BOGAZICI UNIVERSTY

Department of Electrical & Electronics Engineering

03.02.2021

Spring 2021

Sezen Perçin - 2018401000

435

EE 304 Energy Conversions: Final Project

Introduction

Many engineering applications such as the AC motors utilize the fact that in the presence of two magnetic fields, a torque will be created to align these magnetic fields. In the case where a stator and a rotor is used as a source of two magnetic fields to form an AC machine, a change in the magnetic field is desired for the intended operations.

These changes are provided to system by using a three-phase set of currents which have the same magnitude and different phase values separated by 2*pi/3. At different time instances, these currents form magnetic field density vectors which differ in magnitude and direction with respect to each other and form the rotating magnetic poles.

The number of the forms can be changed by the winding structure and the number of the coils. These coils produce a magnetic field density vector that varies with time and orthogonal to the surface defined by the winding scheme. When all the vectors produced in the system is summed the rotating nature of the magnetic field intensity in the stator can be observed.

This project aims to create a script that animates the magnetic field density vectors created by the coils when the number of the poles is supplied to the script at 0.1Hz.

Materials and Methods

For the project, the places of the coils and the starting point of the vectors along with their directions are all determined using trigonometrical equations and the phasor transform techniques that are implemented in the script that is written with the software tool named MATLAB R2020b provided to university students use.

The current relations and the corresponding magnetic fields density vectors for the 2-pole case are stated in Figure 1[1]. For the cases that has more than 2 poles, the same methods with different direction information's are used.

$$i_{aa'}(t) = I_M \sin \omega t \quad \text{A} \qquad \mathbf{B}_{aa'}(t) = B_M \sin \omega t \angle 0^\circ \quad \text{T}$$

$$i_{bb'}(t) = I_M \sin (\omega t - 120^\circ) \quad \text{A} \qquad \mathbf{B}_{bb'}(t) = B_M \sin (\omega t - 120^\circ) \angle 120^\circ \quad \text{T}$$

$$i_{cc'}(t) = I_M \sin (\omega t - 240^\circ) \quad \text{A} \qquad \mathbf{B}_{cc'}(t) = B_M \sin (\omega t - 240^\circ) \angle 240^\circ \quad \text{T}$$

Fig 1. The equations for the current and magnetic field density vectors created by the coils

(Taken from the textbook Electric Machinery Fundamentals[1])

Discussion and Results

In conclusion, when the magnetic field density vectors and their sum hence the magnetic field intensity vector (assuming linear, isotropic and homogenous medium) is observed, it can be seen that the three-phase set of current results in the magnetic field components created by the coils that change in magnitude (not in direction) with time and their ratio with respect to each other cause the total magnetic field to rotate.

With the script that is prepared, sum of the magnetic field intensity at the center of the stator is shown.

In 2-pole setup, it is seen that the two magnetic pole is formed that can be named as “North and South”. The total magnetic field intensity is shown with the color of black and rotates around the circle that is defined by its maximum value.

In the case where pole number is greater than 2 due to the symmetry that is caused by the magnetic field intensity vectors created by the coils the sum of the vectors at the stator center is zero. Therefore, in the animations the total magnetic field intensity vector vanishes except for the 2-pole case. However, other components clearly shows the formation of the “P” number of poles in a “P”-pole setup.

Different current components and the magnetic density vectors caused by them are shown in different colors in the animation that is created by the written script. Also, the changing direction of the currents can be seen in the circles that symbolizes the presence of the coils.

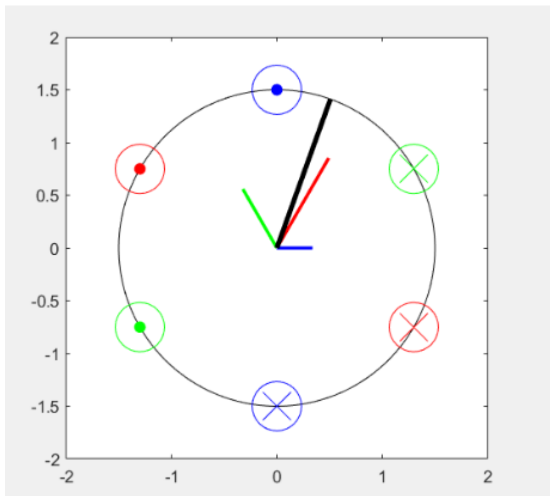

Fig.2 The Magnetic field density vectors created by the coils in 2-pole setup

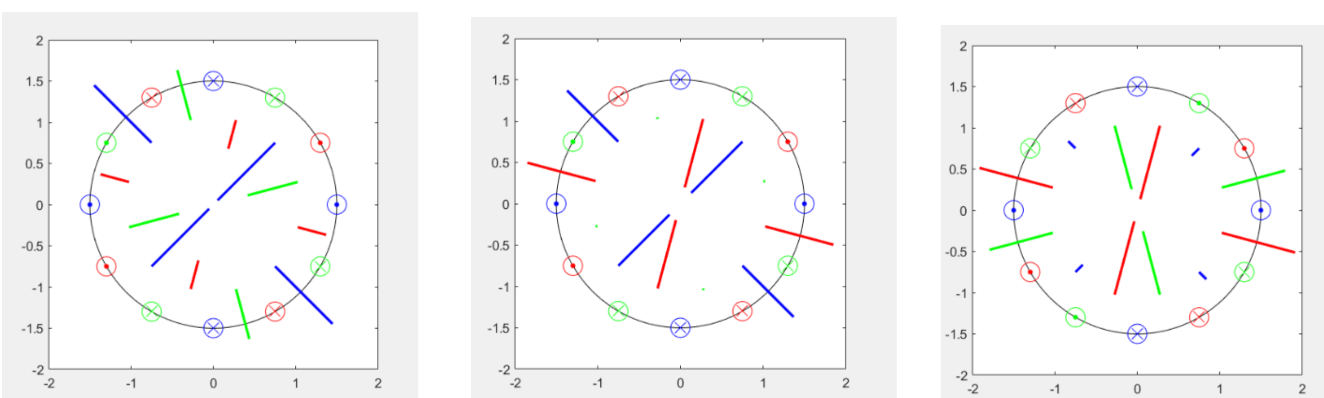

Fig.3 The Magnetic field density vectors created by the coils in 4-pole setup at different times

References

1- Electric Machinery Fundamentals 5th Ed.,2012, Stephen J.Chapman

2 Boğaziçi University, EE304, Spring 2021 Course Materials, Associate Prof. Alpay Özcan

Appendix

```
%P-pole Magnetic field Intensity Vectors Animation Generator
clc;
clear;
%The number of the poles should be stated
pole=4;
%Magnitudes of the currents can be changed to observe the affect
Bm1 =1;
Bm2 =1;
Bm3 =1;
freq =0.1;
w = 2*pi*freq;
t = 0:1/20:1/0.1;
angle1=(2*pi)/pole;
angle2=(2*pi)/(pole*2);
angle3=(2*pi*2)/(pole*3);
circle = 1.5 * (cos(w*t) + j*sin(w*t));
Bxy=[];
I=[];
color=[];
point=[];
if pole==2
   n=1;
else
   n=pole;
end
%generating the magnetic field density equations and the values
for ph=1:3
   for p=1:n
      if ph==1
         color=[color; 'b'];
         Bm=Bm1;
      elseif ph==2
         color=[color; 'g'];
         Bm=Bm2;
      elseif ph==3
         color=[color; 'r'];
         Bm=Bm3;
      end
      imangle=(-pi/2)+angle2+angle1*(p-1)+angle3*(ph-1);
```

```
        toadd=Bm*sin(w*t-(2*pi/3)*(ph-1)).*(cos(imangle) + j*sin(imangle));
        I=[I; sign((Bm*sin(w*t-(2*pi/3)*(ph-1))))];
        Bxy=[Bxy; toadd];
        point=[point; (1.5*cos(imangle)*cos(angle2)) (1.5*sin(imangle)*cos(angle2))];
    end
end
%arranging the windings
if pole>2
    for m=1:n*3
        Bxy(m,:)=Bxy(m,:)*(-1)^(m-1);
    end
end
%net magnetic field density vector
Bnet=sum(Bxy,1);
%stating colors for plotting operations
col=[];
for k=1:pole*3
    if(mod(k,3)==1)
        col(k)='b';
    elseif(mod(k,3)==2)
        col(k)='r';
    elseif(mod(k,3)==0)
        col(k)='g';
    end
end
%generating the direcitons of the currents to plot later
shape=[];
for ii = 1:length(t)
    formshape=[];
    signs=ones(pole*3,1);
    for k=1:3
       if pole==2
           index=k;
       else
           index=1+pole*(k-1);
       end
       start=I(index,ii);
       signs(2*k-1)=start;
       for j=1:pole-1
          if ((2*k-1)+j*3>pole*3)
              index2=mod((2*k-1)+j*3,pole*3)+1;
          else
              index2=(2*k-1)+j*3;
          end
          if mod((2*k-1)+j*3,2)==1
             signs(index2)=start;
          else
```

```
            signs(index2)=start*(-1);
          end
        end
    end
    for k=1:pole*3
        if signs(k)==1
            formshape=[formshape;'x'];
        elseif signs(k)==-1
            formshape=[formshape;'.'];
        else
            formshape=[formshape;'s'];
        end
    end
    shape=[shape formshape];
end
%plotting all the vectors and the informations that is generated before
for ii = 1:length(t)
plot(circle,'k') ; hold on;
    for k=1:pole*3
        plot(1.5*cos(-pi/2+(k-1)*2*pi/(pole*3)),1.5*sin(-pi/2+(k-
1)*2*pi/(pole*3)),strcat(col(k),'o'),'MarkerSize',(20/pole*3));
        plot(1.5*cos(-pi/2+(k-1)*2*pi/(pole*3)),1.5*sin(-pi/2+(k-
1)*2*pi/(pole*3)),strcat(col(k),shape(k,ii)),'MarkerSize',(20/pole*3*4/5));
    end
    for k=1:n*3
        plot([point(k,1) point(k,1)+real(Bxy(k,ii))],[point(k,2)
point(k,2)+imag(Bxy(k,ii))],color(k),'LineWidth', 2 ) ;
    end
    plot([0 real(Bnet(1,ii))],[0 imag(Bnet(1,ii))],'k','LineWidth', 3 ) ;
axis square;
axis([-2 2 -2 2]) ;
drawnow;
hold off;
end
```