

**ALGORITMO DE ENJAMBRE DE PARTÍCULAS UNIFICADO PARA LA
SOLUCIÓN DE ECUACIONES DIOFÁNTICAS LINEALES COMÚNMENTE
ENCONTRADAS EN PROBLEMAS DE INGENIERÍA ELECTRÓNICA**

Autores:

JONATHAN JAVIER RAMÍREZ ZAMBRANO

FAUSTO OSORIO SILVA

UNIVERSIDAD INDUSTRIAL DE SANTANDER

FACULTAD DE INGENIERIAS FISICO-MECANICAS

**ESCUELA DE INGENIERÍAS ELÉCTRICA, ELECTRÓNICA Y
TELECOMUNICACIONES**

BUCARAMANGA

2013

**ALGORITMO DE ENJAMBRE DE PARTÍCULAS UNIFICADO PARA LA
SOLUCIÓN DE ECUACIONES DIOFÁNTICAS LINEALES COMÚNMENTE
ENCONTRADAS EN PROBLEMAS DE INGENIERÍA ELECTRÓNICA**

JONATHAN JAVIER RAMÍREZ ZAMBRANO

FAUSTO OSORIO SILVA

Trabajo de grado para optar por el título de INGENIERO ELECTRONICO

RODRIGO CORREA, Ph.D.

DIRECTOR Y PONENTE DEL TRABAJO DE GRADO

Ing. IVÁN AMAYA, Ph.D. (c)

CODIRECTOR DEL TRABAJO DE GRADO

UNIVERSIDAD INDUSTRIAL DE SANTANDER

FACULTAD DE INGENIERIAS FISICO-MECANICAS

**ESCUELA DE INGENIERÍAS ELÉCTRICA, ELECTRÓNICA Y
TELECOMUNICACIONES**

BUCARAMANGA

2013

TABLA DE CONTENIDO

1	INTRODUCCIÓN	18
2	DESCRIPCIÓN DEL TRABAJO DE GRADO.....	20
2.1	OBJETIVO GENERAL.....	20
2.2	OBJETIVOS ESPECÍFICOS.....	20
3	FORMULACIÓN DEL PROBLEMA.....	21
4	FUNDAMENTOS TEÓRICOS.....	22
4.1	ECUACIONES DIOFÁNTICAS	23
4.2	FUNCIÓN OBJETIVO.....	24
4.3	OPTIMIZACIÓN POR ENJAMBRE DE PARTÍCULAS (PSO)	25
4.4	OPTIMIZACIÓN POR ENJAMBRE DE PARTÍCULAS UNIFICADO (UPS0).....	26
4.5	OPTIMIZACIÓN POR ENJAMBRE DE PARTÍCULAS COMBINATORIO (CPSO).....	27
4.6	ECUACIONES DIOFÁNTICAS EN SISTEMAS DE CONTROL.....	28
5	IMPLEMENTACION DEL ALGORITMO	30
5.1	EJERCICIO DE OPTIMIZACIÓN: EL CHEQUE DESCONOCIDO	32
5.2	EJERCICIO DE OPTIMIZACIÓN: NUMEROS CONSECUTIVOS	35
5.3	EJERCICIO DE OPTIMIZACIÓN: TRIPLA PITAGORICA	37
5.4	EJERCICIO DE OPTIMIZACIÓN: COMPRA DE CELULARES	41
5.5	EJERCICIO DE OPTIMIZACIÓN: PROCESO DE FABRICACIÓN.....	44
5.6	EJERCICIO DE OPTIMIZACIÓN: GANANCIAS DE UN RESTAURANTE 47	
5.7	EJERCICIO DE OPTMIZACIÓN: TRABAJO DE UN SASTRE	50

5.8	EJERCICIO DE OPTIMIZACIÓN: EL GRANJERO Y SUS ANIMALES	52
5.9	EJERCICIO OPTIMIZACIÓN: COMPRA DE INTERIORES	55
5.10	EJERCICIO DE OPTIMIZACIÓN: DISEÑO DE MECANISMOS.....	57
6	APLICACIÓN DEL ALGORITMO EN PROBLEMAS PRESENTADOS EN INGENIERÍA ELECTRÓNICA.....	60
6.1	SISTEMA DE CONTROL DE LAZO CERRADO CON REALIMENTACION UNITARIA.....	61
6.2	SISTEMA DE CONTROL DISCRETO MEDIANTE LA IGUALACIÓN A UN MODELO	68
6.3	ENFOQUE DE ECUACIONES POLINOMIALES PARA EL DISEÑO DEL SISTEMA DE REGULACIÓN Y CUYA SALIDA SIGA LA ENTRADA DE REFERENCIA	77
7	PRUEBAS EN SISTEMAS DE ORDEN SUPERIOR	85
7.1	ALGORITMO DE ENJAMBRE DE PARTÍCULAS UNIFICADO (UPSO) ..	87
7.2	ALGORTIMO DE ENJAMBRE DE PARTÍCULAS (PSO).....	89
7.3	CASO UNO: UBICACIÓN DE 6 POLOS EN -1	90
7.4	SEGUNDO CASO: UBICACIÓN DE 10 POLOS EN -1	94
7.5	TERCER CASO: UBICACIÓN DE 25 POLOS EN -1.....	101
7.6	FUNCIÓN EN MATLAB DE OPTIMIZACION FZERO.	106
8	CONCLUSIONES	107
9	RECOMENDACIONES	109
10	BIBLIOGRAFIA	110
11	ANEXOS.....	113

LISTADO DE TABLAS

Tabla 1. Trabajos de grado 2011-2012.....	17
Tabla 2. Especificaciones del hardware usado para las pruebas.	32
Tabla 3. Resultados con el algoritmo DUPSO para el ejercicio del cheque.....	33
Tabla 4. Resumen de datos obtenidos para el ejercicio del cheque.....	34
Tabla 5. Resultados con el algoritmo DUPSO para el ejemplo de los números consecutivos	36
Tabla 6. Resumen resultados encontrados para el ejercicio de los números consecutivos	36
Tabla 7. Resultados encontrados con el algoritmo para para el ejercicio de la tripla pitagórica.	39
Tabla 8. Comparación de resultados para el ejercicio de la tripla pitagórica.	40
Tabla 9. Resultados con el algoritmo para el ejercicio de la compra de celulares.	42
Tabla 10. Comparación de datos del ejercicio de la compra de celulares	44
Tabla 11. Resultados de la optimización con DUPSO del problema de la fábrica.	46
Tabla 12. Comparación de los resultados de la optimización de la fábrica.....	47
Tabla 13. Resultados de la optimización con DUPSO para el problema del restaurante.....	48
Tabla 14. Comparación con CPSO de los resultados del ejercicio del restaurante	49
Tabla 15. Resultados con DUPSO para el problema del sastre	51
Tabla 16. Comparación de resultados para el ejercicio del sastre.....	52
Tabla 17. Resultados con DUPSO del problema del granjero	53
Tabla 18. Comparación de datos del problema del granjero	54
Tabla 19. Resultados del problema de los interiores con DUPSO.....	55
Tabla 20. Comparación de resultados para el ejercicio de los interiores.	56
Tabla 21. Resultados del problema de diseño de engranajes con DUPSO.....	57
Tabla 22. Comparación de resultados para el ejercicio del engranaje.....	58

Tabla 23. Resultados con el algoritmo DUPSO para diseño del controlador.....	65
Tabla 24. Resumen de datos para el diseño del controlador.....	66
Tabla 25. Resultados de forma fraccionaria hallados con DUPSO.....	73
Tabla 26. Parámetros del controlador encontrados con DUPSO.....	74
Tabla 27. Resultados de forma fraccionaria encontrados con DUPSO	80
Tabla 28. Parámetros para el controlador encontrados DUPSO	81
Tabla 29. Resultados completos del ejercicio 5.1 con otros métodos de optimización.....	113
Tabla 30. Resultados completos del ejercicio 5.2 con otros métodos de optimización.....	114
Tabla 31. Resultados con CPSO para el ejercicio de la tripla pitagórica	115
Tabla 32. Resultados con el algoritmo CPSO para la compra de celulares.....	116
Tabla 33. Resultados con el algoritmo CPSO para el problema del proceso de fabricación	117
Tabla 34. Resultados con CPSO para el ejercicio del restaurante.	118
Tabla 35. Resultados con CPSO para el ejercicio del sastre.....	119
Tabla 36. Resultados del ejercicio del granjero implementado con el algoritmo CPSO.....	120
Tabla 37. Resultados para el ejercicio de los interiores con CPSO.....	121
Tabla 38. Resultados para el ejercicio del mecanismo con CPSO	122
Tabla 39. Datos hallados con el algoritmo CPSO.....	123
Tabla 40. Datos del ejercicio implementado con PSO.....	124
Tabla 41. Datos del ejercicio implementado con UPSO	125

LISTADO DE FIGURAS

Figura 1. Diagrama de bloques del sistema con realimentación unitaria.	61
Figura 2. Respuesta al escalón del sistema de control.	67
Figura 3. Diagrama de Bloques del sistema de control mediante la igualación de un modelo. Basado en [17]	69
Figura 4. Diagrama de bloques del sistema de control mediante la igualación de un modelo. Basado en [17]	75
Figura 5. Resultados de la simulación del sistema.	76
Figura 6. Esquema del ejercicio.	77
Figura 7. Diagrama en Simulink correspondiente al ejercicio.	83
Figura 8. Respuesta al escalón unitario.	83
Figura 9. Respuesta al escalón de amplitud igual a 3.	84
Figura 10. Sistema de control con realimentación unitaria.	86
Figura 11. Función objetivo en el Intervalo entre 0 y 100 iteraciones con UPSO. .	91
Figura 12. Función objetivo en el Intervalo final con UPSO.	91
Figura 13. Valor de la función objetivo entre 0 y 100 iteraciones con PSO.	92
Figura 14. Valor de la función objetivo en el intervalo entre 1000 y 2000 iteraciones con PSO	93
Figura 15. Valor de la función objetivo en el intervalo final de iteraciones con PSO.	93
Figura 16. Valor de la función objetivo en el intervalo entre 0 y 100 iteraciones con UPSO.	94
Figura 17. Valor de la función objetivo en el intervalo entre 1000 y 2000 iteraciones con UPSO.	95
Figura 18. Valor de la función objetivo en el intervalo entre 10000 y 20000 iteraciones con UPSO.	95
Figura 19. Valor de la función objetivo en el intervalo entre 40000 y 70000 iteraciones con UPSO.	96

Figura 20. Valor de la función objetivo en el intervalo entre 70000 y 100000 iteraciones con UPSO.....	96
Figura 21. Valor de la función objetivo en el intervalo entre 0 y 100iteraciones con PSO	97
Figura 22. Valor de la función objetivo en el intervalo entre 1000 y 2000 iteraciones con PSO	98
Figura 23. Valor de la función objetivo en el intervalo entre 10000 y 20000 iteraciones con PSO	98
Figura 24. Valor de la función objetivo en el intervalo entre 40000 y 70000 iteraciones con PSO	99
Figura 25. Valor de la función objetivo en el intervalo entre 70000 y 100000 iteraciones con PSO	99
Figura 26. Valor de la función objetivo en el intervalo entre 100000 y 500000 iteraciones con UPSO.....	100
Figura 27. Valor de la función objetivo en el intervalo entre 500000 y 900000 iteraciones con PSO	100
Figura 28. Valor de la función objetivo en el intervalo entre 0 y 100 iteraciones con UPSO.....	101
Figura 29. Valor de la función objetivo en el intervalo entre 1000 y 10000 iteraciones con UPSO.....	102
Figura 30. Valor de la función objetivo en el intervalo entre 40000 y 100000 iteraciones con UPSO.....	102
Figura 31. Valor de la función objetivo en el intervalo entre 0 y 100 iteraciones con PSO	103
Figura 32. Valor de la función objetivo en el intervalo entre 1000 y 10000 iteraciones con PSO	104
Figura 33. Valor de la función objetivo en el intervalo entre 20000 y 50000 iteraciones con PSO	104
Figura 34. Valor de la función objetivo en el intervalo entre 100000 y 190000 iteraciones con PSO	105

Figura 35. Valor de la función objetivo en el intervalo entre 190000 y 200000 iteraciones con PSO105

LISTADO DE ANEXOS

ANEXO A: TABLAS DE RESULTADOS CON OTROS METODOS PARA EL EJERCICIO DE LA COMPRA DE CELULARES.....	113
ANEXO B: TABLAS DE RESULTADOS PARA EL EJERCICIO DE LOS NUMEROS CONSECUTIVOS	114
ANEXO C: TABLAS DE RESULTADOS DE CPSO PARA EL EJERCICIO DE LA TRIPLA PITAGÓRICA	115
ANEXO D: TABLAS DE RESULTADOS DE CPSO PARA EL EJERCICIO DE LA COMPRA DE CELULARES.....	116
ANEXO E: RESULTADOS CON EL ALGORITMO CPSO PARA EL PROBLEMA DEL PROCESO DE FABRICACIÓN.....	117
ANEXO F: TABLA DE RESULTADOS CON EL ALGORITMO DE CPSO PARA EL EJERCICIO DEL RESTAURANTE.	118
ANEXO G: TABLA DE RESULTADOS CON CPSO PARA EL EJERCICIO DEL SASTRE	119
ANEXO H: RESULTADOS CON CPSO PARA EL EJERCICIO DEL GRANJERO Y SUS ANIMALES	120
ANEXO I: TABLA DE RESULTADOS CON CPSO PARA EL EJERCICIO DE LOS INTERIORES	121
ANEXO J: RESULTADOS ENCONTRADOS CON CPSO PARA EL EJERCICIO DEL MECANISMO.....	122
ANEXO K: TABLAS DE RESULTADOS CON OTROS METODOS PARA LA SECCION 6.1.....	123

RESUMEN

TÍTULO: ALGORITMO DE ENJAMBRE DE PARTÍCULAS UNIFICADO PARA LA SOLUCIÓN DE ECUACIONES DIOFÁNTICAS LINEALES COMÚNMENTE ENCONTRADAS EN PROBLEMAS DE INGENIERÍA ELECTRÓNICA¹

AUTORES: JONATHAN JAVIER RAMÍREZ ZAMBRANO; FAUSTO OSORIO SILVA²

PALABRAS CLAVE: UPSO, DUPSO, función objetivo, ecuaciones diofánticas, optimización, factor de unificación.

CONTENIDO: Este documento tiene como objetivo mostrar una aplicación para el método de optimización por enjambre de partículas unificado (UPSO), y específicamente una variante discreta del método UPSO (DUPSO), para la solución de problemas encontrados en la ingeniería electrónica, caracterizados con ecuaciones diofánticas. Utilizando el Teorema de las Raíces Reales, se convierte el problema de solución de un sistema de ecuaciones, en uno de optimización. Con la finalidad de verificar la validez del algoritmo se muestran diez ejemplos diferentes, solucionados por el método de optimización PSO (incluyendo algunas de sus más conocidas variantes), así como de forma analítica. Además de esto se verifica la validez del algoritmo para el diseño de controladores, a través de los diez ejemplos en los que DUPSO encontró resultados apropiados en términos de tiempo de cómputo y precisión. Se resolvieron unos problemas prácticos en sistemas de control donde el enfoque estuvo centrado en determinar los coeficientes de la función de transferencia de un controlador. Al final el método de PSO se puso a prueba incrementando el orden del sistema, para observar la convergencia que tiene el mismo para ser implementado a sistemas de orden superior como alternativa de solución de los métodos convencionales mostrados en la literatura.

¹ Proyecto de grado.

² Facultad de Ingenierías físico-mecánicas. Escuela de Ingenierías Eléctrica, Electrónica y de Telecomunicaciones. Director: Carlos Rodrigo Correa. Codirector: Iván Mauricio Amaya.

ABSTRACT

TITLE: UNIFIED PARTICLE SWARM OPTIMIZATION ALGORITHM FOR THE SOLUTION OF DIOPHANTINE LINEAR EQUATIONS COMMONLY FOUND IN PROBLEMS OF ELECTRONIC ENGINEERING³

AUTHORS: JONATHAN JAVIER RAMIREZ ZAMBRANO; FAUSTO OSORIO SILVA⁴

KEY WORDS: Unified particle swarm optimization (UPSO), DUPSO, objective function, Diophantine equations, optimization, unified factor.

ABSTRACT: This document has as aim show an application for the method of optimization for swarm of particles unified (UPSO), and specifically a discreet variant of the method UPSO (DUPSO), for the solution of problems found in the electronic engineering, characterized with Diophantine equations. Using the Theorem of the Royal Roots, there turns the problem of solution of a system of equations, in one of optimization. With the purpose of checking the validity of the algorithm there appear ten different examples solved by particle swarm optimization (including some of his more known variants), as well as of analytical form. Besides this the validity of the algorithm happens for the design of controllers, across ten examples in which DUPSO was proved adapted in terms of time of calculation and precision. Some practical problems in control systems were resolved where the approach was determining the coefficients of the transfer function of a controller. Finally, PSO method was tested by increasing the order of the system to observe the convergence of the algorithm to be implemented in a higher system order as an alternative of solution to the conventional methods shown in the literature.

³ Undergraduate thesis.

⁴ Faculty of Physical Mechanical Engineering. School of Electrical, Electronic and Telecommunications. Advisor: Carlos Rodrigo Correa. Co-advisor: Iván Mauricio Amaya.

Grupo CEMOS

Línea de trabajo: *Modelado y Optimización*

La línea de modelado y optimización del grupo CEMOS se encuentra orientada hacia la evaluación de los más importantes algoritmos de optimización y fundamentalmente los basados en la inteligencia artificial y la heurística. Los siguientes son los trabajos de grado que a la fecha tienen como objetivo fortalecer esta línea, dándosele un énfasis a su aplicación en la ingeniería electrónica y eléctrica.

Tabla 1. Trabajos de grado 2011-2012

SOLUCIÓN DEL MODELO MATEMÁTICO DE UN CIRCUITO ELECTRÓNICO COMPUESTO DE ELEMENTOS NO LINEALES MEDIANTE EL OPTIMIZADOR DE FUERZA CENTRAL	ROA, O; RAMIREZ, F
EVALUACIÓN DEL MÉTODO OPTIMIZADOR DE FUERZA CENTRAL FRENTE AL OPTIMIZADOR POR ENJAMBRE DE PARTÍCULAS UNIFICADO EN LA SOLUCIÓN DE ECUACIONES NO LINEALES	BARRERO, K; NAVARRO, O
ANÁLISIS DEL COMPORTAMIENTO DINÁMICO DE UN CIRCUITO ELECTRÓNICO NO LINEAL MEDIANTE LA SOLUCIÓN DE SU MODELO CON HARMONY SEARCH	TRASLADINO, O; GONZALES, S
APLICACIÓN DEL MÉTODO DE OPTIMIZACIÓN POR FUERZA CENTRAL (CFO) AL DISEÑO DE UN ABSORBEDOR ELECTROMAGNÉTICO ÓPTIMO	BAYONA, C; GONZALEZ, J
ALGORITMO HÍBRIDO DEL SIMPLEX CON EL OPTIMIZADOR DE FUERZA CENTRAL (CFO)	GARCÍA, J
PROPUESTA DE ESTRATEGIA PARA LA OBTENCIÓN DE LOS PARÁMETROS DEL ALGORITMO DE OPTIMIZACIÓN POR FUERZA CENTRAL (CFO)	BUITRAGO, J
SOLUCIÓN DE UN SISTEMA DE ECUACIONES NO LINEAL UTILIZANDO LA ESTRATEGIA BASADA EN EL ALGORITMO HARMONY SEARCH	GALEANO, Y
IMPLEMENTACIÓN DEL ALGORITMO DE BÚSQUEDA GRAVITACIONAL (GSA) PARA EL DISEÑO DE UN ABSORBEDOR ELECTROMAGNÉTICO ÓPTIMO	MIRANDA, A; RUIZ, J
SOLUCIÓN DEL MODELO MATEMÁTICO DE UN CIRCUITO NO LINEAL EN DC, A TRAVÉS DE UNA ESTRATEGIA DE OPTIMIZACIÓN	CRUZ, J
USO DEL HÍBRIDO UPSO CON EL SIMPLEX PARA OPTIMIZACIÓN DE FUNCIONES CON RESTRICCIONES EN PROBLEMAS DE INGENIERÍA ELECTRÓNICA	CAÑÓN, E
INTELIGENCIA COMPUTACIONAL INSPIRADA EN LA CUÁNTICA, APLICADA AL DISEÑO DE COMPENSADORES	HERNÁNDEZ, M; CORREDOR, D
OPTIMIZACIÓN PARETO MEDIANTE EL MÉTODO DE ENJAMBRE DE PARTÍCULAS UNIFICADO EN PROBLEMAS DE INGENIERÍA ELECTRÓNICA	QUIROZ, M;
IDENTIFICACIÓN DE PARÁMETROS DE UN MOTOR SÍNCRONO DE IMÁN PERMANENTE MEDIANTE EL ALGORITMO DE LA ESPIRAL	RUTTO, A
EL MÉTODO DE COLONIA ARTIFICIAL DE ABEJAS Y EL CRITERIO DE MÍNIMA ENTROPÍA PARA EL DISEÑO ÓPTIMO DE UN DISIPADOR DE CALOR	ÁVILA, J
IDENTIFICACIÓN DE SISTEMAS NO LINEALES MEDIANTE EL MÉTODO DE OPTIMIZACIÓN DE GOTA DE AGUA INTELIGENTE	CARVAJAL, L; DÁVILA, D
POTENCIAL APLICACIÓN DEL ALGORITMO DE GOTAS DE AGUA INTELIGENTES (GAI-IWD) EN INGENIERÍA ELECTRÓNICA	ARIAS, J
CONTRASTACIÓN DE LOS MÉTODOS DE ENJAMBRE DE PARTÍCULAS UNIFICADO FRENTE AL MÉTODO COLONIA ARTIFICIAL DE ABEJAS, EN LA IDENTIFICACIÓN DE SISTEMAS	PETRO, E; FUENTES, R.
EVALUACIÓN Y COMPARACIÓN ENTRE EL MÉTODO NEWTON RAPHSON Y CENTRAL FORCE OPTIMIZATION PARA EL ANÁLISIS DE FLUJO DE CARGAS DE UN SISTEMA DE POTENCIA	VALDERRAMA, M
EVALUACIÓN Y COMPARACIÓN ENTRE LOS MÉTODOS NEWTON RAPHSON Y ARTIFICIAL BEE COLONY (ABC) PARA EL ANÁLISIS DEL FLUJO DE CARGA DE UN SISTEMA DE POTENCIA	CELIS, J; RINCÓN, F
EVALUACIÓN Y COMPARACIÓN ENTRE LOS MÉTODOS UPSO Y NEWTON-RAPHSON PARA EL ANÁLISIS DE FLUJO DE CARGA EN UN SISTEMA DE POTENCIA	PINZÓN, C; ARDILA, E
DISEÑO DE UN ABSORBEDOR ELECTROMAGNÉTICO MULTICAPA MEDIANTE EL MÉTODO DE LA ESPIRAL	ORTIZ, R; GARCÍA, E
SOLUCIÓN DE LAS ECUACIONES QUE MODELAN UN CIRCUITO NO LINEAL DE CORRIENTE DIRECTA MEDIANTE EL MÉTODO DE ESPIRAL	PORTILLA, J
ALGORITMO DE ENJAMBRE DE PARTÍCULAS PARA SOLUCIONAR ECUACIONES DIOFÁNTICAS LINEALES COMUNMENTE ENCONTRADAS EN PROBLEMAS DE INGENIERÍA ELECTRÓNICA	RAMIREZ, J; OSORIO, F

1 INTRODUCCIÓN

Muchos de los problemas que surgen en ingeniería se pueden representar mediante un modelo matemático, siendo indispensable buscar una solución rápida y confiable. Durante esta investigación se utilizaron algoritmos metaheurísticos, los cuales buscan una solución siguiendo un modelo evolutivo basado en la naturaleza. Estos métodos basados en la heurística combinan la exploración y la explotación del espacio de búsqueda con el fin de encontrar una solución óptima del problema.

PSO (Particle Swarm Optimization) es uno de estos procesos y su funcionamiento está caracterizado por la búsqueda de un valor óptimo local y uno global en cada iteración, comunicándolos al enjambre y trabajando en equipo para encontrar el óptimo global de la función. La fortaleza de PSO está en poder resolver sistemas tanto continuos como de tipo discreto, y al igual que otros métodos de optimización, tiene algunos puntos débiles que no se detallarán en esta introducción (principalmente por limitaciones de espacio)[1].

Este informe presenta los resultados obtenidos al aplicar el método propuesto DUPSO (Discrete Unified Particle Swarm Optimization), una variante del método UPSO (Unified Particle Swarm Optimization) la cual es una muy conocida variante de PSO, en un problema específico encontrado en ingeniería electrónica. UPSO basa su funcionamiento en agregar una componente de unificación entre la exploración y la explotación del enjambre, mejorando en muchos casos la tasa de convergencia del método [2]. En el siguiente capítulo se encuentra la descripción de este trabajo de grado así como el objetivo general y los específicos. En el tercero se encuentra contenida la base teórica en la que se fundamenta este trabajo de grado en la modalidad de investigación. La aplicación del algoritmo se explica en el cuarto capítulo. La validación del algoritmo con problemas de

matemáticas con solución conocida se encuentra en el quinto capítulo. La aplicación en problemas relacionados con la teoría de control y el diseño de controladores se encuentra recopilada en el sexto capítulo. Finalmente las conclusiones y recomendaciones se encuentran en los capítulos séptimo y octavo respectivamente.

2 DESCRIPCIÓN DEL TRABAJO DE GRADO

En este informe de investigación se recopila el proceso seguido y los principales resultados encontrados en el desarrollo del trabajo de grado: “Algoritmo de enjambre de partículas unificado para la solución de ecuaciones diofánticas lineales comúnmente encontradas en problemas de ingeniería electrónica”. De igual manera se evidencia el cumplimiento de todos los objetivos propuestos, los cuales se enuncian a continuación.

2.1 OBJETIVO GENERAL

Implementar una estrategia de optimización para resolver sistemas de ecuaciones diofánticas comúnmente encontradas en el área de sistemas de control y/o criptografía (Sistemas de Seguridad Electrónica)

2.2 OBJETIVOS ESPECÍFICOS

Para el cumplimiento del objetivo general se requiere cumplir con lo siguiente:

- Implementar un algoritmo discreto, fundamentado en UPSO (Algoritmo de enjambre de partículas unificado) en MATLAB (capítulo 5).
- Seleccionar un problema de ingeniería electrónica (Sistemas de control o criptografía) modelado mediante ecuaciones diofánticas (capítulo 6).
- Resolver el problema seleccionado mediante la estrategia propuesta y compararlo con algunos resultados reportados en la literatura (capítulo 6).

3 FORMULACIÓN DEL PROBLEMA

En múltiples ramas de la ingeniería electrónica se presentan problemas al intentar resolver sistemas de ecuaciones algebraicas lineales, debido a su complejidad. En ciertas áreas específicas, ésta complejidad se incrementa ya que aunque el sistema puede tener múltiples soluciones, solo será de interés la o las soluciones de valor entero.

Las ecuaciones diofánticas pueden surgir en el modelamiento de un sistema y se hace indispensable darle solución. Ésta requiere un esfuerzo matemático que aumenta con el orden de la ecuación, y se puede llegar a tener un gran número de incógnitas en una sola expresión.

En esta investigación se aborda el caso particular de control polinomial, y un ejemplo de cómo resolverlo analíticamente se presenta en [3]. Este tipo de control ofrece diferentes modelos para realizar la acción control, y en la mayoría de ellos, su enfoque está puesto en resolver una ecuación diofántica que contiene los coeficientes de los polinomios característicos de la función de transferencia del controlador. Si la acción de control requiere un esfuerzo considerable, la complejidad para hallar estos coeficientes se incrementa y es necesario buscar formas alternativas para solucionar la ecuación [3].

4 FUNDAMENTOS TEÓRICOS

En esta sección se abordan los conceptos teóricos necesarios para abordar de forma adecuada el problema, así como la interpretación de los métodos utilizados. Inicialmente se muestran los fundamentos de los sistemas de ecuaciones diofánticas. Posteriormente, se enseña la forma de convertir un problema de solución de sistemas de ecuaciones algebraicas en uno de optimización. Finalmente se presentan los métodos de optimización implementados en este trabajo (CPSO y DUPSO).

El problema a solucionar puede ser descrito por un sistema de m ecuaciones con n incógnitas que describen el problema (1). Para este trabajo en concreto, hace referencia al modelado matemático de un control en lazo cerrado, donde las variables x_1, x_2, \dots, x_n definirán cada uno de los coeficientes de la función de transferencia del controlador.

$$\begin{aligned} f_1(x_1, x_2, \dots, x_n) &= 0 \\ f_2(x_1, x_2, \dots, x_n) &= 0 \\ f_3(x_1, x_2, \dots, x_n) &= 0 \\ &\vdots \\ f_m(x_1, x_2, \dots, x_n) &= 0 \end{aligned} \tag{1}$$

4.1 ECUACIONES DIOFÁNTICAS

Los sistemas de ecuaciones diofánticas son aquellos cuyas raíces se encuentran ubicadas únicamente en el conjunto de los números enteros. Deben su nombre al matemático griego Diofanto de Alejandría, quien en el siglo III DC, fue de los pioneros en escribir notación simbólica con matemáticas. Una ecuación diofántica lineal de n incógnitas se denota de acuerdo a la siguiente ecuación:

$$a_1x_1 + a_2x_2 + a_3x_3 + \dots + a_nx_n = b \quad (2)$$

En (2) $a_1, a_2, a_3, \dots, a_n$ son números enteros conocidos y $x_1, x_2, x_3, \dots, x_n$ son las incógnitas de valor entero a encontrar[4],[5].

$$Ax_1 + Bx_2 = C \quad (3)$$

En forma general, una ecuación diofántica con dos incógnitas, (3), aunque tenga múltiples soluciones, la única válida existe si el máximo común divisor de A y B es divisor también de C . Esta solución se caracteriza de la siguiente forma:

$$x_1 = x_0 + \beta * \frac{B}{d} \quad (4)$$

$$x_2 = x'_0 - \beta * \frac{A}{d} \quad (5)$$

Donde β es un número entero cualquiera, además x_0 y x'_0 son soluciones conocidas de la ecuación [4], [6].

4.2 FUNCIÓN OBJETIVO

La función objetivo se fundamenta en la necesidad de convertir un problema de solución de un sistema de ecuaciones algebraicas, en uno de optimización. Esta transformación se realiza con el teorema de las raíces reales el cual se describe a continuación:

Sea X un subconjunto de R^n y el sistema (1), donde para cada i, f_i es una función en la cual el dominio contiene a X , y en donde el rango está dentro del conjunto de los números reales. Entonces $f_{obj}(x) \rightarrow R^n$ se encuentra definida por (6). Es necesario destacar que f_i debe ser definida de forma apropiada

$$f_{obj}(x) = \sum_{i=1}^m (f_i(x))^2; \text{ donde } x = (x_1, x_2, \dots, x_n) \quad (6)$$

Además:

Preposición 1: Se supone que (1) tiene solución en x y donde $a = (a_1, a_2, \dots, a_n) \in x$ mínimo para f .

Demostración. Si a minimiza a f_{obj} pero no cumple (1) para cada $i = 1, 2, \dots, n$. Por lo tanto $f_{obj(a)} = f_{obj}(x) \geq 0$ para todo $x \in X$. Si es conocido un valor x donde el sistema tiene solución entonces existirá un $x' \in X$ que cumple con $f_{obj}(x') = 0$, siendo $x' \neq a$. Consecuentemente con lo anterior $f_{obj}(x') \leq f_{obj}(a)$ y esto viola la preposición 1, ya que a minimizaría a f . La condición general de consistencia del sistema es muy importante, ya que siempre es posible construir f para un sistema

dado, y si a minimiza dicha función f , esto no implica que exista una solución. Por esto, la búsqueda de raíces para un sistema de ecuaciones sobre un conjunto x dado, puede ser replanteada como un problema de optimización (minimización para este caso) de la función f_{obj} en el dominio de x [7],[8].

4.3 OPTIMIZACIÓN POR ENJAMBRE DE PARTÍCULAS (PSO)

PSO fue propuesto por Kennedy y Eberhart en 1995 y se encuentra inspirado en el comportamiento colectivo animal [1]. Cada solución es una partícula en el espacio. Su funcionamiento consiste en desplazar las partículas por el espacio en busca de la solución con velocidades adaptables, que permiten la transición de una posición a otra. En su forma general, el algoritmo PSO se rige por dos ecuaciones básicas, las cuales son posición y velocidad [9]. La ecuación (7) hace referencia a la posición de la partícula i mientras que la ecuación (8) se refiere a su velocidad. Estas ecuaciones se actualizan en cada iteración y con la evaluación de la partícula en la función objetivo, se actualiza el global y el local [10].

$$X_i^{j+1} = X_i^j + V_i^{j+1} \quad (7)$$

$$V_i^{j+1} = wV_i^j + c_1rand_1(P_i - X_i^j) + c_2rand_2(P_g - X_i^j) \quad (8)$$

La inercia w limita el efecto de la velocidad, haciendo que no tome valores perjudiciales para la búsqueda, y normalmente se encuentra entre 0.4 y 0.9. Los valores $rand_1$ y $rand_2$ son aleatorios, uniformemente distribuidos entre 0 y 1. P_i representa la mejor posición que ha tenido la partícula i (mejor posición local) y P_g es la mejor posición que ha visitado todo el enjambre (mejor posición global)[11], [12].

4.4 OPTIMIZACIÓN POR ENJAMBRE DE PARTÍCULAS UNIFICADO (UPSO)

UPSO fue desarrollado por Parsopoulos y Vrahatis en 2004 [2], [13]. Este esquema fue propuesto como alternativa para combinar las propiedades de exploración y explotación de las variables local y global en PSO. El esquema modifica las ecuaciones originales en dos formas. La primera consta de dos nuevas ecuaciones que definen la velocidad global y local de la siguiente manera:

$$G_i^{j+1} = \chi[V_i^j + c_1rand_1(P_i - X_i^j) + c_2rand_2(P_g - X_i^j)] \quad (9)$$

$$L_i^{j+1} = \chi[V_i^j + c_1rand_3(P_i - X_i^j) + c_2rand_4(P_{gi} - X_i^j)] \quad (10)$$

Donde P_g es la partícula mejor ubicada del enjambre, P_{gi} es la mejor posición del enjambre actual, P_i es la mejor posición que la partícula i ha visitado, χ es el factor de restricción, cuya misión es evitar que el enjambre diverja al limitar su velocidad (su valor óptimo es de 0,72)[12].

La segunda modificación es la combinación de estas dos velocidades mediante un parámetro de unificación (u), el cual puede tomar valores entre 0 y 1. Su valor óptimo se establece en 0.5 para tener un aporte equitativo de las velocidades global y local [14][15].

$$U_i^{j+1} = (1 - u)L_i^{j+1} + uG_i^{j+1} \quad (11)$$

Esta nueva velocidad combina las propiedades de exploración y explotación. La forma en que se carga la nueva partícula es igual que en PSO.

$$X_i^{j+1} = X_i^j + U_i^{j+1} \quad (12)$$

4.5 OPTIMIZACIÓN POR ENJAMBRE DE PARTÍCULAS COMBINATORIO (CPSO)

Esta segunda modificación de PSO se enfoca en variables discretas, donde solo los números enteros son solución. Al igual que en el algoritmo original, existen dos vectores X_i y V_i correspondientes a la posición y velocidad de cada partícula.

Sea $Y_i^t = \{y_{i1}^t, y_{i2}^t, \dots, y_{in}^t\}$ un vector n-dimensional asociado a la solución $X_i^t = \{X_{i1}^t, X_{i2}^t, \dots, X_{in}^t\}$ el cual toma un valor de -1, 0 o 1 de acuerdo con el estado de la solución de la partícula i en la iteración t . Y^t es una variable que permite la transición de PSO a CPSO y viceversa [16].

$$y_{ij}^t = \left\{ \begin{array}{ll} 1, & \text{si } X_{ij}^t = G_j^t \\ -1, & \text{si } X_{ij}^t = p_{ij}^t \\ -1 \text{ o } 1, & \text{aleatoriamente si } X_{ij}^t = G_{ij}^t = p_{ij}^t \\ 0, & \text{otro valor} \end{array} \right\} \quad (13)$$

Teniendo en cuenta (13), la nueva ecuación de la velocidad en términos de CPSO es:

$$V_{ij}^t = wV_{ij}^{t-1} + rand_1 c_1 (1 - y_{ij}^{t-1}) + rand_2 c_2 (1 - y_{ij}^{t-1}) \quad (14)$$

La búsqueda se realiza según (15) :

$$\lambda_{ij}^t = y_{ij}^{t-1} + V_{ij}^t \quad (15)$$

El valor de y_{ij}^t es ajustado nuevamente de la siguiente forma:

$$y_{ij}^t = \begin{cases} 1, & \text{si } \lambda_{ij}^t > \alpha \\ -1, & \text{si } \lambda < -\alpha \\ 0, & \text{otro valor} \end{cases} \quad (16)$$

La constante α de (16) es una constante del problema que define la intensificación, y la nueva solución es:

$$X_{ij}^t = \begin{cases} G_j^{t-1}, & \text{si } y_{ij}^t = 1 \\ p_{ij}^{t-1}, & \text{si } y_{ij}^t = -1 \\ \text{Número aleatorio, otro valor} \end{cases} \quad (17)$$

4.6 ECUACIONES DIOFÁNTICAS EN SISTEMAS DE CONTROL

Considere la siguiente función de transferencia:

$$\frac{Y(s)}{X(s)} = \frac{B(s)}{A(s)} \quad (18)$$

Donde

$$A(s) = s^n + a_1 s^{n-1} + \dots + a_{n-1} s + a_n \quad (19)$$

$$B(s) = b_0 s^n + b_1 s^{n-1} + \dots + b_{n-1} s + b_n \quad (20)$$

Suponga que la función de transferencia es de estado completamente controlable y completamente observable. Esto es, no existe una cancelación entre polos y ceros en la función de transferencia. Es decir $A(s)$ y $B(s)$ son polinomios coprimos (tomado de [17], este modelo es válido tanto en el dominio de Laplace como de Z).

A continuación se define un polinomio estable $D(s)$ de grado $2n - 1$

$$D(s) = d_0 s^{2n-1} + d_1 s^{2n-2} + \dots + d_{2n-2} s + d_{2n-1} \quad (21)$$

Entonces de (21) existen polinomios únicos de grado $n - 1$, $\alpha(s)$ y $\beta(s)$ tales que:

$$\alpha(s)A(s) + \beta(s)B(s) = D(s) \quad (22)$$

Donde

$$\alpha(s) = \alpha_0 s^{n-1} + \alpha_1 s^{n-2} + \dots + \alpha_{n-2} s + \alpha_{n-1} \quad (23)$$

$$\beta(s) = \beta_0 s^{n-1} + \beta_1 s^{n-2} + \dots + \beta_{n-2} s + \beta_{n-1} \quad (24)$$

5 IMPLEMENTACION DEL ALGORITMO

En este capítulo se describe el cumplimiento del siguiente objetivo específico:

- Implementar un algoritmo discreto, fundamentado en algoritmo de enjambre de partículas unificado (UPSO).

La estrategia a seguir consiste en utilizar velocidades continuas y posiciones discretas [18], [19], [20] y la manera de ejecutarlo se describe a continuación. Otra forma de discretizar la partícula se realiza mediante combinatoria, este es el otro método más popular para realizar PSO discreto. Se implementará un algoritmo con velocidades continuas y posiciones de atracción discretas y sus resultados serán contrastados con PSO combinatorio (CPSO)[1]. A continuación se presenta el algoritmo discreto DUPSO propuesto en este trabajo de grado.

Los pasos del algoritmo se describen a continuación:

1. Definir las constantes del esquema de UPSO: $c_1, c_2, c_3, c_4, \chi, u$.
2. Definir los valores iniciales para la posición de las partículas y sus velocidades, los cuales serán aleatorios para la posición X_i^0 , y se partirá con velocidad inicial cero $V_i^0 = 0$.
3. Encontrar los valores iniciales de P_i y P_g evaluando cada partícula del enjambre en la función objetivo $f(X)$.
4. Se actualizan las velocidades local y global con las siguientes ecuaciones [21], [22]:

$$G_i^{t+1} = \chi[V_i^t + c_1 rand_1(P_i - X_i) + c_2 rand_2(P_g - X_i)] \quad (25)$$

$$L_i^{t+1} = \chi[V_i^t + c_3 rand_3(P_i - X_i) + c_4 rand_4(P_{gi} - X_i)] \quad (26)$$

5. Se actualiza la velocidad unificada de la siguiente forma:

$$U_i^{t+1} = uG_i^{t+1} + (1 - u)L_i^{t+1} \quad (27)$$

6. Se actualiza la partícula como sigue:

$$X_i^{t+1} = X_i^t + U_i^{t+1} \quad (28)$$

7. Evaluar la función objetivo $f(X)$ para cada partícula.
8. Seleccionar la mejor partícula del enjambre y actualizar la mejor posición local.
9. Comparar la mejor posición de cada partícula con su posición actual, si su posición actual es mejor, actualizar P_{gi} .
10. Evaluar la mejor posición del enjambre en la función objetivo y comparar con P_g . Si ese valor es menor, entonces actualizar P_g de la siguiente manera: Como el objetivo de búsqueda es discreto, la información suministrada al enjambre va a ser discreto, es decir, las partículas serán atraídas hacia una posición discreta haciendo la aproximación de la mejor partícula global a su valor entero más cercano. De esta manera las partículas tendrán la información necesaria para buscar una posición discreta, moviéndose en el dominio continuo.
11. Se actualiza la velocidad de PSO para ser cargada a las velocidades de UPSO.

$$V_i^{t+1} = \chi(V_i^t + c_1 \text{rand}(P_i - X_i) + c_2 \text{rand}(P_g - X_i)) \quad (29)$$

12. Preguntar si $f(P_g) = 0$. Si es si, parar, si no, volver al paso 4.

Luego de implementar el algoritmo y a partir de los resultados encontrados en cada prueba, se comparan los resultados conseguidos con cada método. Los criterios de comparación de dichos métodos son: tiempo, número de iteraciones. En el Capítulo 7, se encuentra el algoritmo original de UPSO así como el algoritmo de PSO.

Utilizando la herramienta de computo Matlab® 7.0.4 de Mathworks® sobre la que se implementó el algoritmo con sus pruebas correspondientes. Las especificaciones del equipo de cómputo son las siguientes:

Tabla 2. Especificaciones del hardware usado para las pruebas.

Fabricante	Lenovo®
Modelo	Ideapad Z470
Procesador	Intel® Core™ i5-2410M @ 2,30 GHz
Memoria RAM	4.00 GB
Sistema Operativo	Microsoft® Windows™ 7 Ultimate 64-bits

5.1 EJERCICIO DE OPTIMIZACIÓN: EL CHEQUE DESCONOCIDO

Por medio del siguiente ejercicio, se busca comparar la rapidez del algoritmo desarrollado con otros métodos de optimización implementados en el pasado.

Enunciado del problema: un hombre cobra un cheque por d dólares y c centavos en un banco. El cajero por error le entrega c dólares y d centavos. El hombre nota el error cuando gasta 23 centavos, además en ese momento tiene $2d$ dólares y $2c$ centavos. ¿Cuál era el valor del cheque?[23], [24].

Solución analítica:

c	d
51	25

Por lo tanto el valor del cheque era de 25 dólares con 51 centavos[23].

Solución con el algoritmo:

Tabla 3. Resultados con el algoritmo DUPSO para el ejercicio del cheque

PRUEBA	c	d	TIEMPO [s]	# ITERACIONES
1	51	25	0,18	88
2	51	25	0,14	65
3	51	25	0,1060	44
4	51	25	0,2162	98
5	51	25	0,1396	61
6	51	25	0,1807	81
7	51	25	0,1246	54
8	51	25	0,1585	73
9	51	25	0,2652	253
10	51	25	0,1754	137
PROMEDIO			0,16862	95

De la Tabla 3 se observa que en la iteración 10 el algoritmo tomo 0,1754 segundos y 137 iteraciones para encontrar la respuesta y comparándolo con la primera prueba que tardo 0,18 segundos y 88 iteraciones en hallar esta respuesta. Para este ejercicio, las primeras cuatro pruebas se realizaron con un total de 100 partículas, las siguientes cuatro pruebas fueron hechas con 50 partículas y las dos últimas con 20.

En este trabajo las pruebas fueron realizadas con 100 partículas ya que muestra una relación equitativa entre número de iteraciones y tiempo. Si el número de partículas es muy pequeño, va a tomar más iteraciones en encontrar la respuesta, y si por el contrario, se toma un número elevado de partículas, el algoritmo seguramente tardará más tiempo en hallar la respuesta[25]

La comparación con CPSO se presenta en la **Tabla 4**. Las pruebas se pueden observar en el Anexo A.

Tabla 4. Resumen de datos obtenidos para el ejercicio del cheque

	CPSO	DUPSO
<i>c</i>	51	51
<i>d</i>	25	25
Error	0	0
Tiempo [s]	3,7016	0,16862
Iteraciones	2267	95
Partículas	100	100
Eficiencia	100%	100%

En la Tabla 4, se resumen los resultados promedio encontrados con cada algoritmo. En las primeras filas se muestran los resultados (raíces) encontradas con cada método de optimización utilizado. Con el método discreto de UPSO

(DUPSO) se encontraron los valores deseados, con menor tiempo de cómputo así como menor número de iteraciones.

5.2 EJERCICIO DE OPTIMIZACIÓN: NUMEROS CONSECUTIVOS

Enunciado del problema: Pruébese que 3,4,5 es la única solución de $x^2 + y^2 = z^2$ en enteros positivos consecutivos[24],[26]:

Solución Analítica:

Si x, y, z son tres enteros positivos consecutivos entonces se pueden escribir como $x = n - 1, y = n, z = n + 1$ con $n > 1$; en la ecuación (30) se observa el modelo del problema, la ecuación (31) es el desarrollo algebraico, y en las ecuaciones (32) y (33) se encuentran las respuestas del ejercicio.

$$(n - 1)^2 + n^2 = (n + 1)^2 \quad (30)$$

$$n^2 - 2 * n + 1 + n^2 = n^2 + 1 + 2 * n \quad (31)$$

$$n * (n - 4) = 0 \quad \rightarrow \quad n = 4 \quad (32)$$

$$x = n - 1 = 3, \quad y = n = 4, \quad z = n + 1 = 5 \quad (33)$$

Solución con el Algoritmo:

Tabla 5. Resultados con el algoritmo DUPSO para el ejemplo de los números consecutivos

PRUEBA	x	y	z	TIEMPO [s]	# ITERACIONES
1	3	4	5	0,9531	57
2	3	4	5	0,4656	33
3	3	4	5	1,3946	57
4	3	4	5	0,2862	18
5	3	4	5	0,3003	19
6	3	4	5	0,4483	24
7	3	4	5	0,4038	34
8	3	4	5	0,6715	38
9	3	4	5	0,3415	22
10	3	4	5	0,3522	26
PROMEDIO				0,5617	33

Con los resultados encontrados por el algoritmo implementado Tabla 5, se puede realizar una comparación con otros métodos de optimización. Las pruebas completas se pueden observar en el Anexo B.

Tabla 6. Resumen resultados encontrados para el ejercicio de los números consecutivos

	CPSO	DUPSO
x	3	3
y	4	4
z	5	5
Error	0	0
Tiempo [s]	2,3543	0,5617
Iteraciones	163	33

Partículas	100	100
eficiencia	100%	100%

En la Tabla 6 se hallan los resultados promedio encontrados con cada algoritmo. La convergencia y efectividad del algoritmo propuesto es notoria computacionalmente hablando, ya que el tiempo y las iteraciones son menores. Los resultados completos de CPSO se encuentran en el Anexo C.

5.3 EJERCICIO DE OPTIMIZACIÓN: TRIPLA PITAGORICA

Enunciado del problema: Encuéntrese todos los triángulos pitagóricos cuya área sea numéricamente igual a su perímetro

Solución Analítica:

Sean a, b, c los lados de un triángulo, donde c es la hipotenusa. En la ecuación (34) se modelan los triángulos rectángulos; si el área de dichos triángulos es igual a su perímetro, se plantea la ecuación (35) Se despejan las ecuaciones (34) y (35), y se desarrolla el ejercicio como muestran las ecuaciones

$$a^2 + b^2 = c^2 \quad (34)$$

$$a + b + c = \frac{a * b}{2} \quad (35)$$

$$c = \frac{a * b}{2} - a - b \quad (36)$$

$$a^2 + b^2 = \left(\frac{a * b}{2} - a - b\right)^2 \quad (37)$$

$$a^2 + b^2 = \frac{a^2 * b^2}{4} + (a + b)^2 - a * b * (a + b) \quad (38)$$

$$0 = a * b * \left(\frac{a * b}{4} - 2 * a * b\right) \quad (39)$$

Si a, b son diferentes de cero, entonces $a * b + 8 - 4a - 4b = 0$, por lo tanto, se desarrollan las ecuaciones (40) y (41) así mismo de la ecuación (42) se debe garantizar que $(a - 4) \neq 0$; en esta ecuación se observa que únicamente algunos valores de a mayores que 4, garantizan que la solución este en el rango de los enteros positivos \mathbf{Z}_+ ; es decir que $a = 5, 6, 8$ ó 12 .

$$b * (a - 4) = 4a - 8 = 4 * (a - 4) + 8 \quad (40)$$

$$(b - 4)(a - 4) = 8 \quad (41)$$

$$b = 4 + \frac{8}{a - 4} \quad (42)$$

Debido a que el ejercicio presenta múltiples soluciones solo se tendrán en consideración las soluciones enteras del algoritmo

POSIBLES SOLUCIONES

<i>a</i>	<i>b</i>	<i>c</i>
6	8	10
12	5	3
8	6	10
5	12	13

Solución con el algoritmo:

Tabla 7. Resultados encontrados con el algoritmo para para el ejercicio de la tripla pitagórica.

PRUEBA	<i>a</i>	<i>b</i>	<i>c</i>	TIEMPO [s]	#ITERACIONES
1	6	8	10	0,5297	115
2	6	8	10	0,0331	1
3	6	8	10	0,1005	12
4	5	12	13	0,0532	3
5	6	8	10	0,0313	1
6	6	8	10	0,1214	18
7	12	5	13	7,8656	1187
8	8	6	10	3,7008	673
9	8	6	10	0,0411	2
10	12	5	13	0,0966	12
11	12	5	13	0,1697	26
12	5	12	13	0,1485	23
13	8	6	10	0,0495	3
14	8	6	10	0,0872	7
15	5	12	13	0,0324	1
16	6	8	10	0,0820	6
17	12	5	13	0,1412	21

18	5	12	13	0,3092	60
19	6	8	10	0,0889	10
20	12	5	13	0,0529	3
PROMEDIO				0,6867	109

El algoritmo muestra las posibles soluciones que tiene el problema. Como este tiene múltiples soluciones, en cada prueba realizada se encontraron respuestas diferentes, pero todas las respuestas son válidas para esta ecuación. El algoritmo viaja a través de la información suministrada por las partículas y este puede converger a cualquier solución. En la Tabla 7 se puede observar que la séptima prueba, tiene una alta desviación de la tendencia del algoritmo debido a la heurística del método por la cual el enjambre puede iniciar en cualquier punto del enjambre, haciendo que el punto inicial del enjambre se encuentre en una ubicación cercana o lejana de la respuesta.

Tabla 8. Comparación de resultados para el ejercicio de la tripla pitagórica.

	CPSO	DUPSO
Error	0	0
Tiempo [s]	13,8087	0,6867
Iteraciones	2760	109
Partículas	100	100
eficiencia	100%	100%

La Tabla 8 contiene un resumen del tiempo y el número iteraciones promedio que tomo cada método en hallar una solución. El comportamiento de DUPSO en promedio es altamente satisfactorio, ya que requiere aproximadamente el 5% del tiempo requerido por CPSO para hallar una respuesta y lo mismo pasa con el número de iteraciones. Los datos completos del ejercicio implementado con CPSO se encuentran en el anexo C.

5.4 EJERCICIO DE OPTIMIZACIÓN: COMPRA DE CELULARES

Enunciado del problema: Una persona a través de una tienda en línea compra 12 celulares, unos de gama alta y otros de gama media, por 1200 dólares. Si los de gama alta valen 30 dólares más por unidad y ha comprado el mínimo posible de gama media, ¿Cuántos compró de cada uno?

Solución Analítica:

Se definen x, y, z como el número de teléfonos celulares de gama alta, el número celulares de gama media y el valor de los celulares de gama media respectivamente. Se plantean las ecuaciones (43) y (44). Se despeja, se resuelve y calcula encontrando las ecuaciones (45), (46),(47).

$$x + y = 12 \quad (43)$$

$$x * (z + 30) + y * z = 1200 \quad (44)$$

$$30x + 12z = 1200 \quad (45)$$

$$x_0 = \frac{1200 * 1}{6} = 200 \quad (46)$$

$$y_0 = \frac{1200 * (-2)}{6} = -400 \quad (47)$$

*Se realiza el proceso mencionado y se encuentra que cualquier solución para este problema se presenta como la ecuación(48). Si $200 + 2 * t$ debe ser menor o igual*

a 12 se plantea la ecuación (49). A continuación se encuentran los valores que satisfacen las ecuaciones (43) y (44). Dado que se desea comprar celulares gama media, pero en la mínima cantidad posible; la solución que se adapta a las condiciones del problema es la de la columna resaltada en color.

$$[200 + 2 * t, \quad -400 - 5 * t] \text{ con } t \in \mathbf{Z} - \quad (48)$$

$$200 + 2 * t \leq 12 \quad \rightarrow \quad t \leq -94 \quad (49)$$

SOLUCIONES

x	12	10	8	6	4	2	0
y	0	2	4	6	8	10	12
Z	70	75	80	85	90	95	100

Solución con el algoritmo.

A continuación se muestran los datos encontrados luego de la implementación del algoritmo:

Tabla 9. Resultados con el algoritmo para el ejercicio de la compra de celulares

PRUEBA	<i>x</i>	<i>y</i>	<i>z</i>	TIEMPO [s]	#ITERACIONES
1	12	0	70	0,7531	158
2	10	2	75	2,7714	517
3	4	8	90	0,2939	69

4	12	0	70	0,5427	98
5	12	0	70	0,1953	32
6	4	8	90	1,2441	286
7	8	4	80	0,3201	84
8	12	0	70	0,1828	43
9	6	6	85	2,3784	591
10	12	0	70	0,1897	35
11	4	8	90	0,4181	91
12	10	2	75	0,1465	29
13	4	8	90	0,1406	31
14	12	0	70	0,2033	50
15	10	2	75	0,2331	60
16	4	8	90	0,5113	147
17	6	6	85	0,3488	97
18	10	2	75	0,2659	70
19	12	0	70	0,1904	47
20	10	2	75	0,4551	131
PROMEDIO				0,5892	133

Como el problema es de múltiples soluciones, el algoritmo en cada ejecución hallará una de ellas. En la Tabla 9 se puede observar dicho comportamiento. En la ejecución del algoritmo se encontraron soluciones negativas, estas fueron desechadas ya que no cumplen con los requerimientos del problema. Analizando las soluciones, se concluye que la solución a este problema es:

<i>x</i>	<i>y</i>	<i>z</i>
10	2	75

Ya que esta solución garantiza que los celulares comprados de gama media sean mínimos.

La Tabla 10 muestra una comparación de los resultados encontrados por el algoritmo propuesto y por CPSO.

Tabla 10. Comparación de datos del ejercicio de la compra de celulares

	CPSO	DUPSO
Error	0	0
Tiempo [s]	543,376	0,5892
Iteraciones	103767	133
Partículas	100	100
eficiencia	100%	100%

Esta eficiencia de DUPSO fue medida a partir de la evaluación de la solución en la función, ya que cuando el algoritmo encuentra una solución, esta matemáticamente con los requerimientos. Los datos completos del ejercicio con el algoritmo de CPSO se encuentran en el Anexo C.

5.5 EJERCICIO DE OPTIMIZACIÓN: PROCESO DE FABRICACIÓN

Enunciado del problema: En una empresa de textiles se tienen seis máquinas que permiten confeccionar diferentes tipos de prendas (blusas, camisas, pantalones, chaquetas, toallas, fundas), si en la elaboración de cada una se emplea un determinado tiempo y tiene un gasto en mantenimiento, calcular cuantas prendas se pueden confeccionar al día. Las ecuaciones que modelan el sistema son (50), (51), (52), (53), (54) y (55) tomado de [24].

$$5u + 10v - 5w + y^3 + 8z = 1772 \quad (50)$$

$$3u + 18w - 5y + 17z = 153 \quad (51)$$

$$6u + w - y + (15z)^2 = 1772 \quad (52)$$

$$-u + 5v + 8w - 6x + 15y + 10z = 277 \quad (53)$$

$$(u + v)^2 - 7w + 5x + 12y - 8z = 150 \quad (54)$$

$$v - 5w + 3y - z = -4 \quad (55)$$

La solución analítica del problema es:

SOLUCION

<i>u</i>	<i>v</i>	<i>w</i>	<i>x</i>	<i>y</i>	<i>z</i>
6	3	8	1	12	3

Las anteriores respuestas son tomadas de [24]. De estas respuestas anteriores se puede deducir que se pueden confeccionar al día 6 blusas, 3 camisas, 8 pantalones, 1 chaqueta, 12 toallas y 3 fundas.

Solución con el algoritmo implementado: Los datos encontrados al implementar la función objetivo de este problema en el algoritmo se muestran en la Tabla 11.

Tabla 11. Resultados de la optimización con DUPSO del problema de la fábrica

PRUEBA	<i>u</i>	<i>v</i>	<i>w</i>	<i>x</i>	<i>y</i>	<i>z</i>	TIEMPO [s]	# ITERACIONES
1	6	3	8	1	12	3	4,2535	538
2	6	3	8	1	12	3	0,5320	76
3	6	3	8	1	12	3	7,8136	854
4	6	3	8	1	12	3	1,3194	184
5	6	3	8	1	12	3	1,5537	206
6	6	3	8	1	12	3	6,8037	789
7	6	3	8	1	12	3	1,6434	223
8	6	3	8	1	12	3	1,1015	157
9	6	3	8	1	12	3	1,0595	144
10	6	3	8	1	12	3	1,4435	197
11	6	3	8	1	12	3	4,2261	503
12	6	3	8	1	12	3	6,8156	758
13	6	3	8	1	12	3	3,2835	421
14	6	3	8	1	12	3	7,7864	834
15	6	3	8	1	12	3	1,1318	156
16	6	3	8	1	12	3	0,9717	139
17	6	3	8	1	12	3	2,1695	287
18	6	3	8	1	12	3	0,7452	107
19	6	3	8	1	12	3	9,5609	994
20	6	3	8	1	12	3	0,7513	107
PROMEDIO							3,2483	384

La comparación de estos resultados con los encontrados con el algoritmo CPSO se muestra en la Tabla 12.

Tabla 12. Comparación de los resultados de la optimización de la fábrica.

	CPSO	DUPSO
TIEMPO [s]	64,61035	3,2483
ITERACIONES	4754	384
PARTÍCULAS	100	100
EFICIENCIA	100%	100%

Como se puede observar en la Tabla 12 el algoritmo discreto de UPSO converge mucho más rápido que el de CPSO, así como el tiempo de procesamiento es significativamente menor. Los resultados completos del algoritmo se encuentran en el Anexo E.

5.6 EJERCICIO DE OPTIMIZACIÓN: GANANCIAS DE UN RESTAURANTE

Enunciado del problema: En un local se sirven tres tipos de menú. Un día se sirven 8,16 y 14 menús, respectivamente y la recaudación aumenta en 15 dólares. Sabiendo que el precio de los menús es múltiplo de 0.5. ¿ se puede deducir cual es el precio?[24].

Respuesta Analítica:

Si llamamos u al precio del menú del primer tipo, v al precio del menú del segundo tipo y w al precio del menú del tercer tipo, el problema consiste en resolver el sistema de ecuaciones (68) y (69). Si u, v, w son múltiplos de 0.5, por lo tanto, $x = 2 * u$, $y = 2 * v$, $z = 2 * w$ y se plantea el sistema de ecuaciones diofántica (70) y (71).

$$10u + 15v + 12w = 294 \quad (56)$$

$$8u + 16v + 14w = 309 \quad (57)$$

$$10x + 15y + 12z = 588 \quad (58)$$

$$4x + 8y + 7z = 309 \quad (59)$$

La única solución posible es $x=12$, $y=16$ y $z=19$. Es decir los precios de los menús son $u=6$ dólares, $v=8$ dólares y $w=9,5$ dolares.[24]

SOLUCIÓN		
x	y	z
12	16	19

Solución con el algoritmo:

En la Tabla 13 se encuentran los resultados de implementar el problema en el algoritmo propuesto.

Tabla 13. Resultados de la optimización con DUPSO para el problema del restaurante

PRUEBA	x	y	z	TIEMPO [s]	# ITERACIONES
1	12	16	19	8,8190	1558
2	12	16	19	0,1172	21
3	12	16	19	10,8178	1861
4	12	16	19	0,3129	79
5	12	16	19	0,6623	185

6	12	16	19	1,6025	405
7	12	16	19	0,2032	50
8	12	16	19	1,2018	332
9	12	16	19	0,1019	17
10	12	16	19	0,2144	54
11	12	16	19	0,4047	113
12	12	16	19	0,6428	186
13	12	16	19	0,1941	47
14	12	16	19	0,2503	63
15	12	16	19	0,1092	20
16	12	16	19	0,1598	37
17	12	16	19	8,0318	1548
18	12	16	19	0,1712	40
19	12	16	19	0,9900	279
20	12	16	19	0,2271	57
PROMEDIO				1,7617	348

La comparación de estos resultados con los encontrados con el algoritmo CPSO se muestra en la Tabla 14.

Tabla 14. Comparación con CPSO de los resultados del ejercicio del restaurante

	CPSO	DUPSO
TIEMPO [s]	67,1637	1,7617
ITERACIONES	18336	348
PARTÍCULAS	100	100
EFICIENCIA	100%	100%

Como se puede observar en la Tabla 14, el algoritmo discreto de UPSO converge mucho más rápido que el de CPSO, así como el tiempo de procesamiento es significativamente menor. Los resultados completos del algoritmo se encuentran en el Anexo F.

5.7 EJERCICIO DE OPTIMIZACIÓN: TRABAJO DE UN SASTRE

Enunciado del problema: Un sastre invierte 13 horas en diseñar un modelo de pantalón y 37 horas en diseñar un modelo de una camisa. Si trabaja 2000 horas, ¿Cuántas camisas y pantalones se deberán diseñar para conseguir la mayor combinación entre camisas y pantalones?[24]

Solución analítica:

Se define x como en número de pantalones, así como y como el número de camisas.

Según los requerimientos del problema, la siguiente es la ecuación a resolver:

$$13x + 37y = 2000 \quad (60)$$

Este ejercicio tiene múltiples soluciones, las cuales se muestran a continuación:

x	y	$x * y$
3	53	159
40	40	1600
77	27	2079
114	14	1596

La solución que garantiza una combinación que permite la producción máxima del sastre en las 2000 horas es $x = 77$, $y = 27$.

Solución con el algoritmo:

La Tabla 15, muestra las soluciones obtenidas con el algoritmo, así como su tiempo de cómputo y número de iteraciones.

Tabla 15. Resultados con DUPSO para el problema del sastre

PRUEBA	<i>x</i>	<i>y</i>	TIEMPO [s]	# ITERACIONES
1	40	40	0,4349	84
2	77	27	0,2805	75
3	77	27	0,0997	32
4	40	40	0,5897	251
5	3	53	0,2474	98
6	40	40	0,5211	213
7	3	53	1,3894	471
8	77	27	0,1250	44
9	114	14	0,0965	29
10	40	40	0,0433	4
11	114	14	0,1858	72
12	40	40	0,1320	21
13	40	40	0,0717	18
14	3	53	0,0999	31
15	40	40	0,0626	10
16	3	53	1,9590	713
17	3	53	0,9320	388
18	40	40	0,6796	292
19	114	14	0,3845	165
20	40	40	0,0558	9
PROMEDIO			0,4195	151

A continuación en la Tabla 16 se muestra una comparación de los resultados de DUPSO con CPSO.

Tabla 16. Comparación de resultados para el ejercicio del sastre

	CPSO	DUPSO
TIEMPO [s]	64,61035	0,41952
ITERACIONES	4911	151
PARTÍCULAS	100	100
EFICIENCIA	100%	100%

El tiempo mostrado es el tiempo que tarda el algoritmo en encontrar una respuesta. De la Tabla 16 DUPSO es mucho más eficiente computacionalmente hablando, ya que el tiempo e iteraciones por ejecuciones son menores al de CUPSO, sin embargo en términos de respuesta, los dos métodos llegaron a ella. Los resultados completos del ejercicio con el algoritmo CPSO se encuentran en el Anexo G.

5.8 EJERCICIO DE OPTIMIZACIÓN: EL GRANJERO Y SUS ANIMALES

Enunciado del problema: Un granjero gastó 10.000.000 de pesos, en 100 animales entre pollos, cerdos y terneros. Si compró los pollos los a 5.000 pesos, los cerdos a 100.000 pesos y los terneros a 500.000 pesos, y si adquirió animales de las tres clases, ¿Cuántos compró de cada uno?[24]. Siendo x el número de pollos, y el número de cerdos y z el número de terneros.

Solución analítica:

De acuerdo con el enunciado del problema se tiene el sistema de ecuaciones (68), (69), simplificando (69) se obtiene (70).

$$x + y + z = 100 \quad (61)$$

$$5000x + 100000y + 500000z = 10000000 \quad (62)$$

$$x + 20y + 100z = 2000 \quad (63)$$

Solucionando (70) se encuentra la siguiente solución.

$$x \quad y \quad z$$

$$80 \quad 1 \quad 19$$

Por lo tanto el granjero compro 80 pollos, 1 cerdo y 19 terneros.

Solución con el algoritmo:

Tabla 17. Resultados con DUPSO del problema del granjero

PRUEBA	<i>x</i>	<i>y</i>	<i>z</i>	TIEMPO [s]	# ITERACIONES
1	80	1	19	4,1425	894
2	80	1	19	74,9684	6095
3	80	1	19	217,2607	10957
4	80	1	19	0,4006	85
5	80	1	19	11,0177	1879
6	80	1	19	0,1962	46
7	80	1	19	7,9368	1498
8	80	1	19	8,5899	1532
9	80	1	19	0,4686	129
10	80	1	19	0,2959	61

11	80	1	19	10,4995	1538
12	80	1	19	15,7799	2287
13	80	1	19	3,6910	852
14	80	1	19	8,2187	1549
15	80	1	19	34,7343	3844
16	80	1	19	8,3004	1559
17	80	1	19	0,2739	69
18	80	1	19	1,9389	507
19	80	1	19	5,9721	1214
20	80	1	19	10,7548	1849
PROMEDIO				21,2720	1922

Con los resultados de la Tabla 17, se procede a realizar una comparación de sus resultados con el algoritmo de CPSO, el cual sus resultados completos se encuentran en el ANEXO G.

Tabla 18. Comparación de datos del problema del granjero

	CPSO	DUPSO
TIEMPO [s]	80,3492	21,2720
ITERACIONES	28425	1922
PARTÍCULAS	100	100
EFICIENCIA	100%	100%

De la Tabla 18 podemos observar que el algoritmo discreto de UPSO converge mucho más rápido que el de CPSO, así como el tiempo de procesamiento es significativamente menor. Los datos completos del ejercicio implementado con CPSO se encuentran en el Anexo H.

5.9 EJERCICIO OPTIMIZACIÓN: COMPRA DE INTERIORES

Enunciado del problema: Un hombre compró interiores largos y cortos por 99 dólares. Si un interior largo cuesta 3 dólares más que uno corto, y compró más interiores largos que cortos, ¿Cuántos interiores compró de cada modelo?[24]

Solución analítica:

Denotamos como x como el número de interiores largos, y el número de interiores cortos y z es el precio de un interior corto. Se plantean las ecuaciones (64) y (65).

$$x + y = 12 \tag{64}$$

$$(z + 3) * x + z * y = 99 \tag{65}$$

La solución analítica es $x = 9$, $y = 3$ y $z = 6$.

Solución con el algoritmo:

Tabla 19. Resultados del problema de los interiores con DUPSO

PRUEBA	x	y	z	TIEMPO [s]	# ITERACIONES
1	1	11	8	0,1188	13
2	9	3	6	0,1802	39
3	1	11	8	0,1259	18
4	9	3	6	0,1676	31
5	1	11	8	0,1969	39
6	9	3	6	0,0483	2
7	5	7	7	0,2163	25
8	9	3	6	0,3038	41

9	9	3	6	0,0953	9
10	1	11	8	0,1941	20
11	1	11	8	0,0800	8
12	1	11	8	0,1682	29
13	1	11	8	0,3714	78
14	9	3	6	0,1829	33
15	9	3	6	0,2945	71
16	9	3	6	0,2397	46
17	9	3	6	0,0895	10
18	1	11	8	0,1766	39
19	5	7	7	0,2060	45
20	5	7	7	0,4710	115
PROMEDIO				0,1964	35

Este es un problema de múltiples soluciones, y las soluciones negativas que no tienen relación con el problema son rechazadas. Se dispone a analizar las respuestas y la que corresponde con la respuesta es $x = 9$, $y = 3$ y $z = 6$.

Con los resultados de la Tabla 19 y con los datos encontrados al resolver el problema con el algoritmo de CPSO, se procede a realizar una comparación de resultados. Los resultados completos del problema resuelto con CPSO se encuentran en el ANEXO I.

Tabla 20. Comparación de resultados para el ejercicio de los interiores.

	CPSO	DUPSO
TIEMPO [s]	49,7383	0,1964
ITERACIONES	10097	35
PARTÍCULAS	100	100
EFICIENCIA	100%	100%

5.10 EJERCICIO DE OPTIMIZACIÓN: DISEÑO DE MECANISMOS

Se necesitan dos ruedas dentadas, una de 237 dientes y otra de 295. Se dispone solamente de ruedas de 10, 15, 20, 25, 30, 35 y 40 dientes. Determinar los dientes de cada una de ellas para efectuar el trabajo[27], [24].

Las ruedas utilizadas son de 25 y 20 dientes, ya que son múltiplos de 5. Denotemos x como los dientes utilizados por la rueda de 25 e y como los dientes utilizados por la rueda de 20 y se plantean las siguientes ecuaciones:

$$237x - 295y = 5 \tag{66}$$

$$94x - 117y = 2 \tag{67}$$

Este problema tiene como solución $x = 5$ e $y = 4$ [27]

Solución con el algoritmo:

Tabla 21. Resultados del problema de diseño de engranajes con DUPSO

PRUEBA	x	y	TIEMPO [s]	# ITERACIONES
1	5	4	0,0417	2
2	5	4	0,2667	79
3	5	4	2,4621	662
4	5	4	0,0279	1
5	5	4	0,1422	36
6	5	4	0,0818	12
7	5	4	3,5690	892
8	5	4	1,2124	343

9	5	4	1,8948	512
10	5	4	10,1660	2014
11	5	4	0,6319	212
12	5	4	0,6125	188
13	5	4	5,8550	1376
14	5	4	0,3263	93
15	5	4	12,8152	2374
16	5	4	0,0994	18
17	5	4	1,2652	350
18	5	4	15,8737	2736
19	5	4	1,6888	478
20	5	4	4,3890	1054
PROMEDIO			3,1711	671

Con los resultados mostrados en la Tabla 21 realizamos una comparación con los datos obtenidos con el algoritmo de CPSO.

Tabla 22. Comparación de resultados para el ejercicio del engranaje.

	CPSO	DUPSO
TIEMPO [s]	49,7383	4,1711
ITERACIONES	10097	671
PARTÍCULAS	100	100
EFICIENCIA	100%	100%

En la Tabla 22 se puede observar que los tiempos de cómputo del algoritmo propuesto en este trabajo funciona de forma más rápida para el ejercicio que el

método de CPSO. Los resultados completos del ejercicio implementado en el algoritmo CPSO se encuentran en el Anexo J.

6 APLICACIÓN DEL ALGORITMO EN PROBLEMAS PRESENTADOS EN INGENIERÍA ELECTRÓNICA

Esta sección se desarrolla con el fin de cumplir con los siguientes objetivos específicos:

- Seleccionar un problema en ingeniería electrónica (Sistemas de control o criptografía) modelado mediante ecuaciones diofánticas.
- Resolver el problema seleccionado mediante la estrategia propuesta y compararlo con algunos resultados reportados en la literatura.

Las ecuaciones diofánticas pueden surgir al modelar algunos sistemas físicos y darle solución a estas se hace indispensable. Un ejemplo de estos problemas se presenta en la teoría de control donde el comportamiento de un sistema está caracterizado por sistemas de ecuaciones polinomiales. Dicho comportamiento también puede ser caracterizado a través de ecuaciones diofánticas y pueden ser resueltas con el algoritmo implementado en este trabajo de grado.

El problema seleccionado representa un escenario donde las ecuaciones diofánticas y el control se unen en un problema particular. En este caso, se deben hallar los coeficientes de los polinomios del numerador y el denominador del controlador. Así mismo en [3] se da solución a este problema mediante el algoritmo de Euclides, cuya respuesta pertenece a los números enteros.

6.1 SISTEMA DE CONTROL DE LAZO CERRADO CON REALIMENTACION UNITARIA.

Como una primera prueba, se toma el sistema de control de lazo cerrado de la figura 1, con realimentación unitaria, donde se requiere encontrar el controlador $C(s)$, mediante el método polinomial, que tenga 6 polos en -1, [3] [24].

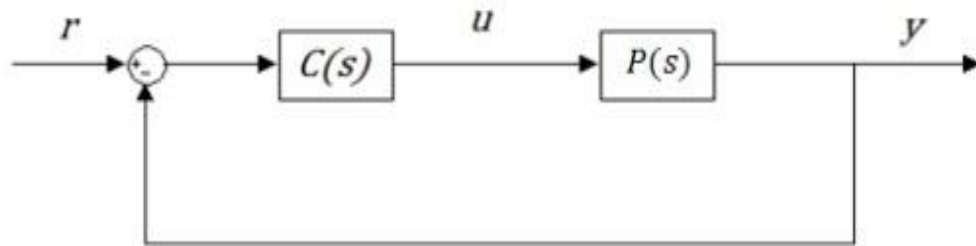


Figura 1. Diagrama de bloques del sistema con realimentación unitaria.

Donde

$$P(s) = \frac{s^2 + s + 1}{s^3 + 3s^2 + 4s + 3} \quad (68)$$

Como es conocido de la teoría de control la función de transferencia en lazo cerrado del sistema se define como:

$$\frac{y(s)}{r(s)} = \frac{C(s) * P(s)}{1 + C(s) * P(s)} \quad (69)$$

Debido al enfoque polinomial que requiere la solución de estos problemas redefinimos la planta del sistema y el controlador de la siguiente forma:

$$P(s) = \frac{B(s)}{A(s)} \quad (70)$$

$$C(s) = \frac{N(s)}{D(s)} \quad (71)$$

Con esta nueva notación de $H(s)$ y $C(s)$ tenemos que la función de transferencia del sistema en lazo cerrado es la siguiente:

$$\frac{y(s)}{r(s)} = \frac{\frac{N(s)}{D(s)} * \frac{B(s)}{A(s)}}{1 + \frac{N(s)}{D(s)} * \frac{B(s)}{A(s)}} = \frac{N(s) * B(s)}{A(s) * D(s) + B(s) * N(s)} \quad (72)$$

De la ecuación (72), utilizando el denominador encontramos la ecuación diofántica que se desea resolver:

$$F(s) = A(s) * D(s) + B(s) * N(s) \quad (73)$$

Donde los parámetros desconocidos son $D(s)$ y $N(s)$, debido a que los otros polinomios ya son conocidos.

$$A(s) = s^3 + 3s^2 + 4s + 3 \quad (74)$$

$$B(s) = s^2 + s + 1 \quad (75)$$

$$D(s) = x_1 s^3 + x_2 s^2 + x_3 s + x_4 \quad (76)$$

$$N(s) = x_5 s^2 + x_6 s + x_7 \quad (77)$$

$$F(s) = (s + 1)^6 \quad (78)$$

La solución analítica de este problema se puede encontrarla en [3], [24]:

$$x_1 = 1, \quad x_2 = 3, \quad x_3 = 2, \quad x_4 = 2, \quad x_5 = 0, \quad x_6 = -3, \\ x_7 = -5,$$

Con la solución teórica de este problema tenemos que el controlador $C(s)$ es de la siguiente manera:

$$C(s) = \frac{-3s - 5}{s^3 + 3s^2 + 2s + 2}$$

SOLUCIÓN CON EL ALGORITMO:

Para llevar a cabo la solución de este problema utilizando el algoritmo, es necesario reemplazar las ecuaciones (74), (75), (76), (77) y (78) en la ecuación (73). De este reemplazo se llega a la siguiente ecuación:

$$\begin{aligned} & ((s^3 + 3s^2 + 4s + 3) * (x_1 s^3 + x_2 s^2 + x_3 s + x_4)) + ((s^2 + s + 1) * (x_5 s^2 + x_6 s + x_7)) \\ & = s^6 + 6s^5 + 15s^4 + 20s^3 + 15s^2 + 6s + 1 \end{aligned} \quad (79)$$

Expandiendo la ecuación (79) se consigue la siguiente ecuación:

$$\begin{aligned}
& s^6(x_1) + s^5(3x_1 + x_2) + s^4(4x_1 + 3x_2 + x_3 + x_5) \\
& + s^3(3x_1 + 4x_2 + 3x_3 + x_4 + x_5 + x_6) \quad (80) \\
& + s^2(3x_2 + 4x_3 + x_4 + x_5 + x_6 + x_7) + s(3x_3 + 4x_4 + x_6 + x_7) \quad) \\
& + (3x_4 + x_7) = (s^6 + 6s^5 + 15s^4 + 20s^3 + 15s^2 + 6s + 1)
\end{aligned}$$

De la ecuación (80) se plantea el sistema de ecuaciones del cual se parte para definir la función objetivo.

$$x_1 - 1 = 0 \quad (81)$$

$$3x_1 + x_2 - 6 = 0 \quad (82)$$

$$4x_1 + 3x_2 + x_3 + x_5 - 15 = 0 \quad (83)$$

$$3x_1 + 4x_2 + 3x_3 + x_4 + x_5 + x_6 - 20 = 0 \quad (84)$$

$$3x_2 + 4x_3 + 3x_4 + x_5 + x_6 + x_7 - 15 = 0 \quad (85)$$

$$3x_3 + 4x_4 + x_6 + x_7 - 6 = 0 \quad (86)$$

$$3x_4 + x_7 - 1 = 0 \quad (87)$$

Con las ecuaciones (81), (82), (83), (84), (85), (86) y (87) se plantea la función objetivo de la siguiente manera:

$$\begin{aligned}
f(x) = & (x_1 - 1)^2 + (3x_1 + x_2 - 6)^2 + (4x_1 + 3x_2 + x_3 + x_5 - 15)^2 \quad (88) \\
& + (3x_1 + 4x_2 + 3x_3 + x_4 + x_5 + x_6 - 20)^2 \quad) \\
& + (3x_2 + 4x_3 + 3x_4 + x_5 + x_6 + x_7 - 15)^2 + (3x_3 + 4x_4 + x_6 + x_7 - 6)^2 \\
& + (3x_4 + x_7 - 1)^2
\end{aligned}$$

Con la función objetivo definida en la ecuación (88) se procede a ejecutar el algoritmo, en el cual las constantes $c_1=2$, $c_2=2$ y $\chi=0,72$ como se han venido utilizando en todos los ejercicios anteriores.

Con el algoritmo discreto DUPSO se consiguen los siguientes resultados los cuales serán comparados con algoritmos previamente mencionados, para evaluar su eficiencia con relación a tiempo de cómputo, número de iteraciones y porcentaje de éxito:

Tabla 23. Resultados con el algoritmo DUPSO para diseño del controlador.

PRUEBA	X1	X2	X3	X4	X5	X6	X7	TIEMPO (S)	# ITERACIONES
1	1	3	2	2	0	-3	-5	21,5137	1647
2	1	3	2	2	0	-3	-5	8,4562	827
3	1	3	2	2	0	-3	-5	7,9145	804
4	1	3	2	2	0	-3	-5	16,3527	1335
5	1	3	2	2	0	-3	-5	22,9968	1734
6	1	3	2	2	0	-3	-5	17,8654	1393
7	1	3	2	2	0	-3	-5	4,4405	465
8	1	3	2	2	0	-3	-5	11,9982	1123
9	1	3	2	2	0	-3	-5	4,4606	420
10	1	3	2	2	0	-3	-5	45,3048	2310
11	1	3	2	2	0	-3	-5	17,4022	1166
12	1	3	2	2	0	-3	-5	6,8946	574
13	1	3	2	2	0	-3	-5	62,3057	2756
14	1	3	2	2	0	-3	-5	1,7097	136
15	1	3	2	2	0	-3	-5	59,3573	3255
16	1	3	2	2	0	-3	-5	45,215	2620
17	1	3	2	2	0	-3	-5	20,0193	1464
18	1	3	2	2	0	-3	-5	52,1366	2738
19	1	3	2	2	0	-3	-5	23,5473	1583
20	1	3	2	2	0	-3	-5	26,0868	1552
PROMEDIO								23,7989	1495

De la Tabla 23 se obtiene que los resultados son:
 $x_1 = 1, x_2 = 3, x_3 = 2, x_4 = 2, x_5 = 0, x_6 = -3, x_7 = -5$.

Comparándolo con CPSO cuyas pruebas completas se pueden observar en el Anexo K se obtienen los siguientes resultados:

Tabla 24. Resumen de datos para el diseño del controlador.

	CPSO	DUPSO
Error	0	0
Tiempo [s]	346,8940	23,7989
Iteraciones	16018	1495
Partículas	100	100
eficiencia	100%	100%

En la Tabla 24 se puede observar la mejora que trae consigo llevar velocidades continuas y un objetivo de búsqueda discreto, ya que el tiempo de cómputo es hasta 6 veces menor y con un promedio de iteraciones mucho más bajo. CPSO tiene un promedio de tiempo de cómputo e iteraciones alto, ya que la base de este algoritmo es probar cual partícula funciona y cual no, y es así como puede llegar a la respuesta rápidamente, ya que puede encontrar una solución satisfactoria también puede demorarse en encontrar la partícula adecuada. En UPSO el factor de unificación que aprovecha al máximo la exploración y la explotación y al combinar esto con un objetivo de búsqueda discreto, los resultados son satisfactorios tanto en tiempo como en iteraciones.

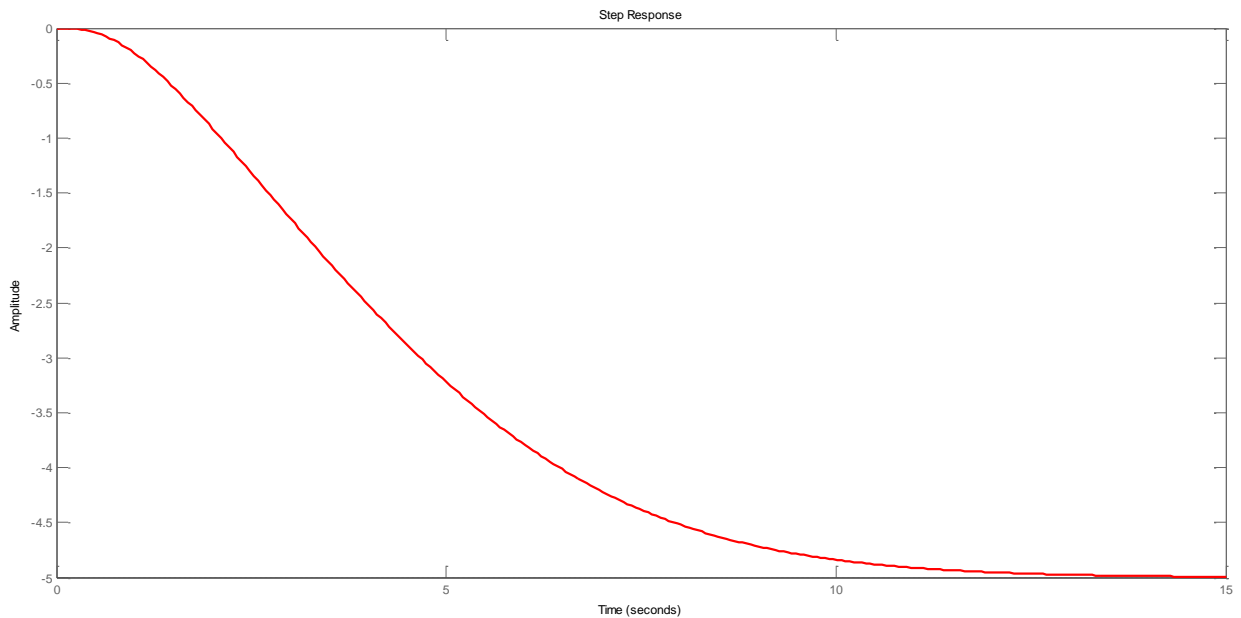


Figura 2. Respuesta al escalón del sistema de control.

Como se puede observar en la figura 2, este sistema de control no es apropiado. Este ejemplo es solo ilustrativo, donde lo que se buscó en este trabajo fue reproducir los resultados como objetivo de investigación. Además, este ejemplo fue tomado como un caso particular donde los coeficientes del controlador son enteros y poder aplicar el algoritmo propuesto.

A continuación, el algoritmo discreto será forzado a hallar los coeficientes de un controlador que están en el dominio continuo, y así trabajar con un sistema real donde el controlador si puede ser implementado.

6.2 SISTEMA DE CONTROL DISCRETO MEDIANTE LA IGUALACIÓN A UN MODELO

En aras de incrementar el nivel de exigencia en las pruebas realizadas al algoritmo discreto (DUPSO), se decidió tomar un segundo ejemplo. En este caso, se presenta un ejercicio tomado textualmente de [17], para evitar sesgar la información con el punto de vista de los autores.

Enunciado del problema: *Considere la planta definida por:*

$$\frac{B(z)}{A(z)} = \frac{0.3679z + 0.2642}{z^2 - 1.3679z + 0.3679} \quad (89)$$

Suponga que el periodo de muestreo es 1s. Se desea diseñar un sistema de control tal que el sistema en lazo cerrado sea:

$$\frac{Y_m(z)}{R_m(z)} = \frac{0.62z - 0.3}{z^2 - 1.2z + 0.5} = G_{\text{modelo}} \quad (90)$$

Para resolver el problema, se empleará el diagrama de bloques de la siguiente figura:

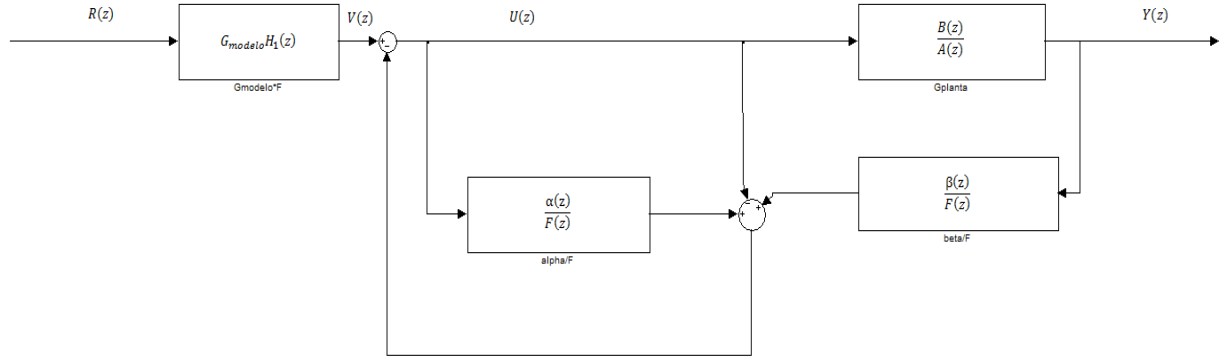


Figura 3. Diagrama de Bloques del sistema de control mediante la igualación de un modelo. Basado en [17]

Se supone que la planta es de estado completamente controlable y completamente observable, esto es, que no existen factores comunes entre $A(z)$ y $B(z)$. Para encontrar $\alpha(z)$ y $\beta(z)$ se resuelve la siguiente ecuación Diofántica:

$$\alpha(z)A(z) + \beta(z)B(z) = F(z)B(z)H_1(z) \quad (91)$$

Es necesario aclarar en (91) que $H_1(z)$ es un polinomio estable de grado $n - 1$ que no es único, puede ser seleccionado arbitrariamente y $F(z)$ es un polinomio estable de grado $n - 1$ el cual puede ser seleccionado de manera arbitraria.

De la Figura 3 se obtiene la función de transferencia pulso del sistema:

$$U(z) = - \left[\frac{\alpha(z)}{F(z)} U(z) - U(z) + \frac{\beta(z)}{F(z)} Y(z) \right] + V(z) \quad (92)$$

De otra forma:

$$\frac{\alpha(z)}{F(z)}U(z) + \frac{\beta(z)}{F(z)}Y(z) = V(z) \quad (93)$$

Ya que

$$U(z) = \frac{A(z)}{B(z)}Y(z) \quad (94)$$

Se tiene

$$\frac{Y(z)}{V(z)} = \frac{F(z)B(z)}{\alpha(z)A(z) + \beta(z)B(z)} = \frac{F(z)B(z)}{F(z)B(z)H_1(z)} = \frac{1}{H_1(z)} \quad (95)$$

Y como

$$V(z) = G_{\text{modelo}}H_1(z)R(z) \quad (96)$$

Entonces

$$\frac{Y(z)}{R(z)} = \frac{Y(z)V(z)}{V(z)R(z)} = \frac{G_{\text{modelo}}H_1(z)}{H_1(z)} = G_{\text{modelo}} \quad (97)$$

La función de transferencia pulso entre la entrada y la salida es igual a G_{modelo} .

Así se logra el control mediante la igualación de un modelo.

Ahora se halla $\alpha(z)$ y $\beta(z)$ resolviendo la ecuación Diofántica propuesta para realizar el controlador (91).

Siendo

$$A(z) = z^2 - 1.3679z + 0.3679 \quad (98)$$

$$B(z) = 0.3679z + 0.2642 \quad (99)$$

Se seleccionan $H_1(z)$ y $F(z)$ arbitrariamente cumpliendo los requerimientos antes mencionados (es decir, estabilidad y orden del polinomio). Por ejemplo, se puede tomar:

$$H_1(z) = z + 0.5 \quad (100)$$

$$F(z) = z \quad (101)$$

Donde $\alpha(z)$ y $\beta(z)$ tienen la siguiente forma:

$$\alpha(z) = X'_1 z + X'_2 \quad (102)$$

$$\beta(z) = X'_3 z + X'_4 \quad (103)$$

De (102) y (103) se tiene que $\alpha(z)$ y $\beta(z)$ son polinomios desconocidos de orden $n - 1$, cuyos coeficientes se buscan a través del algoritmo. Por tanto, se dispone a resolver la siguiente ecuación Diofántica:

$$\alpha(z)A(z) + \beta(z)B(z) = F(z)B(z)H_1(z) \quad (104)$$

$$\begin{aligned} (X'_1 z + X'_2)(z^2 - 1.3679z + 0.3679) + (X'_3 z + X'_4)(0.3679z + 0.2642) \\ = (z)(0.3679z + 0.2642)(z + 0.5) \end{aligned} \quad (105)$$

$$\begin{aligned} (X'_1 - 0.3679)z^3 + (-1.3679X'_1 + X'_2 + 0.3679X'_3 - 0.4482X'_4)z^2 \\ + (0.3679X'_1 - 1.3679X'_2 - 0.2642X'_3 + 0.3679X'_4 - 0.1321)z \\ + (0.3679X'_2 + 0.2642X'_4) = 0 \end{aligned} \quad (106)$$

Siguiendo un proceso analítico, y utilizando la ecuación de Sylvester y las formas canónicas, se encuentran los siguientes resultados:

X'_1	X'_2	X'_3	X'_4
0,3679	0,2642	1,8680	-0,3679

Por otra parte, y para utilizar el algoritmo propuesto como alternativa de solución (es decir, DUPSO), se requiere crear una función objetivo. En este caso, se compone de la suma de cada uno de los coeficientes de la ecuación elevados al cuadrado, como se muestra en la ecuación (107):

$$F_{\text{objetivo}} = f_1^2 + f_2^2 + f_3^2 + f_4^2 \quad (107)$$

Dónde:

$$f_1 = X'_1 - 0.3679 \quad (108)$$

$$f_2 = -1.3679X'_1 + X'_2 + 0.3679X'_3 - 0.4482X'_4 \quad (109)$$

$$f_3 = 0.3679X'_1 - 1.3679X'_2 - 0.2642X'_3 + 0.3679X'_4 - 0.1321 \quad (110)$$

$$f_4 = 0.3679X'_2 + 0.2642X'_4 \quad (111)$$

Como los coeficientes del controlador son números fraccionarios, se utilizará el método para generar un enjambre de partículas que busque solución tanto para el numerador como para el denominador, debido a que el algoritmo propuesto es discreto. Para ello, se establecieron los siguientes parámetros: $\chi = 0.72$, $c_1 = 2$, $c_2 = 2$, $u = 0.5$. Las soluciones encontradas se resumen en la Tabla 25 y los valores decimales correspondientes se presentan en la Tabla 26.

Tabla 25. Resultados de forma fraccionaria hallados con DUPSO.

PRUEBA	NUMERADORES				DENOMINADORES			
	X_1	X_2	X_3	X_4	X_5	X_6	X_7	X_8
1	15516	6977	9310	-8913	42175	26407	4984	24225
2	5234	1958	30654	-13239	14227	7411	16410	35981
3	4909	6648	7197	-8460	13344	25162	3853	22997
4	3193	-6728	15000	-20347	8679	-25464	8030	55306
5	2021	3020	3094	-22795	5492	11429	1656	61927
6	3300	1774	8645	-2781	8970	6714	4628	7559
7	3874	2775	53782	-2138	10530	10503	28791	5811
8	16140	4055	7817	-707	43874	15348	4185	1922
9	3719	3441	5572	-10296	10109	13024	2983	27986
10	6382	12502	-13963	-17335	17347	47318	-7475	47119
11	15111	5310	13373	-5341	41073	20097	7159	14517
12	2875	7327	-31750	-14013	7815	27732	-16997	38087
13	5149	1993	3833	-19573	13996	7543	2052	53206
14	4083	8723	95911	-4152	11098	33015	51344	11285
15	5499	4644	-48822	-6549	14947	17577	-26136	17799
16	9648	7253	7472	-11677	26225	27452	4000	31737
17	10606	2382	6641	-14005	28827	9015	3555	38063
18	18615	10150	44445	-15449	50599	38416	23793	41992
19	4975	1466	8720	-10749	13523	5549	4668	29211

20	19363	10236	11391	-32233	52632	38742	6098	87605
----	-------	-------	-------	--------	-------	-------	------	-------

Dónde:

$$X'_1 = \frac{X_1}{X_5}$$

$$X'_2 = \frac{X_2}{X_6}$$

$$X'_3 = \frac{X_3}{X_7}$$

$$X'_4 = \frac{X_4}{X_7}$$

Tabla 26. Parámetros del controlador encontrados con DUPSO

prueba	X'_1	X'_2	X'_3	X'_4	Tiempo [s]	# Iteraciones	Función Objetivo
1	0,3679	0,2642	1,8680	-0,3679	3,3962	346	7,70E-11
2	0,3679	0,2642	1,8680	-0,3679	26,4759	1737	9,71E-11
3	0,3679	0,2642	1,8679	-0,3679	22,4952	1556	8,66E-10
4	0,3679	0,2642	1,8680	-0,3679	4,3610	439	5,44E-11
5	0,3680	0,2642	1,8684	-0,3681	23,8313	1633	1,14E-08
6	0,3679	0,2642	1,8680	-0,3679	33,7647	2066	5,42E-10
7	0,3679	0,2642	1,8680	-0,3679	3,2114	332	7,50E-11
8	0,3679	0,2642	1,8679	-0,3678	24,6051	1690	1,55E-09
9	0,3679	0,2642	1,8679	-0,3679	24,3230	1682	7,53E-10
10	0,3679	0,2642	1,8680	-0,3679	135,1624	5000	3,79E-10
11	0,3679	0,2642	1,8680	-0,3679	3,3293	346	6,19E-11
12	0,3679	0,2642	1,8680	-0,3679	133,997	5000	4,36E-10
13	0,3679	0,2642	1,8679	-0,3679	24,2485	1666	4,18E-10
14	0,3679	0,2642	1,8680	-0,3679	4,0760	342	3,99E-11
15	0,3679	0,2642	1,8680	-0,3679	2,9832	315	8,79E-11
16	0,3679	0,2642	1,8680	-0,3679	3,5407	368	9,79E-11
17	0,3679	0,2642	1,8681	-0,3679	25,6162	1741	5,61E-10
18	0,3679	0,2642	1,8680	-0,3679	3,0742	323	9,22E-11
19	0,3679	0,2642	1,8680	-0,3680	24,3411	1678	8,95E-10
20	0,3679	0,2642	1,8680	-0,3679	2,3044	247	9,11E-11
Promedio	0,3679	0,2642	1,8680	-0,3679	26,4568	1425	9,3017E-10
Desviación Estándar	2,3495E-05	1,0171E-05	9,7254E-05	4,9952E-05	38,5789	1397	2,5026E-09

El valor de la función objetivo presentado en la Tabla 26 es la minimización de la función objetivo. El método llega a este valor rápidamente, encontrando una respuesta muy cercana a la buscada. Sin embargo, la precisión que se reporta en la tabla corresponde a la máxima que se logró alcanzar, debido principalmente a que la velocidad de las partículas es muy pequeña en este punto (pues la misma depende del valor de la función objetivo). Para validar los resultados obtenidos, se optó por graficar la respuesta al escalón y compararla con el modelo. La Figura 4 muestra el diagrama de bloques del sistema implementado.

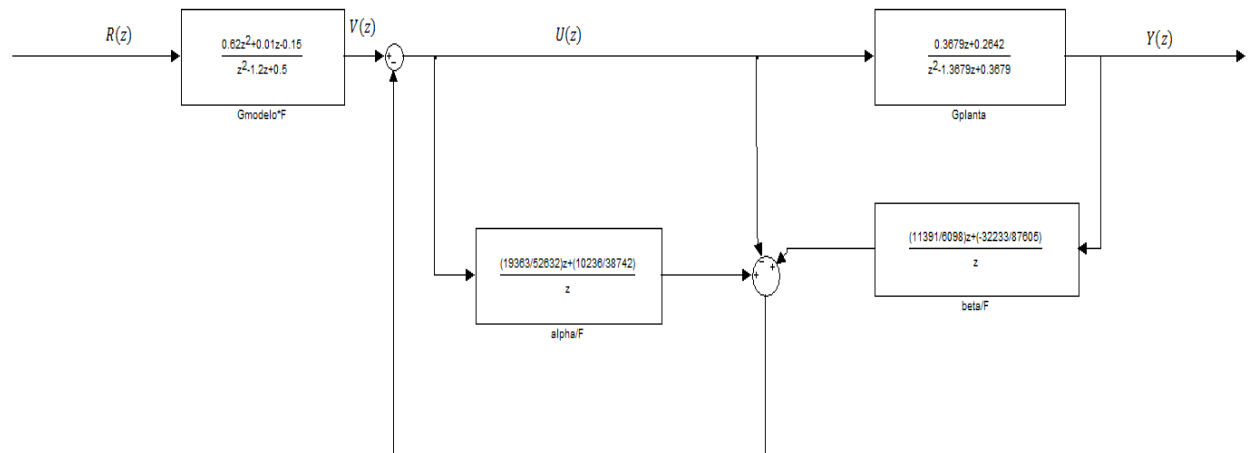


Figura 4. Diagrama de bloques del sistema de control mediante la igualación de un modelo. Basado en [17]

Dónde:

$$\alpha(z) = \frac{19363}{52632}z + \frac{10236}{38742} \quad (112)$$

$$\beta(z) = \frac{11391}{6098}z - \frac{32233}{87605} \quad (113)$$

Los valores de los coeficientes de (112) y (113) fueron tomados de Tabla 25, los cuales representan las soluciones más cercanas al valor medio de la muestra. Los polinomios $H_1(z)$ y $F(z)$ se seleccionaron de la siguiente manera:

$$H_1(z) = z + 0.5 \tag{114}$$

$$F(z) = z \tag{115}$$

Al implementar el diagrama de bloques de la Figura 4 se garantiza que la función de transferencia pulso sea igual al modelo planteado como se demostró anteriormente.

Los resultados de la simulación para el sistema de la Figura 4 son:

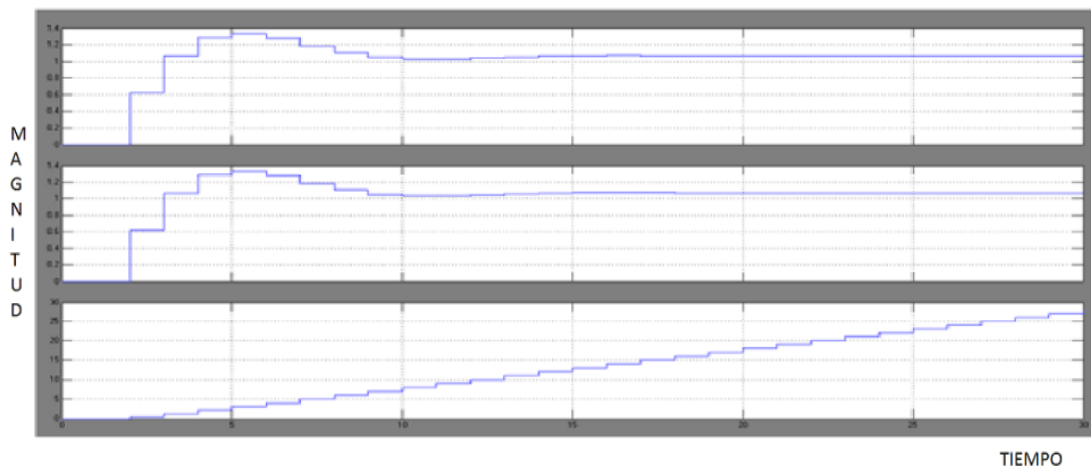


Figura 5. Resultados de la simulación del sistema.

En la parte superior de la Figura 5 se encuentra el comportamiento del sistema implementado, cuyos coeficientes se calcularon por medio del algoritmo DUPSO los cuales se encuentran en la Tabla 25. En la parte central de la Figura 5 se

encuentra el comportamiento al cual se quiere llevar al sistema. Por ultimo en la parte baja Figura 5 se encuentra el comportamiento de la planta. Como se puede observar, la planta tiene un comportamiento inestable, y luego de implementar el controlador propuesto por DUPSO, se logra seguir fielmente el modelo deseado.

6.3 ENFOQUE DE ECUACIONES POLINOMIALES PARA EL DISEÑO DEL SISTEMA DE REGULACIÓN Y CUYA SALIDA SIGA LA ENTRADA DE REFERENCIA

El ejemplo que se abordará en esta sección es tomado de [17] el cual muestra otra forma de realizar una acción de control.

Enunciado del problema:

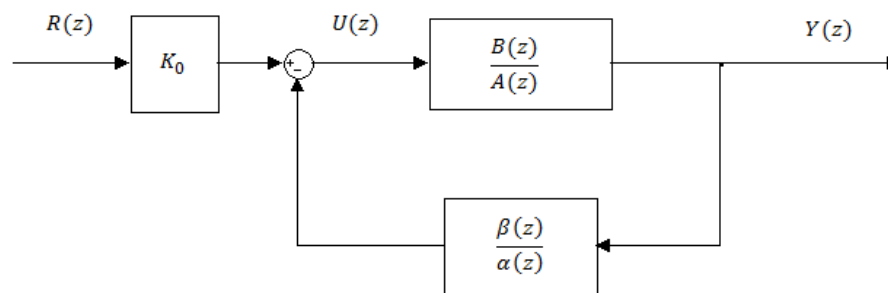


Figura 6. Esquema del ejercicio.

Se quiere diseñar un controlador el cual ubique los polos de lazo cerrado de la Figura 6 en:

$$z_1 = 0.6 + j0.4$$

(116)

$$z_2 = 0.6 - j0.4 \quad (117)$$

Donde la planta es:

$$\frac{B(z)}{A(z)} = \frac{0.02(z + 1)}{(z - 1)^2} \quad (118)$$

También incorporar un observador de orden mínimo para estimar las variables de estado para la realimentación y por último se quiere que la salida siga a la entrada.

El observador de orden mínimo tiene la ecuación del error de observación:

$$\phi(z) = z \quad (119)$$

El procedimiento para hallar este estimador se encuentra en [17] para más detalles.

Teniendo en cuenta el diagrama de bloques de la Figura 6, se obtiene la función de transferencia pulso que caracteriza al sistema.

$$\frac{Y(z)}{R(z)} = \frac{K_0 \left(\frac{B(z)}{A(z)} \right)}{\left(1 + \frac{(B(z)\beta(z))}{(A(z)\alpha(z))} \right)} \quad (120)$$

Reduciendo la ecuación (120) se tiene:

$$\frac{Y(z)}{R(z)} = \frac{K_0(\alpha(z)B(z))}{(\alpha(z)A(z) + \beta(z)B(z))} \quad (121)$$

Con los requerimientos del ejercicio, se dispone a desarrollar la ecuación diofántica siguiente:

$$(\alpha(z)A(z) + \beta(z)B(z)) = H(z)F(z) \quad (122)$$

Dónde:

$$H(z) = z^2 - 1.2z + 0.52 \quad (123)$$

La información de la ubicación de los polos esta en (123) y

$$F(z) = z \quad (124)$$

El polinomio de error del observador (124) tomado de [17].

Los polinomios $\alpha(z)$ y $\beta(z)$ tienen la forma:

$$\alpha(z) = X'_1 z + X'_2 \quad (125)$$

$$\beta(z) = X'_3 z + X'_4 \quad (126)$$

Reemplazando (118), (125) y (126) en la ecuación (122) se tiene la siguiente ecuación diofántica:

$$\begin{aligned} (X'_1 - 1)z^3 + (-2X'_1 + X'_2 + 0.02X'_3 + 1.2)z^2 + (X'_1 - 2X'_2 + 0.02X'_3 + 0.02X'_4 - 0.52)z \\ + (X'_2 + 0.02X'_4) = 0 \end{aligned} \quad (127)$$

Se dispone a utilizar el método propuesto para resolver la ecuación (127) y encontrar un par de soluciones que combinadas como numerador y denominador formarán los coeficientes de los polinomios de $\alpha(z)$ y $\beta(z)$.

En la Tabla 27, se encuentran los resultados del algoritmo implementado mostrando los pares numerador denominador y en la Tabla 28 se encuentran ilustrados los resultados de hacer la operación.

Tabla 27. Resultados de forma fraccionaria encontrados con DUPSO

prueba	Numeradores				Denominadores			
	X_1	X_2	X_3	X_4	X_5	X_6	X_7	X_8
1	6634	3167	-25176	25408	6634	9897	-1049	-1588
2	4147	2688	92352	23024	4147	8400	3848	-1439
3	10439	4256	54696	-99120	10439	13300	2279	6195
4	3610	3112	23208	-2528	3610	9725	967	158
5	5990	4192	-82728	-36096	5990	13100	-3447	2256
6	13061	744	16824	-34160	13061	2325	701	2135
7	4065	768	21528	32912	4065	2400	897	-2057
8	2945	4656	-162792	-154736	2945	14550	-6783	9671
9	5101	864	24144	354001	5101	2700	1006	-22125
10	9748	29210	11088	-12176	9748	91281	462	761
11	38447	3784	26952	-798596	38447	11825	1123	49912
12	11537	6144	56736	36880	11537	19200	2364	-2305
13	3169	5231	27072	26288	3169	16347	1128	-1643
14	4643	5944	30480	-207265	4643	18575	1270	12954
15	17235	2632	14328	-66208	17235	8225	597	4138
16	11524	14481	37512	-19072	11524	45253	1563	1192
17	1441	4449	19608	27472	1441	13903	817	-1717
18	5558	552	18504	-163776	5558	1725	771	10236
19	-13158	1200	179928	157135	-13158	3750	7497	-9821
20	4450	5208	27888	-16512	4450	16275	1162	1032

Dónde:

$$X'_1 = \frac{X_1}{X_5}$$

$$X'_2 = \frac{X_2}{X_6}$$

$$X'_3 = \frac{X_3}{X_7}$$

$$X'_4 = \frac{X_4}{X_7}$$

Tabla 28. Parámetros para el controlador encontrados DUPSO

PRUEBA	X'_1	X'_2	X'_3	X'_4	TIEMPO [s]	#ITERACIONES
1	1,0000	0,3200	24,0000	-16,0000	61,6602	1770
2	1,0000	0,3200	24,0000	-16,0000	46,6462	1451
3	1,0000	0,3200	24,0000	-16,0000	17,5360	562
4	1,0000	0,3200	24,0000	-16,0000	14,8843	783
5	1,0000	0,3200	24,0000	-16,0000	48,3610	1144
6	1,0000	0,3200	24,0000	-16,0000	106,6691	3468
7	1,0000	0,3200	24,0000	-16,0000	43,5376	1832
8	1,0000	0,3200	24,0000	-16,0000	91,8685	3194
9	1,0000	0,3200	24,0000	-16,0000	55,0848	1610
10	1,0000	0,3200	24,0000	-16,0000	19,5925	917
11	1,0000	0,3200	24,0000	-16,0001	44,7182	1341
12	1,0000	0,3200	24,0000	-16,0000	248,2407	4858
13	1,0000	0,3200	24,0000	-16,0000	123,5684	3208
14	1,0000	0,3200	24,0000	-16,0001	231,5956	4288
15	1,0000	0,3200	24,0000	-16,0000	13,8412	566
16	1,0000	0,3200	24,0000	-16,0000	44,0644	1397
17	1,0000	0,3200	24,0000	-16,0000	57,1086	1676
18	1,0000	0,3200	24,0000	-16,0000	29,4436	870
19	1,0000	0,3200	24,0000	-15,9999	14,3423	627
20	1,0000	0,3200	24,0000	-16,0000	86,0290	2324
PROMEDIO	1,0000	0,3200	24,0000	-16,0000	69,9396	1894
DESVIACION ESTANDAR	0	1,2973E-06	0	3,5752E-05	65,9121	1265

Como se puede observar de la Tabla 28 , la desviación estándar en el peor de los casos es del orden de 10^{-5} y como se vio en el ejercicio anterior, al implementar el sistema para seguir a un modelo, estos resultados no son perjudiciales para el

desarrollo del mismo. Nuevamente se incluye en la tabla un valor de exactitud por prueba, la cual muestra la minimización de la función objetivo y el valor mínimo que se obtuvo fue del orden de 10^{-12} .

Tomando los valores promedio de la Tabla 28, la función de transferencia del controlador queda de la siguiente forma

$$\frac{\beta(z)}{\alpha(z)} = \frac{(24z - 16.000005)}{(z - 0.31999991)} \quad (128)$$

Continuando con el desarrollo del ejercicio, se dispone a hallar el valor de ganancia que asegure que la salida siga la entrada de referencia escalón unitario. Para esto se hace que:

$$\lim_{k \rightarrow \infty} y(k) = \lim_{z \rightarrow 1} (1 - z)Y(z) = \lim_{k \rightarrow \infty} \frac{z - 1}{z} K_0 \frac{\alpha(z)B(z)}{H(z)F(z)} \frac{z}{z - 1} = K_0 \frac{\alpha(1)B(1)}{H(1)F(1)} = 1 \quad (129)$$

De (129) donde se obtiene:

$$K_0 = \frac{H(1)F(1)}{\alpha(1)B(1)} \quad (130)$$

Al reemplazar los valores se obtiene el valor de K_0 , el cual es:

$$K_0 = 6.0606$$

Los resultados serán validados mediante el software de simulación Simulink de Matlab, con el cual podremos observar de manera gráfica los resultados obtenidos.

La Figura 7 muestra el diagrama de bloques diseñado para efectos de simulación.

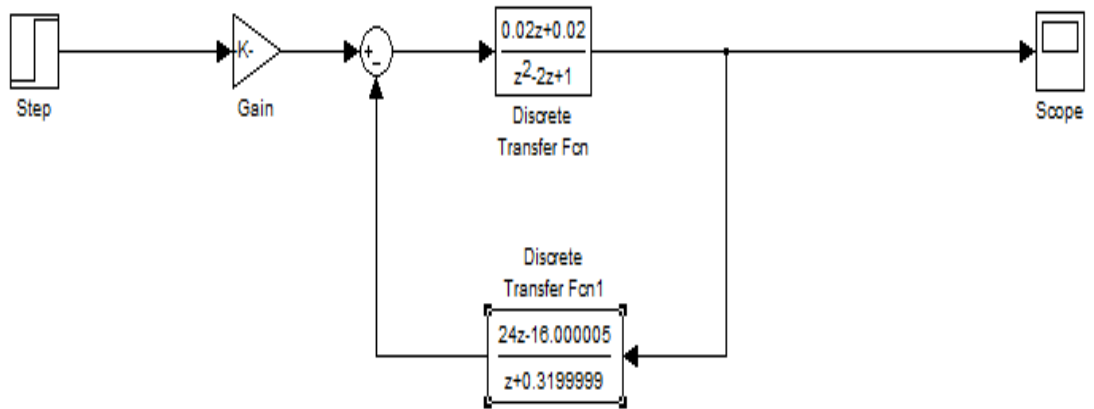


Figura 7. Diagrama en Simulink correspondiente al ejercicio.

Primero se tomara como entrada un escalón unitario, el resultado es mostrado en la Figura 8.

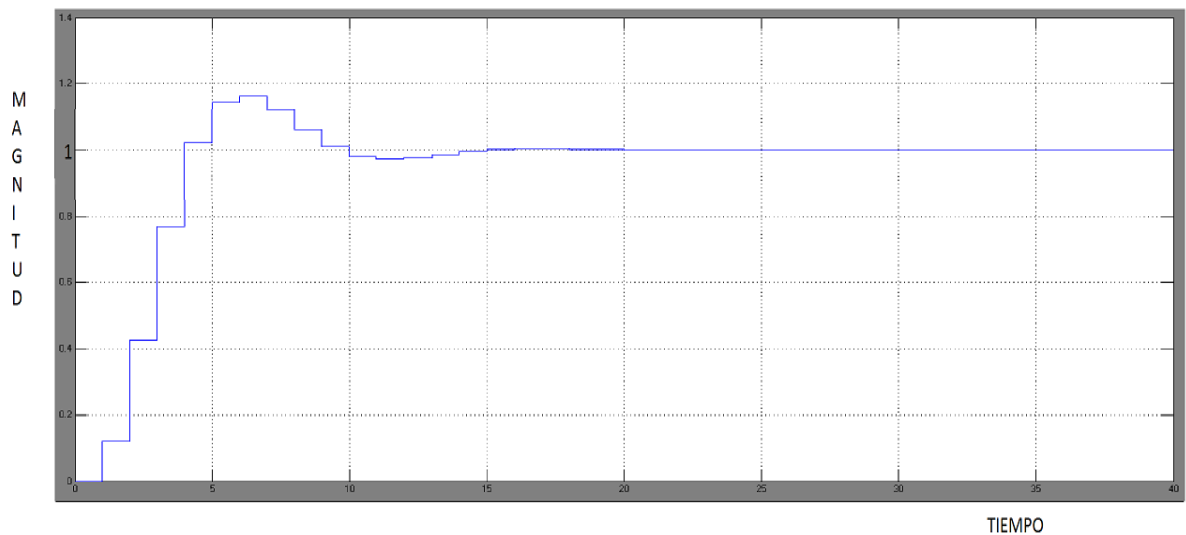


Figura 8. Respuesta al escalón unitario.

Como se puede observar, la salida tiene un comportamiento gobernado por los polos y la salida sigue fielmente la entrada. Ahora se tomará como entrada un escalón de amplitud tres, para poder observar si sigue el comportamiento. El resultado es mostrado en la Figura 9.

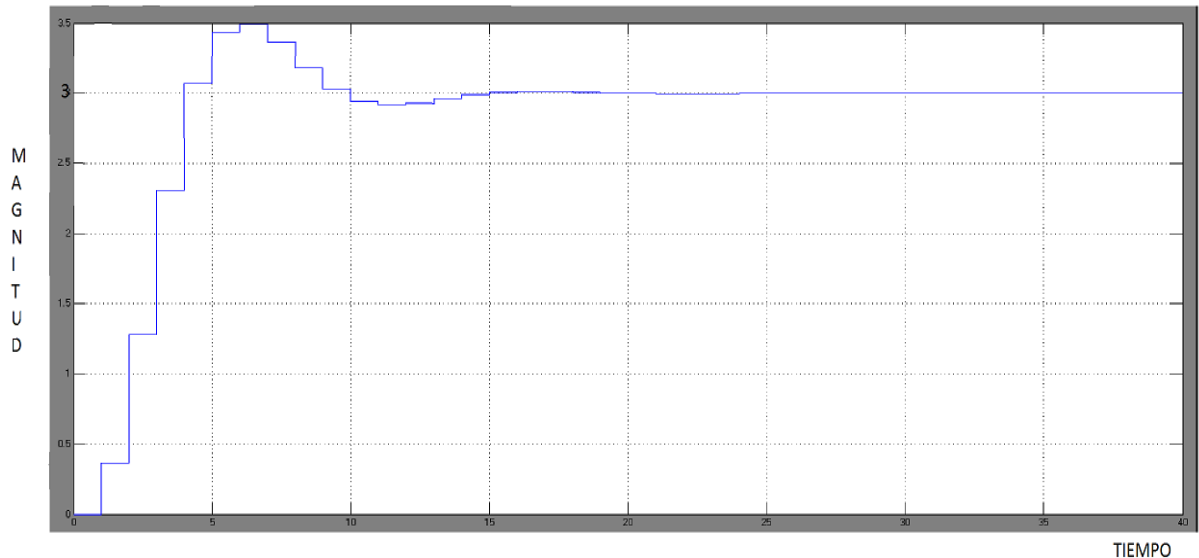


Figura 9. Respuesta al escalón de amplitud igual a 3

El resultado es el esperado, ya que la salida se estabiliza en 3.

El enfoque de ecuaciones polinomiales es un método alternativo al diseño mediante ubicación de polos donde su principal objetivo es resolver ecuaciones diofánticas.

7 PRUEBAS EN SISTEMAS DE ORDEN SUPERIOR

En este capítulo se presentan pruebas donde se somete el método a optimizar sistemas de orden superior. Se realizó una comparación con los métodos de optimización PSO y UPSO, así como con la función propia de optimización de Matlab *Fzeros*. Los resultados se mostrarán y se hará una breve discusión con los resultados obtenidos.

De la aplicación presentada en el capítulo 6 tenemos que en los sistemas de control la ecuación diofántica es definida como:

$$\alpha(s)A(s) + \beta(s)B(s) = D(s) \tag{131}$$

Donde los coeficientes del controlador son números reales.

Ahora se procede a implementar un algoritmo PSO y UPSO para darle solución a problemas reales en la rama de control, donde el sistema es de orden elevado. El sistema es el siguiente:

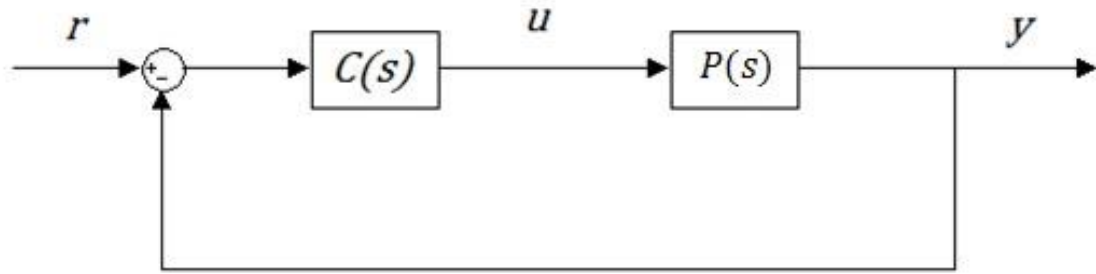


Figura 10. Sistema de control con realimentación unitaria.

Donde se propone que los polos del sistema en lazo cerrado sean de orden elevado. Como primera prueba se introducirán 6, luego 10 y finalmente 25 polos en -1 para ir incrementando la complejidad del sistema. Las pruebas realizadas tienen como objetivo mostrar que el algoritmo PSO converge en un sistema de orden elevado minimizando la función objetivo; el objetivo de estas pruebas no es implementar el controlador.

El orden de los polinomios $\alpha(s)$ y $\beta(s)$ deben ser seleccionados de manera tal que al hacer la multiplicación del polinomio $\alpha(s)$ con el polinomio $A(s)$ de como resultado el número de polos a ubicar.

Entonces la ecuación diofántica es:

$$\alpha(s)A(s) + \beta(s)B(s) = F(s) \quad (132)$$

Donde $F(s)$ contiene la información de los polos.

Se realiza la siguiente operación.

$$\alpha(s)A(s) + \beta(s)B - F(s) = 0 \quad (133)$$

Donde la función objetivo es la suma de cada uno de los coeficientes al cuadrado

$$f_{obj} = c_1^2 + c_2^2 + \dots + c_n^2 + c_{n+1}^2 \quad (134)$$

Donde n es el número de polos deseados.

Se realizó una comparación entre los métodos de UPSO y PSO continuos para el problema con diferente orden, incrementando el número de polos a ubicar.

Los resultados serán mostrados gráficamente, donde se mostrará la minimización de la función objetivo, respecto al número de iteraciones realizadas.

A continuación se presentan los algoritmos con los cuales se comparó el método implementado con los cuales se comparó el método en sistemas de orden superior.

7.1 ALGORITMO DE ENJAMBRE DE PARTÍCULAS UNIFICADO (UPSO)

En esta sección definimos el algoritmo de Enjambre de partículas unificado implementado para estas pruebas:

1. Definir las constantes del esquema de UPSO: $c_1, c_2, c_3, c_4, \chi, u$.
2. Definir los valores iniciales para la posición de las partículas y sus velocidades, los cuales serán aleatorios para la posición X_i^0 , y se partirá con velocidad inicial cero $V_i^0 = 0$.
3. Encontrar los valores iniciales de P_i y P_g evaluando cada partícula del enjambre en la función objetivo $f(X)$.

4. Se actualizan las velocidades local y global con las siguientes ecuaciones.

$$G_i^{t+1} = \chi[V_i^t + c_1rand_1(P_i - X_i) + c_2rand_2(P_g - X_i)] \quad (135)$$

$$L_i^{t+1} = \chi[V_i^t + c_3rand_3(P_i - X_i) + c_4rand_4(P_{gi} - X_i)] \quad (136)$$

5. Se actualiza la velocidad unificada de la siguiente forma:

$$U_i^{t+1} = uG_i^{t+1} + (1 - u)L_i^{t+1} \quad (137)$$

6. Se actualiza la partícula como sigue:

$$X_i^{t+1} = X_i^t + U_i^{t+1} \quad (138)$$

7. Evaluar la función objetivo $f(X)$ para cada partícula.
8. Seleccionar la mejor partícula del enjambre y actualizar la mejor posición local P_i .
9. Comparar la mejor posición de cada partícula con su posición actual, si su posición actual es mejor, actualizar P_{gi} .
10. Evaluar la mejor posición del enjambre en la función objetivo y comparar con P_g . Si ese valor es menor, entonces actualizar P_g .
11. Se actualiza la velocidad de PSO para ser cargada a las velocidades de UPSO.

$$V_i^{t+1} = \chi(V_i^t + c_1rand(P_i - X_i) + c_2rand(P_g - X_i)) \quad (139)$$

12. Preguntar si $f(P_g) = 0$. Si es si, parar, si no, volver al paso 4.

7.2 ALGORITMO DE ENJAMBRE DE PARTÍCULAS (PSO)

1. Definir las constantes del esquema de PSO: c_1, c_2, ω .
2. Definir los valores iniciales para la posición de las partículas y sus velocidades, los cuales serán aleatorios para la posición X_i^0 , y se partirá con velocidad inicial cero $V_i^0 = 0$.
3. Encontrar los valores iniciales de P_i y P_g evaluando cada partícula del enjambre en la función objetivo $f(X)$.
4. Se actualiza la partícula como sigue:

$$X_i^{t+1} = X_i^t + V_i^{t+1} \quad (140)$$

5. Evaluar la función objetivo $f(X)$ para cada partícula.
6. Seleccionar la mejor partícula del enjambre y actualizar la mejor posición local P_i .
7. Evaluar la mejor posición del enjambre en la función objetivo y comparar con P_g . Si ese valor es menor, entonces actualizar P_g .
8. Se actualiza la velocidad de PSO para ser cargada a las velocidades globales de la modificación.

$$V_i^{t+1} = \omega V_i^t + c_1 \text{rand}(P_i - X_i) + c_2 \text{rand}(P_g - X_i) \quad (141)$$

9. Preguntar si $f(P_g) \leq C_p$, donde C_p es el criterio de parada. Si cumple esta condición parar, en caso contrario, volver al paso 4.

7.3 CASO UNO: UBICACIÓN DE 6 POLOS EN -1

Sea la planta del sistema de la Figura 10:

$$P(s) = \frac{s^2 + s + 1}{s^3 + 3s^2 + 4s + 3} \quad (142)$$

Los polinomios del controlador $\alpha(s)$ y $\beta(s)$ para este caso particular son:

$$\alpha(s) = X_1s^3 + X_2s^2 + X_3s + X_4 \quad (143)$$

$$\beta(s) = X_5s^2 + X_6s + X_7 \quad (144)$$

Resultados con el método UPSO.

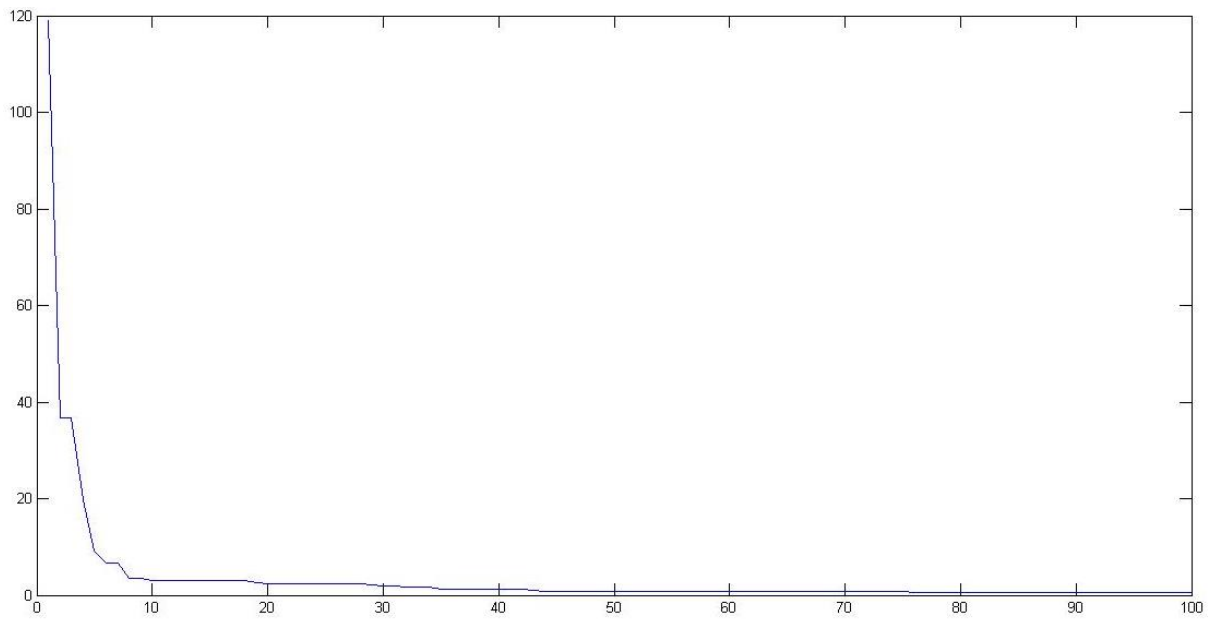


Figura 11. Función objetivo en el Intervalo entre 0 y 100 iteraciones con UPSO.

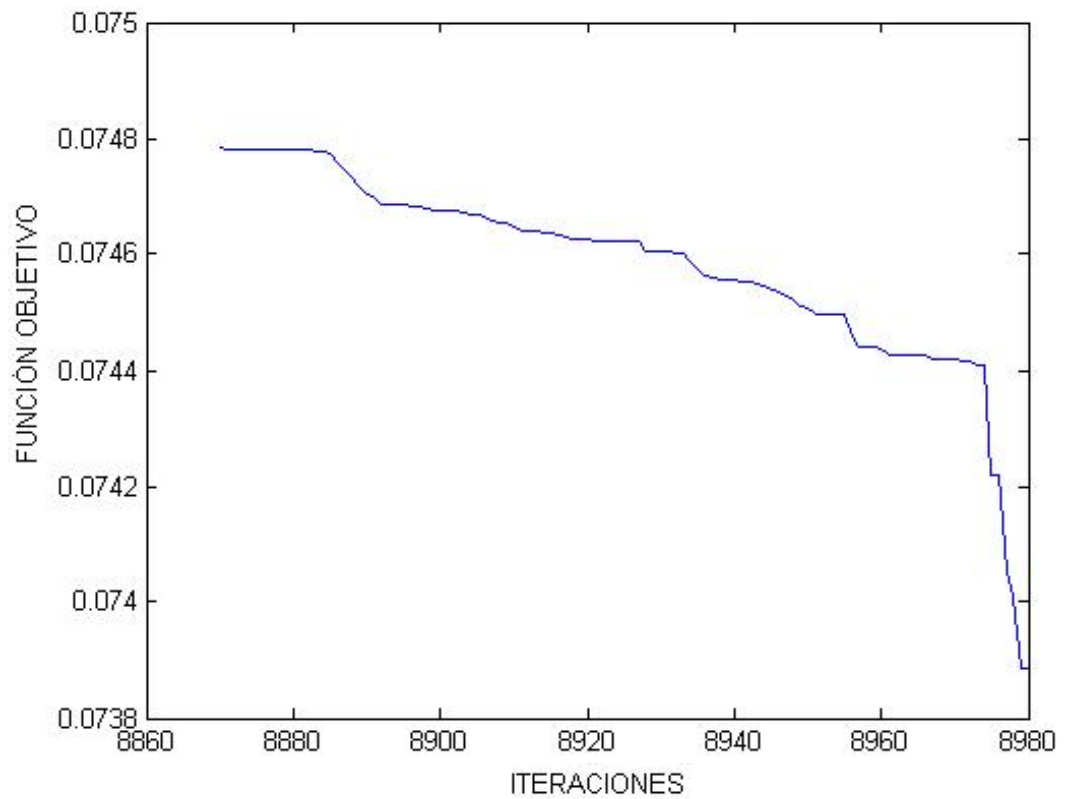


Figura 12. Función objetivo en el Intervalo final con UPSO.

Las gráficas muestran el comportamiento de la función objetivo. El método converge lentamente hacia la minimización de la función, lo cual no lo hace tan ventajoso implementarlo en un sistema real. Como se puede observar, el algoritmo tarda en disminuir la función objetivo de 0.0748 a 0.0739 más de 1000 iteraciones.

Resultados con el método PSO.

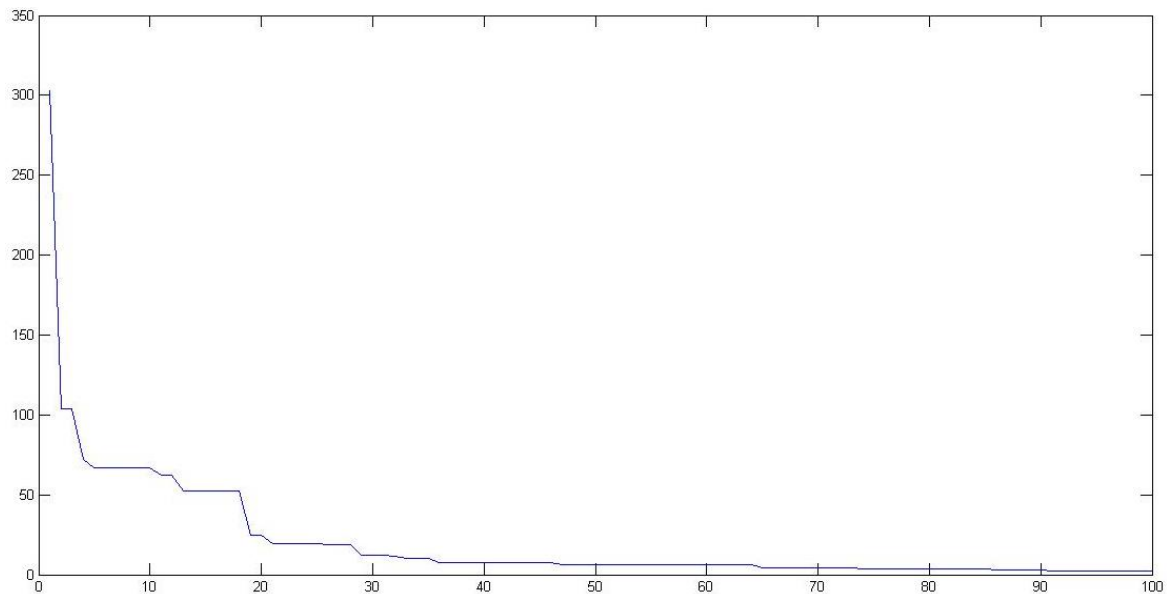


Figura 13. Valor de la función objetivo entre 0 y 100 iteraciones con PSO.

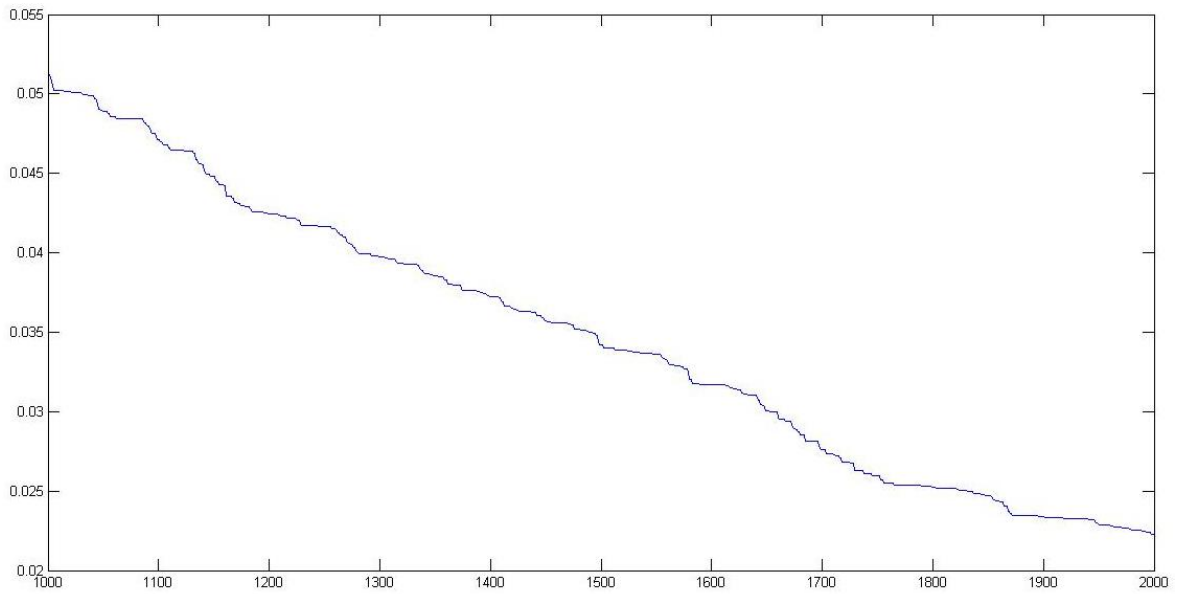


Figura 14. Valor de la función objetivo en el intervalo entre 1000 y 2000 iteraciones con PSO

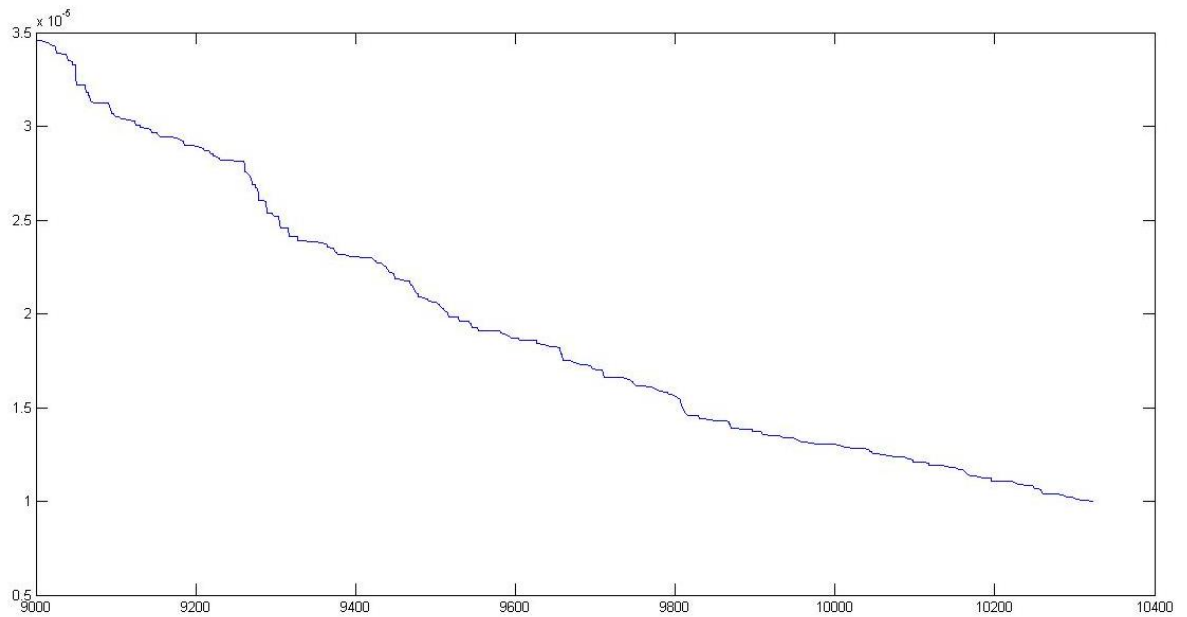


Figura 15. Valor de la función objetivo en el intervalo final de iteraciones con PSO.

Como se puede observar, el algoritmo tardó en minimizar la función objetivo de 0.05 a 0.025 en 1000 iteraciones, desde la iteración 1000 hasta la 2000 y desde la

iteración 9000 a la 10300, el algoritmo ha minimizado la función en un orden de 10^{-5} . Estos resultados son altamente satisfactorios para esta clase de sistemas.

7.4 SEGUNDO CASO: UBICACIÓN DE 10 POLOS EN -1

El controlador se define de la siguiente forma:

$$C(s) = \frac{x_9 s^2 + x_{10} s + x_{11}}{x_1 s^7 + x_2 s^6 + x_3 s^5 + x_4 s^4 + x_5 s^3 + x_6 s^2 + x_7 s + x_8} \quad (145)$$

El sistema a optimizar es de orden 11.

Resultados con el método UPSO.

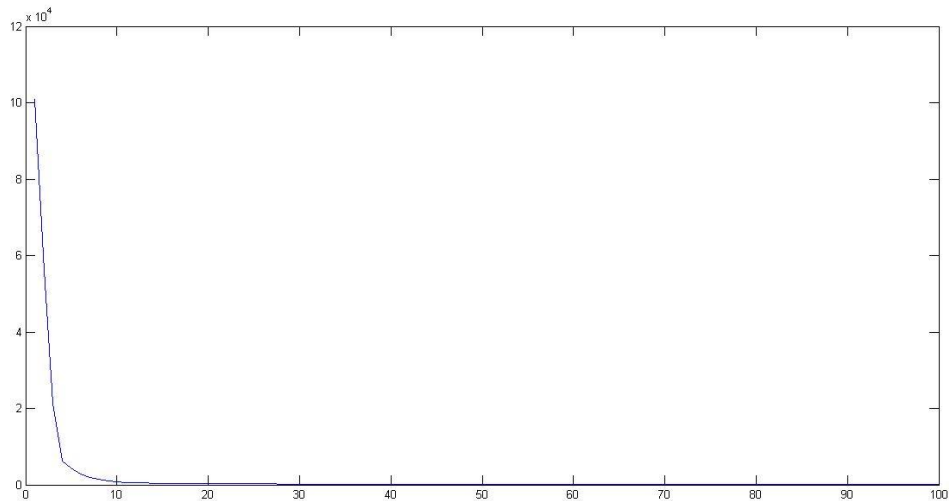


Figura 16. Valor de la función objetivo en el intervalo entre 0 y 100 iteraciones con UPSO

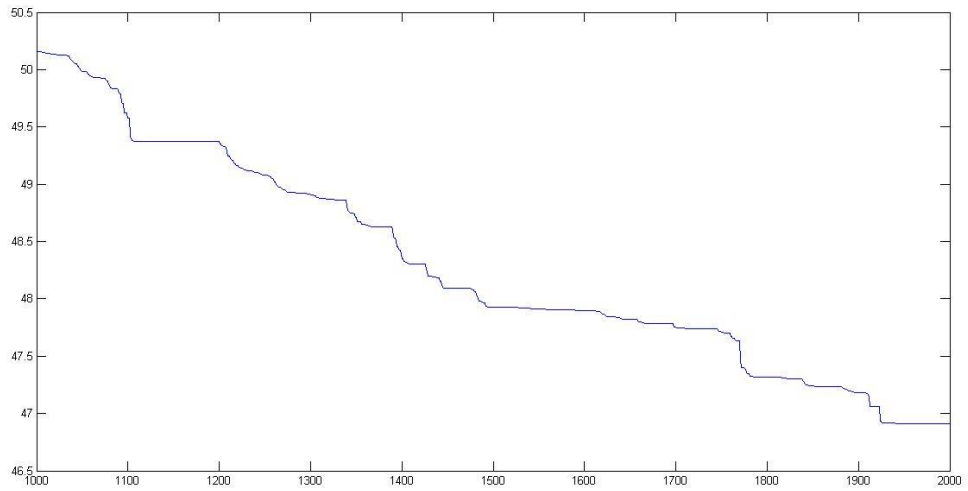


Figura 17. Valor de la función objetivo en el intervalo entre 1000 y 2000 iteraciones con UPSO

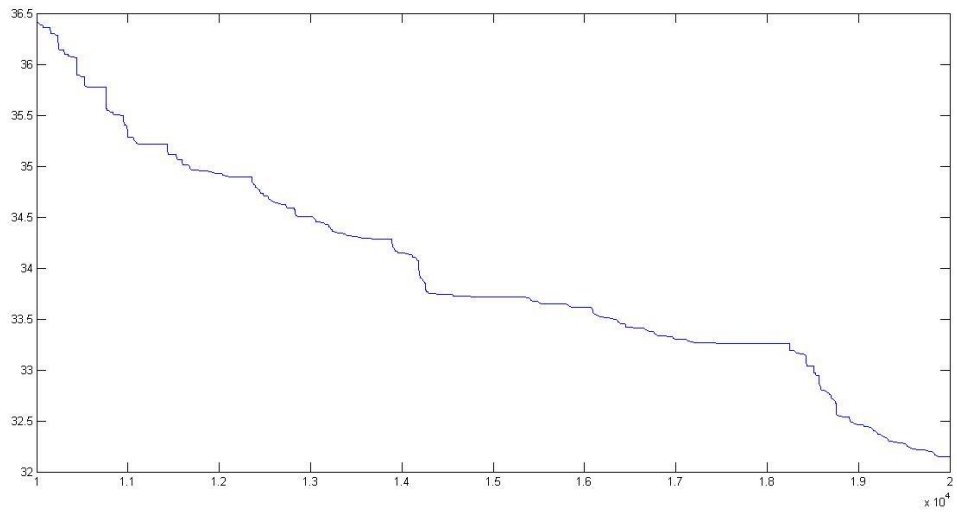


Figura 18. Valor de la función objetivo en el intervalo entre 10000 y 20000 iteraciones con UPSO

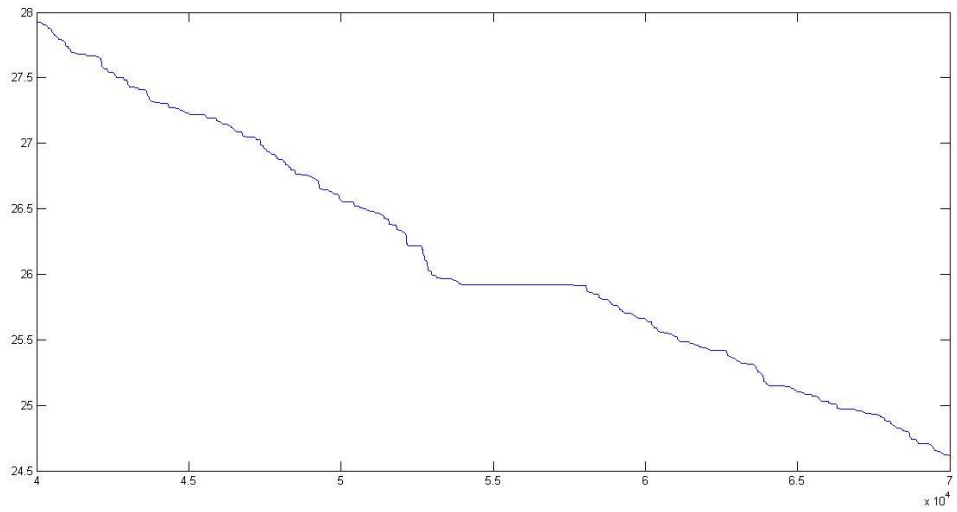


Figura 19. Valor de la función objetivo en el intervalo entre 40000 y 70000 iteraciones con UPSO

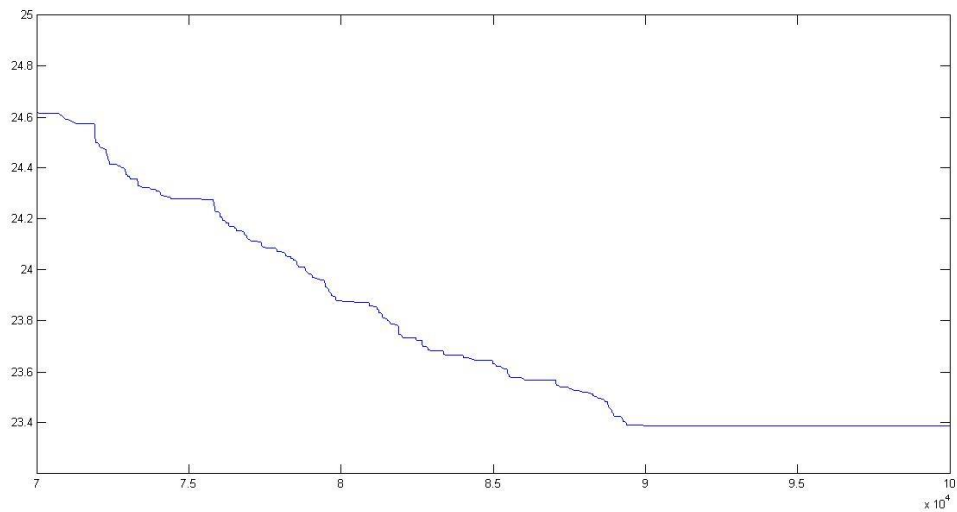


Figura 20. Valor de la función objetivo en el intervalo entre 70000 y 100000 iteraciones con UPSO

El algoritmo minimiza la función objetivo lentamente, llegando a un mínimo de 23.3 en 90000 iteraciones y 10000 iteraciones más, continúa en este valor. Este

resultado muestra que el algoritmo de UPSO falla en su afán de minimizar la función.

Resultados con el método PSO.

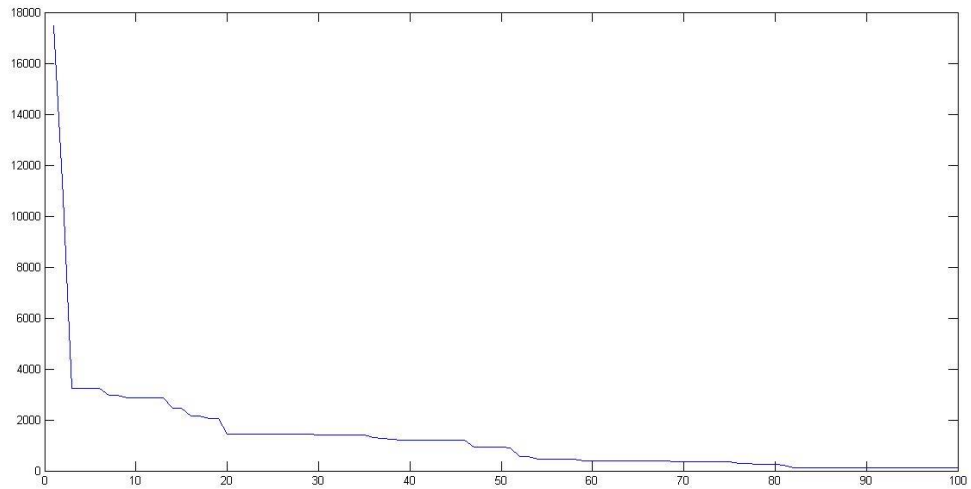


Figura 21. Valor de la función objetivo en el intervalo entre 0 y 100 iteraciones con PSO

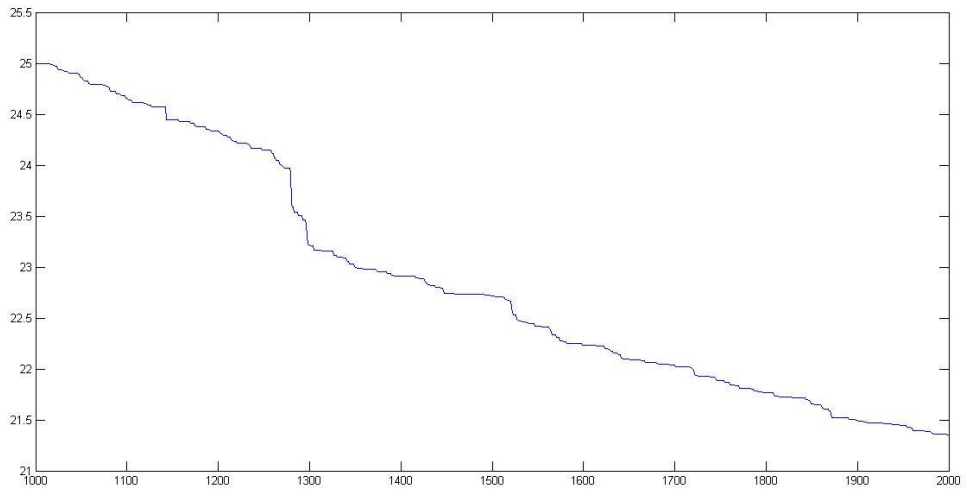


Figura 22. Valor de la función objetivo en el intervalo entre 1000 y 2000 iteraciones con PSO

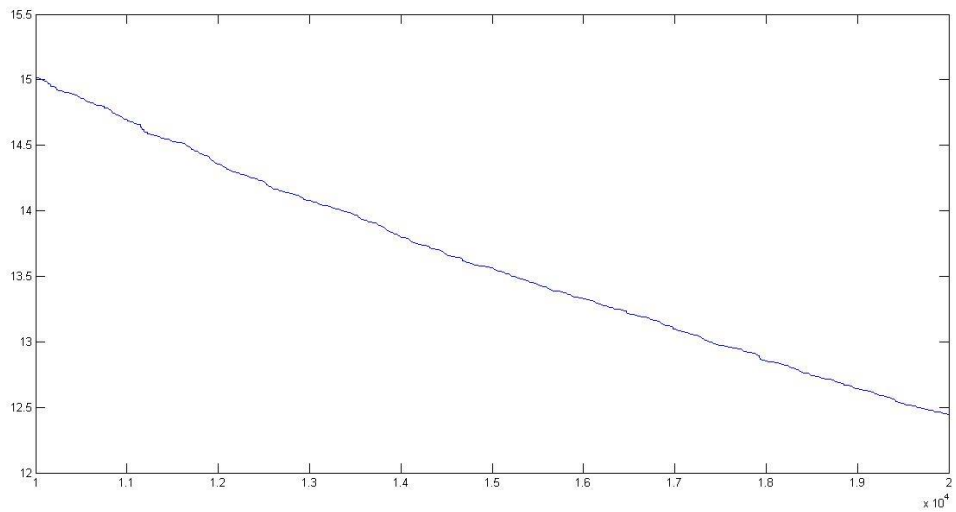


Figura 23. Valor de la función objetivo en el intervalo entre 10000 y 20000 iteraciones con PSO

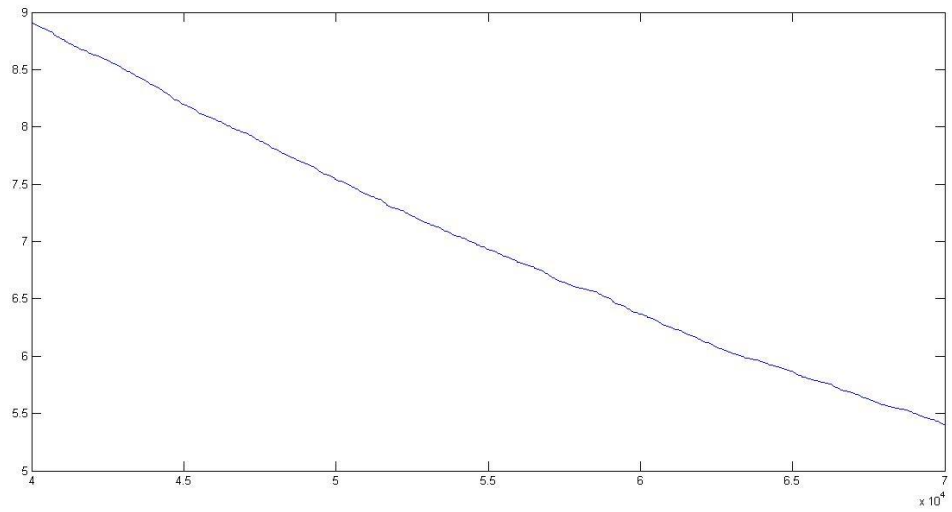


Figura 24. Valor de la función objetivo en el intervalo entre 40000 y 70000 iteraciones con PSO

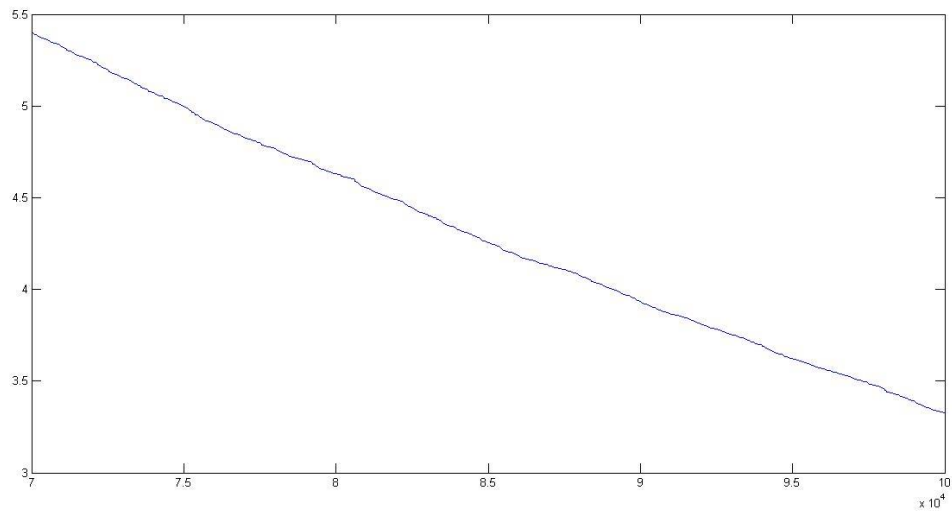


Figura 25. Valor de la función objetivo en el intervalo entre 70000 y 100000 iteraciones con PSO

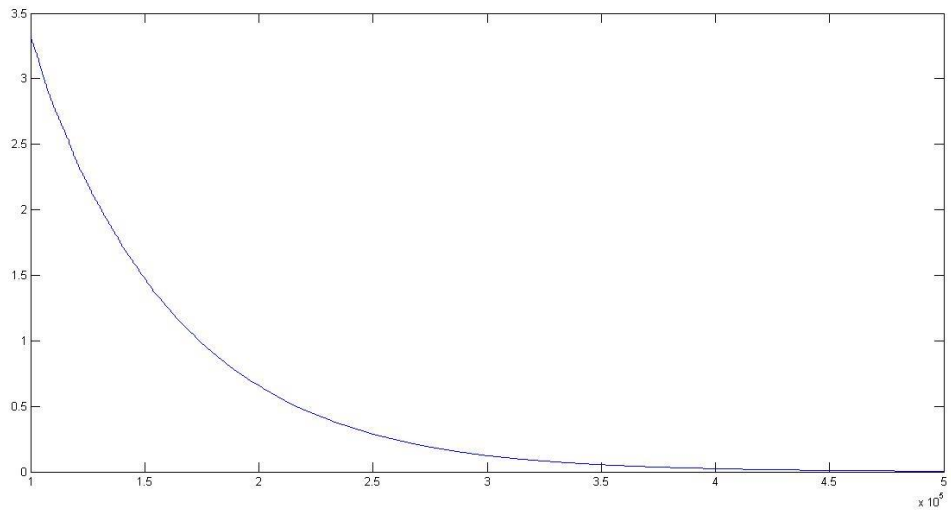


Figura 26. Valor de la función objetivo en el intervalo entre 100000 y 500000 iteraciones con UPSO

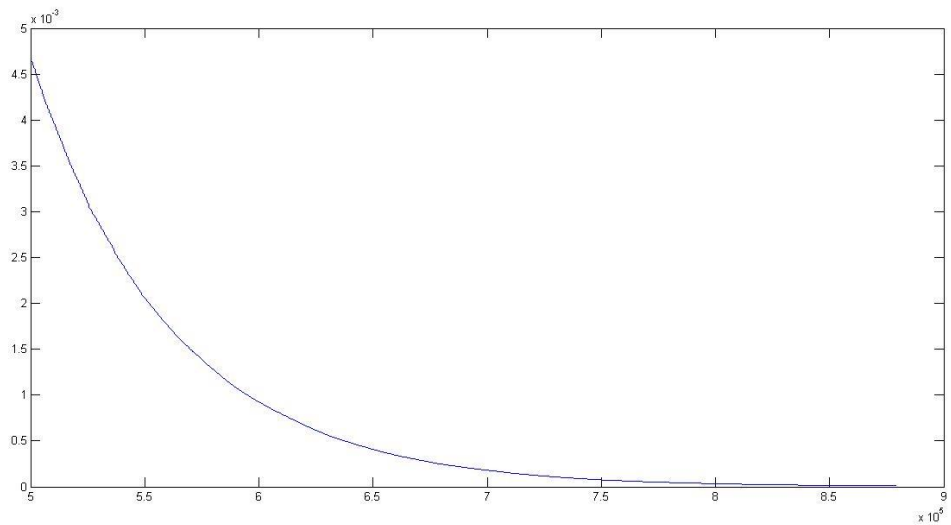


Figura 27. Valor de la función objetivo en el intervalo entre 500000 y 900000 iteraciones con PSO

Como se puede observar al seguir iterando con el algoritmo PSO, la función objetivo sigue disminuyendo hasta llegar a un criterio de parada elegido por el usuario o que el sistema requiera para su implementación.

7.5 TERCER CASO: UBICACIÓN DE 25 POLOS EN -1

Por último se pondrá a prueba el algoritmo con un sistema de orden 46 los cuales buscará los coeficientes de controlador, el cual tiene la siguiente forma:

$$C(s) = \frac{x_{24}s^{22} + x_{25}s^{21} + \dots + x_{44}s^2 + x_{45}s + x_{46}}{x_1s^{22} + x_2s^{21} + \dots + x_{21}s^2 + x_{22}s + x_{23}}$$

Resultados con el método UPSO.

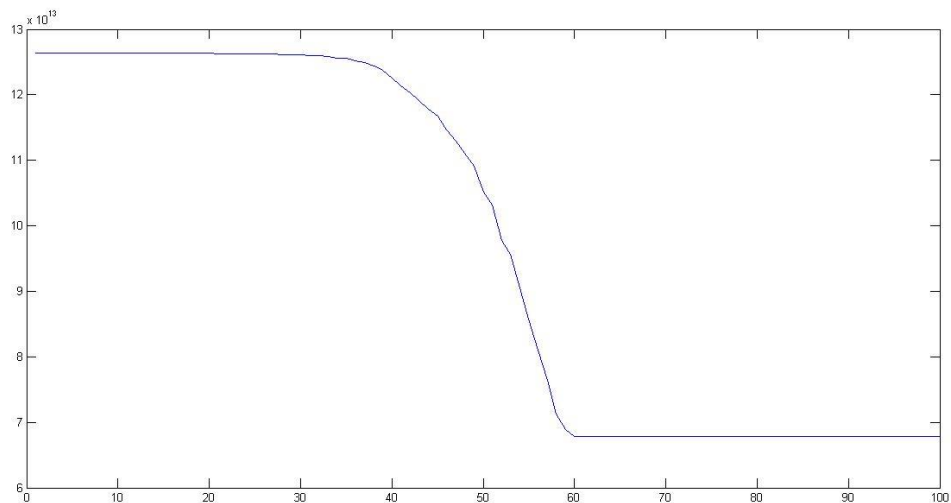


Figura 28. Valor de la función objetivo en el intervalo entre 0 y 100 iteraciones con UPSO

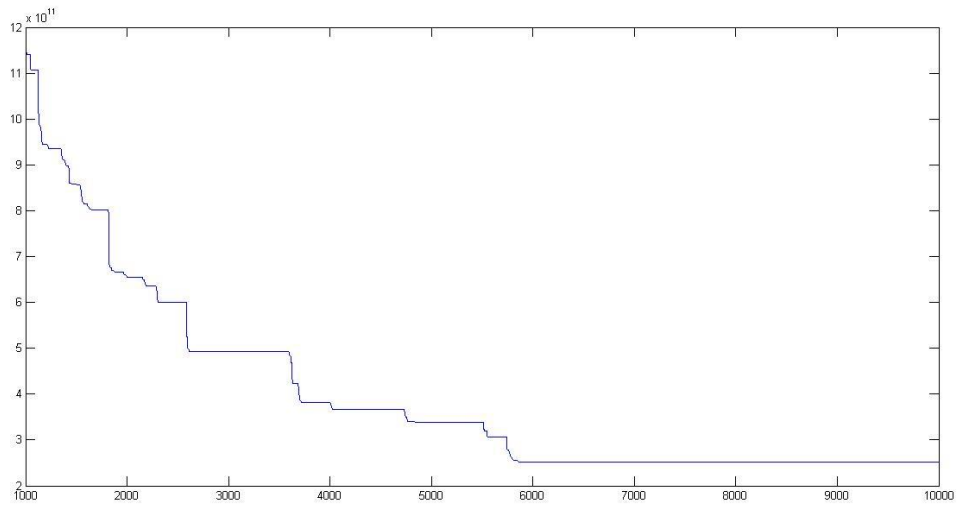


Figura 29. Valor de la función objetivo en el intervalo entre 1000 y 10000 iteraciones con UPSO

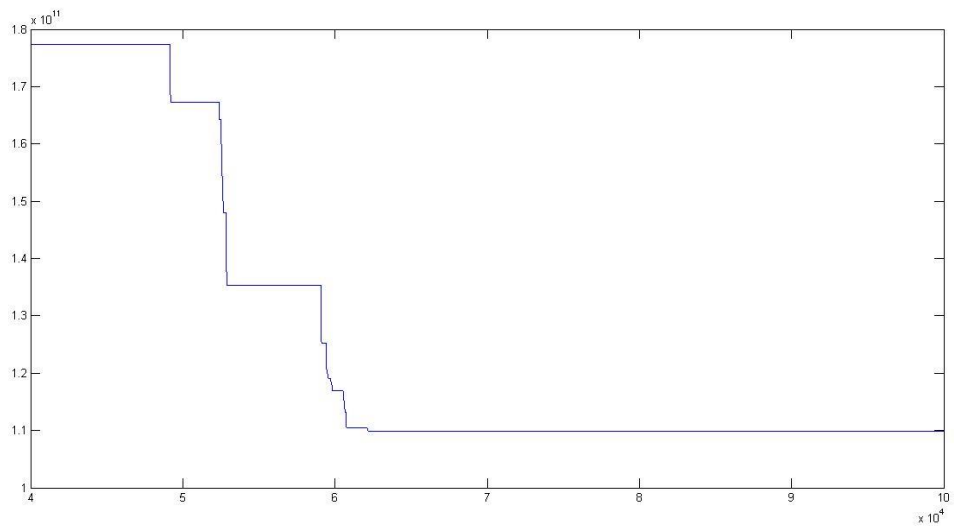


Figura 30. Valor de la función objetivo en el intervalo entre 40000 y 100000 iteraciones con UPSO

El algoritmo UPSO implementado a un sistema de orden superior, para este caso 46, es ineficiente, ya que se detiene por más de 50000 iteraciones sin mostrar una mejora considerable como se puede observar en la figura.

Resultados con el método PSO.

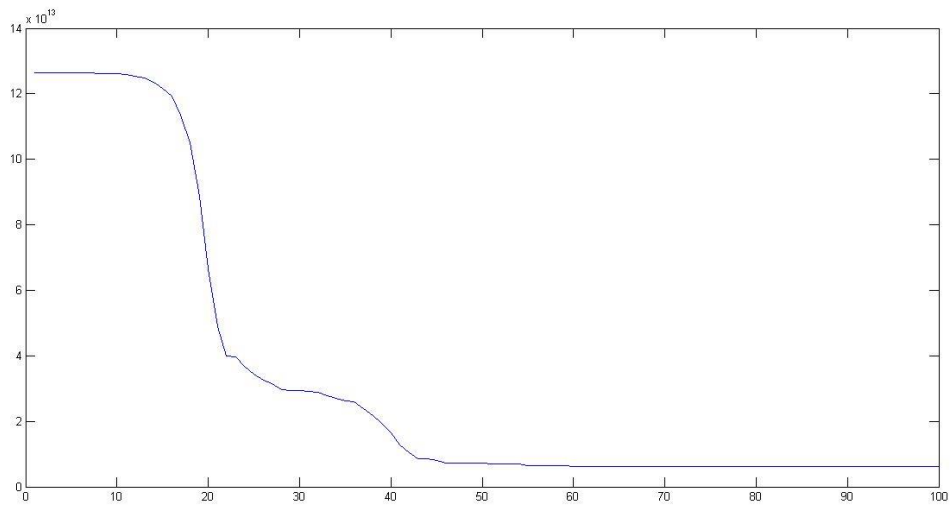


Figura 31. Valor de la función objetivo en el intervalo entre 0 y 100 iteraciones con PSO

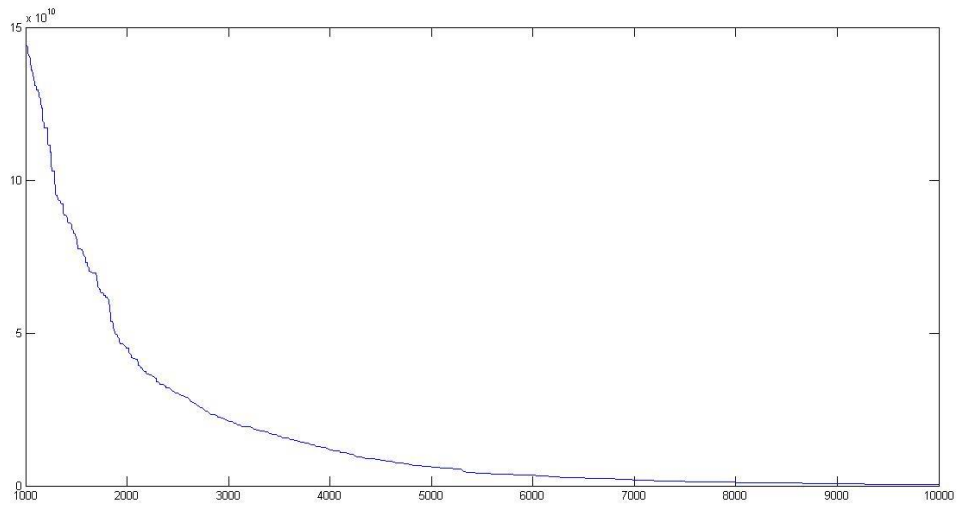


Figura 32. Valor de la función objetivo en el intervalo entre 1000 y 10000 iteraciones con PSO

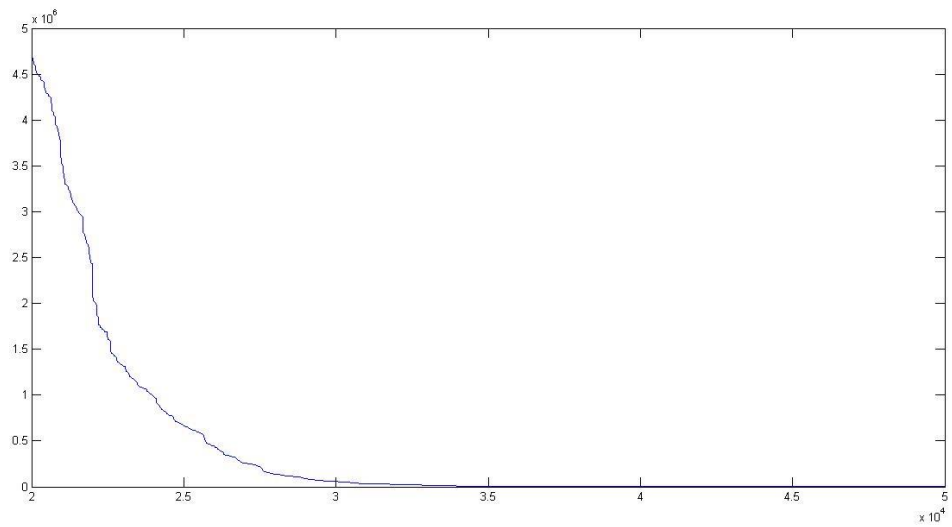


Figura 33. Valor de la función objetivo en el intervalo entre 20000 y 50000 iteraciones con PSO

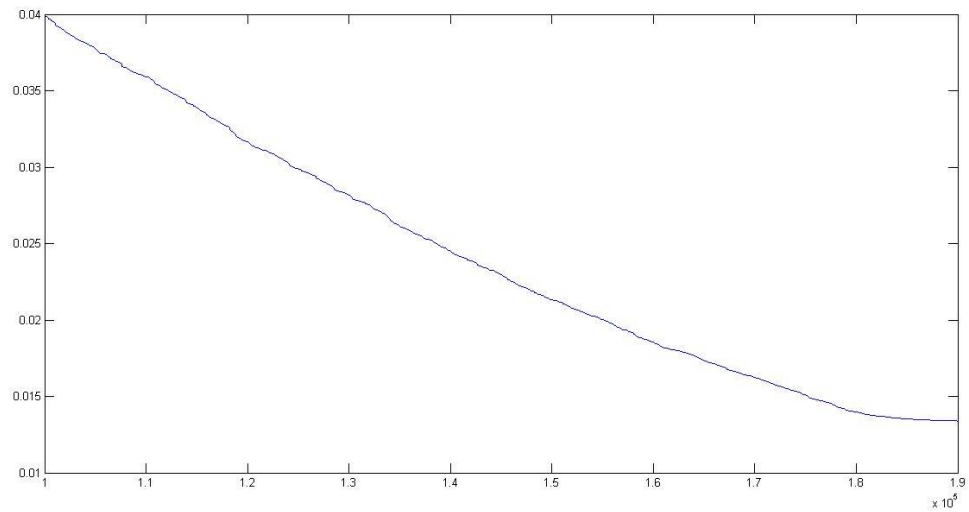


Figura 34. Valor de la función objetivo en el intervalo entre 100000 y 190000 iteraciones con PSO

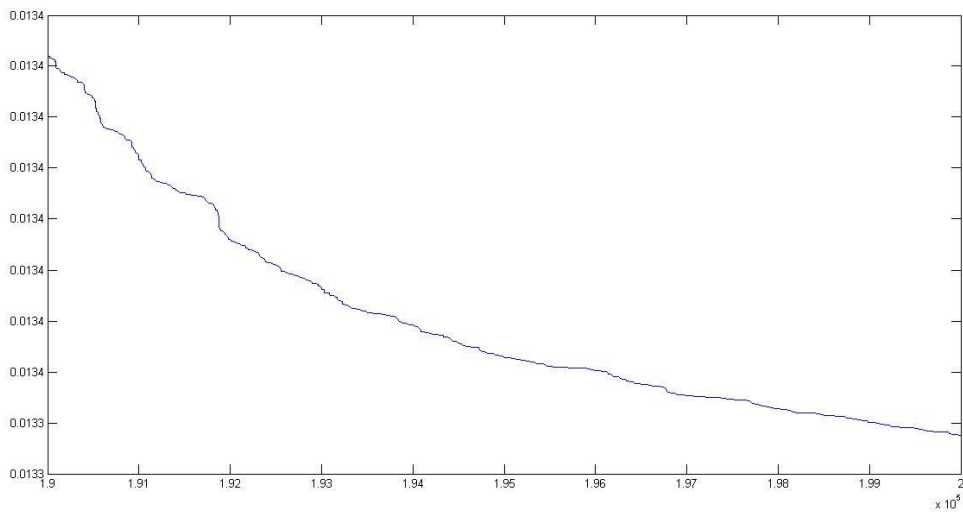


Figura 35. Valor de la función objetivo en el intervalo entre 190000 y 200000 iteraciones con PSO

Al transcurrir 200000 iteraciones el algoritmo sigue su búsqueda intensa de minimizar la función a pesar de las dimensiones del sistema y del tamaño de la

función, en este punto aunque las velocidades son pequeñas, sigue mejorando su valor de función objetivo.

7.6 FUNCIÓN EN MATLAB DE OPTIMIZACION FZERO.

Fzero es una función de MatLab el cual se fundamenta en minimizar una función objetivo, dando como resultado la solución del sistema. El sistema resuelto en el literal anterior, mediante PSO es introducido a MatLab y el resultado es el siguiente:

La función objetivo es minimizada hasta $2.4089 * 10^{+12}$. El anterior resultado puede ser debido a que la función es muy grande, y el método que tiene la función, se satura y no encuentra una mejora para minimizar la función y da escape a la función. PSO mostró que aunque el orden del sistema fuera alto, para este caso de 46, siempre encontraba un valor que podía reducir la función objetivo.

8 CONCLUSIONES

1. El método discreto implementado viaja en el dominio continuo, pero siempre está buscando una posición discreta.
2. El enjambre inicial de CPSO debe contener el dominio de solución, es decir, las partículas generadas aleatoriamente deben abordar un rango amplio y esto hace que el algoritmo emplee más tiempo de cómputo y de iteraciones, lo cual no lo hace tan ventajoso comparado con DUPSO, pues el enjambre puede ser creado en un lugar alejado de la respuesta y aun así, el algoritmo converge rápidamente.
3. Cuando las partículas están lejos de la respuesta en DUPSO, sus velocidades son altas, pero son controladas y el algoritmo converge.
4. En sistemas de orden superior, que presentan alta complejidad para ser resueltos de forma analítica, el algoritmo puede ser implementado como alternativa, sin la necesidad de técnicas matemáticas complejas y encontrando alta precisión en los resultados.
5. Un sistema de control polinomial no siempre posee coeficientes enteros, y al implementar el algoritmo propuesto, se obtuvieron resultados satisfactorios, ya que aunque la respuesta no es exacta, su desviación estándar es casi imperceptible, del orden de 10^{-5} , con lo cual se puede implementar el controlador con resultados alentadores.
6. Para sistemas de control es ineficiente trabajar con métodos discretos, ya que el algoritmo tiene que ser forzado a realizar acciones que no le corresponden.
7. El algoritmo PSO continuo, fue implementado para solucionar un sistema de orden 46 dando muy buenos resultados, ya que el algoritmo converge.
8. El ejemplo ilustrativo de la sección 6.1, no cumple con los requerimientos para realizar un sistema de control. Ese ejemplo es solo un caso particular donde

los coeficientes del controlador son enteros, pero visto desde un punto de vista de aplicación en la acción de control, no tiene ningún significado en esta rama.

9 RECOMENDACIONES

Analizar el comportamiento de otras metaheurísticas implementando la estrategia de optimización propuesta y realizar algoritmos híbridos que potencialicen la búsqueda.

El algoritmo de PSO puede ser implementado para hacer un control adaptativo, el cual pueda realizar la acción en tiempo real.

10 BIBLIOGRAFIA

- [1] B. Jarboui, N. Damak, P. Siarry, and a Rebai, "A combinatorial particle swarm optimization for solving multi-mode resource-constrained project scheduling problems," *Applied Mathematics and Computation*, vol. 195, no. 1, pp. 299–308, Jan. 2008.
- [2] K. Parsopoulos and M. Vrahatis, "Unified particle swarm optimization for solving constrained engineering optimization problems," *Lecture Notes in Computer Science (LNCS)*, vol. 3612, pp. 582–591, 2005.
- [3] S. Wu, "Time - Varying Feedback Systems Design Via Diophantine Equation Order Reduction," *The University Of Texas at Arlington*, no. December, 2007.
- [4] M. Cipu, "Groebner Bases and Solutions to Diophantine Equations," *2008 10th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing*, no. 3, pp. 77–80, 2008.
- [5] K. Plarre, A. Rojas, and R. Rojas, "Analytical solution to the polynomial diophantine equation: development and application to generalised predictive control analysis," *Control'98. UKACC*, no. 455, pp. 1–4, 1998.
- [6] S. N. Deepa and G. Sugumaran, "Model order formulation of a multivariable discrete system using a modified particle swarm optimization approach," *Swarm and Evolutionary Computation*, vol. 1, no. 4, pp. 204–212, Dec. 2011.
- [7] I. Amaya, J. Cruz, and R. Correa, "Real Roots of nonlinear sustems of equations through a metaheuristic algorithm," *Revista Dyna*, vol. 78, no. 170, pp. 15–23, 2011.
- [8] J. Cruz, "Solución del Modelo Matematico de un Circuito Electrónico D.C. No Limeal Mediante Una Estrateguia de Optimizacion," Tesis de Grado, Universidad Industrial de santander, 2012.
- [9] K. Parsopoulos and M. Vrahatis, "Studying the performance of unified particle swarm optimization on the single machine total weighted tardiness problem," *AI 2006: Advances in Artificial Intelligence*, vol. 4304, no. 1, pp. 760–769, 2006.

- [10] H. Zhu, Y. Wang, K. Wang, and Y. Chen, "Particle Swarm Optimization (PSO) for the constrained portfolio optimization problem," *Expert Systems with Applications*, vol. 38, no. 8, pp. 10161–10169, Aug. 2011.
- [11] H.-T. Hsieh and C.-H. Chu, "Improving optimization of tool path planning in 5-axis flank milling using advanced PSO algorithms," *Robotics and Computer-Integrated Manufacturing*, pp. 1–8, May 2012.
- [12] K. Parsopoulos and M. Vrahatis, "Multi-Objective Particles Swarm Optimization Approaches," *Multi-Objective Particle Swarm Optimization Approaches*, pp. 20–42, 2008.
- [13] K. E. Parsopoulos and M. N. Vrahatis, "Unified Particle Swarm Optimization in Dynamic Environments," *Lecture Notes in Computer Science (LNCS)*, vol. 3449, pp. 590– 599, 2005.
- [14] K. E. Parsopoulos and M. N. Vrahatis, "UNIFIED PARTICLE SWARM OPTIMIZATION FOR TACKLING OPERATIONS RESEARCH PROBLEMS," 2005.
- [15] K. E. Parsopoulos and M. N. Vrahatis, "Parameter selection and adaptation in Unified Particle Swarm Optimization," *Mathematical and Computer Modelling*, vol. 46, no. 1–2, pp. 198–213, Jul. 2007.
- [16] C.-J. Liao and P. Luarn, "A discrete version of particle swarm optimization for flowshop scheduling problems," *Computers & Operations Research*, vol. 34, no. 10, pp. 3099–3111, Oct. 2007.
- [17] K. Ogata, *Sistemas de Control en Tiempo Discreto*, PRENTICE H. 1996, pp. 517–565.
- [18] A. Cervantes, "Clasificación mediante enjambre de prototipos," Tesis de Doctorado, Universidad Carlos III de Madrid, 2009.
- [19] Y.-X. Jin, H.-Z. Cheng, J. Yan, and L. Zhang, "New discrete method for particle swarm optimization and its application in transmission network expansion planning," *Electric Power Systems Research*, vol. 77, no. 3–4, pp. 227–233, Mar. 2007.
- [20] H. Shayeghi, M. Mahdavi, and a. Bagheri, "Discrete PSO algorithm based optimization of transmission lines loading in TNEP problem," *Energy Conversion and Management*, vol. 51, no. 1, pp. 112–121, Jan. 2010.

- [21] J. Cruz, I. Amaya, and R. Correa, "Solution of the Mathematical Model of a DC Nonlinear Electronic Circuit Using an Optimization Strategy," *Circuits and Systems (CWCAS), 2012 IEEE 4th Colombian Workshop on*, pp. 1–6, 2012.
- [22] I. Amaya, J. Cruz, and R. Correa, "Aplicación del algoritmo determinístico de fuerza central en el diseño de un reductor de velocidad de Golinski óptimo," 2011.
- [23] F. García, "Un Pequeño Manual Para La Resolución de Problemas," *España*, pp. 7–8, 2002.
- [24] C. Gomez and O. Pérez, "Implementación de un Algoritmo de Optimización Para Sistemas Discretos Fundamentado en la Técnica de Enjambre de Partículas," Tesis de Grado, Universidad Industrial de Santander, 2012.
- [25] S. Reyes and J. Plata, "CÁLCULO DE RAÍCES REALES Y/O COMPLEJAS DE SISTEMAS DE ECUACIONES NO LINEALES MEDIANTE EL MÉTODO DE ENJAMBRE DE PARTÍCULAS," Universidad Industrial de Santander, 2011.
- [26] S. Grossman, *Algebra lineal*, 5th ed. Mexico: MCGRAW-HILL, 2007.
- [27] A. Figueroa, "Ecuaciones Diofánticas y Engranajes Cilíndricos," *Sociedad Colombiana de Matemáticas XV Congreso Nacional de Matemáticas*, vol. Especial, pp. 285–298, 2006.

11 ANEXOS

ANEXO A: TABLAS DE RESULTADOS CON OTROS METODOS PARA EL EJERCICIO DE LA COMPRA DE CELULARES

Para el ejercicio de la sección 5.1 a continuación se presenta los resultados encontrados al implementar la función objetivo con los métodos antes mencionados:

Tabla 29. Resultados completos del ejercicio 5.1 con otros métodos de optimización

PRUEBA	PSO		UPSO		CPSO	
	TIEMPO	ITERACIONES	TIEMPO	ITERACIONES	TIEMPO	ITERACIONES
1	3,1425	815	0,3383	170	0,1500	76
2	2,5440	690	0,4221	214	6,7640	3997
3	2,0519	558	0,4147	201	0,4190	242
4	2,4347	668	0,3545	177	5,3000	3082
5	2,3737	650	0,3877	194	0,4550	252
6	2,4710	674	0,3396	171	4,4850	7,611
7	2,1269	585	0,3775	192	9,6920	5742
8	3,0040	807	0,4442	227	3,2460	1931
9	1,8138	965	0,2372	226	3,3420	3791
10	1,9713	1036	0,2975	280	3,1630	3551
PROMEDIO	2,39338	745	0,36133	205	3,7016	2267

Los parámetros del algoritmo (c_1, c_2 y X) son iguales para todos los métodos implementados.

ANEXO B: TABLAS DE RESULTADOS PARA EL EJERCICIO DE LOS NUMEROS CONSECUTIVOS

Para el ejercicio de la sección 5.2 se presentan los resultados encontrados al implementar la función objetivo con los métodos antes mencionados:

Tabla 30. Resultados completos del ejercicio 5.2 con otros métodos de optimización

PRUEBA	PSO		UPSO		CPSO	
	TIEMPO	#ITERACIONES	TIEMPO	#ITERACIONES	TIEMPO	#ITERACIONES
1	3,3986	175	2,0776	135	103,3990	29923
2	3,1835	171	1,2357	133	112,0790	29425
3	5,0694	505	1,2857	143	2,3440	354
4	3,4748	168	3,8676	217	2,1440	304
5	10,9345	552	1,6692	125	32,3890	8732
6	16,1316	617	2,0829	141	11,7260	1809
7	10,8163	616	3,5036	228	105,7090	30600
8	6,6766	822	1,7349	127	161,9830	33585
9	3,9044	163	3,3937	232	115,3220	24680
10	6,8910	379	2,6918	151	175,5120	54608
PROMEDIO	82,2607	21402	7,0481	417	2,3543	163

ANEXO C: TABLAS DE RESULTADOS DE CPSO PARA EL EJERCICIO DE LA TRIPLA PITAGÓRICA

Tabla 31. Resultados con CPSO para el ejercicio de la tripla pitagórica

PRUEBA	X_1	X_2	X_3	TIEMPO [s]	#ITERACIONES
1	5	12	13	5,58	1295
2	12	5	13	6,822	1613
3	6	8	10	7,857	1833
4	5	12	13	17,400	1611
5	8	6	10	16,301	1734
6	6	8	10	12,509	1353
7	12	5	13	18,434	1907
8	5	12	13	13,150	1931
9	8	6	10	1,371	237
10	6	8	10	6,284	1513
11	5	12	13	10,578	2544
12	12	5	13	28,067	6696
13	5	12	13	21,56	5153
14	8	6	10	43,418	10370
15	6	8	10	5,442	1295
16	5	12	13	11,227	2547
17	12	5	13	7,554	1797
18	8	6	10	11,519	2625
19	5	12	13	5,788	1349
20	6	8	10	25,334	5783
PROMEDIO				13,80975	2760

ANEXO D: TABLAS DE RESULTADOS DE CPSO PARA EL EJERCICIO DE LA COMPRA DE CELULARES.

Tabla 32. Resultados con el algoritmo CPSO para la compra de celulares

PRUEBA	X_1	X_2	X_3	TIEMPO [s]	#ITERACIONES
1	10	2	75	41,675	5556
2	8	4	80	335,465	48602
3	4	8	90	913,502	99265
4	6	6	85	50,035	5503
5	6	6	85	671,517	85912
6	8	4	80	474,738	99265
7	6	6	85	21,18	5503
8	10	2	75	448,767	85912
9	10	2	75	538,737	103767
10	6	6	85	1581,851	305677
11	8	4	80	39,405	8661
12	10	2	75	111,967	18867
13	10	2	75	507,27	94509
14	8	4	80	116,874	21137
15	4	8	90	230,846	42821
16	8	4	80	162,914	29380
17	10	2	75	515,531	99265
18	4	8	90	27,842	5503
19	10	2	75	424,372	85912
20	10	2	75	41,675	5556
PROMEDIO				543,376	103767

ANEXO E: RESULTADOS CON EL ALGORITMO CPSO PARA EL PROBLEMA DEL PROCESO DE FABRICACIÓN

Tabla 33. Resultados con el algoritmo CPSO para el problema del proceso de fabricación

PRUEBA	u	v	w	x	y	z	TIEMPO [s]	# ITERACIONES
1	6	3	8	1	12	3	32,991	2561
2	6	3	8	1	12	3	6,344	478
3	6	3	8	1	12	3	74,195	5184
4	6	3	8	1	12	3	10,184	930
5	6	3	8	1	12	3	152,565	12812
6	6	3	8	1	12	3	105,367	8901
7	6	3	8	1	12	3	123,956	10881
8	6	3	8	1	12	3	2,61	207
9	6	3	8	1	12	3	10,746	673
10	6	3	8	1	12	3	52,872	3326
11	6	3	8	1	12	3	19,71	1629
12	6	3	8	1	12	3	3,897	351
13	6	3	8	1	12	3	3,552	294
14	6	3	8	1	12	3	72,731	6100
15	6	3	8	1	12	3	237,33	15793
16	6	3	8	1	12	3	89,83	5752
17	6	3	8	1	12	3	93,029	6088
18	6	3	8	1	12	3	8,512	578
19	6	3	8	1	12	3	137,547	8877
20	6	3	8	1	12	3	54,239	3668
PROMEDIO							64,61035	4755

**ANEXO F: TABLA DE RESULTADOS CON EL ALGORITMO DE CPSO PARA
EL EJERCICIO DEL RESTAURANTE.**

Tabla 34. Resultados con CPSO para el ejercicio del restaurante.

PRUEBA	X	Y	Z	TIEMPO [s]	#ITERACIONES
1	12	16	19	1,7910	620
2	12	16	19	97,6290	33199
3	12	16	19	0,2880	95
4	12	16	19	128,9820	34971
5	12	16	19	1,1940	331
6	12	16	19	51,0880	15344
7	12	16	19	0,4060	136
8	12	16	19	345,2070	93322
9	12	16	19	1,1770	419
10	12	16	19	26,6370	8837
11	12	16	19	8,4400	2571
12	12	16	19	59,3840	17561
13	12	16	19	20,8040	5530
14	12	16	19	1,7690	624
15	12	16	19	296,3860	74513
16	12	16	19	1,4310	384
17	12	16	19	46,6250	12367
18	12	16	19	108,0770	26210
19	12	16	19	145,1470	39431
20	12	16	19	0,8110	252
PROMEDIO				67,1637	18336

**ANEXO G: TABLA DE RESULTADOS CON CPSO PARA EL EJERCICIO DEL
SASTRE**

Tabla 35. Resultados con CPSO para el ejercicio del sastre

PRUEBA	x	y	TIEMPO [s]	# ITERACIONES
1	77	27	29,541	9626
2	77	27	35,034	11519
3	77	27	26,034	9476
4	77	27	3,15	1042
5	77	27	12,698	4247
6	77	27	20,811	6857
7	77	27	10,025	3378
8	77	27	4,903	1680
9	77	27	9,387	2990
10	40	40	4529	13,395
11	40	40	10,019	3268
12	3	53	20,442	6753
13	3	53	7,699	2588
14	40	40	8,458	2842
15	77	27	27,376	9225
16	3	53	9,561	3253
17	77	27	10,404	3353
18	3	53	10,904	3607
19	77	27	28,742	9649
20	77	27	8,432	2862
PROMEDIO			241,131	4911

ANEXO H: RESULTADOS CON CPSO PARA EL EJERCICIO DEL GRANJERO Y SUS ANIMALES

Tabla 36. Resultados del ejercicio del granjero implementado con el algoritmo CPSO

PRUEBA	X	Y	Z	TIEMPO [s]	#ITERACIONES
1	80	1	19	128,31	47970
2	80	1	19	173,188	59806
3	80	1	19	88,201	29928
4	80	1	19	67,427	24084
5	80	1	19	1,056	375
6	80	1	19	92,959	33688
7	80	1	19	46,561	16068
8	80	1	19	293,732	105642
9	80	1	19	93,674	32853
10	80	1	19	1,744	675
11	80	1	19	0,827	288
12	80	1	19	0,512	196
13	80	1	19	100,565	34810
14	80	1	19	0,843	322
15	80	1	19	0,625	236
16	80	1	19	1,047	379
17	80	1	19	32,945	11642
18	80	1	19	144,599	50792
19	80	1	19	105,474	38075
20	80	1	19	232,694	80710
PROMEDIO				80,34915	28427

**ANEXO I: TABLA DE RESULTADOS CON CPSO PARA EL EJERCICIO DE
LOS INTERIORES**

Tabla 37. Resultados para el ejercicio de los interiores con CPSO

PRUEBA	x	y	z	TIEMPO [s]	# ITERACIONES
1	5	7	7	6,9780	1629
2	5	7	7	46,2250	10558
3	9	3	6	7,9230	1844
4	1	11	8	25,7960	5810
5	1	11	8	7,7110	1832
6	1	11	8	108,4710	24492
7	1	11	8	125,2220	23740
8	1	11	8	14,8010	2793
9	9	3	6	44,1660	9371
10	5	7	7	78,2550	14189
11	1	11	8	20,1310	3668
12	9	3	6	5,2920	1237
13	5	7	7	168,5790	31785
14	1	11	8	155,2250	30248
15	1	11	8	58,4000	13071
16	9	3	6	13,4380	3058
17	1	11	8	7,8400	1457
18	9	3	6	9,2750	1815
19	5	7	7	6,9320	1620
20	1	11	8	84,1060	17726
PROMEDIO				49,7383	10097

ANEXO J: RESULTADOS ENCONTRADOS CON CPSO PARA EL EJERCICIO DEL MECANISMO

Tabla 38. Resultados para el ejercicio del mecanismo con CPSO

PRUEBA	<i>x</i>	<i>y</i>	TIEMPO [s]	# ITERACIONES
1	5	4	4,6040	1551
2	5	4	3,4400	1151
3	5	4	0,6350	193
4	5	4	0,5290	164
5	5	4	20,3840	5801
6	5	4	14,7440	3967
7	5	4	10,4190	3469
8	5	4	2,6900	911
9	5	4	0,4500	140
10	5	4	0,6060	193
11	5	4	0,3680	113
12	5	4	4,4650	1487
13	5	4	0,7680	220
14	5	4	9,4180	3157
15	5	4	4,3130	1438
16	5	4	0,6070	200
17	5	4	3,2720	1091
18	5	4	14,2510	4563
19	5	4	1,9300	640
20	5	4	1,6290	544
PROMEDIO			4,9761	1549

**ANEXO K: TABLAS DE RESULTADOS CON OTROS METODOS PARA LA
SECCION 6.1**

En este anexo se presentan los resultados encontrados con los métodos de CPSO, PSO y UPSO los cuales serán utilizados para llevar a cabo el análisis de la eficiencia del método de CUPSO.

Tabla 39. Datos hallados con el algoritmo CPSO

Prueba	X1	X2	X3	X4	x5	X6	X7	Tiempo (s)	# Iteraciones
1	1	3	2	2	0	-3	-5	179,8390	17170
2	1	3	2	2	0	-3	-5	296,7990	20567
3	1	3	2	2	0	-3	-5	84,7060	3573
4	1	3	2	2	0	-3	-5	106,1180	9645
5	1	3	2	2	0	-3	-5	670,2150	32916
6	1	3	2	2	0	-3	-5	515,6290	17488
7	1	3	2	2	0	-3	-5	249,5310	6917
8	1	3	2	2	0	-3	-5	787,8540	18132
9	1	3	2	2	0	-3	-5	522,8100	17488
10	1	3	2	2	0	-3	-5	106,2570	6917
11	1	3	2	2	0	-3	-5	416,1122	18312
12	1	3	2	2	0	-3	-5	1025,100	39768
13	1	3	2	2	0	-3	-5	255,7180	12774
14	1	3	2	2	0	-3	-5	387,0400	31014
15	1	3	2	2	0	-3	-5	121,5210	12807
16	1	3	2	2	0	-3	-5	179,9300	11821
17	1	3	2	2	0	-3	-5	67,8200	5530
18	1	3	2	2	0	-3	-5	332,6900	11138
19	1	3	2	2	0	-3	-5	298,9860	11250
20	1	3	2	2	0	-3	-5	333,2030	15132
PROMEDIO								346,8940	16018

Tabla 40. Datos del ejercicio implementado con PSO

PRUEBA	X1	X2	X3	X4	X5	X6	X7	TIEMPO [S]	#ITERACIONES	PRECISION
1	0,9993	3,0021	2,0032	2,0029	-0,0067	-3,0122	-5,0089	984,9497	10000	4,94E-07
2	0,9998	3,0006	2,0009	2,0008	-0,0019	-3,0034	-5,0025	810,6500	10000	3,97E-08
3	0,9818	1,0000	1,0000	0,9269	0,8000	0,9381	0,8000	775,8556	10000	1,18E+02
4	1,0000	0,9639	1,0000	0,8414	0,9765	0,9341	0,8000	571,8714	10000	1,14E+02
5	1,0037	2,9878	1,9828	1,9828	0,0384	-2,9329	-4,9518	361,9411	10000	1,55E-06
6	1,0044	2,9861	1,9797	1,9802	0,0436	-2,9203	-4,9401	333,2912	10000	2,17E-05
7	1,0052	2,9839	1,9756	1,9761	0,0506	-2,9043	-4,9278	354,8188	10000	3,11E-06
8	1,0026	2,9917	1,9878	1,9883	0,0263	-2,9516	-4,9649	474,9887	10000	7,68E-06
9	1,0077	2,9759	1,9637	1,9656	0,0766	-2,8589	-4,8959	731,2861	10000	6,63E-05
10	0,9989	3,0035	2,0054	2,0052	-0,0113	-3,0209	-5,0156	486,1983	10000	1,48E-06
11	1,0000	1,0000	1,0000	1,0000	1,0000	1,0000	1,0000	492,7576	10000	1,07E+02
12	1,0028	2,9914	1,9873	1,9877	0,0273	-2,9504	-4,9630	329,0553	10000	8,43E-06
13	1,0091	2,9714	1,9573	1,9587	0,0899	-2,8326	-4,8756	338,2344	10000	9,38E-05
14	0,9982	3,0058	2,0090	2,0082	-0,0185	-3,0345	-5,0250	356,0706	10000	3,94E-06
15	1,0043	2,9861	1,9799	1,9804	0,0439	-2,9214	-4,9408	303,3973	10000	2,15E-05
16	1,0141	2,9551	1,9316	1,9377	0,1443	-2,7369	-4,8113	356,9076	10000	2,29E-04
17	0,9971	3,0093	2,0133	2,0129	-0,0290	-3,0522	-5,0387	356,7123	10000	9,28E-06
18	1,0093	2,9707	1,9561	1,9586	0,0931	-2,8293	-4,8748	361,1142	10000	9,69E-05
19	0,9968	3,0102	2,0145	2,0139	-0,0320	-3,0569	-5,0417	342,6987	10000	1,11E-05
20	1,0011	2,9965	1,9946	1,9950	0,0111	-2,9792	-4,9848	305,3625	10000	1,43E-06
PROMEDIO	1,0032	2,9900	1,9850	1,9857	0,0318	-2,9416	-4,9606	439,8185	10000	3,3983E-05

Las columnas resaltadas de la Tabla 40 contienen las iteraciones donde el método no encontró las raíces propias de la función objetivo.

Tabla 41. Datos del ejercicio implementado con UPSO

PRUEBA	X1	X2	X3	X4	X5	X6	X7	TIEMPO [S]	# ITERACIONES
1	1	3	2	2	0	-3	-5	48,0983	3957
2	1	3	2	2	0	-3	-5	55,4463	4306
3	1	3	2	2	0	-3	-5	32,5375	2984
4	1	3	2	2	0	-3	-5	27,7563	2656
5	1	3	2	2	0	-3	-5	34,3693	3084
6	1	3	2	2	0	-3	-5	34,5145	3063
7	1	3	2	2	0	-3	-5	38,9016	3346
8	1	3	2	2	0	-3	-5	32,2467	2957
9	1	3	2	2	0	-3	-5	36,8427	3279
10	1	3	2	2	0	-3	-5	31,8815	2940
11	1	3	2	2	0	-3	-5	28,4237	2691
12	1	3	2	2	0	-3	-5	37,4766	3317
13	1	3	2	2	0	-3	-5	35,2613	3174
14	1	3	2	2	0	-3	-5	30,2335	2834
15	1	3	2	2	0	-3	-5	29,1651	2747
16	1	3	2	2	0	-3	-5	41,6669	3297
17	1	3	2	2	0	-3	-5	39,2395	3264
18	1	3	2	2	0	-3	-5	34,0679	2950
19	1	3	2	2	0	-3	-5	37,9715	3226
20	1	3	2	2	0	-3	-5	38,0552	3260
PROMEDIO								36,2078	3167