DIRECT TORQUE CONTROL OF DOUBLY FED INDUCTION GENERATOR USING ARTIFICIAL NEURAL NETWORK

Ninad Kiran Gaikwad [#1]

[#] *Depatment of Electrical Engineering, Sardar Patel College Of Engineering (SPCE), Munshi Nagar, Andheri (W), Mumbai-400058, India*

[1] ninadkgaikwad@gmail.com

Abstract—**The Direct Torque Control (DTC), when applied to the Doubly Fed Induction Generator (DFIG) simplifies the overall control structure by eliminating the inner current control loops (required in Vector Control) and improve dynamic performance. However, the conventional DTC strategy (C-DTC) has the drawbacks of variable switching frequency which depends on the sampling frequency, the lookup table structure, hysteresis bands, and the converter switching status. This paper tries to improve the C-DTC by using an Artificial Neural Network (ANN), applied in the switching select voltage vector. This strategy reduces the torque and flux ripple at low switching frequency , even under variable speed operation.**

I. INTRODUCTION

The DFIG is a variable speed wind turbine which has its stator connected directly to the grid and its rotor winding is connected to the grid through a back-to-back VSI Voltage Source Inverter). The most important advantage of variable speed wind turbines are the improved dynamic behaviour, resulting in the reduction of the drive train mechanical stress and electrical power fluctuation, it also increases the power capture.

The DFIG can be operated in the subsynchronous as well as the supersynchronous modes (+- 30% arund synchronous speed). Hence, the power inverter needs to handle a fraction (+- 30%) of the total power to achieve full control of the generator.

DTC was proposed in the 1980's, it has excellent steady state and transient performance. In comparison to Field Oriented Control (FOC) DTC is very simple and robust because the current regulators and complicated coordinate transformations are eliminated. However, due to fixed sampling frequency and limited minimum hysteresis bands, both torque and flux exceed hysteresis bands and this results in undesired ripples. Also, the selected voltage vector based on C-DTC table is not necessarily the most suitable one. Moreover, the slopes of the electromagnetic torque and flux vary according to different operating conditions and time interval, hence variable switching frequency behaviour is unavoidable. Therefore, these two main drawbacks of the C-DTC have become an obstacle for high power application such as wind power generation.

To overcome these difficulties, in this paper, a neuronal controller is used to replace the switching table, where the inputs are the error of the flux, of the electromagnetic torque and the position angle of rotor flux, and the output is the impulses allowing the control of the inverter switches.

II. MODELLING OF THE DFIG

In the synchronous d-q reference frame rotating at ω_s speed, the model of the DFIG is given by the following equation:

Stator voltage Equations:

$$\begin{cases} V_{ds} = R_s I_{ds} + \dfrac{d}{dt}\psi_{ds} - \omega_s \psi_{qs} \\ V_{qs} = R_s I_{qs} + \dfrac{d}{dt}\psi_{qs} + \omega_s \psi_{ds} \end{cases} \quad (1)$$

Rotor voltage equations:

$$\begin{cases} V_{dr} = R_r I_{dr} + \dfrac{d}{dt}\psi_{dr} - (\omega_s - \omega_r)\psi_{qr} \\ V_{qr} = R_r I_{qr} + \dfrac{d}{dt}\psi_{qr} + (\omega_s - \omega_r)\psi_{dr} \end{cases} \quad (2)$$

Stator flux equations:

$$\begin{cases} \psi_{ds} = L_s I_{ds} + L_m I_{dr} \\ \psi_{qs} = L_s I_{qs} + L_m I_{qr} \end{cases} \quad (3)$$

Rotor flux equations:

$$\begin{cases} \psi_{dr} = L_r I_{dr} + L_m I_{ds} \\ \psi_{qr} = L_r I_{qr} + L_m I_{qs} \end{cases} \quad (4)$$

DFIG electromagnetic torque:

$$T_{em} = -\frac{3}{2} p \frac{L_m}{L_r}\left(\psi_{ds} I_{qr} - \psi_{qs} I_{dr}\right) \quad (5)$$

Mechanical equation:

$$T_t = T_{em} + J\frac{d\Omega_r}{dt} + f_r \Omega_r \quad (6)$$

Generator active and reactive powers at the stator side are given by the equations:

$$\begin{cases} P_s = \dfrac{3}{2}\left(V_{ds} I_{ds} + V_{qs} I_{qs}\right) \\ Q_s = \dfrac{3}{2}\left(V_{qs} I_{ds} - V_{ds} I_{qs}\right) \end{cases} \quad (7)$$

III. Modelling of the DFIG in Stator Flux Oriented Reference Frame

The rotor-side-converter is controlled in a synchronously rotating d-q axis frame, with the d-axis oriented along the stator flux vector position (Fig. 1). This decouples the control between the stator active and reactive powers. The influence of the stator resistance can be neglected and the stator flux can be held constant as the stator is connected to the grid, hence:

$$\psi_{ds} = \psi_s \quad and \quad \psi_{qs} = 0 \tag{8}$$

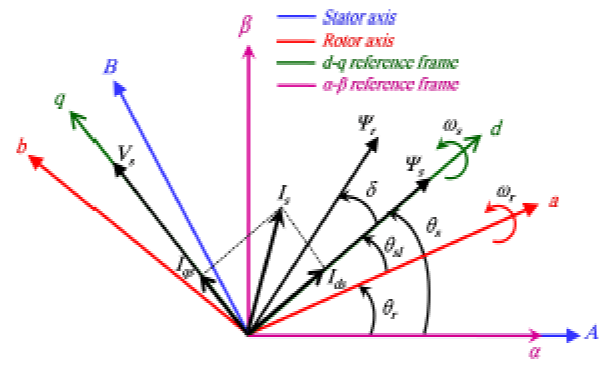


Fig. 1 Field Oriented Control

$$\begin{cases} V_{ds} = 0 \\ V_{qs} = V_s = \omega_s \psi_s \end{cases} \tag{9}$$

$$\begin{cases} \psi_s = L_s I_{ds} + L_m I_{dr} \\ 0 = L_s I_{qs} + L_m I_{qr} \end{cases} \tag{10}$$

$$\begin{cases} I_{ds} = \dfrac{\psi_s}{L_s} - \dfrac{L_m}{L_r} I_{dr} \\ I_{qs} = -\dfrac{L_m}{L_s} I_{qr} \end{cases} \tag{11}$$

$$\begin{cases} P_s = \dfrac{3}{2} V_s I_{qs} \\ Q_s = \dfrac{3}{2} V_s I_{ds} \end{cases} \tag{12}$$

Replacing the stator currents by their expressions given in (11), the equations in (7) are expressed as follows, these are used for obtaining flux reference:

$$\begin{cases} P_s = -\dfrac{3}{2}\dfrac{L_m}{L_s} V_s I_{qr} \\ Q_s = \dfrac{3}{2} V_s \left(\dfrac{V_s}{L_s \omega_s} - \dfrac{L_m}{L_s} I_{dr} \right) \end{cases} \tag{13}$$

The electromagnetic torque is as follows:

$$T_{em} = -\frac{3}{2} p \frac{L_m}{L_s} \psi_s I_{qr} \tag{14}$$

IV. Direct Torque Control Principle

The main goal of DTC is direct control of rotor flux and the electromagnetic torque of the DFIG by choosing the best voltage vector. The schematic of C-DTC is given in Fig. 2.

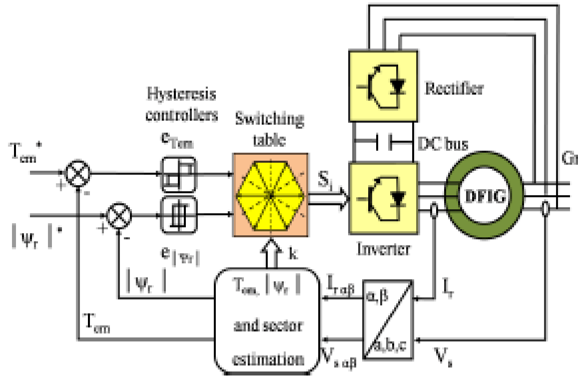


Fig. 2 Schematic of C-DTC

As shown in Fig. 3, the position of the rotor flux is divided into six sectors of 60° each. The inverter has 8 voltage vectors, divided into two types, active voltage vectors (V1,....,V6) and zero voltage vectors (V0 and V7).

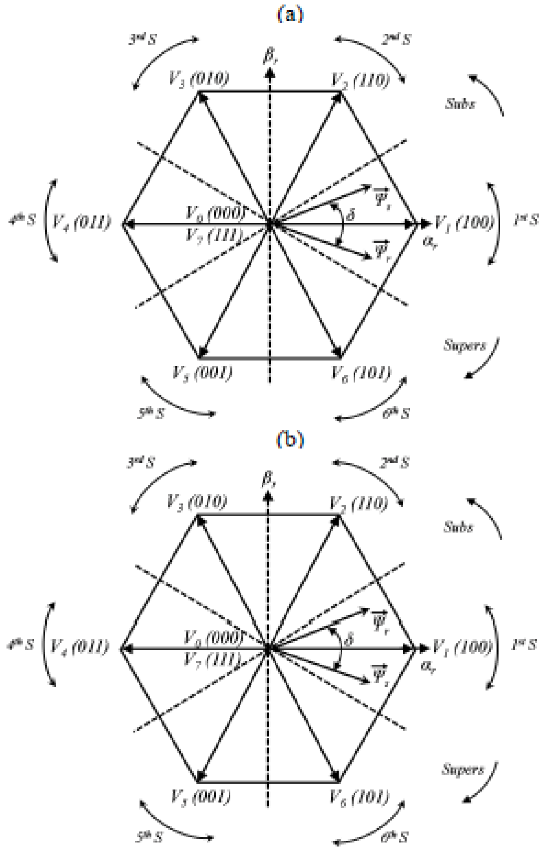


Fig. 3 Flux Space Vectors in the rotor reference frame in motor and generator modes, (a) Motoring Mode, (b) Generating Mode

Based on the operation conditions of the machine and the position of the rotor and stator flux space vectors expressed in the rotor reference frame the rotor voltage vectors affect the torque and rotor flux in different ways.

The stator winding of the DFIG is directly connected to the grid. By neglecting effect of stator resistance and the variation of supply voltage one can effectively consider the stator flux to be constant and rotating at synchronous speed;

$$|\overline{\psi}_s| = \frac{|\overline{V}_s|}{\omega_s} \tag{15}$$

The electromagnetic torque of the DFIG is represented as a function of angle δ between the stator and rotor flux space vectors as follow:

$$T_{em} = -\frac{3}{2} p \frac{L_m}{\sigma L_s L_r} |\psi_s||\psi_r| \sin \delta \tag{16}$$

Where leakage coefficient is given by:

$$\sigma = 1 - \frac{L_m^2}{L_s L_r}$$

Hence, according to the equation (17) the torque control of the DFIG machine can be realized through adjusting the rotor flux vector. The amplitude of the rotor flux vector can be expressed as:

$$|\overline{\psi}_r| = \sqrt{\psi_{\alpha r}^2 + \psi_{\beta r}^2} \tag{18}$$

With:

$$\begin{cases} \psi_{\alpha r} = \left(L_r - \frac{L_m{}^2}{L_s} \right) I_{\alpha r} + \frac{L_m}{L_s} \psi_s \\ \psi_{\beta r} = \left(L_r - \frac{L_m{}^2}{L_s} \right) I_{\beta r} \end{cases} \tag{19}$$

Moreover, as the rotor flux has a circular trajectory the electromagnetic torque becomes the fuction of phase angle δ. Therefore, the control of torque is realized by adjusting phase angle δ.

Hence, the torque is controlled by adjusting the phase angle δ, and the rotor flux is controlled by controlling the magnitude of rotor flux vector directly.

V. Hysteresis Controllers and Rotor voltage vector selection

The DTC technique selects the required rotor voltage vector directly from the rotor flux and electromagnetic torque errors, using hysteresis controllers also called as Bang-Bang controllers. The flux controller uses a two level hysteresis comparator with H_F Hysteresis band. The torque controller uses a three level hysteresis comparator with H_T hysteresis band.

A. *Flux Vector Control*

The two level hysteresis given in Fig. 4 comparator is the simplest solution to control and maintain the end of the flux vector in a circular ring . The exit of the comparator represented by H_F (= -1 or 1) indicates if the module of flux must decrease (H_F = -1) or increase (H_F =1) so as to always maintain;

$$|\psi_r^* - \psi_r| \le \Delta|\psi_r|.$$

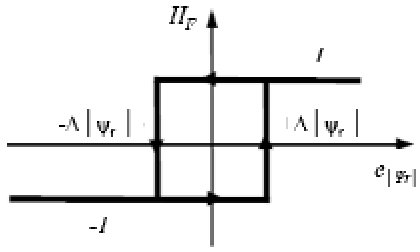


Fig. 4 Two Level Hysteresis Controller

B. *Torque Vector Control*

The three level hysteresis comparator given in Fig. 5 makes it possible to control the device in two directions of rotation i.e. positive or negative torque. The exit of the comparator represented by H_T(= -1 or 0 or 1) indicates if the torque should decrease (H_T= -1) or increase (H_T= 1) or stay unchanged (H_T= 0) so as to always maintain;

$$|T_{em}^* - T_{em}| \le \Delta T_{em}.$$

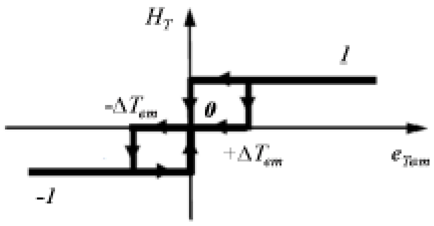


Fig. 5 Three Level Hysteresis Controller

C. *Rotor Voltage Vector Selection*

The output of both the hysteresis controllers that show flux and torque should be increase or decrease. The voltage vectors to be selected from the hysteresis controller outputs and the sector where the rotor flux space vector is located and also the mode of operation i.e. whether subsynchronous (rotor flux vector rotates in direction of stator flux vector) or supersynchronous (rotor flux vector rotates in direction opposite to that of stator flux vector). The selection of the voltage vector is the carried out using the following look-up table.

DFIG speed	$e_{\psi r}$	e_{Te}	S(1)	S(2)	S(3)	S(4)	S(5)	S(6)
Sub.synchronous speed	1	1	V2	V3	V4	V5	V6	V6
		0	V7	V0	V7	V0	V7	V0
		-1	V6	V1	V2	V3	V4	V5
	0	1	V3	V4	V5	V6	V1	V2
		0	V0	V7	V0	V7	V0	V7
		-1	V5	V6	V1	V2	V3	V4
Super synchronous speed	1	1	V6	V1	V2	V3	V4	V5
		0	V7	V0	V7	V0	V7	V0
		-1	V1	V2	V3	V4	V5	V6
	0	1	V5	V6	V1	V2	V3	V4
		0	V0	V7	V0	V7	V0	V7
		-1	V4	V5	V6	V1	V2	V3

Table. 1 Rotor voltage vector selection Look-Up Table

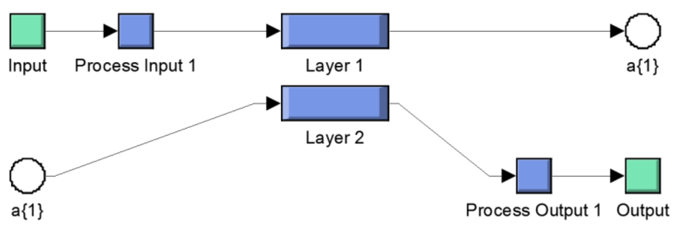
VI. ARTIFICIAL NEURAL NETWORK BASED VOLTAGE VECTOR SELECTION

ANN has a very significant role in the field of artificial intelligence. The artificial neurons learn from the data fed to them and keep on decreasing the error during training time and once trained properly, their results are very much same to the results required from them. The most popular neural network used by researchers are the multilayer feed forward neural networks trained by back propagation algorithm. Here we have used a feed forward network to select the voltage vector which replaces the look-up table in the case of C-DTC.

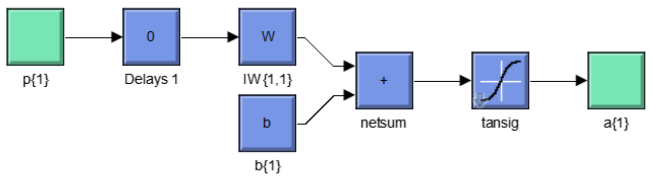
In this paper the DTC control strategy shown in Table 1 has been implemented. Neural network has been devised having inputs as the speed of the rotor, torque error, flux error and the position of the rotor flux sector and the output are the voltage space vector to be generated by the inverter. The ANN block replaces the switching table in Fig. 2, as given in Fig. 6.

The network taken in this paper is a 4-100-3 feed forward network. The back propagation algorithm is used to train the networks. The training function used is the Lavenberg-Marquardt (LVM) back propagation, it updates weights and bias values according to the Lavenberg-Marquardt optimization algorithm. As soon as the training procedure is over, the neural network gives almost same output pattern for the same or nearby values of the input. This tendency of the ANN which approximates the output for new input data is the reason for which they are used in intelligent systems. The ANN Network was prepared using the Neural Network Tool Box in MATLAB/SIMULINK as shown in Fig. 7 and Fig. 8

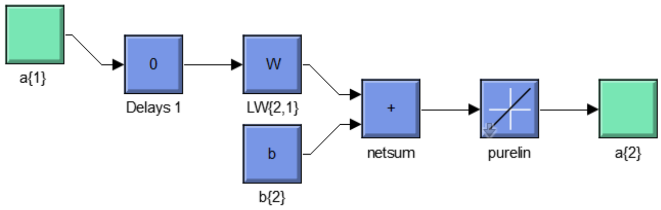
Fig. 7 ANN Schematic



(a)



(b)

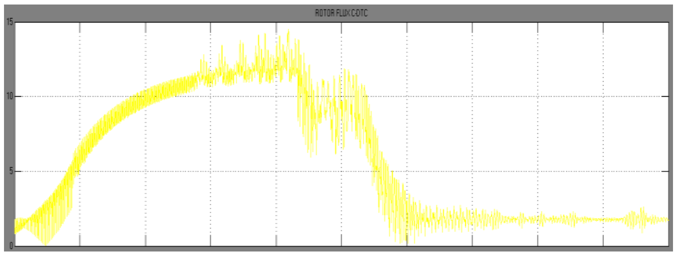


(c)

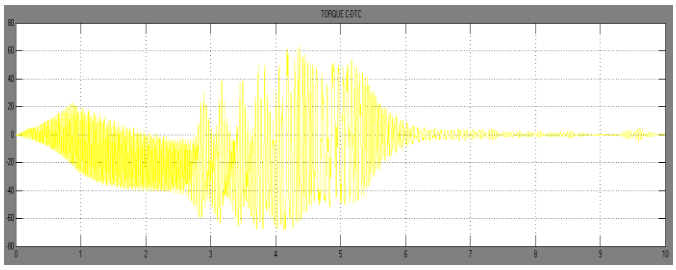
Fig. 8 (a) Neural Network Block; (b) Block Layer 1; (c) Block Layer 2

VII. SIMULATION RESULTS

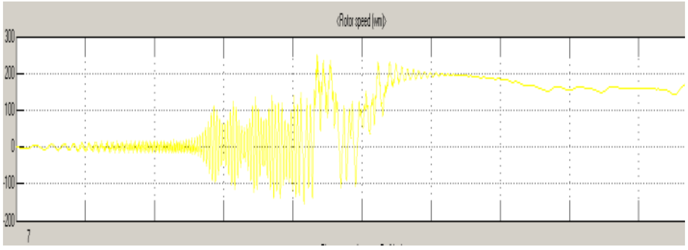
Simulations have been carried out in MATLAB/SIMULINK. The Wind Turbine Parameters and DFIG Parameters are provided in the APPENDIX 1. The detailed SIMULINK models for C-DTC and ANN-DTC are provide in the APPENDIX 2. The results of the simulation are given below for both C-DTC and ANN-DTC.



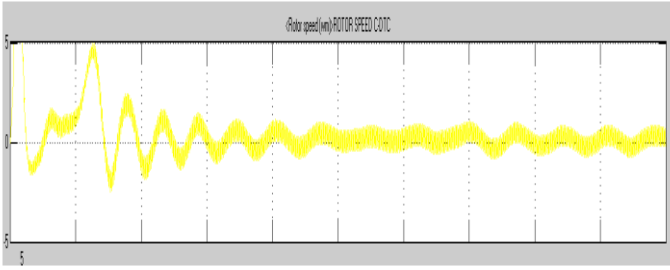
ROTOR FLUX C-DTC



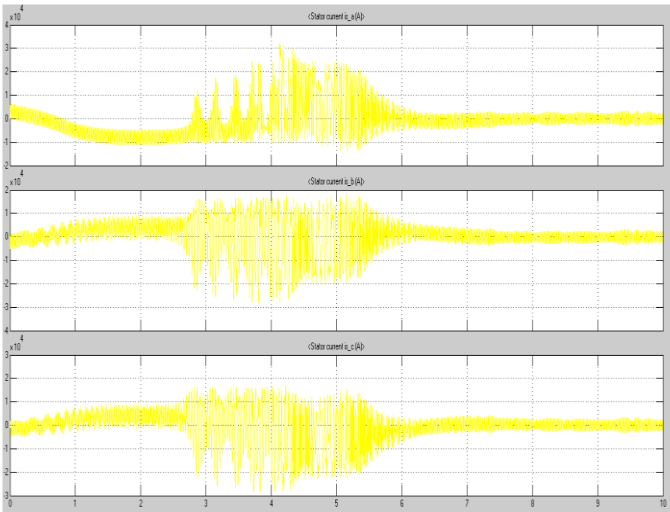
TORQUE C-DTC



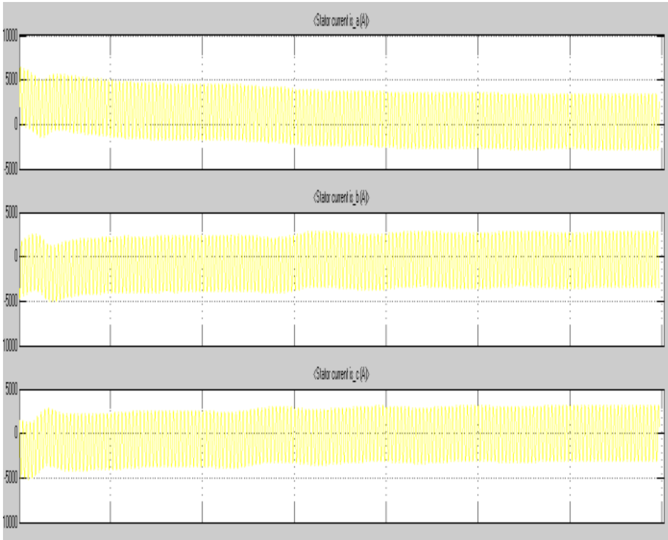
ROTOR SPEED C-DTC



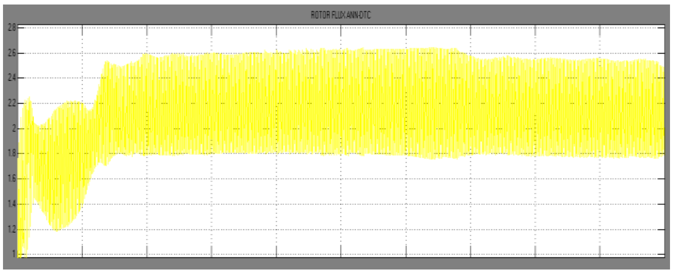
ROTOR SPEED ANN-DTC



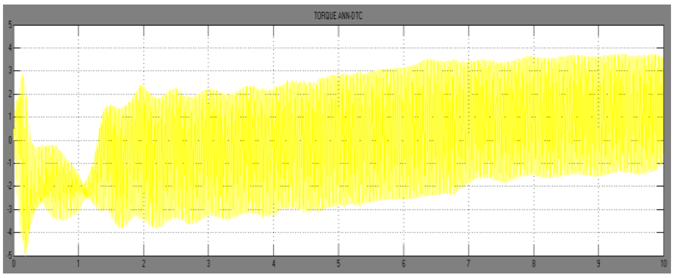
STATOR CURRENTS C-DTC



STATOR CURRENTS ANN-DTC



ROTOR FLUX ANN-DTC



TORQUE ANN-DTC

VIII. Conclusion

This paper presents an improved direct torque control strategy for DFIG using ANN for selecting the rotor voltage vector. This strategy significantly reduces torque and flux ripples and low constant switching frequency is achieved, while the simplicity and robustness of the C-DTC is maintained.

The C-DTC strategy presents a fast and good dynamic torque in steady state behavior. However this strategy because of the variable switching frequency causes high frequency of switching which causes high harmonic distortion of the currents, high ripples of the electromagnetic torque and warming up of semiconductor switches.

The ANN-DTC strategy is a viable alternative to the C-DTC as it has the advantages of fixed and low switching frequency, reduced torque and current ripples.

APPENDIX 1

Table 2
Wind Turbine Parameters

Blade radius, *R*	35.25 m
Number of blades	3
Gearbox ratio, *G*	90
Moment of inertia, *J*	1000 Kg.m^2
Viscous friction coefficient, f_r	0.0024 N.m.s^{-1}
Cut-in wind speed	4 m/s
Cut-out wind speed	25 m/s
Nominal wind speed, *v*	16 m/s

Table 3
DFIG Parameters

Rated power, P_n	1.5 MW
Stator rated voltage, V_s	398/690 V
Rated current, I_n	1900 A
Rated DC-Link voltage U_{DC}	1200 V
Stator rated frequency, *f*	50 Hz
Stator inductance, L_s	0.0137 H
Rotor inductance, L_r	0.0136 H
Mutual inductance, L_m	0.0135 H
Stator resistance, R_s	0.012 Ω
Rotor resistance, R_r	0.021 Ω
Number of pair of poles, *p*	2

REFERENCES

[1] Y. Djeriri, A. Meroufel and A. Massoum, “Artificial Neural Network Based Direct Torque Control of Doubly Fed Induction Generator”, Journal of Electrical Engineering.

[2] A.A. Hassan, A.M. EL-Sawy and O.M. Kamel, “Direct Torque Control of a Doubly Fed Induction Generator Driven by a Variable Speed Wind Turbine”, Journal of Engineering Sciences, Assiut University, 2013, vol. 41.

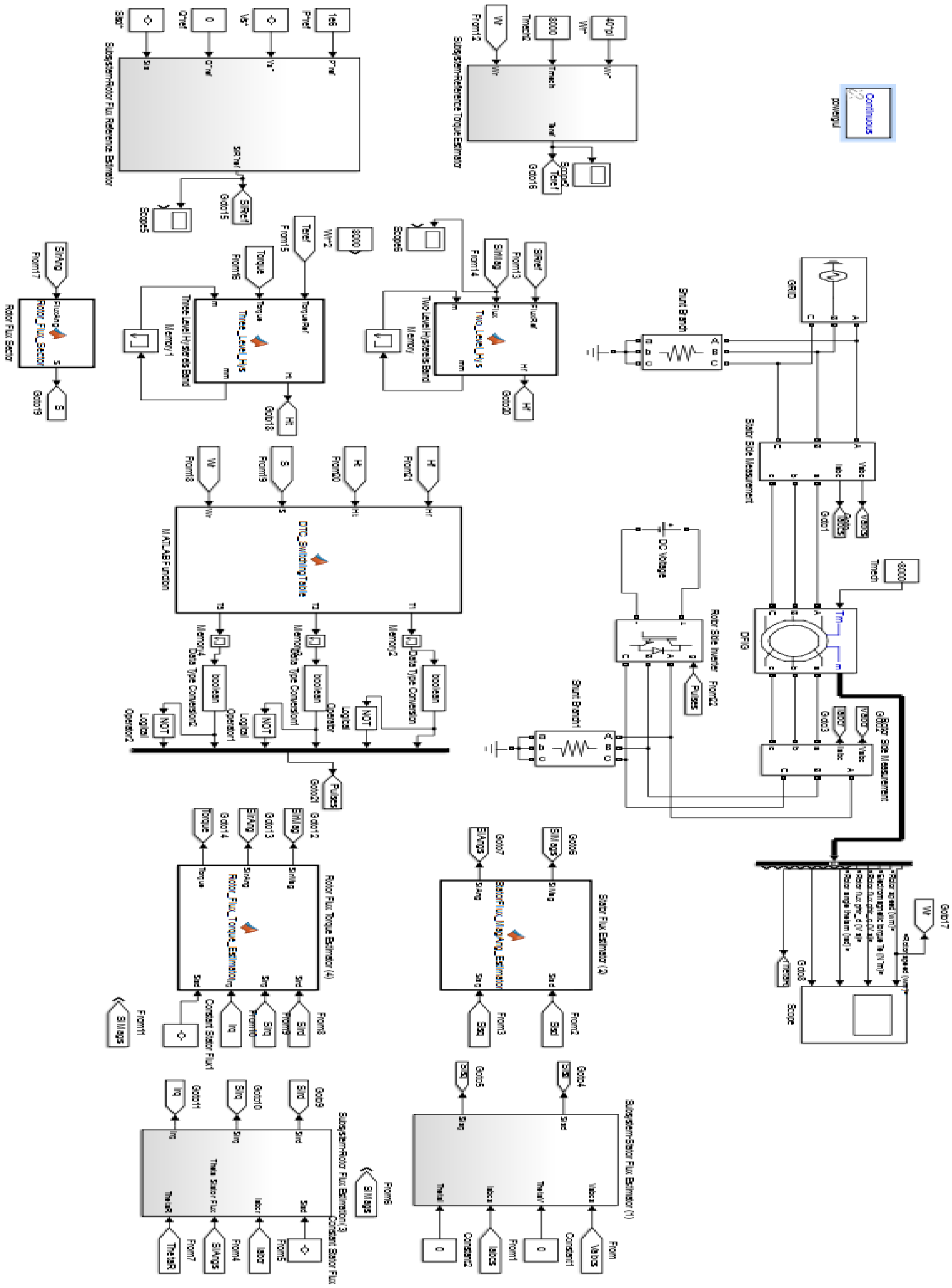
[3] Z. Mahi, C. Serban and H. Siguerdidjane, “ Direct Torque Control of a Doubly Fed Induction Generator of a Variable Speed Wind Turbine Power Regulation”.

[4] Gonzalo Abad Bian, “ Predictive Control Techniques of the Doubly Fed Induction Machine for Wind Energy Generation Application”, PhD Thesis, Mondragon Unibertsitatea, 2008.

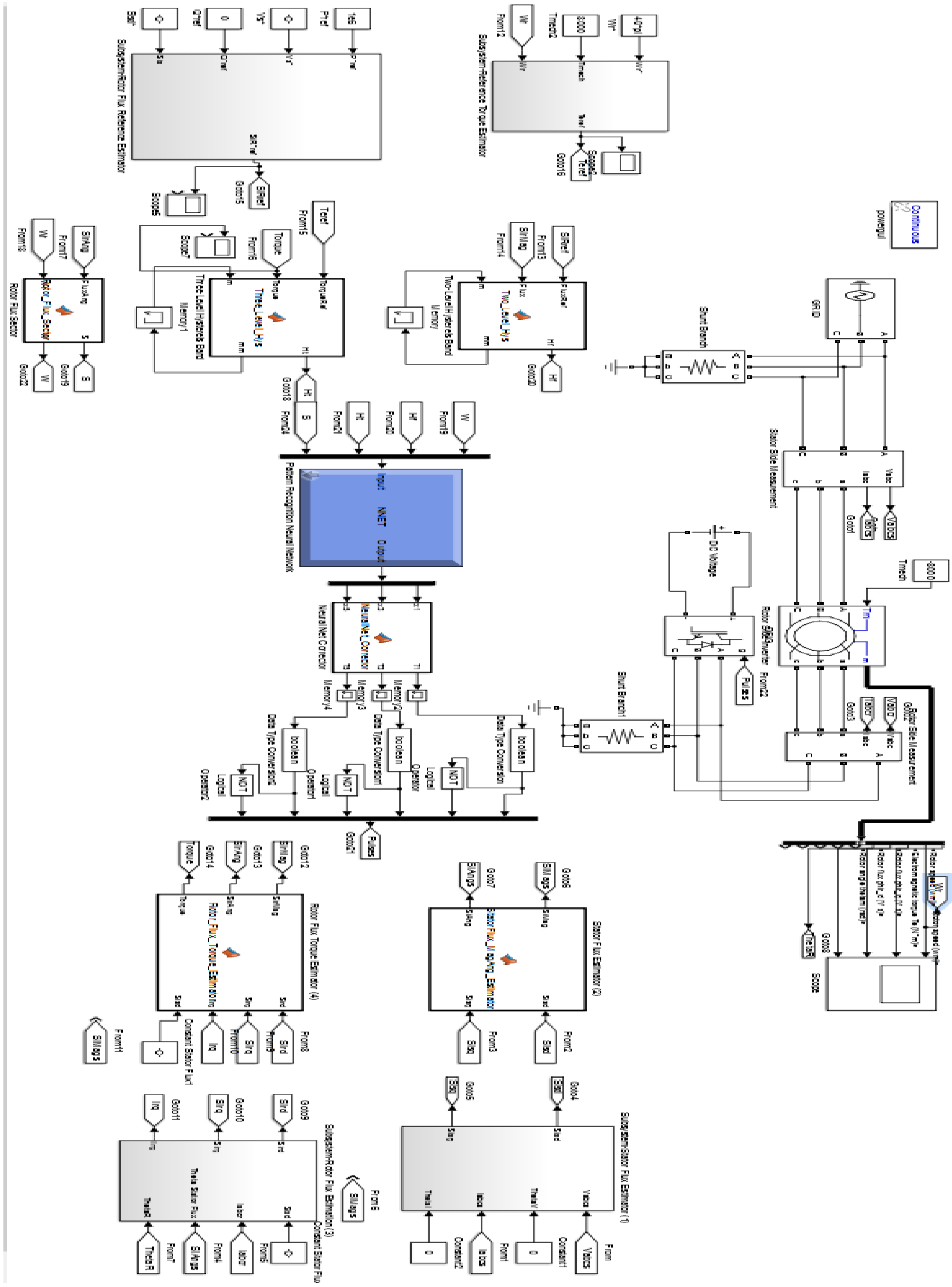
[5] Martino O. Ajangnay, “Optimal PID Controller for Vector Control of Induction Motors,” International Symposium on Power Electronics, Electrical Drives, Automation and Motion, SEEDAM 2010.

APPENDIX 2

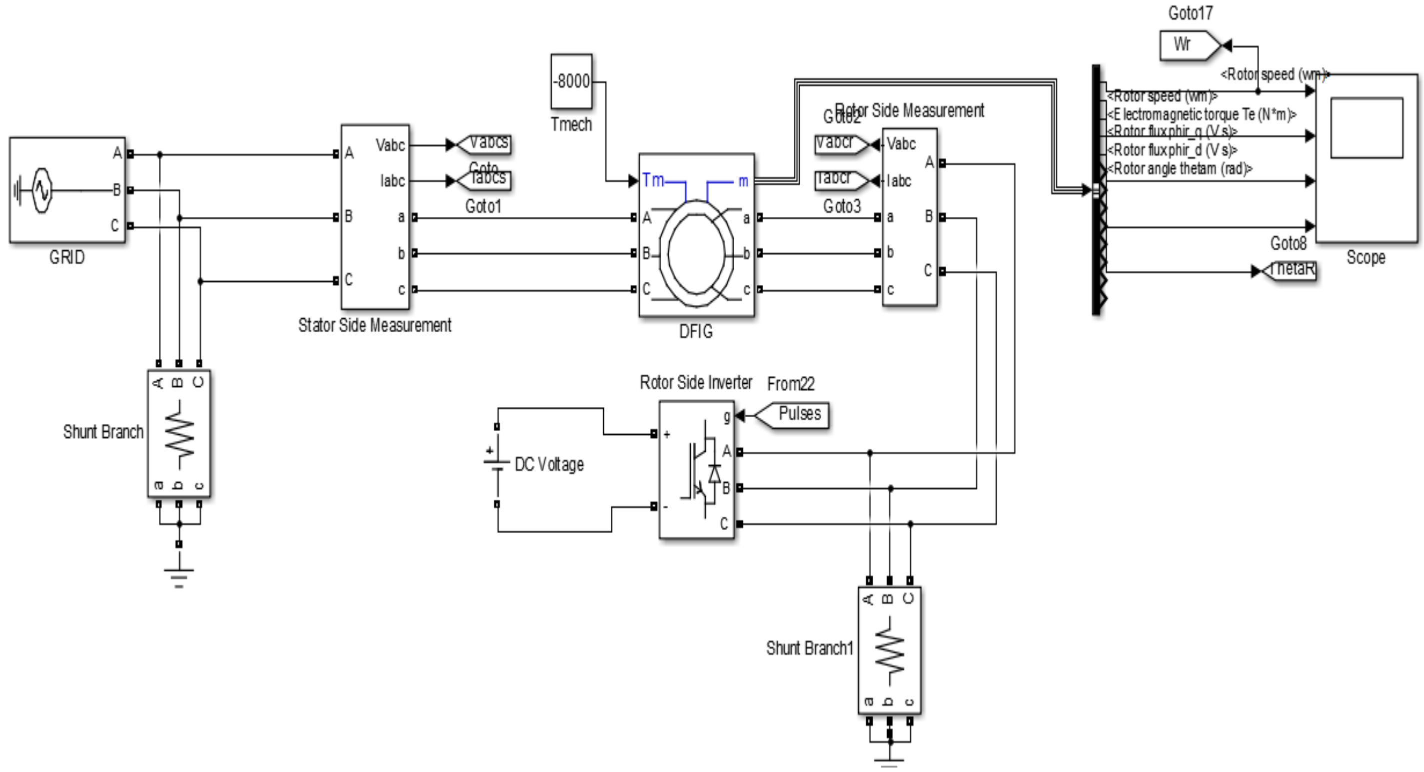
1. Simulink Model of C-DTC:



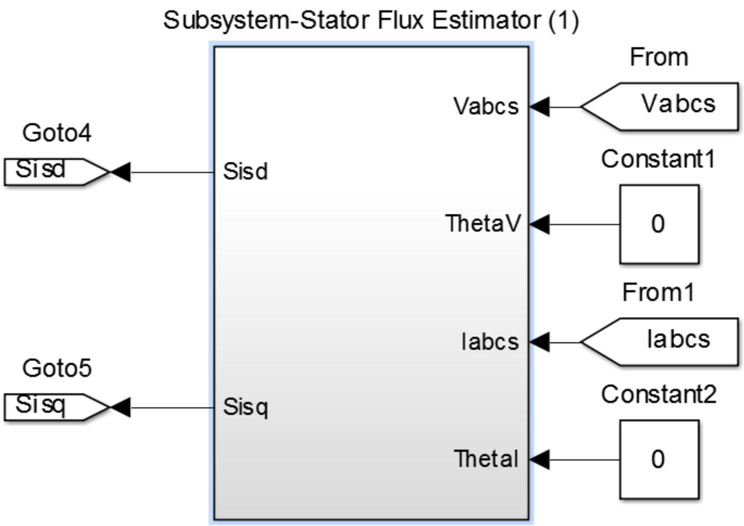
2. Simulink Model of ANN-DTC:



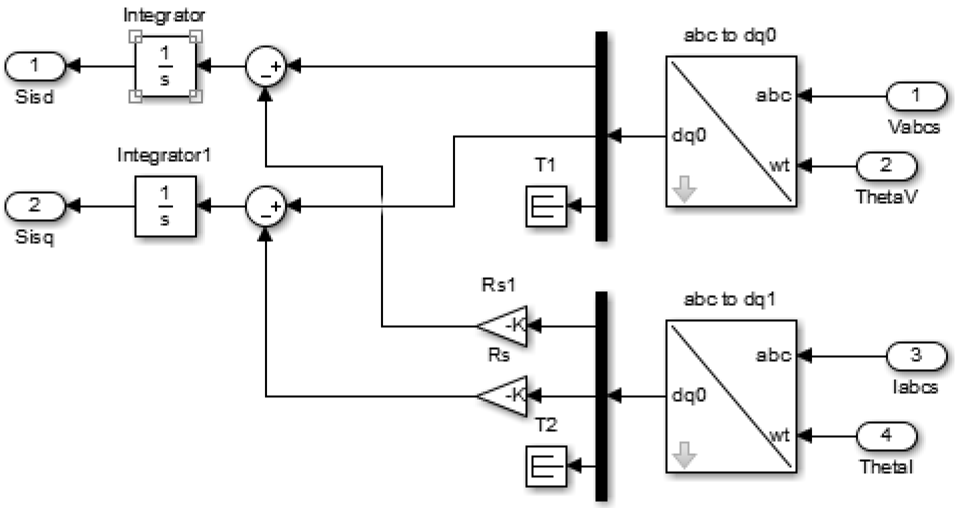
2. Simulink Model of DFIG System:



4. Simulink Model Stator Flux Estimator: (a) Block, (b) Inside the Block

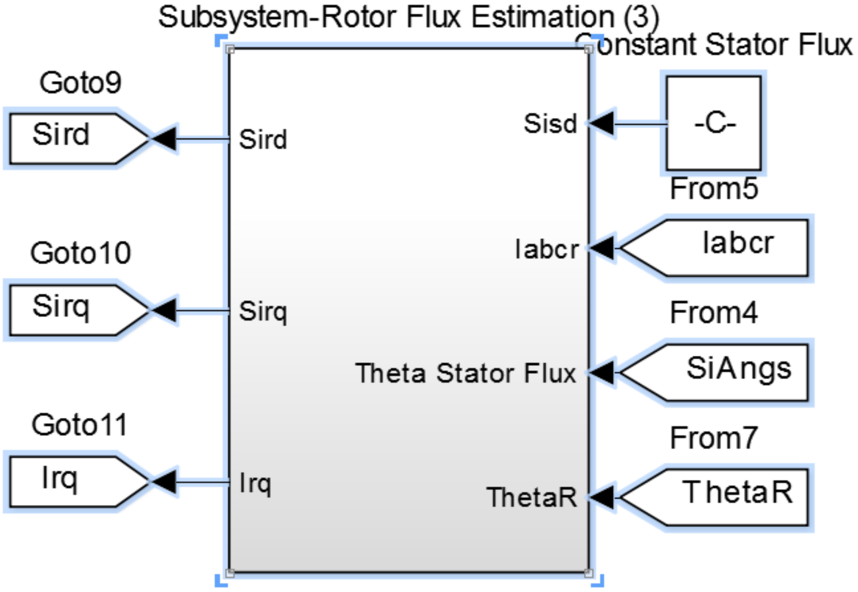


(a)

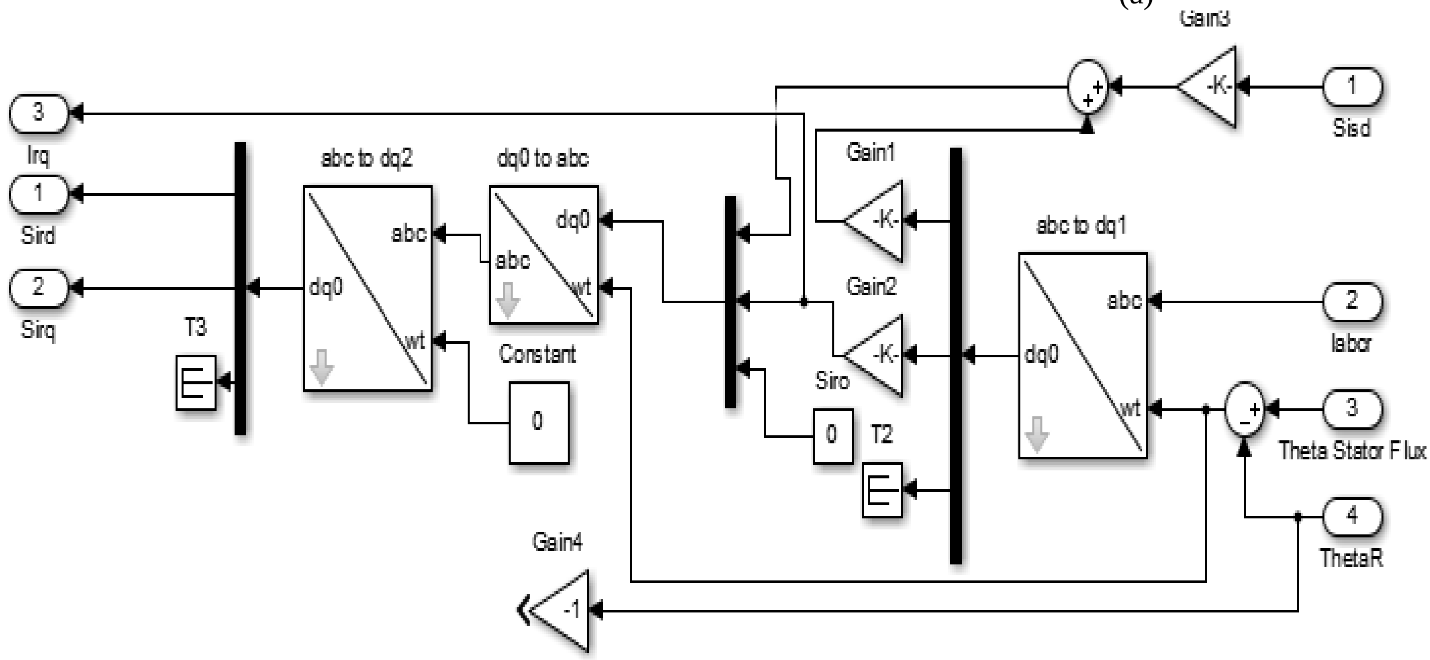


(b)

5. Simulink Model Rotor Flux Estimator: (a) Block, (b) Inside the Block

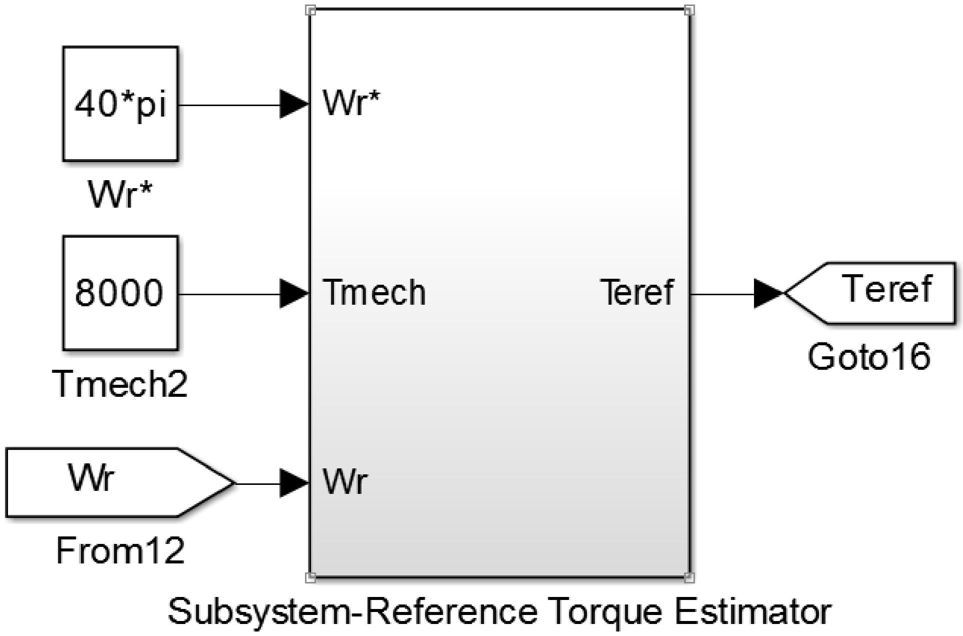


(a)

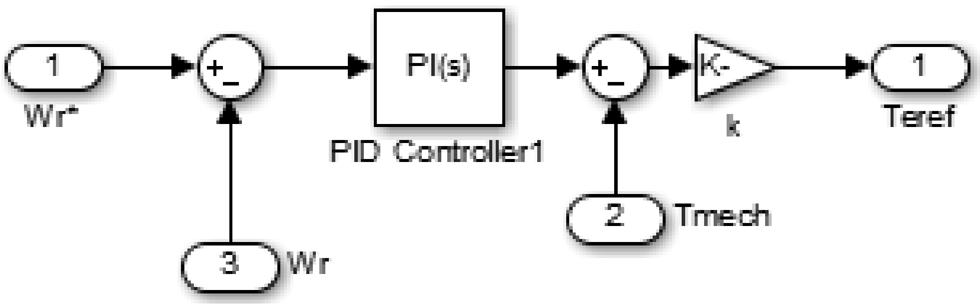


(b)

6. Simulink Model Reference Torque Estimator: (a) Block, (b) Inside the Block

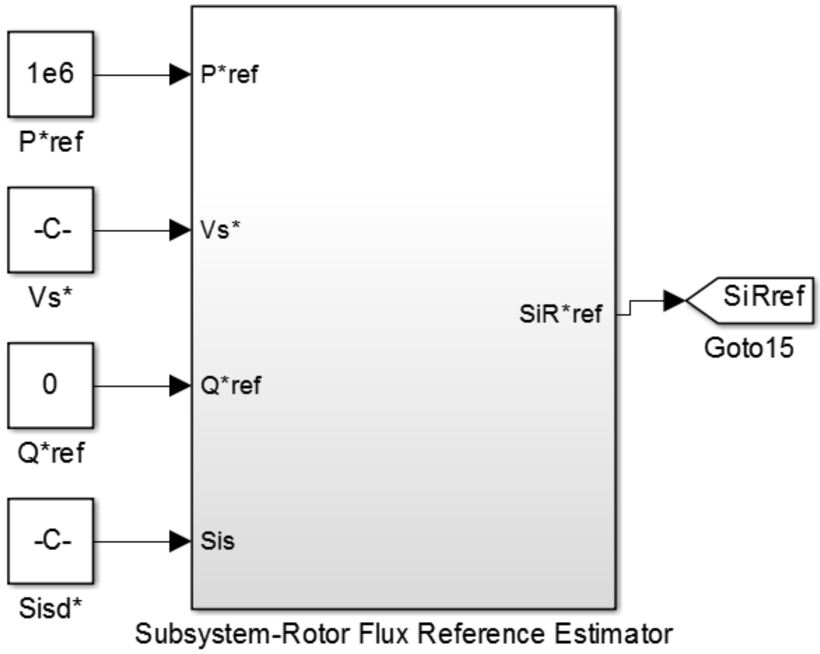


(a)

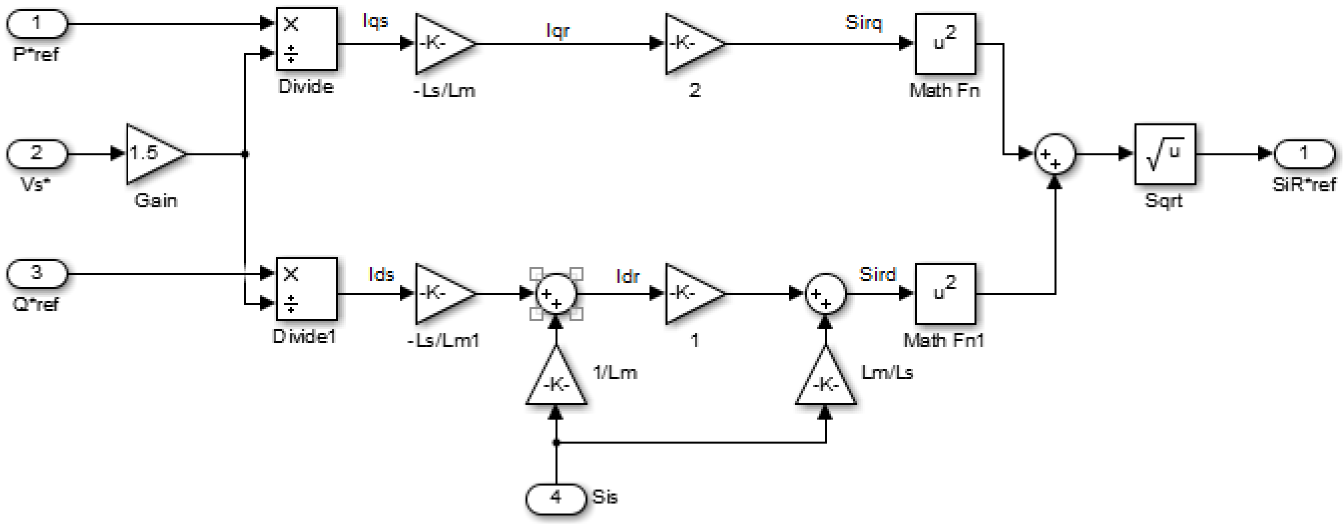


(b)

7. Simulink Model Reference Rotor Flux Estimator: (a) Block, (b) Inside the Block

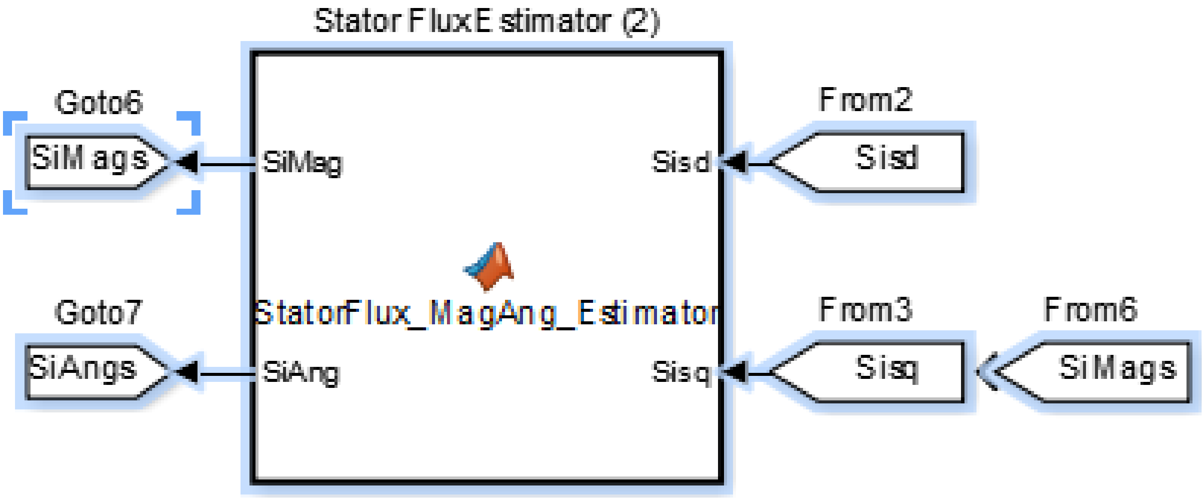


(a)



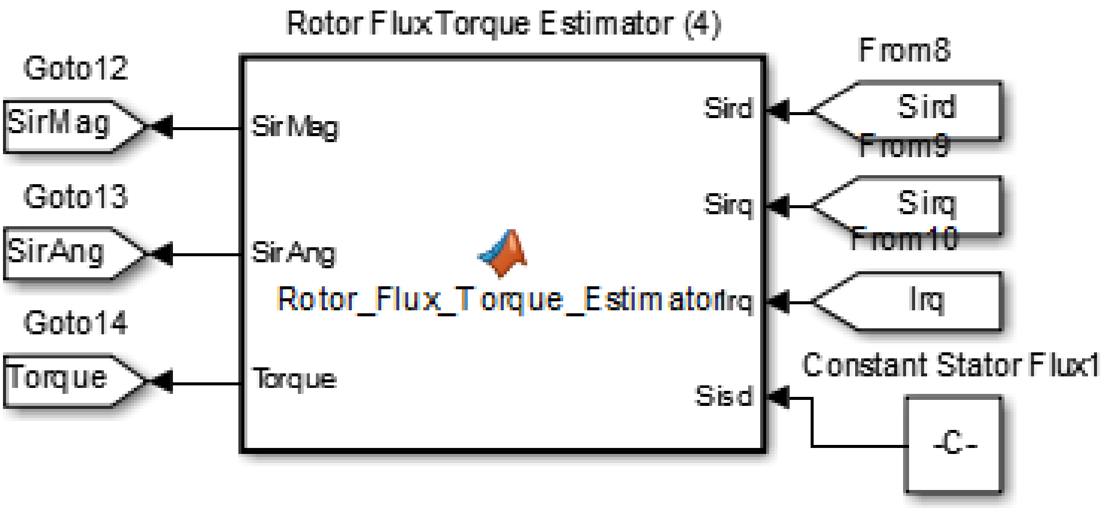
(b)

8. MATLAB Function and Simulink Block for Stator Flux Estimator:



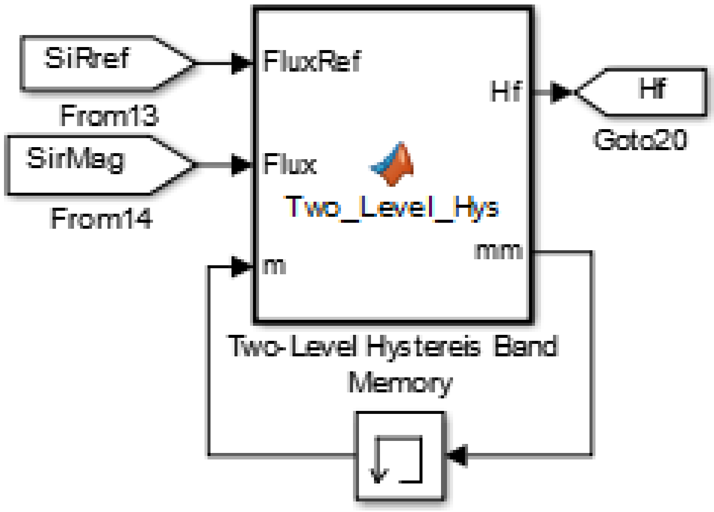

```
function [SiMag,SiAng] = StatorFlux_MagAng_Estimator(Sisd,Sisq)
 SiMag=((Sisd)^(2)+(Sisq)^(2))^(0.5);
 x=atan2(Sisq,Sisd);
 if(x>=0)
    SiAng=x;
 else
    SiAng=x+(2*pi);
 End
```

9. MATLAB Function and Simulink Block for Rotor Flux Estimator:




```
function [SirMag,SirAng,Torque] = Rotor_Flux_Torque_Estimator(Sird,Sirq,Irq,Sisd)
 SirMag=((Sird)^(2)+(Sirq)^(2))^(0.5);
 x=atan2(Sirq,Sird);
 if(x>=0)
    SirAng=x*(180/pi);
 else
    SirAng=x*(180/pi)+360;
 end
 Torque=(-3)*(0.0135/0.0136)*(Sisd*Irq);
```

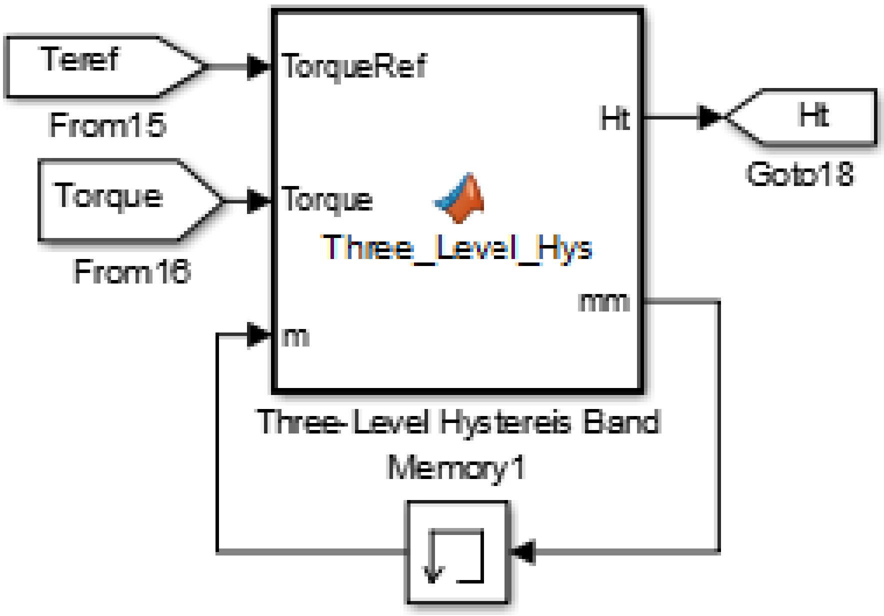
10 . MATLAB Function and Simulink Block for Two Level Hysteresis Controller:




```
function [Hf,mm] = Two_Level_Hys(FluxRef,Flux,m)

DelF=0.025;% Hysteresis Width for Flux
Hf=0;%Initializing Hysteresis Controller Output For Flux
mm=m;
% Flux Two Level Hysteresis Band %
%-------------------------------------------------------------------------%
if(Flux<=(FluxRef-DelF))
    Hf=1;
    mm=Hf;
end
if(mm==1&&(Flux>(FluxRef-DelF))&&(Flux<(FluxRef+DelF)))
    Hf=1;
    mm=Hf;
end
if(Flux>=(FluxRef+DelF))
    Hf=0;
    mm=Hf;
end
if(mm==0&&(Flux>(FluxRef-DelF))&&(Flux<(FluxRef+DelF)))
    Hf=0;
    mm=Hf;
end
```

11 . MATLAB Function and Simulink Block for Three Level Hysteresis Controller:




```
function [Ht,mm] = Three_Level_Hys(TorqueRef,Torque,m)

DelT=0.025;% Hysteresis Width for Torque
Ht=0;%Initializing Hysteresis Controller Output For Torque
mm=m;

% Torque Three Level Hysteresis Band %
%-------------------------------------------------------------------------%
if(Torque<=(TorqueRef-DelT))
    Ht=1;
    mm=Ht;
end
if((Torque>(TorqueRef-DelT))&&(Torque<TorqueRef)&&mm==1)
    Ht=1;
    mm=Ht;
end
if(Torque>=(TorqueRef+DelT))
    Ht=-1;
    mm=Ht;
end
if((Torque<(TorqueRef+DelT))&&(Torque>TorqueRef)&&mm==-1)
    Ht=-1;
    mm=Ht;
end
if((Torque>(TorqueRef-DelT))&&(Torque<TorqueRef)&&mm==-1)
    Ht=0;
    mm=-1;
end
if((Torque<(TorqueRef+DelT))&&(Torque>TorqueRef)&&mm==1)
    Ht=0;
    mm=1;
```

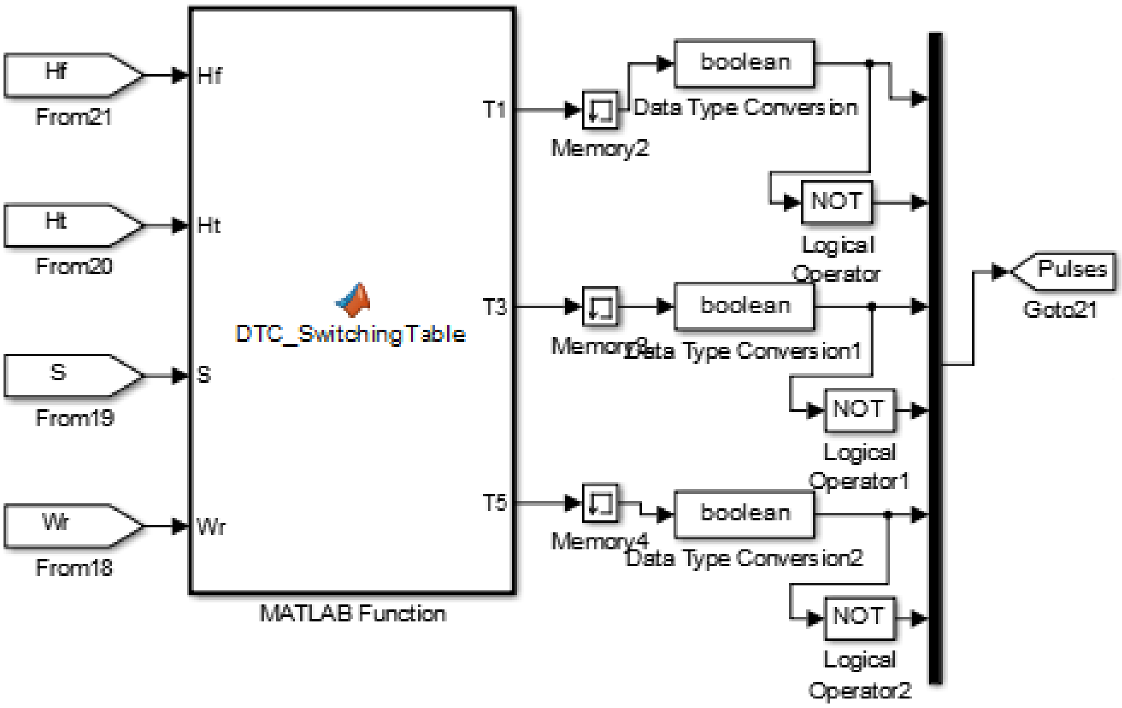
```
end
```

12 . MATLAB Function and Simulink Block for Rotor Flux Sector Calculator:


```
function S= Rotor_Flux_Sector(FluxAng)

S=0;
% DTC Sector Prediction %
%-------------------------------------------------------------------------%
if(FluxAng>=330&&FluxAng<30)
    S=1;
end
if(FluxAng>=30&&FluxAng<90)
    S=2;
end
if(FluxAng>=90&&FluxAng<150)
    S=3;
end
if(FluxAng>=150&&FluxAng<210)
    S=4;
end
if(FluxAng>=210&&FluxAng<270)
    S=5;
end
if(FluxAng>=270&&FluxAng<330)
    S=6;
end
```

12 . MATLAB Function and Simulink Block for DTC Switching Table:




```
function [T1,T3,T5] = DTC_SwitchingTable(Hf,Ht,S,Wr)

T1=0;
T3=0;
T5=0;

% Voltage Vector Synthesis %
%----------------------------------------------------------------------%
if(Wr<=(50*pi))% Sub-Synchronous
%----------------------------------------------------------------------%
    if(Hf==1&&Ht==1&&S==1)
        T1=1;
        T3=0;
        T5=1;
    end
    if(Hf==1&&Ht==1&&S==2)
      T1=0;
      T3=0;
      T5=1;
    end
    if(Hf==1&&Ht==1&&S==3)
      T1=0;
      T3=1;
      T5=1;
    end
    if(Hf==1&&Ht==1&&S==4)
      T1=0;
```

```
      T3=1;
      T5=0;
    end
    if(Hf==1&&Ht==1&&S==5)
      T1=1;
      T3=1;
      T5=0;
    end
    if(Hf==1&&Ht==1&&S==6)
      T1=1;
      T3=0;
      T5=0;
    end

%------------------------------------------------------------------------%
    if(Hf==1&&Ht==0&&S==1)
        T1=1;
        T3=1;
        T5=1;
    end
    if(Hf==1&&Ht==0&&S==2)
      T1=0;
      T3=0;
      T5=0;
    end
    if(Hf==1&&Ht==0&&S==3)
      T1=1;
      T3=1;
      T5=1;
    end
    if(Hf==1&&Ht==0&&S==4)
      T1=0;
      T3=0;
      T5=0;
    end
    if(Hf==1&&Ht==0&&S==5)
      T1=1;
      T3=1;
      T5=1;
    end
    if(Hf==1&&Ht==0&&S==6)
      T1=0;
      T3=0;
      T5=0;
    end
%------------------------------------------------------------------------%
    if(Hf==1&&Ht==-1&&S==1)
        T1=1;
        T3=1;
        T5=0;
    end
    if(Hf==1&&Ht==-1&&S==2)
      T1=1;
      T3=0;
      T5=0;
    end
    if(Hf==1&&Ht==-1&&S==3)
      T1=1;
```

```
      T3=0;
      T5=1;
    end
    if(Hf==1&&Ht==-1&&S==4)
      T1=0;
      T3=0;
      T5=1;
    end
    if(Hf==1&&Ht==-1&&S==5)
      T1=0;
      T3=1;
      T5=1;
    end
    if(Hf==1&&Ht==-1&&S==6)
      T1=0;
      T3=1;
      T5=0;
    end

 %-----------------------------------------------------------------------%

     if(Hf==0&&Ht==1&&S==1)
        T1=0;
        T3=0;
        T5=1;
    end
    if(Hf==0&&Ht==1&&S==2)
      T1=0;
      T3=1;
      T5=1;
    end
    if(Hf==0&&Ht==1&&S==3)
      T1=0;
      T3=1;
      T5=0;
    end
    if(Hf==0&&Ht==1&&S==4)
      T1=1;
      T3=1;
      T5=0;
    end
    if(Hf==0&&Ht==1&&S==5)
      T1=1;
      T3=0;
      T5=0;
    end
    if(Hf==0&&Ht==1&&S==6)
      T1=1;
      T3=0;
      T5=1;
    end
%-----------------------------------------------------------------------%

    if(Hf==0&&Ht==0&&S==1)
        T1=0;
        T3=0;
        T5=0;
    end
```

```
    if(Hf==0&&Ht==0&&S==2)
      T1=1;
      T3=1;
      T5=1;
    end
    if(Hf==0&&Ht==0&&S==3)
      T1=0;
      T3=0;
      T5=0;
    end
    if(Hf==0&&Ht==0&&S==4)
      T1=1;
      T3=1;
      T5=1;
    end
    if(Hf==0&&Ht==0&&S==5)
      T1=0;
      T3=0;
      T5=0;
    end
    if(Hf==0&&Ht==0&&S==6)
      T1=1;
      T3=1;
      T5=1;
    end
  %-------------------------------------------------------------------------%

    if(Hf==0&&Ht==-1&&S==1)
        T1=0;
        T3=1;
        T5=0;
    end
    if(Hf==0&&Ht==-1&&S==2)
      T1=1;
      T3=1;
      T5=0;
    end
    if(Hf==0&&Ht==-1&&S==3)
      T1=1;
      T3=0;
      T5=0;
    end
    if(Hf==0&&Ht==-1&&S==4)
      T1=1;
      T3=0;
      T5=1;
    end
    if(Hf==0&&Ht==-1&&S==5)
      T1=0;
      T3=0;
      T5=1;
    end
    if(Hf==0&&Ht==-1&&S==6)
      T1=0;
      T3=1;
      T5=1;
    end
end
```

```
%-------------------------------------------------------------------------%
if(Wr>(50*pi))% Super-Synchronous
%-------------------------------------------------------------------------%
    if(Hf==1&&Ht==1&&S==1)
        T1=1;
        T3=0;
        T5=1;
    end
    if(Hf==1&&Ht==1&&S==2)
      T1=1;
      T3=0;
      T5=0;
    end
    if(Hf==1&&Ht==1&&S==3)
      T1=1;
      T3=1;
      T5=0;
    end
    if(Hf==1&&Ht==1&&S==4)
      T1=0;
      T3=1;
      T5=0;
    end
    if(Hf==1&&Ht==1&&S==5)
      T1=0;
      T3=1;
      T5=1;
    end
    if(Hf==1&&Ht==1&&S==6)
      T1=0;
      T3=0;
      T5=1;
    end

%-------------------------------------------------------------------------%
    if(Hf==1&&Ht==0&&S==1)
        T1=1;
        T3=1;
        T5=1;
    end
    if(Hf==1&&Ht==0&&S==2)
      T1=0;
      T3=0;
      T5=0;
    end
    if(Hf==1&&Ht==0&&S==3)
      T1=1;
      T3=1;
      T5=1;
    end
    if(Hf==1&&Ht==0&&S==4)
      T1=0;
      T3=0;
      T5=0;
    end
    if(Hf==1&&Ht==0&&S==5)
      T1=1;
```

```
      T3=1;
      T5=1;
    end
    if(Hf==1&&Ht==0&&S==6)
      T1=0;
      T3=0;
      T5=0;
    end
%----------------------------------------------------------------------%
    if(Hf==1&&Ht==-1&&S==1)
        T1=1;
        T3=0;
        T5=0;
    end
    if(Hf==1&&Ht==-1&&S==2)
      T1=1;
      T3=1;
      T5=0;
    end
    if(Hf==1&&Ht==-1&&S==3)
      T1=0;
      T3=1;
      T5=0;
    end
    if(Hf==1&&Ht==-1&&S==4)
      T1=0;
      T3=1;
      T5=1;
    end
    if(Hf==1&&Ht==-1&&S==5)
      T1=0;
      T3=0;
      T5=1;
    end
    if(Hf==1&&Ht==-1&&S==6)
      T1=1;
      T3=0;
      T5=1;
    end

 %----------------------------------------------------------------------%

     if(Hf==0&&Ht==1&&S==1)
        T1=0;
        T3=0;
        T5=1;
    end
    if(Hf==0&&Ht==1&&S==2)
      T1=1;
      T3=0;
      T5=1;
    end
    if(Hf==0&&Ht==1&&S==3)
      T1=1;
      T3=0;
      T5=0;
    end
    if(Hf==0&&Ht==1&&S==4)
```

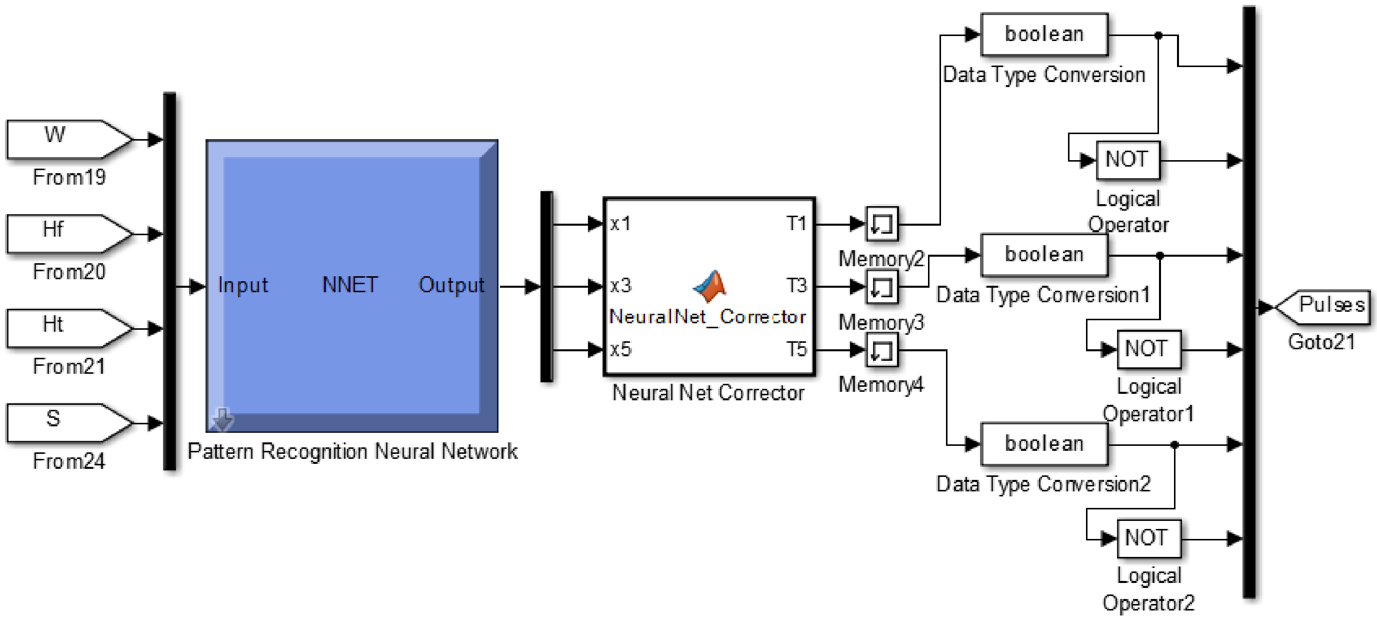
```
      T1=1;
      T3=1;
      T5=0;
    end
    if(Hf==0&&Ht==1&&S==5)
      T1=0;
      T3=1;
      T5=0;
    end
    if(Hf==0&&Ht==1&&S==6)
      T1=0;
      T3=1;
      T5=1;
    end
%------------------------------------------------------------------------%

    if(Hf==0&&Ht==0&&S==1)
        T1=0;
        T3=0;
        T5=0;
    end
    if(Hf==0&&Ht==0&&S==2)
      T1=1;
      T3=1;
      T5=1;
    end
    if(Hf==0&&Ht==0&&S==3)
      T1=0;
      T3=0;
      T5=0;
    end
    if(Hf==0&&Ht==0&&S==4)
      T1=1;
      T3=1;
      T5=1;
    end
    if(Hf==0&&Ht==0&&S==5)
      T1=0;
      T3=0;
      T5=0;
    end
    if(Hf==0&&Ht==0&&S==6)
      T1=1;
      T3=1;
      T5=1;
    end
  %------------------------------------------------------------------------%

    if(Hf==0&&Ht==-1&&S==1)
        T1=0;
        T3=1;
        T5=1;
    end
    if(Hf==0&&Ht==-1&&S==2)
      T1=0;
      T3=0;
      T5=1;
    end
```

```
    if(Hf==0&&Ht==-1&&S==3)
      T1=1;
      T3=0;
      T5=1;
    end
    if(Hf==0&&Ht==-1&&S==4)
      T1=1;
      T3=0;
      T5=0;
    end
    if(Hf==0&&Ht==-1&&S==5)
      T1=1;
      T3=1;
      T5=0;
    end
    if(Hf==0&&Ht==-1&&S==6)
      T1=0;
      T3=1;
      T5=0;
    end
end
```

12 . MATLAB Function and Simulink Block for Neural Network and Neural Network Corrector:




```
function [T1,T3,T5] = NeuralNet_Corrector(x1,x3,x5)
```

```
if(x1>=0.2)
    T1=1;
else
    T1=0;
end
if(x3>=0.2)
    T3=1;
else
    T3=0;
end
if(x5>=0.2)
    T5=1;
else
    T5=0;
end
```