



**IMPLEMENTACIÓN DE UN ALGORITMO DE CONTROL MEDIANTE DSP PARA EL
MOTOR DE INDUCCIÓN**

HENRY ALFONSO SEPULVEDA PACAGUI

**UNIVERSIDAD DE PAMPLONA
PROGRAMA DE INGENIERÍA ELÉCTRICA
FACULTAD DE INGENIERÍAS Y ARQUITECTURA
PAMPLONA
2019**

**IMPLEMENTACIÓN DE UN ALGORITMO DE CONTROL MEDIANTE DSP PARA EL
MOTOR DE INDUCCIÓN**

HENRY ALFONSO SEPULVEDA PACAGUI

Tesis o trabajo de investigación presentado como requisito parcial para optar al título de:
INGENIERO ELÉCTRICO

Director

EDISON CAICEDO PEÑARANDA

M.Sc. Ingeniero Eléctrico

Codirector (a):

LUIS D. PABÓN FERNÁNDEZ

M.Sc. Ingeniero Eléctrico

Línea de Investigación:

Calidad de la energía

Grupo de Investigación:

Sistemas Energéticos

**UNIVERSIDAD DE PAMPLONA
PROGRAMA DE INGENIERÍA ELÉCTRICA
FACULTAD DE INGENIERÍAS Y ARQUITECTURA
PAMPLONA
2019**



DEDICATORIA

Dedico este logro a las personas que me han brindado su confianza, apoyo incondicional y motivación durante todo el proceso de elaboración de mi proyecto de grado.

Este trabajo está dedicado a mi familia, en especial a mi madre BERTHA DILIA PACAGUI MALDONADO y mis hermanos porque ellos han sido mi apoyo incondicional y me han proporcionado todo y cada cosa que he necesitado.

A mi novia quien me apoyo y motivó en los momentos más turbulentos en la realización de mi proyecto de grado.

A mi director de tesis por haberme confiado este proyecto, por su paciencia ante mi inconsistencia, por su valiosa dirección y apoyo para culminar este trabajo.

A todas las personas que de una u otra manera estuvieron a mi lado, que me enseñaron y me dieron ánimos.

DQS is member of:





AGRADECIMIENTOS

En primer lugar, quiero agradecer a Dios por haberme guiado durante este proceso de arduo esfuerzo y por colocarme personas que han sido apoyo incondicional durante este tiempo de una etapa más de vida.

A mi madre que a lo largo de la vida me ha apoyado y motivado, creyendo en mi todo momento.

A cada uno de los docentes que hicieron parte de mi formación, en especial a mi director de tesis EDISON CAICEDO PEÑARANDA y codirector LUIS DAVID PABÓN.

A mis compañeros porque de cada uno aprendí cosas nuevas, mil gracias por todo su apoyo y enseñanzas.

Por último, a la universidad de Pamplona por permitirme hacer parte de esta gran familia, brindándome la oportunidad de formarme como profesional.

DQS is member of:





RESUMEN

Este libro presenta el desarrollo un algoritmo para el control de velocidad para el motor de inducción de rotor bobinado mediante el microcontrolador con núcleo de DSP TMS320F28069M, el cual es el encargado de realizar el tratamiento digital de las señales provenientes de los circuitos de medición acoplados para realizar el cierre del lazo de control propuesto, las señales son filtradas y normalizadas por un circuito de diseñado por Autor, donde la magnitud eléctrica adquirida será transformada a su valor digital para ser manipulada por la lógica de programación elaborada con el fin de cumplir el objetivo propuesto. Los algoritmos se efectuaron en base a los límites de operación de la técnica de control empleada y las características del DSP, con los cuales se va a estimar la velocidad del motor en un punto de operación especificado la cual se compara son la velocidad de set point especificada en el controlador PI, el cual realiza el cambio de velocidad dependiendo de sus constantes proporcional e integral con ayuda de una salida del DSP PWM_DAC la cual controla el elemento final de control. El convertidor de frecuencia adoptado se encuentra en modo remoto recibe las señales de control provenientes del pin PWM_DAC del dispositivo realizando así el cambio de velocidad del motor. Proporcionando la fuente de alimentación al motor de inducción que intrínsecamente lleva al control planteado.

Por último la respuesta del sistema se supervisa con una interfaz la cual ayudo a observar el comportamiento del prototipo planteado.

Palabras clave: Convertidor de frecuencia, TMS320F28069M, motor de inducción, control, sistemas embebidos.

DQS is member of:





ABSTRACT

This book presents the development of an algorithm for speed control for the winding rotor induction motor by means of the microcontroller with DSP core TMS320F28069M, which is responsible for performing the digital treatment of the signals coming from the measurement circuits coupled to perform the closing of the proposed control loop, the signals are filtered and normalized by a circuit designed by Author, where the acquired electrical magnitude will be transformed to its digital value to be manipulated by the programming logic developed in order to meet the objective proposed. The algorithms were made based on the operating limits of the control technique used and the characteristics of the DSP, with which the motor speed will be estimated at a specified operating point which is compared to the set point speed specified in the PI controller, which performs the speed change depending on its proportional and integral constants with the help of an output of the PWM_DAC DSP which controls the final control element. The adopted frequency converter is in remote mode and receives the control signals from the PWM_DAC pin of the device, thus changing the motor speed. Providing the power supply to the induction motor that intrinsically leads to the raised control.

Finally, the response of the system is monitored with an interface which helped to observe the behavior of the raised prototype.

Keywords: Keywords: Frequency converter, TMS320F28069M, induction motor, control, embedded systems.

DQS is member of:



*Formando líderes para la construcción de un
nuevo país en paz*

Tabla De Contenido

INTRODUCCIÓN	12
JUSTIFICACIÓN.....	13
DELIMITACIONES.....	15
1. GENERALIDADES DEL DISPOSITIVO TMS320F28069	17
1.1 C2000™ Piccolo™ 32-bit MCU	17
1.2 Nomenclatura TMS320F28069	21
1.3 Placa de desarrollo	26
1.4 Control Sensorless.....	32
2. DISEÑO DE LOS ALGORITMOS PRINCIPALES DE CONTROL DEL SISTEMA Y DESCRIPCION DEL ELEMENTO FINAL DE CONTROL A ADOPTAR.	43
2.1 Variador de frecuencia	43
2.2 Descripción del algoritmo principal del sistema de control a lazo abierto...48	
3. IMPLEMENTACION DE LA TECNICA DE CONTROL	52
3.1 Adquisición y normalización de las señales de voltaje y corriente	53
3.2 Transformada de Clark implementación.....	62
3.3 Implementación de estimador de flujo	62
3.4 Implementación de estimador de velocidad.....	68
4. RESULTADOS DE FUNCIONAMIENTO DEL SISTEMA DE CONTROL	74
4.1 Pruebas de funcionamientos y validación.....	74
4.2 Resultados obtenidos de las etapas antes propuestas.....	75
4.3 Pruebas a motor de inducción de rotor bobinado	81
4.4 Validación de la estimación del flujo y velocidad	83
4.5 Prueba del sistema a lazo abierto	86
4.6 Funcionamiento de todo el sistema de control a lazo cerrado	90
CONCLUSIONES	95
REFERENCIAS BIBLIOGRÁFICAS	96
ANEXOS.....	100

Tabla De Ilustraciones

Ilustración 1. 1 Microcontrolador Piccolo TM Fuente: [8].....	17
Ilustración 1. 2 Aplicación de los C2000 Fuente: [9]	18
Ilustración 1. 3 Core C28x Fuente: [9].	19
Ilustración 1. 4 Aplicación para control en tiempo real Fuente: [9].....	19
Ilustración 1. 5 Aceleradores de leyes de control CLA. Fuente: [9].....	20
Ilustración 1. 6 Microcontrolador TMS320F28069M	21
Ilustración 1. 7 Diagrama de bloques funcionales TMS320F28069M. Fuente: [12] .	24
Ilustración 1. 8 Parámetros Y Características del TMS320f28069M	25
Ilustración 1. 9 Vista general LAUNCHXL-F28069M. Fuente: [12].....	27
Ilustración 1. 10 Code composer studio. Fuente: [13].....	28
Ilustración 1. 11 Orientación del interruptor de arranque. Fuente: [12]	29
Ilustración 1. 12 Esquema de control Sensorless planteado .Fuente: autor.....	33
Ilustración 1. 13 Ley de control escalar. Fuente: [16].....	34
Ilustración 1. 14 Transformada de Clarke. Fuente: [16]	35
Ilustración 1. 15 Circuito equivalente de motor de inducción. (a) En el marco de referencia d. (b) En q- marco de referencia .Fuente:[17]	36
Ilustración 1. 16 Diagrama en bloques de estimador de flujo. Fuente:[16].....	37
Ilustración 2. 1 Diagrama de control a lazo abierto. Fuente: autor	43
Ilustración 2. 2 Ciclo de trabajo PWM. Fuente[.].....	44
Ilustración 2. 3 Circuito interno variador de frecuencia	45
Ilustración 2. 4 Variador de frecuencia. Fuente: autor	45
Ilustración 2. 5 LM358	47
Ilustración 2. 6 Simulación del circuito amplificador. Fuente: Autor simulink.....	47
Ilustración 2. 7 Diagrama de bloques del funcionamiento a lazo abierto.....	48
Ilustración 2. 8 Modulo e PWM.....	49
Ilustración 2. 9 Configuración salida PWM_DAC	50
Ilustración 2. 10 PWM_DAC	51
Ilustración 2. 11 Angulo de flujo. Fuente: [16].....	70
Ilustración 3. 1 Diagrama general del sistema de control. Fuente: autor.....	52
Ilustración 3. 2 Efecto hall .Fuente:	53
Ilustración 3. 3 Sensor de corriente. Fuente: autor	54
Ilustración 3. 4 Banco de transformadores Fuente: autor	54
Ilustración 3. 5 Amplificador operacional sumador no inversor	55
Ilustración 3. 6 Simulación acondicionamiento de voltaje	55
Ilustración 3. 7 Diagrama de bloques ADC. Fuente: [21]	57
Ilustración 3. 8 Diagrama SOC. Fuente: [21]	58
Ilustración 3. 9 Algoritmo del ADC. Fuente: [21]	59
Ilustración 3. 10 Diagrama en bloques filtro fir. Fuente: [25]	60
Ilustración 3. 11 Cálculo de coeficiente. Fuente: Autor matlab	61



Ilustración 3. 12 Exportación de coeficientes. Fuente: Autor matlab	61
Ilustración 3. 13 Bloque de filtro	62
Ilustración 3. 14 Bloque de estimador de flujo. Fuente:[16]	67
Ilustración 3. 15 Bloque de estimador de velocidad. Fuente:[16].....	71
Ilustración 3. 16 Controlador PI discreto. Fuente: [29]	72
Ilustración 3. 17 Bloque del controlador.....	73
Ilustración 4. 1 Fluke 123.	74
Ilustración 4. 2 Fluke 434-II Fuente: [30].	75
Ilustración 4. 3 Señales de corriente medidas. Fuente: autor	76
Ilustración 4. 4 Señal de corriente con ruido.....	77
Ilustración 4. 5 Señales filtradas de corriente	77
Ilustración 4. 6 Corriente de la fase IA. Fuente: Autor	78
Ilustración 4. 7 Corrientes trifásicas normalizadas. Fuente: Autor	78
Ilustración 4. 8 Señales de voltaje PWM de entrada motor de inducción. Fuente: Autor	79
Ilustración 4. 9 Señal de voltaje fase VA. Fuente: Autor	79
Ilustración 4. 10 Señal de voltaje fase VA. Fuente: Autor	80
Ilustración 4. 11 Señales de voltaje	80
Ilustración 4. 12 señales trifasicas de Voltaje	81
Ilustración 4. 13 Circuito equivalente del motor de induccion	81
Ilustración 4. 14 Prueba en vacío	82
Ilustración 4. 15 Rotor Bloqueado	82
Ilustración 4. 16 Transformada de clark de Corrientes. Fuente: Autor	84
Ilustración 4. 17 Transformada de Clark de voltajes. Fuente: Autor.....	84
Ilustración 4. 18 Flujo del rotor estimado ejes d-q . Fuente: Autor	85
Ilustración 4. 19 Esquema de control a lazo abierto. Fuente: Autor	86
Ilustración 4. 20 Respuesta del sistema ante cambios de velocidad. Fuente: Autor	87
Ilustración 4. 21 Respuesta de la planta ante un escalon.....	88
Ilustración 4. 22 Datos cargados a Ident Fuente: Autor Ident-Matlab	88
Ilustración 4. 23 Modelo obtenido. Fuente: Autor	89
Ilustración 4. 24 Simulación de lazo de control. Fuente: Autor	89
Ilustración 4. 25 Grafica del controlador Fuente: Autor	90
Ilustración 4. 26 Sistema a lazo cerrado.....	90
Ilustración 4. 27 Montaje final. Fuente: Autor.....	91
Ilustración 4. 28 Cambio de velocidad. Fuente: Autor.....	92
Ilustración 4. 29 Cambio de velocidad por debajo de la nominal. Fuente: Autor	93
Ilustración 4. 30 Medidas de velocidad por debajo de la nominal. Fuente: Autor	93
Ilustración 4. 31 cambio velociada por encima nominal. Fuente: Autor.....	94
Ilustración 4. 32 Velocidad medida a 2150Rpms. Fuente: Autor.....	94



Tabla De Tablas

Tabla 1. 1 Nomenclatura TMS320F28069M fuente: [1]	22
Tabla 1. 2 Parámetros Y Características del TMS320f28069M	22
Tabla 1. 3 Descripción del jumper JPx	28
Tabla 1. 4 Serial Connectivity fuente	29
Tabla 1. 5 Configuración del modo de arranque S1	30
Tabla 1. 6 J1 Pin y J3 Pin.....	30
Tabla 1. 7 J4 Pin y J2 Pin.....	31
Tabla 1. 8 J5 Pin y J7 Pin.....	31
Tabla 1. 9 J8 Pin y J6 Pin.....	32

DQS is member of:



INTRODUCCIÓN

Los dispositivos procesadores de señal piccolo™ C2000, DSP surgieron cuando la tecnología permitió su fabricación y las características de las aplicaciones lo necesitaron. La telefónica móvil, la electro medicina, la robótica, las comunicaciones, la reproducción y el procesamiento del sonido y la imagen, el internet, el control de motores, la instrumentación, son algunas de las áreas típicas de los DSP Cuando los microprocesadores y microcontroladores de 8, 16 y 32 bits no fueron capaces de resolver eficientemente las tareas que el procesamiento digital de señales exigía se reforzaron sus arquitecturas, se amplió el repertorio de instrucciones y se le arropó con numerosos periféricos complementarios para dar lugar a los MCU tipo DSP's.

Los DSP son microcontroladores cuyo diseño ha sido enfocado para soportar las especificaciones del tratamiento de señales, que, por su implicación en los campos tecnológicos más demandados recientemente, supone disponer de un dispositivo programable con los recursos físicos y lógicos precisos para la exigencia de dicha área.

En el capítulo 1 se describe a nivel general las características de la tarjeta de desarrollo LAUNCHXL-F28069M adoptada para la ejecución de los objetivos a nivel de algoritmos planteados, se describe completamente las características del control a implementar donde se presenta la forma de estimación de velocidad a partir de unas mediciones de voltaje y corriente de entrada a la maquina se realiza una descripción detallada de la matemática que este control implica en este proyecto.

En el capítulo 2 se explica la forma de cómo se adoptó el variador de frecuencia como un inversor el cual solo se configura como elemento final de control de acuerdo a una generación de modulación por ancho de pulso la cual va ser la entrada del ADC.

En el capítulo 3 se describe la mayoría de componentes a usarse para la implementación de la técnica de control, se realiza una descripción de la forma como se adquieren las señales de voltaje y corriente, se realiza a la descripción del cómo se implementa el bloque de estimación de flujo y velocidad a nivel de algoritmos junto el controlador de velocidad.

El capítulo 4 se describe a detalle la metodología con la cual se realizaron las pruebas y validaciones de todas las etapas planteadas en el proyecto.

JUSTIFICACIÓN

Los motores de inducción trifásicos son ampliamente utilizados en la industria[1], y generalmente estos necesitan de condiciones de velocidad especificadas por la aplicación, para lograr este objetivo se requiere de un elemento encargado de modificar las variables eléctricas que se reflejen en la modificación de la velocidad de operación [1]. Los elementos encargados de modificar las magnitudes eléctricas han evolucionado gracias a las investigaciones realizadas en las últimas décadas, motivadas por los avances tecnológicos en los semiconductores, que han provocado el desarrollo de nuevas topologías de convertidores y métodos de control los cuales no son elementos industriales actuales debido a que se encuentran en etapa de desarrollo[2].

El cambio en los elementos finales de control, implican ventajas en términos de calidad de la energía, eficiencia energética, conservación de los equipos, causando que el control de la máquina de inducción desarrolle metodologías complejas de perfeccionamiento, por lo cual la reducción del contenido armónico es un tema de investigación en los sistemas de control del motor de inducción[3].

Otra perspectiva del problema es el aumento del consumo energético y el agotamiento de combustibles fósiles que causan el aumento del valor de la energía eléctrica, en este campo existen alternativas de utilización de fuentes de generación no convencionales como los sistemas de generación fotovoltaicos, pero para aplicaciones de motores los sistemas necesitan de un sistema de arranque por las limitaciones di/dt , un inversor, un variador de frecuencia, y opcionalmente equipos que mejoren la calidad de la energía, los cuales se pueden compactar en un sistema de control de velocidad controlado por sistemas embebidos, adoptando un variador de frecuencia en modo remoto actuando como un inversor para el control de velocidad aplicada [3].

Por ello esta investigación busca realizar un control de velocidad adoptando un variador de frecuencia en modo remoto de tal forma que actué como elemento final de control donde se aplicará la técnica de control Sensorless incorporada un sistema embebido[4]. Al utilizar técnicas de control implementadas en un TMS320F28069M se logran obtener grandes ventajas en cuanto al procesamiento de datos de entrada dado que cada dato debe ser multiplicado, sumado y además de eso transformado de acuerdo a fórmulas complejas, esto se logra ya que están diseñados para tareas de altas prestaciones repetitivas y numéricamente intensas, las grandes velocidades que nos brindan estos dispositivos permiten que se realicen de forma muy eficiente



el tratamiento, además que nos permiten tener control de muchas variables de nuestro sistema a controlar limitado únicamente por nuestro dispositivo a usar[5], el bajo costo y grades prestaciones de estos dispositivos han aumentado su uso en los últimos años por las prestaciones antes descritas[6].

DQS is member of:



Formando líderes para la construcción de un nuevo país en paz

DELIMITACIONES

Objetivo General

Implementar un algoritmo de control mediante DSP para el motor de inducción trifásico.

Objetivos Específicos

- Determinar la metodología del control y las características del DSP a emplear.
- Diseñar los algoritmos en el DSP para la generación de modulaciones adoptando el inversor trifásico de acuerdo a una metodología de control.
- Implementar la técnica de control para el motor de inducción en el DSP junto con el acondicionamiento de las señales.
- Validar el funcionamiento del prototipo desarrollado, visualizándolo en una interfaz para el monitoreo on-line.

ACOTACIONES

- Se adoptará un inversor trifásico presente en los laboratorios o grupos de investigación del programa.
- Los sensores que registraran las variables a medir, serán sensores que se encuentren en los laboratorios y grupos de investigación.
- El algoritmo de control que se implementará se definirá de acuerdo a las características y límites de operación del motor de inducción en conjunto con el DSP.
- Los ángulos de disparo para las modulaciones serán encontrados de acuerdo a las técnicas de modulación ya previamente desarrolladas e implementadas en trabajos anteriores de los grupos de investigación en conjunto con el algoritmo de control.
- La validación del funcionamiento del control de velocidad se realizará con instrumentos de medición presentes en la universidad y con herramientas computacionales proporcionadas por el fabricante del DSP elegido.

1. GENERALIDADES DEL DISPOSITIVO TMS320F28069

La tarjeta dispuesta para la ejecución de este proyecto es el elemento fundamental en el sistema de control, ya que es el encargado de las tareas más importantes para el control del sistema. Dichas tareas son la adquisición de datos para su posterior procesamiento control y la generación de señal de control del convertidor de frecuencia adoptado poder llevar a cabo un algoritmo de control de velocidad implementado por el usuario para la máquina trifásica [7]. A continuación, se explica cómo el microcontrolador tipo DSP Piccolo™ ilustración 1.1 de bajo costo de 32 bits de Texas Instruments ofrece una solución para controlar la velocidad de un motor. El sistema general está pensado para mejorar la eficiencia y ahorrar tiempo de desarrollo[8].

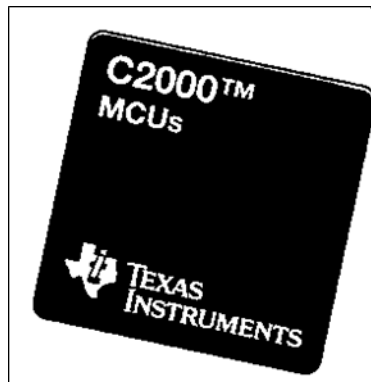


Ilustración 1. 1 Microcontrolador Piccolo™ Fuente: [8]

1.1 C2000™ Piccolo™ 32-bit MCU

La familia C2000 de microcontroladores de alto rendimiento diseñados específicamente para controlar la electrónica de potencia y proporcionar procesamiento avanzado de señales digitales en aplicaciones industriales y automotrices. Una gama de microcontroladores y dispositivos procesadores de señales líderes en control analógico y digital, los MCU C2000 han evolucionado para proporcionar una mejor precisión, procesamiento potente y actuación Premium para permitir crear los sistemas de control de energía más eficientes del mundo, con diferentes aplicaciones las cuales se observan en la ilustración 1.2 [9].

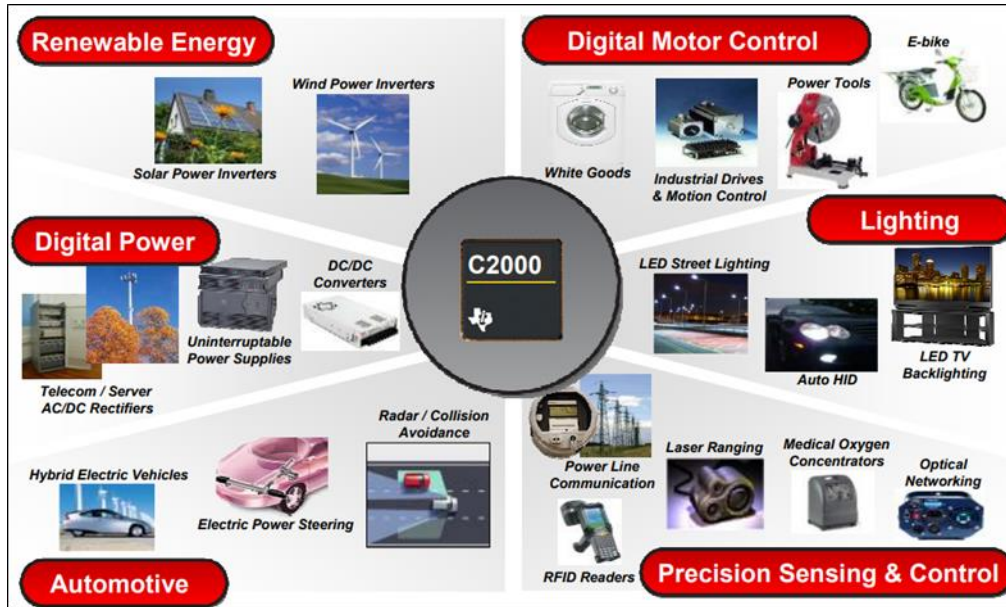


Ilustración 1. 2 Aplicación de los C2000 Fuente: [9]

En esencia, las MCU C2000 se basan en el núcleo DSP TMS320C28x de 32 bits, que presenta multiplicaciones de hardware de 32×32 bits de un solo ciclo y ejecución de instrucción atómica de ciclo único. El núcleo C28x también incluye aceleradores de hardware como la unidad matemática compleja Viterbi para algoritmos de comunicación de líneas de alimentación y la unidad de matemática trigonométrica para acelerar funciones trigonométricas comunes en muchos algoritmos de control de motores. El coprocesador en tiempo real, también conocido como CLA, proporciona una CPU independiente capaz de manejar tareas independientemente del núcleo C28x principal. Aumente el ancho de banda del núcleo C28x descargando tareas intensivas en matemática al CLA, en la ilustración 1.3 se puede apreciar en la de organización del núcleo de C28x[9].

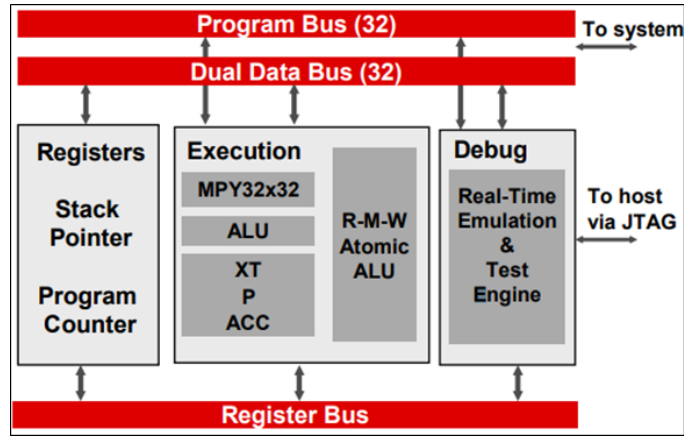


Ilustración 1. 3 Core C28x Fuente: [9].

Control en tiempo real C2000

Los microcontroladores de control en tiempo real C2000 ilustración 1.4, utilizan un núcleo propietario de 32 bits, la CPU C28x, que ofrece operaciones de un solo ciclo y hasta 300 MIPS junto con un bus de periféricos y gestión de interrupciones altamente optimizadas. Con potentes periféricos integrados como se aprecia en la siguiente ilustración, estas soluciones de control de chip único en tiempo real están diseñadas para una variedad de aplicaciones de control [9].

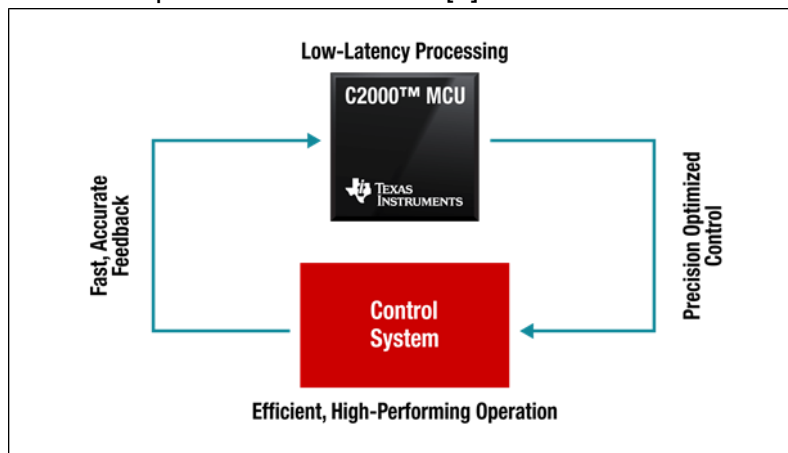


Ilustración 1. 4 Aplicación para control en tiempo real Fuente: [9]

El núcleo C28x optimizado para matemáticas brinda a los diseñadores la flexibilidad para mejorar la eficiencia y la confiabilidad del sistema. Como un cruce entre un

microcontrolador y un procesador de señal digital, los controladores en tiempo real C2000 traen la densidad de código y la velocidad de ejecución de un DSP con la facilidad de uso y la accesibilidad de un microcontrolador [9].

Aceleradores

La combinación de núcleo de la CPU C28x de alto rendimiento, se puede realizar una potencia de procesamiento rápida y eficiente para sistemas de control complejos en tiempo real como se puede apreciar en la ilustración 1.5. Hay cuatro aceleradores de hardware integrados en el chip disponibles que aumentan considerablemente el rendimiento de la MCU C2000 [9]:

- Acelerador de punto flotante
- Unidad de matemática compleja y unidad CRC (VCU)
- Unidad de matemáticas trigonométricas (TMU) acelerador
- Acelerador de la ley de control (CLA)

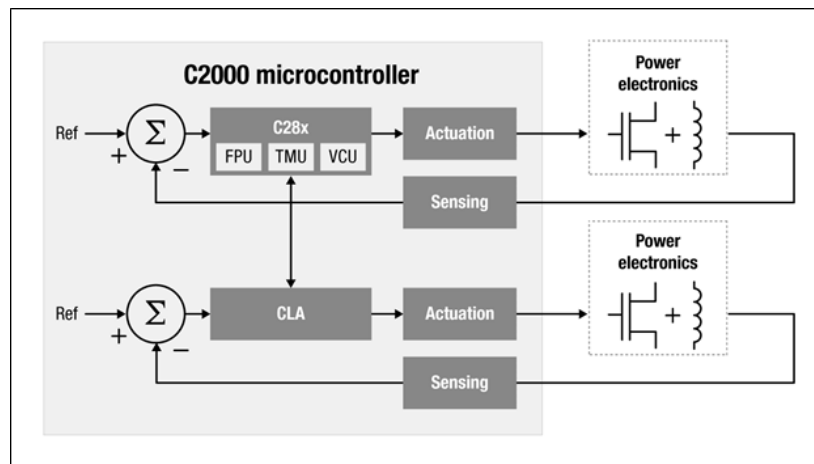


Ilustración 1.5 Aceleradores de leyes de control CLA. Fuente: [9]

Los microcontroladores TMS320F28069M cuentan con un núcleo C28x y el CLA (Acelerador de leyes de control), junto con periféricos de control altamente integrados en dispositivos de bajo número de pines. Es compatible con el código con el código basado en C28x, contando también con un alto nivel de integración analógica [10].

Entre los módulos destacables cuenta con HRPWM para permitir el control de borde doble (modulación de frecuencia), comparadores analógicos con referencias internas

de 10 bits y se pueden enrutar directamente para controlar las salidas de ePWM. El ADC convierte de 0 a 3.3 V rango fijo de escala completa y admite referencias $V_{REFL A}$ / $V_{REFL O}$. El módulo ADC se ha optimizado para bajos gastos generales y latencia [10].

1.2 Nomenclatura TMS320F28069

Los dispositivos TMS y las herramientas de apoyo al desarrollo de TMS se han caracterizado completamente, y la calidad y confiabilidad del dispositivo se han demostrado completamente. En la siguiente Ilustración 1.6 muestra la caracterización del dispositivo F28069M definiendo el significado de cada marca. La revisión del dispositivo puede determinarse mediante los símbolos marcados en la parte superior del paquete [11].



Ilustración 1. 6 Microcontrolador TMS320F28069M

- C=Código de fabrica
- A= Esta revisión de Silicon está disponible como TMS.
- 39A78JW = Código de seguimiento del lote
- 39= Código de año / mes de 2 dígitos
- A78J= lote de ensamblado
- W=código de sitio de ensamble
- G4=Verde (bajo en halógenos y compatible con RoHS).

Tabla 1. 1 Nomenclatura TMS320F28069M fuente: [1]

Prefijo	Familia De Dispositivo	Tecnología	Dispositivo	Tipo De Paquete	Rango De Temperatura
Dispositivo Comprobado y calificado	Familia de MCU TMS320	Flash	28069M	Paquete plano cuádruple de perfil bajo PZ de 100 pines	-40°C a 105°C
TMS	320	F	28069M	PZ	T

Descripción y características generales del MCU

En la tabla 1.2 se realiza una descripción grandes rasgos acerca de las cualidades principales que comprende el MCU TMS320F28069M.

Tabla 1. 2 Parámetros Y Características del TMS320f28069M

CARACTERÍSTICA	28069M (90 MHz)	Descripción general del dispositivo	
Tipo de paquete	100-Pin PZ	Características	
Ciclo de instrucción	11.11 ns	<ul style="list-style-type: none"> • CPU de 32 bits de alta eficiencia (TMS320C28x) -90 MHz (11.11-ns Cycle Time) -16 x 16 y 32 x 32 Operaciones de multiplicación y acumulación (MAC) -16 x 16 Dual MAC -Bus de Arquitectura Harvard -Operaciones atómicas -Rápida respuesta de interrupción y procesamiento -Modelo de Programación de Memoria Unificada -Código Eficiente (en C / C ++ y Assembly) • Unidad de punto flotante (FPU) -Operaciones de punto flotante de precisión única nativa • Acelerador de Ley de Control Programable (CLA) -Acelerador matemático de punto flotante de 32 bits 	<ul style="list-style-type: none"> • Bloque de expansión de interrupción periférica (PIE) que admite todas las interrupciones periféricas • Tres temporizadores de CPU de 32 bits • Periféricos de control avanzado • Hasta 8 módulos de modulador de ancho de pulso mejorado (ePWM) -Total de 16 canales PWM (8 HRPWM-Capable) -Temporizador independiente de 16 bits en cada módulo • Tres módulos de captura mejorada de entrada (eCAP) • Hasta 4 módulos de captura de alta resolución (HRCAP) • Hasta 2 módulos de pulsos de codificador de cuadratura mejorados (eQEP) • Convertidor de analógico a digital (ADC) de 12 bits, doble muestreo y retención (S / H) -Hasta 3.46 MSPS -Hasta 16 canales. • Sensor de temperatura en chip
Unidad de punto flotante (FPU)	Si		
VCU	Si		
CLA	Si		
6-canales DMA	Si		
On-chip Flash (16-bit word)	128K		
On-chip SARAM (16-bit word)	50K		
Código de seguridad para on-chip Flash, SARAM, and OTP blocks	Si		
Boot ROM (32K x 16)	Si		
One-time programable (OTP) ROM (16-bit word)	1K		
Canales ePWM	16		
Canales ePWM de alta resolución	8		
Entradas eCAP	3		
HRCAP	4		

Módulos eQEP		2	-Ejecuta código independientemente de la CPU principal	<ul style="list-style-type: none"> • Llave de seguridad de 128 bits y bloqueo -Protege los bloques de memoria segura -Previene la ingeniería inversa de firmware • Periféricos de puerto serie -Dos módulos de interfaz de comunicaciones serie (SCI) [UART] -Dos Módulos de Interfaz Periférica Serial (SPI) -Un bus de circuito integrado (I2C) -Un bus de puerto serie con almacenamiento intermedio multicanal (McBSP) -Una red de área de controlador mejorada (eCAN) -Bus serie universal (USB) 2.0 (consulte la Tabla de comparación de dispositivos para conocer la disponibilidad) -Modo de dispositivo de velocidad completa - Modo host de velocidad completa o velocidad baja • Hasta 54 pines de entrada / salida de uso general (GPIO) multiplexados y individualmente programables con filtrado de entrada • Funciones avanzadas de emulación -Análisis y funciones de punto de interrupción. -Depuración en tiempo real a través de hardware • Opciones de paquete -PZ PowerPAD™ de 100 pines -Mochilas planas de bajo perfil (LQFP) de 100 pines PZ • Temperatura -T: -40 ° C a 105 ° C
Watchdog timer		Si		
12-Bit ADC	MSPS	3.46		
	Tiempo de Conversión	289 ns	-Extiende el conjunto de instrucciones C28x para soportar la multiplicación compleja, las operaciones de Viterbi y la verificación de redundancia cíclica (CRC)	
	Canales	16		
	Sensor de temperatura	Si		
	Dos canales de muestreo y retención	Si		
32-Bit CPU Timers		3	-2 KB de ROM programable de una sola vez (OTP)	
Comparadores con DACs integrados		3	• Acceso directo a la memoria de 6 canales (DMA)	
I2C		1	• Bajo costo del dispositivo y del sistema	
McBSP		1	-Suministro individual de 3.3 V	
eCAN		1	-Sin requisito de secuencia de poder	
SPI		2	-Reinicio de encendido integrado y reinicio de reducción de tensión	
SCI		2	-Modos de operación de bajo consumo	
USB		1	-Sin pin de soporte analógico	
2-pin Oscilador		1		
0-pin Oscilador		2		
Pines de E / S (compartidos)	GPIO	54	• Soporte de exploración de límites JTAG	
	AIO	6	-IEEE Standard 1149.1-1990 Puerto de acceso de prueba estándar y arquitectura de exploración de límites	
Interrupciones externas		3	• Reloj	
Tensión de alimentación (nominal)		3.3 V	-Dos osciladores internos de cero pin	
Opciones de temperatura	T: -40°C-105°C	PZ		

Diagrama en bloques TMS320F28069M

En la ilustración 1.7 se muestra un diagrama de bloques funcional del dispositivo donde se puede observar la integración de cada una de las partes antes descritas y su interrelación con el resto de sistemas que comprende el dispositivo.

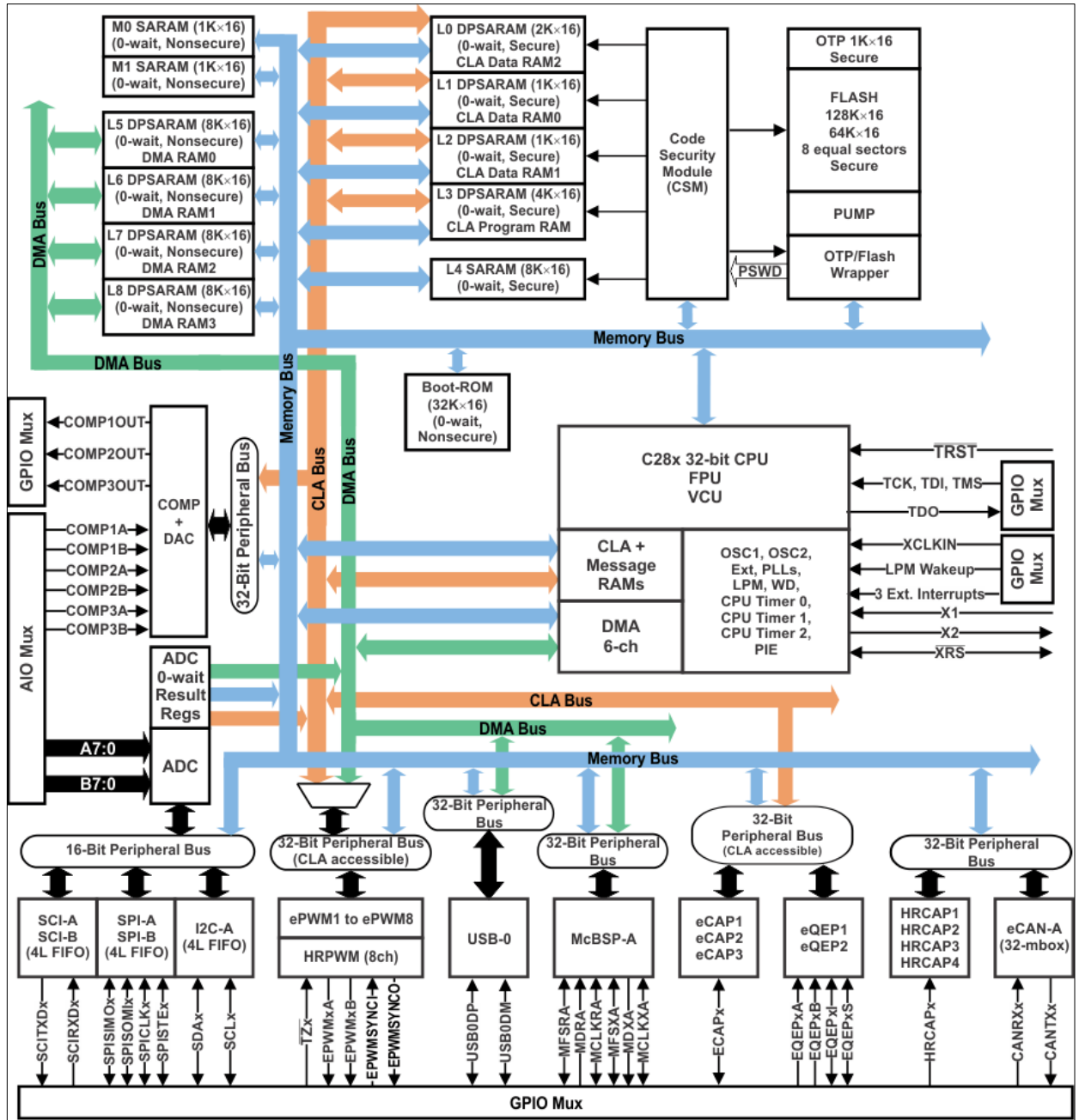
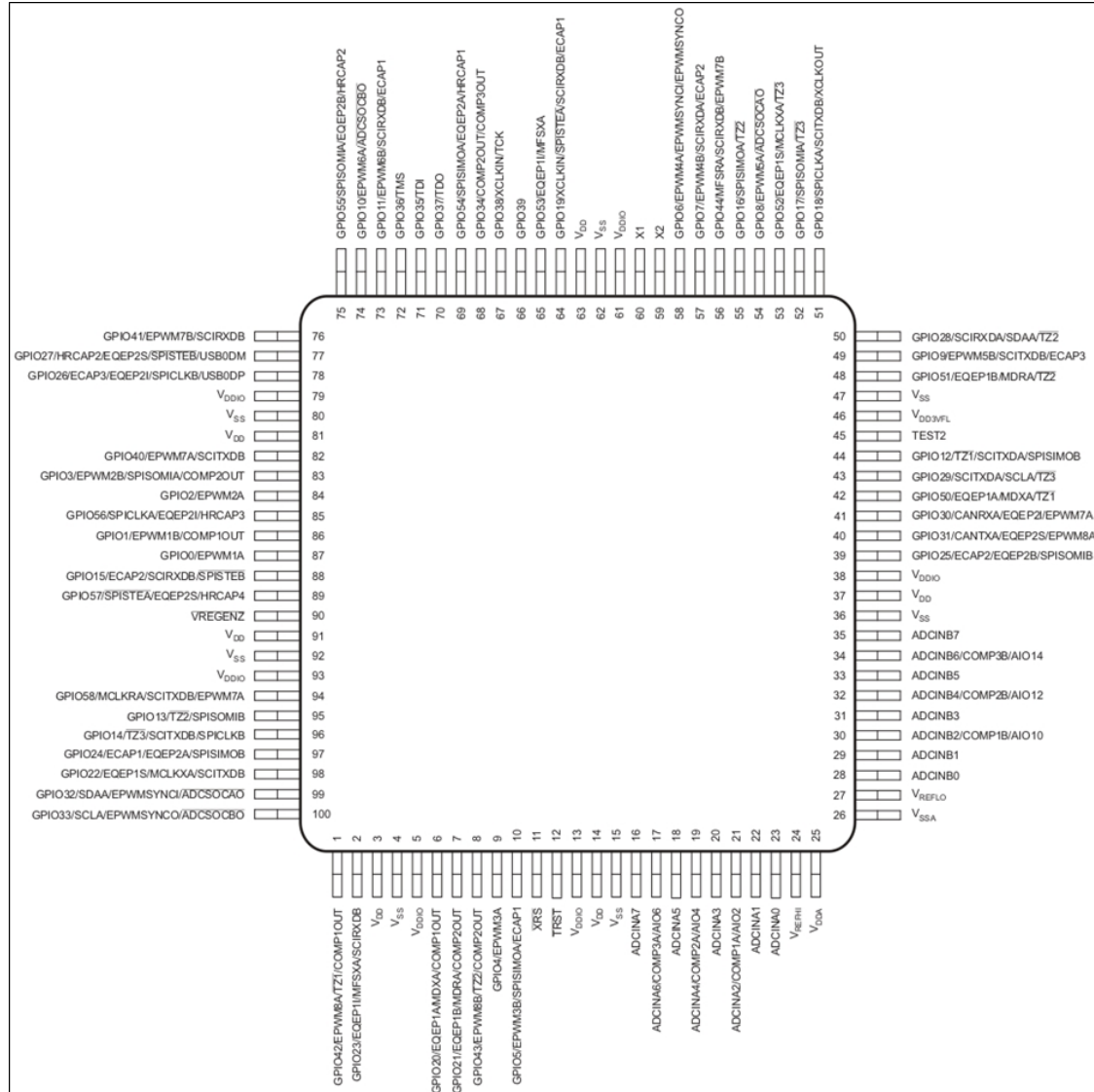


Ilustración 1. 7 Diagrama de bloques funcionales TMS320F28069M. Fuente: [12]

Vista superior TMS320F28069M

En la ilustra 1.9 se puede identificar la organización y ubicación de los pines que tiene el microcontrolador seleccionado para la aplicación donde se pueden observar los distintos periféricos ya descritos en la tabla anterior.



1.3 Placa de desarrollo

LAUNCHXL-F28069M

El C2000™ Piccolo™ LaunchPad™, LAUNCHXL-F28069M, es una placa de desarrollo completa de bajo costo para los dispositivos Texas Instruments Piccolo F2806x. El kit LAUNCHXL-28069M seleccionado presenta alta eficiencia de hardware y software necesarios para desarrollar aplicaciones en el control de motores donde se pueden evaluar desarrollo a nivel académico u investigativo. El LaunchPad se basa en el dispositivo F28069M el cual permite la migración fácilmente a dispositivos F2806x de menor costo una vez que se termine la aplicación para la cual fue seleccionada en este caso el control del motor de inducción. Ofrece una herramienta de emulación JTAG incorporada que permite la interfaz directa a una PC para una fácil programación, depuración y evaluación. Además de la emulación JTAG, la interfaz USB proporciona una conexión serie receptor / transmisor asíncrono universal (UART) desde el dispositivo F2806x al equipo principal[12].

Como se muestra en la ilustración 1.19 se muestran las funciones de LAUNCHXL-F28069M C2000 entra las cuales incluye[12] :

- Interfaz de programación y depuración USB a través de una sonda de depuración XDS100v2 aislada galvánicamente de alta velocidad con conexión USB / UART.
- Dispositivo conjunto F28069M que permite que las aplicaciones migren fácilmente a dispositivos de menor costo.
- Dos LED's de usuario.
- Botón de reinicio del dispositivo.
- Pines de dispositivo de fácil acceso para propósitos de depuración o como enchufes para agregar tableros de extensión personalizados.
- La biblioteca InstaSPIN en ROM, que permite la implementación de las soluciones InstaSPIN-MOTION e InstaSPIN-FOC.
- Interfaces de codificador dual de 5 V en cuadratura.
- Interfaz CAN con transceptor integrado.
- Botones de selección de arranque

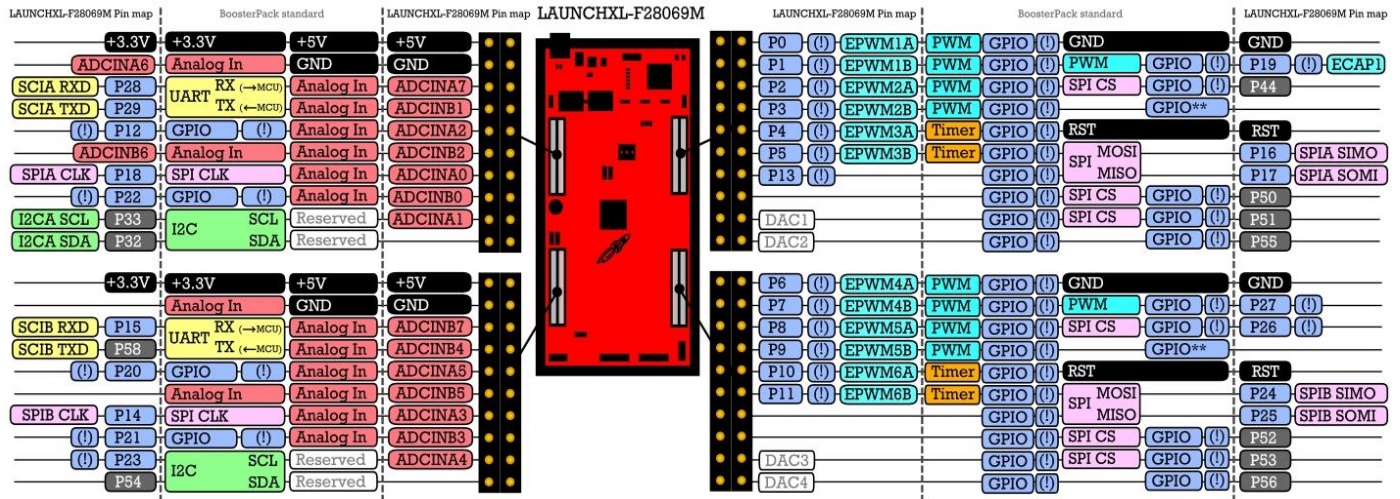


Ilustración 1. 9 Vista general LAUNCHXL-F28069M. Fuente: [12]

Disposición

Para Launchpad F28069M obtenga su mayor rendimiento y mejor eficiencia a nivel software se programa con Code Composer Studio debido a que las modulaciones optimizada que se explicaran más adelante necesitan una eficiencia alta de código. Texas instruments ofrece también una serie de herramienta para desarrollar aplicaciones en la herramienta computacional Matlab, pero no se aplicó por las razones antes expuestas.

Code compositor studio

Code Composer Studio es un entorno de desarrollo integrado (IDE) que admite la cartera de Microcontroladores y procesadores integrados de TI ilustración 1.10. Code Composer Studio comprende un conjunto de herramientas que se utilizan para desarrollar y depurar aplicaciones integradas. Incluye un compilador C / C ++ optimizador, un editor de código fuente, un entorno de compilación de proyectos, un depurador, un generador de perfiles y muchas otras características. El IDE intuitivo proporciona una única interfaz de usuario que lo lleva a través de cada paso del flujo de desarrollo de la aplicación [13].



Ilustración 1. 10 Code composer studio. Fuente: [13]

Configuración del hardware.

Para configurar la placa F28069M LaunchPad de acuerdo a la aplicación a desarrollar se realiza de acuerdo a la siguiente tabla[12].

- **Dominio de la energía.**

El LaunchPad F28069M tiene varios dominios de alimentación diferentes para habilitar el aislamiento de JTAG tabla 1.3. Los puentes JP1, JP2, JP3, JP4 y JP5 configuran dónde se pasa la alimentación[12].

Tabla 1. 3 Descripción del jumper JPx

Jumper	Dominio de poder
JP1	Habilitar 3.3 V desde USB (desactiva el aislamiento).
JP2	Habilitar GND desde USB (deshabilita el aislamiento).
JP3	Habilitar el conmutador de 5 V (apagado del suministro de 3,3 V del dispositivo de destino).
JP4	Conecta el objetivo MCU 3.3 V al segundo conjunto de encabezados Booster Pack.
JP5	Conecta la MCU de destino 5V al segundo conjunto de encabezados de Booster Pack.

- **Conectividad en serie.**

El LAUNCHXL-F28069M tiene un adaptador USB a UART incorporado. Esto facilita la impresión de la información de depuración en la PC host incluso en entornos aislados. El dispositivo F28069M en este LaunchPad contiene dos periféricos SCI

(UART), mientras que el LaunchPad tiene tres lugares donde se deben enrutar estos periféricos. El enrutamiento se configura a través de dos puentes (JP6 y JP7) tabla 1.4. Configure los puentes como se muestra en la Tabla para la conectividad en serie que desea[12].

Tabla 1. 4 Serial Connectivity fuente

MUX_SEL (JP7)	CH_SEL (JP 6)	Función
ON	ON	USB / UART desactivado; J1.3 y J1.4g –GPIO 28 n GPIO 29; J7.3 y J7.4 –GPIO15 y GPIO 58.
ON	OFF	USB / UART - GPIO 28 n GPIO 29, J1.3 y J1.4 - Hi-Z; J7.3 y J7.4 - GPIO15 y GPIO 58.
OFF	ON	USB / UART - GPIO15 y GPIO58; FALLA / OCTW - GPIO28 y GPIO29; J7.3 y J7.4 - Hi-Z.
OFF	OFF	USB / UART - GPIO15 y GPIO58; FALLA / OCTW - GPIO28 y GPIO29; J7.3 y J7.4 - Hi-Z.

- **Selección del modo de arranque.**

Para la selección del modo de arranque del F28069M LaunchPad el cual incluye una ROM de inicio que realiza algunas verificaciones básicas de inicio y permite que el dispositivo arranque de muchas maneras diferentes, se ha proporcionado S1 apreciable en la ilustración 1.11, para permitir a los usuarios configurar fácilmente los pines que la ROM de inicio verifica para tomar esta decisión[12].

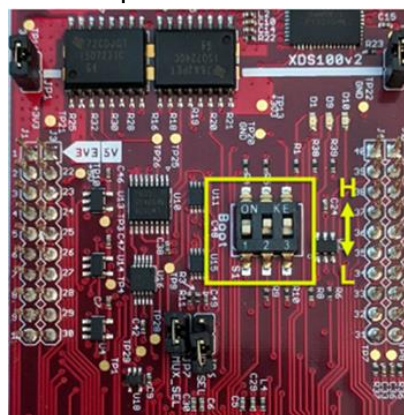


Ilustración 1. 11 Orientación del interruptor de arranque. Fuente: [12]

Los modos de arranque que se muestran en la siguiente Tabla 1.5 donde se pueden seleccionar usando S1:

Tabla 1. 5 Configuración del modo de arranque S1

El modo de inicio.	S1-Switch 1 (GPIO34) H = Pulled to 1 L = Pulled to 0	S1-Switch 2 (GPIO37 / DO) H = Pulled to 1 L = Pulled to 0	S1-Switch 3 (TRSTn) H = XDS100v2 (1) L = Tied to 0
Emulación de arranque.	L	H	H
IO paralelo.	L	L	L
SCI	H	L	L
Espera	L	H	L
Obtener modo	H	H	L

Hardware LAUNCHXL-F28069M

Pines dispuestos físicamente.

La organización física de los pines de conexión del hardware se puede observar en las tablas 1.6, 1.7, 1.8, y 1.9. Del anexo 3 al anexo 9 se muestran los todos los circuitos que comprende el hardware launchxl-F28069M[12].

Tabla 1. 6 J1 Pin y J3 Pin

Mux Value				J1 Pin	J3 Pin	Mux Value			
3	2	1	0			0	1	2	3
			+3.3V	1	21	+5V			
			ADCINA6	2	22	GND			
			J1.3	3	23	ADCINA7			
			J1.4	4	24	ADCINB1			
SPISIMOB	SCITXDA	TZ1	GPIO12	5	25	ADCINA2			
			ADCINB6	6	26	ADCINB2			
XCLKOUT	SCITXDB	SPICLKA	GPIO18	7	27	ADCINA0			
SCITXDB	MCLKXA	EQEP1S	GPIO22	8	28	ADCINB0			
ADCSOCBO	EPWMSYNCO	SCLA	GPIO33	9	29	ADCINA1			
ADCSOCAO	EWPMYNCI	SDAA	GPIO32	10	30	NC			

Tabla 1. 7 J4 Pin y J2 Pin.

Mux Value				J4 Pin	J2 Pin	Mux Value			
3	2	1	0			0	1	2	3
Rsvd	Rsvd	EPWM1A	GPIO0	40	20	GND			
COMP1OUT	Rsvd	EPWM1B	GPIO1	39	19	GPIO19	SPISTEA	SCIRXDB	ECAP1
Rsvd	Rsvd	EPWM2A	GPIO2	38	18	GPIO44	MFSRA	SCIRXDB	EPWM7B
COMP2OUT	SPISOMIA	EPWM2B	GPIO3	37	17	NC			
Rsvd	Rsvd	EPWM3A	GPIO4	36	16	RESET#			
ECAP1	SPISIMOA	EPWM3B	GPIO5	35	15	GPIO16	SPISIMOA	Rsvd	TZ2
SPISOMIB	Rsvd	TZ2	GPIO13	34	14	GPIO17	SPISOMIA	Rsvd	TZ3
			NC	33	13	GPIO50	EQEP1A	MDXA	TZ1
			DAC1	32	12	GPIO51	EQEP1B	MDRA	TZ2
			DAC2	31	11	GPIO55	SPISOMIA	EQEP2A	HRCAP1

Tabla 1. 8 J5 Pin y J7 Pin

Mux Value				J5 Pin	J7 Pin	Mux Value			
3	2	1	0			0	1	2	3
			+3.3V	41	61	+5V			
			NC	42	62	GND			
			J7.3	43	63	ADCINB7			
			J7.4	44	64	ADCINB4			
COMP1OUT	MDXA	EQEP1A	GPIO20	45	65	ADCINA5			
			NC	46	66	ADCINB5			
SPICLKB	SCITXDB	TZ3	GPIO14	47	67	ADCINA3			
COMP2OUT	MDRA	EQEP1B	GPIO21	48	68	ADCINB3			
SCIRXDB	MFSXA	EQEP1I	GPIO23	49	69	ADCINA4			
HRCAP1	EQEP2A	SPISIMOA	GPIO54	50	70	NC			

Tabla 1. 9 J8 Pin y J6 Pin

Mux Value				J8 Pin	J6 Pin	Mux Value			
3	2	1	0			0	1	2	3
Rsvd	Rsvd	EPWM4A	GPIO6	80	60	GND			
COMP1OUT	Rsvd	EPWM4B	GPIO7	79	59	GPIO27	HRCAP2	EQEP2S	SPISTEB
Rsvd	Rsvd	EPWM5A	GPIO8	78	58	GPIO26	ECAP3	EQEP2I	SPICLKB
COMP2OUT	Rsvd	EPWM5B	GPIO9	77	57	NC			
Rsvd	Rsvd	EPWM6A	GPIO10	76	56	RESET#			
ECAP1	Rsvd	EPWM6B	GPIO11	75	55	GPIO24	ECAP1	EQEP2A	SPISIMOB
			NC	74	54	GPIO25	ECAP2	EQEP2B	SPISOMIB
			NC	73	53	GPIO52	EQEP1S	MCLKXA	TZ3
			DAC3	72	52	GPIO53	EQEP1I	MFSXA	Rsvd
			DAC4	71	51	GPIO56	SPICLKA	EQEP2I	HRCAP3

1.4 Control Sensorless

El control de un motor de inducción es complejo y exigen el uso de algoritmos de control de alto rendimiento como el "control Sensorless" y un potente DSP para ejecutar este algoritmo. Un motor ACI puede ser conducido a un estado estable mediante simples esquemas alimentados por voltaje, controlados por corriente o controlados por velocidad[14].

El control desarrollado se fundamenta en el principio del lazo mostrado en la ilustración 1.12 donde el valor de la velocidad del rotor depende de un estimador de flujo y velocidad, la estimación del flujo del estator basado en el modelo de voltaje, la corriente del estator fundamentado con su proporcionalidad con la velocidad del rotor mediante las transformaciones de Clarke[15].

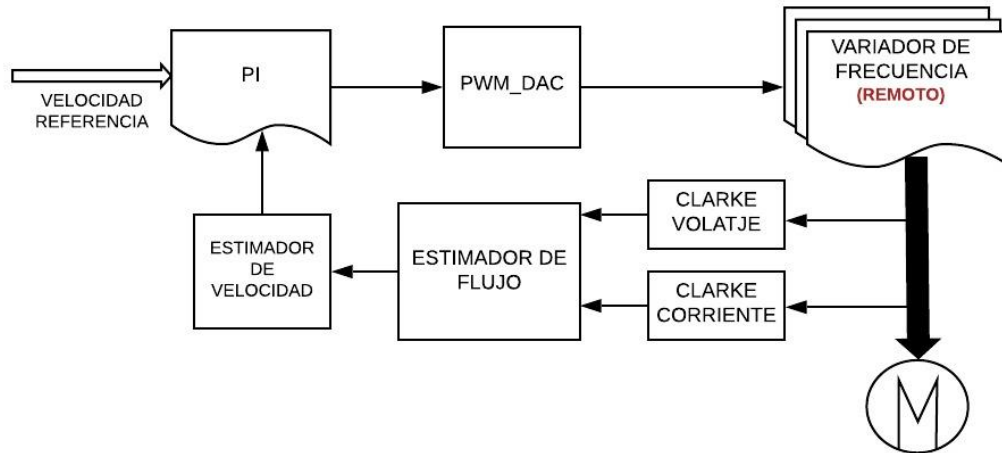


Ilustración 1. 12 Esquema de control Sensorless planteado .Fuente: autor

El control sensorless planteado es el escalar el cual se el cual se fundamenta de tal forma tal que, al reducirse la frecuencia, el voltaje se reduzca proporcionalmente, o sea con la ley de Mando Voltaje / Frecuencia Constante[14].

En la práctica, la relación de tensión a frecuencia del estator se basa generalmente en los valores nominales de estas variables. El perfil típico de $\frac{V}{Hz}$ se puede mostrar en la siguiente figura. Básicamente, hay tres rangos de velocidad en el perfil de $\frac{V}{Hz}$ de la siguiente manera[15]:

- A $0 - f_c$ Hz, se requiere un voltaje, por lo que la caída de voltaje a través de la resistencia del estator no puede descuidarse y debe compensarse aumentando los Vs. Entonces, el perfil $\frac{V}{Hz}$ no es lineal. La frecuencia de corte (f_c) y los voltajes de estator adecuados se pueden calcular analíticamente a partir del circuito equivalente de estado estable con $R_s \neq 0$.
- En $f_c - f_{rated}$ Hz, sigue la relación constante $\frac{V}{Hz}$. La pendiente en realidad representa la cantidad de flujo de espacio de aire como se ve en la Ecuación.
- Con Hz de frecuencia más alta, la relación $\frac{V}{Hz}$ constante no se puede satisfacer porque los voltajes del estator se limitarían al valor nominal para evitar la

ruptura del aislamiento en los devanados del estator. Por lo tanto, el flujo de espacio de aire resultante se reduciría, y esto provocará inevitablemente la disminución del par desarrollado en consecuencia. Esta región es usualmente llamada "región de debilitamiento de campo".

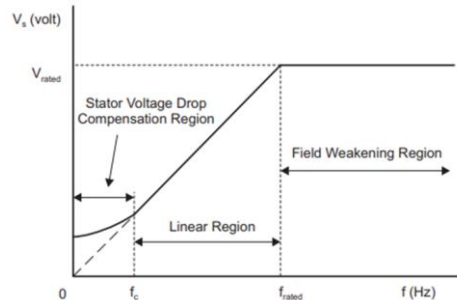


Ilustración 1. 13 Ley de control escalar. Fuente: [16].

Esta ley de control se configurara en el convertidor de frecuencia expuesto en los próximos capítulo el cual se usa como elemento final de control.

- La transformada de Clarke

La transformada de Clarke convierte las componentes del dominio del tiempo de un sistema de tres fases (de un marco abc) en dos componentes de un marco estacionario ortogonal ($\alpha\beta$). Aquí se supone que las tres fases están equilibradas (es decir, $I_a + I_b + I_c = 0$) y tienen una secuencia positiva (ABC) de la siguiente manera[16]:

$$I_a = I * \cos(\omega t) \quad 1$$

$$I_b = I * \cos(\omega t - 2\pi/3) \quad 2$$

$$I_c = I * \cos(\omega t - 4\pi/3) \quad 3$$

$$\begin{cases} I_\alpha = I_a \\ I_\beta = (2I_b + I_c)/\sqrt{3} \end{cases} \quad \text{Lo cual resulta} \quad \begin{cases} I_\alpha = I * \cos(\omega t) \\ I_\beta = I * \sin(\omega t) \end{cases}$$

Esta transformación convierte cantidades equilibradas de tres fases en cantidades equilibradas de cuadratura de dos fases como se muestra en la figura a continuación.

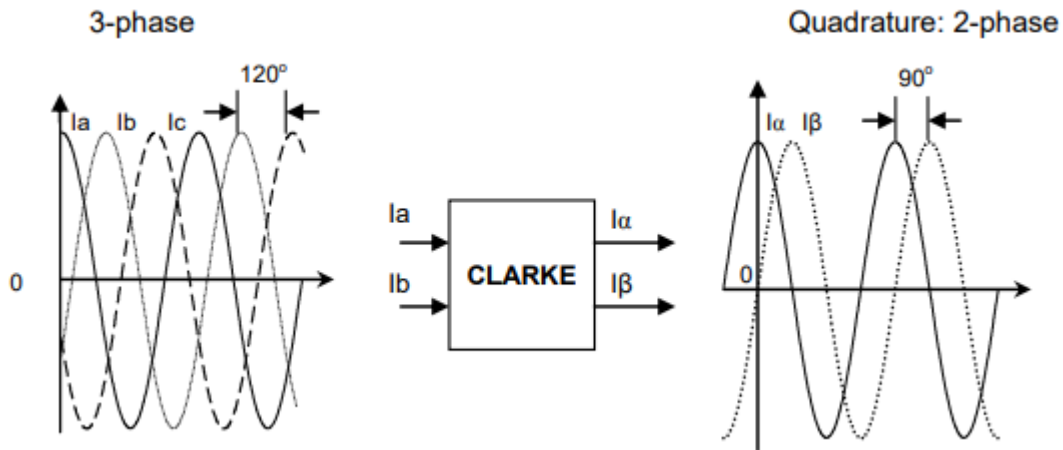


Ilustración 1. 14 Transformada de Clarke. Fuente: [16]

Se realiza la transformada de Clarke tanto a la corriente como al voltaje para alimentar el bloque de estimación de flujo descrito a continuación.

Modelo matemático de motores de inducción

El modelo de motor de inducción se puede expresar en el marco de referencia fijo d-q por ecuaciones de acuerdo a las ilustraciones 1.14[17].

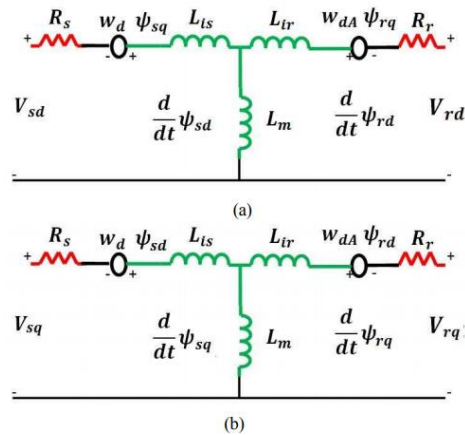


Ilustración 1. 15 Circuito equivalente de motor de inducción. (a) En el marco de referencia d. (b) En q- marco de referencia .Fuente:[17]

$$V_{sdq} = R_s i_{sdq} + \frac{d}{dt} \Psi_{sdq} - j w_g \Psi_{sdq} \quad 4$$

$$0 = R_r i_{rdq} + \frac{d}{dt} \Psi_{rdq} - j(w_g - w_r) \Psi_{rdq} \quad 5$$

$$\Psi_{sdq} = L_s i_{sdq} + L_m i_{rdq} \quad 6$$

$$\Psi_{rdq} = L_r i_{rdq} + L_m i_{sdq} \quad 7$$

w_g, w_r , son referencias genéricas del sistema, velocidad del rotor eléctrico. R_s, R_r son las resistencias del estator y del rotor. L_s, L_r, L_m Son inductancias del estator, el rotor y las magnetizante. (Ψ_{sdq}) , es el flujo del estator en el marco d-q. (Ψ_{rdq}) , es el flujo del rotor en el marco d-q. (i_{sdq}, i_{rdq}) , son las corrientes del estator y del rotor en el marco d-q[17].

- Estimación de la Flujo

Para la estimación de la velocidad se implementa el estimador de flujo con el ángulo de flujo del rotor para el motor de inducción de 3-ph basado en la integral del enfoque de back emf (modelo de voltaje). Para reducir los errores debidos a la integración pura y la medición de resistencia del estator, se introducen los voltajes compensados producidos por los compensadores PI. Por lo tanto, este estimador de flujo puede funcionar en un amplio rango de velocidades, incluso a muy baja velocidad[16].

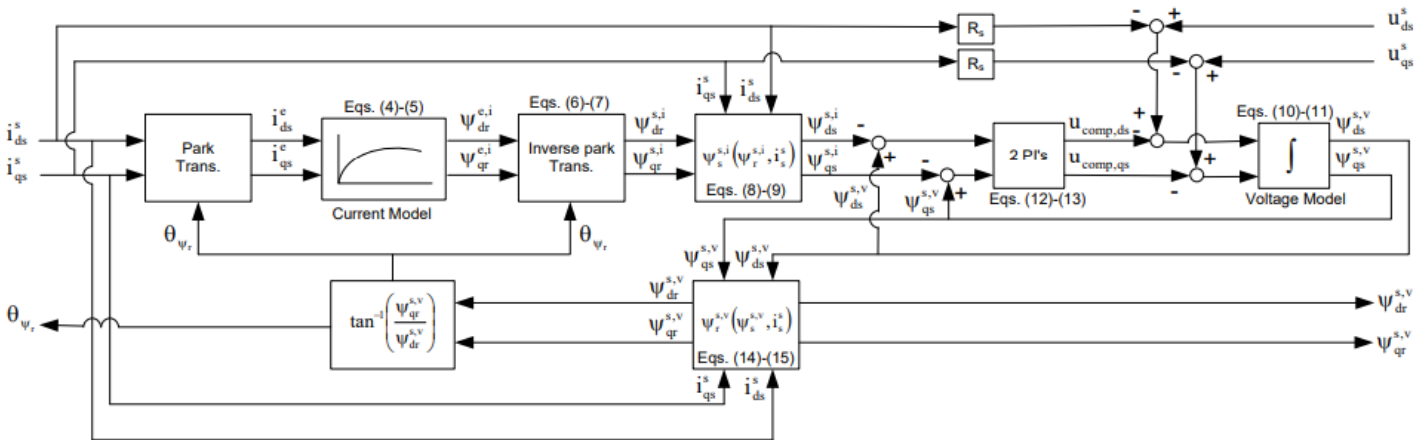


Ilustración 1. 16 Diagrama en bloques de estimador de flujo. Fuente:[16]

El total del estimador de flujo se puede mostrar en la ilustración 1.15. Los enlaces de flujo del rotor en el marco de referencia estacionario se calculan principalmente por medio de la integral de la fem inversa en el modelo de voltaje. Al introducir los voltajes compensados generados por los compensadores PI, se pueden solucionar los errores asociados con el integrador puro y la medición de la resistencia del estator. Las ecuaciones derivadas a partir de las ecuaciones expuestas con anterioridad para este estimador de flujo se resumen a continuación[18]:

En primer lugar, la dinámica de enlace del flujo del rotor en el marco de referencia de rotación sincrónica ($W = W_e = W_{\psi_r}$) [16].

$$\frac{d\psi_{dr}^{ei}}{dt} = \frac{L_m}{\tau_r} i_{ds}^{ei} - \frac{1}{\tau_r} \psi_{dr}^{ei} + (w_e - w_r) \psi_{qr}^{ei} \quad 8$$

$$\frac{d\psi_{qr}^{ei}}{dt} = \frac{L_m}{\tau_r} i_{qs}^{ei} - \frac{1}{\tau_r} \psi_{qr}^{ei} - (w_e - w_r) \psi_{dr}^{ei} \quad 9$$

Donde L_m es la inductancia magnetizante (H), $\tau_r = \frac{L_r}{R_r}$ es la constante de tiempo del rotor (seg), y w_r es la velocidad angular angular del rotor (rad / seg).

En el modelo actual, el enlace de flujo total del rotor se alinea en el componente del eje d, que está modelado por las corrientes del estator, por lo tanto[16].

$$\Psi_r^{ei} = \Psi_{dr}^{ei} \text{ y } \Psi_{qr}^{ei} = 0 \quad 10$$

Sustituyendo $\Psi_{qr}^{ei} = 0$, produce la dinámica de flujo del rotor orientado son:

$$\frac{d\Psi_{dr}^{ei}}{dt} = \frac{L_m}{\tau_r} i_{ds}^{ei} - \frac{1}{\tau_r} \Psi_{dr}^{ei} \quad 11$$

$$\Psi_{qr}^{ei} = 0 \quad 12$$

Tenga en cuenta que (11) y (12) son las ecuaciones de control de vector de flujo de rotor clásico. Luego, los enlaces de flujo del rotor en (11) - (12) se transforman en el marco de referencia estacionario realizado por la transformación del parque inverso[16].

$$\Psi_{dr}^{si} = \Psi_{dr}^{ei} \cos(\theta_{\psi_r}) - \Psi_{qr}^{ei} \sin(\theta_{\psi_r}) = \Psi_{dr}^{ei} \cos(\theta_{\psi_r}) \quad 13$$

$$\Psi_{qr}^{si} = \Psi_{dr}^{ei} \sin(\theta_{\psi_r}) + \Psi_{qr}^{ei} \sin(\theta_{\psi_r}) = \Psi_{dr}^{ei} \sin(\theta_{\psi_r}) \quad 14$$

Donde θ_{ψ_r} es el ángulo de flujo del rotor (rad).

Luego, los enlaces de flujo del estator en el marco de referencia estacionario se calculan a partir de los enlaces de flujo del rotor en (13) - (14) [16].

$$\Psi_{dr}^{si} = L_s i_{ds}^s - L_m i_{dr}^s = \left(\frac{L_s L_r - L_m^2}{L_r} \right) i_{dr}^s - \frac{L_m}{L_r} \Psi_{dr}^{si} \quad 15$$

$$\Psi_{qr}^{si} = L_s i_{qs}^s - L_m i_{qr}^s = \left(\frac{L_s L_r - L_m^2}{L_r} \right) i_{qr}^s - \frac{L_m}{L_r} \Psi_{qr}^{si} \quad 16$$

Donde L_s y L_r son la auto inductancia del estator y del rotor (H), respectivamente.

A continuación, los enlaces de flujo del estator en el modelo de voltaje se calculan mediante la integración de la fem posterior con voltajes compensados[16].

$$\Psi_{dr}^{sv} = \int (u_{ds}^s - i_{ds}^s R_s - u_{comp.ds}) dt \quad 17$$

$$\Psi_{qr}^{sv} = \int (\mathbf{u}_{qs}^s - \mathbf{i}_{qs}^s R_s - \mathbf{u}_{comp.qs}) dt \quad 18$$

Donde R_s es la resistencia del estator (Ω), \mathbf{u}_{ds}^s , \mathbf{u}_{qs}^s son voltajes estacionarios del estator del eje dq, y los voltajes compensados son calculados por la ley de control de PI de la siguiente manera[16].

$$\mathbf{u}_{comp.ds} = Kp(\Psi_{ds}^{sv} - \Psi_{ds}^{si}) + \frac{Kp}{T_1} \int (\Psi_{ds}^{sv} - \Psi_{ds}^{si}) dt \quad 19$$

$$\mathbf{u}_{comp.qs} = Kp(\Psi_{qs}^{sv} - \Psi_{qs}^{si}) + \frac{Kp}{T_1} \int (\Psi_{qs}^{sv} - \Psi_{qs}^{si}) dt \quad 20$$

La ganancia proporcional KP y el tiempo de reinicio TI se eligen de tal manera que los enlaces de flujo calculados por el modelo actual sean dominantes a baja velocidad porque las fem de respaldo calculadas por el modelo de voltaje son extremadamente bajas en este rango de velocidad (incluso las fem de retorno cero a velocidad cero)[19]. Mientras está en el rango de alta velocidad, los enlaces de flujo calculados por el modelo de voltaje son dominantes. Una vez que se calculan los enlaces de flujo del estator en (17) - (18), los enlaces de flujo del rotor basados en el modelo de voltaje se calculan, reorganizando (15) - (16), como[16].

$$\Psi_{dr}^{sv} = - \left(\frac{L_s L_r - L_m^2}{L_r} \right) i_{ds}^s + \frac{L_r}{L_m} \Psi_{ds}^{sv} \quad 21$$

$$\Psi_{qr}^{sv} = - \left(\frac{L_s L_r - L_m^2}{L_r} \right) i_{qs}^s + \frac{L_r}{L_m} \Psi_{qs}^{sv} \quad 22$$

Luego, el ángulo de flujo del rotor basado en el modelo de voltaje finalmente se calcula como.

$$\tan^{-1} \left(\frac{\Psi_{qr}^{sv}}{\Psi_{dr}^{sv}} \right) \quad 23$$

- Estimador de velocidad

El estimador de velocidad de bucle abierto se deriva basándose en las ecuaciones matemáticas del motor de inducción en el marco de referencia estacionario. Los valores precisos de los parámetros de la máquina son inevitablemente requeridos, de

lo contrario puede ocurrir el error de velocidad en estado estable.[19] Sin embargo, la estructura del estimador es mucho más simple en comparación con otras técnicas avanzadas. Todas las ecuaciones representadas aquí están en el marco de referencia estacionario (con superíndice "s"). En primer lugar, las ecuaciones de enlace de flujo del rotor se pueden mostrar a continuación[16]

$$\lambda_{dr}^s = L_r i_{dr}^s + L_m i_{ds}^s \quad 24$$

$$\lambda_{qr}^s = L_r i_{qr}^s + L_m i_{qs}^s \quad 25$$

Donde L_r y L_m son inductancias de rotor y magnetizante (H), respectivamente. Según las ecuaciones (25) - (26), las corrientes del rotor se pueden expresar como[16].

$$i_{dr}^s = \frac{1}{L_r} (\lambda_{dr}^s - L_m i_{ds}^s) \quad 26$$

$$i_{qr}^s = \frac{1}{L_r} (\lambda_{qr}^s - L_m i_{qs}^s) \quad 27$$

En segundo lugar, las ecuaciones de voltaje del rotor se utilizan para encontrar la dinámica de enlace del flujo del rotor[16].

$$0 = R_r i_{dr}^s - w_r \lambda_{qr}^s - \frac{d\lambda_{dr}^s}{dt} \quad 28$$

$$0 = R_r i_{qr}^s - w_r \lambda_{dr}^s - \frac{d\lambda_{qr}^s}{dt} \quad 29$$

Donde w_r es la velocidad del rotor eléctricamente angular (rad / seg), y R_r es la resistencia del rotor (Ω). Sustituyendo las corrientes del rotor de (26) - (24) en (28) - (29), la dinámica de enlace de flujo del rotor se puede encontrar como[16].

$$\frac{d\lambda_{dr}^s}{dt} = \frac{1}{t_r} \lambda_{dr}^s + \frac{L_m}{t_r} i_{ds}^s - w_r \lambda_{qr}^s \quad 30$$

$$\frac{d\lambda_{qr}^s}{dt} = \frac{1}{t_r} \lambda_{qr}^s + \frac{L_m}{t_r} i_{qs}^s - w_r \lambda_{dr}^s \quad 31$$

Donde $t_r = \frac{L_r}{R_r}$ es la constante de tiempo del rotor (seg). Suponga que los enlaces de flujo del rotor en (30) - (31) son conocidos, por lo tanto, su magnitud y ángulo pueden calcularse como.) [16].

$$\lambda_r^s = \sqrt{(\lambda_{dr}^s)^2 + (\lambda_{qr}^s)^2} \quad 32$$

$$\theta_{\lambda_r} = \tan^{-1} \left(\frac{\lambda_{qr}^s}{\lambda_{dr}^s} \right) \quad 33$$

A continuación, la velocidad del flujo del rotor (es decir, sincrónica), ω_e , se puede calcular fácilmente mediante la derivada del ángulo de flujo del rotor en (33).

$$\omega_e = \frac{d\theta_{\lambda_r}}{dt} = \frac{d \left(\tan^{-1} \left(\frac{\lambda_{qr}^s}{\lambda_{dr}^s} \right) \right)}{dt} \quad 34$$

En referencia a la tabla de derivadas, la ecuación (34) se puede resolver como

$$\frac{d(\tan^{-1} u)}{dt} = \frac{1}{1+u^2} \frac{du}{dt} \quad 35$$

Donde $u = \frac{\lambda_{qr}^s}{\lambda_{dr}^s}$

$$\omega_e = \frac{d\theta_{\lambda_r}}{dt} = \frac{(\lambda_{dr}^s)^2}{(\lambda_r^s)^2} \left(\frac{\lambda_{dr}^s \frac{d\lambda_{qr}^s}{dt} - \lambda_{qr}^s \frac{d\lambda_{dr}^s}{dt}}{(\lambda_{dr}^s)^2} \right) \quad 36$$

Sustituyendo (30) - (31) en (36), y reorganizando, finalmente se obtiene.

$$\omega_e = \frac{d\theta_{\lambda_r}}{dt} = \omega_r + \frac{1}{(\lambda_r^s)^2} \frac{L_m}{\tau_r} (\lambda_{dr}^s i_{qs}^s - \lambda_{qr}^s i_{ds}^s) \quad 37$$

El segundo término de la mano izquierda en (37) se conoce como deslizamiento que es proporcional al par electromagnético cuando la magnitud del flujo del rotor se mantiene constante. El par electromagnético se puede mostrar aquí por conveniencia.

$$T_e = \frac{3}{2} \frac{p}{L_r} \frac{L_m}{L_r} (\lambda_{dr}^s i_{qs}^s - \lambda_{qr}^s i_{ds}^s) \quad 38$$

Donde p es el número de polos. Por lo tanto, la velocidad del rotor se puede encontrar como.

$$\omega_r = \omega_c - \frac{1}{(\lambda_r^s)^2} \frac{L_m}{\tau_r} (\lambda_{dr}^s i_{qs}^s - \lambda_{qr}^s i_{ds}^s) \quad 39$$

Ahora, el concepto por unidad se aplica a (39), luego, la ecuación (39) se convierte en.

$$\omega_{r,pu} = \omega_{c,pu} - \frac{1}{\omega_b \tau_r} \left(\frac{\lambda_{dr,pu}^s i_{qs,pu}^s - \lambda_{qr,pu}^s i_{ds,pu}^s}{(\lambda_{r,pu}^s)^2} \right) \quad pu \quad 40$$

Donde $\omega_b = 2\pi f_b$ es la velocidad angular eléctrica (rad / seg), $\lambda_b = L_m I_b$ es el enlace de flujo base (volt.seg), e I_b es la corriente base (amperio). Equivalentemente, otra forma es.

$$\omega_{r,pu} = \omega_{c,pu} - K_1 \left(\frac{\lambda_{dr,pu}^s i_{qs,pu}^s - \lambda_{qr,pu}^s i_{ds,pu}^s}{(\lambda_{r,pu}^s)^2} \right) \quad pu \quad 41$$

Donde $K_1 = \frac{1}{\omega_b \tau_r}$.

La velocidad síncrona por unidad se puede calcular como

$$\omega_{c,pu} = \frac{1}{2\pi f_b} \frac{d\theta_{\lambda_r}}{dt} = \frac{1}{f_b} \frac{d\theta_{\lambda_r,pu}}{dt} \quad pu \quad 42$$

Donde f_b es la frecuencia eléctrica base (suministrada) (Hz) y 2π es el ángulo base (rad).

En el capítulo 3 se describirá como se empleó a estos estimadores a nivel discretos para ser implementados en el DSP

2. DISEÑO DE LOS ALGORITMOS PRINCIPALES DE CONTROL DEL SISTEMA Y DESCRIPCION DEL ELEMENTO FINAL DE CONTROL A ADOPTAR.

Para controlar la planta en el sistema planteado de control visto en el capítulo anterior se adoptó un variador de frecuencia ABB ACS800-U1[20], el cual hace las veces de inversor de trifásico por cuestiones de factibilidad el cual está configurado únicamente en modo remoto con lo cual aseguramos que solo actuara de acuerdo la señal proveniente del sistema de control implementado en el DSP.

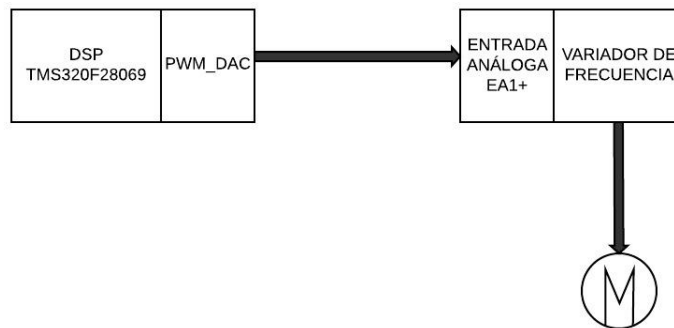


Ilustración 2. 1 Diagrama de control a lazo abierto. Fuente: autor

2.1 Variador de frecuencia

Un variador de frecuencia (siglas VFD, del inglés: Variable Frequency Drive o bien AFD Adjustable Frequency Drive) es un sistema para el control de la velocidad rotacional de un motor de corriente alterna (AC) por medio del control de la frecuencia de alimentación suministrada al motor[21]. Los dispositivos variadores de frecuencia operan bajo el principio de que la velocidad síncrona de un motor de corriente alterna (CA) está determinada por la frecuencia de AC suministrada y el número de polos en el estator, de acuerdo con la relación:

$$Wr = 120 * \frac{f}{P}$$

En los motores asíncronos las revoluciones por minuto son ligeramente menores por el propio asincronismo que indica su nombre. En estos se produce un desfase mínimo

entre la velocidad de rotación (RPM) del rotor (velocidad "real" o "de salida") comparativamente con la cantidad de RPM's del campo magnético (las cuales si deberían cumplir la ecuación arriba mencionada tanto en Motores síncronos como en motores asíncronos) debido a que sólo es atraído por el campo magnético exterior que lo aventaja siempre en velocidad (de lo contrario el motor dejaría de tener par en los momentos en los que alcanzase al campo magnético)[3].

La forma de variar la frecuencia básicamente consta de cambiar el ciclo de trabajo (tiempo ON y tiempo OFF en un período) de una onda cuadrada periódica, de tal forma que el valor medio de la tensión (el promedio) a lo largo del tiempo varíe entre V máximo y V mínimo. La velocidad con que variamos el ciclo de trabajo, o sea su valor medio, será la frecuencia de variación del valor medio[3].

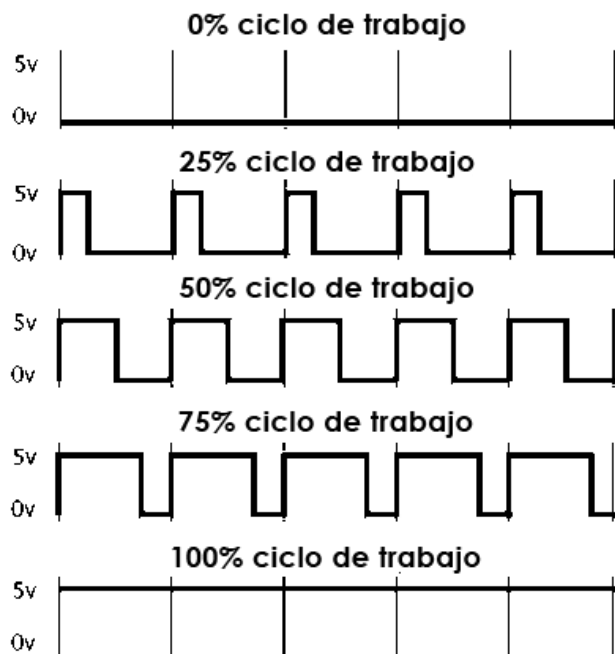


Ilustración 2. 2 Ciclo de trabajo PWM. Fuente[]

Esto físicamente se logra a través de "llaves electrónicas" de conmutación que son los IGBT (transistores bipolares de compuerta aislada) que actúan como interruptores que al cerrarse y abrirse por medio de un software específico conforman la onda cuadrada, que permite obtener la señal sinusoidal por medio de la variación del ciclo

de trabajo de las mencionada llaves electrónica como se puede apreciar a continuación[5].

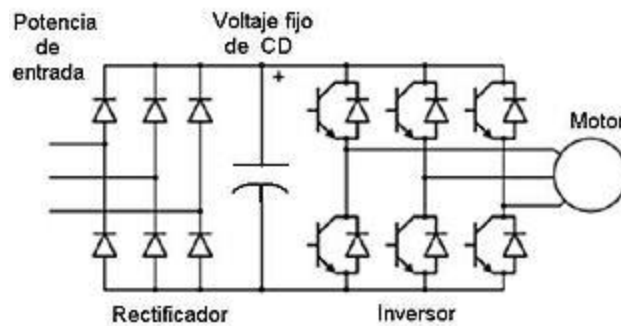


Ilustración 2. 3 Circuito interno variador de frecuencia

De acuerdo a esto se adoptó el variador de frecuencia industrial ABB ACS800-U1 como elemento final de control haciendo las veces de inversor trifásico proporcionara la alimentación trifásica necesaria para mover el motor.

Configuración de variador ABB ACS800-U1

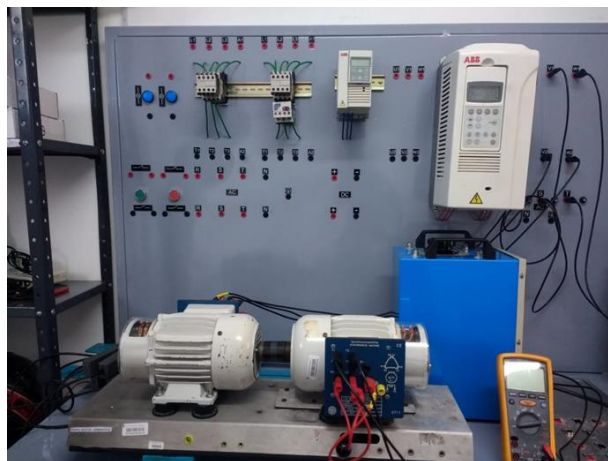


Ilustración 2. 4 Variador de frecuencia. Fuente: autor

Por facilidad para esta práctica el módulo se configuró de acuerdo a trabajos de investigación ya realizados con anterioridad tomando como referencia a [22] el cual en el laboratorio número 2 se visualiza la forma de configurar el variador de frecuencia para que actúe en modo remoto utilizando para variar la velocidad la entrada análoga de acuerdo a la siguiente ilustración.

1	VREF	Tensión de referencia 10 VCC
2	GND	$1 \text{ kohm} \leq R_L \leq 10 \text{ kohm}$
3	EA1+	Ref. velocidad 0(2) ... 10 V, $R_{in} > 200 \text{ kohm}$
4	EA1-	

Tal y como se puede apreciar en la anterior ilustración la referencia de voltaje para realizar el cambio de velocidad se requiere un circuito de acondicionamiento para aumentar la señal de referencia DC de voltaje empleada para variar la velocidad.

- Circuito amplificación de la señal de salida del DSP

La señal máxima de salida de cualquier pin del DSP utilizado es de 3,3 voltios como ya se describió en secciones anteriores dicho voltaje va a ser variado por una señal PWM la cual se le cambie el ciclo de trabajo dependiendo de la lógica de control implementada en el sistema. El circuito propuesto es en base al amplificador operacional LM358 en su configuración con retroalimentación negativa apreciable en la ilustración el cual es un amplificador que consisten de dos amplificadores operacionales independientes, con compensación de frecuencia de ganancia alta diseñados para operar un solo suministro o suministro dividido en una amplia gama de voltajes de fuente única de 3V- 32V, bajo consumo independiente de la magnitud del voltaje de la fuente de alimentación, y ancho de banda de 0,7MHz con lo cual esta muy por encima de la frecuencia de la señal PWM.

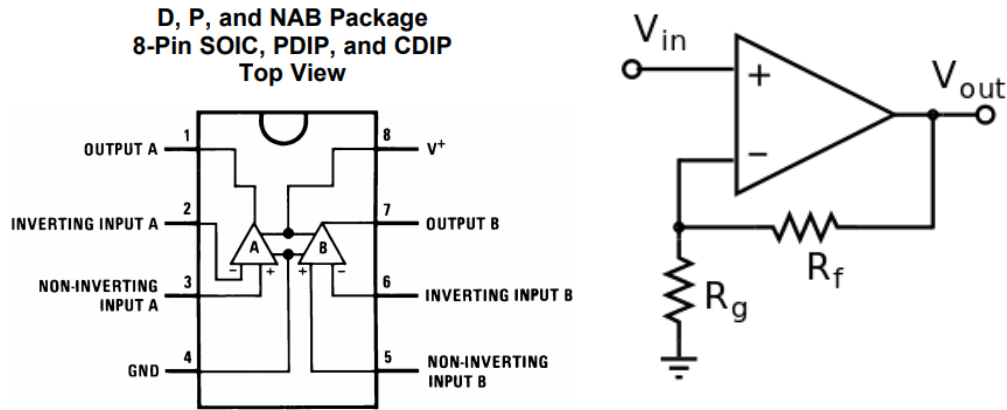


Ilustración 2. 5 LM358

Donde su ganancia viene dada de acuerdo a la ecuación

$$A_{CL} = \frac{V_{out}}{V_{in}} = \left(1 + \frac{R_f}{R_g} \right) \quad 43$$

El circuito se diseñó y simuló con ayuda de herramienta matemática Matlab apreciable en la siguiente ilustración en la cual se aprecia la amplificación del voltaje dc.

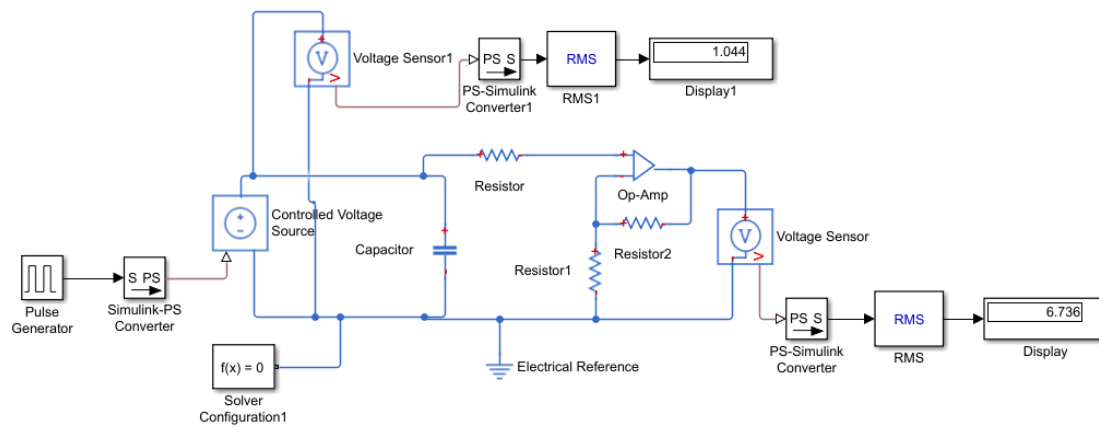


Ilustración 2. 6 Simulación del circuito amplificador. Fuente: Autor simulink

El circuito esquemático realizado se puede ver en el **Anexo 10**.

2.2 Descripción del algoritmo principal del sistema de control a lazo abierto

A partir de este punto del libro cada algoritmo y código descrito se desarrolló en el entorno de programación CODE COMPOSER STUDIO con lenguaje CCS el cual está basado en C, C++ y assembler donde los registros se organizan en estructuras de campos de bits, ya que brinda una gran flexibilidad en cuanto optimización de código y tiempos de ejecución. En el **anexo 1** se muestra una guía de inicio rápido para empezar a codificar en este entorno.

En la siguiente ilustración se puede apreciar el diagrama en bloques de la programación empleada.

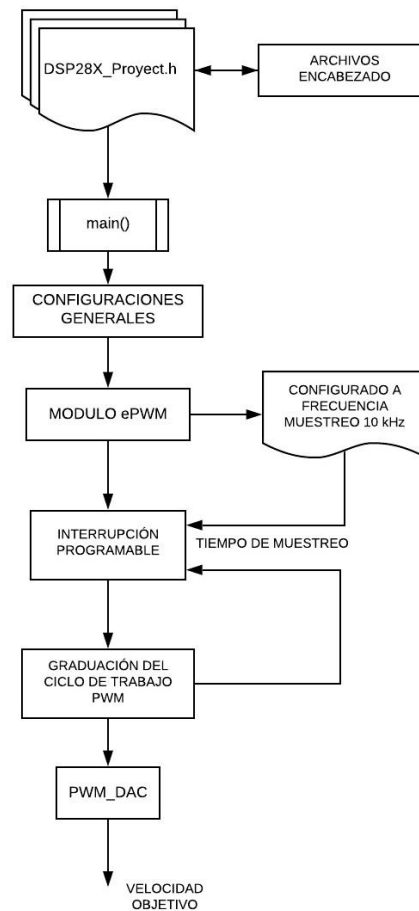


Ilustración 2. 7 Diagrama de bloques del funcionamiento a lazo abierto

En la anterior ilustración se destaca el módulo ePWM con el cual el sistema de control esta calibrado a partir de una frecuencia de muestreo estable a 10KHz lo que es de gran importancia ya que proporciona un periodo estable, el cual es muy importante para los bloques de estimación de flujo estimación de velocidad y el controlador a implementar en capítulos más adelante.

Configuración del módulo ePWM

El principio fundamental de funcionamiento del módulo ePWM utilizado en el desarrollo de este proyecto como una salida PWM_DAC para el control del voltaje de referencia al cual representa una velocidad de la máquina de inducción y el control de los tiempos de ejecución de todos los algoritmos planteados se basa en la generación de una forma de onda triangular apreciable en la ilustración[6].

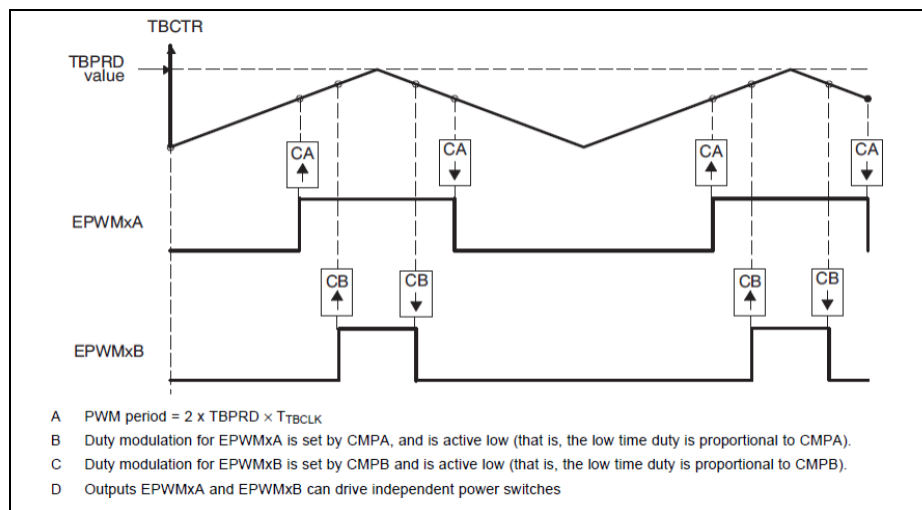


Ilustración 2. 8 Modulo e PWM

Donde un registro de conteo de 16 bits va aumentando de valor cada ciclo de reloj el cual estos previamente configurados para fines de desarrollo del proyecto se dejó a 90Mhz aumenta hasta el valor calculado a partir de la siguiente ecuación [6].

$$TBPRD = \frac{1}{2} * \frac{f_{sysclk}}{f_{pwm} * CLKDIV * HSPclkdiv}$$

- f_{sysclk} : frecuencia máxima del reloj del DSP (90 MHz).
- f_{pwm} : frecuencia de la onda PWM =10kHz.
- $CLKDIV$: parámetro para dividir la frecuencia del reloj=1.
- $HSPclkdiv$: parámetro para dividir la frecuencia del reloj=1.

En el proyecto este valor quedo en 4500 lo que dice que la onda aumentara de valor hasta este valor y luego va a decrecer de hasta llegar a cero controlando así el ciclo de trabajo. Para simular una salida analógica se tiene que variar los puntos de comparación mostrados en la anterior ilustración para así tener una señal analógica para el control de la velocidad. Continuación se muestra el diagrama en bloque de la configuración realizada a este módulo[6].

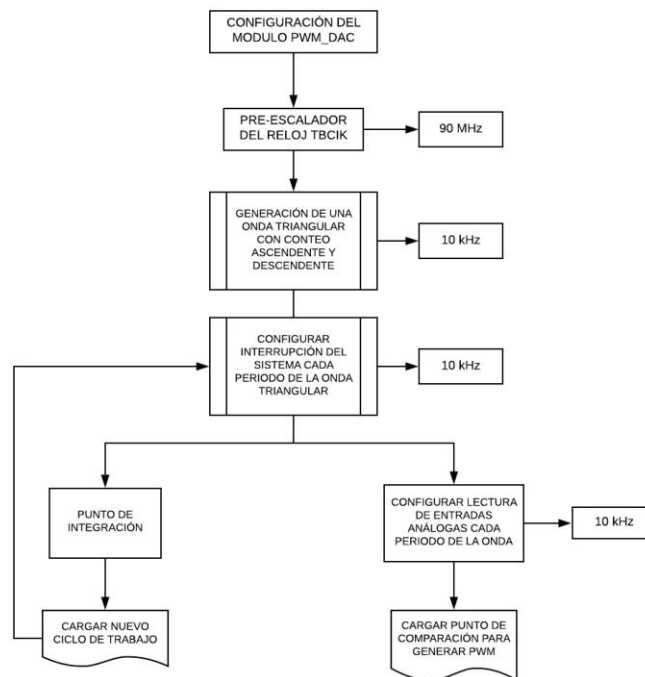


Ilustración 2. 9 Configuración salida PWM_DAC

La salida física de la señal PWM es el GPIO 40 del dispositivo el cual tiene está provisto de un filtro pasa bajas con el fin de obtener una señal de DC variable estable en la siguiente ilustración 2.1 se aprecia esto. El valor de la señal del ciclo de trabajo de la señal PWM de salida se va a ir actualizando cada 10kHz teniendo un tiempo en alto de 0% a 100%.

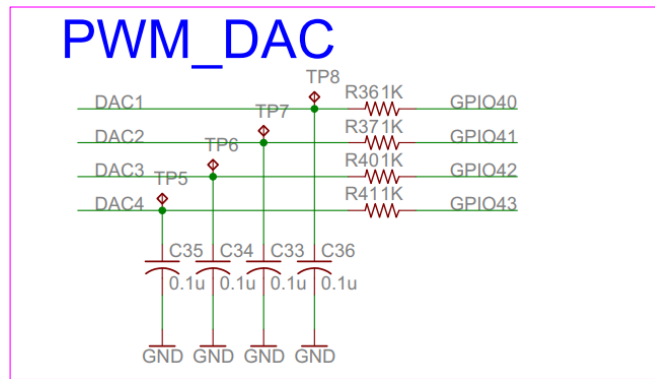


Ilustración 2. 10 PWM_DAC

3. IMPLEMENTACION DE LA TECNICA DE CONTROL

Para la implementación de la técnica de control de velocidad sensorless expuesta en el capítulo 1 se implementa el siguiente diagrama en bloques ilustración 3.1. donde se puede ver más a detalle el sistema de control empleado para el desarrollo de este proyecto en donde su etapa de control a lazo abierto ya se explicó en el capítulo anterior la cual incluye el módulo ePWM y las características generales de operación del sistema de control[23].

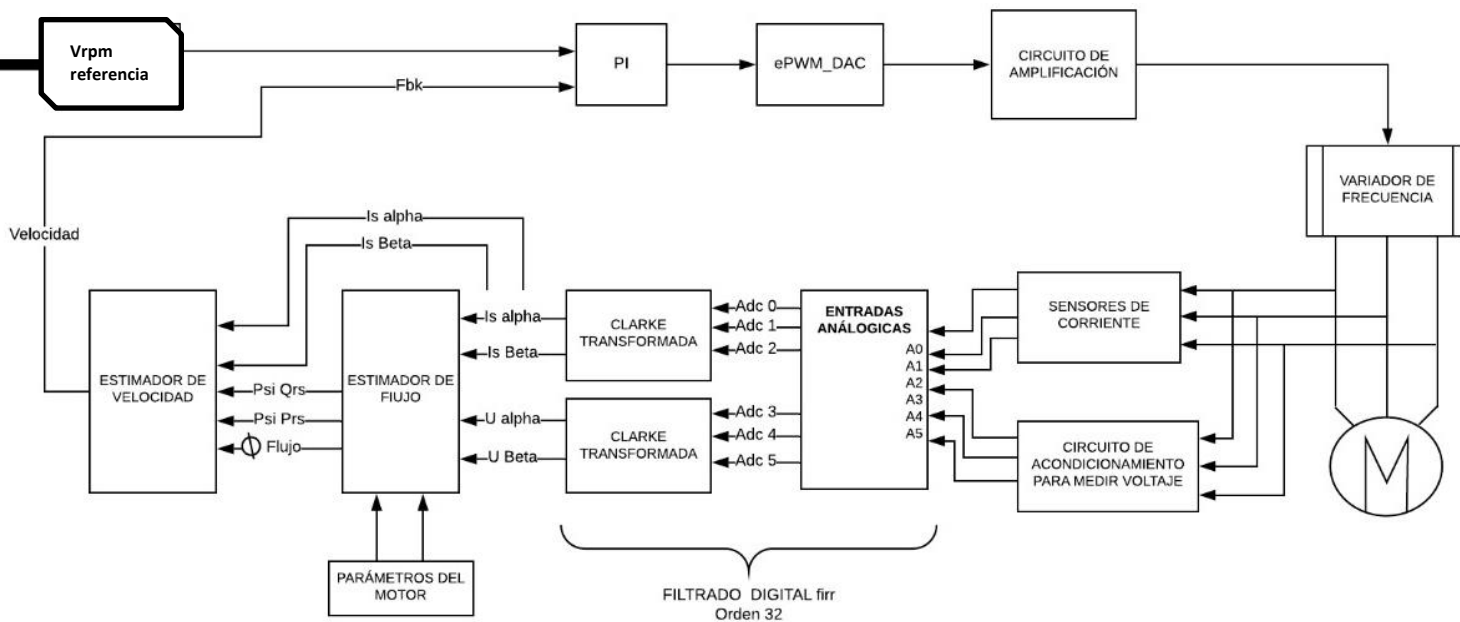


Ilustración 3. 1 Diagrama general del sistema de control. Fuente: autor

Descripción del sistema a nivel general:

- En primer lugar se debe adquirir las señales de voltaje y corriente la cuales deben normalizarse en valores en P.U. de acuerdo a las bases.
- Luego se realiza un filtrado digital con un filtro FIR
- Se aplican las transformaciones de Clark a la corriente y voltaje
- Se estima el flujo con las componentes de voltaje y corriente.

- Se estima la velocidad.
- Se envía al controlador de velocidad el cual calcula el error y da una respuesta.
- El controlador cambia el ciclo de trabajo del PWM_DAC para controla la referencia de voltaje
- La señal proveniente del PWM_DC se amplifica y se envía al variador de frecuencia.

3.1 Adquisición y normalización de las señales de voltaje y corriente

Para adquirir el voltaje y la corriente se debe emplear distintos métodos ya que son variables diferentes pero muy importantes para el control sensorless planteado en el proyecto.

Adquisición y normalización de corriente

Para adquirir corriente se utiliza un sensor presente en el grupo de investigación de sistemas energéticos el cual funciona en base al efecto hall el cual produce un voltaje sensando un campo magnético presente como se ve en la siguiente ilustración 3.2.

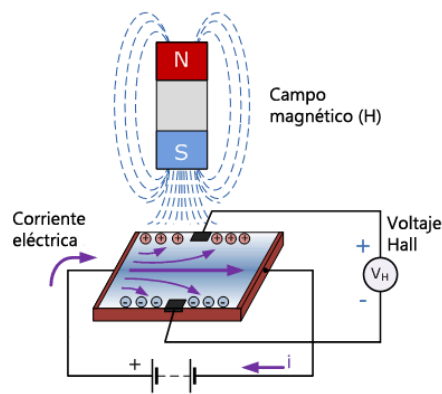


Ilustración 3. 2 Efecto hall .Fuente:

Dicho sensor mide y normaliza la magnitud de la corriente a niveles permitidos por las entradas analógicas del para su posterior tratamiento con el filtro digital, en la siguiente ilustración 3.3 de aprecia el sensor empleado.

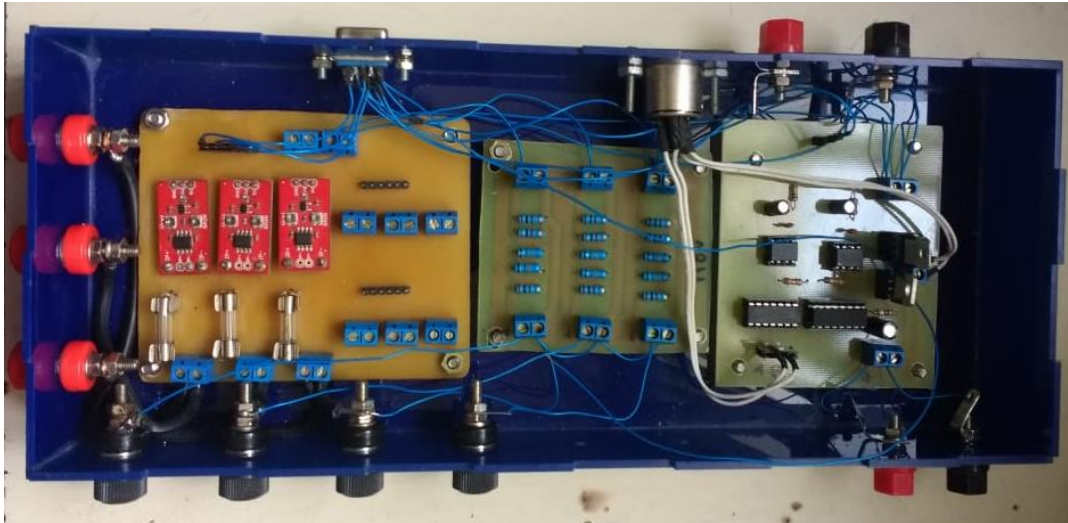


Ilustración 3. 3 Sensor de corriente. Fuente: autor

Adquisición y normalización de voltaje

Para adquirir el voltaje se plante a un circuito de reducción tomando como base un banco de transformadores presentes en el grupo de investigación los cuales se aprecian en la siguiente ilustración 3.4.



Ilustración 3. 4 Banco de transformadores Fuente: autor

Después de la reducción de la señal a un nivel de $\pm 10V$ por el banco de transformadores reductores se diseña un circuito de normalización en base al amplificador operacional LM741 el cual es un integrado de un único amplificador

operacional lo cual es ideal ya que son voltajes a reducir son trifásicos, además de sumarle una señal de offset a la señal de voltaje pues las entradas analógicas solo admiten valores positivos. La configuración del amplificador como sumador no inversor se aprecia en la siguiente ilustración 3.5.

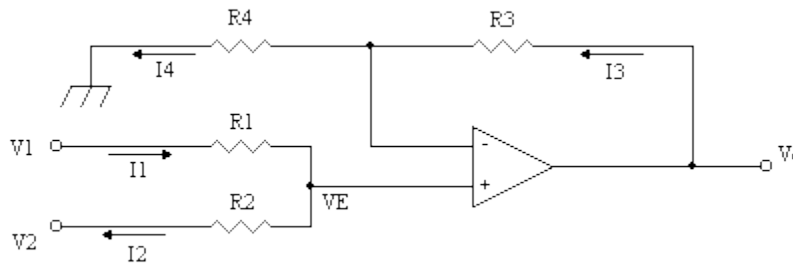


Ilustración 3.5 Amplificador operacional sumador no inversor

El circuito se simulo en Matlab como se puede apreciar en la siguiente ilustración 3.6; El circuito esquemático implementado se puede apreciar en el **Anexo 10**.

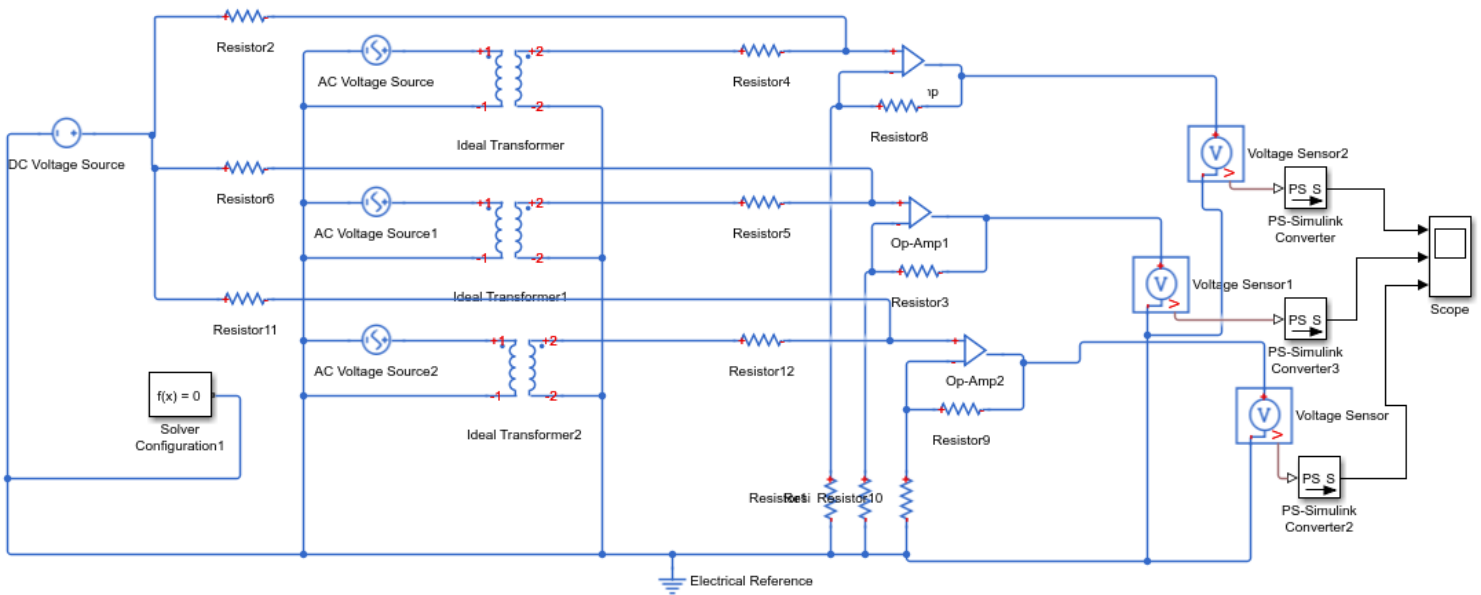


Ilustración 3.6 Simulación acondicionamiento de voltaje

Una vez hecho el procedimiento de acople se procede a realizar la configuración de esta etapa se procede a configurar la entrada analógica del DPS de la ADC0-ADC5

3.2 Muestreo y adquisición de las señales de voltaje y corriente del circuito de acondicionamiento a través del ADC0-ADC5

El módulo ADC programado es de 12 bits. Con 2 circuitos de muestreo y retención (S / H), un único núcleo de conversión vistos en la ilustración 4.2 junto con los registros correspondientes a su configuración. Los circuitos de muestreo y retención se pueden muestrear de forma simultánea o secuencial en este caso solo se muestrea de forma secuencial el módulo ADC cuenta con 16 canales de estrada analógica del cual solo se usa el ADC0. El principio básico de para configurar la operación de ADC se centra en las configuraciones de conversiones individuales, denominadas **SOC** o inicio de conversiones. A continuación se enumeran las características destacables para su correcta configuración[23]:

- Núcleo ADC de 12 bits con doble muestreo y retención incorporados (S / H)
- Muestreo simultáneo o modos de muestreo secuencial.
- Entrada analógica de rango completo: 0 V a 3,3 V fija
- 16 canales.
- 16 SOC, configurables para disparador, ventana de muestra y canal
- 16 registros de resultados para almacenar valores de conversión.
- Múltiples fuentes de disparo
 - S / W - inicio inmediato del software
 - **ePWM 1-8** el dual se configura para que sea la fuente de disparo del ADC lo que significa que cada 10KHz se realiza una nueva conversiopl.
 - XINT2
 - CPU Timers 0/1/2
 - ADCINT1 / 2

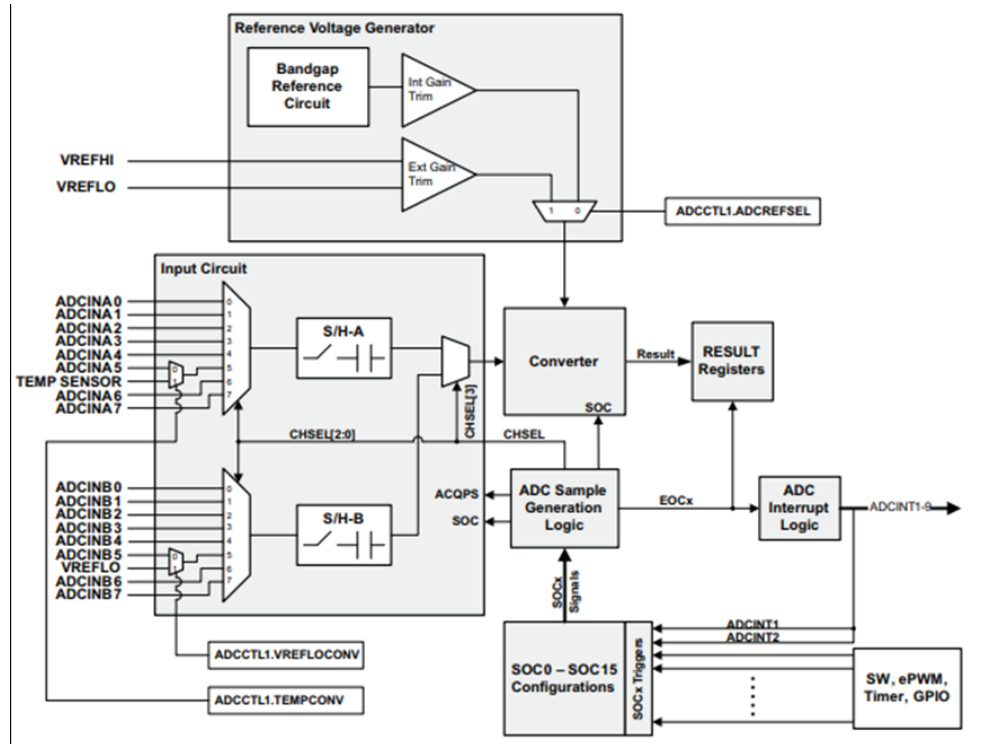


Ilustración 3.7 Diagrama de bloques ADC. Fuente: [21]

▪ Principio de Operación SOC

En general hay que configurar en el SOC tres configuraciones generales: la fuente de activación que inicia la conversión, el canal a convertir y el tamaño de la ventana de adquisición (muestra). La fuente de activación para SOC0-SOC5 se configura mediante una combinación del campo TRIGSEL en el registro ADCSOC0CTL y los bits apropiados en el registro ADCINTSOCSEL1. El canal y el tamaño de la ventana de muestra para SOC0 se configuran con los campos CHSEL y ACQPS del registro ADCSOC0CTL en la ilustración 3.8 se ejemplifica lo antes descrito para analizar que registros son relevantes para el desarrollo[23].

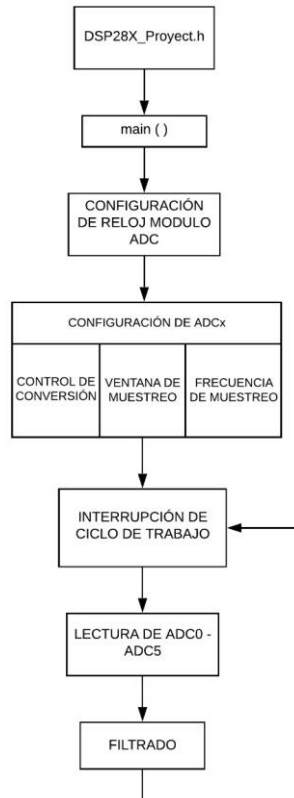


Ilustración 3. 9 Algoritmo del ADC. Fuente: [21]

La lectura de las señales de corriente y voltaje se aprecian en el capítulo 4 donde se podrán apreciar las formas de onda tanto del voltaje como de la corriente.

Filtrado digital de las señales.

- Filtro de respuesta al impulso finito

Los filtros digitales de respuesta impulsional finita (Finite Impulse Response) se basan en obtener la salida a partir, exclusivamente, de las entradas actuales y anteriores. Así, para un filtro de longitud $N[24]$:

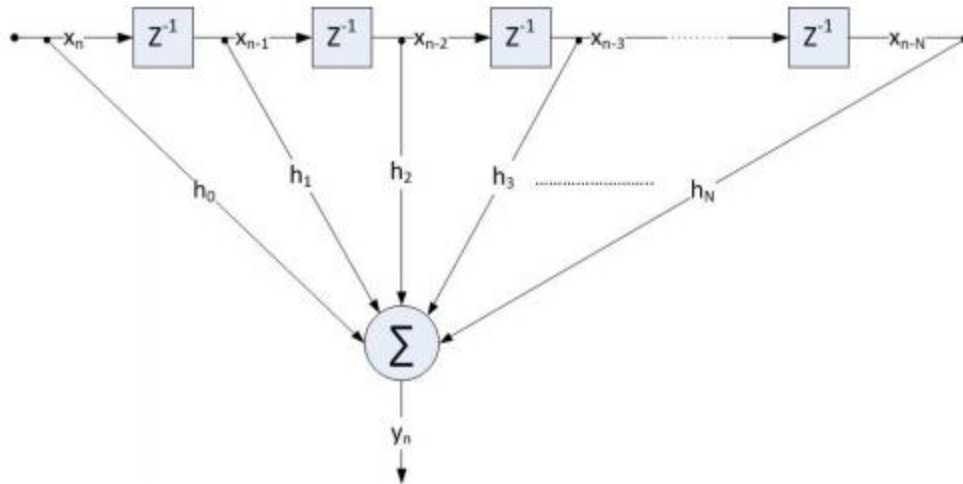


Ilustración 3. 10 Diagrama en bloques filtro fir. Fuente: [25]

De donde se puede deducir la siguiente ecuación que a nivel de implementación computacional es relativamente sencilla su implementación, la Texas Instruments [25] proporciona toda la información técnica sobre cómo implementar el filtro y la forma en cómo se deben calcular sus coeficientes que como se puede apreciar en la anterior ilustración son los encargados de dar la respuesta a la frecuencia específica de corte a la que se diseñan.

$$H(z) = \sum_{k=0}^N h_k \times z^{-k}$$

Los coeficientes FIR pueden ser generados por la herramienta de diseño y análisis de filtros MATLAB (FDATool), en la cual se le agrega el tipo de filtro el cual es pasa bajas, el orden del filtro el cual se establece en 196 y la frecuencia de muestreo la cual es de 10KHz con una frecuencia de paso de 120 y frecuencia de corte de 250Hz.

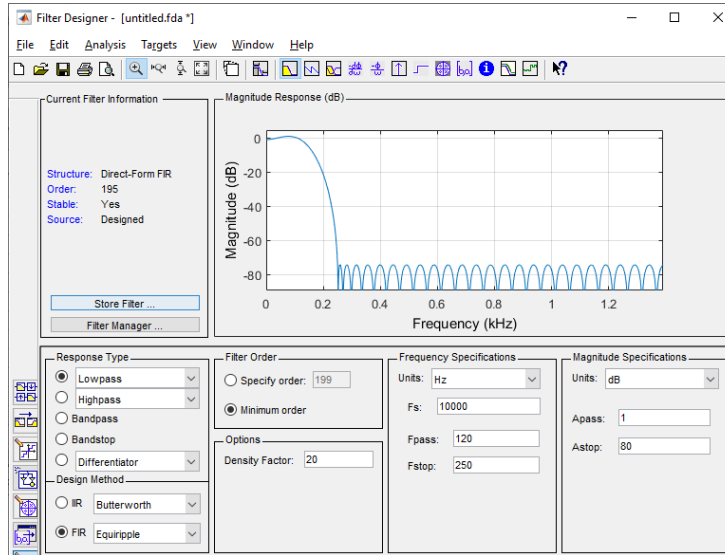


Ilustración 3. 11 Cálculo de coeficiente. Fuente: Autor matlab

Para exportar los archivos de declarar como con signo de 16 bits como se muestra en la siguiente ilustración

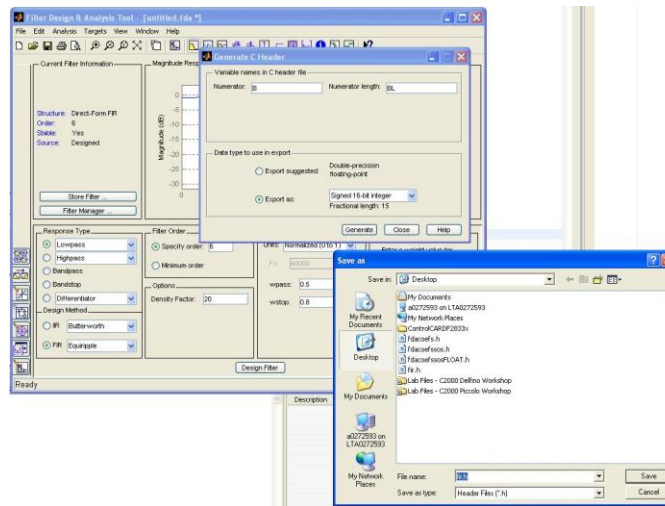


Ilustración 3. 12 Exportación de coeficientes. Fuente: Autor matlab

La implementación a nivel de algoritmo se basó de acuerdo a la estructura proporcionada por la Texas instruments definida en el siguiente bloque donde se le

envía la lectura de una entrada análoga y la estructura del filtro da una respuesta, como se analiza en la formula discreta del filtro se debe dejar un tiempo de captura de muestras para que el filtro funcione correctamente el cual se fije en 1/10kHz.



Ilustración 3. 13 Bloque de filtro

3.3 Transformada de Clark implementación

La tranformada de clar a nivel discreto simplemente resulta de las ecuaciones descritas en el capítulo 1 escribiéndolas en código c de la siguiente manera[16].

```
//=====
#define CLARKE_MACRO(v)
v.Alpha = v.As;
v.Beta = _IQmpy((v.As + _IQmpy2(v.Bs)),_IQ(ONEbySQRT3));
```

Donde se crea una estructura para corriente y voltaje de manera independiente.

3.4 Implementación de estimador de flujo

Reescribiendo las ecuaciones descritas en el capitulo 1 a nivel discreto se obtiene:

La dinámica de flujo orientado del rotor en (11) se discretiza mediante el uso de aproximación hacia atrás de la siguiente manera [16]:

$$\frac{\psi_{dr}^{e,i}(k) - \psi_{dr}^{e,i}(k-1)}{T} = \frac{L_m}{\tau_r} i_{ds}^e(k) - \frac{1}{\tau_r} \psi_{dr}^{e,i}(k) \quad 44$$

Donde T=10kHz es el período de muestreo (seg). Reordenando (44), luego da

$$\psi_{dr}^{e,i}(k) = \left(\frac{\tau_r}{\tau_r + T} \right) \psi_{dr}^{e,i}(k-1) + \left(\frac{L_m T}{\tau_r + T} \right) i_{ds}^e(k) \quad 45$$

A continuación, los enlaces de flujo del estator en (17) - (18) se discretizan mediante el uso de aproximación trapezoidal (o tustin) como

$$\psi_{ds}^{s,v}(k) = \psi_{ds}^{s,v}(k-1) + \frac{T}{2} (e_{ds}^s(k) + e_{ds}^s(k-1)) \quad 46$$

$$\psi_{qs}^{s,v}(k) = \psi_{qs}^{s,v}(k-1) + \frac{T}{2} (e_{qs}^s(k) + e_{qs}^s(k-1)) \quad 47$$

Donde los fem's de atrás se calculan como

$$e_{ds}^s(k) = u_{ds}^s(k) - i_{ds}^s(k)R_s - u_{comp,ds}(k) \quad 48$$

$$e_{qs}^s(k) = u_{qs}^s(k) - i_{qs}^s(k)R_s - u_{comp,q,s}(k) \quad 49$$

Del mismo modo, las leyes de control de PI en (19) - (20) también se discretizan mediante el uso de aproximación trapezoidal como

$$u_{comp,ds}(k) = K_p (\psi_{ds}^{s,v}(k) - \psi_{ds}^{s,i}(k)) + u_{comp,ds,i}(k-1) \quad 50$$

$$u_{comp,q,s}(k) = K_p (\psi_{qs}^{s,v}(k) - \psi_{qs}^{s,i}(k)) + u_{comp,q,s,i}(k-1) \quad 51$$

Donde los términos integrales acumulados son como

$$\begin{aligned} u_{comp,ds,i}(k) &= u_{comp,ds,i}(k-1) + \frac{K_p T}{T_I} (\psi_{ds}^{s,v}(k) - \psi_{ds}^{s,i}(k)) \\ &= u_{comp,ds,i}(k-1) + K_p K_I (\psi_{ds}^{s,v}(k) - \psi_{ds}^{s,i}(k)) \end{aligned} \quad 52$$

$$\begin{aligned} u_{\text{comp,qs,i}}(k) &= u_{\text{comp,qs,i}}(k-1) + \frac{K_p T}{T_I} (\psi_{\text{qs}}^{s,v}(k) - \psi_{\text{qs}}^{s,i}(k)) \\ &= u_{\text{comp,qs,i}}(k-1) + K_p K_I (\psi_{\text{qs}}^{s,v}(k) - \psi_{\text{qs}}^{s,i}(k)) \end{aligned}$$

Donde $K_I = \frac{T}{T_I}$

Tiempo discreto y por unidad: ahora todas las ecuaciones se normalizan en la unidad por las cantidades base especificada. En primer lugar, el enlace de flujo del rotor en el modelo actual (45) se normaliza dividiendo el enlace de flujo base como

$$\psi_{\text{dr,pu}}^{e,i}(k) = \left(\frac{\tau_r}{\tau_r + T} \right) \psi_{\text{dr,pu}}^{e,i}(k-1) + \left(\frac{T}{\tau_r + T} \right) i_{\text{ds,pu}}^e(k) \quad \text{pu} \quad 53$$

Donde $\psi_b = L_m I_b$ es el enlace de flujo base (volt.seg) e I_b es la corriente base (amperio).

A continuación, los enlaces de flujo del estator en el modelo actual (15) - (16) se normalizan de manera similar dividiendo el enlace de flujo base como

$$\psi_{\text{ds,pu}}^{s,i}(k) = \left(\frac{L_s L_r - L_m^2}{L_r L_m} \right) i_{\text{ds,pu}}^s(k) + \frac{L_m}{L_r} \psi_{\text{dr,pu}}^{s,i}(k) \quad \text{pu} \quad 54$$

$$\psi_{\text{qs,pu}}^{s,i}(k) = \left(\frac{L_s L_r - L_m^2}{L_r L_m} \right) i_{\text{qs,pu}}^s(k) + \frac{L_m}{L_r} \psi_{\text{qr,pu}}^{s,i}(k) \quad \text{pu} \quad 55$$

Luego, la fem posterior en (48) - (49) se normaliza dividiendo el voltaje de fase base V_b

$$e_{\text{ds,pu}}^s(k) = u_{\text{ds,pu}}^s(k) - \frac{I_b R_s}{V_b} i_{\text{ds,pu}}^s(k) - u_{\text{comp,ds,pu}}(k) \quad \text{pu} \quad 56$$

$$e_{\text{qs,pu}}^s(k) = u_{\text{qs,pu}}^s(k) - \frac{I_b R_s}{V_b} i_{\text{qs,pu}}^s(k) - u_{\text{comp,qs,pu}}(k) \quad \text{pu} \quad 57$$

A continuación, los enlaces de flujo del estator en el modelo de voltaje (45) - (46) se dividen por el enlace de flujo base

$$\psi_{ds,v}^{s,v}(k) = \psi_{ds,v}^{s,v}(k-1) + \frac{V_b T}{L_m I_b} \left(\frac{e_{ds,pu}^s(k) + e_{ds,pu}^s(k-1)}{2} \right) \text{ pu} \quad 58$$

$$\psi_{qs,v}^{s,v}(k) = \psi_{qs,v}^{s,v}(k-1) + \frac{V_b T}{L_m I_b} \left(\frac{e_{qs,pu}^s(k) + e_{qs,pu}^s(k-1)}{2} \right) \text{ pu} \quad 59$$

Similar a (54) - (55), los enlaces de flujo de rotor normalizados en el modelo de voltaje son.

$$\psi_{dr,pu}^{s,v}(k) = - \left(\frac{L_s L_r - L_m^2}{L_m L_m} \right) i_{ds,pu}^s(k) + \frac{L_r}{L_m} \psi_{ds,pu}^{s,v}(k) \text{ pu} \quad 60$$

$$\psi_{qr,pu}^{s,v}(k) = - \left(\frac{L_s L_r - L_m^2}{L_m L_m} \right) i_{qs,pu}^s(k) + \frac{L_r}{L_m} \psi_{qs,pu}^{s,v}(k) \text{ pu} \quad 61$$

Finalmente las ecuaciones discreta en su respectiva base que codificación en su respectiva estructura de programación son:

Modelo de corriente: enlace de flujo del rotor en marco de referencia giratorio sincrónicamente ($\omega = \omega_{\psi_r}$)

$$\psi_{dr,pu}^{c,i}(k) = K_1 \psi_{dr,pu}^{c,i}(k-1) + K_2 i_{ds,pu}^c(k) \text{ pu} \quad 62$$

Donde $K_1 = \frac{\tau_r}{\tau_r + T}$, y $K_2 = \frac{T}{\tau_r + T}$

Modelo de corriente: enlaces de flujo del rotor en el marco de referencia estacionario ($\omega = 0$)

$$\psi_{ds,pu}^{s,i}(k) = K_4 i_{ds,pu}^s(k) + K_3 \psi_{dr,pu}^{s,i}(k) \text{ pu} \quad 63$$

$$\psi_{qs,pu}^{s,i}(k) = K_4 i_{qs,pu}^s(k) + K_3 \psi_{qr,pu}^{s,i}(k) \quad pu \quad 64$$

Donde $K_3 = \frac{L_m}{L_r}$, y $K_4 = \frac{L_s L_r - L_m^2}{L_r L_m}$.

Modelo de voltaje: en el marco de referencia estacionario ($\omega = 0$)

$$e_{ds,pu}^s(k) = u_{ds,pu}^s(k) - K_5 i_{ds,pu}^s(k) - u_{comp,ds,pu}(k) \quad pu \quad 65$$

$$e_{qs,pu}^s(k) = u_{qs,pu}^s(k) - K_5 i_{qs,pu}^s(k) - u_{comp,qs,pu}(k) \quad pu \quad 66$$

Donde $K_5 = \frac{I_b R_s}{V_b}$

Modelo de voltaje: enlaces de flujo del estator en el marco de referencia estacionario ($\omega = 0$)

$$\psi_{ds,pu}^{s,v}(k) = \psi_{ds,pu}^{s,v}(k-1) + K_6 \left(\frac{e_{ds,pu}^s(k) + e_{ds,pu}^s(k-1)}{2} \right) \quad pu \quad 67$$

$$\psi_{qs,pu}^{s,v}(k) = \psi_{qs,pu}^{s,v}(k-1) + K_6 \left(\frac{e_{qs,pu}^s(k) + e_{qs,pu}^s(k-1)}{2} \right) \quad pu \quad 68$$

Donde $K_6 = \frac{V_b T}{L_m I_b}$

Modelo de voltaje: enlaces de flujo del rotor en el marco de referencia estacionario ($\omega = 0$)

$$\psi_{dr,pu}^{s,v}(k) = -K_8 i_{ds,pu}^s(k) + K_7 \psi_{ds,pu}^{s,v}(k) \quad pu \quad 69$$

$$\psi_{qr,pu}^{s,v}(k) = -K_8 i_{qs,pu}^s(k) + K_7 \psi_{qs,pu}^{s,v}(k) \quad pu \quad 70$$

Donde $K_7 = \frac{L_r}{L_m}$, y $K_8 = \frac{L_s L_r - L_m^2}{L_m L_m}$

Modelo de voltaje: Angulo de flujo del rotor

$$\theta_{\psi_{r,pu}}(k) = \frac{1}{2\pi} \tan^{-1} \left(\frac{\psi_{qr,pu}^{s,v}(k)}{\psi_{dr,pu}^{s,v}(k)} \right) \quad pu$$

Los parámetros requeridos para implementación del código son

Los parámetros de la máquina:

- resistencia del estator (Rs)
- resistencia del rotor (Rr)
- inductancia de fuga del estator (Lsl)
- inductancia de fuga del rotor (Lrl)
- inductancia magnetizante (Lm)

Las cantidades basadas:

- corriente base (Ib)
- tensión de fase base (Vb)

El período de muestreo:

- período de muestreo (T)

El bloque que calcula el flujo del motor en términos simple se puede ver a continuación[16].

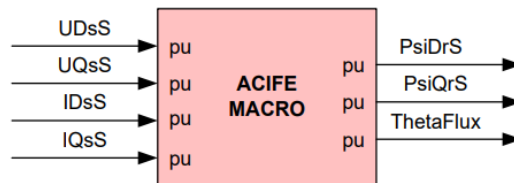


Ilustración 3. 14 Bloque de estimador de flujo. Fuente:[16]

El código en lenguaje C se presenta a continuación

```
#define ACIFE_MACRO(v)
/* Calculate Sine and Cosine terms for Park/IPark transformations */
v.Sine = _IQsinPU(v.ThetaFlux);
```

```

v.Cosine = _IQcosPU(v.ThetaFlux);
/* Park transformation on the measured stator current*/
v.IDsE = _IQmpy(v.IQsS,v.Sine);
v.IDsE += _IQmpy(v.IDsS,v.Cosine);
/* The current model section (Classical Rotor Flux Vector Control
Equation)*/ \
v.FluxDrE = _IQmpy(v.K1,v.FluxDrE) + _IQmpy(v.K2,v.IDsE);
/* Inverse park transformation on the rotor flux from the current model*
v.FluxDrS = _IQmpy(v.FluxDrE,v.Cosine);
v.FluxQrS = _IQmpy(v.FluxDrE,v.Sine);
/* Compute the stator flux based on the rotor flux from current model*/
v.FluxDsS = _IQmpy(v.K3,v.FluxDrS) + _IQmpy(v.K4,v.IDsS);
v.FluxQsS = _IQmpy(v.K3,v.FluxQrS) + _IQmpy(v.K4,v.IQsS);
/* Conventional PI controller section */
v.Error = v.PsiDsS - v.FluxDsS;
v.UCompDsS = _IQmpy(v.Kp,v.Error) + v.UiDsS;
v.UiDsS = _IQmpy(v.Kp,_IQmpy(v.Ki,v.Error)) + v.UiDsS;
v.Error = v.PsiQsS - v.FluxQsS;
v.UCompQsS = _IQmpy(v.Kp,v.Error) + v.UiQsS;
v.UiQsS = _IQmpy(v.Kp,_IQmpy(v.Ki,v.Error)) + v.UiQsS;
/* Compute the estimated stator flux based on the integral of back emf*/
v.OldEmf = v.EmfDsS;
v.EmfDsS = v.UDsS - v.UCompDsS - _IQmpy(v.K5,v.IDsS);
v.PsiDsS = v.PsiDsS + _IQdiv2(_IQmpy(v.K6,(v.EmfDsS + v.OldEmf)));
v.OldEmf = v.EmfQsS;
v.EmfQsS = v.UQsS - v.UCompQsS - _IQmpy(v.K5,v.IQsS);
v.PsiQsS = v.PsiQsS + _IQdiv2(_IQmpy(v.K6,(v.EmfQsS + v.OldEmf)));
/* Estimate the rotor flux based on stator flux from the integral of back
emf*/
v.PsiDrS = _IQmpy(v.K7,v.PsiDsS) - _IQmpy(v.K8,v.IDsS);
v.PsiQrS = _IQmpy(v.K7,v.PsiQsS) - _IQmpy(v.K8,v.IQsS);
/* Compute the rotor flux angle*/
v.ThetaFlux = _IQatan2PU(v.PsiQrS,v.PsiDrS);
#endif // __ACI_FE_H__

```

3.5 Implementación de estimador de velocidad

El paso a seguir es implementar el estimador de velocidad, para realizar la implementación computacional es necesario analizar las formulas mencionadas en el capítulo 1 pero en tiempo discreto para que sea posible implementarla a nivel de código[16].

A nivel de implementación de algoritmos del estimador de velocidad y analizando la ecuación descrita a continuación se obtiene analiza que para obtener la velocidad W_{rpu} [16].

$$\omega_{r,pu} = \omega_{c,pu} - K_1 \left(\frac{\lambda_{dr,pu}^s i_{qs,pu}^s - \lambda_{qr,pu}^s i_{ds,pu}^s}{(\lambda_{r,pu}^s)^2} \right) \quad pu \quad 71$$

Primero se debe calcular la velocidad W_{epu} descrita en la ecuación mostrada a continuación[16].

$$\omega_{e,pu} = \frac{1}{2\pi f_b} \frac{d\theta_{\lambda_r}}{dt} = \frac{1}{f_b} \frac{d\theta_{\lambda_r,pu}}{dt} \quad pu \quad 72$$

Discretizando la ecuación anterior y utilizando una aproximación hacia a tras

$$\omega_{e,pu}(k) = \frac{1}{f_b} \left(\frac{\theta_{\lambda_r,pu}(k) - \theta_{\lambda_r,pu}(k-1)}{T} \right) \quad pu \quad 73$$

Donde T es el período de muestreo (seg). Equivalentemente, otra forma es

$$\omega_{e,pu}(k) = K_2 (\theta_{\lambda_r,pu}(k) - \theta_{\lambda_r,pu}(k-1)) \quad pu \quad 74$$

Donde $K_2 = \frac{1}{f_b T}$

En la práctica, las formas de onda típicas del ángulo de flujo del rotor $\theta_{\lambda_r,pu}$, en ambas direcciones se pueden ver ilustración Para cuidar la discontinuidad del ángulo de 360 a 0 (CCW) o de 0o a 360 (CW), el diferenciador se opera simplemente dentro del rango diferenciable como se ve en esta ilustración. Este rango diferenciable no pierde significativamente la información para calcular la velocidad estimada[16].

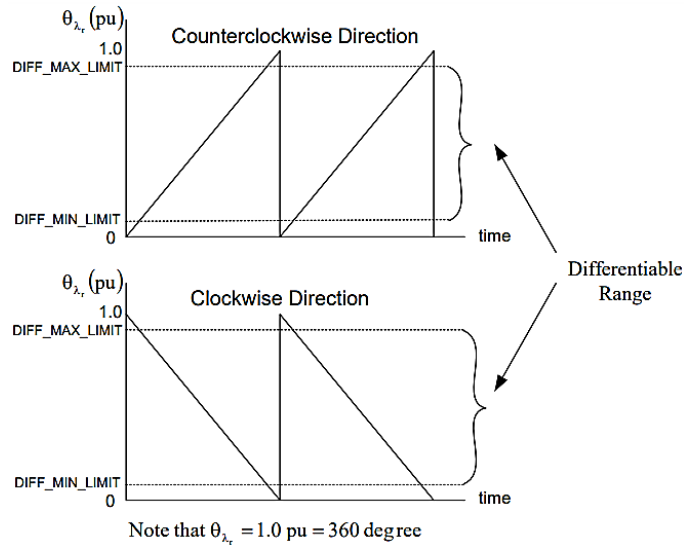


Ilustración 2. 11 Angulo de flujo. Fuente: [16]

Además, la velocidad síncrona en (74) es necesaria ser filtrada por u filtro de paso bajo para reducir el ruido de amplificación generado por el diferenciador puro en (74). Se utiliza el filtro de paso bajo de primer orden simple, luego la velocidad síncrona real que se utilizará es la salida del filtro de paso bajo, $\hat{\omega}_{e,pu}$, visto en la siguiente ecuación. La ecuación de tiempo continuo del filtro de paso bajo del primer orden es como[16].

$$\frac{d\hat{\omega}_{e,pu}}{dt} = \frac{1}{\tau_c} (\omega_{e,pu} - \hat{\omega}_{e,pu}) \quad pu \quad 75$$

Donde $\tau_c = \frac{1}{2\pi f_c}$ es la constante de tiempo de filtro de paso bajo (seg), y f_c es la frecuencia de corte (Hz). Usando la aproximación hacia atrás, (75) finalmente se convierte en tiempo discreto

$$\hat{\omega}_{e,pu}(k) = K_3 \hat{\omega}_{e,pu}(k-1) + K_4 \omega_{e,pu}(k) \quad pu \quad 76$$

Donde $K_3 = \frac{\tau_c}{\tau_c + T}$ y $K_4 = \frac{T}{\tau_c + T}$

De hecho, solo tres ecuaciones (71), (74) y (76) se emplean principalmente para calcular la velocidad estimada en por unidad. Los parámetros requeridos se resumen a continuación[16].

Los parámetros de la máquina:

- número de polos (p)
- resistencia del rotor (R_r)
- inductancia de fuga del rotor (L_{rl})
- inductancia magnetizante (L_m)

Las cantidades base:

- corriente base (I_b)
- base de velocidad angular angular (W_b)

El período de muestreo:

- período de muestreo (T) = 10kHz

Filtro de paso bajo:

- Frecuencia de corte (f_c)

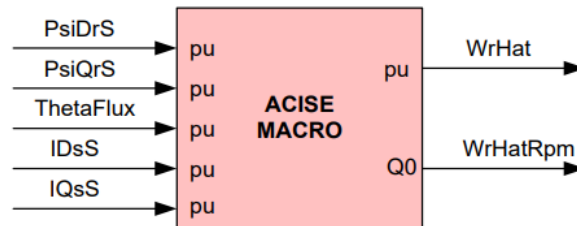


Ilustración 3. 15 Bloque de estimador de velocidad. Fuente:[16]

El código en lenguaje C se presenta a continuación

```
#define ACISE_MACRO(v)
/* Slip computation */
v.SquaredPsi = _IQmpy(v.PsiDrS,v.PsiDrS)+_IQmpy(v.PsiQrS,v.PsiQrS);
v.WSlip= _IQmpy(v.K1,(_IQmpy(v.PsiDrS,v.IQsS) -
_IQmpy(v.PsiQrS,v.IDsS)));
v.WSlip= _IQdiv(v.WSlip,v.SquaredPsi);
/* Synchronous speed computation */
```

```

if ((v.ThetaFlux < DIFF_MAX_LIMIT)&(v.ThetaFlux > DIFF_MIN_LIMIT))
/* Q21 = Q21*(GLOBAL_Q-GLOBAL_Q)*/
    v.WSyn = _IQmpy(v.K2,(v.ThetaFlux - v.OldThetaFlux));
else v.WSyn = v.WPsi;
/* low-pass filter, Q21 = GLOBAL_Q*Q21 + GLOBAL_Q*Q21 */
v.WPsi = _IQmpy(v.K3,v.WPsi) + _IQmpy(v.K4,v.WSyn);
/* Q21 = Q21 - GLOBAL_Q */
v.OldThetaFlux = v.ThetaFlux;
v.WrHat = v.WPsi - _IQtoIQ21(v.WSlip);
/* Limit the estimated speed between -1 and 1 per-unit */
v.WrHat=_IQsat(v.WrHat,_IQ21(1.5),_IQ21(0));
v.WrHat = _IQ21toIQ(v.WrHat);
/* Q0 = Q0*GLOBAL_Q => _IQXmpy(), X = GLOBAL_Q */
v.WrHatRpm = _IQmpy(v.BaseRpm,v.WrHat);
#endif // __ACI_SE_H__
  
```

Descripción del controlador del sistema de control de velocidad

▪ **Controladores PI lineales**

Las aplicaciones como la de control de un motor de inducción que no requieren acción derivada, o que son más sensibles a la eficiencia del ciclo, pueden ser mejor atendidas por la estructura más simple del controlador PI descrita en la siguiente sección por su simplicidad en implementación y su poco tiempo de respuesta[29].

$$u(t) = k_p e(t) + k_i \int_{-\infty}^t e(\tau) d\tau \quad (4.2)$$

Su diagrama en bloque se muestra en la siguiente ilustración 3.16 la cual solo cuenta con la suma de su canal proporcional e integral.

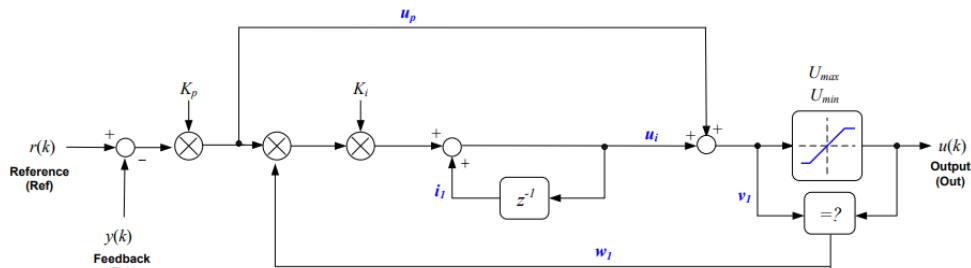


Ilustración 3. 16 Controlador PI discreto. Fuente: [29]

▪ Implementación

La implementación del controlador PI se fundamenta en el anterior diagrama del cual se establece el código implementado en el DSP, a continuación se muestra el bloque básico del controlador[16].

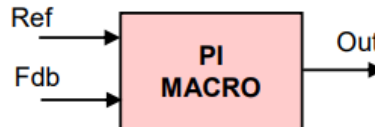


Ilustración 3. 17 Bloque del controlador

A continuación se presenta el código en lenguaje C.

```
#define PI_MACRO(v)
/* proportional term */
v.up = _IQmpy(v.Kp, (v.Ref - v.Fbk));
/* integral term */
v.ui = (v.Out == v.v1)?(_IQmpy(v.Ki, v.up)+ v.i1) : v.i1; \
v.i1 = v.ui;
/* control output */
v.v1 = v.up + v.ui;
v.Out= _IQsat(v.v1, v.Umax, v.Umin);
```

4. RESULTADOS DE FUNCIONAMIENTO DEL SISTEMA DE CONTROL

4.1 Pruebas de funcionamientos y validación

Para las pruebas de funcionamiento y validación de todas las etapas propuestas en los capítulos anteriores se utilizaron diversos elementos de medida y extracción de datos los cuales se encuentran en los laboratorios y/o grupos de investigación de la universidad de pamplona con el fin de verificar el correcto funcionamiento de cada etapa antes planteada para realizar una validación final del funcionamiento, los equipos continuación mencionados están certificados por estándares y normas que los rigen.

Equipos de medición y registro de datos

Osciloscopio digital portátil de 20MHz - Fluke 123

Descripción

Osciloscopio de Almacenamiento Digital ScopeMeter® de 20 MHz con Multímetro y Registrador de Datos; Pantalla Monócroma. Modelo Fluke 123 Código: FLUKE-123/008[30].

Características

- Osciloscopio digital de dos canales de 20 MHz
- TrendPlot™: registrador de dos canales
- Función disparo automatico Connect-and-View™ para funcionamiento sin manos
- Cables simples de prueba con blindaje para mediciones de osciloscopio, resistencia y continuidad
- Duración de las baterías: hasta 7 horas
- Seguridad eléctrica: CAT III 600 V
- Interfaz RS-232 ópticamente aislada
- Equipo compacto y muy robusto
- 2 memorias de pantalla y 10 configuraciones



Ilustración 4. 1 Fluke 123.

Analizador de calidad de la energía eléctrica - Fluke 434-II

Descripción

Analizador de calidad de la energía eléctrica (PQ) trifásico, cuenta con la nueva función de cálculo de pérdida de energía, cuenta también con la medición de potencia eléctrica unificada (UPM), además podrá detectar y solucionar problemas en primer nivel de actuación, realizar un mantenimiento predictivo, hacer análisis a largo plazo y estudios de carga; incluye las nuevas pinzas i430flex-TF. Modelo Fluke 434-II Codigo: FLUKE-434-II[30].

Características

Calculadora de Pérdida de Energía.

- La clasificación de seguridad más alta de la industria: Clasificación de **CAT IV** a 600V y **CAT III** a 1.000V para su uso en la entrada de servicio.
- Permite medir las tres fases y el neutro.
- Todas las mediciones se registran siempre automáticamente, sin necesidad de configuración alguna.
- Diez parámetros de calidad de potencia en una sola pantalla, de acuerdo con la norma de calidad de potencia eléctrica **EN50160**.
- Función de registrador: Configurado para cualquier condición de prueba con memoria de hasta 600 parámetros a intervalos definidos por el usuario.
- Visualización de gráficos y generación de informes.
- Medición de Potencia Eléctrica Unificada.



Ilustración 4. 2 Fluke 434-II Fuente: [30].

4.2 Resultados obtenidos de las etapas antes propuestas

Para realizar el control y monitoreo del sistema se hizo necesario el monitoreo de las variables importantes a través de una interfaz de monitoreo en el prototipo desarrollado como la lectura de los sensores de voltaje y corriente, la lectura de las señales filtradas digitalmente, las transformaciones de los bloques de Clark, el estimador de flujo y velocidad; la respuesta del controlador en el lazo de control;

también desde la misma interfaz se controla el punto de referencia de velocidad del motor del controlador entre otras variables de las cuales se puede obtener información en tiempo real.

Mediada de las señales de corriente

Al implementar los sensores de corrientes antes descritos y realizas la configuración de las entradas análogas del DSP empleado se obtuvieron la señales apreciables en la ilustración 4.4 se puede apreciar que las formas de ondas presentan ruido pero también se puede apreciar que el desfase del sistema trifásico es el apropiado.

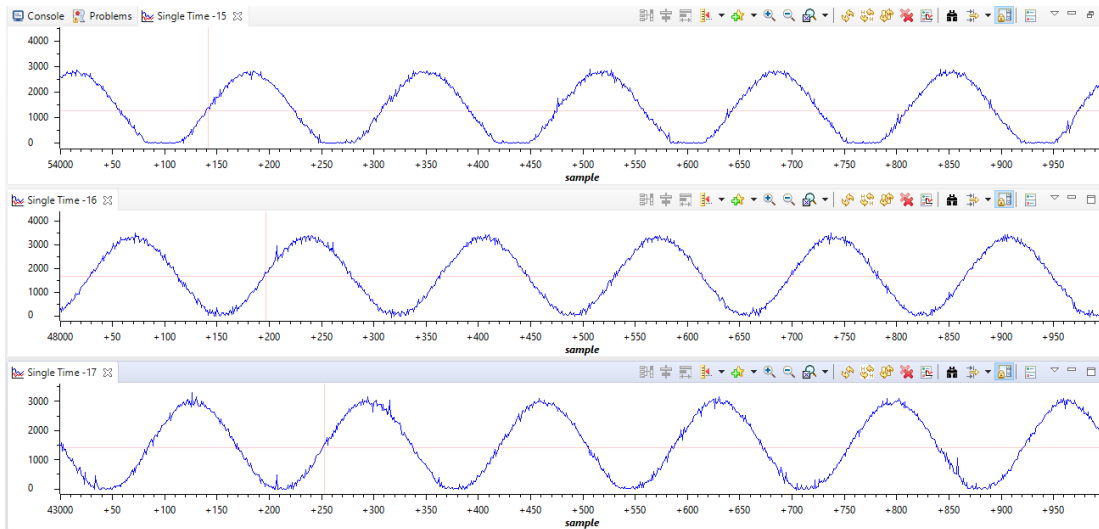


Ilustración 4. 3 Señales de corriente medidas. Fuente: autor

El ruido presente en las señales se puede observar en la siguiente ilustración 4.5.

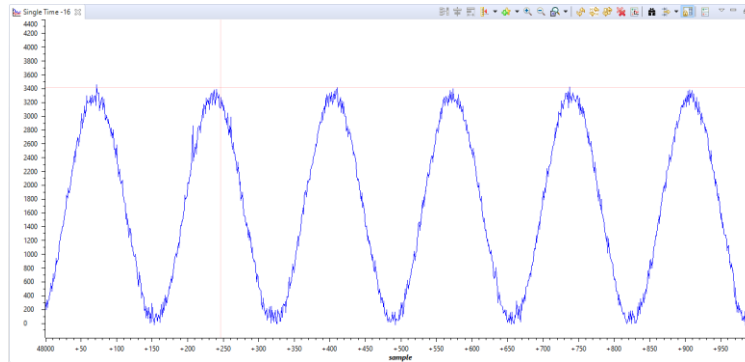


Ilustración 4. 4 Señal de corriente con ruido

Al aplicar el filtrado digital se las señales de corriente se puede obtener una forma de onda casi sinusoidal como se puede apreciar en la siguiente ilustración 4.6, también se aprecia que las señales están normalizadas además de quitarles los valores de offset sumados en la etapa de adquisición analógica.

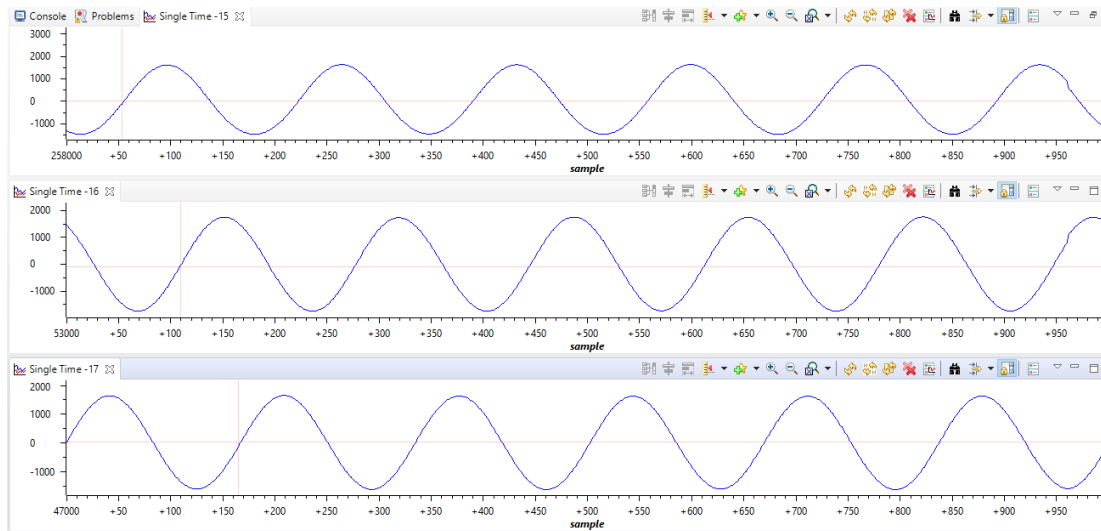


Ilustración 4. 5 Señales filtradas de corriente

La fase *Ia* se aprecia a continuación

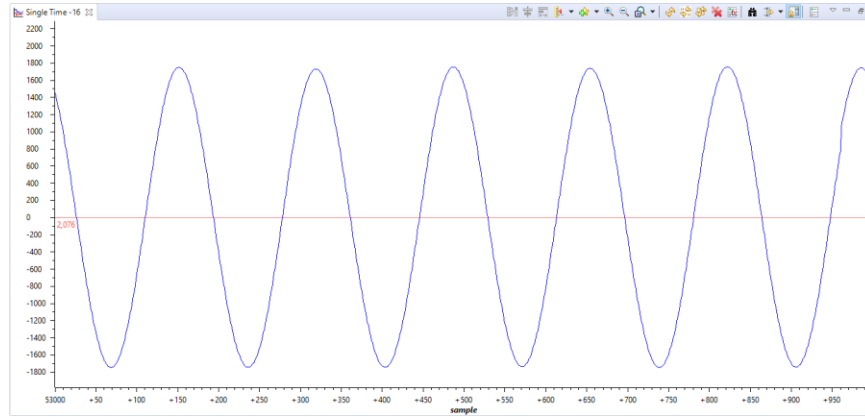


Ilustración 4. 6 Corriente de la fase IA. Fuente: Autor

Las tres señales de corrientes graficadas y normalizadas a valores nominales trifásicas se pueden apreciar a continuación ilustración 4.8 en las cuales se destaca el correcto desfase que presentan las señales de corriente,

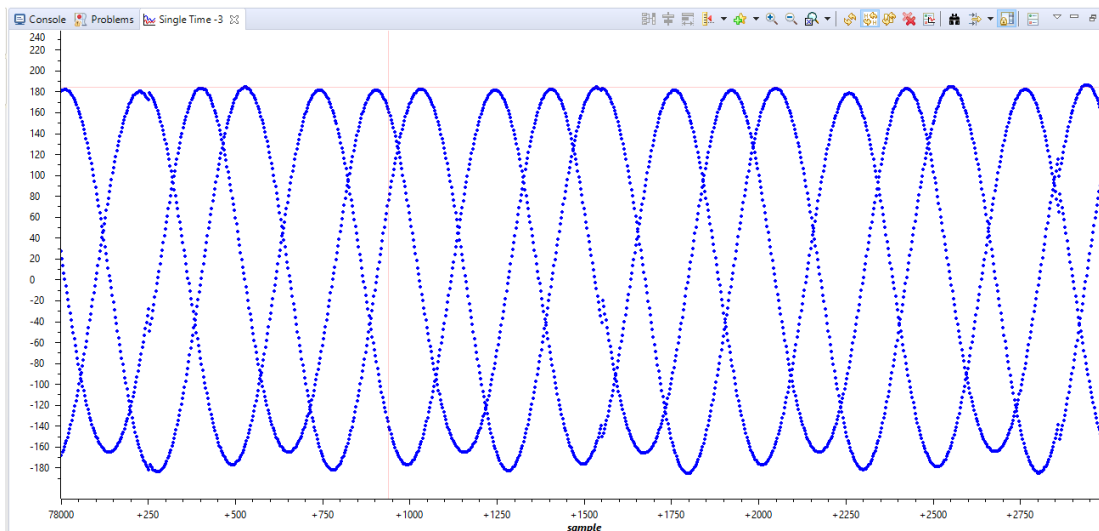


Ilustración 4. 7 Corrientes trifásicas normalizadas. Fuente: Autor

Las señales de voltaje que se obtuvieron con los circuitos de acondicionamiento antes descritos se pueden apreciar en las siguiente ilustración 4.9 las cuales son unas señales PWM provenientes del el convertidos de frecuencia.

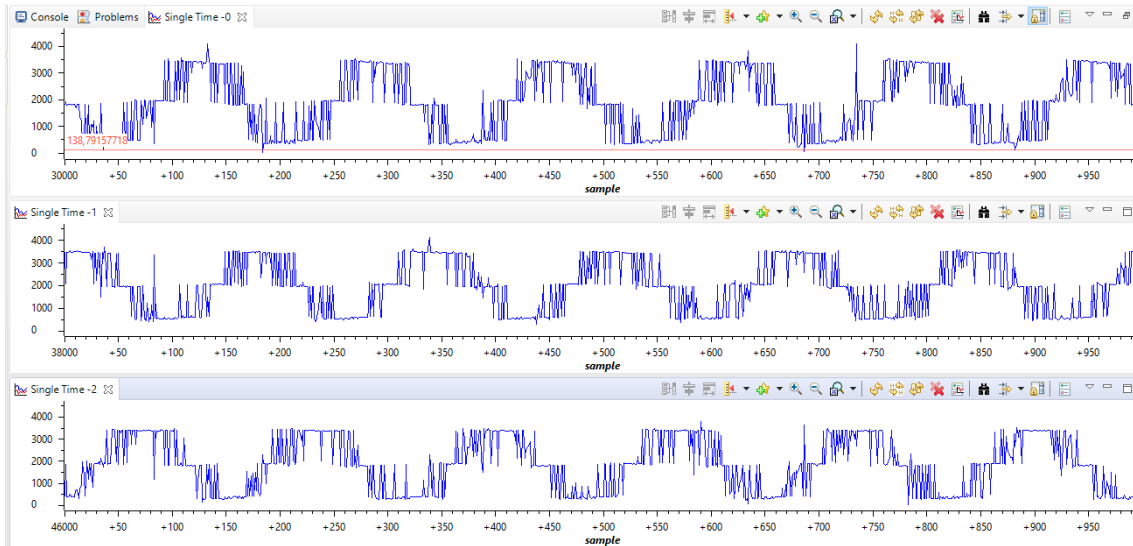


Ilustración 4. 8 Señales de voltaje PWM de entrada motor de inducción. Fuente: Autor

En la siguiente ilustración 4.10 de la señal de voltaje de la fase VA se puede apreciar a forma de onda PWM adquirida la cual presenta un poco de ruido; las señales de voltaje se deben de filtrar con el fin de obtener unas señales sinusoidales necesaria para el correcto funcionamiento de los bloques de estimación de velocidad y flujo.

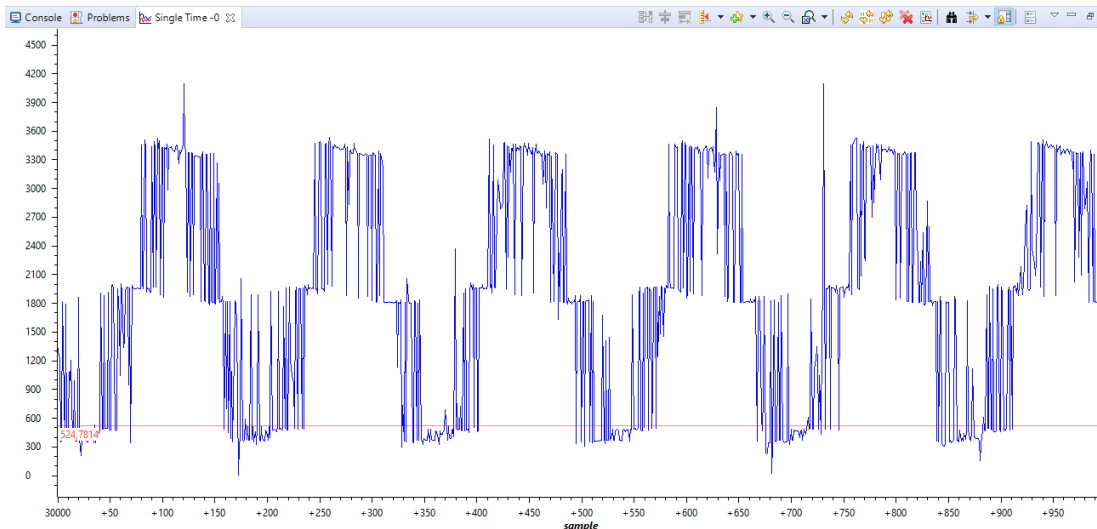


Ilustración 4. 9 Señal de voltaje fase VA. Fuente: Autor

Al aplicar el filtro ante descrito (Fir de 195 coeficientes) se obtiene una forma de onda sinusoidal.

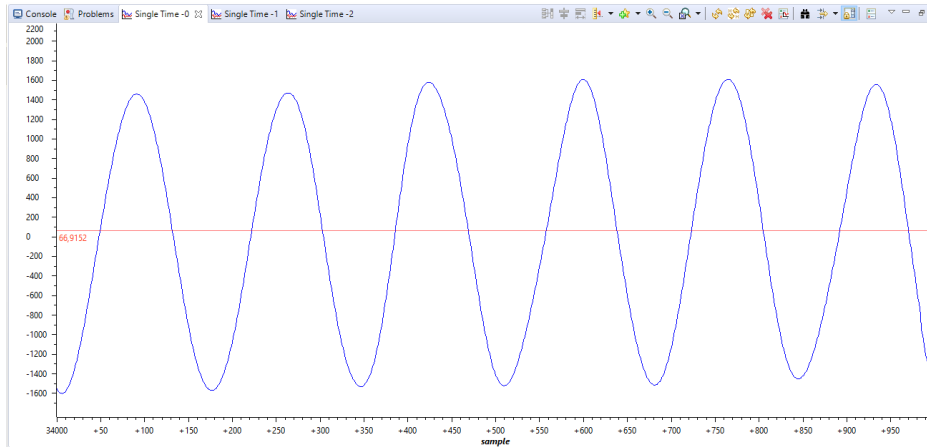


Ilustración 4. 10 Señal de voltaje fase VA. Fuente: Autor

A continuación ilustración 4.12 las señales de voltaje trifásicas.

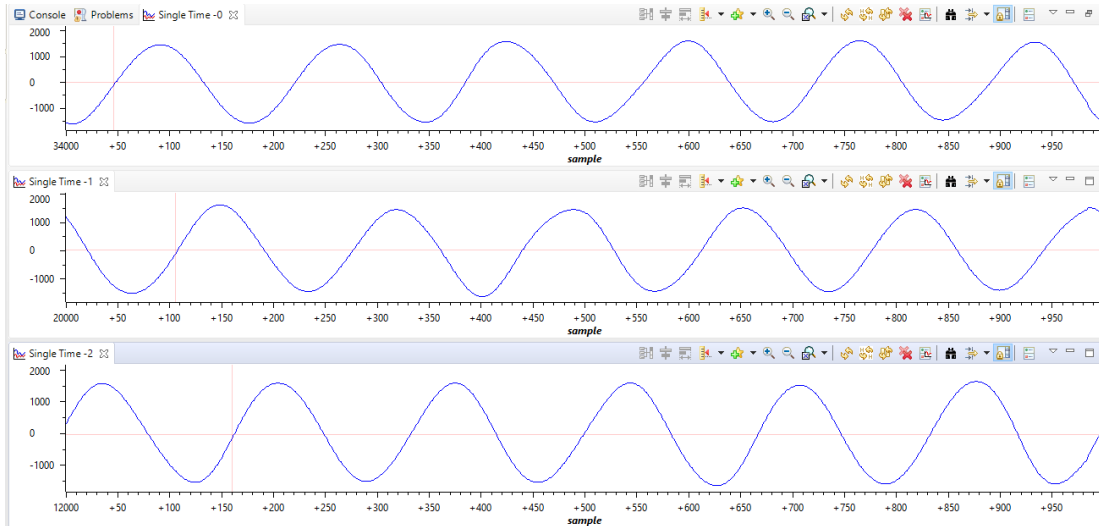


Ilustración 4. 11 Señales de voltaje

Al normalizar y escalar las señales de voltaje en una forma correcta se obtuvo cabe destacar que las formas de ondas presentan discontinuidades en amplitud esto se debe a su naturaleza de señal PWM.

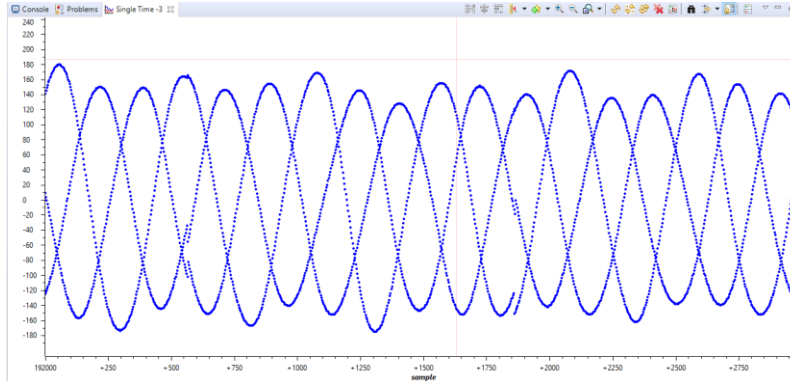


Ilustración 4. 12 señales trifasicas de Voltaje

4.3 Pruebas a motor de inducción de rotor bobinado

Para realizar la estimación de la velocidad es necesario obtener los parámetros eléctricos de motor de inducción en su respectivo modelo mostrados en la siguiente ilustración.

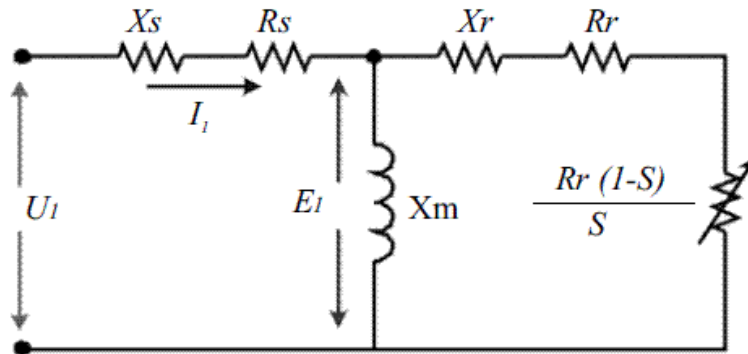


Ilustración 4. 13 Circuito equivalente del motor de induccion

Para obtener la resistencia de rotor R_s se encendió el motor en sus valores nominales dejándolo funcionando por un lapso de tiempo largo y luego se midió la resistencia de un devanado con un multímetro ya que la resistencia cambia con la temperatura.

Obteniendo a:

$$R_s = 7\Omega$$

La siguiente prueba es la de vacío en la cual se pone al funcionar en su voltaje nominal 220 ver **Anexo 11** se puede observar el motor empleado usado para el desarrollo del proyecto.

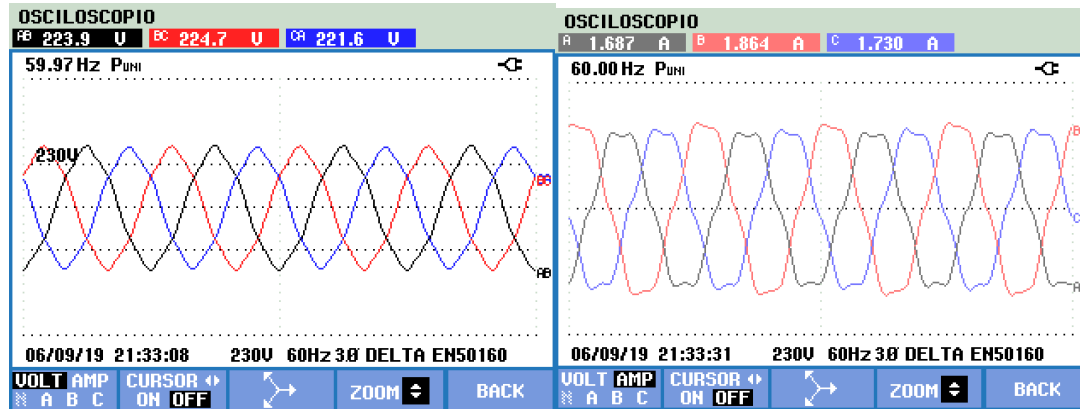


Ilustración 4. 14 Prueba en vacío

De la prueba en vacío del motor de inducción se obtienen los siguientes datos para calcula x_m :

$$X_m + X_1 = Z_{equi} = \frac{V\phi}{I\phi} = \frac{224}{\left(\frac{1.68}{\sqrt{3}}\right)} = j230\Omega$$

De la prueba de rotor bloqueado se obtienen las siguientes datos

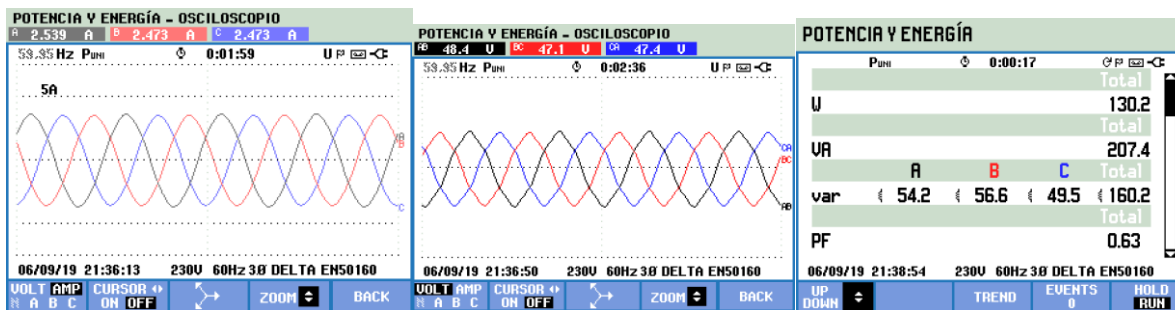


Ilustración 4. 15 Rotor Bloqueado

$$\sin(\theta) = \sin \cos^{-1} 0.63 = 0.777$$

$$Z_{rb} = \frac{V\phi_{rb}}{I\phi_{rb}} = \frac{47.26}{1.42} = 32.77$$

$$R_{rb} = z_{rb} * \cos(\theta) = 20.62$$

$$R_r = 20.62 - 7 = 13.62\Omega$$

$$X_{rb} = Z_{rb} * \sin(\theta) = j25.24\Omega$$

Factor K para motores de inducción de rotor bobinado es de 0.5 por tanto

$$x_r = x_s = 0.5 * 25.24 = 12.62$$

$$X_m = 230.94 - 12.62 == j218.52$$

Los valores finales serian:

$$R_r = 13.62\Omega$$

$$R_s = 7\Omega$$

$$L_r = 0.0334H$$

$$L_s = 0.0334H$$

$$L_m = 0.5791H$$

Estos valores son usados por el macro de estimación de voltaje junto con algunos datos de la máquina del **Anexo 11**

4.4 Validación de la estimación del flujo y velocidad

En la siguiente ilustración se puede observar las transformaciones de Clark de la corriente las cuales presentan su respectivo desfase desacuerdo a las etapas antes planteadas.

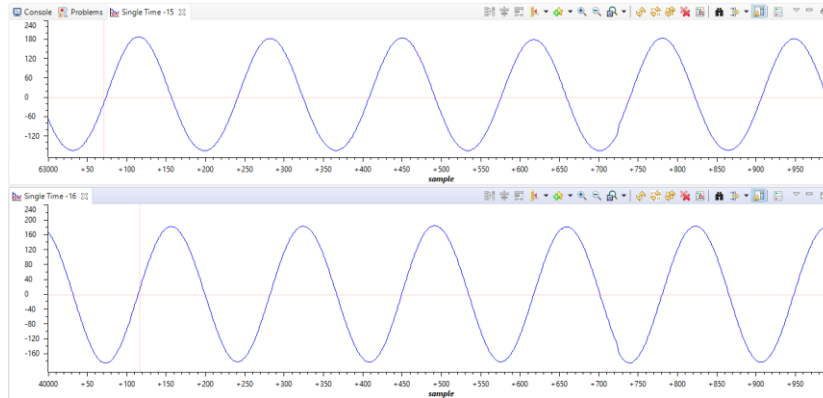


Ilustración 4. 16 Transformada de clark de Corrientes. Fuente: Autor

En la siguiente ilustración se puede observar las transformaciones de Clark del voltaje las cuales presentan su respectivo desfase desacuerdo a las etapas antes planteadas.

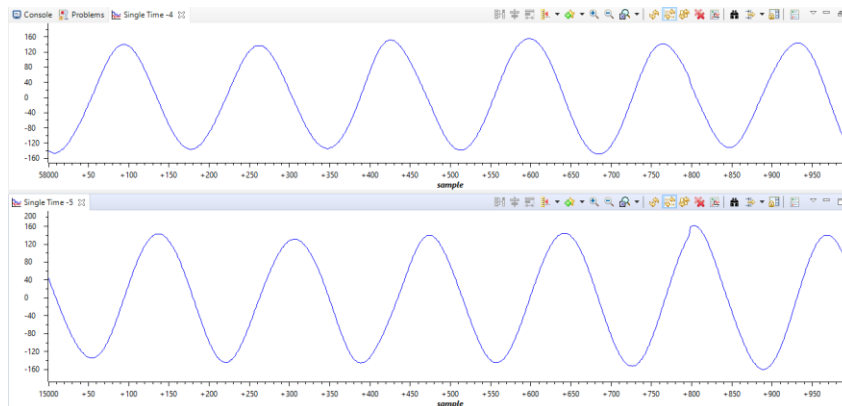


Ilustración 4. 17 Transformada de Clark de voltajes. Fuente: Autor

Las en la siguiente ilustración se puede ver el ángulo del flujo de la máquina de inducción el cual varía de 0 a 2π el cual esta escalado 100 veces sus valores para ser apreciables.

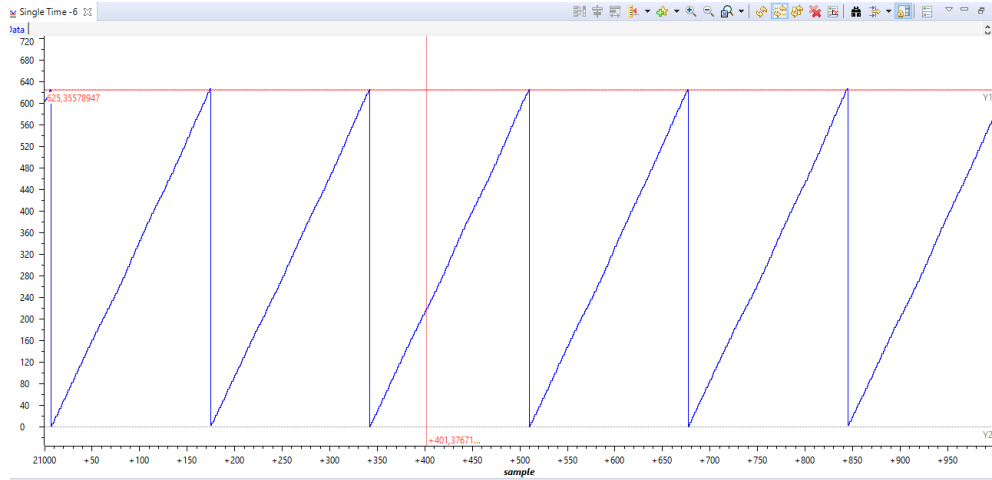


Ilustración 4. 18 Angulo del flujo

A continuación se pueden apreciar el flujo del rotor en el eje estacionario q-d que es la salida del estimador empleado.

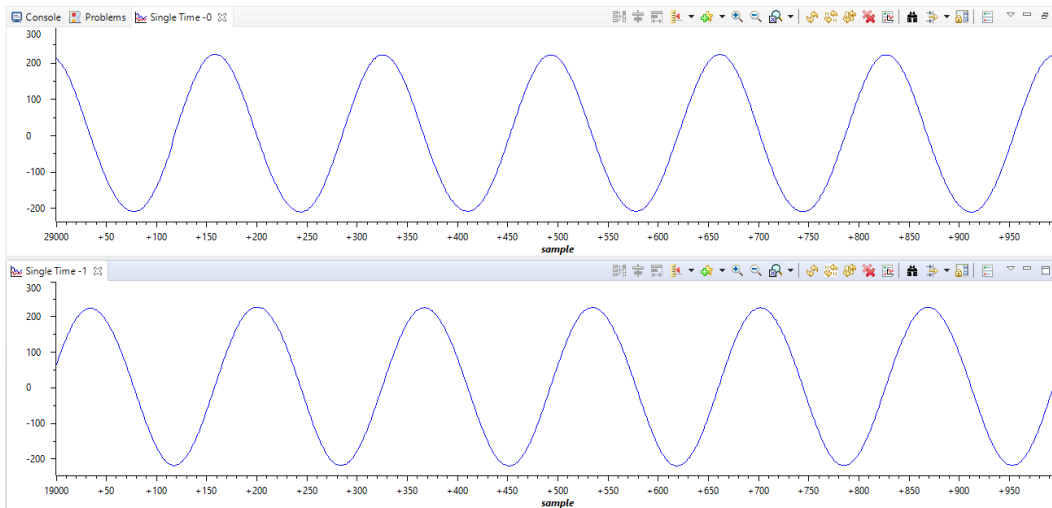


Ilustración 4. 19 Flujo del rotor estimado ejes d-q . Fuente: Autor

4.5 Prueba del sistema a lazo abierto

Para realizar la prueba a lazo abierto se plantea el siguiente estema apreciado en el siguiente ilustración 4.20 donde se puede apreciar las conexiones realizadas en su entrada análoga con y digital logrando controlar el encendido de la maquina junto con su velocidad de forma remota.

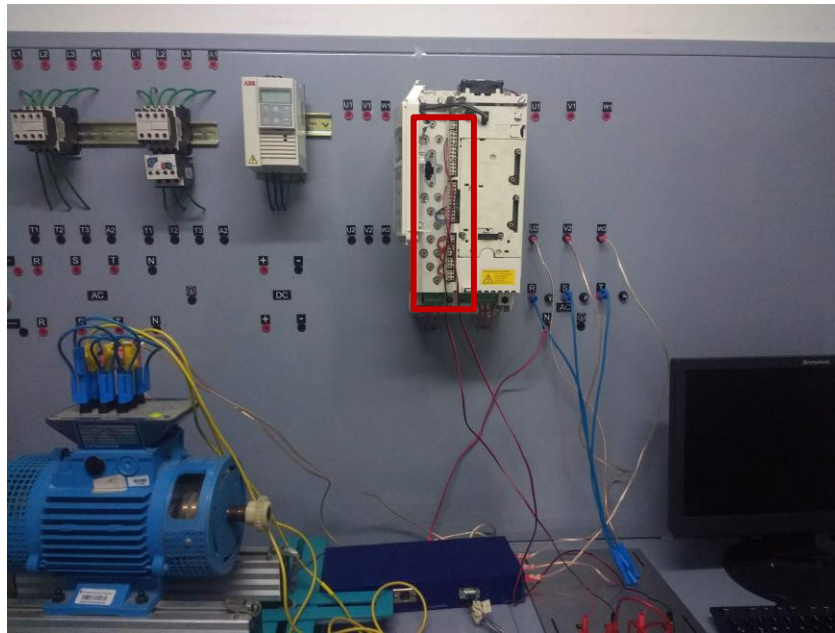


Ilustración 4. 20 Esquema de control a lazo abierto. Fuente: Autor

En primer lugar se prueba el buen funcionamiento del circuito de amplificación descrito anteriormente junto a la correcta configuración del PWM_DAC para la salida de la señal de control del elemento final de control se prueban la salida con un cambio de velocidad de 300Rpm desde 900Rpm el cual se aprecia a continuación.

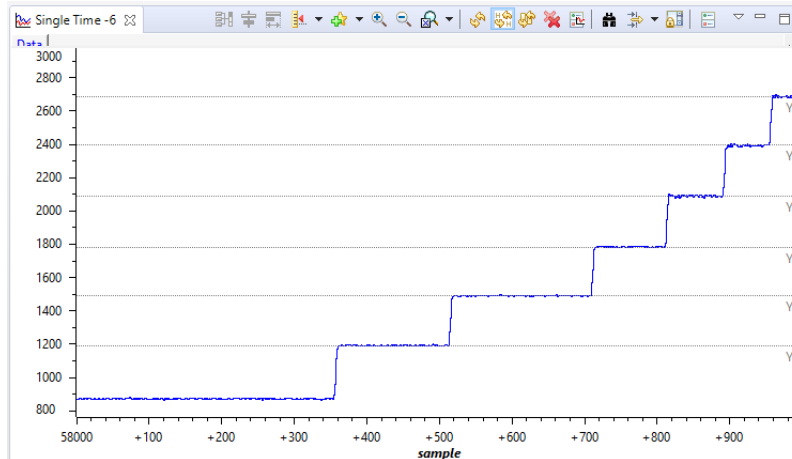


Ilustración 4. 21 Respuesta del sistema ante cambios de velocidad. Fuente: Autor

En la anterior ilustración se puede apreciar que en bajas frecuencias la medida de la velocidad no es muy exacta esto se debe a que el modelo de la máquina de inducción es cambia, pero en la mayoría de puntos presenta una gran exactitud.

Cálculo de las constantes Kp y Ki

Para realizar el cálculo de las constantes Kp y Ki necesaria para la implementación del algoritmo del controlador el cual se va a estar ejecutando en tiempo 0.1 segundos para una mayor exactitud en el control, en primera instancia para el cálculo de parámetros las constantes se hace necesario realizar pruebas de funcionamiento a lazo abierto a la planta (Motor de inducción) de acuerdo al diagrama espuesto en el anterior item.

De acuerdo al montaje realizado se varió el sistema, empezó a funcionar en una velocidad de aproximada de 871 *Rpm*, luego se realizó un cambio de velocidad con un step *U* registrado en el GPIO0 a una velocidad objetivo de 2350 *Rpm*, obteniendo el comportamiento del sistema ante un cambio de velocidad la cual es muestreada con un DAQ a una tasa de adquisición de $\frac{1}{10Hz}$, los resultados obtenidos se puedes apreciar en la ilustración.

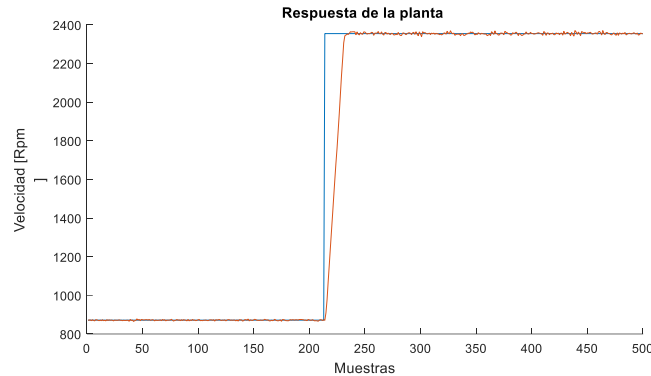


Ilustración 4. 22 Respuesta de la planta ante un escalon

Para el cálculo de los parámetros del controlador implementado se sintonizó en Matlab, partiendo de la función de transferencia obtenida de la identificación completa del sistema en lazo abierto; donde el TMS320F28069M controla el convertidor de frecuencia de acuerdo a las etapas antes planteadas; con la herramienta de identificación de sistemas IDENT de Matlab podemos obtener un modelo del comportamiento de nuestra planta a partir de unas variables de entrada como se puede ver en la ilustración 4.23.

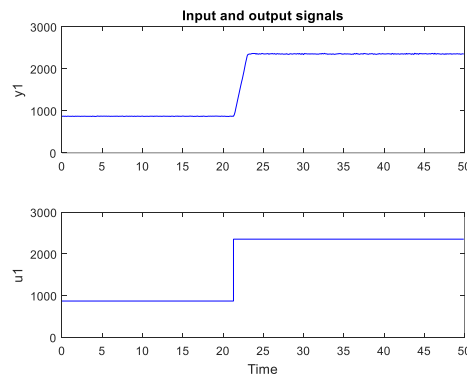


Ilustración 4. 23 Datos cargados a Ident Fuente: Autor Ident-Matlab

Se obtiene un modelo con el 98,85% de similitud ilustración 4.24 con la planta real para poder realizar la sintonización e las constantes del controlador a implementar donde tenemos el siguiente modelo para nuestra planta que se puede observar, cuenta con 3 polos y 2 ceros.

$$\frac{0.5897 * S^2 + 0.66083 * S + 10.053}{S^3 + 4.0456 * S^2 + 9.9217 * S + 10.0521}$$

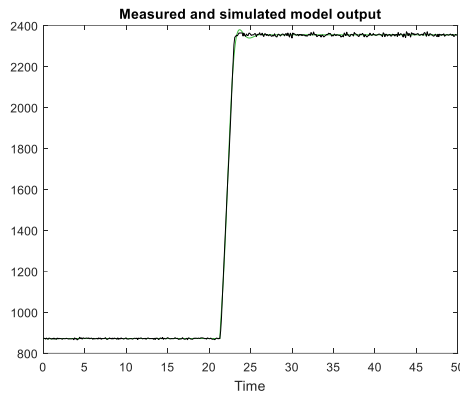


Ilustración 4. 24 Modelo obtenido. Fuente: Autor

Co la ayuda de la herramienta symulink se simula la planta junto con todo el lazo de control además de incluir el controlador proporcional integral en el lazo cerrado de control, que parte de una sintonización automática con la cual se puede obtener una aproximación de las constantes k_p y k_i

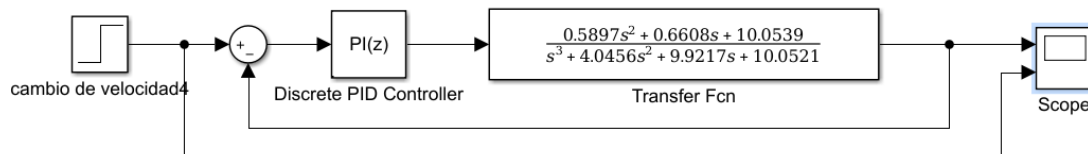


Ilustración 4. 25 Simulación de lazo de control. Fuente: Autor

Al realizar la simulación ilustración 4.25 y sintonización del controlador en tiempo discreto con un tiempo de discretización 100ms tiempo que el controlador da una acción de control, obteniendo las constantes:

$$K_p = 0.3782$$

$$K_i = 0.7909$$

Al simular cambio de velocidad ilustración, junto con un cambio de fallo obtenemos una satisfactoria respuesta en la simulación.

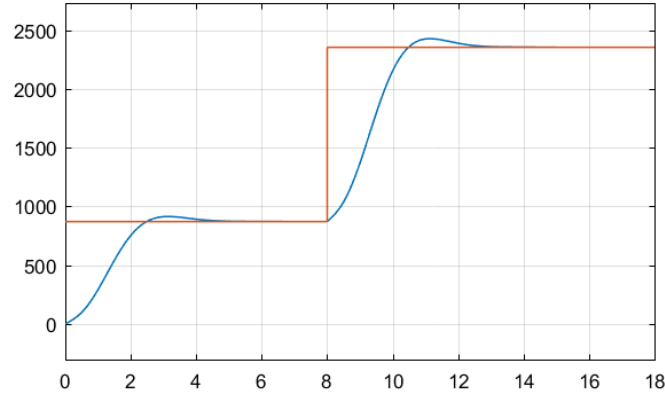


Ilustración 4. 26 Gráfica del controlador Fuente: Autor

Vemos que el controlador simulado funciona correctamente actuando de una forma correcta ante posibles perturbaciones y/o cambios de velocidad, de acuerdo a esto se posee una base de partida para poder realizar variaciones en las constantes del controlador implementado en la tarjeta, en la próxima sección se procederá a realizar la prueba a lazo cerrado del sistema de control.

4.6 Funcionamiento de todo el sistema de control a lazo cerrado

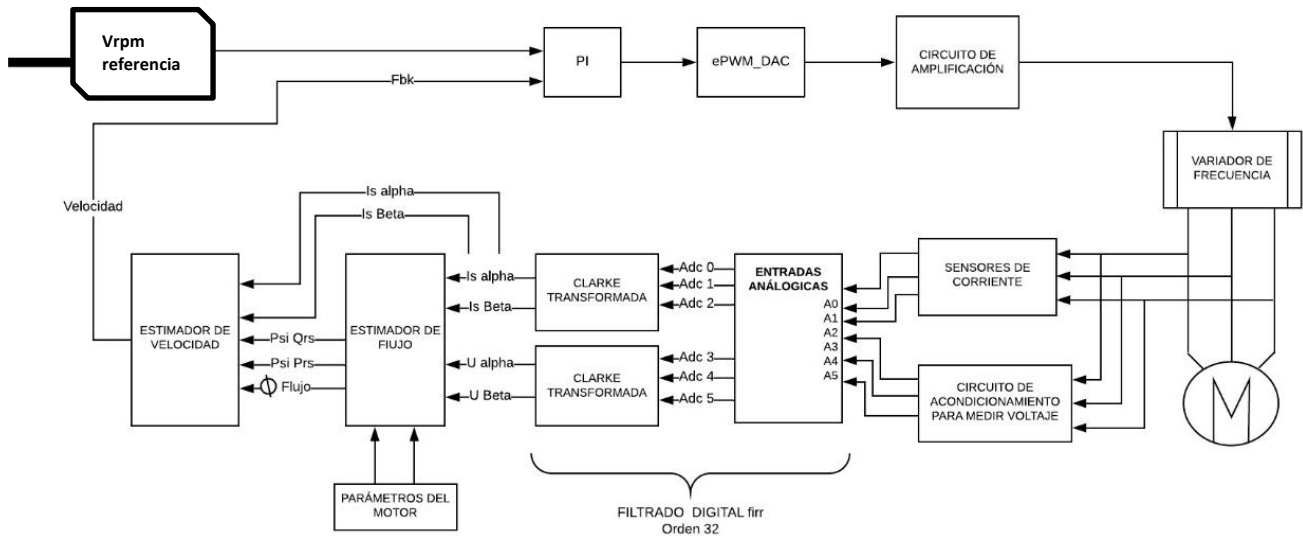


Ilustración 4. 27 Sistema a lazo cerrado

En la ilustración 4.28 se puede apreciar el montaje real efectuado para la realización de las pruebas con todas las etapas ya descritas y fundamentadas en capítulos y secciones anteriores donde la interfaz de comunicaciones esta en conexión directa con la tarjeta TMS320F28069 a través del enlace de conexión provisto por la JTAG de el LAUNCHXL empleado.

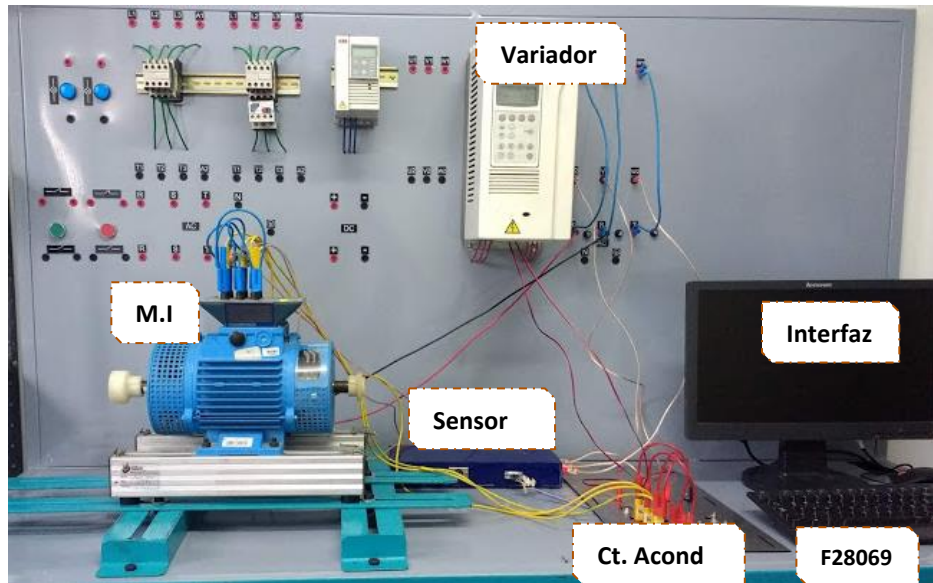


Ilustración 4. 28 Montaje final. Fuente: Autor

De acuerdo a las constantes K_p y k_i encontradas anteriormente se toma como referencia para empezar a realizar pruebas a lazo cerrado, con el controlador funcionando, que como es de esperarse las constantes k_p y k_i encontradas, solo nos dan una base de referencia para empezar a variar las constantes de acuerdo a las respuestas obtenidas.

A través de la interfaz de monitoreo planteada se establece una velocidad objetivo de 1800 *Rpm* cabe resaltar que esta velocidad objetivo es la real del motor validada con el tacómetro digital. En la ilustración 5.29 se puede apreciar una captura de pantalla de la interfaz donde se muestra y_k .

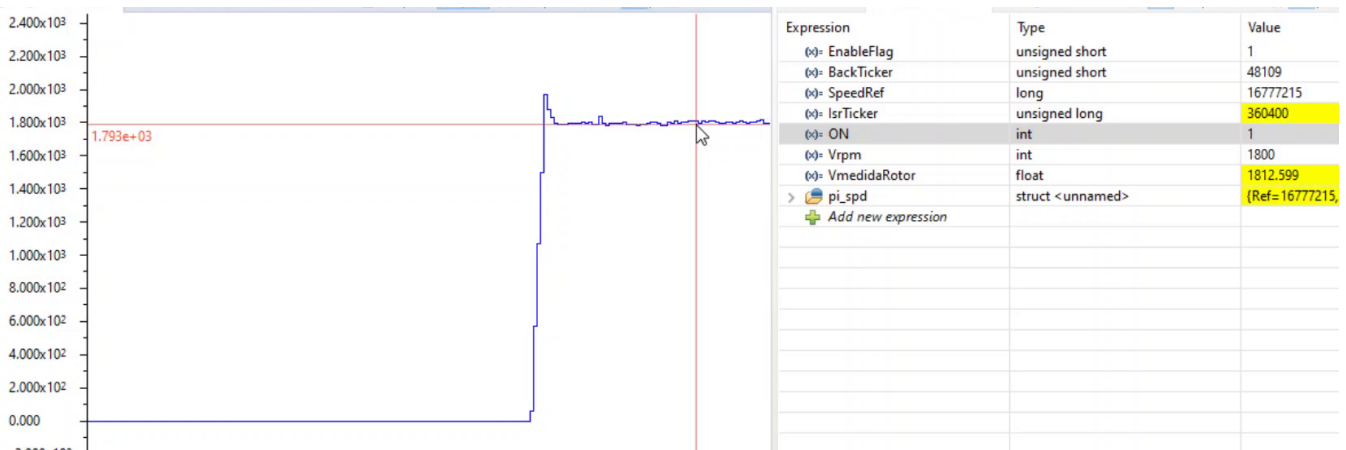


Ilustración 4. 29 Cambio de velocidad. Fuente: Autor

Para validar que la maquina se encuentre en una velocidad de 1800 *Rpm*, en la ilustración se puede apreciar la velocidad real de la máquina de inducción medida a través de un tacómetro digital ilustración 5.38.



Ilustración 4. 30 Velocidad real medida

En la ilustración 4.30 se cambia el set point de 1800 *Rpm* a 1112 *Rpm* con el fin de visualizar el cambio de velocidad de un punto a otro, observando así el controlador en forma directa ante un cambio de velocidad.

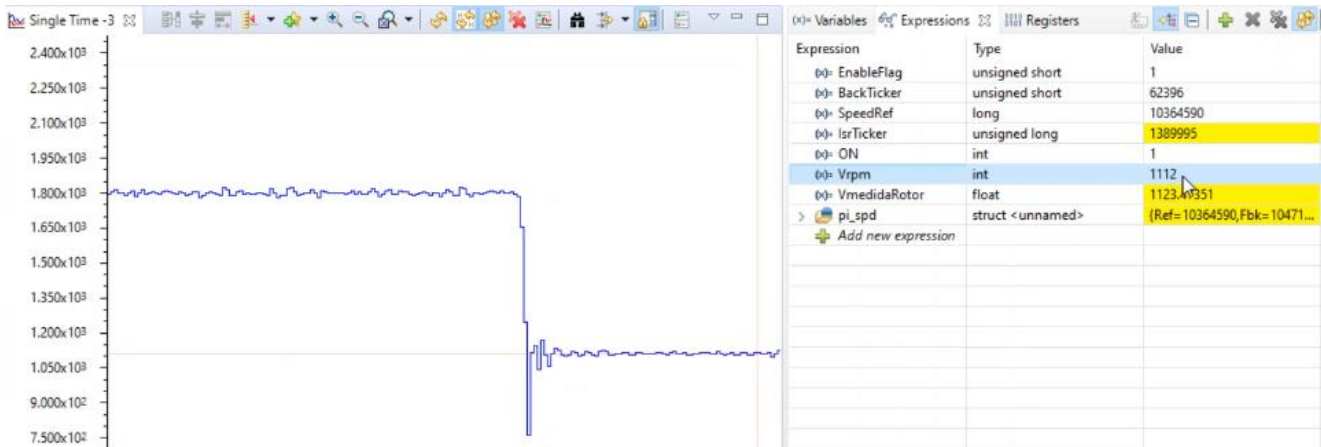


Ilustración 4. 31 Cambio de velocidad por debajo de la nominal. Fuente: Autor

En la siguiente ilustración 4.31 se aprecia la velocidad real del rotor medida con el tacómetro digital en *Rpms* presentando una velocidad estable pero con un pequeño error de estado estacionario; aunque la velocidad más aproximada es la estimada con el prototipo planteado ya que presenta gran exactitud.



Ilustración 4. 32 Medidas de velocidad por debajo de la nominal. Fuente: Autor

Por último se realiza un cambio a la velocidad por encima de la nominal para validar completamente el sistema implementado a 2150Rpm como se puede apreciar en la siguiente ilustración.

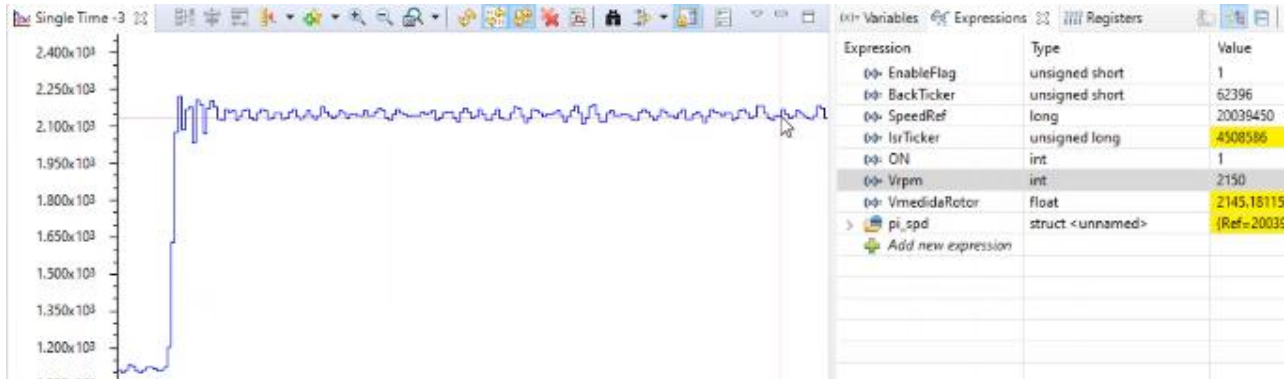


Ilustración 4. 33 cambio veliciada por encima nominal. Fuente: Autor

Al validar las mediciones con el tacómetro y la estimada con el convertidor de frecuencia se observa como la medida estimada de la velocidad realizada con el desarrollo se aproxima más a la real.



Ilustración 4. 34 Velocidad medida a 2150Rpm. Fuente: Autor

5. CONCLUSIONES

- La familia de microcontroladores y microprocesadores C200 de la Texas Instruments proporciona una amplia gama de microcontroladores para aplicaciones de control, procesamiento de datos, redes neuronales con muy bajo coste y relativa dificultad en el desarrollo el TMS320F28069 usado en este proyecto tiene grandes prestaciones, varias de las cuales se usaron en el desarrollo del proyecto permitiendo el agrupamiento de todos los módulos a un bajo consumo de memoria y alta velocidad en el procesamiento.
- El acondicionamiento y filtrado de las señales provenientes del sensor, es vital para cualquier desarrollo donde se involucre un sistema de control a lazo cerrado.
- La programación CCS es compleja por lo que requiere tiempo para entenderla, la Texas Instruments nos da todas las herramientas tipo documentación, soporte, entrenamiento, soporte técnico por profesionales para lograr sacar adelante cualquier desarrollo propuesto, todo esto a base del software CODE COMPOSER STUDIO el cual fue la base de todos los algoritmos desarrollados.
- Para realizar desarrollos en el software code composer studio es necesario tener conocimiento sobre arquitectura de microcontroladores, su organización en los distintos bancos de registros, ya que es la principal característica de este tipo de programación enfocada específicamente a la TMS320F28060 basada en estructuras de campos de bits, que optimizan la memoria y además permite que a cualquier desarrollo en especial al descrito en este proyecto tener un completo manejo en tiempos de ejecución.
- El sistema presenta un error en la estimación en frecuencias bajas pues en primer lugar el modelo de la maquina cambia, y también al estar filtrando señales de naturaleza pwm no presentan la misma forma de onda en todos los rangos de frecuencia presentando discontinuidades las cuales se traducen en errores en la estimación de la velocidad.
- Los bloques empleados para el cálculo de la velocidad son de gran ayuda en el momento de realizar un control sensorless como el que está planteado en este proyecto ya que estos proporcionan una gran información referente a como se configuran.
- En el desarrollo de este proyecto se tuvo especial cuidado con la adquisición de las señales, su correcta secuencia de fases, el correcto escalado a valores

en por unidad de acuerdo a una base fueron la una delas razones principales por las cuales se logró culminar el proyecto a cabalidad con el proyecto plantado.

- El control sensorles presenta grandes ventajas en cuanto al control de velocidad basado en sensores de velocidad ya que las variables eléctricas presenta un mayor rango de valores finitos además de cambiar más rápido de un punto de operación a otro.

6. REFERENCIAS BIBLIOGRÁFICAS

- [1] F. Hernán, M. Sarmiento, J. Jairo, y G. López, «Control escalar en motores de inducción monofásicos», *Tecnura*, vol. 10, n.º 19, pp. 29-37, 2006.
- [2] L. D. P. FERNÁNDEZ, «CONTROL DE VELOCIDAD DEL MOTOR DE INDUCCIÓN MEDIANTE CONVERTIDOR DE POTENCIA MULTINIVEL CON OPTIMIZACIÓN DE ARMÓNICOS», Pamplona, Norte santander, 2016.
- [3] A. Fernando, L. Villamizar, J. Luis, y D. Rodríguez, «Minimización de la distorsión armónica de una modulación PWM con algoritmos genéticos», vol. 8, pp. 79-86, 2011.
- [4] J. Villalón y F. Salas Gómez, «Modelado, simulación y control de convertidores en cascada», Sevilla, 2017.
- [5] K. Preethi, G. Anil, y E. Vani, «Speed Control of Induction Motor Using Eleven

- Levels Multilevel Inverter», *circuit.International J. Sci. Mod. Eng.*, n.º 5, pp. 15-20, 2013.
- [6] Texas Instruments Incorporated, «TMS320C28x CPU and Instruction Set Reference Guide», 2015.
- [7] J. Pérez y J. Barrero, «Diseño de un sistema empotrado basado en DSP C2000 de Texas Instruments para control de armario de potencia de 12 fases», Sevilla, 2015.
- [8] B. Larimore, «HVAC Dual-AC Motor Control with Active PFC Implementation Using Piccolo™ MCUs», 2010.
- [9] Texas Instruments Incorporated, «C2000™ Piccolo™ 32-bit MCU Family», Dallas, Texas, 2010.
- [10] Texas Instruments Incorporated, «TMS320F2806x Piccolo™ Microcontrollers», Dallas, Texas, 2018.
- [11] Texas Instruments Incorporated, «Silicon Errata TMS320F2806x Piccolo™ MCUs», Dallas, Texas, 2018.
- [12] Texas Instruments Incorporated, «User's Guide LAUNCHXL-F28069M Overview», Dallas, Texas, 2019.
- [13] Texas Instruments Incorporated, «Code Composer Studio™ v8 Integrated Development Environment (IDE) for Embedded Processors», Dallas, Texas, 2014.
- [14] A. Pardo y J. L. Díaz, *Aplicaciones de los convertidores de frecuencia estrategias pwm en el control*. Colombia: 2004, 2004.
- [15] Peñaranda y A. E. Caicedo, «Sensorless control of an induction motor with common source multilevel converter», *WSEAS Trans. POWER Syst.*, pp. 1-2, 2018.
- [16] T. Instruments, «Digital Motor Control : Software Library: Target Independent Math Blocks», 2013.
- [17] H. F. Rashag, S. P. Koh, A. N. Abdalla, N. M. L. Tan, y K. H. Chong, «Modified Direct torque control using algorithm control of stator flux estimation and space vector modulation based on fuzzy logic control for achieving high performance from induction motors», *J. Power Electron.*, vol. 13, n.º 3, pp. 369-380, 2013.

- [18] B. Akin y M. Bhardwaj, «Sensorless Field Oriented Control of 3-Phase Induction Motors», *Appl. note*, p. 43, 2010.
- [19] C. Lascu, I. Boldea, y F. Blaabjerg, «Direct torque control of sensorless induction motor drives: A sliding-mode approach», *IEEE Trans. Ind. Appl.*, vol. 40, n.º 2, pp. 582-590, 2004.
- [20] ABB, «Manual de firmware Programa de control estándar 7.x del ACS800». 2011.
- [21] J. FRAILE MORA, *MÁQUINAS ELÉCTRICAS*, 6ª Edición. MCGRAW-HILL / INTERAMERICANA DE ESPAÑA, S.A., 2008.
- [22] J. DIAZ PLATA, «MÉTODO PARA APRENDER A UTILIZAR EL VARIADOR DE FRECUENCIA ABB ACS800-U1 DE FORMA RÁPIDA Y SENCILLA A TRAVÉS DE DIFERENTES PRÁCTICAS DE LABORATORIO», UNIVERSIDAD DE PAMPLONA, 2019.
- [23] Texas Instruments Incorporated, «TMS320x2806x Piccolo Technical Reference Manual». Dallas, Texas, 2017.
- [24] A. . S. M. Martínez, L. Gómez, «Diseño De Filtros Fir», *Esc. Técnica Super. d' Eng. Dep. d' Ingeniería Electrónica*, pp. 1-26, 2010.
- [25] F. Point y D. S. P. Software, «Fixed Point DSP Software Library USER ' S GUIDE», 2019.
- [26] STMicroelectronics, «WIDE BANDWIDTH DUAL J-FET OPERATIONAL AMPLIFIERS s HIGH INPUT IMPEDANCE J – FET INPUT TL074A - TL074B», *Datasheet*, n.º March, pp. 1-10, 2001.
- [27] multisim, «Multisim Live Online Circuit Simulator», 2019. [En línea]. Disponible en: <https://www.multisim.com/>. [Accedido: 28-jul-2019].
- [28] P. Richard, «PID controller tuning», *Comput. Control Eng. J.*, vol. 10, n.º 2, pp. 44-50, 2005.
- [29] T. I. Incorporated, «C2000 Digital Control Library», n.º November, pp. 1-32, 2017.
- [30] Cedesa, «Fluke 123 - Osciloscopio Digital Portátil de 20MHz». [En línea]. Disponible en: <https://www.cedesa.com.mx/fluke/osciloscopios/digitales/123/>. [Accedido: 29-jul-2019].
- [31] ABB, «Convertidores de frecuencia industriales ABB - ACS800, 0,55 a 5600



kW, Catálogo».

DQS is member of:



*Formando líderes para la construcción de un
nuevo país en paz*

7. ANEXOS.

Anexo 1. Guía para crear proyectos en code composer studio en tarjetas C2000.

Para realizar estos pasos se debe de tener descargado e instalado code composer studio, controlSUITE o C2000Ware las últimas son cuales softwares y herramientas de desarrollo para la familia de microcontroladores C2000™.

1. Se Inicie CCS IDE. Cuando abra CCS, se verá la ventana "Workspace Launcher". Escriba C: \ C2000Workspace.

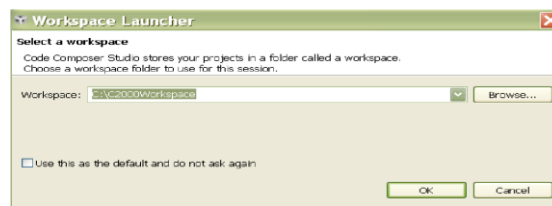


Ilustración 1 CCS

2. Se Cierra la ventana del explorador de recursos y se hace clic en Archivo> Nuevo> Nuevo proyecto CCS para crear un nuevo proyecto.
3. Ingrese el nombre del proyecto y seleccione la configuración del microcontrolador y el programador como se muestra a continuación:

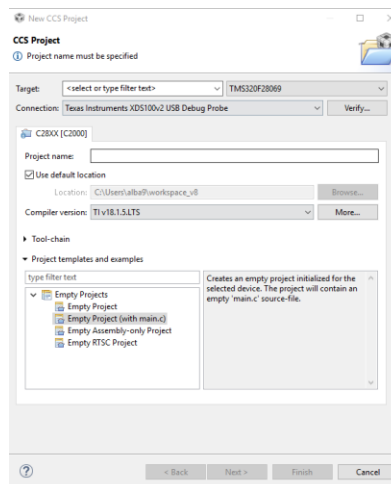


Ilustración 2 Configuración

- Después de crear el proyecto, haga clic en Ventana > Preferencias para editar las preferencias del proyecto. Para crear la Buil Variable como se muestra a continuación direccionándolo donde se encuentra la carpeta C2000

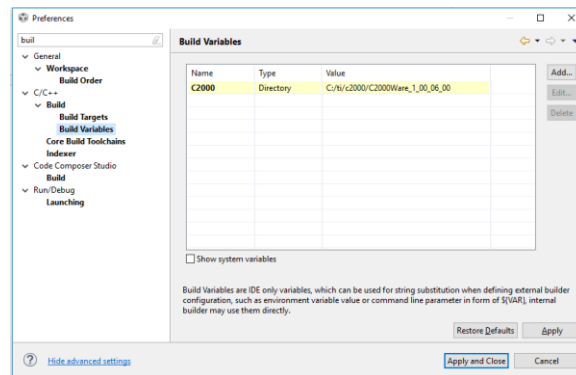


Ilustración 3 Buil Variable

- Al completar este paso nos dirigimos a la dirección mostrada a continuación dependiendo donde se guardó el C200Ware se arrastran los siguientes archivos hacia el título del archivo ejemplo que se creó anteriormente.

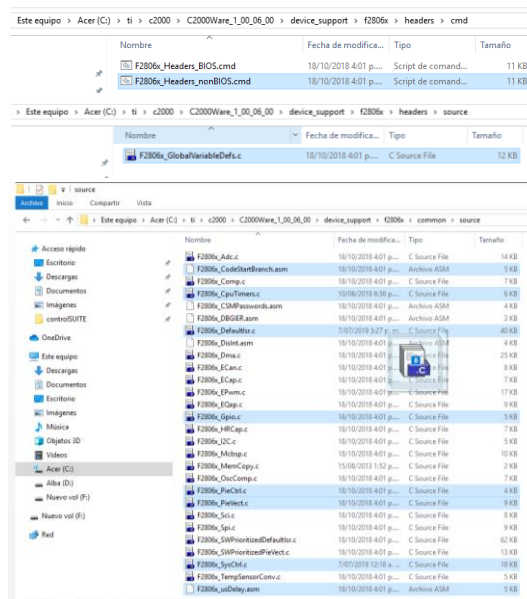


Ilustración 4 Archivos

- Luego se selecciona la variable de control C200 para direccionar todo de una manera correcta

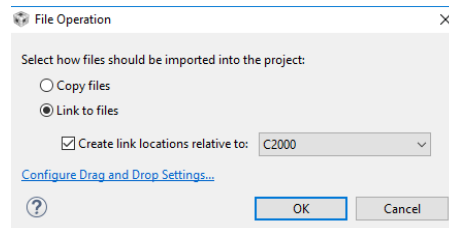


Ilustración 5 Enrutamiento

- Luego se agregan una librerías de la siguiente manera

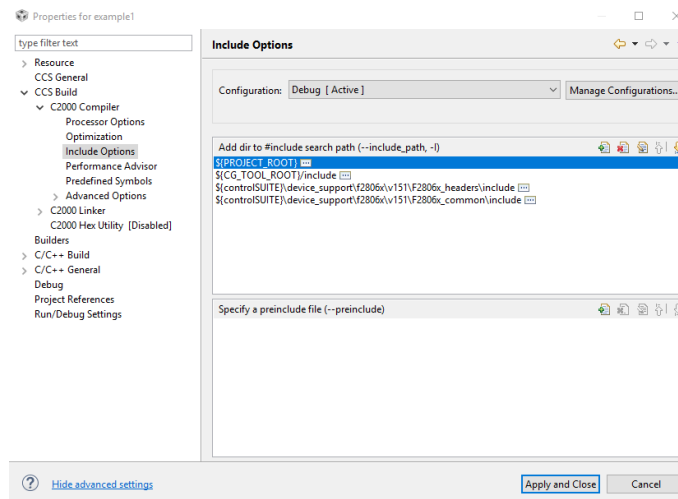


Ilustración 6 Librerías de soporte

- A continuación se presenta un Código ejemplo donde la parte marcada es el ítem para ejecutar y cargar el código a la tarjeta : encender y apagar el led del GPIO34

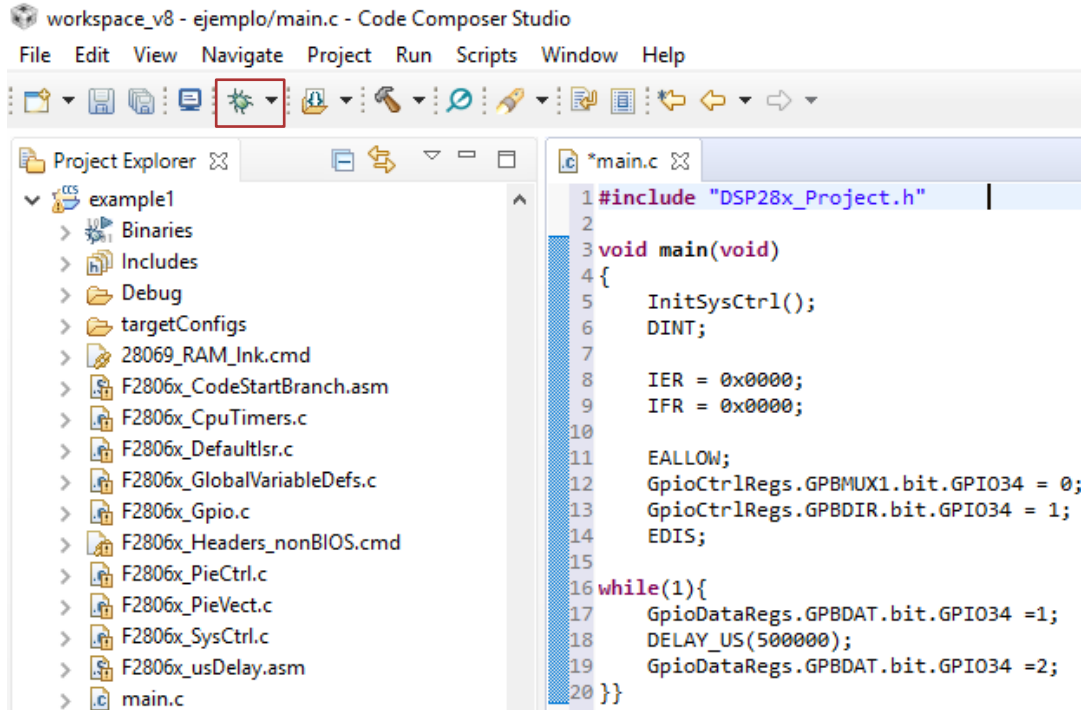


Ilustración 7 ejemplo

Anexo 2. Algoritmo en CCS desarrollado en code composer studio

```
#include<IQmathLib.h>
//#include "PeripheralHeaderIncludes.h"
#define MATH_TYPE IQ_MATH

#include "DSP28x_Project_M.h" // Device Headerfile and Examples Include
File
#include "f2806xileg_vdc_PM.h"
#include "f2806xpwm_PM.h"
#include "filter.h"
#include "HVACI_Sensorless-Settings.h"
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include "string.h"
//

interrupt void OffsetISR(void);
void HVDMC_Protection(void);
interrupt void MainISR(void);
float interu=0.3;
_iq velocidad;
// Global variables used in this system en por unidad

// State Machine func<<tion prototypes
//-----
// Alpha states
void A0(void); //state A0
void B0(void); //state B0
void C0(void); //state C0

// A branch states
void A1(void); //state A1
void A2(void); //state A2
void A3(void); //state A3

// B branch states
void B1(void); //state B1
void B2(void); //state B2
void B3(void); //state B3

// C branch states
void C1(void); //state C1
void C2(void); //state C2
```

```

void C3(void); //state C3

// Variable declarations
void (*Alpha_State_Ptr)(void); // Base States pointer
void (*A_Task_Ptr)(void); // State pointer A branch
void (*B_Task_Ptr)(void); // State pointer B branch
void (*C_Task_Ptr)(void); // State pointer C branch

//int a, b, c, ds1, qs1;
extern Uint16 RamfuncsLoadStart;
extern Uint16 RamfuncsLoadEnd;
extern Uint16 RamfuncsRunStart;
extern Uint16 RamfuncsLoadSize;

int16 VTimer0[4]; // Virtual Timers slaved off CPU Timer 0 (A
events)
int16 VTimer1[4]; // Virtual Timers slaved off CPU Timer 1 (B
events)
int16 VTimer2[4]; // Virtual Timers slaved off CPU Timer 2 (C
events)
int16 SerialCommsTimer;

_iq VdTesting = _IQ(0.2); // Vd reference (pu)
_iq VqTesting = _IQ(0.2); // Vq reference (pu)
_iq IdRef = _IQ(0.1); // Id reference (pu)
_iq IqRef = _IQ(0.05); // Iq reference (pu)
_iq SpeedRef = _IQ(0.3); // Speed reference (pu)

float32 T = 0.001/ISR_FREQUENCY; // Sampling period (sec), see
parameter.h

Uint16 OffsetFlag = 0;
_iq offsetA = 0;
_iq offsetB = 0;
_iq offsetC = 0;
_iq offsetA_V=0;
_iq offsetB_V=0;
_iq offsetC_V=0;
_iq K1 = _IQ(0.998); //Offset filter coefficient K1: 0.05/(T+0.05);
_iq K2 = _IQ(0.001999); //Offset filter coefficient K2: T/(T+0.05);

Uint32 IsrTicker = 0;
Uint16 BackTicker = 0;
Uint16 TripFlagDMC = 0; //PWM trip status

```

```
volatile Uint16 EnableFlag = FALSE;

Uint16 SpeedLoopPrescaler = 10;    // Speed loop prescaler
Uint16 SpeedLoopCount = 1;        // Speed loop counter

// Instance rotor flux and speed estimations
ACIFE fe1 = ACIFE_DEFAULTS;
ACISE se1 = ACISE_DEFAULTS;

// Instance the constant calculations for rotor flux and speed estimations
ACIFE_CONST fe1_const = ACIFE_CONST_DEFAULTS;
ACISE_CONST se1_const = ACISE_CONST_DEFAULTS;

// Instance a few transform objects (ICLARKE is added in SVGEN module)
CLARKE clarke1 = CLARKE_DEFAULTS, clarke2 = CLARKE_DEFAULTS;

// Instance PI regulators to regulate the d and q axis currents, and speed
PI_CONTROLLER pi_spd = PI_CONTROLLER_DEFAULTS;

// Instance a ramp controller to smoothly ramp the frequency
RMP_CNTL rc1 = RMP_CNTL_DEFAULTS;

// Instance a PWM driver instance
PWMGEN pwm1 = PWMGEN_DEFAULTS;

/// contantes del filtro de orden 32-----
-----
#define FIR_ORDER      32
#if(FIR_ORDER & 0x01) // odd
#define FIR_ORDER_REV  (FIR_ORDER + 1) // even
#else
#define FIR_ORDER_REV  (FIR_ORDER + 2) // even
#endif
FIR16 fir= FIR16_DEFAULTS, fir1= FIR16_DEFAULTS, fir2=FIR16_DEFAULTS;
FIR16 firV= FIR16_DEFAULTS, firV1= FIR16_DEFAULTS, firV2=FIR16_DEFAULTS;

int32_t dbuffer[(FIR_ORDER+3)/2];
int32_t dbuffer1[(FIR_ORDER+3)/2];
int32_t dbuffer2[(FIR_ORDER+3)/2];
int16_t coeff[FIR_ORDER+2];

int16_t revCoeff[FIR_ORDER+2];
int16_t revCoeff1[FIR_ORDER+2];
int16_t revCoeff2[FIR_ORDER+2];
```




```
int32_t dbufferV[(FIR_ORDER+3)/2];
int32_t dbufferV1[(FIR_ORDER+3)/2];
int32_t dbufferV2[(FIR_ORDER+3)/2];

int16_t revCoeffV[FIR_ORDER+2];
int16_t revCoeffV1[FIR_ORDER+2];
int16_t revCoeffV2[FIR_ORDER+2];
///-----

//
// Main
//
void main(void)
{
    uint32_t i = 0;
    //Inicio de reloj configurado 90MHz, registros de contro de las
    //interupciones pie
    InitSysCtrl();
    DINT; //desactiva interupciones globales para iniciar
    InitPieCtrl();//
    memcpy(&RamfuncsRunStart, &RamfuncsLoadStart,
    (Uint32)&RamfuncsLoadSize);
    InitFlash();
    IER = 0x0000;
    IFR = 0x0000;
    InitPieVectTable();
    ///-----
    EALLOW;
    GpioCtrlRegs.GPBPUD.bit.GPIO040 = 1; // Disable pull-up on GPIO040
    (EPWM7A)

    GpioCtrlRegs.GPBMUX1.bit.GPIO040 = 1; // Configure GPIO02 as EPWM2A

    GpioCtrlRegs.GPAMUX1.bit.GPIO012 = 1; // 0=GPIO, 1=TZ1, 2=SCITX-A,
    3=SPISIMO-B

    EDIS;

    while (EnableFlag==FALSE)
    {
        BackTicker++;
    }
}
```



```
// Timing sync for background loops
// Timer period definitions found in device specific
PeripheralHeaderIncludes.h
CpuTimer0Regs.PRD.all = mSec1;    // A tasks
CpuTimer1Regs.PRD.all = mSec5;    // B tasks
CpuTimer2Regs.PRD.all = mSec50;   // C tasks

// Tasks State-machine init
Alpha_State_Ptr = &A0;
A_Task_Ptr = &A1;
B_Task_Ptr = &B1;
C_Task_Ptr = &C1;

ADC_MACRO()

// Initialize the aci flux estimator constants module
fe1_const.Rs = RS;
fe1_const.Rr = RR;
fe1_const.Ls = LS;
fe1_const.Lr = LR;
fe1_const.Lm = LM;
fe1_const.Ib = BASE_CURRENT;
fe1_const.Vb = BASE_VOLTAGE;
fe1_const.Ts = T;
ACIFE_CONST_MACRO(fe1_const)

// Initialize the aci flux estimator module
fe1.K1 = _IQ(fe1_const.K1);
fe1.K2 = _IQ(fe1_const.K2);
fe1.K3 = _IQ(fe1_const.K3);
fe1.K4 = _IQ(fe1_const.K4);
fe1.K5 = _IQ(fe1_const.K5);
fe1.K6 = _IQ(fe1_const.K6);
fe1.K7 = _IQ(fe1_const.K7);
fe1.K8 = _IQ(fe1_const.K8);
fe1.Kp = _IQ(2.8);
fe1.Ki = _IQ(T/0.45);

// Initialize the aci speed estimator constants module
se1_const.Rr = RR;
se1_const.Lr = LR;
se1_const.fb = BASE_FREQ;
se1_const.fc = 3;
se1_const.Ts = T;
ACISE_CONST_MACRO(se1_const)
```

```
// Initialize the aci speed estimator module
se1.K1 = _IQ(se1_const.K1);
se1.K2 = _IQ21(se1_const.K2);
se1.K3 = _IQ(se1_const.K3);
se1.K4 = _IQ(se1_const.K4);
se1.BaseRpm = 120*BASE_FREQ/POLES;

// Initialize the PI module for Id
pi_spd.Kp=_IQ(2.0);
pi_spd.Ki=_IQ(T*SpeedLoopPrescaler/0.5);
pi_spd.Umax =_IQ(0.95);
pi_spd.Umin =_IQ(-0.95);

//calculo onda trianla simetrica para PWM
pwm1.PeriodMax = SYSTEM_FREQUENCY*1000000*T/2; // Prescaler X1 (T1),
ISR period = T x 1
pwm1.HalfPerMax=pwm1.PeriodMax/2;
PWM_INIT_MACRO(pwm1)

for(i = 0; i < FIR_ORDER + 2; i++){
    coeff[i] = FIR16_LPF32_TEST[i];
}
coeff[FIR_ORDER + 1] = 0;

if((FIR_ORDER & 0x01) == 0){
    revCoeff[FIR_ORDER_REV-1] = 0;
    revCoeff1[FIR_ORDER_REV-1] = 0;
    revCoeff2[FIR_ORDER_REV-1] = 0;
    revCoeffV[FIR_ORDER_REV-1] = 0;
    revCoeffV1[FIR_ORDER_REV-1] = 0;
    revCoeffV2[FIR_ORDER_REV-1] = 0;}

for(i = 0; i < FIR_ORDER + 2; i = i + 2){
    revCoeff[FIR_ORDER_REV-i-1] = coeff[i/2+FIR_ORDER_REV/2];
    revCoeff1[FIR_ORDER_REV-i-1] = coeff[i/2+FIR_ORDER_REV/2];
    revCoeff2[FIR_ORDER_REV-i-1] = coeff[i/2+FIR_ORDER_REV/2];
    revCoeff[FIR_ORDER_REV-i-2] = coeff[i/2];
    revCoeff1[FIR_ORDER_REV-i-2] = coeff[i/2];
    revCoeff2[FIR_ORDER_REV-i-2] = coeff[i/2];

    revCoeffV[FIR_ORDER_REV-i-1] = coeff[i/2+FIR_ORDER_REV/2];
    revCoeffV1[FIR_ORDER_REV-i-1] = coeff[i/2+FIR_ORDER_REV/2];
    revCoeffV2[FIR_ORDER_REV-i-1] = coeff[i/2+FIR_ORDER_REV/2];
```

```
    revCoeffV[FIR_ORDER_REV-i-2] = coeff[i/2];
    revCoeffV1[FIR_ORDER_REV-i-2] = coeff[i/2];
    revCoeffV2[FIR_ORDER_REV-i-2] = coeff[i/2];
}

fir.order      = FIR_ORDER;
fir1.order     = FIR_ORDER;
fir2.order     = FIR_ORDER;

firV.order     = FIR_ORDER;
firV1.order    = FIR_ORDER;
firV2.order    = FIR_ORDER;

fir.dbuffer_ptr = &dbuffer[0];
fir1.dbuffer_ptr= &dbuffer1[0];
fir2.dbuffer_ptr= &dbuffer2[0];

firV.dbuffer_ptr = &dbufferV[0];
firV1.dbuffer_ptr= &dbufferV1[0];
firV2.dbuffer_ptr= &dbufferV2[0];

fir.coeff_ptr   = (int32_t *)&revCoeff[0];
fir1.coeff_ptr  = (int32_t *)&revCoeff1[0];
fir2.coeff_ptr  = (int32_t *)&revCoeff2[0];

firV.coeff_ptr  = (int32_t *)&revCoeffV[0];
firV1.coeff_ptr = (int32_t *)&revCoeffV1[0];
firV2.coeff_ptr = (int32_t *)&revCoeffV2[0];

fir.init(&fir);
fir1.init(&fir1);
fir2.init(&fir2);

firV.init(&firV);
firV1.init(&firV1);
firV2.init(&firV2);

HVDMC_Protection();
//

EALLOW; // This is needed to write to EALLOW protected registers
```

```
PieVectTable.ADCINT1 = &OffsetISR;

// Enable PIE group 1 interrupt 1 for ADC1_INT
PieCtrlRegs.PIEIER1.bit.INTx1 = 1;

// Enable EOC interrupt(after the 4th conversion)

AdcRegs.ADCINTOVFLR.bit.ADCINT1=1;
AdcRegs.ADCINTFLGCLR.bit.ADCINT1=1;
AdcRegs.INTSEL1N2.bit.INT1CONT=1;
AdcRegs.INTSEL1N2.bit.INT1SEL=4;
AdcRegs.INTSEL1N2.bit.INT1E=1;
//
// Configure GPIO34 as a GPIO output pin
//
GpioCtrlRegs.GPAMUX1.bit.GPIO0 = 0;
GpioCtrlRegs.GPADIR.bit.GPIO0 = 1;
GpioDataRegs.GPACLEAR.bit.GPIO0 = 1; // uncomment if --> Set Low
initially
GpioCtrlRegs.GPBMUX1.bit.GPIO34 = 0;
GpioCtrlRegs.GPBDIR.bit.GPIO34 = 1;
GpioDataRegs.GPBCLEAR.bit.GPIO34 = 1; // uncomment if --> Set Low
initially
GpioCtrlRegs.GPBMUX1.bit.GPIO39 = 0;
GpioCtrlRegs.GPBDIR.bit.GPIO39 = 1;
GpioDataRegs.GPBCLEAR.bit.GPIO39 = 1; // uncomment if --> Set Low
initially

// Enable CPU INT1 for ADC1_INT:
IER |= M_INT1;

// Enable global Interrupts and higher priority real-time debug events:
EINT; // Enable Global interrupt INTM
ERTM; // Enable Global realtime interrupt DBGEM

EDIS;

for(;;)
{
    // State machine entry & exit point
    //=====
    (*Alpha_State_Ptr)(); // jump to an Alpha state (A0,B0,...)
}
```

```

//=====
}
}

void A0(void)
{
    // loop rate synchronizer for A-tasks
    if(CpuTimer0Regs.TCR.bit.TIF == 1)
    {
        CpuTimer0Regs.TCR.bit.TIF = 1; // clear flag

        //-----
        (*A_Task_Ptr)(); // jump to an A Task (A1,A2,A3,...)
        //-----

        VTimer0[0]++; // virtual timer 0, instance 0 (spare)
        SerialCommsTimer++;
    }

    Alpha_State_Ptr = &B0; // Comment out to allow only A tasks
}

void B0(void)
{
    // loop rate synchronizer for B-tasks
    if(CpuTimer1Regs.TCR.bit.TIF == 1)
    {
        CpuTimer1Regs.TCR.bit.TIF = 1; // clear flag

        //-----
        (*B_Task_Ptr)(); // jump to a B Task (B1,B2,B3,...)
        //-----
        VTimer1[0]++; // virtual timer 1, instance 0 (spare)
    }

    Alpha_State_Ptr = &C0; // Allow C state tasks
}

void C0(void)
{
    // loop rate synchronizer for C-tasks
    if(CpuTimer2Regs.TCR.bit.TIF == 1)
    {
        CpuTimer2Regs.TCR.bit.TIF = 1; // clear flag
    }
}
```

```

//-----
(*C_Task_Ptr)();      // jump to a C Task (C1,C2,C3,...)
//-----
VTimer2[0]++;        //virtual timer 2, instance 0 (spare)
}

Alpha_State_Ptr = &A0; // Back to State A0
}

//=====
// A - TASKS (executed in every 1 msec)
//=====
//-----
void A1(void) // SPARE (not used)
//-----
{
    if(EPwm1Regs.TZFLG.bit.OST==0x1)
        TripFlagDMC=1;    // Trip on DMC (halt and IPM fault trip )

    //-----
    //the next time CpuTimer0 'counter' reaches Period value go to A2
    A_Task_Ptr = &A2;
    //-----
}

//-----
void A2(void) // SPARE (not used)
//-----
{
    //-----
    //the next time CpuTimer0 'counter' reaches Period value go to A3
    A_Task_Ptr = &A3;
    //-----
}

//-----
void A3(void) // SPARE (not used)
//-----
{

```



```
//-----  
//the next time CpuTimer0 'counter' reaches Period value go to A1  
A_Task_Ptr = &A1;  
//-----  
}  
  
//=====
```

=====

```
// B - TASKS (executed in every 5 msec)  
//=====
```

=====

```
//----- USER -----  
-----  
  
//-----  
void B1(void) // Toggle GPIO-00  
//-----  
{  
  
    //-----  
    //the next time CpuTimer1 'counter' reaches Period value go to B2  
    B_Task_Ptr = &B2;  
    //-----  
}  
  
//-----  
void B2(void) // SPARE  
//-----  
{  
  
    //-----  
    //the next time CpuTimer1 'counter' reaches Period value go to B3  
    B_Task_Ptr = &B3;  
    //-----  
}  
  
//-----  
void B3(void) // SPARE  
//-----  
{  
  
    //-----  
    //the next time CpuTimer1 'counter' reaches Period value go to B1  
    B_Task_Ptr = &B1;
```




```
    //-----  
}  
  
//=====   
//=====   
// C - TASKS (executed in every 50 msec)   
//=====   
//=====   
  
//----- USER -----  
-----  
  
//-----  
void C1(void) // Toggle GPIO-34  
//-----  
{  
  
    if(EPwm1Regs.TZFLG.bit.OST==0x1) // TripZ for PWMs is low  
(fault trip)  
    { TripFlagDMC=1;  
      GpioDataRegs.GPBTOGGLE.bit.GPIO39 = 1;  
    }  
  
    if(GpioDataRegs.GPADAT.bit.GPIO12 == 1) // Over Current Prot. for  
Integrated Power Module is high (fault trip)  
    { TripFlagDMC=1;  
      GpioDataRegs.GPBTOGGLE.bit.GPIO34 = 1;  
    }  
  
    //GpioDataRegs.GPBTOGGLE.bit.GPIO34 = 1; // Turn on/off LD3 on the  
controlCARD  
    //-----  
    //the next time CpuTimer2 'counter' reaches Period value go to C2  
    C_Task_Ptr = &C2;  
    //-----  
}  
  
//-----  
void C2(void) // SPARE  
//-----  
{  
  
    //-----
```

```
//the next time CpuTimer2 'counter' reaches Period value go to C3
C_Task_Ptr = &C3;
//-----
}

//-----
void C3(void) // SPARE
//-----
{

//-----
//the next time CpuTimer2 'counter' reaches Period value go to C1
C_Task_Ptr = &C1;
//-----
}

interrupt void MainISR(void)
{

GpioDataRegs.GPATOGGLE.bit.GPIO0 = 1;
// Verifying the ISR
IsrTicker++;
velocidad=_IQ(interu);
// -----
// Connect inputs of the RMP module and call the ramp control macro
// -----
rc1.TargetValue = velocidad;
RC_MACRO(rc1)

#if (BUILDLEVEL==LEVEL1)
// -----
// Measure phase currents, subtract the offset and normalize from (-
0.5,+0.5) to (-1,+1).
// Connect inputs of the CLARKE module and call the clarke transformation
macro
// -----
fir.input = AdcResult.ADCRESULT0; // Q15 format
fir.calc(&fir);
```

```
fir1.input = AdcResult.ADCRESULT1; // Q15 format
fir1.calc(&fir1);

fir2.input = AdcResult.ADCRESULT2; // Q15 format
fir2.calc(&fir2);

clarke1.As = _IQmpy2(_IQ12toIQ(fir.output)-offsetA); // Phase A curr.
clarke1.Bs = _IQmpy2(_IQ12toIQ(fir1.output)-offsetB); // Phase B curr.
clarke1.Cs = _IQmpy2(_IQ12toIQ(fir2.output)-offsetC); // Phase B curr.
CLARKE_MACRO(clarke1)

firV.input = AdcResult.ADCRESULT3; // Q15 format
firV.calc(&firV);

firV1.input = AdcResult.ADCRESULT4; // Q15 format
firV1.calc(&firV1);

firV2.input = AdcResult.ADCRESULT5; // Q15 format
firV2.calc(&firV2);

clarke2.As = _IQmpy2(_IQ12toIQ(firV.output)-offsetA_V); // Phase A
curr.
clarke2.Bs = _IQmpy2(_IQ12toIQ(firV1.output)-offsetB_V); // Phase B
curr.
clarke2.Cs = _IQmpy2(_IQ12toIQ(firV2.output)-offsetC_V); // Phase B
curr.
CLARKE_MACRO(clarke2)

// -----
// Connect inputs of the PI module and call the PID SPD controller macro
// -----

if (SpeedLoopCount==SpeedLoopPrescaler)
{
    pi_spd.Ref = rc1.SetpointValue;
    pi_spd.Fbk = se1.WrHat;
    PI_MACRO(pi_spd)
    SpeedLoopCount=1;
}
else SpeedLoopCount++;

//pi_spd.Out;
```

```
// -----  
// -----  
//   Connect inputs of the ACI module and call the flux estimation macro  
// -----  
// -----  
fe1.UDsS = clarke2.Alpha;  
fe1.UQsS = clarke2.Beta;  
fe1.IDsS = clarke1.Alpha;  
fe1.IQsS = clarke1.Beta;  
ACIFE_MACRO(fe1)  
  
// -----  
// -----  
//   Connect inputs of the ACI module and call the speed estimation macro  
// -----  
// -----  
se1.IDsS  = clarke1.Alpha;  
se1.IQsS  = clarke1.Beta;  
se1.PsiDrS = fe1.PsiDrS;  
se1.PsiQrS = fe1.PsiQrS;  
se1.ThetaFlux = fe1.ThetaFlux;  
ACISE_MACRO(se1)  
#endif // (BUILDLEVEL==LEVEL1)  
// Enable more interrupts from this timer  
//EPwm1Regs.ETCLR.bit.INT = 1;  
AdcRegs.ADCINTFLG.bit.ADCINT1=1;  
  
// Acknowledge interrupt to recieve more interrupts from PIE group 3  
//PieCtrlRegs.PIEACK.all = PIEACK_GROUP3;  
PieCtrlRegs.PIEACK.all = PIEACK_GROUP1;  
  
GpioDataRegs.GPATOGGLE.bit.GPIO0 = 1;  
}  
  
interrupt void OffsetISR(void)  
{  
// Verifying the ISR  
  IsrTicker++;  
  
// DC offset measurement for ADC  
  
  if (IsrTicker>=50)  
  {
```

```
        offsetA=
        _IQmpy(K1,offsetA)+_IQmpy(K2,_IQ12toIQ(AdcResult.ADCRESULT0));
        //Phase A offset
        offsetB=
        _IQmpy(K1,offsetB)+_IQmpy(K2,_IQ12toIQ(AdcResult.ADCRESULT1));
        //Phase B offset
        offsetC=
        _IQmpy(K1,offsetC)+_IQmpy(K2,_IQ12toIQ(AdcResult.ADCRESULT2));
        //Phase C offset

        offsetA_V=
        _IQmpy(K1,offsetA_V)+_IQmpy(K2,_IQ12toIQ(AdcResult.ADCRESULT3));
        //Phase A offset
        offsetB_V=
        _IQmpy(K1,offsetB_V)+_IQmpy(K2,_IQ12toIQ(AdcResult.ADCRESULT4));
        //Phase B offset
        offsetC_V=
        _IQmpy(K1,offsetC_V)+_IQmpy(K2,_IQ12toIQ(AdcResult.ADCRESULT5));
        //Phase C offset

        fir.input  = AdcResult.ADCRESULT0; // Q15 format
        fir.calc(&fir);

        fir1.input  = AdcResult.ADCRESULT1; // Q15 format
        fir1.calc(&fir1);

        fir2.input  = AdcResult.ADCRESULT2; // Q15 format
        fir2.calc(&fir2);

        firV.input  = AdcResult.ADCRESULT3; // Q15 format
        firV.calc(&firV);

        firV1.input  = AdcResult.ADCRESULT4; // Q15 format
        firV1.calc(&firV1);

        firV2.input  = AdcResult.ADCRESULT5; // Q15 format
        firV2.calc(&firV2);

    }

    if (IsrTicker > 3200)
    {
        EALLOW;
        PieVectTable.ADCINT1=&MainISR;
    }
```

```
    EDIS;
}

// Enable more interrupts from this timer
AdcRegs.ADCINTFLG.bit.ADCINT1=1;

// Acknowledge interrupt to recieve more interrupts from PIE group 1
PieCtrlRegs.PIEACK.all = PIEACK_GROUP1;

}
void HVDMC_Protection(void)
{

// Configure Trip Mechanism for the Motor control software
// -Cycle by cycle trip on CPU halt
// -One shot IPM trip zone trip
// These trips need to be repeated for EPWM1 ,2 & 3

//=====
//
//Motor Control Trip Config, EPwm1,2,3
//=====
//

    EALLOW;
// CPU Halt Trip
    EPwm1Regs.TZSEL.bit.CBC6=0x1;

    EPwm1Regs.TZSEL.bit.OSHT1 = 1; //enable TZ1 for OSHT

// What do we want the OST/CBC events to do?
// TZA events can force EPWMxA
// TZB events can force EPWMxB

    EPwm1Regs.TZCTL.bit.TZA = TZ_FORCE_LO; // EPWMxA will go low

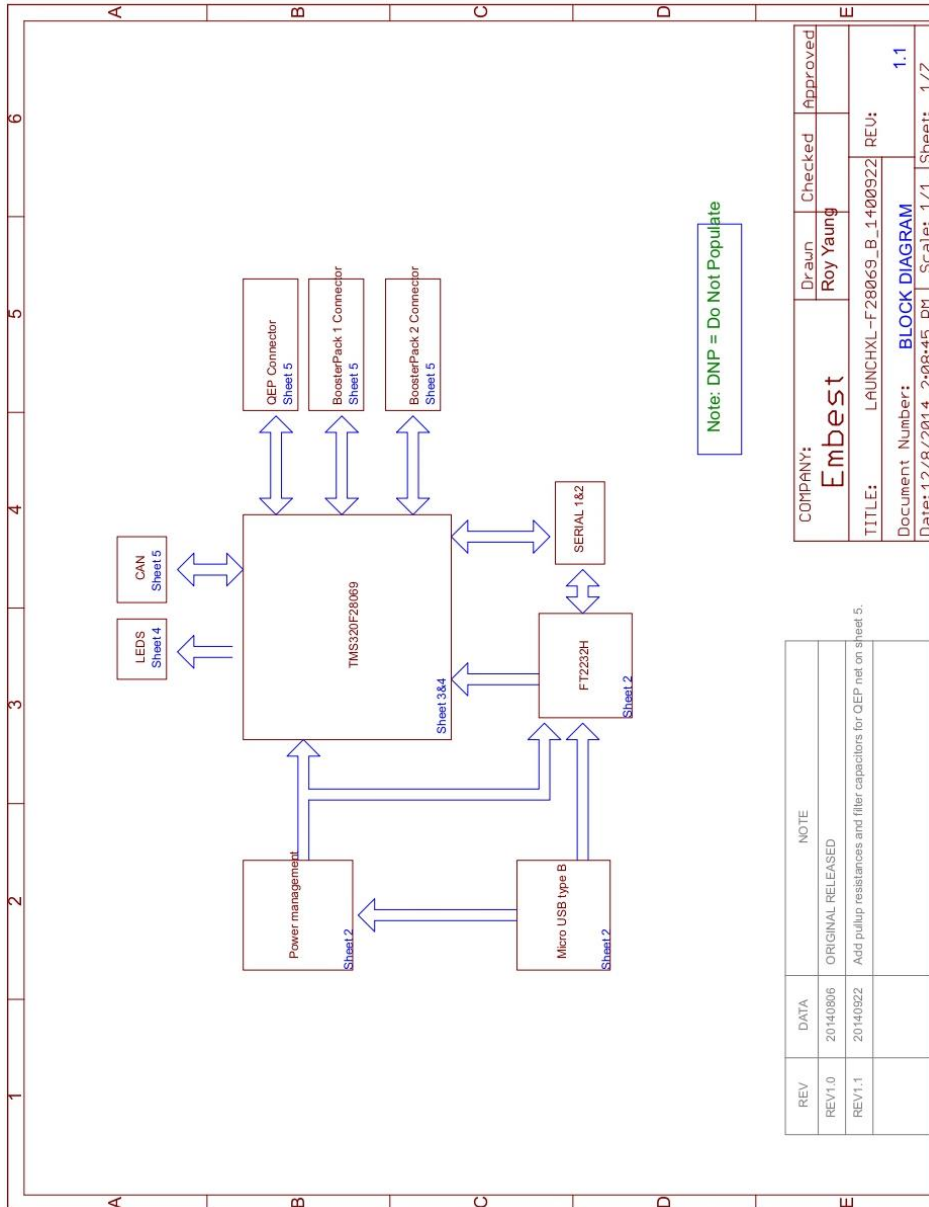
    EDIS;

// Clear any spurious OV trip
    EPwm1Regs.TZCLR.bit.OST = 1;

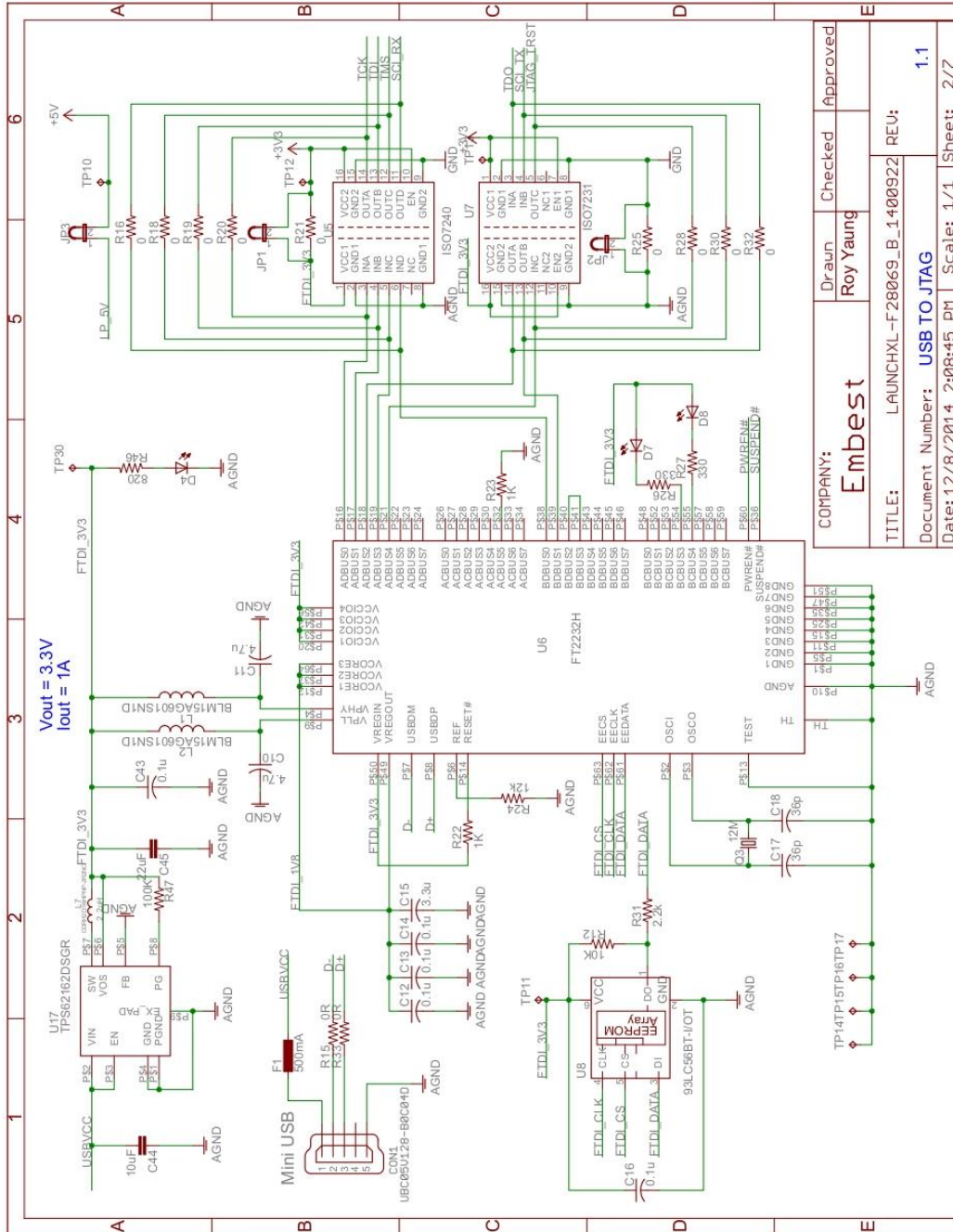
//***** End of Prot. Conf.
//*****//
}

```

Anexo 3. Diagrama en bloques LAUNCHXL-F28069[12].

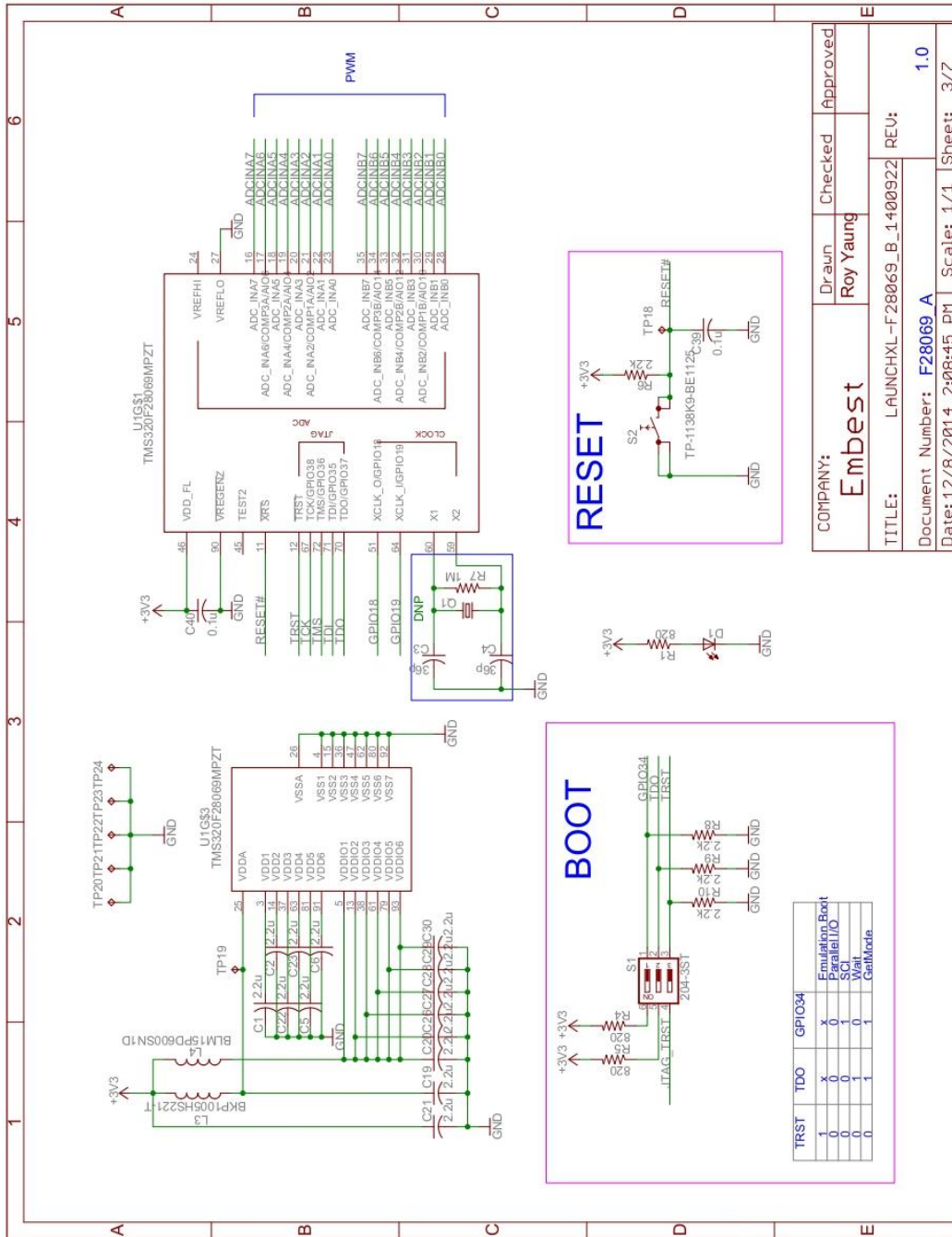


Anexo 4. USB JATAG LAUNCHXL-F28069[12].



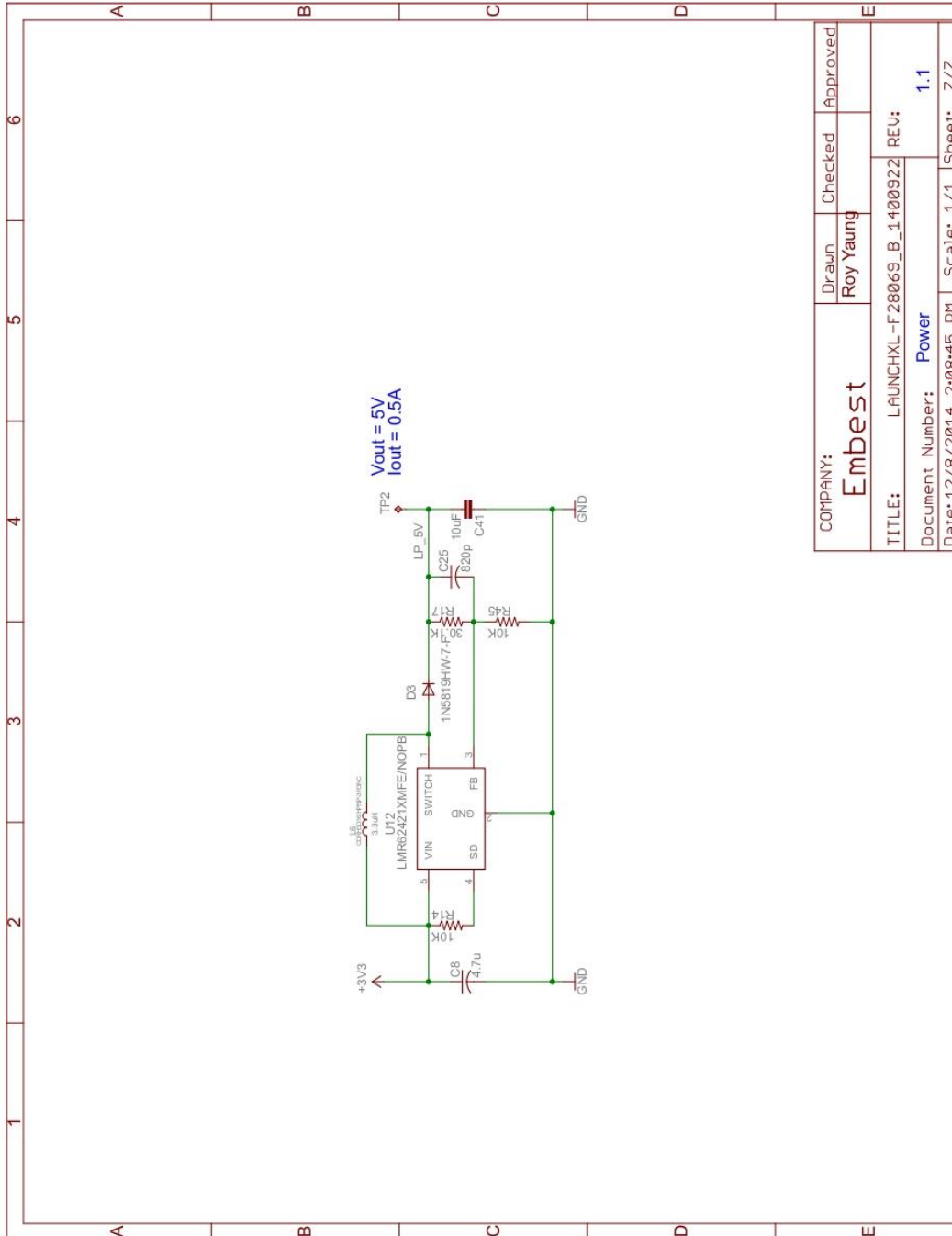
COMPANY:	Embest
Drawn	Roy Young
Checked	Approved
TITLE:	LAUNCHXL-F28069_B_1400922 REV:
Document Number:	USB TO JTAG
Date:	12/8/2014 2:08:45 PM
Scale:	1/1
Sheet:	2/7

Anexo 5. TMS320F28069M PART_A[12].



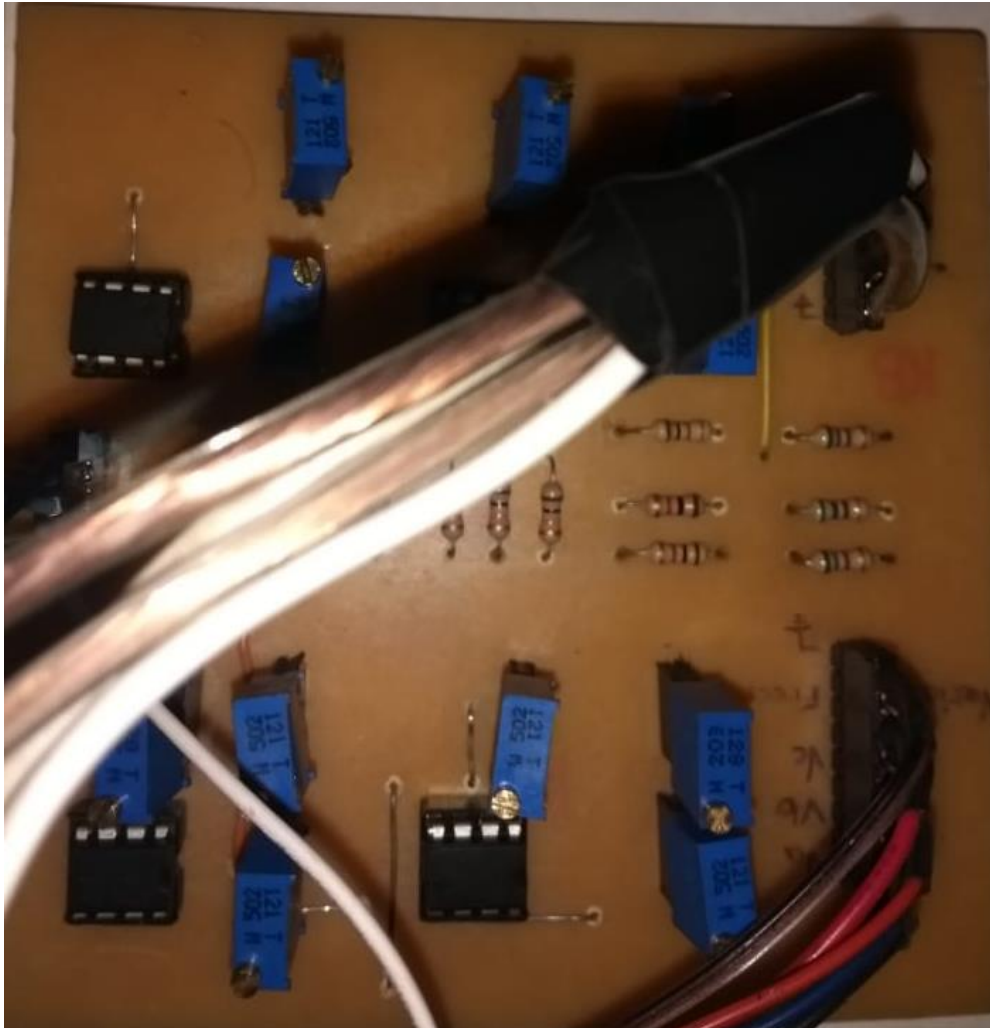
COMPANY:	Embest	Drawn	Checked	Approved
		Roy Young		
TITLE:	LAUNCHXL-F28069_B_1.400922 REV:			
Document Number:	F28069_A	Scale:	1/1	Sheet: 3/7
REV:	1.0			

ANEXO 9. PODER[12].



COMPANY:	Embest	Drawn	Checked	Approved
TITLE:	LAUNCHXL-F28069_B_1400922	REU:		
Document Number:	Power	REU:	1.1	
Date:	12/8/2014 2:08:45 PM	Scale:	1/1	Sheet: 7/7

ANEXO 10 CIRCUITO ESQUEMATICO SENSORICO Y DE AMPLIFICACION DE SEÑAL PWM_DAC PARA CONTROLAR DEL INVERSOR



ANEXO 11 Motor de inducción seleccionado



Tipo AL - 306			
3 ~ Mot.		Nr. A190044	
Δ -Y	220 - 380	V	2,5 - 1,4 A
0,6 Kw		cos φ	0,8
1.450		/min	60 Hz.
Rot. Y	220	V	A
Aisl Cl F	P. 22		