the additional inclusion of a controlled phase shifter on the lower leg of the later stage Butterflies, as detailed when describing the structure of the FFT in Section 3. Thus the QFT can be optically implemented with the waveguide structure described in Section 3 for the optical implementation of the FFT (the difference being that the input values are in a superposition state rather than taking on explicit values).

The FFT decomposition of the QFT using the Hadamard gate as a basic building element together with phase rotation gates described by matrices of the form[12,13]:

$$R_n = \begin{bmatrix} 1 & 0 & 0 & . & 0 \\ 0 & 1 & 0 & . & 0 \\ 0 & 0 & 1 & . & 0 \\ . & . & . & . & . \\ 0 & 0 & 0 & . & e^{-j\frac{2\pi}{2^n}} \end{bmatrix} \tag{20}$$

This allows the QFT structure for a n-qubit input to be represented compactly as shown in Figure 6.
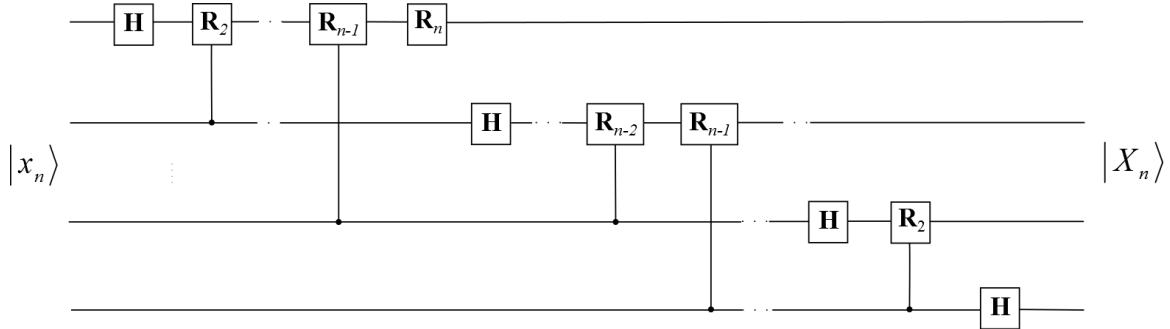

Figure 6. n-qubit QFT quantum gate array structure

The decomposition is thus an FFT structure acting on an binary n-qubit input to yield a complex-valued n-qubit output (in reverse bit order). An extension to a multi-valued QFT with input values weighted by a wavefunction, such as shown by Equation 15, has been proposed by Muthukrishnan and Stroud[11] making the similarity to the coherent optical FFT structure clear. An implementation problem arises in the phase resolution required by the final term in the R_n matrices for large n as this scales as 2^n. However, using methods originally proposed by Coppersmith[13] and with a later refinement by Fowler and Hollenberg[14] it may be possible to QFT a binary integer of several thousand bits with a phase control no better than $\pi/64$ which seems to be technically feasible. Thus the optical implementation of the QFT uses linear optical components which are essentially identical in form and arrangement to those that can be employed for the coherent optical implementation of a Fourier transform processor.

5. CONCLUSION

This paper has described various optical structures that are able to compute the discrete Fourier transform with coherent light, both by direct evaluation of the summation and also by implementation of a structure to implement the basic building block of the Cooley-Tukey fast Fourier transform algorithm, the Butterfly operation. A n-point discretely sampled coherent wavefront can thus be transformed into n complex valued Fourier co-efficients in no longer than the propagation time of the coherent wavefront through the optical system which, if constructed as a waveguide, may only be tens of millimeters long, implying less than nanosecond processing times. However, for some problems, such as the factorisation of large integers, even this very high processing speed is not enough as these problems have solution times which scale exponentially with n. This has led to interest over the last twenty years in the development of algorithms and hardware implementations suitable for quantum computing. The quantum Fourier transform has turned out to form a central component in some of the most important quantum algorithms, most notably the Shor algorithm for integer factorisation. Optical hardware components and configurations proposed to implement the quantum Fourier transform have been shown to be closely related to those proposed for the coherent optical implementation of the discrete Fourier transform using the matrix decompositions employed in the fast Fourier transform algorithm, the essential difference being the wavefront superposition state of the input and output registers in the quantum Fourier transform. This would,

however, make possible the exponential increase in processing power required to solve such problems as large integer factorisation.

REFERENCES

1. Goodman J.W., Introduction to Fourier Optics, McGraw-Hill, Second Edition, (1996).
2. Vander Lugt A., "Signal detection by complex spatial filtering", IEEE Trans. Inform. Theory, Vol. IT-10, pp. 139-145, (1964)
3. Shor P. W., "Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer", SIAM J. Comput., Vol. 26(5), pp. 1484-1509, (1997).
4. Cooley J. W., Tukey, J. W., "An algorithm for the machine computation of complex Fourier series", Maths. Comput. Vol. 19, pp. 297-301, (1965).
5. Brigham E. O., The Fast Fourier Transform, Prentice-Hall Inc, First Edition, (1974).
6. Liu, M-K., Principles and Applications of Optical Communications, Irwin, (1996).
7. Das, P. K., Optical Signal Processing Fundamnetals, Springer-Verlag, (1991).
8. Siegman A., E., "Fibre Fourier optics", Opt. Lett., Vol. 26(16), pp. 1215-1217, (2001).
9. Braunstein S. L., "Factoring numbers", www-users.cs.york.ac.uk/schmue/comp/comp.html, on-line February 2013.
10. Barak R., Ben-Aryeh Y., "Quantum fast Fourier transform and quantum computation by linear optics", J. Opt. Soc. Am. B, Vol. 24(2), pp. 231-240, (2007).
11. Muthukrishan A., Stroud C. R., "Quantum fast Fourier transform using multilevel atoms", Journal of Modern Optics, Vol. 49, pp. 2115-2127, (2002).
12. Ekert A., Jozsa R., "Shor's quantum algorithm for factorizing numbers", Rev. Mod. Phys., Vol. 68, pp. 733-753, (1996).
13. Coppersmith D., "An approximate Fourier transform useful in quantum factoring", IBM Research Report No. RC19642, T.J. Watson Research Center, Yorktown Heights, New York, (1994) (unpublished).
14. Fowler, A. G., Hollenberg L. C. L., "Scalability of Shor's algorithm with limited set of rotation gates", Physical Review A, Vol. 70, 032329-1:7, (2004).