



**Universidad
Zaragoza**

Trabajo Fin de Grado

Grado en Ingeniería de Tecnologías Industriales

Librería de control de sistemas de potencia para PLC
Montelec MTC-4000

Autor

Francisco Javier Melero Alegría

Director

Jesús Oscar Alegría Arriazu

Ponente

Jesús Acero Acero

Escuela de Ingeniería y Arquitectura
Septiembre 2015



Escuela de
Ingeniería y Arquitectura
Universidad Zaragoza

DECLARACIÓN DE AUTORÍA Y ORIGINALIDAD

(Este documento debe acompañar al Trabajo Fin de Grado (TFG)/Trabajo Fin de Máster (TFM) cuando sea depositado para su evaluación).

D./D^a. Francisco Javier Melero Alegría,

con nº de DNI 78756376-Y en aplicación de lo dispuesto en el art.

14 (Derechos de autor) del Acuerdo de 11 de septiembre de 2014, del Consejo de Gobierno, por el que se aprueba el Reglamento de los TFG y TFM de la Universidad de Zaragoza,

Declaro que el presente Trabajo de Fin de (Grado/Máster)
Grado _____, (Título del Trabajo)

Librería de control de sistemas de potencia para PLC Montelec MTC-4000

es de mi autoría y es original, no habiéndose utilizado fuente sin ser citada debidamente.

Zaragoza, 31 de agosto de 2015

Fdo: Francisco Javier Melero Alegría

AGRADECIMIENTOS

Es suficiente con echar un vistazo rápido a este documento para darse cuenta de que es el resultado de un largo proceso de aprendizaje en el cual se han involucrado una gran cantidad de personas, es por eso que me gustaría mostrar mi agradecimiento a cada una de ellas.

En primer lugar quiero dar las gracias al equipo de las empresas Pasaban y Montelec por darme la oportunidad de llevar a cabo la realización de este proyecto en colaboración con ellos y más concretamente a mi director Jesús Oscar por estar siempre dispuesto a ayudar con las dudas que han surgido durante el trabajo así como a proporcionarme acceso al equipo y documentación que han sido necesarios para la elaboración de este.

Quiero también mostrar mi agradecimiento a Jesús Acero, ponente en la universidad de Zaragoza, por su inestimable ayuda en la redacción y enfoque del proyecto así como por su completa disponibilidad y voluntad de trabajo sin las cuales no habría sido posible la redacción de este documento.

No puedo dejar de mencionar a mis padres, que han hecho un gran esfuerzo para permitirme estudiar la carrera que finaliza con la realización de este trabajo y que siempre han estado ahí para apoyarme en los momentos en los que el estrés y el nerviosismo han sido un lastre en la consecución de mis objetivos.

En último lugar quiero también agradecer a mis amigos y amigas su apoyo y su ayuda a lo largo de toda la carrera, gracias por recordarme que no todo es estudiar y por hacerme ver siempre los problemas desde un punto de vista más despreocupado.

RESUMEN

Librería de control de sistemas de potencia para PLC Montelec MTC-4000

A lo largo del siguiente trabajo se lleva a cabo el desarrollo de una librería de control para sistemas de potencia que operará en el PLC Montelec MTC 4000 con la cual es posible el diseño de distintos tipos de control tanto para motores como para otros sistemas eléctricos. El objetivo principal del trabajo es el de crear una **herramienta** capaz de facilitar el proceso de diseño y desarrollo de los controles que se implementarán en las maquinas fabricadas por la empresa Pasaban.

Para ello, en primer lugar se realizan los correspondientes **modelados y simulaciones** con el software SIMULINK tanto de los sistemas a controlar como de los controles que los gobiernan, obteniendo así la respuesta esperada de los sistemas e información acerca de la manera óptima de realizar el control.

A continuación, basándose en las simulaciones se lleva a cabo el **desarrollo de la librería**, que se compone de diferentes bloques de funciones capaces de realizar acciones determinadas. Estos bloques son suficientes para realizar distintos tipos de control al conectarse entre sí.

Una vez desarrollada la librería se realizan tres **ejemplos de aplicación** de la misma para probar su funcionalidad. El primero es el control tensión frecuencia de un motor asíncrono, el segundo es el control vectorial de este mismo motor asíncrono y el tercero es un SRF-PLL capaz de sincronizarse con la red eléctrica.

Para cada uno de los tres ejemplos de aplicación **se comparan los resultados** de las simulaciones con los resultados experimentales obtenidos en la empresa y se analiza la respuesta de los sistemas estudiando así la funcionalidad de la librería y su utilidad a la hora de desarrollar diferentes tipos de control.

En último lugar se realiza un breve análisis del funcionamiento de los controles y de los resultados obtenidos, además se plantean diferentes proyectos que se pueden llevar a cabo con la librería abriendo así nuevas vías para la continuación del desarrollo de este trabajo.

Índice de contenido

1 INTRODUCCIÓN	1
1.1) Breve introducción al proyecto y la empresa.....	1
1.2) Motivación y objetivos.	1
1.3) Alcance.....	3
1.4) Descripción de contenido.....	3
2 DESCRIPCIÓN Y DESARROLLO DE LA LIBRERÍA	5
2.1) Funcionamiento de la librería.....	5
2.2) Bloques de Control.	6
2.2.1) clarke.....	6
2.2.2) clarkei.....	6
2.2.3) park.....	7
2.2.4) parki.....	7
2.2.5) pid.....	8
2.2.6) pid_antiwindup.....	8
2.2.7) integrador.....	9
2.2.8) integrador_ang.....	9
2.2.9) generador_rampa.....	10
2.2.10) funcion_control_FV.....	10
2.2.11) mod_7s.....	11
2.2.12) MODELO_DE_FLUJO.....	14
2.2.13) par_iq.....	15
2.2.14) radianes_grados.....	15
2.2.15) rad_seg_rpm.....	15
2.2.16) rad_seg_hz.....	15
2.3) Bloques de entradas/salidas.....	16
2.3.1) READ_NOMINAL_VALUES.....	16
2.3.2) READ_BUS_VOLTAGE.....	16
2.3.3) READ_CURRENT.....	16
2.3.4) READ_SPEED.....	17
2.3.5) READ_PHASE_VOLTAGES.....	17
2.3.6) TA_TB_TC_OUT.....	17
2.3.7) VA_VB_OUT.....	17
2.3.8) OUT_TO_OSCILOSCOPE.....	17

3 APLICACIONES Y RESULTADOS	19
3.1) Control F-V	20
3.1.1) Principio de funcionamiento	20
3.1.2) Modelo en Matlab del control F-V	22
3.1.3) Programación del control en CoDeSys	23
3.1.4) Estudio de resultados	27
3.1.4) Conclusiones.	30
3.2) Control Vectorial.....	31
3.2.1) Principio de funcionamiento	31
3.2.2) Modelo en Matlab del control vectorial.	32
3.2.3) Programación del control en CoDeSys	34
3.2.4) Estudio de resultados	38
3.2.5) Conclusiones.	41
3.3) SRF-PLL.....	42
3.3.1) Principio de funcionamiento	42
3.3.2) Programación del SRF-PLL en CoDeSys	43
3.3.3) Estudio de resultados	45
3.3.4) Conclusiones.	45
4 CONCLUSIONES Y PERSPECTIVAS FUTURAS	47
4.1) Análisis de la respuesta del sistema y aplicabilidad del proyecto.....	47
4.2) Perspectivas futuras para el proyecto	48
4.2.1) Control vectorial para maquina Síncrona	48
4.2.2) Control sensorless para maquina asíncrona	48
4.2.3) Inversor para la conexión de placas solares en un sistema trifásico	48
5 BIBLIOGRAFÍA	49
6 ANEXOS	53
6.1) Líneas de código	53
6.1.1) Clarke	53
6.1.2) Clarkei.....	53
6.1.3) Park	53
6.1.4) Parki	53
6.1.5) Pid	53
6.1.6) Pid_antiwindup	53
6.1.7) integrador	54
6.1.8) integrador_ang.....	54

6.1.9) generador_rampa	54
6.1.10) rad_seg_rpm	54
6.1.11) funcion_control_FV.....	54
6.1.12) mod_7s.....	55
6.1.13) modelo_de_flujo.....	56
6.1.14) radianes_grados.....	57
6.1.15) rad_seg_hz	57
6.1.16) par_iq	57
6.1.17) READ_NOMINAL_VALUES	57
6.1.18) READ_BUS_VOLTAGE	57
6.1.19) READ_CURRENT	57
6.1.20) READ_SPEED	58
6.1.21) READ_PHASE_VOLTAGES	58
6.1.22) TA_TB_TC_OUT.....	58
6.1.22) VA_VB_OUT	59
6.1.22) OUT_TO_OSCILSCOPE	59
6.2) Equipo Empleado.....	60
6.2.1) Inversor	60
6.2.2) PLC MTC4000	60
6.2.3) Tarjeta de control de motores y regeneración MTC 4011.....	61
6.2.4) Motor Asíncrono VASCAT MAC QI 132 L.....	61
6.2.5) CoDeSys.....	62
6.2.6) Pasaban Drivers.....	62
6.3) Modelado del inversor.....	63
6.4) Modelado del motor.....	64
6.5) Ficha técnica del motor asíncrono VASCAT MAC QI 132 L.....	66
6.6) Ficha técnica del inversor Montelec NFS 200-20	69
6.7) Ficha técnica del PLC MTC4000	70
6.8) Ficha técnica de la tarjeta de control de motores MTC4011	72

MEMORIA

1 INTRODUCCIÓN

1.1) Breve introducción al proyecto y la empresa.

Pasaban es una empresa dedicada a la **fabricación de máquinas de corte** de todo tipo de papel (incluido papel moneda), **bobinadoras y empaquetadoras** de alta precisión. En la Figura 1.1 se puede observar una cortadora industrial Pasaban de tamaño medio. Capaces de cortar papel con una precisión de décima de milímetro y una velocidad de hasta 400 metros por minuto Pasaban posee 600 máquinas en activo en más de 40 países incluyendo algunos de los principales productores industriales del mundo (Alemania, China, India, Estados Unidos, Japón y España entre otros).

Para realizar este tipo de operaciones los motores de las maquinas Pasaban necesitan ser controlados y coordinados con una gran precisión a través de la electrónica digital y de potencia. Es por eso que Pasaban cuenta con una empresa filial llamada MONTELEC (Montajes Electrónicos SL.) propiedad del grupo Pasaban dedicada al diseño y la fabricación de equipos de potencia, tarjetas de comunicación, PLCs e inversores (entre otros dispositivos) para sus máquinas.



Figura 1.1: Cortadora Pasaban.

1.2) Motivación y objetivos.

Actualmente el control de los motores de las cortadoras se realiza mediante el PLC MTC4000 que envía las referencias de par y velocidad a las tarjetas de control de potencia MTC4011 a través de un bus GTLP de alta velocidad. Las tarjetas de control se conectan a la potencia mediante un cable plano y al encoder del motor que puede ser incremental o absoluto SSI.

Este sistema de control garantiza una gran **fiabilidad** durante el proceso de producción pero carece de **flexibilidad** a la hora del desarrollo de controles o la investigación de nuevas aplicaciones para las tarjetas MONTELEC, que puedan conseguir clientes en

diferentes ámbitos del control de la electrónica de potencia (placas solares, aerogeneradores...).

De forma más detallada, el control del inversor que alimenta los motores síncronos y asíncronos de las maquinas Pasaban lo realiza el DSP de Texas Instruments de 32 bits en coma flotante TMS320F335 programado en lenguaje C que se encuentra dentro de las tarjetas de control de potencia MTC4011 y se comunica con el PLC MTC4000 mediante un bus GTLP.

Actualmente Montelec está estudiando nuevos mercados en energías renovables y almacenamiento de energía y por ello se busca un sistema para agilizar los nuevos diseños. El **ciclo de diseño actual** es:

- 1- Simulación en Matlab
- 2- Programación de los algoritmos en el DSP
- 3- Test en bancada.

La realización del test es complejo y por tanto se requiere la utilización de emuladores. No se pueden aplicar puntos de ruptura del programa mientras los equipos de potencia están en marcha, cada cambio implica parar todo el sistema y volver a compilar el programa dentro del DSP. Por todo ello se pretende **modificar el ciclo de diseño**:

- 1- Simulación en Matlab.
- 2- Los bloques de Matlab tendrán su equivalente en CoDeSys. Se probará el control en el PLC. Si se detecta algún error se pueden realizar las modificaciones *online*.
- 4- Probar los algoritmos en el DSP cuando el control está ya desarrollado.
- 5- Test en bancada.

El **objetivo** de este proyecto, por tanto, es el **desarrollo y prueba** de una librería de control de potencia para el PLC MONTELEC MTC4000, que permita realizar este tipo de controles de una manera sencilla e intuitiva gracias a un entorno de programación basado en bloques de control. La ventaja de realizar el control de la electrónica de potencia con un PLC es que este es capaz de realizar cambios en tiempo real, acortándose el tiempo de diseño de nuevos controles o equipos. **La tarjeta MTC4011 pierde el control del motor** y se comporta únicamente como un interface entre la potencia y el PLC, dando información de la corriente y velocidad del motor.

1.3) Alcance.

Este proyecto contempla todas las fases de la programación y prueba de la librería de control descrita en el apartado 1.2) que incluyen:

- 1) Simulación de los controles a programar con el software SIMULINK.
- 2) Programación con el software CoDeSys de los bloques necesarios para ejecutar dichos controles en el PLC MTC4000. Estos bloques componen la librería.
- 3) Realización y estudio de tres ejemplos de aplicación empleando la librería: Control tensión frecuencia, control vectorial y SRF-PLL. Estos estudios incluyen una comparación con las simulaciones previas.
- 4) Conclusiones y perspectivas futuras para el proyecto.

1.4) Descripción de contenido.

Esta memoria se ha estructurado en seis capítulos cuyo título y contenidos son:

1 Introducción: En este capítulo se presenta una visión general del contexto de realización del proyecto así como la necesidad, los objetivos y el alcance del mismo.

2 Descripción y desarrollo de la librería: En este capítulo se explica el funcionamiento de cada uno de los bloques tanto de control como de entradas y salidas que componen la librería, incluyendo una breve explicación de su papel en las aplicaciones de control en caso de ser necesaria.

3 Aplicaciones y resultados: En este capítulo se presentan para cada una de las aplicaciones de la librería (control F-V, control vectorial y SRF-PLL) los principios de funcionamiento, el modelo en SIMULINK, el programa en CoDeSys y la comparación de los resultados experimentales con las simulaciones.

4 Conclusiones y perspectivas futuras: En este capítulo se hace un breve repaso de los principales resultados y conclusiones del proyecto, incluyendo la viabilidad del empleo del PLC MTC4000 para controlar sistemas. Además se plantean varias aplicaciones adicionales de la librería que no han sido llevadas a cabo en este proyecto y que podrían seguir con la línea de desarrollo del mismo.

5 Bibliografía: En este capítulo se citan las fuentes de información que se han empleado para el desarrollo de este proyecto.

6 Anexos: En este capítulo se incluye información relevante para el proyecto. Esta información es:

- Líneas de código de la librería
- Información adicional acerca del modelado en SIMULINK del motor asíncrono y el inversor.
- Equipo y software empleado en la realización del proyecto.
- Hojas de características.

2 DESCRIPCIÓN Y DESARROLLO DE LA LIBRERÍA

2.1) Funcionamiento de la librería.

En este apartado se explican las entradas y salidas de cada uno de los bloques que componen la librería así como el funcionamiento interno de los mismos. Las líneas de código de cada bloque pueden ser consultadas en el anexo 6.1).

La librería se compone de dos tipos de bloques:

- 1) **Bloques de control:** Realizan funciones y operaciones determinadas para controlar el sistema.
- 2) **Bloques de entradas/salidas:** Leen y escriben a través de punteros las variables empeladas para controlar el sistema.

Por convenio los nombres de los bloques de control están escritos en minúscula, los nombres de los bloques de entradas y salidas en mayúscula, los nombres de las variables de entrada y salida de los bloques en mayúscula y los nombres de las variables internas de los bloques en minúscula. **Todas las variables son de tipo REAL** dado que el PLC trabaja en coma flotante (exceptuando los RESET que son variables booleanas).

Los bloques que componen la librería están programados en CoDeSys como “**Function Blocks**”, los programas de tipo Function Block operan como funciones convencionales con la excepción de que **pueden almacenar variables internas** facilitando de este modo el empleo de la librería, por ejemplo el Function Block que realiza el control PID tiene una variable interna llamada “acumulado” que representa el resultado de la integral que está realizado.

Todos los controles que se crean a partir de esta librería son programados en una interrupción en el PLC que se ejecuta cada 250 μ s, es decir, **el control del sistema se realiza a una frecuencia de 4000 Hz** por convenio de la empresa. No obstante la frecuencia de control es configurable.

2.2) Bloques de Control.

2.2.1) clarke.

Entradas: A, B, C

Salidas: ALFA, BETA

Funcionamiento: Esta función realiza la transformación de Clarke aplicando las ecuaciones 2.1 y 2.2 a las entradas A, B, C que pueden ser tensiones o corrientes trifásicas.

De esta forma el estado del sistema trifásico en un momento determinado queda representado como un vector con proyecciones ALFA y BETA en los ejes cartesianos en lugar de tres vectores desfasados 120° como se puede observar en la Figura 2.1 (extraída de [5]). Un sistema trifásico de amplitud I y frecuencia F [Hz] quedará representado a lo largo del tiempo como un vector de modulo I que gira con velocidad angular $F \cdot 2 \cdot \pi$ [rad/s].

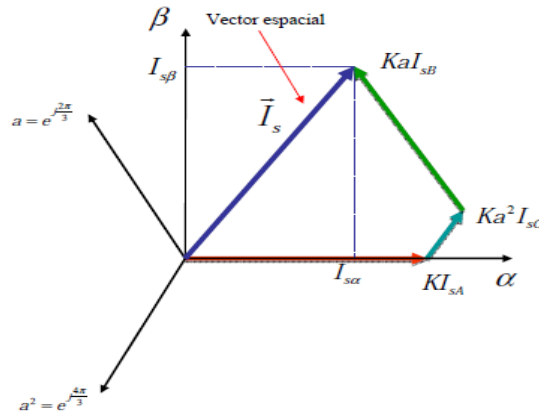


Figura 2.1: Representación gráfica de la transformación de Clarke.

$$I_{s\alpha} = I_{sA} \quad (\text{ec. 2.1})$$

$$I_{s\beta} = \frac{1}{\sqrt{3}} \cdot (I_{sB} - I_{sC}) \quad (\text{ec. 2.2})$$

2.2.2) clarkei.

Entradas: ALFA, BETA

Salidas: A, B, C

Funcionamiento: Esta función realiza la transformación de Clarke inversa aplicando las ecuaciones 2.3, 2.4, 2.5.

$$I_{sA} = I_{s\alpha} \quad (\text{ec. 2.3})$$

$$I_{sB} = -\frac{1}{2} I_{s\alpha} + \frac{\sqrt{3}}{2} I_{s\beta} \quad (\text{ec. 2.4})$$

$$I_{sC} = -\frac{1}{2} I_{s\alpha} - \frac{\sqrt{3}}{2} I_{s\beta} \quad (\text{ec. 2.5})$$

2.2.3) park.

Entradas: ALFA, BETA, SIGMA

Salidas: D, Q

Funcionamiento: Esta función realiza la transformación de Park aplicando las ecuaciones 2.6 y 2.7 a las entradas ALFA y BETA que son producto de la transformación de Clarke de un sistema trifásico, proyectándolas sobre unos ejes cartesianos que forman un ángulo SIGMA* con los ejes cartesianos originales como se puede observar en la Figura 2.2 (extraída de [5]).

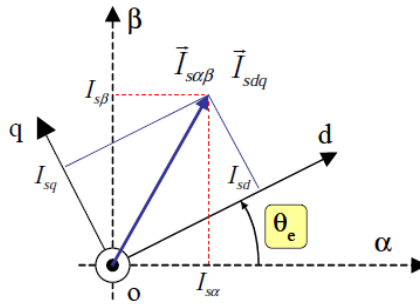


Figura 2.2: Representación gráfica de la transformación de Park.

$$I_{sd} = \cos\theta_e \cdot I_{s\alpha} + \sin\theta_e \cdot I_{s\beta} \quad (\text{ec. 2.6})$$

$$I_{sq} = -\sin\theta_e \cdot I_{s\alpha} + \cos\theta_e \cdot I_{s\beta} \quad (\text{ec. 2.7})$$

2.2.4) parki.

Entradas: D, Q, SIGMA

Salidas: ALFA, BETA

Funcionamiento: Esta función realiza la transformación de Park inversa aplicando las ecuaciones 2.8 y 2.9. Nótese que para realizar esta transformación es necesario un ángulo SIGMA como entrada (al igual que en la transformación de Park).

$$I_{s\alpha} = \cos\theta_e \cdot I_{sd} - \sin\theta_e \cdot I_{sq} \quad (\text{ec. 2.8})$$

$$I_{s\beta} = \sin\theta_e \cdot I_{sd} + \cos\theta_e \cdot I_{sq} \quad (\text{ec. 2.9})$$

*El nombre sigma para el ángulo se ha elegido por convenio de la empresa, es por eso que en las figuras el ángulo está representado con otra letra griega.

2.2.5) pid.

Entradas: ERROR, KP, KI, FREC, RESET

Salidas: ACCION

Variables Internas: acumulado

Funcionamiento: Esta función se trata de un controlador PI sin anti-windup. Calcula la acción para un error de entrada a una frecuencia de control determinada y con unas constantes para el control proporcional e integral, no realiza acción derivativa.

La acción se calcula como la suma del error multiplicado por una ganancia KP (control proporcional) más la integral del error a lo largo del tiempo multiplicada por una ganancia KI (control integral).

Para calcular la integral se emplea la variable interna “acumulado”. En cada llamada a la función que se realiza a una frecuencia FREC (frecuencia de control, normalmente 4000 Hz aunque es configurable) al valor actual de la variable “acumulado” se suma el error dividido de la frecuencia, dicho valor corresponde al incremento del área del acumulado como indica la Figura 2.3. En caso de que RESET=TRUE “acumulado” toma el valor 0.

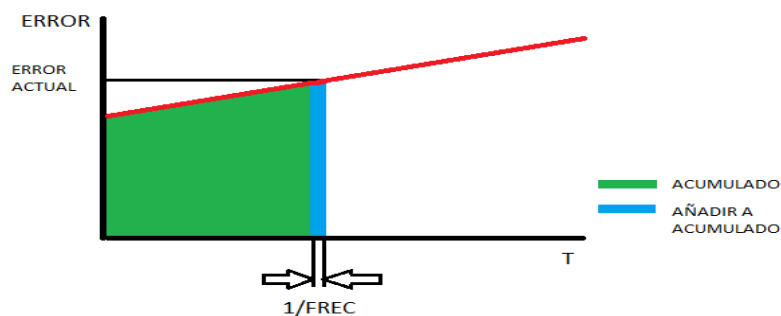


Figura 2.3: Representación gráfica del cálculo de la integral.

2.2.6) pid_antiwindup.

Entradas: ERROR, KP, KI, K_ANT, FREC, ACCION_MAX, ACCION_MIN, RESET

Salidas: ACCION

Variables Internas: acumulado, accion_prop, accion_int, antiwindup, accion_total

Funcionamiento: Esta función se trata de un controlador PI con anti-windup. Calcula la acción para un error de entrada a una frecuencia de control determinada y con unas constantes para el control proporcional e integral, no realiza acción derivativa. En caso de que RESET=TRUE “acumulado” toma el valor 0.

La acción está limitada por los parámetros de entrada “ACCION_MAX” y “ACCION_MIN”. El cálculo de la acción se realiza de forma similar al bloque pid con la excepción de que

a la parte de la acción integral se le resta antes de realizar la integral el término “antiwindup” que depende de la saturación del pid como se puede observar en la Figura 2.4.

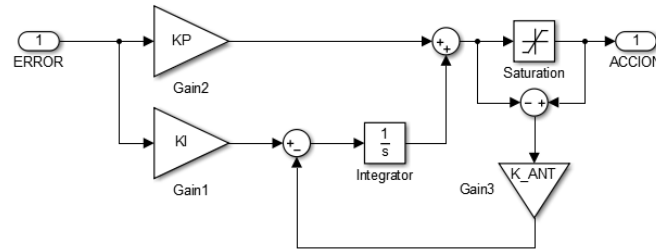


Figura 2.4: Diagrama de bloques en SIMULINK del controlador pid_antiwindup.

2.2.7) integrador.

Entradas: IN, FREC, RESET

Salidas: OUT

Variables Internas: acumulado

Funcionamiento: Esta función calcula la integral en el tiempo de la entrada IN a una frecuencia de control FREC de la misma manera que la realiza la función “pid”. En caso de que RESET=TRUE “acumulado” toma el valor 0.

2.2.8) integrador_ang.

Entradas: IN, FREC, RESET

Salidas: OUT

Variables Internas: acumulado

Funcionamiento: Esta función calcula la integral en el tiempo de la entrada IN a una frecuencia de control FREC de la misma manera que la realiza la función “pid”. OUT varía entre 0 y 2π , de esta forma pueden generarse ángulos OUT que varían con la velocidad angular deseada IN. En caso de que RESET=TRUE “acumulado” toma el valor 0.

2.2.9) generador_rampa.

Entradas: VUELTAS, TIEMPO, FREC, RESET

Salidas: VUELTAS_ACTUALES

Variables Internas: acumulado

Funcionamiento: Esta función genera una rampa en la variable “VUELTAS_ACTUALES” que llega hasta el valor de entrada “VUELTAS” en el tiempo “TIEMPO” medido en segundos. Para ello realiza la integral y en cada llamada a la función le suma al acumulado (valor actual de la integral) el número al que va a llegar (“VUELTAS”) dividido entre el producto de la frecuencia de control (“FREC”) y el tiempo que le va a costar alcanzar dicho valor (“TIEMPO”). Es decir, en cada llamada a la función añade al acumulado el número que hace que este alcance el valor “VUELTAS” en el tiempo “TIEMPO” sabiendo la frecuencia con la que se realiza esta operación. Esta función sirve para generar una referencia en rampa para el motor y observar su respuesta a la misma, de ahí el nombre de las variables. En caso de que RESET=TRUE “acumulado” toma el valor 0.

2.2.10) funcion_control_FV.

Entradas: FRECUENCIA_ONDA

Salidas: V

Funcionamiento: Esta función calcula la tensión de pico en voltios correspondiente a una determinada frecuencia en radianes/segundo para el motor asíncrono VASCAT MAC QI. De esta forma se puede realizar un control tensión-frecuencia del mismo.

Para calcular la tensión correspondiente a cada frecuencia se emplea la ecuación de la recta que pasa por 0 V, 0 rad/s y por el punto de tensión nominal (de pico) y frecuencia nominal. Si la tensión correspondiente a una frecuencia es menor que un determinado valor (en el caso del motor VASCAT 10 V) el valor de la tensión se mantiene constante en ese valor para dicha frecuencia y todas las menores hasta llegar a 0 rad/s como se observa en la figura 2.5, de lo contrario la acción sobre el motor sería insuficiente. El empleo de esta función se explica con más detalle en el apartado 3.1) Control F-V

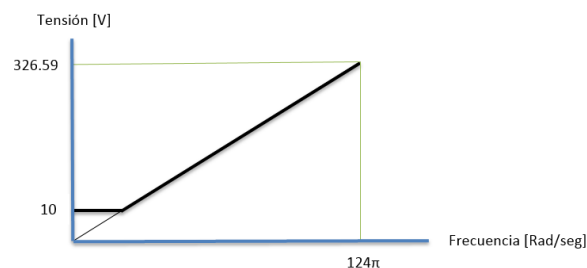


Figura 2.5: Representación gráfica de las tensiones y frecuencias para el motor asíncrono VASCAT MAC QI 132 L.

2.2.11) mod_7s.

Entradas: VALFA, VBETA, XMAX, VBUS

Salidas: TAOUT, TBOUT, TCOUT.

Variables Internas: sector, Tm, Tn, pi, gain, alfa, beta, Ta, Tb, Tc

Funcionamiento: Esta es una de las funciones de **mayor complejidad de la librería**, es por ello que se explica con mayor detalle. Esta función recibe como entradas las componentes ALFA y BETA (transformación de Clarke) de las tensiones trifásicas que queremos aplicar al motor y **realiza la modulación de 7 segmentos** de estas, calculando los tiempos de disparo para cada una de las tres ramas de IGBTs con las que cuenta el inversor.

La modulación vectorial de 7 segmentos representa los 7 posibles estados en los que se pueden encontrar las ramas [A, B, C] del inversor como vectores que forman un hexágono, estos estados pueden tener un valor de 0 si la rama se encuentra a un nivel bajo de tensión o de 1 si la rama se encuentra a un nivel alto de tensión (Tensión del bus de continua que alimenta el inversor).

Los estados posibles para las ramas son los estados básicos $[1,0,0]$, $[0,1,0]$, $[0,0,1]$ y los estados mixtos $[1,1,0]$, $[0,1,1]$, $[1,0,1]$, a estos estados hay que añadir los dos estados de inactividad en los cuales las ramas no tienen diferencia de tensión y en consecuencia no circula corriente entre ellas: $[0,0,0]$, $[1,1,1]$. Los estados se representan gráficamente en la figura 2.6.

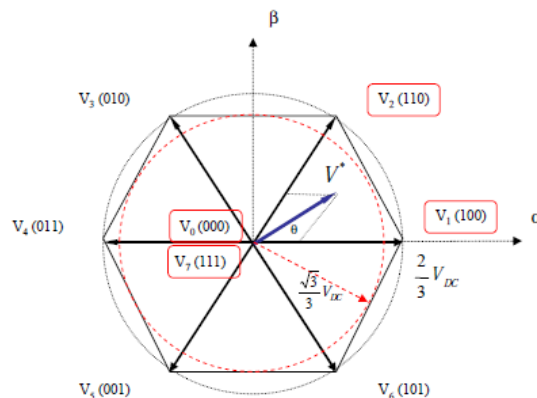


Figura 2.6: Representación gráfica de los estados posibles del inversor en modulación vectorial.

Por tanto, para generar un vector de tensiones representado en los ejes real e imaginario mediante sus proyecciones ALFA y BETA en primer lugar se calcula en que sector se encuentra y seguidamente se le compone mediante los disparos sucesivos (dentro de un tiempo de servicio T) de los dos estados contiguos más los dos estados de inactividad.

Por ejemplo, si se desea generar una tensión cuyo vector se encuentra en el primer sector (numerando desde el eje alfa y en sentido anti horario) se hace mediante los disparos sucesivos de los estados V1 [1,0,0] y V2 [1,1,0]. Para variar el modulo del vector resultante se dispara durante una parte del tiempo de servicio T los estados de inactividad del inversor, a mayor tiempo de estados de inactividad menor será el modulo del vector resultante.

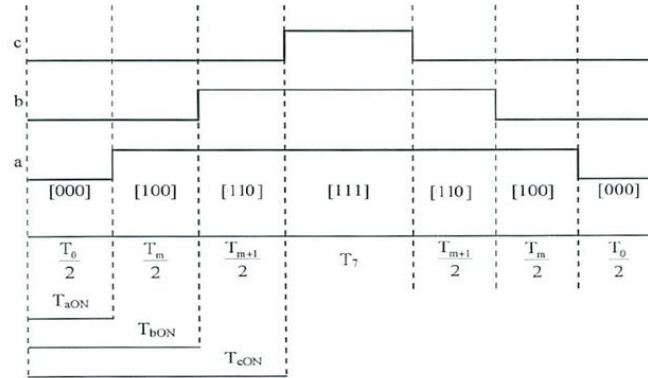


Figura 2.7 Representación gráfica de los estados del inversor durante un tiempo de servicio T.

En la Figura 2.7 se puede observar como los estados del inversor para generar un vector en el sector 1 durante un tiempo de servicio T son:

[0,0,0] [1,0,0] [1,1,0] [1,1,1] [1,1,1] [1,1,0] [1,0,0] [0,0,0]
Inact V1 V2 Inact Inact V2 V1 Inact

En conclusión el problema de la modulación vectorial se resume en, **dadas las proyecciones ALFA y BETA** sobre los ejes real e imaginario del vector que representa las tensiones trifásicas que se quiere aplicar al sistema, **se calcula en que sector se encuentra** este vector atendiendo a las relaciones entre sus proyecciones, por ejemplo si $RAIZ(3) * ABS(alfa) > ABS(beta)$ y $alfa > 0$ el vector pertenece al sector 1.

A continuación **se calculan los tiempos de disparo [Ta Tb Tc]** para cada una de las ramas del inversor, la representación gráfica de estos tiempos tiene forma “de corazón”. Si se realiza la resta entre cada uno de ellos tomados de dos en dos la forma resultante es de seno como se observa en la figura 2.8 (ya que a las fases de la máquina se aplica la diferencia entre ellos). Estos cálculos se realizan conforme a las ecuaciones que se encuentran en [1] y en las líneas de código de la función en el anexo 6.1.

La función recibe como entrada la tensión de bus porque representa el vector de tensión a aplicar al sistema en tanto por uno respecto al valor máximo que puede aplicar, que coincide con $V_{bus}/RAIZ(3)$ (valor de línea). De esta forma se puede considerar que medio tiempo de servicio es 1 y simplificar los cálculos representando los tiempos de disparo como tanto por uno respecto a dicho tiempo de servicio.

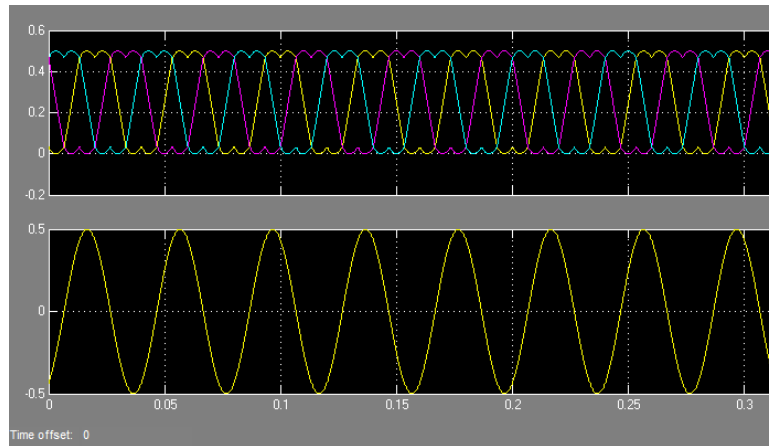


Figura 2.8: Representación gráfica de los tiempos de disparos [Ta Tb Tc] (arriba) y [Ta-Tb] (abajo) para una señal senoidal de amplitud máxima y 25 Hz.

Para realizar los disparos de los IGBTs del inversor el sistema de control realiza **comparaciones en el DSP** de la tarjeta de control MTC4011. Este DSP funciona a una frecuencia de **150 MHz**, es decir, añade 1 al valor de la señal moduladora con la que se comparan los tiempos de disparo $150 \cdot 10^6$ veces por segundo y por tanto si queremos controlar a 4 KHz el valor máximo de las comparaciones será de:

$$\frac{150 \text{ MHz}}{4 \text{ KHz}} = 37500 \text{ (ec. 2.10)}$$

Sin embargo, dado que el comparador ha de tener forma de triángulo como se aprecia en la figura 2.9 **este valor se reduce a la mitad** ya que la cuenta se realiza tanto sumando hasta alcanzar el valor máximo como restando hasta regresar a 0, siendo el valor máximo de la comparación de 18750.

$$\frac{37500}{2} = 18750 \text{ (ec. 2.11)}$$

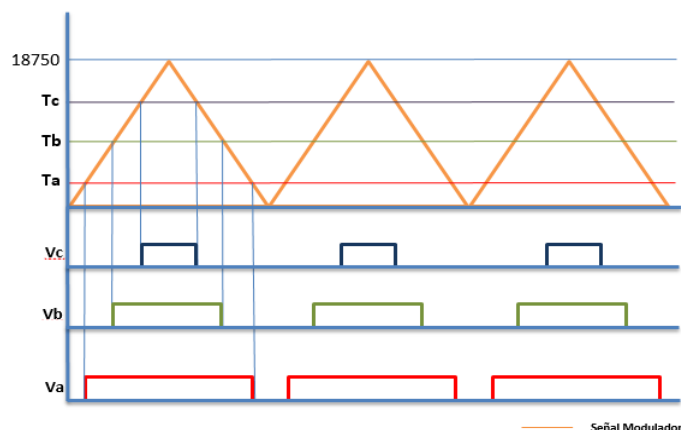


Figura 2.9: Representación gráfica de la comparación de los tiempos de disparo con la señal moduladora en del DSP de la tarjeta MTC-4011.

Por tanto para que sea posible realizar estas comparaciones los tiempos de disparo [Ta, Tb, Tc] que están representados internamente en tanto por uno respecto al tiempo de servicio T deben multiplicarse por una ganancia $\text{gain}=2 \cdot X_{\text{max}}$ que representa el máximo número de la comparación, depende de la frecuencia de control y es una entrada a la función, obteniendo las salidas [Taout, Tbout, Tcout].

2.2.12) MODELO_DE_FLUJO.

Entradas: IR, IS, IT, SPEED.

Salidas: ID, IQ, ANGULO.

Variables Internas: ia, ib, wmr, imr

Funcionamiento: Esta función calcula la velocidad angular del flujo magnético en el rotor Wmr y la integra para calcular su posición (ANGULO) en cada instante. Con este valor realiza las transformaciones de Clarke y Park de las corrientes de entrada al motor [IR IS IT] para obtener las corrientes [ID IQ] que están directamente relacionadas con el flujo magnético en el rotor y el par que entrega el motor a la carga respectivamente.

Controlando estas corrientes es posible realizar el control del motor asíncrono como si se tratase de un motor de corriente continua en bucle de par (Control Vectorial). Para realizar estos cálculos la función necesita las corrientes de entrada al motor IR IS IT y la velocidad angular SPEED del rotor (medida en radianes/segundo) además de los valores de la constante de tiempo del rotor T_r y el número de pares de polos del motor (2). Las ecuaciones implementadas en esta función son las correspondientes al diagrama de bloques de la figura 2.10.

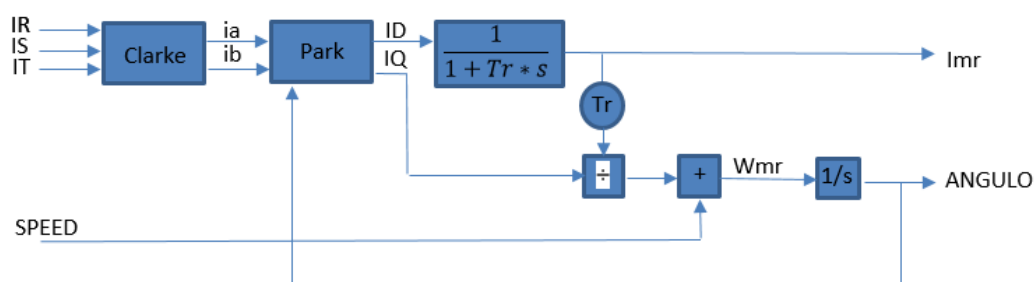


Figura 2.10: Diagrama de bloques del modelo de flujo

2.2.13) par_iq

Entradas: PAR, Lr, Lm, Imr

Salidas: IQ

Funcionamiento: Esta función calcula la corriente de par IQ asociada a un determinado par y a una determinada corriente magnetizante del rotor Imr. Para ello utiliza las inductancias del motor Lr y Lm. Se emplea en el control vectorial para generar una referencia de corriente IQ a partir de una referencia de par obtenida en el bucle de velocidad. La fórmula que emplea se encuentra en [1] y en las líneas de código del anexo 6.1.

2.2.14) radianes_grados

Entradas: RADIANTES

Salidas: GRADOS

Funcionamiento: Esta función transforma un ángulo medido en radianes a su valor en grados.

2.2.15) rad_seg_rpm.

Entradas: RADIANTES

Salidas: RPM

Funcionamiento: Esta función transforma una velocidad angular medida en radianes/segundo a revoluciones/minuto. Es de gran utilidad para interpretar y graficar los datos del sistema dado que este trabaja en rad/s.

2.2.16) rad_seg_hz

Entradas: RADIANTES

Salidas: HZ

Funcionamiento: Esta función convierte la velocidad angular de entrada medida en radianes/segundo a hertzios.

2.3) Bloques de entradas/salidas

A lo largo de esta sección se describen los bloques de entradas y salidas de la librería. La descripción de estos no va a ser tan minuciosa como la de los bloques de control ya que su función simplemente es la de leer o escribir una variable determinada a través de un puntero. Los nombres de los bloques están escritos en inglés de acuerdo al estándar de la empresa.

Para poder leer o escribir estas variables **los bloques necesitan como entrada el SLOT** en el que se está trabajando además del **valor nominal de la variable que miden** debido a que la información está representada en formato Q12 (2^{12} representa el valor nominal de la variable medida). Por tanto para obtener el valor de una entrada en unidades del sistema internacional será necesario aplicar la siguiente fórmula:

$$Variable [S.I] = \frac{Dato\ leído\ en\ memoria}{2^{12}} \times Valor\ nominal \quad (ec. 2.12)$$

Para obtener más información acerca del funcionamiento de estos bloques consúltese el anexo 6.1).

Bloques de entrada:

2.3.1) READ_NOMINAL_VALUES

Este bloque devuelve los valores nominales de las variables del sistema ya que algunos de los otros bloques necesitan saberlos para dar el valor de dichas variables en unidades del sistema internacional, esto es debido al formato en el que llega la información (Q12).

Salidas: Frecuencia nominal de trabajo, corriente nominal en los IGBTs, tensión nominal, corriente nominal, pulsos de encoder nominales.

2.3.2) READ_BUS_VOLTAGE

Este bloque devuelve la tensión en el bus de continua medida en voltios, para ello necesita como entrada el valor de la tensión nominal.

Salidas: BUS_VOL, Tensión en el bus de continua.

2.3.3) READ_CURRENT

Este bloque devuelve la corriente absorbida por cada fase del inversor medida en amperios (que es la misma que la consumida por el motor), para ello necesita como entrada el valor de la corriente nominal.

Salidas: Ir Is It, corrientes en cada una de las fases del inversor.

2.3.4) READ_SPEED

Este bloque devuelve la velocidad angular del rotor del motor medida en radianes/segundo, para ello necesita como entrada el valor de los pulsos de encoder nominales.

Salidas: speed, velocidad medida en radianes/segundo.

2.3.5) READ_PHASE_VOLTAGES

Este bloque devuelve las tensiones de fase medidas por el sistema, se emplea en el SRF-PLL, no necesita parámetros nominales como entrada.

Salidas: Vr Vs Vt, tensiones de fase.

Bloques de salida:

2.3.6) TA_TB_TC_OUT

Este bloque manda a la tarjeta de control de motores MTC 4011 los valores de los tiempos de disparo para los IGBTs [Ta Tb Tc] calculados por el bloque de modulación de 7 segmentos “mod_7s”.

2.3.7) VA_VB_OUT

Este bloque manda a la tarjeta de control de motores MTC 4011 tensiones trifásicas representadas por su transformación de Clarke [Va Vb] para que esta realice la modulación. Este bloque solo se ha empleado en el desarrollo de la librería hasta que se ha completado el desarrollo de los bloques de modulación “mod_7s” y “TA_TB_TC_OUT”, es decir, hasta que la librería ha sido capaz de realizar la modulación por sí misma. Es por eso que este bloque no se emplea en ninguna de las aplicaciones del capítulo 4.

2.3.8) OUT_TO_OSCILSCOPE

Este bloque escribe en memoria las variables deseadas para que el software Pasaban drivers sea capaz de representarlas gráficamente. Posee capacidad para escribir 9 variables y además existe la posibilidad de aplicarles una ganancia para tener más resolución en la representación gráfica.

3 APLICACIONES Y RESULTADOS

En esta sección se presentan tres ejemplos de aplicaciones prácticas que se han desarrollado mediante el empleo de la librería de control descrita en el apartado 2.

El objetivo de este apartado es el de **demostrar la funcionalidad** de la librería a la hora de programar diferentes tipos de controles de potencia y **estudiar y comparar los resultados** de estos con sus respectivas simulaciones de SIMULINK. Para ello se han programado dos tipos de controles para el motor asíncrono VASCAT MAC QI 132 L.

El primero de ellos es un **control tension-frecuencia** simple. Este control ha servido para comprobar el correcto funcionamiento de los distintos bloques de la librería. No obstante como se detalla en la sección 3.1) este sistema no permite realizar control sobre el par entregado por el motor y por tanto no es apto para las máquinas Pasaban. En consecuencia tanto la respuesta a la variación de la referencia como el rechazo de perturbaciones son pobres y de gran brusquedad (haciendo saltar en muchos casos las alarmas de protección del sistema).

El segundo es un **control vectorial** similar a los que actualmente emplean las máquinas Pasaban. Este método de control es más complejo que el control tensión-frecuencia y se caracteriza por su buena respuesta. Este buen comportamiento se debe a que el método permite realizar control sobre el par que entrega el motor a la carga y sobre la corriente de magnetización del rotor.

Finalmente se ha programado un **SRF-PLL** que permite **sincronizar el inversor con la red eléctrica** para de esta forma poder inyectar (o extraer) potencia tanto activa como reactiva de la misma, de igual manera que lo haría un aerogenerador o una placa solar. Esta aplicación ha sido programada con el objetivo de demostrar el potencial de la librería en el campo del almacenamiento y generación de la energía, mercado en el cual la empresa Montelec está buscando nuevos clientes.

3.1) Control F-V

3.1.1) Principio de funcionamiento.

Este método de control es el más básico de los que se pueden realizar en un motor asíncrono y fue el principal método de control hasta los años 90. Como se explica a continuación la respuesta de este control es limitada y es por eso que hasta el desarrollo del control vectorial se empleaban motores de corriente continua en aplicaciones que requirieran gran precisión.

El método consiste en realizar un **control en bucle de velocidad** y como acción de este lazo de control aplicar sobre el motor tensiones trifásicas de amplitud y frecuencia variables. La frecuencia a aplicar al motor será siempre igual a la frecuencia de la velocidad angular eléctrica* del rotor a la cual se suma o resta una frecuencia que es calculada por el control PID y que siempre se encuentra dentro de un rango determinado por el diseñador del control, en el caso de este control se ha determinado experimentalmente que ese rango sea de ± 5 Hz ($\pm 10\pi$ rad/seg).

A cada frecuencia con la que se desee alimentar al motor le corresponde una tensión de acuerdo con la recta que pasa por el punto [0 V, 0 Rad/s] y por la frecuencia y tensión nominales para ese motor. Si la tensión correspondiente a la frecuencia que se desea aplicar es menor que un determinado valor (para el caso del motor VASCAT se han determinado experimentalmente 10 V) la tensión se mantiene constante en ese valor como se indica en la Figura 3.1.1, de lo contrario la acción sobre el motor sería insuficiente.

De la hoja de características del motor asíncrono VASCAT MAC QI 132 L (anexo 6.5) se obtienen los datos de tensión y frecuencia nominales 400V (valor RMS de línea) y 62 Hz, por tanto:

$$400 \text{ V (RMS)} * \sqrt{2} = 565.68 \text{ V de pico} \quad (\text{ec. 3.1})$$

$$\frac{565.68 \text{ V}}{\sqrt{3}} = 326.59 \text{ V (valor de fase)} \quad (\text{ec. 3.2})$$

$$62 \text{ Hz} * 2 * \pi = 124\pi \text{ Rad/seg} \quad (\text{ec. 3.3})$$

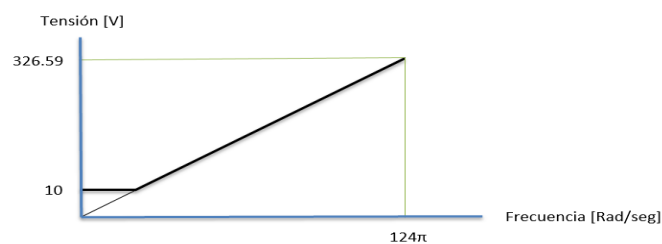


Figura 3.1.1: Representación gráfica de las tensiones y frecuencias para el motor asíncrono VASCAT MAC QI 132 L.

*Velocidad angular eléctrica = Velocidad angular mecánica x Número de pares de polos

Es importante recordar que el motor se encuentra conectado en estrella entre las ramas del inversor y por tanto las tensiones en bornes serán de fase (fase-neutro) y correspondientes a la diferencia entre las tensiones de cada una de las ramas (como se detalla en el anexo 6.2.1).

Las funciones del **bucle de control** descrito en la Figura 3.1.2 son:

- 1: Medir Velocidad.
- 2: Realizar control PID con la diferencia entre la velocidad medida y la velocidad referencia.
- 3: La acción del control PID ha de estar dentro de un rango de $\pm 10\pi$ rad/seg, aplicar saturación y antiwindup en caso de ser necesario.
- 4: Frecuencia a aplicar = Frecuencia de la velocidad eléctrica del rotor \pm Acción PID
- 5: Calcular tensión correspondiente a la frecuencia a aplicar de acuerdo con la regla explicada anteriormente.
- 6: Generar y modular un sistema de tensiones trifásicas de la frecuencia y tensión con las que se desea alimentar al sistema.
- 7: Volver a 1

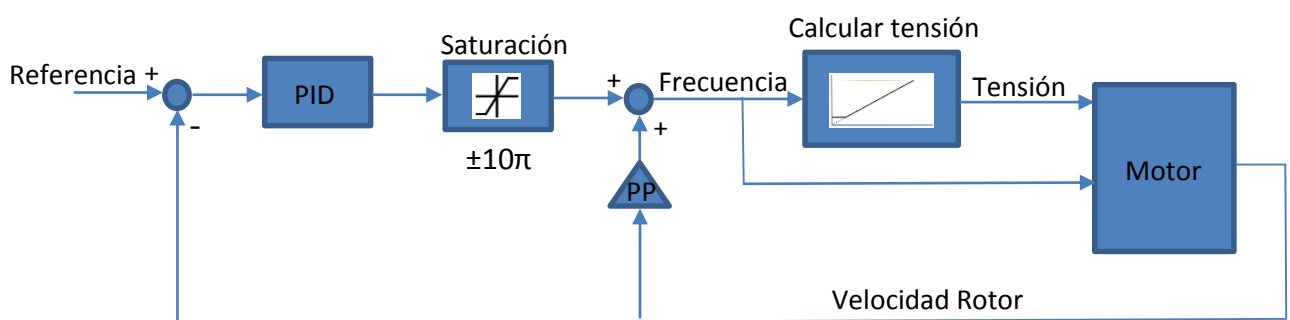


Figura 3.1.2: Diagrama de bloques del control tensión-frecuencia.

3.1.2) Modelo en Matlab del control F-V.

Se ha modelado el sistema con el software SIMULINK para comprobar su respuesta a diferentes referencias en rampa y perturbaciones. Este modelado ha sido realizado según el principio de funcionamiento explicado en la sección 3.1.1) y se observa en las figuras 3.1.3 a) b) c). El modelado del generador de ondas + inversor y el del motor se detallan en los anexos 6.3) y 6.4) respectivamente.

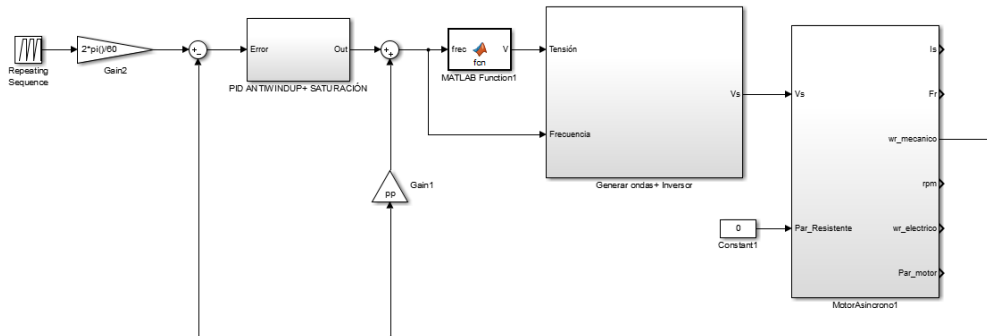


Figura 3.1.3 a): modelado en SIMULINK del control tensión-frecuencia.

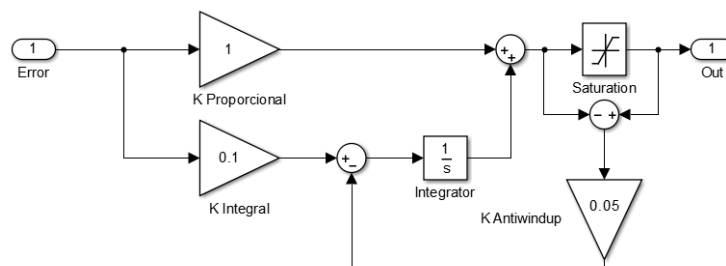


Figura 3.1.3 b): modelado en SIMULINK del controlador PID Antiwindup.

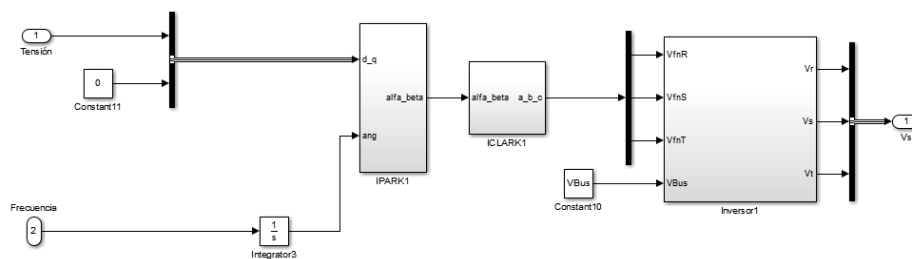


Figura 3.1.3 c): modelado en SIMULINK del generador de ondas + inversor.

3.1.3) Programación del control en CoDeSys.

Para programar este control en CoDeSys se han empleado los distintos bloques de control y de entradas y salidas del sistema descritos en las secciones 2.2) y 2.3) respectivamente.

El programa en CoDeSys es ejecutado secuencialmente y consta de varias partes que se detallan a continuación. Dichas partes se encargan de:

- 1) Leer las variables del sistema y resetearlo. Figura 3.1.4 a)
- 2) Generar la referencia deseada para el sistema. Figura 3.1.4 b)
- 3) Realizar el control F-V. Figura 3.1.4 c)
- 4) Generar y modular las señales senoidales que se desean aplicar al sistema. Figura 3.1.4 d)
- 5) Escribir en memoria las variables que se desean representar gráficamente en el osciloscopio del software Pasaban Drivers. Figura 3.1.4 e)

3.1.3.1) Leer las variables del sistema

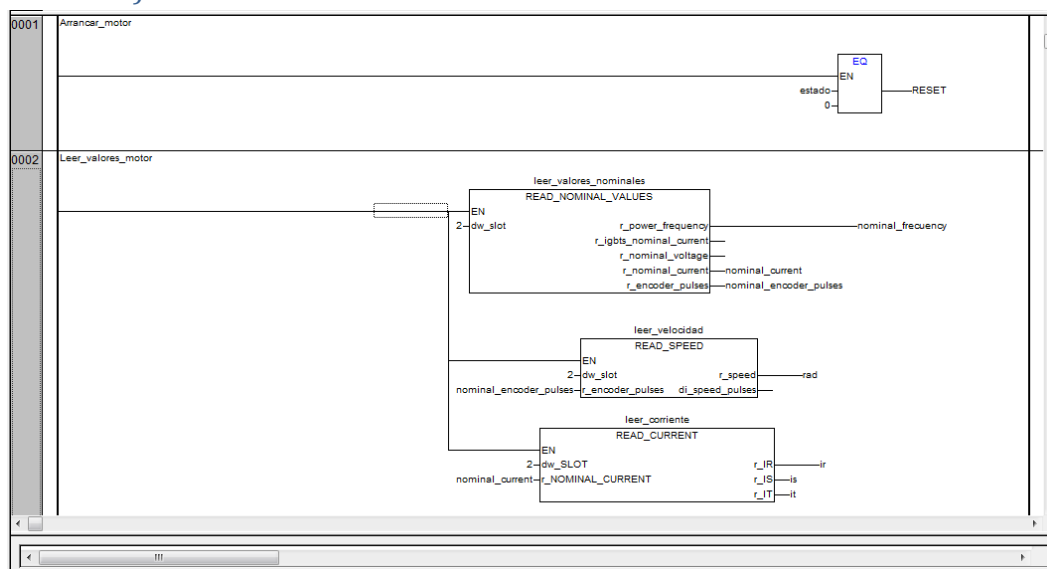


Figura 3.1.4 a): Programa control F-V en CoDeSys 1: Leer valores del motor.

La primera línea hace que la variable “RESET” valga TRUE en caso de que la variable global “estado”=0. La variable global “estado” está asignada a unas condiciones determinadas (arranque, marcha, paro, reset...) desde fuera del programa de control.

En primer lugar se lee la frecuencia a la que trabaja el sistema, las corrientes y los pulsos de encoder nominales con la función READ_NOMINAL_VALUES.

A continuación se lee la velocidad angular del sistema con la función READ_SPEED y se guarda en la variable “rad”. Del mismo modo se leen las corrientes que están entrando al sistema con la función READ_CURRENT y se guardan en las variables “ir” “is” “it”.

3.1.3.2) Generar la referencia para el sistema

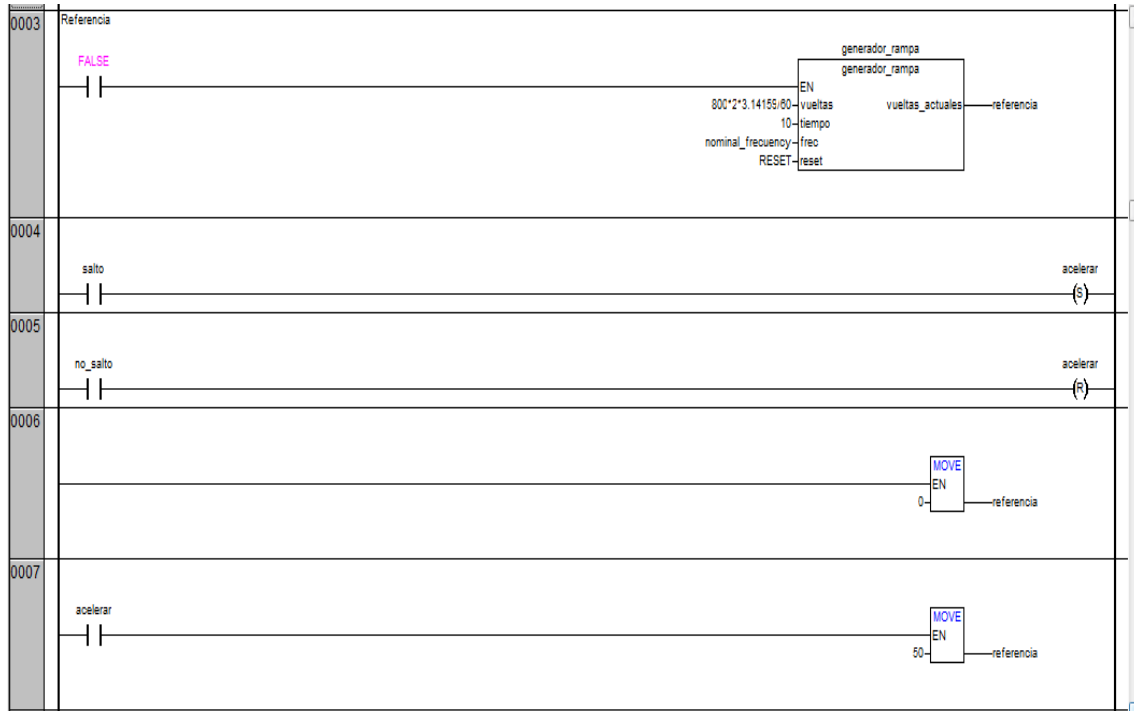


Figura 3.1.4 b): Programa control F-V en CoDeSys 2: Generar Referencia.

Existen **dos posibilidades** para generar la referencia del sistema:

La primera es generar una rampa de las características deseadas con la función de control “generar rampa” y asignar ese valor a la variable “referencia”, nótese que en la Figura 3.1.4 b) este método no se emplea ya que la función “generar rampa” se encuentra deshabilitada con un “false”.

La segunda opción es generar un salto en la referencia mediante la variable “acelerar” que es puesta en SET (S) o RESET (R) dependiendo de las variables “salto” y “no_salto”. Estas a su vez están asignadas a los botones **físicos** “marcha” y “paro” del panel de control del armario de potencia. En caso de que “acelerar” este en modo SET se asigna el valor 50 rad/seg a la variable “referencia”.

3.1.3.3) Control F-V

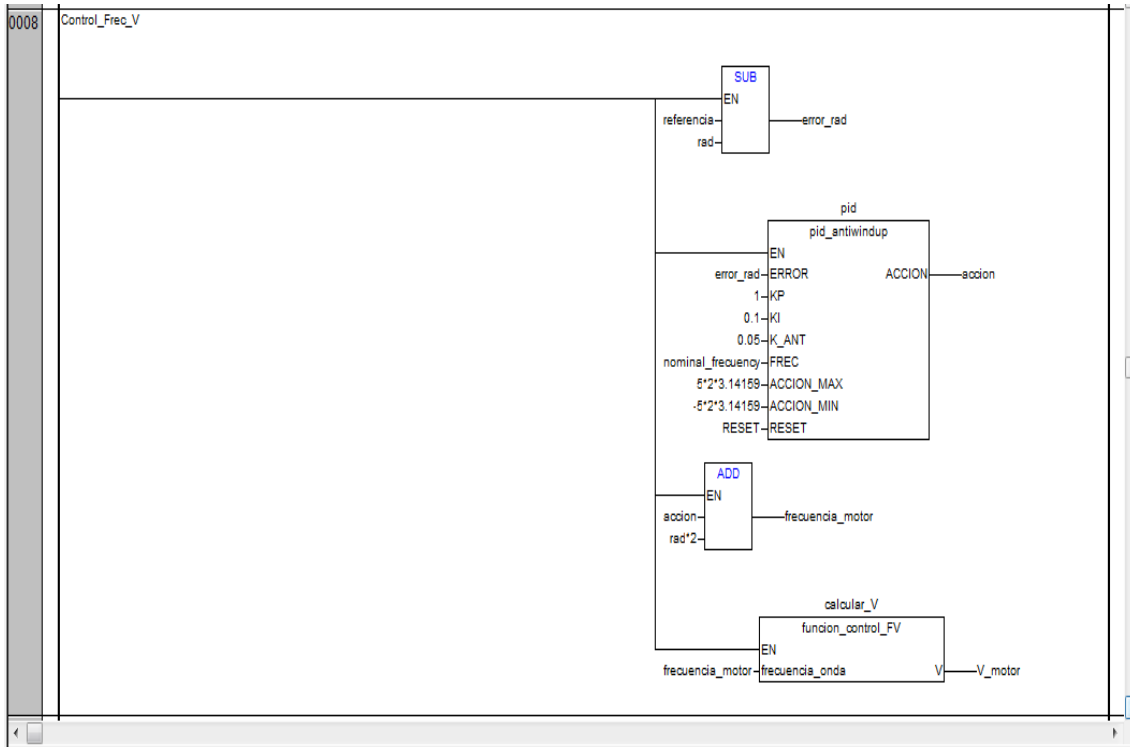


Figura 3.1.4 c): Programa control F-V en CoDeSys 3: control F-V.

En esta parte del programa se realiza el control F-V.

En primer lugar se calcula el “error_rad” restando la “referencia” y la velocidad angular “rad”. A continuación se realiza un control PID con la función “pid_antiwindup”, los valores de las ganancias de control y las limitaciones en la salida pueden observarse en la Figura 3.1.4 c).

En la siguiente línea se suma a la “acción” calculada por el controlador PID la velocidad angular del rotor “rad” multiplicada por el número de pares de polos (2) para calcular la “frecuencia_motor” que queremos aplicar al sistema.

Por último se calcula la tensión “V_motor” que corresponde a esa frecuencia con la “función_control_FV”.

3.1.3.4) Generar y modular las acciones que se aplicaran al sistema.

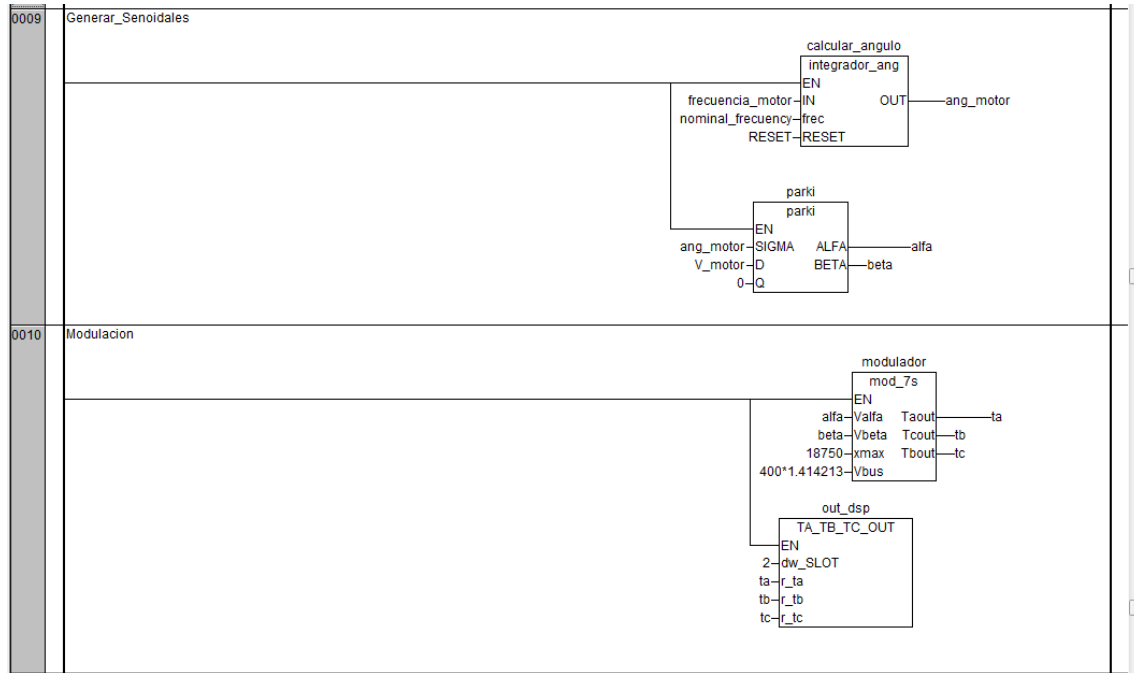


Figura 3.1.4 d): Programa control F-V en CoDeSys 4: Generar y modular tensiones trifásicas.

En este apartado se genera un sistema de tensiones trifásico de la “frecuencia_motor” y la amplitud “V_motor”. Para ello en primer lugar se genera una variable ángulo “ang_motor” que varía con la “frecuencia_motor” integrando esta con la función “integrador_ang”.

A continuación se realiza la transformación de park inversa “parki” para una señal de amplitud “V_motor” en el eje D, 0V en el eje Q y el ángulo “ang_motor”. El resultado es un sistema de tensiones trifásico de la frecuencia y amplitud deseadas representada por sus proyecciones “alfa y “beta”.

Por último se modulan estas señales con la función “mod_7_s” y se obtienen los tiempos de disparo “ta”, “tb”, “tc” para los IGBTs del inversor. Para permitir realizar las comparaciones en el DSP de la tarjeta MTC 4011 se emplea el bloque “TA_TB_TC_OUT”.

3.1.3.5) Escribir variables en memoria para representarlas con el osciloscopio.

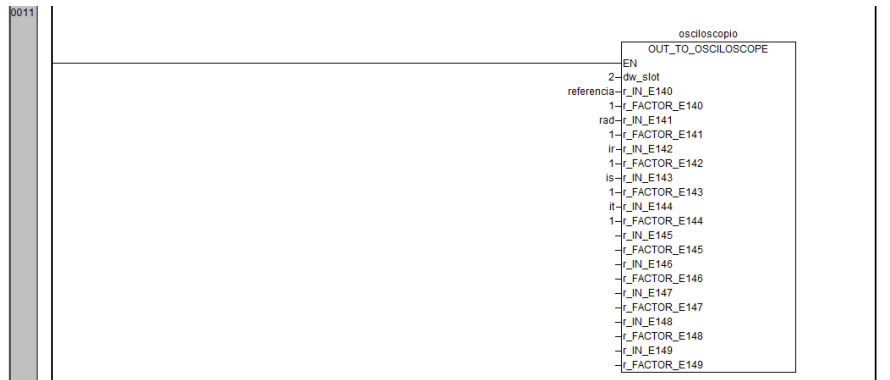


Figura 3.1.4 e): Programa control F-V en CoDeSys 4: Escribir en memoria para representar gráficamente.

En esta parte del programa se escriben en memoria las variables del sistema que se desean representar en Pasaban Drivers para estudiar su respuesta mediante el modulo "OUT_TO_OSCILSCOPE". Estas variables son "referencia", la velocidad del rotor "rad" y las corrientes "ir", "is", "it".

3.1.4) Estudio de resultados.

Se procede a continuación a comparar y comentar los resultados de la simulación y los ensayos experimentales:

3.1.4.1) Comprobación de la validez del modelo en SIMULINK

En primer lugar se comprueba la validez del modelo y su adecuación con el sistema real. Para ello aplicamos al motor unas tensiones trifásicas de amplitud 87V y 105 rad/seg de frecuencia en un ensayo sin par resistente obteniendo los resultados en régimen estacionario de las Figuras 3.1.5 y 3.1.6:

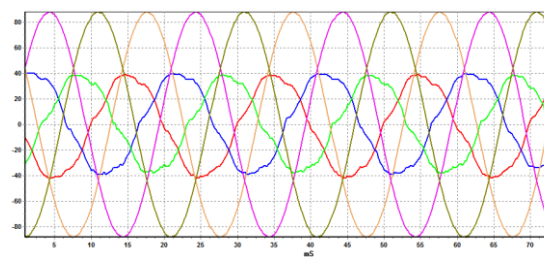


Figura 3.1.5: Corrientes y tensiones en el motor al aplicar 87V y 105 Rad/s en vacío.

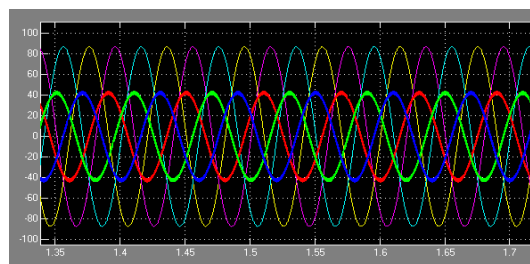


Figura 3.1.6: Simulación de Corrientes y tensiones en el motor al aplicar 87V y 105 Rad/s en vacío.

Se puede observar que el modelado es correcto dado que la simulación es acorde con los resultados experimentales: En ambos casos la corriente consumida por el motor es de 40 amperios de pico. En el caso del resultado experimental las corrientes presentan una forma senoidal levemente deformada como resultado de las vibraciones inducidas en el sistema causadas por la inercia del rotor.

3.1.4.2) Respuesta a rampa de 0 a 800 rpm en 5 segundos.

En las Figuras 3.1.7 y 3.1.8 se observa que la respuesta a la rampa es la esperada:

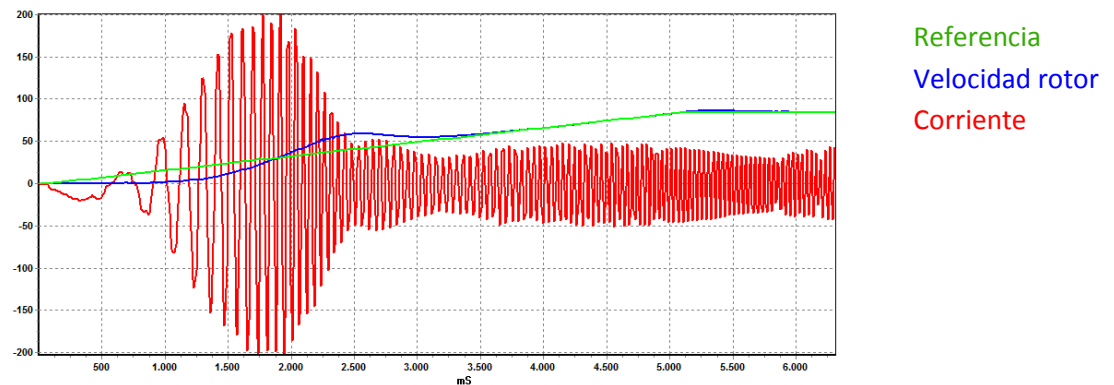


Figura 3.1.7: Respuesta del motor a referencia rampa de 0 a 800 rpm con control F-V.

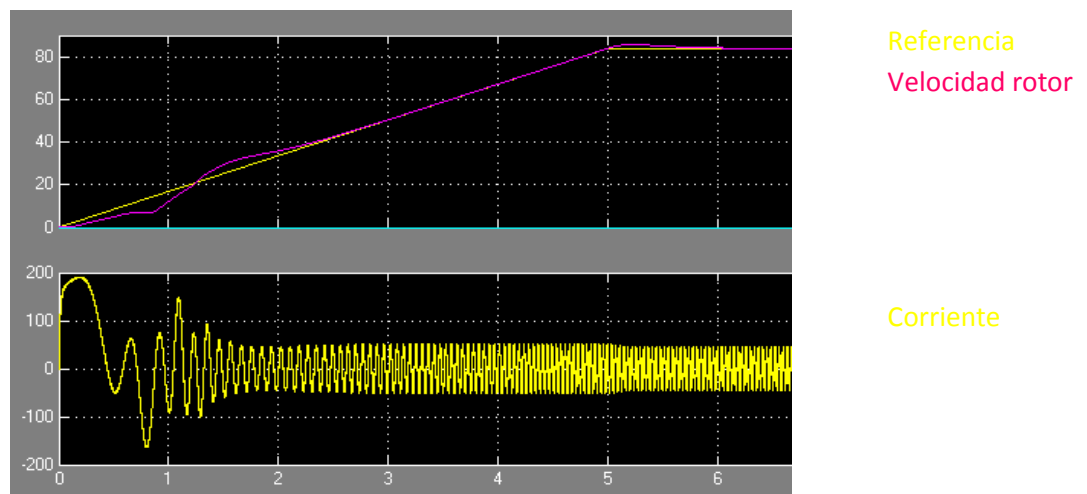


Figura 3.1.8: Simulación de respuesta del motor a referencia rampa de 0 a 800 rpm con control F-V.

En este experimento se ha establecido una referencia en rampa que varía desde 0 hasta 800 rpm (83.77 rad/seg) en 5 segundos ya que si la referencia se establece en forma de escalón la brusquedad del control hace saltar las alarmas de protección de corriente. En ambas gráficas se ha representado la referencia en rampa, la velocidad del rotor en rad/seg y una de las tres corrientes que consume el motor.

Se observa que la respuesta es similar en ambos casos: El sistema comienza teniendo un desfase respecto a la referencia durante 1.5 segundos, a continuación lo compensa en exceso con una sobreoscilación que dura aproximadamente 1 segundo. Cuando esta

termina el motor sigue correctamente la rampa hasta que la referencia se mantiene constante donde se aprecia de nuevo una leve sobreoscilación.

Las corrientes presentan en ambos casos valores de pico similares (50 amperios) durante el periodo en el cual el error es pequeño. Cuando el error es grande se dan ciertas diferencias en el valor de las corrientes entre la simulación y el resultado experimental: Mientras que en el experimento las corrientes son de orden de 150 amperios llegando incluso a 200 en la simulación son del orden de 100 llegando a un valor máximo de 150 amperios.

Nuevamente se pueden explicar estas diferencias atendiendo a fenómenos no tenidos en cuenta en el modelado (como vibraciones mecánicas propias del sistema). Además se ha de tener en cuenta la no linealidad de la respuesta de par del motor, lo cual dificulta la simulación.

3.1.4.3) Corriente magnetizante consumida y par entregado por el motor.

En la Figura 3.1.9 se representan la corriente magnetizante en el rotor y el par entregado por el motor además de la velocidad del rotor y la referencia.

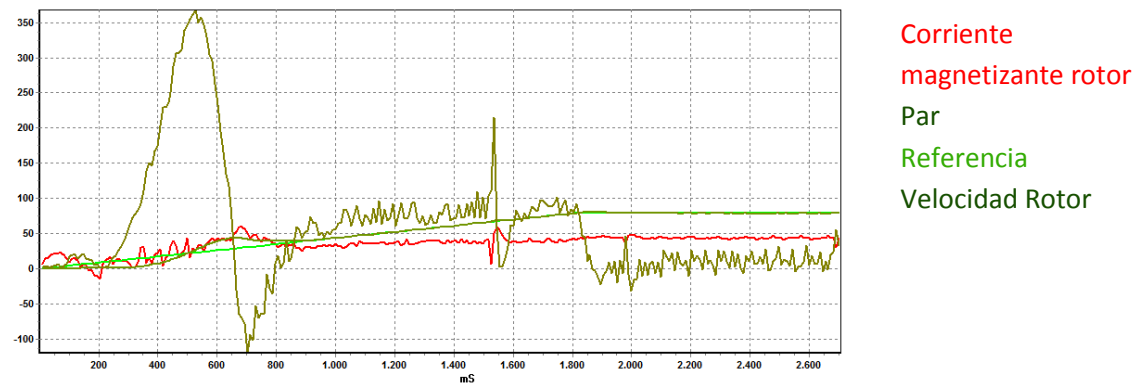


Figura 3.1.9: Respuesta del motor a referencia rampa de 0 a 800 rpm con control F-V.

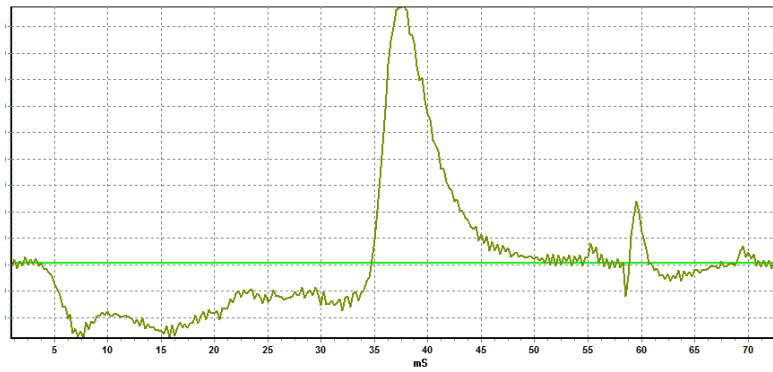
Se observa que la corriente magnetizante se mantiene en torno a un valor de 40 amperios. De la hoja de características del motor se calcula el valor teórico:

$$I_{\text{magnetizante}} = 28.8 \text{ A (RMS)} \times \sqrt{2} = 40.72 \text{ Amperios (ec. 3.4)}$$

Así mismo se observa que el par entregado por el motor es alto cuando es alta la diferencia entre la velocidad del rotor y la referencia para poder acelerarlo, por el contrario cuando se produce sobreoscilación el par es negativo para frenarlo. Cuando la velocidad y la referencia son iguales y la aceleración es 0 el par desarrollado es bajo ya que el motor solo tiene que vencer la fricción dinámica del rotor.

Para calcular el par y la corriente magnetizante en el rotor a partir de las variables del sistema (no se pueden medir) se han empleado las ecuaciones que aparecen en [1].

3.1.4.4) Rechazo de perturbaciones.



Referencia
Velocidad Rotor

Figura 3.1.10: Rechazo de perturbaciones del sistema.

En este experimento se ha estudiado el rechazo de perturbaciones del sistema. Para ello se ha aplicado un par resistente para frenar el motor y a continuación se ha liberado. Como se puede observar en la Figura 3.1.10 al aplicar el par resistente el sistema evoluciona para aumentar la velocidad y aproximarse a la referencia, en el momento en que el par de freno se suelta el motor acelera debido a que la acción sobre el en el momento en el que el par estaba actuando era alta. Entonces el control rebaja la acción para que la velocidad regrese a la referencia.

3.1.4) Conclusiones.

En vista de los resultados experimentales se puede concluir que la respuesta de este sistema de control es pobre dado que se caracteriza por una gran brusquedad consecuencia de la imposibilidad de realizar control sobre el par desarrollado por el motor. Es por eso que este tipo de controles se emplean en aplicaciones que no requieren de gran precisión como cintas transportadoras o ventiladores industriales.

Por tanto este método de control no es válido para las máquinas de la empresa Pasaban y en consecuencia se necesita un método de control de mayor calidad para las mismas: El control vectorial, que se desarrolla en la sección 3.2).

3.2) Control Vectorial

3.2.1) Principio de funcionamiento.

Este es el método de control más complejo y de mejor comportamiento de los posibles para un motor asíncrono. Su teoría fue desarrollada en los años 70 pero no fue hasta los años 90 cuando se empezó a emplear en motores asíncronos revolucionando el ámbito de aplicación de estos ya que este método permite controlar el par generado por el motor y su corriente magnética en el rotor para de esta forma obtener una respuesta mucho más precisa y menos brusca.

El método se basa en realizar la transformación de Clarke y Park **con el ángulo del flujo magnético en el rotor** a las corrientes consumidas por el motor. El resultado de estas transformaciones son dos corrientes ID e IQ que están directamente relacionadas con el flujo magnético en el rotor y con el par entregado por el motor y que por tanto pueden ser controladas. En otras palabras: este método permite controlar un motor asíncrono de la misma forma que se controla un motor de corriente continua.

Para poder realizar las citadas transformaciones es necesario estimar el ángulo del flujo magnético en el rotor (ya que este no puede ser medido) a partir de las variables medidas en el motor: las corrientes absorbidas por este y la velocidad angular del rotor. Para este fin se emplea el “modelo de flujo” que se compone de un conjunto de ecuaciones que derivan del modelo eléctrico del motor y se encuentran en la referencia [1].

Este método de control se encuentra representado en el diagrama de bloques de la Figura 3.2.1.

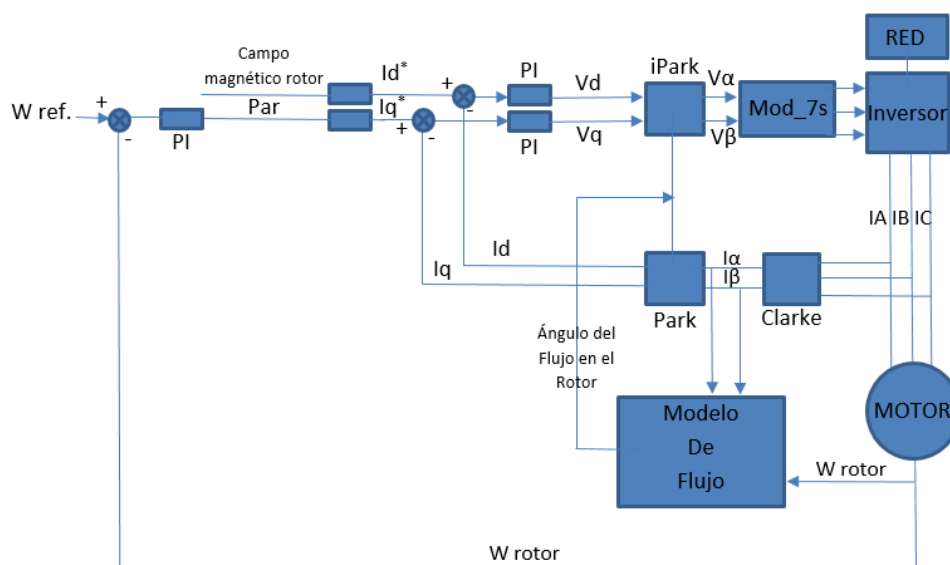


Figura 3.2.1: Diagrama de bloques del control Vectorial.

Cabe mencionar que la transformación de campo magnético en el rotor a I_d^* y de par a I_q^* que se encuentran representadas en el diagrama de bloques de la Figura 3.2.1 por unas cajas rectangulares necesitan el valor del flujo magnético en el rotor (proveniente del modelo de flujo). Esta transformación no se ha dibujado para simplificar la representación del diagrama.

Asimismo se puede mejorar la respuesta del control añadiendo a V_d y V_q unos términos de acoplamiento que también necesitan el flujo magnético en el rotor y tampoco se han representado por la misma razón, no obstante estos términos si se han incluido en el modelado. Para obtener más información sobre el control vectorial consultar la referencia [1].

3.2.2) Modelo en Matlab del control vectorial.

Se ha modelado el sistema con el software SIMULINK para comprobar su respuesta a diferentes referencias y perturbaciones, este modelado ha sido realizado según el principio de funcionamiento explicado en la sección 3.2.1) y se observa en las Figuras 3.2.2 a) b). El modelado del modulador de 7 segmentos y del motor se detallan en los anexos 6.3) y 6.4) respectivamente.

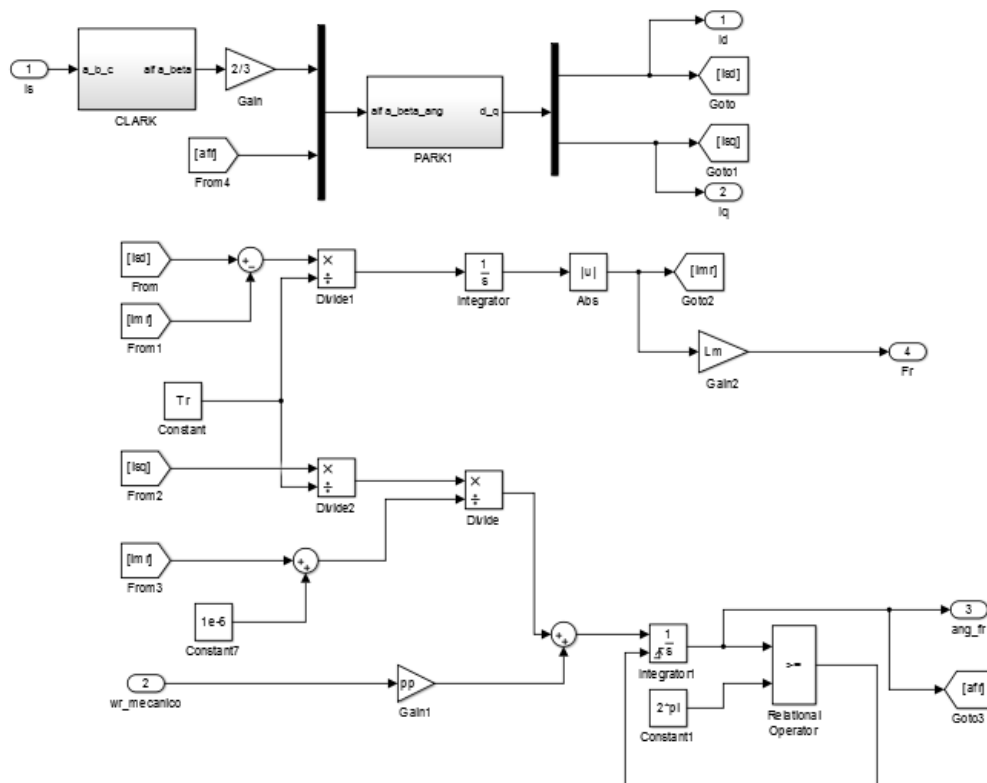


Figura 3.2.2 a): Modelado en SIMULINK del modelo de flujo.

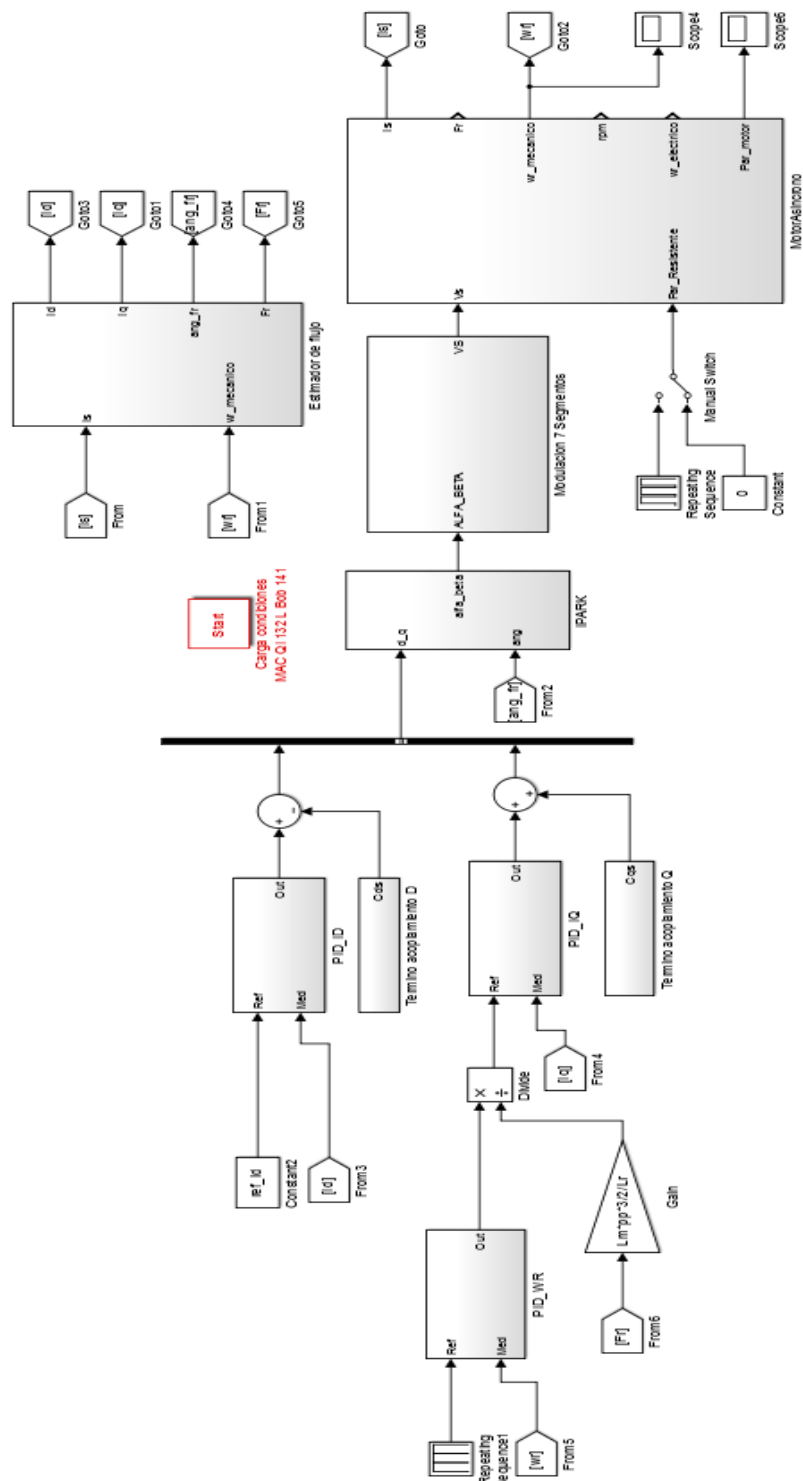


Figura 3.2.2 b): Modelado en SIMULINK del control Vectorial.

3.2.3) Programación del control en CoDeSys.

Para programar este control en CoDeSys se han empleado los distintos bloques de control y de entradas y salidas del sistema descritos en las secciones 2.2) y 2.3) respectivamente.

El programa en CoDeSys es ejecutado secuencialmente y consta de varias partes que se detallan a continuación. Dichas partes se encargan de:

- 1) Leer las variables del sistema y resetearlo. Figura 3.2.3 a)
- 2) Estimar variables con el modelo de flujo. Figura 3.2.3 b)
- 3) Controlar campo magnético en el rotor con I_d . Figura 3.2.3 c)
- 4) Generar la referencia deseada para el sistema. Figura 3.2.3 d)
- 5) Controlar la velocidad. Figura 3.2.3 e)
- 6) Controlar I_q . Figura 3.2.3 f)
- 7) Generar y modular las señales senoidales que se desean aplicar al sistema. Figura 3.2.3 g)
- 8) Escribir en memoria las variables que se desean representar gráficamente en el osciloscopio del software pasaban drivers. Figura 3.2.3 h)

3.2.3.1) Leer las variables del sistema

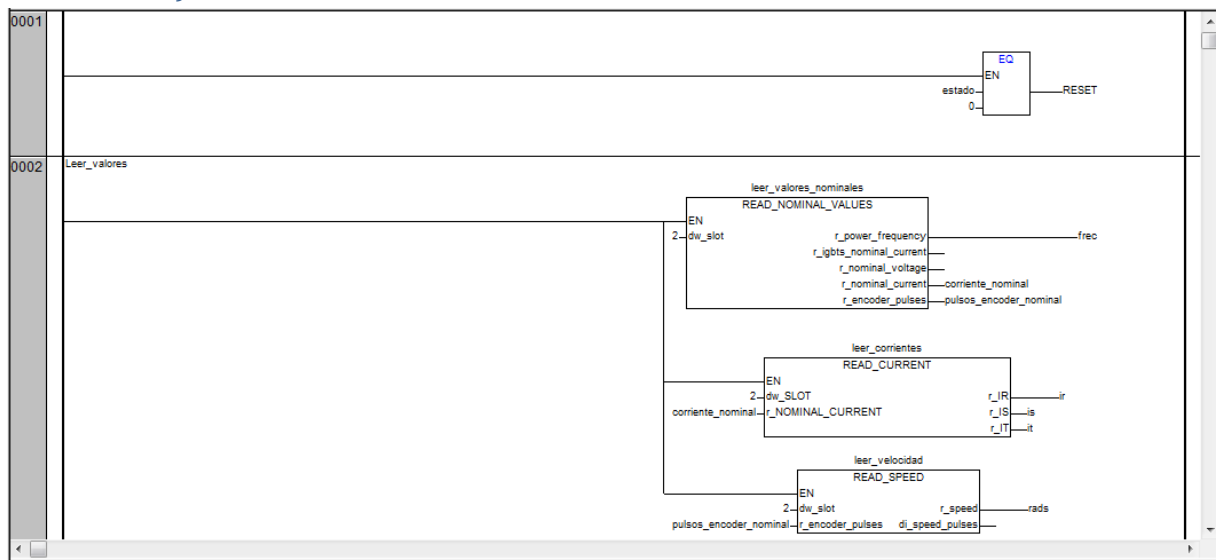


Figura 3.2.3 a): Programa control vectorial en CoDeSys 1: Leer valores del motor.

La primera línea hace que la variable “RESET” valga TRUE en caso de que la variable global “estado”=0. La variable global “estado” está asignada a unas condiciones determinadas (arranque, marcha, paro, reset...) desde fuera del programa de control.

En primer lugar se lee la frecuencia a la que trabaja el sistema, las corrientes y los pulsos de encoder nominales con la función “READ_NOMINAL_VALUES”.

A continuación se lee la velocidad angular del rotor con la función READ_SPEED y se guarda en la variable “rads”. Del mismo modo se leen las corrientes en las fases del motor ir, is, it con la función READ_CURRENT.

3.2.3.2) Estimar variables con el modelo de flujo.

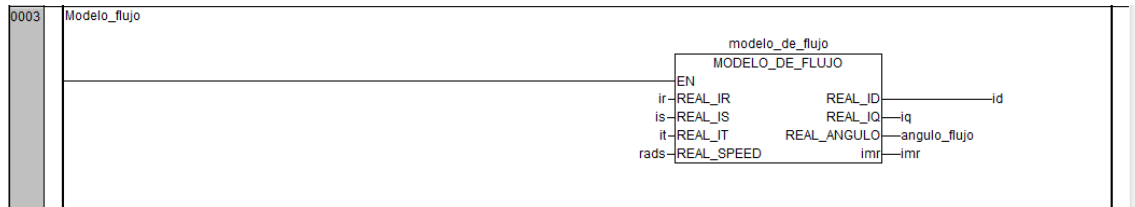


Figura 3.2.3 b): Programa control vectorial en CoDeSys 2: Modelo de flujo.

En este apartado se emplea el “MODELO_DE_FLUJO” para estimar el valor de “id”, “iq”, la intensidad magnetizante en el rotor “imr” y el “angulo_flujo” que más tarde se emplea en las transformaciones de Park.

3.2.3.3) Controlar el campo magnético en el rotor.

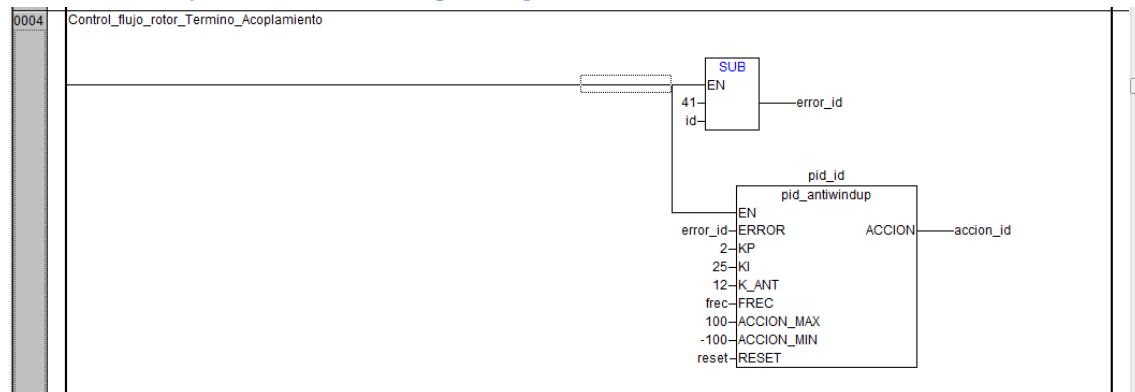


Figura 3.2.3 c): Programa control vectorial en CoDeSys 3: Campo magnético en el rotor.

En primer lugar se calcula el “error_id” como la diferencia entre la corriente asociada al campo magnético “id” y su valor nominal “41”.

Este valor se extrae de la hoja de características del motor:

$$28.8A \text{ (RMS)} \times \sqrt{2} = 40.7 \approx 41 \text{ Amperios (ec. 3.5)}$$

En segundo lugar se realiza un control PID sobre este error con la función de control “pid_antiwindup” y se calcula la “acción_id”.

3.2.3.4) Generar la referencia deseada para el sistema.



Figura 3.2.3 d): Programa control vectorial en CoDeSys 4: Generar referencia.

En esta parte del programa se genera un salto en la referencia mediante la variable “acelerar” que es puesta en SET (S) o RESET (R) dependiendo de las variables “salto” y “no_salto”, estas a su vez están asignadas a los botones **físicos** “marcha” y paro” del panel de control del armario de potencia. En caso de que “acelerar” este en modo SET se asigna el valor 50 rad/seg a la variable “vel_ref”. Al contrario que en la aplicación del control F-V no es necesario generar una rampa ya que la suavidad de este control permite aplicar un salto sin riesgo de que un pico de corriente haga saltar las alarmas.

3.2.3.5) Control de Velocidad.

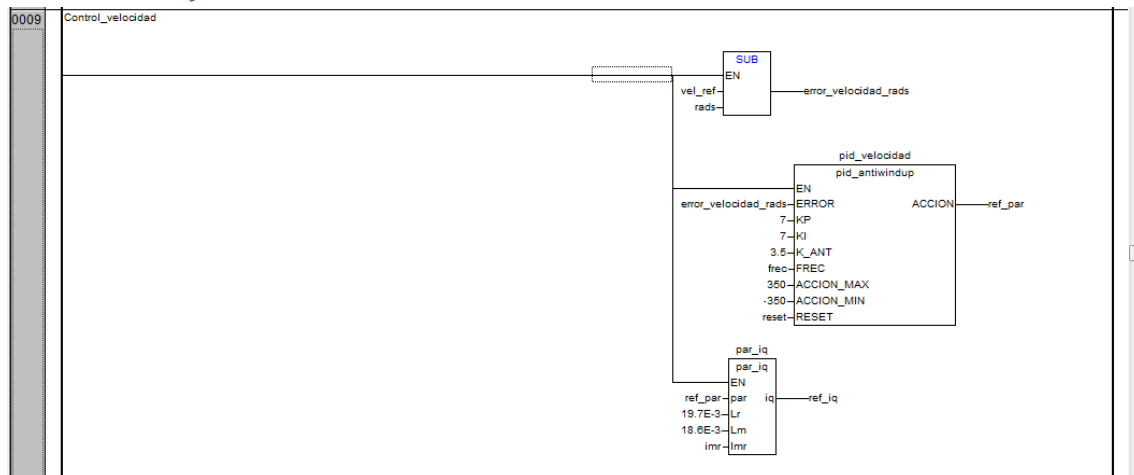


Figura 3.2.3 e): Programa control vectorial en CoDeSys 5: Control de velocidad.

Se calcula el “error_velocidad_rads” como la diferencia entre la referencia “vel_ref” y la velocidad actual “rads”. A continuación se realiza un control PID para este error obteniendo como resultado una referencia de par que debe generar el motor “ref_par”. Este par de referencia tiene una corriente “ref_iq” equivalente. Para calcularla se emplea el bloque de control “par_iq”.

3.2.3.6) Control de Iq.

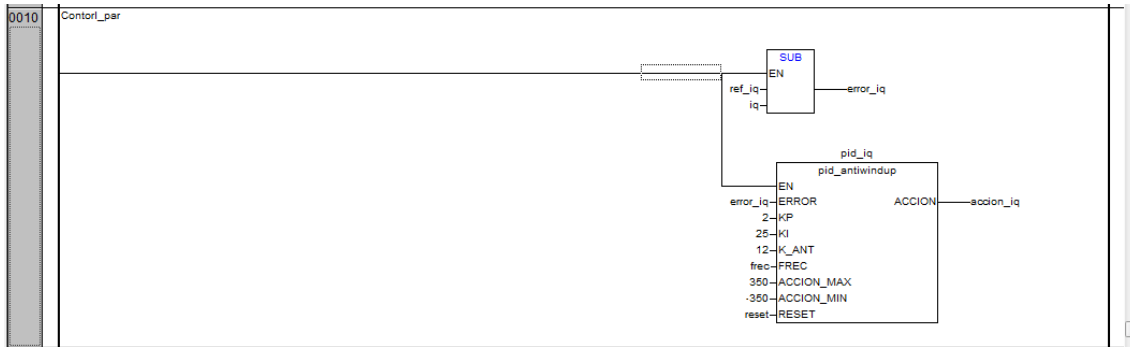


Figura 3.2.3 f): Programa control vectorial en CoDeSys 6: Control de Iq.

Del mismo modo que se ha hecho con Id, se calcula el “error_iq” como la diferencia entre la referencia “ref_iq” y el valor estimado “iq”. A continuación se realiza un control PID para calcular “acción_iq”.

3.2.3.7) Generar y modular las señales que se desean aplicar al sistema.

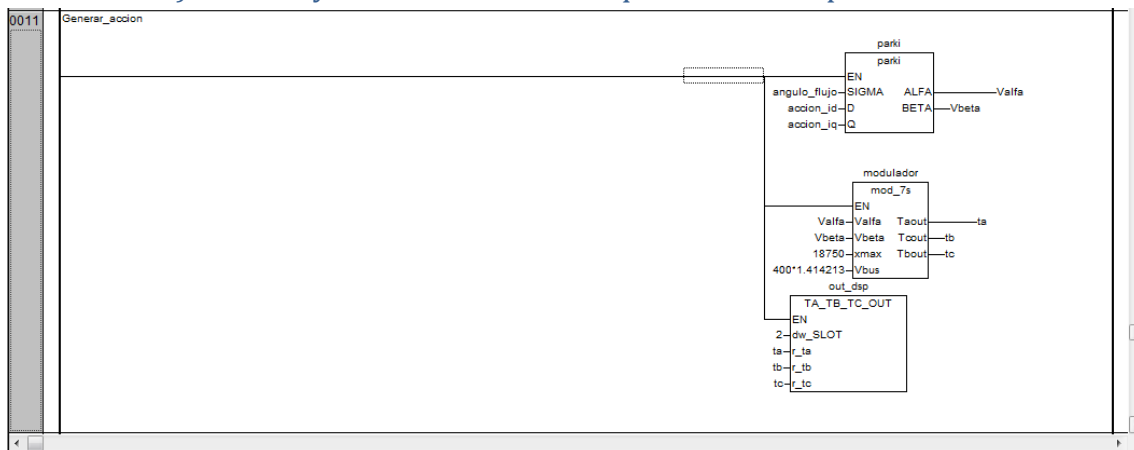


Figura 3.2.3 g): Programa control vectorial en CoDeSys 7: Generar y modular tensiones trifásicas.

Una vez conocidas las acciones “accion_id” y “acción_iq” que se desean inyectar al sistema y que tienen unidades de voltios se realiza la transformación de Park inversa con el “angulo_flujo” estimado para obtener “Valfa” y “Vbeta”. En último lugar se modulan estas tensiones con la función mod_7s y se obtienen los tiempos de disparo “ta”, “tb”, “tc” para los IGBTs del inversor. Para permitir realizar las comparaciones en el DSP de la tarjeta MTC 4011 se emplea el bloque “TA_TB_TC_OUT”.

3.2.3.8) Escribir variables en memoria para representarlas en el osciloscopio.



Figura 3.2.3 h): Programa control vectorial en CoDeSys 8: Escribir en memoria para representar gráficamente.

En esta parte del programa se escriben en memoria las variables del sistema que se desean representar en Pasaban Drivers para estudiar su respuesta mediante el modulo “OUT_TO_OSCILSCOPE”.

3.2.4) Estudio de resultados.

Se procede a continuación a comparar y comentar los resultados de la simulación y los ensayos experimentales. Como se ha demostrado en la sección 3.1.4.1) el modelo en SIMULINK es correcto y acorde a los resultados experimentales.

A lo largo de estos ensayos experimentales solo se han variado las **ganancias del control PID de velocidad**. Esto se debe a que dado que las corrientes son unos parámetros que varían mucho más rápido en el tiempo que la velocidad angular del rotor, los polos y ceros de las corrientes se encuentran dominados por los polos y ceros de la velocidad y por tanto son estos últimos los que determinan el comportamiento del sistema [3].

3.2.4.1) Respuesta a escalón con $K_p=1$ $K_i=4$.

En este experimento se ha determinado la respuesta a un escalón en la referencia de 0 a 50 rad/seg con las constantes del PID de velocidad $K_p=1$ $K_i=4$.

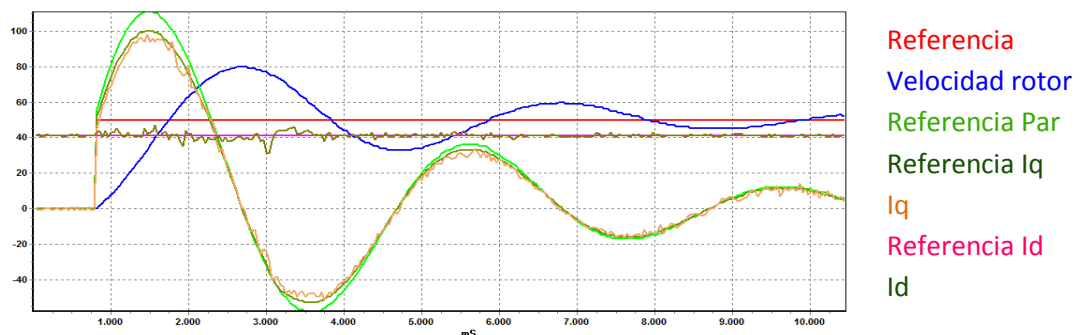
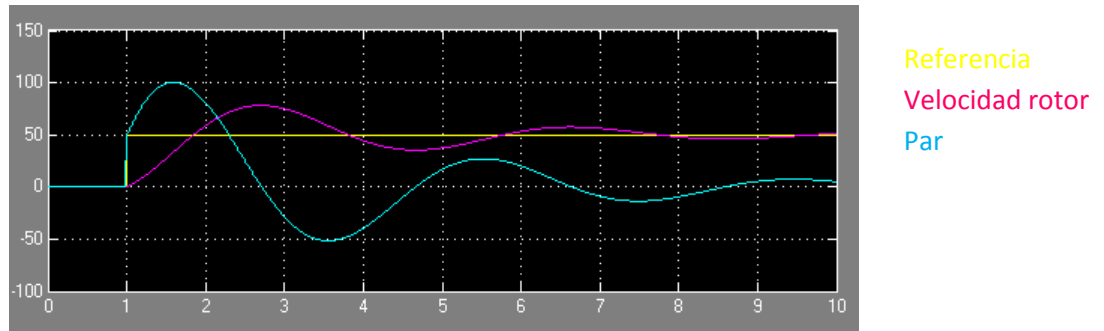


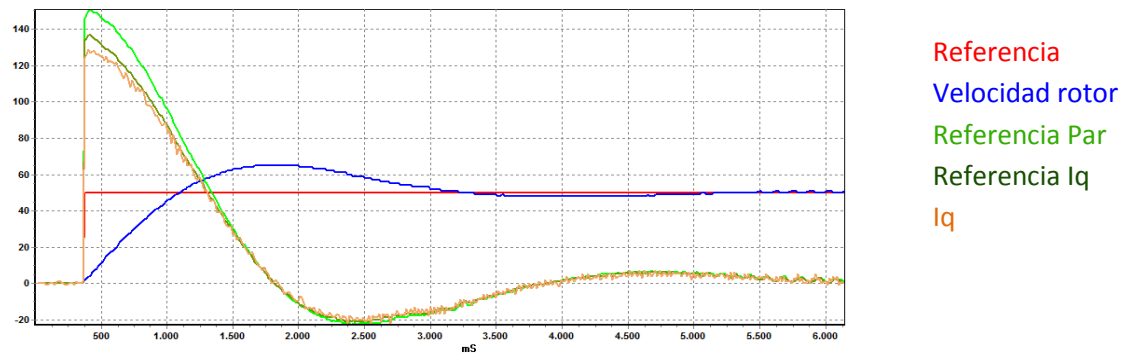
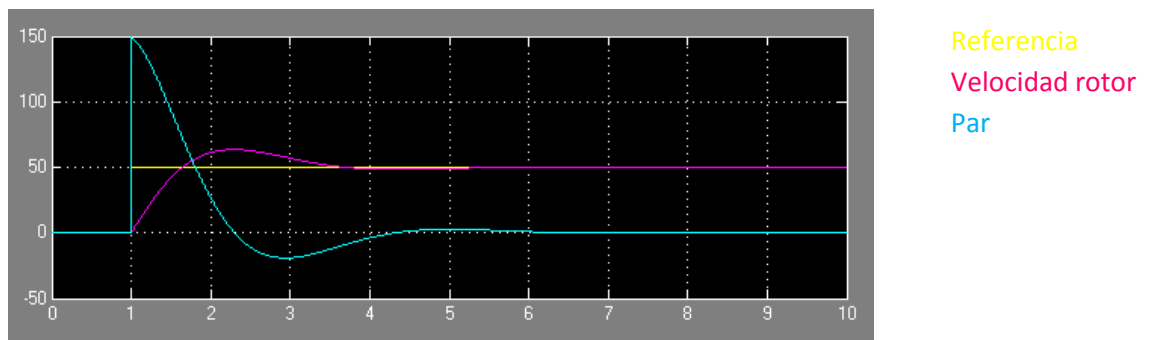
Figura 3.2.4: Respuesta a escalón. $K_p=1$ $K_i=4$

Figura 3.2.5: Simulación respuesta a escalón. $K_p=1$ $K_i=4$

Como se puede observar en las Figuras 3.2.4 y 3.2.5 el resultado experimental y la simulación son coincidentes, no obstante la respuesta del sistema no es óptima ya que presenta una gran sobreoscilación de hasta 80 rad/seg que persiste a lo largo del tiempo. En la Figura 3.2.4 se aprecia como el par calculado por el controlador (referencia de Par) y la referencia de corriente I_q presentan unos comportamientos altamente relacionados y como el valor medido de I_q tiende a seguir el valor de referencia. I_d presenta la misma tendencia a seguir el valor de referencia que se mantiene constante en 41 A dado que el campo magnético en el rotor ha de ser constante.

3.2.4.2) Respuesta a escalón con $K_p=3$ $K_i=5$.

En este experimento se ha determinado la respuesta a un escalón en la referencia de 0 a 50 rad/seg con las constantes del PID de velocidad $K_p=3$ $K_i=5$.

Figura 3.2.6: Respuesta a escalón. $K_p=3$ $K_i=5$ Figura 3.2.7: Simulación respuesta a escalón. $K_p=3$ $K_i=5$

Nuevamente se puede apreciar en Las Figuras 3.2.6 y 3.2.7 que el resultado experimental y la simulación son coincidentes. El comportamiento del sistema en este experimento ha sido mejor que en el anterior, con un tiempo de respuesta de 0.7 segundos y una extinción del régimen transitorio más temprana. Además la sobreoscilación de la velocidad ha llegado a 65 rad/seg frente a los 80 rad/seg alcanzados en el anterior experimento. El par realizado por el motor se encuentra en el orden de los 150 Nm mientras que en el control anterior apenas llegaba a los 110, por tanto este control será más brusco frente al escalón y seguirá mejor la referencia.

3.2.4.3) Freno $K_p=1$ $K_i=2$.

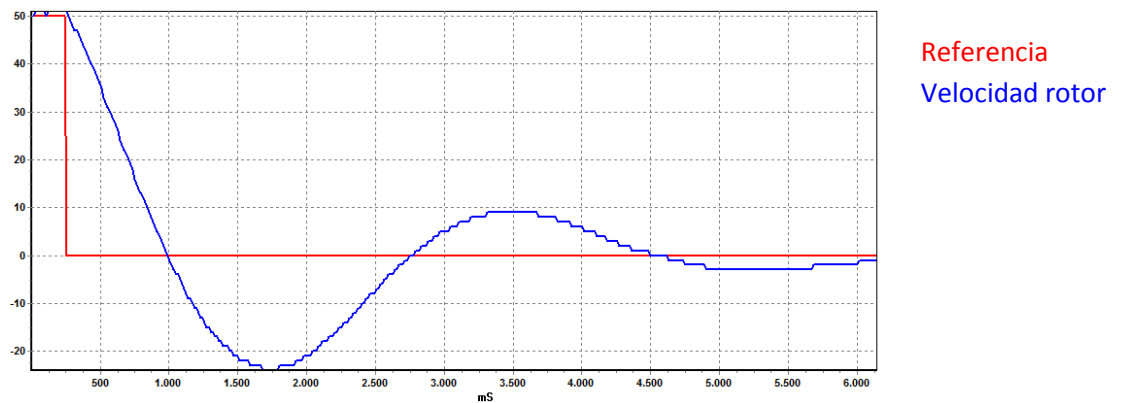


Figura 3.2.8: Freno $K_p=1$ $K_i=2$

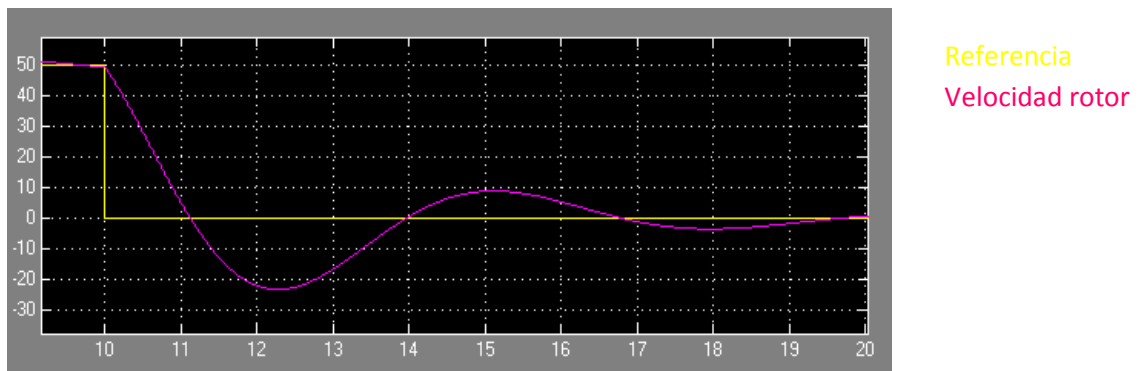


Figura 3.2.9: Simulación freno $K_p=1$ $K_i=2$

En las Figuras 3.2.8 y 3.2.9 se puede observar la respuesta a una frenada del motor. Cabe explicar que no se ha podido realizar un ajuste de las ganancias para obtener una mejor respuesta debido a la imposibilidad de inyectar corriente a la red. Por tanto en caso de hacer disminuir la velocidad en muy poco tiempo la corriente procedente del frenado regenerativo es absorbida por el banco de condensadores aumentando la tensión del bus de continua que alimenta al inversor y haciendo saltar las alarmas de protección.

3.2.5) Conclusiones.

En vista de los resultados experimentales podemos concluir que la respuesta de este control depende del ajuste de las ganancias del PID de velocidad. Por otra parte este método de control permite controlar el motor en bucle de par y por tanto abre un gran abanico de posibles aplicaciones para el motor asíncrono.

Asimismo, este método de control hace que el sistema se comporte de manera lineal, lo cual permite simularlo con mayor precisión y modelos más sencillos.

Por estas razones este es el método de control que emplean los motores de la empresa Pasaban, no obstante para hacerlo operativo se debería realizar un estudio de las ganancias para los PID. Dicho estudio no forma parte del alcance de este proyecto.

3.3) SRF-PLL

3.3.1) Principio de funcionamiento.

Un SRF-PLL es un programa que permite la sincronización con la red eléctrica para de esta forma poder inyectar o extraer potencia activa o reactiva. Este tipo de programas son necesarios en múltiples aplicaciones tanto de generación como almacenamiento de energía como pueden ser aerogeneradores o placas solares.

Sincronizarse con una red eléctrica consiste en saber en qué posición en el tiempo está cada una de sus tres tensiones trifásicas y cuál es su valor de pico. Utilizando la nomenclatura que se ha empleado a lo largo de este proyecto eso implica conocer el ángulo que forma el vector resultante de la transformación de Clarke (posición de las tres tensiones) y su módulo (valor de pico).

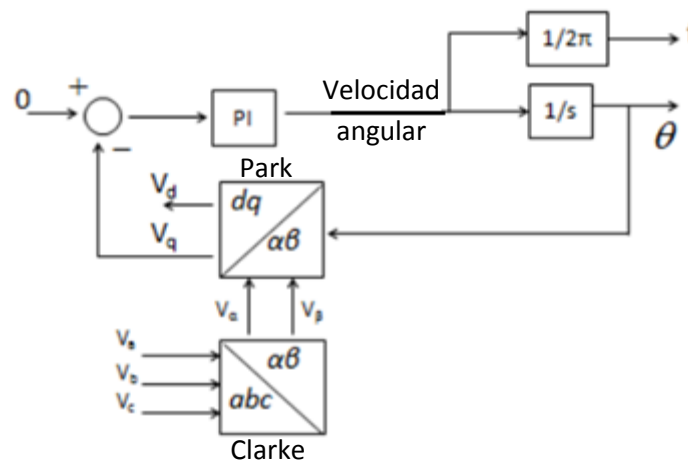


Figura 3.3.1: Diagrama de bloques del SRF-PLL

Como se puede observar en el diagrama de bloques de la Figura 3.3.1 el método consiste en realizar la transformación de Clarke a las tres tensiones V_a V_b V_c medidas en la red para obtener V_α V_β y a continuación realizar la transformación de Park con el ángulo estimado de la posición de estas tensiones θ .

De esta manera se obtienen V_d y V_q . Si se realimenta una de estas dos (V_q por convenio) a un control PI cuya referencia es 0 y cuya acción es la velocidad angular del sistema, cuando V_q esté controlada a 0 voltios V_d tendrá el valor de pico de las tensiones trifásicas y se habrá obtenido la frecuencia angular de la red.

Nótese que para obtener el ángulo de posición θ de las tensiones es necesario integrar la velocidad angular obtenida en el control PI.

3.3.2) Programación del SRF-PLL en CoDeSys

Para programar el SRF-PLL en CoDeSys se han empleado los distintos bloques de control y de entradas y salidas del sistema descritos en las secciones 2.2) y 2.3) respectivamente.

El programa en CoDeSys es ejecutado secuencialmente y consta de varias partes que se detallan a continuación. Dichas partes se encargan de:

- 1) Leer las variables del sistema y resetearlo. Figura 3.3.2 a)
- 2) Ejecutar el programa SRF-PLL. Figura 3.3.2 b)
- 3) Transformar las unidades y guardar variables en memoria para ser representadas gráficamente. Figura 3.3.2 c)

3.3.2.1) Leer las variables del sistema

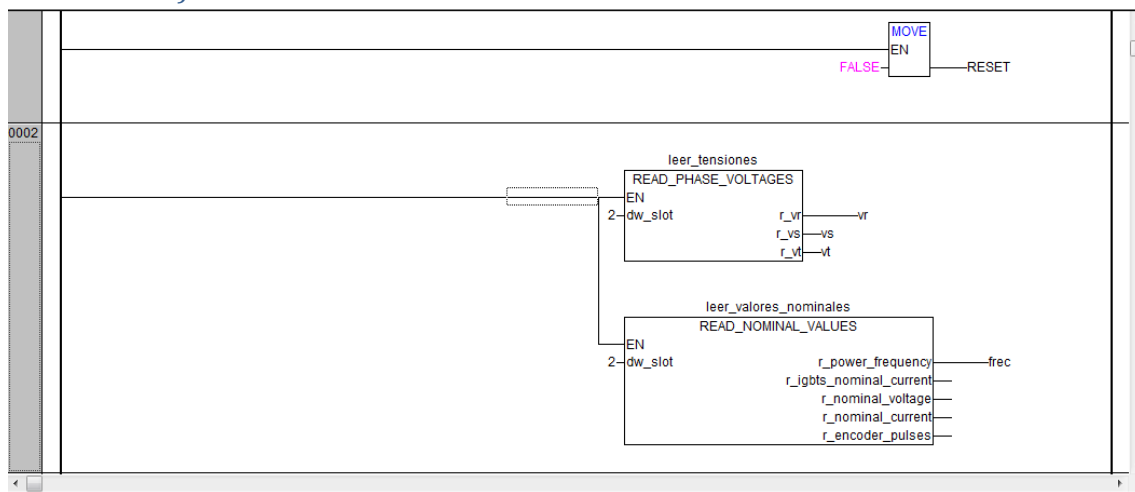


Figura 3.3.2 a): Programa SRF-PLL en CoDeSys 1: Leer variables.

La primera línea hace que la variable “RESET” valga FALSE, este valor puede ser cambiado a TRUE si se desea resetear el sistema.

En primer lugar se leen las tensiones de red “Vr”, “Vs”, “Vt” con la función de e/s READ_PHASE_VOLTAGES. A continuación se lee la frecuencia “frec” a la que trabaja el sistema con la función READ_NOMINAL_VALUES.

3.3.2.2) SRF-PLL

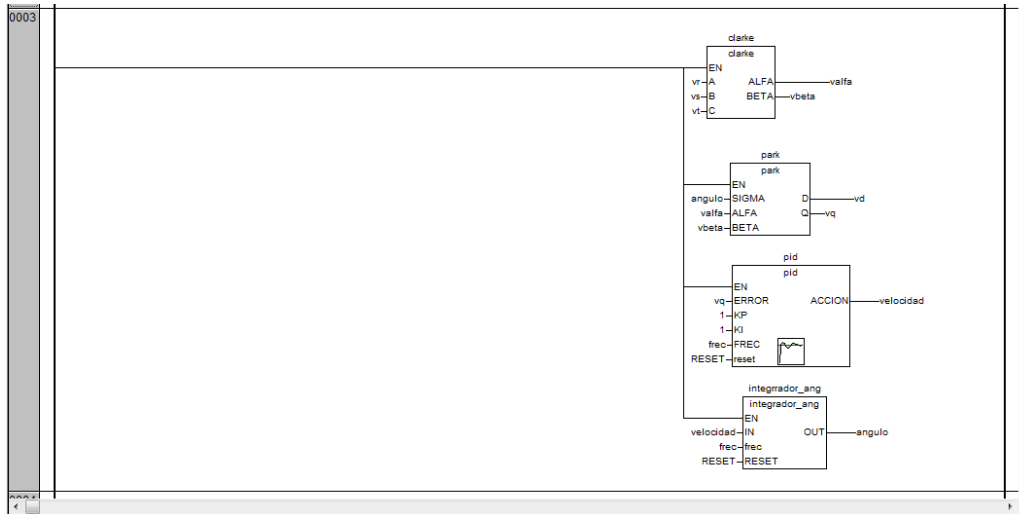


Figura 3.3.2 b): Programa SRF-PLL en CoDeSys 2: SRF-PLL

Esta parte del programa es el SRF-PLL. Realiza la transformación de Clarke de las tensiones medidas para obtener “Valfa” y “Vbeta” y a continuación les aplica la transformación de Park con el “ángulo” estimado para obtener “Vd” y “Vq”. A continuación se realimenta “Vq” como error a un control PID para que este trate de hacer que su valor tienda a 0 y la acción de este controlador es la “velocidad” angular de la red. Para obtener el “ángulo” en el que se encuentra la red es necesario integrar la “velocidad” de esta con el bloque “integrador_ang”.

3.3.2.3) Transformar unidades y guardar en memoria para representar.

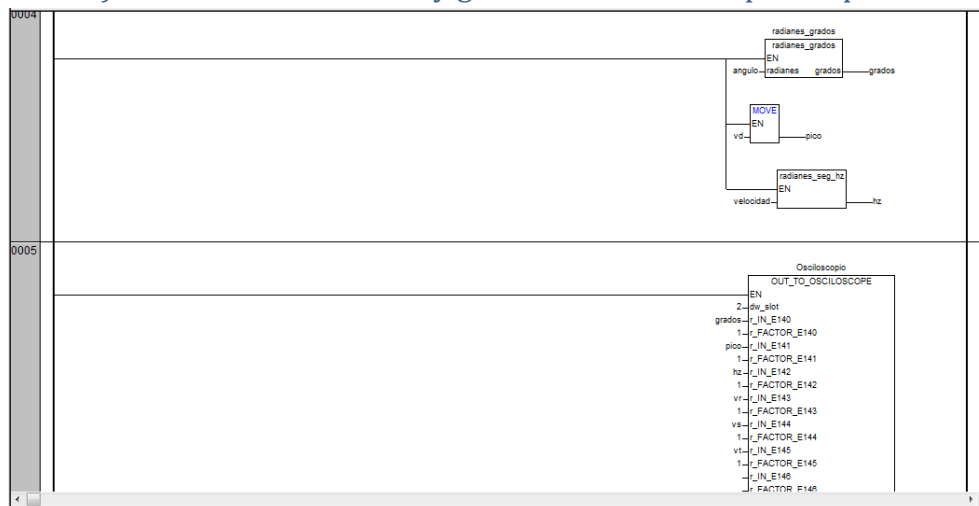


Figura 3.3.2 c): Programa SRF-PLL en CoDeSys 3: Transformar unidades.

En la última parte del programa se transforman las unidades del “ángulo” de radianes a grados y de la “velocidad” de rad/seg a Hz con sus correspondientes bloques de control. A continuación se guardan ciertas variables del sistema en memoria para ser representadas gráficamente con la función “OUT_TO_OSCILOSCOPE”.

3.3.3) Estudio de resultados.

Se ha utilizado el programa descrito en la sección 3.3.2) para medir la red eléctrica que llega a la empresa Montelec obteniendo los resultados mostrados en la Figura 3.3.3:

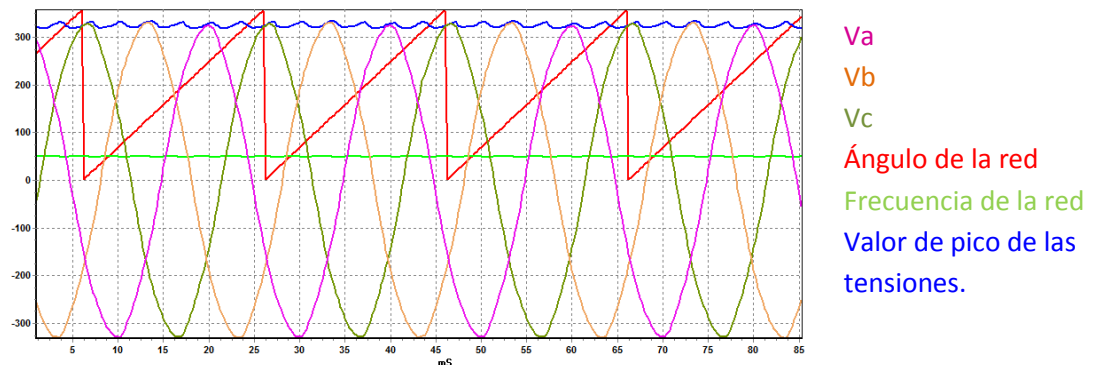


Figura 3.3.3: Resultados del SRF-PLL

En la gráfica se puede observar que el programa ha funcionado correctamente. Para cada ciclo de la red el programa calcula el ángulo de posición que varía entre 0 y 360°. Asimismo la frecuencia se mantiene estable en 50 Hz, valor estándar para Europa. El valor de pico de la tensión de red se mantiene oscilante en torno al valor de 326 V, este valor es correcto de acuerdo a las siguientes ecuaciones:

$$400 \text{ V RMS} \times \sqrt{2} = 565.68 \text{ V de pico (ec. 3.5)}$$

$$\frac{565.68 \text{ V}}{\sqrt{3}} = 326.59 \text{ V Fase - Neutro (ec. 3.6)}$$

Cabe recordar que la tensión que llega a la nave es de 400V RMS valor de línea y que V_s Vs V_t que se miden con la función de e/s "READ_PHASE_VOLTAGES" son valores de fase y se encuentran levemente deformados debido a que el sistema de medida es un prototipo y no está calibrado correctamente, esto produce la oscilación del valor de pico de la red que se observa en la gráfica.

3.3.4) Conclusiones.

El SRF-PLL funciona correctamente y por tanto este método de sincronización es apto para implementarse en diferentes sistemas de inversores conectados a la red eléctrica.

Las aplicaciones de este método son muy amplias, incluyendo entre ellas inyección de energía a la red eléctrica como es el caso de un aerogenerador o un sistema de placas solares. Otra posible aplicación es la de la mejora de la calidad de la red eléctrica con la variación del factor de potencia o el contenido de armónicos.

4 CONCLUSIONES Y PERSPECTIVAS FUTURAS

4.1) Análisis de la respuesta del sistema y aplicabilidad del proyecto

En vista de que los resultados experimentales para el control F-V, el control vectorial y el SRF-PLL han sido satisfactorios y acordes con las simulaciones realizadas previamente se puede concluir que **se puede emplear el PLC Montelec MTC 4000 y la librería que se ha desarrollado en este proyecto para realizar controles de potencia** durante la etapa de desarrollo de los mismos, facilitando de este modo la visualización de errores y estudiando en tiempo real la respuesta del sistema controlado.

Cabe destacar la buena respuesta del control vectorial que además de ser coincidente con la simulación ha demostrado ser apto para el control de los motores de las maquinas Pasaban. Para ello sería necesario realizar un estudio de las ganancias de control adecuadas para dicho sistema.

Del mismo modo la respuesta del control tensión frecuencia ha sido la esperada en la simulación, no obstante este modo de control se encuentra intrínsecamente limitado y es por eso que su respuesta ha sido pobre y brusca haciendo saltar las alarmas de protección por sobrecorriente o sobretensión del bus de continua en repetidas ocasiones.

Por último el SRF-PLL ha funcionado adecuadamente, demostrando así la aplicabilidad de esta librería de control en el ámbito del almacenamiento y generación de energía y apoyando a la empresa Montelec en sus investigaciones en dicho campo.

Es preciso mencionar el abanico de posibilidades que se abre con el desarrollo de esta librería, que permite programar distintos tipos de control en corto espacio de tiempo. Existe la posibilidad de realizar gran cantidad de ejemplos de aplicación aparte de los desarrollados a lo largo de este proyecto, algunos de estos se explican en la sección 4.2).

Desde un punto de vista personal, la realización de este trabajo ha sido de gran ayuda para la comprender el trabajo real que desempeña un ingeniero en el ámbito laboral. Además, el llevar a cabo el proyecto dentro de una empresa dedicada a la investigación y fabricación de elementos electrónicos me ha permitido entender la aplicación industrial de muchas de las asignaturas estudiadas durante la carrera (electrónica de potencia, ingeniería de control, maquinas eléctricas...).

Es por estas razones que se puede concluir que el proyecto ha sido un éxito y que la librería que ha sido desarrollada en el mismo servirá de apoyo al equipo de las empresas Pasaban y Montelec en el desarrollo de proyectos futuros.

4.2) Perspectivas futuras para el proyecto

La facilidad con la que la librería permite desarrollar controles hace que existan una infinidad de aplicaciones industriales de la misma, a continuación se explican brevemente algunas de ellas:

4.2.1) Control vectorial para maquina Síncrona

El control vectorial para un motor síncrono se realiza de forma similar al de un motor asíncrono con la diferencia de que dado que el motor síncrono ya posee un campo magnético en el rotor (procedente de un imán permanente o de una bobina por la que circula una corriente) la corriente I_d ha de ser controlada para tender al valor 0 en lugar de tender al valor de la corriente magnetizante del rotor (esto no es así para todos los motores síncronos).

Además en este control no es necesario modelo de flujo dado que el ángulo del flujo magnético en el rotor viene dado directamente por la posición de este. Por tanto con estos sencillos cambios sería posible emplear el programa de control vectorial para controlar una máquina síncrona.

4.2.2) Control sensorless para maquina asíncrona

El control sensorless es un tipo de control que no necesita medir la velocidad del rotor para controlar la máquina ahorrando de este modo el coste del encoder (que puede llegar a ser muy elevado).

Para ello se emplean unas ecuaciones que parten del modelo eléctrico del motor y estiman la velocidad del rotor a partir de las corrientes absorbidas por el este (de forma similar a como se estima el ángulo del flujo magnético en el control vectorial).

Para realizar este tipo de control hubiese sido necesario programar un bloque de función de control que contenga dichas ecuaciones e incluirlo en el control vectorial ya programado.

4.2.3) Inversor para la conexión de placas solares en un sistema trifásico

Esta aplicación controla la tensión de bus de continua en el inversor. Si a dicho bus de continua se le conecta un equipo de placas solares es posible controlar la tensión de bus en función de la corriente que el equipo está entregando y de esa forma hacer que el equipo de placas solares funcione en el punto de máxima potencia.

Dado que el punto de potencia máxima de unas placas solares viene dado en función de la cantidad de luz que les llega, esta aplicación emplearía el SRF-PLL para inyectar o extraer corriente I_d (asociada a la potencia activa) para controlar la tensión y con ella la potencia en tiempo real.

5 BIBLIOGRAFÍA

En esta sección se referencian los documentos empleados para la realización de este proyecto.

- [1] Peter Vas: Sensorless Vector and Direct Torque Control. Oxford University Press. 1998
- [2] Seung-Ki Sul: Control of electric machine drive system. IEEE Press. 2011.
- [3] Ogata: Ingeniería de control moderna. Pearson Educación. 1998.
- [4] Bimal Bose: Power Electronics and Motor Drives. AP. 2006
- [5] Bibliografía y documentación propia de la empresa Pasaban.

Anexos

6 ANEXOS

6.1) Líneas de código

Funciones de Control:

6.1.1) Clarke

```
ALFA:=A;
BETA:=0.57735*(B-C);
```

6.1.2) Clarkei

```
A:=ALFA;
B:=-0.5*ALFA+0.8660254*BETA;
C:=-0.5*ALFA-0.8660254*BETA;
```

6.1.3) Park

```
D:=COS(SIGMA)*ALFA+SIN(SIGMA)*BETA;
Q:=-SIN(SIGMA)*ALFA+COS(SIGMA)*BETA;
```

6.1.4) Parki

```
ALFA:=COS(SIGMA)*D-SIN(SIGMA)*Q;
BETA:=SIN(SIGMA)*D+COS(SIGMA)*Q;
```

6.1.5) Pid

```
IF reset=TRUE THEN
    acumulado:=0;
END_IF
```

```
acumulado:=acumulado+ERROR/FREC;
ACCION:=ERROR*KP+acumulado*KI;
```

6.1.6) Pid_antiwindup

```
IF reset=TRUE THEN
    acumulado:=0; accion_prop:=0; accion_int:=0; antiwindup:=0; accion_total:=0;
ELSE
    accion_prop:=ERROR*KP;
    accion_int:=ERROR*KI-antiwindup;
    acumulado:=acumulado+accion_int/FREC;
    accion_total:=accion_prop+acumulado;
    ACCION:=accion_total;
    IF accion_total<accion_min THEN
        ACCION:=accion_min;
    END_IF
    IF accion_total>accion_max THEN
        ACCION:=accion_max;
    END_IF

    antiwindup:=(ACCION-accion_total)*K_ANT;
END_IF
```

6.1.7) integrador

```
OUT:=acumulado;
acumulado:=acumulado+IN/frec;
```

6.1.8) integrador_ang

```
OUT:=acumulado;
acumulado:=acumulado+IN/frec;
IF RESET=TRUE THEN
acumulado:=0;
END_IF;
IF acumulado>6.2831853 THEN
    acumulado:=acumulado-6.2831853;
END_IF
```

6.1.9) generador_rampa

```
IF reset=FALSE THEN
    IF acumulado<vueltas THEN
        acumulado:=acumulado+vueltas/(frec*tiempo);
        vueltas_actuales:=acumulado;
    ELSE
        vueltas_actuales:=vueltas;
    END_IF;
ELSE vueltas_actuales:=0;
    acumulado:=0;
END_IF;
```

6.1.10) rad_seg_rpm

```
rpm:=radianes*60/(2*3.14159);
```

6.1.11) funcion_control_FV

```
V:=0.838383*frecuencia_onda;
IF V<10
THEN
    V:=10;
END_IF;
```

6.1.12) mod_7s

```

pi:=3.1415926;
alfa:=Valfa/(Vbus/1.73205);
beta:=Vbeta/(Vbus/1.73205);
gain:=(2*xmax);
IF (*1 *) (* ESTOS TIEMPOS REPRESENTAN EL TIEMPO DE DISPARO DE CADA RAMA, EL TIEMPO TOTAL
EN ACTIVO ES EL DOBLE, DE AHI QUE GAIN MULTIPLIQUE POR 2 *)
beta>0
THEN
  IF 1.73205*ABS(alfa)>ABS(beta)
  THEN (*2 *)
    IF alfa>0 (*3*)
    THEN (*Sector 1*)
      sector := 1;
      Tm :=(alfa*SIN(sector*pi/3)-beta*COS(sector*pi/3));
      Tn :=(beta*COS((sector-1)*pi/3)-alfa*SIN((sector-1)*pi/3));
      Ta:=(1-Tm-Tn)/4;
      Tb:=Ta+Tm/2;
      Tc:=Tb+Tn/2;
    ELSE (*3*) (*Sector 3*)
      sector := 3;
      Tm :=(alfa*SIN(sector*pi/3)-beta*COS(sector*pi/3));
      Tn :=(beta*COS((sector-1)*pi/3)-alfa*SIN((sector-1)*pi/3));
      Tb:=(1-Tm-Tn)/4;
      Tc:=Tb+Tm/2;
      Ta:=Tc+Tn/2;
    END_IF; (*3*)
  ELSE (*2 *)
    sector := 2;
    Tm :=(alfa*SIN(sector*pi/3)-beta*COS(sector*pi/3));
    Tn :=(beta*COS((sector-1)*pi/3)-alfa*SIN((sector-1)*pi/3));
    Tb:=(1-Tm-Tn)/4;
    Ta:=Tb+Tn/2;
    Tc:=Ta+Tm/2;
  END_IF; (*2 *)
ELSE (*1 *)
  IF 1.73205*ABS(alfa)>ABS(beta)
  THEN (*2*)
    IF alfa>0 (*3*)
    THEN (*Sector 6*)
      sector := 6;
      Tm :=(alfa*SIN(sector*pi/3)-beta*COS(sector*pi/3));
      Tn :=(beta*COS((sector-1)*pi/3)-alfa*SIN((sector-1)*pi/3));
      Ta:=(1-Tm-Tn)/4;
      Tc:=Ta+Tn/2;
      Tb:=Tc+Tm/2;
    ELSE (*3*) (*Sector 4*)
      sector := 4;
      Tm :=(alfa*SIN(sector*pi/3)-beta*COS(sector*pi/3));

```

```

    Tn :=(beta*COS((sector-1)*pi/3)-alfa*SIN((sector-1)*pi/3));
    Tc:=(1-Tm-Tn)/4;
    Tb:=Tc+Tn/2;
    Ta:=Tb+Tm/2;
    END_IF;(*3*)
ELSE (*2*)
    sector := 5;
    Tm :=(alfa*SIN(sector*pi/3)-beta*COS(sector*pi/3));
    Tn :=(beta*COS((sector-1)*pi/3)-alfa*SIN((sector-1)*pi/3));
    Tc:=(1-Tm-Tn)/4;
    Ta:=Tc+Tm/2;
    Tb:=Ta+Tn/2;
    END_IF (*2*)
END_IF; (*1 *)

IF Ta>0.5 THEN
Ta:=0.5;
END_IF

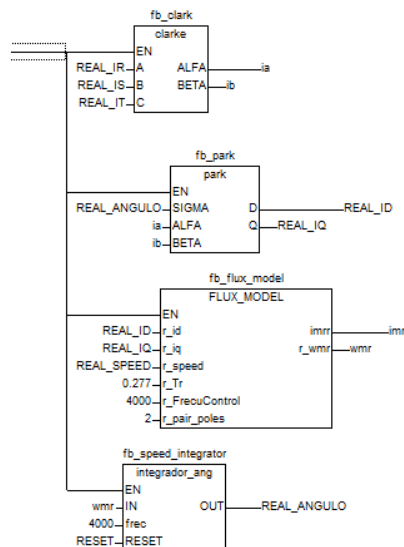
IF Tb>0.5 THEN
Tb:=0.5;
END_IF

IF Tc>0.5 THEN
Tc:=0.5;
END_IF

Taout:=Ta*gain;
Tbout:=Tb*gain;
Tcout:=Tc*gain;

```

6.1.13) modelo_de_flujo



6.1.14) radianes_grados

grados:=radianes*360/(2*3.14159);

6.1.15) rad_seg_hz

hz:=radianes/(6.2831);

6.1.16) par_iq

iq:=par*2*Lr/(3*Imr*Im*Im);

Funciones de Entradas/Salidas:

6.1.17) READ_NOMINAL_VALUES

```

_pointer := 16#11000000; (* RACK base address *)
p_pointer := p_pointer + dw_slot * 16#10000; (* MTC4011 base address *)
p_pointer_frequency := p_pointer + 16#608; (* PWM module frequency pointer *)
p_pointer_igbts_nominal_current := p_pointer + 16#60A; (* Igbts nominal current pointer *)
p_pointer_nominal_voltage := p_pointer + 16#60C; (* Motor nominal voltage pointer *)
p_pointer_nominal_current := p_pointer + 16#60E; (* Motor nominal current pointer *)
p_pointer_encoder_pulses := p_pointer + 16#612; (* Encoder pulses per turn pointer *)
r_power_frequency := INT_TO_REAL(p_pointer_frequency^);
r_igbts_nominal_current := INT_TO_REAL(p_pointer_igbts_nominal_current^);
r_nominal_voltage := INT_TO_REAL(p_pointer_nominal_voltage^);
r_nominal_current := INT_TO_REAL(p_pointer_nominal_current^);
r_encoder_pulses := INT_TO_REAL(p_pointer_encoder_pulses^);

```

6.1.18) READ_BUS_VOLTAGE

```

p_pointer := 16#11000000; (* RACK base address *)
p_pointer := p_pointer + dw_slot * 16#10000; (* MTC4011 base address *)
p_pointer_vbus := p_pointer + 16#206; (* VBUS current pointer *)
vbus_q12 := p_pointer_vbus^; (* VBUS read *)
r_BUS_VOL := r_NOMINAL_VOLTAGE*vbus_q12/2896.309376 (* Ireal=Nominal*sqrt(2)*Iq12/2^12 *)

```

6.1.19) READ_CURRENT

```

p_pointer := 16#11000000; (* RACK base address *)
p_pointer := p_pointer + dw_slot * 16#10000; (* MTC4011 base address *)
p_pointer_ir := p_pointer + 16#200; (* IR current pointer *)
p_pointer_is := p_pointer + 16#202; (* IS current pointer *)
p_pointer_it := p_pointer + 16#204; (* IT current pointer *)
ir_q12 := p_pointer_ir^; (* IR read *)
is_q12 := p_pointer_is^; (* IS read *)
it_q12 := p_pointer_it^; (* IT read *)
r_IR := r_NOMINAL_CURRENT*ir_q12/2896.309376; (* Ireal = Nominal*sqrt(2)*Iq12/2^12 *)
r_IS := r_NOMINAL_CURRENT*is_q12/2896.309376; (* Ireal = Nominal*sqrt(2)*Iq12/2^12 *)
r_IT := r_NOMINAL_CURRENT*it_q12/2896.309376; (* Ireal = Nominal*sqrt(2)*Iq12/2^12 *)

```

6.1.20) READ_SPEED

```

p_pointer := 16#11000000; (* RACK base address *)
p_pointer := p_pointer + dw_slot * 16#10000; (* MTC4011 base address *)
p_pointerl := p_pointer + 16#208;
p_pointerh := p_pointer + 16#20A;
speed_pulses := p_pointerh^;
speed_pulses := SHL(speed_pulses,16);
speed_pulses := speed_pulses + p_pointerl^;
di_speed_pulses := DWORD_TO_DINT(speed_pulses);
r_speed := DINT_TO_REAL(di_speed_pulses)/r_encoder_pulses/2.546479089;

```

6.1.21) READ_PHASE_VOLTAGES

```

p_pointer := 16#11000000; (* RACK base address *)
p_pointer := p_pointer + dw_slot * 16#10000; (* MTC4011 base address *)
p_pointer_aux := p_pointer + 16#200; (* Phase R voltage pointer *)
i_aux := p_pointer_aux^;
r_aux := INT_TO_REAL(i_aux)/9.7979; (* Scale factor *)
r_vr := r_aux;
p_pointer_aux := p_pointer + 16#202; (* Phase S voltage pointer *)
i_aux := p_pointer_aux^;
r_aux := INT_TO_REAL(i_aux)/9.7979; (* Scale factor *)
r_vs := r_aux;
p_pointer_aux := p_pointer + 16#204; (* Phase T voltage pointer *)
i_aux := p_pointer_aux^;
r_aux := INT_TO_REAL(i_aux)/9.7979; (* Scale factor *)
r_vt := r_aux;

```

6.1.22) TA_TB_TC_OUT

```

(* Select codesys modulator *)
dir_base := dw_SLOT;
dir_base := dir_base * 16#10000;
dir_base := dir_base + 16#1100061A;
dir_base^ := 1;
(* Send references *)
dir_base := dw_SLOT;
dir_base := dir_base * 16#10000;
dir_base := dir_base + 16#11000606;
IF dir_base^ = 0 THEN
    dir_base := dw_SLOT;
    dir_base := dir_base * 16#10000;
    dir_base := dir_base + 16#11000600;
    dir_base^ := REAL_TO_INT(r_ta);
    dir_base := dir_base+2;
    dir_base^ := REAL_TO_INT(r_tb);
    dir_base := dir_base+2;
    dir_base^ := REAL_TO_INT(r_tc);
    dir_base := dir_base+2;
    dir_base^ := 16#5555;
END_IF

```


6.1.22) VA_VB_OUT

```
(* Voltage conversion *)
va:=56755.84086 * r_VA /r_VBUS;
vb:=56755.84086 * r_VB /r_VBUS;
dir_base := dw_SLOT;
dir_base := dir_base * 16#10000;
dir_base := dir_base + 16#11000600;
dir_base^:= REAL_TO_INT(va);
dir_base := dir_base+2;
dir_base^:= REAL_TO_INT(vb);
dir_base := dir_base+4;
dir_base^:= 16#5555;
```

6.1.22) OUT_TO_OSCILSCOPE

```
p_pointer := 16#11000000; (* RACK base address *)
p_pointer := p_pointer + dw_slot * 16#10000; (* MTC4011 base address *)
data_pointer := p_pointer + 16#280;
data_pointer^:= REAL_TO_INT(r_IN_E140 *r_FACTOR_E140);
data_pointer := p_pointer + 16#282;
data_pointer^:= REAL_TO_INT(r_IN_E141 *r_FACTOR_E141);
data_pointer := p_pointer + 16#284;
data_pointer^:= REAL_TO_INT(r_IN_E142 *r_FACTOR_E142);
data_pointer := p_pointer + 16#286;
data_pointer^:= REAL_TO_INT(r_IN_E143 *r_FACTOR_E143);
data_pointer := p_pointer + 16#288;
data_pointer^:= REAL_TO_INT(r_IN_E144 *r_FACTOR_E144);
data_pointer := p_pointer + 16#28A;
data_pointer^:= REAL_TO_INT(r_IN_E145 *r_FACTOR_E145);
data_pointer := p_pointer + 16#28C;
data_pointer^:= REAL_TO_INT(r_IN_E146 *r_FACTOR_E146);
data_pointer := p_pointer + 16#28E;
data_pointer^:= REAL_TO_INT(r_IN_E147 *r_FACTOR_E147);
data_pointer := p_pointer + 16#290;
data_pointer^:= REAL_TO_INT(r_IN_E148 *r_FACTOR_E148);
data_pointer := p_pointer + 16#292;
data_pointer^:= REAL_TO_INT(r_IN_E149 *r_FACTOR_E149);
```

6.2) Equipo Empleado

Se detalla a continuación el equipo con el que se ha contado para el proyecto:

6.2.1) Inversor

Se ha empleado el inversor con 6 IGBTs Montelec NFS 200-20 de la Figura 6.1 que posee una corriente máxima de pico de 258 amperios. La tensión máxima de bus es de 800 V, además de estar equipado con un banco de condensadores que aplanan la tensión de alimentación rectificada para generar una tensión continua de mayor calidad. Las fases del motor se encuentran conectadas en estrella a cada rama del inversor, es por eso que la tensión que llega a bornes del motor es la diferencia entre las tensiones en las ramas y se aplica sobre cada fase respecto al neutro de la estrella. Para obtener más información acerca de este inversor consultar la hoja de características en el anexo 6.6)

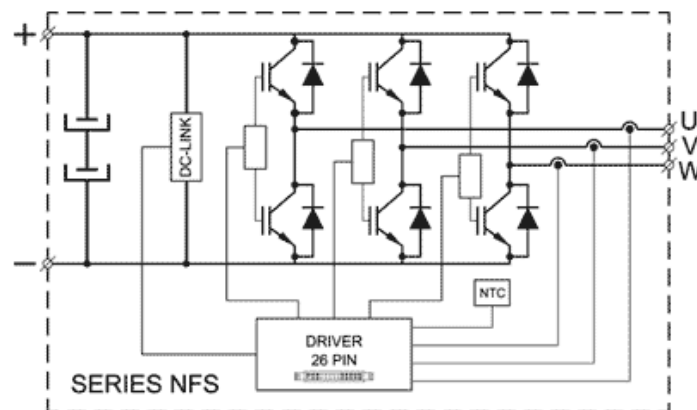


Figura 6.1: Esquema del inversor Montelec NFS 200-20.

6.2.2) PLC MTC4000

El PLC de la Figura 6.2 es el encargado de ejecutar los controles, la librería ha sido programada en CoDeSys para que pueda ser ejecutada por este. Emplea el sistema operativo multitarea en tiempo real FreeRTOS y funciona a 200 MHz, todas las operaciones son realizadas en coma flotante y se comunica mediante un puerto Ethernet y un bus de 60 MHz con tecnología GTLP. Para más información consultar la hoja de características del PLC MTC4000 en el anexo 6.7).



Figura 6.2: PLC MTC4000.

6.2.3) Tarjeta de control de motores y regeneración MTC 4011

Esta tarjeta es empleada para realizar diferentes controles de equipos de potencia y para ello se vale del DSP Texas Instruments TMS320F335. A lo largo de este proyecto se hará trabajar a esta tarjeta como una tarjeta de entradas y salidas (modo 5) y por tanto perderá el control sobre el sistema que será realizado por el PLC MTC4000. Para más información consultar la hoja de características de la tarjeta de control de motores MTC-4011 en el anexo 6.8).

6.2.4) Motor Asíncrono VASCAT MAC QI 132 L

El motor asíncrono VASCAT de la Figura 6.3 se trata de un motor eléctrico asíncrono trifásico de altas prestaciones utilizado como servomotor y equipado con el bobinado 141. Esto le permite desarrollar en régimen nominal **34.4KW de potencia, 1825 rpm de velocidad y 180 Nm de par**. En dicho régimen el motor consume 67 Amperios (RMS) de corriente a 400 Voltios (RMS) 62 Hertzios y un factor de potencia de 0.82. El motor está equipado con un ventilador de refrigeración y tiene un peso de 135 kg. El rotor posee un momento de inercia de 0.077 Kg m². El resto de parámetros técnicos del motor incluyendo los parámetros eléctricos como inductancias y resistencias tanto del rotor como del estator pueden ser consultados en su ficha técnica en el anexo 6.5).

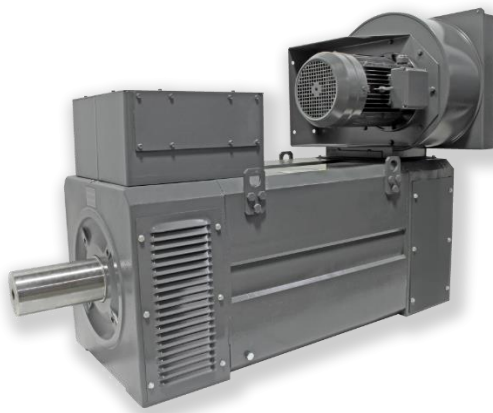


Figura 6.3: Motor asíncrono VASCAT MAC QI.

6.2.5) CoDeSys

CoDeSys es un entorno de programación para controladores automáticos desarrollado y comercializado por la empresa alemana de software 3S-Smart que satisface el estándar industrial IEC 61131-3. Este entorno soporta los seis lenguajes de programación descritos en el IEC 61131-3:

IL (Instruction list): Lenguaje de programación de bajo nivel similar al lenguaje ensamblador.

ST (texto estructurado): Lenguaje de programación similar al lenguaje Pascal.

LD (Ladder Diagram): Lenguaje de programación gráfico basado en contactos de bobinas y relés.

FBD (Function Block Diagram): Lenguaje de programación gráfico basado en bloques de funciones, es un lenguaje conveniente para la programación de lógica booleana.

SFC (Bloques de función secuenciales): Lenguaje de programación gráfico basado en procesos de programación secuencial.

La mayoría de los bloques de la librería están programados en el lenguaje Structured Text a excepción del modelo de flujo, dado que este bloque posee una complejidad mayor y emplea a su vez otros bloques para operar. La programación en Structured Text hace que el las líneas de código de los bloques sean fácilmente comprensibles y permite añadir comentarios para de esta forma facilitar la tarea de mantenimiento del código y comprensión de su funcionamiento.

Las aplicaciones están programadas en el lenguaje LD, de esta forma se pueden insertar los bloques que realizan las diferentes operaciones dentro de estas para visualizar su funcionamiento de forma sencilla. Del mismo modo en este lenguaje también se incluyen los diagramas de contactos que controlan los estados del sistema (marcha, paro, reset...) y del inversor que alimenta al mismo (cargar tensión de bus, resetear alarmas...).

6.2.6) Pasaban Drivers

Este es el software propio de la empresa Pasaban, ha sido desarrollado por Montelec y posee varios modos de trabajo además de un osciloscopio capaz de representar los datos guardados en memoria (que es el que se emplea para representar la evolución de las variables del motor). Este programa es además el encargado de grabar el firmware en la tarjeta de control MTC4011.

A lo largo de este proyecto se ha empleado el modo de trabajo 5 correspondiente a lazo abierto, es decir, el programa configura la tarjeta MTC 4011 como una tarjeta de entradas y salidas y no como un controlador.

6.3) Modelado del inversor

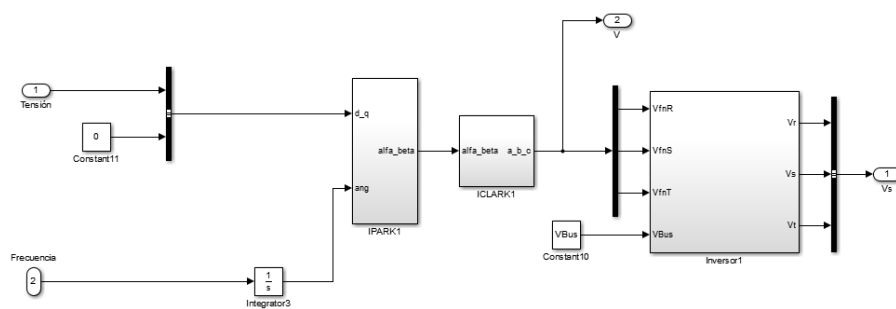


Figura 6.4: Modelo en SIMULINK del inversor 1

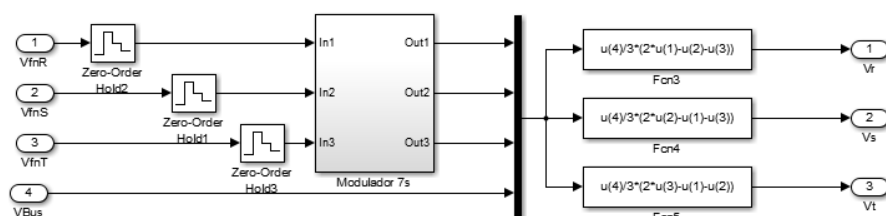


Figura 6.5: Modelo en SIMULINK del inversor 2 (interior del bloque inversor1)

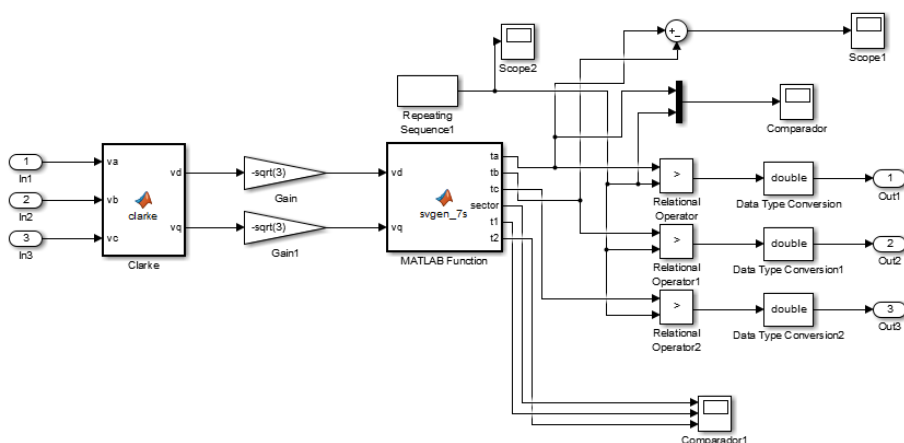


Figura 6.6: Modelo en SIMULINK del inversor 3 (interior del bloque Modulador 7s)

En las Figuras 6.4 6.5 y 6.6 se pueden observar las diferentes partes del modelo en Matlab de inversor que se ha realizado de acuerdo con la documentación [4] y [5].

Nota: La función `svgen_7s` calcula los tiempos de disparo de los IGBTs de la misma forma que la función `mod_7s` de la librería de control. Para consultar las ecuaciones de cálculo de los tiempos de disparo dirigirse a la sección 6.1).

En la Figura 6.7 se representa el modelo completo del motor que incluye tanto la parte mecánica como la eléctrica. La figura 6.8 representa la parte mecánica del modelo, es decir, la aceleración angular del rotor en función de la inercia y el balance de momentos. En la figura 6.9 se representa la parte eléctrica del modelo del motor donde se encuentran implementadas estas ecuaciones en espacio de estados.

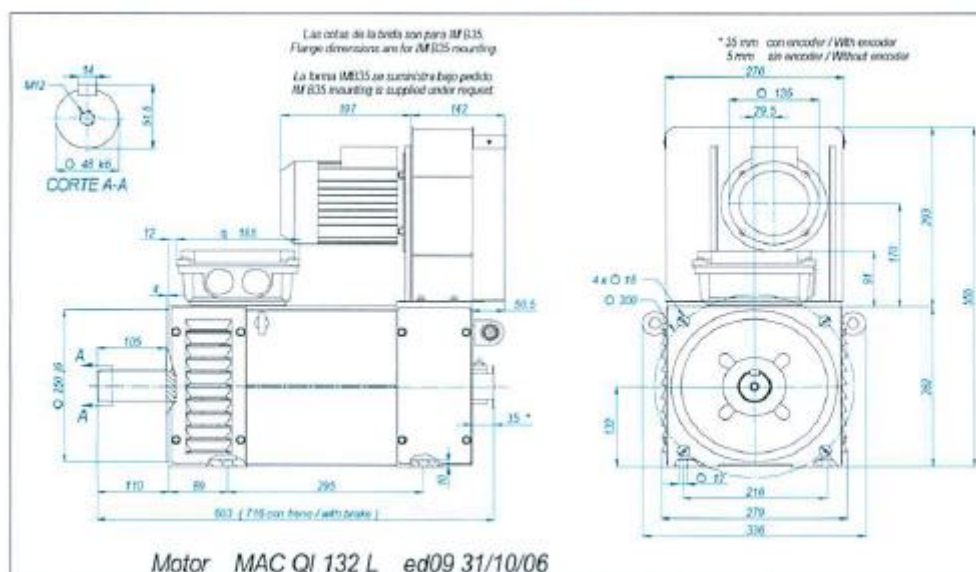
Al igual que el inversor, el motor se ha modelado de acuerdo con la documentación existente en [4] y [5].

6.5) Ficha técnica del motor asíncrono VASCAT MAC QI 132 L

	MOTOR MAC QI 132 L - Hoja de Datos Técnicos - - Technical Datasheet -	Código	DT-QI132L
		Edición	12
		Página	1/1
		Fecha	11/11/2009

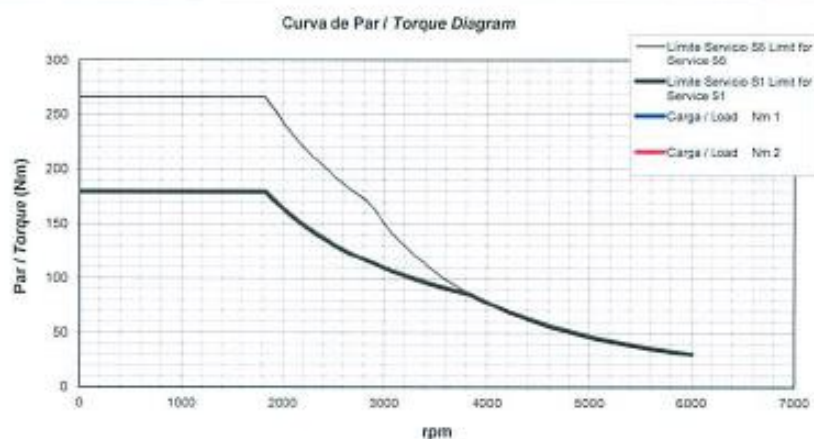
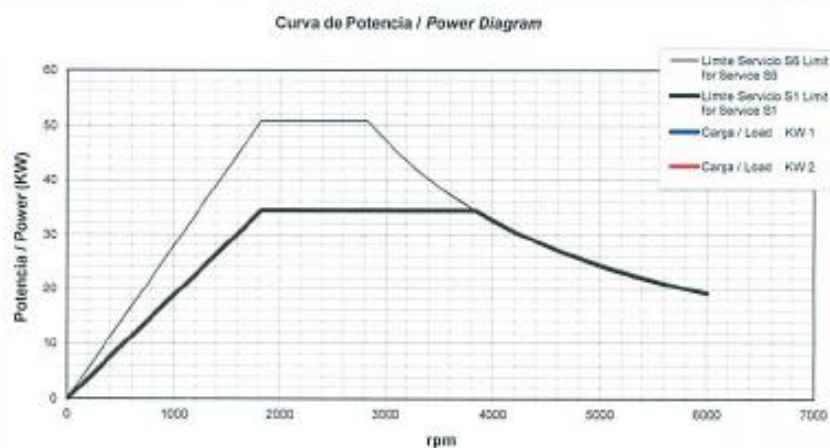
Ventilador / Fan			Rodamientos <i>Bearings</i>		Delantero <i>Drive End</i>		6310 ZZ C3		Trasero <i>Non Drive End</i>		6208 ZZ C3					
Tensión / Voltage	230/400 V	460 V	Momento de inercia <i>Rotor Inertia</i>					J = 0,077 Kg m ²		Peso Motor <i>Motor Weight</i>		135 Kg				
Free. / Frequency	50 Hz	60 Hz	Protección <i>Protection Degree</i>		IP 23		Construcción <i>Mounting</i>		IM B3 / B35		Refrigeración <i>Cooling</i>		IC 06			
Potencia / Power	0,37 KW	0,43 KW	Velocidad / Speed		2800 rpm		3300 rpm									
Caudal / Air flow	900 m ³ /h		Nivel de ruido <i>Noise Level</i>			< 80 dB		Velocidad máxima mecánica <i>Max. Mechanical Speed</i>			6000 rpm					
Presión / Pressure	680 Pa		Equilibrado grado <i>Balancing degree</i>		A		Aislamiento Clase <i>Insulation Class</i>		F		Protección Térmica <i>Thermal Protection</i>		PTC 140 °C			
			Ambiente <i>Ambient</i>		< 40 °C		< 1000 m									
Freno de Bloqueo (opcional) <i>Holding Brake (optional)</i>			150 Nm		24 Vdc		3,5 A		85 W		12 Kg		J = 0.0029 Kg m ²			
Tensión de Base del Motor <i>Motor Base Voltage</i>					Vb		400 V		Tensión Máxima de Salida del Convertidor <i>Inverter max. Output Voltage</i>					Vc	400 V	


	Servicio / Service S1						Servicio / Service S6 - 40%						
Bobinado Winding	Nm	A	KW	rpm	Hz	Velocidad a Pot. Cte Const. Power Speed (rpm)	Nm	A	KW	rpm	Hz	Velocidad a Pot. Cte Const. Power Speed (rpm)	Corriente en vacío No Load Current (A)
111	180	18	7,3	386	14	810	266	24	10,6	386	15	590	8
211	180	23	10,6	564	20	1180	267	32	15,7	564	21	870	10
112	180	29	14,0	744	26	1560	267	41	20,8	744	27	1150	12
121	180	34	16,3	865	30	1810	266	47	24,1	865	31	1330	15
212	180	39	19,7	1044	36	2190	267	55	29,1	1044	37	1610	17
221	180	45	23,1	1224	42	2570	267	64	34,2	1224	43	1890	19
122	180	58	29,8	1584	54	3320	267	81	44,2	1584	55	2450	24
141	180	67	34,4	1825	62	3830	267	93	50,9	1825	63	2830	29
222	180	78	41,1	2184	74	4580	267	110	60,9	2184	75	3380	34
241	180	91	47,4	2515	85	5260	266	127	70,1	2515	86	3900	39
142	180	115	60,9	3235	109	6000	267	162	90,3	3235	110	5010	50

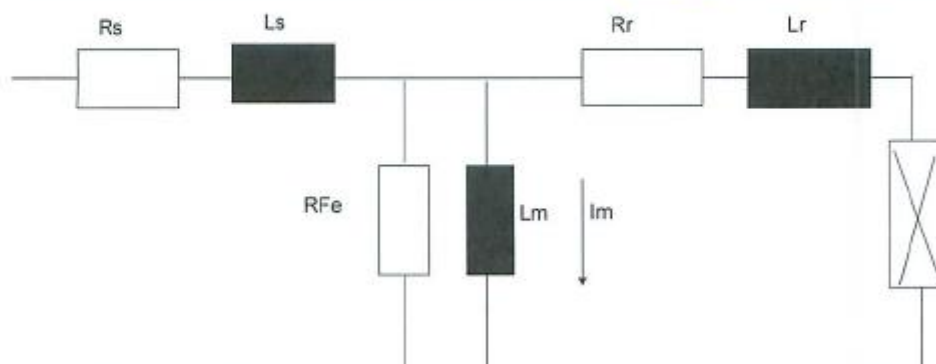


6. Anexos

Bobinado Winding	Servicio / Service S1						Servicio / Service S6 - 40%						Corriente en vacío No Load Current (A)
	Nm	A	KW	rpm	Hz	Velocidad a Pot. Cte Const. Power Speed (rpm)	Nm	A	KW	rpm	Hz	Velocidad a Pot. Cte Const. Power Speed (rpm)	
141	180	67	34,4	1025	62	3830	267	93	50,9	1625	63	2830	29

[illegible][illegible]

		MOTOR MAC QI 132 L										Fecha - Date 11-11-09	
Parâmetros motor / Motor parameters													
PLACA DE CARACTERÍSTICAS / NAMEPLATE DATA													
	QI 132 L Bob. 111	QI 132 L Bob. 211	QI 132 L Bob. 112	QI 132 L Bob. 121	QI 132 L Bob. 212	QI 132 L Bob. 221	QI 132 L Bob. 122	QI 132 L Bob. 141	QI 132 L Bob. 222	QI 132 L Bob. 241	QI 132 L Bob. 142		
Nm	180	180	180	180	180	180	180	180	180	180	180		
rpm	386	564	744	865	1044	1224	1584	1825	2184	2515	3235		
KW	7,3	10,6	14,0	16,3	19,7	23,1	29,8	34,4	41,1	47,4	60,9		
V	400	400	400	400	400	400	400	400	400	400	400		
A	17,5	23,0	28,9	33,8	39,1	45,4	57,5	66,5	78,4	90,7	115,3		
Hz	14,0	20,0	26,0	30,0	36,0	42,0	54,0	62,0	74,0	85,0	109,0		
cos fi	0,83	0,83	0,82	0,82	0,82	0,82	0,82	0,82	0,82	0,82	0,82		
Rend / eff.	0,70	0,77	0,82	0,84	0,86	0,87	0,90	0,90	0,91	0,92	0,93		
J (Kg m2)	0,0798	0,0798	0,0798	0,0798	0,0798	0,0798	0,0798	0,0798	0,0798	0,0798	0,0798		
ESQUEMA EQUIVALENTE POR FASE EN ESTRELLA / PHASE STAR EQUIVALENT SCHEMA													
	QI 132 L Bob. 111	QI 132 L Bob. 211	QI 132 L Bob. 112	QI 132 L Bob. 121	QI 132 L Bob. 212	QI 132 L Bob. 221	QI 132 L Bob. 122	QI 132 L Bob. 141	QI 132 L Bob. 222	QI 132 L Bob. 241	QI 132 L Bob. 142		
Rs (mOhm)	1537,12	830,98	512,37	384,28	278,99	207,74	128,09	96,07	68,25	51,94	32,02		
Rr (mOhm)	1136,80	611,34	378,93	284,20	203,78	152,64	94,73	71,05	50,95	38,21	23,68		
Rfe (Ohm)	653,36	653,35	365,06	310,16	251,94	310,16	153,70	125,98	96,52	125,98	55,95		
Ls (mH)	14,75	7,93	4,92	3,69	2,64	1,58	1,23	0,92	0,66	0,50	0,31		
Lr (mH)	17,88	9,40	5,75	4,30	3,05	2,31	1,42	1,06	0,76	0,57	0,35		
Lm (mH)	297,52	158,85	98,42	74,62	53,33	39,53	24,75	18,64	13,50	9,95	6,18		
Ts (s)	0,20	0,20	0,20	0,20	0,20	0,20	0,20	0,20	0,20	0,20	0,20		
Tr (s)	0,28	0,28	0,27	0,28	0,28	0,27	0,28	0,28	0,28	0,28	0,28		
Kt	346,83	254,34	200,24	173,41	146,84	127,17	100,12	86,71	73,42	63,59	50,06		
Im (A/fase)	7,8	9,7	12,2	15,0	16,7	19,1	24,3	28,6	33,5	39,4	49,8		

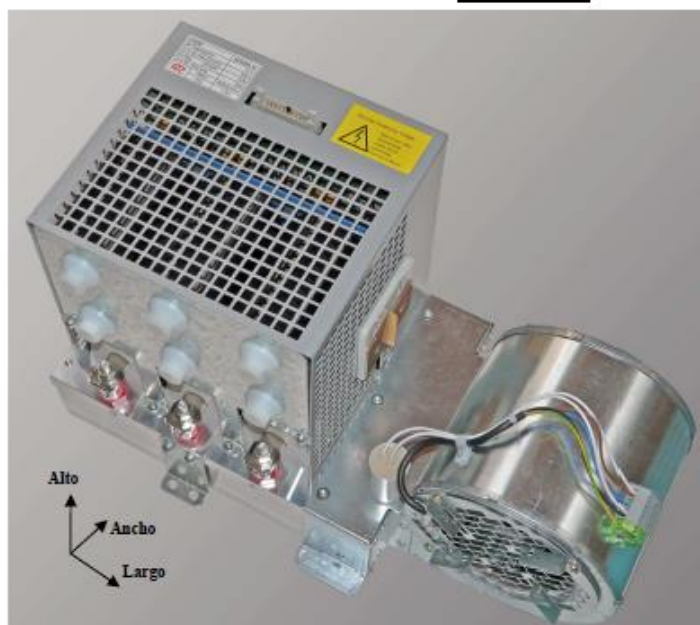


6.6) Ficha técnica del inversor Montelec NFS 200-20



Preliminar:1.0 30/04/2008
NFS-200

Pag:1

NFS-200**Descripción**

Equipo de potencia trifásico de gama media con excelentes prestaciones para el gobierno de cargas.

- $V_{ALIMENTACION}$ 24V_{DC}.
- $V_{BUS(MAX)}$ 800V_{DC}.
- Medida de V_{BUS} .
- Medida de temperatura del disipador.
- Medida de corrientes.
- Conexión con tarjeta de control mediante cable plano 26PIN.
- Banco capacitivo.
- Conexión al Bus por ambos lados.
- Admite varios sentidos de ventilación.
- Varios rangos de medida dentro de un mismo tamaño.

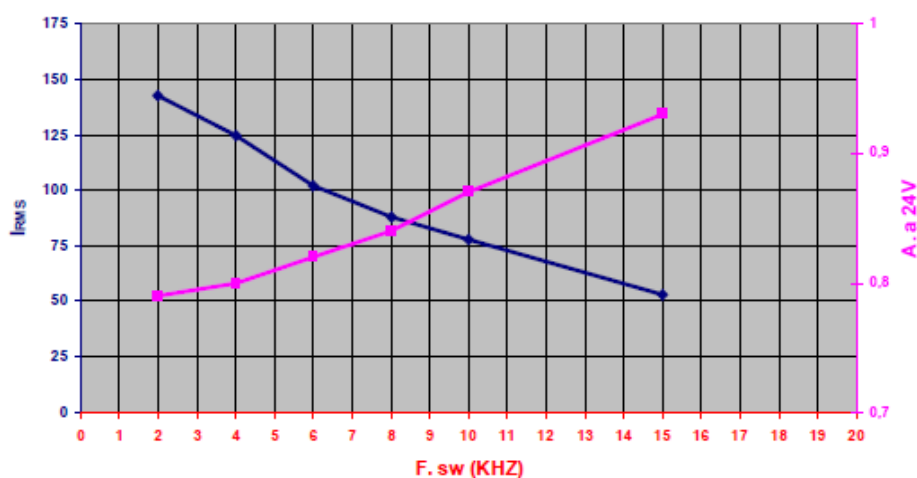
Area considerada	Dimensiones _(mm) (sin ventilador)			Peso _(kg)
	Alto	Ancho	Largo	
Area total necesaria	305	250	250	13,8Kg
Disipador	77	215	250	

Aplicación

Control de motores AC mediante la modulación de tensión continua o regeneración a red.

Respuesta F_{sw}/A . OUT.RMS. & F_{sw}/A . a 24VDC.

($V_{BUS}=560V$, $T_{AMB}=40^{\circ}C$, $F_{out}=50Hz$, Ventilador SKF 16B, Presión_{ATM}=1010mbar)



6.7) Ficha técnica del PLC MTC4000

**PLC DE ALTO RENDIMIENTO MTC-4000**

Pág. 1 de 2

DESCRIPCIÓN

La **MTC-4000** es la CPU de la familia de PLCs modulares de mayor rendimiento. Satisface la gran mayoría de necesidades pudiendo ser utilizada tanto para sencillos automatismos como para las aplicaciones de control más exigentes. Su diseño permite ampliaciones en cualquier momento de forma flexible.

El entorno de desarrollo empleado es *CoDeSys* (www.3s-software.com). **Montelec S.L.** forma parte de la red mundial de asociados al sistema de *CoDeSys*. Hoy en día es la herramienta de desarrollo basada en IEC 61131-3 más extendida en Europa. Entre muchas de sus ventajas se encuentra la posibilidad de combinar hasta cinco lenguajes de programación.

**Resumen de prestaciones:**

- Core de 32 bits y coma flotante.
- Frecuencia de trabajo de 200MHz.
- Sistema operativo multitarea en tiempo real.
- Sistema de ficheros virtual que simplifica tareas como:
 - Almacenar automáticamente el programa y código fuente.
 - Realizar copias de seguridad de la CPU.
 - Actualizar de forma sencilla el firmware de la CPU
 - Conectividad remota y fácil manejo a través de Internet.
- Detección de fallos y manejo de alarmas con historial.
- Direccionamiento IP mediante switches o software.

APLICACIÓN

El PLC (MTC-4000) está especialmente diseñado para disminuir los ciclos de trabajo y aumentar la calidad de cualquier proceso industrial. Su diseño es modular y pensado para una periferia de E/S descentralizada.

La configuración de todos los periféricos que componen el sistema se realiza de una manera fácil y segura a través de un bus de comunicación.

El PLC (MTC-4000) esta pensado especialmente para comandar bienes de equipo, bien sean (S.A.I.S., control de Motores, equipos Eólicos, equipos Solares, maquinaria para la manipulación y fabricación de piezas,...).



PLC DE ALTO RENDIMIENTO MTC-4000

Pág. 2 de 2

ESPECIFICACIONES TÉCNICAS

CPU	
<ul style="list-style-type: none"> 32 bits y operaciones en coma flotante 200MHz Tarjeta formato Europa Firmware actualizable 	
<ul style="list-style-type: none"> Interruptor de estado de operación: RUN, STOP, RESET Un puerto RS422 de propósito general Dos puertos Ethernet con PC 100/10 RJ45. Protocolo TCP/IP Indicación de estados operativos de la CPU mediante diodos led Asignación de dirección IP vía switch interno o mediante software 	
Tiempos de procesamiento	
<ul style="list-style-type: none"> Operaciones con bits 25 ns Operaciones con words 30 ns Aritmética con enteros 30 ns Aritmética en coma flotante 500 ns 	
Memoria	
<ul style="list-style-type: none"> Flash interna 14MB RAM DDR 4MB Memoria Magnetoresistiva NVRAM retentiva 512KB 	
Timers / Contadores	
<ul style="list-style-type: none"> Ilimitados, cada timer o contador queda abstraído por programa Temporización mínima 1ms Temporización máxima 49d17h2m47s295ms Retención programable Reloj de Tiempo Real (RTC) 	
Interrupciones	
<ul style="list-style-type: none"> 3 Interrupciones por eventos / 2 Interrupciones de BUS Cíclicas: período definido en milisegundos Evento de software: cualquier variable global de programa Relojes de precisión: 500ms, 100ms, 50ms 10ms, 5ms, 2ms, 1ms 750µs, 500µs, 125µs Evento externo del BUS 	

Periferia E/S	
<ul style="list-style-type: none"> Canales Digitales: Sin Límite preestablecido Canales Analógicos: Sin Límite preestablecido Direccionamiento del BUS de hasta 12 tarjetas (16KBx12) 	
Racks	
<ul style="list-style-type: none"> Buss mediano ultrarrápido combinado (2+5) MTC-4009 Buss largo ultrarrápido combinado (2+12) MTC-4010 	
Otros periféricos	
<ul style="list-style-type: none"> Controladores de motores MTC-4011 Tarjeta de interrupciones y contadores rápidos MTC-4002 Tarjeta de comunicaciones: <ul style="list-style-type: none"> 2xCAN; 1xRS485; 1xProfibus MTC-4012 Fuente de alimentación 5V/60W MTC-4004 Fuente de alimentación 5V/100W MTC-4013 	

6.8) Ficha técnica de la tarjeta de control de motores MTC4011



CONTROLADORA DE ETAPAS DE POTENCIA MTC-4011

Pág. 1 de 2

DESCRIPCIÓN



La **MTC-4011** es una tarjeta diseñada para controlar etapas de potencia de hasta seis IGBTs y tres fases. Su principal función consiste en el control de motores. También dispone de un control de regeneración a la red.

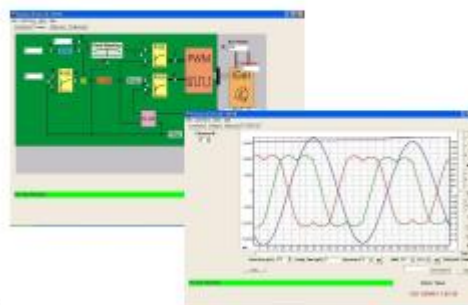
Este controlador de etapas de potencia tiene un alto grado de flexibilidad y excelente respuesta dinámica. Puede ser configurado para adecuarse con precisión a las necesidades de cada aplicación industrial.

La unidad de regulación y la de potencia están separadas, lo que permite manejar con una sola tarjeta MTC-4011 cualquier etapa de potencia serie NFS o superior. Para más información véase la nota de aplicación N° 201.

CARACTERÍSTICAS

La inteligencia del accionamiento y sus funciones de control se encuentran en la tarjeta **MTC-4011**. Esta es capaz de controlar de forma precisa motores de corriente continua, asíncronos y motores brushless (*torque*) en los siguientes modos:

- Regulación vectorial
- Control V/f
- Lazo abierto
- Controles de posición, velocidad y par
- Regeneración a la red.



A continuación se destacan algunas de sus ventajas:

- Configuración, chequeo y monitorización de parámetros rápida e intuitiva.
- Configuración de forma gráfica de los lazos de control y sus parámetros.
- Osciloscopio con escalado, triggering y FFT para análisis de la regulación.
- Historial con análisis gráfico de cualquier salto de alarma haciendo un solo clic!
- Firmware actualizable de forma sencilla incluso desde Internet.
- Las señales de encoder se pueden manejar mediante cableado o bus interno. Esto aporta flexibilidad en controles complejos con motores maestro / esclavo.
- Los bloques de control para el entorno de programación **CoDeSys** (www.3s-software.com), permiten modificar de forma dinámica todas las consignas y comandos relativos a los diferentes modos de control.

ESPECIFICACIONES TÉCNICAS

Formato	
<ul style="list-style-type: none"> Tarjeta formato Europa 	
Alimentación	
<ul style="list-style-type: none"> Control de la tarjeta: Directo del Rack (<i>consumo: 5V/1A</i>) Encoder y control del módulo de potencia: A través del conector frontal 	
Comunicaciones	
<ul style="list-style-type: none"> Etapas de potencia: Cable plano apantallado de 26 hilos PLC MTC-4000: Bus interno Software de configuración: Routing a través del puerto ethernet 	
Encoder	
<ul style="list-style-type: none"> La alimentación proviene del conector frontal de 24V (salida a 15V). Máxima frecuencia de trabajo: 100KHz 	
MTC-4011-C	<ul style="list-style-type: none"> Señal digital cuadrada: (Incremental Push-pull) Encoder Absoluto: (SSI)
MTC-4011-S	<ul style="list-style-type: none"> Señal senoidal: (SinCos, 1Vss) Encoder Absoluto: (SSI)
Modos de control	
<ul style="list-style-type: none"> Control Vectorial Control V/f Control en lazo abierto 	
Lazos de control	
<ul style="list-style-type: none"> Velocidad Par Posición 	
Tipos de controles	
<ul style="list-style-type: none"> Motores de Corriente Continua Motores Asíncronos Motores Torque (<i>brushless</i>) 	
<ul style="list-style-type: none"> Regeneración a la red 	
Frecuencia de conmutación	
<ul style="list-style-type: none"> Frecuencia de la etapa de potencia configurable: 2KHz-20KHz 	
Consignas y estados	
<ul style="list-style-type: none"> Manejables fácilmente desde bloques de función de CoDeSys Accesibles a través del bus del PLC 	
Alarmas	
<ul style="list-style-type: none"> Visualizador alfanumérico (matriz de leds) en el frontal Historial de diez alarmas con registro gráfico 	
Entradas analógicas	
<ul style="list-style-type: none"> Procedentes de la etapa de potencia acoplada vía cable plano 3 ramas de corriente $\pm 10V$ 1 temperatura 0-10V 1 DC-link 0-10V 	

