# FOC

## PID Control Parameter (k,Y) for FOC

FOC is field oriented control which means the stator and rotor magnetic field are aligned at certain angle. The magnetic field is caused by the current so that stator current is the stator field direction.

Normally, the stator and rotor field are aligned at 90 degree or if there is advanced angle, it is 85 degree and so. Now let’s assume degree θ is the desired angle between them. A typical three phase system below:

Iu +Iv +Iw =0 ----------------(1)

Is cos A =Iu , where Is is resultant of stator current vector ----------------(2)

Is cos (120-A) =Iv = Is/2 \*(-cosA+sinA) ---------------(3)

Is cos (240-A) =Iw = Is /2 \* (-cos A- sinA)  ---------------(4)

Iv2+Iw2+Iu2= Is2(cos A2 +1/2 cosA2+3/2sinA2)= 3/2 \* Is2

Is2 (3/2)= Iv2+Iu2+ (Iu+Iv)2=2(Iu2+Iv2+IvIu)

🡺 Is2=4/3 \*(Iu2+Iv2+IvIu)---------------(5)

Is

A

From above 5 equations, we know Is is the stator current controlling the stator field or the torque. A is the stator field with respect to one reference axis. Let ‘s say if we use AD converter and measured the value of Iu and Iv,

We will have (3)/(2), Iv/Iu= 0.5 \* ( -1 + tan A) .

Rearrange it,

F(a+1)=tan A= (2Iv +Iu)/(Iu ) ; A = table[F(a+1)]

Is2=4/3 \*(Iu2+Iv2+IvIu) , Is2 is +ve value

= 4/3\*Iu2(1+(Iv/Iu)2+Iv/Iu)= 4/3\*lu2(1+a+a2) where a=Iv/Iu

= 4/3\* Iv2(1+a+a2) where a =Iu/Iv

Y=Is2\*3/4= (1+a+a2)\* (Iu | Iv)= (Iu | Iv)\* table [G(a+1)]

Construct the table below to find angle A =table[F(a+1)]

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Angle A | Quart 1 (-30 to 60) | Quart 2 (60-150) | Quart 3 (150-240) | Quart 4 (240-330) |
| condition | Iu>0; a=Iv/Iu;-1<a<1 or 0<a+1<2 | Iv>0; a=Iu/Iv; 0<a+1<2 | Iu<0; a=Iv/Iu; 0<a+1<2 | Iv<0; a=Iu/Iv; 0<a+1<2 |
| Searched value | A=  table[F(a+1)]-30; if A<0;A=360-A | A=  150-table[F(a+1)] | A=  150+table[F(a+1)] | A=  330-table[F(a+1)] |

Construct the table below to find magnitude Is2= table[G(a)]

|  |  |  |
| --- | --- | --- |
|  | Magnitude | |
| condition | a=Iv/Iu; -1<a<1;  denominator =Iu | a=Iu/Iv; -1<a<1  denominator=Iv |
| Searched value | **1+a+a2** | **1+a+a2** |
| Final result | **Iu2 (1+a+a2)** | **Iv2(1+a+a2)** |

Now Is and A are both control parameter. Is controls the torque and A is the angle of stator field. In fact, we need the information of rotor field angle for finding A. Let ‘s say the rotor field angle is ϴ with respect to same reference axis. The rotor field is inclined angle ϴ with respect to reference axis.

Projecting Is to the rotor axis,

Id = Is cos (A-ϴ)= 0 🡺 A-ϴ =90

Iq= Is sin (A-ϴ)= Is 🡺 Is is a target value for control

Is

rotor

A

ϴ

stator

Iq

If value of ϴ is known from measuring the back emf, angle A is directly from calculating Iu, Iv

That is if A- (ϴ +90) is +ve, we adjust Vd= Vd-K(A- ϴ -90) ---------------(7)

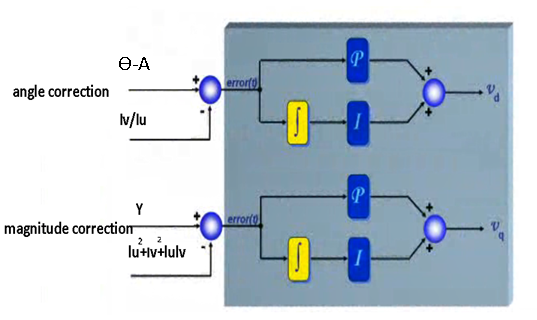
We adjust Vd such that ϴ=A+90;

We assume A-ϴ=90 is done through controlling Vd voltage in the PID loop.

Substitute (7) in to Is, we have

( Is)2 = 4/3 \*(Iu2+Iv2+IvIu)

We replace (Iu2+Iv2+IvIu) = Y , controlling Is2  is same as controlling Is



It should be noted the resultant voltage vector driving the motor and measured current vector are not necessarily in phase. In a motor there is inductive component. Vs is determined through controlling the angle and magnitude of current vector.

Iu

Iv

Angle A

magnitude Is

Vd

A=Table [F(a)]

Y

Y=table [G(a)]\* (Iu|Iv)

9

Vq

ϴ

Angle C

magnitude Vs

SVM table (T1,T2)

Tu

Tv

Tw

PID controller

ɷ

ɷ\*

PI speed control

motor

The whole control system starts with measuring stator current Iu and Iv and then output Vs and angle C.

There is certain value of Vd such that fulfills: k= Iv/Iu

There is certain value of Vq such that fulfills: target Y= lu2+lv2+IuIv

Once Vd and Vq is found : C= artan (Vd/Vq) , Vs= (Vq2+Vd2)1/2

Final step is converting the Vs to PWM pulse

ϴ

Rotor direction

Iq-axis

A

Vs

u-axis

Id-axis

Is

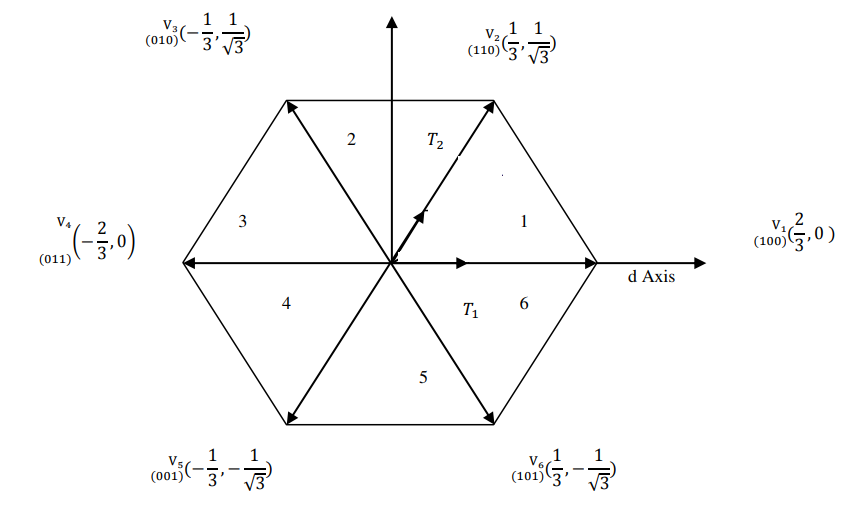
C

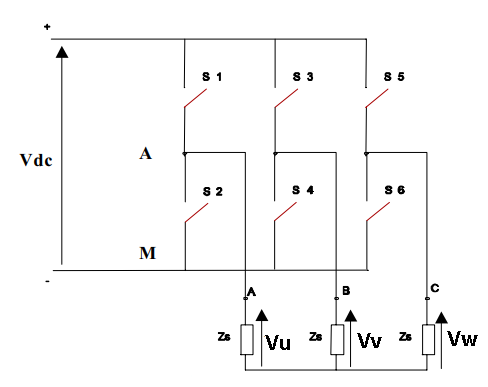
Vd

Vq

Rotor direction

Stator voltage





To project the Vs voltage vector to SVM

ɸ = (C+90+ϴ) mode 60

First find out angle of Vs belongs to which sector (I-6)

Vs will be represented by two adjacent vector V1,V2

## Space Vector Modulation Table (T1,T2)

For a smooth rotation, Vs is running in a unit circle. The resultant Vs is breakdown to two vectors in the SVM table. For example we use V1, V2 and they are separated by 60 degree. Is is the resultant of V1 and V2. We can write equation below:



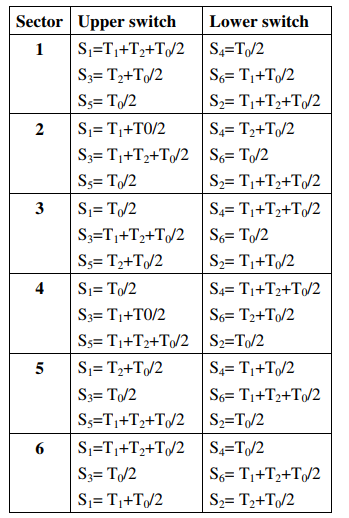
Let sample period is T=T0 +T1+T2

Now Vs = T1/T \*V1+ T2/T \*V2

ɸ

We assume V1 max=V2 max =Vmax=2/3Vdc

Use sin law,



Vs /sin120= (T1/T) \*V1max / sin (60-ɸ)

Vs /sin120 = (T2/T) \* V2max /sin ɸ

Rearranging, we have

T2/T=  sin ɸ \* \* Vs/Vmax = sin ɸ\* K1 --------------------(8)

T1/T= sin (60-ɸ ) \*  \* Vs/Vmax= sin (60-ɸ ) \*K1-------------(9)

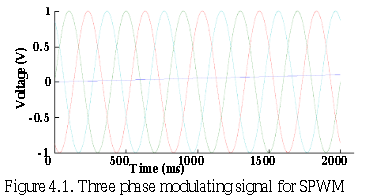
Where K1= \* Vs/Vmax=  Vs/Vdc

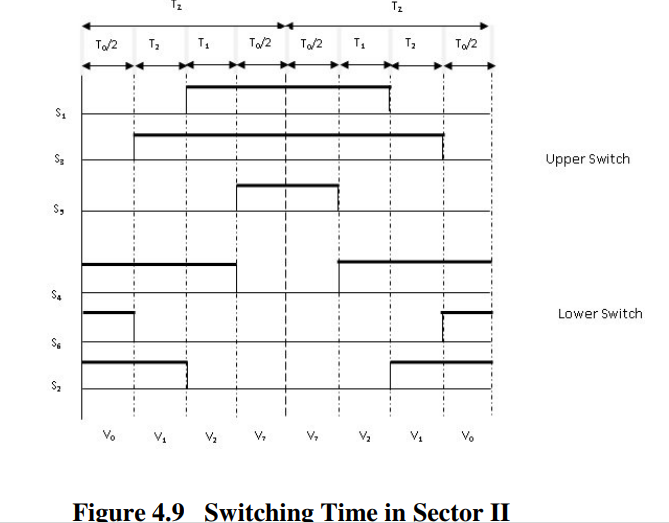
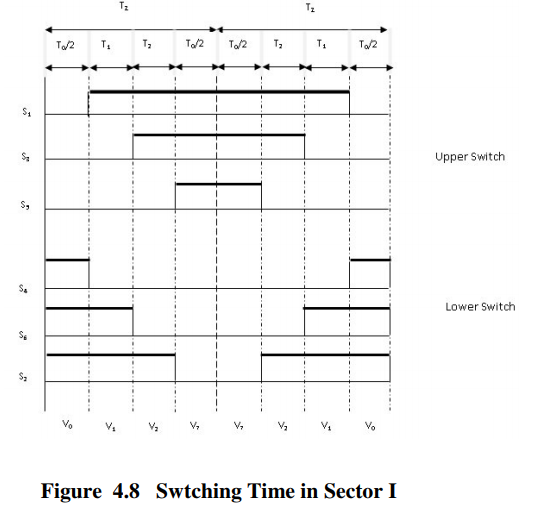
T0/T=1 –(T1/T + T2/T)

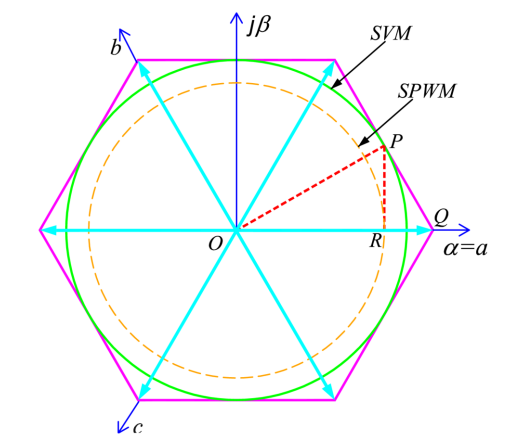
It should be noted (T1+T2)/T is always <1, max of T1+T2 occur at ɸ=30, now,

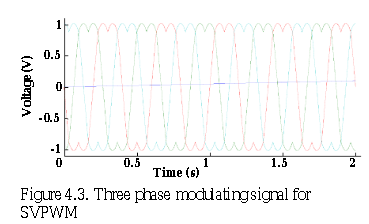
(T1+T2)/T =K1 <1 🡺  Vs< 0.866 Vmax or Vs< Vdc/

That is in SVPWM, Vs is always 14% than Vmax







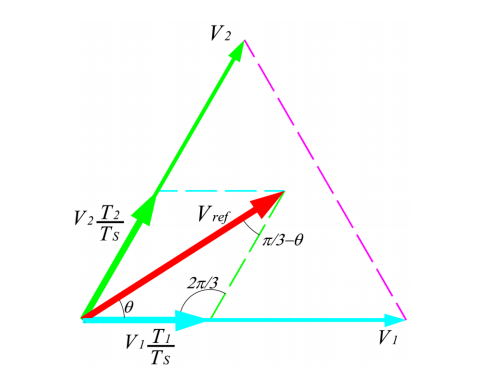


Max power

The reason why SVM cannot use full power is because we want to keep a unit circle trajectory. Previously, it shows θ=0, Vs≤0.866Vmax . What if we allow the trajectory in non-unit circle shape, we shall be able to drive always in maximum power, now atθ=0, Vs=Vmax . The key rule to obey is (T0+T1+T2)/T<1. Let assume T0 is not significant.

(T1+T2)/T =1 ----------------------(10)

new trajectory to utilize full power



from (8),(9), \*  Vs/Vmax (sinɸ+sin (60-ɸ)) =1

Vs/Vmax =(sinɸ+sin (60-ɸ)) ----------------------(11)

Now at ɸ=0, Vs=Vmax,  Vsmin occurs at ɸ=30, Vsmax= 0.866Vmax.

The power increase is counted from the area.

area of hexagon / area of circle in = (0.866)2π=1.10 , we increase the power by 10%

substitute (11) to (8), (9), we have

T1/T=  sin (60-ɸ ) /(sinɸ+sin (60-ɸ)) , T2/T=  sin (ɸ ) /(sinɸ+sin (60-ɸ)).

Summary, for Vs < 0.866Vmax, we keep unity circle and apply

T1/T= sin (60-ɸ )\*K1 ; T2/T= sin (ɸ )\*K1  Where K1= \* Vs/Vmax

If we use max power mode, use a table h(ɸ)= sin (ɸ ) /(sinɸ+sin (60-ɸ))

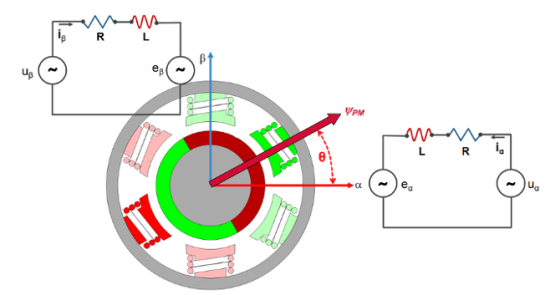
T1/T= h(60-ɸ)\*K1; T2/T= h(ϴ)\*K1  now Vs can extended up to Vmax

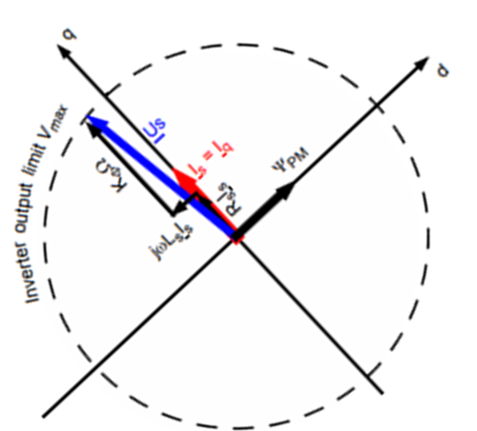
## Rotor Position Sensing using bemf

There are two methods. The zero-crossing method and observer method

Zero crossing method is for 120 degree commutation and need one terminal floating. FOC control is a kind of 180 degree commutation. There is no floating terminal available. To estimate the rotor angle, we should understand the rotor back emf ψ. The voltage equation is Vs=IsR+ d(LIs+ψ)/dt.

Assume ψ= Kɸejθ. dψ/dt= Kɸ\* j\* ejθ\* dθ/dt = Kɸɷ\* j\* ejθ= Kɸɷ \* ejθ+90. The back emf voltage is Kɸɷ and is perpendicular to fluxψ.





q-axis

d-axis

d-axis

Ԑ

IsR

ϴ

ψ

kɸɷ

ɷLsIs

IsR

ϴ

ψ

kɸɷ

ɷLsIs

ϴ

On right picture, what happens if we start with estimate ϴ=0, d-axis align with u-coil. To generate a Is perpendicular to d-axis we will output Vs. In fact, we don’t know the rotor position yet. The real position of rotor may actually incline with angle ϴ to u-axis.

At t=0, we force Id=0, Iq=Is, we will have a set of Vd and Vq

Vd = -ɷLsIs - Kɸɷ sinϴ; Vq=IsR +Kɸɷ cos ϴ

At t=n, we have the rotor rotates and d-axis now incline at estimated angle Ԑ to u-axis, we have:

Vd= -ɷLsIs + Kɸɷ sin(Ԑ-θ)= -ɷLsIs - Kɸɷ sin(θ- Ԑ) --------------(14)

Vq=IsR +Kɸɷ cos (Ԑ-θ) = IsR +Kɸɷ cos (θ- Ԑ) --------------(15)

On d-q plane, our target is to adjust Ԑ 🡪ϴ so that Vd= -ɷLsIs and Vq=IsR +Kɸɷ

We should note that both Ԑ and ϴ share same angular velocity ɷ. The difference is if the estimate angle Ԑ of d-axis may deviate from actual rotor flux angle ϴ by a fixed error value

Remember for Is angle A=90+Ԑ

To solve for ϴ,

We record when A=0 🡺Ԑ=-90, there is a set of Vd1 and Vq1

Vd1= -ɷLsIs - Kɸɷ sin(ϴ+90)= -ɷLsIs - Kɸɷ cos(ϴ)

Vq1=IsR +Kɸɷ cos (ϴ+90)= IsR -Kɸɷ sin (ϴ)

when A=90 🡺Ԑ=0, there are Vd2 and Vq2

Vd2= -ɷLsIs - Kɸɷ sin(ϴ)

Vq2=IsR +Kɸɷ cos (ϴ)

when A=180 🡺 Ԑ=90, there are Vd3 and Vq3

Vd3= -ɷLsIs -Kɸɷ sin(ϴ-90)= -ɷLsIs +Kɸɷ cos(ϴ)

Vq3=IsR+ Kɸɷ cos (ϴ-90) = IsR+ Kɸɷ sin (ϴ)

when A=270 🡺 Ԑ=180, there are two values of Vd4 and Vq4

Vd4= -ɷLsIs -Kɸɷ sin(ϴ-180)= -ɷLsIs +Kɸɷ sin(ϴ)

Vq4=IsR+ Kɸɷ cos (ϴ-180) = IsR- Kɸɷ cos (ϴ)

From the above equations, we arrive at

IsR= (Vq1+Vq2+Vq3+Vq4)/4

ɷLsIs =- (Vd1+Vd2 +Vd3+Vd4)/4

We can now rewrite  
 Vd=(Vd1+Vd2 +Vd3+Vd4)/4- Kɸɷ sin(ϴ-Ԑ)

Vq=(Vq1+Vq2 +Vq3+Vq4)/4+ Kɸɷ cos(ϴ-Ԑ)

In the above equations, we note if ϴ=Ԑ,

Vd=(Vd1+Vd2 +Vd3+Vd4)/4 --------------(16)

Vq=(Vq1+Vq2 +Vq3+Vq4)/4+ Kɸɷ

Only in equation (16), we don’t need to care about Ls or R or ɷ or Kɸ

Now we can use a kalman filter to track the ϴ and ɷ

Kalman filter theory

Kalman filter is used to track a trajectory of a motion. There are 3 values in the equations:

1. Ym 🡺 measured value

Ym

YT

1. Yp 🡺 predicted value
2. YT 🡺 True value of the motion

Yp

In the above case Y represent the location. Y=f(x).

Yp  is obtained from solving the linear equation =Ax+Bu

In real case, predicted/measure/true values are not the same. There are deviations due to noise, due to measurement instrument etc.

∆m

∆p

So for every Ym , there is associated with error ∆m= m2

So for every Yp , there is associated with error ∆p= p2

Ym

YT

Yp

We can measure Ym , we have predicted value Yp . The question is how far the true value deviated from predicted value or in other words how we can adjust the predict value to approach the true value.

Now we assume YT  is somewhere between Yp and Ym

YT= Yp +∆p= Yp+ ∆p \* ∆e/∆e , where ∆e = Ym-Yp= ∆p+∆m

YT= Yp+ ∆p/(∆p+∆m) \* (Ym-Yp), Kalman gain K = ∆p/(∆p+∆m)

YT= Yp+ K \* (Ym-Yp) --------------(17)

Or YT= Yp(1-K) +KYm --------------(18)

K is a value between {0,1}, if K=0, meaning the predicted value converge to true value.

We can start with an estimate error ∆p with the predicted value Yp, ∆m is noise obtained from measurement of Ym which is a fixed value.

∆m=m2= 2

The 3 steps to use Kalman filter is:

1. Find Kalman gain Kn= ∆pn/(∆pn+∆m) , note that the predicted error will change after applied each Kalman filter adjustment
2. Adjusted the predicted value close to true value,

Yk~YT= Yp+ Kn \* (Ym-Yp)

1. After applying Kalman filter adjustment, YT= Yp(1-K) +KYm. Note that the Yp🡺 (1-K)Yp.

The consequence of multiplying Yp by (1-K) is. The error ∆p is multiplied by a factor of (1-K) also. So the new ∆pn+1=(1-K)∆pn.

Every time after applying Kalman filter, the predicted error is reduced by a factor of (1-K). When K =0, it will settle down at ∆pn+1~∆pn.

ϴ and ɷ are used in Kalman filter. Predicted ϴp= Ԑ, Measured ϴm is obtained by Angle A-90. Kalman filter align ϴp to rotor angle ϴ

The complete block diagram is

Iu

Iv

Angle A\*=Table [F(a)]

magnitude Is

Vd

ϴ\*

Y

Y=k\*Is=lu2+Iv2+IuIv

Vq

Angle C

magnitude Vs

SVM table (T1,T2)

Tu

Tv

Tw

PID controller

ɷ

ɷ\*

PI speed control

motor

Kalman Filter

Measured=A predicted=Ԑ+ ∆ϴ\*

dϴ\*/dt

ϴ\*

Secondary form of Kalman filter:

YT= Yp+ Kn \* (Ym-Yp)

YT= (1-Kn) Yp+KnYm

Yp =(YT- KnYm)/ (1-Kn)

Use Kalman filter to replace a PI controller, the block diagram is

Set value =ɷ

Measured value=ɷ\*

Control output= Y

1-K

K

Iu

Iv

Angle A\*=Table [F(a)]

magnitude Is

Vd

ϴ\*

Y

Y=k\*Is=lu2+Iv2+IuIv

Vq

Angle C

magnitude Vs

SVM table (T1,T2)

Tu

Tv

Tw

PID controller

ɷ

ɷ\*

2nd form Kalman filter

motor

Kalman Filter

Measured=A predicted=Ԑ+ ∆ϴ\*

dϴ\*/dt

ϴ\*