

UNIVERSIDAD DE EL SALVADOR  
FACULTAD DE INGENIERÍA Y ARQUITECTURA  
ESCUELA DE INGENIERÍA ELÉCTRICA



**DISEÑO DE UN CONTROLADOR DIGITAL, PARA LA  
VELOCIDAD DE UNA TURBINA EÓLICA**

PRESENTADO POR:

**ERNESTO ALONSO CERÓN RODRÍGUEZ**

**ALEJANDRO JAVIER VENTURA RODRÍGUEZ**

PARA OPTAR AL TÍTULO DE:

**INGENIERO ELECTRICISTA**

CIUDAD UNIVERSITARIA, FEBRERO 2018

**UNIVERSIDAD DE EL SALVADOR**

**RECTOR:**

**MSC. ROGER ARMANDO ARIAS ALVARADO**

**SECRETARIO GENERAL:**

**MSc. CRISTÓBAL HERNÁN RIOS BENÍTEZ**

**FACULTAD DE INGENIERÍA Y ARQUITECTURA**

**DECANO:**

**ING. FRANCISCO ANTONIO ALARCÓN SANDOVAL**

**SECRETARIO:**

**ING. JULIO ALBERTO PORTILLO**

**ESCUELA DE INGENIERÍA ELÉCTRICA**

**DIRECTOR:**

**ING. ARMANDO MARTÍNEZ CALDERÓN**

UNIVERSIDAD DE EL SALVADOR  
FACULTAD DE INGENIERÍA Y ARQUITECTURA  
ESCUELA DE INGENIERÍA ELÉCTRICA

Trabajo de Graduación previo a la opción al Grado de:

**INGENIERO ELECTRICISTA**

Título:

**DISEÑO DE UN CONTROLADOR DIGITAL, PARA LA  
VELOCIDAD DE UNA TURBINA EÓLICA**

Presentado por:

**ERNESTO ALONSO CERÓN RODRÍGUEZ**

**ALEJANDRO JAVIER VENTURA RODRÍGUEZ**

Trabajo de Graduación Aprobado por:

Docente Asesor:

**ING. RICARDO ERNESTO CORTEZ**

SAN SALVADOR, FEBRERO 2018

Trabajo de Graduación Aprobado por:

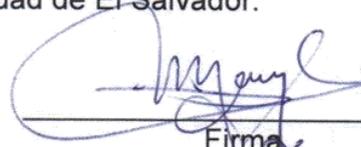
Docente Asesor:

**ING. RICARDO ERNESTO CORTEZ**

## ACTA DE CONSTANCIA DE NOTA Y DEFENSA FINAL

En esta fecha, viernes 2 de febrero de 2018, en la Sala de Lectura de la Escuela de Ingeniería Eléctrica, a las 5:00 p.m. horas, en presencia de las siguientes autoridades de la Escuela de Ingeniería Eléctrica de la Universidad de El Salvador:

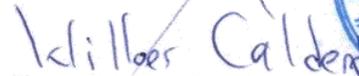
1. Ing. Armando Martínez Calderón  
Director



Firma



2. MSc. José Wilber Calderón Urrutia  
Secretario



Firma

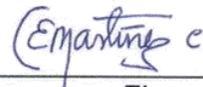
Y, con el Honorable Jurado de Evaluación integrado por las personas siguientes:

- MSC. RICARDO ERNESTO CORTEZ  
(Docente Asesor)



Firma

- DR. CARLOS EUGENIO MARTINEZ CRUZ



Firma

- MSC. HUGO MIGUEL COLATO RODRIGUEZ



Firma

Se efectuó la defensa final reglamentaria del Trabajo de Graduación:

DISEÑO DE UN CONTROLADOR DIGITAL, PARA LA VELOCIDAD DE UNA TURBINA EÓLICA

A cargo de los Bachilleres:

- CERÓN RODRÍGUEZ ERNESTO ALONSO

- VENTURA RODRÍGUEZ ALEJANDRO JAVIER

Habiendo obtenido en el presente Trabajo una nota promedio de la defensa final: 8.8

(Ocho punto ocho) -

## **Agradecimientos.**

A Dios primeramente porque hasta el día de hoy su fidelidad nunca se ha apartado de mí, gracias por darme la inteligencia, sabiduría y fuerzas que esta carrera requiere.

A mi mamá Gladys Ivania Rodríguez Castillo, por ser mi apoyo incondicional, por su amor, por siempre animarme y darme las palabras necesarias para seguir adelante, sobre todo en los momentos que quise tirar la toalla.

A mi papá Mauricio Ernesto Cerón Cruz, por enseñarme a ser una persona ordenada, a nunca conformarme, a dar lo mejor de mí y siempre luchar por superar las metas que me propongo.

A mis hermanos Cesar y Kevin, por su apoyo en las noches de desvelo haciendo trabajos, por motivarme a ser un ejemplo para ellos.

A mi tía Ana Florencia, gracias por ser una inspiración para mi vida y demostrarme que con esfuerzo, dedicación y sacrificio se puede llegar muy lejos, a mi tío Guillermo Antonio gracias por tu ayuda económica, y por siempre animarme a seguir adelante.

A mi novia Ingrid Chávez, gracias por llegar a mi vida en una etapa tan crucial y llenar mi vida de amor, por darme siempre una esperanza en medio de la angustia, por tus cuidados, ayuda y comprensión, por creer en mí y ser mi ayuda idónea.

A mi compañero y amigo Allan Montenegro, por ayudarme en la realización de este trabajo, y por el apoyo a lo largo de la carrera.

A mis compañeros con quienes compartí materias, con los que disfrute en las canchas jugando fútbol, los miembros de ASEIE y a los que tuve oportunidad de impartirles laboratorios, gracias porque de todos aprendí y espero haberles enseñado algo también.

A mis amigos de la iglesia Renato, Yami, Jorge, Astrid, Peraza y a los pastores Roberto, Rudis y Gilberto, por llevarme en sus oraciones y siempre estar pendiente de las pruebas y logros a lo largo de esta carrera.

A maestros que han marcado mi vida, en cada una de sus etapas: Belia Chávez, Alexander Gil, María Bayona y Melvin Orellana. Gracias por sus consejos y palabras de ánimo.

Al Ing. Ernesto Figueroa y a su esposa la Lic. Roxana de Figueroa, por brindarme mi primer trabajo formal referente a la carrera, gracias porque eso represento la adquisición de nuevos conocimientos, así como de recursos para culminar este trabajo.

Al Ing. Julio Rodríguez, porque en poco tiempo que coincidimos en el trabajo, aportó muchos conocimientos a mi vida y gracias por los consejos.

**Ernesto Alonso Cerón Rodríguez**

## **Agradecimientos.**

Agradezco primeramente a Dios por estar conmigo dándome fuerzas en todo este proceso de convertirme en profesional, no solamente en el proceso universitario, sino también en todo el camino de haber elegido tal carrera y por darme la oportunidad de demostrar que todo es posible si se cree en él.

Luego darle gracias a mi familia, mis padres Gerardo Antonio Ventura y Doris Elizabeth Rodríguez, quienes en todo momento me dieron su apoyo incondicional de superar y transformar mis estudios en un éxito, darle gracias a mis abuelas Blanca Lidia Romero y María Concepción Mejía, quienes a pesar de no estar presente en mi culminación de la carrera, vieron y demostraron todo su afecto y deseos de superación hacia a mí al entrar a la universidad, muchas gracias a ellas quienes les debo mucho y que en persona quedaré en deuda siempre, pero que en espíritu les ofreceré mis frutos cosechados.

Por ultimo a mis amigos y compañeros que iniciaron conmigo este camino de ser profesionales, que desde un principio me dieron su aliento y apoyo de seguir adelante juntos, para un día convertirnos en lo que por un principio deseamos iniciar este viaje.

**Alejandro Javier Ventura Rodríguez.**

# Índice

OBJETIVOS DEL TRABAJO:.....	VIII
JUSTIFICACIÓN .....	IX
CAPÍTULO 1 .....	1
INTRODUCCIÓN A LAS TURBINAS EÓLICAS.....	1
1.1    Clasificación de las turbinas eólicas.....	3
1.1.1    Turbinas de eje vertical (VAWT).....	3
1.1.2    Turbinas de eje horizontal (HAWT).....	4
1.2    Partes de una turbina eólica (HAWT).....	5
1.3    Principio de funcionamiento de una turbina eólica.....	7
1.3.1    Fuerza de Sustentación.....	7
1.3.2    Fuerza de Arrastre.....	8
1.4    Sistemas de control de turbinas eólicas.....	8
1.4.1    Control de velocidad.....	8
1.4.2    Control de potencia.....	11
1.5    Potencia del viento.....	12
1.6    Coeficiente de potencia <b>CP</b> y el límite de Betz.....	13
1.7    Modelo matemático de la turbina eólica.....	15
1.7.1    Modelo Aerodinámico.....	15
1.7.2    Modelo Mecánico - Eléctrico.....	17
1.7.3    Sistema de Control.....	21
1.8    Cambio de sistema de referencia ( <b>dqo</b> ).....	22
1.8.1    Representación vectorial de magnitudes físicas.....	22
1.8.2    Transformada de Clarke.....	23
1.8.3    Transformación de Park.....	24
Conclusiones del Capítulo 1.....	26
CAPÍTULO 2 .....	27
MATLAB, SIMULINK Y LA LIBRERÍA SIMSCAPE.....	27
2.1    Matlab.....	29
2.2    Simulink.....	30
2.3    Librería Simscape.....	32
2.3.1    Maquina Asíncrona.....	32
2.3.2    Fuente de tensión trifásica programable.....	40
2.3.3    Medición trifásica V-I.....	45
2.3.4    Puente Universal.....	48

2.3.5	Powergui.....	52
2.3.6	Generador PWM (2 niveles).....	59
	Conclusiones del Capítulo 2. ....	64
	CAPÍTULO 3 .....	65
	IMPLEMENTACIÓN DEL MODELO MATEMÁTICO EN SIMULINK.....	65
3.1	Elementos del Sistema.....	67
3.2	Modelo Aerodinámico.....	67
3.3	Modelo Mecánico - Eléctrico.....	69
3.4	Sistema de Control.....	76
3.4.1	Transformación de señales medidas. ....	76
3.4.2	Control de velocidad. ....	78
	Conclusiones del Capítulo 3. ....	92
	CAPÍTULO 4 .....	93
	GENERALIDADES E IMPLEMENTACIÓN DE LA DSP.....	93
4.1	Procesador Digital de Señales (DSP).....	95
4.2	Las familias de DSP de Texas Instrument. ....	95
4.2.1	SERIE C2000 .....	95
4.2.2	SERIE C5000 .....	96
4.2.3	SERIE C6000 .....	97
4.3	El diagrama de bloques del TMS320F2812.....	100
4.4	La CPU del F2812.....	101
4.5	Unidad Matemática del F2812.....	102
4.6	Placa eZdsp de TMS320F2812 .....	103
4.7	Conectores de la eZdsp F2812 .....	107
4.7.1	P1, Interfaz JTAG.....	107
4.7.2	P2, Interfaz de expansión.....	108
4.7.3	P3, Interfaz de Puerto Paralelo/ JTAG .....	109
4.7.4	P4/P8/P7, Interfaz de I/O .....	109
4.7.5	P5/P9 Interfaz Analógica. ....	110
4.7.6	P6, Conector de Poder.....	111
4.8	Implementacion de la DSP (eZdsp F2812) en el sistema de control.....	111
4.8.1	Medición y acondicionamiento de la señal. ....	112
4.9	Programación de la eZdsp f2812 en Matlab/Simulink.....	114
4.9.1	Code Composer Studio (CCS).....	115
4.9.2	Embedded IDE Link.....	116

4.9.3    Generación de Código a partir del sistema de control implementado.....	122
Conclusiones del Capítulo 4. ....	129
Glosario.....	130
Referencias Bibliográficas.....	131
ANEXOS .....	132
1- Código de Variables utilizadas.....	133
2- Guía de inicio rápido para la tarjeta eZdsp F2812 .....	135
Requerimientos de hardware.....	135
Requerimientos de software .....	136
Configuración básica de Simulink .....	138
3- Código C generado de RTW.....	138

# Índice de Figuras

Figura 1.1 Turbina de Eje Vertical tipo Darreius.....	3
Figura 1.2 Turbina de Eje Horizontal de tres palas.....	4
Figura 1.3 Componentes principales de una Turbina Eólica de Eje Horizontal.....	5
Figura 1.4 Líneas de corriente alrededor del alerón. Fuerza de sustentación $FL$ .....	7
Figura 1.5 Alerón en pérdida.....	8
Figura 1.6 Curva de velocidad de una Turbina Eólica.....	9
Figura 1.7 Conexión del generador para una turbina de velocidad fija.....	10
Figura 1.8 Conexión del generador para una turbina de velocidad variable.....	10
Figura 1.9 Control de potencia por cambio del ángulo de paso.....	12
Figura 1.10 Tubo de Viento a través de una turbina.....	13
Figura 1.11 Grafica del coeficiente de potencia $C_p$ en función de $a$ .....	15
Figura 1.12 $C_p$ vs Lambda ( $\lambda$ ) en función del ángulo de inclinación de la pala.....	16
Figura 1.13 $C_t$ vs Lambda $\lambda$ .....	17
Figura 1.14 Rotor y Estator de la máquina de Inducción.....	18
Figura 1.15 Conexión de la Máquina de Inducción Doblemente Alimentada.....	18
Figura 1.16 Esquema de flujos de potencia en operación subsíncrona.....	20
Figura 1.17 Esquema de flujos de potencia en operación supersíncrona.....	20
Figura 1.18 Diagrama de control de velocidad de la máquina.....	21
Figura 1.19 Vector espacial $S(t)$ en el sistema de ejes $(a, b, c)$ .....	22
Figura 1.20 Descomposición vectorial del vector $S(t)$ .....	23
Figura 1.21 Cambio de base entre los ejes $\alpha\beta\gamma$ a los nuevos ejes $dqo$ .....	24
Figura 1.22 Representación del vector espacial $v$ en la nueva base $dqo$ .....	25
Figura 2.1 Interfaz de Matlab.....	29
Figura 2.2 Librerías de Simulink (Izq.) y Área de trabajo para crear el modelo (Der.).....	30
Figura 2.3 Ejecución pasó a pasó del Real Time Workshop de Matlab.....	31
Figura 2.4 Bloque de la maquina asíncrona.....	32
Figura 2.5 Modelo de la maquina asíncrona referida al eje q (izq.) y al eje d (der.).....	33
Figura 2.6 Ventana de configuración de la maquina asíncrona.....	35
Figura 2.7 Selección de un modelo predefinido de una maquina con rotor Jaula de ardilla.....	36
Figura 2.8 Convención de signos para la operación de la máquina.....	36
Figura 2.9 Ventana de parámetros de la maquina asíncrona.....	37
Figura 2.10 Ventana de opciones avanzadas de la maquina asíncrona.....	38
Figura 2.11 Ventana Load Flow de la maquina asíncrona.....	39
Figura 2.12 Bloque de la fuente de tensión trifásica programable.....	40
Figura 2.13 Ventana de parámetros de la fuente de tensión trifásica.....	41
Figura 2.14 Programación de cambios en el tiempo de los parámetros de la fuente.....	41
Figura 2.15 Configuración de armónicos en la fuente tensión.....	44
Figura 2.16 Ventana Load Flow de la fuente trifásica.....	45
Figura 2.17 Bloque de medición trifásica V-I.....	45
Figura 2.18 Configuración del bloque de medición trifásica.....	46
Figura 2.19 Bloque del puente universal.....	49
Figura 2.20 Orden de conmutación de los dispositivos electrónicos de potencia.....	49
Figura 2.21 Configuración del puente universal.....	50
Figura 2.22 Bloque powergui.....	52
Figura 2.23 Pestaña solucionador del powergui.....	53

Figura 2.24 Pestaña herramientas del powergui .....	54
Figura 2.25 Pestaña preferencias del powergui.....	57
Figura 2.26 Pestaña preferencias del powergui, para tipo de simulación discreto. ....	59
Figura 2.27 Bloque Generador PWM.....	59
Figura 2.28 Muestreo Natural.....	60
Figura 2.29 Muestreo Asimétrico.....	60
Figura 2.30 Muestreo Simétrico.....	60
Figura 3.1 Implementación del sistema en Simulink.....	67
Figura 3.2 Entradas y salida del modelo aerodinámico.....	67
Figura 3.3 Implementación del Modelo Aerodinámico.....	68
Figura 3.4 Función para el cálculo de lambda.....	68
Figura 3.5 Cálculo de la variable $Ct$ .....	69
Figura 3.6 Función para el cálculo del Par de salida.....	69
Figura 3.7 Implementación del Modelo Mecánico-Eléctrico.....	70
Figura 3.8 Parámetros de la fuente de tensión trifásica programable.....	70
Figura 3.9 Tipo de generador de la fuente de tensión trifásica programable.....	70
Figura 3.10 Configuración del bloque de medición del lado del estator.....	71
Figura 3.11 Configuración de la maquina Asíncrona.....	71
Figura 3.12 Parámetros de la Maquina Asíncrona.....	72
Figura 3.13 Opciones Avanzadas de la Maquina Asíncrona.....	72
Figura 3.14 Potencia mecánica de la maquina Asíncrona.....	73
Figura 3.15 Selección de señales en el bus selector.....	73
Figura 3.16 Diagrama para la selección de señales de la maquina asíncrona.....	74
Figura 3.17 Configuración del bloque de medición del lado del rotor.....	74
Figura 3.18 Configuración del puente universal.....	75
Figura 3.19 Configuración de la Fuente de Voltaje de DC.....	75
Figura 3.20 Diagrama de control implementado en Simulink.....	76
Figura 3.21 Cálculo de ángulo de referencia $\theta_r$ .....	76
Figura 3.22 Implementación en Simulink del cálculo de ángulo de referencia $\theta_r$ .....	77
Figura 3.23 Implementación de la transformada abc-- $\alpha\beta$ .....	77
Figura 3.24 Implementación de la transformada $\alpha\beta$ --dco.....	78
Figura 3.25 Entradas del sistema de control.....	78
Figura 3.26 Control de velocidad y obtención de $i_q$ de referencia.....	79
Figura 3.27 Implementación en Simulink del control de velocidad.....	79
Figura 3.28 Controlador PI de velocidad.....	79
Figura 3.29 Configuración del bloque de saturación en el control de velocidad.....	80
Figura 3.30 Configuración del integrador en el controlador PI de velocidad.....	80
Figura 3.31 Controlador PI para la corriente $i_d$ .....	81
Figura 3.32 Controlador PI para la corriente $i_q$ .....	81
Figura 3.33 Configuración del bloque de saturación para los controladores de corriente.....	82
Figura 3.34 Configuración del Integrador para los controladores de corriente.....	82
Figura 3.35 Generación de la señal PWM.....	84
Figura 3.36 Transformación de señales de corriente a señales de voltaje.....	84
Figura 3.37 Transformada de la señal a la referencia abc.....	84
Figura 3.38 Transformación del sistema dco al $\alpha\beta$ .....	85
Figura 3.39 Transformación del sistema $\alpha\beta$ al abc.....	85
Figura 3.40 Generador de señales PWM.....	86
Figura 3.41 Configuración del generador de señales PWM.....	86

Figura 3.42 Configuración del powergui. ....	87
Figura 3.43 Interior del bloque de monitoreo. ....	87
Figura 3.44 Curva de Potencia vs Velocidad del viento para una turbina de 2 MW. ....	88
Figura 3.45 Simulación para $v=6$ m/s y $J=127$ kg.m <sup>2</sup> . ....	88
Figura 3.46 Simulación para $v=8$ m/s y $J=127$ kg.m <sup>2</sup> . ....	89
Figura 3.47 Simulación para $v=10$ m/s y $J=127$ kg.m <sup>2</sup> . ....	89
Figura 3.48 Simulación para $v=11$ m/s y $J=127$ kg.m <sup>2</sup> . ....	89
Figura 3.49 Simulación para $v=12$ m/s y $J=127$ kg.m <sup>2</sup> . ....	90
Figura 3.50 Simulación para $v=13$ m/s y $J=127$ kg.m <sup>2</sup> . ....	90
Figura 3.51 Simulación para $v=6$ m/s y $J=327$ kg.m <sup>2</sup> . ....	91
Figura 3.52 Simulación para $v=8$ m/s y $J=327$ kg.m <sup>2</sup> . ....	91
Figura 3.53 Simulación para $v=10$ m/s y $J=327$ kg.m <sup>2</sup> . ....	91
Figura 4.1. Familias de DSP de Texas Instrument TMS320. ....	95
Figura 4.2. Áreas de aplicación de las DSP por familias. ....	99
Figura 4.3. Evolución de las familias de DSP. ....	99
Figura 4.4 Diagrama de bloques del C281X. ....	100
Figura 4.5 Diagrama de bloques de la CPU del C28XX. ....	101
Figura 4.6 Multiplicador y Unidad Aritmética-Lógica del C28XX. ....	102
Figura 4.7 Diagrama de bloques de la eZdsp F2812. ....	104
Figura 4.8 Distribución física de la placa eZdsp F2812. ....	105
Figura 4.9 Distribución del Mapa de memoria. ....	106
Figura 4.10 Disposición de los conectores en la placa eZdsp F2812. ....	107
Figura 4.11 Distribución de pines para el conector P1. ....	107
Figura 4.12 Distribución de pines para el conector P2. ....	108
Figura 4.13 Distribución de los pines del conector P3. ....	109
Figura 4.14 Distribución de los pines del conector P4/P8/P7. ....	109
Figura 4.15 Distribución de los pines del conector P5/P9. ....	110
Figura 4.16 Diagrama del conector P6. ....	111
Figura 4.17 Implementación de la DSP en el sistema de control. ....	111
Figura 4.18 Anemómetro de cubeta. ....	113
Figura 4.19 Veleta de paletas separadas (Izq.) y Veleta de paleta típica (Der). ....	113
Figura 4.20 Transformadores de corriente para medicion en el rotor y estator. ....	114
Figura 4.21 Interfaz de usuario del CCS 3.1. ....	115
Figura 4.22 Descripción de botones del CCS 3.1. ....	115
Figura 4.23 Bloque ADC en Simulink para la DSP F2812. ....	117
Figura 4.24 Bloque PWM en Simulink para la DSP F2812. ....	118
Figura 4.25 Opciones bloque PWM en Simulink. ....	118
Figura 4.26 Bloques RTDX en Simulink. ....	118
Figura 4.27 Opciones bloque "From RTDX" en Simulink. ....	119
Figura 4.28 Opciones bloque "To RTDX" en Simulink. ....	120
Figura 4.29 Diagrama de flujo del proceso del modelo de Simulink con la tarjeta eZdsp. ....	121
Figura 4.30 Modificación del sistema de control DSP para obtención de código ejecutable. ....	122
Figura 4.31 Entradas al sistema de control Bloques ADC. ....	123
Figura 4.32 Definición de parámetros en el bloque C281x ADC. ....	123
Figura 4.33 Parámetros del convertidor de tipo de datos. ....	124
Figura 4.34 Pestañas Timer y Outputs del bloque PWM. ....	125
Figura 4.35 Pestañas Logic, DeadBand y ADC Control del bloque PWM. ....	125
Figura 4.36 Pestaña para configurar los parámetros de generación de código. ....	126

Figura 4.37 Configuración de la pestaña Solver para nuestra creación de código. ....	126
Figura 4.38 Generación de código en Real Time Workshop.....	127
Figura 4.39 Proyecto creado a partir del código generado por Real Time Workshop en CCS. ....	128
Figura 4.40 Archivos creados y ejecutable por RTW en el directorio de Matlab. ....	128
Figura A2. 1 Contenido del kit de eZdsp F2812. ....	135
Figura A2. 2 Bloques que conforman la librería Embedded Target for TI C2000 DSP. ....	137
Figura A2. 3 Lista de tarjetas instaladas en el PC.....	137

## Índice de Tablas

Tabla 2.1 Subíndices de la maquina asíncrona. ....	33
Tabla 2.2 Parámetros de la maquina asíncrona. ....	34
Tabla 2.3 Variables medidas de la maquina asíncrona.....	40
Tabla 2.4 Etiquetas de las mediciones en el puente universal. ....	52
Tabla 2.5 Vector del pulsos de entrada al puente universal. ....	52
Tabla 4.1.Lista de Conectores de la eZdsp D2812 .....	107
Tabla 4.2. Nombre de los pines del conector P1. ....	108
Tabla 4.3. Nombre de los pines del conector P2. ....	108
Tabla 4.4. Nombre de los pines del conector P4/P8/P7. ....	110
Tabla 4.5 Nombre de los pines del conector P5/P9. ....	110
Tabla A2. 1 Tarjetas soportadas por Simulink.....	135
Tabla A2. 2 Versión del software utilizado. ....	136
Tabla A2. 3 Complementos de Matlab y CCS.....	136

## **OBJETIVOS DEL TRABAJO:**

### **GENERALES:**

- Obtener un control digital realimentado de calidad académica aceptable de acuerdo con especificaciones de operación

### **ESPECIFICOS:**

- Definir el método a usar para diseñar el controlador digital
- Investigar la DSP a usar
- Definir el modelo matemático de la turbina
- Establecer un procedimiento de diseño de un control digital de velocidad realimentado utilizando Matlab, una DSP y una computadora.

# JUSTIFICACIÓN

Contar con un documento base que permita definir el procedimiento práctico de diseño digital del control de turbinas eólicas cuya utilización se generaliza cada vez más en materia de generación de energía eléctrica

**CAPÍTULO 1.**  
**INTRODUCCIÓN A LAS**  
**TURBINAS EÓLICAS.**

## **CAPÍTULO 1.**

### **INTRODUCCIÓN A LAS TURBINAS EÓLICAS.**

En la actualidad, ante la creciente demanda de energía eléctrica en el mundo, y el alto costo de cubrir dicha demanda utilizando combustibles, ha generado la búsqueda y el uso de las energías renovables, siendo una de ellas la energía eólica.

La energía eólica, se obtiene a partir de la energía cinética del viento, la cual hace girar las palas de la turbina que se encuentran conectadas al rotor de un generador, el cual puede ser un generador síncrono o asíncrono, los generadores asíncronos también son llamados de generadores de inducción.

Las maquinas síncronas son las más utilizadas en la generación de energía eléctrica, sin embargo en la producción de energía eólica las maquinas asíncronas han ganado terreno debido a su mayor adaptabilidad a cambios en la carga, lo que le permite producir energía eléctrica para diferentes velocidades del viento en un rango determinado, mientras que los generadores síncronos deben de mantenerse operando en su velocidad síncrona.

En el presente capítulo se presentaran los tipos y clasificación de las turbinas eólicas, las partes que la conforman, el principio de funcionamiento, tipos de control que existen, además se mostraran las ecuaciones matemáticas que rigen el comportamiento de los componentes de la turbina eólica.

Por último se hablara de las transformadas de Clarke y de Park las cuales se aplicaran en el sistema de control.

## 1.1 Clasificación de las turbinas eólicas.

Una turbina eólica también conocida como aerogenerador, es una máquina capaz de convertir la energía cinética del viento, en energía eléctrica. Existen diferentes formas de clasificar a las instalaciones eólicas, una de ellas es de acuerdo a su conexión, pudiendo catalogarse en tres grupos.

a) Conectadas en instalaciones aisladas, b) Conectadas e integradas a la red en una edificación y c) Conectadas en parques eólicos integrados en el sistema eléctrico.

Una de las principales formas de clasificar las turbinas eólicas, es por el eje de giro del rotor, según la orientación de éste las turbinas pueden ser: Turbinas de eje vertical llamadas VAWT por sus siglas en inglés ("Vertical Axis Wind Turbine") y turbinas de eje horizontal llamadas HAWT ("Horizontal Axis Wind Turbine").

### 1.1.1 Turbinas de eje vertical (VAWT)

Están diseñadas con un rotor de eje vertical (perpendicular al suelo), el generador y la caja de engranes se colocan en la parte inferior de la turbina, lo que facilita y disminuye el precio de la construcción y mantenimiento, en la figura 1.1 se muestra una turbina de eje vertical tipo Darreius.



Figura 1.1 Turbina de Eje Vertical tipo Darreius.

Entre las ventajas que presenta este tipo de turbina es que pueden situarse más cerca unas de otras, debido a que no se produce el efecto de frenado del viento (es algo que si ocurre con las turbinas de eje horizontal) por lo que no ocupan mucha superficie. Otra ventaja es que no necesitan un sistema de orientación, dado que las palas del rotor están diseñadas de forma única para captar la energía del viento sin importar la dirección.

Sin embargo, poseen diversas desventajas una de ellas es la baja eficiencia comparadas con las de eje horizontal, así como su complicada estabilidad estructural y las elevadas fluctuaciones de par. También es importante mencionar la ausencia de par de arranque, por lo que se requiere de conexión a la red para poder arrancar utilizando el generador como motor.

### 1.1.2 Turbinas de eje horizontal (HAWT)

Llamadas de esta forma debido a que los ejes principales se encuentran paralelos a suelo, tienen un rotor de eje horizontal, una caja de cambios, un freno y un generador situado en la parte superior de la torre, en la figura 1.2 se muestra una turbina de eje horizontal de tres palas.



Figura 1.2 Turbina de Eje Horizontal de tres palas.

La construcción de este tipo de turbina es más habitual debido a su eficiencia, confiabilidad y capacidad de adaptarse a diferentes potencias. Se caracteriza por su alto rendimiento y estabilidad estructural. Las fluctuaciones de par son moderadas y cuenta con par de arranque.

Entre sus desventajas están la necesidad de un sistema de orientación, aunque con este es posible obtener una mayor producción energética, otra es el alto precio de la construcción y mantenimiento.

Los rotores HAWT se pueden clasificar de acuerdo a:

- La orientación del rotor (barlovento y sotavento)
- Diseño de buje (rígido y flexible)
- Control del rotor (ángulo de inclinación o pérdida de sustentación)
- Tipo de palas (multipala o hélice)
- Alineación con el viento (orientación de curso libre o activo).

Dentro de los rotores tipo hélice el más empleado es el tripala, debido a su mayor rendimiento, estabilidad estructural y aerodinámica, mayor par de arranque y menor emisión de ruido, frente a la monopala y la bipala.

En las turbinas eólicas modernas, el proceso de conversión utiliza la fuerza aerodinámica de sustentación para producir un par neto positivo en un eje giratorio, produciendo potencia mecánica y posteriormente la transformación en energía eléctrica mediante un generador.

## 1.2 Partes de una turbina eólica (HAWT).

Los componentes principales de una turbina de eje horizontal HAWT típica se presentan en la figura 1.3.

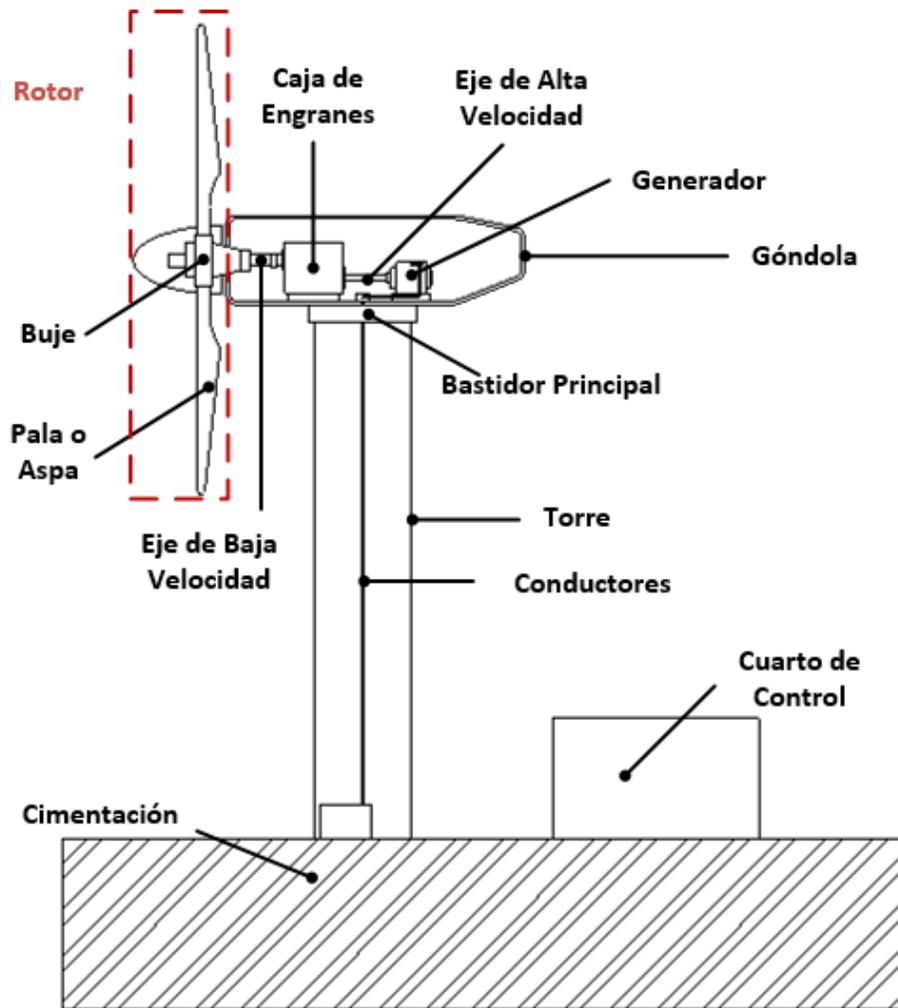


Figura 1.3 Componentes principales de una Turbina Eólica de Eje Horizontal.

A continuación se presenta una breve descripción de los componentes más importantes que conforman la turbina.

- **Rotor**

Es considerado uno de los componentes más importantes de una turbina eólica, está formado por las palas y el buje. En la actualidad la mayoría de turbinas son diseñadas con rotor de tipo barlovento y de 3 palas.

- **Palas**

Están construidas con fibra de vidrio o de carbono y se diseñan para transformar la energía cinética del viento en un momento de torsión en el eje del equipo. Las dimensiones de las palas ronda los 30 metros.

- **Buje**

Es el encargado de unir todas las palas de la turbina, y se conecta al eje de baja velocidad, y transmite el par motriz a la caja de engranes.

- **Tren de Impulsión.**

Está formado por las partes giratorias de la turbina, principalmente por el eje de baja velocidad, la caja de engranes y el eje de alta velocidad.

El objetivo de la caja de engranes es aumentar la velocidad de giro del rotor (decenas de rpm), a una velocidad adecuada para la operación de un generador estándar (cientos de rpm). El eje de alta velocidad cuenta con un freno mecánico, el cual se usa en caso de una falla en el freno aerodinámico o para realizar mantenimiento en la turbina.

- **Generador.**

En la mayoría de turbinas se emplean generadores síncronos o de inducción, ambos operan a velocidad constante o casi constante.

En la mayoría de diseños conectados a la red, se utilizan generadores de inducción, el cual opera en un estrecho rango por encima de su velocidad síncrona (para un generador de 4 polos operando en una red a 60 Hz su velocidad síncrona es de 1800 rpm), la ventaja de este tipo de generador es que son baratos y fáciles de conectar a la red.

- **Góndola y Sistema de Orientación.**

La góndola protege el contenido de las condiciones climáticas, en su interior se encuentran contenidos los elementos del tren de impulsión, el generador, el sistema de orientación y el bastidor principal el cual facilita el montaje y la alineación de los componentes del tren de impulsión.

El objetivo del sistema de orientación es mantener el eje del rotor alineado con la dirección del viento, la pieza principal es un rodamiento que conecta el bastidor principal con la torre, dicho mecanismo se alinea de manera automática mediante un sensor llamado veleta que determina la dirección del viento, el cual se encuentra montado en la góndola.

- **Torre y cimentación.**

La altura de la torre normalmente es de 1 a 1.5 veces el diámetro del rotor (de la circunferencia descrita por las palas y el buje). La mayoría de torres son del tipo auto soportado utilizando tubos de acero o concreto. Una de las características más importantes es la dureza de la torre ya que esta influye en la dinámica de la turbina, debido a la posibilidad de que se acoplen las vibraciones del rotor con la torre.

Las características del sitio influyen en la sección de la torre. La altura y sección de la torre, permiten determinar la profundidad y el tipo de cimentación.

- **Cuarto de control.**

Formado por uno o varios ordenadores, que monitorean de manera continua las condiciones de operación de la turbina, desde ahí también se puede controlar el sistema de orientación. Y en caso de presentarse alguna anomalía, sacar de operación a la turbina, hasta que un encargado autorice la puesta en marcha nuevamente.

### 1.3 Principio de funcionamiento de una turbina eólica.

La obtención de la potencia de una turbina eólica, se consigue convirtiendo la fuerza del viento en un par que actúa sobre las palas del rotor. La cantidad de energía transferida al rotor, depende de la densidad del aire, el área de barrido de las palas y la velocidad del viento.

La energía cinética de un cuerpo en movimiento es proporcional a su masa. Por lo tanto la energía cinética del viento depende de la densidad del aire. A presión atmosférica normal y a 15°C la densidad del aire es 1,255 Kg/m<sup>3</sup> aunque éste valor disminuye ligeramente con el aumento de la humedad.

En referencia al área de barrido de las palas, ésta determina cuanta energía del viento es capaz de capturar la turbina eólica. A mayor diámetro de palas, la superficie es mayor y por lo tanto la energía que absorbe el rotor es mayor.

La velocidad del viento es un parámetro muy importante para la cantidad de energía que la turbina eólica puede transformar en electricidad. A mayor velocidad de viento, la energía que capta la turbina eólica es mayor.

La energía cinética del viento es capturada por la turbina eólica gracias a las palas de rotor. Cuando el viento incide contra las palas, el aire es obligado a fluir por las caras superior e inferior de un perfil inclinado, lo cual genera una diferencia de presiones entre ambas caras, produciendo dos fuerzas (fuerza de sustentación y fuerza de arrastre.) que actúa sobre el perfil, haciendo que las palas giren.

#### 1.3.1 Fuerza de Sustentación.

Las líneas de corriente de flujo, alrededor caras de un alerón se muestran en la figura 1.4. El ángulo de ataque del alerón " $\gamma$ " produce una desviación de aire hacia abajo, por lo cual, encima del alerón, las líneas de corriente se concentran y por debajo se separan. Por la ley de continuidad o de conservación de la masa, al disminuir el área de las líneas de flujo, la velocidad aumenta por encima del alerón y disminuye por debajo del alerón, por el aumento del área. Luego, por el principio de Bernoulli, se produce una diferencia de presión entre ambas caras del alerón, dando lugar a una fuerza ascendente llamada **fuerza de sustentación  $F_L$** .

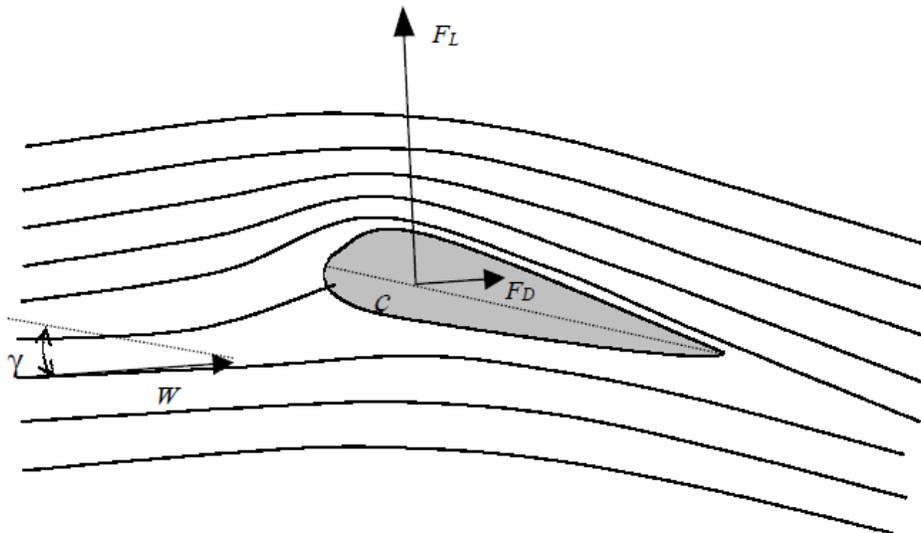


Figura 1.4 Líneas de corriente alrededor del alerón. Fuerza de sustentación  $F_L$ .

### 1.3.2 Fuerza de Arrastre.

Todo cuerpo sumergido en un fluido viscoso experimenta una fuerza en la misma dirección del flujo llamada fuerza de arrastre  $F_D$ . El diseño del alerón evita las turbulencias del flujo de aire, y minimizar la fuerza de arrastre  $F_D$ . Sin embargo, cuando el ángulo de ataque " $\gamma$ " supera cierto valor, el flujo deja de ser laminar, es decir, las líneas de corriente se separan de la cara superior del alerón, como se muestra en la figura 1.5. Cuando el alerón trabaja en estas condiciones, la fuerza de arrastre aumenta considerablemente, y se dice que el alerón entra en pérdida.

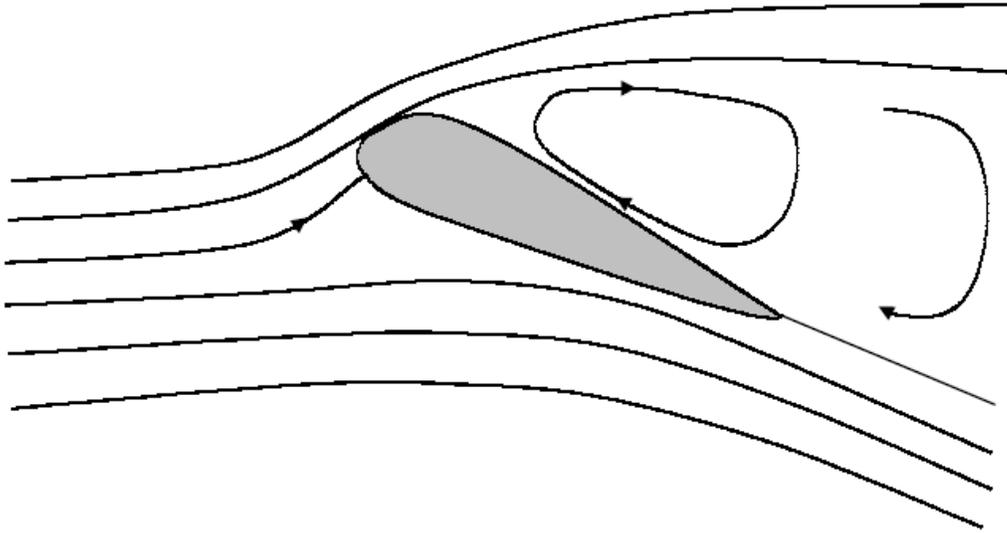


Figura 1.5 Alerón en pérdida.

La fuerza dominante que producirá el par motor, será de arrastre o sustentación, dependiendo del montaje de las palas respecto al viento y al eje de rotación. En las HAWT la fuerza dominante es la de sustentación, esto debido a que permite obtener, mayores potencias por unidad de área de rotor, con un menor peso y costo.

## 1.4 Sistemas de control de turbinas eólicas.

Las turbinas eólicas presentan un comportamiento inusual, con respecto a otros equipos generadores que se conectan a la red eléctrica. Ya que en estas máquinas, el rotor de la turbina suministra una potencia muy variable y un momento de torsión altamente fluctuante. Por ello, es necesario realizar un control de la velocidad de giro de la turbina eólica y de la potencia, por razones tanto de seguridad como de obtención de una potencia óptima.

### 1.4.1 Control de velocidad

Los sistemas de regulación de velocidad se emplean como elementos de control, ya que permiten a la turbina funcionar dentro de los límites de operación evitando vibraciones y cargas indeseadas, accionando el sistema de frenado.

Esto se consigue mediante distintos sensores que captan parámetros como la velocidad de rotación del rotor, la potencia en el alternador, temperaturas en los circuitos hidráulicos, etc., para enviar señales de control al sistema de regulación.

Cuando la velocidad del viento es suficiente como para liberar los frenos, el rotor empieza a girar a la llamada velocidad de conexión o arranque ( $V_a$ ). Esto se produce a bajas velocidades de viento 3-5 m/s aproximadamente y, como es evidente, la generación de energía eléctrica también será baja,

debido a que la energía mecánica que el eje transmite al generador lo es. Si la velocidad del viento aumenta hasta llegar a la velocidad nominal ( $V_n$ ) de giro del rotor la energía cinética que se transmite desde el viento a las palas será la óptima. Una vez que el generador está recibiendo la potencia mecánica óptima, no interesa un aumento de velocidad de viento y es entonces necesaria la actuación de los mecanismos de regulación. Si el viento continúa incrementando, puede peligrar la integridad del rotor, por lo que el mecanismo de regulación deberá pararlo a la velocidad conocida como velocidad de corte o desconexión ( $V_c$ ) aproximadamente 25 m/s. La figura 1.6 muestra una representación de lo anteriormente mencionado.

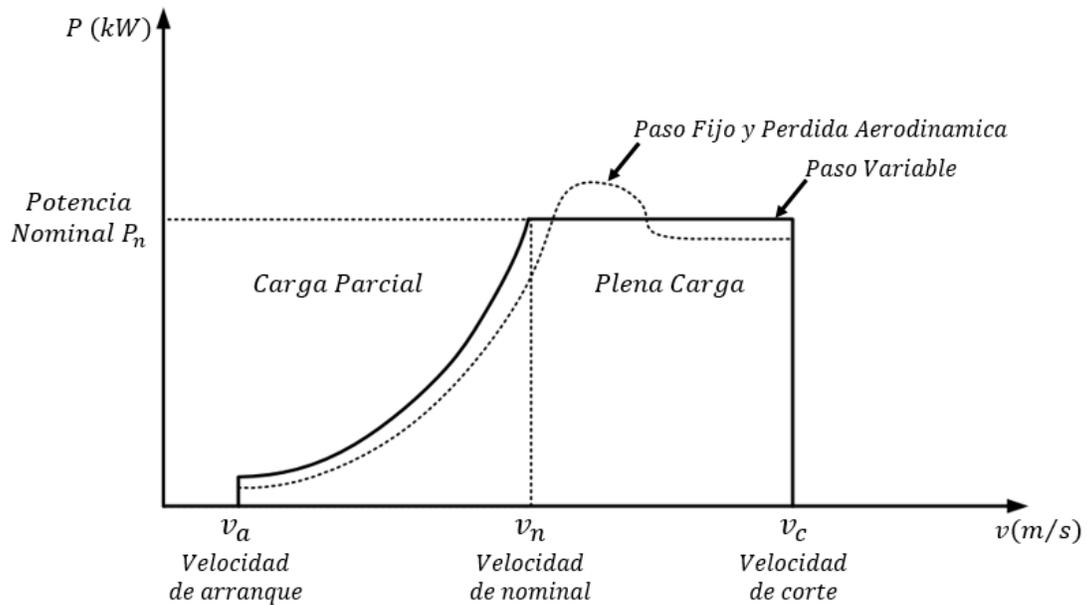


Figura 1.6 Curva de velocidad de una Turbina Eólica.

De acuerdo a la velocidad las turbinas eólicas se pueden clasificar en dos categorías, las de velocidad fija y las de velocidad variable.

- **Velocidad Fija**

El generador de las turbinas de velocidad fija se conecta directamente a la red eléctrica, dicho generador gira a velocidad casi constante, la velocidad de operación de este se determina por la relación de engranes, la frecuencia de la red y el número de pares de polos de la máquina. La máxima eficiencia de conversión puede alcanzarse solo para una velocidad del viento dada y la eficiencia del sistema decae para otras velocidades de viento distintas a esta, en la figura 1.7 se muestra el diagrama de conexión de este tipo de turbinas eólicas.

Este tipo de turbina se protege de los posibles daños causados por altas ráfagas de viento mediante el sistema de regulación por pérdida aerodinámica pasiva.

La principal ventaja de las turbinas eólicas de velocidad fija es su simplicidad y fiabilidad por la menor presencia de mecanismos, así como se evitan las partes móviles del rotor y un complejo sistema de control.

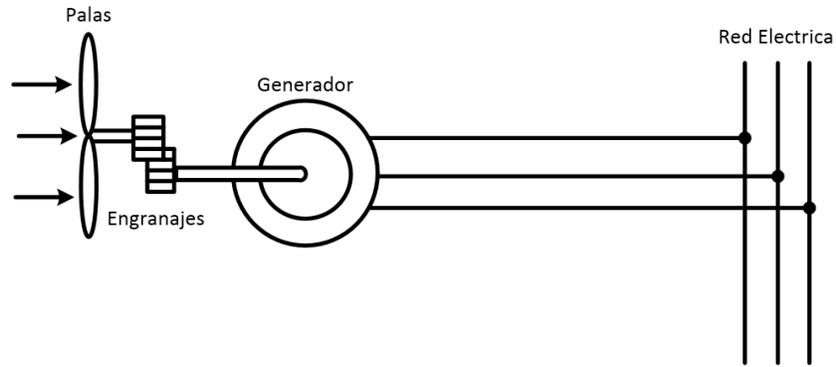


Figura 1.7 Conexión del generador para una turbina de velocidad fija.

La desventaja que tiene este sistema frente al de velocidad variable es que oscilaciones del viento ocasionan sobrecargas mecánicas provocando perturbaciones de potencia y de tensión, entregando así, aunque filtrada, una peor calidad de energía. Estas oscilaciones son perjudiciales para el mismo aerogenerador, ya que los pares pulsantes provocan fatiga en el eje, palas y elementos de transmisión.

- **Velocidad Variable**

Este tipo de turbina puede ajustar continuamente su velocidad rotacional de acuerdo con la velocidad del viento. Con este proceso se consigue el valor máximo de conversión de energía a diferentes valores de la velocidad del viento.

Para conseguir que la velocidad de la turbina sea ajustable normalmente se conecta el generador de la turbinas eólicas a la red de servicio a través de un sistema de conversión de potencia. Como se muestra en la figura 1.8.

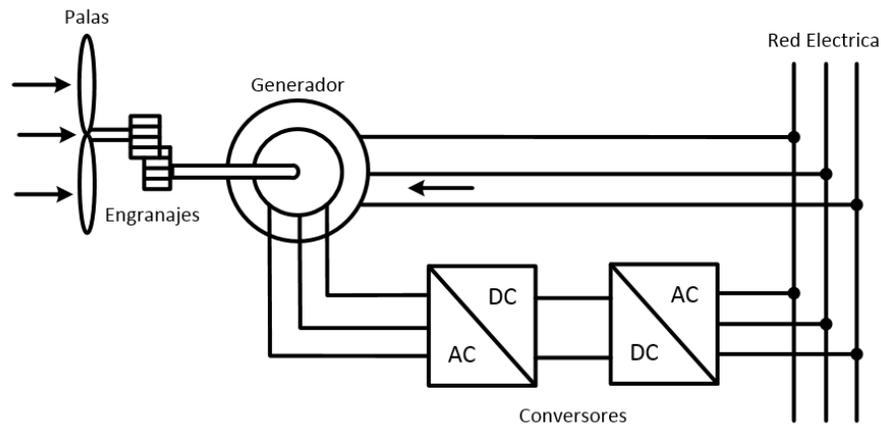


Figura 1.8 Conexión del generador para una turbina de velocidad variable.

Las principales ventajas de estos sistemas son que cuentan ratios de conversión de energía elevados, se mejora la calidad de la potencia y presenta menores tensiones mecánicas su principal problema es el coste.

Por último, las fluctuaciones de potencia y las cargas, se pueden reducir apreciablemente, aplicando flexibilidad al sistema, tanto en las palas como en la velocidad de rotación. Por ello, el control de potencia de paso variable es mucho más útil en sistemas de velocidad variable, que en sistemas de velocidad fija.

#### **1.4.2 Control de potencia.**

Las turbinas eólicas buscan producir energía eléctrica de la forma más económica posible. Por lo que generalmente están diseñadas para rendir al máximo a velocidades alrededor de 12 m/s. Una recomendación importante es no diseñar turbinas eólicas que maximicen su rendimiento a vientos más fuertes, ya que los vientos tan fuertes no son comunes.

En el caso de vientos más fuertes es necesario eliminar parte del exceso de la energía del viento para evitar daños en la turbina eólica. En consecuencia, se debe dotar a la turbina eólica de algún tipo de control de potencia.

Actualmente son dos los tipos de sistemas de control más utilizados en las turbinas eólicas: Pérdida aerodinámica pasiva y Cambio del ángulo de paso.

- **Regulación por Pérdida Aerodinámica Pasiva**

Las turbinas eólicas con regulación por pérdida aerodinámica pasiva, “stall controlled”, tienen las palas del rotor unidas al buje en un ángulo fijo. Sin embargo, el perfil de la pala ha sido aerodinámicamente diseñado para asegurar que en el momento que la velocidad del viento sea demasiado alta, se cree turbulencia en la parte de la pala que no da al viento. Esta pérdida de sustentación evita que la fuerza ascensional de la pala actúe sobre el rotor.

Conforme aumenta la velocidad real del viento en la zona, el ángulo de ataque de la pala del rotor también aumenta, hasta llegar al punto en el que empieza a perder sustentación.

En las turbinas eólicas regulados por pérdida aerodinámica se puede observar que la pala está ligeramente torsionada a lo largo de su eje longitudinal. Esto es así, en parte, para asegurar que la pala pierde la sustentación de forma gradual, en lugar de hacerlo bruscamente, cuando la velocidad del viento alcanza su valor crítico.

Este tipo de regulación representa un problema de diseño aerodinámico muy complejo, y comporta retos en la dinámica estructural de toda la turbina, para evitar las vibraciones provocadas por la pérdida de sustentación.

A pesar de estas desventajas, alrededor de las dos terceras partes de las turbinas eólicas que actualmente se están instalando en todo el mundo son máquinas de regulación por pérdida aerodinámica.

- **Regulación por Cambio Del Angulo De Paso**

En una turbina eólica con regulación por cambio del ángulo de paso, se comprueba varias veces por segundo la potencia generada mediante un controlador electrónico en la turbina. Cuando esta alcanza un valor demasiado elevado, el controlador envía una orden al mecanismo de cambio del ángulo de paso, que instantáneamente hace girar las palas del rotor ligeramente fuera del viento y a la inversa. Las palas del rotor deben ser capaces de girar alrededor de su eje longitudinal, lo que se conoce como variación del ángulo de paso, tal y como se muestra en la figura 1.9.

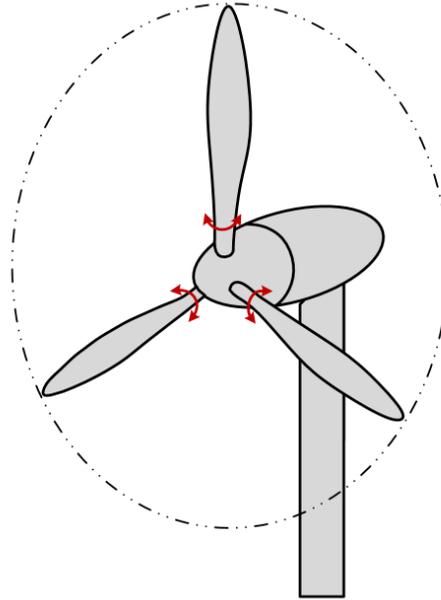


Figura 1.9 Control de potencia por cambio del ángulo de paso.

El diseño de turbinas eólicas controladas mediante la regulación del ángulo de paso de una turbina, requiere una ingeniería muy desarrollada, para asegurar que las palas giren exactamente en ángulo deseado. En este tipo de aerogeneradores, el sistema de control generalmente hará girar las palas unos pocos grados cada vez que el viento cambie, para mantener un ángulo óptimo que proporcione el máximo rendimiento a todas las velocidad de viento.

El mecanismo de cambio del ángulo de paso suele funcionar de forma hidráulica.

### 1.5 Potencia del viento.

La potencia del generador se ve limitada por la potencia del viento, Para calcularla es necesario evaluar la energía cinética ( $E_K$ ) de la masa de aire ( $m$ ), que atraviesa la área cubierta por las palas ( $A$ ), por unidad de tiempo.

$$P_{viento} = \frac{E_K}{t} = \frac{1}{2} m \frac{V^2}{t} \quad \text{Ecuación 1.1}$$

La cantidad de masa de aire que atraviesa el área  $A$  en un tiempo  $t$ , esta dado por

$$m = \rho A d$$

Además sabemos que  $d = Vt$ , donde en nuestro caso  $V$  es la velocidad del viento, por lo tanto.

$$P_{viento} = \frac{1}{2} (\rho A d) \frac{V^2}{t} = \frac{1}{2} \rho A V^2 \frac{d}{t} = \frac{1}{2} \rho A V^3$$

$$P_{viento} = \frac{1}{2} \rho A V^3 \quad \text{Ecuación 1.2}$$

### 1.6 Coeficiente de potencia $C_p$ y el límite de Betz

En la práctica no toda la energía del viento se convierte en energía mecánica de rotación, esta fracción se denomina “coeficiente de potencia” ( $C_p$ ).

$$P_{Turbina} = C_p P_{Viento} = C_p \frac{1}{2} \rho A V^3 \quad \text{Ecuación 1.3}$$

No es posible extraer toda la energía cinética del viento, debido a que el aire no se detiene al atravesar la turbina, por lo tanto  $C_p < 1$ . La velocidad del viento disminuye  $V_1$  a  $V_2$ , por conservación de masa, si la velocidad disminuye, el área del tubo de corriente considerado aumenta como se muestra en la figura 1.10.

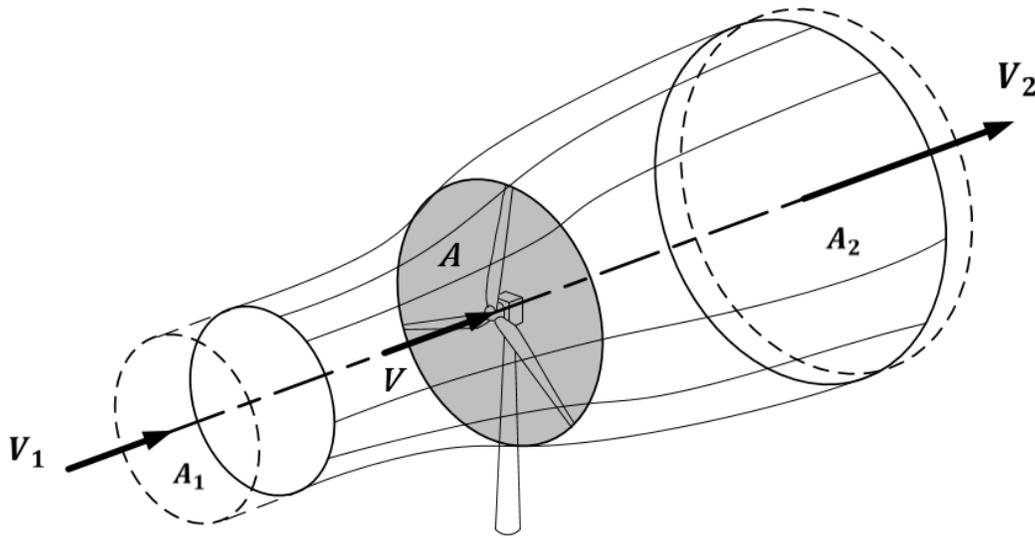


Figura 1.10 Tubo de Viento a través de una turbina.

A barlovento de la hélice, el aire posee una velocidad  $V_1$  (velocidad del viento sin perturbar) en la sección transversal  $A_1$ , mientras que la velocidad  $V_2$  corresponde a la sección transversal  $A_2$  a sotavento de la zona en que se encuentra la hélice.

En el plano que contiene la hélice, es atravesado por la velocidad útil, se supondrá que las aspas forman un disco de área  $A$ , el cual capta parte de la energía del aire en movimiento, que llega a él, es decir  $V_2 < V_1$ . Sin embargo, es obvio que  $V_2$  nunca será cero (dado que no puede extraerse toda la energía cinética del aire).

El caudal másico es constante (por el principio de conservación de masa), es decir:

$$Q_m = \rho Q = \rho A_1 V_1 = \rho A_2 V_2 = \rho A V \quad \text{Ecuación 1.4}$$

Lo anterior explica por qué el tubo se ensancha tras la turbina, como  $V_2 < V_1$ , entonces  $A_2 > A_1$ . Si  $V_1$  es la velocidad inicial del viento,  $V$  es la velocidad del viento en el disco actuador,  $V_2$  es la velocidad final del viento y  $\rho$  es la densidad del aire. Se genera una fuerza  $F$ , por la caída de presión, aplicando la ecuación de Bernoulli de conservación de energía se obtiene la siguiente expresión.

$$F = \frac{1}{2}\rho(V_1^2 - V_2^2)A \quad \text{Ecuación 1.5}$$

Y definimos la velocidad en el disco actuador de la siguiente manera.

$$V = \frac{(V_1 + V_2)}{2} \quad \text{Ecuación 1.6}$$

Si se define el factor adimensional  $a$ , como se muestra en la siguiente ecuación.

$$a = \frac{(V - V_1)}{V_1} \quad \text{Ecuación 1.7}$$

Se obtienen las relaciones de la velocidad del aire de salida y en el disco en función del parámetro  $a$

$$V = (1 - a)V_1 \quad \text{Ecuación 1.8}$$

$$V_2 = (1 - 2a)V_1 \quad \text{Ecuación 1.9}$$

Se pueden obtener las expresiones de la fuerza y de la potencia del viento extraída por el disco como se muestra en las ecuaciones.

$$F = 2a(1 - a)V_1^2\rho A \quad \text{Ecuación 1.10}$$

$$P = 2a(1 - a)^2V_1^3\rho A \quad \text{Ecuación 1.11}$$

Se define el coeficiente de potencia  $C_p$  como la relación entre la potencia extraída por el disco actuador y la potencia del viento sin considerar perturbación.

$$C_p = \frac{2a(1 - a)^2\rho AV_1^3}{\frac{1}{2}\rho AV_1^3}$$

$$C_p = 4a(1 - a)^2 \quad \text{Ecuación 1.12}$$

El máximo valor de  $C_p$  se obtiene con  $a = 1/3$ , este valor se conoce como límite de Bezt, y se puede concluir que la eficiencia de extracción de potencia no puede superar el valor de

$$C_p = \frac{4}{3}\left(1 - \frac{1}{3}\right)^2$$

$$C_p = \frac{4}{3} \times \frac{4}{9}$$

$$C_p = \frac{16}{27}$$

$$C_p = 59.3\% \quad \text{Ecuación 1.13}$$

La grafica de la figura 1.11 muestra la gráfica de  $C_p$  en función de  $a$ .

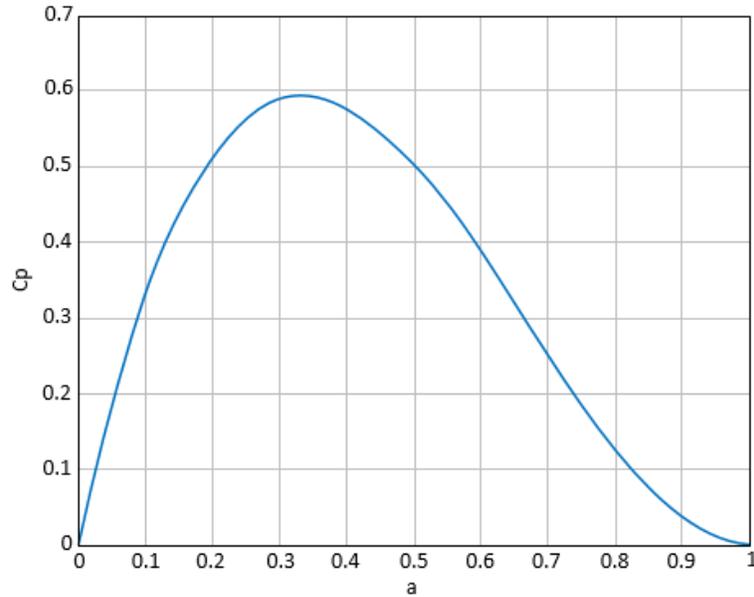


Figura 1.11 Grafica del coeficiente de potencia  $C_p$  en función de  $a$ .

## 1.7 Modelo matemático de la turbina eólica

Para describir el modelo de la turbina lo dividiremos en las siguientes partes.

- Modelo Aerodinámico.
- Modelo Mecánico – Eléctrico.
- Sistema de Control

### 1.7.1 Modelo Aerodinámico

Está formado por los elementos que componen el rotor, esta parte considera la extracción de potencia del rotor, para el cálculo del par o torque mecánico, en función de la velocidad del viento que actúa sobre las palas, por lo que tendrá como entrada la velocidad del viento  $V_w$ , la velocidad angular del rotor  $\omega_r$  y el ángulo de inclinación de las palas  $\beta$ . La salida de este modelo será el par generado en el rotor  $T_r$ .

Como se mencionó en la sección anterior, la potencia extraíble del viento está limitada por el coeficiente de potencia  $C_p$ , el cual puede ser calculado en función de la velocidad en la punta de la pala ( $\lambda$ ) y el ángulo de inclinación ( $\beta$ ) de esta. Una expresión comúnmente utilizada y fácil de adaptar a diferentes turbinas es.

$$C_p = C_1 \left( \frac{C_2}{\lambda_i} - C_3\beta - C_4 \right) e^{-\frac{C_5}{\lambda_i}} \quad \text{Ecuación 1.14}$$

Donde

$$\frac{1}{\lambda_i} = \frac{1}{\lambda + C_6\beta} + \frac{C_7}{\beta^3 + 1} \quad \text{Ecuación 1.15}$$

Las constantes  $C_1, C_2, C_3, C_4, C_5, C_6$  y  $C_7$  representan constantes propias de cada turbina obtenidas por aproximación numérica, en la figura 1.12 podemos ver, las diferentes curvas del coeficiente de potencia  $C_p$  vs lambda  $\lambda$  en función del ángulo de inclinación de la pala  $\beta$ .

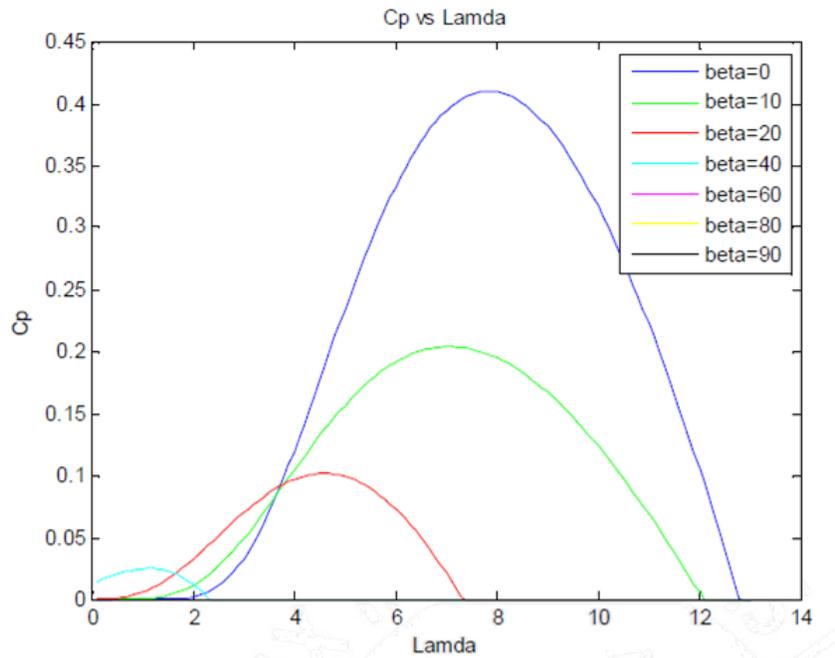


Figura 1.12  $C_p$  vs Lambda ( $\lambda$ ) en función del ángulo de inclinación de la pala.

La velocidad en la punta de la pala denominada  $\lambda$ , está definida por la siguiente expresión.

$$\lambda = \frac{R\omega_r}{V_w} \quad \text{Ecuación 1.16}$$

Donde  $R$  es el radio del rotor en metros ( $m$ ),  $\omega_r$  es la velocidad del rotor en el eje de baja velocidad en ( $rad/s$ ) y  $V_w$  la velocidad del viento en ( $m/s$ ).

La velocidad del viento puede considerarse como la velocidad del viento incidente distribuida en el área barrida por las palas con el objetivo de evaluar el par medio en el eje de baja velocidad. El par generado en el rotor ha sido definido por la siguiente expresión.

$$T_r = \frac{1}{2} \rho \pi R^3 V_w^2 C_t \quad \text{Ecuación 1.17}$$

Donde,  $\rho$  es la densidad del aire,  $R$  es el radio del área cubierta por las palas,  $V_w$  como se mencionó anteriormente es la velocidad del viento y  $C_t$  es el coeficiente de torsión, el cual se calcula al dividir el coeficiente de potencia  $C_p$ , entre la velocidad en la punta de la pala  $\lambda$ , es decir.

$$C_t = \frac{C_p}{\lambda} \quad \text{Ecuación 1.18}$$

En la figura 1.13 podemos ver la gráfica de  $C_t$  vs  $\lambda$ , y la gráfica de potencia de salida vs velocidad del viento.

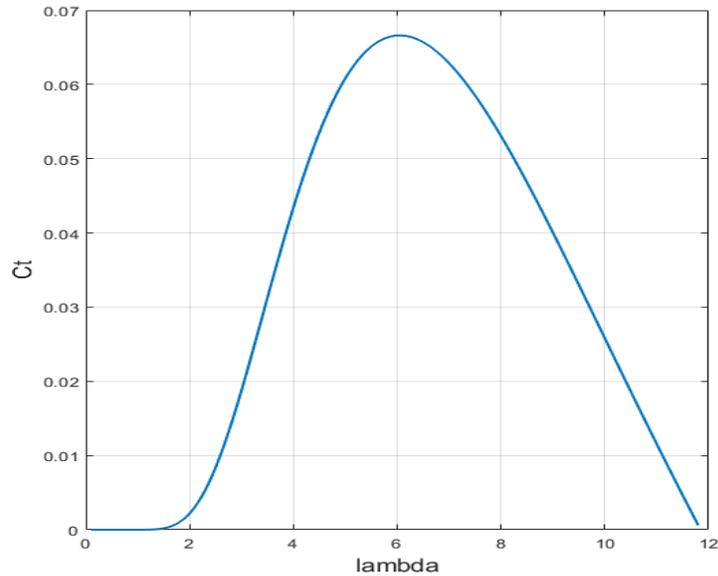


Figura 1.13  $C_t$  vs Lambda  $\lambda$ .

### 1.7.2 Modelo Mecánico - Eléctrico

La parte Mecánica-Eléctrica, está formada por el tren de impulsión (eje de baja velocidad, caja de engranes y eje de alta velocidad) y como generador se considerara un generador de inducción doblemente alimentado, denominado DFIG por sus siglas en inglés (Doubly Fed Induction Generator).

El tren de impulsión como se explicó anteriormente, su función es la de incrementar las revoluciones del eje de baja velocidad (decenas de rpm) a una velocidad adecuada para la operación del generador (cientos de rpm). El incremento de velocidad estará definido por la relación de la caja de engranes que se tenga.

La máquina de inducción o asíncrona, es una máquina de corriente alterna, en la cual la velocidad de rotación del rotor es menor que la del campo magnético del estator, dependiendo de la carga a la que se encuentre conectada. Esta puede funcionar como generador o como motor, esto dependerá del torque en la entrada.

La máquina de inducción posee dos partes principales: estator y rotor. El estator es la parte fija de la máquina en cuyo interior hay ranuras donde se coloca el devanado trifásico que se alimenta con corriente alterna trifásica. La parte giratoria de la máquina se llama rotor y en sus ranuras también se coloca un devanado. El estator y el rotor se arman de chapas estampadas de acero electrotécnico, las partes de la maquina se muestran en la figura 1.14.

El generador de inducción de doble alimentación (DFIG) se ha utilizado durante años para accionamientos de velocidad variable. En este caso, el estator está conectado directamente a la red y el rotor es alimentado por un convertidor bidireccional que también está conectado a la red, como se muestra en la figura 1.15.



- Necesidad de anillos deslizantes.
- Necesidad de engranaje (normalmente engranaje de dos etapas).

El lado del estator se alimenta con una tensión trifásica y frecuencia constante, ya que encuentra conectado directamente a la red. La velocidad y el flujo de potencia activa y reactiva a través del rotor y el estator se controlan ajustando la amplitud, fase y frecuencia de la tensión introducida en el rotor.

- **Control de velocidad del rotor del DFIG**

Al variar la velocidad mecánica de la máquina, se pueden controlar las potencias transferidas desde el rotor hacia el estator, dependiendo de su magnitud respecto a una velocidad conocida como velocidad de sincronismo.

El generador doblemente alimentado, produce un flujo magnético rotatorio originado por el rotor cuya velocidad dependerá de la cantidad de pares de polos de la máquina y la frecuencia de la fuente de alimentación, es decir.

$$\omega_r = \frac{2\pi f_r}{p} \quad \text{Ecuación 1.19}$$

Donde

$\omega_r$  Es la velocidad del flujo magnético inducido por los devanados del rotor.

$f_r$  Es la frecuencia de la corriente que circula en los devanados del rotor en Hz.

$p$  Es la cantidad de pares de polos de la máquina.

De la misma forma la corriente que circula en los devanados del estator produce un flujo magnético cuya velocidad, también conocida como velocidad síncrona, queda expresada por

$$\omega_s = \frac{2\pi f}{p} \quad \text{Ecuación 1.20}$$

Donde  $f$  es la frecuencia de la red, conectada directamente a los enrollados del estator de la máquina de inducción.

De acuerdo con el principio de las máquinas eléctricas rotatorias respecto a las frecuencias angulares de los campos magnéticos entre los devanados del rotor y estator, debe cumplirse la siguiente expresión.

$$\omega_s = \omega_m + \omega_r \quad \text{Ecuación 1.21}$$

Donde  $\omega_m$  corresponde a la velocidad mecánica del rotor, y se calcula por medio de la siguiente expresión.

$$\omega_m = \frac{2\pi(f - f_r)}{p} \quad \text{Ecuación 1.22}$$

A partir de la expresión anterior podemos determinar que es posible controlar la velocidad de giro del rotor, variando la frecuencia de la corriente que alimenta sus devanados. El deslizamiento es una variable importante en las maquinas giratorias y se define por medio de la siguiente expresión.

$$s = \frac{\omega_r}{\omega_s} = \frac{\omega_s - \omega_m}{\omega_s} \quad \text{Ecuación 1.23}$$

- **Operación a velocidad subsíncrona y supersíncrona.**

La velocidad del rotor es una variable fluctuante, esto se debe a que la velocidad del viento no es una magnitud constante, pero al variar la velocidad del rotor se pueden controlar las potencias transferidas al estator y a la red, ya que el comportamiento de estas varía dependiendo de la magnitud de la velocidad del rotor, con respecto a la velocidad de sincronismo.

Se dice que un generador opera a velocidad subsíncrona cuando la velocidad mecánica  $\omega_m$  del rotor es menor a la velocidad de sincronismo  $\omega_s$ . En este caso se tiene un deslizamiento  $s$  positivo y el estator recibe potencia eléctrica  $P_s$  debido a la potencia mecánica del rotor  $P_m$  y a la potencia activa transferida desde la red por medio del rotor  $P_r$ , como se muestra en la figura 1.16.

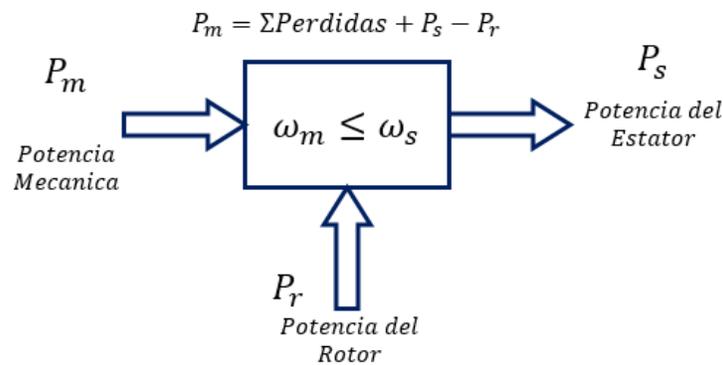


Figura 1.16 Esquema de flujos de potencia en operación subsíncrona.

Cuando el generador opera a velocidad supersíncrona cuando la velocidad mecánica es superior a la velocidad de sincronismo. Por consiguiente el deslizamiento es negativo, y la potencia mecánica transfiere potencia activa tanto al estator como al rotor, como se muestra en la figura 1.17.

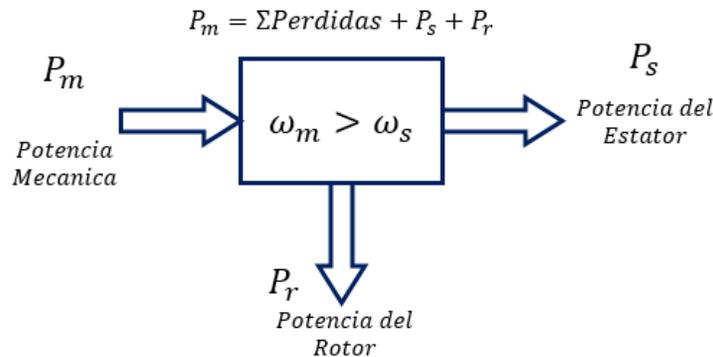


Figura 1.17 Esquema de flujos de potencia en operación supersíncrona.

### 1.7.3 Sistema de Control

Aunque existe una variedad de métodos de control de un generador de inducción doblemente alimentado, en el presente trabajo se ha desarrollado un control vectorial o control de campo orientado.

Este control vectorial permite manipular el flujo y el torque eléctrico de la máquina de manera independiente, referenciando el sistema trifásico alterno a un nuevo sistema de coordenadas orientado a uno de los flujos magnéticos internos del generador. Con ello se obtienen tensiones y corrientes que pueden ser tratadas como vectores rotatorios.

El control vectorial directo se implementa a partir de la medición directa del módulo y la posición del vector espacial de flujo, mientras que el método indirecto estima estos parámetros a partir del desarrollo de un modelo del generador. En el desarrollo de cada uno de los esquemas de control se emplea la transformación de Park o dq para representar las ecuaciones que describen el comportamiento del generador. Esto establece un sistema de referencia de dos ejes en cuadratura d y q tanto en el estator como en el rotor, lo que permite aplicar los esquemas de control con relativa simplicidad. El sistema de control a implementar se muestra en la figura 1.18, donde se muestra la aplicación de la transformada de Park que se definirá más adelante.

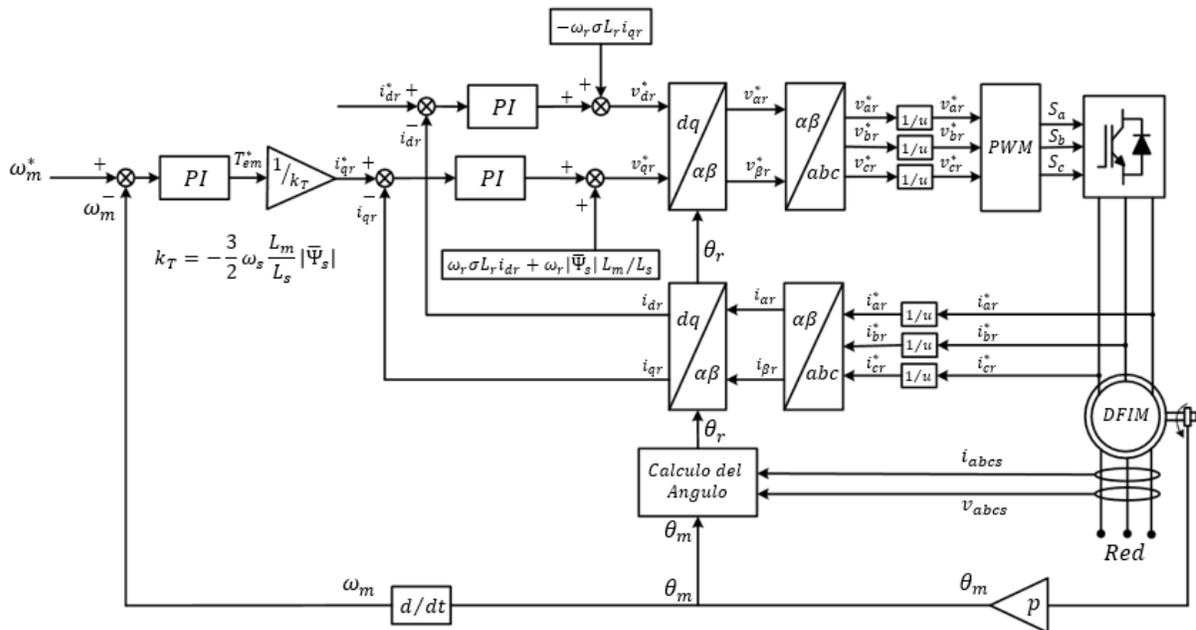


Figura 1.18 Diagrama de control de velocidad de la máquina.

## 1.8 Cambio de sistema de referencia (*dqo*)

### 1.8.1 Representación vectorial de magnitudes físicas

El espacio temporal se puede representar mediante tres ejes ortogonales  $a, b, c$ . Cualquier magnitud trifásica se puede definir a través de sus coordenadas  $x_a(t), x_b(t), x_c(t)$  sobre estos ejes, de modo que a cada valor de la variable  $t$  le corresponde un punto concreto, definiendo así un vector espacial  $S(t)$  que tiene como origen, el origen de coordenadas.

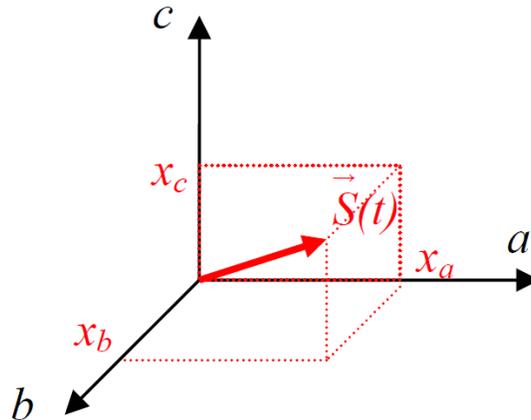


Figura 1.19 Vector espacial  $S(t)$  en el sistema de ejes  $(a, b, c)$ .

Todo vector se puede descomponer como suma de vectores. Resulta útil descomponer el vector espacial  $S(t)$  como suma de dos vectores perpendiculares,  $S'(t)$  y  $H(t)$ , donde  $H(t)$  es un vector formado por componentes homopolares  $h(t)$  iguales.

$$S(t) = S'(t) + H(t) \quad \text{Ecuación 1.24}$$

Donde

$$S'(t) = \begin{bmatrix} x'(t) \\ y'(t) \\ z'(t) \end{bmatrix} \quad H(t) = \begin{bmatrix} h(t) \\ h(t) \\ h(t) \end{bmatrix}$$

La componente homopolar  $h(t)$  se define como:

$$h(t) = \frac{x(t) + y(t) + z(t)}{3} \quad \text{Ecuación 1.25}$$

Los dos vectores permiten definir una nueva base de coordenadas. Utilizando la dirección  $[1,1,1]$  en coordenadas  $abc$  del vector  $H(t)$  define el eje  $\gamma$  del nuevo sistema, mientras que el vector  $S'(t)$  puesto que es perpendicular a  $H(t)$ , estará contenido en un plano denominado  $\Pi$  que es perpendicular al eje  $\gamma$  y definido por los ejes  $\alpha\beta$ , siendo  $\alpha$  la proyección del eje  $a$  en el plano. La figura 1.20 ilustra la descomposición del vector  $S(t)$  en dos vectores  $S'(t)$  y  $H(t)$  perpendiculares y los nuevos ejes  $\alpha\beta\gamma$ .

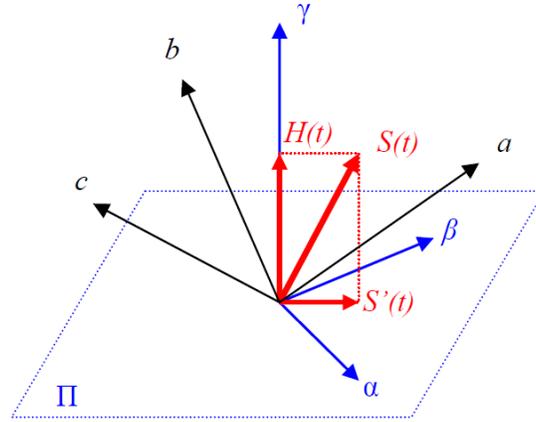


Figura 1.20 Descomposición vectorial del vector  $S(t)$ .

### 1.8.2 Transformada de Clarke

Si se considera un sistema trifásico simétrico y balanceado, se puede demostrar que las corrientes y tensiones trifásicas del sistema se pueden representar mediante un vector espacial de módulo constante que gira a velocidad constante en el plano  $\Pi$ . En primer lugar se define una tensión trifásica  $v(t)$  cuyas componentes  $a, b, c$  son tensiones senoidales simétricas y equilibradas, es decir, de igual módulo  $V_{max}$  y desfasadas  $120^\circ$ , tal y como muestra la ecuación 1.26.

$$v(t) = \begin{bmatrix} v_a \\ v_b \\ v_c \end{bmatrix} = \begin{bmatrix} V_{max} \cos(\omega t) \\ V_{max} \cos(\omega t - 2\pi/3) \\ V_{max} \cos(\omega t + 2\pi/3) \end{bmatrix} \quad \text{Ecuación 1.26}$$

Donde

$$v_a + v_b + v_c = 0 \quad \text{Ecuación 1.27}$$

Dado que el sistema es simétrico y equilibrado, la componente homopolar  $h(t)$  es nula, lo cual implica que  $H(t) = 0$  y  $S(t) = S'(t)$ , es decir, que el vector espacial está contenido en el plano  $\Pi$ .

Para deducir la trayectoria que describe el vector  $v(t)$  en el plano es útil realizar un cambio de base de los ejes  $abc$  a los ejes  $\alpha\beta\gamma$  mediante la aplicación de la matriz de cambio de base  $[T]_{abc \rightarrow \alpha\beta\gamma}$  definida en la siguiente expresión.

$$\begin{bmatrix} v_\alpha \\ v_\beta \\ v_\gamma \end{bmatrix} = [T]_{abc \rightarrow \alpha\beta\gamma} \begin{bmatrix} v_a \\ v_b \\ v_c \end{bmatrix} \quad \text{Ecuación 1.28}$$

Donde

$$[T]_{abc \rightarrow \alpha\beta\gamma} = \frac{2}{3} \begin{bmatrix} 1 & -\frac{1}{2} & -\frac{1}{2} \\ 0 & \frac{\sqrt{3}}{2} & -\frac{\sqrt{3}}{2} \\ \frac{1}{2} & \frac{1}{2} & \frac{1}{2} \end{bmatrix} \quad \text{Ecuación 1.29}$$

La transformación de Clarke  $[T]_{abc \rightarrow \alpha\beta\gamma}$  anterior preserva la amplitud de las variables eléctricas a las que se aplica. Si consideramos un sistema trifásico balanceado, como se mencionó anteriormente, se puede realizar una simplificación, dado que  $v_a + v_b + v_c = 0$  y por lo tanto  $v_\gamma = 0$  y considerar la transformación simplificada, que se muestra a continuación.

$$\begin{bmatrix} v_\alpha \\ v_\beta \end{bmatrix} = \frac{2}{3} \begin{bmatrix} 1 & -\frac{1}{2} & -\frac{1}{2} \\ 0 & \frac{\sqrt{3}}{2} & -\frac{\sqrt{3}}{2} \end{bmatrix} \begin{bmatrix} v_a \\ v_b \\ v_c \end{bmatrix} \quad \text{Ecuación 1.30}$$

Dicha transformación corresponde a la transformación original de Clarke con la tercera ecuación desechada. En un sistema simétrico y equilibrado, este razonamiento es cierto tanto para tensiones simples y compuestas, como para corrientes. No obstante, si el sistema no es equilibrado, no se cumple la relación de desfase de  $120^\circ$  entre las componentes de las tensiones fase-neutro y las corrientes, por lo que las deducciones anteriores sólo serán válidas para tensiones compuestas.

### 1.8.3 Transformación de Park

Una vez hecho el cambio de base, las tensiones y corrientes del sistema girarán con velocidad angular constante  $\omega$  sobre el plano  $\Pi$ , definido por los ejes  $\alpha\beta$ . No obstante, lo que realmente interesa es que los vectores espaciales de tensión y corrientes permanezcan fijos en una determinada posición, de modo que en régimen permanente sea un vector constante. Esto se consigue definiendo un sistema de referencia que gire sobre el plano  $\Pi$  a la misma velocidad  $\omega$  que el vector espacial, de modo que el vector permanezca fijo en el este nuevo sistema.

La nueva base se define mediante dos ejes  $dq$ , que giran en el plano  $\Pi$  en torno a un tercer eje  $o$ , coincidente con el eje  $\gamma$ . La figura 1.21 muestra cómo se realiza el cambio de ejes sobre el plano  $\Pi$ .

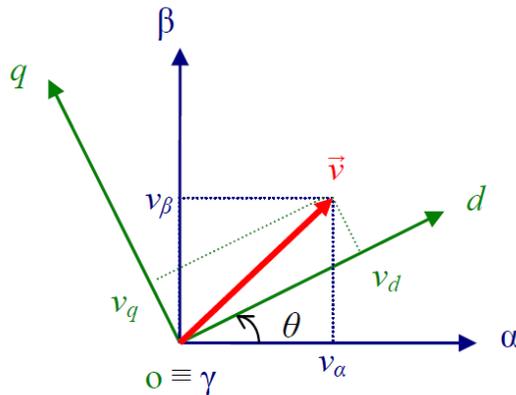


Figura 1.21 Cambio de base entre los ejes  $\alpha\beta\gamma$  a los nuevos ejes  $dqo$ .

A partir del ángulo  $\theta$  de rotación de los ejes  $dq$ , se puede definir la matriz  $[T]_{\alpha\beta\gamma \rightarrow dqo}$  que permite el cambio de base de  $\alpha\beta\gamma$  a  $dqo$ , y se define de la siguiente manera.

$$\begin{bmatrix} v_d \\ v_q \\ v_o \end{bmatrix} = [T]_{\alpha\beta\gamma \rightarrow dqo} \begin{bmatrix} v_\alpha \\ v_\beta \\ v_\gamma \end{bmatrix} \quad \text{Ecuación 1.31}$$

Donde

$$[T]_{\alpha\beta\gamma \rightarrow dqo} = \begin{bmatrix} \cos \theta & \sin \theta & 0 \\ -\sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad \text{Ecuación 1.32}$$

Finalmente, se ha conseguido cambiar el sistema de referencia del vector espacial desde el dominio temporal  $abc$ , hasta el dominio de ejes giratorios  $dq$ . Este cambio de ejes se conoce con el nombre de transformación de Park.

La transformada de Park es la matriz que combina las dos transformaciones anteriores  $[T]_{abc \rightarrow \alpha\beta\gamma}$  y  $[T]_{\alpha\beta\gamma \rightarrow dqo}$  para permitir el cambio de base directo desde  $abc$  a  $dqo$ . La matriz  $[T]_{abc \rightarrow dqo}$  resultante, es la matriz de transformación de Park.

$$[T]_{abc \rightarrow dqo} = \frac{2}{3} \begin{bmatrix} \cos(\theta) & \cos\left(\theta - \frac{2\pi}{3}\right) & \cos\left(\theta + \frac{2\pi}{3}\right) \\ -\sin(\theta) & -\sin\left(\theta - \frac{2\pi}{3}\right) & -\sin\left(\theta + \frac{2\pi}{3}\right) \\ \frac{1}{2} & \frac{1}{2} & \frac{1}{2} \end{bmatrix} \quad \text{Ecuación 1.33}$$

Y su matriz inversa

$$[T]_{dqo \rightarrow abc} = \frac{2}{3} \begin{bmatrix} \cos(\theta) & -\sin(\theta) & 1 \\ \cos\left(\theta - \frac{2\pi}{3}\right) & -\sin\left(\theta - \frac{2\pi}{3}\right) & 1 \\ \cos\left(\theta + \frac{2\pi}{3}\right) & -\sin\left(\theta + \frac{2\pi}{3}\right) & 1 \end{bmatrix} \quad \text{Ecuación 1.34}$$

En la nueva base los ejes  $dq$  giran sobre el plano  $\Pi$  a la misma velocidad angular  $\omega$  que el vector espacial  $v$ , tal y como muestra la figura 1.22. Este vector espacial aparecería como rotante en la base  $abc$ , pero al girar ahora a igual velocidad que los nuevos ejes, se transforma en un vector fijo en la base  $dqo$ .

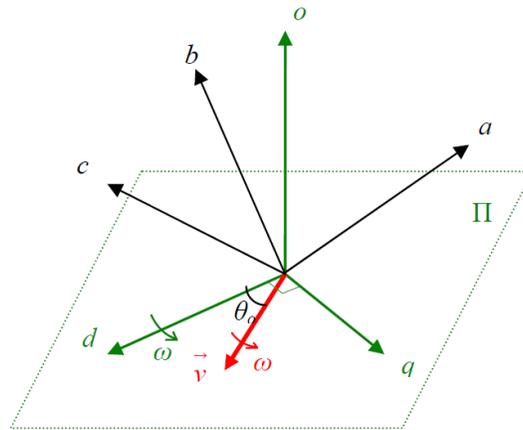


Figura 1.22 Representación del vector espacial  $v$  en la nueva base  $dqo$ .

### **Conclusiones del Capítulo 1.**

Conocer las variables físicas y eléctricas de la turbina es indispensable para hacer construir los modelos Aerodinámico y Mecánico-Eléctrico ya que a partir de esto se puede crear un sistema de control más sofisticado si es necesario, por ejemplo controlar la dirección de la góndola, para sistemas en los que es posible ajustar la dirección de ésta, para que el viento incida de manera perpendicular y se obtenga la máxima obtención de potencia del viento.

Además el conocer las variables físicas y eléctricas de la turbina nos facilitará la simulación del control digital, ya que mediante el código fuente creado en los softwares de lenguaje alto nivel y el modelado mismo de la turbina, se pueden cambiar los parámetros fácilmente sin tener que estar haciendo grandes cambios en el control digital de la turbina.

El estudio de las transformadas de Clarke y Park son de gran utilidad, ya que con estas podemos llevar a un plano más sencillo las variables de interés en el control digital de la turbina, porque sin estas transformadas el proceso, modelado y simulado de la turbina sería muy engorroso ya que las variables de interés estarían variando con el tiempo y necesitaríamos herramientas matemáticas más complejas.

**CAPÍTULO 2.**  
**MATLAB, SIMULINK Y LA**  
**LIBRERÍA SIMSCAPE.**

## CAPÍTULO 2.

### **MATLAB, SIMULINK Y LA LIBRERÍA SIMSCAPE.**

El software base para el desarrollo del modelado y simulaciones de la turbina eólica es MATLAB (MATrix LABoratory, es el acrónimo del programa) es una herramienta para programación de lenguaje de alto nivel y para un entorno interactivo para técnicas de computación, e incluye funciones y algoritmos desarrollados, análisis de datos, computación numérica y visualización.

Una de las herramientas más utilizadas de Matlab es Simulink, el cual es un programa que permite el modelado de sistemas empleando bloques, Simulink posee diferentes librerías en las cuales se encuentran almacenados los bloques que realizan funciones especiales y específicas.

Una de las librerías de Simulink es la librería Simscape, que permite crear modelos de sistemas físicos rápidamente dentro del entorno de Simulink. Con Simscape se pueden crear modelos de sistemas basados en los componentes y conexiones entre ellos.

En este capítulo se presentarán los bloques especiales de la librería Simscape que se utilizarán en la implementación del modelo de la turbina eólica. Para conocer los parámetros de cada uno de ellos, configuraciones, entre otros que nos permitan simular un sistema que se aproxime a la operación de un sistema real.

## 2.1 Matlab

Matlab es un entorno desarrollado donde se puede trabajar código de programación de una manera sencilla y fácil, posee herramientas que son muy útiles en la ingeniería, entre ellas está el diseño y construcción de procesos matemáticos que se pueden representar mediante vectores, matrices e interfaces creadas por el usuario, además, procesos de propósito general. En él se puede encontrar documentación sobre sus elementos de programación el cual facilita el uso y entendimiento.

Algunos usos y aplicaciones son:

- Matemáticas y cálculo.
- Desarrollo de algoritmos.
- Adquisición de datos.
- Modelado, simulación y prototipado.
- Análisis y procesamiento de datos.
- Gráficos científicos y de ingeniería.
- Desarrollo de aplicaciones.

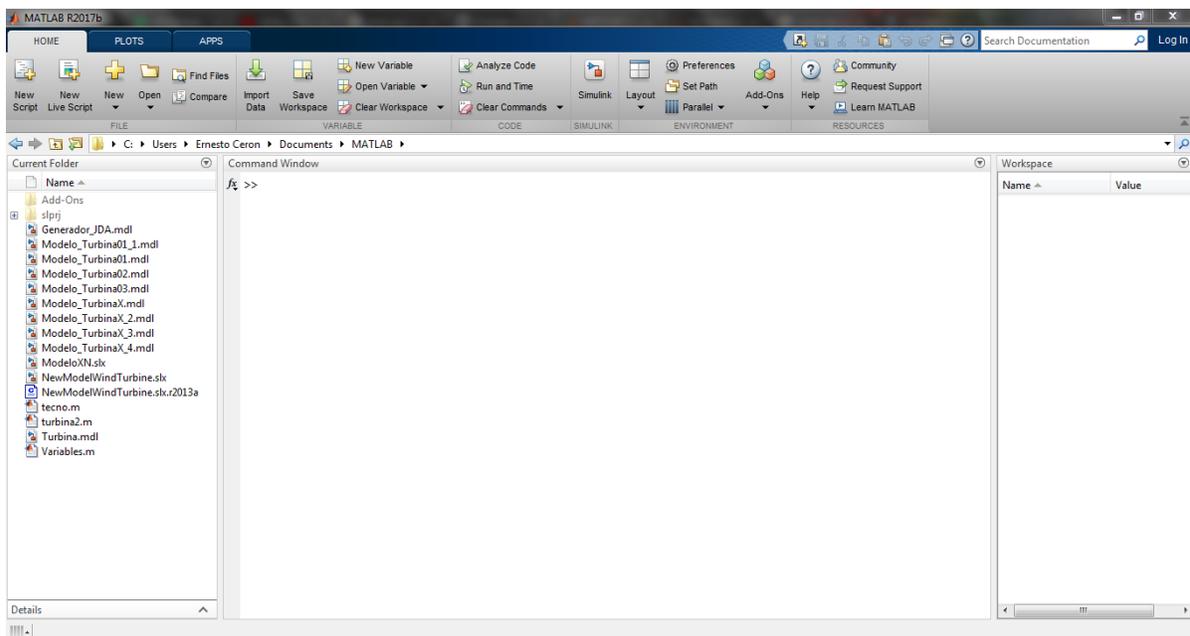


Figura 2.1 Interfaz de Matlab.

El programa se puede dividir en 4 bloques:

1. **Entorno de desarrollo.** Es el escritorio de trabajo del software, en él se encuentra un conjunto de herramientas que facilitan el uso de funciones en Matlab, así como ficheros. Muchas de estas utilidades son interfaces graficas de usuario. Incluye el espacio de trabajo Matlab y la ventana de comandos.
2. **La librería de funciones matemáticas Matlab.** Se trata de un amplio conjunto de algoritmos de cálculo, comprendiendo las funciones más elementales como la suma, senos y cosenos o la aritmética de números complejos, hasta funciones más sofisticadas como la inversión de matrices, funciones de Bessel y transformadas rápidas de Fourier.

3. **Gráficos.** Matlab dispone de un conjunto de utilidades destinadas a visualizar vectores y matrices en forma de gráficos. Existe una gran cantidad de posibilidades para ajustar el aspecto de los gráficos, destacando la visualización 3-D con opciones de iluminación y sombreado, y la posibilidad de crear animaciones.
4. **El Interfaz de Aplicación de Matlab (API).** Consiste en una librería que permite escribir programas ejecutables independientes en C y otros lenguajes, accediendo, mediante DLLs, a las utilidades de cálculo matricial de Matlab.

## 2.2 Simulink

Simulink es una herramienta que incluye Matlab, en ella se permite modelar cualquier sistema físico que se desee, turbinas eólicas, generadores, motores, aplicaciones de radiofrecuencia, procesamiento de señales, etc. Y crear sistemas de control en ellos, todo esto mediante diagrama de bloques, aparte de ello, Matlab permite el control e incluso la opción de crear un código fuente a partir del modelo creado en Simulink.

La ventaja de utilizar Simulink se da en la ingeniería de control, donde controlar un proceso o la observación de la realización de un trabajo es bien importante.

Para poder acceder a Simulink, se puede escribir en la misma palabra "simulink" en la línea de comandos del espacio de trabajo de Matlab o abriendo cualquier fichero de la computadora con la extensión ".mdl". Al abrir Simulink con cualquiera de las dos maneras descritas deberá aparecer el entorno de trabajo de Simulink y sus librerías, donde se encuentran los diagramas de bloques de las diferentes funciones que puede ofrecer. En la figura 2.2. Se encuentra el entorno de Simulink.

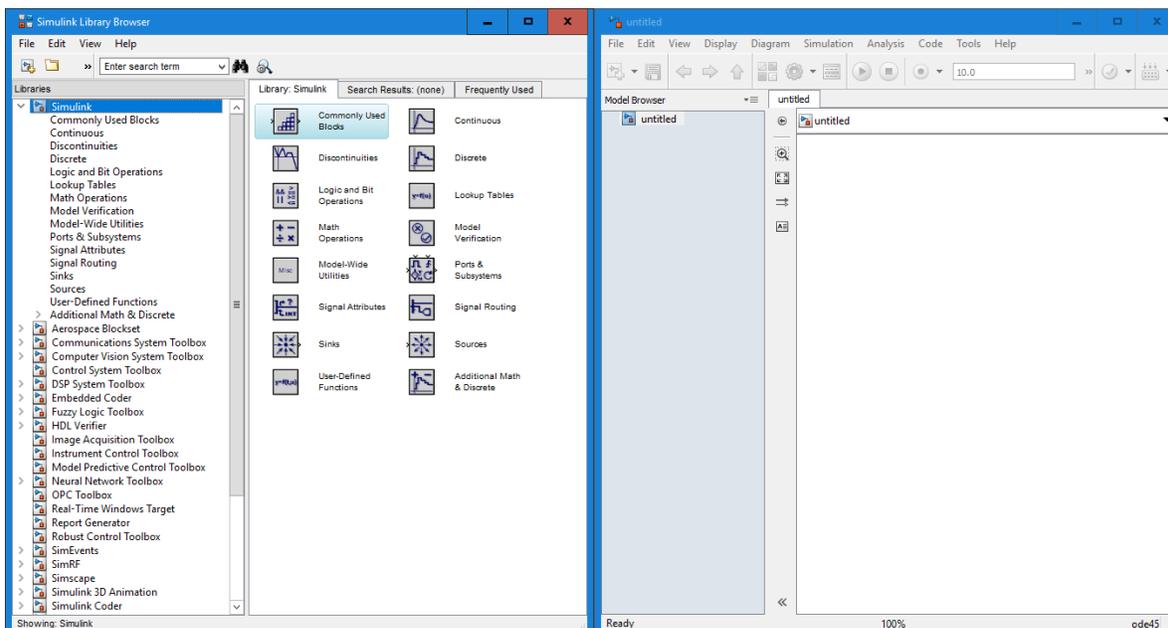


Figura 2.2 Librerías de Simulink (Izq.) y Área de trabajo para crear el modelo (Der.).

A la izquierda están las librerías de Simulink que son todos los bloques disponibles que se instalaron, la cantidad de estos está limitada según, la versión y los paquetes descargados a la hora de instalar Matlab.

De esta lista de bloques nos enfocamos en “*IDE link Embedded Target for TI C2000 DSP*” (el nombre varía según la versión instalada de Matlab), esta librería es de suma importancia ya que esta librería posee bloques que permiten el modelado, simulación y adaptación de la tarjeta eZdsp f2812.

Otra herramienta y que es de gran importancia que contiene Simulink, es el Real Time Workshop, que ya viene predeterminado en la instalación de Matlab.

*Real Time Workshop* automáticamente genera, ejecuta y compila un código fuente independiente del modelo de Simulink para la creación y desarrollo de aplicaciones de programa en tiempo real en una gran gama de sistemas, incluyendo aceleración de simulaciones, prototipado rápido y prueba de hardware.

Real Time Workshop es la base de la generación de código de Simulink. Este genera código C que cumple con el ANSI/ISO para un modelo entero o para un subsistema individual, habilitando el código para que se ejecute en cualquier microprocesador.

Real Time Workshop es una parte integral del ambiente de trabajo de Simulink. Se puede interactuar con él a través del explorador de modelos, donde se puede configurar los parámetros para la generación de código en Simulink. A través del explorador de modelos se puede:

- Generar código a partir de modelos o subsistemas en Simulink.
- Seleccionar un objetivo para el Real Time Workshop.
- Configurar el objetivo para la generación de código.

Real Time Workshop (RTW) construye los programas automáticamente para aplicaciones en tiempo real en una variedad de entornos, RTW compila y enlaza el código fuente generado. La figura 2.3 muestra como RTW ejecuta sus acciones.

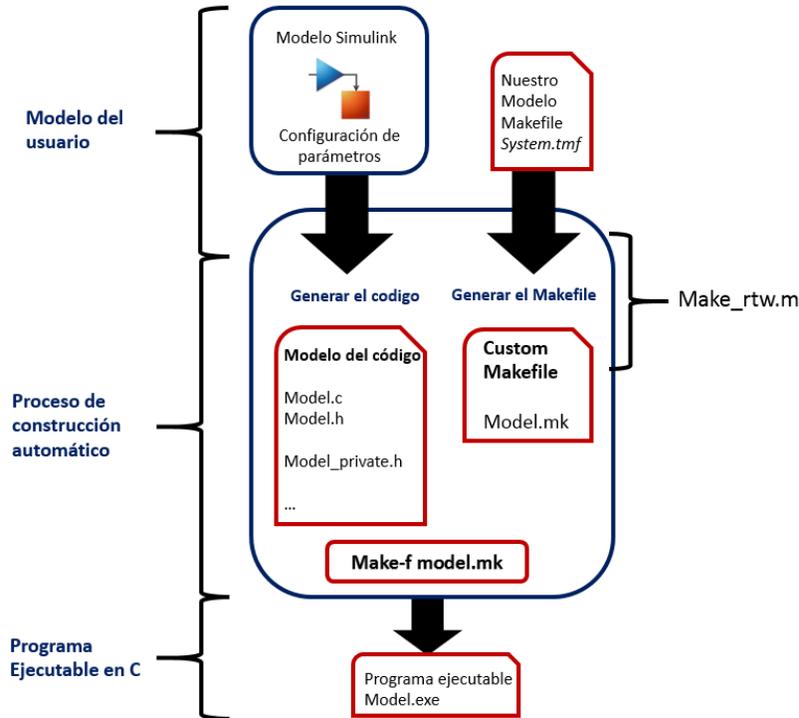


Figura 2.3 Ejecución pasó a paso del Real Time Workshop de Matlab.

Cuando el software de Real Time Workshop genera código, este produce los siguientes archivos:

- *model.c* o *model.cpp*: Código C o C++ generado del diagrama de bloques de Simulink. Este código implementa las ecuaciones del sistema del diagrama de bloques, así como también, inicializa y actualiza las salidas.
- *Model\_data.c* o *model\_data.cpp*: Archivo opcional que contiene los datos para parámetros y constantes del bloque de E/S, las cuales además deben ser declaradas externas.
- *Model.h*: Archivo cabecera que contiene los parámetros de simulación del diagrama de bloques, estructuras de E/S y otras definiciones.

Estos archivos son llamados en base al nombre del modelo de Simulink de donde se generan.

Ahora el Embedded Target for TI C2000 DSP se auxilia de la herramienta Real Time Workshop (RTW) para automáticamente generar, empaquetar y compilar el código fuente que se creó a partir del modelo en Simulink, de esta librería se profundizara en el capítulo 4.

### 2.3 Librería Simscape

La librería Simscape es una herramienta que permite modelar y simular sistemas físicos que involucren diversas áreas (mecánica, eléctrica, hidráulica, etc.) en el entorno de Simulink. La construcción de los modelos, se realiza en base a los componentes y conexiones físicas, por medio de bloques, lo que permite simular la operación del sistema, para desarrollar sistemas de control, realizar pruebas de rendimiento y estudiar el comportamiento del sistema ante alguna anomalía.

Debido a que la librería Simscape será de gran utilidad para la implementación de nuestro modelo en Simulink, es necesario realizar una descripción de los bloques a utilizar, para conocer los parámetros y la configuración que estos necesitan.

#### 2.3.1 Máquina Asíncrona

Modelo dinámico de una máquina asíncrona trifásica, también conocida como máquina de inducción.

##### Librería

Simscape/Power System/Specialized Technology/Fundamental Blocks/Machines.

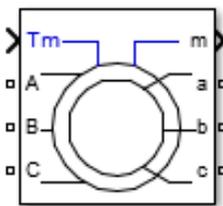


Figura 2.4 Bloque de la máquina asíncrona.

##### Descripción

El bloque de máquinas asíncronas, implementa una máquina asíncrona trifásica (rotor bobinado, jaula de ardilla simple o jaula de ardilla doble). Funciona en modo generador o motor. El modo de operación está dictado por el signo del par mecánico:

- Si  $T_m$  es positivo, la máquina actúa como un motor.
- Si  $T_m$  es negativo, la máquina actúa como un generador.

La parte eléctrica de la máquina está representada por un modelo de variables de estado de cuarto orden (o sexto orden para la máquina de doble jaula de ardilla) y la parte mecánica por un sistema de segundo orden. Todas las variables y parámetros eléctricos se refieren al estator, indicado por los signos principales en las siguientes ecuaciones de la máquina. Todas las cantidades de estator y rotor están en el marco de referencia arbitrario de dos ejes (marco  $dq$ ). Los subíndices utilizados se definen en la tabla 2.1.

Subíndice	Definición
d	Cantidad referida al eje d
q	Cantidad referida al eje q
r	Cantidad referida al rotor (rotor bobinado o de una sola jaula)
s	Cantidad referida al Estator
l	Inductancia de fuga
m	Inductancia de magnetización

Tabla 2.1 Subíndices de la máquina asíncrona.

### Sistema eléctrico del rotor bobinado o la máquina jaula de ardilla.

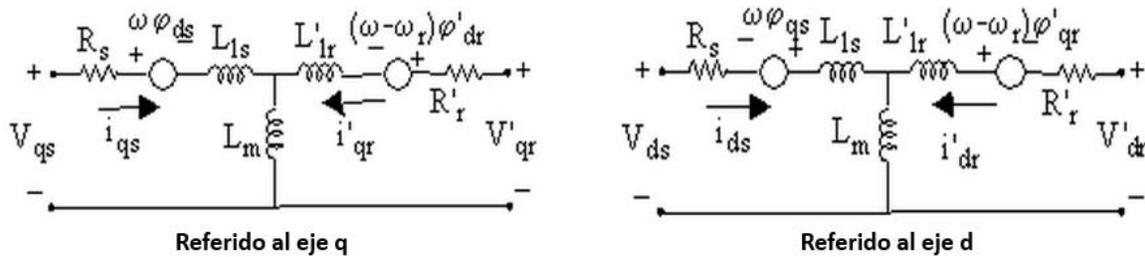


Figura 2.5 Modelo de la máquina asíncrona referida al eje q (izq.) y al eje d (der.).

Donde

$\omega$  — Velocidad angular de referencia

$\omega_r$  — Velocidad angular eléctrica

$$V_{qs} = R_s i_{qs} + d\phi_{qs}/dt + \omega\phi_{ds} \quad \text{Ecuación 2.1}$$

$$V_{ds} = R_s i_{ds} + d\phi_{ds}/dt - \omega\phi_{qs} \quad \text{Ecuación 2.2}$$

$$V'_{qr} = R'_r i'_{qr} + d\phi'_{qr}/dt + (\omega - \omega_r)\phi'_{dr} \quad \text{Ecuación 2.3}$$

$$V'_{dr} = R'_r i'_{dr} + d\phi'_{dr}/dt - (\omega - \omega_r)\phi'_{qr} \quad \text{Ecuación 2.4}$$

$$T_e = 1.5p(\phi_{ds}i_{qs} - \phi_{qs}i_{ds}) \quad \text{Ecuación 2.5}$$

$$\phi_{qs} = L_s i_{qs} + L_m i'_{qr} \quad \text{Ecuación 2.6}$$

$$\phi_{ds} = L_s i_{ds} + L_m i'_{dr} \quad \text{Ecuación 2.7}$$

$$\phi'_{qr} = L'_r i'_{qr} + L_m i_{qs} \quad \text{Ecuación 2.8}$$

$$\phi'_{dr} = L'_r i'_{dr} + L_m i_{ds} \quad \text{Ecuación 2.9}$$

$$L_s = L_{1s} + L_m \quad \text{Ecuación 2.10}$$

$$L'_r = L'_{lr} + L_m \quad \text{Ecuación 2.11}$$

### Sistema Mecánico

$$\frac{d}{dt} \omega_m = \frac{1}{2H} (T_e - F \omega_m - T_m) \quad \text{Ecuación 2.12}$$

$$\frac{d}{dt} \theta_m = \omega_m \quad \text{Ecuación 2.13}$$

Los parámetros del bloque máquina asincrónica se definen de la siguiente manera, ver tabla 2.2 (todas las cantidades se refieren al estator).

Parámetros	Definición
$R_s, L_{ls}$	Resistencia e inductancia de fuga del Estator
$L_m$	Inductancia de Magnetización
$L_s$	Inductancia del Estator
$V_{qs}, i_{qs}$	Voltaje y Corriente del Estator en el eje q
$V_{ds}, i_{ds}$	Voltaje y Corriente del Estator en el eje d
$\phi_{qs}, \phi_{ds}$	Flujos del Estator en el eje q y d
$\omega_m$	Velocidad angular del rotor
$\theta_m$	Posición angular del rotor
$p$	Numero de par de polos
$\omega_r$	Velocidad angular eléctrica ( $\omega_m \times p$ )
$\theta_r$	Posición angular eléctrica ( $\theta_m \times p$ )
$T_e$	Torque electromagnético
$T_m$	Torque mecánico en el eje
$J$	Combina los coeficientes de Inercia del rotor y la carga.
$H$	Combina los coeficientes de Inercia del rotor y la carga. (PU)
$F$	Combina los coeficientes de fricción viscosa del rotor y la carga
$L'_r$	Inductancia del Rotor
$R'_r, L'_{lr}$	Resistencia e inductancia de fuga del Rotor
$L_m$	Inductancia de Magnetización
$V'_{qr}, i'_{qr}$	Voltaje y Corriente del Rotor en el eje q
$V'_{dr}, i'_{dr}$	Voltaje y Corriente del Rotor en el eje d
$\phi'_{qr}, \phi'_{dr}$	Flujos del Rotor en el eje q y d

Tabla 2.2 Parámetros de la máquina asíncrona.

### Configuración

La librería Simscape cuenta con dos bloques para la máquina asíncrona, La diferencia entre ellos, es las Unidades para especificar los parámetros eléctricos y mecánicos de a máquina, siendo SI para unidades en el sistema internacional y PU para unidades en por unidad.

La ventana de configuración del bloque es similar a la mostrada en la figura 2.6.

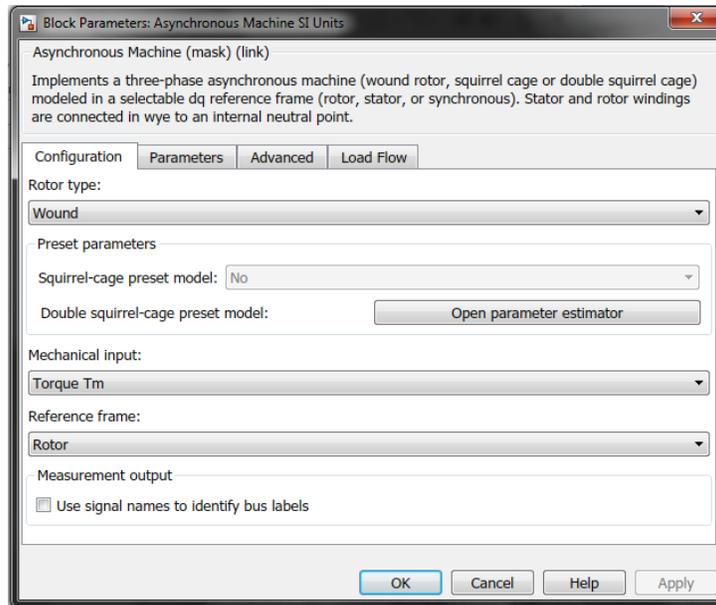


Figura 2.6 Ventana de configuración de la maquina asíncrona.

- **Tipo de rotor**

Especifica el tipo de rotor: Bobinado (por defecto para unidades SI), jaula de ardilla (por defecto para unidades PU), o jaula de ardilla doble. Al seleccionar uno rotor tipo jaula de ardilla ya sea simple o doble, el bloque cuenta con modelos preestablecidos.

#### **Modelos preestablecidos para el rotor tipo jaula de ardilla**

Proporciona un conjunto de modelos preestablecidos, con parámetros eléctricos y mecánicos predeterminados para varias clasificaciones de potencia de la máquina asíncrona (HP), voltaje de fase a fase (V), frecuencia (Hz) y velocidad nominal (rpm).

Cuando se selecciona un modelo preestablecido, los parámetros eléctricos y mecánicos en la pestaña Parámetros del cuadro de diálogo se vuelven no modificables (es decir, no disponible). Para comenzar desde un modelo preestablecido dado y luego modificar los parámetros de la máquina:

- Seleccione el modelo preestablecido que desea inicializar los parámetros.
- Cambie el valor del parámetro del modelo predefinido a No. Esto no cambia los parámetros de la máquina. Al hacerlo, acaba de romper la conexión con el modelo predeterminado particular.
- Modifique los parámetros de la máquina como desee, luego haga clic en Aplicar.

En la figura 2.7 se muestra el listado de modelos predefinidos.

#### **Modelos preestablecidos para el rotor doble jaula de ardilla**

Al igual que para la maquina asíncrona jaula de ardilla se cuenta con una serie de modelos predefinidos para la máquina de doble jaula.

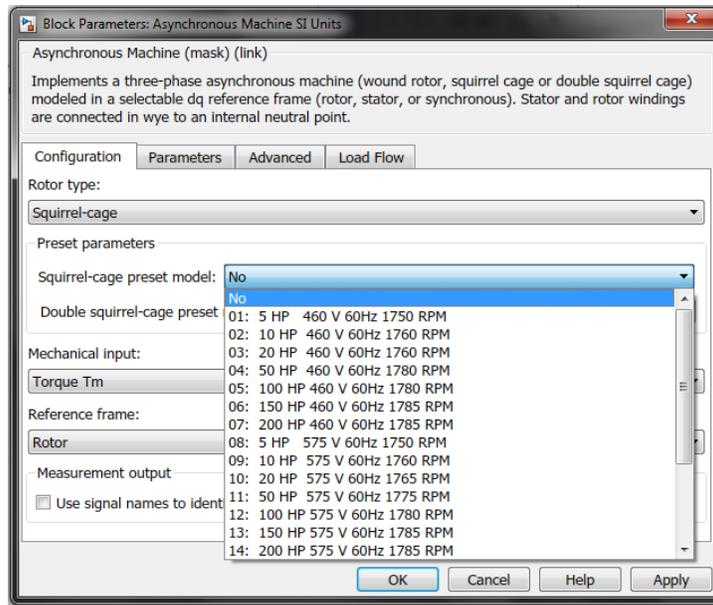


Figura 2.7 Selección de un modelo predefinido de una maquina con rotor Jaula de ardilla

- **Entrada mecánica**

Como entrada de la maquina asíncrona se puede seleccionar el par aplicado al eje o la velocidad del rotor. Por defecto se encuentra selecciona el torque  $T_m$  para especificar una entrada de par, en  $N.m$  o en PU, la velocidad de la máquina está determinada por la máquina Inercia  $J$  (o la constante de inercia  $H$  para la máquina en PU) y por la diferencia entre el par mecánico aplicado  $T_m$  y el par electromagnético interno  $T_e$ .

La convención de signos para el par mecánico es: cuando la velocidad es positiva, una señal de par positivo indica el modo del motor y una señal negativa indica el modo del generador.

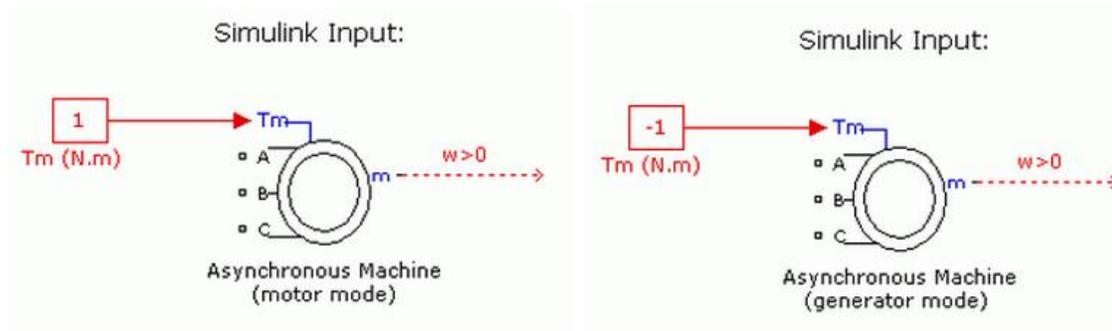


Figura 2.8 Convención de signos para la operación de la máquina.

Si se selecciona la velocidad del rotor  $\omega$  para especificar una entrada de velocidad, en  $rad/s$  o en PU. La velocidad de la máquina se impone y la parte mecánica del modelo (Inercia  $J$ ) se ignora. Usando la velocidad como la entrada mecánica permite modelar un acoplamiento mecánico entre dos máquinas.

## Parámetros

Esta pestaña permite definir el valor de los parámetros eléctricos y mecánicos de la máquina y además configurar condiciones iniciales de operación, como se muestra en la figura 2.9.

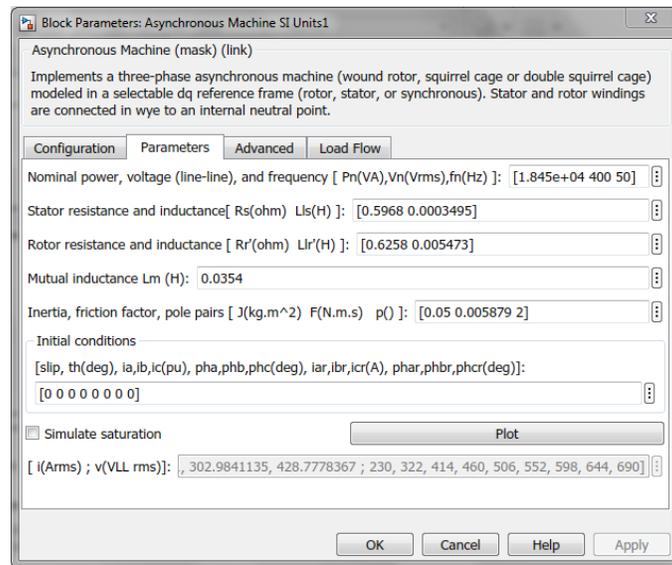


Figura 2.9 Ventana de parámetros de la maquina asíncrona.

- **Potencia nominal, voltaje (línea de línea) y frecuencia.**

La potencia aparente nominal  $P_n$  (VA), la tensión RMS línea a línea  $V_n$  (V) y la frecuencia  $f_n$  (Hz). El valor predeterminado es [3730 460 60] para unidades PU y [1.845e + 04 400 50] para unidades SI.

- **Resistencia e inductancia del estator**

La resistencia del estator  $R_s$  ( $\Omega$  o PU) y la inductancia de fuga  $L_{ls}$  (H o PU). El valor predeterminado es [0.01965 0.0397] para unidades PU y [0.5968 0.0003495] para unidades SI.

- **Resistencia e inductancia del rotor**

La resistencia del rotor  $R_r'$  ( $\Omega$  o PU) y la inductancia de fuga  $L_{lr}'$  (H o PU), ambas referidas al estator. Este parámetro solo es visible cuando el parámetro de tipo Rotor en la pestaña Configuración está configurado como bobinado o jaula de ardilla. El valor predeterminado es [0.01909 0.0397] para unidades PU y [0.6258 0.005473] para unidades SI.

- **Inductancia mutua**

La inductancia de magnetización  $L_m$  (H o PU). El valor predeterminado es 1.354 para unidades PU y 0.0354 para unidades SI.

- **Constante de inercia, factor de fricción y pares de polos.**

Para el cuadro de diálogo de unidades SI: el coeficiente combinado de inercia de la máquina y la carga  $J$  ( $kg.m^2$ ), el coeficiente de fricción viscoso combinado  $F$  ( $N.m.s$ ) y los pares de polos  $p$ . El par de fricción  $T_f$  es proporcional a la velocidad del rotor  $\omega$  ( $T_f = F.\omega$ ). El valor predeterminado es [0.05 0.005879 2].

Para el cuadro de diálogo unidades en PU: la constante de inercia  $H$  (s), el coeficiente de fricción viscoso combinado  $F$  (PU) y los pares de polos  $p$ . El valor predeterminado es [0.09526 0.05479 2].

- **Condiciones iniciales**

Especifica el deslizamiento inicial  $s$ , el ángulo eléctrico  $\theta_e$  (grados), la magnitud de la corriente del estator (A o pu) y los ángulos de fase (grados):

[*slip, th, ias, ibs, ics, phaseas, phasebs, phasecs*]

Si el parámetro Tipo de rotor está configurado como Bobinado, también puede especificar valores iniciales opcionales para la magnitud de la corriente del rotor (A o pu) y ángulos de fase (grados):

[*slip, th, ias, ibs, ics, phaseas, phasebs, phasecs, iar, ibr, ibr, phasear, phasebr, phasecr*]

El valor predeterminado es [1,0 0,0 0,0,0] para unidades PU y [0 0 0 0 0 0 0] para unidades SI.

### Avanzada

Las opciones avanzadas de la maquina permite configurar el tiempo de muestreo y el método de solución que se aplicara.

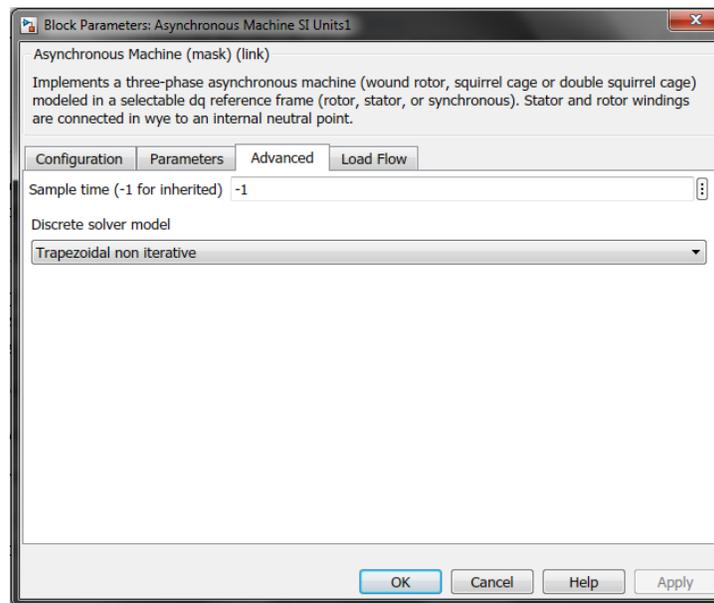


Figura 2.10 Ventana de opciones avanzadas de la maquina asincrona.

- **Tiempo de muestreo**

Especifica el tiempo de muestreo utilizado por el bloque. Para heredar el tiempo de muestreo especificado en el bloque *Powergui* (el cual se describirá más adelante), establezca este parámetro en -1 (valor predeterminado).

- **Modelo de solución discreta.**

Especifica el método de integración utilizado por el bloque cuando el parámetro de tipo *Solver* del bloque *Powergui* se establece en Discreto. Las opciones son: Trapezoidal no iterativo (predeterminado) y Trapezoidal iterativo.

La opción Forward Euler ya no se recomienda para discretizar modelos de máquina porque requiere que agregue carga de derivación no despreciable en los terminales de la máquina para garantizar la estabilidad de la simulación.

### Flujo de carga

Los parámetros en esta pestaña los utiliza la herramienta *Load Flow* del bloque *Powergui*. Estos parámetros de flujo de carga se usan solo para la inicialización del modelo. No tienen impacto en el modelo de bloque o en el rendimiento de la simulación.

### Potencia mecánica (W)

Se especifica la potencia mecánica aplicada al eje de la máquina, en vatios. Cuando la máquina funciona en modo motor, especifique un valor positivo. Cuando la máquina funciona en modo generador, especifique un valor negativo. El valor predeterminado es 0 para unidades PU y  $1.492e+006$  para unidades SI, como se muestra en la figura 2.11.

### Entradas y salidas

Los terminales del estator del bloque de la máquina asíncrona se identifican con las letras A, B y C, los terminales del rotor se identifican con las letras a, b y c. Las conexiones neutras del estator y del devanado del rotor no están disponibles; se asumen conexiones en Y de tres hilos.

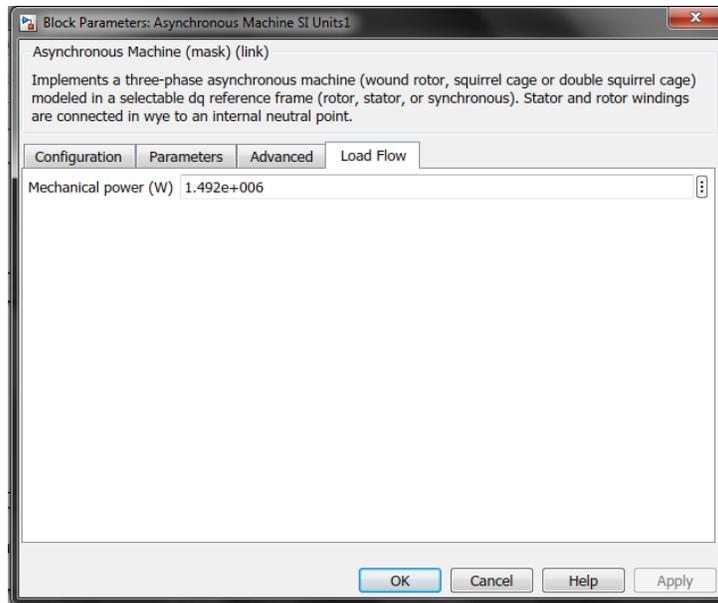


Figura 2.11 Ventana Load Flow de la máquina asíncrona.

- **T<sub>m</sub>**

Es la entrada del par mecánico en el eje de la máquina. Cuando la entrada es una señal positiva, la máquina asíncrona se comporta como un motor. Cuando la entrada es una señal negativa, la máquina asíncrona se comporta como un generador.

Si el bloque esta en unidades del sistema internacional SI, la señal de entrada será en  $N \cdot m$ , de lo contrario esta en PU.

- $w$

Una entrada alternativa del bloque es la velocidad de la máquina. Dependiendo del sistema de unidades seleccionado para parámetros SI, la entrada es una señal en  $rad/s$  o en PU.

- $m$

La salida del bloque es un vector que contiene señales de medición de variables eléctricas del rotor y del estator, así como de variables mecánicas. Estas señales se pueden separar utilizando el bloque “Bus Selector” ubicado en la librería “Bloques Comúnmente Utilizados” de Simulink. Dependiendo del sistema de unidades que utilice, las unidades están en el sistema internacional o en PU.

Nombre	Definición	Unidades
iar	Corriente de Rotor $i_{r\_a}$	A o PU
ibr	Corriente de Rotor $i_{r\_b}$	A o PU
icr	Corriente de Rotor $i_{r\_c}$	A o PU
iqr	Corriente de Rotor $i_q$	A o PU
idr	Corriente de Rotor $i_d$	A o PU
phiqr	Flujo del Rotor $\phi_{ir\_q}$	V.s o PU
phidr	Flujo del Rotor $\phi_{ir\_d}$	V.s o PU
vqr	Voltaje de Rotor $V_{r\_q}$	V o PU
vdr	Voltaje de Rotor $V_{r\_d}$	V o PU
ias	Corriente del Estator $i_{s\_a}$	A o PU
ibs	Corriente del Estator $i_{s\_b}$	A o PU
ics	Corriente del Estator $i_{s\_c}$	A o PU
iqs	Corriente del Estator $i_{s\_q}$	A o PU
ids	Corriente del Estator $i_{s\_d}$	A o PU
phiqs	Flujo del Estator $\phi_{ir\_q}$	V.s o PU
phids	Flujo del Estator $\phi_{ir\_d}$	V.s o PU
vqs	Voltaje del Estator $V_{r\_q}$	V o PU
vds	Voltaje del Estator $V_{r\_d}$	V o PU
w	Velocidad del Rotor	rad/s
Te	Torque Electromagnético	N.m o PU
theta	Angulo del Rotor $\theta_{m}$	rad

Tabla 2.3 Variables medidas de la maquina asíncrona.

### 2.3.2 Fuente de tensión trifásica programable

Implementa una fuente de tensión trifásica con variación de amplitud, fase, frecuencia y armónicos programable en el tiempo.

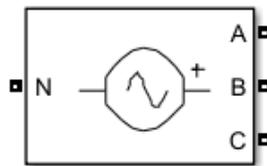


Figura 2.12 Bloque de la fuente de tensión trifásica programable.

### Librería

Simscape/Power System/Specialized Technology/Fundamental Blocks/Electrical Sources

## Descripción

Este bloque se utiliza para generar una tensión sinusoidal trifásica con parámetros variables en el tiempo. Puede programar la variación de tiempo para la amplitud, fase o frecuencia de la componente fundamental de la fuente. Además, dos armónicos se pueden programar y superponer a la señal fundamental.

La ventana para configurar dicho bloque es como la que se muestra en la figura 2.13.

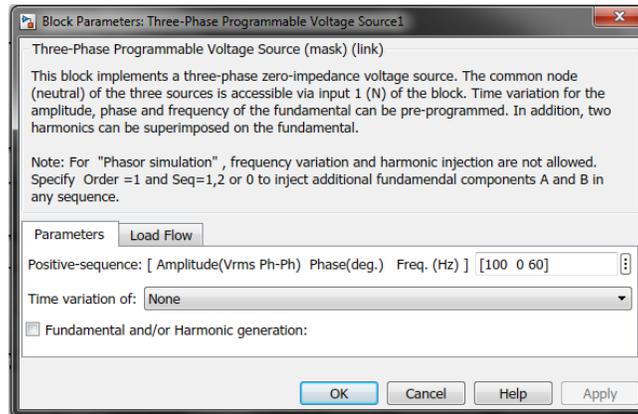


Figura 2.13 Ventana de parámetros de la fuente de tensión trifásica.

## Parámetros

- **Secuencia positiva**

Se introduce la amplitud en voltios RMS fase a fase, la fase en grados y la frecuencia en hercios de la componente de secuencia positiva de los tres voltajes. El valor predeterminado es [100 0 60].

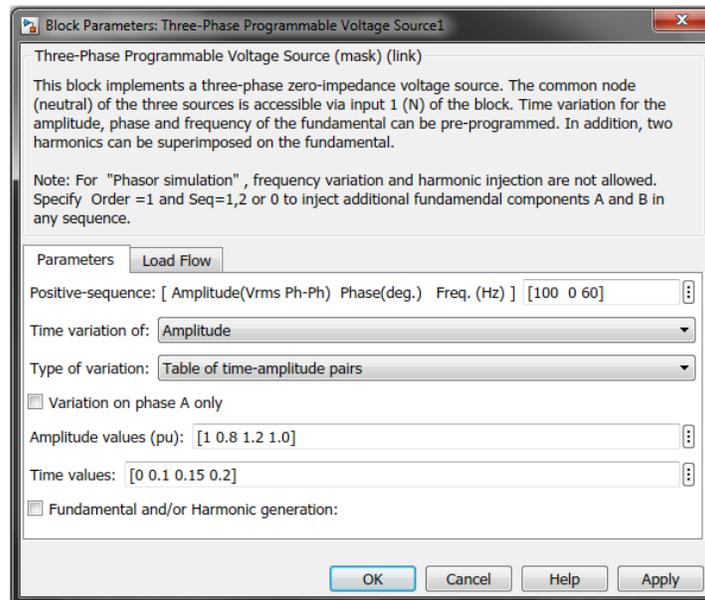


Figura 2.14 Programación de cambios en el tiempo de los parámetros de la fuente.

- **Variación de tiempo de**

Dado que esta es una fuente programable, nos permite realizar variaciones de la amplitud, la fase y la frecuencia a lo largo del tiempo de simulación, esta opción nos permite el análisis de respuestas del sistema ante anomalía.

La variación se aplica a las tres fases de la fuente por igual, excepto cuando el parámetro “Tipo de Variación” se establezca en “Tabla de pares”. En ese caso, puede aplicar una variación a la fase A solamente.

### **Tipo de variación**

Selecciona el tipo de cambio a realizar en el parámetro seleccionado

- *Step* para programar una variación por pasos.
- *Ramp* para programar una variación de rampa.
- *Modulation* para programar una variación modulada.
- *Table of time-amplitude pairs* para programar una serie de cambios escalonados de amplitudes en momentos específicos.

La opción *Table of time-amplitude pair* está disponible solo cuando la variación de tiempo del parámetro se establece en Amplitud. La tabla de pares de amplitud es la predeterminada cuando la variación de tiempo de está configurada en Amplitud. El paso es el predeterminado cuando la variación de tiempo de está configurada en Fase o Frecuencia.

- **Variación en la fase A solamente**

La variación de tiempo programada de la amplitud se aplica solo a la fase A del bloque de fuente de tensión programable trifásica. El valor predeterminado está despejado.

Este parámetro solo se habilita cuando la variación de tiempo del parámetro se establece en Amplitud y el parámetro Tipo de variación se establece en Tabla de pares de amplitud.

- **Valores de amplitud (PU)**

Especifica los puntos de variación de amplitud tomados por la fuente de voltaje durante la simulación. Los valores especificados están en PU, en función de la amplitud de secuencia positiva de la fuente de tensión. Los tiempos correspondientes donde tendrá lugar la variación se definen mediante el parámetro de Valores de tiempo. El valor predeterminado es [1 0.8 1.2 1.0].

Este parámetro solo se habilita cuando la variación de tiempo del parámetro se establece en Amplitud y el parámetro Tipo de variación se establece en *Table of time-amplitude pair*.

- **Valores de tiempo**

Especifica los tiempos de transición, en segundos, de la variación de amplitud definida en el parámetro Valores de amplitud (PU). El valor predeterminado es [0 0.1 0.15 0.2].

Este parámetro solo se habilita cuando la variación de tiempo del parámetro se establece en Amplitud y el parámetro Tipo de variación se establece en *Table of time-amplitude pair*.

- **Magnitud del paso**

Este parámetro está habilitado solo si el parámetro Tipo de variación está configurado en *Step*.

Especifica la amplitud del cambio de paso. La variación de la amplitud se especifica en PU de la amplitud de secuencia positiva. El valor predeterminado es 0.5.

- **Tasa de cambio**

Este parámetro está habilitado solo si el parámetro Tipo de variación está configurado en *Ramp*.

Especifica la tasa de cambio, en *voltios/segundos*. La velocidad de cambio de voltaje se especifica en (PU del voltaje de secuencia positiva) sobre segundo. El valor predeterminado es 10.

- **Amplitud de la modulación**

Este parámetro solo está habilitado si el parámetro Tipo de variación se establece en *Modulation*.

Especifica la amplitud de la modulación para el parámetro fuente que se especifica en la variación de tiempo del parámetro. Cuando la cantidad variable es la amplitud de voltaje, la amplitud de la modulación se especifica en PU de la amplitud de secuencia positiva. El valor predeterminado es 0.3.

- **Frecuencia de la modulación**

Este parámetro solo está habilitado si el parámetro Tipo de variación se establece en *Modulation*.

Especifica la frecuencia de la modulación para el parámetro fuente especificado en la variación de tiempo del parámetro. El valor predeterminado es 2.

- **Tiempo (s) de variación**

Especifica el tiempo, en segundos, cuando la variación programada de tiempo surte efecto y el tiempo cuando se detiene. El valor predeterminado es [1 2].

- **Generación fundamental y/o armónica**

Al seleccionar esta opción, se puede programar dos armónicos para superponerse a la tensión fundamental de la fuente. El valor predeterminado está despejado.

#### **A: [Orden Amplitud Fase Seq]**

Este parámetro solo está habilitado si está seleccionada la casilla de verificación "Generación fundamental y/o Armónica".

Se debe especificar el orden, la amplitud, la fase y el tipo de secuencia (1 = secuencia positiva, 2 = secuencia negativa, 0 = secuencia cero) del primer armónico que se superpondrá a la señal fundamental. La tensión del armónico se especifica en PU del voltaje de secuencia positiva. El valor predeterminado es [3 0.2 -25 0].

#### **B: [Orden Amplitud Fase Seq]**

Este parámetro solo está habilitado si está seleccionada la casilla de verificación "Generación fundamental y/o Armónica".

Se debe especificar el orden, la amplitud, la fase y el tipo de secuencia (0 = secuencia cero, 1 = secuencia positiva, 2 = secuencia negativa) del segundo armónico que se superpondrá a la señal fundamental. La tensión del armónico se especifica en PU del voltaje de secuencia positiva. El valor predeterminado es [2 0.15 35 2].

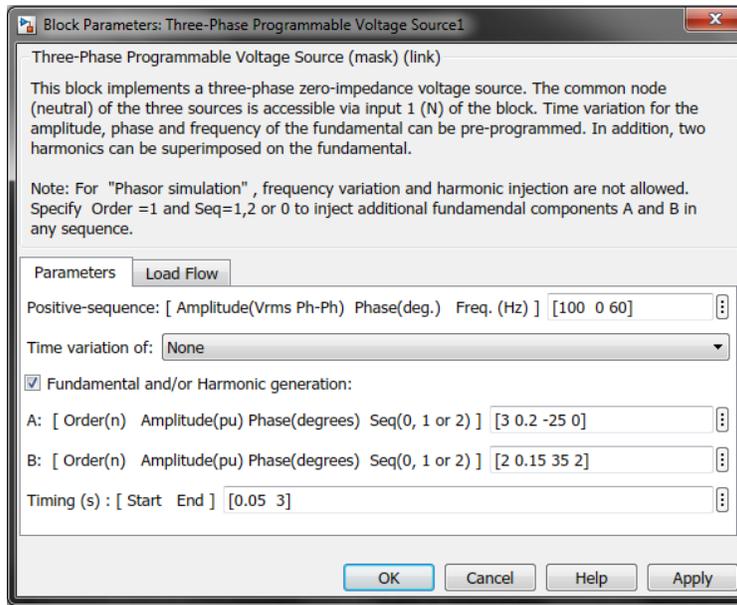


Figura 2.15 Configuración de armónicos en la fuente tensión.

### Tiempo (s)

Este parámetro está habilitado solo si está seleccionada la casilla de verificación “Generación fundamental y/o Armónica”.

En este parámetro especifica el tiempo en segundos, cuando la generación armónica se superpone a la señal fundamental y el momento en que se detiene. El valor predeterminado es [0.05 3].

### Flujo de carga

Los parámetros de flujo de carga se utilizan para definir parámetros de bloque para su uso con la herramienta *Load Flow* del bloque *Powergui*. Estos parámetros de flujo de carga se usan solo para la inicialización del modelo. No tienen impacto en el modelo de bloque y en el rendimiento de la simulación.

La configuración de la pestaña Flujo de carga depende de la opción seleccionada para el parámetro de tipo de generador.

### Tipo de generador

**Oscilante** (“*Swing*”, predeterminado), para implementar un generador que controle su magnitud y ángulo de fase de su voltaje.

**PV** para implementar un generador que controle su potencia activa de salida P y la magnitud de tensión V.

**PQ** para implementar un generador que controla su potencia activa de salida P y potencia reactiva Q.

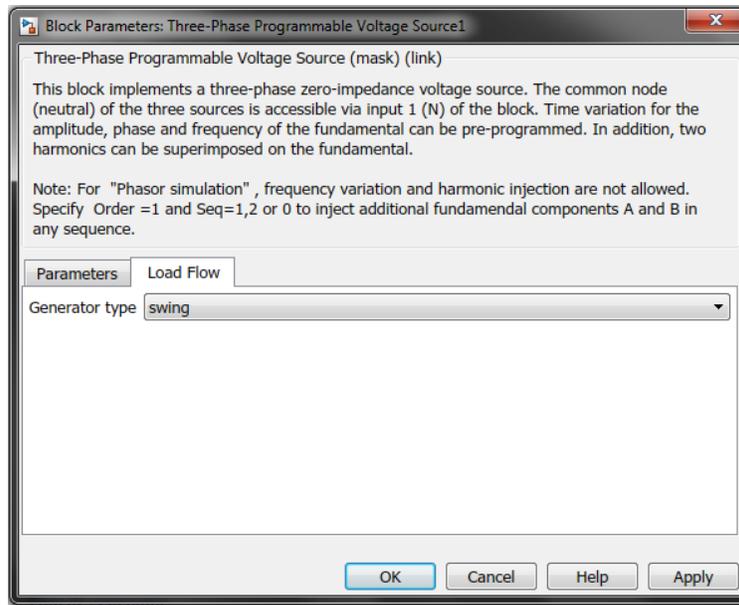


Figura 2.16 Ventana Load Flow de la fuente trifásica.

### 2.3.3 Medición trifásica V-I

Mide corrientes y tensiones trifásicas en el circuito

#### Librería

Simscape/Power System/Specialized Technology/Fundamental Blocks/Measurements

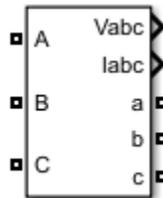


Figura 2.17 Bloque de medición trifásica V-I.

#### Descripción

El bloque de medición trifásico V-I se utiliza para medir voltajes y corrientes trifásicas instantáneas en un circuito. Cuando se conecta en serie con elementos trifásicos, devuelve los voltajes y corrientes pico de fase a tierra o de fase a fase.

El bloque puede emitir los voltajes y las corrientes en valores de unidad (PU) o en voltios y amperios  $V_{nom}$  y  $P_{base}$  son especificados en el cuadro de dialogo mostrado en la figura 2.18.

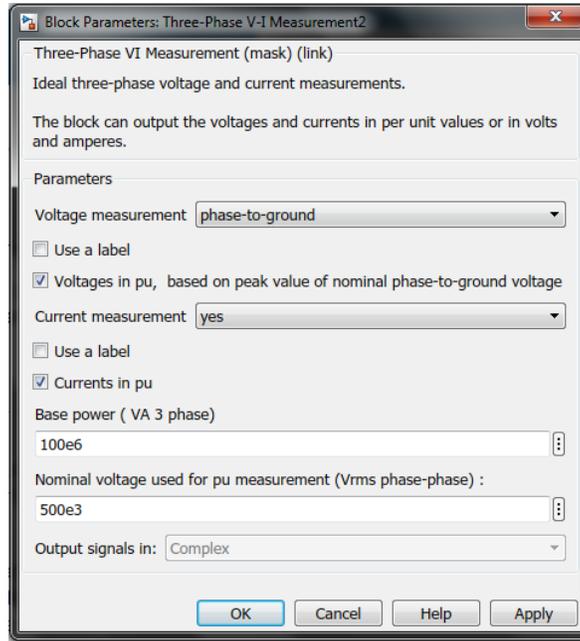


Figura 2.18 Configuración del bloque de medición trifásica.

Si se selecciona medir voltajes de fase a tierra por unidad, el bloque convierte los voltajes medidos en función del valor pico de la tensión nominal de fase a tierra:

$$V_{abc}(pu) = \frac{V_{fase\ a\ tierra} (V)}{V_{base} (V)} \quad \text{Ecuación 2.14}$$

Donde.

$$V_{base} = \frac{V_{nom}(V_{rms})}{\sqrt{3}} \times \sqrt{2} \quad \text{Ecuación 2.15}$$

Si se selecciona medir voltajes de fase a fase en por unidad, el bloque convierte los voltajes medidos en función del valor pico de la tensión nominal de fase a fase:

$$V_{abc}(pu) = \frac{V_{fase\ a\ fase} (V)}{V_{base} (V)} \quad \text{Ecuación 2.16}$$

Donde.

$$V_{base} = V_{nom}(V_{rms}) \times \sqrt{2} \quad \text{Ecuación 2.17}$$

En cuanto a la corriente si se selecciona medir las corrientes en por unidad, el bloque convierte las corrientes medidas en función del valor máximo de la corriente nominal:

$$I_{abc}(pu) = \frac{I_{abc} (A)}{I_{base} (A)} \quad \text{Ecuación 2.18}$$

Donde.

$$I_{base} = \frac{P_{base}}{V_{nom}} \times \frac{\sqrt{2}}{\sqrt{3}} \quad \text{Ecuación 2.19}$$

Los fasores de voltaje y corriente de estado estacionario, medidos por el bloque de medición V-I trifásico se pueden obtener del bloque Powergui seleccionando voltajes y corrientes de estado estable. Las magnitudes fasoriales que se muestran en el *Powergui* permanecen en valores máximos o RMS incluso si las señales de salida se convierten en PU.

## Parámetros

- **Medida de voltaje**

Si únicamente se desea medir la corriente se debe seleccionar la opción no, en caso de que la medición de tensión sea requerida se deberá especificar si esta es fase a tierra o fase a fase.

- **Usa una etiqueta**

El habilitar este campo, la señal de voltaje medida recibe una etiqueta, especificada en la etiqueta de la señal, lo que permite el uso de un bloque From para leer los voltajes, dicho bloque tendrá que hacer referencia a la etiqueta especificada por el parámetro etiqueta de la señal. El valor predeterminado está despejado.

- **Etiqueta de señal**

Especifica una etiqueta para las mediciones de voltaje. El valor predeterminado es Vabc.

- **Voltajes en PU, basados en el valor máximo de voltaje nominal de fase a tierra**

Los voltajes de fase a tierra medidos se convierten en PU. El valor predeterminado está despejado.

- **Voltajes en PU, basados en el valor pico de la tensión nominal de fase a fase**

Los voltajes de fase a fase medidos se convierten en PU. Este parámetro no está disponible si el parámetro de medición de voltaje está configurado en fase a tierra o no. El valor predeterminado está despejado.

- **Tensión nominal utilizada para la medición de la PU (fase-fase Vrms)**

La tensión nominal, en voltios RMS, utilizada para convertir los voltajes medidos en PU. Este parámetro no está disponible si el parámetro de medición de voltaje está establecido en no. El valor predeterminado es 500e3.

- **Medida de Corriente**

Por defecto el valor seleccionado es Sí, lo que permite medir las corrientes trifásicas que fluyen a través del bloque. Si únicamente se desea medir la tensión se debe seleccionar la opción no

- **Usa una etiqueta**

Al igual que para la tensión al habilitar este campo, la señal de corriente medida recibe una etiqueta, especificada en la etiqueta de la señal, lo que permite el uso de un bloque From para leer las corrientes, dicho bloque tendrá que hacer referencia a la etiqueta especificada por el parámetro etiqueta de la señal. El valor predeterminado está despejado.

- **Etiqueta de señal**

Especifica una etiqueta de etiqueta para las mediciones de corriente. El valor predeterminado es Iabc.

- **Corrientes en PU**

Las corrientes trifásicas se convierten en PU, de lo contrario, se miden en amperios. El valor predeterminado está despejado.

- **Potencia de base (VA trifásicos)**

La potencia de base trifásica, en voltios-amperios (VA), se utiliza para convertir las corrientes medidas en PU. El parámetro de potencia Base (VA trifásicos) no está disponible si no se seleccionan Corrientes en PU.

- **Señales de salida en**

Especifica el formato de las señales medidas cuando el bloque se usa en una simulación de fasores. Las señales de salida en el parámetro no están disponibles cuando el bloque no se utiliza en una simulación de fasores. La simulación de fasores se activa mediante un bloque *Powergui* colocado en el modelo.

- **Complejo** (predeterminado) para enviar las tensiones y corrientes medidas como valores complejos.
- **Real-Imag** para generar las partes real e imaginaria de las tensiones y corrientes medidas.
- **Magnitude-Angle** para generar las magnitudes y ángulos de las tensiones y corrientes medidas.
- **Magnitude** para generar las magnitudes de las tensiones y corrientes medidas. La salida es un valor escalar.

### **Entradas y salidas**

#### **Vabc**

Los tres voltajes medidos fase a tierra o fase a fase, la salida de Vabc desaparece cuando se selecciona el parámetro Usar una etiqueta o cuando el menú de medición de Voltaje se establece en no.

#### **Iabc**

Las tres corrientes de línea medidas, la salida Iabc desaparece cuando se selecciona el parámetro Usar etiqueta o cuando el menú Medición actual se establece en no.

### **2.3.4 Puente Universal**

Permite implementar un convertidor de potencia universal, seleccionando los dispositivos electrónicos de potencia, así como las topologías.

#### **Librería**

Simscape/Power System/Specialized Technology/Fundamental Blocks/Power Electronic

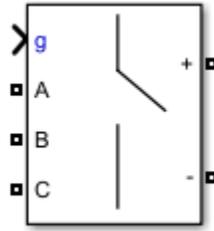


Figura 2.19 Bloque del puente universal.

### Descripción

El bloque Universal Bridge implementa un convertidor de potencia trifásico universal que consta de hasta seis interruptores de potencia conectados en una configuración de puente. El tipo de interruptor de alimentación y la configuración del convertidor se pueden seleccionar desde el cuadro de diálogo.

El bloque Universal Bridge permite la simulación de conversores que utilizan dispositivos electrónicos de potencia conmutados naturalmente (o conmutados por línea) (diodos o tiristores) y dispositivos de conmutación forzada (GTO, IGBT, MOSFET).

El bloque Universal Bridge es el bloque básico para construir convertidores de fuente de voltaje de dos niveles (VSC).

La numeración del dispositivo es diferente si los dispositivos electrónicos de potencia se conmutan naturalmente o se conmutan forzosamente. Para un convertidor trifásico con conmutación natural, la numeración sigue el orden natural de conmutación mostrado en la figura 2.20.

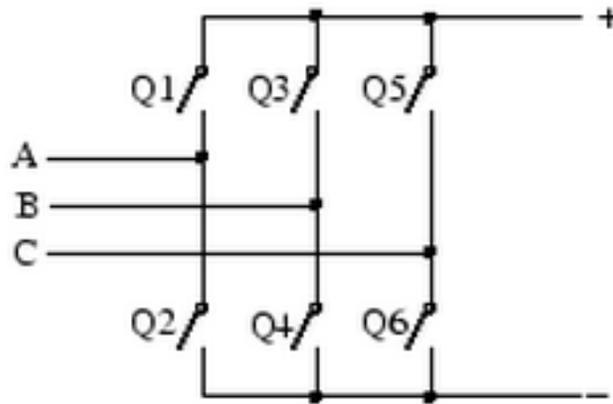


Figura 2.20 Orden de conmutación de los dispositivos electrónicos de potencia.

En la figura 2.20 se muestra el cuadro de diálogo que permite configurar los parámetros del puente universal.

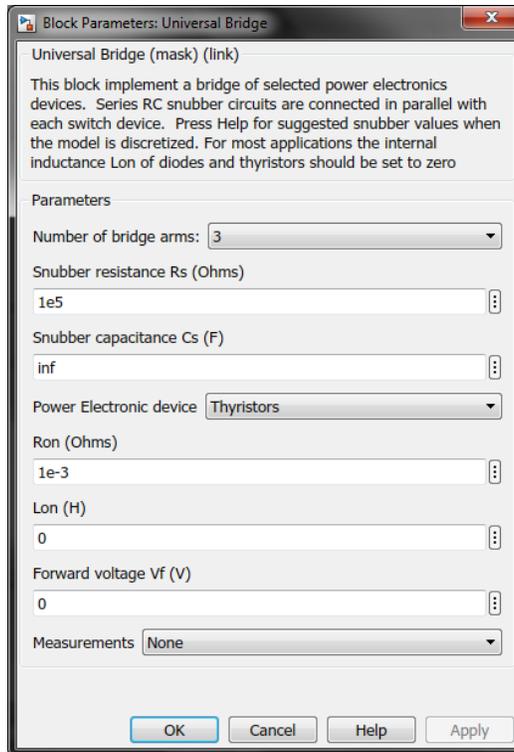


Figura 2.21 Configuración del puente universal.

## Parámetros

- **Número de brazos de puente**

Para un convertidos monofásico se debe selecciona 1 o 2 (dos o cuatro dispositivos de conmutación), si se selecciona 3 corresponderá a un convertidor trifásico conectado en la configuración de puente Graetz (seis dispositivos de conmutación). El valor predeterminado es 3.

- **Resistencia de Snubber Rs**

La resistencia del amortiguador, en ohmios ( $\Omega$ ). El valor predeterminado es 1e5. Si se ajusta el parámetro Rs a inf eso elimina los amortiguadores del modelo.

- **Capacitancia de Snubber Cs**

La capacitancia de amortiguamiento, en faradios (F). El valor predeterminado es inf. Si el parámetro Cs se ajusta en 0 se eliminan los amortiguadores y si se ajusta a inf se obtiene un amortiguador resistivo.

Cuando está utilizando el solucionador continuo, puede eliminar los amortiguadores en todos los dispositivos electrónicos de potencia si selecciona la opción Desactivar los amortiguadores en los dispositivos de conmutación en la pestaña Preferencias del bloque *Powergui*.

Cuando su sistema está discretizado, puede simular dispositivos electrónicos de potencia virtualmente sin amortiguadores al especificar amortiguadores puramente resistivos con una resistencia muy grande, produciendo corrientes de fuga insignificantes. El puente funciona satisfactoriamente con amortiguadores puramente resistivos.

- **Dispositivo electrónico de potencia**

Se especifica el tipo de dispositivo electrónico de potencia para usar en el puente. El predeterminado es tiristores.

Cuando selecciona VSC basado en la función de conmutación, se utiliza un modelo equivalente al tipo de convertidor de fuente de voltaje de conmutación, donde los interruptores se reemplazan por dos fuentes de voltaje en el lado de CA y una fuente de corriente en el lado de CC. Este modelo usa los mismos impulsos de disparo que para otros dispositivos electrónicos de potencia y representa correctamente los armónicos generados normalmente por el puente.

- **Ron**

Resistencia interna del dispositivo seleccionado, en ohmios ( $\Omega$ ). El valor predeterminado es 1e-3.

- **Lon**

Inductancia interna, en henrios (H), para el diodo o el dispositivo tiristor. El valor predeterminado es 0. Cuando el puente está discretizado, el parámetro Lon debe establecerse en cero.

- **Voltaje directo Vf**

Este parámetro está disponible solo cuando el dispositivo electrónico de potencia seleccionado es Diodos o Tiristores.

Voltaje hacia adelante, en voltios (V), a través del dispositivo cuando está conduciendo. El valor predeterminado es 0.

- **Voltajes directos [Dispositivo Vf, Diodo Vfd]**

Este parámetro está disponible cuando el dispositivo electrónico de potencia seleccionado es GTO/Diodos o IGBT/Diodos.

Voltajes directos, en voltios (V), de los dispositivos de conmutación forzada (GTO, MOSFET o IGBT) y de los diodos antiparalelos. El valor predeterminado es [0 0].

- **Mediciones**

El valor predeterminado es Ninguno.

**Voltajes del dispositivo** para medir los voltajes a través de los seis terminales del dispositivo electrónico de potencia.

**Corrientes del dispositivo** para medir las corrientes que fluyen a través de los seis dispositivos electrónicos de potencia. Si se usan diodos antiparalelos, la corriente medida es la corriente total en el dispositivo de conmutación forzada (GTO, MOSFET o IGBT) y en el diodo antiparalelo. Por lo tanto, una corriente positiva indica una corriente que fluye en el dispositivo de conmutación forzada y una corriente negativa indica una corriente que fluye en el diodo. Si se definen dispositivos de amortiguación, las corrientes medidas son las que fluyen a través de los dispositivos electrónicos de potencia solamente.

**Voltajes de UDC UBC UCA UDC** para medir los voltajes de los terminales (CA y CC) del bloque Universal Bridge.

**Todos los voltajes y las corrientes** para medir todos los voltajes y corrientes definidos para el bloque Universal Bridge.

Coloque un bloque de multímetro en su modelo para mostrar las medidas seleccionadas durante la simulación. En el menú Mediciones disponibles del bloque Multímetro, la medición se identifica con una etiqueta seguida del nombre del bloque.

Medición	Etiqueta
Voltaje del dispositivo	Usw:
Corriente de Rama	Isw1:
Voltaje en los terminales	Uab:

Tabla 2.4 Etiquetas de las mediciones en el puente universal.

### Entradas y salidas

- **g**

La entrada de compuerta para los dispositivos de conmutación controlados. El orden de impulsos en el vector de las señales de compuerta corresponde al número de conmutador indicado en los seis circuitos que se muestran en la sección Descripción. Para los puentes de diodos y tiristores, el orden de impulsos corresponde al orden natural de conmutación. Para todos los demás conmutadores de conmutación forzada, los pulsos se envían a los interruptores superior e inferior de las fases A, B y C.

Topología	Vector de Pulsos en la entrada g
Un Brazo	[Q1, Q2]
Dos Brazos	[Q1, Q2, Q3, Q4]
Tres Brazos	[Q1, Q2, Q3, Q4, Q5, Q6]

Tabla 2.5 Vector del pulsos de entrada al puente universal.

### 2.3.5 Powergui

Bloque de entorno para los modelos de tecnología especializada Simscape Power Systems

#### Librería

Simscape/Power System/Specialized Technology/Fundamental Blocks



Figura 2.22 Bloque powergui.

#### Descripción

El bloque powergui le permite elegir uno de estos métodos para resolver su circuito:

- **Continúo**, que usa un solucionador de paso variable de Simulink.
- **Discreto** para una solución en pasos de tiempo fijos
- **Fasores**

El bloque powergui también abre herramientas para el análisis de resultados de simulación y estado estable y para el diseño avanzado de parámetros.

Este bloque es necesario para simular cualquier modelo Simulink que contenga bloques de tecnología especializada Simscape Power Systems. Almacena el circuito Simulink equivalente que representa las ecuaciones de estado del modelo.

El bloque powergui se desactiva durante la actualización del modelo. Para garantizar la ejecución correcta del modelo. Puede usar múltiples bloques de Powergui en un sistema que contenga dos o más circuitos eléctricos independientes que desee simular con diferentes solucionadores de Powergui.

## Parámetros

### Solucionador

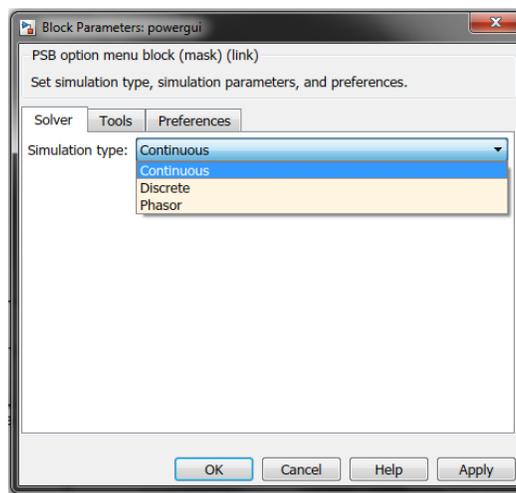


Figura 2.23 Pestaña solucionador del powergui.

La configuración de la pestaña Solver depende de la opción que seleccione de la lista Tipo de simulación.

- **Tipo de simulación**

*Continuo* (predeterminado) para realizar una solución continua del modelo.

*Discreto* para realizar una discretización del modelo. Usted especifica el tiempo de muestreo en el parámetro Tiempo de muestreo.

*Fasor* para realizar la simulación de fasores del modelo, a la frecuencia especificada por el parámetro de frecuencia fasorial.

- **Tiempo de muestreo (s)**

Es el tiempo de muestreo utilizado para discretizar el circuito eléctrico. Este parámetro es visible solo cuando el parámetro Tipo de simulación está configurado como Discreto.

Establezca el parámetro de tiempo de muestreo  $t$  en un valor mayor que 0. El bloque de powergui muestra el valor del tiempo de muestreo, el valor predeterminado es  $50e-6$  s.

- **Frecuencia (Hz)**

Es la frecuencia para realizar la simulación fasorial del modelo. Este parámetro está habilitado solo cuando configura el tipo de simulación en Fasor. El bloque powergui muestra el valor de la frecuencia del fasor el valor predeterminado es 60 Hz.

## Herramientas

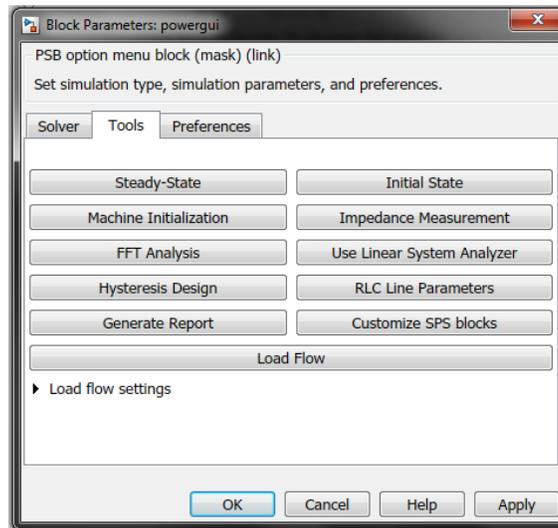


Figura 2.24 Pestaña herramientas del powergui.

- **Estado estable**

Abre el cuadro de diálogo “Herramienta de Corrientes y Tensiones” de estado estable para visualizar los voltajes y corrientes de estado estacionario del modelo.

- **Estado inicial**

Abre el cuadro de diálogo “Herramienta de configuración de estados iniciales” para visualizar y modificar los voltajes iniciales del condensador y las corrientes del inductor del modelo.

- **Inicialización de máquina**

Abre el cuadro de diálogo “Herramienta de inicialización de máquina” para inicializar redes trifásicas que contienen máquinas trifásicas para que la simulación comience en estado estable. La herramienta de inicialización de la máquina ofrece funciones simplificadas de flujo de carga, pero aún puede inicializar las corrientes iniciales de la máquina de sus modelos.

- **Medida de impedancia**

Abre el cuadro de diálogo “Herramienta de medición de impedancia”, para mostrar la impedancia en función de la frecuencia definida por los bloques de medición de impedancia.

- **Análisis de FFT**

Abre el cuadro de diálogo “Herramienta de análisis FFT” para realizar análisis de Fourier de señales almacenadas en una estructura con formato de tiempo.

- **Utilice el analizador de sistema lineal**

Abre una ventana para generar el modelo de espacio de estado de su sistema (si tiene instalado el software Control System Toolbox) y abre la interfaz del “Analizador de sistema lineal” para respuestas de dominio de tiempo y frecuencia.

- **Diseño de histéresis**

Abre una ventana para introducir características de histéresis para el núcleo saturable del bloque transformador y los bloques del transformador trifásico (dos y tres vueltas).

- **Parámetros de línea RLC**

Abra una ventana para calcular los parámetros RLC de una línea aérea de transmisión desde las características del conductor y la geometría de la torre.

- **Generar informe**

Abre el cuadro de diálogo generar informe para generar un informe de variables de estado estable, estados iniciales y flujo de carga de la máquina para un modelo.

- **Personalice los bloques SPS**

Abre power\_customize para crear bloques personalizados de tecnología especializada de Simscape Power Systems.

- **Flujo de carga**

Abra el cuadro de diálogo Load Flow Tool para realizar un flujo de carga e inicializar redes y máquinas trifásicas para que la simulación comience en estado estable.

La herramienta Load Flow usa el método de Newton-Raphson para proporcionar una solución de convergencia robusta y más rápida en comparación con la herramienta de inicialización de la máquina.

- **Iteraciones máximas**

Define el número máximo de iteraciones que la herramienta de flujo de carga itera hasta que la discrepancia de las potencias P y Q en cada bus es menor que el valor del parámetro de tolerancia PQ (en  $PU/Pbase$ ). El desajuste de potencia se define como la diferencia entre la potencia neta inyectada en el bus por generadores y cargas y la potencia transmitida en todos los enlaces que salen de ese bus.

- **Frecuencia (Hz)**

La frecuencia utilizada por la herramienta Flujo de carga para calcular la matriz de admisión de red Ybus normalizada del modelo y para realizar los cálculos de flujo de carga. El valor predeterminado es 60 Hz.

- **Potencia base (VA)**

La potencia base utilizada por la herramienta Load Flow para calcular la matriz de admitancia de la red Ybus normalizada en  $PU/Pbase$  y voltajes base del bus del modelo, a la frecuencia especificada por el parámetro Load flow frequency.

Para evitar una matriz Ybus mal acondicionada, seleccione el valor de potencia base en el rango de potencias nominales y cargas del modelo. Para una red de transmisión con voltajes que van de 120 kV a 765 kV, normalmente se selecciona una base de 100 MVA. Para una red de distribución o para una pequeña planta que consta de generadores, motores y cargas con una potencia nominal en el rango de cientos de kilovatios, se adapta mejor una base de energía de 1 MVA. El valor predeterminado es 100e6 VA.

- **Tolerancia PQ (pu)**

El valor de tolerancia entre P y Q cuando la herramienta de flujo de carga se detiene para iterar. El valor predeterminado es 0.0001.

- **Unidades de voltaje**

Define las unidades de voltaje (V, kV) utilizadas por la herramienta Load Flow para mostrar los voltajes. El valor predeterminado es kV.

- **Unidades de potencia**

Define las unidades de potencia (W, kW, MW) utilizadas por la herramienta Load Flow para mostrar las potencias. El valor predeterminado es MW.

## **Preferencias**

Los parámetros de flujo de carga (Load Flow) son solo para la inicialización del modelo. No tienen un impacto en el rendimiento

- **Deshabilitar las advertencias ST de Simscape Power Systems**

Cuando se selecciona esta casilla de verificación, las advertencias de Simscape Power Systems no se muestran durante el análisis y la simulación del modelo. Por defecto, esta opción no está seleccionada.

- **Mostrar los mensajes de compilación de Simscape Power Systems ST**

Seleccione para habilitar los mensajes de eco de línea de comando durante el análisis del modelo. Por defecto, esta opción no está seleccionada.

- **Utilice el archivo TLC cuando esté en el modo de simulación aceleradora y para la generación de código**

Seleccione para usar las funciones S de estado de TLC en el modo Acelerador y para la generación de código.

Desactive esta casilla si observa una disminución en el rendimiento cuando usa el modo acelerador, en comparación con versiones anteriores. Esta ralentización se produce si tiene instalado el compilador LCC como el compilador predeterminado para construir una interfaz externa. Por defecto, esta opción no está seleccionada.

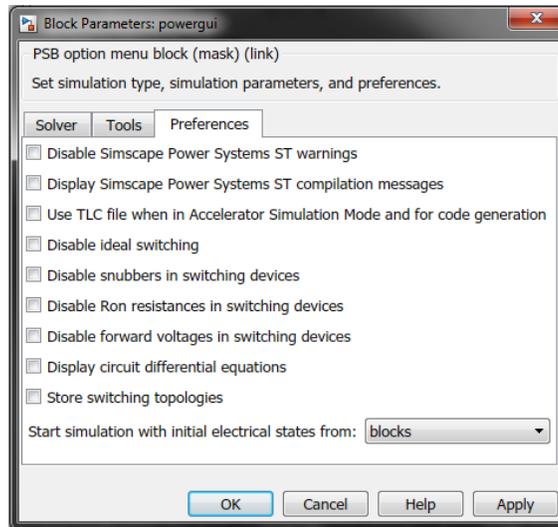


Figura 2.25 Pestaña preferencias del powergui.

- **Deshabilitar conmutación ideal**

Seleccione esta opción para modelar los dispositivos de conmutación como fuentes de corriente. De manera predeterminada, esta opción no está seleccionada, lo que corresponde a la configuración recomendada para la mayoría de sus aplicaciones.

El modelado de interruptores, como interruptores automáticos o dispositivos electrónicos de potencia, como fuentes de corriente implica que la resistencia del interruptor en estado activado Ron no puede ser cero. En este método de modelado, los interruptores no se pueden conectar en serie con un circuito inductivo o con otro interruptor o fuente de corriente.

Cuando esta opción está habilitada, debe agregar un circuito (R o RC snubber) en paralelo con los interruptores en su modelo para que su impedancia de estado de apagado tenga un valor finito. Si su circuito real no utiliza amortiguadores, o si desea simular interruptores ideales sin amortiguador, al menos debe usar amortiguadores resistivos con un alto valor de resistencia para introducir una corriente de fuga insignificante. El inconveniente de la introducción de tales amortiguadores de alta impedancia es que la gran diferencia entre la impedancia del conmutador de estado activado y de estado exterior produce un modelo rígido de espacio de estado.

- **Desactivar amortiguadores en dispositivos de conmutación**

Seleccione para deshabilitar los dispositivos de amortiguación de la electrónica de potencia y los bloques de interruptores en su modelo. Este parámetro está habilitado solo cuando el parámetro Tipo de simulación está configurado en Continuo.

- **Deshabilitar la resistencia de Ron en dispositivos de conmutación**

Seleccione para desactivar la resistencia interna de los interruptores y dispositivos electrónicos de potencia y para forzar el valor a cero ohmios. Este parámetro está habilitado solo cuando el parámetro Tipo de simulación está configurado en Continuo.

- **Deshabilite la tensión directa en los dispositivos de conmutación ( $V_f = 0$ )**

Seleccione para deshabilitar el voltaje de avance interno de los dispositivos electrónicos de potencia y para forzar el valor a cero voltios. Este parámetro está habilitado solo si el parámetro Tipo de simulación está configurado en Continuo.

- **Ecuaciones diferenciales del circuito de pantalla**

Seleccione para mostrar las ecuaciones diferenciales del modelo en el Visor de diagnóstico cuando comience la simulación. Este parámetro está habilitado solo cuando el parámetro Tipo de simulación está configurado en Continuo.

- **Solucionador discreto**

Establezca en Tustin/Backward Euler (TBE) para simular el modelo eléctrico utilizando una combinación de métodos de Tustin y Backward Euler.

Configure a Tustin para discretizar el modelo eléctrico usando el método de Tustin. Si usa este solucionador, debe especificar los valores del amortiguador  $R_s$  y  $C_s$  para evitar oscilaciones numéricas cuando los impulsos de disparo están bloqueados (el puente funciona como un rectificador). En esta condición, debe usar valores apropiados de  $R_s$  y  $C_s$ . Puede usar las siguientes fórmulas para calcular los valores aproximados de  $R_s$  y  $C_s$ :

$$R_s > \frac{2T_s}{C_s} \quad \text{Ecuación 2.20}$$

$$C_s < \frac{P_n}{1000 \times 2\pi f V_n^2} \quad \text{Ecuación 2.21}$$

Dónde

$P_n$  Es la potencia nominal del convertidor monofásico o trifásico, en VA.

$V_n$  Es la tensión CA nominal de línea a línea, en  $V_{rms}$ .

$f$  Es la frecuencia fundamental, en Hz.

$T_s$  Es el tiempo de muestra, en s.

Estos valores se derivan de los siguientes dos criterios:

- La corriente de fuga del amortiguador a la frecuencia fundamental es menor que 0.1% de la corriente nominal cuando los dispositivos electrónicos de potencia no conducen.
- La constante de tiempo RC de los amortiguadores es mayor que dos veces el tiempo de muestra  $T_s$ .

Establezca Backward Euler para discretizar el modelo eléctrico utilizando el método de Euler hacia atrás. El método predeterminado y recomendado es el método Tustin / Backward Euler (TBE). Este parámetro está habilitado solo si configura el parámetro Tipo de simulación en Discreto, como se muestra en la figura 2.26.

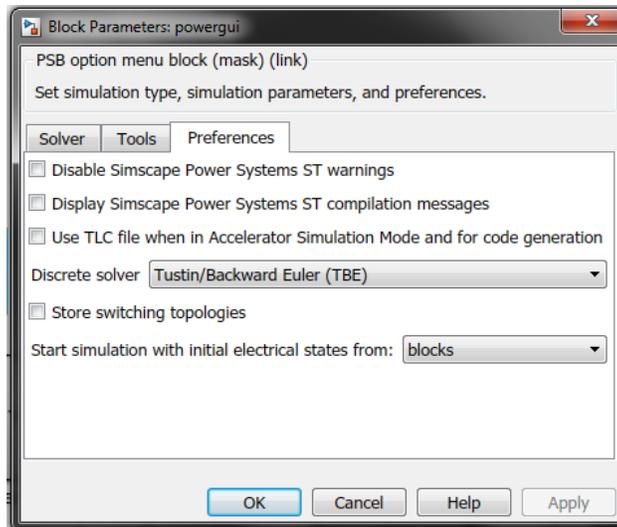


Figura 2.26 Pestaña preferencias del powergui, para tipo de simulación discreto.

### 2.3.6 Generador PWM (2 niveles)

Generador pulsos para un convertidor de 2 niveles controlado por PWM.

#### Librería

Simscape/Power System/Specialized Technology/Control & Measurements/Pulse & Signal Generators

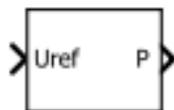


Figura 2.27 Bloque Generador PWM.

#### Descripción

El bloque Generador PWM (2 niveles) genera impulsos para convertidores de modulación de ancho de pulsos (PWM) basados en portadora que utilizan topología de dos niveles. El bloque puede controlar dispositivos de conmutación (FET, GTO o IGBT) de tres tipos diferentes de convertidor: medio puente monofásico (1 brazo), puente completo monofásico (2 brazos) o puente trifásico (3 brazos). La señal de referencia (entrada de Uref), también llamada señal de modulación, se compara con un portador de triángulo simétrico. Cuando la señal de referencia es mayor que la portadora, el pulso para el dispositivo de conmutación superior es alto (1), y el pulso para el dispositivo inferior es bajo (0).

Para controlar un dispositivo de puente completo monofásico, puede seleccionar la modulación PWM unipolar o bipolar. Usando la modulación unipolar, cada brazo se controla de forma independiente. Una segunda señal de referencia se genera internamente mediante el desplazamiento de fase de la señal de referencia original en 180 grados. Usando la modulación bipolar, el estado del dispositivo de conmutación inferior del segundo brazo es el mismo que el del interruptor superior del primer brazo, y el estado del interruptor superior del segundo brazo es el mismo que el estado

del interruptor inferior del primer brazo. La modulación unipolar produce una forma de onda de CA de mejor calidad, pero la modulación bipolar produce una tensión de modo común de muy baja variación.

Las tres técnicas para muestrear la señal de referencia  $U_{ref}$ , son natural, asimétrica y simétrica. La técnica de muestreo natural modela el comportamiento de una implementación analógica de un generador de PWM, como se muestra en la figura 2.28. Usando las dos técnicas de muestreo regular,  $U_{ref}$  puede muestrearse dos veces tanto en el valle como en el pico de la portadora, este se conoce como muestreo asimétrico, como en se muestra en la figura 2.29 o solo una vez en el valle de la portadora este último se llama muestreo simétrico, como en se muestra en la figura 2.30.

### Técnicas de muestreo de señal de referencia

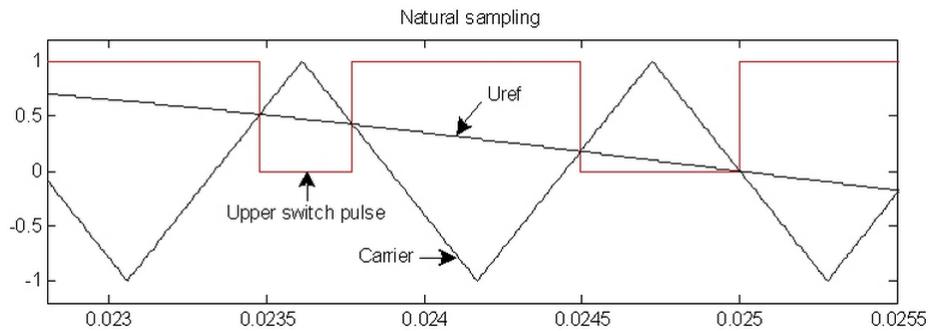


Figura 2.28 Muestreo Natural.

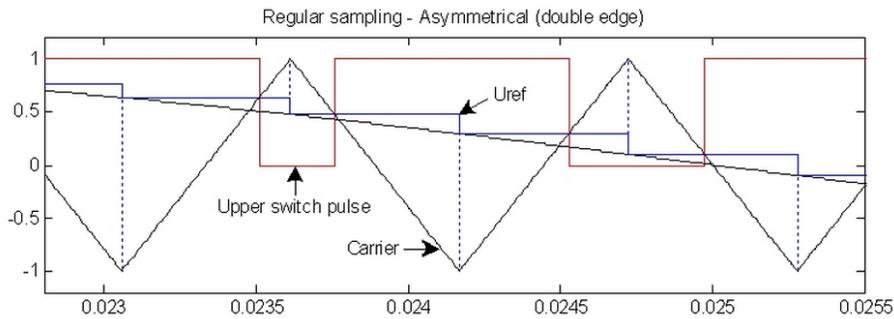


Figura 2.29 Muestreo Asimétrico.

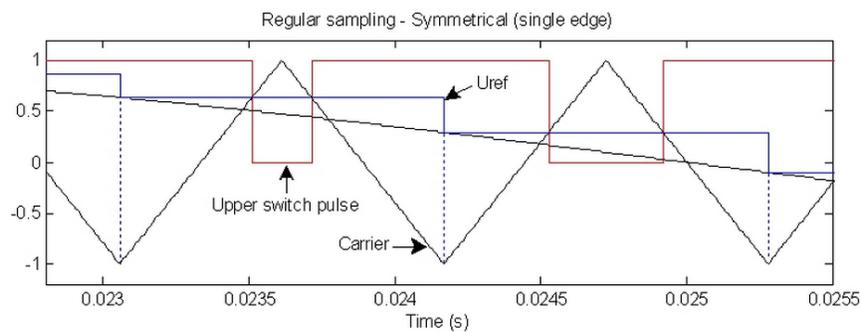


Figura 2.30 Muestreo Simétrico.

## Parámetros

- **Tipo de generador**

La cantidad de pulsos generados por el bloque es proporcional a la cantidad de brazos de puente que se deben disparar.

Half-bridge monofásico (2 pulsos) para disparar los dispositivos autoconmutados de un convertidor de medio puente monofásico. El pulso 1 dispara el dispositivo superior, y el pulso 2 dispara el dispositivo inferior.

Puente monofásico completo (4 pulsos) para disparar los dispositivos autoconmutados de un convertidor monofásico de puente completo. Luego se generan cuatro pulsos. Los pulsos 1 y 3 disparan los dispositivos superiores del primer y segundo brazo. Los pulsos 2 y 4 disparan los dispositivos inferiores.

Puente monofásico completo - Modulación bipolar (4 pulsos) para disparar los dispositivos autoconmutados de un convertidor monofásico de puente completo. Luego se generan cuatro pulsos. Los pulsos 1 y 3 disparan los dispositivos superiores del primer y segundo brazo. Los pulsos 2 y 4 disparan los dispositivos inferiores. Los pulsos 1 y 4 son idénticos. Los pulsos 2 y 3 son idénticos.

Puente trifásico (6 pulsos), este tipo de generador es el predeterminado, para disparar los dispositivos autoconmutados de un convertidor de puente trifásico. Los pulsos 1, 3 y 5 disparan los dispositivos superiores del primer, segundo y tercer brazo. Los pulsos 2, 4 y 6 disparan los dispositivos inferiores.

## Portadora:

- **Modo de operación**

Cuando se establece en No sincronizado (predeterminado), la frecuencia de la señal de la portadora no sincronizada se determina mediante el parámetro Frecuencia.

Cuando se establece en Sincronizado, la señal de la portadora se sincroniza con una señal de referencia externa (entrada *wt*) y la frecuencia portadora se determina mediante el parámetro Relación de conmutación.

- **Frecuencia (Hz)**

La frecuencia, en Hertz, de la señal de portadora triangular. El valor predeterminado es  $27 \times 60$ . Este parámetro está disponible solo si el parámetro Modo de operación está configurado en No sincronizado.

- **Fase inicial (grados)**

La fase inicial de la portadora, en grados. El valor predeterminado es 90. Un valor de 90 grados significa que la posición inicial del transportador triangular se establece en el punto medio entre su valor mínimo y máximo y la pendiente es positiva.

Este parámetro está disponible solo si el parámetro Modo de operación está configurado en Sincronizado.

- **Valores mínimos y máximos**

Los valores mínimo (valle) y máximo (pico) de la señal de portadora triangular. El valor predeterminado es [-1 1].

- **Relación de conmutación (frecuencia portadora / frecuencia de salida)**

La frecuencia ( $F_c$ ) de la señal de portadora triangular.

$$F_c = \frac{\text{Relacion de Cambio}}{\text{Frecuencia de Voltaje de Salida}} \quad \text{Ecuación 2.22}$$

El valor predeterminado es 27. Este parámetro está disponible solo si el parámetro Modo de funcionamiento está configurado como Sincronizado.

**Señal de referencia:**

- **Técnica de muestreo**

Selección la señal de referencia: natural (predeterminado), regular asimétrico (borde doble) o regular simétrico (borde sencillo).

Si selecciona una técnica de muestreo regular, el parámetro de tiempo de la muestra debe ser un submúltiplo entero del período de muestreo. El período de muestreo es igual a  $1 / (F_c / 2)$  para muestreo asimétrico y a  $1 / F_c$  para muestreo simétrico.

- **Generación interna de señal de referencia**

Cuando se selecciona, la señal de referencia es generada por el bloque. El valor predeterminado está despejado.

Cuando no se selecciona, las señales de referencia externas se utilizan para la generación de impulsos. Este parámetro está disponible solo si el parámetro Modo de operación está configurado en Sincronizado.

- **Índice de modulación**

El índice de modulación para controlar la amplitud del componente fundamental de la tensión de salida del convertidor. El índice de modulación debe ser mayor que 0 e inferior o igual a 1. El valor predeterminado es 0.8. Este parámetro está disponible solo cuando se selecciona la generación interna del parámetro de señal de referencia.

- **Frecuencia (Hz)**

La frecuencia de la tensión de salida utilizada para controlar la frecuencia del componente fundamental de la tensión de salida del convertidor. El valor predeterminado es 60. Este parámetro está disponible solo cuando se selecciona la generación interna del parámetro de señales moduladoras.

- **Fase (grados)**

Especifique este parámetro para controlar la fase del componente fundamental de la tensión de salida del convertidor. El valor predeterminado es 0. Este parámetro está disponible solo cuando se selecciona la generación interna del parámetro de señales modulantes.

- **Tiempo de muestreo**

El tiempo de muestreo del bloque, en segundos. Establézcalo en 0 para implementar un bloque continuo. El valor predeterminado es 0. Si selecciona una técnica de muestreo regular, el parámetro de tiempo de la muestra debe ser un submúltiplo entero del período de muestreo. El período de muestreo es igual a  $1 / (F_c / 2)$  para muestreo asimétrico y a  $1 / F_c$  para muestreo simétrico.

- **Mostrar puerto de medición**

Esta casilla de verificación permite agregar una salida de Simulink al bloque. La salida devuelve la señal portadora que se usa para determinar los impulsos de salida y la señal de referencia muestreada. El valor predeterminado está despejado.

### **Entradas y salidas**

#### **Uref**

La señal de referencia utilizada para generar los pulsos de salida. La entrada es visible solo cuando la generación interna de señal de modulación no está seleccionada. Conecte esta entrada a una señal sinusoidal monofásica cuando el bloque se utiliza para controlar un convertidor monofásico de puente medio o completo, o a una señal sinusoidal trifásica cuando el bloque generador PWM está controlando un convertidor puente trifásico. Para el funcionamiento lineal de este bloque, la magnitud de Uref debe estar entre -1 y +1.

#### **wt**

La señal de referencia externa utilizada para sincronizar la portadora. Este puerto de entrada no está presente cuando la portadora está configurado en el modo de operación No sincronizado.

#### **p**

La salida contiene las dos, cuatro o seis señales de pulso usadas para disparar los dispositivos autoconmutados (MOSFET, GTO o IGBT) de un convertidor de uno, dos o tres brazos.

#### **m**

La salida de medición que devuelve la señal de portadora utilizada para determinar los impulsos de salida y la señal de referencia muestreada. Este puerto está presente solo cuando se selecciona Mostrar puerto de medición.

### **Conclusiones del Capítulo 2.**

El modelado de la turbina por ser complejo necesita de un software especializado para poder simularlo, gracias a Matlab y Simulink con sus librerías se puede crear todo tipo de variantes en el modelo del control de la turbina eólica.

Librerías y diagramas de bloques muy específicos de la librería Simscape se necesitan para desarrollar el control, mucha de la implementación provienen de estos diagramas de bloques, así como sus especificaciones de los parámetros de simulación.

Sin estas herramientas de Matlab y Simulink el modelado se podría haber trabajado solo mediante código, pero la simulación y el desarrollo del mismo serían demasiado complicado y llevaría mucho tiempo poder crearlo, es por eso, la utilización de este software, porque viene todo incluido, simulación, modelado, codificación, compilación, depuración y ejecución del sistema de control digital.

**CAPÍTULO 3.**

**IMPLEMENTACIÓN DEL  
MODELO MATEMÁTICO EN  
SIMULINK.**

## CAPÍTULO 3

### IMPLEMENTACIÓN DEL MODELO MATEMÁTICO EN SIMULINK.

En el presente capítulo se mostrara la implementación en Simulink del modelo de la turbina eólica, separando el sistema en los modelos descritos en el capítulo 1 los cuales son:

- Modelo Aerodinámico
- Modelo Mecánico - Eléctrico
- Sistema de control

Modelo aerodinámico: se realizara el cálculo del torque transmitido por el rotor al modelo mecánico - eléctrico aplicando las ecuaciones mostradas en el capítulo 1.

Modelo mecánico - eléctrico: se hará uso de los bloques descritos en el capítulo 2, para la simulación del generador de inducción doblemente alimentado (DFIG), mostrando la configuración y parámetros requeridos para la correcta simulación del sistema.

Sistema de control: incluye la implementación de las transformadas presentadas en el capítulo 1, para el cambio de sistema de referencia  $abc$  al  $dq$  y la aplicación de los controladores de velocidad y corriente para el control vectorial.

### 3.1 Elementos del Sistema

Para la implementación en Simulink se consideraran las partes mostradas en el capítulo 1, por lo que es necesario la creación de subsistemas para separarlos, las partes mencionadas son.

- Modelo Aerodinámico.
- Modelo Mecánico-Eléctrico.
- Sistema de Control.

Además se colocara un subsistema para monitorear el comportamiento de la velocidad y el torque de la máquina, en la figura 3.1 se muestran los subsistemas antes mencionados.

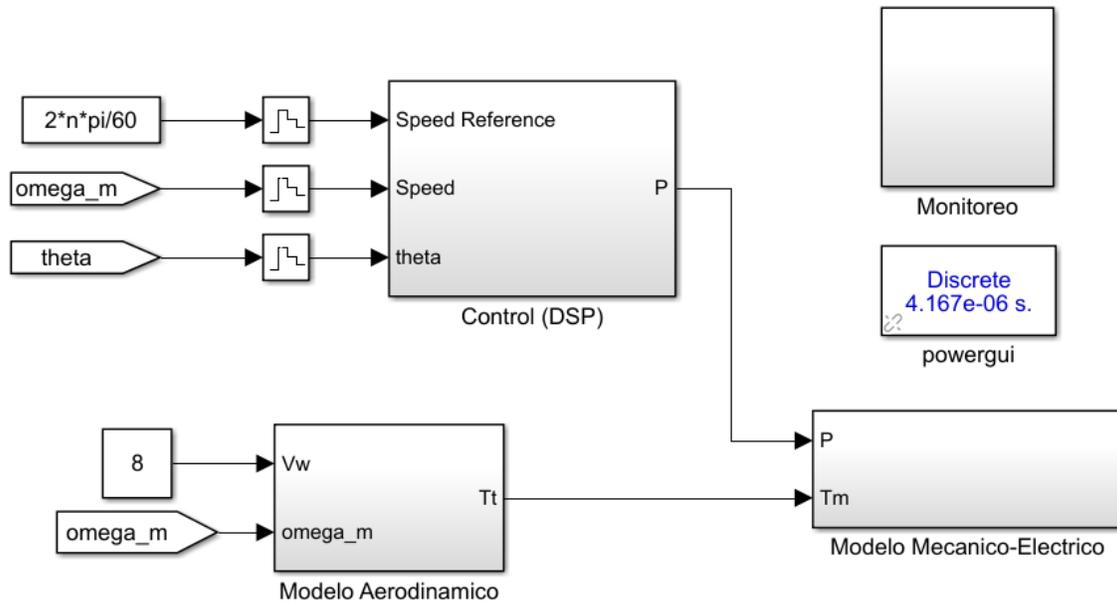


Figura 3.1 Implementación del sistema en Simulink.

### 3.2 Modelo Aerodinámico.

El modelo aerodinámico como se mencionó en el capítulo 1, considera la conversión de la energía cinética del viento, en un torque mecánico, es por ello que las entradas para este modelo son la velocidad del viento y la velocidad del rotor, la salida es el torque transmitido al eje de baja velocidad por rotor, como se muestra en la figura 3.2.

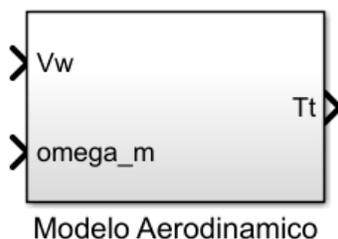


Figura 3.2 Entradas y salida del modelo aerodinámico.

El ángulo de inclinación de las palas con el que se obtiene el coeficiente de potencia máximo, es cuando este es de  $0^\circ$  con respecto a la dirección de la velocidad del viento, como se observa en la

figura 1.12 del capítulo 1, es por ello que no se considera como entrada. El interior del bloque mostrado en la figura 3.2, se muestra en la figura 3.3, donde podemos observar el cálculo de la velocidad en la punta de la pala  $\lambda$  lambda, el coeficiente de torsión  $C_t$  y luego el torque del rotor  $T_r$ .

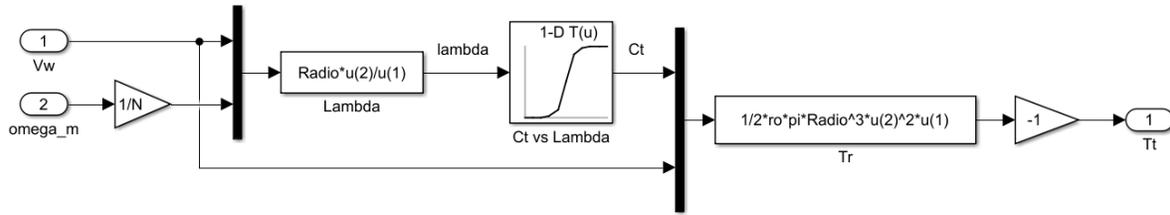


Figura 3.3 Implementación del Modelo Aerodinámico.

Los bloques de ganancia se han utilizado para realizar la conversión del tren de impulsión, para el cálculo de  $\lambda$  y  $T_r$  se ha utilizado el bloque “Fcn” de la librería “User-Defined Funtions” y se colocan las expresiones correspondientes a partir de las entradas.

El cálculo de la velocidad en la punta de la pala está definido por la ecuación 1.16:

$$\lambda = \frac{R\omega_r}{V_w}$$

Por ello la introducción de la función a al bloque “Fcn” se muestra en la figura 3.4.

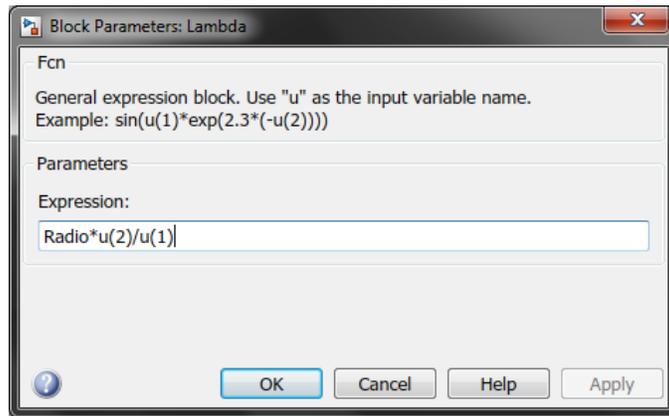


Figura 3.4 Función para el cálculo de lambda.

Para el cálculo del coeficiente de torsión  $C_t$  se implementa un lazo que calcula de manera iterativa el valor de este, para los diferentes valores de  $\lambda$ , dicho lazo se encuentra en el script mostrado en el Anexo 1. En dicho script se crea la variable  $C_t$  (vector) la cual introducimos en el bloque “1-D Lookup Table” de la librería “Lookup Table”. Como se muestra en la figura 3.5.

El torque producido en el rotor se define por medio de la ecuación 1.17.

$$T_r = \frac{1}{2} \rho \pi R^3 V_w^2 C_t$$

Por ello la introducción de la función a al bloque se muestra en la figura 3.6.

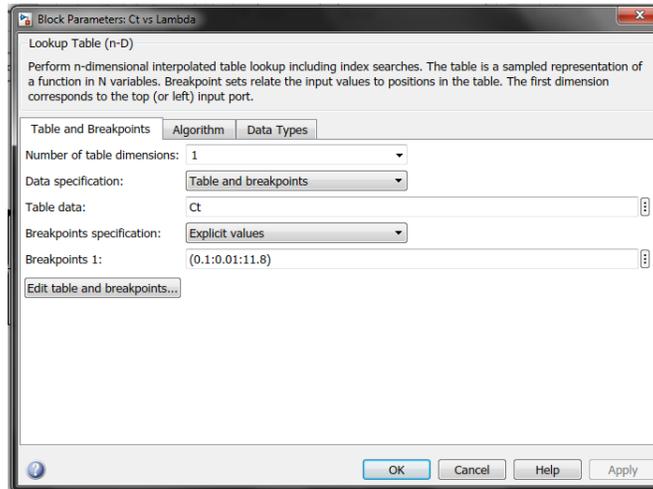


Figura 3.5 Cálculo de la variable  $C_t$ .

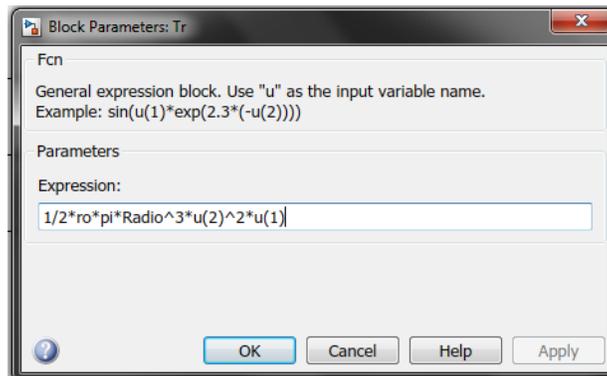


Figura 3.6 Función para el cálculo del Par de salida.

El bloque de ganancia ubicado entre el cálculo del par y la salida en la figura 3.3, con valor igual a -1, realiza un cambio de signo al torque, ya que como se explicó en el capítulo 2, para que la maquina asíncrona opere como generador, el torque debe ser negativo.

### 3.3 Modelo Mecánico - Eléctrico.

El modelo mecánico eléctrico, como se mencionó en el capítulo 1, lo conforman el tren de impulsión y el generador, el tren de impulsión comprende los ejes de baja y alta velocidad, así como la caja de engranes, que es la encargada de elevar la velocidad, esta es considerada la parte mecánica, la parte eléctrica estará formada por el generador, que en este caso se trata de una maquina asíncrona de rotor bobinado.

Para el tren de impulsión se considera un bloque de ganancia, de valor igual a la relación de engranes y la conexión de la maquina asíncrona doblemente alimentada se muestra en la figura 1.15, para la implementación en Simulink solo se considerara el convertidor del rotor y se omitirá el control del lado de la red, lo anteriormente descrito se muestra en la figura 3.7.

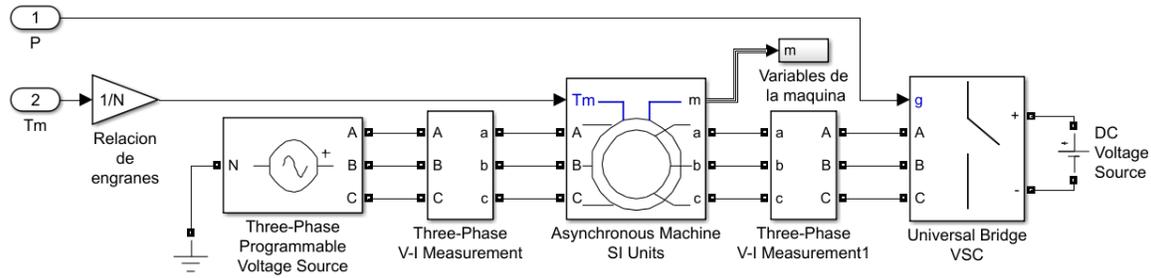


Figura 3.7 Implementación del Modelo Mecánico-Eléctrico.

Los elementos que están situados a la izquierda del bloque “Asynchronous Machine SI Units” son los elementos que se conectan al estator y representa a la Red y los que están situados a la derecha son los elementos que se conectan al rotor y representan el VSC “Voltage Source Converters”. Cada uno de estos elementos se explica en detalle en el capítulo 2, por lo que a continuación solo se presentaran los parámetros que se configuraron en cada uno de ellos.

Iniciando con la fuente de tensión trifásica programable la cual se configura para que suministre una tensión  $V_s$ , a frecuencia  $f$  estos parámetros han sido definidos en el script del anexo 1. Este bloque representa la Red eléctrica en las figuras 3.8 y 3.9 se muestra la configuración de este bloque.

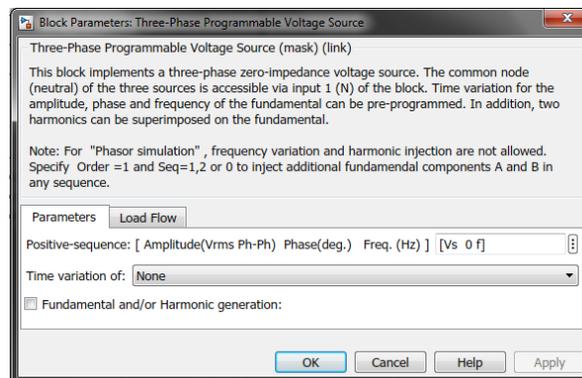


Figura 3.8 Parámetros de la fuente de tensión trifásica programable.

En el caso del tipo de generador se selecciona uno oscilante (“Swing”).

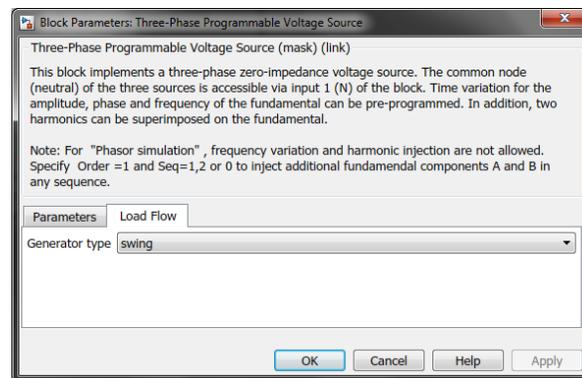


Figura 3.9 Tipo de generador de la fuente de tensión trifásica programable.

A continuación del bloque que representa a la Red eléctrica, se coloca un bloque para la medición de tensión y corriente trifásica, esto debido a que estas variables se utilizarán más adelante, por lo que es necesario conocer su magnitud. Del lado del estator se medirán tanto la tensión como la corriente en la figura 3.10 se muestra la configuración del bloque.

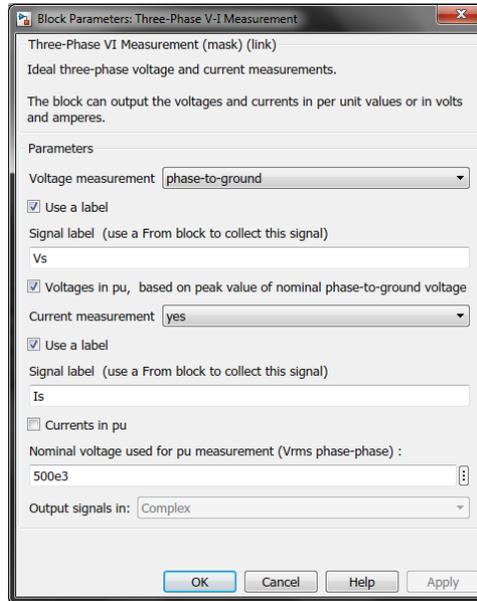


Figura 3.10 Configuración del bloque de medición del lado del estator.

A la salida de la medición se encuentra conectada el bloque denominado “Asynchronous Machine SI Units” el cual representa a la maquina asíncrona o de inducción, para este bloque como se muestra en el capítulo 2 es necesario la configuración de 4 pestañas.

En la primera se configura el tipo de rotor, que en nuestro caso es un rotor bobinado, el tipo de entrada la cual será el torque que es generado en el rotor de la turbina.

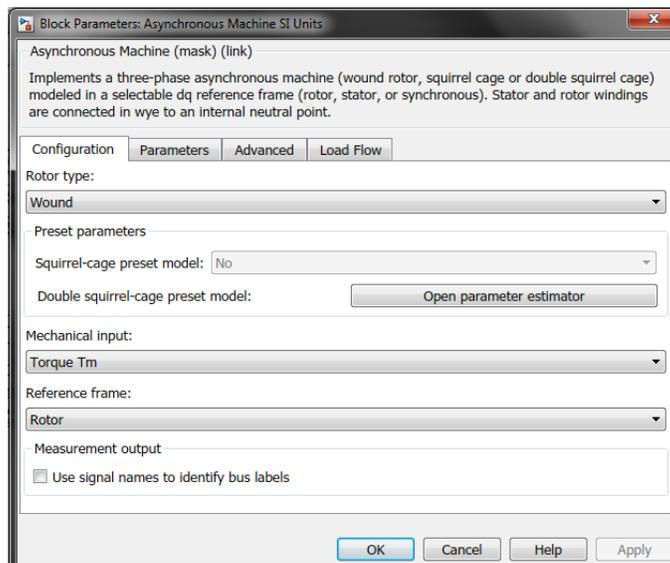


Figura 3.11 Configuración de la maquina Asíncrona.

En la segunda se introducen los parámetros de la maquina los cuales pueden ser obtenidos de los datos de placa o de ensayos realizados a esta, en la figura 3.12 se muestra la introducción de dichos parámetros al bloque, los cuales son variable definidas en el script del anexo 1.

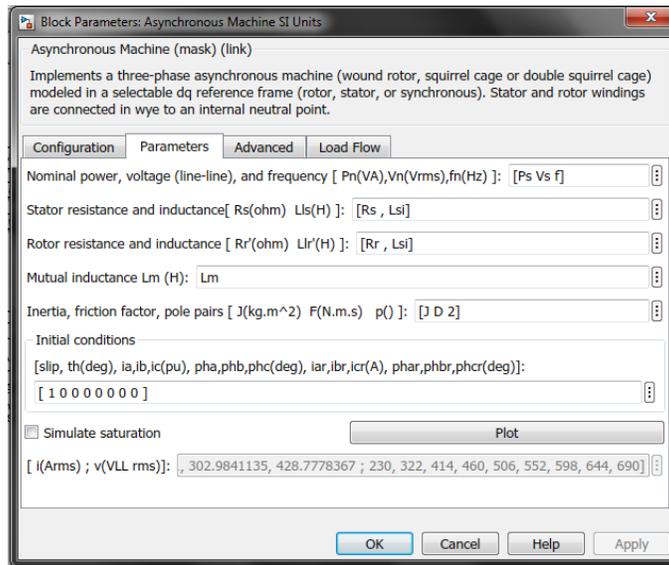


Figura 3.12 Parámetros de la Maquina Asíncrona.

En las opciones avanzadas se configura el solucionador discreto a implementar y el tiempo de muestreo que en este caso se coloca -1 para que sea heredado, es decir para que trabaje con el mismo tiempo al que trabaja el sistema.

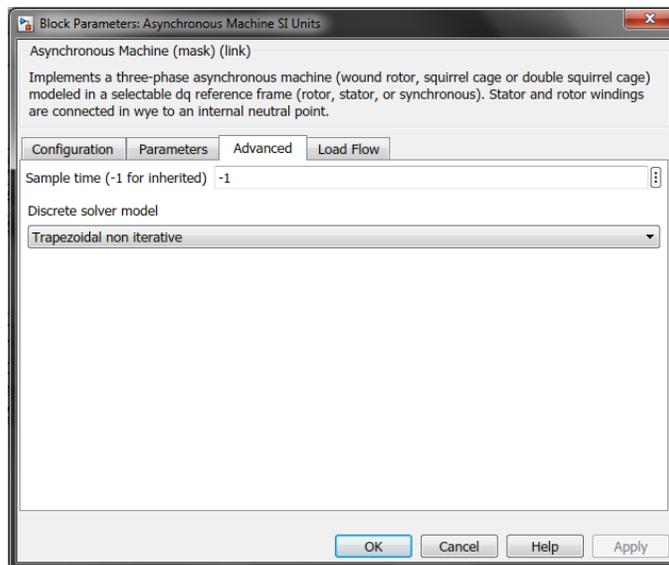


Figura 3.13 Opciones Avanzadas de la Maquina Asíncrona.

En la última pestaña que corresponde al flujo de carga se debe de colocar la potencia mecánica de la máquina que en este caso se dejara el valor que trae por defecto el bloque.

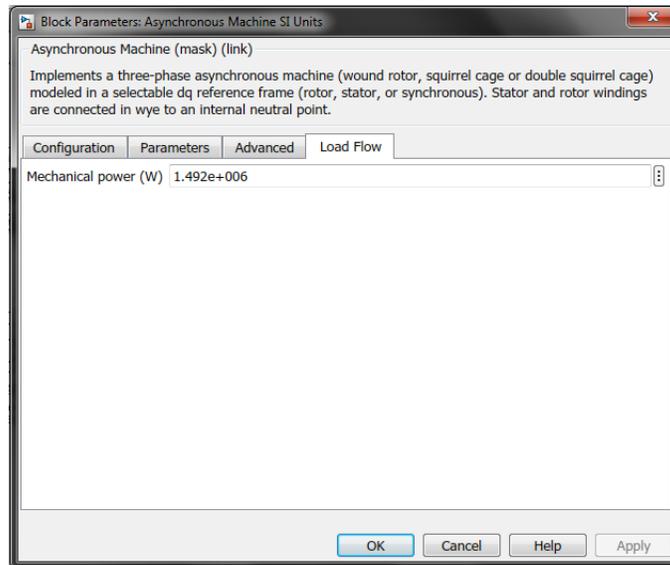


Figura 3.14 Potencia mecánica de la maquina Asíncrona.

En la entrada  $T_m$  del bloque de la maquina asíncrona, se conectara el torque de salida generado en el rotor (modelo aerodinámico) y en la salida  $m$  es un vector que contiene diferentes variables mecánicas y eléctricas de la máquina, (ver el capítulo 2 para conocer todas las variables), las variables de interés son las mecánicas como velocidad del rotor, toque electromecánico y la posición angular del rotor las cuales se seleccionan por medio de un bus selector como se muestra en la figura 3.15.

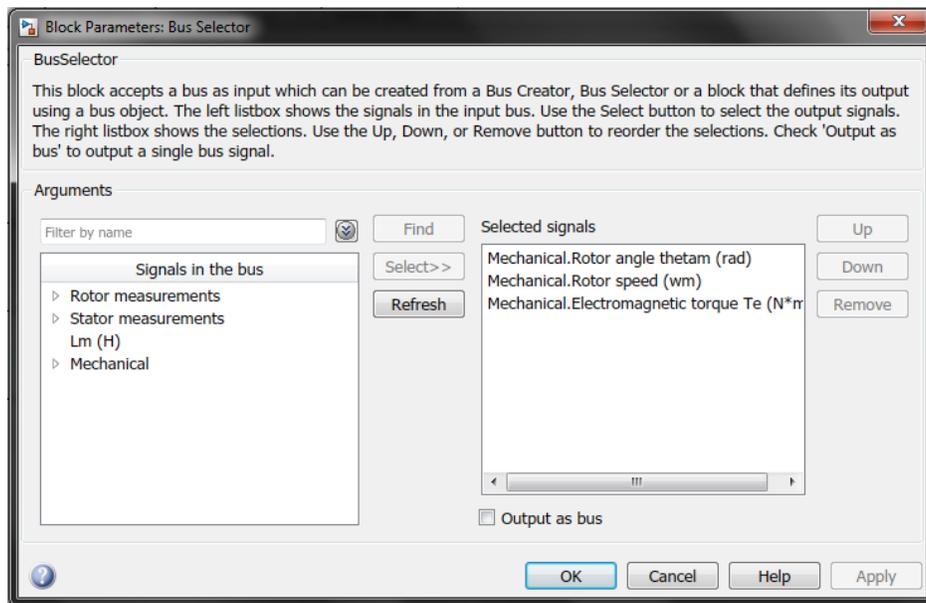


Figura 3.15 Selección de señales en el bus selector.

El bus selector se encuentra en el bloque denominado “Variables de la maquina” de la figura 3.7, además se ocupan tres bloques Goto, como se muestra en la figura 3.16, los bloques Goto permiten que las variables medidas puedan ser utilizadas en otra parte del sistema utilizando bloques From.

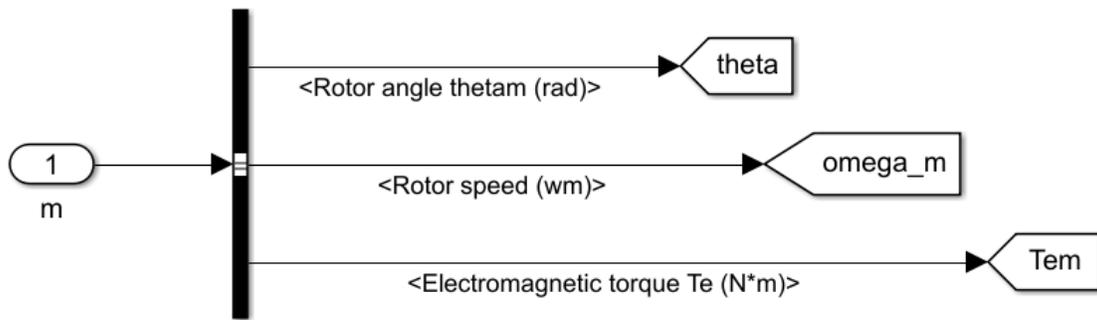


Figura 3.16 Diagrama para la selección de señales de la maquina asíncrona.

Al lado derecho y conectado a la maquina asíncrona encontramos, la medición en el lado del rotor, este bloque al igual que la medición en el estator, sirve para medir la magnitud del voltaje y de la corriente, solo que en este caso solo nos interesa conocer la corriente del rotor, por lo que la medición de voltaje se inhabilitara, en la figura 3.17 se muestran la configuración del bloque de medición.

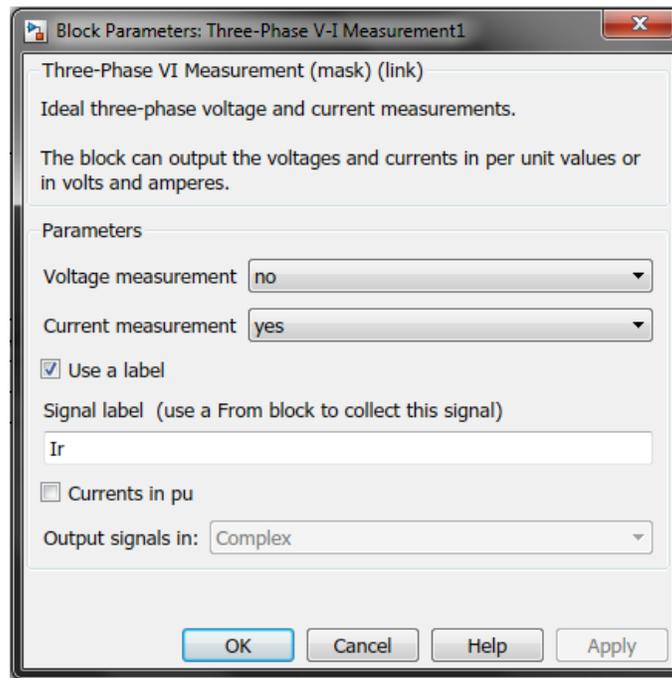


Figura 3.17 Configuración del bloque de medición del lado del rotor.

Del lado del rotor como se muestra en la figura 3.7, se utiliza un puente universal, que representa una VSC "Voltage Source Converter" diseñada con elementos de electrónica de potencia, el cual se encarga de generar una señal de alterna que alimentara el rotor, a partir de una fuente de DC y una señal que será generada por la etapa de control. En este caso el puente es de 3 brazos dado que el sistema es trifásico y el elemento de potencia a utilizar son switches ideales, la configuración de este bloque se muestra en la figura 3.18.

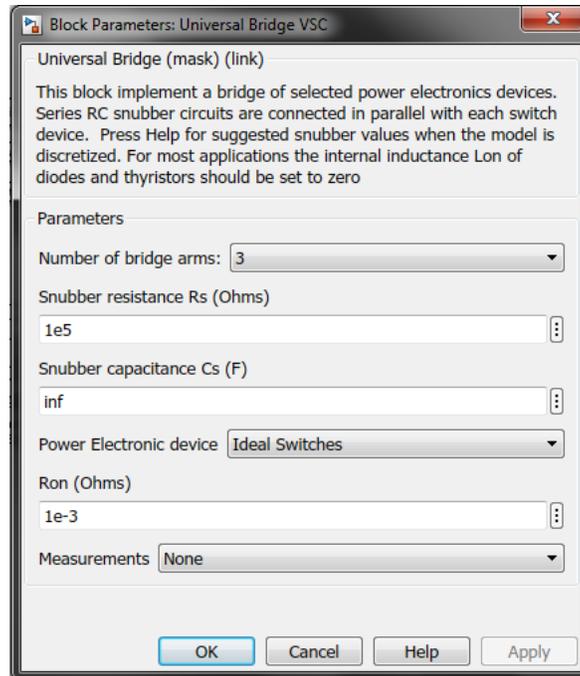


Figura 3.18 Configuración del puente universal.

Por ultimo encontramos la fuente de DC, que será la encargada de proporcionar la segunda alimentación de la máquina, este bloque permite reemplaza el convertidor del lado de la red, que se encarga de proporcionar la señal de DC al convertidor del lado del rotor, esta es una simplificación que permite realizar Simulink, el valor de la fuente se muestra en el script del anexo 1.

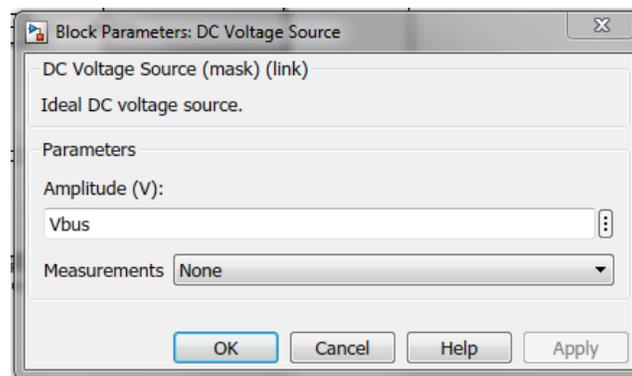


Figura 3.19 Configuración de la Fuente de Voltaje de DC.

### 3.4 Sistema de Control.

La etapa de control, es la más importante ya que esta será la encargada del regular el correcto funcionamiento de la máquina, para el desarrollo de este se tomara como base el diagrama mostrado en la figura 1.18 y simplificado para facilitar la implementación.

#### 3.4.1 Transformación de señales medidas.

Como se mencionó anteriormente es necesario medir variables mecánicas y eléctricas de la máquina, y realizar transformaciones para poder ajustar el comportamiento de la misma. En la figura 3.20 se muestra la implementación en Simulink del diagrama de control.

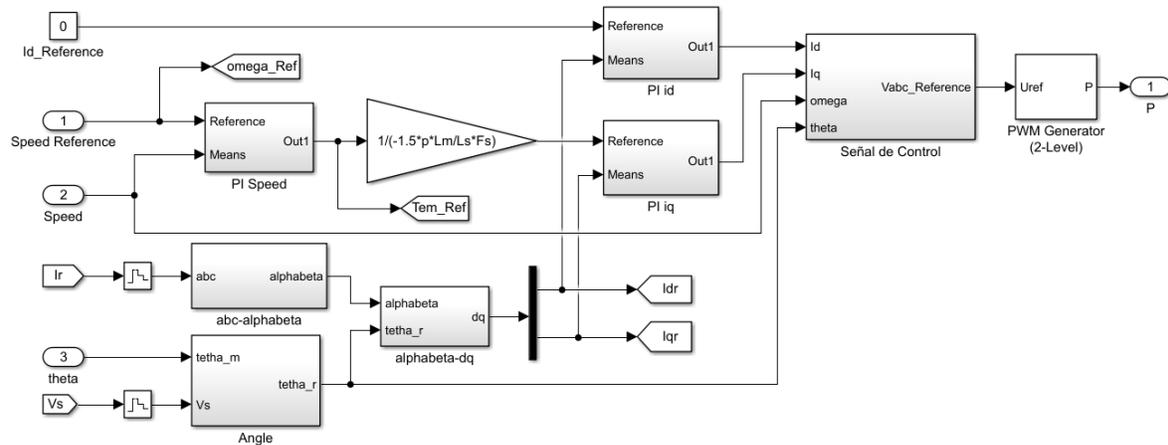


Figura 3.20 Diagrama de control implementado en Simulink.

Los bloques 1/u de la figura 1.18 sirven para referir las variables del rotor al estator, pero en este caso no se colocaran dado que el bloque que se está utilizando de la maquina asincrona refiere las variables del rotor al estator como se mencionó en el capítulo 2.

Primero se realizara el cálculo de la posición angular de referencia  $\theta_r$ , a partir de la posición angular de la máquina y la tensión en el estator, la posición angular de la tensión se obtiene, por medio de la transformada definida en el bloque  $abc-\alpha\beta$ , la cual permite cambiar un sistema de referencia  $abc$  a un sistema de referencia  $\alpha\beta\gamma$ , dicha transformada es conocida como la transformada de Clarke, como se muestra en la figura 3.21.

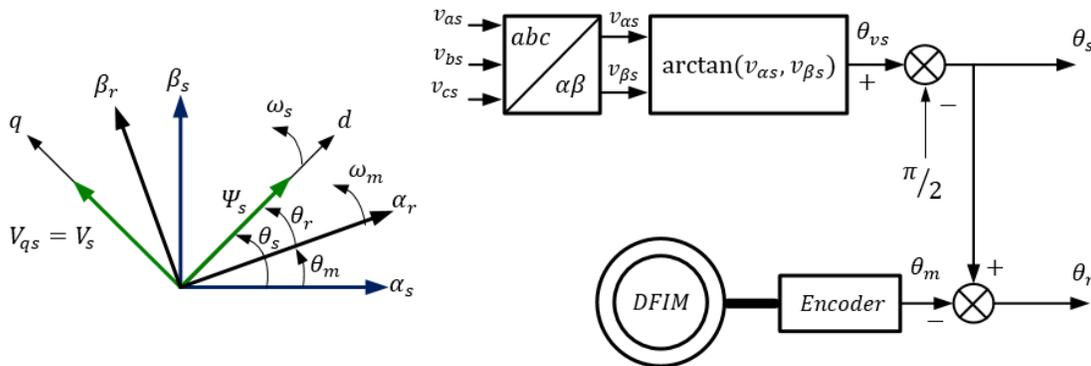


Figura 3.21 Cálculo de ángulo de referencia  $\theta_r$ .

La implementación en Simulink del diagrama anterior se muestra en la figura 3.22.

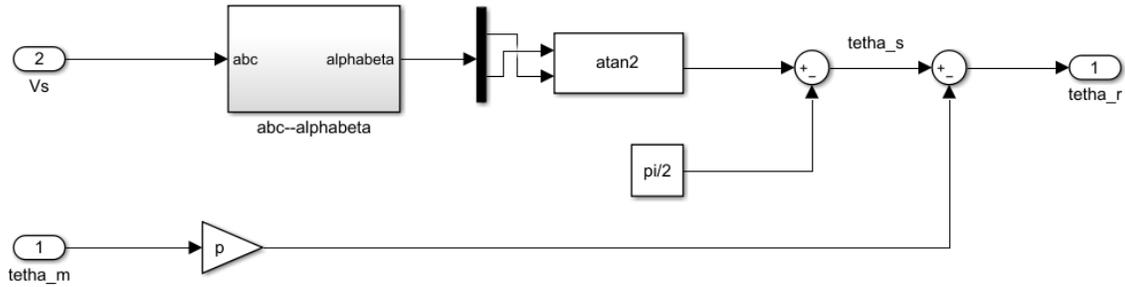


Figura 3.22 Implementación en Simulink del cálculo de ángulo de referencia  $\theta_r$ .

La transformada del bloque  $abc-\alpha\beta$  está definida para la ecuación 1.30, la cual que se presentan a continuación.

$$\begin{bmatrix} v_\alpha \\ v_\beta \end{bmatrix} = \frac{2}{3} \begin{bmatrix} 1 & -\frac{1}{2} & -\frac{1}{2} \\ 0 & \frac{\sqrt{3}}{2} & -\frac{\sqrt{3}}{2} \end{bmatrix} \begin{bmatrix} v_a \\ v_b \\ v_c \end{bmatrix}$$

Para su implementación en Simulink se utiliza el bloque “Fnc” que permite definir ecuaciones a partir de sus entradas, como se muestra en la figura 3.23.

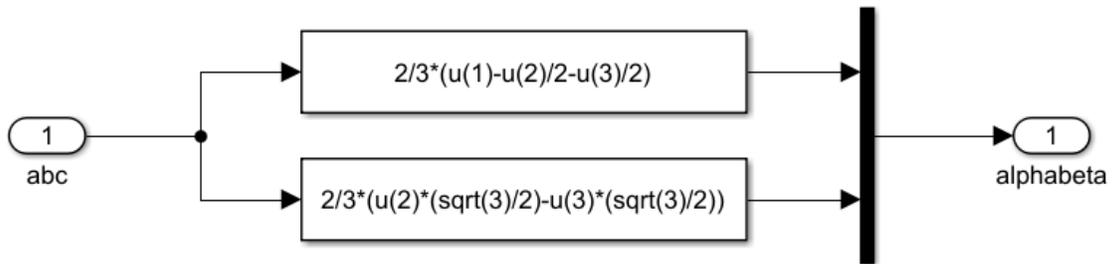


Figura 3.23 Implementación de la transformada  $abc-\alpha\beta$ .

Para obtener la señal medida de  $i_d$  e  $i_q$  será necesario realizar una transformación a partir de las corrientes medidas del rotor  $abc$ , primeramente al sistema  $\alpha\beta\gamma$  y luego haciendo uso del ángulo  $\theta_r$ , se realizara la conversión al sistema  $dqo$ . A esta transformación se le conoce como transformada de Park. Para la primera transformación se utiliza un bloque similar al mostrado en la figura 3.23, ya que la transformada de Clark aplica para la corriente y la tensión como se explica en el capítulo 1.

Para realizar la transformación  $\alpha\beta\gamma-dqo$  mencionada anteriormente se realizara utilizando la ecuación 1.32 mostrada en el capítulo 1.

$$\begin{bmatrix} v_d \\ v_q \\ v_o \end{bmatrix} = \begin{bmatrix} \cos \theta & \sin \theta & 0 \\ -\sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} v_\alpha \\ v_\beta \\ v_\gamma \end{bmatrix}$$

La implementación en Simulink se realiza de la misma manera que la transformación  $abc-\alpha\beta$  haciendo uso de los bloques "Fnc" y se muestra en la figura 3.24.

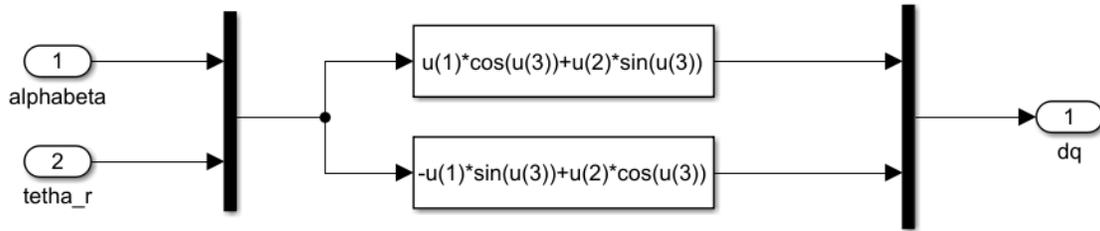


Figura 3.24 Implementación de la transformada  $\alpha\beta\text{--}dq$ .

Después de realizar estas transformaciones se obtienen los valores medidos  $i_d$  e  $i_q$  por lo que ahora es necesario establecer los valores de referencia. Los cuales como se observa en la figura 1.18, Se obtiene a partir del control de velocidad.

### 3.4.2 Control de velocidad.

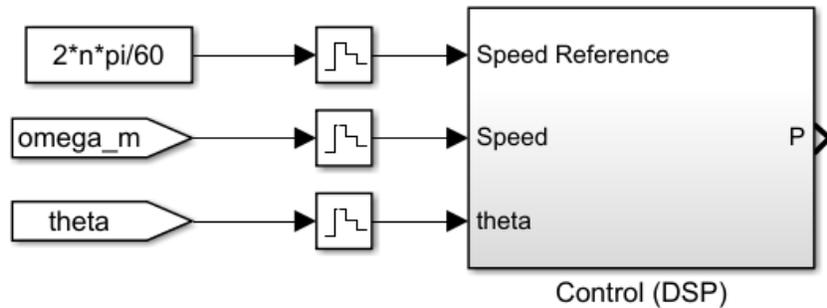


Figura 3.25 Entradas del sistema de control.

Para el control de velocidad se necesitan la velocidad de referencia y la velocidad medida de la maquina asíncona, la velocidad de referencia se establece en base a las características de la máquina y se calcula a partir de la siguiente ecuación.

$$n = \frac{60 \times f}{p} \text{ rpm} \quad \text{Ecuación 3.1}$$

Dado que la frecuencia del sistemas es de 60 Hz y los polos de la maquina son 2.

$$n = \frac{60 \times 60}{2} = \frac{3600}{2} = 1800 \text{ rpm}$$

Como podemos ver en la figura 3.25 la velocidad de referencia debe convertirse de  $rpm$  a  $m/s$ , por eso se utiliza el factor de  $2\pi/60$  y  $n$  es definida en el script, la señal  $omega\_m$  es la velocidad medida de la máquina.

Si realizamos un acercamiento en el diagrama de la figura 1.18, podemos observar el control de velocidad y la obtención de la señal  $i_q$  de referencia, como se muestra en la figura 3.26.

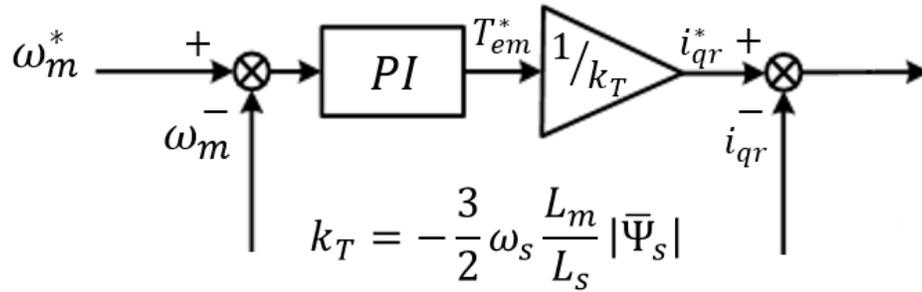


Figura 3.26 Control de velocidad y obtención de  $i_q$  de referencia.

La implementación en Simulink de lo anterior se muestra en la figura 3.27.

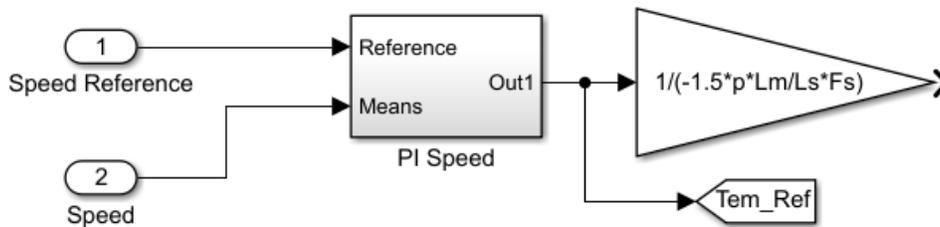


Figura 3.27 Implementación en Simulink del control de velocidad.

La implementación del controlador PI de velocidad se realizara utilizando un bloque de ganancia para la parte proporcional y un bloque de ganancia conectado en serie con un integrador de tiempo discreto, para la parte integradora, lo anterior se muestra en la figura 3.28.

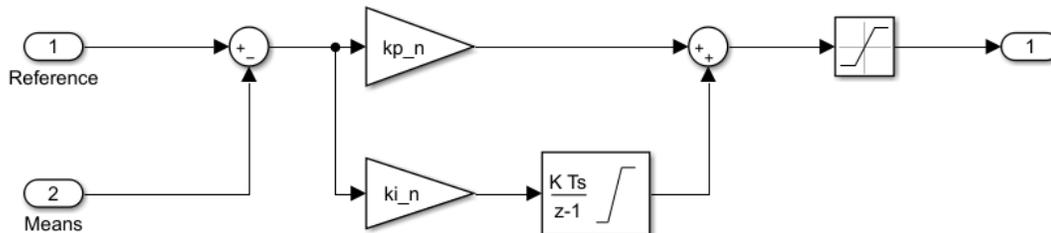


Figura 3.28 Controlador PI de velocidad.

Las variables  $kp_n$  y  $ki_n$  están definidas por medio de las siguientes ecuaciones.

$$kp_n = \frac{2\omega_n J}{p} \quad \text{Ecuación 3.2}$$

$$ki_n = \frac{\omega_n^2 J}{p} \quad \text{Ecuación 3.3}$$

Dichas ecuaciones han sido implementadas en el script y la configuración del bloque de saturación y del integrador se muestra en las figuras 3.29 y 3.30.

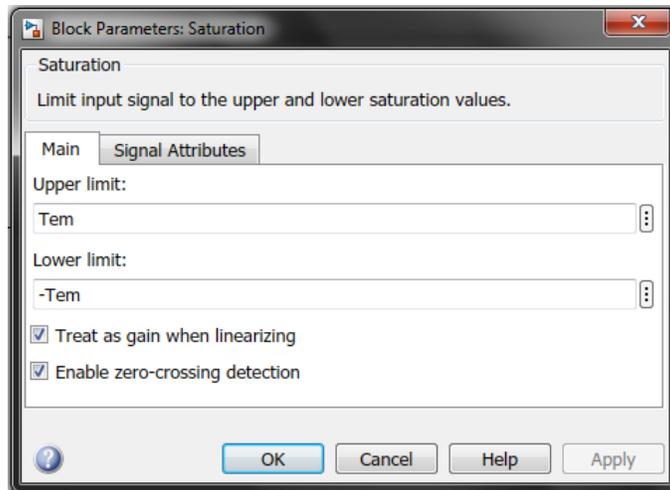


Figura 3.29 Configuración del bloque de saturación en el control de velocidad.

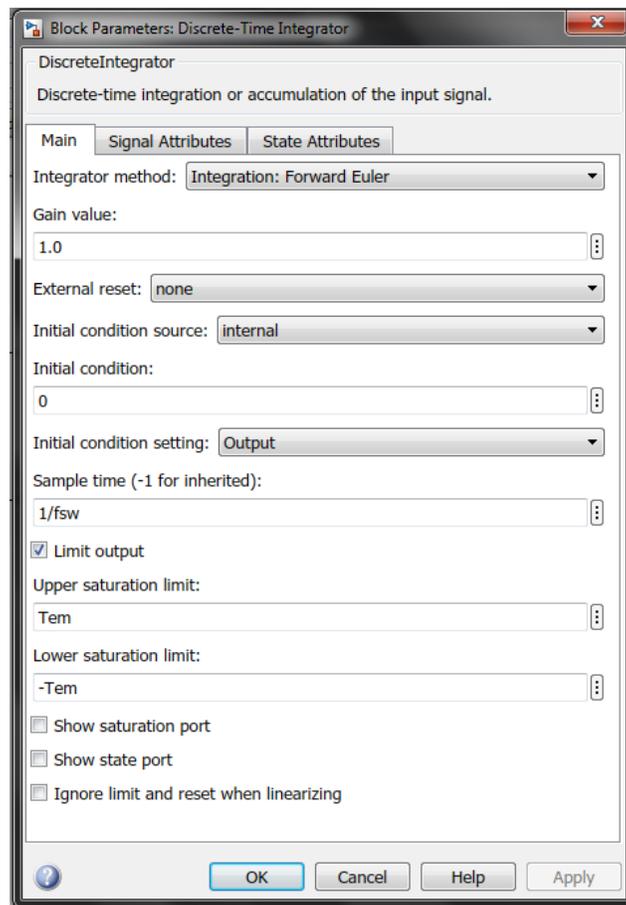


Figura 3.30 Configuración del integrador en el controlador PI de velocidad.

El bloque de saturación se coloca para evitar que el torque de referencia sobrepase los límites de la máquina.

A la salida de este bloque se obtiene el torque producido por la velocidad del rotor y al multiplicarlo por la ganancia  $1/k_r$ , se obtiene la señal  $i_q$  de referencia, como se muestra en la figura 3.26. Una vez obtenidas las señales de referencia y las medidas, se desarrollan los controladores PI para la corriente  $i_d$  e  $i_q$ .

La topología de los controladores PI para las corrientes  $i_d$  e  $i_q$  es similar a la utilizada en el controlador de velocidad, la diferencia será los valores de ganancia a utilizar. En el caso del controlador para la corriente  $i_d$  la ganancia proporcional será  $kp_{id}$  y para la parte integradora  $ki_{id}$ , como se muestra en la figura 3.31.

Las variables  $kp_n$  y  $ki_n$  están definidas por medio de las variables de la maquina en las siguientes ecuaciones.

$$kp_i = 2\omega_n\sigma L_r - R_r \quad \text{Ecuación 3.4}$$

$$ki_i = \omega_n^2\sigma L_r \quad \text{Ecuación 3.5}$$

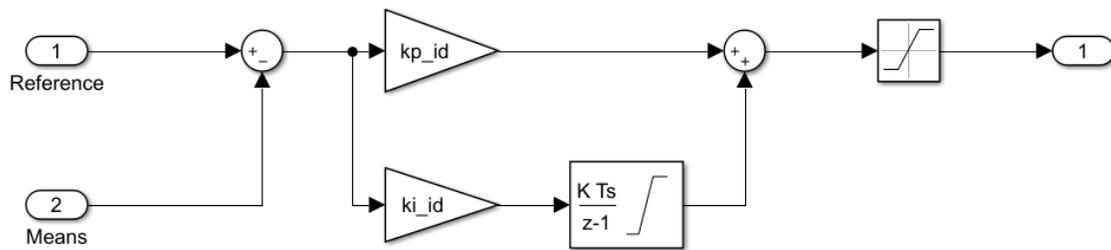


Figura 3.31 Controlador PI para la corriente  $i_d$ .

Para la corriente  $i_q$  la ganancia proporcional será  $kp_{iq}$  y para la parte integradora  $ki_{iq}$ , como se muestra en la figura 3.32.

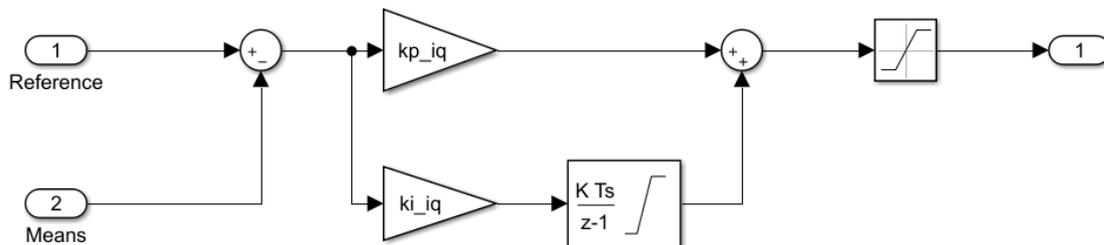


Figura 3.32 Controlador PI para la corriente  $i_q$ .

Los límites del bloque de saturación son los mismos que se utilizan en el integrador como lo podemos ver en la figura 3.33. Para ambos controladores la configuración del integrador será la misma, en la figura 3.34 se muestran la condición inicial, el tiempo de muestreo y los límites superior e inferior de la señal.

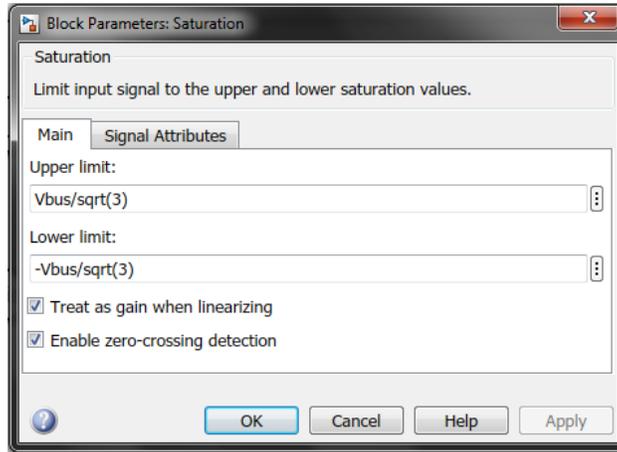


Figura 3.33 Configuración del bloque de saturación para los controladores de corriente.

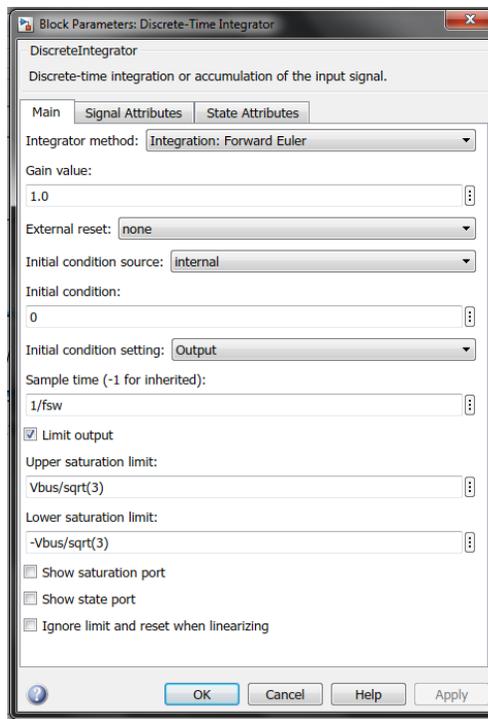


Figura 3.34 Configuración del Integrador para los controladores de corriente.

La ecuación del rotor de la máquina separada en sus dos componentes  $dq$  se representa por las ecuaciones.

$$v_{dr} = R_r i_{dr} - (\omega_s - \omega_m) \Psi_{qr} + \frac{d}{dt} \Psi_{dr} \quad \text{Ecuación 3.6}$$

$$v_{qr} = R_r i_{qr} - (\omega_s - \omega_m) \Psi_{dr} + \frac{d}{dt} \Psi_{qr} \quad \text{Ecuación 3.7}$$

A la diferencia entre la velocidad de sincronismo  $\omega_s$  y la velocidad del rotor de la maquina  $\omega_m$  se le denominara velocidad de deslizamiento  $\omega_r$ .

Si la maquina trabaja sin saturarse, el flujo del estator será la suma del flujo producido por la corriente circulante por el mismo, más una parte del producido por las corrientes del rotor. De la misma manera ocurre en el rotor, las siguientes ecuaciones describen el comportamiento anterior.

$$\Psi_s = L_s i_s + L_m i_r \quad \text{Ecuación 3.8}$$

$$\Psi_r = L_r i_r + L_m i_s \quad \text{Ecuación 3.9}$$

Donde,  $L_s, L_r$  y  $L_m$  son las inductancias de los enrollados de estator y rotor y la inductancia magnetizante respectivamente. Despejando  $i_s$  en ambas, igualando y operando, se obtiene la relación entre  $\Psi_s$  y  $\Psi_r$ , representada por la ecuación.

$$\Psi_r = \left( L_r - \frac{L_m^2}{L_s} \right) i_r + \frac{L_m}{L_s} \Psi_s \quad \text{Ecuación 3.10}$$

Separando la expresión anterior en sus componentes  $d$  y  $q$ , orientando el eje  $d$  según el flujo del estator, la componente en el eje  $q$  es nula, y por consiguiente  $\Psi_{ds} = \Psi_s$ .

Definiendo

$$\sigma = 1 - \frac{L_m^2}{L_s L_r} \quad \text{Ecuación 3.11}$$

Podemos reescribir la

$$\sigma L_r = L_r - \frac{L_m^2}{L_s} \quad \text{Ecuación 3.12}$$

Relacionando las ecuaciones anteriores podemos definir  $v_{dr}$  y  $v_{qr}$  de la siguiente manera.

$$v_{dr} = R_r i_{dr} - \omega_r \sigma L_r i_{rq} + \sigma L_r \frac{d}{dt} i_{dr} + \frac{L_m}{L_s} \frac{d}{dt} \Psi_{ds} \quad \text{Ecuación 3.13}$$

$$v_{qr} = R_r i_{qr} - \omega_r \sigma L_r i_{rd} + \sigma L_r \frac{d}{dt} i_{qr} + \omega_r \frac{L_m}{L_s} \Psi_{ds} \quad \text{Ecuación 3.14}$$

En la ecuación de la tensión del rotor en eje  $d$  existen dos sumandos que dependen directamente de  $i_{rd}$ . Lo mismo ocurre en la segunda ecuación en el eje  $q$ . La misión del regulador PI será ajustar la salida para que estos dos sumandos obtengan el valor que deban. El resto de sumandos, a excepción de la derivada del flujo en la primera ecuación que es nula en régimen permanente, se denominan términos de compensación, o términos cruzados, pues dependen cada uno de la corriente obtenida en la otra ecuación.

La conversión de las señales de corriente en voltaje se muestra en la figura 3.35, así como la aplicación de la transformada de Park inversa, obteniendo así la señal de referencia para producir la señal PWM, que controlara los dispositivos de potencia del VSC.

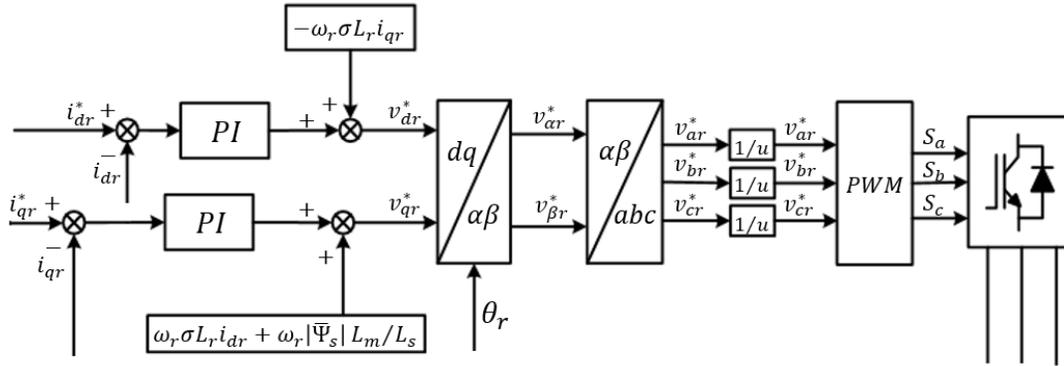


Figura 3.35 Generación de la señal PWM.

En la figura 3.36 se muestra la implementación en Simulink de la transformación de las señales de corriente  $i_{dr}$  e  $i_{qr}$  en voltaje  $v_{dr}$  y  $v_{qr}$ .

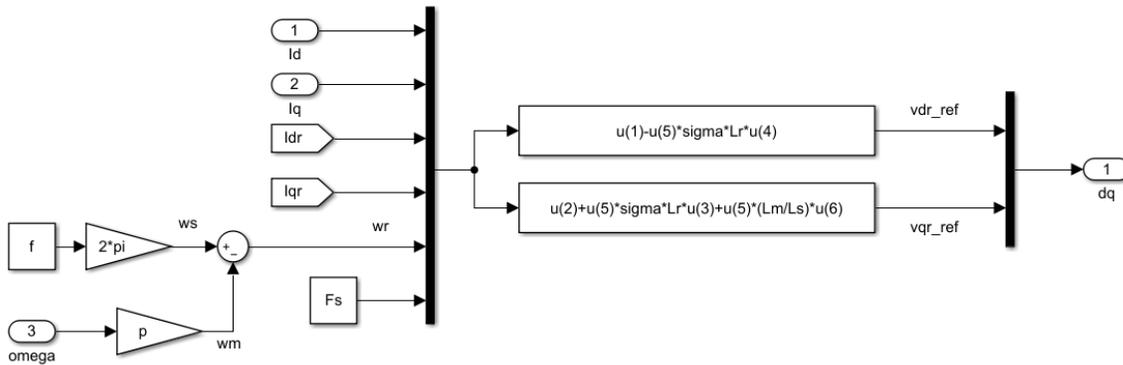


Figura 3.36 Transformación de señales de corriente a señales de voltaje.

Después de ese ajuste es necesario regresar a la referencia  $abc$ , por lo que es necesario aplicar la transformada de Park inversa.

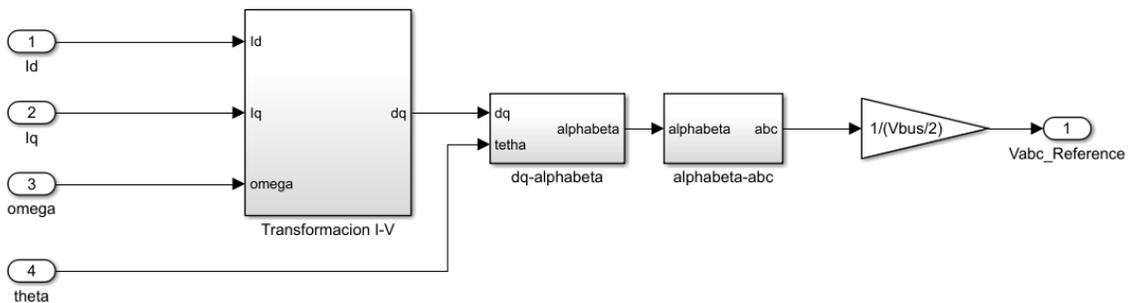


Figura 3.37 Transformada de la señal a la referencia abc.

La transformada de Park se realiza en dos parte primero transformando del sistema de referencia  $dqo$  al sistema  $\alpha\beta\gamma$  como se muestra en la figura 3.37 que se determina por la siguiente ecuación.

$$\begin{bmatrix} v_\alpha \\ v_\beta \\ v_\gamma \end{bmatrix} = [T]_{dqo \rightarrow \alpha\beta\gamma} \begin{bmatrix} v_d \\ v_q \\ v_o \end{bmatrix}$$

Donde

$$[T]_{dqo \rightarrow \alpha\beta\gamma} = \begin{bmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad \text{Ecuación 3.15}$$

La ecuación 3.15 es la matriz transpuesta de la matriz mostrada en la ecuación 1.32. La implementación de la transformación anterior se muestra en la figura 3.38. Dado que el eje  $\gamma$  coincide con el eje  $o$  no se toma en cuenta en la transformada.

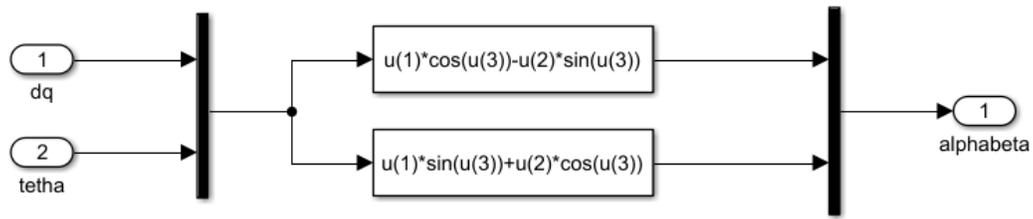


Figura 3.38 Transformación del sistema dqo al  $\alpha\beta\gamma$ .

Ahora que ya se tiene en el sistema  $\alpha\beta\gamma$  se transforma al sistema  $abc$ , por medio de la siguiente ecuación.

$$\begin{bmatrix} v_a \\ v_b \\ v_c \end{bmatrix} = [T]_{\alpha\beta\gamma \rightarrow abc} \begin{bmatrix} v_\alpha \\ v_\beta \\ v_\gamma \end{bmatrix}$$

Donde

$$[T]_{\alpha\beta\gamma \rightarrow abc} = \begin{bmatrix} 1 & 0 & 1 \\ -\frac{1}{2} & \frac{\sqrt{3}}{2} & 1 \\ -\frac{1}{2} & -\frac{\sqrt{3}}{2} & 1 \end{bmatrix} \quad \text{Ecuación 3.16}$$

La ecuación 3.16 es la matriz transpuesta de la mostrada en la ecuación 1.29. Dado que la señal de  $v_\gamma$  será igual a cero por las transformaciones anteriores, al implementarlo en Simulink se desprecia como se muestra en la figura 3.39.

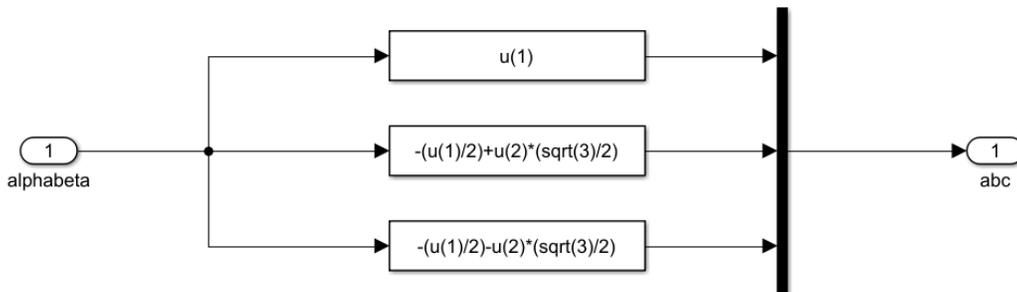


Figura 3.39 Transformación del sistema  $\alpha\beta\gamma$  al  $abc$ .

Por ultimo encontramos el bloque encargado del accionamiento de los dispositivos de potencia que conforman el puente de la VSC, este es el generador de señales PWM. Como el que se muestra en la figura 3.40.

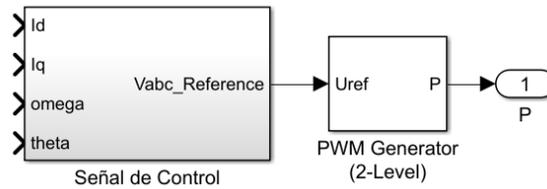


Figura 3.40 Generador de señales PWM.

La configuración del generador de señales PWM se muestra en la figura 3.41. En este caso se utiliza para un puente trifásico, ya que, como se mencionó anteriormente el puente a utilizar es de tres brazos (trifásico).

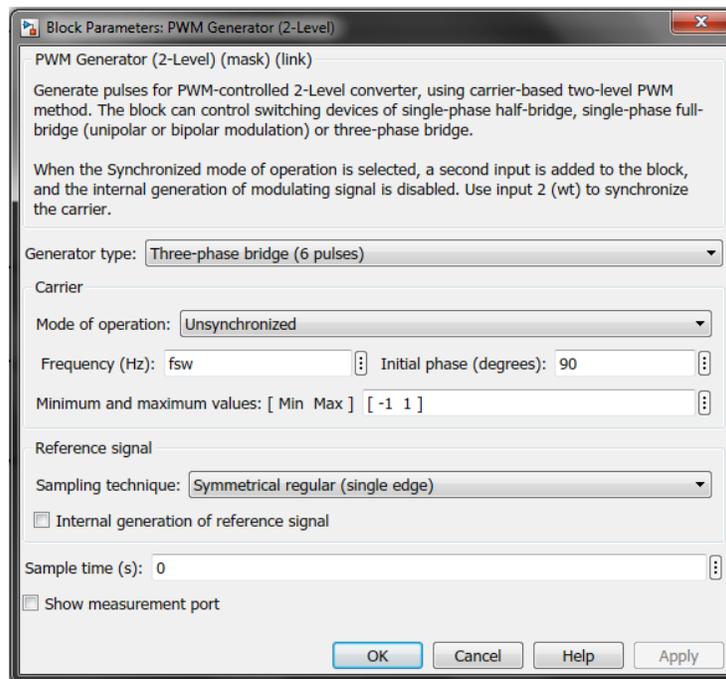


Figura 3.41 Configuración del generador de señales PWM.

La frecuencia de trabajo es igual a  $f_{sw}$  y la técnica de muestreo es la regular simétrica, este tipo de muestreo se explica en el capítulo 2.

En el sistema completo mostrado en la figura 3.1, podemos observar que además, de los bloques del modelo aerodinámico, del modelo mecánico-eléctrico y el de control, se encuentra el bloque powergui, el cual se encarga del correcto funcionamiento de los elementos de la librería Simscape la configuración de este bloque se muestra en la figura 3.42.

En el bloque powergui se establece el tipo de simulación a utilizar, que en este caso es Discreto y por eso se nos solicita el tiempo de muestreo, existen diferentes tipos de simulación los cuales se explican en el capítulo 2.

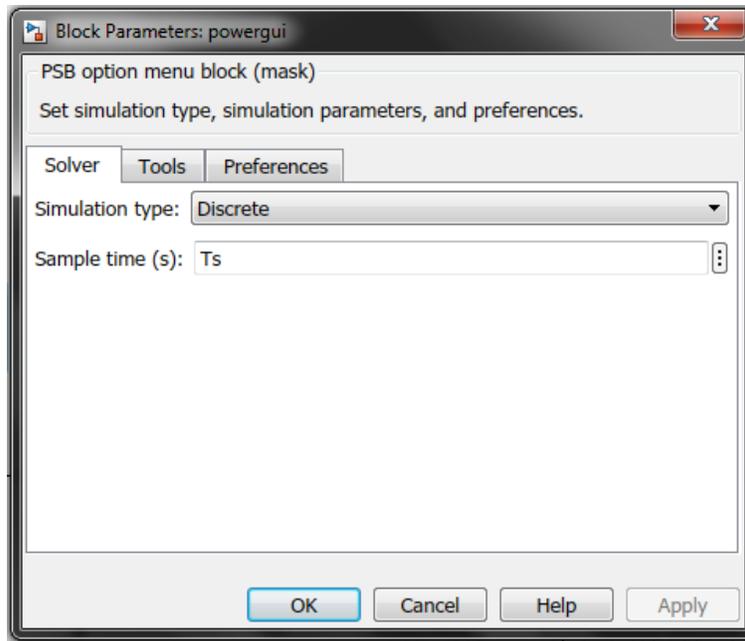


Figura 3.42 Configuración del powergui.

Además se encuentra el bloque de monitoreo, el cual contiene un Scope, como se muestra en la figura 3.43, que nos permite observar las señales de velocidad y torque de referencia, así como las medidas de la máquina.

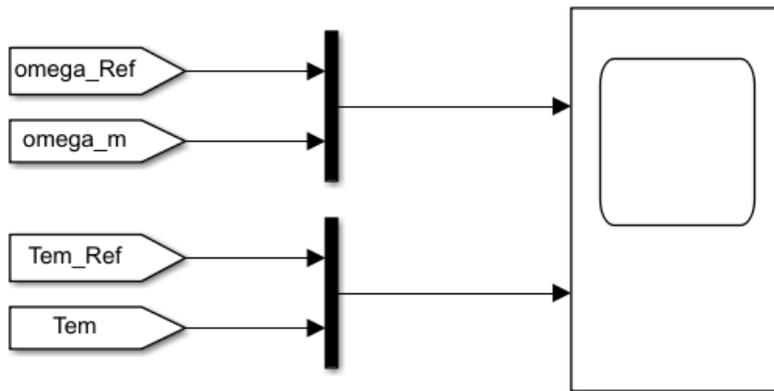


Figura 3.43 Interior del bloque de monitoreo.

Debido a que este sistema de control es de velocidad variable es necesario tener en cuenta la curva de potencia de la máquina, para conocer la velocidad del viento máxima de operación, la turbina implementada tiene una capacidad nominal de 2 MW y su curva se muestra en la figura 3.44.

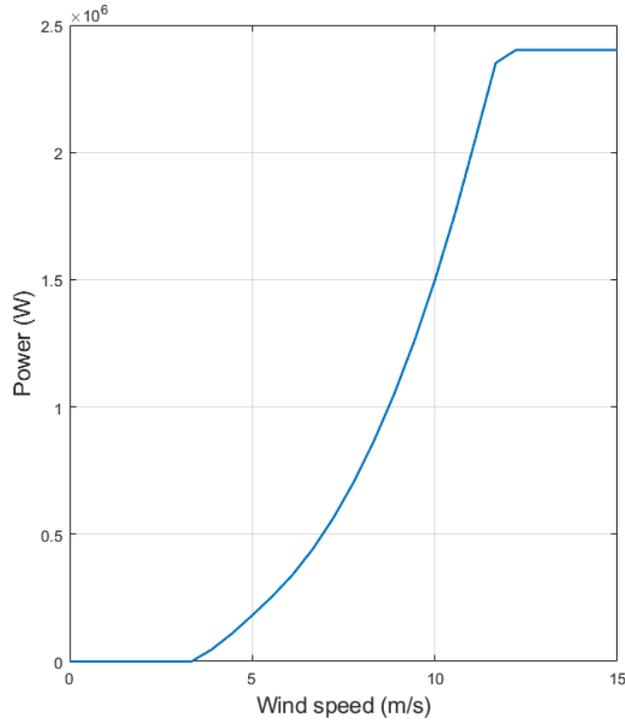


Figura 3.44 Curva de Potencia vs Velocidad del viento para una turbina de 2 MW.

De la curva de potencia podemos observar que la velocidad en la cual la maquina trabaja a su capacidad nominal es cercana a los 11  $m/s$ , por lo que por debajo de ese valor la maquina realizara los ajustes necesarios para operar en óptimas condiciones, pero por encima de ese valor la maquina podría sufrir daños por lo que no se recomienda trabajar por encima de dicha velocidad, además como se mencionó anteriormente la maquina no será capaz de producir una potencia mayor a su capacidad nominal.

En las figuras 3.45-3.48 la velocidad del viento es inferior o igual a 11  $m/s$  y podemos observar que se logra realizar el ajuste de velocidad correctamente.

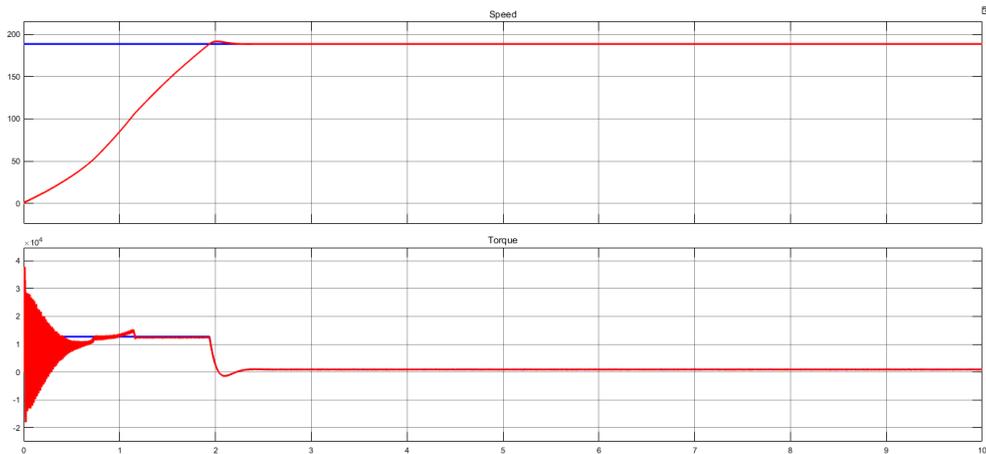


Figura 3.45 Simulación para  $v=6 m/s$  y  $J=127 kg.m^2$ .

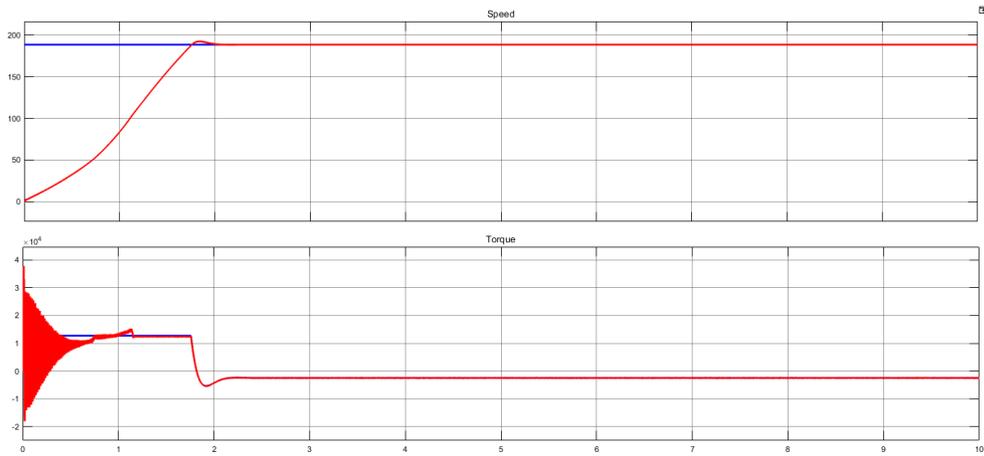


Figura 3.46 Simulación para  $v=8$  m/s y  $J=127$  kg.m<sup>2</sup>.

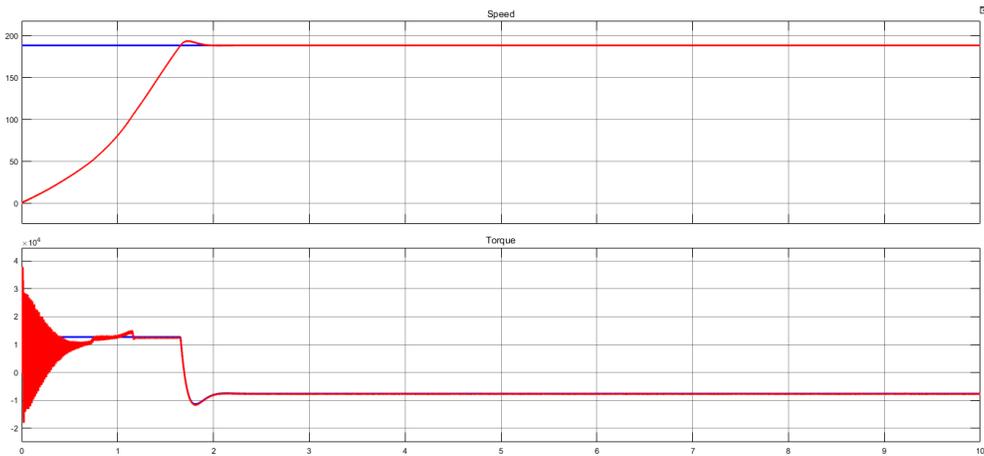


Figura 3.47 Simulación para  $v=10$  m/s y  $J=127$  kg.m<sup>2</sup>.

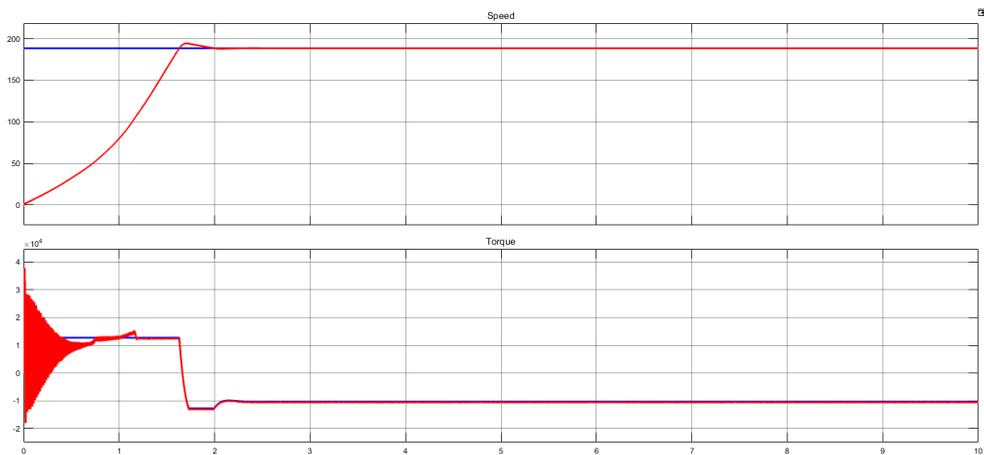


Figura 3.48 Simulación para  $v=11$  m/s y  $J=127$  kg.m<sup>2</sup>.

En la figura 3.49 podemos ver como la velocidad de la maquina sobrepasa la velocidad de referencia sin embargo se podría trabajar en estas condiciones por tiempos muy cortos.



Figura 3.49 Simulación para  $v=12$  m/s y  $J=127$  kg.m<sup>2</sup>.

En la figura 3.50 podemos ver como la velocidad de la maquina se desborda después de superar la velocidad de referencia, dicha condición representa daños en la máquina, si se considera una velocidad del viento mayor el desborde aumentara también.

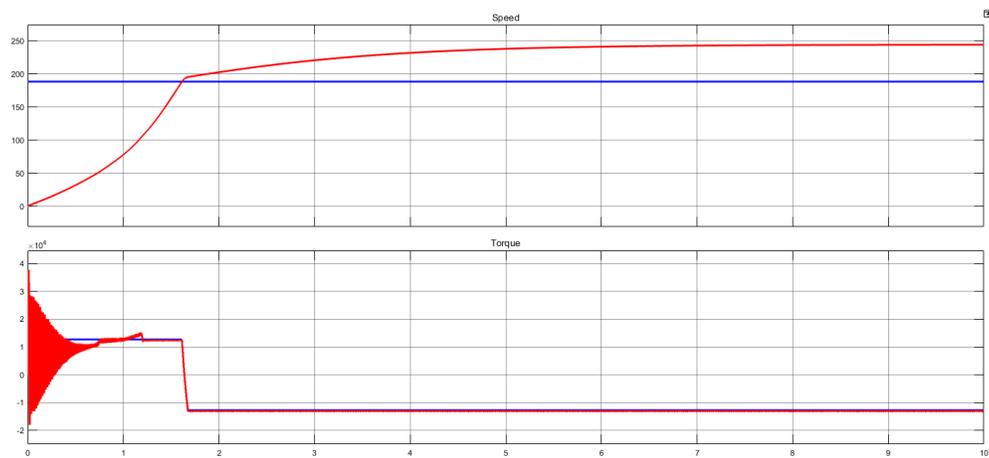


Figura 3.50 Simulación para  $v=13$  m/s y  $J=127$  kg.m<sup>2</sup>.

Otra variable importante es la Inercia, para poder observar el comportamiento en tiempos de simulación breves se han tomado valores de inercia pequeños, pero si dicho valor se modifica podemos ver como el tiempo que tarda la maquina en alcanzar la velocidad de referencia aumenta, en las figuras 3.51-3.53 podemos ver la respuesta para las mismas velocidades de las figuras 3.45-3.47 pero para un valor de inercia de 327 kg.m<sup>2</sup> en lugar de 127 kg.m<sup>2</sup>. Los parámetros han sido modificados de manera que se permita ver la respuesta del sistema en un tiempo corto de simulación, en el sistema real la maquina tendrá tiempos de respuesta mucho mayores a los mostrados en la simulación.

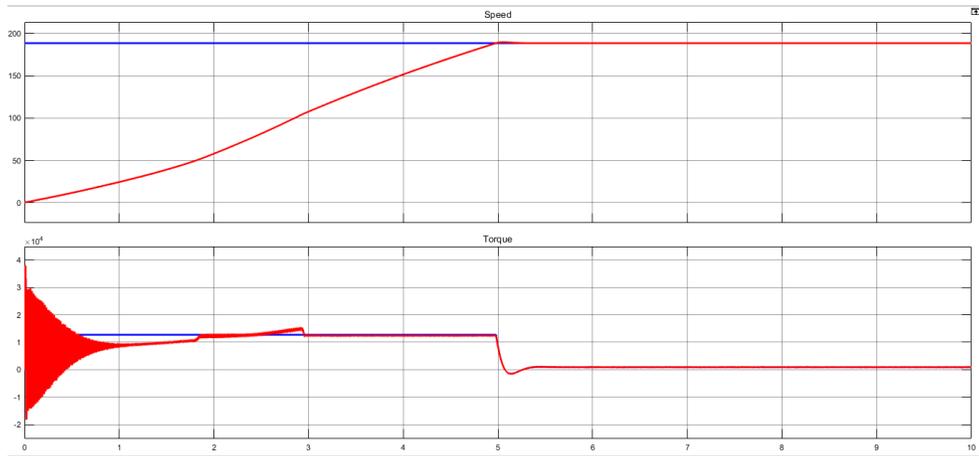


Figura 3.51 Simulación para  $v=6$  m/s y  $J=327$  kg.m<sup>2</sup>.

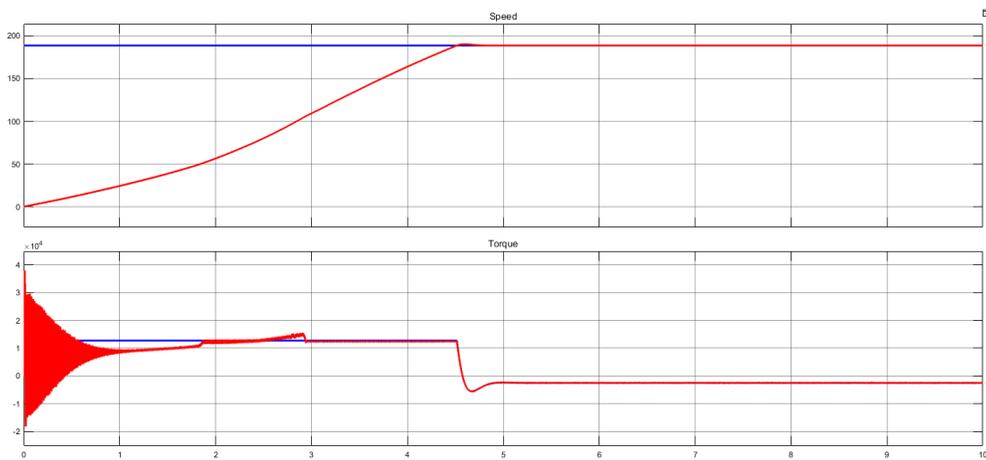


Figura 3.52 Simulación para  $v=8$  m/s y  $J=327$  kg.m<sup>2</sup>.

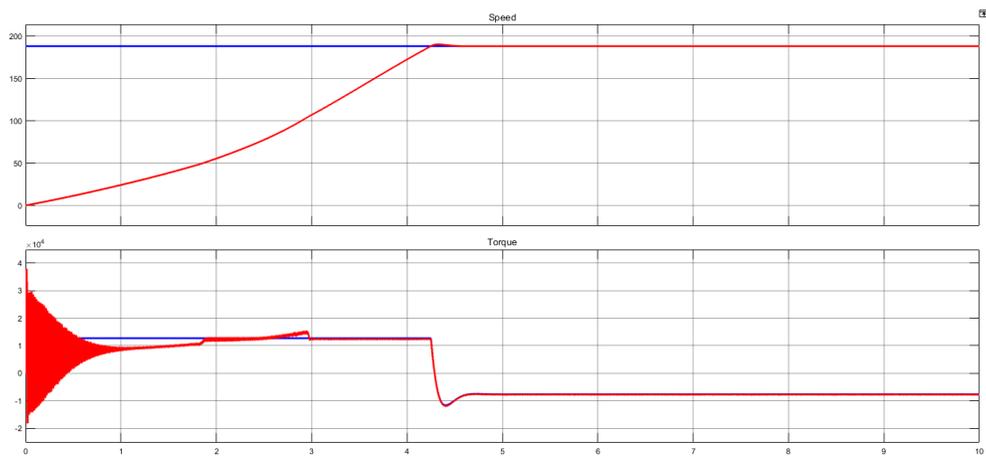


Figura 3.53 Simulación para  $v=10$  m/s y  $J=327$  kg.m<sup>2</sup>.

### Conclusiones del Capítulo 3.

El sistema de control implementado en este capítulo corresponde a un sistema de velocidad variable, ya que la configuración adoptada para el desarrollo de este es la mostrada en la figura 1.18.

El control de velocidad variable le permite al sistema operar en un rango de velocidades inferiores a la velocidad de la potencia máxima, por encima de este valor el comportamiento de la máquina representa un riesgo para la máquina, así como para sus alrededores.

La potencia extraída del viento por la turbina se ve limitada por el coeficiente de potencia, el cual varía dependiendo del cada diseño y cada fabricante. Por consiguiente al variar el coeficiente de potencia también varía el coeficiente de torsión.

El ángulo de la inclinación de la pala siempre será el óptimo con respecto a la dirección del viento, en caso de simular un sistema con control de orientación tendría que agregarse la medición del ángulo e incluirlo en el cálculo de los coeficientes de potencia y de torsión.

Para obtener tiempos de respuesta cortos es necesario realizar cambios en las variables, ya que una simulación en tiempo real de la máquina tardaría un par de minutos en alcanzar su velocidad de operación.

La aplicación de las transformadas de Clarke y Park permite en cambio de un sistema trifásico  $abc$  a uno de dos fases  $dq$  al alinear el flujo del estator con el eje de referencia  $d$  se elimina la componente del flujo en el eje  $q$  lo que simplifica el modelo.

**CAPÍTULO 4.**

**GENERALIDADES E  
IMPLEMENTACIÓN DE LA DSP.**

## CAPÍTULO 4

### GENERALIDADES E IMPLEMENTACIÓN DE LA DSP.

Una DSP es una tarjeta digital procesadora de señales, entre los fabricantes más destacados se encuentra Texas Instrument, la cual posee diferentes familias dedicadas a diferentes aplicaciones, una de ellas es la familia C2000, la cual ha sido desarrollada para aplicaciones de control de motores. La gran ventaja que tienen estas tarjetas sobre otros microprocesadores es su rapidez de respuesta, permite una interacción mediante un software especializado en “tiempo real” entre la tarjeta y el sistema físico empleado.

La tarjeta DSP será la encargada de reemplazar el bloque de control, en el modelo creado en Simulink en el capítulo 3, a partir del bloque se obtiene el código a implementar por la DSP. Para el uso de esta tarjeta es necesario realizar un acondicionamiento de las señales medidas en la máquina.

El diseño de la interfaz de comunicación entre el sistema a controlar y la DSP, se realizara en base a las señales que se necesitan acondicionar. Y la comunicación entre la DSP y la computadora se realiza por medio del Code Composer Studio.

#### 4.1 Procesador Digital de Señales (DSP).

El procesador digital de señales es un dispositivo diseñado para la manipulación de operaciones matemáticas típicas, normalmente se utilizan con sensores digitales. El objetivo primordial es realizar el proceso lo más rápido posible, para generar una señal de salida con un nuevo valor (corregido) en “tiempo real”.

#### 4.2 Las familias de DSP de Texas Instrument.

La familia de procesadores TMS320 de Texas Instrument está formada por tres generaciones de DSP's de punto fijo; la C2000 con sus familias C24x y C28x; la C5000 con las familias C54x y C55x; y la C6000 con las familias C62x, C64x y la familias C67x que es de punto flotante. Los procesadores poseen diferentes arquitecturas diseñadas específicamente para aplicaciones en tiempo real.



Figura 4.1. Familias de DSP de Texas Instrument TMS320.

##### 4.2.1 SERIE C2000

- Familia TMS320C24x.

La familia de procesadores TMS320C24x son dispositivos con un formato de punto fijo de 16 bits y un rendimiento de 40 MIPS. Estos DSP de bajo costo, están diseñados para aplicaciones de control de motores (DMC, Digital Motor Control) y otras aplicaciones embebidas, por lo que se les conoce también como Controlador Digital de Señal basado en DSP. Esta serie se basa en el núcleo c2xLP de punto fijo de 16 bits, y es complementado en un mismo encapsulado con una amplia gama de periféricos tal como; manejadores de eventos para la generación PWM y tiempo muerto, ADC ultrarrápidos, circuito de captura al impulso, timer, y puerto serie asíncrono. Los dispositivos TMS320LF24xx pueden incluir hasta 64 kB de memoria flash, mientras los LC24xx incluyen memoria ROM/RAM.

La arquitectura del controlador DSP 320C24x se basa en una arquitectura Harvard mejorada. El C24x tiene múltiples espacios de memoria accesibles desde tres buses paralelos: un bus de programa, un bus de lectura de datos y un bus de escritura de datos.

La serie 240xA se compone de los siguientes dispositivos: 2401A, 2402A, 2403A, 2404A, 2406A, 2407A, los cuales pueden tener una variante en cuanto a su configuración de memoria, teniendo ya sea memoria flash o ROM.

- **Familia TMS320C28x**

La familia de procesadores TMS320C28x comprende dispositivos de 32 bits de punto fijo y una velocidad que va de los 60MHz a los 150MHz, voltaje de alimentación de 1.8V para dispositivos flash y 1.9V para los RAM/ROM. Estos dispositivos tienen mejores características que su antecesora; además, se tiene un amplio grupo de periféricos; ADC, Módulo CAN, SCI, SPI, IIC (I2C), todos integrados en un mismo encapsulado. Gracias a su arquitectura Harvard modificada y un reducido conjunto de instrucciones RISC el CPU puede leer instrucciones y datos mientras escribe simultáneamente para mantener la operación en un solo ciclo a través del paralelismo. Esta tarea la hace el CPU por medio de seis buses separados de dirección y datos.

La familia incluye los dispositivos TMS320x2810, TMS320x2811, y TMS320x2812, donde x puede ser reemplazada por una F para los dispositivos Flash, C para los dispositivos ROM y R para los dispositivos RAM. De igual modo se tienen los dispositivos TMS320x2801, TMS320x2802, TMS320x2806 y TMS320x2808 con sus diferentes variantes en cuanto al tipo de memoria de la que disponen. Al igual que la familia TMS320C24x, estos DSP son de bajo costo y están diseñados para una amplia gama de aplicaciones de control de motores DMC, censado para control de velocidad, PWM y corrección del factor de potencia.

Una importante característica de la familia es que tiene una librería de trabajo denominada "IQ-math" y le da al diseñador la facilidad de desarrollar transparentemente algoritmos de punto flotante con el bajo costo de una arquitectura de punto fijo.

#### 4.2.2 SERIE C5000

- **Familia TMS320C54x**

La familia TMS320C54x son dispositivos con un formato de punto fijo de 16 bits. Los primeros miembros fueron introducidos en Japón en 1994 y en Estados Unidos en 1995. El dispositivo más veloz corre a una velocidad de 160 MHz y un voltaje de alimentación del núcleo de 1.6 V. Por su parte, el más lento de la familia corre a 120 MHz y 1.5 V. Existe además la familia TMS320C547x, la cual integra un sistema que incluye un DSP basado en la arquitectura del TMS320C54x y un microcontrolador tipo RISC basado en el núcleo ARM7TDMI. El sistema microcontrolador integra tres timers, I/O de propósito general, una interface de memoria externa y una interface Ethernet. Las aplicaciones planeadas para los TMS320C54x incluyen comunicaciones inalámbricas y celulares, módems, audio-Internet, telefonía por cable y redes.

Existen otros dispositivos de la familia TMS320C54x que son multi-núcleo; es decir integran dos o hasta cuatro núcleos con la misma arquitectura (TMS320VC5420/5421/5441). Cada uno de estos núcleos integrados en un solo dispositivo tiene sus propios espacios de datos y de programa, permitiendo con su arquitectura Harvard modificada accesos simultáneos a las instrucciones de programa y datos. El procesador accede a la memoria con cuatro juegos de buses de datos y de direcciones. Un juego de bus de programa, dos juegos de buses de lectura de datos y un juego de bus de escritura de datos.

Una característica esencial de esta familia es que proporciona tres modos de control para un menor consumo de energía; estos modos se programan con las instrucciones IDLE1, IDLE2, e IDLE3. En

todos los modos IDLE, el reloj del núcleo del procesador se apaga, reduciendo el consumo de energía. Los demás modos son variantes en los que desconectan algunos periféricos o el cristal oscilador.

- **Familia TMS320C55x**

Otra de las familias de la serie C5000 son los DSP de 16 bits de punto fijo TMS320C55x, anunciados en febrero del 2000. Pueden ejecutar hasta dos instrucciones en paralelo, con una anchura de instrucción que varía de los 8 a 48 bits, dependiendo del número de operandos y operaciones paralelas. Los TMS320C55x se basan en su familia antecesora, el TMS320C54x, pero agregan mejoras a la arquitectura del procesador y juego de instrucciones. Los TMS320C55x son parcialmente compatibles en código ensamblador con los TMS320C54x. El sistema de memoria dispone de una arquitectura de Harvard modificada con espacios de memoria para programa y datos por separado. El TMS320C55x posee instrucciones que usan un bus de dirección de 24 bits y un bus de datos de 32-bit para la memoria de programa. El TMS320C55x incluye cinco juegos de bus de memoria de datos unidireccionales: tres buses de lectura de datos y dos buses de escritura de datos. Cada juego de bus incluye un bus de dirección de 24 bits y un bus de datos de 16 bits.

La familia está enfocada para aplicaciones que requieren una combinación fuerte entre rendimiento del DSP y una eficiencia en el consumo de energía. Las aplicaciones planeadas para los TMS320C55x incluyen teléfonos celulares y módems e infraestructura de telecomunicaciones, tal como voz vía IP y bancos de módems multi-canal. Las interfaces usadas directamente a SDRAM los hace satisfactorios para el uso en productos portátiles, donde se requieren amplios buffers de memoria, como cámaras digitales y reproductores de audio digital portátiles (CD player). Algunos de los miembros de esta familia corren a una velocidad de 300 MHz y un voltaje de 1.26 V, como es el caso del TMS320C5501.

Una variante que tiene la familia TMS320C55x es la serie OMAP59xx, en la cual se integra un núcleo de DSP basado en el TMS320C55x y una unidad de procesador (MPU) basada en el ARM9TDMI, para aplicaciones de alto rendimiento y bajo consumo de energía. Esta arquitectura permite ofrecer capacidades de procesamiento de señales en tiempo real por parte del DSP y acoplarlo a las funcionalidades de control del procesador ARM9TDMI. Los periféricos actuales en todos los miembros de esta familia son; interfaz para puerto host, tres puertos serie (dos en el TMS320C5501), dos o tres temporizadores, seis canales de controlador DMA. Varios de ellos además incluyen temporizador watchdog, y reloj en tiempo real.

En algunas versiones se encuentran disponibles Interfaces tal como I2C y UART. Los TMS320C5507 y TMS320C5509A incluyen un puerto USB 2.0 y los TMS320C5509A incluyen un puerto serie Digital y Multimedia Card/Secure.

#### **4.2.3 SERIE C6000**

- **Familia TMS320C62x**

La familia de procesadores TMS320C62x son dispositivos con un formato de punto fijo. Están basados en una arquitectura VLIW, la cual permite ejecutar hasta ocho instrucciones RISC por ciclo de reloj. El primer miembro de la familia, el TMS320C6201, tiene una velocidad de 200 MHz, voltaje de alimentación del núcleo de 1.8 V y una ejecución de 400 MMACs/seg. En septiembre de 1999, aparecen dos miembros más, el TMS320C6202 y el TMS320C6211, con velocidades de procesamiento de 250 MHz y 150 MHz, respectivamente. Texas Instruments ofrece además la familia TMS320C67x, la cual extiende la arquitectura del TMS320C62x con soporte de aritmética en punto flotante y 64 bits de datos. El sistema de memoria del TMS320C62x implementa una arquitectura Harvard modificada, manteniendo separados los espacios de dirección para el programa y memoria de datos. La memoria

del programa tiene un bus de dirección de 32 bits y un bus de datos de 256 bits. La memoria de datos tiene dos buses de dirección de 32 bits y dos buses de datos de 32 bits.

Los periféricos que comúnmente tiene esta familia son cuatro controladores DMA, interfaz para puerto host, dos temporizadores de 32 bits, dos puertos serie multicanal, etc. Algunas de sus aplicaciones incluyen las médicas, industrial, reconocimiento de lenguaje, imagen y gráficos en 3D.

- **Familia TMS320C64x**

La familia TMS320C64x, se anunció en febrero del 2000. Estos dispositivos comprenden el TMS320C6410, TMS320C6412, TMS320C6413 y son de punto fijo con un formato de palabra VLIW. El TMS320C64x es sucesor de la familia TMS320C62x y su juego de instrucciones es superior. Otra de las características que agrega respecto al TMS320C62x es la capacidad de procesamiento SIMD y un número especializado de instrucciones orientadas a las telecomunicaciones, entre otras mejoras. La familia TMS320C64x puede ejecutar el código de la familia TMS320C62x, sin ningún problema, pero el TMS320C62x no puede ejecutar todas las instrucciones del TMS320C64x. La arquitectura de memoria que implementa es de tipo Harvard modificada, en donde se mantienen separados los espacios de direcciones para instrucción y memoria de datos. El TMS320C64x busca instrucciones usando un bus de dirección de 32 bits y un bus de datos de 256 bits.

En el 2003, los miembros más rápidos de la familia eran el TMS320C6414, TMS320C6415, y TMS320C6416, los cuales trabajan a una frecuencia base de 720 MHz, y pueden llegar hasta 1GHz en versiones del mismo modelo de DSP.

Otros dispositivos en esta familia son; TMS320DM640, TMS320DM641, TMS320DM642, correspondientes a la serie TMS320DM64x, están especialmente diseñados para aplicaciones digitales de multimedia. Estos DSP integran puertos seriales de audio, puertos de video, controladores DMA multicanal, interfaz para puerto host, temporizadores de 32 bits, puertos serie multicanal, permitiendo aplicaciones de alto rendimiento tales como estaciones base inalámbricas, módems multilínea, equipo de diagnóstico de imagen, audio, video, sistemas de radar y sonar. Para el caso específico del TMS320C6416, éste se enfoca a aplicaciones de comunicaciones de alto desempeño, incluye un coprocesador que realiza operaciones de turbo decodificación y Viterbi.

- **Familia TMS320C67x**

La familia TMS320C67x es una familia de DSP de punto flotante de 32 bits, que se deriva de la familia TMS320C62x de punto fijo y 16 bits de ancho de palabra. Esta familia puede ejecutar todas las instrucciones del TMS320C62x. Además, integra el estándar IEEE-754 para la representación de números en punto flotante con precisión simple (de 32 bits) o precisión doble (de 64 bits) y también cómo deben realizarse las operaciones aritméticas con ellos. Las aplicaciones de esta familia se enfocan a aquellas de alto nivel de procesamiento, audio, gráficos en 3D, imágenes médicas, radar y reconocimiento de lenguaje. El TMS320C67x implementa una arquitectura de memoria Harvard modificada, en donde se mantienen separados los espacios de direcciones para instrucción y memoria de datos. El TMS320C67x busca instrucciones usando un bus de dirección de 32 bits y un bus de datos de 256 bits.

Los primeros miembros de la familia TMS320C67x se anunciaron en abril de 1998 y para mediados de 2003 la familia incluía los DSP; TMS320C6701, TMS320C6711, TMS320C6712 y TMS320C6713. El dispositivo más rápido de la familia es el TMS320C6713 con una velocidad de 300 MHz y un voltaje

de alimentación del núcleo de 1.4 V. Otros miembros de la familia operan el intervalo de 150 a 250 MHz, dependiendo de la variante del dispositivo.

El TMS320C67x es compatible con el TMS320C62x; es decir; puede ejecutar cualquier código de este último, pero el TMS320C62x no puede ejecutar todas las instrucciones del TMS320C67x. Tampoco es compatible con la familia TMS320C64x.

En la figura 4.2 se muestra un resumen de las principales áreas de aplicación y en la figura 4.3 observamos la evolución que han tenido las DSP's de Texas Instruments.

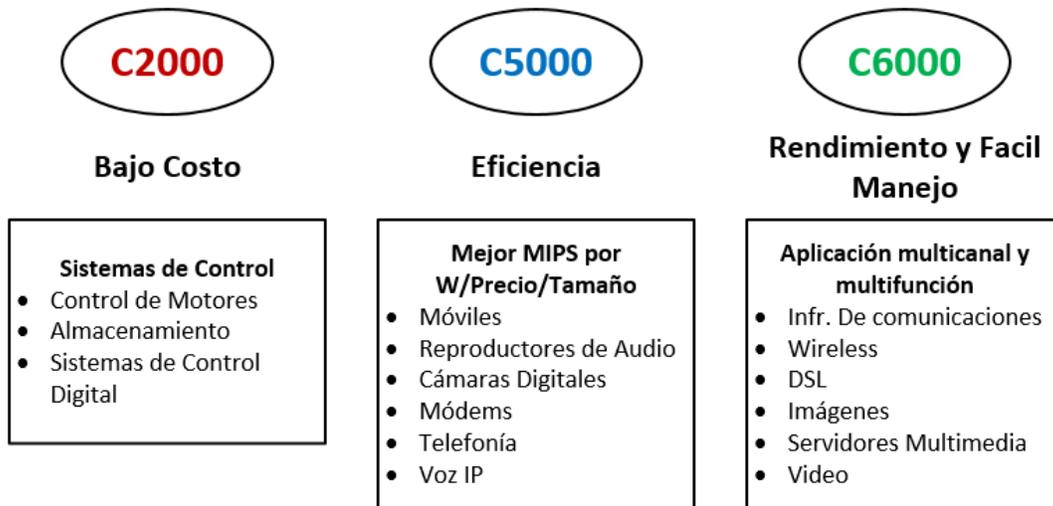


Figura 4.2. Áreas de aplicación de las DSP por familias.

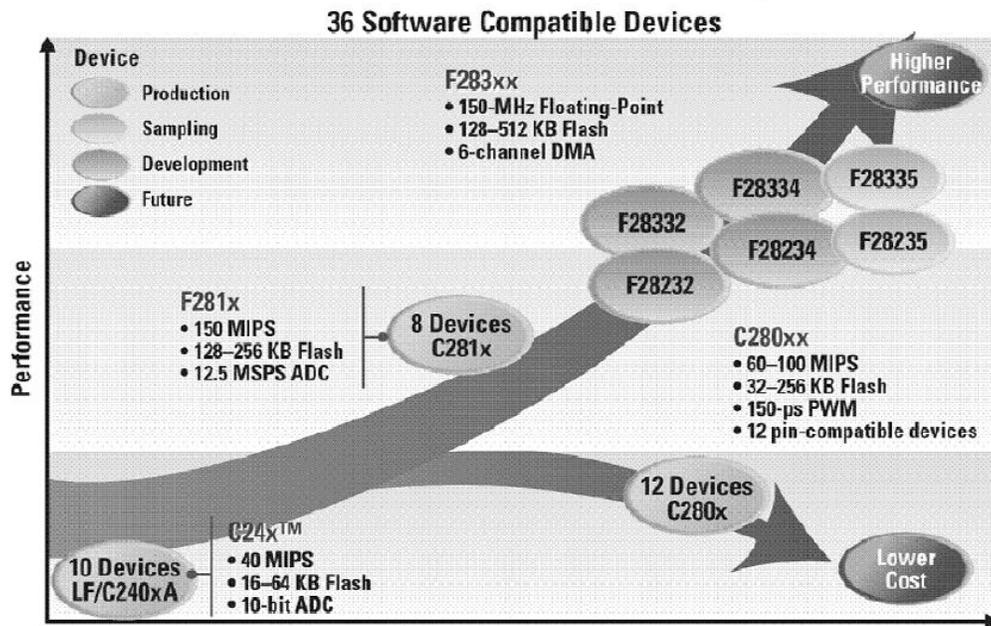


Figura 4.3. Evolución de las familias de DSP.

### 4.3 El diagrama de bloques del TMS320F2812.

El diagrama de bloque TMS320F2812 se puede dividir en 4 bloques funcionales:

- Bus interno y externo
- Unidad central de proceso (CPU)
- Memoria
- Periféricos

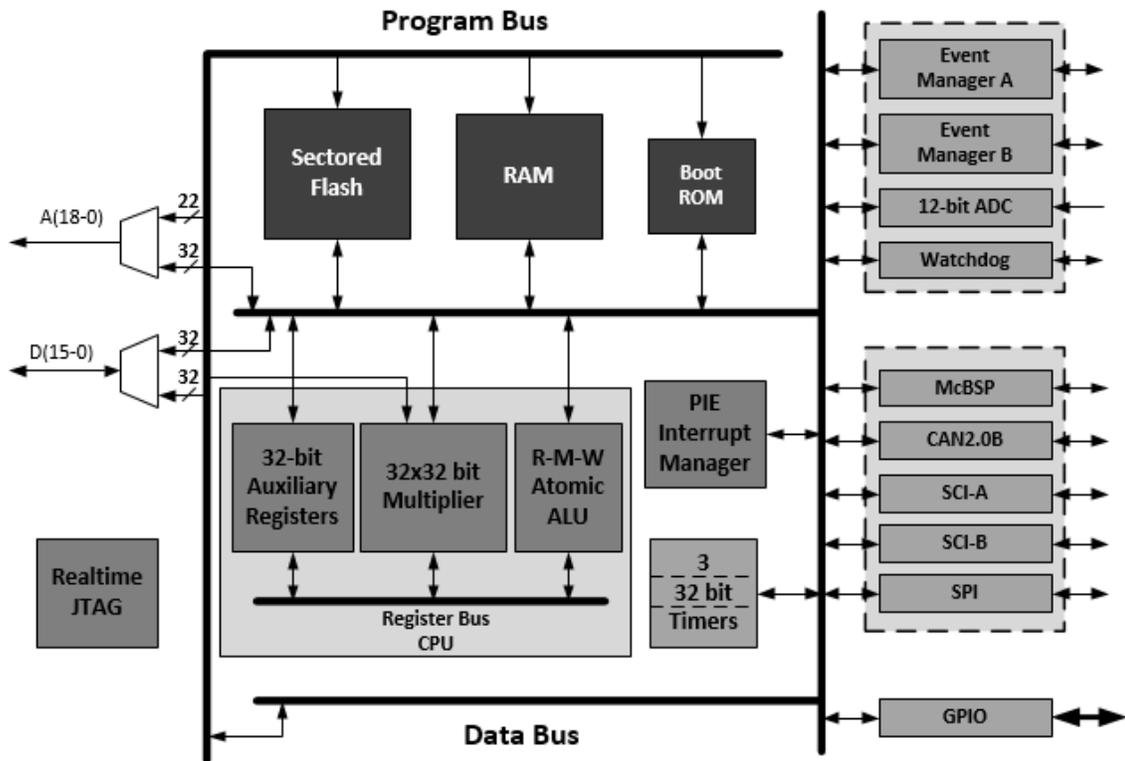


Figura 4.4 Diagrama de bloques del C281X.

Para recuperar dos operandos de la memoria en la CPU en un solo ciclo de reloj el F2812 cuenta con 2 buses independientes: el bus de programa y bus de datos. Este tipo de arquitectura se le conoce como "Arquitectura Harvard". Pero debido a que el F2812 tiene la capacidad de leer operandos no solamente de la memoria de los datos sino también de la memoria del programa. Se dice que este dispositivo tiene una "Arquitectura Harvard Modificada"

En el lado izquierdo de la figura se pueden ver dos bloques, uno del multiplexor para los datos (D15-D0) y el otro para la dirección (A18-A0). Es una interfaz para conectar los dispositivos externos con el F2812. Ahora obsérvese que.

1. La anchura del Bus de datos externo es solamente 16 bits y que
2. No se puede tener acceso a los datos externos del bus de programa y a los datos del bus de datos en el mismo tiempo. Comparado a un solo ciclo para el acceso interno a dos operandos de 32-bits, el acceso a memoria externa realiza al menos 4 ciclos.

#### 4.4 La CPU del F2812

La CPU del F2812 puede ejecutar la mayoría de las instrucciones matemáticas gracias a los registros internos que posee y a una gama de instrucciones que son utilizadas comúnmente por los microcontroladores. La arquitectura también incluye modos de direccionamiento especiales, que facilitan la obtención de un código muy compacto tanto en ensamblador como programación en C.

El F2812 es eficiente en tareas matemáticas de procesamiento así como en las tareas del control de sistema que son manejadas típicamente por los dispositivos del microcontrolador.

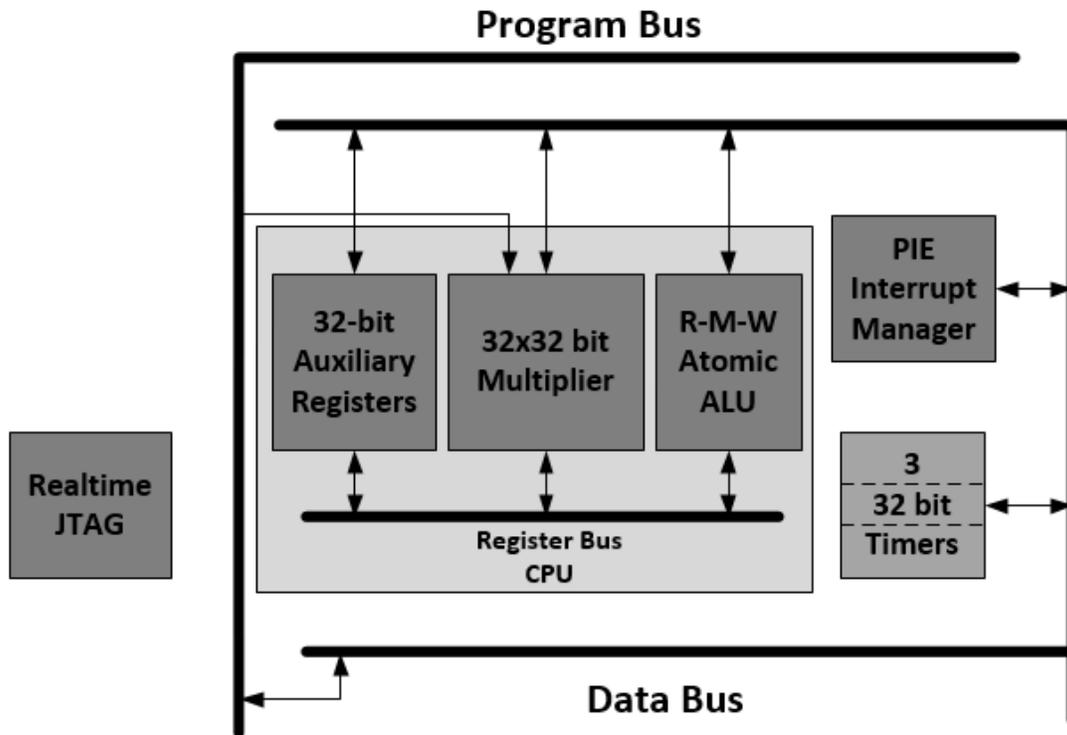


Figura 4.5 Diagrama de bloques de la CPU del C28XX.

Esta CPU incluye tres contadores o temporizadores de 32-bits, que pueden utilizarse para propósitos generales o para la generación de retardos en los sistemas de tiempo real. También cuenta con un gestor de interrupciones del periférico de extensión de interrupciones (PIE) y permite una respuesta rápida a una interrupción de varias fuentes de señales y de acontecimientos externos e internos.

El Hardware multiplicador y una unidad aritmético lógica de 32-bits (ALU) se pueden utilizar en paralelo para ejecutar una multiplicación y una suma simultáneamente. El banco auxiliar de registros está equipado con su propia unidad aritmético lógica (ARAU) también usado en paralelo para mejorar el indicador aritmético.

El interfaz JTAG es una herramienta para facilitar el intercambio de datos en tiempo real entre el DSP y un host durante la fase de eliminación de errores del desarrollo del proyecto. Es posible observar variables en tiempo real a la vez que el código está funcionando, sin ningún retraso del código de control.

#### 4.5 Unidad Matemática del F2812

Los Multiplicadores y Acumuladores (MAC) de 32 bits del F2812 y sus 64 bits internos, permiten a la DSP manejar de manera eficiente los problemas de resolución numéricos de alto nivel, que podrían necesitar la implementación de un procesador de coma flotante los que resultaría más costoso. Además existe la posibilidad de hacer funcionar dos 16 x 16 bites que multipliquen y acumulen instrucciones simultáneamente (MAC Dual) (DMAC).

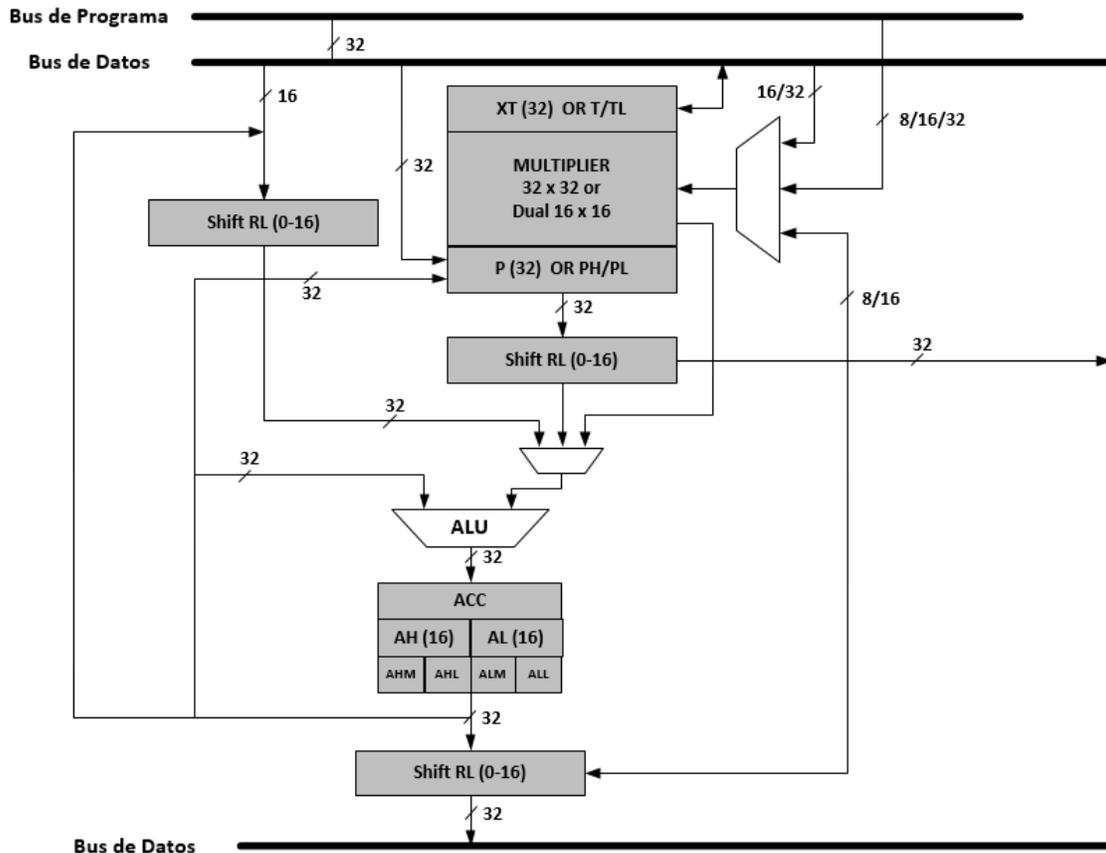


Figura 4.6 Multiplicador y Unidad Aritmética-Lógica del C28XX.

Para la multiplicación es necesario que el primer operando se cargue en el registro XT y para el segundo se debe de cargar en memoria. Si XT se carga de una posición de memoria de datos y el segundo operando se carga de una posición de memoria del programa, la operación de multiplicación puede ser realizada en un solo ciclo. El resultado de una multiplicación se puede cargar en el registro P (producto) o directamente en el acumulador (ACC).

Si la multiplicación es de números de 32 bits, el resultado será de 64 bit. Entonces el F2812 divide la respuesta en dos y coloca una en el registro P y la otra en ACC.

Tres registros de desplazamiento del hardware se pueden utilizar en paralelo a otras unidades del hardware de la CPU. Los registros de desplazamiento son utilizados generalmente para escalar números intermedios en un lazo de control en tiempo real o para multiplicar o dividir por  $2^n$ .

La unidad aritmético lógica (ALU) se encarga de hacer el resto de operaciones matemáticas. El primer operando es siempre el contenido del acumulador (ACC) o de una parte de él. El segundo operando para una operación, se carga de la memoria de los datos, de memoria del programa, del registro de P o directamente de la unidad multiplicadora.

#### **4.6 Placa eZdsp de TMS320F2812**

La eZdsp F2812 es una placa independiente o autónoma, que soporta al Procesador Digital de Señales (DSP) TMS320F2812 de la Texas Instrument, por lo que se convierte en una excelente opción para el desarrollo de software con dicho procesador.

La tarjeta eZdsp F2812 es una placa de circuito impreso doble cara de 5.25 x 3.0 pulgadas. La cual requiere una fuente externa de 5v y 500 mA. La alimentación se le aplicará mediante el conector P6. Si la placa de expansión está conectada al eZdsp se necesitará un amperaje mayor.

La eZdsp F2812 posee las siguientes características:

- Procesador Digital de Señales TMS320F2812
- Velocidad de operación de 150 MIPS (Millones de instrucciones por segundo).
- 18K palabras en memoria RAM
- 128K palabras en memoria Flash.
- Reloj de 30 MHz
- 2 Conectores de Expansión (Análogo, I/O).
- Controlador JTAG IEEE 1149.1
- Operación única a 5V.
- Drivers para la aplicación TI F28XX Code Composer Studio

Entre las características de la DSP TMS320F2812 tenemos:

- Velocidad de Reloj: 150 MHz
- Memoria: 256 Kb (expandible hasta 1 Mb)
- Señales PWM (Pulse Width Modulation): 16 canales
- Conversión Analógica Digital: 16 canales, resolución de 12 bit, tiempo de conversión de 80 ns.
- Pines de Entrada/Salida: Hasta 56
- Niveles de señal: (0 - 3.3V), (0 - 3V) en pines ADC.

La eZdsp consta de 4 bloques lógicos principales:

- Conector de Interfaz Analógica
- Conector de Interfaz I/O
- Interfaz JTAG
- Controlador Interfaz de Puerto Paralelo

La configuración básica de la eZdsp F2812 se muestra mediante el siguiente diagrama de bloques mostrado en la figura 4.7 las interfaces principales del eZdsp son las interfaces JTAG y la interfaz de expansión.

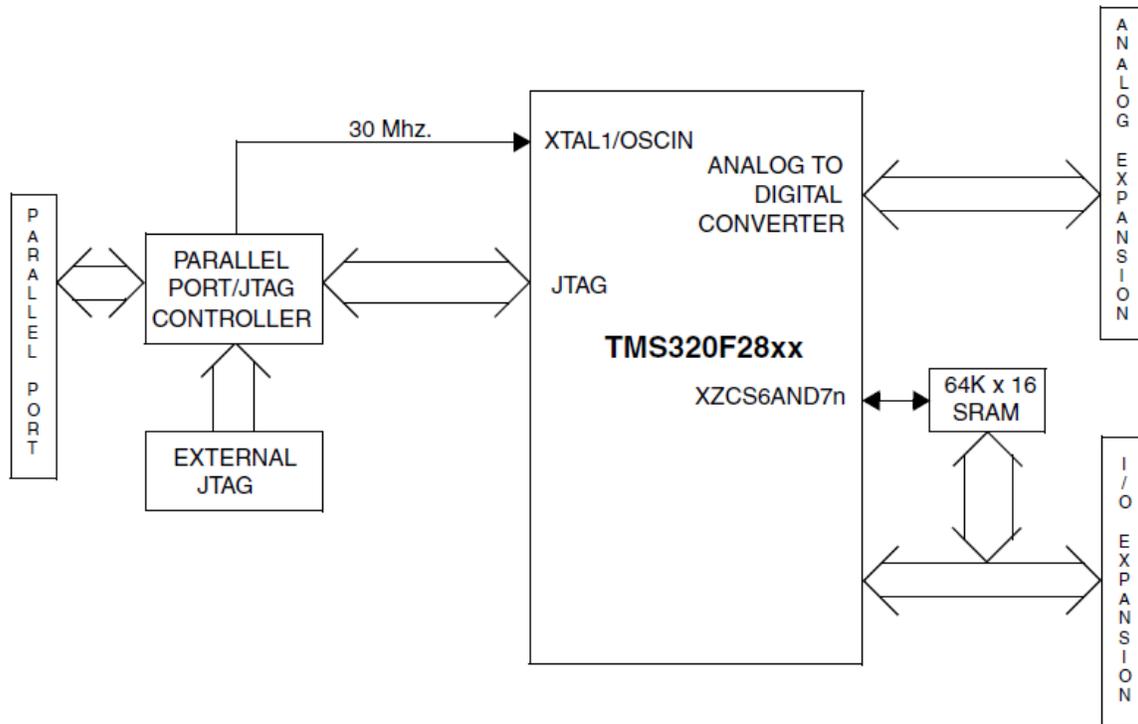


Figura 4.7 Diagrama de bloques de la eZdsp F2812.

La DSP se conecta a la computadora anfitrión a través de un puerto paralelo estándar usando un interfaz JTAG (Joint Test Action Group, IEEE 1149.1). La F281x ofrece JTAG en tiempo real, una característica que no está disponible en otros procesadores de la serie C2000. JTAG en tiempo real permite al usuario la opción de modificar el contenido de la memoria y periféricos mientras el procesador está corriendo. Debido a que los programas son a través de Simulink no se tomara ventaja de esta característica.

Existen dos tipos de tarjeta eZdsp F2812, una que posee base de conexión y otra que no. En la figura 4.8 se muestra la disposición de ambas.

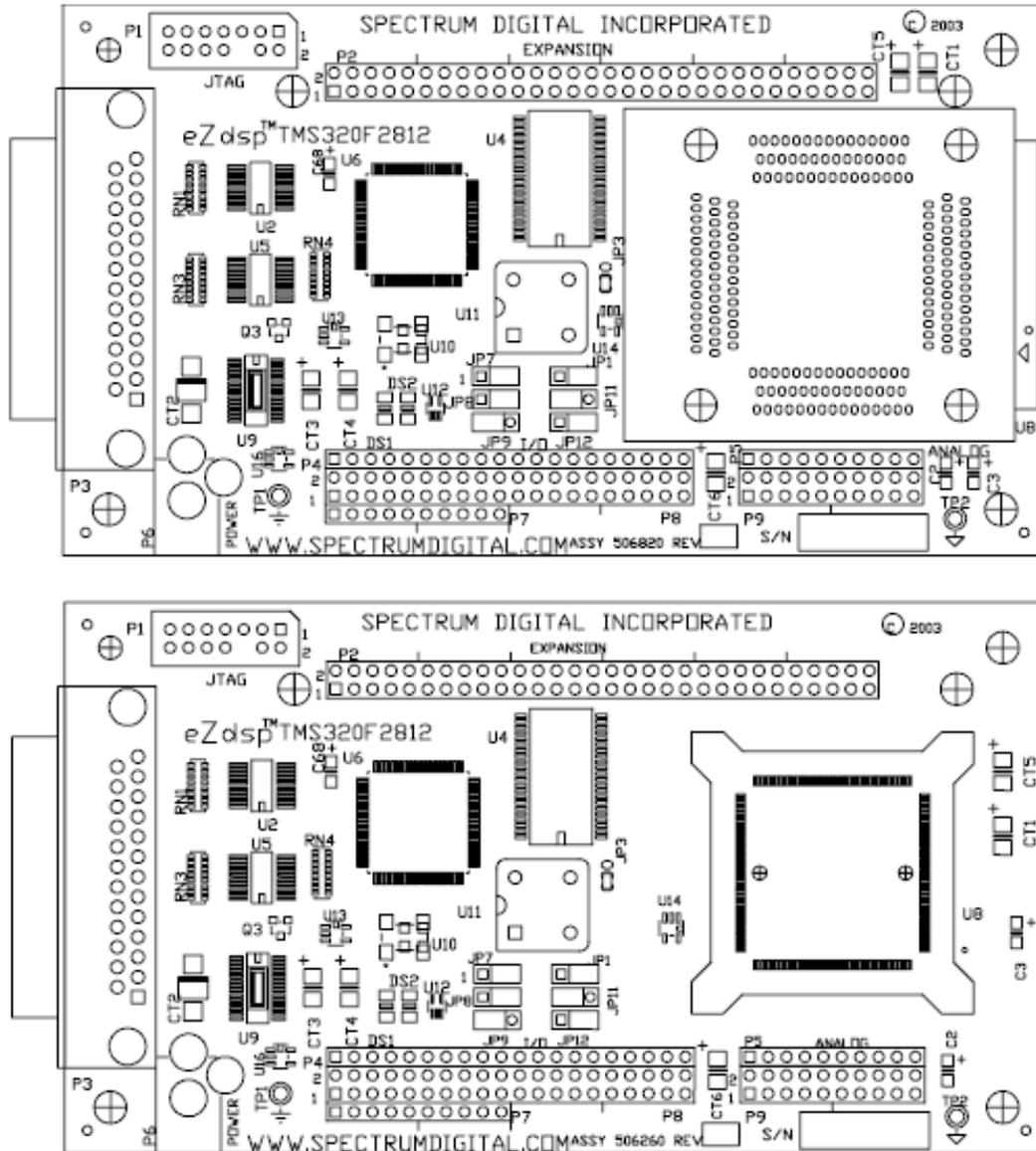


Figura 4.8 Distribución física de la placa eZdsp F2812.

La diferencia principal radica en que la tarjeta F2812 inferior no posee base para colocar el chip TMS320F2812, es decir que ya viene incorporado a la placa, mientras que la tarjeta superior si posee dicha base, lo que facilita la instalación o extracción del chip.

Entre las capacidades que posee la tarjeta eZdsp tenemos:

- 128 x 16 Flash
- 2 Bloques de 4K x 16 SARAM (Single Access RAM)
- 1 Bloque de 8K x 16 SARAM
- 2 Bloques de 1K x 16 SARAM

La configuración del mapa de memoria de la eZdsp F2812 se muestra en la figura 4.9.

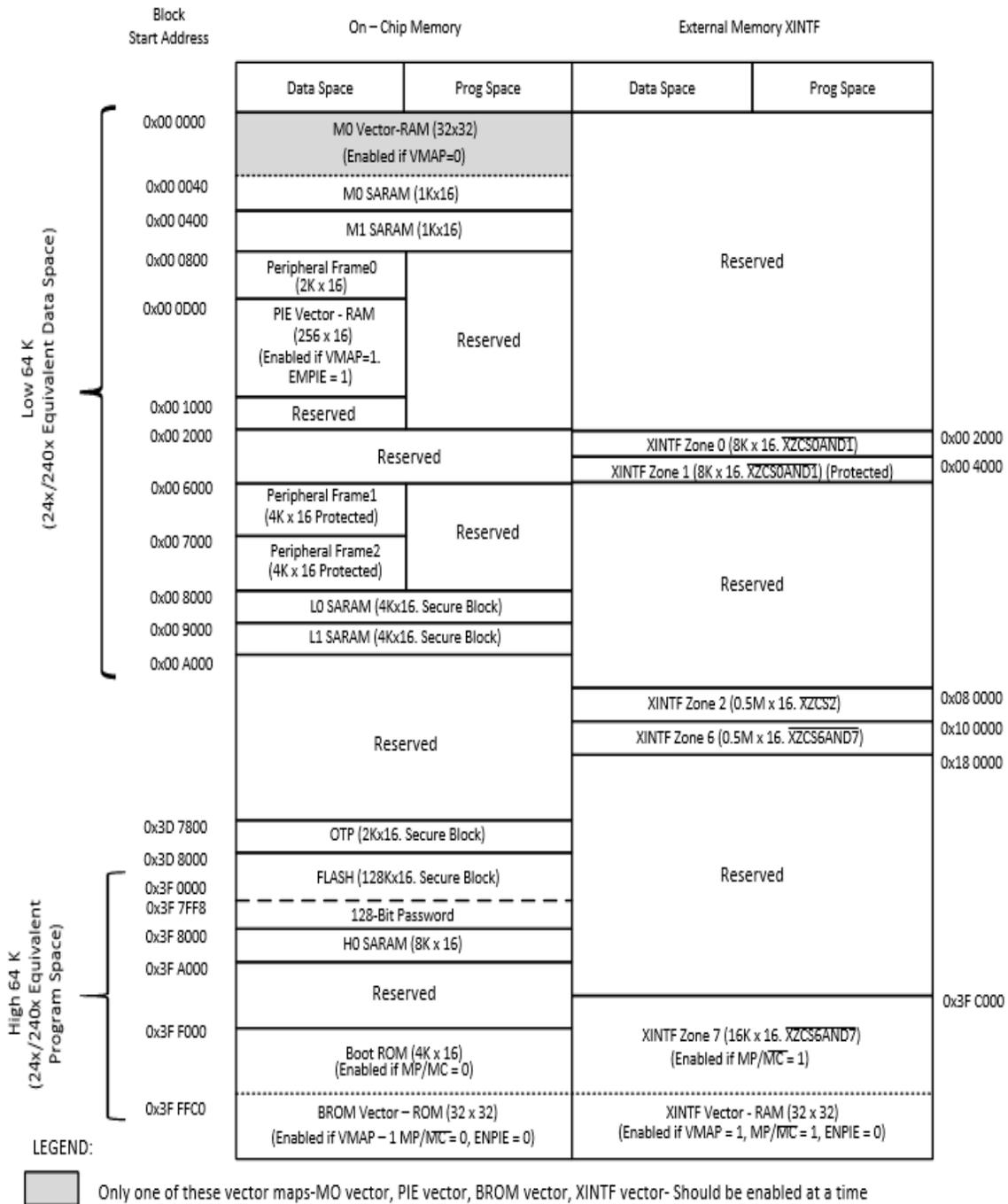


Figura 4.9 Distribución del Mapa de memoria.

La tarjeta eZdsp F2812 posee 6 conectores. El pin 1 de cada uno se identifica por una soldadura en forma cuadrada en la pista. La identificación de cada uno de los conectores se especifica en la tabla 4.1.

Conector	Función
P1	Interfaz JTAG
P2	Interfaz de Expansión
P3	Puerto Paralelo/ Interfaz de Control JTAG
P4/P8/P7	Interfaz de I/O
P5/P9	Interfaz Analógica
P6	Conector de poder

Tabla 4.1. Lista de Conectores de la eZdsp D2812

La disposición física de los 6 conectores se muestra en la figura 4.10.

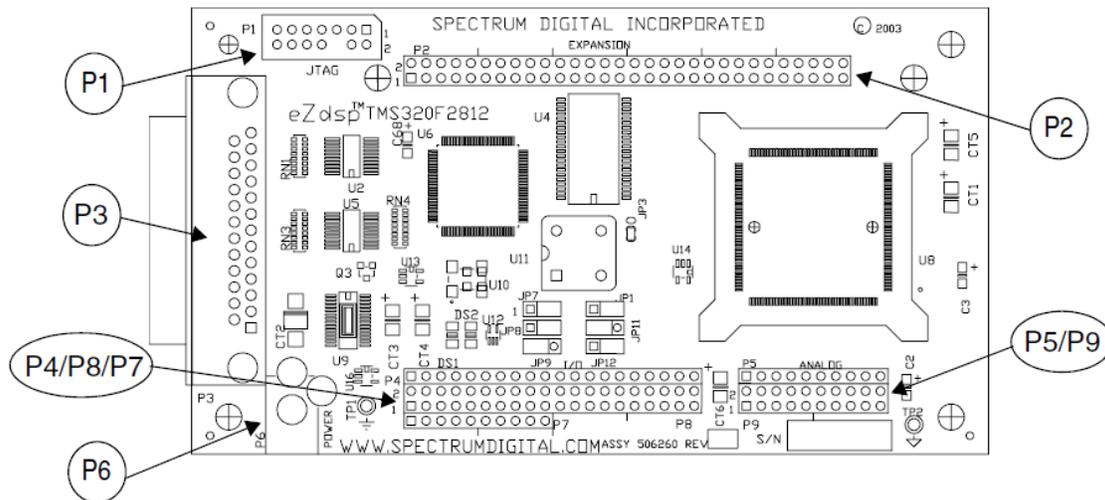


Figura 4.10 Disposición de los conectores en la placa eZdsp F2812.

#### 4.7 Conectores de la eZdsp F2812

La descripción de los diferentes conectores se presenta a continuación:

##### 4.7.1 P1, Interfaz JTAG.

Esta es una interfaz estándar, de 14 pines, utilizada por los emuladores JTAG para la interacción con las DSP de la Texas Instrument. En la siguiente figura se muestra la disposición de los 14 pines vistos desde la parte superior de la eZdsp.

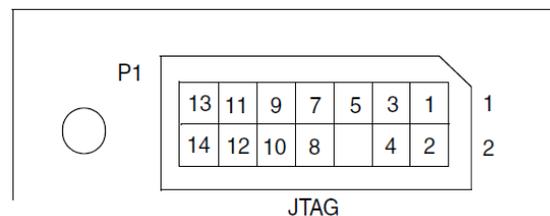


Figura 4.11 Distribución de pines para el conector P1.

Las señales de cada pin del conector P1 se muestran en la tabla 4.2.

Pin	Señal	Pin	Señal
1	TMS	8	GND
2	TRST-	9	TCK-RET
3	TDI	10	GND
4	GND	11	TCK
5	PD (+5V)	12	GND
6	NO PIN	13	EMUO
7	TDO	14	EMU1

Tabla 4.2. Nombre de los pines del conector P1.

#### 4.7.2 P2, Interfaz de expansión.

El conector P2 es un conector de 60 pines, la distribución de dichos pines se muestra en la figura 4.12, vistos desde la parte superior de la eZdsp.

P2

2	4	6	8	10	12	14	16	18	20	22	24	26	28	30	32	34	36	38	40	42	44	46	48	50	52	54	56	58	60
1	3	5	7	9	11	13	15	17	19	21	23	25	27	29	31	33	35	37	39	41	43	45	47	49	51	53	55	57	59

Figura 4.12 Distribución de pines para el conector P2.

La especificación de cada pin del conector P2, se detalla en la tabla 4.3.

Pin	Señal	Pin	Señal	Pin	Señal
1	+3.3V/+5V/NC	21	XA2	41	XRnW
2	+3.3V/+5V/NC	22	XA3	42	10K Pull-up
3	XA0	23	XA4	43	XWE
4	XA1	24	XA5	44	XRn
5	XA2	25	XA6	45	+3.3V
6	XA3	26	XA7	46	XNMI/INT13
7	XA4	27	XA8	47	XRSn/RSn
8	XA5	28	XA9	48	No Conectar
9	XA6	29	XA10	49	GND
10	XA7	30	XA11	50	GND
11	XA8	31	XA12	51	GND
12	XA9	32	XA13	52	GND
13	XA10	33	XA14	53	XA16
14	XA11	34	XA15	54	XA17
15	XA12	35	GND	55	XA18
16	XA13	36	GND	56	XHOLDn
17	XA14	37	XZCS0AND1n	57	XHOLDAn
18	XA15	38	XZCS2n	58	No Conectar
19	XA0	39	XREADY	59	No Conectar
20	XA1	40	10K Pull-up	60	No Conectar

Tabla 4.3. Nombre de los pines del conector P2.

#### 4.7.3 P3, Interfaz de Puerto Paralelo/JTAG

La tarjeta eZdsp F2812 posee un dispositivo de interfaz de puerto paralelo estándar, como el que se muestra en la figura 4.13. El cual admite comunicaciones ECP, EPP y SPP8/bidireccional. Este dispositivo tiene acceso a la interfaz integrada JTAG.



Figura 4.13 Distribución de los pines del conector P3.

#### 4.7.4 P4/P8/P7, Interfaz de I/O

El conector dispone de señales de Entrada/Salida (I/O) desde la DSP. La ubicación de los pines se muestra en la figura 4.14.

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	P4
2	4	6	8	10	12	14	16	18	20	22	24	26	28	30	32	34	36	38	40	
1	3	5	7	9	11	13	15	17	19	21	23	25	27	29	31	33	35	37	39	P8
1	2	3	4	5	6	7	8	9	10	P7										

Figura 4.14 Distribución de los pines del conector P4/P8/P7.

La especificación de cada pin del conector P4/P8/P7, se detalla en la tabla 4.4.

P4		P8				P7	
Pin	Señal	Pin	Señal	Pin	Señal	Pin	Señal
1	+3.3V/+5V/NC	1	+3.3V/+5V/NC	21	No Conectar	1	C1TRIPn
2	XINT2/ADCSOC	2	+3.3V/+5V/NC	22	XINT1N/XBIO <sub>n</sub>	2	C2TRIPn
3	MCLKXA	3	SCITXDA	23	SPISIMOA	3	C3TRIPn
4	MCLKRA	4	SCIRXDA	24	SPISOMIA	4	T2CTRIPI <sub>n</sub> /EVASOC <sub>n</sub>
5	MFSXA	5	XINT1 <sub>n</sub> /XBIO <sub>n</sub>	25	SPICLKA	5	C4TRIPn
6	MFSRA	6	CAP1/QEP1	26	SPISTEA	6	C5TRIPn
7	MDXA	7	CAP2/QEP2	27	CANTXA	7	C6TRIPn
8	MDRA	8	CAP3/QEP1	28	CANRXA	8	T4CTRIPI <sub>n</sub> /EVBSOC <sub>n</sub>

9	No Conectar	9	PWM1	29	XCLKOUT	9	No Conectar
10	GND	10	PWM2	30	PWM7	10	GND
11	CAP5/QEP4	11	PWM3	31	PWM8		
12	CAP6/QEPI2	12	PWM4	32	PWM9		
13	T3PWM/T3CMP	13	PWM5	33	PWM10		
14	T4PWM/T4CMP	14	PWM6	34	PWM11		
15	TDIRB	15	T1PWM/T1CMP	35	PWM12		
16	TCLKINB	16	T2PWM/T2CMP	36	CAP4/QEP3		
17	XF/XPLLDISn	17	TDIRA	37	T1CTRIP/PDPINTAn		
18	SCITXDB	18	TCLKINA	38	T3CTRIP/PDPINTBn		
19	SCIRXDB	19	GND	39	GND		
20	GND	20	GND	40	GND		

Tabla 4.4. Nombre de los pines del conector P4/P8/P7.

#### 4.7.5 P5/P9 Interfaz Analógica.

La disposición de los 30 pines del conector P5/P9 se muestran en la figura 4.15.

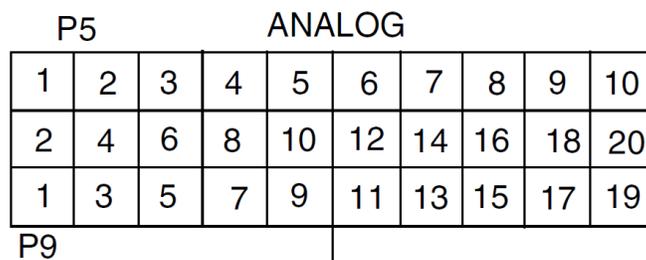


Figura 4.15 Distribución de los pines del conector P5/P9.

La especificación de cada pin del conector P5/P9, se detalla en la tabla 4.5.

P5		P9			
Pin	Señal	Pin	Señal	Pin	Señal
1	ADCINB0	1	GND	11	GND
2	ADCINB1	2	ADCINA0	12	ADCINA5
3	ADCINB2	3	GND	13	GND
4	ADCINB3	4	ADCINA1	14	ADCINA6
5	ADCINB4	5	GND	15	GND
6	ADCINB5	6	ADCINA2	16	ADCINA7
7	ADCINB6	7	GND	17	GND
8	ADCINB7	8	ADCINA3	18	VREFLO
9	ADCREFM	9	GND	19	GND
10	ADCREFP	10	ADCINA4	20	No Conectar

Tabla 4.5 Nombre de los pines del conector P5/P9.

El pin VREFLO debe de conectarse a GND para el correcto funcionamiento del convertidor analógico digital. La tarjeta está equipada de 16 canales de conversión de análogo a digital. Cada canal tiene una resolución de 12 bit y un tiempo de conversión de 80 ns. La máxima frecuencia de muestreo es de 25 MHz. La señal de entrada debe estar entre 0-3V. Sin embargo, a través de un circuito dedicado de interfaz se logra el rango de -15 V a 15 V.

#### 4.7.6 P6, Conector de Poder

La alimentación a la tarjeta eZdsp F2812se realiza por medio del conector P6. El conector tiene un diámetro exterior de 5.5mm y un diámetro interno de 2 mm. En la figura 4.16 se muestra un diagrama del conector P6.

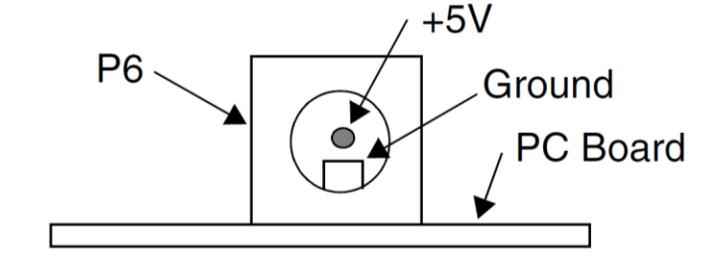


Figura 4.16 Diagrama del conector P6.

#### 4.8 Implementacion de la DSP (eZdsp F2812) en el sistema de control.

Para la implementacion de la DSP en el sistema de control es necesario realizar la medicion de las variables requeridas, mostradas en el capitulo 3, dichas señales deberan ser acondicionadas para que la eZdsp F2812 pueda procesarlas y emitir una señal de control, que modifique las características del sistema, para obtener el comportamiento deseado. El proceso anteriormente descrito se muestra en la figura 4.17.

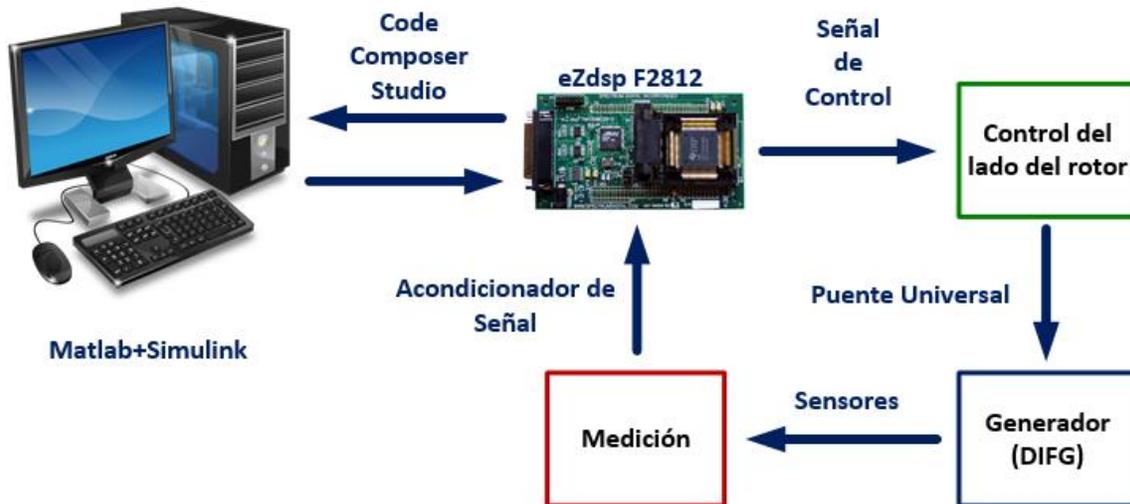


Figura 4.17 Implementación de la DSP en el sistema de control.

En el diagrama de bloques se muestran los principales elementos que interactúan para el control digital en escala real, estos son:

- El computador, que será el encargado de mostrar el comportamiento de las variables medidas de la máquina de inducción, interactuar con la DSP, para modificar las acciones de control, registrar eventos, etc. Y por medio de Matlab y Simulink generar el código a implementar por la DSP.
- La tarjeta de procesamiento de señales (DSP), en este caso la eZdsp F2812 de Texas Instruments, cuya misión será la de censar las variables de la máquina de inducción, procesar dichas señales y emitir una señal de control, la cual se calcula en base a las condiciones óptimas de la máquina.
- El control del lado del rotor y el generador han sido explicados en el capítulo 3.
- La medición de las variables de la maquina se realizaran con los dispositivos (sensores) diseñados para ello, los cuales se explicaran más adelante, y para el acondicionamiento de las señales se realizara una interfaz.

Para la interacción de la computadora y la DSP, Texas Instruments posee un entorno de desarrollo integrado (IDE) llamado Code Composer Studio, el cual permite la compilación, ejecución, verificación, simulación, depuración y carga del código a ejecutar por la DSP.

Esta interacción permite el monitoreo en tiempo real de la operación de la turbina eólica.

Una de las partes importantes del diagrama mostrado en la figura 4.17 es la parte de la medición y acondicionamiento de señales.

#### 4.8.1 Medición y acondicionamiento de la señal.

Las variables que son necesarias medir son

- Velocidad ( $m/s$ ) y Dirección del viento ( $^{\circ}$ )
- Velocidad ( $rpm$ ) y Posición angular del rotor ( $rad$ )
- Corriente del devanado del rotor ( $A$ )
- Corriente del devanado del estator ( $A$ )

Para que la tarjeta eZdsp pueda conocer las variables, se deben instalar sensores que envíen las señales a las entradas analógicas-digitales que tiene disponible la tarjeta.

- **Velocidad y dirección del viento.**

El viento se considera como una cantidad vectorial, ya que nos importa estudiar tanto su magnitud como su dirección, estas dos variables son totalmente independientes entre sí. Los principales instrumentos para la obtención de estas variables son.

Para la velocidad el anemómetro rotativo de cubeta o el anemómetro de hélice. Ambos tipos de anemómetros constan de un sensor y un transductor. El sensor es el dispositivo que rota por acción de la fuerza del viento. El transductor es el que genera la señal eléctrica que representa la velocidad medida. La salida es una señal analógica de voltaje, proporcional a la velocidad instantánea en metros por segundo ( $m/s$ ). El anemómetro de cubeta se muestra en la figura 4.18.



Figura 4.18 Anemómetro de cubeta.

Con este instrumento se facilita la lectura de datos, ya que trae incorporado el transductor que arroja una salida lineal y que solo es necesario ingresar el voltaje de salida del sensor en una entrada análoga (ADC) de la tarjeta procesadora de señales.

Para la dirección del viento, en los sistemas de generación con orientación activa, se pueden utilizar una clásica veleta industrial, de paletas separadas o paletas de viento típicas, tal como se muestra en la figura 4.19.



Figura 4.19 Veleta de paletas separadas (Izq.) y Veleta de paleta típica (Der).

De igual forma que la adquisición de la variable del viento, existen varios métodos para obtener la posición del viento, en este caso podríamos acoplar un potenciómetro, y este sería el transductor que se ocuparía para la generación de señal. Los potenciómetros son unos dispositivos capaces de medir la posición angular y pequeños desplazamientos de posición lineal. Su principio básico de funcionamiento se basa en obtener una relación entre la diferencia de potencial que se crea entre sus bornes de alimentación y la posición angular en la que se encuentra la veleta.

- **Velocidad y posición angular del rotor.**

La variable de salida en el lazo de control es la velocidad de rotación del eje, medida en Revoluciones por Minuto (RPM), esta también es la referencia del lazo cerrado, luego es la variable de mayor interés

en el sistema. Para obtener el valor de las RPM se puede implementar un Encoder digital. El Encoder envía una señal de respuesta que puede ser utilizado para determinar la posición, contar, velocidad o dirección, en este caso la velocidad. Un dispositivo de control puede usar esta información para enviar un comando para una función particular. El sensor usado para este fin suele constar de un transmisor y receptor óptico compatible con señales TTL, lo que simplifica el circuito de acondicionamiento.

En el sistema físico las variables de referencia de las velocidades se introducirán a la DSP por medio de fuentes de voltaje con un valor proporcional a la velocidad requerida.

- **Corriente de los devanados del rotor y estator (A)**

Como se explicó anteriormente en la descripción de la maquina DFIG, que el circuito del rotor esta accesible para el usuario, el dispositivo más utilizado para medir la corriente es el amperímetro o en su defecto un sensor de estado sólido, también la medición puede realizarse por medio de transformadores de corriente como el que se muestra en la figura 4.20.



Figura 4.20 Transformadores de corriente para medición en el rotor y estator.

Debido a las diferentes escalas que pueden presentarse por los sensores la interfaz de la DSP tendrá que diseñarse de modo que las señales puedan trabajar en una misma base, y cada una de las magnitudes medidas tendrá que conectarse a una de las entradas analógico digital y las salidas PWM de la DSP, a las entradas de la VSC como se mostró en el capítulo 3. El número de cada entrada y salida a utilizar de la DSP tendrá que tenerse en cuenta ya que al momento de implementarla en Simulink será necesaria introducirla en los bloques RTDX.

#### **4.9 Programación de la eZdsp f2812 en Matlab/Simulink**

Como se mencionó en el capítulo 2, Matlab es un software de cálculo matemático que posee múltiples herramientas, para el modelado de sistemas entre ellas se encuentra Simulink, dicha herramienta permite una programación por medio de bloques.

Uno de los softwares más importantes para la comunicación entre la computadora y la DSP es el Code Composer Studio, en el anexo 2 se presenta el contenido del kit para la eZdsp F2812, así como los requerimientos necesarios en la computadora, y parte de la configuración básica para el uso y la creación de una interfaz que permita visualizar las acciones que realiza la DSP en tiempo real.

#### 4.9.1 Code Composer Studio (CCS)

Después de la creación del código fuente utilizando el Real Time Workshop para que este pueda ser cargado y ejecutado por la DSP, es necesario el uso del software denominado Code Composer Studio (CCS) de Texas Instruments, este compila un proyecto en CCS a partir del código C generado del modelo Simulink.

El software Code Composer Studio es creado por Texas Instruments como una herramienta de desarrollo para las DSP fabricadas por ellos mismos y es el encargado de la depuración, ejecución y compilación del código fuente creado por RTW. La interfaz de usuario del programa se muestra en la figura 4.21, donde se pueden distinguir algunas barras de herramientas importantes.

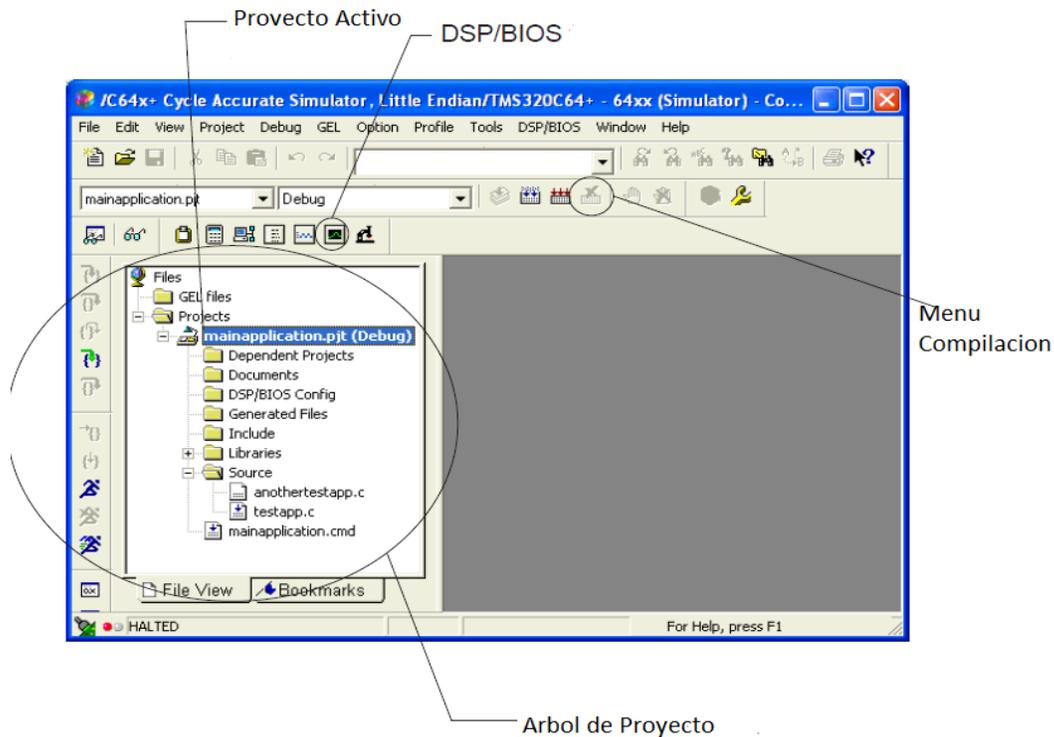


Figura 4.21 Interfaz de usuario del CCS 3.1

Algunos iconos importantes dentro del programa se pueden observar junto con su descripción en la figura 4.22.



Figura 4.22 Descripción de botones del CCS 3.1.

Para compilar un proyecto o un archivo fuente individual existen varias opciones disponibles; desde el menú de proyecto o desde los iconos en la barra de compilación.

<i>Compilar:</i>	Compila un archivo individual.
<i>Compilar incrementalmente:</i>	Compila solamente los archivos que han cambiado.
<i>Compilar Todo:</i>	Compila todo el proyecto.

El resultado de la compilación aparece en la pantalla de depuración en la parte inferior de la pantalla y notifica al usuario si hubiera errores o advertencias.

Cuando el programa ha sido generado o compilado, es necesario cargarlo a la DSP. Esto se puede hacer seleccionando “Cargar Programa” desde el menú archivo(o alternativamente, “Recargar Programa” si previamente se había cargado el mismo proyecto).

Al cargar un programa se genera un archivo de salida, visible en el menú proyecto, el cual es guardado en la carpeta llamada Debug. Cuando el programa es cargado el código en ensamblador se abre automáticamente.

Algunos comandos muy útiles para el software son:

<i>F5:</i>	Ejecuta el programa.
<i>Shift-F5:</i>	Detiene la ejecución del programa.
<i>F8:</i>	Paso a través del programa (útil para depurar programas).

#### **4.9.2 Embedded IDE Link.**

Embedded IDE Link conecta a Matlab Simulink con el software para ambientes de desarrollo. Esta herramienta permite generar, ensamblar, probar y optimizar código para prototipado o producción. El automatiza la depuración, la generación del proyecto y la verificación del código en procesadores o simuladores.

Se puede utilizar Embedded IDE Link con Real Time WorkShop para generar proyectos completos en ambientes de desarrollo integrados (IDE) soportados por la herramienta, estos se pueden utilizar para realizar una aplicación en tiempo real para los procesadores soportados. El proyecto generado no solo crea código C para la aplicación, sino que genera el software del marco de trabajo requerido para inicializar y manejar la ejecución del código de la aplicación del procesador en tiempo real.

Se puede utilizar Matlab para analizar y depurar el código interactivamente que se ejecutara en el procesador a través del IDE, y visualizar los resultados de la ejecución usando los gráficos del mismo Matlab.

Las principales características son:

- Depuración, verificación y análisis automatizado de código escrito o generado usando Matlab Simulink.
- Capacidad de ejecución en tiempo real.
- Configuración personalizada de mapeo de memoria para IDE específicas.
- Soporte para IDEs y procesadores de la siguientes vendedores: Altium, Analog Devices, Freescale, Renesas, STMicroelectronics y Texas Instruments.

Embedded IDE Link permite que se usen las funciones de Matlab para comunicarse con el software Code Composer Studio, explicado en la sección siguiente. Además, se hace fácil la verificación de código en Code Composer Studio usando un modelo creado en Simulink.

## RTDX

Real-Time Data Exchange (RTDX) de la Texas Instrument agrega la característica de tener visibilidad continua en el funcionamiento en modo real de la tarjeta DSP. RTDX permite transferir información entre la tarjeta destino y la computadora anfitrión sin interrumpir el funcionamiento de la tarjeta.

RTDX se utiliza con Embedded IDE Link y el Code Composer Studio para acelerar el desarrollo de sistemas para los procesadores de la familia C2000 de Texas Instruments. Por ejemplo, utilizando RTDX se puede:

- Mandar y recibir datos desde la memoria del procesador.
- Cambiar las características operativas del programa.
- Hacer cambios al algoritmo sin necesidad de detener el programa.

Es importante recordar la importancia de habilitar la interacción en tiempo real de un proceso o de uno algoritmo, y poder observar los resultados mientras se desarrollan.

## Bloques Embedded IDE Link

Entre los bloques a destacar de la librería de este toolbox se encuentran los siguientes:

- **ADC**

El bloque ADC maneja la configuración de cuales entradas tendrán la conversión análoga a digital activada. El usuario puede configurar el tiempo de muestreo así como el tipo de dato de salida preferido. Además, existen opciones para cuales módulos inicializar. El número de entradas puede estar entre 1 a 16. El bloque de Simulink se muestra en la figura 4.23.

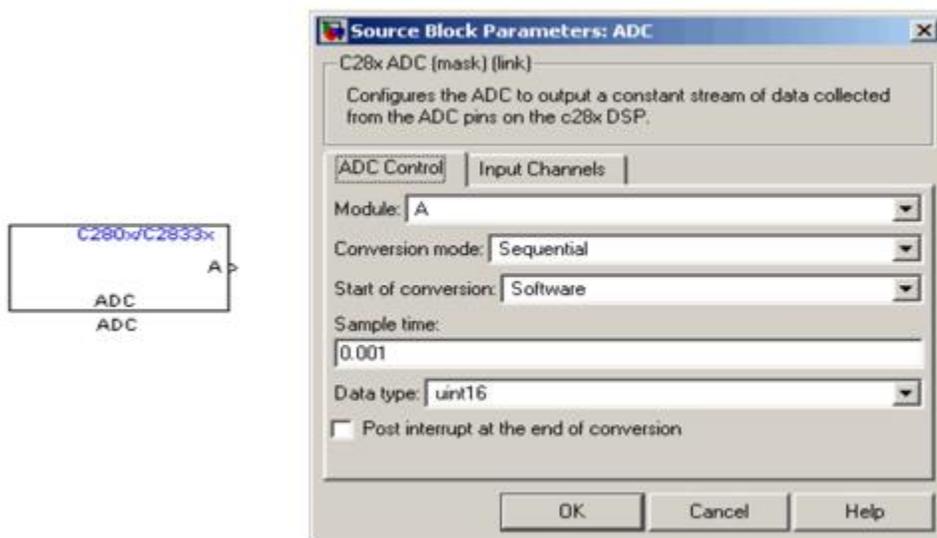


Figura 4.23 Bloque ADC en Simulink para la DSP F2812.

- **PWM**

A diferencia del CCS (Code Composer Studio), configurar una señal PWM en Simulink es muy fácil. El bloque ofrece muchas opciones, por ejemplo, el periodo de la señal, si la señal es asimétrica o no, y la más importante cual salida habilitar.

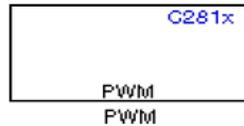


Figura 4.24 Bloque PWM en Simulink para la DSP F2812.

Muchas opciones dependiendo de la lógica de control están disponibles, dándole al usuario control total sobre como la señal resultante debe comportarse. Se debe tener en cuenta que todas las entradas al bloque PWM deben ser valores escalares.

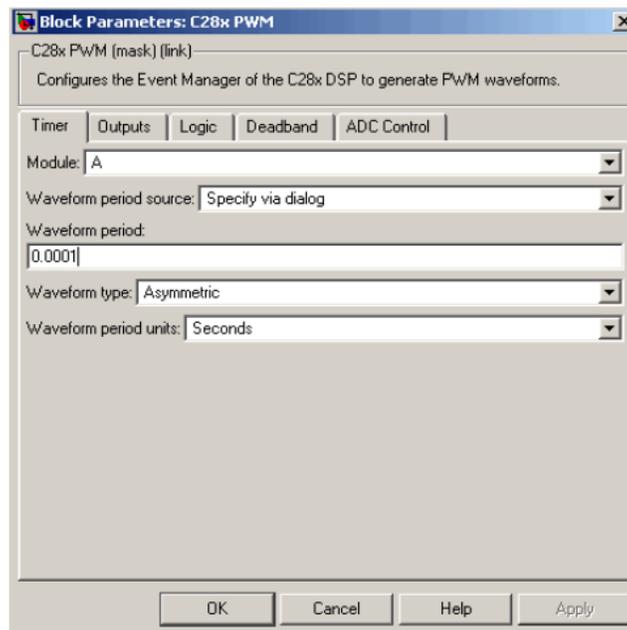


Figura 4.25 Opciones bloque PWM en Simulink.

- **Bloques RTDX**

Otro bloque importante incluido en la librería Embedded target for TI C2000 DSP son los bloques RTDX que proveen comunicación en tiempo real entre Matlab y la tarjeta, además, tienen un gran potencial y son de suma importancia para el desarrollo de los controladores de velocidad y posición. En la figura 4.26. Se muestran los bloques de Simulink.

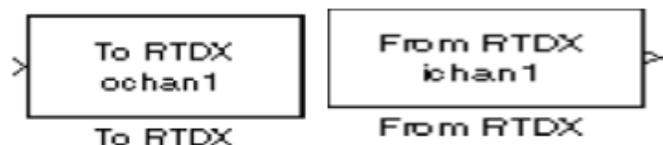


Figura 4.26 Bloques RTDX en Simulink.

## From RTDX

Agrega un canal de entrada RTDX. Cuando se genera código desde Simulink utilizando Real Time Workshop y este incluye un bloque "From RTDX", la generación del código agrega los comandos en C para crear un canal de entrada RTDX. Los canales de entrada transfieren datos desde la computadora hacia la tarjeta. El comando que se incluye en el archivo C es:

`RTDX_enableInput(&channelname);` donde `&channelname` es el nombre asignado al canal de entrada.

Para usar bloques RTDX en un modelo, se debe descargar y correr el modelo en la tarjeta destino, se deben habilitar los canales RTDX desde Matlab o en el menú de dialogo del bloque se marca "Enable RTDX channel on start up". Las opciones disponibles en el menú de dialogo de bloque se detallan a continuación en la figura 4.27.

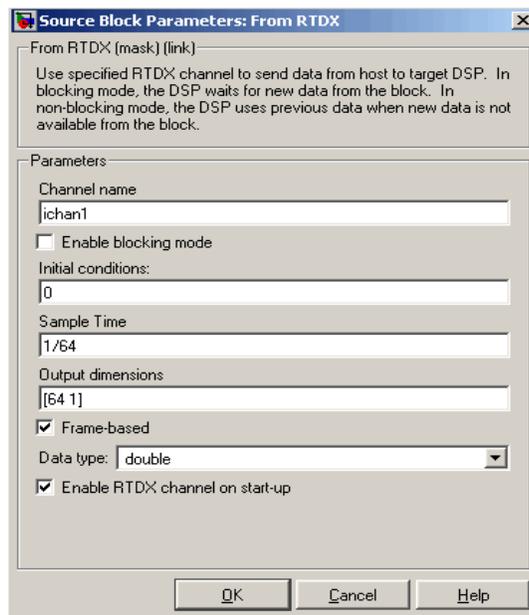


Figura 4.27 Opciones bloque "From RTDX" en Simulink.

**Channel Name:** Nombre del canal de entrada que se creara por el código generado.

**Enable blocking mode:** El modo de bloqueo instruye al procesador a pausar el proceso hasta que nueva información esté disponible del bloque RTDX. Si se habilita y nuevos datos no están disponibles cuando el procesador lo requiera el proceso se detiene.

**Initial Conditions:** Es el dato que el procesador lee del el bloque RTDX en la primera lectura. Dejando esta opción en blanco causa un error en Real Time Workshop.

**Sample Time:** Tiempo entre muestras de la señal. Por defecto es 1 segundo.

**Output dimensión:** Dimensiones de la matriz para la señal de salida del bloque. El primer valor es el número de filas y el segundo el número de columnas.

**Data Type:** Tipo de datos que viene del bloque. Se puede elegir entre "double", "single", "Uint8", "Int16", "Int32".

**Enable RTDX channel on start-up:** Se habilita el canal RTDX cuando se inicializa el canal desde Matlab. El canal siempre se debe de abrir en Matlab.

### To RTDX

Agrega un canal de salida RTDX. Cuando se genera código desde Simulink utilizando Real Time Workshop y este incluye un bloque "From RTDX", la generación del código agrega los comandos en C para crear un canal de salida RTDX. Los canales de salida transfieren datos desde la tarjeta hacia la computadora. El comando que se incluye en el archivo C es:

*RTDX\_enableOutput(&channelname);* donde *&channelname* es el nombre asignado al canal de entrada.

Para usar bloques RTDX en un modelo, se debe descargar y correr el modelo en la tarjeta destino, se deben habilitar los canales RTDX desde Matlab o en el menú de dialogo del bloque se marca "Enable RTDX channel on start up". Las opciones disponibles en el menú de dialogo de bloque se detallan a continuación en la figura 4.28.



Figura 4.28 Opciones bloque "To RTDX" en Simulink.

**Channel Name:** Nombre del canal de salida que se crea por el código generado.

**Enable blocking mode:** En el modo de bloqueo, la escritura de datos se suspende mientras el canal RTDX está ocupado, esto es, cuando datos están siendo escritos en cualquier dirección.

**Enable RTDX channel on start-up:** Se habilita el canal RTDX cuando se inicializa el canal desde Matlab. El canal siempre se debe de abrir en Matlab.

Por último se muestra el diagrama de flujo entre la interacción de código fuente entre Matlab/Simulink y la tarjeta eZdsp F2812 a través del Code Composer Studio en la figura 4.29.

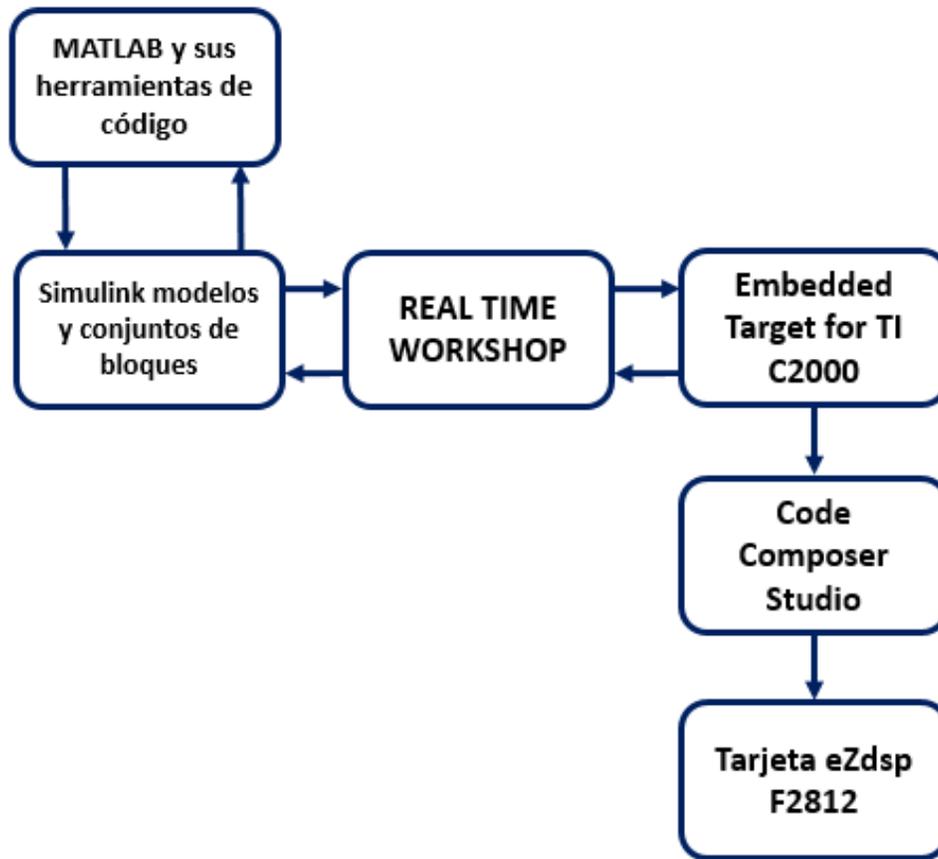


Figura 4.29 Diagrama de flujo del proceso del modelo de Simulink con la tarjeta eZdsp.

En los primeros bloques se encuentra Matlab con Simulink y RTW para crear el código fuente del modelo del sistema físico en cuestión en este caso del controlador digital de la turbina eólica, pero antes se debe modificar la configuración de parámetros como lo explica la figura 4.26.

Luego RTW se encarga de hacer este proceso rápido por medio del acelerador de simulaciones, característica propia de RTW y de la configuración de parámetros. Una vez hecho el modelo, se pasa al siguiente bloque que es el Embedded Target for TI C2000 DSP, este se encarga de prototipar el modelo que se trabajó y además este se encarga de comunicarse con el CCS y utilizar información almacenada en memoria y registros del DSP.

El CCS se encarga de compilar el código fuente dentro de la DSP y a su vez utiliza RTW para generar el código ejecutable para la DSP. En el último bloque se encuentra la DSP donde se descarga el código ejecutable.

### 4.9.3 Generación de Código a partir del sistema de control implementado.

Para generar código ejecutable a partir del modelo en Simulink, nos auxiliamos de la herramienta Real Time Workshop descrita en el capítulo II. Para ello se hacen modificaciones en el diagrama de bloques del sistema de control de la figura 3.1, se elimina la parte del modelo Aerodinámico y Mecánico-Eléctrico de la turbina eólica; así que la generación de código solamente se hace con el bloque Control DSP.

A este bloque se le hacen modificaciones de modo a que las entradas al sistema (DSP) sean las variables medidas en la máquina de inducción (corriente, tensiones, velocidad, etc.) y las salidas sean la señal de control mediante los módulos PWM de la tarjeta. El diagrama de control queda como la figura 4.30.

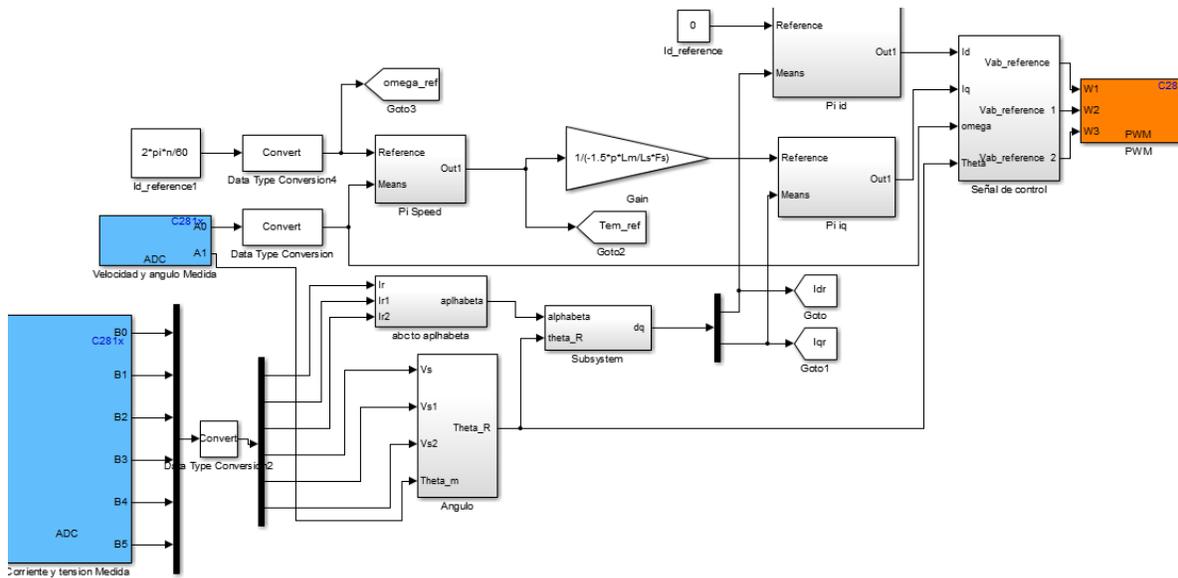


Figura 4.30 Modificación del sistema de control DSP para obtención de código ejecutable.

Como se ve ya no forma parte del modelo las ecuaciones que describen a la turbina eólica, ni si quiera su parte de control de potencia (Puente universal), ya que si se pretende implementar en un sistema real, la única parte de control en la simulación la hacen los controladores PI. A partir de este modelo en Simulink se pretende cargar el código Ejecutable en la tarjeta objetivo (eZdsp F2812).

Para las entradas de este sistema se colocaron de la librería Embedded target for TI C2000 DSP, los bloques ADC de la familia de DSP's C281x. los que se muestran en la figura 4.31.

Estos bloques son para definir los canales de entrada de la tarjeta objetivo, que tendrán activada la conversión análoga digital. Para nuestro caso Dado que se va a medir tensión en el estator y corriente en el rotor, estas por ser trifásicas necesitan tener 3 canales activados para la conversión de cada variable. Es por ello que en la figura se muestran 6 canales en total, que van directo al control que se encarga de las transformadas de Clarke y de Park.

En los parámetros que se pueden definir en este tipo de bloques ADC son parecidos a los ya descritos anteriormente, solo que con ciertas variantes. Dado que la tarjeta eZdsp F2812 cuenta con 16 canales de conversión ADC, se puede escoger entre ellos cuales serán utilizados.

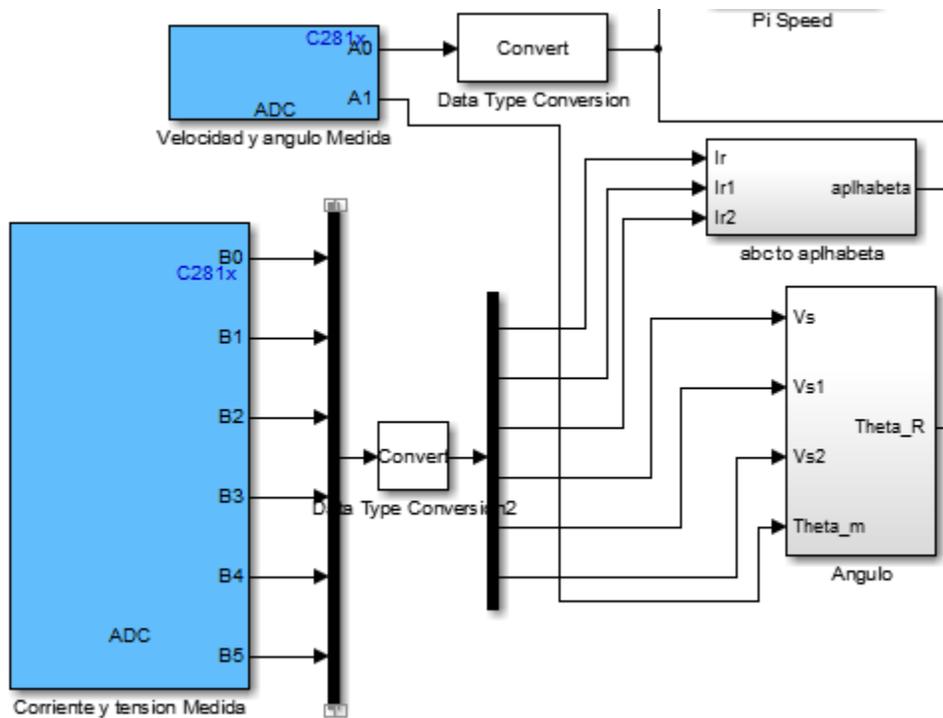


Figura 4.31 Entradas al sistema de control Bloques ADC.

En la figura 4.32 se muestran los parámetros definidos en el bloque ADC.

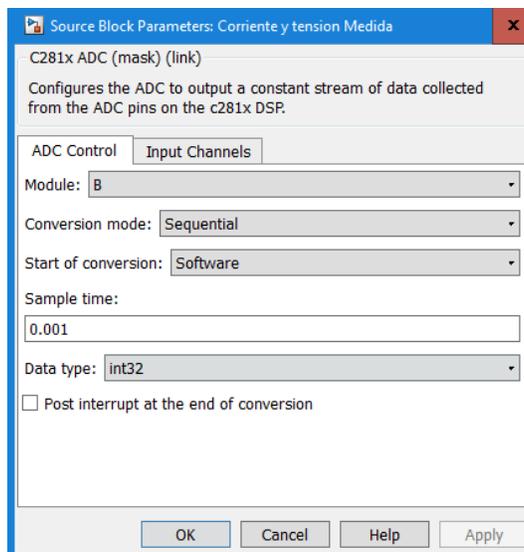


Figura 4.32 Definición de parámetros en el bloque C281x ADC.

**Modulo:** Especifica que módulo de la DSP se utilizará.

- A: Activa los canales ADC del módulo A (ADCINA0 hasta ADCINA7)
- B: Activa los canales ADC del módulo B (ADCINB0 hasta ADCINB7)

**Modo de conversión:** Especifica el tipo de muestreo a usar para las señales:

- Secuencial: Muestra el canal secuencialmente.
- Simultaneo: Muestra el correspondiente canal de los módulos A y B al mismo tiempo.

**Comienzo de la conversión:** Especifica el tipo de señal que activa la conversión

- Software: Señal desde el software
- EVA: Señal desde el Event Manager A
- EVB: Señal desde el Event Manager B
- External: Señal desde el exterior del Hardware

**Tiempo de muestreo:** El tiempo de muestreo que se desea para las entradas, esto está limitado por el tiempo o frecuencia de muestreo que tiene la tarjeta eZdsp F2812 que es de 25 MHz.

**Tipo de Dato:** Tipo de dato a la salida, los tipos son: auto, double, single, int8, uint8, int16, uint16, int32 or uint32.

**Señal de interrupción al final de una conversión:** Selecciona esta casilla si desea realizar una interrupción asíncrona al final de cada conversión de dato.

Siguiente al bloque de ADC se coloca un convertidor de tipo de datos, este se utiliza mayormente para trabajar datos iguales tanto en la entrada como en la salida o para cambiar el tipo de dato a la salida. En la figura 4.33 se muestra la configuración.

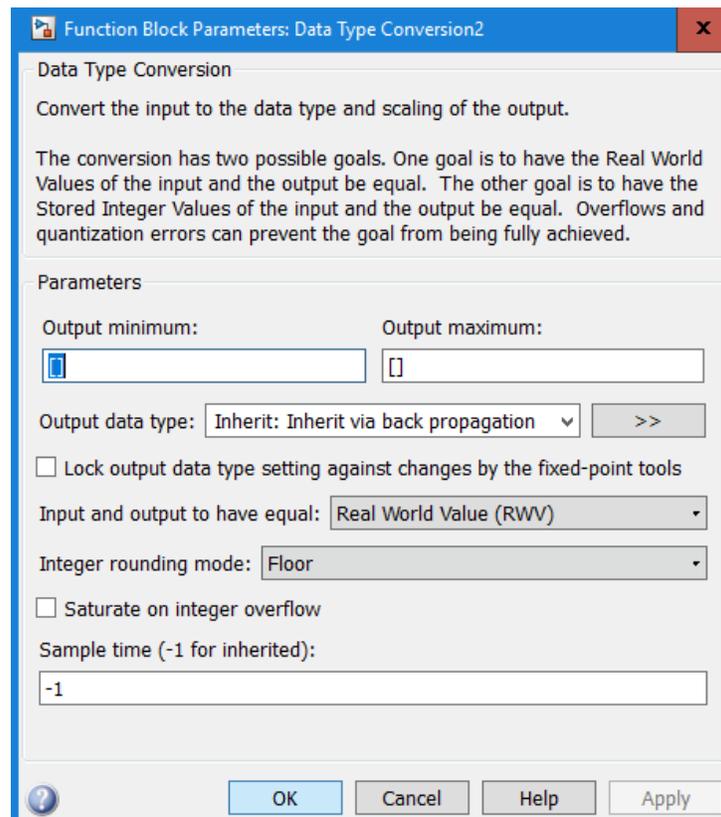


Figura 4.33 Parámetros del convertidor de tipo de datos.

Esta configuración es la misma para todos los bloques de este tipo en el sistema de control.

Por última modificación al sistema de control de la figura 3.1 son las salidas PWM de la librería Embedded Target for TI C2000. Al igual que las entradas de los ADC, el sistema real implementado necesita 3 variables, las cuales son variables de tensión y también por ello se ven 3 canales PWM utilizados en el bloque.

En las siguientes figuras se detallan los parámetros de este bloque.

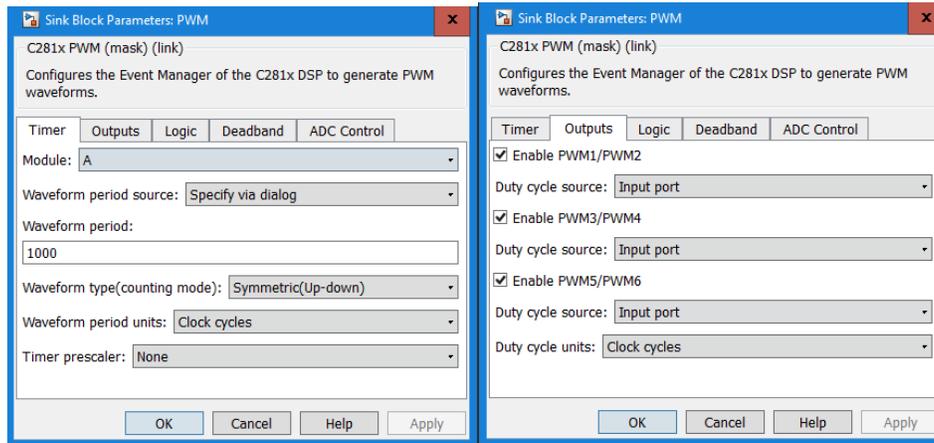


Figura 4.34 Pestañas Timer y Outputs del bloque PWM.

En la primera se configuran el tipo de pulsos PWM, periodo, etc. En la segunda se habilitan las salidas de la DSP.

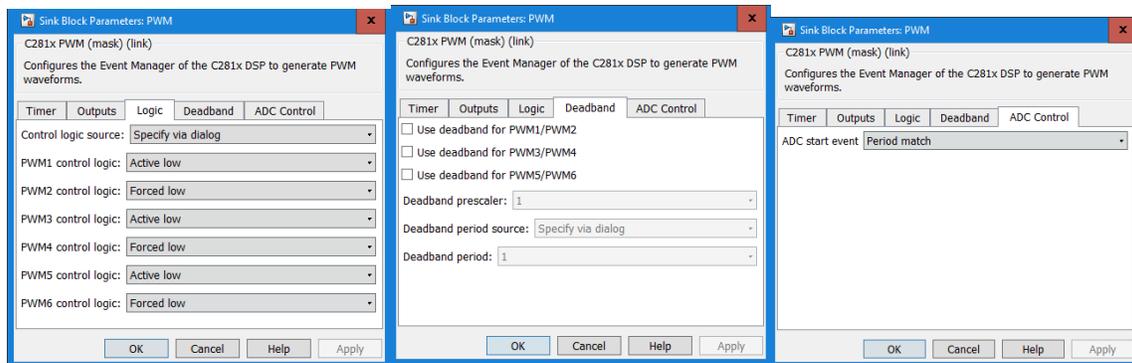


Figura 4.35 Pestañas Logic, DeadBand y ADC Control del bloque PWM.

La primera pestaña contiene la lógica de la generación PWM, la segunda especifica la banda muerta para los pulsos y la tercera se configura el evento de comienzo de la conversión ADC.

A partir del diagrama de bloques ya podemos crear el código que se cargara en la tarjeta objetivo, este código se puede ver en el anexo 3.

Para ello debemos disponer del IDE para la DSP F2812, este es el Code Composer Studio descrito anteriormente, este ambiente de desarrollo integrado nos permitirá mediante las librerías Embedded Target for TI C2000, cargar, depurar, compilar el código hecho por Real Time Workshop a la tarjeta.

Primeramente se configuran los parámetros de creación de código en RTW, este proceso lo podemos iniciar, tal como se muestra en la figura 4.36.

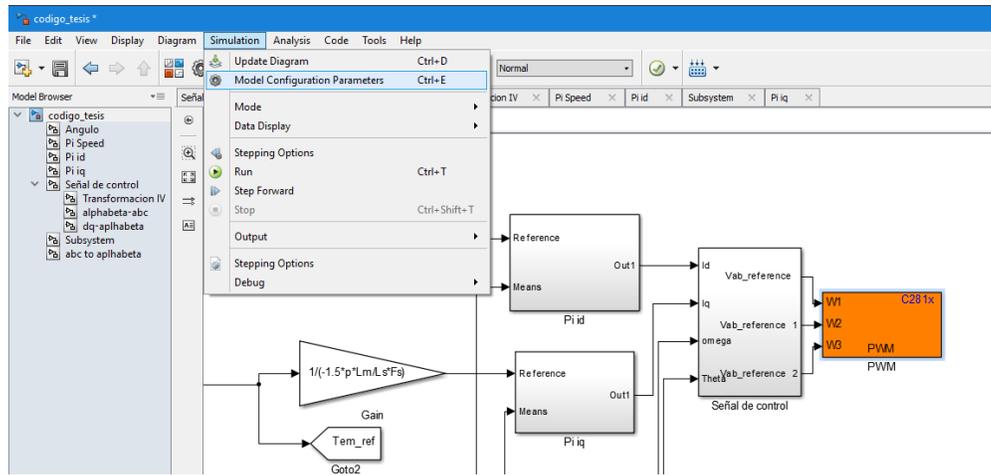


Figura 4.36 Pestaña para configurar los parámetros de generación de código.

Luego de seleccionar esa opción de la pestaña de simulation, nos aparecerá un cuadro de opciones como el de la figura 4.37, donde debemos modificar los parámetros de la pestaña “solver” porque por defecto la solución que trae es por “paso variable” y también “Continuous” para ello le cambiamos a “paso Fijo” y “discreto” respectivamente, ya que nuestro sistema se trabaja en tiempo discreto.

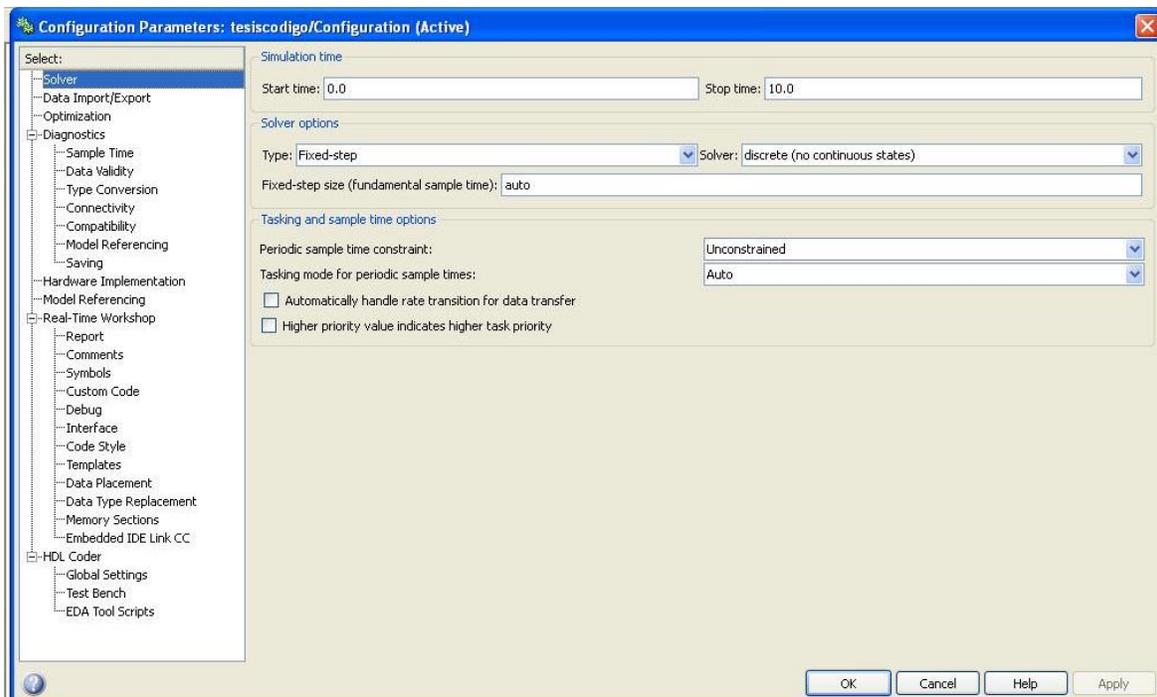


Figura 4.37 Configuración de la pestaña Solver para nuestra creación de código.

Después en la parte izquierda seleccionamos la pestaña Real-Time Workshop y nos sale el cuadro de parámetros siguiente, ver figura 4.38. En este ya podemos generar el código en un botón situado a la derecha donde dice "Generate code", por lo general los parámetros ya están diseñados para ser compilados, verificados y depurados para la tarjeta o microprocesador objetivo.

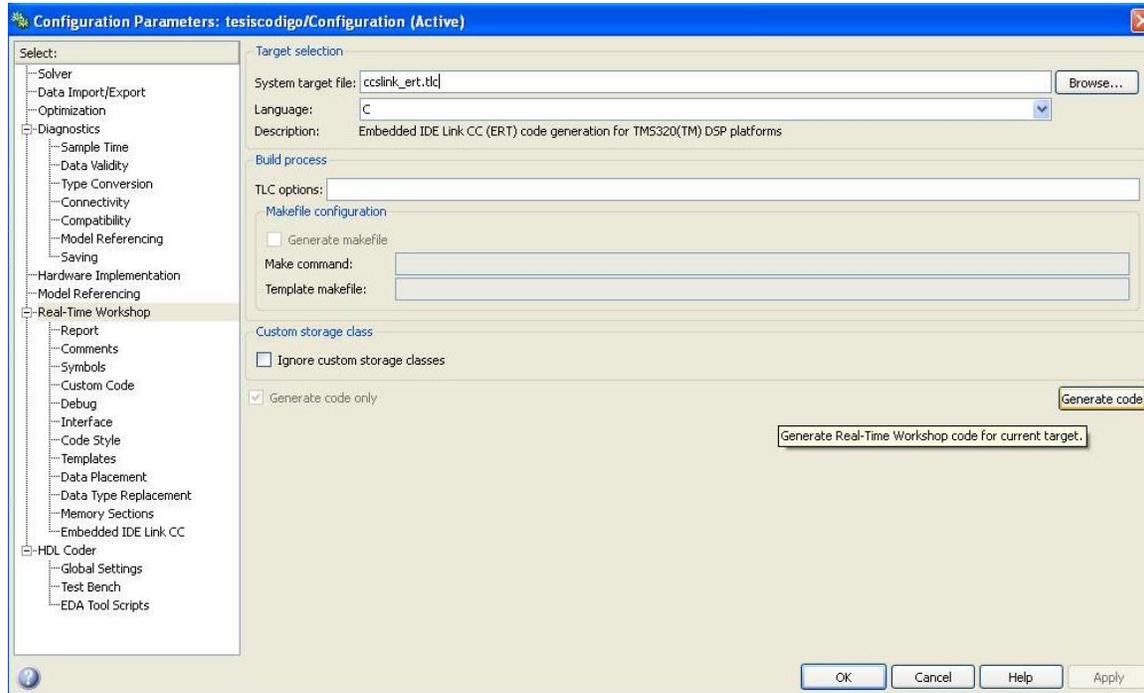


Figura 4.38 Generación de código en Real Time Workshop.

Un punto muy importante a la hora de generar código es el de tener Code Composer Studio abierto, ya que automáticamente después de seleccionar "Generate Code" este nos enlazara con CCS y nos creará un proyecto en CCS, con todos los archivos C, cabecera y archivos opcionales, más el ejecutable que será cargado en la tarjeta.

El proyecto creado a partir de la generación de código en CCS es el que se muestra en la figura 4.39. Como se observa en la columna izquierda, se encuentran todos los archivos .c y .h creados por RTW, además en la pestaña de abajo "build" nos muestra el proceso de depuración, verificación y compilación del código ejecutable creado. Como pueden ver ningún error, ni tampoco ninguna advertencia se mostró a la hora de compilar y ejecutar el archivo creado.

En la parte inferior izquierda se muestra un icono, que muestra Running este se debe a que ya está conectada la tarjeta eZdsp con la computadora y Code Composer Studio, esto se logra seleccionando la pestaña "debug" en el menú principal y luego la opción "Connect", con esto ya tenemos la comunicación entre la tarjeta objetivo, CCS y Simulink. Todos los archivos generados a partir de RTW se crean en el directorio de trabajo de Matlab, donde se puede tener acceso en caso de que se quiera observar una variable en específico o modificar parte del código.

En la figura 4.40 se muestran los archivos generados, más la dirección en la que se crearon y el ejecutable generado. Un aspecto importante para poder simular este código ejecutable, sería el de conseguir un emulador de DSP, cargar el .exe o .cmd en la tarjeta emuladora y así obtener un

resultado de simulación y verificación del código generado por RTW del sistema de control. No se pudo implementar debido a que no se encontró algún emulador de DSP capaz de ejecutar el archivo ejecutable. Una opción podría ser Proteus, pero se necesita de otro IDE parecido a CCS para poder cargar el .cmd en una tarjeta emuladora.

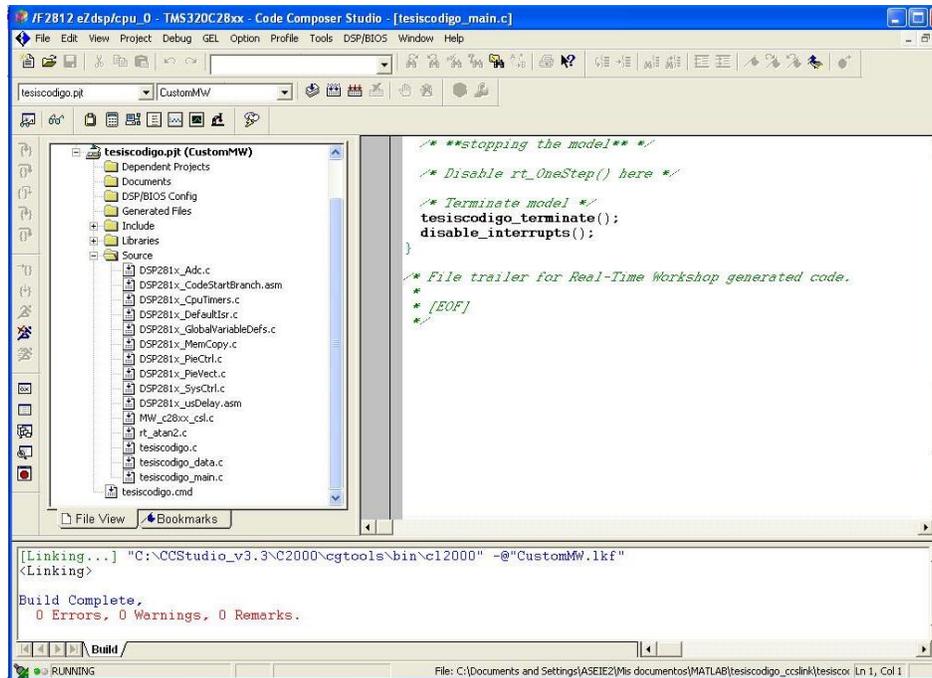


Figura 4.39 Proyecto creado a partir del código generado por Real Time Workshop en CCS.

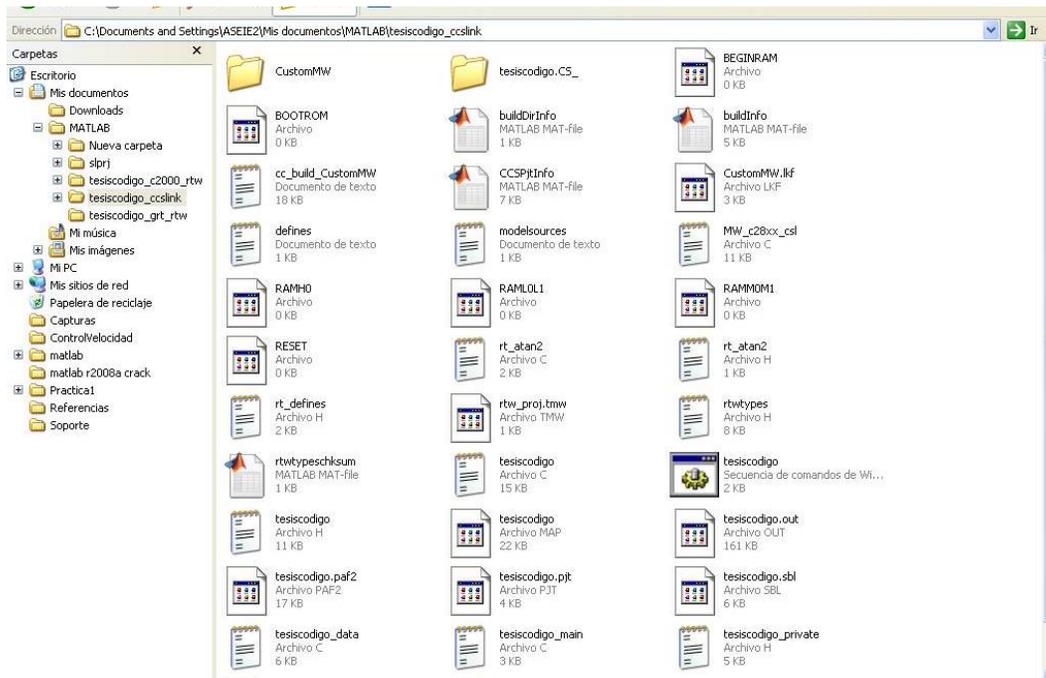


Figura 4.40 Archivos creados y ejecutable por RTW en el directorio de Matlab.

#### **Conclusiones del Capítulo 4.**

La implementación del control digital de la turbina eólica se puede llevar a cabo a través de diferentes variantes en la elección del equipo, todo dependerá para que fin y con qué exactitud se requieran los resultados de la implementación. Por ejemplo se puede cambiar la tarjeta eZdsp por un microcontrolador, pero estos tienen un nivel de respuesta más lento, lo que llevaría a una corrección de las variables medidas más atrasada y puede que con este retraso el sistema de control digital ya no responda como debe ser, en "tiempo real".

Caso contrario también puede suceder, que se puede elegir una DSP de alto rendimiento o velocidad de reloj de respuesta alta y esto puede sobrepasar los requerimientos, ya que se puede desperdiciar la capacidad de la DSP con el control del generador de la turbina, ya que tampoco necesita DSP de la familia C6000 que son de alto nivel de respuesta.

Para la generación del código por medio del RTW se utilizó Matlab 2008 y CCS 3.1. Para la implementar el controlador con DSP's modernas será necesario actualizar ambos software.

## Glosario

**CCS:** Code Composer Studio, software desarrollado por Texas Instrument para comunicar a la DSP con los elementos de Simulink.

**DFIG:** Doubly Fed Induction Generator en español Generador de Inducción Doblemente Alimentado, es una DFIM trabajando como generador.

**DFIM:** Doubly Fed Induction Machine en español Maquina de Inducción Doblemente Alimentado la cual puede ser usada como generador o como motor.

**DSP:** Digital Signal Processor en español, Procesador Digital de Señales.

**eZdsp:** DSP de la familia C2000 de Texas Instrument.

**HAWT:** Horizontal Axis Wind Turbine, en español Turbina de eje Horizontal.

**IDE:** Integrated Development Environment, en español Entorno de Desarrollo Integrado.

**Matlab:** abreviatura de MATrix LABoratory, "laboratorio de matrices" es un software matemático que posee un entorno de desarrollo integrado, con un lenguaje de programación propio.

**PU:** Sistemas en Por Unidad.

**PWM:** Pulse Width Modulation en español, Modulación por Ancho de pulsos.

**RTDX:** Real Time Data eXchange,

**RTW:** Real Time Workshop es una herramienta que permite la creación de código en lenguaje C a partir de un modelo en Simulink.

**SI:** Sistema Internacional.

**Simscape:** Librería especializada para la simulación de sistemas eléctricos, mecánicos de fluidos entre otros.

**Simulink:** Entorno que permite el modelado de sistema por medio de bloques, la simulación, generación automática de código y pruebas continuas de sistemas embebidos.

**Stall Controlled:** Control por Pérdida Aerodinámica Pasiva.

**TI:** Texas Instrument, es una de las principales compañías dedicadas al diseño de tarjetas procesadoras de Señales (DSP) y creadora del Code Composer Studio (CCS).

**VAWT:** Vertical Axis Wind Turbine, en español Turbina de eje Vertical.

**VSC:** Voltage Source Converters en español Conversores de fuente de voltaje.

## Referencias Bibliográficas

- [1] E. A. Sierra, "Análisis del diseño y control de un generador trifásico doblemente alimentado", Universidad de Chile, Santiago, Chile, Mayo, 2012.
- [2] J. A. Álvarez, "Estudio de Diferentes Topologías de Generación Eólica", Universidad Pública de Navarra, Pamplona, España, Junio, 2011.
- [3] J. P. Cárdenas, "Simulación y control de una turbina eólica para generación de energía eléctrica", Universidad Católica del Perú, Lima, Perú, 2015.
- [4] MathWorks, Specialized Technology, Disponible:  
<https://la.mathworks.com/help/physmod/sps/specialized-technology.html>
- [5] Haitham Abu-Rub; Mariusz Malinowski; Kamal Al-Haddad, *Power Electronics for Renewable Energy Systems, Transportation and Industrial Applications*, Wiley-IEEE Press, 2014.
- [6] Gonzalo Abad, Jesús López, Miguel Rodríguez, Luis Marroyo, and Grzegorz Iwanski, *Doubly Fed Induction Machine: Modeling and Control for Wind Energy Generation*, Wiley, 2011.
- [7] Lesyani T. León Viltre, Francisco de Assis Dos Santos Neves, Miriam Vilaragut Llanes, *Análisis y modelado del generador de inducción doblemente alimentado*, 2013.
- [8] R. O. Domínguez, "Emulador de Turbina Eólica para el Banco de Pruebas de Generación Eoloeléctrica", Centro Nacional de Investigación y Desarrollo Tecnológico, Cuernavaca, Morelos, México, Julio, 2007.
- [9] S. E. Argueta, C. A. Palacios, M. A. Vásquez, "Diseño de una estación de laboratorio de automatización y control automático para la Escuela de Ingeniería Eléctrica de la UES", Universidad de El Salvador, San Salvador, El Salvador, Noviembre, 2010.
- [10] *eZdspF2812 Technical Reference*, Digital Spectrum, 2003

# ANEXOS

## 1- Código de Variables utilizadas.

```
1. % UNIVERSIDAD DE EL SALVADOR
2. % FACULTAD DE INGENIERIA Y ARQUITECTURA
3. % ESCUELA DE INGENIERIA ELECTRICA
4.
5. % TEMA: DISEÑO DE UN CONTROLADOR DIGITAL, PARA LA VELOCIDAD DE UNA
   TURBINA EÓLICA
6.
7. % CREADO POR: GONZALO ABAD, MONDRAGON UNIBERSITATEA
8. % ADAPTADO POR: ERNESTO ALONSO CERON RODRIGUEZ
9.
10.  clear all
11.  clc
12.
13.  %% DFIG parameters - Rotor parameters referred to stator side
14.
15.  f=60;           %Frecuencia del Estator (Hz)
16.  Ps=2e6;         %Potencia Nominal del Estator (W)
17.  n=1800;         %Velocidad Rotacional Nominal (rev/min)
18.  Vs=480;         %Voltaje Nominal del Estator (V)
19.  Is=1760;        %Corriente Nominal del Estator (A)
20.  Tem=12732;     %Toque Nominal (N.m)
21.
22.  p=2;           %Par de polos
23.  u = 1/3;       %Relacion de vueltas entre el Estator y el Rotor
24.  Vr = 2070;     %Voltaje Nominal del Rotor (V)
25.  smax = 1/3;   %Delizamiento Maximo
26.  Vr_stator = (Vr*smax)*u; %Voltaje Nominal del Rotor referido
   al Estator
27.  Rs = 2.6e-3;  %Resistencia del Estator (ohm)
28.  Lsi = 0.087e-3; %Inductancia de Fuga entre Rotor y
   Estator (H)
29.  Lm = 2.5e-3; %Inductancia de Magnetizacion
30.  Rr = 2.9e-3; %Resistencia del Rotor referido al Estator (ohm)
31.  LS = Lm+Lsi; %Inductancia del Estator (H)
32.  Lr = Lm+Lsi; %Inductancia del Rotor (H)
33.  Vbus = Vr_stator*sqrt(2); %Voltaje del bus de DC
   referidoal Estator (V)
34.  sigma = 1-Lm^2/(LS*Lr);
35.  Fs = Vs*sqrt(2/3)/(2*pi*f); %Flujo del Estator (Wb)
36.
37.  J=127;           %Inercia
38.  D=1e-3;         %Amortiguamiento
39.  fsw=4e3;        %Frecuencia de Muestreo (Hz)
40.  Ts=1/fsw/60;   %Tiempo de Muestreo (sec)
41.
42.  %Reguladores PI
43.  tau_i=(sigma*Lr)/Rr;
44.  tau_n=0.05;
45.  wni=100*(1/tau_i);
46.  wnn=1/tau_n;
47.
48.  kp_id=(2*wni*sigma*Lr)-Rr;
49.  kp_iq=kp_id;
50.  ki_id=(wni^2)*Lr*sigma;
```

```

51.     ki_iq=ki_id;
52.     kp_n=(2*wnn*J)/p;
53.     ki_n=((wnn^2)*J)/p;
54.     %Modelo de la turbina de 3 palas
55.
56.     N=100; %Relacion de la caja de cambios
57.     Radio=42; %Radio o longitud de la pala.
58.     ro=1.225; %Densidad del aire
59.
60.     %Curvas de Cp y Ct
61.     beta=0; %Angulo de la pala
62.     ind2=1;
63.
64.     for lambda=0.1:0.01:11.8
65.         lambdai(ind2)=(1./((1./(lambda-
0.02.*beta)+(0.003./(beta^3+1)))));
66.         Cp(ind2)=0.73*(151./lambdai(ind2)-0.58.*beta-
0.002.*beta^2.14-13.2).* (exp(-18.4./lambdai(ind2)));
67.         Ct(ind2)=Cp(ind2)/lambda;
68.         ind2=ind2+1;
69.     end
70.     tab_lambda=(0.1:0.01:11.8);
71.
72.     %Curva de potencia en funcion de la velocidad del viento
73.
74.     P =
1.0e+06*[0,0,0,0,0,0,0,0.0472,0.1097,0.1815,0.2568,0.3418,...
75.         0.4437,0.5642,0.7046,0.8667,1.0518,1.2616,1.4976,1.7613,2
.0534,...
76.         2.3513,2.4024,2.4024,2.4024,2.4024,2.4024,2.4024];
77.
78.     V
= [0.0000,0.5556,1.1111,1.6667,2.2222,2.7778,3.3333,3.8889,4.4444,..
79.     .4444,...
80.     10.0000,10.5556,11.1111,11.6667,12.2222,12.7778,13.3333,1
3.8889,...
81.     14.4444,15.0000];
82.
83.     figure
84.     subplot(1,2,1)
85.     plot(tab_lambda,Ct,'linewidth',1.5)
86.     grid
87.     xlabel('lambda','fontsize',14)
88.     ylabel('Ct','fontsize',14)
89.
90.     subplot(1,2,2)
91.     plot(V,P,'linewidth',1.5)
92.     grid
93.     xlabel('Wind speed (m/s)','fontsize',14)
94.     ylabel('Power (W)','fontsize',14)

```

## 2- Guía de inicio rápido para la tarjeta eZdsp F2812

A continuación se presentan los requerimientos de hardware y software necesarios, así como la configuración, para la puesta en operación de la tarjeta eZdsp F2812 por medio de Matlab/Simulink y el uso del Code Composer Studio.

### Requerimientos de hardware.

El computador con el que se trabajará deberá cumplir con los siguientes requerimientos mínimos necesarios para poder arrancar la tarjeta eZdsp F2812 mediante Matlab/Simulink.

- Procesador Intel Pentium o superior compatible.
- 64 MB de memoria RAM (recomendado 128 MB).
- 3.1 GB libres de disco duro para instalar Matlab/Simulink y CCS.
- Monitor a color
- Un puerto paralelo para conectar la tarjeta eZdsp F2812 al computador.
- Unidad lectora de CD o DVD.

Las tarjetas soportadas por la herramienta de Simulink Embedded Target for TI C200 se enlistan en la tabla A2.1. El contenido del kit eZdsp F2812 se presenta en la figura A2.1.

Tarjetas de inicio (DSK) de la compañía Spectrum Digital	Tarjetas basadas en cualquiera de los siguientes DSC's de TI	
DSK eZdsp320F2812	TMS320F2801	TMS320F2810
DSK eZdsp320F2808	TMS320F2802	TMS320C2811
DSK eZdsp320LF2407	TMS320F2806	TMS320F2811
	TMS320F2808	TMS320R2811
	TMS320F2809	TMS320C2812
	TMS320C2810	TMS320F2812

Tabla A2. 1 Tarjetas soportadas por Simulink.



Figura A2. 1 Contenido del kit de eZdsp F2812.

### Requerimientos de software

En esta sección se presentan los puntos clave necesarios para la instalación del software Matlab/Simulink y del CCS. El espacio requerido en disco duro para la instalación del Matlab/Simulink es de 2.65 GB (aumenta en versiones más recientes) y para la instalación del CCS es de 450 MB.

Antes de comenzar con la instalación es necesario considerar la compatibilidad entre ambos paquetes, ya que es necesario instalar la versión adecuada de cada uno de ellos; si no se instalan las versiones adecuadas se generarán errores al momento de compilar algún archivo.

Una última consideración que no debemos pasar por alto es la ruta de instalación de ambos paquetes, ya que no debe tener espacios en blanco; si se dejan espacios en blanco en la ruta de instalación se generaran errores al momento de compilar. En la tabla A2.2 se muestran las rutas de instalación típicas o que por defecto utiliza cada software.

Software	Versión	Carpeta de instalación
MATLAB	R2006A	C:\R2006a
CCS	3.1	C:\CCStudio_v3.1

Tabla A2. 2 Versión del software utilizado.

Junto con lo anterior, se debe verificar que se incluyen los complementos adicionales de Matlab, ya que estos son necesarios para que el funcionamiento del sistema sea correcto. Los complementos deben incluirse al inicio de la instalación, sin embargo si ya se cuenta con una instalación únicamente se debe instalar el complemento faltante. Los complementos necesarios se listan en la tabla A2.3.

Por otro lado, con la instalación del software CCS también se instalan complementos adicionales y se debe verificar que estos se hayan instalado. En la tabla A2.3. Se presentan los complementos que deben instalarse junto con el CCS. Si se instalan las versiones mostradas de la tabla A2.2, todo el software anteriormente descrito y presentado en la tabla A2.3 será compatible entre sí.

Complementos de MATLAB	Complementos de CCS
SIMULINK Fixed Point	Assembler
Link for Code Composer Studio	Compiler
Real-Time Workshop	Linker
Fixed-Point Toolbox	TI C2000 miscellaneous utilities
Signal Processing Toolbox	Code Composer Setup Utility
Real-Time Workshop Embedded Coder	
Signal Processing Blockset	
SIMULINK Accelerator	

Tabla A2. 3 Complementos de Matlab y CCS

Para determinar si la Embedded Target for TI C2000 DSP está instalada en el sistema, se debe teclear en el espacio de trabajo o línea de comando de Matlab el siguiente comando:

*C2000lib*

Matlab despliega el bloque Embedded Target for TI C2000 DSP, figura A2.2. Conteniendo las librerías que lo conforman.

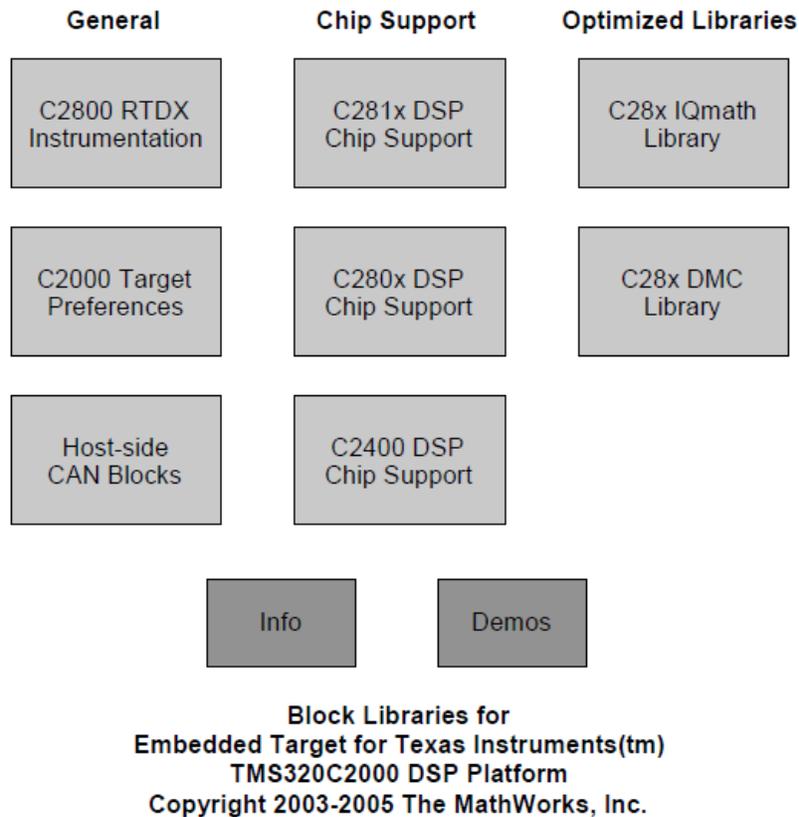


Figura A2. 2 Bloques que conforman la librería Embedded Target for TI C2000 DSP.

Si no se despliega el bloque que se presenta en la figura A2.2 o Matlab no reconoce el comando, entonces será necesario instalar Embedded Target for TI C2000 DSP. Sin este software no será posible utilizar Simulink y Real Time Workshop para desarrollar aplicaciones con la tarjeta eZdsp F2812.

Para verificar si el software CCS se encuentra instalado en el sistema, se debe poner en la línea de comandos de Matlab:

*Ccsboardinfo*

Con el CCS instalado y configurado, Matlab regresa información acerca de las tarjetas que son reconocidas en la PC. En la figura A2.3. Se muestra la manera en que Matlab despliega la información en pantalla.

Board Num	Board Name	Proc Num	Processor Name	Processor Type
1	F2812 Simulator	0	CPU	TMS320C28xx
0	F2812 PP Emulator	0	CPU_1	TMS320C28xx

Figura A2. 3 Lista de tarjetas instaladas en el PC.

Cumpliendo con lo anteriormente descrito se tendrá un sistema listo para operar, restando configurar algunas opciones de Simulink que son presentadas en la siguiente y última sección.

### Configuración básica de Simulink

Antes de generar un modelo en Simulink se deben configurar ciertos parámetros básicos, pero debe dejarse claro que esta configuración podría variar según la aplicación que desee generar.

Cuando se arrastra el bloque *Target Preferences* en el diagrama de Simulink, automáticamente se configuran los parámetros básicos de la simulación. Sin embargo, también se pueden modificar esos valores de modo manual, accediendo mediante: **Simulation Configuration Parameters**.

En modo manual se configura lo siguiente:

- Siguiendo la ruta **Simulation -> Configuration Parameters -> Solver -> Solver Options** se debe especificar el método de integración.
- Ahora bajo la ruta **Simulation -> Configuration Parameters -> Real Time Workshop -> System target file** se especifica el tipo de Sistema que se requiere trabajar y para la eZdsp F2812 se selecciona el archivo `ti_c2000_ert.tlc`.

Un sistema genérico en tiempo real (GRT por sus siglas en inglés), es la configuración destinada a generar código para un sistema en tiempo real si el código resultante será ejecutado en una estación de trabajo o PC, si se requiere un sistema con estas características se debe elegir el archivo `ti_c2000_grt.tlc` bajo la ruta **Simulation -> Configuration Parameters -> Real Time Workshop -> System target file**.

Un sistema embebido en tiempo real (ERT por sus siglas en inglés), es la configuración destinada para generar código para un sistema en tiempo real si el código resultante será ejecutado de modo independiente en una tarjeta de desarrollo, si se requiere un sistema con estas características se debe elegir el archivo `ti_c2000_ert.tlc` bajo la ruta **Simulation -> Configuration Parameters -> Real Time Workshop -> System target file**. Esta opción requiere del complemento Real Time Workshop Embedded Coder.

Con la configuración anterior se genera código en tiempo real y se descarga en la tarjeta eZdsp F2812 dando clic en **Build** bajo la ruta **Simulation -> Configuration Parameters -> Real Time Workshop**.

### 3- Código C generado de RTW.

```
/*
 * File: tescodigo.c
 *
 * Real-Time Workshop code generated for Simulink model tescodigo.
 *
 * Model version                : 1.18
 * Real-Time Workshop file version : 7.1 (R2008a) 23-Jan-2008
 * Real-Time Workshop file generated on : Thu Feb 05 15:28:29 2015
 * TLC version                  : 7.1 (Jan 18 2008)
 * C/C++ source code generated on   : Thu Feb 05 15:28:29 2015
 */

#include "tescodigo.h"
#include "tescodigo_private.h"

/* Block signals (auto storage) */
BlockIO_tescodigo tescodigo_B;
```

```

/* Block states (auto storage) */
D_Work_tesiscodigo tesiscodigo_DWork;

/* Real-time model */
RT_MODEL_tesiscodigo tesiscodigo_M_;
RT_MODEL_tesiscodigo *tesiscodigo_M = &tesiscodigo_M_;

/* Model step function */
void tesiscodigo_step(void)
{
    /* local block i/o variables */
    real_T rtb_DataTypeConversion1;
    real_T rtb_Sum_1;
    real_T rtb_Fcn1_c;
    real_T rtb_Sum1_m;
    real_T rtb_Sum_g;

    {
        real_T rtb_DataTypeConversion_idx;
        real_T rtb_DataTypeConversion_idx_0;
        real_T rtb_DataTypeConversion_idx_1;
        real_T rtb_DataTypeConversion_idx_2;

        /* S-Function Block: <Root>/ADC (c28xadc) */
        {
            AdcRegs.ADCTRL2.bit.RST_SEQ1 = 0x1; // Sequencer reset
            AdcRegs.ADCTRL2.bit.SOC_SEQ1 = 0x1; // Software start of conversion
            while (AdcRegs.ADCST.bit.INT_SEQ1 == 0) {
                //Wait for Sequencer Busy bit to
clear
            }

            asm(" RPT #11 || NOP");
            tesiscodigo_B.ADC_o1 = (AdcRegs.ADCRESULT0) >> 4;
            tesiscodigo_B.ADC_o2 = (AdcRegs.ADCRESULT1) >> 4;
            tesiscodigo_B.ADC_o3 = (AdcRegs.ADCRESULT2) >> 4;
            tesiscodigo_B.ADC_o4 = (AdcRegs.ADCRESULT3) >> 4;
            tesiscodigo_B.ADC_o5 = (AdcRegs.ADCRESULT4) >> 4;
            tesiscodigo_B.ADC_o6 = (AdcRegs.ADCRESULT5) >> 4;
            AdcRegs.ADCTRL2.bit.RST_SEQ1 = 0x1; // Sequencer reset
            AdcRegs.ADCST.bit.INT_SEQ1_CLR = 1; // Clear INT_SEQ1 bit
        }

        /* DataTypeConversion: '<Root>/Data Type Conversion' */
        rtb_DataTypeConversion_idx = (real_T)tesiscodigo_B.ADC_o2;
        rtb_DataTypeConversion_idx_0 = (real_T)tesiscodigo_B.ADC_o3;
        rtb_DataTypeConversion_idx_1 = (real_T)tesiscodigo_B.ADC_o5;
        rtb_DataTypeConversion_idx_2 = (real_T)tesiscodigo_B.ADC_o6;

        /* Fcn: '<S7>/Fcn' incorporates:
        *   DataTypeConversion: '<Root>/Data Type Conversion'
        */
        rtb_Fcn1_c = (((real_T)tesiscodigo_B.ADC_o1 -
rtb_DataTypeConversion_idx /

```



```

/* DiscreteIntegrator: '<S3>/Discrete-Time Integrator' */
rtb_DataTypeConversion_idx_2 =
  tesiscodigo_DWork.DiscreteTimeIntegrator_DSTATE;
if (tesiscodigo_DWork.DiscreteTimeIntegrator_DSTATE >=
  tesiscodigo_P.DiscreteTimeIntegrator_UpperSat) {
  rtb_DataTypeConversion_idx_2 =
    tesiscodigo_P.DiscreteTimeIntegrator_UpperSat;
} else {
  if (tesiscodigo_DWork.DiscreteTimeIntegrator_DSTATE <=
    tesiscodigo_P.DiscreteTimeIntegrator_LowerSat) {
    rtb_DataTypeConversion_idx_2 =
      tesiscodigo_P.DiscreteTimeIntegrator_LowerSat;
  }
}

/* Sum: '<S3>/Sum1' incorporates:
 * Gain: '<S3>/Gain'
 */
rtb_DataTypeConversion_idx_2 += tesiscodigo_P.Gain_Gain *
  rtb_DataTypeConversion_idx_1;

/* Saturate: '<S3>/Saturation' */
rtb_DataTypeConversion_idx_2 =
rt_SATURATE(rtb_DataTypeConversion_idx_2,
  tesiscodigo_P.Saturation_LowerSat,
  tesiscodigo_P.Saturation_UpperSat);

/* DataTypeConversion: '<Root>/Data Type Conversion1' */
rtb_DataTypeConversion1 = (real_T)tesiscodigo_B.ADC1_ol;

/* Sum: '<S5>/Sum' incorporates:
 * Constant: '<Root>/id_referencel'
 */
rtb_Sum_1 = tesiscodigo_P.id_referencel_Value -
rtb_DataTypeConversion1;

/* DiscreteIntegrator: '<S5>/Discrete-Time Integrator' */
rtb_Sum_g = tesiscodigo_DWork.DiscreteTimeIntegrator_DSTATE_1;
if (tesiscodigo_DWork.DiscreteTimeIntegrator_DSTATE_1 >=
  tesiscodigo_P.DiscreteTimeIntegrator_UpperS_j) {
  rtb_Sum_g = tesiscodigo_P.DiscreteTimeIntegrator_UpperS_j;
} else {
  if (tesiscodigo_DWork.DiscreteTimeIntegrator_DSTATE_1 <=
    tesiscodigo_P.DiscreteTimeIntegrator_LowerS_c) {
    rtb_Sum_g = tesiscodigo_P.DiscreteTimeIntegrator_LowerS_c;
  }
}

/* Sum: '<S5>/Sum1' incorporates:
 * Gain: '<S5>/Gain'
 */
rtb_Sum_g += tesiscodigo_P.Gain_Gain_b * rtb_Sum_1;

```

```

    /* Fcn: '<S8>/Fcn1' */
    rtb_Fcn1_c = (-rtb_Fcn1_c) * sin(rtb_DataTypeConversion_idx) +
rtb_Sum1_m *
    cos(rtb_DataTypeConversion_idx);

    /* Sum: '<S4>/Sum' incorporates:
    * Gain: '<Root>/Gain'
    * Saturate: '<S5>/Saturation'
    */
    rtb_Sum_g = tesiscodigo_P.Gain_Gain_o * rt_SATURATE(rtb_Sum_g,
    tesiscodigo_P.Saturation_LowerSat_c,
tesiscodigo_P.Saturation_UpperSat_o)
    - rtb_Fcn1_c;

    /* DiscreteIntegrator: '<S4>/Discrete-Time Integrator' */
    rtb_Sum1_m = tesiscodigo_DWork.DiscreteTimeIntegrator_DSTATE_g;
if (tesiscodigo_DWork.DiscreteTimeIntegrator_DSTATE_g >=
    tesiscodigo_P.DiscreteTimeIntegrator_UpperS_p) {
    rtb_Sum1_m = tesiscodigo_P.DiscreteTimeIntegrator_UpperS_p;
} else {
    if (tesiscodigo_DWork.DiscreteTimeIntegrator_DSTATE_g <=
        tesiscodigo_P.DiscreteTimeIntegrator_LowerS_o) {
        rtb_Sum1_m = tesiscodigo_P.DiscreteTimeIntegrator_LowerS_o;
    }
}

    /* Sum: '<S4>/Sum1' incorporates:
    * Gain: '<S4>/Gain'
    */
    rtb_Sum1_m += tesiscodigo_P.Gain_Gain_of * rtb_Sum_g;

    /* Saturate: '<S4>/Saturation' */
    tesiscodigo_B.Gain1_p = rt_SATURATE(rtb_Sum1_m,
    tesiscodigo_P.Saturation_LowerSat_l,
tesiscodigo_P.Saturation_UpperSat_e);

    /* Gain: '<S9>/Gain' incorporates:
    * Constant: '<S9>/Constant'
    */
    tesiscodigo_B.Gain2 = tesiscodigo_P.Gain_Gain_l *
    tesiscodigo_P.Constant_Value_f;

    /* Gain: '<S9>/Gain1' */
    tesiscodigo_B.Fcn1 = tesiscodigo_P.Gain1_Gain *
rtb_DataTypeConversion1;

    /* Sum: '<S9>/Sum' */
    tesiscodigo_B.Gain2 = tesiscodigo_B.Gain2 - tesiscodigo_B.Fcn1;

    /* Fcn: '<S9>/Fcn' incorporates:
    * Constant: '<S9>/Constant1'
    */
    tesiscodigo_B.Fcn1 = rtb_DataTypeConversion_idx_2 -
tesiscodigo_B.Gain2 *

```

```

0.066128417951312746 * 0.0025869999999999999 * rtb_Fcn1_c;

/* Fcn: '<S9>/Fcn1' incorporates:
 * Constant: '<S9>/Constant1'
 */
tesiscodigo_B.Gain = (tesiscodigo_B.Gain2 * 0.066128417951312746 *
0.0025869999999999999 *
rtb_DataTypeConversion_idx_0 +
    tesiscodigo_B.Gain1_p) + tesiscodigo_B.Gain2 *
9.6637031310398147E-001 * tesiscodigo_P.Constant1_Value;

/* Fcn: '<S10>/Fcn' */
tesiscodigo_B.Gain2 = tesiscodigo_B.Fcn1 *
cos(rtb_DataTypeConversion_idx) -
    tesiscodigo_B.Gain * sin(rtb_DataTypeConversion_idx);

/* Fcn: '<S10>/Fcn1' */
tesiscodigo_B.Fcn1 = tesiscodigo_B.Fcn1 *
sin(rtb_DataTypeConversion_idx) +
    tesiscodigo_B.Gain * cos(rtb_DataTypeConversion_idx);

/* Fcn: '<S11>/Fcn' */
tesiscodigo_B.Gain = tesiscodigo_B.Gain2;

/* Gain: '<S6>/Gain' */
tesiscodigo_B.Gain = tesiscodigo_P.Gain_Gain_i * tesiscodigo_B.Gain;

/* Fcn: '<S11>/Fcn1' */
tesiscodigo_B.Gain1_p = -(sqrt(3.0) / 2.0 * tesiscodigo_B.Fcn1 +
    tesiscodigo_B.Gain2 / 2.0);

/* Gain: '<S6>/Gain1' */
tesiscodigo_B.Gain1_p = tesiscodigo_P.Gain1_Gain_l *
tesiscodigo_B.Gain1_p;

/* Fcn: '<S11>/Fcn2' */
tesiscodigo_B.Gain2 = -(tesiscodigo_B.Gain2 / 2.0 - sqrt(3.0) / 2.0 *
    tesiscodigo_B.Fcn1);

/* Gain: '<S6>/Gain2' */
tesiscodigo_B.Gain2 = tesiscodigo_P.Gain2_Gain * tesiscodigo_B.Gain2;

/* S-Function Block: <Root>/PWM (c28xpwm) */
{
    EvaRegs.CMPR1 = (uint16_T) (tesiscodigo_B.Gain / 2);
    EvaRegs.CMPR2 = (uint16_T) (tesiscodigo_B.Gain1_p / 2);
    EvaRegs.CMPR3 = (uint16_T) (tesiscodigo_B.Gain2 / 2);
}

/* Gain: '<S3>/Gain1' */
tesiscodigo_B.Gain1 = tesiscodigo_P.Gain1_Gain_e *
    rtb_DataTypeConversion_idx_1;

/* Gain: '<S4>/Gain1' */

```

```

tesiscodigo_B.Gain1_b = tesiscodigo_P.Gain1_Gain_e4 * rtb_Sum_g;

/* Gain: '<S5>/Gain1' */
tesiscodigo_B.Gain1_k = tesiscodigo_P.Gain1_Gain_f * rtb_Sum_l;

/* Update for DiscreteIntegrator: '<S3>/Discrete-Time Integrator' */
tesiscodigo_DWork.DiscreteTimeIntegrator_DSTATE =
    tesiscodigo_P.DiscreteTimeIntegrator_gainval * tesiscodigo_B.Gain1
+
    tesiscodigo_DWork.DiscreteTimeIntegrator_DSTATE;
if (tesiscodigo_DWork.DiscreteTimeIntegrator_DSTATE >=
    tesiscodigo_P.DiscreteTimeIntegrator_UpperSat) {
    tesiscodigo_DWork.DiscreteTimeIntegrator_DSTATE =
        tesiscodigo_P.DiscreteTimeIntegrator_UpperSat;
} else {
    if (tesiscodigo_DWork.DiscreteTimeIntegrator_DSTATE <=
        tesiscodigo_P.DiscreteTimeIntegrator_LowerSat) {
        tesiscodigo_DWork.DiscreteTimeIntegrator_DSTATE =
            tesiscodigo_P.DiscreteTimeIntegrator_LowerSat;
    }
}

/* Update for DiscreteIntegrator: '<S5>/Discrete-Time Integrator' */
tesiscodigo_DWork.DiscreteTimeIntegrator_DSTATE_l =
    tesiscodigo_P.DiscreteTimeIntegrator_gainva_n *
tesiscodigo_B.Gain1_k +
    tesiscodigo_DWork.DiscreteTimeIntegrator_DSTATE_l;
if (tesiscodigo_DWork.DiscreteTimeIntegrator_DSTATE_l >=
    tesiscodigo_P.DiscreteTimeIntegrator_UpperS_j) {
    tesiscodigo_DWork.DiscreteTimeIntegrator_DSTATE_l =
        tesiscodigo_P.DiscreteTimeIntegrator_UpperS_j;
} else {
    if (tesiscodigo_DWork.DiscreteTimeIntegrator_DSTATE_l <=
        tesiscodigo_P.DiscreteTimeIntegrator_LowerS_c) {
        tesiscodigo_DWork.DiscreteTimeIntegrator_DSTATE_l =
            tesiscodigo_P.DiscreteTimeIntegrator_LowerS_c;
    }
}

/* Update for DiscreteIntegrator: '<S4>/Discrete-Time Integrator' */
tesiscodigo_DWork.DiscreteTimeIntegrator_DSTATE_g =
    tesiscodigo_P.DiscreteTimeIntegrator_gainva_b *
tesiscodigo_B.Gain1_b +
    tesiscodigo_DWork.DiscreteTimeIntegrator_DSTATE_g;
if (tesiscodigo_DWork.DiscreteTimeIntegrator_DSTATE_g >=
    tesiscodigo_P.DiscreteTimeIntegrator_UpperS_p) {
    tesiscodigo_DWork.DiscreteTimeIntegrator_DSTATE_g =
        tesiscodigo_P.DiscreteTimeIntegrator_UpperS_p;
} else {
    if (tesiscodigo_DWork.DiscreteTimeIntegrator_DSTATE_g <=
        tesiscodigo_P.DiscreteTimeIntegrator_LowerS_o) {
        tesiscodigo_DWork.DiscreteTimeIntegrator_DSTATE_g =
            tesiscodigo_P.DiscreteTimeIntegrator_LowerS_o;
    }
}

```

```

}
}

/* Model initialize function */
void tesiscodigo_initialize(boolean_T firstTime)
{
    (void)firstTime;

    /* Registration code */

    /* initialize error status */
    rtmSetErrorStatus(tesiscodigo_M, (const char_T *)0);

    /* block I/O */
    (void) memset(((void *) &tesiscodigo_B),0,
        sizeof(BlockIO_tesiscodigo));

    {
        int_T i;
        void *pVoidBlockIORegion;
        pVoidBlockIORegion = (void *) (&tesiscodigo_B.Gain1);
        for (i = 0; i < 7; i++) {
            ((real_T*)pVoidBlockIORegion)[i] = 0.0;
        }
    }

    /* states (dwork) */
    {
        real_T *dwork_ptr = (real_T *)
            &tesiscodigo_DWork.DiscreteTimeIntegrator_DSTATE;
        dwork_ptr[0] = 0.0;
        dwork_ptr[1] = 0.0;
        dwork_ptr[2] = 0.0;
    }

    InitAdc();
    config_ADC_A (5U, 12816U, 84U, 0U, 0U);
    config_ADC_B (1U, 152U, 0U, 0U, 0U);
    EALLOW;
    GpioMuxRegs.GPAMUX.all |= 63U;          // EVA PWM 1-6 pins
    config_PWM_A (500.0,1,1,"INPUT_PORT",0.0,
        1,"INPUT_PORT",0.0,1,"INPUT_PORT",0.0,273,
        0, 0, 0, 0, 1, 2);

    EDIS;

    /* InitializeConditions for DiscreteIntegrator: '<S3>/Discrete-Time
Integrator' */
    tesiscodigo_DWork.DiscreteTimeIntegrator_DSTATE =
        tesiscodigo_P.DiscreteTimeIntegrator_IC;

    /* InitializeConditions for DiscreteIntegrator: '<S5>/Discrete-Time
Integrator' */
    tesiscodigo_DWork.DiscreteTimeIntegrator_DSTATE_l1 =
        tesiscodigo_P.DiscreteTimeIntegrator_IC_e;

```

```
/* InitializeConditions for DiscreteIntegrator: '<S4>/Discrete-Time
Integrator' */
    tesiscodigo_DWork.DiscreteTimeIntegrator_DSTATE_g =
        tesiscodigo_P.DiscreteTimeIntegrator_IC_1;
}

/* Model terminate function */
void tesiscodigo_terminate(void)
{
    /* (no terminate code required) */
}

/* File trailer for Real-Time Workshop generated code.
 *
 * [EOF]
 */
```