

Deney 3:

Ayrık Fourier Dönüşümü (DFT) & Hızlı Fourier Dönüşümü (FFT)

Amaç

Bu deneyin amacı Ayrık Fourier Dönüşümü (DFT) ve Hızlı Fourier Dönüşümünün (FFT) tanıtılmasıdır.

Giriş

Bir önceki deneyde ayrık-zamanlı Fourier dönüşümünü incelemiştik. Ayrık-zamanlı Fourier dönüşümü mutlak toplanabilir diziler için frekans bölgesi gösterimini sağlamaktadır. Ancak, Fourier dönüşümü sonsuz uzunlukta bir dizi için tanımlıdır ve daha önemlisi, sürekli bir değişken olan ω açısal frekansının bir fonksiyonudur. MATLAB kullanırken dizileri sınırlandırmamız ve sınırlı sayıda nokta için değerlendirme yapmamız gerekir. Ayrık Fourier Dönüşümü (DFT) bu problemleri gidermektedir.

Ayrık zamanlı Fourier dönüşümü (DFT), ayrık zamanlı sinyal işleme algoritma ve sistemlerinin analizi, tasarımı, gerçekleştirilmesi ile doğrusal filtreleme, korelasyon analizi ve spektrum analizi gibi sinyal işleme uygulamalarında önemli bir rol oynar. DFT'nin bu öneme sahip olmasının ardındaki temel neden DFT'yi hesaplamakta kullanılan verimli algoritmaların varlığıdır.

DFT, Fourier dönüşümünün eşit aralıklı frekanslardaki örneklerine özdeştir. Sonuç olarak N -noktalı bir DFT'nin hesaplanması Fourier dönüşümünün N örneğinin, N eşit aralıklı frekanslarla ($w_k = 2\pi k / N$), z -düzlemindeki birim çember üzerinde N nokta ile hesaplanmasına karşılık gelir. Burada temel amaç N -noktalı DFT'nin hesaplanması için verimli algoritmaların kullanılmasıdır. Bu algoritmalar ortak olarak hızlı Fourier dönüşümü (FFT) algoritmaları adını alır.

Ayrık Fourier Dönüşümü (DFT)

Ayrık-zaman Fourier dönüşümü (DTFT) aşağıdaki şekilde verilir.

$$X(e^{j\omega}) = F\{x(n)\} = \sum_{n=-\infty}^{\infty} x(n)e^{-j\omega n} \quad (3.1)$$

$F\{\}$ operatörü $x(n)$ ayrık işaretini gerçel değerli ω 'nın karmaşık değerli ve sürekli bir fonksiyonu olan $X(e^{j\omega})$ 'ya dönüştürmektedir. $X(e^{j\omega})$ 'yı $\Delta\omega$ aralıklarıyla periyodik olarak örneklediğimizi varsayalım. $X(e^{j\omega})$, 2π periyodunda bir fonksiyon olduğundan dolayı sadece temel frekans aralığında kalan örnekler yeterli olacaktır. Kolaylık için $0 \leq \omega < 2\pi$ aralığında N tane eşit aralıklı örnek alalım ($\Delta\omega = 2\pi/N$). Eğer (3.1) denklemini $\omega = 2\pi k/N$ noktasında değerlendirecek olursak:

$$X(e^{j2\pi k/N}) = \sum_{n=-\infty}^{\infty} x(n)e^{-j2\pi kn/N} \quad k = 0, 1, 2, \dots, N-1 \quad (3.2)$$

L uzunluğunda sınırlı uzunluklu bir $x(n)$ dizisinin ($x(n) = 0, n < 0$ ve $n \geq L$ için) Fourier dönüşümü aşağıdaki şekilde verilir:

$$X(e^{j\omega}) = F\{x(n)\} = \sum_{n=0}^{L-1} x(n)e^{-j\omega n} \quad (3.3)$$

$X(e^{j\omega})$ 'yı eşit aralıklı $\omega_k = 2\pi k/N, k = 0, 1, 2, \dots, N-1$ ($N \geq L$) frekanslarında örneklediğimizde oluşan örnekler

$$X(k) = X(e^{j2\pi k/N}) = \sum_{n=0}^{L-1} x(n)e^{-j2\pi kn/N} \quad (3.4)$$

şeklindedir. $n \geq L$ için $x(n) = 0$ olduğundan,

$$\boxed{X(k) = \sum_{n=0}^{N-1} x(n)e^{-j2\pi kn/N} \quad k = 0, 1, 2, \dots, N-1} \quad (3.5)$$

elde edilir. Burada verilen ilişki ($L \leq N$) uzunluklu bir $X(n)$ dizisini N uzunluklu bir frekans dizisine $\{X(k)\}$ dönüştürme işlemidir. Frekans örnekleri, Fourier dönüşümü $X(e^{j\omega})$ 'yı N ayrık frekans noktasında değerlendirerek bulunduğu için, (3.5)'de verilen ilişki $x(n)$ 'nin Ayrık Fourier Dönüşümü (DFT) olarak adlandırılır.

$$\boxed{x(n) = \frac{1}{N} \sum_{k=0}^{N-1} X(k)e^{j2\pi kn/N} \quad n = 0, 1, 2, \dots, N-1} \quad (3.6)$$

işlemi ise ters DFT (IDFT) olarak adlandırılır.

DFT'nin özellikleri

Özellikler ispatsız olarak verilecektir. $X(k)$, $x(n)$ 'in N -noktalı DFT'si olsun.

1) Doğrusallık

$$\text{DFT} \{ax(n) + by(n)\} = a \text{DFT} \{x(n)\} + b \text{DFT} \{y(n)\} \quad (3.7)$$

2) Periyodiklik

$$\begin{aligned} x(n+N) &= x(n) \\ X(k+N) &= X(k) \end{aligned} \quad (3.8)$$

Periyodiklik özelliği DFT ve IDFT dönüşüm formüllerinden çıkmaktadır.

3) Bir dizinin dairesel simetrikliği

Gördüğümüz üzere, $L \leq N$ uzunluklu $x(n)$ dizisinin N -noktalı DFT'si, $x(n)$ in periyodik olarak genişletilmesiyle elde edilen N periyotlu $x_p(n)$ dizisinin

$(x_p(n) = \sum_{l=-\infty}^{\infty} x(n-lN))$ N -noktalı DFT sine eşittir. Periyodik $x_p(n)$ dizisini k kadar

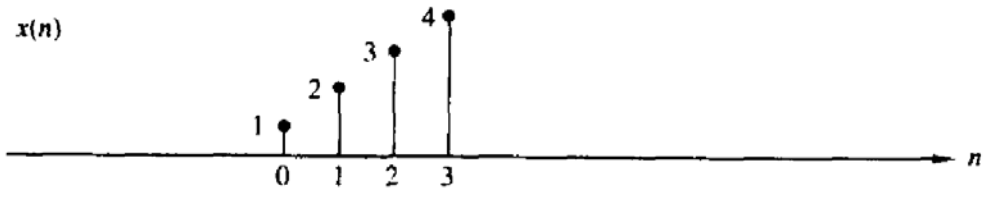
sağa ötelediğimizi varsayalım. Böylece farklı bir periyodik dizi elde ederiz.

$$x'_p(n) = x_p(n-k) = \sum_{l=-\infty}^{\infty} x(n-k-lN) \quad (3.9)$$

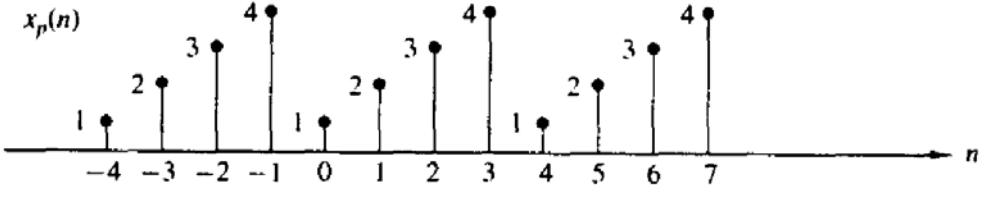
Sınırlı uzunluklu

$$x'(n) = \begin{cases} x'_p(n), & 0 \leq n \leq N-1 \\ 0, & \text{dışında} \end{cases}$$

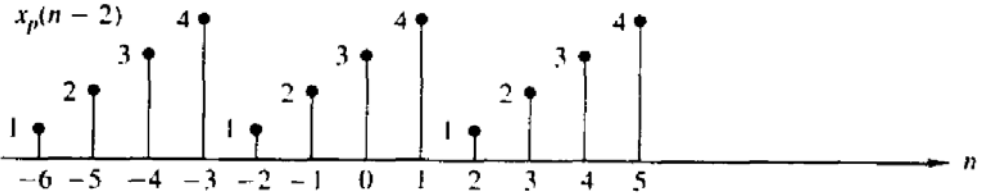
dizisi $x(n)$ dizisinin dairesel ötelemesiyle elde edilmiştir. Bu ilişki Şekil 3.1'de $N=4$ için gösterilmiştir.



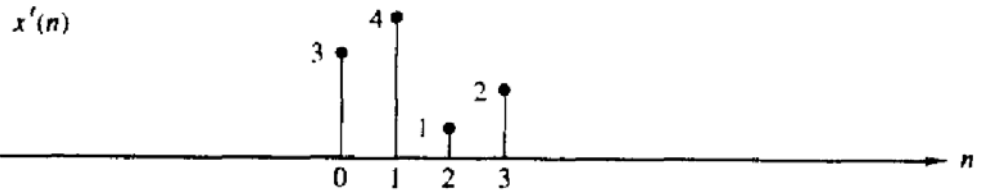
(a)



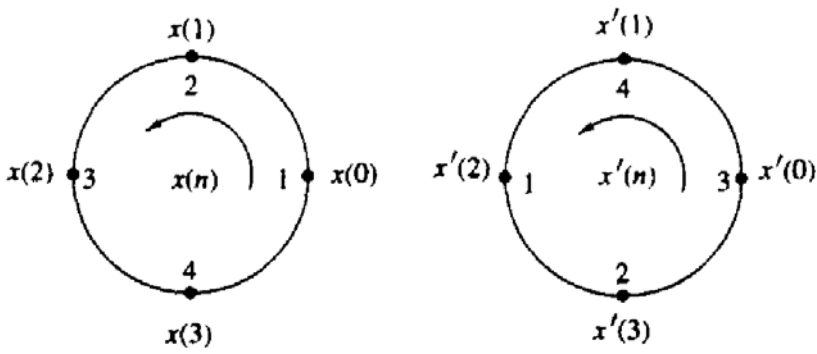
(b)



(c)



(d)



(e)

Şekil 3.1: Bir dizinin dairesel ötelenmesi

4) DFT'nin simetri özellikleri

DFT'nin simetri özellikleri Fourier dönüşümü için uygulanan yöntemin tekrarıyla elde edilebilir. N -uzunluklu $x(n)$ ve DFT'sinin karmaşık değerli olduğunu varsayalım. Diziler aşağıdaki şekilde verilebilir.

$$x(n) = x_R(n) + jx_I(n) \quad 0 \leq n \leq N-1 \quad (3.10)$$

$$X(k) = X_R(k) + jX_I(k) \quad 0 \leq k \leq N-1 \quad (3.11)$$

(3.10)'u, (3.5)'de verilen DFT ifadesine yerleştirerek,

$$X_R(k) = \sum_{n=0}^{N-1} [x_R(n) \cos \frac{2\pi kn}{N} + x_I(n) \sin \frac{2\pi kn}{N}] \quad (3.12)$$

$$X_I(k) = -\sum_{n=0}^{N-1} [x_R(n) \sin \frac{2\pi kn}{N} - x_I(n) \cos \frac{2\pi kn}{N}] \quad (3.13)$$

elde edilir. Benzer şekilde (3.11)'i, (3.6)'da verilen IDFT ifadesine yerleştirerek

$$x_R(n) = \frac{1}{N} \sum_{k=0}^{N-1} [X_R(k) \cos \frac{2\pi kn}{N} - X_I(k) \sin \frac{2\pi kn}{N}] \quad (3.14)$$

$$x_I(n) = \frac{1}{N} \sum_{k=0}^{N-1} [X_R(k) \sin \frac{2\pi kn}{N} + X_I(k) \cos \frac{2\pi kn}{N}] \quad (3.15)$$

5) Gerçel-değerli diziler

$x(n)$ gerçel-değerli ise, DFT tanımından aşağıdaki özellik bulunur.

$$X(N-k) = X^*(k) = X(-k) \quad (3.16)$$

Dolayısıyla, $|X(N-k)| = |X(k)|$ olacaktır.

6) Gerçel ve çift diziler

$x(n)$ gerçel ve çift ise, yani

$$x(n) = x(N-n) \quad 0 \leq n \leq N-1 \quad (3.17)$$

(3.13)'den $X_I(k) = 0$ olur. Böylece DFT

$$X(k) = X_R(k) = \sum_{n=0}^{N-1} x(n) \cos \frac{2\pi kn}{N}, \quad 0 \leq k \leq N-1 \quad (3.18)$$

şeklinde basitleşir. Bu fonksiyon gerçel ve çifttir. $X_I(k) = 0$ olduğundan IDFT aşağıdaki şekilde basitleşir.

$$x(n) = \frac{1}{N} \sum_{k=0}^{N-1} X(k) \cos \frac{2\pi kn}{N}, \quad 0 \leq n \leq N-1 \quad (3.19)$$

7) Gerçel ve tek diziler

$x(n)$ gerçel ve tek ise, yani

$$x(n) = -x(N-n), \quad 0 \leq n \leq N-1 \quad (3.20)$$

(3.12)'den $X_R(k) = 0$ olur. Böylece DFT

$$X(k) = -j \sum_{n=0}^{N-1} x(n) \sin \frac{2\pi kn}{N}, \quad 0 \leq k \leq N-1 \quad (3.21)$$

şeklinde basitleşir. Bu fonksiyon karmaşık ve tektir. Bundan dolayı IDFT aşağıdaki şekilde basitleşir.

$$x(n) = \frac{j}{N} \sum_{k=0}^{N-1} X(k) \sin \frac{2\pi kn}{N}, \quad 0 \leq n \leq N-1 \quad (3.22)$$

8) Sanal değerli diziler ($x(n) = jx_I(n)$)

Bu durumda

$$X_R(k) = \sum_{n=0}^{N-1} x_I(n) \sin \frac{2\pi kn}{N} \quad (3.23)$$

$$X_I(k) = \sum_{n=0}^{N-1} x_I(n) \cos \frac{2\pi kn}{N} \quad (3.24)$$

9) Dairesel evrişim

$$x_1(n) \leftrightarrow X_1(k) \quad x_2(n) \leftrightarrow X_2(k) \quad (3.25)$$

ise

$$x_1(n) \circledast x_2(n) \leftrightarrow X_1(k)X_2(k) \quad (3.26)$$

Burada,

$$x_1(n) \circledast x_2(n) = \sum_{m=0}^{N-1} x_1(m)x_2((n-m))_N \quad (3.27)$$

olup \circledast dairesel evrişimi belirtmektedir.

$$((n-m))_N = (n-m) \bmod N$$

Dairesel evrişimin nasıl gerçekleştiğini daha sonra bir örnekle göreceğiz (Örnek 3.2).

10) Çarpım

$$x_1(n)x_2(n) \leftrightarrow \frac{1}{N} X_1(k) \textcircled{N} X_2(k) \quad (3.28)$$

Örnek 3.1

$x(n)$, 4-noktalı bir dizi olsun.

$$x(n) = \begin{cases} 1 & 0 \leq n \leq 3 \\ 0 & \text{dışında} \end{cases}$$

a) Ayrık zaman Fourier dönüşümünü hesaplayın ve genlik ve fazını çizin.

b) $x(n)$ 'in 4-noktalı DFT'sini hesaplayın.

c) $x(n)$ 'in 16-noktalı DFT'sini hesaplayın.

Çözüm

a) Ayrık-zaman Fourier dönüşümü aşağıdaki şekilde verilir.

$$X(e^{j\omega}) = \sum_{n=0}^3 x(n)e^{-j\omega n} = 1 + e^{-j\omega} + e^{-j2\omega} + e^{-j3\omega} = \frac{1 - e^{-j4\omega}}{1 - e^{-j\omega}} = \frac{\sin 2\omega}{\sin(\omega/2)} e^{-j3\omega/2}$$

Böylece

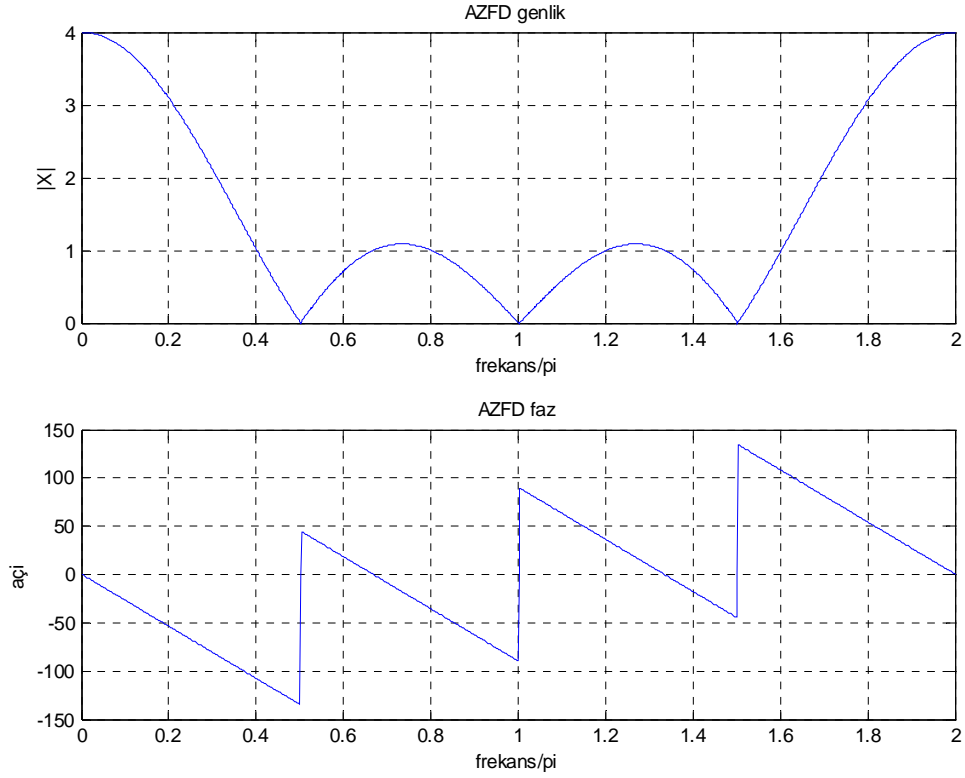
$$|X(e^{j\omega})| = \frac{\sin 2\omega}{\sin(\omega/2)} \quad \text{ve} \quad \angle X(e^{j\omega}) = \begin{cases} \frac{-3\omega}{2} & \frac{\sin 2\omega}{\sin(\omega/2)} > 0 \\ \frac{-3\omega}{2} \pm \pi & \frac{\sin 2\omega}{\sin(\omega/2)} < 0 \end{cases}$$

b) (5.7) kullanarak 4-noktalı DFT hesaplayalım.

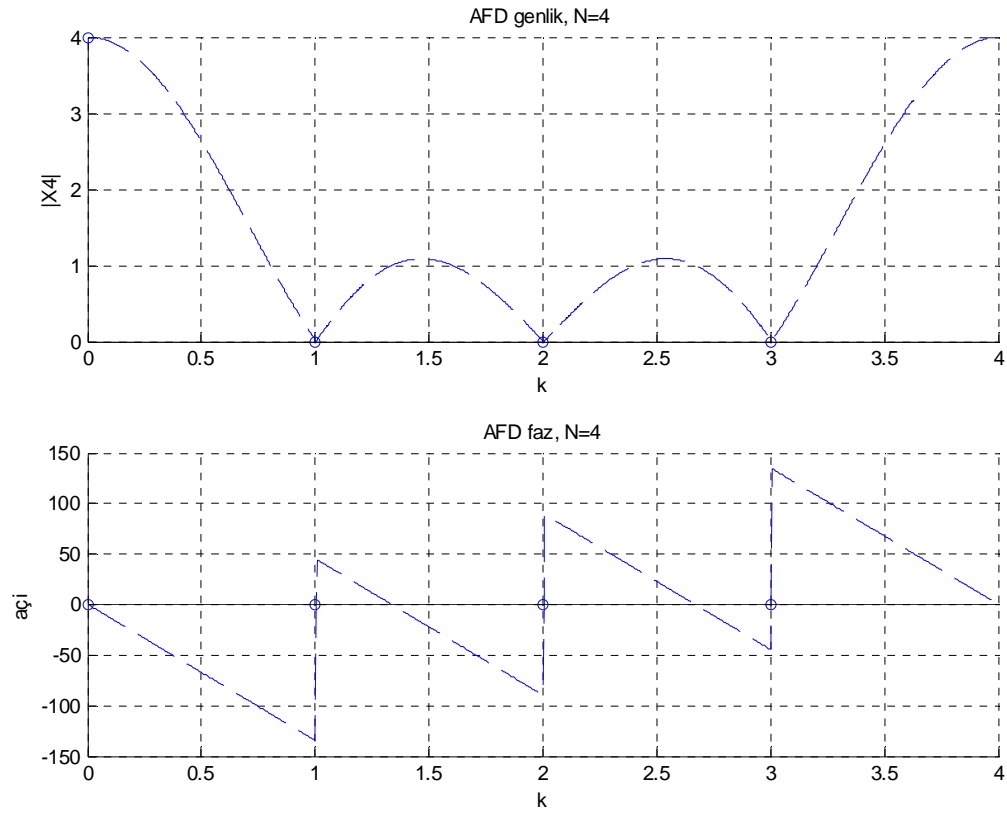
$$X_4(k) = \sum_{n=0}^3 x(n)e^{-j2\pi kn/4} \quad k = 0, 1, 2, 3$$

MATLAB kullanarak bunu yapabiliriz.

```
>>n=[0:N-1];           % n için satır vektörü
>>k=[0:N-1];           % k için satır vektörü
>>WN=exp(-1i*2*pi/N);  % Wn vektörü
>>nk=n'*k;             % nk değerlerini içeren NxN matris
>>WNnk=WN.^nk;        % DFT matrisi
>>X4=x*WNnk;           % DFT katsayıları için satır vektörü
>> magX4=abs(X4); phaX4=angle(X4);
```



Şekil 3.2: Genlik ve faz diyagramı



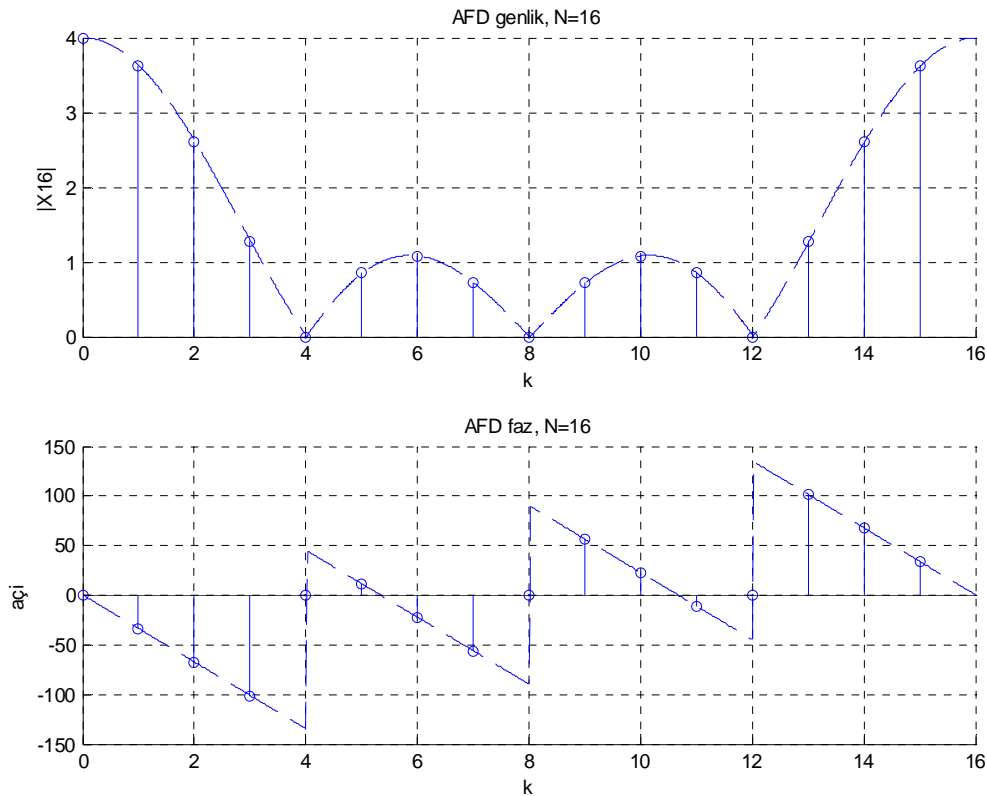
Şekil 3.3: 4 -noktalı DFT

c) (3.5)'i kullanarak 16-noktalı DFT hesaplayalım.

$$X_{16}(k) = \sum_{n=0}^{15} x(n)e^{-j2\pi kn/16} \quad k = 0,1,2,\dots,15$$

MATLAB kullanarak bunu yapabiliriz.

```
>>x16=[x zeros(1,N-length(x))]; % sıfır dolgulama
>>n=[0:N-1]; % n için satır vektörü
>>k=[0:N-1]; % k için satır vektörü
>>WN=exp(-j*2*pi/N); % Wn vektörü
>>nk=n'*k; % nk değerlerini içeren NxN matris
>>WNnk=WN.^nk; % DFT matrisi
>>X16=x16*WNnk; % DFT katsayıları için satır vektörü
>>magX16=abs(X16); phaX16=angle(X16);
```



Şekil 3.4: 16 -noktalı DFT

DTFT, 4-noktalı DFT ve 16-noktalı DFT karşılaştırılarak DFT'nin DTFT'nin örneklenmesinden oluştuğunu görebiliriz. Frekans bölgesinde alınan örnek sayısı DFT'nin uzunluğunu vermektedir.

Örnek 3.2 :

$x_1(n) = \{1, 2, 3\}$ ve $x_2(n) = \{1, 2, 3, 4\}$ olmak üzere 4-noktalı dairesel evrişimi $(x_1(n) \textcircled{4} x_2(n))$ hesaplayınız.

Çözüm:

$x_1(n)$ 3-noktalı bir dizi olduğundan ilk önce onu sıfır dolgulayarak 4-noktalı uzunluğa getirmemiz gerekir. Bu problemi hem zaman hem frekans bölgesinde çözeceğiz. Zaman bölgesinde (3.28) kullanılırken frekans bölgesinde ise (3.5) ile verilen DFT kullanılacaktır.

1) Zaman bölgesi yaklaşımı

4-noktalı dairesel evrişim aşağıdaki şekilde verilir.

$$x_1(n) \textcircled{4} x_2(n) = \sum_{m=0}^3 x_1(m)x_2((n-m))_4$$

dairesel olarak katlanmış ve ötelenmiş $x_2((n-m))_4$ dizisini oluşturmamız, bunu $x_1(m)$ ile çarpmamız, ve her n için bu çarpım değerlerini toplamamız gerekmektedir.

$x_1(n) = \{1, 2, 3, 0\}$ ve $x_2(n) = \{1, 2, 3, 4\}$ göz önüne alalım.

$n = 0$ için

$$\begin{aligned} \sum_{m=0}^3 x_1(m)x_2((0-m))_4 &= \sum_{m=0}^3 [\{1, 2, 3, 0\} \cdot \{1, 4, 3, 2\}] \\ \sum_{m=0}^3 [\{1, 8, 9, 0\}] &= 18 \end{aligned}$$

$n = 1$ için

$$\begin{aligned} \sum_{m=0}^3 x_1(m)x_2((1-m))_4 &= \sum_{m=0}^3 [\{1, 2, 3, 0\} \cdot \{2, 1, 4, 3\}] \\ \sum_{m=0}^3 [\{2, 2, 12, 0\}] &= 16 \end{aligned}$$

$n = 2$ için

$$\begin{aligned} \sum_{m=0}^3 x_1(m)x_2((2-m))_4 &= \sum_{m=0}^3 [\{1, 2, 3, 0\} \cdot \{3, 2, 1, 4\}] \\ \sum_{m=0}^3 [\{3, 4, 3, 0\}] &= 10 \end{aligned}$$

$n = 3$ için

$$\sum_{m=0}^3 x_1(m)x_2((3-m))_4 = \sum_{m=0}^3 [\{1,2,3,0\} \cdot \{4,3,2,1\}]$$

$$\sum_{m=0}^3 [\{4,6,6,0\}] = 16$$

Böylece $x_1(n) \textcircled{4} x_2(n) = \{18,16,10,16\}$ olur.

2) Frekans bölgesi yaklaşımı

Bu yaklaşımda ilk önce $x_1(n)$ ve $x_2(n)$ için 4-noktalı DFT hesaplanır ve bunlar birbiriyle çarpılır. Sonucun IDFT'si alınarak dairesel evrişim elde edilir.

$x_1(n)$ nin DFT'si

$$x_1(n) = \{1,2,3,0\}, X_1(k) = \{6, -2-2j, 2, -2+2j\}$$

$x_2(n)$ nin DFT'si

$$x_2(n) = \{1,2,3,4\}, X_2(k) = \{10, -2+2j, -2, -2-2j\}$$

$$X_1(k)X_2(k) = \{60, 8, -4, 8\}$$

IDFT sonrası $x_1(n) \textcircled{4} x_2(n) = \{18,16,10,16\}$

Ödev 3.1:

1) 32 uzunluklu $x(n) = 5 \cos(\frac{\pi}{7}n) + 3 \sin(\frac{3\pi}{10}n)$, $0 \leq n \leq 31$, işareti verilmiş olsun. Dizinin DFT'sini $N = 32, 128, 512$ uzunlukları için hesaplayınız ve bu N -noktalı DFT'lerin genliğini çizdiriniz. DTFT, 32-noktalı DFT, 128-noktalı DFT ve 512-noktalı DFT'yi karşılaştırınız (DFT hesaplamak için bir program yazmalısınız).

2) a) MATLAB'de dairesel evrişim işleminin gerçekleştiren bir program yazınız. (Yazdığınız program herhangi N değeri için çalışmalı ve girişine verdiğiniz aynı N uzunluklu iki dizinin dairesel evrişimini zaman bölgesinde hesaplamalıdır.)

b) $x1=[0:0.1:0.9]$; $x2=[2:-0.1:1.1]$; komutları ile üreteceğiniz iki dizinin dairesel evrişimini a) şıkında yazmış olduğunuz program ile elde ediniz.

3) Aşağıdaki diziler için dairesel evrişimi zaman bölgesinde hesaplayınız. MATLAB yardımıyla frekans bölgesinde de hesaplayınız. 2. soruda yazmış olduğunuz dairesel evrişim programını bu iki dizi için çalıştırarak bulduğunuz sonuçları doğrulayınız.

$$x_1(n) = \{0,1,1,-1\}, x_2(n) = \{1,0,-1,1\}$$

Hızlı Fourier Dönüşümü (FFT)

Ayrık Fourier dönüşümünü tekrar hatırlayacak olursak,

$$X(k) = \sum_{n=0}^{N-1} x(n)W_N^{nk} \quad 0 \leq k \leq N-1 \quad (3.29)$$

Burada $W_N = e^{-j(2\pi/N)}$ 'dir. (3.29) denklemiyle gösterilen Ayrık Fourier Dönüşümünün (DFT) doğrudan hesaplanmasında her bir $X(k)$ değeri için N adet karmaşık çarpma ve $N-1$ adet karmaşık toplama işlemi yapılmaktadır. Bundan dolayı N adet DFT değeri bulunurken, N^2 adet çarpma ve $N(N-1)$ adet toplama işlemi gereklidir. Ayrıca her karmaşık çarpma işlemi dört gerçel çarpma ve iki gerçel toplama işlemi ve her bir karmaşık toplama iki gerçel toplama işlemi ile gerçekleştirilmektedir. Sonuç olarak, dizi uzunluğu olan N 'nin büyük olması durumunda DFT'nin doğrudan bulunması çok fazla miktarda işlem yapılmasını gerektirir. Yani, N sayısı artarken yapılan işlem sayısı yüksek hızla artmakta ve işlem sayısı kabul edilemez bir seviyeye doğru gitmektedir. 1965 yılında Cooley ve Tukey Ayrık Fourier Dönüşümü için gerekli işlem miktarını azaltacak bir prosedür geliştirdiler¹. Bu prosedür, sayısal işaret işleme ve diğer alanlarda DFT uygulamalarında ani bir artış olmasına sebep oldu. Ayrıca başka algoritmaların geliştirilmesine ön ayak olmuştur. Tüm bu algoritmalar **Hızlı Fourier Dönüşüm (FFT)** algoritmaları olarak bilinir. Bu algoritmalar ile DFT hesabı için yapılması gereken işlem sayısı büyük ölçüde azaltılarak işlem kolaylığı sağlanmıştır. Her ne kadar dönüşüm olarak adlandırılırsa, Hızlı Fourier Dönüşümü (FFT) Ayrık Fourier Dönüşümü (DFT)'den farklı değildir. FFT, DFT hesaplanması için etkili ve ekonomik bir algoritmadır.

Verimli Hesaplama :

Verimli bir şekilde tasarlanmış algoritmada işlem sayısı veri örneği başına sabit olmalıdır ve bundan dolayı toplam işlem sayısı N 'e bağlı olarak lineer olmalıdır. Genel olarak bir toplama işlemi için gereken işleme (processing) zamanı bir çarpma işlemi için gereken işleme zamanına göre çok daha azdır. Bu yüzden daha çok 4

¹ Cooley, J.W., Tukey, J.W., *An Algorithm for the Machine Computation of Complex Fourier Series*, Mathematics of Computation, Vol. 19, pp. 297-301, April 1965.

gerçel çarpma ve 2 gerçel toplama gerektiren kompleks çarpma işlemleri sayısını dikkate alacağız.

$\{W_N^{kn}\}$ faktörünün periyodiklik ve simetri özellikleri kullanılarak işlem sayısı azaltılabilir.

- Periyodiklik Özelliği

$$W_N^{kn} = W_N^{k(n+N)} = W_N^{(k+N)n}$$

- Simetri Özelliği

$$W_N^{kn+N/2} = -W_N^{kn}$$

Periyodiklik ve simetri özelliklerinin işlem sayısını azaltmadaki faydasını bir örnekle görelim.

Örnek 3.3:

Dört noktalı bir DFT'nin hesaplanması ve hesaplanması için verimli bir algoritma geliştirilmesi,

$$X(k) = \sum_{n=0}^3 x(n)W_4^{nk} \quad , \quad 0 \leq n \leq 3, \quad W_4 = e^{-j2\pi/4} = -j$$

Çözüm:

Yukarıdaki hesaplama 16 tane kompleks çarpma işlemi içeren matris formunda gösterilebilir:

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

Verimli Yaklaşım: Periyodikliği kullanarak,

$$W_4^0 = W_4^4 = 1 \quad ; \quad W_4^1 = W_4^5 = -j$$

$$W_4^2 = W_4^6 = -1 \quad ; \quad W_4^3 = j$$

bu eşitlikleri yukarıdaki matris formunda yerine koyarsak aşağıdaki ifadeyi elde ederiz.

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

Buradan_aşağıdaki denklemleri elde edebiliriz:

$$X(0) = x(0) + x(1) + x(2) + x(3) = \underbrace{[x(0) + x(2)]}_{g_1} + \underbrace{[x(1) + x(3)]}_{g_2}$$

$$X(1) = x(0) - jx(1) - x(2) + jx(3) = \underbrace{[x(0) - x(2)]}_{h_1} - j \underbrace{[x(1) - x(3)]}_{h_2}$$

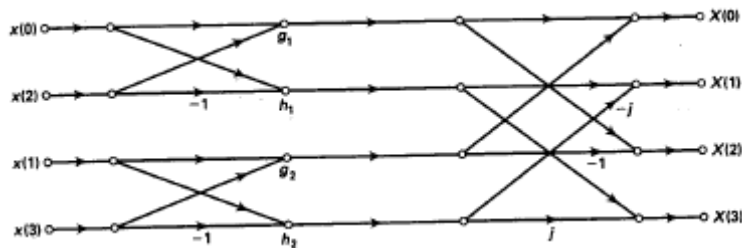
$$X(2) = x(0) - x(1) + x(2) - x(3) = \underbrace{[x(0) + x(2)]}_{g_1} - \underbrace{[x(1) + x(3)]}_{g_2}$$

$$X(3) = x(0) + jx(1) - x(2) - jx(3) = \underbrace{[x(0) - x(2)]}_{h_1} + j \underbrace{[x(1) - x(3)]}_{h_2}$$

Dolayısıyla verimli bir algoritma,

<i>Basamak 1</i>		<i>Basamak 2</i>
$g_1 = x(0) + x(2)$		$X(0) = g_1 + g_2$
$g_2 = x(1) + x(3)$		$X(1) = h_1 + jh_2$
$h_1 = x(0) - x(2)$		$X(2) = g_1 - g_2$
$h_2 = x(1) - x(3)$		$X(3) = h_1 + jh_2$

Bu algoritmaya göre yapılacak işlem sadece 2 kompleks çarpımayı içerir. Bu algoritmanın yapısı Şekil 3.5'deki gibidir.



Şekil 3.5: Örnek 3.3 için tasarlanmış algoritma yapısı

Hızlı Fourier Dönüşümü (FFT) Algoritmaları

DFT'nin hesaplanmasında birçok farklı algoritma geliştirilmiştir. Bu algoritmaların işlem sayısını azaltmak için temelde kullandığı özellikler temel fonksiyon olan W_N 'in periyodiklik ve simetri özelliğidir. Burada *Parçala-Birleştir yaklaşımını* (divide-combine approach) kullanarak türetilen radix-R FFT algoritmalarından radix-2 (yani data boyu $N = 2^k$ olan veriler için) algoritması olan Zamanda Desimasyonlu FFT (decimation-in-time FFT) ve Frekansda Desimasyonlu FFT (decimation-in-frequency FFT) algoritmalarını inceleyeceğiz.

Zamanda Desimasyonlu FFT

İşaretin uzunluğu olan N sayısı FFT algoritmalarında önemli rol oynamaktadır. N 'nin ikinin katları olması durumunda algoritma basit ve etkin olmaktadır. $N = 2^R$ olmakla birlikte bu algoritmaya Radix-2 FFT algoritması denir. N çift tamsayı olup $x(n)$ dizisi $N/2$ uzunluklu iki diziye ayrılabilir. Bu ayırmda, ilk dizinin elemanları tek sayı indisli, ikinci dizinin elemanları çift sayı indisli seçilir ve $n = 2r$ çift indisliler için, $n = 2r + 1$ tek indisliler için kullanılır. Bu ayırım aşağıda gösterilmiştir;

$$\begin{aligned} X_k &= \sum_{n \text{ çift}} x(n)W_N^{nk} + \sum_{n \text{ tek}} x(n)W_N^{nk} \\ &= \sum_{r=0}^{(N/2)-1} x(2r)W_N^{2rk} + \sum_{r=0}^{(N/2)-1} x(2r+1)W_N^{(2r+1)k} \\ k &= 0, 1, 2, \dots, N-1 \end{aligned} \quad (3.30)$$

$W_N^2 = W_{N/2}$ ifadesi kullanılarak aşağıdaki şekle getirilir;

$$\begin{aligned} X_k &= \sum_{r=0}^{(N/2)-1} x(2r)W_{N/2}^{rk} + W_N^k \sum_{r=0}^{(N/2)-1} x(2r+1)W_{N/2}^{rk} \\ &= AFD[N/2 \text{ çift noktalar}] + W_N^k AFD[N/2 \text{ tek noktalar}] \end{aligned} \quad (3.31)$$

Burada $N/2$ uzunluğunda iki DFT yardımıyla N uzunluklu dizinin DFT'si hesaplanmaktadır. $x(n)$ dizisinin tek ve çift noktaları;

$$\begin{aligned} x^c(n) &= x(2n); n = 0, 1, 2, \dots, (N/2) - 1 \\ x^t(n) &= x(2n + 1); n = 0, 1, 2, \dots, (N/2) - 1 \end{aligned} \quad (3.32)$$

şeklinde gösterelim. (3.32)'deki tanımlar (3.31)'de yerine konularak

$$X_k = \sum_{n=0}^{(N/2)-1} x^c(n)W_{N/2}^{nk} + W_N^k \sum_{n=0}^{(N/2)-1} x^t(n)W_{N/2}^{nk} \quad (3.33)$$

bulunur. Bu bağlantıda $X_{k+(N/2)}$ hesaplanırsa

$$X_{k+(N/2)} = \sum_{n=0}^{(N/2)-1} x^c(n)W_{N/2}^{n(k+(N/2))} + W_N^{k+(N/2)} \sum_{n=0}^{(N/2)-1} x^t(n)W_{N/2}^{n(k+(N/2))} \quad (3.34)$$

ifade edilir.

$$\begin{aligned} W_N^{nN/2} &= \left[e^{-j2\pi/(N/2)} \right]^{(nN/2)} = e^{-j2\pi n} = 1 \\ W_{N/2}^{N/2} &= 1 \end{aligned}$$

ifadeleri kullanarak,

$$X_{k+(N/2)} = \sum_{n=0}^{(N/2)-1} x^{\zeta}(n)W_{N/2}^{nk} - W_N^k \sum_{n=0}^{(N/2)-1} x^t(n)W_{N/2}^{nk} \quad (3.35)$$

elde edilir. k indeksini $0 \leq k \leq (N/2)-1$ aralığında sınırlayarak çift ve tek noktaların DFT'leri ayrı ayrı belirtilebilir:

$$\begin{aligned} X_k^{\zeta} &= \sum_{n=0}^{(N/2)-1} x^{\zeta}(n)W_{N/2}^{nk} \\ X_k^t &= \sum_{n=0}^{(N/2)-1} x^t(n)W_{N/2}^{nk} \end{aligned} \quad (3.36)$$

Bu tanımlardan yararlanarak (3.33) ve (3.35) bağıntıları tekrar yazılabilir:

$$\begin{aligned} X_k &= X_k^{\zeta} + W_N^k X_k^t \\ X_{k+(N/2)} &= X_k^{\zeta} - W_N^k X_k^t \\ k &= 0, 1, 2, \dots, (N/2)-1 \end{aligned}$$

N -noktalı DFT'nin tek ve çift noktalarının oluşturduğu $N/2$ noktalı iki DFT'den elde edilebileceği görülmektedir. Aynı şekilde, $N/2$ noktalı iki DFT de yeniden belirlenecek $N/4$ noktalı tek ve çift noktalı dizilerden benzer biçimde elde edilir. $N = 2^L$ varsayıldığında L adım gidilecek olursa, sonuçta sadece 2 noktalı bir dizinin DFT'sinin hesabı yeterli olmaktadır. Takip eden örnekte bu durum ayrıntılı bir şekilde açıklanmaktadır.

Örnek 3.4:

Zamanda desimasyonlu FFT algoritmasını açıklayabilmek amacıyla sekiz noktalı bir dönüşümü ele alalım. Buna göre, $N=8$ ve $N=2^L$ den $L=3$ olmaktadır. Şekil 3.6, sekiz noktalı DFT'nin iki adet dört noktalı DFT'ye ayrıştırılmasını göstermektedir. Bu ayrıştırma işlemine devam ederek dört noktalı DFT'ler de iki noktalı DFT'lere Şekil 3.7'de gösterildiği gibi ayrıştırılabilir. En son aşamada $N/2$ adet iki noktalı DFT elde edilecektir. DFT tanımından, bu işlem şu şekilde basitleşecektir

$$\begin{aligned} X(0) &= x(0) + x(1) \\ X(1) &= x(0) - x(1) \end{aligned}$$

Şekil 3.8, sekiz noktalı zamanda desimasyonlu FFT'nin akış diagramını göstermektedir. Bu akış diagramı üzerinden üç adım ile (bkz. Şekil 3.9) $x(n), 0 \leq n \leq 7$ dizisinin FFT'si şu şekilde hesaplanabilir:

Adım 1

$$x'(0) = x(0) + W_N^0 x(4) = x(0) + x(4)$$

$$x'(4) = x(0) + W_N^4 x(4) = x(0) - x(4)$$

$$x'(2) = x(2) + W_N^0 x(6) = x(2) + x(6)$$

$$x'(6) = x(2) + W_N^4 x(6) = x(2) - x(6)$$

$$x'(1) = x(1) + W_N^0 x(5) = x(1) + x(5)$$

$$x'(5) = x(1) + W_N^4 x(5) = x(1) - x(5)$$

$$x'(3) = x(3) + W_N^0 x(7) = x(3) + x(7)$$

$$x'(7) = x(3) + W_N^4 x(7) = x(3) - x(7)$$

Adım 2

$$x''(0) = x'(0) + W_N^0 x'(2) = x'(0) + x'(2) = x(0) + x(4) + x(2) + x(6)$$

$$x''(4) = x'(4) + W_N^2 x'(6) = x(0) - x(4) + W_N^2 \{x(2) + x(6)\}$$

$$x''(2) = x'(0) + W_N^4 x'(2) = x'(0) - x'(2) = x(0) + x(4) - \{x(2) + x(6)\}$$

$$x''(6) = x'(4) + W_N^6 x'(6) = x(0) - x(4) + W_N^6 \{x(2) + x(6)\}$$

$$x''(1) = x'(1) + W_N^0 x'(3) = x'(1) + x'(3) = x(1) + x(5) + x(3) + x(7)$$

$$x''(5) = x'(5) + W_N^2 x'(7) = x(1) - x(5) + W_N^2 \{x(3) - x(7)\}$$

$$x''(3) = x'(1) + W_N^4 x'(3) = x'(1) - x'(3) = x(1) + x(5) - \{x(3) + x(7)\}$$

$$x''(7) = x'(5) + W_N^6 x'(7) = x(1) - x(5) + W_N^6 \{x(3) - x(7)\}$$

Adım 3

$$X(0) = x''(0) + W_N^0 x''(1) = x''(0) + x''(1)$$

$$X(1) = x''(4) + W_N^1 x''(5)$$

$$X(2) = x''(2) + W_N^2 x''(3)$$

$$X(3) = x''(6) + W_N^3 x''(7)$$

$$X(4) = x''(0) + W_N^4 x''(1) = x''(0) - x''(1)$$

$$X(5) = x''(4) + W_N^5 x''(5)$$

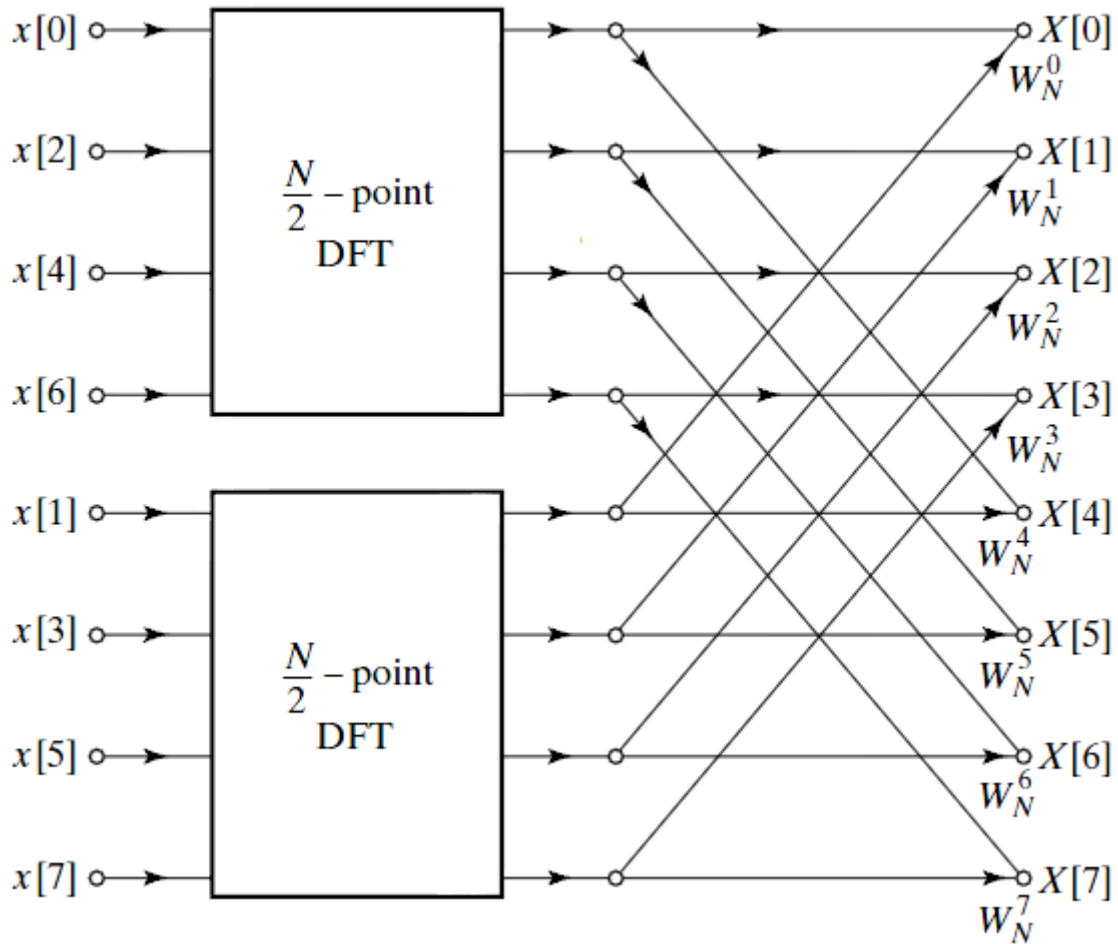
$$X(6) = x''(2) + W_N^6 x''(3)$$

$$X(7) = x''(6) + W_N^7 x''(7)$$

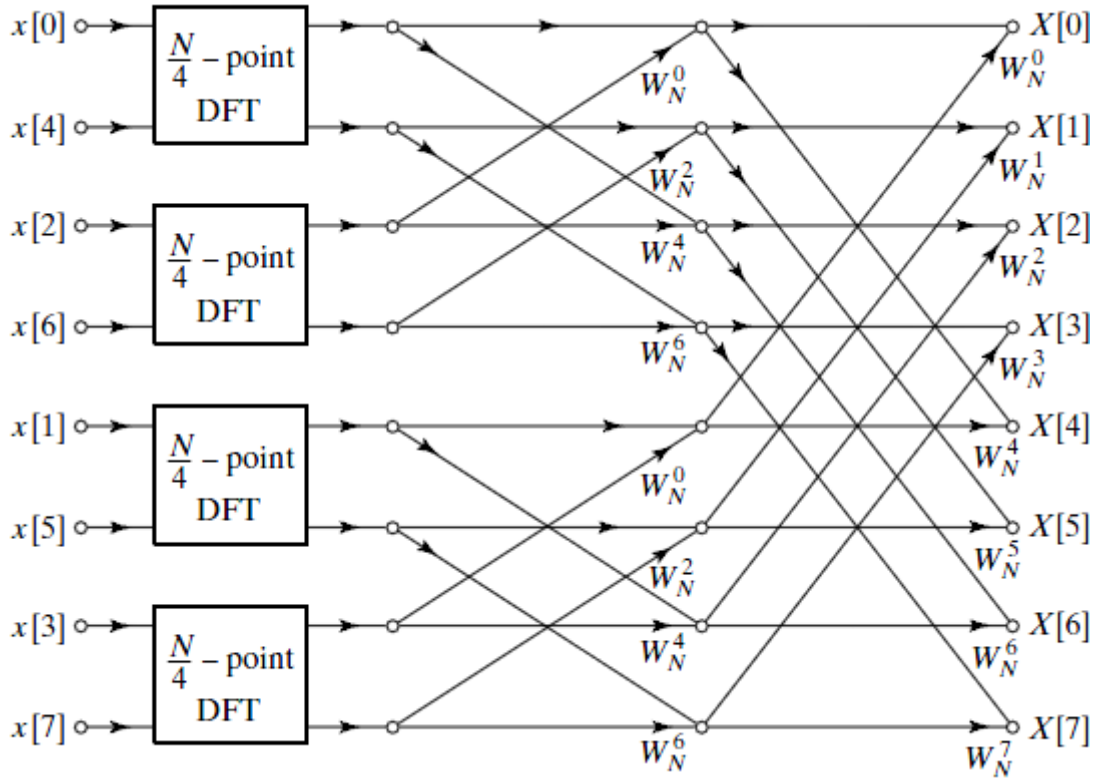
Son olarak 2. adımda bulunan değerlere 3. adımda yerlerine konulursa,

$$\begin{aligned}
X(0) &= x(0) + x(4) + x(2) + x(6) + x(1) + x(5) + x(3) + x(7) \\
X(1) &= x(0) - x(4) + W_N^2 \{x(2) + x(6)\} + W_N^1 \{x(1) - x(5) + W_N^2 \{x(3) - x(7)\}\} \\
X(2) &= x(0) + x(4) - \{x(2) + x(6)\} + W_N^2 \{x(1) + x(5) - \{x(3) + x(7)\}\} \\
X(3) &= x(0) - x(4) + W_N^6 \{x(2) + x(6)\} + W_N^3 \{x(1) - x(5) + W_N^6 \{x(3) - x(7)\}\} \\
X(4) &= x(0) + x(4) + x(2) + x(6) - \{x(1) + x(5) + x(3) + x(7)\} \\
X(5) &= x(0) - x(4) + W_N^2 \{x(2) + x(6)\} + W_N^5 \{x(1) - x(5) + W_N^2 \{x(3) - x(7)\}\} \\
X(6) &= x(0) + x(4) - \{x(2) + x(6)\} + W_N^6 \{x(1) + x(5) - \{x(3) + x(7)\}\} \\
X(7) &= x(0) - x(4) + W_N^6 \{x(2) + x(6)\} + W_N^7 \{x(1) - x(5) + W_N^6 \{x(3) - x(7)\}\}
\end{aligned}$$

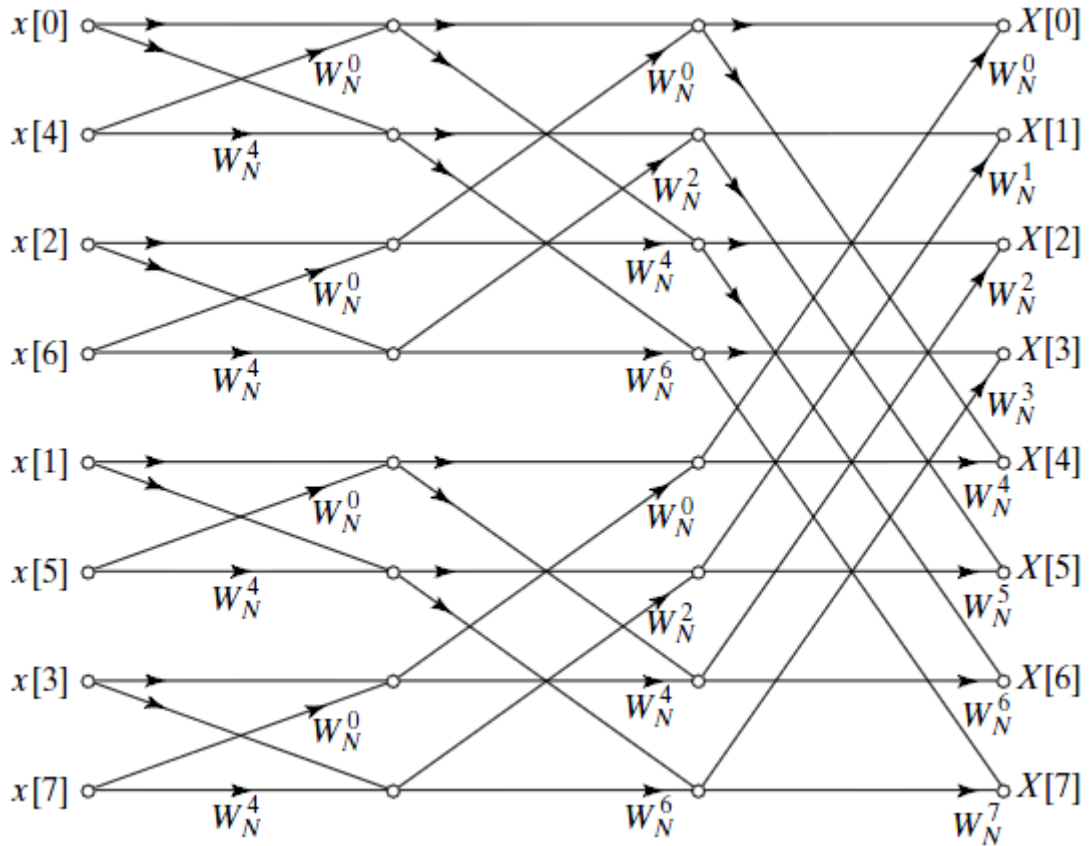
elde edilmiş olur.



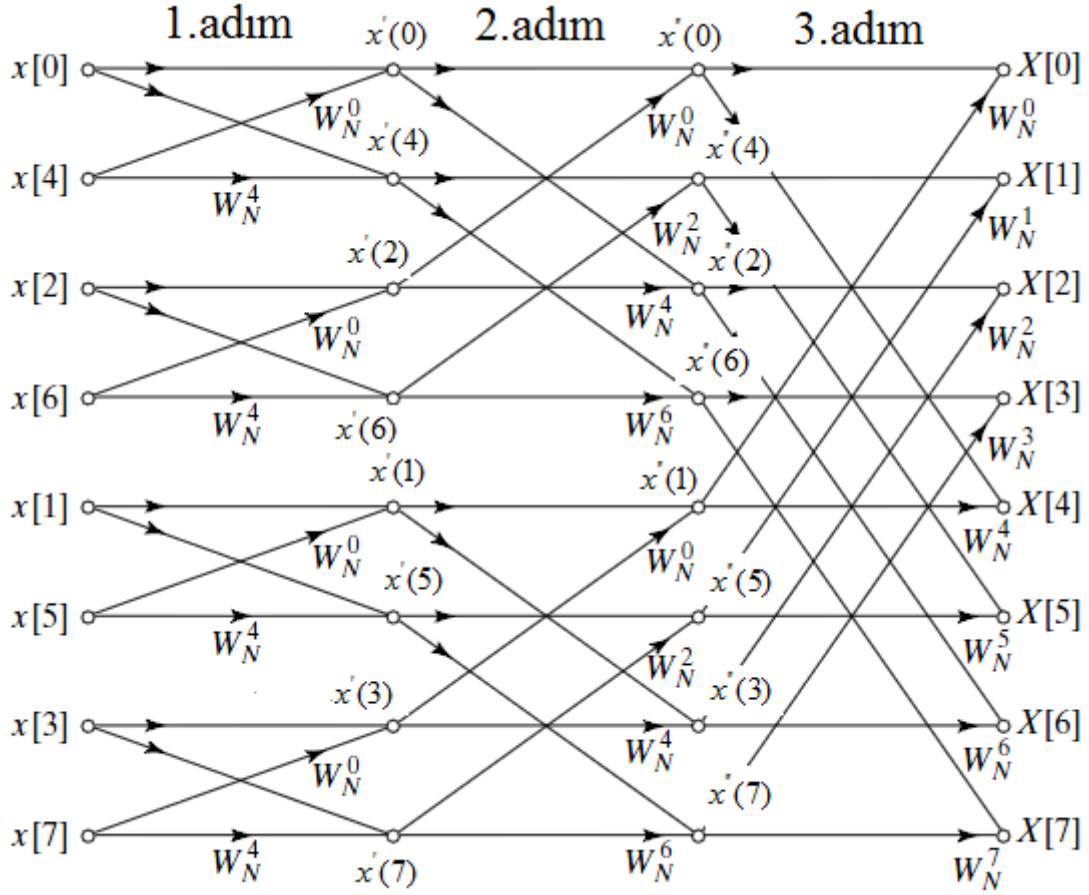
Şekil 3.6: Sekiz noktalı DFT'nin zamanda desimasyon ile iki adet dört noktalı DFT'ye ayrıştırılması



Şekil 3.7: İki adet dört noktalı DFT'nin zamanda desimasyon ile dört adet iki noktalı DFT'ye ayrıştırılması



Şekil 3.8: Sekiz noktalı zamanda desimasyonlu FFT için akış diagramı



Şekil 3.9: Sekiz noktalı zamanda desimasyonlu FFT için akış diagramı

Frekansta Desimasyonlu FFT

Alternatif FFT algoritması diziyi orta noktasından ikiye ayırarak ve aynı işlem uygulanarak geliştirilebilir.

$$\begin{aligned}
 x^{(1)}(n) &= x(n) \\
 x^{(2)}(n) &= x(n + N/2) \\
 n &= 0, 1, 2, \dots, (N/2) - 1
 \end{aligned}$$

biçiminde ikiye ayrılısın.

$$\begin{aligned}
 X_k &= \sum_{n=0}^{N-1} x(n) W_N^{nk} \\
 &= \sum_{n=0}^{(N/2)-1} x(n) W_N^{nk} + \sum_{n=N/2}^{N-1} x(n) W_N^{nk} \\
 &= \sum_{n=0}^{(N/2)-1} x(n) W_N^{nk} + \sum_{n=0}^{(N/2)-1} x(n + N/2) W_N^{(n+(N/2))k}
 \end{aligned} \tag{3.37}$$

şeklinde yazılabilir. Tanımlamalar yerlerine konursa,

$$X_k = \sum_{n=0}^{(N/2)-1} x^{(1)}(n)W_N^{nk} + \sum_{n=0}^{(N/2)-1} x^{(2)}(n)W_N^{(n+(N/2))k} \quad (3.38)$$

elde edilir.

Frekans desimasyonunda çift ve tek frekans domeni noktaları X_{2k} ve X_{2k+1} $k=0,1,\dots,(N/2)-1$ aralığında kullanılarak DFT elde edilebilir. k yerine $2k$ koyarsak,

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

yazılabilir.

$W_N^{2nk} = W_{N/2}^{nk}$ ve $W_N^{kN} = 1$ özelliklerinden yararlanılarak,

$$X_{2k} = \sum_{n=0}^{(N/2)-1} x^{(1)}(n)W_{N/2}^{nk} + \sum_{n=0}^{(N/2)-1} x^{(2)}(n)W_{N/2}^{nk} \quad (3.39)$$

elde edilir.

k yerine $2k+1$ konarak ve benzer şekilde

$$\begin{aligned} X_{2k+1} &= \sum_{n=0}^{(N/2)-1} x^{(1)}(n)W_N^{n(2k+1)} + \sum_{n=0}^{(N/2)-1} x^{(2)}(n)W_N^{(n+(N/2))(2k+1)} \\ &= \sum_{n=0}^{(N/2)-1} W_N^n x^{(1)}(n)W_{N/2}^{nk} - \sum_{n=0}^{(N/2)-1} W_N^n x^{(2)}(n)W_{N/2}^{nk} W_N^{Nk} W_N^{N/2} \end{aligned}$$

elde edilir.

$W_N^{Nk} = 1$ ve $W_N^{N/2} = -1$ özelliklerinden yararlanarak,

$$X_{2k+1} = \sum_{n=0}^{(N/2)-1} W_N^n x^{(1)}(n)W_{N/2}^{nk} - \sum_{n=0}^{(N/2)-1} W_N^n x^{(2)}(n)W_{N/2}^{nk} \quad (3.40)$$

elde edilir.

$k=0,1,\dots,(N/2)-1$ için X_{2k} ve X_{2k+1} 'den N için frekansta desimasyonlu FFT görülmektedir:

$$\begin{aligned} X_{2k} &= \sum_{n=0}^{(N/2)-1} [x^{(1)}(n) + x^{(2)}(n)] W_{N/2}^{nk} \\ X_{2k+1} &= \sum_{n=0}^{(N/2)-1} [x^{(1)}(n) - x^{(2)}(n)] W_N^n W_{N/2}^{nk} \end{aligned} \quad (3.41)$$

Hesaplama Sayısı Karşılaştırılması

DFT hesaplanmasında N^2 karmaşık çarpma ve $N(N-1)$ karmaşık toplama gereklidir. FFT yardımıyla $N = 2^R$ noktadan oluşan dizinin DFT'sinin hesaplanmasında $NR/2$ karmaşık çarpma ve NR karmaşık toplama işlemi yeterlidir. Adım sayısı $R = \log_2 N$ olarak yazılırsa işlem yoğunluğu açısından DFT ile FFT kıyaslanabilir.

Matlab Uygulamaları

Örnek 3.5: DFT ile FFT arasındaki işlem süresinin kıyaslanması.

TIC ve TOC fonksiyonu kullanılarak yapılabilir. TIC fonksiyonu zamanlayıcıyı başlatır TOC fonksiyonunda zamanlayıcı bilgisini okur.

```
N=2048;
x=rand(1,N);
tic
fft(x,N);
toc

tic
radix2hfd(x,N);
toc

tic
n=[0:1:N-1];
k=[0:1:N-1];
WN=exp(-j*2*pi/N);
nk=n*k;
WNnk=WN.^nk;
Xk=x*WNnk;
toc

elapsed_time =
    0.0940

elapsed_time =
    56.172000

elapsed_time =
    17.8750
```

Örnek 3.6: Zamanda desimasyonlu FFT algoritmasının incelenmesi.

```
function[X]=radix2hfd(x,M)
if nargin ==2
    N=M;
else
    N=length(x);
end
%global M;
N=length(x);
L=log2(N);
if L==1
    X=[x(1)+x(2) x(1)-x(2)]
end
if L>1
    WN=exp(-j*2*pi/N);
    W(1,:)=WN.^([0:1:N/2-1]);
    x0(1,:)=x(1:2:end);
    x1(1,:)=x(2:2:end);
    X0(1,:)= radix2hfd(x0)+W.* radix2hfd(x1);
    X1(1,:)= radix2hfd(x0)-W.* radix2hfd(x1);
    X=[X0 X1];
end
```

Ödev 3.2:

1) $N=8$ için frekansda desimasyonlu Hızlı Fourier Dönüşümünün akış diyagramını çizin ve sekiz noktalı FFT'yi Örnek 3.4'dekine benzer şekilde üç adımda hesaplayınız.

2) Matlab'da fft komutu kullanmadan frekans desimasyonlu FFT algoritmasını üretiniz.