



UNIVERSIDAD AUTÓNOMA DEL ESTADO DE MÉXICO
CENTRO UNIVERSITARIO UAEM ATLACOMULCO



MAESTRÍA EN CIENCIAS DE LA COMPUTACIÓN

“DISEÑO E IMPLEMENTACIÓN DE UN CONTROLADOR PID
AUTOAJUSTABLE BASADO EN UN ALGORITMO HÍBRIDO”

T E S I S

QUE PARA OBTENER EL TÍTULO DE:

MAESTRO EN CIENCIAS DE LA COMPUTACIÓN

P R E S E N T A

JUAN CARLOS AMBRIZ POLO

ASESOR: DR. ALLAN ANTONIO FLORES FUENTES

Atacomulco México, Noviembre 2015.

ÍNDICE

CAPÍTULO 1

INTRODUCCIÓN

1.1	Antecedentes	6
1.2	Planteamiento del problema.....	10
1.3	Hipótesis.....	11
1.4	Pregunta de investigación.....	12
1.5	Estado del Arte.....	12
1.6	Objetivo general.....	14
1.7	Objetivos específicos.....	15
1.8	Justificación.....	15

CAPÍTULO 2

MARCO TEÓRICO

2.1	Algoritmos computacionales; fundamentos.....	17
2.2	Algoritmos Heurísticos.....	18
2.3	Ejemplos de algoritmos heurísticos.....	19
2.3.1	Algoritmo evolutivo.....	19
2.3.2	Algoritmo Genético.....	20
2.3.3	Colonia de Hormigas.....	22
2.3.4	Optimización por Temple simulado.....	23
2.3.5	Optimización por Enjambres de Partículas.....	24
2.4	Optimización.....	25
2.5	Teoría de Lógica Difusa.....	26
2.6	Teoría del control moderna.....	28
2.6.1	Sistema de control.....	29
2.6.2	Fundamentos teóricos de los sistemas de control.....	29
2.6.3	Función de Transferencia.....	30
2.6.4	Diagramas de Bloques.....	30
2.6.5	Sistemas de control en lazo abierto.....	31
2.6.6	Sistemas de control en lazo cerrado.....	31
2.6.7	Teoría de Control Moderna.....	32

2.6.8	Respuesta en estado estable y transitorio de un sistema de control.....	33
2.6.9	Tipos de controladores.....	35
2.6.10	Controlador tipo Proporcional-Integral-Derivativo (PID).....	35
2.6.11	Métodos de sintonización para controladores PID.....	36
2.7	Aplicación del control automático en Electrónica de Potencia.	38
2.8	Convertidores estáticos de energía.	38
2.8.1	Topología del Convertidor Multicelular.	40
2.8.2	Equilibrio de voltaje en los capacitores flotados.	41
2.8.3	Estrategias de Conmutación para convertidores de energía.	42

Capítulo 3

Diseño y Desarrollo del sistema

3.1	Propuesta y diseño del inversor multicelular de tres células.....	45
3.1.1	Construcción del inversor de tres células.....	46
3.1.2	Construcción de la etapa de alimentación CD del inversor.....	47
3.1.3	Generación de señales de conmutación mediante FPGA.....	48
3.1.4	Generación de señales complementarias (S1', S2', S3') y tiempos de seguridad (Dead-Time).	54
3.1.5	Acoplamiento de señales de conmutación (S1, S1', S2, S2', S3, S3').	56
3.2	Operación del convertidor de tres células en lazo abierto.....	58
3.3	Operación del convertidor en lazo cerrado.....	60
3.3.1	Diseño y codificación del Algoritmo Híbrido.....	61
3.3.2	Etapa para detección de las variables de voltaje.	64
3.3.3	Controlador PID digital.....	65

CAPÍTULO 4

Resultados y Conclusiones

4.1	Ejecución del AHHL.	70
4.2	Control del ciclo de trabajo de las señales de conmutación.....	72
4.3	Resultados del control PID implementado al inversor de tres células con capacitores flotados.	74
4.4	Conclusiones.	77
4.5	Trabajo a futuro.	79

Referencias

Anexo A.....	85
--------------	----

Anexo B	86
Anexo C	87

CAPÍTULO 1

INTRODUCCIÓN

En esta sección se describen los puntos con base en la metodología de la investigación, seguidos para el desarrollo de este trabajo tales como; la delimitación, descripción del problema, estado del arte, objetivos planteados, hipótesis y justificación.

1.1 Antecedentes

Dentro de la rama de ingeniería existen técnicas que ayudan a ordenar y simplificar la solución de problemas, específicamente en el área de computación son nombrados algoritmos. Éstos corresponden a la descripción de un patrón de comportamiento, expresado en términos de un conjunto finito de pasos, que tiene como objetivo establecer una solución a un problema específico [1]. En el área de la algoritmia existen diversas clasificaciones, tales como; algoritmos determinísticos, de búsqueda, clasificación, seguimiento, ordenamiento, heurísticos entre otras clasificaciones no tan implementadas. En el presente trabajo se aborda el estudio y aplicación de los clasificados como algoritmos heurísticos, los cuales son considerados técnicas que pueden dar una solución óptima de problema a un costo computacional razonable, pero no garantizan lograr la solución en todos los casos, y son aplicados a problemas complejos donde los espacios de búsqueda de soluciones son indefinidos, y donde algoritmos como los determinísticos no logran encontrar solución, además de requerir un tiempo exponencial de ejecución. Entre los algoritmos que destacan bajo la clasificación de heurísticos se encuentran los algoritmos bioinspirados, tales como; algoritmos evolutivos, genéticos, enjambres de partículas, colonia de hormigas, estructuras de vecindad, algoritmos de aleatoriedad, algoritmos inmunológicos artificiales, redes neuronales, estrategias evolutivas, algoritmos de expansión clonal, algoritmos de redes, así como múltiples adaptaciones e hibridaciones entre ellos [1-3, 5].

Posterior a los algoritmos heurísticos, surgen los nombrados algoritmos metaheurísticos en donde el prefijo “meta” describe la acción de ir más allá o buscar un nivel superior. Esto es atribuido a que este tipo de algoritmos generalmente tienen un mejor desempeño que los heurísticos [2-5]. Dentro de las características que comparten los algoritmos heurísticos y metaheurísticos se encuentra la utilización de procesos estocásticos, así como un conocimiento previo de posibles soluciones para los problemas que se desea resolver. Recientemente existe la tendencia de establecer como metaheurísticos a todos los algoritmos estocásticos que incluyen un grado de aleatoriedad y búsqueda local, este enfoque hace resaltar que para los algoritmos heurísticos, una búsqueda local, así como una rápida convergencia (tendencia a una misma solución), representa una desventaja. En contra parte para las metaheurísticas, las características antes mencionadas son tomadas como ventajas que permiten disminuir el costo computacional de estas técnicas, al encontrar soluciones rápidas y óptimas para los problemas [2, 3].

Dentro del mismo contexto de la Inteligencia Artificial (IA), surge un rama de estudio denominada lógica difusa o borrosa, la cual propone analizar información en una escala entre lo falso y lo verdadero, permitiendo que los sistemas manejen conceptos no precisos del lenguaje humano (o también nombrados subjetivos) tales como: “es muy bajo”, “es muy alto”, que comúnmente al traducirse al contexto de la lógica clásica pierden parte de su significado, cuestión que podría ser importante para problemas en los que los algoritmos determinísticos no pueden dar una solución y que el sentido común podría dar debido a que hace referencia a cantidades y acciones que involucran incertidumbre. La introducción del termino de lógica difusa dentro del área de control en ingeniería, ha permitido que múltiples algoritmos enfocados en el control de sistemas, mejoren su desempeño al generar restricciones que son útiles para la evaluación de soluciones, debido a que no sólo se basa en la lógica bi-valuada (cierto y falso) si no a una lógica multi-valuada (más de dos valores) donde se encuentran puntos intermedios, permitiendo que las restricciones sean cada vez más parecidas al razonamiento del ser humano y así lograr una mayor eficiencia en la evaluación de las soluciones que generen los algoritmo [4, 5].

Las metaheurísticas proporcionan un marco general para crear nuevos algoritmos híbridos, combinando diferentes conceptos derivados de la inteligencia artificial, la evolución biológica, procedimientos estadísticos y lógica difusa [3-5]; si bien es comprobable la eficiencia de algoritmos metaheurísticos por si solos como técnicas de búsqueda [6, 8, 9], apostar por la combinación de varias de estas técnicas así como la implementación de restricciones, permite superar las expectativas que se tiene de los algoritmos metaheurísticos simples, conjugando las características que cada técnica aporta.

Es preciso resaltar que los algoritmos heurísticos y metaheurísticos, así como los híbridos son principalmente aplicados a problemas de búsqueda y optimización. Una de las áreas de ingeniería donde han resultado ser eficientes, es en el área de ingeniería de control automático, la cual se encarga del estudio de subsistemas denominados controladores, que permiten la manipulación manual o automática de una sistema, nombrado técnicamente como planta, de tipo; mecánico, eléctrico, electrónico, neumático, entre otros. Específicamente un controlador compara la salida de una planta con la entrada (el valor deseado), determina la desviación (error) y produce una señal que reducirá la desviación a un valor mínimo aproximado a "0" o "cero" (señal de control). La manera en la que el controlador produce la señal de control se denomina acción de control. El controlador más implementado en el área de control automático es nombrado Proporcional-Integral-Derivativo o PID, por sus siglas en inglés. Actualmente debido a su eficiencia y versatilidad, así como su fácil implementación no sólo de manera analógica sino también digital, es ampliamente utilizado para efectuar acciones de control en diversas áreas de aplicación [7, 10].

A principios de la década de los 90's, los algoritmos heurísticos se implementaron por primera vez como un medio alternativo de ajuste de controladores, empleándolos para determinar estimaciones iniciales de los parámetros que constituyen los controladores tipo PID. El control proporcional, integral y derivativo consiste en la suma de tres acciones de control en uno solo: a) Control Proporcional, b) Control Integral y c) Control Derivativo. En el Control Proporcional la salida del controlador utiliza una "proporción" del error del

sistema para su acción. Su desventaja radica en introducir un error de desviación en el sistema debido al tiempo que tarda en calcular y aplicar la señal de control. Con respecto al Control Integral la salida del controlador es proporcional a la cantidad de tiempo que existe en el error presente en el sistema. La acción integral elimina el desfase introducido por el control proporcional, pero introduce un retardo de fase en el sistema debido a tiempo que tarda en establecerse el sistema. Mediante el Control Derivativo la salida del controlador es proporcional a la velocidad de cambio del error (d_{error}/dt), en otras palabras la derivada se utiliza para eliminar el desfase introducido por la acción integral (se compensan uno con respecto al otro) así; la suma de las tres acciones de control sumadas en uno solo conforman un controlador robusto de tal manera que es eficiente para conseguir el efecto deseado de un sistema dicho de otra forma logra reducir e incluso eliminar el error [7,10]. Es importante definir el error mediante la ec. (1) donde Val_{Ref} es el valor de referencia o entrada y Val_{Sal} es el valor de salida del sistema [7].

$$error = Val_{Ref} - Val_{Sal} \quad (1)$$

Una de las áreas donde es preponderante la implementación de sistemas de control automático, es la electrónica de potencia (dada la complejidad de este tipo de sistemas) debido a que proporciona control en parámetros eléctricos de voltaje, corriente o potencia, y en ocasiones de frecuencia de operación. Lo anterior tiene como consecuencia que técnicas de control PID se utilicen para lograr un mejor desempeño en los parámetros eléctricos de entrada/salida, en convertidores de energía estáticos. Por esta razón, mediante la introducción del concepto de Soft-Computing a el área de control y de forma particular en la sintonización de parámetros del controlador PID surge la necesidad de implementar algoritmos específicos que sean capaces no sólo de sintonizar sino también de optimizar los parámetros de este tipo de controladores. Este concepto ha sido estudiado en el área de electrónica de potencia en los siguientes trabajos [11-15], en los cuales se hace referencia de los controladores PID, como una de las mejores opciones de control que conllevan al máximo rendimiento convertidores de energía.

Dentro del área de electrónica de potencia, un convertidor de energía estático, es un sistema electrónico conformado por un conjunto de elementos estáticos (semiconductores) que tiene como propósito la transferencia de energía eléctrica de un generador a un receptor; para entender el funcionamiento de los inversores multinivel de corriente directa (CD) a corriente alterna (CA). Es indispensable considerarlos como seleccionadores de voltaje; donde la salida del inversor es un voltaje alterno que se obtiene a partir de los diferentes niveles de voltaje de CD de entrada. Éstos a su vez están determinados por la diferencia de potencial de los capacitores flotantes o flotados que integra el circuito del convertidor de energía estático [14], en los cuales es esencialmente importante mantener un equilibrio del voltaje (valor de referencia específico); siendo necesario implementar en muchas ocasiones un control automático (típicamente PID) a los capacitores flotados, para su correcto funcionamiento como fuente de voltaje ideal en el convertidor de energía estático.

1.2 Planteamiento del problema.

Un inversor de energía estático, convierte energía de tipo C.D. a C.A, puede ser una magnitud de voltaje (Inversor configurado en Fuente de Voltaje o VSI, por sus siglas en inglés) o bien de corriente (Inversor configurado en Fuente de Corriente o CSI, por sus siglas en inglés). El control automático de alguno de los parámetros antes mencionados es parte fundamental de un convertidor de energía, debido a que en la medida que esto se logra se tienen dos características importantes: a) Control en la calidad de energía; es decir reducir la Distorsión Armónica Total (THD por sus siglas en inglés), y b) El control de la eficiencia del sistema de potencia (η), o bien la relación entre la potencia de salida con respecto a la de entrada dada en porcentaje $\eta = P_o/P_i$.

En el caso particular de un inversor de tipo multicelular, la acción de control para manipular estos parámetros de voltaje y/o corriente se puede logra de manera sencilla y eficaz mediante un control en lazo cerrado tipo *feedback* (retroalimentación negativa) en donde se logra medir los parámetros a la salida de inversor (carga), y además en los capacitores flotados que son parte fundamental para el correcto funcionamiento del inversor. Así, en la medida de que se conozcan los valores deseados se establece un valor

de referencia de voltaje o corriente, el cual al ser comparado con el valor de salida del inversor estático, generándose un error. Éste parámetro es al que se aplica la acción de control PID, para reducirlo al mínimo y teóricamente que se vuelva cero en el estado estacionario del sistema.

Con base en este argumento se requiere diseñar e implementar un Algoritmo Híbrido que combine y complete los beneficios de técnicas heurísticas y lógica difusa para sintonizar y optimizar los parámetros de ajuste de un controlador automático tipo Proporcional-Integral-Derivativo (PID), implementado al sistema de control en lazo cerrado de un inversor de energía estático de tres células en configuración medio puente.

El uso de tecnología basada en inversores de tipo multicelular y multinivel, ha permitido que en la actualidad se maneje cada vez mayor cantidad de potencia en los sistemas industriales, tales como; conversión de energía fotovoltaica, eólica, térmica entre las más típicas. Por esta razón, y con la expectativa para el desarrollo de nuevas tecnologías, no sólo en electrónica de potencia sino principalmente en el área de control automático asistido por computación evolutiva, es un área emergente en la toma de las decisiones de manera automática para sistemas más confiables y eficientes.

1.3 Hipótesis.

Mediante el diseño e implementación de un controlador basado en un Algoritmo Híbrido, desarrollado con heurísticas y restricciones apoyadas en teoría difusa se logrará sintonizar y optimizar los parámetros de un controlador PID (K_p, K_i, K_d), implementado a los capacitores flotados que conforman el inversor de energía estático de topología multinivel. El control implementado, minimizará el error total de las respuestas de carga de voltaje de los capacitores flotados. Para minimizar el error se consideran las características de redondeo del sistema, las cuales son: a) Máximo sobre impulso (M_p), Tiempo de establecimiento (s_r) y Tiempo de subida o de levantamiento (r_r). Es importante destacar que dentro de la literatura, la implementación de sistemas de control PID sintonizados mediante técnicas heurísticas, aplicados a este tipo de convertidores sólo lleva a cabo

mediante simulaciones, mientras que este trabajo de tesis se considera realizar la implementación real del sistema para evaluar si un control tipo PID es una estrategia de control apropiada para este tipo de sistemas.

1.4 Pregunta de investigación.

¿Qué resultados se obtienen al implementar un controlador PID sintonizado mediante un Algoritmo Híbrido (heurísticas-teoría difusa) inversor de energía estático? y ¿si se sintonizan y optimizan las constantes del controlador K_p , K_i y K_d , que efecto se logra en el desempeño del inversor en lazo cerrado con respecto al control en lazo abierto?

1.5 Estado del Arte.

En el área de Control Automático aplicado a la Electrónica de Potencia, las heurísticas son una herramienta aplicada para la sintonización de parámetros de controladores de tipo PID, dentro de estas técnicas se encuentran, los algoritmos genéticos, algoritmo de colonias de hormigas, enjambres de partículas, redes neuronales, aplicación de algoritmos difusos, así como la combinación de éstos con otras técnicas de búsqueda y optimización (algoritmos híbridos), las cuales han demostrado, mediante simulaciones su eficacia al ser comparadas con técnicas tradicionales de sintonización de parámetros PID. Algunos de los trabajos encontrados se describen a continuación. En el trabajo de Aytakin Bagis *et al.* [16], se menciona el rendimiento mejorado de una algoritmo híbrido aplicando la combinación de un Algoritmo Genético (GA, por sus siglas en inglés, Genetic Algorithm) con una estructura de vecindad utilizado para la sintonización de los parámetros de un PID. Además Xiangzhong Meng *et al.* [17], proponen una modificación a los GAs nombrándolo *Fast Genetic Algorithm* (FGA, por sus siglas en inglés) en el cual se mejoran los argumentos de la generación de población, la mutación y cruce de los individuos para lograr una rápida convergencia del algoritmo. Otros autores como Liqing Xiao *et al.* [18] establecen en su trabajo; que un GA es obsoleto en cuanto a la rapidez de convergencia y desarrollan un *Hybrid Genetic Algorithm*, aplicado a un controlador PID, el cual optimiza el proceso de mutación del Algoritmo Genético mediante la implantación de la técnica de Enjambre de Partículas (PSO, por sus siglas en inglés, Particle Swarm Optimization);

técnica de tipo heurística estudiada dentro de la misma área del Soft-Computing. Continuando con las aplicaciones de los Algoritmos Híbridos implementados en conjunto con otras técnicas para mejorar sus desempeño, en el trabajo de Shao Xiao-Gen *et al.* [9], se combina un GA típico y un algoritmo de intervalo, para la optimización de un controlador PID limitando la generación aleatoria de la población en un rango de inicialización con el fin de mejorar la velocidad de convergencia del GA [8].

Por otra parte, la teoría de lógica difusa también ha tenido importantes avances en cuanto a aplicaciones de auto sintonización de controladores PID, uno de los ejemplos de la aplicación de la lógica difusa se describe en el trabajo de Hyung-Soo Hwang *et al.* [19] el cual describe el diseño de un algoritmo de auto ajuste de los parámetros del PID partiendo de la sintonización de la ganancia proporcional (Kp) obtenida mediante el método típico gráfico Ziegler- Nichols y la implementación de una función de pertenencia basada en el error. George K. I. Mann *et al.* [20] en su trabajo “*Two-Level Tuning of Fuzzy PID Controllers*” menciona que un algoritmo difuso de sintonización para controladores PID requiere de dos niveles de ajuste un nivel bajo donde se determinen las características lineales de la salida de sistema y un nivel alto donde a través de heurísticas se determinen las características no lineales del sistema. Jian-Xin Xu *et al.* [21] realiza un algoritmo de aprendizaje iterativo basado en la observación de las características de rendimiento del sistema como tiempo de establecimiento, tiempo de subida, sobre impulso y va proponiendo valores aleatorios para los parámetros del PID los cuales se modifican en cada iteración. Jiangming Kan *et al.* [22] en su trabajo “*Fuzzy Immune Self-Tuning PID Controller and its Simulation*” señala combinación de un algoritmo inmune y un algoritmo difuso para el auto ajuste de los parámetros del PID proponiendo en su algoritmo tomar las ventajas de las dos técnicas anteriores para sintonización del parámetro Kp y la sintonización de Ki y Kd se ajustan en línea de acuerdo con razonamiento difuso.

Por su parte Lv Wenge *et al.* [23] resalta que no se puede estandarizar un algoritmo para el autoajuste óptimo de un PID debido a que existen diversos factores que hacen variar el comportamiento de los sistemas y que cada controlador PID se debe autoajustar

dependiendo del sistema en el que se implementara y propone en sus trabajo “*Tuning Digital PID Controllers for Discrete-time System by Election Campaign Optimization Algorithm*” un algoritmo de campaña donde se generan candidatos (soluciones) posibles aleatoriamente y con base a una función objetivo que evalúa rendimiento a través de la Integral del Error Absoluto (IAE, por sus siglas inglés) se genera un índice de empatía, se realizan evaluaciones a todos los candidatos y se comparan sus índices de empatía siendo el ganador de la campaña el que mejor empatía adquiriera. Qi-yao Yang *et al.* [24] diseña un controlador PID difuso adaptativo en el que las variables de control del PID se evalúan por el 70% RMS del error y los valor iniciales de los parámetros PID se ajustan manualmente, posteriormente los parámetros se corrigen mediante razonamiento difuso para encontrar un óptimo ajuste del PID. Sinthipsomboon, K. *et al.* [25] en su trabajo “*A Hybrid of Fuzzy and Fuzzy self-tuning PID Controller for Servo Electro-hydraulic System*” realiza una combinación de dos controladores, por una parte un control difuso para controlar el sistema cuando el valor de salida del sistema está alejado del valor deseado y en segundo lugar un control PID auto sintonizado mediante teoría difusa, aplicado cuando el valor de salida del sistema se encuentra cercano al valor deseado.

Por lo tanto, con base en lo anterior se puede concluir que la implementación de heurísticas en específico a través Algoritmos Híbridos son utilizados para la búsqueda y solución de problemas de optimización entre los que destacan la sintonización de parámetros de controladores de tipo PID, en diferentes áreas, con el objetivo lograr un creciente desempeño de los sistemas de control automático.

1.6 Objetivo general.

Diseñar e implementar un Algoritmo Híbrido Heurística - Lógica Difusa (AHHLD, por sus siglas en español) para optimizar y sintonizar los parámetros de un controlador PID, implementado para controlar la respuesta del voltaje en los capacitores flotados de un convertidor de energía estático de topología multicelular, que será construido en el presente trabajo.

1.7 Objetivos específicos.

1. Construir un convertidor de energía estático de topología multicelular de tres celdas.
2. Diseñar e implementar un sistema digital de mando para las señales de conmutación del convertidor
3. Diseñar e implementar un sistema de medición de voltaje de los capacitores flotados del convertidor.
4. Generar la función de optimización, para reducir los parámetros de error en estado estable en base a los criterios de rendimiento (M_p, s_t, r_t) de las respuestas de los capacitores flotados del convertidor de energía estático.
5. Diseñar y codificar Algoritmo Híbrido Heurística - Lógica Difusa para optimizar y sintonizar los parámetros del controlador PID.
6. Implementar el controlador PID al convertidor de energía estático de topología multicelular de tres celdas.
7. Analizar los resultados y establecer conclusiones.

1.8 Justificación.

Debido a que en el área de control más de la mitad de los controladores industriales que se usan hoy en día utilizan esquemas de control PID o PID modificado (control PI-D y el control I-P-D), es importante buscar y aplicar nuevas técnicas que permitan la sintonización y optimización de los parámetros del controlador para mejorar el rendimiento de sistemas, por mencionar algunos; hidráulicos, neumáticos, electrónico, eléctrico o una combinación entre los antes mencionados [7].

Por lo tanto las herramientas tecnológicas, en cuanto a algoritmos computacionales se refieren, y con base a la literatura los Algoritmos Híbridos heurísticos en ingeniería de control tienen como principal aporte optimizar la sintonización de parámetros PID con mayor eficiencia en comparación con técnicas tradicionales de sintonización y algoritmos heurísticos simples, Por estas razones y para conocer la efectividad y la aplicabilidad de los Algoritmo Híbrido en sistemas de control se desarrolla e implementa un Algoritmo Híbrido

Heurística - Lógica Difusa, aplicado a los capacitores flotados que conforman un inversor de energía estático de tres células en configuración de medio puente.

CAPÍTULO 2

MARCO TEÓRICO

En esta sección se documenta la información que brinda sustento teórico a la propuesta descrita en este trabajo de investigación. Se describen los conceptos, características, procedimientos, teorías y ejemplos para el desarrollo de técnicas heurísticas, así como lo referente a teoría de control y el convertidor de energía estático.

2.1 Algoritmos computacionales; fundamentos.

Un algoritmo se define como un conjunto de instrucciones bien definidas, ordenadas y finitas, que permiten realizar una actividad mediante pasos sucesivos. La solución de problemas algorítmicamente, implica el hecho de poder realizar un programa de computadora que produzca una respuesta correcta para el problema, mediante una entrada y un tiempo de ejecución [26]. Existen diversas clasificaciones para los algoritmos, pueden clasificarse con base al proceso que ejecutan como algoritmos determinísticos los cuales siguen un riguroso proceso respecto a su trayectoria y variables, un ejemplo de algoritmo determinístico es un algoritmo de burbuja para ordenamiento, que dado un conjunto de valores los ordenará de manera creciente o decrecientemente, independientemente de cuándo y cuántas veces se ejecute el algoritmo el resultado será el mismo. Otra clasificación es el caso de los algoritmos estocásticos los cuales se fundamentan bajo el concepto de lo aleatorio; un ejemplo de estos son los Algoritmos Genéticos los cuales generan diferentes soluciones cada vez que son ejecutado, y aun cuando el resultado puede ser el mismo, la trayectoria que este siguió para obtenerlo es difícil que se repita. De la misma forma existen algoritmos híbridos entre lo determinístico, lo estocástico y algunas otras teorías del área de Inteligencia Artificial (IA), que buscan como objetivo la optimización de su funcionamiento [3]. En el presente trabajo abordaremos la clasificación

de algoritmos heurísticos los cuales son una rama de los algoritmos estocásticos, como se muestra en la figura 2.1.

2.2 Algoritmos Heurísticos.

Cuando se localizan problemas que son difíciles de resolver o bien para los cuales existe un espacio de búsqueda de soluciones muy grande y que además los algoritmos o métodos más eficientes que existen para resolver el problema requieren de tiempo exponencial para encontrar solución, es claro observar que las técnicas clásicas de búsqueda y optimización son insuficientes; es ahí cuando se hace uso de las Heurísticas. La palabra heurística se deriva del griego “Heuriskein”, que significa encontrar, hallar, descubrir, inventar [27].

Una heurística se puede definir como una técnica que pueden dar una solución óptima para un problema a un costo computacional moderado, pero no garantizan lograr la solución en todos los casos, debido a los factores estocásticos que emplean; las heurísticas fueron adoptadas en el área de la computación, para ser exactos dentro de la rama de la Inteligencia Artificial (IA, por su acrónimo en inglés), dando sustento a los algoritmos estocásticos, que son todos aquellos que manejan el concepto de aleatoriedad, entre los principales encontramos: Algoritmos Genéticos, Enjambres de Partículas, Colonias de Hormigas, Algoritmos evolutivos, Recocido simulado Algoritmos de campaña entre otros [2,5].

La principal ventaja de estas estrategias es que proveen a los algoritmos con algún elemento aleatorio, permitiendo ampliar el espacio de una búsqueda local a una búsqueda global, además de generar diversas trayectorias para solucionar los problemas, emulando el razonamiento humano. Debido a estas características, es posible llevar a cabo su adaptación a problemas diferentes para los que fueron diseñados en un inicio, tomando en cuenta que, su principal función y aplicación son problemas de optimización. El sustento a las heurísticas son todas las teorías referentes a la naturaleza así como del propio ser humano ya sea por sus características de supervivencia, evolución y razonamiento principalmente; en específico algunas de las teorías que han tomado fuerza dentro del área de la ingeniería

en particular en el área de optimización son las teorías de la genética, el comportamiento de los insectos, fenómenos físicos e inclusive de la misma estructura cerebral humana.

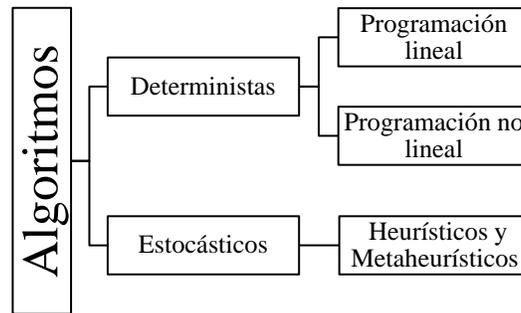


Figura 2.1. Diagrama de clasificación de algoritmos.

De manera general, una heurística requiere los siguientes argumentos:

1. Una representación para las posibles soluciones, esto es la codificación que se usará para las soluciones, comúnmente son representaciones numéricas ya sean de tipo real o cadenas binarias.
2. Una función objetivo que es la función matemática que representa el problema a optimizar.
3. Un método para generación de soluciones iniciales, generalmente se utilizan procesos aleatorios.
4. Uno o varios procesos que generan nuevas soluciones a partir de soluciones generadas inicialmente.

2.3 Ejemplos de algoritmos heurísticos.

En esta sección se detallan algunos de los algoritmos heurísticos más usados en la actualidad mencionados en la sección 2.2, de los cuales se comentaron algunas de sus aplicaciones en la sección 1.5.

2.3.1 Algoritmo evolutivo.

Los algoritmos evolutivos son una abstracción de la evolución biológica, por lo que no se requiere el uso de un operador de recombinación si no de un operador de cambio como el

operador de mutación, así como un operador de selección normalmente probabilístico el cual garantiza una exploración totalmente aleatoria. El algoritmo básico de la programación evolutiva es el siguiente [5]:

- I. Generar aleatoriamente una población inicial.
- II. Proceso de mutación.
- III. Se calcula la aptitud de cada hijo y se usa un proceso de selección estocástico para determinar cuáles soluciones se conservan y cuales se desechan.

Estos algoritmos son la base para la mayoría de algoritmos bioinspirados, debido a que en todos se trabaja con una población inicial y una función de aptitud, las diferencias radican en los procesos que afectan a los individuos de la población ya que dependiendo del enfoque del algoritmo, estos van variando.

2.3.2 Algoritmo Genético.

Los Algoritmos Genéticos (GAs, por sus siglas en inglés) son técnicas estocásticas, se fundamentan en los principios de evolución y selección natural, la cual consisten en que los organismos más fuertes son los que logran sobrevivir al tener la capacidad de adaptarse mejor a su medio. Éstos fueron desarrollados por un investigador de la Universidad de Michigan, John Holland en la década de los 60's, y han sido aplicados y puestos a prueba principalmente en la solución de problemas de optimización en donde han demostrado ser eficientes y confiables [3, 5]. Para terminar la ejecución de un algoritmo genético se debe cumplir con alguna de las dos condiciones:

- a) Fijar un número máximo de iteraciones y detener el algoritmo cuando se llegue a ese máximo.
- b) Permitir que el algoritmo haga iteraciones hasta que el resultado tienda a una misma solución, esto es denominado convergencia del algoritmo.

La estructura de un algoritmo genético se define por los siguientes pasos:

- i. Generar una población aleatoria de N individuos.

- ii. Definir una función de aptitud o función objetivo.
- iii. Evaluar cada individuo de la población en la función objetivo.
- iv. Selección de soluciones a reproducirse.
- v. Cruzamiento de la población.
- vi. Mutación de elementos resultantes del cruzamiento.
- vii. Reemplazo de elementos de la población de la *i*-ésima generación con los mejores elementos de mutación y cruzamiento.
- viii. Detener si se cumple el criterio de paro o volver a evaluar la función objetivo.

Para entender el funcionamiento de un GA obsérvese el diagrama de la figura 2.2, en el que se muestra el proceso que realiza el algoritmo. En términos generales, para simular el proceso de un GA en una computadora se requiere: a) Codificar las posibles soluciones (individuos de la población), b) Una función de aptitud, c) Un mecanismo de selección, y d) Operaciones que afecten a los individuos.

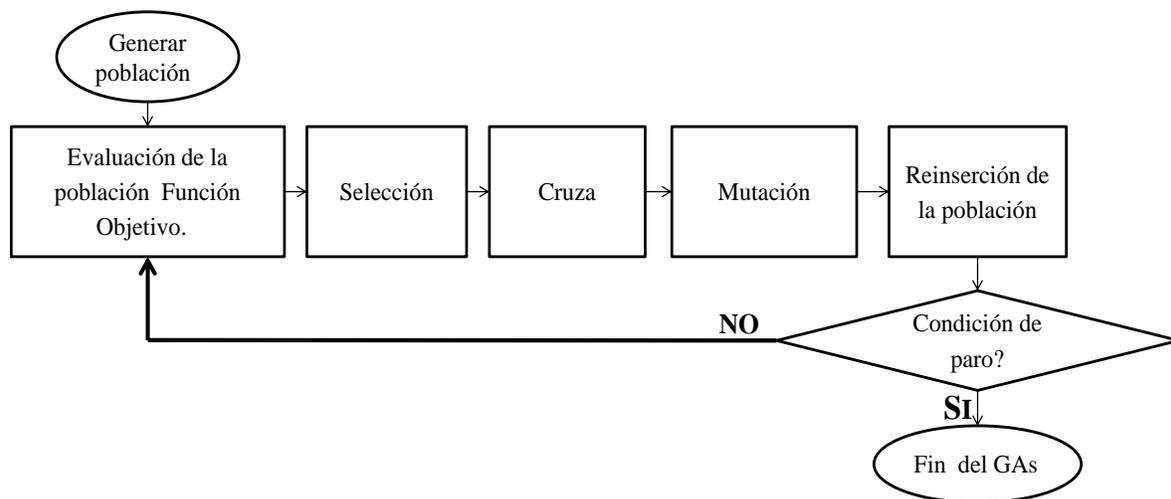


Figura 2.2 Diagrama del funcionamiento de un Algoritmo Genético.

Las principales aplicaciones de los GAs son en problemas de búsqueda y optimización en los cuales los espacios de búsqueda son extensos o para los cuales los algoritmos existentes no logran optimizar los resultados, algunos ejemplo de problemas de aplicación son; el problema del agente viajero, consultas en bases de datos, búsqueda de pesos para redes

neuronales, búsqueda de patrones en tratamiento de imágenes, sintonización de controladores así como múltiples problemas de optimización matemática.

2.3.3 Colonia de Hormigas.

El algoritmo de Optimización por Colonia de Hormigas (*Ant Colony Optimization*, ACO por sus siglas en inglés) se basa en la función de las hormigas para buscar comida. Este proceso señala que la hormiga sale de su hormiguero y comienza a explorar un área, de forma aleatoria en diferentes direcciones al mismo tiempo va dejando un rastro de feromonas que le indican el trayecto recorrido, al encontrar comida, ésta es capaz de regresar al hormiguero siguiendo dicho rastro; cada que una hormiga recorre el trayecto marcado va dejando una nueva capa de feromonas haciendo que la concentración de feromonas muy alta y cada vez más hormigas puedan detectarla y sigan el trayecto a la comida; una vez que las hormigas terminan esa fuente de alimento dejan de recorrer el camino y la feromona se va evaporando[1-3,5]. El algoritmo general se describe a continuación:

1. Comenzar
2. Inicializar (entradas y datos)
3. for c1=1 hasta i
 - for c2=1 hasta h
 - for c3=1 hasta n
 - $\alpha' S \leftarrow$ Construir Solución (c3)
 - $$b' P_{ij}(c3) = \frac{\tau_{ij}^{\alpha} n_{ij}^{\beta}}{\sum \tau_{ih}^{\alpha} n_{ih}^{\beta}}$$
 - end for
 - end for
 - end for
4. Guardar la mejor solución (S)
5. Actualizar rastro τ_{ij}
6. End for
7. Entrega solución (S)
8. Termina

Las variables o datos inmersos en el algoritmo se describen a continuación: **Entrada**; h es número de hormigas, n es número de elementos de combinatoria, α es el parámetro correspondiente al rastro, y β es el parámetro correspondiente a la heurística. **Dato**; i es el contador de las iteraciones, P_{ij} es la matriz de probabilidades, τ_{ij} representa el rastro de feromonas, y n_{ij} es la Matriz del inverso de la distancia. **Salida**; S es la mejor solución.

2.3.4 Optimización por Temple simulado.

El algoritmo de Temple Simulado (*Simulated Annealing*, SA por sus siglas en inglés) es un algoritmo de optimización global originalmente desarrollado para ser aplicado a problemas de optimización combinatoria y numérica, es clasificado como un método de optimización que puede ser aplicado a búsquedas arbitrarias y espacios de problemas. Sólo requiere de un individuo inicial como punto de partida y una operación de búsqueda. Se basa en el principio del templado, que consiste en un tratamiento de material con la finalidad de alterar sus propiedades tal como su dureza y usualmente requiere un proceso en el cual se eleva el material a altas temperaturas, garantizando que la energía almacenada en las partículas del material, sean utilizadas y estas sufran un cambio en su estructura. Después al dejar que el material se enfríe de manera lenta la estructura es transformada al alcanzar un punto de equilibrio [3, 5]. El algoritmo se describe a continuación:

1. Comenzar
 - a) $pnueva.g \leftarrow crear()$
 - b) $pact \leftarrow pnueva$
 - c) $p^* \leftarrow pnueva$
 - d) $t \leftarrow 0$
 - e) mientras $!= criterioDeParo()$ hacer
 - $\Delta E \leftarrow f(pnueva.x) - f(pact.x)$
 - Si $\Delta E \leq 0$ entonces
 - a' $pact \leftarrow pnueva$
 - b' Si $f(pact.x) < f(p^*.x)$ entonces $p^* \leftarrow pact$
 - de lo contrario
 - a' $T \leftarrow obtener\ temperatura(t)$
 - b' Si $random() < e^{-\frac{\Delta E}{k_B T}}$ entonces $pact \leftarrow pnueva$
 - fin de condición Si
 - $pnueva.g \leftarrow mutar(pact.g)$
 - $t \leftarrow t + 1$
 - f) entrega $p^*.x$
2. Termina

Las variables y datos del algoritmo se describen a continuación: **Entrada;** f representa la función objetivo a ser minimizada. **Dato;** $pnueva$ es individuo recién generado, $pact$ es el individuo que está siendo explorado, p^* es el mejor individuo encontrado hasta el momento, T la temperatura del sistema que ha decrecido con el tiempo, t es el índice que representa al tiempo (iteraciones) y ΔE la diferencia de energía entre $xnueva$ y $xact$. **Salida;** X^* es el conjunto de elementos encontrados.

2.3.5 Optimización por Enjambres de Partículas

El algoritmo de enjambre de partículas (*Particle Swarm Optimization* o PSO, por sus siglas en inglés) es una técnica de optimización/búsqueda en el campo del aprendizaje automático. Fue descrito a mediados de la década de los 90's por James Kennedy, se inspira en el comportamiento de los enjambres de insectos en la naturaleza, un ejemplo es en un enjambre de abejas al momento de buscar polen, buscan la región de un espacio de búsqueda en la que existe mayor densidad de flores y se concentran en ella.

En el PSO, las partículas se sitúan en el espacio de búsqueda y se mueven a través del mismo de acuerdo a unas reglas que tienen en cuenta el conocimiento personal de cada partícula y el conocimiento global del enjambre. A través de sus movimientos por la función *fitness* (o función a seguir) las partículas llegan a descubrir valores particularmente altos para $f(x, y)$. Cada partícula (individuo) tiene una posición (x, y) , en el espacio de búsqueda y una velocidad (v_x, v_y) , con la que se mueve a través del espacio. Además, como partículas de un mundo real físico, tienen una cantidad de inercia, que los mantiene en la misma dirección en la que se movían, así como una aceleración (cambio de velocidad), que depende principalmente de las dos características siguientes:

- Cada partícula es atraída hacia la mejor localización que ella personalmente, ha encontrado en su historia (mejor personal).
- Cada partícula es atraída hacia la mejor localización que ha sido encontrada globalmente en el espacio de búsqueda (mejor global).

La fuerza con que las partículas son empujadas en cada una de estas direcciones depende de dos parámetros; atracción-al-mejor-personal y atracción-al-mejor-global, de esta manera a medida que las partículas se alejan de estas localizaciones, la fuerza de atracción es mayor. También se suele incluir un factor aleatorio que influye en cómo las partículas son empujadas hacia estas localizaciones [2, 30].

Las aplicaciones de este algoritmo y los mencionados en las secciones anteriores se dan dentro del área de ingeniería, principalmente para resolver problemas de optimización, sin

embargo gracias a sus atributos de adaptación cada vez son más usados en áreas como economía para predecir comportamientos financieros, en sociología para programación de agentes que permitan reconstruir y simular etapas históricas, así como en ciencias biológicas y geográficas que permitan crear sistemas de predicción de fenómenos naturales.

2.4 Optimización.

Uno de los objetivos de las metaheurísticas en la búsqueda de un resultado óptimo, es decir, optimizar un proceso de búsqueda de parámetros, los cuales son utilizados para la sintonización de un controlador PID.

La optimización es una disciplina de la ingeniería encargada de minimizar o maximizar el valor de una variable. De manera formal se define como calcular o determinar el valor mínimo o el valor máximo de una función con base en sus variables y se puede describir matemáticamente mediante la ec. (2). Durante el proceso de optimización se toman en cuenta las restricciones o limitaciones asociadas a la función, las cuales son expresiones matemáticas que incorporan las características del problema que se desea resolver [29, 30].

$$\min(\max) f(x), x = [x_1, x_2, \dots, x_n]^T \in \mathbb{R}^n \quad (2)$$

$$j = 1, 2, \dots, m$$

$$g_j(x) \leq 0;$$

$$h_j(x) = 0$$

donde $f(x)$, $g_j(x)$ y $h_j(x)$ representan las funciones, los componentes de $x = [x_1, x_2, \dots, x_n]^T$ son denominados variables, $f(x)$ es la función objetivo, $g_j(x)$ y $h_j(x)$ son funciones que describen las condiciones de inequidad e igualdad respectivamente las cuales son restricciones para el problema.

Existe una gran variedad de métodos y técnicas clásicas para optimización de problemas, pero son ineficaces ante problemas con tiempos y espacios de solución exponencial, es ahí donde las heurísticas como técnicas de solución para problemas de búsqueda, optimización y aprendizaje son aplicables; al ser técnicas basadas en un conocimiento previo del problema, permiten generar nuevos algoritmos de optimización que además de eficientes son flexibles y se pueden modificar fácilmente para su aplicación a diversos problemas. Estos algoritmos pueden ser, simples solo utilizando una heurística o híbridos a partir de una serie de combinaciones entre heurísticas (como se describe en la sección 1.5) o bien heurísticas con diversas teorías de áreas de ingeniería, en específico del área de la IA, tal como la teoría de lógica difusa la cual se describe en la siguiente sección 2.5.

2.5 Teoría de Lógica Difusa.

Teórica y prácticamente, las computadoras manejan datos precisos que se reducen a cadenas de unos (1) y ceros (0) y proposiciones que son ciertas y falsas (lenguaje máquina). El cerebro humano puede razonar con información que involucra incertidumbre o juicios de valor como: “es un poquito alto” o “está muy caliente” esto asociado a un sentido común que les permite razonar en un mundo donde las cosas son parcialmente ciertas. La lógica difusa es una rama de la Inteligencia Artificial que le permite a una computadora analizar información del mundo real en una escala entre lo falso y lo verdadero. Los sistemas difusos permiten agregar un grado de humanidad a los sistemas, mediante consideraciones del lenguaje natural ya que cuando se traduce el lenguaje humano al contexto de la lógica clásica se pierde el sentido del significado, pérdida que puede ser importante para diseñar un sistema experto, el cual emula la función de un ser humano especializado en determinada área. La lógica difusa puede controlar o describir el sistema usando reglas de sentido común que se refieren a cantidades indefinidas [2]. Los sistemas difusos frecuentemente tienen reglas tomadas de expertos, pero cuando no hay experto los sistemas difusos adaptivos aprenden las reglas observando cómo la gente manipula sistemas reales [4].

La lógica difusa hoy en día es muy común y se encuentra en diferentes sectores de la tecnología, tanto en la electrónica como el control, las matemáticas, la robótica, etc. El objetivo principal de la lógica difusa es crear un sistema basado en el comportamiento y pensamiento humanos. Esto se logra gracias al planteamiento de un modelo en cualquier contexto y traducirlo a lenguaje humano; el éxito de la lógica difusa se basa en la experiencia. El sistema toma el banco de conocimiento del experto y con él crea sus reglas para desarrollar una propuesta.

Tres son las etapas para obtener un resultado deseado mediante lógica difusa. La primera etapa se basa en un proceso donde las variables tienen un grado de incertidumbre lingüístico. Por lo tanto, el rango de valores (universo de discurso) de cada variable puede clasificarse por conjuntos difusos por ejemplo baja, media, alta. Cuando se miden las variables, los valores pasan a un proceso de “FUSIFICACIÓN” que consiste en pasar dichos valores a un rango de pertenencia entre cero (0) y uno (1). Se busca determinar en qué grado el valor que se está adquiriendo pertenece a un conjunto difuso. Los conjuntos difusos son caracterizados mediante funciones de membresía, las cuales están sintonizadas al punto de operación adecuado para el funcionamiento del sistema.

En la segunda etapa se proponen reglas lingüísticas de “INFERENCIA” que servirán de guía para que el sistema se comporte de manera más adecuada, idónea o deseada según el modelo de referencia o los objetivos del usuario. El grado de pertenencia de cada una de las variables se evalúa en un conjunto de reglas de inferencia. Dichas reglas de inferencia fueron determinadas con ayuda de un experto. El conjunto de reglas de inferencia determina una consecuencia, es decir, asigna un grado de pertenencia a un conjunto difuso que caracteriza a las salidas. Una vez obtenidas las consecuencias, la tercera etapa es un proceso para determinar los valores óptimos de salida, conocido como “DEFUSIFICACIÓN” y que consiste en pasar el grado de pertenencia, proveniente de la consecuencia de la regla de inferencia, a un valor nítido o real. Para hacer esto, el paso previo es sintonizar funciones de membresía de cada una de las salidas, con el fin de obtener un valor cuantificable [2,4]. La figura 2.3 muestra el diagrama de un controlador

difuso además del flujo de los datos a través de los procesos aplicados mediante lógica difusa.

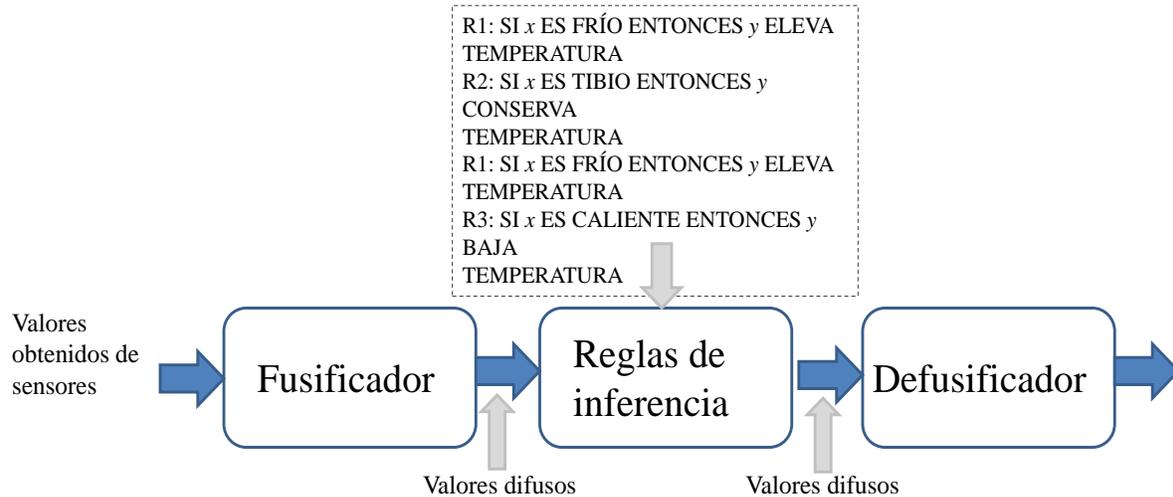


Figura 2.3 Diagrama general de un controlador difuso.

Como se ha descrito hasta esta sección, las heurísticas así como la teoría de lógica difusa aportan características que permiten generar algoritmos complejos, a partir de búsquedas aleatorias y restricciones, los cuales explorara de manera más eficiente los espacios de búsqueda para problemas de optimización, como es el caso de ajuste de controladores tipo PID en el área de ingeniería de control, el cual se detalla en las siguientes secciones.

2.6 Teoría del control moderna.

En la vida cotidiana se interactúa con sistemas de todo tipo, desde los más sencillos hasta los de mayor complejidad, los cuales en ocasiones no es posible dominar, dado que existen sistemas que funcionan de manera casi automática, sin necesidad de la manipulación del ser humano, también es cierto que sin la acción que realiza el ser humano para que los sistemas se comporten como se desea, estos serían inútiles; a todas esas acciones y decisiones que se toman con base en referencias, patrones y comportamientos se le define como control. El control tiene como premisa reducir el error de un sistema, definido como la desviación que existe entre la entrada de referencia del sistema (valor deseado) y la salida del mismo (valor medido). El control tiene sus inicios con la intervención directa del ser humano en sistemas de tipo mecánicos, hidráulicos, electrónicos entre los más comunes. Con el paso del tiempo

y el afán del hombre por facilitar los procesos en los que está involucrado, el control evoluciona de los controles manuales a controles automáticos cada día más autónomos y eficaces [7,10].

2.6.1 Sistema de control.

Un sistema de control es un conjunto de elementos que interactúan entre sí, con el objetivo en común de regular y manipular los procesos que realiza un determinado sistema, con el fin de minimizar o eliminar el error y lograr el funcionamiento óptimo del sistema [7]. Existen diversos tipos de control, pero pueden clasificarse en dos grandes áreas como, controles automáticos y controles manuales, estos se conforman por distintos elementos, descritos en las siguientes secciones.

2.6.2 Fundamentos teóricos de los sistemas de control.

Para entender el funcionamiento de los sistemas de control es necesario conocer algunos conceptos básicos [7, 10], los cuales se definen a continuación:

- 1) **Variable controlada:** es la cantidad o condición que se mide y controla (salida del sistema).
- 2) **Planta:** cualquier objeto o medio físico el cual se va a controlar.
- 3) **Proceso:** operación progresiva que consiste en una serie de acciones o movimientos controlados, sistemáticamente dirigidos hacia un resultado o propósito determinados (acción que se va a controlar).
- 4) **Sistema:** Un sistema es una combinación de componentes que actúan juntos y realizan un objetivo determinado.
- 5) **Perturbación:** señal que tiende a afectar negativamente el valor de la salida de un sistema.

2.6.3 Función de Transferencia.

La función de transferencia de un sistema, es el modelo matemático que describe el funcionamiento de un sistema dinámico, mediante una ecuación diferencial lineal e invariante con el tiempo, constituida por el cociente de la transformada de Laplace de la función de respuesta (salida del sistema), entre la transformada de Laplace de la función de referencia (entrada del sistema), bajo la suposición de que todas las condiciones iniciales son cero [7,10]. Lo anterior se representa mediante la siguiente expresión matemática:

$$G(s) = \frac{O(s)}{I(s)} \quad (3)$$

La función de transferencia $G(s)$ determina el funcionamiento del sistema, al relacionar la función de entrada $I(s)$ con respecto a la función de salida $O(s)$, independientemente de sus magnitudes, la utilidad de $G(s)$ es facilitar el análisis del funcionamiento del sistema que describe, si $G(s)$ es desconocida, se puede obtener experimentalmente introduciendo entradas conocidas (entre las más comunes, funciones rampa, escalón e impulso unitario) y observado la salida del sistema [7].

2.6.4 Diagramas de Bloques.

Un sistema de control puede tener varios componentes, así, un diagrama de bloques es una representación gráfica de las funciones o modelo matemático (en variable tiempo $[t]$ o Laplace $s=j\omega$) que lleva a cabo cada componente del sistema, mostrando las relaciones existentes entre los diversos componentes; esta representación gráfica presenta la ventaja de indicar el flujo de las señales del sistema real (izquierda entrada y derecha salida). En un diagrama de bloques se representan las variables del sistema mediante bloques funcionales, que contiene las funciones de transferencia de los componentes, y se conectan mediante flechas para indicar la dirección de flujo de la señal como se muestra en la figura 2.4.

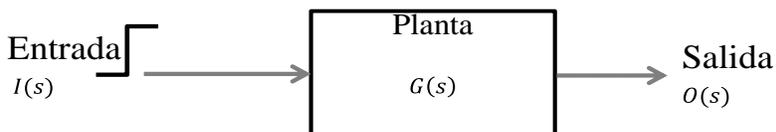


Figura 2.4 Diagrama de bloques con función escalón de entrada aplicada a una planta.

2.6.5 Sistemas de control en lazo abierto.

Se definen como los sistemas en los cuales la salida no afecta la acción de control. Un sistema de control en lazo abierto no mide la salida ni se realimenta para compararla con la entrada. En cualquier sistema de control en lazo abierto, la salida no se compara con la entrada de referencia (véase figura 2.5). Por tanto, a cada entrada de referencia le corresponde una condición operativa fija; como resultado, la precisión del sistema depende de la calibración por parte del usuario de manera empírica. Ante la presencia de perturbaciones, un sistema de control en lazo abierto no realiza la tarea deseada. En la práctica, el control en lazo abierto sólo se usa, si se conoce la relación entre la entrada y la salida y si no hay perturbaciones internas ni externas [7].

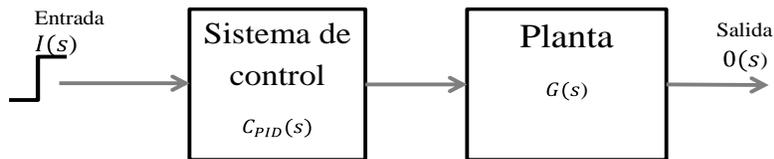


Figura 2.5 Diagrama de bloques de un sistema de control en lazo abierto.

Nótese en la figura 2.5 que la salida del sistema nunca interactúa con la entrada o con el controlador, para manipular el proceso y por tanto es imposible corregir el error automáticamente, por lo tanto la salida en términos de una función matemática en dominio de Laplace queda:

$$O(s) = C_{PID}(s) * G(s) * I(s) \quad (4)$$

Finalmente la función de transferencia queda como:

$$F.T_{LA} = \frac{O(s)}{I(s)} = C_{PID}(s) * G(s) \quad (5)$$

donde, $F.T_{LA}$ representa la función de transferencia de un sistema de control en lazo abierto, $C_{PID}(s)$ representa la función de transferencia de un controlador PID y $G(s)$ representa la función de transferencia de la planta.

2.6.6 Sistemas de control en lazo cerrado.

También denominados sistemas de control retroalimentados; los términos control retroalimentado y control en lazo cerrado se usan indistintamente. En un sistema de control

en lazo cerrado, se alimenta al controlador con la señal de error de actuación, que es la diferencia entre la señal de entrada y la señal de salida, a fin de reducir el error y llevar la salida del sistema a un valor deseado. En este diagrama de la figura 2.6 se puede observar como la salida retroalimenta la entrada para poder medir el error automáticamente y corregirlo en la medida de lo posible, expresado esto en términos de una función de transferencia en dominio de Laplace se define como:

$$F.T_{LC} = \frac{O(s)}{I(s)} = \frac{C_{PID}(s)}{1+C_{PID}*G(s)} \quad (6)$$

donde, $T.F_{LC}$ expresa la función de transferencia de un sistema de control en lazo cerrado.

2.6.7 Teoría de Control Moderna.

Desde la década de los 60's comenzó la evolución de la teoría de control, debido a la necesidad de alcanzar los requerimientos, cada vez más restrictivos en el desempeño de los sistemas, así como el aumento en su la complejidad y el acceso fácil a tecnologías cada día más poderosas, de este modo surge la denominada teoría de control moderna, que es un nuevo enfoque del análisis y diseño de sistemas de control complejos. La teoría de control moderna se aplica a sistemas con entradas y salidas múltiples, que pueden ser lineales o no lineales, en tanto que la teoría de control clásica sólo se aplica a sistemas lineales con una entrada y una salida e invariantes con el tiempo, otro punto importantes es que la teoría del control moderna se enfoca esencialmente en el dominio del tiempo, mientras que la clásica se basa en el dominio de la frecuencia.

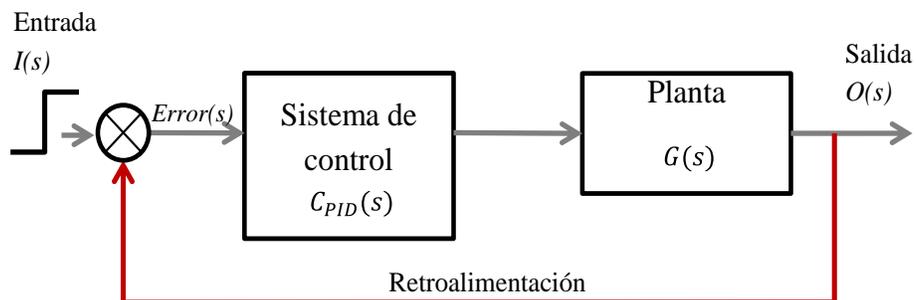


Figura 2.6 Diagrama de bloques de un sistema de control en lazo cerrado.

2.6.8 Respuesta en estado estable y transitorio de un sistema de control.

Se han descrito algunas de las herramientas para el análisis de un sistema, en primer lugar, un modelo matemático que permita describir el comportamiento del sistema (función de transferencia), así como también representaciones gráficas (diagramas de bloques) con los que es posible manipular fácilmente dichas funciones para su análisis, en segundo lugar es preciso mencionar que las señales de prueba son funciones del tiempo muy simples, usadas como entradas para los sistemas y con ellas realizar análisis matemáticos, experimentales y observar el comportamiento del sistema, por lo regular estas señales son funciones escalón, rampa, parábola, impulso y senoidal.

Toda vez que se obtiene un análisis del funcionamiento, es preciso realizar la evaluación del desempeño y rendimiento del sistema, este análisis es posible hacerlo mediante la respuesta del sistema, la cual está compuesta por una respuesta en estado estable y una respuesta transitoria, como a continuación se describen:

1. **Respuesta transitoria:** Se hace referencia a la parte de la respuesta que va del estado inicial al estado final.
2. **Respuesta en estado estable:** Se define a la manera en la cual se comporta la salida del sistema conforme el tiempo (t) tiende a infinito [7, 10].

Para conocer el rendimiento de un sistema mediante la respuesta del sistema de control; se utilizan las características más significativas las respuesta en su mayoría son determinadas en el tiempo, debido a que casi todos los sistemas de control, son sistemas en el dominio del tiempo; es decir, deben presentar respuestas de tiempo aceptables (rápidas). La figura 2.7 ilustra de manera gráfica las características de rendimiento obtenidas de la respuesta de un sistema y que se describen a continuación:

- a. **Tiempo de retardo t_d :** el tiempo de retardo es el tiempo requerido para que la respuesta alcance por primera vez la mitad del valor final.

- b. **Tiempo de levantamiento r_t** : el tiempo de levantamiento es el tiempo requerido para que la respuesta pase del 10 al 90%, del 5 al 95% o del 0 al 100% de su valor final, suele usarse el tiempo de levantamiento de 10 a 90%.
- c. **Tiempo pico t_p** : el tiempo pico es el tiempo requerido para que la respuesta alcance el primer pico del sobrepaso.
- d. **Máximo sobreimpuesto M_p** : es el valor pico máximo de la curva de respuesta, medido a partir de la unidad. Si el valor final en estado estable de la respuesta es diferente de la unidad, es común usar el porcentaje del máximo sobre impulso.
- e. **Tiempo de asentamiento s_t** : el tiempo de asentamiento o establecimiento; es el tiempo que se requiere para que la curva de respuesta alcance un rango alrededor del valor final del tamaño especificado por el porcentaje absoluto del valor final (por lo general, de 2 a 5%) y permanezca dentro de él [7].

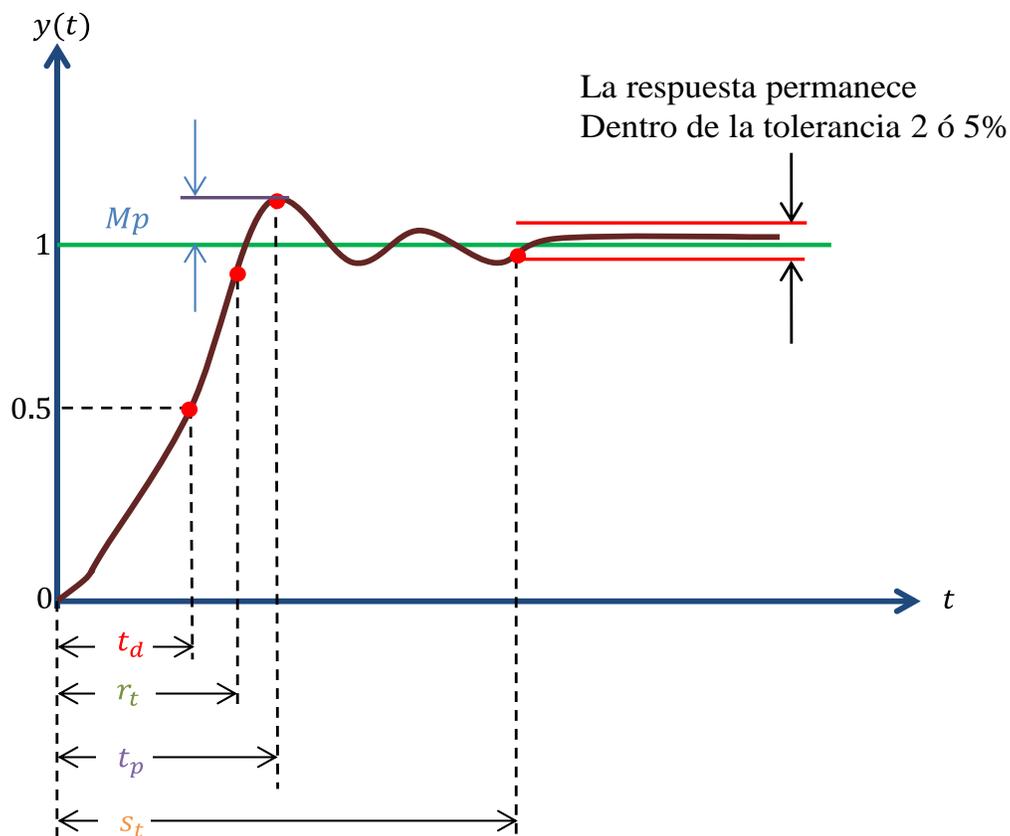


Figura 2.7 Grafica de la respuesta de un sistema de control a una entrada de tipo escalón.

2.6.9 Tipos de controladores.

Un controlador automático compara el valor real de la salida de una planta con la entrada de referencia (el valor deseado), determina la desviación o error y produce una señal de control que reducirá la desviación a cero o a un valor pequeño. La manera en la cual el controlador automático produce la señal de control se denomina acción de control [7]. Existen una variedad de controladores, entre los más usados a nivel industrial, se clasifican de la siguiente forma de acuerdo con sus acciones de control:

- a) De dos posiciones o de encendido y apagado (ON/OFF).
- b) Proporcionales.
- c) Integrales.
- d) Proporcionales-integrales.
- e) Proporcionales-derivativos.
- f) Proporcionales-integrales-derivativos.

Aunque también podrían clasificarse según la energía que utilizan para su operación como neumáticos, electrónicos e hidráulicos. Para el caso de estudio de la presente tesis se empleara un controlador tipo PID clasificado de acuerdo a sus acciones de control Proporcional-Integral-Derivativa, aplicado a un sistema de tipo electrónico.

2.6.10 Controlador tipo Proporcional-Integral-Derivativo (PID).

La combinación de una acción de control proporcional, una acción de control integral y una acción de control derivativo se denomina acción de control proporcional-integral-derivativo (PID). Esta acción combina las ventajas de cada una de las tres acciones de control individuales y se define matemáticamente como:

$$G_{pid}(s) = Kp(1 + \frac{1}{Tis} + Tds) = Kp + \frac{Ki}{s} + Kds \quad (7)$$

donde $G_{pid}(s)$ representa la función de transferencia de un controlador tipo PID (Kp, Ki, Kd), donde Kp es la ganancia del control proporcional, Ki es la ganancia del control integral y Kd es la ganancia del control derivativo.

En la acción proporcional, el controlador utiliza una "proporción" del error del sistema para controlar el sistema, su desventaja es que introduce un error de desviación en el sistema. La acción integral, elimina el desfase introducido por el control proporcional, pero introduce un retardo de fase en el sistema y puede producir una respuesta oscilatoria. La acción derivativa por su parte se utiliza para reducir o eliminar el exceso de tiempo y presenta una acción de avance de fase que elimina el desfase introducido por la acción integral así como la reducción de oscilaciones estabilizando de esta manera el sistema, así, la suma del efecto de los tres tipos de control integrados en uno solo, tienen un efecto que puede optimizarse de tal manera que se tenga el efecto deseado [7, 10].

El correcto funcionamiento de un controlador tipo PID depende estrictamente de la sintonización de los parámetros de ganancias que lo integran (K_p, K_i, K_d) de ahí la importancia de escoger cuidadosamente el método o técnica con el cual se han de sintonizar dicho parámetros. La sintonización de los parámetros (K_p, K_i, K_d) se refiere al ajuste del valor de los parámetros (ganancias), de tal manera que al combinarse en el controlador PID se reduzca el error del sistema.

2.6.11 Métodos de sintonización para controladores PID.

Existe diversos métodos para la sintonización de los parámetros de un controlador tipo PID desde sintonización manual basada en la observación, hasta métodos más sobresalientes, basados en las respuestas gráficas de los sistemas, tal es el caso del método gráfico Ziegler-Nichols el cual logra sintonizar los parámetros del controlador, pero no encontrar los parámetros óptimos para el mismo; a diferencia de estos, existen métodos de búsqueda estocásticos, capaces no solo de sintonizar dichos controladores si no de optimizar los parámetros del controlador tipo PID, proporcionando resultados aceptables como es el caso de los Algoritmos Genéticos, Optimización por Enjambre de Partículas, Controladores difusos, así como la combinación de estos en Algoritmos Híbridos como se menciona en la sección 1.5.

A. Método de Ziegler-Nichols.

Este método sugiere reglas para sintonizar los controladores PID (lo cual significa establecer valores K_p, K_i, K_d). Las reglas de Ziegler-Nichols, son muy convenientes cuando no se conocen los modelos matemáticos de las plantas, estas reglas se establecen con base en las características de respuesta transitoria de una planta específica. Existen dos métodos denominados reglas de sintonización de Ziegler-Nichols. En el primer método se limita el sistema a la gráfica de salida del sistema aplicando una entrada escalón ya que la salida de sistema debe adoptar forma de **S** para poder sintonizar los parámetros y el segundo método requiere de una salida amortiguada para poder establecer las reglas de sintonización, la figura 2.8 describe gráficamente el funcionamiento del primer método.

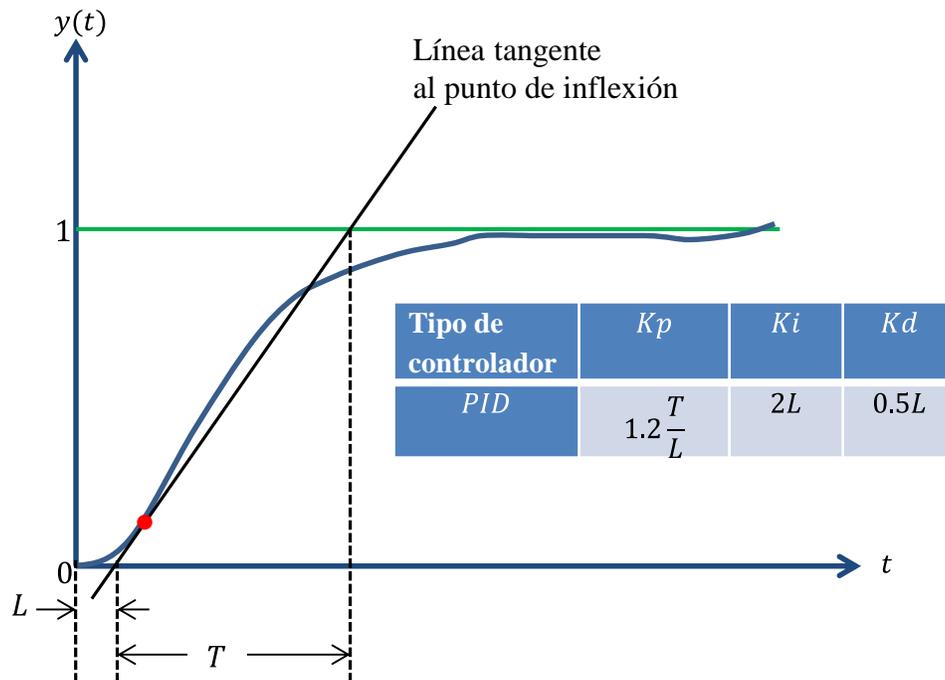


Figura 2.8 Aplicación del método de Ziegler-Nichols a un sistema de control.

La figura 2.8 muestra las reglas de sintonización de parámetros PID mediante el método Ziegler-Nichols, donde L es el tiempo de retardo y T la constante de tiempo. El tiempo de retardo y la constante de tiempo se determinan dibujando una recta tangente en el punto de inflexión de la curva con forma de **S** y determinando las intersecciones de esta tangente con el eje del tiempo y la línea de referencia como se muestra.

Limitantes graficas como las de Ziegler-Nichols, permiten que algoritmos heurísticas tomen cada vez más terreno dentro del área de control, al ser capaz de sintonizar parámetros para los controladores, con tan solo obtener una función objetivo que permita evaluar el rendimiento del sistema con cada posible solución que generada por los algoritmos.

2.7 Aplicación del control automático en Electrónica de Potencia.

La electrónica de potencia es una de las ramas de la Electrónica que más se ha desarrollado en las últimas décadas y es la responsable de la conversión de la forma de energía eléctrica disponible a otra adecuada para la carga que se alimentara. En otras palabras, la electrónica de potencia es la disciplina que estudia los sistemas, encargados de realizar la transformación de la energía eléctrica (convertidores de energía estáticos) en sus distintas formas (corriente continua, corriente alterna). Para realizar este proceso de transformación de la energía se basa en la conmutación “controlada” de dispositivos semiconductores de potencia utilizados como interruptores electrónicos. La electrónica de potencia combina la electrónica y el control, entre otras áreas que interaccionan entre sí, convirtiéndose en un sistema complejo y totalmente completo capaz de denominarse máquina; un ejemplo claro es un aerogenerador o generador a partir de energía eólica. La electrónica por su parte se encarga de los dispositivos y circuitos de estado sólido (semiconductores) para el procesamiento de la energía y el control se encarga de la respuesta en estado estable del circuito para su funcionamiento óptimo, a este tipo de sistemas se les nombra convertidores de energía estáticos [13,14].

2.8 Convertidores estáticos de energía.

Así se le nombra al sistema electrónico conformado por un conjunto de semiconductores, capaz de realizar la conversión de energía, como se describe en la sección 1.1, existen diferentes tipos, su objetivo principal es manejar y convertir la energía. Por lo anterior estos convertidores son hoy en día uno de los principales objetos de estudio del área de la electrónica de potencia. Se le clasifica como estático, debido a que no hay partes en movimiento en el sistema, como sucede en un convertidor basado en estator-rotor, para

convertir energía tal como un generador eléctrico, al cual se le clasifica como convertidor dinámico de energía [14, 15].

A. Convertidor estático de energía de tipo Multicelular.

Este tipo de convertidor de energía es una topología derivada de los convertidores de tipo multinivel como se describen en la teoría [15]. Los convertidores multicelulares, están constituidos mediante arreglos de semiconductores de potencia y capacitores flotados, funcionando como Fuentes Ideales de Voltaje (VSI, por acrónimo en inglés) o de corriente (CSI, por acrónimo en inglés) [14]. Existe una amplia diversidad de convertidores de energía estáticos, sin embargo en este trabajo nos enfocaremos únicamente en un convertidor multicelular de tres células, el cual ya ha sido diseñado y probado en otros trabajos [14, 15]. Este tipo de convertidores se fundamentan en células de conmutación, las cuales están compuestas por un par de semiconductores de potencia (Mosfet y Mosfet1) como se aprecia en la figura 2.9, utilizados como interruptores electrónicos, además de un capacitor flotado C_n y C_{n+1} donde n es el capacitor ubicado en la n -ésima posición y $n+1$ en la $n+1$ -ésima posición, tal y como se describe en la figura 2.10.

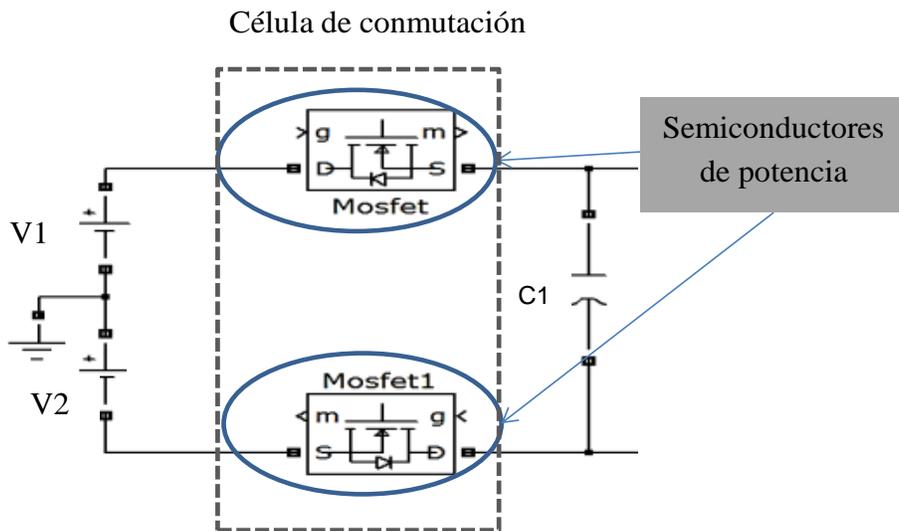


Figura 2.9 Diagrama de una celula basica de conmutacion de un convertidor multicelular.

2.8.1 Topología del Convertidor Multicelular.

Estos convertidores tienen como principio de funcionamiento, distribuir el voltaje de alimentación del convertidor a través de las células de conmutación asociadas en serie que lo conforman. El objetivo consiste en asegurar que el voltaje de bloqueo sea el mismo en cada uno de los interruptores que conforman cada célula de conmutación, de manera que en cada estado de conmutación el voltaje total aplicado al convertidor se distribuya de manera equitativa a través de los interruptores que están en estado de bloqueo. Para el funcionamiento correcto de las células básicas de conmutación de un convertidor multinivel, los interruptores que las conforman en ningún instante de tiempo deberán presentar un estado de conducción de manera simultánea, ya que en esta condición las fuentes flotadas de voltaje estarían en corto circuito y tampoco deberán bloquearse al mismo tiempo ambos interruptores ya que se establecería un circuito abierto desde la fuente de alimentación de voltaje hacia la carga, estableciendo un circuito abierto para la fuente de corriente. Por lo anterior, los dispositivos de conmutación de la célula deben tener estados complementarios de conmutación y además con el objeto de asegurar las condiciones de funcionamiento y para compensar los retardos de apagado, se deben adicionar a las señales de conmutación tiempos de seguridad denominados también como tiempos muertos los cuales son tiempos en el que ambos semiconductores de la célula de conmutación permanecen apagados, evitando con ello efectos de corto circuito [14].

Este tipo de convertidores están conformados por la asociación de células de conmutación, el número de células puede ir desde dos, hasta n cantidad de células, en la figura 2.10 se observa el modelo general de un convertidor multicelular de n células colocadas en serie donde Pn representa el número máximo de células de conmutación, cada célula se encuentra separada por dos fuentes ideales de voltaje implementadas mediante capacitores flotados ($C1, C2, \dots, Cn$), nombrados así porque no tienen ninguna referencia a tierra. Con base en lo anterior un convertidor integrado por Pn células tendrá asociados $Cn-1$ capacitores flotados, V_E es el voltaje de entrada y queda descrito mediante la ec. (8), V_k es el voltaje en el capacitor definido por la ec. (9) [15].

$$V_E = V1 + V2 \quad (8)$$

$$V_k = \frac{k \cdot V_E}{n} \quad (9)$$

El voltaje que debe soportar cada interruptor de la célula de conmutación de referencia (célula enmarcada en la figura 2.10) cuando está apagado, es el mismo para cada uno de los interruptores de las demás células de conmutación que conforman el convertidor y queda definida en [15] mediante:

$$k \frac{V_E}{n} - (k - 1) \frac{V_E}{n} = \frac{V_E}{n} \quad (10),$$

para $k=1, \dots, n-1$.

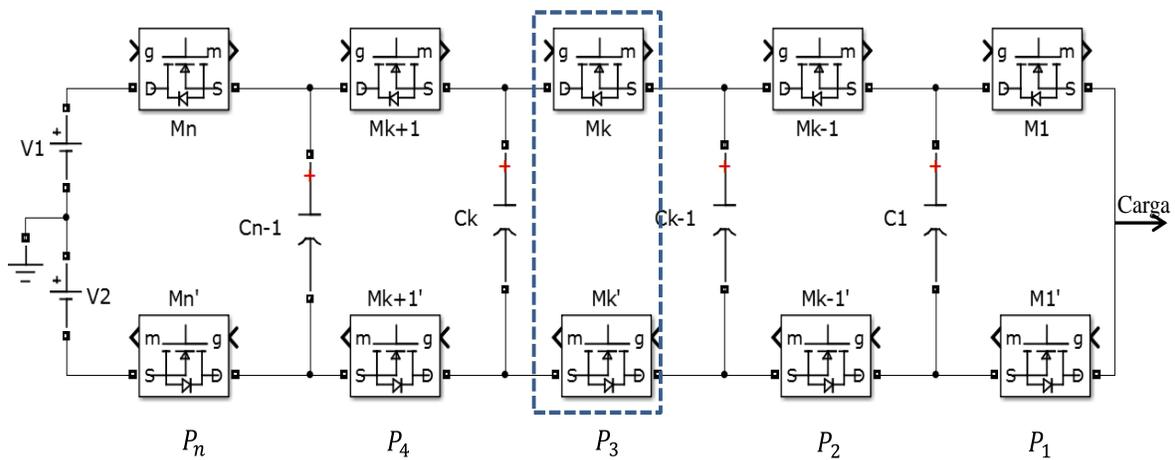


Figura 2.10 Esquema electrónico generalizado de un convertidor multicelular en medio puente.

2.8.2 Equilibrio de voltaje en los capacitores flotados.

En los convertidores multicelulares es fundamental que el voltaje en cada uno de los capacitores sea el establecido, en el caso del inversor de tres células, y calculando los valores mediante la ec. (9), se define que el valor de $VC1$ es igual a una tercera parte del voltaje de entrada del convertidor, mientras que el valor de $VC2$ es dos terceras partes del voltaje de entrada (ver figura 2.11), además estos voltajes deben presentar la mínima desviación posible en cualquier condición de trabajo [14], en caso contrario, aumenta la distorsión armónica en la salida y los interruptores deben soportar sobretensiones que incluso pueden provocar daño físico de los componentes, lo anterior es una cuestión fundamental para el funcionamiento de estos convertidores.

En la literatura, la tarea de mantener la distribución o equilibrio del voltaje en las terminales de los capacitores, suele asumirse como el funcionamiento normal del convertidor, ya que de manera general mediante la implementación de una estrategia de conmutación apropiada, es posible establecer un equilibrio natural de potenciales aunque no todo el tiempo se logra esto, de ahí la necesidad por implementar una estrategia de control, que se encargue de regular las variables deseadas (corriente y voltaje).

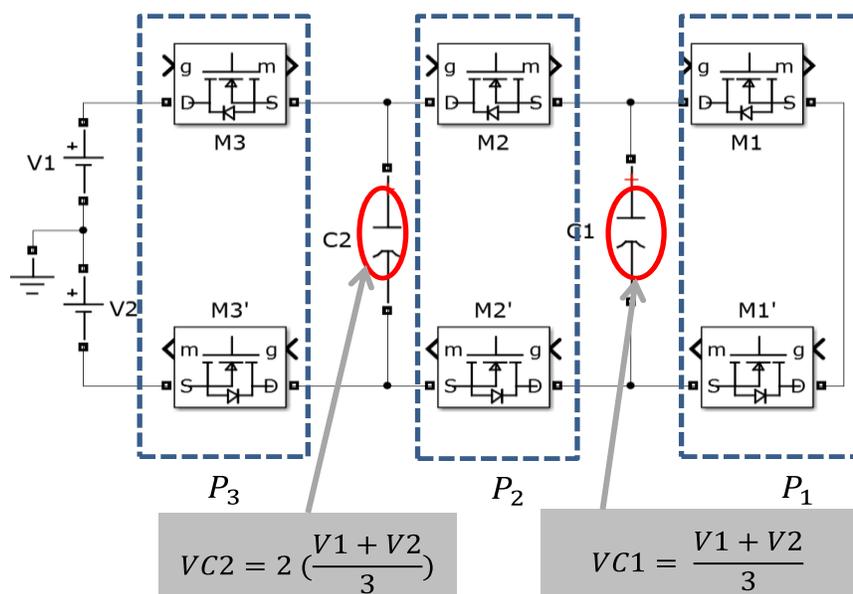


Figura 2.11 Convertidor multicelular de tres células.

El equilibrio natural de los capacitores flotados se logra bajo dos condiciones importantes de la estrategia de conmutación: a) el ciclo de trabajo debe ser el mismo para todas las señales de conmutación de los semiconductores y, b) la fase entre las señales debe ser $2\pi/3$ [15]. Debido a lo anterior, el control de voltaje de los capacitores flotados en los convertidores multinivel se efectuará mediante la implementación de un controlador tipo PID el cual será sintonizado y optimizado a partir de un Algoritmo Híbrido.

2.8.3 Estrategias de Conmutación para convertidores de energía.

Como parte importante en el funcionamiento del convertidor, se han mencionado las señales de conmutación, que deben aplicarse a los interruptores que conforman las células

básicas del convertidor, estas señales se encargan de controlar el tiempo de activación y desactivación de los dispositivos semiconductores, esto puede lograrse mediante una técnica de Modulación de Ancho de Pulso (PWM, por su acrónimo en inglés). Un PWM es una señal digital cuadrada donde la frecuencia es constante pero la fracción de tiempo en que la señal permanece activa (ciclo de trabajo) es variable entre 0 y 100% como se aprecia en la figura 2.12. Existen diversas técnicas para la generación de PWM aplicadas a convertidores multinivel y pueden clasificarse con base a la frecuencia de operación de los dispositivos semiconductores. Algunas se enlistan a continuación: modulación de alta y baja frecuencia, mientras que de acuerdo al tipo de algoritmo implementado, estos métodos pueden clasificarse como: a) modulación por vectores de espacio y b) algoritmos basados en niveles de voltaje. Un ejemplo de estrategia PWM es la conocida como modulación por ancho de pulso con cambio de fases (PS-PWM, por su acrónimo en inglés) la cual genera un número i de señales portadoras asociadas a cada una de las células de conmutación que se encuentran desfasadas un ángulo establecido por $360^\circ/i$. Las portadoras son moduladas mediante una señal senoidal con frecuencia inferior a la de las portadoras, produciendo como resultado i trenes de pulsos modulados [15].

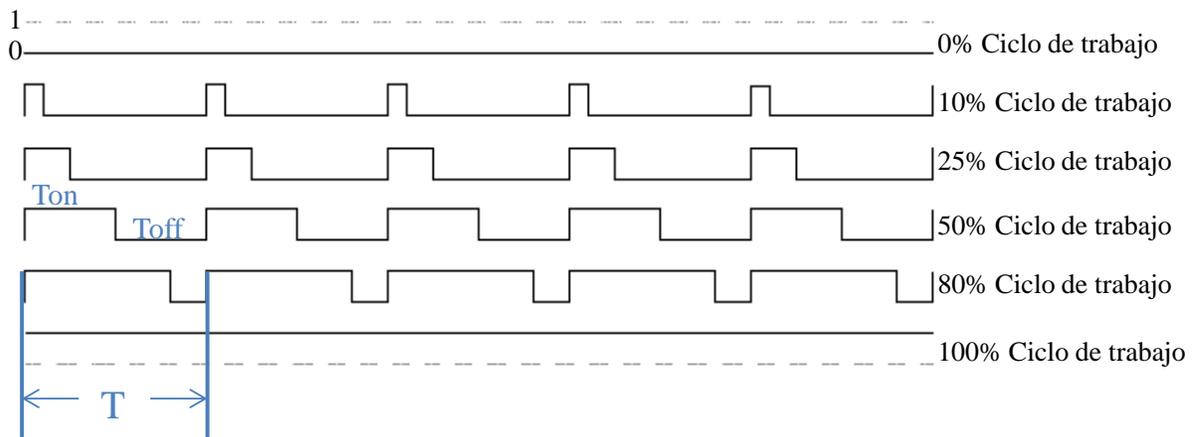


Figura 2.12 Comportamiento del Ciclo de trabajo de señal PWM.

Otra estrategia para generar señales PWM es mediante contadores (temporizadores), estos realizan un conteo generando una señal triangular o diente de sierra y mediante un registro que almacena el punto de corte de la señal se indica el tiempo de encendido de la señal Ton, que es definido como ciclo de trabajo del PWM y un Toff, nombrado tiempo de

apagado de la señal, la suma de los dos estados de la señal es igual al periodo T de la señal como se aprecia en la figura 2.12.

En el siguiente capítulo se detalla la estrategia PWM utilizada e implementada para generar las señales de conmutación para el inversor de tres celdas así como los circuitos de acoplamiento y aislamiento para las señales.

Capítulo 3

Diseño y Desarrollo del sistema

En este capítulo se describe el desarrollo del trabajo realizado, se presenta el procedimiento para la construcción del inversor multicelular de tres células, los circuitos de acoplamiento para las señales de conmutación, la generación de éstas mediante un FPGA y la puesta en funcionamiento del sistema en lazo abierto; asimismo se describe el diseño y codificación del programa del Algoritmo híbrido. Finalmente se presenta la implementación del sistema de control en lazo cerrado.

3.1 Propuesta y diseño del inversor multicelular de tres células.

La construcción del inversor de tres células está integrada por las siguientes etapas: a) Construcción del inversor de tres células, b) Etapa de alimentación de CD para el inversor, c) Generación de señales de conmutación mediante FPGA, d) Etapa de señales complementarias y tiempos de seguridad y e) Acoplamiento y aislamiento de señales de conmutación, tal y como se describe en la figura 3.1 en donde se muestra el diagrama del sistema completo. Cada etapa se describe de manera particular en las siguientes secciones de este capítulo.

Cabe resaltar que el diseño electrónico del inversor no es objeto de estudio en este documento y que se tomó como referencia el diseño realizado en [15] dada la experiencia y óptimo funcionamiento presentado en ese trabajo. La principal diferencia del diagrama general del inversor presentado en la figura 3.1 respecto con el trabajo realizado en [15], está en la generación de las señales de conmutación de las células de inversor, debido a que en este trabajo se generan mediante un Arreglo de Compuertas Lógicas Programables (FPGA, por su acrónimo en inglés) y las señales proporcionadas son de tipo cuadrada, por lo tanto, la salida del inversor será una onda cuadrada de CA, mientras que en el trabajo presentado por Flores-Fuentes *et al.*, las señales de conmutación se generan a partir de un

sistema analógico-digital, con una modulación por ancho de pulso utilizando como portadora una señal senoidal dieciséis veces inferior a la frecuencia de conmutación de los interruptores [15].

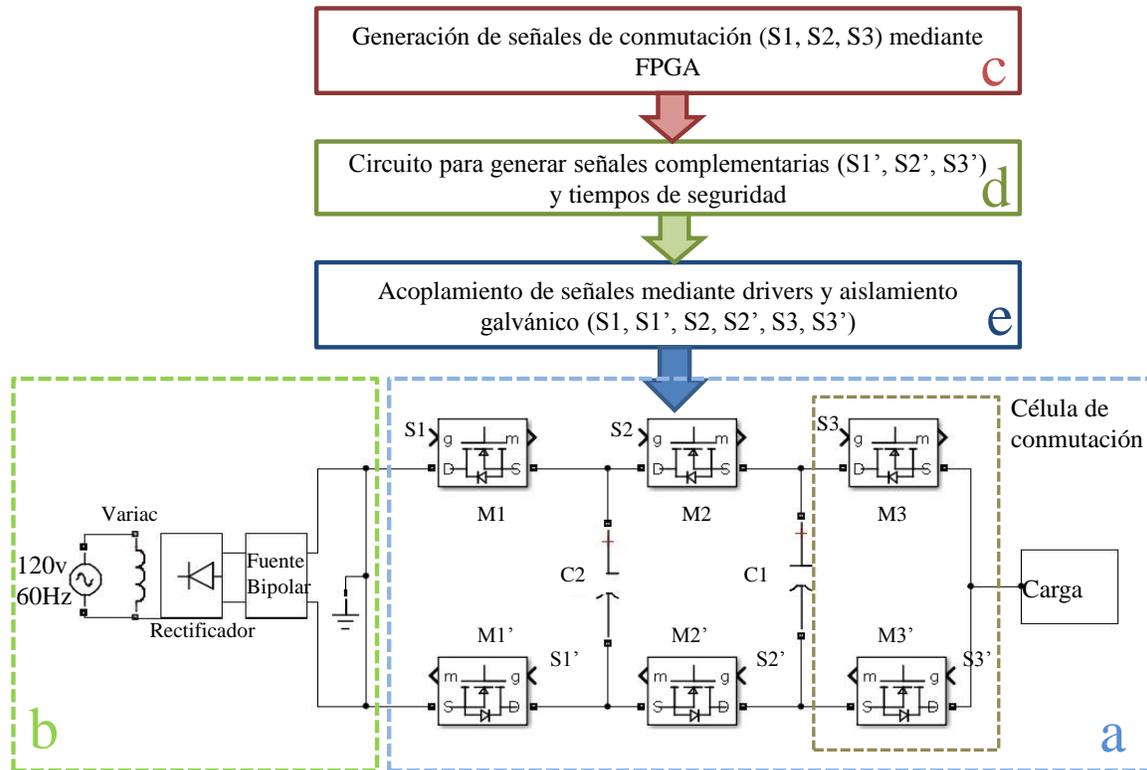


Figura 3.1 Diagrama general del inversor de tres células.

3.1.1 Construcción del inverso de tres células.

Las tres células de conmutación esta conectadas en serie conformando un convertidor de medio puente como se aprecia en la figura 3.2. Cada célula de conmutación está integrada a partir de dos MOSFETS (superior M3 e inferior M3' para el caso de la tercera célula como se aprecia en la sección a de figura 3.1), los cuales fueron implementados mediante la matrícula ATP6030BVR, debido a las características de operación del este semiconductor permitiendo manejar un voltaje máximo de 600V y una corriente de hasta 21A [31]. De igual forma se integraron los capacitores flotados característicos del diseño de este tipo de convertidores como se describió en la sección 2.8.1. Por último se agregó una red de protección (Red Snubber) calculada en [15] y cuya principal función es suprimir picos de voltaje y amortiguar la oscilación transitoria provocada cuando se abre el interruptor de esta

manera se evitan someter a estrés electrónico a los semiconductores de potencia que integran al inversor. La Red Snubber absorbe la energía procedente de la fuente de alimentación del circuito durante el proceso de conmutación, controlando de esta manera parámetros tales como, el incremento de voltaje dv/dt o corriente di/dt en el interruptor.

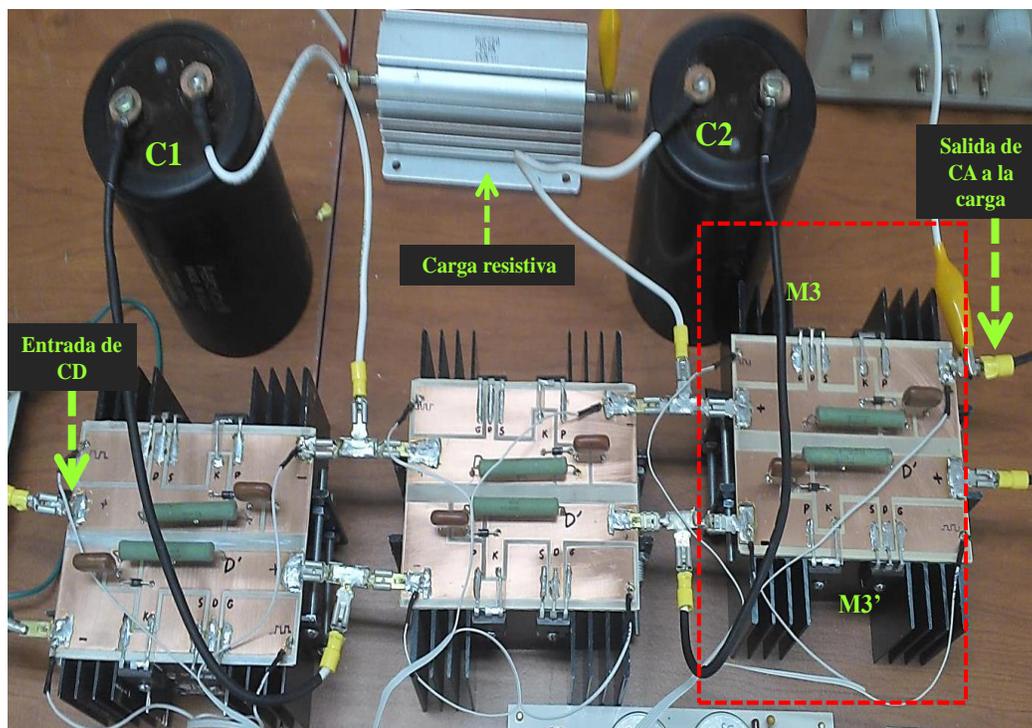


Figura 3.2. Inversor de tres células ensamblado en el laboratorio.

3.1.2 Construcción de la etapa de alimentación CD del inversor

La etapa de alimentación de CD del inversor se describe en la sección b de la figura 3.1 y está conformada por un Variac ESTACO ENERGY 1520CT, mediante el cual se varía la alimentación de la línea de CA desde 0 hasta 220V a un frecuencia de 60 Hz, esta energía es transferida a un puente rectificador de onda completa para su conversión a CD y que a su vez alimenta un divisor capacitivo de voltaje para constituir una fuente bipolar con referencia a tierra con la que es alimentado el inversor de tres células; la construcción de esta etapa se aprecia en la figura 3.3.

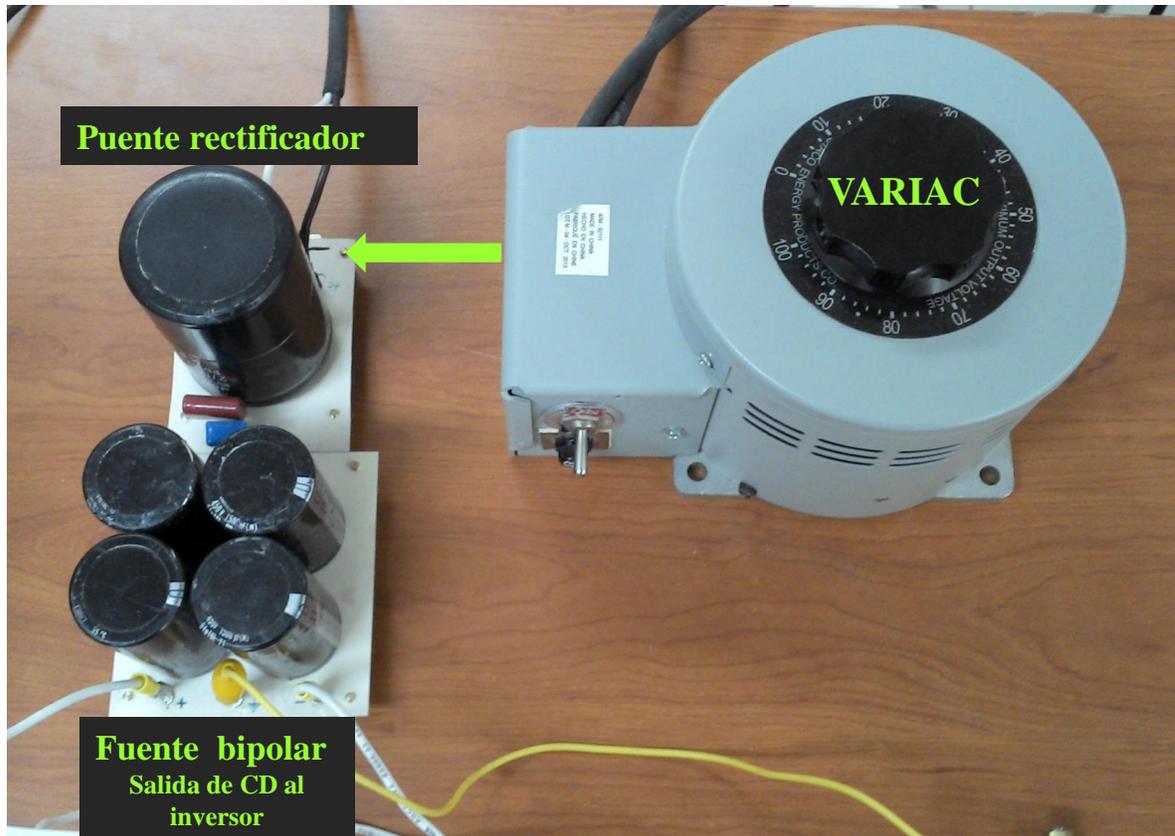


Figura 3.3. Etapa de alimentación de CD del inversor de tres células.

3.1.3 Generación de señales de conmutación mediante FPGA

Para la generación de las señales de conmutación se propusieron diversas tarjetas digitales de desarrollo tales como: a) DSP (por sus siglas en ingles *Digital Signal Processor*) el cual fue descartado debido a que el modelo de la tarjeta es obsoleta, respecto al año de fabricación y a la compatibilidad con sistemas operativos actuales (Windows); b) Se probó con una tarjeta de desarrollo Arduino® UNO (basada en arquitectura con microcontrolador), mediante esta tarjeta era posible genera las señales, con la frecuencia y el desfase necesario, pero fue descartada debido a que el tiempo de ejecución de instrucciones, limitaba el control del ciclo de trabajo de las señales de conmutación, aspecto que es de suma importancia para el control de inversor de tres células; c) Como última opción se planteó el uso de una arquitectura que nos permitiera generar el hardware necesario para la creación y manejo de las señales de conmutación, esta es una arquitectura

basada en un FPGA, este dispositivo tiene la principal característica de permitir programar hardware a la medida y se detalla a continuación.

Un FPGA (*Field-Programmable Gate Array*, por su traducción en inglés) es un circuito integrado que puede configurarse para llevar a cabo todo tipo de función lógica que se programe, siguiendo la especificación de un Lenguaje de Descripción de Hardware (HDL, por sus siglas en inglés y traducción en el mismo como *Hardware Description Language*) el cual permite programar el dispositivo por líneas de código o por esquemas electrónicos. Así, un FPGA permite diseñar circuitos lógicos simples mediante lógica secuencial y combinatoria tal como compuertas lógicas, registros, flip-flop, sumadores, contadores hasta bloques de circuitos más complejos como se requiera, es decir permite crear hardware a la medida del usuario [32].

Como se describió en la sección 2.8.3, para el funcionamiento de las células del inversor se requiere de señales de conmutación, las cuales se generaron mediante una estrategia PWM simple, implementada en un *FPGA SPARTAN 3A* del fabricante XILINX® y cuyas características principales se muestran en la tabla 1 [33].

Las señales PWM se generan a partir de un temporizador, integrado por un registro de 16 bits el cual lleva un conteo incrementable en 1 a razón del flanco de subida del reloj principal del FPGA, comenzando desde 0 hasta N, que es el límite del conteo y se obtiene aplicando un divisor de frecuencia dado que el reloj principal del FPGA es de 50 MHz. Por consideraciones de diseño se elige una frecuencia de 30kHz dado que es un rango de frecuencia no audible para la percepción humana, y además de que es una frecuencia de trabajo encontrada en la literatura. Así con base en este parámetro, se establece la ec. (11). Una vez que se establece el límite del conteo la salida se activa cuando el temporizador está en 0, y se apaga cuando el temporizador coincide con N o con el registro que guarda el ciclo de trabajo de la señal. Este modo se conoce como modo *Fast PWM* y se describe en la figura 3.4.

$$N = \frac{\text{Frecuencia clk}}{\text{Frecuencia deseada}} \quad (11)$$

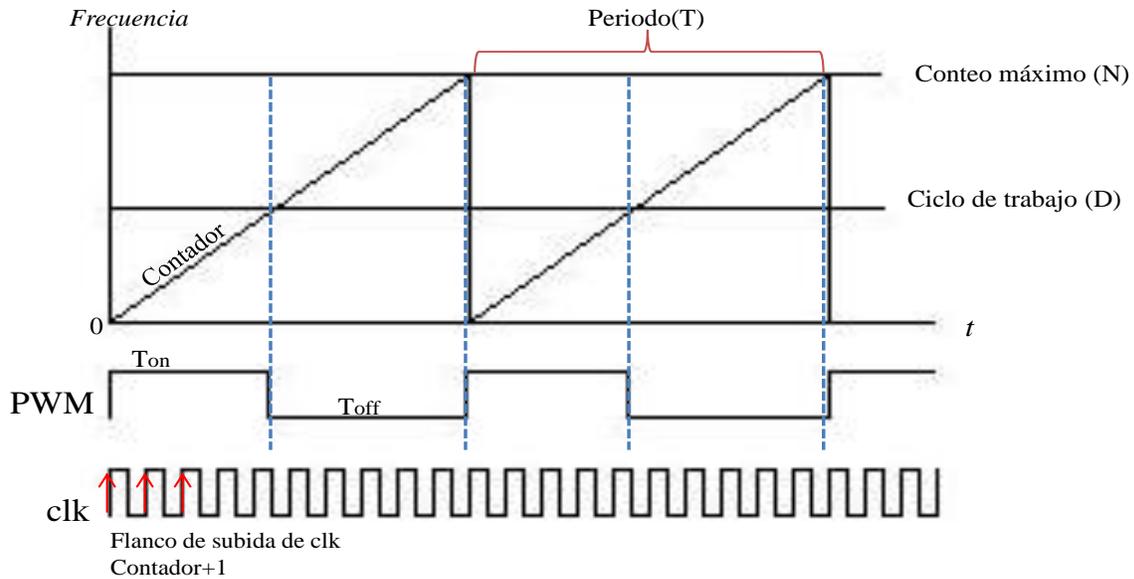


Figura 3.4. Modo FastPWM mediante FPGA

Tabla 1. Resumen de características generales del FPGA SPARTAN 3A.

Elementos que integran la tarjeta FPGA	DESCRIPCIÓN
Chip	XC3S700A
Número de compuertas lógicas	700 000 equivalentes a 13248 celdas lógicas
Reloj(clk)	clk principal de 50MHz, clk secundario 123MHz y entrada para clk auxiliar
Memoria SRAM	360 KB
Memoria Flash	8 MB
Puertos genéricos I/O	372
Puertos de comunicación	Serial RS-232, VGA, PS/2 y Ethernet
Otras	Pantalla LCD, 4 interruptores, 8 leds, 4 botones y 1 botón tipo perilla con <i>push button</i> para uso general.

El pseudocódigo de la sección 1 describe la manera en la que se programó en el FPGA la generación de las señales de conmutación (S1, S2, S3) para los Mosfets superiores de las células de conmutación de inversor, las cuales requieren de un desfase de 120° como se describió en la sección 2.8.2. Las variables que se utilizan son: a) `clk_in` la cual establece el reloj principal del FPGA y es la base de tiempo para todo el diseño, b) `N` es un

registro que almacena el conteo máximo necesario para obtener la frecuencia deseada, c) *contador* es el registro que lleva el conteo para generar las señales y se incrementa cada que el flanco de subida del *clk_in* es activado, d) *d1* y *d2* indican los desfases de 120° y 240° respectivamente para la señal 2 y 3, e) *ciclo*, *ciclo2* y *ciclo3* son registros que indican el ciclo de trabajo ($D=Ton/Toff$) de cada señal, f) *ciclo2aux* y *ciclo3aux* son registros que almacena el resto del ciclo de trabajo en caso que los ciclos determinados para S2 y S3 no alcancen a terminar cuando N es reiniciado a 0. Esto debido al desfase de 120° y 240°, de esta manera cuando inicia el nuevo conteo S2 y S3 concluyen el ciclo de trabajo establecido, por ultimo *pwm1*, *pwm2* y *pwm3* son los registros de salida para las tres señales.

Seudocódigo Sección 1

Generación de S1

```

Siempre que (flanco de subida clk_in)
Comenzar ciclo para S1
    Si(contador <= N y contador <= ciclo)
        entonces
            pwm1= encendida;
            contador=contador+1;
            fin de condición
    Si(contador <= N y contador > ciclo)
        entonces
            pwm1=apagada;
            contador=contador+1;
        fin de condición
    Si(contador =N )
        entonces
            contador=0;
        fin de condición
Fin de ciclo para S1

```

Generación de S2

```

Siempre que (flanco de subida clk_in)
Comenzar ciclo para S2
    Si(ciclo2>N)
        comenzar
            ciclo2aux=ciclo2-N;
        fin de la condición
    Si(ciclo2<=N)
        comenzar

```

```

        ciclo2aux=0;
    fin de la condición
Si(contador <= d1 y contador <= N )
    comenzar
        pwm2<=0;
    fin de la condición
Si(contador > d1 y contador <= N y contador <= ciclo2)
    comenzar
        pwm2<=1;
    fin de la condición
Si(contador > d1 y contador <= N y contador > ciclo2 )
    comenzar
        pwm2<=0;
    fin de la condición
Si(contador <= d1 y contador <= N y contador > ciclo2 )
    comenzar
        pwm2<=0;
    fin de la condición
Si(contador <= d1 y contador <= N y contador <=ciclo2aux )
    comenzar
        pwm2<=1;
    fin de la condición
Fin de ciclo para S2

```

Generación de S3

```

Siempre que (flanco de subida clk_in)
Comenzar ciclo para S3
    Si(ciclo3>N)
        comenzar
            ciclo3aux=ciclo3-N;
        fin de la condición
    Si(ciclo3<=N)
        comenzar
            ciclo3aux=0;
        fin de la condición
    Si(contador <= d2 y contador <= N )
        comenzar
            pwm3<=0;
        fin de la condición
    Si(contador > d2 y contador <= N y contador <= ciclo3)
        comenzar
            pwm3<=1;
        fin de la condición
    Si(contador > d2 y contador <= N y contador > ciclo3 )
        comenzar
            pwm3<=0;

```

```

    fin de la condición
Si(contador <= d2 y contador <= N y contador > ciclo3 )
comenzar
    pwm3<=0;
fin de la condición
Si(contador <= d2 y contador <= N y contador <=ciclo3aux )
comenzar
    pwm3<=1;
fin de la condición
Fin de ciclo para S3

```

La programación del FPGA se realiza mediante el IDE (por sus siglas en inglés y título *Integrated Development Environment*) XILINX ISE[®] proporcionado por el fabricante, el cual permite la programación mediante diversos lenguajes de descripción de hardware , tales como: HDL(por sus siglas en inglés, *Hardware Description Language*), VHDL (derivado de las siglas en inglés, *VHSIC (Very High Speed Integrated Circuits)*) y Verilog[®] así como la programación mediante esquemas electrónicos a partir de bloques de diversos componentes electrónicos predefinidos y la posible creación de nuevos bloques. En la figura 3.5 se muestran las señales obtenidas del FPGA las cuales tiene la frecuencia definida en 30 KHz así como los desfases necesarios para la correcta conmutación de las células del inversor de tres células, mismas que son conectadas al circuito de generación de señales complementarias y tiempos de seguridad que se describe en la siguiente sección.

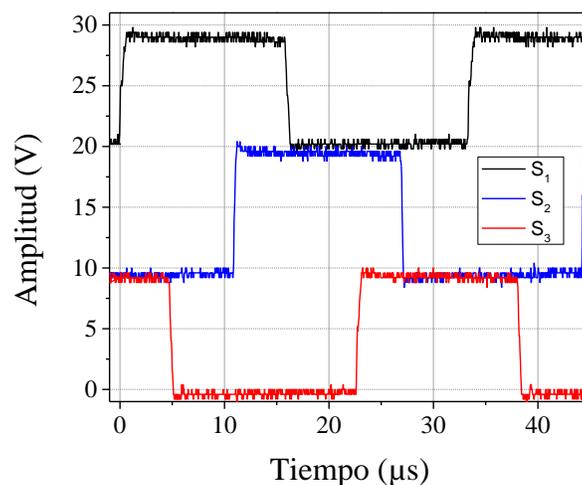


Figura 3.5. Señales desfasadas 120° generadas por el FPGA.

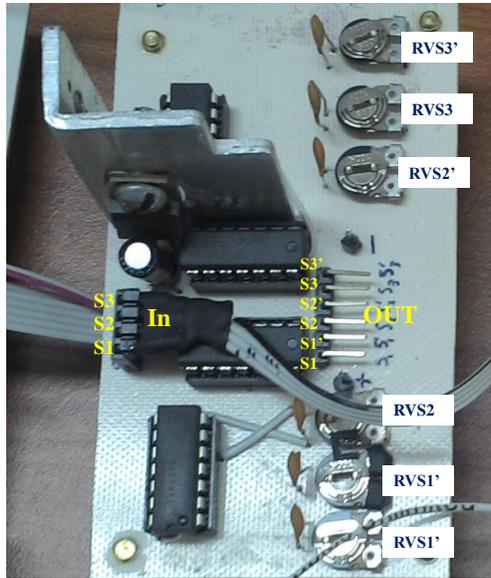


Figura 3.6b. Fotografía del circuito construido en el laboratorio para generar señales complementarias y tiempos de seguridad

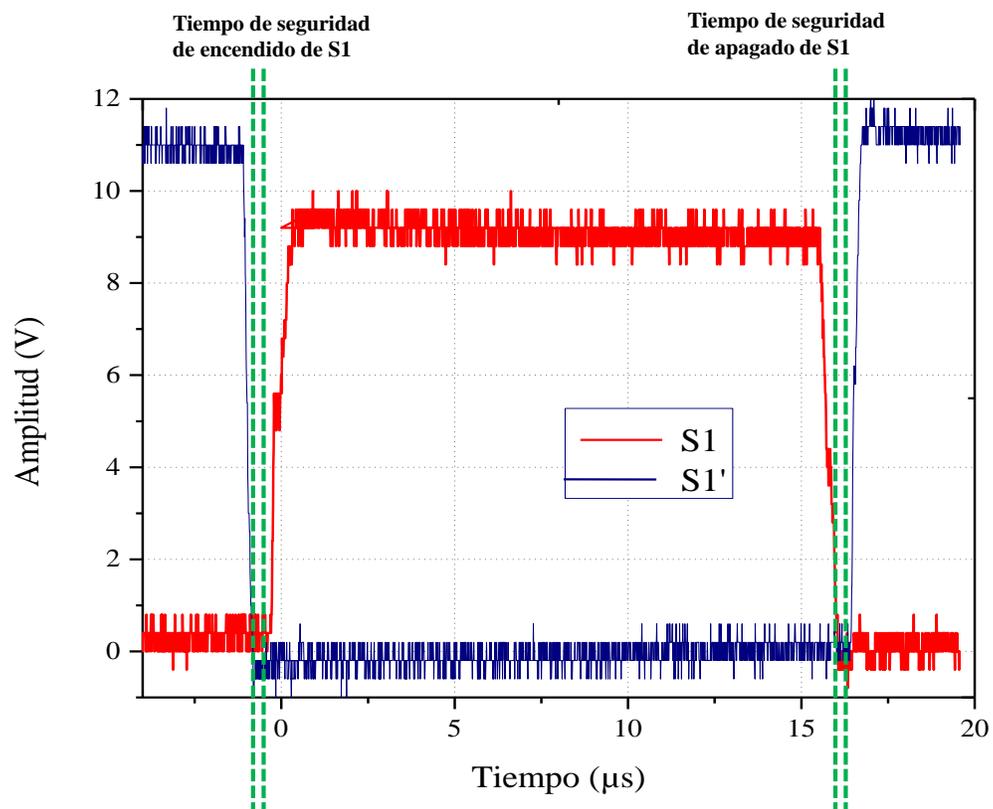


Figura 3.6c. Señales S1 y S1' con tiempos de seguridad obtenidos mediante el circuito.

3.1.5 Acoplamiento de señales de conmutación (S1, S1', S2, S2', S3, S3').

En esta etapa, se implementó un circuito para acoplar y aislar las señales de la etapa del circuito electrónico y la etapa de potencia conformada por el inversor de tres células. La parte del acoplamiento de las señales se realizó la implementación de drivers de alta frecuencia en específico mediante el componente con matrícula TC4422 cuyas características principales según el fabricante [34] son: tiempos de subida (t_R) y bajada (t_F) de 30ns , voltaje de operación de 4.5V a 18V y una corriente de salida de 9A pico, los cuales se encargan de acondicionar la señales tanto en voltaje y corriente debido a que es importante para manejar dispositivos MOSFET adecuadamente, al reducir al máximo los tiempos de retardo en la compuerta. La etapa de aislamiento galvánico está conformada por transformadores de pulso encargados de la trasferencia de la onda de tipo cuadrada desde su devanado primario hasta su devanado secundario. La figura 3.7a muestra a detalle el circuito implementado cuyo diseño electrónico fue extraído de [15] y que es descrito a continuación. Los capacitores C_{C1} y C_{C2} permiten el acoplamiento de voltaje de las señales entre el primario y el secundario del transformador, mientras que R_{GS} y D_c limitan el voltaje y la corriente respectivamente en el devanado secundario. El arreglo R_c y el diodo D permiten reducir las señales de baja frecuencia y el nivel de CD.

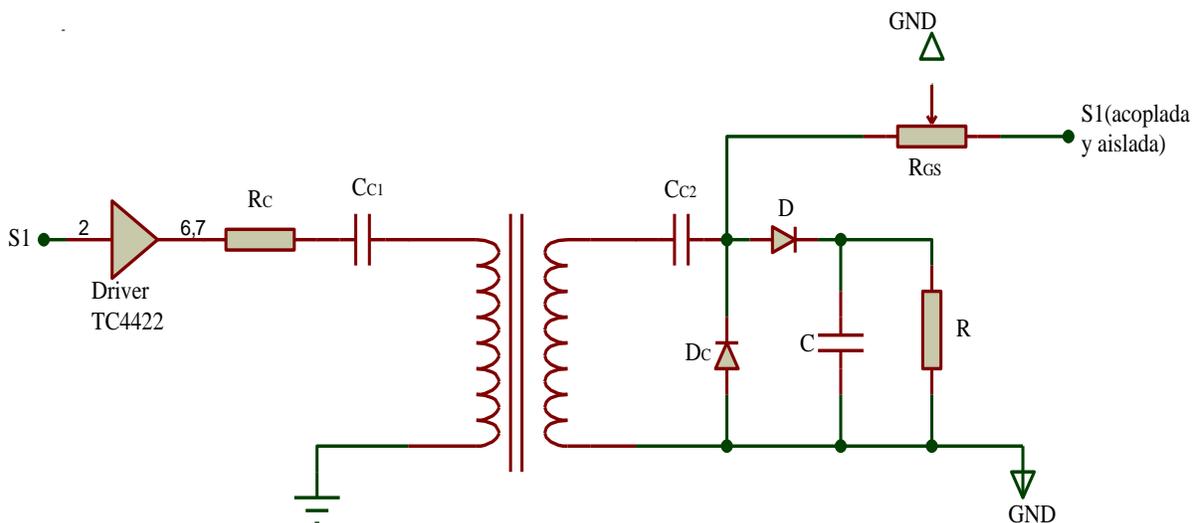


Figura 3.7a. Diagrama electrónico del circuito de acoplamiento y aislamiento.

Para la construcción de los transformadores se puede resaltar que la relación de vueltas es uno a uno (1:1), entonces las vueltas en ambos devanados son las mismas. El número de

vueltas para el transformador de pulsos está dado por la ec. (12) [15] donde N es el número de vueltas tanto para el devanado primario como para el secundario, t_D es la duración del pulso de la señal, ΔB es el flujo magnético y A_e es el área efectiva. Los últimos dos factores de la ecuación están determinados por el fabricante (Ferrotec) de la ferrita CP42-29 empleada para construcción de los transformadores [35].

Una vez construidos los transformadores de paso se realizaron una serie de pruebas para determinar la inductancia de los mismo y asegurar que los transformadores no provocaran atenuaciones en las señales obteniendo como resultado los datos de la tabla 2, es preciso detallar que debido, a que la construcción de los devanados se realizó de forma manual las inductancias no son las mismas para los 6 transformadores de paso pero se encuentran entre los límites marcados en la tabla y que debido a ello en algunas señales se percibe una atenuación de las mismas, pero no interfieren con el correcto funcionamiento de la conmutación de las células del inversor.

$$N = \frac{V_{DD} \cdot t_D}{\Delta B \cdot A_e} \quad (12)$$

Tabla 2. Mediciones de parámetros de los transformadores de paso.

Parámetros	Devanado primario	Devanado secundario
Inductancia	31.4 mH - 33.8mH	30.6 mH – 33.7 mH
Resistencia	1.4 Ω - 2.2 Ω	1.4 Ω - 2.2 Ω
Factor de calidad(Q)	267 – 280	263 -280
Frecuencia de operación	30KHz	30KHz

Por ultimo en las figura 3.7b se muestra la fotografía del circuito de acoplamiento y aislamiento construido en laboratorio, en el cual a la salida se obtienen las señales de conmutación con su referencia a GND, así como en la figura 3.7c se muestran las señales finales que realizan la conmutación de las células del convertidor una vez que han sido acopladas y aisladas.

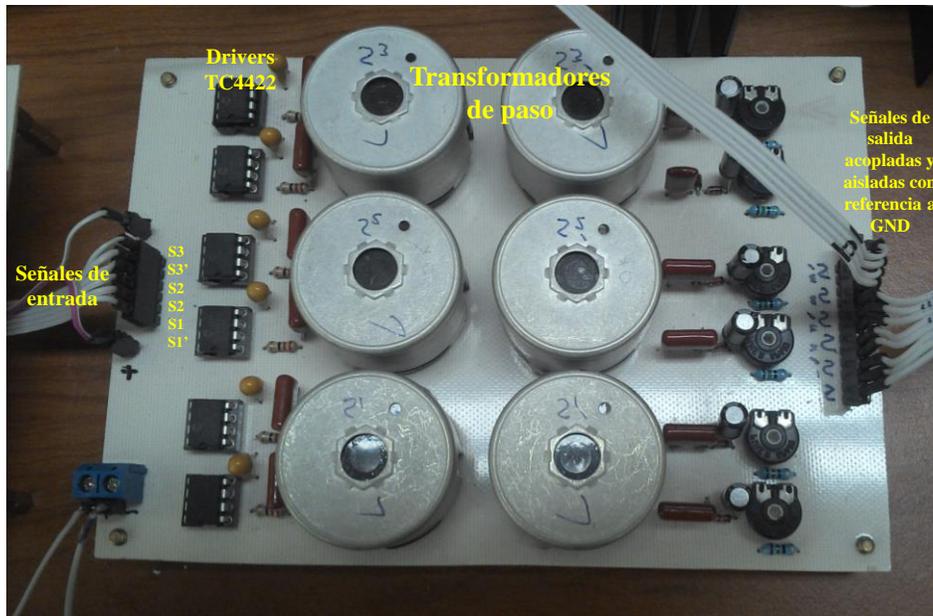


Figura 3.7b. Fotografía del circuito de acoplamiento y aislamiento construido en laboratorio.

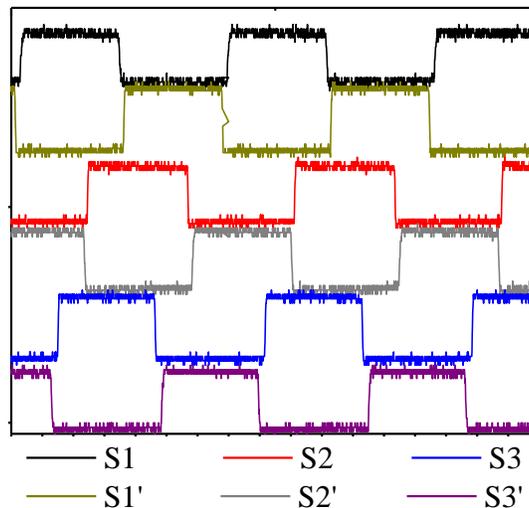


Figura 3.7c. Señales de disparo de los Mosfets del convertidor de tres células.

3.2 Operación del convertidor de tres células en lazo abierto.

La puesta a punto del convertidor estático se concluyó con la construcción del inversor de tres células y los circuitos necesarios para su funcionamiento, posteriormente se puso a prueba para corroborar su correcto funcionamiento con base en la teoría de convertidores multicelulares descrita en la sección 2.8. Así considerando que la frecuencia de conmutación establecida para cada célula es de 30KHz y un voltaje de entrada de 60V CD, se tiene a la salida del inversor se obtiene una tensión de 20V en CA, una tercera parte con

respecto a la entrada. Por otra parte la frecuencia de salida f_o es tres veces mayor a la a la frecuencia de conmutación f_{sw} , como se aprecia en la figura 3.8a. Además en la figura 3.8b se muestra la corriente a la misma frecuencia. Cabe mencionar que estas pruebas se realizan con una carga resistiva de 20Ω debido a los recursos con los que actualmente se cuentan. Por otra parte, en la figura 3.8c se muestra el equilibrio en estado estacionario del voltaje en los capacitores flotados VC1 y VC2 que conforman el convertidor de tres células.

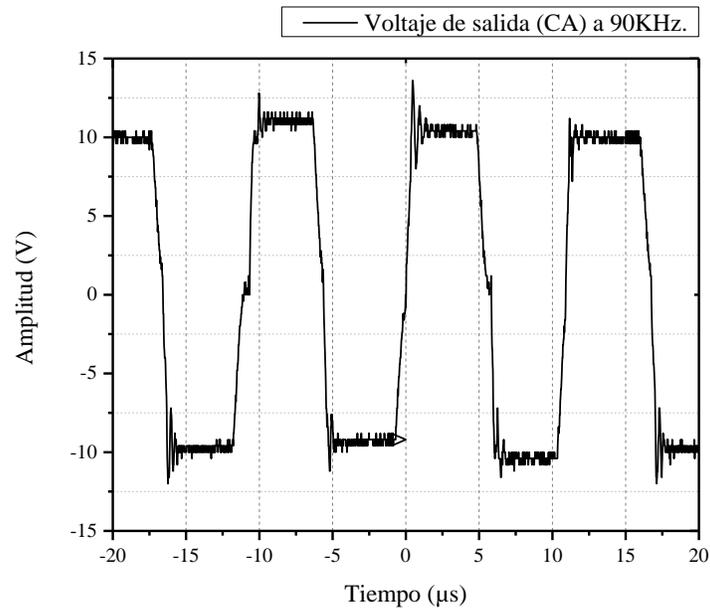


Figura 3.8a. Tensión de salida del convertidor.

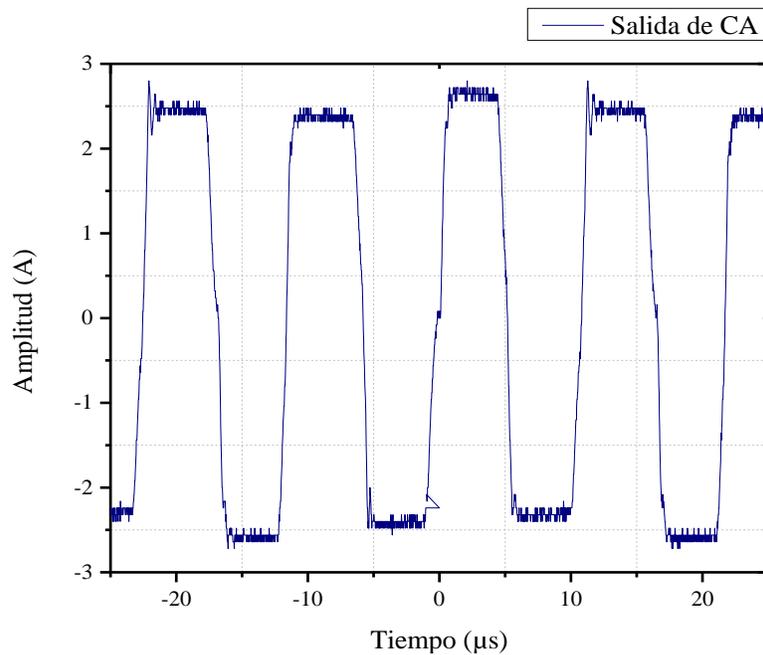


Figura 3.8b. Corriente de salida del convertidor.

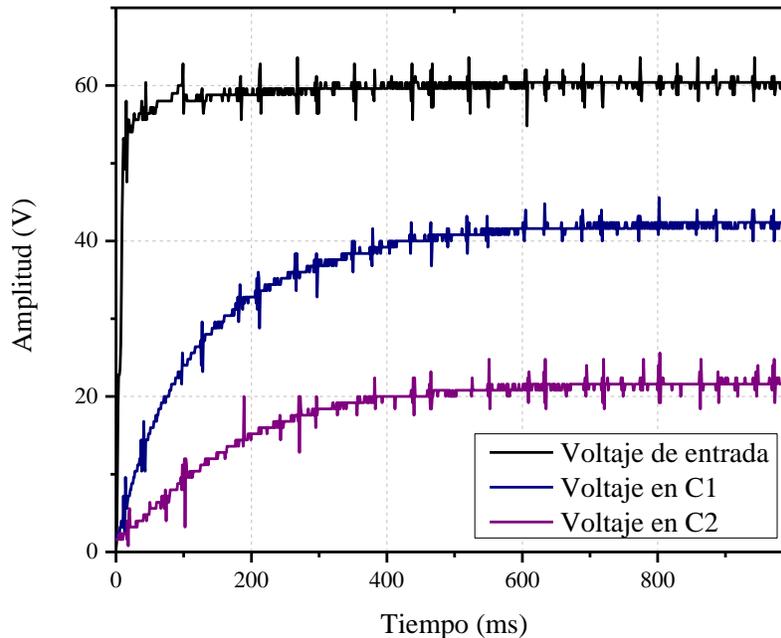


Figura 3.8c. Equilibrio de voltaje en los capacitores flotados en lazo abierto

3.3 Operación del convertidor en lazo cerrado.

Como se detalla en la sección 2.8.2, la tarea de mantener la distribución o equilibrio del voltaje en las terminales de los capacitores flotados, es fundamental para el óptimo funcionamiento del convertidor debido a que el correcto voltaje en éstos, permite que la distribución de los voltajes en cada célula sea equitativa llevando a un buen desempeño a cada dispositivo que integra el inversor. De manera general mediante la implementación de una estrategia de conmutación apropiada, es posible establecer un equilibrio natural de potenciales aunque no todo el tiempo es posible esto, debido a perturbaciones externas que pudieran afectar el funcionamiento del convertidor, de ahí la necesidad por implementar una estrategia de control que se encargue de controlar el equilibrio del voltaje de los capacitores flotados.

En la figura 3.8 se detalla el diagrama de la implementación del control en lazo cerrado del convertidor, esto mediante un control tipo PID el cual es sintonizado por un Algoritmo Híbrido (AHHL) mismo que es descrito en las secciones siguientes.

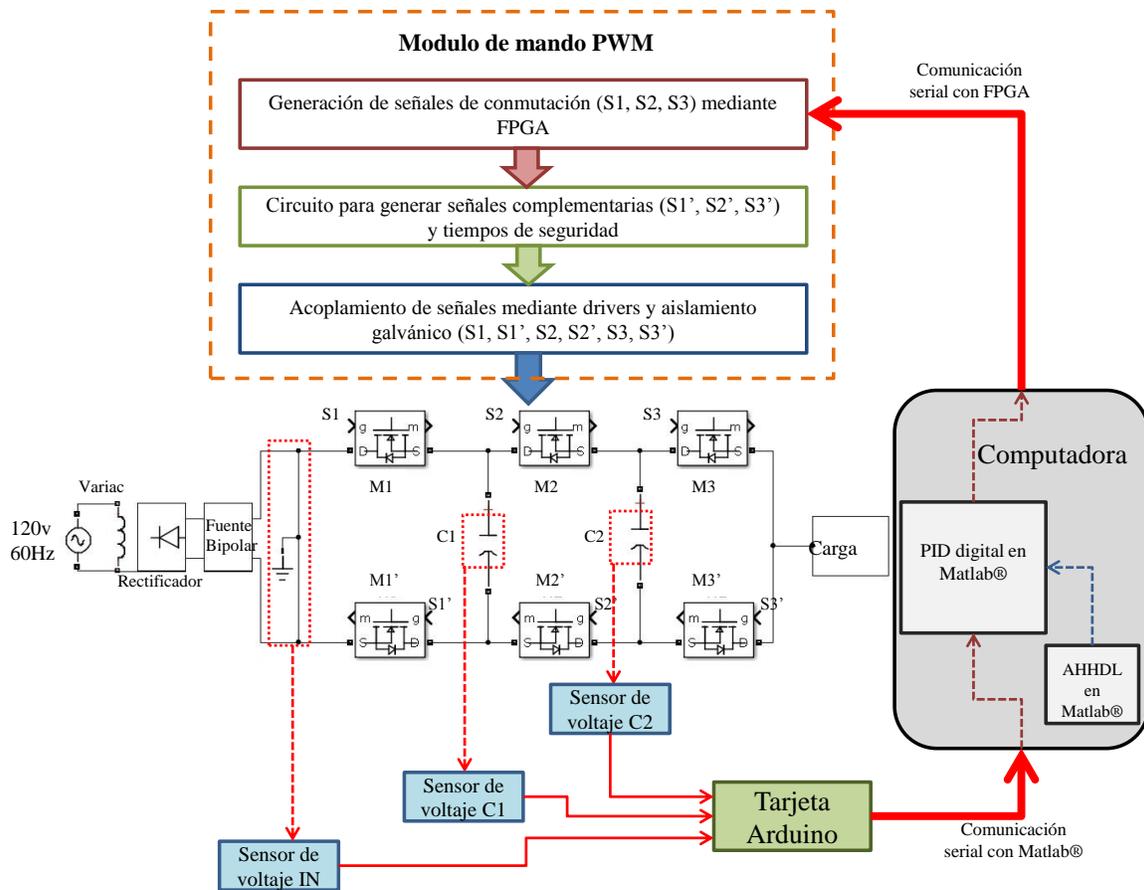


Figura 3.8. Diagrama de control en lazo cerrado del inversor de tres células.

3.3.1 Diseño y codificación del Algoritmo Híbrido.

En esta sección, se diseñó y codificó un algoritmo que permite sintonizar los parámetros del controlador PID, aplicado a los capacitores flotados del inversor de tres células, el cual es un Algoritmo Híbrido nombrado como AHHLD (por las siglas, Algoritmo Híbrido Heurística-Lógica Difusa) porque está basado en un Algoritmo Heurístico y la teoría básica de lógica difusa (*Si-entonces*). El AHHLD propuesto está descrito por el diagrama de flujo en la figura 3.9. En la primera etapa se genera sólo un individuo basado en codificación real y está integrado por tres parámetros (k_p , k_i , k_d), mismos que son generados de manera estocástica, éstos representan las ganancias Proporcional, Integral y Derivativa del controlador PID. Estos deben ser sintonizados y optimizados de tal manera que control tipo PID funcione óptimamente como se describe en la sección 2.6.10 y logre obtener el máximo rendimiento de la respuesta de los capacitores flotados del convertidor

de energía estático; en esta etapa también se inicializan las constantes de mutación (m_{kp} , m_{kd} , m_{ki}), que servirán para modificar los valores de k_p , k_i , k_d , así como los umbrales de r_t y s_t que se utilizan para permitir un margen de error al sistema.

En la segunda etapa se evalúa el individuo en la función objetivo (FO) definida por la ec. (13) y que está integrada a partir de tres características de rendimiento del sistema, como se describió en la sección 2.6.8. Las características tomadas en cuenta son: máximo sobre impulsó (Mp) definido matemáticamente por la ec. (14) donde y_{max} es el valor máximo que alcanza la respuesta, y_f el valor final de la respuesta; el tiempo de establecimiento (s_t) descrito mediante la ec. (15) como el $\pm 5\%$, donde ref es la magnitud de referencia, t es tiempo asociado al valor de la respuesta; finalmente la última característica tomada en cuenta es el tiempo de subida o de levantamiento (r_t) y se define por la ec. (16).

$$FO = \frac{1}{Mp + s_t + r_t} \quad (13)$$

$$Mp = \frac{y_{max} - y_f}{y_f} * 100 \quad (14)$$

$$s_t = y(t) \begin{cases} t \geq ref * 95\% \\ t \leq ref + (ref * 5\%) \end{cases} \quad (15)$$

$$r_t = t_1 - t_2 \quad \begin{cases} t_1 = y(t) | t = ref * 10\% \\ t_2 = y(t) | t = ref * 90\% \end{cases} \quad (16)$$

En la tercera etapa se compara el valor de FO , se establece que debe aproximarse a 0; es decir, disminuir el error en la respuesta de los capacitores flotados del convertidor. Lo anterior se logra mediante la asociación entre las características de rendimiento de sistema (Mp , s_t , r_t) y el efecto que tiene cada una de las acciones de control del controlador PID, al mutar los valores de K_p , K_i , K_d por medio de la teoría básica de lógica difusa “Si-entonces”. Por ejemplo; si FO es diferente del valor propuesto entonces mutarán K_p , K_i , K_d , incrementando o disminuyendo su valor con base a las asociaciones que se tienen respecto a las características de rendimiento como se describe en la tabla 3, donde el

símbolo (\uparrow) indica un incremento, el símbolo (\downarrow) indica una disminución y el símbolo ($-$) indica que no causa ningún efecto.

Tabla 3. Asociación de efectos de control PID sobre características de rendimiento del sistema.

Ganancia \ Característica	r_t	Mp	s_t	Error en estado estable
Kp	\downarrow	\uparrow	$-$	\downarrow
Ki	\downarrow	\uparrow	\uparrow	\downarrow
Kd	$-$	\downarrow	\downarrow	$-$

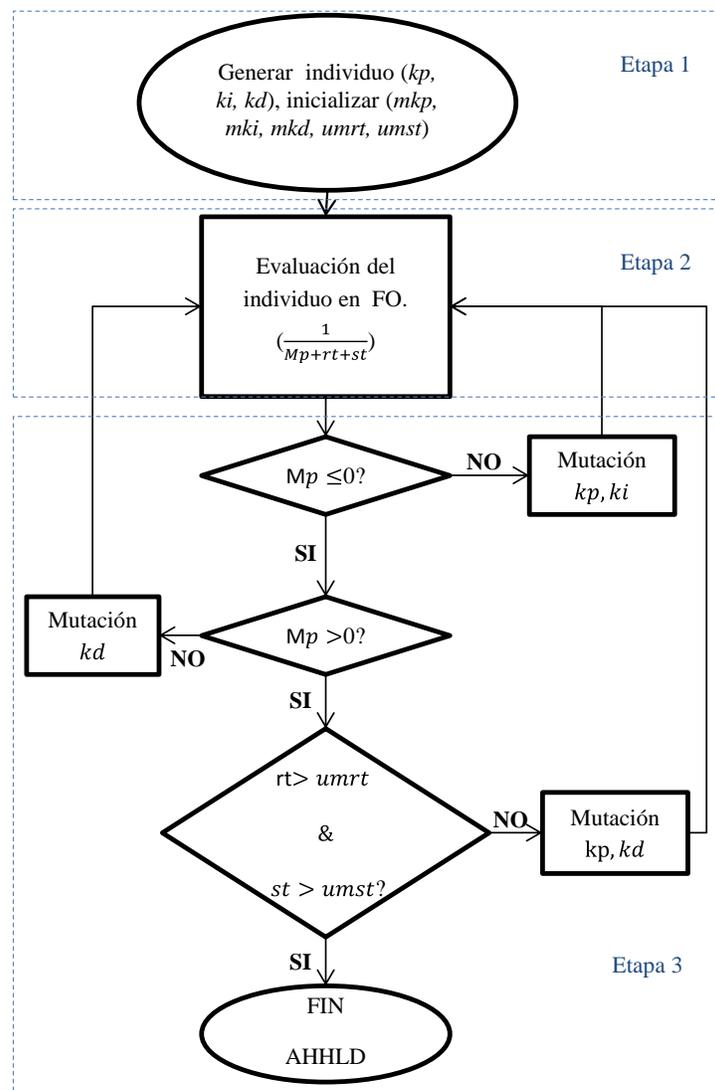


Figura 3.9. Diagrama de flujo del algoritmo propuesto.

El algoritmo se detendrá al cumplir las siguientes condiciones: a) $Mp = 0$, así se garantiza que el máximo sobre impulso se eliminará por completo del sistema, b) $rt \leq umrt$, y c) $st \leq umst$, donde $umrt$ y $umst$ son umbrales establecidos para rt y st respectivamente mediante observación, con los cuales se establece en qué orden de exactitud responderá el algoritmo y el tiempo que tarda en sintonizar los parámetros del controlador.

3.3.2 Etapa para detección de las variables de voltaje.

Las variables consideradas para desarrollar el control son el voltaje en los capacitores flotados C1 y C2, y el voltaje de alimentación de entrada. En esta sección se presenta como se implementan los sensores para detectar las variaciones de voltaje en los capacitores flotados del convertidor, que posteriormente son convertidas para su adquisición en una tarjeta digital Arduino® UNO, la cual se encarga de acondicionar las señales de voltaje y transferirlas mediante una comunicación serial al programa del controlador PID digital codificado en el software Matlab®. El sensor implementado para detectar el voltaje es un transductor modelo LEM LV 25-P cuyas características principales se detallan en la tabla 4.

Tabla 4. Características principales de operación del sensor de voltaje LV 25-P.

Característica	Valor
Taza de conversión	25:10
Corriente nominal de entrada	10 mA
Corriente nominal de salida	25 mA
Resistencia medida en la salida	100-350 Ω
Precisión	$\pm 9\%$
Voltaje de medición	10-500 V
Voltaje de operación	$\pm 12 - \pm 15$ V
Tiempo de respuesta	40 μ s

El cálculo se realiza de manera que el voltaje máximo medido corresponda a la corriente nominal de entrada del primario como se muestra en la figura 3.10a, donde se describe el cálculo para la medición de los capacitores flotados. En la figura 3.10b se muestra el transductor implementado mediante un PCB diseñado en el laboratorio. EL LV 25-P es un transductor basado en el principio de inducción electromagnética como un transformador,

por lo tanto tiene un devanado primario y uno secundario. Para la adecuación del transductor al rango de medición requerido se debe calcular la resistencia en el devanado primario (R1).

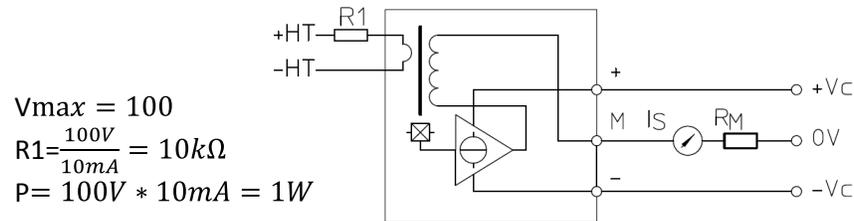


Figura 3.10a. Diagrama interno del transductor LV 25-P y cálculo de parámetros de diseño.

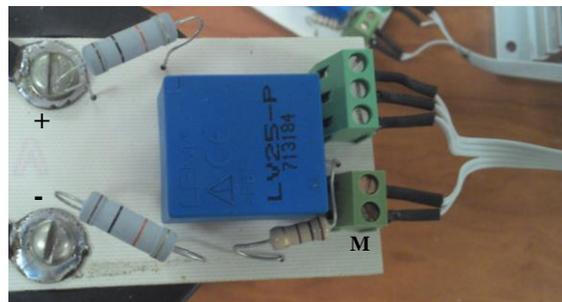


Figura 3.10b. Fotografía física del transductor de voltaje puesto en operación.

3.3.3 Controlador PID digital.

El controlador PID se programó mediante el software Matlab®, el programa se ejecuta de la siguiente manera: a) Se establecen dos puertos para comunicación serial, uno de los puertos sirve para la comunicación con el FPGA, tal y como se describió en la sección 3.1.3, este dispositivo es el encargado de genera la señales de conmutación; el segundo puerto sirve para establecer la comunicación con la tarjeta Arduino® UNO la cual es utilizada para la adquisición de los datos detectados en cada uno de capacitores flotados del convertidor, mediante los sensores descritos en la sección 3.3.2. b) Toda vez establecidos los puerto de comunicación se calculan las ganancias del controlador PID (K_p , K_i , K_d), esto mediante el AHHLD descrito en la sección 3.3.1 las cuales son útiles posteriormente para calcular las acciones de control del PID, c) se establece la referencia de entrada de voltaje del convertidor (como prueba se utilizan 60V), así como las de los capacitores flotados, a partir de las ecuaciones de equilibrio de voltaje de los mismos descritas en la sección 2.8.2,

d) el FPGA produce los ciclos de trabajo de las señales de conmutación en un 50%, e) comienza el ciclo del control PID; el cual recibe los voltajes detectados por los capacitores flotados y se resta con las referencias de voltaje establecidas, para calcular el error del voltaje en cada capacitor C1 y C2. Mediante las ganancias (K_p , K_i , K_d) calculadas previamente por el AHHL y el error obtenido, se realizan cada una de las acciones de control, Proporcional, Integral y Derivativa, las cuales se suman en una sola acción de control encargada de reducir el error en el voltaje de los capacitores flotados, por medio de la variación en los ciclos de trabajo de las señales de conmutación aplicadas de manera correspondiente a cada una de las células que conforman el inversor. Por ejemplo si el error calculado para C1 es negativo, esto significa que el voltaje en C1 sobrepasa la referencia y tendrá que reducirse para que el inversor realice su función correctamente, esto se logra incrementando el ciclo de trabajo de las señales de conmutación mediante la señal de control. Por lo contrario si el error calculado para C1 es positivo significa que el voltaje está por debajo de la referencia y tendrá que compensarse disminuyendo el ciclo de trabajo de las señales de conmutación mediante la señal de control. En el siguiente pseudocódigo describe la codificación del programa del control PID Digital diseñado.

```

//Crear variables para comunicación serial

ins=serial(Puerto,baudios) // puerto serial para recepción de datos especificando
número de puerto y velocidad de transferencia de datos

outs=serial(Puerto,baudios) // puerto serial para envío de ciclos de trabajo de las
señales PWM (S1, S2, S3) al FPGA

abrir(ins) //se abre el puerto para comenzar la recepción de datos

abrir(outs) //se abre el puerto para comenzar la transmisión de datos

teje=infinito //definimos el tiempo de ejecución del programa a infinito

t=cero //inicializamos el tiempo

kp= sintonización por el AHHL //variable proporcional del PID
ki= sintonización por el AHHL //variable integral del PID
kd= sintonización por el AHHL //variable derivativa del PID

ref=60 // establecemos el voltaje de entrada del convertidor
refc1=(ref/3)*2 // se genera la referencia para capacitor 1

```

refc2=(ref/3) // se genera la referencia para capacitor 2

ti=cero //inicializar tiempo de integración

td=cero //inicializar tiempo derivativo

enviar datos por el puerto(outs,50) //envió ciclo de trabajo de 50% a los PWM para encender el convertidor

inicia ciclo del PID

lectura de datos(ins) // se leen los datos censados de los capacitores en un vector columna

v1=lectura(1) //se extrae la lectura del voltaje de entrada del convertidor y se guarda en una variable

v2=lectura(2) //se extrae la lectura de voltaje del capacitor 1

v3=lectura(3) //se extrae la lectura de voltaje del capacitor 2

errorc1=refc1-v2 // se calcula el error del capacitor 1

ap=kp*errorc1 //se calcula la acción proporcional

ai=ki*(errorc1+ti) //se calcula acción integral

ad=kd*(errorc1-td)//se calcula la acción derivativa

señaldecontrol=ap+ai+ad//se calcula la señal de control sumando las acciones de control

//condiciones para el ajuste del ciclo de trabajo de las señales (S1, S2, S3) del FPGA con base a la señal de control del PID

si señaldecontrol < 0

enviar dato(outs,ciclo de trabajo menor al actual);

fin de la condicion

si señaldecontrol > 0

enviar dato(outs,ciclo de trabajo mayor al actual);

fin de la condicion

si señaldecontrol = 0

enviar dato(outs,ciclo de trabajo 50%);

fin de la condicion

ti=ai //guarda resultado de la integración anterior para el siguiente calculo

td=errorc1//guarda el error anterior para sumarlo al siguiente calculo

retardo de 50 ms para el siguiente calculo

fin del ciclo de PID //Finaliza el ciclo del programa

enviar datos(outs,0) //envió ciclo de trabajo de 0% a los PWM para apagar el convertidor

cerrar puertos(ins,outs) //cierra los puertos para no generar errores en la siguiente ejecución del programa

Así, toda vez que se concluyó con la etapa de programación del controlador PID, y que las demás etapas antes descritas operan de manera correcta, se integró e implementó el sistema general bajo el esquema que se presenta en la figura 3.11, con el objetivo de realizar las pruebas y establecer los resultados y conclusiones del trabajo.

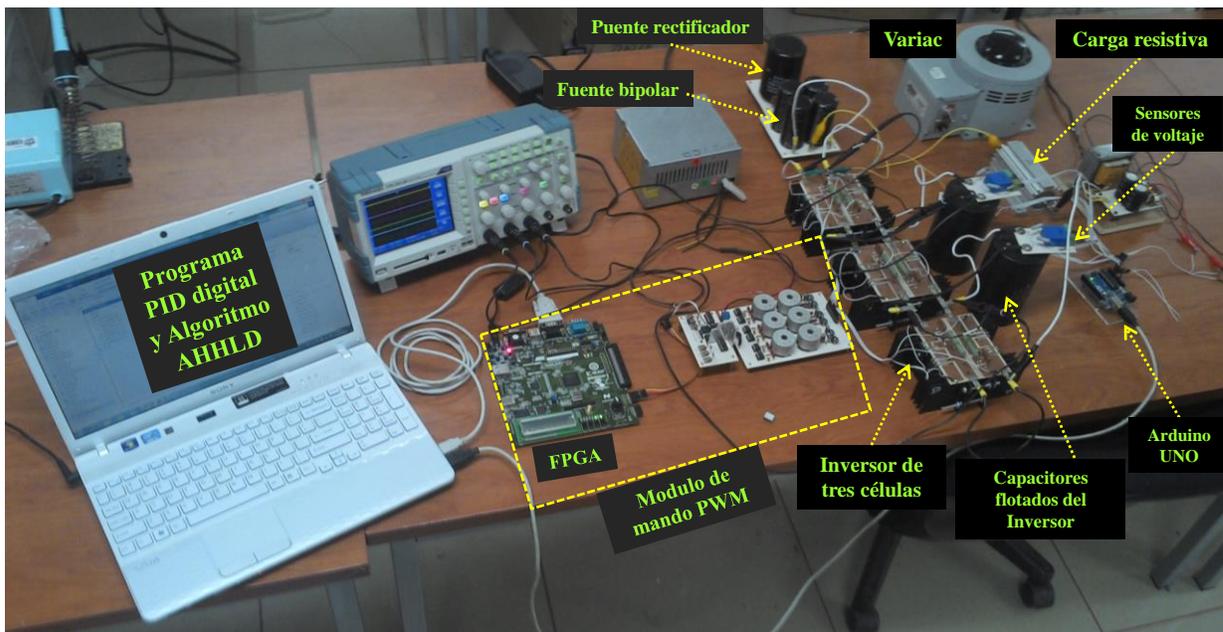


Figura 3.11 Sistema de control PID ajustado mediante AHHL aplicado a inversor de tres células

En la figura 3.11 se puede apreciar las etapas del sistema general, y descritas en los capítulos anteriores. De izquierda a derecha, se aprecia en primer lugar la etapa de control integrada por el AHHL y el control digital PID programados en Matlab®, la cual interactúa mediante la comunicación serial de la computadora con el módulo de mando PWM integrado por el FPGA *Spartan Kit 3A*, el circuito de complementos de las señales y

tiempos muertos así como el circuito de acondicionamiento, acoplamiento y aislamiento de las señales; a un costado del módulo de mando PWM se aprecia el inversor de tres células. En la parte superior de éste se aprecia la etapa de alimentación constituida por el Variac, el puente rectificador y la fuente bipolar. Por último en la parte derecha de la imagen se muestra etapa de detección de voltajes, integrada por los sensores de voltaje montados en las terminales de los capacitores flotados, los cuales son tomados por la tarjeta Arduino UNO® y enviados mediante comunicación serial hacia la computadora para completar el sistema en lazo cerrado.

CAPÍTULO 4

Resultados y Conclusiones

En este capítulo se muestran los resultados obtenidos del trabajo realizado en la presente tesis así como las conclusiones, además se plantean algunas recomendaciones y trabajo a futuro.

4.1 Ejecución del AHHL D.

Para evaluar el funcionamiento del AHHL D diseñado la sección 3.3.1 se utilizaron diferentes funciones de transferencia de algunos sistemas obtenidos de [7, 10], así como las funciones de transferencia de los capacitores flotados del convertidor generadas a partir de datos experimentales previos, obtenidos en lazo abierto. A estas últimas se aplicó un controlador tipo PID el cual fue ajustado mediante el algoritmo propuesto, las funciones utilizadas para las pruebas se detallan en la tabla 5 donde se comparan las características de rendimiento (MP , r_t , s_t) en lazo abierto, con las obtenidas mediante la implementación del controlador PID. Cabe resaltar que este proceso se realizó mediante simulaciones dentro del entorno de desarrollo del software Matlab® para validar los resultados.

En la figura 4.1 se presenta las respuestas en voltaje obtenidas por la sintonización y optimización del controlador PID mediante el AHHL D, y la respuesta en lazo abierto de las funciones de transferencia de los capacitores flotados. Por otra parte, la tabla 6 muestra los valores de los parámetros K_p , K_i y K_d obtenidos por medio del AHHL D y usados para la sintonización del PID en la simulación, se toma como referencia C1 Referencia=40V y C2 Referencia=20V como se aprecia en la figura 4.1. Con base en los valores de la tabla 5, el parámetro M_p obtenido mediante el AHHL D se optimiza (minimiza) en un 100%, para el caso de ambos capacitores (C1, C2); el parámetro s_t se reduce hasta ~538 ms para C1 y hasta ~558 ms para C2 con respecto a r_t se reduce hasta ~304 ms para el caso de C1 y hasta ~353 ms para C2, los valores anteriores muestran una comparativa entre el control PID y la respuesta de los capacitores en lazo abierto, bajo las mismas condiciones.

Tabla 5. Comparación de rendimiento de sistemas en lazo abierto y lazo cerrado con PID.

Función de transferencia	Sistemas en lazo abierto			Sistemas en lazo cerrado		
	M_p (%)	r_t (s)	s_t (s)	M_p (%)	r_t (s)	s_t (s)
$C1 = \frac{14291}{s^2 + 1911s + 13658}$	2.3	0.3062	0.5459	0	0.00262	0.0078415
$C2 = \frac{184.71}{s^2 + 33.765s + 170.34}$	1.57	0.371	0.67469	0	0.01891	0.026407
$Ft1 = \frac{1}{s^2 + 0.5s + 1}$	44.323	1.2687	14.116	0	0.000963	0.0017186
$Ft2 = \frac{1}{s + 4}$	0	0.5492	0.97802	0	0.001877	0.0033564
$Ft3 = \frac{0.5}{s^2 + 2.5s + 4}$	16.89	0.7830	3.9244	0	0.000989	0.001775
$Ft4 = \frac{0.9976}{0.00065s^2 + 0.051s + 1}$	0	0.087	0.1527	0	0.00022	0.0004416

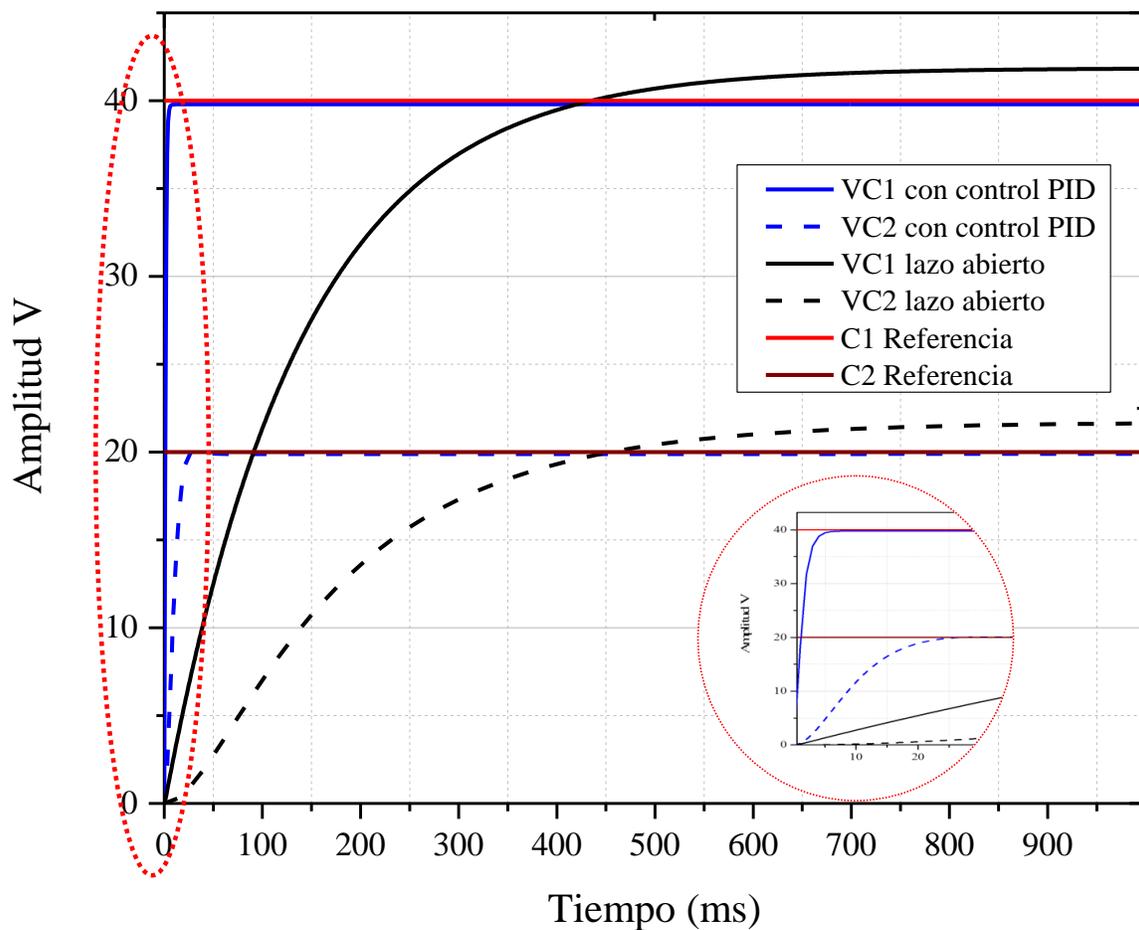


Figura 4.1. Curvas de respuestas de simulación de VC1 y VC2 en lazo abierto y en lazo cerrado.

Tabla 6. Valores de los parámetros del controlador PID obtenidos mediante el AHHL D.

Parámetros Planta	K_p	K_i	K_d
C1	181.33758	0.04816	0.11097
C2	167.84982	0.72734	1.39575

Es importante resaltar que el AHHL D puede mejorar en cada iteración los resultados mostrados, pero con el inconveniente que los parámetros para el controlador nunca converjan a un valor estable, sino todo lo contrario, que los valores se incrementen considerablemente y teóricamente tiendan al infinito, por lo tanto los valores no serán logrados por el sistema real, debido a las magnitudes que alcanzan. Por otra parte, el tiempo de ejecución del algoritmo es indefinido, ya que al ser un algoritmo de naturaleza estocástica no es posible calcular en cuantas iteraciones encontrar los parámetros necesarios para la sintonización del PID, debido a que cada iteración es única y aunque los resultados son aproximados pueden variar en las cifras decimales de manera significativa.

4.2 Control del ciclo de trabajo de las señales de conmutación.

Como se describe en la sección 2.8.2 el óptimo funcionamiento del inversor de tres células se alcanza si y solo si, control de las señales de conmutación cumple con dos características de manera rigurosa; la primera es el correcto desfase entre señales y la segunda es la manipulación del ciclo de trabajo. El primero se logra mediante los registros de desfase, descritos en el pseudocódigo del programa de generación de señales en el FPGA, definido en la sección 3.1.3. En el segundo se tiene el inconveniente de no poder manipular los ciclos de trabajo de manera individual para cada señal generada, esto debido a la sincronización en la velocidad de transferencia de datos por el puerto serial, entre la computadora y el FPGA, si bien es un problema de sincronización en los tiempos, es importante establecer que es necesario implementar otro tipo de arquitectura que integre todo el sistema, con el objetivo de sobreponer estos inconvenientes al momento de aplicar sistemas que dependen de unos con respecto al momento de realizar la interfaz. El problema se resolvió parcialmente generando rutinas que especifican los ciclos de trabajo de las

señales de conmutación (S1, S2, S3), estas rutinas son establecidas a partir de la comunicación serial con el FPGA. Es decir, desde el Programa del PID digital se envía por el puerto serial el número que representa el ciclo de trabajo deseado, de esta manera se establece la rutina de ciclos de trabajo para las señales dentro del programa del FPGA. En la figura 4.2 se muestra la respuesta de carga de voltajes de C1 y C2 funcionando a diferentes ciclo de trabajo de las señales de conmutación del inversor. Para comparar el funcionamiento de los ciclos de trabajo se obtuvieron las respuestas en lazo abierto cuando se aplica un ciclo de trabajo de 50% (color negro) y se compara con dos diferentes ciclos de trabajo. Así, si se reduce el ciclo de trabajo de las señales de conmutación (color azul) se reduce el tiempo de subida (rt), sin embargo se produce un sobre impulso y un error en estado estable; efectos que proporcionan estrés electrónico a los Mosfets del inversor. Por otra parte, si el ciclo de trabajo se incrementa (color rojo) se elimina el sobre impulso (Mp), pero el error en estado estable incrementa evitando con ello que el equilibrio de los capacitores flotados se logre. Con base en las observaciones se asocian los ciclos de trabajo a la señal de control producida por el PID digital para poder implementar el control del inversor de tres células.

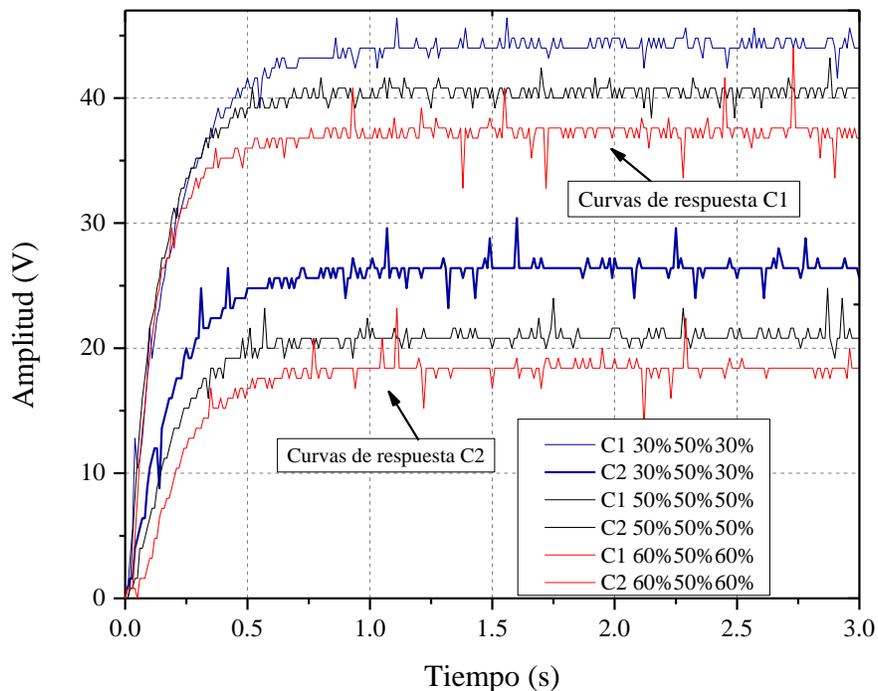


Figura 4.2. Respuestas de C1 y C2 con diferentes ciclos de trabajo de las señales de conmutación.

4.3 Resultados del control PID implementado al inversor de tres células con capacitores flotados.

En esta sección se muestran los resultados obtenidos de la implementación del control PID al inversor de tres células. En primer lugar se muestran la respuesta de los capacitores flotados, una vez que se aplicó el control PID sólo al capacitor flotado uno (VC1). Como se observa en la figura 4.3 las curvas de respuesta presentan un M_p en ambos casos, para VC1 este logra un valor aproximado del 8% mientras que para VC2 alcanza un valor aproximado de 25%, respecto a s_t se alcanza después de ~ 1.5 segundos para ambos capacitores. De igual forma se observa una caída de voltaje en la entra de alimentación del inversor de tres células (V_{in}), esto una vez que se pone en funcionamiento el inversor y comienza la carga de los capacitores flotados, generando un error en estado estable para este, mismo que no se compensa durante el tiempo de funcionamiento del inversor.

Posteriormente, se implementó el controlador PID tanto para el capacitor uno como para el capacitor dos y los resultados obtenidos se aprecia en la figura 4.4, donde se observa que el M_p que presentaban tanto VC1 como VC2 en la figura 4.3 son eliminados, asimismo s_t es aproximado a 750 ms, alrededor del 50% del valor generado en la figura 4.3.

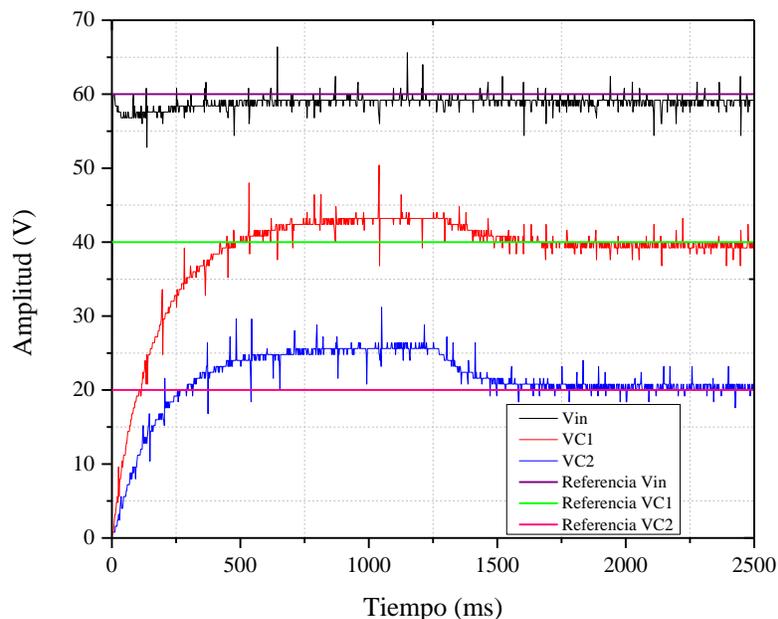


Figura 4.3. Curvas de respuesta de los capacitores flotados, aplicando control sólo en el capacitor uno.

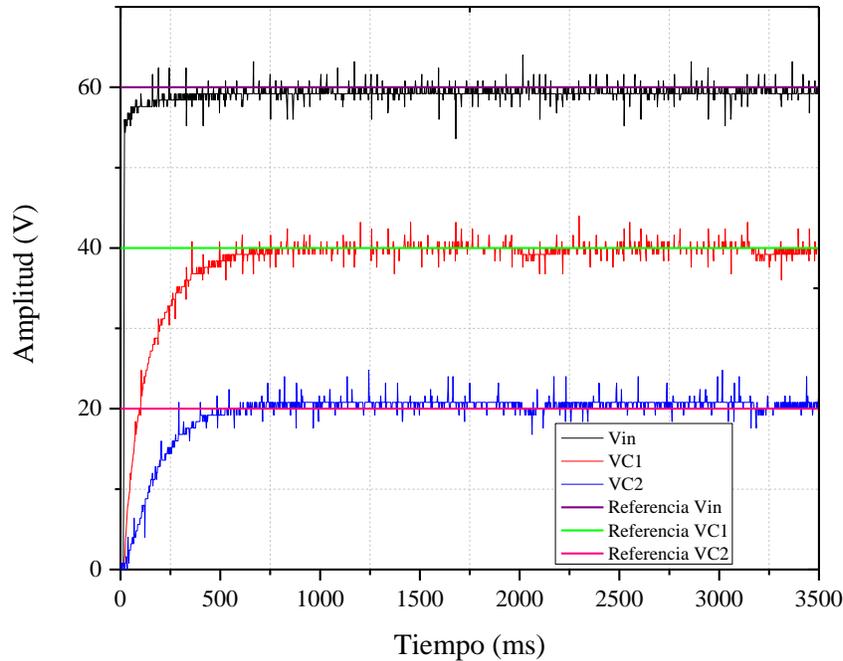


Figura 4.4. Curvas de respuesta de los capacitores flotados, aplicando control a los dos capacitadores flotados del inversor.

Cabe mencionar que aunque se mejoran las características de rendimiento en la respuesta transitoria de los capacitores flotados del inversor, en la respuesta en estado estable se generan oscilaciones notables, en comparación con la respuesta de los capacitores en lazo abierto mostrada en la figura 3.8c de la sección 3.2. Este efecto de oscilación no se logra compensar a lo largo del tiempo de funcionamiento del inversor, por lo contrario, pueden producirse oscilaciones mayores. Estas oscilaciones se pueden atribuir a los tiempos de respuesta de los sensores de voltaje, el tiempo de cálculo de la señal de control del PID, el tiempo de transferencia de datos a través de la comunicación serial, no sólo en la detección del voltaje, sino también en el tiempo de cálculo para la acción de control en las señales, así como las algunas perturbaciones que se pudieran presentar en los capacitores del inversor. Por ejemplo, si al momento de calcular la señal de control el sensor mide una perturbación, el cálculo del error será elevado, por lo que la señal de control tratara de compensar el error en ese instante mandando un cambio brusco del ciclo de trabajo de las señales, provocando con ello que se generen cálculos erróneos por parte del controlador PID y por lo mismo oscilaciones en la respuestas. Como todo sistema de control realizable tiene un límite en cuanto a las capacidades que puede otorgar, debido a que no siempre las ganancias en un

controlador son realizables, por lo tanto existen criterios y limitantes que se presentan toda vez que el sistema general se pone en marcha para su experimentación.

Otro tipo de prueba realizada, es mediante la aplicación de una perturbación generada mediante una carga resistiva en el sistema, específicamente en los capacitores flotados, no sólo en lazo abierto sino también en lazo cerrado. En la figura 4.5 se muestran las respuestas en estado estable de los capacitores flotados en lazo abierto, el procedimiento saca fuera de valor de voltaje (valor de referencia) a VC1, considerando que la estrategia de conmutación no sufre modificaciones, se espera que el valor de voltaje vuelva a su valor de referencia debido al equilibrio natural del que se ha estudiado ampliamente en este tipo de convertidores multicelulares, el tiempo que toma en tardar en corregirse es ~ 1.2 segundos, lo anterior genera un efecto en el equilibrio del capacitor dos VC2, provocando una perturbación menor, debido a que los voltajes se desequilibran o se desvían de su valor de referencia durante el mismo tiempo.

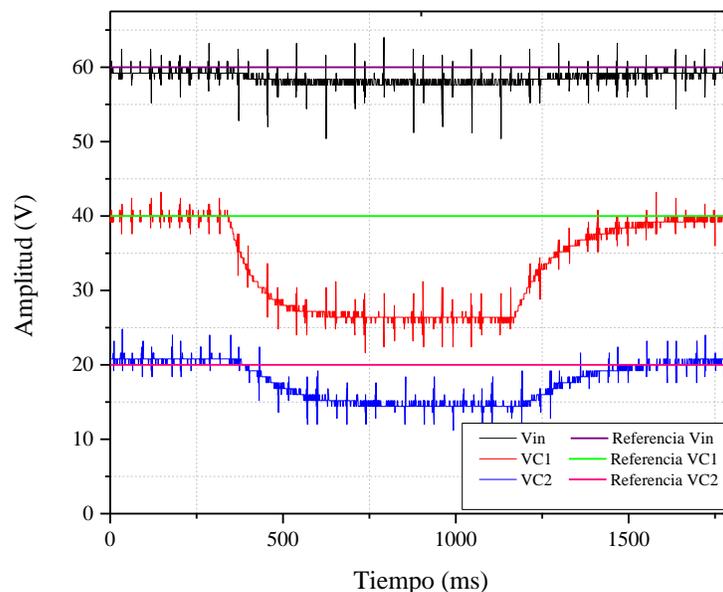


Figura 4.5. Respuesta de los capacitores flotados en lazo abierto, aplicando una perturbación al capacitor uno.

Para el caso del funcionamiento en lazo cerrado se aplicó la misma perturbación para el capacitor uno VC1, los resultados se aprecian en la figura 4.6 donde observa que el tiempo

que tarda en recuperarse de la perturbación es ~ 700 ms y la perturbación en VC2 es mucho menor que la que se presentaba en el funcionamiento en lazo abierto. Sin embargo la recuperación a la perturbación, genera un error en estado estable, lo cual puede causar oscilaciones en la respuesta de sistema, como se describió en el párrafo anterior.

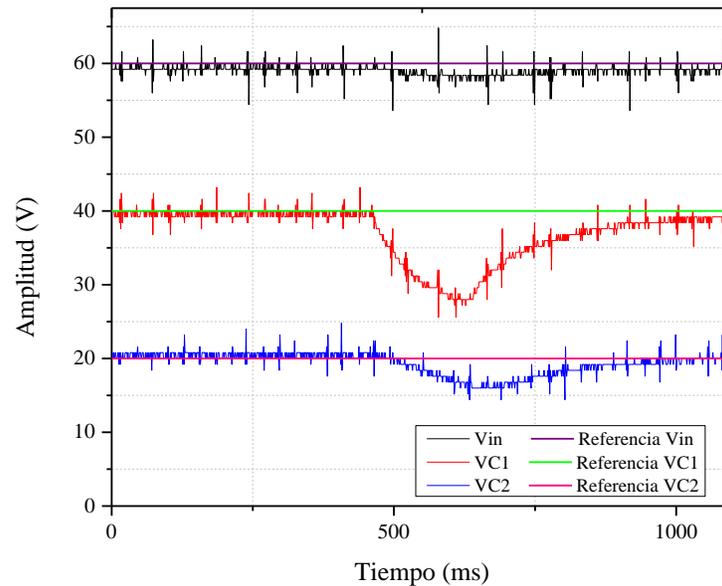


Figura 4.6. Respuesta de los capacitores flotados en lazo cerrado, aplicando una perturbación al capacitor uno.

4.4 Conclusiones.

Con base en la hipótesis planteada para este trabajo de tesis y los resultados obtenidos se establece que, el algoritmo desarrollado (AHHL) no sólo es capaz de sintonizar sino también de optimizar los parámetros del controlador tipo PID, mediante la función objetivo del algoritmo, integrada a partir de las tres características de rendimiento del sistema (M_p , st y rt). Respecto al control PID, dentro de la implementación en simulaciones, presenta resultados ideales, debido a que no se toman en cuenta muchas de las limitantes de los sistemas reales tales como: a) tiempo de muestreo de los sensores, b) tiempos de cálculos de la señales de control, c) respuestas reales de los sistemas a controlar así como las limitantes intrínsecas de los componentes del sistema, y d) tiempos de transferencia de datos entre sensores, algoritmo de control y actuadores.

La implementación del control PID al inversor de tres células es posible, sin embargo pierde eficiencia, debido a los tiempos de retraso al momento de integrar el sistema e implementarlo de manera real, siendo estas las principales limitantes. Otro factor que limita el sistema, es el control individual sobre los ciclos de trabajo de las señales de conmutación debido a que todas están sincronizadas con el mismo temporizador o señal de reloj y además se utilizan los mismos recursos de cada sistema; en el caso del FPGA y del set de Arduino. Esto ocasiona que los desfases necesarios para la conmutación de las células del inversor y la frecuencia de conmutación tengan una limitante en cuanto al uso de los recursos de cada una de las arquitecturas utilizadas.

Por otra parte, la implementación de controladores PID para el inversor de tres células es buena opción cuando se trabaja con frecuencias de conmutación relativamente bajas en el orden de las decenas y hasta las centenas de Hertz, ya que a pesar de ser un controlador robusto, éstos son típicamente aplicados para control de motores los cuales operan a frecuencias de las decenas de Hertz, y para los cuales han demostrado con creces su eficiencia, mientras que para los inversores de topología multicelular uno de los objetivos es poder genera CA a altas frecuencias en el orden de los KHz.

Con respecto al inversor de tres células se logra su correcto funcionamiento en lazo abierto mediante la generación de las señales de conmutación mediante un sistema digital basado en un FPGA. Es importante resaltar que la principal ventaja de los FPGA es generar hardware a la medida, aspecto que permite genera un módulo de mando PWM con los desfases necesarios, así como la frecuencia requerida para las señales de conmutación de las células de conmutación del inversor. Por otra parte, cabe destacar que el desarrollo e incorporación de otras técnicas de control para sistemas de electrónica de potencia tales como el inversor de tres células, son base fundamental para el desarrollo de nuevas tecnologías donde se requiera voltaje de CA a frecuencias del orden de los kilo Hertz.

Como parte complementaria se destaca la versatilidad del Software Matlab® para el desarrollo de algoritmos, no sólo en el área de control automático, sino también en la parte

de optimización para lograr una integración y comunicación mediante sistemas tales como tarjetas de desarrollo Arduino UNO y FPGA.

Finalmente, se ha llevado a cabo un sistema experimental y en prototipo de un sistema basado en cómputo inteligente, que tiene ventajas en cuanto a que no se requiere un modelado matemático que pueda ser no aplicable, como sucede en el diseño teórico, sin embargo este documento deja establecidos criterios para considerarse en un trabajo a futuro.

4.5 Trabajo a futuro.

Con base en las conclusiones obtenidas se proponen algunas ideas de mejora para el trabajo, las cuales no se aplicaron debido a la limitante del tiempo de desarrollo de este proyecto

Una de las propuestas fundamentales es mejorar el diseño del sistema digital implementado en el FPGA, el cual es el encargado de generar las señales de conmutación; la mejora consiste en realizar la modificación de los ciclos de trabajo de la señales de conmutación de manera individual, es decir en lugar de generar rutinas que modifiquen los ciclos de trabajo de S1, S2, S3 simultáneamente se tiene que implementar una estrategia que permita modificar esos ciclos de trabajo de manera individual, esto debido a la importancia de la conmutación del inversor de tres células para obtener su óptimo funcionamiento. Una solución para el problema anterior es mediante Máquinas de Estados Finitos, ya que con base en la literatura es posible realizar una estrategia con las características antes mencionadas.

Otra propuesta enfocada al módulo de mando PWM del inversor es, generar desde el FPGA las señales complementarias S1', S2', S3', así como los tiempos de seguridad de las mismas, de esta forma se omitiría la construcción del circuito electrónico-analógico descrito en la sección 3.1.4 el cual se encarga de dicho proceso.

La propuesta enfocada a la etapa de control, es codificar el control PID de manera que pueda ser implementado en el propio FPGA, de esta forma el sistema se convertiría en un sistema embebido, y cuya ventaja principal sería que, al ejecutarse todo en un solo sistema de procesamiento se eliminaran tiempos de transferencias de datos optimizando con ello tiempo de respuesta de la etapa de control.

Por último se propone, poner a prueba una nueva estrategia de control para el inversor de tres células, en específico un control difuso, debido al auge y los buenos resultados de este dentro de la literatura, esto con el fin de comparar la eficiencia del control PID con respecto a este tipo de controladores sustentados bajo el área de la IA.

Referencias

- [1] Nivio Z., (2007), “Diseño de Algoritmos con Implementaciones en Pascal y C”, International Tomson Editores Spain, Madrid España, pp. 341-366.
- [2] Ponce P., (2010). “Inteligencia artificial con aplicaciones a la ingeniería”, Alfaomega, México, pp. 50-145.
- [3] Díaz F. J., (2011) Tesis de Maestría en Ciencias en Sistemas Digitales “Planificación Óptima Discreta para Problemas NP-Complejos mediante Algoritmos Inmunológicos Artificiales” Instituto Politécnico Nacional Centro de Investigaciones y Desarrollo De Tecnología Digital, Tijuana B.C México.
- [4] Enric T., Gutiérrez R. J., (1992), “Aplicaciones de la Lógica Borrosa”, Csic-Csic Press, Madrid España, pp. 110-205.
- [5] Coello Coello C., (2004), “Introducción a la computación evolutiva”, CINVESTAV-IPN Departamento de Ingeniería Eléctrica Sección de Computación., Av. Instituto Politécnico Nacional No. 2508, Col. San Pedro Zacatenco, D.F. 07300, México Reporte Técnico, Enero 2004.
- [6] Fleming P. J., Purshouse R. C., (2001), “Genetic Algorithms In Control Systems Engineering” Department of Automatic Control and Systems Engineering, University of Sheffield, May 2001, Research Report No. 789.
- [7] Katsuhiko O., (1998), “Ingeniería de Control Moderna”, Prentice-Hall Hispanoamericana, S.A. México, pp. 10-219.
- [8] Griffiz L., Bruton J., (2008), “Online PID Controller Tuning using Genetic Algorithms” Power Electronics and Motion Control Conference, EPE-PEMC, 13th, September 1-3, Poznan, Poland.
- [9] Xiao-Gen S., Li-Qing X., Cheng-Chun H., (2009), “Optimization of PID Parameters Based on Genetic Algorithm and Interval Algorithm ” Xuzhou Institute of Technology, Xuzhou 22, China, Control and Decision Conference. CCDC '09, pp. 741-745

- [10] Katsuhiko O., (1996), “Sistemas de control en tiempo discreto”, Prentice-Hall Hispanoamericana, S.A., México, pp. 23-122.
- [11] Melício R., Mendes V., (2014), “Simulation of Power Converters for Wind Energy Systems”, International Journal of Mathematics and Computer Sciences, Vol. 25, pp. 671-680.
- [12] Molina M. G., Mercado P. E., (2008), “Estrategia de control para Maximizar la Potencia extraída de aerogeneradores de velocidad variable conectados a la red eléctrica”, ASADES Avances en energías Renovables y Medio Ambiente, Vol. 12, pp. 65-72.
- [13] Leidhold R., García G., Valla M. I., (1997), “Control para máximo rendimiento de generadores eólicos de velocidad variable, con limitación de velocidad y potencia” Mendeley Control Vol. 3, pp. 3121-3126.
- [14] Rossano Díaz I. O., (2009) Tesis de maestría en Ciencias en Ingeniería Electrónica, ”Sistema digital aplicado a un convertidor multinivel estático” Instituto Tecnológico de Toluca – Instituto Nacional de Investigaciones Nucleares, Toluca, Estado de México, México.
- [15] Flores Fuentes A. A., (2009), Tesis de Doctorado en Ciencias en Ingeniería Electrónica, “Generador de RF de alta potencia por medio de un inversor multinivel”, Instituto Tecnológico de Toluca – Instituto Nacional de Investigaciones Nucleares, Toluca, Estado de México, México.
- [16] Aytekin B., (2011), “Determination of the PID Controller Parameters by Modified Genetic Algorithm for Improved Performance”, Journal of Information Science and Engineering, Vol. 23, No. 5, pp. 1469-1480.
- [17] Xiangzhong M., Baoye S., (2007), “Fast Genetic Algorithms Used for PID Parameter Optimization”, Proceedings of the IEEE, International Conference on Automation and Logistics, 18-21 Aug, Jinan, China, pp. 2144 - 2148.

- [18] Liqing X., Chengchun H., Xiaojun X., Weiyong H., (2010), “Hybrid Genetic Algorithm and Application to PID Controllers”, Conference Publications: Control and Decision Conference (CCDC), 26-28 May, Xuzhou, China, pp. 586 - 590.
- [19] Hyung-Soo H., Jeong-Nae C., Won-Hyok L., Jin-Kwon K., (1999), “A Tuning Algorithm for The PID Controller Utilizing Fuzzy Theory”, International Joint Conference in Neural Networks IJCNN '99, 10 Jul 1999-16 Jul 1999, Washington, DC, pp. 2210 – 2215.
- [20] Mann G., Bao-Gang H., Gosine R. G., (2001), “Two-level tuning of fuzzy PID controllers” IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics, Volume: 31, Issue: 2, pp. 263 – 269.
- [21] Jian-Xin X., Deqing H., (2007), “Optimal Tuning of PID Parameters Using Iterative Learning Approach” 22nd IEEE International Symposium on Intelligent Control Part of IEEE Multi-conference on Systems and Control, Singapore, pp. 226-231.
- [22] Kan J., Li W., Liu J., (2008), “Fuzzy Immune Self-Tuning PID Controller and its Simulation” Industrial Electronics and Applications 3rd IEEE Conference, Singapore, pp. 625 – 628,
- [23] Wenge L., Deyuan L., Siyuan C., Shaoming L., Zeyu C., (2010), “Tuning Digital PID Controllers for Discrete-time System by Election Campaign Optimization Algorithm”, International Conference on Mechanic Automation and Control Engineering MACE , Wuhan China, pp. 2559 – 2562.
- [24] Qi-Yao Y., Zhang W., Xu X., Zhou K., Wang C., (2010), “Fuzzy adaptive PID control of semi-active air suspension” International Conference on Computer, Mechatronics, Control and Electronic Engineering CMCE 2010, Changchun China, 24-26 Aug.2010, Volume:2 , pp. 513 – 516.
- [25] Sinthipsomboon K., Hunsacharoonroj I., Khedari J., Pongae W., Pratumswan P. (2011) “A hybrid of fuzzy and fuzzy self-tuning PID controller for servo electro-hydraulic

system” Industrial Electronics and Applications (ICIEA), 6th IEEE Conference, Beijing, pp. 220 – 225.

[26] Baase S., Gelder A., (2002) “Algoritmos computacionales introducción al análisis y diseño”, Pearson Educación, México, pp. 31-97.

[27] <http://arquepoetica.azc.uam.mx/escritos/heuristica.html>, “Heurística”, Roberto Real León *et al.* Universidad Autónoma Metropolitana, recuperado 20/08/2015.

[28] <http://canek.uam.mx/Calculo1/Teoria/Optimizacion/FTOptimizacion.pdf> Información académica e institucional “Problemas de optimización”, Universidad Autónoma Metropolitana, recuperado 25/08/2015.

[29] Snyman J., (2005), “Practical Mathematical Optimization: An Introduction to Basic Optimization Theory and Classical and New Gradient-Based Algorithms”, Springer Science Inc., United States of America, pp. 18-26.

[30] <http://www.cs.us.es/~fsancho/?e=70> “PSO: Optimización por enjambres de partículas”, Dpto. Ciencias de la Computación e Inteligencia Artificial, Universidad de Sevilla, recuperado 25/08/15.

[31] http://www.datasheetcatalog.com/datasheets_pdf/A/P/T/6/APT6030.shtml “Hoja de datos del mosfet APT6030”, Advanced Power Technology, recuperado 30/06/14.

[32] Chung Pong P., (2008) “FPGA prototyping by VHDL examples”, John Wiley & Sons, New Jersey E.U.A., pp.10-180.

[33] <http://www.xilinx.com/support/documentation/devices/mature-s/spartan-3a.html> “Documentación FPGA SPARTAN 3A”, XILINX®, recuperado 12/06/15.

[34] http://www.datasheetcatalog.com/datasheets_pdf/T/C/4/4/TC4422.shtml “Hoja de datos del driver TC4422”, Microchip®, recuperado 25/05/14.

[35] <http://www.ferroxcube.com/FerroxcubeCorporateReception/datasheet/p4229.pdf> “Hoja de datos de ferrita CP42-29”, Ferroxcube®, recuperado 25/05/14.

Anexo A

Hoja de datos de los Mosfets empleados para la construcción de las células de conmutación del inversor multicelular de tres células.

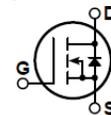
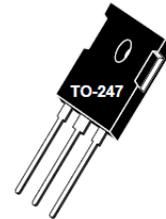


APT6030BVR

600V 21A 0.300Ω

POWER MOS V[®]

Power MOS V[®] is a new generation of high voltage N-Channel enhancement mode power MOSFETs. This new technology minimizes the JFET effect, increases packing density and reduces the on-resistance. Power MOS V[®] also achieves faster switching speeds through optimized gate layout.



- Faster Switching
- Lower Leakage
- 100% Avalanche Tested
- Popular TO-247 Package

MAXIMUM RATINGS

All Ratings: $T_C = 25^\circ\text{C}$ unless otherwise specified.

Symbol	Parameter	APT6030BVR	UNIT
V_{DSS}	Drain-Source Voltage	600	Volts
I_D	Continuous Drain Current @ $T_C = 25^\circ\text{C}$	21	Amps
I_{DM}	Pulsed Drain Current ^①	84	
V_{GS}	Gate-Source Voltage Continuous	± 30	Volts
V_{GSM}	Gate-Source Voltage Transient	± 40	
P_D	Total Power Dissipation @ $T_C = 25^\circ\text{C}$	300	Watts
	Linear Derating Factor	2.4	W/ $^\circ\text{C}$
T_J, T_{STG}	Operating and Storage Junction Temperature Range	-55 to 150	$^\circ\text{C}$
T_L	Lead Temperature: 0.063" from Case for 10 Sec.	300	
I_{AR}	Avalanche Current ^① (Repetitive and Non-Repetitive)	21	Amps
E_{AR}	Repetitive Avalanche Energy ^①	30	mJ
E_{AS}	Single Pulse Avalanche Energy ^④	1300	

STATIC ELECTRICAL CHARACTERISTICS

Symbol	Characteristic / Test Conditions	MIN	TYP	MAX	UNIT
BV_{DSS}	Drain-Source Breakdown Voltage ($V_{GS} = 0V, I_D = 250\mu\text{A}$)	600			Volts
$I_{D(on)}$	On State Drain Current ^② ($V_{DS} > I_{D(on)} \times R_{DS(on)}$ Max, $V_{GS} = 10V$)	21			Amps
$R_{DS(on)}$	Drain-Source On-State Resistance ^② ($V_{GS} = 10V, 0.5 I_{D(Cont)}$)			0.30	Ohms
I_{DSS}	Zero Gate Voltage Drain Current ($V_{DS} = V_{DSS}, V_{GS} = 0V$)			25	μA
	Zero Gate Voltage Drain Current ($V_{DS} = 0.8 V_{DSS}, V_{GS} = 0V, T_C = 125^\circ\text{C}$)			250	
I_{GSS}	Gate-Source Leakage Current ($V_{GS} = \pm 30V, V_{DS} = 0V$)			± 100	nA
$V_{GS(th)}$	Gate Threshold Voltage ($V_{DS} = V_{GS}, I_D = 1.0\text{mA}$)	2		4	Volts

CAUTION: These Devices are Sensitive to Electrostatic Discharge. Proper Handling Procedures Should Be Followed.

APT Website - <http://www.advancedpower.com>

USA
405 S.W. Columbia Street

Bend, Oregon 97702-1035

Phone: (541) 382-8028

FAX: (541) 388-0364

EUROPE

Avenue J.F. Kennedy Bât B4 Parc Cadéra Nord

F-33700 Merignac - France

Phone: (33) 5 57 92 15 15

FAX: (33) 5 56 47 97 61

050-5517 Rev A

Anexo B

Hoja de datos de los driver empleados para el acoplamiento de las señales de disparo de los Mosfets de las células de conmutación del inversor.



TC4421/TC4422

9A High-Speed MOSFET Drivers

Features

- High Peak Output Current: 9A
- Wide Input Supply Voltage Operating Range:
 - 4.5V to 18V
- High Continuous Output Current: 2A Max
- Fast Rise and Fall Times:
 - 30 ns with 4,700 pF Load
 - 180 ns with 47,000 pF Load
- Short Propagation Delays: 30 ns (typ)
- Low Supply Current:
 - With Logic '1' Input – 200 μ A (typ)
 - With Logic '0' Input – 55 μ A (typ)
- Low Output Impedance: 1.4 Ω (typ)
- Latch-Up Protected: Will Withstand 1.5A Output Reverse Current
- Input Will Withstand Negative Inputs Up To 5V
- Pin-Compatible with the TC4420/TC4429 6A MOSFET Driver
- Space-saving 8-Pin 6x5 DFN Package

Applications

- Line Drivers for Extra Heavily-Loaded Lines
- Pulse Generators
- Driving the Largest MOSFETs and IGBTs
- Local Power ON/OFF Switch
- Motor and Solenoid Driver

General Description

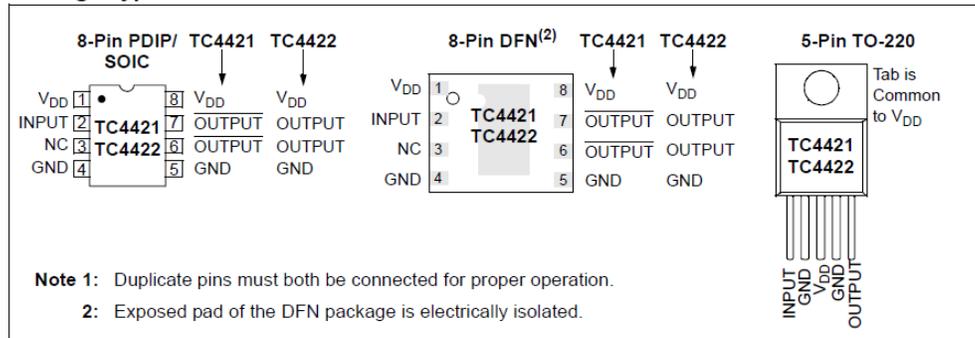
The TC4421/TC4422 are high-current buffer/drivers capable of driving large MOSFETs and IGBTs.

These devices are essentially immune to any form of upset, except direct overvoltage or over-dissipation. They cannot be latched under any conditions within their power and voltage ratings. These parts are not subject to damage or improper operation when up to 5V of ground bounce is present on their ground terminals. They can accept, without damage or logic upset, more than 1A inductive current of either polarity being forced back into their outputs. In addition, all terminals are fully protected against up to 4 kV of electrostatic discharge.

The TC4421/TC4422 inputs may be driven directly from either TTL or CMOS (3V to 18V). In addition, 300 mV of hysteresis is built into the input, providing noise immunity and allowing the device to be driven from slowly rising or falling waveforms.

With both surface-mount and pin-through-hole packages and four operating temperature range offerings, the TC4421/22 family of 9A MOSFET drivers fit into most any application where high gate/line capacitance drive is required.

Package Types⁽¹⁾



Anexo C

Ilustración de la descripción general del FPGA STARTER KID 3A que provee el fabricante de la tarjeta.

