



Universidad Autónoma de Querétaro

Facultad de Ingeniería

Diseño de arquitecturas basadas en FPGA para Algoritmos Genéticos, aplicado a la optimización de controladores en sistemas de posicionamiento

Tesis

Que como parte de los requisitos para obtener el grado de Doctor en Ingeniería

Presenta

M. en C. Arturo Yosimar Jaen Cuellar

Dirigido por:

Dr. Roque Alfredo Osornio Rios

Santiago de Querétaro, Qro, México, Octubre de 2014



Universidad Autónoma de Querétaro
Facultad de Ingeniería
Doctorado en Ingeniería

Diseño de arquitecturas basadas en FPGA para Algoritmos Genéticos, aplicado a la optimización de controladores en sistemas de posicionamiento

Tesis

Que como parte de los requisitos para obtener el Grado de Doctor en Ingeniería

Presenta:

M. en C. Arturo Yosimar Jaen Cuellar

Dirigido por:

Dr. Roque Alfredo Osornio Ríos

Dr. Roque Alfredo Osornio Ríos
Presidente


Firma

Dr. René de Jesús Romero Troncoso
Secretario


Firma

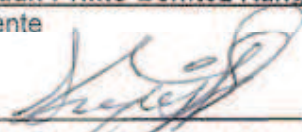
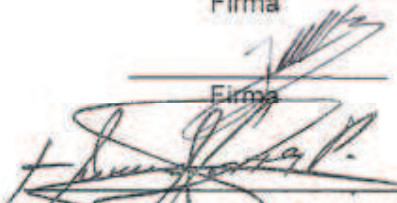
Dr. Luis Morales Velázquez
Vocal


Firma

Dr. Miguel Trejo Hernández
Suplente


Firma

Dr. Juan Primo Benítez Rangel
Suplente


Firma
Dr. Aurelio Domínguez González
Director de la Facultad
Dr. Ilíneo Torres Pacheco
Director de Investigación y Posgrado

RESUMEN

Los sistemas de control de posicionamiento han sido ampliamente estudiados desde hace varias décadas, y entre los distintos enfoques cubiertos para mejorar su desempeño se encuentran los algoritmos y estrategias de control, la identificación del modelo del sistema, el ajuste de parámetros del controlador, la generación de perfiles de movimiento, la reducción de recursos del sistema digital, etc. Entre los tópicos mencionados, la sintonización de parámetros juega un rol esencial en el diseño de sistemas de control, y existen distintas metodologías empíricas y analíticas desarrolladas para este objetivo. Sin embargo, estas técnicas no proveen los resultados adecuados, sin mencionar que son técnicas que requieren conocimiento específico y trabajan sobre modelos matemáticos aproximados. Además, pueden quitar tiempo significativo de desarrollo de un proyecto y no se pueden improvisar todos los parámetros del sistema, por lo que, en este trabajo se abordará sobre la sintonía de controladores, específicamente hablando del controlador proporcional-integral-derivativa, PID. También, se aborda en técnicas de optimización modernas, conocidas como técnicas heurísticas de optimización, que imitan el comportamiento de fenómenos naturales sin recurrir a los cálculos de matemática formal clásica. La técnica adoptada en el presente trabajo son los Algoritmos Genéticos, que son una técnica que mimetiza la dinámica de la genética natural basada en la teoría de Darwin sobre la supervivencia del más apto. Se implementa una variante de estos algoritmos conocida como Micro-Algoritmos Genéticos, MAG, que utiliza un concepto que simplifica sustancialmente los recursos computacionales requeridos para su implementación. Primeramente, se desarrolla una metodología de optimización basada en los MAG aplicada a la sintonización de parámetros de un controlador PID para sistemas de posicionamiento industriales, y se realiza su implementación en software. Posteriormente, se diseñan las arquitecturas en hardware necesarias para su implementación en un sistema real en línea. La validación de la metodología y de las estructuras digitales se realiza a través de varias pruebas en el sistema físico.

(Palabras clave: Control PID, Micro-Algoritmos Genéticos, Técnicas de optimización, Sintonización de controladores)

SUMMARY

The positioning control systems have been widely studied for several decades, and through the different approaches covered to improve its performance are the strategies and control algorithms, the system model identification, the tuning of controller parameters, the generation of motion profiles, the reduction of resources in the digital system, etc. Regarding the aforementioned topics, the tuning of controller parameters plays an essential role in control systems design, and there are several analytic and empirical methodologies developed for this purpose. However, these techniques do not provide adequate results, not to mention that they are techniques that require specific knowledge and they work over approximate mathematical models. Besides, these techniques can waste significant time in the Project development and not all the system parameters can be improved, hence, this work will tackle the controller's tuning, specifically speaking of the controller proportional-integral-derivative, PID. Also, modern optimization techniques are addressed, known as heuristic optimization techniques that imitate the behavior of natural phenomena without resorting to the calculus of classic formal mathematics. The adopted technique in this work is the Genetic Algorithm, which is a technique that mimics the dynamics of natural genetic based on Darwin's theory of survival of the fittest. A variant of these algorithms known as the Micro-Genetic Algorithm, MGA, it is implemented. This uses a concept that substantially simplifies the computational resources required for its implementation. Firstly, an optimization methodology based on MGA will be developed; this methodology will be applied to the tuning of parameters of a PID controller for industrial positioning systems, and its implementation will be carried out in software. Secondly, the necessary hardware architectures are designed for their implementation in a real system online. The validation of both, this methodology and the digital structures' is made through several tests in the physical system.

(Key words: PID control, Micro-Genetic Algorithms, Modern optimization techniques, Controllers's tuning)

DEDICATORIAS

“A mi madre y a mi padre por ayudarme a llegar tan lejos”

“A mis hermanos por todo su apoyo”

“A mi esposa por estar en todo momento a mi lado”

AGRADECIMIENTOS

Agradecimiento especial a la Universidad Autónoma de Querétaro por permitirme desarrollar como profesional durante todo este tiempo.

Al Consejo Nacional de Ciencia y Tecnología (CONACyT) por el apoyo No. 328338 otorgado a través del Programa de Becas para Estudios de Posgrado y realizar mis estudios doctorales en la Universidad Autónoma de Querétaro.

A mis asesores el Dr. Roque Alfredo Osornio Ríos, por el apoyo y la confianza que me brindo para desarrollar este proyecto, y Al Dr. René de Jesús Romero Troncoso, por el apoyo y asesoría en el desarrollo de este proyecto. Y en general a ambos por sus acertados comentarios que permitieron lograr el desarrollo de este trabajo.

Agradecimiento al Dr. Luis Morales Velázquez porque me brindo un apoyo y ayuda incondicional y me enseñó tantas cosas que me hicieron crecer en el ámbito personal y profesional.

Al Dr. Juan Primo Benítez Rangel, por el apoyo y asesoría y la oportunidad de desarrollarme en el ámbito académico de la universidad y creer en mí.

A todos mis compañeros de posgrado: Benigno, Armando, David, Martín, Carlos, Paco, Irving.

A mis alumnos de licenciatura que me enseñan continuamente que siempre se puede aprender algo nuevo.

TABLA DE CONTENIDOS

RESUMEN.....	I
SUMMARY	II
DEDICATORIAS	III
AGRADECIMIENTOS.....	IV
TABLA DE CONTENIDOS	IV
ÍNDICE DE FIGURAS.....	VII
ÍNDICE DE TABLAS	IX
1. INTRODUCCIÓN.....	1
1.1 ANTECEDENTES.....	3
1.2 OBJETIVOS.....	8
1.2.1 <i>Objetivo general</i>	8
1.2.2 <i>Objetivos particulares</i>	9
1.3 DESCRIPCIÓN DEL PROBLEMA	9
1.4 JUSTIFICACIÓN	12
1.5 PLANTEAMIENTO GENERAL DEL PROYECTO	13
2. REVISIÓN DE LA LITERATURA	15
2.1 ESQUEMAS DE CONTROL INDUSTRIAL	15
2.1.1 <i>Estructura típica</i>	16
2.1.2 <i>Estructura práctica</i>	17
2.1.3 <i>Estructura ISA</i>	19
2.2 SINTONIZACIÓN DE CONTROLADORES PID.....	19
2.2.1 <i>Métodos empíricos</i>	20
2.2.2 <i>Métodos analíticos</i>	24
2.3 ABORDANDO EL PROBLEMA DE OPTIMIZACIÓN	30
2.4 TÉCNICAS META-HEURÍSTICAS.....	31
2.5 ALGORITMOS GENÉTICOS ESTÁNDAR, AGE	32
2.5.1 <i>Estructura de los AGE</i>	34

2.5.2	<i>Representación de los AGE</i>	36
2.5.3	<i>Operadores genéticos</i>	37
2.5.4	<i>Asignación de valor de adaptabilidad</i>	39
2.5.5	<i>Métodos de selección</i>	40
2.6	MICRO-ALGORITMOS GENÉTICOS, MAG.....	41
3.	METODOLOGÍA	43
3.1	DEFINICIÓN DE LA METODOLOGÍA BASADA EN LOS MAG	45
3.2	IMPLEMENTACIÓN DE LA METODOLOGÍA EN SOFTWARE	51
3.3	DISEÑO DE ARQUITECTURAS PARA IMPLEMENTACIÓN EN HARDWARE.....	55
3.3.1	<i>Implementación mediante la estrategia en SW</i>	60
3.3.2	<i>Implementación mediante la estrategia en HS</i>	61
3.4	PLATAFORMA DE PRUEBAS, FRESADORA CNC	63
4.	EXPERIMENTACIÓN Y RESULTADOS	66
4.1	RESULTADOS DE LA IMPLEMENTACIÓN EN SOFTWARE	66
4.1.1	<i>Resultados de optimización para un modelo teórico</i>	66
4.1.2	<i>Resultados de implementación en software aplicados a servosistemas</i>	68
4.2	CONCLUSIÓN DE PLATAFORMA DE PRUEBAS.....	74
4.3	RESULTADOS DEL DESARROLLO DE ARQUITECTURAS PARA IMPLEMENTACIÓN EN HARDWARE	75
4.3.1	<i>Recursos computacionales de la implementación en hardware</i>	80
5.	CONCLUSIONES Y PROSPECTIVAS	82
5.1	CONCLUSIONES	82
5.2	PROSPECTIVAS	83
6.	REFERENCIAS	85
7.	ANEXOS	90
7.1	ARTÍCULOS PUBLICADOS EN REVISTAS JCR.....	90

ÍNDICE DE FIGURAS

Figura

1.1. DIAGRAMA A BLOQUES DE LOS MÓDULOS DE UN SISTEMA DE CONTROL DE POSICIONAMIENTO.	2
1.2. PLANTEAMIENTO GENERAL DE LA METODOLOGÍA.	14
2.1. ESQUEMA DE CONTROL PID DE LAZO SIMPLE (UNA ENTRADA Y UNA SALIDA).....	15
2.2. DIAGRAMA A BLOQUES DE LA ESTRUCTURA TÍPICA DE UN CONTROLADOR PID.	16
2.3. ESTRUCTURA PID PRÁCTICA EN LAZO CERRADO.....	18
2.4. CURVA DE RESPUESTA AL ESCALÓN.	21
2.5. RESPUESTA DE OSCILACIÓN SOSTENIDA CON P_{cr}	22
2.6. REPRESENTACIÓN DE LOS POLOS Y CEROS DE UNA PLANTA EN EL PLANO COMPLEJO.	25
2.7. SISTEMA EN LAZO CERRADO ANALIZADO POR LGR.	25
2.8. DIAGRAMA A BLOQUES DE LOS ELEMENTOS PARA ARF.	27
2.9. RELACIÓN DE LAS GANANCIAS DEL CONTROLADOR PID.	29
2.10. CONCEPTO GENERAL DE LOS AGE.	34
2.11. ESTRUCTURA GENERAL DE LOS AGE.	35
2.12. REPRESENTACIÓN DE INDIVIDUOS BINARIA PARA LOS AGE.	37
2.13. EJEMPLO GRÁFICO DE OPERACIÓN DE CRUZA EN LOS AGE.	38
2.14. EJEMPLO GRÁFICO DE OPERACIÓN DE MUTACIÓN EN LOS AGE.	39
2.15. CONCEPTO DE MICRO-POBLACIÓN DE LOS MAG.	41
3.1. DIAGRAMA GENERAL DE LA METODOLOGÍA SEGUIDA.	44
3.2. ESQUEMA GENERAL DE LOS MAG.	46
3.3. ESTRUCTURA DE INDIVIDUOS Y VARIABLES DE DISEÑO EN LOS MAG.	47
3.4. DISTRIBUCIÓN INICIAL DE LOS INDIVIDUOS PARA LOS MAG.	49
3.5. ESQUEMA PARA LA IMPLEMENTACIÓN EN SOFTWARE DE LA METODOLOGÍA BASADA EN MAG.....	51
3.6. ESQUEMA PARA LA IMPLEMENTACIÓN EN HARDWARE DE LA METODOLOGÍA BASADA EN LOS MAG.	56
3.7. SISTEMA EMBEBIDO PARA OPTIMIZACIÓN DE PARÁMETROS EN LÍNEA.....	57
3.8. ESTRATEGIA EN A) SOFTWARE Y B) HARDWARE PARA LA IMPLEMENTACIÓN DE LA METODOLOGÍA BASADA EN LOS MAG.	58
3.9. ACCIONES SEGUIDAS EN EL PROCESADOR EMBEBIDO PARA LAS ESTRATEGIAS SW Y HS.....	59
3.10. DIAGRAMA DE FLUJO PARA LA IMPLEMENTACIÓN DE LA ESTRATEGIA SW.	60
3.11. ARQUITECTURA DEL MÓDULO GENÉTICO PARA LA ESTRATEGIA HS.	62
3.12. DIAGRAMA A BLOQUES DEL GENERADOR DE NÚMEROS ALEATORIOS.....	63
3.13. INSTRUMENTACIÓN Y SISTEMA DE CONTROL DE FRESADORA CNC.....	64

4.1. RESULTADOS DE LA SIMULACIÓN PARA LOS MÉTODOS MAG-GMF, AGR, Y GMF.	67
4.2. SERVOSISTEMAS USADOS, A) TORNO CNC, B) FRESADORA CNC, Y C) ROBOT PUMA.	70
4.3. RESPUESTA DE LOS SISTEMAS A) TORNO, B) FRESADORA, Y C) PUMA, Y ACERCAMIENTO DEL ESCALÓN MÁS GRANDE PARA D) TORNO, E) FRESADORA, Y F) PUMA.	73
4.4. PLATAFORMA DE PRUEBAS PARA LA METODOLOGÍA PROPUESTA.	74
4.5. CONFIGURACIÓN PARA LA VALIDACIÓN DE LAS PRUEBAS EN LA PLATAFORMA DESARROLLADA.	76
4.6. EVOLUCIÓN DE LAS GANANCIAS PARA ESTRATEGIA A) SW, Y B) HS.	78
4.7. DINÁMICA DE MOVIMIENTO, A) Y B), Y ERROR DE SEGUIMIENTO, C) Y D), PARA LAS ESTRATEGIAS SW Y HS RESPECTIVAMENTE.	79

ÍNDICE DE TABLAS

Tabla

2.1. REGLAS DE ZN PARA SINTONIZACIÓN POR RESPUESTA AL ESCALÓN. -----	22
2.2. REGLAS DE ZN PARA SINTONIZACIÓN POR GANANCIA CRÍTICA. -----	23
3.1. DEFINICIÓN DE PARÁMETROS DEL MÉTODO PROPUESTO BASADO EN MAG. -----	50
3.2. EVENTOS GENÉTICOS PROGRAMADOS PARA LA ESTRATEGIA SW. -----	61
4.1. PARÁMETROS OPTIMIZADOS Y GANANCIAS DEL CONTROLADOR OBTENIDAS -----	71
4.2. DESEMPEÑO EN LAS PRUEBAS DE MOVIMIENTO PARA CADA SERVOSISTEMA-----	72
4.3. RANGO DE DISEÑO PARA LAS GANANCIAS DEL CONTROLADOR.-----	76
4.4. TIEMPO DE CÓMPUTO REQUERIDO POR LAS ESTRATEGIAS IMPLEMENTADAS.-----	77
4.5. RECURSOS REQUERIDOS POR EL SISTEMA EMBEBIDO. -----	80
4.6. RECURSOS REQUERIDOS POR EL CO-PROCESADOR GENÉTICO. -----	81

1. INTRODUCCIÓN

Los sistemas de control de posicionamiento, los cuales realizan el control de un sistema con capacidad de movimiento, en forma rápida, suave y precisa, juegan un rol primordial en la industria de la manufactura, donde los principales objetivos son mejorar la calidad, la reducción continua de costos y acortar los tiempos de desarrollo del producto y del proceso (Figuera, 2006). La implementación de éstos sistemas es cada vez mayor, la industria se ve forzada a aumentar su nivel de calidad y competitividad a través de la modernización de su planta productiva, ya que los productos que ésta ofrece deben cumplir con las exigencias del mercado (Osornio, 2007).

El campo de investigación en los sistemas de control de posicionamiento es extenso y ha sido fuertemente estudiado durante varias décadas. En un principio las metas en las investigaciones relacionadas con los sistemas de posicionamiento eran el diseño, desarrollo e implementación de sistemas de control con ciertas características tales como; eficiencia, flexibilidad, modularidad, arquitectura abierta, costo adecuado y en general robusto en funcionalidad. La problemática consistía en aspectos como seguimiento de referencias, estabilidad del sistema, reducir el tiempo de respuesta, aspectos de monitoreo, implementación de leyes de control, entre otros. Los avances tecnológicos ayudaron a superar varias de esas dificultades con la implementación de sistemas de control digital. Sin embargo, para ese momento la problemática se concentró en implementar algoritmos y arquitecturas con el mínimo de recursos, computacionalmente eficientes, presentar soluciones SoC (System on a Chip; sistema en un chip), sistemas reconfigurables, plataformas con sistemas embebidos, etc. Hoy día, existen numerosos trabajos que solucionan la

problemática inicial. Una nueva perspectiva de investigación se dio con la necesidad de incrementar la eficiencia y desempeño de los sistemas de control existentes. En consecuencia, los esfuerzos en las investigaciones de sistemas de control de posicionamiento se han enfocado en el estudio de técnicas de optimización en dos vertientes: metodologías y desarrollo de arquitecturas.

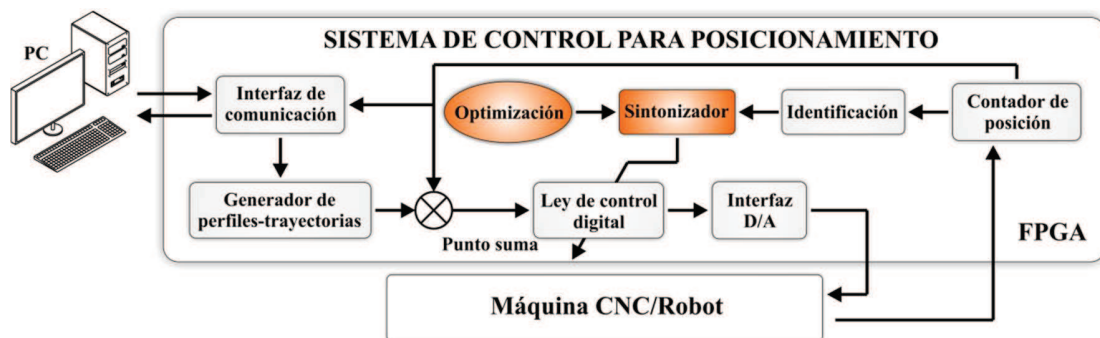


Figura 1.1. Diagrama a bloques de los módulos de un sistema de control de posicionamiento.

El diagrama de la Figura 1.1 muestra los módulos típicos de un sistema de control para sistemas de posicionamiento, como máquinas CNC y robots manipuladores, implementados en un controlador digital. Los módulos considerados en el presente trabajo de investigación son el de optimización y el de sintonización del controlador digital. El módulo de sintonización por si solo se encarga de ajustar los parámetros de la ley de control, en este caso un controlador PID, para que el sistema de posicionamiento tenga una buena dinámica de movimiento en su operación normal. Sin embargo, las metodologías de sintonización existentes no proveen de resultados adecuados ya que dependen de diversos factores como el conocimiento del usuario que realiza la sintonización, si esta se efectúa de forma manual, o puede depender de la obtención de un modelo matemático del sistema que generalmente se obtiene de forma aproximada. Por lo tanto, el módulo de optimización se combinará con el de sintonización a través de

una metodología que se describirá en capítulos siguientes para optimizar el ajuste de los parámetros del controlador.

1.1 Antecedentes

Se han presentado diversos trabajos que han tenido repercusión a nivel mundial al haber sido publicados en revistas indizadas abordando temáticas sobre alguno de los módulos que integran un sistema de control de posicionamiento. Ejemplo de ello son los trabajos por parte de los miembros del grupo de investigación HSPdigital de la Universidad Autónoma de Querétaro, UAQ, con el cual se desarrolló esta investigación. Los trabajos del grupo se han centrado principalmente en el desarrollo de metodologías y arquitecturas en hardware en cuatro temas principalmente: algoritmos de control, generación de perfiles-trayectorias de movimiento, monitoreo de dinámica e identificación del sistema.

En lo relacionado a leyes de control usando el concepto de cómputo reconfigurable, Osornio et al. (2008) presentaron el desarrollo y la implementación de un controlador PID basado en tecnología FPGA (Field Programmable Gate Array; Arreglo de Compuertas Programables en Campo) aplicado a una máquina CNC de alta velocidad, demostrando la eficiencia de utilizar éstos dispositivos para cubrir los requerimientos en tiempo de muestreo y configurabilidad. Posteriormente, Osornio et al. (2008) presentaron el diseño de una plataforma basada en FPGA implementando IP-cores en el controlador para máquinas CNC. Consiste en un rack de conexiones que integra varias tarjetas para procesamiento de datos en tiempo real; una solución SoC no es posible debido a los recursos que cada tarjeta consume. Más tarde, Morales et al. (2010) presentaron una arquitectura abierta basada en FPGA bajo una filosofía de control distribuido multi-agente, aplicada a sistemas de control en máquinas CNC. Ellos implementaron

una técnica de co-diseño hardware-software en la cual se implementaron módulos *IP-core* con las funciones que el control requería, dependiendo del usuario definirlos de forma eficiente. En lo que refiere a la generación de perfiles, Osornio et al. (2007) presentaron una técnica para la generación de perfiles polinomiales de aceleración-desaceleración en tiempo discreto, basado en un análisis paramétrico computacionalmente eficiente en una aplicación basada en FPGA. Sin embargo, los recursos utilizados para la reconstrucción del perfil crecen linealmente al grado del polinomio. Más tarde, Osornio et al. (2009) presentaron la implementación en FPGA de un generador de perfiles polinomiales de grado mayor para operación en línea, computacionalmente eficiente y de bajo costo, para control en máquinas CNC y robots industriales. El sistema es susceptible a un incremento del error debido a la naturaleza de la integración digital, por lo que la resolución en bits de los datos es grande. Posteriormente, Rivera et al. (2010) propusieron una metodología de diseño de perfiles polinomiales a trozos de grado mayor, restringidos dinámicamente y que minimizan el error de seguimiento. La descripción del perfil se realizó en base a sus parámetros dinámicos. Sin embargo, si se requerían utilizar otros parámetros era necesario repetir la etapa de cálculo a fin de incluir nuevas restricciones. Acerca de la dinámica en máquinas CNC, de Santiago et al. (2008) presentaron un método basado en FPGA que reconstruía la dinámica a partir de información proporcionada por un encoder incremental óptico acoplado a la flecha del motor, combinando la técnica de diferencias finitas y un filtro basado en la transformada wavelet discreta. Sin embargo, la reconstrucción se llevó a cabo fuera de línea. Por su parte, Rangel et al. (2010) desarrollaron un instrumento para análisis de vibración basado en la fusión de la técnica de transformada de Fourier rápida y la transformada wavelet discreta, éste fue implementado en un dispositivo de bajo costo FPGA para monitoreo en maquinaria CNC. Por su parte, Morales et al. (2010) presentaron una arquitectura en hardware de un procesador de aplicación específico para la identificación de parámetros en servo-sistemas, que implementa el método de mínimos cuadrados recursivo. La arquitectura fue diseñada para mejorar la velocidad de

procesamiento del algoritmo en un dispositivo FPGA de bajo costo, permitiendo una implementación en línea. A pesar de que los trabajos citados han aportado en el ámbito de investigación en uno o más de los aspectos que integran un sistema de control de posicionamiento, mediante algoritmos y metodologías novedosas con el desarrollo de nuevas arquitecturas, el problema de la optimización en la sintonización del controlador aún no ha sido abordado.

A nivel internacional se han presentado diversas investigaciones que desarrollan metodologías de optimización aplicadas a la problemática presente en sistemas de control. En relación con leyes de control existen trabajos que presentan implementaciones de controladores digitales basados en dispositivos de gran capacidad de procesamiento como el FPGA (Yau et al., 2006), o en un DSP (Wang et al., 2010). Por otro lado, se ha presentado el desarrollo de controladores adaptivos basados en conceptos de redes neuronales (Kamalasadan y Ghandakly, 2007). En lo referente a la optimización en la generación de trayectorias y perfiles de referencia restringidos dinámicamente en velocidad, aceleración, y jerk, se presentan trabajos como los de Erkorkmaz y Heng (2008) y Mattmüller y Gisler (2009). Otros trabajos aportan en un nuevo método para la generación de trayectorias en el maquinado CNC de alta velocidad pero requieren pre-procesamiento por lo que implementaciones en línea no son posibles (Held y Spielberger, 2009). Desde otro punto de vista en esta área, se han presentado esquemas basados en mallas que describen la superficie de maquinado para la generación de trayectorias tal como el de He et al. (2009) y Kim (2010). Sin embargo, la mayoría de las propuestas mencionadas con anterioridad se validan a través de simulaciones omitiendo quizá efectos que se presentan en situaciones reales. También en robótica industrial se han presentado trabajos donde la generación de trayectorias con disminución del jerk son el objetivo de importancia (Gasparetto y Zanotto, 2010). En relación al monitoreo del estado de la máquina, sin necesidad de sensores especializados, se han presentado trabajos que utilizan

el análisis de vibraciones en maquinaria CNC (Zhang y Chen, 2008) o a través del análisis de corriente (Xiaoli, 2005). Por otro lado, en la dinámica de movimiento de máquinas CNC se tiene el trabajo de Pessoles et al. (2010), que proporciona modelos matemáticos para mejorar las simulaciones de maquinado en CAM. Otros trabajos presentan la identificación de parámetros sin recurrir a sensores especializados mediante análisis de corriente o voltaje (Inoue et al., 2009). Finalmente, en cuanto al tema de sintonización de controladores se han presentado trabajos que desarrollaron métodos de autoajuste para los parámetros de controladores PID (Kiyong et al., 2010) o controladores autoajustables en línea basados en lógica difusa (Masiala et al., 2008). No obstante, estas metodologías de sintonización propuestas se aplican sobre modelos matemáticos conocidos por lo que el resultado es un tanto idealizado.

Una tendencia relativamente nueva en las investigaciones sobre sistemas de control es en el manejo de técnicas de optimización modernas mejor conocidas como *técnicas meta-heurísticas*, estas técnicas imitan el comportamiento de fenómenos naturales y usualmente se usan para la búsqueda de parámetros. Ejemplo de estas técnicas, y que será adoptado en la presente investigación, son los Algoritmos Genéticos, AG, que mimetizan el comportamiento de la genética natural. Algunas investigaciones relacionadas con los AG presentan aplicación en la selección óptima de la secuencia de herramientas en el maquinado de plantillas (Ahmad et al. 2010), su método se describió bajo lenguaje C++ y se validó por simulación. En otro ejemplo, se aplica los AG modificados en la planeación de procesos de manufactura de partes múltiples, la efectividad del algoritmo se demostró por simulación (Musharavati y Hamouda, 2011). Pero la aplicación de los AG también abarca cuestiones en el desempeño de controladores, tal es el caso de Kim (2011) que combinó los AG convencionales con un algoritmo basado en el fenómeno de dispersión de bacterias para la selección de los parámetros del controlador, y la eficiencia de la propuesta se validó a través de simulaciones. En

relación a este campo, Huang et al. (2007) presentaron un esquema de PID autoajustable mediante los AG para un proceso de soldadura por arco con electrodo revestido. El método de sintonización se probó en Matlab/Simulink fuera de línea. Posteriormente, Van der Lee et al. (2008) desarrollaron un algoritmo de sintonización en controladores de modelo predictivo, combinando una estructura simple de AG y un algoritmo multi-objetivo para la toma de decisiones por lógica difusa. La eficiencia del algoritmo se valida a través de simulaciones utilizando Matlab para un proceso de control de nivel y temperatura en un tanque de agua. Por otro lado, Valarmathi et al. (2009) presentaron un algoritmo para la identificación de los cambios en los parámetros del sistema y selección óptima de los parámetros del controlador en procesos químicos no lineales del nivel de pH. El algoritmo propuesto se valida por simulaciones en computadora usando Matlab bajo el ambiente de Simulink. Mientras tanto, Zhang et al. (2009) propusieron un método para controladores PID auto-ajustables basado en AG, en su propuesta presentan un operador de selección dominante y un operador de mutación cíclica. La implementación del método se realizó en software y se validó a través de simulaciones. Más tarde, Sharkawy (2010) desarrolló un esquema de control PID auto-ajustable para aplicación en el sistema de frenado antibloqueo (ABS), mediante una combinación de tres sistemas difusos Takagi-Sugeno de primer orden para la sintonía de cada parámetro del PID. Los parámetros de los sistemas difusos se determinaron usando un AG modificado fuera de línea. La robustez del esquema se examinó por medio de pruebas numéricas y simulaciones. En otro caso, Coelho y Pessôa (2011) presentaron una técnica de sintonización para controladores PI y PID en control multivariable, mediante la combinación del método de evolución diferencial con la aplicación de secuencias basadas en un mapa caótico de Zaslavskii. La técnica se simula para una columna de destilación basada en el modelo binario (lineal) Wood-Berry utilizando Matlab bajo el ambiente de Simulink. Pan et al. (2011) implementaron un controlador PID difuso para plantas de orden mayor y con tiempos de retardo. Los parámetros del PID difuso son sintonizados por medio de los AG y dos variantes del método de

optimización por enjambre de partículas, aplicados fuera de línea. La validación se realizó por simulación en Matlab. De acuerdo al análisis de los trabajos anteriores se concluye que el desempeño de un sistema de control no depende solamente de la ley, esquema o estructura de control seleccionada, también depende de una adecuada sintonización del controlador. La selección óptima de los parámetros de la ley de control es importante y se hace más evidente cuando las condiciones del proceso cambian continuamente, por lo que es necesario reajustar tales parámetros. La mayoría de los métodos de sintonización reportados han sido validados por simulación, o en algunos otros casos aplicados fuera de línea, y hasta el momento no se ha reportado una arquitectura en hardware que implemente los AG para optimizar la sintonización en sistemas de control para posicionamiento industriales aplicada en un proceso en línea. Adicionalmente, el problema en la implementación de los métodos basados en AG es la carga computacional que implican, debido a la recursividad del método y a la cantidad de datos que se manejan. En la mayoría de los casos la validación se limita a simulaciones por computadora.

1.2 Objetivos

1.2.1 Objetivo general

Generar una metodología de optimización aplicado al módulo de sintonización de control y diseñar una arquitectura que la implemente, para mejorar el desempeño de una ley de control de tres tiempos en un sistema de posicionamiento, mediante la combinación de AG y el diseño de arquitecturas basadas en FPGA.

1.2.2 Objetivos particulares

- Desarrollar la metodología de optimización basada en AG que se aplicará al módulo sintonizador mediante su implementación en software para probar cuando menos dos leyes de control de tres tiempos enfocadas al sistema de posicionamiento.

- Diseñar arquitecturas basadas en FPGA para implementar la metodología de optimización basada en AG propuesta.

- Comparar en cuanto a tiempo y recursos de cómputo la metodología desarrollada en software y la arquitectura desarrollada en hardware para evaluar la mejora del sistema propuesto.

- Probar la eficiencia y funcionalidad del sintonizador optimizado y la ley de control sobre sistemas de posicionamiento (Maquinaria CNC y Robots) para demostrar el desempeño de la metodología de optimización y las arquitecturas diseñadas.

1.3 Descripción del problema

Debido a la cantidad de investigaciones realizadas y a los esfuerzos en mejorar continuamente los sistemas de control de posicionamiento, hoy día, se ha llegado a tal punto en que las metodologías para mejorar dichos sistemas han tenido un avance significativo. La problemática en el desarrollo de estos sistemas de control se puede analizar sobre dos vertientes principales: ámbitos científico y tecnológico. En el ámbito científico, se han desarrollado una gran variedad de

investigaciones relacionadas con la mejora en uno o más de los módulos que integran al sistema de control. Cada módulo se encarga de manipular alguna o diversas variables de interés, por ejemplo, en la generación de trayectorias se busca lograr un buen seguimiento de la herramienta con error mínimo y bajo ciertas restricciones dinámicas a fin de evitar vibración y saturación de los actuadores. Otro caso es en la identificación y modelado del sistema, por lo que, un correcto modelo permite una adecuada elección y sintonización de la ley de control, cuyo fin es minimizar el error en el seguimiento de referencias. Es precisamente en el módulo de sintonización de controladores para sistemas de posicionamiento donde menos se han concentrado las investigaciones, y donde incluso los controladores comerciales disponibles carecen de algoritmos óptimos para tal objetivo. Típicamente, para la tarea de sintonización o ajuste de parámetros del algoritmo de control se hace recurrencia a técnicas clásicas tanto empíricas como analíticas. Ejemplo de técnicas empíricas están las reglas propuestas por Ziegler y Nichols, ZN, o aquellas propuestas por Cohen y Coon, CC; por otra parte, ejemplo de técnicas analíticas se tienen el análisis de lugar geométrico de las raíces, LGR, o el análisis de respuesta en frecuencia, ARF. El problema con todas estas técnicas es que dependen del conocimiento de quien las usa, solo pueden ser usadas para espacios lineales de diseño, no todos los parámetros pueden ser mejorados, y fuertemente dependen de la exactitud del modelo matemático obtenido del sistema, por mencionar solo algunos puntos importantes. Por otro lado, el desarrollo formal en cuanto a los métodos de optimización modernos implican alguna o varias de las siguientes características a la vez: complejidad matemática, alta carga computacional y gran cantidad de recursos tecnológicos, las cuales se convierten en el mayor problema de dichos métodos. Estas características se convierten en factores que comprometen su implementación a través de una arquitectura digital en aplicaciones de tiempo real o en aplicaciones en línea, ya que en la mayoría de los casos los métodos se implementan en software y se aplican fuera de línea.

En el ámbito tecnológico se tiene que en diversos casos la implementación de las metodologías propuestas por muchos trabajos de investigación se realizan en sistemas basados en dispositivos especializados como microcontroladores o en procesadores digitales de señales, DSPs. Éstas tecnologías tienen una arquitectura en hardware que permite una alta velocidad de cómputo y se encuentran optimizadas en recursos. Sin embargo, sus desventajas son mayores cuando la metodología implementada o arquitectura de integración superan las capacidades del dispositivo debido a la complejidad computacional que se alcanza. Por otra parte, la arquitectura de estas tecnologías es completamente fija, y la consecuencia directa de ésta característica es que no se puede desarrollar una plataforma independiente donde la conectividad del hardware sea definida por el usuario, y por tanto, carece de la característica de portabilidad y configurabilidad. Es evidente que la problemática en el ámbito tecnológico observada en muchas investigaciones hasta el momento, es la implementación de soluciones propuestas en dispositivos que carecen de características como alta capacidad y velocidad de procesamiento, portabilidad y configurabilidad, etc. Además que las tecnologías en las que se desarrolla la mayoría de las aplicaciones son susceptibles de obsolescencia.

En resumen, la problemática en los ámbitos científico y tecnológico de los sistemas de control para posicionamiento en lo relacionado al módulo de sintonización de controladores es que se carece de propuestas de investigación relacionadas con la optimización formal de dicho módulo. Por otro lado, otro problema evidente es la carencia de arquitecturas que permitan implementar un método de optimización moderno, como los AG adoptados en el presente trabajo, ya que generalmente la implementación de estos se realiza por software y si se llega a proponer una arquitectura digital generalmente se valida por medio de simulaciones o fuera de línea en muchos casos, en ocasiones se llegan a usar

para la validación modelos ideales en los que no se consideran restricciones físicas presentes en sistemas reales.

1.4 Justificación

La investigación aquí presentada tiene dos vertientes y/o justificaciones: científica y tecnológica. Con base en el análisis realizado sobre los antecedentes, se concluye que existe una posibilidad real de aportar en el ámbito científico en el módulo de sintonización de controladores mediante el desarrollo de una metodología formal de optimización basada en una técnica meta-heurística como lo son los Algoritmos Genéticos. A pesar de que existen trabajos que han utilizado y mostrado la eficiencia de los AG en la sintonización de controladores, a la fecha no se tiene reporte concreto de la utilización de éstos en la optimización del proceso de sintonización en controladores PID para sistemas de posicionamiento o servosistemas. Para el desarrollo de la investigación será necesario utilizar el módulo de identificación de sistemas desarrollado por el grupo HSPdigital basado en el método de mínimos cuadrados, MMC, que proporcionará el modelo del sistema de posicionamiento usado para definir la metodología formal de optimización (Morales et al., 2010). Además, la justificación principal que lleva a realizar éste trabajo es la necesidad de desarrollar arquitecturas digitales que permitan la implementación del método de optimización basado en AG que otorgarán características inherentes tales como: una estructura optimizada en recursos y en velocidad de procesamiento, configurabilidad, procesamiento paralelo, etc.

En al ámbito tecnológico una implementación basada en un dispositivo de alto desempeño FPGA se justifica de la siguiente forma: los dispositivos FPGAs permiten bajo una arquitectura de procesamiento paralelo una alta velocidad de

cómputo, además el desarrollo de cualquier aplicación en estos dispositivos es descrita bajo un lenguaje de descripción en hardware que hace al diseño portable en una plataforma en hardware independiente. Permiten desarrollar soluciones tipo SoC que han estado ganando mercado fuertemente, ya que en estos dispositivos se pueden integrar unidades de procesamiento que son definidas por el usuario, así como otros periféricos relacionados o necesarios. Se combinan arquitecturas abiertas que no dependen de algún fabricante o plataforma específica, en este trabajo se integran módulos desarrollados en investigaciones previas. Este tipo de dispositivos proporcionan al diseñador la libertad suficiente para lograr una tarea específica desarrollando un nivel de arquitectura que también representa una unidad única de bajo costo. El arreglo de los bloques lógicos dentro del FPGA y la característica que le permite al usuario definir su interconectividad lo convierten en un dispositivo programable y configurable en una arquitectura completamente abierta. Por otra parte, la velocidad y el tamaño de esta tecnología son comparables con los circuitos integrados de aplicación específica, pero con mayor flexibilidad. La portabilidad del diseño basado en tecnología FPGA le confiere a los desarrollos realizados la inmunidad a la obsolescencia, problema tan recurrente en tecnología comercial y en aplicaciones industriales.

1.5 Planteamiento general del proyecto

El diagrama de la Figura 1.2, presenta el planteamiento general del proyecto. Por un lado se integran los conocimientos sobre sistemas de posicionamiento y la teoría de control en aspectos relacionados con el diseño de controladores digitales PID y se enfocan en el ajuste de sus parámetros (sintonización). Por otro lado, la optimización que se propondrá se basa en técnicas de búsqueda meta-heurísticas, de entre las que existen se adoptan los Algoritmos Genéticos, y se aplican al proceso de sintonización para desarrollar

una metodología forma de optimización de controladores PID en sistemas de posicionamiento, implementada inicialmente en software. La metodología propuesta se definirá de tal manera que sea susceptible de ser implementada a través de arquitecturas en hardware implementadas en un dispositivo de alto desempeño FPGA.

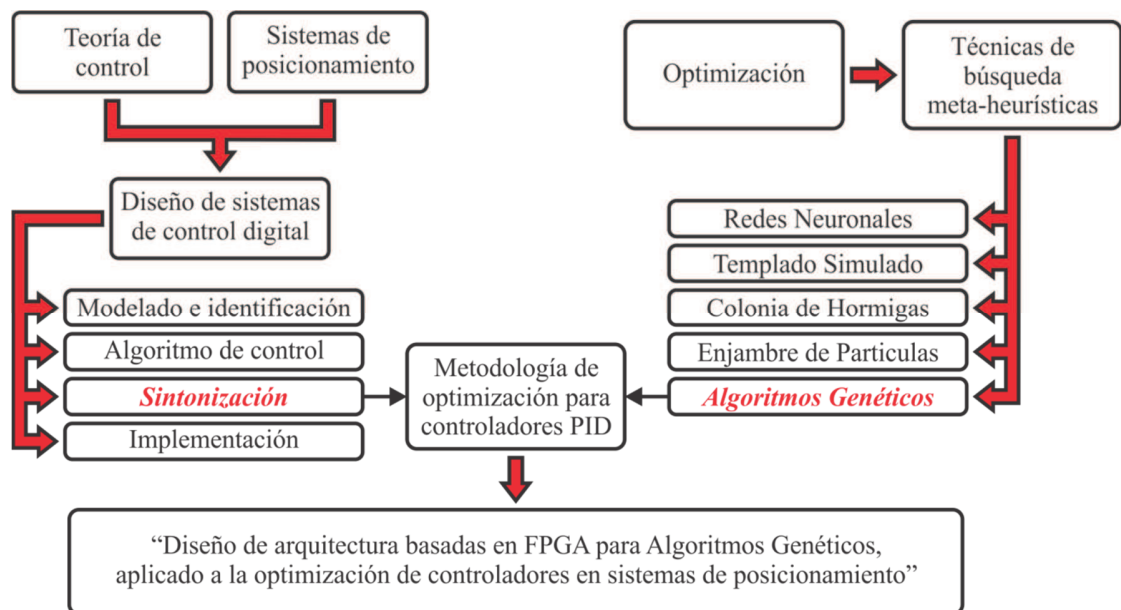


Figura 1.2. Planteamiento general de la metodología.

2. REVISIÓN DE LA LITERATURA

En este capítulo se presenta la recopilación de información relevante para el desarrollo del proyecto, los temas a tratar van desde los sistemas de posicionamiento, hasta la temática del problema de optimización y las técnicas meta-heurísticas así como el desarrollo de arquitecturas digitales.

2.1 Esquemas de control industrial

Se estima que más del 95% de los controladores que actualmente son implementados en la industria son los controladores de tres tiempos Proporcional-Integral-Derivativo (Aguado, 2000), y de éstos aproximadamente el 85% se implementan en lazos retroalimentados de una entrada y una salida (Li et al., 2006), tal como en la Figura 2.1.

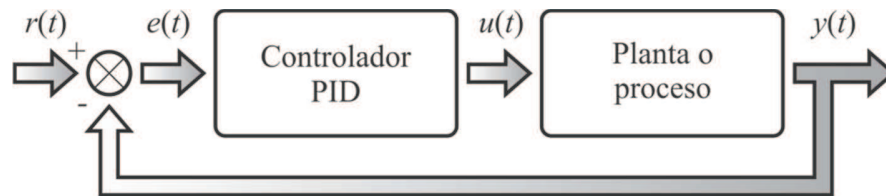


Figura 2.1. Esquema de control PID de lazo simple (una entrada y una salida).

El esquema simple de control PID es uno de los más utilizados en la industria por sus ya bien conocidas ventajas como un diseño simple tanto de concepto como en su implementación, robusto en funcionalidad y eficiente (dependiendo de la sintonización), y en general la mayor parte de los procesos pueden ser bien regulados con este tipo de esquema. A continuación en base al

análisis desarrollado por Milosawlewitsch (2010), se presentan algunas de las estructuras más utilizadas para el modelado de los controladores PID.

2.1.1 Estructura típica

La Figura 2.2 muestra la estructura más utilizada de un PID (estructura típica), sobre todo en simulaciones. Se observa del diagrama que la señal de excitación resultante, $u(t)$, es la suma de la señal de error, $e(t)$, afectada de manera independiente por las acciones proporcional, K_p , integral, K_i/s , y derivativa, $K_d s$, (West Instruments, 2003). Es la señal de salida, $u(t)$, la que excita a la planta para corregir el error.

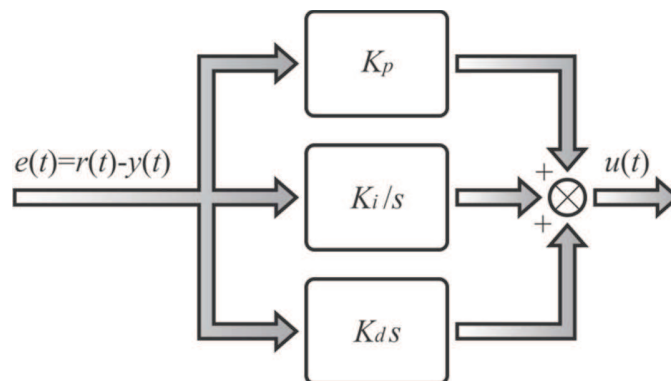


Figura 2.2. Diagrama a bloques de la estructura típica de un controlador PID.

La señal de excitación de salida en el dominio de Laplace se expresa mediante la ecuación (1):

$$U(s) = E(s) \left(K_p + \frac{K_i}{s} + K_d s \right) \quad (1)$$

De manera general, las variables que se deben ajustar en un controlador PID son la banda proporcional primaria, K , la constante de tiempo integral, T_i , y, la constante de tiempo derivativa, T_d . Por tanto las equivalencias entre las constantes del controlador y los tiempos de integración y derivativos, así como la ganancia de la banda proporcional son:

$$K_p = K \quad (2)$$

$$K_i = \frac{K}{T_i} \quad (3)$$

$$K_d = KT_d \quad (4)$$

Como se verá en el capítulo tres, es posible mediante la ecuación (1) obtener la función de transferencia del controlador que bien puede ser implementado en una estructura digital mediante una ecuación en diferencias.

2.1.2 Estructura práctica

De acuerdo con Lequin et al. (2003), la estructura PID más utilizada en aplicaciones industriales, es aquella en la que la acción derivativa no afecta a la señal de referencia; esto es, tanto la acción proporcional como la integral afectan la señal de error, $e(t)$, en tanto que, la parte derivativa actúa tan solo sobre la señal de salida, $y(t)$, Figura 2.3. De acuerdo con el diagrama, la señal de salida, $u(t)$, se puede escribir como:

$$U(s) = C_r(s)R(s) - C_y(s)Y(s) \quad (5)$$

Donde:

$$C_r(s) = K \left(1 + \frac{1}{T_i s} \right) \quad (6)$$

$$C_y(s) = K \left(1 + \frac{1}{T_i s} \right) + T_d s \quad (7)$$

Considerando que:

$$\frac{K_p}{K} = 1 \quad (8)$$

$$\frac{K_i}{K} = \frac{1}{T_i} \quad (9)$$

$$\frac{K_d}{K} = T_d \quad (10)$$

$$U(s) = R(s)K \left(1 + \frac{1}{T_i s} \right) - Y(s)K \left(1 + \frac{1}{T_i s} + T_d s \right) = K \left\{ (R(s) - Y(s)) \left[1 + \frac{1}{T_i s} \right] - Y(s)T_d s \right\} \quad (11)$$

O bien en el dominio del tiempo:

$$u(t) = K_p[r(t) - y(t)] + K_i \int [r(t) - y(t)] dt - K_d \frac{d}{dt} y(t) \quad (12)$$

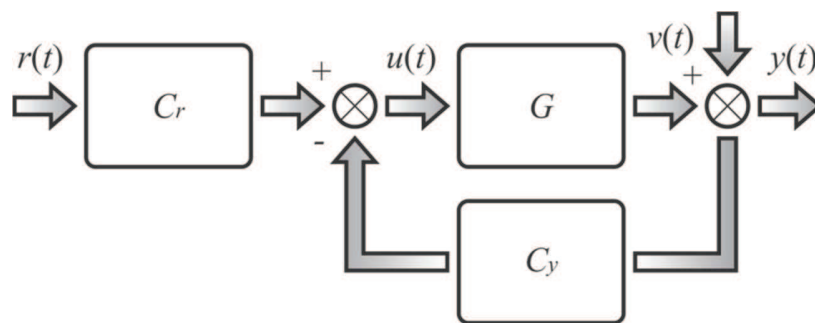


Figura 2.3. Estructura PID práctica en lazo cerrado.

2.1.3 Estructura ISA

La forma ISA (International Society of Automation; Sociedad Internacional de Automatización) es la estructura más completa de un PID de dos grados de libertad, y se expresa como:

$$U(s) = K \left\{ B[R(s) - Y(s)] + \frac{1}{T_i s} [R(s) - Y(s)] + \frac{T_d s}{\frac{T_d}{N} s + 1} C[R(s) - Y(s)] \right\} \quad (13)$$

Donde el término $\frac{1}{\frac{T_d}{N} s + 1}$, Corresponde a una parte derivativa filtrada. Si los coeficientes $B = C = 1$, se tiene la estructura de un PID con un grado de libertad. El objetivo de filtrar la acción derivativa es amortiguar la señal de salida del controlador, ya que al utilizar como referencia un escalón, la parte derivativa tiene por un instante a infinito. De las estructuras aquí presentadas la que se adoptará para el desarrollo de la metodología formal será la estructura típica por medio de una ecuación en diferencias como se explicará más adelante.

2.2 Sintonización de controladores PID

Al proceso de seleccionar los parámetros de un controlador que cumplan con unas especificaciones de desempeño se conoce como *proceso de sintonización* (Ogata, 1998). El objetivo de la sintonización en controladores PID es comúnmente definido como: “alcanzar un nuevo valor de referencia tan rápido como sea posible” (Milosawlewitsch, 2010). Existen diversos métodos para realizar la tarea de sintonización, sin embargo, este trabajo se concentra en el diseño de controladores PID aplicados a procesos industriales, específicamente hablando, el interés está en sistemas de posicionamiento o servosistemas. Por lo tanto, se puede clasificar a las metodologías de sintonización existentes en dos grupos

principales: métodos empíricos y métodos analíticos. Ambos grupos se consideran como metodologías clásicas y normalmente se efectúan en sitio y de forma manual. Existen algunas versiones de controladores comerciales que integran una funcionalidad de autoajuste de parámetros pero los resultados obtenidos generalmente requieren de un “ajuste fino” para que el sistema se comporte en la forma deseada. A continuación se describen estos grupos.

2.2.1 Métodos empíricos

Estos métodos se basan en el conocimiento empírico y en la obtención de valores por medio de pruebas experimentales sobre un proceso. Normalmente, los métodos empíricos requieren de la aplicación de reglas simples definidas a través de las relaciones existentes entre los valores obtenidos de forma experimental y los criterios que se deben cumplir en el desempeño del controlador.

En 1942 Ziegler y Nichols, ZN, presentaron dos métodos empíricos para la sintonización de controladores PID con base en la respuesta escalón o basadas en el valor experimental de la ganancia proporcional que se produce en la estabilidad marginal del sistema (cuando únicamente se usa la acción proporcional del controlador). Las reglas de ZN permiten determinar el valor de la ganancia proporcional, K_p , el valor del tiempo integral, T_i , y el valor del tiempo derivativo, T_d , con base en características de respuesta transitoria de una planta especificada. A continuación se tiene una descripción sobre estados dos metodologías.

➤ *Método de ZN de la respuesta escalón*

Para este método se analiza la respuesta al escalón obtenida de forma experimental del proceso en lazo abierto sin controlador. Si la planta no contiene

polos dominantes complejos conjugados, ni integradores la respuesta tiene la forma mostrada en la Figura 2.4.

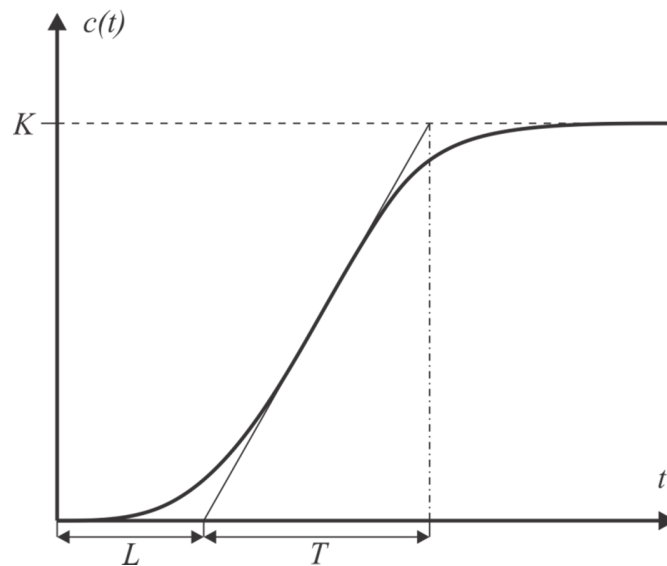


Figura 2.4. Curva de respuesta al escalón.

También se puede hacer el análisis para el modelo matemático de la planta mediante simulación de dinámica. Se pueden observar dos parámetros de interés en la respuesta, obtenidos al dibujar una línea tangente en el punto de inflexión de la curva y determinando su intersección con la línea de tiempo y el valor de la respuesta en estado estacionario, que son el tiempo de retardo, L , y la constante de tiempo, T .

Los valores de K_p , T_i , y T_d , se obtienen de las fórmulas propuestas por ZN y que aparecen en la Tabla 2.1.

Tabla 2.1. Reglas de ZN para sintonización por respuesta al escalón.

Tipo de controlador	K_p	T_i	T_d
P	$\frac{T}{L}$	∞	0
PI	$0.9 \frac{T}{L}$	$\frac{L}{0.3}$	0
PID	$1.2 \frac{T}{L}$	$2L$	$0.5L$

➤ *Método de ZN de la ganancia crítica*

En este método se considera la planta que se desea controlar en lazo cerrado integrando el controlador PID. Sin embargo, al inicio de este método de sintonización se requiere establecer los valores de las ganancias $K_i \approx 0$, y $K_d = 0$, dejando únicamente la acción proporcional en juego. El valor de K_p se especifica inicialmente en 0 y su valor se va incrementando hasta que la respuesta del sistema muestra un comportamiento oscilatorio sostenido, tal como se muestra en la Figura 2.5.

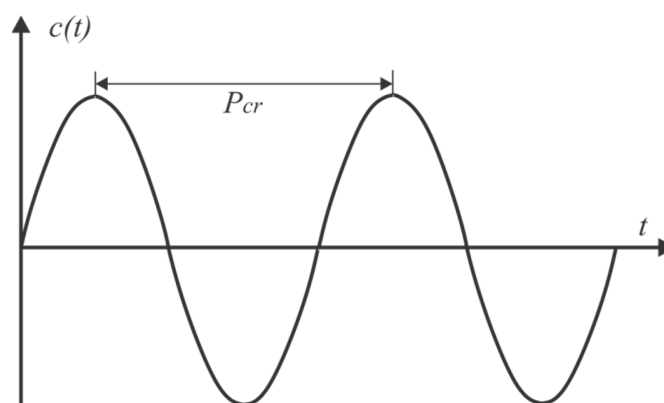


Figura 2.5. Respuesta de oscilación sostenida con P_{cr} .

En el momento en que la respuesta de la planta presenta oscilaciones sostenidas el valor de la ganancia proporcional es K_{cr} , y las oscilaciones tendrán un periodo P_{cr} . Estos valores se conocen como ganancia crítica y periodo crítico respectivamente. Las reglas de sintoniza propuestas por ZN para este método se presentan en la Tabla 2.2.

En general para ambos métodos se pretende cumplir con ciertas restricciones dinámicas, como un sobrepaso máximo general de 25% en la respuesta escalón. Sin embargo, en la práctica los métodos pueden proporcionar resultados que no son los más apropiados pese a que la planta que se desea controlar cumple bien con cada paso requerido por uno de los métodos. Sin mencionar que el tiempo requerido para la sintonización se puede incrementar si los resultados buscados no se obtienen pues se tiene que realizar un ajuste manual de los parámetros hasta alcanzar la respuesta requerida.

Tabla 2.2. Reglas de ZN para sintonización por ganancia crítica.

Tipo de controlador	K_p	T_i	T_d
P	$0.5K_{cr}$	∞	0
PI	$0.45K_{cr}$	$\frac{1}{1.2}P_{cr}$	0
PID	$0.6K_{cr}$	$0.5P_{cr}$	$0.125P_{cr}$

Si no se conocen los parámetros de la planta, estos métodos son muy útiles y prácticos ya que se basan en resultados experimentales. Por otro lado, si se conoce el modelo matemático de la planta y no es posible obtener los parámetros requeridos en forma experimental, se puede hacer simulación dinámica con el

modelo de la planta, considerando que los resultados son susceptibles de error si el modelo utilizado no es exacto.

2.2.2 Métodos analíticos

Los métodos analíticos son más precisos y pueden proporcionar mejores resultados en la sintonización de controladores que los métodos empíricos, ya que estos métodos se basan en un análisis detallado del modelo de la planta. Entonces, la efectividad de los parámetros obtenidos queda comprometida por la eficiencia en el modelado de la planta o la obtención de un modelo aproximado por métodos alternos. Una ventaja de los métodos analíticos es que se pueden tratar modelos de plantas de orden mayor y se pueden mejorar parámetros específicos en la respuesta del sistema tal como la estabilidad, el valor del error en estado estacionario, el valor del sobrepaso, y el tiempo de respuesta, entre otros. Ejemplos clásicos de métodos basados en el análisis son el lugar geométrico de las raíces, LGR, y el método de análisis de respuesta en frecuencia, ARF.

➤ *Diseño de controladores por LGR.*

El lugar geométrico de las raíces es una representación gráfica de los polos de una planta en lazo cerrado y se usa para describir cualitativamente la operación de un sistema cuando se cambian uno o varios parámetros. La representación de los polos se realiza en un plano complejo como en la Figura 2.6.

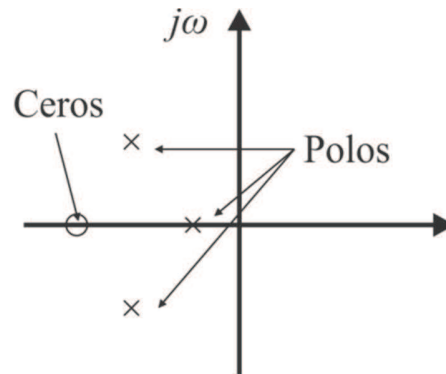


Figura 2.6. Representación de los polos y ceros de una planta en el plano complejo.

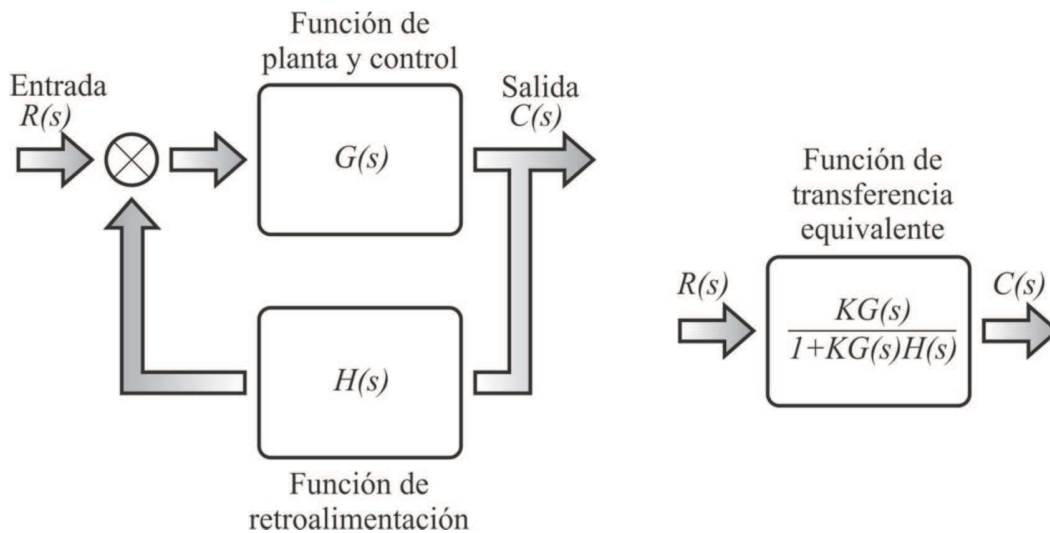


Figura 2.7. Sistema en lazo cerrado analizado por LGR.

Tal como se mencionó esta técnica es útil para analizar modelos de plantas de orden superior a dos. El análisis por LGR proporciona información gráfica de los márgenes tanto de estabilidad como de inestabilidad y permite conocer que condiciones cambian esta situación haciendo que una planta cambie de estable a inestable o viceversa. Adicionalmente, el comportamiento en el transitorio también se puede analizar al variar parámetros y observar como repercuten por ejemplo en

el porcentaje de sobrepaso, el tiempo de asentamiento o el tiempo pico, entre otras cuestiones. El LGR es la trayectoria que siguen los polos en lazo cerrado de un sistema cuando se varía algún parámetro como la ganancia del sistema por ejemplo. Finalmente, las propiedades del lugar geométrico de las raíces, que sirven para el análisis y diseño de sistemas de control, se obtienen a partir del sistema en lazo cerrado como se muestra en la Figura 2.7.

Entonces la función de transferencia equivalente del lazo cerrado, $G_{LC}(s)$, que integra las funciones del controlador, de la planta, y de la retroalimentación se expresa como:

$$G_{LC}(s) = \frac{KG(s)}{1 + KG(s)H(s)} \quad (14)$$

De la ecuación (14), un polo “s”, va a existir cuando sea cero el polinomio característico de la función en lazo cerrado, y se tiene entonces:

$$KG(s)H(s) = -1 = 1\angle(2k + 1)180^\circ; \quad k = 1, 2, 3 \dots \quad (15)$$

Usando estas propiedades se puede realizar el diseño de un sistema de control a partir de análisis por LGR.

➤ *Sintonización por ganancia y margen de fase, GMF*

El método de sintonización por GMF, es un método derivado del análisis ARF, y de acuerdo con Tal (1994) este método pretende atacar el diseño de un controlador en base a dos aspectos importantes de un sistema: su estabilidad, y su velocidad de respuesta. La estabilidad de un sistema indica la naturaleza de la respuesta. Por ejemplo, el nivel de estabilidad de un sistema está directamente relacionado con la cantidad de sobrepaso que presenta en su respuesta, y a esto

se le llama amortiguamiento, por lo tanto, entre menor sea el valor de amortiguamiento mayor será el sobrepaso. Normalmente, los sistemas inestables pueden llegar a presentar un comportamiento oscilatorio antes de llegar a su valor en estado estacionario. Dentro del análisis ARF la estabilidad se expresa comúnmente como un margen de fase, θ_m , y entre mayor sea su valor mayor será el nivel de estabilidad. Se tiene como regla general en el análisis ARF que un sistema será estable si el valor de θ_m es positivo y será inestable cuando su valor sea negativo, además en un rango de valores de 30° a 45° se presenta comúnmente una respuesta amortiguada apropiada. Por otro lado, la velocidad de respuesta en un sistema indica cuanto tiempo le toma realizar una pequeña corrección que no cause saturación (tiempo de respuesta). En el análisis ARF la respuesta de un sistema se caracteriza por la frecuencia de cruce, ω_c , que está directamente relacionada con el tiempo de respuesta, $\tau = 1/\omega_c$.

El análisis por ARF para sistemas de posicionamiento inicia con el modelo matemático en lazo abierto del sistema, $G_{LA}(s)$. Esta función incluye el algoritmo de control, $G_c(s)$, el modelo de la planta, $G_p(s)$, la constante del actuador, K_a , y los retrasos asociados con el muestro ejecutado por el controlador digital, $G_d(s) = e^{-Ts/2}$. El sistema considerado para el análisis se muestra en la Figura 2.8.

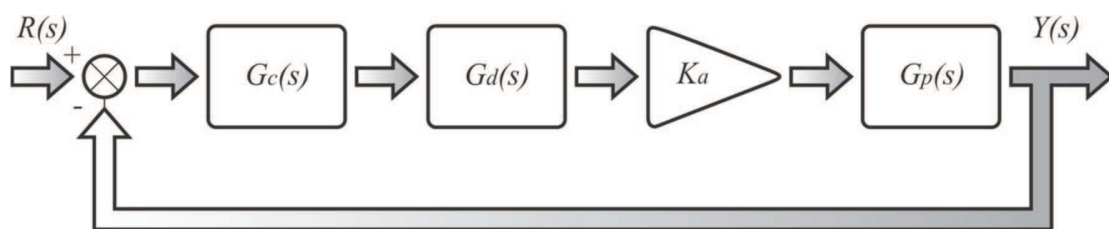


Figura 2.8. Diagrama a bloques de los elementos para ARF.

Entonces la función de transferencia $G_{LA}(s)$ queda expresada como:

$$G_{LA}(s) = G_c(s)G_d(s)G_p(s)K_a = \left(P + \frac{I}{s} + Ds\right) \cdot \left(e^{-\frac{T_s}{2}}\right) \cdot G_p(s) \cdot K_a \quad (16)$$

Para evaluar la respuesta de la función en el dominio de la frecuencia, $G_{LA}(j\omega)$, se sustituye “s” por “j ω ”. Ahora, el método de sintonización por GPM requiere que se satisfagan dos criterios para la estabilidad que son:

$$|G_{LA}(j\omega)| = 1 \text{ para } \omega = \omega_c \quad \text{y} \quad \theta_m = 180^\circ + \arg[G_{LA}(j\omega)] \quad (17)$$

Considerando ahora los elementos de los bloques en $G_{LA}(j\omega_c)$ se obtiene:

$$|G_{LA}(j\omega_c)| = \left|P + \frac{I}{j\omega_c} + Dj\omega_c\right| \cdot 1 \cdot K_a \cdot A \quad (18)$$

$$\theta_m = 180^\circ + \arg[G_{LA}(j\omega_c)] - \frac{\omega_c T}{2} \left(\frac{180^\circ}{\pi}\right) + \alpha \quad (19)$$

De la ecuación (18), que son las magnitudes aportadas por cada función de transferencia individual, se tiene que $A = |G_p(j\omega_c)|$, y la función del retardo tiene una magnitud unitaria, por lo tanto no afecta en la ecuación. Por otra parte, en la ecuación (19), que son las aportaciones de fase de cada función, se tiene que la fase asociada con el retardo por el efecto del muestreo en el sistema digital es $(\omega_c T/2)(180^\circ/\pi)$, y mientras tanto la planta aporta $\alpha = \arg[G_d(j\omega_c)]$. Además, el efecto de la acción integral es más pronunciado en bajas frecuencias, por lo que, se puede asumir que no afecta en el valor de la magnitud, mientras que causa un ligero degrado en la fase que puede ser limitado a -5° . Así, las expresiones anteriores se pueden reescribir como:

$$|P + Dj\omega_c| \cdot K_a \cdot A = 1 \quad (20)$$

$$\theta_m = 175^\circ + \tan^{-1}\left(\frac{D \cdot \omega_c}{P}\right) - \left(\frac{\omega_c \cdot T \cdot 180^\circ}{2\pi}\right) + \alpha \quad (21)$$

La ecuación (20) se interpreta como la relación existente entre las magnitudes del controlador PID y el resto de las funciones en $G_{LA}(j\omega)$. Así mismo, la ecuación (21) representa la suma de la contribución en fase de todos los elementos del sistema en lazo abierto para lograr el margen de fase requerido. Por lo tanto, se puede definir como β únicamente a la contribución en fase del controlador, y de esta manera se derivan expresiones más útiles para el método de sintonización GPM.

$$|P + Dj\omega_c| = \frac{1}{K_a \cdot A} \quad (22)$$

$$\tan^{-1}\left(\frac{D \cdot \omega_c}{P}\right) = \theta_m - 175^\circ + \left(\frac{\omega_c \cdot T \cdot 180^\circ}{2\pi}\right) - \alpha = \beta \quad (23)$$

Las relaciones expresadas en las ecuaciones (22) y (23) se muestran gráficamente en la Figura 2.9. Y finalmente, las ecuaciones (24) a (26) establecen los valores de las ganancias del controlador PID para sistemas de posicionamiento.

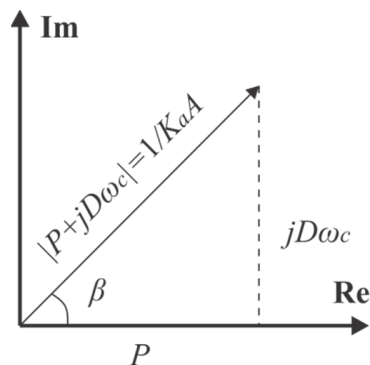


Figura 2.9. Relación de las ganancias del controlador PID.

$$P = \frac{\cos(\beta)}{A \cdot K_a} \quad (24)$$

$$D = \frac{\sin(\beta)}{A \cdot K_a \cdot \omega_c} \quad (25)$$

$$I = \tan(5^\circ) \cdot P \cdot \omega_c \quad (26)$$

En el presente trabajo se adopta el método de GMF para la definición de la metodología formal de optimización. Además la validación de la metodología propuesta se valida comparándola con otros métodos de entre los cuales se encuentra este.

2.3 Abordando el problema de optimización

El problema común en la mayoría de las investigaciones aplicadas a la industria es la existencia de un valor ideal o valores límite y cuantificables en los métodos que se proponen, es decir, un valor óptimo. Un problema de optimización, por tanto, es un ajuste en medida para conseguir un valor más deseable de la misma, o bien reducir la variación alrededor de esta medida. La palabra optimización utilizada estrictamente se refiere simultáneamente a la optimización de la medida y a la reducción de la variación (Figuera, 2006). En la industria, diversas aplicaciones requieren de la investigación en técnicas de optimización para mejorar el sistema de control existente, como en el caso de los sistemas de control para posicionamiento. Las tendencias de investigación son algoritmos de optimización, conocidos también como técnicas meta-heurísticas, para buscar el valor deseado de variables específicas (valor óptimo). Por lo tanto, se requiere del desarrollo de metodologías formales que se basen en tales técnicas, y el desarrollo de arquitecturas donde se puedan implementar para una rápida ejecución en la tarea de búsqueda y optimización (en ocasiones muy limitada). Uno de los grandes problemas de la optimización, al menos en los sistemas digitales, son los recursos y la complejidad computacional que se alcanza, que son el factor limitante para implementaciones en procesos de tiempo

real o en implementaciones en línea. Actualmente, se han realizado diversas investigaciones a nivel local e internacional en las que se aborda la problemática de los sistemas CNC y robots industriales, pero el esfuerzo aún no se concentra formalmente en el desarrollo de arquitecturas para la optimización del sistema.

2.4 Técnicas meta-heurísticas

Uno de los temas más importantes y que será utilizado para el desarrollo de la presente investigación es el de las técnicas de optimización modernas, también conocidas como técnicas meta-heurísticas. Algunas de estas técnicas son:

- Optimización por enjambre de partículas
- Optimización por colonia de hormigas
- Templado simulado
- Programación evolucionaria
- Algoritmos genéticos
- Lógica difusa
- Redes neuronales
- Combinaciones de las técnicas anteriores

Estas técnicas están basadas en fenómenos que ocurren en la naturaleza y consideran principios, operaciones, y algoritmos para conformar las metodologías utilizadas en la búsqueda de valores óptimos a problemas en los que las técnicas clásicas basadas en gradientes no son eficientes en dar solución, quizá por la naturaleza del problema. En general, los algoritmos genéticos son una poderosa técnica aplicada a problemas de optimización mediante una búsqueda heurística. Los AG están basados en la teoría de Darwin sobre la supervivencia del más apto,

y tienen muchas ventajas sobre las técnicas clásicas de optimización. Por ejemplo, los AG no requieren de información basada en operaciones de gradiente que implican el uso de cálculo diferencial, cálculo integral, o el asumir condiciones iniciales específicas según el problema de optimización. Lo que se menciona en líneas anteriores significa que el uso de los AG puede ser extendido a espacios de búsqueda no solamente lineales, sino también no-lineales, así como no-convexos. Además de que por su naturaleza y estructura los AG pueden ser usados para la búsqueda de valores óptimos en problemas de una o múltiples variables.

Sin embargo, una de las características inherentes de esta técnica es la gran cantidad de operaciones que se realizan a través de un proceso iterativo, esto aunado a la gran cantidad de recursos computacionales requeridos para su implementación. De acuerdo con lo mencionado con anterioridad, sería conveniente desarrollar una metodología basada en los AG que minimice la cantidad de recursos computacionales que se requieren para poder implementar dicha metodología más allá de una mera implementación en software, que en muchos casos se valida por simulaciones utilizando modelos matemáticos teóricos o se implementa fuera de línea.

2.5 Algoritmos Genéticos Estándar, AGE

Los AGE fueron originalmente introducidos por Holland en 1975, y están basados en los principios de la genética y la selección natural, filosóficamente se basan en la teoría de Darwin de la “supervivencia del más apto” (Rao, 2009). En los AGE las variables a optimizar se denominan variables de diseño y se encuentran representadas por medio de cadenas binarias de tamaño finito que representan analógicamente a los genes naturales. Así mismo, una agrupación de variables de diseño (agrupación de genes) forma un individuo, representando a su

vez a un cromosoma. Una cantidad finita de individuos (un número finito de cromosomas) es lo que conforma la población de los AGE, que inicialmente se generan de forma aleatoria. El objetivo de tener una cantidad significativa de individuos aleatorios es para abarcar la mayor parte del espacio de diseño, el cuál es el rango de valores en el que se pueden hallar los valores óptimos buscados. En otras palabras, se realiza una distribución aleatoria de individuos de tal forma que alguno o algunos de ellos se localicen cerca del valor óptimo. Entonces los individuos se seleccionan y se ejecuta la recombinación por medio de las operaciones genéticas, *cruza* de bits entre individuos y *mutación* o alteración de bits entre individuos. Por analogía con la naturaleza, a través de la selección de individuos aptos, soluciones potenciales al problema de optimización viven para reproducir, y los individuos débiles tienden a desaparecer. La recombinación de la población seleccionada origina una nueva población y el proceso se repite hasta que toda la población se asemeja, momento en el que se ha evolucionado hasta un valor nominal o valor óptimo. La figura 2.10 muestra un esquema en el que representa de forma general el concepto de los AGE.

Una característica notoria de los AGE es que mantiene durante todo el proceso de búsqueda una población fija a diferencia de otras metodologías que solo trabajan con o pocas variables de diseño, es decir, los AGE son un proceso de optimización multi-objetivo. Además, por la generación inicial aleatoria y las operaciones genéticas, los espacios de diseño (o rangos de búsqueda) pueden ser tanto lineales, como no lineales. En base al análisis presentado por Renner y Ekárt (2003), a continuación se presenta la información más relevante para este trabajo sobre los AGE.

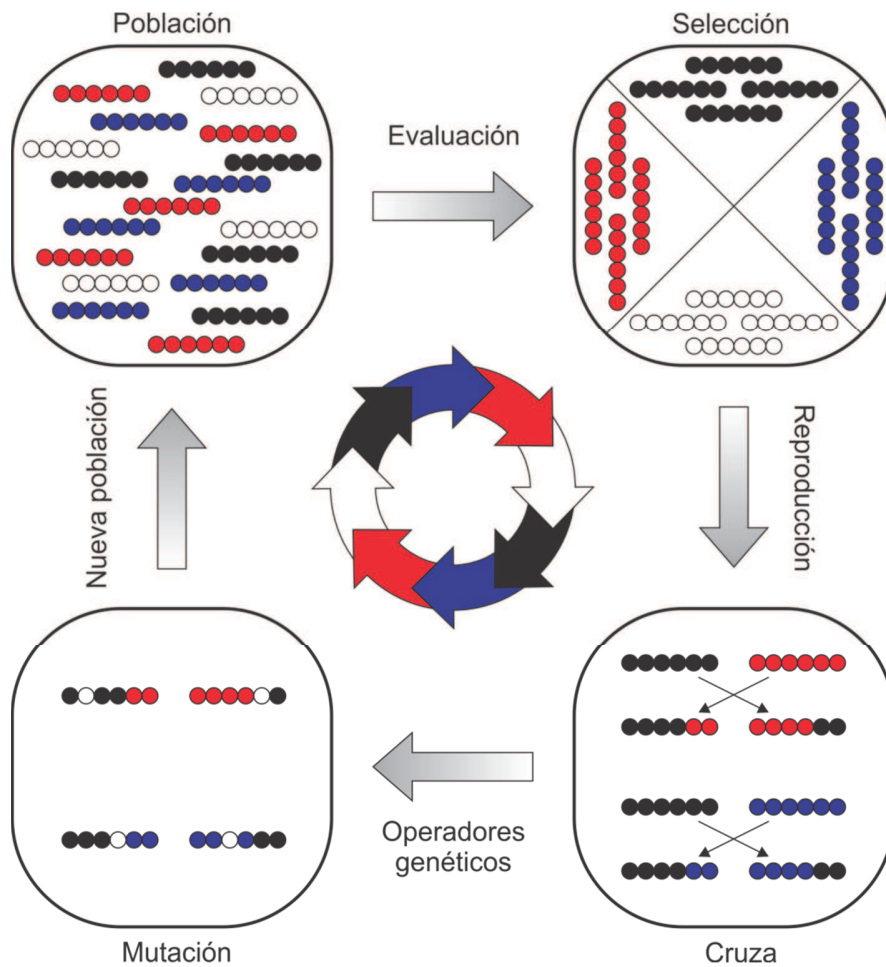


Figura 2.10. Concepto general de los AGE.

2.5.1 Estructura de los AGE

Cuando se resuelve un problema utilizando los AGE, primero una representación adecuada y medición de la adaptabilidad deben ser diseñadas. El siguiente paso debe ser definir un criterio de terminación, que usualmente permite llevar a cabo un número de iteraciones y posteriormente verificar si una solución aceptable ha sido encontrada, Figura 2.11.

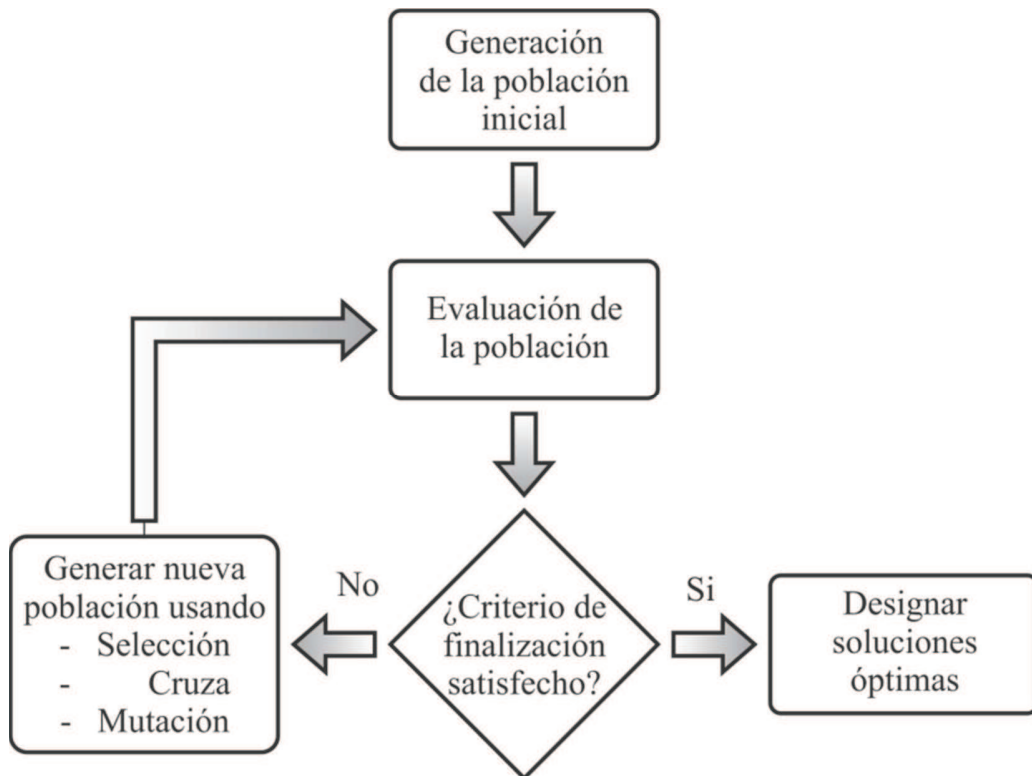


Figura 2.11. Estructura general de los AGE.

De acuerdo con la estructura general mostrada en la figura, la operación de los AGE se describe a continuación.

1. La población inicial se llena con individuos que son generados aleatoriamente. En ocasiones los individuos son soluciones encontradas por métodos determinados por el dominio del problema.
2. Cada individuo de la población es evaluado de acuerdo a un criterio de desempeño, que finalmente produce un valor de adaptabilidad.
3. Si el criterio de terminación se cumple, se retorna la mejor solución encontrada.

4. De la población actual, los individuos son seleccionados basados en los valores de adaptabilidad calculados previamente. La adaptabilidad de un individuo indica si debe eliminarse, ya que no es una buena solución, o si debe mantenerse, ya que es una solución con potencial que podría derivar en la solución óptima. Una nueva población se forma aplicando los operadores genéticos a los individuos, los cuales son: selección, cruce y mutación. Los individuos seleccionados son llamados padres y los individuos resultantes su descendencia. Las implementaciones de los AGE difieren en la forma de construcción de la nueva población. Existen AGE que no realizan generaciones en absoluto, pero si continuo reemplazo.
5. Las acciones realizadas a partir del paso 2 se repiten hasta que el criterio de terminación se cumple. Una iteración es llamada generación.

2.5.2 Representación de los AGE

Cuando se diseñan los AGE para un problema dado, la elección de la representación (es decir, construcción de cromosomas) es el primer paso. Algunas representaciones pueden conducir exitosamente a buenas soluciones, mientras a otras les toma mucho tiempo o fallan en converger. La razón es porque la representación junto con la recombinación de operadores limita la exploración del espacio de diseño a ciertas regiones. Una representación común es la cadena binaria de longitud fija. En el caso de problemas más complejos una representación más sofisticada conduce a mejores resultados. La Figura 2.12 muestra gráficamente la representación antes descrita y clarifica la diferencia entre variable de diseño e individuo, gen y cromosoma respectivamente en analogía con la genética natural.

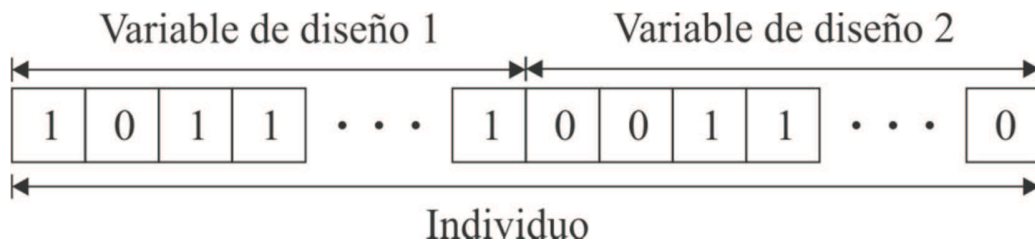


Figura 2.12. Representación de individuos binaria para los AGE.

2.5.3 Operadores genéticos

En cada generación (iteración), los operadores genéticos son aplicados a los individuos seleccionados de la población actual a fin de crear una nueva población, este proceso se conoce de forma general como recombinación. Usando probabilidades para aplicar estos operadores, la velocidad de convergencia puede ser controlada. A continuación se hace una descripción de cada operador genético:

- **Selección.** Una parte de la nueva población puede ser creada copiando sin modificar a individuos de la población actual, de aquellos individuos que representan soluciones altamente potenciales de ser valores óptimos o converger a ellos. Esto da la posibilidad de supervivencia de soluciones aptas ya desarrolladas. El resto de los individuos debe seleccionarse en base a criterios de desempeño, los cuales dependen del problema de optimización. Los individuos menos potenciales son eliminados.
- **Cruza.** Nuevos individuos generalmente son creados como descendencia de los padres, es decir, de la combinación de información entre cadenas binarias que dan lugar a nuevas cadenas (como tal, la cruza es un operador binario). Uno o más puntos de cruza son seleccionados, usualmente de forma aleatoria, en la cadena binaria de cada individuo y en el mismo lugar

para los individuos que van a ser operados. Las partes delimitadas por los puntos de cruce son intercambiadas entre los padres, como muestra la Figura 2.13. Cabe aclarar que existen operadores de cruce más sofisticados

- **Mutación.** Un nuevo individuo es creado haciendo modificaciones en el individuo seleccionado. Las modificaciones pueden consistir en cambiar, borrar o agregar uno o más valores en la representación. En los AGE la mutación es una fuente de variabilidad y se aplica adicionalmente a la selección y cruce, simplemente intercambiando el valor binario en el punto o puntos de mutación seleccionados por su complemento. La Figura 2.14 muestra un ejemplo gráfico de esta operación considerando el resultado mostrado en la Figura 2.13 como antecedente.

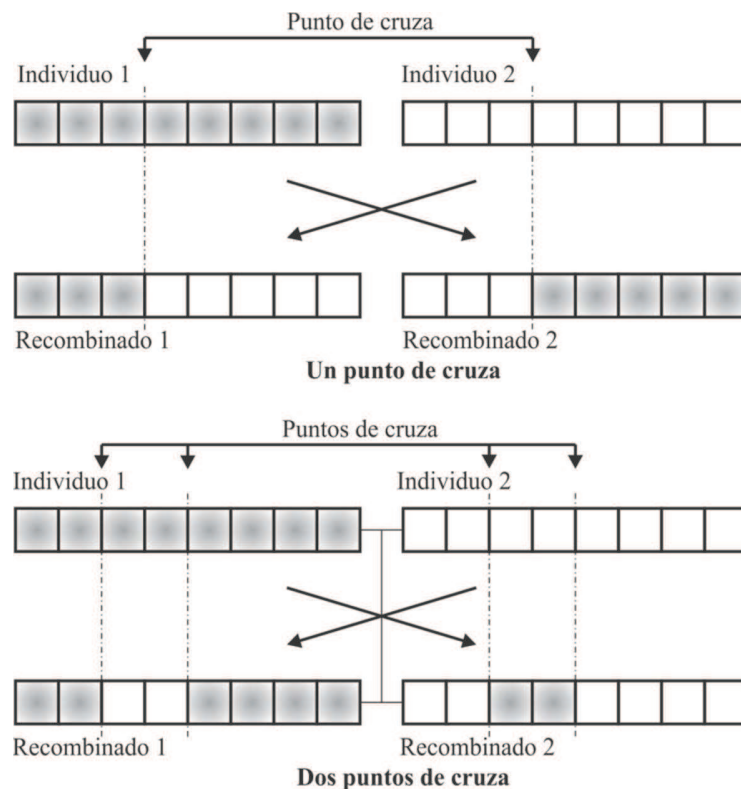


Figura 2.13. Ejemplo gráfico de operación de cruce en los AGE.

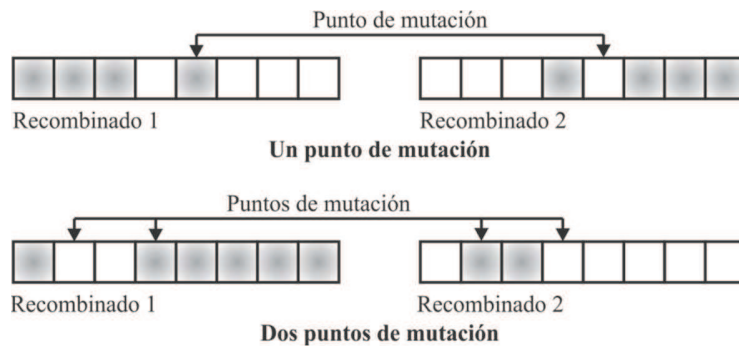


Figura 2.14. Ejemplo gráfico de operación de mutación en los AGE.

2.5.4 Asignación de valor de adaptabilidad

La probabilidad de supervivencia de cualquier individuo se determina por su adaptabilidad: a través de la evolución de los individuos más aptos por sobre los más débiles. Por tanto, la función de adaptabilidad debe indicar que tan bien cumple con los requerimientos la solución para un problema dado. La asignación de adaptabilidad se puede realizar de diferentes maneras:

- Se define una función de adaptabilidad que es incorporada en los AGE. La función de adaptabilidad es evaluada para todos los individuos.
- En muchos casos la función de adaptabilidad se calcula por software de análisis dedicado en forma separada. En tales casos, el cálculo puede tomar mucho tiempo, volviendo lento el algoritmo.
- En ocasiones no existen funciones de adaptabilidad explícita y por tanto un evaluador debe asignar la adaptabilidad según el problema que se le presente.

2.5.5 Métodos de selección

La selección es basada en la adaptabilidad, los individuos con mejor valor de adaptabilidad se toman con mayor frecuencia que los individuos de peor adaptabilidad. Algunos esquemas de selección comúnmente utilizados son:

- **Selección de adaptabilidad proporcional.** El mecanismo es similar a una ruleta dividida en piezas, cada individuo tiene una parte proporcional a su adaptabilidad. Cuando la ruleta gira, un individuo tiene una oportunidad de ser seleccionado correspondiente al tamaño de su parte.
- **Selección por rangos.** En la selección basada en rangos, los individuos son ordenados de acuerdo a su adaptabilidad. Los individuos son entonces seleccionados con una probabilidad basada en alguna función lineal de su rango.
- **Selección por torneo.** Un conjunto de n individuos es seleccionado de una población en forma aleatoria. Entonces el mejor del grupo es seleccionado. Para $n = 1$, el método es equivalente a la selección aleatoria. Cuando más alto el valor de n mejor se establece la selección dirigida hacia los mejores individuos.

Debido a las características de los AG, esta técnica es una excelente candidata para ser adaptada a la optimización de controladores PID usados en sistemas de control de posicionamiento online a través de una estructura digital. Por lo tanto, para llevar a los AG a una implementación en un sistema digital antes se reducirán los recursos computacionales adoptando un concepto bastante útil que ha tenido lugar de forma relativamente reciente dentro de la literatura, y es el “concepto de micro-población” que da origen a la variante MAG.

2.6 Micro-Algoritmos Genéticos, MAG

Se introduce el tema de los MAG en ésta subsección ya que conviene describir cuáles serán las consideraciones y diferencias respecto a los AGE en el desarrollo de la metodología de optimización propuesta en la presente investigación. Hasta ahora ya se han mencionado las características, ventajas, y desventajas de los AGE; por lo tanto, una forma de superar las limitaciones acerca de la gran cantidad de recursos que se demandan es aplicar el concepto de “micro-población”. La micro-población es una característica variante de los AG en la cual se define una cantidad muy pequeña de individuos que conforman la población del algoritmo, Figura 2.15.

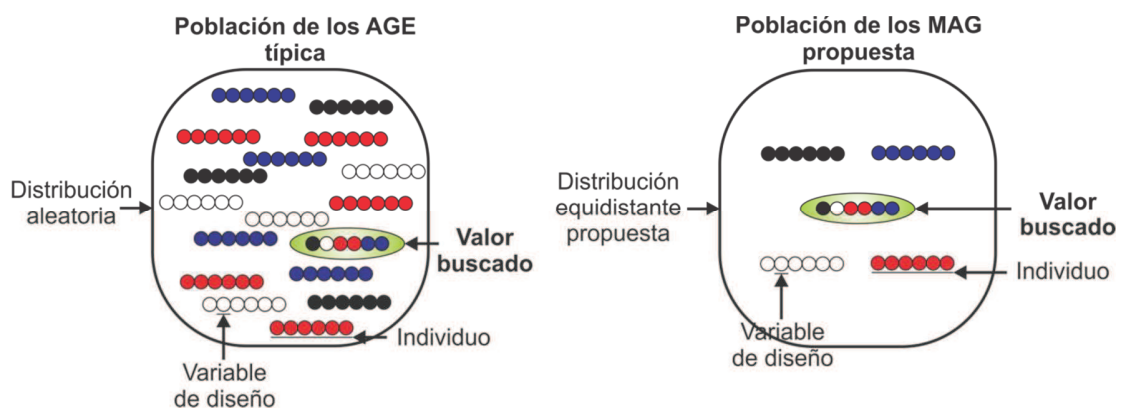


Figura 2.15. Concepto de micro-población de los MAG.

De acuerdo con la literatura una micro-población se puede componer desde cuatro hasta ocho o diez individuos, no se tiene un estándar específico como tal. Adicionalmente a la micro-población, se pueden definir operadores genéticos de tal manera que una rápida convergencia y una búsqueda de los valores óptimos globales estén asegurados. Los algoritmos genéticos con una pequeña población tiene un comportamiento similar a un AGE, pero se disminuirán significativamente

los recursos computacionales requeridos, ya que la cantidad de operaciones se minimizan al ser pocos individuos, además de que la cantidad de iteraciones se reducen siempre y cuando los operadores genéticos definidos sean eficientes. En conclusión, los MAG son aquellos con esta característica de micro-población y reducción de recursos, con el consecuente incremento en la velocidad de búsqueda de valores óptimos.

3. METODOLOGÍA

En el presente capítulo se describe en detalle, a partir de la recopilación de información en el estado del arte y de los antecedentes, el desarrollo de la metodología formal de optimización basada en una técnica meta-heurística como lo son los AG. Se adopta una variante de los AGE, descritos en el capítulo anterior, conocida como Micro-Algoritmos Genéticos, MAG, y que será descrita en las secciones siguientes. Esta metodología se desarrolla para ajustarse a la optimización de parámetros en controladores PID utilizados en sistemas de posicionamiento, como lo son los servosistemas CNC. En la metodología se adecúan los conceptos y algoritmos de los MAG para operar en el esquema de control de un PID típico a través de una metodología de sintonización clásica, en este trabajo se utiliza el método de GMF, que ha probado ser útil en la sintonización de parámetros en servosistemas (Tal, 1994). La Figura 3.1 muestra el diagrama general de la metodología seguida. A continuación se da una breve descripción de cada fase mostrada en la figura y en los subtemas siguientes se describirán en detalle.

La primer etapa consiste en el desarrollo de la metodología de ajuste de parámetros basada en los MAG; una técnica meta-heurística de optimización usada para la búsqueda de valores óptimos en problemas de una o múltiples variables con las características descritas en la sección 2.5. La etapa dos y tres son la implementación de la metodología desarrollada en software, de la cual se analizarán y discutirán los resultados obtenidos. La etapa cuatro consiste en el diseño de las arquitecturas digitales para la implementación de la metodología propuesta en un controlador digital, con el fin de optimizar los parámetros de un algoritmo PID para servosistemas.

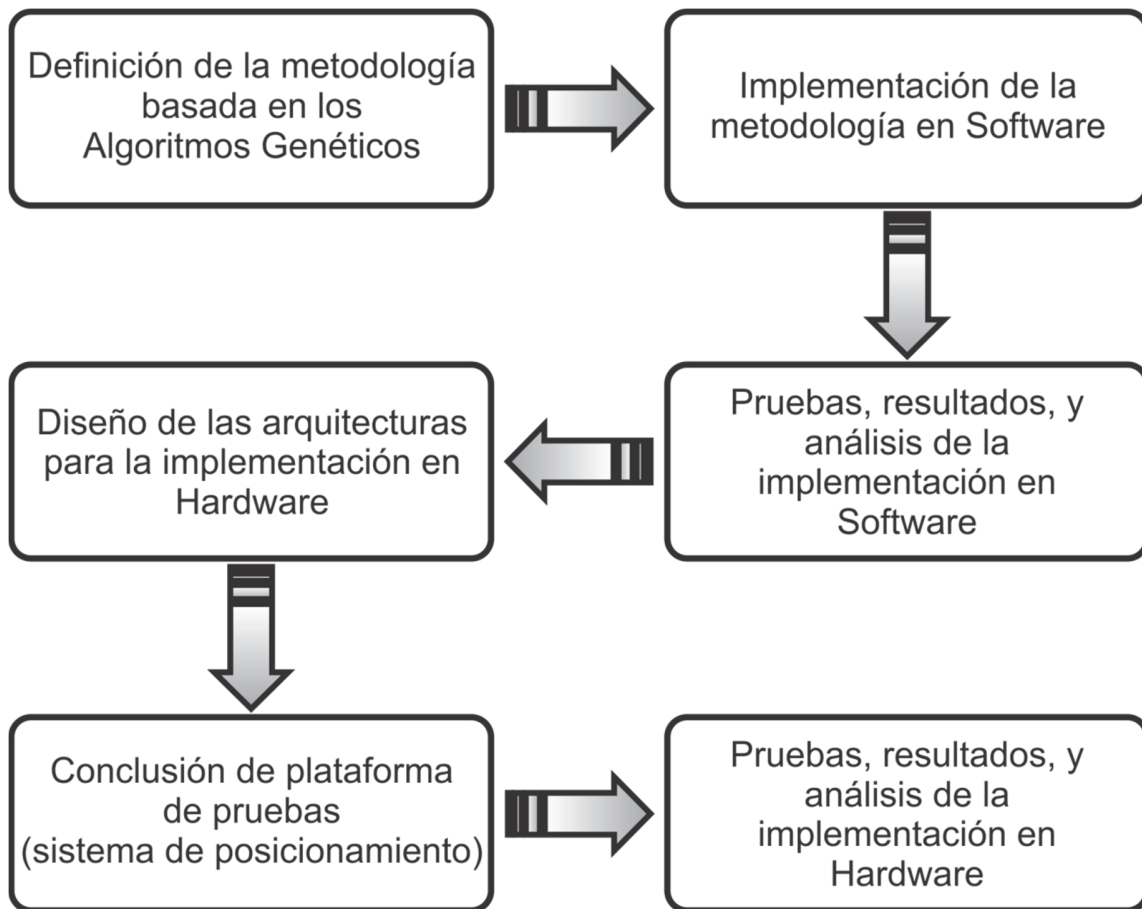


Figura 3.1. Diagrama general de la metodología seguida.

La quinta etapa consiste en la conclusión de la plataforma de pruebas en la que se validará el funcionamiento de la metodología propuesta, es decir, se concluye la instrumentación y control de un servosistema prototipo que consiste en una fresadora CNC de tres ejes. Finalmente, la etapa seis implica la implementación de las arquitecturas en hardware en el controlador basado en FPGA para realizar la optimización en línea del algoritmo PID usado en el control de la máquina CNC.

3.1 Definición de la metodología basada en los MAG

En el capítulo anterior se explicó la teoría en detalle sobre los AGE, y se presentó el esquema general en la Figura 2.11, en el cual se observa que se trata de una técnica de búsqueda heurística de valores óptimos para un problema de optimización, y en la que se cumple un criterio de satisfacción que condiciona el fin de un proceso iterativo. La búsqueda de los valores óptimos inicia definiendo una cantidad significativa de valores iniciales de forma aleatoria, por lo que su distribución inicial abarca todo el espacio de búsqueda. Estos valores iniciales representan las posibles soluciones del problema de optimización, pero es gracias a los operadores genéticos que se logra adicionar a la metodología la variabilidad que el proceso iterativo requiere de tal forma que se logra modificar a estas soluciones iniciales hasta que se convierten en el valor óptimo buscado. En resumen, los valores iniciales van evolucionando a lo largo del proceso de optimización mediante mecanismos genéticos especificados, operaciones de selección, cruza y mutación, hasta alcanzar una convergencia nominal, momento en el cual se ha encontrado la solución o soluciones óptimas.

La metodología a desarrollar seguirá un esquema que minimice los recursos computacionales requeridos por los AGE. Por lo tanto, se utilizó como esquema base a los Micro-Algoritmos Genéticos, Figura 3.2, para ello se hará hincapié en los puntos principales que marcan la diferencia respecto a los AGE, y se enlistan a continuación:

1. Se aplica el concepto de **micro-población**, visto en la sección 3.1 del presente capítulo.
2. Se usa una distribución equidistante de la población inicial en lugar de generarlos aleatoriamente.

3. Se seleccionan operadores genéticos simples que consuman menos recursos, permitiendo incrementar la velocidad de cómputo.
4. Se definen y especifican los criterios de desempeño, así como la función objetivo en base al problema de optimización de sistemas de posicionamiento.
5. Se utiliza una probabilidad de mutación adaptativa.

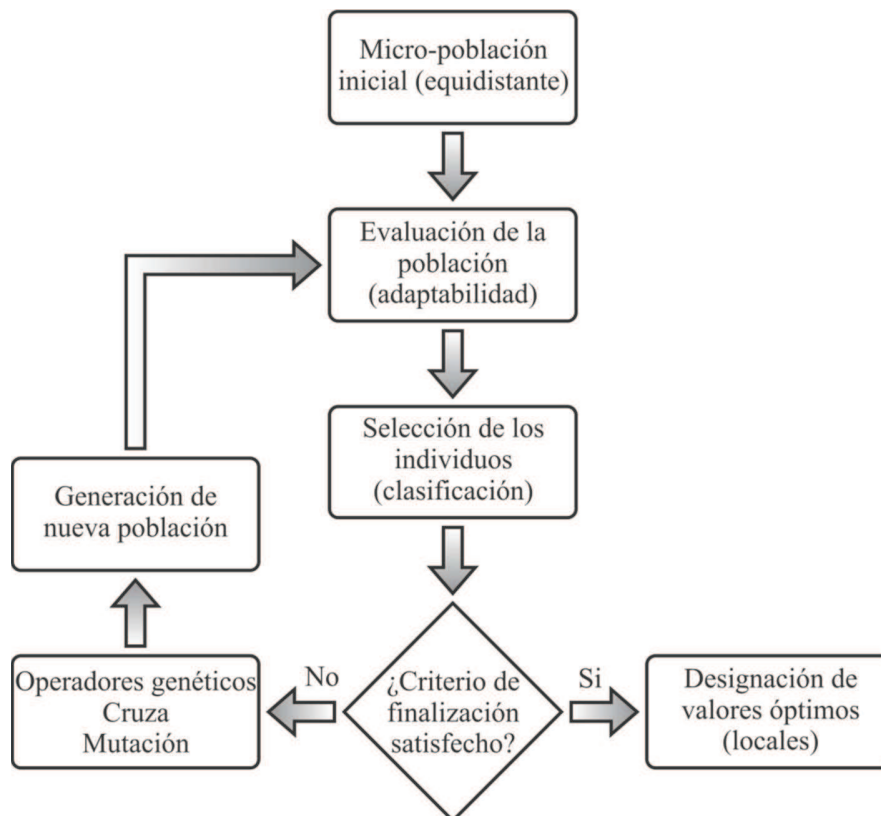


Figura 3.2. Esquema general de los MAG.

A continuación se explicará en detalle cada uno de los puntos anteriores. En el primer punto se tiene que si unos cuantos individuos son generados,

entonces la cantidad de operaciones y el tiempo de cómputo se reducen. El objetivo es minimizar los recursos computacionales requeridos por los AGE convencionales. En la presente investigación se define una población de solo cuatro individuos configurados tal como se observa en la Figura 3.3. Se puede apreciar de la figura que cada individuo consiste en cadenas binarias que concatenadas representan al individuo, cada cadena binaria corresponde a una variable de diseño que para la presente investigación serán las variables que se desean optimizar.

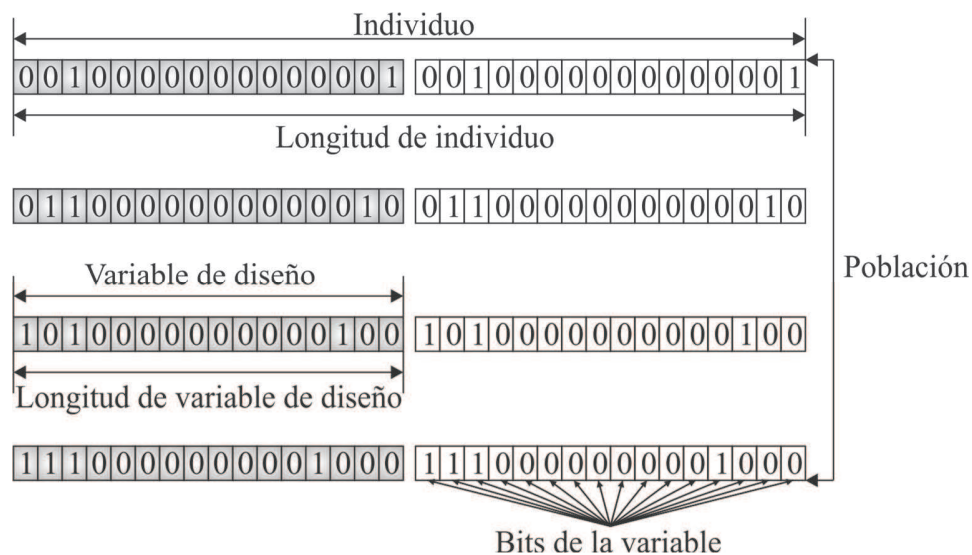


Figura 3.3. Estructura de individuos y variables de diseño en los MAG.

La decodificación de cada individuo se realiza de acuerdo a la ecuación (27):

$$v_{real} = v_{min} + \frac{v_{max} - v_{min}}{2^{li} - 1} \cdot d_{ind} \quad (27)$$

Dónde:

v_{real} , es el valor real del individuo.

v_{max} , es el máximo valor del rango de diseño.

v_{min} , es el mínimo valor del rango de diseño.

d_{ind} , es el valor decimal del individuo.

l_{ind} , es la longitud de la cadena binaria de cada individuo.

Ahora bien, los rangos de diseño se especifican según el problema y las variables que se desean optimizar. En este escrito se definen como variables de diseño para el desarrollo de la metodología formal a los parámetros requeridos por el método de sintonía de GMF. Entonces, el rango de diseño será todo el conjunto de valores que tomarán el margen de fase, θ_m , y el tiempo de respuesta, τ .

El punto número dos complementa y cubre el requisito de los AGE sobre la distribución inicial de los individuos generados aleatoriamente. Recordando que al adoptar el concepto de micro-población, existe la posibilidad de generar una población de unos cuantos individuos que se encuentren muy cercanos unos de otros en un amplio espacio de búsqueda, si se generan de forma aleatoria. En cambio, al distribuirlos de forma equidistante se asegura que los pocos individuos utilizados se encuentren distribuidos a lo largo de todo el espacio de búsqueda, como se observa en la Figura 3.4.

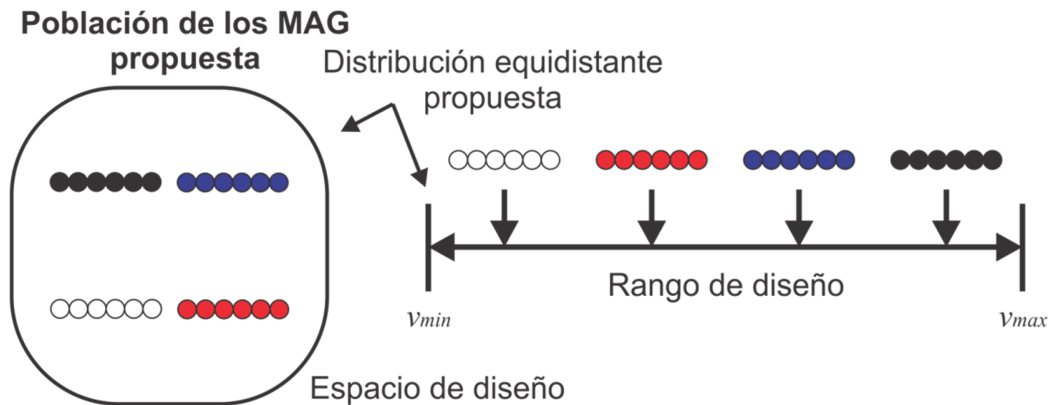


Figura 3.4. Distribución inicial de los individuos para los MAG.

El tercer punto se refiere a la correcta selección de los operadores genéticos que como sabemos son tres principalmente: la selección, la cruce y la mutación. Se define un proceso de selección elitista, un proceso de cruce de punto simple y una mutación de un punto como los operadores genéticos en el presente trabajo, los diagramas representativos son como los mostrados en las Figuras 2.13 y 2.14.

El cuarto punto establece que es necesario definir los criterios de desempeño que proporcionan el valor de adaptabilidad de los individuos. Esto es, que cada individuo debe ser evaluado y, por lo tanto, tiene un desempeño, este desempeño se cuantifica al asignarle al individuo un valor de adaptabilidad. Para esto, los individuos que son seleccionados son clasificados de acuerdo a su valor de adaptabilidad, $f=1/J$, típicamente proporcionado mediante una función objetivo, J . En este trabajo se adoptan la suma de la integral del error absoluto, IEA , y la integral del error cuadrático, IEC , definidas en las ecuaciones (28) a (30).

$$IEA = \int_0^{\infty} abs[e(t)] dt \quad (28)$$

$$IEC = \int_0^{\infty} [e(t)]^2 dt \quad (29)$$

$$J = IEA + IEC \quad (30)$$

Finalmente, el punto número cinco establece que el mecanismo de variabilidad en el proceso de optimización además de la que proporciona por sí mismo la operación de cruza, es definir una probabilidad de mutación adaptiva. Esto significa que se definirá una probabilidad de mutación al inicio del proceso de optimización de forma empírica, y a lo largo del proceso iterativo esta probabilidad irá disminuyendo de forma lineal. El objetivo de esto es incrementar la velocidad de búsqueda de los valores al problema de optimización en las iteraciones iniciales y decrementarla en las finales a fin de evitar la pérdida de información genética importante. La Tabla 3.1 muestra en resumen los parámetros especificados para la metodología propuesta y en las secciones siguientes se describe el esquema desarrollado en esta sección, considerando los cinco puntos, ajustados a la optimización de los parámetros de un controlador PID usado en un sistema de control de posicionamiento.

Tabla 3.1. Definición de parámetros del método propuesto basado en MAG.

Parámetro	Valor
Máximo número de iteraciones, G	64
Tamaño de la población, P_s	4
Longitud de la variable (bits), l_{ind}	16
Probabilidad de cruza, C_p	1
Probabilidad de mutación adaptiva (valor inicial), aM_p	0.4
Valor objetivo, J	$IAE + ISE$
Valor de adaptabilidad, f	$1/J$

3.2 Implementación de la metodología en Software

Los puntos mencionados en la sección anterior sobre los MAG se consideran en esta sección para la optimización de los parámetros de un controlador PID para sistemas de control de posicionamiento. En esta sección se describe la implementación de la metodología en software. El diagrama de la Figura 3.5 muestra el esquema de los MAG modificado para optimizar los parámetros del controlador de forma indirecta a través del método de sintonización GMF basado en el análisis de ARF.

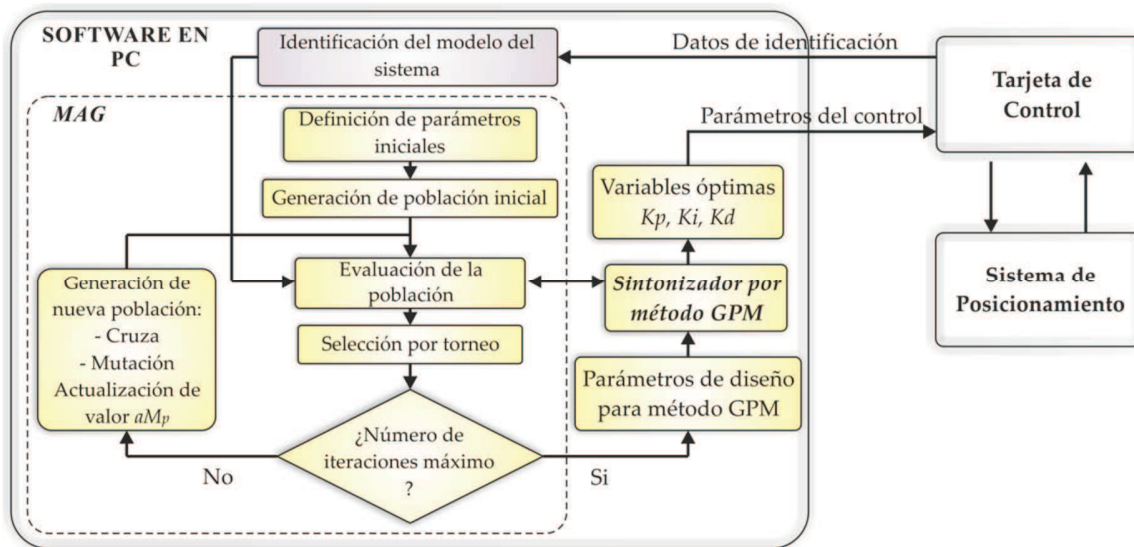


Figura 3.5. Esquema para la implementación en software de la metodología basada en MAG.

En la figura se pueden apreciar los siguientes bloques principales:

- **Sistema de posicionamiento.** Que en este caso se trata de un servosistema, la plataforma de pruebas principal consiste en una máquina fresadora CNC de tres ejes descrita en la sección 3.4.

- **Tarjeta de control.** Consiste en una tarjeta de control basada en un dispositivo del alto desempeño FPGA en la cual se encuentran implementados los módulos en hardware con las funciones necesarias para lograr un sistema en lazo cerrado para control de posición de la máquina y visualización de datos en una interfaz de usuario en la PC.
- **Software PC.** Dentro de este bloque se tiene toda la metodología basada en los MAG y funciones adicionales integradas para la optimización de parámetros del controlador.

A continuación se explicará en detalle la implementación en software de la metodología propuesta de optimización, cabe resaltar que esta implementación se realiza en línea aunque no en tiempo real. Como ya se mencionó antes esta metodología acopla a los MAG con la técnica de sintonización por GMF, por lo tanto de aquí en adelante se llamara a esta metodología propuesta MAG-GMF.

En primer lugar se explicará el bloque de la PC, dentro de este se tiene la función de “identificación del modelo del sistema”, este tema se encuentra fuera del alcance de la investigación por lo que no se entrará en detalle sobre el mismo, pero vale la pena mencionar que el Método de Mínimos Cuadrados, MMC, se define para llevar a cabo la tarea de la identificación. La metodología propuesta se basa en el análisis de ARF, por lo que, para proponer las ganancias de un controlador éste método requiere de algunos parámetros de la planta que se va a controlar como lo son su aportación de fase y magnitud; para ello se requiere de un modelo matemático. Entonces, a fin de obtener el modelo del sistema de posicionamiento, pequeñas referencias aleatorias se generan en el módulo de identificación en la PC y se envían al controlador, este a su vez envía las señales

de control a la máquina para ejecutar los movimientos. Los datos recolectados usando la información de un encoder incremental óptico, se envían a la tarjeta de control y a la PC en secuencia, donde se emplea el MMC para obtener un modelo matemático de la máquina. El modelo se obtiene en forma discreta como se muestra en la ecuación (31).

$$G(z) = \frac{c_n z^n + c_{n-1} z^{n-2} + \dots + c_0 z^0}{d_n z^n + d_{n-1} z^{n-2} + \dots + d_0 z^0} \quad (31)$$

El método de GMF requiere de la aportación en fase, α , y magnitud, A , de la planta, como se describió en la sección 2.2. Por lo tanto, para una planta discreta de cualquier orden la magnitud y fase se obtienen a partir de la forma compleja de z definida en la ecuación (32).

$$z = Ae^{-j\omega} = A \cdot \cos(\omega) + j \cdot A \cdot \sin(\omega) \quad (32)$$

Por lo tanto las componentes real e imaginaria se obtienen a partir de las ecuaciones (33) y (34).

$$re = \sum_{i=1}^n c_{n+1-i} \cdot \cos[(n+1-i) \cdot \omega] \quad (33)$$

$$im = \sum_{i=1}^n d_{n+1-i} \cdot \sin[(n+1-i) \cdot \omega] \quad (34)$$

Donde n es el orden del polinomio del numerador y del denominador de acuerdo con la ecuación (31), c_{n+1-i} y d_{n+1-i} son los coeficientes del polinomio del numerador y del denominador, respectivamente. Finalmente, la aportación de α y A de la planta se obtiene al calcular la tangente inversa y el módulo de las partes real e imaginaria, ecuaciones (33) y (34).

Dentro del cuadro punteado en la Figura 3.5 se encuentra el esquema de los MAG propuesto, el cual se utiliza para optimizar los parámetros de entrada del método de sintonización por GMF; que son el tiempo de respuesta, τ , y el margen de fase, θ_m . Estos dos parámetros se toman como las variables de diseño de los MAG con la estructura mostrada en la Figura 3.3, por lo tanto, cada individuo en la población de los MAG estará conformado por dos cadenas binarias concatenadas con una longitud de cadena especificada por l_{ind} . Se debe resaltar que los parámetros iniciales de los MAG están especificados de acuerdo a la Tabla 3.1, y que se genera inicialmente la micro-población equidistantemente a todo el ancho del rango de diseño de cada variable que se va a optimizar. Cada individuo de la población es evaluado, para ello son decodificados de acuerdo con la ecuación (27), y sometidos al método de GMF, la salida del método de sintonización son los parámetros del controlador PID. Posteriormente, estos parámetros se ingresan en un bloque de control en MATLAB que evalúa su desempeño usando la respuesta simulada del modelo de la planta obtenido mediante los MMC, como se explicó con anterioridad. La respuesta en lazo cerrado de los modelos controlador-planta se registra y se calcula una señal de error entre el desempeño deseado y el obtenido, y esa información se utiliza para obtener los índices de desempeño, de acuerdo con las ecuaciones (28) y (29), que corresponden a los criterios de desempeño especificados. En base al desempeño obtenido se asigna un valor de adaptabilidad, ecuación (30), que en este caso se utiliza para clasificar y seleccionar a los individuos. La selección es como sigue: el individuo más fuerte se copia directamente en la siguiente población, el peor individuo se reemplaza de forma aleatoria y los dos individuos restantes se obtienen por medio de los operadores genéticos de cruce y mutación, los cuales se describen en la sección 2.5. El proceso de optimización se detiene hasta que el criterio de número máximo de generaciones, parámetro G de la Tabla 3.1, es satisfecho. Finalmente, si el criterio de fin del proceso se satisface, se asignan los parámetros óptimos al

controlador digital, en el cual se tiene implementado el algoritmo de control PID, y se validan los resultados de la metodología propuesta usando estos parámetros en la ejecución de movimientos en el servosistema real como se verá en el siguiente capítulo.

3.3 Diseño de arquitecturas para implementación en Hardware

Para la implementación en hardware de la metodología propuesta en las secciones 3.1 y 3.2, se considera el diagrama mostrado en la Figura 3.6. Se puede observar que se trata de un esquema similar al mostrado en el diagrama de la Figura 3.5 pero con varias diferencias significativas. En esta ocasión se observa que la metodología no se encuentra como una función en software, es simplemente el esquema de los MAG con una estructura genérica tal que permite hacer la descripción de una arquitectura digital. Es decir, la estructura del diagrama tiene como entradas y salidas datos digitales, y dentro del bloque se tienen únicamente operaciones relacionadas con eventos genéticos. En los siguientes párrafos se presentará la descripción detallada sobre la implementación en hardware. En esta ocasión se definen las ganancias del controlador directamente como las variables de diseño de los MAG, es decir, las constantes, K_p , K_i , K_d , son ahora los parámetros a optimizar por la metodología propuesta. *“Por ahora solo se dirá que adecuar los MAG para la optimización de parámetros a través de una técnica clásica de sintonización para un PID es susceptible de inexactitud en tanto el modelo de la planta sea inexacto también, en el siguiente capítulo se analizan los resultados obtenidos por los cuales se realizó en cambio en las variables de diseño originales”.*

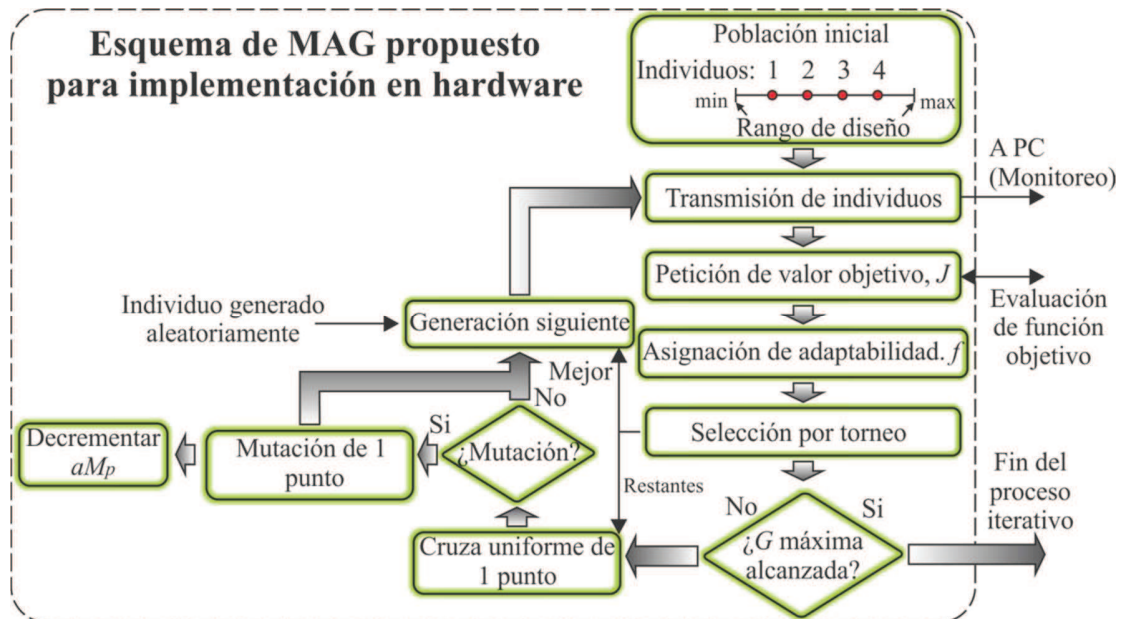


Figura 3.6. Esquema para la implementación en Hardware de la metodología basada en los MAG.

Los parámetros generales de los MAG para su implementación en hardware también obedecen a los valores mostrados en la Tabla 3.1. Se definen los mismos operadores genéticos de selección, cruce y mutación hasta ahora usados, así como los criterios de desempeño de las ecuaciones (28) y (29) para la asignación de la adaptabilidad, ecuación (30). También, la dinámica para la generación de la nueva población es aplicada tal como en la implementación en software descrita en la sección anterior.

A diferencia de la implementación en software en esta ocasión el esquema de los MAG no será acoplada a ninguna técnica de sintonización clásica, y como ya se mencionó, de la observación de los resultados del siguiente capítulo se concluye que el esquema de los MAG tiene un mejor comportamiento cuando afecta directamente a las ganancias del controlador. El esquema de los MAG se

integrará en esta ocasión junto con los módulos del sistema dentro de la tarjeta de control basada en FPGA y se incluirá un procesador embebido propietario para el control de flujo de datos como se verá más adelante. La principal razón y justificación de utilizar una tarjeta de control que incluye un dispositivo de alto desempeño es que tiene características como procesamiento paralelo, al ser una estructura descrita en hardware tiene una mayor velocidad de procesamiento, puede ser configurable y un usuario puede desarrollar todos aquellos módulos con las funciones que el sistema necesite. En la Figura 3.7 se muestra el diagrama general de una plataforma hardware-software para la implementación de la metodología propuesta.

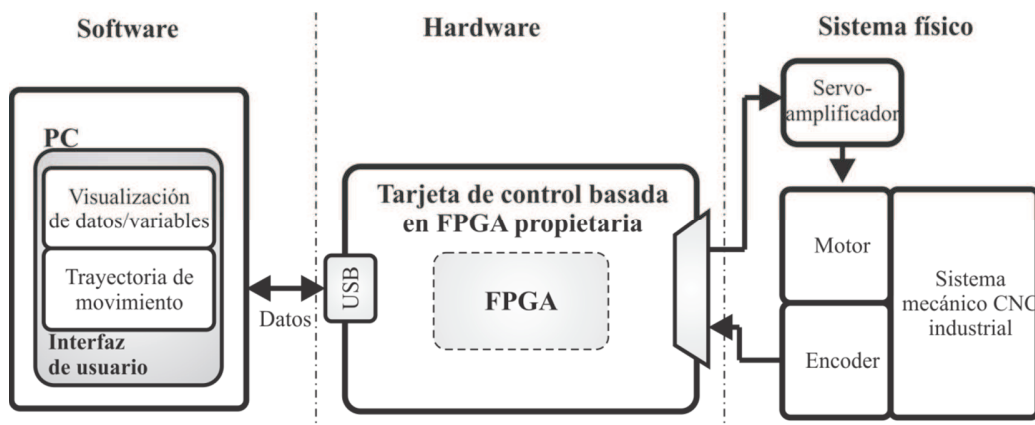


Figura 3.7. Sistema embebido para optimización de parámetros en línea.

La Figura 3.7 muestra un sistema compuesto por tres bloques principales:

- Software
- Hardware
- Sistema físico

La implementación de la metodología basada en los MAG, se realizará a través de dos estrategias, ambas consisten en arquitecturas digitales en hardware dentro del FPGA, tal como se muestra en la Figura 3.8. La primer estrategia

consiste en una implementación dentro de un procesador propietario embebido en hardware bajo una codificación en software dentro del procesador, a partir de ahora se definirá a esta estrategia como estrategia en software, SW. La segunda estrategia consiste en el desarrollo de un módulo en hardware que funcione como un co-procesador genético que brinde soporte al procesador embebido, de aquí en adelante esta estrategia será referida como estrategia en co-diseño hardware-software, HS.

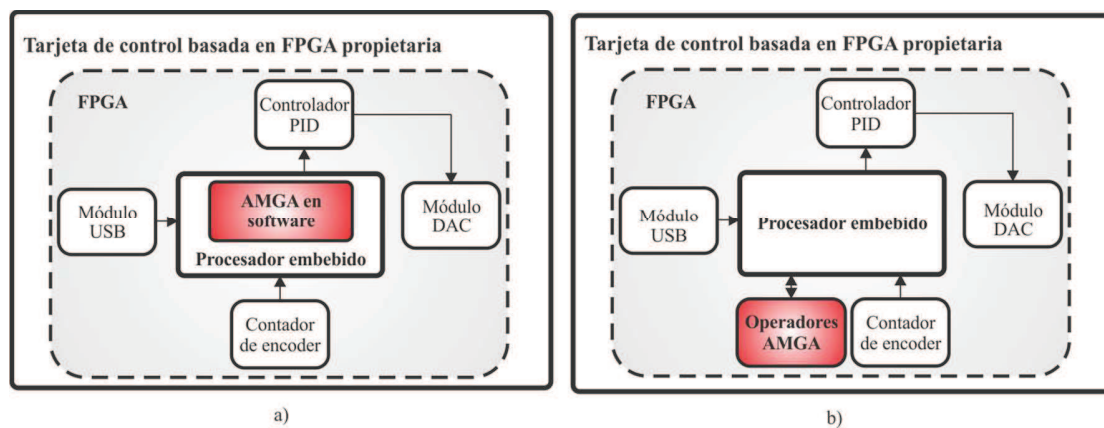


Figura 3.8. Estrategia en a) software y b) hardware para la implementación de la metodología basada en los MAG.

Puede parecer confuso que hablando de una implementación en hardware se pongan expresiones tales como “estrategia en software”, así que a fin de aclarar las cosas se tiene que; la tarjeta de control está basada en un dispositivo FPGA, el dispositivo FPGA contiene los módulos en hardware necesarios para lograr un sistema de control en lazo cerrado en conjunto con el sistema físico y la interfaz de usuario. Entre los módulos en hardware se tiene un procesador embebido propietario, desarrollado por miembros del grupo de investigación HSPdigital de la Universidad Autónoma de Querétaro. Finalmente, dentro del procesador embebido se tiene implementada la metodología basada en el esquema de los MAG a través de la estrategia SW. En esta estrategia el esquema

se implementa mediante código C, ya que es una característica del procesador, que puede ser configurado y programado haciendo uso de este lenguaje. En la segunda estrategia, HS, se describe la arquitectura de un módulo co-procesador genético mediante lenguaje de descripción en hardware, HDL, que realiza las operaciones genéticas y sirve de soporte al procesador embebido principal.

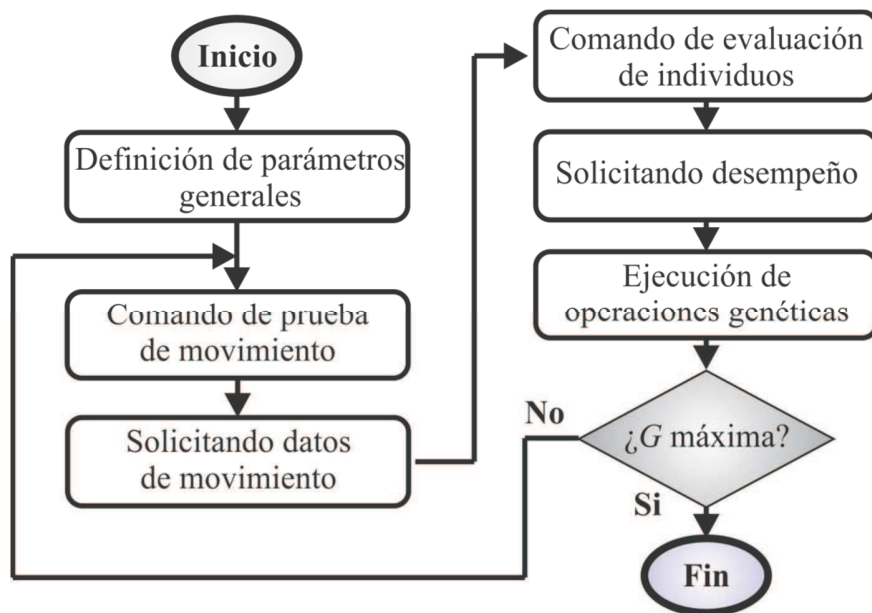


Figura 3.9. Acciones seguidas en el procesador embebido para las estrategias SW y HS.

El bloque del software está constituido por una interfaz de usuario en la PC, la cual controla y administra las acciones que serán ejecutadas en el procesador; además la visualización de los datos se lleva a cabo aquí. El procesador propietario embebido y todos los periféricos requeridos para el proceso de optimización se describen bajo HDL, *específicamente se trabaja con el lenguaje de descripción en hardware para circuitos integrados de muy alta velocidad, VHDL*, y son sintetizados en una tarjeta de control propietaria de bajo costo basada en FPGA, que en conjunto integran el bloque del hardware. Finalmente, el

bloque del sistema físicamente nuevamente está constituido por el servo-sistema, la máquina fresadora CNC de tres ejes. La secuencia de acciones administradas por la interfaz de usuario y ejecutadas en el procesador embebido se presenta mediante el diagrama de flujo de la Figura 3.9. Estas acciones se siguen tanto para la estrategia SW como para la estrategia HS.

3.3.1 Implementación mediante la estrategia en SW

Esta implementación será dentro del procesador embebido en codificación por software, por lo tanto, se realiza de acuerdo al diagrama de flujo mostrado en la Figura 3.10. Los eventos genéticos programados dentro del procesador se presentan mediante el pseudocódigo mostrado en la Tabla 3.2. Se puede observar que, estos únicamente incluyen las operaciones de selección, cruce, mutación y el cambio en la probabilidad de mutación adaptativa.

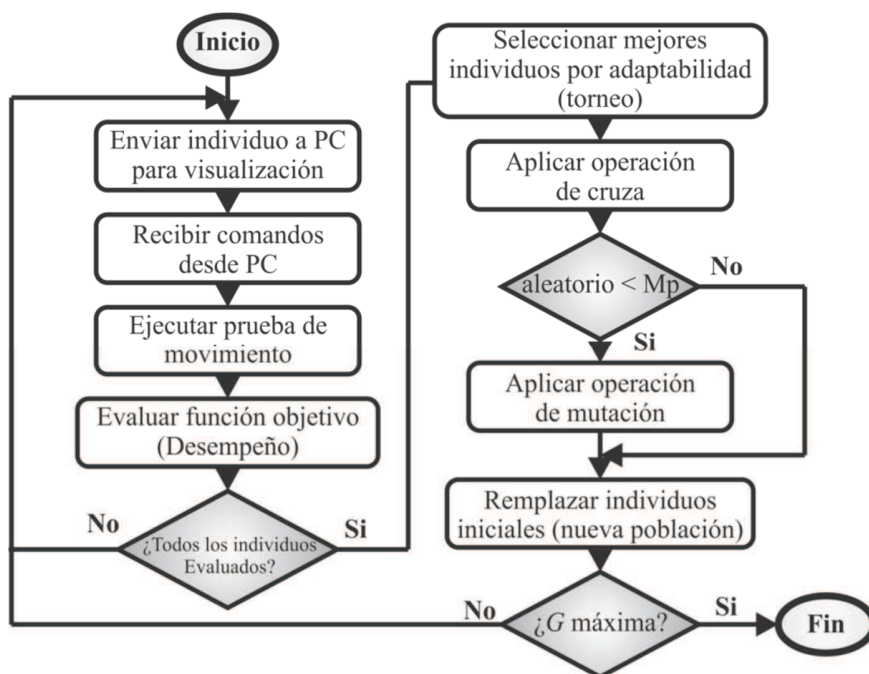


Figura 3.10. Diagrama de flujo para la implementación de la estrategia SW.

Tabla 3.2. Eventos genéticos programados para la estrategia SW.

Cruza	Mutación
<pre>function Crossover{ c_point = randnumber; head_mask = 0xFFFF << c_point; tail_mask = 0xFFFF << 16-c_point; Head1 = Individual1 & head_mask; Tail1 = Individual1 & tail_mask; Head2 = Individual2 & head_mask; Tail2 = Individual2 & tail_mask; NewIndividual1 = Head1 Tail2; NewIndividual2 = Head2 Tail1; }</pre>	<pre>function Mutation{ m_point = randnumber; m_mask = 1 << m_point; if randnumber < mutation_probability NewIndividual = NewIndividual ^ m_mask;}</pre>
Selección	Mutación adaptiva
<pre>Function Select_individuals{ for(i=0; i<3;i++) for(j=0; j<2;j++){ if Objective_value[j]>Objective_value[j+1]{ auxiliar = Objective_value[j]; Objective_value[j]=Objective_value[j+1]; Objective_value[j+1]=auxiliar; auxiliar2=Individual[j]; Individual[j]=Individual[j+1]; Individual[j+1]=auxiliar2;} } if Objective_value[0] < ObjectiveBest { ObjectiveBest = Objective_value[0]; BestIndividual = Individual[0];} }</pre>	<pre>function Adaptive_mutation{ value=MaxIterations/mutation_probability; if ActualIteration=value decrement mutation_probability;}</pre>

3.3.2 Implementación mediante la estrategia en HS

El diagrama general a bloques mostrado en la Figura 3.11 ilustra la implementación de la metodología basada en los MAG a través de la estrategia HS; se trata de un módulo en hardware independiente que funge como un co-procesador genético y ejecuta las operaciones genéticas de cruza y mutación, brindando soporte al procesador embebido. En este caso el procesador embebido solo funciona como una unidad de control de flujo de datos del proceso de optimización de acuerdo a lo que se aprecia en la Figura 3.9b. Este decir, en esta

estrategia el procesador controla la interacción del resto de los módulos del sistema de control y realiza operaciones básicas de soporte.

El módulo en hardware recibe a los individuos a ser operados, I1 e I2, entonces, un número aleatorio, RND, define los puntos de operación. El valor de la probabilidad de mutación, MP, define si la operación de mutación será ejecutada o no, y un valor de configuración, CNF, que define si la operación genética será cruce, mutación o ambas. Finalmente, la salida del módulo es un individuo modificado por operación genética llamado descendencia, OFFSPRING, y una bandera que indica el fin de la operación realizada, RDY.

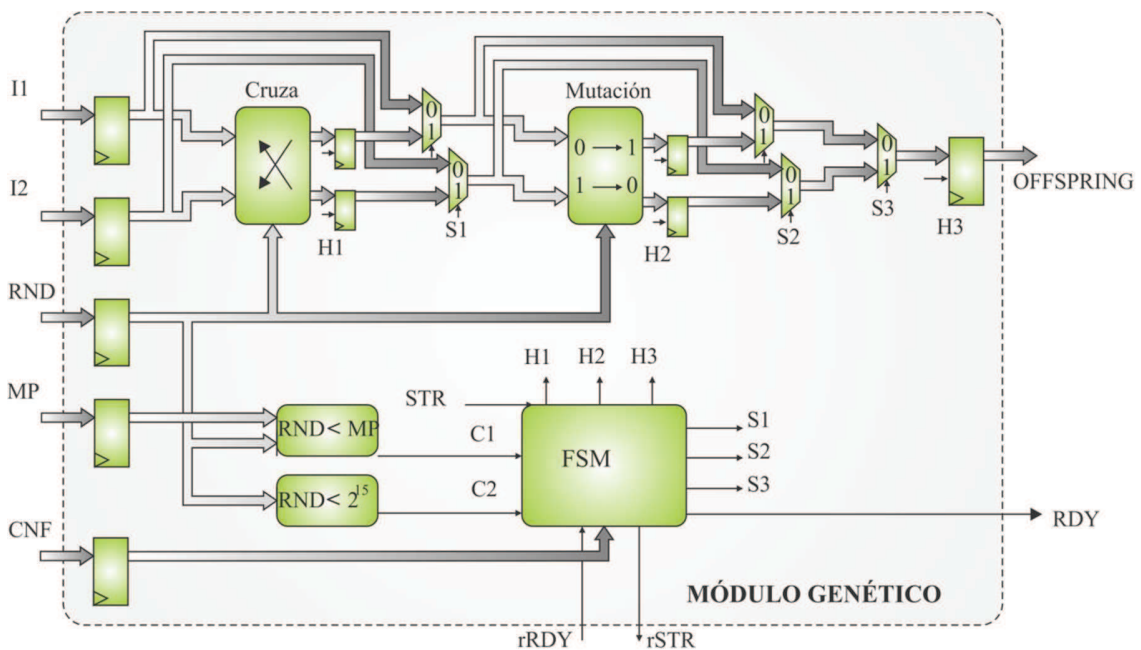


Figura 3.11. Arquitectura del módulo genético para la estrategia HS.

Debido a que todo el proceso de optimización depende de la generación de números aleatorios, un módulo generador de valores aleatorios se desarrolla

también en hardware para proporcionar dicha función. La Figura 3.12 muestra el diagrama a bloques de la arquitectura del generador de aleatorios. Se aclara que éste módulo también es utilizado en la implementación a través de la estrategia en SW. En general el funcionamiento del módulo es a través de un convertidor analógico-digital que captura la señal de una antena de ruido y la magnitud de la señal codificada en un dato binario se utiliza para la generación del valor aleatorio. Se toman como base los bits menos significativos del valor binario ya que son los que más varían. Entonces en cada lectura del convertidor se tendrá un valor distinto.

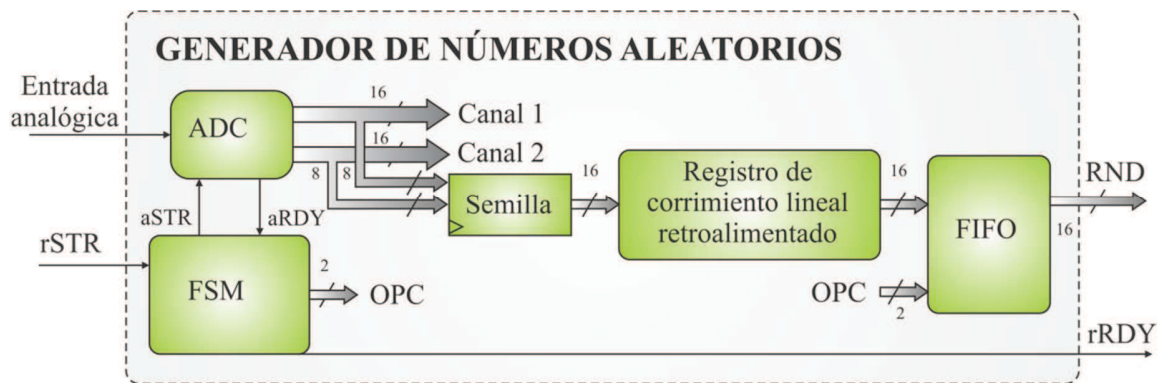


Figura 3.12. Diagrama a bloques del generador de números aleatorios.

En el capítulo siguiente se presentan los resultados de la metodología aquí descrita. Para las pruebas de la implementación en software se aplicará la metodología en la plataforma de pruebas desarrollada y en dos servosistemas adicionales.

3.4 Plataforma de pruebas, fresadora CNC

Un aspecto fundamental en el desarrollo de una metodología de optimización de controladores PID para sistemas de control de posicionamiento es

poder validar y verificar dicha metodología, no solamente vía simulaciones sino también en aplicaciones reales. Para lograr esto último es necesario contar con una plataforma de pruebas que permitirá implementar dicha metodología. Específicamente hablando, se trata de una máquina fresadora CNC de tres ejes, de la cual solo se tiene la fundición, por lo que, se requiere de realizar la instrumentación y diseñar el sistema de control de la misma, Figura 3.13.

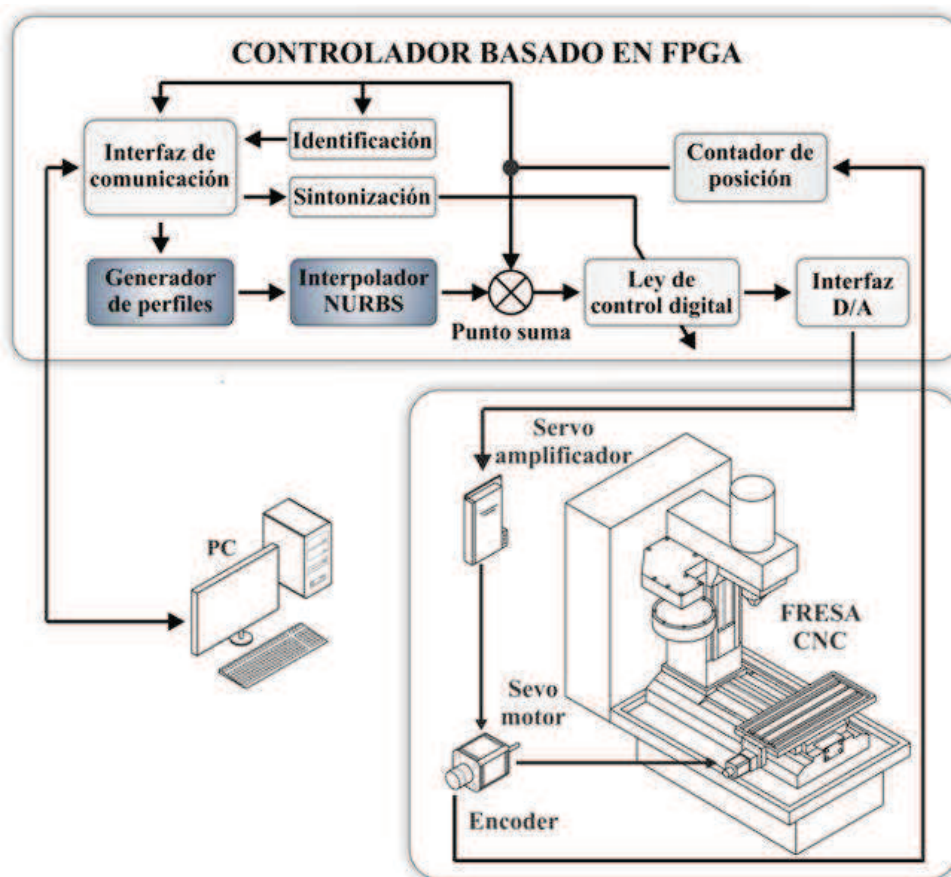


Figura 3.13. Instrumentación y sistema de control de fresadora CNC.

Como se puede apreciar la parte de la instrumentación y montaje de equipo, consiste en montar los actuadores en la máquina, básicamente el

acoplamiento de motores y su conexión, el manejo de los mismos es mediante el servoamplificador que recibe la señal del controlador digital. El estado de la máquina es retroalimentado a través de un encoder incremental óptico al controlador digital, y finalmente la visualización de varios parámetros de interés se realiza a través de una interfaz de usuario conectada al sistema digital. Por otro lado, el diseño del sistema de control consiste en una arquitectura digital en hardware que integra los módulos *IP-core* que aportan funcionamientos parciales. Por ejemplo, en la figura se aprecian: el módulo de comunicación con la PC, el algoritmo de control, el codificador de posición, los generadores de trayectorias y perfiles de movimiento, la interfaz de conversión digital-analógica, y la sintonización. También, se observa que los módulos son las arquitecturas digitales que se implementan en el dispositivo de alto desempeño, basado en un arreglo de compuertas programables en campo, FPGA (field programmable gate array), y se interconectan para lograr el lazo cerrado del sistema de control.

La Figura 3.13 muestra el procedimiento para lograr la instrumentación y control de la máquina por cada uno de sus ejes, por lo tanto, el esquema mostrado debe aplicarse a los dos ejes restantes. Los resultados de la conclusión de la máquina se muestran adelante en el siguiente capítulo.

4. EXPERIMENTACIÓN Y RESULTADOS

En capítulo presenta los resultados obtenidos por la metodología propuesta, en general se presentan los resultados de la implementación en software realizada fuera de línea, y los resultados de la implementación en hardware, realizada en línea sobre el proceso.

4.1 Resultados de la implementación en Software

En esta sección se presentan los resultados de la implementación de la metodología MAG-GMF, la cual se realizó para dos casos de estudio:

- Optimización de un modelo teórico
- Optimización de modelos obtenidos por MMC, de tres servosistemas.

4.1.1 Resultados de optimización para un modelo teórico

A fin de validar la metodología de optimización MAG-GMF propuesta para la implementación en software, se realiza en MATLAB una prueba por simulación únicamente. Los resultados se comparan contra otra variante de los AGE conocida como algoritmos genéticos de codificación real, AGR, y contra la técnica de GMF clásica. Por otro lado, el modelo teórico usado para la simulación es el simplificado por Tal (1999) para servo-sistemas mostrado en la ecuación (35).

$$G_p(s) = \frac{K_a \cdot K_t \cdot K_f}{J_m \cdot s^2} \quad (35)$$

Donde:

K_a , es la ganancia del amplificador con un valor especificado de 1 [A/A]

K_t , es el valor de la constante de torque con un valor especificado de 0.2 [N.m/A]

K_f , es la ganancia del sensor de posición con un valor especificado de 636 [cuentas/rad]

J_m , es el valor del momento de inercia con un valor especificado de $1.1 \cdot 10^{-4}$ [kg.m²]

En la Figura 4.1 se presenta el resultado de la simulación, en la cual se presenta la comparación en desempeño de las metodologías para la optimización de los parámetros del controlador.

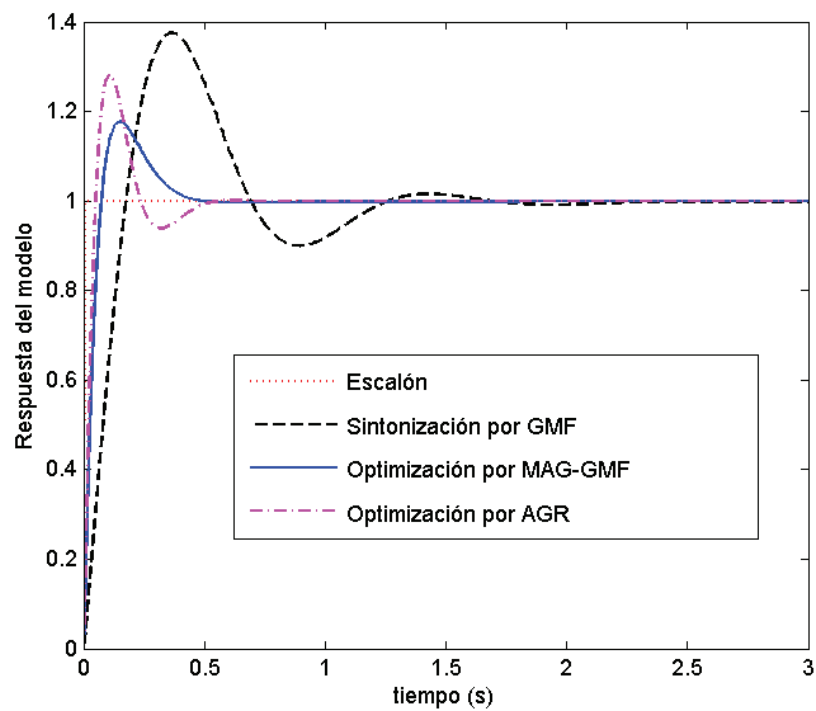


Figura 4.1. Resultados de la simulación para los métodos MAG-GMF, AGR, y GMF.

Las gráficas de la figura muestran la respuestas a una entrada escalón de del modelo matemático de la planta, ecuación (35), y del controlador en lazo cerrado sintonizado para las distintas metodologías. Se observa que la dinámica obtenida para el método clásico de sintonización es buena, aunque tiene un sobrepaso significativo y no fue la mejor, esto indica que a pesar de las restricciones consideradas por el método clásico esto no garantiza un comportamiento óptimo. En el caso de la técnica de optimización por AGR, también se obtuvo una buena dinámica, sin embargo el mejor desempeño se observa en la dinámica para la metodología de MAG-GMF, cuyo desempeño fue rápido, el más suave y con menos sobrepaso tanto positivo como negativo. Se debe especificar que los parámetros generales para la metodología de AGR fueron como los de la Tabla 3.1, a excepción que se requirieron 200 iteraciones y una población de 100 individuos.

4.1.2 Resultados de implementación en software aplicados a servosistemas

La metodología de MAG-GMF también se valida de forma experimental sobre la plataforma de pruebas descrita en la sección 3.4, y para dos servosistemas adicionales como se verá a continuación.

➤ *Configuración del experimento*

La implementación del método se realizó usando MATLAB en la PC en funciones. Tres servosistemas, que se muestran en la Figura 4.2, se utilizaron para las pruebas experimentales con las siguientes características. Para la máquina fresadora las pruebas se realizaron sobre el eje z, el cual utiliza un servomotor brushless 3845-ME8137 de MCG con un encoder incremental óptico acoplado al eje de 4000 cuentas por vuelta. El servo-amplificador funciona por

modulación de ancho de pulso, PWM, de la marca Advanced Motion Controls. Los dos servosistemas adicionales son un torno CNC de dos ejes, para el cual se utiliza un servomotor 705-0-101 marca EZG con un encoder incremental óptico de 400 cuentas por vuelta, manejado por un servo-amplificador 413 marca Copley Control Corp. El eje de prueba del torno fue el eje y. El otro servosistema es un robot de brazo manipulador universal programable, PUMA, de seis grados de libertad donde el eje de prueba es el primer eslabón. Cuenta con un servomotor estándar, un encoder acoplado de 4000 cuentas por vuelta, y un servo-amplificador 403 marca Copley Control Corp. Se utiliza la misma tarjeta de control, en pruebas independientes, para todos los servosistemas. La tarjeta es un diseño propietario realizado por miembros del grupo de investigación HSPdigital de la UAQ, y está basada en el dispositivo de alto desempeño FPGA. El dispositivo incluye los módulos de propiedad intelectual en hardware para la ley de control PID y todos los módulos complementarios necesarios para formar el lazo cerrado con los servosistemas. Los modelos de la planta se obtienen por MMC y se utilizan para aplicar las metodologías de GMF, AGR, y MAG-GMF. Las ganancias obtenidas se validan a través de pruebas de movimiento en los sistemas físicos al establecerlas dentro de la tarjeta de control. Para cada conjunto de ganancias de cada metodología se realizaron diez pruebas de movimiento, y cada movimiento consistió de seis referencias escalón. Para los casos de la fresadora y el torno CNC las referencias son 1000, 4000, 10000, 4000, 1000, y 0 cuentas para un seguimiento total de 20000 cuentas. En el caso del robot PUMA las referencias son 100, 400, 1000, 400, 100, y 0, para un seguimiento total de 2000 cuentas.

Las funciones de transferencia obtenidas para la máquina fresadora CNC, el torno CNC, y el robot PUMA industrial se presentan en las ecuaciones (36) a (38), respectivamente.

$$G_{p_fresa}(z) = \frac{2.98e - 005z^2 + 0.000306z - 0.000233}{z^2 - 2z + 0.999} \quad (36)$$

$$G_{p_torno}(z) = \frac{2.091e - 007z^2 + 3.293e - 007z - 4.06e - 005}{z^2 - 1.78z + 0.78} \quad (37)$$

$$G_{p_puma}(z) = \frac{5.95e - 005z^2 - 0.000104z + 6.98e - 05}{z^2 - 1.96z + 0.956} \quad (38)$$

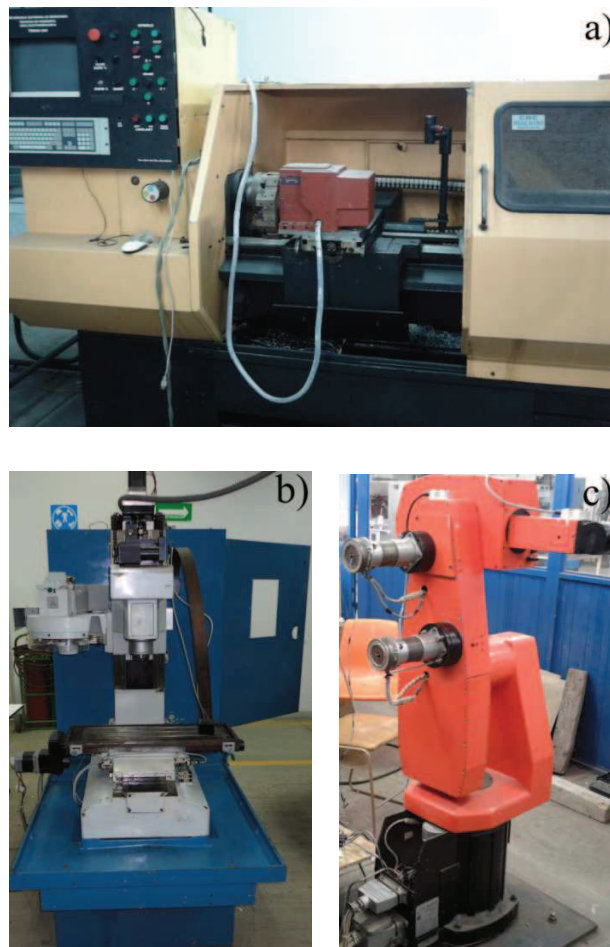


Figura 4.2. Servosistemas usados, a) torno CNC, b) fresadora CNC, y c) robot PUMA.

En la Tabla 4.1 se presentan los parámetros optimizados proporcionados por la metodología MAG-GMF y los parámetros propuestos para la técnica de

sintonización clásica de GMF. Así mismo, se presentan las ganancias del controlador derivadas de estos parámetros. Para el caso de la técnica basada en los AGR, se aclara que esta técnica afecta directamente a las ganancias del controlador.

Tabla 4.1. Parámetros optimizados y ganancias del controlador obtenidas

	Método clásico GMF	Optimización MAG-GMF	Optimización clásica AGR	Servosistema
τ	0.2	0.2310	---	Fresadora CNC de tres ejes
θ_m	45	46.3750	---	
K_p	19.0010	14.2863	49.5412	
K_i	58.0277	33.9929	19.2312	
K_d	0.5052	0.5365	0.9600	
τ	0.2	0.4011	---	Torno CNC de dos ejes
θ_m	45	40.0073	---	
K_p	40.8838	29.6787	97.0836	
K_i	112.3706	40.6772	466.4310	
K_d	0.5439	0.9632	4.2785	
τ	0.25	0.3538	---	Robot PUMA industrial
θ_m	45	32.5024	---	
K_p	148.4444	152.7212	99.5051	
K_i	326.4040	237.2675	315.9438	
K_d	4.5883	4.6012	4.9393	

Las pruebas de movimiento en los servosistemas usaron las ganancias obtenidas a través de las técnicas de optimización y la técnica clásica de la Tabla 4.1. Por lo tanto, se obtuvo el promedio de los desempeños de las pruebas considerando como criterios los valores del sobrepaso en porcentaje, %SP, la integral del error cuadrático, *IEC*, y el error cuadrático medio, *ECM*. En la Tabla 4.2 se muestra el resumen del desempeño.

La Tabla 4.2 muestra que el desempeño de la metodología de MAG-GMF propuesta proporciona excelentes resultados cuando se le compara contra las

otras dos metodologías. Ya que proporciona el menor porcentaje de sobrepaso en todos los casos, cuestión que implica movimientos suaves y con mayor exactitud. En cuanto a los valores de la integral del error cuadrático se observan valores similares entre las dos técnicas de optimización, que son mejores en comparación con los valores de la técnica clásica. Finalmente, en general la técnica aquí propuesta presenta menor valor para el error cuadrático medio que sus contrapartes.

Tabla 4.2. Desempeño en las pruebas de movimiento para cada servosistema

Metodología	%SP	IEC (cuentas)	ECM (%)	Servosistema
<i>MAG-GMF</i>	5.65	3.7436×10^9	0.6649	Fresadora CNC de tres ejes
<i>AGR</i>	29.26	3.6888×10^9	0.6553	
<i>GMF</i>	13.0	$4.4.3736 \times 10^9$	0.7690	
<i>MAG-GMF</i>	6.9	3.7470×10^9	0.6660	Torno CNC de dos ejes
<i>AGR</i>	19.28	4.1936×10^9	0.7452	
<i>GMF</i>	27.3	4.9102×10^9	0.8729	
<i>MAG-GMF</i>	17	4.5668×10^7	0.0081	Robot PUMA industrial
<i>AGR</i>	18.16	4.7432×10^7	0.0084	
<i>GMF</i>	37.16	5.0649×10^7	0.0090	

Por otra parte la Figura 4.3 muestra las gráficas obtenidas de las pruebas de movimiento. En las gráficas de la figura se observa que el inciso a) consiste en la respuesta obtenida del torno para seis referencias escalón, el inciso b) la respuesta de la máquina fresadora para las mismas seis referencias, y el inciso c) la respuesta del PUMA. Los acercamientos de las respuestas mostradas en los incisos d), e), y f) corresponden directamente al torno, fresadora y robot PUMA, en estos acercamientos se observa claramente que los sobrepasos más grandes corresponden siempre a la técnica clásica de GMF y para el caso de la metodología de AGR se tiene un menor sobrepaso alcanzando rápidamente la referencia pero esto se traduce en un movimiento brusco de la máquina, lo cual afecta en producto final. Finalmente, la metodología por MAG-GMF proporciona los mejores resultados no solamente al tener el sobrepaso más bajo en todos los

casos, sino que también, genera un movimiento suave y alcanzando su referencia en un tiempo similar al de la metodología por AGR.

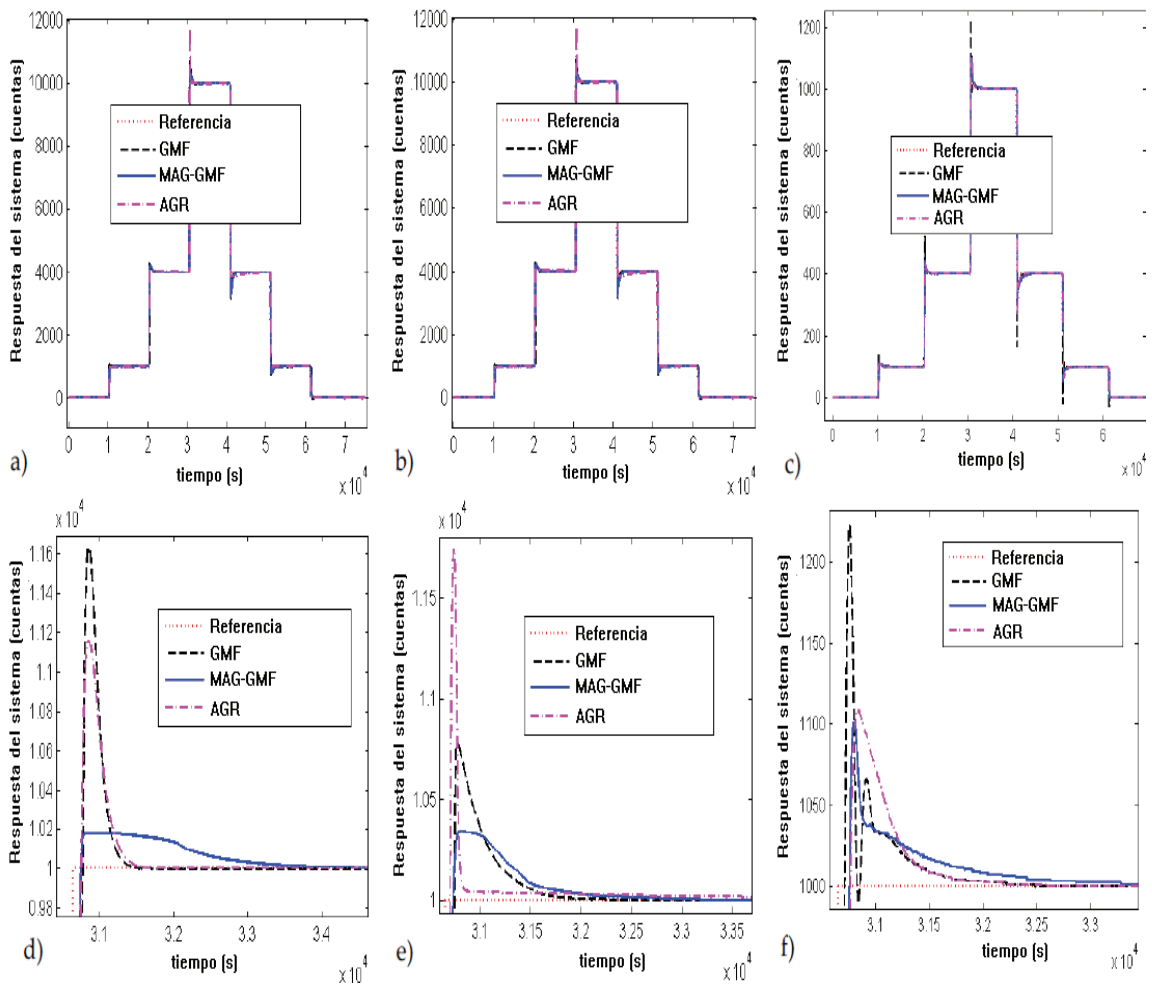


Figura 4.3. Respuesta de los sistemas a) torno, b) fresadora, y c) PUMA, y acercamiento del escalón más grande para d) torno, e) fresadora, y f) PUMA.

Se puede afirmar en base a los resultados de las pruebas en los servosistemas que la metodología de optimización de los parámetros del controlador mejora notoriamente respecto a la técnica clásica.

4.2 Conclusión de plataforma de pruebas

El montaje de actuadores, sensores y sistema de control al sistema de posicionamiento, máquina fresadora CNC de tres ejes, dio como resultado una plataforma de pruebas en las que se validó la metodología de optimización basada en los MAG. La Figura 4.4 muestra la plataforma de pruebas mencionada.

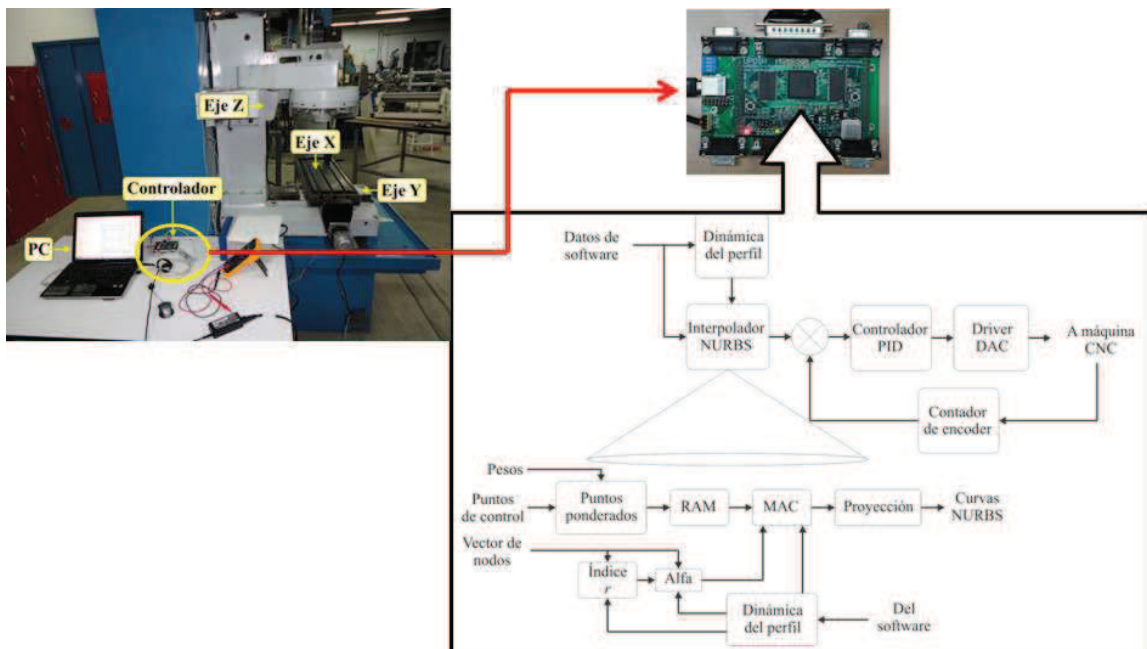


Figura 4.4. Plataforma de pruebas para la metodología propuesta.

En el diagrama se pueden observar los tres bloques descritos en la Figura 3.7, pues se observa el sistema físico, que es la máquina fresadora con sensores y actuadores, se observa la tarjeta de control, en la cual se encuentran los módulos para integrar un sistema en lazo cerrado. Finalmente, la interfaz de usuario desarrollada en código C en la PC.

4.3 Resultados del desarrollo de arquitecturas para implementación en hardware

La implementación en hardware se realizó en dos estrategias; SW y HS, descritas en el capítulo anterior, para las cuales los resultados se presentan a continuación.

➤ *Configuración del experimento*

Las pruebas experimentales de las dos estrategias se llevan a cabo utilizando la tarjeta propietaria basada en FPGA. Dentro de la tarjeta se encuentra el módulo de procesador embebido operando a 24 MHz, también se encuentran todos los módulos necesarios para integrar el sistema para implementar la metodología a través de las dos estrategias. La interfaz de usuario se desarrolla bajo lenguaje C++ con librerías abiertas en una Laptop con sistema operativo de 64 bits, 16 GB de RAM, un procesador Intel Core i7-6312QM operando a 2.1 GHz. De la máquina fresadora el eje para las pruebas seleccionado en esta ocasión es el eje x, en el cual se tiene un servomotor brushless 3485-ME8137 marca MCG con un encoder incremental óptico acoplado de 4000 cuentas por vuelta. El servomotor es manejado por un servo-amplificador basado en modulación de ancho de pulso, PWM, de la marca Advanced Motion Control. Las pruebas de movimiento para el proceso de optimización consisten en una referencia de movimiento de 200 cuentas, se tomarán 300 muestras adquiridas a una frecuencia de 1kHz. La prueba de movimiento para evaluar el desempeño de la metodología utiliza un perfil trapezoidal para una referencia de 1000 cuentas adelante y 1000 cuentas hacia atrás. Los valores especificados del perfil se establecen en un máximo de 25600 cuentas/s para la velocidad máxima, y +- 256000 cuentas/s², para la aceleración y deceleración respectivamente. Por último, se realizan 40

pruebas para cada estrategia. En la Figura 4.5 se muestra el sistema con la plataforma de pruebas y el eje de pruebas seleccionado.

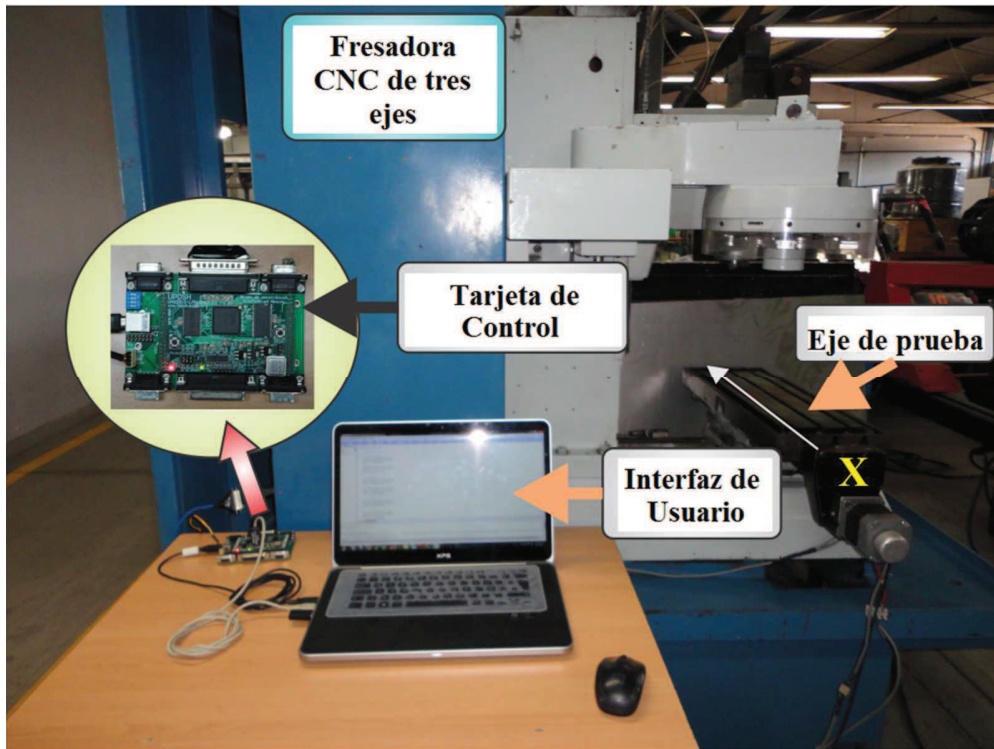


Figura 4.5. Configuración para la validación de las pruebas en la plataforma desarrollada.

Se debe remarcar que en las implementaciones de las estrategias SW y HS, las variables de optimización, por lo que la Tabla 4.3 muestra los rangos de diseño para cada variable.

Tabla 4.3. Rango de diseño para las ganancias del controlador.

Variable de diseño	Rango de diseño
Ganancia proporcional, K_p	[1, 5000]
Ganancia integral, K_i	[0.1, 50000]
Ganancia derivativa, K_d	[0.1, 50]

Los tiempos de cómputo requeridos para cada estrategia implementada se muestran en la Tabla 4.4, en ella solamente se consideran los eventos relacionados al proceso genético. La estrategia HS requiere menos tiempo de cómputo debido a que los operadores genéticos se ejecutan por hardware en un módulo independiente que da soporte al procesador embebido, mientras que en la estrategia SW estos operadores se ejecutan dentro del procesador.

Tabla 4.4. Tiempo de cómputo requerido por las estrategias implementadas.

Tiempo de cómputo	SW	HS
Criterios de desempeño	2.86 ms	2.86 ms
Selección	≈ 47.763 μs	≈ 47.763 μs
Cruza	168.0 μs	160.0 ns
Cruza de todos los individuos *	1.0 ms	960.0 ns
Mutación	453.3 μs	480.0 ns
Mutación de todos los individuos *	2.72 ms	2.88 μs
Individuo aleatorio	300.0 ns	160.0 ns
Individuos aleatorios ^a	900.0 μs	480.0 ns
Probabilidad de mutación adaptiva	12.7 μs	12.7 μs
Tiempo total requerido	≈ 6.6413 ms	≈ 2.9247 ms

* Tiempo consumido en la ejecución de las operaciones genéticas para todos los individuos por generación en el proceso de optimización. ^a Tiempo consumido en la generación de todos los individuos reemplazados aleatoriamente.

En la Figura 4.6 se muestra la evolución de los individuos hasta su convergencia nominal. Se observa que la convergencia nominal de ambos individuos es similar debido a que se trata de la implementación del mismo esquema pero mediante dos estrategias distintas.

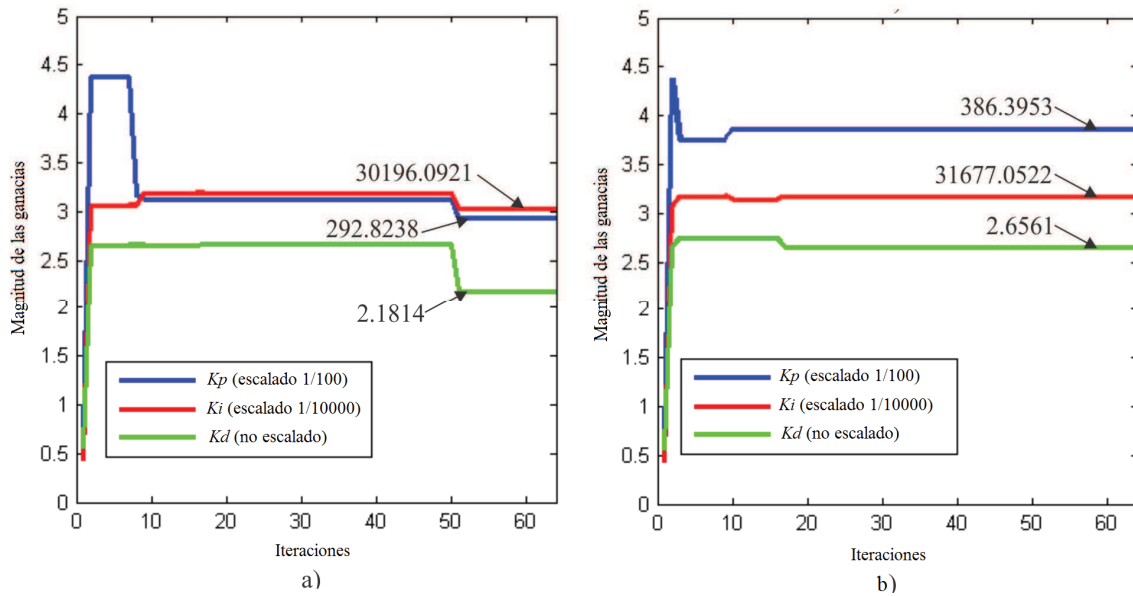


Figura 4.6. Evolución de las ganancias para estrategia a) SW, y b) HS.

Finalmente el desempeño se observa en base al error obtenido para cada estrategia, por lo cual se ejecutan las pruebas de movimiento usando el perfil descrito en la configuración del experimento. Las gráficas de la Figura 4.7 muestran el seguimiento y error de seguimiento para la estrategia en SW, incisos a) y c), y para la estrategia en HS, incisos b) y d).

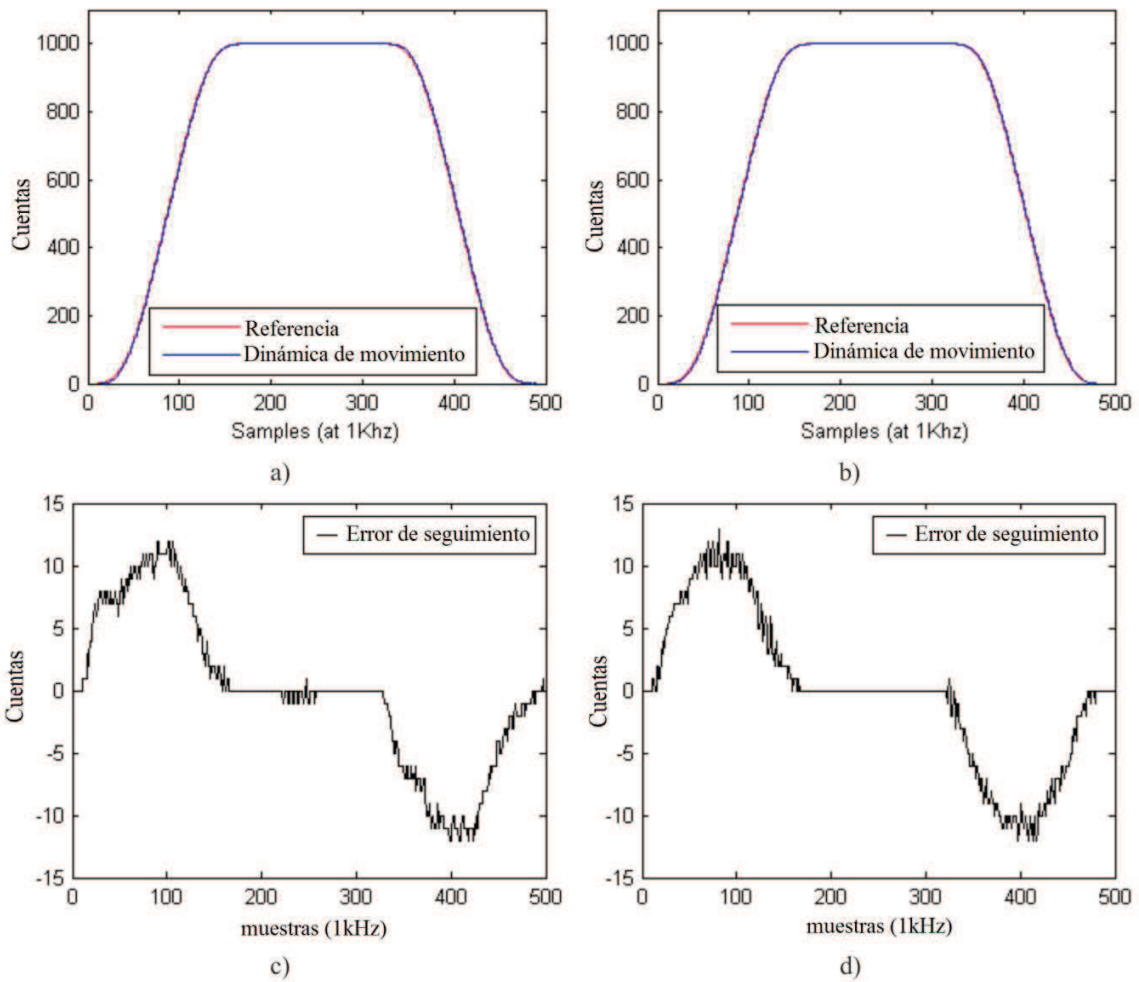


Figura 4.7. Dinámica de movimiento, a) y b), y error de seguimiento, c) y d), para las estrategias SW y HS respectivamente.

De la figura se puede observar que el error de seguimiento es notoriamente disminuido, en la parte central de las gráficas de error se aprecia que hay una pequeña variación para la estrategia SW, mientras que para la estrategia HS se tiene error en estado estacionario cero cuando el sistema alcanza la referencia.

4.3.1 Recursos computacionales de la implementación en hardware

La implementación de la metodología basada en los MAG implicó además del desarrollo de las arquitecturas de las Figuras 3.11 y 3.12 la integración del sistema de control embebido. La síntesis del sistema, como se muestra en la Figura 3.8, se realizó como se explica a continuación.

El dispositivo FPGA utilizado para la síntesis, que es el dispositivo en el que está basada la tarjeta de control propietaria usada en este trabajo, es un Spartan 3E – 1600 de Xilinx. La tarjeta utiliza un oscilador de 48 MHz y ya incorpora todos los periféricos necesarios para el control de CNC, por ejemplo convertidores ADC y DAC. Entonces, la síntesis del código VHDL de la metodología de optimización en el FPGA se realizó para una optimización en tiempo y sin mantener la jerarquía de los módulos en la síntesis. Los resultados de la síntesis se muestran en la Tabla 4.5.

Tabla 4.5. Recursos requeridos por el sistema embebido.

Recursos	Utilizados	Porcentaje equivalente
Flip-flops	6340/29504	21%
LUTs de 4 entradas	10322/29504	34%
RAM de 16 bits	1/36	2%
Multiplicadores de 18x18	6/36	16%
Slices		
De lógica relacionada	5453/5453	100%
De lógica no relacionada	0/5453	0%

Se debe aclarar que el sistema embebido integra los módulos siguientes.

- Respecto al sistema de control: los módulos del algoritmo de control, el contador de posición de encoder, los drives de conversión ADC y DAC, temporizadores.
- Respecto al procesador embebido: el driver de la memoria RAM, el manejador de interrupciones, el módulo de operaciones de punto flotante, el depurador de código, etc.
- Respecto a la metodología basada en MAG: el módulo de operaciones micro-genéticas, el módulo generador de números aleatorios.
- Otro bloques requeridos: el módulo de comunicación con la PC por USB.

Así mismo, los recursos requeridos por la arquitectura de operaciones micro-genéticas y del módulo generador de números aleatorios se presentan en la tabla 4.6.

Tabla 4.6. Recursos requeridos por el co-procesador genético.

Recursos	Utilizados	Porcentaje equivalente
Slices	304/14752	2%
Slice Flip-Flop	401/29504	1%
LUTs de 4 entradas	478/29504	1%

5. CONCLUSIONES Y PROSPECTIVAS

5.1 Conclusiones

El correcto funcionamiento de un sistema de posicionamiento industrial depende de cada uno de los elementos que lo compone, por ejemplo en este trabajo se analiza el módulo de sintonización. El sistema instrumentado aquí presentado implementa perfiles de movimiento obtenidos por interpolación de curvas NURBS que permiten al sistema realizar movimientos suaves y continuos, aun así para que la máquina tenga una buena dinámica, el algoritmo de control debe ser ajustando en parámetros de forma óptima. El controlador PID es el más utilizado en la industria y en los sistemas de posicionamiento juega un rol esencial ya que estos sistemas son condicionalmente estables por lo que su correcto diseño (o sintonización) afectan significativamente en el desempeño del sistema. Los métodos clásicos para la sintonización de controladores PID aún son ampliamente utilizados por su practicidad. Sin embargo, estos métodos aunque proveen buenos resultados (en ocasiones), distan de ser óptimos.

El uso de técnicas meta-heurísticas en el problema de optimización tiene ventajas sobre sus contrapartes clásicas al tener un concepto que no requiere matemática compleja y al dar solución efectiva a problemas que presentan características no lineales, o problemas que se presentan en espacios de diseño no convexos, incluso a problemas de optimización multi-objetivo. Las características de los Algoritmos Genéticos generalmente permiten que la implementación de esta técnica se desarrolle en software debido a la gran cantidad de recursos computacionales que demanda, por lo que implementaciones en tiempo real o en línea son una tarea compleja que continua en estudio. Usando el concepto de micro-población y definiendo las operaciones genéticas adecuadas

se reducen significativamente la cantidad de recursos computacionales demandados en la implementación de los AG estándar.

Al definir una arquitectura en hardware implementada en tecnología de alto desempeño como lo es el FPGA para el cómputo de los operadores genéticos, como una unidad de co-procesamiento genético, se optimizan los recursos utilizados y permite al sistema operar en línea. Adicionalmente el sistema tiene características inherentes tales como configurabilidad, portabilidad y procesamiento paralelo. En este proyecto de investigación se desarrolló una metodología basada en Algoritmos Genéticos adecuada a la optimización en línea de los parámetros de controladores PID usados en sistemas de posicionamiento, cuestión de interés para la industria ya que se proporciona una sintonización efectiva que deriva en una mejora del comportamiento del sistema superior a la generada por el uso de técnicas clásicas.

5.2 Prospectivas

Quedan como prospectivas para esta investigación que la metodología basada en MAG se extienda en funcionalidad, no solamente optimizando los parámetros del controlador, sino también otros módulos integrados en un sistema de control de posición. Por ejemplo en el módulo de identificación de plantas, aunque al final la metodología ajusta directamente los parámetros del controlador, es interesante observar como muchos métodos clásicos de sintonización dependen de la exactitud de un modelo matemático de la planta, por ello sería interesante adecuar la metodología a este tema.

Es interesante que los resultados obtenidos en la investigación sean valores óptimos locales que en general para la aplicación estudiada fueron bastante buenos en cuanto al desempeño, pero sería interesante investigar la manera en que los valores óptimos globales pueden obtenerse. Incluso, sería interesante desarrollar más arquitecturas que permitan implementar otras técnicas meta-heurísticas para hacer un estudio comparativo, ya que en general cada técnica tiene características distintas que podría hacerlas más convenientes en ciertos problemas de optimización que otras.

Por otra parte, sería interesante extender la aplicación de esta metodología a otras áreas de investigación, ya que en general es una metodología que puede ser con unas pocas consideraciones aplicada a otros problemas de optimización de otras áreas. Por ejemplo, para el análisis de señales en calidad de la energía. Lo anterior a través de adecuar los MAG e integrarlos con otras metodologías, ya sea como algoritmo de soporte o como metodología principal. Otra aplicación podría ser en análisis de vibraciones.

6. REFERENCIAS

- Aguado Behar, Alberto. 2000. Temas de Identificación y Control Adaptable. Instituto de Cibernética, Física y Matemática. La Habana, Cuba.
- Ahmad, Zaryab., Rahmani, Keyvan., D'Souza, Roshan M. 2010. Applications of Genetic Algorithms in Process Planning: Tool Sequence Selection for 2.5-Axis Pocket Machining. *Journal of Intelligent Manufacturing*. Vol. 21 (2010). 461 – 470.
- Coelho, Leandro dos Santos., Pessôa, Marcelo Wichhoff. 2011. A Tuning Strategy for Multivariable PI and PID Controllers Using Differential Evolution Combined with Chaotic Zaslavskii Map. *Expert Systems with Applications*. doi:10.1016/j.eswa.2011.04.156.
- De Santiago-Perez, J. Jesús., Osornio-Ríos, Roque Alfredo., Romero-Troncoso, René de Jesús., Herrera-Ruíz, Gilberto., Delgado-Rosas, Manuel. 2008. DSP Algorithm for the Extraction of Dynamics Parameters in CNC Machine Tool Servomechanisms from an Optical Incremental Encoder. *International Journal of Machine Tools & Manufacture*. Vol. 48 (2008). 1318 – 1334.
- Erkorkmaz, K., Heng, M. 2008. A Heuristic Feedrate Optimization Strategy for NURBS Toolpaths. *CIRP Annals – Manufacturing Technology*. Vol. 57 (2008). 407 – 410.
- Figuera Vinúe, Pau. 2006. Optimización de Productos y Procesos Industriales. Ediciones Gestión. Barcelona. ISBN 84-96426-63-7.
- Gasparetto, A., Zanotto, V. 2010. Optimal Trajectory Planning for Industrial Robots. *Advances in Engineering Software*. Vol. 41 (2010). 548 – 556.
- He, Wei., Lei, Ming., Bin, Hongzan. 2009. Iso-Parametric CNC Tool Path Optimization Based on Adaptive Grid Generation. *International Journal of Advanced Manufacturing Technology*. Vol. 41 (2009). 538 – 548.
- Held, Martin., Spielberger, Christian. 2009. A Smooth Spiral Tool Path for High Speed Machining of 2D Pockets. *Computed-Aided Design*. Vol. 41 (2009). 539 – 550.
- Huang, Yan-Wen., Tung, Pi-Cheng., Wu, Cheng-Yu. 2007. Tuning PID Control of an Automatic Arc Welding System Using a SMAW Process. *International Journal of Advanced Manufacturing Technology*. Vol. 34 (2007). 56 - 61.
- Inoue, Y., Yamada, K., Morimoto, S., Sanada, M. 2009. Effectiveness of Voltage Error Compensation and Parameter Identification for Model-Based Sensorless

- Control of IPMSM. IEEE Transactions on Industry Applications. Vol. 45 (2009). 213 - 221.
- Kamalasadan, S., Ghandakly, A.A. 2007. A Neural Network Parallel Adaptive Controller for Dynamic System Control. IEEE Transactions on Instrumentation and Measurement. Vol. 56 (2007). 1786 - 1796.
- Kim, Hyun-Chul. 2010. Tool Path Generation for Contour Parallel Milling with Incomplete Mesh Model. International Journal of Advanced Manufacturing Technology. Vol. 48 (2010). 443 - 454.
- Kim, Dong Hwa. 2011. Hybrid GA-BF Based Intelligent PID Controller Tuning for AVR System. Applied Soft Computing. Vol. 11 (2011). 11 – 22.
- Kiyong, Kim., Rao, P., Burnworth, J.A. 2010. Self-Tuning of the PID Controller for a Digital Excitation Control System. IEEE Transactions on Industry Applications. Vol. 46 (2010). 1518 - 1524.
- Lequin, Oliver., Gevers, Michel., Mossberg, Magnus., Bosmans, Emmanuel., Triest, Lionel. 2003. Iterative Feedback Tuning of PID Parameters: Comparison with Classical Tuning Rules. Control Engineering Practice. Vol. 11 (2003). 1023 – 1033.
- Li, Y., Ang, K.H., Chong, G.C.Y. 2006. Patents, Software and Hardware for PID Control: An Overview and Analysis of the Current Art. IEEE Control Systems Magazine. Vol. 26 (2006). 42 – 54.
- Masiala, M., Vafakhah, B., Salmon, J., Knight, A.M. 2008. Fuzzy Self-Tuning Speed Control of an Indirect Field-Oriented Control Induction Motor Drive. IEEE Transactions on Industry Applications. Vol. 44 (2008). 1732 - 1740.
- Mattmüller Jan, Gisler Damian. 2009. Calculating a Near Time-Optimal Jerk-Constrained Trajectory Along a Specified Smooth Path. International Journal of Advanced Manufacturing Technology. Vol. 45 (2009). 1007 - 1016.
- Milosawlewitsch Aliaga, Mirko. 2010. Método de Resintonización de Controladores PID Mediante Identificación en Lazo Cerrado. Tesis de Maestría. Universidad Autónoma de Querétaro. Facultad de Ingeniería. San Juan del Río, Qro.
- Morales-Velázquez, Luis., Romero-Troncoso, René de Jesús., Osornio-Ríos, Roque Alfredo., Herrera-Ruiz, Gilberto., de Santiago-Perez, J. Jesus. 2010. Special Purpose Processor for Parameter Identification of CNC Second Order Servo Systems on a Low-Cost FPGA Platform. Mechatronics. Vol. 20 (2010). 265 – 272.
- Morales-Velázquez, Luis., Romero-Troncoso, René de Jesús., Osornio-Ríos, Roque Alfredo., Herrera-Ruiz, Gilberto., Cabal-Yepez, Eduardo. 2010. Open-

- Architecture System Based on a Reconfigurable Hardware-Software Multi-Agent Platform for CNC Machines. *Journal of Systems Architecture*. Vol. 56 (2010). 407 – 418.
- Musharavati, F., Hamouda, A.S.M. 2011. Modified Genetic Algorithms for Manufacturing Process Planning in Multiple Parts Manufacturing Lines. *Expert Systems with Applications*. Vol. 38 (2011). 10770 – 10779.
- Ogata, Katsuhiko. 1998. *Ingeniería de control moderna*. 3er edición, Pearson education, Prentice Hall. ISBN 970-17-0048-1
- Osornio R, Roque A. 2007. *Diseño de Sistema de Control para CNC de Alta Velocidad*. Tesis Doctoral. Universidad Autónoma de Querétaro. Facultad de Ingeniería. Campus San Juan del Río, Qro.
- Osornio-Ríos, Roque Alfredo. Romero-Troncoso, René de Jesús., Herrera-Ruíz, Gilberto., Castañeda-Miranda, Rodrigo. 2007. Computationally Efficient Parametric Analysis of Discrete-Time Polynomial Based Acceleration-Deceleration Profile Generation for Industrial Robotics and CNC Machinery. *Mechatronics*. Vol. 17 (2007). 511 – 523.
- Osornio-Ríos, Roque Alfredo., Romero-Troncoso, René de Jesús., Herrera-Ruíz, Gilberto., Castañeda-Miranda, Rodrigo. 2008. The Application of Reconfigurable Logic to High Speed CNC Milling Machines Controllers. *Control Engineering Practice*. Vol. 16 (2008). 674 – 684.
- Osornio-Ríos, Roque Alfredo. Romero-Troncoso, René de Jesús., Herrera-Ruíz, Gilberto., Castañeda-Miranda, Rodrigo. 2009. FPGA Implementation of Higher Degree Polynomial Acceleration Profiles for Peak Jerk Reduction in Servomotors. *Robotics and Computer-Integrated Manufacturing*. Vol. 25 (2009). 379 – 392.
- Osornio-Ríos, Roque A., Romero-Troncoso, Rene de J., Morales-Velazquez, Luis., de Santiago-Perez, J. Jesus., Rivera-Guillen, Rooney de J., Rangel-Maldonado, J. de Jesus. 2008. A Real-Time FPGA Based Platform for Applications in Mechatronics. *International Conference on Reconfigurable Computing and FPGAs*.
- Pan, Indranil., Das, Saptarshi., Gupta, Amitava. 2011. Tuning of an Optimal Fuzzy PID Controller with Stochastic Algorithms for Networked Control Systems with Random Time Delay. *ISA Transactions*. Vol. 50 (2011). 28 – 36.
- Pessoles Xavier., Landon Yann., Rubio Walter. 2010. Kinematic Modeling of a 3-Axis NC Machine Tool in Linear and Circular Interpolation. *International Journal of Advanced Manufacturing Technology*. Vol. 47 (2010). 639 – 655.

- Sharkawy, Abdel Badie. 2010. Genetic Fuzzy Self-Tuning PID Controllers for Antilock Braking Systems. *Engineering Applications of Artificial Intelligence*. Vol. 23 (2010). 1041 – 1052.
- Rangel-Magdaleno, de Jesus J., Romero-Troncoso, de Jesús R., Osornio-Ríos, R.A., Cabal-Yepez, E., Dominguez-Gonzalez, A. 2010. FPGA-Based Vibration Analyzer for Continuous CNC Machinery Monitoring With FFT-DWT Signal Processing. *IEEE Transactions on Instrumentation and Measurement*. Vol. 59. (2010). 3184 – 3194.
- Rao, Singiresu S. 2009. *Engineering Optimization: Theory and Practice*. Cuarta Edición. John Wiley & Sons, Inc. ISBN. 978-0-470-18352-6.
- Renner, Gábor., Ekárt, Anikó. 2003. Genetic Algorithms in Computer Aided Design. *Computer-Aided Design*. Vol. 35 (2003). 709 – 726.
- Rivera-Guillen, Jesús Rooney., Romero-Troncoso, René de Jesús., Osornio-Ríos, Roque Alfredo., García-Pérez, Arturo., Herrera-Ruiz, Gilberto. 2010. Design Methodology for Fully Dynamic-Controlled Polynomial Profiles and Reduced Tracking Error in CNC Machines. *International Journal of Advanced Manufacturing Technology*. Vol. 51 (2010). 723 - 737.
- Tal, J., *Step by Step Design of Motion Control Systems*, (1994) Galil Motion Control, Inc.
- Valarmathi, K., Devaraj, D., Radhakrishnan, T.K. 2009. Real-Coded Genetic Algorithm for System Identification and Controller Tuning. *Applied Mathematical Modelling*. Vol. 33 (2009). 3392 – 3401.
- Van der Lee, J.H., Svrcek, W.Y., Young, B.R. 2008. A Tuning Algorithm for Model Predictive Controllers Based on Genetic Algorithms and Fuzzy Decision Making. *ISA Transactions*. Vol. 47 (2008). 53 – 59.
- Wang, Lin., Cao, Jianfu F., Li, Yuqiang Q. 2010. Speed Optimization Control Method of Smooth Motion for High-Speed CNC Machine Tools. *International Journal of Advanced Manufacturing Technology*. Vol. 49 (2010). 313 – 325.
- West Instruments. 2003. 1/4, 1/8 and 1/16 DIN Controllers, Indicators & Profiles User Guide. Manual Part Number 59305-2, Issue 2.
- Xiaoli, Li. 2005. Development of Current Sensor for Cutting Force Measurement in Turning. *IEEE Transactions on Instrumentation and Measurement*. Vol. 54 (2005). 289 - 296.

- Yau, Hong Tzong., Lin, Ming-Tzong., Tsai, Meng-Shiun. 2006. Real-Time NURBS Interpolation Using FPGA for High Speed Motion Control. *Computer-Aided Design*. Vol. 38 (2006). 1123 – 1133.
- Zhang Jinhua., Zhuang Jian., Du Haifeng., Wang Sun'an. 2009. Self-Organizing Genetic Algorithm Based Tuning of PID Controllers. *Information Sciences*. Vol. 179 (2009). 1007 – 10018.
- Zhang, Julie Z., Chen, Joseph C. 2008. Tool Condition Monitoring in an End-Milling Operation Based on the Vibration Signal Collected Through a Microcontroller-Based Data Acquisition System. *International Journal of Advanced Manufacturing Technology*. Vol. 39 (2008). 118 - 128.

7. ANEXOS

7.1 Artículos publicados en revistas JCR

Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering

<http://pii.sagepub.com/>

A hardware–software system for coordinated multi-axis control based on a non uniform rational B-splines interpolator applied to industrial computer numerically controlled machines

Arturo Y Jaen-Cuellar, Jesus J De Santiago-Perez, Rene J Romero-Troncoso, Luis Morales-Velazquez, Juan P Benitez-Rangel and Roque A Osornio-Rios

Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering published online 10 January 2012

DOI: 10.1177/0959651811432669

The online version of this article can be found at:

<http://pii.sagepub.com/content/early/2012/01/05/0959651811432669>

Published by:



<http://www.sagepublications.com>

On behalf of:



Institution of Mechanical Engineers

Additional services and information for *Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering* can be found at:

Email Alerts: <http://pii.sagepub.com/cgi/alerts>

Subscriptions: <http://pii.sagepub.com/subscriptions>

Reprints: <http://www.sagepub.com/journalsReprints.nav>

Permissions: <http://www.sagepub.com/journalsPermissions.nav>

>> Proof - Jan 10, 2012

[What is This?](#)

A hardware–software system for coordinated multi-axis control based on a non uniform rational B-splines interpolator applied to industrial computer numerically controlled machines

Arturo Y Jaen-Cuellar, Jesus J De Santiago-Perez, Rene J Romero-Troncoso, Luis Morales-Velazquez, Juan P Benitez-Rangel and Roque A Osornio-Rios

Proc IMechE Part I:
J Systems and Control Engineering
0(0) 1–10
© IMechE 2012
Reprints and permissions:
sagepub.co.uk/journalsPermissions.nav
DOI: 10.1177/0959651811432669
pii.sagepub.com


Abstract

Motion controllers are widely used in many mechatronic applications such as computer numerically controlled machine tools and robots. These applications demand the use of high performance controllers. Some desirable characteristics in motion controllers are coordinated multi-axis control, smooth movements, and advanced interpolation methods. An important manufacturing aspect involves the production of a workpiece in a free form or using the parametric surfaces in modern computer aided design/computer aided manufacturing systems. The use of parametric curves and surfaces has become standard due to their ability to represent objects having complex surfaces with less data. Another mandatory characteristic of motion controllers is the real-time operation. The contribution of the present work is the design and implementation of a hardware–software system for multi-axis motion control featuring non uniform rational B-splines interpolation. Advantages over other reported works are the hardware implementation which allows the application to real computer numerically controlled machines; also the parallel processing permits the incorporation of additional modules to form an integral system without dependence on external devices. Two experiments were carried out in a retrofitted-to-computer numeric control milling machine in order to show the efficiency of the developed system.

Keywords

Computer numeric control, field programmable gate array, non uniform rational B-splines, hardware–software, motion control, computer aided design/computer aided manufacturing

Date received: 18 July 2011; accepted: 17 November 2011

Introduction

Motion controllers are widely used in many mechatronic applications such as computer numerically controlled (CNC) machine tools and robots. These applications demand the use of high performance controllers. Some desirable characteristics in motion controllers are coordinated multi-axis control, smooth movements, and advanced interpolation methods.^{1–3} An important manufacturing aspect involves the production of a workpiece in a free form or using parametric surfaces in modern computer aided design/computer aided manufacturing (CAD/CAM) systems. The use of parametric curves and surfaces has become

standard due to their ability to represent objects having complex surfaces with less data.⁴ There are many different representations for parametric curves; among them, non uniform rational B-splines (NURBS) has become the standard curve and surface description in CNC systems with Fanuc G06.2 codes.⁵ Another

HSPdigital Research Group, Universidad Autonoma de Queretaro, Mexico

Corresponding author:

Roque A Osornio-Rios, Universidad Autonoma de Queretaro, Rio Moctezuma 249, Col. San Cayetano, San Juan del Rio 76807, Mexico.
Email: raosornio@hspdigital.org

mandatory characteristic of motion controllers is the real-time operation. In the case of machine tools and robots, a servo loop update time lower than 1 ms is required.^{6,7}

Several CNC systems are designed using sequential microprocessors or personal computer (PC)-based systems, and some using a combination of PC and digital signal processor (DSP) for implementing curve interpolation techniques such as curve interpolation and tracking control,⁸ NURBS interpolation,⁹ and off-line algorithm for the feed rate scheduling constrained by chord tolerance, acceleration, and jerk limitations.¹⁰ In order to enhance the capabilities and performance of sequential controllers, several parallel architectures have been proposed. Among them, field programmable gate array (FPGA) motion controllers are presented in some works for the design of smooth movements¹¹ and proportional–integral–derivative (PID) cores.¹² Also FPGA–DSP structures have been reported for numerical control¹³ and the addition of MATLAB/Simulink developments.¹⁴ The Cox–deBoor algorithm is an efficient method for NURBS interpolation, which is reported in some CNC controllers based on FPGA¹⁵ and PC–FPGA.¹⁶ These works deal with the controller design through software and hardware approaches; however, many of them only present simulations.^{14,17} Despite the fact that some works show experimentation even for multi-axis control, there are some drawbacks due to the design, which in general is based on sequential microprocessors or non-optimal platforms such as MATLAB/Simulink that compromise the real-time operation, and synchronization problems due to different communication protocols as a result of the combination of several devices. Besides, there are some commercial controllers on the market with the functionality of multi-axis control and parametric curve interpolations like NURBS, B-splines, and Bezier.¹⁸ Nevertheless, the disadvantage of these products is the closed algorithms since the user cannot manipulate the structures of the profile generator, curve interpolator, or controller. Then, features like reconfigurability and modular structure are not achieved; moreover, the cost is not accessible for small industries. Therefore, it would be desirable to have a motion controller platform featuring the characteristics of high performance motion controllers without affecting the real-time operation, being low cost and reconfigurable.

The contribution of the present work is the design and implementation of a hardware–software system for multi-axis motion control featuring NURBS interpolation. Advantages over other reported works are the hardware implementation which allows the application to real CNC machines; also the parallel processing permits the incorporation of additional modules to form an integral system without dependence on external devices. The proposal is based on FPGA technology, since this can achieve real-time operation in industrial

applications due to its characteristics of modularity and parallel processing. The software part consists of a Linux-based user interface; the non-vendor dependence reduces the overall cost of the platform. Another contribution is the hardware architecture design, which performs the NURBS interpolation based on the Cox–de Boor algorithm, the kinematic profile generation for smooth motion behavior, the PID controller, the quadrature decoder, the universal series bus (USB) interface, and the digital-to-analog converter (DAC) driver in an embedded system approach.

Proposed system

A CNC motion controller is developed in this work that is composed of hardware and software structures, interacting through a USB interface in order to make an integral CNC system. Figure 1 shows the architecture of the proposed system. The main advantage is the development of functional structures for the motion controller, which is capable of accomplishing tasks of control, NURBS interpolation, variable monitoring, and hardware–software interaction without the use of additional modules. The software interface was developed under Linux; it is dedicated to capture and validate the machining information by means of the editor, compilation, and simulation modules. The software module also controls the hardware–software communication protocol for input–output data from the software system to the hardware platform. The hardware platform includes the USB interface performing the communication between the computer and the FPGA; the NURBS interpolator based on the Cox–de Boor algorithm; the kinematic profile generator based on polynomial design; the quadrature decoder for the feedback to the control loop; the PID structure for the motion controller; and the DAC driver, which sends digital signals to the external DAC device. The user can monitor and save the internal signals such as quadrature decoder, servo amplifier, and control signals. The motion controller outputs signals to the servo amplifier via the DAC, which supplies power to move the axes of the CNC machine. It is important to highlight that the NURBS interpolator and additional structures needed for the motion controller are embedded in the same circuit (FPGA) due to the advantages of parallel processing and reconfigurability.

NURBS interpolator

NURBS are the most popular mathematical forms for curve design and have many applications in manufacturing industries, such as automobile body design, ship hull design, etc. NURBS offer a unified form not only for representation of free-form curves, but also for the precise representation of closed-form shapes such as lines, conics, quadratics, etc.¹⁹ A NURBS curve is defined as

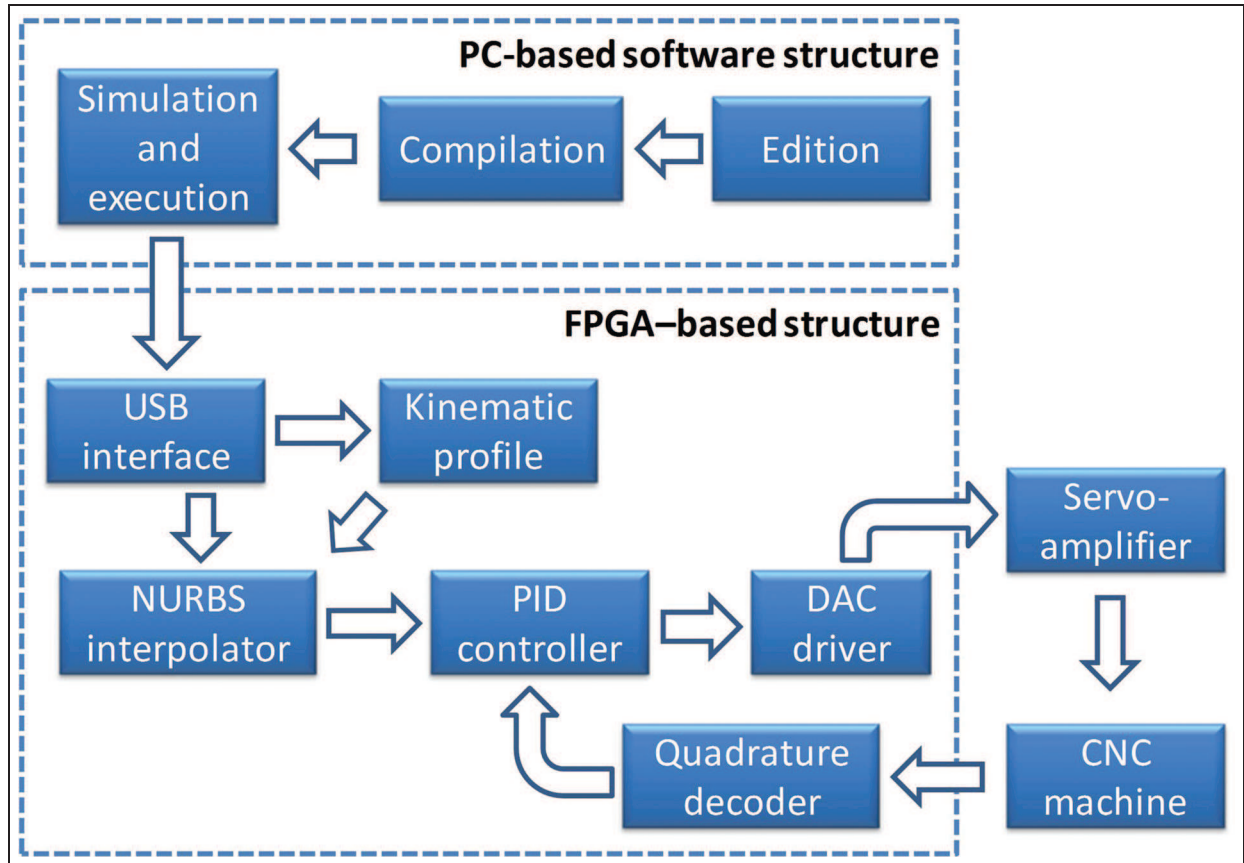


Figure 1. Block diagram of the hardware–software motion controller.

$$s(u) = \frac{\sum_{i=0}^n N_{i,p}(u)P_i w_i}{\sum_{i=0}^n N_{i,p}(u)w_i} \quad (1)$$

where $\{P_i\}$ represents the control points, $\{w_i\}$ are the corresponding weights of $\{P_i\}$, $\{w_i P_i\}$ are the weighted points, $n + 1$ is the number of the control points, and p is the degree of the curve.²⁰ $N_{i,p}$ is the p th-degree B-spline basis function, and can be calculated using the recursive formulas given as

$$N_{i,p}(u) = \begin{cases} 1, & u_i \leq u \leq u_{i+1} \\ 0, & \text{otherwise} \end{cases} \quad (2)$$

$$N_{i,p}(u) = \frac{u - u_i}{u_{i+p} - u_i} N_{i,p-1}(u) + \frac{u_{i+p+1} - u}{u_{i+p+1} - u_{i+1}} N_{i+1,p-1}(u) \quad (3)$$

$\{u_0, \dots, u_{n+p+2}\}$ represents the knot vector and u is the interpolation parameter.

Cox–deBoor algorithm

Most of the time consumed in carrying out the NURBS interpolator is in computation of the basis functions. The computation of $N_{i,p}$ functions is recursive and it requires $2^p - 1$ times of function calls to itself. The computation of one point on the NURBS curve requires p basis functions; finally $p(2^p - 1)$ function

calls are needed to calculate one point of the curve.²⁰ Considering the heavy computational load for recursive computation of the B-splines basis functions, the Cox–de Boor algorithm is introduced in order to interpolate NURBS curves with a lower computational load compared with the recursive formula. Cox–de Boor is an efficient mathematical algorithm for NURBS interpolation without the direct calculation of B-splines basis. The algorithm is based on the following equation

$$P_i^j(u) = (1 - \alpha_i^j) P_i^{j-1}(u) + \alpha_i^j P_{i+1}^{j-1}(u) \quad (4)$$

where

$$\alpha_i^j = \frac{u - u_{r-j+i}}{u_{r+i} - u_{r-j+i}} \quad (5)$$

for $j = 1, 2, \dots, p$ and $i = 0, 1, \dots, p - j$. The parameter u is within the interval $u_r \leq u \leq u_{r+1}$, and the NURBS point is the resulting point P_0^p .

Kinematic profile

Kinematic profile is an important aspect that should be considered in a control process related to computerized numerical control and robots because it interferes directly with the system performance and can generate unwanted vibrations and large peaks of jerk. Furthermore, the kinematic profile is essential to

achieving high precision and high quality in machine tool processes.²¹ Feed, acceleration, and jerk are the key parameters for the kinematic profile design. In this work, a sixth-degree piecewise polynomial profile is designed for the motion along the tool path generated by the NURBS curves. The polynomial profile avoids large peaks of jerk and ensures a smooth dynamic behavior of the machining process as reported by Rivera-Guillen et al.²² The profile is designed based on several dynamic constraints defined by the user according to the machining requirements. The parameters defined are maximum feed V_m and maximum acceleration and deceleration values, A_m and D_m . First, the arc length of the NURBS curve L is estimated; this parameter is calculated by means of an accumulative sum. Then, the feed profile designed is subjected to kinematic restriction and can be presented as

$$y_v(t) = \begin{cases} \frac{16}{5\tau_a^5} A_m t^5 - \frac{8}{\tau_a^4} A_m t^4 + \frac{16}{3\tau_a^3} A_m t^3, & 0 \leq t \leq \tau_a \\ V_m, & \tau_a \leq t \leq \tau_c \\ V_m + \frac{16}{5\tau_d^5} D_m (t - \tau_a - \tau_c)^5 \\ - \frac{8}{\tau_d^4} D_m (t - \tau_a - \tau_c)^4 + \frac{16}{3\tau_d^3} D_m (t - \tau_a - \tau_c)^3, & \tau_a + \tau_c \leq t \leq \tau_a + \tau_c + \tau_d \end{cases} \quad (6)$$

where τ_a , τ_c , and τ_d represent the time intervals of acceleration, constant feed, and deceleration, respectively. These intervals are defined by equations (7) to (9)

$$\tau_a = \frac{15V_m}{8A_m} \quad (7)$$

$$\tau_d = -\frac{15V_m}{8A_m} \quad (8)$$

$$\tau_c = \frac{L}{V_m} - \frac{\tau_a}{2} - \frac{\tau_d}{2} \quad (9)$$

Equation (6) states the feed profile, and then the position profile is calculated by means of numeric integration. Figure 2 presents an example of the kinematic profile. The feed profile represents a velocity profile which is the designed curve expressed by equation (6), where the three sections of acceleration, constant feed, and deceleration are defined by the time intervals τ_a , τ_c , and τ_d , respectively. The position profile represents the advance of the machine cutter, and it is obtained by numeric integration of the feed profile. The acceleration profile is the first derivative of the feed profile; this profile also highlights the three time intervals. Since the feed profile is a smooth curve defined by a six-degree polynomial, acceleration and position profiles are also smooth curves.

The sixth-degree polynomial also allows smooth motion and limited jerk during the movement; the online evaluation of the profile is performed numerically by the efficient parametric technique reported by Osornio-Rios et al.⁶

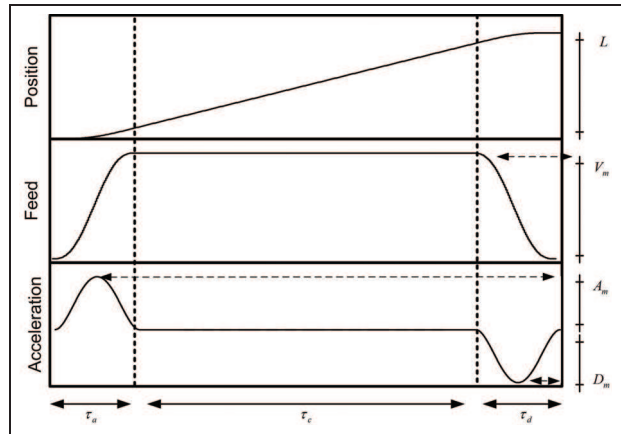


Figure 2. Kinematic profile.

Design of the hardware–software motion controller

In order to develop the CNC system, software and hardware structures presented in the section “Proposed system” were developed. Making efficient designs gives the possibility of controlling the interaction between them so synchronicity problems are avoided. Moreover, adequate hardware design allows embedding several modules into the same FPGA and external modules such as DSPs or commercial cards are not necessary. Efficient software design permits a correct interface between the user and the control system so that tasks of monitoring, simulation, and execution are achieved. In this section, the proposed hardware–software system for the motion controller is described.

Software design

The software part is dedicated to the design of a graphic user interface (GUI) for the communication between users and the hardware platform; it also controls the port connection for data transfer between hardware and software. The software was developed under the Linux operating system under C++ programming. Glade tools were employed for the graphic interfaces since they allow an easy integration with Integrated Development Environment (IDE). The software is divided into four main sections: editor, compiler, simulator, and jog module. Figure 3 shows the main window of the developed software. The editor loads the curve information from a file *.txt. The NURBS curves are described in the standard G06.2 where control points, knots, weights, and feed rate are defined. The editor allows the inclusion of linear and circular curves based on G codes (ISO 6983). Furthermore, the software allows typing the code directly into the editor. Once the code for the curve is in the editor, the software performs a program compilation for checking syntax and eliminating unnecessary spaces and tabulations. When the

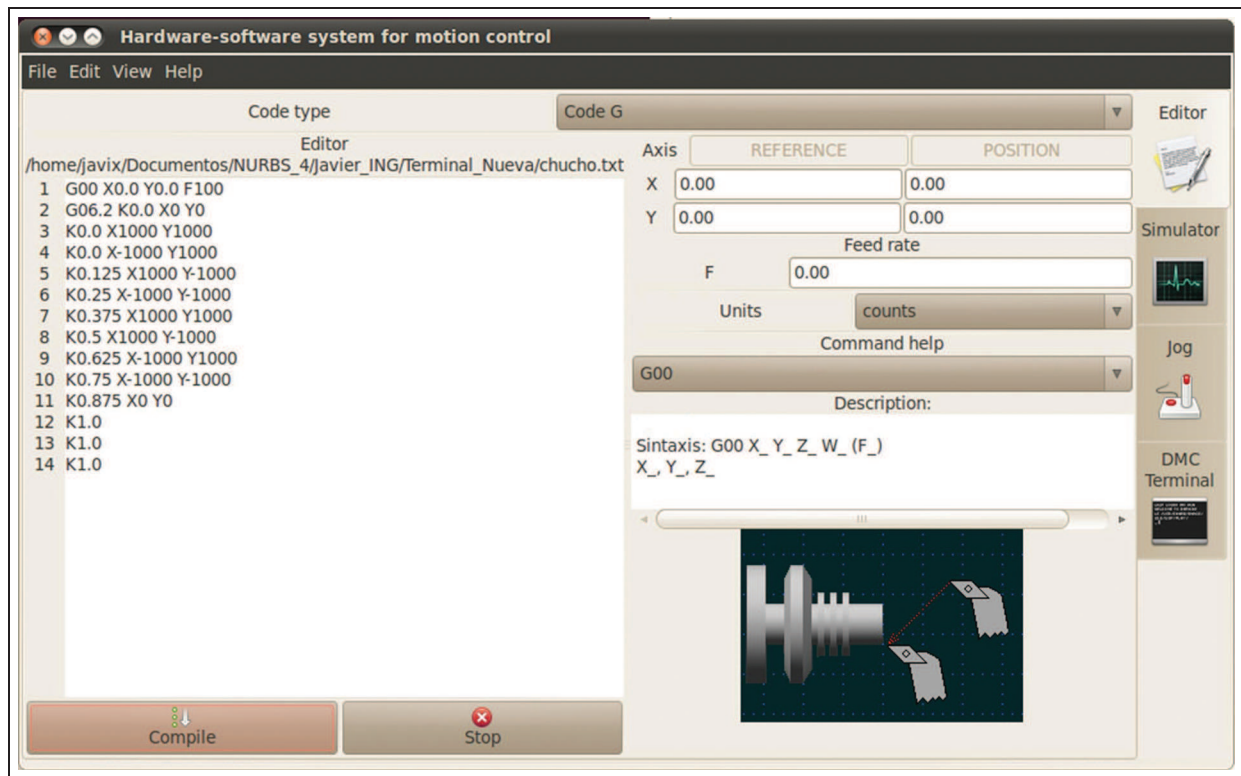


Figure 3. Developed software.

compilation is successfully performed without syntax errors, the simulator window is enabled. In this module the user can simulate the machining or directly execute the program. Program execution achieves command interpretation and sends it to the FPGA board through the USB interface. Additionally, the simulator interface plots the desired curve described by the code and also overlaps the curve generated by movement of the axis machine in order to compare them. Another software option is the jog module, used for positioning the axis machine to home position, and defines the initial position before the machining starts. The software includes additional windows where the parameters of the PID controller are stated, being obtained after the identification process described by Morales-Velazquez et al.²³

Hardware design

Figure 4(a) presents a block diagram of the hardware design for the control loop. Hardware design for the motion CNC controller is implemented in an FPGA device which contains the structure for executing commands of profile generation, NURBS generator, PID controller, DAC driver, and USB interface. In this paper, one contribution is the FPGA implementation of the NURBS interpolator; it was made according to the Cox–Boor algorithm based on node insertion. Figure 4(b) shows the proposed design by a block diagram where weighted control points, knot vector, and the parameter u are the inputs and the NURBS curve is the output. The design is based on equations (4) and (5). First, control points are multiplied by the corresponding weights and

the random access memory (RAM) module stores the partial results. The index module finds the value r which satisfies $u[u_r, u_{r+1})$ and u is the parameter generated by the dynamic profile module according to Osornio-Rios et al.⁶ The alpha module and the Multiplier Accumulator (MAC) module perform equations (4) and (5). Finally, the data obtained are projected to axes X, Y, Z in accordance with Pourazady and Xu.¹⁹ The resulting NURBS curve is the input to the PID motion controller according to Figure 4(a).

Hardware–software interaction

The success of the presented CNC system is the development of a communication protocol which allows the flow of machining commands and reciprocally measured data between software and hardware. In the hardware part there are three main levels: digital processing, interconnection, and user interface. The digital processing level concerns the development of a micro-processing unit (MPU) into the FPGA device which is dedicated to the data processing task. The interconnection level is composed of the system for interconnection of media (SIM). This system is the physical card which contains the necessary elements to connect the MPU with external elements like converters (digital-to-analog, analog-to-digital), buffers, voltage supply, and connectors to machine and to computer. The hardware platform can be summarized as the conjunction of the MPU with the SIM; it is denominated a distributed reconfigurable controller (DRC). The DRC is a unit capable of performing the monitoring and control task.

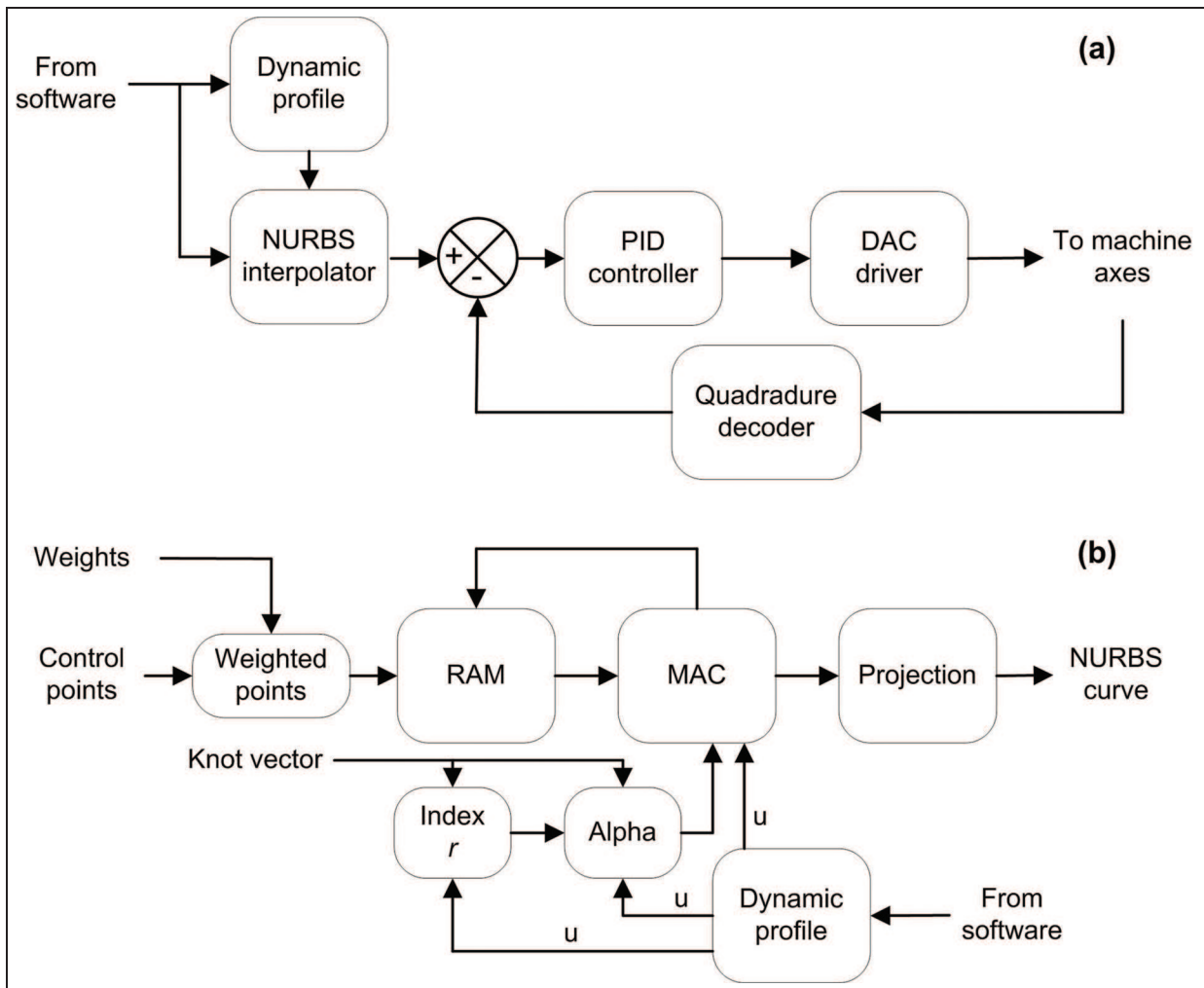


Figure 4. Block diagram of the hardware design: (a) control loop; (b) NURBS interpolator.

In a complete industrial application the integration of several DRCs is necessary in which every DRC is dedicated to a specific function; for example, system identification, PID tuning, system fault detection, dynamic and vibration analysis, etc. In the same way, communication between DRCs is necessary to coordinate every action, synchronization, and allowance of data flow; a communication protocol is defined to cover this aspect. The protocol is based on two messages: one message is for data requirement and the other is to reply to the data request. Both messages contain information of the origin and destination device. The reply message contains the size data in bytes and the required data. The first byte of every message corresponds to an identifier called a token. There are several valid tokens: request, synchronization for request or reply message, and error for reply message. The syntax of the messages is

Request: Token – Source DRC – Destiny DRC

Reply: Token – Source DRC – Destiny

DRC – Data size – Data

Finally, the user interface level is composed of the communication method between the DRC and the GUI

developed in the computer. The DRC data hardware has a memory region in the computer and there is a virtual representation of the DRC unit; this element is equivalent to a DRC and is called the DRC agent. The DRC agent is executed in a thread independent of the GUI. The DRC agent and the DRC hardware are communicated by a USB interface which allows the request and reply of data. When the simulation of the NURBS path is performed successfully in the GUI, the user can execute the machine motion by sending the machining information to the hardware platform. Curve parameters, kinematics specifications, and control parameters are sent into a file with Extensible Markup Language (XML) structure through the USB port. The hardware microprocessor interprets the command to reconstruct the curve with the specified kinematic profile and performs the motion control of the machine's axes.

Simulation

Two study cases were designed in order to test the efficiency of the NURBS interpolator. In this stage, the developed methodology for curve interpolation was evaluated. This methodology is based on the Cox–de

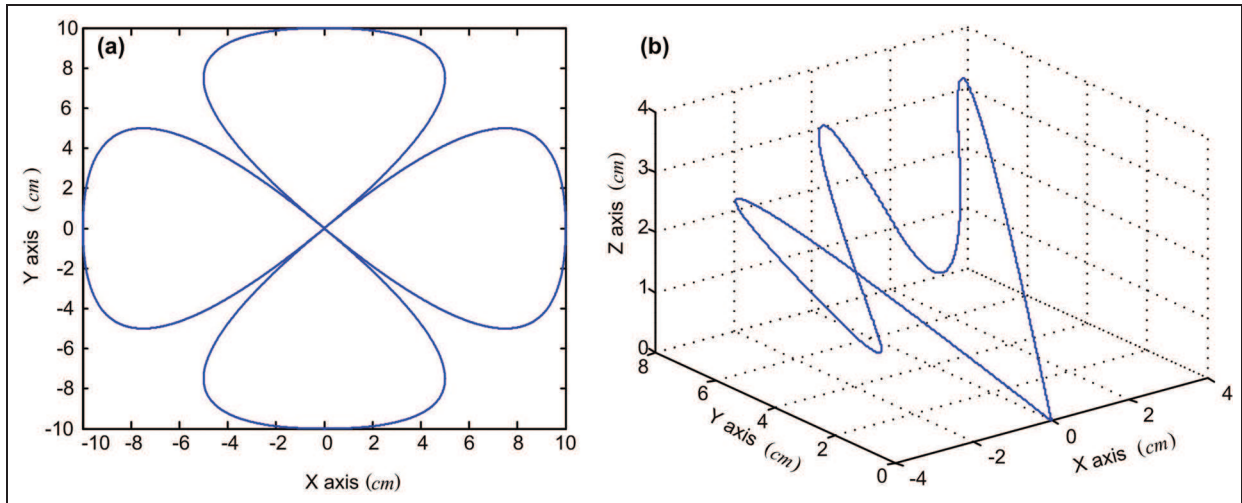


Figure 5. Simulations: (a) 2D NURBS curve; (b) 3D NURBS curve.

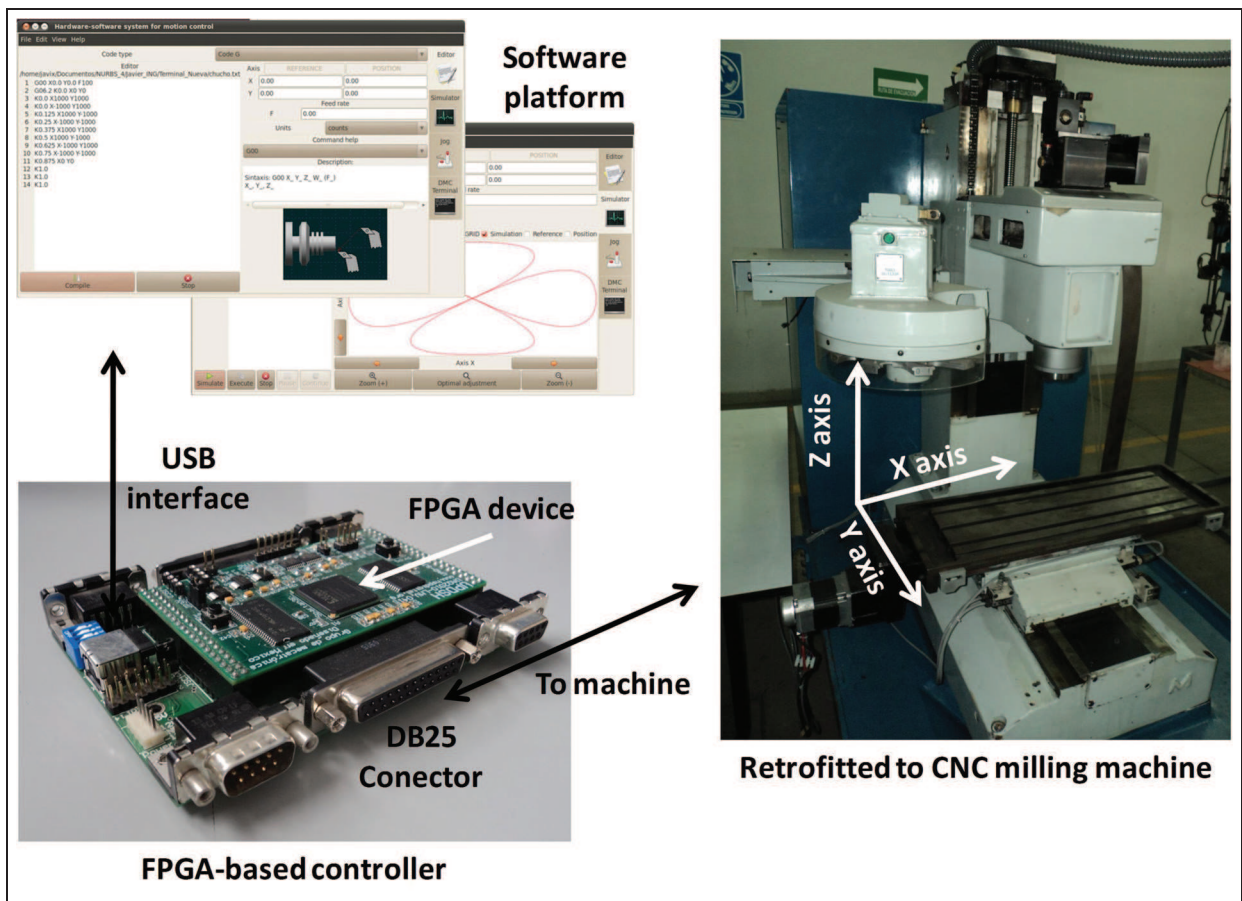


Figure 6. Experimental setup.

Boor algorithm and the profile generator described in the sections “Cox–deBoor algorithm” and “Kinematic profile”. The first case corresponds to a two-dimensional (2D) NURBS path; for the curve design the control points were $\{(0,0), (10,10), (-10,10), (10,-10), (-10,-10), (10,10), (10,-10), (-10,10), (-10,-10), (0,0)\}$, the knot vector was $\{0, 0, 0, 0.125, 0.25, 0.375, 0.5, 0.625, 0.75, 0.875, 1, 1, 1\}$, and the weight vector was

$\{1, 1, 1, 1, 1, 1, 1, 1, 1, 1\}$. The interpolated curve is depicted in Figure 5(a). Another important feature of the proposed system is the reconfigurability; this property allows one to adjust the NURBS interpolator for three or more axes. The second study case corresponds to a three-dimensional (3D) curve, the parameters are control points $\{(0,0,0), (5,10,5), (1,4,0), (0,10,5), (-1,4,0), (-5,10,5), (0,0,0)\}$, knot vector $\{0, 0, 0, 0.2, 0.4, 0.6, 0.8,$

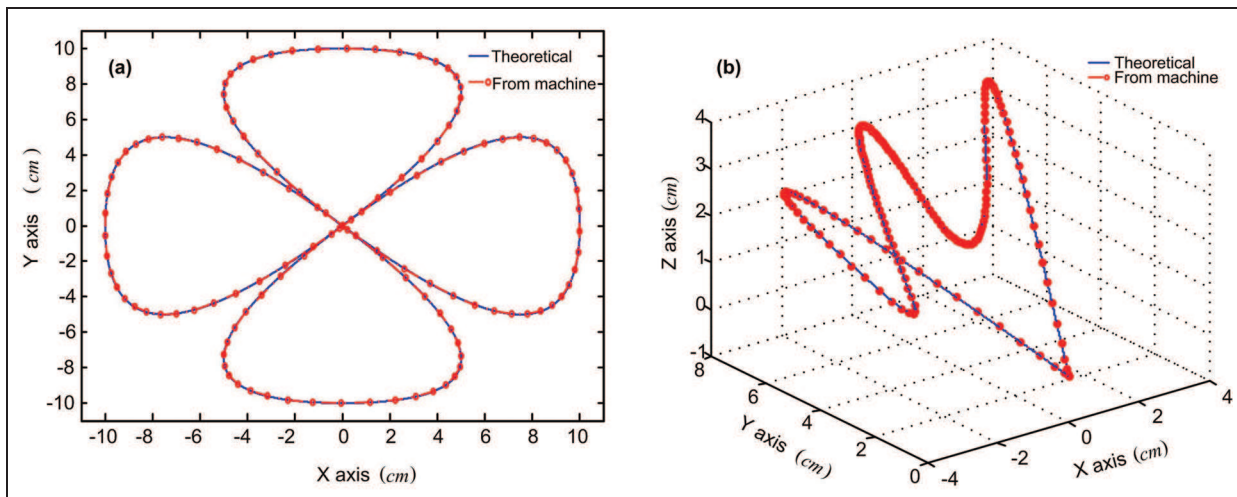


Figure 7. Experimental results: (a) first study case; (b) second study case.

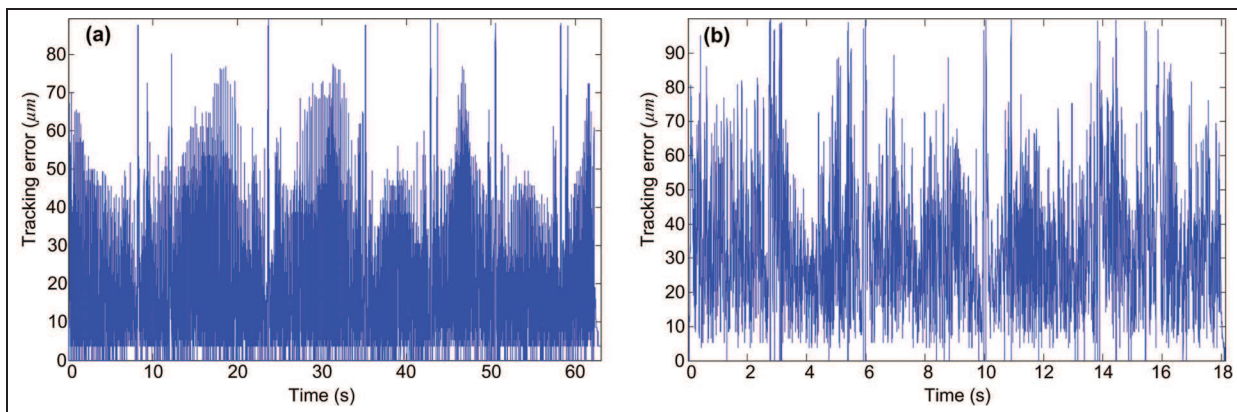


Figure 8. Tracking error: (a) first study case; (b) second study case.

1, 1, 1}, and weights {1, 1, 1, 1, 1, 1}. The interpolated curve based on the FPGA NURBS interpolator is shown in Figure 5(b).

Experimentation

Figure 6 presents an overview of the experimental setup. The GUI is the software module that allows the user operation of the system. The FPGA-based controller is the hardware module that interacts with the software and the milling machine through a USB and a DB25 connector, respectively. The experimental part of this work is dedicated to evaluate the implementation of the motion controller system with the NURBS interpolator. In order to test the efficiency of the system, the same curves designed in the simulation were tested in the FPGA-based system. The experiment was performed on a retrofitted-to-CNC milling machine. The system was developed into a Spartan-3 FPGA from Xilinx. It is important to highlight that the NURBS interpolator along with the motion controller and the additional modules described in Figure 1 were embedded into the same FPGA device.

Results

The experimental results of the first and second study cases are shown in Figure 7(a) and (b), respectively. The sampling time corresponding to the controller for both cases was $T = 1$ ms which is standard in industrial applications, the specified maximum feed was $V_m = 1200$ mm/min for the design of the sixth-order kinematic profile stated in the section “Kinematic profile”, and the resolution of the optical encoder was $1 \mu\text{m}/\text{count}$. The first study case spends 63 s for the curve generation with a total length of 123.3 cm, while the second curve spends 18.1 s for a length curve of 29 cm. A comparison between the theoretical model and the experimentally measured data was performed in order to validate the efficiency of the interpolators. The root mean square of the tracking error was $22.1 \mu\text{m}$ and $35.2 \mu\text{m}$ for the first and second study case, respectively, representing percentage errors of 1.2% and 2%. In order to show the behavior of the tracking error Figure 8(a) and (b) presents the results of both study cases, where each plot exhibits that the error is bounded to a maximum value of $90 \mu\text{m}$ for the first study case and $100 \mu\text{m}$ for the second study case.

Conclusions

This paper presents a hardware–software system for NURBS interpolation and motion control of CNC machines. The main advantage of the system is the connection of the software and hardware platforms by means of a USB interface with the defined protocol of the DRC. Communication between the DRC and software allows the correct data flow for controlling and monitoring the machining process, in distinction from other reported works where several devices are employed for the CNC system. Also, the development of the FPGA-based motion controller allows the reconfigurability to two, three, or more axes. Another property is the presentation of the scheme for NURBS interpolation. NURBS curves provide smooth and continuous trajectories, and motion profile generation ensures a controlled dynamic motion. Since the majority of the reported works highlight the heavy computational load for NURBS interpolation, another contribution of this work is the design and implementation of the interpolator together with the CNC controller into an FPGA device. The advantages of parallel processing and reconfigurability allow the implementation of the interpolator and additional modules: PID controller, USB interface, DAC driver. The computationally efficient Cox–de Boor algorithm allows implementation of the NURBS interpolator; also this implementation permits the interpolation technique of more parametric curves such as Bezier and B-splines since these curves are a particular case of NURBS. Moreover, the simulation and the experimental results prove the system efficiency in real applications because it presents a small tracking error. The system is flexible due to supporting NURBS curves and the traditional linear and circular interpolations, which feature makes an integral hardware–software system for machining applications. Furthermore, the system avoids the dependence on external devices like other reported works presented in the literature. The system is also capable of modifying the hardware structure due to the feature of reconfigurability while the closed algorithms of commercial systems do not permit this.

Funding

This project was partially supported by CONACyT scholarship 260421 and SEP-CONACyT 84723 project.

References

1. Zhang L, Bian Y, Chen H, et al. Implementation of a CNC NURBS curve interpolator based on control of speed and precision. *Int J Prod Res* 2009; 47(6): 1505–1519.
2. Li W, Liu Y and Yamazaki K. The design of a NURBS pre-interpolator for five-axis machining. *Int J Adv Mfg Technol* 2008; 36(9–10): 927–935.
3. Uchiyama N, Takagi S and Sano S. Design of discrete-time adaptive repetitive controllers and application to

- feed drive systems. *Proc IMechE, Part I: J Systems and Control Engineering* 2007; 221(1): 39–47.
4. Ulker E, Turanalp ME and Halkaci HS. An artificial immune system approach to CNC tool path generation. *J Intell Mfg* 2009; 20(1): 67–77.
5. Wang T, Cao Y, Chen Y, et al. A new feedrate adaptation control NURBS interpolation based on de Boor algorithm in CNC systems. In: *American control conference*, Seattle, WA, USA, 11–13 June 2008, pp.4075–4080.
6. Osornio-Rios RA, Romero-Troncoso RJ, Herrera-Ruiz G, et al. FPGA implementation of higher degree polynomial acceleration profiles for peak jerk reduction in servomotors. *Robot Comput Integrated Mfg* 2009; 25(2): 379–392.
7. Osornio-Rios RA, Romero-Troncoso RJ, Herrera-Ruiz G, et al. The application of reconfigurable logic to high speed CNC milling machines controllers. *Contr Eng Pract* 2008; 16(6): 674–684.
8. Shukun C, Heng Z, Xiangbo H, et al. Software and hardware platform design of open CNC system. In: *IEEE international symposium on knowledge acquisition and modeling workshop*, Wuhan, China, 21–22 December 2008, pp.139–142.
9. Tsai M-C, Yen CL and Yau HT. Development of a hybrid iterative learning control for contouring NURBS curves. *Asian J Cont* 2011; 13(1): 107–125.
10. Lee A-C, Lin M-T, Pan Y-R, et al. The feedrate scheduling of NURBS interpolator for CNC machine tools. *Comput Aided Des* 2011; 43(6): 612–628.
11. Cho JU and Jeon JW. A motion-control chip to generate velocity profiles of desired characteristics. *ETRI J* 2005; 27(5): 563–568.
12. Astarloa A, Lazaro J, Bidarte U, et al. FPGA technology for multi-axis control systems. *Mechatronics* 2009; 19(2): 258–268.
13. Wang Q-Y, Huang M-F, Kuang B, et al. On the application of compute numerical control based on DPS. In: *Third international conference on genetic and evolutionary computing*, Guilin, China, 14–17 October 2009, pp.586–589.
14. Liu D, Mei X, Tao T, et al. Research on design method and application of PFPGA-based motion control. *Intell Robot Applic* 2008; 5315: 419–430.
15. Yau H-T, Lin M-T, Chan Y-T, et al. Design and implementation of real-time NURBS interpolator using a PFPGA-based motion controller. In: *International conference on mechatronics*, Taipei, Republic of China, 10–12 July 2005, pp.56–61.
16. Lin M-T, Yau H-T, Nien H-W, et al. FPGA-based motion controller with real-time look-ahead function. In: *International conference on advanced intelligent mechatronics*, Xian, China, 2–5 July 2008, pp.1406–1411.
17. Pislaru C, Ford DG and Holroyd G. Hybrid modeling and simulation of a computer numerical control machine tool feed drive. *Proc IMechE, Part I: J Systems and Control Engineering* 2004; 218(2): 111–120.
18. Siemens Automation Technology SINUMERIK 840D, <http://www.automation.siemens.com/mc/mc/en/automation-systems/cnc-sinumerik/sinumerik-controls/sinumerik-840d/Pages/sinumerik-840d.aspx> (accessed August 2011).
19. Pourazady M and Xu, X. Direct manipulations of B-spline and NURBS curves. *Adv Eng Softw* 2000; 31(2): 107–118.

20. Tsai M-S, Nien H-W and Yau H-T. Development of an integrated look-ahead dynamics-based NURBS interpolator for high precision machinery. *Comput Aided Des* 2008; 40(5): 554–566.
21. De Santiago-Perez JJ, Osornio-Rios RA, Romero-Troncoso RJ, et al. DSP algorithm for the extraction of dynamics parameters in CNC machine tool servomechanisms from an optical incremental encoder. *Int J Machine Tools Mf* 2008; 48(12–13): 1318–1334.
22. Rivera-Guillen JR, Romero-Troncoso RJ, Osornio-Rios RA, et al. Design methodology for fully dynamic-controlled polynomial profiles and reduced tracking error in CNC machines. *Int J Adv Mfg Technol* 2010; 51(5–8): 723–737.
23. Morales-Velazquez L, Romero-Troncoso RJ, Osornio-Rios RA, et al. Special purpose processor for parameter identification of CNC second order servo systems on a low-cost FPGA platform. *Mechatronics* 2010; 20(2): 265–272.

PID-Controller Tuning Optimization with Genetic Algorithms in Servo Systems

Regular Paper

Arturo Y. Jaen-Cuellar¹, Rene de J. Romero-Troncoso²,
Luis Morales-Velazquez¹ and Roque A. Osornio-Rios^{1,*}¹ Facultad de Ingenieria, Universidad Autonoma de Queretaro, Campus San Juan del Rio, Rio, Qro., Mexico² DICIS, Universidad de Guanajuato, Salamanca, Gto., Mexico

* Corresponding author E-mail: raosornio@hspdigital.org

Received 30 Jul 2012; Accepted 27 May 2013

DOI: 10.5772/56697

© 2013 Jaen-Cuellar et al.; licensee InTech. This is an open access article distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/3.0>), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Abstract Performance improvement is the main goal of the study of PID control and much research has been conducted for this purpose. The PID filter is implemented in almost all industrial processes because of its well-known beneficial features. In general, the whole system's performance strongly depends on the controller's efficiency and hence the tuning process plays a key role in the system's behaviour. In this work, the servo systems will be analysed, specifically the positioning control systems. Among the existent tuning methods, the Gain-Phase Margin method based on Frequency Response analysis is the most adequate for controller tuning in positioning control systems. Nevertheless, this method can be improved by integrating an optimization technique. The novelty of this work is the development of a new methodology for PID control tuning by coupling the Gain-Phase Margin method with the Genetic Algorithms in which the micro-population concept and adaptive mutation probability are applied. Simulations using a positioning system model in MATLAB and experimental tests in two CNC machines and an industrial robot are carried out in order to show the effectiveness of the proposal. The obtained results are compared with both the classical Gain-Phase Margin

tuning and with a recent PID controller optimization using Genetic Algorithms based on real codification. The three methodologies are implemented using software.

Keywords Frequency Response (FR), Gain-Phase Margin, Genetic Algorithm (GA), Positioning System, PID control

1. Introduction

Positioning control systems control systems with motion capabilities in a quick, smooth and precise way, playing a key role in the manufacturing industry [1]. These systems are wide-spread in automated industries and are used for quality improvement, the reduction of production time and profit increase, despite the initial investment [2]. The principal and most common industrial applications are in CNC machinery and industrial robotics. When designing control systems in the industry, approximately 95 percent of these systems are Proportional-Integral-Derivative (PID) controllers [3] and 85 percent of these are implemented on closed-loop systems with a single input and a single output configuration [4]. The reasons for the wide use of these controllers are the easiness of their

implementation in digital systems, their robustness, low cost maintenance, and so on [5, 6]; their effectiveness has been demonstrated in positioning control applications as shown in [7, 8]. Now, the positioning systems are normally unstable when they are implemented in a closed-loop configuration, so when the controller is introduced into the closed-loop it needs to be effectively tuned. However, PID controller tuning for positioning control systems is a time-consuming task. Typically, the adjusting of the controller parameters is carried out using formulas to provide a performance, which, although not deficient, is far from optimal. Therefore, much effort has been given to analysing the complex plant models of these systems in order to obtain an efficient tuning.

The most important tasks in control design are the system model identification, the law or the control algorithm, the tuning and the control implementation. Besides from the proper operation of all the elements within a control-loop, the performance of the whole control system mainly depends on the controller parameters which are obtained by its tuning [9, 10]. For many years much research has been done with the aim of adjusting the controller parameters. The most popular methodologies used for PID controller tuning, in industrial applications, can be classified as i) empirical methods such as Ziegler-Nichols (ZN) and Cohen-Coon (CC) [11], and ii) analytical methods such as root locus (RL) and frequency response (FR) analysis [12, 13]. The empirical methods have the advantage of applying easy rules to simple mathematical models, but do not provide as good results as expected. The analytical methods can deal with some higher order models; they are more precise and have the advantage of improving some specific parameters of the system response such as stability, rising time, steady state error, overshoot, etc. Both empirical and analytical methods are time consuming; however, the Gain-Phase Margin (GPM) method based on FR analysis has been demonstrated to be more efficient than the other methods when it is applied to PID controller tuning for positioning control systems such as for CNC machines and industrial robots, as shown in [14, 15]. This method has the advantage over time domain analysis that it provides information about the relative and absolute system stability. Additionally, the generality and versatility of this method allows it to be used to automatically compute the parameters of a PID controller, as some available commercial controllers like Galil Motion Control do [16]. Nevertheless, this method has some problems with its design and optimization; sometimes the adjustment of the PID controller parameters does not go as expected. Recently, modern optimization techniques such as Genetic Algorithm (GA), Particle Swarm Optimization (PSO) and Ant Colony Optimization (ACO) are applied in the parameter optimization of PID controllers; a comparative study is presented in [17]. Nonetheless, the actual

tendency in parameter optimization is the application of the GA due to their powerful searching capabilities and heuristic characteristics, more specific details about the GA can be reviewed in [18]. The effectiveness of this technique has been demonstrated in positioning control systems for robotics path generation in [19-21]. Therefore, GA has also been used in control tuning applications, being shown to obtain better results than classical techniques. Besides, as can be seen in [22, 23] the micro-GA, a relatively new concept, could be useful for reducing computational resources and for providing appropriate solutions in many optimization problems when they are carefully designed and genetic diversity is ensured.

Due to the advantages and despite the disadvantages of the FR analysis, many works have used this technique for PID control tuning. For instance, simulation software developed using C++ for PID tuning was presented in [24]. To obtain the PID coefficients the open and closed loops models, including the PID structure in a Laplace form, were evaluated in the interest frequencies. The software was tested on simulated models up to the third order. Meanwhile, a controller design methodology was proposed in [25] based on process FR data, identified through relay feedback tests and using multiple points on the Nyquist curve. The controller model was obtained by using the least squares method and the proposal was validated with simulations over linear systems and compared with classical ZN and GPM methods. On the other hand, an approach for an adaptive control for PI controllers was presented in [26]. The proportional and integral gains were adjusted using a Nelder-Mead algorithm. Also, open-loop gain and phase margin constraints were included in it. Simulations and experimental tests were carried out on a simple pilot column distillation loop with a commercial controller but using large sample periods for the tuning process. Recently, the FR methodology was used in [27] to efficiently design a feed forward compensator for an industrial motion stage in a transmission electron microscope. The approach does not necessarily provide optimal tracking behaviour; additional control action may be required. Simulations using a commercial controller for data acquisition and MATLAB/SIMULINK validated the proposal. Also, the GPM method has been applied in tuning applications tackling a specific problem in positioning control systems, as seen in [7, 28]. However, this method was applied offline because the controller tuning took longer. In contrast, GA is a powerful searching algorithm that mimics natural behaviour based on natural genetics and natural selection [29, 30]. Many works have proposed the use of GA to optimize the PID gains in control systems, but the standard GA has the drawback of pre-maturity and stagnation, while it looks for a global optimal solution. Recently, in [31] parameter

optimization was carried out for a small helicopter based on GA focusing on the stability and adaptability of the system. The performance of the system is validated for the model simulations. A methodology for a PID controller design was presented in [10], where the antibodies concept was used to improve a simple GA providing population diversity and searching speed. The validation of this approach was performed using simulations for an artificial leg. In [32] an adaptive niche GA was proposed to find the optimal PID parameters, an elitist strategy ensured the stable convergence, the niche ideology kept the population diversity, and the adaptive mutations and crossover probabilities enhance the local search ability. The methodology was tested on four proposed models of first and second order by simulations. The GA and chaos optimizing concept were integrated in [33] for PID parameter optimization. Here, the initial population was generated by a logic sequence and adaptive mutation probability was used to jump the local optima. Simulation tests were performed over a third order proposed plant model. A Real-coded Genetic Algorithm (RGA) approach using floating point variables representation presented in [34] was used to optimize the PID parameters of a temperature system model, obtained by software identification tools; the system response was simulated compared to ZN and CC methods. Recently in [35] the RGA technique was applied for system identification and control tuning for a model reference adaptive control for a hybrid tank system. The model is obtained using recollected data through an acquisition card and the performance of the proposal was simulated using MATLAB. Moreover, an approach was developed in [36] of a self-tuning PID controller based on GA. A dominant selection and cyclic mutation operators were used to improve the fitness of the GA population. The method implementation was carried out using software and was validated by simulations. A tuning approach for PID controllers based on GA using adaptive mutation and crossover probabilities to avoid premature convergence was developed in [37]. Simulations of this approach were conducted in MATLAB for two simple third order models. Also, in [30] an adaptive GA with variable mutation probability was applied to optimize the PID controller gains for an ideal second order model representing a coal transportation system. Simulations of the proposal were carried out in SIMULINK. Meanwhile, in [38] a fuzzy PID controller was used for high order plants with time delays. The controller parameters were tuned by using the GA and two variants of PSO techniques offline. The proposal validation was executed by simulations in MATLAB. Finally, a multi-objective non-dominated sorting GA was developed in [39] for the tuning of a PID controller applied to a robotic manipulator. The results of simulations demonstrated the performance of the approach. From the analysis of these works, several aspects about control tuning methods can

be concluded. For instance, the FR analysis has notorious advantages over time domain analysis and it can be implemented as an automatic tuner for positioning control systems. The GA is a powerful optimization technique when it is combined with other methodologies to avoid premature local convergence on tuning applications. PID control tuning has been used for simulated models in the majority of the works presented up to now. Yet, an integration of GA, using the micro-GA concept, and FR analysis have not been carried out for the optimal adjustment of the PID control parameters; besides, it would be desirable to implement this approach in real positioning control systems, which is the proposal of this work, beyond just simulations.

Given the necessity of increasing the performance in positioning control systems, the contribution of this paper is a new methodology for the optimization of PID controller parameters. This methodology couples the GPM method which is based on the FR analysis and the Adaptive Micro Genetic Algorithm (AMGA). One interesting characteristic of the proposal is the adaptive mutation probability which is implemented in order to increase the searching speed and avoid losing genetic information. On the one side, the GPM method is used to calculate the controller gains for any order model plant of real processes. The input parameters of the GPM method are the response time and the phase margin desired of the positioning control system. On the other hand, the AMGA technique ensures the GPM method will have optimized input parameters; the approach makes use of adaptive mutation probability to avoid local stagnation and uses micro-population concept to reduce computing resources and increase searching speed. The gains obtained with this approach are implemented in a proprietary controller board for three servo systems: a CNC lathe with two axes, a CNC milling machine with three axes and a six DOF industrial robot PUMA (Programmable Universal Manipulation Arm). Experimental results are compared with the classic GPM tuning method applied manually for the adjusting of the PID controller gains and with PID parameters optimization using a RGA method as presented in [35]. A simulation using a theoretical model of a positioning system is also performed to compare the methodologies implemented.

2. Background

2.1 PID Control scheme

The PID controller has been widely used since its invention in 1910 [40]. There are many PID control configurations, but the most common implementation of this controller, as shown in Figure 1, is the feedback-loop with a single input and a single output, as explained in [41].

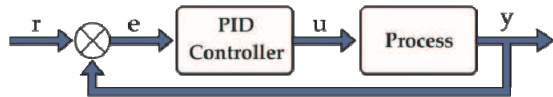


Figure 1. Block diagram of the simplest closed-loop system.

2.1.1 Typical structure

The typical implementation of the PID control is shown in Figure 2. The signal error, $e(t)$, enters the PID control block and the resulting excitation signal is the sum of the error signal affected by the proportional, integral and derivative actions.

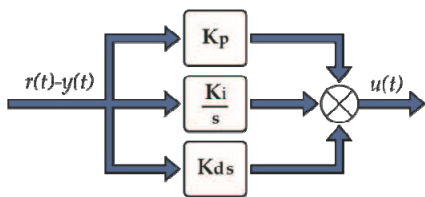


Figure 2. Block diagram of typical PID control structure.

The output excitation signal of the PID controller, where $e(t) = r(t) - y(t)$, is:

$$u(t) = K_p e(t) + K_i \int e(t) dt + K_d \frac{d}{dt} e(t) \quad (1)$$

Also, the equation (1) can be rewritten in Laplace form as:

$$u(s) = e(s) \left(K_p + \frac{K_i}{s} + K_d s \right) \quad (2)$$

Finally, under the above scheme the transfer function of the PID controller or the control law is specified by

$$C(s) = \frac{u(s)}{e(s)} = K_p + \frac{K_i}{s} + K_d s \quad (3)$$

The equation (3) can be implemented on a digital system as an equation in differences for a second order infinite impulse response (IIR) filter specified by

$$y(k) = a_0 u(k) + a_1 u(k-1) + a_2 u(k-2) \dots - b_1 y(k-1) - b_2 y(k-2) \quad (4)$$

The corresponding discrete transfer function of (3) considering (4) is:

$$C(z) = \frac{a_0 + a_1 z^{-1} + a_2 z^{-2}}{1 - b_1 z^{-1} - b_2 z^{-2}} \quad (5)$$

Therefore, the relationship between the PID gains and its implementation in a digital system is shown in Table 1.

$C(z)$	a_0	a_1	a_2	b_1	b_2
$C(s)$	$K_p + K_i + K_d$	$-(K_p + 2K_d)$	K_d	1	0

Table 1. Equivalence of parameters for PID controllers.

2.2 System frequency response analysis

According to Tal in [42], system analysis is a method for predicting the system response. With regards to frequency, this analysis considers two main aspects:

- The system stability
- The system response speed

The nature of the response is indicated by the system stability. The stability grade is usually expressed as the phase margin, θ_m . On the other hand, the response speed measures the time in which the system performs a small correction without saturation. Usually the crossover frequency parameter, ω_c , is directly related to the system response time, τ , which is approximately

$$\tau = \frac{1}{\omega_c} \quad (6)$$

The first thing to perform in the FR analysis is to obtain the open-loop transfer function of the system, $L(s)$. Then, the open-loop frequency response, $L(j\omega)$, is obtained by substituting "s" with "j ω ". Here, the main parameters to be considered for the system design are θ_m and ω_c . Finally, the system stability is guaranteed by satisfying the following restrictions:

$$|L(j\omega)| = 1 \text{ at } \omega = \omega_c \quad (7)$$

$$\theta_m = 180 + \arg[L(j\omega)] \quad (8)$$

In order to ensure the system stability for almost all the positioning control systems and to satisfy (7) and (8), there is a useful rule in the FR analysis. This rule establishes that phase margin values in the range of $30^\circ - 45^\circ$ provides a well-damped system response, higher values indicate larger overshoot and lower values indicate an under-damped system response.

2.2.1 Gain-phase margin tuning method

Consider the closed-loop system shown in Figure 3.



Figure 3. Block diagram of the elements for GPM tuning process.

Where:

$C(s)$, is the control law as stated in (3).

$D(s) = e^{-\frac{T_s}{2}}$, is the time delay associated with the sampling process of the digital controller.

K , is the actuator constant.

$G(s)$, is the plant model.

$L(s) = C(s)D(s)G(s)K$, is the open-loop transfer function.

Then, the tuning process consists of solving equations (7) and (8) for the system frequency response. Therefore, the general expressions for the GPM tuning method are:

$$|L(j\omega_c)| = \left| K_p + \frac{K_i}{j\omega_c} + K_d j\omega_c \right| \cdot 1 \cdot K \cdot A \quad (9)$$

$$\theta_m = 180^\circ + \text{Arg} \left[K_p + \frac{K_i}{j\omega_c} + K_d j\omega_c \right] - \frac{180\omega_c T}{2\pi} + \alpha \quad (10)$$

From equation (9), $A = |G(j\omega_c)|$, and the delay function has a magnitude of unity and hence has no effect in the equation. On the other hand, in equation (10), the phase shift associated with the delay is expressed as the term $(180\omega_c T) / (2\pi) \text{deg}$, and $\alpha = \text{Arg}[G(j\omega_c)]$. Now, it must be emphasized that the integrator effects mostly occur in the low frequency range, so it can be assumed that K_i has no effect on the magnitude. However, although the integrator makes the system response more precise it slightly degrades the system stability, so according to Tal in [42] the effect of K_i in the phase can be limited to a shift of -5° . Then, equation (9) and (10) can be rewritten as:

$$\left| K_p + K_d j\omega_c \right| \cdot A \cdot K = 1 \quad (11)$$

$$\theta_m = 180^\circ + \tan^{-1} \left(\frac{\omega_c K_d}{K_p} \right) - \frac{180\omega_c T}{2\pi} - 5^\circ + \alpha \quad (12)$$

Equation (11) is the relationship between the magnitudes of the PID controller and the plant model according to restriction (7), so a more useful expression can be defined by

$$\left| K_p + K_d j\omega_c \right| = \frac{1}{A \cdot K} \quad (13)$$

Equation (12) is the sum of all the contributed phases in the open-loop frequency response according to restriction (8). Now, β is defined as the contributed phase of the proportional and derivatives actions and is specified by

$$\tan^{-1} \left(\frac{\omega_c K_d}{K_p} \right) = \theta_m - 175^\circ + \frac{180\omega_c T}{2\pi} - \alpha = \beta \quad (14)$$

Therefore, the relationship between the PID controller gains and the plant model frequency response in the complex plane is shown in Figure 4.

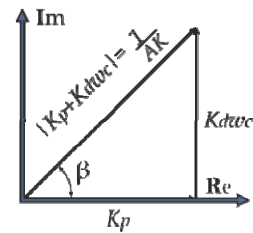


Figure 4. Frequency response of the plant and controller gains relationship.

Finally, the PID controller gains are obtained from the following expressions:

$$K_p = \frac{\cos(\beta)}{A \cdot K} \quad (15)$$

$$K_d = \frac{\sin(\beta)}{A \cdot K \cdot \omega_c} \quad (16)$$

$$K_i = \tan(5^\circ) \cdot K_p \cdot \omega_c \quad (17)$$

2.3 Genetic algorithms scheme

The Genetic Algorithm, GA, according to Rao in [29] is a powerful optimization searching technique based on the principles of natural genetics and natural selection. A flow chart of the general scheme of the implementation of the GA is shown in Figure 5.

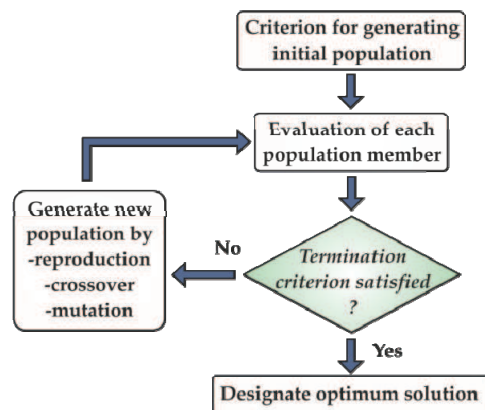


Figure 5. General flowchart of the Genetic Algorithm.

In the GA, normally the design variables, corresponding to the genomes in the natural genetic, are represented as binary strings and they are concatenated to form an individual, corresponding in turn to a chromosome in natural genetics. Other representations can be used. However, a binary representation is more adequate if an implementation in a digital system has to be carried out.

The main elements of natural genetics, according to Renner in [18], used for the searching procedure are:

- Reproduction
- Crossover
- Mutation

The *reproduction operation* consists of selecting individuals from the present population without changes to form part of the new population, in order to provide the possibility of survival for already developed fit solutions. Meanwhile, the *crossover operation* consists of creating new individuals (offspring) from the present individuals (parents), according to a crossover probability, C_p , by selecting one or more crossover points within the chromosome of each parent at the same place. Then, the parts delimited by these points are interchanged between the parents. On the other hand, the *mutation operation* makes modifications to a selected individual according to a mutation probability, M_p , by modifying one or more values in the binary representation. It is worth noting that if M_p is too large the GA is a purely stochastic search, but if M_p is too small it will be difficult to create population diversity in the GA.

In order to obtain the optimal solutions, the population of the GA must be evaluated. This evaluation consists of calculating the fitness value of each individual by minimizing or maximizing an objective function, J . The objective function is defined based on the optimization problem; the fitness function, f , is then specified by

$$f = 1 / J \quad (18)$$

The objective of the GA is to search for the global optimal solutions in the optimization problems. The fitness value is an indicator of the best and the worst individuals in the population. Therefore, the optimal solutions are obtained by selecting the best individuals in each generation of the optimization process; a generation is a complete cycle in which the population is generated, the genetic operators are applied and finally the population members are evaluated. The iterative cycles finish when the termination criterion is finally satisfied. There are many methods for selecting the best individuals according to their fitness value. For instance, the most popular methods applied in the GA are the proportional selection, the roulette wheel selection, the ranked selection and the tournament selection.

2.3.1 The micro genetic algorithm

The Micro-Genetic Algorithm (MGA) is a relatively new concept applied to improve the performance of Standard Genetic Algorithm (SGA). According to Coello in [23], the MGA refers to a GA with a very small population. This population is randomly generated and then the genetic

operators are applied until nominal convergence is reached; just at the moment when individuals have similar or identical characteristics. The best individual selected from the evaluated present population is transferred to the new population and the rest of the new population is randomly generated in order to inject population diversity. The MGA is applied to avoid the premature convergence and the stagnation problem of the SGA, as seen in [22]. Also, this concept can reduce the use of computational resources and increase the searching speed in software implementations. However, in some cases this technique does not guarantee a global optimal solution, but for applications such as that in the present work, this technique can provide appropriate local optimal solutions for the optimization problems which other deterministic methods find difficult in accordance with Hultmann in [39].

3. PID control frequency response tuning optimization based on Adaptive Micro-GA

In this section the methodology for tuning the optimization of PID controllers is described. As mentioned in the introduction section, the optimization approach couples the GPM method based on FR analysis and the AMGA technique.

3.1 Methodology description

Positioning control systems generally consist of three main blocks: the PC-software, the controller board and the positioning system. In Figure 6 the general scheme of the proposed methodology is shown in the PC-software block. In the following paragraphs the complete description of the proposed methodology is presented.

The PC-software block contains the system model identification, which is not the scope of this paper, but it is worth mentioning that the Least Squares Method (LSM) as presented in [3] is used to carry out the identification task. In order to obtain the plant model for each servo system mentioned in the introduction section, little random references are followed by the positioning system and the recollected data is employed in the LSM. Then, the plant model is provided in the discrete form as:

$$G(z) = \frac{nc_n \cdot z^n + nc_{n-1} \cdot z^{n-1} + \dots + nc_0}{dc_n \cdot z^n + dc_{n-1} \cdot z^{n-1} + \dots + dc_0} \quad (19)$$

The AMGA technique is implemented to optimize the input parameters of the GPM tuning method. Finally, when the optimization process ends, the output parameters of the GPM tuning are the optimized PID controller gains. The optimization process is implemented offline. The obtained parameters are sent to the controller board of the positioning system.

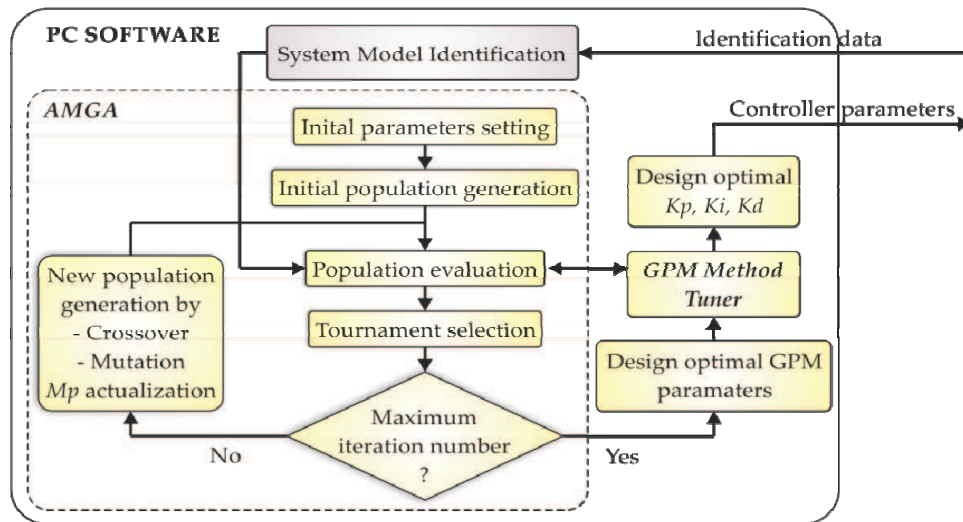


Figure 6. General diagram of proposed methodology; coupling GPM method and AMGA technique to optimize the controller gains.

The AMGA technique works as follows. First, the initial parameters are specified. In order to avoid infinitely cycled loops, the iterative optimization process has a termination criterion defined by a maximum number of generations, G . As the micro-population concept is applied for the GA, a small number of individuals have to be defined, $PSize$; in this work four individuals are generated. Besides, due to the micro-population concept, the crossover operation is applied to all the individuals by choosing two crossover points randomly. An adaptive mutation probability is used, so the initial mutation probability is also specified, Mp . The string length of each variable must be specified, Svl . Then, the string length of the chromosome, Scl , is the concatenation of each string variable. The searching space range must be defined for each variable, which are the minimum, v_{min} , and maximum, v_{max} , values that the corresponding design variable can assume. Finally, the weighted values, ω_i , of the objective function, J , are established.

The next step is to generate the initial population. Normally, when a large population size is defined, the initial population is generated randomly and the searching space distribution of the individuals is generally accepted. However, for this work a good distribution of the individuals must be ensured because a small population is used. Therefore, the individuals initially generated were distributed according to each one's variable range in a fixed way. Notice that the optimization variables (genomes) of the AMGA technique are τ and θ_m , so their defined ranges, shown in Table 2, are used to distribute the initial individuals. The next step in the optimization approach is the population evaluation. To calculate the fitness value of each individual, system response simulations are carried out. For this, the variables of the four individuals are decoded into their corresponding numerical values,

taking into account each variable range. The general expression for binary string decoding to its real value is:

$$V_{dec} = v_{min} + \frac{v_{max} - v_{min}}{2^{Svl} - 1} \sum_{i=0}^{Svl} 2^i V_{bin}^i \quad (20)$$

In which, V_{dec} is the real value of the optimization variable and V_{bin} is the corresponding bit value in the binary string representation. Then, the decoded values are used to simulate the system response in MATLAB as follows. The plant model and the PID control model are used to evaluate the closed-loop system response. However, the control model requires the PID gains in order to perform the simulation. For this reason, the "GPM Method Tuner" function is used. This function, according to equations (11) and (12), requires the magnitude, A , and the contributed phase of the plant model, α . In order to calculate these values for any order model plant, the following procedure is used. The complex form of z is specified by

$$z = Ae^{-j\omega} = A \cos(\omega) + jA \sin(\omega) \quad (21)$$

Therefore, the real and the imaginary components of the discrete transfer function specified by (19) can be obtained by applying:

$$re = \sum_{i=1}^{n+1} c_{n-i+1} \cdot \cos[(n-i+1) \cdot \omega] \quad (22)$$

$$im = \sum_{i=1}^{n+1} c_{n-i+1} \cdot \sin[(n-i+1) \cdot \omega] \quad (23)$$

Where n is the order of the numerator or denominator in the discrete transfer function, the coefficients c_{n-i+1} are specified by nc for the numerator and by dc for the denominator in expression (19). Then, the magnitude and the contributed phase of the plant can be obtained by

calculating the modulus and the inverse tangent using the real and the imaginary components, respectively. Finally, the PID controller gains are obtained by solving equations (15) to (17). When the model plant and the model of the PID controller are known, the step response of the closed-loop, as shown in Figure 1, is simulated. Furthermore, the system response performance criteria defined in this work are the Integral of the Squared Error (ISE), the Mean Squared Error (MSE) and maximum overshoot value (OS). The ISE and MSE are defined by

$$ISE = \int_0^{\infty} e^2(t)dt \quad (24)$$

$$MSE = \frac{\sigma^2}{n_e} \quad (25)$$

From equation (25) σ is the standard deviation of the error and n_e is the length of the error vector. The objective function, J , must be minimized in order to avoid larger overshoot values and to obtain satisfactory dynamics in the transient response. This function is then specified by

$$J = w_1 \cdot ISE + w_2 \cdot OS + w_3 \cdot MSE \quad (26)$$

From equation (26), w_1 , w_2 and w_3 are the weighted values used to enhance the corresponding performance criteria, ISE, MSE or OS. Then, the fitness function is specified by equation (18). The selection of the best individuals is the next step. The "tournament selection" is applied in this work. As the number of individuals in the population is small, the tournament selection is applied for all the individuals. The individual with the highest fitness value is selected as the best individual and the individual with the worst fitness value disappears.

After the selection of the best individuals, the termination criterion of the optimization process is evaluated. Therefore, if the maximum number of generations is reached, the AMGA function designates the optimized GPM parameters. Meanwhile, if the maximum number of generations is not reached yet, a new population is generated. The criterion for generating the new population is as follows. The worst individual will not appear in the new population. The best individual is selected and directly introduced into the new population. The rest of the offspring is generated by applying the crossover operation between the best individual and the two remaining individuals of the present population, and between the two remaining individuals only. The mutation operation is applied to the new population, except for the best individual, according to the adaptive mutation probability, Mp , in order to infuse material diversity. Three mutation points for each individual are selected for this process. When the AMGA starts, the Mp value is large in order to increase the searching speed, but when the generations elapse, this value is decreased to

avoid losing genetic information. The adaptive mutation probability is implemented according to the next expression.

$$Mp = 0.4 - [j \cdot (a/G)] \quad (27)$$

Where $0 < j \leq G$, is the current generation cycle. From several tests it was found that the value of a when proposed as 0.006 provides good results. When the optimization process ends, the optimal GPM parameters are designed. Then, the "GPM Method Tuner" function is called upon one last time to obtain the optimized PID control gains. The parameters defined for the AMGA technique are summarized in Table 2. Also, the parameters of the RGA optimization technique for the purposes of comparison are summarized in Table 3.

3.2 Simulation test

In order to validate the effectiveness of the optimization approach, a simulation test is performed in MATLAB. The results are compared with the classical tuning using the GPM method and with a PID optimization based on RGA as presented in [35]. The theoretical model used is that simplified by Tal in [42] for positioning systems:

$$G(s) = \frac{K_a K_t K_f}{Js^2} \quad (28)$$

AMGA parameters		Specified value
Maximum generations number, G		30
Variables number, Vn		2
Population size, $PSize$		4
Variable string length, Svl		16
Variable range [v_{min} v_{max}]	Response time, τ	[0.01, 1]
	Phase margin, θ_m	[30, 70]
Weighted value for ISE, w_1		0.3
Weighted value for OS, w_2		0.3
Weighted value for MSE, w_3		0.4

Table 2. Defined parameters for the AMGA technique.

RGA parameters		Specified value
Maximum generations number, G		100
Variables number, Vn		3
Population size, $PSize$		20
Variable range [v_{min} v_{max}]	Proportional gain, K_p	[0.01, 100]
	Integral gain, K_i	[0.01, 50]
	Derivative gain, K_d	[0.01, 300]
Weighted value for ISE, w_1		0.3
Weighted value for OS, w_2		0.3
Weighted value for MSE, w_3		0.4

Table 3. Defined parameters for the RGA technique.

Where, K_a is the amplifier gain with a specified value of 1 [A/A], K_t is the torque constant with a specified value of 0.2 [Nm/A], K_f is the position sensor gain with a specified value of 636 [counts/rad], and J_m is the inertia moment with a specified value of $1.1 \cdot 10^{-4}$ [kg.m²]. The results are shown in Figure 7.

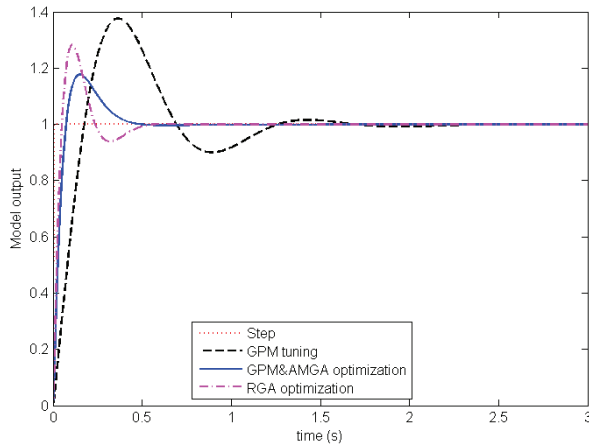


Figure 7. Simulation of the step response; comparing the optimization techniques with classical GPM tuning.

The features of the Laptop used for the implementation of the three methodologies are a 64-bits operative system, 8 GB of RAM, an Intel Core i7-6312QM CPU operating at 2.1 GHz. From Figure 7 it can be seen that the classical GPM tuning provides a good response in general for positioning systems, but the improvement in the system response is notoriously bad when using the optimization techniques. The GPM-AMGA optimization and the RGA optimization present a significant reduction in the overshoot when compared to the classical GPM tuning. However, the GPM-AMGA optimization reduces the overshoot even more when compared with the RGA optimization. The response time is also a significant improvement for both optimization techniques when compared with the classical GPM tuning. In this case the response times for the GPM-AMGA optimization and for the RGA optimization are very similar. The computing time required for the RGA optimization is 21.4786 seconds while the respective computing time required for the GPM-AMGA optimization is 1.3783 seconds.

4. Experimentation and results

The efficiency of the optimization approach is validated by experimental tests performed over three servo systems.

4.1 Experimental setup

The optimization approach was implemented using PC-software offline using MATLAB in function blocks. Three servo systems, shown in Figure 8, are used for the

experimental tests with the following characteristics. For the retrofitted CNC lathe the y -axis was selected to perform the tests. In this axis a servomotor 705-0-101 from EZG with a 4000 counts/rev incremental encoder is used. The servo motor is driven by a servoamplifier 413 from Copley Control Corp. In the CNC milling machine the z -axis was selected to perform the tests. The axis machine uses a brushless servo motor 3485-ME8137 from MCG with a 4000 counts/rev incremental encoder. The driver of the motor is a brushless PWM servo amplifier from Advanced Motion Controls. Finally, for the 6-DOF industrial PUMA robot the first robot link is selected to perform the tests. A standard servo motor with a 4000 counts/rev is used. The driver is a servoamplifier 403 from Copley Control Corp. The controller used in all the servo systems is a proprietary controller board based on a high-performance FPGA technology. This controller includes the PID control law and the necessary complementary modules; for specific details about the controller board see [43]. The plant models are obtained by using recollected data from the encoders of each servo system using the LSM approach, and for each real plant model the GPM-AMGA optimization, the RGA optimization and the classical GPM tuning method for the PID controller are applied. For the three tuning methods, ten motion tests are conducted. Each motion test consists of six references sent to the controller as step functions in order to determine the system dynamics. The motion commands for the CNC machines are 1000, 4000, 10000, 4000, 1000 and 0 counts for a total tracking of 20000 counts. In the case of the industrial robot the motion references are 100, 400, 1000, 400 100 and 0 counts for a total tracking of 2000 counts. The following subsection presents the results considering the high repeatability obtained.

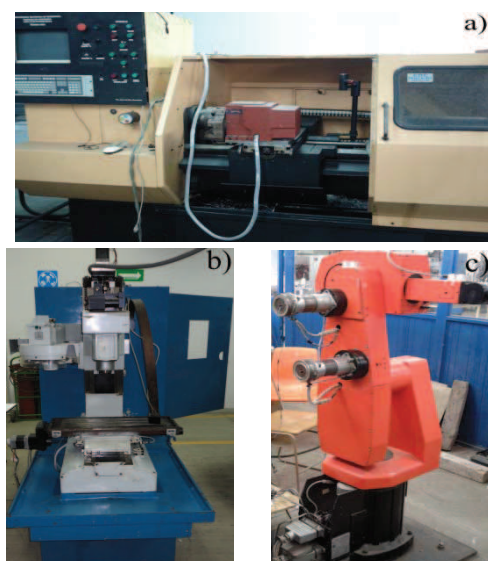


Figure 8. Servo systems used, a) CNC Lathe, b) CNC milling machine, and c) Industrial PUMA robot.

4.2 Study cases

The first case corresponds to the CNC lathe, for this system the identified plant model is specified by equation (29). The second case corresponds to the CNC milling machine and its corresponding plant model is specified by equation (30). Finally, the third case corresponds to the industrial PUMA robot; the plant model identified for this system is specified by equation (31).

$$G(z) = \frac{2.091e-007z^2 + 3.293e-007z + 4.06e-005}{z^2 - 1.78z + 0.78} \quad (29)$$

$$G(z) = \frac{2.98e-005z^2 + 0.000306z - 0.000233}{z^2 - 2z + 0.999} \quad (30)$$

$$G(z) = \frac{5.95e-005z^2 - 0.000104z + 6.98e-005}{z^2 - 1.96z + 0.956} \quad (31)$$

In Table 4, only the optimized parameters obtained from the GPM-AMGA optimization and the proposed parameters for the classical GPM tuning are presented. It must be clarified that the RGA optimization directly modifies the PID controller gains and for this reason there are no RGA parameter in Table 4. Meanwhile, the optimized parameters finally obtained by the two optimizations techniques and the parameters obtained by the classical tuning are presented in Table 5.

	GPM-AMGA Optimized parameters	GPM Proposed parameters
<i>Study case: CNC Lathe</i>		
τ	0.4011	0.2
θ_m	40.0073	45
<i>Study case: CNC Milling Machine</i>		
τ	0.2310	0.2
θ_m	46.3750	45
<i>Study case: Industrial PUMA Robot</i>		
τ	0.3538	0.25
θ_m	32.5024	45

Table 4. AMGA optimized parameters and user proposed parameters for each servo system.

The performances obtained from the experimental tests for the three servo systems are shown in Table 6. The performance criteria are the OS, the ISE and the MSE. In order to demonstrate the reduction in the computing time (increasing the searching speed), the time elapsed for the PID parameter optimization process of the RGA and the GPM-AMGA optimization techniques is presented in Table 7. The system response using the two optimization techniques and the classical GPM tuning is shown in Figure 9, for the CNC lathe, the CNC milling machine and the industrial PUMA robot, respectively. Finally, in Figures 10, 11 and 12 the respective input, the error signal

in counts, the output and the voltage command signal, at the PID controller are shown.

4.3 Results and discussion

The results obtained from the experimental tests show the GPM-AMGA optimization technique proposed provides excellent results when compared with the RGA optimization technique and with the classical GPM

	GPM-AMGA optimization	RGA optimization	Classical GPM tuning
<i>Study case: CNC Lathe</i>			
K_p	29.6787	97.0836	40.8838
K_i	40.6772	466.4310	112.3706
K_d	0.9632	4.2785	0.5439
<i>Study case: CNC Milling Machine</i>			
K_p	14.2863	49.5412	19.0010
K_i	33.9929	19.2312	58.0277
K_d	0.5365	0.9600	0.5052
<i>Study case: Industrial PUMA Robot</i>			
K_p	152.7212	99.5051	148.4444
K_i	237.2675	315.9438	326.4040
K_d	4.6012	4.9393	4.5883

Table 5. PID controller parameters obtained.

Tuning method	OS (%)	ISE Counts	MSE (%)
<i>CNC Lathe</i>			
GPM-AMGA	6.9	3.7470 x10 ⁹	0.6660
RGA	19.28	4.1936 x10 ⁹	0.7452
GPM	27.3	4.9102 x10 ⁹	0.8729
<i>CNC Milling Machine</i>			
GPM-AMGA	5.65	3.7436x10 ⁹	0.6649
RGA	29.26	3.6888 x10 ⁹	0.6553
GPM	13.0	4.3736e x10 ⁹	0.769
<i>Industrial PUMA Robot</i>			
GPM-AMGA	17	4.5668x10 ⁷	0.0081
RGA	18.16	4.7432x10 ⁷	0.0084
GPM	37.16	5.0649x10 ⁷	0.0090

Table 6. Percentage average of the ten motion tests for each servo system of the OS, the ISE and the MSE.

Optimization technique	Computing time (seconds)
<i>Study case: CNC Lathe</i>	
GPM-AMGA	3.138513
RGA	20.693711
<i>Study case: CNC Milling Machine</i>	
GPM-AMGA	2.972523
RGA	20.259184
<i>Study case: Industrial PUMA Robot</i>	
GPM-AMGA	1.722917
RGA	21.467128

Table 7. Computing time required for the PID parameters optimization.

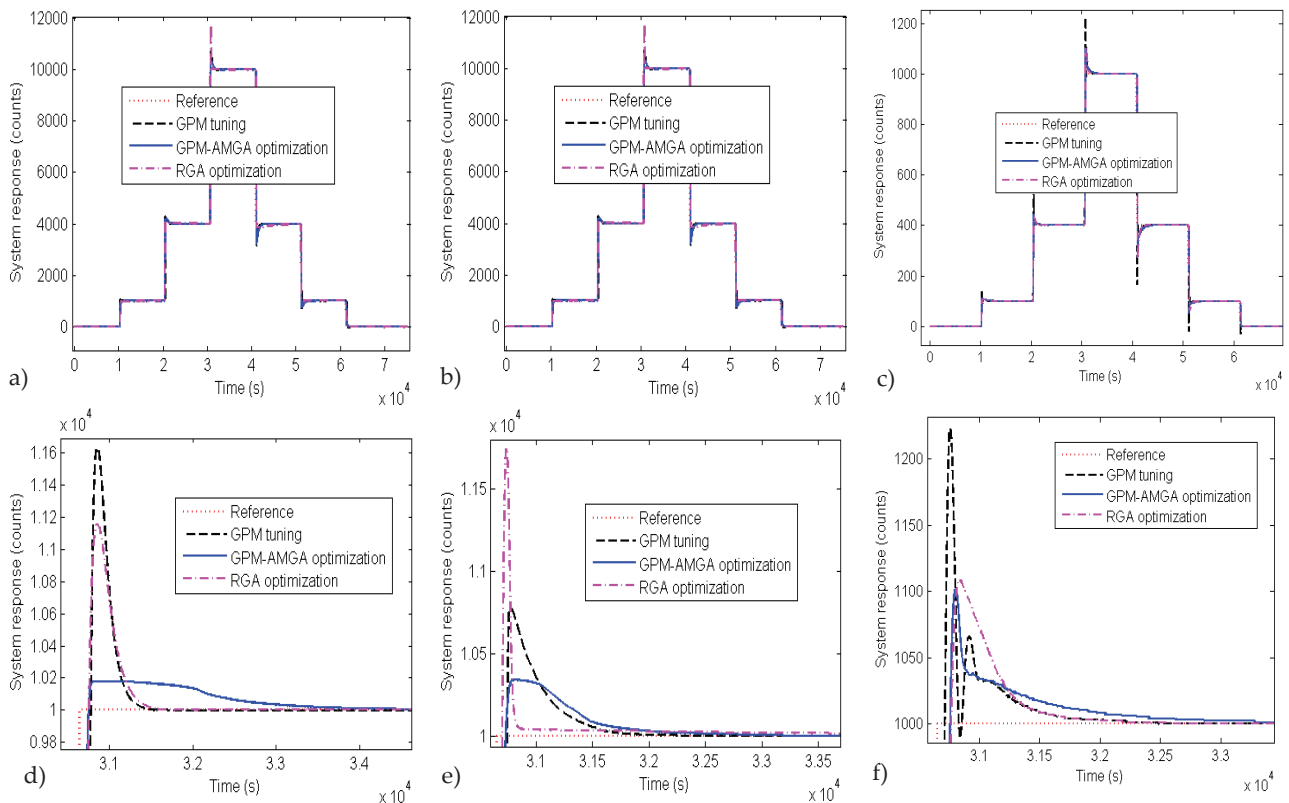


Figure 9. System response obtained, a) CNC lathe, b) CNC milling machine, c) industrial PUMA robot, d) zoom of the CNC lathe response, e) zoom on the CNC milling machine response, and f) zoom on the industrial PUMA robot response.

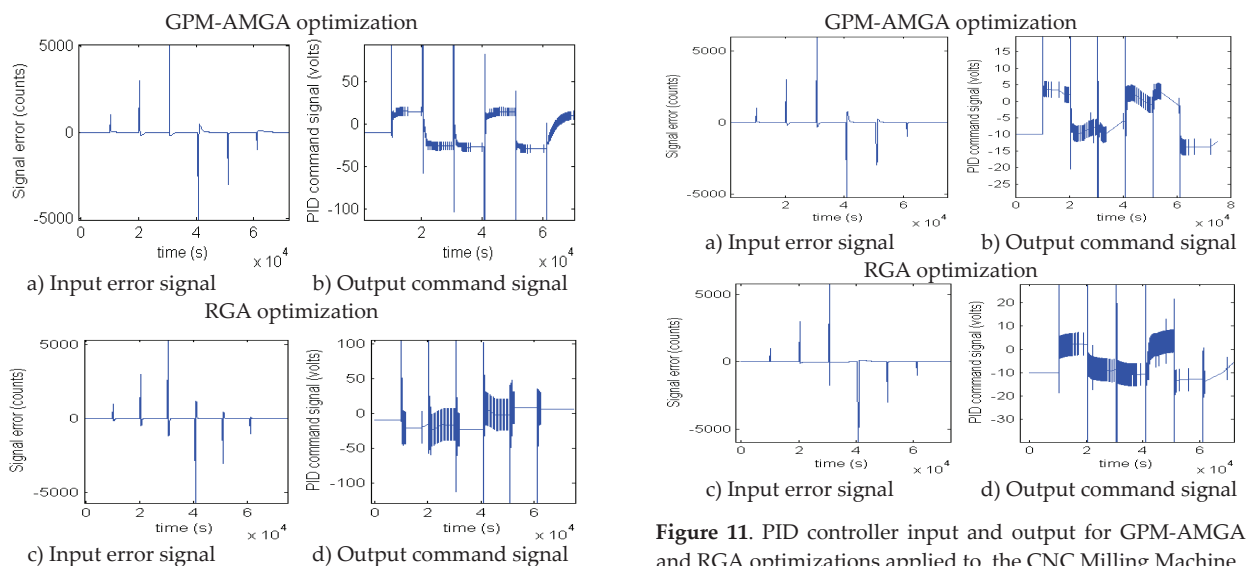


Figure 10. PID controller input and output for GPM-AMGA and RGA optimizations applied to the CNC Lathe.

method for tuning applications. The computational time and resources for the proposed optimization technique are significantly reduced compared with the RGA optimization. This affirmation is based on the characteristics from each optimization technique shown in Tables 2 and 3. For the RGA optimization technique the minimum value for the population size is 20 and the

Figure 11. PID controller input and output for GPM-AMGA and RGA optimizations applied to the CNC Milling Machine.

minimum amount of generations for the optimization process is 100 in order to reach nominal convergence. Even when the RGA algorithm is based on a floating-point representation of the individuals and using a crossover and mutation operations with floating-point values, such characteristics being appropriate for software applications, the computing time and the resources are much higher than those used in the AMGA algorithm. On the other hand, the AMGA optimization

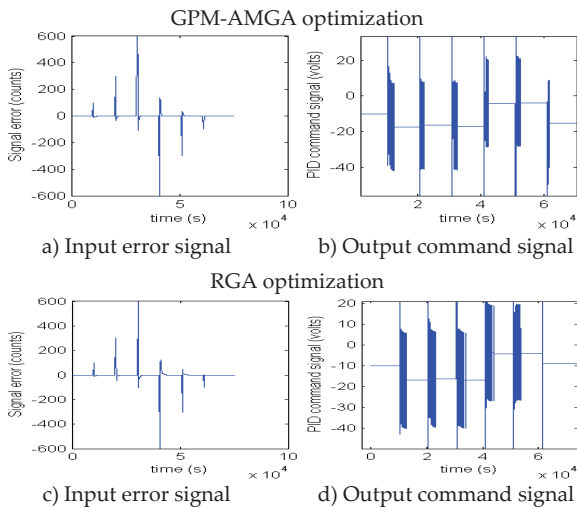


Figure 12. PID controller input and output for GPM-AMGA and RGA optimizations applied to the industrial PUMA robot.

algorithm requires a population size of four individuals, which means that only four evaluations are performed in order to obtain the fitness value of the population as opposed to the RGA where 20 evaluations are performed for the same goal. Besides, the minimum amount of generations needed in order to reach nominal convergence for the optimization process of the AMGA algorithm is 30. Therefore, the computational effort required for the AMGA algorithm is less in comparison with the RGA algorithm and the results shown in Table 7 confirm this affirmation. From Table 7, it can be observed that the time required for the AMGA algorithm is almost an order of magnitude less than the time required for a RGA algorithm in the PID parameters optimization. Meanwhile, in the classical GPM method the controller tuning task can take from several minutes to hours to find favourable controller parameters. For instance, the GPM parameters for the classical tuning were adjusted by hand several times in order to present a just comparison against the two optimization techniques. The GPM method was only implemented on software requiring only the proposed parameters as input. The optimized parameters of the AMGA algorithm and the proposed parameters for the classical GPM are shown in Table 4. In this table there is no RGA parameter because the RGA optimization algorithm modifies the PID controller gains directly. Based on the parameters of each tuning technique for the PID controller, shown in Table 5, the graphics of the system response of the CNC lathe, the CNC milling machine and the industrial PUMA robot are shown in Figures 9a, b, and c, respectively. In Figures 9d, e and f the zooms are shown for the larger motion reference of each servo system. The RGA algorithm does not consider the response time and the phase margin desired and the effects of this are observed in Figures 9d, e and f. Although the system response for the RGA optimization is improved, in general, large overshoots are seen. The GPM-AMGA optimization is based on the

frequency response analysis considering as primordial parameters for the system design the response speed and system stability. For this reason, the effects in the system response are observed as minimal overshoot values and quick response at the same time. It must be remarked that although many tuning methods claim to produce overshoot values within a specified range, in practical cases this is not totally true, such as the case of the ZN method. On the other hand, in Figures 10, 11 and 12 the input and output signals at the PID are observed controller only for the GPM-AMGA and RGA optimization techniques for the purposes of comparison. As can be observed the signal error for all the cases is very similar, a deep analysis shows that the error signal is lower for the optimization technique proposed. However, the output signal at the PID controller in the proposed methodology has lower oscillations than the RGA method, requiring less effort for the control action. Finally, in Table 6, the performance of the optimization techniques and the classical tuning method are summarized. From this table it can be observed that the GPM-AMGA optimization proposal produces the minimal overshoot value for all the case studies. Also, for the CNC lathe and the industrial robot the proposed optimization approach presents better results in MSE and ISE indexes than the RGA optimization. Only in the case of the CNC milling machine are the results obtained from the GPM-AMGA optimization very similar to the RGA optimization results. However, considering that the requirements of the positioning control systems in the manufacturing industry are the quality increase of the products, which means overshoot reduction and quick response, the GPM-AMGA technique can provide the expected solution.

5. Conclusions

A new methodology for PID control tuning is presented in this paper which combines the GPM tuning method, which is based on the FR analysis, and the AMGA optimization technique, applying the micro-population concept and the adaptive mutation probability. The use of the GPM method ensures some characteristics of the system response, such as speed response and system stability. The final outputs of the proposed methodology are the optimized PID controller parameters. These parameters are used in real positioning control systems, considerably reducing the overshoot value without compromising the speed response as opposed to the RGA optimization in which the desired overshoot is not considered, even when the error and response time is reduced. The GPM-AMGA optimization significantly reduces the computational resources and the computing effort when compared with the RGA, demonstrating the effectiveness of the proposed approach. From the experimental results it can be observed that even when the optimized parameters are not guaranteed to be the global optimal solutions, they are at least local optimal

solutions presenting an excellent performance for the positioning systems studied, improving the system behaviour satisfactorily. It must be emphasized that the classical GPM method for control tuning makes it harder and a time consuming task, taking several minutes in the best cases and up to hours in the worst cases. However, the proposed optimization methodology significantly reduces the time required for the controller tuning. Besides, this methodology is flexible in the sense that it can be applied to different positioning control systems. An interesting point is what happens if the parameters of the real system change. If these parameters change drastically after the offline tuning optimization the system could behave unexpectedly. However, in real systems the commercial controllers normally include tuning software which is applied at the beginning of large operation periods and then if system efficiency decreases because the system parameters change a re-tuning is made. The same situation applies for the optimization approach. Under normal operation conditions the parameters of the real system would change for many reasons (equipment changes, wearing, aging, etc); when these changes are detected a new identification and a tuning optimization process must be performed.

6. Acknowledgments

This project was partially supported by CONACyT scholarship 331141, FOFIUAQ-2012, and SEP-PROMEPE-PROMEPE-74/REV-04 projects.

7. References

- [1] Figuera, V. P., Optimization of Industrial Products and Processes, (2006) Barcelona, ISBN 84-96426-63-7.
- [2] Osornio-Rios, R. A., Romero-Troncoso, R. J., Herrera-Ruiz, G., Castañeda-Miranda, R., FPGA Implementation of Higher Degree Polynomial Acceleration Profiles for Peak Jerk Reduction in Servomotors, (2009) Robotics and Computer-Integrated Manufacturing, 25 (2), pp. 379-392.
- [3] Aguado-Behar, A., Topics on Identification and Adaptive Control (in Spanish), (2000) Book edited by ICIMAF. La Habana, Cuba.
- [4] Li, Y., Ang, K. H., Chong, G. C. Y., Patents, Software and Hardware for PID Control: An Overview and Analysis of the Current Art, (2006) IEEE Control Systems Magazine, 26 (1), pp. 42 – 54.
- [5] Sun, L., Mei, T., Yao, Y., Cai, L., Meng, M. Q.-H., PID Controller Based Adaptive GA and Neural Networks, (2006) Proceedings of the 6th World Congress on Intelligent Control and Automation, WCICA (2006), Dalian, China, 2, pp. 6564-6568.
- [6] Juang, Y. T., Chang, Y. T., Huang, C. P., Design of Fuzzy PID Controllers Using Modified Triangular Membership Functions, (2008) Information Sciences: an International Journal, 178 (5), pp. 1325-1333.
- [7] Osornio-Rios, R. A., Romero-Troncoso, R. J., Morales-Velazquez, L., Santiago-Perez, J. J., Rivera-Guillen, J. J., Rangel-Magdaleno, J. J., A Real-Time FPGA Based Platform for Applications in Mechatronics, (2008) IEEE International Conference on Reconfigurable Computing and FPGAs. ReConFig '08, DOI 10.1109/ReConFig.2008.55, pp. 289-294.
- [8] Osornio-Rios, R. A., Romero-Troncoso, R. J., Herrera-Ruiz, G., Castañeda-Miranda, R., Computationally Efficient Parametric Analysis of Discrete-Time Polynomial Based Acceleration-Deceleration Profile Generation for Industrial Robotics and CNC Machinery, (2007) Mechatronics, 17 (9), pp. 511 – 523.
- [9] Kermanshachi, Sh., Sadati, N., Genetic Multivariable PID Controller Based on IMC, (2007) Annual Meeting of the North American Fuzzy Information Processing Society, NAFIPS '07, pp. 174-177.
- [10] Tan, G., Jiang, B., Yang, L., A Novel Immune Genetic Algorithm-Based PID Controller Design and its Application to CIP-Intelligent Leg, (2007) IEEE Computer Society. 3rd International Conference on Natural Computation, ICNC 2007, 4, pp. 282-286.
- [11] O'Dwyer, A., Handbook of PI and PID Controller Tuning Rules, (2009) 3rd edition, Imperial College Press, ISBN 13 978-1- 84816-242-6.
- [12] Nise, N. S. Control System Engineering, (2006) 3rd edition, CECSA, ISBN 0-471-36601-3.
- [13] Ogata, K., Modern Control Engineering, (2003) 4th Edition, Pearson Prentice Hall, ISBN 84-205-3678-4.
- [14] Osornio-Rios, R. A., Romero-Troncoso, R. J., Herrera-Ruiz, G., Castañeda-Miranda, R., The Application of Reconfigurable Logic to High Speed CNC Milling Machines Controllers, (2008) Control Engineering Practice, 16 (6), pp. 674 – 684.
- [15] Rivera-Guillen, J. R., Romero-Troncoso, R. J., Osornio-Rios, R. A., Garcia-Perez, A., Herrera-Ruiz G., Design Methodology for Fully Dynamic-Controlled Polynomial Profiles and Reduced Tracking Error in CNC Machines, (2010) The International Journal of Advanced Manufacturing Technology, 51 (2010), pp. 723–737.
- [16] Galil Motion, DMC-18x2 user manual, Galil Motion Control Inc, (2005), www.galilmc.com.
- [17] Nagaraj, B., Muruganath, N., A Comparative Study of PID Controller Tuning Using GA, EP, PSO and ACO, (2010) IEEE International Conference on Communication Control and Computing Technologies, ICCCT, pp. 305-313.
- [18] Renner, G., Ekart, A., Genetic Algorithms in Computer Aided Design, (2003) Computer-Aided Design, 35 (8), pp. 709 – 726.
- [19] Abo-Hammour, Z. S., Alsmadi, O.-MK., Bataineh, S. I., Al-Omari, M. A., Affach, N., Continuous Genetic Algorithms for Collision-Free Cartesian Path Planning of Robot Manipulators, (2011) International Journal of Advanced Robotic Systems, 8 (6), pp. 14-36.

- [20] Ghanbari, A., Noorani, SMRS., Optimal Trajectory Planning Design of a Crawling Gait in a Robot Using Genetic Algorithm, (2011) *International Journal of Advanced Robotic Systems*, 8 (1), pp. 29-36.
- [21] Wang, X., Lu, T., Zhang, P., State Generation Method for Humanoid Motion Planning Based on Genetic Algorithm, (2012) *International Journal of Advanced Robotic Systems*, 9, pp. 1-8.
- [22] Dennis, B. H., Dulikravich, G. S., Optimization of Magneto-Hydrodynamic Control of Diffuser Flows Using Micro-Genetic Algorithms and Least-Squares Finite Elements, (2001) *Finite Elements in Analysis and Design*, 37 (5), pp. 349-363.
- [23] Coello-Coello, C. A., Toscano-Pulido, G., A Micro-Genetic Algorithm for Multiobjective Optimization, (2001) *Book Chapter, Lecture Notes in Computer Science*, 1993, pp. 126-140.
- [24] Atchiraju, Ch. S., Nagraja, T., A computer Software for PID Tuning by New Frequency Domain Design Method, (2003) *Computers and Electrical Engineering*, 29 (1), pp. 251-257.
- [25] Ma, M. D., Zhu, X. J., A Simple Auto-Tuner in Frequency Domain, (2006) *Computers and Chemical Engineering*, 30 (4), pp. 581-586.
- [26] Natarajan, K., Gilbert, A. F., Patel, B., Siddha, R., Frequency Response Adaptation of PI Controllers Based on Recursive Least-Squares Process Identification, (2006) *ISA Transactions*, 45 (4), pp. 517-528.
- [27] Rijlaarsdam, D., Nuij, P., Schoukens, J., Steinbuch, M., Frequency Domain Based Nonlinear Feed Forward Control Design for Friction Compensation, (2012) *Mechanical Systems and Signal Processing*, 27, pp. 551-562.
- [28] Morales-Velazquez, L., Romero-Troncoso, R. J., Osornio-Rios, R. A., Herrera-Ruiz, G. Santiago-Perez, J. J., Special Purpose Processor for Parameter Identification of CNC Second Order Servo Systems on a Low-Cost FPGA Platform, (2010) *Mechatronics*, 20, pp. 265-272.
- [29] Rao, S. S., *Engineering Optimization Theory and Practice*, (2009) 4th Edition, John Wiley & Sons Inc, ISBN 978-0-470-18352-6.
- [30] Lu, D., Zhu, L., Li, X., PID Parameter Optimization of Shunting and Winch Control System in Coal Transportation Based Online Adaptive Genetic Algorithm, (2010) *International Conference on E-Product E-Service and E-Entertainment, ICEEE*, pp. 1-4.
- [31] Weiping, Z., Dongzhou, Y., Zhanshuang, H., Parameters optimization for small helicopter highly controller based on genetic algorithm, 2012, *World Automation Congress (WAC)*, pp. 1 – 4.
- [32] Li, H. Y., The Adaptive Niche Genetic Algorithm for Optimum Design of PID Controller, (2007) *Proceedings of 6th International Conference on Machine Learning Cybernetics*, 1, pp. 487-491.
- [33] Wu, T., Cheng, Y., Tan, J., Zhou, T., The Application of Chaos Genetic Algorithm in the PID Parameter Optimization, (2008) *Proceedings of the 3rd International Conference on Intelligent System and Knowledge Engineering, ISKE 2008*, 1, pp. 230-234.
- [34] Nazir, A. J. A., Gautham., Surajan, R., Binu, L. S., A Simplified Genetic Algorithm for Online Tuning of PID Controller in LabView, (2009) *World Congress on Nature and Biologically Inspired Computing, NaBIC 2009*, pp. 1516-1519.
- [35] Asan M. K., Saravanakumar. G., Valamarthi. K., Devaraj. D., Radhakrishnan. T. K., Real-Coded Genetic Algorithm for System Identification and Tuning of a Modified Model Reference Adaptive Controller for a Hybrid Tank System, (2013) *Applied Mathematical Modelling*, 37, pp. 3829-3847.
- [36] Zhang, J., Zhuang, J., Du, H., Wang, S., Self-Organizing Genetic Algorithm Based Tuning of PID Controllers, (2009) *Information Sciences*, 179 (7), pp. 1007 – 1018.
- [37] Lin, G., Liu, G., Tuning PID Controller Using Adaptive Genetic Algorithms, (2010) *2010 5th International Conference on Computer Science and Education, ICCSE*, pp. 519-523.
- [38] Pan, I., Das, S., Gupta, A., Tuning of an Optimal Fuzzy PID Controller with Stochastic Algorithms for Networked Control Systems with Random Time Delay, (2011) *ISA Transactions*, 50 (1), pp. 28 – 36.
- [39] Hultmann A. H. V., do Santos C. L., Tuning of PID Controller Based on a Multiobjective Genetic Algorithm Applied to a Robotic Manipulator, (2012) *Expert Systems with Applications*, 39, pp. 8968-8974.
- [40] Firoozshahi A., Innovative Tank Management System based on DCS (2010), *IEEE, ELMAR 2010 Proceedings*, pp. 323-328.
- [41] Milosawlewitsch, M., Osornio-Rios, R. A., Romero-Troncoso, R. J., Model-Based Iterative Feedback Tuning for Industrial PID Controllers, (2010) *Journal of Scientific and Industrial Research*, 69 (12), pp. 930-936.
- [42] Tal, J., Step by Step Design of Motion Control Systems, (1994) *Galil Motion Control, Inc.*
- [43] Morales-Velazquez, L., Romero-Troncoso, R. J., Osornio-Rios, R. A., Herrera-Ruiz, G., Cabal-Yepez, E., Open-Architecture System Based on a Reconfigurable Hardware-Software Multi-Agent Platform for CNC Machines, (2010) *Journal of Systems Architecture*, 56 (9), pp. 407 – 418.