

Ters DFT

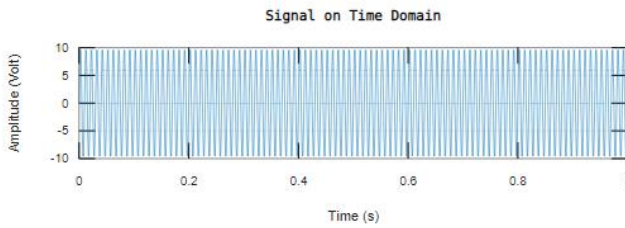
Ad Soyad: Reşat KARAKAYA **No:** 348284 **Öğretim:** 1.Öğretim

- a) 100 Hz'lik Sinüs sinyalinin Matlab ortamında oluşturulması, Fourier Dönüşümünün ve Ters Fourier Dönüşümünün gerçekleştirilmesi.

```
1 Fs = 1000; % frequency
2 t1 = 0:1/Fs:1-1/Fs;
3 fm = 100; % hertz
4
5 y = 10 * sin(2*pi*fm*t1);
6
7
8 subplot(3,1,1);
9 plot(t1,y);
10 title('Signal on Time Domain')
11 xlabel('Time (s)')
12 ylabel('Amplitude (Volt)')
13 grid on
14
15 dft = fft(y);
16 t2 = 0:1/Fs:1-1/Fs;
17 subplot(3,1,2);
18 mg = abs(dft);
19 plot(t2(1:500),mg(1:500));
20 title('Signal on Frequency Domain (DFT)')
21 xlabel('Frequency (Hz)')
22 ylabel('Normalized Amplitude')
23 grid on
24
25 idft = ifft(dft);
26 t3 = 0:1/Fs:1-1/Fs;
27 subplot(3,1,3);
28 plot(t3,abs(idft));
29 title('IDFT')
30 xlabel('Frequency (Hz)')
31 ylabel('Amplitude')
32 grid on
```

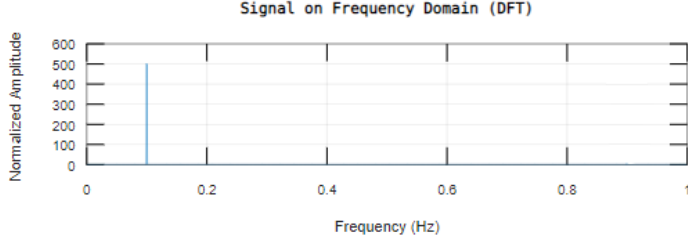
ifft fonksiyonu ile Kod

Matlab ortamında 100 Hz'lik Sinüs sinyalinin çizdirmek için ilk önce F_s ile bir Frekans tanımlaması yapılıyor. Frekans'a 1000 değeri atanıyor. Daha sonra t_1 ile zaman vektörü tanımlaması yapılıyor. Burada frekans değeri de kullanılıyor. f_m ile işlem yapılacak hertz değeri tanımlanıyor. Bu işlemler yapıldıktan sonra y değişkeniyle çizilecek sinüs fonksiyonu ataması yapılıyor. Subplot ile birden fazla grafik çizdirmeye hizalama yapılması sağlanır. Burada subplot(3,1,1) kullanılarak 3 satır 1 sütun olduğu belirtildi ve 1.satırda ilk grafik olan Sinüs Sinyalinin çizilmesi sağlandı. Plot ile de çizilecek grafiğin parametreleri yazılarak ekranda çizilmesi sağlanıyor. Burada ilk parametresine zaman vektörü olan t_1 'yi, ikinci parametresine ise çizilecek sinüs fonksiyonu olan y değerleri yazılıyor.



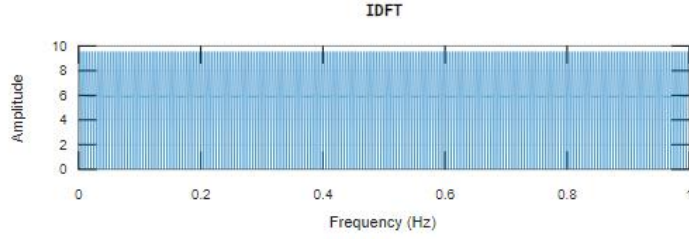
100 Hz'lik Sinüs Sinyalinin Çizdirilmesi

$plot(t,y)$ ile sinüs sinyalinin elde ettim. Fourier Dönüşümü için ise bir dft değişkeni tanımladım ve fft fonksiyonunu atayıp parametre olarak da y değişkenini yolladım. Daha sonra t_2 ile zaman vektörü tanımlaması yaptım. Subplot ile grafik çizdirmek için hizalama yaptım. mg değişkeni Fourier Dönüşümü için abs fonksiyonu ile birlikte parametre olarak dft almasıyla daha net sonuç alınmasını sağlar. $plot$ ile grafiği çizdirme işlemi yaptım. $Plot$ 'un ilk parametresine zaman vektörü olan t_2 'yi sınırlandırarak yolladım ve ikinci parametresine Fourier Dönüşümü için mg değişkenini yine aynı şekilde sınırlandırarak yolladım. Bu şekilde Fourier Dönüşümü elde edildiği zaman sadece sinyal sol tarafta çıkacak. Eğer t_2 ve mg değişkenlerini sınırlandırarak yollamasaydım Fourier Dönüşümü simetrik olduğu için hem sol hem sağ tarafta sinyal elde edilecekti.



Fourier Dönüşümünün Elde Edilmesi

Fourier Dönüşümünü tek bir sinyal olarak elde ettikten sonra Ters Fourier Dönüşümünü elde etmek için *idft* değişkeni tanımladım ve *ifft* fonksiyonunu atayıp parametre olarak da *dft* değişkenini yolladım. Yine *t3* zaman vektörü oluşturdum ve subplot ile hizalama yaptım. Plot'un ilk parametresine zaman vektörü olan *t3*, ikinci parametresine Ters Fourier Dönüşümü için *idft* değişkenini *abs* fonksiyonu ile birlikte yolladım.

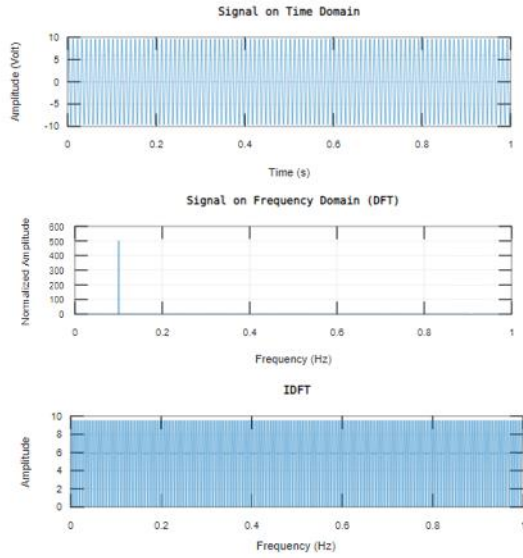


Ters Fourier Dönüşümünün Elde Edilmesi

Buraya kadar olan kısımda Ters Fourier Dönüşümü Matlab'da bulunan fonksiyon sayesinde gerçekleştirdim. Ders sunumunda verilen formülü kullanarak Ters Fourier Dönüşümü gerçekleştirdiğim zaman ise kod aşağıdaki gibi oluyor ve aynı çıktıları veriyor.

```
1 Fs = 1000; % frequency
2 t1 = 0:1/Fs:1-1/Fs;
3 fm = 100; % hertz
4
5 y = 10 * sin(2*pi*fm*t1);
6
7 dft = fft(y);
8 mg = abs(dft);
9 re = real(dft);
10 re = re / 500;
11 im = -imag(dft);
12 im = im / 500;
13
14 idft = zeros(1,Fs);
15 i=1;
16 while(i <= Fs)
17     k=1;
18     real = 0;
19     imag = 0;
20     while(k <= Fs/2)
21         real = real + re(k) * cos(2*pi*k*i ./ Fs);
22         imag = imag + im(k) * sin(2*pi*k*i ./ Fs);
23         k=k+1;
24     end
25     idft(i) = imag + real;
26     i= i+1;
27 end
28
29 subplot(3,1,1);
30 plot(t1, y);
31 title('Signal on Time Domain')
32 xlabel('Time (s)')
33 ylabel('Amplitude (Volt)')
34 grid on
35
36 t2 = 0:1/Fs:1-1/Fs;
37 subplot(3,1,2);
38 plot(t2(1:500), mg(1:500));
39 title('Signal on Frequency Domain (DFT)')
40 xlabel('Frequency (Hz)')
41 ylabel('Normalized Amplitude')
42 grid on
43
44 t3 = 0:1/Fs:1-1/Fs;
45 subplot(3,1,3);
46 plot(t3, idft);
47 title('IDFT')
48 xlabel('Frequency (Hz)')
49 ylabel('Amplitude')
50 grid on
```

Formül Kullanılan Kod



Formül Kullanılan Kod Grafikleri

Burada sinüs sinyalinin oluşması ve Fourier Dönüşüm aynı şekilde gerçekleşiyor ama Ters Fourier Dönüşüm ders sunumunda verilen formül ile gerçekleşiyor. İlk önce Ters Fourier Dönüşümü için idft değişkeni oluşturdum. Formülde toplam sembolü olduğu için bir while döngüsü oluşturdum. Real ve image kısımları için değişkenler oluşturdum ve döngü her başa döndüğünde 0 değeri atadım. Burada bir while döngüsü daha oluşturdum ve formülde olan $N/2$ yerine $F_s/2$ 'ye kadar devam ediyor. İçteki while döngüsünden çıktığı zaman real ve image kısımları toplanıp idft değişkenine atanıyor ve dıştaki while döngüsü başa dönüyor. Bu şekilde while döngüsü sona erdiği zaman idft değişkeni Ters Fourier Dönüşümünü oluşturuyor. `plot(t3,idft)` ile de grafiği ekranda çizdiriyor.

- b)** 100 Hz'lik ve 150 Hz'lik Sinüs sinyallerinin çarpılıp çarpılmış sinyallerin Matlab ortamında oluşturulması, Fourier Dönüşümünün ve Ters Fourier Dönüşümünün gerçekleştirilmesi.

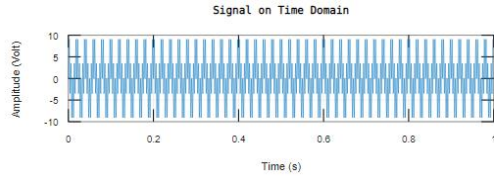
```

1  Fs = 500; % frequency
2  t1 = 0:1/Fs:1-1/Fs;
3  f1 = 100; % hertz1
4  f2 = 150; % hertz2
5
6  y1 = 10 * sin(2*pi*f1*t1);
7  y2 = 10 * sin(2*pi*f2*t1);
8  y3 = y1 .* y2;
9
10 subplot(3,1,1);
11 plot(t1,y3);
12 title('Signal on Time Domain')
13 xlabel('Time (s)')
14 ylabel('Amplitude (Volt)')
15 grid on
16
17 dft = fft(y3);
18 t2 = 0:1/Fs:1-1/Fs;
19 subplot(3,1,2);
20 mg = abs(dft);
21 plot(t2(1:250),mg(1:250));
22 title('Signal on Frequency Domain (DFT)')
23 xlabel('Frequency (Hz)')
24 ylabel('Normalized Amplitude')
25 grid on
26
27 idft = ifft(dft);
28 t3 = 0:1/Fs:1-1/Fs;
29 subplot(3,1,3);
30 plot(t3,abs(idft));
31 title('IDFT')
32 xlabel('Frequency (Hz)')
33 ylabel('Amplitude')
34 grid on

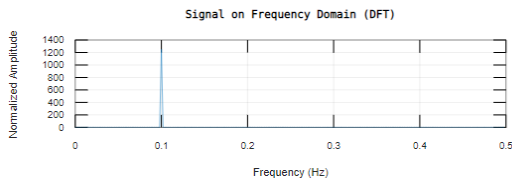
```

ifft fonksiyonu ile Kod

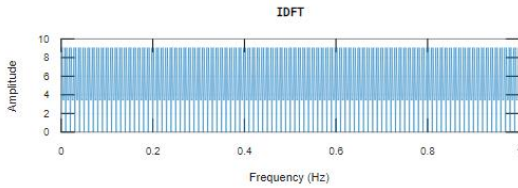
Burada ilk kodda olduğu gibi ilk önce değişken tanımlamaları yaptım. İki sinüs sinyali çarpıldığı için f1 ve f2 olarak işlem yapılacak Hertz değeri tanımladım ve sırasıyla 100 Hertz ve 150 Hertz değerlerini verdim. Yine iki tane Sinüs olduğu için y1 ve y2 değişkenleri tanımladım ve Sinüs sinyali oluşturdum. y1 ve y2 oluşturduktan sonra y3 değişkeni tanımladım. y1 ile y2 sinyallerini çarpıp y3 değerine atadım. Frekans değerini 500 verdim. Bunları yaptıktan sonra kalan işlemlerin mantığı ilk kodda olduğu gibi gerçekleşiyor.



100 Hz ve 150 Hz'lik Sinüs Sinyallerinin Çarpımından Oluşan Sinyal



Fourier Dönüşümünün Elde Edilmesi

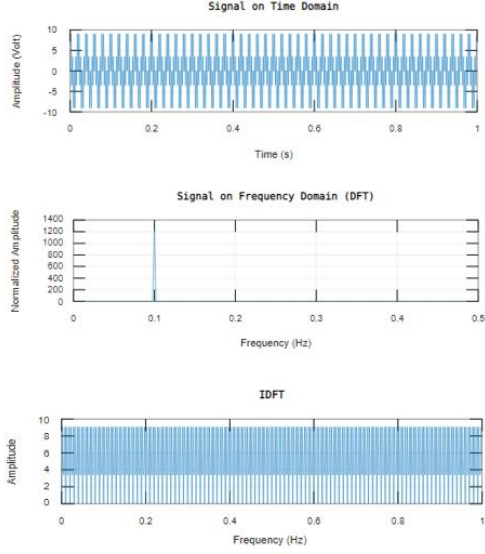


Ters Fourier Dönüşümünün Elde Edilmesi

Ders sunumunda verilen formülü kullanarak Ters Fourier Dönüşümü gerçekleştirdiğim zaman ise kod aşağıdaki gibi oluyor ve aynı çıktıları veriyor.

```
1 Fs = 500; % frequency
2 t1 = 0:1/Fs:1-1/Fs;
3 f1 = 100; % hertz1
4 f2 = 150; % hertz2
5
6 y1 = 10 * sin(2*pi*f1*t1);
7 y2 = 10 * sin(2*pi*f2*t1);
8 y3 = y1 .* y2;
9
10 dft = fft(y3);
11 mg = abs(dft);
12 re = real(dft);
13 re = re / 250;
14 im = - imag(dft);
15 im = im / 250;
16
17 idft = zeros(1,Fs);
18 i=1;
19 while(i <= Fs)
20     k=1;
21     real = 0;
22     imag = 0;
23     while(k <= Fs/2)
24         real = real + re(k) * cos(2*pi*k*i ./ Fs);
25         imag = imag + im(k) * sin(2*pi*k*i ./ Fs);
26         k=k+1;
27     end
28     idft(i) = imag + real;
29     i= i+1;
30 end
31
32 subplot(3,1,1);
33 plot(t1,y3);
34 title('Signal on Time Domain')
35 xlabel('Time (s)')
36 ylabel('Amplitude (Volt)')
37 grid on
38
39 t2 = 0:1/Fs:1-1/Fs;
40 subplot(3,1,2);
41 plot(t2(1:250),mg(1:250));
42 title('Signal on Frequency Domain (DFT)')
43 xlabel('Frequency (Hz)')
44 ylabel('Normalized Amplitude')
45 grid on
46
47 t3 = 0:1/Fs:1-1/Fs;
48 subplot(3,1,3);
49 plot(t3,idft);
50 title('IDFT')
51 xlabel('Frequency (Hz)')
52 ylabel('Amplitude')
53 grid on
```

Formül Kullanılan Kod



Formül Kullanılan Kod Grafikleri

Burada Ters Fourier Dönüşümü ders sunumunda verilen formül ile gerçekleştirdim. Sinüs sinyali oluşturma ve Fourier Dönüşümü aynı şekilde oluşturdum.

Sonuç Olarak sinüs sinyali oluşturdum ve bu sinüs sinyalinin Fourier Dönüşümünü ve Ters Fourier Dönüşümünü elde ettim. Ters Fourier Dönüşümünü iki şekilde elde edebiliyorum. Bunlardan birincisi Matlab'da bulunan fonksiyonu kullanarak, ikincisi ise ders sunumunda verilen formülü kullanarak elde ediliyor. Sinüs sinyalini çizdirdikten sonra Fourier Dönüşümünü elde ettim. Fourier Dönüşümünde sadece tek bir sinyal olmasını istiyorum ama ilk önce simetrik olduğu için hem sol hem sağ tarafta sinyal oluştu. Sadece sol tarafta sinyal oluşturmak için ise plot kısmında frekansın yarı değeriyle sınırlandırdım ve istediğim gibi tek bir sinyal oluştu. Ters Fourier Dönüşümü elde ettiğim zaman ise Fourier Dönüşümüne göre daha farklı bir sinyal oluştu. Oluşan bu sinyal ilk oluşan sinüs sinyaline daha yakın bir görüntü verdi. Çizdiğim ilk grafik olan 100 Hz'lik sinüs sinyalinde daha net bir sinüs sinyali elde ettim ama çizdiğim ikinci grafik olan 100 Hz ve 150 Hz'lik sinüs sinyallerinin çarpımından elde ettiğim sinüs sinyali daha dağınık bir sinyal verdi. Tek sinüs sinyalinden oluşturduğum Fourier Dönüşümü iki sinüs sinyalinin çarpımından elde ettiğim Fourier Dönüşümüne göre daha sabit bir halde olurken iki sinüs sinyalinin çarpımında elde ettiğim biraz daha dalgalı bir sinyal oluştu. Bu şekilde bir sinüs sinyalinin veya iki sinüs sinyalinin çarpımından oluşan sinyalin grafiğini elde ettim. Fourier ve Ters Fourier Dönüşümünün sinüs sinyalinden elde edildiği zaman ortaya çıkan sonucu gözlemlemiş oldum.