



UNIVERSIDAD DE CHILE
FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS
DEPARTAMENTO DE INGENIERÍA ELÉCTRICA

DISEÑO E IMPLEMENTACIÓN DE SISTEMA DE CONTROL PARA MICRO
HIDRÁULICA PLUG & PLAY

MEMORIA PARA OPTAR AL TÍTULO DE INGENIERO CIVIL ELECTRICISTA

PAULINA ANGÉLICA RAMÍREZ DEL BARRIO

SANTIAGO DE CHILE
JULIO 2012



UNIVERSIDAD DE CHILE
FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS
DEPARTAMENTO DE INGENIERÍA ELÉCTRICA

DISEÑO E IMPLEMENTACIÓN DE SISTEMA DE CONTROL PARA MICRO
HIDRÁULICA PLUG & PLAY

MEMORIA PARA OPTAR AL TÍTULO DE INGENIERO CIVIL ELECTRICISTA

PAULINA ANGÉLICA RAMÍREZ DEL BARRIO

PROFESOR GUÍA:
LORENZO REYES CHAMORRO

MIEMBROS DE LA COMISIÓN:
HÉCTOR AGUSTO ALEGRÍA
JUAN ANTEZANA NUÑEZ

SANTIAGO DE CHILE
JULIO 2012

RESUMEN DE LA MEMORIA
PARA OPTAR AL TÍTULO DE
INGENIERO CIVIL ELECTRICISTA
POR: PAULINA RAMÍREZ DEL BARRIO
FECHA: JULIO, 2012
PROF. GUÍA: Sr. LORENZO REYES CHAMORRO

“DISEÑO E IMPLEMENTACIÓN DE SISTEMA DE CONTROL
PARA MICRO HIDRÁULICA PLUG & PLAY ”

Nuestro país posee un gran potencial hidráulico para la generación de energía eléctrica constituido por gran variedad de caídas de agua y ríos de pequeños caudales (40 a 60 [l/s]). Sin embargo no existe en el mercado una solución tecnológica que permita su utilización independiente de las características del recurso que se tenga; modo de operación que se desee, sea conectado a la red eléctrica y/o en forma aislada; que sea de utilización simple y posea una interfaz amigable para el usuario final. En adición a lo anterior no se observa en la literatura internacional un esfuerzo por automatizar este tipo de medio de generación, lo que constituye un nicho de desarrollo tecnológico para el país.

A modo de mejorar las soluciones existentes de micro-generación (1 a 100 [kW]) y resolver el problema anterior, se origina en el Centro de Energía de la Universidad de Chile, la idea de diseñar una máquina inteligente que pueda operar en las condiciones antes mencionadas. Surge la Micro Central Hidráulica (MCH) “*Plug & Play*”.

El objetivo de este trabajo de título es diseñar e implementar un sistema de control electrónico automático para esta MCH inteligente de 10 [kW] de potencia, que permita explotar pequeños recursos hidráulicos mediante la aplicación de electrónica para su automatización.

Se toma un modelo genérico de MCH y se complementa para obtener una simulación de la planta y de los controladores de frecuencia y tensión en operación en vacío y con carga aislada. Se aprovechan los actuadores de los que se dispone y modifica su accionamiento manual a automático.

Se consigue implementar controladores de tipo PI, con cuidado de la sintonización, que muestran que una estrategia de este tipo es suficiente para controlar un generador sincrónico como el de la MCH. Se implementa el control de tensión mediante modificación de corriente de campo a través de un *chopper* y el control de frecuencia a través de resistencias de desahogo, modificando su potencia consumida.

Se obtiene como resultado final un prototipo de planta MCH, incluyendo la electrónica que necesita para su operación y la programación del sistema de control. Constituyéndose de esta forma la experiencia adquirida en *hardware* y *software* la mayor parte del aprendizaje.

A futuro se propone modificar el *hardware* por componentes superficiales a modo de reducir espacio utilizado. Modificar los actuadores por otros más eficientes. Y como recomendación final se propone reemplazar los actuadores por un equipo de electrónica de potencia que permita que la MCH genere según el recurso hidráulico y el equipo se encargue de adecuar la generación a los valores deseados de tensión y frecuencia.

A mis seres queridos,
con especial cariño a
Clara y Mercedes.

Agradecimientos

Agradecida de Dios principalmente, pues en su gracia es que todo ha sucedido. Gracias Dios.

Gracias a mi familia: A mi padre por su paciencia; a mi madre por sus cuidados; a la Gaby, la Caro y la Kote por sus ánimos y preocupación. A mi tía Clara por la calidez que me entregó durante la infancia, que junto a la influencia de mi madre, me ayudo a ver el mundo de forma optimista. A mi tía Mercedes por su guía y sabiduría durante la adolescencia, que junto a la influencia de mi padre, me ayudó a formar carácter.

A mis profesores: a tía Sarita por enseñarme que las matemáticas saben a frugales; a madre M^a Carmen por enseñarme a confiar en la bondad de las personas; a profe Marianne por enseñarme que los profesores son amigos.

Al amor, la amistad y al enamoramiento, que me permitieron conocer a mis mejores amigos, entre ellos a Carlos, quien me invitó a las experiencias de laboratorio durante sus estudios, donde descubrí mi vocación por la ingeniería eléctrica.

A Alejandro y a Miguel por confiar en mis capacidades para trabajar en el laboratorio de electrónica. A Juan por invitarme al proyecto MCH. A Pablo, Toño y Jorge por su compañía de lunes a sábados durante el desarrollo del proyecto. A Lorenzo y Héctor por su tiempo y disposición. A Paola, Bernardita y don Pedro, por su buena voluntad.

A la familia Dharmawidjaja Muñoz por soportarme innumerables días en su casa mientras escribía este documento. Especialmente a Jack por todos esos pequeños detalles que me ayudaron en gran medida durante este periodo.

Hay muchas mas personas a las que agradecer: Con, Feña, Fabián, Daniel, Gabriel, Matías... Si no los nombro a todos es porque tengo mala memoria, por lo que de todas formas les digo gracias para cuando lean estas líneas.

Índice General

Índice de Figuras	v
Índice de Tablas	xi
Acrónimos	xii
1. Introducción	1
1.1. Motivación	2
1.2. Objetivos	3
1.2.1. Objetivos específicos	3
1.3. Estructura del documento	4
2. Contextualización	5
2.1. Energías Renovables No Convencionales	5
2.2. Generación hidráulica	7
2.3. Proyecto Micro Central Hidráulica inteligente	10
3. Prototipo MCH	11
3.1. Diagnóstico componentes mecánicos y eléctricos	11
3.1.1. Válvula mariposa	11

3.1.2.	Turbina turgo e inyectores	12
3.1.3.	Generador sincrónico y electrónica disponible	13
3.2.	Infraestructura	14
3.3.	Interfaz de usuario	15
3.4.	Sistema electrónico	16
3.5.	Controlador Digital de Señales (DSC)	17
3.6.	Sensores	18
3.7.	Actuadores	20
4.	Sistema a controlar	21
4.1.	Mediciones y procesamiento	21
4.1.1.	Medición de tensión y corriente alterna	22
4.1.2.	Medición de frecuencia	24
4.1.3.	Medición de corriente de campo	25
4.2.	Modos de operación del sistema	25
4.2.1.	Modo aislado de la red	27
4.2.2.	Modo conectado a la red	27
4.3.	Diagramas de estado y control	28
4.3.1.	Control de tensión	31
4.3.2.	Control de frecuencia y desequilibrio	33
4.3.3.	Procedimiento de partida	37
4.3.4.	Procedimiento de sincronización	38
5.	Simulaciones e implementación de esquema de control para MCH	41

5.1. Modelación de la planta	41
5.1.1. Modelo y control de la turbina	42
5.1.2. Modelo y control del generador sincrónico	44
5.2. Simulaciones de bloques del sistema de control	49
5.2.1. Control de corriente de campo mediante circuito chopper	49
5.2.2. Control de frecuencia mediante resistencias de desahogo	55
5.2.3. Control de frecuencia mediante variación del recurso hídrico	58
5.2.4. PLL para sincronización	61
5.3. Resultados experimentales	62
5.3.1. Resultados MCH modo espera	63
5.3.2. Resultados MCH modo aislado	65
5.4. Discusión de Resultados	69
5.4.1. Control de tensión en modo de espera	69
5.4.2. Control de frecuencia en modo de espera	70
5.4.3. Control en operación en isla	70
6. Conclusiones	72
6.1. Trabajo futuro y recomendaciones	73
Bibliografía	75
A. Programación DSC	77
A.1. Programa principal	77
A.2. Configuración de ADC	91
A.3. Configuración de puertos digitales	94

A.4. Configuración puerto serial	96
B. Documentación electrónica	98
B.1. Cúpula	98
B.1.1. DSC	98
B.1.2. Buffer	100
B.1.3. Suministro	102
B.1.4. Filtros	104
B.1.5. Disparo resistencias de desahogo (circuito 1)	107
B.1.6. Fin de carrera de válvula y estado de contactores	109
B.2. Electrónica del tablero	111
B.2.1. Sensores corriente y tensión	111
B.2.2. Relés	112
B.2.3. Disparo resistencias de desahogo (circuito 2)	113
B.2.4. Chopper	115
B.2.5. Disparo puente rectificador	117
B.2.6. Puente rectificador	118

Índice de Figuras

2.1. Esquema de sistema hidroeléctrico a filo de agua. <i>BUN-CA.[1]</i>	8
2.2. Grupo Turbina-Generador. <i>[5]</i>	10
3.1. Válvula Mariposa Eléctrica modelo X05. <i>ICO-TEC (Empresa manufacturadora).</i>	12
3.2. Turbina tipo Turgo con inyectores. <i>Microhidráulica Inteligente.[8]</i>	12
3.3. Curva de potencia eléctrica en el plano H-Q. <i>[7]</i>	13
3.4. Sala de máquinas en fibra de vidrio. <i>[8]</i>	14
3.5. Cúpula con interfaz. <i>[8]</i>	15
3.6. Interfaz.	15
3.7. Single Board Computer.	16
3.8. Kit DSC. <i>[11]</i>	17
3.9. Filtro pasa bajos utilizado.	18
3.10. Esquemático de placa de filtros.	19
4.1. Procesamiento.	21
4.2. Obtención de valores efectivos mediante máximo.	22
4.3. Obtención de valores efectivos mediante transformada.	22
4.4. Obtención de frecuencia mediante cruces por cero.	24
4.5. Obtención de frecuencia mediante PLL.	24

4.6.	PLL para estimación de frecuencia.	24
4.7.	Esquema de modos de operación de MCH.	26
4.8.	Modo de operación aislado.	27
4.9.	Modo de operación conectado a la red.	27
4.10.	Esquema principal de la MCH.	29
4.11.	Esquema de consignas.	31
4.12.	Regulador de voltaje automático.	32
4.13.	Esquemático de <i>chopper</i> implementado.	32
4.14.	Diagrama de comportamiento de <i>chopper</i>	33
4.15.	Diagrama de control de <i>chopper</i>	33
4.16.	Mecanismo gobernador de velocidad.	34
4.17.	Actuador resistencias de desahogo.	35
4.18.	Diagrama de comportamiento de desahogo.	35
4.19.	Diagrama de control de resistencias de desahogo.	36
4.20.	Esquema compensación de carga.	36
4.21.	Compensación de carga.	37
4.22.	Sistema de accionamiento de la válvula por relés.	37
4.23.	Utilización de válvula.	38
4.24.	Sincronización llevando $\delta \epsilon$ entre dos puntos a cero	39
4.25.	Sincronización mediante transformadas.	40
5.1.	Esquema simplificado del sistema.	41
5.2.	Sistema hidráulico.	42
5.3.	Modelo ideal de la turbina. <i>Sección 9.1, Kundur[19]</i>	43

5.4. Representación ejes en una máquina síncrona. <i>Sección 4.2 [21]</i>	44
5.5. Circuito equivalente monofásico de generador síncrono.	47
5.6. Planta MCH.	47
5.7. Planta MCH.	48
5.8. Válvula mariposa.	48
5.9. Modelo con control de campo.	49
5.10. Alternativas de control.	49
5.11. Control P de tensión.	50
5.12. Control PI de tensión.	50
5.13. Control PID de tensión.	51
5.14. Control PI de tensión ante escalón de referencia.	51
5.15. Control PID de tensión ante escalón de referencia.	52
5.16. Control PI de tensión ante escalón de carga.	52
5.17. Control PI, variable manipulada $I[r]$ ante escalón de carga.	53
5.18. Control PID de tensión ante escalón de carga.	53
5.19. Control PID, variable manipulada $I[r]$ ante escalón de carga.	54
5.20. Corriente de excitación v/s ciclo de trabajo de PWM.	54
5.21. Corrientes de desahogo v/s ciclo de trabajo de PWM.	55
5.22. Variación de frecuencia v/s ciclo de trabajo de PWM.	55
5.23. Modelo para control de frecuencia.	56
5.24. Control de frecuencia en vacío.	56
5.25. Variable manipulada, ciclo [%] en vacío.	57
5.26. Control de frecuencia con carga.	57

5.27. Variable manipulada, ciclo [%] ante escalón de carga.	58
5.28. Modelo para control de frecuencia.	59
5.29. Control de frecuencia con carga.	59
5.30. Potencia hidráulica manipulada ante escalón de carga.	60
5.31. Control de tensión con carga.	60
5.32. Corriente de campo manipulada ante escalón de carga.	61
5.33. Modelo PLL para simulación.	61
5.34. Respuesta PLL.	62
5.35. Control de frecuencia en vacío.	63
5.36. Control de tensión en vacío.	63
5.37. Control de frecuencia en vacío ante variaciones del recurso hídrico.	64
5.38. Control de tensión en vacío ante variaciones del recurso hídrico.	64
5.39. Variacion del recurso hídrico.	65
5.40. Control de frecuencia con carga.	66
5.41. Control de tensión con carga.	66
5.42. Variaciones de carga trifásica.	67
5.43. Control de frecuencia con carga.	67
5.44. Control de tensión con carga.	68
5.45. Variaciones de carga trifásica.	68
B.1. Esquemático de placa para DSC	99
B.2. Diseño de placa para DSC	99
B.3. Placa para DSC	100
B.4. Esquemático de placa de buffers	101

B.5. Diseño de placa de buffers	101
B.6. Placa de buffers	102
B.7. Esquemático de placa de suministro	103
B.8. Diseño de placa de suministro	103
B.9. Placa de suministro	104
B.10. Esquemático de placa de filtros	105
B.11. Diseño de placa de filtros	105
B.12. Placa de filtros	106
B.13. Esquemático de placa de filtros	106
B.14. Diseño de placa de filtros	107
B.15. Placa de filtros	107
B.16. Esquemático de placa de disparo de resistencias de desahogo	108
B.17. Diseño de placa de disparo de resistencias de desahogo	108
B.18. Placa de disparo de resistencias de desahogo	109
B.19. Esquemático de placa de fin de carrera y estado de contactores	109
B.20. Diseño de placa de fin de carrera y estado de contactores	109
B.21. Placa de fin de carrera	110
B.22. Placa de estado de contactores	110
B.23. Esquemático de placa de sensores	111
B.24. Diseño de placa de sensores	111
B.25. Placa de sensores	112
B.26. Esquemático de placa de relés	112
B.27. Diseño de placa de relés	113

B.28.Placa de relés	113
B.29.Esquemático de placa de disparo de resistencias de desahogo	114
B.30.Diseño de placa de disparo de resistencias de desahogo	114
B.31.Placa de disparo de resistencias de desahogo	115
B.32.Pines de TRIAC BTA25-700B para conexión	115
B.33.Esquemático de placa de chopper	116
B.34.Diseño de placa de chopper	116
B.35.Placa de chopper	116
B.36.Esquemático de placa de disparo de puente rectificador	117
B.37.Diseño de placa de disparo de puente rectificador	117
B.38.Placa de disparo de puente rectificador	118
B.39.Esquemático de placa de puente rectificador	118
B.40.Diseño de placa de puente rectificador	119
B.41.Placa de puente rectificador	119

Índice de Tablas

2.1. Clasificación para centrales de filo de agua. <i>Fuente: BUN-CA.[1]</i>	8
3.1. Datos de placa del generador. <i>[8]</i>	13
3.2. Parámetros obtenidos del generador. <i>Lorenzo Reyes.[8]</i>	14
3.3. Adquisición de datos	18
3.4. Actuadores	20
4.1. Relación mediciones	25

Acrónimos

GH	<i>Generación Hidroeléctrica -</i>
PCH	<i>Pequeña central hidroeléctrica -</i>
GCH	<i>Gran central hidroeléctrica -</i>
ESHA	<i>European Small Hydropower Association -</i> Asociación Europea de Pequeñas Hidroeléctricas
UNIPEDDE	<i>International Union of Producers and Distributors of Electrical Energy -</i> Unión Internacional de Productores y Distribuidores de Electricidad
MCH	<i>Micro central hidroeléctrica -</i>
ER	<i>Energía Renovable -</i>
ERNC	<i>Energía Renovable No Convencional -</i>
PMGD	<i>Pequeño Medio de Generación Distribuida -</i>
PMG	<i>Pequeño Medio de Generación -</i>
MGNC	<i>Medio de Generación No Convencional -</i>

CE-FCFM	<i>Centro de Energía, Facultad de Ciencias Físicas y Matemáticas -</i>
AVR	<i>Automatic Voltage Regulator -</i> Regulador Automático de Voltaje
DSC	<i>Digital Signal Controller -</i> Controlador Digital de Señales
CCS	<i>Code Composer Studio -</i>
SBC	<i>Single Board Computer -</i>
ARM	<i>Advanced RISC Machine -</i>
RISC	<i>Reduced Instruction Set Computer -</i>
CAN	<i>Controller Area Network -</i>
I²C	<i>Inter-Integrated Circuit -</i>
RS232	<i>Recommended Standard 232 -</i>
PLL	<i>Phase-locked loop -</i> Lazo enganchado en fase

Capítulo 1

Introducción

La Generación Hidroeléctrica (GH) utiliza la fuente de energía renovable con mayor abundancia en el planeta: el agua [1]. Aprovecha la energía potencial del agua para transformarla en energía mecánica para accionar un generador y obtener de esta forma energía eléctrica.

La mayoría de las centrales generadoras son de gran escala por lo que requieren la construcción de represas y embalses para su funcionamiento, razón por la cual producen impactos ambientales y sociales como: la obstrucción de la corriente de ríos, la inundación de áreas considerables y la reubicación de comunidades, generando así controversia en torno a los proyectos de construcción de este tipo de centrales.

Los sistemas a pequeña escala no presentan los problemas antes mencionados y pueden contribuir a brindar electricidad a zonas aisladas de los sistemas de transmisión. Para su funcionamiento requieren una porción de caudal de un río o caída de agua.

Según la potencia instalada se puede distinguir entre pequeña (PCH) y gran central hidroeléctrica (GCH). Respecto a las PCH, no existe consenso en su clasificación. Algunos países como Portugal, España, Irlanda, Grecia y Bélgica, consideran como tal todas cuya potencia instalada no supere los 10 [MW]; en Italia, 3 [MW]; en Francia, 12 [MW]. La Comisión Europea, la ESHA y la UNIPEDA han decidido considerar como PCH aquellas que tengan una potencia instalada de 10[MW]. En esta última las PCH se subdividen en Nano (menos de 1[kW]), Micro (1 a 100 [kW]), Mini (100 a 1000 [kW]) y pequeñas como tal (1 a 10[MW])[1].

Como toda central generadora las micro centrales hidroeléctricas (MCH) necesitan de instalaciones civiles, hidráulicas, mecánicas y eléctricas para su funcionamiento. Y para asegurar una correcta operación de la misma, utilizan sistemas electrónicos, de comunicaciones y de control que les permita trabajar en condiciones aceptables para el sistema al cual se encuentren conectadas.

1.1. Motivación

En Chile se cuenta con un gran recurso hidráulico, dentro del cual las pequeñas caídas de agua y ríos pueden ser aprovechados para generación de energía eléctrica mediante la instalación de MCH. Un problema habitual en la instalación de este tipo de centrales corresponde al gran costo de ésta y a la diversidad de controladores no estándar disponibles en el mercado. Asimismo, hay que considerar que las soluciones para demanda aislada no están diseñadas para adaptarse y operar sincronizadamente con la red lo que genera un riesgo de quedar obsoletas con la eventual llegada de la red eléctrica en localidades donde hayan sido aplicadas.

En el mercado las MCH que existen constan básicamente de un conjunto turbina-generador y de mecanismos básicos de control. Estos productos se aplican en soluciones de electrificación rural, esto es, para funcionar aislados de la red eléctrica. Por otra parte, estas centrales requieren de una sala de máquinas para ser instaladas en su interior y no proporcionan una solución estandarizada para el resto de las otras obras involucradas tales como tubería de presión, desarenador, canal de derivación y bocatoma.

La mayoría de las MCH disponibles en el mercado son de fabricación china y están disponibles en Chile a través de distintos distribuidores. No existe una central a precios razonables que cuente con los sistemas de control necesarios para funcionar en forma totalmente automática en forma aislada y/o conectada a la red eléctrica, ni se identifica en la literatura internacional un esfuerzo por automatizar estos medios de generación, lo que se constituye en un nicho de desarrollo tecnológico para el país [2].

En el marco de un proyecto del Centro de Energía (CE-FCFM) de la Universidad de Chile, a modo de solucionar el problema anterior, surge la idea de diseñar una MCH “*Plug & Play*”, considerando los aspectos civiles y eléctricos de la implementación, para así mejorar las soluciones existentes de microgeneración resolviendo el abastecimiento de demanda aislada y demanda conectada a la red mediante la aplicación de electrónica para su control automático, aplicando además conceptos para una fácil instalación disminuyendo la complejidad que el usuario normalmente enfrenta para explotar el recurso hídrico. Refiriéndose por “*plug & play*” a una central inteligente que al ser conectada funcione reconociendo el mejor modo de operación disponible.

Es posible aplicar esta solución en lugares remotos que presentan buenos recursos hídricos y/o que se encuentren conectados a la red. De estar conectados a la red, dicha aplicación, requiere una operación sincronizada al sistema eléctrico y puede rentabilizarse adoptando los modelos de negocios considerados por la regulación vigente inherente a generación distribuida.

Una central de este tipo permite el aprovechamiento de economías de escala y la capacidad de implementar un proyecto de generación hidráulico sin tener que incurrir en costosos estudios y obras civiles propias de la GH. Contar con la electrónica que admita la flexibilidad y automatización de la operación, permitirá proveer a un segmento de la demanda actualmen-

te no cubierto. Adicionalmente, esta misma flexibilidad permitirá poder implementar micro redes inteligentes en que varias tecnologías podrán operar conjuntamente en forma óptima para poder entregar suministro eléctrico de mejor calidad y a mínimo costo.

1.2. Objetivos

El objetivo general que se persigue es diseñar e implementar un sistema de control electrónico para una MCH inteligente de 10 [kW] de potencia, que permita explotar pequeños recursos hidráulicos de forma rentable, de simple implementación y operación por parte del usuario. Que tenga una interfaz usuario amigable, que permita interacción adaptable para su instalación y operación en cualquier topología eléctrica. Se pretende diseñar e implementar la inteligencia necesaria para que la micro-central pueda operar en forma 100 % automática frente a necesidades de demanda eléctrica aislada (lugares alejados de la red eléctrica), como generador distribuido (en conexión a redes de distribución) y operar en micro redes de forma coordinada con otros medios de generación de diverso recurso energético.

1.2.1. Objetivos específicos

Particularmente es necesario desarrollar la electrónica y la estrategia de control que permita la utilización de este tipo de sistemas en medios aislados, híbridos e interconectados, de fácil implementación y que sea replicable.

En lo que concierne a la electrónica y al sistema de control, cabe mencionar que se busca reemplazar lo que actualmente posee la turbina, que se tiene disponible, con un desarrollo propio que tenga nuevas características y potencialidades que permitan operar la turbina de forma totalmente autónoma y segura.

En cuanto a arquitectura del sistema electrónico de control se está considerando dividirlo en una etapa de obtención de información analógica y una de toma de decisión digital. Esto para poder descongestionar el procesador digital y aprovechar una interfaz analógica mucho más rápida en el procesamiento continuo de las señales. Aún así las tomas de decisión se resuelven en el algoritmo de control anidado en el procesador digital.

El sistema de control central entregará consignas sobre:

- Control de corriente de campo o excitación.
- Control de las resistencias de desahogo.
- Control de válvula mariposa.
- Sistema de sincronización.

- Sistema de protección.
- Interfaz humana.

1.3. Estructura del documento

Este trabajo tiene la siguiente estructura:

En el capítulo 2, se realiza una contextualización del problema a resolver. Se revisan en forma general algunos conceptos de energías renovables y generación hidráulica. Y se presenta el proyecto Micro Central Hidráulica inteligente.

En el capítulo 3, se da a conocer en forma breve el prototipo de la micro central, los componentes mecánicos y eléctricos con lo que se comienza el desarrollo y los elementos que se han agregado para la obtención del prototipo final.

En el capítulo 4, se expone el sistema a controlar; cómo se realizan las mediciones y el procesamiento de las variables; los modos de operación de la central y sus diagramas de estado y control.

En el capítulo 5, se llevan a cabo las simulaciones y la implementación del sistema de control, presentando y discutiendo los resultados obtenidos.

Finalmente, en el capítulo 6, se entregarán las conclusiones del trabajo realizado.

Capítulo 2

Contextualización

El objetivo del presente capítulo es ubicar al lector en el entorno en el cual se desarrolla este trabajo de título.

2.1. Energías Renovables No Convencionales

Desde el comienzo de la Revolución Industrial se han utilizado combustibles fósiles como fuente energética del desarrollo económico del planeta. Estos recursos son de muy lenta velocidad de regeneración respecto de su explotación por lo que se clasifican como no renovables. Y su utilización genera impactos ambientales como el cambio climático, lluvias ácidas, adelgazamiento de la capa de ozono y otros. Por lo que la comunidad internacional se ha visto en la necesidad de buscar un nuevo modelo de desarrollo que no comprometa las necesidades de futuras generaciones. Donde los países desarrollados, en especial, la Unión Europea han sido los pioneros en la búsqueda de nuevas alternativas de suministro energético basados en fuentes renovables que aparecen como una opción limpia, segura y sobre todo inagotable.

Se considera como Energía Renovable (ER) a aquella cuya fuente primaria se renueva en una escala de tiempo comparable a la vida humana. El período de renovación puede variar de algunas horas a décadas. Como fuentes de este tipo se consideran: la hidráulica, la solar (térmica y fotovoltaica), la eólica y la de los océanos. Además, dependiendo de su forma de explotación, también pueden ser catalogadas como ER aquellas provenientes de la biomasa y de fuentes geotérmicas.

En Chile, de acuerdo a la Ley 20.257 [3] “Ley General de Servicios Eléctricos”, los medios de generación renovables no convencionales (ERNC) son los que presentan cualquiera de las siguientes características:

1. Aquellos cuya fuente de energía primaria sea la energía de la biomasa.
2. Aquellos cuya fuente de energía primaria sea la energía hidráulica y cuya potencia máxima sea inferior a 20.000 [kW].
3. Aquellos cuya fuente de energía primaria sea la energía geotérmica, entendiéndose por tal la que se obtiene del calor natural del interior de la tierra.
4. Aquellos cuya fuente de energía primaria sea la energía solar, obtenida de la radiación solar.
5. Aquellos cuya fuente de energía primaria sea la energía eólica, correspondiente a la energía cinética del viento.
6. Aquellos cuya fuente de energía primaria sea la energía de los mares, correspondiente a toda forma de energía mecánica producida por el movimiento de las mareas, de las olas y de las corrientes, así como la obtenida del gradiente térmico de los mares.

Los medios de ERNC, agrupan un conjunto de sub-clasificaciones a las que la Ley 19.940, Ley 20.257 y el reglamento del D.S. 244 han conferido derechos y obligaciones particulares [4]:

1. Pequeño Medio de Generación Distribuida (PMGD): Medios de generación cuyos excedentes de potencia sean menores o iguales a 9.000 [kW], conectados a instalaciones de una empresa concesionaria de distribución, o a instalaciones de una empresa que posea líneas de distribución de energía eléctrica que utilicen bienes nacionales de uso público. A los PMGD se les confiere el derecho a conectarse a las redes de distribución.
2. Pequeño Medio de Generación (PMG): Medios de generación cuyos excedentes de potencia suministrables al sistema sean menores o iguales a 9.000 [kW] conectados a instalaciones pertenecientes a un sistema troncal, de sub-transmisión o adicional.
3. Medio de Generación No Convencional (MGNC): Medios de generación cuya fuente sea no convencional y sus excedentes de potencia suministrada al sistema sean inferiores a 20.000 [kW]. La categoría de MGNC, no es excluyente con las categorías indicadas en los dos puntos precedentes. Esta categoría junto a los proyectos ERNC menores a 20 [MW] también incluye los proyectos de cogeneración eficiente a base de combustibles fósiles menores a 20 [MW].

2.2. Generación hidráulica

Este tipo de generación en base al recurso agua es la fuente renovable más utilizada a nivel mundial. El aprovechamiento de la energía del agua en movimiento se remonta a épocas antiguas, donde su principal uso era la molienda de granos, martillos para trabajos metalúrgicos o transporte de mercancías mediante barcazas, contribuyendo al desarrollo económico e industrial de muchos países desde la Edad Media hasta la Revolución Industrial.

El potencial hidráulico se compone de la energía potencial respecto de un nivel de referencia y de la energía cinética del caudal. De esta forma, se puede expresar como:

$$Potencia\ hidráulica = \gamma Q H_n \quad (2.1)$$

Donde γ es el peso específico del agua (1.000 [kg/m³]), Q el caudal en [m³/s] y H_n la altura neta en [m].

La cantidad de potencia y energía disponible en el agua de un río o una quebrada, está en relación directa a la altura o caída disponible, así como del caudal.

El aprovechamiento de la energía potencial del agua para producir electricidad constituye la base de la generación hidroeléctrica. El conjunto de instalaciones e infraestructura para aprovechar este potencial se denomina central hidroeléctrica.

Se pueden distinguir principalmente dos tipos de centrales hidroeléctricas: las que utilizan el agua según discurre normalmente por el cauce de un río y aquellas a las que ésta llega, convenientemente regulada, desde un lago o embalse [1].

Las centrales de agua embalsada regulan los caudales de salida para utilizarlos cuando sea necesario. Tienen la capacidad de generar cantidades considerables de electricidad en forma constante durante ciertos períodos, pero causan impactos ambientales y sociales como: la obstrucción de la corriente de ríos, la inundación de áreas considerables y la re-ubicación de comunidades. Este tipo de centrales son de tamaño mediano o grande en donde el caudal aprovechado por las turbinas es proporcionalmente muy grande al caudal promedio anual disponible en el río.

Las centrales a filo de agua son aquellas instalaciones que mediante una obra de toma, captan una parte del caudal del río y lo conducen hacia la central para su aprovechamiento y después lo devuelven al cauce del río. Esta disposición es característica de las centrales medianas y pequeñas, en las que se utiliza una parte del caudal disponible en el río para la producción de energía hidroeléctrica.

Dentro de las centrales de filo de agua (fig. 2.1) se puede apreciar una clasificación según a potencia generada:

Tamaño/Potencia	Aplicaciones
Nano o Pico-hidro: menos de 1[kW]	Uso familiar y aplicaciones mecánicas.
Micro-hidro: de 1 a 100[kW]	Red eléctrica comunal (sistema aislado).
Mini-hidro: de 100 a 1.000[kW]	Varias comunidades dentro de un radio de 10 a 40 [km], y/o conexión a la red nacional.
Pequeña central: de 1 a 5 [MW]	Pequeña ciudad y comunidades aledañas, además de conexión a la red.

Tabla 2.1: Clasificación para centrales de filo de agua. Fuente: BUN-CA.[1]

Las MCH están catalogadas como PMGD dentro de las ERNC por su menor nivel de implementación y potencia [3].

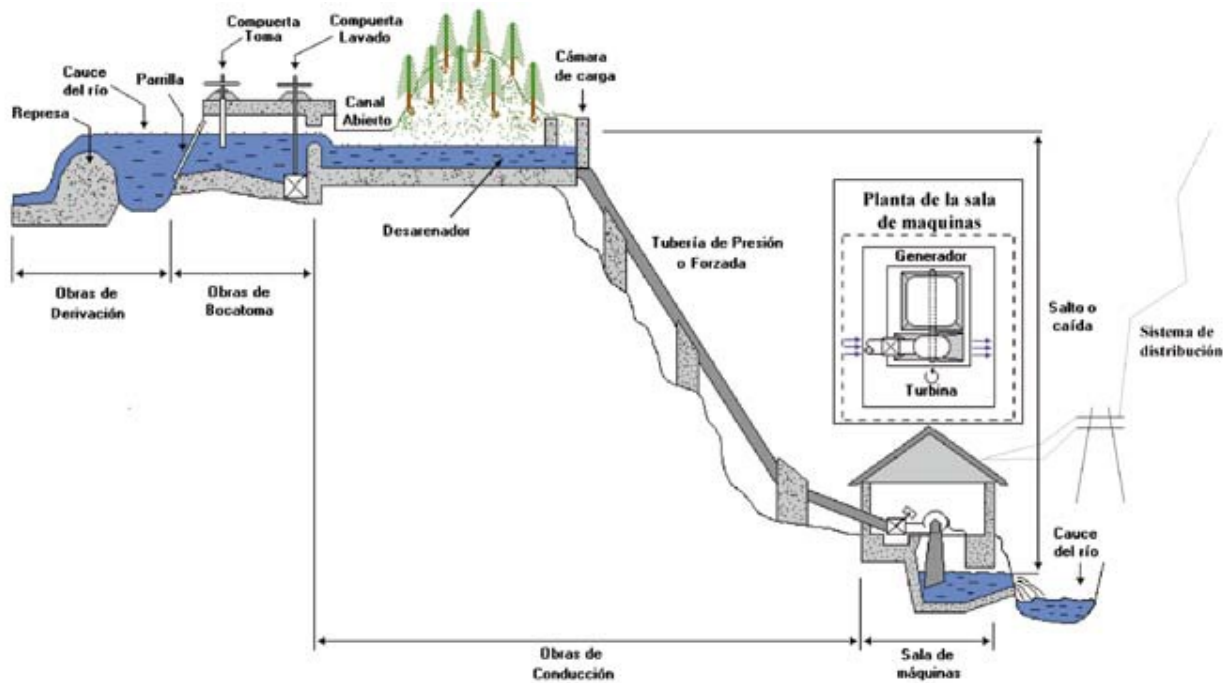


Figura 2.1: Esquema de sistema hidroeléctrico a filo de agua. BUN-CA.[1]

En la figura 2.1 se muestra el esquema de un sistema hidroeléctrico a filo de agua típico con todos sus elementos, los cuales se describen a continuación [1]:

- a) Obras de derivación: Tipo de pequeña represa transversal al cauce del río para producir un remanso que facilite la derivación del agua hacia la bocatoma.
- b) Obras de bocatoma: Para introducir y controlar el ingreso de agua al canal. Incluye una compuerta de toma del recurso hídrico y una compuerta de lavado, previo al ingreso del agua al desarenador.
- c) Obras de conducción: Consideran el desarenador para eliminar la arena y sedimentos de la corriente en el canal; el canal para conducir el agua desde la bocatoma hasta la entrada a la tubería de presión; la cámara de carga como punto de acumulación del agua antes de entrar a la tubería de y para sedimentar las impurezas del agua, retirar los elementos flotantes, controlar la entrada de agua a la planta y desviar el exceso; y la tubería de presión que conduce el agua hasta la turbina.
- d) Sala de máquinas: Considera la turbina encargada de transformar en energía mecánica la energía contenida en el agua; el generador encargado de convertir la energía mecánica recibida de la turbina a través de un eje, en energía eléctrica; el transformador o inversor para elevar el voltaje de la corriente generada.
- e) Líneas de transmisión: Para la conducción de la corriente eléctrica a los sitios donde se necesita la energía eléctrica.
- f) Líneas de distribución: Para repartir la electricidad hasta los puntos finales de utilización, pueden ser líneas aéreas o subterráneas.
- g) Aliviaderos: Puede ser necesario usar aliviaderos en la bocatoma, canal, cámara de carga y desfogue de la turbina para que los excesos de agua sean retirados del sistema y debidamente conducidos hacia un cauce estable.

Según el tipo de central se puede prescindir de alguno de los componentes antes mencionados. Por ejemplo para los proyectos MCH no se requiere necesariamente un transformador y en ocasiones se les instala un inversor.

2.3. Proyecto Micro Central Hidráulica inteligente

El proyecto, en el cual se desarrolla este trabajo de título, forma parte de los proyectos de redes inteligentes y generación distribuida del CE-FCFM ¹ de la Universidad de Chile.

Este proyecto cuenta con una micro-turbina hidráulica tipo Turgo asociada a un generador sincrónico de 10[kVA] de potencia nominal proveniente de China [5]. Dicho grupo turbina generador (fig. 2.2) puede operar en isla y posee un sistema semiautomático de sincronización a la red trifásica. Asimismo, viene equipada con una válvula de control de flujo de tipo mariposa operada por un servo motor.



Figura 2.2: Grupo Turbina-Generador. [5]

El objetivo general del proyecto es desarrollar una central microhidráulica comercial de 10 [kW] que pueda funcionar de forma autónoma o conectada a la red en base al grupo generador chino mencionado.

Se reemplazará el sistema electrónico actual y se desarrollará un sistema de control que permita cumplir con los objetivos del proyecto. En este contexto el memorista participa en la implementación de la electrónica del prototipo de la MCH y en el diseño del sistema de control.

¹www.centroenergia.cl

Capítulo 3

Prototipo MCH

A continuación se presenta en forma breve el desarrollo del prototipo de la MCH.

3.1. Diagnóstico componentes mecánicos y eléctricos

Se realizó un diagnóstico de las componentes mecánicas del grupo turbina-generator por parte de un ingeniero mecánico, Kim Hauser [7], en el marco de su trabajo de título. Y de forma similar se evaluaron las componentes eléctricas por parte de los ingenieros eléctricos Lorenzo Reyes y Pablo Jiménez [5], con participación de la alumna memorista del presente documento.

3.1.1. Válvula mariposa

La micro turbina hidráulica está equipada con dos válvulas mariposa (fig. 3.1). Normalmente este tipo de válvula se ocupa para abrir o cerrar el paso del agua hacia la turbina.

Una de las válvulas es operada por un servo motor controlable de manera semiautomática desde el panel de control, abriéndose o cerrándose, pero además tiene la posibilidad de manipulación mediante una llave allen en caso de no estar energizada o conectada al panel. El servomotor presenta la posibilidad de adquisición de información sobre los estados de fin de carrera (completamente abierta o completamente cerrada).



Figura 3.1: Válvula Mariposa Eléctrica modelo X05.ICO-TEC (Empresa manufacturadora).

3.1.2. Turbina turgo e inyectores

Una turbina turgo [6] puede trabajar en saltos con alturas comprendidas entre 50 y 300 [m]. Se trata de una turbina de acción, donde el chorro de agua incide con un ángulo de 20° respecto al plano diametral del rodete, entrando por un lado del disco y saliendo por el otro, simultáneamente sobre varios alabes. Su menor diámetro conduce, para igual velocidad periférica, a una mayor velocidad angular, lo que facilita su acoplamiento directo al generador.

La turbina en cuestión (fig. 3.2) posee dos inyectores fijos y uno de estos se puede inhabilitar mediante una válvula secundaria manual.



Figura 3.2: Turbina tipo Turgo con inyectores. *Microhidráulica Inteligente.*[8]

Se realizaron pruebas para determinar el comportamiento del grupo turbina generador representando curvas de operación en el plano H-Q (altura de la columna de agua vs. caudal). Obteniendo que el equipo del cual se dispone, utilizando un inyector, es para una caída de 50 [m.c.a] y un caudal de 32 a 38 [l/s]. Y utilizando dos inyectores, para 34 [m.c.a] y un caudal de 45 a 50 [l/s] [7].

Se utilizaran los dos inyectores para operar en la curva demarcada con un triángulo en la figura 3.3.

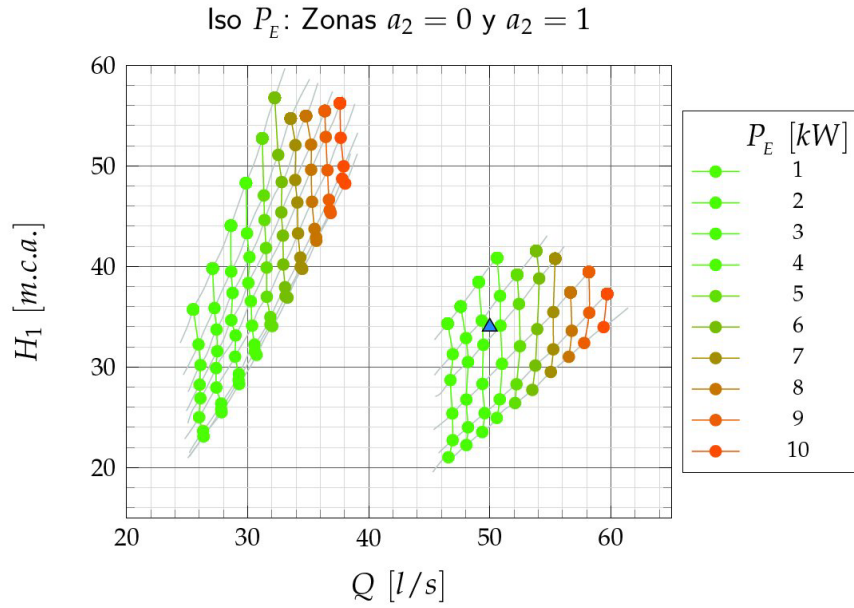


Figura 3.3: Curva de potencia eléctrica en el plano H-Q. [7]

3.1.3. Generador sincrónico y electrónica disponible

En [5] se presentan los dispositivos con los que cuenta la MCH. Describe los esquemas de conexión eléctricos del grupo turbina-generator; los modos de operación conectado a la red, isla y apagado; la estructura del tablero; los modos semiautomáticos de sincronización y control de válvula; y el regulador de voltaje automático para control de corriente de campo. En base al documento se puede apreciar la necesidad de elaborar un nuevo tablero más compacto y sistemas electrónicos de control automáticos de reemplazo para los existentes.

En [8] se exhiben los datos de placa del generador y los parámetros obtenidos mediante las pruebas de cortocircuito y saturación [9].

Datos de placa	
Conexión	Y
n_{nom} [rpm]	1500
V_{nom} [Vff]	380
I_{nom} [A]	15,19
P_{nom} [kVA]	10
f_{nom} [Hz]	50
Z_{base} [Ω]	15,19

Tabla 3.1: Datos de placa del generador. [8]

Parámetros	$[\Omega]$	p.u.
Z_s	13,37	0,88
$Z_{s_{sat}}$	11,61	0,76
R_s	0,15	0,01
R_{s_m}	0,80	0,05
X_s	13,37	0,88
$X_{s_{sat}}$	11,61	0,76
R_{f_m}	7,20	0,47

Tabla 3.2: Parámetros obtenidos del generador. *Lorenzo Reyes.*[8]

3.2. Infraestructura

Se enunciaron anteriormente los elementos necesarios en un sistema hidroeléctrico a filo de agua como lo es el del proyecto MCH. Fue labor del diseñador industrial José Antonio Marín el desarrollar una solución tecnológica para la infraestructura de la micro central. Realizó los diseños de bocatoma, desarenador, cámara de carga, sala de máquinas y otros. Para obtener mayor información respecto a este apartado dirigirse a [8].

Para albergar los componentes electromecánicos y el tablero eléctrico, se construye, a modo de sala de máquinas, una estructura de fibra de vidrio (fig. 3.4) similar al caparazón de un caracol, compacta y móvil. Tomando en cuenta variables de protección, estructurales, estabilidad, interfaz hombre-máquina, instalación y de relación con el medio ambiente. Para contener el módulo turbina-generador, el tablero de potencia, de control y la interfaz.



Figura 3.4: Sala de máquinas en fibra de vidrio. [8]

Es en la parte superior del caparazón (fig. 3.5), donde va la interfaz para el usuario y a su interior se tiene el espacio disponible para el desarrollo de los circuitos electrónicos para el control de la MCH, por lo que se deben tener en consideración los límites espaciales de la cúpula y el tablero al momento de diseñar las placas.



Figura 3.5: Cúpula con interfaz. [8]

3.3. Interfaz de usuario

Es necesaria una interfaz para que el usuario pueda encender la central, seleccionar modo de operación y monitorear el estado de diversas variables de la central.

Esta interfaz (fig. 3.6) fue programada por Jorge Dharmawidjaja y se encuentra ubicada en la cúpula de la central.

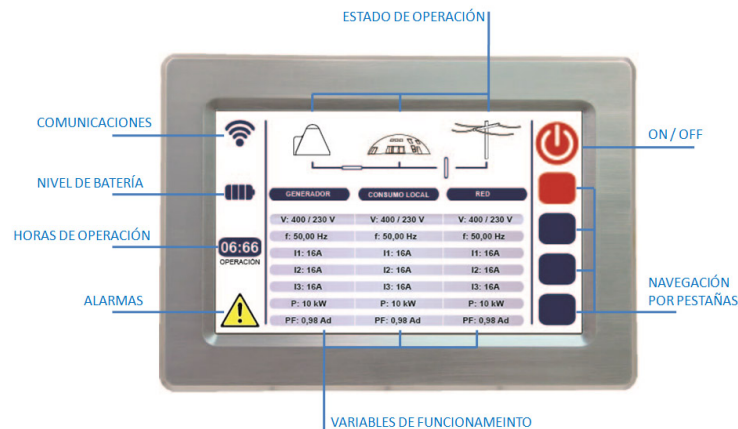


Figura 3.6: Interfaz.

La interfaz es accesible mediante un Single Board Computer (SBC) (fig. 3.7), que es un mini computador embebido con sistema operativo Linux Debian, con un procesador ARM ², con pantalla táctil, 3 entradas de comunicación serial RS232 y 2 puertos USB que permiten la consideración futura de utilización de comunicación inalámbrica con la central. Este SBC fue configurado por el ingeniero civil electricista Pablo Weber.



Figura 3.7: Single Board Computer.

3.4. Sistema electrónico

El sistema electrónico de la MCH se piensa para: la adquisición de datos de mediciones de la central; adaptación de las señales para el control; procesamiento de la información para toma de decisiones acorde a la estrategia de control escogida; comunicación entre controlador e interfaz y viceversa; ejecución de ordenes mediante actuadores para llevar a la central a la operación deseada.

La electrónica se puede dividir en:

- Electrónica de la cúpula: compuesta por el controlador DSC; la placa de suministro de energía del sistema; los filtros para adaptación análoga de las mediciones y el circuito de control del desahogo.
- Electrónica del tablero: compuesta por las placas de sensores; placa de reles para control de la válvula, inversor y contactores; placa de accionamiento de desahogo y *chopper* para control de la corriente de campo.

En el anexo B: “Documentación electrónica” se presenta el desarrollo de los diseños de los circuitos utilizados en el proyecto MCH.

²Advanced RISC (Reduced Instruction Set Computer) Machines

3.5. Controlador Digital de Señales (DSC)

Para poder llevar a cabo el control de la central se decide utilizar un DSC de Texas Instruments, TMS320F28335 [10] (fig. 3.8), que cuenta con:

- 16 canales análogos-digitales de 12 bit.
- 18 canales de PWM
- 88 canales digitales de propósito general de entrada o salida (GPIO).

Lo que le confiere una característica versátil al momento de pensar el control y diseñar los sistemas electrónicos.

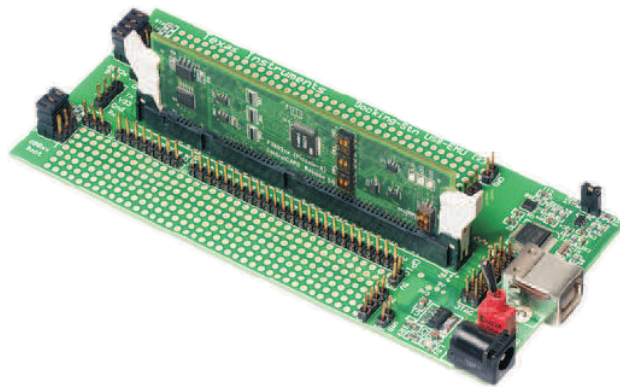


Figura 3.8: Kit DSC. [11]

Para programar el controlador se utiliza Code Composer Studio (CCS) [12], también de Texas Instruments, que admite utilizar los lenguajes C, C++ y Asembler. Y en la interfaz permite el acceso a herramientas de monitoreo en tiempo real de variables que interesen observar.

Debido a la versatilidad del DSC, se pueden utilizar los protocolos de comunicación CAN (Controller Area Network), I²C (Inter-Integrated Circuit) y RS232 (Recommended Standard 232), por lo que en el diseño de la placa base del DSC son considerados los tres anteriores a modo de permitir a futuro su utilización.

Se opta por la utilización de una conexión serial RS232 bidireccional entre el SBC y el DSP, con una velocidad de 115 kBps, transmisión de 8 bit, sin bit de paridad ni control de flujo por hardware. Donde los datos son enviados y recibidos en forma de cadenas de caracteres.

La selección de RS232 se realiza por ser un protocolo conocido y por que mediante la herramienta hyperterminal del SBC se pueden observar los datos enviados y recibidos.

3.6. Sensores

Para la realización de mediciones se utilizan sensores: LEM para corriente continua, LEM para corriente alterna y transformadores de señal de razón 230/6 [V] para obtener los valores de tensión en la red y el generador.

Los datos a adquirir se presentan en la siguiente tabla.

Datos de	Mediciones	Sensor
Corrientes de línea en la red	$I_{ared}, I_{bred}, I_{cred}$	LTSR 25-NP 25A 5V
Corrientes de línea en el generador	$I_{agen}, I_{bgen}, I_{cgen}$	LTSR 25-NP 25A 5V
Tensiones fase-neutro en la red	$V_{anred}, V_{bnred}, V_{cnred}$	BV020-5480.0 230V/6V 13.3MA
Tensiones fase-neutro en el generador	$V_{angen}, V_{bngen}, V_{cngen}$	BV020-5480.0 230V/6V 13.3MA
Corriente de excitación	I_r	HXS 20-NP 0A 5V
Corrientes de línea en las resistencias de desahogo	$I_{ades}, I_{bdes}, I_{cdes}$	LTSR 25-NP 25A 5V

Tabla 3.3: Adquisición de datos

Debido a que las entradas análogas del DSC sólo pueden recibir señales positivas y en un rango de 0 a 3[V], es necesario que las mediciones realizadas sean acondicionadas para su procesamiento. Por lo que se diseña e implementa una placa de filtros (fig. 3.10). Se utilizan divisores de tensión para limitar las señales a los 3[V] admisibles y luego se les suma una señal continua de 1,5 [V] para obtener siempre valores positivos. Las señales alternas pasan por un filtro pasa bajos a modo de eliminar ruido de alta frecuencia [8] (fig. 3.9). Y luego todas las mediciones ingresan al DSC mediante los puertos ADC y posteriormente por un filtro digital programado en el controlador para obtener de esta forma mediciones confiables.

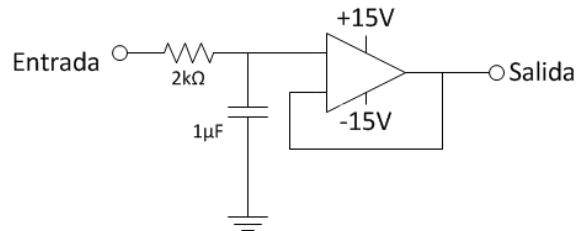


Figura 3.9: Filtro pasa bajos utilizado.

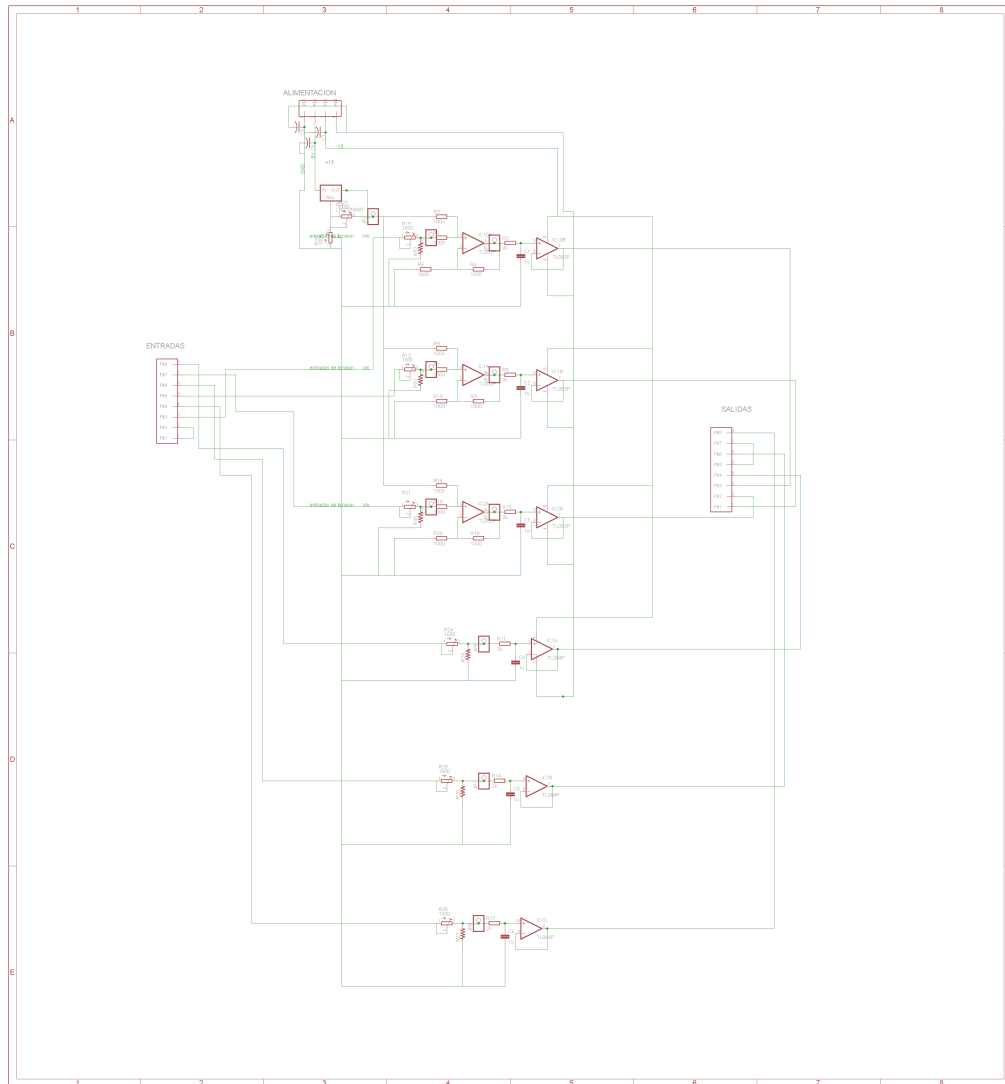


Figura 3.10: Esquemático de placa de filtros.

3.7. Actuadores

Para que el control pueda ejercer acciones sobre la central se necesitan diversos actuadores. En la tabla 3.4 se presentan los actuadores de la MCH.

Actuador	Modifica	Controla
válvula mariposa con servomotor	caudal de circuito hidráulico	frecuencia y potencia activa
chopper	corriente de excitación	tensión y potencia reactiva
punteo rectificador	corriente de carga de baterías	tensión en baterías
triac con resistencias de desahogo	potencia disipada	frecuencia y potencia activa

Tabla 3.4: Actuadores

También se podrían considerar como actuadores los relés que accionan la válvula mariposa, pero éstos se consideran parte del bloque de control asociado.

Un elemento que efectúa una labor importante de mencionar son los contactores, pues estos permiten cerrar o abrir los circuitos eléctricos de la central, permitiendo operar en isla o sincronizado. En la MCH se dispone de un contactor para la red K_{Red} y un contactor para el generador K_{Gen} .

Capítulo 4

Sistema a controlar

En el presente capítulo se expone el prototipo de la MCH como sistema, con la obtención de mediciones de las variables y los diagramas de control utilizados.

4.1. Mediciones y procesamiento

Como parte del desarrollo electrónico del prototipo se implementó la adquisición y el filtrado análogo de las señales desde los sensores. Por lo que es necesario el procesamiento digital de estas para poder obtener las mediciones de las variables a controlar y poder efectivamente tomar las decisiones de control y efectuar las acciones correspondientes (fig. 4.1).

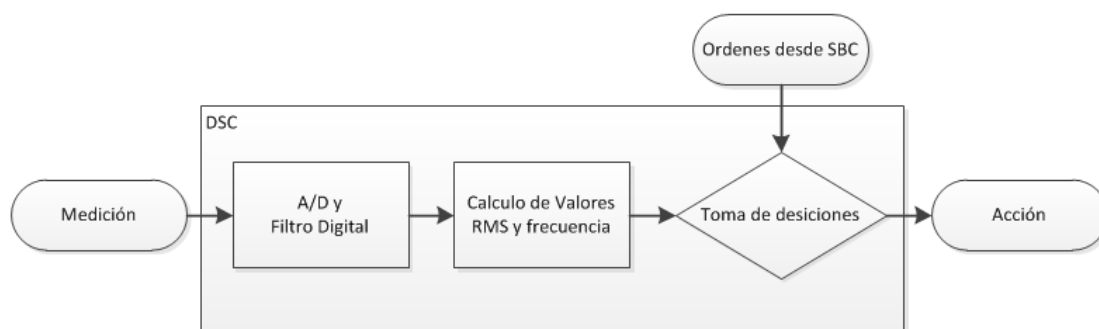


Figura 4.1: Procesamiento.

En el primer bloque se obtienen valores digitales y se acondicionan para corresponder con los valores instantáneos de las mediciones. En el segundo bloque se realizan las transformaciones necesarias para obtener las mediciones de tensiones efectivas, corrientes efectivas,

frecuencia y secuencia. Y en el ultimo bloque del procesamiento se concentra la programación del sistema de control de la MCH.

A continuación se explica como se procesa la información instantánea para la obtención de las mediciones deseadas.

4.1.1. Medición de tensión y corriente alterna

La obtención de los valores efectivos se puede realizar mediante cálculos simples como obtención del máximo y división por raíz de dos, como también mediante la aplicación de transformadas entre espacios (capítulo 2 [13]).

Aprovechando la herramienta de CCS se implementan ambas alternativas (fig. 4.2 y 4.3).

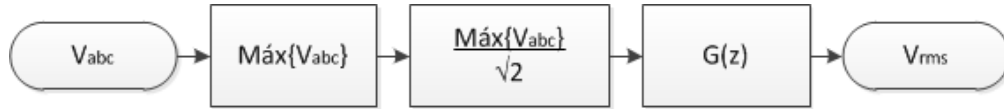


Figura 4.2: Obtención de valores efectivos mediante máximo.

Para la obtención del máximo se comparan el valor instantáneo de tensión y el máximo anterior. La variable máximo se reinicia cada 10000 ciclos del programa, lo que equivale a 2,38 [s]. Luego el valor obtenido se divide por raíz de dos y es filtrado a 1 [HZ], adquiriendo finalmente el valor instantáneo de tensión.



Figura 4.3: Obtención de valores efectivos mediante transformada.

Suponiendo que la tiene secuencia positiva, la obtención de los valores efectivos mediante la transformada de Clarke se realiza como sigue:

$$\vec{v}_{\alpha\beta\gamma} = T_{\alpha\beta\gamma} \vec{v}_{abc} \quad (4.1)$$

$$\begin{bmatrix} v_{\alpha} \\ v_{\beta} \\ v_{\gamma} \end{bmatrix} = \begin{bmatrix} 1 & -\frac{1}{2} & -\frac{1}{2} \\ 0 & \frac{\sqrt{3}}{2} & -\frac{\sqrt{3}}{2} \\ \frac{1}{2} & \frac{1}{2} & \frac{1}{2} \end{bmatrix} \begin{bmatrix} v_a \\ v_b \\ v_c \end{bmatrix} \quad (4.2)$$

Luego se normaliza el vector obtenido para obtener un valor efectivo, suponiendo un consumo equilibrado:

$$V_{rms} = \frac{2}{3\sqrt{3}} \sqrt{v_\alpha^2 + v_\beta^2 + v_\gamma^2} \quad (4.3)$$

En ambas alternativas se utiliza un filtro pasabajos de 1 [Hz] calculado mediante la discretización de un filtro continuo de 1 [Hz] (capítulo 4 [14]):

$$G(s) = \frac{2\pi}{s + 2\pi} \quad (4.4)$$

Discretizando con MATLAB:

$$G(z) = \frac{0,0014948780567423}{z - 0,99850519432577} \quad (4.5)$$

Los procedimientos anteriores permiten calcular un valor promedio estimado de las tensiones y corrientes. Si se pretende controlar potencia, es necesario utilizar la transformada de Park para obtener las mediciones en los ejes dq0 a partir de Clarke, para simplificar el control.

$$\vec{v}_{dq0} = T_{dq0} \vec{v}_{\alpha\beta\gamma} \quad (4.6)$$

$$\begin{bmatrix} v_d \\ v_q \\ v_0 \end{bmatrix} = \begin{bmatrix} \cos(\theta) & \sin(\theta) & 0 \\ -\sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} v_\alpha \\ v_\beta \\ v_\gamma \end{bmatrix} \quad (4.7)$$

Para lo cual es necesario obtener θ , lo cual se explica en la sección siguiente.

Para poder seleccionar una de las alternativas anteriores se debe considerar el desempeño de los sensores, pues la utilización de transformadas es muy sensible a errores de medición dado que sus cálculos son instantáneos.

4.1.2. Medición de frecuencia

Tal como en la sección anterior la medición de frecuencia se puede realizar mediante cálculos simples como lo es el contar los cruces por cero o partiendo de la transformada de Clarke con un PLL (Phase-locked loop), figuras 4.4 y 4.6 respectivamente.

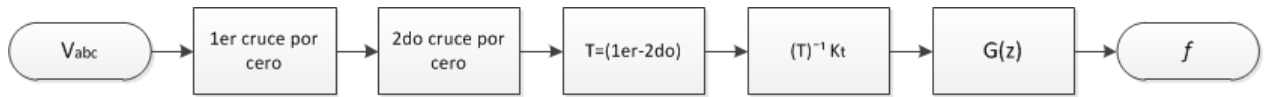


Figura 4.4: Obtención de frecuencia mediante cruces por cero.



Figura 4.5: Obtención de frecuencia mediante PLL.

La obtención de frecuencia mediante PLL se realiza como sigue:

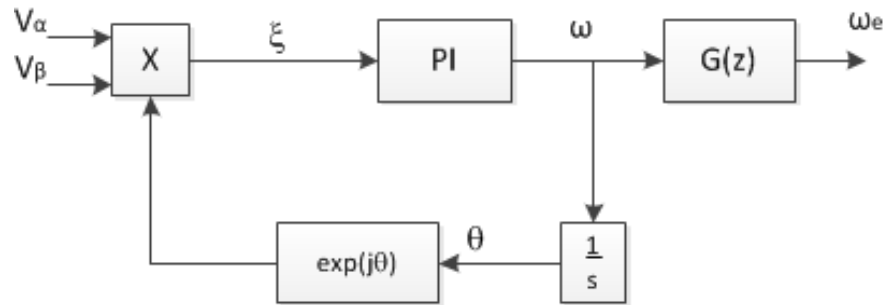


Figura 4.6: PLL para estimación de frecuencia.

Donde ξ corresponde a:

$$\xi = v_{\beta} \cos \theta - v_{\alpha} \sin \theta \quad (4.8)$$

Para la estimación de frecuencia, el PLL ajusta el ángulo θ mediante un control PI, intentando llevar ξ a cero, lo que indica que θ corresponde al ángulo de la señal. La salida del bloque PI corresponde a la velocidad angular, que al ser integrada permite calcular θ , para luego ser filtrado a 1[Hz] por $G(z)$ de la ecuación 4.5.

Ambos métodos permiten estimar frecuencia en un sistema carente de sensores para este propósito. La diferencia principal entre ambos métodos radica en que el PLL no se enclava ante variaciones bruscas de frecuencia de la señal, por lo que esta alternativa no puede ser utilizada a no ser que la señal tenga una frecuencia estable como es el caso de la red eléctrica.

4.1.3. Medición de corriente de campo

Tras realizar pruebas al sensor de corriente continua, se obtiene la siguiente relación tensión en la salida del sensor respecto a la corriente medida con un multímetro:

Corriente [A]	Tensión [V]
1,94	2,193
1,8	2,189
1,68	2,187
1,4	2,181
1,18	2,176
1,09	2,174
0,82	2,168
0,73	2,166
0,62	2,164
0,48	2,161
0	2,151

Tabla 4.1: Relación mediciones

Donde se puede apreciar que se necesita precisión de hasta tres decimales en la medida, dado que el rango de mediciones es de orden de las milésimas. Por lo que se aplica el filtro digital de la ecuación 4.5 para eliminar los posibles ruidos.

4.2. Modos de operación del sistema

La MCH al ser pensada como un dispositivo inteligente que escoja automáticamente el modo de operación adecuado, en su diseño debe tener en cuenta los modos en los cuales puede operar, los estados y las posibilidades de control que presenta cada uno.

Al utilizarse un generador sincrónico se obtienen dos modos de operación: aislado de la red eléctrica y sincronizado a la misma. Adicionalmente se consideran dos estados para el encendido y el apagado de la central: en espera y detenido.

El estado detenido corresponde a la central apagada y desconectada tanto del consumo como de la red, con la válvula mariposa cerrada. El estado en espera corresponde a llevar a la central al punto de operación de 50 [Hz] y 220 [V] a la espera de conectarse al consumo o a la red.

En la figura 4.7 los recuadros azules corresponden a los modos de operación y los rojos a estados intermedios que tienen relación con el control que se debe realizar para ir de un modo a otro. La máquina parte en estado detenido, se realiza la acción de encendido y se lleva a estado en espera, donde si hay o no red decide si sincroniza o toma carga. De modo conectado puede pasar a isla mediante la acción aislar. Y de los modos isla y conectado puede pasar a detenido apagando la central.

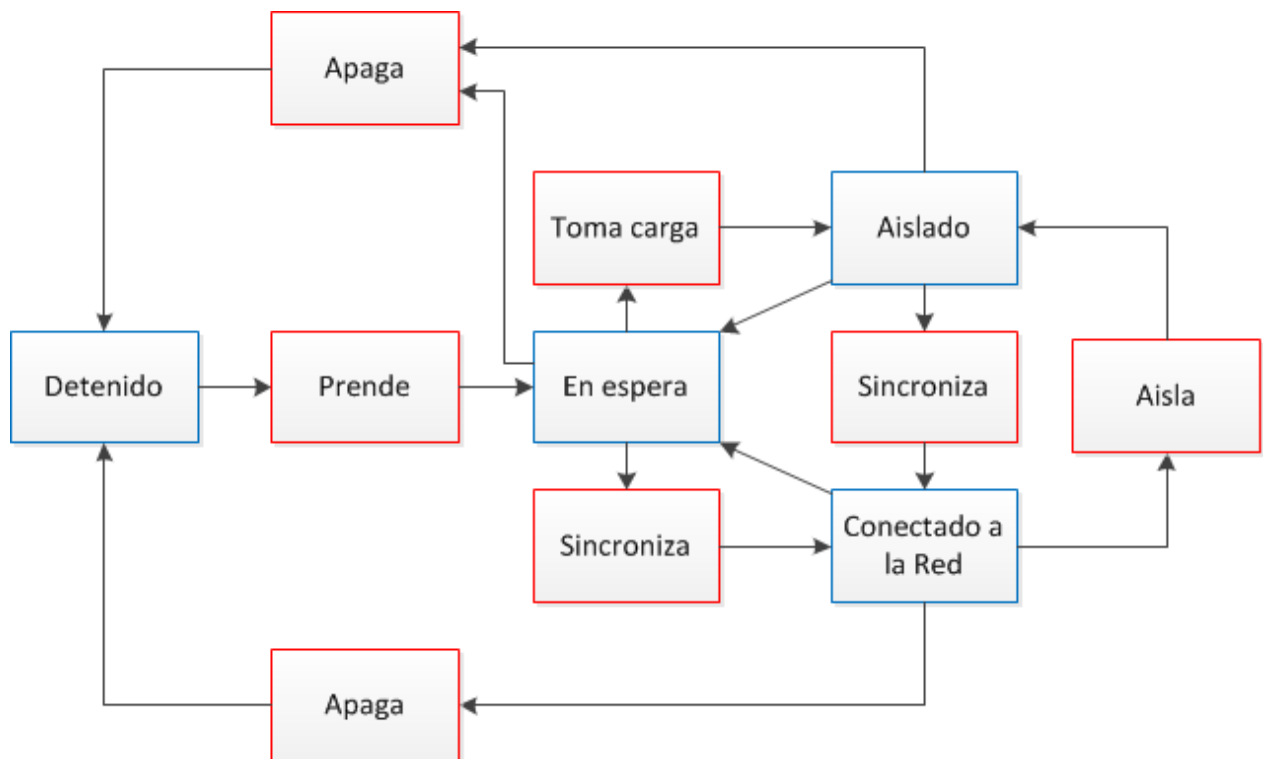


Figura 4.7: Esquema de modos de operación de MCH.

A continuación se describen brevemente los modos aislado y conectado.

4.2.1. Modo aislado de la red



Figura 4.8: Modo de operación aislado.

Como dice el nombre, el funcionamiento del equipo es en modo isla, alimentando un consumo independiente de la red y conectado directamente al sistema MCH, el cual debe cumplir con los requerimientos de tensión y frecuencia constantes (380 a 400 [V_{ff}] y 50 [Hz]). Para satisfacer lo anterior se deben controlar el caudal que afecta la velocidad de giro e la turbina y a su vez a la frecuencia, y la corriente de campo que determina la tensión generada. Pero, como verá en la sección 5.1.2, la tensión en bornes no es independiente de la velocidad de giro, lo cual se debe tener en cuenta al momento de pensar la estrategia de control.

Dadas las características de la válvula mariposa, sólo en condiciones de laboratorio se puede modificar el caudal del circuito hidráulico. Por lo que en la práctica se estudiará el recurso del que se disponga determinando un intervalo de caudales durante el año y se limitará la operación de la plata para operar con en base a esta restricción.

4.2.2. Modo conectado a la red

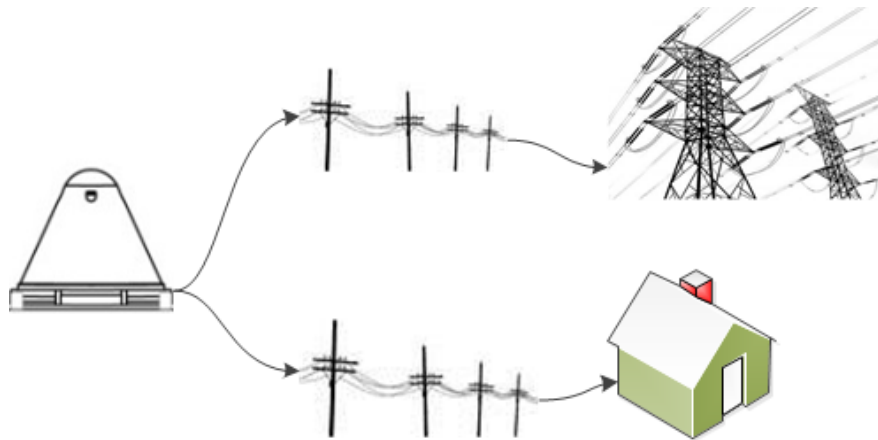


Figura 4.9: Modo de operación conectado a la red.

Este modo corresponde a la MCH sincronizada a la red eléctrica. La operación de sincronización [15] se logra mediante la minimización de los valores absolutos de los siguientes parámetros:

- Diferencia entre frecuencia de la red y el generador.
- Diferencia de tensión entre la red y el generador.
- Diferencia de ángulo de fase entre los fasores de tensión del generador y la red.

Posterior a la conexión, si se considera a la red como una barra infinita, el sistema MCH es muy pequeño en comparación a ésta, por lo que en teoría ya no es posible ejercer un control sobre la frecuencia y la tensión pues son fijados por la red. Por lo que el efecto de utilizar los actuadores se traduce en control sobre potencias activa y reactiva de la MCH (capítulo 8 [16]). Ahora bien dadas las características de la red eléctrica chilena, la conexión de un PMGD en un alimentador puede permitir control sobre tensión en el punto de conexión, mas no sobre la frecuencia (capítulo V [17]).

4.3. Diagramas de estado y control

El correcto funcionamiento del sistema depende del adecuado desempeño del generador sincrónico. Los lazos de control importantes son:

- Control de frecuencia y
- Control de tensión

Gracias al desarrollo de la electrónica se tiene la adquisición de información y los actuadores que permiten manipular ciertas variables con miras a que las mediciones se acerquen a los puntos de operación deseados. Falta presentar como se realiza el control con las herramientas mencionadas y junto a esto es ahondará en el desarrollo electrónico asociado a los actuadores.

Es necesario tener una visión general de la MCH como planta a controlar. La figura 4.10 permite comprender cuales son las variables controladas.

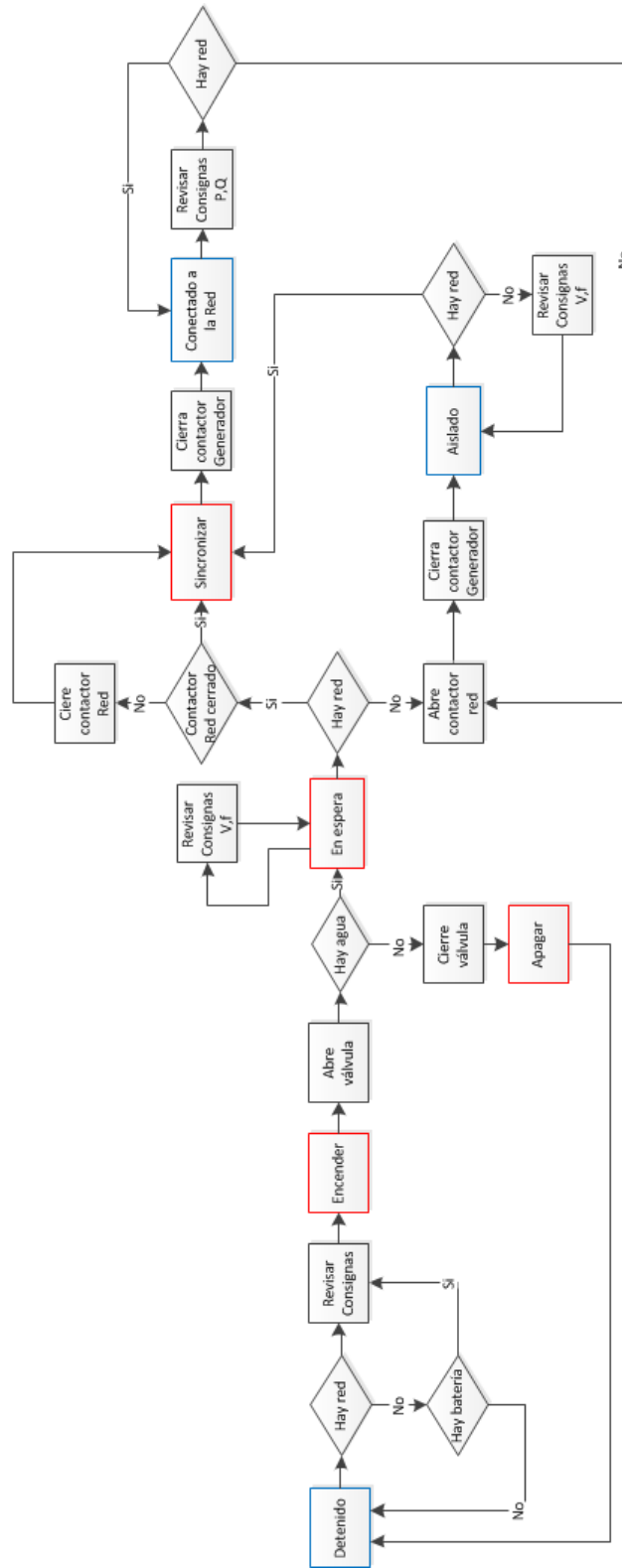


Figura 4.10: Esquema principal de la MCH.

A continuación se explica el esquema de la figura 4.10:

- 1.- El estado inicial de la planta es detenido. Donde si hay red, la electrónica de la MCH se alimenta de ésta y si no, entonces revisa si las baterías se encuentran cargadas, de no ser así se mantendría en este estado.
- 2.- Si puede ser energizada la electrónica, se accede a la MCH mediante el SBC y determina el modo de funcionamiento automático o manual, lo cual corresponde a la primera revisión de consignas, donde la máquina reconoce como debe operar y pasa a estado encender.
- 3.- En estado encender se abre completamente la válvula mariposa y alimenta el campo del generador con I_r correspondiente a la generación de 220[V] lo cual sólo se debe cumplir si existe el recurso para esto, con lo cual se verifica si hay recurso suficiente. De ser así pasa a modo en espera, si no, se cierra la válvula, apaga el sistema y pasa a detenido.
- 4.- En espera se verifica generación de 220[V] y 50[Hz]. Si cumple consignas procede a comprobar existencia de red. Para corroborar existencia de red eléctrica, se revisan las mediciones de los sensores dispuestos para ésto.
 - a.- Si hay red se revisa estado de K_{Red} . Si se encuentra abierto se debe cerrar, con lo cual el consumo se encontraría siendo energizado por la red. Con K_{Red} cerrado se procede a sincronizar el generador con la red. Lograda la sincronización se cierra K_{Gen} con lo cual se produce la conexión a la red. De esta forma la MCH se encuentra en modo de operación conectado a la red, donde las acciones posibles de control serán sobre potencias activa P y reactiva Q. Constantemente se constata existencia de red, de producirse desconexión se abre K_{Red} y cierra K_{Gen} cambiando a modo Aislado. De permanecer la red conectada se mantiene en el modo de operación.
 - b.- Si no hay red se abre K_{Red} y cierra K_{Gen} alimentando de esta forma el consumo desde el generador con lo cual la MCH se encuentra operando en modo aislado. Constantemente se comprueba existencia de red, lo cual al verificarse permite el paso a estado sincronizar. De no haber red se mantiene revisión de consignas de tensión y frecuencia.

Como se puede observar en la figura 4.10 es necesario establecer consignas que se revisen a modo de que se puedan realizar acciones de control. La figura 4.11 permite la apreciación de éstas según el modo de operación escogido.

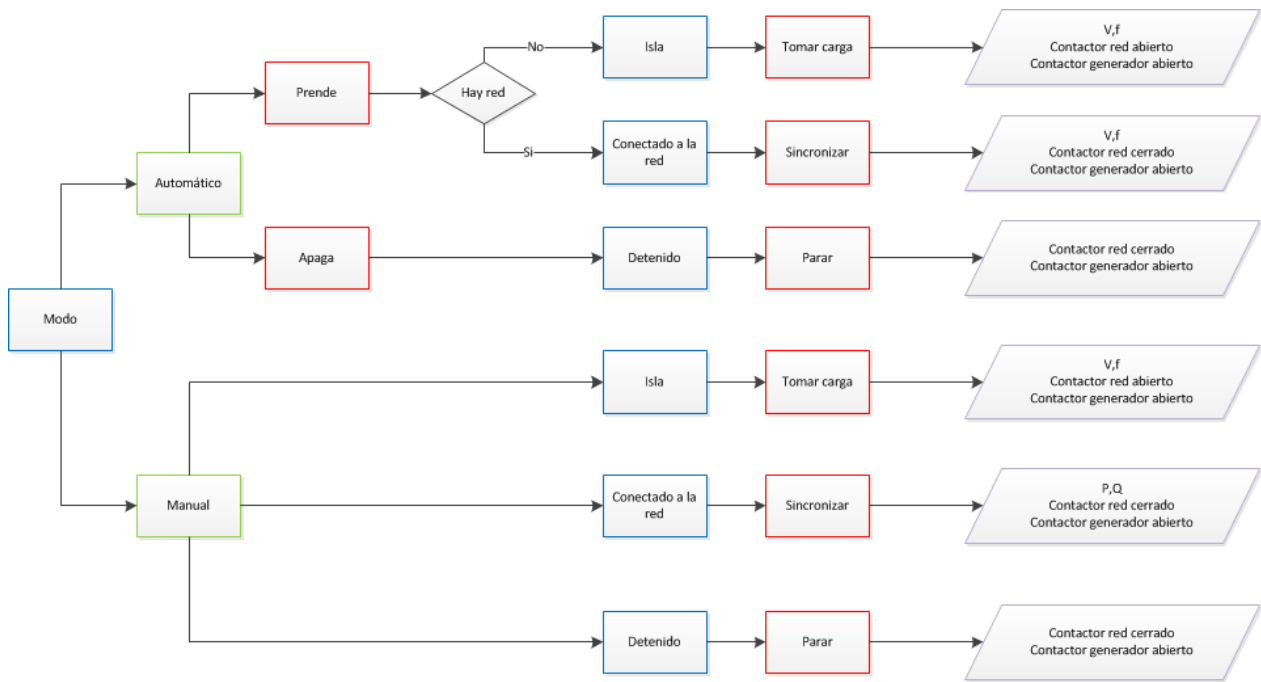


Figura 4.11: Esquema de consignas.

Primeramente se revisan las consignas de modo automático o manual y luego según el modo de operación se revisan las consignas y estados de contactores respresentados por rombos en la figura 4.11.

Con lo anterior se puede profundizar en los subsistemas, asociados a los actuadores, que hacen posible el control de la MCH.

4.3.1. Control de tensión

El control de voltaje mediante un AVR (Automatic Voltage Regulator) permite operar el sistema de excitación del generador, ésto es modificar la corriente rotórica o de excitación que afecta directamente la tensión generada E .

Existen diversos sistemas de excitación (ver capítulo 7 [21]), se opta por un *chopper* o recortador (fig. 4.12), un sistema de excitación estático independiente correspondiente a un convertidor DC-DC acorde a una señal de control PWM proveniente del DSC. Obteniendo como resultado una corriente continua variable para alimentación del campo del generador.

Según el modo de operación de la MCH, la variable a controlar por este sistema es la tensión (caso aislado) o bien la potencia (caso sincronizado), pero en ambos casos la variable manipulada es la corriente de excitación mediante el actuador chopper.

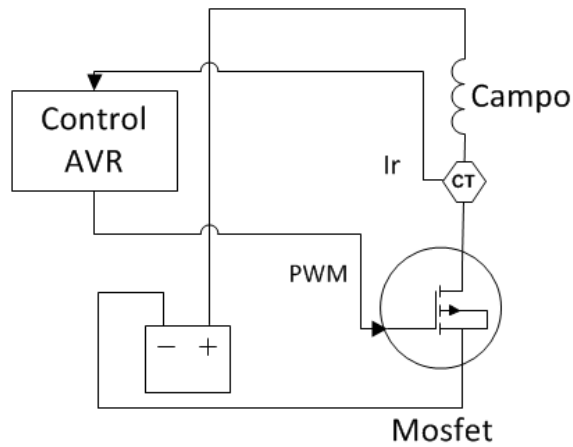


Figura 4.12: Regulador de voltaje automático.

Materializando lo esquematizado en 4.12 se diseña e implementa el actuador *chopper* en 4.13.

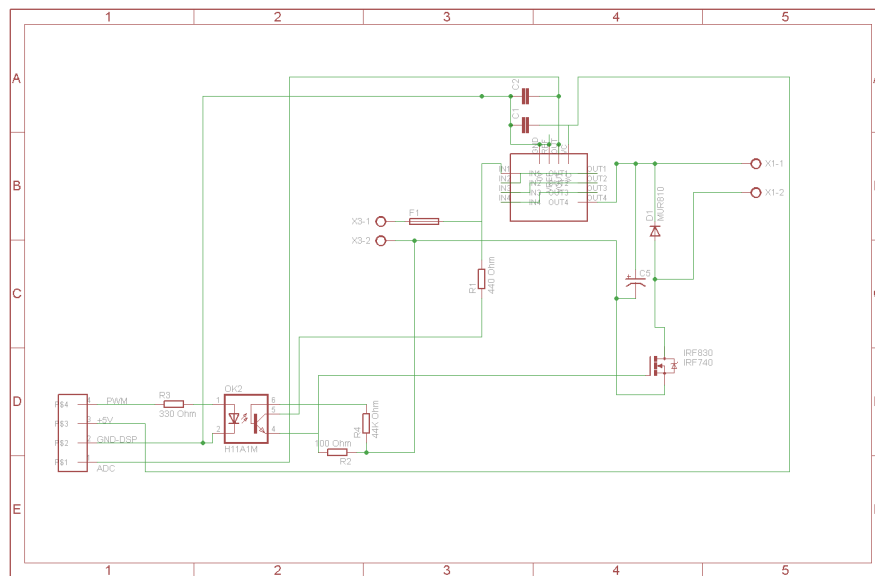


Figura 4.13: Esquemático de *chopper* implementado.

Donde se puede apreciar en las coordenadas B1, BC3 y B5 las entradas y salidas del circuito. B1 corresponde a la entrada de la señal de control, alimentación para el sensor y a su vez la salida de la medición del campo. BC3 es la conexión de las baterías, esto es la señal DC que será recortada. Y B5 es la salida del *chopper*, la señal DC recortada. El optoacoplador en D2 permite aislar electrónicamente la señal de control que es de menor potencia respecto a la señal proveniente de las baterías. En B4 se ubica el sensor HXS 20-NP enunciado en la tabla 3.3 para monitoreo de I_r . Y finalmente es el mosfet en D4 el encargado de realizar la conversión DC-DC que permite la obtención de I_r .

En la figura 4.14 se presenta un diagrama, a modo explicativo, para entender como se debe comportar el sistema de excitación y las acciones de control que se deben realizar cuando el objetivo es mantener un voltaje de referencia.

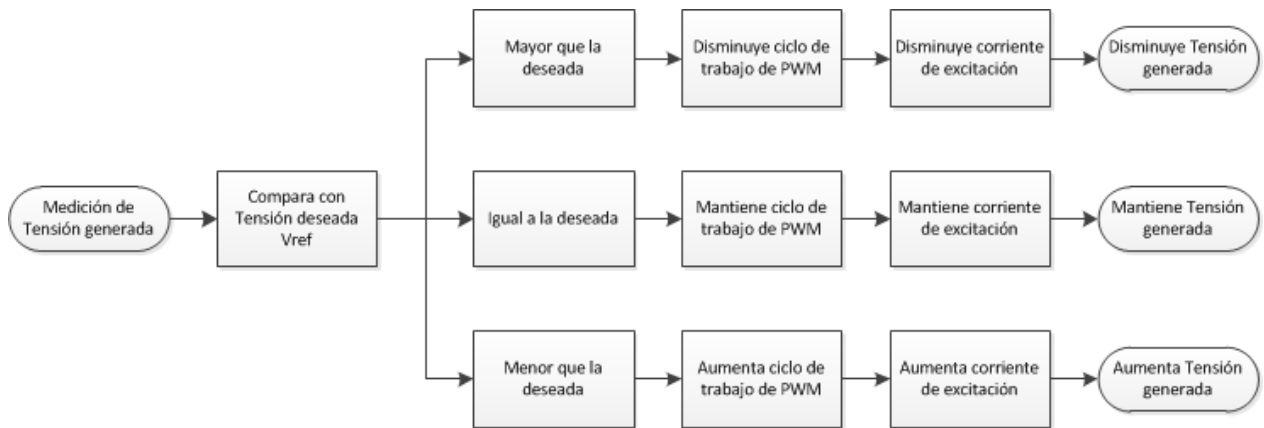


Figura 4.14: Diagrama de comportamiento de *chopper*.

El diagrama de control que satisface lo anterior corresponde a la figura 4.15, en la cual el controlador se diseña en el capítulo 5.



Figura 4.15: Diagrama de control de *chopper*.

Para el aumento y disminución de la corriente se deben tener en consideración los límites aceptados por el campo de corriente mínima de 0,5 [A] y máxima de 1,5 [A] para que el generador no salga de sincronismo.

Para el control de potencia reactiva durante el modo de operación sincronizado se procede de forma análoga.

4.3.2. Control de frecuencia y desequilibrio

El control de la frecuencia en la literatura (capítulo 6 [16]) se consigue mediante mecanismos gobernadores de velocidad. Éstos se componen por: un gobernador que en este caso es

el error de frecuencia calculado; una válvula controlada, en el sistema de tipo mariposa; un mecanismo de control de velocidad, como se dijo anteriormente un servomotor; un cambiador de velocidad, que considera a los relés que controlan al servo. Por lo que para la MCH se tendría el mecanismo de la figura 4.16.

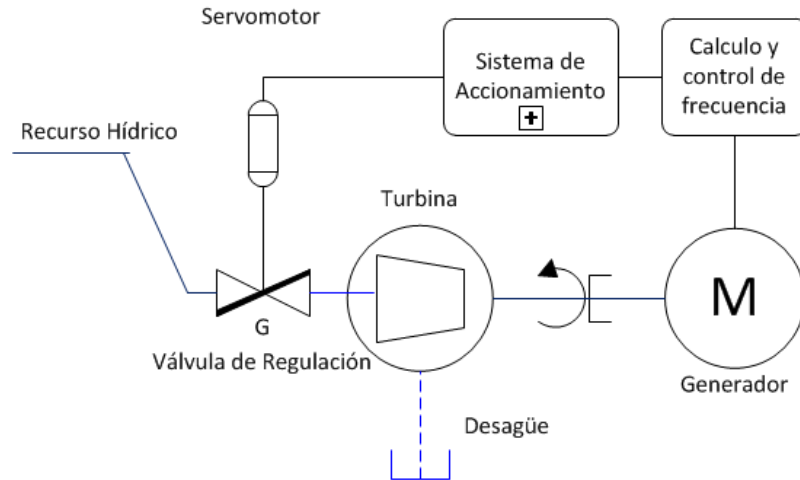


Figura 4.16: Mecanismo gobernador de velocidad.

Donde el sistema de accionamiento considera la utilización de relés que será explicado en la sección 4.3.3.

El mecanismo anterior permite el control del caudal que ingresa a la turbina por lo que influye directamente en la velocidad de giro del eje del generador, pero éste es un mecanismo de lenta respuesta ya que la válvula en realizar una carrera completa demora 30 segundos, por lo que es necesario un sistema de respuesta rápida que permita a la central reaccionar ante variaciones de carga dándole tiempo a la válvula colocarse en una posición deseada. Por esto se piensa en la utilización de resistencias de desahogo, lo cual será explicado en los siguientes párrafos.

Las resistencias de desahogo, como actuador, se utilizan para regular la frecuencia de la máquina en modo en espera, mediante disipación de energía por las mismas emulando a un consumo. En modo aislado, permite regular frecuencia frente a las variaciones de potencia del consumo y en modo conectado a la red, para regular la potencia activa. El subsistema considera una resistencia trifásica, de valor por fase de aproximadamente $10[\Omega]$, conectada en estrella con capacidad para disipar hasta $3 [kW]$, lo que se traduce en disipación de hasta $13,64[A]$ a $220[V]$ por fase.

Para la utilización de las resistencias se utilizó un diseño existente (revisar anexo B.1.5 y B.2.3), el cual se modificó para accionamiento de las mismas y en conjunto conformar al actuador. En la figura 4.17 se puede apreciar en un diagrama simplificado el actuador.

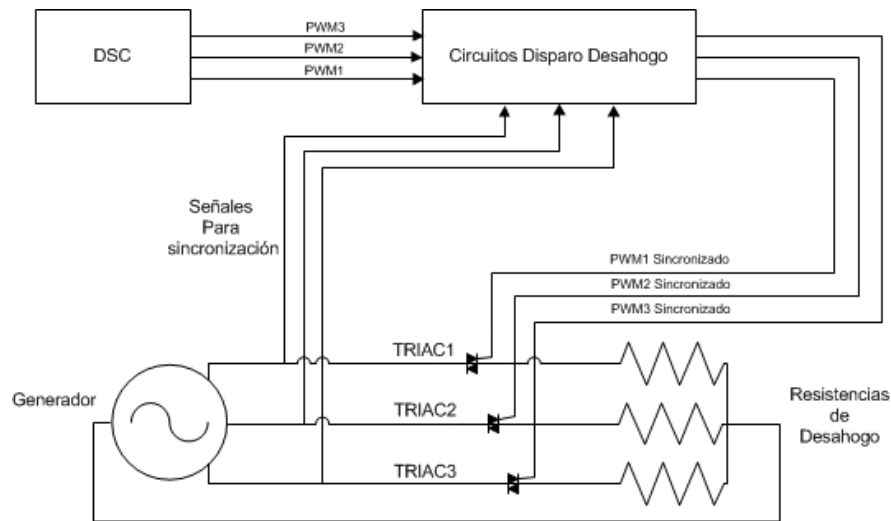


Figura 4.17: Actuador resistencias de desahogo.

El control del subsistema se logra mediante el uso de TRIACs que permiten regular el voltaje aplicado a las resistencias. Para gatillar estos dispositivos se utilizan tres señales continuas variables generadas a partir de tres PWM originados en el DSC. Que son filtrados para obtener una señal continua que controla el ancho de pulsos sincronizados con las tensiones del generador. Estos pulsos son optoacoplados para aislar las señales de control de menor potencia en comparación a las señales en el bloque de los triacs.

En la figura 4.18 se presenta un diagrama para entender como se debe comportar el sistema de resistencias de desahogo y las acciones de control que se deber realizar cuando el objetivo es mantener una frecuencia de referencia.

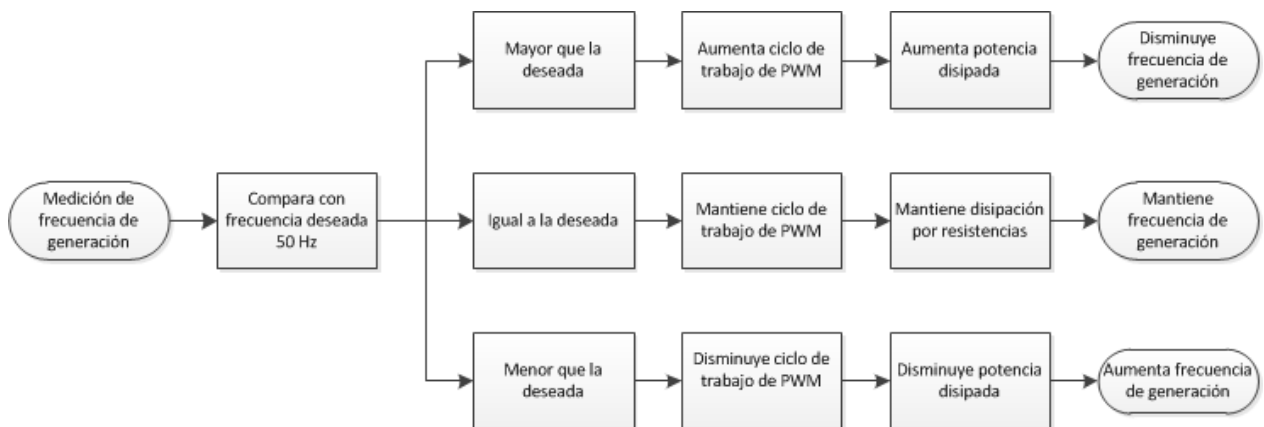


Figura 4.18: Diagrama de comportamiento de desahogo.

El diagrama de control que satisface lo anterior corresponde a la figura 4.19, en la cual el controlador se diseña en el capítulo 5. El objetivo es que las resistencias sean una reserva



Figura 4.19: Diagrama de control de resistencias de desahogo.

primaria de potencia generada y al mismo tiempo una cantidad de potencia desconectable, admitiendo una reacción rápida ante efectos del consumo, para luego, mediante la válvula mariposa, equilibrar la potencia hidráulica del sistema dejando al desahogo volver a su referencia.

Adicionalmente se piensa como forma de controlar los posibles desbalances y las variaciones de demanda, compensando la pérdida o incremento instantáneo de carga como se ejemplifica en la figura 4.20.

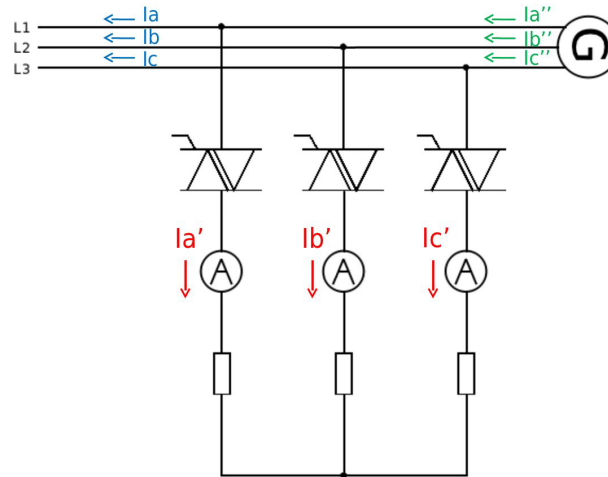


Figura 4.20: Esquema compensación de carga.

Donde el consumo de la carga es I , la disipación mediante desahogo es I' y la demanda equilibrada que ve el generador es I'' .

Por lo que al balancear la carga se obtiene:

$$I'' = I + I' \quad (4.9)$$

Lo cual se puede apreciar en forma gráfica en el siguiente gráfico a modo ejemplificativo.

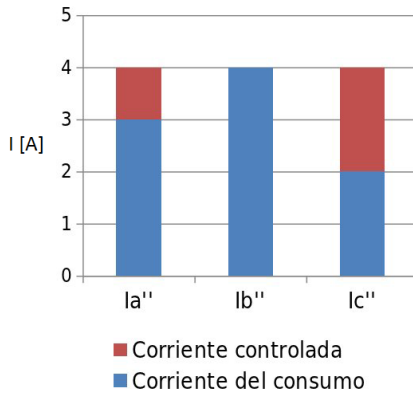


Figura 4.21: Compensación de carga.

4.3.3. Procedimiento de partida

La partida de la MCH considera la transición del estado detenido a en espera (fig. 4.10). En esta transición, para encender la central, debe manipularse la posición de la válvula mariposa a modo de obtener el recurso hidráulico que se necesita.

Tanto para el accionamiento de la válvula como para la utilización de K_{Red} y K_{Gen} se diseña e implementa una placa de relés, cuyo desarrollo se puede revisar en el anexo B.2.2.

En la figura 4.22 se pueden apreciar los relés necesarios para el accionamiento del motor que mueve la válvula.

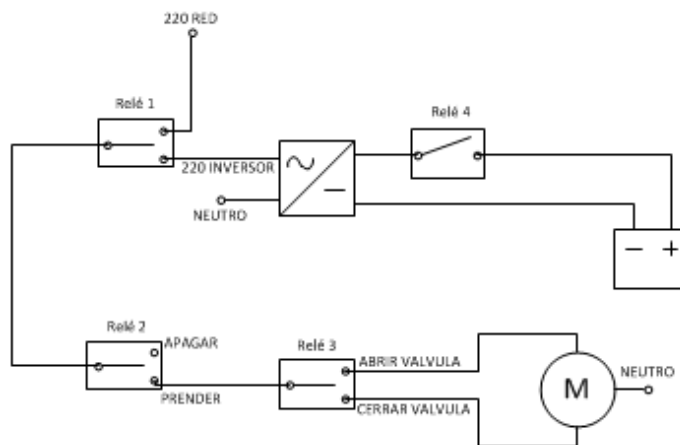


Figura 4.22: Sistema de accionamiento de la válvula por relés.

En 4.22 los relés enumerados realizan las siguientes labores:

- 1.- Su accionamiento permite la conexión del motor de la válvula a la red o al inversor.
- 2.- Se utiliza para prender o apagar el motor.
- 3.- Permite abrir o cerrar la válvula, indicando el sentido de giro del motor.
- 4.- Realiza la conexión del inversor a las baterías. Pues en caso de no existir red a la cual conectarse, el inversor permite obtener alimentación alterna mediante la transformación DC-AC desde las baterías.

La secuencia para poder mover la válvula es como sigue:

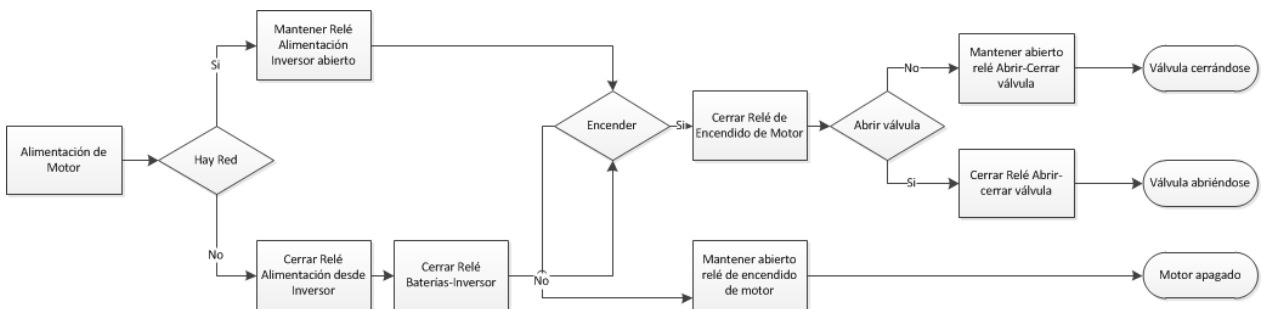


Figura 4.23: Utilización de válvula.

Para que la MCH pueda generar electricidad es necesario abrir completamente la válvula y para el apagado de la central debe cerrarse de igual forma. En primera instancia se controlará la válvula con este propósito y posteriormente se evaluará el efectuar un control de frecuencia y potencia activa con este elemento.

El procedimiento de partida finaliza llevando a la MCH al estado en espera resultado en el control de tensión y frecuencia para obtener como punto de operación el nominal de 220 [V] y 50 [Hz] y así pasar al siguiente modo de operación.

4.3.4. Procedimiento de sincronización

Existiendo red, se observan los parámetros de ésta y controlan tensión y frecuencia para equiparar los parámetros de la MCH a los de la red. Como se vió en 4.2.2 no es suficiente lo anterior, se debe también igualar el ángulo del fasor tensión. Esto se puede abordar de distintas formas. Una de ellas es midiendo red y generador e ir modificando la frecuencia del generador, mediante variación de potencia desahogada, a modo de lograr que durante un intervalo de tiempo establecido la diferencia entre dos puntos $\Delta \epsilon$, entorno al cero de las mediciones sea nula (fig. 4.24), lo cual indicaría que las señales se mueven en el mismo sentido y con la misma amplitud.

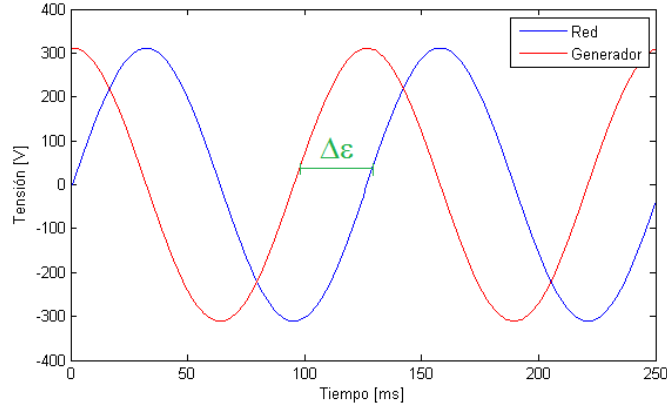


Figura 4.24: Sincronización llevando $\delta \epsilon$ entre dos puntos a cero .

Otra forma de lograr las condiciones para sincronización es mediante un PLL y control de tensión en el espacio de Clarke (fig. 4.25) del generador considerando la secuencia positiva de las tensiones de frecuencia fundamental [18]. Esta metodología se explica a continuación.

El vector de secuencia positiva de tensión de frecuencia fundamental se obtiene mediante una variación de Fortescue:

$$\vec{v}_{+abc}^f = T_{+seq} \vec{v}_{abc}^f \quad (4.10)$$

$$\begin{bmatrix} v_{a+}^f \\ v_{b+}^f \\ v_{c+}^f \end{bmatrix} = \begin{bmatrix} 1 & a & a^2 \\ a^2 & 1 & a \\ a & a^2 & 1 \end{bmatrix} \begin{bmatrix} v_a^f \\ v_b^f \\ v_c^f \end{bmatrix} \quad (4.11)$$

Donde

$$a = e^{j\frac{2\pi}{3}} = -\frac{1}{2} + \frac{\sqrt{3}}{2}e^{j\frac{\pi}{2}} \quad (4.12)$$

Luego se aplica Clarke al vector de tensiones y mediante su normalización se obtiene un vector cuyas componentes se pueden entender como las funciones seno y coseno usadas generalmente en técnicas de sincronización a la red.

$$v_{\alpha\beta+n}^f = \frac{v_{\alpha\beta+}^f}{\|v_{\alpha\beta+}^f\|_2} \quad (4.13)$$

Donde

$$\|v_{\alpha\beta+}^f\|_2 = \sqrt{(v_{\alpha+}^f)^2 + (v_{\beta+}^f)^2} \quad (4.14)$$

$$\sin(\theta_{f+}) = v_{\beta+n}^f \quad (4.15)$$

$$\cos(\theta_{f+}) = v_{\alpha+n}^f \quad (4.16)$$

Por lo que estos último se utilizarían para el PLL y de esta forma controlar la frecuencia del generador con miras a que tanto secuencia y fase sean iguales a las de la red.

Debido a limitaciones de programación del DSC no se puede trabajar con números complejos por lo que para obtener la parte imaginaria de Fortescue se debe retrasar la señal en un cuarto de ciclo, lo que se puede lograr mediante un filtro pasa bajos diseñados con tal objetivo o bien almacenando una cantidad de muestras determinadas que permitan obtener la medida correspondiente a un cuarto del periodo en retraso.

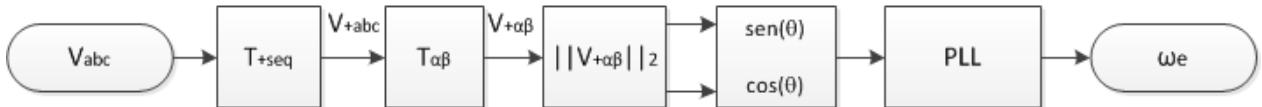


Figura 4.25: Sincronización mediante transformadas.

Luego comparando las velocidades angulares de la red y el generador, se ajusta la frecuencia del generador mediante el PLL para que se encuentren en la misma fase.

Cumpliendo con todo lo anterior debería ser inmediata la conexión de la MCH a la red, por lo que se necesitan contactores de sincronización. Estos contactores son activados por relés: un contactor para el generador y un contactor para la red.

Los estados de los contactores según la operación fueron expuestos en el esquema de consignas anteriormente. En general cuando encienda la MCH ambos contactores se encontrarán abiertos y cuando se cumplan las condiciones de sincronización el contactor de la red se cerrará.

Capítulo 5

Simulaciones e implementación de esquema de control para MCH

5.1. Modelación de la planta

En la MCH se pueden distinguir dos bloques necesarios para el proceso de generación. El primer bloque considera el aprovechamiento del recurso hídrico donde el principal protagonista es la turbina tipo turgo con dos inyectores. Y un segundo bloque contiene los componentes eléctricos, siendo conformado por el generador y el tablero de potencia con los bornes de conexión (fig. 5.1).

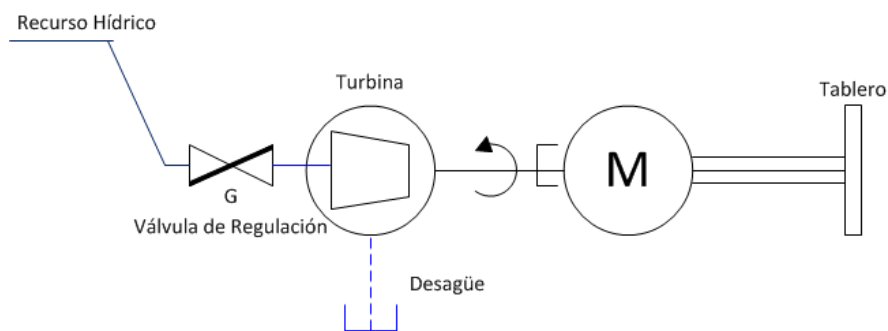


Figura 5.1: Esquema simplificado del sistema.

Para controlar el sistema es necesario comprender y modelar los elementos principales que lo componen.

5.1.1. Modelo y control de la turbina

Como se vio en 3.1.2, la turbina es de acción por lo que la energía transmitida es enteramente la cinética del agua.

El modelamiento de una turbina hidráulica para efectos de control requiere considerar variables de transmisión del agua en las tuberías, pero esto escapa de los alcances de este trabajo, por lo que se considerará una forma simplificada (fig. 5.2) [19] y en base a las simplificaciones asumidas se tienen los siguientes supuestos:

1. Resistencia hidráulica despreciable.
2. Tubería inelástica y agua incompresible.
3. La velocidad del agua varía de forma proporcional a la apertura de la válvula.
4. La potencia de salida de la turbina es proporcional al flujo de agua.

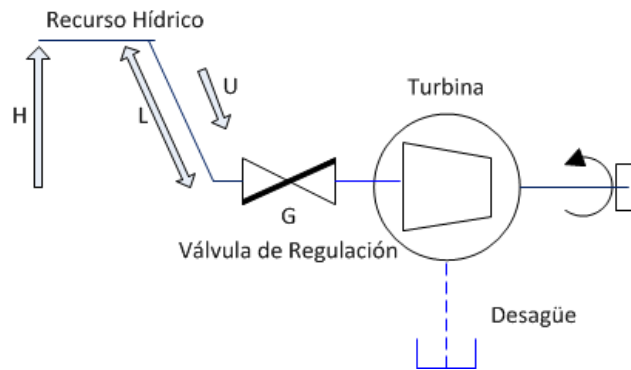


Figura 5.2: Sistema hidráulico.

Las características de la turbina y tubería son determinadas por tres ecuaciones básicas:

- a) Velocidad del agua en la tubería:

$$U = K_u G \sqrt{H} \quad (5.1)$$

Donde U es la velocidad del agua, G la posición de la válvula, H la altura hidráulica a la válvula y K_u es una constante de proporcionalidad.

- b) Potencia mecánica de la turbina P_m :

$$P_m = K_P H U \quad (5.2)$$

Donde K_P es una constante de proporcionalidad.

c) Aceleración de la columna de agua:

$$(\rho LA) \frac{d\delta U}{dt} = -A(\rho a_g) \delta H \quad (5.3)$$

Donde L es el largo del conducto, A el área de la tubería, ρ la densidad de masa, a_g la aceleración dada la gravedad, ρLA es la masa del agua en el conducto, $\rho a_g \delta H$ el cambio incremental en la presión a la entrada de la turbina y t el tiempo en segundos.

Con lo anterior se puede obtener un modelo ideal de la turbina y obtener un modelo análogo eléctrico (fig. 5.3) donde U es análoga a la corriente, G a la conductancia y H al voltaje.

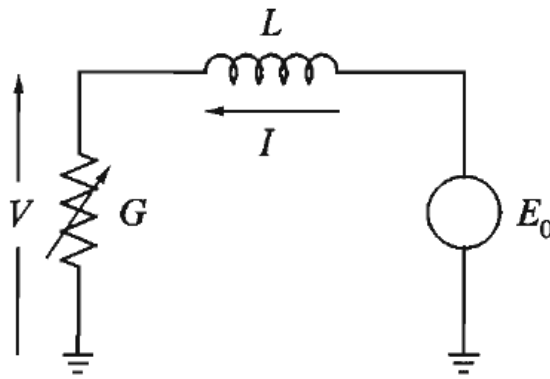


Figura 5.3: Modelo ideal de la turbina. Sección 9.1, Kundur[19]

Entonces cuando la carga disminuye inesperadamente, la corriente no cambia inmediatamente, pero el voltaje a través de la carga aumenta repentinamente a causa de la reducción de la conductancia o aumento de la resistencia. Esto causa que la potencia de salida abruptamente aumente; con una relación determinada por la inductancia L, I disminuye exponencialmente hasta un nuevo valor estable estableciendo una nueva potencia de salida.

De lo antes visto se pueden apreciar cuatro requerimientos especiales para el control de este bloque del sistema:

1. Limitar el aumento de velocidad a un límite aceptable ante un rechazo de carga.
2. Controlar la potencia generada mediante el control de la posición de la válvula.
3. Controlar la velocidad a la partida y al momento de conexión a la red.
4. Igualar la potencia generada a la requerida por la carga respondiendo a cambios de frecuencia en operación desconectada de la red.

Por lo que es de vital importancia considerar como actuador la válvula mariposa de la que se dispone, pero ésta tiene una respuesta muy lenta ante cambios bruscos en el sistema, por lo que se debe tener un segundo sistema de apoyo para solucionar los problemas de desequilibrio y desconexión.

5.1.2. Modelo y control del generador sincrónico

Las características constructivas y de funcionamiento de un generador sincrónico (fig. 5.4) se pueden apreciar en los capítulos 31 y 32 de [20].

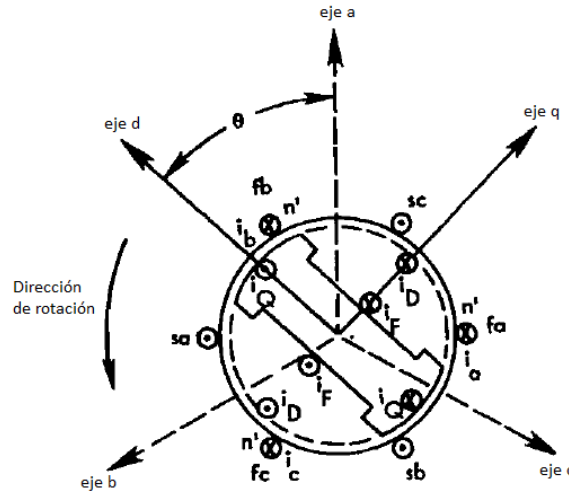


Figura 5.4: Representación ejes en una máquina sincrónica. Sección 4.2 [21]

Para estudiar el comportamiento de este tipo de máquinas resulta conveniente utilizar dos ejes ficticios denominados eje directo (d) y eje en cuadratura, (q) que giran de forma solidaria al rotor a velocidad de sincronismo, permitiendo esto simplificar las ecuaciones que modelan el comportamiento del sistema, pero aun así se tienen parámetros como las inductancias mutuas que varían en función del ángulo θ que depende del tiempo, por lo que para obtener ecuaciones más simples se transforman las ecuaciones a un sistema denominado dq0 mediante la transformada de Park.

Transformada dq0

Utilizando una proyección sobre los ejes d-q de las corrientes se tiene:

$$i_q = \frac{2}{3} [i_a \sin(\theta) + i_b \sin(\theta - \frac{2\pi}{3}) + i_c \sin(\theta + \frac{2\pi}{3})] \quad (5.4)$$

$$i_d = \frac{2}{3} [i_a \cos(\theta) + i_b \cos(\theta - \frac{2\pi}{3}) + i_c \cos(\theta + \frac{2\pi}{3})] \quad (5.5)$$

Para que la transformación sea a tres variables es necesario considerar una tercera componente asociada con las corrientes simétricas:

$$i_0 = \frac{1}{3}(i_a + i_b + i_c) \quad (5.6)$$

Para condiciones de equilibrio se obtiene:

$$i_a + i_b + i_c = 0 \quad (5.7)$$

Por lo que $i_0=0$.

El efecto de la transformada de Park es simplemente transformar las variables del estator de las fases a,b y c en nuevas variables en un marco de referencia que se mueva con el rotor, por lo que utiliza componentes de los ejes d y q y una tercera variable estacionaria proporcional a la corriente de secuencia cero por lo que se obtiene por definición (Capitulo 4 [21]):

$$i_{0dq} = P i_{abc} \quad (5.8)$$

Donde se definen los vectores de corriente:

$$i_{0dq} = \begin{bmatrix} i_0 \\ i_d \\ i_q \end{bmatrix} \quad i_{abc} = \begin{bmatrix} i_a \\ i_b \\ i_c \end{bmatrix} \quad (5.9)$$

Donde la transformada de Park P se define como:

$$P = \sqrt{\frac{2}{3}} \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \cos(\theta) & \cos(\theta - \frac{2\pi}{3}) & \cos(\theta + \frac{2\pi}{3}) \\ \sin(\theta) & \sin(\theta - \frac{2\pi}{3}) & \sin(\theta + \frac{2\pi}{3}) \end{bmatrix} \quad (5.10)$$

Y su transformada inversa se puede calcular:

$$P^{-1} = \sqrt{\frac{2}{3}} \begin{bmatrix} \frac{1}{\sqrt{2}} & \cos(\theta) & \sin(\theta) \\ \frac{1}{\sqrt{2}} & \cos(\theta - \frac{2\pi}{3}) & \sin(\theta - \frac{2\pi}{3}) \\ \frac{1}{\sqrt{2}} & \cos(\theta + \frac{2\pi}{3}) & \sin(\theta + \frac{2\pi}{3}) \end{bmatrix} \quad (5.11)$$

Obteniendo entonces las ecuaciones para tensión y potencia:

$$v_{0dq} = P v_{abc} \quad (5.12)$$

$$p = v_{0dq} {}^t i_{0dq} \quad (5.13)$$

Consideraciones para el control

Para el modelamiento se consideran los siguientes supuestos [22] :

1. Los enrollados trifásicos del estator son simétricos.
2. La capacitancia de los enrollados es despreciable.
3. Los enrollados distribuidos puede ser representado por uno concentrado.
4. El cambio de inductancia de los devanados del estator debido a la posición del rotor es sinusoidal y no contiene mayores armónicos.
5. Las pérdidas por histéresis son despreciables.
6. En estados transientes y subtransientes la velocidad del rotor es cercana a la síncrona.
7. Los circuitos magnéticos son lineales (no saturados) y los valores de las inductancias no dependen de la corriente.

Para el modelamiento se utiliza la notación de fasor ($\dot{}$) y se consideran los parámetros: resistencia por fase de estator R_s ; reactancia de eje directo X_d ; reactancia de eje en cuadratura X_q .

Las corrientes por los devanados ficticios \dot{I}_d e \dot{I}_q , están desfasados en 90° , y su suma vectorial corresponde a la corriente por fase de los enrollados reales:

$$\dot{I} = \dot{I}_d + \dot{I}_q \quad (5.14)$$

El voltaje por fase de los enrollados reales queda dado por:

$$\dot{E} = R_s \dot{I} + jX_d \dot{I}_d + jX_q \dot{I}_q + \dot{V} \quad (5.15)$$

Donde \dot{V} es la tensión en bornes y \dot{E} es el voltaje en vacío por fase, que depende proporcionalmente de la corriente por el rotor (I_r) y de la velocidad de giro (ω).

El generador de la central es de tipo cilíndrico por lo que $X_q=X_d=X_s$, denominándose esta última reactancia sincrónica. Dando como resultado el modelo simplificado de la figura 5.5 y la siguiente ecuación simplificada con el circuito equivalente del sistema:

$$\dot{E} = (R_s + jX_s) \dot{I} + \dot{V} \quad (5.16)$$

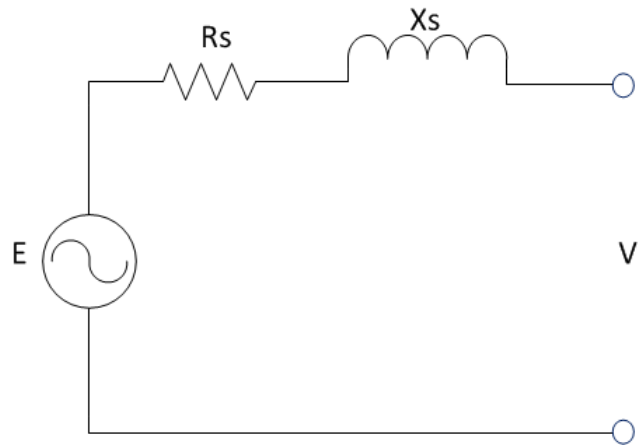


Figura 5.5: Circuito equivalente monofásico de generador síncrono.

Donde en general R_s puede despreciarse respecto a X_s .

De lo anterior se puede obtener que si aumenta \dot{E} entonces aumenta \dot{V} , por lo que para obtener un valor deseado de voltaje en bornes se deben controlar las variables que determinan \dot{E} , que como se dijo son la corriente rotórica I_r y la velocidad de giro ω . Entonces para control de este bloque del sistema se deben manipular la corriente de campo o excitación que corresponde a la rotórica y la apertura de la válvula mariposa que afecta la velocidad de giro del grupo turbina-generador. Para simular la planta MCH se consideran las recomendaciones para modelación enunciadas anteriormente y se utiliza la herramienta Simulink de MATLAB, tomando como base el desarrollo realizado por Pablo Jiménez [23].

En la figura 5.6 se presenta el modelo de la planta:

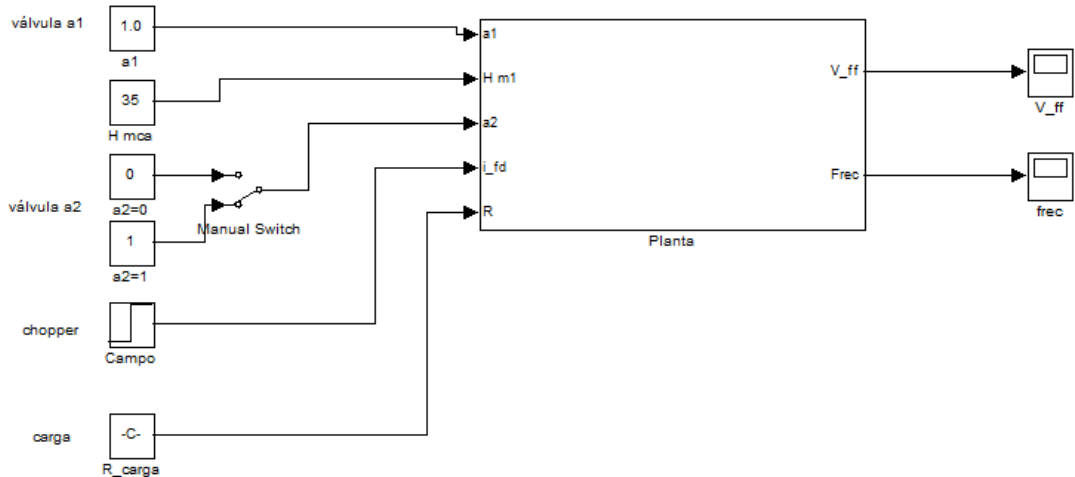


Figura 5.6: Planta MCH.

La cual en su interior posee los modelos de la válvula mariposa, de la turbina y del generador.

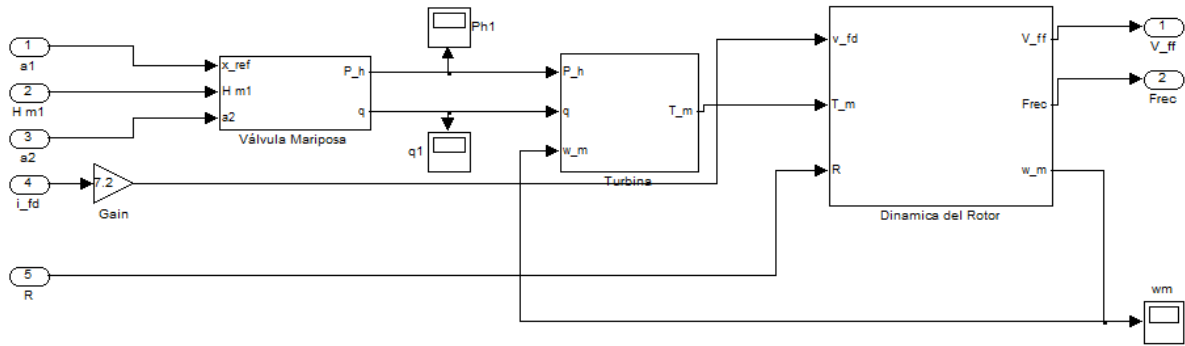


Figura 5.7: Planta MCH.

El bloque de la válvula mariposa permite simular el recurso hidráulico. Las entradas del bloque son las aperturas de las válvulas y la altura en m.c.a de agua. La válvula a1 corresponde a la controlada por el servomotor y a2 puede dejarse completamente cerrada o completamente abierta, según se desee. Las salidas son el caudal y la potencia hidráulica que son utilizados por el bloque de la turbina.

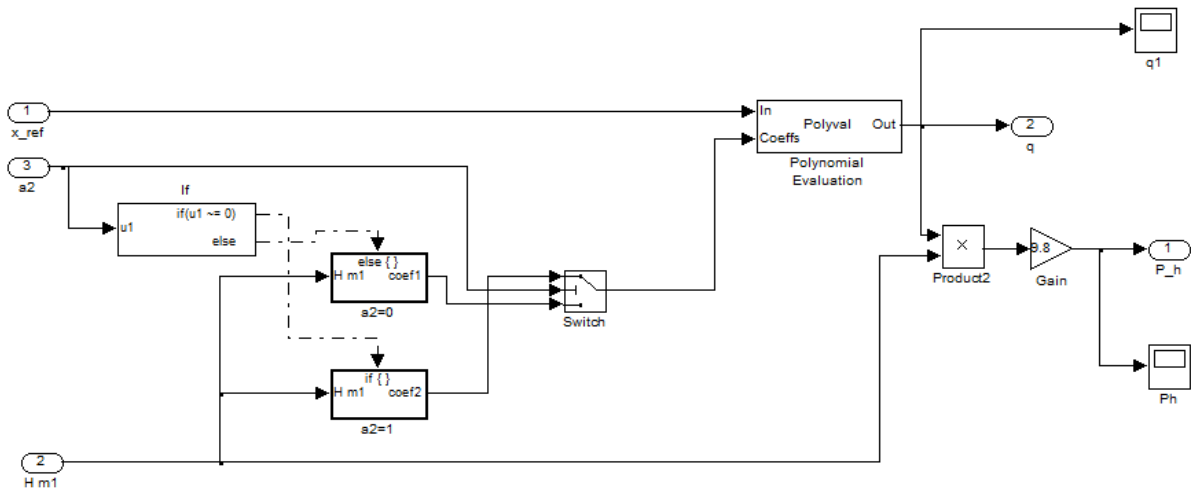


Figura 5.8: Válvula mariposa.

El bloque de la turbina tiene por entradas las provenientes de la válvula y la velocidad angular del eje del rotor. Y por salida el torque mecánico que utiliza el generador.

El bloque del generador tiene por entradas el torque mecánico, la corriente de excitación y una carga resistiva trifásica que emula la demanda eléctrica. Y por salidas tiene la tensión entre fases, la frecuencia y la velocidad angular del rotor.

Este modelo se pensó para simular la operación en isla de la planta, por lo que permite realizar control sobre la misma modificando la altura de agua como parámetro, la corriente de campo y la carga resistiva como variables manipuladas.

5.2. Simulaciones de bloques del sistema de control

Para poder efectuar las simulaciones del sistema de control de la MCH, se consideraran de forma independiente los actuadores, pudiendo de esta forma simular los resultados del control de cada subsistema.

Se plantea la utilización de controladores PID por su bajo costo y simple implementación [24]. Es necesario simular las variaciones de este tipo de control P, PI y PID, para poder seleccionar el más apropiado para los controladores de la MCH.

5.2.1. Control de corriente de campo mediante circuito chopper

En la figura 5.9 se adosa al modelo de la planta el control del actuador *chopper*.

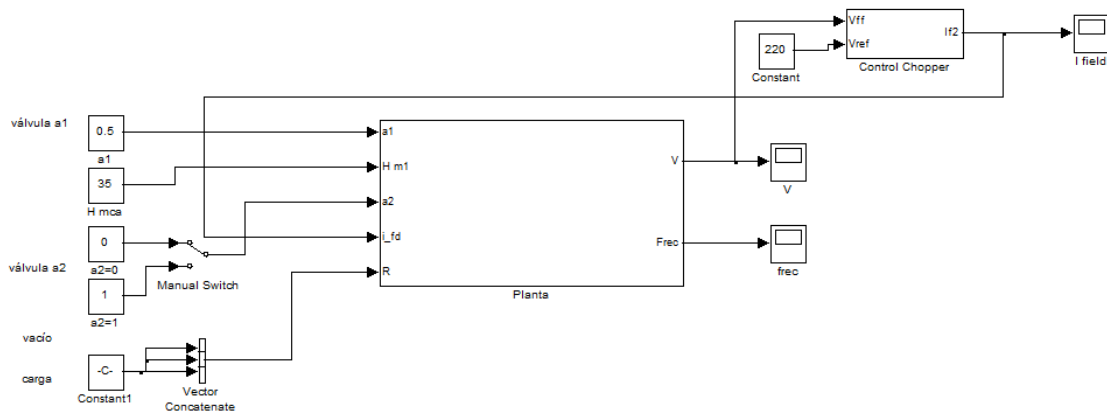


Figura 5.9: Modelo con control de campo.

En la figura 5.10 se consideran las alternativas de control P, PI y PID, las cuales son sintonizadas mediante el método de Ziegler y Nichols [25].

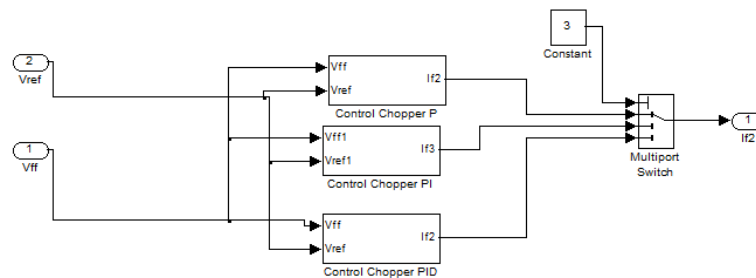


Figura 5.10: Alternativas de control.

Se realiza la simulación del control de campo en vacío, para evaluar que tipo de estrategia es mejor. Obteniendo los resultados del control P en la figura 5.11, PI en 5.12 y PID en 5.13.

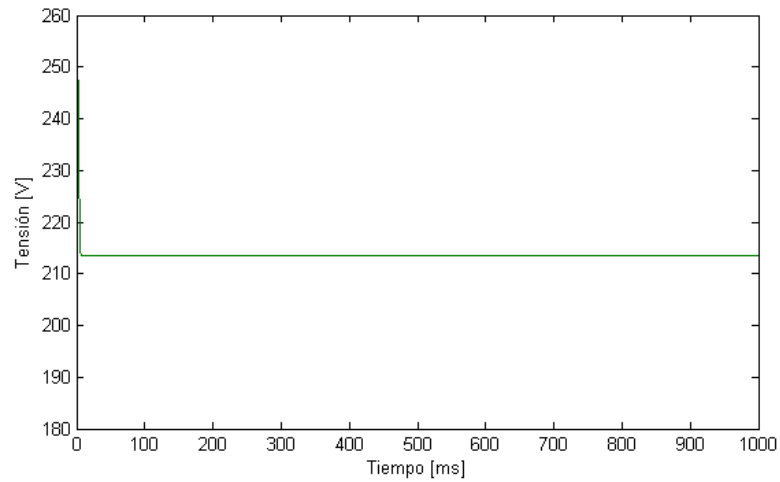


Figura 5.11: Control P de tensión.

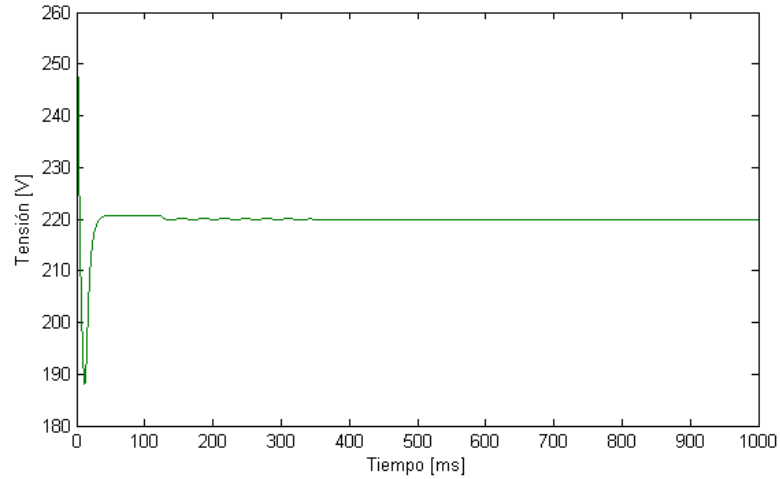


Figura 5.12: Control PI de tensión.

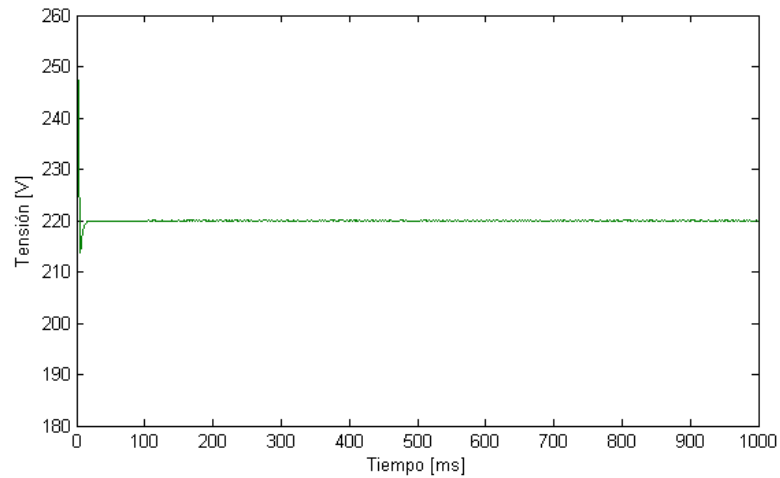


Figura 5.13: Control PID de tensión.

Se puede apreciar que el control P presenta un error estacionario; el control PI, una sobre oscilación sucedida por pequeñas oscilaciones para luego estabilizarse; el PID, muy pequeñas oscilaciones que se mantienen en el tiempo.

Por lo anterior se descarta la estrategia P y se prosigue a evaluar las estrategias PI y PID frente a un escalón de referencia de 180 [V] a 220 [V], pero esta vez se sintonizan los controladores mediante la herramienta MATLAB.

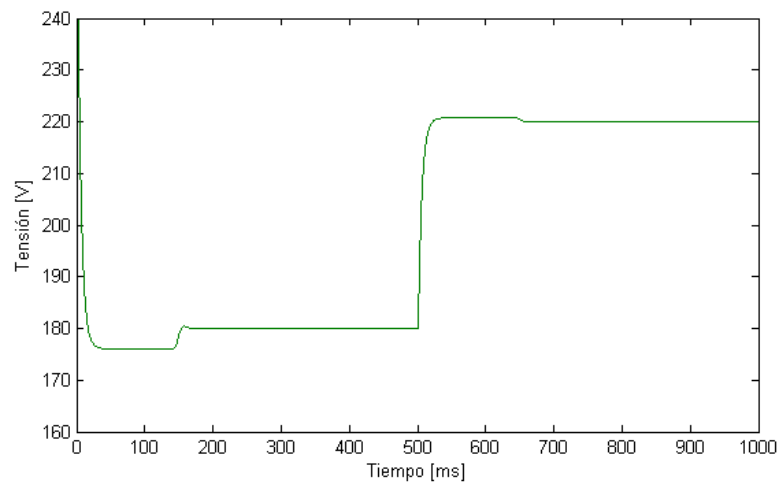


Figura 5.14: Control PI de tensión ante escalón de referencia.

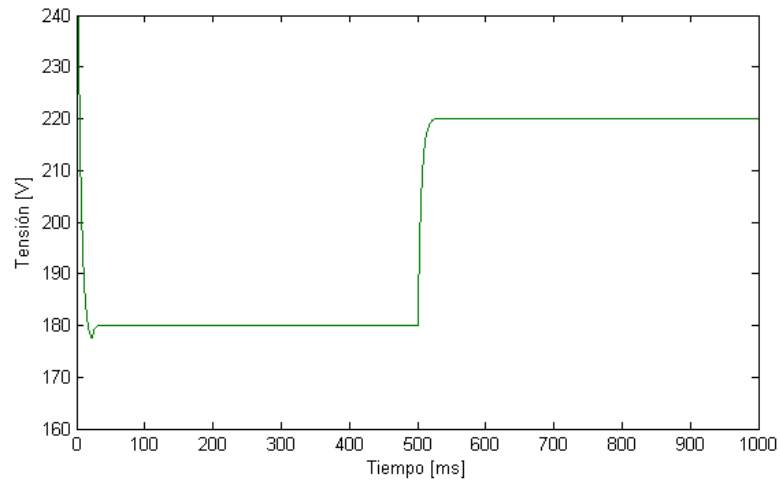


Figura 5.15: Control PID de tensión ante escalón de referencia.

Donde se aprecian nuevamente las pequeñas oscilaciones mencionadas anteriormente, pero para efectos prácticos del control no son determinantes, por lo que se simula la respuesta frente a un escalón de carga a los 500 [ms].

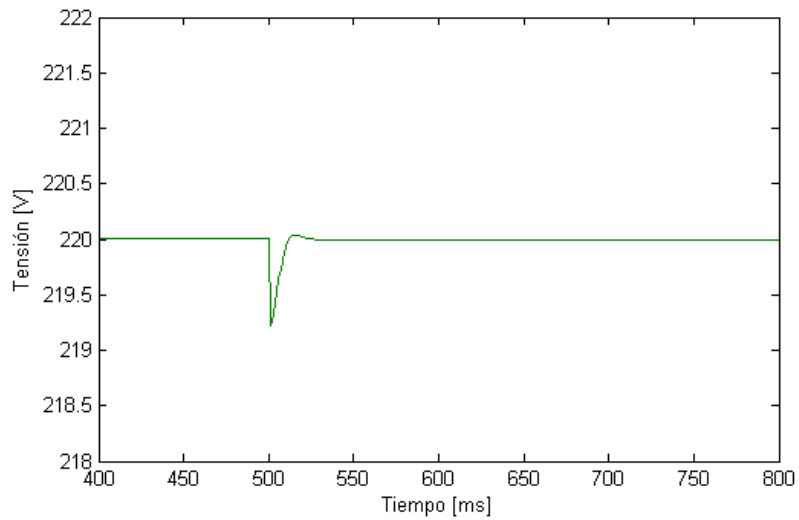


Figura 5.16: Control PI de tensión ante escalón de carga.

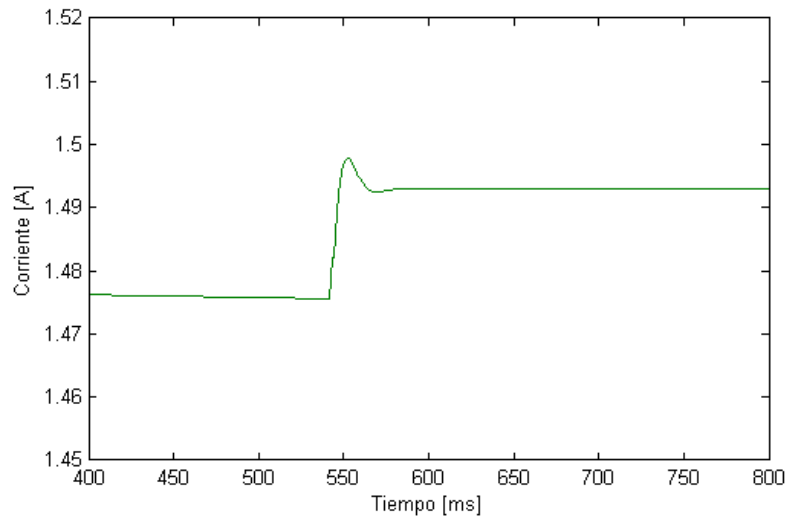


Figura 5.17: Control PI, variable manipulada $I[r]$ ante escalón de carga.

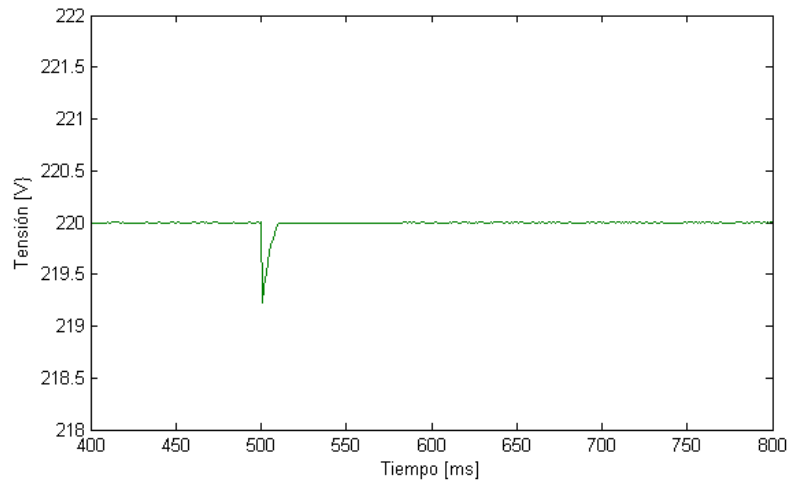


Figura 5.18: Control PID de tensión ante escalón de carga.

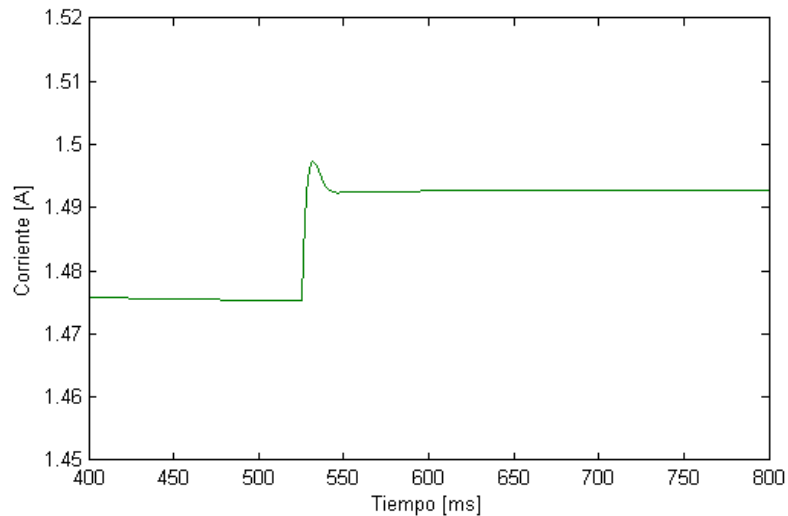


Figura 5.19: Control PID, variable manipulada $I[r]$ ante escalón de carga.

Se puede observar que para ambas alternativas, se obtiene una buena respuesta de la planta. En las figuras 5.16 y 5.18 el controlador sigue la referencia de 220 [V]. Y en las figuras 5.17 y 5.19, la variable manipulada se comporta según lo expuesto en 4.3.1. Por lo que se puede usar cualquiera de las estrategias. Siendo determinante para la elección la facilidad de programación en el DSC. Por lo anterior, se utilizará el control PI, ya que programar la fracción D utiliza un mayor tiempo de procesamiento.

Para poder implementar el control del *chopper* en el prototipo de la MCH es necesario probar el actuador, a modo de establecer los límites de operación del mismo. Esto se realiza variando el ciclo de trabajo del PWM generado por el DSC obteniendo los resultados de corriente de campo de la figura 5.20.

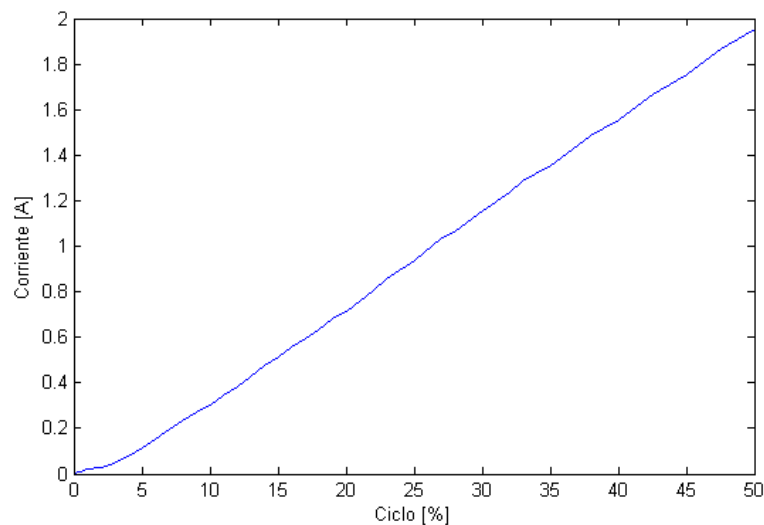


Figura 5.20: Corriente de excitación v/s ciclo de trabajo de PWM.

Con lo anterior se obtiene la siguiente relación entre corriente de campo y ciclo porcentual:

$$I_r = 0,0382ciclo[\quad (5.17)$$

Con lo cual se puede programar el DSC, respetando los límites expuestos en 4.3.1, para que realice el control de la corriente de excitación.

5.2.2. Control de frecuencia mediante resistencias de desahogo

Para simular el control mediante el actuador de resistencias de desahogo es necesario obtener la relación entre corrientes de desahogo, frecuencia del generador y ciclo de trabajo de los PWM.

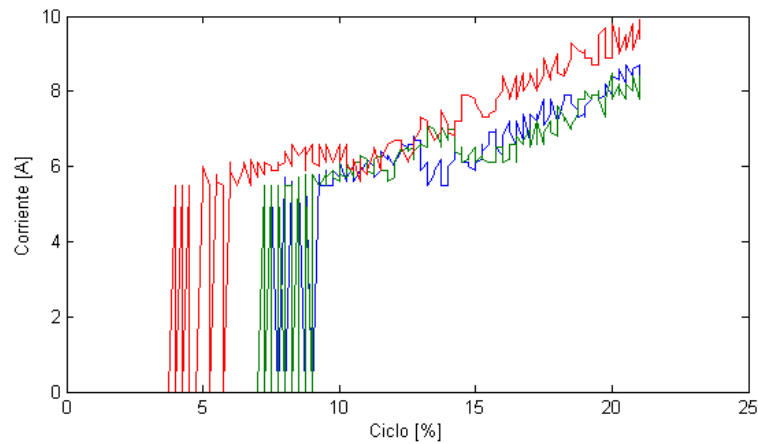


Figura 5.21: Corrientes de desahogo v/s ciclo de trabajo de PWM.

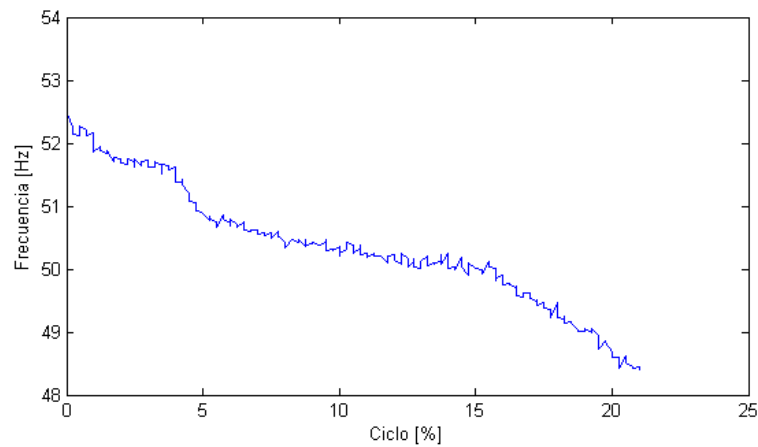


Figura 5.22: Variación de frecuencia v/s ciclo de trabajo de PWM.

Si bien la corriente de desahogo no presenta un comportamiento lineal, su efecto sobre la frecuencia puede ser linealizado, por lo que se puede utilizar un control de tipo PI.

Para su simulación, no se puede utilizar una resistencia variable conectada al generador, pues el modelo considera la carga resistiva para el cálculo de las tensiones y corrientes generadas, por lo cual se opta por utilizar un modelo de caja negra con una función polinomial, obtenida de la figura 5.22, para probar el control.

$$f = 52,603ciclo^3 - 0,0243ciclo^2 + 0,00005ciclo - 0,0000001 \quad (5.18)$$

Consiguiendo el siguiente modelo, que considera a la planta como una caja negra cuya entrada es el PWM generado por el DSC y salida la frecuencia a la cual esta generando la MCH.

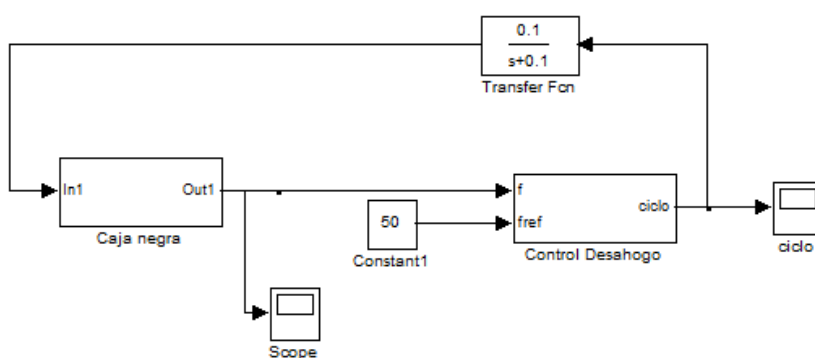


Figura 5.23: Modelo para control de frecuencia.

Con el cual se obtiene la respuesta de frecuencia sin perturbación en la fig. 5.24 y las variaciones del ciclo del PWM, como variable manipulada, en la fig. 5.25.

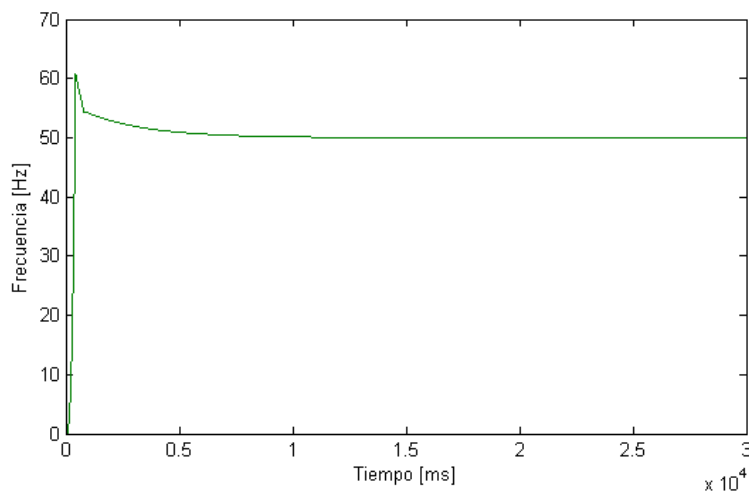


Figura 5.24: Control de frecuencia en vacío.

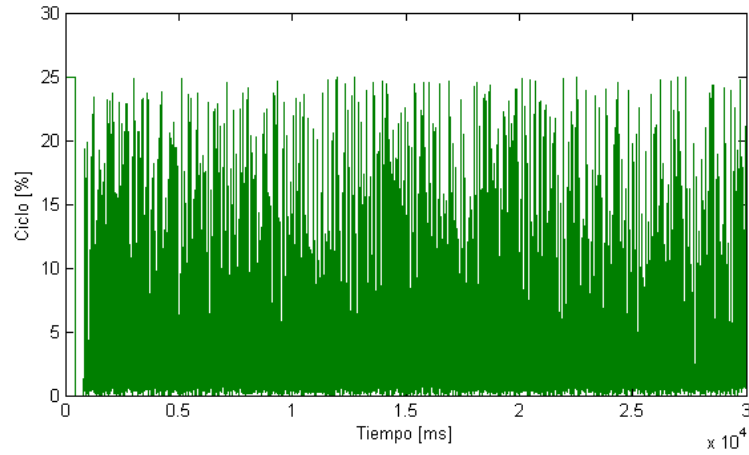


Figura 5.25: Variable manipulada, ciclo [%] en vacío.

En una segunda simulación se añade una perturbación equivalente a conexión y desconexión de carga obteniendo la respuesta al escalón en la fig. 5.26 y el comportamiento de la variable manipulada en la fig.5.27.

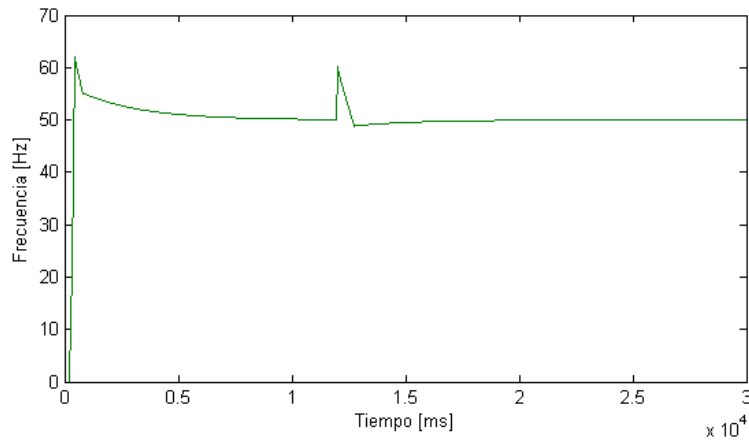


Figura 5.26: Control de frecuencia con carga.

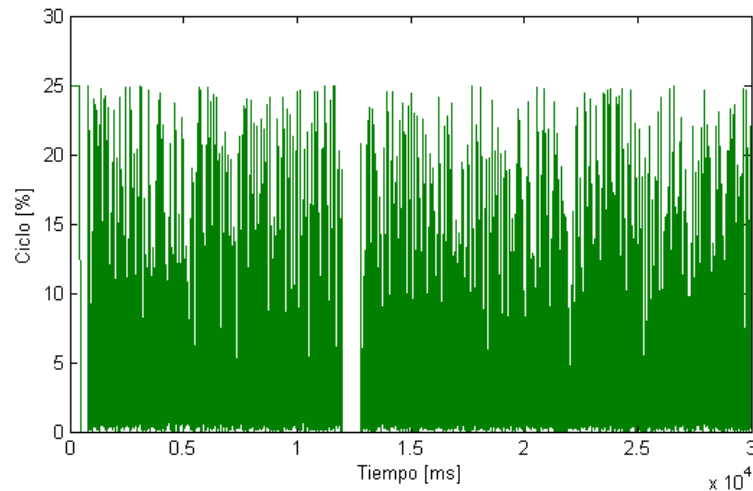


Figura 5.27: Variable manipulada, ciclo [%] ante escalón de carga.

En las figuras 5.26 y 5.27 se aprecia que al conectar una carga, el controlador reacciona llevando la variable manipulada a cero, por lo cual aumenta la frecuencia, para luego seguir la referencia de 50 [Hz].

Si bien esta simulación se realiza en base a un modelo de caja negra, a causa de la función que compone este bloque se obtuvo experimentalmente, se considera como válido para representar lo que se espera suceda en la realidad al implementar el control de frecuencia mediante las resistencias de desahogo.

5.2.3. Control de frecuencia mediante variación del recurso hídrico

Si bien se quería en un comienzo realizar control de frecuencia mediante la válvula mariposa, esta opción se ve descartada debido a la lenta respuesta de la misma y al potencial desgaste que le produciría si se utiliza con tal propósito [7]. Por ello se propone un control de frecuencia modificando el recurso hídrico, específicamente la altura de agua. Lo cual en la práctica puede realizarse sólo en condiciones de laboratorio mediante el variador de frecuencia del cual se dispone. Por lo que las siguientes simulaciones tienen un propósito experimental solamente.

Se simulan los controles conjuntos de tensión y frecuencia con el modelo que se tiene de la planta (fig. 5.28). Obteniendo los resultados de control de frecuencia en la figura 5.29 y de control de tensión en 5.31, y el comportamiento de las variables manipuladas de potencia hidráulica (fig. 5.30) y corriente de campo (fig. 5.32).

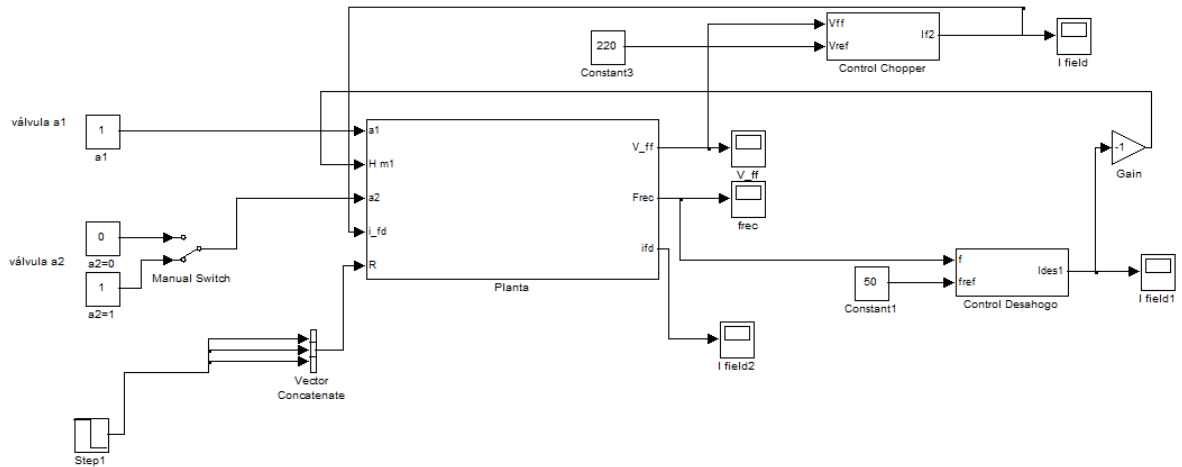


Figura 5.28: Modelo para control de frecuencia.

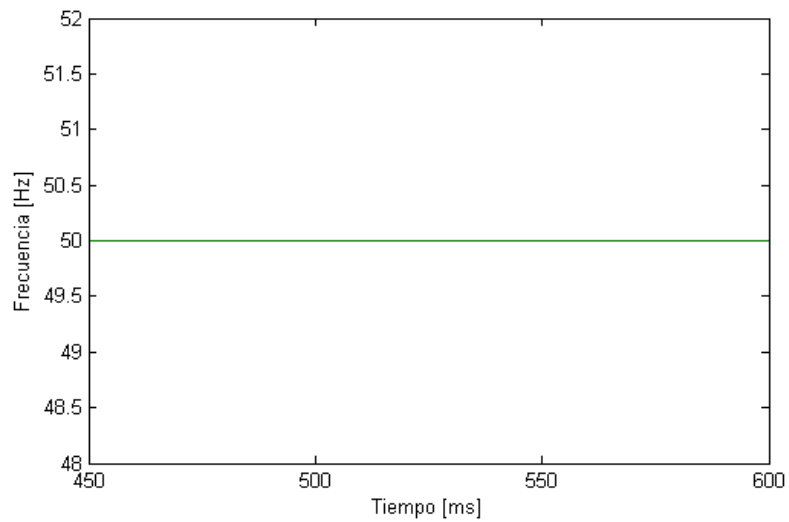


Figura 5.29: Control de frecuencia con carga.

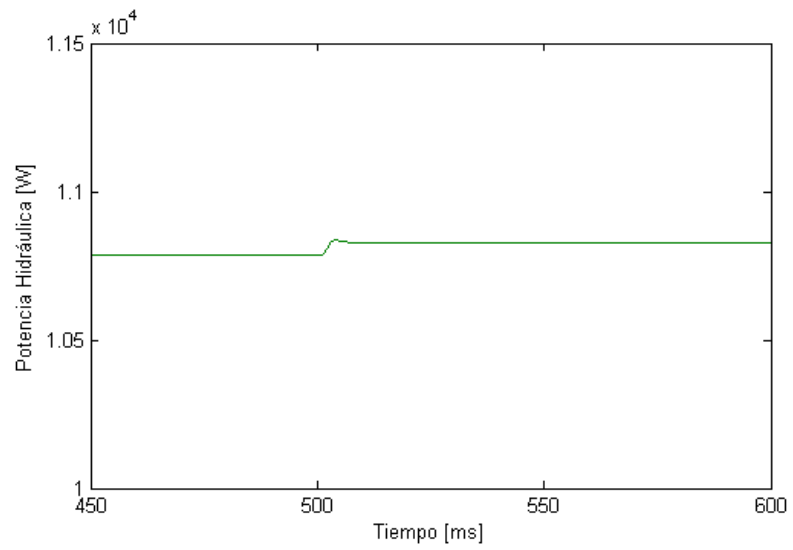


Figura 5.30: Potencia hidráulica manipulada ante escalón de carga.

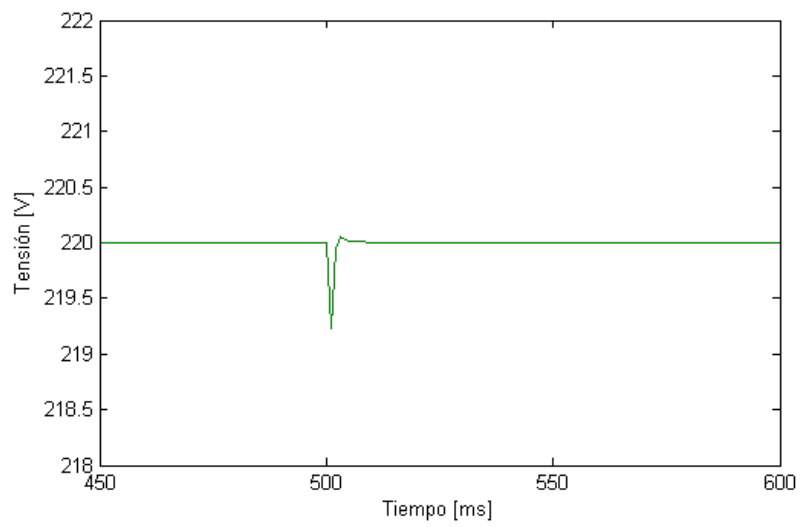


Figura 5.31: Control de tensión con carga.

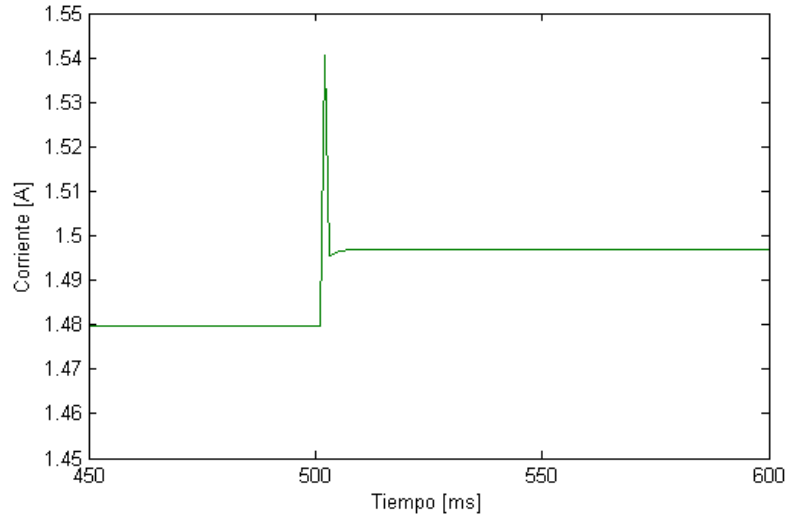


Figura 5.32: Corriente de campo manipulada ante escalón de carga.

Observándose una rápida respuesta frente al escalón de carga, del orden de $10[\text{ms}]$. Distinguiéndose que para seguir la referencia de tensión aumenta la corriente de campo (fig. 5.32) y para mantener la frecuencia, la potencia hidráulica, lo cual corresponde al comportamiento esperado en 4.3.1. y a la teoría expuesta en 5.1.

5.2.4. PLL para sincronización

Como se enunció en el capítulo anterior, para la sincronización es necesario utilizar un PLL, por lo que se realiza la simulación de éste [26].

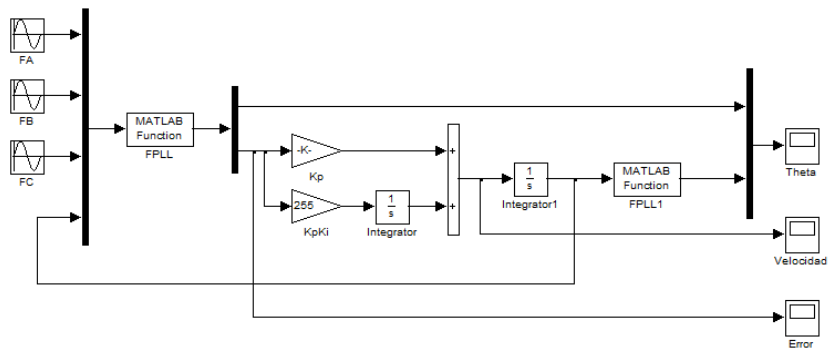


Figura 5.33: Modelo PLL para simulación.

Obteniendo la respuesta de la figura 5.34.

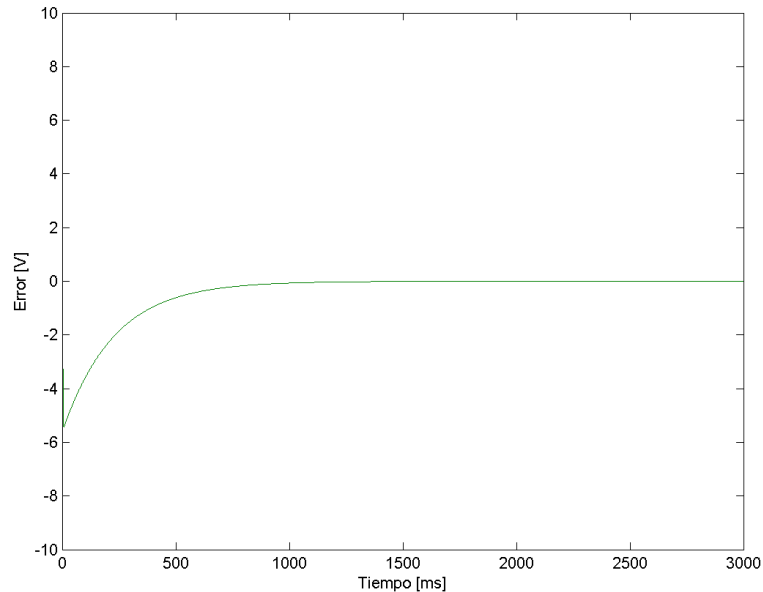


Figura 5.34: Respuesta PLL.

Donde el error corresponde a ξ de la ecuación 4.7.

Para poder simular la sincronización utilizando las funciones de las transformadas que posee MATLAB, es necesario modelar el sistema en a,b,c, pues el modelo del que se dispone es en d,q,0. Dado el tiempo y el prototipo del que se dispone, no se realizará la nueva modelación de la planta ni se implementará la estrategia de sincronización. Pues es necesario, primeramente, obtener un prototipo funcional en modo aislado, por lo que se enfoca el trabajo para tal propósito. Quedando la simulación e implementación del modo conectado a la red como propuesta para trabajo futuro.

5.3. Resultados experimentales

Las pruebas experimentales se llevan a cabo en el taller mecánico de la universidad, ubicado en la calle Molina. Utilizando las instalaciones de laboratorio establecidas para tal propósito [27].

Para generar el recurso hídrico se utiliza un convertidor o variador de frecuencia, SIEMENS-80A simovert vc, que acciona una moto bomba. El control del variador se efectúa de forma manual indicando la frecuencia a la cual se desea producir el movimiento del recurso. El que su manipulación sea manual restringe la precisión en la determinación del recurso y el tiempo que demora en su modificación.

Para observar el comportamiento de la planta durante las pruebas experimentales se utiliza un Fluke de la serie 1740, que permite almacenar datos de corrientes, tensiones y frecuencia, entre otros, en una escala de tiempo a intervalos de 1 segundo.

Se realiza el montaje del prototipo MCH en el laboratorio y se programa mediante CCS la partida en vacío de la central y el funcionamiento en isla con control de tensión mediante *chopper* y de frecuencia mediante resistencias de desahogo.

5.3.1. Resultados MCH modo espera

Primeramente se realiza una prueba en vacío con el variador de frecuencia generando el recurso hidráulico a 38 [Hz] lo que equivale a $Q=46,208$ [l/s] y a una potencia hidráulica de 9[kW] aproximadamente [7] (fig. 5.35).

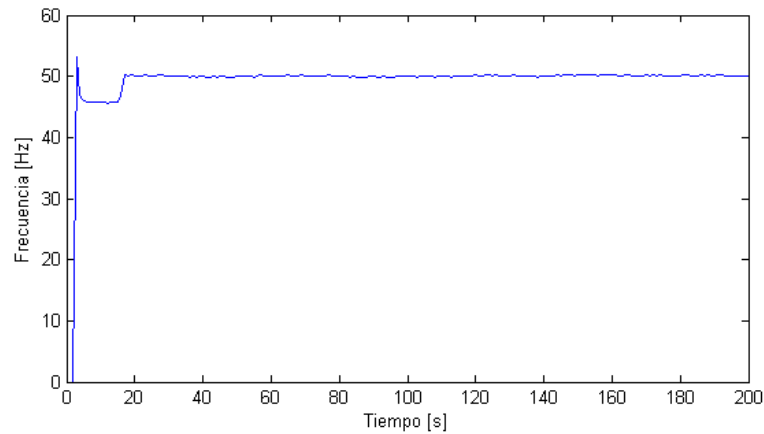


Figura 5.35: Control de frecuencia en vacío.

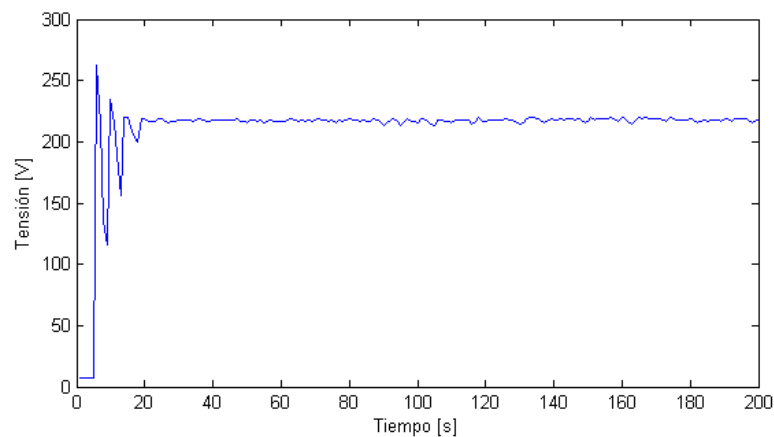


Figura 5.36: Control de tensión en vacío.

Donde se puede apreciar que los controladores cumplen su propósito, pero con una respuesta más lenta de lo esperada en comparación a las simulaciones.

Luego se realiza la misma prueba pero modificando la frecuencia del variador, de 38 a 36[Hz] y de 36 a 40[Hz].

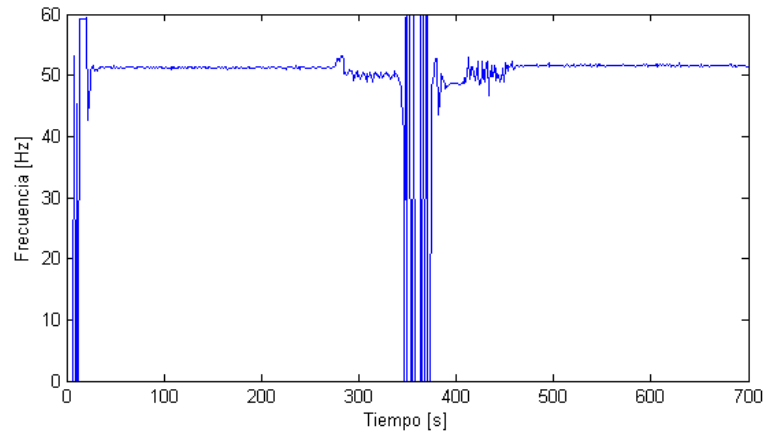


Figura 5.37: Control de frecuencia en vacío ante variaciones del recurso hídrico.

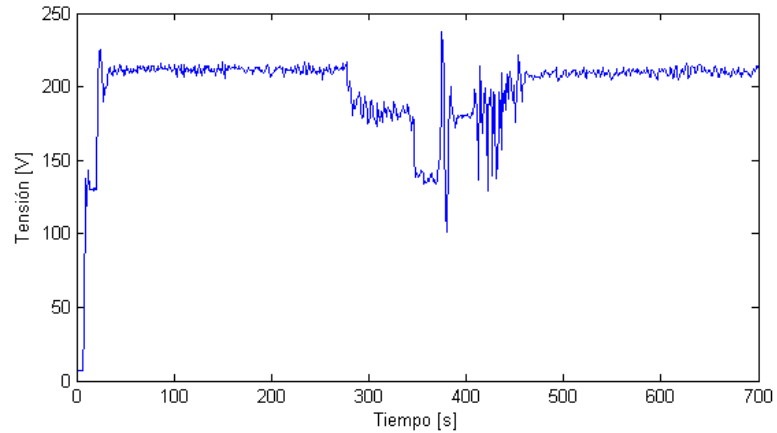


Figura 5.38: Control de tensión en vacío ante variaciones del recurso hídrico.

Al modificar la frecuencia del convertidor se puede percibir una singularidad en los gráficos, la cual se produjo al llegar a 36 [Hz]. Al notar que el sistema de control no puede reaccionar a tan bajo recurso se aumenta a 40 [Hz] permitiéndole de esta forma volver a controlar. El que no se pudiese llevar a cabo el control de la planta a los 36[Hz] se debe a que el grupo turbina-generador con configuración de dos inyectores sólo debe operar con el convertidor entre 37 y 50[Hz] [7]. Las potencias hidráulicas correspondientes a las variaciones de frecuencia con el convertidor se presentan en la figura 5.39. Donde se puede apreciar que

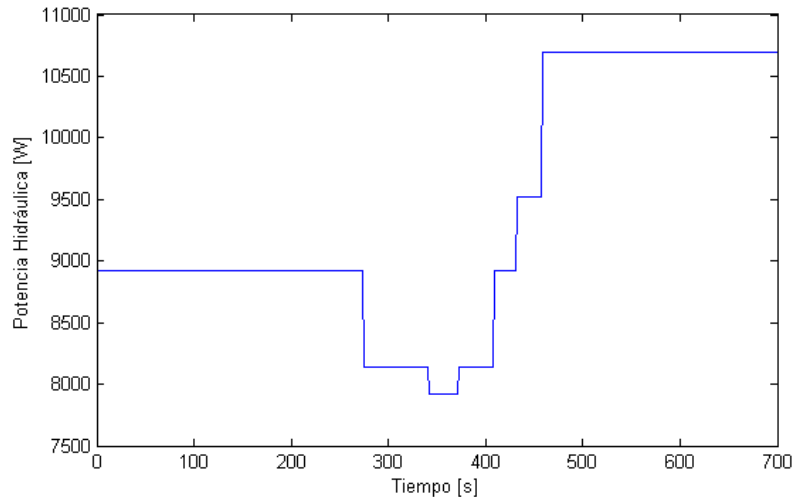


Figura 5.39: Variacion del recurso hídrico.

la menor potencia hidráulica corresponde a los 36[Hz] del variador y su ocurrencia coincide con la de la singularidad en la operación en la fig. 5.37.

5.3.2. Resultados MCH modo aislado

Obteniendo resultados aceptables en vacío se procede a conectar y desconectar carga a la MCH emulando la operación aislada.

Durante la realización de pruebas se advierte que las mediciones realizadas por el DSC no coinciden siempre con las observadas mediante el instrumento de medición Fluke. Ésto se debe a los sensores utilizados, específicamente a los transformadores, pues sus valores nominales son para transformaciones de 230 [V] a 6[V], presentan saturación e histéresis afectando en gran medida la toma de datos. Lo anterior no se percibió con las pruebas en vacío, pues se detecta al modificar carga.

Debido a que el prototipo es el único del que se dispone se procede a realizar la prueba de todas formas y se evalúa el comportamiento del control teniendo en consideración si el DSC sigue la referencia de acuerdo a la medición que éste observa.

Se realiza una prueba de conexión y desconexión de cargas de 150[W] por fase, con el variador a 38[Hz], equivalente a una potencia hidráulica de 9[kW]. Obteniendo los resultados de las figuras 5.40 y 5.41. En las gráficas se distinguen los siguientes eventos:

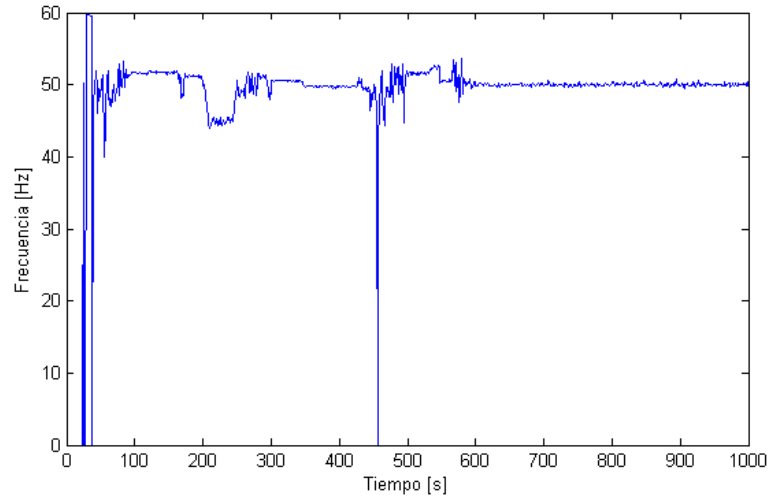


Figura 5.40: Control de frecuencia con carga.

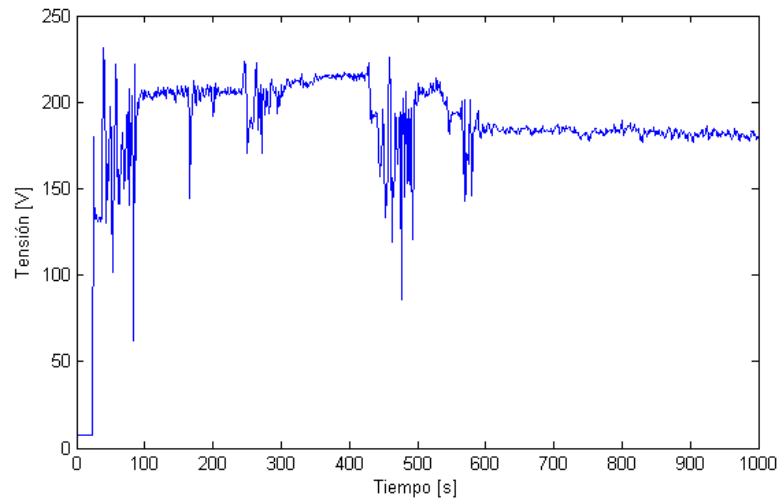


Figura 5.41: Control de tensión con carga.

- 1.- En el tiempo 180 [s] se produce una disminución de la frecuencia de generación a causa de la conexión de carga de una ampolleta de 150[W] por fase.
- 2.- En el instante 410 [s], la frecuencia disminuye abruptamente, a causa de que en la condición de operación del convertidor de frecuencia el ciclo de trabajo del PWM asociado al desahogo se encuentra al mínimo. Lo cual se produjo por la conexión de carga

adicional de tres ampolletas (450 [W]) por fase. Ante lo cual se tuvo que desconectar la carga añadida y reemplazar por dos ampolletas (300 [W]), para que el controlador pudiese actuar nuevamente.

Lo anterior puede ser observado en 5.42, donde los escalones de potencia consumida coinciden con los tiempos de ocurrencia de los eventos mencionados. Se realiza una segunda

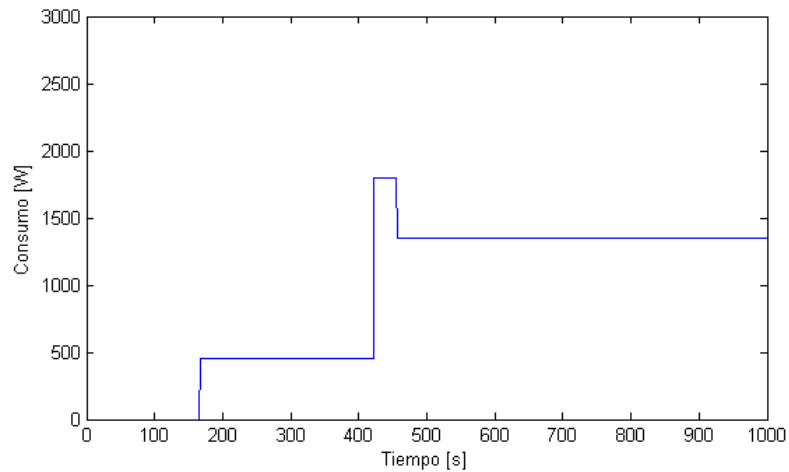


Figura 5.42: Variaciones de carga trifásica.

prueba, con el variador a 40[Hz], equivalente a 10,7[kW] de potencia hidráulica, realizando conexiones sucesivas de cargas de 150[W] por fase.

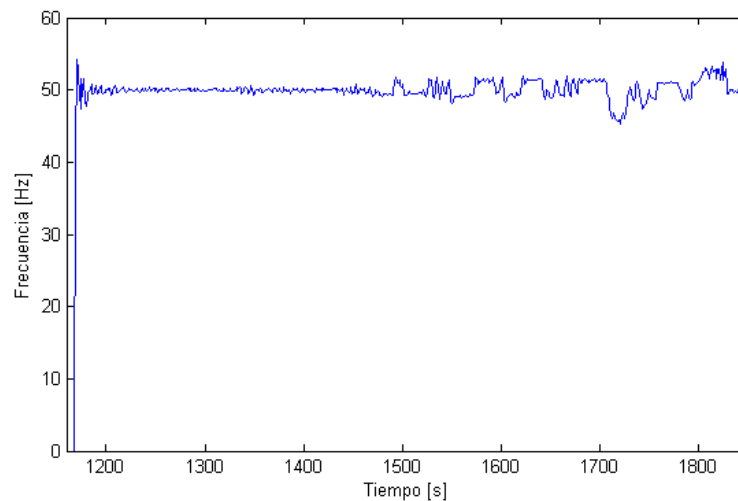


Figura 5.43: Control de frecuencia con carga.

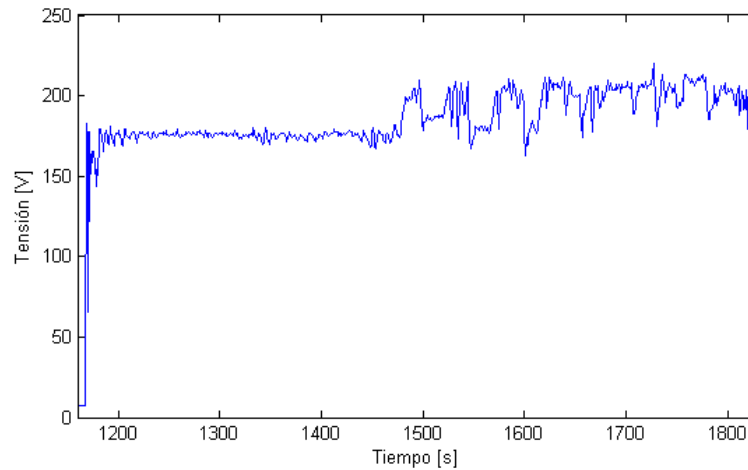


Figura 5.44: Control de tensión con carga.

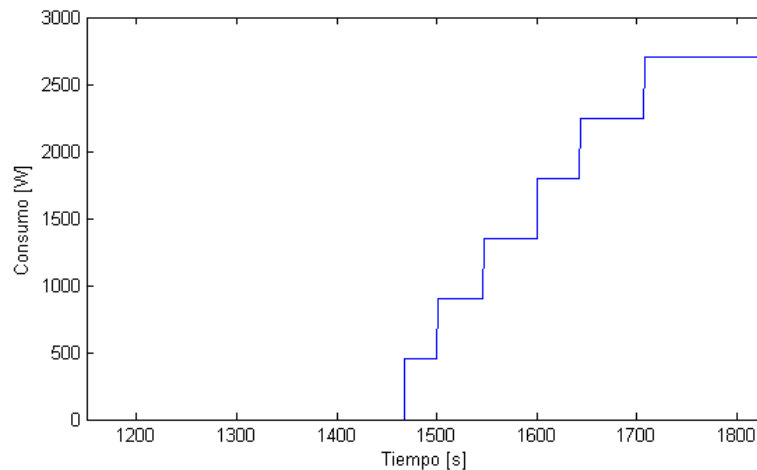


Figura 5.45: Variaciones de carga trifásica.

Donde se distingue en la sección de tiempo 1500 a 1800 [s] las respuestas ante conexiones de cargas de una a seis ampolletas (150[W] c/u) como se puede apreciar en la fig. 5.45. Consiguiendo la alimentación de un consumo máximo de 2,7 [kW] trifásico.

Si bien las mediciones del instrumento Fluke no reflejan un seguimiento por parte del controlador de tensión de las referencia de 220[V], sino mas bien pareciera que fuese de 180[V]. El DSC en sus mediciones de tensión del generador, observables mediante CCS, indica un correcto seguimiento de la referencia, por lo que desde el punto de vista del DSC, se puede considerar válida la prueba realizada.

La diferencia de mediciones entre el instrumento Fluke y DSC se debe al problema antes mencionado asociado a los transductores.

5.4. Discusión de Resultados

Se procede a comparar los resultados teóricos con los experimentales obtenidos.

Las mediciones implementadas finalmente en el prototipo fueron mediante cruces por cero para la estimación de la frecuencia y obtención de máximo para la tensión. Esta selección se debe a los problemas presentados por los transductores.

Experimentalmente se notó que los transductores requieren una hora de operación para alcanzar la temperatura a la cual las mediciones se estabilizan. Y aún pasado este tiempo presentan problemas en su comportamiento, pues al momento de sensar variaciones de tensión en un sentido de aumento, entrega valores menores y en el sentido de disminución, entrega valores por sobre los reales. Un ejemplo de ésto se puede apreciar en la prueba de modo isla donde los sensores indicaban una tensión de 220 [V], siendo que en la realidad se tenían 180 [V].

El tipo de control utilizado fue PI a causa de ser un controlador de fácil programación mediante CCS. Lo cual permite a futuro, dadas las características de un controlador PI, migrar el control digital a análogo que es de bajo costo y simple implementación .

5.4.1. Control de tensión en modo de espera

En la fig.5.12, se obtiene la simulación del control en vacío de la tensión y en la fig. 5.36, la respuesta obtenida en la práctica.

Se puede apreciar que los tiempos de respuesta difieren en gran medida, de 0,01 [s] en la simulación a 20 [s] con el prototipo. Esta enorme brecha se debe principalmente a que los transductores no son los indicados para esta aplicación, por lo que el tiempo de cálculo de variables de tensión y frecuencia, limita el tiempo de respuesta de los controladores. Ésto mismo influye al momento de sintonizar los parámetros PI, pues los sintonizados mediante MATLAB difieren en gran manera de las utilizadas para la implementación, dadas las limitaciones de *hardware*.

Otra diferencia a destacar son las sobre-oscilaciones y pequeñas oscilaciones posteriores. En la fig. 5.12 la sobre-oscilación es menor a los 10 [V], mientras que en la fig. 5.36, es de hasta 50[V]. Respecto a las pequeñas variaciones posteriores, en la fig.5.36, estas tienen su explicación en la forma de cálculo de mediciones y su dependencia del transductor de tensión. Pues la tensión se calcula según el método de valor máximo expuesto en 4.1.1, por lo que considera el valor instantáneo entregado por el transductor para estimación del máximo, y si este presenta variaciones dados los problemas del sensor, éstas se ven reflejadas en el resultado rms obtenido en base al cual se toma la decisión de control.

Obviando el tiempo de respuesta, en régimen permanente el controlador diseñado logra

cumplir con su objetivo de llevar y mantener a la planta en torno a la referencia de 220 [V] en el estado de operación en espera, validando de esta forma la estrategia propuesta.

5.4.2. Control de frecuencia en modo de espera

En la fig. 5.24, se obtiene la simulación del control en vacío de la frecuencia y en la fig. 5.35, la respuesta obtenida en la práctica.

De forma similar al control de tensión, existe disimilitud en los tiempos de respuesta. En la fig. 5.24 demora 5 [s] en alcanzar el régimen permanente mientras que en la fig. 5.35, 20 [s]. La diferencia no es de la misma proporción que en el caso anterior a causa de que para la simulación mediante caja negra se utilizaron los datos obtenidos experimentalmente de la operación del actuador de desahogo.

Respecto a la sobreoscilación percibida, en la fig. 5.35 es menor a la mitad de la sobreoscilación obtenida en la fig. 5.24, lo cual se explica a causa de que el modelo de caja negra se obtiene mediante la obtención de línea de tendencia polinomial de grado tres, por lo que incurre en aproximaciones al momento de realizar la simulación.

En régimen permanente tanto 5.24 como 5.35 permiten llevar a la planta a la operación deseada de 50[Hz].

5.4.3. Control en operación en isla

Para realizar la prueba con carga en operación aislada, se realiza la conexión de una carga resistiva de hasta cuatro ampolletas de 150[W] c/u por fase, pues los transformadores que miden tensión tienen un rango muy acotado de operación.

En la fig. 5.31, se obtiene la simulación del control de tensión con carga y en las figuras 5.41 y 5.44, los resultados obtenidos en la práctica.

Por los problemas antes mencionados, nuevamente, se distinguen notorias diferencias en tiempos de respuesta y sobre-oscilaciones. A lo cual se suma que al momento de realizar la prueba experimental, los transductores se encontraban saturados presentando el problema antes mencionado de sensar un valor por sobre el real, lo cual fue comprobado durante la realización de pruebas. Ésto se tradujo en que el DSC observara en sus mediciones 220 [V] mientras que el instrumento Fluke indicaba un valor menor a 200 [V], por causa del error introducido por los mismos transductores.

Desde el punto de vista del DSC la estrategia de control se ve validada, pero desde el punto de vista de la planta no se logra llevar al punto de operación deseado.

En la fig. 5.29, se obtiene la simulación del control de frecuencia con carga y en las figuras 5.40 y 5.43, los resultados obtenidos en la práctica.

En forma reiterada se presentan diferencias en el tiempo de respuesta y sobre-oscilaciones. Pero pese al error introducido por los transductores, el control de frecuencia se puede realizar en forma satisfactoria. Ésto es gracias a la forma en que se estima la medición, pues al no considerar la tensión, sino, sólo su cruce por cero, el resultado obtenido presenta casi un nulo error respecto a la realidad medida mediante el instrumento fluke. Lo que permite validar la estrategia de control de frecuencia.

Capítulo 6

Conclusiones

Al comienzo de este trabajo de título, se tenían grandes ambiciones, diseñar e implementar un sistema de control electrónico para una MCH inteligente que opere de forma automática. Para llevar a cabo esto era necesario primeramente diseñar e implementar un prototipo de MCH para el cual pensar el sistema de control.

Se dedicó gran parte del tiempo al desarrollo del prototipo, pues este incluía los actuadores a utilizar para el control de la planta. También se dedicó muchas horas hombre en la familiarización con la arquitectura del DSC y el lenguaje de programación necesario para su utilización. Y recién en los últimos meses se pudo trabajar en la obtención de datos y operación de los mismos con miras a controlar la planta.

Todo el tiempo dedicado, dio frutos, no en la totalidad de lo esperado, pero si en una parte satisfactoria. Se pudo implementar el control de la MCH en operación modo espera e isla, obteniendo resultados aceptables, dentro de un rango de operación restringido, que permiten validar las estrategias dentro de lo posible y disponible, pues es necesario mejorar el *hardware* que afecta las mediciones y los tiempos de respuesta.

Se pudo apreciar que un control de tipo PI es suficiente para el control de la planta, pues las variables a controlar tienen un comportamiento que puede ser linealizado si se consideran las simplificaciones utilizadas en este trabajo.

Se genera la electrónica necesaria para la implementación del prototipo detallada en el anexo B. Dentro de la cual se logra implementar la circuitería para la utilización de los actuadores *chopper* y resistencias de desahogo. Y corroborar tanto en simulaciones como experimentalmente el comportamiento de las variables manipuladas con miras a controlar tensión y frecuencia generadas.

Se obtiene un prototipo funcional para modo en isla, pero no se alcanza a aprovechar la potencialidad del generador debido a las restricciones de *hardware*. Para la operación se obtuvo un máximo de 2,7 [kW] equivalente casi a un cuarto de los 10[kW] nominales de la máquina. Por lo que aún queda trabajo por realizar para cumplir con las expectativas del proyecto.

6.1. Trabajo futuro y recomendaciones

Actualmente la electrónica de la planta requiere al menos una hora para alcanzar la temperatura de operación a la cual se estabilizan las mediciones por parte de los transductores, lo cual se pudo apreciar durante la calibración de las mediciones digitales y la realización de pruebas. Éste es un tiempo considerable para un producto de este tipo, por lo que como trabajo futuro se propone reemplazar los sensores utilizados por transductores que entreguen directamente el valor efectivo de la medición mediante señal de corriente al DSC, para evitar de esta forma pérdidas en la transmisión de señales y liberar al DSC del cálculo de los valores RMS. Los transductores que presentan una necesidad inmediata de cambio son los de tensión y el de corriente del *chopper*, pues no permiten una obtención de datos confiable.

Una vez que sea mejorado el sistema de mediciones, es posible implementar la estrategia de sincronización y migrar de la forma en que se estiman los datos mediante operaciones simples a la utilización de transformadas, lo que facilitaría el procesamiento de información para la sincronización.

Como se buscaba una implementación simple del prototipo, al momento de diseñar la electrónica no se tuvo especial cuidado en la disposición de los componentes ni en el dimensionamiento ni ubicación final de las placas, por lo que se recomienda re-diseñar la electrónica utilizando componentes superficiales considerando los espacios que se disponen y los distintos tipos de señales que deben ser transmitidas.

De los actuadores disponibles en la MCH, sólo dos pudieron ser ocupados: el *chopper* y las resistencias de desahogo. Pues la válvula no es la indicada para efectuar control de frecuencia, que era lo que se esperaba que realizase en un comienzo. Por esto se propone cambiar la válvula por una de tipo aguja que permite un control más rápido y presenta mayor vida útil que la mariposa. Respecto a las resistencias de desahogo, a causa de que utilizan triacs para el control de las mismas, aportan componentes armónicas a la salida de la planta, por lo que se sugiere la utilización de electrónica de potencia para eliminar los aportes de armónicos o bien para reemplazar el actuador. Pues si se utiliza para equilibrar el consumo de hasta 10[kW], siempre se encontraría disipando energía, lo cual no es lo deseado cuando se piensa en un sistema de generación que busca ser eficiente y óptimo en su propósito de generación.

Si bien la malla curricular entrega una muy buena base general en ciencias de la ingeniería, para el desarrollo de un tema de este tipo es necesaria la especialización en materias de circuitos electrónicos y programación de software orientado a *hardware*. Pues a causa de la inexperiencia en estos temas es que gran parte del trabajo se realizó en base a ensayo y error, muestra de ello son las modificaciones en el momento de las placas y la gran cantidad de versiones del código para utilización del DSC. Por lo que se recomienda a quien continúe este proyecto que se capacite previamente en lo mencionado.

Si se juzga el prototipo obtenido, éste no podría ser considerado un producto final, pues no se consideró el cumplimiento de estándares en ningún nivel del diseño ni implementación, por lo que se recomienda revisar las normas existentes a nivel internacional para la implementación del *hardware* y las restricciones en cuanto a los límites para la operación, para poder re-diseñar la MCH.

Bibliografía

- [1] BUN-CA. Biomass Users Network. “Manuales sobre energía renovable: Hidráulica a pequeña escala.”, 1 ed.-San José, C.R.:(BUN-CA), 2002.
- [2] ANTEZANA J., MARÍN J., P., WEBER P. Informe de diseño ‘Central Micro Hidráulica inteligente *plug & play*.’, *Centro de Energía, Universidad de Chile, 2011.*
- [3] DIARIO OFICIAL DE LA REPÚBLICA DE CHILE. N° 39.025 “Procedimiento de Determinación de Potencia Firme y Balances de Potencia bajo Ley 19.940”. Informe Final, *Martes 1 de Abril de 2008.*
- [4] ALARCÓN I., JIMÉNEZ G.,PALMA R. Proyecto Energías Renovables No Convencionales (CNE/GTZ) “Las Energías Renovables No Convencionales en el Mercado Chileno.”, *publicación CNE/GTZ, Marzo 2009.*
- [5] JIMÉNEZ P. “Manual de Central Micro-hidráulica China”, *Centro de Energía, Universidad de Chile, Diciembre de 2010.*
- [6] ESHA (European Small Hydropower Association). “Guía para el desarrollo de una pequeña central hidroeléctrica”, *publicación ESHA, 2006.*
- [7] HAUSER K. “Caracterización de turbina hidráulica tipo turgo para microgeneración”, *memoria para optar al título de ingeniero civil mecánico, Universidad de Chile, 2012.*
- [8] ANTEZANA J., DHARMAWIDJAJA J., MARÍN J., RAMÍREZ P., WEBER P. “Microhidráulica inteligente”, *Centro de Energía, Universidad de Chile, Febrero 2012.*
- [9] MARTÍNEZ M. “Guía para la determinación de límites de operación, curva de capacidad, pruebas y mantenimiento predictivo de generadores síncronos”, *trabajo de graduación, Facultad de Ingeniería, Universidad San Carlos de Guatemala, Octubre 2003.*
- [10] TEXAS INSTRUMENTS. “TMS320F28335, TMS320F28334, TMS320F28332, TMS320F28235, TMS320F28234, TMS320F28232 Digital Signal Controllers (DSCs), Data Manual. Literature Number: SPRS439K”, *publicación Texas Instruments Incorporated, Junio 2007, revisión Abril 2012.*
- [11] TEXAS INSTRUMENTS. “Quick Start Guide TMS320C2000? Experimenter Kit Overview. Literature Number: SPRUFR5E”, *publicación Texas Instruments Incorporated, Junio 2008, revisión Febrero 2011.*

- [12] TEXAS INSTRUMENTS. “Code Composer User’s Guide. Literature Number: SPRU296”, *publicación Texas Instruments Incorporated, Febrero 1999.*
- [13] RODRÍGUEZ P. “Aportes a los acondicionadores activos de corriente en derivación para redes trifásicas de cuatro hilos”, *Tesis doctoral, Universidad Politécnica de Catalunya, Enero 2005.*
- [14] GARCÍA J. “Procesamiento Digital de Señales”, *Serie de libros electrónicos, Instituto Politécnico Nacional, www.tijbc.com/pds/libros/libro-pds.htm 2005.*
- [15] STRANG W. M., CHAIRMAN, MOZINA C., VICE CHAIRMAN. IEEE POWER SYSTEM RELAYING COMMITTEE REPORT. “Generator synchronizing, industry survey results”, *IEEE Transactions on Power Delivery, 1(1):174-183, Enero 1996.*
- [16] MURTY P.S.R. “Operation and Control in Power Systems, Stability and Control”, *Publicaciones BS, 2008*
- [17] MOHR R. “Inserción de generadores de energía renovable en redes de distribución”, *tesis para optar al grado de magíster en ciencias de la ingeniería, Pontificia Universidad Católica de Chile, Agosto 2007*
- [18] DE CAMARGO R., PEREIRA A., PINHEIRO H., GEPOC UFSM. “New Synchronization Method for Three-Phase Three-Wire PWM Converters under Unbalance and Harmonics in the Grid Voltages”, *conferencia PESC 2005, publicación 36 IEEE, paginas 506-512.*
- [19] KUNDUR P. “Power System Stability and Control”, *McGraw-Hill, Inc., 1994.*
- [20] KIAMER P. “Power Generation Handbook, Selection, Applications, Operation and Maintenance”, *McGraw-Hill, Inc., Agosto 28 2002.*
- [21] ANDERSON P. M., FOUAD A.A. “Power System Control and Stability”, *2 ed. Institute of Electrical and Electronics Engineers, Inc., IEEE Press Power Engineering Series, 2003*
- [22] BIALEK J., BUMBY J., MACHOWSKI J. “Power System Dynamics, Stability and Control”, *2 ed. John Wiley & Sons, Ltd., 2008*
- [23] JIMÉNEZ P. “Modelamiento y Control de una Central Microhidráulica”, *memoria para optar al título de ingeniero civil electricista, Universidad de Chile, Enero 2012.*
- [24] OGATA K. “Ingeniería de Control Moderna”, *3 ed. Pearson, Prentice Hall, 1998.*
- [25] ALFARO V. “Métodos de sintonización de controladores PID que operan como reguladores”, *publicación Ingeniería 12 (1,2): 21-36, Universidad de Costa Rica, 2002*
- [26] DIAZ M., JARA P. “Diseño, simulación e implementación de un prototipo de sistema de conversión eólica para estudios de comportamientos ante fallas en la red eléctrica”, *trabajo de tesis para obtener el título de ingeniero civil en electricidad y el grado de magister en ciencias de la ingeniería, Universidad Santiago de Chile, 2011.*
- [27] REYES L., HAUSER K., Centro de Energía, Universidad de Chile, FCFM. “Plug & Play Micro Hydro Laboratory Description” *Centro de Energía, Universidad de Chile, 2011.*

Apéndice A

Programación DSC

A.1. Programa principal

```
1  /*
2  *  MicrohidraulicaIsla.c
3  *
4  *   Created on: 09-07-2012
5  *   Author: Aniluap
6  */
7  #include "DSP28x_Project.h"
8  #include "math.h"
9  #include "time.h"
10 //-----
11 // Variables globales del programa
12 //-----
13 // Constantes
14 #define PI 3.1415927410125732
15 #define DOS_PI 6.2831853071795865
16 #define UNO_DIV_DOS_PI 0.1591549430918953
17 #define RAIZ_DE_DOS 1.4142135623730950
18 #define UNO_DIV_RAIZ_DE_DOS 0.7071067811865475
19 #define CPU_FREQ 150E6
20 // La frecuencia del reloj. Esto no configura la frecuencia sino que pasa el dato para que se use en el
    programa
21 #define FALSE 0
22 #define TRUE 1
23 // PWM interrupcion ADC
24 #define EPWM_TIMER_TBPRD 17858 // Indica hasta dónde cuenta el PWM en ciclos de TBCLK, cuenta para
    arriba y para abajo, luego el período es el doble en ciclos TBCLK
25 #define EPWM_TIMER_TBPRD_FLOAT 17858.0 // Igual al anterior sólo que como punto flotante para algunos
    cálculos
26 // Configuración ADC a 37.5 MHz, por lo que puede estar afectando la comunicación serial que se determina
    con el LSPCLK
27 #define ADC_MODCLK 0x2 // HSPCLK = SYSCLKOUT/(2*ADC_MODCLK2) = 150/(2*2) = 37.5 MHz
28 // Valores nominales de tensión y frecuencia
29 #define W_NOMINAL 2*PI*50
30 #define V_NOMINAL_FN 220.0 // A lo largo del programa las tensiones están expresadas siempre en valor
    fase-neutro
31 // Constantes de tiempo del programa
32 #define PERIODODEFINE ((EPWM_TIMER_TBPRD_FLOAT)/((float)CPU_FREQ)) // El período entre dos muestreos [
    segundos]
33 // para control en código de andrés
34 #define GANANCIAINTEGRALW 1000.0*PERIODODEFINE // Ganancia del PLL que determina la frecuencia
35 #define W_MAX_SATURACION 377.0 // 2*PI*60 es el límite que se puede medir
36 #define W_MIN_SATURACION 251.3 // 2*PI*40 es el límite que se puede medir
37 // Constantes de Calibración del ADC. Hay q establecerla para cada medición
38 #define FACTORCORRIENTES 0.047450584437 // (105.6*3.0)/4096.0;
39
40 // Funciones que se cargan en RAM. Corren más rápido que las otras
41 #pragma CODE_SECTION(epwm5_isr, "ramfuncs");
42 #pragma CODE_SECTION(epwm6_isr, "ramfuncs");
43 #pragma CODE_SECTION(adc_isr, "ramfuncs");
44 #pragma CODE_SECTION(Filtrar5Hz, "ramfuncs");
45 #pragma CODE_SECTION(Filtrar100Hz, "ramfuncs");
46
47 //-----
48 // Definición de estructuras
49 //-----
50 typedef struct ResultadosADC // Define la estructura llamada ResultadosADC, y los elementos que la componen
```

```

51 {
52     float VGenerador [3];
53     float IGenerador [3];
54     float VRed [3];
55     float IRed [3];
56     float IDesahogo [3];
57     float ICampo;
58 }ResultadosADC;
59 typedef struct ResultadosClarkefull// Define la estructura llamada ResultadosADC, y los elementos que la
    componen
60 {
61     float VGenerador [3];
62     float IGenerador [3];
63     float VRed [3];
64     float IRed [3];
65     float IDesahogo [3];
66 }ResultadosClarkefull;
67
68 typedef struct ResultadosParkfull// Define la estructura llamada ResultadosADC, y los elementos que la
    componen
69 {
70     float VGenerador [3];
71     float IGenerador [3];
72     float VRed [3];
73     float IRed [3];
74     float IDesahogo [3];
75 }ResultadosParkfull;
76 typedef struct Filtrocontinuo
77 {
78     float Salida;
79 }Filtrocontinuo;
80
81 typedef struct Filtrovector
82 {
83     float Salida [3];
84 }Filtrovector;
85
86 //-----
87 //estructuras utilizadas
88 //-----
89 ResultadosADC UltimoADC;
90 ResultadosClarkefull Clarkealfabetagamma;
91 ResultadosParkfull ParkDQ0;
92 Filtrovector VoltajeGenerador;
93 Filtrovector CorrienteGenerador;
94 Filtrovector VoltajeRed;
95 Filtrovector CorrienteRed;
96 Filtrovector CorrienteDesahogo;
97 Filtrocontinuo CorrienteCampo;
98
99 //-----
100 //Funciones externas cargadas
101 //-----
102 extern void InitCpuTimers(void);
103 extern void ConfigGPIOregister(void);
104 extern void InitAdc(void);
105 extern void ConfigADCregister(void);
106 extern void ConfigSERIALregister(void);
107 extern void ConfigCpuTimer(struct CPUTIMER_VARS *, float, float);
108
109 //-----
110 //Funciones internas
111 //-----
112 void Changeduty(int, int);
113 void ConfigurarEPWM(void);
114 void PIDChopper(float setpoint, float actual_position, int duty);
115 void PIDDesahogo(float setpoint, float actual_position, int duty);
116 void PIDEquilibrar(float setpoint, float v1, float v2, float v3, int duty1, int duty2, int duty3);
117 void Filtrar2HzCont(float salida, float Entrada);
118 float Absoluto(float f);
119 void InicializaFiltrovector(struct Filtrovector *elfiltro, float CInicial);
120 void Filtrar2Hz(struct Filtrovector *elfiltro, float Entrada [3]);
121 void Filtrar1Hz(struct Filtrocontinuo *elfiltro, float Entrada);
122 void Filtrar100Hz(struct Filtrovector *elfiltro, float Entrada [3]);
123 void InicializaFiltrocontinuo(struct Filtrocontinuo *elfiltro, float CInicial);
124 void inicializaChopper();
125 void Filtrar1HzCont(struct Filtrocontinuo *elfiltro, float Entrada);
126
127 //-----
128 //Interrupciones
129 //-----
130 interrupt void epwm5_isr(void);
131 interrupt void adc_isr(void);
132 interrupt void cpu_timer0_isr(void);
133 interrupt void SCIA_RX_isr(void);
134
135 //-----
136 //variables utilizadas
137 //-----
138 float anguloGenerador=0;
139 float anguloRed=0;
140 float pre_error = 0, pre_errordes=0, pre_error1 = 0, pre_error2 = 0, pre_error3 = 0;
141 float integral = 0, integraldes=0, integral1 = 0, integral2 = 0, integral3 = 0;
142 float error=0, errordes=0, error1=0, error2=0, error3=0;
143 float derivative=0, derivativedes=0, derivative1=0, derivative2=0, derivative3=0;

```



```

144 float output=0, outputdes=0, output1=0, output2=0, output3=0;
145 int dutyChopper=0, dutyD1=0, dutyD2=0, dutyD3=0;
146 float frecgen=0;
147 int delta=500;//por determinar, corresponde al porcentaje del pwm que permite obtener la referencia de 1A
148 float If=0, Ifmin=0.5, Ifmax=1.6, Ifprox=0;
149 float Vref=220, VGenerador=0, VGenerador2=0, Verror=0;
150 //Matrices de transformación
151 float m1[2][3]={{ -1.0/3.0,1.0/6.0,1.0/6.0},{0, -1.73205080757/6.0,1.73205080757/6}};
152 float m2[2][3]={{0, -1.73205080757/6.0,1.73205080757/6.0},{1.0/3.0, -1.0/6.0, -1.0/6.0}};
153 //Variables Auxiliares
154 int cambio=0,deq1=0,deq2=0,deq3=0;
155 float cdesa=0;
156 int contiendoentraPID=0, contiendoentraPIDDES=0, contiendoentraequilibrio=0 ;
157 int dutyPID=0, dutyPIDDES=0;
158 int blablabla=0;
159 float periodo=PERIODODEFINE;//El período de muestreo. Lo defino acá para que se haga una sola vez la
    división de PERIODODEFINE
160 int contador = 0;
161 int duty = 0;
162 int duty2=0;
163 //Variables para comunicación serial
164 char character;
165 char buffer[15] = 0;
166 char estado[4];
167 float vredrmsclarkefull=0;
168 float iredrmsclarkefull=0;
169 float vredrms=0;
170 float iredrms=0;
171 float vgenrmsclarkefull=0;
172 float igenrmsclarkefull=0;
173 float vgenrms=0;
174 float igenrms=0;
175 float idesrmsclarkefull=0;
176 float idesrms=0;
177
178 //Variables auxiliares tension generador
179 float auxminlvgen=4000, auxmaxlvgen=0, auxmin2vgen=4000, auxmax2vgen=0, auxmin3vgen=4000, auxmax3vgen=0;
180 float auxmin1lvgen=200, auxmax1lvgen=-200, auxmin2lvgen=-200, auxmax2lvgen=200, auxmin3lvgen=-200,
    auxmax3lvgen=200;
181 float auxlvgen=0, aux2vgen=0, aux3vgen=0;
182 //Variables auxiliares corriente generador
183 float auxminligen=4000, auxmaxligen=0, auxmin2igen=4000, auxmax2igen=0, auxmin3igen=4000, auxmax3igen=0;
184 float auxmin1ligen=200, auxmax1ligen=-200, auxmin2ligen=-200, auxmax2ligen=200, auxmin3ligen=-200,
    auxmax3ligen=200;
185 float auxligen=0, aux2igen=0, aux3igen=0;
186 //variables auxiliares tension red
187 float auxmin1vred=4000, auxmax1vred=0, auxmin2vred=4000, auxmax2vred=0, auxmin3vred=4000, auxmax3vred=0;
188 float auxmin1lvred=200, auxmax1lvred=-200, auxmin2lvred=-200, auxmax2lvred=200, auxmin3lvred=-200,
    auxmax3lvred=200;
189 float auxlvred=0, aux2vred=0, aux3vred=0;
190 //variables auxiliares corriente red
191 float auxmin1ired=4000, auxmax1ired=0, auxmin2ired=4000, auxmax2ired=0, auxmin3ired=4000, auxmax3ired=0;
192 float auxmin1lired=200, auxmax1lired=-200, auxmin2lired=-200, auxmax2lired=200, auxmin3lired=-200,
    auxmax3lired=200;
193 float auxlired=0, aux2ired=0, aux3ired=0;
194 //variables auxiliares corriente desahogo
195 float auxmin1ides=4000, auxmax1ides=0, auxmin2ides=4000, auxmax2ides=0, auxmin3ides=4000, auxmax3ides=0;
196 float auxmin1lides=200, auxmax1lides=-200, auxmin2lides=-200, auxmax2lides=200, auxmin3lides=-200,
    auxmax3lides=200;
197 float auxlides=0, aux2ides=0, aux3ides=0;
198 float contandoblaba=0;
199 float auxconta=0;
200 //variable auxiliar para observar pll
201 float cosenoangulopll=0;
202 float senoangulopll=0;
203 float sumatrigonometricas=0;
204 float clarkecos=0;
205 float clarkesen=0;
206 float auxiliarwtred=0;
207 float contadorfrecuenciainterrupcion=0;
208 //variables PLL
209 float errorangulored=0;
210 float errorangulogenerador=0;
211 float angulopllred1=0;
212 float angulopllgenerador1=0;
213 float angulopllred0=0;
214 float angulopllgenerador0=0;
215 float frecuenciared=0;
216 float frecuenciagenerador=0;
217 float kppll=0;
218 float kippll=0;
219 float ctea=0.9;//por determinar
220 float ctec=0.5;//por determinar
221 float integral0pllred=0;
222 float integral0pllgenerador=0;
223 float integral1pllred=0;
224 float integral1pllgenerador=0;
225 float wredpll=0;
226 float wredpll1=0;
227 float wredpll0=0;
228 float wgeneradorpll=0;
229 float wredpllfiltrada=0;
230 float wgeneradorpllfiltrada=0;
231 float errorestimado1=0;
232 float errorestimado0=0;

```

```

233 float frecuenciaredfiltrada=0;
234
235 //cálculo frecuencia cruce por cero para red
236 int primerred=0;
237 int segunred=0;
238 int contandofrecuenciared=0;
239 int mediore=0;
240 int signored=0;
241 float frecuenciaredcruce=0;
242 float frecuenciaredcrucefiltrada=0;
243
244 //cálculo frecuencia cruce por cero para generador
245 int primergen=0;
246 int segungen=0;
247 int contandofrecuenciagen=0;
248 int mediogen=0;
249 int signogen=0;
250 float frecuenciagencruce=0;
251 float frecuenciagencrucefiltrada=0;
252
253 //calculo rms
254 float maximored=0;
255 float maximogen=0;
256 float contadorrms=0;
257 float rmsred=0;
258 float rmsgen=0;
259 float rmsredfiltrada=0;
260 float rmsgenfiltrada=0;
261 float rmsredfiltrada2=0;
262 float rmsgenfiltrada2=0;
263
264 extern Uint16 RamfuncsLoadStart;
265 extern Uint16 RamfuncsLoadEnd;
266 extern Uint16 RamfuncsRunStart;
267
268 void main(void)
269 {
270 // Step 1. Initialize System Control: PLL, WatchDog, enable Peripheral Clocks
271   InitSysCtrl();
272   // Configuración del reloj para ADC
273   EALLOW; // Para poder escribir en los registros protegidos
274   SysCtrlRegs.HISPCP.all = ADCMODCLK;
275   EDIS;
276
277 // Step 3. Clear all interrupts and initialize PIE vector table:
278
279   DINT; // Deshabilitar interrupciones
280   InitPieCtrl(); // Inicializar registros de control del PIE en estado por defecto
281   IER = 0x0000; // Disable CPU interrupts
282   IFR = 0x0000; // Clear all CPU interrupt flags;
283
284 // Initialize the PIE vector table with pointers to the shell Interrupt
285 // Service Routines (ISR).
286 // This will populate the entire table, even if the interrupt
287 // is not used in this example. This is useful for debug purposes.
288 // The shell ISR routines are found in DSP2833x_DefaultIsr.c.
289 // This function is found in DSP2833x_PieVect.c.
290   InitPieVectTable();
291
292 // Step 4. Initialize all the Device Peripherals:
293
294   InitCpuTimers();
295   ConfigurarEPWM();
296
297 //   HabilitarSerial();
298
299
300   EALLOW;
301   //HABILITAR PINES
302   ConfigGPIOregister();
303
304   EDIS;
305
306 // Step 5. Habilitar interrupciones:
307   memcpy(&RamfuncsRunStart, &RamfuncsLoadStart, &RamfuncsLoadEnd - &RamfuncsLoadStart);
308
309   InitFlash(); //Setea waitstates de la RAM. Esta función se manda a la RAM con DSP2833x_SysCtrl.c
310   InitAdc();
311
312 // Configurar el ADC:
313   ConfigADCregister();
314   ConfigSERIALregister();
315
316 // Redirige interrupciones a funciones ISR de este archivo
317   EALLOW;
318
319   PieVectTable.EPWM5_INT = &epwm5_isr;
320   PieVectTable.ADCINT = &adc_isr;
321   PieVectTable.SCIRXINTA = &SCIA_RX_isr;
322
323
324   EDIS; // Desactiva escritura en registros protegidos
325
326 // Habilitar interrupciones ADC
327   PieCtrlRegs.PIEIER1.bit.INTx6 = 1;

```

```

328 IER |= M.INT1;
329 EINT; // Enable Global interrupt INTM
330 ERTM; // Enable Global realtime interrupt DBGEM
331
332 //Habilitar interrupciones EPWM
333 IER |= M.INT3; // Enable CPU INT3 which is connected to EPWM1-3 INT:
334
335
336 PieCtrlRegs.PIEIER3.bit.INTx5 = 1; // Enable EPWM INTn in the PIE: Group 3 interrupt 1-3 Para las
    interrupciones del ePWM 5
337
338
339 // Habilitar interrupciones para la comunicación serial
340 PieCtrlRegs.PIECTRL.bit.ENPIE = 1; // Habilita el bloque PIE
341 PieCtrlRegs.PIEIER9.bit.INTx1 = 1; // Habilita SCIRXINTA
342
343 IER |= M.INT9; // Enable CPU Interrupt 9
344
345 EINT; // Enable Global interrupt INTM
346 ERTM; // Enable Global realtime interrupt DBGEM higher priority real-time debug
347
348 inicializaChopper();
349
350 // Step 6. Loop infinito:
351 for(;;)
352 {
353     asm("    NOP");
354 }
355
356 }
357
358
359 interrupt void adc_isr(void) //esta programada a 8400 Hz, pero esta funcionando a 4500 Hz aprox
360 {
361     GpioDataRegs.GPATOGGLE.bit.GPIO12 = 1;
362     contadorfrecuenciainterrupcion++;
363     if (contandoblaba==100000){
364         contandoblaba=0;
365     }
366     if (contandoblaba==0){
367         //inicializacion auxiliares generador
368         auxmax1lvgen=auxmax22vgen=auxmax33vgen=auxmax1vgen=auxmax2vgen=auxmax3vgen=0;
369         auxmin1lvgen=auxmin22vgen=auxmin33vgen=auxmin1vgen=auxmin2vgen=auxmin3vgen=5000;
370         auxmax1ligen=auxmax22igen=auxmax33igen=auxmax1igen=auxmax2igen=auxmax3igen=0;
371         auxmin1ligen=auxmin22igen=auxmin33igen=auxmin1igen=auxmin2igen=auxmin3igen=5000;
372         //inicializacion auxiliares red
373         auxmax1lvred=auxmax22vred=auxmax33vred=auxmax1vred=auxmax2vred=auxmax3vred=0;
374         auxmin1lvred=auxmin22vred=auxmin33vred=auxmin1vred=auxmin2vred=auxmin3vred=5000;
375         auxmax1lired=auxmax22ired=auxmax33ired=auxmax1lired=auxmax2ired=auxmax3ired=0;
376         auxmin1lired=auxmin22ired=auxmin33ired=auxmin1lired=auxmin2ired=auxmin3ired=5000;
377         //inicializacion auxiliares desahogo
378         auxmax1lides=auxmax22ides=auxmax33ides=auxmax1lides=auxmax2ides=auxmax3ides=0;
379         auxmin1lides=auxmin22ides=auxmin33ides=auxmin1lides=auxmin2ides=auxmin3ides=5000;
380     }
381
382
383
384
385 //-----
386 //Estimación de ceros para centrar las mediciones
387 //-----
388
389 //Estimación cero del voltaje generador
390 if ((float) (AdcRegs.ADCRESULT0 >>4) <= auxmin1vgen) auxmin1vgen = (float) (AdcRegs.ADCRESULT0 >>4);
391 if ((float) (AdcRegs.ADCRESULT0 >>4) >= auxmax1vgen) auxmax1vgen = (float) (AdcRegs.ADCRESULT0 >>4);
392 aux1vgen = (auxmax1vgen + auxmin1vgen) / 2;
393 if ((float) (AdcRegs.ADCRESULT1 >>4) <= auxmin2vgen) auxmin2vgen = (float) (AdcRegs.ADCRESULT1 >>4);
394 if ((float) (AdcRegs.ADCRESULT1 >>4) >= auxmax2vgen) auxmax2vgen = (float) (AdcRegs.ADCRESULT1 >>4);
395 aux2vgen = (auxmax2vgen + auxmin2vgen) / 2;
396 if ((float) (AdcRegs.ADCRESULT2 >>4) <= auxmin3vgen) auxmin3vgen = (float) (AdcRegs.ADCRESULT2 >>4);
397 if ((float) (AdcRegs.ADCRESULT2 >>4) >= auxmax3vgen) auxmax3vgen = (float) (AdcRegs.ADCRESULT2 >>4);
398 aux3vgen = (auxmax3vgen + auxmin3vgen) / 2;
399
400 //Estimación cero de corriente generador
401 if ((float) (AdcRegs.ADCRESULT3 >>4) <= auxmin1igen) auxmin1igen = (float) (AdcRegs.ADCRESULT3 >>4);
402 if ((float) (AdcRegs.ADCRESULT3 >>4) >= auxmax1igen) auxmax1igen = (float) (AdcRegs.ADCRESULT3 >>4);
403 aux1igen = (auxmax1igen + auxmin1igen) / 2;
404 if ((float) (AdcRegs.ADCRESULT4 >>4) <= auxmin2igen) auxmin2igen = (float) (AdcRegs.ADCRESULT4 >>4);
405 if ((float) (AdcRegs.ADCRESULT4 >>4) >= auxmax2igen) auxmax2igen = (float) (AdcRegs.ADCRESULT4 >>4);
406 aux2igen = (auxmax2igen + auxmin2igen) / 2;
407 if ((float) (AdcRegs.ADCRESULT5 >>4) <= auxmin3igen) auxmin3igen = (float) (AdcRegs.ADCRESULT5 >>4);
408 if ((float) (AdcRegs.ADCRESULT5 >>4) >= auxmax3igen) auxmax3igen = (float) (AdcRegs.ADCRESULT5 >>4);
409 aux3igen = (auxmax3igen + auxmin3igen) / 2;
410
411 //Estimación cero del voltaje red
412 if ((float) (AdcRegs.ADCRESULT6 >>4) <= auxmin1vred) auxmin1vred = (float) (AdcRegs.ADCRESULT6 >>4);
413 if ((float) (AdcRegs.ADCRESULT6 >>4) >= auxmax1vred) auxmax1vred = (float) (AdcRegs.ADCRESULT6 >>4);
414 aux1vred = (auxmax1vred + auxmin1vred) / 2;
415 if ((float) (AdcRegs.ADCRESULT8 >>4) <= auxmin2vred) auxmin2vred = (float) (AdcRegs.ADCRESULT8 >>4);
416 if ((float) (AdcRegs.ADCRESULT8 >>4) >= auxmax2vred) auxmax2vred = (float) (AdcRegs.ADCRESULT8 >>4);
417 aux2vred = (auxmax2vred + auxmin2vred) / 2;
418 if ((float) (AdcRegs.ADCRESULT7 >>4) <= auxmin3vred) auxmin3vred = (float) (AdcRegs.ADCRESULT7 >>4);
419 if ((float) (AdcRegs.ADCRESULT7 >>4) >= auxmax3vred) auxmax3vred = (float) (AdcRegs.ADCRESULT7 >>4);
420 aux3vred = (auxmax3vred + auxmin3vred) / 2;
421
422 //Estimación cero de corriente red

```

```

422 if ((float) (AdcRegs.ADCRESULT9 >>4) <= auxmin1ired) auxmin1ired = (float) (AdcRegs.ADCRESULT9 >>4);
423 if ((float) (AdcRegs.ADCRESULT9 >>4) >= auxmax1ired) auxmax1ired = (float) (AdcRegs.ADCRESULT9 >>4);
424 aux1ired = (auxmax1ired + auxmin1ired) / 2;
425 if ((float) (AdcRegs.ADCRESULT10 >>4) <= auxmin2ired) auxmin2ired = (float) (AdcRegs.ADCRESULT10 >>4);
426 if ((float) (AdcRegs.ADCRESULT10 >>4) >= auxmax2ired) auxmax2ired = (float) (AdcRegs.ADCRESULT10 >>4);
427 aux2ired = (auxmax2ired + auxmin2ired) / 2;
428 if ((float) (AdcRegs.ADCRESULT11 >>4) <= auxmin3ired) auxmin3ired = (float) (AdcRegs.ADCRESULT11 >>4);
429 if ((float) (AdcRegs.ADCRESULT11 >>4) >= auxmax3ired) auxmax3ired = (float) (AdcRegs.ADCRESULT11 >>4);
430 aux3ired = (auxmax3ired + auxmin3ired) / 2;
431
432 // Estimación cero de corriente desahogo
433 if ((float) (AdcRegs.ADCRESULT12 >>4) <= auxminlides) auxminlides = (float) (AdcRegs.ADCRESULT12 >>4);
434 if ((float) (AdcRegs.ADCRESULT12 >>4) >= auxmaxlides) auxmaxlides = (float) (AdcRegs.ADCRESULT12 >>4);
435 aux1lides = (auxmaxlides + auxminlides) / 2;
436 if ((float) (AdcRegs.ADCRESULT13 >>4) <= auxmin2ides) auxmin2ides = (float) (AdcRegs.ADCRESULT13 >>4);
437 if ((float) (AdcRegs.ADCRESULT13 >>4) >= auxmax2ides) auxmax2ides = (float) (AdcRegs.ADCRESULT13 >>4);
438 aux2ides = (auxmax2ides + auxmin2ides) / 2;
439 if ((float) (AdcRegs.ADCRESULT14 >>4) <= auxmin3ides) auxmin3ides = (float) (AdcRegs.ADCRESULT14 >>4);
440 if ((float) (AdcRegs.ADCRESULT14 >>4) >= auxmax3ides) auxmax3ides = (float) (AdcRegs.ADCRESULT14 >>4);
441 aux3ides = (auxmax3ides + auxmin3ides) / 2;
442
443
444 //-----
445 // Mediciones instantáneas
446 //-----
447
448
449 if (contandoblable > 100) {
450
451 /* el sensor de tensión tiene por valor nominal 230 V, por lo que alejandose de éste se descalibran las
452 mediciones
453 * hay que sacar la curva de histeresis del transformador para calibrar bien los sensores
454 */
455
456 UltimoADC.VGenerador[0] = 0.229113709985704 * ((float) (AdcRegs.ADCRESULT0 >>4) - aux1vgen);
457 UltimoADC.VGenerador[1] = 0.233036021732131 * ((float) (AdcRegs.ADCRESULT1 >>4) - aux2vgen);
458 UltimoADC.VGenerador[2] = 0.210464408506768 * ((float) (AdcRegs.ADCRESULT2 >>4) - aux3vgen);
459
460 UltimoADC.VRed[0] = 0.228361743497944 * ((float) (AdcRegs.ADCRESULT6 >>4) - aux1vred);
461 UltimoADC.VRed[1] = 0.214547114996597 * ((float) (AdcRegs.ADCRESULT8 >>4) - aux2vred);
462 UltimoADC.VRed[2] = 0.209212605653782 * ((float) (AdcRegs.ADCRESULT7 >>4) - aux3vred);
463
464 UltimoADC.IGenerador[0] = 0.0460608933688959 * ((float) (AdcRegs.ADCRESULT3 >>4) - aux1igen);
465 UltimoADC.IGenerador[1] = 0.0427941658365769 * ((float) (AdcRegs.ADCRESULT4 >>4) - aux2igen);
466 UltimoADC.IGenerador[2] = 0.0428955726490664 * ((float) (AdcRegs.ADCRESULT5 >>4) - aux3igen);
467
468 UltimoADC.IRed[0] = 0.0496215296454552 * ((float) (AdcRegs.ADCRESULT9 >>4) - aux1ired);
469 UltimoADC.IRed[1] = 0.0508569186174042 * ((float) (AdcRegs.ADCRESULT10 >>4) - aux2ired);
470 UltimoADC.IRed[2] = 0.0503003434237189 * ((float) (AdcRegs.ADCRESULT11 >>4) - aux3ired);
471
472 UltimoADC.IDesahogo[0] = FACTORCORRIENTES * ((float) (AdcRegs.ADCRESULT12 >>4) - aux1ides);
473 UltimoADC.IDesahogo[1] = FACTORCORRIENTES * ((float) (AdcRegs.ADCRESULT13 >>4) - aux2ides);
474 UltimoADC.IDesahogo[2] = FACTORCORRIENTES * ((float) (AdcRegs.ADCRESULT14 >>4) - aux3ides);
475
476 UltimoADC.ICampo = ((float) (AdcRegs.ADCRESULT15 >>4) - 1898);
477
478 }
479 contandoblable++;
480
481 //-----
482 // Para observar maximos y minimos instantáneos
483 //-----
484
485 if (UltimoADC.VRed[0] <= auxmin11vred) auxmin11vred = UltimoADC.VRed[0];
486 if (UltimoADC.VRed[0] >= auxmax11vred) auxmax11vred = UltimoADC.VRed[0];
487 if (UltimoADC.VRed[1] <= auxmin22vred) auxmin22vred = UltimoADC.VRed[1];
488 if (UltimoADC.VRed[1] >= auxmax22vred) auxmax22vred = UltimoADC.VRed[1];
489 if (UltimoADC.VRed[2] <= auxmin33vred) auxmin33vred = UltimoADC.VRed[2];
490 if (UltimoADC.VRed[2] >= auxmax33vred) auxmax33vred = UltimoADC.VRed[2];
491
492 if (UltimoADC.IRed[0] <= auxmin11ired) auxmin11ired = UltimoADC.IRed[0];
493 if (UltimoADC.IRed[0] >= auxmax11ired) auxmax11ired = UltimoADC.IRed[0];
494 if (UltimoADC.IRed[1] <= auxmin22ired) auxmin22ired = UltimoADC.IRed[1];
495 if (UltimoADC.IRed[1] >= auxmax22ired) auxmax22ired = UltimoADC.IRed[1];
496 if (UltimoADC.IRed[2] <= auxmin33ired) auxmin33ired = UltimoADC.IRed[2];
497 if (UltimoADC.IRed[2] >= auxmax33ired) auxmax33ired = UltimoADC.IRed[2];
498
499 if (UltimoADC.IDesahogo[0] <= auxmin11ides) auxmin11ides = UltimoADC.IDesahogo[0];
500 if (UltimoADC.IDesahogo[0] >= auxmax11ides) auxmax11ides = UltimoADC.IDesahogo[0];
501 if (UltimoADC.IDesahogo[1] <= auxmin22ides) auxmin22ides = UltimoADC.IDesahogo[1];
502 if (UltimoADC.IDesahogo[1] >= auxmax22ides) auxmax22ides = UltimoADC.IDesahogo[1];
503 if (UltimoADC.IDesahogo[2] <= auxmin33ides) auxmin33ides = UltimoADC.IDesahogo[2];
504 if (UltimoADC.IDesahogo[2] >= auxmax33ides) auxmax33ides = UltimoADC.IDesahogo[2];
505
506 if (UltimoADC.VGenerador[0] <= auxmin11vgen) auxmin11vgen = UltimoADC.VGenerador[0];
507 if (UltimoADC.VGenerador[0] >= auxmax11vgen) auxmax11vgen = UltimoADC.VGenerador[0];
508 if (UltimoADC.VGenerador[1] <= auxmin22vgen) auxmin22vgen = UltimoADC.VGenerador[1];
509 if (UltimoADC.VGenerador[1] >= auxmax22vgen) auxmax22vgen = UltimoADC.VGenerador[1];
510 if (UltimoADC.VGenerador[2] <= auxmin33vgen) auxmin33vgen = UltimoADC.VGenerador[2];
511 if (UltimoADC.VGenerador[2] >= auxmax33vgen) auxmax33vgen = UltimoADC.VGenerador[2];
512
513 if (UltimoADC.IGenerador[0] <= auxmin11igen) auxmin11igen = UltimoADC.IGenerador[0];
514 if (UltimoADC.IGenerador[0] >= auxmax11igen) auxmax11igen = UltimoADC.IGenerador[0];
515 if (UltimoADC.IGenerador[1] <= auxmin22igen) auxmin22igen = UltimoADC.IGenerador[1];
516 if (UltimoADC.IGenerador[1] >= auxmax22igen) auxmax22igen = UltimoADC.IGenerador[1];

```

```

516     if (UltimoADC . IGenerador [2] <= auxmin33igen) auxmin33igen=UltimoADC . IGenerador [2];
517     if (UltimoADC . IGenerador [2] >= auxmax33igen) auxmax33igen=UltimoADC . IGenerador [2];
518
519
520
521
522 //-----
523 // Para filtrar , en este instante esta sin filtrar
524 //-----
525
526     Filtrar100Hz(&VoltajeGenerador , UltimoADC . VGenerador );
527     Filtrar100Hz(&CorrienteGenerador , UltimoADC . IGenerador );
528     Filtrar100Hz(&VoltajeRed , UltimoADC . VRed );
529     Filtrar100Hz(&CorrienteRed , UltimoADC . IRed );
530     Filtrar100Hz(&CorrienteDesahogo , UltimoADC . IDesahogo );
531
532     Filtrar1HzCont(&CorrienteCampo , UltimoADC . ICampo );
533
534
535
536 //-----
537 // Calculo Clarke
538 //-----
539
540     // Clarke Red
541     Clarkealfabetagamma . VRed [0] = VoltajeRed . Salida [0] - ( VoltajeRed . Salida [1] + VoltajeRed . Salida [2] ) * 0.5;
542     Clarkealfabetagamma . VRed [1] = 0.8660254037844386 * ( VoltajeRed . Salida [1] - VoltajeRed . Salida [2] );
543     Clarkealfabetagamma . VRed [2] = 0.5 * ( VoltajeRed . Salida [0] + VoltajeRed . Salida [1] + VoltajeRed . Salida [2] );
544
545     Clarkealfabetagamma . IRed [0] = CorrienteRed . Salida [0] - ( CorrienteRed . Salida [1] + CorrienteRed . Salida [2] )
546     * 0.5;
547     Clarkealfabetagamma . IRed [1] = 0.8660254037844386 * ( CorrienteRed . Salida [1] - CorrienteRed . Salida [2] );
548     Clarkealfabetagamma . IRed [2] = 0.5 * ( CorrienteRed . Salida [0] + CorrienteRed . Salida [1] + CorrienteRed . Salida
549     [2] );
550
551     // Clarke Generador
552     Clarkealfabetagamma . VGenerador [0] = VoltajeGenerador . Salida [0] - ( VoltajeGenerador . Salida [1] +
553     VoltajeGenerador . Salida [2] ) * 0.5;
554     Clarkealfabetagamma . VGenerador [1] = 0.8660254037844386 * ( VoltajeGenerador . Salida [1] - VoltajeGenerador .
555     Salida [2] );
556     Clarkealfabetagamma . VGenerador [2] = 0.5 * ( VoltajeGenerador . Salida [0] + VoltajeGenerador . Salida [1] +
557     VoltajeGenerador . Salida [2] );
558
559     Clarkealfabetagamma . IGenerador [0] = CorrienteGenerador . Salida [0] - ( CorrienteGenerador . Salida [1] +
560     CorrienteGenerador . Salida [2] ) * 0.5;
561     Clarkealfabetagamma . IGenerador [1] = 0.8660254037844386 * ( CorrienteGenerador . Salida [1] -
562     CorrienteGenerador . Salida [2] );
563     Clarkealfabetagamma . IGenerador [2] = 0.5 * ( CorrienteGenerador . Salida [0] + CorrienteGenerador . Salida [1] +
564     CorrienteGenerador . Salida [2] );
565
566     // Clarke Desahogo
567     Clarkealfabetagamma . IDesahogo [0] = CorrienteDesahogo . Salida [0] - ( CorrienteDesahogo . Salida [1] +
568     CorrienteDesahogo . Salida [2] ) * 0.5;
569     Clarkealfabetagamma . IDesahogo [1] = 0.8660254037844386 * ( CorrienteDesahogo . Salida [1] - CorrienteDesahogo .
570     Salida [2] );
571     Clarkealfabetagamma . IDesahogo [2] = 0.5 * ( CorrienteDesahogo . Salida [0] + CorrienteDesahogo . Salida [1] +
572     CorrienteDesahogo . Salida [2] );
573
574
575
576 //-----
577 // Calculo ángulo
578 //-----
579
580     anguloGenerador = atan2 ( Clarkealfabetagamma . VGenerador [1] , Clarkealfabetagamma . VGenerador [0] );
581     if ( anguloGenerador > DOS_PI ) anguloGenerador = anguloGenerador - DOS_PI;
582     else if ( anguloGenerador < 0 ) anguloGenerador = anguloGenerador + DOS_PI;
583
584     anguloRed = atan2 ( Clarkealfabetagamma . VRed [1] , Clarkealfabetagamma . VRed [0] );
585     if ( anguloRed > DOS_PI ) anguloRed = anguloRed - DOS_PI;
586     else if ( anguloRed < 0 ) anguloRed = anguloRed + DOS_PI;
587
588
589
590 //-----
591 // Calculo Park
592 //-----
593
594     // Red
595     ParkDQ0 . VRed [0] = Clarkealfabetagamma . VRed [0] * ( cos ( anguloRed ) ) + Clarkealfabetagamma . VRed [1] * ( sin (
596     anguloRed ) );
597     ParkDQ0 . VRed [1] = - Clarkealfabetagamma . VRed [0] * ( sin ( anguloRed ) ) + Clarkealfabetagamma . VRed [1] * ( cos (
598     anguloRed ) );
599     ParkDQ0 . VRed [2] = Clarkealfabetagamma . VRed [2];
600
601     ParkDQ0 . IRed [0] = Clarkealfabetagamma . IRed [0] * ( cos ( anguloRed ) ) + Clarkealfabetagamma . IRed [1] * ( sin (
602     anguloRed ) );
603     ParkDQ0 . IRed [1] = - Clarkealfabetagamma . IRed [0] * ( sin ( anguloRed ) ) + Clarkealfabetagamma . IRed [1] * ( cos (
604     anguloRed ) );
605     ParkDQ0 . IRed [2] = Clarkealfabetagamma . IRed [2];
606
607     // Generador
608     ParkDQ0 . VGenerador [0] = Clarkealfabetagamma . VGenerador [0] * ( cos ( anguloGenerador ) ) + Clarkealfabetagamma .
609     VGenerador [1] * ( sin ( anguloGenerador ) );
610     ParkDQ0 . VGenerador [1] = - Clarkealfabetagamma . VGenerador [0] * ( sin ( anguloGenerador ) ) + Clarkealfabetagamma
611     . VGenerador [1] * ( cos ( anguloGenerador ) );
612     ParkDQ0 . VGenerador [2] = Clarkealfabetagamma . VGenerador [2];

```

```

594
595 ParkDQ0.IGenerador[0]=Clarkealfabetagamma.IGenerador[0]*(cos(anguloRed))+Clarkealfabetagamma.
IGenerador[1]*(sin(anguloRed));
596 ParkDQ0.IGenerador[1]=-Clarkealfabetagamma.IGenerador[0]*(sin(anguloRed))+Clarkealfabetagamma.
IGenerador[1]*(cos(anguloRed));
597 ParkDQ0.IGenerador[2]=Clarkealfabetagamma.IGenerador[2];
598
599 //Desahogo
600 ParkDQ0.IDesahogo[0]=Clarkealfabetagamma.IDesahogo[0]*(cos(anguloRed))+Clarkealfabetagamma.
IDesahogo[1]*(sin(anguloRed));
601 ParkDQ0.IDesahogo[1]=-Clarkealfabetagamma.IDesahogo[0]*(sin(anguloRed))+Clarkealfabetagamma.
IDesahogo[1]*(cos(anguloRed));
602 ParkDQ0.IDesahogo[2]=Clarkealfabetagamma.IDesahogo[2];
603
604
605
606 //-----
607 //Calculo rms y filtrado con clarke
608 //-----
609
610 //Red
611 vredrmsclarkefull=0.5396871084938658017*sqrt(Clarkealfabetagamma.VRed[0]*Clarkealfabetagamma.VRed
[0]+Clarkealfabetagamma.VRed[1]*Clarkealfabetagamma.VRed[1]+Clarkealfabetagamma.VRed[2]*
Clarkealfabetagamma.VRed[2]);
612
613 // vredrmsclarkefull=0.5326871084938658017*sqrt(Clarkealfabetagamma.VRed[0]*Clarkealfabetagamma.VRed
[0]+Clarkealfabetagamma.VRed[1]*Clarkealfabetagamma.VRed[1]+Clarkealfabetagamma.VRed[2]*
Clarkealfabetagamma.VRed[2]);
614 vredrms=0.00298752145308*vredrmsclarkefull+0.99701247854692*vredrms;//filtrado de 1 Hz con
discretizacion a 2100Hz
615
616 iredrmsclarkefull=0.5326871084938658017*sqrt(Clarkealfabetagamma.IRed[0]*Clarkealfabetagamma.IRed
[0]+Clarkealfabetagamma.IRed[1]*Clarkealfabetagamma.IRed[1]+Clarkealfabetagamma.IRed[2]*
Clarkealfabetagamma.IRed[2]);
617 iredrms=0.00298752145308*iredrmsclarkefull+0.99701247854692*iredrms;//filtrado de 1 Hz con
discretizacion a 2100Hz
618
619 //0.471404520791032000 0.5326871084938658017 0000
620
621 //Generador RMS
622 vgenrmsclarkefull=0.5165871084938658017*sqrt(Clarkealfabetagamma.VGenerador[0]*Clarkealfabetagamma.
VGenerador[0]+Clarkealfabetagamma.VGenerador[1]*Clarkealfabetagamma.VGenerador[1]+
Clarkealfabetagamma.VGenerador[2]*Clarkealfabetagamma.VGenerador[2]);
623 // vgenrmsclarkefull=0.5136871084938658017*sqrt(Clarkealfabetagamma.VGenerador[0]*Clarkealfabetagamma.
VGenerador[0]+Clarkealfabetagamma.VGenerador[1]*Clarkealfabetagamma.VGenerador[1]+Clarkealfabetagamma.
VGenerador[2]*Clarkealfabetagamma.VGenerador[2]);
624 igenrmsclarkefull=0.6426871084938658017*sqrt(Clarkealfabetagamma.IGenerador[0]*Clarkealfabetagamma.
IGenerador[0]+Clarkealfabetagamma.IGenerador[1]*Clarkealfabetagamma.IGenerador[1]+
Clarkealfabetagamma.IGenerador[2]*Clarkealfabetagamma.IGenerador[2]);
625
626 //Filtradas a 1Hz
627 vgenrms=0.00298752145308*vgenrmsclarkefull+0.99701247854692*vgenrms;
628 igenrms=0.00298752145308*igenrmsclarkefull+0.99701247854692*igenrms;
629 //Filtradas a 5Hz
630 vgenrms=0.0148486206677959*vgenrmsclarkefull+0.9851513793322041*vgenrms;
631 // igenrms=0.0148486206677959*igenrmsclarkefull+0.9851513793322041*igenrms;
632 //Filtradas a 10Hz
633 vgenrms=0.0294767597998557*vgenrmsclarkefull+0.9705232402001443*vgenrms;
634 // igenrms=0.0294767597998557*igenrmsclarkefull+0.9705232402001443*igenrms;
635
636 //Desahogo
637 // idesrmsclarkefull=0.5326871084938658017*sqrt(Clarkealfabetagamma.IDesahogo[0]*Clarkealfabetagamma.
IDesahogo[0]+Clarkealfabetagamma.IDesahogo[1]*Clarkealfabetagamma.IDesahogo[1]+Clarkealfabetagamma.
IDesahogo[2]*Clarkealfabetagamma.IDesahogo[2]);
638 idesrmsclarkefull=0.5326871084938658017*sqrt(Clarkealfabetagamma.IDesahogo[0]*Clarkealfabetagamma.
IDesahogo[0]+Clarkealfabetagamma.IDesahogo[1]*Clarkealfabetagamma.IDesahogo[1]);
639
640 idesrms=0.001395289079375*idesrmsclarkefull+0.998604710920625*idesrms;
641
642
643 //-----
644 //Cálculo de tension rms con máximo
645 //-----
646 contadorrms++;
647 if(contadorrms==10000){
648     maximored=0;
649     maximogen=0;
650     contadorrms=0;
651 }
652 if(VoltajeRed.Salida[0]>maximored){
653     maximored=VoltajeRed.Salida[0];
654 }
655 if(VoltajeGenerador.Salida[0]>maximogen){
656     maximogen=VoltajeGenerador.Salida[0];
657 }
658 if(maximored!=0 && contadorrms>2000)rmsred=maximored*UNO_DIV_RAIZ_DE_DOS;
659 if(maximogen!=0 && contadorrms>2000)rmsgen=maximogen*UNO_DIV_RAIZ_DE_DOS;
660 //filtro de 0.5 Hz
661 rmsredfiltrada=rmsred*0.0014948780567423+rmsredfiltrada*0.99850519432577;
662 rmsgenfiltrada=rmsgen*0.0014948780567423+rmsgenfiltrada*0.99850519432577;
663 //filtro de 1 Hz
664 rmsredfiltrada2=rmsredfiltrada*0.0014948780567423+rmsredfiltrada2*0.99850519432577;
665 rmsgenfiltrada2=rmsgenfiltrada*0.0014948780567423+rmsgenfiltrada2*0.99850519432577;
666 // rmsredfiltrada=rmsred*0.00298752145308008+rmsredfiltrada*0.99701247854691992;
667 // rmsgenfiltrada=rmsgen*0.00298752145308008+rmsgenfiltrada*0.99701247854691992;

```

```

668
669
670
671
672 //-----
673 // cálculo de frecuencia de red por cruces por cero
674 //-----
675
676     contandofrecuenciared++;
677     //cruces por cero
678
679     if( VoltajeRed.Salida[0]<0 && signored==1){
680         primerred=contandofrecuenciared;
681     }
682     else if (VoltajeRed.Salida[0]>0 && signored==0){
683         segured=contandofrecuenciared;
684         contandofrecuenciared=0;
685     }
686     if( VoltajeRed.Salida[0]<0) signored=0;
687     else if (VoltajeRed.Salida[0]>0) signored=1;
688
689     mediored=Absoluto(segured-primerred); //800 us
690     frecuenciaredcruce=(1/(mediored*0.000476190476190476));
691
692     // filtro 1 Hz
693     // frecuenciaredcrucefiltrada=frecuenciaredcruce*0.002987522145308+ frecuenciaredcrucefiltrada
694     *0.997012477854692;
695     // filtro 0.5 Hz
696     frecuenciaredcrucefiltrada=frecuenciaredcruce*0.0014948780567423+ frecuenciaredcrucefiltrada
697     *0.99850519432577;
698
699 //-----
700 // cálculo de frecuencia de generador
701 //-----
702     contandofrecuenciagen++;
703     //cruces por cero
704
705     if( VoltajeGenerador.Salida[0]<0 && signogen==1){
706         primergen=contandofrecuenciagen;
707     }
708     else if (VoltajeGenerador.Salida[0]>0 && signogen==0){
709         segungen=contandofrecuenciagen;
710         contandofrecuenciagen=0;
711     }
712     if( VoltajeGenerador.Salida[0]<0) signogen=0;
713     else if (VoltajeGenerador.Salida[0]>0) signogen=1;
714
715     mediogen=Absoluto(segungen-primergen); //800 us
716     frecuenciagencruce=(1/(mediogen*0.000476190476190476)); //frecuencia con error
717
718     // filtro 1Hz
719     // frecuenciagencrucefiltrada=frecuenciagencruce*0.002987522145308+ frecuenciagencrucefiltrada
720     *0.997012477854692;
721     // filtro 0.5 Hz
722     frecuenciagencrucefiltrada=frecuenciagencruce*0.0014948780567423+ frecuenciagencrucefiltrada
723     *0.99850519432577;
724
725     //////////////////////////////////////
726     //*****
727     //*INICIO CODIGO DE CONTROL*
728     //*****
729     //////////////////////////////////////
730
731 //-----
732 // Configuraciones para recordar
733 //-----
734 //-----
735
736     //     contactor generador     abierto     GpioDataRegs.GPBSET.bit.GPIO58 = 1;
737     //     contactor generador     cerrado     GpioDataRegs.GPBCLEAR.bit.GPIO58 = 1;
738     //     contactor red abierto     GpioDataRegs.GPBSET.bit.GPIO60 = 1;
739     //     contactor red cerrado     GpioDataRegs.GPBCLEAR.bit.GPIO60 = 1;
740     //     contactor motor valvula abierto     GpioDataRegs.GPBSET.bit.GPIO61 = 1;
741     //     contactor motor valvula cerrado     GpioDataRegs.GPBCLEAR.bit.GPIO61 = 1;
742     //     Abrir - Cerrar Valvula abierto     GpioDataRegs.GPASET.bit.GPIO8 = 1;
743     //     Abrir - Cerrar Valvula cerrado     GpioDataRegs.GPACLEAR.bit.GPIO8 = 1;
744     //     On-off inversor abierto     GpioDataRegs.GPBSET.bit.GPIO59 = 1;
745     //     On-off inversor cerrado     GpioDataRegs.GPBCLEAR.bit.GPIO59 = 1;
746     //     Inversor -220 abierto     GpioDataRegs.GPASET.bit.GPIO0 = 1;
747     //     Inversor -220 cerrado     GpioDataRegs.GPACLEAR.bit.GPIO0 = 1;
748     //     Chopper     Changeduty(1,duty);
749     //     Desahogo 1     Changeduty(2,duty);
750     //     Desahogo 2     Changeduty(3,duty);
751     //     Desahogo 3     Changeduty(4,duty);
752
753
754 //-----
755 // Reconocimiento de ciclo de trabajos actuales
756 //-----
757 //-----
758     dutyChopper=EPwm1Regs.CMPB;

```

```

759     dutyD1=EPwm2Regs.CMPB;
760     dutyD2=EPwm3Regs.CMPB;
761     dutyD3=EPwm4Regs.CMPB;
762
763
764 //-----
765 // Para la partida se lleva al generador a valores de referencia 220V y 50Hz
766 //-----
767 //     if(GpioDataRegs.GPBDAT.bit.GPIO58 == 1){//Contactor Generador abierto
768 //         /*
769 //             *   Control Corriente de Campo
770 //             *   Primero se controla tensión con el chopper
771 //             */
772 //         PIDChopper(220.0,rmsgenfiltrada2 ,dutyChopper);
773 //         PIDChopper(220.0,vgenrms ,dutyChopper);
774 //
775 //         /*
776 //             *   Control Corriente Desahogo
777 //             *   Se controla frecuencia con resistencias de desahogo y de forma embedida se controla
778 //             *   potencia disipada
779 //             *   Se controla frecuencia sola cuando no hay consumo
780 //             */
781 //         PIDDesahogo(50.0,frecuenciagencrucefiltrada ,dutyD1);//controla frecuencia sin carga
782 //     }
783 //-----
784 // Para operación en isla
785 //-----
786 //         /*
787 //             *   Tas logrados los valores nominales de 50 Hz y 220 V se conecta el consumo
788 //             */
789 //         if(Absoluto(frecuenciagencrucefiltrada -50.0) < 0.1 && Absoluto(rmsgenfiltrada -220.0) < 0.1)
790 //             GpioDataRegs.GPBCLEAR.bit.GPIO58 = 1;//se conecta el consumo
791 //
792 //         /*
793 //             *   Conectado el consumo se contra potencia en desahogo para equilibrar manteniendo frecuencia y
794 //             *   tensión de referencias
795 //             */
796 //         if(GpioDataRegs.GPBDAT.bit.GPIO58 == 0){
797 //             //Contactor Generador cerrado
798 //             PIDEquilibrar(50.0, 220.0, frecuenciagencrucefiltrada , rmsgenfiltrada ,dutyD1,dutyD2,dutyD3
799 //             );//controla frecuencia sin carga
800 //         }
801 //-----
802 // Reiniciar ADC
803 //-----
804     AdcRegs.ADCCTRL2.bit.RST_SEQ1 = 0x1;           // Resetea el contador de SEQ1
805     AdcRegs.ADCST.bit.INT_SEQ1_CLR = 1;           // Clear INT SEQ1 bit
806     PieCtrlRegs.PIEACK.all = PIEACK_GROUP1;      // Acknowledge interrupt to PIE
807
808     return;
809 }
810
811 void inicializaChopper(){
812     Changeduty(1,565);// se fija If en la referencia
813     Changeduty(2,cambio);
814     Changeduty(3,cambio);
815     Changeduty(4,cambio);
816     return;
817 }
818
819 //-----
820 // Definición de parámetros para control
821 //-----
822 #define epsilon 10
823 #define dt 0.01
824 #define Kp 15
825 #define Ki 0.01
826 //lexperimenta
827 // #define Kp 50
828 // #define Ki 0.5
829 //2
830 // #define Kp 24
831 // #define Ki 1.7
832 //3
833 // #define Kp 54
834 // #define Ki 6.94
835 #define Kd 0.000001
836
837 float integralbla=0;
838 float integralbla2=0;
839
840 //-----
841 // Control PI Corriente de Campo
842 //-----
843 void PIDChopper(float setpoint ,float actual_position , int duty)
844 {
845     error = (setpoint - actual_position);
846     integralbla= integral + error*Ki;
847     output = (Kp*error +integralbla);
848     if (output >4500)output=4500;
849     else if (output <300)output=300;

```



```

850     else integral=integralbla;
851     duty=(int)(output/4.59);
852     dutyPID=duty;
853     Changeduty(1,duty);
854 }
855
856 //-----
857 //Control PI frecuencia
858 //-----
859
860 void PIDDesahogo(float setpoint,float actual_position , int duty){
861     errordes = (-setpoint + actual_position);
862     integralbla2 = integraldes + errordes*KI;
863     outputdes =(Kp*errordes + integralbla2);
864     if(outputdes > 700)outputdes = 700;
865     else if(outputdes < 0)outputdes = 0;
866     else integraldes=integralbla2;
867     duty=(int)(outputdes/1.05);
868     dutyPIDDES=duty;
869     Changeduty(2,duty);
870     Changeduty(3,duty);
871     Changeduty(4,duty);
872 }
873
874 //-----
875 //Control PI Equilibrar
876 //-----
877
878 void PIDEquilibrar(float setpoint ,float v1, float v2, float v3, int duty1,int duty2,int duty3){
879     error1=(-setpoint+v1)*dt;
880     error2=(-setpoint+v2)*dt;
881     error3=(-setpoint+v3)*dt;
882     //In case of error too small then stop intergration
883     if(abs(error1) > epsilon){
884         integrall = integrall + error1*dt;
885     }
886     if(abs(error2) > epsilon){
887         integral2 = integral2 + error2*dt;
888     }
889     if(abs(error3) > epsilon){
890         integral3 = integral3 + error3*dt;
891     }
892     derivative1 = (error1 - pre_error1)/dt;
893     derivative2 = (error2 - pre_error2)/dt;
894     derivative2 = (error3 - pre_error3)/dt;
895
896     output1 =(Kp*error1 + Ki*integrall + Kd*derivative1);
897     output2 =(Kp*error2 + Ki*integral2 + Kd*derivative2);
898     output3 =(Kp*error3 + Ki*integral3 + Kd*derivative3);
899
900     duty1=(int)(duty1+output1);
901     duty2=(int)(duty2+output2);
902     duty3=(int)(duty3+output3);
903
904     deq1=duty1;
905     deq2=duty2;
906     deq3=duty3;
907     if(duty1 > 1000)duty1 = 1000;
908     if(duty1 < 0)duty1 = 0;
909
910     if(duty2 > 1000)duty2 = 1000;
911     if(duty2 < 0)duty2 = 0;
912
913     if(duty3 > 1000)duty3 = 1000;
914     if(duty3 < 0)duty3 = 0;
915
916     pre_error1 = error1;
917     pre_error2 = error2;
918     pre_error3 = error3;
919
920     if(abs(error1/dt)>0,1){
921         Changeduty(2,duty1);
922     }
923     if(abs(error2/dt)>0,1){
924         Changeduty(3,duty2);
925     }
926     if(abs(error3/dt)>0,1){
927         Changeduty(4,duty3);
928     }
929     contiandoentraequilibrio++;
930 }
931
932 //-----
933 //Configuración interrupcion para ejecución de adc
934 //-----
935
936 interrupt void epwm5_isr(void) // Inicio de ciclo. Aquí se hace SOC y se fija la bandera que usa la
937     // interrupción del adc pa saber en q parte del pwm estamos
938 {
939     AdeRegs.ADCTRL2.bit.SOC_SEQ1 = 1; // Comanda el inicio de la adquisición de datos
940     GpioDataRegs.GPATOGGLE.bit.GPIO15 = 1;
941     //GpioDataRegs.GPASET.bit.PIN_PROCESAMIENTO = 1; // Para ver cuanto tiempo de procesamiento está
942     // usado
943     // Clear INT flag for this timer

```

```

943     EPwm5Regs.ETCLR.bit.INT = 1;
944
945     // Acknowledge this interrupt to receive more interrupts from group 3
946     PieCtrlRegs.PIEACK.all = PIEACK.GROUP3;
947
948 }
949
950
951 //-----
952 // Configuración de PWMS
953 //-----
954
955 void ConfigurarEPWM()
956 {
957     EALLOW;
958     SysCtrlRegs.PCLKCR0.bit.TBCLKSYNC = 0; // Deshabilita el reloj del EPWM
959     EDIS;
960
961     InicializaFiltrovector(&VoltajeGenerador,0);
962     InicializaFiltrovector(&CorrienteGenerador,0);
963     InicializaFiltrovector(&VoltajeRed,0);
964     InicializaFiltrovector(&CorrienteGenerador,0);
965     InicializaFiltrovector(&CorrienteRed,0);
966     InicializaFiltrovector(&CorrienteDesahogo,0);
967
968     InicializaFiltrocontinuo(&CorrienteCampo,0);
969
970
971 //-----
972 // Configuración Periodo PWM
973 //-----
974     //
975     //      TBPRD=          TPWM
976     //
977     //
978     //
979     // En el caso de 5 y 6 TBPRD=17858 y TPWM=1/FrecPWM ----->FrecPWM
980     // =4199,7984096763355358942770747004
981     // En el caso de 1,2 y 3 TBPRD=2000 ----->FrecPWM=586 <-----revisar si funca mejor con
982     //   frec 1176
983     //
984 //-----
985 // >>>>> CONFIGURACIÓN EPWM5 <<<<<<
986 //-----
987     // Setup TBCLK
988     EPwm5Regs.TBPRD = EPWM_TIMER_TBPRD; // Fijar período del PWM
989     EPwm5Regs.TBCTR = 0x0000; // Clear counter
990     // Setup counter mode
991     EPwm5Regs.TBCTL.bit.CTRMODE = 2; // Modo simétrico (ie contador triangular)
992     EPwm5Regs.TBCTL.bit.HSPCLKDIV = 0; // TBLK = SYSCLKOUT/(CLKDIV x HSPCLKDIV) este registro es
993     // pa tener un control más fino de la frecuencia
994     EPwm5Regs.TBCTL.bit.CLKDIV = 0; // Clock ratio to SYSCLKOUT en este caso: TB_DIV1 -> TBLK
995     // = SYSCLKOUT
996     // Set actions
997     EPwm5Regs.AQCTLB.all = 0x0600;
998     // Interrupt where we will change the Compare Values
999     EPwm5Regs.ETSEL.bit.INTSEL = ET_CTR_ZERO; // Select INT on Zero event
1000     EPwm5Regs.ETSEL.bit.INTEN = 1; // Enable INT
1001     EPwm5Regs.ETPS.bit.INTPRD = ET_1ST; // Genera interrupción cada vez
1002 //-----
1003 // >>>>> CONFIGURACIÓN EPWMS 1,2,3<<<<<
1004 //-----
1005     // Setup TBCLK
1006     EPwm1Regs.TBCTL.bit.CLKDIV = 3;
1007     EPwm1Regs.TBCTL.bit.HSPCLKDIV = 4;
1008     EPwm1Regs.TBCTL.bit.CTRMODE = 2;
1009     EPwm1Regs.AQCTLB.all = 0x0600;
1010     EPwm1Regs.TBPRD = 2000; // aprox 1KHz - PWM
1011     // signal
1012     EPwm2Regs.TBCTL.bit.CLKDIV = 3;
1013     EPwm2Regs.TBCTL.bit.HSPCLKDIV = 4;
1014     EPwm2Regs.TBCTL.bit.CTRMODE = 2;
1015     EPwm2Regs.AQCTLB.all = 0x0600;
1016     EPwm2Regs.TBPRD = 2000;
1017     EPwm3Regs.TBCTL.bit.CLKDIV = 3;
1018     EPwm3Regs.TBCTL.bit.HSPCLKDIV = 4;
1019     EPwm3Regs.TBCTL.bit.CTRMODE = 2;
1020     EPwm3Regs.AQCTLB.all = 0x0600;
1021     EPwm3Regs.TBPRD = 2000;
1022     EPwm4Regs.TBCTL.bit.CLKDIV = 3;
1023     EPwm4Regs.TBCTL.bit.HSPCLKDIV = 4;
1024     EPwm4Regs.TBCTL.bit.CTRMODE = 2;
1025     EPwm4Regs.AQCTLB.all = 0x0600;
1026     EPwm4Regs.TBPRD = 2000;
1027 //-----
1028 // Comienza con CMPB (ciclo de trabajo) en cero con ciclo máximo y puede aumentar hasta el maximo de
1029 // 2000 obteniendo un cero
1030
1031

```

```

1032 //Se debe considerar que la logica es inversa debido a la electrónica con que se implementó el
1033 // prototipo
1034 //-----
1035 EPwm1Regs.CMPB = 0;
1036 EPwm2Regs.CMPB = 0;
1037 EPwm3Regs.CMPB = 0;
1038 EPwm4Regs.CMPB = 0;
1039
1040
1041 EALLOW;
1042 SysCtrlRegs.PCLKCR0.bit.TBCLKSYNC = 1; // Habilitar el reloj del EPWM
1043 EDIS;
1044
1045 }
1046
1047 //-----
1048 // Funciones filtros
1049 //-----
1050 //-----
1051
1052 void Filtrar2Hz(struct Filtrovector *elfiltro ,float Entrada[3]){
1053     elfiltro->Salida[0]=0.00278863132713509*Entrada[0]+0.997211368672865*elfiltro->Salida[0];
1054     elfiltro->Salida[1]=0.00278863132713509*Entrada[1]+0.997211368672865*elfiltro->Salida[1];
1055     elfiltro->Salida[2]=0.00278863132713509*Entrada[2]+0.997211368672865*elfiltro->Salida[2];
1056 }
1057
1058 void Filtrar1HzCont(struct Filtrocontinuo *elfiltro ,float Entrada){
1059     elfiltro->Salida=0.001395289079375*Entrada+0.998604710920625*elfiltro->Salida;
1060 }
1061
1062 void Filtrar5Hz(struct Filtrocontinuo *elfiltro ,float Entrada){
1063     elfiltro->Salida=0.0007477*Entrada+0.9993523*elfiltro->Salida;
1064 }
1065
1066 void Filtrar100Hz(struct Filtrovector *elfiltro ,float Entrada[3]){
1067     elfiltro->Salida[0]=1*Entrada[0];
1068     elfiltro->Salida[1]=1*Entrada[1];
1069     elfiltro->Salida[2]=1*Entrada[2];
1070 //     elfiltro->Salida[0]=0.07207*Entrada[0]+0.9279*elfiltro->Salida[0];
1071 //     elfiltro->Salida[1]=0.07207*Entrada[1]+0.9279*elfiltro->Salida[1];
1072 //     elfiltro->Salida[2]=0.07207*Entrada[2]+0.9279*elfiltro->Salida[2];
1073 }
1074 //-----
1075 // Funciones inicializar
1076 //-----
1077
1078 void InicializaFiltrovector(struct Filtrovector *elfiltro ,float CInicial){
1079     elfiltro->Salida[0]=CInicial;
1080     elfiltro->Salida[1]=CInicial;
1081     elfiltro->Salida[2]=CInicial;
1082 }
1083
1084 void InicializaFiltrocontinuo(struct Filtrocontinuo *elfiltro ,float CInicial){
1085     elfiltro->Salida=CInicial;
1086 }
1087
1088 //-----
1089 // Configuración
1090 //-----
1091
1092 interrupt void SCIA_RX_isr(void){
1093
1094     //indicador2++; //Sólo es un indicador de que se está ingresando a la interrupción
1095
1096     character = SciaRegs.SCIRXBUF.bit.RXDT;
1097
1098     if (character == '#' && contador == 0){
1099         buffer[contador] = character;
1100         contador++;
1101     }
1102     else if (contador != 0){
1103         if ( character == '+' ){
1104             buffer[contador] = character;
1105             contador=0;
1106             //GpioDataRegs.GPBTGGLE.bit.GPIO34 = 1;
1107         }
1108         else{
1109             buffer[contador] = character;
1110             contador++;
1111         }
1112     }
1113
1114     SciaRegs.SCIFFRX.bit.RXFIFORESET = 0; // reset RX-FIFO pointer
1115     SciaRegs.SCIFFRX.bit.RXFIFORESET = 1; // enable RX-operation
1116     SciaRegs.SCIFFRX.bit.RXFFINTCLR = 1; // clear RX-FIFO INT Flag
1117
1118     PieCtrlRegs.PIEACK.all = PIEACK_GROUP9;
1119
1120
1121
1122
1123 } //Close SCIA_RX_isr()
1124
1125 //-----

```

```

1126 // Funcion modificación ciclo PWMS
1127 //-----
1128 void Changeduty(int PWM, int dutycycle){
1129     if (PWM==1){// PWM corriente campo
1130         EPwm1Regs.CMPB = dutycycle;           // duty cycle %
1131         return;
1132     }
1133     if (PWM==2){//PWM corriente desahogo A
1134         EPwm2Regs.CMPB = dutycycle;           // duty cycle %
1135         return;
1136     }
1137     if (PWM==3){//PWM corriente desahogo B
1138         EPwm3Regs.CMPB = dutycycle;           // duty cycle %
1139         return;
1140     }
1141     if (PWM==4){//PWM corriente desahogo C
1142         EPwm4Regs.CMPB = dutycycle;           // duty cycle %
1143         return;
1144     }
1145 }
1146 //Close Changeduty()
1147
1148 float Absoluto(float f){
1149     if(f>=0.0){
1150         return f;
1151     }else{
1152         return -f;
1153     }
1154 }
1155
1156 //-----
1157 // FIN
1158 //-----
1159

```

A.2. Configuración de ADC

```

1  #include "DSP2833x_Device.h"      // DSP2833x Header file Include File
2
3  /*
4  *ADCTRL1, ADCTRL2 y ADCTRL3 register set up
5  *
6  * SEQUENCER MODE: Selects whether we use state machine of Auto Sequencer as a single 16 stage state
7  * machine ("Cascade Mode")
8  * or as a pair of two independent 8-stage measurement unit ("Dual sequencer Mode").
9  *
10 * SAMPLING MODE: Convert 2 analog input signals at one time in "Simultaneous Sampling". If we choose "
11 * Sequential Sampling"
12 * only one multiplexed input channel is converted at one time.
13 *
14 * START MODE: Selecting "Single Sequence Mode" (or "Start/Halt - Mode") the auto sequencer starts at the
15 * first input trigger
16 * signal, performs the predefined number of conversions and stops at the end.
17 *
18 */
19
20 void ConfigADCregister(void)
21 {
22     AdcRegs.ADCTRL1.all = 0;          // Set ADCTRL1 in Zero
23     /*
24     bit 15          x: Reserved      (Write no have effect)
25     bit 14          0: RESET          Reset.
26     bit 13-12      00: SUSMOD          Emulation suspend Mode          00 = Emulation
27     bit 11-8       111: ACQ_PS        Acquisition window size      Acquisition time =
28     bit 7          0: CPS             Core Clock Prescaler          0 = ADCCLK=F_clk/1,
29     bit 6          1 = ADCCLK=F_clk/2
30     bit 5          0: CONTRUN         Continuous run                0 = Stop
31     bit 4          0: SEQ_OVRD        Sequencer Override            0 =
32     bit 3-0        1: SEQ_CASC        Sequencer Mode                0 =
33     bit 3-0        xxx: Reserved      (Write no have effect)
34     */
35     AdcRegs.ADCTRL1.bit.ACQ_PS       = 0xf; //0x7
36     AdcRegs.ADCTRL1.bit.CPS          = 0;
37     AdcRegs.ADCTRL1.bit.CONTRUN      = 0;
38     AdcRegs.ADCTRL1.bit.SEQ_CASC     = 1;
39
40     AdcRegs.ADCTRL2.all = 0;
41     /*
42     bit 15          0: ePWM_SOCB_SEQ1 ePWM SOCB enable for cascade sequencer
43     bit 14          0 = No action, 1 = Allows the cascade seq to be started by an ePWM SOCB
44     bit 13          0: RST_SEQ1       Reset sequencer 1
45     bit 12          0 = No action, 1 = Immediate reset SEQ1 to "
46     bit 11          initial state"
47     bit 10          0: SOC_SEQ1       SOC trigger for seq 1
48     bit 9           0 = Clear a pending SOC trigger, 1 = Software trigger - Satart SEQ1 from
49     bit 8           currently stopped position
50     bit 7           x: Reserved
51     bit 6           1: INT_ENA_SEQ1   SEQ1 interrupt enable
52     bit 5           0 = Interrupt request by INT_SEQ1 is disabled, 1 = Interrupt request by
53     bit 4           INT_SEQ1 is enabled
54     bit 3           0: INT_MOD_SEQ1   SEQ1 interrupt mode
55     bit 2           0 = INT_SEQ1 is set at the end of every SEQ1 sequence, 1 =
56     bit 1           INT_SEQ1 is set at the end of every other SEQ1 seq
57     bit 0           x: Reserved
58     bit 0           1: ePWM_SOCA_SEQ1 ePWM SOCA enable bit for SEQ1
59     bit 0           0 = SEQ1 cannot be started by ePWMx SOCA trigger, 1 = Allows SEQ1/SEQ2 to
60     bit 0           be started by ePWMx SOCA trigger
61     bit 0           0: EXT_SOC_SEQ1   External signal SOC for SEQ1
62     bit 0           0 = No action, 1 = Enable ADC autoconversion seq by a signal from GPIO
63     bit 0           port A, pin GPIO31-0
64     bit 0           0: RST_SEQ2       Reset Sequencer 2
65     bit 0           0 = No action, 1 = Immediate reset SEQ2 to "
66     bit 0           initial state"
67     bit 0           0: SOC_SEQ2       SOC trigger for seq 2
68     bit 0           0 = Clear a pending SOC trigger, 1 = Software trigger -
69     bit 0           Satart SEQ2 from currently stopped position
70     bit 0           0: Reserved
71     bit 0           0: INT_ENA_SEQ2   SEQ2 interrupt enable
72     bit 0           0 = Interrupt request by INT_SEQ2 is disabled, 1 = Interrupt
73     bit 0           request by INT_SEQ2 is enabled
74     bit 0           0: INT_MOD_SEQ2   SEQ2 interrupt mode
75     bit 0           0 = INT_SEQ2 is set at the end of every SEQ2 sequence, 1 =
76     bit 0           INT_SEQ2 is set at the end of every other SEQ2 seq
77     bit 0           0: Reserved

```

```

57      bit 0          0: ePWM_SOCB_SEQ2          ePWM SOCA enable bit for SEQ2
          0 = SEQ2 cannot be started by ePWMx SOCB trigger, 1 = Allows SEQ2 to be
          started by ePWMx SOCB trigger
58  */
59  AdcRegs.ADCCTRL2.bit.INT_ENA_SEQ1          = 1;
60  AdcRegs.ADCCTRL2.bit.EPWM_SOCA_SEQ1       = 1;
61  AdcRegs.ADCCTRL2.bit.INT_MOD_SEQ1         = 0;
62  AdcRegs.ADCCTRL2.bit.RST_SEQ1 = 0x1;
63
64
65  AdcRegs.ADCCTRL3.bit.ADCCLKPS = 1;
66  /*
67      bit 15-8          xxxxxxxx: Reserved
68      bit 7-6          00: ADCBGRFDN          ADC Bandgap and Reference Power Down
          00 = Powered down, 11 = Powered up
69      bit 5            0: ADCPWDN            ADC power Down
          0 = Powered down, 1 = Powered up
70      bit 4-1          0000: ADCCLKPS        ADC Clock Prescale
          0 = FCLK=HSPCLK, 1 to F = FCLK=HSPCLK/(2*ADCCLKPS)
71      bit 0            0: SMODE_SEL          Sampling Mode Select
          0 = Sequential sampling mode, 1 = Simultaneous sampling mode
72  */
73
74  AdcRegs.ADCMAXCONV.all = 15;
75  /*
76      bit 15-7          xxxxxxxx: Reserved
77      bit 6-4          000: MAX_CONV2        Define the maximum number of conversion
          executed en autoconversion, For SEQ2 operation bits MAX_CONV2[2:0] are used
78      bit 3-0          000: MAX_CONV1        For SEQ1 operation bits MAX_CONV1[2:0] are
          used and For SEQ operation bits MAX_CONV1[3:0] are used
79  */
80
81
82
83  AdcRegs.ADCCHSELSEQ1.bit.CONV00 = 0; // Setup ADCINA0 as 1st SEQ1 conv.
84  AdcRegs.ADCCHSELSEQ1.bit.CONV01 = 0x8;
85  AdcRegs.ADCCHSELSEQ1.bit.CONV02 = 0x1;
86  AdcRegs.ADCCHSELSEQ1.bit.CONV03 = 0x9;
87  AdcRegs.ADCCHSELSEQ2.bit.CONV04 = 0x2;
88  AdcRegs.ADCCHSELSEQ2.bit.CONV05 = 0xa;
89  AdcRegs.ADCCHSELSEQ2.bit.CONV06 = 0x3;
90  AdcRegs.ADCCHSELSEQ2.bit.CONV07 = 0xb;
91  AdcRegs.ADCCHSELSEQ3.bit.CONV08 = 0x4;
92  AdcRegs.ADCCHSELSEQ3.bit.CONV09 = 0xc;
93  AdcRegs.ADCCHSELSEQ3.bit.CONV10 = 0x5;
94  AdcRegs.ADCCHSELSEQ3.bit.CONV11 = 0xd;
95  AdcRegs.ADCCHSELSEQ4.bit.CONV12 = 0x6;
96  AdcRegs.ADCCHSELSEQ4.bit.CONV13 = 0xe;
97  AdcRegs.ADCCHSELSEQ4.bit.CONV14 = 0x7;
98  AdcRegs.ADCCHSELSEQ4.bit.CONV15 = 0xf;
99
100  /*
101  ADCCHSELSEQ1
102      bit 15-12          0x9: CONV03
103      bit 11-8           0x1: CONV02
104      bit 7-4            0x8: CONV01
105      bit 3-0            0x0: CONV00
106
107  ADCCHSELSEQ2
108      bit 15-12          0xb: CONV03
109      bit 11-8           0x3: CONV02
110      bit 7-4            0xa: CONV01
111      bit 3-0            0x2: CONV00
112
113  ADCCHSELSEQ3
114      bit 15-12          0xd: CONV03
115      bit 11-8           0x5: CONV02
116      bit 7-4            0xc: CONV01
117      bit 3-0            0x4: CONV00
118
119  ADCCHSELSEQ4
120      bit 15-12          0xf: CONV03
121      bit 11-8           0x7: CONV02
122      bit 7-4            0xe: CONV01
123      bit 3-0            0x6: CONV00
124
125
126  CONVnn Value      ADC input channel selected
127      0000          ADCINA0
128      0001          ADCINA1
129      0010          ADCINA2
130      0011          ADCINA3
131      0100          ADCINA4
132      0101          ADCINA5
133      0110          ADCINA6
134      0111          ADCINA7
135      1000          ADCINB0
136      1001          ADCINB1
137      1010          ADCINB2
138      1011          ADCINB3
139      1100          ADCINB4
140      1101          ADCINB5
141      1110          ADCINB6
142      1111          ADCINB7
143  */

```

144
145

}

A.3. Configuración de puertos digitales

```

1  #include "DSP2833x_Device.h"    // DSP2833x Header file Include File
2
3
4
5  void ConfigGPIOregister(void){
6
7      EALLOW;
8
9      //General configuration as general purpose I/O
10
11      GpioCtrlRegs.GPAMUX1.all = 0;
12      /*
13       * bit 31-0          0: GPIO 15 ... GPIO 0   Configure GPIO like general purpose I/O
14       */
15      GpioCtrlRegs.GPAMUX2.all = 0;
16      /*
17       * bit 31-0          0: GPIO 31 ... GPIO 16  Configure GPIO like general purpose
18       * I/O
19       */
20      GpioCtrlRegs.GPBMUX1.all = 0;           // GPIO47 ... GPIO32 = General Purpose I/O
21      GpioCtrlRegs.GPBMUX2.all = 0;           // GPIO63 ... GPIO48 = General Purpose I/O
22      GpioCtrlRegs.GPCMUX1.all = 0;           // GPIO79 ... GPIO64 = General Purpose I/O
23      GpioCtrlRegs.GPCMUX2.all = 0;           // GPIO87 ... GPIO80 = General Purpose I/O
24
25      //GpioCtrlRegs.GPAMUX2.bit.GPIO31 = 0; //led
26      //GpioCtrlRegs.GPBMUX1.bit.GPIO34 = 0; //led
27
28      //Particular configuration
29
30      GpioCtrlRegs.GPAMUX2.bit.GPIO28 = 1;
31      GpioCtrlRegs.GPAMUX2.bit.GPIO29 = 1;
32      /*
33       * bit 27-26          01: GPIO 29   Configure GPIO like SCITXDA
34       * bit 25-24          01: GPIO 28   Configure GPIO like SCIRXDA
35       */
36
37      GpioCtrlRegs.GPAMUX1.bit.GPIO1 = 1;     // PWM Campo
38      GpioCtrlRegs.GPAMUX1.bit.GPIO3 = 1;     // PWM Resistencia desahogo fase A
39      GpioCtrlRegs.GPAMUX1.bit.GPIO5 = 1;     // PWM Resistencia desahogo fase B
40      GpioCtrlRegs.GPAMUX1.bit.GPIO7 = 1;     // PWM Resistencia desahogo fase C
41
42      //Define as input or output
43
44      GpioCtrlRegs.GPADIR.all = 0;
45      /*
46       * bit 31-0          0: GPIO 31-0   0 = Configure GPIO as input          1 =
47       * Configure GPIO as output
48       */
49      GpioCtrlRegs.GPADIR.bit.GPIO0 = 1;       // output Reles 220 inversor
50      GpioCtrlRegs.GPAPUD.bit.GPIO0 = 1;       // 0 = Pull up on, 1 = Pull up off
51      GpioDataRegs.GPASET.bit.GPIO0 = 1;       // 0 = ignored, 1 = output high
52
53      GpioCtrlRegs.GPADIR.bit.GPIO1 = 1;       // output PWM1B Campo
54      //GpioCtrlRegs.GPADIR.bit.GPIO2 = 1;     // output PWM Extra
55      GpioCtrlRegs.GPADIR.bit.GPIO3 = 1;       // output PWM2B Resistencia desahogo fase
56      A
57      //GpioCtrlRegs.GPADIR.bit.GPIO4 = 1;     // output PWM Extra
58      B
59      //GpioCtrlRegs.GPADIR.bit.GPIO5 = 1;     // output PWM3B Resistencia desahogo fase
60      C
61      GpioCtrlRegs.GPADIR.bit.GPIO6 = 1;       // output PWM Extra
62      GpioCtrlRegs.GPADIR.bit.GPIO7 = 1;       // output PWM4B Resistencia desahogo fase
63
64      GpioCtrlRegs.GPADIR.bit.GPIO8 = 1;       // output Reles Motor Valvula
65      GpioCtrlRegs.GPAPUD.bit.GPIO8 = 1;       // 0 = Pull up on, 1 = Pull up off
66      GpioDataRegs.GPASET.bit.GPIO8 = 1;       // 0 = ignored, 1 = output pin high
67
68      GpioCtrlRegs.GPADIR.bit.GPIO9 = 1;       // PWM5B
69      GpioCtrlRegs.GPADIR.bit.GPIO11 = 1;      // PWM6B
70
71      GpioCtrlRegs.GPADIR.bit.GPIO10 = 1;      // output GPIO extra
72      GpioCtrlRegs.GPADIR.bit.GPIO12 = 1;      // output GPIO extra
73      GpioCtrlRegs.GPADIR.bit.GPIO15 = 1;      // output GPIO Led 1
74      GpioCtrlRegs.GPADIR.bit.GPIO16 = 1;      // output GPIO Led 3
75
76      GpioCtrlRegs.GPADIR.bit.GPIO25 = 0;      // in GPIO Fin carrera
77
78      GpioCtrlRegs.GPADIR.bit.GPIO26 = 1;      // output GPIO Led 2
79      GpioCtrlRegs.GPADIR.bit.GPIO27 = 0;      // in GPIO Fin carrera
80      GpioCtrlRegs.GPADIR.bit.GPIO31 = 1;      // output LED DSC
81
82      GpioCtrlRegs.GPBDIR.all = 0;             // GPIO63-32 as inputs
83      GpioCtrlRegs.GPBDIR.bit.GPIO34 = 1;     // output LED DSC
84
85      GpioCtrlRegs.GPBDIR.bit.GPIO48 = 1;     // output GPIO extra abrir cerrar valvula
86      GpioCtrlRegs.GPBPUd.bit.GPIO48 = 1;     // 0 = Pull up on, 1 = Pull up off
87      GpioDataRegs.GPASET.bit.GPIO48 = 1;     // 0 = ignored, 1 = output pin high

```



```

86
87     GpioCtrlRegs.GPBDIR.bit.GPIO59 = 1;
88     GpioCtrlRegs.GPBPUD.bit.GPIO59 = 1;
89     GpioDataRegs.GPASET.bit.GPIO59 = 1;
90
91     GpioCtrlRegs.GPBDIR.bit.GPIO58 = 1;
92     GpioCtrlRegs.GPBPUD.bit.GPIO58 = 1;
93     GpioDataRegs.GPASET.bit.GPIO58 = 1;
94
95     GpioCtrlRegs.GPBDIR.bit.GPIO60 = 1;
96     GpioCtrlRegs.GPBPUD.bit.GPIO60 = 1;
97     GpioDataRegs.GPASET.bit.GPIO60 = 1;
98
99     GpioCtrlRegs.GPBDIR.bit.GPIO61 = 1;
100    GpioCtrlRegs.GPBPUD.bit.GPIO61 = 1;
101    GpioDataRegs.GPASET.bit.GPIO61 = 1;
102
103
104
105
106    GpioCtrlRegs.GPCDIR.all = 0;
107    GpioCtrlRegs.GPCDIR.bit.GPIO84 = 1;
108    GpioCtrlRegs.GPCDIR.bit.GPIO86 = 1;
109
110
111    EDIS;
112
113
114
115 }

```

A.4. Configuración puerto serial

```

1  #include "DSP2833x_Device.h"    // DSP2833x Header file Include File
2
3  void ConfigSERIALregister()
4  {
5      SciaRegs.SCICCR.all =0x0007;
6
7      /*
8      bit 7 0: STOP bit        SCI number of stop bit 0 = One stop bit, 1 = Two stop bit
9      bit 6 0: EVEN/ODD PARITY  SCI parity odd/even selection 0 = Odd parity, 1 = Even parity
10     bit 5 0: PARITY ENABLE    SCI parity enable 0 = Parity disabled, 1 = Parity enabled
11     bit 4 0: LOOP BACK ENA    Loop back test enable 0 = Loop back test disabled, 1 = Loop
12     back test enabled
13     bit 3 0: ADDR/IDLE MODE    SCI multiprocessor mode control bit 0 = Idle line mode protocol
14     selected, 1 = Address bit mode protocol selected
15     bit 2-0 111: SCI CHAR      Character lenght control bits 111 = Character length (bits)=8,
16     ... , 000 = Character length (bits)=1
17     */
18     SciaRegs.SCICTL1.all =0x0023;
19
20     /*
21     bit 7 0: Reserved
22     bit 6 0: RX ERR INT ENA    SCI receive error interrupt enable 0 = Receive error interrupt
23     disabled, 1 = Receive error interrupt enabled
24     bit 5 1: SW RESET          SCI software reset 0 = Initializes SCI state machine, 1 = Re-
25     enable SCI
26     bit 4 0: Reserved
27     bit 3 0: TXWAKE            SCI transmitter wake up method selected 0 = Transmit feature is not
28     selected, 1 = Transmit feature selected is dependent on the mode, idle-line or address bit
29     bit 2 0: SLEEP             SCI Sleep 0 = Seelp mode disabled, 1 = Sleep mode enabled
30     bit 1 1: TXENA             SCI transmitter enabled 0 = Transmitter disabled, 1 = Transmitter
31     enabled
32     bit 0 1: RXENA             SCI receiver enabled 0 = Prevent received characters from transfer into
33     the SCIRXEMU and SCIRXBUF receiver buffers, 1 = Send received characters to SCIRXEMU and
34     SCIRXBUF
35     */
36
37     //SciaRegs.SCIHBAUD = 40 >> 8; // Highbyte
38     SciaRegs.SCIHBAUD = 0x0000;
39     SciaRegs.SCILBAUD = 40 & 0x00FF; // Lowbyte
40     //SciaRegs.SCILBAUD = 0x27;
41
42     /*
43     bit 15-0 40: BAUD15-BAUD0  SCI 16-bit baud selection Registers SCIHBAUD (MSbyte) and
44     SCILBAUD (LSbyte) are concatenated to form a 16-bit baud value, BRR.
45     The SCI baud rate is calculated using the following equation:
46     SCI Asynchronous Baud = LSPCLK / ((BRR + 1)*8)
47     115200 = 37.5MHz / (BRR + 1)*8 => BRR = 39.69 == 40 (No es un cálculo excato)
48     */
49
50     SciaRegs.SCICTL2.bit.RXBKINTENA = 1;
51
52     /*
53     bit 7 0: TXRDY            Transmitter buffer register ready flag 0 = SCITXBUF is full, 1 =
54     SCITXBUF is ready to receive the next character
55     bit 6 0: TX EMPTY          Transmitter empty flag 0 = Transmitter buffer or shift register or both
56     are loaded with data
57     bit 5-2 xxxxx: Reserved
58     bit 1 1: RX/BK INT ENA     Receiver buffer/break interrupt enabled 0 = Disable RXRDY/BRKDT
59     interrupt, 1 = Enable RXRDY/BRKDT interrupt
60     bit 0 0: TX INT ENA       SCITXBUF register interrupt enable 0 = Disable TXRDY interrupt, 1 =
61     Enable TXRDY interrupt
62     */
63
64     SciaRegs.SCIFFTX.all=0xC024;//El FIFO de salida tiene 4 palabras de 8 bits
65     SciaRegs.SCIFFTX.bit.TXFIFOXRESET=1;
66
67     /*
68     bit 15 1: SCIRST           SCI Reset
69     channels, 1 = SCI FIFO can resume transmit or receive 0 = Reset SCI transmit and receive
70     bit 14 1: SCIFFENA         SCI FIFO enable
71     enhancements are enabled 0 = SCI FIFO enhancements are disabled, 1 = SCI FIFO
72     bit 13 1: TXFIFO Reset     Transmit FIFO reset
73     transmit FIFO operation 0 = Reset the FIFO pointer to zero and hold reset, 1 = Re-enabled
74     bit 12-8 00000: TXFFST4-0 0000 = Transmit FIFO is empty
75     bit 7 0: TXFFINT Flag      Transmit FIFO interrupt
76     interrupt has occurred, read-only bit 0 = TXFIFO interrupt has not occurred, read only bit, 1 = TXFIFO
77     bit 6 0: TXFFINT CLR       Transmit FIFO clear
78     0 = Has no effect on TXFIFINT flag bit, 1 = Clear TXFFINT flag in
79     bit 7
80     bit 5 1: TXFFIENA          Transmit FIFO interrupt enable
81     0 = TX FIFO interrupt based on TXFFIVL match is disabled, 1 = TX
82     FIFO interrupt based on TXFFIVL match is enabled
83     bit 4-0 00100: TXFFIL4-0   Transmit FIFO interrupt level bits
84     Transmit FIFO will generate interrupt when the FIFO status bits (TXFFST4-0)
85     and FIFO level bits (TXFFIL4-0) match (less than or equal to). Default value should be 0

```

```

62         x00000
63     */
64     SciaRegs.SCIFFRX.all=0x0021;//El FIFO de entrada tiene 1 palabras (8 bits)
65     SciaRegs.SCIFFRX.bit.RXFIFORESET=1;
66
67     /*
68     bit 15          0: RXFFOVF                      Receive FIFO overflow
69     = Receive FIFO has overflow, read only bit. More than 16 word have been received
70     bit 14          0: RXFFOVF CLR                  RXFFOVF clear
71     back a zero, 1 = Clear RXFFOVF flag bit 15
72     bit 13          0: RXFIFO Reset                Receive FIFO reset
73     = Reset the FIFO pointer to zero, and hold in reset, 1 = Re-
74     enabled receive FIFO operation
75     bit 12-8 00000: RXFFST4-0
76     0000 = Receive FIFO is empty
77     bit 7           0: RXFFINT Flag                Receive FIFO interrupt
78     = RXFIFO interrupt has not occurred, read only bit, 1 = RXFIFO
79     interrupt has occurred, read-only bit
80     bit 6           0: RXFFINT CLR                Receive FIFO clear
81     = Has no effect on RXFIFINT flag bit, 1 = Clear RXFFINT flag in
82     bit 7
83     bit 5           1: RXFFIENA                    Receive FIFO interrupt enable
84     = RX FIFO interrupt based on RXFFIVL match is disabled, 1 = RX FIFO
85     interrupt based on RXFFIVL match is enabled
86     bit 4-0 00001: RXFFIL4-0                    Receive FIFO interrupt level bits
87     Receive FIFO will generate interrupt when the FIFO status bits (RXFFST4
88     -0) and FIFO level bits (RXFIL4-0) match (less than or equal to). Default value should be
89     11111
90     */
91     SciaRegs.SCIFFCT.all=0x00;
92 }

```

Apéndice B

Documentación electrónica

En este anexo se presenta el desarrollo electrónico de la microcentral. Se exponen cada una de las placas que forman parte del proyecto, mostrando el diseño realizado y el resultado obtenido. Además se indican comentarios para la conexión y utilización de las mismas.

Se espera sea un documento útil para poder entender, continuar y mejorar el trabajo realizado hasta el momento.

B.1. Cúpula

B.1.1. DSC

En circuito para utilización del DSC considera el conector para la tarjeta del micro controlador, un bloque de alimentación, conectores para utilización de los periferales del DSC, componentes necesarios para utilización de dos puertos CAN, 3 puertos RS232, 1 JTAG, I2C y XBee.

A continuación se puede apreciar el esquemático con los distintos bloques enunciados.

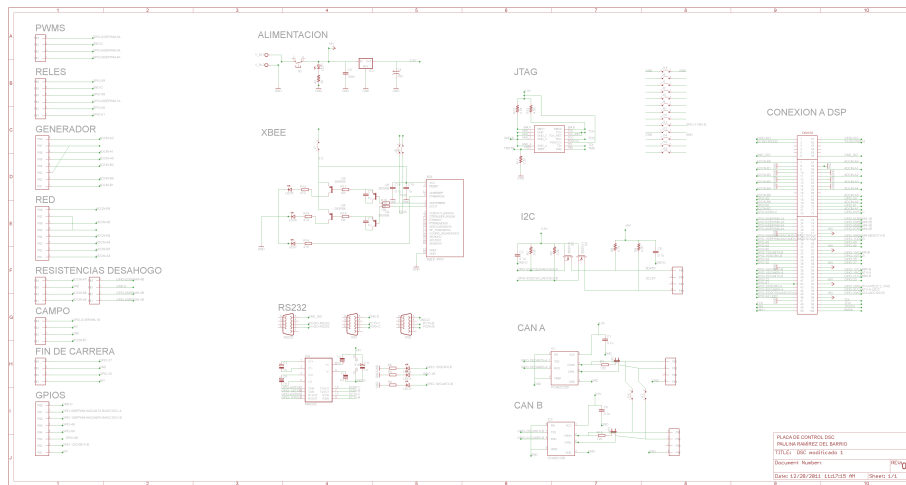


Figura B.1: Esquemático de placa para DSC

Se realiza el diseño de la placa considerando las recomendaciones de Texas Instruments

1

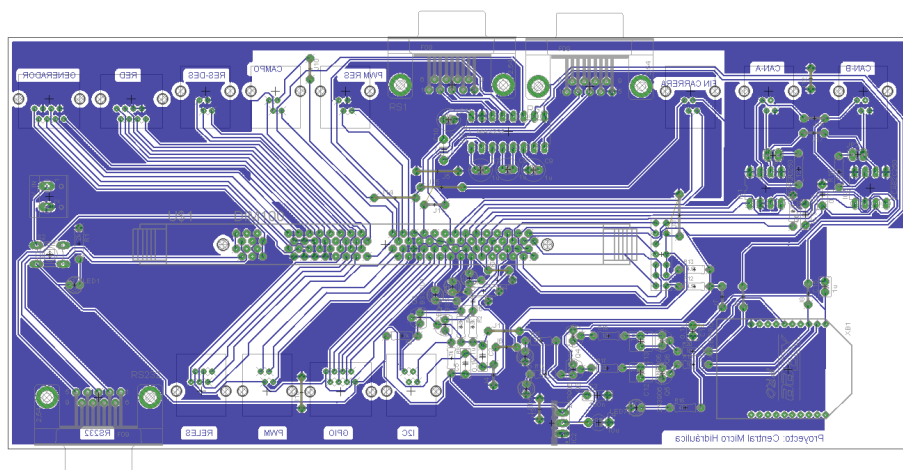


Figura B.2: Diseño de placa para DSC

Se elabora la placa en una faz y se soldan los componentes obteniendo el resultado expuesto en la siguiente imagen.

¹ Hardware Design Guidelines for TMS320F28xx and TMS320F28xxx DSCs

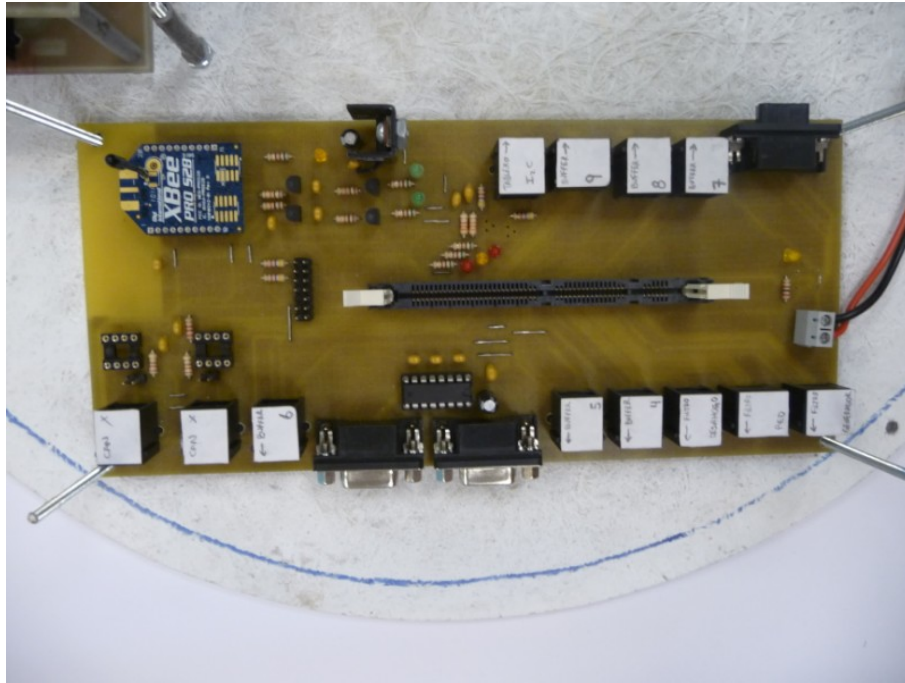


Figura B.3: Placa para DSC

B.1.2. Buffer

Para utilización de las salidas digitales del DSC es necesario amplificarlas por lo que diseña una placa con transistores para amplificar todas las salidas. Se consideran como entrada los mismos conectores de la placa del DSC y como salida los conectores necesarios para conexión con las placas correspondientes a las siguientes etapas.

Los transistores utilizan como alimentación la misma que utiliza el DSC, por lo que es necesario re-diseñar la placa o bien modificar las conexiones de los cables entre los conectores.

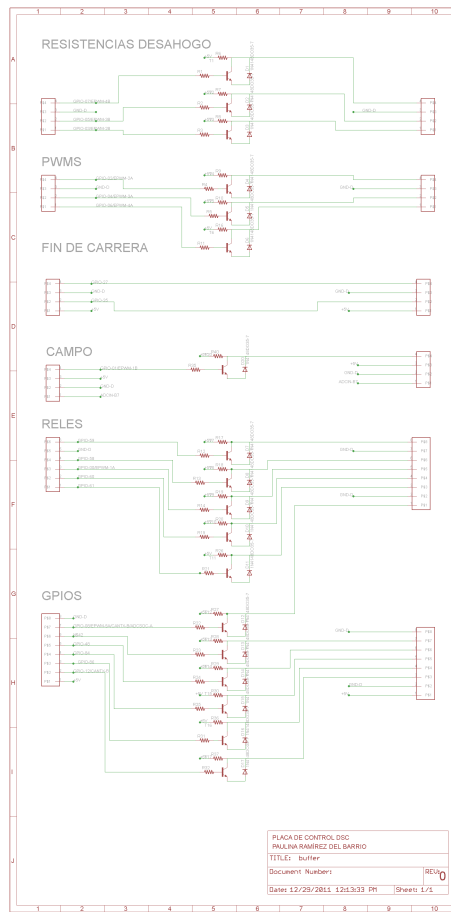


Figura B.4: Esquemático de placa de buffers

Por consideraciones de espacio se realiza el diseño lo más compacto posible.

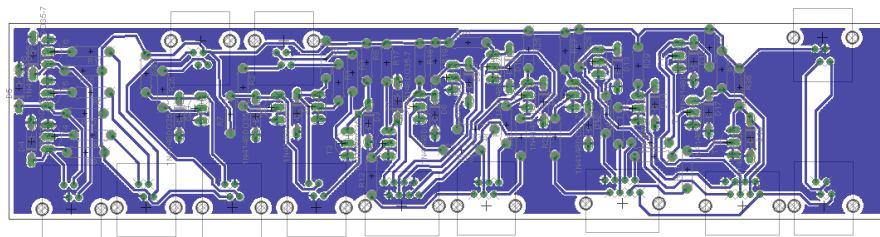


Figura B.5: Diseño de placa de buffers

Se obtiene el siguiente resultado.

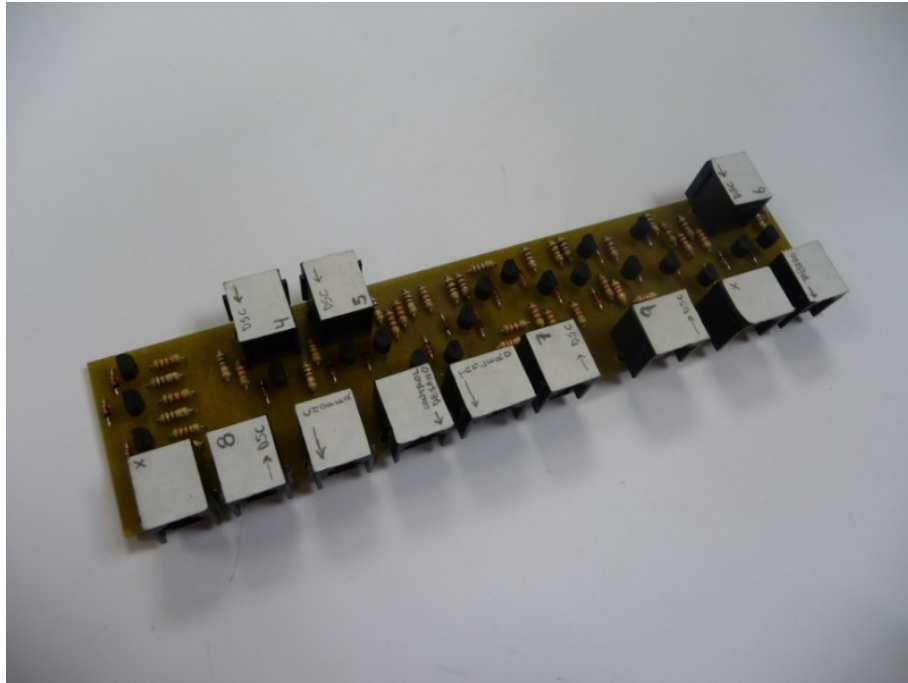


Figura B.6: Placa de buffers

En la práctica se deciden utilizar algunos periferales como entradas y no salidas, por lo que no es necesario conectarlas a esta placa, quizás sea necesario realizar una nueva placa de adaptación para que el DSC no se vea afectado por tensiones superiores a las recomendadas. Además se logra apreciar que debido a que es alimentada desde el DSC, si se prende el sistema con esta placa conectada, el programa se reinicia, no pudiendo ser ejecutado. Por ello es necesario utilizar otra alimentación proveniente de la placa de suministro.

B.1.3. Suministro

Para alimentar todos los circuitos se diseña la placa de suministro, que considera conectores para todos los circuitos.

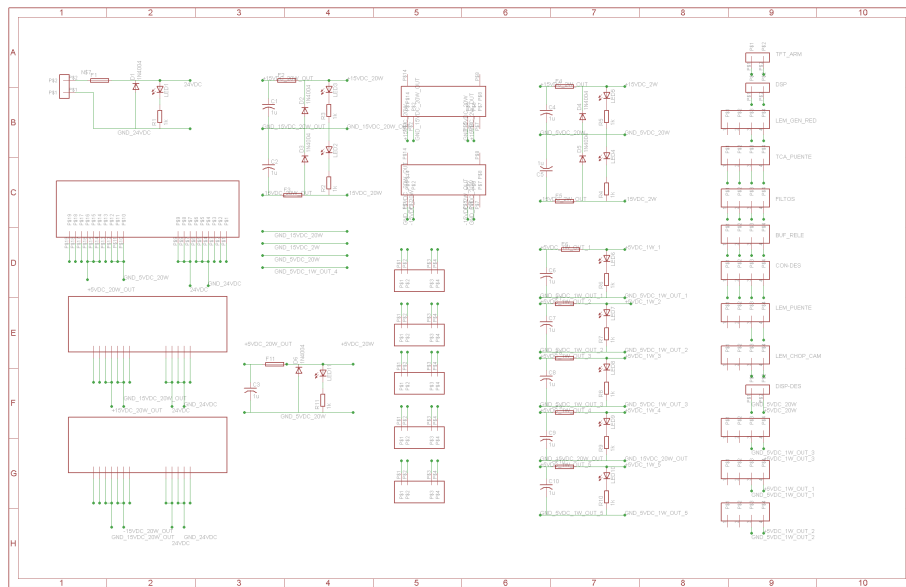


Figura B.7: Esquemático de placa de suministro

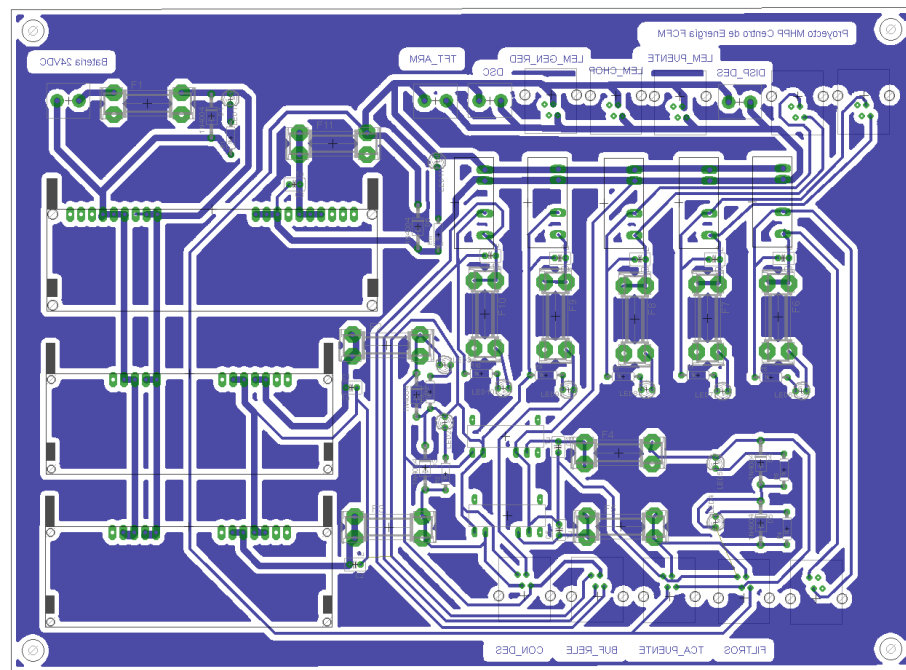


Figura B.8: Diseño de placa de suministro

En la placa se debieron hacer modificaciones que se encuentran incorporadas en los diseños anteriores y se aprecia en el resultado final la utilización de leds indicadores de que las distintas fuentes de alimentación están funcionando.

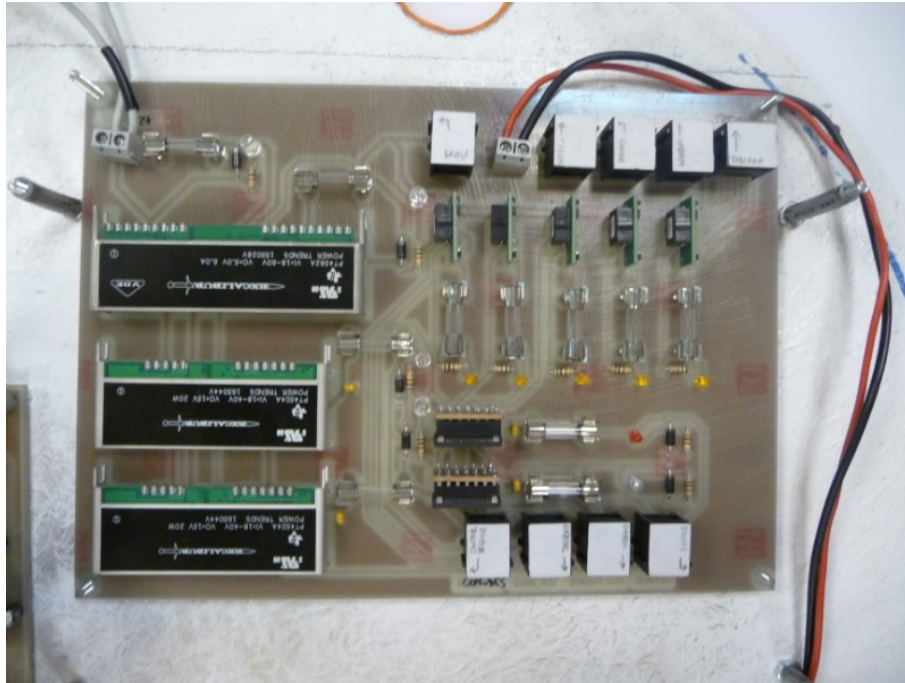


Figura B.9: Placa de suministro

B.1.4. Filtros

Para poder realizar las mediciones de variables necesarias, se debe considerar filtrar las señales para eliminar ruidos no deseados, por ello Richard Dederich dimensionó el filtro a utilizar, y su cálculo fue el utilizado para diseñar los circuitos.

Es necesario filtrar las mediciones provenientes de la red (3 de corriente y 3 de voltaje), del generador (6 mediciones) y las corrientes de las resistencias de desahogo (3 mediciones). Por lo que se diseñan dos placas de filtros. La primera considera sumadores para las mediciones de voltaje y referencias de tensión para ajustar las señales, ésta es replicada para red y generador. Ambas placas tienen un regulador de tensión para la alimentación de los opamps y son alimentadas desde la placa de suministro.

Las primeras 3 figuras corresponden a la placa replicada.

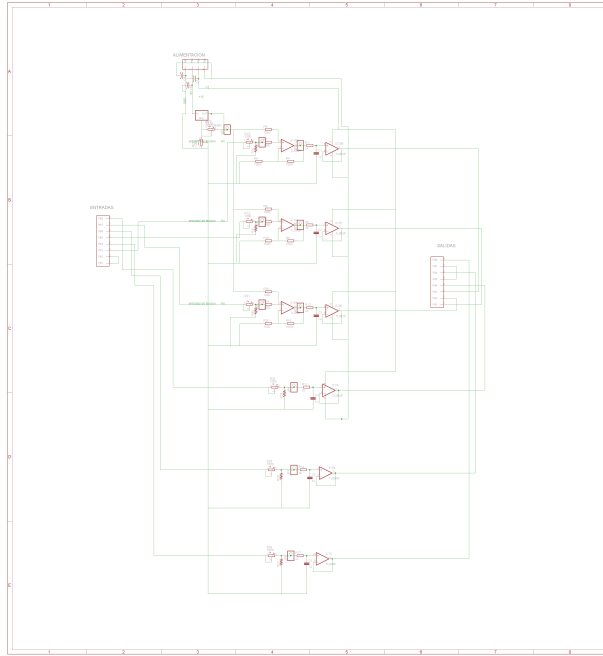


Figura B.10: Esquemático de placa de filtros

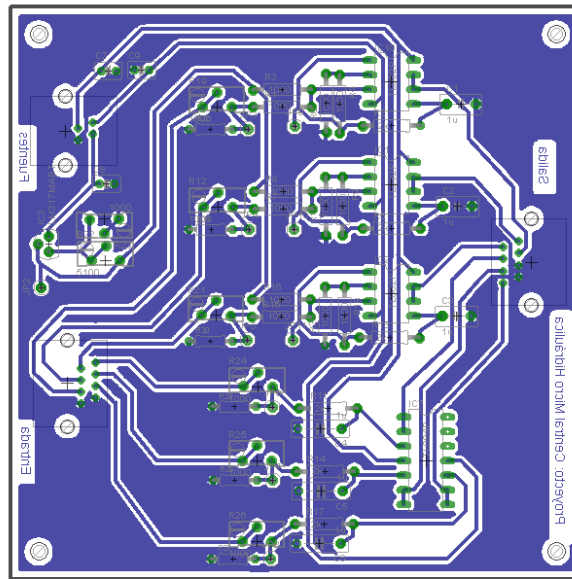


Figura B.11: Diseño de placa de filtros

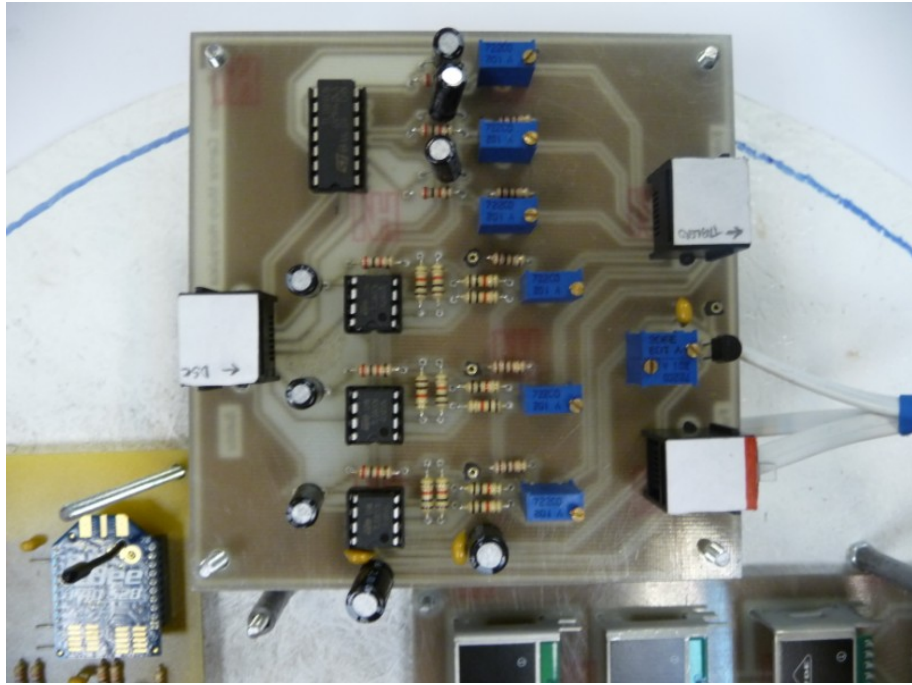


Figura B.12: Placa de filtros

Los siguientes diseños corresponden a la placa de filtro para las resistencias de desahogo.

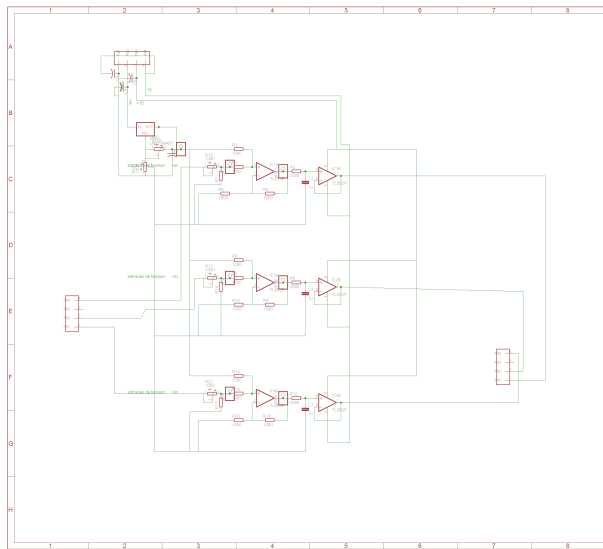


Figura B.13: Esquemático de placa de filtros

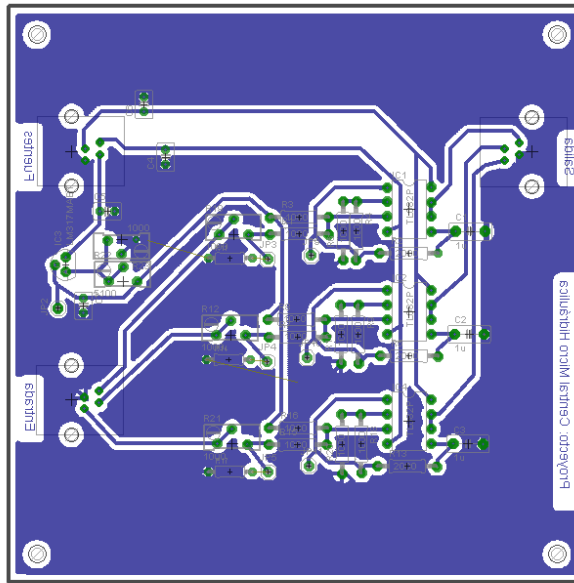


Figura B.14: Diseño de placa de filtros

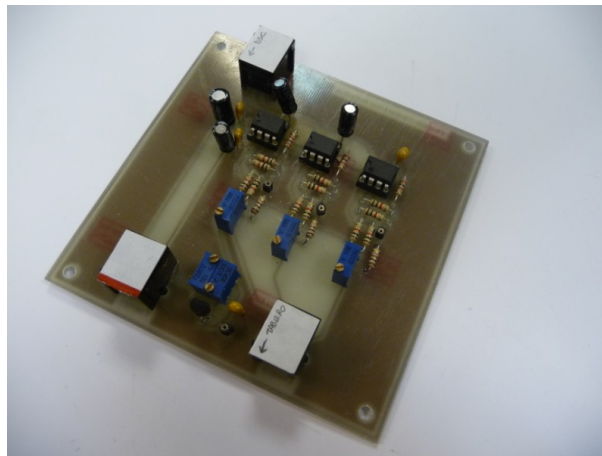


Figura B.15: Placa de filtros

B.1.5. Disparo resistencias de desahogo (circuito 1)

Para poder utilizarlas resistencias de desahogo es necesario controlar el disparo de los triacs encargados de su utilización. Se diseña un circuito que reciba los PWMs del DSC y los sincronice con las señales provenientes de las resistencias. Se optoacopla la entrada desde el DSC, luego se filtra, se pasa a continua para controlar los TCA, los TCA sincronizan la señal y generan los pulsos acordes a la continua proveniente de los PWMs, se amplifica y pasa al siguiente bloque.

Control de disparo resistencias de desahogo - MHPP

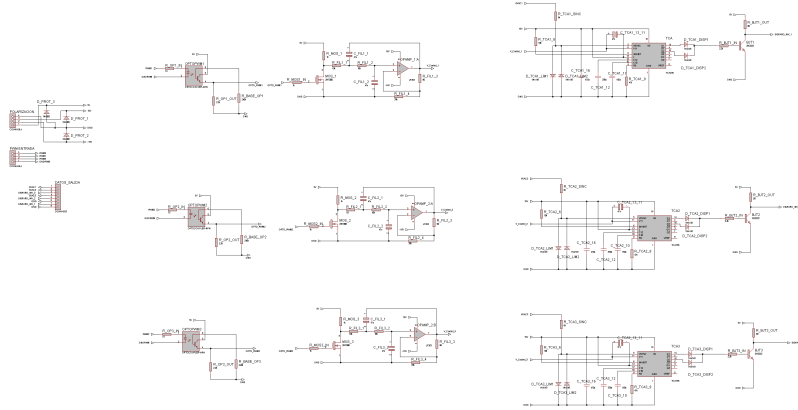


Figura B.16: Esquemático de placa de disparo de resistencias de desahogo

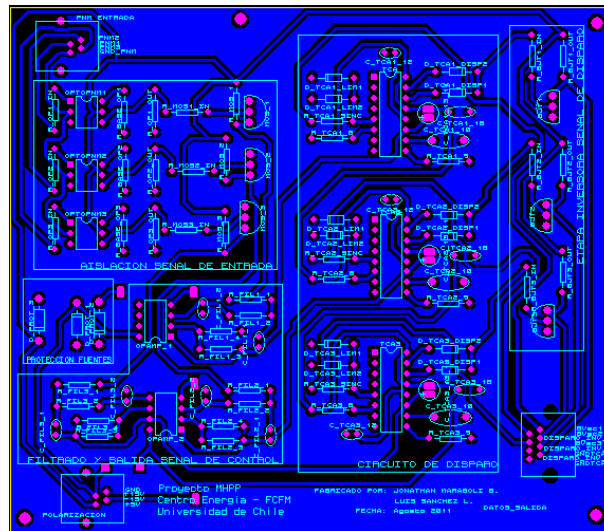


Figura B.17: Diseño de placa de disparo de resistencias de desahogo

En la práctica fue necesario modificar el diseño. Cambiando bloques de transistores que invertían la señal por resistencias y cambiando los tipos de transistores utilizados, lo cual se puede apreciar en la siguiente imagen.

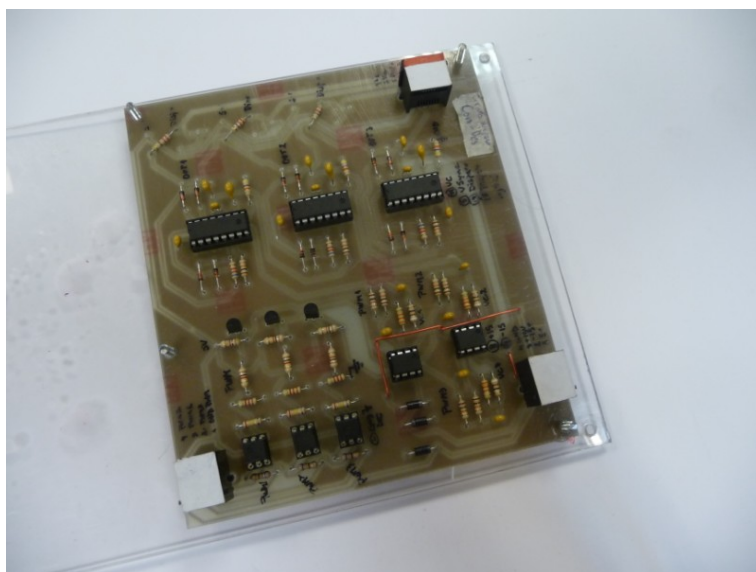


Figura B.18: Placa de disparo de resistencias de desahogo

B.1.6. Fin de carrera de válvula y estado de contactores

La válvula mariposa y los contactores del tablero tienen salidas digitales que se van a utilizar para indicar al DSC los estados en que se encuentran, pero es necesario adaptar las señales al nivel de voltaje aceptable de los GPIOs (3,3V). Entonces se diseña un circuito de adaptación de tensiones en base a un regulador de tensión.

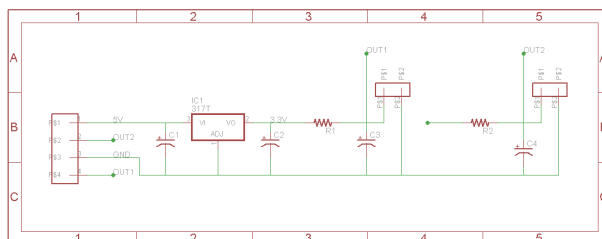


Figura B.19: Esquemático de placa de fin de carrera y estado de contactores

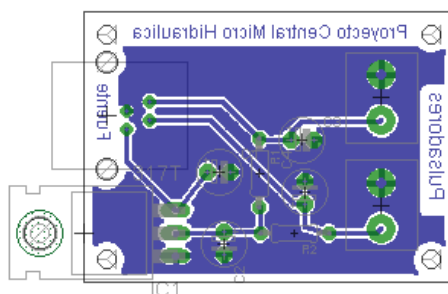


Figura B.20: Diseño de placa de fin de carrera y estado de contactores

Se utiliza el mismo diseño para la válvula y los contactores.

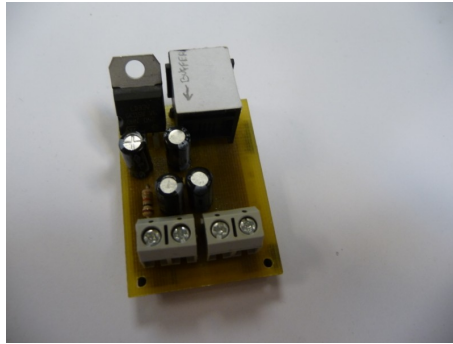


Figura B.21: Placa de fin de carrera

Luego es necesario hacer una modificación a la placa de los contactores, separando la alimentación de las salidas digitales para poder alimentarla desde la placa de suministro sin necesidad de pasar por la placa de buffers como es el caso de el final de carrera para la válvula.

Adicionalmente es necesario modificar la adaptación pues el diseño se pensó para señales digitales de 5 [V], siendo que en la realidad son señales de 220[V].

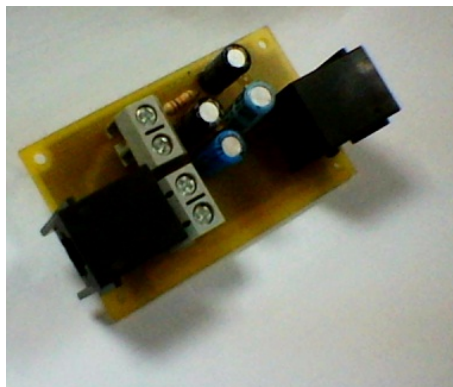


Figura B.22: Placa de estado de contactores

B.2. Electrónica del tablero

B.2.1. Sensores corriente y tensión

Se diseña un circuito para medición de variables de tensión y corriente de la red y el generador. Se utilizan LEMs y transformadores de bajada.

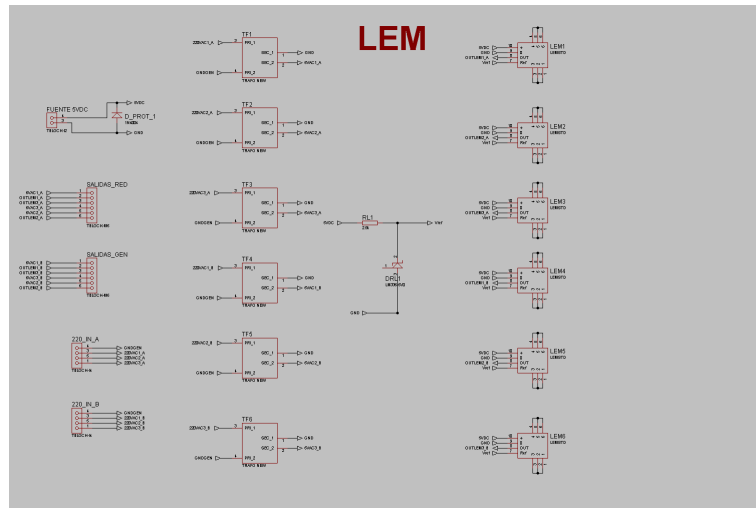


Figura B.23: Esquemático de placa de sensores

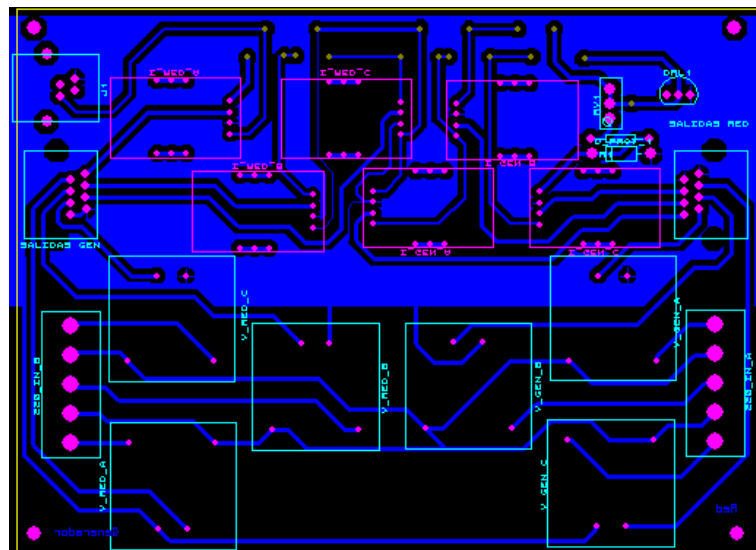


Figura B.24: Diseño de placa de sensores

En la siguiente imagen se puede apreciar como deben ir las conexiones de esta placa.

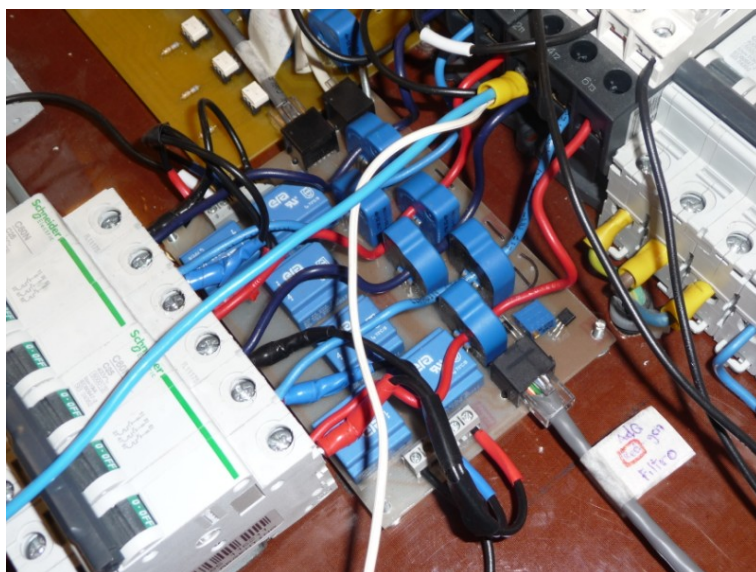


Figura B.25: Placa de sensores

Es necesario ajustar la configuración de la referencia de tensión de los LEMs para la medición de corriente.

B.2.2. Relés

Se utilizan relés para el accionamiento de la válvula y para los contactores de la red y el generador. Se debe optoacoplar la señal que opera los relés para no dañar al DSC. Se realiza el diseño y la construcción como muestran las siguientes imágenes.

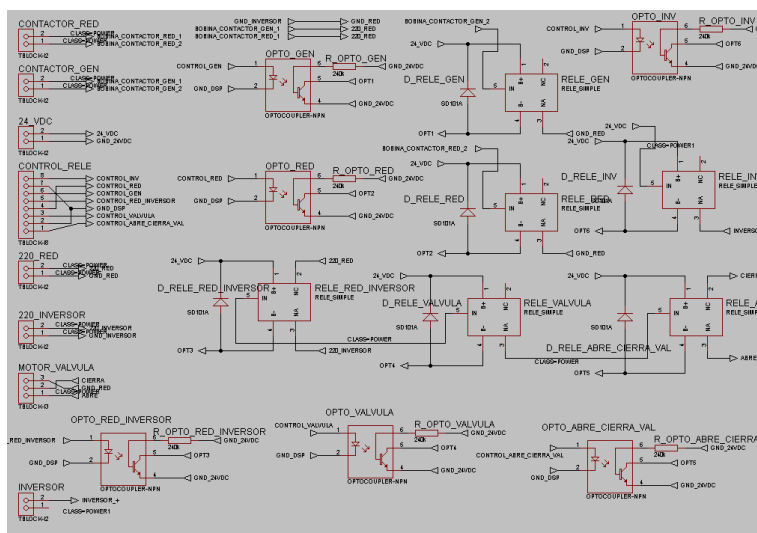


Figura B.26: Esquemático de placa de relés

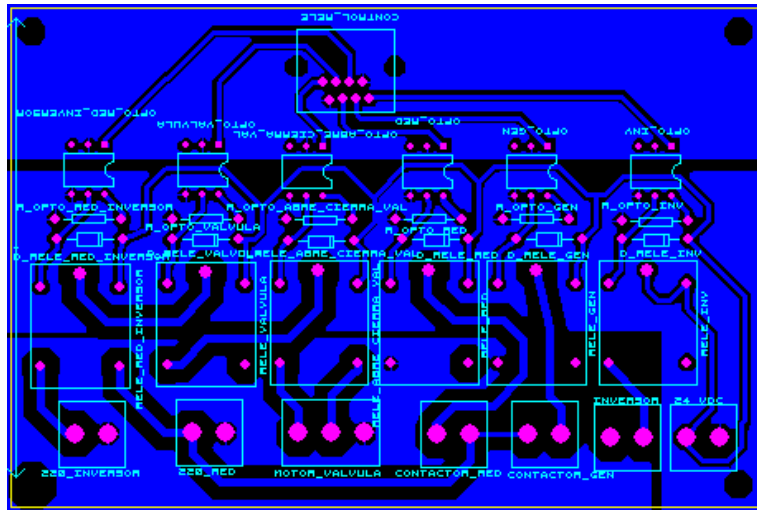


Figura B.27: Diseño de placa de relés

En la siguiente figura se puede apreciar la placa de relés con los conectores indicando a que corresponde cada uno.



Figura B.28: Placa de relés

B.2.3. Disparo resistencias de desahogo (circuito 2)

En la cúpula se posiciona el primer bloque necesario para el funcionamiento de las resistencias de desahogo y en el tablero el segundo, que considera un optoacoplamiento para proteger la electrónica de la placa en la cúpula, contiene los transformadores de bajada para la sincronización de las señales y los LEM para medición de corriente en las resistencias de desahogo.

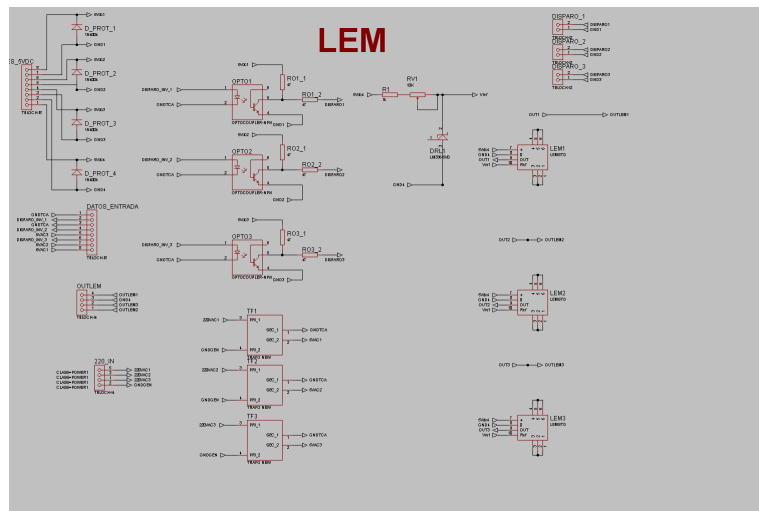


Figura B.29: Esquemático de placa de disparo de resistencias de desahogo

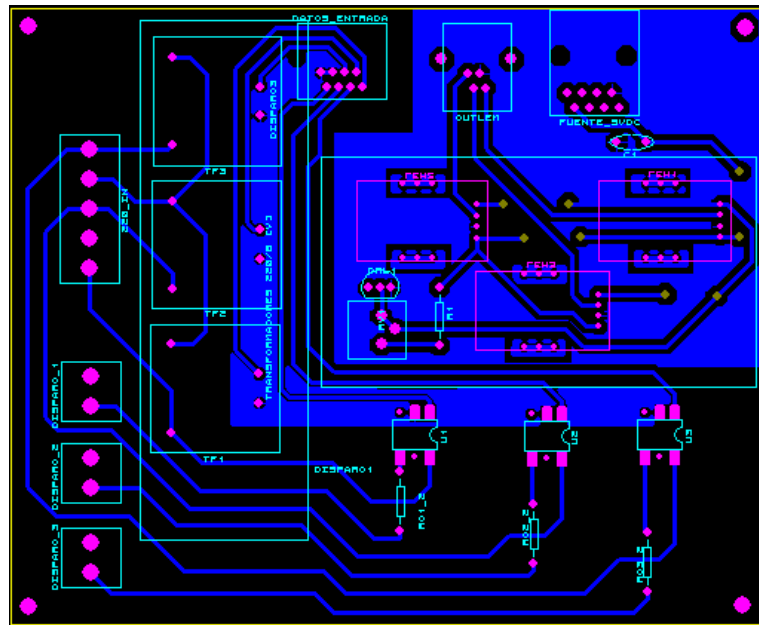


Figura B.30: Diseño de placa de disparo de resistencias de desahogo

En primera instancia como muestra el diseño anterior se pensó utilizar optoacopladores en base a transistores, pero ello requiere alimentación independiente para cada integrado, por ello se modifica la placa para utilizar optotriacs permitiendo liberar fuentes de la placa de suministro que pueden ser utilizadas para alimentar los circuitos de los contactores y buffers.



Figura B.31: Placa de disparo de resistencias de desahogo

Se utiliza el triac BTA25-700B por su encapsulado y la característica de que es aislado por lo cual se puede utilizar un mismo disipador para los tres TRIACS de las resistencias de desahogo. Se debe tener cuidado de conectar A2 del triac al generador y A1 a la resistencia.

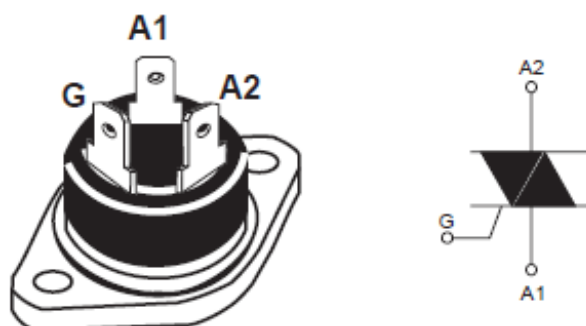


Figura B.32: Pines de TRIAC BTA25-700B para conexión

B.2.4. Chopper

Para el control de la corriente de campo se diseña un chopper, que considera un conector para la entrada del PWM y la alimentación y para la salida de la medición de la corriente de excitación para el generador. Se utiliza un optoacoplador para proteger la salida digital del PWM.

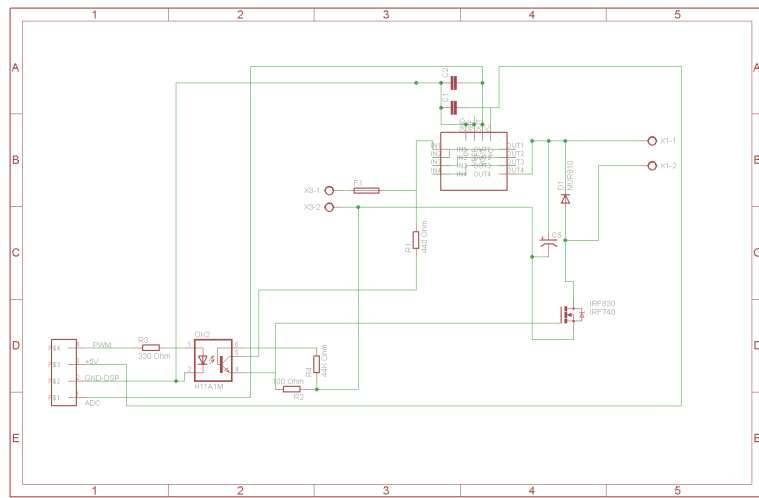


Figura B.33: Esquemático de placa de chopper

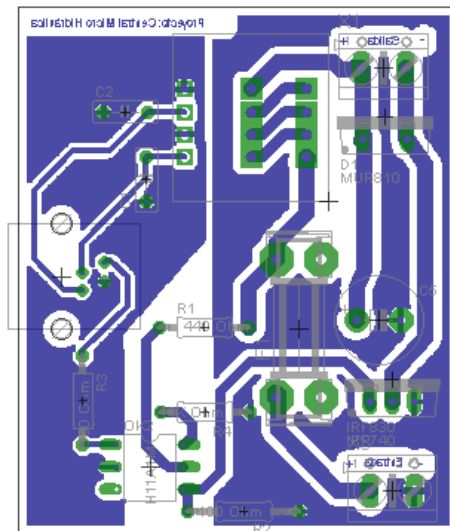


Figura B.34: Diseño de placa de chopper

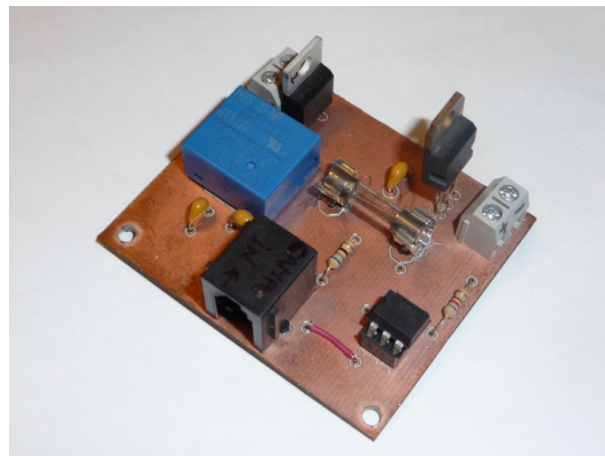


Figura B.35: Placa de chopper

B.2.5. Disparo puente rectificador

Para la carga del banco de baterías se considera un puente rectificador semicontrolado, el cual es necesario controlar. Debido a que todos los periferales de análogo a digital se encuentran en uso, se decide diseñar un circuito que controle de forma independiente el puente, mediante un PIC, y que se pueda comunicarse con el DSC mediante I2C para informar el estado de carga de las baterías.

Se realiza el diseño y la implementación con alimentación desde la placa suministro y se utilizan optotiristores para el acomplamiento de la señal de disparo de los tiristores del puente.

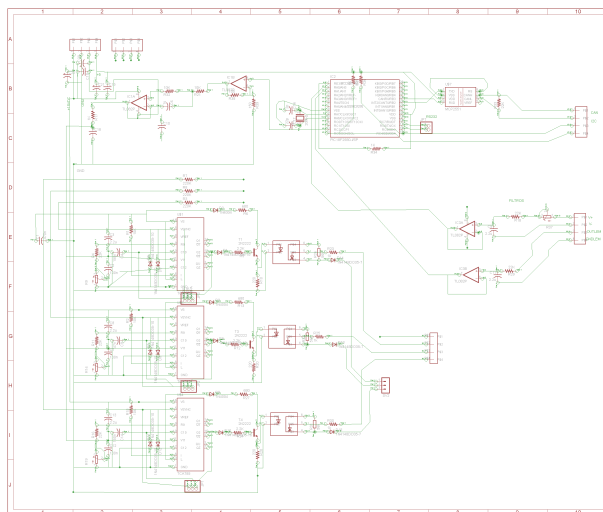


Figura B.36: Esquemático de placa de disparo de puente rectificador

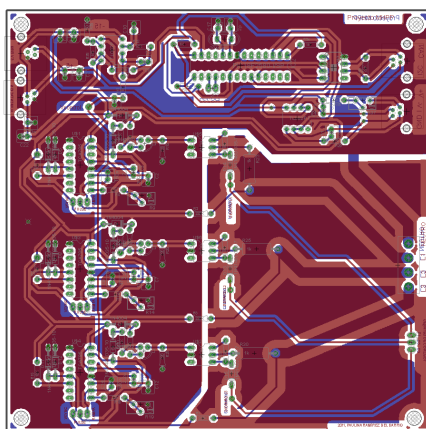


Figura B.37: Diseño de placa de disparo de puente rectificador

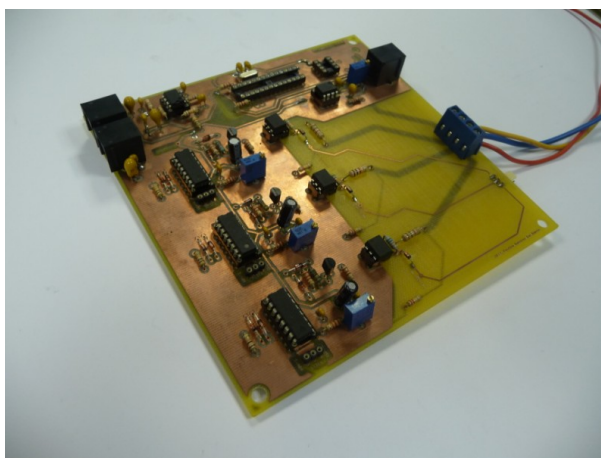


Figura B.38: Placa de disparo de puente rectificador

Se pone a prueba este bloque de la carga de baterías, pero no en conjunto con el puente en sí.

B.2.6. Puente rectificador

Se diseña el puente y consideran un LEM de continua para medición de corriente y un divisor de tensión para medición de voltaje, se coloca un condensador sobre dimensionado para eliminación del rizado.

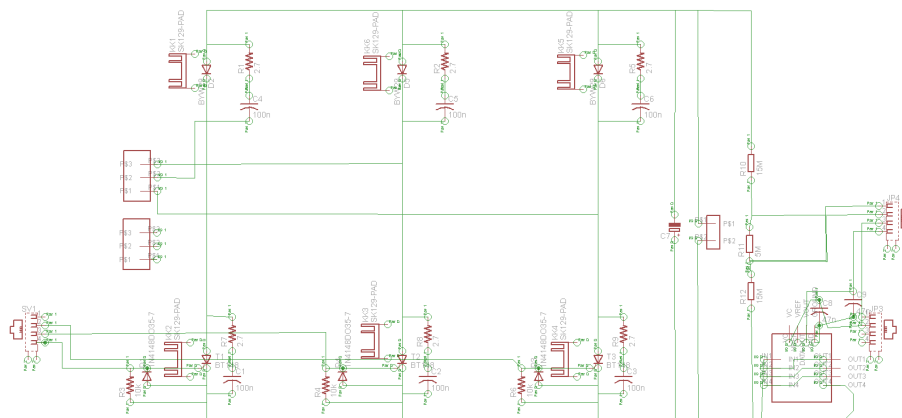


Figura B.39: Esquemático de placa de puente rectificador

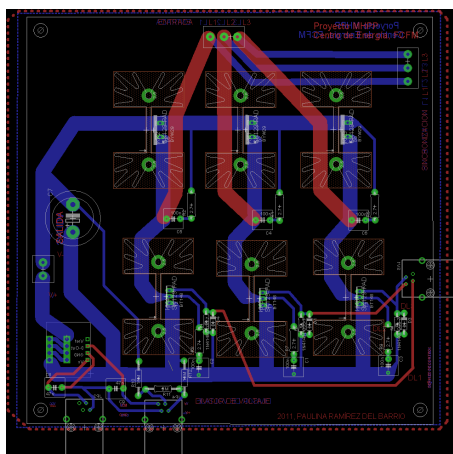


Figura B.40: Diseño de placa de puente rectificador

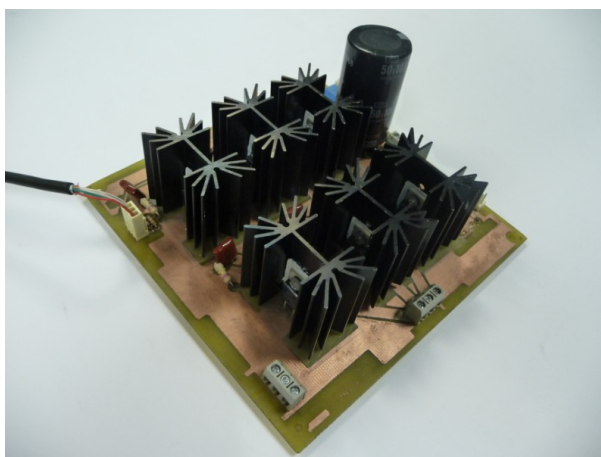


Figura B.41: Placa de puente rectificador

Es necesario volver a dimensionar el condensador, pues no se logran obtener buenos resultados con esta placa, ya que demora bastante en cargar y descargar el capacitor.

Tanto el puente como la placa de disparo requieren la utilización de tres transformadores, por lo que si se consideran como un sólo bloque, éste utiliza un espacio considerable. Por esto se recomienda para la MCH o bien rediseñar los circuitos o adquirir un cargador de baterías disponible en el mercado.