



Universidad Autónoma de Querétaro
Facultad de Ingeniería
Maestría en Instrumentación y Control Automático

Sintonización de un controlador PID basado en un algoritmo heurístico para el control de un Ball and Beam.

Tesis

Que como parte de los requisitos para obtener el Grado de Maestría en Ciencias en instrumentación y control automático

Presenta:

Salvador Ortiz Santos


Dirigido por:

Dr. Luis Alfonso Franco Gasca

Dr. Luis Alfonso Franco Gasca
Presidente


Firma

Dr. Roberto Valentín Carrillo Serrano
Secretario


Firma

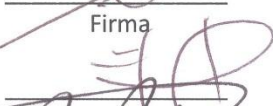
Dr. Manuel Toledano Ayala
Vocal



Firma


Dr. Gonzalo Macías Bobadilla
Suplente


Firma

Dr. Juvenal Rodríguez Reséndiz
Suplente


Firma


Dr. Aurelio Domínguez González
Director de la Facultad


Dr. Irineo Torres Pacheco
Director de Investigación y Posgrado

Centro Universitario
Querétaro, Qro.
Noviembre del 2014
México

Resumen

Sintonizar controladores de tipo Proporcional + Integral + Derivativo (PID) no es trivial y a pesar de que existen diversos estudios para sintonizar estos controladores parece no haberse resuelto el problema para tener una técnica que pueda ser implementado en los diversos sistemas, por ejemplo, los sintonizadores clásicos están limitados a sistemas que son estables en lazo abierto o los métodos inteligentes como las redes neuronales están limitados a los recursos computacionales. Por ello, en este trabajo se presentan los algoritmos heurísticos como herramientas para sintonizar controladores de tipo PID, donde estos algoritmos son robustos en el sentido de que pueden ser adaptados a diversos sistemas y que requieren recursos computacionales pequeños comparados con las redes neuronales. En primera etapa se presentan dos algoritmos heurísticos para sintonizar controladores de tipo PID: el algoritmo genético (AG) y la optimización por cúmulo de partículas (PSO) que de acuerdo a la literatura estos presentan resultados satisfactorios para sintonizar controladores PID. Para probar estos algoritmos se utilizó el sistema *Ball and Beam* que cuenta con las siguientes características: es un sistema subactuado, no-lineal, inestable en lazo abierto, cuenta con controladores en cascada y las acciones de control no son abruptas. Por estas características el sistema *Ball and Beam* es capaz de poner en manifiesto los algoritmos propuestos. Como segunda parte se implementa de forma experimental el algoritmo PSO que presentó resultados con mejores aptitudes que el AG. Posteriormente se presentan los resultados obtenidos de forma experimental.

Palabras Claves: Sintonización, controlador PID, AG, PSO, *Ball and Beam*.

Summary

Tuning of Proportional + Integral + Derivative controllers (PID) is not trivial and although there are several studies to tune these controllers, it seems not to have been solved the problem of a technique that can be implemented in several systems, eg. classical tuners are limited to systems which are stable in open loop or intelligent methods such as neural networks are limited to computing resources. Therefore, in this thesis, heuristic algorithms are presented as tools robust for tuning PID controllers in the sense that they can be adapted to various systems and require little computational resources compared to neural networks. In the first stage, two heuristic algorithms for tuning PID controllers are presented: the genetic algorithm (GA) and particle swarm optimization (PSO) which according to literature, both present satisfactory results for tuning PID controllers. To test these algorithms, the Ball and Beam system with the following characteristics was used: nonlinear, open-loop unstable and underactuated. This system has cascaded controllers and not abrupt control actions. For these features the Ball and Beam system is capable of manifesting the proposed algorithms. For second part of this work, the experimental implementation of the PSO algorithm is presented, showing better results than GA. Subsequently the experimental results are shown.

Key words: Tuning, PID controller, GA, PSO, Ball and Beam.

**A mis padres (Marina y Ricardo), a mis hermanos (Hilda, Adrian, Imeldo),
a mi familia, a mis amigos.
-Por hacer esto posible.
Gracias por todo.**

Dedicatoria especial a mi prima **Yesenia Santos Pérez**
que me dejó la mejor enseñanza de vida.
Querer y amar la vida.

Siempre vivirás en nuestros corazones.

Agradecimientos

Estoy muy contento de culminar con una maestría que siempre fue de exigencias. Por lo que le estoy muy agradecido a toda mi familia por ese gran apoyo que me brindaron y sé que seguirán apoyándome en las decisiones de mi vida.

Agradezco a mi país y al CONACYT por el apoyo que me brindaron durante estos dos años para poder realizar mis estudios y llevar a cabo este trabajo.

Gracias a mis asesores, al Dr. Luis Alfonso Franco por dirigir este trabajo, al Dr. Roberto Valentín Carrillo Serrano por las horas dedicadas para realizar este trabajo, facilitarme el prototipo experimental y dirigir proyectos de grupo que forman parte de mi experiencia profesional. Gracias a mis profesores por sus enseñanzas.

Por último y no menos importante quiero agradecer a todos mis amigos y compañeros por todos los momentos compartidos.

Índice general

Resumen.....	II
Summary	III
INTRODUCCIÓN	1
<i>DESCRIPCIÓN DEL PROBLEMA</i>	3
<i>JUSTIFICACIÓN</i>	4
<i>OBJETIVOS E HIPÓTESIS DEL TRABAJO</i>	6
<i>ESTADO DEL ARTE</i>	7
METODOLOGÍA	13
<i>SISTEMA BALL AND BEAM</i>	14
<i>MODELO MATEMÁTICO</i>	16
<i>Modelo del movimiento del balón</i>	17
<i>Modelo del motor eléctrico de DC y riel ó canal</i>	20
<i>DISEÑO DE CONTROLADORES</i>	23
<i>CONTROLADORES DEL BALL AND BEAM</i>	25
<i>Controlador PID</i>	25
<i>Controlador PD</i>	27
<i>TIPOS DE SINTONIZADORES</i>	29
<i>Ziegler-Nichols</i>	29
<i>Cohen-Coon</i>	31
<i>Otros</i>	32
<i>ALGORITMOS HEURÍSTICOS</i>	33
<i>Optimización mediante el cúmulo de partículas (PSO)</i>	33
<i>Algoritmos Genéticos</i>	37
DESARROLLO	42
<i>PARÁMETROS DEL SISTEMA BALL AND BEAM</i>	42
<i>Identificación de parámetros del balón</i>	46
<i>EL SISTEMA BALL AND BEAM EN TIEMPO DISCRETO</i>	48
<i>INSTRUMENTACIÓN DEL SISTEMA BALL AND BEAM</i>	49
<i>IMPLEMENTACIÓN DEL ALGORITMO HEURÍSTICO</i>	52

<i>Algoritmo PSO</i>	52
<i>Algoritmo Genético</i>	56
RESULTADOS	58
<i>RESULTADOS SIMULADOS</i>	58
<i>Simulación con el Algoritmo Genético</i>	58
<i>Simulación del algoritmo PSO</i>	61
<i>RESULTADOS EXPERIMENTALES</i>	66
<i>Resultados con controlador PD en cascada</i>	66
<i>Resultados del algoritmo PSO</i>	69
CONCLUSIÓN	73
<i>TRABAJOS A FUTURO</i>	74
APENDICE	75
<i>SISTEMA BALL AND BEAM EN TIEMPO DISCRETO</i>	75
<i>Motor de DC con carga en tiempo discreto</i>	77
<i>Modelo del balón en tiempo discreto</i>	80
<i>CÓDIGO DEL ALGORITMO GENÉTICO</i>	82
<i>CÓDIGO DEL ALGORITMO PSO</i>	84
<i>FUNCIÓN DE APTITUD O CALIDAD</i>	87
BIBLIOGRAFÍA.....	89

Índice de tablas

Tabla 2.1. Método de Ziegler-Nichols: Fórmulas para respuesta al escalón.....	30
Tabla 2.2. Método de Ziegler-Nichols: Fórmulas para respuesta en frecuencia.	31
Tabla 2.3. Método Cohen-coon.....	32
Tabla 2.4. Parámetros del algoritmo PSO.	36
Tabla 4.1. Resultados del algoritmo genético.....	65
Tabla 4.2. Resultados del algoritmo PSO.....	65

Índice de figuras

Figura 1: Sistema <i>Ball and Beam</i>	3
Figura 2: Sintonización de un controlador PID basado en un algoritmo genético e implementado en un FPGA (Luna-Ortiz, 2011).	8
Figura 3: Diagrama del controlador $PI\lambda D\mu$ sintonizado con el algoritmo PSO mejorado (Qu et al., 2010).	9
Figura 4: Sistema de control difuso basado en SFL (Duc-Hoang y Thai-Hoang 2008).	10
Figura 5: La arquitectura de un controlador PD en cascada (Sung-Kwun et al., 2009).	11
Figura 6: La arquitectura del controlador fuzzy en cascada (Sung-Kwun et al., 2009).	11
Figura 7: Diagrama de bloques de un sistema <i>Ball and Beam</i> con controladores PID en cascada (Asif-Rana et al., 2011).	12
Figura 8: Control por retroalimentación.....	13
Figura 9: Respuesta de un sistema a una entrada escalón.	14
Figura 10: El prototipo <i>Ball and Beam</i>	15
Figura 11: Movimientos angulares del riel para posicionar el balón en un punto deseado.....	15
Figura 12: Diagrama del sistema <i>Ball and Beam</i>	17
Figura 13: Variables del Balón.	18
Figura 14: Motor de DC con riel acoplada.....	20
Figura 15: Diagrama de bloques para el control del <i>Ball and Beam</i>	24
Figura 16: Método de Ziegler-Nichols: Respuesta al escalón.....	30
Figura 17: Método Cohen-Coon para sintonizar controladores PID.....	31
Figura 18: Optimización por cúmulo de partículas (PSO) en la naturaleza.	34
Figura 19: Movimiento de una partícula i en el espacio de soluciones.....	37
Figura 20: Algoritmo genético.	38
Figura 21: Diagrama del algoritmo genético.....	39
Figura 22: Control proporcional de posición para el riel.	43
Figura 23: Resultado experimental usando $kp = 27$	45
Figura 24: Identificación experimental del parámetro ρ	47
Figura 25: Sistema <i>Ball and Beam</i> en su representación s	48
Figura 26: Sistema <i>Ball and Beam</i> en su representación z	48
Figura 27: Diagrama esquemático para el control del sistema <i>Ball and Beam</i>	49
Figura 28: Sensor óptico para el riel (Encoder).	50
Figura 29: Diagrama del sensor de posición del balón.	51
Figura 30: Diagrama de flujo para el algoritmo PSO.....	55
Figura 31: Recombinación de los padres para crear hijos.....	56
Figura 32: Esquema del algoritmo genético en pseudocódigo.....	57
Figura 33: Realizado con el algoritmo genético a) Respuesta del balón con la sintonización del controlador PID b) respuesta del riel.....	59
Figura 34: Evolución de las ganancias de controlador PID con el algoritmo genético.....	60

Figura 35: Señal de control con el algoritmo genético.....	61
Figura 36: Realizado con el algoritmo PSO a) Respuesta del balón con la sintonización del controlador PID b) respuesta del riel.....	62
Figura 37: Evolución de las ganancias de controlador PID con el PSO.	63
Figura 38: Señal de control con el algoritmo PSO.....	64
Figura 39: Posición del balón con controladores PD.	66
Figura 40: Posición del riel con controladores PD.....	67
Figura 41: Saturación de la señal de control u1	68
Figura 42: Señal de control para el <i>Ball and Beam</i> generado por los controladores PD.....	68
Figura 43: Saturación de la señal de control u2	69
Figura 44: Posición del balón con la sintonización del controlador PID a través del PSO. Primera prueba.....	70
Figura 45: Posición del riel con algoritmo PSO. Primera prueba.	70
Figura 46: Señal de control con algoritmo PSO. Primera prueba.	71
Figura 47: Posición del balón con la sintonización del controlador PID a través del PSO. Segunda prueba.....	72
Figura 48: Posición del riel con algoritmo PSO. Segunda prueba.	72
Figura 49: Señal de control con algoritmo PSO. Segunda prueba.	72
Figura 50: Muestreador y retenedor de orden cero.	75
Figura 51: Función del muestreador y retenedor de orden cero.....	76
Figura 52: Modelo matemático del muestreador y retenedor de orden cero.	76
Figura 53: Diagrama de un motor de DC con retenedor de orden cero y muestreo de 10ms	77
Figura 54: Diagrama del modelo del balón con retenedor de orden cero y muestreo de 10ms	80

INTRODUCCIÓN

Diseñar un controlador para un sistema o proceso es cada vez más estricto, sería muy ambicioso pretender crear un controlador que presente resultados aceptables para todos los sistemas. Sin embargo, el concepto de retroalimentación ha revolucionado la manera en como se realiza el control. Donde el controlador Proporcional + Integral + Derivativo (PID) es por mucho la forma más utilizada de control por retroalimentación. Debido a las características que este tipo de controlador presenta. El controlador PID no es el más adecuado para ciertos sistemas. Sin embargo, es el controlador preferido por las industrias (Aström and Hägglund, 2001).

El controlador PID tiene la característica de poseer tres ganancias (proporcional, integral y derivativo). A pesar de contar con esta característica resulta que existen diferentes maneras de procesar la señal de error, es decir, existen diversas estructuras de PID, que se pueden utilizar dependiendo del proceso o sistema en el que se desee implementar.

No importa que estructura de PID se haya elegido, resulta ser que el controlador debe ser sintonizado, es decir, se deben seleccionar las ganancias del controlador. La sintonización de los controladores no es de forma trivial y define en gran medida el desempeño del sistema en general. Para ello existen diversas maneras de sintonizar un controlador PID, en muchos de los casos toman en cuenta aspectos del controlador y del proceso. A pesar de que se han reportado diversos métodos de sintonización en la literatura, en la industria esto parece no haber dado respuesta a sus necesidades.

Las técnicas de sintonización convencionales funcionan bien en casos particulares, tal como, en sistemas donde son estables en lazo abierto. Actualmente se estudia diversas técnicas de sintonización para ampliar el número de procesos en los que se sintoniza el controlador PID, tal como el método adaptativo o la herramienta de optimización conocidos comúnmente como algoritmos heurísticos.

Dentro de los algoritmos heurísticos se encuentra el algoritmo genético (AG) así como la optimización por cumulo de partículas (PSO) estos algoritmos se utilizaron en este trabajo para sintonizar el controlador PID. Donde la idea general de estas técnicas es encontrar una solución a un problema a través de la optimización de un criterio específico, esto se hace utilizando un espacio de búsqueda que se define en un principio y que va evolucionando conforme el algoritmo avanza. Estas técnicas imitan total o parcialmente el comportamiento de algún fenómeno que ocurre en la naturaleza (Glover y Kochenberger, 2003).

Resulta que los algoritmos heurísticos presentan resultados satisfactorios. Sin embargo, éstos no pueden asegurar que la solución encontrada sea la óptima, debido a que el espacio de búsqueda es sobre un conjunto aleatorio. Por ello, los algoritmos heurísticos encuentran resultados de manera rápida comparada con las redes neuronales donde su búsqueda es de manera exhaustiva. Además de ser rápidos los algoritmos heurísticos requieren pocos recursos computacionales. Por lo tanto, los algoritmos heurísticos pueden ser implementados sobre algún dispositivo electrónico, tal como, micro-controladores, FPGA (del inglés *Field Programmable Gate Array*), un procesador digital de señales (DSP, siglas en inglés de *Digital Signal Processor*), entre otros.

Antes de continuar es importante mencionar el sistema *Ball and Beam* (este nombre es como usualmente se conoce el sistema) que fue elegida para implementar un controlador PID y sintonizar éste controlador con un algoritmo heurístico.

El sistema *Ball and Beam* consiste en posicionar un balón en una posición deseada sobre un riel cambiando el ángulo de ésta. La acción de control se consigue a través de un motor de DC. Por otro lado, el *Ball and Beam* se considera como un sistema sub-actuado debido a que presenta más grados de libertad que actuadores además de ser un sistema no lineal (Barbu et al., 1997). Siendo así, el sistema *Ball and Beam* es una excelente herramienta para probar o validar diferentes teorías o algoritmos de control. El sistema *Ball and Beam* se muestra en la figura 1.

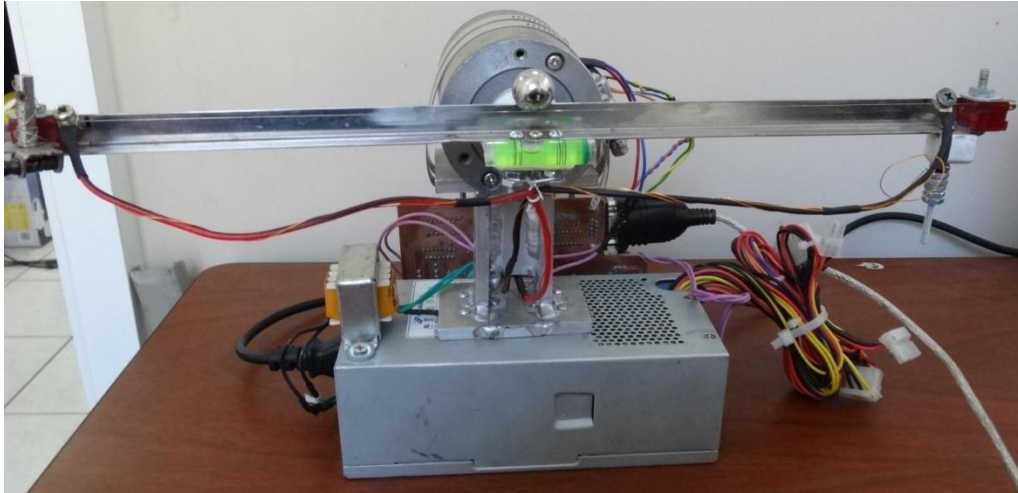


Figura 1: Sistema *Ball and Beam*.

DESCRIPCIÓN DEL PROBLEMA

En los procesos industriales se busca siempre tener una respuesta de manera eficiente, Por ende, existen diversos estudios que buscan mejorar la eficiencia de éstos y como una consecuencia se han diseñado diversos tipos de controladores.

A pesar de que existen una variedad de controladores, el controlador más utilizado en la industria es el controlador de tipo PID por ser versátil, contar con una simple arquitectura y además es uno de los controladores que no requiere el conocimiento de los parámetros del sistema. Cabe mencionar, que este controlador no siempre es la solución más adecuada sin embargo es la más utilizada en la industria.

Una de las características principales de los controladores es ajustar las ganancias (seleccionar valores adecuados), debido a que el desempeño del sistema depende del ajuste de ganancias, a este proceso también se le conoce como sintonización del controlador.

Sintonizar los controladores no es trivial. Sin embargo, existen diversos estudios para sintonizar controladores, pero al parecer éstos no están teniendo un impacto en la industria o se encuentran limitados al tipo de sistema. Por ejemplo, los métodos de sintonización clásicos para el controlador PID están limitados a sistemas que son estables

en lazo abierto. Por otra parte los métodos de sintonización inteligentes como las redes neuronales requieren de grandes recursos computacionales en comparación con algún algoritmo heurístico.

Resulta que el sistema *Ball and Beam* es un sistema que no es estable en lazo abierto además de ser un sistema no lineal. Por ello, los métodos de sintonización clásicos no son aplicables a este sistema. También posee dos controladores en cascada maestro y esclavo. De tal manera que los controlares son dependientes para estabilizar dicho sistema, es decir, para los valores de las ganancias del controlador primario existen ciertos valores de ganancias del controlador secundario adecuados para el buen desempeño del sistema *Ball and Beam*.

JUSTIFICACIÓN

En los procesos industriales, más del 95% de los controladores dedicados a la regulación son de tipo PID. Hace más de 60 años que se publicó la regla de sintonización de Ziegler - Nichols y desde entonces se han publicado numerosos métodos de sintonización. Por ello, se pensaría que la sintonización de los controladores PID presentan resultados satisfactorios en la industria, desafortunadamente este no es el caso (Cheng – Ching, 2006; Santos-Coelho y Wichhoff-Pessôa, 2011; Hultmann-Ayala y Santos-Coelho, 2012). Las siguientes cifras que señala Cheng – Ching (2006) nos dan un panorama general de la situación que se vive en la industria.

- Sólo el 20% de los controladores PID en los procesos industriales presentan un desempeño satisfactorio.
- El 30% tienen un desempeño pobre a causa de una sintonización deficiente del controlador.
- El 30% tienen un desempeño pobre a causa de los problemas que se presentan en los sistemas electrónicos o mecánicos.
- El otro 20% presentan un desempeño pobre a causa de un mal diseño en el controlador al no ser el adecuado a la aplicación.

Las cifras anteriores nos motivan a realizar un trabajo que permita contribuir con la solución al problema de la sintonización de los controladores de tipo PID, debido a que es de gran utilidad en los procesos industriales. Por otro lado, es claro que las muchas reglas de sintonización ya propuestas en la literatura no están teniendo un impacto en la industria.

En los últimos años, en la comunidad internacional de la inteligencia artificial ha surgido el interés por desarrollar algoritmos para resolver problemas de optimización. Tales como los algoritmos heurísticos que tiene la propiedad de ser robustos, en el sentido que estos algoritmos pueden trasladarse de un problema a otro. Esto debido a que son habitualmente fáciles de implementar.

De acuerdo a la literatura el algoritmo genético y la optimización por cumulo de partículas (que pertenecen a los algoritmos heurísticos) son herramientas de optimización que presentan resultados satisfactorios para sintonizar controladores de tipo PID (Iruthayarajan y Baskar, 2009; Sung-Kwun et al., 2009; Luna-Ortiz, 2011; Hultmann-Ayala y Santo-Coelho, 2012). Por lo tanto, en este trabajo se presentan estas herramientas para sintonizar el controlador PID del sistema *Ball and Beam*.

Para validar el algoritmo heurístico se utilizó el sistema *Ball and Beam* debido a que presenta una dinámica compleja, por ello, controlar este sistema resulta tener una alta dificultad, de tal manera que pone en manifiesto los algoritmos propuestos.

OBJETIVOS E HIPÓTESIS DEL TRABAJO

Hipótesis General

Es posible diseñar e implementar un algoritmo heurístico para la sintonización de un controlador PID, reduciendo el tiempo de cómputo de sistemas análogos basados en software y que permita controlar un *Ball and Beam* en el modo en línea.

Objetivo general

Diseñar un sintonizador del controlador PID basado en un algoritmo heurístico para el control de un *Ball and Beam* en el modo en línea.

Objetivos particulares

- Diseñar un algoritmo heurístico que sea capaz de sintonizar un controlador PID para un *Ball and Beam*.
- Simular el algoritmo heurístico en software para validar su funcionamiento.
- Validar el funcionamiento del algoritmo heurístico en el sistema *Ball and Beam* de manera experimental.

ESTADO DEL ARTE

Como ya se ha mencionado existen diversos tipos de sintonizadores para los controladores de tipo PID. En esta sección se presenta los antecedentes de sintonizadores PID basadas en algoritmos heurísticos así como trabajos realizados con el sistema *Ball and Beam*.

En el trabajo de Wei (2007) propone una modificación de la formula de recombinación del algoritmo genético y este método se usa para determinar las ganancias de los controladores PID en procesos multivariables, es decir, múltiples-entradas múltiples-salidas (MIMO, por sus siglas en inglés, *Multiple-Input Multiple-Output*). Donde nos asegura que el algoritmo genético es uno de los métodos convincentes y de búsquedas optimas para la optimización y la solución de problemas de control. El cruzamiento o recombinación clásico en los algoritmos genéticos se hace a través de dos cromosomas, en este artículo se propone que la recombinación sea con más cromosomas, en su ejemplo usa tres cromosomas, a este proceso lo llama multi-cruce.

Por otra parte, en el artículo de Willjuice-Iruthayarajan y Baskar (2009) presentan una comparación del desempeño de los algoritmos evolutivos para la sintonización de los controladores PIs y PIDs multi-variables, tales algoritmos son: algoritmos genéticos de código real (RGA), optimización por cúmulo de partículas modificado (MPSO), la adaptación de la matriz de covarianza con estrategia de evolución (CMAES) y evolución diferencial (DE). A través de las simulaciones realizadas revelan que los cuatro algoritmos considerados son adecuados para la sintonización del controlador PID en modo fuera de línea. Sin embargo, sólo los algoritmos CMAES y MPSO son adecuados para la sintonización en línea del controlador PID.

En el siguiente artículo se hace la combinación de los algoritmos genéticos (GA) con el controlador PID convencional para encontrar automáticamente los parámetros del controlador. Esto se lleva a cabo mediante el auto – aprobación, auto – organización y auto – aprendizaje habilidades de los algoritmos inteligentes (Yajuan y Qinghai, 2011).

En la Universidad Autónoma de Querétaro también se han realizado trabajos para sintonizar controladores de tipo PID. Es el caso del trabajo de tesis de Luna-Ortiz (2011) que lleva como nombre: “Diseño e implementación de un algoritmo genético en FPGA para la sintonización de controladores PID”. En este trabajo se implementó un controlador PID para controlar la velocidad de un motor de DC, donde el controlador fue sintonizado con un algoritmo genético e implementado en un FPGA como se muestra en la figura 2.

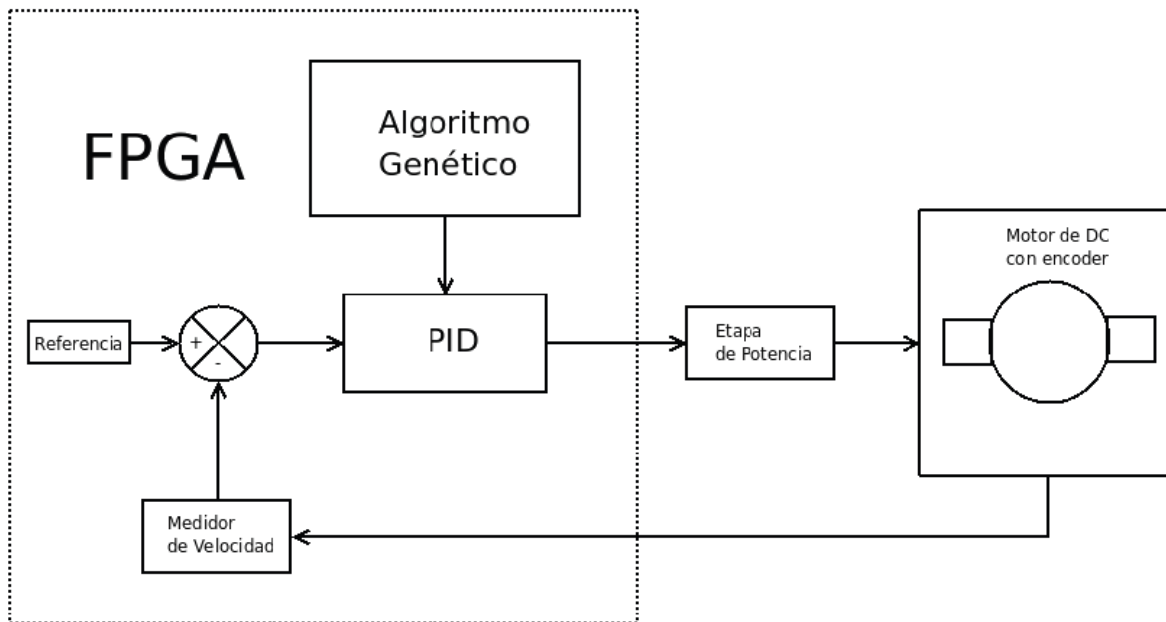


Figura 2: Sintonización de un controlador PID basado en un algoritmo genético e implementado en un FPGA (Luna-Ortiz, 2011).

Otro de los algoritmos heurísticos que se emplea ampliamente para optimizar sistemas es el caso de la optimización por cúmulo de partículas (PSO).

En el trabajo de Qu et al., (2010) se modifica la optimización por cúmulo de partículas para sintonizar controladores PID de orden fraccional y se implementa en un FPGA (figura 3). La modificación consiste en cambiar la manera de calcular la velocidad de vuelo de cada partícula con el objetivo de prevenir una convergencia prematura. De acuerdo a los resultados de este método implementado en un FPGA tiene algunas ventajas tales como: diseño flexible, sintonización en línea, alta confiabilidad y altas velocidades de respuesta. Cabe mencionar que se simuló con un sistema de segundo orden.

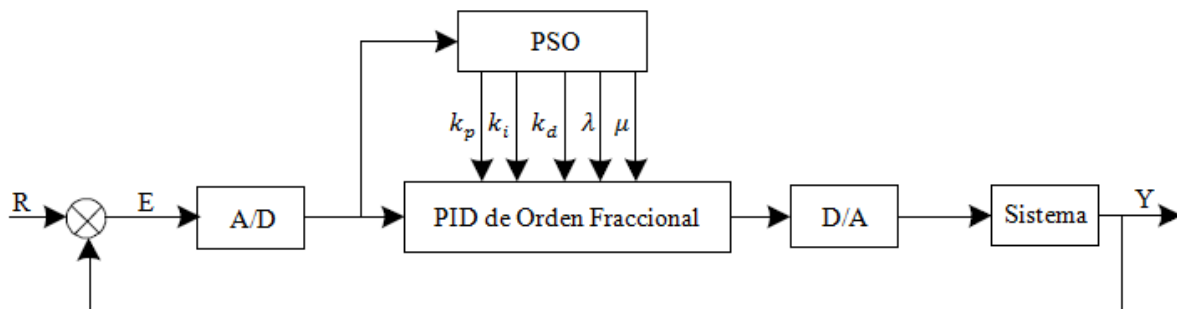


Figura 3: Diagrama del controlador $PI^\lambda D^\mu$ sintonizado con el algoritmo PSO mejorado (Qu et al., 2010).

También en el artículo de Fang et al. (2011) presentan una modificación de la manera en como se calcula la velocidad de vuelo de cada partícula del algoritmo PSO. La optimización por cúmulo de partículas modificado sintoniza un controlador PID para un regulador de turbina de agua.

A continuación se presentan algunos trabajos realizados con el sistema *Ball and Beam*.

En el trabajo de Duc-Hoang y Thai-Hoang (2008) sintonizan los parámetros del controlador por lógica difusa que estabiliza el sistema *Ball and Beam*, esto utilizando el algoritmo arrastrada salto de rana (SFLA por sus siglas en inglés, Shuffled Frog Leaping Algorithm) como se muestra en la figura 4. El SFLA es un método de búsqueda meta-heurística inspirada en la evolución mimética de un grupo de ranas en la búsqueda de alimentos. Este método se compara con un controlador lineal cuadrático (LQR por sus siglas en inglés, *Lineal-Quadratic Regulator*).

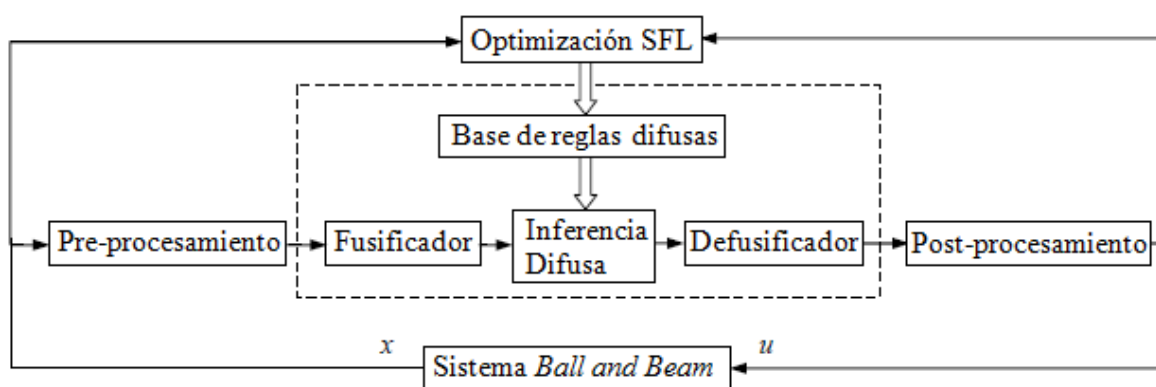


Figura 4: Sistema de control difuso basado en SFL (Duc-Hoang y Thai-Hoang 2008).

Por otra parte, en el artículo de Sung-Kwun et al. (2009) proponen controladores difusos en cascada para estabilizar el sistema *Ball and Beam* y es comparado con controladores PD's. Donde la sintonización de los parámetros de estos controladores se realizó mediante el algoritmo genético basado en la competencia leal jerárquica (HFCGA por sus siglas en inglés, *hierarchical fair competition-based genetic algorithm*) como se muestra en la figura 5 y 6. De acuerdo a las simulaciones presentados los controladores difusos en cascada tienen un desempeño aún más eficiente que los controladores PD's para el sistema *Ball and Beam*.

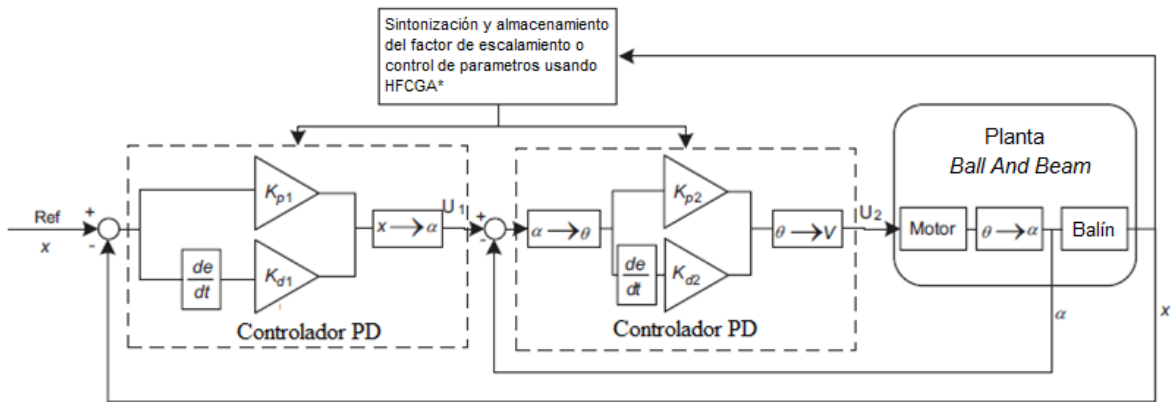


Figura 5: La arquitectura de un controlador PD en cascada (Sung-Kwun et al., 2009).

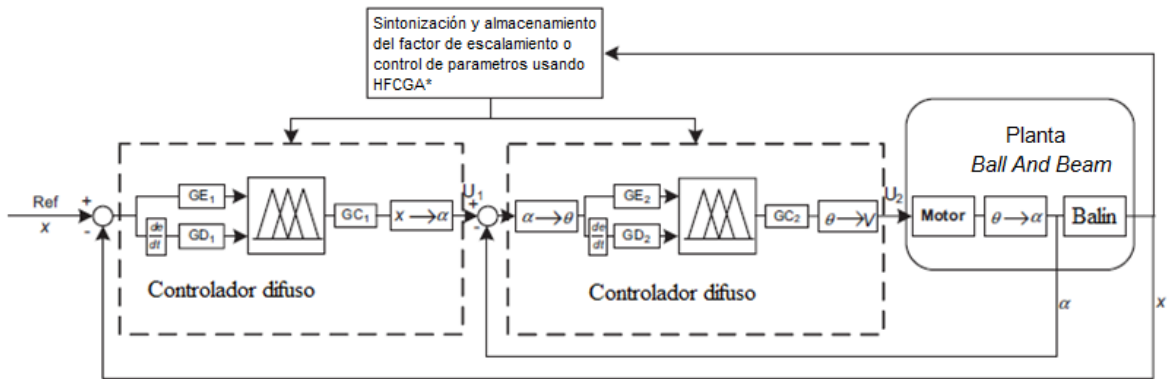


Figura 6: La arquitectura del controlador fuzzy en cascada (Sung-Kwun et al., 2009).

Nganga y Okou (2011) proponen un controlador por retro-alimentación de estados adaptativo para el sistema *Ball and Beam* y es comparado con el controlador Backstepping adaptativo. Los resultados de acuerdo a las simulaciones hechas muestran que el controlador Backstepping tiene un desempeño superior si es implementado al sistema linealizado. Sin embargo, este tiene un desempeño pobre si es aplicado al sistema no lineal. También los resultados muestran que el desempeño del controlador por retro-alimentación de estados adaptativo es independiente del modelo usado, es decir, el controlador tiene un buen desempeño en ambos sistemas: lineal o no lineal.

Posteriormente, en el trabajo de Asif-Rana et al. (2011) la optimización por cúmulo de partículas se presentó como una técnica de optimización altamente útil y robusta para sintonizar las ganancias de los controladores PID's de un sistema *Ball and Beam* clásico (figura 7). Los resultados del PSO son comparados con el controlador por lógica difusa y con la respuesta escalón optimizado a través de las ecuaciones ITAE (por sus siglas en inglés, *Integral Time Absolute Error*). Donde el PSO presentó ventajas en el desempeño del sistema *Ball and Beam* comparadas con las dos técnicas anteriormente mencionados.

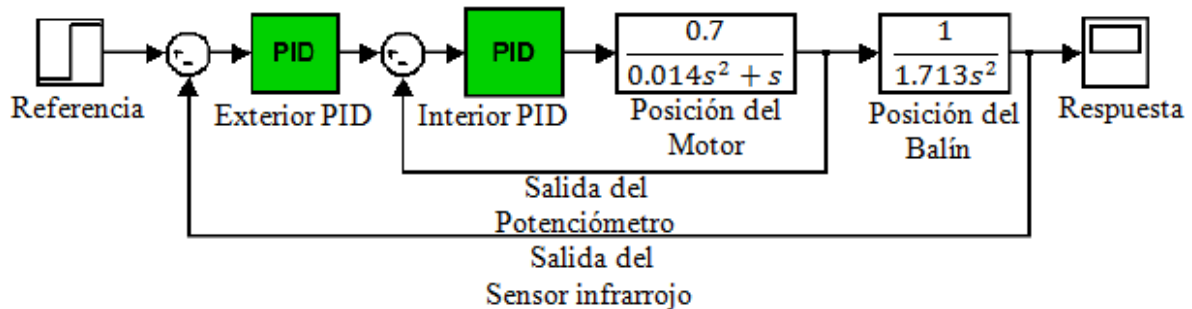


Figura 7: Diagrama de bloques de un sistema *Ball and Beam* con controladores PID en cascada (Asif-Rana et al., 2011).

METODOLOGÍA

Es fácil pensar que los sistemas respondan siempre de la misma manera, es decir, que para una cierta entrada exista una única salida en todo un intervalo de tiempo muy grande, en la realidad esto no existe. Por ello se han diseñado un número importante de controladores, donde es evidente pedirle que este mantenga una salida deseada a pesar de que en el sistema puedan existir perturbaciones o cambios dinámicos. En la figura 8 se muestra un lazo de control por retroalimentación, donde el controlador se le pide estabilizar el sistema o planta aún con la presencia de perturbaciones.

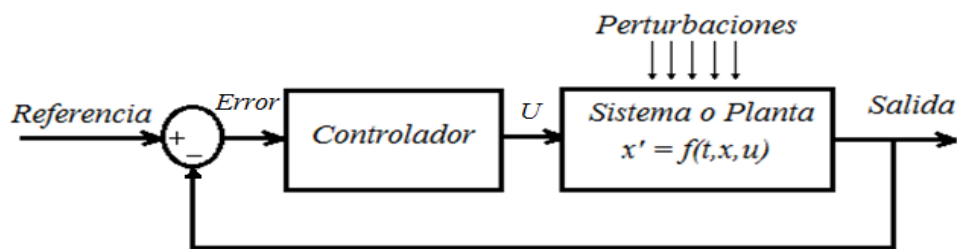


Figura 8: Control por retroalimentación.

Para un controlador no es suficiente con mantener la respuesta deseada, también se pide que éste converja en un tiempo finito, otra manera de ver este acontecimiento, es llevar el error a cero o muy cercano a cero en un lapso muy corto de tiempo sin tener grandes sobrepasos en la respuesta. Este acontecimiento se muestra en la figura 9.

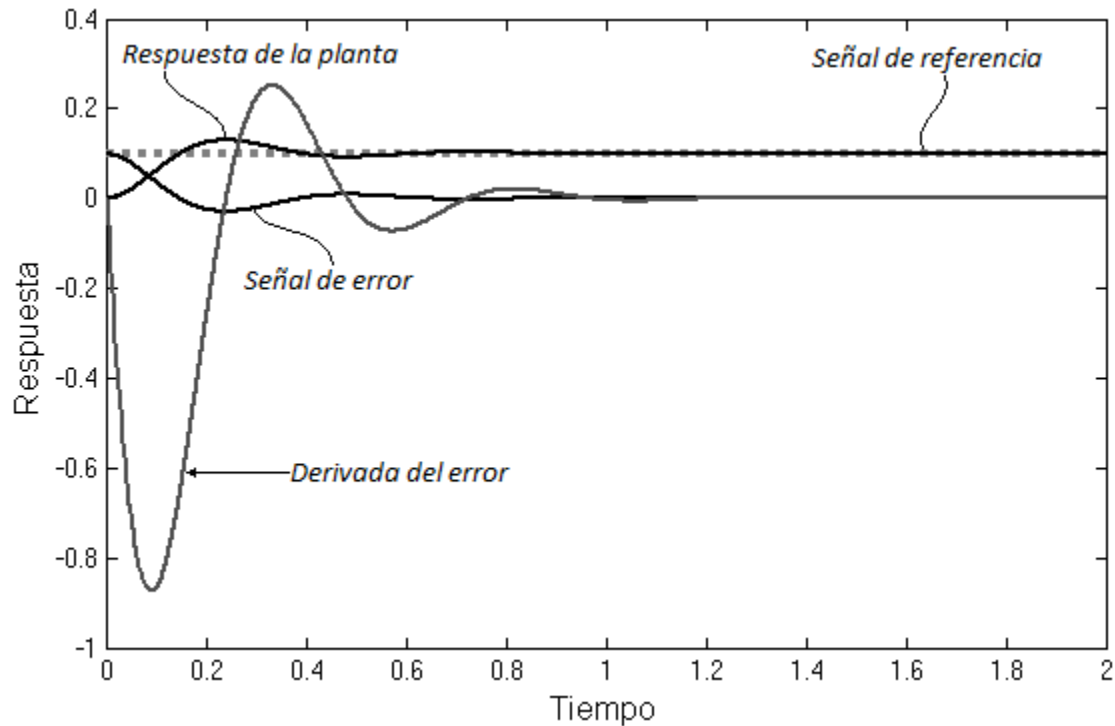


Figura 9: Respuesta de un sistema a una entrada escalón.

A continuación se presenta el sistema *Ball and Beam*, posteriormente los controladores que se usan para estabilizar este sistema.

SISTEMA BALL AND BEAM

El sistema *Ball and Beam* (figura 10) como es conocido generalmente y que tiene una traducción directa al español como balón y riel, trata precisamente de un balón que viaja a lo largo de un riel, donde el objetivo es posicionar el balón en un punto deseado sobre el riel y éste tratar de permanecer en su posición horizontal. Cabe mencionar que el *Ball and Beam* es un sistema no lineal y que es difícil de controlar debido a su naturaleza. Además es un sistema sub-actuado, se dice que es sub-actuado ya que cuenta con más grados de libertad que de actuadores. El *Ball and Beam* es un prototipo que sirve para probar leyes o técnicas de control (Nganga-Kouya y Okou, 2011). Por ello en este trabajo se eligió trabajar con este sistema ya que pone en manifiesto los algoritmos propuestos.

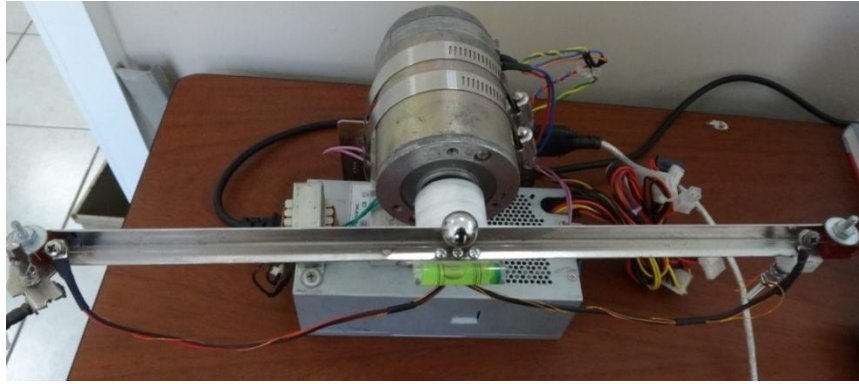


Figura 10: El prototipo *Ball and Beam*.

El *Ball and Beam* es también relacionado con problemas de control en el mundo real. Tal como la estabilización horizontal de un avión durante el aterrizaje o en turbulencias. O en tratar problemas de balanceo generados por el movimiento de un robot (Sung-Kwun et al., 2009).

Por otro lado, este sistema cuenta con dos grados de libertad, el primer grado consiste en el movimiento angular del riel, el segundo grado de libertad trata del movimiento traslacional - rotacional del balón sobre el riel. El sistema *Ball and Beam* cuenta con un actuador que es un motor eléctrico de DC (Corriente Directa) el cual tiene acoplada a su flecha el riel donde viaja libremente el balón. Los movimientos angulares generados por el motor eléctrico de DC permiten posicionar el balón en un punto deseado sobre el riel al aplicar un voltaje en la armadura de éste (figura 11).

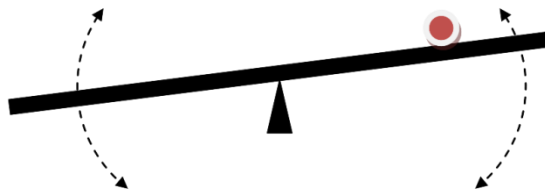


Figura 11: Movimientos angulares del riel para posicionar el balón en un punto deseado.

MODELO MATEMÁTICO.

El comportamiento dinámico del sistema *Ball and Beam*, están descritas por las ecuaciones Euler-Langrange (EL), se hace la suposición de que la masa del balón es constante para simplificar las ecuaciones (Ortega, 2002).

$$\ddot{q}_1 + g \sin(q_2) - q_1 \dot{q}_2^2 = 0 \quad (1)$$

$$(L^2 + q_1^2)\ddot{q}_2 + 2q_1\dot{q}_1\dot{q}_2 + gq_1 \cos(q_2) = u \quad (2)$$

Donde q_1 , q_2 es la posición del balón y posición angular del riel, respectivamente, L es la longitud del riel, g es la aceleración de la gravedad, y u es par mecánico generado por el motor. De la ecuación (1,2) podemos decir que el sistema es no-lineal debido a que existe productos entre los estados q_1 y q_2 , y además hay estados que son elevados a una potencia. Como ya se había mencionado el sistema es sub-actuado, esto debido a que la ecuación (1) es igual a cero esto implica que no tiene entrada directa, mientras que en la ecuación (2) es igual a u , tiene entrada directa.

La representación no lineal del sistema puede ser simplificado aplicando un nivel de tensión adecuado al motor de DC, con el fin de que no aparezca el efecto centrífugo que sufre el balón como resultado del giro del riel, es decir, si el riel tiene velocidades pequeñas el fenómeno centrífugo no tiene efecto en el sistema. Entonces si se supone que el riel y el balón se mueven a bajas velocidades y si el riel se mueve en ángulos cercanos al eje horizontal, el modelo dinámico del sistema se puede considerar como lineal ver más en (Amjad et al., 2010; Chen et al., 2012; Hernández-Guzmán et al., 2013).

El modelo lineal del sistema *Ball and Beam* se encuentra por dos sub-sistemas: el primero de ellos es el modelo matemático de un balón que viaja sobre un riel y el segundo modelo matemático consiste en el modelo de un motor de DC con carga.

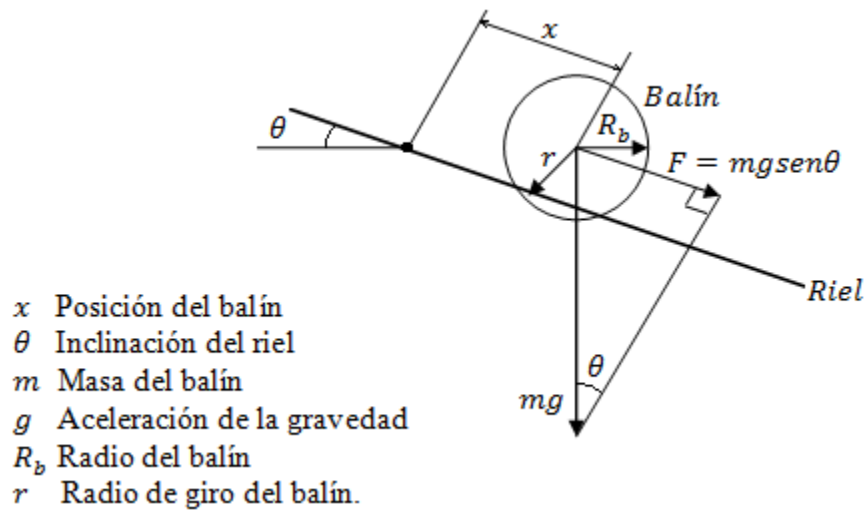


Figura 12: Diagrama del sistema *Ball and Beam*.

Modelo del movimiento del balín.

Antes de presentar el modelo matemático del balín se tiene que hacer algunas suposiciones. Una de las primeras suposiciones que se considera es que el riel tiene velocidades $\dot{\theta}$ pequeñas, esto se debe a que el balín puede tener un efecto centrífugo, por ejemplo: si el balín trata de escapar, es decir que x aumenta, al tratar de regresar el balín se cambia la inclinación θ del riel rápidamente pero debido a la fuerza centrífuga el balín continuará escapando en lugar de regresar. Otra de las suposiciones es la inclinación θ que debe tomar valores cercanos al eje horizontal.

Una vez hechas dichas suposiciones se dice que el sistema *Ball and Beam* cuenta con dos fuerzas. 1) Fuerza translacional f_t actuando a lo largo de la dirección x debido a la gravedad y 2) fuerza rotacional f_r resultado del torque producido por la aceleración rotacional del balón. Estas dos fuerzas son gobernadas por la siguiente expresión:

$$f_t = mg \sin(\theta) \quad (3)$$

Para la fuerza rotacional necesitamos conocer algunas características del balón como se muestra en la figura 13.

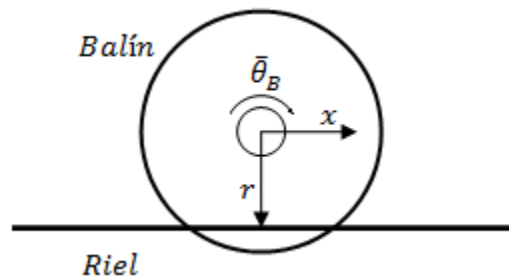


Figura 13: Variables del Balón.

Donde $\bar{\theta}_B$ posición angular del balón, x movimiento traslacional del balón y r radio de giro del balón. Si los términos se presentan como vectores resulta que la velocidad angular del balón es perpendicular al radio de giro del balón, por lo tanto.

$$\bar{\theta}_B r = x$$

$$\bar{\theta}_B = \frac{x}{r} \quad (4)$$

El torque producido por el movimiento rotacional del balón es igual al radio del balón R_b multiplicado por la fuerza rotacional (opuesto al trayecto del balón).

$$T_r = f_r R_b$$

Donde T_r es el torque del balón. Usando la Segunda Ley Newton, sabemos que el torque es igual al momento de inercia del balón J multiplicado por su aceleración angular, cual entonces también puede ser escrita como el momento de inercia del balón por la doble derivada del movimiento traslacional del balón dividido por el radio de giro del balón (4), tenemos

$$T_r = f_r R_b = J \ddot{\theta} = J \frac{\ddot{x}}{r^2} \quad (5)$$

Donde $J = \frac{2}{5} m R_b^2$ es la inercia de una esfera sólida que gira sobre su propio eje. De acuerdo a la ecuación anterior tenemos que la fuerza rotacional esta dado como:

$$f_r = \frac{2}{5} m \ddot{x} \frac{R_b^2}{r^2} \quad (6)$$

En virtud de la Segunda Ley de Newton nosotros tenemos que:

$$m \ddot{x} = \sum F = f_t - f_r = m g \sin(\theta) - \frac{2}{5} m \ddot{x} \frac{R_b^2}{r^2} \quad (7)$$

Simplificando (7) tenemos que

$$\left(1 + \frac{2 R_b^2}{5 r^2}\right) \ddot{x} = g \sin(\theta) \quad (8)$$

Como se ha mencionado anteriormente la inclinación del riel (θ) toma valores muy pequeños ($\theta \approx 0$), en tal caso nosotros podemos asumir que $\sin(\theta) \approx \theta$, por lo tanto la ecuación (8) se convierte en:

$$\left(1 + \frac{2 R_b^2}{5 r^2}\right) \ddot{x} = g \theta \quad (9)$$

Usando la transformada de Laplace con condiciones iniciales igual a cero en la ecuación (9), tenemos:

$$\frac{X(s)}{\theta(s)} = \frac{\rho}{s^2} \quad \rightarrow \quad \rho = \frac{g}{1 + \frac{2R_b^2}{5r^2}} \quad (10)$$

La ecuación (10) representa el modelo matemático del movimiento del balón.

Modelo del motor eléctrico de DC y riel ó canal

En esta sección se modela un motor de DC considerando que éste tiene una carga acoplado en su extremo, en el caso del *Ball and Beam* el riel acoplado a la flecha es una carga. Existen una variedad de motores, el motor que se utilizó es un motor de DC con escobillas y con imán permanente. El modelo matemático de este motor es de forma lineal y además trata de un sistema de una entrada – una salida.

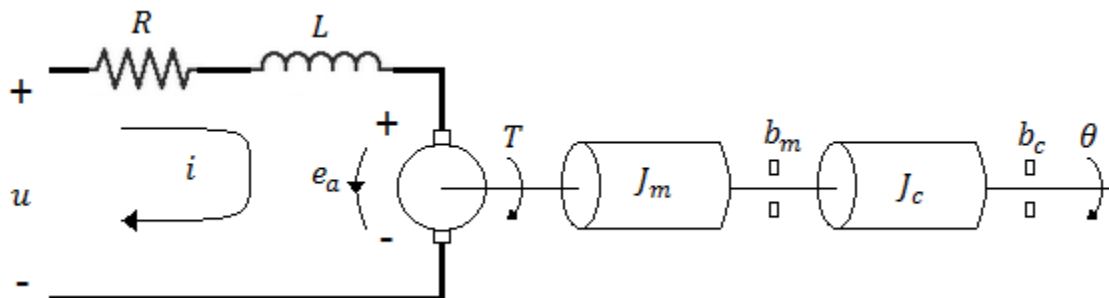


Figura 14: Motor de DC con riel acoplada.

En la figura 14 se muestra un motor de DC con escobillas y de imán permanente que mueve una carga. La nomenclatura utilizada es la siguiente:

- u es el voltaje aplicado al motor.
- i es la corriente eléctrica de la armadura.
- e_a es la fuerza contra-electromotriz.
- T es el par electromagnético generado.

- θ es la posición angular del riel.
- J_m es la inercia del rotor del motor.
- J_c es la inercia de la carga.
- b_m es la constante de fricción viscosa del motor.
- b_c es la constante de fricción viscosa de la carga.
- R es la resistencia de la armadura.
- L es la inductancia de armadura.

Nótese que un motor de DC es un sistema electromecánico compuesto de un subsistema eléctrico y un subsistema mecánico. El modelo matemático de este sistema es la unión de los dos subsistemas modelados. Por ello se presenta el modelo de cada subsistema por separado.

Modelo del subsistema eléctrico del motor de DC con carga

Del subsistema eléctrico se aplica la Ley de Kirchhoff de voltajes se tiene que:

$$\begin{aligned} \text{Volateje aplicado} &= \sum \text{caidas de voltajes en la malla} \\ u &= L \frac{di}{dt} + Ri + e_a \end{aligned} \quad (11)$$

La fuerza contra-electromotriz está dada como:

$$e_a = k_e \frac{d\theta}{dt} \quad (12)$$

Donde $k_e > 0$ es la constante de fuerza contra-electromotriz. La fuerza contra-electromotriz surge debido a que el motor de DC se comporta como un generador eléctrico de DC al incrementar la velocidad angular.

Modelo del subsistema mecánico del motor DC con carga

Del subsistema mecánico se aplica la Segunda Ley de Newton, tenemos:

$$M \frac{d^2\theta}{dt} + N \frac{d\theta}{dt} = T$$

Si $M = J_m + J_c$ son las inercias y $N = b_m + b_c$ son las constantes de fricción viscosas, entonces

$$(J_m + J_c) \frac{d^2\theta}{dt} + (b_m + b_c) \frac{d\theta}{dt} = T \quad (13)$$

Y el par electromagnético generado por el motor de DC está dado como:

$$T = k_m i \quad (14)$$

Donde k_m es la constante de par.

Modelo del motor de DC con carga

Usando (11), (12), (13) y (14) se obtiene el modelo del motor con carga de la siguiente manera

$$M \frac{d^2\theta}{dt} = k_m i - N \frac{d\theta}{dt} \quad (15)$$

$$L \frac{di}{dt} = u - Ri - k_e \frac{d\theta}{dt} \quad (16)$$

Aplicando la transformada de Laplace a las ecuaciones (15) y (16) considerando que las condiciones iniciales son iguales a cero, tenemos:

$$\theta(s) = \frac{k_m}{Ms^2 + Ns} I(s) \quad (17)$$

$$I(s) = \frac{1}{Ls + R} [U(s) - k_e s \theta(s)] \quad (18)$$

Considerar el modelo completo para el sistema de un motor con carga resulta ser complejo, pero se sabe que el subsistema mecánico tiene una respuesta lenta comparada con el subsistema eléctrico. En control a esto se denomina que el subsistema mecánico cuenta con polos dominantes, es decir, tiene polos cercanos a cero o al eje imaginario. Por ello se usó la ecuación (17) para modelar el motor con carga donde la función de transferencia está dado como:

$$\frac{\theta(s)}{I(s)} = \frac{k}{s(s + a)} \quad (19)$$

Donde $k = \frac{k_m}{M}$ y $a = \frac{N}{M}$ constantes.

DISEÑO DE CONTROLADORES

El modelo matemático del *Ball and Beam* está dado por la combinación del modelo del balón (10) y del modelo del motor con carga (19), esto es

$$\frac{X(s)}{\theta(s)} = \frac{\rho}{s^2}, \quad \frac{\theta(s)}{I(s)} = \frac{k}{s(s + a)} \quad (20)$$

De acuerdo a la ecuación (20) el modelo de la planta tiene tres polos en $s = 0$ por lo que el sistema es inestable en lazo abierto. Por ello, el objetivo de diseñar un controlador es precisamente conseguir la estabilidad en lazo cerrado. En la figura 15 se muestra el diseño del controlador en diagrama de bloques para el sistema *Ball and Beam*.

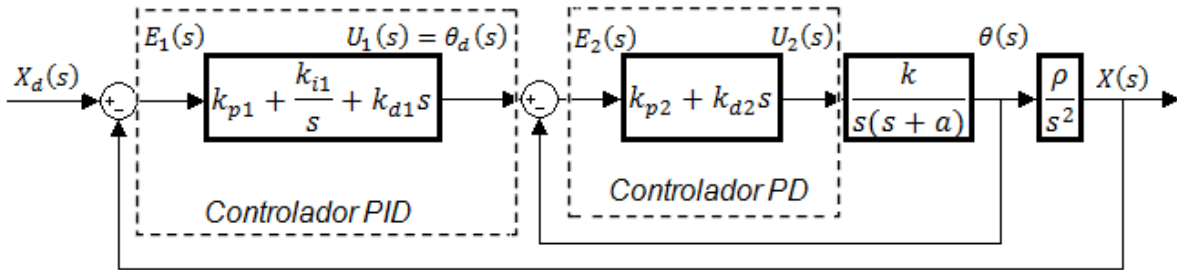


Figura 15: Diagrama de bloques para el control del *Ball and Beam*.

El sistema *Ball and Beam* usa dos controladores, para este trabajo un controlador de tipo PID (controlador maestro) la cual permite posicionar el balón en el punto deseado y un controlador de tipo PD (controlador esclavo) que permite posicionar horizontalmente el riel. La nomenclatura que utiliza el *Ball and Beam* es la siguiente

- $X_d(s)$ posición deseada del balón.
- $X(s)$ posición del balón.
- $\theta_d(s)$ posición deseada del riel.
- $\theta(s)$ posición del riel.
- $E_1(s)$ error de posición del balón.
- $E_2(s)$ error de la posición del riel.
- $U_1(s)$ señal de control del controlador PID.
- $U_2(s)$ señal de control del controlador PD.
- k_{p1}, k_{p2} ganancias proporcionales.
- K_{i1} ganancia del integrador.
- k_{d1}, k_{d2} ganancias derivativas.

El sistema *Ball and Beam* busca que $\lim_{t \rightarrow \infty} x(t) = x_d$ donde x_d es una constante, si la función de transferencia de lazo cerrado $X(s)/X_d(s)$ es estable. Por tanto, el único problema que resta es encontrar las ganancias del controlador primario así como del secundario que estabilicen el sistema.

CONTROLADORES DEL BALL AND BEAM

Como se mencionó anteriormente el sistema *Ball and Beam* cuenta con dos controladores, un controlador maestro de tipo PID y un controlador esclavo de tipo PD. Para encontrar la estabilidad de este sistema se debe seleccionar los valores de las ganancias de los controladores de forma adecuada para que el sistema presente un buen desempeño. A continuación se presenta el controlador PID y posteriormente el controlador PD.

Controlador PID

El controlador PID se usa para sistemas SISO (por sus siglas en inglés, *simple input simple output*) donde su propósito es llevar el error del sistema a cero. Este error se procesa de tres maneras diferentes, dichas maneras son: lo amplifica o atenúa, lo deriva ponderadamente y lo integra ponderadamente, dicho de otra manera, el error se ve afectado por la parte amplificadora, de alguna manera se puede predecir lo que sucederá, esto mediante la acción derivativa e incluso se puede decir que se tiene idea del pasado, esto es gracias a la acción integral del control (Aström y Hägglund, 2006).

De manera formal el controlador PID se puede expresar como

$$u_1(t) = Ke(t) + \frac{K}{T_i} \int e(t)dt + KT_d \frac{de(t)}{dt} \quad (11)$$

Esto en su representación temporal, donde $e(t)$ es la señal de error y K, T_i y T_d son constantes de ganancias del controlador, en control es común utilizar la representación en s , es decir, utilizando la transformada de Laplace, esta representación es como sigue

$$U_1(s) = K_p E(s) + K_d s E(s) + \frac{K_i}{s} E(s) = \left(K_p + sK_d + \frac{K_i}{s} \right) E(s) \quad (12)$$

Donde $K_p = K$, $K_i = K/T_i$ y $K_d = KT_d$ conocidas como ganancias proporcional, integral y derivativa, respectivamente.

Controlador PID digital

La manera como se utiliza el controlador PID ha cambiado conforme la tecnología avanza, en un principio se implementó utilizando técnicas mecánicas, especialmente neumáticas, posteriormente esta técnica se evolucionó y se aplicó la técnica electrónica de manera analógica y actualmente la implementación que más se utiliza es la electrónica digital.

Para utilizar un controlador PID en su forma digital se deben tomar en cuenta diversos aspectos que están ligadas a las señales digitales. Entre estos aspectos se encuentra el tiempo de muestreo, filtrado anti-alias, discretización y efectos de cuantificación, para profundizar en estos fenómenos se pueden utilizar las referencias como (Katsuhiko, 1995; Aström and Hägglund, 2006; Kannan, 2007).

A continuación se presenta el controlador PID en la transformada z , para ver más sobre la transformada del PID analógico a la transformada z se recomienda ver Ogata (1996).

$$U_1(z) = \left[K_p + \frac{K_I}{1 - z^{-1}} + K_D(1 - z^{-1}) \right] E(z) \quad (13)$$

Donde $K_p = K - \frac{K_I}{2}$ ganancia proporcional, $K_I = \frac{KT}{T_i}$ ganancia integral y $K_D = \frac{KT_d}{T}$ ganancia derivativa y T es el tiempo de muestreo.

Existen dos versiones de controladores PID's para su programación directa, la llamada de posición y la incremental o de velocidad. A continuación se presentan cada una de ellas.

Controlador PID en su forma posicional

$$u_1(n) = K_p \left[e(n) + \frac{T_d}{T_c} (e(n) - e(n-1)) + \frac{T_c}{T_i} \sum_{i=0}^n e(i) \right] \quad (14)$$

Forma de velocidad

$$\begin{aligned} u_1(n) &= u_1(n-1) \\ &+ K_p \left[(e(n) - e(n-1)) + \frac{T_c}{T_i} e(n) \right. \\ &\left. + \frac{T_d}{T_c} (e(n) - 2e(n-1) + e(n-2)) \right] \end{aligned} \quad (15)$$

Donde K_p es la ganancia proporcional, T_i es una ganancia integral, T_d es una ganancia derivativa y T_c es el tiempo de muestreo. En este trabajo se utilizó el controlador PID en su forma posicional debido a que es menos sensible a las señales indeseadas, es decir, al ruido que puede estar presente en la señal.

Controlador PD

El controlador PD (Proporcional y Derivativo) al igual que el controlador PID es utilizado para sistemas SISO, donde la acción derivativa proporciona una señal de control a la velocidad de cambio de la señal de error. En otras palabras, la acción derivativa es mantener el error al mínimo corrigiéndolo proporcionalmente con la misma velocidad que se produce; de esta manera evita que el error se incremente. El controlador PD se representa como:

$$u_2(t) = Ke(t) + KT_d \frac{de(t)}{dt} \quad (16)$$

Su representación en s es de la forma siguiente

$$U_1(s) = K_p E(s) + K_d s E(s) = (K_p + sK_d)E(s) \quad (17)$$

Donde $K_p = K$ y $K_d = KT_d$ conocidos como ganancia proporcional y derivativo, respectivamente.

Controlado PD en su forma digital

El controlador PD en su forma digital se presenta en la ecuación (18). Esta ecuación fue programada en Matlab 2008 al igual que la ecuación (14) y de la misma manera puede ser implementado en un dispositivo electrónico tal como un micro-controlador, FPGA, DSP o cualquier otro dispositivo programable.

$$u_2(n) = k_p \left[e(n) + \frac{T_d}{T_c} (e(n) - e(n-1)) \right] \quad (18)$$

Donde k_p ganancia proporcional, T_d ganancia derivativa y T_c tiempo de muestreo.

El controlador PID que se implementa es la ecuación (14) y el controlador PD es la ecuación (18). Ahora se presenta algunas formas de sintonización de estos controladores.

TIPOS DE SINTONIZADORES.

El controlador PD se sintonizó de forma empírica, mientras que el controlador PID se sintonizó usando un algoritmo heurístico, es importante mencionar que existen diversas formas de sintonizar un controlador PID. A continuación se menciona de manera breve una serie de métodos clásicos de sintonización de controladores PID ver más en (Astrom and Hagglund, 1995).

Ziegler-Nichols

El sintonizador de controladores PID propuestos por Ziegler-Nichols es válido sólo para sistemas estables en lazo abierto. Este método se basa en la determinación de algunas características de la dinámica de la planta a controlar y posteriormente los parámetros del controlador se expresan en fórmulas que relacionan los parámetros del controlador con los obtenidos de manera experimental.

Ziegler y Nichols sugirieron dos procedimientos para sintonizar controladores PID basándose respectivamente en la respuesta experimental a una entrada escalón, o en base al valor de k_p que produce una estabilidad marginal (oscilatorio) mediante el uso de un controlador proporcional. Los procedimientos de Ziegler-Nichols, que se presentan a continuación, son muy convenientes cuando no se conoce el modelo matemático de la planta; naturalmente, estos procedimientos también se pueden aplicar a sistemas con modelos matemáticos conocidos.

Método de Ziegler-Nichols basado en la respuesta al escalón.

El primer método propuesto por Ziegler-Nichols está basado en las características de la respuesta transitoria de una planta ante una entrada escalón unitario en lazo abierto. Esta respuesta se puede caracterizar por dos parámetros (a y L), como se muestra en la figura 16.

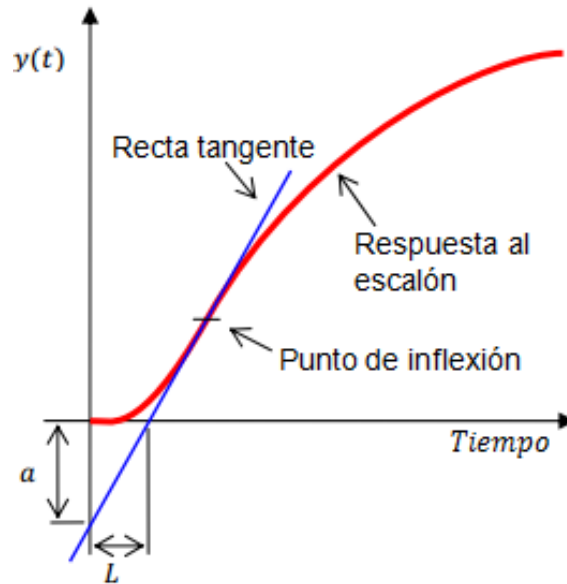


Figura 16: Método de Ziegler-Nichols: Respuesta al escalón.

Partiendo de esta aproximación Ziegler-Nichols sugirieron expresar los parámetros K , T_i y K_d del controlador PID, directamente en función de los parámetros a y L , de acuerdo con los valores que aparecen en la siguiente tabla. Esta tabla también muestra una estimación del periodo de la dinámica dominante (T_p) del sistema en lazo cerrado.

controlador	K	T_i	T_d	T_p
P	$1/a$			$4L$
PI	$0.9a$	$3L$		$5.7L$
PID	$1.2a$	$2L$	$L/2$	$3.4L$

Tabla 2.1. Método de Ziegler-Nichols: Fórmulas para respuesta al escalón.

Método de Ziegler-Nichols basado en la respuesta en frecuencia.

El método diseñado por Ziegler-Nichols conduce a fórmulas simples para obtener los parámetros del controlador PID, en términos de la ganancia crítica (K_u) y el periodo crítico (T_u), como se puede ver en la siguiente tabla.

Controlador	K	T_i	T_d	T_p
P	$0.5K_u$			T_u
PI	$0.4K_u$	$0.8T_u$		$1.4T_u$
PID	$0.6K_u$	$0.5T_u$	$0.125T_u$	$0.85T_u$

Tabla 2.2. Método de Ziegler-Nichols: Fórmulas para respuesta en frecuencia.

Cohen-Coon

Este método se aplica a sistemas estables en lazo abierto. Y se caracteriza con dos parámetros, el tiempo de atraso τ y la constante de tiempo T . El tiempo de atraso y la constante de tiempo se determinan trazando una línea tangente al punto de inflexión y se determinan las intersecciones de esta línea tangente con el eje del tiempo y con la entrada escalón, como se muestra en la figura 17.

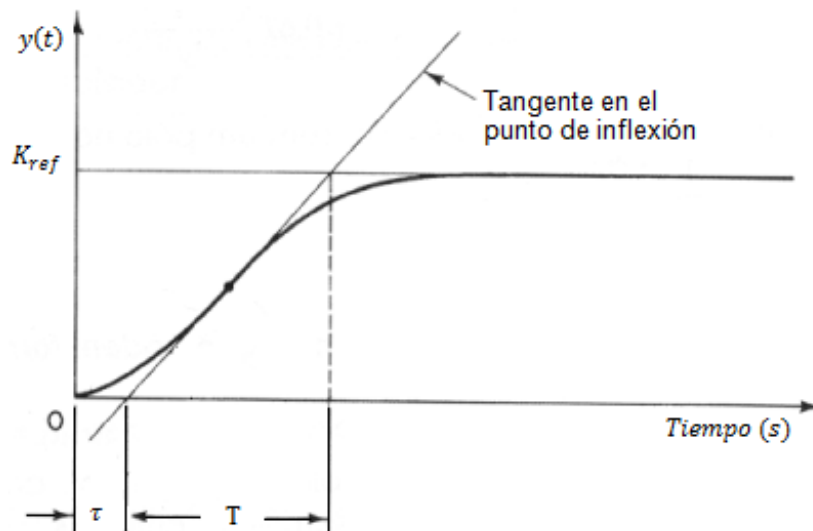


Figura 17: Método Cohen-Coon para sintonizar controladores PID.

Una vez identificado los parámetros del sistema, se obtiene los parámetros del controlador utilizando la siguiente tabla.

Controlador	K	T_i	T_d
P	T/τ		
PI	$0.9 T/\tau$	$\tau/0.3$	
PID	$1.2 T/\tau$	2τ	0.5τ

Tabla 2.3. Método Cohen-coon.

Otros

Existen varios métodos distintos al método propuesto por Ziegler-Nichols, sin embargo, este método o sus derivados obtienen resultados aceptables por los usuarios en la mayoría de las ocasiones. A continuación se dará un breve resumen de los métodos que no son derivados del propuesto por Ziegler-Nichols.

Existen métodos en los cuales se encuentra el modelo de la planta y mediante procedimientos analíticos se trata de dar valores a los parámetros del PID para que cuando se combine en cascada con la planta y se realice la retroalimentación, el sistema tenga los polos y ceros de tal forma que el resultado sea el deseado, un ejemplo de estos métodos es el de Haalman, en el cual se puede profundizar en (Aström and Hägglund, 2006).

También existen los métodos que se basan en la optimización (Tae-Hyoung et al., 2008; Hongqing et al., 2011; Hultmann-Ayala y Santo-Coelho, 2012). Dichos métodos tratan de optimizar el criterio de diseño, éstos lo logran generando una o varias funciones o desigualdades, en las cuales involucren los parámetros del PID y el criterio de mayor interés en el diseño. A pesar de que estos métodos son una herramienta poderosa, se debe tener cuidado en ciertos aspectos, tal como la función o desigualdad que se utilizará para minimizar, si éstos no se diseñan de forma adecuada, es posible que el resultado obtenido sea la optima de dicha función, sin embargo, no implicaría que el sistema tenga el desempeño deseado. Otro de los aspectos que se deben tener en cuenta es que estos algoritmos cuentan con un problema natural como el de mínimos locales.

ALGORITMOS HEURÍSTICOS

Los algoritmos heurísticos son métodos que buscan soluciones a funciones o a problemas de optimización. En años recientes, los algoritmos heurísticos han incrementado considerablemente su uso, debido a que encuentran soluciones satisfactorias a problemas complejos, a pesar de no ser capaces de probar que la solución encontrada es la óptima. En este trabajo se presenta dos métodos heurísticos, la optimización por cúmulo de partículas (PSO) y los algoritmos genéticos (AG).

Optimización mediante el cúmulo de partículas (PSO)

La optimización por cúmulos de partículas o PSO (por sus siglas en inglés, *Particle Swarm Optimization*) es una técnica metaheurística desarrollada por Kennedy y Eberhart en 1995, basándose en un enfoque conocido como la “metáfora social” esto se resume en: los individuos que conviven en una sociedad tienen una opinión que es parte de un “conjunto de creencias” compartido por todos los posibles individuos. Cada individuo puede modificar su propia opinión basándose en los siguientes factores: su conocimiento sobre su entorno, por sus experiencias (su memoria) y por las experiencias de su vecindad o sociedad (Hongqing et al., 2011; Yajuan y Qinghai, 2011; Juing-Shian et al., 2012).

A través de ciertas reglas de interacción, los individuos de la población son influenciadas sus creencias por las experiencias de los individuos con más éxito y con el tiempo las creencias de cada individuo está estrechamente relacionadas con toda la población.

La técnica PSO fue inspirada por el comportamiento social de las bandadas de aves o banco de peces (figura 18), suponiendo que dentro de una de estas bandadas se establecen relaciones entre individuos, definiendo jerarquías de acuerdo a las características y aptitudes de cada uno de ellos, existiendo así un líder que es reconocido por el grupo y que cuenta con la habilidad que le permite mantener el control sobre el grupo, donde su

principal tarea es ser el guía para buscar comida o moverse a otro lugar cuando la comida es escasa. Los individuos confían en las capacidades del líder para dirigirlos, pero en estos grupos se suele cambiar de líder si surge un individuo con mejores capacidades que el líder actual. Cada individuo se ve influenciado por dos factores: El primero es su conocimiento y habilidades propias adquiridas durante su vida y el segundo es la influencia del líder. Cuando el grupo emprende el vuelo en busca de comida, el líder guía al grupo por donde tiene la sospecha de encontrar comida, sin embargo, todos los individuos durante su vuelo tratan de avistar comida y avisar al grupo de la nueva dirección que se debe tomar para tener éxito en su misión. Durante el vuelo se debe tener noción de la posición propia y la de los demás para evitar colisiones o perder la dirección de búsqueda del líder. Este comportamiento de las bandadas de aves es la base del PSO.



Figura 18: Optimización por cúmulo de partículas (PSO) en la naturaleza.

Descripción del algoritmo PSO

Un algoritmo PSO consiste en un proceso iterativo y estocástico sobre un cúmulo de individuos (llamadas partículas o agentes inteligentes). La posición de cada partícula representa una solución potencial al problema de optimización que se está resolviendo. Una partícula p_i está compuesta de tres vectores:

- El vector de posición $x_i = [x_{i1}, x_{i2}, \dots, x_{in}]$ almacena la posición actual de la partícula que se toma como referencia para calcular la nueva dirección de búsqueda.
- El vector p_{id} ó $pBest_i = [p_{i1}, p_{i2}, \dots, p_{in}]$ almacena la posición de la mejor solución encontrada hasta esa generación, y representa la memoria de la partícula.
- El vector de velocidad $v_i = [v_{i1}, v_{i2}, \dots, v_{in}]$ almacena el gradiente (dirección del vuelo de la partícula).

El PSO determina la nueva posición de cada partícula, simplemente, añadiendo al vector de posición x_i el vector velocidad v_i para obtener un nuevo vector de posición.

El vector de velocidad de cada partícula es modificado en cada iteración utilizando su velocidad anterior, un componente que llamaremos *cognitivo* y otro componente que llamaremos *social*. El modelo matemático que representa el algoritmo PSO se presenta en la ecuación (19) que modifica la velocidad de la partícula o la dirección de búsqueda y en la ecuación (20) que modifica la posición de la partícula.

$$v_i^{k+1} = \omega \cdot v_i^k + c_1 \cdot rand_1 \cdot (pBest_i - x_i^k) + c_2 \cdot rand_2 \cdot (gBest_i - x_i^k) \quad (19)$$

$$x_i^{k+1} = x_i^k + v_i^{k+1} \quad (20)$$

En la tabla (2.4) se presentan algunas características de la ecuación (19) y (20):

VARIABLES	DESCRIPCIÓN
$c_1 \cdot rand_1 \cdot (pBest_i - x_i^k)$	Se le llama componente <i>cognitivo</i> y representa la distancia entre la posición actual y la mejor conocida por su memoria, es decir, tomará una decisión que está influenciada de acuerdo a su propia experiencia en su vida.
$c_2 \cdot rand_2 \cdot (gBest_i - x_i^k)$	Se le llama componente <i>social</i> y representa la distancia entre la posición actual y la mejor posición del grupo, es decir, tomará una decisión que estará influenciada con la experiencia del grupo.
v_i^k	Velocidad de la partícula <i>i</i> en la iteración <i>k</i> y representa la dirección actual que tiene la partícula.
ω	Factor <i>inercia</i> , regula la influencia de la velocidad actual de la partícula en el cálculo del nuevo valor de velocidad.
c_1 y c_2	La razón o rapidez de aprendizaje, también son conocidos como peso, que controlan los componentes de cognitivo y social.
$rand_1$ y $rand_2$	Representan un número real aleatorio entre [0 1].
x_i^k	Posición actual de la partícula <i>i</i> en la iteración <i>k</i> .
$pBest_i$	La mejor posición de su memoria de la partícula <i>i</i> hasta el momento.
$gBest_i$	Representa la mejor posición del cúmulo, es decir el líder.

Tabla 2.4. Parámetros del algoritmo PSO.

En la Figura 19 se muestra una representación gráfica del movimiento de una partícula *i* en el espacio de soluciones. En la gráfica: v_i^k es la velocidad de la mejor posición tomada por la partícula, v_g^k es la velocidad de la mejor partícula encontrada en el grupo. También se presenta la ruta que toma la partícula *i* que se mueve desde x_i^k hasta x_i^{k+1} el cambio de dirección que este tiene depende de la influencia de las demás direcciones que intervienen en su movimiento.

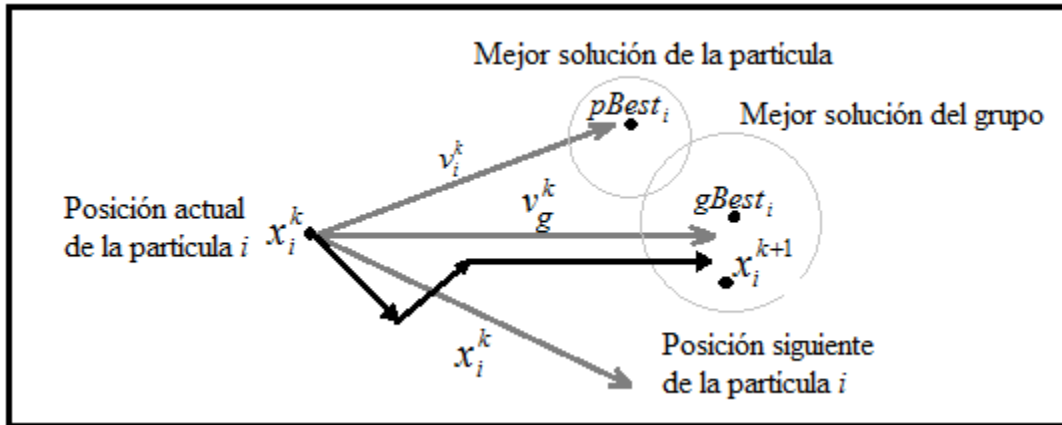


Figura 19: Movimiento de una partícula i en el espacio de soluciones.

Algoritmos Genéticos

El algoritmo genético (AG) es una técnica de búsqueda que se utiliza para optimizar sistemas o encontrar soluciones a funciones. El AG está inspirado en la evolución de Charles Darwin. Es decir, se basa en los mecanismos de selección que utiliza la naturaleza, donde los individuos más aptos de una población son los que sobreviven al adaptarse más fácilmente a los cambios que se producen en su entorno. El principio básico del algoritmo genético fue establecido por John Holland en 1975 (Glover y Kochenberger, 2003; Sivanandam y Deepa, 2008; Sung-Kwun et al., 2009; Luna-Ortiz, 2011).

De acuerdo a la literatura, existen muchas variantes de los algoritmos genéticos pero la esencia de esta técnica es la misma: dada una población de individuos (posibles soluciones del problema), la influencia del ambiente o el entorno ocasiona selección natural (la supervivencia del más apto), misma que da lugar a un incremento en la aptitud de la población. En una forma explícita, se diseña una función de calidad a ser maximizada o minimizada, posteriormente se crea de forma aleatoria un conjunto de soluciones candidatas, es decir, elementos del dominio de la función, y aplicar la función de calidad como una medida de aptitud abstracta. Basado sobre la aptitud de cada individuo, algunos

de los mejores candidatos son seleccionados para pasar a la siguiente generación y aplicarles recombinación y/o mutación. La recombinación es un operador aplicado a dos o más candidatos seleccionados (llamados padres). Realizando la recombinación y la mutación se conduce a un conjunto de nuevos candidatos (hijos) que compiten junto con los padres por un lugar en la siguiente generación. Este proceso puede ser iterativo hasta que un candidato con suficiente calidad (una solución) sea encontrado o que algún criterio de finalización establecido sea previamente alcanzado (figura 20).

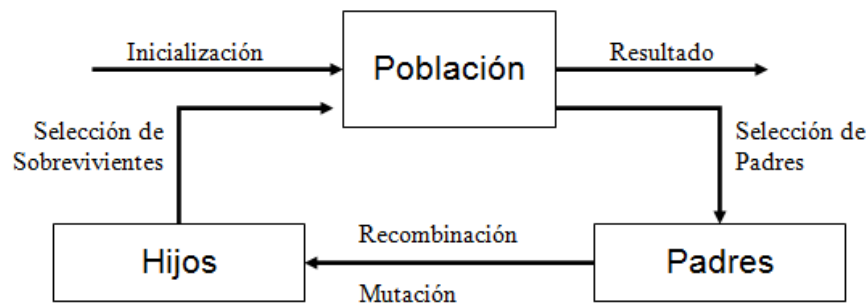


Figura 20: Algoritmo genético.

El algoritmo genético es una herramienta que pertenece a la rama de la inteligencia artificial, la cual ha probado ser útil para encontrar soluciones a problemas reales. Como se ha mencionado el AG es una herramienta de optimización. Un enfoque a problemas de control, la optimización de una función por lo general resulta ser el error entre la referencia y la salida del sistema. En muchos sistemas se espera que la respuesta del sistema tenga tiempos cortos, sobrepasos pequeños y sin errores a lo largo del tiempo.

De manera formal el algoritmo genético tiene el siguiente diagrama.

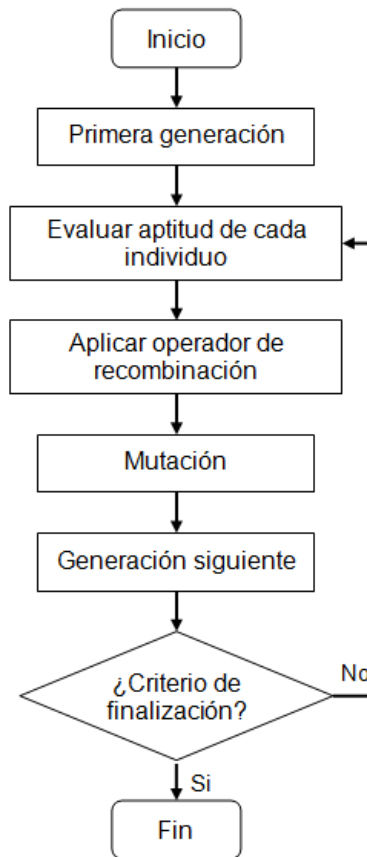


Figura 21: Diagrama del algoritmo genético.

Primera generación

Como en la mayoría de los programas el primer paso es inicializar las variables a utilizar. Sin embargo, en el algoritmo genético se genera la primera generación tomando en cuenta el espacio de búsqueda de la solución, el espacio de búsqueda se genera de forma aleatoria y cada elemento de esta generación se le conoce como individuo. Donde cada individuo es una posible solución al problema. Otro factor que se considera al generar la primera generación es el tamaño de la población es decir el número de individuos que competirán para sobrevivir y por último que tan aleatorio se pretende generar el espacio de búsqueda.

Evaluar aptitud de cada individuo

Para encontrar la solución al problema de optimización, se debe evaluar cada individuo de la generación en una función de calidad o aptitud. Al someter cada individuo en la función de aptitud lo que se hace es cuantificar sus cualidades, de tal manera y bajo algún criterio se seleccionan a los individuos que participarán en la recombinación y generar nuevos candidatos que competirán por sobrevivir y ocupar un espacio en la siguiente generación o incluso puede llegar el caso de que uno de estos elementos cumpla con todas las cualidades buscadas y sea la solución.

La función de aptitud es un punto crítico del algoritmo genético. Esta función debe ser capaz de poner en manifiesto la aptitud de cada individuo para el problema que se desea optimizar.

Selección

Después de conocer la aptitud de cada uno de los individuos se procede a elegir los individuos que serán padres. Para este proceso existe diferentes técnicas de selección, una de ellas es seleccionar la mitad de la generación que haya mostrado ser la más apta para sobrevivir, a este proceso se le conoce como elitismo. Sin embargo, esta técnica en algunos problemas presenta convergencia prematura, en tal caso existen otras alternativas de hacer la selección de padres como la selección aleatoria o la conocida como ruleta.

Operador de recombinación

Una vez hecha la selección de los padres se prosigue hacer el cruce entre padres, el operador que más se utiliza es seleccionar dos padres y a partir de ellos crear dos hijos, la generación de hijos se conoce como el proceso de recombinación. Existen diversas maneras de generar los hijos dependiendo de la codificación que se usa, en este trabajo se usan números reales y el operador de recombinación se hace de manera aritmética.

Mutación

La mutación es opcional, en muchos de los casos es preferible contar con este operador, ya que puede llegar el caso en que el algoritmo genético llegue a una solución local, donde este operador ayuda a encontrar una solución global. Cabe mencionar que la mutación surge de manera aleatoria y uno de los individuos de la población mutará, es decir, cambiará completamente por otro agente del espacio de búsqueda que es generado de forma aleatoria.

Generación siguiente

Para generar la siguiente generación se toma los padres que fueron seleccionados con respecto a su aptitud y los hijos que previamente fueron creados a partir de la recombinación de los padres y además de la mutación si este fue generado. La nueva generación cuenta con el mismo número de individuos como la generación pasada y están listos para competir por un puesto en la siguiente generación.

Criterio de finalización

Este paso es esencial para que el algoritmo genético detenga la búsqueda, la búsqueda puede ser detenida si uno de los individuos presenta las aptitudes deseadas, es decir, el algoritmo genético llegó a la solución. O puede detenerse si ya cumplió con un número de generaciones, es decir, un número de iteraciones que previamente se ha elegido.

DESARROLLO

Una vez presentado el sistema *Ball and Beam*, los controladores, las técnicas de sintonización de los controladores y la teoría del algoritmo PSO y AG se procedió a encontrar los parámetros del sistema *Ball and Beam* ya que para poder aplicar los algoritmos heurísticos se debe tener la certeza de que el sistema que se desea optimizar se conocen los parámetros correctos, en caso contrario, estos algoritmos encontrarán soluciones aceptables, sin embargo, la respuesta del sistema no será de la misma manera.

PARÁMETROS DEL SISTEMA BALL AND BEAM

Para sintonizar el controlador PID del sistema *Ball and Beam* basado en algún algoritmo heurístico el primer paso es diseñar la función que se desea minimizar, es decir, diseñar la función en el cual se desea encontrar el mínimo. En el caso del *Ball and Beam* se ha presentado el modelo matemático que representa la dinámica del sistema, además de los controladores que hacen posible estabilizar este sistema. Para el algoritmo heurístico el modelo matemático junto con los controladores es la función que se desea minimizar. Sin embargo, es importante conocer los parámetros del sistema *Ball and beam* ya que sin ellas el algoritmo heurístico encontraría las ganancias óptimas del controlador PID pero esto no implicaría que el sistema tenga un buen desempeño. A continuación se presenta el proceso que se utilizó para encontrar los parámetros del *Ball and Beam*.

Identificación de parámetros del sistema *Ball and Beam*

El sistema *Ball and Beam* se puede representar por dos modelos matemáticos en cascada, dichos modelos son: modelo matemático de un motor eléctrico de DC con carga y un modelo matemático de un balón que gira a través de un riel. Siendo así, la identificación de parámetros se realizó de dos maneras distintas: el primero fue identificar los parámetros del motor de DC con carga y posteriormente los parámetros del modelo del balón (Hernández-Guzmán et al., 2013).

Identificación de parámetros de un motor de DC con carga.

Se realizó un experimento que permitió conocer los valores de las constantes k y a del modelo dinámico del motor de DC con carga, ecuación (19). A continuación se explica la manera en el que se diseñó dicho experimento. Suponga que no hay ninguna perturbación externa aplicada al sistema, dicho modelo se representa como sigue:

$$\frac{\theta(s)}{I(s)} = \frac{k}{s(s+a)} \quad (21)$$

Para encontrar los parámetros se diseña un controlador proporcional de posición, como se muestra en la siguiente ecuación.

$$I^*(s) = k_p(\theta_d(s) - \theta(s)) \quad (22)$$

El modelo (21) junto con el controlador (22) en lazo cerrado se muestra en la siguiente figura.

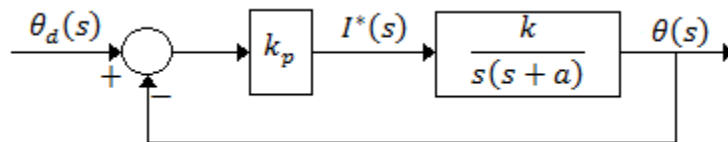


Figura 22: Control proporcional de posición para el riel.

Donde la función de transferencia correspondiente es:

$$\frac{\theta(s)}{\theta_a(s)} = \frac{k_p k}{s^2 + as + k_p k} = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} \quad (23)$$

Tal que

$$k = \frac{\omega_n^2}{k_p}, \quad a = 2\zeta\omega_n \quad y \quad \omega_n = \sqrt{k_p k} \quad (24)$$

Donde ω_n se conoce como la frecuencia natural y ζ es un factor de amortiguamiento.

En la figura 23 se presenta el resultado experimental obtenido cuando $k_p = 27$. Este experimento se realizó sin colocar el balón sobre el riel. Donde $t_r = 0.1143$ y $M_p(\%) = 76.5$ medidos directamente en la figura 23.

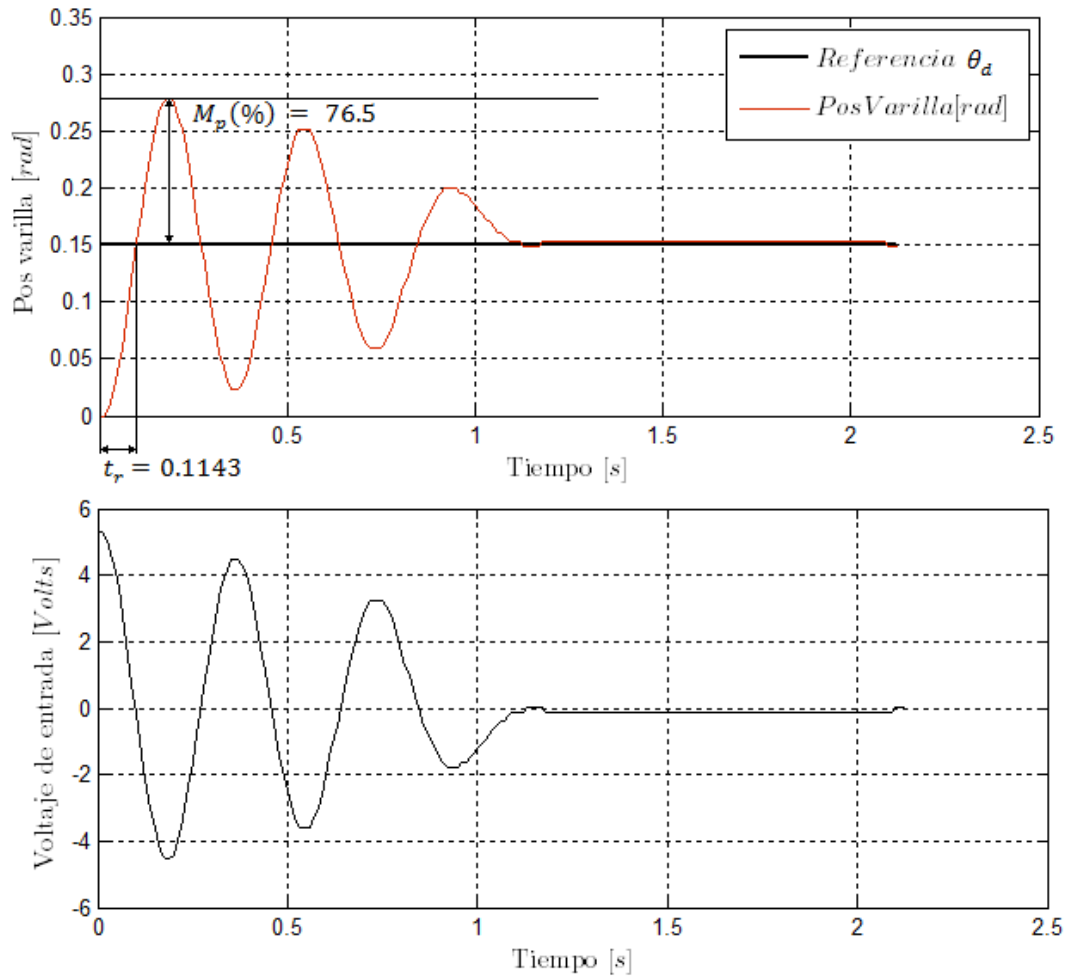


Figura 23: Resultado experimental usando $k_p = 27$.

Los resultados obtenidos experimentalmente se utilizan en las siguientes expresiones:

$$\zeta = \frac{\ln^2\left(\frac{M_p(\%)}{100}\right)}{\sqrt{\ln^2\left(\frac{M_p(\%)}{100}\right) + \pi^2}} \quad (25)$$

$$\omega_d = \frac{1}{t_r} \left[\pi - \tan^{-1}\left(\frac{\sqrt{1-\zeta^2}}{\zeta}\right) \right] \quad (36)$$

$$\omega_n = \frac{\omega_d}{\sqrt{1-\zeta^2}} \quad (27)$$

De acuerdo a las ecuaciones 25, 26 y 27 tenemos que $\zeta = 0.085$ y $\omega_n = 14.5395[\text{rad/s}]$.

Finalmente utilizando la expresión (24) con $k_p = 27$, se encuentra:

$$k = 7.8296, \quad a = 2.4706 \quad (28)$$

Identificación de parámetros del balón

En esta sección se presenta el proceso para encontrar el valor de ρ del modelo matemático del balón (Ecuación 10) que se expresa de la siguiente manera.

$$\frac{X(s)}{\theta(s)} = \frac{\rho}{s^2} \quad (29)$$

Se aplica la transformada inversa de Laplace considerando condiciones iniciales iguales a cero al modelo dinámico del balón ecuación (29) se tiene:

$$\ddot{x} = \rho\theta(t) \quad (30)$$

Ahora suponga que $\theta(t) = \theta_0$ donde θ_0 es una constante, entonces integrando dos veces la ecuación (40), tenemos:

$$x(t) = \frac{1}{2}\rho\theta_0 t^2 \quad (31)$$

A continuación se presenta el experimento que se realizó para encontrar ρ .

- Primero se fijó el riel en una posición $\theta_0 = 0.1 [\text{rad}]$ (el equivalente a 5.729 grados) se escogió este ángulo debido a que $\sin 0.1 = 0.099$. Es decir, cumple con la condición $\sin(\theta_0) \approx \theta$ propuesta para linealizar el sistema *Ball and Beam*.

- Posteriormente se posicionó el balón en $x = 0$ y se dejó rodar libremente sobre el riel hasta un extremo de éste sin que el riel cambie su inclinación antes mencionado.
- Se gráfica la trayectoria del balón (figura 24).
- Entonces se propone diferentes valores de ρ hasta que la gráfica de la función $\frac{1}{2}\rho\theta_0 t^2$ coincidiera con los datos experimentales en todo el tiempo de la experimentación.

Realizado todos los pasos anteriores se encuentra que:

$$\rho = 4.2 \quad (32)$$

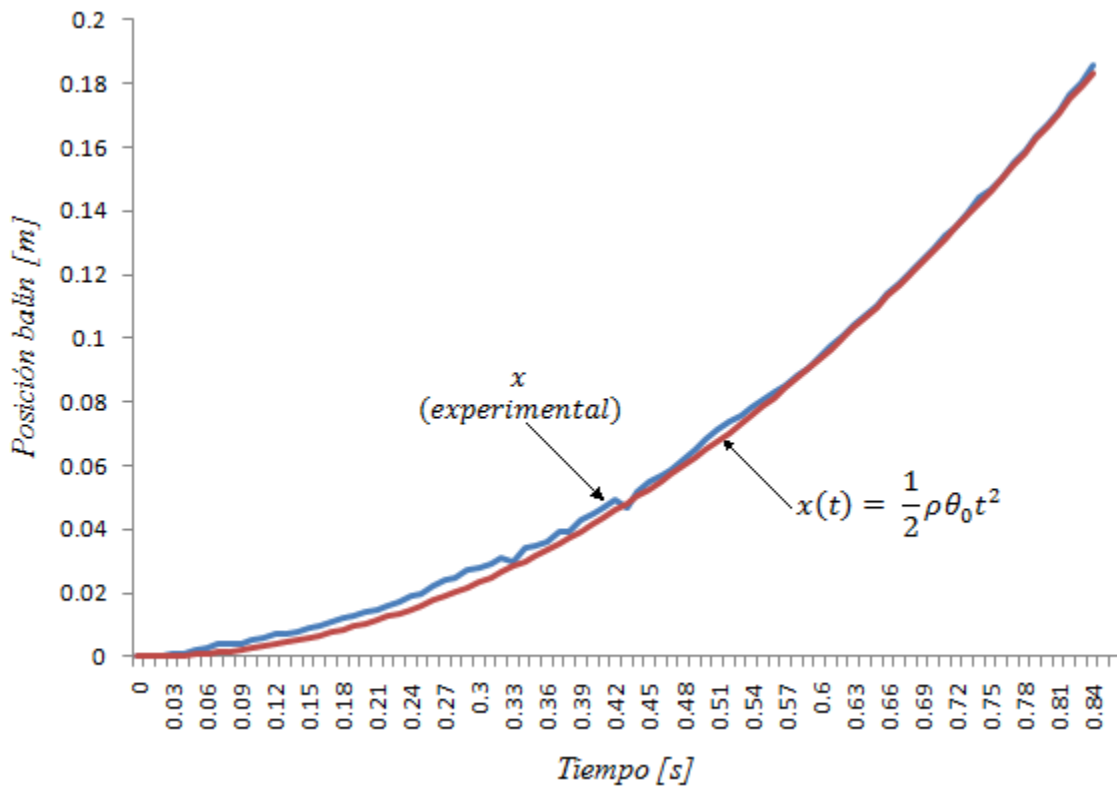


Figura 24: Identificación experimental del parámetro ρ .

El sistema *Ball and Beam* con los parámetros correspondientes se representa de la siguiente forma.

$$\frac{X(s)}{\theta(s)} = \frac{4.2}{s^2}, \quad \frac{\theta(s)}{I(s)} = \frac{7.8296}{s(s + 2.4706)} \quad (43)$$

EL SISTEMA BALL AND BEAM EN TIEMPO DISCRETO

La tendencia actual de controlar los sistemas dinámicos en forma digital en lugar de analógica, se debe principalmente a la disponibilidad de dispositivos electrónicos digitales de bajo costo. Donde en esta sección se presenta el sistema *Ball and Beam* en tiempo discreto. En la siguiente figura se muestra el sistema *Ball and Beam* en la representación s , es decir, en la transformada de Laplace.

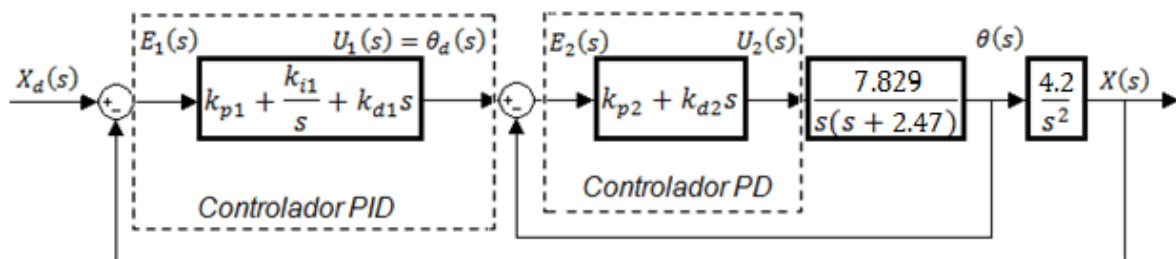


Figura 25: Sistema *Ball and Beam* en su representación s .

En la figura 26. Se presenta el sistema en su representación z (en la parte de anexo de este documento se muestra como se hace dicha transformación) tomando en cuenta que el tiempo de muestreo es de 10 ms y con los controladores antes presentados, tenemos:

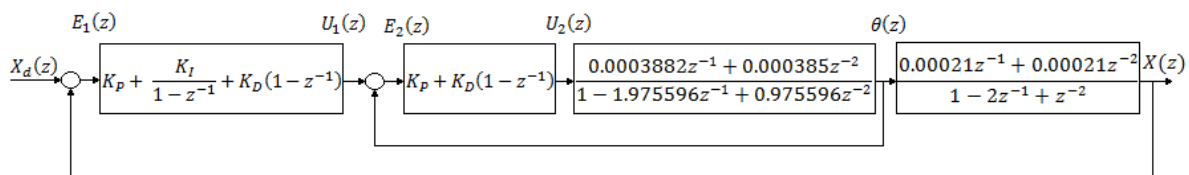


Figura 26: Sistema *Ball and Beam* en su representación z .

INSTRUMENTACIÓN DEL SISTEMA BALL AND BEAM.

Se utilizó una computadora para realizar el control del sistema *Ball and Beam* y el programa está bajo la plataforma Matlab 2008. También se utilizó un protocolo de comunicación de tipo serial RS-232 que hace comunicación con el prototipo a través de un micro-controlador PIC16F877A donde este último hace la lectura de los sensores y manda la señal de control a un regulador de voltaje que se encarga de proporcionar el voltaje requerido al motor de DC. La descripción antes mencionada se muestra en la figura 27.

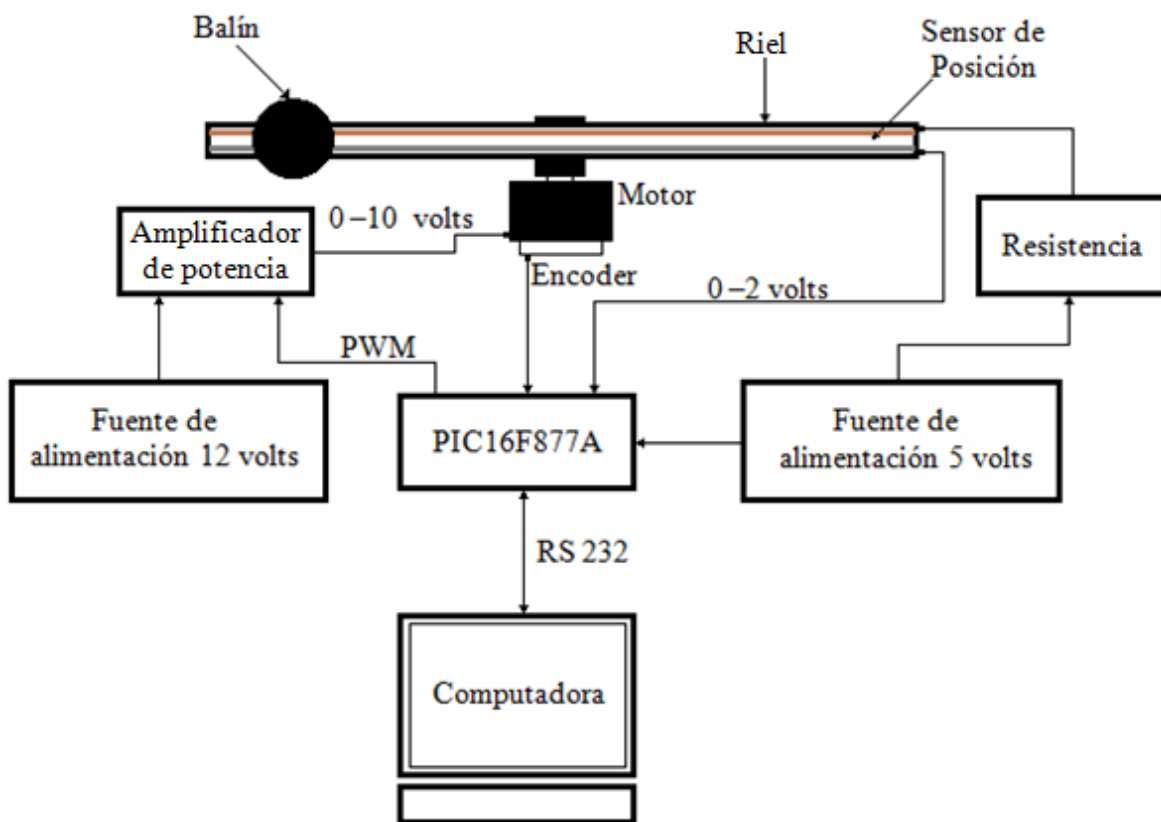


Figura 27: Diagrama esquemático para el control del sistema *Ball and Beam*.

A continuación se presenta una descripción de los sensores que se utilizaron.

Sensores de posición del riel

Se utilizó un sensor óptico para conocer la posición angular del riel, este sensor se encuentra acoplado al motor de DC como se muestra en la figura 28. Este sensor también se conoce como encoder.



Figura 28: Sensor óptico para el riel (Encoder).

El encoder tiene una resolución de 400 cuentas por revolución y este es leído por el micro-controlador. Para procesar esta señal en Matlab se realizó el siguiente escalamiento:

$$\theta = \frac{2 * \pi}{cpr} pos \quad (34)$$

Donde θ es la posición angular del riel en radianes, $cpr = 400$ cuentas, resolución del encoder, pos posición actual del encoder leído por el micro-controlador.

Sensor de posición para el balín

Para conocer la posición del balín se utilizó el método que se muestra en la figura 29. En la práctica se utilizó una cuerda de guitarra metálica que funciona como una resistencia eléctrica variable donde el balín debe ser un conductor eléctrico que al viajar a lo largo de esta cuerda pueda variar la señal de salida. Cabe mencionar que se le agregó una resistencia de protección de 3.3Ω 10 *watts* en serie para reducir la corriente que circula a través de ella debido a que la cuerda tiene una resistencia de valor pequeño. La señal de salida está en el rango 0 – 2 volts pero antes de ser leído por la entrada analógica del micro-controlador se le agrega un filtro RC para eliminar el ruido eléctrico que pueda existir en la señal.

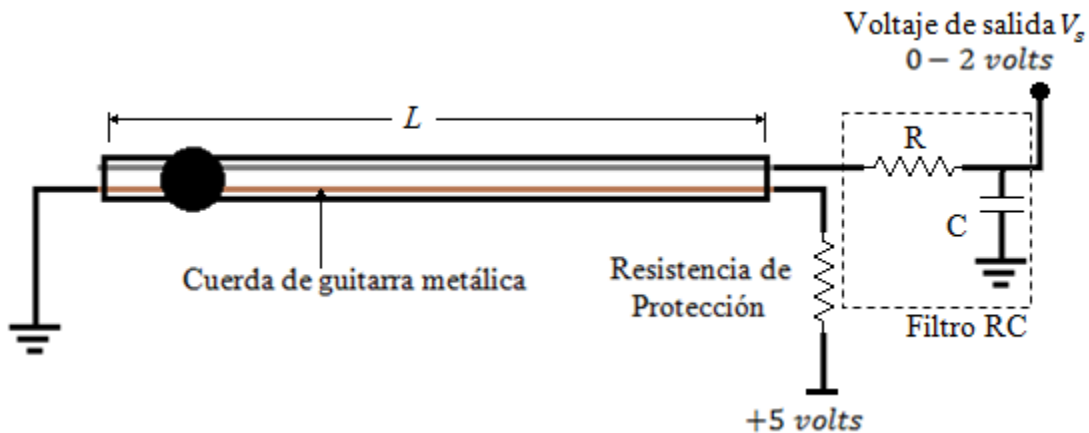


Figura 29: Diagrama del sensor de posición del balín.

Donde $L=0.314m$ es la longitud del riel, R resistencia eléctrica, C capacitor eléctrico. De igual forma que la señal del riel se hace un escalamiento, este escalamiento es de la siguiente manera

$$x = \frac{L}{cADC} V_s \quad (35)$$

Del escalamiento anterior $cADC = 1024$ es la resolución de la entrada analógica del micro-controlador, x posición del balín en metros y V_s representa la posición del balín en voltaje.

IMPLEMENTACIÓN DEL ALGORITMO HEURÍSTICO.

Algoritmo PSO

Para comenzar con el algoritmo PSO antes se define el número de partículas o aves que harán la búsqueda, para esta variable no existe en la literatura un número recomendable pero como es de pensar si se ocupa una cantidad enorme de aves entonces eso implica que el espacio de búsqueda será de igual forma, de tal manera, que se tendrá una mayor probabilidad de encontrar la solución en pocas iteraciones. Tener un número cuantioso de aves puede resultar contraproducente debido a que al cuantificar la aptitud de cada ave hará que el tiempo de búsqueda incremente considerablemente. Por otro parte, el número de aves también puede estar limitado por la memoria de los dispositivos electrónicos. En este trabajo se eligió $n = 100$ aves que buscaran la solución. Con un número máximo de $ite = 50$ iteraciones. Estos números se establecieron de acuerdo a pruebas hechas en simulaciones. Cabe mencionar que la dimensión del problema es de orden tres $dim = 3$ esto es debido a que se cuenta con tres variables $[k_p, k_i, k_d]$.

En la etapa de inicializar las variables también se definen los parámetros del PSO, en este caso $c_1 = 0.2$ y $c_2 = 0.2$ que son la razón o rapidez de aprendizaje del componente cognitivo y social, respectivamente. También se inicializó los parámetros R_1 y R_2 de manera aleatoria en un intervalo de $[0, 1]$. Posteriormente se inicializó la posición y velocidad de cada partícula de manera aleatoria.

Una vez definido los parámetros del PSO y generados las partículas se cuantifica la aptitud de cada partícula. La cuantificación de la aptitud de cada partícula se hace a través de la integral del error al cuadrado o como se conoce comúnmente ISE (por sus siglas en inglés, *Integral Squared Error*) la intención de este método es minimizar el error de salida, en la función del tiempo transcurrido (Acedo-Sánchez, 2006).

$$ISE = \int_0^T e^2(t)dt \quad (36)$$

El ISE, es relativamente insensible a pequeños errores, pero los grandes errores contribuyen fuertemente al valor de la integral. Consecuentemente, utilizar el ISE como criterio de optimización dará como resultado una respuesta con pequeños sobrepasos en el sistema pero con un largo tiempo de estabilización puesto que los pequeños errores a lo largo del tiempo contribuyen muy poco a la integral.

Existen otros criterios de optimización tales como utilizar la integral del valor absoluto del error, IAE (por sus siglas en inglés, *Integral Absolute Error*).

$$IAE = \int_0^T |e(t)|dt \quad (37)$$

Este criterio es más sensible a pequeños errores pero es menos sensible que el ISE a grandes errores.

También se encuentra la integral del tiempo multiplicado por el valor absoluto del error, ITAE (por sus siglas en inglés, *Integral Time Absolute Error*)

$$ITAE = \int_0^T t|e(t)|dt \quad (38)$$

El ITAE, es insensible a los errores iniciales, y a veces inevitables, pero penalizan fuertemente los errores que permanecen a lo largo del tiempo. La respuesta óptima definida por ITAE, consecuentemente, mostrará tiempos cortos en la estabilidad y un mayor sobrepaso en la respuesta del sistema que los otros criterios.

Una vez finalizado el proceso de cuantificar la aptitud de cada individuo o partícula se procede a buscar el individuo con la mejor aptitud, en otras palabras, como se desea encontrar el mínimo del sistema entonces se busca el individuo que presenta un número menor de ISE. Este individuo será el líder que guiará al resto de la población. En la primera generación cada individuo no cuenta con un historial de tal manera que su experiencia es su posición actual. De esta forma se determina la siguiente posición y velocidad de acuerdo a la ecuación 19 y 20.

La ecuación 19 y 20 generan la nueva generación. Por ende se repite el procedimiento hasta cumplir con el número máximo de iteraciones, Cabe mencionar, si en una de las generaciones existe un individuo que exhibe una mejor aptitud que el líder actual, el individuo con la mejor aptitud será el nuevo líder que guiará al resto de la población. De igual manera, la experiencia de cada individuo va evolucionando. Por último, al terminar el proceso el líder será la solución del problema.

Todo lo descrito anteriormente se presenta en la siguiente figura.

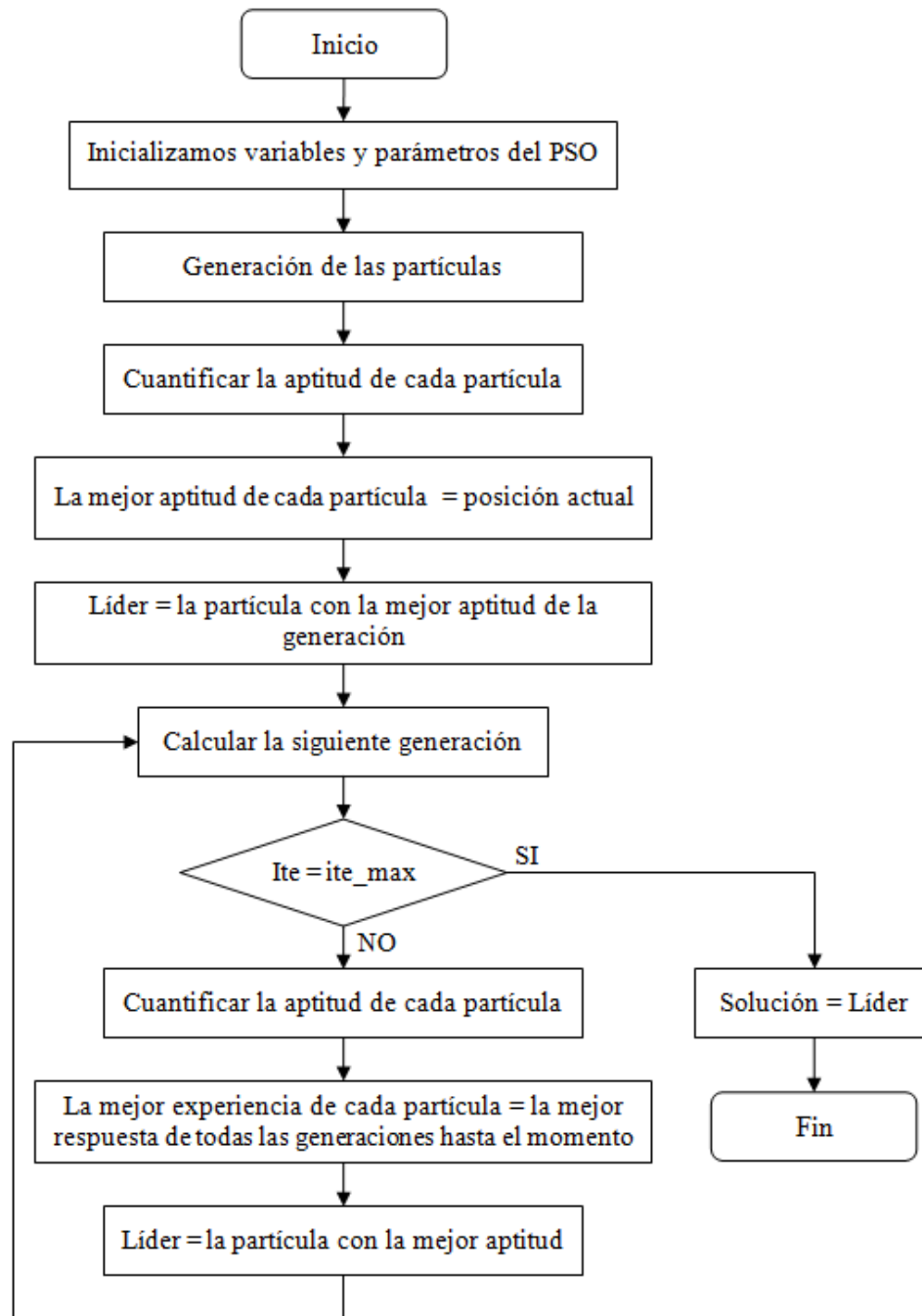


Figura 30: Diagrama de flujo para el algoritmo PSO.

Algoritmo Genético

El algoritmo genético o como cualquier otro algoritmo heurístico se debe definir el tamaño de la población, en el caso del algoritmo genético al igual que el PSO no existe una recomendación en la literatura para definir el tamaño de éste. Sin embargo, esta variable depende del problema a optimizar, tal como: la memoria del dispositivo electrónico, el tiempo de búsqueda, entre otros. En este trabajo la población del algoritmo genético es de $n = 200$ individuos y con una dimensión de tercer orden al problema de optimización. También se define el número de iteraciones máximo para finalizar la búsqueda, en este caso $ite = 25$ iteraciones donde estos parámetros se establecieron a prueba y error hechos en simulación.

Crear la población de manera aleatoria es el siguiente paso a realizar, donde cada individuo de la población es candidata a la solución. Por otra parte se evalúa la aptitud de cada uno de los individuos y posteriormente se ordena desde el individuo más apto hasta el más débil, de esta manera facilita el proceso de selección de los padres que se recombinaran y pasaran a la siguiente generación. La aptitud se mide de acuerdo al ISE que cuenta cada individuo. Para este problema de optimización los padres P son los primeros cien individuos con mejor aptitud y la recombinación de estos se hizo de la siguiente manera.

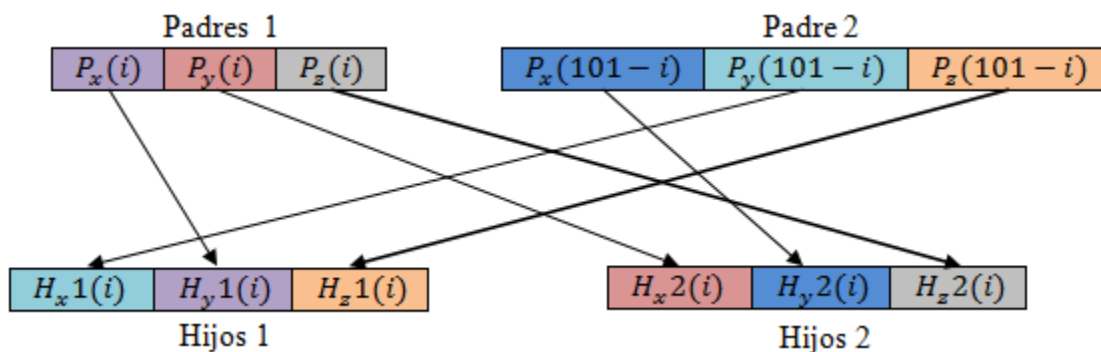


Figura 8: Recombinación de los padres para crear hijos.

Donde $i = 1,2,3 \dots 50$, de esta manera se combinan individuos con mejores aptitudes con otros de menor aptitud para crear hijos H como se muestra en la figura 31. Posteriormente, se genera la siguiente generación, esta generación se compone de los padres seleccionados y los hijos generados por ellos. Donde competirán de nuevo por sobrevivir en la siguiente generación, esto se repite hasta llegar al número máximo de iteraciones, una vez finalizado la solución será aquel que cuente con la mejor aptitud.

El proceso antes mencionado se describe en la siguiente figura:

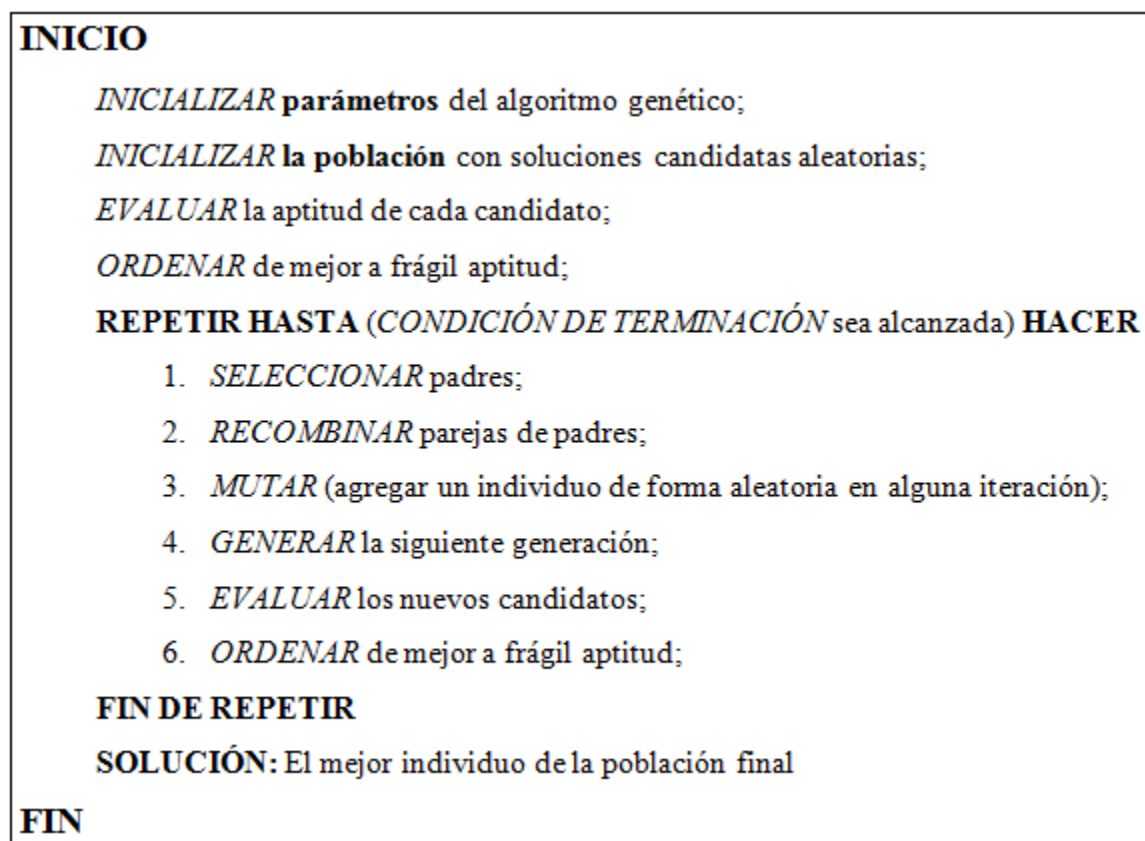


Figura 32: Esquema del algoritmo genético en pseudocódigo.

RESULTADOS

En esta sección se presenta los resultados que se obtuvo tanto en simulación como en la parte experimental del sistema *Ball and Beam*.

RESULTADOS SIMULADOS

En esta sección se presentan los resultados obtenidos en simulación. Las simulaciones fueron realizadas en Matlab 2008. En la primera parte se presenta los resultados del algoritmo genético y posteriormente los resultados del algoritmo PSO.

Simulación con el Algoritmo Genético

Como se mencionó anteriormente el objetivo del *Ball and Beam* es posicionar el balón en algún punto deseado sobre el riel, en este caso, el punto deseado o referencia se consideró en $ref = 0.05 m$. En la parte a) de la figura 33, se muestra la respuesta de la posición del balón, mientras que en la parte b) se muestra la respuesta de la posición del riel, en este último cabe mencionar que la posición deseada θ_d es generado a través de la señal del controlador PID.

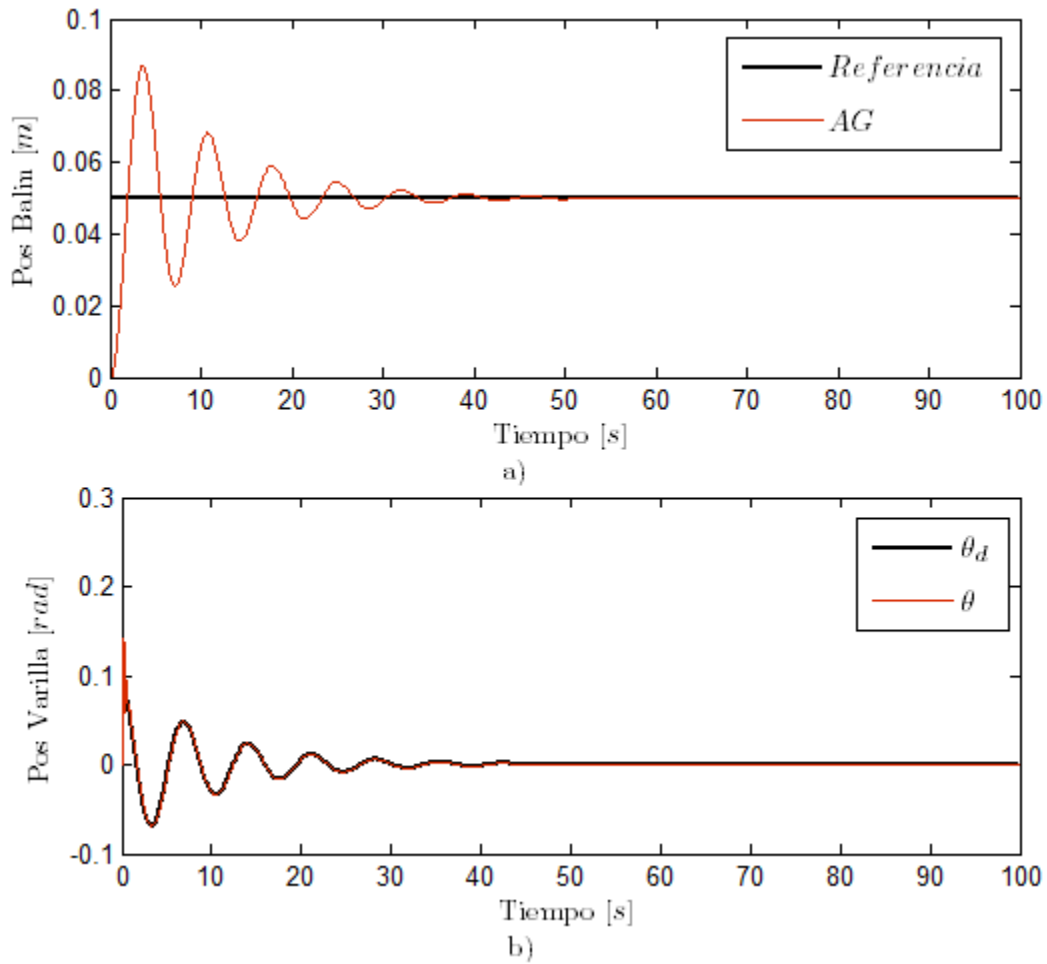


Figura 33: Realizado con el algoritmo genético a) Respuesta del balón con la sintonización del controlador PID b) respuesta del riel.

Los algoritmos heurísticos tal como el algoritmo genético tienen condiciones para terminar el proceso de búsqueda, estas condiciones dependen del proceso a optimizar o por el número de iteraciones. En este caso se eligió un número límite de iteraciones tomando en cuenta la convergencia del algoritmo, este número se estableció de acuerdo a las pruebas que se realizaron. En la figura 34 se presenta la evolución de las ganancias del controlador PID. El número de iteraciones máximo es de veinticinco pero resulta que el algoritmo genético converge a partir de la octava iteración.

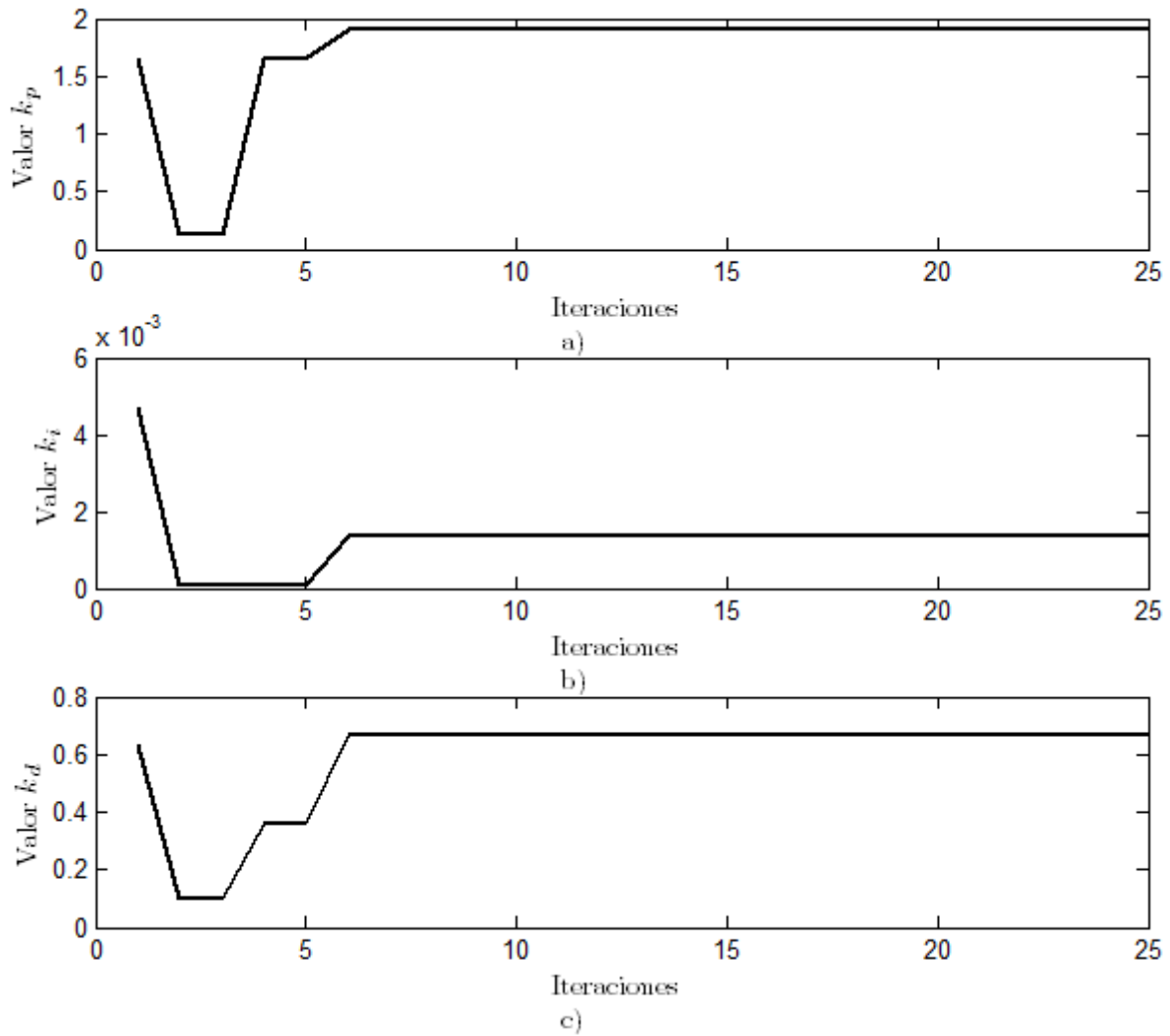


Figura 34: Evolución de las ganancias de controlador PID con el algoritmo genético.

Para poder aplicar este algoritmo de manera experimental es primordial conocer la señal de control, es decir, el par mecánico que requiere ser introducido al sistema. Por ello en la siguiente figura se muestra la evolución de la señal de control que se traduce en voltaje que es aplicado al motor eléctrico de DC y como se muestra en la figura el rango de voltaje requerido para estabilizar el sistema *Ball and Beam* es bajo.

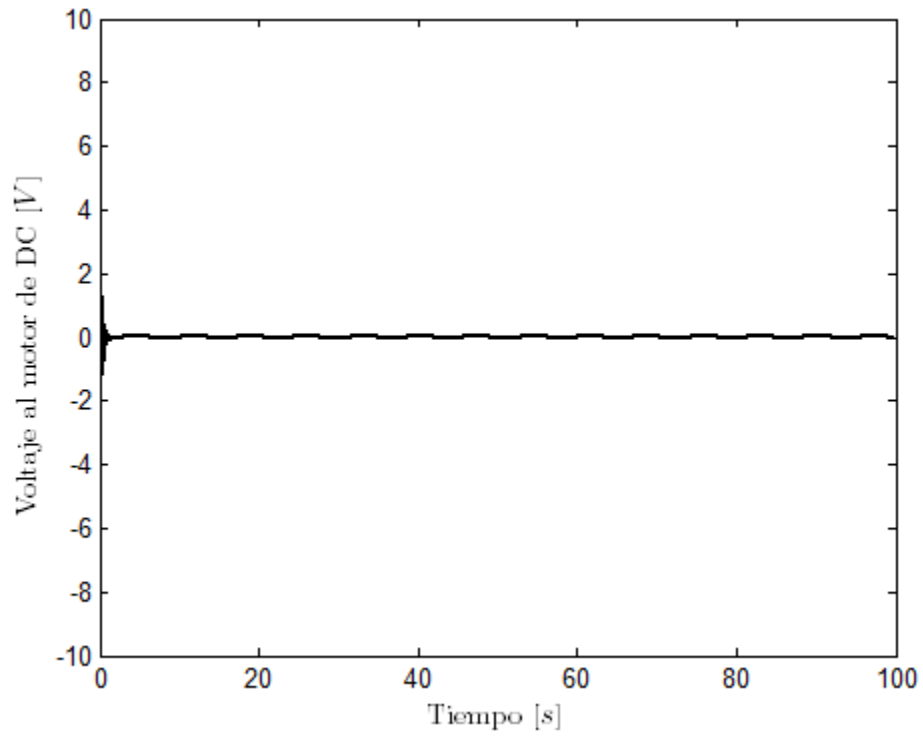


Figura 35: Señal de control con el algoritmo genético.

Simulación del algoritmo PSO

En la figura 36 se muestra el desempeño del algoritmo PSO y a comparación del algoritmo genético éste tiene un sobrepaso pequeño y un intervalo de tiempo de convergencia menor.

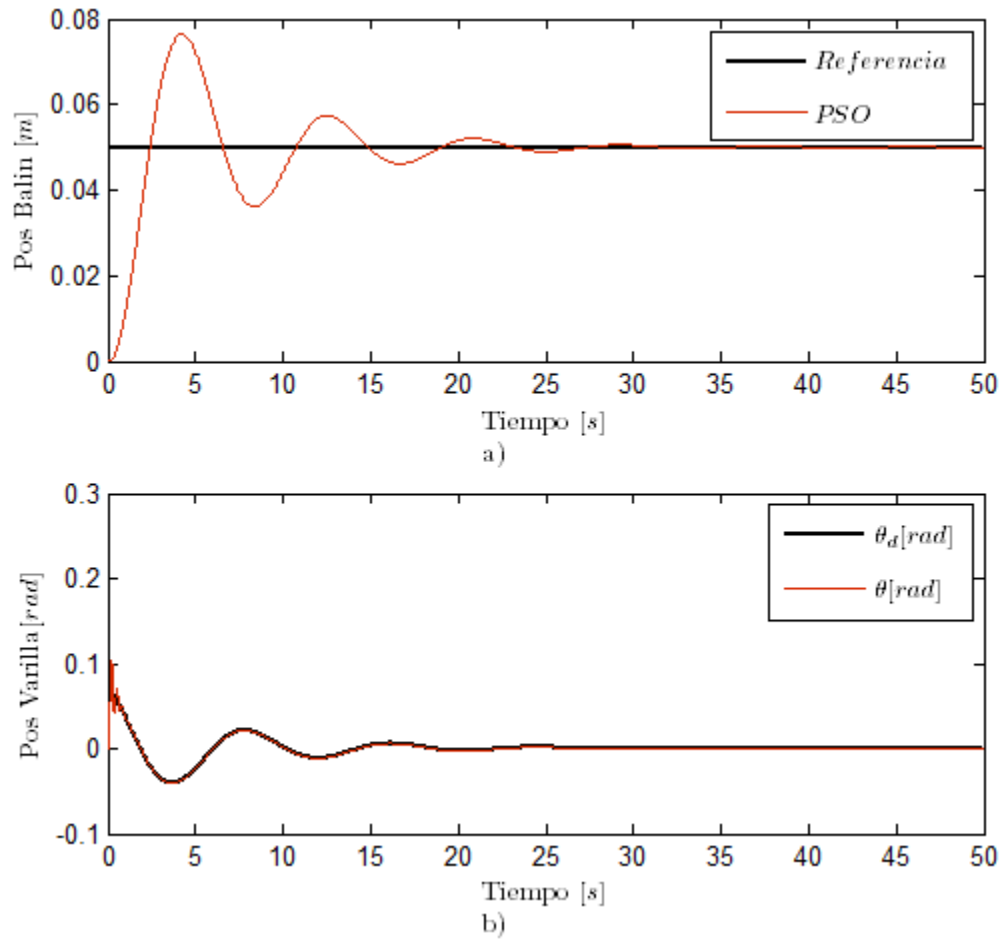


Figura 36: Realizado con el algoritmo PSO a) Respuesta del balón con la sintonización del controlador PID b) respuesta del riel.

De acuerdo a los resultados obtenidos el algoritmo PSO tiene un número de iteraciones mayor que el algoritmo genético. Como se muestra en la figura 37 el PSO encuentra una solución a partir de la iteración treinta.

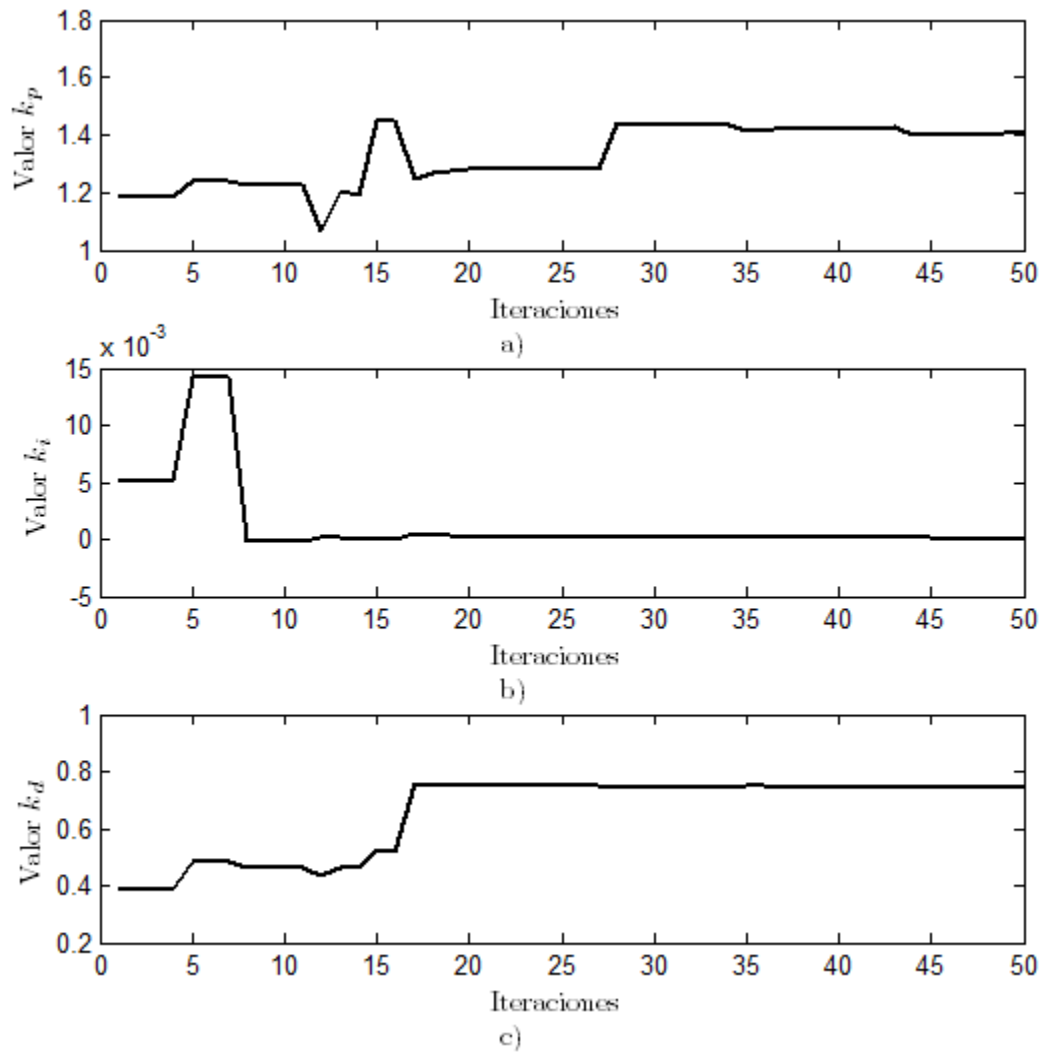


Figura 37: Evolución de las ganancias de controlador PID con el PSO.

En la figura 38 se muestra la evolución de la señal de control donde su amplitud esta dentro de los limites de voltaje del prototipo experimental (± 10 volts).

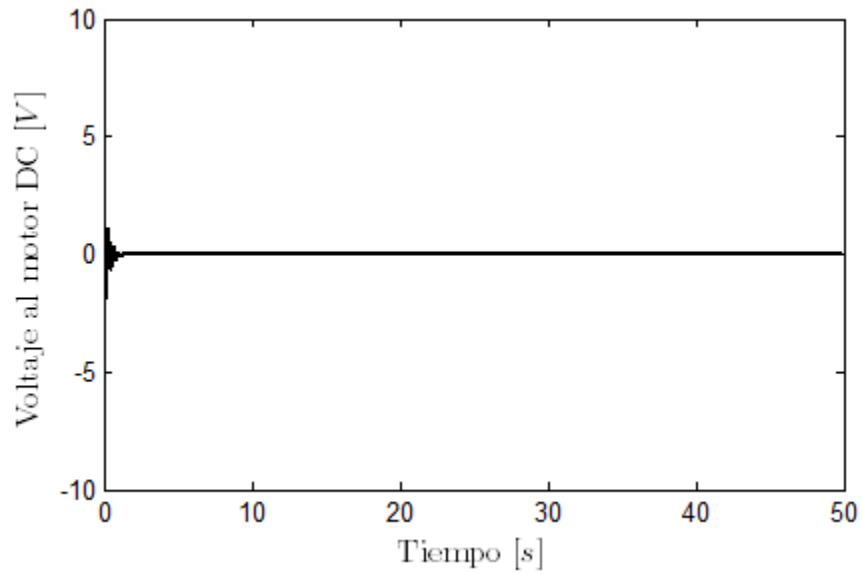


Figura 38: Señal de control con el algoritmo PSO.

En las dos siguientes tablas se muestra los resultados de cada algoritmo que se utilizó. En la tabla 4.1 se muestra los resultados del algoritmo genético y en la tabla 4.2 se muestra los resultados del algoritmo PSO. De estas tablas podemos concluir que el algoritmo PSO tiene un mejor desempeño para sintonizar el controlador PID del sistema *Ball and Beam*. De acuerdo a los resultados, las soluciones encontradas por el algoritmo PSO presentan una mejor aptitud comparados con los del algoritmo genético.

Prueba	Controlador PID			Controlador PD		ISE
	kp	ki	kd	kp	kd	
1	1.426	0.00035444	0.7262	48	0.8	0.5363
2	1.3077	0.0018	0.696	48	0.8	0.9964
3	1.8956	0.0024	0.8935	48	0.8	0.6179
4	0.5511	0.00011966	0.5839	48	0.8	0.7182
5	1.3909	0.00037731	0.7221	48	0.8	0.5448
6	1.2951	0.0017	0.8307	48	0.8	0.7412
7	1.4471	0.0015	0.6826	48	0.8	0.8032
8	1.6162	0.00044816	0.7591	48	0.8	0.5152
9	1.2114	0.0014	0.6397	48	0.8	0.961

Tabla 4.1. Resultados del algoritmo genético.

Prueba	Controlador PID			Controlador PD		ISE
	kp	ki	kd	kp	kd	
1	1.4531	0.00037892	0.8767	48	0.8	0.4628
2	1.1925	1.5859E-05	0.8455	48	0.8	0.4723
3	1.07228	0.00035558	0.8755	48	0.8	0.5003
4	1.3691	4.7852E-05	0.8117	48	0.8	0.4609
5	1.6134	0.00013145	0.8672	48	0.8	0.4358
6	1.5663	0.00026519	8.8767	48	0.8	0.4454
7	1.3109	0.00068	1.0103	48	0.8	0.461
8	1.4293	6.1969E-06	0.7477	48	0.8	0.4811
9	1.6891	0.00033279	0.8911	48	0.8	0.4395

Tabla 4.2. Resultados del algoritmo PSO.

RESULTADOS EXPERIMENTALES

Resultados con controlador PD en cascada

Una estrategia de control que se utilizó en una primera etapa fue el controlador PD en ambos lazos de control los resultados obtenidos se presentan en las figuras 39, 40 y 41. Las ganancias de los controladores PD fueron seleccionadas de forma empírica y son las siguientes.

Controlador primario

$$k_{p1} = 2 \text{ y } k_{d1} = 0.5$$

Controlador secundario

$$k_{p2} = 48 \text{ y } k_{d2} = 0.8$$

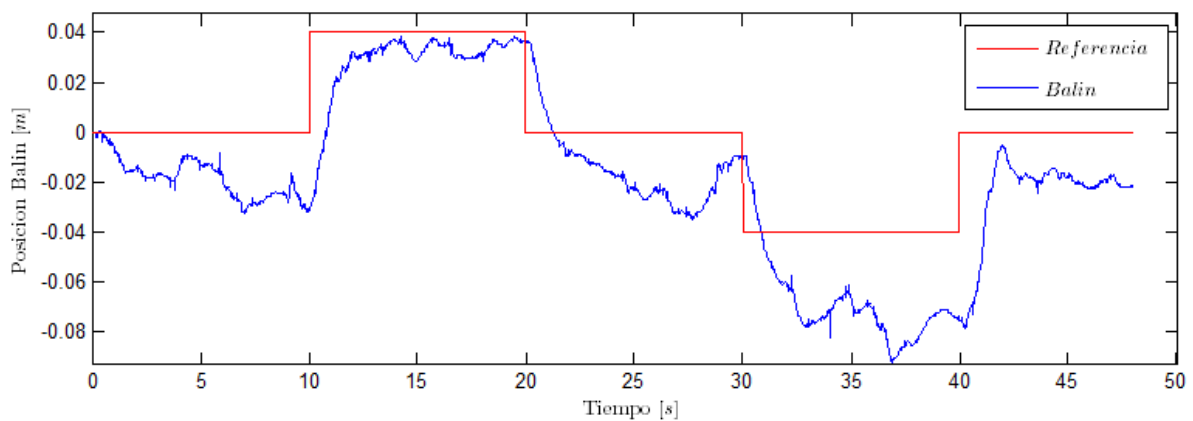


Figura 39: Posición del balón con controladores PD.

Como se ha mencionado la sintonización de los controladores no es trivial. Por ello, los resultados pueden presentar un desempeño pobre como se muestra en la figura 39. Además los controladores PD por su naturaleza presentan errores en estado estacionario. Por ello se eligió un controlador PID para contrarrestar este problema.

En la figura 40 se muestra la respuesta del lazo secundario, es decir, el control de la posición del riel.

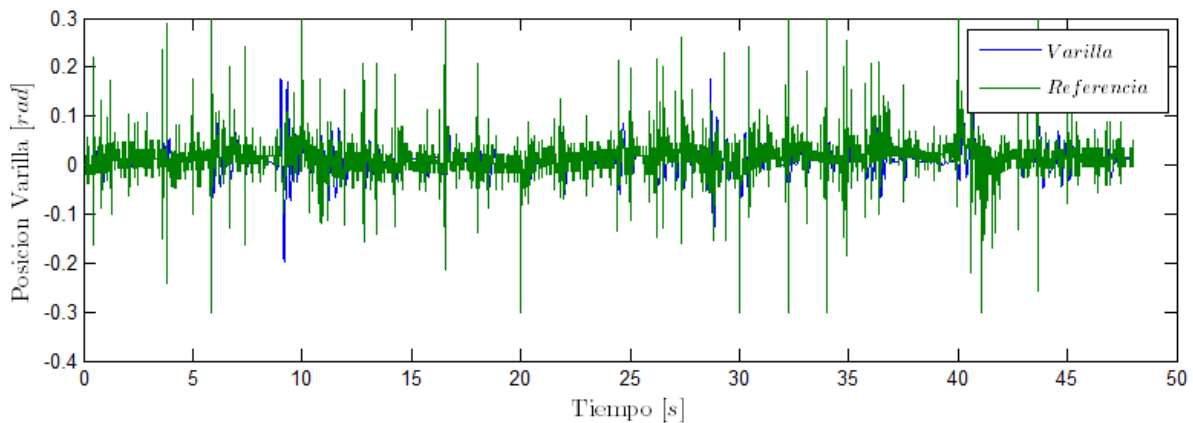


Figura 40: Posición del riel con controladores PD.

Como se ha mencionado el sistema *Ball and Beam* requiere de velocidades pequeñas para evitar el fenómeno de la fuerza centrífuga del balón. Para esto, las señales de control están acotadas, es decir, se les aplicó una función de saturación. En el caso de la posición del riel no puede ir más de ± 0.3 radianes como se muestra en la figura 41.

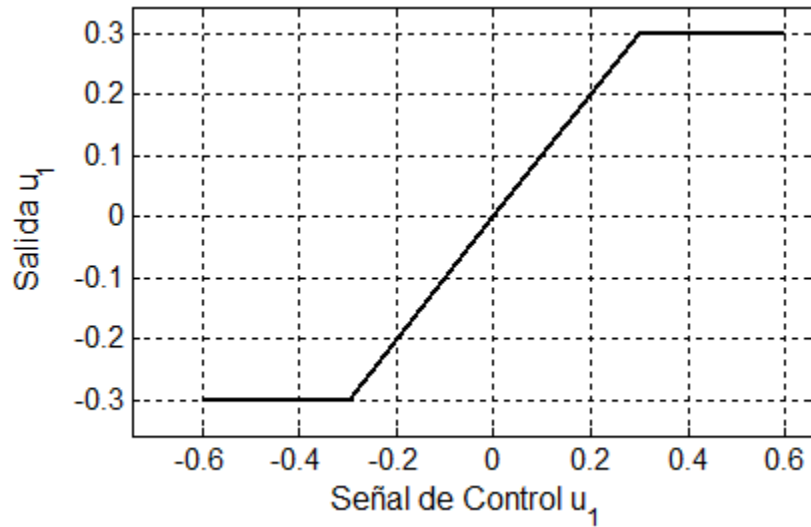


Figura 41: Saturación de la señal de control u_1 .

En este trabajo la señal de control se traduce en voltaje que se le aplica al motor eléctrico de DC, esta señal se muestra en la siguiente figura.

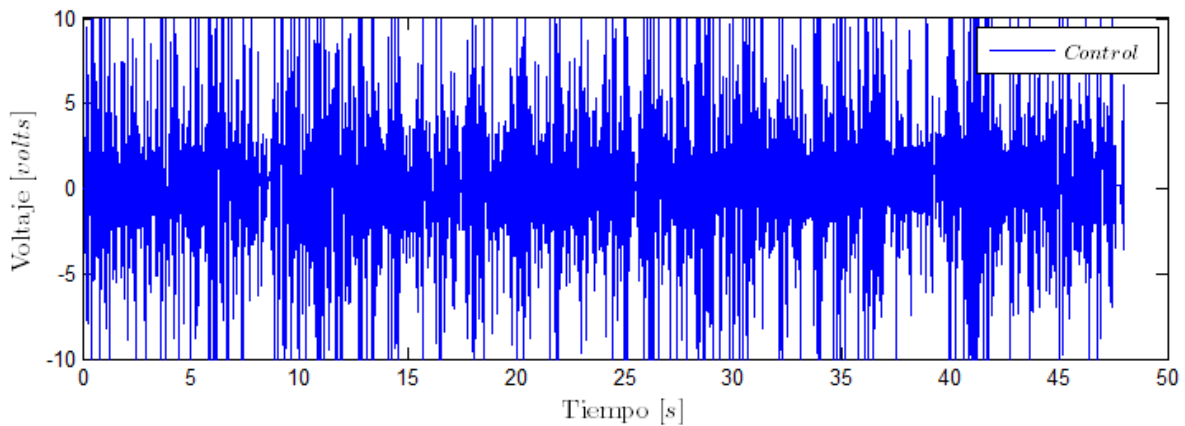


Figura 42: Señal de control para el *Ball and Beam* generado por los controladores PD.

La señal de control se acota por una función de saturación (figura 43), en este caso la señal de control equivale al voltaje aplicado al motor de DC el cual los valores máximos son de ± 10 volts.

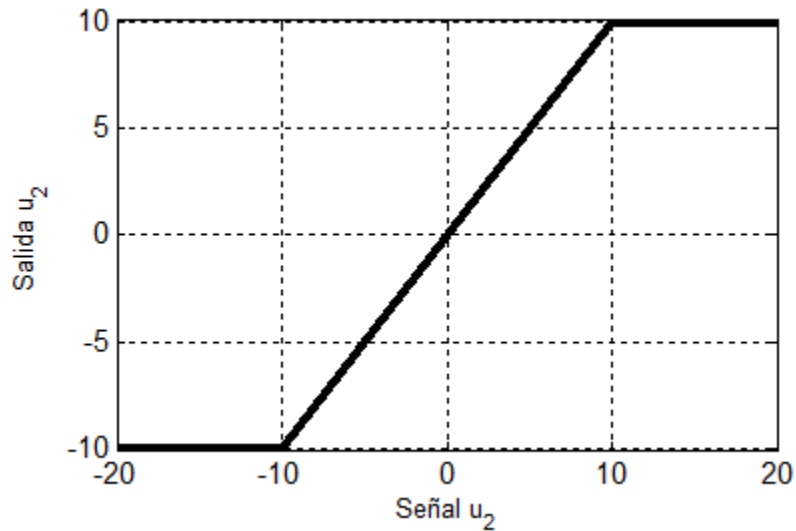


Figura 43: Saturación de la señal de control u_2 .

Resultados del algoritmo PSO

Los resultados que se presentan a continuación son resultados que se obtuvieron sintonizando el controlador PID con el algoritmo PSO. En la figura 44, 45 y 46 el algoritmo PSO encontró las siguientes ganancias.

$$k_p = 1.1834, \quad k_i = 0.0011 \quad y \quad k_d = 0.3998$$

Con las ganancias del controlador PD fijas en:

$$k_p = 48 \quad y \quad k_d = 0.8$$

En la figura 44 se muestra la posición del balón, de acuerdo a la figura la posición de referencia cambian cada 10 segundos en los primeros 40 segundos donde el balón trata de seguir la referencia.

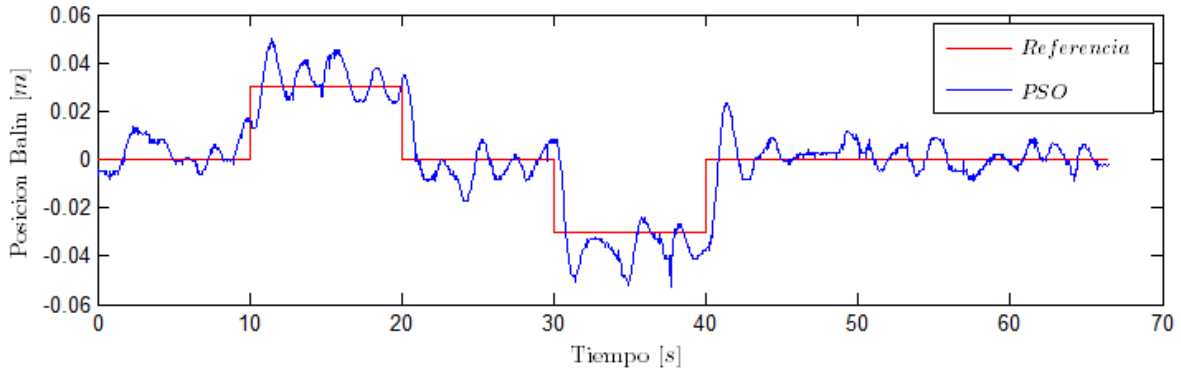


Figura 44: Posición del balón con la sintonización del controlador PID a través del PSO. Primera prueba.

En la figura 45 se muestra la posición del riel del *Ball and Beam*, como se mencionaba la posición del riel debe estar en cero o muy cercano a éste. Y como se muestra en la figura el rango es pequeño esto hace que cumpla con la condición $\sin \theta \approx \theta$.

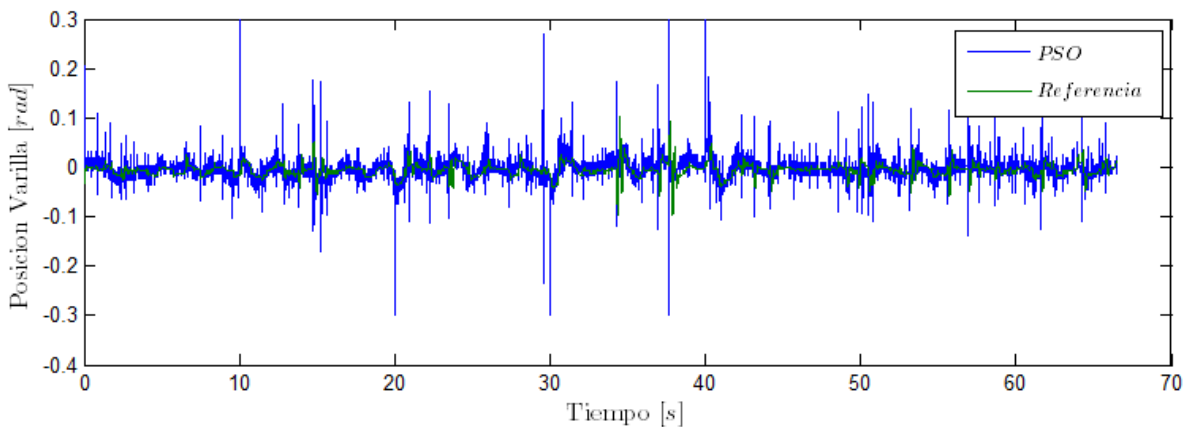


Figura 45: Posición del riel con algoritmo PSO. Primera prueba.

En la figura 46 se muestra la señal de control, el rango de esta señal es de ± 10 volts que se le aplica al motor de DC.

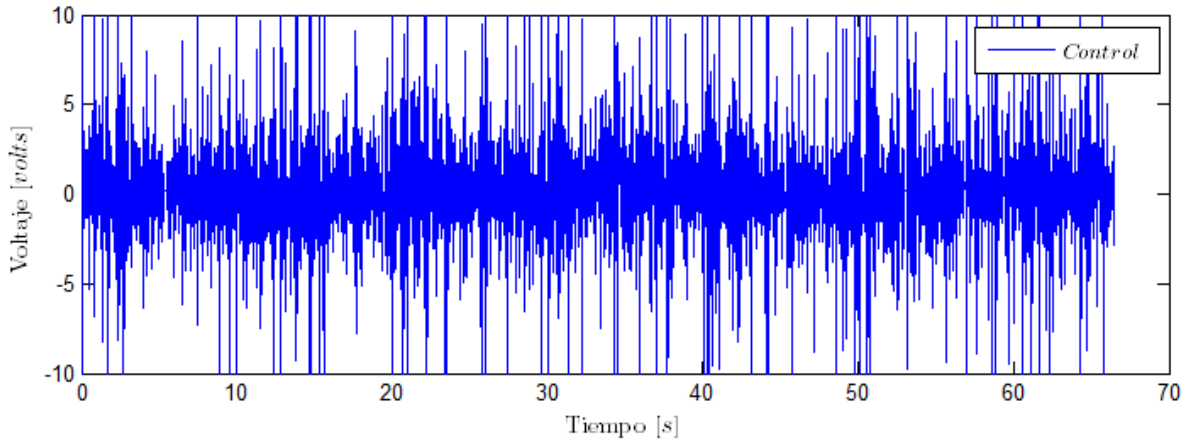


Figura 46: Señal de control con algoritmo PSO. Primera prueba.

En la figura 47, 48 y 49 se muestran los resultados de una segunda prueba que se realizó con el algoritmo PSO. Las ganancias para el controlador PID encontradas por el algoritmo PSO en su búsqueda son las siguientes.

$$k_p = 1.972, \quad k_i = 0.0056 \quad y \quad k_d = 0.6987$$

Con las ganancias del controlador PD fijas en:

$$k_p = 48 \quad y \quad k_d = 0.8$$

Al implementar un controlador de tipo PID hace que el error en estado estacionario tienda a cero cuando el tiempo tiende a infinito. En la figura 47 la posición del balón trata de permanecer en la señal de referencia. Por otra parte, en la figura 48 se muestra la posición del riel que de igual forma sigue la señal de referencia. Esta señal es generada por el controlador PID. Posteriormente, en la figura 49 se muestra la señal de control que se le introduce al motor de DC.

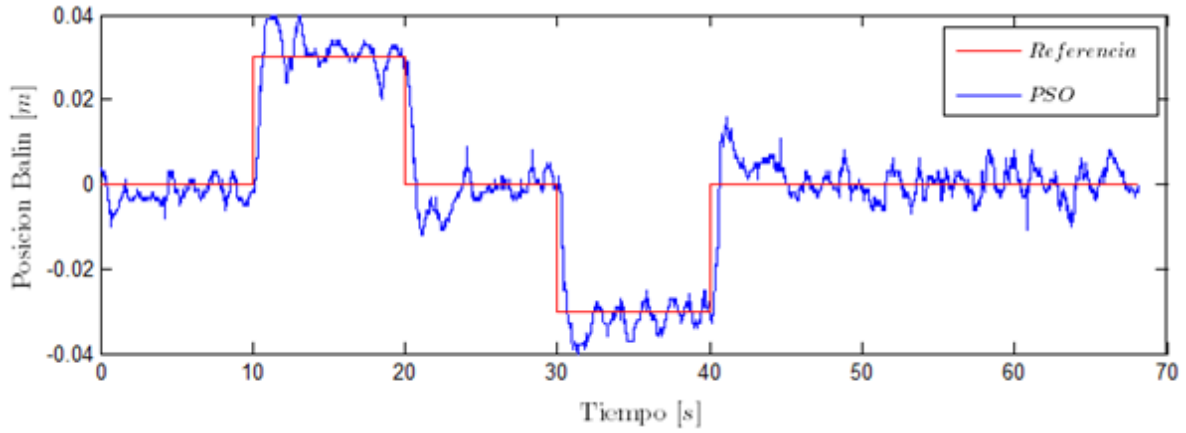


Figura 47: Posición del balón con la sintonización del controlador PID a través del PSO. Segunda prueba.

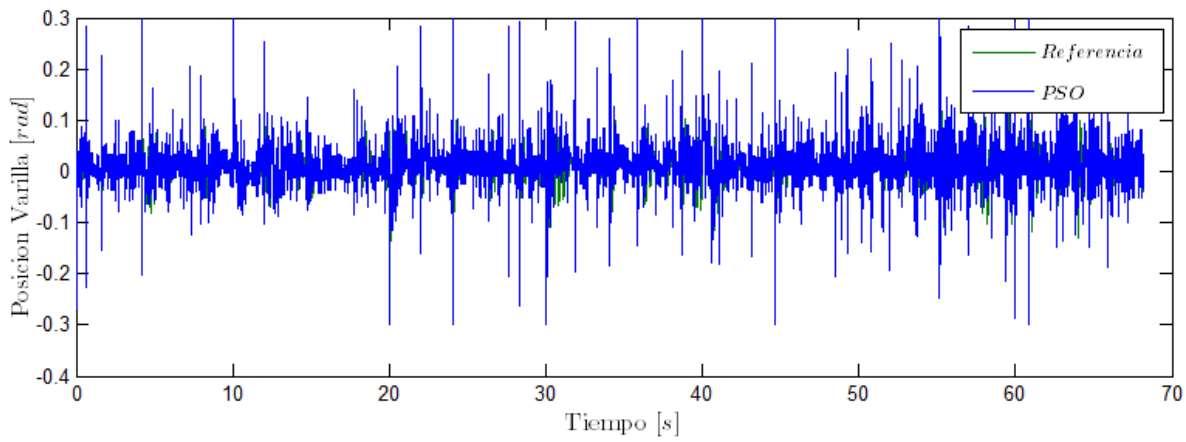


Figura 48: Posición del riel con algoritmo PSO. Segunda prueba.

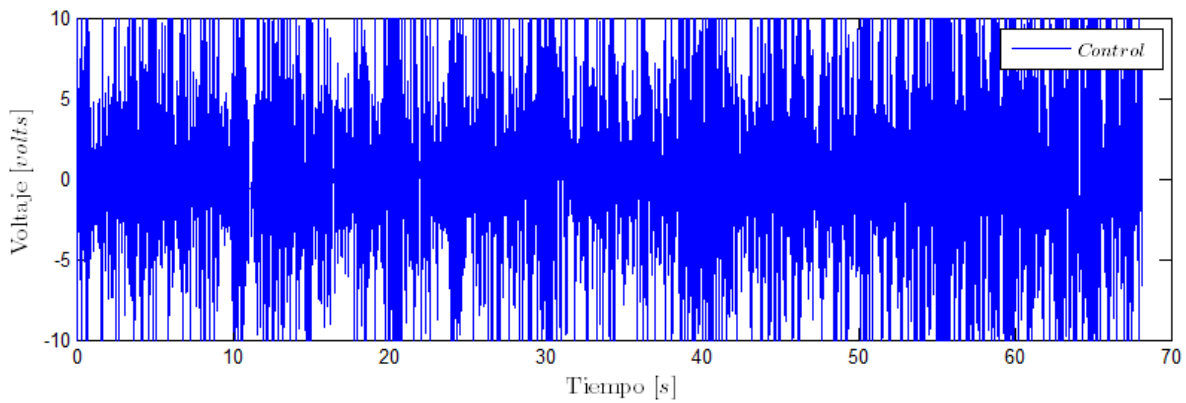


Figura 49: Señal de control con algoritmo PSO. Segunda prueba.

CONCLUSIÓN

En este trabajo se presentó un algoritmo heurístico que sintoniza un controlador PID para el control de un *Ball and Beam*. En una primera etapa se eligió dos algoritmos heurísticos: el Algoritmo Genético (AG) y la Optimización por Cúmulo de Partículas (PSO) que de acuerdo a la literatura estos algoritmos presentan un desempeño satisfactorio para sintonizar controladores PID en línea, además de encontrar soluciones satisfactorias estos también encuentran la solución en pocas interacciones. Por otra parte, el algoritmo PSO y AG son flexibles debido a que pueden adaptarse a diversos problemas de control. Comparadas con las redes neuronales estos algoritmos son rápidos en su búsqueda debido a que su espacio de búsqueda es sobre una región aleatoria en cambio las redes neuronales su búsqueda es de forma exhaustiva, por ello, los algoritmos heurísticos requieren pocos recursos computacionales y pueden ser implementados en dispositivos con bajos recursos computacionales. De acuerdo a las simulaciones realizadas el algoritmo PSO encontró soluciones con mejores aptitudes que las soluciones del AG, sin embargo, la convergencia del algoritmo genético son en pocas interacciones. Entonces si se requiere atacar un problema de control en base a los tiempos de convergencia del algoritmo heurístico, el algoritmo genético tiene esta ventaja sobre el algoritmo PSO. Sin embargo, si en la aplicación es primordial el desempeño del sistema, como es el caso el sistema *Ball and Beam* entonces el algoritmo PSO es la herramienta que presenta ventajas sobre el algoritmo genético. Para una segunda etapa se implementa el algoritmo heurístico que presentó una mejor aptitud en la solución encontrada, es el caso del algoritmo PSO. Donde los resultados obtenidos fueron satisfactorios debido a que las soluciones encontradas por el algoritmo PSO para sintonizar el controlador PID fueron capaces de estabilizar el sistema *Ball and Beam*.

De acuerdo a los resultados obtenidos el algoritmo PSO se considera como una herramienta poderosa para sintonizar controladores PID en línea aplicados a sistemas subactuados, inestables en lazo abierto, con controladores en cascada y en donde las acciones de control no sean abruptas, como es el caso del sistema *Ball and Beam*.

TRABAJOS A FUTURO

En este trabajo los parámetros fueron estimados de forma experimental. Sin embargo, se sabe que los parámetros pueden sufrir cambios por un deterioro físico, ha esto en control se dice que los parámetros son variantes en el tiempo. Por lo tanto, si los parámetros sufren algún cambio entonces el desempeño del sistema *Ball and Beam* también será afectado aún cuando el algoritmo PSO presente soluciones satisfactorias. Para contrarrestar este problema, este trabajo puede ser complementado con un estimador de parámetros, ya sea por alguna técnica de control o por algún algoritmo inteligente. De tal manera, que el estimador de parámetros retroalimente el algoritmo PSO que sintoniza el controlador PID y con ello el desempeño del sistema *Ball and Beam* no sea afectado por cambios en los parámetros.

También se ha mencionado que el sistema *Ball and Beam* cuenta con dos controladores en cascada. Siendo así, en este trabajo sólo se sintoniza el controlador PID dejando afuera el controlador PD que también puede ser sintonizado con algún algoritmo heurístico.

Otro de los trabajos que se deja a considerar es implementar controladores no lineales al sistema *Ball and Beam* y sintonizar estos controladores con los algoritmos heurísticos.

APENDICE

SISTEMA BALL AND BEAM EN TIEMPO DISCRETO

Para representar este modelo en tiempo discreto, es decir, aplicar la transformada z antes se debe considerar algunos factores tal como el tiempo de muestreo y el tipo de retenedor que se desea utilizar.

En este trabajo el tiempo de muestreo es de $T = 10$ milisegundos y se utiliza un retenedor de orden cero que se presenta a continuación:

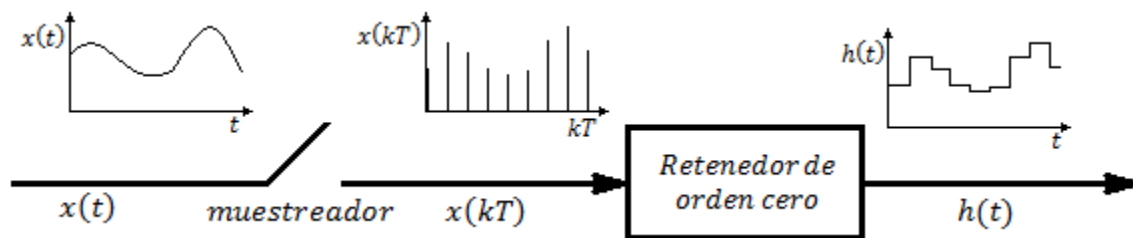


Figura 9: Muestreador y retenedor de orden cero.

Retenedor de orden cero. En la figura 50 se muestra un muestreador y retenedor de orden cero. La señal de entrada $x(t)$ se muestrea en instantes discretos y la señal muestreada se pasa a través del retenedor de orden cero. La cual es constante desde el último valor muestreada hasta que se puede disponer de la siguiente muestra. Esto es,

$$h(kT + t) = x(kT), \quad \text{para} \quad 0 \leq t \leq T$$

El modelo matemático de la combinación del muestreador y el circuito de retenedor de orden cero se muestra en la figura 51.

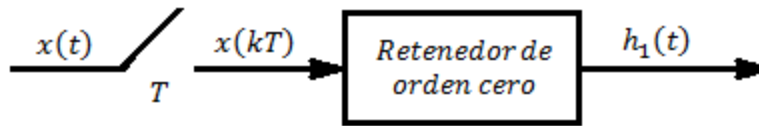


Figura 10: Función del muestreador y retenedor de orden cero.

A partir del hecho de que la integral de una función impulso es una constante, entonces se puede suponer que el retenedor de orden cero es un integral y la entrada del circuito del retenedor de orden cero es un tren de impulsos. Por lo tanto un modelo matemático del muestreador y retenedor de orden cero se puede construir como se muestra en la figura 52, donde $G_{h0}(s)$ es la función de transferencia del retenedor de orden cero y $x^*(t)$ es la señal muestreada mediante impulsos de $x(t)$.



Figura 11: Modelo matemático del muestreador y retenedor de orden cero.

La función de transferencia en s del retenedor de orden cero está dada por

$$G_{h0} = \frac{1 - e^{-Ts}}{s} \quad (\text{A.1})$$

El sistema *Ball and Beam* se representa por dos modelos matemáticos donde cada modelo se discretiza por separado. Esta tarea se presenta a continuación

Motor de DC con carga en tiempo discreto

A continuación se presenta el modelo matemático de un motor de DC con carga, el cual se representa en Laplace como:

$$\frac{\theta(s)}{I(s)} = \frac{k}{s(s+a)} \quad (\text{A.2})$$

En la siguiente figura se muestra el modelo matemático del motor de DC con carga y retenedor de orden cero con tiempo de muestreo $T = 10\text{ms}$.

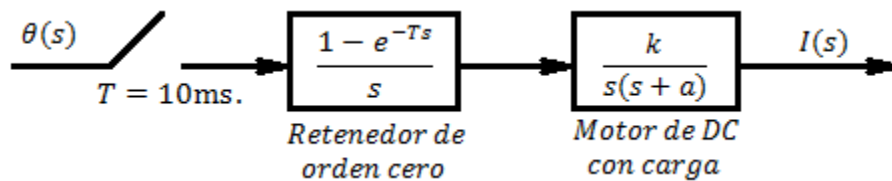


Figura 12: Diagrama de un motor de DC con retenedor de orden cero y muestreo de **10ms**.

Reduciendo el modelo anterior tenemos que

$$G_1(s) = \frac{\theta(s)}{I(s)} = \frac{1 - e^{-Ts}}{s} \frac{k}{s(s+a)} \quad (\text{A.3})$$

El modelo anterior se le aplica la transformada z , esto es como sigue.

$$G_1(z) = \frac{\theta(z)}{I(z)} = Z \left[(1 - e^{-Ts}) \frac{k}{s^2(s+a)} \right]$$

$$G_1(z) = Z[(1 - e^{-Ts})] Z \left[\frac{A}{s} + \frac{B}{s^2} + \frac{C}{s+a} \right] \quad (\text{A.4})$$

Donde $A = -\frac{k}{a^2}$, $B = \frac{k}{a}$ y $C = \frac{k}{a^2}$

Si

$$Z[e^{Ts}] = z, \quad Z \left[\frac{1}{s} \right] = \frac{1}{1 - z^{-1}}, \quad Z \left[\frac{1}{s^2} \right] = \frac{Tz^{-1}}{(1 - z^{-1})^2} \quad (\text{A.5})$$

Y

$$Z \left[\frac{1}{s+a} \right] = \frac{1}{1 - e^{-aT} z^{-1}} \quad (\text{A.6})$$

Entonces

$$G_1(z) = (1 - z^{-1}) \left(\frac{A}{1 - z^{-1}} + \frac{BTz^{-1}}{(1 - z^{-1})^2} + \frac{C}{1 - e^{-aT} z^{-1}} \right) \quad (\text{A.7})$$

Reduciendo la ecuación (A.7) tenemos

$$G_1(z) = A + \frac{BTz^{-1}}{1 - z^{-1}} + \frac{(1 - z^{-1})C}{1 - e^{-aT} z^{-1}} \quad (\text{A.8})$$

$$= \frac{A(1 - z^{-1})(1 - e^{-aT} z^{-1}) + BTz^{-1}(1 - e^{-aT} z^{-1}) + C(1 - z^{-1})^2}{(1 - z^{-1})(1 - e^{-aT} z^{-1})} \quad (\text{A.9})$$

$$G_1(z) = \frac{(A + C) + (-A - Ae^{-aT} + BT - 2C)z^{-1} + (Ae^{-aT} - BTe^{-aT} + C)z^{-2}}{1 - (1 + e^{-aT})z^{-1} + e^{-aT}z^{-2}} \quad (\text{A.10})$$

Sustituyendo los parámetros $k = 7.8296$, $a = 2.4706$ del sistema *Ball and Beam* y tiempo de muestreo de $T = 10$ ms, tenemos que:

$$A = -1.2827, \quad B = 3.1691 \quad \text{y} \quad C = 1.2827$$

Por lo tanto

$$G_1(z) = \frac{\theta(z)}{I(z)} = \frac{0.0003882z^{-1} + 0.000385z^{-2}}{1 - 1.975596z^{-1} + 0.975596z^{-2}} \quad (\text{A.11})$$

La ecuación (A.11) es la función de transferencia en transformada z del modelo del motor de DC con carga. A continuación se presenta la función discreta del modelo anterior para su programación directa.

$$\begin{aligned} (1 - 1.975596z^{-1} + 0.975596z^{-2})\theta(z) \\ = (0.0003882z^{-1} + 0.000385z^{-2})I(z) \end{aligned} \tag{A.12}$$

Aplicando la propiedad de la transformada z:

$$x(n + k) = z^{-k}X(z) \tag{A.13}$$

Tenemos que

$$\begin{aligned} \theta(n) = 1.975596\theta(n - 1) - 0.975596\theta(n - 2) \\ + 0.0003882I(n - 1) + 0.000385I(n - 2) \end{aligned} \tag{A.14}$$

La ecuación (A.14) es la función que representa al modelo dinámico del motor de DC con carga y es el que se programa en Matlab o en algún dispositivo electrónico.

Modelo del balón en tiempo discreto

El segundo modelo dinámico del *Ball and Beam* consiste en un balón que gira sobre un riel donde su modelo esta dado como:

$$\frac{X(s)}{\theta(s)} = \frac{\rho}{s^2} \quad (\text{A.15})$$

Para su conversión en la transformada z se diseña el siguiente modelo:

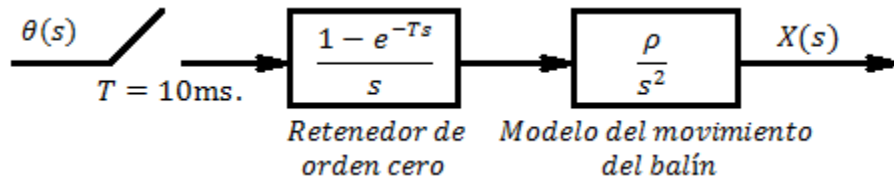


Figura 13: Diagrama del modelo del balón con retenedor de orden cero y muestreo de **10ms**.

Reduciendo el modelo anterior tenemos

$$G_2(s) = \frac{X(s)}{\theta(s)} = \frac{1 - e^{-Ts}}{s} \frac{\rho}{s^2} \quad (\text{A.16})$$

Aplicando la transformada z a la ecuación anterior, de la siguiente manera:

$$G_2(z) = \frac{X(z)}{\theta(z)} = Z \left[(1 - e^{-Ts}) \left(\frac{\rho}{s^3} \right) \right]$$

$$G_2(z) = Z[(1 - e^{-Ts})] Z \left[A \frac{2}{s^3} \right] \quad (\text{A.17})$$

Donde $A = \rho/2$

Si

$$Z[e^{Ts}] = z \quad y \quad Z \left[\frac{2}{s^3} \right] = \frac{T^2 z^{-1} (1 + z^{-1})}{(1 - z^{-1})^3} \quad (\text{A.18})$$

Entonces

$$\begin{aligned}
 G_2(z) &= (1 - z^{-1}) \left(\frac{AT^2 z^{-1} (1 + z^{-1})}{(1 - z^{-1})^3} \right) \\
 &= \frac{AT^2 z^{-1} (1 + z^{-1})}{(1 - z^{-1})^2} \\
 &= \frac{AT^2 z^{-1} + AT^2 z^{-2}}{1 - 2z^{-1} + z^{-2}} \tag{A.19}
 \end{aligned}$$

Sustituyendo el parámetro $\rho = 4.2$ del *Ball and Beam*, entonces $A = 2.1$ y si el tiempo de muestreo de $T = 10\text{ms}$, por lo tanto:

$$G_2(z) = \frac{X(z)}{\theta(z)} = \frac{0.00021z^{-1} + 0.00021z^{-2}}{1 - 2z^{-1} + z^{-2}} \tag{A.20}$$

Para su programación directa utilizando la propiedad de la transformada z , ecuación (A.20), tenemos que:

$$x(n) = 2x(n - 1) - x(n - 2) + 0.00021\theta(n - 1) + 0.00021\theta(n - 2) \tag{A.21}$$

La ecuación anterior es la función que representa al modelo de movimiento del balón y es el que se programa en Matlab o en algún dispositivo electrónico.

A continuación se presenta el algoritmo genético programado en Matlab.

CÓDIGO DEL ALGORITMO GENÉTICO

```
% Algoritmo Genético para el sistema Ball and Beam.
% Elaborado por: Salvador Ortiz Santos
% Dirigido por: Dr.Luis Alfonso Franco Gasca

clear
clc
n = 200; % Tamaño de población
dim = 3; % Dimensión del problema
n_ite = 25; % Número de iteraciones

P = rand(dim, n); % Generar tamaño de población.

P(1,1:n) = 2*rand(1, n); % Posición actual de la primera dimensión.
P(2,1:n) = 0.2*rand(1, n); % Posición actual de la segunda dimensión.
P(3,1:n) = 0.9*rand(1, n); % Posición actual de la tercera dimensión.

for ite = 1 : n_ite % Condición para finalizar las iteraciones

    % Analizar la aptitud de cada uno de los individuos
    for i = 1 : n
        aptitud_actual(i) = Modelo(P(:,i)); % Función para la aptitud.
    end

    [FA ind] = sort(aptitud_actual); % Ordenar de menor a mayor.

    % Selección de los padres que se reproducirán.
    for i = 1 : n
        Px(i) = P(1,ind(i));
        Py(i) = P(2,ind(i));
        Pz(i) = P(3,ind(i));
    end

    %Combinación y generación de los hijos.
    for i = 1 : n/4
        Hx1(i) = Py(101 - i);
        Hy1(i) = Px(i);
        Hz1(i) = Pz(101 - i);
        Hx2(i) = Py(i);
```

$H_{y2}(i) = P_x(101 - i);$

$H_{z2}(i) = P_z(i);$

end

P1 = Px;

P2 = Py;

P3 = Pz;

% Generación de la nueva población.

P1(101 : 150) = Hx1;

P2(101 : 150) = Hy1;

P3(101 : 150) = Hz1;

P1(151 : n) = Hx2;

P2(151 : n) = Hy2;

P3(151 : n) = Hz2;

% Nueva población.

P(1,1:n) = P1;

P(2,1:n) = P2;

P(3,1:n) = P3;

% Individuo con mejor aptitud.

kpv1(ite) = P1(1);

kiv1(ite) = P2(1);

kdv1(ite) = P3(1);

end

% Solución que se encontró en la búsqueda.

% Controlador PID

kp1 = P1(1) % Ganancia Proporcional

ki1 = P2(1) % Ganancia Integral

kd1 = P3(1) % Ganancia Derivativo

% Controlador PD

kp2 = 44 % Ganancia Proporcional

kd2 = 0.8 % Ganancia Derivativo

A continuación se presenta el algoritmo PSO

CÓDIGO DEL ALGORITMO PSO

%% Optimización por cúmulo de partículas para el Ball and Beam.

% Elaborado por: Salvador Ortiz Santos

% Dirigido por: Dr.Luis Alfonso Franco Gasca

clc;

clear all;

close all;

%% ----- Inicializar variables -----

n = 100; % Tamaño del cúmulo "No. de aves"

n_bird = 50; % Número máximo de iteraciones

dim = 3; % Dimensión del problema [kp,ki,kd]

c1 = 0.2; % Parámetro C1 del PSO

c2 = 0.2; % Parámetro C2 del PSO

w = 0.4; % Inercia del PSO

%-----%

% Inicializamos Parámetros del PSO %

%-----%

R1 = rand(dim, n); % rand 1 de 0-1

R2 = rand(dim, n); % rand 2 de 0-1

aptitud_actual = zeros(n,1);

%-----%

% Inicialización del cúmulo, velocidad y posición %

%-----%

pos_actual = 2*(rand(dim, n)); % Tamaño de la posición actual

```
% Inicializar los valores en un espacio de búsqueda
```

```
pos_actual(1,1:n) = 2*(rand(1,n));
```

```
pos_actual(2,1:n) = 0.2*rand(1,n);
```

```
pos_actual(3,1:n) = 0.9*rand(1,n);
```

```
velocidad = 0.5*rand(dim, n);
```

```
pos_local_best = pos_actual;
```

```
%-----%
```

```
% Evaluación inicial de la población %
```

```
%-----%
```

```
for i = 1 : n
```

```
    aptitud_actual(i) = Modelo(pos_actual(:,i)); % Aptitud inicial
```

```
end
```

```
aptitud_local_best = aptitud_actual; % La mejor Aptitud local
```

```
[aptitud_global_best, g] = min(aptitud_local_best); % Encontrar el líder.
```

```
for i = 1 : n
```

```
    pos_global_best(:,i) = pos_local_best(:,g); % La mejor Aptitud global.
```

```
end
```

```
%-----%
```

```
% Calcular la velocidad %
```

```
%-----%
```

```
velocidad = w*velocidad + c1*(R1.*abs(pos_local_best - pos_actual)) +  
c2*(R2.*abs(pos_global_best - pos_actual));
```

```
%-----%
```

```
% Calcular la posición %
```

```
%-----%
```

```
pos_actual = pos_actual + velocidad;
```

```
%-----%
```

```
% Evaluar una nueva partícula %
```

```
%-----%
```

```

%% Búsqueda
for ite = 1 : n_bird % condición para finalizar la búsqueda.

    % Evaluar la aptitud de cada partícula
    for i = 1 : n
        aptitud_actual(i) = Modelo(pos_actual(:,i)) ;
    end

    % Encontrar la mejor aptitud local
    for i = 1 : n
        if aptitud_actual(i) < aptitud_local_best(i)
            aptitud_local_best(i) = aptitud_actual(i);
            pos_local_best(:,i) = pos_actual(:,i);
        end
    end

    % Encontrar la partícula con la mejor aptitud actual
    [aptitud_actual_global_best, g] = min(aptitud_local_best);

    % Cambiar el líder o seguir con el mismo según su aptitud.
    if aptitud_actual_global_best < aptitud_global_best
        aptitud_global_best = aptitud_actual_global_best;
        for i = 1 : n
            pos_global_best(:,i) = pos_local_best(:,g);
        end
    end

    % Calcular la velocidad
    velocidad = w*velocidad + c1*(R1.*(pos_local_best - pos_actual)) +
c2*(R2.*(pos_global_best - pos_actual));

    % Calcular la posición
    pos_actual = pos_actual + velocidad;

    % Evolución de las partículas
    kpv1(ite) = pos_global_best(1,1);
    kiv1(ite) = pos_global_best(2,1);
    kdvl(ite) = pos_global_best(3,1);
end

```



```
[Jbest_min,I] = min(aptitud_actual) % Aptitudes mínimas  
pos_actual(:,I); % La mejor solución
```

```
% Solución encontrada por el PSO
```

```
% Controlador PID
```

```
kp1 = pos_actual(1,I) % Ganancia Proporcional
```

```
ki1 = pos_actual(2,I) % Ganancia integral
```

```
kd1 = pos_actual(3,I) % Ganancia Derivativo
```

```
% Controlador PD
```

```
kp2 = 48.0 % Ganancia Proporcional
```

```
kd2 = 0.8 % Ganancia Derivativo
```

FUNCIÓN DE APTITUD O CALIDAD

En el siguiente código se presenta la forma en el que se evalúa la aptitud de cada individuo.

```
function error = Modelo(k)
```

```
tm = 6000; %Tamaño del vector
```

```
%%----- Inicializar las variables -----
```

```
tetaM = 0.3; % teta máximo.
```

```
VM = 10; % Voltaje máximo.
```

```
teta = zeros(tm,1); % Posición angular del riel.
```

```
pos = zeros(tm,1); % Posición del Balín.
```

```
u1 = zeros(tm,1); % Señal del controlador PID.
```

```
u2 = zeros(tm,1); % Señal del controlador PD.
```

```
e1 = zeros(tm,1); % Señal de error del Balín.
```

```
e2 = zeros(tm,1); % Señal de error del riel.
```

```
error = 0; % ISE (Integral Square Error)
```

```
ts = 0.01; % Tiempo de muestreo.
```

```
ref = 0.05; % Referencia deseada.
```

```
% Controlador PID
```

```
kp1 = k(1); % Ganancia Proporcional
```

```
ki1 = k(2); % Ganancia Integral
```

```
kd1 = k(3); % Ganancia Derivativo
```

```

% Controlador PD
kp2 = 44; % Ganancia Proporcional
kd2 = 0.8; % Ganancia Derivativo
integral = 0; % Inicializar el integrador el cero.
%% Control ball and beam
for t = 3 : tm
    % Sistema Ball and Beam y controladores.
    teta(t) = 1.975596*teta(t - 1) - 0.975596*teta(t - 2) +
        0.00038827*u2(t - 1) + 0.000385091*u2(t - 2);
    pos(t) = 2*pos(t - 1) - pos(t - 2) + 0.00021*teta(t - 1) + 0.00021*teta(t - 2);

    e1(t) = ref - pos(t);

    integral = integral + ki1*e1(t);

    u1(t) = kp1*e1(t) + (kd1/ts)*(e1(t) - e1(t - 1)) + integral; % PID
    % Acortar señal del controlador PID
    if(u1(t) > tetaM)
        u1(t) = tetaM;
    elseif (u1(t) < -tetaM)
        u1(t) = -tetaM;
    end;

    e2(t) = u1(t) - teta(t);
    u2(t) = kp2*e2(t) + (kd2/ts)*(e2(t) - e2(t - 1)); % PD

    % Acotar la señal de Voltaje
    if (u2(t) > VM)
        u2(t) = VM;
    elseif(u2(t) < - VM)
        u2(t) = -VM;
    end;
    error = error + e1(t)*e1(t); % ISE (Integral Squared Error)
end

```

BIBLIOGRAFÍA

Acedo-Sánchez, J. 2006. *Instrumentación*. Editorial: Diaz de Santos.

Amjad, M., M. I. Kashif, S. S. Abdullah, and Z. Shareef. 2010. *Fuzzy Logic Control of Ball and Beam System*. 2nd International Conference on Education Technology and Computer (ICETC). IEEE: 978-1-4244-6370-1.

Asif-Rana, M., Z. Usman, and Z. Shareef. 2011. *Automatic control of ball and beam system using particle swarm optimization*. 12th International Symposium on Computational Intelligence and Informatics. IEEE: 978-1-4577-0045-3/11.

Aström, K. J., and T. Hägglund. 2001. *The future of PID control*. Control Engineering Practice, (9):1163-1165.

Aström, K. J., and T. Hägglund. 2006. *Advanced PID Control*. Instrument Society of America.

Aström, K. J., and T. Hägglund. 1995. *PID Controllers: Theory, Design and Tuning*. Instrument Society of America.

Barbu. C., R. Sepulchre, W. Lin, and P. V. Kokotović. 1997. *Global asymptotic stabilization of the ball and beam system*. 36th Conference on Decision & Control, San Diego, California USA. IEEE: 0-7803-3970-81/97.

Bobál, V., J. Böhm, J. Fessl, and J. Macháček. 2005. *Digital Self-tuning Controllers. Algorithms, Implementation and Applications*. Springer-Verlag.

Chen, W., X. Sui, and Y. Xing. 2012. *Modeling and modulation of nonlinear ball-beam system controller based on matlab*. 9th International Conference on Fuzzy Systems and Knowledge Discovery, IEEE: 978-1-4673-0024-7/10.

Cheng-Ching, Y. 2006. *Autotuning of PID controllers. A relay feedback approach*. 2nd Edition, Springer.

Duc-Hoang, N., and H. Thai-Hoang. 2008. *A SFLA-Based fuzzy controller for balancing a ball and beam system*. 10th Intl. Conf. on Control, Automation, Robotics and Vision Hanoi, Vietnam. IEEE: 978-1-4244-2287-6/08.

Enzeng, D., G. Shuxiang, L. Xichuan, L. Xiaoqiong, and W. Yunliang. 2012. *A Neural Network-based Self-tuning PID Controller of an Autonomous Underwater Vehicle*. International Conference on Mechatronics and Automation. Chengdu, China. IEEE: 978-1-4673-1278-3/12.

Fang, H., L. Chen, and Z. Shen. 2011. *Application of an improved PSO algorithm to optimal tuning of PID gains for water turbine governor*. Energy Conversion and Management 52 (2011) 1763–1770.

Glover, F., and G. A. Kochenberger. 2003. *HANDBOOK OF METAHEURISTICS* Kluwer Academic Publishers

Hernández-Guzmán, V. M., R. Silva-Ortigoza, and R. V. Carrillo-Serrano. 2013. *Control Automático: Teoría de diseño, construcción de prototipos, modelado, identificación y pruebas experimentales*. CIDETEC, Primera edición.

Hongqing, F., C. Long, and S. Zuyi. 2011. *Application of an improved PSO algorithm to optimal tuning of PID gains for water turbine governor*. Energy Conversion and Management 52 (2011) 1763–1770.

Hultmann-Ayala, H. V., and L. D. Santo-Coelho. 2012. *Tuning of PID controller based on a multiobjective genetic algorithm applied to a robotic manipulator*. Expert Systems with Applications 39 (2012) 8968–8974.

Iruthayarajan, M. W., and S. Baskar. 2009. *Evolutionary algorithms based design of multi-variable PID controller*. Expert Systems with Applications 36 (2009) 9159–9167.

Jie, Y., and W. Zhong-Hua. *Adaptive robust dissipative desing on stability control for ball and beam system*. IEEE: 978-1-4244-5586-7/10.

Juing-Shian, C., T. Shun-Hung, and L. Ming-Tang. 2012. *A PSO-based adaptive fuzzy PID-controllers*. Simulation Modelling Practice and Theory 26 (2012) 49–59.

Kannan, M. 2007. *Digital Control*. Wiley.

Katsuhiko, O. 1995. *Discrete-Time Control Systems*. Prentice Hall, Inc, second edition.

Liguo, Q., H. Haibo, and H. Yourui. 2010. *Fractional Order PID controller based on Parti-cle Swarm Optimization Implemented with FPGA*. International Conference on Artificial Intelligence and Computational Intelligence. IEEE: 978-0-7695-4225-6/10.

Mohammad, K., J. Ali-Fellah, M. Abolfazl, A. Mohammad-Hadi, and X. Wen-Fang. 2012. *Modeling and control of ball and beam system using model based and non-model based control approaches*. International journal on smart sensing and intelligent systems. ISSN 1178-5608.

Naoyuky, H., T. Masaaki, and K. Keiji. 2011. *Experimental Evaluation of Model Predictive Control of Ball and Beam Systems*. American Control Conference on O'Farrell Street , San Francisco, CA, USA. IEEE: 978-1-4577-0081-1/11.

Nganga-Kouya, D., and F. A. Okou. 2011. *A new adaptive state feedback controller for the ball and beam system*. IEEE CCECE: 9978-1-4244-9789-8/11.

Nurbaiti, W., and H. Nurhaffizah. 2012. *Self-tuning Fuzzy PID Controller Design for Aircraft Pitch Control*. 3th International Conference on Intelligent Systems Modelling and Simulation. IEEE: 978-0-7695-4668-1/12.

Ortega, R., M. W. Spong, F. Gómez-Estern, and Blankenstein. 2002. *Stabilization of a Class of Underactuated Mechanical Systems Via Interconnection and Damping Assignment*. Transactions on automatic control, IEEE: 0018-9286/02.

Qu, L., H. Hu, and Y. Huang. 2010. *Fractional order PID controller based on Particle Swarm Optimization implemented with FPGA*. 2010 International Conference on Artificial Intelligence and Computational Intelligence IEEE: 978-0-7695-4225-6/10

Luna-Ortiz, C. R. 2011. *Diseño e Implementación de un algoritmo genético en FPGA para sintonización de controladores PID*. Tesis, Universidad Autónoma de Querétaro.

Santos-Coelho, L., and M. Wicthoff-Pessôa. 2011. *A tuning strategy for multivariable PI and PID controllers using differential evolution combined with chaotic Zaslavskii map*. Expert systems with Applications 38: 13694–13701.

Sivanandam, S. N., and S. N. Deepa. 2008. *Introduction to Genetic Algorithms*. Springer-Verlag.

Sung-Kwun, O., J. Han-Jong, and P. Witold. 2009. *The design of a fuzzy cascade controller for ball and beam system: A study in optimization with the use of parallel genetic algorithms*. Engineering Applications of Artificial Intelligence 22 (2009) 261–271.

Tae-Hyoung, K., M. Ichiro, and S. Toshiharu. 2008. *Robust PID controller tuning based on the constrained particle swarm optimization*. Automatica 44 (2008) 1104 – 1110.

Ting-Li, C., C. Chung-Cheng, T. Ming-Chun, and C. Ying-Chen. 2010. *Control of AMIRA's ball and beam system via improved fuzzy feedback linearization approach*. Applied Mathematical Modelling 34 (2010) 3791–3804.

Wei-Der C. 2007. *A multi-crossover genetic approach to multivariable PID controller tuning*. Expert Systems with Applications 32 (2007) 620-626.

Wei-Der C., and S. Shun-Peng. 2010. *PID controller design of nonlinear systems using an improved particle swarm optimization approach*. Commun Nonlinear Sci Numer Simulat 15 (2010) 3632–3639.

Wieneke, J., and W. N. White. 2011. *A feasibility assessment of using ultrasonic sensor position feedback for a ball and beam apparatus*. American control conference on O'Farrell Street, San Francisco, CA, USA. IEEE: 978-1-4577-0081-1/11.

Willjuice-Iruthayarajan M., and S. Baskar. 2009. *Evolutionary algorithms based design of multivariable PID controller*. Expert Systems with Applications 36 (2009) 9159-9167.

Yajuan, C., and W. Qinghai. 2011. *Design of PID Controller Based on PSO Algorithm and FPGA*. 2nd International conference on intelligent control and information processing. IEEE: 978-1-4577-0816-9.

Yeong-Hwa, C., C. Chia-Wen, T. Chin-Wang, L. Hung-Wei, and T. Jin-Shiuh. 2012. *Fuzzy sliding-mode control for ball and beam system with fuzzy ant colony optimization*. Expert Systems with Applications 39 (2012) 3624–3633.

Yeong-Hwa C., C. Wei-Shou, and C. Chia-Wen. 2013. *T-S Fuzzy Model-Based Adaptive Dynamic Surface Control for Ball and Beam System*. Transactions on industrial electronics. IEEE: TIE.2012.2192891.