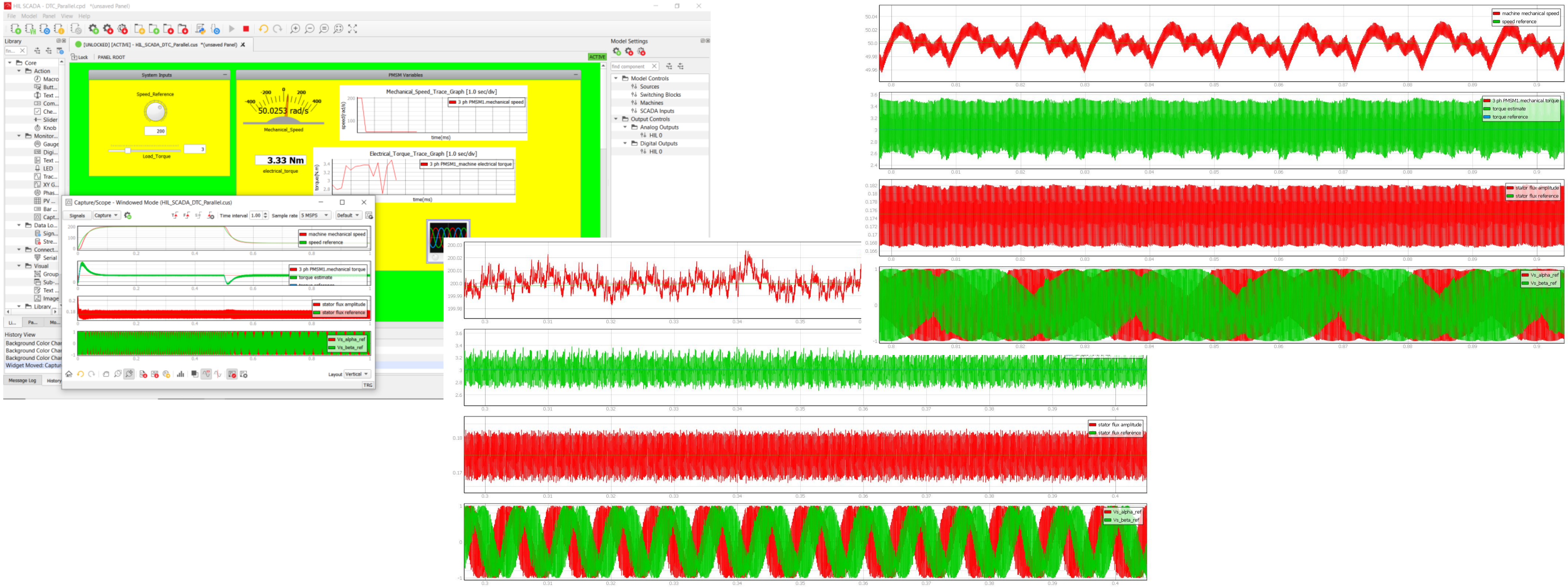


PMSM Speed Control Based on Direct Torque Control with Space Vector Modulation(DTC-SVM)

Boboye Oladosu Okeya (BOO)

What is DTC-SVM

- Direct torque control with space vector modulation (DTC-SVM) is a control strategy for electrical machines which applies parallel and cascade PI controllers for directly controlling the machine's electromagnetic torque and stator flux in the synchronously rotating reference frames [1].
- There are two basic structures for DTC-SVM[2]
  - Parallel structure
  - Cascade structure
- **All useful information about the dynamic equations and controller designed are gotten from the equations in [1].**

PMSM Equation in (d-q) &(x-y) reference frame [1]

$$v_d = r_s i_d - \omega_r \lambda_q + \dot{\lambda}_d \quad (1)$$

$$v_q = r_s i_d - \omega_r \lambda_d + \dot{\lambda}_q \quad (2)$$

$$\lambda_d = L_d i_d + \lambda'_m \quad (3)$$

$$\lambda_q = L_q i_q \quad (4)$$

$$T_e = \frac{3}{2}\frac{P}{2}(\lambda_d i_q - \lambda_q i_d)\,, \quad (5)$$

$$J\dot{\omega}_m + B\omega_m + T_L = T_e \quad (6)$$

$$\omega_r = (\frac{P}{2})\omega_m\,, \quad (7)$$

where

r_s denotes the d-q axis winding resistance. λ_d, λ_q, L_d, L_q, i_d, i_q, v_d, and v_q are stator flux linkages, inductances, currents, voltages in direct- and quadrature-axis, respectively; λ'_m is the armature back EMF constant; ω_r and ω_m are the electrical and mechanical rotor speed; P denotes the number of motor poles. J, B, T_L, and T_e are the moment of inertia, friction coefficient, load torque and electromagnetic torque of the motor, respectively.

- **PMSM Reference frames & Transformations:**
  - The rotor rotating reference frame (d-q)
  - The stator rotating reference frame (x-y)
  - The stationary reference frame (α-β)

$$\begin{bmatrix} f_d \\ f_q \end{bmatrix} = \begin{bmatrix} \cos\delta & -\sin\delta \\ \sin\delta & \cos\delta \end{bmatrix} \begin{bmatrix} f_x \\ f_y \end{bmatrix} \quad (8)$$

where f represents the voltage, current, and flux linkage. load angle, δ, the angle between the stator and rotor flux linkages when the stator resistance is neglected

$$\sin\delta = \lambda_q / \lambda_s\,,\ \cos\delta = \lambda_d / \lambda_s \quad (9)$$

Substituting (8) and (9) for the current into (5) gives

$$T_e = \frac{3}{2}\frac{P}{2}|\lambda_s| i_y \quad (10)$$

Considering PMSM with uniform airqap, (3) and (4) can be simplified as (11) and (12), i.e., L_d=L_q=L_s.

$$\lambda_s = L_s i_x + \lambda'_m \cos\delta \quad (11)$$

$$0 = L_s i_y - \lambda'_m \sin\delta \quad (12)$$

Then, from (12), i_y can be obtained as

$$i_y = \frac{1}{L_s}\lambda'_m \sin\delta \quad (13)$$

Substituting (13) into (10) leads to

$$T_e = \frac{3P}{4}\frac{|\lambda_s|}{L_s}\lambda'_m \sin\delta \quad (14)$$

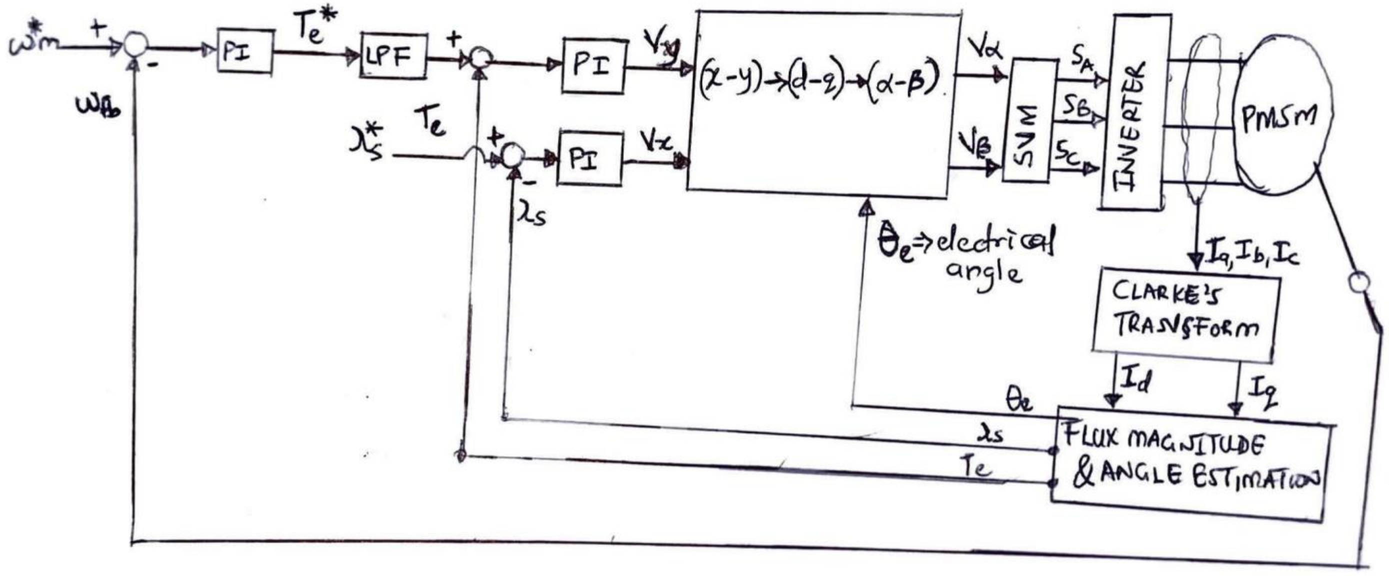
Similarly, (1) and (2) can be also transformed to stator flux (x-y) reference frame as below:

$$v_x = r_s i_x + \dot{\lambda}_s \quad (15)$$

$$v_y = r_s i_y + \dot{\delta}\lambda_s = r_s i_y + (\omega_s - \omega_r)\lambda_s \quad (16)$$

where δ=θ_s-θ_r and their corresponding speeds ($\dot{\delta}$=ω_s-ω_r) are applied. It is seen that v_x and v_y can directly affect the motor's flux and speed dynamics, respectively.

DTC-SVM Architecture [1]



Flux Magnitude and Angle Estimation

$$\lambda_d = L_d i_d + \lambda'_m \rightarrow \text{Permanent Magnet flux} \quad \text{— ③}$$

$$\lambda_q = L_q i_q \quad \text{— ④}$$

$$\lambda_s = \sqrt{\lambda_d^2 + \lambda_q^2}$$

$$T_e = \frac{3}{2}\left(\frac{P}{2}\right)\left(\lambda_d i_q - \lambda_q i_d\right) \quad \text{— ⑤}$$

$$\sin\delta = \lambda_q/\lambda_s \;\&\; \cos\delta = \lambda_d/\lambda_s$$

$$\tan\delta = \frac{\sin\delta}{\cos\delta} = \lambda_q/\lambda_d$$

$$\delta = \tan^{-1}(\lambda_q/\lambda_d)$$

Controller Design

Flux Controller Design

Assuming small change of δ i.e $\cos\delta = 1$

$$V_x = \tilde{r}_x \lambda_s + \dot{\lambda}_s - \tilde{r}_s \lambda'_m \quad -(17)$$

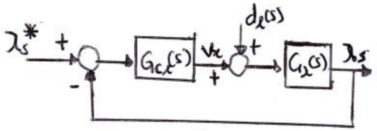
$$\tilde{r}_s = r_s / L_s$$

→ Open Loop Transfer function (OLTF):

$$G_\lambda(s) = \frac{\lambda_s(s)}{V_x(s) + d_\lambda(s)} = \frac{1}{s + \tilde{r}_s} \quad -(18)$$

where $d_\lambda(s) = \tilde{r}_s \lambda'_m$

→ PI Controller ⟹ $G_{c\lambda}(s) = K_{p\lambda} + \dfrac{K_{p\lambda}}{sT_{i\lambda}}$



→ Closed Loop Transfer function (CLTF):

$$G_1(s) = \frac{|\lambda_s|}{|\lambda_s^*|} = \frac{K_{p\lambda}(s + T_{i\lambda}^{-1})}{s^2 + (\tilde{r}_s + K_{p\lambda})s + K_{p\lambda}T_{i\lambda}^{-1}} \quad -(19)$$

→ PI parameter obtained through Pole placement:

$$(s+P)^2 = s^2 + P_1 s + P_2$$

$$P_1 = \tilde{r}_s + K_{p\lambda} \quad \therefore \quad \boxed{K_{p\lambda} = P_1 - \tilde{r}_s}$$

$$P_2 = \frac{K_{p\lambda}}{T_{i\lambda}} \quad \therefore \quad \boxed{T_{i\lambda} = \frac{K_{p\lambda}}{P_2}}$$

Torque Controller Design

Assuming small changes of δ i.e $\sin\delta = \delta$

$$i_y = \frac{\delta \lambda'_m}{L_s} = \frac{4}{3P|\lambda_s|} T_e \quad -(20)$$

$$V_y = \frac{r_s 4}{3P|\lambda_s|} T_e + \left(\frac{L_s 4}{\lambda'_m 3P|\lambda_s|}\frac{dT_e}{dt} + \frac{P}{2}\omega_m\right)|\lambda_s| \quad -(21)$$

Ignoring the effects of load torque T_L & friction coefficient

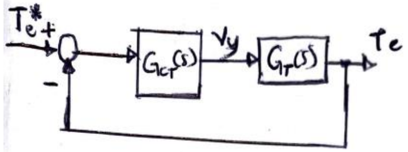
→ OLTF:

$$G_T(s) = \frac{T_e(s)}{V_y(s)} = \frac{A_1 s}{s^2 + B_1 s + C_1} \quad -(22)$$

where: $A_1 = \dfrac{3P\lambda'_m}{4L_s}$; $B_1 = \dfrac{\tilde{r}_s \lambda'_m}{|\lambda_s|}$; $C_1 = \dfrac{3P^2 \lambda'_m |\lambda_s|}{8JL_s}$

here, let $|\lambda_s| = \lambda_s^*$

→ PI Controller: $G_{cT}(s) = K_{pT} + \dfrac{K_{pT}}{sT_{iT}}$



→ CLTF:

$$G_2(s) = \frac{T_e(s)}{T_e^*(s)} = \frac{A_1 K_{pT}(s + T_{iT}^{-1})}{s^2 + (B_1 + A_1 K_{pT})s + A_1 K_{pT} T_{iT}^{-1} + C_1}$$

→ PI Parameter from Pole placement:

$$(s+k)^2 = s^2 + k_1 s + k_2$$

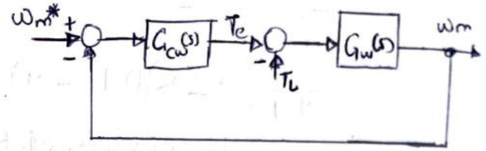
$$k_1 = B_1 + A_1 K_{pT} \quad \therefore \quad \boxed{K_{pT} = \frac{K_1 - B_1}{A_1}}$$

$$k_2 = A_1 K_{pT} T_{iT}^{-1} + C_1 \quad \therefore \quad \boxed{T_{iT} = \frac{A_1 K_{pT}}{K_2 - C_1}}$$

A torque reference prefilter $\left[G_{T_1}(s) = \dfrac{1}{s + T_{iT}^{-1}}\right]$ gives the torque loop CLTF a DC gain of

$$E = \frac{A_1 K_{pT}}{A_1 K_{pT} + C_1 T_{iT}}$$

Speed Controller Design



$$G_{c\omega} = K_{p\omega} + \frac{K_{p\omega}}{sT_{i\omega}} \rightarrow \text{PI Controller}$$

→ CLTF:

$$G_3(s) = \frac{\omega_m}{\omega_m^*} = \frac{E_1 K_{p\omega}(s + T_{i\omega}^{-1})}{s^2 + (a_1 + E_1 K_{p\omega})s + E_1 K_{p\omega} T_{i\omega}^{-1}}$$

$$a_1 = B/J \; ; \; E_1 = E/J$$

→ PI Parameters selection through Pole Placement

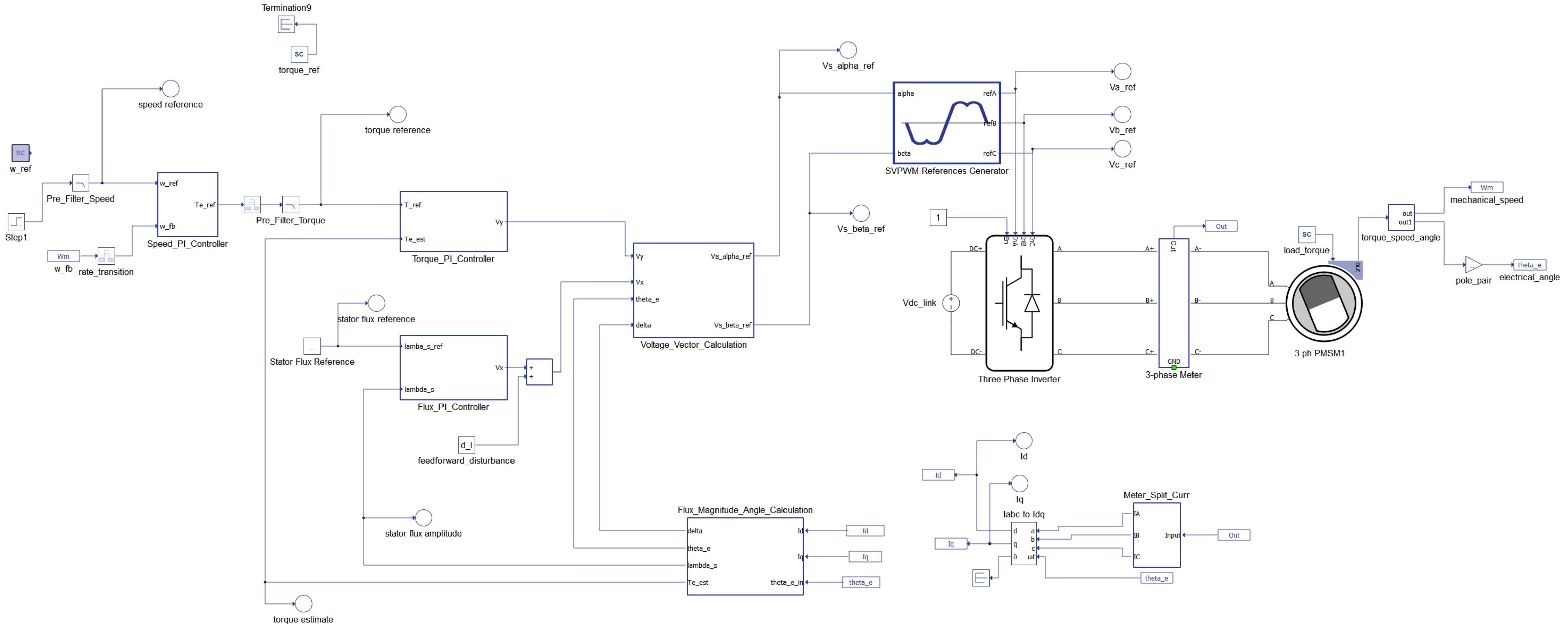
$$(s+n)^2 = s^2 + n_1 s + n_2$$

→ $a_1 + E_1 K_{p\omega} = n_1 \quad \therefore \quad \boxed{K_{p\omega} = \dfrac{n_1 - a_1}{E_1}}$

→ $n_2 = \dfrac{E_1 K_{p\omega}}{T_{i\omega}^{-1}} \quad \therefore \quad \boxed{T_{i\omega} = \dfrac{E_1 K_{p\omega}}{n_2}}$

→ A Speed Command Prefilter $\left[G_{T\omega}(s) = \dfrac{1}{s + T_{i\omega}^{-1}}\right]$ is used to reduce the speed overshoot.

Typhoon HIL Simulation in VHIL Mode



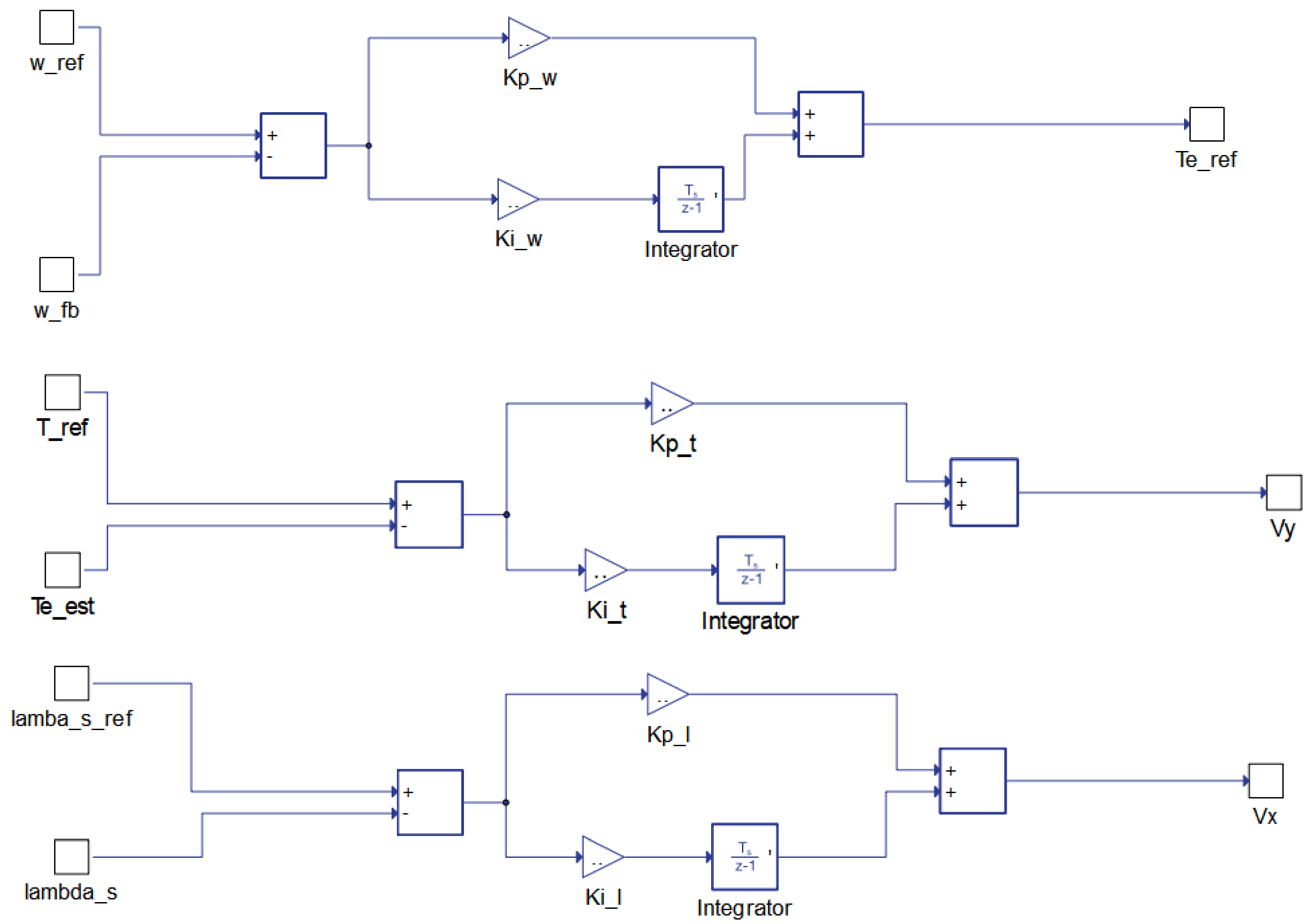
Simulation Parameters

Model initialization function

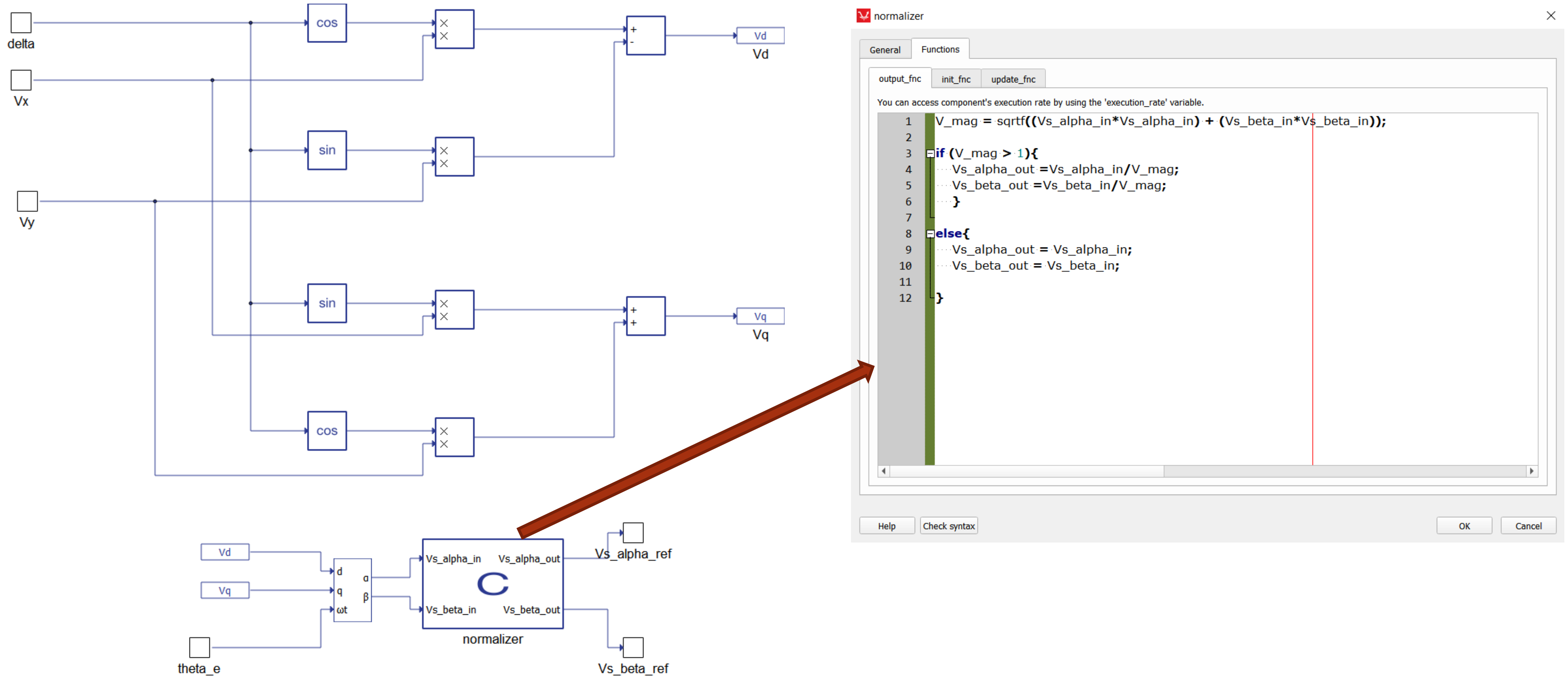
Here you can declare your variables that will be added into the namespace in the process of compilation.

```
#Power Stage & Inverter Parameters
Vdc = 500 #DC-link Voltage
fsw = 50e3 #Switching Frequency
dt = 1e-6 #inverter switch dead-time period

#Execution rates
Ts = 1/fsw #Sampling time of the inner current loop
T_inner = Ts
T_outer = 3*T_inner

#PMSM Parameters
Rs = 2.8750 #Stator Resistance-Ohms
Ls = 8.5e-3 #Stator d and q-axis inductance-Henrys H
Ld = Ls
Lq = Ls
lambda_m = 0.175 #Permanent magnet flux-Webers
P = 8 #Poles
J = 0.8e-3 #Inertia-J
B = 0 #Coefficient of friction

#***Inner flux PI controller***
tau_s = Rs/Ls #flux loop time constant
d_l = tau_s*lambda_m #the disturbance

#Pole Placement for flux loop
pp = -340*4 #Define the roots of the polynomial
#Create the polynomial
poly = sp.poly1d([1, -2*pp, pp**2])
p1 = poly.coeffs[1]
p2 = poly.coeffs[2]
Kp_l = p1 - tau_s #Proportional gain
Ti_l = Kp_l/p2 #Integral time constant

#***Inner Torque PI Controller***
lambda_s_ref = lambda_m
A1 = (3*P*lambda_m)/(4*Ls)
B1 = (tau_s*lambda_m)/lambda_s_ref
C1 = (3*P*P*lambda_m*lambda_s_ref)/(8*J*Ls)

#Pole Placement for torque loop
k = -330*3 #Define the roots of the polynomial
#Create the polynomial
poly = sp.poly1d([1, -2*k, k**2])
k1 = poly.coeffs[1]
k2 = poly.coeffs[2]
Kp_t = (k1-B1)/A1 #Proportional gain
Ti_t = (A1*Kp_t)/(k2-C1) #Integral time constant
wc_t = 1/Ti_t #Cut-off frequency of the Torque reference pre-filter
Fc_t = wc_t/(2*np.pi)

#Outer Speed Controller
a1 = B/J
E = (A1*Kp_t)/((A1*Kp_t) + (C1*Ti_t))
E1 = E/J
#Pole Placement for speed loop
n = -25*3 #Define the roots of the polynomial
#Create the polynomial
poly = sp.poly1d([1, -2*n, n**2])
n1 = poly.coeffs[1]
n2 = poly.coeffs[2]
Kp_w = (n1-a1)/E1
Ti_w = (E1*Kp_w)/n2
wc_w = 1/Ti_w #Cut-off frequency of the speed reference pre-filter
Fc_w = wc_w/(2*np.pi)
```

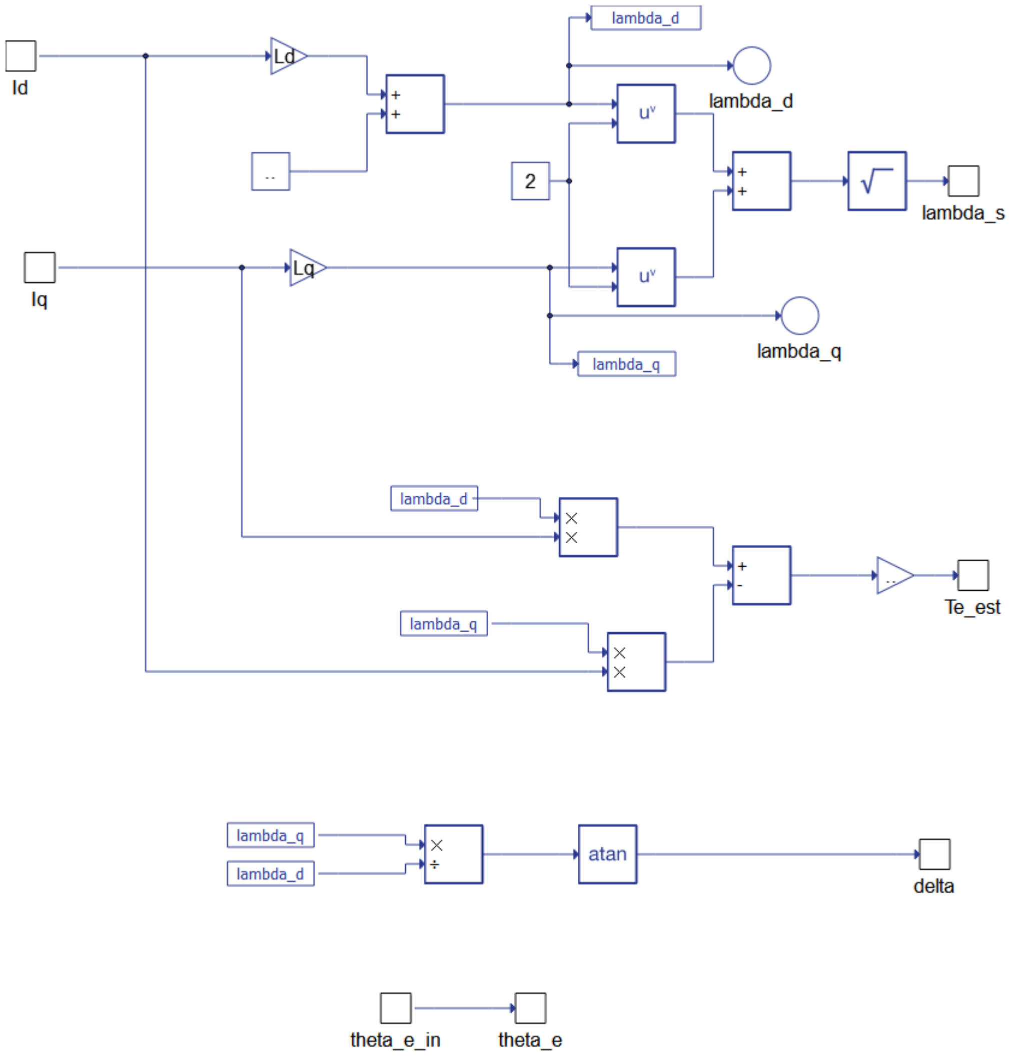
PI Controllers



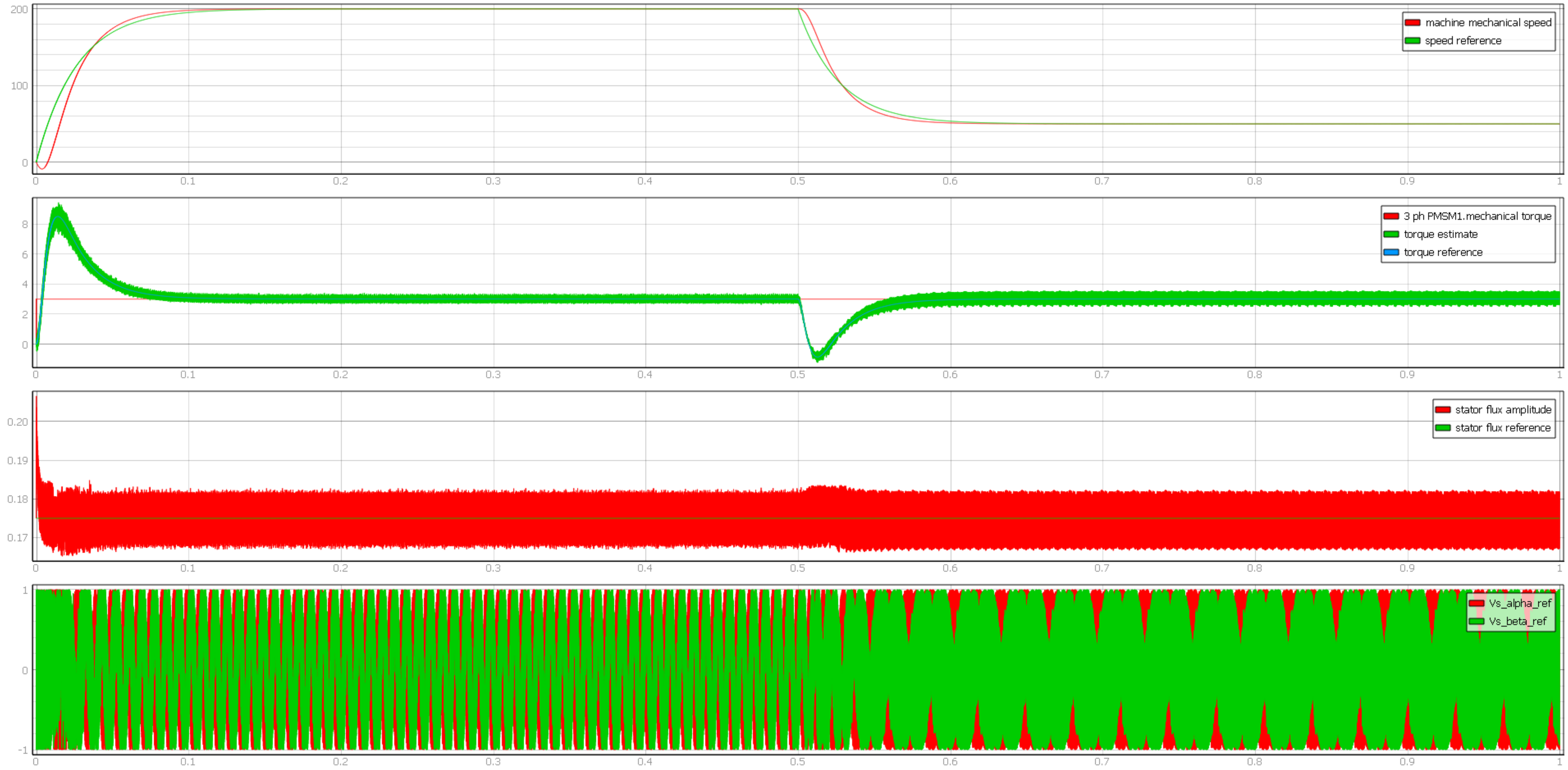
Voltage Vector Calculation Block



Flux Magnitude and Angle Calculation Block

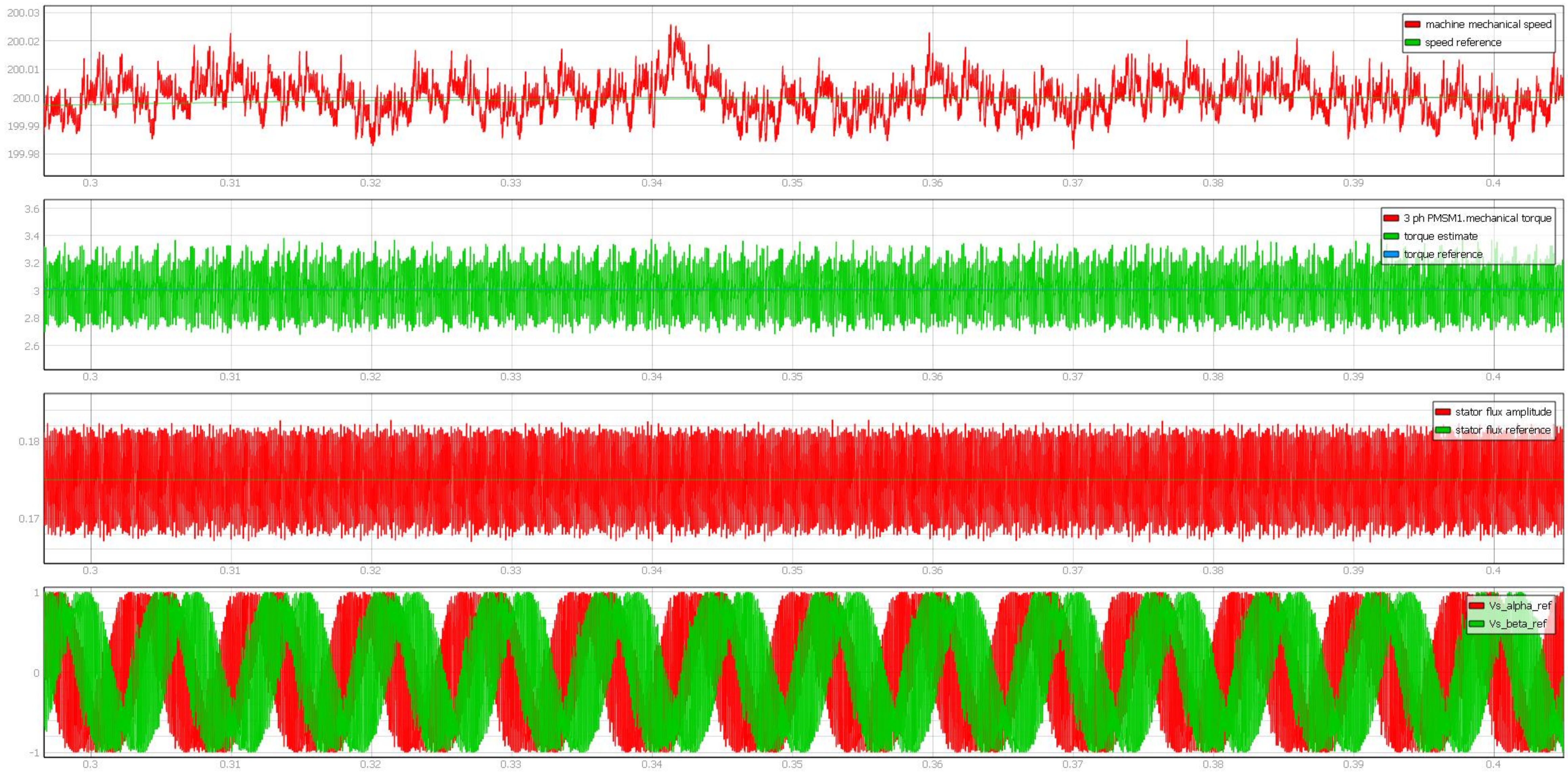


Result1

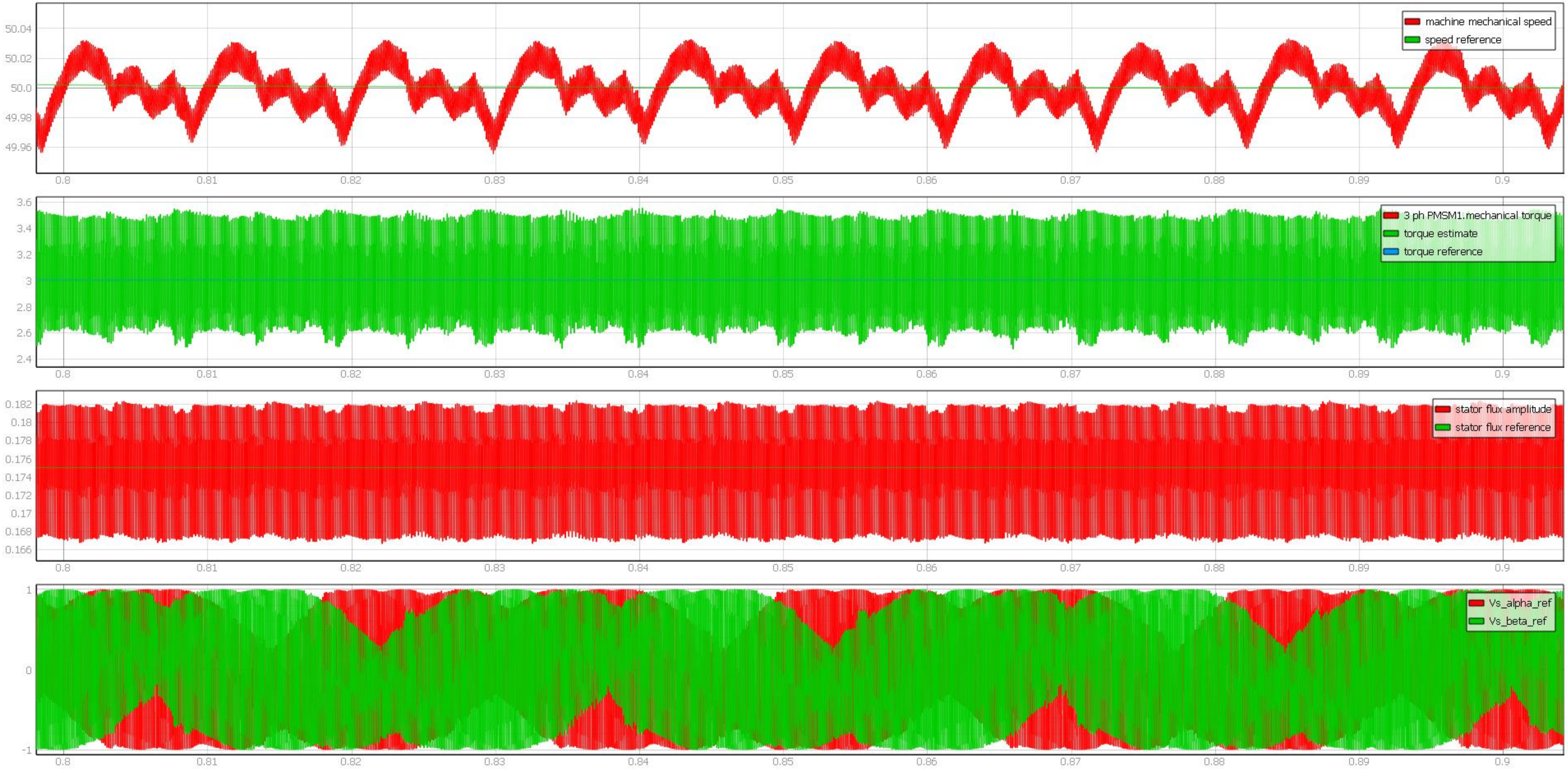


200
100
0
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
machine mechanical speed
speed reference
8
6
4
2
0
3 ph PMSM1.mechanical torque
torque estimate
torque reference
0.20
0.19
0.18
0.17
stator flux amplitude
stator flux reference
1
0
-1
Vs_alpha_ref
Vs_beta_ref

Result2- Steady State at 200 rad/s



Result2- Steady State at 50 rad/s



References

[1]. Chen, C.-Y., Hsu, C.-H., Yu, S.-H., Yang, C.-F., & Huang, H.-H. (2009). Cascade PI Controller Designs for Speed Control of Permanent Magnet Synchronous Motor Drive Using Direct Torque Approach. 2009 Fourth International Conference on Innovative Computing, Information and Control (ICICIC). doi:10.1109/icicic.2009.133

[2] https://www.yumpu.com/en/document/read/23535462/direct-torque-control-with-space-vector-modulation-dtc-svm-of-

Trick Question

Assume the control system is tuned and the system is stable. If the speed loop is disconnected such that the torque reference is set to a random value(say 3Nm)—well within the rating of the PMSM of interest—while a load torque, of say, 6Nm is applied to the machine. The flux reference is still set to be the permanent magnet's flux. What value do you think the PMSM's electromagnetic torque will stabilize at? And what's your reasoning behind that?