Indian Institute of Technology Bombay
Dept of Electrical Engineering

Handout 11 EE 603 Digital Signal Processing and Applications
Lecture Notes 4 September 2, 2016

1 Discrete-Time Fourier Transform (DTFT)

We have seen some advantages of sampling in the last section. We showed that by choosing the sampling rate wisely, the samples will contain almost all the information about the original continuous time signal. It is very convenient to store and manipulate the samples in devices like computers.

Many a times the samples need to be processed before *playing it back* (reconstructed). A branch of signal processing known as Digital Signal Processing (DSP) deals entirely with this. This should be contrasted with the naive way of reconstructing the continuous time waveform first and then applying analog processing techniques. Let us enumerate the two approaches.

1. Do all the processing/computations on the samples $x[n], n \in \mathbb{Z}$ and perform a final reconstruction to obtain a signal $y(t)$.
2. Given the samples $x[n], n \in \mathbb{Z}$, first reconstruct the signal $x(t)$, and then do analog processing to generate $y(t)$.

An important question is whether both the approaches give the same $y(t)$ from a given set of samples. If the answer is affirmative, the first method will be a clear winner in terms of feasibility and convenience, since we have the freedom of using devices like computer, Digital Signal Processors (DSPs). The primary data processing application we have in mind is filtering, which is nothing but frequency shaping of the input signals. Linear filtering involves convolution by the impulse response of the filter. Equivalently we have to multiply the responses in the frequency domain. What is the frequency domain representation of a sampled signal.

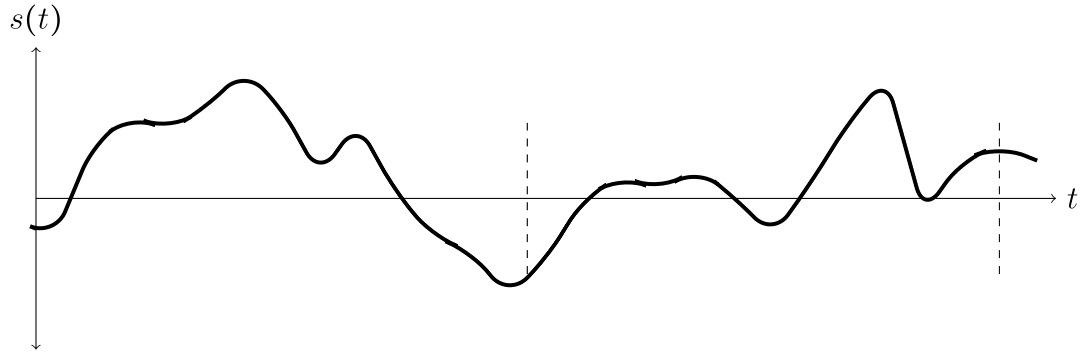
1.1 Do samples have frequency

Samples, after all, are a given set of values. How can they have a frequency associated with it? To find this, it is fruitful to look from the reverse angle. Given a set of samples, what could have been the original waveform from which these samples were taken. This will automatically identify the samples with some time interval T. For each value of T you may find a possibly different *parent* waveform. Furthermore, for a given T there can by multiple parent waveforms which generate the samples. All these connections imply that the frequency components that we attach to the samples have to relate to all possible parent waveforms. Thus it makes sense to fix T, as a real number such that

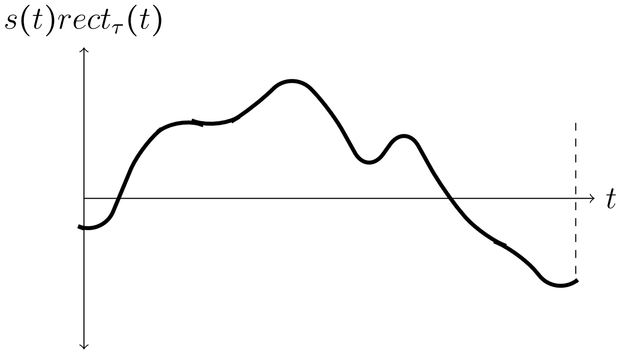
$$x[n] = x(nT)$$

where $x[n], n \in \mathbb{Z}$ are the samples and $x(t)$ the corresponding parent waveform, which has a continuous-valued index. Suppose we know a signal $x(t)$ whose samples are the given points $x[n], n \in \mathbb{Z}$. There is an intimate relationship between the Fourier Transform $X(f)$ of $x(t)$ and the frequency contents of $x[n]$. In fact our generalized Fourier Transform allows us to view both the continuous and discrete signal as part of the same tapestry. Here is how it is done, illustrated by an example.

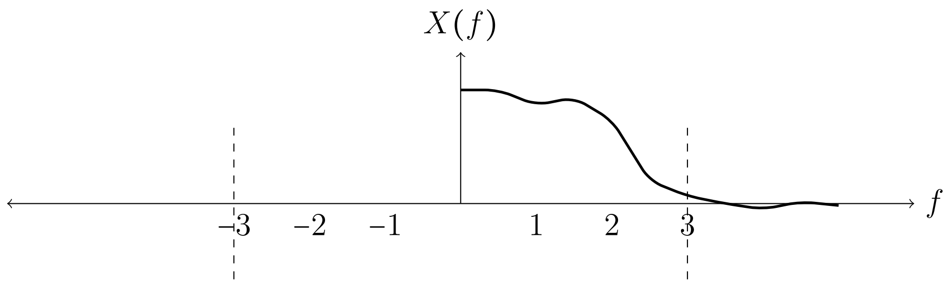
Consider the following signal which is a segment from the output of a microphone.



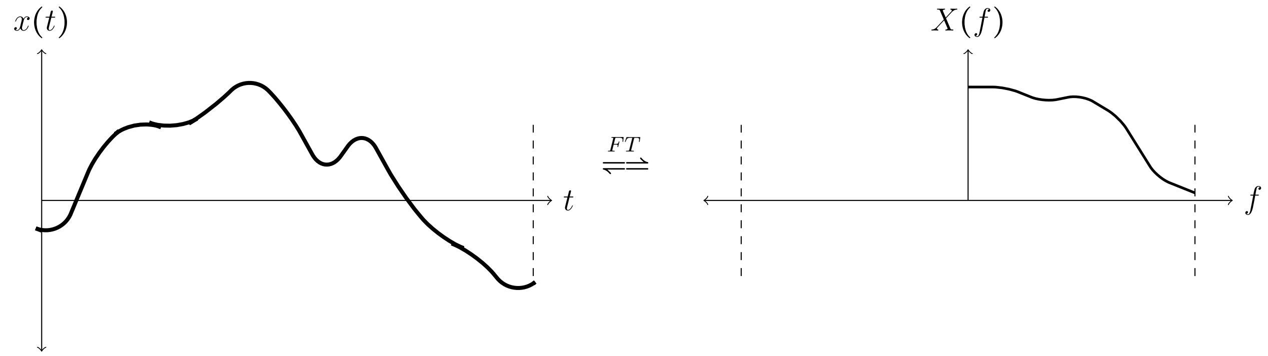
Many a times we cannot afford the luxury of waiting till the end of the signal and then computing the FT. So we wait for a reasonable delay, crop the signal (multiply by a window) and take its Fourier Transform. The dashed lines shows the points where the signal is windowed (segmented). Consider the first segment, call it $x(t)$.



The FT $X(f)$ of $x(t)$ is not bounded in time (why?). Assume that $X(f)$ is given by the following figure (This is not an exact computation, only a visual aid for pedagogical purposes). One more comment is in order here. In most plots of the FT, we will simply show the magnitude alone. The reader is expected to imagine some phase component $\exp(j\phi(f))$, where $\phi(f)$ is an odd function if the signal $x(t)$ is real, i.e. $\phi(f) = -\phi(-f)$. We will deal with the phase component in a more precise manner later, when we design practical filters.



It is very difficult to keep track of all the frequencies in $[0, \infty]$. A practical approach is to limit the components to the significant values. For example, we consider the values outside $[-f_m, f_m]$ to be insignificant for all computational purposes. This is not only reasonable, but also of great help in analytical considerations.



Once the frequency components are limited, we know that the signal can be loss-lessly stored by $2f_m$ samples per second, where f_m is the highest frequency content present. This is illustrated in the following figure, where sampling and the corresponding Fourier Transform are shown [1], where $T = \frac{1}{2f_m}$.

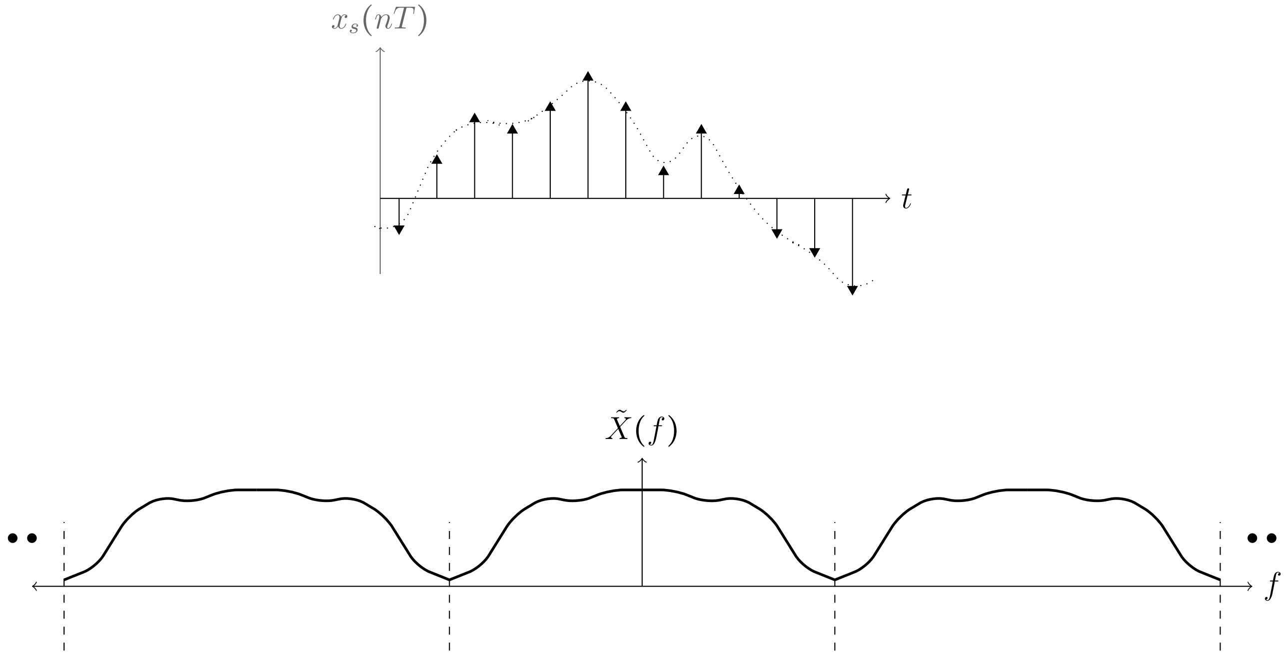


Figure 1: FT of the time samples

The FT of the samples is given by an appropriate repetition of $X(f)$ by the convolution-multiplication theorem. We can sample at any rate greater than $2f_m$ samples per second, In particular,

$$\begin{aligned}\tilde{X}(f) &= \int_{\mathbb{R}} \sum_n x(nT)\delta(t - nT)e^{-j2\pi ft}dt && (1)\\ &= \sum_n x(nT)\exp(-j2\pi fnT). && (2)\end{aligned}$$

We have already introduced the notation that $x(nT) = x[n]$. Using this

$$\tilde{X}(f) = \sum_n x[n]\exp(-j2\pi fnT) \tag{3}$$

[1] In the figures, we will use the symbol •• to denote that there is a repetition of the pattern in the direction of the symbol, see for example Figure 1.

Notice that the RHS of (3) is $2f_m$-periodic in f. This also tells that the samples of $x(t)$ are taken at integer multiples of $T = \frac{1}{2f_m}$. However, once the samples are taken, they are dead values, in the sense that they can be written on a piece of paper, or stored in memory arrays in diverse fashion. It thus makes sense to define a Fourier transform for our samples which is somewhat independent of the sampling interval, but only depends on the sequence of sample-values. This can be achieved by a suitable scaling. In particular,

$$\hat{X}(f) \coloneqq \tilde{X}(\frac{f}{T}) = \sum_n x[n] \exp(-j2\pi f n). \tag{4}$$

Note that $\hat{X}(f)$ has unit period, we call this the DTFT of $x[n]$. From our generalized Fourier Theory, the inverse of DTFT should correspond to the input samples, which are spaced at unit intervals. However, we have learned that for a periodic waveform, the generalized Fourier representation is obtained by computing the Fourier Series coefficients. Thus,

$$x[n] = \int_{-\frac{1}{2}}^{\frac{1}{2}} \tilde{X}(f) e^{+j2\pi f n} df. \tag{5}$$

Notice the slight difference from the original FS formula. Here we use $e^{+j2\pi f t}$, to be consistent with the formula for DTFT in (4).

DTFT and Inverse-DTFT

$$\hat{X}(f) = \sum_n x[n] \exp(-j2\pi f n) \tag{6}$$

$$x[n] = \int_{-\frac{1}{2}}^{+\frac{1}{2}} \hat{X}(f) \exp(+j2\pi f n) df \tag{7}$$

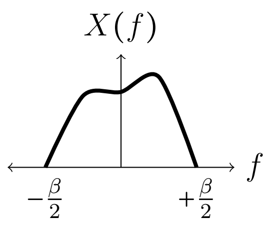
2 Digitizing Signals and Systems

Imagine a bandlimited waveform $x(t)$, with the frequency domain representation $X(f)$, i.e. Fourier Transform of $x(t)$. Since $x(t)$ is bandlimited, it is unlimited time. If $x(t)$ is real, $X(f)$ is complex symmetric around the origin. However, there is no reason that our signal should be real, and we will consider complex signals of the form,

$$x(t) = x_R(t) + j x_I(t).$$

For illustration, we will consider a $x(t)$ waveform with Fourier Transform $X(f)$ as depicted below.

$$x(t) \quad \overset{FT}{\rightleftharpoons}$$

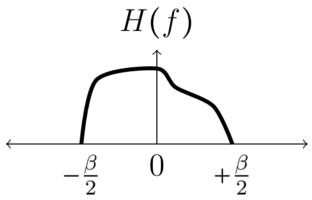


We will repeatedly use the above pictorial representation. We should keep couple of things in mind while using these pictures. First, the inverse Fourier transform (IFT) of such an $X(f)$ may correspond to a non-causal signal. Thus our descriptions are not for the absolute time, and we do not really worry about causality. Secondly, we do not emphasize the phase information in these pictures. This is partly due to the fact that we expect the phase response to be linear in the frequencies of interest, this will be covered later.

2.1 Frequency Response of a System

The frequency response of a LTI system can be determined by sending adequately spaced complex sinusoids and measuring the output attenuation, in terms of magnitude scalings and phase-rotations. From this we can find the response which is continuous over the frequency range of interest. Note that certain frequencies may propagate better, whereas some other frequencies may combine destructively at the output of the system. Let us consider the following system.

$h(t) \quad \overset{FT}{\rightleftharpoons}$



The signal at the output of the system is

$$y(t) = x(t) * h(t),$$

where $h(t)$ is the IFT of $H(f)$.Taking Fourier Transform,

$$Y(f) = X(f)H(f).$$

Exercise 1. *Argue that $y(t)$ may correspond to a complex waveform even when the input is real.*

Take $h(t)$ to be a time continuous impulse response function. This will imply that the system possibly superposes a continuum of delayed replicas of the transmitted signal, also known as echoes (try visualizing this). It turns out that a much more simpler visualization is possible. Notice that

$$Y(f) = X(f)H_c(f)\mathbb{1}_{\{-\frac{\beta}{2}\le f\le\frac{\beta}{2}\}},$$

for any $H_c(f)$ such that

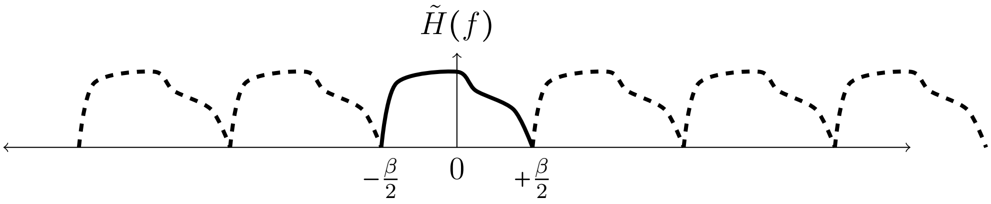
$$H_c(f) = H(f)\,,\ -\frac{\beta}{2} \le f \le \frac{\beta}{2}. \tag{8}$$

Thus, we can choose any $H_c(f)$ with the above property, and our output $Y(f)$ stays unchanged. On taking IFT, this implies

$$y(t) = x(t) * \tilde{h}(t),$$

where $\tilde{h}(t)$ is the IFT of $H_c(f)$. Let us now choose a $H_c(f)$ which satisfies (8), but has a convenient inverse. In particular, we will periodically repeat $H(f)$ over the entire frequency axis, i.e.

$$H_c(f) = \sum_{k\in Z} H(f - k\beta).$$



The periodic nature of $\tilde{H}(f)$ implies that $\tilde{h}(t)$ is of discrete in nature. Furthermore, since $\tilde{H}(f)$ is the periodic repetition of $H(f)$ in frequency, the time domain response $\tilde{h}(t)$ is nothing but a time-sampled version of $h(t)$. In other words,

$$\tilde{h}(t) = \sum_k \frac{1}{\beta} h_B(\frac{k}{\beta})\delta(t - \frac{k}{\beta}). \tag{9}$$

The scaling $\frac{1}{\beta}$ is due to the fact that a periodic impulse train in time will have a Fourier Transform of a scaled impulse train, with their periods in inverse relationship. Equation (9) assures that the effective channel can be visualized as a tapped delay line between the input and the output of the system. We will also call the value of the taps as the system impulse response and denote them by $h_k, k \in \mathbb{Z}$, clearly $h_k = \frac{1}{\beta} h(\frac{k}{\beta})$. With this,

$$y(t) = \sum_{k \in \mathbb{Z}} h_k \, x(t - \frac{k}{\beta}). \tag{10}$$

While this model is remarkably simple, we did not loose any generality in obtaining this. This simplicity should be compared with the physical reality where a continuum of delayed inputs are are superposed in an actual LTI system.

2.2 Discrete-time Signals and System

In the last section, we constructed an equivalent discrete-time system for a bandlimited LTI system, which gives us both conceptual and analytic simplicity. Can we convert the inputs and outputs also to an equivalent discrete-time system. The advantages are many fold, as several DSP books will advertise in the first chapter. How to do this is explained in the second chapters there, the premise of Sampling Theorems.

Exercise 2. *What is the Nyquist sampling rate to preserve all the information of $y(t)$ in (10)?*

Let us sample $y(t)$ at the rate of β samples per second.

$$y_B(\frac{m}{\beta}) = \sum_k h_k x_B(\frac{m}{\beta} - \frac{k}{\beta}).$$

Denoting $u(\frac{m}{\beta}) = u_m$, and dropping the subscripts,

$$y_m = \sum_k h_k x_{m-k},$$

or in our usual DSP notation,

$$y[m] = h[m] \star x[m].$$

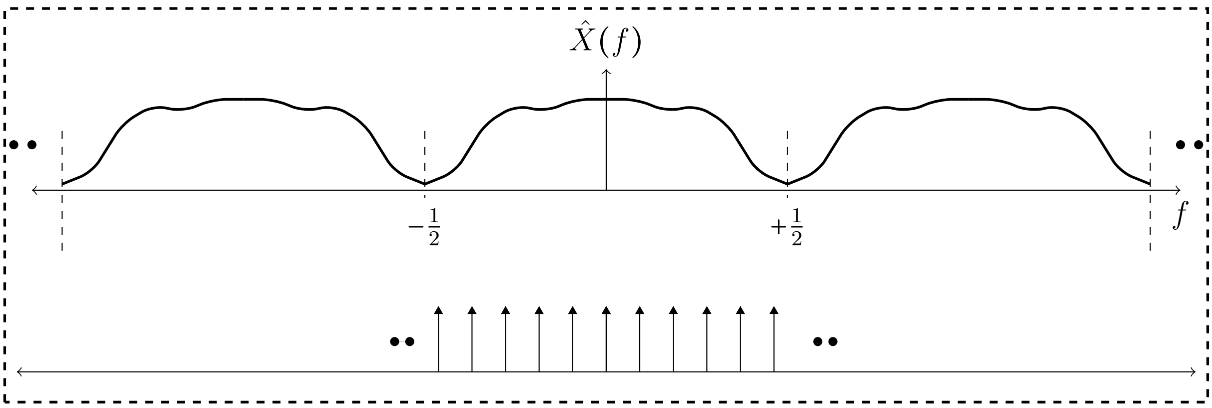
3 DFT: Sampling the DTFT

Reconstruction of a signal from its samples can be visualized as windowing in the frequency domain, recall the sampling theorem and its reconstruction formula. Most often we need to manicure the frequency contents before reconstruction. Consider for example the generation of surround sound and special effects. To generate an unexpected or surprising sensation, it is better to have a high frequency signal played from the back of the head for a very short duration. Low frequency rumblings accentuates the movie effects if played as surround sound. So we have to first multiply the spectrum of the composite sound (with the entire audible spectrum) by a suitably shaped frequency window before reconstructing an analog signal. This multiplication can also be done post reconstruction, but does not take advantage of the modern day digital technologies. Observe that multiplying by a frequency shaping function is equivalent to the convolution by a time domain filter.

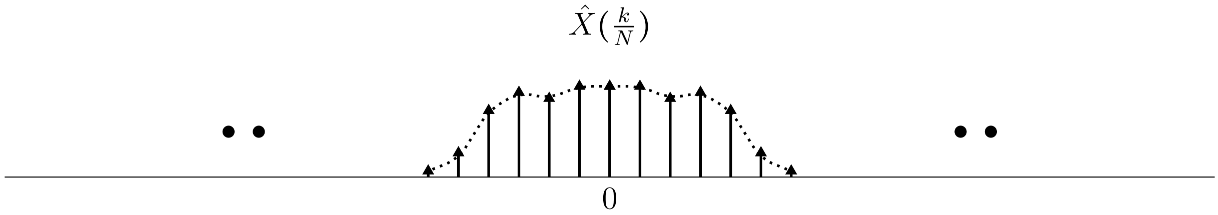
Most often we have a time domain signal/samples at our disposal and needs to do filtering (convolution) using an appropriate system. However, filter operations are best visualized in the frequency domain itself. But, can we physically perform these operations in the frequency domain, i.e., do filtering as multiplications in frequency instead of time-convolutions. The affirmative answer is popularly employed in a Digital Signal Processor (DSP, your mobile phone has one inside) or general purpose computer.

Recall that the DTFT yields a continuous and periodic frequency response. How will we manipulate a continuous time signal in a CPU/DSP. We were faced with a similar problem earlier, albeit in the time domain. Our solution was to sample, by the aid of the Sampling Theorem. Can we sample $\hat{X}(f)$ and replace filtering with a finite number of point-wise multiplications. This is almost true. If the time domain signal has bounded support, and we can use this property to sample the frequency domain representation without any loss.

Consider a discrete-time signal $x[n], 0 \le n \le N-1$. We can take it to be initially zero outside this interval. Since the DTFT spans from $f \in [-\frac{1}{2}, +\frac{1}{n}]$, with periodic repetitions thereafter, the period of $\hat{X}(f)$ is unit. Sampling $\hat{X}(f)$ at integer multiples of $\frac{1}{N}$ will equivalently cause $x[n]$ to be repeated after N samples in the time domain (convolution-multiplication theorem). However, $x[n]$ is defined for $0 \le n \le N-1$. Thus sampling $\hat{X}(f)$ at the rate of N samples per period will cause $x[n]$ to be repeated, albeit in an undistorted manner, every N successive samples of the repeated pattern will have all the information about $x[n]$. Observe the sampling the DTFT will correpond to a periodic discrete-time signal in the time-domain. This is crucial, every time we sample a DTFT, we have to deal with a periodic signal in the time domain. This is illustrated below for the example from the previous subsection.



Multiplying $\hat{X}(f)$ by an impulse train is shown in the figure above. The resulting samples in the frequency domain are shown below.



Since there are N samples in the interval $[-\frac{1}{2}, \frac{1}{2}]$, the spacing of the impulses in frequency is taken to be $\frac{1}{N}$. This corresponds to convolution with a time domain impulse train, where the impulses are apart by a distance of N. See the figure below for the time domain signal. Notice that our original $x[n]$ is present in each period.

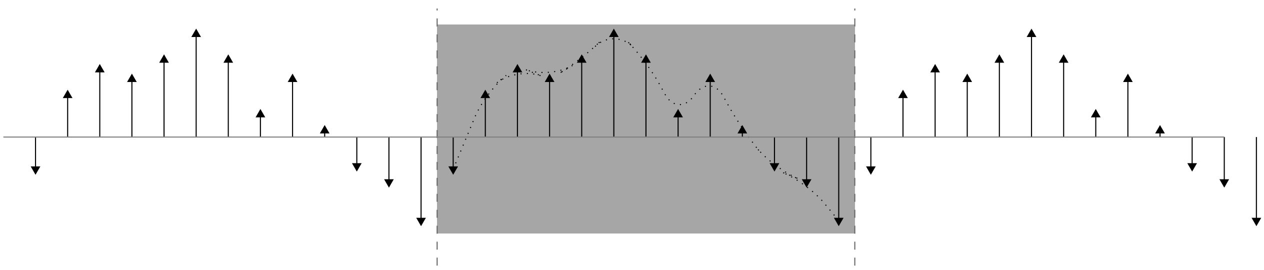


Figure 2: Sampling DTFT causes repetition of samples in time domain

The sampled version of the DTFT is known as Discrete Fourier Transform or DFT. Here the frequency is discrete or sampled as opposed to the DTFT, which is discrete in time and continuous in frequency. The power of DFT lies in the availability of an effective way of computing it, known as the Fast Fourier Transform (FFT). FFT is an algorithm (or a collection of algorithms) which evaluates the DFT in a quick manner.

4 Discrete Fourier Transform (DFT)

The Discrete Fourier Transform is now a days an indispensable tool in a signal processor's kit. Its at most importance is partly due to the fact that it is the only version of the Fourier Transform which can be calculated on a computer or finite-precision digital signal processor. But what is DFT?. DFT is nothing but the samples of DTFT, in a fashion very similar to the Fourier Series coefficients being samples of the Fourier Transform. In fact, the connection between FS and DFT is much more than this perceptible similarity. It is the discrete-time version of the FS in a stronger sense, which will become clear as we go on.

Let us now sample the DTFT $\hat{X}(f)$ at equi-spaced intervals which are apart by $\frac{1}{N}$ to obtaine

$$\hat{X}(\frac{k}{N}) = \sum_{n=0}^{N-1} x[n] \exp(-j2\pi \frac{k}{N} n). \tag{11}$$

Let us denote $X[k] = \hat{X}(\frac{k}{N})$. Thus,

$$X[k] = \sum_{n=0}^{N-1} x[n] \exp(-j2\pi \frac{k}{N} n) \ , \quad 0 \le k \le N-1. \tag{12}$$

The $X[k]$ above is known as the N−point DFT of $x[n]$ [2].

Let $\bar{X}$ be a $N \times 1$ column vector containing the DFT values of a signal $x[n]$. The DFT has a pleasing Matrix representation.

$$\bar{X} = A\bar{x}, \tag{13}$$

where $\bar{x}$ is column vector containing the signal samples and the the element at (m, n) of the matrix A is

$$A_{m,n} = \exp(-j2\frac{\pi}{N}mn)\ ,\ 0 \le m, n, \le N-1. \tag{14}$$

Exercise 3. *Show that the above matrix representation is equivalent to the expression* (12)*.*

Exercise 4. *A square matrix Q is **orthonormal** if $QQ^\dagger = QQ^\dagger = I$, where † represents the conjugate transpose, i.e., $Q^\dagger = (Q^T)^*$. Show that the Fourier matrix A satisfies $AA^\dagger = NI$. Using this, write the DFT as a multiplication by an orthonormal matrix.*

Since $A^\dagger A = NI_N$, where I_N is the $N \times N$ identity matrix, we can easily invert (13). All we have to do is to multiply the expression by $\frac{1}{N}A^\dagger$. Thus,

$$\frac{1}{N}A^\dagger \bar{X} = \frac{1}{N}A^\dagger A x = x. \tag{15}$$

This is known as the Inverse Discrete Fourier Transform or the IDFT. Expanding the matrix elements,

$$x[n] = \frac{1}{N}\sum_k X[k]\exp(+j2\frac{\pi}{N}nk). \tag{16}$$

Note the striking similarity between the forward the inverse DFTs. Their difference is mostly in our perception of time and frequency, in fact we can interchange the concepts and operations in these two domains. Essentially, the sampled waveforms and their respective DFT representations are as such devoid of any bindings to measurable quantities like time and frequency. They are merely numbers (in $\mathbb{C}$), to be stored and played at our convenience, but to be handled with due respect. Let us collect the definitions of DFT and IDFT here for future use,

$$X[k] = \sum_{n=0}^{N-1} x[n]\exp(-j2\frac{\pi}{N}kn)\ ,\quad 0 \le k \le N-1 \tag{17}$$

$$x[n] = \frac{1}{N}\sum_{k=0}^{N-1} X[k]\exp(+j2\frac{\pi}{N}nk)\ ,\quad 0 \le n \le N-1. \tag{18}$$

5 Fast Fourier Transform (FFT)

FFT is an effective way of computing DFT. With the growth of digital technologies, it became evident that fast and parallel ways of computing matrix products is key to many signal processing/communication applications. We have already seen that DFT and IDFT

[2] Some sources define DFT with a scale of $\frac{1}{N}$, and some others with a scale of $\frac{1}{\sqrt{N}}$. However, in those cases a corresponding scale is applied on the inverse DFT

effectively does matrix multiplications. Multipying an arbitrary $N \times N$ matrix with a vector amounts to N^2 multiplications and $N(N-1)$ additions. We will say that a matrix-vector product has $O(N^2)$ computations[3]. How do we speed up matrix multiplications. One strategy is to identify the structural properties of the matrices. Recall that DFT matrix A has the property that

$$A_{mn} = \exp(-j\frac{2\pi}{N}mn).$$

Let us illustrate the structure of the matrix A by taking the 4–point DFT of a vector $\bar{x} = (x_0, x_1, x_2, x_3)^T$. With $\alpha = e^{-j\frac{\pi}{2}}$

$$\begin{bmatrix} X_0 \\ X_1 \\ X_2 \\ X_3 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & \alpha & \alpha^2 & \alpha^3 \\ 1 & \alpha^2 & \alpha^4 & \alpha^6 \\ 1 & \alpha^3 & \alpha^6 & \alpha^9 \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \end{bmatrix}. \tag{19}$$

Substituting for α,

$$\bar{X} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -j & -1 & j \\ 1 & -1 & 1 & -1 \\ 1 & j & -1 & -j \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \end{bmatrix}. \tag{20}$$

The above product can be rewritten as,

$$\bar{X} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & -j & j \\ 1 & 1 & -1 & -1 \\ 1 & -1 & j & -j \end{bmatrix} \begin{bmatrix} x_0 \\ x_2 \\ x_1 \\ x_3 \end{bmatrix}, \tag{21}$$

where we swapped the input elements x_1 and x_2 and correspondingly changed the associated columns of A. See that some structure emerges in the matrix. We can further simplify the product as,

$$\bar{X} = \begin{bmatrix} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} x_0 \\ x_2 \end{bmatrix} \\ \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} x_0 \\ x_2 \end{bmatrix} \end{bmatrix} + \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & -j & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & j \end{bmatrix} \begin{bmatrix} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_3 \end{bmatrix} \\ \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_3 \end{bmatrix} \end{bmatrix} \tag{22}$$

Notice the clear structure, which resembles a butterfly network as shown below. In the

$$\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix} \quad \Leftrightarrow$$

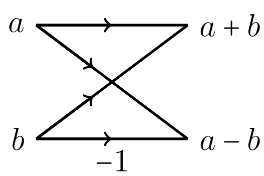
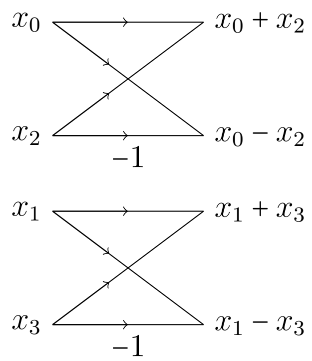


Figure 3: Butterfly Structure of FFT

butterfly structure, each incoming node simply adds all the values coming in. The label on

[3] $O(N)$ as in Knuth's notation.

the branch represents the multiplicative factor that we need to apply on the value which is being passed through that link. If there is no label, the multiplicative factor is unity. From (22) , apart from the multiplication by a diagonal matrix, only two operations are required, each operation comprising of an addition and a subtraction. A schematic representation of this will enable generalizing our 4-pt DFT to N-pt DFTs, where N is a power of 2. To this end, notice that the essential building block is the butterfly shown in Figure 3. The butterfly needs to be applied to both (x_0, x_2) and (x_1, x_3) separately. Now we have to scale



the outputs of the butterfly networks as per (22) and combine them. This task can be made simple by observing that,

$$\begin{bmatrix} X_0 \\ X_1 \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} x_0 \\ x_2 \end{bmatrix} + \begin{bmatrix} 1 & 0 \\ 0 & -j \end{bmatrix} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_3 \end{bmatrix}$$
$$\begin{bmatrix} X_2 \\ X_3 \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} x_0 \\ x_2 \end{bmatrix} - \begin{bmatrix} 1 & 0 \\ 0 & -j \end{bmatrix} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_3 \end{bmatrix} \tag{23}$$

Notice that the new structure is nothing but another butterfly overlayed on top of the original. Before the second butterfly stage, we multiply the bottom-most path by α, which

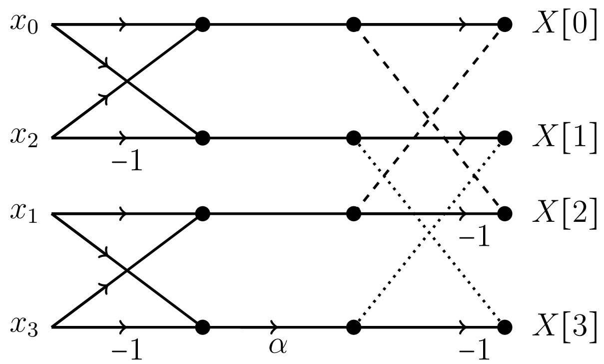


Figure 4: Butterfly for 4−point FFT

evaluates to $-j$ for a 4−point DFT. This is to generate the output $-j(x_1 - x_3)$, which is needed in (23). Notice that the multiplication by diagonal matrix in (23) is nothing but generating $x_1 + x_3$ and $-j(x_1 - x_3)$.

Let us now extend our FFT to compute 8−point DFT. A brute-force development is presented, which clearly shows the underlying structure. Let us denote

$$\alpha = \exp(-j\frac{2\pi}{8})$$

$$\begin{bmatrix} 1 & 0 \\ 0 & -j \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix} \quad \Leftrightarrow$$

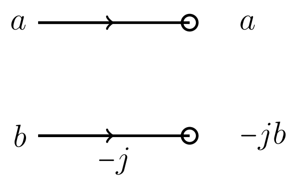


Figure 5: Intermediate Structure

$$A = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & \alpha^1 & \alpha^2 & \alpha^3 & \alpha^4 & \alpha^5 & \alpha^6 & \alpha^7 \\ 1 & \alpha^2 & \alpha^4 & \alpha^6 & \alpha^0 & \alpha^2 & \alpha^4 & \alpha^6 \\ 1 & \alpha^3 & \alpha^6 & \alpha^1 & \alpha^4 & \alpha^7 & \alpha^2 & \alpha^5 \\ 1 & \alpha^4 & \alpha^0 & \alpha^4 & \alpha^0 & \alpha^4 & \alpha^0 & \alpha^4 \\ 1 & \alpha^5 & \alpha^2 & \alpha^7 & \alpha^4 & \alpha^1 & \alpha^6 & \alpha^3 \\ 1 & \alpha^6 & \alpha^4 & \alpha^2 & \alpha^0 & \alpha^6 & \alpha^4 & \alpha^2 \\ 1 & \alpha^7 & \alpha^6 & \alpha^5 & \alpha^4 & \alpha^3 & \alpha^2 & \alpha^1 \end{bmatrix}$$

Notice that we used the modulo arithmetic rule that $\alpha^N = 1$. Rearranging the input vector, Ax takes the form shown in Figure 6. The structure of the matrix product becomes

$$Ax = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & \alpha^4 & \alpha^2 & \alpha^6 \\ 1 & \alpha^0 & \alpha^4 & \alpha^4 \\ 1 & \alpha^4 & \alpha^6 & \alpha^2 \\ 1 & \alpha^0 & \alpha^0 & \alpha^0 \\ 1 & \alpha^4 & \alpha^2 & \alpha^6 \\ 1 & \alpha^0 & \alpha^4 & \alpha^4 \\ 1 & \alpha^4 & \alpha^6 & \alpha^2 \end{bmatrix} \begin{bmatrix} x_0 \\ x_4 \\ x_2 \\ x_6 \end{bmatrix} + \begin{bmatrix} 1 & 1 & 1 & 1 \\ \alpha^1 & \alpha^5 & \alpha^3 & \alpha^7 \\ \alpha^2 & \alpha^2 & \alpha^6 & \alpha^6 \\ \alpha^3 & \alpha^7 & \alpha^1 & \alpha^5 \\ \alpha^4 & \alpha^4 & \alpha^4 & \alpha^4 \\ \alpha^5 & \alpha^1 & \alpha^7 & \alpha^3 \\ \alpha^6 & \alpha^6 & \alpha^2 & \alpha^2 \\ \alpha^7 & \alpha^3 & \alpha^5 & \alpha^1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_5 \\ x_3 \\ x_7 \end{bmatrix}$$

Figure 6: Structure of DFT

apparent by noticing that,

$$Ax = \begin{bmatrix} B_1 & B_2 \\ B_1 & -B_2 \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix}, \tag{24}$$

where $u = [x_0, x_4, x_2, x_6]^T$ and $v = [x_1, x_5, x_3, x_7]^T$. Furthermore, row j, $0 \le j \le \frac{N}{2} - 1$ of the matrix B_2 is readily obtained my multiplying the same row of B_1 by α^j. Thus we need to essentially find the products $B_1 u$ and $B_1 v$. But $B_1 u$ is nothing but multiplying u with the 4–point FFT matrix as in (21), which is same as the 4–point DFT of the sequence $[x_0, x_2, x_4, x_6]$. Observe that we changed the order of the input data in (21). Thus, we end up with a very simple structure.

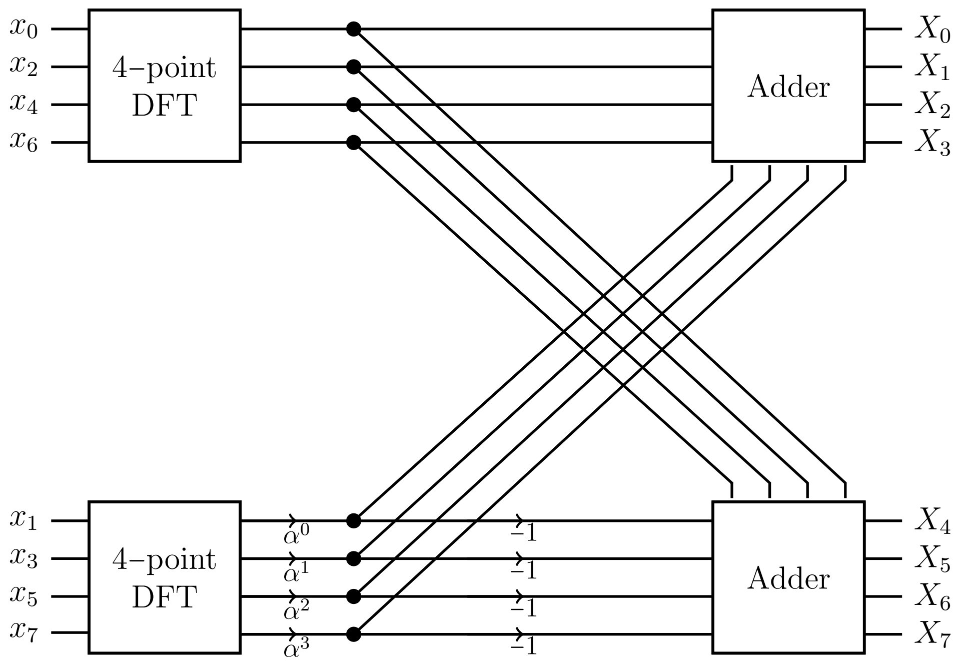


Figure 7: Seeing the Butterflies in 8–point FFT

In retrospect, we could have depicted the $8-FFT$ structure directly from our discussion on 4–point FFT. However, we took the longer route so that all the aspects of generalizing the FFT become transparent. Thus, with $\alpha = \exp(-j\frac{2\pi}{N})$, a N-FFT can be pictorially represented as a bifurcating recursion as shown in Figure 8

6 Circular Convolution

Let

$$x[n] \overset{DFT}{\rightleftharpoons} X[k]$$

$$h[n] \overset{DFT}{\rightleftharpoons} H[k].$$

What will be the inverse DFT of $X[k]H[k]$. Can we write it as a convolution between $x[n]$ and $h[n]$. The answer is close to affirmative, 'close' because due care is needed. In particular $X[k]$ is a sampled version of the DTFT $\tilde{X}(f)$, which implies that the discrete time signal is periodically repeated in time. If N samples of the DTFT are taken in frequency, in the

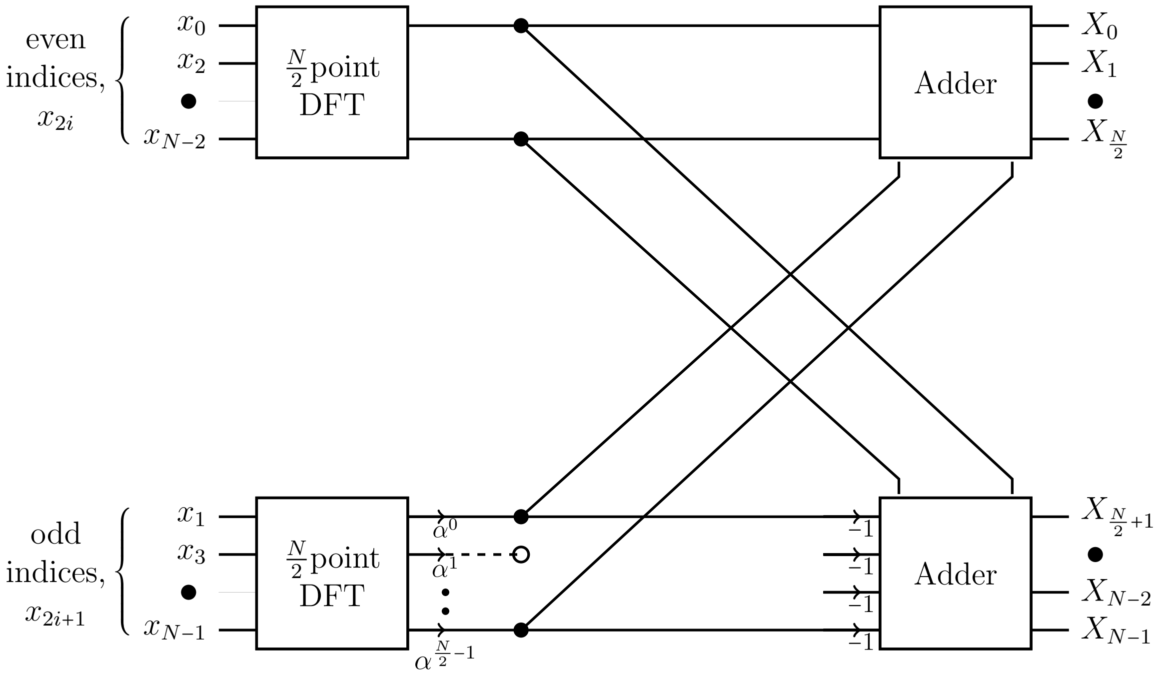


Figure 8: Recursive Computation of $N-$point FFT

time domain we will get non-overlapping periodic replicas of $x[n]$ as the samples. By our formal rules of Dirac manipulation, $X[k]H[k]$ is same as $\tilde{H}(f = k\frac{2f_m}{N})X[k]$, which in time domain will imply that a periodic repetition of $x[n]$ is convolved with $h[n]$. This is known as circular convolution. Notice that the output that we obtain is periodic, and hence we need to take only one such interval into consideration.