MUHAMMED HALİT TOKLUOĞLU 161110070

1. Soru Kod açıklaması:

$$x[n] = \frac{1}{T}\int_T x(t)e^{-jnw_0 t}, w_0 = 2\pi/T$$

Bu formülden x[n] leri bulmaya çalıştım.

Bu iş için yazdığım fonksiyon frr_series(t,y,Order,Period).

```
function r = frr_series(t,y,Order,Period)

r = [-Order,Order];
counter = 1;
for element = [-Order:Order]
    r(counter) = find_fourier_coef(t,y,element,Period);
    counter = counter + 1;
end
```

Period sinyalin periodu. Order ise kaç tane fourier hesaplaması yapılması gerektiğini belirten sayı örneğin 9 ise x[-9] dan x[9] kadar olan fourier katsayılarını hesaplayıcak. y fonksiyonumuzun t'ye göre verileri. Bu fonksiyonda yapılan olay bir counter atanıyor ve verilen Order sayısına göre x[n] ler find_fourier_coef fonksiyonu ile hesaplanıyor.

find_fourier_coeff(t,y,n,Period) fonksiyonu:

```
function frr = find_fourier_coef(t,y,n,Period)
w0 = 2*pi/Period;
new = t;
j = sqrt(-1);
interval = t(2)-t(1);
counter = 1;
for count=t
    new(counter) = y(counter) * exp(-j*n*w0*count);
    counter = counter + 1;
end
new_intgrl = intgrl(t,new);
T_init = 1;
T_end = T_init + round(Period/interval);
frr = (new_intgrl(T_end)-new_intgrl(T_init))/Period;
end
```

Bu fonksiyonda

$$x[n] = \frac{1}{T}\int_T x(t)e^{-jnw_0 t} \quad , w_0 = 2\pi/T$$

formülü kullanılıyor. Verilen n değerine göre fourier katsayısı hesaplanıyor. For döngüsü içinde $x(t)e^{-jnw_0 t}$ ifadesi verisel olarak hesaplanıyor. Daha sonra intgrl fonksiyonu ile bu veriler nümerik olarak integrasyon ediliyor. Sonra yeni integrasyon fonksiyonunun ilk verisi ile ilk verisinden bir period sonrasindaki veri farkı alınıp Period'a bölünüyor. Kısacası x[n], n sayısına göre bulunuyor. n = 3 için x[3] bulunuyor.

ingrl(t,data) fonksiyonu

```
function new = intgrl(t,data)
dt = t(2)-t(1);
new = [0:(length(t)-2)];
counter = 1;
for count = data
    if count ~= data(end)
        if counter>1
            new(counter) = dt*(data(counter)+ data(counter+1))/2 + new(counter-1);
            counter = counter + 1;
        else
            new(counter) = dt*(data(counter)+ data(counter+1))/2;
            counter = counter + 1;
        end
    end
end
end
```

Bu fonksiyonda veriler numerik olarak enterasyon yapılıyor. İlk her veri arasındaki zaman farkı olan dt bulunuyor. Sonra for döngüsünde bir önceki değer ile şimdiki değerin ortalaması alınıp dt ile çarpılıyor ve elde edilen sonuç bir önceki elde edilen sonuç ile toplanıyor. If else'li yapı ise bir counter 1 iken new(0) olamayacağından oradaki oluşacak olan hatayı düzeltmek için yapılmıştır. Bu fonksiyon gelen herhangi bir verinin integralini alıp geri yolluyor. Ancak veri vektöründe 1 azalmaya sebep oluyor. Örneğin 1001 veriyi alıp entegre edip 1000 veri olarak geri yolluyor. Bu soru için bu bir engel teşkil etmediğinden bunun için ayrı bir yapı yazılmamıştır.

find_xt(xn,t,Order,Period) fonksiyonu

```
function xt = find_xt(xn,t,Order,Period)
j = sqrt(-1);
w0 = 2*pi/Period;
xt = 0;
counter = 1;
for n=[-Order:Order]
    xt = xt + xn(counter)*exp(j*w0*n*t);
    counter = counter + 1;
end
end
```

Bu fonksiyon

$$x(t) = \sum_{n=-Order}^{n=Order} x[n]e^{jnw_0 t} \quad , w_0 = 2\pi/T$$

işlemini yaparak verilen Order değerine göre fourier verilerilerini veriyor. For döngüsünün içinde toplama işlemi gerçekleşiyor. x(t) t'ye göre bulunuyor.

rect(Period) fonksiyonu:

```
function [t, y] = rect(Period)
inc = Period / 5000;
t = [-1.5*Period:inc:1.5*Period];
y = t;
inr = Period*0.25/inc;
counter = 1;
state = 0;
for e=y
    if counter == round(inr*1) || counter == round(5*inr) || counter == round(9*inr)
        state = 1;
    elseif counter == round(3*inr) || counter == round(7*inr) || counter == round(11*inr)
        state = 0;
    end
    if state == 0
        y(counter) = 0;
    else
        y(counter) = 1;
    end
    counter = counter + 1;
end
end
```

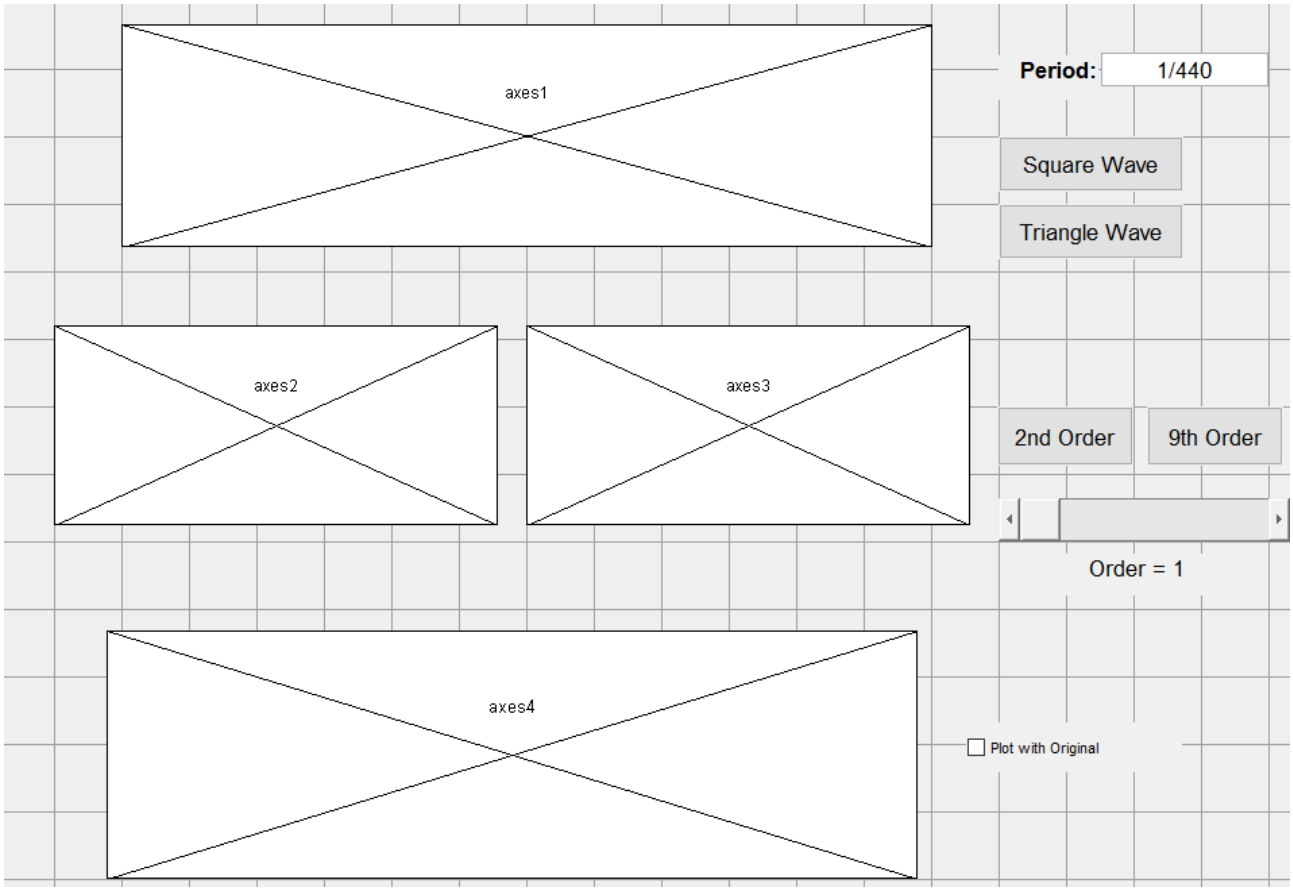
Bu fonksiyon Period değerini alıyor 5000'e bölüp arttırma değeri olan inc'i hesaplıyor. Sonra 3 periodu kapsayacak ve 0 noktasına göre simetrik olacak t eksenini çizdiriyor. Aynı sayıda elemana sahip olması için y t'ye eşitleniyor. For döngüsü içinde (verimiz bir rectengle olduğundan ve 1 ve 0 değerlerine sahip olduğundan) 2 adet state bulunmaktadır. Counter'ımız 1 den başlayıp y'nin son değerine kadar artıyor. Ve ilk if'li kısımda state'ler belirleniyor. Rectangle fonksiyonu ilk 0.25 Periodu boyunca 0 sonra 0.5 Periodu boyunca 1 sonra geri kalan 0.25 Periodu boyunca sıfırdır. Period ile 0.25 sayısı çarpılıp artış değerine bölündüğünde verisel olarak 0.25 Periodların hangi noktada olduğunu buluruz. Bu değer inr'de barınıyor. Ve inr ile counter ilk if'lerde kıyaslanıyor. Burada round fonksiyonun kullanılma sebebi ise irrasyonel bir sayı örneğin π Period' değerinde varsa inr değeride bir irrasyonel sayı olacaktır ve counter integer bir sayı olduğundan asla o değere eşit olamayacaktır. Round fonksiyonu ile sayı irrasyonel olması durumunda bile integer'a çevirilerek bu kontrol gerçekleştirilebilir. Bu fonksiyonun geri döndürdüğü veriler ise y yani rectangle fonksiyonu t yani y'nin ekseni.

tri(Period) fonksiyonu

```
function [t,y] = tri(Period)
inc = Period/5000;
t = [-1.5*Period:inc:1.5*Period];
y = t;
inr = Period*0.5/inc;
dt = 1/inr;
counter = 1;
state = 0;
for e=y
    if counter == round(inr*1) || counter == round(3*inr) || counter == round(5*inr)
        state = 1;
    elseif counter == round(2*inr) || counter == round(4*inr) || counter == round(6*inr)
        state = 0;
    end
    if state == 0
        if counter == 1
            y(counter) = 0;
        else
            y(counter) = y(counter-1) + dt;
        end
    else
        y(counter) = y(counter-1) - dt;
    end
    counter = counter + 1;
end
end
```

Bu fonksiyonda yapılanlar rect fonksiyonuna benzemektedir. Burada triangle fonksiyonunun yapısından dolayı (0.5 Period boyunca yükselişte olması ve 0.5 Period boyunca düşüşte olması) inr değeri 0.5 ile çarpılmıştır. Bu fonksiyon içinde 2 state vardır yükseliş ve düşüş. 1/inr değeri ise bize artışı verecektir. Çünkü unit triangle'ın Magnitude değeri 1 dir. Bu yüzden 1/inr değeri bize her t aralığında ne kadar fonksiyonun ne kadar artması veya azalması gereken değer olan dt'yi verecektir. Yine 3 period çizilmiştir. Burada 1 adet fazladan if else kullanılmıştır sebebi ise counter 1 iken önceki değerin olmamasıdır.

GUI Kısmı:



GUI oluşturmak için guide kullanıldı.

Burada hesaplamaları ve oluşan değerleri çizdirmek için 3 fonksiyon oluşturuldu.

draw_first_axis(handles) fonksiyonu

```
function draw_first_axis(handles)
global T;
global y;
global t;
global fnc;
T = str2num(get(handles.period_edit,'String'));
axes(handles.axes1);
if(strcmp(fnc,'tri'))
    [t,y] = tri(T);
    plot(t,y);
    title('Triangle Wave');
elseif(strcmp(fnc,'rect'))
    [t,y] = rect(T);
    plot(t,y);
    title('Square Wave');
end
xlim([-1.5*T 1.5*T]);
ylim([-0.5 1.5]);
```

T > Period, y t’ye göre veri fnc ise hangi fonksiyonun çizildiğini gösteren bir string. Basit olarak bu fonksiyon veriyi rectengle ise rect fonksiyonu ile triangle ise tri fonksiyonu ile elde edip çizdiriyor.

draw_xn(handles) fonksiyonu

```
function draw_xn(handles)
global xn;
global T;
global order;
global t;
global y;
xn = frr_series(t,y,order,T);
xn_axis = [-((length(xn)-1)/2):((length(xn)-1)/2)];

axes(handles.axes2);
stem(xn_axis,abs(xn),'filled');
xlim([xn_axis(1)-1 xn_axis(end)+1]);
title('x[n] Magnitude');


axes(handles.axes3);
stem(xn_axis,angle(xn),'filled');
xlim([xn_axis(1)-1 xn_axis(end)+1]);
title('x[n] Phase');
```

Bu fonksiyon frr_series fonksiyonu kullanarak x[n] değerini order’a göre hesaplıyor. Sonra xn_axis xn değerlerine göre 0 noktasına simetrik olacak şekilde oluşturuyor. Ardından x[n] Büyüklüğünü ve Fazını sırasıyla ilgili eksenlere çizdiriyor.

draw_xt(handles) fonksiyonu

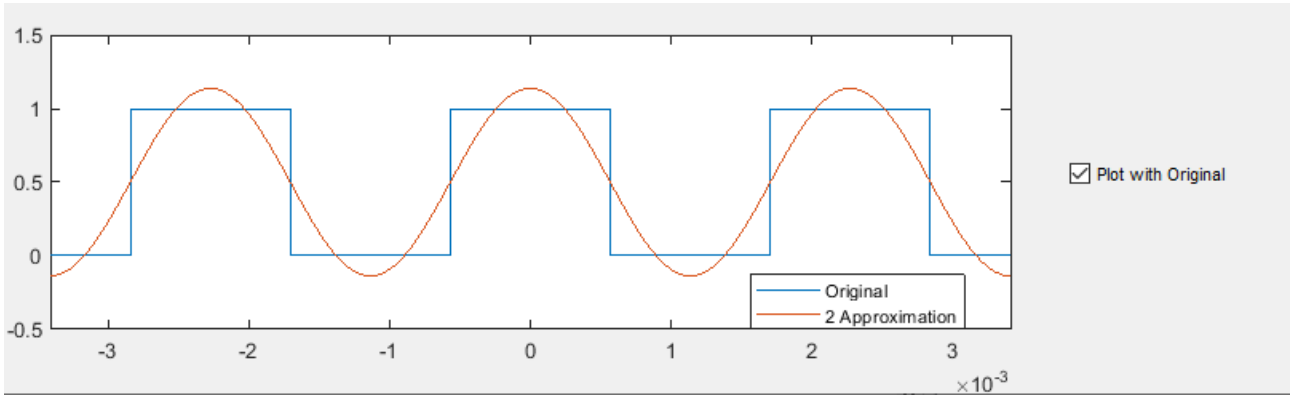
```
function draw_xt(handles)
global xt;
global xn;
global t;
global y;
global T;
global order;
global dsply_original;

xt = find_xt(xn,t,order,T);
axes(handles.axes4);

if dsply_original
    cla;
    plot(t,y);
    hold on;
    plot(t,real(xt));
    xlim([-1.5*T 1.5*T]);
    ylim([-0.5 1.5]);
    legend('Original',string(order)+' Approximation');
else
    cla;
    plot(t,real(xt));
    xlim([-1.5*T 1.5*T]);
    ylim([-0.5 1.5]);
    legend(string(order)+' Approximation');
end
```

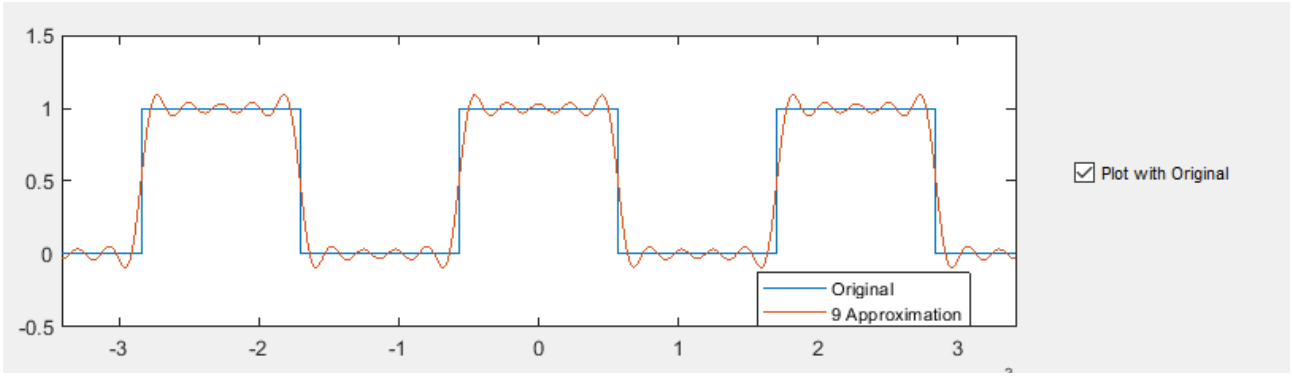
Bu fonksiyon x(t) değerini verilen order'a göre buluyor ve dsply_original içindeki değer 1 ise orijinal fonksiyonla aynı plotta çizidiriyor. 0 ise farklı fonksiyonda çizdiriyor.

Gui kısmında
-> Başlangıçta order 2 olarak atanıyor ve Order'ı gösteren static text deki değer günceleniyor aynı zamanda slider üzerindeki değer de güncelleniyor. dsply_original ise 0 olarak atanıyor.
-> Rectangle Wave butonu ile fnc = 'rect' olarak işaretlenip ilk eksende draw_first axis ile çizdiriliyor. Aynı zamanda başlangıçta default bir order değeri atıldığından draw_xn ve draw_xt fonksiyonları ile diğer eksenlerde çizdiriliyor.
-> Triangle Wave butonunda Rectangle Wave butonundan farklı olarak fnc = 'tri' oluyor.
-> Period'un alındığı edit text'te enter tuşuna basılıp basılmadığı kontrol ediliyor ve basıldıysa fnc değerine bakılıyor. rect ise rect fonksiyonu rect değilse ve herhangi bir değeri yoksa tri fonksiyonu çizdirip order'a göre diğer değerler bulunup onlar da ilgili eksenlere çizdiriliyor.

-> 2nd order butonunda ise order 2 yapılıp slider ve order static text'i güncellenip çizdirme fonksiyonları çağrılıyor.
-> 9th order butonunda ise order 9 yapılıp 2nd order butonu ile aynı şeyler yaptırılıyor.
-> Slider ile order istenilen değerde alınıp çizdirme fonksiyonları kullanarak ilgili eksenlere değerler çizidiriliyor.
-> Plot with Original checkbox'ından ise dsply_original değeri alınıyor. Eğer değer 1 olursa x(t) orijinal fonksiyonla birlikte çizidiriliyor.

SONUÇLARI DEĞERLENDİRME:





Fourier seriesini açtıkça elde edilen sonuç orijinal fonksiyona daha da yaklaşmaktadır. Burada ilk şekilde 2.dereceden fourier serisi açılmıştır. İkinci şekilde ise 9.dereceden fourier serisi açılmıştır. Şekillerden anlaşıldığı üzere 9. dereceden fourier serisi açılımından elde edilen fonksiyon 2.dereceden açılan fonksiyona göre orijinal fonksiyona daha yakındır.

MUHAMMED HALİT TOKLUOĞLU 161110070