

UŞAK ÜNİVERSİTESİ

ELEKTRİK-ELEKTRONİK MÜHENDİSLİĞİ 2022-2023 GÜZ DÖNEMİ

BİLGİSAYAR PROGRAMLAMAYA GİRİŞ DERSİ PROJE RAPORU

Ödev No: Proje 2

Ödev Tarihi :22 Aralık Perşembe 2022

Ödev Teslim Tarihi: 28 Aralık Çarşamba 2022

Adı Soyadı: Şerif Batıkan Çobanoğlu

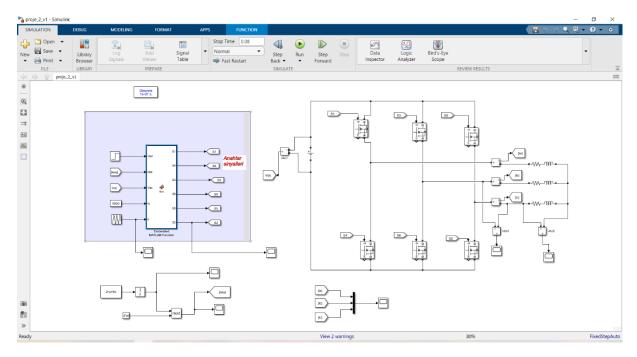
Öğrenci No:190517012

İÇİNDEKİLER

PROJE3
1)UZAY VEKTÖR MODÜLASYONU(SPACE VECTOR MODULATION):
2)MatlabFonksiyonKodları:6
3)Sinyal Cıktıları:

1.PROJE

Bu projede bir inverter simülasyonu yapılmıştır. İnverter devreleri DC'den AC'ye dönüşüm yapılmak istenildiğinde kullanılır. Resim 1'de Matlab/Simulink ortamında oluşturulmuş simülasyon devresi gösterilmektedir. Bu proje kapsamında uzay vektör modülasyon(Space Vector Modulation) yöntemi kullanılarak anahtarlama sinyalleri oluşturulmuştur.



Resim 1 Matlab/Simulink ortamında oluşturulmuş inverter benzetimi

1.1)UZAY VEKTÖR MODÜLASYONU(SPACE VECTOR MODULATION):

Uzay vektörünün dönüşü, 3 fazlı sinüs dalgaları üreten VSI tarafından uygulanabilir. Yöntemin avantajları arasında, PWM anahtarlama frekansında düşük harmonik değeri, sinüsoidal PWM ile karşılaştırıldığında daha az hafıza talebi, vb... sayılabilir.

Çok fazla hesaplama gereksinimi ve DC geriliminin tam olarak değerlendirilememesi bu yöntemin önemli dezavantajlarıdır.

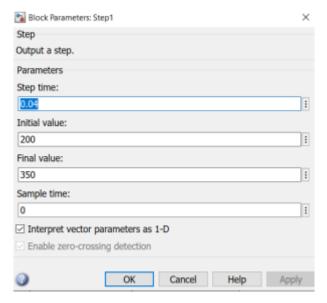
Uzay Vektörü Kavramı

Uzay vektör kavramı, matematiksel bağıntılardan değil, fiziksel gerçeklerden ortaya çıkmıştır. Zamanla sinüs biçimli değişen bir büyüklük uzay fazörle gösterilebilir. Uzay fazörü, temsil ettiği uzay dalgasının en büyük pozitif değerini, büyüklüğü ise dalganın genliğini gösterir. Elektrik makinalarının uzay vektörleri ile incelenmesi aşağıdaki avantajları sağlayacaktır:

- Makinanın akım, gerilim, akı gibi büyüklükleri uzay vektörleri ile ifade edilebilir.
- Tüm uzay vektörleri matematiksel bağıntılarla ifade edilebilir.
- Simetrik ve simetrik olmayan makinaların incelenmesi faz sayılarına bağlı olmadan uzay harmonikleri de eklenerek yapılabilir.
- Moment ifadesi çok daha basit hale dönüşür.
- Uzay fazörleri ile asenkron makinanın karşılıklı etkileşimli denklemleri birbirinden bağımsız hale gelir.
- Sürekli çalışma durumunda uzay fazör ve zaman fazör diyagramları aynıdır.
- Uzay fazörleri ile denklem sayısı azalır.

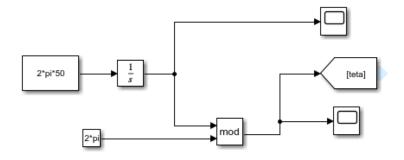
Fonksiyonumuza referans gerilim (Vref), θ açısı, DC gerilim değeri, anahtarlama frekansı ve bir üçgen dalga giriş olarak belirtildi. Çıkış olarak ise S1, S2, S3, S4, S5 ve S6 anahtarlama sinyalleri tanımlandı.

Vref değeri için step fonksiyon kullanınız ve Resim 2'deki değerler girildi.

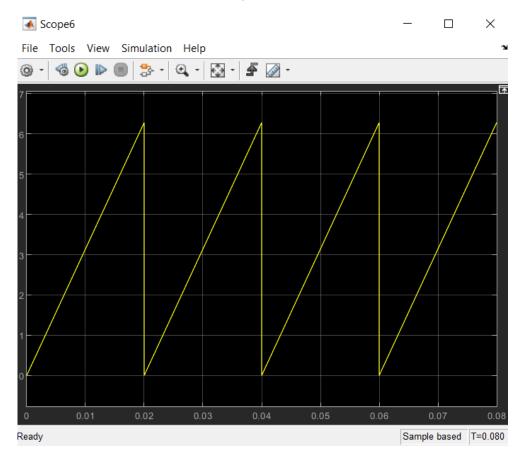


Resim 2 Step function parametreleri

 θ (teta) açısı Resim 3'deki gibi elde edilmiştir. θ (teta) sinyali ise Resim 4'te gösterilmiştir.

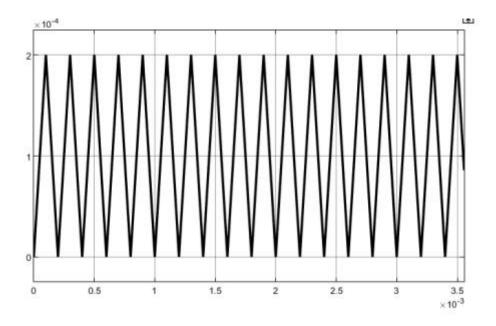


Resim 3 & açısının elde edilmesi.



Resim 4 ϑ sinyali.

Vdc ise DC kaynak geriliminin değeridir. Anahtarlama frekansı 5000 olarak belirlenmiştir. Üçgen dalga işareti ise Resim 5'te gösterilmiştir.



Resim 5 Üçgen dalga işareti.

1.2) Matlab Fonksiyon Kodları:

Matlab fonksiyonu proje föyünde verilen uzay vektör modülasyonu için belirtilen adımlara göre if/else/elseif yapıları ile hazırlanmıştır. Yazılan Matlab fonksiyon kodları aşağıdaki gibidir.

```
\%\%Şerif Batıkan Çobanoğlu 190517012
```

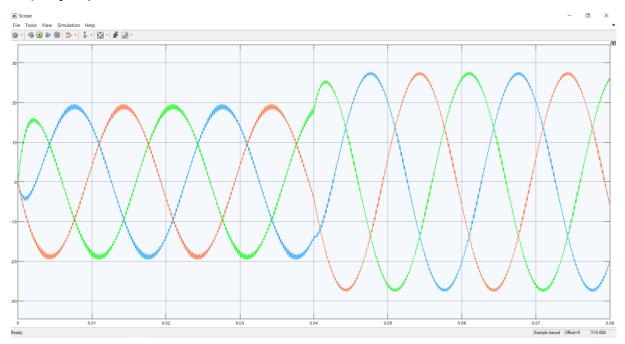
```
\textcolor{red}{\textbf{function}} \hspace{0.1cm} [S1,\!S4,\!S3,\!S6,\!S5,\!S2] = \text{fcn}(Vref,\!teta,\!Vdc,\!fs,\!tr)
S2=0;
S3=0;
S4=0;
S5=0;
S6=0;
ma=(sqrt(3)*Vref)/Vdc;
Ts=1/fs;
if(teta>0 && teta<pi/3) %SEKTOR 1
  Ta = Ts*ma*sin(pi/3-teta);
  Tb = Ts*ma*sin(teta);
  To = Ts-Ta-Tb;
     if(tr < Ta)
        S1=1; S6=1; S2=1;
       S4=0; S3=0; S5=0;
     end
     if(tr > Ta \&\& tr < Ta + Tb)
       S1=1; S3=1; S2=1;
       S4=0; S6=0; S5=0;
     end
     if(tr > Ta + Tb \&\& tr < Ta + Tb + To)
       S1=1; S3=1; S5=1;
       S4=0; S6=0; S2=0;
     end
elseif(teta>pi/3 && teta<(2*pi)/3) %SEKTOR 2
  Ta = Ts*ma*sin((2*pi)/3-teta);
  Tb = Ts*ma*sin(teta-pi/3);
  To = Ts-Ta-Tb;
     if(tr < Ta)
       S1=1; S3=1; S2=1;
```

```
S4=0; S6=0; S5=0;
     end
    if(tr > Ta \&\& tr < Ta + Tb)
       S4=1; S3=1; S2=1;
       S1=0; S6=0; S5=0;
     end
    if(tr > Ta + Tb \&\& tr < Ta + Tb + To)
       S1=1; S3=1; S5=1;
       S4=0; S6=0; S2=0;
    end
elseif(teta>(2*pi)/3 && teta<(3*pi)/3) % SEKTOR 3
  Ta = Ts*ma*sin((3*pi)/3-teta);

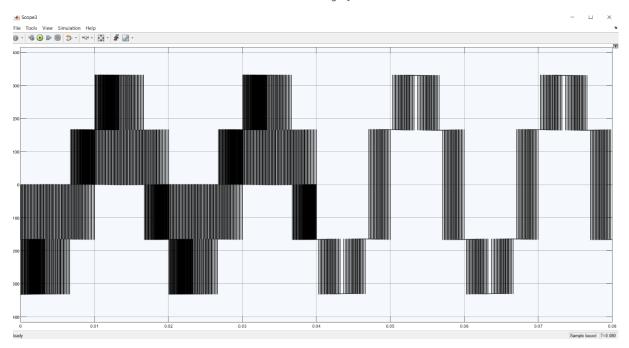
Tb = Ts*ma*sin(teta-(2*pi)/3);
  To = Ts-Ta-Tb;
    if(tr < Ta)
       S4=1; S3=1; S2=1;
       S1=0; S6=0; S5=0;
     end
    if(tr > Ta \&\& tr < Ta + Tb)
       S4=1; S3=1; S5=1;
       S1=0; S6=0; S2=0;
     end
    if(tr > Ta + Tb \&\& tr < Ta + Tb + To)
       S1=1; S3=1; S5=1;
       S4=0; S6=0; S2=0;
    end
elseif((teta>(3*pi)/3) && (teta<(4*pi)/3)) %SEKTOR 4
  Ta = Ts*ma*sin((4*pi)/3-teta);
  Tb = Ts*ma*sin(teta-(3*pi)/3);
  To = Ts-Ta-Tb;
    if(tr < Ta)
       S4=1; S3=1; S5=1;
       S1=0; S6=0; S2=0;
     if(tr > Ta \&\& tr < Ta + Tb)
       S4=1; S6=1; S5=1;
       S1=0; S3=0; S2=0;
     end
     if(tr > Ta + Tb &\& tr < Ta + Tb + To)
       S1=1; S3=1; S5=1;
       S4=0; S6=0; S2=0;
elseif(teta>(4*pi)/3 && teta<(5*pi)/3) % SEKTOR 5
  Ta = Ts*ma*sin((5*pi)/3-teta);
  Tb = Ts*ma*sin(teta-(4*pi)/3);
  To = Ts-Ta-Tb;
    if(tr < Ta)
       S4=1; S6=1; S5=1;
       S1=0; S3=0; S2=0;
     end
    if(tr > Ta \&\& tr < Ta + Tb)
       S1=1; S6=1; S5=1;
       S4=0; S3=0; S2=0;
     end
     if(tr > Ta + Tb && tr < Ta + Tb + To)
       S1=1; S3=1; S5=1;
       S4=0; S6=0; S2=0;
     end
elseif(teta>(5*pi)/3 && teta<2*pi) %SEKTOR 6
  Ta = Ts*ma*sin((6*pi)/3-teta);
  Tb = Ts*ma*sin(teta-(5*pi)/3);
  To = Ts-Ta-Tb;
    if(tr < Ta)
       S1=1; S6=1; S5=1;
       S4=0; S3=0; S2=0;
     end
     if(tr > Ta \&\& tr < Ta + Tb)
       S1=1; S6=1; S2=1;
       S4=0; S3=0; S5=0;
     end
    if(tr > Ta + Tb && tr < Ta + Tb + To)
       S1=1; S3=1; S5=1;
```

```
S4{=}0;\,S6{=}0;\,S2{=}0;\\ end\\ end\\
```

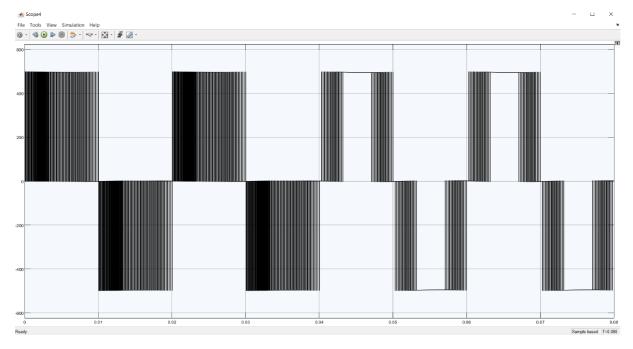
1.3)Sinyal Çıktıları:



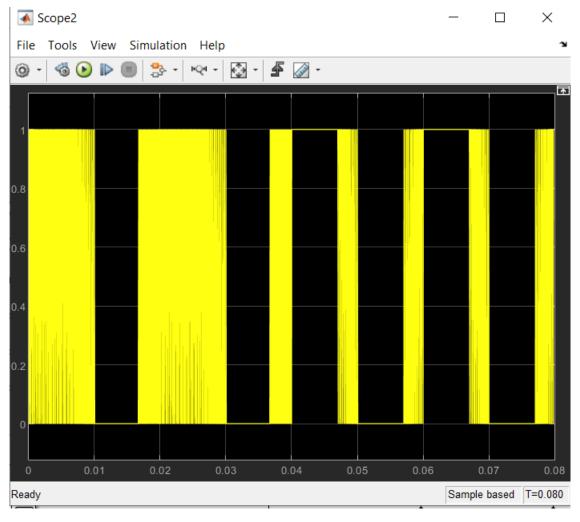
Resim 6 Akım Dalga Şekilleri



Resim 7 Vbc2 Çıktısı



Resim 8 Vbc 3 Çıktısı



Resim 9 S2 Çıktısı