Appendix A: Rotating (D-Q) Transformation and Space Vector Modulation Basic Principles

A.1 Rotating Transformation

The DQ transformation is a transformation of coordinates from the three-phase stationary coordinate system to the *dq* rotating coordinate system. This transformation is made in two steps:

1) a transformation from the three-phase stationary coordinate system to the two-phase, so-called ab, stationary coordinate system and

2) a transformation from the ab stationary coordinate system to the *dq* rotating coordinate system.

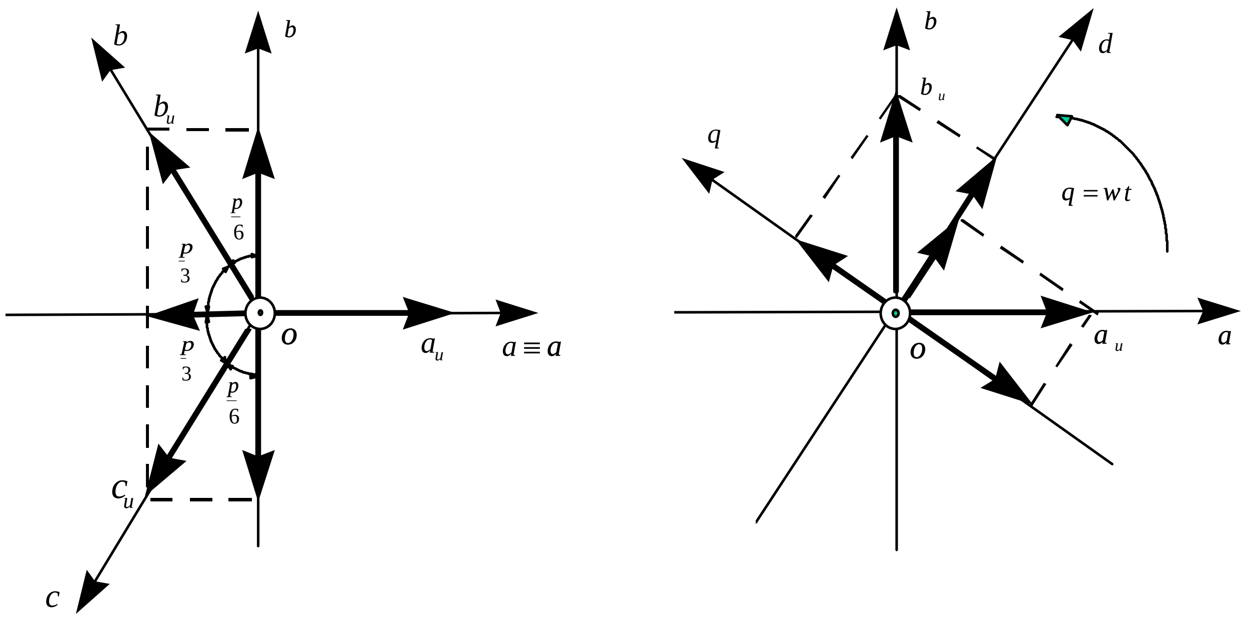
These steps are shown in Figure A.1. A representation of a vector in any n-dimensional space is accomplished through the product of a transpose n-dimensional vector (base) of coordinate units and a vector representation of the vector, whose elements are corresponding projections on each coordinate axis, normalized by their unit values. In three phase (three dimensional) space, it looks like this:

$$X_{abc} = \begin{bmatrix} a_u & b_u & c_u \end{bmatrix} \begin{bmatrix} x_a \\ x_b \\ x_c \end{bmatrix} \tag{A.1}$$

Assuming a balanced three-phase system ($x_o = 0$), a three-phase vector representation transforms to dq vector representation (zero-axis component is 0) through the transformation matrix T, defined as:

$$T = \frac{2}{3}\begin{bmatrix} cos(w\,t) & cos(w\,t - \frac{2}{3}p) & cos(w\,t + \frac{2}{3}p) \\ -sin(w\,t) & -sin(wt - \frac{2}{3}p) & -sin(w\,t + \frac{2}{3}p) \end{bmatrix} \tag{A.2}$$

In other words, the transformation from $X_{abc} = \begin{bmatrix} X_a \\ X_b \\ X_c \end{bmatrix}$ (three-phase coordinates) to $X_{dq} = \begin{bmatrix} X_d \\ X_q \end{bmatrix}$ (dq rotating coordinates), called *Park's transformation*, is obtained through the multiplication



$$\begin{bmatrix} a_u & b_u & o_u \end{bmatrix} = \begin{bmatrix} a_u & b_u & c_u \end{bmatrix} \frac{2}{3} \begin{bmatrix} 1 & 0 & \frac{1}{2} \\ -\frac{1}{2} & \frac{\sqrt{3}}{2} & \frac{1}{2} \\ -\frac{1}{2} & -\frac{\sqrt{3}}{2} & \frac{1}{2} \end{bmatrix}$$

$$\begin{bmatrix} d_u & q_u & o_u \end{bmatrix} = \begin{bmatrix} a_u & b_u & o_u \end{bmatrix} \begin{bmatrix} cosq & -sinq & 0 \\ sinq & cosq & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\begin{bmatrix} d_u & q_u & o_u \end{bmatrix} = \begin{bmatrix} a_u & b_u & c_u \end{bmatrix} \frac{2}{3} \begin{bmatrix} cosq & -sinq & \frac{1}{2} \\ cos(q-\frac{2p}{3}) & -sin(q-\frac{2p}{3}) & \frac{1}{2} \\ cos(q+\frac{2p}{3}) & -sin(q+\frac{2p}{3}) & \frac{1}{2} \end{bmatrix}$$

Figure A.1 Park's transformation from three-phase to rotating dq0 coordinate system of the vector X_{abc} by the matrix T:

$$X_{dq} = TX_{abc} \tag{A.3}$$

The inverse transformation matrix (from dq to abc) is defined as:

$$T^{'} = \begin{bmatrix} cos(w\,t) & -sin(w\,t) \\ cos(w\,t - \frac{2}{3}p) & -sin(w\,t - \frac{2}{3}p) \\ cos(w\,t + \frac{2}{3}p) & -sin(w\,t + \frac{2}{3}p) \end{bmatrix} \tag{A.4}$$

The inverse transformation is calculated as:

$$X_{abc} = T^{'} X_{dq} \tag{A.5}$$

A.2 Space Vector Modulation Basic Principles

The space vector modulation (SVM) basic principles are shown in Figure A.2. A classical sinusoidal modulation limits the phase duty cycle signal to the inner circle. The space vector modulation schemes extend this limit to the hexagon by injecting the signal third harmonic. The result is about 10% (2/1.73 x 100%) higher phase voltage signal at the inverter output. The PWM modulation chops alternatively two adjacent phase voltage and zero voltage signals in a certain pattern producing the switching impulses for the inverter S_a, S_b and S_c. Various SVM modulation schemes have been proposed in literature [72-78] and some recent analyzes show that there is a trade-off between the switching loses and the harmonic content, so-called THD, produced by the SVM modulation [25].

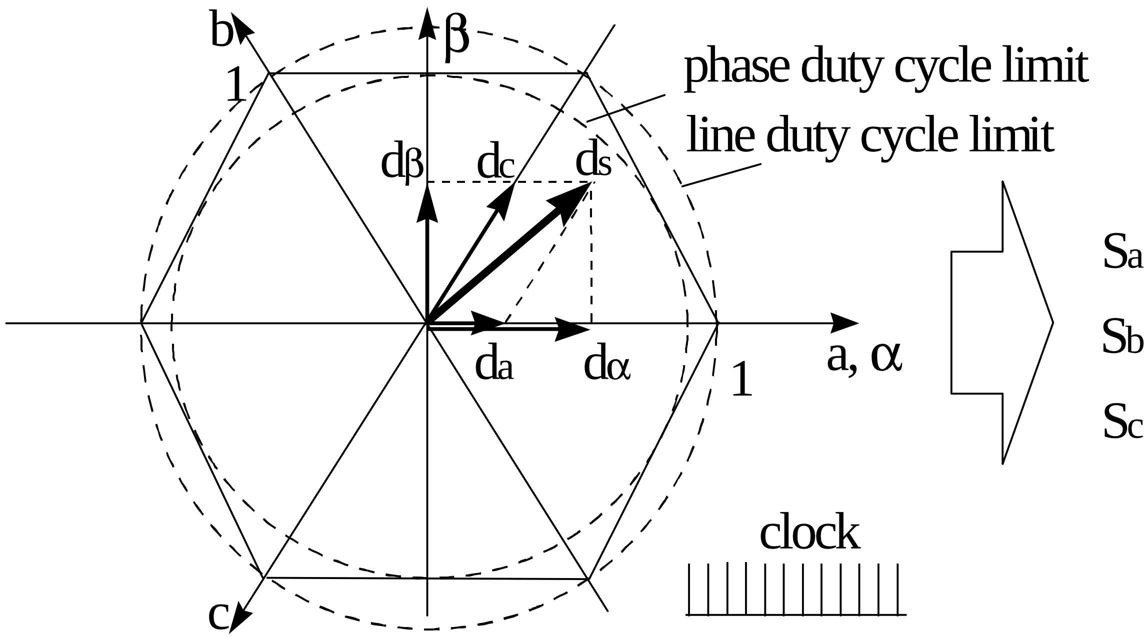


Figure A.2 Space Vector Modulation Basic Principles

Appendix B: Derivation of the Flux-Weakening Equations

B.1 Constant Voltage Constant Power Control

This flux-weakening control method is based on two constraints - constant power and constant phase voltage vector, Eq.s (B.1) - implemented in the PMSM drive d-q model in Eq.s (12) to define the i_d and i_q current reference algorithms, Eq.s (B.2). For the sake of simplicity, the i_d current base value is set to be zero.

$$v_d = V_{db} \qquad v_q = V_{qb}$$

$$P = v_d i_d + v_q i_q = V_{qb} I_{qb} \tag{B.1}$$

$$\left.\begin{array}{l} Ri_d - pL_q w i_q = -pL_q w I_{qb} \Rightarrow i_q \approx I_{qb}\dfrac{W_b}{w} \\ Ri_q\, pL_d w i_d + k_t w = RI_{qb} + k_t W_b \end{array}\right\} \Rightarrow i_d = -\frac{k_t}{pL_d}\left(1 - \frac{W_b}{w}\right) \tag{B.2}$$

The linear relationship between i_d and i_q comes from Eq.s (B.2):

$$i_q = I_{qb}\left(1 + \frac{pL_d}{k_t}\right) \tag{B.3}$$

The critical speed, ω_{cr}, is derived from the VSI maximum current limit, I_s, supposed to be maintained before entering the flux-weakening region, Eq. (B.4) and reached again at $w = w_{cr}$.

$$\sqrt{i_d^2 + i_q^2} = I_s = I_{qb} \tag{B.4}$$

Substituting i_d and i_q in Eq. (B.4) with the expressions from Eq.s (B.3), and solving for ω, the solution for ω_{cr} becomes Eq. (B.5).

$$w_{cr} = \frac{V_{qb}^2 + V_{db}^2}{V_{qb}^2 - V_{db}^2} W_b \tag{B.5}$$

Applying the PMSM dq model Eq.s (12), in Eq. (B.1), the Eq. (B.6) is derived:

$$Ri_d^2 + Ri_q^2 - pL_q w i_q i_d + pL_d w i_d i_q + k_t w i_q = RI_{qb}^2 + k_t W_b I_{qb} \tag{B.6}$$

The constant power is maintained only under the assumption from Eq. (B.7) and negligible voltage drops across inductances L_d and L_q.

$$Ri_d^2 + Ri_q^2 - RI_{qb}^2 \approx p\left(L_q - L_d\right) w i_d i_q \tag{B.7}$$

B.2 Constant Current Constant Power Control

Besides the constant power, this strategy tries to maintain a constant magnitude of the phase current vector, as defined in Eq.s (B.9).

$$\begin{aligned} &\sqrt{i_d^2 + i_q^2} = \sqrt{I_{db}^2 + I_{qb}^2} = I_{qb} \\ &P = v_d i_d + v_q i_q = V_{qb} I_{qb} \end{aligned} \tag{B.9}$$

Substituting the voltages v_d and v_q in the power equation by there expressions from the PMSM drive d-q model, Eq. (B.10), and solving the Eq.s (B.9) for currents i_d and i_q determines the d-q current references, Eq.s (B.11).

$$Ri_d^2 - pL_q w\, i_q i_d + Ri_q^2 + pL_d w\, i_d i_q + k_t w\, i_q = RI_{qb}^2 + k_t W_b I_{qb} \tag{B.10}$$

$$i_q \approx I_{qb} \frac{W_b}{w}\,; \quad i_d \approx -I_{qb}\sqrt{1-\left(\frac{W_b}{w}\right)^2} \tag{B.11}$$

The assumption that $k_t >> p\left(L_d - L_q\right)i_d$ through the entire flux-weakening region is made for the sake of simplicity and is a reasonable assumption. By neglecting the voltage drop across the stator resistance R (which is negligible at high speeds) and substituting i_q and i_d from Eq. (B.11) into the PMSM model Eq.s (12), we can get the v_d and v_q voltage trajectories:

$$\begin{aligned} &v_d = V_{db} = -pL_q I_{qb} W_b \\ &v_q = V_{qb} \frac{w}{W_b} + \frac{L_d}{L_q} V_d \sqrt{\left(\frac{w}{W_b}\right)^2 - 1} \end{aligned} \tag{B.12}$$

The critical speed, ω_{cr}, can be obtained by equalizing the v_q voltage with its base value V_{qb}. The result is the same as the one obtained for the constant voltage, shown in Eq. (B.5). The prevailing speed at which the v_q voltage reaches its minimum, see Figure 29, is calculated from Eq.s (B.12):

$$\frac{dv_q}{dw} = \frac{V_{qb}}{W_b} + \frac{V_d^{'}}{W_b\sqrt{1-\frac{W_b^2}{w^2}}} = 0 \Rightarrow w_p = \frac{W_b}{\sqrt{1-\left(-\frac{V_d^{'}}{V_{qb}}\right)^2}} \tag{B.13}$$

B.3 Optimum Current Vector Control

In contrast to constant power flux-weakening strategies, this strategy leaves the active power to change with the change of the power factor, while maintaining both maximum current and maximum voltage, Eq.s (B.14). In other words, it uses the maximum accessible power.

$$\begin{aligned} \sqrt{i_d^2 + i_q^2} &= \sqrt{I_{db}^2 + I_{qb}^2} = I_{qb} \\ \sqrt{v_d^2 + v_q^2} &= \sqrt{V_{db}^2 + V_{qb}^2} = V_s \end{aligned} \tag{B.14}$$

Developing the voltage constraint from the voltage equations in Eq.s (12), we are getting Eq. (B.15).

$$\left(Ri_d - pL_q w\, i_q\right)^2 + \left(Ri_q + pL_d w\, i_d + k_t w\right)^2 = \left(-pL_q W_b I_{qb}\right)^2 + \left(RI_{qb} + k_t W_b\right)^2 \tag{B.15}$$

After solving (B.14) for i_q and substituting in (B.15), we are getting the quadratic Eq. (B.16) on the variable i_d.

$$Ai_d^2 + Bi_d + C = 0 \tag{B.16}$$

where

$$A = \left(pL_d\right)^2 - \left(pL_q\right)^2; \quad B = 2\,pL_d k_t; \quad C = \left[\left(pL_q I_{qb}\right)^2 + k_t^{\,2}\right]\left(1 - \frac{W_b^2}{w^2}\right) \tag{B.17}$$

and which solution for $i_d<0$ is:

$$i_d < 0 \Rightarrow i_d = -\frac{k_t}{pL_d}\frac{L_d^2}{L_d^2 - L_q^2}\left[1 - \sqrt{1 - \frac{\left(L_d^2 - L_q^2\right)}{L_d^2}\left(\frac{\left(pL_q I_{qb}\right)^2}{k_t^{\,2}} + 1\right)\left(1 - \frac{W_b^2}{w^2}\right)}\right] \tag{B.18}$$

In the case of non-salient PMSM, the solution is more trivial, since $A=0$:

$$i_d = -\frac{C}{B} = -\frac{\left[\left(pL_q I_{qb}\right)^2 + k_t^{\,2}\right]\left(1 - \frac{W_b^2}{w^2}\right)}{2\,pL_d k_t} \tag{B.19}$$

The Eq. (B.18) can be expressed as

$$i_d = I_p\left[1 - \sqrt{1 + K\left(1 - \frac{W_b^2}{w^2}\right)}\right] \tag{B.20}$$

where

$$L_{eq} = \frac{L_q^2 - L_d^2}{L_d}; \quad I_p = \frac{k_t}{pL_{eq}}; \quad K = \frac{L_{eq}}{L_d}\left(\frac{I_{qb}^2}{I_p^2}\frac{L_q^2}{L_{eq}^2} + 1\right) \tag{B.21}$$

Finally, by solving for i_q from Eq.s (B.20) and (B.14) we are getting the i_q current algorithm for the OCV flux-weakening control, Eq. (B.22).

$$i_q = I_{qb}\sqrt{1 - \left(\frac{I_p}{I_{qb}}\right)^2\left[1 - \sqrt{1 + K\left(1 - \frac{W_b^2}{w^2}\right)}\right]^2} \tag{B.22}$$

The v_d and v_q trajectories are obtained by substituting i_d and i_q current in voltage Eq.s (12) by Eq.s (B.20) and (B.22).

$$v_d \approx -pL_q i_q w = -pL_q I_{qb}\sqrt{w^2 - \left(\frac{I_p}{I_{qb}}\right)^2\left[w - \sqrt{w^2 + K\left(w^2 - W_b^2\right)}\right]^2} \tag{B.23}$$

$$v_q \approx pL_d i_d w + k_t w = pL_d I_p\left[w - \sqrt{w^2 + K\left(w^2 - W_b^2\right)}\right] + k_t w \tag{B.24}$$

The speed where the voltage component v_q reaches its minimum from Figure 30, can be obtained from the first derivative of v_q over speed ω in Eq. (B.24).

$$\frac{dv_q}{dt} = pL_d I_p\left[1 - \frac{(1+K)w}{\sqrt{w^2 + K\left(w^2 - W_b^2\right)}}\right] + k_t = 0$$

$$\frac{(1+K)w}{\sqrt{w^2 + K\left(w^2 - W_b^2\right)}} = 1 + \frac{k_t}{pL_d I_p} = \frac{L_q^2}{L_d^2} \Rightarrow w = W_b\sqrt{\frac{\frac{K}{(1+K)}}{1 - (1+K)\left(\frac{L_d^2}{L_q^2}\right)^2}} \Rightarrow v_q = v_{q\,min} \tag{B.25}$$

Appendix C: Program Listings for the PMSM Drive Small and Large Signal Analyses

As an example, here is given a listing of the Matlab code for the Bode analysis and control design of the PMSM drive system (modified for the two-column editorial purposes). The output file, called display, is given on the last page. The system model is developed and stored in the Simulink file vpbode11.m. Because of the model complexity, only the highest hierarchical level is given in Figure C.1. However, this model, as well as the time-domain simulation Matlab models and the equivalent Saber model library, are available at Virginia Power Electronics Center (VPEC) at Virginia Tech.

```
% File K1_bode.m
%
% MATLAB PROGRAM FOR BODE ANALYSIS OF
THE CONTROL OF DQ MODELS OF PERMANENT
MAGNET SYNCHRONOUS MOTORS (PMSM)

% Created by Zoran Mihailovic at VPEC, Virginia
Tech, 1996

clear;
x=[];u=[];y=[];
delete diary;
diary on
% ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
disp ' '
disp 'PARAMETERS OF THE SYSTEM:'
disp '~~~~~~~~~~~~~~~~~~~~~~~~~~'
disp '_s - series mode; _p - parallel mode'
disp ' '
disp ' Motor_d-q_model:'
disp '~~~~~~~~~~~~~~~~~'
disp ' '
disp 'Stator resistance in parallel op. mode [Ohm]:'
R=0.08
disp 'Stator inductance in parallel op. mode [H]:'
L=0.19e-3
disp 'Stator ind. in parallel op. mode in q-axis in [H]:'
Lq=0.8*L
disp 'Stator ind. in parallel op. mode in d-axis in [H]:'
Ld=0.4*L
disp 'Torque constant in parallel op. mode [Nm/A]:'
kt=0.19
disp 'Number of pairs of poles:'
p=3
disp 'Moment of inertia of the rotor [kgm^2]:'
Jm=0.0017
disp 'Maximum speed (in parallel mode) [rad/s]:'
Wmax=14650/9.55
disp 'Series to parallel switch threshold const. [rpm]:'
st1=5500
disp 'Motor shut down [rpm]:'
st2=11500
st=st1;
disp ' '
```

```
disp 'Motor Periferies:'
disp '~~~~~~~~~~~~~~~~~~'
disp 'SVM modulation coefficient:'
Fm=1/sqrt(3)
disp 'Inverter switching frequency [Hz]:'
fs=44000
disp 'Filter inductance per phase [H]:'
Lf=0.34e-3;
disp 'DC link voltage [V]:'
Vdc=370
%Lq=L; Ld=L; Lf=0; % d and q axis ind. for a non-salient PMSM without external filter inductance
disp ' '
disp 'Limiters:'
disp '~~~~~~~~~~'
disp ' '
disp 'Maximum phase voltage [V]'
Vs=Vdc*Fm
disp 'Maximum phase current [A]:'
Is=50
disp 'Current scaling factor (normalization) [1/A]:'
Kim=1/Is
disp 'Speed scaling factor (normalization) [1/(rad/s)]
Kwm=1/(Wmax)
% ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
% LOAD:
% ~~~~~
% ***************************************
% Examples:
% 1) Specified load by look-up table model or
% 2) DC motor (VPEC's testing load):
Ra=0.045;       % resistance of the dc motor windings
La=0.33*1e-3;    % dc motor windings inductance
kt_dc=0.56*1.11;             % torque sensitivity
kb_dc=0.59*0.955*1.11;  % voltage sensitivity (if not saturated kb_dc=kt_dc)
Tfr_dc=1.9;                    % static friction
Bl_dc=0.568*9.55/1000;   % viscous damping
J_dc=0.064;                    % rotor inertia
Rext=10;                        % external stator resistance
Wmax_dc=2250/9.55;      % max. speed [rad/s] with a given DC motor load
% Maximum torque with the stator closed by Rext:
B1=kt_dc*kb_dc;  % torque constant [Nm/(rad/s)]
%B1=0;              % open stator windings
Temax_dc=B1/Rext*Wmax_dc;
% Total load on the PMSM shaft:
J=(J+J_dc);         % inertia on the rotor shaft
Tfr=1;                  % static friction
Blt=Bl_dc;          % viscous damping
kl=Blt+B1/Rext;  % total damping [Nm/(rad/s)]
Wdcmax=2500/9.55;  % maximum speed [rad/s] of the DC motor
% ****************************************
disp ' '
disp 'Torque resistance, kl is given as a load torque vs. speed look-up table.'
disp 'Load inertia [kgm2]:'
J_load=0.316
disp 'Total inertia on the rotor shaft [kgm2]:'
J=Jm+J_load
disp ' '
disp 'Sampling and zero-order hold delays:'
disp '~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~'
disp 'Sampling delay [s]:'
T=1.5/fs;
T=20e-6
disp 'Zero-order hold delay [s]:'
Tz=1.5/fs
Ti=20e-6;    % sampling delay in current loops [s]
% ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
```

```
disp 'CONTROL'
disp '~~~~~~~~'
disp ' '
disp 'Operating point [rad/s]:'
disp '~~~~~~~~~~~~~~~~~~~~~~~~~~~'
wop=5000*pi/30  % operating point w[rad/s]
disp ' '
disp 'Current controllers & decoupling:'
disp '~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~'
disp ' '
% Sampling delay, T(s)=exp(-sTi) causes phase drop of approximately 360deg. at frequency 3/Ti, so the maximum bandwidth (for about 45deg. phase drop caused by the delay) is about 1/(3*Ti). Also, to avoid the influence of the switching, the cross-over frequency should be smaller than fs/5.
% To avoid combined influence of above mentioned, choose the cross-over frequency, fc, smaller or equal to min(fs/10,1/(5*Ti)).
disp 'Open current loop cross-over frequencies [rad/s] for ideally decoupled system:'
disp 'Parallel operating mode:'
wcod_p=sqrt((Vs*Kim)^2-R^2)/(Ld+Lf)     % open d-axis loop cross-over  frequency
wcoq_p=sqrt((Vs*Kim)^2-R^2)/(Lq+Lf)     % open q-axis loop cross-over frequency
disp 'Series operating mode:'
wcod_s=sqrt((Vs*Kim)^2-(4*R)^2)/(4*Ld+Lf)
% open d-axis loop cross-over  frequency
wcoq_s=sqrt((Vs*Kim)^2-(4*R)^2)/(4*Lq+Lf)
% open q-axis loop cross-over frequency
% Desired cross-over frequency:
disp 'Current loop (desired) cross-over freq. [rad/s]:'
fc=min(fs/10,1/(5*Ti)); % desired bandwidth (cross-over frequency) [Hz]
wc=2*pi*fc   % desired bandwidth (cross-over frequency) [rad/s]
disp 'Desired phase margin [deg.]:'
phm_deg=45                 % in degrees
phm=phm_deg*pi/180;     % in radians
disp 'Desired gain margin:'
Gm=0.5                     % in absolute units
Gm_dB=20*log10(Gm)    % in dB
disp 'Gains of current loop PI controllers:'
Kp=wc*L/(Vs*Kim); % init. guess for proportional gain of the PI_s regulator (without filter Lf=0)
Ki=Kp*R/L; % initial guess for integral gain of the PI_s regulator (without filter - Lf=0)
disp 'Parallel operating mode:'
Kpq_p=Kp*(Lq+Lf)/L      % proportional gain of the PI_p regulator in q-axis
Kpd_p=Kp*(Ld+Lf)/L      % proportional gain of the PI_p regulator in d-axis
Kiq_p=Kpq_p*R/(Lq+Lf) % integral gain of the PI_p regulator in q-axis
Kid_p=Kpd_p*R/(Ld+Lf) % integral gain of the PI_p regulator in d-axis
disp 'Series operating mode:'
Kpq_s=Kp*(4*Lq+Lf)/L  % proportional gain of the PI_s regulator in q-axis
Kpd_s=Kp*(4*Ld+Lf)/L  % proportional gain of the PI_s regulator in d-axis
Kiq_s=Kpq_s*4*R/(4*Lq+Lf)       % integral gain of the PI_s regulator in q-axis
Kid_s=Kpd_s*4*R/(4*Ld+Lf)       % integral gain of the PI_s regulator in d-axis
% Series and parallel mode rated speed values (flux-weakening base speed values):
% ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
disp 'Series mode rated speed [rpm]:'
```

```
wb_s=(-4*R*Is*2*kt+sqrt((4*R*Is*2*kt)^2+(Vs^2-
(4*R*Is)^2)*(4*kt^2+...
((Lf+4*Lq)*Is)^2)))/(4*kt^2+((Lf+Lq*4)*Is)^2)*9.55
            % rated speed [rpm]
disp 'Parallel mode rated speed [rpm]:'
wb_p=(-R*Is*kt+sqrt((R*Is*kt)^2+(Vs^2-
(R*Is)^2)*(kt^2+((Lf+Lq)*Is)^2)))/...
(kt^2+((Lf+Lq)*Is)^2)*9.55  % rated speed [rpm]
disp 'wait'
% Speed Loop Controller (symmetrical optimum):
%  ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
% Evaluation of the load torque profile
x0=zeros*[];
options(1)=1e-3; % relative error (default 1e-3).
options(2)=1e-4; % min. step size (def. tend/2000).
options(3)=1;    % max. step size (default tend/50).
tend=1600;
[t,x,y]=gear('loadvp',tend,x0,options);
load load.mat;
[kl,w1,Tl1] = kload(y); % calling Matlab file kload.m
for evaluation of the load resist.  (load torque slope)
w=wop;
for i=1:size(w1)-1
if ((w1(i,1)<=w) & (w1(i+1,1)>w))
kl1=kl(i)
Tlop=Tl1(i)
end
end
kl1=1e-8;
wp1=kl1/J
% Series mode:
Cw=3/2*(2*kt)*Kwm;
wp2=Kiq_s*Vs*Kim/(4*R);
Gw=(wp1+wp2)^3/(8*abs(wp1)*wp2);
wz=Gw*2*abs(wp1)*wp2/(wp1^2+wp2^2);
Kiw_s=Gw*abs(kl1)/(2*Cw);
Kpw_s=Kiw_s/wz;
if abs(wp1)-wp2/20<=0
wcs=wc/10;
Kpw_s=J*wcs/Cw;
Kiw_s=Kpw_s*abs(wp1);
end
disp 'Speed loop PI compensator gains for the series
mode:'
Kpw_s
Kiw_s
% Parallel mode:
Cw=3/2*kt*Kwm;
wp2=Kiq_p*Vs*Kim/R;
Gw=(wp1+wp2)^3/(8*abs(wp1)*wp2);
wz=Gw*2*abs(wp1)*wp2/(wp1^2+wp2^2);
Kiw_p=abs(kl1)/Cw*Gw;
Kpw_p=Kiw_p/wz;
if abs(wp1)-wp2/20<=0
wcs=wc/10;
Kpw_p=J*wcs/Cw;
Kiw_p=Kpw_s*abs(wp1);
end
disp 'Speed loop PI regul. gains for the parallel mode:'
Kpw_p
Kiw_p
% Without back emf elimin. (equivalent DC motor):
% ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
% Parallel mode:
Leq_p=Lq+Lf;
wel_p=R/Leq_p;
C_p=1.5*kt^2/(Leq_p*J);
A_p=0.5*(wel_p+wp1);
B_p=sqrt(1-4*(wel_p*wp1+C_p)/(wel_p+wp1)^2);
sp1_p=A_p*(1-B_p);   sp2_p=A_p*(1+B_p);
Kpq_pdc=abs(Leq_p*sqrt(wc^2+sp2_p^2)/(Vs*Kim));
Kiq_pdc=abs(Kpq_p*sp1_p);
```

```
% Series mode:
Leq_s=4*Lq+Lf;
wel_s=4*R/Leq_s;
C_s=1.5*4*kt^2/(Leq_s*J);
A_s=0.5*(wel_s+wp1);
B_s=sqrt(1-4*(wel_s*wp1+C_s)/(wel_s+wp1)^2);
sp1_s=A_s*(1-B_s);  sp2_s=A_s*(1+B_s);
Kpq_sdc=abs(Leq_s*sqrt(wc^2+sp2_s^2)/(Vs*Kim));
Kiq_sdc=abs(Kpq_s*sp1_s);
Kpq_p=Kpq_pdc;
Kiq_p=Kiq_pdc;
Kpq_s=Kpq_sdc;
Kiq_s=Kiq_sdc;
% Closed/open loop switches:
% ~~~~~~~~~~~~~~~~~~~~~~~~~~
c1=-1;  % command to open(1)/close(-1) current loops
c2=-1;  % command to open(1)/close(-1) speed loop
c3=-1;  % command to open(1)/close(-1) decoup. loops
c4=-1; % command to open(-1)/close(1) anti-windup
% ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
% ESTIMATION OF THE STEADY STATE VALUES AND LINEARIZATION OF THE SYSTEM AT A CHOSEN OPERATING POINT
% ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
% Descriptions of the 'trim' and 'linmod' commands can be obtained by typing 'help trim' and 'help linmod' commands in the matlab workspace
% ALWAYS CHOOSE INITIAL GUESS VALUES SOMETHING HIGHER THAN EXPECTED VALUES IN STEADY STATE; TAKE CARE ABOUT DUTY CYCLE SATURATION!
% Initial guess for the op. point (steady state) values:
Ip_s=kt/(p*(Lq+Lf));
Ip_p=2*kt/(p*(4*Lq+Lf));
if w<=wb_s/9.55
Idref=0;
Iqref=Is;
Tmref=1.5*(2*kt*Iqref+p*4*(Ld-Lq)*Idref*Iqref);
elseif w<st1/9.55
Idref=(Is^2+Ip_s^2)/(2*Ip_s)*((wb_s/9.55/w)^2-1);
Iqref=sqrt(Is^2-Idref^2);
Tmref=1.5*(2*kt*Iqref+p*(Ld-Lq)*Idref*Iqref);
elseif w<=wb_p/9.55
Idref=0; Iqref=Is;
Tmref=1.5*(kt*Iqref+p*(Ld-Lq)*Idref*Iqref);
elseif w<=st2
Idref=(Is^2+Ip_p^2)/(2*Ip_p)*((wb_p/9.55/w)^2-1);
Iqref=sqrt(Is^2-Idref^2);
Tmref=1.5*(kt*Iqref+p*(Ld-Lq)*Idref*Iqref);
else
Idref=0; Iqref=0; Tmref=0;
end
Tlref=Tmref-Tlop;
% Determining the oper. point state space variables:
disp 'If you get warning messages: "Divide by zero." or "Matrix is close to singular or badly scaled." or you want to speed up convergence process, move slightly your initial guess vector around the operating point inside the trim command.'
disp 'To continue press any key.'
pause
w=wop+1;   % moving the initial guess around the desired operating point
disp ' '
disp 'Steady state values at given operating point:'
vpbode11; % calling SIMULINK MODEL stored in file vpbode11.m
%Idref=0;
wmin=0; wmax=2;
[x,u,y,dx]=trim('vpbode11',[0;w;0;Iqref;Idref;0;0;0;0;0;0;0;0;0],[Idref;Iqref;Tlref;w],[Idref;Iqref;0;0;Idref;Iqref;w;Tmref;w;Iqref;0],[],[1;2;4],[])
```

```
disp ' '
disp 'Locations of state space variables on the simulink block diagram vpbode11:'
x0=x;
[sizes,x0,xstr]=vpbode11
% Idm=Idref; Iqm=Iqref; Id=Idref; Iq=Iqref;
% CLOSED SPEED LOOP TRANSFER FUNCTIONS
% ************************************
c1=-1;c2=-1;c3=-1; % commands for closed speed loop
[A1,B1,C1,D1]=linmod('vpbode11',x,u);
%  linearization of the system at the operating point
% Outputs:
% 1 - id current (sampled)
% 7  - motor speed [rad/s]
% 2 - iq current (sampled)
% 8  - motor torque [Nm]
% 3 - duty cycle command in d-axis, d_d
% 9  - reference speed [rad/s]
% 4 - duty cycle command in q-axis, d_q
% 10 - reference iq current
% 5 - id current on the motor terminal
% 11 - reference id current
% 6 - iq current on the motor terminal
% 12 - speed loop gain (speed PI controller) output
n=12; % number of outputs
for i=1:n
[ng33,dg33]=ss2tf(A1,B1,C1(i,:),D1(i,:),3);
nng33(i,1:length(ng33))=ng33;
ddg33(i,1:length(dg33))=dg33;
[ng34,dg34]=ss2tf(A1,B1,C1(i,:),D1(i,:),4);
nng34(i,1:length(ng34))=ng34;
ddg34(i,1:length(dg34))=dg34;
end;
% CLOSED CURRENT LOOP - OPEN SPEED LOOP TRANSFER FUNCTIONS
% ****************************************
c1=-1;c2=1;c3=-1; % switch commands for closed current loop analysis
[A2,B2,C2,D2]=linmod('vpbode11',x,u);
%linearization of the system at the operating point
for i=1:n
[ng1,dg1]=ss2tf(A2,B2,C2(i,:),D2(i,:),1);
nng1(i,1:length(ng1))=ng1;
ddg1(i,1:length(dg1))=dg1;
[ng2,dg2]=ss2tf(A2,B2,C2(i,:),D2(i,:),2);
nng2(i,1:length(ng2))=ng2;
ddg2(i,1:length(dg2))=dg2;
[ng3,dg3]=ss2tf(A2,B2,C2(i,:),D2(i,:),3);
nng3(i,1:length(ng3))=ng3;
ddg3(i,1:length(dg3))=dg3;
end
% OPEN CURRENT LOOP TRANS. FUNCTIONS
% ************************************
% a) with decoupling:
c1=1; c2=1; c3=-1; % open current loop commands
[A3,B3,C3,D3]=linmod('vpbode11',x,u);
% linearization of the system at the operating point
for i=1:n
[ng11,dg11]=ss2tf(A3,B3,C3(i,:),D3(i,:),1);
nng11(i,1:length(ng11))=ng11;
ddg11(i,1:length(dg11))=dg11;
[ng22,dg22]=ss2tf(A3,B3,C3(i,:),D3(i,:),2);
nng22(i,1:length(ng22))=ng22;
ddg22(i,1:length(dg22))=dg22;
end;
% b) without decoupling:
c1=1; c2=1; c3=1; % open current loop commands
[A3a,B3a,C3a,D3a]=linmod('vpbode11',x,u);
% linearization of the system at the operating point
for i=1:n
[ng11a,dg11a]=ss2tf(A3a,B3a,C3a(i,:),D3a(i,:),1);
nng11a(i,1:length(ng11a))=ng11a;
```

```
ddg11a(i,1:length(dg11a))=dg11a;
[ng22a,dg22a]=ss2tf(A3a,B3a,C3a(i,:),D3a(i,:),2);
nng22a(i,1:length(ng22a))=ng22a;
ddg22a(i,1:length(dg22a))=dg22a;
end;
disp ' '
disp 'SUMMARY'
disp '~~~~~~~'
disp 'Op. point [rpm]:'
w_rpm=y(7)*30/pi
disp 'Full load (+15deg.C); Vdc=370V; fs=44kHz;'
disp 'Digital delays: T=1.5/fs; Tz=1.5/fs'
disp ' '
if w_rpm<=st1
disp 'Operating mode:'
disp 'Series'
disp ' '
disp 'Compensator gains in id current loop:'
disp 'Integral gain:'
Kid_s=Kid_s
disp 'Proportional gain:'
Kpd_s=Kpd_s
disp 'Compensator gains in iq current loop:'
disp 'Integral gain:'
Kiq_s=Kiq_s
disp 'Proportional gain:'
Kpq_s=Kpq_s
disp 'Speed Loop Compensator Gains:'
disp 'Proportional gain:'
Kpw_s=Kpw_s
disp 'Integral gain:'
Kiw_s=Kiw_s
else
disp 'Operating mode:'
disp 'Parallel'
disp ' '
disp 'Compensator gains in id current loop:'
disp 'Integral gain:'
Kid_p=Kid_p
disp 'Proportional gain:'
Kpd_p=Kpd_p
disp 'Compensator gains in iq current loop:'
disp 'Integral gain:'
Kiq_p=Kiq_p
disp 'Proportional gain:'
Kpq_p=Kpq_p
disp 'Speed Loop Compensator Gains:'
disp 'Proportional gain:'
Kpw_p=Kpw_p
disp 'Integral gain:'
Kiw_p=Kiw_p
disp 'Estimated load torque slope:'
kload=kl1
disp 'Estimated load torque:'
Tload=Tlop
end
K1_zpk;            % zeros, poles & gains
K1_bp;  % Bode, Nyquist & Root Locus plots
diary off
disp 'To begin step resp. simulation, press any key.'
pause
% Step response:
% ~~~~~~~~~~~~~
c1=-1;c2=-1;c3=-1;c4=1;
wmin=y(7); wmax=wmin+10;
vpbode11;
x0=x; tf=1;
options(1)=1e-4; % relative error (tol.)
options(2)=1e-6; % min step size
options(3)=1e-3; % max step size
[t,x,y]=gear('vpbode11',tf,x0,options);
Step_resp  % calling the file for plotting the sim. data
```

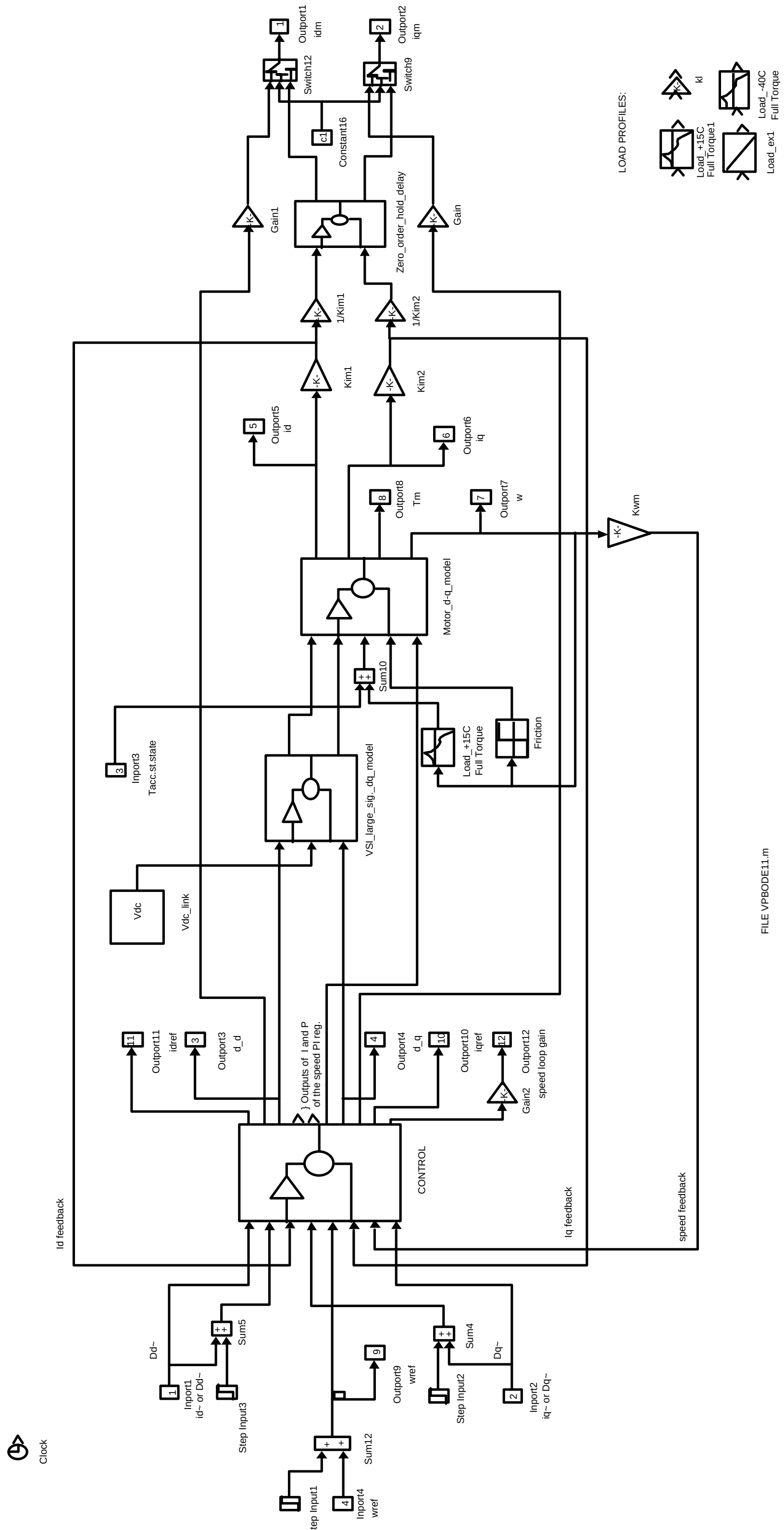


Figure C.1 Simulation (Simulink) hierarchical model for control design of PMSM drives

The output file, modified for printing onto one page, is following:

Conditions:Complete decoupling, calculated load, w=5000rpm - series mode in flux-weakening reg.
PARAMETERS OF THE SYSTEM:
~~~~~~~~~~~~~~~~~~~~~~~~~~
_s - series mode; _p - parallel mode

Motor_d-q_model:
~~~~~~~~~~~~~~~~~~
Stator resistance in parallel op. mode [Ohm]:

R =
0.08000000000000

Stator inductance in parallel op. mode [H]:

L =
1.900000000000000e-004

Stator inductance in parallel operating mode in q-axis [H]:

Lq =
1.520000000000000e-004

Stator inductance in parallel operating mode in d-axis [H]:

Ld =
7.600000000000000e-005

Torque constant in parallel op. mode [Nm/A]:

kt =
0.19000000000000

Number of pairs of poles:

p =
3

Moment of inertia of the rotor [kgm^2]:

Jm =
0.00170000000000

Maximum speed (in parallel mode) [rad/s]:

Wmax =
1.534031413612565e+003

Series to parallel switch treshold constant [rpm]:

st1 =
5500

Motor shut down [rpm]:

st2 =
11500

Motor Periferies:
~~~~~~~~~~~~~~~~~~
SVM modulation coefficient:

Fm =
0.57735026918963

Inverter switching frequency [Hz]:

fs =
44000

Filter inductance per phase [H]:
DC link voltage [V]:

Vdc =
370

Limiters:
~~~~~~~~~~
Maximum phase voltage [V]

Vs =
2.136195996001616e+002

Maximum phase current [A]:

Is =
50

Current scaling factor (normalization) [1/A]:

Kim =
0.02000000000000

Speed scaling factor (normalization) [1/(rad/s)]:

Kwm =
6.518771331058021e-004

Load specs:
~~~~~~~~~~~~~~~~~~~~~~~~~
Torque resistance, kl is given as load torque vs. speed look-up table.
Load inertia [kgm^2]:

J_load =
0.31600000000000

Total inertia on the rotor shaft [kgm^2]:

J =
0.31770000000000

Sampling and zero-order hold delays:
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
Sampling delay [s]:

T =
2.000000000000000e-005

Zero-order hold delay [s]:

Tz =
3.409090909090909e-005

SUMMARY
~~~~~~~

Op.point [rpm]:

w_rpm =
5.028742146299546e+003

Full load (+15deg.C); Vdc=370V; fs=44kHz;
Sampl. & zero order hold delays:T=1.5/fs;Tz=1.5/fs

Operating mode:
Series

Compensator gains in id current loop:
Integral gain:

Kid_s =
2.070672571493334e+003

Proportional gain:

Kpd_s =
4.16722855013033

Compensator gains in iq current loop:
Integral gain:

Kiq_s =
2.070672571493334e+003

Proportional gain:

Kpq_s =
6.13436749304900

Speed Loop Compensator Gains:
Proportional gain:

Kpw_s =
2.363791448167225e+006

Integral gain:

Kiw_s =
1.278991973370939e+006

kl1 =
-0.17190000000000
~~~~~~~