

UNIVERSITÉ DE PICARDIE JULES VERNE

Grado en Ingeniería en Electrónica Industrial y Automática

# ANÁLISIS DE LA SIMULACIÓN DEL DFIG CON CONVERTIDOR BACK-TO-BACK.

Autor:

Rodríguez Cordero, Estelí.

Tutor:

Henao Fernández, Humberto  
Laboratoire des Technologies

Innovantes. UPJV

Amiens, enero de 2015.



## RESUMEN

En el siguiente trabajo se realiza el estudio de la simulación propuesta de un Generador de Inducción Doblemente Alimentado (*Double Fed Induction Generator, DFIG*). El objetivo es entender el sistema para adaptarlo a un generador que se encuentra en el laboratorio de la Universidad donde se ha realizado la presente investigación.

La simulación sobre la cual está basado el trabajo, está desarrollada en el entorno de programación visual *Simulink*, Matlab. La misma está enfocada en la parte de control de los convertidores usados en el generador. El estudio se basó en la simulación del sistema así como la consulta de bibliografía.

Se hace un análisis de cada bloque que lo integra y finalmente se resume en un diagrama de flujo. Cambiando los parámetros que difieren con nuestro modelo, se podrá implementar este tipo de control en el generador real, hacer pruebas de laboratorio sobre el mismo, así como posibles correcciones.

## ABSTRACT

*In the following project I presented the simulation proposal of a Double Fed Induction Generator, DFIG. The objective is to understand the system to be able to adapt a generator that is found in the University Laboratory which the investigation is taking place.*

*The simulation in what the presentation is based on, is being developed in the visual programming environment Simulink of Matlab. The same one is focused on the control part of the inverter used in the generator. The following study was based on bibliographic system simulation and theoretical consultation documents.*

*An analysis of each block that integrates and finally summarized in a flowchart is performed. Changing the parameters that differ our model, it may implement this type of control in the real generator, performing laboratory testing on it and also possible corrections.*

## PALABRAS CLAVES

DFIG, Convertidor Back to Back, SVPWM, Control, Transformación de Park.

## KEYWORDS

DFIG, Back to back Converter, SVPWM, Control, Park Transformation.



**ÍNDICE**

RESUMEN .....	3
ABSTRACT .....	3
PALABRAS CLAVES .....	3
KEYWORDS .....	3
INTRODUCCIÓN .....	7
OBJETIVOS .....	7
DESARROLLO .....	9
DFIG_MOTOR_BACK_TO_BACK_VGRID_SIMU .....	9
PRIMERA ETAPA .....	11
SEGUNDA ETAPA .....	30
BLOQUES EXTERIORES: .....	61
SPEED-COUPLE. ....	61
OTROS PARÁMETROS. ....	63
DIAGRAMA DE BLOQUES SIMPLIFICADO .....	65
CONCLUSIONES Y TRABAJO FUTURO .....	67
BIBLIOGRAFÍA .....	69
ANEXOS .....	71
MARCOS DE REFERENCIA. TRANSFORMACIÓN DE COORDENADAS .....	71
SVPWM .....	73
GLOSARIO DE TÉRMINOS .....	75



## INTRODUCCIÓN

En una máquina de inducción, como es el caso del *DFIG*, el par ( $T$ ), la velocidad ( $\omega$ ) y el flujo ( $\varphi$ ), varían de forma simultánea al modificar la tensión y frecuencia de alimentación. El comportamiento anterior, indica que estos parámetros están acoplados. Se ha demostrado que si se logra dividir la corriente de alimentación en dos componentes perpendiculares entre sí, el motor de inducción podría controlarse como uno de corriente continua, donde el flujo y el par están desacoplados.

Si se utiliza un convertidor *Back-to-Back* controlado vectorialmente, se garantiza la generación de energía a la frecuencia y tensión nominal de la red independientemente de la velocidad del *rotor*.

La importancia del control de los convertidores radica en que mediante el control de potencia activa, se puede hacer trabajar el aerogenerador en su *Punto de Extracción de Máxima Potencia (MPPT)* [2]. Se provee al generador de un margen específica de generación menor al disponible y el de una respuesta rápida ante cambios en la frecuencia [5]. Se mantiene a la turbina bajo condiciones apropiadas de trabajo y se minimiza la carga mecánica [1].

## OBJETIVOS

El presente trabajo está motivado porque se cuenta con un aerogenerador sobre el cual se están realizando investigaciones y al mismo tiempo, se tiene un posible sistema de control. Esta simulación se ha estudiado y analizado para en un futuro adaptarlo a nuestro sistema real.



## DESARROLLO

### DFIG\_MOTOR\_BACK\_TO\_BACK\_VGRID\_SIMU

Cuando se provee a un *DFIG* de un convertidor *Back-to-Back*, sus prestaciones aumentan, dado que permite absorber e inyectar *Potencia Reactiva* a la red. *Control Vectorial de Flujo Orientado* es implementado para desacoplar el flujo y el par en el generador.

El objetivo fundamental de este convertidor, es lograr un control preciso sobre la *Potencia Activa* (gobernada por la potencia que se captura en las aspas del aerogenerador y el control sobre el *rotor*) y la *Potencia Reactiva* (condicionada por el control de *GSC* y el *RSC*). Gracias a los dispositivos de electrónica de potencia actuales, se ha logrado el control preciso de los parámetros anteriores [1].

El modelo sobre el cual se realiza el estudio es el mostrado en la *Figura 1*. Para una mejor comprensión del mismo se ha realizado el estudio en dos etapas y se ha enumerado cada bloque que lo integra.

En ocasiones los mismos bloques son usados varias veces o con ligeras modificaciones. En esos casos se ha obviado la descripción para evitar redundancias.

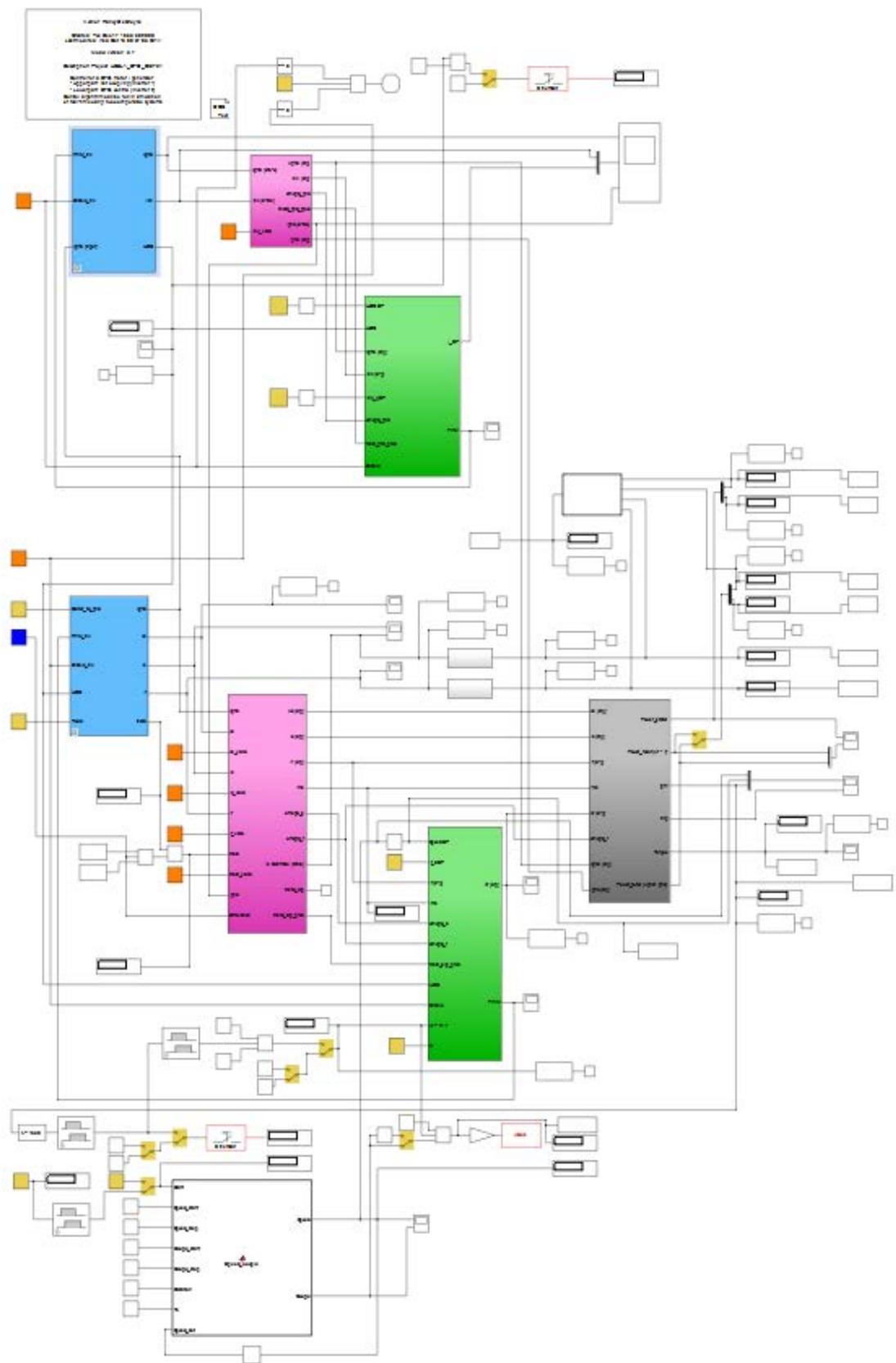


Figura 1: Dfig\_Motor\_Back\_To\_Back\_Vgrid\_Simu.

### PRIMERA ETAPA

La primera etapa está formada por los bloques “Net coupling model, Preprocessing y Control”. Implementa el cálculo del control del inversor del lado de la red (*Grid Side Converter*). El objetivo principal del GSC es mantener constante la tensión en el condensador de enlace, independientemente del valor de la potencia en el rotor y el sentido en que esta fluya [1].

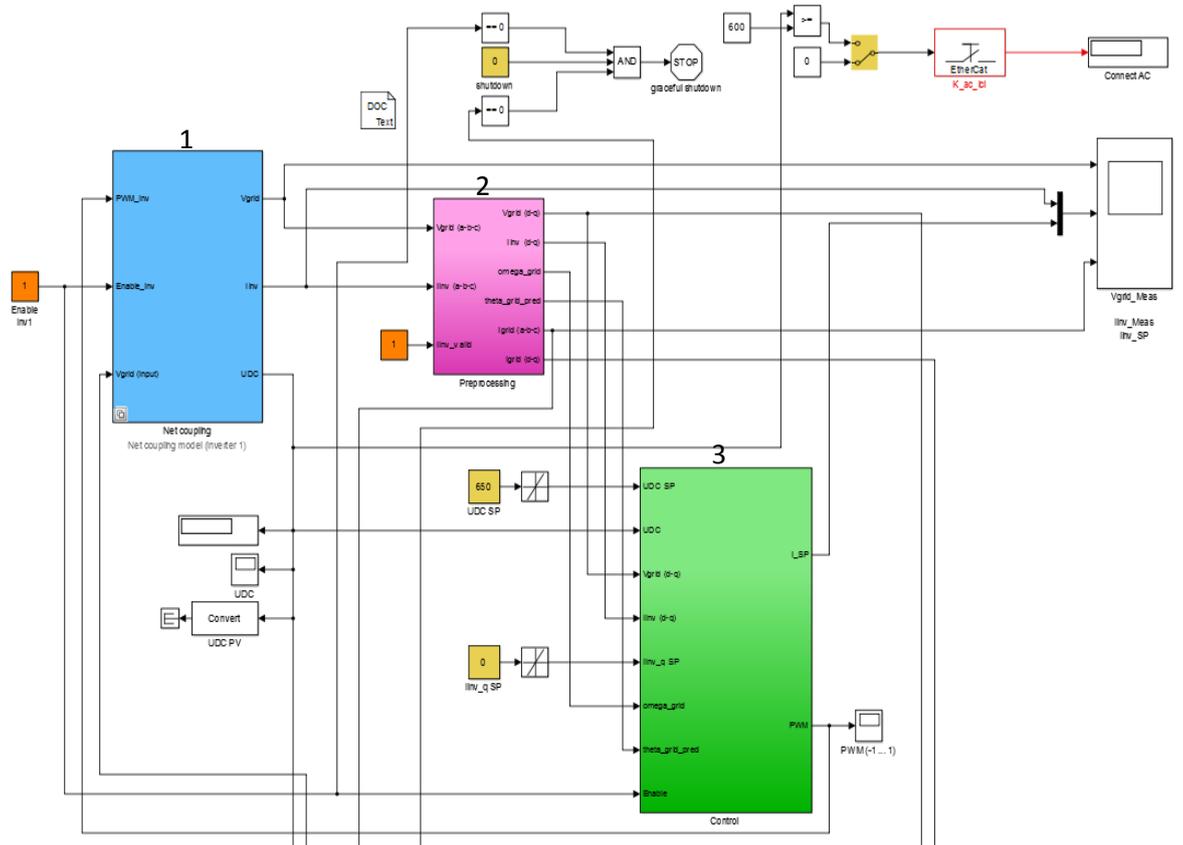


Figura 2: Primera Etapa.

## 1. Net coupling model (Inverter 1).

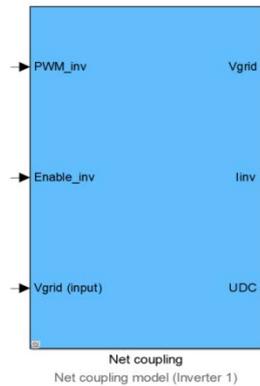


Figura 3: Net coupling model.

En este bloque se fija el valor de la tensión del condensador que funciona como nexo entre los dos convertidores y se encuentra el inversor del lado del *stator*.(GSC).

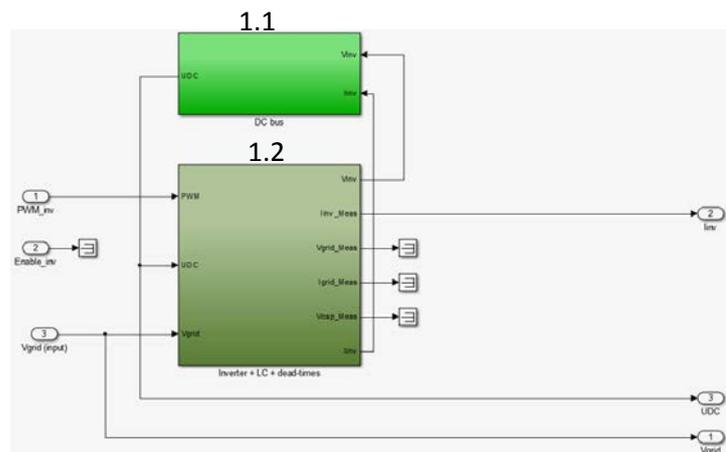


Figura 4: Net coupling model.

### 1.1. DC bus (*DC\_Link*).

Se tiene un condensador que funciona como nexo entre el inversor y el rectificador del convertidor *Back-to-Back*, (*Grid Side Converter* y *Rotor Side Converter*) (La conversión CA/CA se hace de forma indirecta). La ventaja de utilizar un condensador como enlace entre los convertidores es que existe un desacople entre la red y el generador [1].

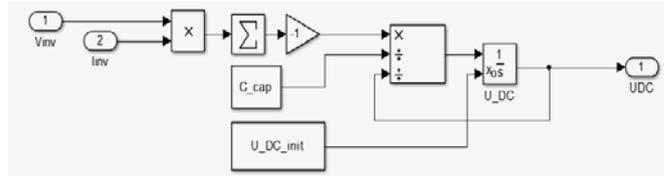


Figura 5: DC bus.

La tensión  $UDC$  generada en este bloque, es luego utilizada en el bloque de “Control2” para generar la señal PWM que controla el RSC.

Cálculo de la tensión en el condensador:

$$I_{inv} \cdot V_{ins} = P_{inv} \tag{1}$$

$$\frac{P_{inv}}{C_{cap} \cdot UDC} = \frac{1}{C_{bus}} \cdot I_{inv} \tag{2}$$

$$\frac{1}{C_{bus}} \int i_c \cdot dt = UDC \tag{3}$$

### 1.2. Inverter + LC + dead-times

El objetivo del GSC, es mantener la tensión  $UDC$  constante e independiente de la magnitud o sentido de la potencia activa en el rotor.

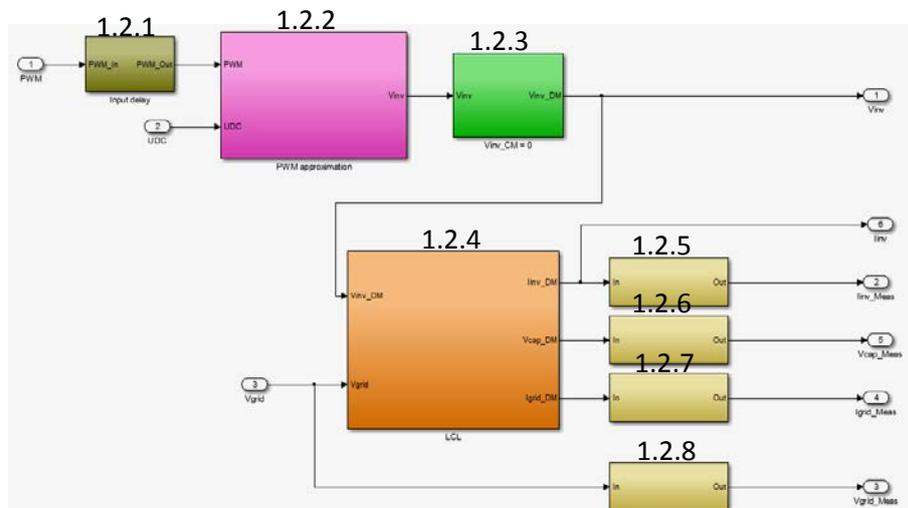


Figura 6: Inverter+LCL+dead-times.

### 1.2.1. Input delay.

Se introduce un retraso en la señal PWM para compensar el retardo del sistema en lazo cerrado.

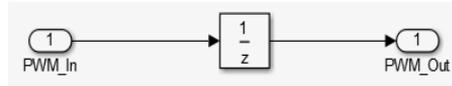


Figura 7: Retardo de entrada.

### 1.2.2. PWM approximation.

Esta señal PWM, junto con el valor de  $UDC$ , forman la tensión de salida del GSC.

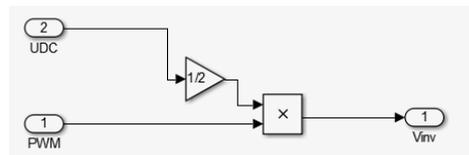


Figura 8: PWM aproximación.

En la salida tengo una señal de tensión gobernada por el control PWM (igual forma de onda pero distinto nivel de tensión).

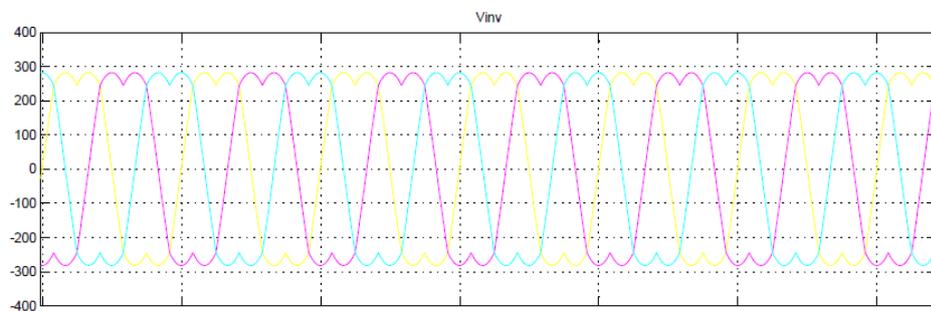


Figura 9: Tensión en el inversor obtenida con el PWM.

### 1.2.3. $V_{inv\_CM}=0$ .

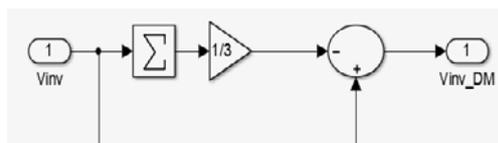


Figura 10:  $V_{inv\_DM}$ .

La tensión en el inversor queda fijada por la señal PWM y la tensión en el DC\_link.

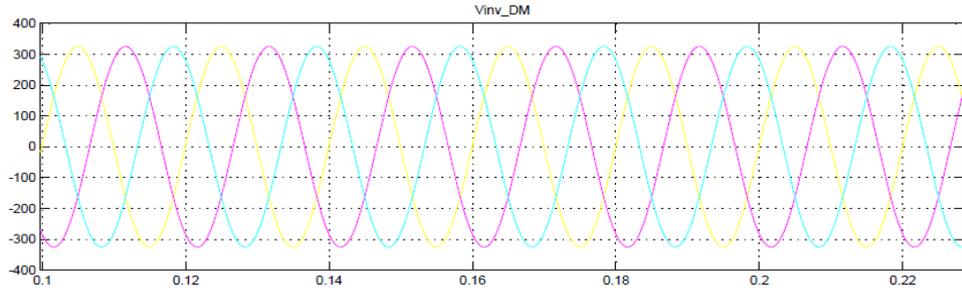


Figura 11: Tensión en el inversor conectado a la red.

#### 1.2.4. LCL.

Para disminuir los armónicos producidos por la conmutación de los transistores, se utiliza un filtro LCL.

Este se encuentra conectado entre la salida del GSC y la red. Conociendo el valor de tensión del inversor ( $V_{inv\_DM}$ ), se puede calcular la corriente que circula por el mismo ( $I_{inv\_DM}$ ). Los valores de  $V_{cap\_DM}$  y  $I_{grid\_DM}$ , no son usados en la simulación.

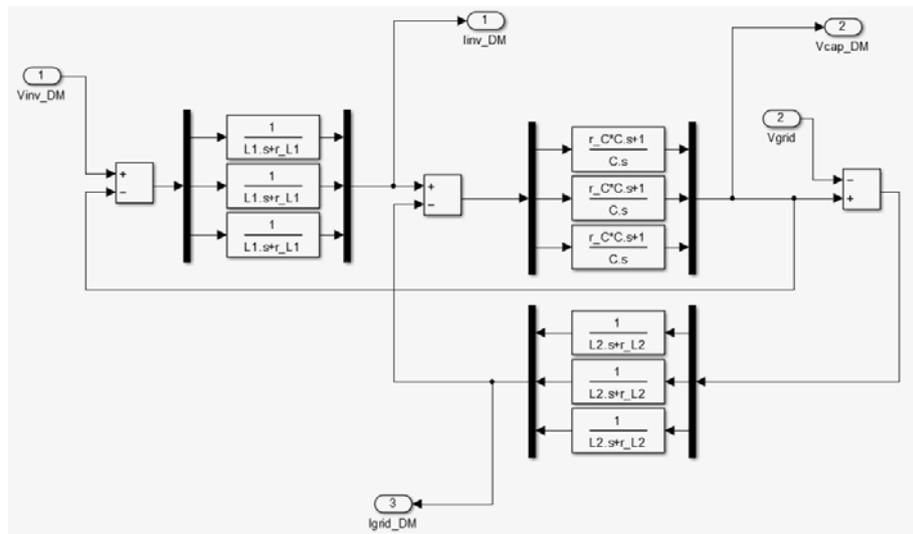
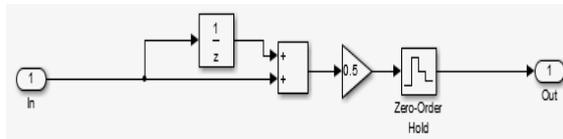


Figura 12: Filtro.

$$I_{invDM} = \frac{V_{invDM}}{L_1 \cdot s + r_{L1}} \quad (4)$$

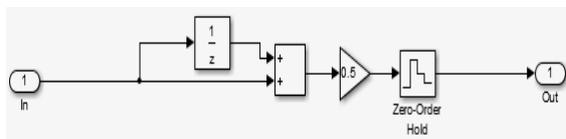
### 1.2.5. Averaging for fast PWM.

Se recalcula la  $I_{inv\_DM}$ , teniendo en cuenta el retardo que se ha introducido en la señal PWM.



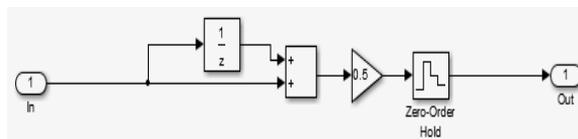
*Figura 13: Averaging for fast PWM.*

### 1.2.6. Averaging for fast PWM1.



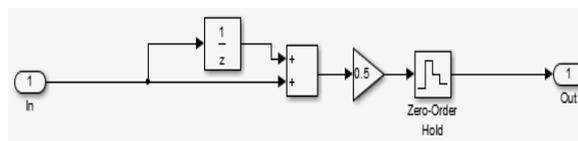
*Figura 14: Averaging for fast PWM.*

### 1.2.7. Averaging for fast PWM2.



*Figura 15: Averaging for fast PWM.*

### 1.2.8. Averaging for fast PWM3.



*Figura 16: Averaging for fast PWM.*

## 2. Preprocessing.

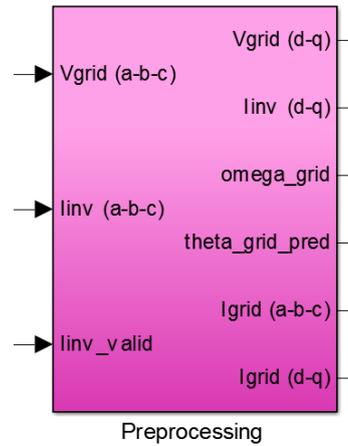


Figura 17: Preprocessing.

En este bloque, de manera general, se hace la transformación del sistema en tres coordenadas ( $a - b - c$ ), a un sistema de dos coordenadas ( $d - q$ ). El objetivo es disminuir la complejidad del control. Se proyecta el sistema trifásico en los ejes de un marco de referencia que gira a la velocidad de la red  $\omega_{grid}$ .

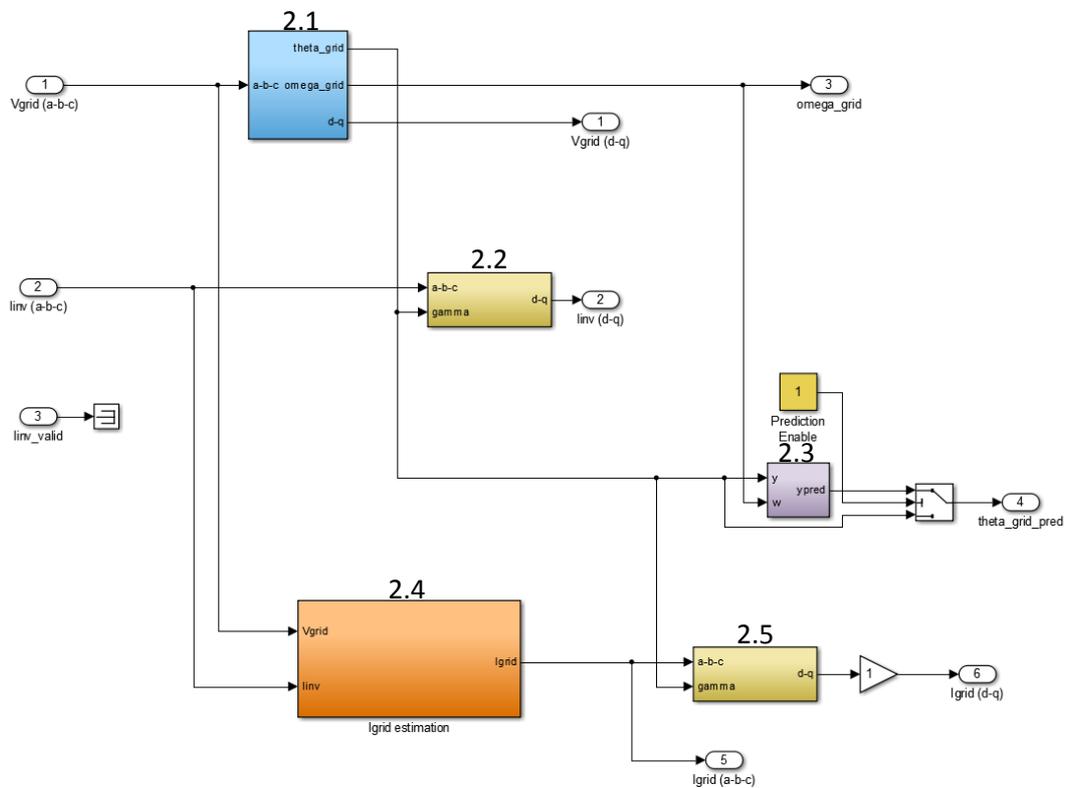


Figura 18: Preprocessing.

## 2.1. Theta (PLL)

Se transforma la tensión de la red de coordenadas  $(a - b - c)$  a coordenadas  $(d - q)$ . ( $V_{grid(a-b-c)} \rightarrow V_{grid(d-q)}$ ).

El PLL (*Phase Locked Loop*) se utiliza para determinar el ángulo y la velocidad de la tensión de la red. En este caso el marco de referencia está alineado con la tensión de la red (eje  $d$  del marco de referencia rotativo, se sincroniza con la red). El eje  $q$  se lleva a cero para lograr un factor de potencia unitario [5].

Lograr factor de potencia unitario permite la transferencia de potencia activa en ambas direcciones pero cero intercambio de potencia reactiva.

El valor del ángulo  $\theta_{grid}$  quedará establecido cuando, mediante la realimentación, el valor de la componente  $q$  sea nulo. Y es, a su vez, el ángulo utilizado en la *Transformación de Park*. De esta manera, el marco de referencia rotativo y  $V_{grid(d-q)}$ , quedan sincronizados.

La tensión  $V_{grid(d-q)}$ , es luego utilizada en los bloques de "Control" para generar la señal PWM, y en el "Postprocessing" para calcular  $Power_{rotor}(V_{grid} - I_{grid})$ .

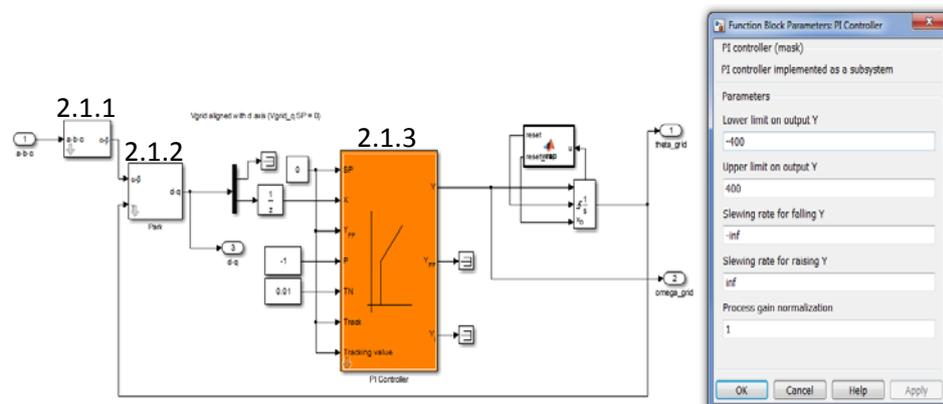


Figura 19: Cálculo de Theta.

% Mantiene el valor de  $\theta_{grid}$  en  $2\pi$ .

```
function [reset,reset_val] = wrap(u)
```

% This block supports the Embedded MATLAB subset.

```
reset = 0;
reset_val = 0;
```

```
if (u > 2*pi)
    reset = 1;
    reset_val = u - 2*pi;
```

```
end
```

```
if (u < 0)
    reset = 1;
    reset_val = u + 2*pi;
end
```

### 2.1.1. Transformada de Fortescue.

Se transforma de  $(a - b - c) \rightarrow (\alpha - \beta)$ .

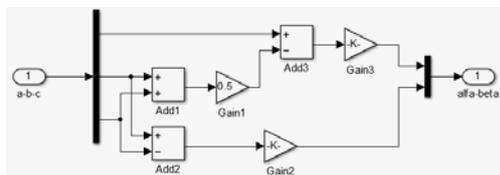


Figura 20: Transformada de Fortescue.

### 2.1.2. Transformada de Park.

Se transforma de  $(\alpha - \beta) \rightarrow (d - q)$ .

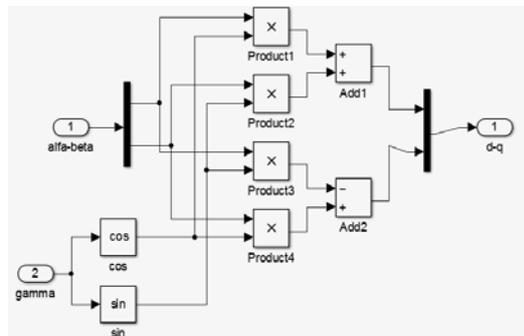


Figura 21: Transformada de Park.

### 2.1.3. PI Controller.

Se alinea la tensión  $V_{grid(a-b-c)}$  con con el eje  $d$ . Esto implica que la componente  $q$  tiene que valer cero. Un regulador  $PI$  es el encargado de mantener el valor en la referencia. En su parte más externa se observa la acción Proporcional.

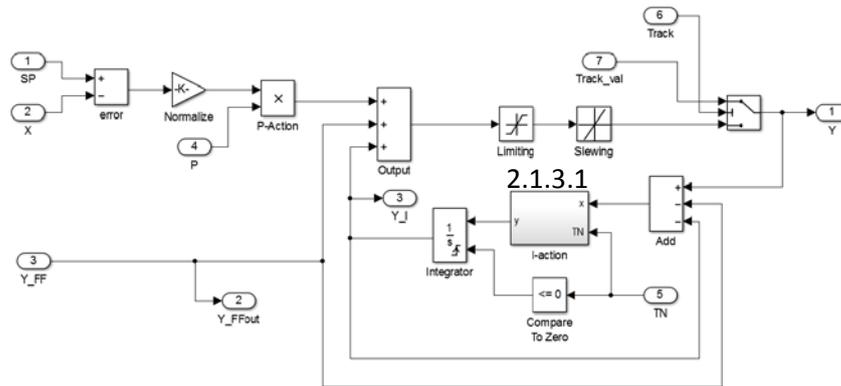


Figura 22: Controlador PI.

2.1.3.1. I-action

Acción Integral del regulador PI.

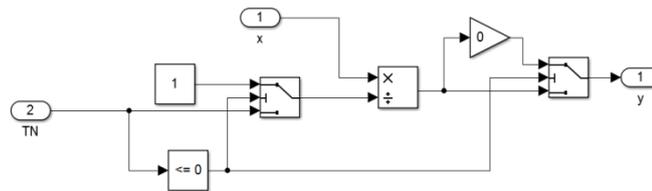


Figura 23: Acción integral.

2.2. Subsystem2.

Transformación de Fortescue + Transformación de Park.

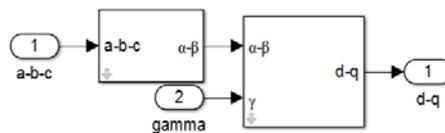


Figura 24: Subsystem2.

2.3. Subsystem.

Predicción de  $\theta_{grid}$ .

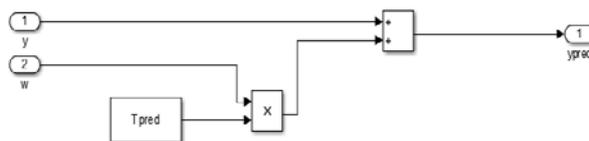
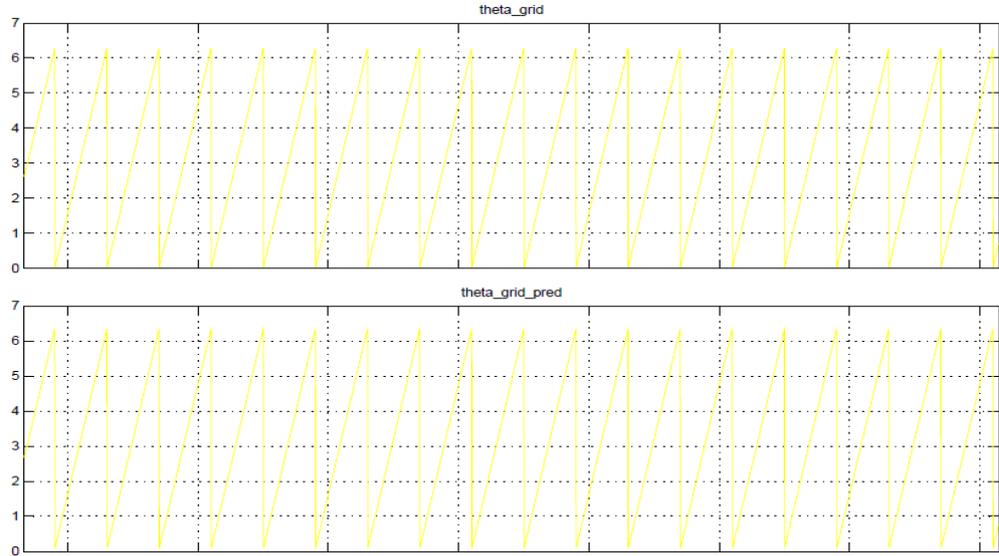


Figura 25: Subsystem.

La salida del sistema es el valor utilizado de  $\theta$ , siempre y cuando la predicción del mismo esté habilitada.

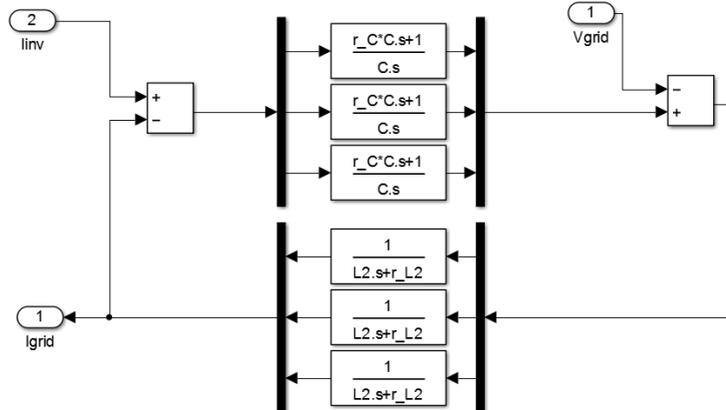
En la *Figura 26*, se observa que no hay diferencia entre  $\theta_{grid}$  (gráfica superior) y  $\theta_{grid\_pred}$  (gráfica inferior).



*Figura 26: theta\_grid y theta\_grid\_pred.*

2.4. Igrid estimation.

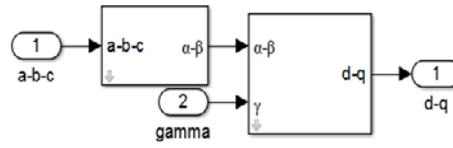
Conocida la tensión de la red y la corriente en el inversor, se hace una estimación de la corriente de la red. Posteriormente es utilizada en los bloques de *Preprocessing* en coordenadas  $(a - b - c)$  y *Postprocessing* en coordenadas  $(d - q)$ .



*Figura 27: Igrid estimación.*

## 2.5. Subsystem1

*Transformación de Fortescue + Transformación de Park.*



*Figura 28: Subsystem1.*

3. Control.

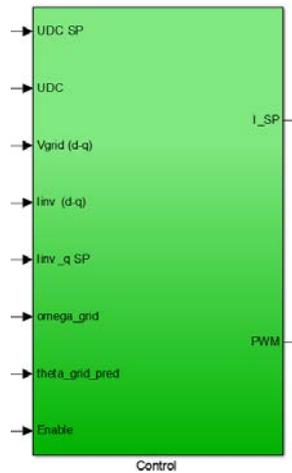


Figura 29: Control.

En este bloque se genera la señal de control del inversor GSC. La técnica utilizada es el control PWM generada con Space Vector Modulation (SVPWM).

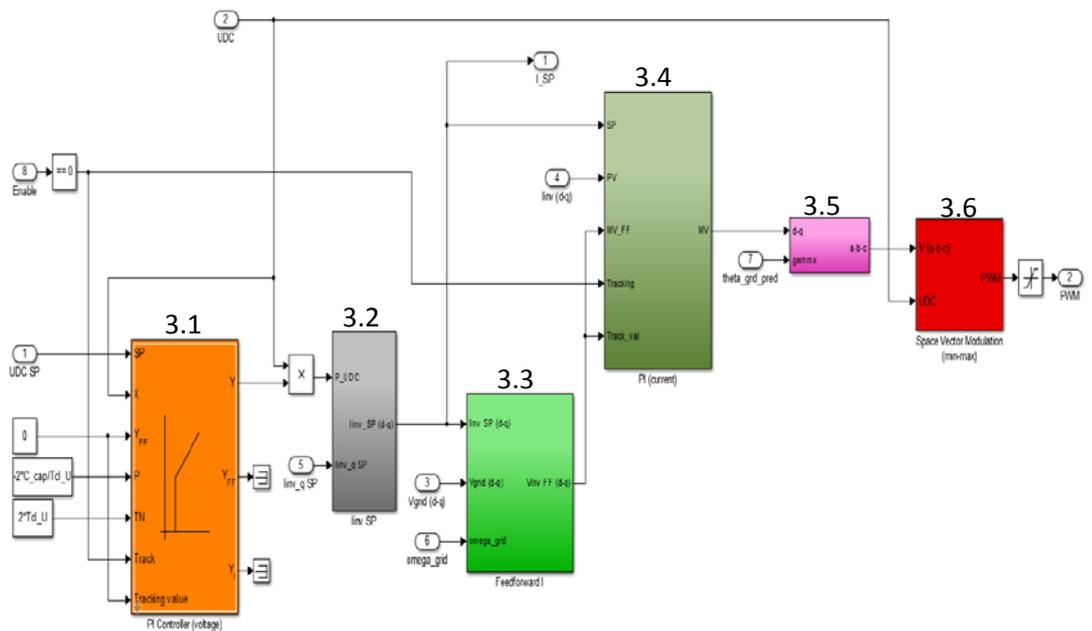


Figura 30: Control.

3.1. PI Controller (voltage).

El objetivo es mantener el valor de la tensión en el “DC\_Link” en su valor de consigna (650V). Para la regulación de *UDC*, un controlador *PI* es utilizado.

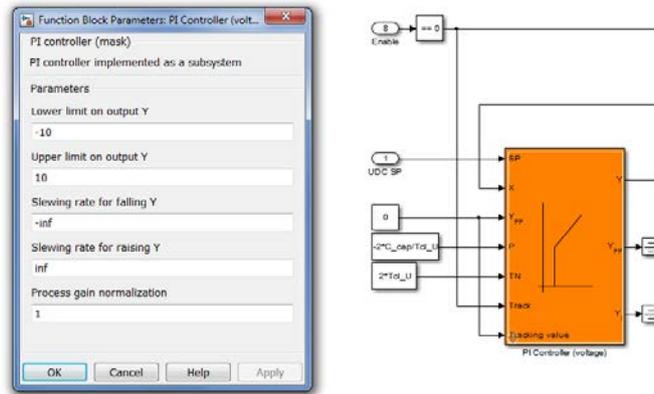


Figura 31: Controlador PI.

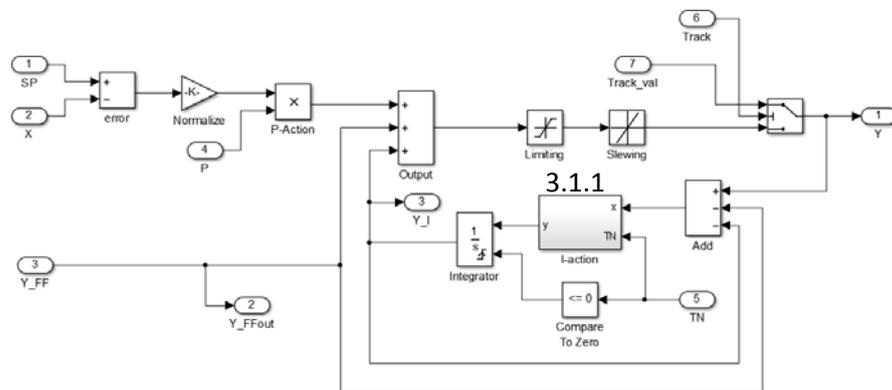


Figura 32: Controlador PI.

### 3.1.1. I-action.

Acción Integral del regulador.

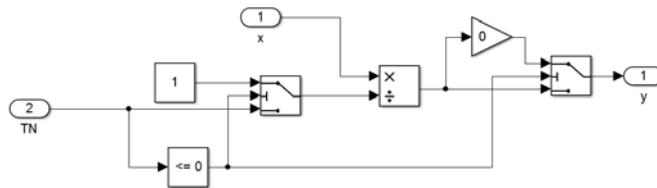
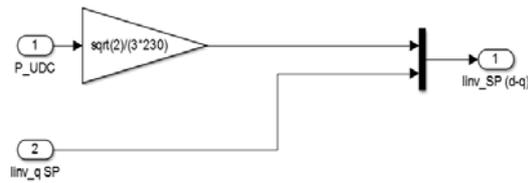


Figura 33: Qcción integral.

### 3.2. linv SP.

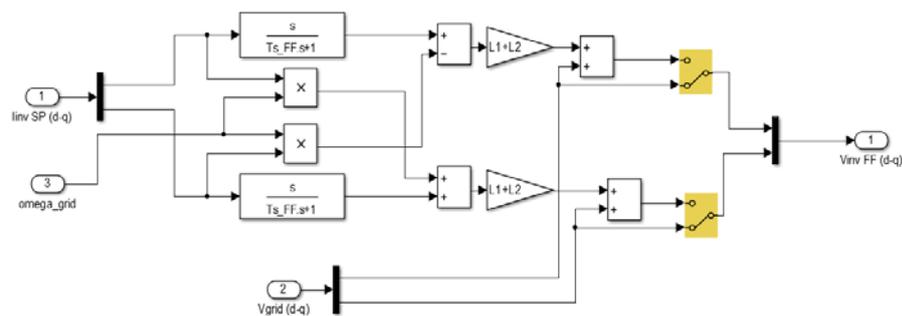
Se genera una consigna de corriente ( $I_{inv-SP(d-q)}$ ) donde la componente  $d$  depende de la tensión en el "DC\_Link" y la componente  $q$  es el valor de  $I_{inv-q}SP = 0$ .

Figura 34: *linv SP*.

### 3.3. Feedforward I.

En este caso el controlador *Feedforward* no es utilizado en la simulación (está deshabilitado). El valor de tensión a la salida corresponde con el valor de la red en coordenadas  $d-q$ :  $V_{invFF(d-q)} = V_{grid(d-q)}$ .

La compensación *Feedforward* se utiliza para cancelar el efecto que ejerce la red en el controlador.

Figura 35: *Feedforward I*.

### 3.4. PI (courrent).

Para la limitación de corriente se divide el control en dos etapas sucesivas. En una primera etapa se encuentra la acción proporcional para eliminar el error en Estado Estacionario. En la segunda etapa o bloque, se encuentra la acción integral del controlador. Es entonces que queda corregido el error del Estado Transitorio. La salida del controlador integral, es realimentada a la parte proporcional.

Con el valor de tensión que resulta de este bloque, se genera posteriormente la señal PWM.

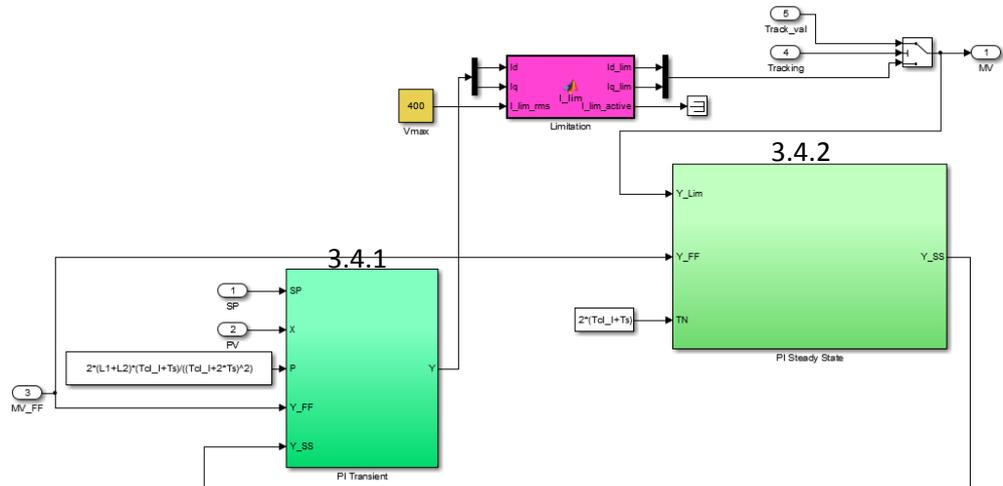


Figura 36: Limitación de corriente.

```
% Limitación de corriente.
function [Id_lim,Iq_lim,I_lim_active] =
I_lim(Id,Iq,I_lim_rms)

I_lim = sqrt(2)*I_lim_rms;
I_amp = sqrt(Id^2+Iq^2);
I_lim_active = 0;

if I_amp ~= 0

    if abs(Iq) >= abs(Id)
        theta = atan(Iq/Id);
    else
        theta = acot(Id/Iq);
    end

    if I_amp >= I_lim
        I_amp = I_lim;
        Id_lim = I_amp * sign(Id) * abs(cos(theta));
        Iq_lim = I_amp * sign(Iq) * abs(sin(theta));
        I_lim_active = 1;
    else
        Id_lim = Id;
        Iq_lim = Iq;
    end

else
    Id_lim = Id;
    Iq_lim = Iq;
end
```

### 3.4.1. PI Transient.

Controla la parte transitoria de la de la corriente del inversor calculada en el bloque "Net coupling", manteniendo su valor en el de referencia fijado por  $I_{inv}SP$ .

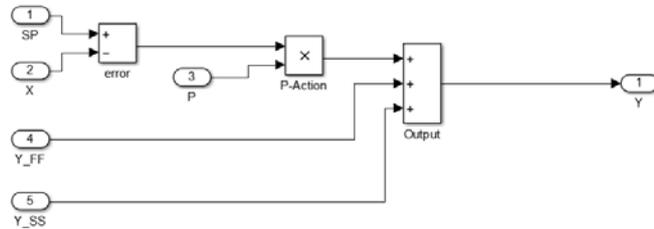


Figura 37: Estado transitorio.

### 3.4.2. PI Steady State.

Controla la parte estacionaria de la de la corriente del inversor calculada en el bloque "Net coupling", manteniendo su valor en el de referencia fijado por  $I_{inv}SP$ .

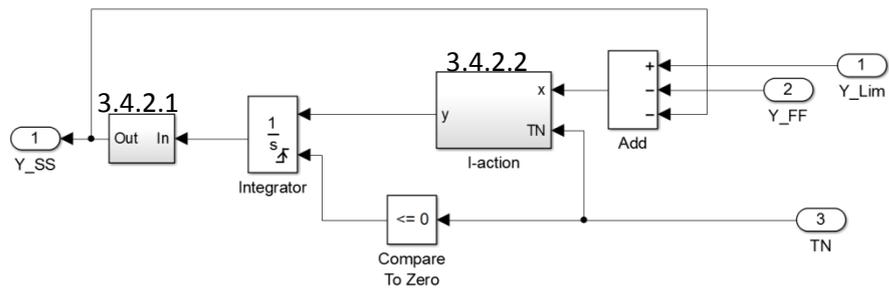


Figura 38: Estado estacionario.

#### 3.4.2.1. Subsystem.

Redondeo de la señal.

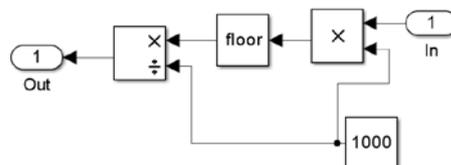


Figura 39: Subsystem.

## 3.4.2.2. I-action.

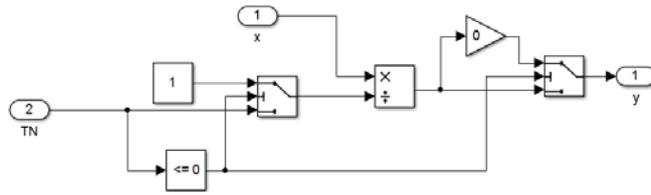


Figura 40: Acción integral.

## 3.5. Subsystem1

Park Inversa + Fortescue Inversa.

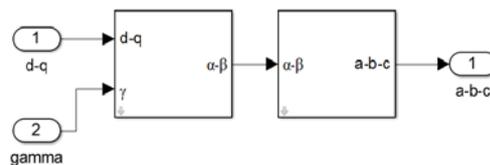


Figura 41: Park inversa + Fortescue inversa.

## 3.6. Space Vector Modulation (min-max).

Se genera la señal PWM para el control del GSC. Obtenida a partir del valor de tensión en “DC\_Link” y la tensión de la red usando la técnica “Min\_Max Method” del SVPWM (Ver Anexos).

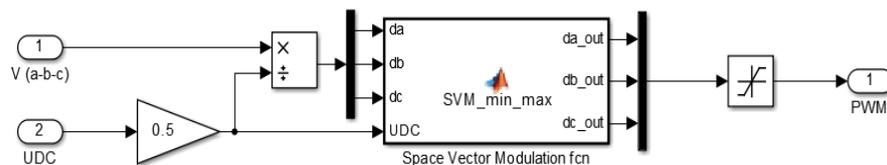


Figura 42: Space Vector Modulation.

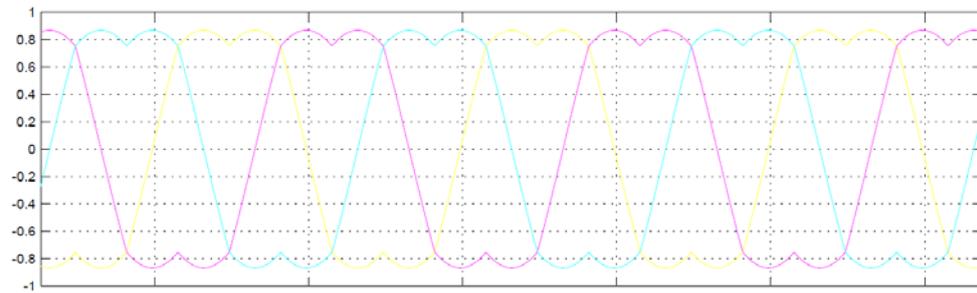
% Cálculo de los vectores para el control SVPWM.

```
function [da_out,db_out,dc_out] = SVM_min_max(da,db,dc,UDC)
```

```
dmin = min([da db dc]);
dmax = max([da db dc]);
```

```
da_out = da - (dmax + dmin) * 0.5;
db_out = db - (dmax + dmin) * 0.5;
dc_out = dc - (dmax + dmin) * 0.5;
```

En la *Figura 43*, se observa la forma de onda de la señal SVPWM.



*Figura 43: Señal SVPWM.*

## SEGUNDA ETAPA

La segunda etapa está formada por los bloques “DFIG, Preprocessing2, Control2 y Postprocessing”. Se encuentran representados el modelo de la máquina y el inversor del lado del rotor (*Rotor Side Converter*). El control en el RSC regula el intercambio de potencia activa y reactiva con la red.

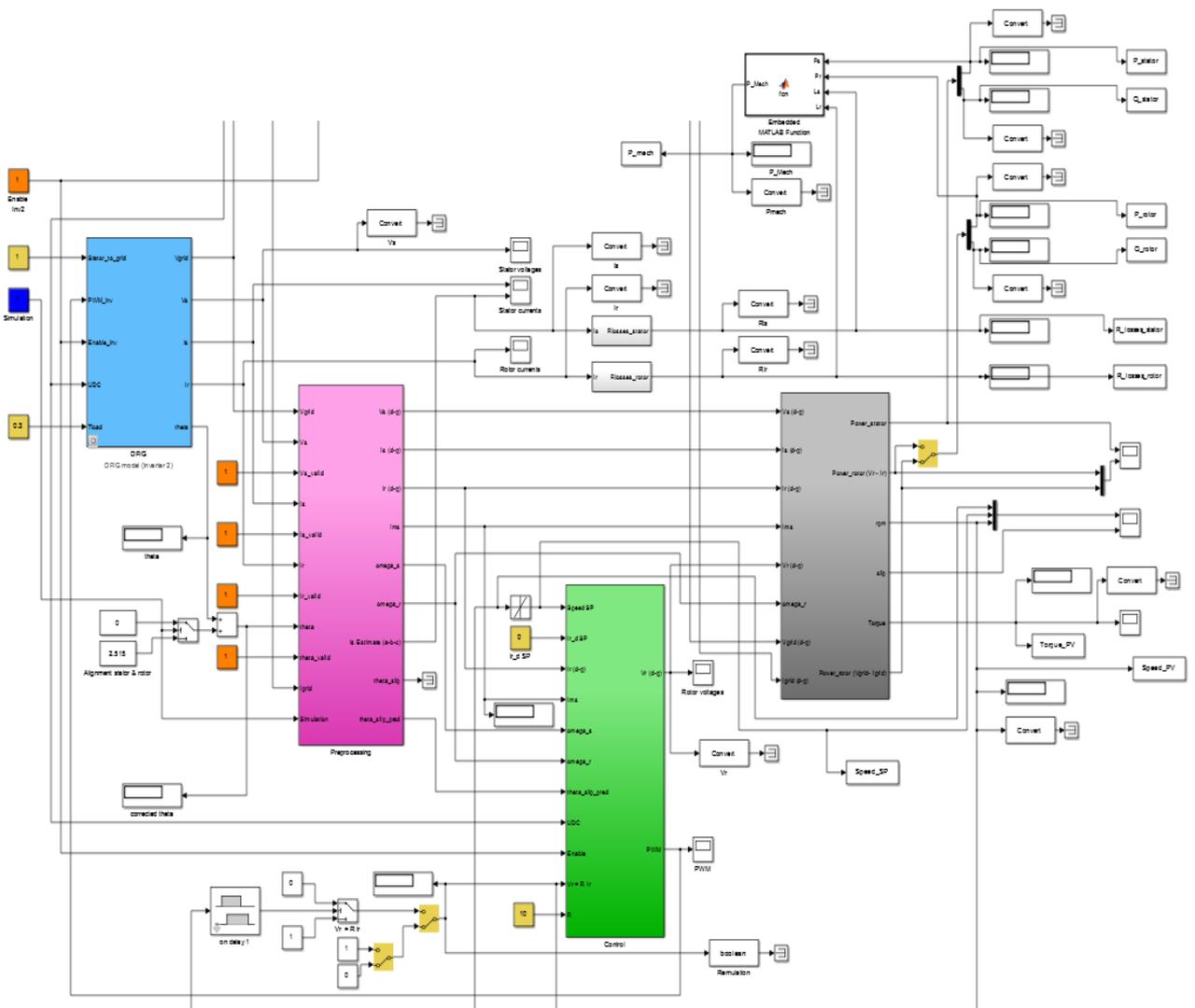


Figura 44: Segunda Etapa.



#### 4.1. Grid voltage.

Se simula la tensión de la red  $V_{grid(d-q)}$  y se calculan la velocidad y ángulo de giro ( $\omega_s$  y  $\theta_s$ ).

El valor de  $V_{grid}$  es utilizado en “Net coupling model”. para calcular la corriente en el inversor; en el bloque de “Control”, para generar la señal PWM y en el “Preprocessing2” para la estimación de la corriente del stator..

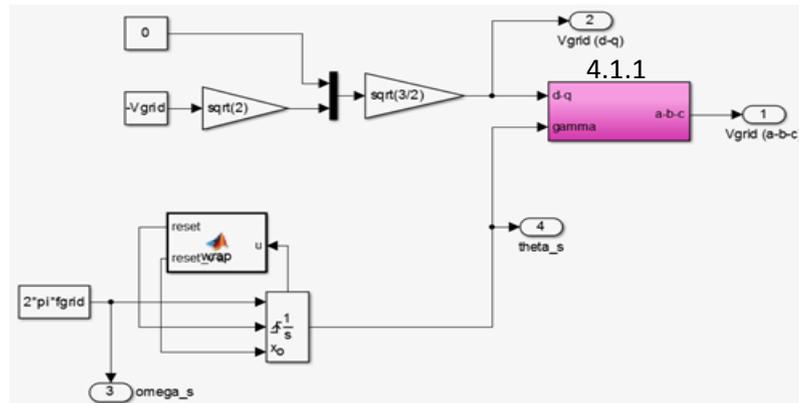


Figura 47: Tensión de la red.

% Mantiene el valor de  $\theta_s$  en  $2\pi$ .

```
function [reset,reset_val] = wrap(u)
% This block supports the Embedded MATLAB subset.

reset = 0;
reset_val = 0;

if (u > 2*pi)
    reset = 1;
    reset_val = u - 2*pi;
end

if (u < 0)
    reset = 1;
    reset_val = u + 2*pi;
end
```

4.1.1. Subsystem 2

Park Inversa + Fortescue Inversa.

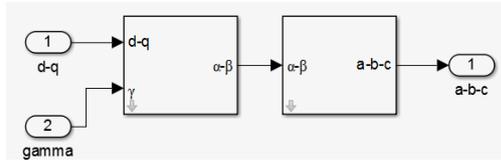


Figura 48: Subsystem 2.

4.2. Rotor voltages.

Conocida la señal PWM que controla el inversor del rotor y el valor de tensión en el “DC\_Link”, se calcula la tensión del rotor ( $V_{r(d-q)}$ ).

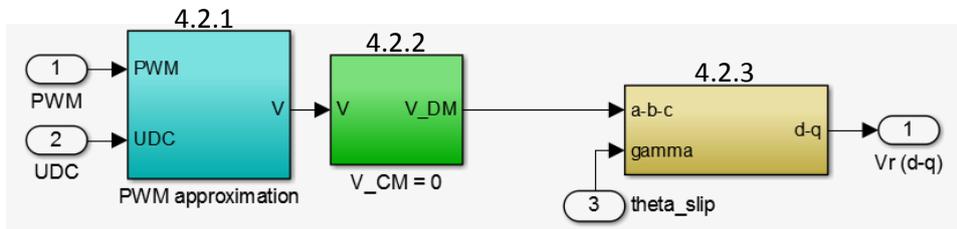


Figura 49: Tensión en el rotor.

4.2.1. PWM Approximation.

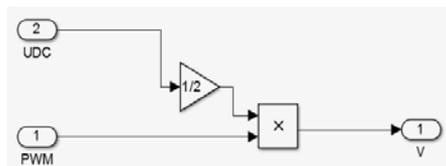


Figura 50: PWM Aproximación.

4.2.2. VC\_M = 0.

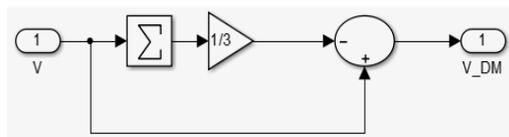


Figura 51: VC\_M.

## 4.2.3. Subsystem4.

Transformación de Fortescue + Transformación de Park.

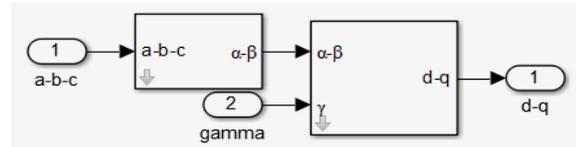


Figura 52: Subsystem4.

## 4.3. Stator connected to the grid.

Conocidos los valores de  $V_{s(d-q)}$ ,  $\omega_{s}$ ,  $V_{r(d-q)}$  y  $\omega_{slip}$ , se calculan los parámetros del *stator*. Pueden haber dos posibilidades: “*Stator connected to the grid*” o “*Stator not connected to the grid*”. En el caso del *stator* conectado a la red, la tensión de ambos coinciden.

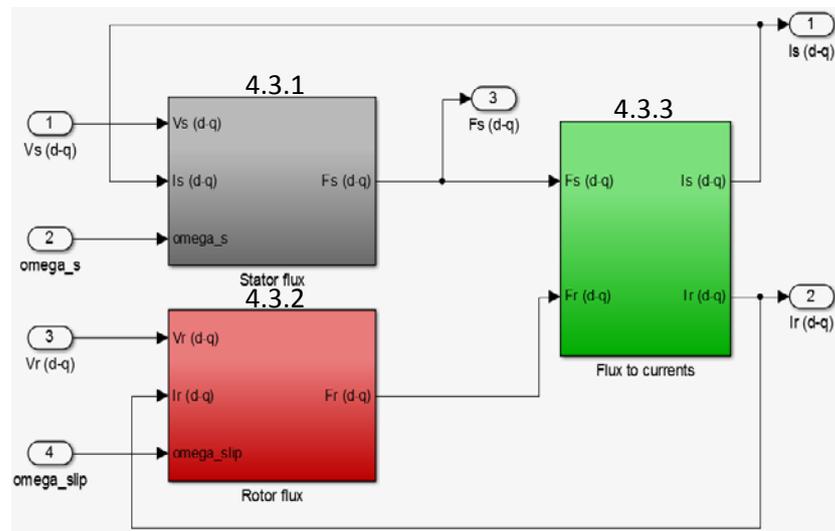


Figura 53: Stator conectado a la red.

## 4.3.1. Stator flux.

Cálculo del flujo del *stator*.

Del modelo dinámico de la máquina de inducción se obtiene [7]:

$$\vec{V}_s = R_s \cdot \vec{I}_s + \frac{d\vec{\varphi}_s}{dt} - j\omega_s \cdot \vec{\varphi}_s \quad (5)$$

Depejando de (5):

$$\overline{\varphi}_s = \int (\overline{V}_s - \overline{I}_s \cdot R_s - \omega_s \cdot \overline{\varphi}_s) \tag{6}$$

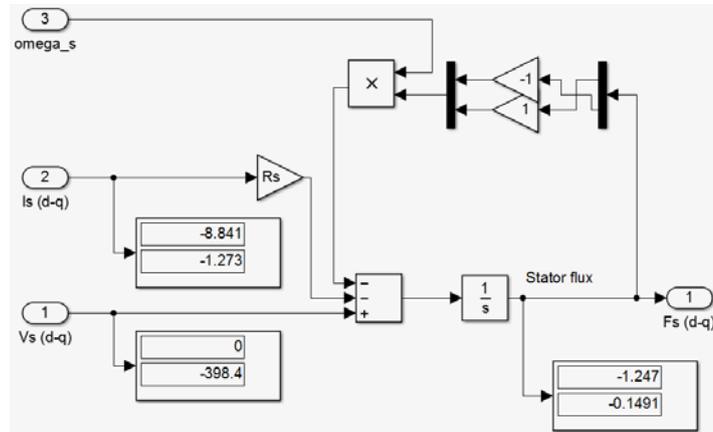


Figura 54: Flujo del stator.

4.3.2. Rotor flux.

Cálculo del flujo del rotor.

Similar al caso anterior, partiendo del modelo dinámico de la máquina de inducción:

$$\overline{V}_r = R_r \cdot \overline{I}_r + \frac{d\overline{\varphi}_r}{dt} + (j\omega_s - j\omega_r) \cdot \overline{\varphi}_r \tag{7}$$

Depejando de (7):

$$\overline{\varphi}_r = \int (\overline{V}_r - \overline{I}_r \cdot R_r + \omega_{slip} \cdot \overline{\varphi}_r) \tag{8}$$

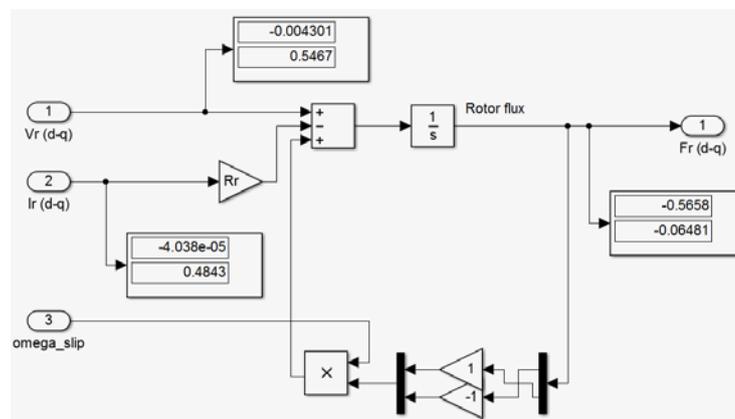


Figura 55: Flujo del rotor.

#### 4.3.3. Flux to currents.

Una vez conocidos los valores de flujos del *rotor* y *stator*, se calculan sus corrientes.

Se supone que el generador no se satura, por lo que el flujo producido en el *rotor/stator* será la suma del flujo que produce la corriente que circula por él más una parte de la producida en el *stator/rotor*:

$$\varphi_s = L_s \cdot I_s + L_h \cdot I_r \quad (9)$$

$$\varphi_r = L_r \cdot I_r + L_h \cdot I_s \quad (10)$$

Despejando  $I_r$  de (9) y (10) e igualando, se obtiene  $I_s$ :

$$\frac{\varphi_s - L_s \cdot I_s}{L_h} = \frac{\varphi_r - L_h \cdot I_s}{L_r} \quad (11)$$

$$I_s = \frac{\varphi_s - \varphi_r \cdot \frac{L_h}{L_r}}{L_s - \frac{L_h^2}{L_r}} \quad (12)$$

De manera similar se calcula  $I_r$ :

$$\frac{\varphi_s - L_m \cdot I_r}{L_s} = \frac{\varphi_r - L_r \cdot I_r}{L_m} \quad (13)$$

$$I_r = \frac{\varphi_r - \varphi_s \cdot \frac{L_h}{L_s}}{\frac{L_h^2}{L_s} - L_r} \quad (14)$$

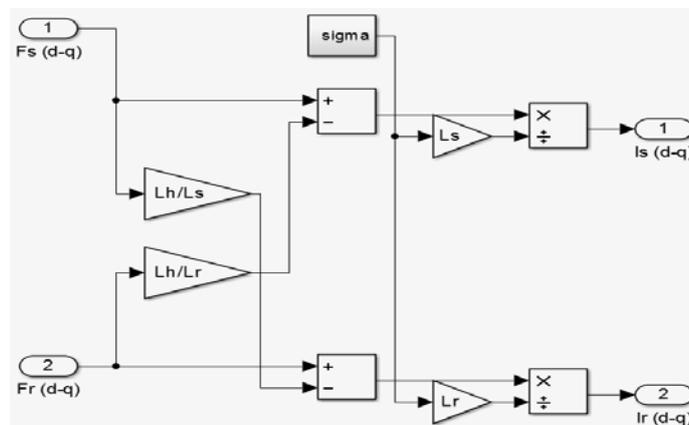


Figura 56: Conversión de flujo a corriente.

## 4.3.3.1. Sigma.

Es una variable intermedia (relación de inductancias).

$$\sigma = 1 - \frac{L_h^2}{L_s \cdot L_r} \quad (15)$$

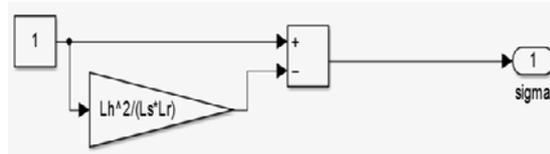


Figura 57: Sigma.

## 4.4. Stator not connected to the grid.

En el caso de que el *stator* no estuviera conectado a la red, se utilizaría este bloque. En la simulación, su uso está deshabilitado pero cambiando el valor de la constante *Stator\_to\_grid* =1, su uso queda habilitado.

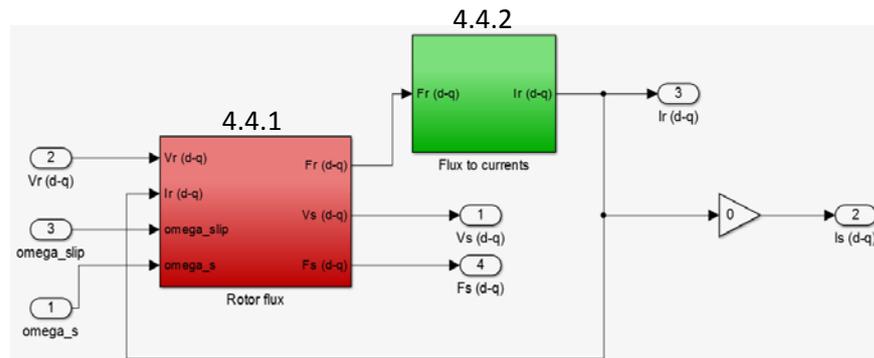


Figura 58: Stator no conectado a la red.

## 4.4.1. Rotor flux.

Cálculo del flujo del rotor.

Utilizando la ecuación (8):

$$\overline{\varphi_r} = \int (\overline{V_r} - \overline{I_r} \cdot R_r + \omega_{slip} \cdot \overline{\varphi_r}) \quad (16)$$

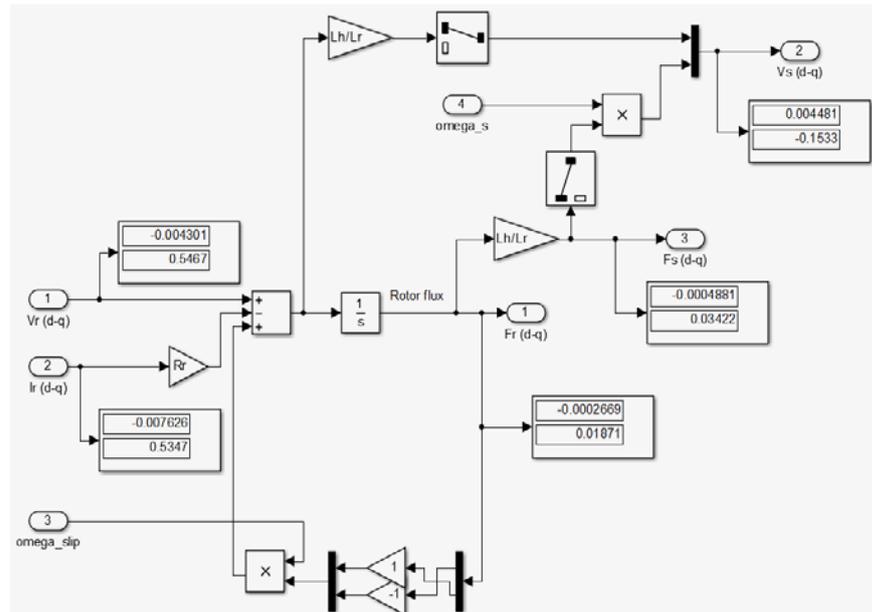


Figura 59: Flujo del rotor.

4.4.2. Flux to currents.

Cálculo de la corriente del rotor  $I_{r(d-q)}$  a partir de  $F_{r(d-q)}$ . La corriente por el rotor  $I_{s(d-q)} = 0$ .

$$I_{r(d-q)} = F_{r(d-q)} \cdot L_r \tag{17}$$

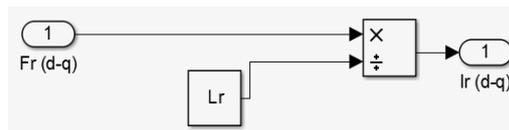


Figura 60: Conversión de flujo a corriente.

4.5. Torque/speed.

Se calcula la velocidad de giro de la máquina a partir del valor del par:

$$T_e - T_{load} = J \cdot \frac{d\omega}{dt} + f \cdot \omega \tag{18}$$

$$\omega = \int \frac{1}{J} (F_s \cdot I_s \cdot p - T_{load} - f \cdot \omega) \tag{19}$$

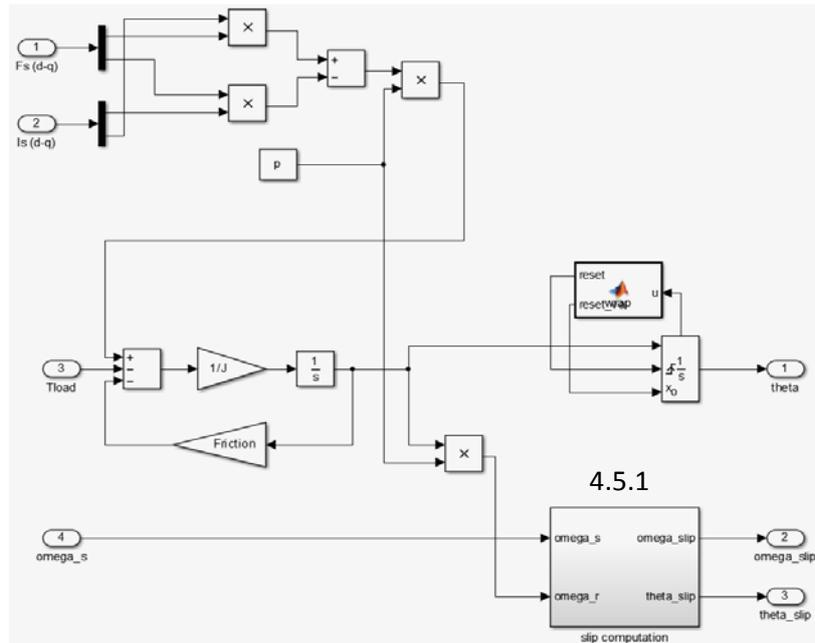


Figura 61: Torque/speed.

% Mantiene el valor de *theta* en  $2\pi$ .

```
function [reset,reset_val] = wrap(u)
% This block supports the Embedded MATLAB subset.
reset = 0;
reset_val = 0;
if (u > 2*pi)
    reset = 1;
    reset_val = u - 2*pi;
end
if (u < 0)
    reset = 1;
    reset_val = u + 2*pi;
end
```

#### 4.5.1. Slip computation.

La diferencia entre la velocidad de variación del flujo en el *stator* y la velocidad de giro en el *rotor*, da como resultado la velocidad de deslizamiento.

$$\omega_{slip} = \omega_s - \omega_r \quad (20)$$

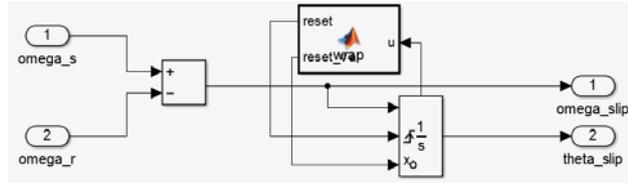


Figura 62: Slip.

% Mantiene el valor de *theta\_slip* en  $2\pi$ .

```
function [reset,reset_val] = wrap(u)
% This block supports the Embedded MATLAB subset.
```

```
reset = 0;
reset_val = 0;
```

```
if (u > 2*pi)
    reset = 1;
    reset_val = u - 2*pi;
end
```

```
if (u < 0)
    reset = 1;
    reset_val = u + 2*pi;
end
```

#### 4.6. dq to abc.

Transformación de coordenadas ( $d - q$ ) a coordenadas ( $a - b - c$ ).  
Para la transformación de la tensión y corriente del *stator*, se utiliza *theta\_s*. Para la transformación de la corriente del *rotor*, es utilizado *theta\_slip*.

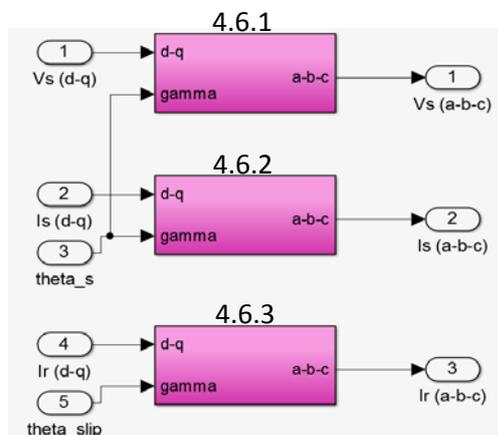
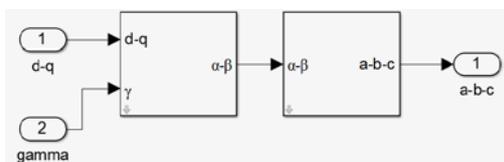
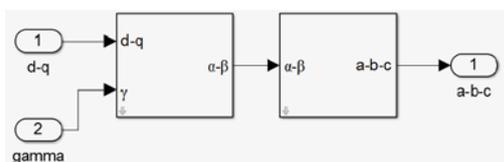


Figura 63: Transformación dq-abc.

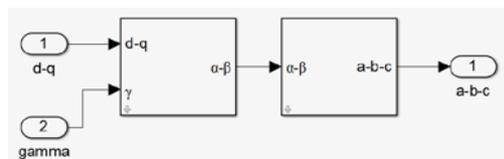
## 4.6.1. Subsystem2

*Park Inversa + Fortescue Inversa.**Figura 64: Park inversa + Fortescue inversa.*

## 4.6.2. Subsystem1.

*Park Inversa + Fortescue Inversa.**Figura 65: Park inversa + Fortescue inversa.*

## 4.6.3. Subsystem3.

*Park Inversa + Fortescue Inversa.**Figura 66: Park inversa + Fortescue inversa.*

5. Preprocessing2.

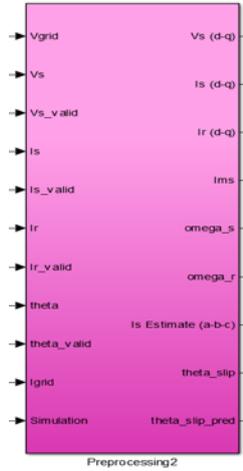


Figura 67: Preprocessing2.

Su función es similar al “Preprocessing” estudiado anteriormente (cambio de coordenadas). También se calcula el valor de la corriente magnetizante y la corriente estimada en el stator, así como  $\omega_r$  y  $\omega_s$ .

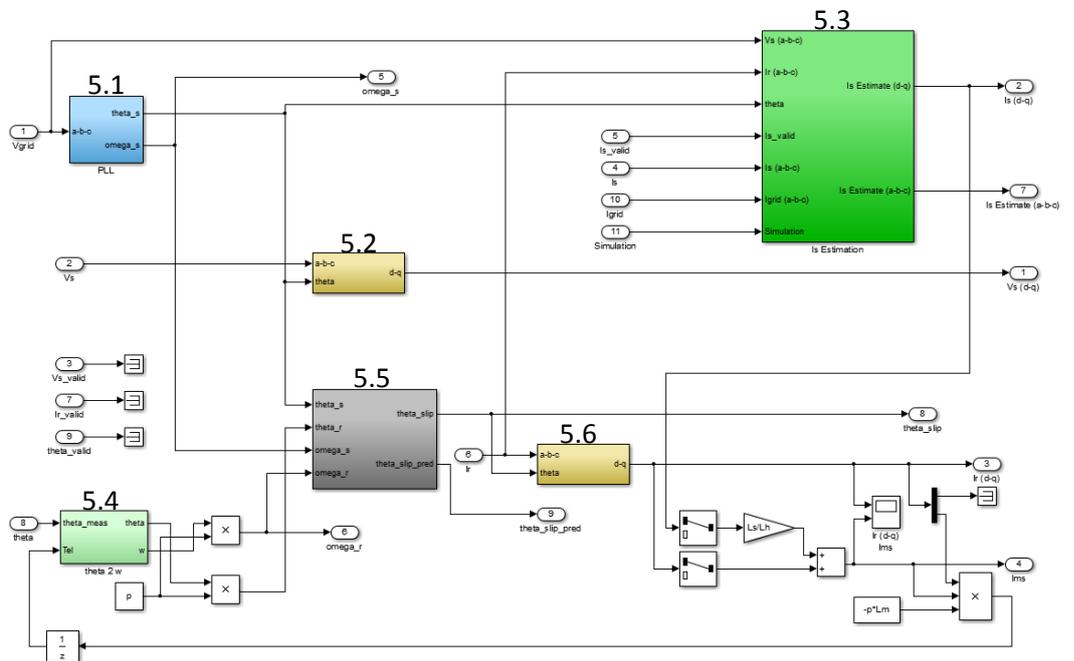


Figura 68: Preprocessing2.

El vector  $\overline{I_{ms}}$  es colinear con el vector flujo del stator ( $\overline{\varphi_s}$ ), (el marco de referencia está alineado al flujo de stator):

$$\overline{\varphi}_s = L_m \cdot \overline{I}_{ms} \quad (21)$$

Sabiendo que:

$$\overline{\varphi}_s = L_s \cdot \overline{I}_s + L_m \cdot \overline{I}_r \quad (22)$$

Igualando (21) y (22):

$$\overline{I}_{ms} = \overline{I}_s \cdot \frac{L_s}{L_m} + \overline{I}_r \quad (23)$$

### 5.1. PLL

El marco de referencia está alineado con el flujo del *stator*, esto es equivalente a la tensión del *stator* alineada con el eje *q*, por lo tanto,  $V_{s,d}SP = 0$ .

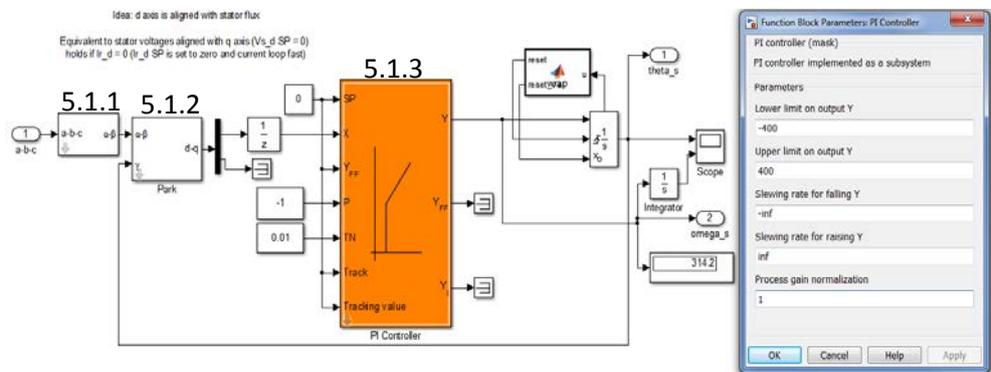


Figura 69: PLL.

% Mantiene el valor de  $\theta_s$  en  $2\pi$ .

```
function [reset,reset_val] = wrap(u)
% This block supports the Embedded MATLAB subset.
% See the help menu for details.
```

```
reset = 0;
reset_val = 0;

if (u > 2*pi)
    reset = 1;
    reset_val = u - 2*pi;
end

if (u < 0)
    reset = 1;
    reset_val = u + 2*pi;
end
```

5.1.1. Transformada de Fortescue.

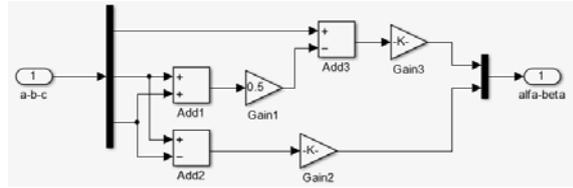


Figura 70: Transformada de Fortescue.

5.1.2. Transformada de Park.

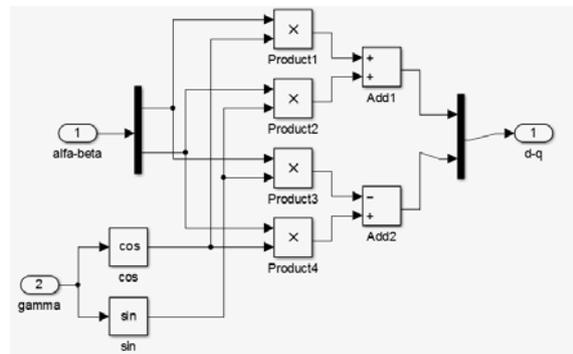


Figura 71: Transformada de Park.

5.1.3. PI Controller.

Un controlador *PI* está implementado para controlar  $V_{grid}$  y por tanto, las potencias activas y reactivas.

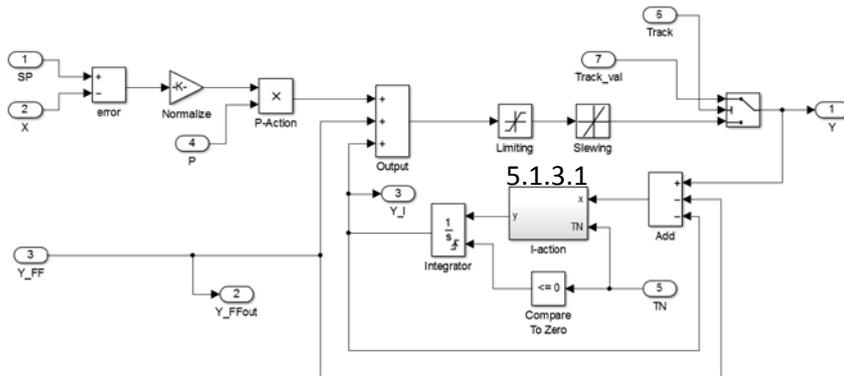


Figura 72: Controlador PI.

5.1.3.1. I-action.

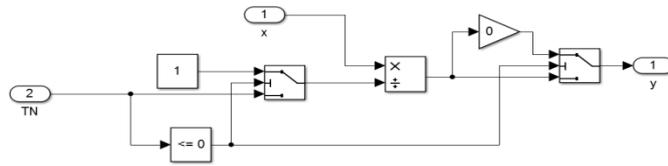


Figura 73: Acción integral.

5.2.Subsystem3.

Transformación de Fortescue + Transformación de Park.

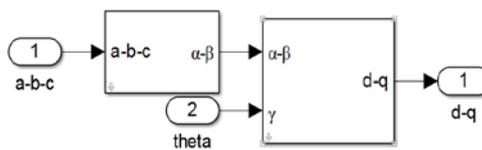


Figura 74: Fortescue transformación + Park transformación.

5.3. Is Estimation.

En la simulación no se usa el valor estimado de la corriente por el stator, sino su valor calculado en el bloque “DFIG model”.

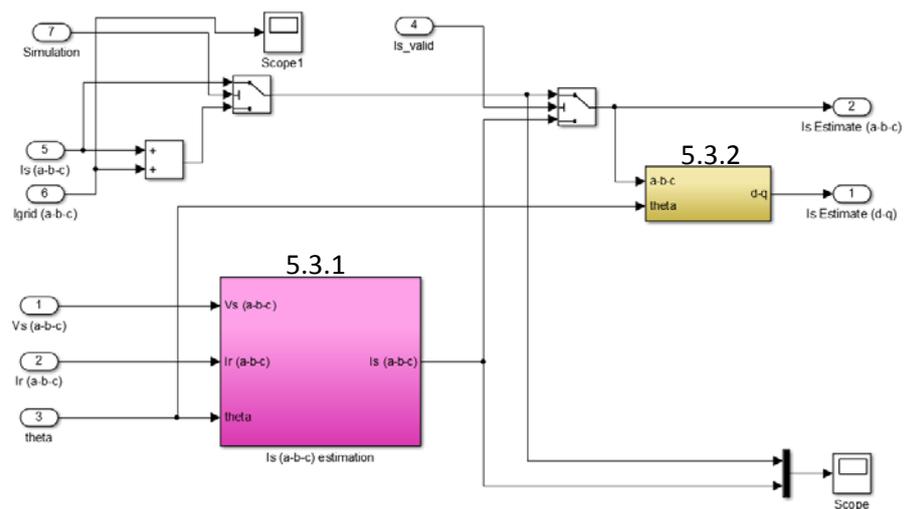


Figura 75: Estimación de Is.

5.3.1. Is (a-b-c) estimation.

Con la tensión del stator y la corriente del rotor, estiman el valor de corriente del stator.

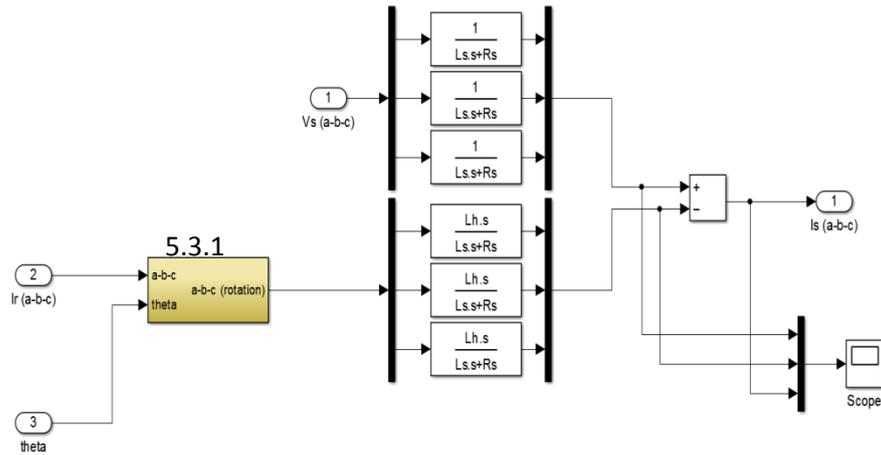


Figura 76: Estimación de Is en coordenadas a-b-c.

5.3.1.1. Subsystem3.

Transformación de Fortescue + Transformación de Park + Fortescue Inversa.

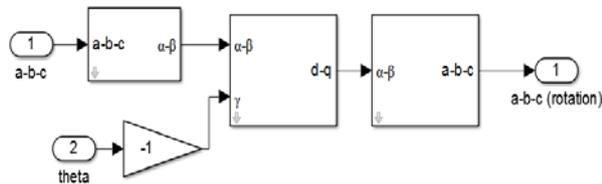


Figura 77: Fortescue transformación + Park transformación + Fortescue inversa.

5.3.2. Subsystem1.

Transformación de Fortescue + Transformación de Park .

La salida es la corriente del stator estimada ( $I_s Estimate_{(d-q)}$ ).

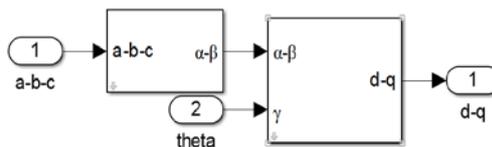


Figura 78: Fortescue transformación + Park transformación.

#### 5.4. Theta 2 w.

Con el valor del par y  $\theta$ , calcula la velocidad del rotor.

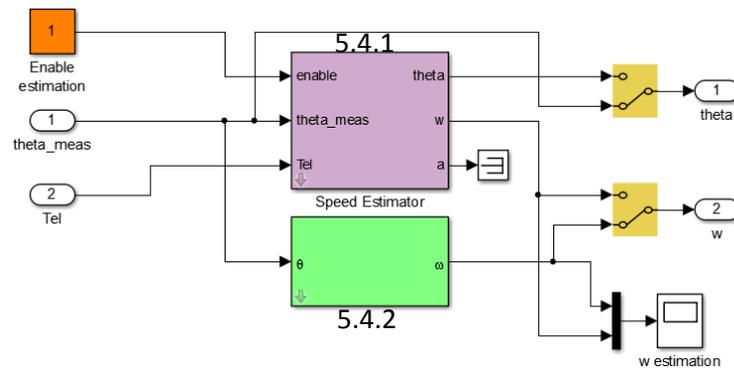


Figura 79: Theta/W.

#### 5.4.1. Speed Estimator.

Esta función está deshabilitada en la simulación.

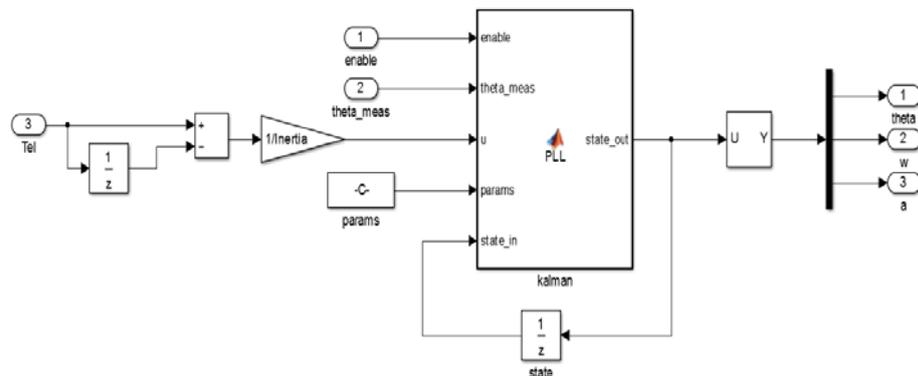


Figura 80: Estimador de velocidad.

```
function state_out = PLL(enable, theta_meas, u, params,
state_in)
```

```
X = state_in(1:3);
P = reshape(state_in(4:12),3,3);
Ts = params(1);
```

```
if enable
```

```
    QN = reshape(params(2:10),3,3);
    RN = params(11);
```

```

% Propagation
% -----
A = [1, Ts, 0; 0, 1, Ts; 0, 0, 1];
B = [0; 0; 1];
X = A*X + B*u;
if (X(1) > pi)
    X(1) = X(1) - 2*pi;
end
if (X(1) < -pi)
    X(1) = X(1) + 2*pi;
end
P = A*P*A' + QN;

% Update
% -----
C = [1, 0, 0];
err = theta_meas - X(1);
if (err > pi)
    err = err - 2*pi;
end
if (err < -pi)
    err = err + 2*pi;
end
H = P*C';
K = H/(C*H + RN);

X = X + K*err;

% Joseph form!
P = P - K*(C*P);
P = P - (P*C')*K';
P = P + (K*RN)*K';

else
    X(1) = params(12);
    X(2) = params(13);
    X(3) = params(14);
    P = reshape(params(15:23),3,3);
end

% Output
% -----
state_out = [X; reshape(P,9,1)];

```

#### 5.4.2. Position to Speed1.

Hace la conversión de posición angular  $\theta$  a velocidad angular  $\omega$ .

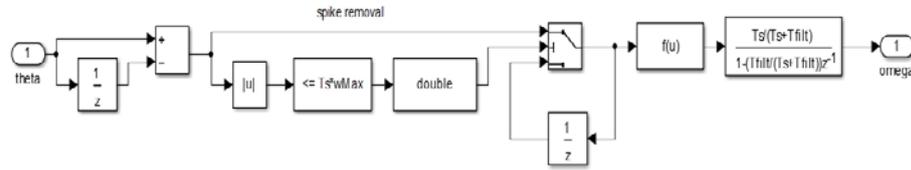


Figura 81: Conversión de posición a velocidad.

#### 5.5. Subsystem.

Cálculo del ángulo de deslizamiento y predicción del mismo.

$$\theta_{slip} = \theta_s - \theta_r \quad (24)$$

$$\theta_{slip\_pred} = \theta_{s\_pred} - \theta_{r\_pred} \quad (25)$$

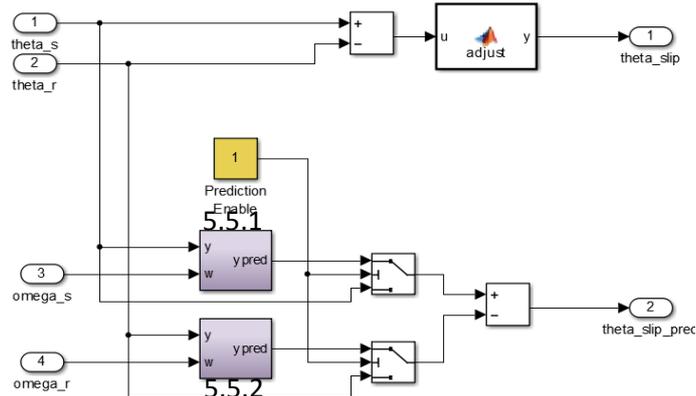


Figura 82: Ángulo de deslizamiento.

% Mantiene el valor de  $\theta_{slip}$  en  $2\pi$ .

```
function y = adjust(u)
```

```
% This block supports the Embedded MATLAB subset.
```

```
y = u;
```

```
if (u < 0)
```

```
    y = u + 2*pi;
```

```
end
```

```
if (u > 2*pi)
```

```
    y = u - 2*pi;
```

```
end
```

5.5.1. Subsystem.

$$\theta_{s\_pred} = \theta_s + (\omega_s \cdot T_{pred}) \tag{26}$$

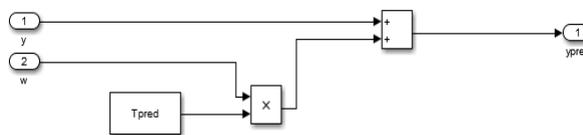


Figura 83: Predicción de  $\theta_s$ .

5.5.2. Subsystem1.

$$\theta_{r\_pred} = \theta_r + (\omega_r \cdot T_{pred}) \tag{27}$$

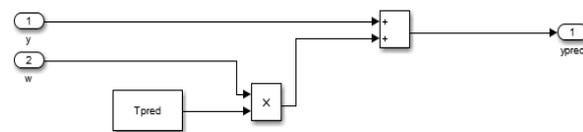


Figura 84: Predicción de  $\theta_r$ .

5.6. Subsystem1

Transformación de Fortescue + Transformación de Park.

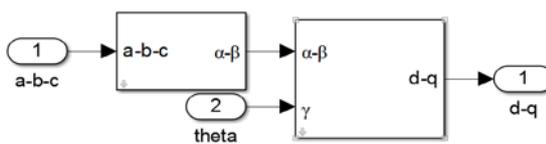


Figura 85: Fortescue transformación + Park transformación.

6. Control2.

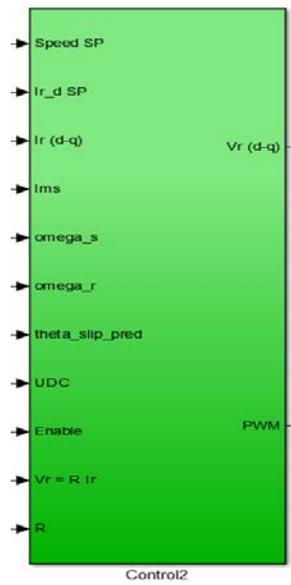


Figura 86: Control2.

De igual manera que en el bloque de “Control” de la primera etapa, este se encarga de generar la señal de control que gobernará el convertidor del lado del rotor.

Con el control del RSC logro manejar el par y por ende la potencia, dado que se tiene el control sobre  $V_r$  (e  $I_r$ ) en las bobinas del rotor [1].

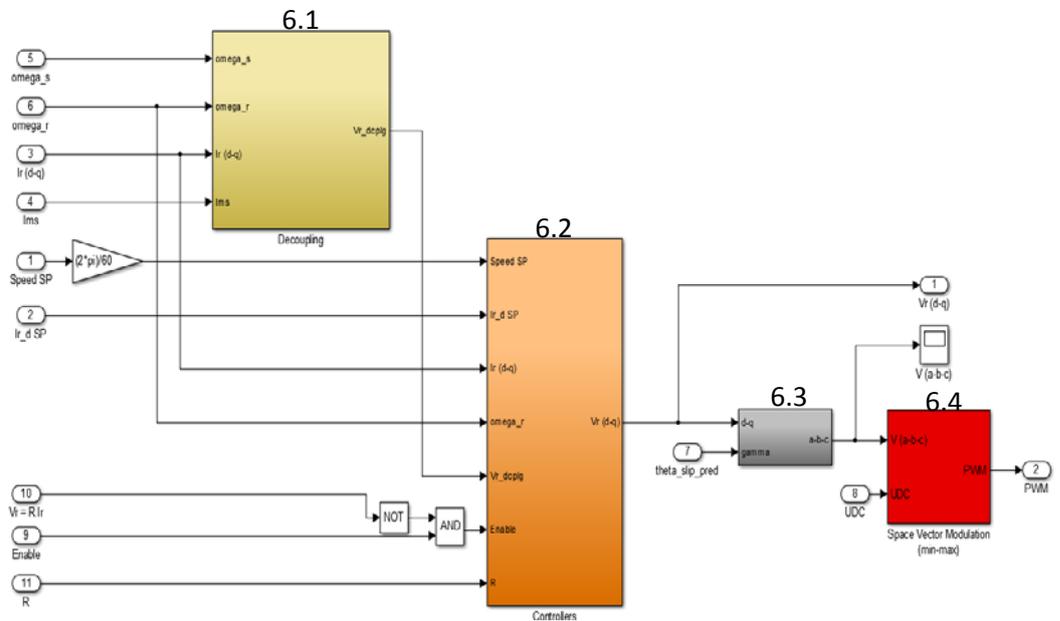


Figura 87: Control2.

### 6.1. Decoupling.

Con el desacoplo de la corriente del rotor (tensión) es suficiente para garantizar la separación entre el control de la potencia activa y reactiva.

La tensión desacoplada queda de la siguiente manera:

$$V_{r_{dcplg}} = \left\{ -L_r \left[ 1 - \frac{L_h^2}{L_s \cdot L_r} \right] \omega_{slip} I_{rq} + I_{ms} L_m \omega_{slip} + I_{rd} L_r \left[ 1 - \frac{L_h^2}{L_s \cdot L_r} \right] \omega_{slip} \right\} \quad (28)$$

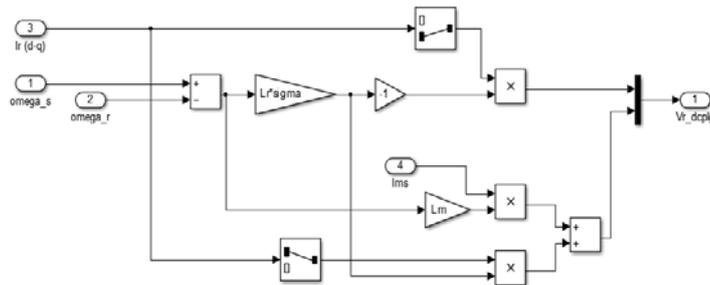


Figura 88: Desacoplamiento.

### 6.2. Controllers.

En los siguientes bloques se controla la velocidad y la corriente para lograr la velocidad de acople con la red y poder conectar el generador a la misma. Para ello se utiliza un controlador *PI*, el cual mediante la realimentación, mantiene las variables controladas en el valor deseado.

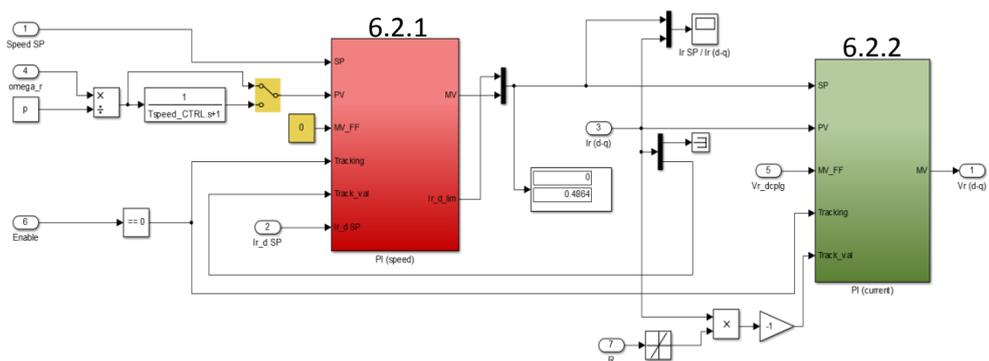


Figura 89: Controladores.

#### 6.2.1. PI (speed).

Con el control de velocidad del rotor, llevándolo a su punto óptimo, se extrae la máxima potencia.

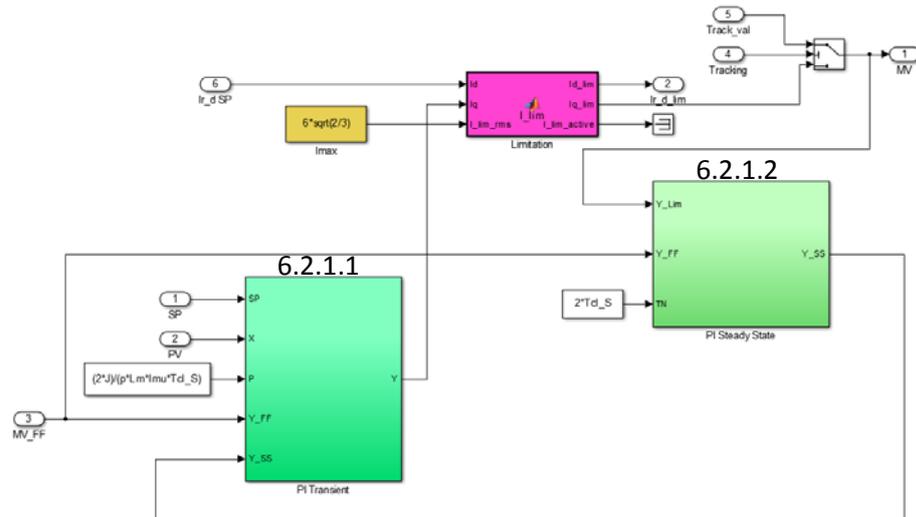


Figura 90: PI velocidad.

```

function [Id_lim,Iq_lim,I_lim_active] =
I_lim(Id,Iq,I_lim_rms)

I_lim = sqrt(2)*I_lim_rms;
I_amp = sqrt(Id^2+Iq^2);
I_lim_active = 0;

if I_amp ~= 0

    if abs(Iq) >= abs(Id)
        theta = atan(Iq/Id);
    else
        theta = acot(Id/Iq);
    end

    if I_amp >= I_lim
        I_amp = I_lim;
        Id_lim = I_amp * sign(Id) * abs(cos(theta));
        Iq_lim = I_amp * sign(Iq) * abs(sin(theta));
        I_lim_active = 1;
    else
        Id_lim = Id;
        Iq_lim = Iq;
    end
end

else

    Id_lim = Id;
    Iq_lim = Iq;
end
end

```

### 6.2.1.1. PI Transient.

La parte proporcional se obtiene multiplicando la constante de proporcionalidad  $P$  por el error entre la referencia y el valor actual de la señal. El objetivo es lograr un error nulo en Estado Transitorio. La acción integral sobre la variable, es sumada a la acción proporcional y de esta manera se obtiene el control *Proporcional-Integral*.

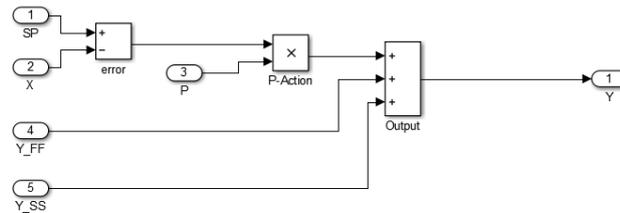


Figura 91: Estado transitorio.

### 6.2.1.2. PI Steady State.

Así como en el estado transitorio se usa el control proporcional, en la zona estacionaria es el control integral quien actúa.

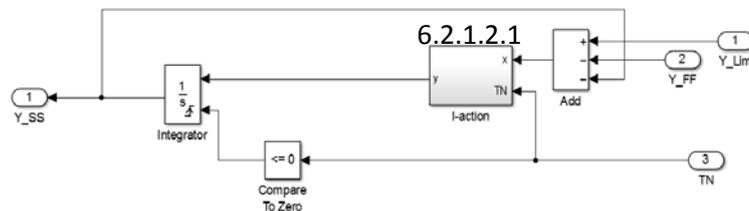


Figura 92: Estado estacionario.

#### 6.2.1.2.1. I-action.

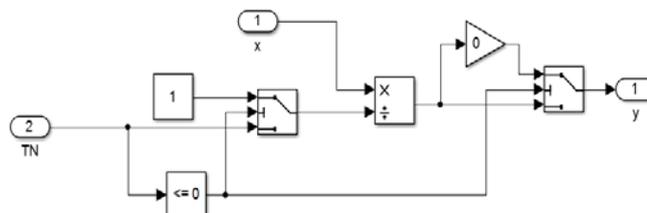


Figura 93: Acción integral.

## 6.2.2. PI (courrent).

Gracias el control PWM, en el RSC, se controla la  $V_r$  y de esta manera queda también controlada la corriente el rotor.

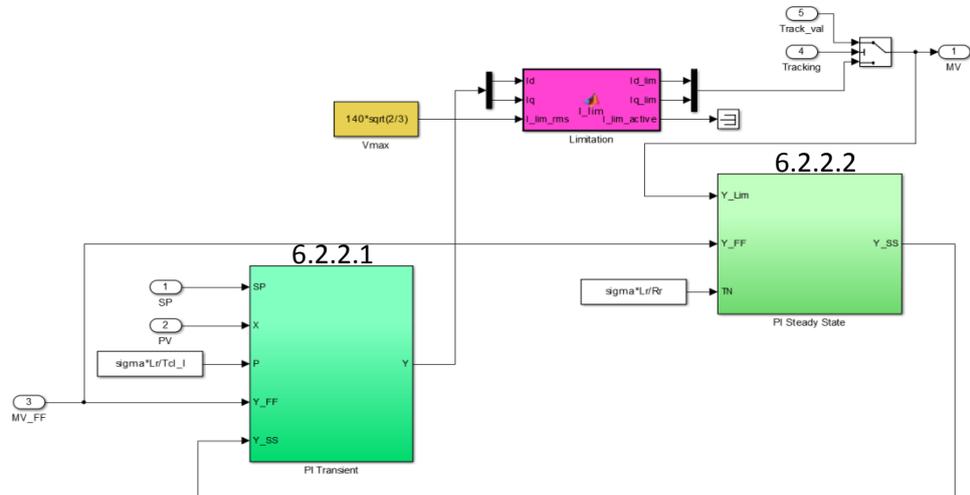


Figura 94: PI corriente.

```
function [Id_lim,Iq_lim,I_lim_active] =
I_lim(Id,Iq,I_lim_rms)
    I_lim = sqrt(2)*I_lim_rms;
    I_amp = sqrt(Id^2+Iq^2);
    I_lim_active = 0;

    if I_amp ~= 0

        if abs(Iq) >= abs(Id)
            theta = atan(Iq/Id);
        else
            theta = acot(Id/Iq);
        end

        if I_amp >= I_lim
            I_amp = I_lim;
            Id_lim = I_amp * sign(Id) * abs(cos(theta));
            Iq_lim = I_amp * sign(Iq) * abs(sin(theta));
            I_lim_active = 1;
        else
            Id_lim = Id;
            Iq_lim = Iq;
        end
    else
        Id_lim = Id;
        Iq_lim = Iq;
    end
end
```

6.2.2.1. PI Transient.

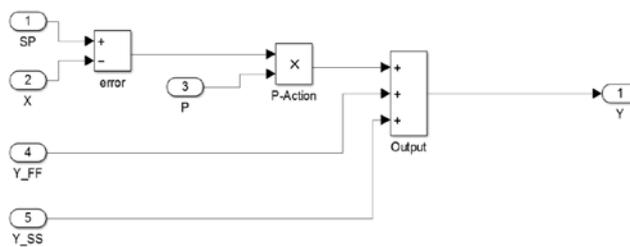


Figura 95: Estado transitorio.

6.2.2.2. PI Steady State.

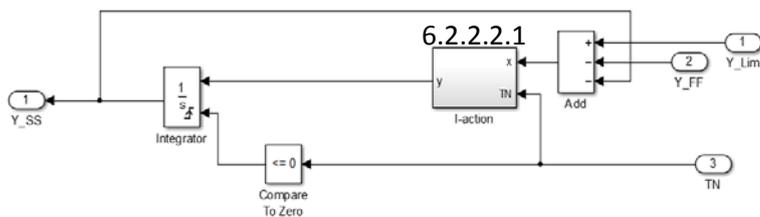


Figura 96: Estado estacionario.

6.2.2.2.1. I-action.

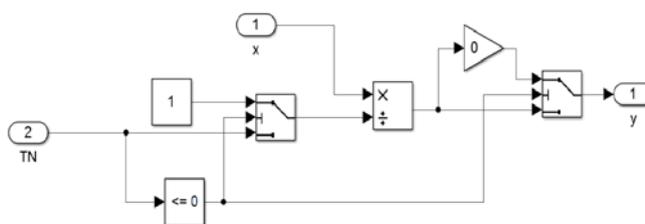
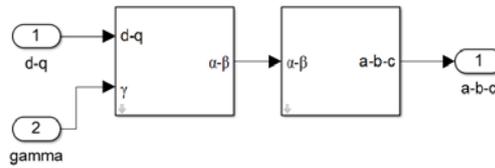


Figura 97: Acción integral

## 6.3. Subsystem1.

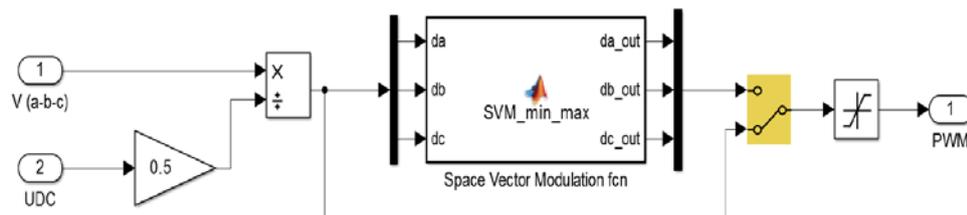
*Park Inversa + Fortescue Inversa.*



*Figura 98: Park inversa + Fortescue inversa.*

## 6.4. Space Vector Modulation (min-max).

Esta función no se encuentra habilitada, por lo tanto, la señal de control se forma con los valores de tensión del rotor,  $V_{(a-b-c)}$  y el "DC\_Link",  $UD$ .



*Figura 99: Space Vector Modulation.*

```
function [da_out,db_out,dc_out] = SVM_min_max(da,db,dc)
```

```
dmin = min([da db dc]);
```

```
dmax = max([da db dc]);
```

```
da_out = da - (dmax + dmin) * 0.5;
```

```
db_out = db - (dmax + dmin) * 0.5;
```

```
dc_out = dc - (dmax + dmin) * 0.5;
```

## 7. Postprocessing.

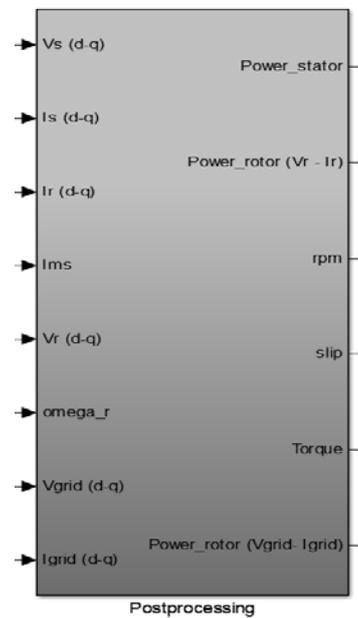


Figura 100: Postprocessing.

Todos los parámetros calculados en este bloque, son utilizados como información del sistema (resultados), pero no en la simulación.

En este bloque son obtenidos los valores de potencia del *rotor* y *stator*, así como el par, la velocidad y el deslizamiento [1].

$$rpm = \omega_r \cdot \frac{60}{2\pi \cdot p} \quad (29)$$

$$slip = \frac{\omega_r - 2\pi f}{2\pi f} \quad (30)$$

$$\omega_s = 2\pi f \quad (31)$$

$$P = V_d \cdot I_d + V_q \cdot I_q \quad (32)$$

$$Q = V_d \cdot I_q + I_d \cdot V_q \quad (33)$$

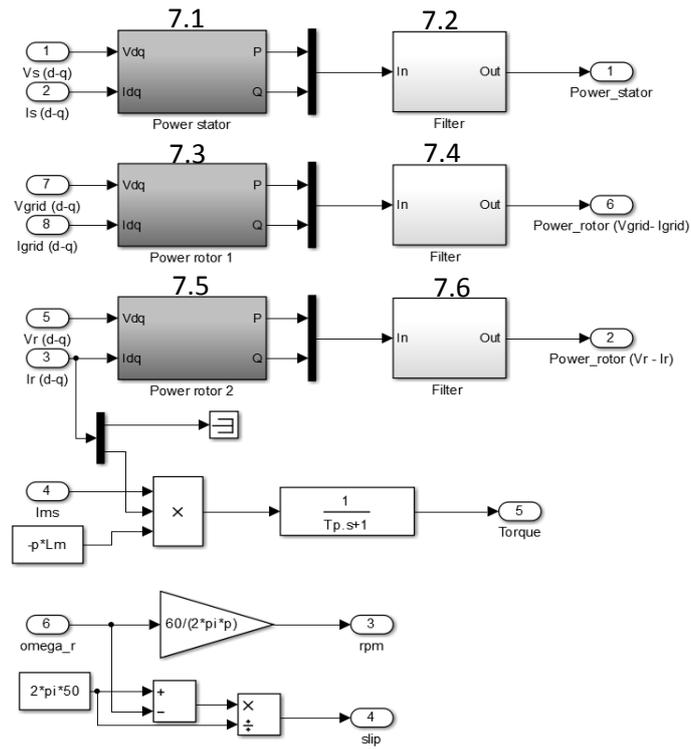


Figura 101: Postprocessing.

7.1. Power stator.

Se aplican las ecuaciones (32) y (33) para el cálculo de la potencia en el stator.

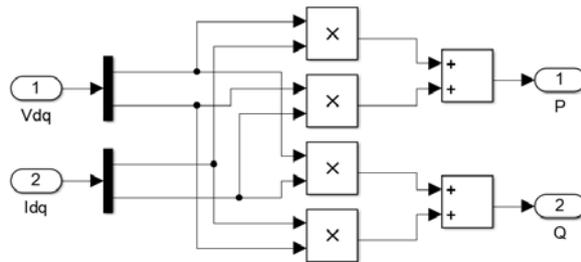


Figura 102: Potencia en el stator.

7.2. Filter.

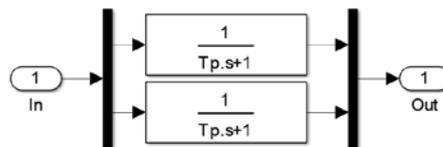


Figura 103: Filtro.

### 7.3. Power rotor 1.

Se aplican las ecuaciones (32) y (33) para el cálculo de potencia en el rotor pero los valores utilizados son los de la red.

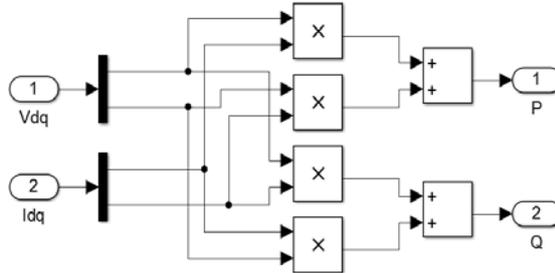


Figura 104: Potencia en el roto ( $V_{grid} - I_{grid}$ ).

### 7.4. Filter.

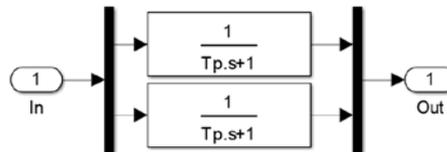


Figura 105: Filtro.

### 7.5. Power rotor 2.

Se aplican las ecuaciones (32) y (33) para el cálculo de potencia en el rotor.

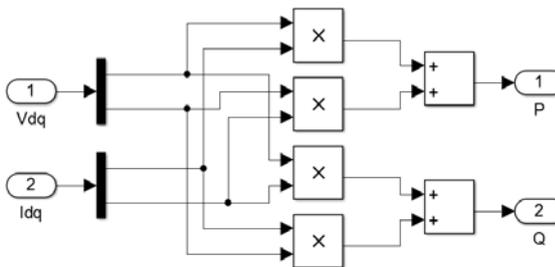


Figura 106: Potencia en el rotor.

### 7.6. Filter.

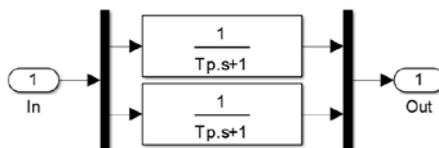


Figura 107: Filtro.

## BLOQUES EXTERIORES:

### SPEED-COUPLE.

Calcula la velocidad de giro del generador y estima si puede o no ser conectada a la red.

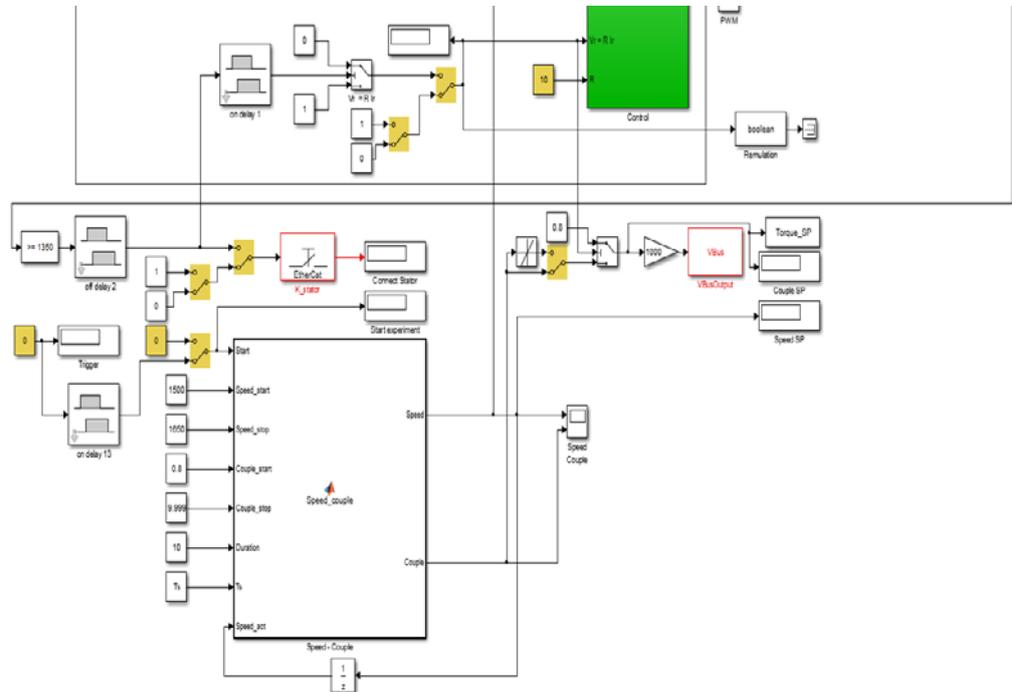


Figura 108: Velocidad/acoplamiento.

```
function [Speed,Couple] = Speed_couple (Start, Speed_start,
Speed_stop, Couple_start,Couple_stop,Duration,Ts,Speed_act)
```

```
if Start
    Speed = Speed_act + (Ts/Duration)*(Speed_stop -
Speed_start);
```

```
else
    Speed = Speed_act - (Ts/Duration)*(Speed_stop -
Speed_start);
end
```

```
if (Speed < Speed_start)
    Speed = Speed_start;
```

```
end
```

```
if (Speed > Speed_stop)
    Speed = Speed_stop;
```

```
end
```

```
Couple      =      Couple_start      +      (Couple_stop      -  
Couple_start)*(((Speed      -      Speed_start)/(Speed_stop      -  
Speed_start))^2);
```

```
if Couple < Couple_start  
    Couple = Couple_start;  
end  
if Couple > Couple_stop  
    Couple = Couple_stop;  
End
```

**OTROS PARÁMETROS.**

Cálculo de las pérdidas en el *stator* y el *rotor*.

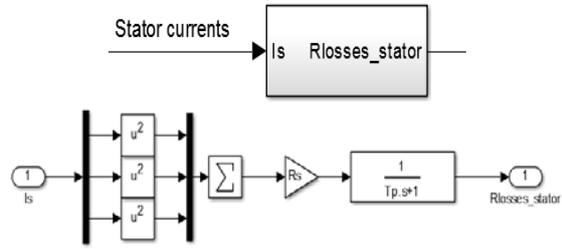


Figura 109: Pérdidas en el stator.

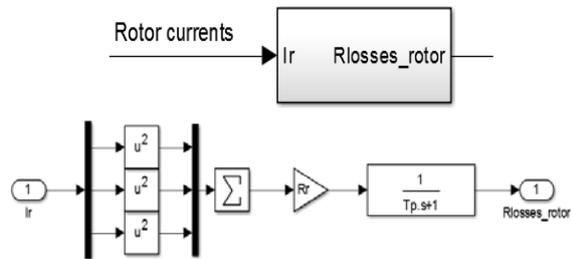


Figura 110: Pérdidas en el rotor.

Inicializa la simulación.

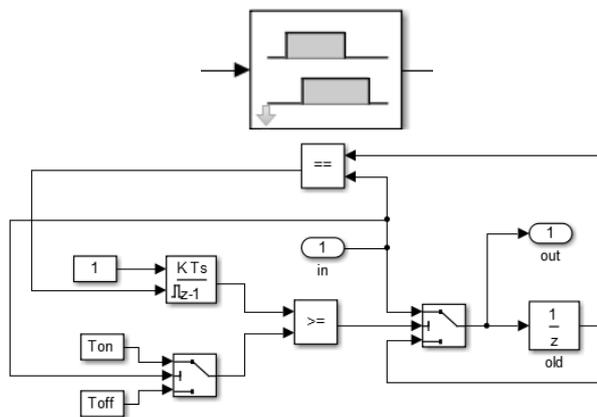


Figura 111: Delay.

Cálculo de la potencia mecánica en el eje.

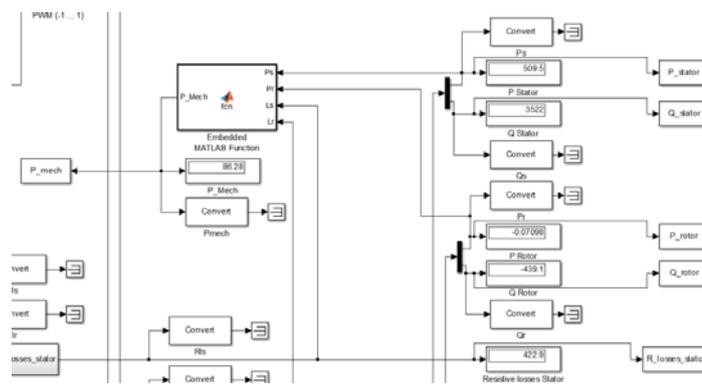


Figura 112: Potencia mecánica.

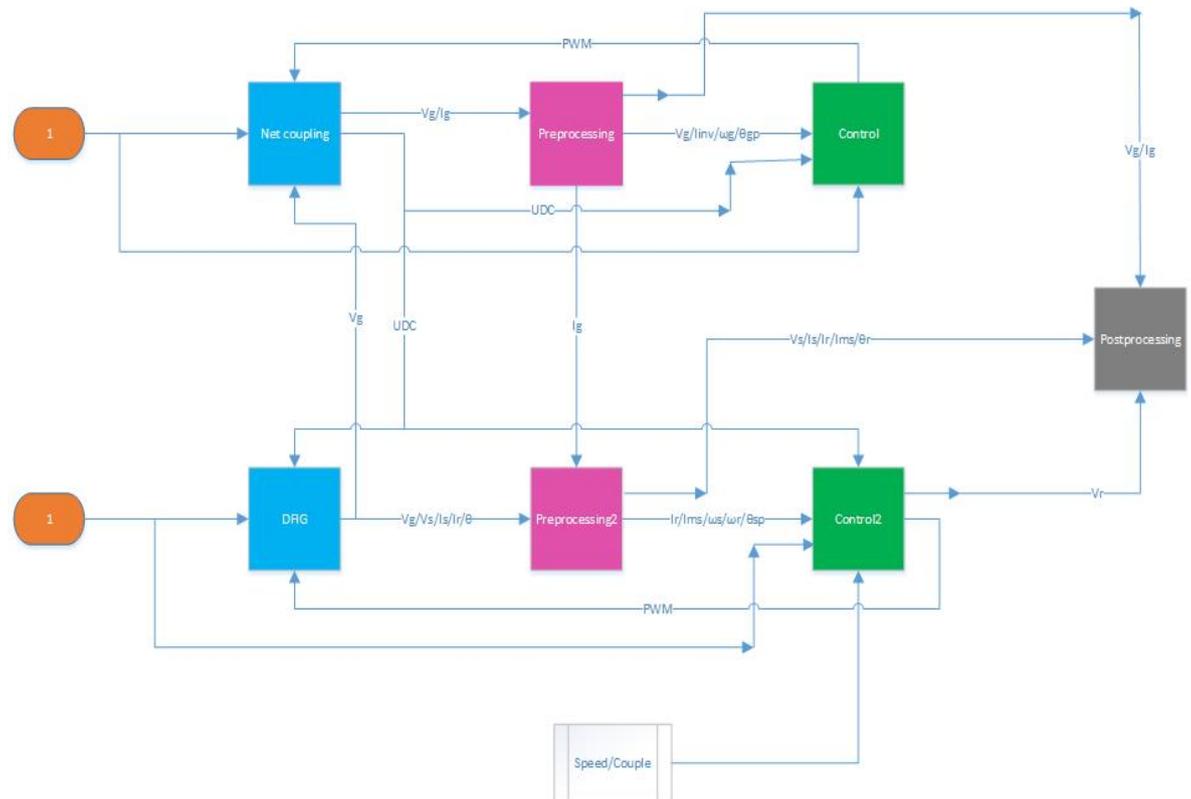
```
function P_Mech = fcn(Ps,Pr,Ls,Lr)
```

```
PPs = Ps - Ls;
```

```
PPr = Pr - Lr;
```

```
P_Mech = PPs + PPr;
```

## DIAGRAMA DE BLOQUES SIMPLIFICADO



1. Net coupling: fija el valor de DC\_Link y simula el GSC.
2. Preprocessing: transformación de coordenadas  $(a - b - c) \rightarrow (d - q)$ .
3. Control: generación de la señal de control SVPWM para el GSC.
4. DFIG: representa el modelo del generador.
5. Preprocessing2: cálculo de  $I_{ms}$ ,  $\omega_r$  y  $\omega_s$ . Transformación de coordenadas  $(a - b - c) \rightarrow (d - q)$ .
6. Control2: generación de la señal de control SVPWM para el RSC.
7. Postprocessing: resultados numéricos a la salida del generador.
8. Speed/Couple: cálculo de la velocidad del generador.



## CONCLUSIONES Y TRABAJO FUTURO

Dado que se cuenta con un aerogenerador en el laboratorio de la Universidad donde se ha realizado el presente estudio, como trabajo futuro se propone probar este sistema de control sobre él.

Los parámetros con los que está simulado este modelo, no coinciden con los del aerogenerador real del laboratorio, una adaptación de los mismos es imprescindible para su posible implementación.

En el apartado de *Anexos* se ha incluido una breve explicación de la *Transformación de Coordenadas*, mostrando la configuración de *Simulink* con la que se obtiene dicha transformación y la base matemática que lo respalda.

En el mismo apartado, se ha desarrollado de manera breve pero concisa una explicación de la técnica de control *SVM* que se adecua a lo encontrado en el sistema. No se ha encontrado en la documentación estudiada una explicación que se corresponda a la situación encontrada en el modelo y por ello ha parecido interesante el desarrollo de la misma.



## BIBLIOGRAFÍA

- [1] Abad, G., López, J., Rodríguez, M.A., Marroyo, L. y Iwanski, G.(2011). *DOUBLY FED INDUCTION MACHINE. MODELING AND CONTROL FOR WIND ENERGY GENERATION*. New Jersey: John Wiley & Sons, Inc.
- [2] Acuña, D. (2013). *Opciones de Control de Potencia Activa y Reactiva en Aerogeneradores con Generador de Inducción Doblemente Alimentado (DFIG)*. [Memoria para optar el Título de Ingeniero Civil Eléctrico]. Universidad de Chile. [Consulta: enero 2015].
- [3] Amaris, O., Alonso, M. y Álvarez, O. (2013). *Reactive Power Management of Power Networks with Wind Generation*. Springer Verlag London.
- [4] Berruti, F. *Control desacoplado de potencia activa y reactiva instantánea en aerogeneradores basados en máquinas de inducción doblemente alimentadas*. IEEE, 8° Encuentro de especialistas en Energía, Potencia, Instrumentación y Medidas. 26 y 27 de Noviembre de 2010. Montevideo – Uruguay.
- [5] Clark, K., Miller, N.W. y Sanchez-Gasca, J.J. (2010). *Modeling of GE Wind Turbine-Generators for Grid Studies*. General Electric International, Inc.
- [6] Fletcher, J. y Yang, J. *Introduction to Doubly-Fed Induction Generator for Wind Power Applications*. University of Strathclyde, Glasgow United Kingdom. [Consulta: enero 2015].
- [7] Hopfensperger, B., Atkinson, D.J. y Lakin, R.A. (2000). *Stator-flux-oriented control of a doubly-fed induction machine with and without position encoder*. IEE Proc.-Electr. Power Appl., Vol. 147, No. 4. <http://cdn.intechweb.org/pdfs/12519.pdf>
- [8] Microsemi. (2014). *Space Vector Modulation IP. User Guide*. [http://www.microsemi.com/document-portal/doc\\_view/133496-space-vector-pulse-width-modulation-hardware-implementation-user-guide](http://www.microsemi.com/document-portal/doc_view/133496-space-vector-pulse-width-modulation-hardware-implementation-user-guide)
- [9] Neacsu, D.O.(2001). *SPACE VECTOR MODULATION –An Introduction*. IECON'01: The 27th Annual Conference of the IEEE Industrial Electronics Society.
- [10] Tapia, A., Tapia, G., Ostolaza, J.X. y Sáenz, J.R. *Modeling and Control of a Wind Turbine Driven Doubly Fed Induction Generator*. IEEE Transactions on Energy Conversion, Vol. 18, No. 2, June 2003. [Consulta: diciembre 2015].



## ANEXOS

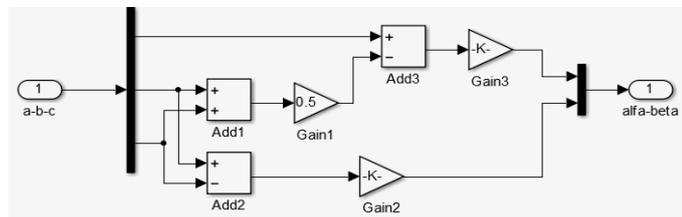
### MARCO DE REFERENCIA. TRANSFORMACIÓN DE COORDENADAS.

Para facilitar el control sobre variables como la corriente o la tensión, se hace la transformación de un sistema trifásico a otro con dos coordenadas. La transformación consiste en la proyección de las variables en los ejes de un marco de referencia. Se sustituye un sistema multifase por otro con un solo vector. La amplitud de la tensión se representa con el módulo del vector y la frecuencia con la velocidad de giro del vector [1].

La transformación de Park, se basa en el teorema de Fortescue, donde tres fasores con la misma frecuencia, pueden descomponerse en tres sistemas (directo, inverso y homopolar). La transformación de Park es el conjunto de dos transformaciones sucesivas  $((a - b - c) \rightarrow (\alpha - \beta) \rightarrow (d - q - 0))$ .

*Transformada de Fortescue (Clark)  $[(a - b - c) \rightarrow (\alpha - \beta)]$ .*

$$\begin{bmatrix} x_\alpha \\ x_\beta \\ x_\gamma \end{bmatrix} = \sqrt{\frac{2}{3}} \cdot \begin{bmatrix} 1 & -1/2 & -1/2 \\ 0 & \sqrt{3}/2 & -\sqrt{3}/2 \\ 1/\sqrt{2} & 1/\sqrt{2} & 1/\sqrt{2} \end{bmatrix} \cdot \begin{bmatrix} x_a \\ x_b \\ x_c \end{bmatrix} \quad (34)$$



*Figura 113: Transformada de Fortescue.*

*Transformada de Park  $[(\alpha - \beta) \rightarrow (d - q)]$ .*

$$\begin{bmatrix} x_d \\ x_q \\ x_0 \end{bmatrix} = \begin{bmatrix} \cos \theta & \sin \theta & 0 \\ -\sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x_\alpha \\ x_\beta \\ x_\delta \end{bmatrix} \quad (35)$$

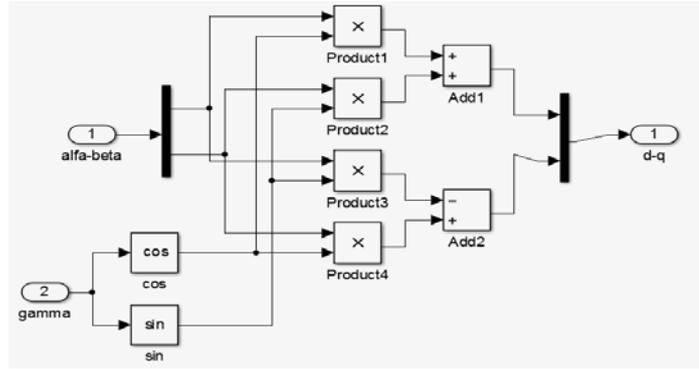


Figura 114: Transformada de Park.

Park Inversa  $[(d - q) \rightarrow (\alpha - \beta)]$ .

$$\begin{bmatrix} x_\alpha \\ x_\beta \\ x_\delta \end{bmatrix} = \begin{bmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x_d \\ x_q \\ x_0 \end{bmatrix} \tag{36}$$

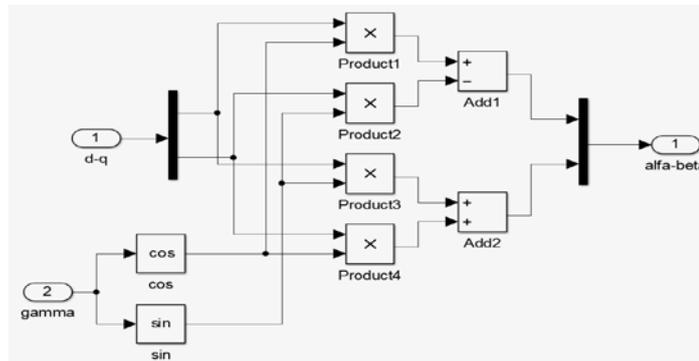


Figura 115: Transformada inversa de Park.

Fortescue inversa (2to3. Clark inverse)  $[(\alpha - \beta) \rightarrow (a - b - c)]$ .

$$\begin{bmatrix} x_a \\ x_b \\ x_c \end{bmatrix} = \sqrt{\frac{2}{3}} \cdot \begin{bmatrix} 1 & 0 & 1 \\ -1/2 & \sqrt{3}/2 & 1 \\ -1/2 & -\sqrt{3}/2 & 1 \end{bmatrix} \cdot \begin{bmatrix} x_\alpha \\ x_\beta \\ x_\gamma \end{bmatrix} \tag{37}$$

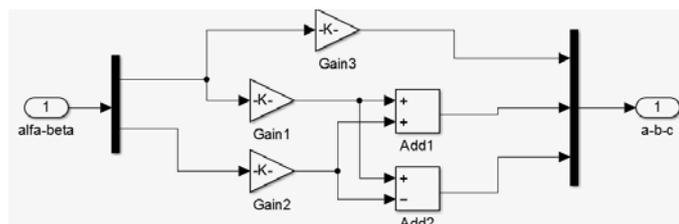


Figura 116: Transformada inversa de Fortescue.

## SVPWM

*Space Vector Modulation (SVM)* es una técnica para el control de la tensión PWM para la conmutación de los interruptores de potencia en los convertidores. Cada posible estado del convertidor se representa por un vector de tensión, y la totalidad de los mismos, forma el diagrama vectorial de tensiones del convertidor.

Mediante el uso de esta técnica vectorial, se provee a la máquina de inducción de un control independiente (desacoplado) de la velocidad y par (funcionamiento similar al del motor de corriente continua).

Esta técnica mejora la señal de salida del inversor, aumentando la ganancia de tensión y disminuyendo las pérdidas. Reduce el número de pequeños pulsos (cuando los valores máximos de las señales moduladora y portadora se encuentran muy próximos) que controlan los transistores, como en el caso del PWM convencional [8] [1].

La técnica SVM se puede implementar con: “*Min\_Max Method*” o “*Direct Injection of Third Harmonic*”.

En la técnica SVPWM se modifica la señal de voltaje de entrada (la fase) para incrementar la ganancia de voltaje del inversor (la tensión de salida) y reducir las pérdidas en el mismo.

En este caso, la técnica SVPWM, se ha implementado con el método “*Min\_Max*”. Este método requiere muy poco esfuerzo computacional.

En la técnica “*Min\_Max Method*”, se resta a cada fase de entrada, el valor en modo común. Esta resta implica la inserción del tercer armónico pero la tensión línea a línea no se ve afectada porque la resta se efectúa a todas las fases.

Cuando se añade el tercer armónico, se reduce el pico de la señal de referencia.

En la *Figura 117*, se muestra gráficamente la diferencia en la forma de onda de la señal convencional PWM y la señal SVPWM. También se ha añadido la señal en modo común.

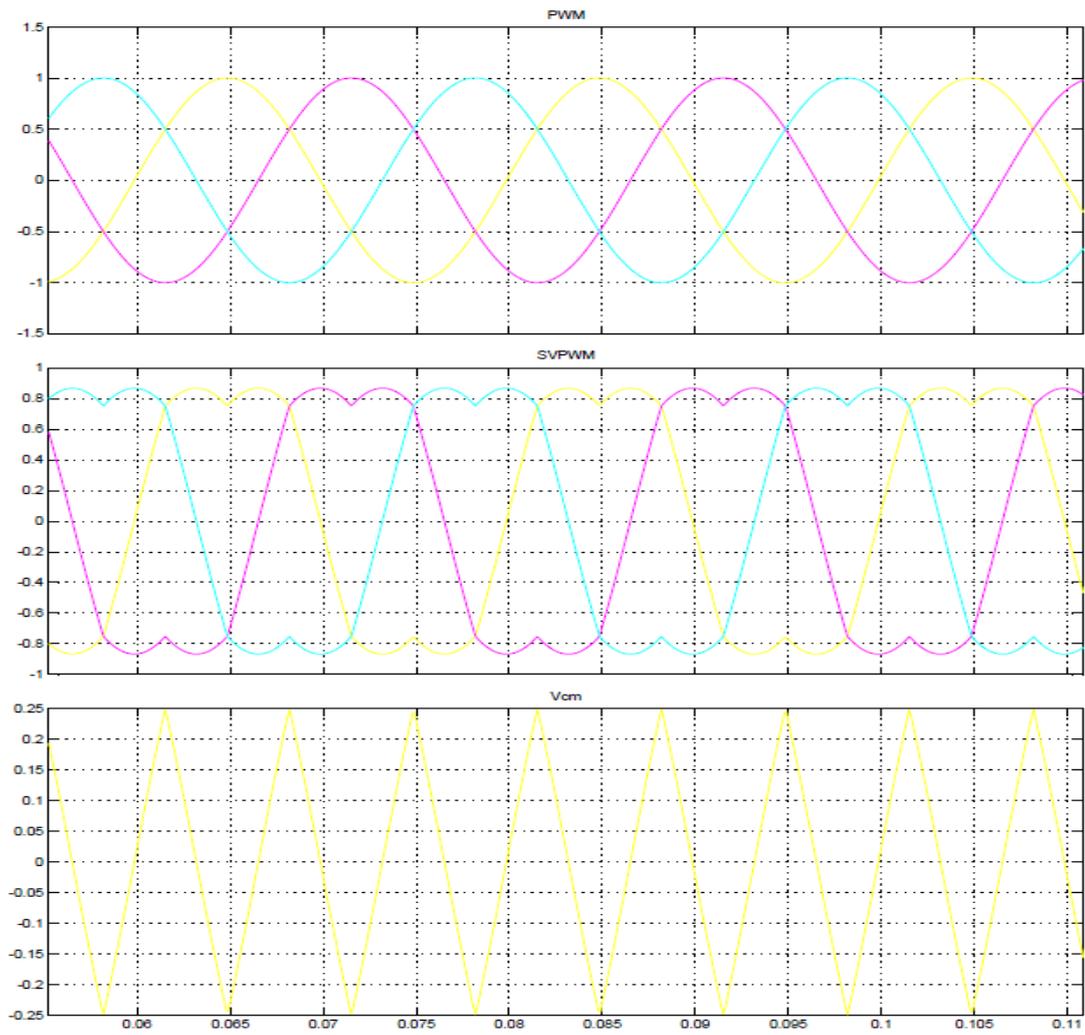


Figure 117: SVPWM.

## GLOSARIO DE TÉRMINOS

Símbolo	Descripción
$a - b - c$	Sistema trifásico
$d - q$	Sistema en el espacio de Park
$UDC$	Tensión en el $DC\_Link$
$C_{cap}$	Condensador (cte)
$C_{bus}$	Condensador del $DC\_Link$ (cte)
$V_{grid}$	Tensión en la red
$I_{grid}$	Corriente en la red
$V_{inv}$	Tensión en el inversor
$I_{inv}$	Corriente en el inversor
$V_r$	Tensión en el <i>rotor</i>
$I_r$	Corriente en el <i>rotor</i>
$R_r$	Resistencia del <i>rotor</i>
$L_r$	Inductancia del <i>rotor</i>
$\omega_r$	Velocidad del <i>rotor</i>
$\varphi_r$	Flujo del <i>rotor</i>
$V_s$	Tensión en el <i>stator</i>
$I_s$	Corriente en el <i>stator</i>
$R_s$	Resistencia del <i>stator</i>
$L_s$	Inductancia del <i>stator</i>
$\omega_s$	Velocidad del flujo en el <i>stator</i>
$\varphi_s$	Flujo del <i>stator</i>
$slip$	Deslizamiento
$\omega_s$	Velocidad de deslizamiento
$rpm$	Velocidad síncrona en r.p.m.
$P$	Potencia activa
$Q$	Potencia reactiva
$P_{inv}$	Potencia activa en el inversor
$I_{ms}$	Corriente magnetizante
$L_h$	Inductancia magnetizante
$I_{inv}^{SP}$	Referencia de corriente
$V_{s,d}^{SP}$	Referencia de tensión
$\omega$	Velocidad de rotación
$\theta_{grid}$	Ángulo de la red
$T$	Par
$f$	Fricción
$J$	Momento de inercia
$PI$	Controlador Proporcional Integral