16

Linearization of Permanent Magnet Synchronous Motor Using MATLAB and Simulink

A. K. Parvathy and R. Devanathan
Hindustan Institute of Technology and Science, Chennai
India

1. Introduction

Permanent magnet machines, particularly at low power range, are widely used in the industry because of their high efficiency. They have gained popularity in variable frequency drive applications. The merits of the machine are elimination of field copper loss, higher power density, lower rotor inertia and a robust construction of the rotor (Bose 2002).
In order to find effective ways of designing a controller for PM synchronous motor (PMSM), the dynamic model of the machine is normally used. The dynamic model of PM motor can be derived from the voltage equations referred to direct (d) and quadrature (q) axes (Bose 2002).The model derived essentially has quadratic nonlinearity. Linear control techniques generally fail to produce the desired performance. Feedback linearization is a technique that has been used to control nonlinear systems effectively.
By applying exact linearization technique (Cardoso & Schnitman 2011) it is possible to linearize a system and apply linear control methods. But this requires that certain system distributions have involutive property. An approximate feedback linearization technique was formulated by Krener (Krener 1984) based on Taylor series expansion of distributions for non-involutive systems.
Chiasson and Bodson (Chiasson & Bodson 1998) have designed a controller for electric motors using differential geometric method of nonlinear control based on exact feedback linearization. But from a practical point of view, this technique suffers from singularity issues. If the system goes into a state, during the course of the system operation, where the singularity condition is satisfied, then the designed controller will fail.
Starting with the quadratic model of PMSM, we apply quadratic linearization technique based on coordinate and state feedback. The linearization technique used is the control input analog of Poincare's work (Arnold 1983) as proposed by Kang and Krener (Kang & Krener 1992) and further developed by Devanathan (Devanathan 2001,2004) .The quadratic linearization technique proposed is on the lines of approximate linearization of Krener (Krener 1984) and does not introduce any singularities in the system compared to the exact linearization methods reported in (Chiasson & Bodson 1998).
MATLAB simulation is used to verify the effectiveness of the linearization technique proposed. In this chapter, MATLAB/SIMULINK modeling is used to verify the effectiveness of the quadratic linearization technique proposed. In particular, the application of MATLAB and SIMULINK as tools for simulating the following is described:

i. Dynamic model of a sinusoidal Permanent Magnet Synchronous machine(PMSM)
ii. Application of nonlinear coordinate and state feedback transformations to linearize the PMSM model and
iii. Tuning the transformations against a linear system model put in Brunovsky form employing error back propogation.

As these applications were somewhat sophisticated , customization and improvisation of the MATLAB/SIMULINK tools were essential. These applications are described in detail in this chapter.
In section 2, linearization of dynamic model of PMSM is discussed and simulation results using MATLAB are given. In section 3, linearization of PMSM machine model is given. Construction of PMSM model using SIMULINK and verification of linearization of PMSM SIMULINK model is given in Section 4. Tuning of the linearizing transformations to account for unmodelled dynamics is discussed in Section 5. In section 6, the chapter is concluded.

2. Linearization of dynamic model of PMSM

2.1 Dynamic model of PMSM

The dynamic model of a sinusoidal PM machine, considering the flux-linkage λ_f to be constant and ignoring the core-loss, can be written as (Bose 2002).

$$\dot{i}_d = \frac{v_d}{L_d} - \frac{R}{L_d} i_d + \frac{L_q}{L_d} p\omega_r i_q \tag{1}$$

$$\dot{i}_q = \frac{v_q}{L_q} - \frac{R}{L_q} i_q - \frac{L_d}{L_q} p\omega_r i_d - \frac{\lambda_f p\omega_r}{L_q} \tag{2}$$

$$\dot{\omega}_r = \frac{1.5p}{J}[\lambda_f i_q + (L_d - L_q) i_d i_q] \tag{3}$$

where all quantities in the rotor reference frame are referred to the stator.
L_q, L_d are q and d axis inductances respectively; R is the resistance of the stator windings; i_q, i_d are q and d axis currents respectively; v_q, v_d are q and d axis voltages respectively; ω_r is the angular velocity of the rotor; λ_f is the amplitude of the flux induced by the permanent magnets of the rotor in the stator phases and p is the number of pole pairs.
Using linear coordinate and state feedback transformations (Kuo 2001) the dynamic model can be written with the linear part put in Brunovsky form (Parvathy et. al. 2005,2006) as

$$\dot{x} = Ax + Bu + f^{(2)}(x) \tag{4}$$

where $u = [\,u_1 \quad u_2\,]^T = [\,v_q \quad v_d\,]^T ; x = [i_q \quad i_d \quad \omega_e]^T$

$$A = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \qquad B = \begin{bmatrix} 0 & 0 \\ 0 & 1 \\ 1 & 0 \end{bmatrix};\ f^{(2)}(x) = \begin{bmatrix} b_1 x_2 x_3 \\ b_2 x_1 x_3 \\ b_3 x_1 x_2 \end{bmatrix}$$

where b_1, b_2, b_3 are constants derived from the motor parameters.

2.2 Linearization of the dynamic model

Coordinate and state feedback transformations in quadratic form (Kang & Krener 1992)

$$y = x + \phi^{(2)}(x) \tag{5}$$

$$u = (I_2 + \beta^{(1)}(x))v + \alpha^{(2)}(x) \tag{6}$$

Where

$$y = [y_1 \quad y_2 \quad y_3]^T ; v = [v_1 \quad v_2]^T$$

$\phi^{(2)}(x); \alpha^{(2)}(x)$ and $\beta^{(1)}(x)$ are derived by solving the Generalized Homological Equations (Kang & Krener 1992) .Applying the transformations (5) and (6), (4) is reduced to

$$\dot{y} = Ay + Bv + O^{(3)}(y,v)$$

where $O^{(3)}(y,v)$ represent third and higher order nonlinearities .

2.3 Verification of linearization using MATLAB function

The problem now is to apply MATLAB to verify the theoretical result on quadratic linearization of the dynamic model (4). Expanding (4), we can see that the expression of the derivative of each state variable has the other two state variables in it. This becomes difficult to solve using manual methods of differential equation solution. The tool selected for solving the dynamic equations is the MATLAB function called ODE45.

ODE45 is a MATLAB function that solves initial value problems for ordinary differential equations (ODEs). It uses the iterative Runge Kutta method of solving equations. Hence, this function does not return the solution as an expression, but the values of the solution function at discrete instants of time. ODE45 is based on an explicit Runge-Kutta formula, the Dormand-Prince pair. It is a *one-step* ode45 –in the sense that, in computing $y(t_n)$, it needs only the solution at the immediately preceding time point $y(t_{n-1})$ .In general, ode45 is the best function to apply as a "first try" for most problems.

[t,Y] = ode45(odefun,tspan,y0) with tspan = [t0 tf] integrates the system of differential equations from time t0 to tf with initial conditions y0. Function f = odefun(t,y) is defined, where t corresponds to the column vector of time points and y is the solution array. Each row in y corresponds to the solution at a time returned in the corresponding row of t. To obtain solutions at the specific times t0, t1,...,tf (all increasing or all decreasing), we use tspan = [t0,t1,...,tf].

[t, Y] = ode45(odefun,tspan,y0,options) solves as above with default integration parameters replaced by property values specified in 'options'. Commonly used properties include a scalar relative error tolerance RelTol (1e-3 by default) and a vector of absolute error tolerances AbsTol (all components are 1e-6 by default).

The PLOT function is used to create a computer-graphic plot of any two quantities or a group of quantities with respect to time.

A simulation of (4) was carried out for different values of inputs u_1 and u_2 in the open loop before applying the linearizing transformations where b_1 = -0.0165*(10^-3) ; b_2 -186.96; b_3 =5754386. Fig 1 shows a plot of x_3 (angular velocity) versus time for pulse inputs

u_1= 0.1{u(t)-u(t-1)} and u_2=0.1{u(t)-u(t-1)} where u(t) is a unit step function . The system is oscillatory as seen from Fig 1. Fig. 2 shows the result of simulation of equation (4) after the linearizing transformations (5) and (6) are applied to the system. It shows a plot of y_1,y_2,y_3 against time for pulse inputs v_1=0.2{u(t)-u(t-1)}, v_2=0.2{u(t)-u(t-1)}. From Fig 2 it is seen that the system shows a stable response for a pulse input in v_1 and v_2 even under open loop conditions.

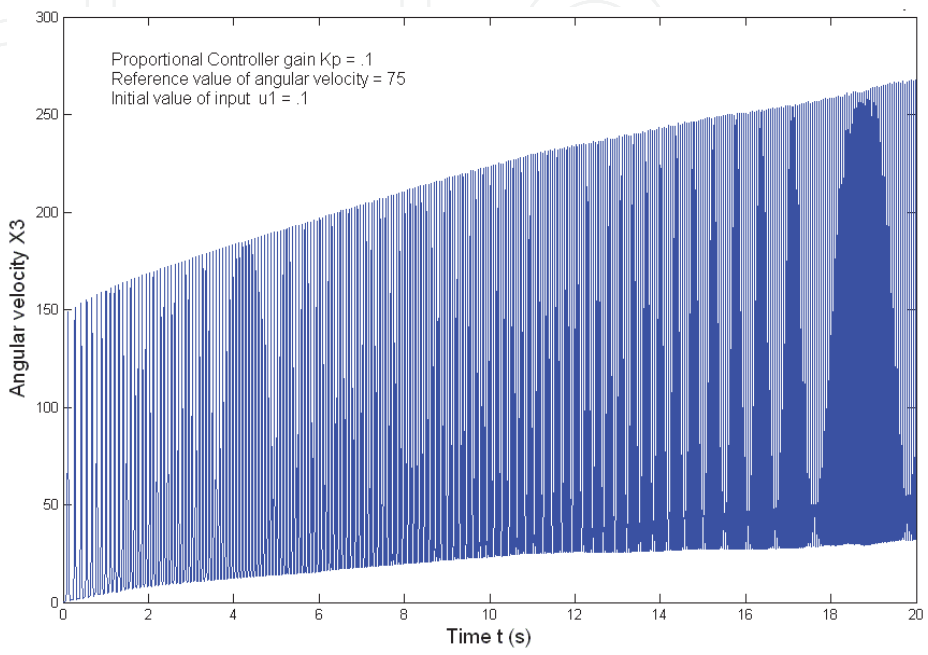

Fig. 1. Time response of angular velocity with v_1=0.1 and v_2=0.1

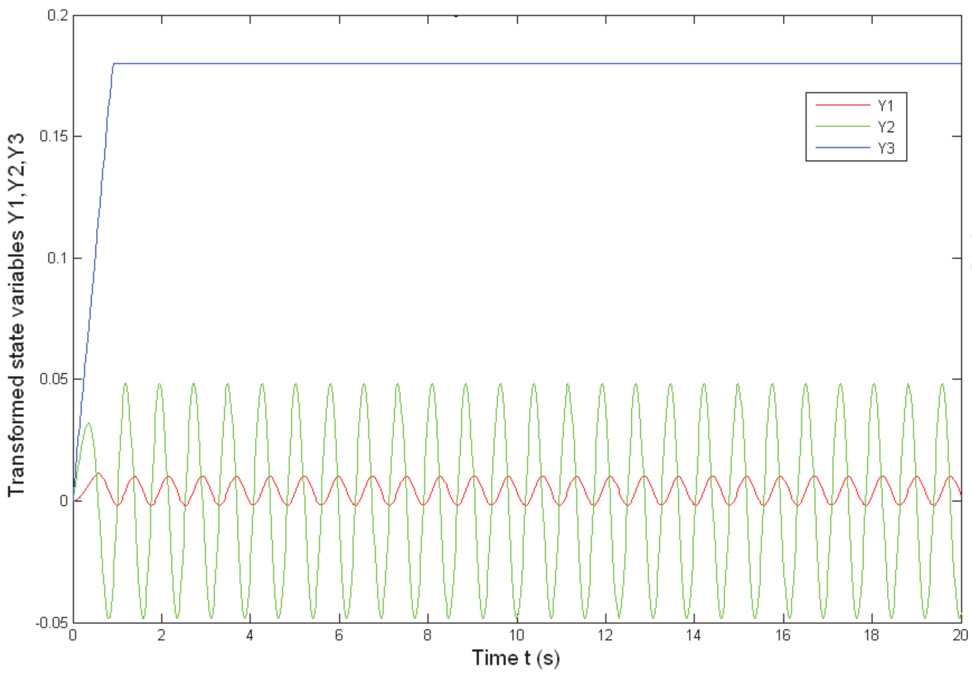

Fig. 2. Time response of Y_1 ,Y_2 ,Y_3 with v_1=0.2 and v_2=0.2

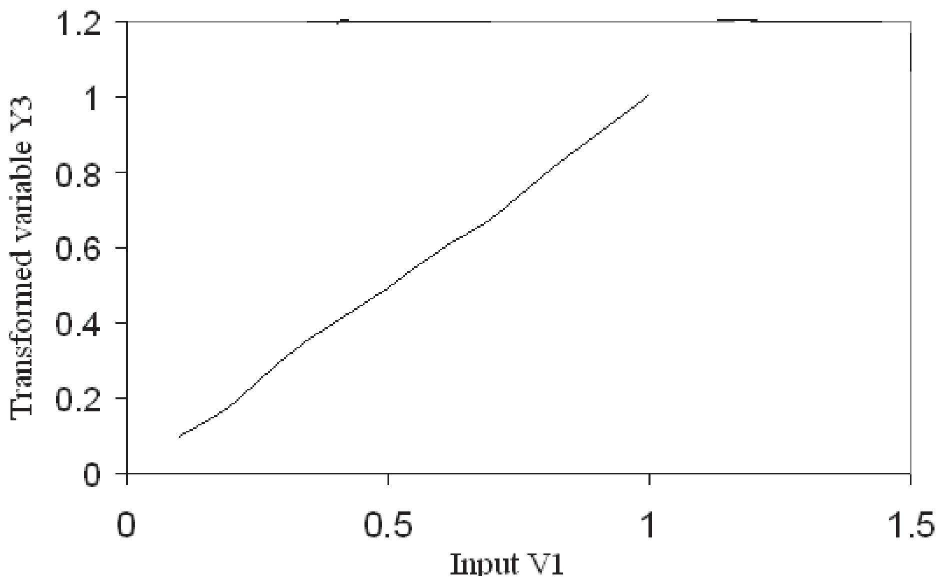

Fig. 3. Variation of transformed variable Y_3 with input v_1 (keeping v_2=0.1)

Fig. 3 shows the steady state gain of y_3 with respect to v_1 while v_2 is maintained constant. It is observed that the plot between y_3 and v_1 is almost linear, thus verifying that the linearizing system is almost linear. A similar test before linearization revealed that the steady gain of x_3 with respect to input u_1 varied over a large range thus revealing the nonlinearity.
Thus the linearization of the dynamic model of PMSM was verified through the use of MATLAB function ODE45 .

3. Linearization of PMSM machine model

3.1 PMSM model in normal form

The 4 – dimensional PM machine model can be derived as below (Bose 2002).

$$\begin{aligned}
&\dot{x} = Ax + Bu + f^{(2)}(x) \\
&x = [x_1 \quad x_2 \quad x_3 \quad x_4]^T = [\theta \quad \omega_e, \quad i_q, \quad i_d]^T ; \\
&u = [u_1 \quad u_2]^T = [v_q \quad v_d]^T
\end{aligned} \tag{7}$$

where v_q, v_d, i_q, i_d represent the quadrature and direct axis voltages and currents respectively and θ, ω_e represent rotor position and angular velocity respectively.

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & \frac{1.5p\lambda}{J} & 0 \\ 0 & \frac{-\lambda p}{L_q} & \frac{-R}{L_q} & 0 \\ 0 & 0 & 0 & \frac{-R}{L_d} \end{bmatrix} ; B = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ \frac{1}{L_q} & 0 \\ 0 & \frac{1}{L_d} \end{bmatrix} ; \; f^{(2)}(x) = \begin{bmatrix} 0 \\ \frac{1.5p(L_d - L_q)i_d i_q}{J} \\ \frac{-L_d p \omega_e i_d}{L_q} \\ \frac{L_q p \omega_e i_q}{L_d} \end{bmatrix}$$

λ is the flux induced by the permanent magnet of the rotor in the stator phases. L_d, L_q are the direct and quadrature inductances respectively. R is the stator resistance. p is the number of pole pairs and J is the system moment of inertia.
The model (7) can be reduced to normal form for two inputs (Brunovsky 1970), in a standard way using the following transformations (Kuo 2001),

$$x = \begin{bmatrix} a_1 c_1 & 0 & 0 & 0 \\ 0 & a_1 c_1 & 0 & 0 \\ 0 & 0 & c_1 & 0 \\ 0 & 0 & 0 & a_4 \end{bmatrix} y$$

$$u = \begin{bmatrix} 1 & 0 \\ 0 & a_4 / c_2 \end{bmatrix} u' + \begin{bmatrix} 0 & -a_1 a_2 & -a_3 & 0 \\ 0 & 0 & 0 & -a_4^2 / c_2 \end{bmatrix} y \tag{8}$$

where

$$a_1 = \frac{1.5 p \lambda}{J};\ a_2 = \frac{-\lambda p}{L_q}, a_3 = \frac{-R}{L_q},\ a_4 = \frac{-R}{L_d};\ c_1 = \frac{1}{L_q}; c_2 = \frac{1}{L_d}$$

The Brunovsky form for two inputs is given below (where x, u, A and B are retained for simplicity of notation).

$$\overset{\bullet}{x} = Ax + Bu + f^{(2)}(x) \tag{9}$$

Where

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}; B = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}; f^{(2)}(x) = \begin{bmatrix} 0 \\ k_1 x_3 x_4 \\ k_2 x_2 x_4 \\ k_3 x_2 x_3 \end{bmatrix}$$

$$k_1 = \frac{1.5 p (L_d - L_q) a_4}{J a_1}, k_2 = \frac{-L_d p a_1 a_4}{L_q}, k_3 = \frac{L_q {C_1}^2 a_1}{L_d a_4}$$

3.2 Linearization of PMSM normal form model

Given the 4 dimensional model of a PM synchronous motor (IPM model) of the form (9), the system can be linearized using the following transformations

$$y = x + \phi^{(2)}(x) \tag{10}$$

$$u = (I_2 + \beta^{(1)}(x))v + \alpha^{(2)}(x) \tag{11}$$

where $u=[u_1 \quad u_2]^T; v=[v_1 \quad v_2]^T$

$$\phi^{(2)}(x)=\begin{bmatrix}0\\0\\k_1x_3x_4\\0\end{bmatrix},\alpha^{(2)}(x)=\begin{bmatrix}-k_2x_2x_4\\-k_3x_2x_3\end{bmatrix},\beta^{(1)}(x)=-\begin{bmatrix}k_1x_4 & k_1x_3\\0 & 0\end{bmatrix}$$

where I_2 is the identity matrix of order 2. The system then reduces to

$$\dot{y}=Ay+Bv+O^{(3)}(y,v)$$

where $O^{(3)}(y,v)$ represents third and higher order nonlinearities .

4. Linearization of PMSM model using SIMULINK

4.1 Construction of PMSM model

Customization of PMSM machine model using MATLAB/SIMULINK tools had to be carried out as the standard library available contained only special cases. The PM motor drive simulation was built in several steps through the construction of q-axis circuit, d-a xis circuit, torque block and speed block.

4.1.1 q-axis circuit

By using the following system equation the q-axis circuit is constructed.

$$v_q=R_qi_q+\omega_r(L_di_d+\lambda_f)+\rho L_qi_q$$

q-axis circuit in the SIMULINK is shown in Fig.4 .

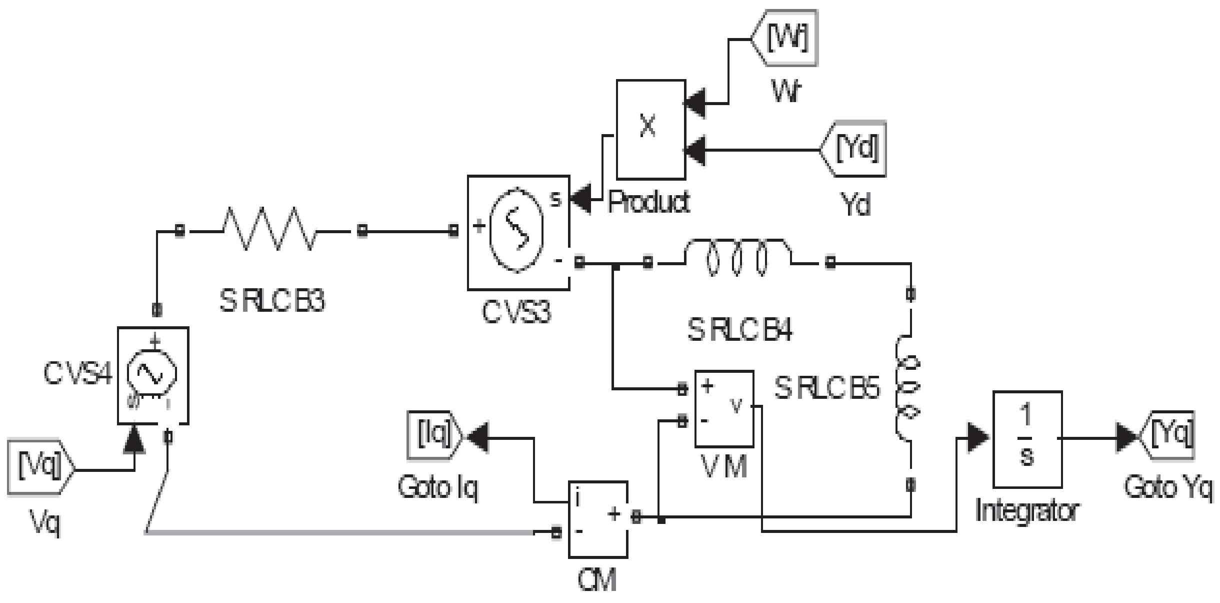

Fig. 4. q-axis circuit

4.1.2 d-axis circuit

By using the following system equation the d-axis circuit is constructed.

$$v_d = R_d i_d + \omega_r (L_q i_q) + \rho(\lambda_f + L_d i_d)$$

Simulation of the d-axis circuit is shown in Fig. 5.

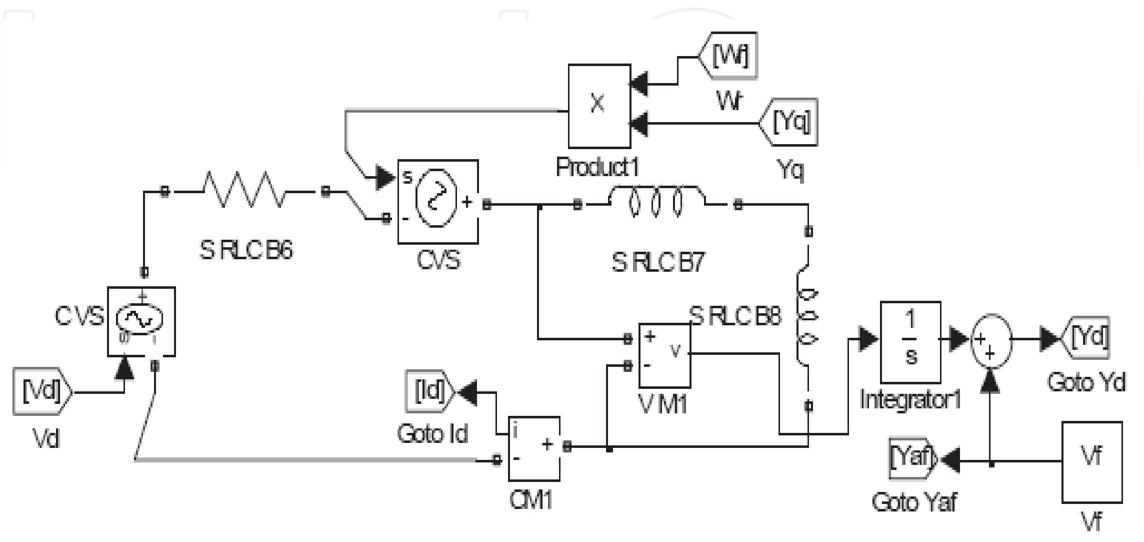

Fig. 5. d-axis circuit

4.1.3 Torque block

By using the following torque equation the torque block is constructed.

$$T_e = (3/2)(p/4)(\lambda_d i_q - \lambda_q i_d)$$

The simulation of torque T_e circuit is shown in the Fig 6.

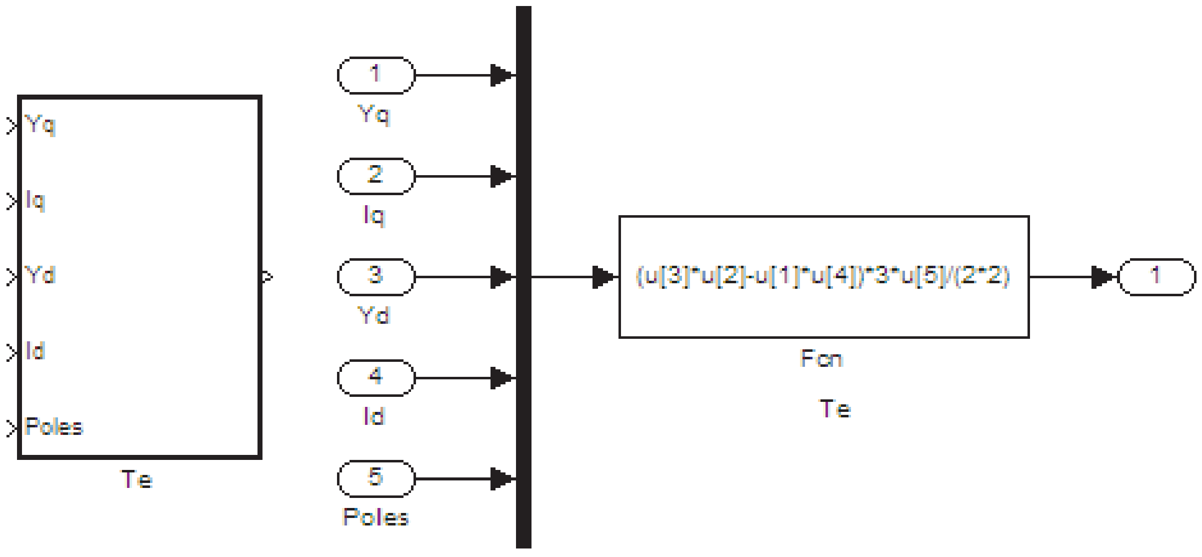

Fig. 6. T_e Block in Simulink

4.1.4 Speed block

By using the following equations the speed block is constructed.

$$\omega_m = \int (T_e - T_l - B\omega_m)/J dt$$

$$\omega_m = \omega_r (2/p)$$

The simulation of ω_m circuit is shown in the Fig.7.

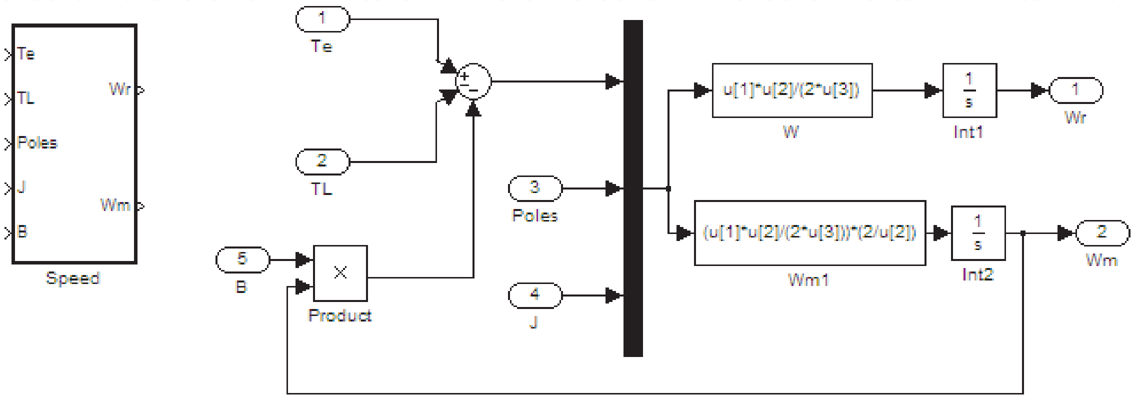

Fig. 7. Speed Block in Simulink

4.1.5 Designed PMSM with V_q and V_d as inputs

The PMSM is constructed by using q-axis, d-axis, torque and speed blocks (figures 4,5, 6, and 7) and is shown in figure 8.

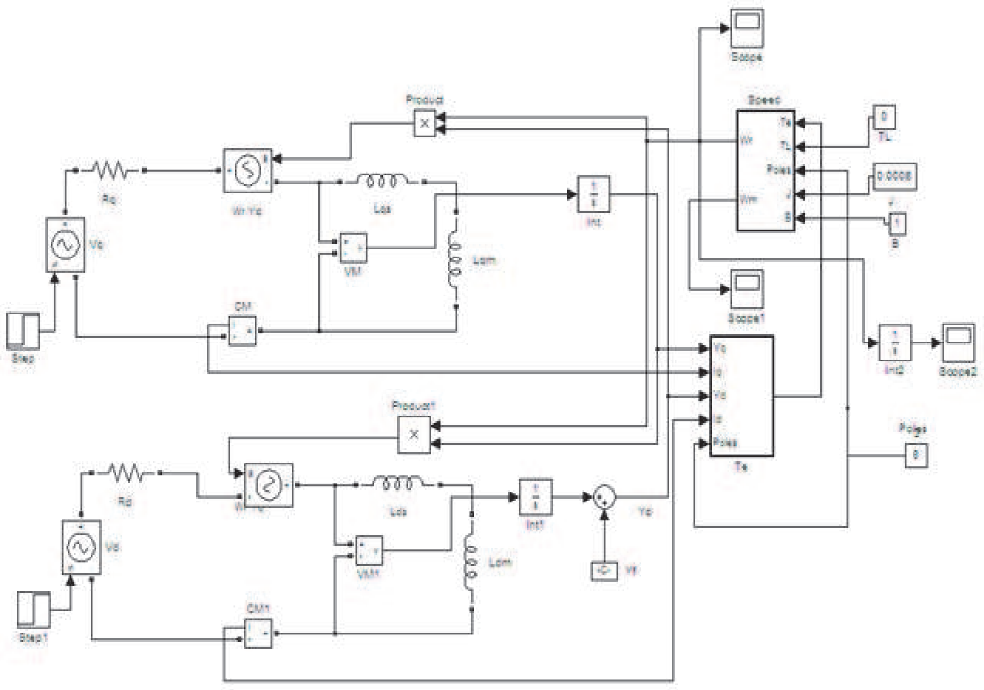

Fig. 8. Permanent Magnet Synchronous Motor with V_q and V_d as inputs

4.2 Linearization of PMSM SIMULINK model

The PMSM model is first converted to controller normal form of the linear part (9) using transformations as given in (8). Then linearization of the PMSM is carried out using quadratic coordinate and state feedback as given in (10) and (11). Fig.9 gives the PMSM model including linearization blocks. N_1 and N_2 include the linear transformations (8) while L_1 and L_2, denote the nonlinear transformations (11) and (10) respectively. N_1, N_2, L_1 and L_2 are implemented using function blocks.

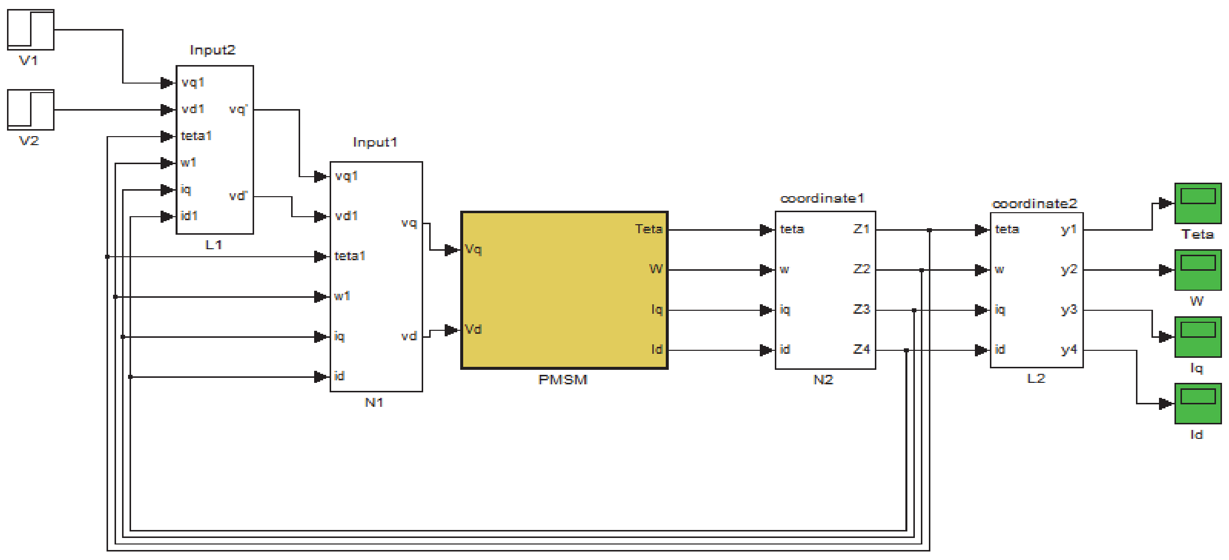

Fig. 9. Linearization of PMSM model

4.3 Verification of linearization of PMSM-simulation results

For the Interior PMSM, parameters are taken as follows:
Stator resistance R = 2.875Ω ; q- axis Inductance L_q = 9mH; d-axis Inductance L_d = 7mH;
Flux induced in magnets λ_f = 0.175 wb; Moment of Inertia J = 0.0008 kg.m^2
Friction factor B = 1 N.m.s; No. of pole pairs p = 4

v_q	ω_e	K=d ω_e /d v_q
5	2.55	-
10	4.7709	0.44418
15	6.4706	0.33994
20	7.6082	0.22752
25	8.2567	0.1297
30	8.5326	0.05518

Table 1. Steady state gain of ω_e versus v_q for the system in open loop (Prior to linearization)

v_1	y_2 *10^(-6)	K=d y_2 /d v_1 *10^(-6)
5	2.176	-
10	4.366	0.438
15	6.585	0.4438
20	8.845	0.452
25	11.162	0.4634
30	13.55	0.4776

Table 2. Steady state gain of y_2 vs v_1 for the linearized system in open loop

Prior to linearization, the open loop steady state gain of ω_e versus v_q of the PMSM model is investigated and the results are given in Table 1. In Table 1, it is observed that the open loop steady state gain of ω_e versus v_q (keeping v_d constant) is not constant because of the system nonlinearity. To verify the linearity of the system after linearization, we investigated the variation of its gain of y_2 (a scaled version of ω_e as can be seen from (8) and (10)) with input v_1 (see Fig. 9) and the results are given in Table 2. The table reveals that the gain of the system is nearly constant thus verifying that by applying the homogeneous linearizing transformation, the PMSM model is made nearly linear for the given set of inputs.

Figures 10 and 11 show the time response of angular velocity ω_e by closing the loop around PMSM model (Fig. 8) before linearization when v_q = 5 units and 30 units respectively. It is observed that the dynamic response for v_q = 5 is more oscillatory compared to the case of v_q = 30 with a fixed controller of proportional gain = 50 and integral constant =2. This is to be expected since the loop gain is higher in the former case with a higher static gain in the plant or motor as can be seen from Table I.

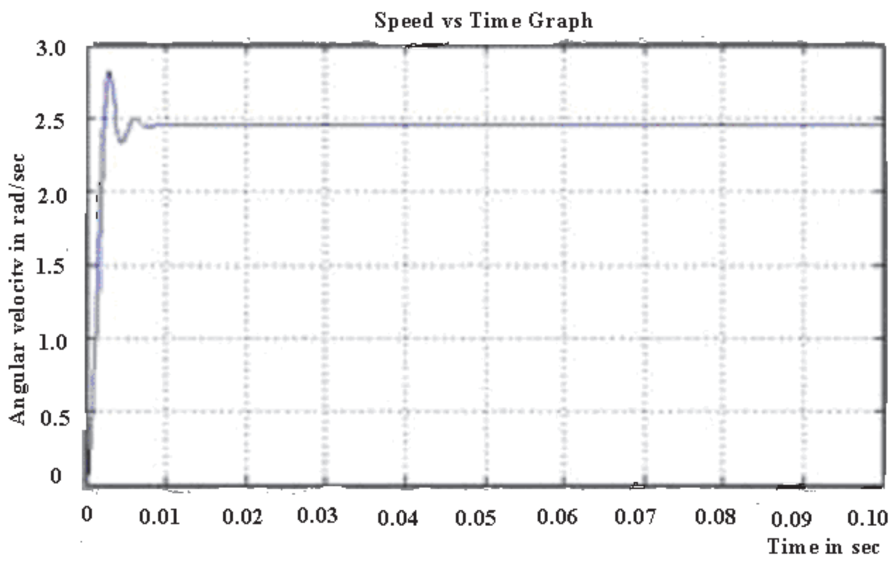

Fig. 10. Time response of angular velocity in closed loop when v_q = 5; $k_p = 50; k_i = 2$ (before linearization)

Figures 12 and 13 show the time response of y_2 of the transformed PMSM system (Fig. 9) in closed loop when v_1 = 5 units and 30 units respectively . It is observed that a uniform output response is obtained in the closed loop after linearization when the reference is varied. Since the static gain in Table II is nearly uniform, the loop gain is also nearly constant for the extreme points in the operating range, thus resulting in the uniform dynamic responses in Fig. 12 and 13.
Simulation results show that the nearly constant gain of the linearized model, results in a uniform response on a range of set point and load inputs with a fixed controller. This is in contrast to the case before linearization under the corresponding conditions.

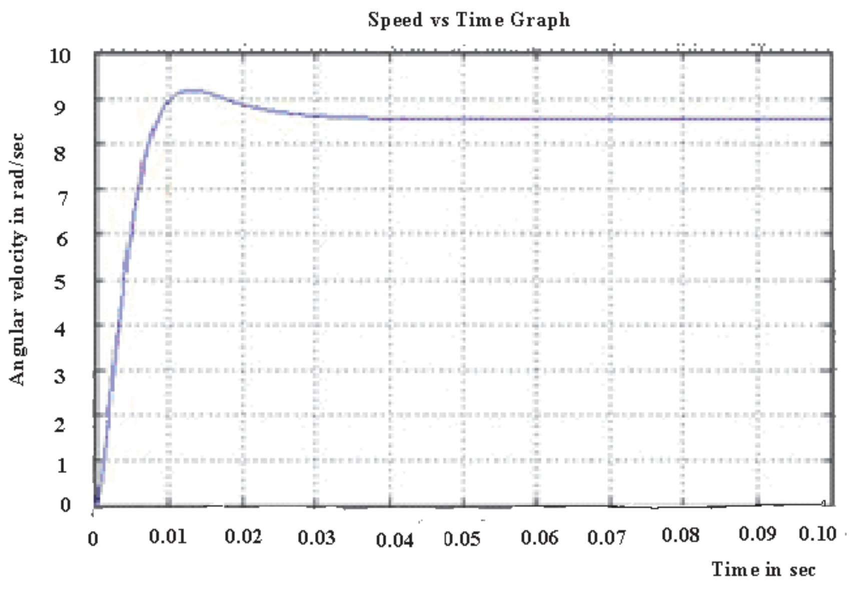

Fig. 11. Time response of angular velocity in closed loop when v_q = 30; $k_p = 50; k_i = 2$ (before linearization)

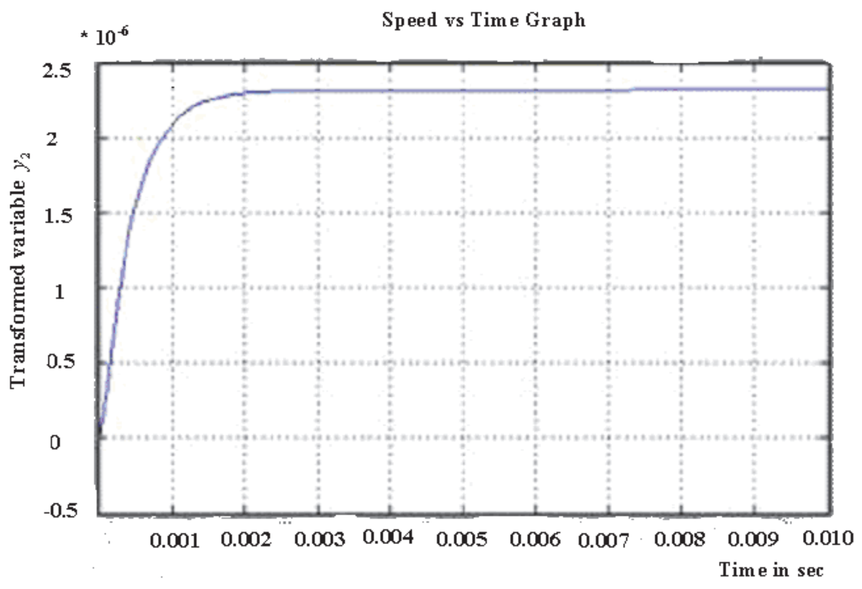

Fig. 12. Time Response of y_2 for the linearized system in closed loop when v_1 = 5; $k_p = 50; k_i = 2$

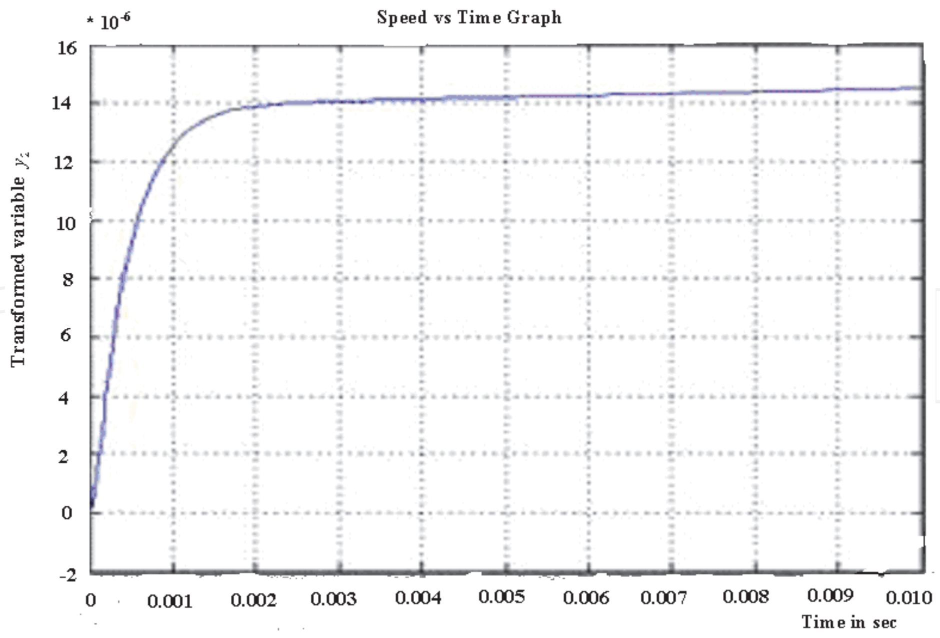

Fig. 13. Time Response of y_2 for the linearized system in closed loop when v_1 = 30; $k_p = 50; k_i = 2$

5. Tuning linearizing transformations

Unmodelled dynamics coupled with the third and higher order nonlinearities introduced due to quadratic linearization, are best accounted for by tuning the transformations (Levin & Narendra 1993).

To further improve the linearity of the system taking into account unmodelled dynamics and higher order nonlinearities, tuning of the transformation parameters against an actual PM machine is done on the lines similar to those proposed by Narendra (Levin & Narendra 1993). Fig 14 shows the block diagram for tuning.

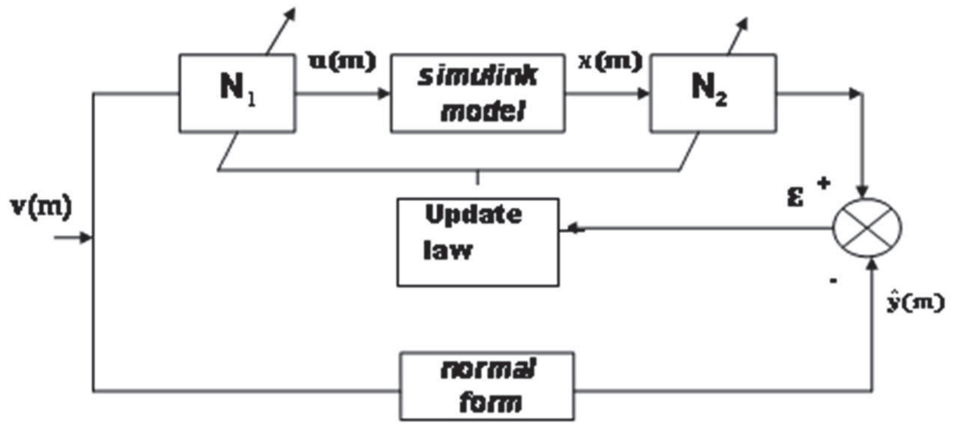

Fig. 14. Block diagram for tuning of transformation

Referring to Fig. 14, error (E) can be calculated as

$$E = (\varepsilon^T \varepsilon) = \left[(y - \hat{y})^T (y - \hat{y}) \right]^{1/2} \tag{12}$$

where $\varepsilon = (\varepsilon_1 \quad \varepsilon_2 \quad \varepsilon_3 \quad \varepsilon_4)^T$; $\varepsilon_i = y_i - \hat{y}_i; i = 1,2,3,4$

The error can be written as

$$E = \left(\varepsilon_1^2 + \varepsilon_2^2 + \varepsilon_3^2 + \varepsilon_4^2\right)^{1/2}$$

Since $\phi^{(2)}(x)$ and $\beta^{(1)}(x)$ are both functions of k_1, we shall redefine

$$\phi^{(2)}(x) = \begin{bmatrix} 0 \\ 0 \\ k_1 x_3 x_4 \\ 0 \end{bmatrix}$$

and

$$\beta^{(1)}(x) = -\begin{bmatrix} k_1' x_4 & k_1' x_3 \\ 0 & 0 \end{bmatrix}$$

so that $\phi^{(2)}(x)$ and $\beta^{(1)}(x)$ can be independently tuned by tuning k_1 and k_1' respectively and $\alpha^{(2)}(x)$ is not varied.

5.1 Updation of N_2 transformation coefficients

Tuning of N_2 transformation implies the tuning of $\phi^{(2)}(x)$. As $\phi^{(2)}(x)$ is a function of only $k_1 x_3 x_4$, the coefficient k_1 has to be updated based on the error between the outputs of quadratic linearized system and normal form. The updation law is derived as follows.

$$\Delta k_1 = \frac{\partial E}{\partial k_1} = \frac{\partial E}{\partial y}\frac{\partial y}{\partial k_1}$$

From (12), it is seen that

$$\frac{\partial E}{\partial y_i} = \frac{\varepsilon_i}{E}; i = 1,2,3,4$$

Hence

$$\frac{\partial E}{\partial y} = \begin{bmatrix} \frac{\varepsilon_1}{E} & \frac{\varepsilon_2}{E} & \frac{\varepsilon_3}{E} & \frac{\varepsilon_4}{E} \end{bmatrix}.$$

$$\therefore \Delta k_1 = \begin{bmatrix} \frac{\varepsilon_1}{E} & \frac{\varepsilon_2}{E} & \frac{\varepsilon_3}{E} & \frac{\varepsilon_4}{E} \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ x_3 x_4 \\ 0 \end{bmatrix} = \frac{\varepsilon_3}{E} x_3 x_4 \qquad (13)$$

Updation of $\phi^{(2)}(x)$ is done by using the formula:

$$k_1(m) = k_1(m-1) - \rho\Delta k_1(m); 0 < \rho < 1 \tag{14}$$

where m corresponds to the updating step and ρ correspond to the accelerating factor.

5.2 Updation of *N₁* transformation coefficients

Tuning of N_1, transformation is achieved by tuning of $\beta^{(1)}(x)$. As $\beta^{(1)}(x)$ is a function of $k_1' x_3$ and $k_1' x_4$, the coefficient k_1' has to be updated based on the error between the outputs of quadratic linearized system and normal form . The updation law is derived as follows.

$$\Delta k_1' = \frac{\partial E}{\partial k_1'} = \frac{\partial E}{\partial y}\frac{\partial y}{\partial x}\frac{\partial x}{\partial u_1}\frac{\partial u_1}{\partial k_1'}$$

where ; $\frac{\partial E}{\partial y} = \begin{bmatrix} \frac{\varepsilon_1}{E} & \frac{\varepsilon_2}{E} & \frac{\varepsilon_3}{E} & \frac{\varepsilon_4}{E} \end{bmatrix}$

$$\frac{\partial y}{\partial x} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & (1+k_1' x_4) & k_1' x_3 \\ 0 & 0 & 0 & 1 \end{bmatrix}; \frac{\partial x}{\partial u_1} = \begin{bmatrix} 0 \\ -1/k_2 x_4 \\ 0 \\ -1/k_2 x_2 \end{bmatrix}$$

Assuming that the steady state of the Simulink model is reached within the tuning period.

$$\frac{\partial u_1}{\partial k_1'} = -v_1 x_4 - v_2 x_3$$

Hence

$$\Delta k_1' = \frac{(v_1 x_4 + v_2 x_3)}{E}\left(\frac{\varepsilon_2}{k_2 x_4} + \frac{\varepsilon_3 k_1' x_3 + \varepsilon_4}{k_2 x_2}\right) \tag{15}$$

Tuning of the quadratic linearizing transformations is done by updating the transformation coefficients of vector polynomials $\phi^{(2)}(x)$ and $\beta^{(1)}(x)$.

5.3 Construction of controller tuning blocks

Updation of $\phi^{(2)}(x)$ is done by using (14) and Δk_1 can be obtained from (13).
The tuning is done using Memory blocks and they are constructed in simulink as shown. Fig. 15 shows the construction of Del k_1 block. Similar construction can be done for Del k_1' where $\Delta k_1'$ can be obtained from (14).
Simulation of updation of $\phi^{(2)}(x)$ and $\beta^{(1)}(x)$ can be done using the simulation diagrams Fig. 15, 16 and 17.

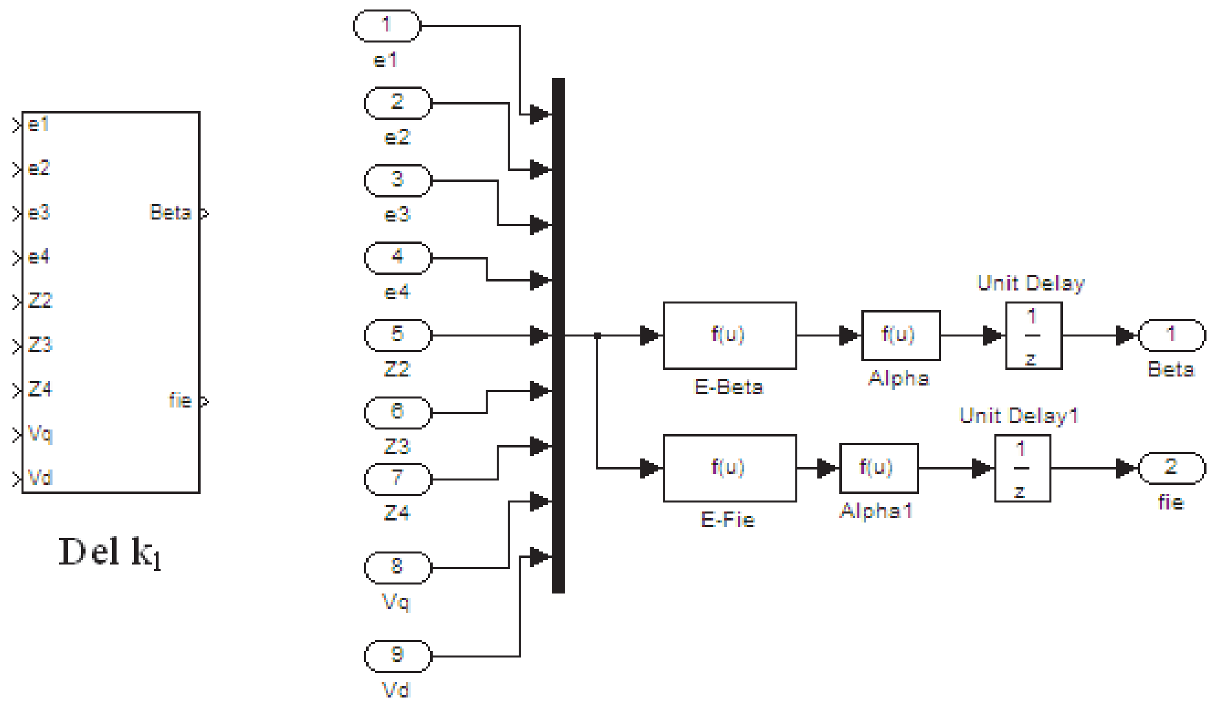

Fig. 15. Calculation of del K_1 for the updation of $\phi^{(2)}(x)$

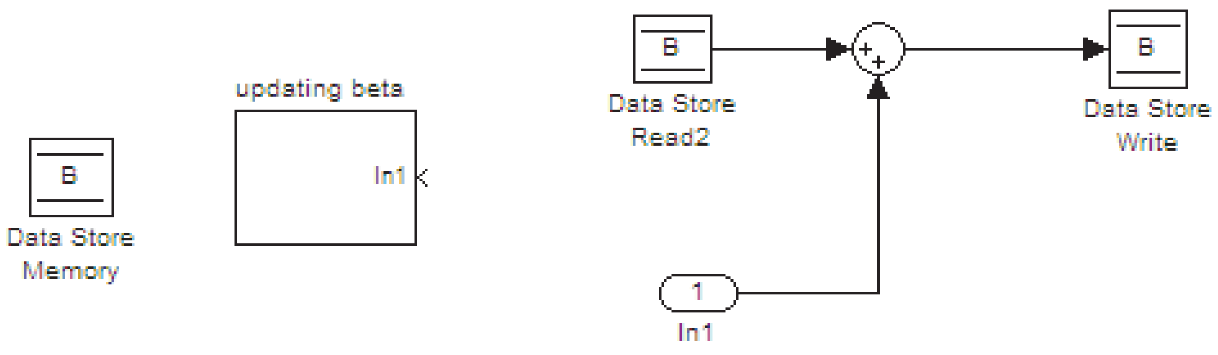

Fig. 16. Updation rule using Memory Read and memory Write Blocks

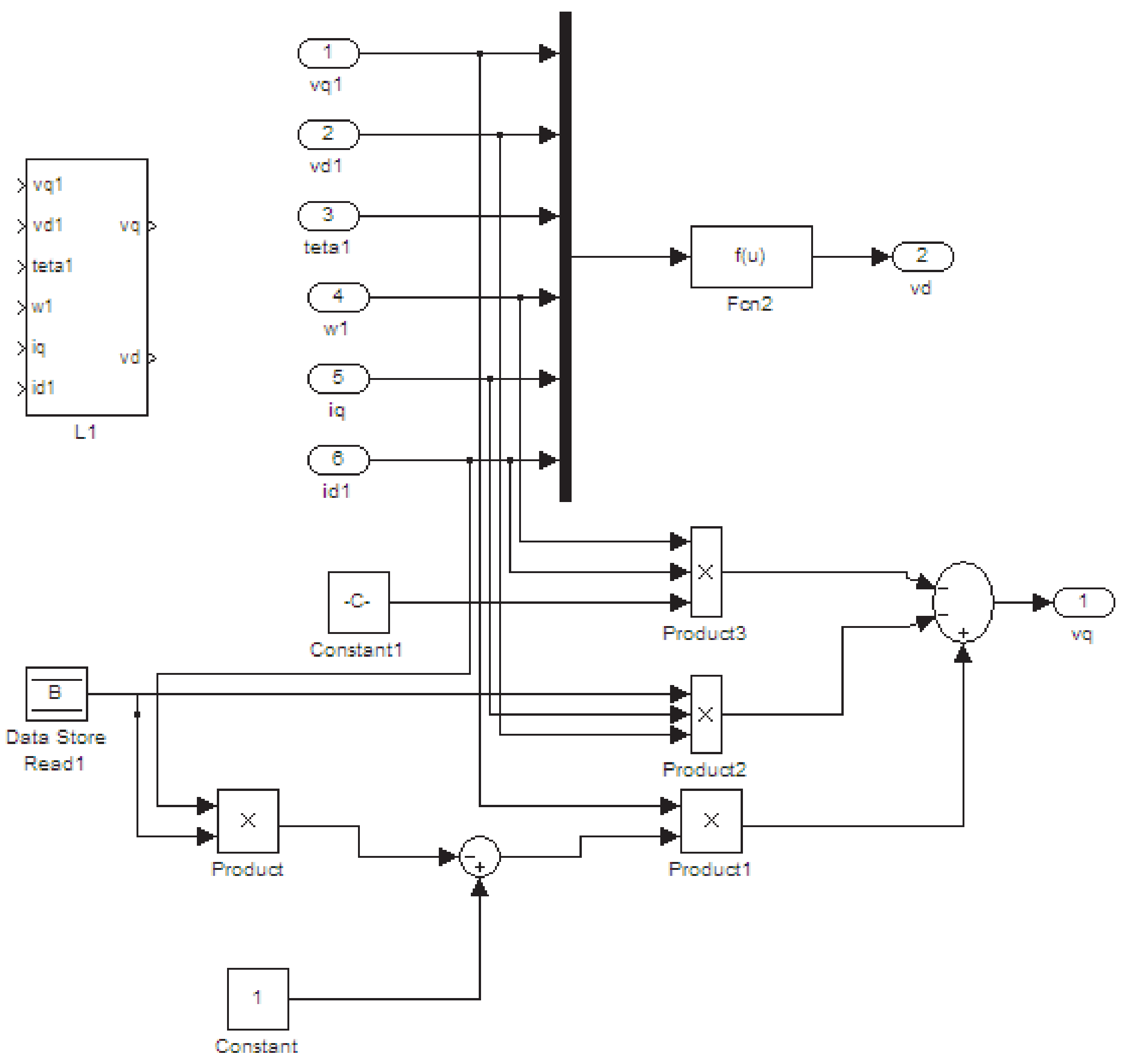

Fig. 17. Updation of input transformation

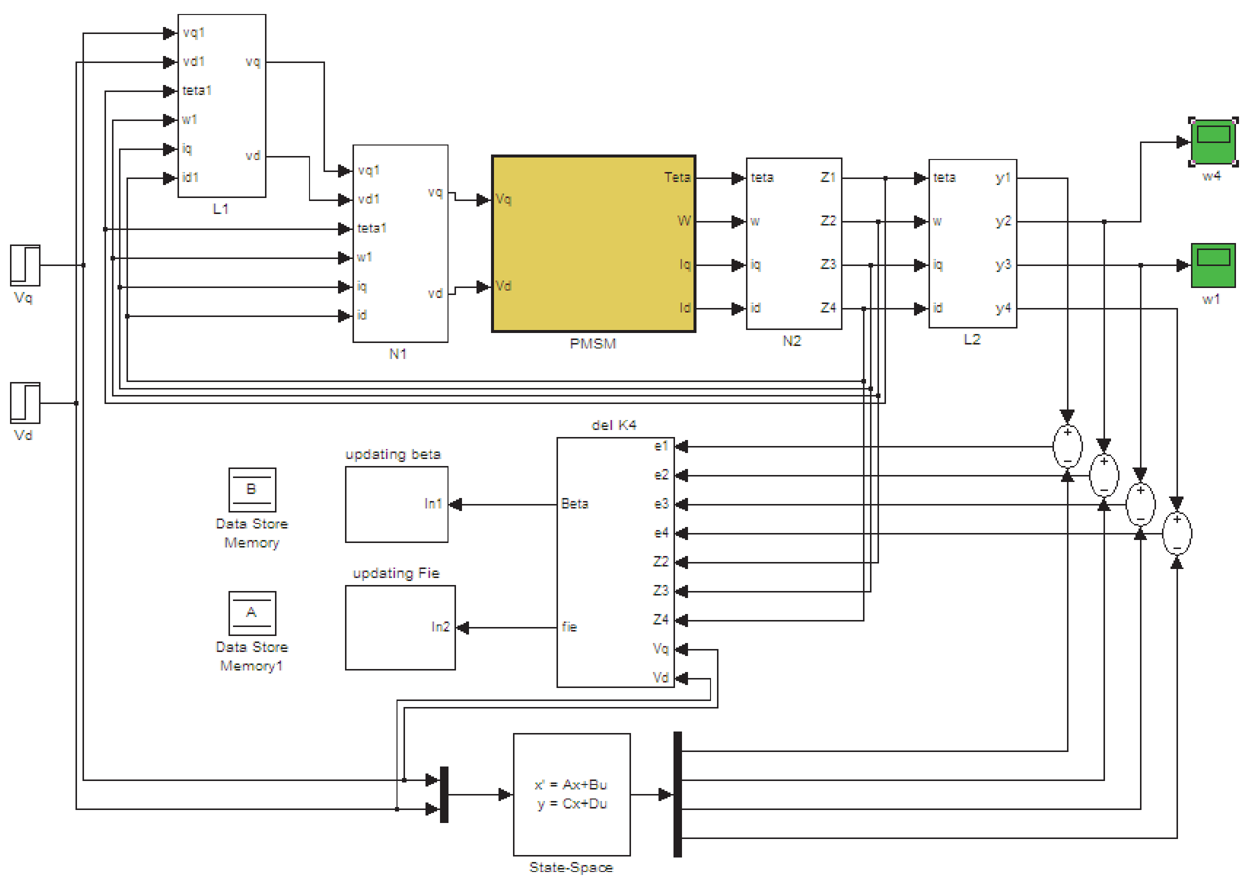

Fig. 18. Simulation diagram including controller tuning

v_1	y_2 *10^-6	K=d y_2 / d v_1 *10^(-6)
106	7.142	--
108	7.38	0.048
110	7.334	0.048
112	7.43	0.048
114	7.525	0.0475
116	7.62	0.0475
118	7.714	0.047

Table 3. Steady state gain of y_2 versus v_1 for the linearized system in open loop after tuning

Complete simulation diagram including conversion to Brunovsky form, linearization and tuning is given in Fig.18. It is seen that the error after tuning is reduced to 0.01.
In Table 3, improvements are obtained for the steady state gain of y_2 versus v_1 for the linearized system after incorporating tuning of the transformation parameters. Table 3 reveals that the gain of the system is even more constant compared to that shown in Table 2, thus verifying that by tuning the homogeneous linearizing transformation, the linearity of the system has been improved for the given set of inputs.

6. Conclusion

Application of MATLAB and SIMULINK tools for the verification of linearization of permanent magnet synchronous motor is considered in this chapter. Simulation is done using the dynamic model of PMSM, application of nonlinear coordinate and state feedback transformations to the SIMULINK model which is customized to PMSM and tuning the transformations against a linear system model employing error back propogation to account for unmodelled dynamics.
Initially, linearization of PMSM is verified using the dynamic model of PMSM. The dynamic model of a PM synchronous motor involving quadratic nonlinearity is linearized and simulated using MATLAB. Steps are given to perform dynamic simulation for the nonlinear system using the dynamic equations based on parameters of the machine, together with the state and input transformations using MATLAB function ODE45.
The SIMULINK model of Interior Permanent Magnet machine is specially developed by integrating various blocks as standard library functions do not cater to generic purposes . The PMSM machine model, together with the state and input transformations, are simulated using SIMULINK. The simulation results show that the linearizing transformations effectively linearize the system thus supporting the theory.
To account for the unmodelled dynamics and third and higher order nonlinearities, tuning of the transformation parameters is done by comparing the output of the linearized system with a normal form output. Tuning the transformation functions $\phi^{(2)}(x)$ and $\beta^{(1)}(x)$ is shown further to improve the linearity of the resulting system.
More details of the simulation results are given in (Parvathy et. al. 2011).

7. Acknowledgement

The authors acknowledge the financial support rendered by the management of Hindustan Institute of Technology and Science, Chennai in publishing this chapter.

8. References

Bimal .K. Bose(2002) *Modern Power Electronics and Ac Drives* , Pearson Education.

Gildeberto S. Cardoso & Leizer Schnitman (2011) Analysis of exact linearization and approximate feedback linearization techniques. *www.hindawi.com/journals/mpe/aip/205939.pdf*

Krener A.J. (1984) Approximate linearization by state feedback and coordinate change. *Systems and control Letters.* Vol. 5,pp. 181-185.

John Chiasson& Marc Bodson (1998) .Differential – Geometric methods for control of lectric motors. *International Journal of Robust and Nonlinear Control.* No.8, pp.923-954.

Arnold V.I.(1983) *Geometric methods in the theory of ordinary differential equations.* Springer-Verlag, NewYork, pp 177-188.

Kang .W & Krener.A.J. (1992) . Extended quadratic controller normal form and dynamic state feedback linearisation of nonlinear systems. *SIAM Journal of Control and optimization* . No.30,pp.1319-1337.

Devanathan R. (2001) ._Linearisation Condition through State Feedback. *IEEE transactions on Automatic Control* .ol 46,no.8,pp.1257-1260

Devanathan R. (2004). Necessary and sufficient conditions for quadratic linearisation of a linearly controllable system. *INT.J. Control*, Vol.77, No.7,pp. 613-621

Benjamin.C.Kuo(2001) . *Automatic Control Systems.* Prentice-Hall India.

A.K.Parvathy, Aruna Rajan, R.Devanathan (2005) . Complete Quadratic Linearization of PM Synchronous Motor Model. , *proceedings of NEPC conference.* IIT Karagpur, pp 49-52.

A.K.Parvathy, R.Devanathan (2006) Linearisation of Permanent Magnet Synchronous Motor Model.*proc.of IEEE ICIT 2006*,pp. 483-486.

Pavol Brunovsky (1970) *A classification of linear controllable systems.* Kybernetika,Vol. 6. No.3, pp. 173-188

Asriel U Levin & Kumpati S Narendra (1993). Control of nonlinear dynamical systems using neural networks: controllability and stabilization. *IEEE Transactions on neural networks.*Vol-4, No.2,1993, pp.192-206.

A.K.Parvathy, V.Kamaraj & R.Devanathan (2011) A generalized quadratic linearization technique for PMSM. accepted for publication in , *European Journal of Scientific Research.*