



UNIVERSIDAD CARLOS III DE MADRID

Departamento de Sistemas y Automática

Ingeniero Técnico Industrial: Electrónica Industrial

PROYECTO FIN DE CARRERA

**Desarrollo de un controlador para motores
DC brushless basado en CompactRIO y
LabVIEW de National Instruments para el
estudio de nuevos algoritmos de control.**

Autor: Juan Miguel García Haro

Tutor: Dr. José María Armingol Moreno
Dr. Roberto Ponticelli Lima

Leganés, Noviembre de 2011

Título: Desarrollo de un controlador para motores DC brushless basado en CompactRIO y LabVIEW de National Instruments para el estudio de nuevos algoritmos de control.

Autor: Juan Miguel García Haro

Tutor: Dr. José María Armingol Moreno
Dr. Roberto Ponticelli Lima

EL TRIBUNAL

Presidente: _____

Vocal: _____

Secretario: _____

Realizado el acto de defensa y lectura del Proyecto Fin de Carrera el día 14 de Noviembre de 2011 en Leganés, en la Escuela Politécnica Superior de la Universidad Carlos III de Madrid, acuerda otorgarle la CALIFICACIÓN de

VOCAL

SECRETARIO

PRESIDENTE

A mi familia, dedido este proyecto por la cual he podido alcanzar y cumplir mi objetivo de tener mis estudios de Ingeniería, y que gracias a su constante ayuda, incluso en los momentos más difíciles, he conseguido cumplir mis sueños.

Gracias.

Índice

	Página
1. Introducción.	1
1.1. Historia del control automático.	1
1.2. Centro de Automática y Robótica.	2
1.3. Motivaciones para el proyecto.	3
1.4. Fases de desarrollo.	4
1.5. Esquema de la memoria.	4
2. Objetivos.	6
3. Marco teórico.	8
3.1. National Instruments y su entorno de programación LabVIEW	8
3.1.1. Introducción.	8
3.1.2. Instrumentos de Medida y Automatización basados en PC.	8
3.1.3. Software.	8
3.1.4. Instrumentos.	9
3.1.5. Adquisición de datos.	9
3.1.6. Adquisición de imágenes y control de ejes.	9
3.1.7. Redes industriales.	9
3.1.8. El punto fuerte.	9
3.1.9. Páginas Web desarrolladas por National Instruments.	10
3.1.10. NI LabVIEW.	11
3.1.10.1. ¿Qué es NI LabVIEW?	11
3.1.10.2. Introducción.	11
3.1.10.3. Historia sobre la búsqueda de un lenguaje de programación.	11
3.1.10.4. LabVIEW: Programación gráfica de flujo de datos.	12
3.1.10.5. Beneficios de la programación G.	13
3.2. Motores eléctricos – Motores DC Brushless.	20
3.2.1. Introducción.	20
3.2.2. Ventajas e inconvenientes.	20
3.2.3. Tipo de motores eléctricos.	21
3.2.3.1. Motores de corriente continua.	21
3.2.3.2. Motores de corriente alterna.	22
3.2.4. Motores DC brushless.	23
3.2.4.1. Introducción.	23
3.2.4.2. Topología de un motor DC brushless.	25
3.2.4.3. Sensores de posición de efecto Hall.	26
3.2.4.4. Principio de funcionamiento.	28
3.2.4.5. Control en bucle cerrado.	31
3.2.4.6. Características par/velocidad.	32
3.3. Drivers de potencia	33
3.3.1. Topología del inversor.	33
3.3.2. Inversor trifásico.	34
3.3.2.1. Conducción a 180°.	35
3.3.2.2. Conducción a 120°.	37
3.3.3. Tecnología PWM - modulación de ancho de pulso.	39

3.4. Sistemas de control automático.	44
3.4.1. Introducción.	44
3.4.2. Tipos de sistemas de control.	44
3.4.2.1. Sistemas de Control Clásico.	44
3.4.2.2. Sistemas de Control Moderno.	44
3.4.3. Controlador.	45
3.4.4. Acciones de control de un regulador o controlador.	45
3.4.4.1. Controlador de acción Proporcional (P).	46
3.4.4.2. Controlador de acción Integral (I).	47
3.4.4.3. Controlador de acción proporcional e integral (PI).	47
3.4.4.4. Controlador de acción proporcional y derivativa (PD).	48
3.4.4.5. Controlador de acción PID.	49
3.4.5. Controlador.	50
3.4.5.1. Introducción.	50
3.4.5.2. Arquitectura de la CompactRIO.	51
4. Proyecto.	53
4.1. Primera fase: aprendizaje de LabVIEW.	53
4.1.1. Cursos de LabVIEW - manual teórico y práctico.	53
4.1.1.1. LabVIEW Core 1. Temario y objetivos.	54
4.1.1.2. LabVIEW Core 1. Temario y objetivos.	54
4.1.1.3. LabVIEW FPGA. Temario y objetivos.	55
4.2. Segunda fase: desarrollo del controlador de motor BLDC basado en CompactRIO para el estudio de nuevos algoritmos de control.	55
4.2.1. Planteamiento de las bases del controlador de motor BLDC basado en CompactRIO.	55
4.2.2. Desarrollo de la Placa PCB.	56
4.2.2.1. Inversor trifásico.	57
4.2.2.2. Regulador de 12 voltios.	61
4.2.2.3. Regulador de 5 voltios.	61
4.2.2.4. Conectores.	62
4.2.3. Programa de control desarrollado con el lenguaje G dentro del entorno de desarrollo LabVIEW de National Instruments.	64
4.2.3.1. Código Main.vi.	64
4.2.3.2. Código FPGA PID PWM control.vi.	67
4.2.3.3. Código Discrete PID.vi.	72
5. Experimentación y resultados.	73
5.1. Introducción.	73
5.2. Método de Oscilación de Ziegler y Nichols.	74
5.3. Experimentos sin carga.	75
5.4. Experimentos con carga.	76
6. Presupuesto.	79
7. Conclusiones, trabajos futuros y recomendaciones.	85
8. Bibliografía.	87
Anexos.	

Índice de figuras.	Página
Figura 3.1. Un bucle <i>While</i> en G se representa por un lazo gráfico que se ejecuta hasta que se cumple una condición de parada.	12
Figura 3.2. Los datos se originan en la función de adquisición y luego fluyen a las funciones de análisis y almacenamiento a través de los cables.	13
Figura 3.3. La ejecución Highlight provee de una forma intuitiva de entender el orden de ejecución del código G.	14
Figura 3.4. La barra de herramientas del diagrama de bloques ofrece acceso a las herramientas de depuración estándar como la ejecución paso a paso.	14
Figura 3.5. Las sondas son formas eficientes de ver los valores moviéndose por los cables de la aplicación, incluso para secciones paralelas del código.	14
Figura 3.6. La flecha rota de ejecución provee de información inmediata de errores sintácticos en el código G.	15
Figura 3.7. La lista de errores muestra una explicación detallada de cada error sintáctico en la jerarquía completa del código.	15
Figura 3.8. El patrón de diseño Productor/Consumidor de LabVIEW es usado para aumentar el rendimiento de las aplicaciones que requieren tareas paralelas.	16
Figura 3.9. El código de LabVIEW FPGA con paralelismo genera caminos independientes en el chip de la FPGA.	17
Figura 3.10. Formula Node usa sintaxis similar a C para representar expresiones matemáticas de una forma sucinta en formato de texto.	18
Figura 3.11. Con MathScript Node, usted puede crear o reutilizar archivos .m para el procesamiento de señales y el análisis de datos.	18
Figura 3.12. Topología de un motor DC <i>brushless</i> .	25
Figura 3.13. Aspecto de un motor de corriente continua sin escobillas trifásico.	26
Figura 3.14 Distribución de los sensores de efecto <i>Hall</i> de un motor DC <i>brushless</i> .	28
Figura 3.15. Relación gráfica de la fuerza contraelectromotriz, el par motor y la corriente de fase, en función de las entradas de los sensores de efecto <i>Hall</i> .	29
Figura 3.16. Secuencia de activación de los devanados en función de las entradas de los sensores de efecto <i>Hall</i> .	30
Figura 3.17. Ejemplo genérico de la relación par motor/velocidad.	32
Figura 3.18. Topología de Inversores de una, dos y tres ramas.	33

Figura 3.19. Inversor trifásico con dos niveles de tensión.	35
Figura 3.20. Secuencia de la señales de excitación de los transistores a 180°.	36
Figura 3.21. Distintas configuraciones del inversor en función del estado de los interruptores, para una conducción de 180°.	37
Figura 3.22. Secuencia de la señales de excitación de los transistores a 120°.	37
Figura 3.23. Distintas configuraciones del inversor en función del estado de los interruptores, para una conducción de 120°.	38
Figura 3.24. Esquema de modulación de ancho de pulso.	39
Figura 3.25. Contenido de armónicos del esquema PWM.	40
Figura 3.26. Zonas de operación del esquema PWM.	41
Figura 3.27. Contenido de armónicos del esquema PWM, para operación en onda cuadrada.	42
Figura 3.28. Forma de onda de voltaje en Inversor trifásico con esquema PWM.	43
Figura 3.29. Controlador de acción Proporcional (P).	46
Figura 3.30. Controlador de acción Proporcional-Integral (PI).	47
Figura 3.31. Controlador de acción Proporcional-Derivativo (PD).	48
Figura 3.32. Controlador de acción Proporcional-Integral-Derivativo (PID).	49
Figura 3.33. Componentes de la arquitectura de la CompactRIO: controlador Real-Time cRIO-9022, fuente alimentación NI PS-15 Power Supply, chasis reconfigurable cRIO-9113, módulo I/O digital y bidireccional High-Speed NI 9401.	52
Figura 4.1. Control de lazo cerrado a través de un PID.	56
Figura 4.2. Esquema eléctrico del inversor trifásico.	57
Figura 4.3. Gráfica de cálculo del tiempo muerto en función de las resistencias HDEL y LDEL.	58
Figura 4.4. Circuito equivalente en el proceso de carga y descarga del CBS.	59
Figura 4.5. Esquema eléctrico del driver del MOSFET.	60
Figura 4.6. Esquema eléctrico del regulador de 12 voltios.	61
Figura 4.7. Esquema eléctrico del regulador de 5 voltios.	62

Figura 4.8. Jerarquía del programa de control.	64
Figura 4.9. Bucle “Lazo de control PID”.	68
Figura 4.10. Bucle “generación PWM”.	69
Figura 4.11. Bucle “máquina de estados”.	70
Figura 4.12. Cálculo del sentido de giro de la velocidad real.	70
Figura 4.13. Contador de pulsos.	71
Figura 4.14. Bucle “calculo de velocidad”.	71
Figura 4.15. Tareas que ejecuta el código discreto PID.vi.	72
Figura 5.1. Lazo cerrado solo con ganancia proporcional.	74
Figura 5.2. Esquema físico del <i>driver</i> de potencia – motor DC <i>brushless</i> .	75
Figura 5.3. Gráfica del experimento de escalones de subida y bajada.	76
Figura 5.4. Esquema físico del <i>driver</i> de potencia – motor DC <i>brushless</i> - carga.	77
Figura 5.5. Gráfica del experimento de perturbación de impulso.	77
Figura 5.6. Gráfica del experimento de perturbación de escalón.	78

Índice de tablas.	Página
Tabla 3.1. Diferencias entre motores <i>brushless</i> y motores <i>brushed</i> .	24
Tabla 3.2. Diferencias entre motores <i>brushless</i> y motores AC inducción.	24
Tabla 3.3. Activación de los drivers del motor según el orden de los sensores <i>Hall</i> , para la configuración de giro horario.	30
Tabla 3.4. Activación de los drivers del motor según el orden de los sensores <i>Hall</i> , para la configuración de giro anti horario.	30
Tabla 3.5. Tensiones de cada rama respecto al punto intermedio de la alimentación 'O' y entre dos fases, en función del estado de los interruptores, para una conducción de 180°.	36
Tabla 3.6. Tensiones de cada rama respecto al punto intermedio de la alimentación 'O' y entre dos fases, en función del estado de los interruptores, para una conducción de 120°.	38
Tabla 4.1. Descripción del conexionado del jumper de 8 pines entre la placa PCB y el motor DCBL.	62
Tabla 4.2. Descripción del conexionado del conector de 15 pines entre la placa PCB y los dos módulos NI 9401 de National Instruments.	63
Tabla 4.3. Descripción de las variables usadas en el programa.	66
Tabla 5.1. Variables prefijadas para su posterior aplicación en los experimentos.	73
Tabla 5.2. Parámetros de ajuste (método de oscilación).	74

Capítulo 1. Introducción.

1.1. Historia del control automático.

El Control Automático ha tenido un rápido desarrollo en los últimos años. Sin embargo a lo largo del tiempo ha sufrido un proceso de evolución que se inició, aproximadamente, en el año 300 a.C., cuando los griegos empezaron a tener la necesidad de medir empíricamente el tiempo, y que ha continuado con etapas tan importantes como lo fueron la Revolución Industrial y las Guerras Mundiales.

Cuando la máquina de vapor fue inventada y se dio inicio a la Revolución Industrial, se inició paralelamente la necesidad de realizar un sistema de control para poder manipular los diferentes parámetros de esta máquina. Un ejemplo de esto fue el desarrollo del regulador de presión con el fin de controlar este parámetro en el sistema.

Tanto en la Primera como en la Segunda Guerra Mundial, la necesidad de implementar diferentes controles para barcos, aviones y demás, motivó y, a su vez, provocó que diferentes personas se vieran involucradas en el desarrollo de los mismos, dándose en estas etapas un gran aporte al desarrollo del Control Automático.

Desde el año de 1950, cuando empieza la era espacial y de la informática, y hasta nuestros días, con el desarrollo tecnológico en las diferentes áreas, el Control Automático es algo con lo que se convive día a día, y haciendo que la vida de cada persona sea más fácil. Conforme las plantas modernas con muchas entradas y salidas se vuelven más complejas, la descripción de un sistema de control moderno requiere una gran cantidad de ecuaciones. La teoría de control clásica que trata de los sistemas con una entrada y una salida, pierde su solidez ante sistemas con entradas y salidas múltiples.

Desde 1960, debido a que la disponibilidad de las computadoras digitales hizo posible el análisis en el dominio del tiempo de sistemas complejos, la teoría de control moderna, basada en el análisis en el dominio del tiempo y la síntesis a partir de variables de estados, se ha desarrollado para enfrentar la creciente complejidad de las plantas modernas y los requerimientos limitativos respecto a la precisión, el peso y el coste en aplicaciones militares, espaciales e industriales. De 1980 hasta la fecha, los descubrimientos en la teoría de control moderna se centran en el control robusto, el control de H_∞ y temas asociados.

Ahora que las computadoras digitales se han vuelto más baratas y más compactas, se usan como parte integral de los sistemas de control. Las aplicaciones recientes de la teoría de control moderna incluyen sistemas ajenos a la ingeniería, como los biológicos y los económicos [1].

1.2. Centro de Automática y Robótica.

Por un acuerdo entre la Universidad Politécnica de Madrid (UPM) y el Consejo Superior de Investigaciones Científicas (CSIC) se ha creado el Centro de Automática y Robótica (CAR). El Centro de Automática y Robótica se concibe como un centro de investigación en Automática y Robótica cuya estrategia se sustenta en dos grandes pilares:

- Adquirir conocimientos científicos y tecnológicos al más alto nivel, en el campo de la automatización, cultivando líneas de investigación científica acordes con las prioridades marcadas por el Programa Marco de la Unión Europea y el Plan Nacional de Investigación.
- Transferir dicha capacidad a la sociedad mediante la implicación del centro en proyectos de innovación. Es decir, participar, realizando tareas de investigación científica y tecnológica, en proyectos de I+D, en colaboración con otros organismos y/o empresas, para resolver todos los problemas de desarrollo, industrialización y comercialización que toda auténtica innovación comporta.

El Departamento de Control Automático del Centro de Automática y Robótica es en el que se ha desarrollado este proyecto, y tiene como objetivo fundamental investigar acerca del análisis y el diseño de sistemas de control. Las actividades del departamento se centran en la realización de investigaciones teóricas, en la formación de personal científico en el área del control automático, y en la elaboración y ejecución de proyectos de investigación que actualmente se focalizan de manera significativa en la resolución de los problemas que presentan las nuevas generaciones de robots.

Las líneas de investigación son:

- Teoría de control
- Robots para medios hostiles: robots caminantes y manipuladores específicos
- Control de accionamientos electromecánicos

La investigación que se realiza en el departamento está organizada dinámicamente en proyectos, cada uno de los cuales tienen unos objetivos determinados, requiriendo normalmente una dedicación de tres a diez hombres/año. Se pretende no aislar la ejecución de cada proyecto sino ligar unos a otros en cada momento, bajo las líneas temáticas principales y a su vez en el tiempo, buscando por una parte, la profundización en la adquisición de los conocimientos y, por otra, la obtención de resultados tecnológicos para la industria y la sociedad en general.

De este modo de trabajo se distinguen tres tipos de proyectos:

- Proyectos de investigación básica orientada:

Se trata de investigaciones cuya finalidad principal es la generación de nuevos conocimientos que sitúen a los investigadores al más alto nivel en materias avanzadas y que necesariamente presenten un interés para la sociedad a medio o largo plazo. La ejecución se lleva a cabo esencialmente por el centro, y a veces con la colaboración de otras instituciones.

- Proyectos de investigación aplicada:

El interés de estos otros trabajos se centra en el dominio de las tecnologías básicas necesarias para la resolución de un problema concreto llegando normalmente a la realización de prototipos de las soluciones elaboradas.

- Proyectos de desarrollo tecnológico:

Estos proyectos a corto plazo tienen como objetivo último la obtención de un producto, o en su caso de un proceso, nuevo o mejorado que permita situar a una o más empresas en unas condiciones mejores de mercado.

1.3. Motivaciones para el proyecto.

En el Departamento de Control Automático del Centro de Automática y Robótica se está desarrollando un proyecto conjunto aplicado a las nuevas generaciones de robots, donde se investiga la posibilidad de añadir nuevas tecnologías en el desarrollo de estos robots.

Durante mucho tiempo en el centro se ha empleado motores DC con escobillas para la mayoría de sus diseños robóticos. Pero recientemente, gracias a los nuevos avances tecnológicos y la reducción de costos, los motores *brushless* se están convirtiendo en una opción rentable que aportan beneficios tanto en relación par-motor/tamaño como en relación par-motor /precio.

Por otro lado el Departamento de Control Automático ha contratado algunos de los instrumentos y servicios de la empresa National Instruments. En particular, el Departamento ha comprado un controlador de automatización programable (PAC) NI CompactRIO. Éste es un sistema reconfigurable de control y adquisición de bajo coste diseñado para aplicaciones que requieren alto rendimiento y fiabilidad. El sistema combina una arquitectura embebida abierta con un tamaño pequeño, extrema robustez y módulos industriales de E/S intercambiables en vivo. CompactRIO es alimentado por la tecnología de arreglo de compuertas programables en campo (FPGA) de E/S reconfigurable (RIO).

Con la idea de mejorar la tecnología y aplicarla en la industria, surge este proyecto, en el que se pretende diseñar e implementar un controlador de motores DC *brushless* mediante el uso de CompactRIO y LabVIEW de National Instruments como estudio de nuevos algoritmos de control.

1.4. Fases de desarrollo.

Para poder desarrollar este proyecto se han tenido que seguir unos pasos estrictos:

- En la primera etapa se ha tenido que estudiar manuales de uso del programa de entorno gráfico LabVIEW.
- En la segunda etapa se ha definido las características básicas del programa de control. Para ellos se ha concretado que objetos se iban a usar (CompactRIO, módulos E/S, motor DCBL,...) para que se agilizase el proceso de elaboración del código del programa de control.
- En la tercera etapa se ha desarrollado el programa de control. Teniendo en cuenta las distintas características que definen el controlador, se ha diseñado un código de programa que sea capaz de controlar la velocidad del motor en función de la velocidad deseada por el usuario.
- En la cuarta y última etapa, se ha construido una base de datos donde se puede comprobar cómo el sistema en conjunto es capaz de funcionar y en qué condiciones puede funcionar.

1.5. Esquema de la memoria.

La memoria de este proyecto se divide en siete partes, sin contar los anexos. Estos capítulos están ordenados de tal forma que se facilite la tarea al lector.

- En el capítulo dos se explica detenidamente cuáles son los objetivos principales y los subobjetivos que se intentan conseguir en este proyecto.
- En el capítulo tres se definen los contenidos teóricos básicos para, posteriormente, entender cómo se ha elaborado el proyecto y porqué. En este capítulo se describe el entorno de programación gráfica LabVIEW, el funcionamiento de los motores DC *brushless*, la forma de operar de los inversores trifásico como *driver* de potencia del motor DCBL, y la arquitectura de la CompactRIO incluyendo los diferentes componentes que la estructuran.
- En el capítulo cuatro se comenta el contenido principal del proyecto. En él, se explica cómo se ha ido desarrollando todas las partes que constituyen el proyecto. Primero se hace el estudio de los manuales de uso del entorno de programación gráfica LabVIEW. Segundo se puntualizan las características básicas del proyecto para crear una base sobre la que desarrollar la placa PBC y el programa de control. Tercero se

diseña y se construye la placa PCB que suministra potencia al motor DCBL entre otras funciones. Y cuarta y última se desarrolla el programa de control a través del entorno de programación gráfica LabVIEW, destinado a comprobar que es posible hacer funcionar el controlador a través de la elaboración del banco de pruebas.

- En el capítulo cinco se exponen los resultados obtenidos en el banco de pruebas, definiendo los cuatro experimentos aplicados al controlador (tanto sin carga como con carga) y comentando los resultados obtenidos de los propios experimentos.
- En el capítulo seis se explica con detalle que componentes se han usado y el precio que tienen, dando una valoración final del presupuesto total. Para comentar el presupuesto, éste se ha dividido en cinco secciones claramente diferenciadas. La primera sección se explica el desglose presupuestario del personal. La segunda sección se detalla el *hardware* utilizado. En la tercera sección se desarrolla los costes del *software*. En la cuarta sección se explica qué componentes eléctricos se han utilizado para construir la placa PCB. Por último, en la quinta sección se hace un resumen global del presupuesto de todas las secciones y se expone un presupuesto final.
- En el capítulo siete y último se explican las conclusiones de este proyecto, a las que se han llegado a través de los resultados de la experimentación. Además se proponen algunas propuestas para mejorar la capacidad del sistema global, de tal manera que en los proyectos futuros se trabaje de una manera más eficiente.

Capítulo 2. Objetivos.

Este proyecto que el CAR inició hace pocos años tiene como objetivo principal el estudio y desarrollo de nuevas tecnologías en el campo de actuación y control automático, que servirá de base para otras futuras investigaciones dentro del centro.

La tecnología a la que se hace mención se refiere al control de actuadores basados en motores DC *brushless* (BLDC *Motors*) empleando el sistema de hardware embebido CompactRIO y programación LabVIEW de National Instruments.

Tradicionalmente se emplea en el CAR motores DC con escobillas para la mayoría de sus diseños robóticos. Recientemente, gracias a los nuevos avances tecnológicos y la reducción de costos, los motores *brushless* se están convirtiendo en una opción rentable que aportan beneficios tanto en relación par-motor/tamaño como en relación par-motor/precio. Por otro lado, las tarjetas electrónicas de control antiguas que aún se emplean en el CAR no ofrecen las ventajas y versatilidad de los nuevos sistemas embebidos. Es por ello que el centro estudia la integración del sistema embebido CompactRIO, desarrollado por National Instruments, con los robots desarrollados en el CAR. Entre otras ventajas, este sistema embebido permite ser reprogramado, tanto en software (programa que ejecuta una CPU) como en hardware (reconfiguración de una FPGA), permitiendo que el actuador pueda ser empleado prácticamente en cualquier tipo de aplicación en el campo de la automática y la robótica.

Como se acaba de comentar, el CSIC estudia la posibilidad de incorporar estas nuevas tecnologías en sus proyectos futuros. Para ello, desarrollaremos un proyecto, que como objetivo principal, consiste en diseñar e implementar un controlador de motores DC *brushless* mediante el uso de CompactRIO y LabVIEW de National Instruments.

Dividiremos este proyecto en cuatro partes. De estas cuatro partes, las tres últimas nos definirán a su vez los tres objetivos principales, los cuales se explican a continuación:

1. Aprendizaje y desarrollo de un nivel alto de programación del lenguaje G que se necesita para elaborar cualquier tipo de programa a través del entorno de programación grafica llamado LabVIEW, proporcionado por National Instruments. El curso de aprendizaje consiste en estudiar tres manuales de usuario suministrados por la propia empresa National Instruments. En los dos primeros se explica con detenimiento todas las herramientas y aplicaciones que posee LabVIEW (LabVIEW Core 1 y LabVIEW Core 2). En el tercero se explica cómo saber utilizar las herramientas y aplicaciones específicas de la FPGA que incorpora la CompactRIO (LabVIEW FPGA).

2. Desarrollo de un programa específico para el control de motores DC *brushless*. Este programa a su vez se divide en dos subobjetivos. El primero es el diseño del control de velocidad en lazo cerrado. El programa consiste en una sensorización y procesamiento de datos, tanto de la posición del motor a través de unos sensores de efecto *hall*, como de otras entradas auxiliares, que permiten medir la velocidad y posición del motor. A partir de estos valores se implementa un lazo cerrado de control empleando un módulo regulador (un PID). El segundo es la implementación de la máquina de estados para el control de conmutación, de tal forma que finalmente se generen las consignas de actuación codificadas en señales moduladas en ancho de pulso (PWM). Estas señales de salida activan las puertas de un inversor trifásico basado en transistores MOSFET que proporciona potencia al motor.

Se utiliza como *hardware* el controlador de automatización programable CompactRIO y como *software* se utiliza el entorno de desarrollo gráfico LabVIEW, ambos suministrados por National Instruments.

3. Diseño y posterior fabricación de una etapa de suministro de potencia para un motor DC *brushless*. Se denomina comúnmente *driver*, que en este caso consiste en un conjunto de transistores organizados en configuración de puente H triple o inversor trifásico (de tres ramas). El motor DC *brushless* es un motor eléctrico que no emplea escobillas para realizar el cambio de polaridad de los electroimanes dentro del rotor, requiriendo que la conmutación (cambio de polaridad) sea realizada de manera externa al motor. El motor que usaremos en el proyecto contiene tres fases (tres bobinados), las cuales se activan o se desactivan sincronizadamente para mover el motor. Para activar estas tres fases también se construye un puente H triple que controle el suministro de energía a cada bobinado.

Además la etapa de potencia cuenta a su vez con una etapa de acondicionamiento de señales que deberán activar las entradas a las puertas del puente H triple, señales que provendrán de la CompactRIO a través de sus módulo E/S. con un regulador de 12V que suministre energía a las demás partes de la placa PCB y un regulador de 5V que alimenta los sensores *Hall*.

4. Por último, desarrollo de pruebas de control y experimentación sobre el motor aplicado a casos reales. Se le incorpora al motor DC *brushless* en un banco de pruebas y se le somete a distintos experimentos para calcular factores de trabajo.

Capítulo 3. Marco teórico.

3.1. National Instruments y su entorno de programación Labview.

3.1.1. Introducción.

Desde su fundación en 1976, National Instruments ha revolucionado la forma en que trabajan los técnicos e investigadores de todo el mundo. Esta empresa desarrolla y fabrica una enorme variedad de productos *software* y *hardware* integrados que, utilizados junto al ordenador estándar, permiten sustituir o integrar instrumentos tradicionales para monitorizar y controlar procesos. Como empresa líder en la producción de instrumentos de Medida y Automatización basados en PC, National Instruments propone soluciones innovadoras para ayudar a los clientes a lograr sus objetivos de la forma más rápida y eficaz posible, y manteniendo unos costes reducidos [19].

3.1.2. Instrumentos de Medida y Automatización basados en PC.

National Instruments une a su propia experiencia en el campo de la Medida las tecnologías existentes más avanzadas para crear productos que garantizan mejores prestaciones, un aumento de la productividad y una reducción de los costes del sistema. Los clientes de National Instruments utilizan estos productos para llevar a cabo aplicaciones en el ámbito de la investigación, el desarrollo, la producción y el control, en muy diversos sectores industriales tales como las telecomunicaciones, el sector aeroespacial, electrónico, químico o farmacéutico.

National Instruments es una empresa líder en el ámbito de los instrumentos de Medida y Automatización y como tal tiene como objetivo garantizar una mejora continua y una innovación constante en sus propios productos. Para ver la lista completa de instrumentos basados en PC y de los servicios que se ofrecen, se puede consultar el catálogo online en la dirección ni.com, en el que los productos están divididos en cinco categorías: *software*, instrumentos, adquisición de datos, adquisición y control de imágenes y redes industriales.

3.1.3. Software.

Gracias al *software* National Instruments, técnicos e investigadores pueden aumentar la productividad y reducir los costes. Las herramientas de desarrollo que ofrece la empresa, entre los que destaca su producto estrella LabVIEW™, son los instrumentos más utilizados en el ámbito industrial, como lo demuestra el hecho de que LabVIEW ha sido el producto más nominado como "Test Product of the Year" por los lectores de Test & Measurement World.

3.1.4. Instrumentos.

En 1976, National Instruments lanzó su primer producto, GPIB, una tarjeta de interfaz que permite conectar instrumentos tradicionales al PC. Desde ese momento la empresa ha continuado ampliando y mejorando su gama de productos hasta el desarrollo de instrumentos de altas prestaciones basados en PC o modulares como PXI™/CompactPCI y VXI.

3.1.5. Adquisición de datos.

Aprovechando la potencia de los ordenadores personales, el *hardware* de National Instruments para Adquisición de Datos y Acondicionamiento de Señal permite adquirir y analizar datos físicos tales como la temperatura, la presión y la vibración. La combinación del *hardware* y *software* de National Instruments para Adquisición de Datos genera los llamados "instrumentos virtuales" que sustituyen a los tradicionales ofreciendo costes más reducidos y soluciones flexibles.

3.1.6. Adquisición de imágenes y control de ejes.

En 1997 National Instruments amplió su línea de productos para Adquisición de Datos, incluyendo nuevos productos para adquisición de imágenes y control de ejes. También en este caso, el *software* y *hardware* de National Instruments simplifican el desarrollo de soluciones integradas para todos los clientes que operan en esos sectores.

3.1.7. Redes Industriales.

National Instruments continúa reafirmando en el sector de la automatización industrial ofreciendo a sus clientes tarjetas de interfaz basadas en PC para redes industriales, FOUNDATION Fieldbus, CAN, DeviceNet y PROFIBUS. Gracias a estas tarjetas y al *software* para Automatización de National Instruments (Lookout™, OPC Server™ y LabVIEW Datalogging and Supervisory Control™ - un nuevo módulo de LabVIEW), los usuarios pueden monitorizar y controlar sus propias instalaciones de forma eficaz.

3.1.8. El punto fuerte.

El ordenador personal ha enriquecido nuestra forma de trabajar. Desde el ámbito financiero al académico, el PC garantiza un acceso rápido a cualquier información en cualquier parte del mundo y permite utilizar instrumentos *hardware* y *software* para aumentar la productividad.

El objetivo de National Instruments es ofrecer a sus clientes los instrumentos *hardware* y *software* que necesitan para satisfacer sus exigencias de medida y automatización. Ya sea con ordenadores portátiles estándar para controlar laboratorios en un entorno hostil o con PC's modulares especiales para la monitorización de materiales de gran importancia y valor, National Instruments ofrece la solución de medida y control más adecuada.

Por todo ello podemos afirmar que el PC constituye la base de nuestra estrategia. Todos los productos National Instruments, de un modo u otro, aprovechan las mejoras continuas en términos de velocidad y prestaciones de la industria informática, lo que se traduce, desde el punto de vista del cliente, en mejores resultados con un coste inferior.

3.1.9. Páginas Web desarrolladas por National Instruments.

- **Página Web National Instruments.**

- ni.com

Con más de un millón de visitas al trimestre, ni.com es una de las páginas preferidas por los técnicos e investigadores de todo el mundo. Desde esta página se puede acceder a información detallada sobre todos los productos *hardware* y *software* de National Instruments. La tienda online ofrece tablas de configuración que ayudan a los clientes a elegir y pedir los cables o accesorios adecuados, mientras los diversos links ayudan a los usuarios en la aplicación de soluciones completas para Medida y Automatización basadas en PC.

- **NI Developer Zone**

- zone.ni.com

Los técnicos e investigadores de todo el mundo pueden unirse a un grupo online de expertos que le ayudará a proyectar y llevar a cabo sus sistemas de Medida y Automatización. Los visitantes de NI Developer Zone podrán participar en discusiones sobre newsgroup, acceder a informaciones técnicas y efectuar investigaciones de productos según características específicas.

National Instruments ha desarrollado NI Developer Zone en colaboración con Prentice Hall Professional Technical Reference (PTR), una editorial académica y técnica líder en su sector, y con GlobalSpec.com, especializada en el desarrollo de instrumentos paramétricos de investigación online. Esta página ofrece más de 1.200 programas de ejemplo, 1.600 instrumentos driver, 500 tutoriales y presentaciones técnicas e instrumentos avanzados de selección para los otros 180.000 productos que ofrecen más de 450 empresas productoras de sensores y componentes para la automatización.

- **MyNI**

- my.ni.com

National Instruments ofrece una herramienta para personalizar los contenidos de la página ni.com de acuerdo a las informaciones que resulten más interesantes dentro del campo de la Medida y Automatización. Es posible crear una página web que contenga novedades, información sobre servicios y productos según las preferencias personales y recibir por correo electrónico noticias personalizadas sobre National Instruments y sus productos, además de sugerencias y soporte técnico. Los usuarios de MyNI pueden además investigar dentro de diversas áreas del sitio web y visualizar, de acuerdo con el área geográfica donde residan, el nombre y dirección de la sede de National Instruments más cercana y el calendario de eventos programados para su zona.

3.1.10. NI LabVIEW.

3.1.10.1. ¿Qué es NI LabVIEW?

LabVIEW es un entorno de programación gráfica usado por miles de ingenieros e investigadores para desarrollar sistemas sofisticados de medida, pruebas y control usando íconos gráficos e intuitivos y cables que parecen un diagrama de flujo. Ofrece una integración incomparable con miles de dispositivos de *hardware* y brinda cientos de bibliotecas integradas para análisis avanzado y visualización de datos, todo para crear instrumentación virtual. La plataforma LabVIEW es escalable a través de múltiples objetivos y sistemas operativos, desde su introducción en 1986 se ha vuelto un líder en la industria.

3.1.10.2. Introducción.

Durante más de 20 años, NI LabVIEW se ha sido utilizado por millones de ingenieros y científicos para desarrollar test sofisticados y aplicaciones de medida y control. Además de que LabVIEW provee de una variada gama de características y herramientas de asistentes e interfaces de usuario configurables, se diferencia por ser un lenguaje de programación gráfico de propósito general (conocido como G), con su compilador asociado, su enlazador, y herramientas de depuración.

3.1.10.3. Historia sobre la búsqueda de un lenguaje de programación.

Para entender mejor el valor añadido de la programación gráfica de LabVIEW, es útil remontarse al primer lenguaje de programación de alto nivel. En los albores de la edad moderna de la computación a mediados de los 50, un reducido grupo de IBM decidió crear una alternativa práctica a la programación de la enorme unidad central IBM 704 (un supercomputador en su época) en lenguaje ensamblador, el más moderno disponible en aquel entonces. El resultado fue FORTRAN, un lenguaje de programación más legible cuyo propósito era acelerar el proceso de desarrollo.

La comunidad ingenieril fue, en principio, escéptica de que este método pudiese superar los programas desarrollados a mano en ensamblador, pero pronto se demostró que los programas hechos con FORTRAN se ejecutaban casi tan eficientemente como aquellos escritos en ensamblador. Al mismo tiempo, FORTRAN redujo el número de sentencias necesarias en un programa en un factor 20, por lo que es considerado a menudo el primer lenguaje de desarrollo de alto nivel. No sorprende que FORTRAN ganase rápidamente la aceptación de la comunidad científica.

Cincuenta años más tarde, hay todavía importantes lecciones en esta anécdota. Primero, durante más de 50 años, los ingenieros han buscado formas más fáciles y rápidas de solucionar sus problemas de programación. Después, los lenguajes de programación elegidos para traducir sus tareas han tendido hacia niveles mayores de abstracción. Estas lecciones ayudan a explicar la inmensa popularidad y la extensa adopción de G desde su aparición en 1986; G representa un lenguaje de programación de extremadamente alto nivel cuyo propósito es aumentar la productividad de sus usuarios ejecutándose a casi la misma velocidad que los lenguajes de programación de niveles inferiores como FORTRAN, C y C++.

3.1.10.4. LabVIEW: Programación gráfica de flujo de datos.

LabVIEW es diferente de la mayoría de lenguajes de propósito general principalmente en dos vertientes. Primero, la programación G se desarrolla cableando iconos gráficos en un diagrama que compila directamente a código máquina de modo que los procesadores del ordenador pueden ejecutarlo sin preocuparse por obtener problemas de compatibilidad.

Aunque se representa gráficamente en lugar de texto, G contiene los mismos conceptos de programación que se pueden encontrar en la mayoría de los lenguajes tradicionales. Por ejemplo, G incluye todas las construcciones estándar tales como tipos de datos, bucles, eventos, variables, recursividad y programación orientada a objetos.

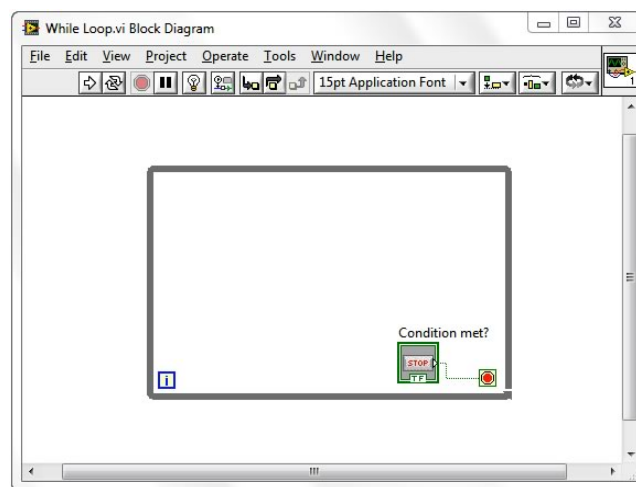


Figura 3.1. Un bucle *While* en G se representa por un lazo gráfico que se ejecuta hasta que se cumple una condición de parada.

La segunda diferencia principal es que el código G desarrollado en LabVIEW se ejecuta de acuerdo con las reglas del flujo de datos en lugar del acercamiento más tradicional (en otros palabras, una serie secuencial de comandos para ser llevados a cabo) que se encuentran en la mayoría de los lenguajes de programación basados en texto como C y C++. Los lenguajes de flujo de datos como G (también VEE de Agilent, Microsoft Visual y Apple Quartz Composer) promueven los datos como concepto principal detrás de cualquier programa. La ejecución de un datagrama es dirigida por el dato o dependiente del mismo. El flujo de datos entre los nodos del programa, líneas no secuenciales de texto determina el orden de ejecución.

Esta distinción puede ser menor a priori, pero el impacto es extraordinario ya que presenta rutas de datos entre partes del programa para ser el centro de atención del desarrollador. Los nodos en un programa de LabVIEW (en otras palabras, funciones y estructuras como bucles y subrutinas) tienen entradas, procesan datos y generan salidas. Una vez que todas las entradas de los nodos dados contienen un dato válido, el nodo ejecuta su lógica, produce datos de salida y pasa los datos al siguiente nodo en la secuencia del flujograma. Un nodo que recibe datos de otro, se puede ejecutar solo después de que el primero complete su ejecución.

3.1.10.5. Beneficios de la programación G.

- **Programación gráfica intuitiva.**

Como todo el mundo, los ingenieros y científicos aprenden observando y procesando imágenes sin necesidad de pensamiento consciente. Se denominan “pensadores visuales”, ya que son especialmente adeptos a organizar información con procesamiento visual. En otras palabras, piensan mejor en imágenes. Esto se refuerza a menudo en facultades y universidades donde se anima a los estudiantes a modelar soluciones a problemas como diagramas de proceso. Sin embargo, la mayoría de los lenguajes de programación de propósito general requieren el empleo de cantidades ingentes de tiempo en aprender la sintaxis necesaria asociada con el lenguaje y mapear la estructura del mismo al problema a solventar. La programación gráfica con G provee de una experiencia más intuitiva.

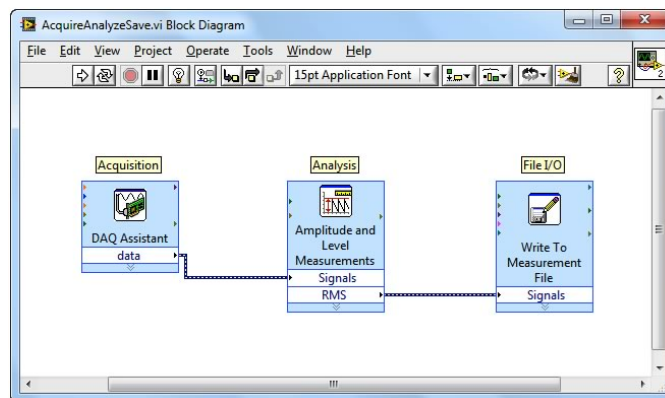


Figura 3.2. Los datos se originan en la función de adquisición y luego fluyen a las funciones de análisis y almacenamiento a través de los cables.

El código G es más sencillo de entender por ingenieros y científicos porque están familiarizados con la visualización y la modelización gráfica de procesos y tareas en términos de diagramas de bloque y flujogramas (que siguen también las reglas del flujo de datos). Además, ya que los lenguajes de flujo de datos requieren basar la estructura del programa en el propio flujo, a usted se le anima a pensar en términos del problema que quiere solucionar. Por ejemplo, un programa típico en G puede adquirir, en primer lugar, de varios canales de datos de temperatura, después pasarlos a una función de análisis y finalmente escribirlos a disco. En conjunto, el flujo de datos y los pasos involucrados en este programa son sencillos de comprender en el diagrama de LabVIEW.

- **Herramientas de depuración interactiva**

Puesto que el código G de LabVIEW es sencillo de comprender, las tareas más comunes de programación como el depurado, se vuelven más intuitivas también. Por ejemplo, LabVIEW provee de herramientas de depuración únicas que se puede usar para observar el movimiento interactivo de los datos por los cables de un programa de LabVIEW y ver los valores que pasan de una función a otra (conocido en el entorno de LabVIEW como ejecución hightlighting).

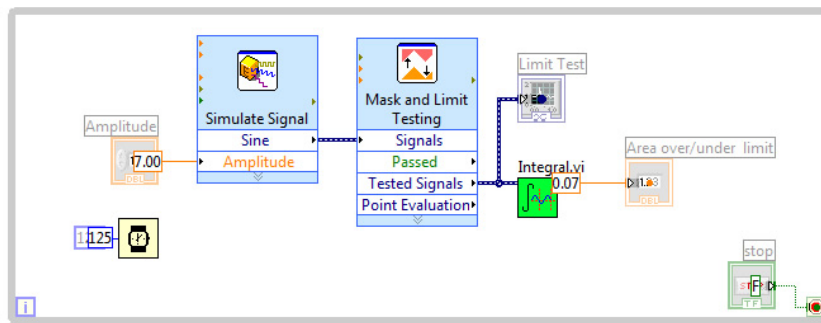


Figura 3.3. La ejecución Highlight provee de una forma intuitiva de entender el orden de ejecución del código G.

LabVIEW también ofrece herramientas de depuración para G comparables con aquellas que se encuentran en los lenguajes tradicionales. Estas herramienta, accesibles desde la barra de herramientas de un diagrama, incluyen sondas, puntos de parada y ejecución paso a paso.



Figura 3.4. La barra de herramientas del diagrama de bloques ofrece acceso a las herramientas de depuración estándar como la ejecución paso a paso.

Con las herramientas de depuración de G, se puede sondear los datos en muchas partes del programa simultáneamente, pausar la ejecución, y ejecutar paso a paso una subrutina sin programación compleja. Aunque esto es posible en otros lenguajes de programación, es más fácil visualizar el estado del programa y la relación entre las partes paralelas del código (que son comunes en G por su naturaleza gráfica).

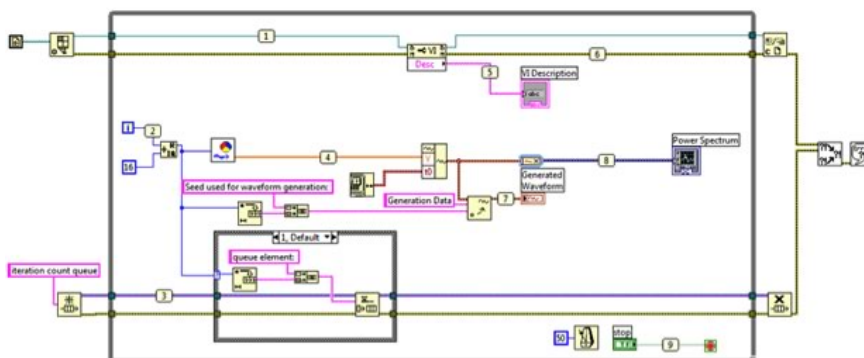


Figura 3.5. Las sondas son formas eficientes de ver los valores moviéndose por los cables de la aplicación, incluso para secciones paralelas del código.

Una de las más comunes herramientas de depuración usadas en LabVIEW se encuentra en el compilador. Mientras se está desarrollando un programa, el compilador continuamente busca errores y provee de información semántica y sintáctica de la aplicación. Si existe un error, no se puede ejecutar el programa, sólo se ve un botón de ejecución roto en la barra de herramientas.



Figura 3.6. La flecha rota de ejecución provee de información inmediata de errores sintácticos en el código G.

Presionando el botón roto de ejecución, se abre una lista con los problemas que se han de arreglar. Una vez hecho, el compilador de LabVIEW transforma el programa en código máquina. Una vez compilado, el rendimiento de los programas de G es comparable al de aquellos basados en texto como C.

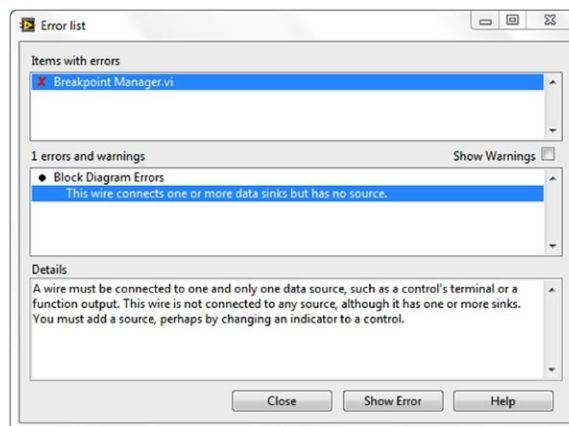


Figura 3.7. La lista de errores muestra una explicación detallada de cada error sintáctico en la jerarquía completa del código.

- **Paralelismo y rendimiento automáticos**

Los lenguajes de flujo de datos como LabVIEW permiten paralelizar automáticamente. A diferencia de los lenguajes secuenciales como C y C++, los programas gráficos contienen de forma inherente información sobre qué partes del código se pueden ejecutar en paralelo. Por ejemplo, un patrón común de diseño en G es el Productor/Consumidor, en el que dos bucles *while* se ejecutan independientemente: el primero es el responsable de la producción de datos y el segundo del procesamiento. En la ejecución en paralelo (posiblemente a frecuencias diferentes) los datos pasan entre los bucles usando colas, que son estructuras de datos estándar en los lenguajes de programación de propósito general.

El paralelismo es importante en la programación ya que desbloquea las ganancias de rendimiento relativas a los programas secuenciales debido a cambios recientes en el diseño de los procesadores. Durante más de 40 años, los fabricantes de chips incrementaron la velocidad del reloj del procesador para aumentar el rendimiento. Hoy, en cambio, el aumento de las velocidades de reloj para obtener mejoras en el rendimiento no es viable debido al consumo de energía y a las restricciones de disipación de calor. Como resultado, los fabricantes de chips han diseñado nuevas arquitecturas con múltiples núcleos de procesamiento en un único chip.

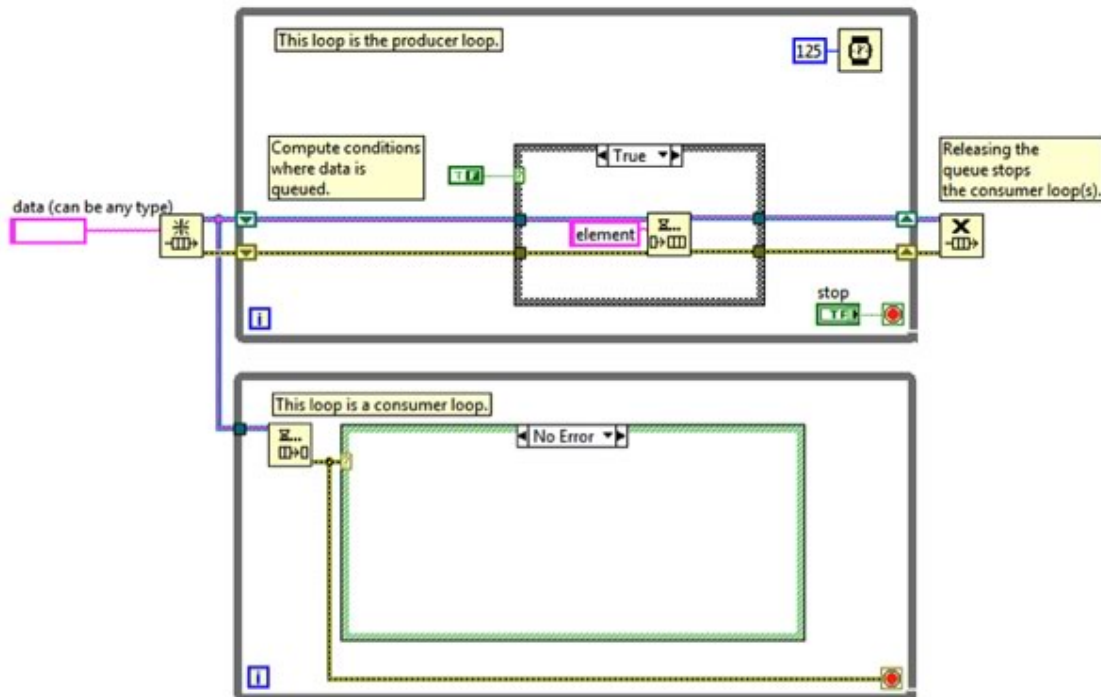


Figura 3.8. El patrón de diseño Productor/Consumidor de LabVIEW es a menudo usado para aumentar el rendimiento de las aplicaciones que requieren tareas paralelas.

Para sacar provecho al rendimiento disponible en procesadores multinúcleo, se ha de usar el multihilo en sus aplicaciones (en otras palabras, dividir las aplicaciones en secciones discretas que puedan ser ejecutadas de forma independiente). Si se emplea los tradicionales lenguajes basados en texto, debe crear y administrar hilos para implementar el paralelismo, un desafío de envergadura para programadores no expertos.

Por el contrario, la naturaleza paralela del código G hace a la multitarea y multihilo fáciles de implementar. El compilador trabaja continuamente para identificar secciones paralelas del código. Siempre que el código G tiene una rama en un cable o una secuencia paralela de nodos en un diagrama, el compilador intenta ejecutar el código en paralelo del conjunto de hilos administrados por LabVIEW. En términos de computación científica, esto se conoce como paralelismo implícito porque no se tiene que escribir el código con el propósito de la ejecución paralela, el lenguaje G se encarga de ello por su cuenta.

Más allá del multihilo en un sistema multinúcleo, G provee de mayor ejecución en paralelo extendiendo la programación gráfica a las FPGAs. Éstas son chips reprogramables de silicio de naturaleza paralela – con cada tarea de procesamiento independiente asignado a una sección del chip – pero sin estar limitadas por el número de núcleos disponibles. Como resultado, el rendimiento de una parte de la aplicación no se ve afectado al añadir más procesamiento.

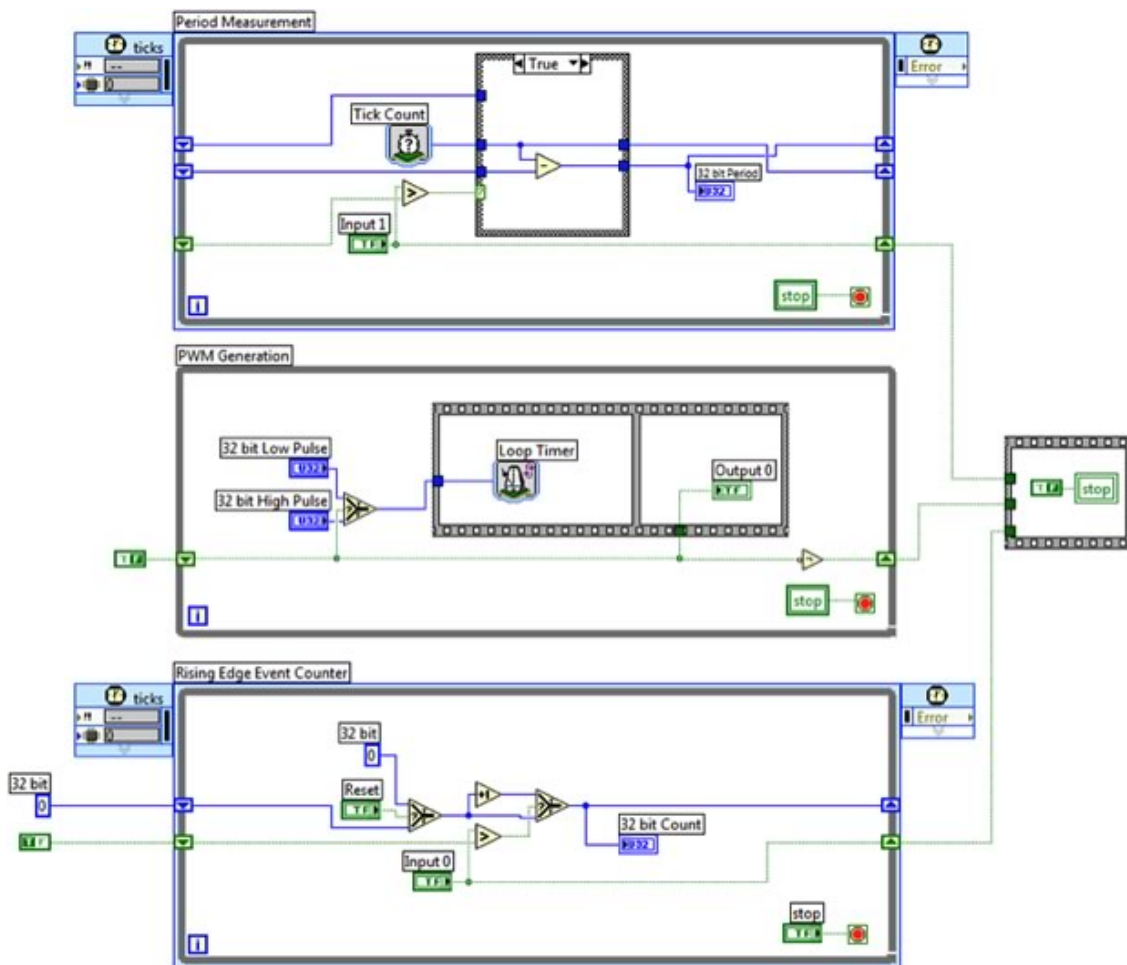


Figura 3.9. El código de LabVIEW FPGA con paralelismo genera caminos independientes en el chip de la FPGA.

Históricamente, la programación en FPGA eran el campo de sólo expertos formados con un profundo conocimiento de los lenguajes de diseño *hardware*. Aumentan los ingenieros no expertos en FPGA que quieren usar el *hardware* personalizado de la FPGA para retinas únicas de temporización y disparo, control ultra rápido, protocolos digitales, procesamiento digital de la señal (DSP), RF y comunicaciones y muchas otras aplicaciones que requieren *hardware* de alta velocidad, fiabilidad, personalización y alto determinismo. G encaja perfectamente con la programación de las FPGAs ya que claramente representa el paralelismo y el flujo de datos y está creciendo rápidamente en popularidad como herramienta para desarrolladores que buscan procesamiento paralelo y ejecución determinista.

- **Combinando G con otros lenguajes**

Aunque el código G provee de una representación excelente para el paralelismo y elimina el requisito de los desarrolladores de entender y administrar el uso de memoria, no es necesariamente ideal para todas las tareas. En particular, las fórmulas matemáticas y las ecuaciones pueden ser representadas sucintamente en texto. Por esa razón, se puede usar LabVIEW para combinar la programación gráfica con varias formas de programación en texto.

```

int32 sp = 0;

// initialize stack
// which contains a pair of index
stack[sp++] = 0;
stack[sp++] = sizeofDim(numArr,0) - 1;

// as long as stack is not empty
// continue calculation
while(sp)
{
  int32 p, r, j, i;
  float f;

  // take beginning and ending
  // index off the stack
  p = stack[sp - 2];

```

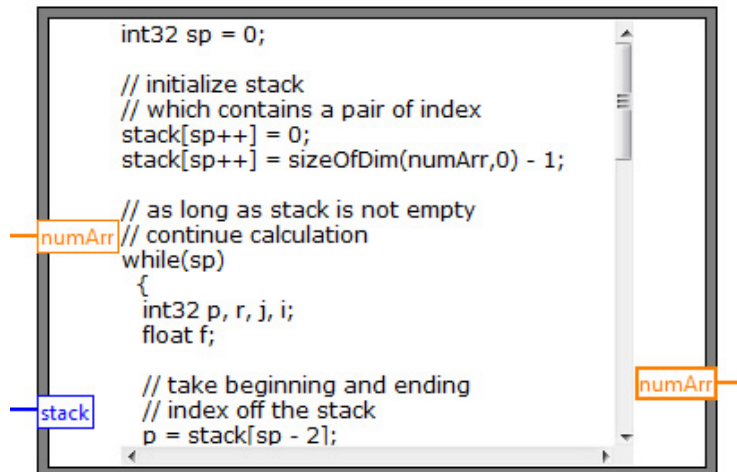


Figura 3.10. Formula Node usa sintaxis similar a C para representar expresiones matemáticas de una forma sucinta en formato de texto.

Trabajando con LabVIEW, se puede elegir un enfoque textual, gráfico o combinado. Por ejemplo, LabVIEW contiene el concepto de Formula Node que evalúa las fórmulas y expresiones matemáticas de un modo similar a C en el diagrama de bloques. Estas fórmulas matemáticas se pueden ejecutar codo con codo e integrarlas en el código gráfico de LabVIEW.

Igualmente, el MathScript Node añade programación textual matemática en LabVIEW, generalmente compatible con la sintaxis de los archivos .m.

```

1  %Vibration Analysis
2  timerstart;
3
4  for ii=1:1:1000
5      Limit_High(ii) = 5.0;
6      Limit_Low(ii) = -5.0;
7  end
8
9  Vib_total = sqrt(Vib_X.^2 + Vib_Y.^2);
10 limit = Vib_total > Limit_High;
11
12 fft_x = fft(Vib_X);
13 fft_x = fft_x(1:end/2);
14 fft_y = fft(Vib_Y);
15
16 fft_y = fft_y(1:end/2);
17 fft_total(1,:) = abs(fft_x);
18 fft_total(2,:) = abs(fft_y);
19
20 timer = timerstop;

```

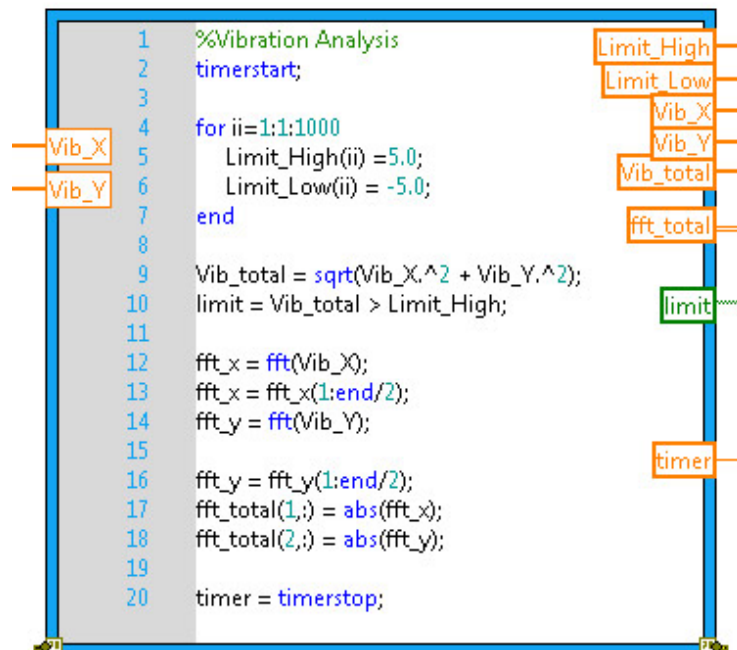


Figura 3.11. Con MathScript Node, usted puede crear o reutilizar archivos .m para el procesamiento de señales y el análisis de datos.



- **Una manera mejor de solucionar problemas**

LabVIEW y su lenguaje de programación gráfico de flujo de datos provee de una mejor manera de solucionar los problemas que las alternativas tradicionales de bajo nivel y la prueba está en su longevidad. Las claves diferenciadoras de la programación en G son el código gráfico intuitivo que se crea y las reglas de movimiento de los datos que gobiernan la ejecución que se combinan para ofrecer una experiencia de programación que expresa el pensamiento de los procesos de sus usuarios de forma más cercana que otros lenguajes. A pesar de que G es un lenguaje de alto nivel, se puede lograr rendimientos comparables a los de los lenguajes como C gracias al compilador de LabVIEW.

3.2. Motores Eléctricos – Motores DC *Brushless*.

3.2.1. Introducción.

Faraday, Michael (1791–1867), fue el que descubrió el principio del motor eléctrico, él descubrió la inducción. Inducción es la generación de una corriente eléctrica en un conductor en movimiento en el interior de un campo magnético físico. A partir de ese descubrimiento se potencio el estudio sobre la electrónica. Para calcular la inducción magnética se tiene que aplicar la Ley de Faraday-Lenz (ecuación 3.1.), la cual relaciona la fuerza electromotriz (FEM) inducida sobre un conjunto de espiras con la variación del flujo magnético.

$$\varepsilon = -N \bullet \frac{d\phi}{dt} \quad \text{Ecuación 3.1}$$

Donde:

- ε : tensión inducida (V).
- N: número de espiras.
- Φ : flujo del campo magnético (Wb).

Los motores eléctricos son máquinas eléctricas rotatorias que transforman la energía eléctrica en energía mecánica. Debido a sus múltiples ventajas, entre las que cabe citar su economía, limpieza, comodidad y seguridad de funcionamiento, el motor eléctrico ha reemplazado en gran parte a otras fuentes de energía, tanto en la industria como en el transporte, las minas, el comercio, o el hogar.

Los motores eléctricos satisfacen una amplia gama de necesidades de servicio, desde arrancar, acelerar, mover, o frenar, hasta sostener y detener una carga. Estos motores se fabrican en función de la potencia que se necesita suministrar y que varían desde una pequeña fracción de caballo hasta varios miles, y con una amplia variedad de velocidades, que pueden ser fijas, ajustables o variables.

3.2.2. Ventajas e inconvenientes.

Un motor eléctrico contiene un número mucho más pequeño de piezas mecánicas que un motor de combustión interna o uno de una máquina de vapor, por lo que es menos propenso a los fallos. Los motores eléctricos son los más ágiles de todo a lo que respecta a variación de potencia y pueden pasar instantáneamente desde la posición de reposo a la de funcionamiento al máximo. Su tamaño es más reducido y pueden desarrollarse sistemas para manejar las ruedas desde un único motor, como en los automóviles. Como los motores eléctricos usan energía eléctrica, se les pueden considerar actuadores limpios, que apenas contaminan, si los comparamos con motores térmicos de combustión u otros actuadores como los neumáticos o los hidráulicos.

El inconveniente es que las baterías son los únicos sistemas de almacenamiento de electricidad, y ocupan mucho espacio. Además, cuando se gastan, necesitan varias horas para recargarse antes de poder funcionar otra vez, mientras que en el caso de un motor de combustión interna basta sólo con llenar el depósito de combustible. Este problema se soluciona, en el ferrocarril, tendiendo un cable por encima de la vía, que va conectado a las plantas de generación de energía eléctrica. La locomotora obtiene la corriente del cable por medio de una pieza metálica llamada patín. Así, los sistemas de almacenamiento de electricidad no son necesarios.

Cuando no es posible o no resulta rentable tender la línea eléctrica, para encontrar una solución al problema del almacenamiento de la energía se utilizan sistemas combinados, que consisten en el uso de un motor de combustión interna o uno de máquina de vapor conectado a un generador eléctrico. Este generador proporciona energía a los motores eléctricos situados en las ruedas. Estos sistemas, dada su facilidad de control, son ampliamente utilizados no sólo en locomotoras, sino también en barcos.

El uso de los motores eléctricos se ha generalizado a todos los campos de la actividad humana desde que sustituyeran en la mayoría de sus aplicaciones a las máquinas de vapor. Existen motores eléctricos de las más variadas dimensiones, desde los pequeños motores fraccionarios empleados en pequeños instrumentos hasta potentes sistemas que generan miles de caballos de fuerza, como los de las grandes locomotoras eléctricas.

3.2.3. Tipo de motores eléctricos.

En cuanto a los tipos de motores eléctricos genéricamente se distinguen motores monofásicos, que contienen un juego simple de bobinas en el estator, y polifásicos, que mantienen dos, tres o más conjuntos de bobinas dispuestas en círculo [12].

Según la naturaleza de la corriente eléctrica transformada, los motores eléctricos se clasifican en motores de corriente continua, también denominada directa, y motores de corriente alterna, que a su vez, se agrupan, según su sistema de funcionamiento, en motores de inducción y motores sincrónicos. Tanto unos como otros disponen de todos los elementos comunes a las máquinas rotativas electromagnéticas

3.2.3.1. Motores de corriente continua.

La conversión de energía en un motor eléctrico se debe a la interacción entre una corriente eléctrica y un campo magnético. Un campo magnético, que se forma entre los dos polos opuestos de un imán, es una región donde se ejerce una fuerza sobre determinados metales o sobre otros campos magnéticos. Un motor eléctrico aprovecha este tipo de fuerza para hacer girar un eje, transformándose así la energía eléctrica en movimiento mecánico.

Los dos componentes básicos de todo motor eléctrico son el rotor y el estator. El rotor es una pieza giratoria, un electroimán móvil, con varios salientes laterales, que llevan cada uno a su alrededor un bobinado por el que pasa la corriente eléctrica. El estator, situado alrededor del rotor, es un electroimán fijo, cubierto con un aislante. Al igual que el rotor, dispone de una serie de salientes con bobinados eléctricos por los que circula la corriente.

Cuando se introduce una espira de hilo de cobre en un campo magnético y se conecta a una batería, la corriente pasa en un sentido por uno de sus lados y en sentido contrario por el lado opuesto. Así, sobre los dos lados de la espira se ejerce una fuerza, en uno de ellos hacia arriba y en el otro hacia abajo. Si la espira de hilo va montada sobre el eje metálico, empieza a dar vueltas hasta alcanzar la posición vertical. Entonces, en esta posición, cada uno de los hilos se encuentra situado en el medio entre los dos polos, y la espira queda retenida.

Para que la espira siga girando después de alcanzar la posición vertical, es necesario invertir el sentido de circulación de la corriente. Para conseguirlo, se emplea un conmutador o colector, que en el motor eléctrico más simple, el motor de corriente continua, está formado por dos chapas de metal con forma de media luna, que se sitúan sin tocarse, como las dos mitades de un anillo, y que se denominan delgas. Los dos extremos de la espira se conectan a las dos medias lunas. Dos conexiones fijas, unidas al bastidor del motor y llamadas escobillas, hacen contacto con cada una de las delgas del colector, de forma que, al girar la armadura, las escobillas contactan primero con una delga y después con la otra.

Cuando la corriente eléctrica pasa por el circuito, la armadura empieza a girar y la rotación dura hasta que la espira alcanza la posición vertical. Al girar las delgas del colector con la espira, cada media vuelta se invierte el sentido de circulación de la corriente eléctrica. Esto quiere decir que la parte de la espira que hasta ese momento recibía la fuerza hacia arriba, ahora la recibe hacia abajo, y la otra parte al contrario. De esta manera la espira realiza otra media vuelta y el proceso se repite mientras gira la armadura. El esquema descrito corresponde a un motor de corriente continua, el más simple dentro de los motores eléctricos, pero que reúne los principios fundamentales de este tipo de motores.

3.2.3.2. Motores de corriente alterna.

Los motores de corriente alterna tienen una estructura similar, con pequeñas variaciones en la fabricación de los bobinados y del conmutador del rotor. Según su sistema de funcionamiento, se clasifican en motores de inducción, motores sincrónicos y motores de colector.

- **Motores de inducción.**

El motor de inducción no necesita escobillas ni colector. Su armadura es de placas de metal magnetizable. El sentido alterno de circulación, de la corriente en las espiras del estator genera un campo magnético giratorio que arrastra las placas de metal magnetizable, y las hace girar. El motor de inducción es el motor de corriente alterna más utilizado, debido a su fortaleza y sencillez de construcción, buen rendimiento y bajo coste así como a la ausencia de colector y al hecho de que sus características de funcionamiento se adaptan bien a una marcha a velocidad constante.

- **Motores sincrónicos.**

Los motores sincrónicos funcionan a una velocidad sincrónica fija proporcional a la frecuencia de la corriente alterna aplicada. Su construcción es semejante a la de los alternadores. Cuando un motor sincrónico funciona a potencia constante y sobreexcitado, la corriente absorbida por éste presenta, respecto a la tensión aplicada, un ángulo de desfase en avance que aumenta con la corriente de excitación. Esta propiedad es la que ha mantenido la

utilización del motor sincrónico en el campo industrial, pese a ser el motor de inducción más simple, más económico y de cómodo arranque, ya que con un motor sincrónico se puede compensar un bajo factor de potencia en la instalación al suministrar aquél la corriente reactiva, de igual manera que un condensador conectado a la red.

3.2.4. Motores DC *brushless*.

3.2.4.1. Introducción.

Los motores de corriente continua sin escobillas (BLDC) o motores *brushless* son uno de los tipos de motores que más popularidad ha ganado en los últimos años. Actualmente, los motores BLDC se emplean en sectores industriales tales como: Automóvil, Aeroespacial, Consumo, Médico, equipos de automatización e instrumentación [5].

Los motores BLDC tienen la característica de que no emplean escobillas en la conmutación para la transferencia de energía; en este caso, la conmutación se realiza electrónicamente. Esta propiedad elimina el gran problema que poseen los motores eléctricos convencionales con escobillas, los cuales producen rozamiento, disminuyen el rendimiento, desprenden calor, son ruidosos y requieren una sustitución periódica y, por tanto, un mayor mantenimiento.

Los motores BLDC tienen muchas ventajas frente a los motores DC con escobillas y frente a los motores de inducción. Algunas de estas ventajas son:

- Mayor rendimiento (mayor duración de las baterías para la misma potencia).
- Mejor ratio velocidad-par motor y rango de velocidades al no tener limitaciones mecánicas.
- Mayor respuesta dinámica
- Mayor eficiencia y por tanto una mayor vida útil.
- Menor ruido
- Mejor disipación de calor.

Además, la relación par motor-tamaño es mucho mayor, lo que implica que se puedan emplear en aplicaciones donde se trabaje con un espacio reducido.

Por otra parte, los motores BLDC tienen dos desventajas, que son las siguientes:

- 1.- tienen un mayor coste
- 2.- requieren un control bastante más complejo

En las siguientes tabla 3.1 y 3.2 se puede observar y comparar con mayor facilidad, las ventajas e inconvenientes que tienen los motores *brushless* respecto a los motores *brushed* y motores de inducción.

Características	motor DC <i>brushless</i>	motor DC <i>brushed</i>
Conmutación	Conmutación electrónica basada en sensores de posición de efecto <i>Hall</i>	Conmutación por escobillas
Mantenimiento	mínimo	periódico
Curva velocidad/par	Plana. Operación a todas las velocidades con la carga definida	Media. A altas velocidades la fricción de las escobillas se incrementa, reduciendo el par
Eficiencia	Alta. Sin caída de tensión por las escobillas	Moderada
Potencia de salida/tamaño	Alta. Menor tamaño debido a mejores características térmicas porque los bobinados están en el estator, que al estar en la carcasa tiene una mejor disipación de calor	Baja. El calor producido en la armadura es disipado en el interior aumentando la temperatura y limitado las características
Inercia del motor	Baja. Debido a los imanes permanentes en el rotor	Alta. Limita las características dinámicas
Rango de velocidades	Alto. Sin limitaciones mecánicas impuestas por escobillas/conmutador	Bajo. El límite lo imponen principalmente las escobillas
Ruido eléctrico generado	Bajo	Arcos en las escobillas
Costes de construcción	Alto. Debido a los imanes permanentes	Bajo.
Control	Complejo y caro	Simple y barato
Requisitos de control	Un controlador es requerido siempre para mantener el motor funcionando. El mismo puede usarse para variar la velocidad	No se requiere control si no se requiere una variación de velocidad

Tabla 3.1. Diferencias entre motores *brushless* y motores *brushed*.

Características	motor DC <i>brushless</i>	motor AC inducción
Curva velocidad/par	Plana. Operación a todas las velocidades con la carga definida	No lineal. Par motor bajo a velocidades bajas
Potencia de salida/tamaño	Alta. Con los imanes permanentes en el rotor, se puede minimizar el tamaño manteniendo la misma potencia de salida	Moderada. Al tener bobinados tanto en el rotor como en el estator, la relación potencia de salida/tamaño es menor que en el motor BLDC
Inercia del motor	Baja. Buenas características dinámicas	Alta. Malas características dinámicas
Corriente de arranque	No requiere de un circuito de arranque especial	Normalmente utiliza un motor de arranque estrella-triángulo.
Requerimientos de control	Un controlador es requerido siempre para mantener el motor funcionando.	Sólo es necesario si se desea variar velocidad.
Desfase	No se experimenta entre el estator y el rotor	El rotor funciona a una frecuencia inferior a la del estator por frecuencia de deslizamiento y por tanto se incrementa el desfase con la carga en el motor.

Tabla 3.2. Diferencias entre motores *brushless* y motores AC inducción.

3.2.4.2. Topología de un motor DC *brushless*.

Los motores BLDC son un tipo de motor síncrono. Esto significa que el campo magnético generado por el estator y el campo magnético generado por el rotor giran a la misma frecuencia. Los motores BLDC no experimentan el desfase que se ve normalmente en los motores de inducción. Los motores BLDC se dividen en motores de una sola fase, motores de dos fases y motores de tres fases. En función de cada tipo, el estator tiene el mismo número de vueltas. Por supuesto, los motores de tres fases son los más populares y utilizados [14].

- **Estator.**

El estator de un motor BLDC consiste en un conjunto de láminas de acero apiladas con bobinados colocados en las ranuras de forma axial a lo largo de la periferia interna como se muestra en figura 3.12. Tradicionalmente, el estator se asemeja a un motor de inducción, sin embargo, las bobinas se distribuyen de una manera diferente. La mayoría de los motores BLDC tienen tres fases en el estator conectado en estrella. Cada una de estas fases está construida por numerosas espiras interconectadas para formar cada una de éstas. Una o más bobinas colocadas en las ranuras se interconectan para que formen una fase. Cada una de estas bobinas se distribuye en la periferia del estator para formar un número par de polos.



Figura 3.12. Topología de un motor DC *brushless*.

Hay dos tipos de bobinados del estator: motores trapezoidal y sinusoidal. Esta diferenciación se centra en la base de la interconexión de las espiras en los bobinados del estator, obteniendo los dos tipos de Fuerza contra electromotriz (*back FEM*).

Como sus nombres indican, el motor trapezoidal da una fuerza contra electromotriz de forma trapezoidal y para un motor sinusoidal, la fuerza contra electromotriz tiene forma sinusoidal. Además de la fuerza contra electromotriz, la forma de la señal de corriente también tiene variaciones trapezoidales y sinusoidales en función del tipo de motor. Esto hace que el par de salida para un motor sinusoidal sea más suave que para un motor trapezoidal. Sin embargo, este beneficio conlleva con un coste extra. Esto es debido a que los motores sinusoidales necesitan de una mayor cantidad de cobre para interconectar los distintos bobinados distribuidos en el estator.

- **Rotor.**

El rotor es de imán permanente y puede variar desde dos hasta ocho pares de polos alternativos de Norte (N) y Sur (S). En función de la densidad de campo magnético requerido en el rotor, se escoge el material magnético adecuado para hacer el rotor. Los imanes de ferrita se utilizan tradicionalmente para hacer imanes permanentes.

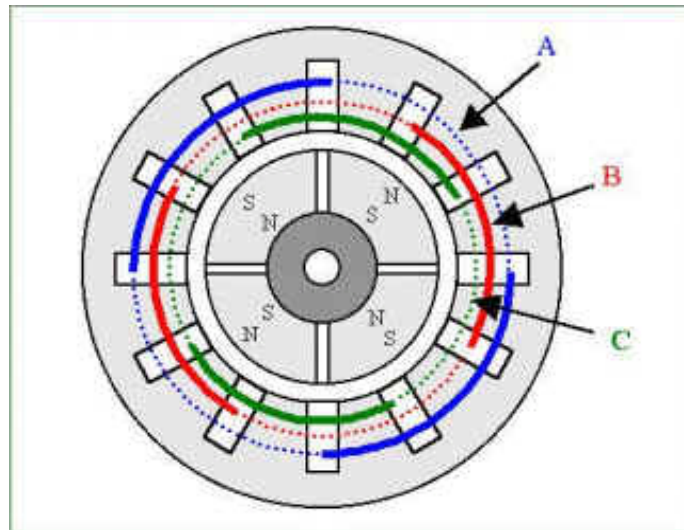


Figura 3.13. Aspecto de un motor de corriente continua sin escobillas trifásico.

A medida que avanza la tecnología, surgen imanes de aleaciones de tierras raras, ganando popularidad. Los imanes de ferrita son los menos costosos pero tienen la desventaja de tener una densidad de flujo baja para un volumen determinado. Por el contrario, los materiales de aleación de alta densidad magnética por volumen permiten comprimir y reducir el tamaño del rotor para el mismo par. Además los imanes aleados mejoran la relación tamaño/peso, proporcionando un par motor mayor que para el mismo tamaño de un motor que usa imanes de ferrita. Algunos materiales empleados que se caracterizan por tener un campo magnético permanente son: el Neodimio (Nd), el samario cobalto (Sm-Co) y la aleación de neodimio, ferrita y boro (Nd-Fe-B). Estos son algunos ejemplos de imanes de aleación de tierras.

3.2.4.3. Sensores de posición de efecto *Hall*.

Si un conductor está conduciendo una corriente eléctrica, en este se genera un campo magnético. El campo magnético genera una fuerza transversal en los portadores de la carga en movimiento que tiende a empujar a un lado del conductor. Esto es más evidente en un conductor plano delgado. Una acumulación de la carga en los lados del conductor va a equilibrar esta influencia magnética, produciéndose variación de tensión entre los lados del conductor. La presencia de esta diferencia de potencial transversal medible se llama efecto *Hall* y fue Edwin Herbert Hall quien lo descubrió en 1879.

Los sensores de efecto *Hall* tienen numerosas aplicaciones. Algunos ejemplos que permiten medir estos sensores son:

- La movilidad de una partícula cargada eléctricamente (electrones, lagunas, etc.).
- Los campos magnéticos (Teslámetros)
- La intensidad de corrientes eléctricas (sensores de corriente de efecto *Hall*)
- También permiten la elaboración de sensores o detectores de posición sin contacto, utilizados particularmente en el automóvil, para detectar la posición de un árbol giratorio (caja de cambios, paliers, etc.).
- Encontramos también sensores de efecto *Hall* bajo las teclas de los teclados de los instrumentos de música modernos (órganos, órganos digitales, sintetizadores) evitando así el desgaste que sufren los contactos eléctricos tradicionales.
- Encontramos sensores de efecto *Hall* en el codificador de un motor de CD.
- Los motores de efecto *Hall* (HET) son aceleradores de plasma de gran eficacia.

A diferencia de un motor de escobillas de corriente continua, la conmutación de un motor BLDC se controla electrónicamente. Para girar el motor BLDC, los bobinados del estator deberían estar activados en una secuencia. Es importante conocer la posición del rotor para poder entender cómo deben ser activadas las bobinas según la secuencia de activación. La posición del rotor se detecta mediante sensores de efecto *Hall* integrándose en el estator. La mayoría de los motores BLDC tienen tres sensores *Hall* integrados en el estator en el extremo opuesto al rotor del motor. Cada vez que los polos magnéticos del rotor pasan cerca de los sensores *Hall*, dan una señal de alto o bajo, lo que indica el polo N o S está pasando cerca de los sensores. Basado en la combinación de estas tres señales del sensor *Hall*, la secuencia exacta de conmutación se puede determinar.

La figura 3.14 muestra una sección transversal de un motor BLDC con un rotor que tiene imanes alternativos N y S permanentes. Los sensores *Hall* se incrustan en la parte fija del motor. La incorporación de los sensores de efecto *Hall* en el estator es un proceso complejo, ya que cualquier desajuste en los sensores de efecto *Hall*, con respecto a los imanes del rotor, se generaría un error en la determinación de la posición del rotor. Para simplificar el proceso de montaje de los sensores de efecto *Hall* en el estator, algunos motores pueden tener sensores magnéticos *Hall* en el rotor, además del rotor magnético principal. Estos son una versión reducida de la réplica del rotor. Por lo tanto, cada vez que el rotor gira, los sensores magnéticos *Hall* proporcionan el mismo efecto que los imanes principales. Los sensores *Hall* se montan normalmente en una placa de circuito impreso y se fija a la tapa de caja en el extremo de no conducción. Esto permite a los usuarios ajustar el conjunto completo de sensores de efecto *Hall*, para alinearse con los imanes del rotor, con el fin de lograr el mejor rendimiento. Sobre la base de la posición física de los sensores *Hall*, hay dos versiones de la producción. Los sensores de *Hall* pueden estar en 60° o 120° de variación de una fase a la otra. Sobre esta base, el fabricante del motor define la secuencia de conmutación, que se deben seguir cuando se controla el motor.

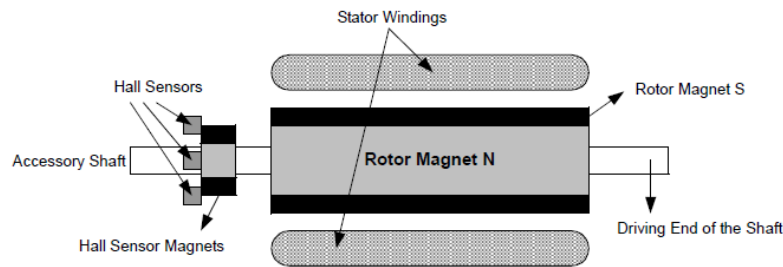


Figura 3.14 Distribución de los sensores de efecto *Hall* de un motor DC *brushless*.

Los sensores *Hall* requieren una fuente de alimentación. La tensión puede variar de 4 voltios a 24 voltios. La corriente necesaria puede variar desde 5 hasta 15 miliamperios. Durante el diseño del controlador, es necesario consultar las propias especificaciones técnicas del motor, para poder trabajar con rangos de tensiones y corrientes exactas requeridas por los sensores de efecto *Hall* utilizado. La salida del sensor *Hall* es normalmente un tipo de colector abierto. Es por ello que puede ser requerida una resistencia de *pull-up* en el lado del controlador.

3.2.4.4. Principio de funcionamiento.

A cada secuencia de conmutación le corresponde un estado de activación de las bobinas, de tal forma que una de los devanados tiene una tensión de alimentación positiva (corriente entra en el devanado), un segundo devanado tiene una tensión de alimentación negativo (las salidas de corriente de la bobina) y el tercero se encuentra en estado no activado. El par motor se produce debido a la interacción entre el campo magnético generado por las bobinas del estator y los imanes permanentes. Idealmente, el par máximo se produce cuando estos dos campos se encuentran a 90° uno del otro y decae a cero cuando ambos campos magnéticos se mueven juntos. A fin de mantener el motor en marcha, el campo magnético producido por las bobinas debe cambiar de posición (debe cambiar la activación y desactivación de las bobinas), de tal forma que según el movimiento del campo magnético del rotor, se actualice el campo magnético del estator. Lo que define la secuencia de la activación de los bobinados, la cual se conoce como "Seis Pasos de conmutación" [6].

La figura 3.15 muestra un ejemplo de las señales de los sensores de efecto *Hall* con respecto a la fuerza contraelectromotriz y la corriente de fase. La figura 3.16 muestra la secuencia de conmutación que se deben seguir con respecto a los sensores *Hall*. Los números de secuencia en la figura 3.15 corresponden a los números indicados en la figura 3.16. Cada 60 grados eléctricos de la rotación, uno de los sensores *Hall* cambia del estado. Teniendo en cuenta esto, se necesitan seis pasos para completar un ciclo eléctrico. Sincronizadamente, con cada 60 grados eléctricos, el cambio de fase actual debe ser actualizado. Sin embargo, un ciclo eléctrico no tiene porque corresponder a una revolución mecánica completa del rotor. El número de ciclos eléctricos que se repite para completar una rotación mecánica está determinado por los pares de polos del rotor. Para cada par de polos del rotor, un ciclo eléctrico se ha completado. Por lo tanto, el número de ciclos eléctricos entre rotaciones mecánicas es igual a los pares de polos del rotor.

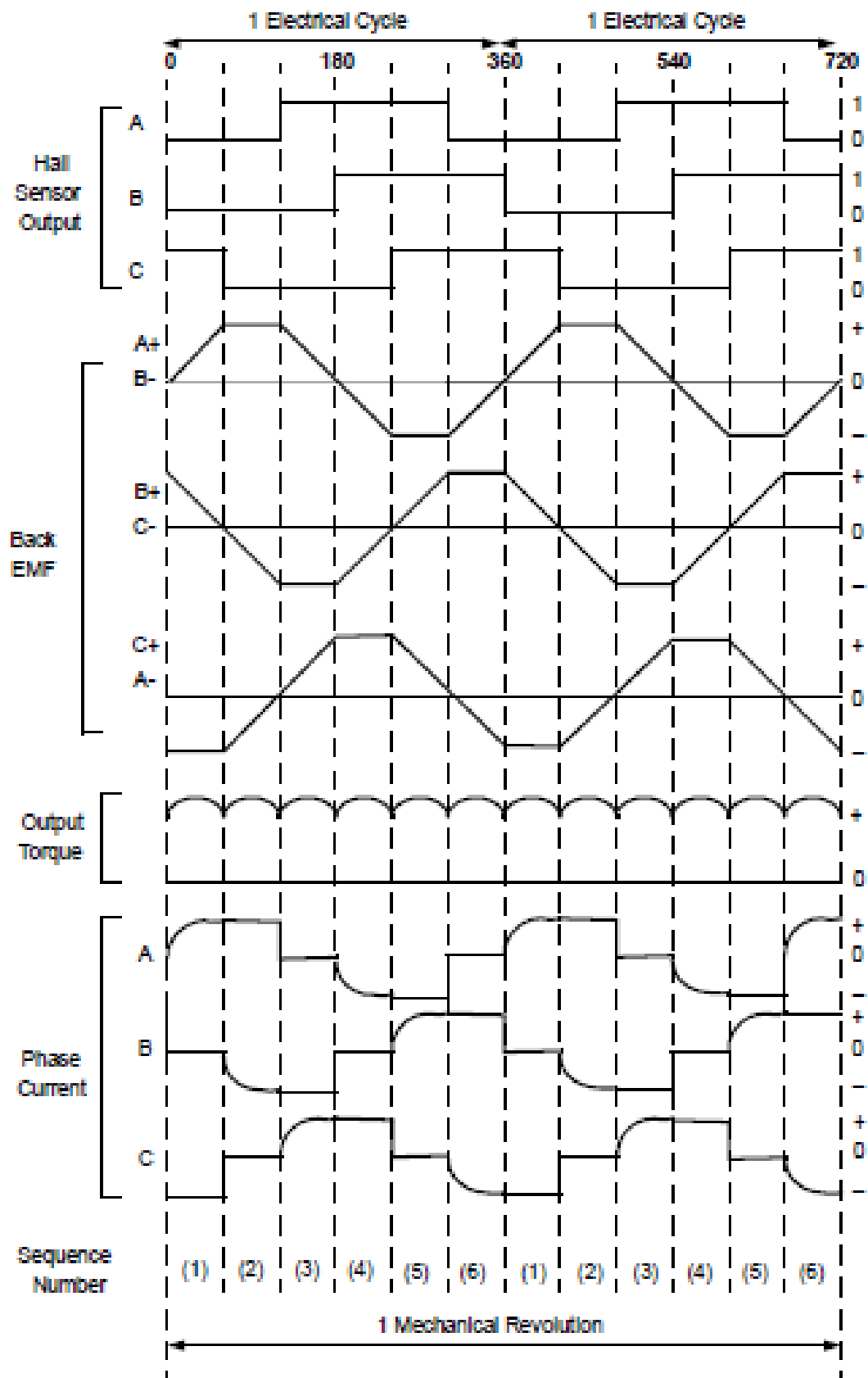


Figura 3.15. Relación grafica de la fuerza contraelectromotriz, el par motor y la corriente de fase, en función de las entradas de los sensores de efecto *Hall*.

Un motor DC *brushless* necesita de una etapa de potencia para suministrar tensión y corriente a las bobinas del motor. Para ellos al motor BLDC se le conecta un inversor trifásico, el cual consiste entre 3 ramas inversores, con dos interruptores de potencia por cada rama. Cada uno de estos interruptores se activa en función de una señal PWM y que es distinta para cada uno. Esto inversores se explican con más detenimiento en el siguiente capítulo.

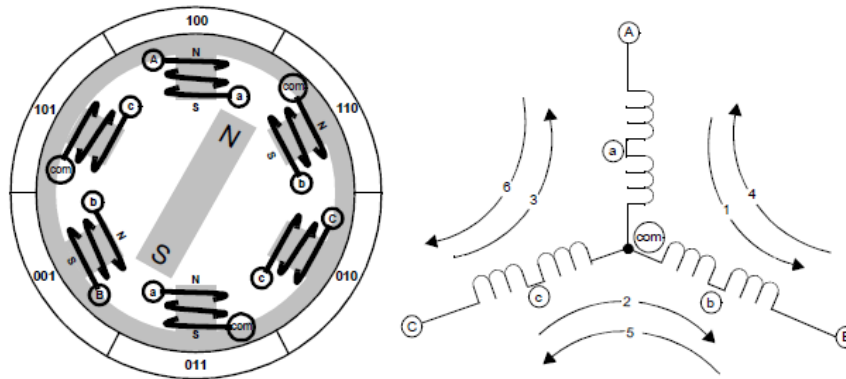


Figura 3.16. Secuencia de activación de los devanados en función de las entradas de los sensores de efecto Hall.

Pin	RE2	RE1	RE0	RC5	RC4	RC3	RC2	RC1	RC0
Phase	Sensor C	Sensor B	Sensor A	C High Drive	C Low Drive	B High Drive	B Low Drive	A High Drive	A Low Drive
6	0	0	1	0	1	0	0	1	0
4	0	1	0	0	0	1	0	0	1
5	0	1	1	0	1	1	0	0	0
2	1	0	0	1	0	0	1	0	0
1	1	0	1	0	0	0	1	1	0
3	1	1	0	1	0	0	0	0	1

Tabla 3.3. Activación de los drivers del motor según el orden de los sensores Hall, para la configuración de giro horario.

Pin	RE2	RE1	RE0	RC5	RC4	RC3	RC2	RC1	RC0
Phase	Sensor C	Sensor B	Sensor A	C High Drive	C Low Drive	B High Drive	B Low Drive	A High Drive	A Low Drive
/6	0	0	1	1	0	0	0	0	1
/4	0	1	0	0	0	0	1	1	0
/5	0	1	1	1	0	0	1	0	0
/2	1	0	0	0	1	1	0	0	0
/1	1	0	1	0	0	1	0	0	1
/3	1	1	0	0	1	0	0	1	0

Tabla 3.4. Activación de los drivers del motor según el orden de los sensores Hall, para la configuración de giro anti horario.

Las tablas 3.3 y 3.4 muestran la secuencia en la que estos interruptores de potencia deben activarse o desactivarse según las entradas de los sensores de efecto *Hall*, A, B y C. La tabla 3.3 es para la rotación del motor con sentido positivo a las agujas del reloj y en la tabla 3.4 es la rotación del motor con sentido contrario a las agujas del reloj. Estas tablas son un ejemplo en el que las señales de los sensores de efecto *Hall* tienen un cambio de fase de 120 grados con respecto a ellos mismos. Como se ha explicado previamente en la sección de sensores de efecto *Hall*, los sensores pueden estar desfasados a 60 ° o 120 °.

Si las señales PWM'x que llegan a las puertas (*Gates*) de los interruptores, están encendiendo y apagando según la secuencias definidas en las tablas 3.3 y 3.4, el motor funcionará a la velocidad nominal. Esto es suponiendo que la tensión del bus DC es igual a la tensión nominal del motor, más las pérdidas a través de los interruptores. Para variar la velocidad, estas señales deben ser de ancho de pulso modulado (PWM) a una frecuencia mucho más alta que la frecuencia del motor. Como regla general, la frecuencia de PWM es por lo menos 10 veces la de la frecuencia máxima del motor. Cuando el ciclo de trabajo del PWM es muy variado (poca resolución) dentro de las secuencias, la tensión media suministrada a la del estator se reduce, lo que reduce la velocidad. Otra de las ventajas de tener PWM es que, si la tensión del bus se muy superior a la tensión nominal del motor, el motor se puede controlar mediante la limitación del porcentaje de PWM deber que corresponde al de la tensión nominal del motor de ciclo. Esto añade la flexibilidad al regulador para conectar motores con tensiones nominales diferentes e igualar la salida de voltaje media del regulador con la tensión nominal de motor, y todo ello controlado a través del ciclo de trabajo PWM.

3.2.4.5. Control en bucle cerrado.

La velocidad puede ser controlada en un lazo de control cerrado a través de la medición de la velocidad real del motor. El error se calcula restando la velocidad consiga y la velocidad real. Se puede usar un controlador PID que amplifique el error de velocidad y ajuste dinámicamente el ciclo de trabajo del PWM. Para aplicaciones en las que se requiera baja resolución de velocidad o bajo coste, se puede usar las señales de los sensores de efecto *Hall* para medir la velocidad real. También se puede usar un temporizador para contar entre dos transiciones de los sensores *Hall*. Con esta cuenta, la velocidad real del motor se puede calcular, dividiendo la diferencia en grados de la distancia entre las transiciones de los sensores *Hall* entre la cuenta del temporizador. Para las mediciones de velocidad de alta resolución, se puede montar un *encoder* óptico en el motor, el cual da dos señales con una diferencia de fase de 90 grados. A través del uso de estas dos señales desfasadas se puede determinar la velocidad y la dirección de rotación. Además, la mayoría de los *encoder* ópticos tienen una tercera señal “índice”, que consiste en la activación de un pulso cada vez que el motor realice una revolución. Esto puede ser usado para aplicaciones de posicionamiento. Los *encoders* ópticos están disponibles con diferentes opciones de pulso por revolución (PPR), que van desde cientos a miles, y que indican al usuario la resolución de éstos.

3.2.4.6. Características par/velocidad.

La Figura 3.17 muestra un ejemplo de las características de par / velocidad. Hay dos parámetros de par que se utiliza para definir un motor BLDC: el par máximo o de pico (TP – *Peak Torque*) y el par nominal (TR - *Rated Torque*). Durante las operaciones continuas, el motor puede es cargado con el par nominal y dicho par se mantiene constante para un rango de velocidad hasta la velocidad nominal. El motor puede funcionar a la velocidad máxima, que puede ser hasta el 150% de la velocidad nominal, pero el par comienza a caer. En las aplicaciones en las que aparecen arranques y paradas frecuentes y cambios frecuentes de rotación con carga en el motor, la demanda crece por encima del par nominal. Este requisito viene porque durante un breve período tiempo, especialmente cuando el motor arranca desde parado y durante una aceleración, el par adicional es necesario para poder superar la inercia de la carga y el propio rotor. El motor puede entregar un par mayor hasta un valor máximo, siempre y cuando se siga la curva de par/velocidad.

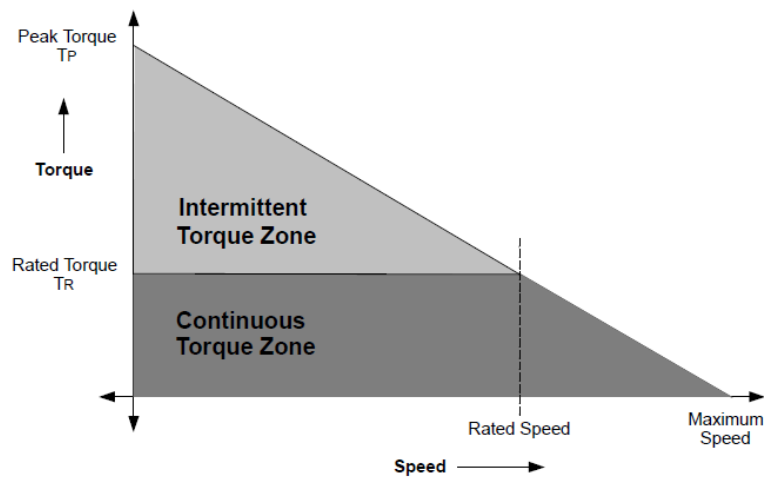


Figura 3.17. Ejemplo genérico de la relación par motor/velocidad.

3.3. Driver de potencia.

3.3.1. Topología del inversor.

Los convertidores DC-AC son conocidos por inversores. Ellos pueden tener salida variable en voltaje y frecuencia. Y es típico usar estos inversores con *drivers* de potencia para motores.

La topología del inversor queda definida por el modo de conexión de los dispositivos de conmutación que conforman el puente. Cada rama del puente consta de dos dispositivos de conmutación. Una o dos ramas –puente H- (dos o cuatro dispositivos de conmutación) son empleados para puentes monofásicos y bifásicos. Tres ramas pueden ser conectadas para conformar un convertidor trifásico (seis dispositivos de conmutación) siendo este el de mayor uso, aunque pueden emplearse mayor número de ramas para aumentar la capacidad de conducción [13].

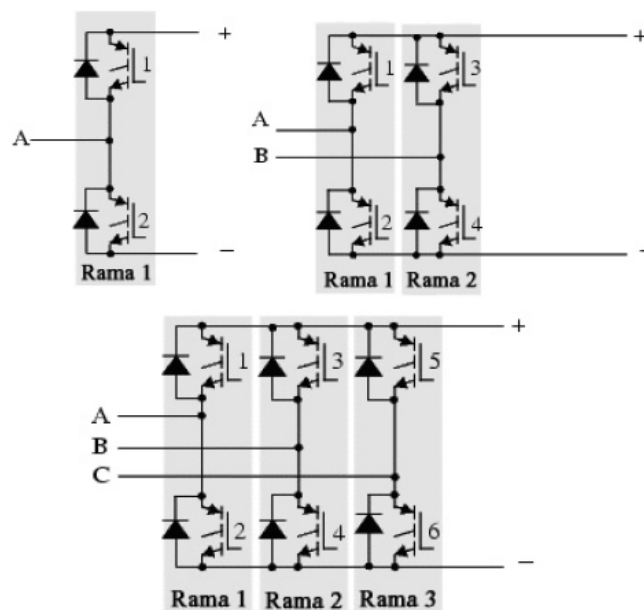


Figura 3.18. Topología de Inversores de una, dos y tres ramas.

La configuración más simple es el inversor monofásico, de medio puente, el cual requiere una fuente DC de tres conductores, sin embargo los inversores trifásicos se utilizan normalmente en aplicaciones de alta potencia. Tres inversores monofásicos de medio puente (o puente completo) pueden conectarse para formar un inversor trifásico, siempre que sus señales de puerta estén desfasadas 120° . Las configuraciones de seis dispositivos de conmutación, son los más empleados, pudiéndose aplicar señales de control: conducción a 120° o conducción a 180° , es decir, permitiendo la conducción por dos o tres dispositivos de conmutación respectivamente.

Algunas aplicaciones de inversores requieren de medios de control sobre el voltaje de salida. En la mayoría de esas aplicaciones el control es usualmente requerido a fin de proveer un ajuste continuo (*stepless*) del voltaje de salida. Los métodos de control pueden ser agrupados en tres grandes categorías:

- Control de voltaje suministrado al inversor.
- Control del voltaje entregado por el inversor.
- Control del voltaje dentro del inversor.

Hay un número de bien conocidos métodos de controlar el voltaje DC suministrado al inversor o del voltaje AC entregado por el inversor. Esto incluye el uso de Chopper DC, amplificadores magnéticos, reguladores de inducción, rectificadores de fase controlado y transistores series o reguladores *shunt*. La principal desventaja de estos métodos es que la potencia entregada por el inversor es manejada dos veces, una vez por el control de voltaje DC o AC y otra por el inversor. Esto generalmente involucra más equipos que los que serían necesario que si el control de voltaje es hecho dentro del inversor. El control de la salida del inversor puede ser logrado por la incorporación de controles de relación de tiempo dentro del circuito del inversor.

Un método de controlar el voltaje dentro del inversor involucra el uso de las técnicas de modulación de ancho de pulso (PWM). Con esta técnica el voltaje de salida del inversor es controlado por la variación de la duración de los pulsos de voltaje de salida.

Es importante considerar que los interruptores precisan de un tiempo mínimo, tanto en la apertura para anular la corriente, como para el cierre para su establecimiento. Por tanto se debe prestar atención al instante de cierre de un interruptor durante un tiempo de bloque necesario del interruptor complementario a la misma rama. Esta corriente de descarga circulará por los diodos dispuestos en paralelo con cada interruptor. Una vez que la corriente sea nula, se permitirá el cierre del interruptor complementario. Este tiempo de espera se denomina generalmente tiempo muerto y debe de ser respetado y tenido en cuenta durante el diseño [18]. Es por eso que las señales de control de los dos interruptores de un mismo brazo deben ser complementarias para el inversor de conducción a 180° , a fin de no cortocircuitar la fuente de continua de alimentación, o para el caso de conducción a 120° , que las señales de control nunca estén activas al mismo tiempo. También es importante usar protecciones en los apagados y encendidos de los interruptores, de tal forma que se pueda absorber la energía procedente de los elementos reactivos del circuito durante el proceso de conmutación controlando parámetros tales como la evolución de la tensión o corriente en el interruptor, o bien limitando los valores máximos de tensión que ha de soportar.

3.3.2. Inversor trifásico.

El objetivo de un inversor trifásico es generar energía eléctrica de corriente alterna a partir de una fuente de energía de corriente continua, con magnitudes y frecuencias deseadas. Se constituye principalmente por dispositivos electrónicos de potencia, que trabajan como interruptores operando en corte y saturación con una secuencia apropiada para obtener tres tensiones de salida simétricas y balanceadas. El controlador es otro componente fundamental

en la constitución del convertidor, es el que genera las señales de encendido y apagado de los dispositivos semiconductores y garantiza su buen comportamiento. Cualquier tipo de inversor (monofásico ó trifásico) utilizan dispositivos con activación y desactivación controlada (es decir BJT, MOSFET, IGBT, MCT, SIT, GTO) o tiristores de conmutación forzada, según la aplicación.

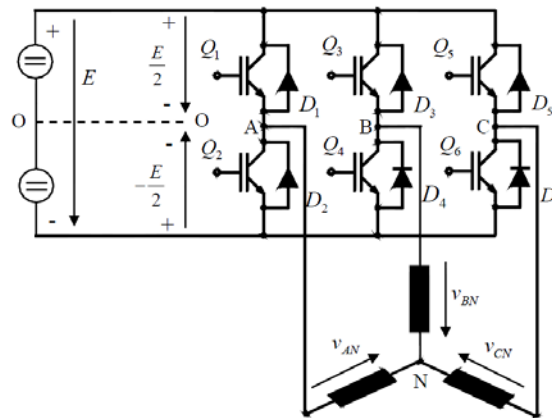


Figura 3.19. Inversor trifásico con dos niveles de tensión.

La figura 3.19 muestra el caso de un puente con tres ramas, o inversor trifásico en puente completo, el cual se compone de seis dispositivos de conmutación (6 transistores IGBTs) designados Q1 a Q6 y seis diodos de libre circulación (D1 a D6) dispuesto en antiparalelo o conexión inversa con los interruptores, empleados para conducir la corriente reactiva de retorno a la fuente de tensión E. Estos diodos aseguran por un lado la continuidad de la corriente en la carga inductiva y por otro lado la reversibilidad de la potencia al poder inyectar corriente desde la carga a la batería de continua. Cada brazo del inversor está formado por dos interruptores o dispositivos de conmutación en paralelo con sus diodos de libre circulación, estando la salida a cada fase del motor situada en el punto medio del brazo. Estos inversores se dividen según su forma de operar en: conducción a 180° de cada elemento, con lo cual habrá 3 elementos en conducción al mismo tiempo y conducción a 120° , con 2 elementos por vez. Además pueden alimentar los dos tipos característicos de cargas trifásicas simétricas: conexión delta y estrella [15].

3.3.2.1. Conducción a 180° .

Cada transistor conducirá durante 180° . Tres transistores se mantienen activos durante cada instante del tiempo. Cuando el transistor Q1 está activado, la fase “a” se conecta con la terminal positiva del voltaje de entrada. Cuando se activa el transistor Q4, la fase “a” se lleva a la terminal negativa de la fuente DC. En cada ciclo existen seis modos de operación, cuya duración es de 60 grados. Los transistores se numeran según su secuencia de excitación por ejemplo (612, 123, 234, 345, 456, 561, 612). Las señales de excitación mostradas en la figura 3.20 están desplazadas 60 grados unas de otras, para obtener voltajes trifásicos balanceados [3].

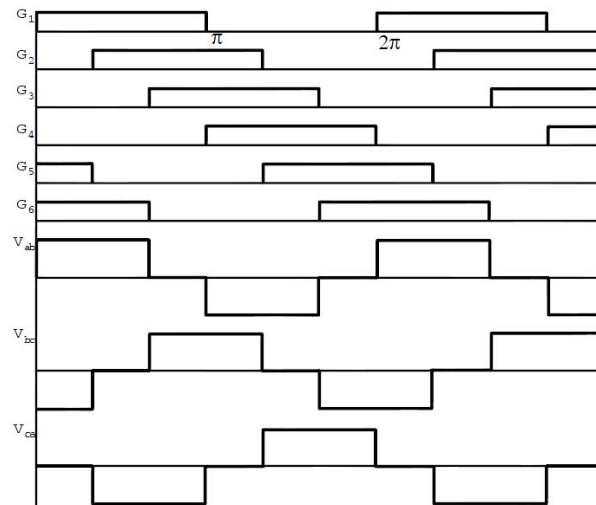


Figura 3.20. Secuencia de la señales de excitación de los transistores a 180°.

La limitación de la simultaneidad de la conducción de los dos interruptores de un mismo brazo, implica que solamente existen ocho configuraciones posibles de salida del inversor. Los diferentes estados de la conmutación de los tres brazos del inversor conducen a las tensiones que se indican en la tabla 3.5, expresada en función del estado de los tres transistores superiores de cada rama (Q1, Q3, Q5) mostrados en la figura 3.19. Las tensiones calculadas corresponden a los valores respecto a la referencia “o”, situada en el punto medio ficticio de la alimentación de continua del inversor (V_{ao} , V_{bo} , V_{co}). También se muestra las tensiones entre dos fases de salida (V_{ab} , V_{bc} , V_{ca}).

Estas tensiones se obtienen en función del estado de los diferentes interruptores, tal y como se muestra en la figura 3.21. se puede apreciar que en el primer estado y en el último las tensiones de alimentación al motor son nulas, por lo que a veces estos estados se denominan “estados de libre circulación”.

Q1	Q2	Q3	V_{ao}	V_{bo}	V_{co}	V_{ab}	V_{bc}	V_{ca}
A	A	A	$-E/2$	$-E/2$	$-E/2$	0	0	0
C	A	A	$+E/2$	$-E/2$	$-E/2$	$+E$	0	$-E$
C	C	A	$+E/2$	$+E/2$	$-E/2$	0	$+E$	$-E$
A	C	A	$-E/2$	$+E/2$	$-E/2$	$-E$	$+E$	0
A	C	C	$-E/2$	$+E/2$	$+E/2$	$-E$	0	$+E$
A	A	C	$-E/2$	$-E/2$	$+E/2$	0	$-E$	$+E$
C	A	C	$+E/2$	$-E/2$	$+E/2$	$+E$	$-E$	0
C	C	C	$+E/2$	$+E/2$	$+E/2$	0	0	0

Tabla 3.5. Tensiones de cada rama respecto al punto intermedio de la alimentación ‘O’ y entre dos fases, en función del estado de los interruptores, para una conducción de 180°. A: abierto; C: cerrado.

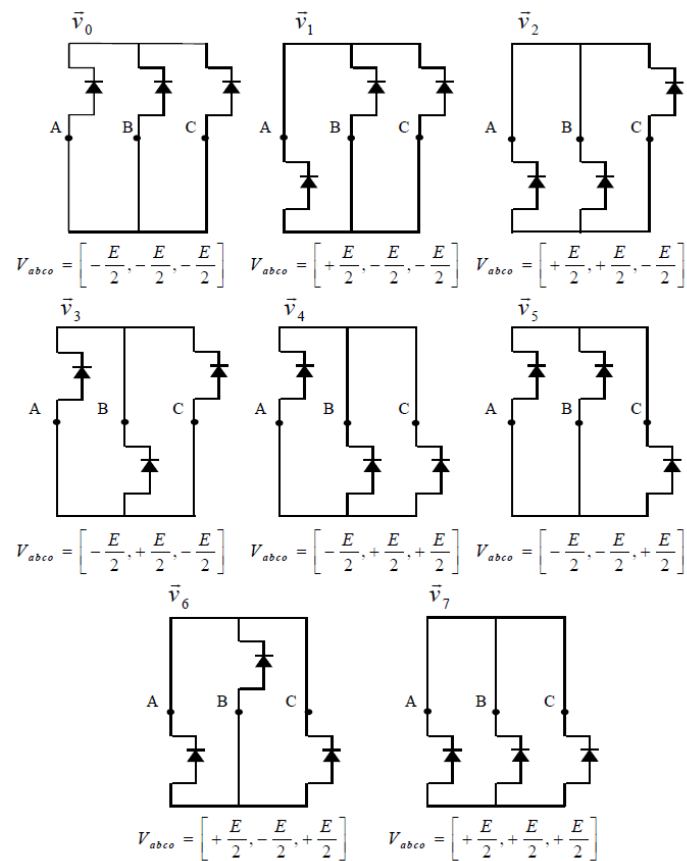


Figura 3.21. Distintas configuraciones del inversor en función del estado de los interruptores, para una conducción de 180°.

3.3.2.2. Conducción a 120°.

En este tipo de control, cada transistor conduce durante 120°. En cualquier instante del tiempo, sólo conducen dos transistores, dándose la posibilidad de que los dos transistores de una misma rama estén abiertos. Las señales de excitación se muestran en la figura 3.22. La secuencia de conducción de los transistores es 61, 12, 23, 34, 45, 56, 61, ... [4].

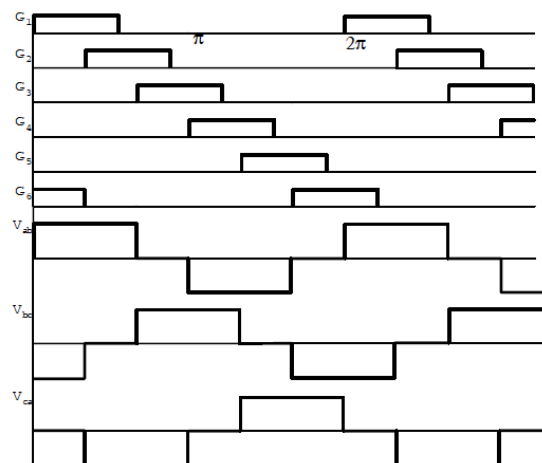


Figura 3.22. Secuencia de la señales de excitación de los transistores a 120°.

La limitación de la simultaneidad de la conducción de los dos interruptores de un mismo brazo es una propiedad que se repite de nuevo en el inversor trifásico con conducción a 120°, pero la diferencia entre estos dos tipos de inversores radica en posibilidad de tener dos interruptores de una misma rama en abierto. Esta nueva característica implica la que existen más de ocho combinaciones posibles de salida del inversor. Los diferentes estados de la conmutación de los tres brazos del inversor conducen a las tensiones que se indican en la tabla 3.6, expresada en función del estado de los seis transistores (Q1, Q2, Q3, Q4, Q5, Q6) mostrados en la figura 3.19. Las tensiones calculadas corresponden a los valores respecto a la referencia “o”, situada en el punto medio ficticio de la alimentación de continua del inversor (Vao, Vbo, Vco). También se muestra las tensiones entre dos fases de salida (Vab, Vbc, Vca).

Estas tensiones se obtienen en función del estado de los diferentes interruptores, tal y como se muestra en la figura 3.23. Se puede apreciar en la tabla sólo aparecen seis casos posibles, ya que estos son los que definen el funcionamiento correcto para producir movimiento en un motor DC *brushless*.

Q1	Q2	Q3	Q4	Q5	Q6	Vao	Vbo	Vco	Vab	Vbc	Vca
A	A	A	C	C	A	0	-E/2	+E/2	0	-E	0
C	A	A	C	A	A	+E/2	-E/2	0	+E	0	0
C	A	A	A	A	C	+E/2	0	-E/2	0	0	-E
A	A	C	A	A	C	0	+E/2	-E/2	0	+E	0
A	C	C	A	A	A	-E/2	+E/2	0	-E	0	0
A	C	A	A	C	A	-E/2	0	+E/2	0	0	+E

Tabla 3.6. Tensiones de cada rama respecto al punto intermedio de la alimentación ‘O’ y entre dos fases, en función del estado de los interruptores, para una conducción de 120°. A: abierto; C: cerrado.

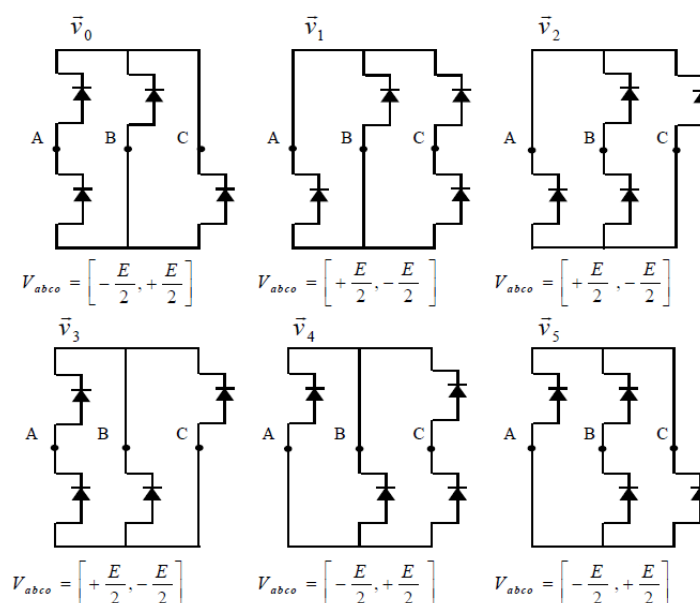


Figura 3.23. Distintas configuraciones del inversor en función del estado de los interruptores, para una conducción de 120°.

Para la aplicación de inversores trifásicos en motores, debe utilizarse la misma señal PWM para controlar los seis interruptores, aunque esta señal es producida y generada por el propio programa dentro del controlador. Esta señal PWM varía la tensión de salida y su frecuencia, pero para activar los interruptores en la secuencia correcta, se diseña un señal a partir de una puerta lógica AND de dos entradas, que se forma a partir de la señal PWM y un maquinas de estados programado dentro del controlador. Por tanto se obtiene seis señales distintas que indican que interruptores deben estar abiertos y cuales cerrados. Esta máquina de estado está definida en función de la activación de los sensores *hall* del propio motor. Por tanto, la función principal que tiene esta máquina de estados es desfasar la señal PWM. (Ver tabla x.x).

3.3.3. Tecnología PWM - modulación de ancho de pulso.

La tecnología PWM, es una tecnología robusta. Esta técnica de control se basa en emplear una modulación múltiple (varios pulsos de disparo en cada medio ciclo de voltaje de salida), el ancho de cada pulso varía en proporción con la amplitud de una onda senoidal evaluada en el centro del mismo pulso [17]. (Ver figura 3.24).

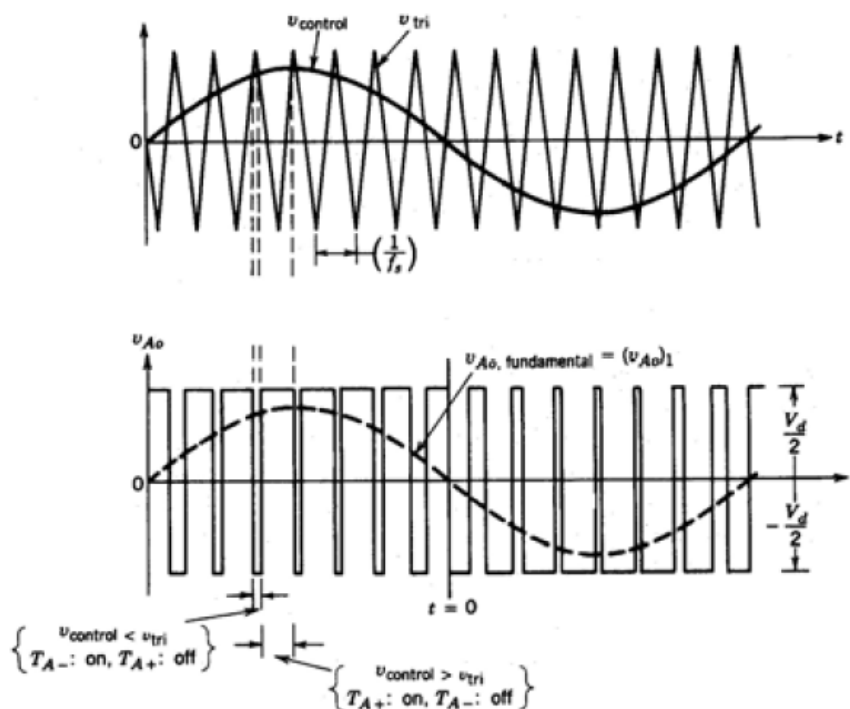


Figura 3.24. Esquema de modulación de ancho de pulso.

La modulación se logra empleando una señal de control senoidal “ $V_{control}$ ” a la frecuencia de salida deseada “ f_1 ”, que es comparada con una onda portadora triangular para generar las señales de disparo. La frecuencia de la forma de onda triangular establece la frecuencia de conmutación del inversor “ f_s ”, y esta se mantiene constante. La relación de modulación de amplitud es:

$$m_a = V_{\text{control}} / V_{\text{tri}} \quad \text{Ecuación 3.2}$$

Donde:

- “ V_{control} ” es la amplitud pico de la señal de control (V).
- “ V_{tri} ” es la amplitud pico de la onda portadora triangular (V).
- “ m_a ” es la relación de modulación de amplitud.

La relación de modulación de frecuencia es:

$$m_f = f_s / f_1 \quad \text{Ecuación 3.3}$$

Donde:

- “ m_f ” es la relación de modulación en frecuencia.
- “ f_s ” es la frecuencia de la onda portadora triangular (Hz).
- “ f_1 ” es la frecuencia de la señal de control (Hz).

En el caso de un puente completo de dos ramas, se cumple: cuando $V_{\text{control}} > V_{\text{trip}}$, TA+ y TB- se mantienen operando y $V_0 = V_d$. Cuando $V_{\text{control}} < V_{\text{trip}}$, TA- y TB+ se mantienen operando y $V_0 = -V_d$. El voltaje de salida fluctúa entre $+V_d$ y $-V_d$.

El voltaje de salida del inversor contiene armónicas, cuyo espectro armónico se muestra en la figura siguiente. (Ver figura 3.25).

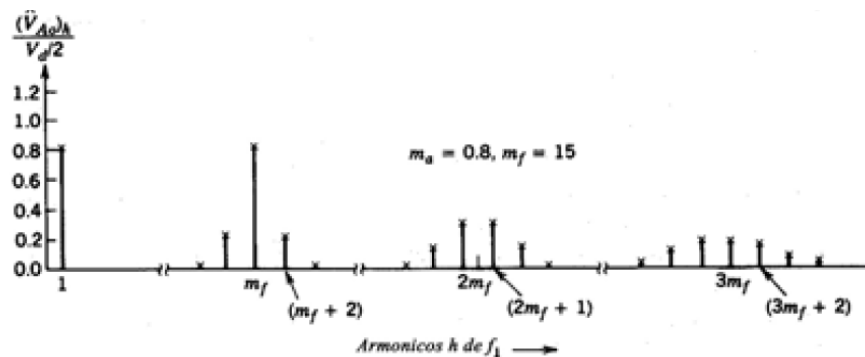


Figura 3.25. Contenido de armónicos del esquema PWM.

Para relación de modulación de amplitud menor a uno ($m_a < 1$), se opera en el rango lineal del inversor, y el voltaje pico de la componente de frecuencia fundamental de voltaje de salida preserva una relación lineal entre el índice de modulación de amplitud y la tensión V_d . En la ecuación 3.4 se muestra dicha relación lineal mientras m_a sea menor que la unidad.

$$V_{A0} = m_a * V_d \quad \text{Ecuación 3.4}$$

Donde:

- “ V_{A0} ” es la amplitud de pico de la señal de salida (V).
- “ m_a ” es la relación de modulación de amplitud.
- “ V_d ” es la tensión de alimentación del circuito inversor (V).

El PWM empuja los armónicos de la onda de voltaje de salida al rango de las altas frecuencias, alrededor de la frecuencia de conmutación f_s y sus múltiplos como m_f , $2m_f$, $3m_f$ y más. m_f debe ser un número entero impar tal que la forma de onda de voltaje de salida solo contenga armónicas impares. De esta forma, es más fácil de eliminar los armónicos, filtrando las altas frecuencias. De ahí, que sea deseable usar una frecuencia de conmutación lo más alta posible, aunque esto tiene la desventaja que las pérdidas de conmutación aumentan proporcionalmente. El valor seleccionado que delimita entre grande y bajo el valor de m_f es 21.

Para pequeños valores del índice de modulación de frecuencia ($m_f < 21$), la forma de onda de la señal triangular y la señal de control deben estar sincronizadas. Esto es llamado PWM sincronizado, donde m_f no es un entero, y se producen subarmónicos de la frecuencia fundamental, los cuales no son deseables. Cuando m_f se hace grande, los subarmónicos debido al PWM asincrónico son pequeños. De ahí que PWM asincrónico se emplee más.

Para PWM's con $m_a < 1$, la amplitud del voltaje fundamental varía linealmente con m_a . Pero en contraposición la magnitud de la componente de frecuencia fundamental es menor. Cuando m_a tiene un valor mayor que uno, la amplitud también se incrementa lo cual produce una sobremodulación. Por tanto, la forma de onda de salida contiene muchos más armónicos en el lado de la banda comparado con el rango lineal.

En la figura 3.26 se muestra la amplitud normalizada del pico de la componente de frecuencia fundamental V_{A0}/V_d como función de la relación del índice de modulación de amplitud.

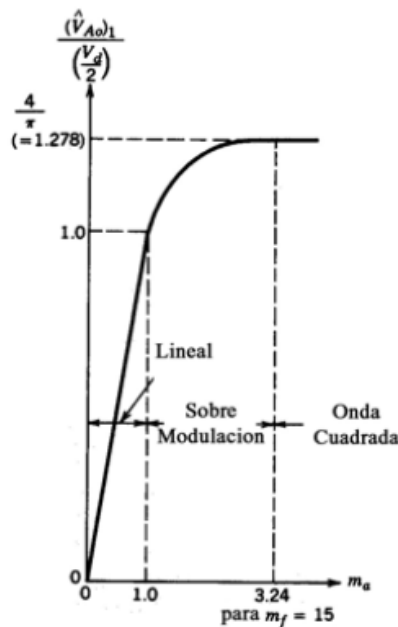


Figura 3.26. Zonas de operación del esquema PWM.

Los armónicos que son dominantes en el rango lineal pueden no ser dominantes durante la sobremodulación. La amplitud de la componente fundamentalmente no varía linealmente con m_a . En aplicaciones de potencia, la región de sobre-modulación debe ser evitada para minimizar la distorsión en el voltaje de salida. Cuando m_a es suficientemente

grande ($m_a \gg 1$), la forma de onda del voltaje de salida se degenera desde una PWM a una forma de onda cuadrada. Y en este caso, la magnitud del voltaje fundamental máximo de salida dependerá de la tensión de alimentación V_d , a través de la ecuación 3.5. (Ver figura 3.27).

$$V_{A0\max} = 4V_d / \pi = 1.278V_d \quad \text{Ecuación 3.5}$$

Donde:

- “ $V_{A0\max}$ ” es la tensión fundamental de salida máxima (V).
- “ V_d ” es la tensión de alimentación del circuito de potencia (V).

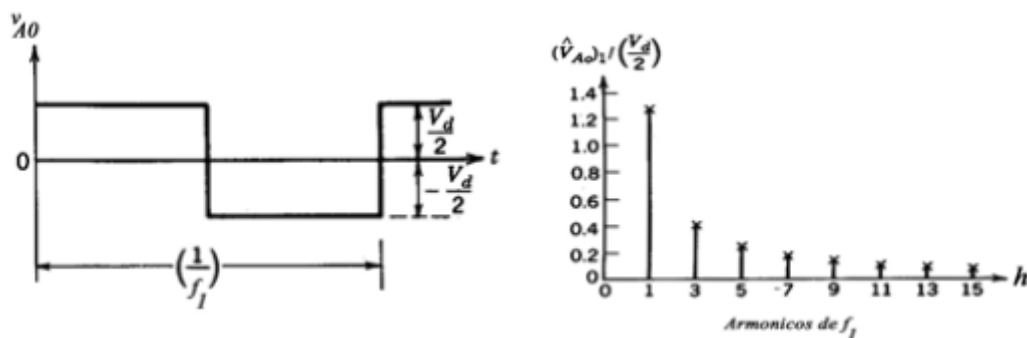


Figura 3.27. Contenido de armónicos del esquema PWM, para operación en onda cuadrada.

Los inversores trifásicos ofrecen una significativa ventaja. Si m_f es elegido tal que cumple que es impar y múltiplo de tres (por ejemplo 3, 9, 15, 21, 29...) se logra que la forma de onda de voltaje de salida sea más senoidal, incluso los armónicos estarán ausentes en el voltaje de fase. Además es deseable empujar m_f al valor más alto posible. Cuando m_f es alto, los armónicos estarán a muy altas frecuencias, de tal forma que la onda de salida apenas se verá afectada por estos armónicos y obteniendo una mejora en las pérdidas de conmutación. (Ver figura 3.28).

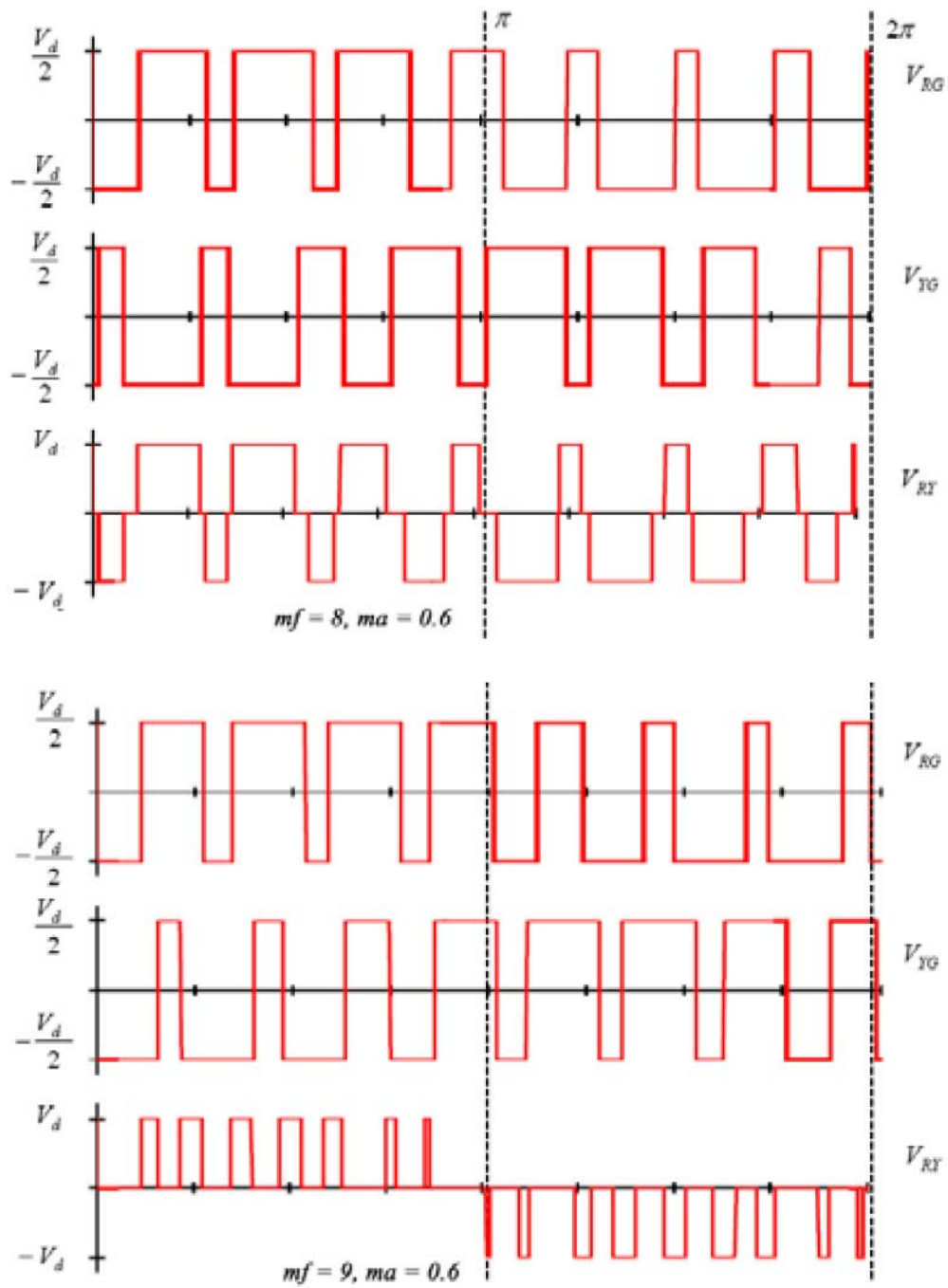


Figura 3.28. Forma de onda de voltaje en Inversor trifásico con esquema PWM.

3.4. Sistemas de control automático.

3.4.1. Introducción.

El objetivo de cualquier estrategia de control es mantener una variable llamada controlada, próxima a un valor deseado conocido como punto de ajuste “set-point”. El término regularización es usado para describir la acción de control de agentes de perturbación del estado de equilibrio de la variable controlada. Un sistema de control, solamente puede llegar a la regulación, aplicando en oposición a las fuerzas perturbadoras llamadas cargas, correcciones equivalentes en una o más variables denominadas manipuladas. La variable controlada permanecerá estable en el proceso mientras se encuentre en estado estacionario. Este equilibrio puede ser alcanzado usualmente por distintos sistemas de control [2].

Hay varias clasificaciones dentro de los sistemas de controladores.

- Atendiendo a su naturaleza son analógicos, digitales o mixtos.
- Atendiendo a su estructura (número de entradas y salidas) puede ser control clásico o control moderno.
- Atendiendo a su diseño pueden ser por lógica difusa, redes neuronales...

3.4.2. Tipos de sistemas de control.

3.4.2.1. Sistemas de Control Clásico.

- Sistemas de Control de Lazo Abierto:

Se denominan sistemas de control de lazo abierto cuando la salida no tiene efecto sobre la acción de control, es decir no se compara la salida con la entrada de referencia. Por lo tanto, para cada entrada de referencia corresponde una condición de operación fija. Así, la precisión del sistema depende de la calibración y del operador cuya función será la del controlador.

- Sistema de Control de Lazo Cerrado:

Se denomina sistema de control de lazo cerrado cuando frente a presencia de perturbaciones, tiende a reducir la diferencia entre la salida del sistema y el valor deseado o “set point”; realizando el control de forma automática [1].

3.4.2.2. Sistemas de Control Moderno.

- Control Adaptativo:

Es un método en el cual la respuesta de un controlador varía automáticamente basado en los cambios de las condiciones dentro del proceso y puede emplearse en diversas aplicaciones, como en el control del pH.

- Control Difuso:

Este control utiliza la lógica difusa a través de conceptos de inteligencia artificial capaz de convertir una muestra de la señal real a números difusos, para tratarlos según las reglas de inferencia y las bases de datos determinados en las unidades de decisión, logrando estabilizar el sistema sin la necesidad de fijar un punto de referencia.

- Redes Neuronales Artificiales:

Están diseñadas para actuar como lo hace el cerebro humano conectando la red entre los elementos de la forma más sencilla para poder ser entrenados y realizar funciones complejas en diversos campos de aplicación.

3.4.3. Controlador.

El controlador es una componente del sistema de control que detecta los desvíos existentes entre el valor medido por un sensor y el valor deseado o “set point”, programado por un operador; emitiendo una señal de corrección hacia el actuador.

Un controlador es un bloque electrónico encargado de controlar uno o más procesos. Al principio los controladores estaban formados exclusivamente por componentes discretos, conforme la tecnología fue desarrollándose se emplearon procesadores rodeados de memorias, circuitos de entrada y salida. Actualmente los controladores integran todos los dispositivos mencionados en circuitos integrados que conocemos con el nombre de microcontroladores.

Los controladores son los instrumentos diseñados para detectar y corregir los errores producidos al comparar y computar el valor de referencia o “Set point”, con el valor medido del parámetro más importante a controlar en un proceso.

La actuación puede ser de forma clásica de acuerdo al tamaño y tiempo de duración del error, así como la razón de cambio existente entre ambos o aplicando sistemas expertos a través de la lógica difusa y redes neuronales.

El Control Realimentado es la propiedad de una sistema de lazo cerrado que permite que la salida (o cualquier otra variable controlada del sistema) sea comparada con la entrada al sistema (o con una entrada de cualquier componente interno del mismo o con un subsistema) de manera tal que se pueda establecer una acción de control apropiada como función de la diferencia entre la entrada y la salida.

Más generalmente se dice que existe realimentación en un sistema cuando existe una secuencia cerrada de relaciones de causa y efecto entre las variables del sistema.

Los controladores pueden ser del tipo: manual, eléctrico, electrónico, neumático ó digitales; así como las computadoras con tarjetas de adquisición de datos y los PLC (Controladores Lógicos Programables).

Actualmente en la industria se utiliza para controlar las variables de operación; sensores inteligentes, controladores lógicos programables (PLC), supervisando y adquiriendo los datos a través de las computadores personales e integrándolas por una red y logrando un sistema de control distribuido (SOC).

3.4.4. Acciones de control de un regulador o controlador.

El controlador o regulador constituye el elemento fundamental en un sistema de control, pues determina el comportamiento del bucle, ya que condiciona la acción del elemento actuador en función del error obtenido. La forma en que el regulador genera la señal de control se denomina acción de control. Algunas de estas acciones se conocen como acciones básicas de control, mientras que otras se pueden presentar como combinaciones de las acciones básicas [8].

1. Acciones básicas:
 - Proporcional (P).
 - Derivador (D).
 - Integrador (I).
2. Combinación de acciones básicas:
 - Proporcional - Integrador (PI).
 - Proporcional - Derivador (PD).
 - Proporcional – Integrador - Derivador (PID).

3.4.4.1. Controlador de acción Proporcional (P).

En este regulador la señal de accionamiento es proporcional a la señal de error del sistema. Si la señal de error es grande, el valor de la variable regulada es grande y si la señal de error del sistema es pequeña, el valor de la variable regulada es pequeño.

Es el más simple de todos los tipos de control y consiste simplemente en amplificar la señal de error antes de aplicarla a la planta o proceso. La función de transferencia de este tipo de control se reduce a una variable real, denominada K_p que determinará el nivel de amplificación del elemento de control. (Ver figura 3.29).

Llamando $y(t)$ a la señal de salida (salida del regulador) y $e(t)$ a la señal de error (entrada al regulador), en un control proporcional tendremos que la función de transferencia del bloque controlador (no la total del sistema), es la mostrada en la ecuación 3.6.

$$Y(S) / E(S) = K_p \quad \text{Ecuación 3.6}$$

Donde:

- $Y(S)$ es la salida del regulador o controlador en el dominio de Laplace.
- $E(S)$ es la señal de error en el dominio de Laplace.
- K_p es la ganancia del bloque de control.

Teóricamente, en este tipo de controlador, si la señal de error es cero, la salida del controlador también será cero. La respuesta, en teoría es instantánea, con lo cual el tiempo no interviene en el control. Sin embargo, en la práctica, esto no es así, de forma que, si la variación de la señal de entrada es muy rápida, el controlador no puede seguir dicha variación y seguirá una trayectoria exponencial hasta alcanzar la salida deseada.

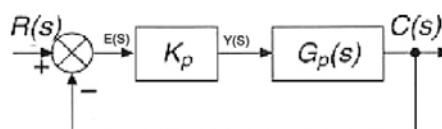


Figura 3.29. Controlador de acción Proporcional (P).

Una propiedad importante del regulador P es que como resultado de la rígida relación entre la señal de error del sistema y la variable regulada siempre queda alguna señal de error del sistema. El controlador P no puede compensar esta señal de error remanente (permanente) del sistema (señal de OFFSET).

3.4.4.2. Controlador de acción Integral (I).

En un controlador integral, la señal de salida del mismo varía en función de la desviación y del tiempo en que se mantiene la misma, o dicho de otra manera, el valor de la acción de control es proporcional a la integral de la señal de error. Esto implica que mientras que en la señal proporcional no influía el tiempo, sino que la salida únicamente variaba en función de las modificaciones de la señal de error, en este tipo de control la acción varía según la desviación de la salida y el tiempo durante el que esta desviación se mantiene.

Llamando $y(t)$ a la señal de salida (salida del regulador) y $e(t)$ a la señal de error (entrada al regulador), en un control integral tendremos que la función de transferencia del bloque controlador es la mostrada en la ecuación 3.7.

$$Y(S) / E(S) = K_i / S \quad \text{Ecuación 3.7}$$

Donde:

- $Y(S)$ es la salida del regulador o controlador en el dominio de Laplace.
- $E(S)$ es la señal de error en el dominio de Laplace.
- K_i es la pendiente de la rampa de acción integral, lo que implica que la velocidad de respuesta del sistema de control dependerá del valor de K_i .

El problema principal del controlador integral radica en que la respuesta inicial es muy lenta, y hasta pasado un tiempo, el controlador no empieza a ser efectivo. Sin embargo elimina el error remanente que tenía el controlador proporcional.

3.4.4.3. Controlador de acción proporcional e integral (PI).

En la práctica no existen controladores que tengan sólo acción integral sino que llevan combinada una acción proporcional. Estas dos acciones se complementan. La primera en actuar es la acción proporcional (instantáneamente) mientras que la integral actúa durante un intervalo de tiempo. Así y por medio de la acción integral se elimina la desviación remanente (proporcional). (Ver figura 3.30).

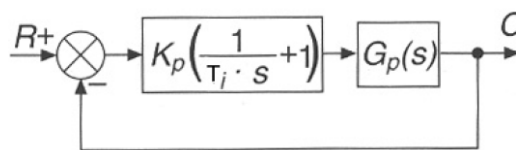


Figura 3.30. Controlador de acción Proporcional-Integral (PI).

La función de transferencia del bloque de control PI responde a la ecuación 3.8.

$$Y(S) / E(S) = K_p * (1 + 1 / T_i * S) \quad \text{Ecuación 3.8}$$

Donde:

- $Y(S)$ es la salida del regulador o controlador en el dominio de Laplace.
- $E(S)$ es la señal de error en el dominio de Laplace.
- K_p es la ganancia del bloque de control.
- T_i es el tiempo integral y controla la acción integral del sistema.

Si T_i es muy grande la pendiente de la rampa, correspondiente al efecto integral será pequeña y, por tanto, el efecto de esta acción suave, y viceversa. A T_i se le llama también tiempo de duplicación ya que es el tiempo que tarda la acción integral en igualar a la acción proporcional ante un error de tipo escalón.

La respuesta del controlador PI es la suma de las respuestas de un controlador proporcional y un controlador integral lo que proporciona una respuesta instantánea al producirse la correspondiente señal de error provocada por el control proporcional y un posterior control integral que se encargará de extinguir totalmente la señal de error.

3.4.4.4. Controlador de acción proporcional y derivativa (PD).

Esta acción, al igual que la integral, no se emplea solo, sino que va unida a la acción proporcional (PD). (Ver figura 3.31).

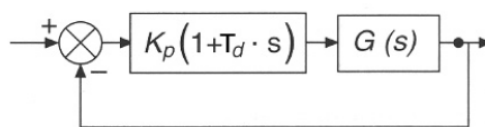


Figura 3.31. Controlador de acción Proporcional-Derivativo (PD).

En el control proporcional y derivativo PD, la función de transferencia del bloque de control responde a la siguiente ecuación 3.9.

$$Y(S) / E(S) = K_p * (1 + T_d * S) \quad \text{Ecuación 3.9}$$

Donde:

- $Y(S)$ es la salida del regulador o controlador en el dominio de Laplace.
- $E(S)$ es la señal de error en el dominio de Laplace.
- K_p es la ganancia del bloque de control.
- T_d es el tiempo derivativo o de adelanto, controla la acción derivativa del sistema.

A T_d se le llama también tiempo de duplicación ya que es el tiempo que tarda la acción proporcional en igualar el efecto de la acción derivativa ante una señal de error de tipo rampa (es una medida de la rapidez con que compensa un controlador PD un cambio en la variable regulada, comparado con un controlador P puro).

En este tipo de controladores, debemos tener en cuenta que la derivada de una constante es cero y, por tanto, en estos casos, el control derivativo no ejerce ningún efecto, siendo únicamente útil en los casos en los que la señal de error varía en el tiempo de forma continua.

Este tipo de controlador se utiliza en sistemas que deben actuar muy rápidamente, puesto que la salida está en continuo cambio. La acción derivativa por sí sola no se utiliza, puesto que para señales lentas, el error producido en la salida en régimen permanente es muy grande y si la señal de mando deja de actuar durante un tiempo largo la salida tenderá hacia cero y no se realizará entonces ningún control. La utilidad de este tipo de controlador radica en aumentar la velocidad de respuesta de un sistema de control, ya que, como se comentó en

los controladores proporcionales, aunque la velocidad de respuesta teórica de un controlador proporcional es instantánea, en la práctica no es así, pudiendo ser una rampa o una exponencial de una duración considerable.

Al incorporar a un controlador proporcional las características de un controlador derivativo, se mejora sustancialmente la velocidad de respuesta del sistema, a costa de una menor precisión en la salida (durante el intervalo de tiempo en que el control derivativo esté funcionando).

3.4.4.5. Controlador de acción PID.

Aprovecha las características de los tres reguladores anteriores, de forma, que si la señal de error varía lentamente en el tiempo, predomina la acción proporcional e integral y, si la señal de error varía rápidamente, predomina la acción derivativa. Tiene la ventaja de tener una respuesta más rápida y una inmediata compensación de la señal de error en el caso de cambios o perturbaciones. Tiene como desventaja que el bucle de regulación es más propenso a oscilar y los ajustes son más difíciles de realizar. (Ver figura 3.32).

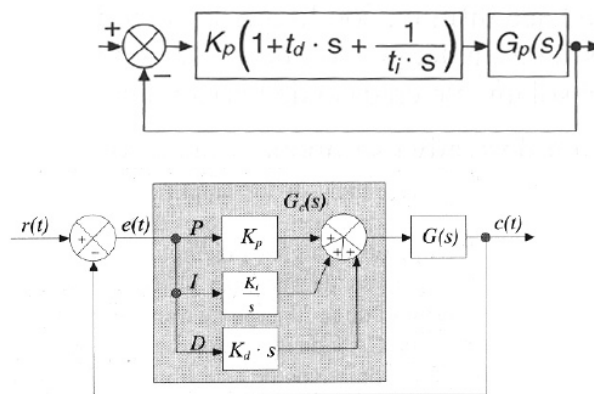


Figura 3.32. Controlador de acción Proporcional-Integral-Derivativo (PID).

La función de transferencia del regulador viene dada por la ecuación 3.10.

$$Y(S) / E(S) = K_p * (1 + T_d * S + 1 / T_i * S) \quad \text{Ecuación 3.10}$$

Donde:

- Y(S) es la salida del regulador o controlador en el dominio de Laplace.
- E(S) es la señal de error en el dominio de Laplace.
- Kp es la ganancia del bloque de control.
- Td es el tiempo derivativo o de adelanto, controla la acción derivativa del sistema.
- Ti es el tiempo integral y controla la acción integral del sistema.

Cuando se da una orden de cambio de magnitud, en una maniobra normal, la acción de control predominante del sistema es la proporcional, que aproximará la dirección al punto deseado de forma más o menos precisa. Una vez que la dirección esté cerca del punto deseado, comenzará la acción integral que eliminará el posible error producido por el control

proporcional, hasta posicionar el volante en el punto preciso. Si la maniobra es lenta, la acción derivativa no tendrá apenas efecto. Si la maniobra requiere mayor velocidad de actuación, la acción de control derivativo adquirirá mayor importancia, aumentando la velocidad de respuesta inicial del sistema y posteriormente actuará la acción proporcional y finalmente la integral. En el caso de una maniobra muy brusca, el control derivativo tomará máxima relevancia, quedando casi sin efecto la acción proporcional e integral, lo que provocará muy poca precisión en la maniobra.

3.4.5. CompactRIO.

3.4.5.1. Introducción.

El controlador de automatización programable (PAC) NI CompactRIO es un sistema reconfigurable de control y adquisición de bajo coste diseñado para aplicaciones que requieren alto rendimiento y fiabilidad. El sistema combina una arquitectura embebida abierta con un tamaño pequeño, extrema robustez y módulos industriales de E/S intercambiables en vivo. CompactRIO es alimentado por la tecnología de arreglo de compuertas programables en campo (FPGA) de E/S reconfigurable (RIO) [19].

Los sistemas CompactRIO consisten en chasis reconfigurable que aloja el FPGA programable por el usuario, módulos de E/S intercambiables en vivo, controlador en tiempo real para comunicación y procesamiento determinísticos y software gráfico LabVIEW para una rápida programación.

Las características que definen las virtudes del controlador CompactRIO son:

- Control Avanzado:

Con NI CompactRIO, el usuario puede desarrollar sistemas de control desde control simple de algoritmo proporcional integral y derivativo (PID) hasta control avanzado dinámico como un control predictivo de modelos. Puede ejecutar estos algoritmos de control de manera determinística y debido a la naturaleza de paralelismo del procesamiento FPGA, el añadir cálculos no reduce el rendimiento de la aplicación. Para sistemas de control de movimiento, NI SoftMotion ofrece la habilidad de crear controladores de movimiento personalizados para mejor rendimiento y flexibilidad.

- Medidas analógicas, de calidad y de alta velocidad:

National Instruments se ha especializado en medidas de alta calidad con diseño analógico de alto rendimiento. Varias aplicaciones requieren una combinación de medidas estáticas de baja velocidad, como temperatura o medidas dinámicas de alta velocidad, como sonido y vibración. NI CompactRIO proporciona varios tipos de medidas de alta calidad, todas en un solo sistema.

- Hardware embebido, confiable y robusto:

NI CompactRIO está diseñado para aplicaciones en entornos severos y lugares pequeños. El tamaño, el peso y la densidad del canal de E/S son requerimientos críticos de diseño en muchas aplicaciones embebidas. Al aprovechar el extremo rendimiento y tamaño pequeño de los dispositivos FPGA, CompactRIO es capaz de ofrecer habilidades sin precedentes de control y adquisición en un paquete compacto y robusto con certificaciones industriales y clasificaciones para operación en entornos industriales severos.

- **Procesamiento y análisis de señales:**

NI CompactRIO utiliza la plataforma de programación de diseño gráfico de sistemas de LabVIEW, la cual incluye miles de funciones avanzadas creadas específicamente para aplicaciones industriales de medidas y control. Puede usar estas herramientas para realizar fácilmente procesamiento de señales avanzado, análisis de frecuencia y procesamiento de señales digitales.

- **Plataforma de generación de prototipos flexible y modular:**

NI CompactRIO tiene una variedad de controladores, chasis reconfigurable y módulos de E/S de la Serie C intercambiables en vivo para brindar la flexibilidad requerida para ir desde generación de prototipos y desarrollo al mantenerse en la misma plataforma.

3.4.5.2. Arquitectura de la CompactRIO.

El sistema CompactRIO consiste en un chasis reconfigurable que aloja el FPGA programable por el usuario, módulos de E/S intercambiables en vivo, un controlador en tiempo real y una fuente de alimentación para dar energía a todo el sistema. (Ver figura 3.33).

- **Controlador Real-Time cRIO-9022.**

El NI cRIO-9022 cuenta con un controlador de 533 MHz industrial Freescale MPC8347 procesador en tiempo real determinístico y fiable para aplicaciones en tiempo real. Este controlador embebido está diseñado para extrema robustez, fiabilidad y bajo consumo de energía con dos entradas de 9 a 35VDC que proporcionan potencia aislada al chasis CompactRIO y un intervalo de temperaturas de -20° a 55°C. Con el 10/100Mbps/s y 10/100/1000Mbps/s Ethernet y los puertos de serie, puede comunicarse a través de TCP/IP, UDP, Modbus/TCP, y protocolos de serie. El cRIO-9022 también incorpora características de Web y servidores de archivos y un puerto Hi-Speed USB.

- **Fuente alimentación NI PS-15 Power Supply.**

El PS-15 NI es una fuente de alimentación industrial que cuenta con una larga vida útil, grandes reservas de energía y tamaño compacto. Las fuentes de 24V son ideales para alimentar cualquier sistema de National Instrumenta, así como gran cantidad de accesorios como controladores de motores DC. El sistema de montaje en carril DIN y terminales de abrazadera de resorte no requieren de herramientas y permiten hacer una instalación rápida y fácil.

- **Chasis reconfigurable cRIO-9113.**

El chasis reconfigurable de NI CompactRIO es el corazón del sistema CompactRIO. El Chasis es embebido y reconfigurable de 4 ranuras que acepta cualquier módulo de E/S CompactRIO. Esta complementado con un núcleo FPGA de E/S reconfigurable (RIO) Xilinx Virtex-5 para maximizar el poder de procesamiento. Dispone de la habilidad para sintetizar automáticamente circuitos personalizados de control y procesamiento de señales usando LabVIEW y permite opciones de montaje en carril DIN.

- Módulo I/O digital y bidireccional High-Speed NI 9401.**

El NI 9401 es un módulo de ocho canales, de entrada digital bidireccional 100ns, de la Serie C para NI CompactDAQ o cualquier chasis CompactRIO. Puede configurar la dirección de las líneas digitales en el módulo NI 9401 para entrada o salida por nibble (cuatro bits). Por lo tanto, se puede programar el módulo NI 9401 para tres configuraciones - ocho entradas digitales, ocho salidas digitales, o cuatro entradas digitales y cuatro salidas digitales. Con la tecnología E/S reconfigurable (RIO) (CompactRIO únicamente), puede utilizarse el Módulo NI LabVIEW FPGA para programar el módulo NI 9401 para aplicaciones personalizadas, contador de alta velocidad, temporizadores, comunicación digital, generación de pulsos, y mucho más. Cada canal es compatible con 5V / TTL señales y tiene características de aislamiento de 1.000 Vrms en transitorio entre los canales de E/S.



Figura 3.33. Componentes de la arquitectura de la CompactRIO: controlador Real-Time cRIO-9022, fuente alimentación NI PS-15 Power Supply, chasis reconfigurable cRIO-9113, módulo I/O digital y bidireccional High-Speed NI 9401.

Capítulo 4. Desarrollo del controlador para motores DCBL.

El proyecto “desarrollo de un controlador para motores DC *brushless* basado en CompactRIO y LabVIEW de National Instruments” se ha dividido en dos fases principales debido al tiempo que han exigido en su ejecución. La primera fase se explica que herramientas han sido necesarias estudiar, para poder aplicarlas con posterioridad, y en la segunda fase se describe el proceso íntegro de construcción del controlador.

4.1. Primera fase: aprendizaje de las herramientas del entorno de programación gráfica LabVIEW.

En esta primera fase de desarrollo del proyecto, es necesario adquirir conocimiento acerca de las herramientas que se usarán más adelante. Para ello, se han llevado a cabo tres cursos de aprendizaje de National Instruments sobre el programa LabVIEW. Cada curso consta de un manual teórico y un manual práctico que se describen a continuación.

4.1.1. Cursos de LabVIEW - manual teórico y práctico.

Estos manuales tienen como objetivo enseñar conceptos, técnicas, características, VI y funciones de programación de LabVIEW que puede utilizarse para crear aplicaciones de prueba y medida, adquisición de datos, control de instrumentos, registro de datos, análisis de medidas y generación de informes. Estos manuales presuponen que el usuario está familiarizado con Windows y que tiene experiencia en escribir algoritmos en forma de diagrama de flujos o diagramas funcionales (y que se ha realizado el curso LabVIEW Core 1 para el caso del curso de LabVIEW Core 2), (y que se ha realizado los cursos LabVIEW Core 1 y LabVIEW Core 2 para el caso del curso de LabVIEW FPGA). Los manuales de ejercicios y del curso están divididos en lecciones, descritas de este modo

En el manual teórico, cada lección consta de lo siguiente:

- Una introducción que describe el objetivo de la lección y lo que se aprenderá.
- Una descripción de los temas de la lección.
- Un cuestionario de resumen que prueba los conocimientos enseñados en la lección.

En el manual de ejercicios, cada lección consta de un conjunto de ejercicios para reforzar los conocimientos adquiridos de cada tema y de secciones de ejercicios opcionales y de retos o un conjunto de ejercicios adicionales para realizar si el tiempo lo permite.

4.1.1.1. LabVIEW Core 1. Temario y objetivos.

El primer curso de National Instruments contiene las siguientes lecciones [9]:

- Lección 1 – configuración del hardware.
- Lección 2 – explorando LabVIEW.
- Lección 3 – resolución de problemas y depuración de VI's.
- Lección 4 – implementación de un VI.
- Lección 5 – relacionar datos.
- Lección 6 – gestión de recursos.
- Lección 7 – desarrollo de aplicaciones modulares.
- Lección 8 – técnicas y modelos de diseño comunes.
- Lección 9 – uso de variables.

Al final de este curso, el usuario debería estar preparado para hacer las siguientes tareas:

- Comprender los paneles frontales, los diagramas funcionales, los iconos y los paneles conectores.
- Usar las estructuras de programación y los tipos de datos existentes en LabVIEW.
- Usar varias técnicas de edición y depuración.
- Crear y guardar VI's para poder utilizarlos como subVI's.
- Mostrar y registrar datos.
- Crear aplicaciones que utilicen dispositivos DAQ.
- Crear aplicaciones que usen instrumentos de puerto serie y GPIB.

4.1.1.2. LabVIEW Core 2. Temario y objetivos.

El segundo curso de National Instruments contiene las siguientes lecciones [10]:

Lección 1 – introducción.

Lección 2 – técnicas de sincronización.

Lección 3 – programación de eventos.

Lección 4 – gestión de errores.

Lección 5 – control de la interface de usuario.

Lección 6 – técnicas de E/S de fichero.

Lección 7 – mejora de un VI existente.

Lección 8 – creación y distribución de aplicaciones.

Al final de este curso, el usuario debería saber hacer las siguientes tareas:

- Aplicar patrones de diseño comunes que utilicen notificaciones, colas y eventos.
- Usar la programación de eventos de forma eficaz.
- Controlar programáticamente objetos de la interfaz de usuario.
- Evaluar formatos de E/S de archivos binarios y utilizarlos en aplicaciones.
- Modificar código existente para mejorar la usabilidad.
- Preparar, crear y desplegar aplicaciones independientes.

4.1.1.3. LabVIEW FPGA. Temario y objetivos.

El curso de FPGA de National Instruments contiene las siguientes lecciones [11]:

- Lección 1 – introducción.
- Lección 2 – crear FPGA VI's.
- Lección 3 – gestión de recursos.
- Lección 4 – ejecución de FPGA VI's.
- Lección 5 – depuración de FPGA VI's.

Al final de este curso, el usuario debería estar preparado para hacer las siguientes tareas:

- Ejecutar nuevas tareas y funciones propias del módulo FPGA.
- Usar varias técnicas de edición y depuración.
- Crear y guardar FPGA VI's para poder aplicarlos.
- Comunicarse con FPGA VI's.

4.2. Segunda fase: desarrollo del controlador de motor BLDC para el estudio de nuevos algoritmos de control.

A su vez esta segunda fase se divide en tres partes. En la primera parte se plantea unas condiciones o características iniciales, las cuales se necesitan para definir las bases del controlador. En la segunda parte se explica el diseño de la placa PCB donde se integran los sistemas de alimentación, los sistemas de protección y el *driver* de potencia. Y en la tercera parte se explica la elaboración del programa de control desarrollado con el lenguaje G dentro del entorno de desarrollo LabVIEW.

4.2.1. Planteamiento de las bases del controlador de motor BLDC basado en CompactRIO

Una vez concluido el proceso de aprendizaje de las herramientas que se pueden utilizar en LabVIEW, se procede a diseñar el controlador del motor DC *brushless*. Para empezar es necesario saber cuáles son los parámetros con los que se va a trabajar.

Lo primero es necesario observar las características más importantes del sistema CompactRIO que se utiliza. El sistema CompactRIO consiste en chasis reconfigurable que aloja el FPGA programable por el usuario, módulos de E/S intercambiables, controlador en tiempo real para comunicación y procesamiento determinísticos y software gráfico LabVIEW para rápida programación de FPGA. Al disponer de un FPGA, se puede desarrollar parte del programa aquí, lo que permitirá ejecutar instrucciones mucho más rápido que si se ejecutase en el controlador. Por otro lado, solo se disponen de módulos NI 9401. Estos módulos disponen de ocho canales digitales bidireccionales de alta velocidad. Esta condición implica que todas las señales de entrada y de salida sean digitales.

Lo segundo es necesario observar las características del motor DCBL que se usa. Accediendo a su hoja técnica se comprueba que el sistema de medición de la posición del eje del motor se realiza a través de tres sensores *Hall*, los cuales necesitan de una toma de tierra y de una tensión de alimentación específica de 5 voltios. También se observa que para dar potencia al motor, y que éste pueda girar, es necesario alimentar las tres fases que lo componen. Debido a las distintas propiedades mencionadas, el motor necesita de ocho entradas y salidas para funcionar, cinco son para los sensores *Hall* y tres para las fases del motor.

El método más eficiente y económico para alimentar las tres fase del motor es diseñar un inversor trifásico. Como consecuencia directa de usar un inversor trifásico para controlar los interruptores de cada rama y que las I/O del modulo NI 9401 sean digitales, se debe programar seis señales PWM distintas, una para cada transistor MOSFET. Además los sensores de efecto *Hall* están situados a una distancia de 120° entre ellos, por tanto programar una conducción a 120° es la forma más fácil de diseñar las distintas señales PWM que activan los interruptores del inversor.

Lo tercero es necesario tomar decisiones sobre las distintas posibilidades de programar el controlador, teniendo en cuenta las condiciones anteriormente subrayadas. Para poder controlar el motor se necesita manipular dos parámetros que son la “velocidad real” y la “velocidad consigna”, los cuales se convierten en las variables principales del programa. A partir de estas variables, se necesita un control PID que regule la velocidad real igualándola a la velocidad consigna a través de las señales PWM (ver figura 4.1).

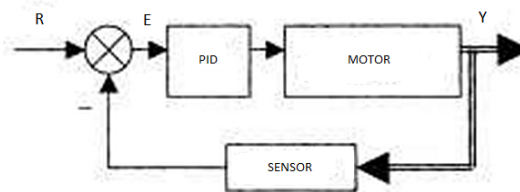


Figura 4.1. Control de lazo cerrado a través de un PID.

4.2.2. Desarrollo de la Placa PCB.

El diseño de la placa PCB está dispuesto de tal forma que se puedan realizar varias funciones. El bloque principal es el inversor trifásico junto con los *drivers* de los transistores de potencia de tipo MOSFET. Los otros bloques secundarios son el regulador de 12V, necesario para alimentar el *driver* de los transistores MOSFET y el regulador de 5V, necesario para alimentar los sensores *Hall* del motor DC *brushless*. La placa está diseñada para trabajar con un rango de tensiones de entrada situado entre los 12V y los 48V, de tal forma que tanto el motor como la placa PCB funcionan con la misma tensión. Además está diseñada para soportar una corriente máxima de 10A. (Ver planos del nº1 al nº4).

Esta placa está basada en un diseño anterior previamente desarrollado en el Centro de Automática y Robótica, en el que el objetivo era dar suministro de potencia a un motor DC con escobillas. Esta versión anterior se compone de los mismos componentes que la placa desarrollada para este proyecto: un inversor de dos ramas (puente H), un *driver* para los transistores MOSFET y un regulador de 12V.

Por tanto, el diseño de la placa PCB para un motor DC brushless recoge las bases de proyectos anteriormente realizados para motores con escobillas en el Centro de Automática y Robótica, para adaptarlos a nuevos proyectos aplicados a motores DC brushless.

4.2.2.1. Inversor trifásico.

Como se explica en el capítulo 3.3, el inversor trifásico se compone principalmente por dispositivos electrónicos de potencia, que trabajan como interruptores operando en corte y conducción con una secuencia apropiada para obtener tres tensiones de salida simétricas y balanceadas. En este proyecto las señales de entrada a los interruptores son de tipo PWM, las cuales deben ser tratadas para el correcto funcionamiento del inversor trifásico.

En la figura 4.2 se muestran los componentes asociados al inversor trifásico. Para éste se usan como interruptores dos tipos de transistores MOSFET distintos. Los transistores que están conectados a la tensión de alimentación son los irf540n y los que están conectados a tierra son los irc540pbf. Ambos transistores son de potencia y tienen unas condiciones de trabajado muy similares: una rápida conmutación, altas temperaturas de trabajo, baja resistencia $R_{DS(on)}$, etc. La única diferencia notable entre estos dos transistores está en que el dispositivo irc540pbf dispone de dos patillas extra que se usan como sensor de corriente.

Los diodos conectados en antiparalelo a los transistores MOSFET son los RHRP860. Estos son componentes fijos en el diseño de puentes, los cuales proveen un camino para la corriente que circula por el motor en el momento de la conmutación. Estos diodos deben ser de recuperación rápida y deben manejar la misma corriente que los transistores.

Por último, se tienen las resistencias de 47Ω en paralelo con los diodos MUR120. Esta combinación de componentes permite modificar los tiempo de encendido y apagado del *gate* del transistor. Durante el encendido el *gate* se carga lentamente a través de la resistencia y en el apagado, se descarga rápidamente a través del diodo MUR120. Esta combinación funciona como sistema de seguridad para evitar la conducción simultanea de los transistores superiores e inferiores.

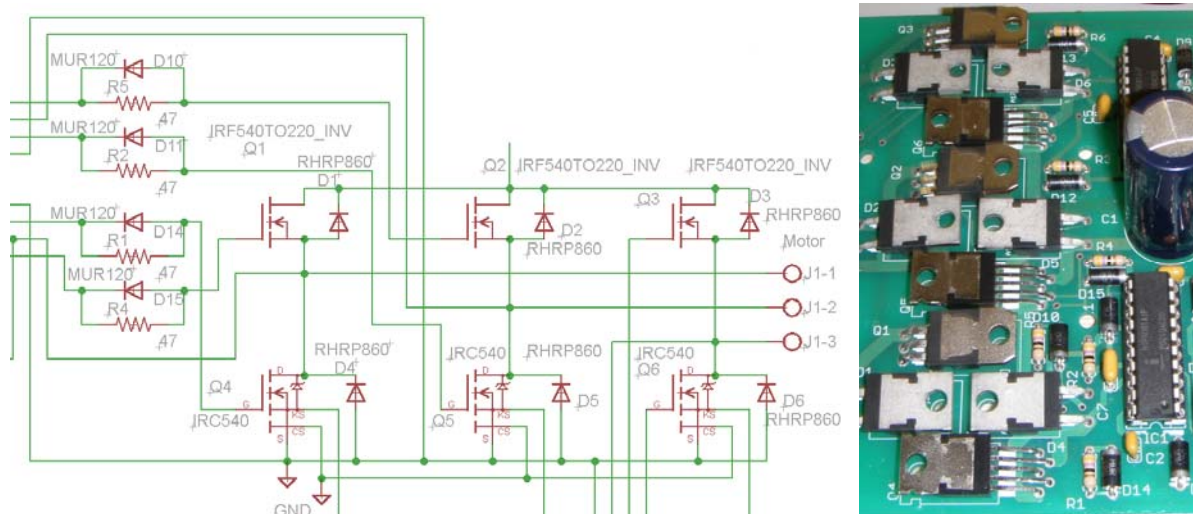


Figura 4.2. Esquema eléctrico del inversor trifásico.

Para poder acondicionar la señal PWM que llega del controlador y que debe entrar a los MOSFET, se ha introducido el componente HIP4081A. Este componente está diseñado para proporcionar las señales de *gate* de los transistores, cumpliendo con sus requisitos de corriente, además de realizar la translación de voltajes para la activación de los transistores superiores. También ofrece controlar los tiempos muertos entre las transiciones de nivel alto a nivel bajo y viceversa de los transistores superiores e inferiores. El control de retardo de propagación es un elemento importante de la HIP4081A. Dos sub-circuitos idénticos dentro del chip retrasan la conmutación de la puerta del MOSFET de potencia. Para ello, se coloca una resistencia entre el pin HDEL y VSS que controla los retrasos de conmutación superior y la resistencia entre el pin LDEL y VSS que controla el retraso de conmutación inferior. Cada uno de las resistencias establece una corriente que es inversamente proporcional al retraso creado. El retraso creado por estas resistencias se convierte en el tiempo muerto, necesario para evitar la conducción simultánea de los transistores superiores e inferiores [21].

El valor de las resistencias que determinan el tiempo muerto se calcula a partir de la gráfica mostrada a continuación (ver figura 4.3). Esta gráfica se encuentra en el *datasheet* del HIP4081A y en ésta se muestra el valor del tiempo muerto en función de la resistencia que se conecte. Para este proyecto se ha decidido colocar unas resistencias con un valor de 220K Ω cada una, lo que supone un tiempo muerto aproximado de 100ns. Este valor es más que suficiente para los transistores empleados.

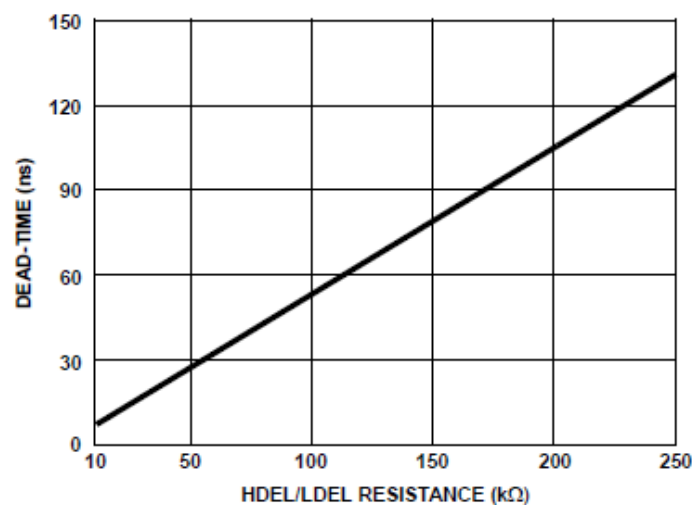


Figura 4.3. Gráfica de cálculo del tiempo muerto en función de las resistencias HDEL y LDEL.

Para conseguir una correcta tensión del motor se ha decidido emplear la función de *bootstrap* del *driver* HIP4081A. Esta función permite disponer de una fuente flotante de voltaje asociada al transistor superior. La alimentación *bootstrap* está formada por un diodo (D_{BS}) y un condensador *bootstrap* (C_{BS}). Este método tiene como ventajas que es muy simple y barato; pero tiene inconvenientes impuestos por el proceso de carga del condensador. Estos son que el tiempo que está conduciendo el transistor de la parte de arriba está limitado porque depende del tiempo que tarda en descargarse el condensador C_{BS} , y que la frecuencia máxima de trabajo se limita al tiempo que dura la carga de dicho condensador.

Partiendo del funcionamiento de una sola rama del inversor trifásico, el circuito *bootstrap* opera de la siguiente forma:

- Durante el tiempo que el transistor Q1 está en 'OFF', y el transistor Q2 está en 'ON', el condensador C_{BS} se carga a un valor de tensión aproximadamente de $+V_{DD}$ (ver figura 4.4). La carga es muy rápida porque la t_{CARGA} es muy pequeña. Ésta se calcula a partir de la ecuación 4.1.

$$t_{CARGA} = C_{BS} * (r_{DS(ON)Q2} + r_{dDes}) \quad \text{Ecuación 4.1}$$

Dónde:

- t_{CARGA} es el tiempo de carga del condensador C_{BS} (seg).
 - C_{BS} es la capacidad del condensador C_{BS} (F).
 - $r_{DS(ON)Q2}$ es la resistencia en 'ON' del MOSFET Q2 (Ω).
 - r_{dDes} es la resistencia en directo del diodo D_{BS} (Ω).
- Cuando Q2 pasa a 'OFF' y se desea que Q1 pase a estado 'ON', el diodo D_{BS} queda polarizado en inverso (ver figura 4.4). Con la tensión almacenada en C_{BS} durante el estado anterior, se polariza al driver superior del circuito de control con el que se excita a Q1, de tal forma que pasa a estado 'ON'.

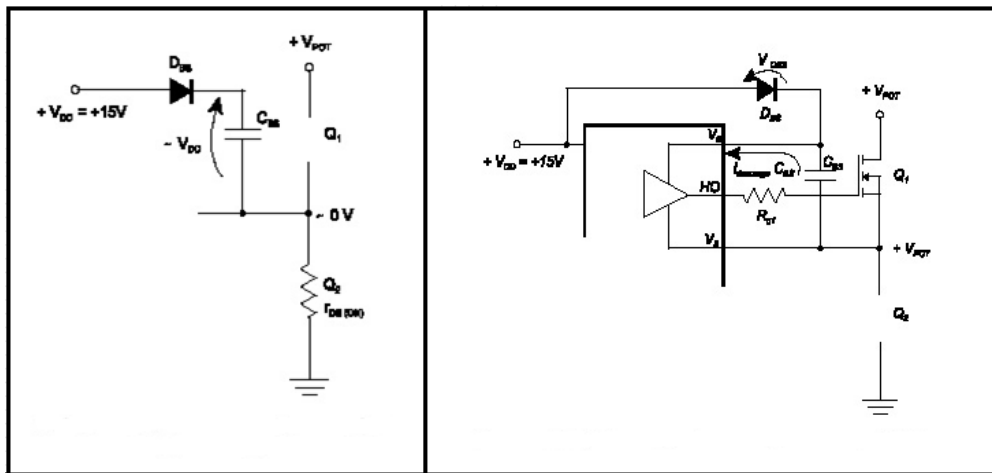


Figura 4.4. Circuito equivalente en el proceso de carga y descarga del C_{BS} .

Entonces se deduce que interesa que C_{BS} sea muy pequeña para que se cargue lo más rápidamente posible, pero por otro lado interesa que C_{BS} sea muy grande para que tarde mucho tiempo en descargarse. Por tanto es necesario tomar una solución de compromiso. Prácticamente, el condensador C_{BS} debe ser de un tamaño 10 veces superior a la capacidad del condensador del *gate* del transistor del MOSFET. De forma analítica este condensador se puede calcular a partir de la ecuación 4.2.

$$C_{BS} = \frac{Q_G + Q_{rr} + \frac{(I_{DR} + I_{QBS})}{f_{PWM}}}{V_{BS1} - V_{BS2}}$$

Ecuación 4.2

Donde:

- I_{DR} es la corriente inversa de pérdidas del diodo *bootstrap*.
- I_{QBS} es la corriente de polarización que excita al transistor de la parte alta del puente.
- Q_{rr} es la carga de recuperación en inversa del diodo *bootstrap*.
- Q_G es la carga transferida a la carga en el paso de corte a conducción.
- f_{PWM} es la frecuencia de la señal PWM.
- V_{BS1} es la tensión que tiene el condensador C_{BS} inmediatamente después del refresco.
- V_{BS2} es la tensión que tiene el condensador C_{BS} inmediatamente antes del refresco.
- C_{BS} es la capacidad *bootstrap*.

De esta ecuación se obtiene que los valores de los componentes necesarios para hacer funcionar correctamente el sistema. El diodo MUR120 y un condensador C_{BS} de 100nF cumplen con estas condiciones. El esquema eléctrico se muestra en la figura 4.5. En esta figura solo aparece un chip HIP4081A, pero es necesario usar dos. Esto es debido a que con un chip HIP4081A solo se puede controlar dos ramas en un puente H. Para este proyecto, el inversor trifásico consta de tres ramas, luego es necesario disponer de dos HIP4081A para poder controlar los seis transistores MOSFET.

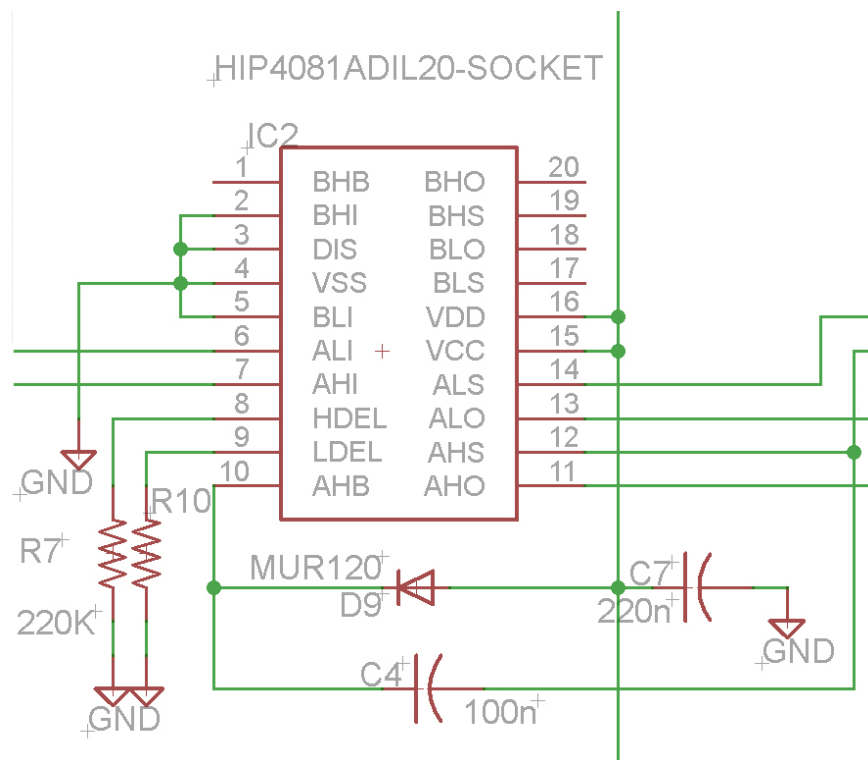


Figura 4.5. Esquema eléctrico del driver del MOSFET.

4.2.2.2. Regulador de 12 voltios.

Para diseñar el regulador de 12V, primero se tiene que tener en cuenta la tensión con la que se alimenta los demás bloques eléctricos. Para este diseño se utiliza el circuito integrado LM2575. Este dispositivo es un regulador reductor de conmutación (*buck*). El regulador de 12V necesita de un número mínimo de componentes para completar el circuito regulador de voltaje. Para calcular los valores de estos componentes se debe acceder al *datasheet* del circuito integrado LM2575. En éste se describe como construir el regulador. El esquema eléctrico se muestra en la figura 4.6. A la salida del regulador se ha añadido una resistencia en serie con un LED para poder indicar que el regulador está suministrando correctamente los 12 voltios que necesitan los demás bloques eléctricos [22].

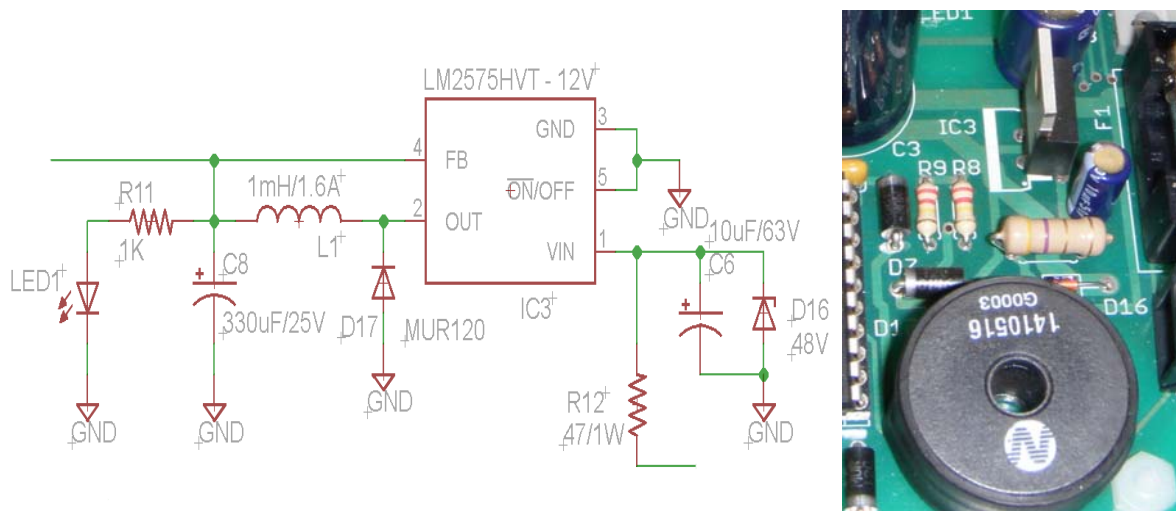


Figura 4.6. Esquema eléctrico del regulador de 12 voltios.

4.2.2.3. Regulador de 5 voltios.

El regulador de 5 voltios es necesario para que funcionen los sensores de efecto *Hall* del motor DC *brushless*. Accediendo a la hoja técnica del motor DCBL utilizado, se comprueba los requerimientos eléctricos de los sensores, los cuales necesitan de una tensión de 5V para que trabajen correctamente. El dispositivo utilizado es el AP78L05, un regulador positivo de tres terminales con tensión de salida fija de 5V. Para que los sensores *Hall* funcionen correctamente es necesario introducir una resistencia *pull-up* en serie con su salida. Esta resistencia de polarización asegura que la señal lógica que proviene del sensor no quede en un estado flotante. En algunos tipos de dispositivos lógicos, si no se pusieran las resistencias *pull-up*, el estado lógico 1 podría quedar con un valor de tensión intermedio entre cero y uno y confundirse su estado. Son resistencias normales, solo llevan el nombre *pull-up* por la función que cumplen [23]. (Ver figura 4.7).

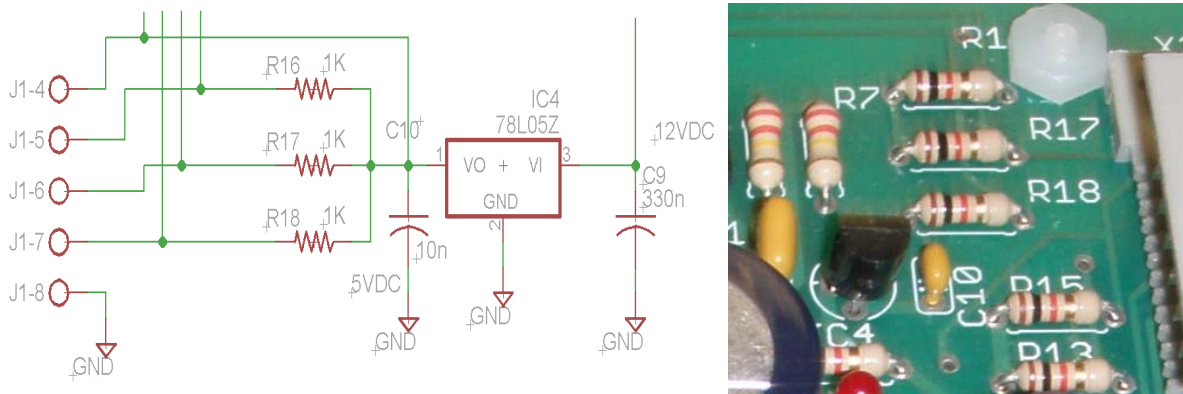


Figura 4.7. Esquema eléctrico del regulador de 5 voltios.

4.2.2.4. Conectores.

Para poder comunicar la placa PCB con los demás sistemas, se acoplan a la propia placa dos conectores distintos. El conector de 15 *pines* y el jumper de 8 *pines*. El conector de 15 *pines* tiene como objetivo comunicar la placa PCB con los módulos NI 9401 de la CompactRIO a través de un *bus*. El jumper de 8 *pines* trata de conectar la placa PCB con el motor DC *brushless* y los sensores de efecto *Hall*. En las tablas 4.1 y 4.2 se muestran como están conectadas las distintas señales indicando el pin asociado de cada conector. Es importante que todas las tomas a tierra sean la misma, para ello se han unido todas ellas a través de las distintas conexiones entre la placa PCB y el resto del sistema.

Jumper	Conexión	Descripción
J1	cable negro	Conexión entre la placa y la fase A del motor
J2	cable rojo	Conexión entre la placa y la fase B del motor
J3	cable blanco	Conexión entre la placa y la fase C del motor
J4	cable verde	Conexión entre la placa y la señal de Vcc Hall
J5	cable rojo/gris	Conexión entre la placa y la señal del sensor Hall C
J6	cable negro/gris	Conexión entre la placa y la señal del sensor Hall B
J7	cable blanco/gris	Conexión entre la placa y la señal del sensor Hall A
J8	cable azul	Conexión entre la placa y la señal de GND Hall

Tabla 4.1. Descripción del conexionado del jumper de 8 pines entre la placa PCB y el motor DCBL.

PIN	Descripción
pin 1	conexión de las señal de salida PWM A-high que proviene del modulo NI 9401 con la placa PCB
pin 2	conexión de las señal de salida PWM A-low que proviene del modulo NI 9401 con la placa PCB
pin 3	conexión de las señal de salida PWM B-high que proviene del modulo NI 9401 con la placa PCB
pin 4	conexión de las señal de salida PWM B-low que proviene del modulo NI 9401 con la placa PCB
pin 5	conexión de las señal de salida PWM C-high que proviene del modulo NI 9401 con la placa PCB
pin 6	conexión de las señal de salida PWM C-low que proviene del modulo NI 9401 con la placa PCB
pin 7	conexión de la señal de entrada del sensor de corriente A que proviene de la placa PCB con el modulo ni 9401
pin 8	conexión de la señal de entrada del sensor de corriente B que proviene de la placa PCB con el modulo ni 9401
pin 9	conexión de la señal de entrada del sensor de corriente C que proviene de la placa PCB con el modulo ni 9401
pin 10	conexión de la señal de entrada del sensor Hall C que proviene de la placa PCB con el modulo ni 9401
pin 11	conexión de la señal de entrada del sensor Hall B que proviene de la placa PCB con el modulo ni 9401
pin 12	conexión de la señal de entrada del sensor Hall A que proviene de la placa PCB con el modulo ni 9401
pin 13	conexión entre el modulo NI 9401 y la tensión del regulador de 5V
pin 14	conexión entre el modulo NI 9401 y la tensión del regulador de 12V
pin 15	conexión entre la tierra que proviene del modulo NI 9401 y la tierra de la fuente de alimentación de la placa PCB

Tabla 4.2. Descripción del conexionado del conector de 15 pines entre la placa PCB y los dos módulos NI 9401 de National Instruments.

Para desarrollar tanto el esquema eléctrico como el *layout*, se ha utilizado el programa EAGLE. Este programa contiene un editor de esquemas y un editor de diseño de PCB. El editor de esquemas es necesario para el diseño de esquemas eléctricos. Las piezas se pueden colocar en distintas hojas y ser conectados entre sí a través de puertos. El editor de diseño de PCB permite volver a la anotación de esquema y el enrutamiento automático para conectar los dispositivos automáticamente sobre la base de las conexiones definidas en el esquema eléctrico.

4.2.3. Programa de control desarrollado con el lenguaje G dentro del entorno de desarrollo LabVIEW de National Instruments.

El programa del control del motor está compuesto por tres códigos distintos. Estos códigos tienen una jerarquía que le da una clara importancia dependiendo del nivel en el que están. En la figura 4.8 se muestra como están distribuidos los diferentes códigos dentro de la CompactRIO.

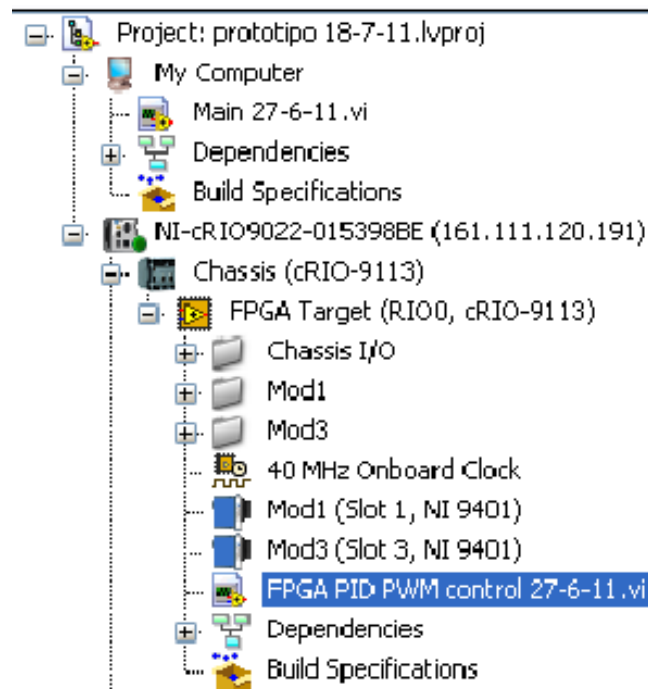


Figura 4.8. Jerarquía del programa de control.

Como se observa el código *main.vi* es el programa principal. Éste se ejecuta dentro del ordenador, aunque puede compilarse dentro del controlador. Esto es debido a que la velocidad de comunicación entre la CompactRIO y el ordenador es suficientemente más grande que la velocidad con la que se ejecutan las instrucciones dentro del código *main.vi*, de esta forma la compilación es más rápida. El segundo código es *FPGA PID PWM control.vi*. Éste se compila dentro de la FPGA debido a que este código desarrolla funciones que deberán ejecutarse lo más rápido posible para obtener el mayor control posible. Por último está el código *discrete PID.vi*, que se compila como una librería el cual se ejecuta cada vez que el código *FPGA PID PWM control.vi* acceda a ella.

4.2.3.1. Código Main.vi

El código *main.vi* es el que permite al usuario interactuar con el controlador. Con este código se puede manejar las variables de control de tal forma que se les puedan asignar el valor que mas interese, pero también se puede recibir información de lo que sucede dentro del código a través de las variables indicadores, de modo que el usuario pueda responder en consecuencia (ver plano nº5 y nº6). Este código tiene tres funciones principales:

- El primero consiste en inicializar todos los objetos y variables. Es importante que todas las variables tengan un valor determinado (aunque sea cero), evitando que el programa asigne aleatoriamente un valor a cualquier variable no inicializada. En la tabla 4.3 se muestran todas las variables, indicadores y constantes dentro de este código y se explica para que sirve cada una de ellas. En paralelo, se inicializa un archivo TDMS que sirve para poder recoger los datos que se vayan desarrollando durante la ejecución del programa. Además hay que añadir que todas las variables que entran dentro del código *FPGA PID PWM control.vi* se comportan como variables de control. Análogamente sucede lo mismo con todas las variables que entran en el código *discrete PID.vi*. por el contrario todas las variables que viene del código *discrete PID.vi* y llegan al código *FPGA PID PWM control.vi* se comportaran con variables indicadores, y sucede lo mismo con las variables que vienen del código *FPGA PID PWM control.vi* y llegan al código *main.vi*.
- La segunda parte del código consiste en ejecutar un bucle infinito que se ejecuta cada 100ms y que únicamente termina cuando se pulsa el interruptor “stop” o se detecta un error interno dentro del FPGA. Dentro de este bucle se desarrolla la parte principal de este código. Al inicio del éste, las variables se inicializan con los valores del ciclo anterior del bucle, pudiendo haber sido modificadas o haber sido mantenidas. Las variables más importantes que requieren un control más exhaustivo son “PID gains”, “velocidad consigna (rpm)”, “tiempo de muestreo (ms)” y “control de frecuencia PWM (Hz)”. La variable “PID gains” pasa por un bloque funcional que consiste en transformar las variables Kp, Ti y Td a las variables Kp, Ki y Kd. Después estas variables son introducidas al código *FPGA PID PWM control.vi*, de tal forma que este código pueda trabajar con los nuevos valores introducidos. Por último, del FPGA se sacan los valores del código *FPGA PID PWM control.vi* y se escriben en los indicadores asignados, de tal forma que se pueda operar y comprobar cuál ha sido el resultado obtenido en un ciclo del bucle. Al mismo tiempo los valores de estos indicadores y de las variables de control son copiados y guardados en el archivo TDMS.
- La tercera y última parte consiste en cerrar el programa. Si se ha producido cualquier error o se ha pulsado el interruptor “stop”, entonces se procede a terminar las funciones de escritura de datos en el archivo TDMS, a cambiar los resultados de las variables por valores seguros que no implican algún acción peligrosa y a cerrar el programa.

código	nombre	tipo	descripción
main.vi	control de frecuencia PWM	DBL	Es la frecuencia con la que se quiere caracterizar las señales PWM.
	PID gains	cluster	Es la estructura que guarda las constantes del PID Kp, Ti y Td.
	Ts	DBL	Es el tiempo de muestreo que debe asignarse al PID en seg.
	scaled gains	cluster	Es la estructura que guarda las constantes del PID Kp, Ki y Kd.
	tiempo de muestreo	FXP	Es el tiempo que tarda en ejecutarse el bucle "cálculo de velocidad" en mseg.
	velocidad consigna	DBL	Es la velocidad que se quiere que tenga el motor en rpm.
	velocidad real	I16	Es un indicador de la velocidad real, en rpm.
	Duty Cycle	DBL	Es un indicador del porcentaje que esta activa la señal PWM, en %.
	Direction Line	BOOL	Es un indicador del sentido de giro del motor.
FPGA PID PWM control.vi	rps 2	I16	Es la velocidad instantánea que tiene el motor, en rps.
	velocidad consigna	I16	Es la velocidad que proviene del código main.VI, en rps.
	velocidad PID ganancias	cluster	Es la estructura que guarda las constantes del PID Kp, Ti y Td.
	salida PID	I32	Es un indicador del valor porcentual de la salida PWM.
	PWM period	I32	Es la frecuencia característica de la señal PWM, en ticks.
	PWM High Time	I32	Es un indicador del tiempo que la señal PWM debe estar a nivel alto, en ticks
	contador del periodo	I32	Es un indicador del contador que se ejecuta en el bucle "generación PWM".
	sentido de giro PWM	BOOL	Es un indicador del sentido que debe tener la señal PWM.
	señal PWM	BOOL	Es un indicador de la forma de la señal PWM.
	entradas hall - A/B/C	BOOL	Son las señales que provienen de los sensores Hall.
	codigo hall	U8	Es un indicador que convierte las tres señales Hall en un número entero.
	sentido de giro de la velocidad real	BOOL	Es un indicador del sentido de giro que tiene la velocidad del motor.
	fase	I16	Es un indicador del estado en el que se encuentra la máquina de estados.
	salida A/B/C high	BOOL	Es un indicador del valor de las tres señales PWM que llegan a los MOSFET superiores.
	salida A/B/C low	BOOL	Es un indicador del valor de las tres señales PWM que llegan a los MOSFET inferiores.
	pulsos	U32	Es un indicador del contador que se incrementa cada vez que cambia la variable "código hall".
var.pulsos	U32	Es un indicador de la variación de pulsos durante un ciclo del bucle "cálculo de velocidad".	

código	nombre	tipo	descripción
Discrete PID.vi	set point	I16	Es la velocidad de consigna que proviene FPGA PID PWM contro.vi.
	process variable	I16	Es la velocidad real que proviene FPGA PID PWM contro.vi.
	PID gains	cluster	Es la estructura que guarda las constantes del PID Kp, Ti y Td y que viene del FPGA PID PWM contro.vi.
	output range	cluster	Es la estructura que contiene los límites de la señal de salida del PID.
	output	I16	Es el indicador del valor de salida del PID.

Tabla 4.3. Descripción de las variables usadas en el programa.

4.2.3.2. Código FPGA PID PWM control.vi

El código *FPGA PID PWM control.vi* es el más importante del programa (ver plano nº7 y nº8). Sin duda alguna, el sistema funciona gracias al análisis y operación de las distintas instrucciones que componen este código. Para poder describir de una manera eficiente este código, éste se ha dividido en cuatro partes. En realidad estas cuatro partes son cuatro subcódigos que trabajan en paralelo en el que todos están relacionados entre sí, compartiendo información entre ellos.

1. La primera parte corresponde al bucle “lazo de control PID”. En este bucle entran como variables de control: “velocidad consigna”, “velocidad real”, “velocidad PID ganancias”, “PWM period (ticks)” y “rango de velocidad de lazo PID”. Y como salida se tiene el indicador “PWM high time (ticks)” y “rango de salida”.

Primero es importante indicar la velocidad con la que se quiere ejecutar este bucle. Éste requiere velocidades muy altas, debido a que la salida PWM high time (ticks) debe estar continuamente actualizada. Por eso se requiere una variable que controle de la velocidad del bucle, de tal forma que se consiga una mejor y rápida respuesta del sistema. Hay que tener en cuenta que cada uno de los cuatro bucles de este código requieren una mayor prioridad de ejecución que otros. Es por eso que se define el bucle “lazo de control PID” con una necesidad de prioridad alta.

El funcionamiento de este bucle consiste en introducir las variables de control “velocidad consigna”, “velocidad real” y “velocidad PID ganancias” al bloque PID discrete, que es otro código que se explicará más adelante. De este código se obtiene la variable “salida PID”. Con el valor de “salida PID” y de “PWM period (ticks)” se calcula el valor porcentual de la variable indicador “PWM high time (ticks)”. Esta variable permitirá al siguiente bucle obtener la señal de salida que se necesitará para activar las distintas fases del motor DCBL. (Ver figura 4.9).

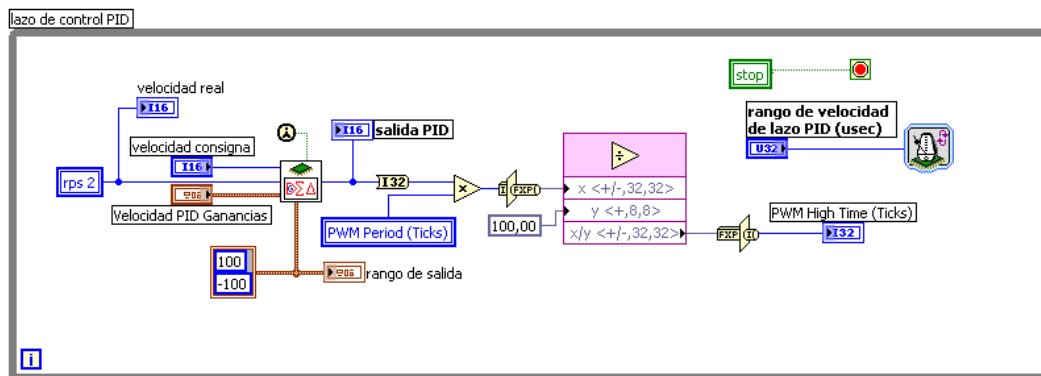


Figura 4.9. Bucle “Lazo de control PID”.

- La segunda parte corresponde al bucle “generación PWM”. En este bucle entran como variables de control: “PWM period (ticks)” y “PWM high time (ticks)”. Y como salida se tienen los indicadores “sentido de giro PWM” y “señal PWM”. Además se añade un indicador gráfico, el cual nos muestra a través de una gráfica en tiempo real el valor de la variable “señal PWM”.

Respecto a la velocidad con la que se ejecuta este bucle, se debe indicar que se usa un tipo de bucle distinto. Esto es debido a dos razones. La primera es porque se sabe de las características del FPGA de la CompactRIO, que la velocidad del FPGA es de 40MHz. Y la segunda es porque se sabe que la frecuencia del PWM es de 20 KHz, la cual ha sido predeterminada en el código “main.vi”. Por ello se inicializa el bucle insertándole al bucle una constante de reloj con el valor de 40 MHz para que éste se repita a esta velocidad.

Partiendo de que el bucle se ejecuta a la velocidad de 40MHz, se crea la variable “contador del periodo” que se incrementa una unidad cada vez que se ejecuta el bucle. Este incremento se mantendrá hasta que su valor sea igual al valor de la variable “PWM period (ticks)” la cual está fijada de antemano al valor de 20KHz. Cuando se produzca esta regla, la variable “contador del periodo” se reiniciará a cero y volverá a empezar a incrementarse.

Por otro lado, se tiene que calcular la señal PWM con su signo. Para ello, se usa el valor de “PWM high time (ticks)”, obtenido del bucle “lazo de control PID”. Para calcular el valor de la variable “sentido de giro PWM” se debe comparar el valor de “PWM high time (ticks)” con cero, de tal forma que si “PWM high time (ticks)” es mayor o igual que cero, “sentido de giro PWM” será positivo y si “PWM high time (ticks)” es menor que cero, “sentido de giro PWM” será negativo. Para calcular el valor de “señal PWM”, se debe comparar el valor absoluto de “PWM high time (ticks)” con el valor de la variable “contador del periodo”. Si “PWM high time (ticks)” es mayor que “contador del periodo”, entonces “señal PWM” será positivo. Y si “PWM high time (ticks)” es menor o igual que “contador del periodo”, entonces “señal PWM” será negativo. Con estas simples reglas se ha construido la señal PWM. Con el valor de esta variable y el bucle “maquina de estados” se construirá las seis señales PWM necesarias. (Ver figura 4.10).

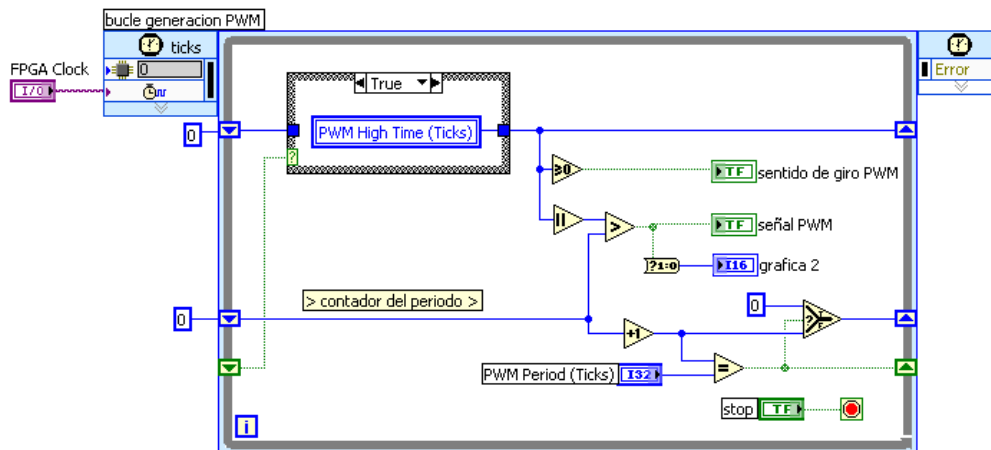


Figura 4.10. Bucle “generación PWM”.

- La tercera parte corresponde al bucle “maquina de estados”. En este bucle entran como variables de control: “sentido de giro PWM”, “señal PWM”, “entradas hall – A/B/C” y “count(usec)”. Y como salida se tienen los indicadores “sentido de giro de la velocidad real”, “salidas A/B/C high” y “salidas A/B/C high”.

Para empezar, lo primero es definir en qué consiste la variable “entradas hall – A/B/C”. Ésta proviene de la señales generadas por lo sensores *Hall* a través del módulo NI9401. Al empezar el bucle estas tres señales se juntan para formar una variable de tipo entero. Esta variable se llama “código hall”, la cual nos indica qué sensores están activados y cuáles no, en función de la tabla 3.3 explicada en el capítulo 3.2. Esta variable permite al bucle desarrollar tres funciones.

La primera consiste en generar las seis señales PWM que se deben enviar al *driver* de potencia. Para decidir cuando un transistor esta activado y cuando no, se debe diseñar un sistema de desfase de señales. En función de la variables “código hall” y “sentido de giro PWM”, el *switch* “salida bobinas” define cuáles de las seis señales (A-high, B-high, C-high, A-low, B-low y C-low) están activadas y cuáles no están activadas. Para construir este *switch*, se utiliza las tablas 3.3 y 3.4, las cuales indican que señales se deben activar en función de los sensores *Hall* y del sentido de giro de la señal PWM generada. Para definir totalmente las seis señales PWM, es necesario introducir en seis puertas AND (una para cada señal), las salidas del *switch* “salida bobinas” y la variable “señal PWM”, la cual define las características físicas de la señal PWM explicada en el apartado anterior. Después de aplicar la instrucción de la puerta AND, se envían las señales PWM a través del módulo NI 9401. Para poder comprobar que las señales se construyen correctamente, se observa los indicadores “salidas A/B/C high” y “salidas A/B/C high”. (Ver figura 4.11).

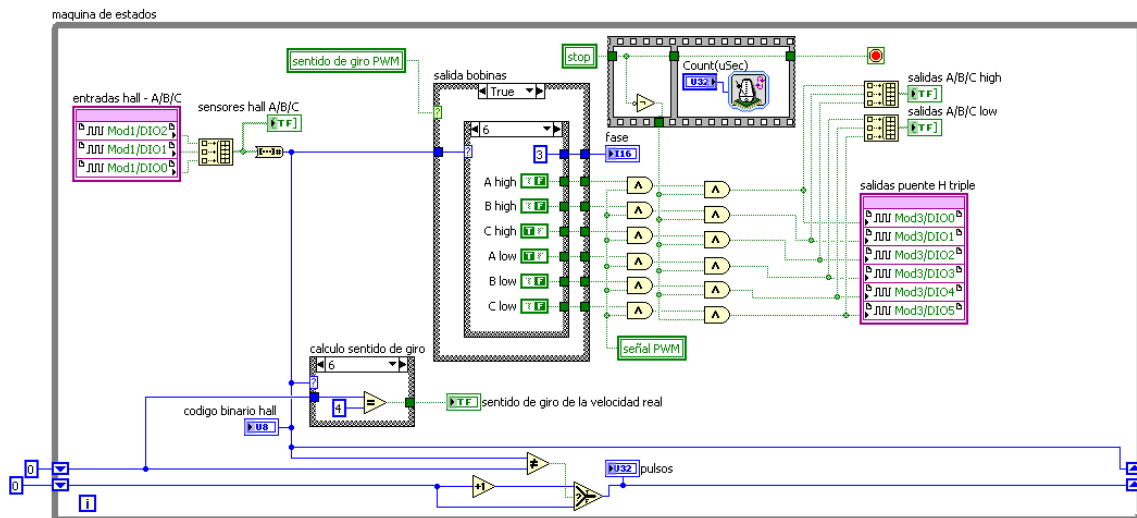


Figura 4.11. Bucle “máquina de estados”.

La segunda consiste en definir cuál es el sentido de giro de la velocidad real del motor. Para ello, es necesario volver a aplicar los datos de la tabla 3.3. A partir de ésta, el *switch* “calcula sentido de giro” define si la variable “sentido de giro de la velocidad real” está a nivel alto o bajo en función del valor de la variable “código hall”. Dependiendo del valor de “código hall”, dentro del *switch* se compara si el valor del ciclo anterior de “código hall” es igual a la constante prefijada con la ayuda de la tabla 3.3. En caso afirmativo, la variable “sentido de giro de la velocidad real” estará a nivel alto, y viceversa. Para conseguir el valor anterior, se coloca un registro de desplazamiento que memorice el valor que tuvo la variable asignada en el ciclo anterior. (Ver figura 4.12).

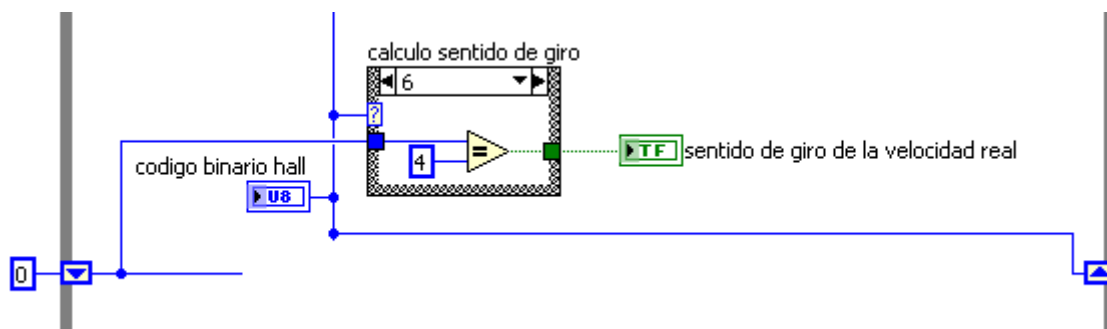


Figura 4.12. Cálculo del sentido de giro de la velocidad real.

La tercera consiste en generar un contador. Este contador llamado “pulsos” incrementa una unidad su valor cada vez que se modifica el valor de la variable “código hall”. El tipo de variable utilizada para este contador no tiene problemas de desbordamiento. Es importante añadir, que este bucle incorpora un mecanismo de seguridad. Este consiste en poder a cero las seis señales PWM si ocurre algún error o se pulsa el botón de “stop”. De esta forma se evita que el motor siga funcionando una vez finalizado el programa. (Ver figura 4.13).

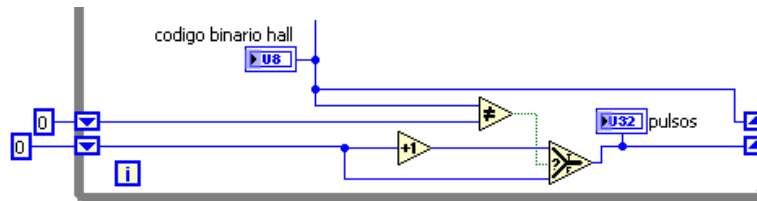


Figura 4.13. Contador de pulsos.

4. La cuarta parte corresponde al bucle “calculo de velocidad”. En este bucle entran como variables de control: “pulsos”, “tiempo de muestreo” y “sentido de giro de la velocidad real”. Y como salida se tienen los indicadores “var.pulsos” y “rps 2”.
- Para calcular la velocidad, se calcula el desplazamiento del motor en un tiempo controlado. Para empezar se definirá el objeto de la variable pulsos. Esta variable cuenta el número de veces que cambia la variable “código hall”. Como esta variable solo tiene seis estados posibles, esto significa que en una revolución se tiene que haber producido seis cambios de estado, que en definitiva consiste en que cada cambio de estado el motor se desplaza 60° . Por tanto si se fija de antemano el valor de “tiempo de muestreo”, se calculo el número de pulsos durante este intervalo de tiempo (“var.pulsos”) y por último se divide el valor de “var.pulsos” entre “tiempo de muestreo”, se obtiene la velocidad del motor en valor absoluto.
- Para calcular el verdadero valor de la velocidad real (“rps 2”), se usa un comparador. Si el valor de la variable “sentido de giro de la velocidad real” es positiva, entonces “rps 2” es positivo (sentido de giro horario). Y si el valor de “sentido de giro de la velocidad real” es negativo, entonces “rps 2” es negativo (sentido de giro antihorario). Además al bucle se le incorpora un filtro que calcula una media ponderada de la velocidad del motor. Este mecanismo de seguridad permitirá asegura que no se produzcan picos en el valor de la variable “rps 2” y por tanto que se produzcan picos de corriente en la placa PCB. (Ver figura 4.14).

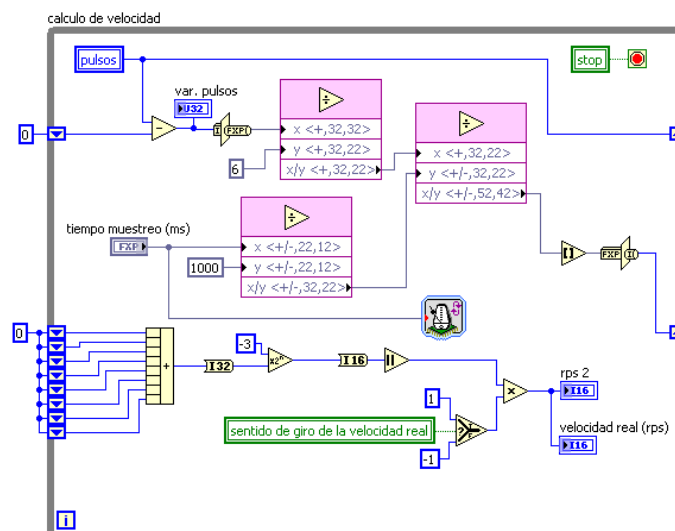


Figura 4.14. Bucle “calculo de velocidad”.

4.2.3.3. Código Discrete PID.vi

El código *discrete PID.vi* es el código que realmente hace la función de control (ver plano nº9 y nº10). Para definir esta librería hay que dividirlo en dos partes la primera consiste en crear la señal de error y la segunda parte consiste en operar y calcular la señal de control. (Ver figura 4.15).

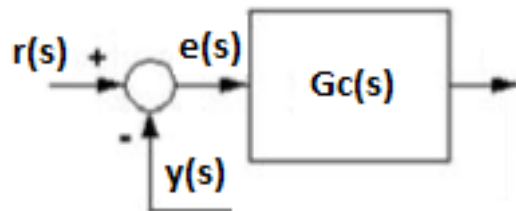


Figura 4.15. Tareas que ejecuta el código discrete PID.vi.

1. La primera parte del código consiste en calcular la señal de error. Como se define en el capítulo 3.4, la señal de error es necesaria para más tarde poder obtener la señal de control. La señal de error se define en la ecuación 4.3.

$$E(s) = R(s) - Y(s)$$

Ecuación 4.3

Donde:

- $Y(s)$ es la señal de process variable (“velocidad real”).
 - $R(s)$ es la señal de set point (“velocidad consigna”).
 - $E(s)$ es la señal de error.
2. Una vez obtenido el valor de la señal de error, se pasa a calcular la señal de control. Para ello, además se necesita conocer los valores de las constantes del regulador K_p , K_i y K_d , y los límites superior e inferior que se quiere que tenga la señal de control. Para conseguir el valor de la señal de control se debe aplicar el algoritmo expresado en la ecuación 3.10, la cual relaciona la señal de control con la señal de salida a través de las constantes del regulador.
Una vez calculado el valor de la señal de control, se compara el valor con los límites superior e inferior de salida, de tal forma que si se sobrepasa cualquiera de ellos, el valor de la señal de control se cambia por el valor del límite superado. Los valores de los límites de salida están fijados a través de unas constantes con los valores -100 y 100. Esto es debido a que queremos que la señal de control sea porcentual para más tarde aplicarla al bucle “generación PWM”.

5. Experimentación y resultados.

5.1. Introducción.

En este capítulo se presentan los experimentos realizados sobre un motor Maxon – EC32 *brushless* [20], aplicando el código de programación elaborado. Con estos experimentos se intenta demostrar que es posible controlar dichos motores a través del lenguaje G de LabVIEW y aplicado sobre el controlador CompactRIO.

Para ello se realiza cuatro experimentos. Estos experimento simulan básicamente las acciones que puede ejecutar un motor DC *brushless* en una aplicación real. Estos cuatro experimentos se dividen en dos partes: la primera parte consta de dos experimentos en los que se trabaja sin carga. El eje del motor DC *brushless* gira sin oposición de forma libre, de tal modo que se mantiene la potencia generada en todo momento.

La segunda parte consta de los dos experimentos restantes. Estos dos experimentos tratan de comprobar el funcionamiento del motor DCBL con una carga acoplada al eje y de las transiciones de sin carga y con carga.

Durante los experimentos se han realizado pequeñas pruebas de ensayo y error para comprobar cuáles son los mejores valores para los distintos parámetros, de tal forma que cuando se ha realizado los cuatro experimentos, se han aplicado unos valores fijos. En la tabla 5.1 se muestra que parámetros han sido prefijados de antemano después de realizar las pruebas anteriormente mencionadas, con sus respectivos valores. Esto significa que cada uno de los cuatro experimentos realizados se inicializa con unos valores predeterminados, lo cual ayuda a que el sistema completo programa-controlador-motor funcione correctamente.

Para calcular los valores de las constantes del regulador PID, se ha aplicado el método de sintonización “Ziegler y Nichols”, el cual se explica en el capítulo 5.2.

variables	valores
frecuencia PWM (Hz)	20000
Tv - tiempo de muestreo de velocidad (ms)	25
Tb - tiempo de repetición del bucle PID (μ s)	5000
Ts - tiempo de muestreo del PID (s)	0.1
Kc - constante proporcional	0.1
Ti - tiempo integral	0.005
Td - tiempo derivativo	0.0125

Tabla 5.1. Variables prefijadas para su posterior aplicación en los experimentos.

5.2. Método de Oscilación de Ziegler y Nichols.

La sintonización de los controladores Proporcional - Integral - Derivativo o simplemente controladores PID, consiste en la determinación del ajuste de sus parámetros (K_c , T_i , T_d), para lograr un comportamiento del sistema de control aceptable y robusto de conformidad con algún criterio de desempeño establecido [7].

En la industria de procesos, la mayor parte de los controladores se utilizan para responder a un cambio en la perturbación, y se requiere de una buena regulación para seguir el cambio en el valor deseado, con excepción de los controladores esclavos en los sistemas de control en cascada.

Si el controlador se ha sintonizado para lograr una buena respuesta en un cambio en el valor deseado, no eliminará las perturbaciones en forma efectiva, si el sistema contiene un integrador o si su constante de tiempo es grande.

Este procedimiento es válido solo para plantas estables a lazo abierto y se lleva a cabo siguiendo los siguientes pasos:

1. Utilizando sólo control proporcional, comenzando con un valor de ganancia pequeño, incrementar la ganancia hasta que el lazo comience a oscilar. Notar que se requieren oscilaciones lineales y que éstas deben ser observadas en la salida del controlador.

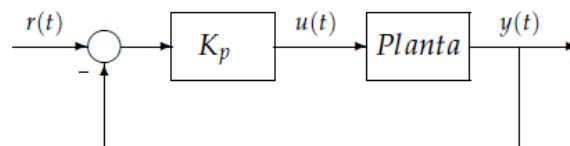


Figura 5.1. Lazo cerrado solo con ganancia proporcional.

2. Registrar la ganancia crítica del controlador $K_p = K_c$ y el periodo de oscilación de la salida del controlador, P_c . (En el diagrama de Nyquist, corresponde a que $K_c G(j\omega)$ cruza el punto $(-1, 0)$ cuando $K_p = K_c$).
3. Ajustar los parámetros del controlador según la tabla 5.2.

	K_p	T_i	T_d
P	$0,5 * K_c$	—	—
PI	$0,45 * K_c$	$P_c / 1,2$	—
PID	$0,6 * K_c$	$0,5 * P_c$	$P_c / 8$

Tabla 5.2. Parámetros de ajuste (método de oscilación).

5.3. Experimentos sin carga.

Los dos primeros experimentos se realizan juntos y comparten las mismas necesidades. Como se muestra en la figura 5.2, los dos experimentos necesitan únicamente del *driver* de potencia y el motor DC *brushless*. Estos dos experimentos consisten en aplicar una señal escalón de subida y otra señal escalón de bajada sobre el motor DC *brushless*, de tal forma que se pueda comprobar si el motor es capaz de aguantar cambios bruscos de velocidad. Esta señal escalón se comporta como una variación de velocidad infinita, donde la velocidad de consigna introducida a través de la interface visual pasa de un valor de 4000 rpm a un valor de 9000 rpm para el escalón de subida, y de un valor de 9000 rpm a un valor de 4000 rpm para el escalón de bajada.

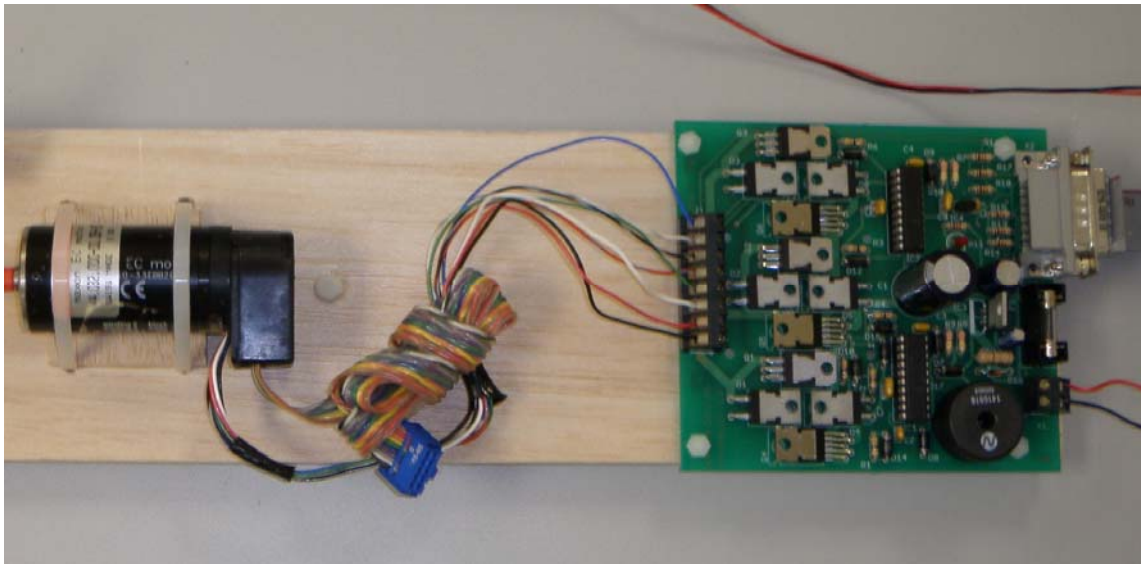


Figura 5.2. Esquema físico del *driver* de potencia – motor DC *brushless*.

En la figura 5.3 se muestra la gráfica que representa los dos primeros experimentos. En la gráfica se observa como la velocidad real se modifica a través del regulador PID, para poder tener el mismo valor que la velocidad de consigna, ya sea con el escalón de subida o con el escalón de bajada.

Para el experimento del escalón de subida el regulador aumenta el porcentaje de señal PWM, para así poder aumentar la velocidad del motor. Una vez superada la velocidad de consigna, la salida del regulador oscila amortiguadamente hasta que las dos velocidades se igualan. Para el escalón de bajada sucede análogamente la misma situación pero a la inversa. El regulador disminuye la señal de PWM para reducir la velocidad.

En esta gráfica se observa una situación peculiar, ya que ambos experimentos no son simétricos. En el caso del escalón de bajada, ocurre que el tiempo que tardan en igualarse las velocidades es mayor que para el caso del escalón de subida. Esto es debido a que en los cambios de velocidades negativos, la inercia acumulada por el motor es mayor que en los cambios de velocidades positivos.

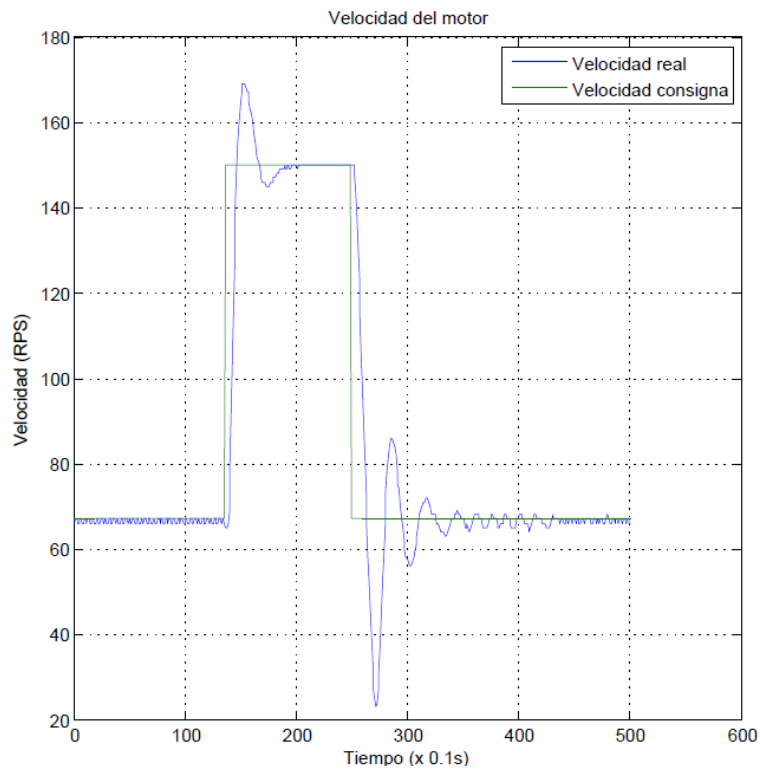


Figura 5.3. Gráfica del experimento de escalones de subida y bajada.

5.4. Experimentos con carga

El tercer experimento consiste en evaluar el comportamiento que tiene el motor DC *brushless* ante la perturbación de un impulso. Este experimento intenta simular la situación en la que un motor debe responder ante la aplicación momentánea de una carga. Para ello se induce al motor una velocidad de 4000 rpm. Cuando la velocidad real sea constante se aplica una carga instantánea. En la figura 5.4 se muestra como está predispuesto el experimento, el cual también sirve para realizar el cuarto experimento. La etapa de potencia y el motor *brushless* siguen igual, pero se añaden dos nuevas partes que conforman la carga. Al eje de motor DCBL se le conecta un segundo motor DC de escobillas, y a las escobillas del segundo motor se le conecta en paralelo un interruptor y una resistencia de gran potencia. Esta disposición permite aplicar una carga al motor DCBL a través del accionamiento del interruptor. Cuando el interruptor no está pulsado, el motor DC de escobillas, que funciona como generador, no tiene ningún componente para liberar la energía acumulada. Pero cuando está pulsado el interruptor, el motor DC de escobillas libera la energía en la resistencia de gran potencia, a la vez que demanda energía del motor DCBL, el cual aumenta la velocidad de giro para mantener el par motor.

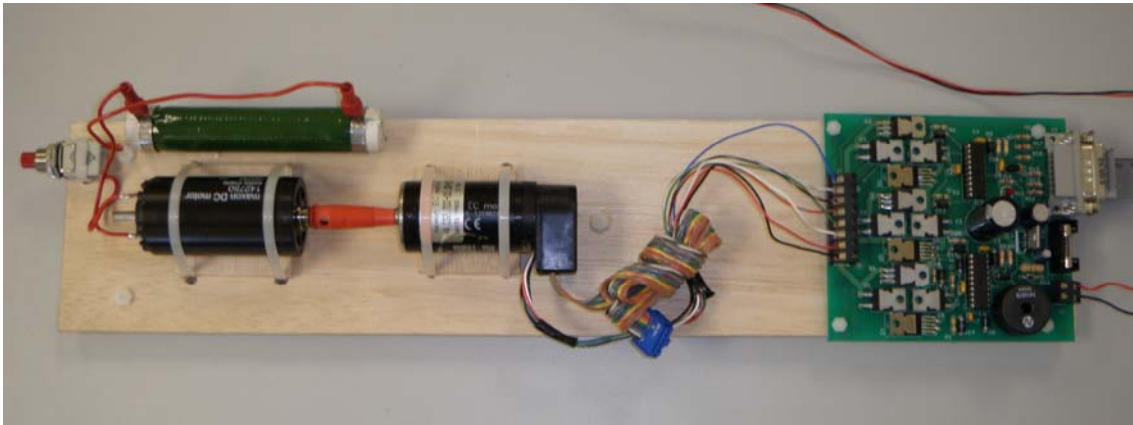


Figura 5.4. Esquema físico del *driver* de potencia – motor DC *brushless* - carga.

En la figura 5.5 se muestra la gráfica del tercer experimento. En esta se observa como el sistema funciona correctamente, ya que el controlador actúa rápidamente para mantener la velocidad real igual a la velocidad consigna. En el momento del impulso de la carga, la velocidad real decae debido a la necesidad de una mayor potencia que no dispone. Para poder conseguir dicha potencia, la corriente aumenta, de tal modo que se vence la carga. Como este proceso sucede casi instantáneamente, la variaciones de velocidad y corriente son pequeñas, de tal modo que en cuanto se suelta el interruptor, el sistema vuelve a las condiciones iniciales de sin carga.

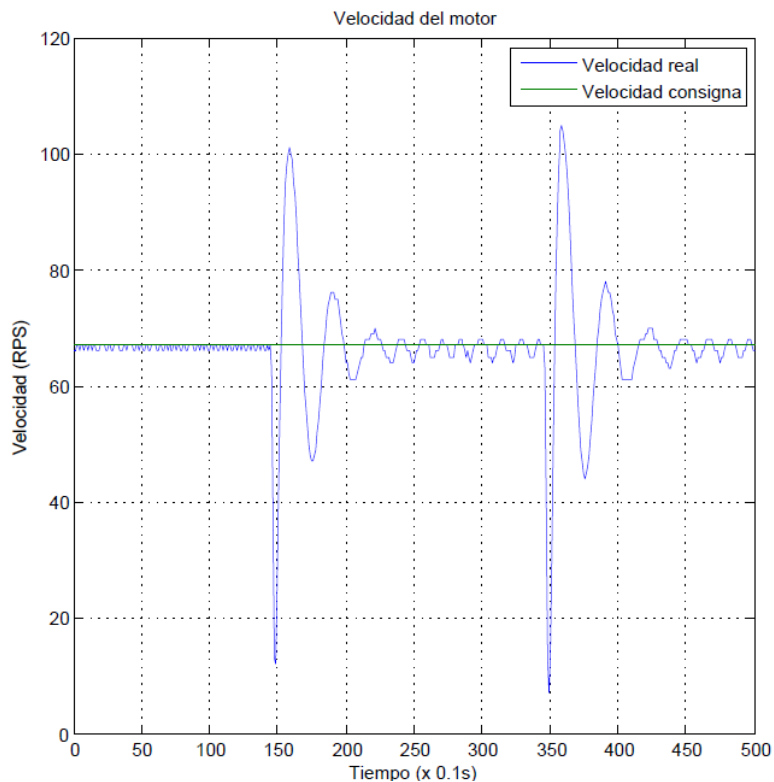


Figura 5.5. Gráfica del experimento de perturbación de impulso.

En el último y cuarto experimento se aplica la misma estructura explicada anteriormente y mostrada en la figura 5.4. En este experimento se prueba la capacidad del motor de soportar los cambios de carga y sin carga, de tal forma que se simula la actuación de un motor DC *brushless* que tiene que funcionar en los cambios de sin carga – con carga y viceversa. Para ello, se predispone al motor a una velocidad consigna de 4000 rpm. Una vez estable la velocidad real, se realiza una activación de la carga a través del interruptor, como si de una señal escalón de subida se tratase. Cuando la velocidad real se estabiliza, se desactiva la carga simulando una señal escalón de bajada.

En la figura 5.6 se muestra la cuarta gráfica. En ella se puede observar como el sistema permanece estable en todo momento. Al igual que sucedía con el tercer experimento, al activarse la carga, la corriente eléctrica aumenta, disminuyendo por el contrario la velocidad momentáneamente. Pero al mantenerse la carga, la corriente eléctrica sigue aumentando, luego la potencia generada aumenta, por tanto la velocidad crece. El controlador PID actúa hasta que la carga se iguala con la potencia generada, y la velocidad real con la velocidad consigna.

Para la desactivación de la carga sucede al contrario. La corriente eléctrica disminuye mientras la velocidad real se dispara debido a la inercia acumulada. Al seguir disminuyendo la corriente, la potencia y la velocidad real disminuyen, hasta que el controlador PID iguala las velocidades.

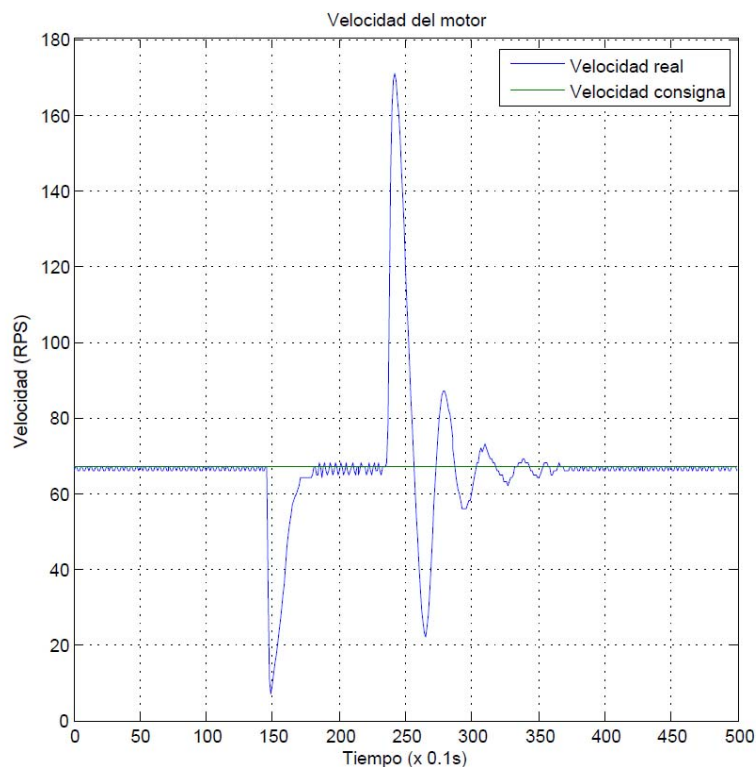


Figura 5.6. Gráfica del experimento de perturbación de escalón.

Con estos tipos de experimentos, se obtiene información acerca de los condiciones de trabajo del motor y del controlador, de las prestaciones y de las necesidades que se requiere para un correcto funcionamiento, las cuales se explican detenidamente en el capítulo 7.

Capítulo 6. Presupuesto.

6.1. Introducción.

Para poder desarrollar este proyecto hay que tener en cuenta que se ha utilizado distintos tipos de materiales. En este capítulo se explica con detalle que componentes se han usado y el precio que tienen, dando una valoración final del presupuesto total. Para comentar el presupuesto, éste se ha dividido en cinco secciones claramente diferenciadas. La primera sección se explica el desglose presupuestario del personal. En este caso se describe el coste de las horas trabajadas para este proyecto. La segunda sección se detalla el *hardware* utilizado. Para este proyecto se desglosa las distintas partes que constituye la CompactRIO. En la tercera sección se desarrolla los costes del *software*. Dicho *software* es el entorno de programación gráfica LabVIEW con sus módulos Real-Time y FPGA. En la cuarta sección se explica qué componentes eléctricos se han utilizado para construir la placa PCB, la cual consta del regulador de 12V, el regulador de 5V y el *driver* de potencia del motor DC *brushless*. Por último, en la quinta sección se hace un resumen global del presupuesto de todas las secciones y se expone un presupuesto final.

El presupuesto total de este proyecto asciende a la cantidad de 20.728,20 €



1	Unidad	SECCIÓN 01.- PERSONAL	Cantidad	Precio unitario	Precio total
				€	€
01.01	horas	HORAS REALIDAS DURANTE ESTE PROYECTO	500	11,36	5.681,81
		<p>Este proyecto se ha llevado a cabo en el Centro de Automática y Robótica del CSIC. Para poder realizar dicho proyecto en el centro se ha tenido que crear una relación con el departamento del SOPP de la universidad Carlos III de Madrid. Esta relación conlleva algunos acuerdos. De entre todos ellos, existe un acuerdo que limita el número de horas que el alumno puede realizar para llevar a cabo el proyecto en esta empresa, y por tanto influye en el cálculo del presupuesto. Este acuerdo explica que el número máximo de horas permitidas es de 500.</p>			
					4.659,08
		TOTAL SECCIÓN 1			
		TOTAL SECCIÓN 1 + 18% IVA			5.681,81



2	Unidad	SECCIÓN 02.- HARDWARE	Cantidad	Precio unitario €	Precio total €
02.01	Ud.	<p>CONTROLADOR Real-Time cRIO-9022</p> <p>El NI cRIO-9022 cuenta con un controlador de 533 MHz industrial Freescale MPC8347 procesador en tiempo real determinístico y fiable para aplicaciones en tiempo real. Este controlador embebido está diseñado para extrema robustez, fiabilidad y bajo consumo de energía con dos entradas de 9 a 35VDC que proporcionan potencia aislada al chasis CompactRIO y un intervalo de temperaturas de -20° a 55°C. Con el 10/100Mbps/s y 10/100/1000Mbps/s Ethernet y los puertos de serie, puede comunicarse a través de TCP/IP, UDP, Modbus/TCP, y protocolos de serie. El cRIO-9022 también incorpora características de Web y servidores de archivos y un puerto Hi-Speed USB.</p>	1	2.849,00	2.849,00
02.02	Ud.	<p>FUENTE ALIMENTACIÓN NI PS-15 Power Supply</p> <p>El PS-15 NI es una fuente de alimentación industrial que cuenta con una larga vida útil, generosas reservas de energía y tamaño compacto. Las fuentes de 24V son ideales para alimentar cualquier sistema de National Instruments, así como gran cantidad de accesorios como controladores de motores DC. El sistema de montaje en carril DIN y terminales de abrazadera de resorte no requieren de herramientas y permiten hacer una instalación rápida y fácil. Amplia gama de voltajes de selección automática de entrada y ayudas para evitar los errores del usuario. El rango de temperaturas de funcionamiento y la extraordinaria inmunidad EMI permite un funcionamiento sin problemas incluso en condiciones difíciles.</p>	1	199,00	199,00
02.03	Ud.	<p>CHASIS RECONFIGURABLE cRIO-9113</p> <p>El chasis reconfigurable de NI CompactRIO es el corazón del sistema CompactRIO. El Chasis es embebido y reconfigurable de 4 ranuras que acepta cualquier módulo de E/S CompactRIO. Esta complementado con un núcleo FPGA de E/S reconfigurable (RIO) Xilinx Virtex-5 para maximizar el poder de procesamiento. Dispone de la habilidad para sintetizar automáticamente circuitos personalizados de control y procesamiento de señales usando LabVIEW y permite opciones de montaje en carril DIN.</p>	1	1.799,00	1.799,00
02.04	Ud.	<p>MÓDULO I/O digital y bidireccional High-Speed NI 9401</p> <p>El NI 9401 es un modulo de ocho canales, de entrada digital bidireccional 100ns, de la Serie C para NI CompactDAQ o cualquier chasis CompactRIO. Puede configurar la dirección de las líneas digitales en el NI 9401 para entrada o salida por <i>nibble</i> (cuatro bits). Por lo tanto, se puede programar el NI 9401 para tres configuraciones - ocho entradas digitales, ocho salidas digitales, o cuatro entradas digitales y cuatro salidas digitales. Con la tecnología E/S reconfigurable (RIO) (CompactRIO únicamente), puede utilizarse el Módulo NI LabVIEW FPGA para programar el NI 9401 para aplicaciones personalizadas, contador de alta velocidad, temporizadores, comunicación digital, generación de pulsos, y mucho más. Cada canal es compatible con 5V / TTL señales y tiene características de aislamiento de 1.000 Vrms en transitorio entre los canales de E/S.</p>	2	249,00	498,00
		TOTAL SECCIÓN 2			4.383,00
		TOTAL SECCIÓN 2 + 18% IVA			5.345,00



3	Unidad	SECCIÓN 03.- SOFTWARE	Cantidad	Precio unidad €	Precio total €
03.01	Ud.	PROGRAMA LabVIEW Profesional LabVIEW Completo es un entorno de desarrollo gráfico estándar e la industria para crear aplicaciones profesionales científicas y de ingeniería. Incluye todas las características de LabVIEW Base, así el usuario puede integrarse perfectamente con cualquier dispositivo de hardware y construir sus propias interfaces de usuario al programar con íconos gráficos e intuitivos y cables que parecen un diagrama de flujo. Además el usuario puede tener acceso a miles de funciones específicas para ingeniería como análisis de frecuencia, ajuste de curvas y más. También puede realizar post procesamiento o análisis en línea en tiempo real en señales adquiridas.	1	4.549,00	4.549,00
03.02	Ud.	MODULO NI LabVIEW Real-Time El Módulo LabVIEW Real-Time ayuda a desarrollar aplicaciones que se ejecutan de manera confiable y determinística como sistemas autónomos en dispositivos en tiempo real como PXI, NI Compact FieldPoint y NI CompactRIO, así como PCs de escritorio estándares. El SO embebido en tiempo real para estos objetivos es un solo kernel dedicado que proporciona máxima fiabilidad para código embebido.	1	2.549,00	2.549,00
03.03	Ud.	MODULO NI LabVIEW FPGA El Módulo NI LabVIEW FPGA le ayuda a configurar arreglos de compuertas programables en campo (FPGAs) en <i>hardware</i> sin la complejidad de VHDL. Usted puede crear VIs de FPGA embebidos que combinan acceso directo a E/S con la lógica de LabVIEW definida por el usuario. Las aplicaciones incluyen dispositivos personalizados para rápida generación de prototipos, simulaciones de alta velocidad, protocolos de comunicación personalizados y control de lazo cerrado.	1	2.549,00	2.549,00
TOTAL SECCIÓN 3					7.911,00
TOTAL SECCIÓN 3 + 18% IVA					9.647,00



4	Unidad	SECCIÓN 04.- COMPONENTES ELÉCTRICOS	Cantidad	Precio unidad €	Precio total €
04.01	Ud.	RESISTENCIA 47Ω (1W)	1	0,40	0,40
04.02	Ud.	RESISTENCIA 47Ω (1/4W)	6	0,05	0,30
04.03	Ud.	RESISTENCIA 1KΩ (1/4W)	7	0,05	0,35
04.04	Ud.	RESISTENCIA 220KΩ (1/4W)	4	0,05	0,20
04.05	Ud.	BOBINA 1mH	1	0,30	0,30
04.06	Ud.	CONDESADOR 10nF	1	0,15	0,15
04.07	Ud.	CONDESADOR 100nF	3	0,15	0,15
04.08	Ud.	CONDESADOR 220nF	2	0,15	0,15
04.09	Ud.	CONDESADOR 330nF	1	0,15	0,15
04.10	Ud.	CONDESADOR 10μF	1	0,50	0,50
04.11	Ud.	CONDESADOR 330μF	1	0,50	0,50
04.12	Ud.	CONDESADOR 820μF	1	0,50	0,50
04.13	Ud.	DIODO MUR120	10	0,15	1,50
04.14	Ud.	DIODO RHRP860	6	1,00	6,00
04.15	Ud.	TRANSISTOR MOSFET IRF540	3	1,69	5,07
04.16	Ud.	TRANSISTOR MOSFET IRF540	3	4,99	14,97
04.17	Ud.	REGULADOR 12V LM2575	1	2,35	2,35
04.18	Ud.	REGULADOR 5V AP78L05	1	0,70	0,70
04.19	Ud.	DRIVER MOSFET HIP4081	2	9,90	19,80
04.20	Ud.	FUSIBLE RÁPIDO CRISTAL 6X32	1	0,35	0,35
		TOTAL SECCIÓN 4			44,60
		TOTAL SECCIÓN 4 + 18% IVA			54,39



5	SECCIÓN 05.- PRESUPUESTO TOTAL	Precio total €
	TOTAL sección 01.- Personal.	5.681,81
	TOTAL sección 02.- Hardware.	5.345,00
	TOTAL sección 03.- Software.	9.647,00
	TOTAL sección 04.- Componentes Eléctricos.	54,39
	TOTAL sección 05.- Presupuesto Total.	16.997,12
	TOTAL sección 05.- Presupuesto + 18% IVA	20.728,20

7. Conclusiones, trabajos futuros y recomendaciones.

Se ha diseñado y ha construido un controlador para motores DC brushless, obteniéndose un sistema de estructura compacta y robusta gracias a dos aspectos. Como *software*, al lenguaje G de programación utilizado en el entorno de programación gráfica LabVIEW, el cual ofrece una integración incomparable con miles de dispositivos de *hardware* y brinda cientos de bibliotecas integradas para análisis avanzado y visualización de datos, para crear instrumentación virtual. Y como *hardware*, al controlador de automatización programable CompactRIO, el cual ofrece un sistema reconfigurable de control y adquisición de bajo coste diseñado para aplicaciones que requieren alto rendimiento y fiabilidad. Una vez construido el sistema que además incluye una placa PCB con el *driver* de potencia, éste fue sometido a pruebas prácticas con resultados exitosos.

Al programa de control de velocidad del motor se ha incorporado variables indicadores, las cuales están distribuidas tanto en el código *main.vi* como en el código *FPGA PWM PID control.vi*. Estas variables permiten visualizar en tiempo real el valor de las señales importantes a través de la interfaz visual, de tal forma que ha permitido obtener un mayor control del sistema y corregir los errores que han ido surgiendo durante el desarrollo del proyecto.

Los resultados obtenidos en las pruebas realizadas han sido satisfactorios en todos los sentidos, pudiéndose comprobar prácticamente las buenas propiedades de la tecnología CompactRIO y LabVIEW, de tal forma que se puede controlar todos los aspectos del entorno desde un único dispositivo, sin tener que añadir electrónica física adicional. Este efecto se nota en la tecnología embebida usada que permite trabajar de una manera robusta y rápida, como se puede observar en el capítulo 5.

Este completar este proyecto se ha desembolsado una gran cantidad de dinero. A simple vista parece que el controlador CompactRIO junto con el entorno de programación gráfica LabVIEW es muy caro y que no se ha exprimido todo su potencial. Esto es debido a que este proyecto es el primer paso de un proyecto mucho mas grande. Este proyecto en cuestión se titula: técnicas de sensorización y control para la detección y localización eficiente de minas antipersonas. En dicho proyecto un robots hexápodo rastrea a través de un detector de minas como actuador, las posibles minas antipersona que puedan encontrarse delante del robot en cuestión. El objetivo que se pretende es mejorar el sistema central, cambiandolo por el controlador compactrio, de tal forma que no solo controle la funcionalidad de los motores de robot, sino también el brazo actuador.

Para mejorar el controlador creado, se puede cambiar varios aspectos del programa de control de velocidad:

- Para empezar, se puede modificar el regulador PID por algunos sistemas de control modernos, ya sea control fuzzy, redes neuronales, etc. Este cambio puede mejorar la toma de decisiones haciendo más preciso y eficiente al controlador.
- También se pueden incorporar nuevos y sofisticados métodos de seguridad que mejoren el funcionamiento, evitando que se produzcan errores fatales.
- Otro aspecto puede ser cambiar y observar si el método de conducción a 180° de los transistores MOSFET es más eficiente que el método elegido de conducción a 120°.
- De igual forma, se puede modificar la programación y por tanto para de la placa PCB, para que solo se tenga que programar una única señal PWM y desfasarla a través de la electrónica física, en vez de las seis señales diseñadas. Este cambio podría compactar y simplificar el programa y reducir costes al tener que usar menos salidas desde el módulo NI 9401.
- Además, para mejorar el cálculo de velocidad del motor se podría cambiar los tres sensores Hall del motor por un *encoder*. Con el *encoder* se gana en resolución y por tanto la precisión. Con un valor más exacto de la velocidad se conseguirá un mejor control.

Capítulo 8. Bibliografía.

Libros.

- [1] Richar C. Dorf, Robert H. Bishop: ‘Sistemas de Control Moderno’, (Pearson Education, 2005, 10th edn.), pp. 4-723
- [2] Juan Pedro Romera, Juan Antonio Lorite: ‘Automatización’, (Thomson Paraninfo, 1994, 4th edn.), pp. 1-20
- [3] Muhammad H. Rashid: ‘Electrónica de Potencia. Circuitos, dispositivos y aplicaciones’, (Pearson Education, 2004, 3rd edn.), pp 273
- [4] Daniel W. Hart, C: ‘Electrónica de Potencia’ (Pearson Education, 1997, 1st edn.), cap. 4
- [5] Jesús Fraile Mora: ‘Máquinas Eléctricas’, (Mc Graw Hill, 2008, 6th edn.), pp. 590
- [6] Theodore Wildi: ‘Máquinas Eléctricas y sistemas de potencia’, (Pearson Education, 1994, 6th edn.), pp. 579
- [7] Katsuhiko Ogata: Ingeniería de Control Moderna. (Pearson Education, 2010, 5th edn) pp568
- [8] Benkamin C. Kuo: Sistemas de Control Automático. (Prentice Hall, 1996, 7th edn) pp568
- [9] Manual de curso y ejercicios: LabVIEW Core 1. (Publicado por National Instruments, vers. 2009)
- [10] Manual de curso y ejercicios: LabVIEW Core 2. (Publicado por National Instruments, vers. 2009)
- [11] Manual de curso y ejercicios: LabVIEW FPGA. (Publicado por National Instruments, vers. 2009)

Revistas.

- [12] Kike Benito: “La evolución de los motores eléctricos”, Energías renovables, n. 63 (dic. 2007 / ene. 2008), p. 92-96

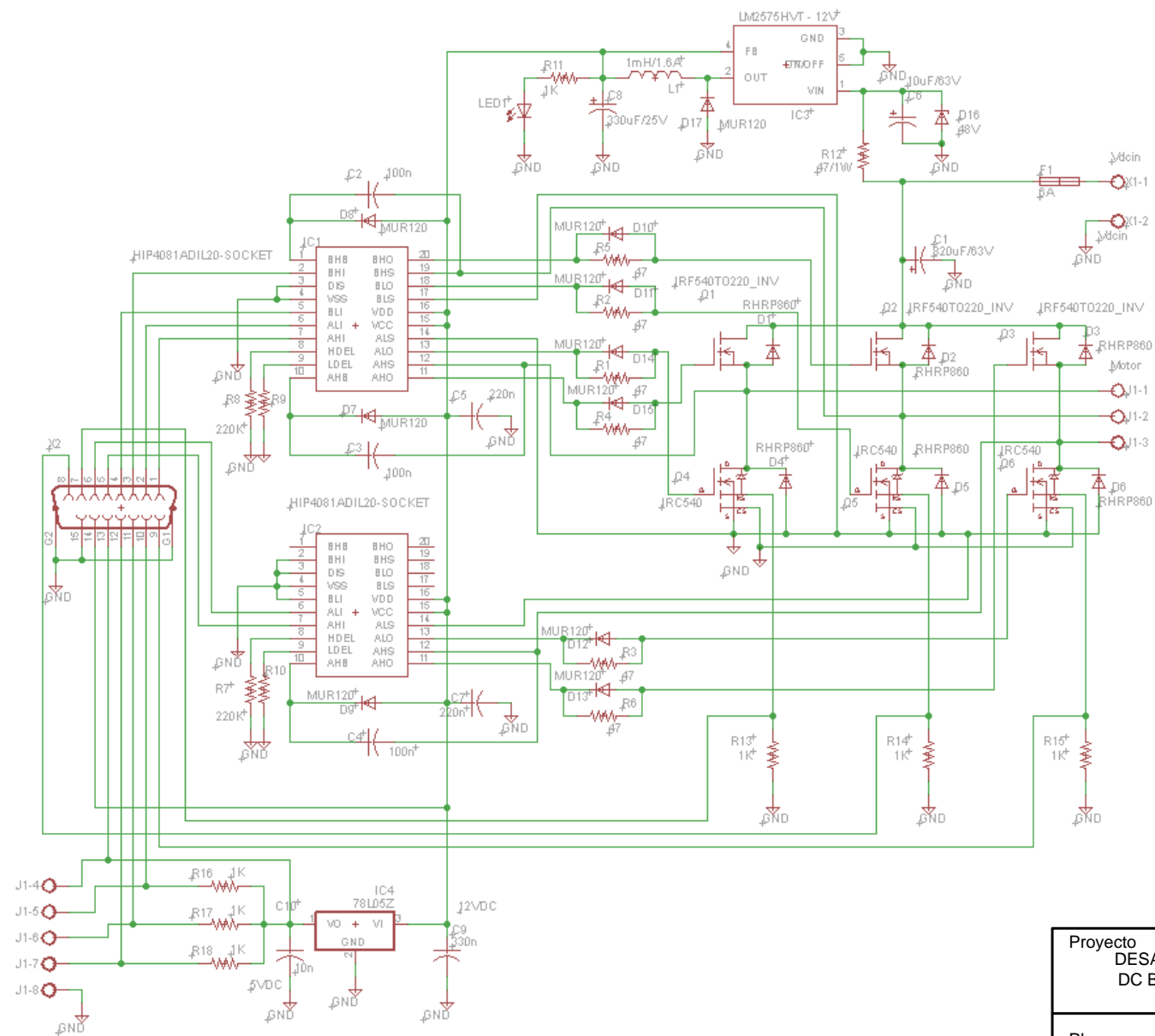
Proyecto fin de carrera.

- [13] Oñoro Hernando, Enrique: ‘Diseño de las etapas de potencia y control de un inversor trifásico’. Proyecto fin de carrera, Universidad Carlos III de Madrid, 2006. L/PFC 02981

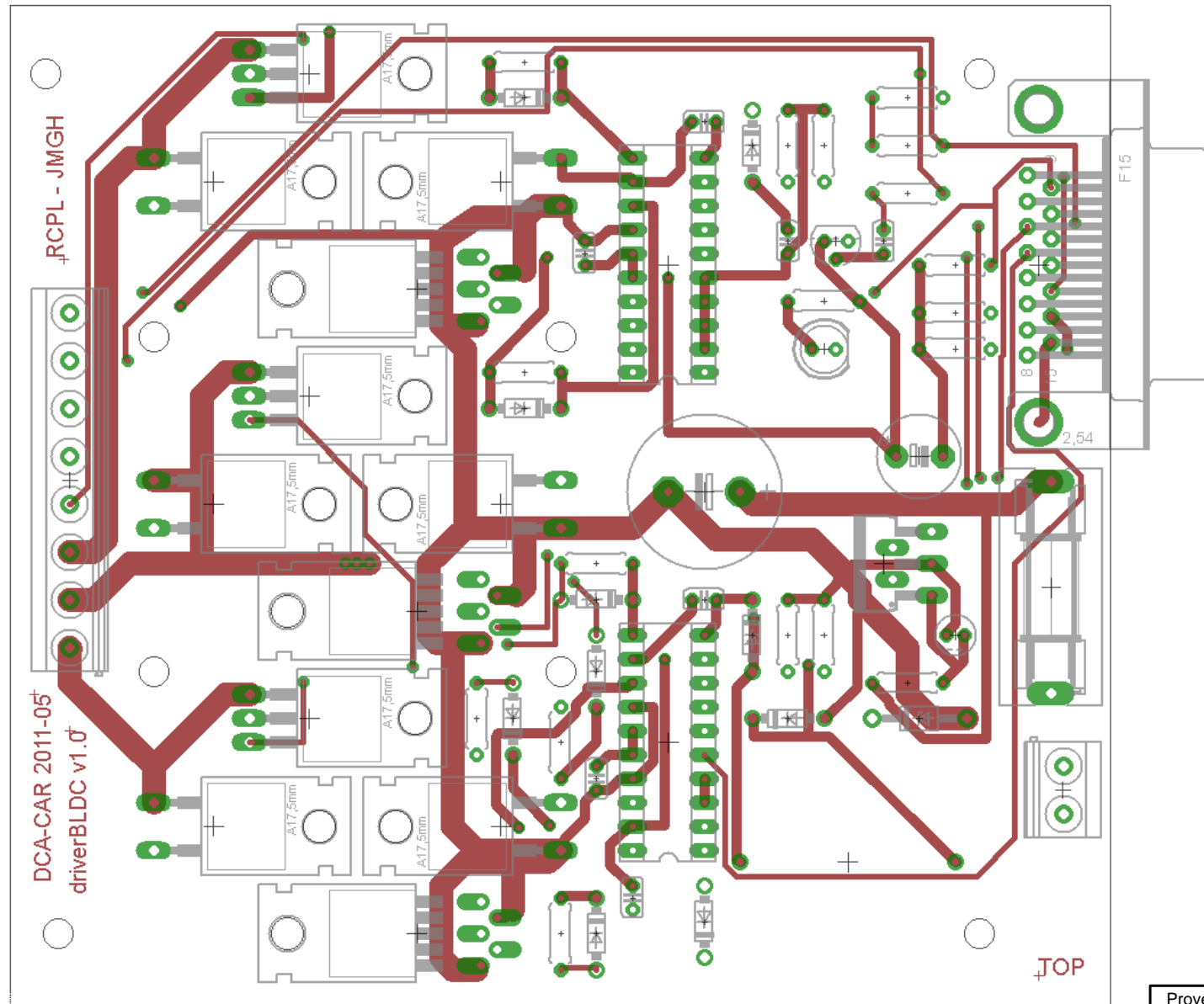
Páginas o documentos electrónicos en la red

- [14] <http://www.brushless-rc.com.ar/Caracteristicas.html>
- [15] <http://www.ingenia-cat.com/reference/learn/TEC.PAP.0422232450.pdf>
- [16] <http://es.scribd.com/doc/30757679/Control-Par-Pwm-Trifasico-2>
- [17] http://www.uma.es/investigadores/grupos/electronicapotencia/index.php?option=com_content&view=article&id=81&Itemid=100
- [18] http://www.uma.es/investigadores/grupos/electronica_potencia/index.php?option=com_content&view=article&id=82&Itemid=101
- [19] <http://www.ni.com/products/esa/>
- [20] Hoja de características del motor Maxon – EC 32: <http://microsat.sm.bmstu.ru/e-library/Algorithms/Hardware/wheels/Maxon%20Motor%20-%20EC.pdf>
- [21] Hoja de características del HIP4081A: <http://www.intersil.com/data/fn/fn3659.pdf>
- [22] Hoja de características del regulador LM2575:
<http://www.datasheetcatalog.org/datasheet/nationalsemiconductor/DS011475.PDF>
- [23] Hoja de características del regulador AP78L05:
<http://pdf1.alldatasheet.com/datasheet-pdf/view/177596/DIODES/AP78L05.html>

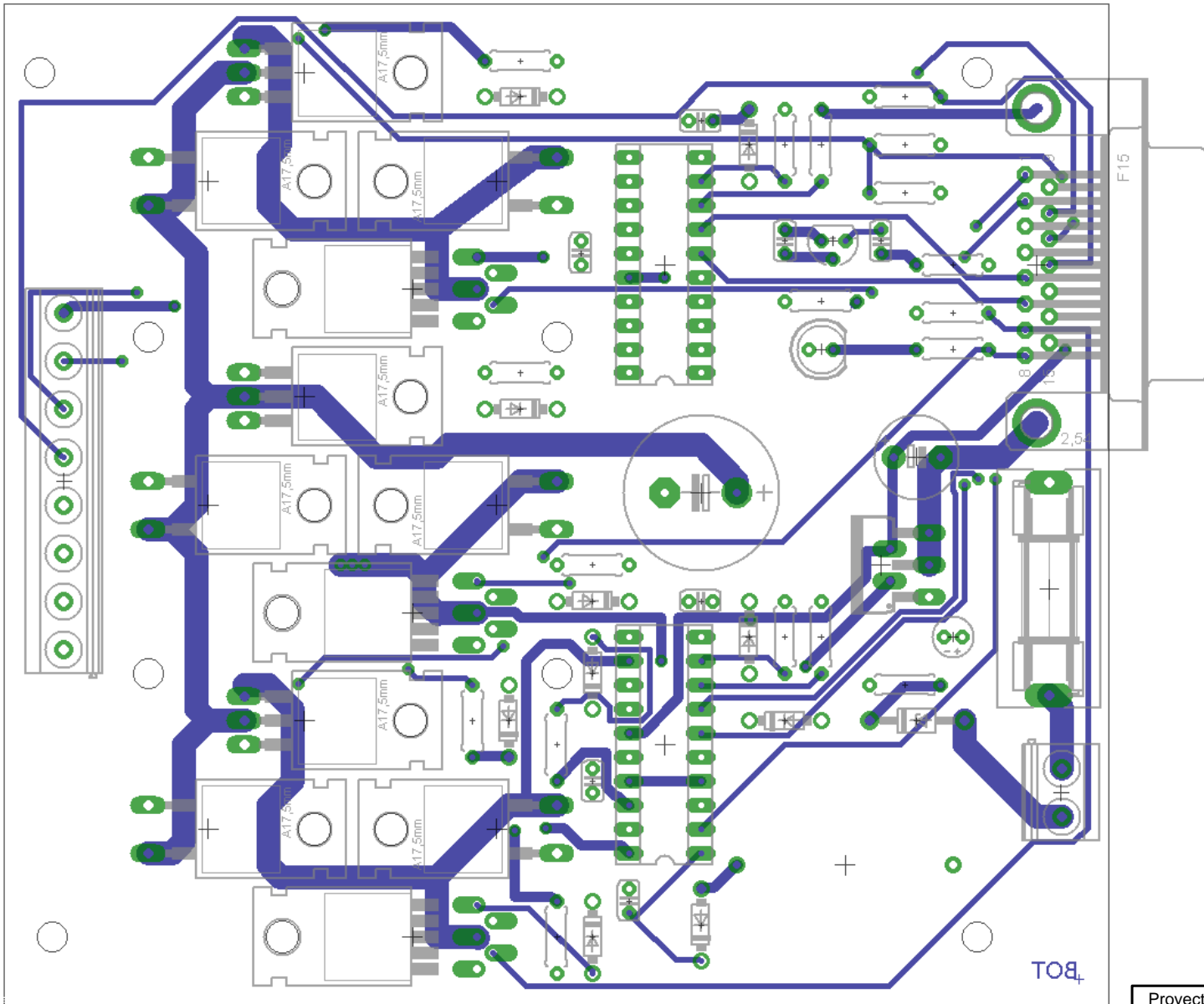
ANEXOS



Proyecto DESARROLLO DE UN CONTROLADOR PARA UN MOTOR DC BRUSHLESS BASADO EN COMPACTRIO Y LABVIEW DE NATIONAL INSTRUMENTS	
Plano ESQUEMÁTICO DE LA ETAPA DE POTENCIA DEL MOTOR DCBL	Código PLANO_Nº1
Autor JUAN MIGUEL GARCIA HARO	Fecha 1/9/11



Proyecto		
DESARROLLO DE UN CONTROLADOR PARA UN MOTOR DC BRUSHLESS BASADO EN COMPACTRIO Y LABVIEW DE NATIONAL INSTRUMENTS		
Plano	DISEÑO ELETRÓNICO DE LA ETAPA DE POTENCIA – CAPA TOP	Código PLANO_Nº2
Autor	JUAN MIGUEL GARCIA HARO	Fecha 1/9/11



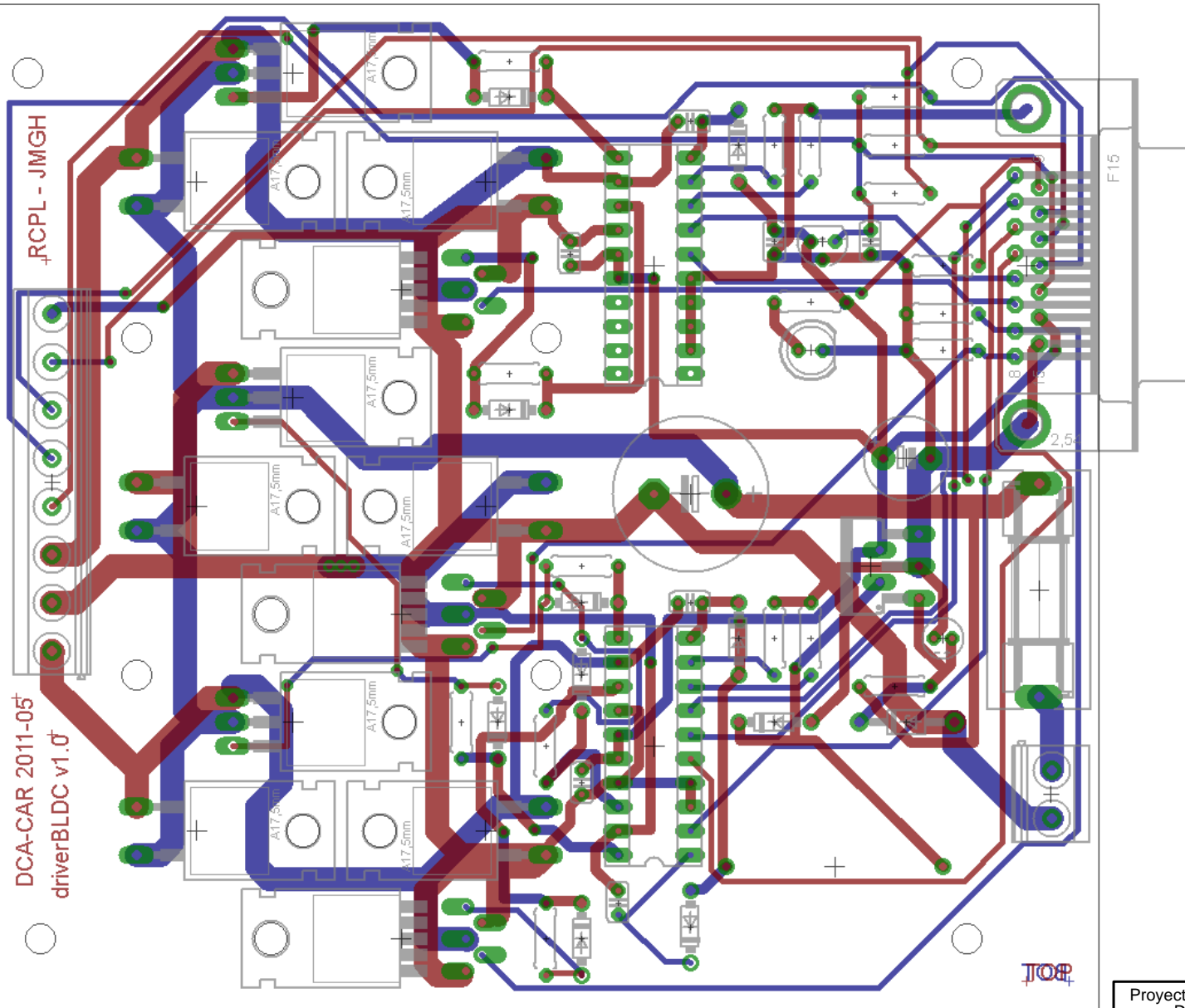
Proyecto
 DESARROLLO DE UN CONTROLADOR PARA UN MOTOR
 DC BRUSHLESS BASADO EN COMPACTRIO Y LABVIEW
 DE NATIONAL INSTRUMENTS

Plano
 DISEÑO ELETRÓNICO DE LA ETAPA
 DE POTENCIA – CAPA BOTTOM

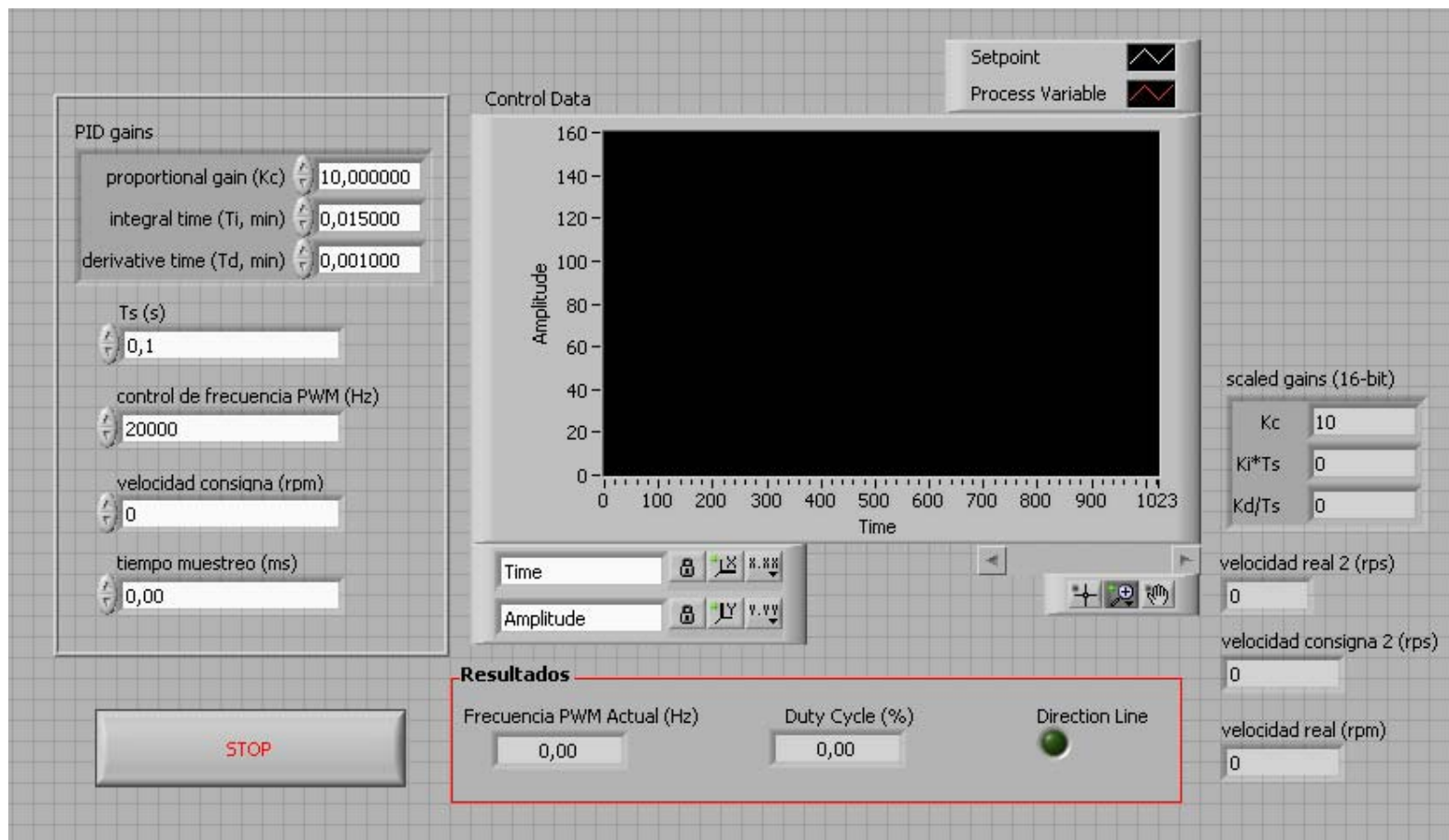
Código
 PLANO_Nº3

Autor
 JUAN MIGUEL GARCIA HARO

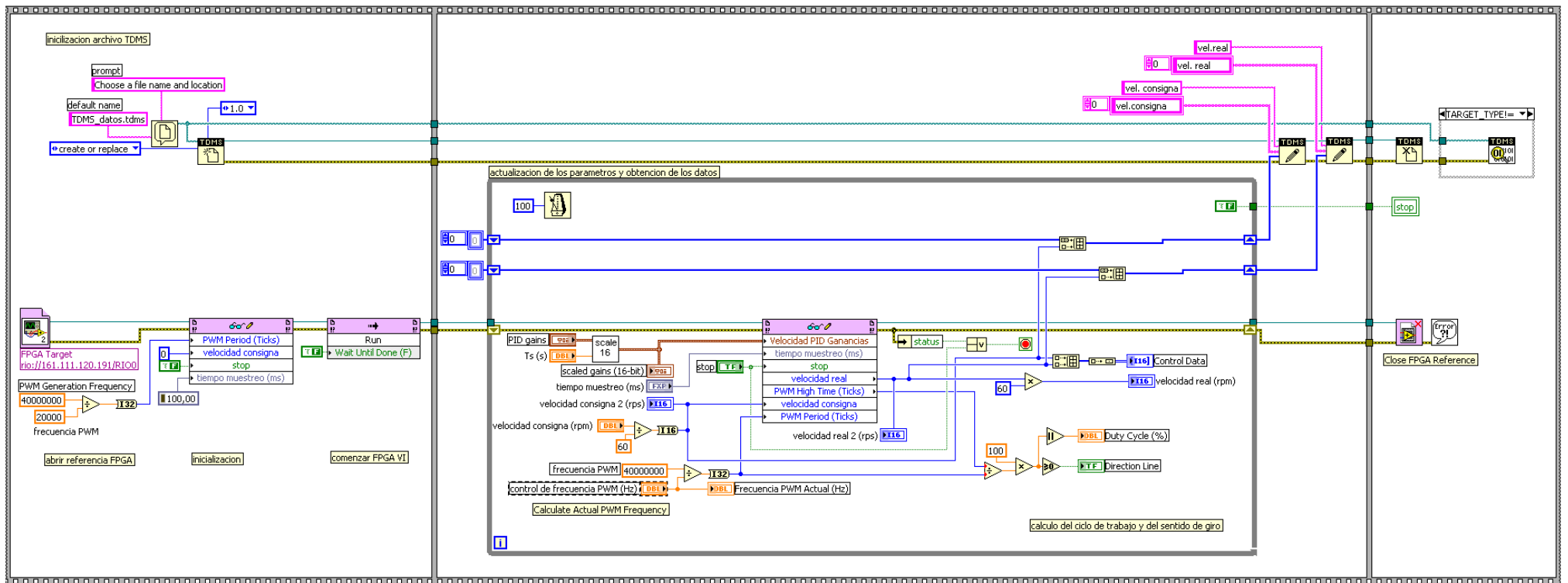
Fecha
 1/9/11



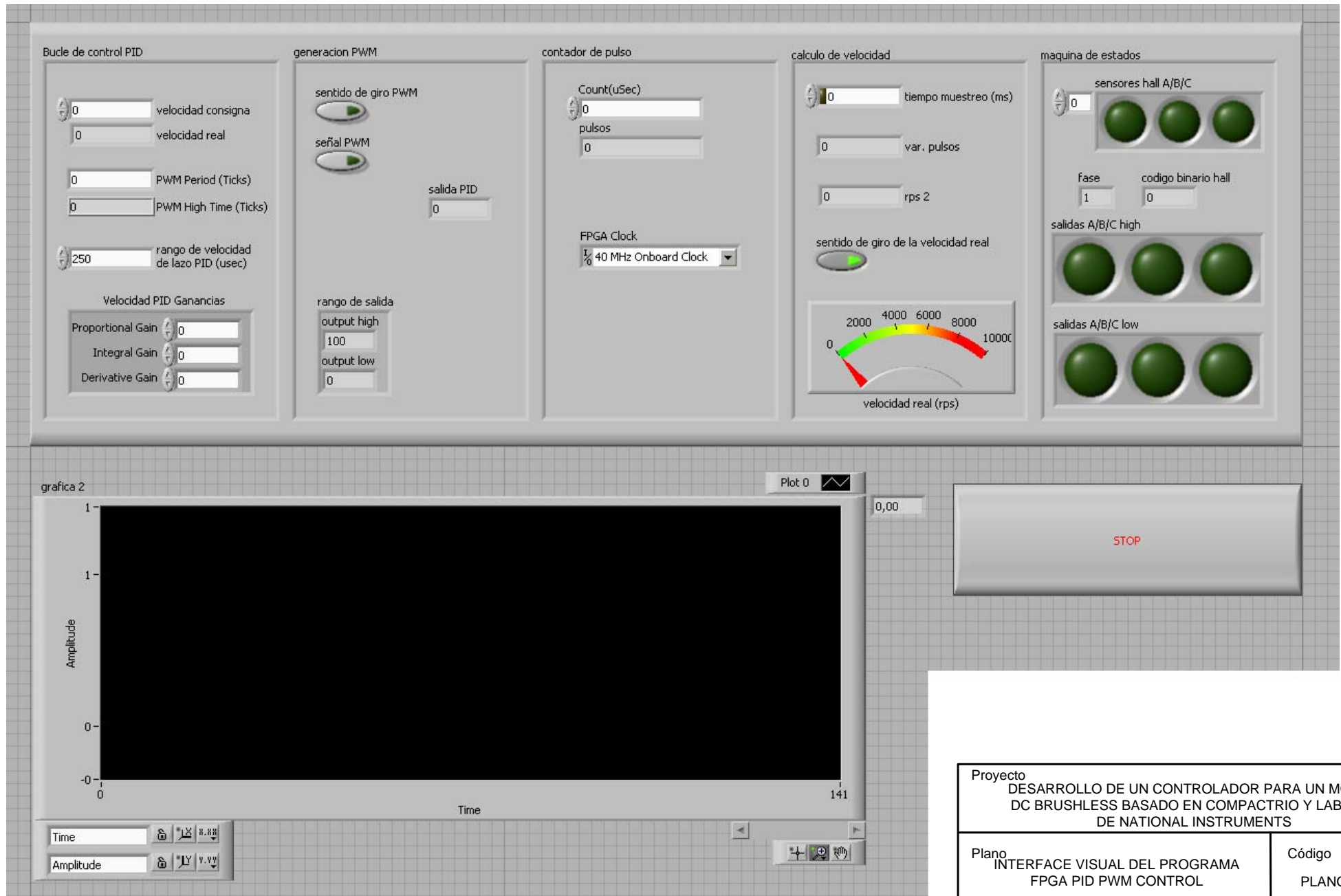
Proyecto DESARROLLO DE UN CONTROLADOR PARA UN MOTOR DC BRUSHLESS BASADO EN COMPACTRIO Y LABVIEW DE NATIONAL INSTRUMENTS	
Plano DISEÑO ELETRÓNICO DE LA ETAPA DE POTENCIA	Código PLANO_Nº4
Autor JUAN MIGUEL GARCIA HARO	Fecha 1/9/11



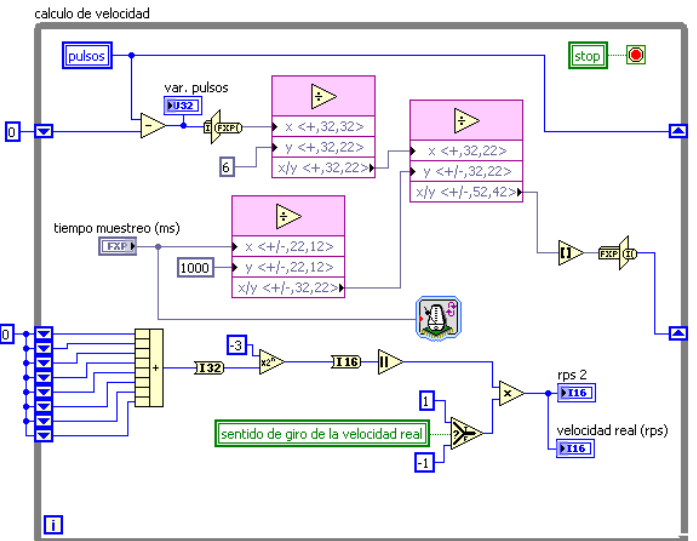
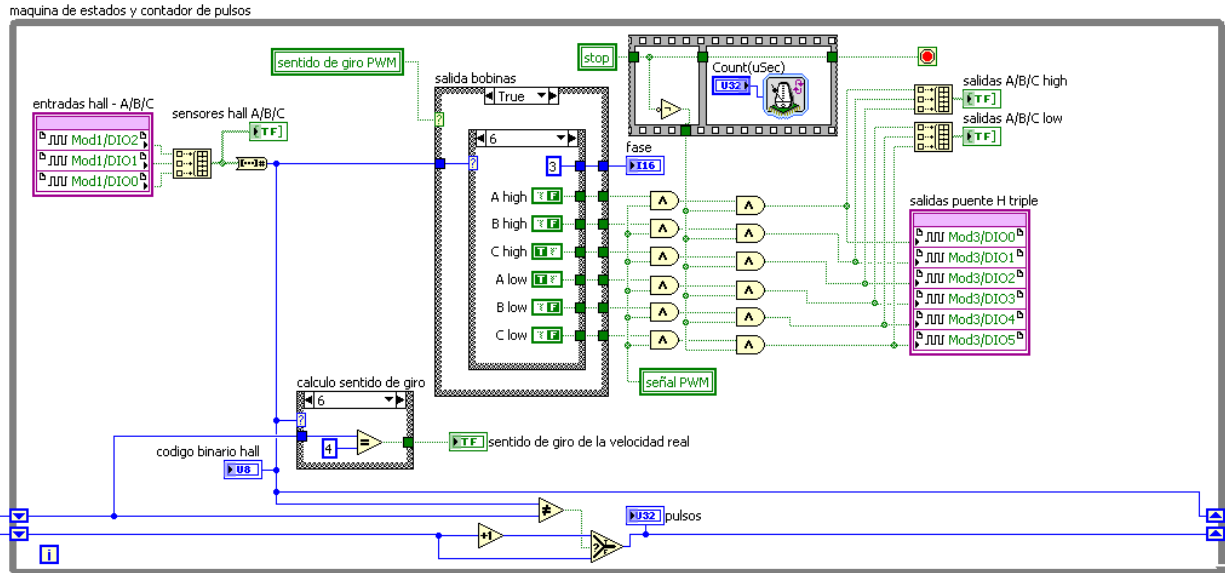
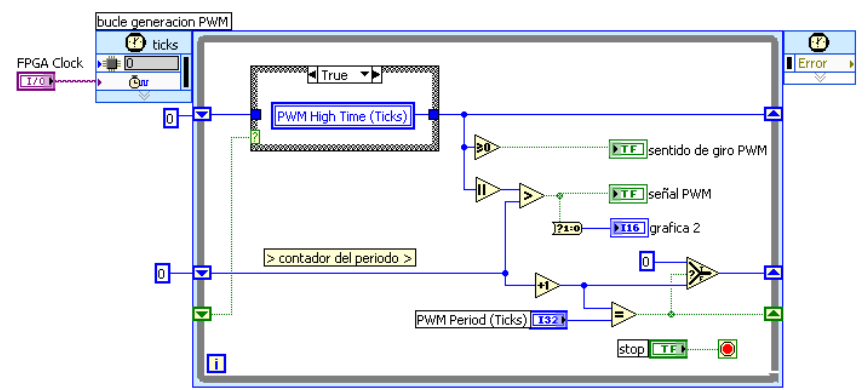
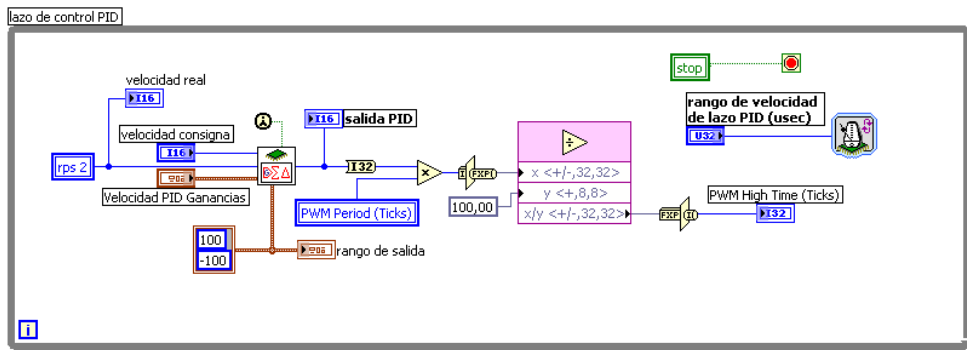
Proyecto DESARROLLO DE UN CONTROLADOR PARA UN MOTOR DC BRUSHLESS BASADO EN COMPACTRIO Y LABVIEW DE NATIONAL INSTRUMENTS		
Plano	INTERFACE VISUAL DEL PROGRAMA MAIN	Código PLANO_N°5
Autor	JUAN MIGUEL GARCIA HARO	Fecha 1/9/11



Proyecto DESARROLLO DE UN CONTROLADOR PARA UN MOTOR DC BRUSHLESS BASADO EN COMPACTRIO Y LABVIEW DE NATIONAL INSTRUMENTS		
Plano	DIAGRAMA DE BLOQUES DEL PROGRAMA MAIN	Código
		PLANO_N°6
Autor	JUAN MIGUEL GARCIA HARO	
	Fecha	1/9/11



Proyecto DESARROLLO DE UN CONTROLADOR PARA UN MOTOR DC BRUSHLESS BASADO EN COMPACTRIO Y LABVIEW DE NATIONAL INSTRUMENTS	
Plano INTERFACE VISUAL DEL PROGRAMA FPGA PID PWM CONTROL	Código PLANO_Nº7
Autor JUAN MIGUEL GARCIA HARO	Fecha 1/9/11



Proyecto DESARROLLO DE UN CONTROLADOR PARA UN MOTOR DC BRUSHLESS BASADO EN COMPACTRIO Y LABVIEW DE NATIONAL INSTRUMENTS		
Plano DIAGRAMA DE BLOQUES DEL PROGRAMA FPGA PID PWM CONTROL	Código PLANO_Nº8	
Autor JUAN MIGUEL GARCIA HARO	Fecha 1/9/11	

setpoint

initialize

process variable

output

PID gains ($\times 2^8$)

proportional gain (K_c)

integral gain ($K_c \cdot T_s / T_i$)

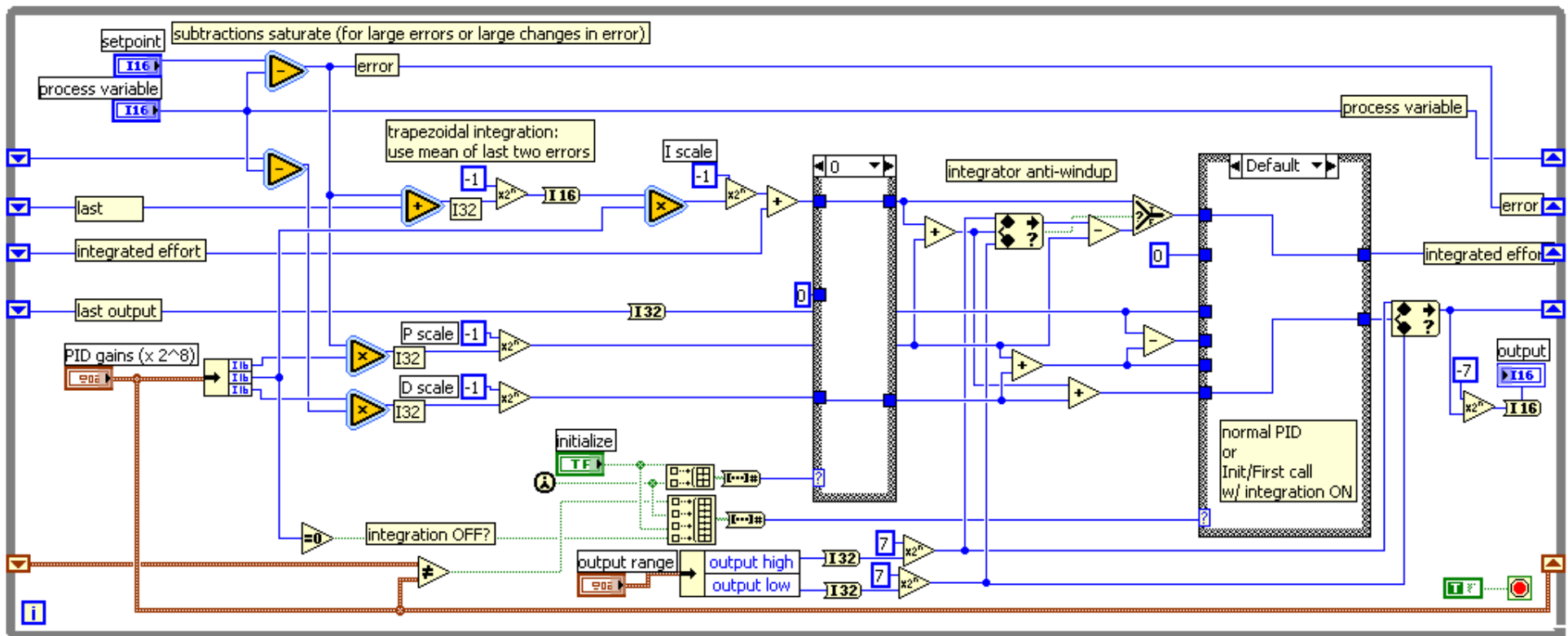
derivative gain ($K_c \cdot T_d / T_s$)

output range

output high

output low

Proyecto DESARROLLO DE UN CONTROLADOR PARA UN MOTOR DC BRUSHLESS BASADO EN COMPACTRIO Y LABVIEW DE NATIONAL INSTRUMENTS		
Plano	INTERFACE VISUAL DEL PROGRAMA DISCRETE PID	Código PLANO_Nº9
Autor	JUAN MIGUEL GARCIA HARO	Fecha 1/9/11



Proyecto DESARROLLO DE UN CONTROLADOR PARA UN MOTOR DC BRUSHLESS BASADO EN COMPACTRIO Y LABVIEW DE NATIONAL INSTRUMENTS		
Plano	DIAGRAMA DE BLOQUES DEL PROGRAMA DISCRETE PID	Código PLANO_Nº10
Autor	JUAN MIGUEL GARCIA HARO	Fecha 1/9/11