



UNIVERSIDAD AUTÓNOMA DEL ESTADO
DE HIDALGO

INSTITUTO DE CIENCIAS BÁSICAS E
INGENIERÍA

ÁREA ACADÉMICA DE INGENIERÍA

Sintonización de un control
retroalimentado para el péndulo
doble usando computación
evolutiva

T E S I S
PARA OBTENER EL TÍTULO DE
Maestro en Ciencias en Ingeniería de Manufactura
P R E S E N T A :
Abraham Manilla García

DIRECTOR DE TESIS:

Dr. Norberto Hernández Romero

CODIRECTOR DE TESIS:

Dr. Oleksander Karelin

Febrero de 2015



UNIVERSIDAD AUTÓNOMA DEL ESTADO DE HIDALGO
INSTITUTO DE CIENCIAS BÁSICAS E INGENIERÍA
DIRECCIÓN

M. en A. Julio César Leines Medécigo
Director de Administración Escolar
P R E S E N T E

Por este conducto le comunico que el jurado asignado al pasante, de la Maestría en Ciencias en Ingeniería en Manufactura, **C. Abraham Manilla García**, con número de cuenta **281344**, quien presenta el trabajo de tesis titulado **"Sintonización de un control retroalimentado para el péndulo doble usando computación evolutiva"**, después de revisar el trabajo en reunión de sinodales, ha decidido autorizar la impresión del mismo una vez realizadas las correcciones que fueron acordadas.

A continuación se anotan las firmas de conformidad de los integrantes del jurado.

PRESIDENTE: Dr. Juan Carlos Seck Tuoh Mora
SECRETARIO: Dr. Oleksandr Karelin
VOCAL: Dr. Norberto Hernández Romero
SUPLENTE: Dr. Joselito Medina Marín

Sin otro particular, reitero a Usted la seguridad de mi atenta consideración.

ATENTAMENTE

"Amor, Orden y Progreso"

Mineral de la Reforma, Hgo., a 23 de enero de 2015.

Dr. Orlando Avila Pozos
Director del ICBI



Ciudad Universitaria Carretera Pachuca-Tulancingo
Km. 4.5. s/n Col. Carboneras C.P. 42184
Mineral de la Reforma, Hidalgo. México.
Tel: (771)7172000 ext. 2230 , 2231



CONTENIDO

Contenido	i
Anexos	v
Índice de figuras y tablas	vi
 Resumen	 I
Abstract	II
Objetivo general.....	III
Objetivos particulares	III
Justificación	IV
Hipótesis.....	IV
 Capítulo 1. Marco teórico	 1
1.1 Introducción	1
1.2 Sistemas subactuados.....	2
1.3 Sistema subactuado péndulo doble invertido	3
1.4 Teoría de control.....	5
1.5 Inteligencia artificial.....	5
1.6 Computación evolutiva.....	6
1.7 Aplicaciones de la computación evolutiva	8
Referencias	8
 Capítulo 2. Modelación dinámica del péndulo doble invertido.....	 10
2.1 Introducción	10
2.2 Modelación de sistemas dinámicos.....	11
2.2.1 Fases de construcción de un modelo	12
2.3 Dinámica Lagrangiana	13
2.3.1 Fuerzas disipativas en la ecuación de Lagrange	14
2.4 Modelación dinámica del péndulo doble invertido	14
2.5 Ecuaciones de estado del modelo dinámico de péndulo doble invertido	17
Referencias	18

Capítulo 3. Control de posición del péndulo doble por retroalimentación de estados	19
3.1 Introduccion	19
3.2 Puntos críticos de equilibrio del sistema	19
3.3 Desarrollo del control para el sistema de péndulo doble invertido lineal.....	21
3.3.1 Modelo dinámico en variables de estado	21
3.3.2 Análisis de puntos críticos del sistema dinámico	22
3.3.3 Linealización del sistema dinámico	24
3.3.4 Desarrollo del control retroalimentado del sistema	32
3.4 Desarrollo del control del sistema dinámico del péndulo doble invertido no lineal	33
3.5 Modelo computacional del péndulo doble invertido	34
3.6 Parámetros de prueba en el diseño del péndulo doble invertido	40
3.7 Análisis dinámico del sistema péndulo doble invertido	41
3.8 Análisis de control lineal en el modelo dinámico	43
3.9 Análisis de control no lineal en el modelo dinámico	45
Capítulo 4. Algoritmos evolutivos y de enjambre.....	48
4.1 Introducción	48
4.2 Métodos de optimización basados en computación evolutiva	48
4.3 Método de sintonización del control mediante AG.....	49
4.3.1 Población inicial	51
4.3.2 Evaluación	51
4.3.3 Selección	51
4.3.3.1 Selección por ruleta	52
4.3.3.2 Selección por truncamiento	52
4.3.3.3 Selección basada en ranking	52
4.3.3.4 Selección por torneo	53
4.3.4 Cruce	53
4.3.5 Mutación	53
4.4 Método de sintonización del control mediante ED.....	54
4.4.1 Población inicial	56
4.4.2 Mutación	56
4.4.3 Cruce	57
4.4.4 Selección	57

4.5 Método de sintonización del control mediante PSO	57
4.6 Caso de aplicación al modelo dinámico no lineal de péndulo doble invertido.....	60
Referencias	62
Capítulo 5. Desarrollo de pruebas con algoritmos evolutivos	63
5.1 Introducción	63
5.2 Análisis de control no lineal en el modelo dinámico basado en algoritmos genéticos.	63
5.3 Análisis de control no lineal en el modelo dinámico basados en evolución diferencial	67
5.4 Análisis de control no lineal en el modelo dinámico basados en optimización por enjambre de partículas	70
Capítulo 6. Análisis de datos	75
6.1 Introducción	75
6.2 Análisis de sensibilidad de los algoritmos referentes a los límites de las condiciones iniciales.....	76
6.2.1 Variación de las condiciones iniciales del control del modelo linealizado	76
6.2.2 Variación de las condiciones iniciales en el control del modelo no lineal.....	77
6.2.3 Variación de las condiciones iniciales del control del modelo no lineal basado en algoritmos genéticos	78
6.2.4 Variación de las condiciones iniciales del control del modelo no lineal basado en evolución diferencial	79
6.2.5 Variación de las condiciones iniciales del control del modelo no lineal basado en optimización por enjambre de partículas.....	79
6.3 Sensibilidad de algoritmos referentes a los límites de las ganancias	80
6.3.1 Sensibilidad presente en el algoritmo genético	80
6.3.2 Sensibilidad presente en el algoritmo de evolución diferencial	82
6.3.3 Sensibilidad presente en el algoritmo de optimización por enjambre de partículas	84
6.4 Variación en el tiempo de computo del algoritmo	86
6.5 Comportamiento de los algoritmos frente a la disminución de individuos.....	88
6.5.1 Comportamiento del algoritmo genético	88
6.5.2 Comportamiento del algoritmo de evolución diferencial.....	90
6.5.3 Comportamiento del algoritmo de optimización por enjambre de partículas.....	95
6.6 Sensibilidad de los algoritmos frente a la variación en el número de evoluciones.....	97
6.6.1 Variación del algoritmo genético	97
6.6.2 Variación del algoritmo de evolución diferencial.....	100

6.6.3 Variación del algoritmo de optimización por enjambre de partículas	103
6.7 Tasa de mutación presente en los sistemas	107
6.7.1 Mutación en el algoritmo genético	107
6.7.2 Mutación en el algoritmo de evolución diferencial	108
6.7.3 Mutación en el algoritmo de optimización por enjambre de partículas.....	109
Capítulo 7. Conclusiones	110
7.1 Introducción	110
7.2 Análisis de sensibilidad de los algoritmos referentes a los límites de las condiciones iniciales.....	110
7.3 Sensibilidad de algoritmos referentes a los límites de las ganancias	111
7.4 Variación en el tiempo de cómputo	112
7.5 Comportamiento de los algoritmos frente a la disminución de individuos.....	112
7.6 Sensibilidad de los algoritmos frente a la variación en el número de evoluciones....	113
7.7 Tasa de mutación	113
7.8 Conclusiones generales.....	114
7.9 Trabajos futuros.....	115

ANEXOS

I Programa en Matlab del modelo dinámico del sistema péndulo doble invertido no lineal	116
II Solución numérica del sistema péndulo doble invertido no lineal	117
III Programación en Matlab para obtener el modelo dinámico en variables de estado ...	118
IV Programación en Matlab para obtener el modelo dinámico en variables de estado linealizado.....	119
V Programa en Matlab para desarrollar prueba de controlabilidad.....	120
VI Programa en Matlab para controlar el sistema dinámico lineal por retroalimentación de estados.....	121
VII Programa en Matlab para controlar el sistema dinámico no lineal por retroalimentación de estados	122
VIII Programa Matlab sintonización por medio de algoritmos genéticos.....	123
IX Programa Matlab sintonización por medio de algoritmo evolución diferencial.....	125
X Programa Matlab sintonización por medio de algoritmo optimización por enjambre de partículas	128

INDICE DE FIGURAS Y TABLAS

Fig. 1.1. Ejemplo de sistemas dinámicos subactuados	2
Fig. 1.2. Sistema dinámico péndulo doble.....	3
Fig. 1.3. Ramas de la computación evolutiva.....	7
Fig. 2.1. péndulo doble invertido	11
Fig. 3.1. Posición de equilibrio estable del péndulo doble invertido.....	19
Fig. 3.2. Posición de equilibrio inestable del péndulo doble invertido	20
Fig. 3.3. Punto de equilibrio para efectuar el control	24
Fig. 3.4. Control por retroalimentación de estados del péndulo doble invertido linealizado	32
Fig. 3.5. Control por retroalimentación de estados del péndulo doble invertido no lineal..	33
Fig. 3.6. Ensamble de modelo de prueba péndulo doble invertido	40
Fig. 3.7. Comportamiento dinámico de las variables del sistema dinámico de estudio bajo variación inicial 1	42
Fig. 3.8. Comportamiento dinámico de las variables del sistema dinámico de estudio bajo variación inicial 2	43
Fig. 3.9. Comportamiento dinámico control lineal variación primera.....	44
Fig. 3.10. Comportamiento dinámico control lineal variación segunda	44
Fig. 3.11. Comportamiento dinámico control no lineal variación primera.....	46
Fig. 3.12. Comportamiento dinámico control no lineal variación segunda	46
Fig. 3.13. Comportamiento dinámico control no lineal variación límite	47
Fig. 4.1. Diagrama de flujo de la estructura del algoritmo genético continuo.....	50
Fig. 4.2. Diagrama de flujo de la estructura del algoritmo evolución diferencial.....	55
Fig. 4.3. Diagrama de flujo de la estructura del algoritmo PSO	59
Fig. 4.4. Control por retroalimentación de estados del péndulo doble invertido.....	61
Fig. 5.1. Iteración 1–4 de búsqueda de solución óptima	64
Fig. 5.2. Iteración 5–8 de búsqueda de solución óptima	64
Fig. 5.3. Iteración 9–12 de búsqueda de solución óptima	65
Fig. 5.4. Iteración 13–16 de búsqueda de solución óptima	65
Fig. 5.5. Iteración 17–20 de búsqueda de solución óptima	66
Fig. 5.6. Solución óptima: algoritmo genético.....	66

Fig. 5.7. Iteración 1–4 de búsqueda de solución óptima	67
Fig. 5.8. Iteración 5–8 de búsqueda de solución óptima	68
Fig. 5.9. Iteración 9–12 de búsqueda de solución óptima	68
Fig. 5.10. Iteración 13–16 de búsqueda de solución óptima	69
Fig. 5.11. Iteración 17–20 de búsqueda de solución óptima	69
Fig. 5.12. Solución óptima: evolución diferencial.....	70
Fig. 5.13. Iteración 1–4 de búsqueda de solución óptima	71
Fig. 5.14. Iteración 4–8 de búsqueda de solución óptima	71
Fig. 5.15. Iteración 9-12 de búsqueda de solución óptima	72
Fig. 5.16. Iteración 13-16 de búsqueda de solución óptima	72
Fig. 5.17. Iteración 17–20 de búsqueda de solución óptima	73
Fig. 5.18. Solución óptima: optimización por enjambre de partículas	73
Fig. 6.1. Comportamiento de las variables de estado, variación de 90° (izquierda), variación 180 (derecha)	77
Fig. 6.2. Comportamiento de las variables de estado no lineal, estable (izquierda), inestable (derecha)	78
Fig. 6.3. Sintonización de las variables de estado no lineal por método de los algoritmos genéticos valor crítico, estable (derecha), inestable (izquierda)	78
Fig. 6.4. Sintonización de las variables de estado no lineal por método de evolución diferencial valor crítico, estable (derecha), inestable (izquierda)	79
Fig. 6.5. Sintonización de las variables de estado no lineal por método de optimización por enjambre de partículas valor crítico, estable (derecha), inestable (izquierda)	80
Fig. 6.6. Sintonización AG valores de ganancias estables, ganancias 1 (izquierda), ganancias 2 (derecha)	81
Fig. 6.7. Sintonización AG valores de ganancias inestables, ganancias 1 (izquierda), ganancias 2 (derecha)	81
Fig. 6.8. Sintonización AG valores de ganancias estables críticos, ganancias 1 (izquierda), ganancias 2 (derecha)	82
Fig. 6.9. Sintonización ED valores de ganancias estables, ganancias 1 (izquierda), ganancias 2 (derecha)	83
Fig. 6.10. Sintonización ED valores de ganancias inestables, ganancias 1 (izquierda), ganancias 2 (derecha)	83
Fig. 6.11. Sintonización ED valores de ganancias estables críticas, ganancias 1 (izquierda), ganancias 2 (derecha).....	84
Fig. 6.12. Sintonización PSO valores de ganancias estables, ganancias 1 (izquierda), ganancias 2 (derecha)	84

Fig. 6.13. Sintonización PSO valores de ganancias inestables, ganancias 1 (izquierda), ganancias 2 (derecha)	85
Fig. 6.14. Sintonización PSO valores de ganancias inestables (izquierda), estables (derecha)	86
Fig. 6.15. Variación de tiempo de AG	87
Fig. 6.16. Variación de tiempo de ED.....	87
Fig. 6.17. Variación de tiempo de PSO	88
Fig. 6.18. Comportamiento AG 300 individuos.....	89
Fig. 6.19. Comportamiento AG 260 individuos.....	89
Fig. 6.20. Comportamiento AG 240 individuos.....	90
Fig. 6.21. Comportamiento ED 300 individuos	91
Fig. 6.22. Comportamiento ED 280 individuos	91
Fig. 6.23. Comportamiento ED 260 individuos	92
Fig. 6.24. Comportamiento ED 240 individuos	92
Fig. 6.25. Comportamiento ED 220 individuos	93
Fig. 6.26. Comportamiento ED 200 individuos	93
Fig. 6.27. Comportamiento ED 180 individuos	94
Fig. 6.28. Comportamiento ED 160 individuos	94
Fig. 6.29. Comportamiento ED 140 individuos	95
Fig. 6.30. Comportamiento PSO 300 individuos.....	96
Fig. 6.31. Comportamiento PSO 280 individuos.....	96
Fig. 6.32. Comportamiento AG 20 evoluciones	97
Fig. 6.33. Comportamiento AG 16 evoluciones	98
Fig. 6.34. Comportamiento AG 12 evoluciones	98
Fig. 6.35. Comportamiento AG 8 evoluciones.....	99
Fig. 6.36. Comportamiento AG 4 evoluciones	99
Fig. 6.37. Comportamiento AG 1 evolución.....	100
Fig. 6.38. Comportamiento ED 20 evoluciones	100
Fig. 6.39. Comportamiento ED 16 evoluciones	101
Fig. 6.40. Comportamiento ED 12 evoluciones	101
Fig. 6.41. Comportamiento ED 8 evoluciones	102
Fig. 6.42. Comportamiento ED 4 evoluciones	102
Fig. 6.43. Comportamiento ED 1 evolución.....	103
Fig. 6.44. Comportamiento PSO 20 evoluciones.....	104

Fig. 6.45. Comportamiento PSO 16 evoluciones.....	104
Fig. 6.46. Comportamiento PSO 12 evoluciones.....	105
Fig. 6.47. Comportamiento PSO 8 evoluciones.....	105
Fig. 6.48. Comportamiento PSO 4 evoluciones.....	106
Fig. 6.49. Comportamiento PSO 1 evolución	106
Fig. 6.50. Resultado comportamiento AG tasa de mutación 1 (izquierda), tasa de mutación 5% (derecha)	107
Fig. 6.51. Resultado comportamiento AG tasa de mutación 10% (izquierda), tasa de mutación 20% (derecha).....	107
Fig. 6.52. Resultado comportamiento ED tasa de mutación 1% (izquierda), tasa de mutación 2% (derecha).....	108
Fig. 6.53. Resultado comportamiento ED tasa de mutación 5% (izquierda). Tasa de mutación 7% (derecha).....	108
Fig. 6.54. Resultado comportamiento PSO tasa de mutación 1% (izquierda), tasa de mutación 2% (derecha).....	109
Fig. 6.55. Resultado comportamiento PSO tasa de mutación 5% (izquierda). Tasa de mutación 7% (derecha).....	109
Tabla 3.1. Características mecánicas del rodamiento soporte	35
Tabla 3.2. Características mecánicas de la tuerca de unión	36
Tabla 3.3. Características mecánicas del soporte fijo	36
Tabla 3.4. Características mecánicas de la base de soporte	37
Tabla 3.5. Características mecánicas del perno de ajuste	37
Tabla 3.6. Características mecánicas del brazo inferior	38
Tabla 3.7. Características mecánicas del perno de unión	38
Tabla 3.8. Características mecánicas del brazo superior	39
Tabla 7.1. Resumen de resultados basados en las pruebas realizadas.....	114

RESUMEN

Este trabajo trata sobre la modelación del control a nivel simulación de un sistema subactuado denominado péndulo doble invertido o Pendubot, en el cual se desea analizar el comportamiento del sistema de control del mismo para posiciones de equilibrio inestable mediante la sintonización de un controlador basado en algoritmos evolutivos.

Se desarrolla este trabajo en cuatro partes, en la primera parte se desprende una introducción al comportamiento del sistema dinámico y las variables que intervienen en el mismo, los puntos de equilibrio del sistema, la formulación matemática y la teoría referente al control convencional del mismo.

En la segunda parte se analiza el comportamiento dinámico del sistema y se desarrolla el control del modelo matemático del mismo, primeramente mediante teorías de linealización basadas en series de Taylor para posteriormente utilizar ésta para el control del sistema dinámico no lineal.

En la tercera parte se implementa la sintonización del control por retroalimentación de estados del sistema dinámico no lineal mediante el empleo de técnicas basadas en algoritmos evolutivos, se implementan tres métodos para llevar a cabo este procedimiento; algoritmos genéticos, evolución diferencial y optimización por enjambre de partículas.

Finalmente con la información obtenida se evalúan las técnicas de control del sistema dinámico no lineal y se obtienen conclusiones en base a parámetros establecidos de la efectividad de cada acción de control.

ABSTRACT

This thesis studies the modeling of a control for an underactuated double inverted pendulum system called Pendubot. In this system, it is desirable to analyze the behavior of the control system for the unstable equilibrium positions by tuning a controller based on evolutionary algorithms.

This manuscript is divided in four parts, the first part introduces the dynamic behavior of the system and the variables involved in it, the equilibrium points of the system, the mathematical formulation and the theory concerning the conventional control.

In the second part the dynamic behavior of the system is analyzed and a control model is developed. Linearization theories based on Taylor series are applied to control the nonlinear dynamic system.

In the third section, the tuning of the state feedback control for the nonlinear case is achieved by evolutionary algorithms based on three methods; genetic algorithms, differential evolution and particle swarm optimization.

Finally, with the information obtained, the distinct control techniques of the nonlinear dynamic system are evaluated. Concluding remarks are presented based on the effectiveness of each control.

OBJETIVO GENERAL

Desarrollar los algoritmos evolutivos que logren sintonizar el control basado en retroalimentación de estados del sistema péndulo doble invertido no lineal mediante los métodos de algoritmos genéticos, evolución diferencial y optimización por enjambre de partículas.

OBJETIVOS PARTICULARES

- Determinar el modelo dinámico del sistema péndulo doble mediante la formulación de Lagrange.
- Analizar el comportamiento dinámico del sistema dinámico en cuestión.
- Determinar los puntos críticos del modelo matemático del sistema de estudio.
- Linealizar el sistema dinámico de estudio en el punto de equilibrio superior inestable.
- Desarrollar el control por retroalimentación de estados del sistema dinámico de estudio linealizado en el punto de equilibrio inestable.
- Optimizar un control de retroalimentación de estados mediante un algoritmo genético.
- Optimizar un control de retroalimentación de estados mediante evolución diferencial.
- Optimizar un control de retroalimentación de estados mediante optimización por enjambre de partículas.
- Analizar el comportamiento de los controles optimizados por algoritmos evolutivos.

JUSTIFICACIÓN

Los sistemas subactuados constituyen un conjunto muy rico en recursos y en aplicaciones físicas, tal es el caso de la robótica, sin embargo, dada su naturaleza constitutiva; las formulaciones matemáticas para efectuar los controles de los mismos resultan ser muy complicadas. Los algoritmos basados en computación evolutiva constituyen un nicho de oportunidad para sintonizar los sistemas dinámicos que, en general ocupan sistemas dinámicos complejos. El estudio de técnicas de control basados en sistemas no gradientales permite entre otras cosas reducir la complejidad del análisis de este tipo de sistemas.

El sistema dinámico péndulo invertido tiene muchas aplicaciones, la más común es la de ser la base para el modelado de sistemas reales por citar un ejemplo se tiene el caso de robots seriales los cuales basan su arquitectura en sistemas articulados de varios grados de libertad donde la formulación más simple de los mismos se basa en un modelo de dos grados de libertad.

Lograr manipular este tipo de sistemas en base a métodos sencillos permite entre otras cosas que el conocimiento del funcionamiento de los mismos se difunda de una manera más amplia en la comunidad y que se potencialice su uso y desarrollo.

HIPÓTESIS

Mediante métodos basados en computación evolutiva se puede lograr eficientemente la acción de sintonización de sistemas dinámicos subactuados sin la necesidad de desarrollar métodos analíticos que en general consumen demasiado tiempo en la implementación y definición de los mismos.

CAPÍTULO 1

MARCO TEÓRICO

Resumen:

En el presente capítulo se realiza una introducción al tema de estudio, primeramente se describe que es un sistema dinámico, las características del mismo, y como identificarlo, posteriormente se define el sistema dinámico subactuado, dado que el péndulo doble invertido es de este tipo, se dan a conocer las características de este, posteriormente se realiza una reseña sobre la teoría de control y su importancia, la relación a la inteligencia artificial y como es que esta se involucra en el desarrollo de controladores de sistemas dinámicos mediante sistemas evolutivos.

1.1.- INTRODUCCIÓN

Un sistema se llama dinámico si su salida en el presente depende de una entrada en el pasado; si su salida en curso depende solamente de la entrada en curso, el sistema se conoce como estático. La salida de un sistema estático permanece constante si la entrada no cambia y cambia solo cuando la entrada cambia. En un sistema dinámico la salida cambia con el tiempo cuando no está en su estado de equilibrio [1.1].

En el transcurso de la búsqueda de soluciones y modelamiento de sistemas de ingeniería capaces de resolver algún problema característico se llega a la necesidad de que los sistemas dinámicos adquieran comportamientos determinados, es ahí donde nace la teoría de control [1.2].

Como consecuencia de estos comentarios previos son necesarios nuevos procedimientos para poder obtener comportamientos más cercanos a la morfología y los recursos del robot, que, como se ha comentado, serían demasiado complejos para diseñar por parte de un humano por los métodos tradicionales [1.3].

Tratando de imitar a la naturaleza en la forma en como optimiza y realiza la selección de las mejores características surgen los algoritmos evolutivos, los cuales tienen sus inicios entre los años 1950 y 1970. La Computación Evolutiva (CE) es un campo reconocido recientemente y el término fue desarrollado en 1991 [1.4]. La CE representa el esfuerzo de unir a los investigadores que tienen diferentes acercamientos con la simulación de los diferentes aspectos de la evolución. Todas estas técnicas, Programación Evolutiva (PE), Estrategias de Evolución (EE) y Algoritmos Genéticos (AG), tienen muchos aspectos comunes, tales como la población de individuos, mecanismos de evolución sobre la población tales como selección, cruce, mutación, etc. [1.5].

Este campo tiene diversas aplicaciones y se desarrolla a un ritmo muy acelerado, entre estas aplicaciones se puede citar la búsqueda de parámetros de control de sistemas robóticos como el desarrollado en el presente trabajo así como aplicaciones en otros campos como los listados a continuación [1.6]:

- Gestión y control: análisis inteligente, fijación de objetivos.
- Fabricación: diseño, planificación, programación, monitorización, control, gestión de proyectos, robótica simplificada y visión computarizada.
- Educación: adiestramiento práctico, exámenes y diagnóstico.
- Ingeniería: diseño, control y análisis.
- Equipamiento: diseño, diagnóstico, adiestramiento, mantenimiento, configuración, monitorización y ventas.

- Cartografía: interpretación de fotografías, diseño, resolución de problemas cartográficos.
- Software: enseñanza, especificación, diseño, verificación, mantenimiento.
- Sistemas de armamento: guerra electrónica, identificación de objetivos, control adaptativo, proceso de imágenes, proceso de señales.
- Proceso de datos: educación, interface en lenguaje natural, acceso inteligente a datos y gestores de bases de datos, análisis inteligente de datos.
- Finanzas: planificación, análisis, consultoría.

1.2.- SISTEMAS SUBACTUADOS

La modelación y el diseño de la mayoría de los modelos de sistemas mecánicos, tales como manipuladores robóticos, se basa en la suposición de que los enlaces individuales o miembros son rígidos. Esta es una aproximación correcta, en algunos casos, mientras que en otros no lo es. Si se toma la dinámica de sistemas no rígidos más realistas en cuenta, entonces todos esos modelos son esencialmente subactuados.

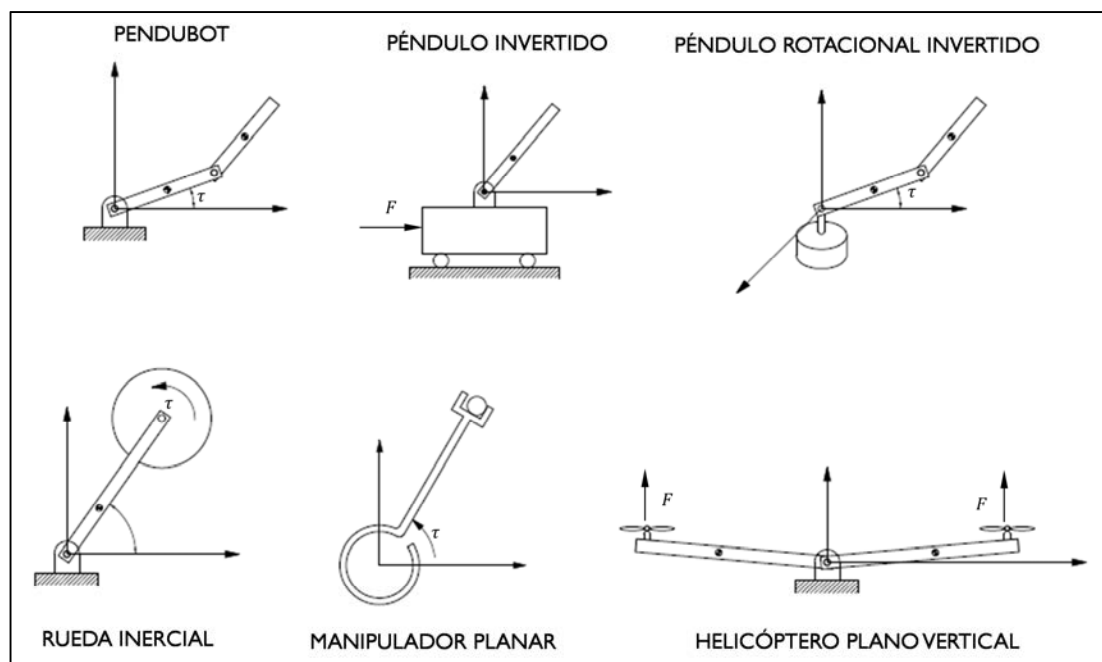


Fig. 1.1. Ejemplo de sistemas dinámicos subactuados

Los sistemas mecánicos subactuados son aquellos que cuentan con menos actuadores que grados de libertad, como ejemplo se pueden citar diversos sistemas, como los mostrados en la figura 1.1 [1.7], cuando un brazo manipulador es adherido a una plataforma móvil, un espacio de plataforma, o un vehículo sumergido, o porque en el modelo matemático es incluido el diseño de una junta

flexible en la acción de control, es cuando surgen sistemas dinámicos subactuados por citar un ejemplo.

En tal sentido, todos los sistemas mecánicos son subactuados si se desean modos de control basados en el acoplamiento de juntas flexibles, o en acciones de nivelación para incluir algún elemento como actuador dinámico en la descripción del modelo.

En los últimos años, ha existido un gran interés en el desarrollo de algoritmos para la estabilización de los sistemas mecánicos subactuados. El interés viene de la necesidad de estabilizar los sistemas como barcos, vehículos submarinos, helicópteros, aviones, aeronaves, aerodeslizadores, satélites, los robots caminantes, etc., que, dada su naturaleza, su diseño puede ser subactuado [1.8].

Los sistemas mecánicos subactuados son sistemas físicos de gran complejidad y se caracterizan porque las técnicas de control convencionales son ineficaces cuando se requiere estabilizarlos en un punto de operación, un sistema ampliamente usado es el péndulo doble invertido, debido a que permite ilustrar la aplicación y validación de diferentes técnicas de control y su implementación en diferentes dispositivos.

1.3.- SISTEMA SUBACTUADO PÉNDULO DOBLE INVERTIDO

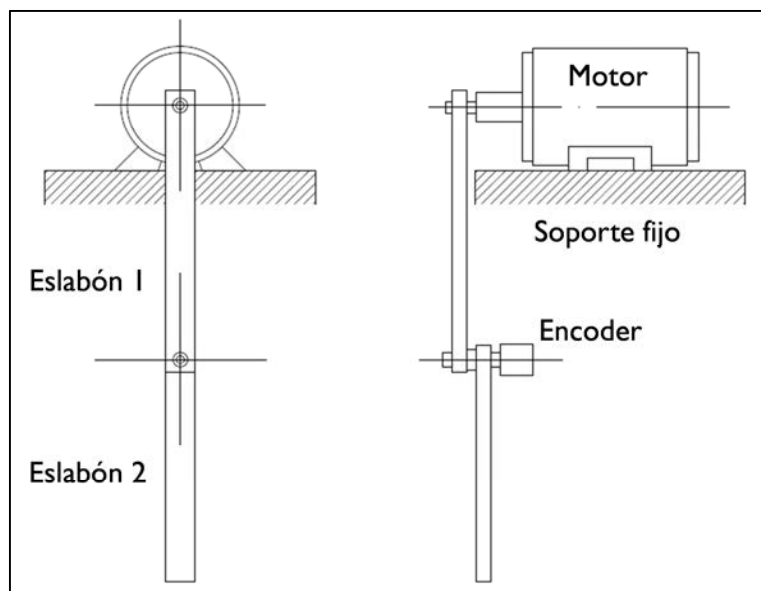


Fig. 1.2. Sistema dinámico péndulo doble

El péndulo es un sistema físico que puede oscilar bajo la acción gravitatoria u otra característica física (elasticidad, por ejemplo) y que está configurado por una masa

suspendida de un punto o de un eje horizontal fijo mediante un hilo, una varilla, u otro dispositivo.

En la figura 1.2 se muestra el péndulo doble invertido de estudio. Este es un sistema compuesto por dos péndulos, con el segundo colgando del extremo del primero. Normalmente se sobreentiende que se hace referencia a un péndulo doble plano, con dos péndulos planos coplanarios. Este sistema físico posee dos grados de libertad, Su movimiento está gobernado por dos ecuaciones diferenciales ordinarias acopladas. Por encima de cierta energía, su movimiento es caótico, es decir movimientos tendientes a ser impredecibles o indeterminados bajo métodos analíticos.

El péndulo doble es un sistema dinámico subactuado muy utilizado en el modelado de sistemas mecánicos que en general requieren estudios referentes al campo de la estabilidad en campos tan variados como son el diseño de prótesis en el área de biomédica y en el diseño de estabilizadores de estructuras como son los rascacielos [1.9, 1.10, 1.11].

Dentro de los múltiples sistemas que pueden modelarse matemáticamente a través de péndulos dobles existe una aplicación en el campo de la robótica que es utilizada, entre otras aplicaciones, para modelar brazos de robots.

Este sistema subactuado denominado péndulo doble invertido se utiliza para la investigación en el control no lineal en diversos conceptos como la dinámica no lineal, la robótica y el diseño de controladores de sistemas.

Este dispositivo es un robot planar de dos eslabones con un actuador en el hombro (eslabón 1) y sin actuador en el codo (eslabón 2). El eslabón 2 se mueve libremente alrededor del eslabón 1 y el objetivo de control es llevar el mecanismo a puntos de equilibrio inestable.

Fue desarrollado originalmente en *Coordinated Science Laboratory, University of Illinois at Urbana Champaign*, por el profesor Mark W. Spong [1.12]. El Pendubot, nombre formado por la contracción de las palabras *pendulum* y *robot*, de enlaces rígidos, planar y rotacional de dos grados de libertad, diseñado para investigación [1.13].

La complejidad del péndulo doble invertido, por ser un sistema no lineal de segundo orden, es diseñar un sistema apropiado para identificar los parámetros no lineales de fricción, ya que, presenta los efectos de gravedad, Inercia y Fuerzas de Coriolis [1.14], de los sistemas mecánicos, los cuales se presentan en sistemas servomecanismos en medicina, como los robots de cirugía, en sistemas de posicionamiento de antenas, en la manufactura de microchips y fibra óptica, etc.

1.4.- TEORÍA DE CONTROL

Con base al cambio de los sistemas dinámicos definidos por muchas entradas y salidas, los cuales se vuelven más complejos, la descripción de los sistemas de control moderno requiere una gran cantidad de ecuaciones. La teoría de control clásica que trata de los sistemas con una entrada y una salida, pierde su potencialidad cuando se trabaja con entradas y salidas múltiples [1.15].

La teoría de control moderna está basada en el conocimiento del comportamiento interno de los sistemas definidos por las variables componentes del mismo. Estas variables son definidas como los estados del sistema, la teoría de control moderna está basada en el conocimiento de todas las variables que intervienen en el comportamiento dinámico del sistema, lo cual permite muchas veces un control más potente y el estudio de sistemas más complejos.

Las ventajas existentes entre la teoría de control basada en retroalimentación de estados y la teoría clásica se pueden definir como:

- Este tipo de control se aplica a sistemas multivariantes en los que existe interacción entre las variables del mismo.
- La teoría moderna de control se aplica tanto a sistemas lineales como a no lineales.

1.5.- INTELIGENCIA ARTIFICIAL

La inteligencia artificial (IA) es un área multidisciplinaria que, a través de ciencias como la informática, la lógica y la filosofía, estudia la creación y diseño de entidades capaces de razonar por sí mismas utilizando como paradigma la inteligencia humana. Existen distintos tipos de percepciones y acciones, que pueden ser obtenidas y producidas, respectivamente, por sensores físicos y sensores mecánicos en máquinas, pulsos eléctricos u ópticos en computadoras, tanto como por entradas y salidas de bits de un software y su entorno software [1.16].

Los sistemas de IA actualmente son parte de la rutina en campos como economía, medicina, ingeniería y la milicia, y se ha usado en gran variedad de aplicaciones de software, juegos de estrategia, como ajedrez de computador, y otros videojuegos. Stuart Russell y Peter Norvig diferencian estos tipos de la inteligencia artificial:

- Sistemas que piensan como humanos
- Sistemas que actúan como humanos
- Sistemas que piensan racionalmente

- Sistemas que actúan racionalmente (idealmente)

En el presente capítulo se describen los algoritmos basados en inteligencia artificial bajo los cuales se generan las ganancias del sistema péndulo doble invertido:

- Algoritmos genéticos binarios
- Evolución diferencial
- Enjambre de partículas

1.6.- COMPUTACIÓN EVOLUTIVA

La computación evolutiva es una rama de la inteligencia artificial que involucra problemas de optimización combinatoria. Se inspira en los mecanismos de la Evolución biológica [1.17].

Durante los años 50 se comenzó a aplicar los principios de Charles Darwin en la resolución de problemas. Durante los años 60 y 70, varias corrientes de investigación independientes comenzaron a formar lo que ahora se conoce como computación evolutiva:

- Programación evolutiva
- Estrategias Evolutivas
- Algoritmos genético

La programación evolutiva nació en la década de 1960 y su creador fue *Lawrence J. Fogel*. Este desarrollo comenzó como un esfuerzo encaminado a crear inteligencia artificial basada en la evolución de máquinas de estado finitas. Las estrategias evolutivas fueron propuestas por *Inge Rechenberg* y *Hans-Paul Schwefel* en la década de 1970. Su principal objetivo era el de optimizar de parámetros. Los algoritmos evolutivos son métodos de optimización y búsqueda de soluciones basados en los postulados de la evolución biológica. En ellos se mantiene un conjunto de entidades que representan posibles soluciones, las cuales se mezclan, y compiten entre sí, de tal manera que las más aptas son capaces de prevalecer a lo largo del tiempo, evolucionando hacia mejores soluciones cada vez.

Los algoritmos evolutivos, y la computación evolutiva, son una rama de la inteligencia artificial. Son utilizados principalmente en problemas con espacios de búsqueda extensos y no lineales, en donde otros métodos no son capaces de encontrar soluciones en un tiempo razonable.

Siguiendo la terminología de la teoría de la evolución, las entidades que representan las soluciones al problema se denominan individuos o cromosomas, y el conjunto de estos, población.

Los individuos son modificados por operadores genéticos, principalmente el sobrecruzamiento, que consiste en la mezcla de la información de dos o más individuos; la mutación, que es un cambio aleatorio en los individuos; y la selección, consistente en la elección de los individuos que sobrevivirán y conformarán la siguiente generación. Dado que los individuos que representan las soluciones más adecuadas al problema tienen más posibilidades de sobrevivir, la población va mejorando gradualmente.

Los algoritmos evolutivos poseen diversos parámetros los cuales deben ser cuidadosamente elegidos para obtener un buen desempeño y evitar problemas tales como la Convergencia prematura. En el diagrama mostrado en la figura 1.3 se ilustra a gran escala la clasificación de los algoritmos evolutivos e inteligencia de enjambres en las cuales se puede dividir las ramas de la computación evolutiva, del mismo modo se muestran los métodos bajo los cuales se efectúa la sintonización del control.

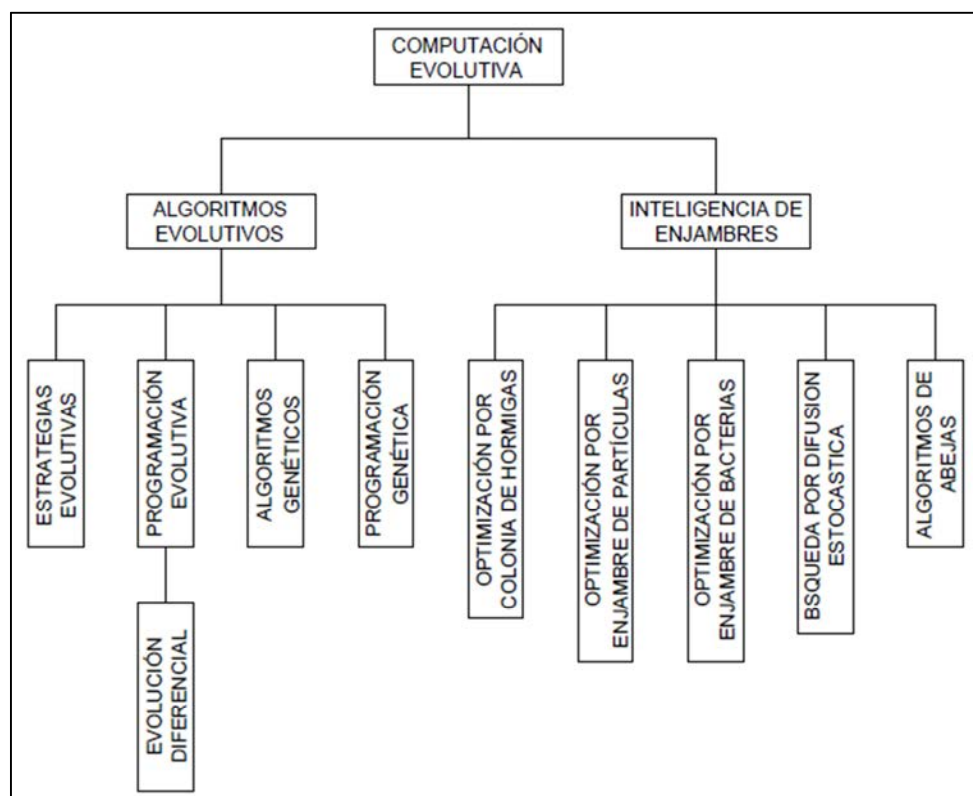


Fig. 1.3. Ramas de la computación evolutiva

1.7.- APLICACIONES DE LA COMPUTACIÓN EVOLUTIVA

Con base a la teoría expuesta anteriormente se puede observar que la computación evolutiva es una rama en creciente progreso de la ingeniería que está en constante desarrollo y evolución a lo largo del tiempo. Se listan a continuación algunas aplicaciones de la computación evolutiva [1.18].

- Diseño automatizado, incluyendo investigación en diseño de materiales y diseño multiobjetivo de componentes automovilísticos: mejor comportamiento ante choques, ahorros de peso, mejora de aerodinámica, etc.
- Diseño automatizado de equipamiento industrial.
- Diseño automatizado de sistemas de comercio en el sector financiero.
- Diseño de topologías de circuitos impresos.
- Diseño de topologías de redes computacionales.
- En Teoría de juegos, resolución de equilibrios.
- Aprendizaje de comportamiento de robots.
- Aprendizaje de reglas de Lógica difusa.
- Optimización de producción y distribución de energía eléctrica.

La aplicación desarrollada en el presente trabajo es la sintonización del control retroalimentado del sistema péndulo doble invertido, como se mencionó anteriormente.

REFERENCIAS

- [1.1] Invitation to Dynamical Systems. Edward R. Scheinerman Department of Mathematical Sciences. The Johns Hopkins University. 1996
- [1.2] Control of Nonlinear Dynamic Systems: Theory and Applications. J. K. Hedrick and A. Girard. 2010
- [1.3] Reduced-order stable controllers for two-link underactuated planar robots; Xin Xin, Yannian Liu. 9 May 2013.
- [1.4] T. Back, D. B. Fogel, and T. Michalewicz. Evolutionary Computation 1, Basic Algorithms and Operators. Institute of Physics Publishing, 2000.
- [1.5] R. Baldick. Applied Optimization: Formulation and Algorithms for Engineering Systems. Cambridge: University Press, 2009.
- [1.6] Ciencia y el hombre. Revista de divulgación científica de la universidad veracruzana. Volumen xvii. Número 3. Septiembre-diciembre 2004.

- [1.7]** Unified control of n-link underactuated manipulator with single passive joint: A reduced order approach; Xu Zhi Lai, Chang-Zhong Pan, Min Wu, Simon X. Yang. 4 July 2012.
- [1.8]** Stability analysis and time-varying walking control for an under-actuated planar biped robot; Ting Wang, Christine Chevallereau. 15 March 2011.
- [1.9]** A Planar Robot for High-Performance Manipulation. Arthur E. Quaid; The Robotics Institute; Carnegie Mellon University. Pittsburgh, Pennsylvania 2000.
- [1.10]** John H. Holland. Adaptation in Natural and Artificial Systems, An Introductory Analysis with Applications to Biology, Control and Artificial Intelligence. The MIT Press, 1975.
- [1.11]** The Forced Damped Pendulum: Chaos, Complication and Control. John H. Hubbard. Department of Mathematics - Cornell University. 2010
- [1.12]** Mark W. Spong, Seth Hutchinson, and M. Vidyasagar, Robot Modeling and Control. 2006. John Wiley.
- [1.13]** Mechanical design and control of the Pendubot daniel jerome block, m.s. department of general engineering. university of illinois at urbana-champaign, 1996. mark w. spong, adviser
- [1.14]** Inverted Pendulum Analysis, Design and Implementation. Khalil Sultan. Institute of Industrial Electronics Engineering, Karachi, Pakistan. 200
- [1.15]** Control en el espacio de estados. Sergio Dominguez. Editorial Pearson. 2012.
- [1.16]** D.E. Goldberg. Computer-aided gas pipeline operation using genetic algorithms and rule learning. PhD thesis, University of Michigan, 1983. Dissertation Abstracts International, 44(19) 1374B. University Microfilms No. 8402282.
- [1.17]** D. Goldberg, B. Korb, K. Deb. Messy genetic algorithms: Motivation, analysis, and first results. Complex Systems, 3:493--530, 1990.
- [1.18]** M. Munetomo, M. Takahashi, Y. Takai, Y. Sato. A Cooperative Search Strategy Using Hierarchical Genetic Algorithms and Its Implementation on the UNIX-Network. IPSJ SIGNotes Contents Artificial Intelligence No.091.

CAPÍTULO 2

MODELACIÓN DINÁMICA DEL PÉNDULO DOBLE INVERTIDO

Resumen:

En el presente capítulo se desarrolla la teoría y modelación del sistema dinámico péndulo doble invertido. Se desarrollan las ecuaciones rectoras de la dinámica y algunos teoremas importantes para el desarrollo del control del mismo. como es el caso de la dinámica basada en la formulación de Lagrange y la disipación de energía del sistema, con este tratamiento se presentan las ecuaciones rectoras en forma de variables de estado.

2.1.- INTRODUCCIÓN

En la Universidad de Illinois, una extensa investigación y desarrollo ha generado el concepto y diseño del sistema dinámico subactuado llamado robot planar Acrobot [2.1]. Al ampliar esta investigación en los vínculos planos subactuados, nace la idea del péndulo doble.

El péndulo doble es un homólogo del Acrobot en el concepto de que se montan sus dos enlaces verticalmente, pero en lugar de tener el accionamiento en su codo, este sistema es accionado en la articulación al hombro. Esto hace que para un diseño de control un poco más sencillo en comparación con el Acrobot, pero todos los problemas de control similares pueden ser estudiados e implementados.

El objetivo del controlador del sistema dinámico de estudio es hacer pivotar el mecanismo de su configuración estable a lazo abierto en puntos de equilibrio inestable y luego coger el enlace no accionada y balancear allí.

Un paso crucial para diseñar un control adecuado del sistema dinámico en cuestión es definir el modelo matemático del mismo lo más aproximado a la realidad, El péndulo doble invertido es un sistema que posee puntos de equilibrio, los cuales los podemos clasificar en estables e inestables.

Un modelo matemático es uno de los tipos de modelos científicos que emplea algún tipo de formulismo matemático para expresar relaciones, proposiciones sustantivas de hechos, variables, parámetros, entidades y relaciones entre variables y/o entidades u operaciones, para estudiar comportamientos de sistemas complejos ante situaciones difíciles de observar en la realidad.

En concreto en esas áreas se trabajan con "modelos formales". Un modelo formal para una cierta teoría matemática es un conjunto sobre el que se han definido un conjunto de relaciones unarias, binarias y trinarias, que satisface las proposiciones derivadas del conjunto de axiomas de la teoría. La rama de la matemática que se encarga de estudiar sistemáticamente las propiedades de los modelos es la teoría de modelos.

La fig. 2.1 ilustra el modelo en cuestión. Se comienza definiendo el péndulo doble invertido dado, que en el modelo del sistema está formado por dos péndulos unidos entre sí a través de una articulación. Los parámetros que intervienen en el modelo son los siguientes:

m_1 : Masa del eslabón 1

m_2 : Masa del eslabón 2

L_1 : Longitud del eslabón 1

L_2 : Longitud del eslabón 2

l_{c1} : Distancia del origen al centro de masa del eslabón 1

l_{c2} : Distancia del origen al centro de masa del eslabón 2

I_1 : Momento de inercia del eslabón 1 alrededor de su centroide

I_2 : Momento de inercia del eslabón 2 alrededor de su centroide

g : Aceleración de la gravedad

q_1 : Ángulo del eslabón 1 al eje horizontal

q_2 : Ángulo del eslabón 2 al eje del eslabón 1

τ_1 : Torque aplicado al eslabón 1

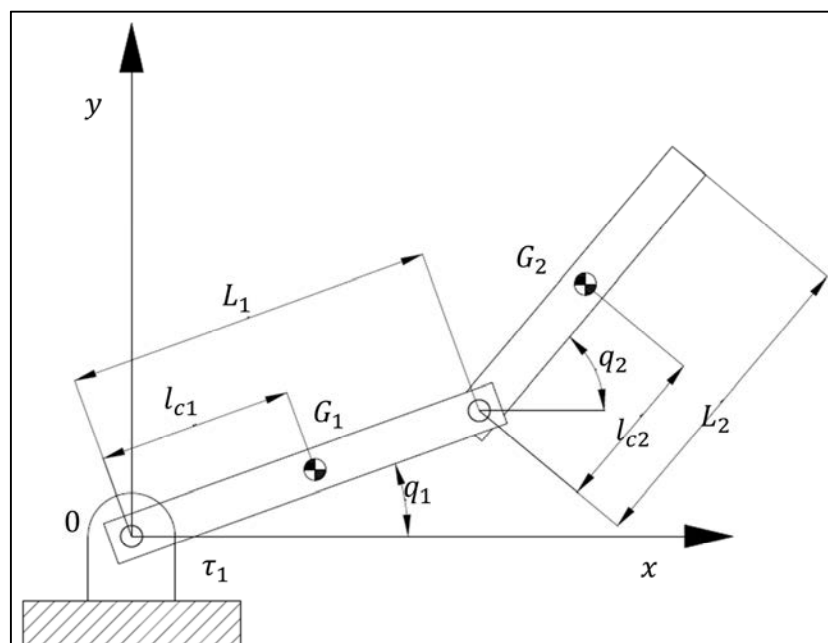


Fig. 2.1. péndulo doble invertido

Una vez definidos los parámetros que están presentes en el modelo físico del sistema es necesario comenzar con la formulación del modelo matemático.

2.2. MODELACIÓN DE SISTEMAS DINÁMICOS

El propósito del modelado matemático es representar todos los detalles importantes de los sistemas con el propósito de derivar las ecuaciones matemáticas que rigen el comportamiento del sistema.

El modelo matemático puede ser lineal o no lineal según el comportamiento de los componentes del sistema. La mayoría de los sistemas dinámicos presentes en la naturaleza son aproximaciones de modelo matemático de sistemas dinámicos eléctricos, mecánicos, térmicos y de flujo de fluidos. Los cuales, si son lineales

tienen ciertas analogías entre sí. En general se poseen dos formulaciones para determinar los modelos de los sistemas dinámicos:

- Una de las formulaciones está basada en la mecánica newtoniana, la cual es denominada métodos vectoriales.
- La segunda formulación está basada en el trabajo de J. L. Lagrange y de L. Euler, el cual es denominada métodos variacionales.

Los métodos basados en formulación newtoniana son muy utilizados para generar ecuaciones de sistemas dinámicos, sin embargo conforme el número de partículas, grados de libertad, o ligaduras del problema van aumentando resulta complicado determinar el modelo matemático que rige el comportamiento del sistema. Un camino alternativo es el llamado formulación Lagrange-Euler [2.2], el cual está basado en principios variacionales.

Los métodos variacionales surgen de la necesidad de determinar máximos o mínimos de funciones que tienen como principal característica que su dominio de definición está compuesto por funciones, y derivadas de funciones las cuales reciben el nombre de funcionales. Se llaman funcionales a las magnitudes variables cuyos valores se determinan mediante la elección de una o de varias funciones.

2.2.1 FASES DE CONSTRUCCION DE UN MODELO

En muchos casos la construcción o creación de modelos matemáticos útiles sigue una serie de fases bien determinadas:

1. Identificación de un problema o situación compleja que necesita ser simulada, optimizada o controlada y por tanto requeriría un modelo matemático predictivo.
2. Elección del tipo de modelo, esto requiere precisar qué tipo de respuesta o salida pretende obtenerse, cuales son los datos de entrada o factores relevantes, y para qué pretende usarse el modelo. Esta elección debe ser suficientemente simple como para permitir un tratamiento matemático asequible con los recursos disponibles. Esta fase requiere además identificar el mayor número de datos fidedignos, rotular y clasificar las incógnitas (variables independientes y dependientes) y establecer consideraciones, físicas, químicas, geométricas, etc. que representen adecuadamente el fenómeno en estudio.
3. Formalización del modelo en la que se detallarán qué forma tienen los datos de entrada, qué tipo de herramienta matemática se usará, como se adaptan a la información previa existente. También podría incluir la

confección de algoritmos, ensamblaje de archivos informáticos, etc. En esta fase posiblemente se introduzcan también simplificaciones suficientes para que el problema matemático de modelización sea tratable computacionalmente.

4. Comparación de resultados los resultados obtenidos como predicciones necesitan ser comparados con los hechos observados para ver si el modelo está prediciendo bien. Si los resultados no se ajustan bien, frecuentemente se vuelve a la fase 1.

Es importante mencionar que la inmensa mayoría de modelos matemáticos no son exactos y tienen un alto grado de idealización y simplificación, ya que una modelización muy exacta puede ser más complicada de tratar de una simplificación conveniente y por tanto menos útil. Es importante recordar que el mecanismo con que se desarrolla un modelo matemático repercute en el desarrollo de otras técnicas de conocimientos.

2.3.- DINÁMICA LAGRANGIANA

En física, un Lagrangiano es una función escalar a partir de la cual se pueden obtener la evolución temporal, las leyes de conservación y otras propiedades importantes de un sistema dinámico. De hecho, en física moderna el Lagrangiano se considera el operador más fundamental que describe un sistema físico.

El formalismo Lagrangiano permite alcanzar, tanto las leyes de Newton como las ecuaciones de Maxwell, los cuales pueden ser derivados como las ecuaciones de Euler-Lagrange de un Lagrangiano clásico. Igualmente la forma del Lagrangiano determina las propiedades básicas del sistema en teoría cuántica de campos.

Históricamente el formalismo Lagrangiano surgió dentro de la mecánica clásica para sistemas con un número finito de grados de libertad. Este Lagrangiano permitía escribir las ecuaciones de movimiento de un sistema totalmente general que tenía restricciones de movimiento o era no-inercial de modo muy sencillo.

Más tarde el concepto se generalizó a sistemas con un número no finito de grados de libertad como los medios continuos o los campos físicos. Más tarde el concepto pudo generalizarse también a la mecánica cuántica, particularmente en la teoría cuántica de campos.

Esta es una ecuación especializada para sistemas que tienen un número discreto de grados de libertad y que poseen dependencia de la energía cinética. La

ecuación de Lagrange produce n ecuaciones (comúnmente diferenciales) para las n coordenadas generalizadas. Es definida por la ecuación:

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{q}_i} \right) - \frac{\partial L}{\partial q_i} = Q_i$$

Para $i = 1, 2, 3, \dots$

Donde se define:

$L = T - U$: Operador Lagrangiano

T : Energía cinética total del sistema

U : Energía potencial total del sistema

q_i : Coordenada generalizada

Q_i : Fuerzas no conservativas

2.3.1.- FUERZAS DISIPATIVAS EN LA ECUACIÓN DE LAGRANGE

Si se consideran las variaciones virtuales de los desplazamientos de las coordenadas generalizadas denotados por:

$$\delta x_1, \delta x_2, \delta x_3, \dots, \delta x_n$$

Entonces el trabajo virtual $\delta\omega$ hecho por las fuerzas no conservativas en el sistema, debido a las variaciones de las coordenadas generalizadas se puede escribir como:

$$\delta\omega = \sum_{i=1}^{i=n} Q_i \delta x_i$$

Lo cual establece la relación a partir de la cual se deben identificar las fuerzas no conservativas generalizadas.

2.4.- MODELACIÓN DINÁMICA DEL PÉNDULO DOBLE INVERTIDO

Con base a la teoría expuesta anteriormente; el paso a desarrollar a continuación es modelar el comportamiento dinámico del péndulo invertido para lo cual es utilizada la teoría de Lagrange.

La energía cinética referente al eslabón 1 es definida por [2.3]:

$$K_1 = \frac{1}{2} [m_1 l_{c_1}^2 \dot{q}_1^2 (\sin^2 q_1 + \cos^2 q_1) + \gamma_{33} \dot{q}_1^2]$$

Donde:

γ_{33} = tensor de inercia resultante del eslabón 1 de acuerdo al eje de rotación z

El cual se calcula como:

$$\gamma_{33} = \frac{m_{rm}}{12} (r_{rm}^2) + \frac{m_{rs1}}{12} (r_{rs1}^2) + m_{as2} \left(\frac{13}{12} r_{as2}^2 + 2L_1 r_{as2} + L_1^2 \right) + \frac{m_{b1}}{3} (b_1^2 + L_1^2)$$

Con lo cual la energía cinética del primer eslabón toma la forma:

$$K_1 = \frac{1}{2} [I_1 \dot{q}_1^2]$$

La energía cinética referente al eslabón 2 es:

$$K_2 = \frac{1}{2} (I_2) \dot{q}_1^2 + \frac{1}{2} m_2 [L_1^2 \dot{q}_1^2 + 2L_1 l_{c_2} \dot{q}_1 \dot{q}_2 \cos(q_1 - q_2)]$$

Donde:

$$I_2 = m_2 l_{c_2}^2 + \gamma_{33}$$

Se tiene que la energía cinética total del sistema toma la forma:

$$K = K_1 + K_2$$

$$K = \frac{1}{2} (I_1 + m_2 L_1^2) \dot{q}_1^2 + m_2 L_1 l_{c_2} \dot{q}_2 \dot{q}_1 \cos(q_1 - q_2) + \frac{1}{2} I_2 \dot{q}_2^2$$

La energía potencial del primer eslabón es:

$$P_1 = m_1 g l_{c_1} \sin q_1$$

La energía potencial del segundo eslabón es:

$$P_2 = m_2 g (L_1 \sin q_1 + l_{c_2} \sin q_2)$$

La energía potencial total del sistema dinámico toma la estructura:

$$P = m_1 g l_{c_1} \sin q_1 + m_2 g (L_1 \sin q_1 + l_{c_2} \sin q_2)$$

A partir de las funciones de energía anteriores se define la función Lagrangiana del sistema como:

$$L = K - P$$

$$L = \frac{1}{2}(I_1 + m_2 L_1^2) \dot{q}_1^2 + m_2 L_1 l_{c_2} \dot{q}_2 \dot{q}_1 \cos(q_1 - q_2) + \frac{1}{2} I_2 \dot{q}_2^2 - m_1 g l_{c_1} \sin q_1 - m_2 g (L_1 \sin q_1 + l_{c_2} \sin q_2)$$

El modelo correspondiente a la ecuación de movimiento derivado con el uso de la ecuación de Lagrange:

$$\frac{d}{dt} \left\{ \frac{\partial L}{\partial \dot{q}}(q, \dot{q}) \right\} - \frac{\partial L}{\partial q}(q, \dot{q}) = \tau$$

Dónde:

$$q = (q_1, \dots, q_n)^T$$

Representa las coordenadas generalizadas, las cuales representan los grados de libertad del sistema. Las fuerzas externas aplicadas al sistema son:

$$\tau = (\tau_1, \dots, \tau_n)^T$$

En el caso de estudio se tiene:

$$q = (q_1, q_2)^T$$

$$\tau = (\tau_1, 0)^T$$

Con lo cual se tiene:

$$\begin{aligned} \left(\frac{\partial L}{\partial q_1} \right) &= -m_2 L_1 l_{c_2} \sin(q_1 - q_2) \dot{q}_2 \dot{q}_1 - g(m_1 l_{c_1} + m_2 L_1) \cos q_1 \\ \left(\frac{\partial L}{\partial q_2} \right) &= m_2 L_1 l_{c_2} \sin(q_1 - q_2) \dot{q}_2 \dot{q}_1 - m_2 g l_{c_2} \cos q_2 \\ \left(\frac{\partial L}{\partial \dot{q}_1} \right) &= \dot{q}_1 (I_1 + m_2 L_1^2) + m_2 L_1 l_{c_2} \dot{q}_2 \cos(q_1 - q_2) \\ \left(\frac{\partial L}{\partial \dot{q}_2} \right) &= \dot{q}_2 I_2 + m_2 L_1 l_{c_2} \dot{q}_1 \cos(q_1 - q_2) \end{aligned}$$

Al acoplar estas derivadas con la ecuación de Lagrange se obtiene:

$$\tau_1 = \ddot{q}_1(I_1 + m_2 L_1^2) + m_2 L_1 l_{c_2} \cos(q_1 - q_2) \ddot{q}_2 + m_2 L_1 l_{c_2} \sin(q_1 - q_2) \dot{q}_2^2 + g(m_1 l_{c_1} + m_2 L_1) \cos q_1$$

$$\tau_2 = m_2 L_1 l_{c_2} \cos(q_1 - q_2) \ddot{q}_1 + \ddot{q}_2 I_2 - m_2 L_1 l_{c_2} \sin(q_1 - q_2) \dot{q}_1^2 + m_2 g l_{c_2} \cos q_2$$

Las anteriores ecuaciones diferenciales son las que determinan el comportamiento del sistema dinámico de péndulo doble. Definiendo el siguiente cambio de variables:

$$\begin{aligned}\theta_1 &= I_1 + m_2 L_1^2 \\ \theta_2 &= I_2 \\ \theta_3 &= m_2 L_1 l_{c_2} \\ \theta_4 &= g(m_1 l_{c_1} + m_2 L_1) \\ \theta_5 &= m_2 g l_{c_2}\end{aligned}$$

Se obtiene un sistema dinámico de la forma:

$$\tau_1 = \ddot{q}_1 \theta_1 + \theta_3 \cos(q_1 - q_2) \ddot{q}_2 + \theta_3 \sin(q_1 - q_2) \dot{q}_2^2 + \theta_4 \cos q_1$$

$$\tau_2 = \theta_3 \cos(q_1 - q_2) \ddot{q}_1 + \ddot{q}_2 \theta_2 - \theta_3 \sin(q_1 - q_2) \dot{q}_1^2 + \theta_5 \cos q_2$$

2.5.- ECUACIONES DE ESTADO DEL MODELO DINÁMICO DEL PÉNDULO DOBLE INVERTIDO

Una vez definidas las ecuaciones que gobiernan el comportamiento dinámico del sistema, estas son expresadas en ecuaciones de estado. Dado que con esta formulación se logra disminuir el orden de las ecuaciones diferenciales de estudio y con ello se facilita la solución del sistema. Sea:

$$\begin{aligned}q_1 &= x_1 \\ q_2 &= x_2 \\ \dot{q}_1 &= x_3 \\ \dot{q}_2 &= x_4\end{aligned}$$

Las variables de estado del sistema, de tal manera que sustituyendo en las ecuaciones del modelo se tienen:

$$\begin{aligned}\dot{x}_1 &= x_3 \\ \dot{x}_2 &= x_4 \\ \tau_1 &= \dot{x}_3 \theta_1 + \theta_3 \cos(x_1 - x_2) \dot{x}_4 + \theta_3 \sin(x_1 - x_2) x_4^2 + \theta_4 \cos x_1 \\ \tau_2 &= \theta_3 \cos(x_1 - x_2) \dot{x}_3 + \dot{x}_4 \theta_2 - \theta_3 \sin(x_1 - x_2) x_3^2 + \theta_5 \cos x_2\end{aligned}$$

De tal manera que el arreglo matricial del sistema queda definido por:

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & h_1 & h_2 \\ 0 & 0 & h_3 & h_4 \end{bmatrix} \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \end{bmatrix} = \begin{bmatrix} x_3 \\ x_4 \\ -\theta_3 \sin(x_1 - x_2) x_4^2 - \theta_4 \cos x_1 \\ \theta_3 \sin(x_1 - x_2) x_3^2 - \theta_5 \cos x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \tau_1 \\ \tau_2 \end{bmatrix}$$

Dónde:

$$\begin{bmatrix} h_1 \\ h_2 \\ h_3 \\ h_4 \end{bmatrix} = \begin{bmatrix} \theta_1 \\ \theta_3 \cos(x_1 - x_2) \\ \theta_3 \cos(x_1 - x_2) \\ \theta_2 \end{bmatrix}$$

Es decir:

$$[A(X)][\dot{X}] = G(X) + F(\tau)$$

Dada la naturaleza de las ecuaciones de estudio, no existen métodos analíticos con los cuales se determinen soluciones para estas. Se debe recurrir a métodos de solución numéricos, estos métodos reciben el nombre de métodos de Runge-Kutta.

REFERENCIAS

- [2.1]** Control of Nonlinear Dynamical Systems. Methods and Applications. Felix L. Chernousko · Igor M. Ananievski. Sergey A. Reshmin. Springer.
- [2.2]** Modeling, Simulation and Control of Nonlinear Engineering Dynamical Systems State-of-the-Art, Perspectives and Applications. Jan Awrejcewicz. Springer.
- [2.3]** Identificación de Parámetros no Lineales de Fricción del Sistema Pendubot Usando Algoritmos Evolutivos. Pedro Lagos Eulogio. Diciembre de 2012. Centro de Investigación Avanzada en Ingeniería Industrial

CAPÍTULO 3

CONTROL DE POSICIÓN DEL PÉNDULO DOBLE POR RETROALIMENTACIÓN DE ESTADOS

Resumen:

En el presente capítulo se desarrolla el controlador del sistema dinámico péndulo doble por métodos convencionales, se utiliza el método de retroalimentación de estados por asignación de polos del sistema dinámico en su forma lineal, primeramente el controlador es desarrollado tomando en cuenta el modelo dinámico linealizado en un punto de equilibrio para posteriormente realizar la acción de control en el punto de equilibrio del sistema no lineal. Para efectuar el tratamiento anterior se describe mediante simulación computacional las propiedades mecánicas de un prototipo de estudio, las cuales son introducidas en el modelo matemático para su análisis.

3.1.- INTRODUCCIÓN

En el estudio del péndulo doble invertido en cuestión es necesario y muy recomendable primeramente efectuar un análisis del comportamiento del sistema de estudio por métodos convencionales basados en el modelo dinámico lineal del mismo.

Esta primera aproximación muestra información sobre el comportamiento en puntos de equilibrio establecidos del sistema, esto es importante porque permite tener en una primera instancia la robustez del sistema y el tiempo de establecimiento del mismo. Dado que en un punto de equilibrio es bajo el cual se desea establecer el control del sistema dinámico.

En base al sistema lineal se puede extender el estudio del sistema dinámico al caso no lineal donde, bajo el punto de equilibrio se analiza el comportamiento del mismo con las ganancias definidas.

Posteriormente con el análisis anterior y bajo el punto de equilibrio de estudio se procederá a analizar el comportamiento del sistema en forma no lineal para que mediante técnicas basadas en computación evolutiva se generen las ganancias del mismo y se proceda a determinar las mejores del mismo.

3.2.- PUNTOS CRÍTICOS DE EQUILIBRIO DEL SISTEMA

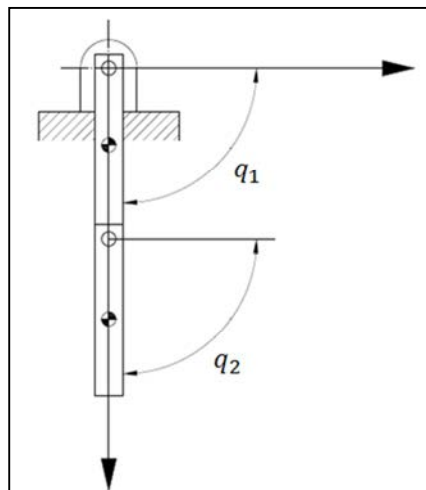


Fig. 3.1. Posición de equilibrio estable del péndulo doble invertido

Una vez establecidos los criterios de estabilidad de un sistema dinámico y especificadas las técnicas bajo las cuales se puede linealizar el mismo (en caso de ser no lineal), es deseable determinar puntos bajo los cuales se desarrollará la linealización del sistema, estos puntos correspondientes al comportamiento

dinámico del sistema son puntos críticos, los cuales pueden ser estables o inestables, los cuales bajo la adecuada manipulación se pueden volver estables. Los puntos críticos de un sistema dinámico reciben vital importancia dado que representan soluciones bajo las cuales un sistema puede poseer atractores naturales y también puntos inestables de comportamiento.

El punto de equilibrio estable corresponde a un atractor natural del mismo localizado en la posición ilustrada en la figura 3.1. El punto de equilibrio inestable es ilustrado en la figura 3.2.

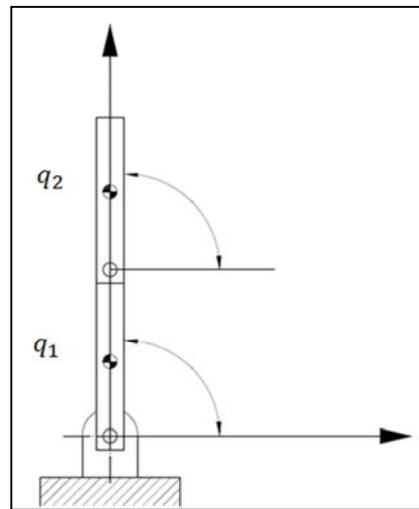


Fig. 3.2. Posición de equilibrio inestable del péndulo doble invertido

Matemáticamente un punto crítico de una función es cualquier valor en el dominio en donde la función no es diferenciable o cuando su derivada es 0. Un punto de equilibrio del sistema dinámico representa las condiciones de las variables del sistema, en donde este se encuentra estático. Por ejemplo, en el caso de una partícula si se encuentra en reposo sin alguna fuerza externa que representa una entrada, entonces se dice que se encuentra en un punto de equilibrio. Entonces, un punto de equilibrio está dado en:

$$x' = f(x_0, y_0) \equiv 0$$

Con el uso de la ecuación anterior se pueden determinar los puntos críticos para posteriormente utilizar criterios de linealización y estabilización del sistema dinámico del péndulo doble invertido.

El problema de control consiste en lograr que el péndulo doble invertido que posee un atractor natural como el mostrado en la figura 3.3 permanezca en la posición de equilibrio inestable mostrado en la figura 3.4 por medio de las metodologías de

control en variables de estados convencionales, en el presente trabajo se define próximamente el punto de equilibrio de control.

3.3.- DESARROLLO DE CONTROL PARA EL SISTEMA DE PÉNDULO DOBLE INVERTIDO LINEAL

Una vez desarrollada y analizada la teoría referente al control tradicional de sistemas dinámicos se procede a desarrollar la misma en el sistema dinámico de estudio. El desarrollo de la acción de control basada en el método tradicional de asignación de polos se llevará a cabo primeramente mediante la definición del sistema dinámico del péndulo invertido en variables de estado, se determinan los puntos críticos del sistema para posteriormente realizar la linealización en algún punto crítico de interés, se desarrolla la prueba de controlabilidad y observabilidad del sistema, se desarrollan el método de asignación de polos para estabilizar el sistema dinámico y finalmente se desarrolla el control del sistema en una referencia dada.

3.3.1.- MODELO DINÁMICO EN VARIABLES DE ESTADO

El modelo dinámico del péndulo doble invertido es descrito por las ecuaciones obtenidas en el capítulo anterior, las cuales toman la estructura:

$$\tau_1 = \ddot{q}_1 \theta_1 + \theta_3 \cos(q_1 - q_2) \ddot{q}_2 + \theta_3 \sin(q_1 - q_2) \dot{q}_2^2 + \theta_4 \cos q_1$$

$$\tau_2 = \theta_3 \cos(q_1 - q_2) \ddot{q}_1 + \ddot{q}_2 \theta_2 - \theta_3 \sin(q_1 - q_2) \dot{q}_1^2 + \theta_5 \cos q_2$$

Las cuales, una vez definidas, son reescritas utilizando el siguiente cambio de variables:

$$q_1 = x_1$$

$$q_2 = x_2$$

$$\dot{q}_1 = x_3$$

$$\dot{q}_2 = x_4$$

Toman la siguiente estructura:

$$\dot{x}_1 = x_3$$

$$\dot{x}_2 = x_4$$

$$\tau_1 = \dot{x}_3 \theta_1 + \theta_3 \cos(x_1 - x_2) \dot{x}_4 + \theta_3 \sin(x_1 - x_2) x_4^2 + \theta_4 \cos x_1$$

$$\tau_2 = \theta_3 \cos(x_1 - x_2) \dot{x}_3 + \dot{x}_4 \theta_2 - \theta_3 \sin(x_1 - x_2) x_3^2 + \theta_5 \cos x_2$$

El sistema de ecuaciones diferenciales anterior es el denominado en espacio de estados del péndulo doble invertido, el cual es el punto de partida para el desarrollo del controlador.

3.3.2.- ANÁLISIS DE PUNTOS CRÍTICOS DEL SISTEMA DINÁMICO

El sistema dinámico anteriormente obtenido expresado puede ser reescrito como:

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & h_1 & h_2 \\ 0 & 0 & h_3 & h_4 \end{bmatrix} \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \end{bmatrix} = \begin{bmatrix} x_3 \\ x_4 \\ -\theta_3 \sin(x_1 - x_2)x_4^2 - \theta_4 \cos x_1 \\ \theta_3 \sin(x_1 - x_2)x_3^2 - \theta_5 \cos x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \tau_1 \\ \tau_2 \end{bmatrix}$$

Dónde definiendo el siguiente vector de constantes se tiene la representación anteriormente expuesta:

$$\begin{bmatrix} h_1 \\ h_2 \\ h_3 \\ h_4 \end{bmatrix} = \begin{bmatrix} \theta_1 \\ \theta_3 \cos(x_1 - x_2) \\ \theta_3 \cos(x_1 - x_2) \\ \theta_2 \end{bmatrix}$$

Este sistema de ecuaciones debe establecerse en su forma autónoma partiendo del sistema para determinar primeramente los puntos críticos del sistema dinámico de estudio:

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & h_1 & h_2 \\ 0 & 0 & h_3 & h_4 \end{bmatrix} \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \end{bmatrix} = \begin{bmatrix} x_3 \\ x_4 \\ -\theta_3 \sin(x_1 - x_2)x_4^2 - \theta_4 \cos x_1 \\ \theta_3 \sin(x_1 - x_2)x_3^2 - \theta_5 \cos x_2 \end{bmatrix}$$

La forma matricial del sistema anterior puede definirse como:

$$H(X)[\dot{X}] = G(X)$$

Dónde:

$$G(X) = \begin{bmatrix} x_3 \\ x_4 \\ -\theta_3 \sin(x_1 - x_2)x_4^2 - \theta_4 \cos x_1 \\ \theta_3 \sin(x_1 - x_2)x_3^2 - \theta_5 \cos x_2 \end{bmatrix}$$

La forma autónoma del sistema es definida entonces por la ecuación diferencial matricial:

$$[\dot{X}] = F(X)$$

Dónde:

$$F(X) = [H(X)]^{-1}G(X)$$

Para determinar los puntos críticos es necesario resolver la ecuación vectorial:

$$F(X) = [H(X)]^{-1}G(X) = 0$$

Se sabe que:

$$H(X) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & h_1 & h_2 \\ 0 & 0 & h_3 & h_4 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & \theta_1 & \theta_3 \cos(x_1 - x_2) \\ 0 & 0 & \theta_3 \cos(x_1 - x_2) & \theta_2 \end{bmatrix}$$

Con base a la cual se calcula la inversa de esta matriz para posteriormente efectuar el producto matricial buscado, la matriz inversa de A(X) toma la forma:

$$[H(X)]^{-1} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & \left[\frac{\theta_2}{\theta_1 \theta_2 - \theta_3^2 \cos^2(x_1 - x_2)} \right] & \left[\frac{-\theta_3 \cos(x_1 - x_2)}{\theta_1 \theta_2 - \theta_3^2 \cos^2(x_1 - x_2)} \right] \\ 0 & 0 & \left[\frac{-\theta_3 \cos(x_1 - x_2)}{\theta_1 \theta_2 - \theta_3^2 \cos^2(x_1 - x_2)} \right] & \left[\frac{\theta_1}{\theta_1 \theta_2 - \theta_3^2 \cos^2(x_1 - x_2)} \right] \end{bmatrix}$$

Con el cálculo de esta matriz inversa se procede a efectuar el producto matricial anteriormente expresado como:

$$[H(X)]^{-1} G(X) = \begin{bmatrix} x_3 \\ x_4 \\ \frac{[\theta_3 \cos(x_1 - x_2)][-\theta_3 x_3^2 \sin(x_1 - x_2) + \theta_5 \cos(x_2)]}{\theta_1 \theta_2 - \theta_3^2 \cos^2(x_1 - x_2)} - \frac{\theta_2 [\theta_3 x_4^2 \sin(x_1 - x_2) + \theta_4 \cos(x_1)]}{\theta_1 \theta_2 - \theta_3^2 \cos^2(x_1 - x_2)} \\ \frac{[\theta_3 \cos(x_1 - x_2)][\theta_3 x_4^2 \sin(x_1 - x_2) + \theta_4 \cos(x_1)]}{\theta_1 \theta_2 - \theta_3^2 \cos^2(x_1 - x_2)} - \frac{\theta_1 [-\theta_3 x_3^2 \sin(x_1 - x_2) + \theta_5 \cos(x_2)]}{\theta_1 \theta_2 - \theta_3^2 \cos^2(x_1 - x_2)} \end{bmatrix}$$

Con base al sistema anterior se establece la condición necesaria para obtener los puntos críticos de un sistema, es decir:

$$\begin{bmatrix} x_3 \\ x_4 \\ \frac{[\theta_3 \cos(x_1 - x_2)][-\theta_3 x_3^2 \sin(x_1 - x_2) + \theta_5 \cos(x_2)]}{\theta_1 \theta_2 - \theta_3^2 \cos^2(x_1 - x_2)} - \frac{\theta_2 [\theta_3 x_4^2 \sin(x_1 - x_2) + \theta_4 \cos(x_1)]}{\theta_1 \theta_2 - \theta_3^2 \cos^2(x_1 - x_2)} \\ \frac{[\theta_3 \cos(x_1 - x_2)][\theta_3 x_4^2 \sin(x_1 - x_2) + \theta_4 \cos(x_1)]}{\theta_1 \theta_2 - \theta_3^2 \cos^2(x_1 - x_2)} - \frac{\theta_1 [-\theta_3 x_3^2 \sin(x_1 - x_2) + \theta_5 \cos(x_2)]}{\theta_1 \theta_2 - \theta_3^2 \cos^2(x_1 - x_2)} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

Con base a lo cual se establece que las funciones que determinan la localización de los puntos críticos del sistema dinámico son:

$$\begin{aligned} x_3 &= 0 \\ x_4 &= 0 \\ \left\{ \frac{[\theta_3 \cos(x_1 - x_2)][-\theta_3 x_3^2 \sin(x_1 - x_2) + \theta_5 \cos(x_2)]}{\theta_1 \theta_2 - \theta_3^2 \cos^2(x_1 - x_2)} - \frac{\theta_2 [\theta_3 x_4^2 \sin(x_1 - x_2) + \theta_4 \cos(x_1)]}{\theta_1 \theta_2 - \theta_3^2 \cos^2(x_1 - x_2)} \right\} &= 0 \\ \left\{ \frac{[\theta_3 \cos(x_1 - x_2)][\theta_3 x_4^2 \sin(x_1 - x_2) + \theta_4 \cos(x_1)]}{\theta_1 \theta_2 - \theta_3^2 \cos^2(x_1 - x_2)} - \frac{\theta_1 [-\theta_3 x_3^2 \sin(x_1 - x_2) + \theta_5 \cos(x_2)]}{\theta_1 \theta_2 - \theta_3^2 \cos^2(x_1 - x_2)} \right\} &= 0 \end{aligned}$$

Con base a este sistema de ecuaciones sustituyendo los valores de las variables anteriormente definidos se determinan los puntos críticos siguientes:

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} \frac{\pi}{2} \\ 0 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} -\frac{\pi}{2} \\ 0 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} \frac{\pi}{2} \\ \pi \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} -\frac{\pi}{2} \\ \pi \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} \frac{\pi}{2} \\ -\pi \\ 0 \\ 0 \end{bmatrix}$$

El objetivo del presente trabajo es linealizar alrededor del punto de equilibrio inestable determinado por:

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} -\frac{\pi}{2} \\ \pi \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} \frac{3\pi}{2} \\ \pi \\ 0 \\ 0 \end{bmatrix}$$

El cual corresponde a un punto inestable de equilibrio como el que se ilustra en la figura 3.3.

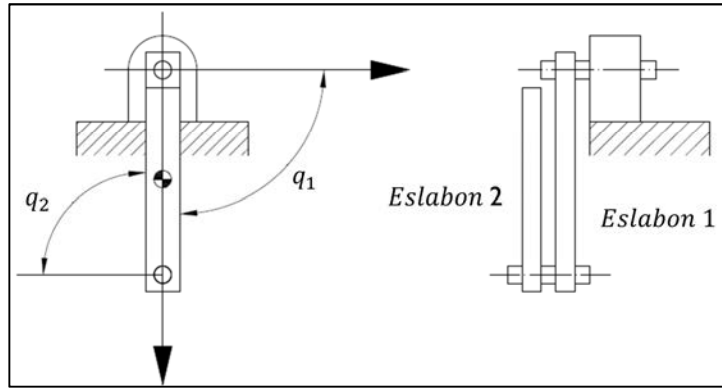


Fig. 3.3. Punto de equilibrio para efectuar el control

3.3.3.- LINEALIZACIÓN DEL SISTEMA DINÁMICO

Se definió anteriormente el producto matricial:

$$F(X) = [H(X)]^{-1}G(X)$$

$$= \begin{bmatrix} \frac{[\theta_3 \cos(x_1 - x_2)][-\theta_3 x_3^2 \sin(x_1 - x_2) + \theta_5 \cos(x_2)]}{\theta_1 \theta_2 - \theta_3^2 \cos^2(x_1 - x_2)} - \frac{\theta_2 [\theta_3 x_4^2 \sin(x_1 - x_2) + \theta_4 \cos(x_1)]}{\theta_1 \theta_2 - \theta_3^2 \cos^2(x_1 - x_2)} \\ \frac{[\theta_3 \cos(x_1 - x_2)][\theta_3 x_4^2 \sin(x_1 - x_2) + \theta_4 \cos(x_1)]}{\theta_1 \theta_2 - \theta_3^2 \cos^2(x_1 - x_2)} - \frac{\theta_1 [-\theta_3 x_3^2 \sin(x_1 - x_2) + \theta_5 \cos(x_2)]}{\theta_1 \theta_2 - \theta_3^2 \cos^2(x_1 - x_2)} \end{bmatrix}$$

Y ahora se define el producto:

$$T_1(X) = [H(X)]^{-1}T_0(X) = \begin{bmatrix} 0 \\ 0 \\ \frac{\theta_2\tau_1}{\theta_1\theta_2 - \theta_3^2\cos^2(x_1 - x_2)} - \frac{\theta_3\tau_2\cos(x_1 - x_2)}{\theta_1\theta_2 - \theta_3^2\cos^2(x_1 - x_2)} \\ \frac{\theta_1\tau_2}{\theta_1\theta_2 - \theta_3^2\cos^2(x_1 - x_2)} - \frac{\theta_3\tau_1\cos(x_1 - x_2)}{\theta_1\theta_2 - \theta_3^2\cos^2(x_1 - x_2)} \end{bmatrix}$$

Dónde:

$$T_0(X) = \begin{bmatrix} 0 \\ 0 \\ \tau_1 \\ \tau_2 \end{bmatrix}$$

De tal forma que el sistema dinámico no lineal del péndulo doble invertido toma la forma:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \end{bmatrix} = \begin{bmatrix} x_3 \\ x_4 \\ \frac{[\theta_3\cos(x_1 - x_2)][-\theta_3x_3^2\sin(x_1 - x_2) + \theta_5\cos(x_2)]}{\theta_1\theta_2 - \theta_3^2\cos^2(x_1 - x_2)} - \frac{\theta_2[\theta_3x_4^2\sin(x_1 - x_2) + \theta_4\cos(x_1)]}{\theta_1\theta_2 - \theta_3^2\cos^2(x_1 - x_2)} \\ \frac{[\theta_3\cos(x_1 - x_2)][\theta_3x_4^2\sin(x_1 - x_2) + \theta_4\cos(x_1)]}{\theta_1\theta_2 - \theta_3^2\cos^2(x_1 - x_2)} - \frac{\theta_1[-\theta_3x_3^2\sin(x_1 - x_2) + \theta_5\cos(x_2)]}{\theta_1\theta_2 - \theta_3^2\cos^2(x_1 - x_2)} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \frac{\theta_2\tau_1}{\theta_1\theta_2 - \theta_3^2\cos^2(x_1 - x_2)} - \frac{\theta_3\tau_2\cos(x_1 - x_2)}{\theta_1\theta_2 - \theta_3^2\cos^2(x_1 - x_2)} \\ \frac{\theta_1\tau_2}{\theta_1\theta_2 - \theta_3^2\cos^2(x_1 - x_2)} - \frac{\theta_3\tau_1\cos(x_1 - x_2)}{\theta_1\theta_2 - \theta_3^2\cos^2(x_1 - x_2)} \end{bmatrix}$$

El primer paso en el desarrollo de la linealización alrededor del punto de equilibrio es definir el modelo de linealización el cual toma la estructura:

$$[\dot{X}] = F(X) + T_1(X)$$

Donde se plantea la aproximación lineal a partir del punto de equilibrio crítico deseado para cada uno de los elementos presentes en las ecuaciones no lineales que en este caso corresponden a la ecuación 2 y 4 del sistema no lineal descrito por:

$$F(X) + T_1(X)$$

Con base a lo cual se establecen las aproximaciones lineales a partir de la serie de Taylor de las funciones no lineales presentes:

$$F(X_3) = \frac{[\theta_3\cos(x_1 - x_2)][-\theta_3x_3^2\sin(x_1 - x_2) + \theta_5\cos(x_2)]}{\theta_1\theta_2 - \theta_3^2\cos^2(x_1 - x_2)} - \frac{\theta_2[\theta_3x_4^2\sin(x_1 - x_2) + \theta_4\cos(x_1)]}{\theta_1\theta_2 - \theta_3^2\cos^2(x_1 - x_2)}$$

$$F(X_4) = \frac{[\theta_3 \cos(x_1 - x_2)][\theta_3 x_4^2 \sin(x_1 - x_2) + \theta_4 \cos(x_1)]}{\theta_1 \theta_2 - \theta_3^2 \cos^2(x_1 - x_2)} - \frac{\theta_1 [-\theta_3 x_3^2 \sin(x_1 - x_2) + \theta_5 \cos(x_2)]}{\theta_1 \theta_2 - \theta_3^2 \cos^2(x_1 - x_2)}$$

$$T_1(X_3) = \frac{\theta_2 \tau_1}{\theta_1 \theta_2 - \theta_3^2 \cos^2(x_1 - x_2)} - \frac{\theta_3 \tau_2 \cos(x_1 - x_2)}{\theta_1 \theta_2 - \theta_3^2 \cos^2(x_1 - x_2)}$$

$$T_1(X_4) = \frac{\theta_1 \tau_2}{\theta_1 \theta_2 - \theta_3^2 \cos^2(x_1 - x_2)} - \frac{\theta_3 \tau_1 \cos(x_1 - x_2)}{\theta_1 \theta_2 - \theta_3^2 \cos^2(x_1 - x_2)}$$

Las aproximaciones lineales son definidas como:

$$F_A(X_3) = F(X_{03}) + \sum_{i=1}^{i=4} F(X_{03})_{xi} [x_i - x_{0i}]$$

$$F_A(X_4) = F(X_{04}) + \sum_{i=1}^{i=4} F(X_{04})_{xi} [x_i - x_{0i}]$$

$$T_A(X_3)_1 = T(X_{03})_1 + \sum_{i=1}^{i=4} T(X_{03})_{xi} [x_i - x_{0i}]$$

$$T_A(X_4)_1 = T(X_{04})_1 + \sum_{i=1}^{i=4} T(X_{04})_{xi} [x_i - x_{0i}]$$

Se determinan las derivadas de cada una de las funciones involucradas en las aproximaciones lineales comenzando con la función $T_1(X_3)$ con respecto a cada una de las coordenadas generalizadas.

Con respecto a x_1 se tiene:

$$T_1(X_3)_{x1} = \frac{\tau_2 \theta_3 \sin(x_1 - x_2)}{\theta_1 \theta_2 - \theta_3^2 \cos^2(x_1 - x_2)} + \frac{2\tau_2 \theta_3^2 \cos^2(x_1 - x_2) \sin(x_1 - x_2)}{[\theta_1 \theta_2 - \theta_3^2 \cos^2(x_1 - x_2)]^2}$$

$$- \frac{2\tau_1 \theta_2 \theta_3^2 \cos(x_1 - x_2) \sin(x_1 - x_2)}{[\theta_1 \theta_2 - \theta_3^2 \cos^2(x_1 - x_2)]^2}$$

Con respecto a x_2 se tiene:

$$T_1(X_3)_{x2} = \frac{2\tau_1 \theta_2 \theta_3^2 \cos(x_1 - x_2) \sin(x_1 - x_2)}{[\theta_1 \theta_2 - \theta_3^2 \cos^2(x_1 - x_2)]^2} - \frac{2\tau_2 \theta_3^2 \cos^2(x_1 - x_2) \sin(x_1 - x_2)}{[\theta_1 \theta_2 - \theta_3^2 \cos^2(x_1 - x_2)]^2}$$

$$- \frac{\tau_2 \theta_3 \sin(x_1 - x_2)}{\theta_1 \theta_2 - \theta_3^2 \cos^2(x_1 - x_2)}$$

Con respecto a x_3, x_4 se tiene:

$$T_1(X_3)_{x3} = T_1(X_3)_{x4} = 0$$

Continuando con la evaluación de la segunda función no lineal y obteniendo sus derivadas se tiene con respecto a x_1 :

$$T_1(X_4)_{x_1} = \frac{\tau_1 \theta_3 \sin(x_1 - x_2)}{\theta_1 \theta_2 - \theta_3^2 \cos^2(x_1 - x_2)} + \frac{2\tau_1 \theta_3^2 \cos^2(x_1 - x_2) \sin(x_1 - x_2)}{[\theta_1 \theta_2 - \theta_3^2 \cos^2(x_1 - x_2)]^2} - \frac{2\tau_2 \theta_2 \theta_3^2 \cos(x_1 - x_2) \sin(x_1 - x_2)}{[\theta_1 \theta_2 - \theta_3^2 \cos^2(x_1 - x_2)]^2}$$

Con respecto a x_2 se tiene:

$$T_1(X_4)_{x_2} = \frac{2\tau_2 \theta_2 \theta_3^2 \cos(x_1 - x_2) \sin(x_1 - x_2)}{[\theta_1 \theta_2 - \theta_3^2 \cos^2(x_1 - x_2)]^2} - \frac{2\tau_1 \theta_3^2 \cos^2(x_1 - x_2) \sin(x_1 - x_2)}{[\theta_1 \theta_2 - \theta_3^2 \cos^2(x_1 - x_2)]^2} - \frac{\tau_1 \theta_3 \sin(x_1 - x_2)}{\theta_1 \theta_2 - \theta_3^2 \cos^2(x_1 - x_2)}$$

Con respecto a x_3, x_4 se tiene:

$$T_1(X_4)_{x_3} = T_1(X_4)_{x_4} = 0$$

Las derivadas de la tercera ecuación correspondiente a la función $F(X)$ son calculadas a continuación:

Con respecto a x_1 se tiene:

$$F(X_3)_{x_1} = \frac{\theta_2 [-\theta_3 \cos(x_1 - x_2) x_4^2 + \theta_4 \sin x_1]}{\theta_1 \theta_2 - \theta_3^2 \cos^2(x_1 - x_2)} - \frac{\theta_3 x_3^2 \cos^2(x_1 - x_2)}{\theta_1 \theta_2 - \theta_3^2 \cos^2(x_1 - x_2)} - \frac{\theta_3 \sin(x_1 - x_2) [-\theta_3 \sin(x_1 - x_2) x_3^2 + \theta_5 \cos x_2]}{\theta_1 \theta_2 - \theta_3^2 \cos^2(x_1 - x_2)} - \frac{2\theta_3^2 \cos^2(x_1 - x_2) \sin(x_1 - x_2) [-\theta_3 x_3^2 \sin(x_1 - x_2) + \theta_5 \cos x_2]}{[\theta_1 \theta_2 - \theta_3^2 \cos^2(x_1 - x_2)]^2} + \frac{2\theta_2 \theta_3^2 \cos(x_1 - x_2) \sin(x_1 - x_2) [\theta_3 x_4^2 \sin(x_1 - x_2) + \theta_4 \cos x_1]}{[\theta_1 \theta_2 - \theta_3^2 \cos^2(x_1 - x_2)]^2}$$

Con respecto a x_2 se tiene:

$$F(X_3)_{x_2} = \frac{\theta_3 \sin(x_1 - x_2) [-\theta_3 x_3^2 \sin(x_1 - x_2) + \theta_5 \cos x_2]}{\theta_1 \theta_2 - \theta_3^2 \cos^2(x_1 - x_2)} - \frac{\theta_3 \cos(x_1 - x_2) [-\theta_3 x_3^2 \cos(x_1 - x_2) + \theta_5 \sin x_2]}{\theta_1 \theta_2 - \theta_3^2 \cos^2(x_1 - x_2)} + \frac{2\theta_3^2 \cos^2(x_1 - x_2) \sin(x_1 - x_2) [-\theta_3 x_3^2 \sin(x_1 - x_2) + \theta_5 \cos x_2]}{[\theta_1 \theta_2 - \theta_3^2 \cos^2(x_1 - x_2)]^2} + \frac{\theta_2 \theta_3 x_4^2 \cos(x_1 - x_2)}{\theta_1 \theta_2 - \theta_3^2 \cos^2(x_1 - x_2)} - \frac{2\theta_2 \theta_3^2 \cos(x_1 - x_2) \sin(x_1 - x_2) [\theta_3 \sin(x_1 - x_2) x_4^2 + \theta_4 \cos x_1]}{[\theta_1 \theta_2 - \theta_3^2 \cos^2(x_1 - x_2)]^2}$$

Con respecto a x_3 se tiene:

$$F(X_3)_{x_3} = \frac{-2\theta_3^2 x_3 \cos(x_1 - x_2) \sin(x_1 - x_2)}{\theta_1 \theta_2 - \theta_3^2 \cos^2(x_1 - x_2)}$$

Con respecto a x_4 se tiene:

$$F(X_3)_{x_4} = \frac{-2\theta_2 \theta_3 x_4 \sin(x_1 - x_2)}{\theta_1 \theta_2 - \theta_3^2 \cos^2(x_1 - x_2)}$$

Finalmente las derivadas de la cuarta ecuación correspondiente a la función $F(X)$ son determinadas con respecto a cada una de las variables de estado.

Con respecto a x_1 se tiene:

$$\begin{aligned} F(X_4)_{x_1} = & \frac{\theta_1 \theta_3 x_3^2 \cos(x_1 - x_2)}{\theta_1 \theta_2 - \theta_3^2 \cos^2(x_1 - x_2)} - \frac{\theta_3 \sin(x_1 - x_2) [\theta_3 x_4^2 \sin(x_1 - x_2) + \theta_4 \cos x_1]}{\theta_1 \theta_2 - \theta_3^2 \cos^2(x_1 - x_2)} \\ & - \frac{2\theta_3^2 x_3 \cos^2(x_1 - x_2) \sin(x_1 - x_2) [\theta_3 x_4^2 \sin(x_1 - x_2) + \theta_4 \cos x_1]}{\theta_1 \theta_2 - \theta_3^2 \cos^2(x_1 - x_2)} \\ & - \frac{\theta_3 \cos(x_1 - x_2) [-\theta_3 x_4^2 \cos(x_1 - x_2) + \theta_4 \sin x_1]}{\theta_1 \theta_2 - \theta_3^2 \cos^2(x_1 - x_2)} \\ & + \frac{2\theta_1 \theta_3^2 x_3 \cos(x_1 - x_2) \sin(x_1 - x_2) [-\theta_3 x_3^2 \sin(x_1 - x_2) + \theta_5 \cos x_2]}{[\theta_1 \theta_2 - \theta_3^2 \cos^2(x_1 - x_2)]^2} \end{aligned}$$

Con respecto a x_2 se tiene:

$$\begin{aligned} F(X_4)_{x_2} = & \frac{\theta_1 [-\theta_3 x_3^2 \cos(x_1 - x_2) + \theta_5 \sin x_2]}{\theta_1 \theta_2 - \theta_3^2 \cos^2(x_1 - x_2)} - \frac{\theta_3^2 x_4^2 \cos^2(x_1 - x_2)}{\theta_1 \theta_2 - \theta_3^2 \cos^2(x_1 - x_2)} \\ & + \frac{\theta_3 \sin(x_1 - x_2) [\theta_3 x_4^2 \sin(x_1 - x_2) + \theta_4 \cos x_1]}{\theta_1 \theta_2 - \theta_3^2 \cos^2(x_1 - x_2)} \\ & + \frac{2\theta_3^2 x_3 \cos^2(x_1 - x_2) \sin(x_1 - x_2) [\theta_3 x_4^2 \sin(x_1 - x_2) + \theta_4 \cos x_1]}{[\theta_1 \theta_2 - \theta_3^2 \cos^2(x_1 - x_2)]^2} \\ & - \frac{2\theta_1 \theta_3^2 \cos(x_1 - x_2) \sin(x_1 - x_2) [-\theta_3 x_3^2 \sin(x_1 - x_2) + \theta_5 \cos x_2]}{[\theta_1 \theta_2 - \theta_3^2 \cos^2(x_1 - x_2)]^2} \end{aligned}$$

Con respecto a x_3 se tiene:

$$F(X_4)_{x_3} = \frac{2\theta_1 \theta_3 x_3 \sin(x_1 - x_2)}{\theta_1 \theta_2 - \theta_3^2 \cos^2(x_1 - x_2)}$$

Con respecto a x_4 se tiene:

$$F(X_4)_{x_4} = \frac{2\theta_3^2 x_4 \cos(x_1 - x_2) \sin(x_1 - x_2)}{\theta_1 \theta_2 - \theta_3^2 \cos^2(x_1 - x_2)}$$

Una vez obtenidas las derivadas de las funciones involucradas en la linealización se procede a evaluar el comportamiento de las funciones de aproximación con

respecto a los valores establecidos en el punto de equilibrio de tal manera que toman el comportamiento para la primera aproximación lineal:

$$F(X_{03}) = 0$$

$$F(X_{03})_{x_1} = \frac{-\theta_4\theta_2}{\theta_1\theta_2} + \frac{\theta_5\theta_3}{\theta_1\theta_2}$$

$$F(X_{03})_{x_2} = \frac{-\theta_5\theta_3}{\theta_1\theta_2}$$

$$F(X_{03})_{x_3} = 0$$

$$F(X_{03})_{x_4} = 0$$

La aproximación lineal correspondiente a la primera función buscada es:

$$F_A(X_3) = F(X_{03}) + \sum_{i=1}^{i=4} F(X_{03})_{xi} [x_i - x_{0i}]$$

$$F_A(X_3) = F(X_{03}) + F(X_{03})_{x_1} [x_1 - x_{01}] + F(X_{03})_{x_2} [x_2 - x_{02}] + F(X_{03})_{x_3} [x_3 - x_{03}] + F(X_{03})_{x_4} [x_4 - x_{04}]$$

$$F_A(X_3) = F(X_{03})_{x_1} [x_1 - x_{01}] + F(X_{03})_{x_2} [x_2 - x_{02}]$$

Es decir:

$$F_A(X_3) = \left[\frac{-\theta_4\theta_2}{\theta_1\theta_2} + \frac{\theta_5\theta_3}{\theta_1\theta_2} \right] \left[x_1 + \frac{\pi}{2} \right] + \left[\frac{-\theta_5\theta_3}{\theta_1\theta_2} \right] [x_2 - \pi]$$

Evaluando el comportamiento de la función cuatro y sus derivadas se tiene:

$$F(X_{04}) = \frac{\theta_5\theta_1}{\theta_1\theta_2}$$

$$F(X_{04})_{x_1} = 0$$

$$F(X_{04})_{x_2} = 0$$

$$F(X_{04})_{x_3} = 0$$

$$F(X_{04})_{x_4} = 0$$

Con lo cual la aproximación lineal se construye como:

$$F_A(X_4) = F(X_{04}) + \sum_{i=1}^{i=4} F(X_{04})_{xi} [x_i - x_{0i}]$$

$$F_A(X_4) = F(X_{04}) + F(X_{04})_{x_1} [x_1 - x_{01}] + F(X_{04})_{x_2} [x_2 - x_{02}] + F(X_{04})_{x_3} [x_3 - x_{03}] + F(X_{04})_{x_4} [x_4 - x_{04}]$$

$$F_A(X_4) = F(X_{04})$$

$$F_A(X_4) = \frac{\theta_5 \theta_1}{\theta_1 \theta_2}$$

La aproximación lineal alrededor del punto de equilibrio queda definida como:

$$F_A(X_4) = \frac{\theta_5}{\theta_2}$$

Del mismo modo al aplicar la linealización a la función correspondiente a la excitación externa de la primera ecuación se tiene:

$$T_{1A}(X_3) = T(X_{03})_1 + \sum_{i=1}^{i=4} T(X_{03})_{xi} [x_i - x_{0i}]$$

Los valores que toman las derivadas y la función excitación en el punto de equilibrio se tiene:

$$T(X_{03})_1 = \frac{\tau_1}{\theta_1}$$

$$T_1(X_{03})_{x1} = \frac{\tau_2 \theta_3}{\theta_1 \theta_2}$$

$$T_1(X_3)_{x2} = -\frac{\tau_2 \theta_3}{\theta_1 \theta_2}$$

$$T_1(X_3)_{x3} = 0$$

$$T_1(X_3)_{x4} = 0$$

Con lo cual la aproximación lineal se define como:

$$T_A(X_3)_1 = T(X_{03})_1 + T(X_{03})_{x1} [x_1 - x_{01}] + T(X_{03})_{x2} [x_2 - x_{02}] + T(X_{03})_{x3} [x_3 - x_{03}] + T(X_{03})_{x4} [x_4 - x_{04}]$$

$$T_A(X_3)_1 = \frac{\tau_1}{\theta_1} + \left[\frac{\tau_2 \theta_3}{\theta_1 \theta_2} \right] \left[x_1 + \frac{\pi}{2} \right] + \left[-\frac{\tau_2 \theta_3}{\theta_1 \theta_2} \right] [x_2 - \pi]$$

Continuando con la evaluación de la segunda función de aproximación no lineal de excitación y evaluando su comportamiento en el punto de equilibrio tenemos:

$$T(X_{04})_1 = \frac{\tau_2}{\theta_2}$$

$$T_1(X_{04})_{x1} = \frac{\tau_1 \theta_3}{\theta_1 \theta_2}$$

$$T_1(X_{04})_{x2} = -\frac{\tau_1 \theta_3}{\theta_1 \theta_2}$$

$$T_1(X_{04})_{x3} = 0$$

$$T_1(X_{04})_{x4} = 0$$

Con lo cual la aproximación lineal se define como:

$$T_A(X_4)_1 = T(X_{04})_1 + \sum_{i=1}^{i=4} T(X_{04})_{xi} [x_i - x_{0i}]$$

$$T_A(X_4)_1 = T(X_{04})_1 + T(X_{04})_{x1} [x_1 - x_{01}] + T(X_{04})_{x2} [x_2 - x_{02}] + T(X_{04})_{x3} [x_3 - x_{03}] + T(X_{04})_{x4} [x_4 - x_{04}]$$

$$T_A(X_4)_1 = \frac{\tau_2}{\theta_2} + \left[\frac{\tau_1 \theta_3}{\theta_1 \theta_2} \right] \left[x_1 + \frac{\pi}{2} \right] + \left[-\frac{\tau_1 \theta_3}{\theta_1 \theta_2} \right] [x_2 - \pi]$$

De tal forma que el sistema lineal aproximado en el punto de equilibrio es:

$$\begin{aligned} \dot{x}_1 &= x_3 \\ \dot{x}_2 &= x_4 \\ \dot{x}_3 &= \left[\frac{-\theta_4 \theta_2}{\theta_1 \theta_2} + \frac{\theta_5 \theta_3}{\theta_1 \theta_2} \right] \left[x_1 + \frac{\pi}{2} \right] + \left[\frac{-\theta_5 \theta_3}{\theta_1 \theta_2} \right] [x_2 - \pi] + \frac{\tau_1}{\theta_1} + \left[\frac{\tau_2 \theta_3}{\theta_1 \theta_2} \right] \left[x_1 + \frac{\pi}{2} \right] + \left[-\frac{\tau_2 \theta_3}{\theta_1 \theta_2} \right] [x_2 - \pi] \\ \dot{x}_4 &= \frac{\theta_5}{\theta_2} + \frac{\tau_2}{\theta_2} + \left[\frac{\tau_1 \theta_3}{\theta_1 \theta_2} \right] \left[x_1 + \frac{\pi}{2} \right] + \left[-\frac{\tau_1 \theta_3}{\theta_1 \theta_2} \right] [x_2 - \pi] \end{aligned}$$

Se definen los siguientes cambios de variables en el sistema, los cuales definen las variables de estado presentes en la modelación del sistema dinámico:

$$\begin{aligned} y_1 &= x_1 + \frac{\pi}{2} \\ y_2 &= x_2 - \pi \\ y_3 &= x_3 \\ y_4 &= x_4 \end{aligned}$$

Recordando que el valor del torque 2 aplicado es cero, es decir:

$$\tau_2 = 0$$

El sistema dinámico, toma la forma entonces:

$$\begin{aligned} \dot{y}_1 &= y_3 \\ \dot{y}_2 &= y_4 \\ \dot{y}_3 &= \left[\frac{-\theta_4}{\theta_1 \theta_2} + \frac{\theta_5 \theta_3}{\theta_1 \theta_2} \right] [y_1] + \left[\frac{-\theta_5 \theta_3}{\theta_1 \theta_2} \right] [y_2] + \frac{\tau_1}{\theta_1} \\ \dot{y}_4 &= \frac{\theta_5}{\theta_2} + \left[\frac{\tau_1 \theta_3}{\theta_1 \theta_2} \right] [y_1] + \left[-\frac{\tau_1 \theta_3}{\theta_1 \theta_2} \right] [y_2] \end{aligned}$$

En el apéndice A-1 al A-3 se muestra el desarrollo de la linealización del sistema dinámico junto con la prueba de controlabilidad del mismo de manera que, una vez establecido y definido el sistema anterior se tiene que:

$$[A, B] = [B \quad \hat{A}B \quad \hat{A}^2B \quad \hat{A}^3B]$$

La matriz anterior es de orden cuatro, y el determinante es:

$$DET(A, B) \neq 0$$

Por lo tanto, de acuerdo al criterio de controlabilidad, el sistema linealizado en el punto de equilibrio es controlable

3.3.4.- DESARROLLO DEL CONTROL RETROALIMENTADO DEL SISTEMA

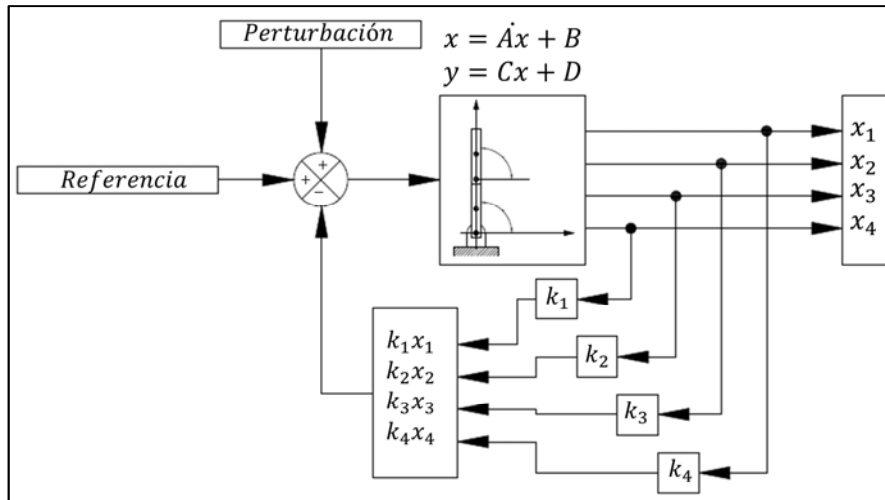


Fig. 3.4. Control por retroalimentación de estados del péndulo doble invertido linealizado

Una vez establecida la aproximación lineal en el punto de equilibrio se procede a efectuar el control del mismo, para lo cual se desarrolla en Matlab el código de control y programación dinámica bajo el cual se efectuaran las pruebas con diferentes puntos de localización de los polos, se procede a obtener la respuesta por medios computacionales a una condición inicial la cual tomara diferentes valores numéricos para efectuar el análisis de sensibilidad correspondiente en capítulos posteriores definida como:

$$X_{ini} = [x_{ini1} \quad x_{ini2} \quad x_{ini3} \quad x_{ini4}]$$

Se desarrolla un programa utilizando el lenguaje de programación en Matlab mostrado en el apéndice donde es analizado y desarrollado el control del péndulo doble invertido linealizado en el punto de equilibrio establecido, con base a lo cual se obtienen las respuestas gráficas de cada una de las variables de estado y la solución del sistema dinámico linealizado perturbado bajo la condición inicial establecida.

La figura 3.4 ilustra el diagrama de control por medio de retroalimentación de estados que es utilizado para el péndulo doble invertido en la posición de equilibrio

de estudio. Posteriormente se hace la extensión de este sistema de control al modelo no lineal bajo un parámetro de error admisible que será determinado.

3.4.- DESARROLLO DE CONTROL DEL SISTEMA DINÁMICO DEL PÉNDULO DOBLE INVERTIDO NO LINEAL

Una vez definido el control del péndulo doble mediante linealización en el punto de equilibrio se procede a efectuar el control del sistema no lineal basados en el control lineal bajo el punto de equilibrio inestable.

La figura 3.5 ilustra el esquema del control para el sistema dinámico no linealizado establecido en un entorno cercano al punto de equilibrio de linealización del mismo recordando que si se sale este sistema de este punto de control de un error permitido por el próximo sistema dinámico, el control no surtirá efecto ya en el sistema.

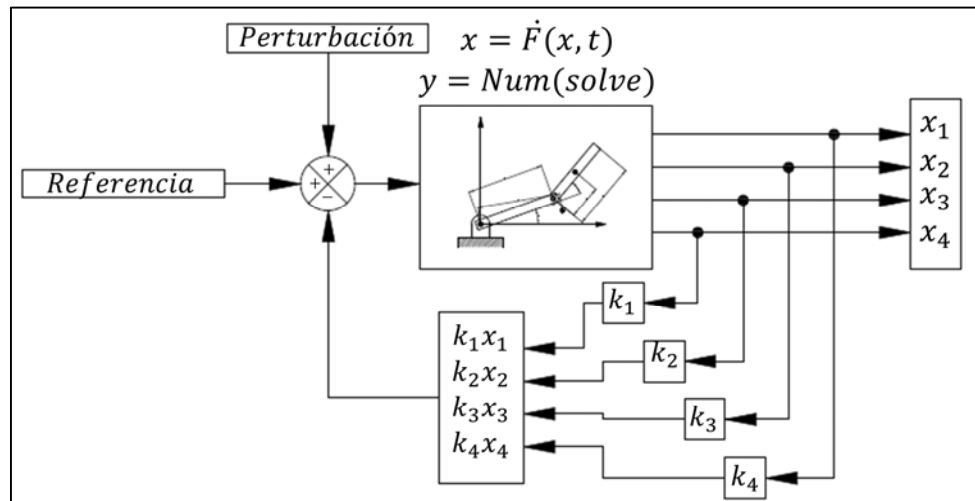


Fig. 3.5. Control por retroalimentación de estados del péndulo doble invertido no lineal

El sistema en forma no autónoma del péndulo doble invertido es definido por el modelo dinámico no lineal:

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & h_1 & h_2 \\ 0 & 0 & h_3 & h_4 \end{bmatrix} \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \end{bmatrix} = \begin{bmatrix} x_3 \\ x_4 \\ -\theta_3 \sin(x_1 - x_2) x_4^2 - \theta_4 \cos x_1 \\ \theta_3 \sin(x_1 - x_2) x_3^2 - \theta_5 \cos x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \tau_1 \\ \tau_2 \end{bmatrix}$$

Sustituyendo los valores obtenidos a través de la obtención de las matrices inversas se tiene un sistema de la forma:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \end{bmatrix} = \begin{bmatrix} x_3 \\ x_4 \\ \frac{[\theta_3 \cos(x_1 - x_2)][-\theta_3 x_3^2 \sin(x_1 - x_2) + \theta_5 \cos(x_2)]}{\theta_1 \theta_2 - \theta_3^2 \cos^2(x_1 - x_2)} - \frac{\theta_2 [\theta_3 x_4^2 \sin(x_1 - x_2) + \theta_4 \cos(x_1)]}{\theta_1 \theta_2 - \theta_3^2 \cos^2(x_1 - x_2)} \\ \frac{[\theta_3 \cos(x_1 - x_2)][\theta_3 x_4^2 \sin(x_1 - x_2) + \theta_4 \cos(x_1)]}{\theta_1 \theta_2 - \theta_3^2 \cos^2(x_1 - x_2)} - \frac{\theta_1 [-\theta_3 x_3^2 \sin(x_1 - x_2) + \theta_5 \cos(x_2)]}{\theta_1 \theta_2 - \theta_3^2 \cos^2(x_1 - x_2)} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \frac{\theta_2 \tau_1}{\theta_1 \theta_2 - \theta_3^2 \cos^2(x_1 - x_2)} - \frac{\theta_3 \tau_2 \cos(x_1 - x_2)}{\theta_1 \theta_2 - \theta_3^2 \cos^2(x_1 - x_2)} \\ \frac{\theta_1 \tau_2}{\theta_1 \theta_2 - \theta_3^2 \cos^2(x_1 - x_2)} - \frac{\theta_3 \tau_1 \cos(x_1 - x_2)}{\theta_1 \theta_2 - \theta_3^2 \cos^2(x_1 - x_2)} \end{bmatrix}$$

Tomando el sistema anterior una forma que puede definirse como:

$$[\dot{X}] = F(X) + [B_0]u$$

Recordando que la acción de control es definida como:

$$u = -KX$$

Las variables matriciales involucradas en el sistema toman la forma:

$$F(X) = \begin{bmatrix} x_3 \\ x_4 \\ \frac{[\theta_3 \cos(x_1 - x_2)][-\theta_3 x_3^2 \sin(x_1 - x_2) + \theta_5 \cos(x_2)]}{\theta_1 \theta_2 - \theta_3^2 \cos^2(x_1 - x_2)} - \frac{\theta_2 [\theta_3 x_4^2 \sin(x_1 - x_2) + \theta_4 \cos(x_1)]}{\theta_1 \theta_2 - \theta_3^2 \cos^2(x_1 - x_2)} \\ \frac{[\theta_3 \cos(x_1 - x_2)][\theta_3 x_4^2 \sin(x_1 - x_2) + \theta_4 \cos(x_1)]}{\theta_1 \theta_2 - \theta_3^2 \cos^2(x_1 - x_2)} - \frac{\theta_1 [-\theta_3 x_3^2 \sin(x_1 - x_2) + \theta_5 \cos(x_2)]}{\theta_1 \theta_2 - \theta_3^2 \cos^2(x_1 - x_2)} \end{bmatrix}$$

$$B_0 = \begin{bmatrix} 0 \\ 0 \\ \frac{\theta_2 \tau_1}{\theta_1 \theta_2 - \theta_3^2 \cos^2(x_1 - x_2)} - \frac{\theta_3 \tau_2 \cos(x_1 - x_2)}{\theta_1 \theta_2 - \theta_3^2 \cos^2(x_1 - x_2)} \\ \frac{\theta_1 \tau_2}{\theta_1 \theta_2 - \theta_3^2 \cos^2(x_1 - x_2)} - \frac{\theta_3 \tau_1 \cos(x_1 - x_2)}{\theta_1 \theta_2 - \theta_3^2 \cos^2(x_1 - x_2)} \end{bmatrix}$$

En el apéndice A-5 se define el programa en Matlab bajo el cual se desarrolla el control del sistema dinámico no lineal bajo los métodos de aproximación lineal mostrados anteriormente. Se procede, mediante condiciones iniciales a localizar el sistema dinámico del péndulo doble invertido no lineal en la posición de equilibrio anteriormente definida

3.5.- MODELO COMPUTACIONAL DE PÉNDULO DOBLE INVERTIDO

El primer paso para definir las propiedades mecánicas requeridas por el sistema de péndulo doble invertido a analizar es asignar un material de fabricación de los

componentes del modelo de prueba, en este caso se plantea una aleación de aluminio para algunos componentes y de acero estructural para otros componentes, dado que son materiales relativamente económicos y fáciles de conseguir en el mercado entre otras características que poseen. La aleación de aluminio seleccionada es la 6063-T6, dadas las características de mediana resistencia mecánica, alta resistencia a la corrosión, buen acabado superficial, especial para anodizar. La aleación de acero estructural seleccionada es Cold Rolled 1018 debido entre otros factores a que es una de las aleaciones más comerciales que existe y que, dadas las propiedades mecánicas del mismo.

El siguiente paso consiste en definir propiamente el modelo geométrico del péndulo doble invertido, con ello, se determinan completamente los parámetros que están incluidos en el modelo dinámico. El modelo geométrico se desarrolla en el entorno computacional solidworks, dado que este software posee las características de interacción entre propiedades mecánicas y propiedades geométricas. El péndulo doble invertido propuesto se compone de las partes:

- Rodamiento soporte
- Tuerca de unión
- Soporte fijo
- Base de soporte
- Perno de ajuste
- Brazo inferior (eslabón 1)
- Perno de unión
- Brazo superior (eslabón 2)

Se definen a continuación las características mecánicas de cada componente.

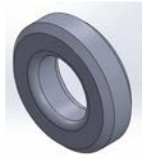
Nombre: Rodamiento soporte		Características
		Usados en una gran variedad de aplicaciones. Son fáciles de diseñar, no separables, capaces de operar en altas e incluso muy altas velocidades y requieren poca atención o mantenimiento en servicio.

Tabla 3.1. Características mecánicas del rodamiento soporte

Un rodamiento también denominado rulemán es un tipo de cojinete, que es un elemento mecánico que reduce la fricción entre un eje y las piezas conectadas a éste por medio de rodadura, que le sirve de apoyo y facilita su desplazamiento. El primer componente a utilizar es un rodamiento, el cual tendrá como función principal entrelazar el brazo inferior y el soporte del péndulo doble invertido con el uso de un perno de ajuste, a continuación se ilustra el modelo generado.

La tuerca siempre debe tener las mismas características geométricas del perno con el que se acopla, por lo que está normalizada según los sistemas generales de roscas.

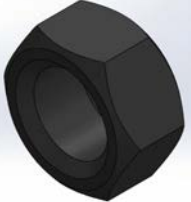
Nombre: Tuerca de unión		Características
		Se usa en tornillería y para aplicaciones de uso común. Las roscas métricas ISO de paso normal se designan anteponiendo la letra M al diámetro nominal en milímetros. Su forma detallada y dimensiones se especifican en la norma UNE 17-702, equivalente a la DIN 13 e ISO 261.

Tabla 3.2. Características mecánicas de la tuerca de unión

El siguiente componente tiene como función servir de soporte a los brazos del péndulo doble y junto con su base anclarse para poder efectuar posteriormente la acción de control mediante dispositivos electrónicos y sensores. Es mostrado en la tabla 1.3 y lleva por nombre soporte fijo.

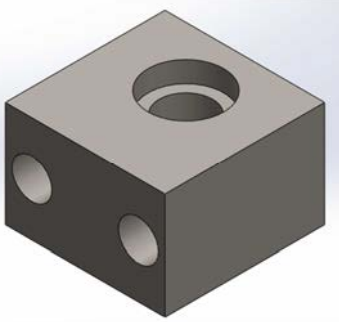
Nombre: Soporte fijo	Características
	Densidad = 0.01 gramos por milímetro cúbico Masa = 248.60 gramos Volumen = 31588.48 milímetros cúbicos Área de superficie = 8216.12 milímetros cuadrados Centro de masa: (milímetros) X = -0.00 Y = 12.49 Z = -0.02 Ejes principales de inercia y momentos principales de inercia: (gramos * milímetros cuadrados) Medido desde el centro de masa. $I_x = (1.00, -0.00, -0.00)$ $P_x = 45257.64$ $I_y = (-0.00, -0.01, -1.00)$ $P_y = 46448.02$ $I_z = (0.00, 1.00, -0.01)$ $P_z = 64450.10$ Momentos de inercia: (gramos * milímetros cuadrados) Obtenidos en el centro de masa y alineados con el sistema de coordenadas de resultados. $L_{xx} = 45257.64$ $L_{xy} = -0.02$ $L_{xz} = -1.05$ $L_{yx} = -0.02$ $L_{yy} = 64446.48$ $L_{yz} = 255.24$ $L_{zx} = -1.05$ $L_{zy} = 255.24$ $L_{zz} = 46451.64$ Momentos de inercia: (gramos * milímetros cuadrados) Medido desde el sistema de coordenadas de salida. $I_{xx} = 84062.45$ $I_{xy} = -0.95$ $I_{xz} = -1.05$ $I_{yx} = -0.95$ $I_{yy} = 64446.62$ $I_{yz} = 181.44$ $I_{zx} = -1.05$ $I_{zy} = 181.44$ $I_{zz} = 85256.32$

Tabla 3.3. Características mecánicas del soporte fijo

La base del soporte es el elemento fijo mediante el cual se logra el anclaje final de todos los componentes del sistema, es unida a la base soporte mediante tornillos. Es mostrada a continuación en la tabla 1.4.

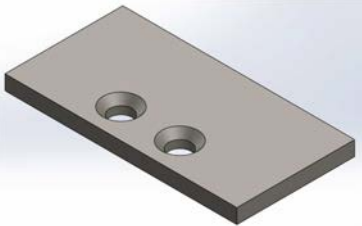
Nombre: Base soporte	Características
	Densidad = 0.01 gramos por milímetro cúbico Masa = 244.09 gramos Volumen = 31015.68 milímetros cúbicos Área de superficie = 12086.77 milímetros cuadrados Centro de masa: (milímetros) $X = -0.00$ $Y = 3.15$ $Z = -0.51$ Ejes principales de inercia y momentos principales de inercia: (gramos * milímetros cuadrados) Medido desde el centro de masa. $I_x = (1.00, -0.00, -0.00)$ $P_x = 53683.96$ $I_y = (-0.00, 0.00, -1.00)$ $P_y = 210710.04$ $I_z = (0.00, 1.00, 0.00)$ $P_z = 262764.58$ Momentos de inercia: (gramos * milímetros cuadrados) Obtenidos en el centro de masa y alineados con el sistema de coordenadas de resultados. $L_{xx} = 53683.96$ $L_{xy} = -0.35$ $L_{xz} = -15.20$ $L_{yx} = -0.35$ $L_{yy} = 262764.48$ $L_{yz} = -75.48$ $L_{zx} = -15.20$ $L_{zy} = -75.48$ $L_{zz} = 210710.14$ Momentos de inercia: (gramos * milímetros cuadrados) Medido desde el sistema de coordenadas de salida. $I_{xx} = 56171.58$ $I_{xy} = -3.98$ $I_{xz} = -14.62$ $I_{yx} = -3.98$ $I_{yy} = 262827.64$ $I_{yz} = -466.81$ $I_{zx} = -14.62$ $I_{zy} = -466.81$ $I_{zz} = 213134.61$

Tabla 3.4. Características mecánicas de la base de soporte

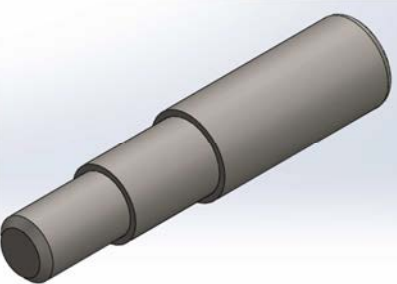
Nombre: Perno de ajuste	Características
	Densidad = 0.01 gramos por milímetro cúbico Masa = 31.54 gramos Volumen = 4007.34 milímetros cúbicos Área de superficie = 1756.57 milímetros cuadrados Centro de masa: (milímetros) $X = 0.00$ $Y = 0.00$ $Z = -16.30$ Ejes principales de inercia y momentos principales de inercia: (gramos * milímetros cuadrados) Medido desde el centro de masa. $I_x = (0.00, 0.00, 1.00)$ $P_x = 419.70$ $I_y = (0.00, -1.00, 0.00)$ $P_y = 6271.68$ $I_z = (1.00, 0.00, 0.00)$ $P_z = 6271.68$ Momentos de inercia: (gramos * milímetros cuadrados) Obtenidos en el centro de masa y alineados con el sistema de coordenadas de resultados. $L_{xx} = 6271.68$ $L_{xy} = 0.00$ $L_{xz} = 0.00$ $L_{yx} = 0.00$ $L_{yy} = 6271.68$ $L_{yz} = 0.00$ $L_{zx} = 0.00$ $L_{zy} = 0.00$ $L_{zz} = 419.70$ Momentos de inercia: (gramos * milímetros cuadrados) Medido desde el sistema de coordenadas de salida. $I_{xx} = 14647.85$ $I_{xy} = 0.00$ $I_{xz} = 0.00$ $I_{yx} = 0.00$ $I_{yy} = 14647.85$ $I_{yz} = 0.00$ $I_{zx} = 0.00$ $I_{zy} = 0.0$ $I_{zz} = 419.70$

Tabla 3.5. Características mecánicas del perno de ajuste.

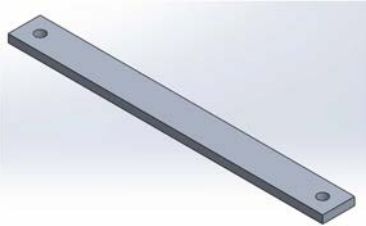
Nombre: Brazo inferior	Características
	Densidad = 0.00 gramos por milímetro cúbico Masa = 107.17 gramos Volumen = 39694.06 milímetros cúbicos Área de superficie = 16316.34 milímetros cuadrados Centro de masa: (milímetros) $X = 0.00$ $Y = 3.17$ $Z = 0.00$
	Ejes principales de inercia y momentos principales de inercia: (gramos * milímetros cuadrados) Medido desde el centro de masa. $I_x = (1.00, 0.00, 0.00)$ $P_x = 6206.70$ $I_y = (0.00, 0.00, -1.00)$ $P_y = 545990.05$ $I_z = (0.00, 1.00, 0.00)$ $P_z = 551476.50$
	Momentos de inercia: (gramos * milímetros cuadrados) Obtenidos en el centro de masa y alineados con el sistema de coordenadas de resultados. $L_{xx} = 6206.70$ $L_{xy} = 0.00$ $L_{xz} = 0.00$ $L_{yx} = 0.00$ $L_{yy} = 551476.50$ $L_{yz} = 0.00$ $L_{zx} = 0.00$ $L_{zy} = 0.00$ $L_{zz} = 545990.05$
	Momentos de inercia: (gramos * milímetros cuadrados) Medido desde el sistema de coordenadas de salida. $I_{xx} = 7287.08$ $I_{xy} = 0.00$ $I_{xz} = 0.00$ $I_{yx} = 0.00$ $I_{yy} = 551476.50$ $I_{yz} = 0.00$ $I_{zx} = 0.00$ $I_{zy} = 0.00$ $I_{zz} = 547070.43$

Tabla 3.6. Características mecánicas del brazo inferior

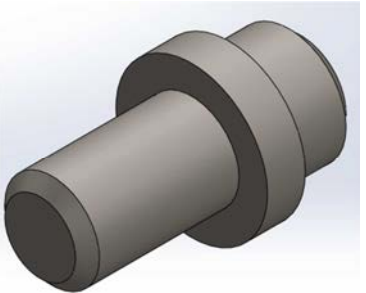
Nombre: Perno de unión	Características
	Densidad = 0.01 gramos por milímetro cúbico Masa = 11.28 gramos Volumen = 1432.83 milímetros cúbicos Área de superficie = 841.53 milímetros cuadrados Centro de masa: (milímetros) $X = 0.00$ $Y = 0.00$ $Z = 0.48$
	Ejes principales de inercia y momentos principales de inercia: (gramos * milímetros cuadrados) Medido desde el centro de masa. $I_x = (0.00, 0.00, 1.00)$ $P_x = 137.56$ $I_y = (0.00, -1.00, 0.00)$ $P_y = 458.77$ $I_z = (1.00, 0.00, 0.00)$ $P_z = 458.77$
	Momentos de inercia: (gramos * milímetros cuadrados) Obtenidos en el centro de masa y alineados con el sistema de coordenadas de resultados. $L_{xx} = 458.77$ $L_{xy} = 0.00$ $L_{xz} = 0.00$ $L_{yx} = 0.00$ $L_{yy} = 458.77$ $L_{yz} = 0.00$ $L_{zx} = 0.00$ $L_{zy} = 0.00$ $L_{zz} = 137.56$
	Momentos de inercia: (gramos * milímetros cuadrados) Medido desde el sistema de coordenadas de salida. $I_{xx} = 461.42$ $I_{xy} = 0.00$ $I_{xz} = 0.00$ $I_{yx} = 0.00$ $I_{yy} = 461.42$ $I_{yz} = 0.00$ $I_{zx} = 0.00$ $I_{zy} = 0.00$ $I_{zz} = 137.56$

Tabla 3.7. Características mecánicas del perno de unión

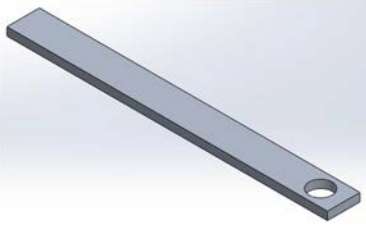
Nombre: Brazo superior	Características
	Densidad = 0.00 gramos por milímetro cúbico Masa = 104.51 gramos Volumen = 38706.62 milímetros cúbicos Área de superficie = 16047.73 milímetros cuadrados Centro de masa: (milímetros) X = -4.47 Y = 3.17 Z = 0.00 Ejes principales de inercia y momentos principales de inercia: (gramos * milímetros cuadrados) Medido desde el centro de masa. Ix = (1.00, 0.00, 0.00) Px = 6116.07 Iy = (0.00, 0.00, -1.00) Py = 515262.19 Iz = (0.00, 1.00, 0.00) Pz = 520675.93 Momentos de inercia: (gramos * milímetros cuadrados) Obtenidos en el centro de masa y alineados con el sistema de coordenadas de resultados. Lxx = 6116.07 Lxy = 0.00 Lxz = 0.00 Lyx = 0.00 Lyy = 520675.93 Lyz = 0.00 Lzx = 0.00 Lzy = 0.00 Lzz = 515262.19 Momentos de inercia: (gramos * milímetros cuadrados) Medido desde el sistema de coordenadas de salida. Ixx = 7169.58 Ixy = -1482.18 Ixz = 0.00 Iyx = -1482.18 Iyy = 522761.20 Iyz = 0.00 Izx = 0.00 Izy = 0.00 Izz = 518400.97

Tabla 3.8. Características mecánicas del brazo superior

El elemento mostrado en la tabla 1.5 es el perno que sirve de interacción entre la base fija y el brazo inferior del péndulo doble invertido de prueba, este a su vez va unido al rodamiento y al rotor de acción de control. El siguiente elemento es el brazo inferior, mostrado en la tabla 1.6 del péndulo doble, el cual está diseñado de aleación de aluminio 6063-T6, las características del mismo son listadas a continuación junto a la imagen del mismo. El elemento mostrado en la tabla 1.7 es el perno de unión entre el brazo inferior y el brazo superior del péndulo doble invertido junto con el rodamiento, el cual tiene la función de disminuir la fricción entre los elementos. Este elemento va fijo al brazo inferior mediante una tuerca de fijación y móvil en el brazo superior, es fabricado de acero estructural cold rolled 1018.

El elemento del modelo de prueba presentado en la tabla 1.8 es el brazo superior, el cual es diseñado de aleación de aluminio 6063-T6, este elemento viene unido al péndulo inferior por medio del perno ilustrado anteriormente y se diseña con una cavidad para alojar al rodamiento bajo el cual se lleva acabo el efecto de eliminación de la fricción entre los elementos mecánicos, lo cual a la vez ejerce el problema principal del sistema, el cual es la propiedad dinámica de sistema subactuado. Una vez definidos los elementos que forman el ensamble general del

sistema a definir se desarrolla el ensamble general del sistema dinámico péndulo doble invertido, el cual es ilustrado en la figura 3.6.

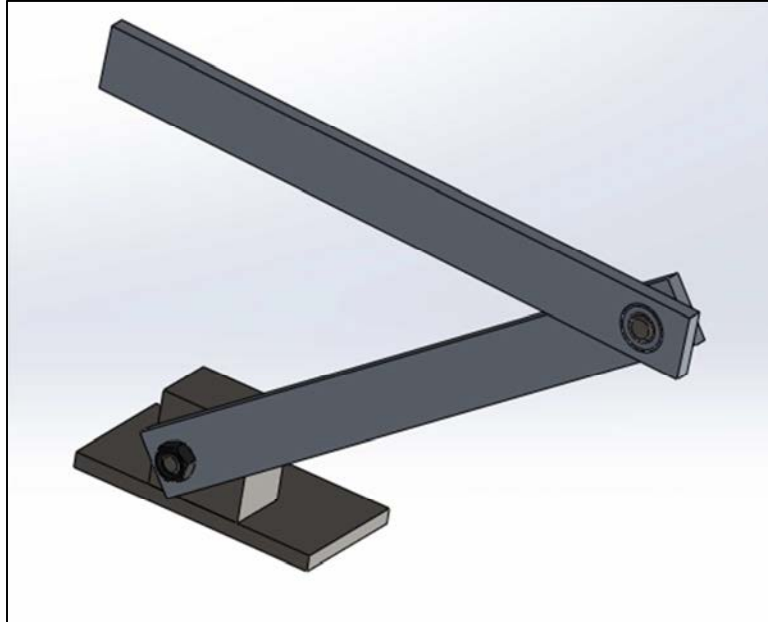


Fig. 3.6. Ensamble de modelo de prueba péndulo doble invertido

3.6.- PARÁMETROS DE PRUEBA EN EL DISEÑO DEL PÉNDULO DOBLE INVERTIDO

Con el modelo computacional anteriormente definido y los materiales seleccionados junto con las propiedades mecánicas del mismo se tiene la posibilidad de alimentar la ecuación característica con los mismos. Los cuales toman los valores:

m_1 : 0.10717 kg

m_2 : 0.10451 kg

L_1 : 0.25 m

L_2 : 0.25 m

l_{c1} : 0.125 m

l_{c2} : 0.1697 m

I_1 : 545990.05 grmm²=0

I_2 : 515262.19 grmm²=0 kgm²

g : 9.81 m/s²

$q_1 = q_1(t)$ (Grado de libertad)

$q_2 = q_2(t)$ (Grado de libertad)

τ_1 : 0 (Modelo dinámico libre)

Tomando las definiciones de parámetros expuestos en el capítulo 2 se tiene que:

$$\begin{aligned}
\theta_1 &= I_1 + m_2 L_1^2 \\
\theta_2 &= I_2 \\
\theta_3 &= m_2 L_1 l_{c_2} \\
\theta_4 &= g(m_1 l_{c_1} + m_2 L_1) \\
\theta_5 &= m_2 g l_{c_2}
\end{aligned}$$

Con lo cual las variables definidas anteriormente toman la forma:

$$\begin{aligned}
\theta_1 &= (545990.05) + (0.10451)(0.25)^2 = 0.00653 \\
\theta_2 &= (515262.19) = 0 \\
\theta_3 &= (0.10451)(0.25)(0.1697) = 0.00433 \\
\theta_4 &= (9.81)[(0.10717)(0.125) + (0.10451)(0.25)] = 0.387 \\
\theta_5 &= (0.10451)(9.81)(0.1697) = 0.173
\end{aligned}$$

En base a estos valores establecidos se derivan las ecuaciones:

$$\begin{aligned}
\tau_1 &= \ddot{q}_1 \theta_1 + \theta_3 \cos(q_1 - q_2) \ddot{q}_2 + \theta_3 \sin(q_1 - q_2) \dot{q}_2^2 + \theta_4 \cos q_1 \\
\tau_2 &= \theta_3 \cos(q_1 - q_2) \ddot{q}_1 + \ddot{q}_2 \theta_2 - \theta_3 \sin(q_1 - q_2) \dot{q}_1^2 + \theta_5 \cos q_2
\end{aligned}$$

Las cuales, una vez sustituidos los parámetros determinados de prueba, toman la estructura:

$$\begin{aligned}
\tau_1 &= \ddot{q}_1 [0.00653] + [0.00433] \cos(q_1 - q_2) \ddot{q}_2 + [0.00433] \sin(q_1 - q_2) \dot{q}_2^2 + [0.387] \cos q_1 \\
\tau_2 &= [0.00433] \cos(q_1 - q_2) \ddot{q}_1 - [0.00433] \sin(q_1 - q_2) \dot{q}_1^2 + [0.173] \cos q_2
\end{aligned}$$

Las ecuaciones anteriores son evaluadas dinámicamente para observar el comportamiento de las mismas.

3.7.- ANÁLISIS DINÁMICO DEL PÉNDULO DOBLE INVERTIDO

Los métodos de Runge-Kutta son utilizados para proporcionar la solución del sistema dinámico planteado con las condiciones iniciales definidas son implementados en primera instancia por medio del software Matlab, se procede a caracterizar la dinámica del péndulo doble invertido cuando los movimientos en el sistema poseen diferentes condiciones iniciales. Para el primer análisis mostrado en la figura 3.7 se establecen las condiciones iniciales:

$$\begin{aligned}
q_1 &= 270^\circ \\
q_2 &= 90^\circ \\
\dot{q}_1 &= 0 \\
\dot{q}_2 &= 0
\end{aligned}$$

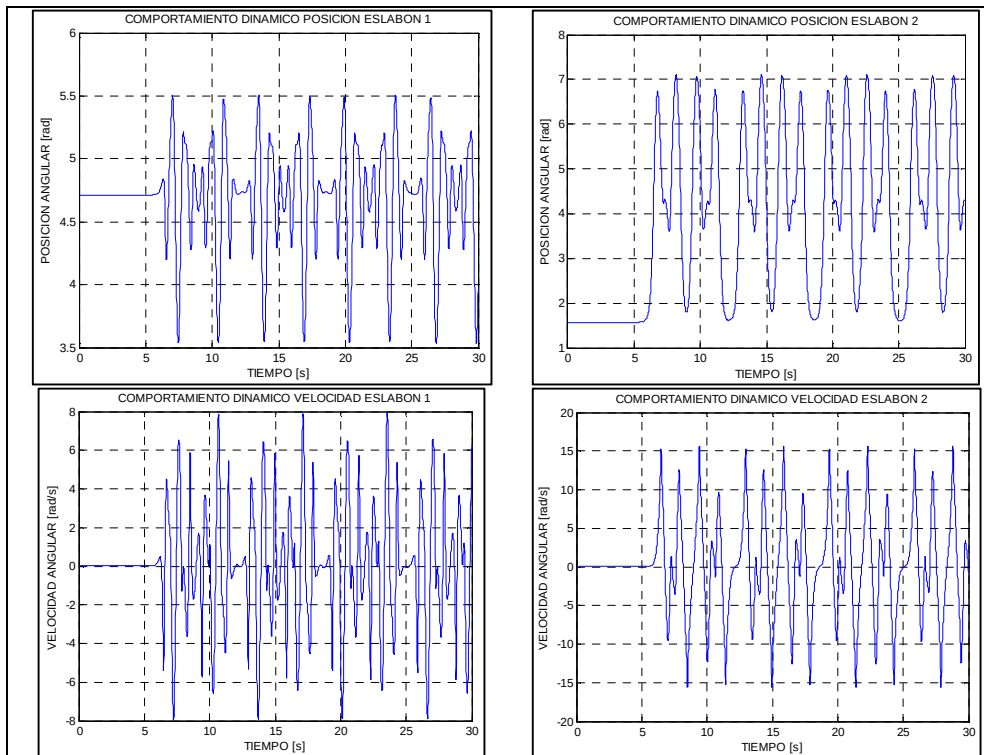


Fig. 3.7. Comportamiento dinámico de las variables del sistema dinámico de estudio bajo variación inicial 1

Se observa que el sistema en los primeros instantes de tiempo permanece en este punto de equilibrio pero posteriormente, después de unos segundos, la inercia presente en el sistema dinámico es vencida y se inicia el movimiento del mismo, buscando un punto de equilibrio estable, el cual corresponderá a un atractor natural del sistema. El sistema permanece en movimiento por tiempo indeterminado sin disminución de la amplitud del mismo y en un movimiento armónico, esto debido a que, el sistema no toma en cuenta parámetros friccionales de disipación de energía. Se efectúa una prueba más para determinar el comportamiento del sistema dinámico, bajo las condiciones iniciales siguientes:

$$\begin{aligned} q_1 &= 60^\circ \\ q_2 &= 0^\circ \\ \dot{q}_1 &= 0 \\ \dot{q}_2 &= 0 \end{aligned}$$

El sistema dinámico toma las respuestas mostradas en la figura número 3.8. Se observa con base a las respuestas dinámicas que el sistema si mantiene un comportamiento armónico los primeros instantes de tiempo para posteriormente realizar desplazamientos angulares en una sola dirección, es decir el sistema tiende a desplazarse indefinidamente.

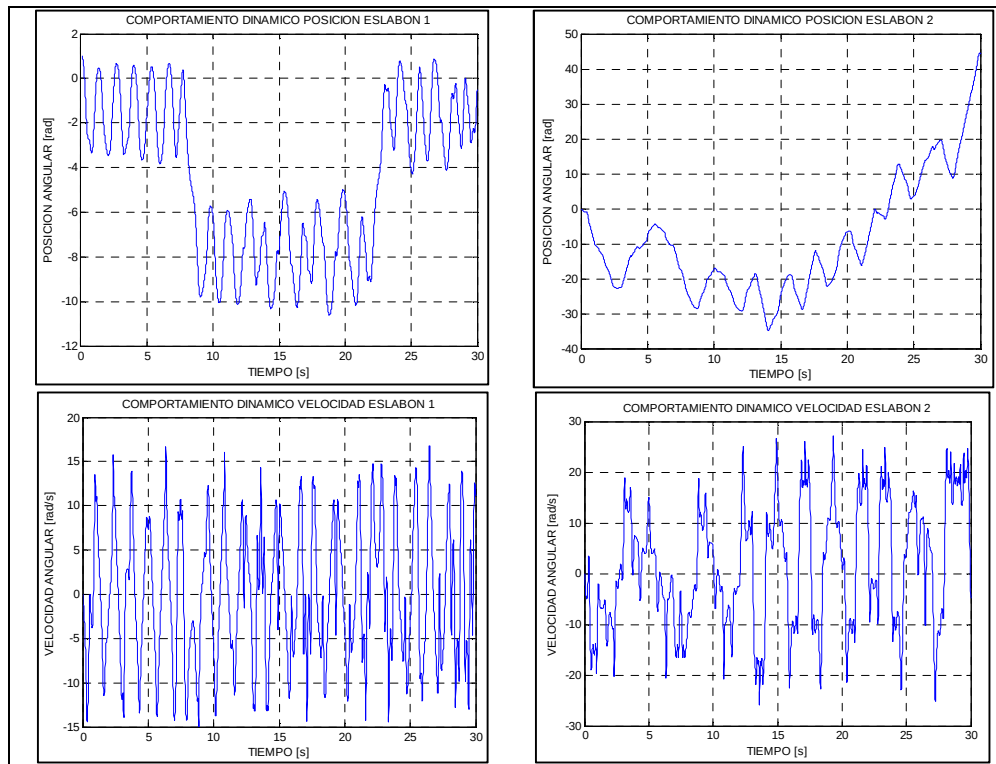


Fig. 3.8. Comportamiento dinámico de las variables del sistema dinámico de estudio bajo variación inicial 2

Las pruebas realizadas anteriormente con condiciones iniciales diferentes tienen un comportamiento periódico que se puede aproximar mediante series de armónicos estables en la mayoría de los casos, cabe señalar que no existe en el modelo dinámico hasta este momento acoplamiento de elementos de disipación de energía.

3.8.- ANÁLISIS DE CONTROL LINEAL EN EL MODELO DINÁMICO

Se procede a analizar el comportamiento del mismo y a obtener el comportamiento del sistema al ejercer variaciones en las condiciones iniciales del sistema para poder determinar la capacidad del control para llevar al sistema a la posición de referencia deseada. En primera instancia se describen las perturbaciones al sistema en las variables de estado siguientes, Con lo cual se observa que el sistema dinámico toma el comportamiento ilustrado en las figura 3.9.

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 30^\circ \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

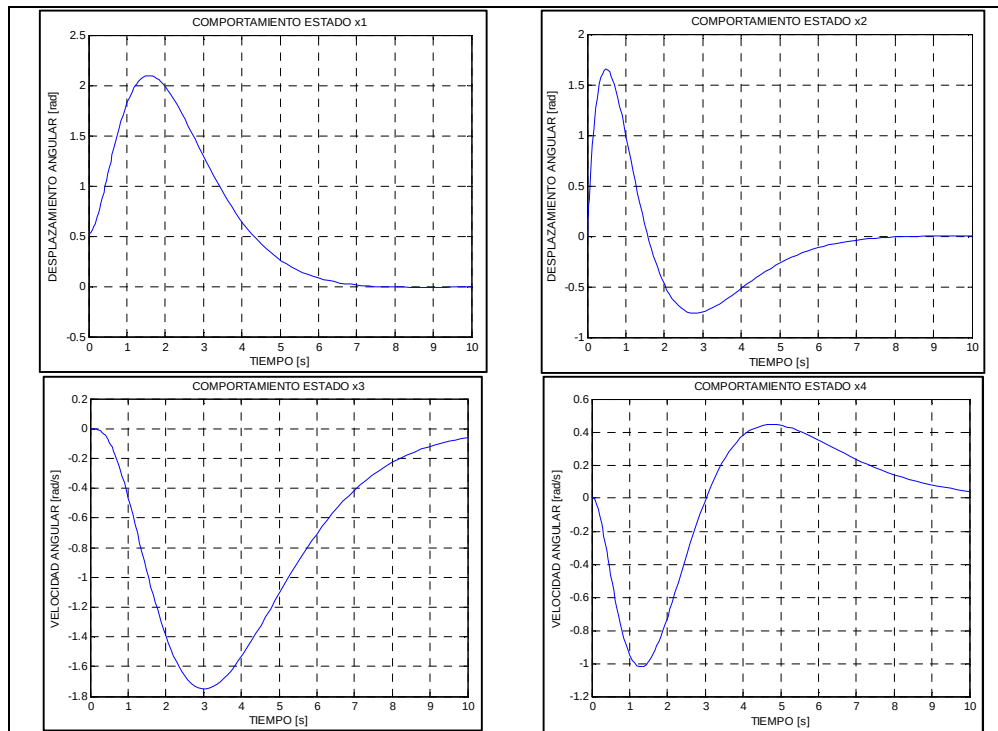


Fig. 3.9. Comportamiento dinámico control lineal variación primera

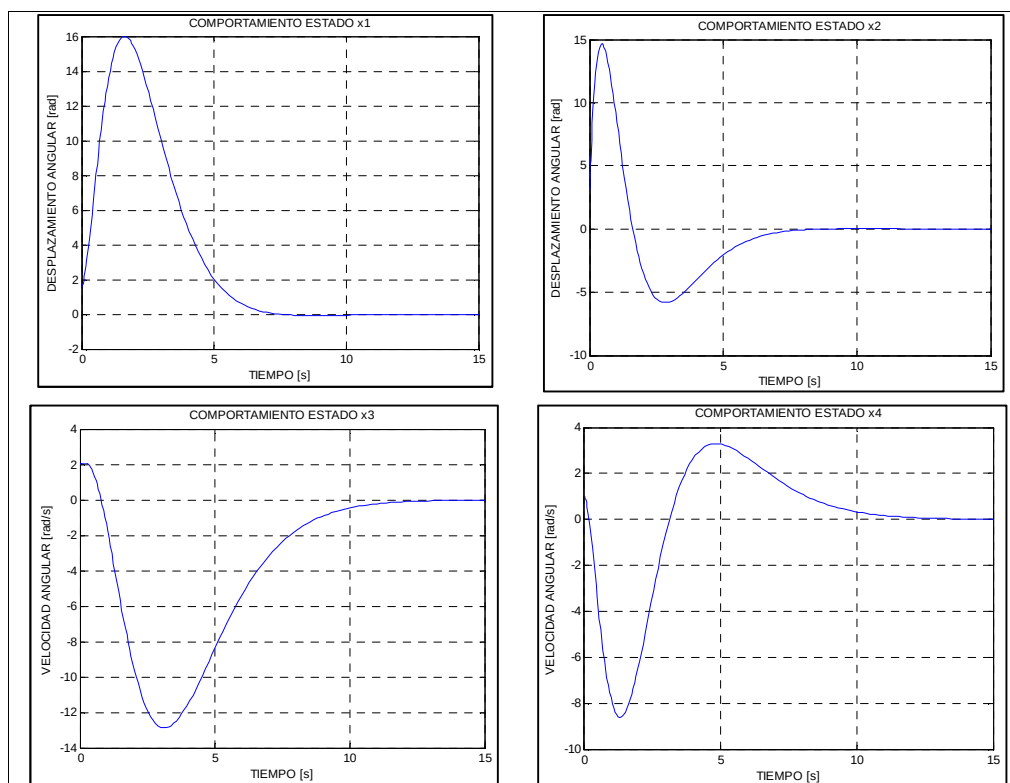


Fig. 3.10. Comportamiento dinámico del control lineal variación segunda.

Se observa que el control como las ganancias desarrolladas ejerce un atractor adecuado sobre el sistema linealizado del péndulo doble invertido. Se efectúa una perturbación mayor en el sistema dinámico para determinar la acción de control donde puede observarse un comportamiento estable y tiende a posicionarse en el punto de equilibrio expuesto pero en un tiempo de estabilidad mayor como se ilustra en la figura 3.10. La perturbación inicial es definida como:

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 90^\circ \\ 180^\circ \\ 2 \\ 1 \end{bmatrix}$$

Con base a estos análisis se concluye que la acción de control lineal del sistema del péndulo doble invertido logra controlar adecuadamente el sistema lineal, no importando el orden de las variaciones iniciales en las variables de estado.

3.9.- ANÁLISIS DE CONTROL NO LINEAL EN EL MODELO DINÁMICO

El control no lineal del sistema dinámico en cuestión está basado en el comportamiento linealizado alrededor del punto de equilibrio inestable:

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} -\frac{\pi}{2} \\ \pi \\ 0 \\ 0 \end{bmatrix}$$

En torno al cual el sistema no lineal permanece en equilibrio estable bajo la acción del control, en esta sección se investigara bajo que perturbaciones el control puede mantener en péndulo doble invertido en el atractor deseado sin que falle el control. Se define en primera instancia una perturbación inicial de:

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} -\frac{\pi}{2} + 0.1 \\ \pi \\ 0 \\ 0 \end{bmatrix}$$

En la figura 3.11 se observa que el control del sistema dinámico no lineal logra estabilizar la perturbación inicial expuesta. Posteriormente se define la perturbación al sistema de:

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} -\frac{\pi}{2} + 0.3 \\ \pi \\ 0 \\ 0 \end{bmatrix}$$

Donde se observa que el sistema no lineal aun soporta esa variación en las condiciones iniciales definidas en la figura 3.12.

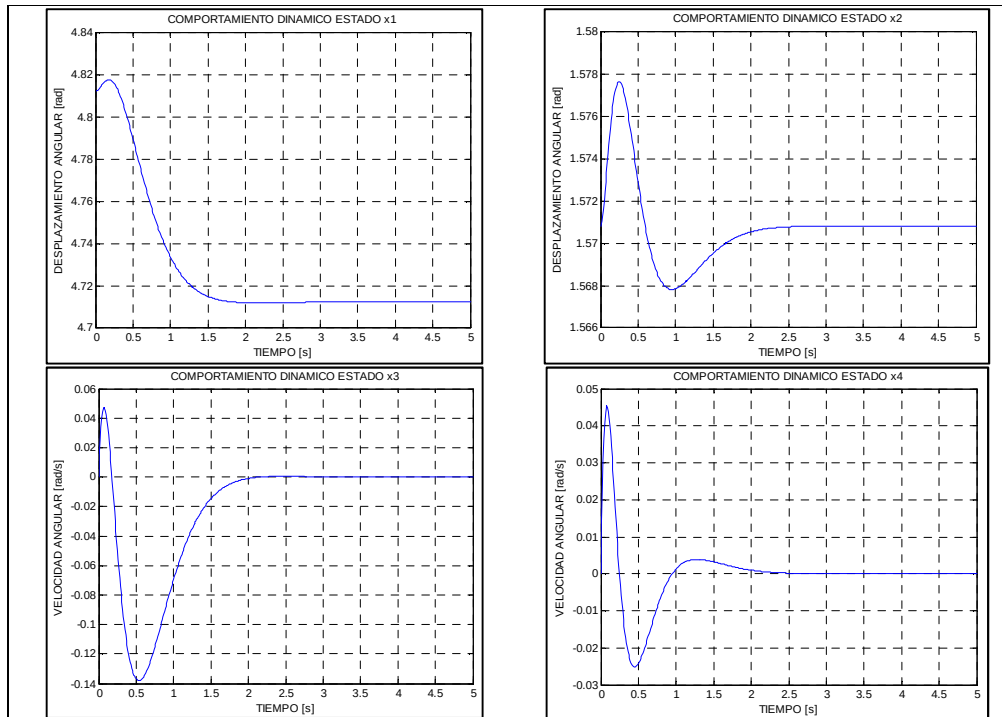


Fig. 3.11. Comportamiento dinámico del control no lineal variación primera

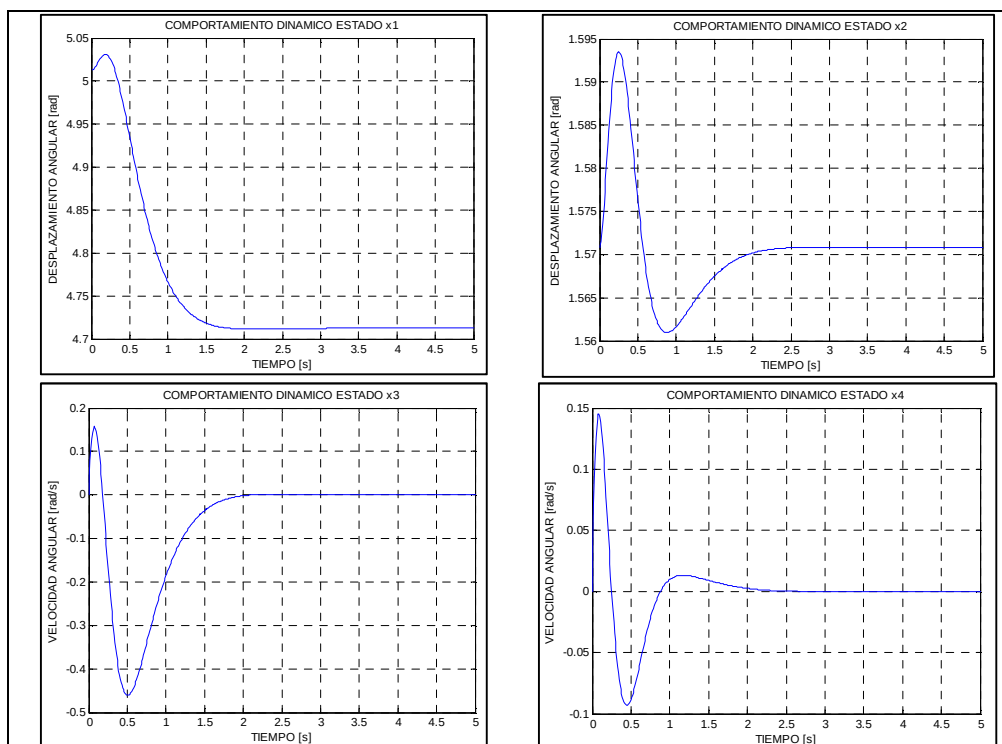


Fig. 3.12. Comportamiento dinámico del control no lineal variación segunda

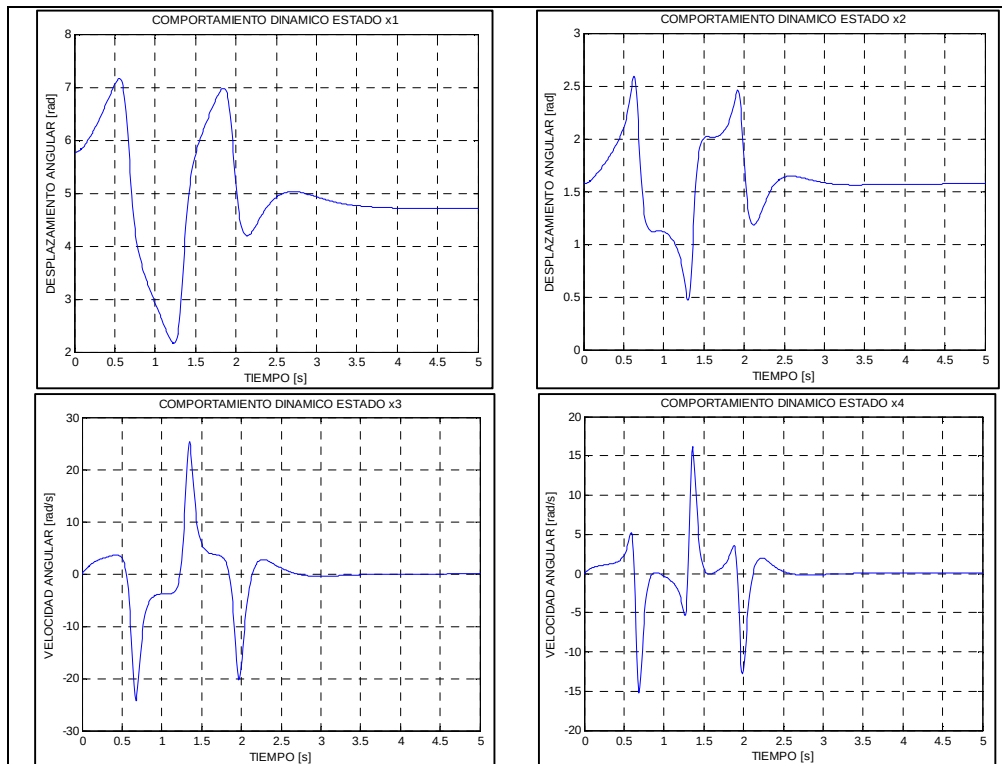


Fig. 3.13. Comportamiento dinámico del control no lineal variación límite.

En las figura 3.13 la variación límite a la que es sometido el sistema a la perturbación bajo la cual el sistema se vuelve inestable, la cual es de magnitud:

$$\Delta x_1 = 1.053017003$$

CAPÍTULO 4

ALGORITMOS EVOLUTIVOS Y DE ENJAMBRE

Resumen:

En el presente capítulo se desarrolla la teoría referente a los algoritmos evolutivos utilizados para sintonizar el control por retroalimentación de estados estudiado en capítulos anteriores. Primeramente, se describe el origen de los algoritmos basados en computación evolutiva y la clasificación de los principales algoritmos. Posteriormente, se describe el comportamiento de los algoritmos genéticos, los principales pasos a seguir para la construcción del algoritmo así como en qué consisten cada uno de los componentes del mismo. En tercer lugar se define la segunda técnica de sintonización basada en algoritmos de evolución diferencial, la estructura del mismo y las partes constituyen el algoritmo. Del mismo modo se plantea la estructura del algoritmo de optimización por enjambre de partículas y la constitución del mismo. Finalmente, se propone la propuesta de solución acoplada al sistema dinámico de estudio, se explica cómo se efectúa el acoplamiento y en que consiste el desarrollo de la acción de sintonización.

4.1.- INTRODUCCIÓN

El desarrollo de las técnicas de sintonización de sistemas dinámicos basadas en técnicas evolutivas debe de ser explicada y entendida a través de la comprensión del comportamiento de estos algoritmos para posteriormente desarrollar el control basado en estos métodos.

En el capítulo 1 se proporcionó un breve panorama sobre los métodos utilizados así como el rango de aplicación de los mismos, en las próximas páginas se proporciona el funcionamiento de los algoritmos para que bajo su lógica se desarrolle el lenguaje en Matlab para generar la sintonización del controlador. Estas técnicas de sintonización pueden ser consideradas recientes en el área de control bajo lo cual es importante describir su alcance mediante la implementación a sistemas en los cuales es muy complejo el análisis y la sintonización por métodos numéricos o analíticos. Bajo esta perspectiva se brinda el desarrollo de los algoritmos basados en computación evolutiva para posteriormente explicar el funcionamiento de los algoritmos genéticos y de evolución diferencial utilizados en el presente trabajo y finalmente explicar los algoritmos basados en enjambres como es el caso de estudio del algoritmo de enjambre de partículas, utilizado del mismo modo para sintonizar el controlador del sistema dinámico en cuestión.

4.2.- MÉTODOS DE OPTIMIZACIÓN BASADOS EN COMPUTACIÓN EVOLUTIVA

Los algoritmos genéticos (AG) no son el único algoritmo de optimización basados en el enfoque de imitar los modelos de procesos naturales [4.1].

A continuación se desarrolla brevemente algunas de las metodologías basadas en algoritmos actuales que se están utilizando para la optimización global. Se busca dar respuesta a la pregunta, con base a la variación existente entre los distintos tipos de algoritmos evolutivos; cual es el mejor bajo ciertos parámetros. Con base a desarrollos e investigaciones se ha observado que el rendimiento promedio de todos los algoritmos de búsqueda con base a los problemas de estudio es igual (Wolpert, 1997).

- Algoritmo de recocido simulado
- Optimización por enjambre de partículas
- Algoritmo de colonia de hormigas
- Optimización de múltiples enjambres
- Algoritmo de colonia de abejas
- Algoritmo de altruismo
- Sistemas inmunológicos artificiales

- Algoritmo murciélago
- Algoritmo de búsqueda de sistema cargado
- Algoritmo de búsqueda Cuckoo
- Algoritmo de búsqueda diferencial
- Optimización por enjambre de luciérnagas
- Algoritmo de búsqueda gravitacional
- Caída inteligente de gotas de agua
- Algoritmo de optimización magnética
- Búsqueda de difusión estocástica
- Optimización por programación genética
- Optimización por algoritmos culturales
- Optimización por estrategias evolutivas
- Optimización por evolución diferencial
- Algoritmo de ranas saltarinas

4.3.- MÉTODO DE SINTONIZACIÓN DE CONTROL MEDIANTE AG (ALGORITMOS GENÉTICOS)

Las bases de los algoritmos genéticos (AG) fueron establecidas por John Holland en la universidad de Michigan en los años sesenta y la idea fue desarrollada en 1975 en su famoso libro "*adaptation in natural and artifitial systems*". Este tipo de algoritmos se caracteriza por el hecho de que el factor dominante para el propósito de la evolución es el cruce. La mutación solo es una forma de generar variedad en la población y presenta una baja probabilidad de ocurrencia [4.2].

El algoritmo genético desarrollado por Holland tiene la característica de presentarse en forma binaria. El cual resuelve muchos problemas relacionados con la teoría de optimización que resultan difíciles para las técnicas tradicionales, sin embargo posee una limitación de cuantificación, es decir cuando se trata de resolver problemas; por ejemplo; en donde los valores de las variables son continuas y se desea conocer a la precisión los valores de la variable; en tal problema cada variable requiere muchos bits para ser representada. Si el número de variables es grande, el tamaño del cromosoma también es grande.

Cuando las variables son continuas, es más lógico para que los represente en los números en forma flotante. El algoritmo genético continuo (GA) también tiene la ventaja de requerir menos espacio de almacenamiento que el algoritmo genético binario porque un solo número flotante representa la variable en lugar de varios bits. El algoritmo genético continuo es inherentemente más rápido que el binario, debido a que los cromosomas no tienen que ser decodificados antes de la

evaluación de la función de costo. Por las razones expuestas anteriormente el algoritmo desarrollado en el presente trabajo será genético continuo (GA).

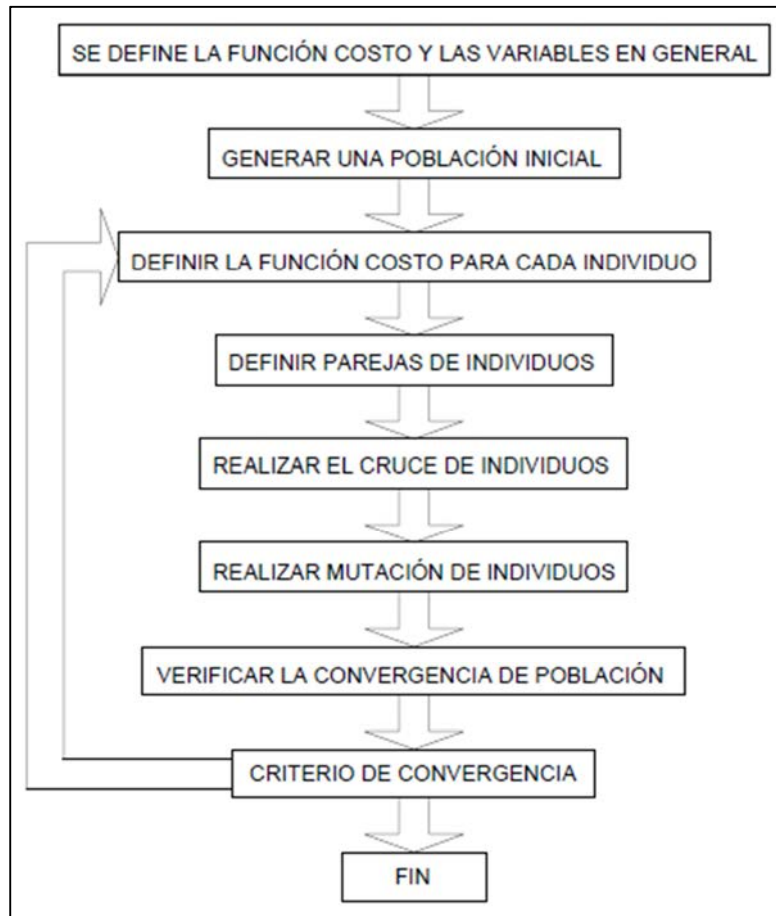


Fig. 4.1. Diagrama de flujo de la estructura del algoritmo genético continuo

La figura 4.1 ilustra la secuencia de pasos que se debe seguir para la construcción del algoritmo genético, el cual es desarrollado en el presente trabajo para el desarrollo del sintonizador del control del péndulo doble invertido.

Un problema importante que es muy relevante para la evolución de controladores cuando se usan AG es el denominado problema de "engañosidad". A menudo, un individuo que proviene del cruce de dos padres con alta calidad y que conserva los genes que determinan esa alta calidad presenta, sin embargo, una baja calidad. Cuando un problema es "deceptivo" es muy difícil para el AG obtener una solución óptima y su rendimiento se puede degradar hasta el nivel de una búsqueda aleatoria.

En cada generación, los cromosomas son evaluados usando alguna medida de aptitud. Las siguientes generaciones (nuevos cromosomas), son generadas

aplicando los operadores genéticos repetidamente, siendo estos los operadores de selección, cruzamiento, mutación y reemplazo.

4.3.1.- POBLACIÓN INICIAL

Se genera aleatoriamente la población inicial, que está constituida por un conjunto de cromosomas los cuales representan las posibles soluciones del problema. En caso de no hacerlo aleatoriamente, es importante garantizar que dentro de la población inicial, se tenga la diversidad estructural de estas soluciones para tener una representación de la mayor parte de la población posible o al menos evitar la convergencia prematura.

Para comenzar el GA, se define una población inicial de cromosomas (POP). La matriz representa la población con cada fila de valores continuos de forma aleatoria y distribuida en un dominio previamente definido. Dada una población inicial definida por el comando en Matlab:

$$pop = rand(Npop, Nvar)$$

4.3.2.- EVALUACIÓN

A cada uno de los cromosomas de esta población se aplicará un criterio de evaluación denominado en la función objetivo para saber, en base a los valores obtenidos en la población, cuál es su respectivo comportamiento en el sistema.

4.3.3.- SELECCIÓN

El método selección decide, una vez evaluada la población, y por lo tanto asignada una calidad a cada cromosoma, cuáles de ellos van a producir nuevos descendientes. Se podrían utilizar múltiples estrategias para seleccionar aquellos individuos que van a procrear. De hecho, el único criterio a tener en cuenta es que esta selección ha de proporcionar una ventaja a aquellos que mejor se adaptan a los objetivos.

Un método con una gran presión selectiva seleccionara al mejor individuo con una probabilidad muchísimo mayor que al peor. En un método con baja presión selectiva, aunque seleccionara al mejor individuo con más probabilidad que al peor, la diferencia de probabilidades de selección será pequeña.

Cuando hay muchísima presión selectiva, los genes de los individuos mejor adaptados acaban dominando la población de forma muy temprana, o, dicho de otra manera, los genes correspondientes a individuos de baja calidad desaparecen de la población rápidamente.

Independientemente del operador selección de individuos elegido, a menudo se utiliza también lo que se denomina elitismo, que significa que el mejor individuo de una generación pasa a la siguiente sin aplicar sobre él ningún operador genético. Esta posibilidad implica que, si la calidad no se ve afectada por factores variables de generación en generación, la calidad del mejor individuo de la población nunca será decreciente.

4.3.3.1.- SELECCIÓN POR RULETA

Cuando se utiliza el primer método elegido, a cada cromosoma se le asigna una casilla en una ruleta circular de un área equivalente a su calidad. Se gira la ruleta, la bola cae en una de las casillas y se selecciona el cromosoma correspondiente para reproducir.

De esta forma se emplea una selección proporcional a la calidad, porque el número de veces que un cromosoma es elegido para reproducción es igual a su calidad dividida entre la suma de calidad de toda la población. La Selección de ruleta (*roulette-wheel selection*) es también conocida como Selección proporcional a la función de desempeño (*fitness proportionate selection*). Esta selección permite que los mejores individuos sean elegidos con una mayor probabilidad, pero al mismo tiempo permite a los peores individuos ser elegidos, lo cual puede ayudar a mantener la diversidad de la población, en contraste con la selección por truncamiento.

Un problema de la selección de ruleta se presenta cuando existe una pequeña fracción de la población (en el límite, sólo un individuo) que posee una medida de desempeño excesivamente superior al resto. Esto provoca pérdida de diversidad y puede conducir a convergencia prematura pues la mayor parte de los individuos seleccionados será una copia de los pocos predominantes. En este caso es preferible utilizar selección basada en ranking o selección por torneo.

4.3.3.2.- SELECCIÓN POR TRUNCAMIENTO

En esta selección las soluciones candidatas son ordenadas según su función de desempeño, y una proporción p de los individuos con mejor desempeño es seleccionada y reproducida $1/p$ veces. Esta selección es menos sofisticada que la mayoría de los métodos de selección, y generalmente no es usada en la práctica.

4.3.3.3.- SELECCIÓN BASADA EN RANKING

En esta selección los individuos se ordenan según su medida de desempeño y luego son asignados con una segunda medida de desempeño, inversamente proporcional a su posición en el ranking (esto es, otorgando una mayor

probabilidad a los mejores). Los valores de esta segunda asignación pueden ser lineales o exponenciales.

Finalmente, los individuos son seleccionados proporcionalmente a esta probabilidad. Este método disminuye el riesgo de convergencia prematura que se produce cuando se utiliza selección de ruleta en poblaciones con unos pocos individuos con medidas de desempeño muy superiores a las del resto.

4.3.3.4.- SELECCIÓN POR TORNEO

El segundo método elegido para realiza la selección denominado selección por torneo, consiste en la realización de “torneos” para obtener los individuos que se van a procrear. Un torneo no es más que una selección aleatoria de un número de individuos de la población y de entre ellos procrea el que tiene mayor calidad.

Si estos torneos se realizan tantas veces como progenie se necesite, se obtiene una distribución de individuos proporcional a su calidad.

Esta selección se efectúa mediante un torneo (comparación) entre un pequeño subconjunto de individuos elegidos al azar desde la población. Los beneficios de este tipo de selección son la velocidad de aplicación (dado que no es necesario evaluar ni comparar la totalidad de la población) y la capacidad de prevenir, en cierto grado, la convergencia prematura. La principal desventaja es la necesidad de establecer el parámetro correspondiente al tamaño del subconjunto.

4.3.4.- CRUCE

Una vez seleccionados los progenitores se tiene que realizar el quinto paso en el esquema de la figura 3 esto es, combinar el material genético de los progenitores para producir un descendiente por medio del operador cruce.

Existen diversas formas de realizar el cruce, una de ellas es seleccionar un punto de cruce en los cromosomas de dos individuos e intercambiar estos formando dos individuos descendientes con la mezcla de los cromosomas de los individuos padres.

Son posibles otras formas de cruce como utilizar más de un punto para intercambiar el material genético o intercambiar de modo alterno los genes de ambos progenitores.

4.3.5.- MUTACIÓN

El operador mutación cambia el contenido del material genético de un determinado cromosoma. En el caso de la decodificación binaria el único cambio posible es la

inversión del bit correspondiente, pero usando otra codificación diferente, como números reales, los posibles valores después de una mutación serán infinitos. Las mutaciones representan, por tanto, variaciones en el gen o genes de un cromosoma para así conseguir cromosomas cercanos a uno dado que muta, pero con la posibilidad de que dicha variación pueda incrementar de modo considerable la calidad del mismo.

Con base a los pasos anteriormente descritos se logran obtener poblaciones de individuos que, conforme el algoritmo genético continuo va evolucionando se acercan más a una solución óptima, la cual será convergente bajo un criterio de error aceptable definido por el programador o en base a las necesidades planteadas.

4.4.- MÉTODO DE SINTONIZACIÓN DE CONTROL MEDIANTE ED (EVOLUCIÓN DIFERENCIAL)

Idealmente, la solución de un problema de optimización difícil no debe ser en sí difícil, por ejemplo, un ingeniero estructural experto, con un conocimiento de principios de mecánica no deberían también tienen que ser un experto en teoría de optimización sólo para mejorar sus diseños. Además de ser fácil de usar, un algoritmo de optimización global también debe ser lo suficientemente potente como para converger de forma fiable al óptimo verdadero. Además, el tiempo dedicado a la computadora en busca de una solución no debe ser excesivo. Por lo tanto, un método de optimización global verdaderamente útil debe ser sencillo de aplicar, fácil de usar, fiable y rápido [4.3].

La metodología de optimización denominada evolución diferencial (DE) es un método de este tipo. Desde su creación en 1995, DE se ha ganado una reputación como un optimizador mundial muy eficaz. Su récord de rendimiento fiable y robusto exige que cada ingeniero y científico deba conocerlo. DE se originó con el algoritmo de recocido genético desarrollado por *Kenneth Price* y publicado en la edición de *Dr. Dobbs* de octubre 1994, en la revista de un programador popular.

El algoritmo de recocido genético está basado en la población, el cual es un algoritmo de optimización combinatoria que implementa un criterio de recocido a través de umbrales. Después de que el algoritmo de recocido genético aparecido en DDJ, Ken fue contactado por el *Dr. Rainer Storn*, (entonces con Siemens, mientras que en el Instituto Internacional de Ciencias de la Computación en la Universidad de California en Berkeley; ahora en *Rohde & Schwarz GmbH, Munich*, Alemania) sobre la posibilidad de utilizar recocido genético para resolver el problema de ajuste polinomio de Chebyshev. La determinación de los coeficientes

de los polinomios de Chebyshev es considerado por muchos como una tarea difícil para un optimizador de propósito general.

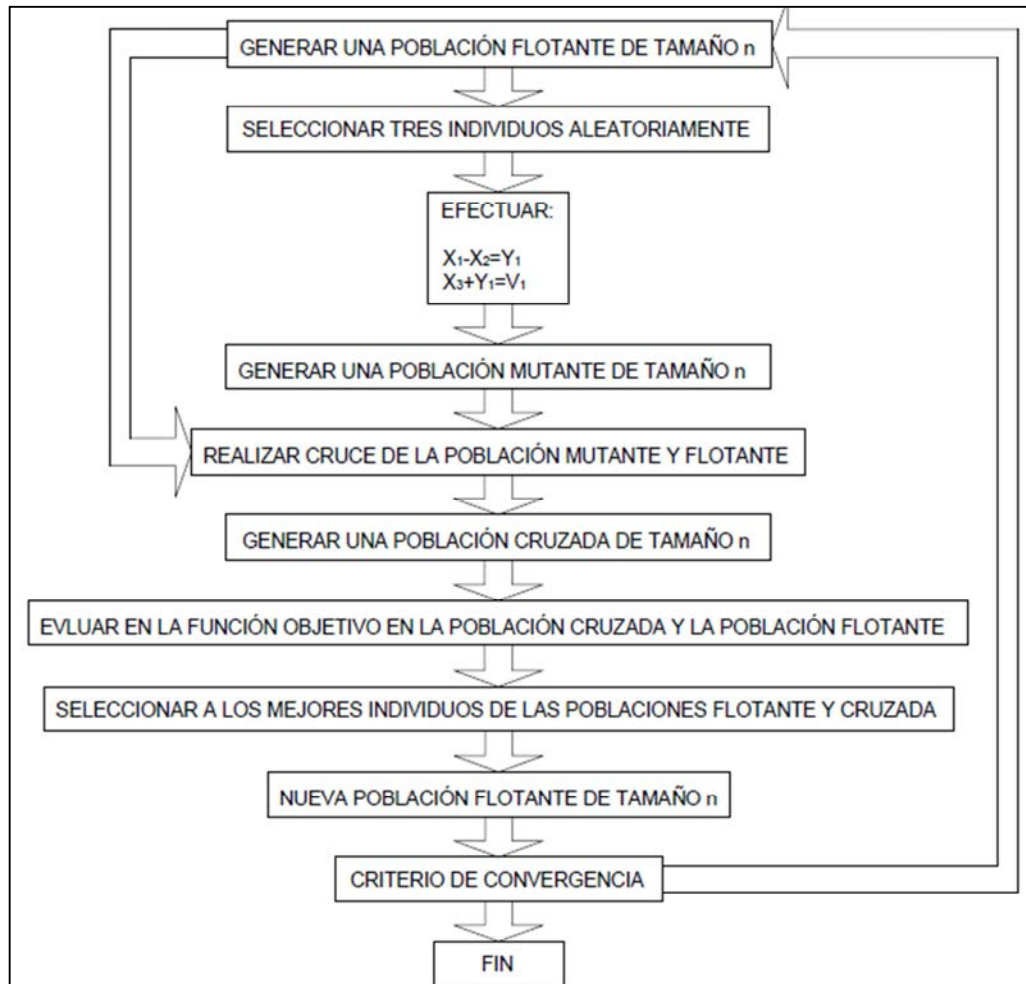


Fig. 4.2. Diagrama de flujo de la estructura del algoritmo evolución diferencial

Price y Storn desarrollaron el método de evolución diferencial (DE) el cual logro ser una función de optimización fiable y versátil que también es fácil de usar. La primera publicación escrita sobre DE apareció como un informe técnico en el año 1995 (Storn y Price 1995). Desde entonces, DE ha sido probado en competencias como Concurso Internacional de la IEEE sobre optimización evolutiva (ICEO) en 1996 y 1997 y en el mundo real en una amplia variedad de aplicaciones. Recientemente, el software Mathematica añadió DE para su paquete optimizador numérico.

Como casi todos los algoritmos evolutivos, DE es un optimizador basado en la población que ataca el problema comenzando del punto de muestreo de la función objetivo en múltiple, al azar elegido puntos iniciales. Límites de los parámetros

preestablecidos definen el dominio de la que se eligen los vectores. La figura 4.2 ilustra el esquema a seguir para desarrollar el algoritmo de evolución diferencial utilizado en la presente sintonización de control retroalimentado.

Se describen a continuación cada una de las etapas de interés en el desarrollo del algoritmo de evolución diferencial para aclarar el funcionamiento de este. El cual es desarrollado en el diagrama de flujo mostrado anteriormente.

4.4.1.- POBLACIÓN INICIAL

Antes de que la población se pueda inicializar, ambos límites, superior e inferior deben ser especificados.

Estos valores pueden tomar diversas formas, entre otras pueden formarse en dos vectores de inicialización BL y Bu, para el que los subíndices L y U indican los límites inferior y superior, respectivamente. Una vez que los límites de inicialización se han especificado, un generador de números aleatorios asigna a cada parámetro de cada vector un valor dentro del rango establecido. Una de las estructuras que puede tomar esta instrucción es:

$$x_i = \text{rand}(0,1)[BU_i - BL_i] + BL_i$$

El generador de números aleatorios, rand, devuelve una distribución uniformemente de número aleatorio desde dentro de la gama (0,1), es decir, $0 \leq \text{rand}_j(0,1) < 1$. El subíndice j, indica que un nuevo valor aleatorio se genera para cada parámetro.

Incluso si una variable es discreta o continua, debe ser inicializado con un valor real, ya que DE trata internamente todas las variables como valores de punto flotante, independientemente de su tipo.

4.4.2.- MUTACIÓN

Una vez inicializado, DE muta y recombina la población para producir una población de vectores de prueba. En particular, la mutación diferencial añade una escala de muestreo aleatorio, la cual genera un segundo vector. La ecuación siguiente muestra cómo combinar tres vectores diferentes, elegidos al azar de entre la población original creada para crear un vector mutante, v_{ig}

$$v_{ig} = x_{i,G} + F(x_{i,G} - x_{i+n,G})$$

El factor de escala, $F \in (0,1+)$, es un número real positivo que controla la velocidad en la que la población evoluciona. Si bien no hay límite superior en F, a partir de los valores, rara vez es superior a 1.

4.4.3.- CRUCE

Como complemento a la estrategia de búsqueda de mutación diferencial, DE también emplea un cruce uniforme. A veces se refiere a la recombinación como valores discretos, el operador cruce construye vectores de prueba de valores de los parámetros que han sido copiados a partir de dos vectores diferentes. En particular, DE cruza cada vector con un vector mutante.

La probabilidad de cruce, es un valor definido por el usuario que controla la fracción de los valores de los parámetros que se copian de los mutantes a designar la fuente que contribuye un parámetro, el cruce uniforme compara a la salida de un generador de números aleatorios uniforme. Si el número aleatorio es menor o igual a algún valor preestablecido, el parámetro de ensayo se hereda de los mutantes v_{ig} ; de lo contrario, el parámetro se copia del vector, x . Además, el parámetro de ensayo con índice al azar elegido, se toma del mutante para asegurar que el vector de prueba no duplique al vector seleccionado de la población original.

4.4.4.- SELECCIÓN

La selección de los individuos se realiza en comparación con los individuos de la población original y la población cruzada, si el individuo cruzado tiene un valor de la función objetivo igual o mayor al de su correspondiente elemento en la posición de la población original se sustituye al elemento de la población original por el de la población cruzada, de lo contrario, el elemento de la población original conserva su lugar en la población por lo menos un más generación más.

Mediante la comparación de cada elemento de ensayo evaluado en la función objetivo, la formula siguiente ilustra el proceso seguido por DE:

$$x_i = \begin{cases} u_i & \text{si } f(u_i) \geq f(x_i) \\ x_i & \text{en caso contrario} \end{cases}$$

Una vez que la nueva población está definida, el proceso de mutación, recombinación y selección se repite hasta que el óptimo es encontrado, o un bajo algún criterio predefinido de terminación se cumple, por ejemplo, el número de generaciones alcanza un máximo preestablecido, o un error menor a algún valor es alcanzado, etc.

4.5.- MÉTODO DE SINTONIZACIÓN DE CONTROL MEDIANTE PSO (OPTIMIZACIÓN POR ENJAMBRE DE PARTÍCULAS)

La Inteligencia de enjambres, que corresponde a un grupo de técnicas que están basadas en el estudio del comportamiento colectivo en sistemas auto organizados

y descentralizados (distribuidos). Estos sistemas están conformados típicamente por una población de agentes computacionales simples capaces de percibir y modificar su ambiente de manera local. Tal capacidad hace posible la comunicación entre los individuos, que detectan los cambios en el ambiente generado por el comportamiento de sus semejantes. Aunque normalmente no hay una estructura centralizada de control que dictamina cómo los agentes deben comportarse, las interacciones locales entre los agentes usualmente llevan a la emergencia de un comportamiento global [4.4].

La optimización por enjambre de partículas (PSO), desarrollada por Kennedy y Eberhart (1995), es un método de optimización para funciones no lineales en espacios continuos y discretos, basado en la simulación de un modelo social simple del desplazamiento de cardúmenes y bandadas.

En un sistema PSO, la búsqueda se realiza utilizando una población de partículas que corresponden a los individuos, cada uno de los cuales representa una solución candidata al problema. Las partículas cambian su estado al “volar” a través del espacio de búsqueda hasta que se ha encontrado un estado relativamente estable. Un sistema PSO combina un modelo “únicamente social”, el cual sugiere que los individuos ignoran su propia experiencia y ajustan su conocimiento de acuerdo a las creencias exitosas de los individuos en la vecindad; y un modelo “únicamente cognitivo”, el cual trata a los individuos como seres aislados. Una partícula cambia de posición utilizando estos dos modelos.

Dentro de PSO, la i -ésima partícula es tratada como un punto dentro de un espacio N -dimensional, representada por $X_i = (x_{i1}, x_{i2}, \dots, x_{iN})$. La mejor posición encontrada por la partícula anteriormente, o sea aquella donde se obtuvo el mejor valor en la función de costo, es representada como $p_i = (p_{i1}, p_{i2}, \dots, p_{iN})$. La mejor posición encontrada por el total de la población es representada por el símbolo g . La tasa de cambio de posición (velocidad) para una partícula i es representada como $v_i = (v_{i1}, v_{i2}, \dots, v_{iN})$.

La primera ecuación mostrada a continuación es utilizada para calcular la nueva velocidad de la partícula de acuerdo a su velocidad anterior y las distancias de su posición actual a su mejor posición y la mejor posición dentro del grupo. Luego la partícula se desplaza hacia una nueva posición de acuerdo a la segunda ecuación mostrada.

$$v_{in}(t+1) = wv_{in}(t) + c_1R_1[p_{in} - x_{in}(t)] + c_2R_2[p_{gn} - x_{in}(t)] \quad (1)$$

$$x_{in}(t+1) = x_{in}(t) + v_{in}(t+1) \quad (2)$$

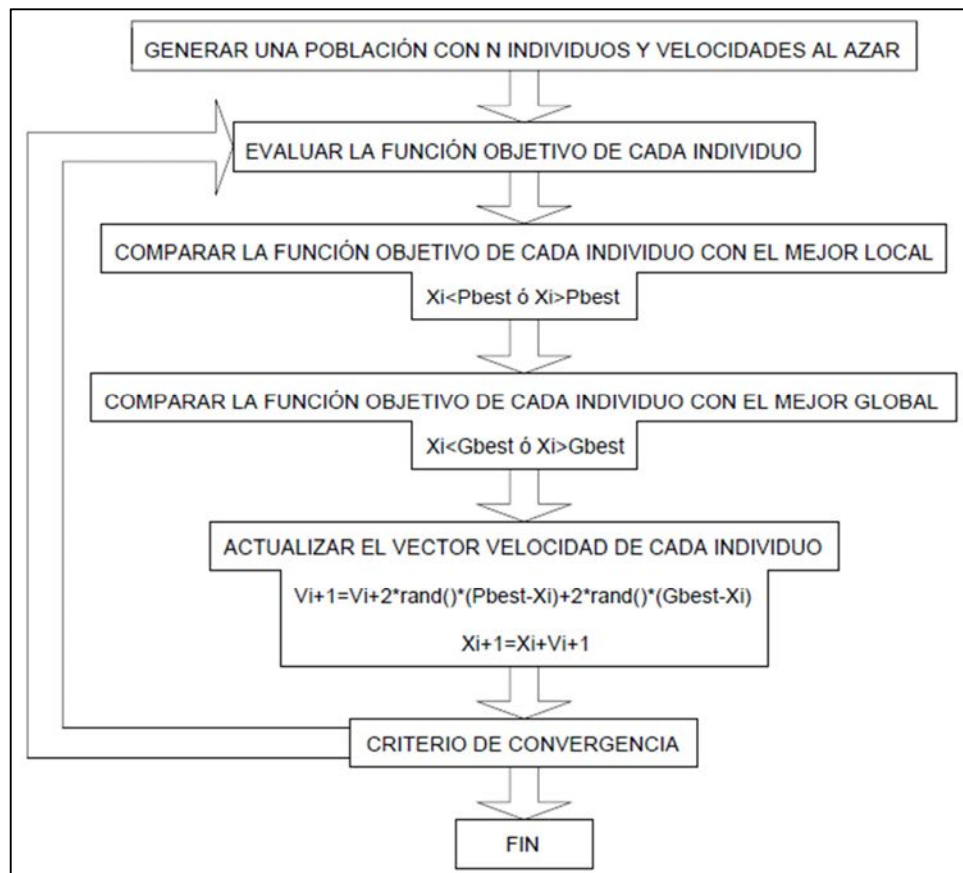


Fig. 4.3. Diagrama de flujo de la estructura del algoritmo PSO

El peso inercial w , formulado por Shi y Eberhart (1998), es utilizado para controlar el impacto de las velocidades previas en la velocidad actual, influenciando el cambio entre las habilidades de exploración global (rango amplio) o local (rango corto) de las partículas. Un peso inercial mayor facilita la exploración global para buscar nuevas áreas, mientras que un peso inercial menor tiende a facilitar exploración local para sintonizar finamente el área de búsqueda actual. Una correcta selección del peso inercial puede proveer un balance entre las habilidades de búsqueda local y global, por lo tanto requiere menores iteraciones para encontrar el punto óptimo.

Conceptualmente el algoritmo de PSO es bastante simple, sin embargo existe interés en encontrar maneras de mejorar el desempeño del algoritmo por medio de varias técnicas. La primera modificación la realizaron Eberhart y Kennedy (1995) al cambiar el concepto vecindad, seleccionando en lugar de la mejor posición global pg , la mejor posición de las m partículas más cercanas (pl). Clerc (1999), hace uso de un factor de constricción K que ayuda a la convergencia del algoritmo PSO al evitar que las partículas detengan su movimiento.

Otros campos de investigación sobre el algoritmo PSO corresponden al análisis matemático o estadístico para conocer mejor el comportamiento del sistema. También se buscan nuevas aplicaciones del algoritmo donde se busca obtener mejores resultados que con otros métodos de optimización. El diagrama de flujo ilustrado en la figura 4.3 muestra el comportamiento que toma el algoritmo por enjambre de partículas desarrollado. El algoritmo por enjambre de partículas junto con la población generada debe de cumplir con ciertas cualidades en general:

- Proximidad: Capacidad para realizar cálculos de tiempo y espacio.
- Calidad: Capacidad para responder a los factores de calidad ambiental.
- Respuesta: Habilidad para producir una pluralidad de respuestas diferentes.
- Estabilidad: Capacidad para mantener comportamientos robustos ante cambios ambientales leves.
- Adaptabilidad: Posibilidad de cambiar el comportamiento cuando está dictada por factores externos.

4.6.- CASO DE APLICACIÓN AL MODELO DINÁMICO NO LINEAL DEL PÉNDULO DOBLE INVERTIDO

En base a la teoría desarrollada anteriormente se presenta a continuación la propuesta de solución para efectuar la sintonización del control retroalimentado del péndulo doble.

La figura 4.4 ilustra la propuesta de solución, bajo la cual se establecerán las sintonizaciones bajo cada una de las técnicas de sintonización planteadas, se muestra que en primer instancia el control se realiza basados en una referencia, bajo la cual se empezara a generar poblaciones de individuos que serán evaluados para posteriormente aplicar cada operación correspondiente a la metodología utilizada (Genético continuo, Evolución diferencial, Optimización por enjambre de partículas), es decir mutación, cruzamiento, para posteriormente efectuar la selección bajo los criterios definidos, a los mejores individuos que minimicen el error con la referencia establecida y la posición que ofrecen estos.

Una vez establecida la teoría referente al desarrollo de los algoritmos evolutivos para obtener la sintonización del control por retroalimentación de estados del péndulo doble invertido y establecido la lógica del algoritmo bajo el cual se operan los mismos, así como el esquema de solución planteado; se procede a desarrollar mediante Matlab cada algoritmo de sintonización.

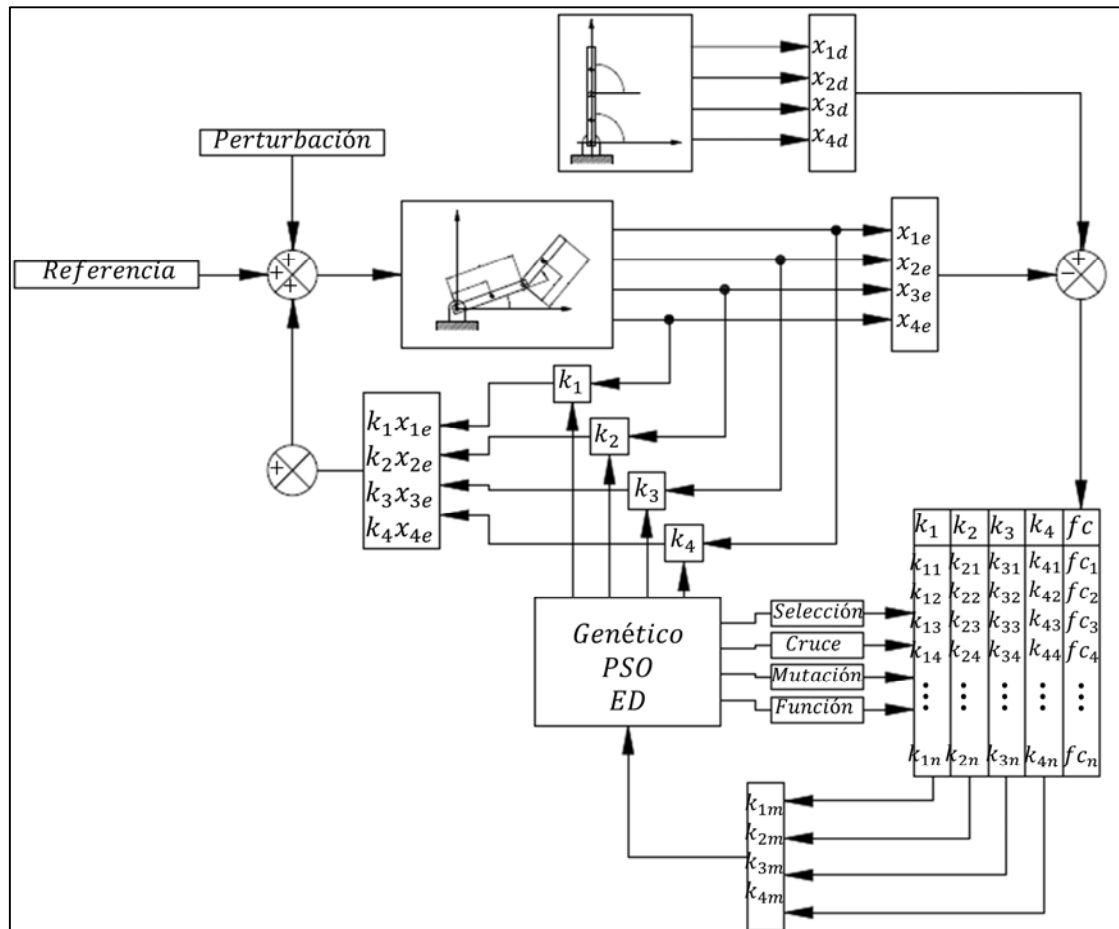


Fig. 4.4. Control por retroalimentación de estados del péndulo doble invertido

La propuesta de solución mostrada anteriormente sigue el siguiente procedimiento para cada método de sintonización seleccionado:

1. Inicialmente se define la posición de referencia del modelo del sistema dinámico de estudio, así como los parámetros mecánicos del mismo, como las condiciones iniciales del modelo.
2. El algoritmo en cuestión genera una población inicial con un determinado número de individuos y con base a las características definidas anteriormente, según sea la metodología empleada.
3. La población inicial es definida como ganancias k_1, k_2, k_3, k_4 y son introducidas al sistema de control acoplado al sistema dinámico.
4. Estas ganancias en el sistema dinámico generan una respuesta del mismo, la cual es comparada con la referencia o valor deseado del sistema.
5. Esta evaluación es medida por el algoritmo en cuestión y es definida como la función objetivo, la cual es óptima bajo la premisa de tender a cero, es decir minimizar el error en el sistema.

6. Con base a la evaluación efectuada, el algoritmo ordena a los individuos de acuerdo a un criterio de ordenamiento y aplica las operaciones de selección, cruce, mutación, dependiendo del caso correspondiente.
7. Con base a las operaciones anteriores se genera una nueva población la cual es nuevamente introducida al modelo dinámico de estudio.
8. El paso 3 al 7 se repite hasta que se alcanza un criterio de convergencia o se determina un número total de evoluciones en el algoritmo computacional.

Bajo el proceso anterior y con base a los algoritmos definidos en este capítulo se inicia el proceso de sintonización por medio de computación evolutiva. En los apéndices se mencionan los algoritmos generados en Matlab para la implementación del proceso.

REFERENCIAS

- [4.1] S. Nesmachnow, Aplicación de las técnicas de procesamiento paralelo a la implementación de Algoritmos Genéticos. Technical Report INCO 06-02, Universidad de la República, Uruguay, 2002.
- [4.2] Practical genetic algorithms, second edition, Randy L. Haupt, Sue Ellen Haupt a John Wiley & Sons, inc., publication.
- [4.3] Differential Evolution A Practical Approach to Global Optimizatio. Kenneth V. Price. Rainer M. Storn. Jouni A. Lampinen
- [4.4] Particle Swarm Optimization and Intelligence: Advances and Applications. Konstantinos E. Parsopoulos. University of Ioannina, Greece. Michael N. Vrahatis. University of Patras, Greece.

CAPÍTULO 5

DESARROLLO DE PRUEBAS CON ALGORITMOS EVOLUTIVOS

Resumen:

En el presente capítulo se desarrollan las pruebas dinámicas del comportamiento del modelo del péndulo doble invertido basados en el modelo computacional y matemático analizado en el capítulo 3. Bajo la metodología planteada en el capítulo 4 basada en algoritmos evolutivos y los pasos que los diferentes métodos de sintonización deben de tomar para efectuar el control del sistema dinámico no lineal, se muestran en el presente capítulo los resultados del desarrollo de la sintonización de los mismos, primeramente el algoritmo genético, posteriormente el algoritmo de evolución diferencial y finalmente la sintonización por medio del algoritmo de optimización por enjambre de partículas. En los apéndices se muestra la programación desarrollada en Matlab para efectuar las simulaciones computacionales.

5.1.- INTRODUCCIÓN

En los apéndices se puede observar la implementación de los algoritmos mostrados en el capítulo anterior en entorno Matlab, se desarrolla la simulación de cada uno de los algoritmos de estudio mediante una corrida de simulación, el código se ejecuta tal y como se presenta en los apéndices y este es ligado a algunos otros programas ahí presentados por lo que se recomienda que todos los códigos se ejecuten tal y como vienen en los mismos y sean guardados en una carpeta conjunta. Se muestran las imágenes en forma secuencial conforme el algoritmo va evolucionando en un entorno característico que es proporcionado por el modelo de prueba desarrollado anteriormente, los algoritmos muestran todo el grupo de familias de soluciones. Las líneas muestran el comportamiento de todas las soluciones presentes en el sistema para cada tipo de evolución tratada. Es por ello que se ve en las imágenes un comportamiento tendiente a la inestabilidad de la solución dado que muchas ganancias no logran sintonizar el sistema dinámico.

5.2.- ANALISIS DE CONTROL NO LINEAL EN EL MODELO DINÁMICO BASADO EN ALGORITMOS GENETICOS (AG)

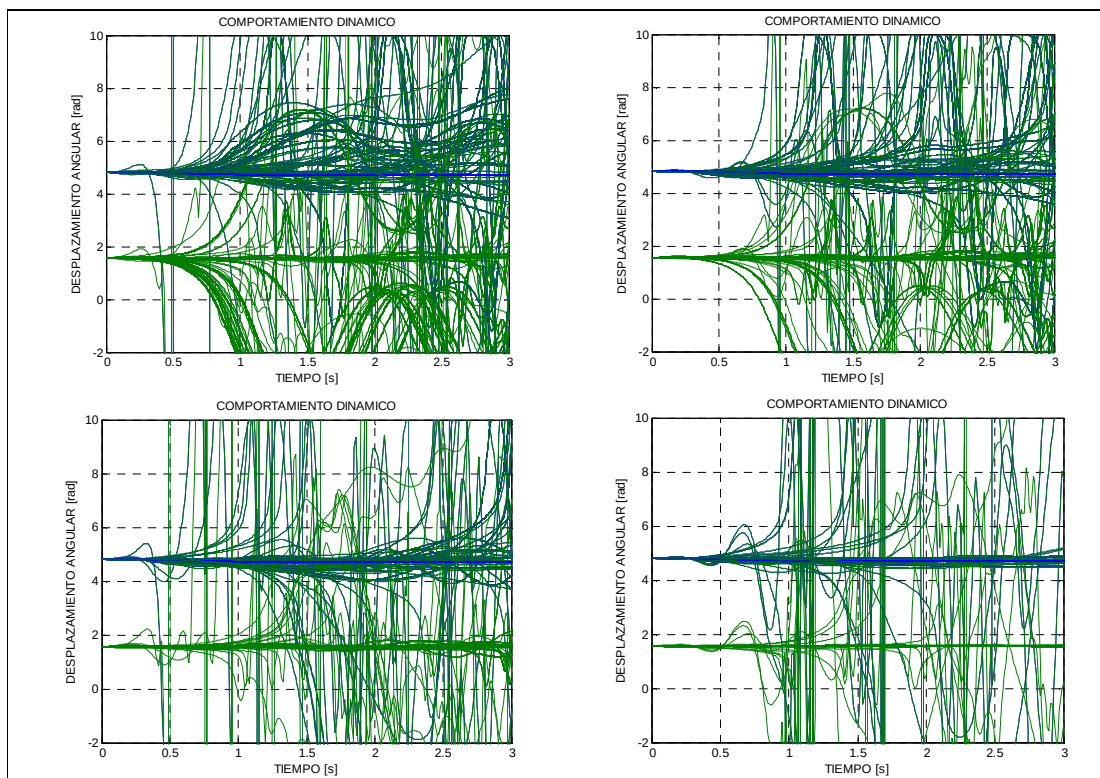
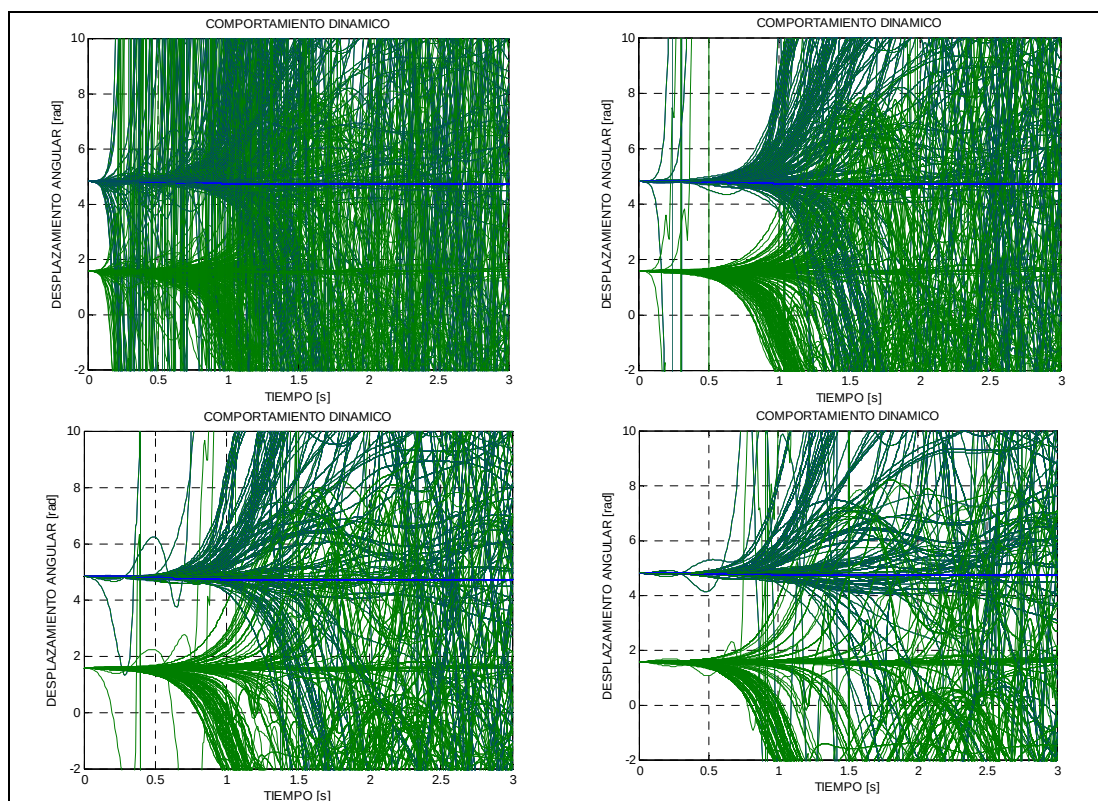
Basado en los algoritmos desarrollados y mostrados en los apéndices A, se plantea el comportamiento del péndulo doble invertido sometido a sintonización por medio de algoritmos genéticos, en el presente capítulo se desarrollan las simulaciones de la búsqueda de ganancias optimas donde se observa el criterio de convergencia del mismo.

El sistema de sintonización parte de parámetros definidos de entrada, los cuales son:

$$\begin{aligned} \text{No. de individuos} &= 300 \\ \text{No. de ganancias} &= 4 \\ \text{No. evoluciones} &= 20 \\ \text{Porcentaje de mutacion} &= 7\% \\ \text{Tiempo final} &= 3 \text{ s} \\ \text{Ancho de ntervalo} &= 0.1 \end{aligned}$$

Con base a estos parámetros el algoritmo genético empieza a realizar iteraciones entre los individuos buscando valores óptimos que logren sintonizar el comportamiento del mismo.

En las figuras 5.1 a 5.5 se muestra cada una de las evoluciones donde está graficando el comportamiento de 300 individuos del sistema, algunos de los cuales tienen propiedades bajo las cuales se puede controlar el sistema.



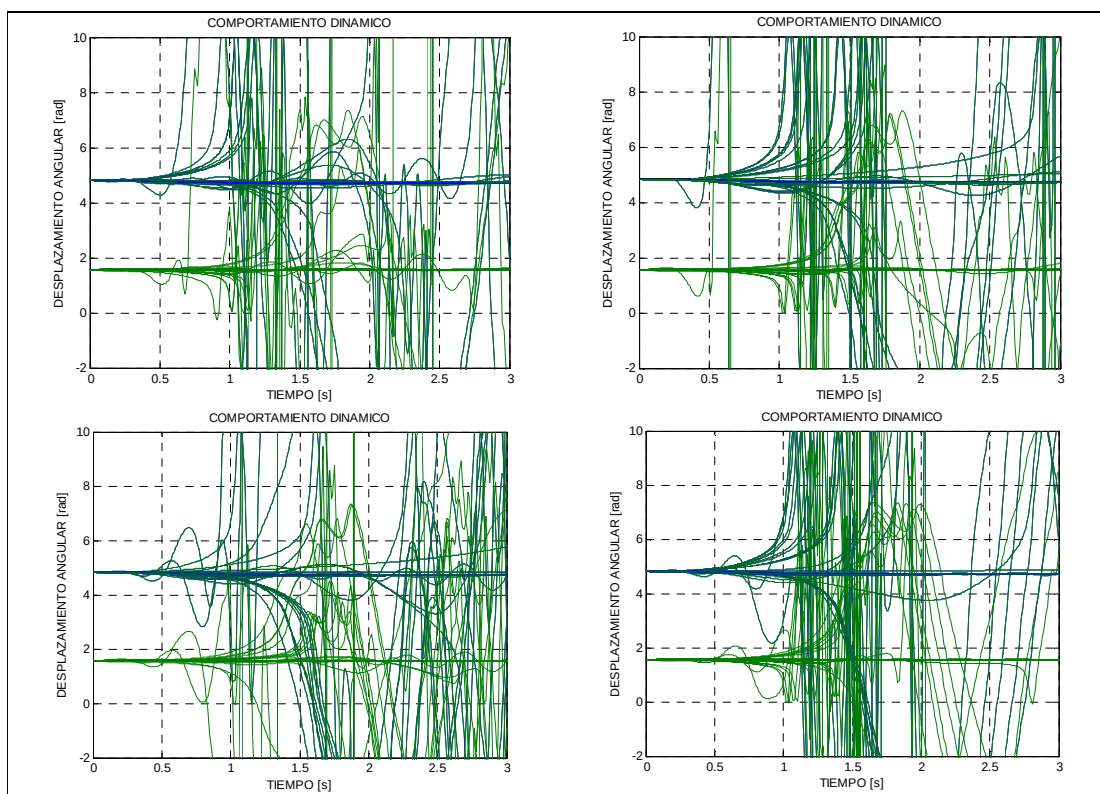


Fig. 5.3. Iteración 9–12 de búsqueda de solución óptima

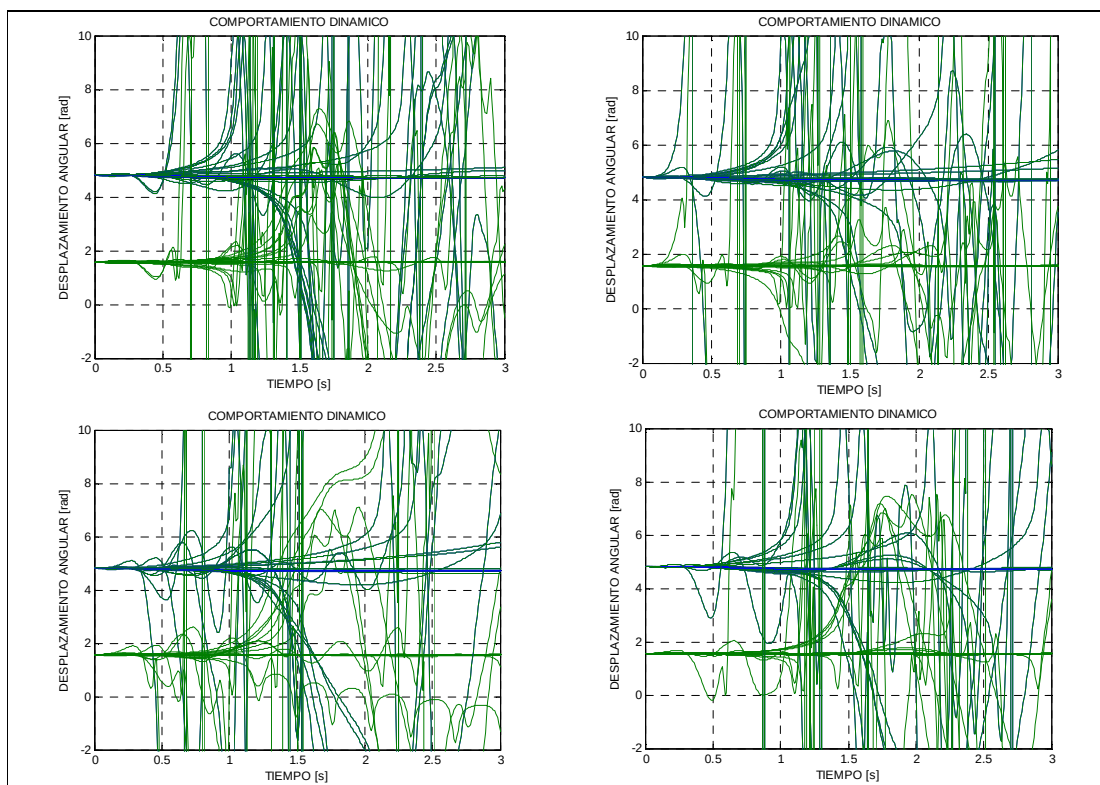


Fig. 5.4. Iteración 13–16 de búsqueda de solución óptima

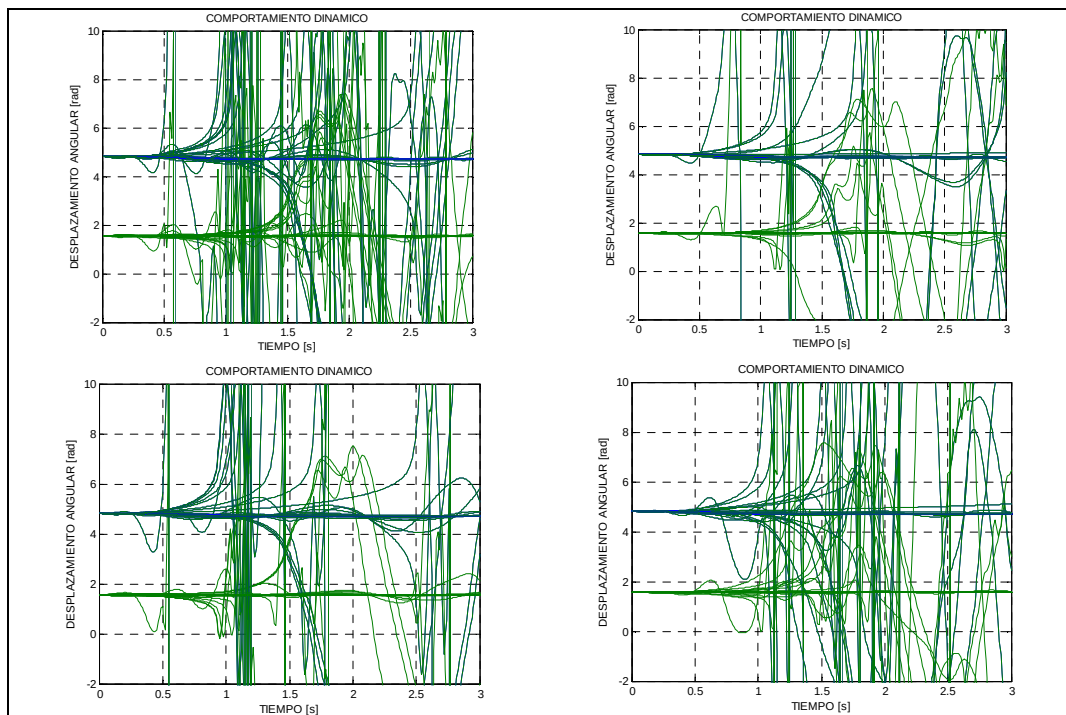


Fig. 5.5. Iteración 17–20 de búsqueda de solución óptima

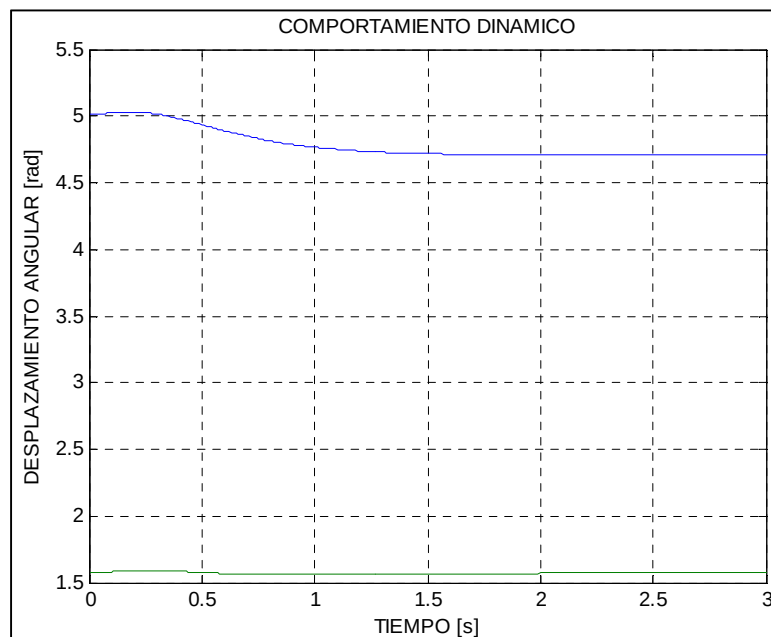


Fig. 5.6. Solución óptima: algoritmo genético

Con base a las iteraciones generadas se obtienen las ganancias que logran controlar el sistema dinámico no lineal del péndulo doble invertido como lo ilustra la figura 5.6. Las cuales son:

$$\begin{aligned}
 k_1 &= -0.6952 \\
 k_2 &= 2.98550 \\
 k_3 &= -0.1458 \\
 k_4 &= 0.46500
 \end{aligned}$$

5.3.- ANÁLISIS DE CONTROL NO LINEAL EN EL MODELO DINÁMICO BASADO EN EVOLUCIÓN DIFERENCIAL (ED)

Se procede a desarrollar el control retroalimentado por medio del algoritmo de evolución diferencial basado en el diagrama de flujo del capítulo 4, primeramente se definen los parámetros de prueba del mismo como:

$$\begin{aligned}
 \text{No. de individuos} &= 300 \\
 \text{No. de ganancias} &= 4 \\
 \text{No. evoluciones} &= 20 \\
 \text{Tiempo final} &= 3 \text{ s} \\
 \text{Longitud de intervalo} &= 0.1 \\
 \text{Tasa de mutacion} &= 7\%
 \end{aligned}$$

Estos parámetros alimentan el algoritmo generado en el entorno Matlab, el cual es mostrado en el anexo A-9, se desarrollan pruebas sobre los parámetros descritos.

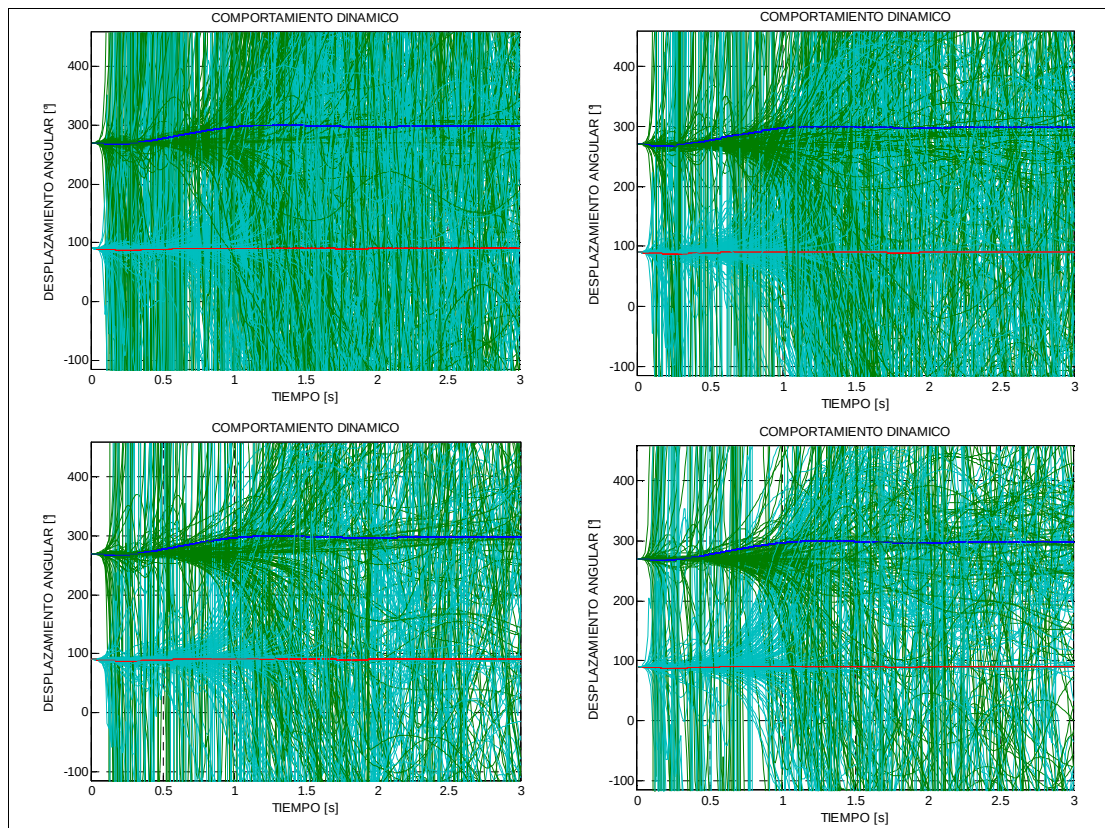


Fig. 5.7. Iteración 1–4 de búsqueda de solución óptima

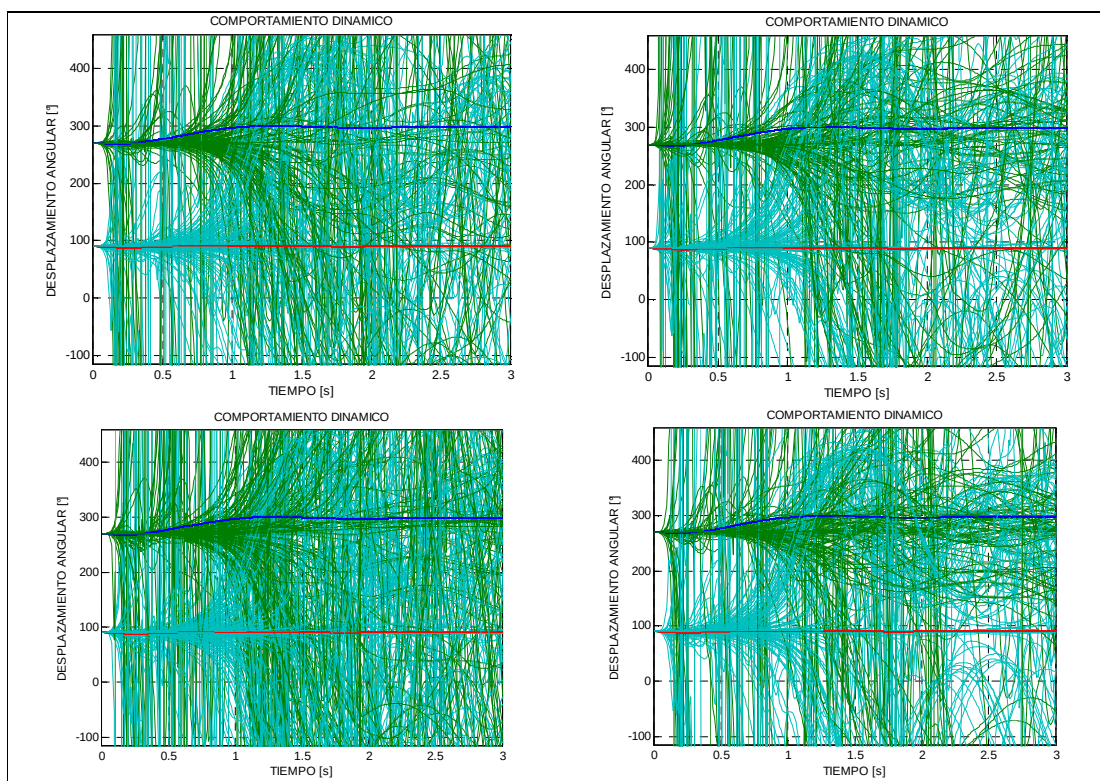


Fig. 5.8. Iteración 5–8 de búsqueda de solución óptima

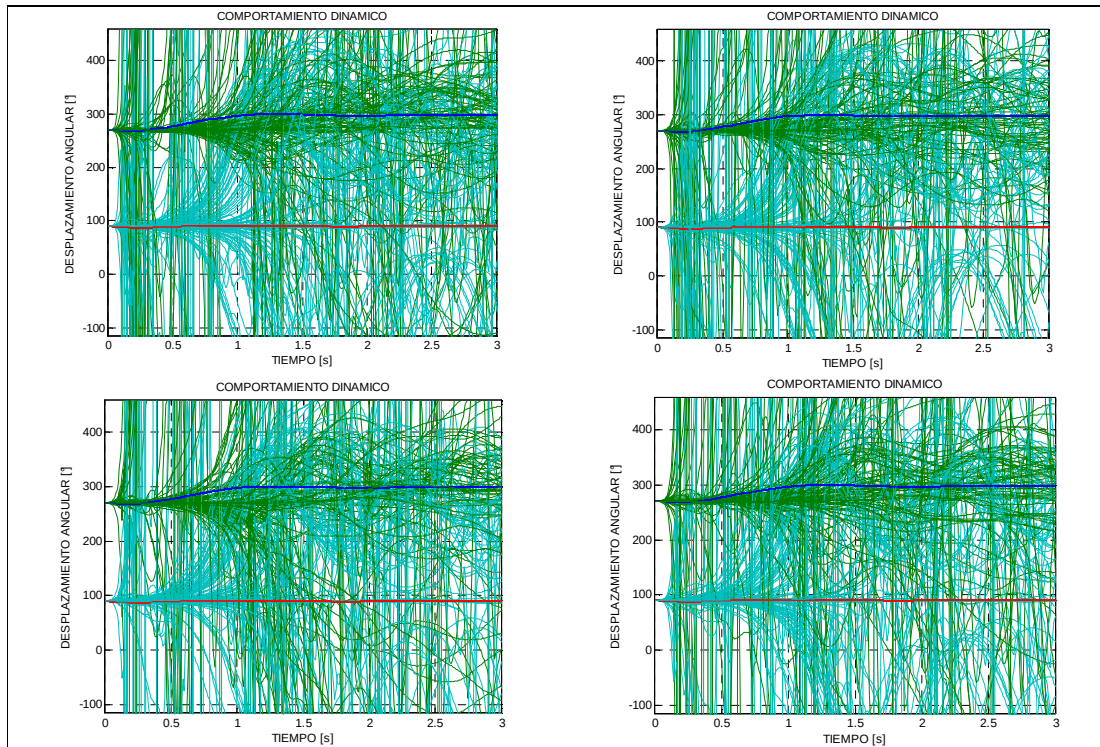


Fig. 5.9. Iteración 9–12 de búsqueda de solución óptima

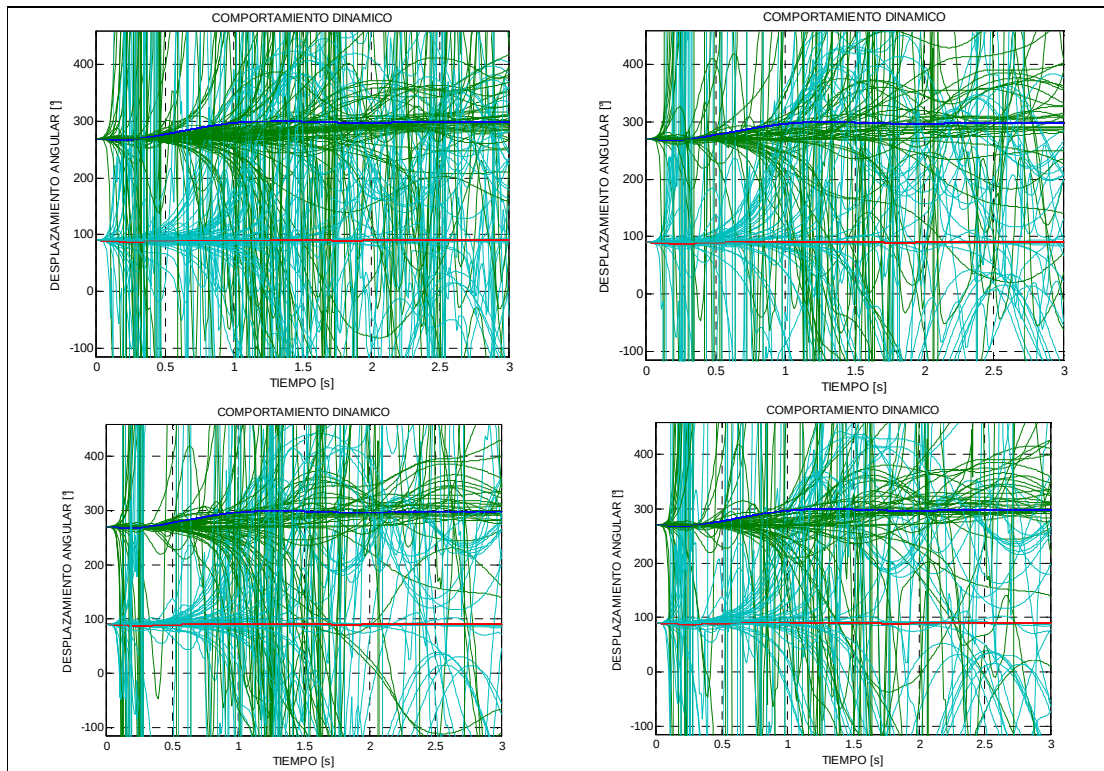


Fig. 5.10. Iteración 13–16 de búsqueda de solución óptima

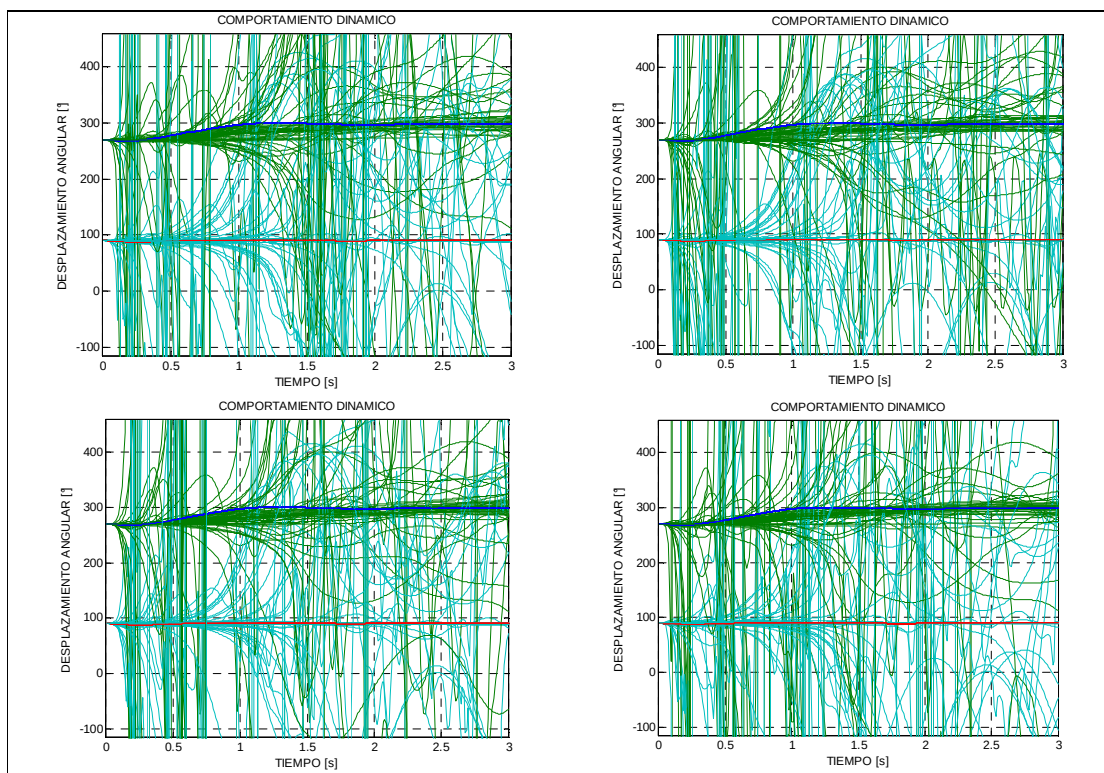


Fig. 5.11. Iteración 17–20 de búsqueda de solución óptima

Las ganancias que, en esta simulación lograron controlar al sistema dinámico son:

$$k_1 = -0.7389$$

$$k_2 = 3.32320$$

$$k_3 = -0.1540$$

$$k_4 = 0.29300$$

La gráfica del sistema controlado mediante estas ganancias es mostrada en la figura 5.12 y las iteraciones del algoritmo en las figuras 5.7 a 5.11.

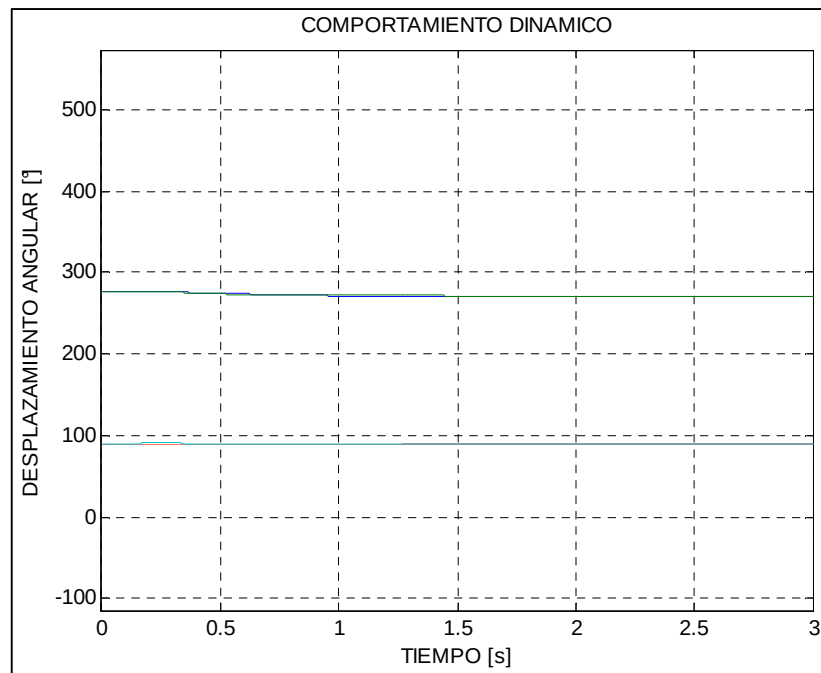


Fig. 5.12. Solución óptima: evolución diferencial

5.4.- ANÁLISIS DE CONTROL NO LINEAL EN EL MODELO DINÁMICO BASADO EN OPTIMIZACIÓN POR ENJAMBRE DE PARTICULAS (PSO)

El siguiente algoritmo a desarrollar corresponde a la optimización por enjambre de partículas cuyo algoritmo de sintonización es presentado en el anexo A-10, con base a los parámetros definidos a continuación se muestran las iteraciones para localizar las ganancias.

$$\text{No. de individuos} = 300$$

$$\text{No. de ganancias} = 4$$

$$\text{No. evoluciones} = 20$$

$$\text{Porcentaje de mutacion} = 7\%$$

$$\text{Tiempo final} = 3 \text{ s}$$

$$\text{Ancho de intervalo} = 0.1$$

$$\text{Tasa de mutacion} = 7\%$$

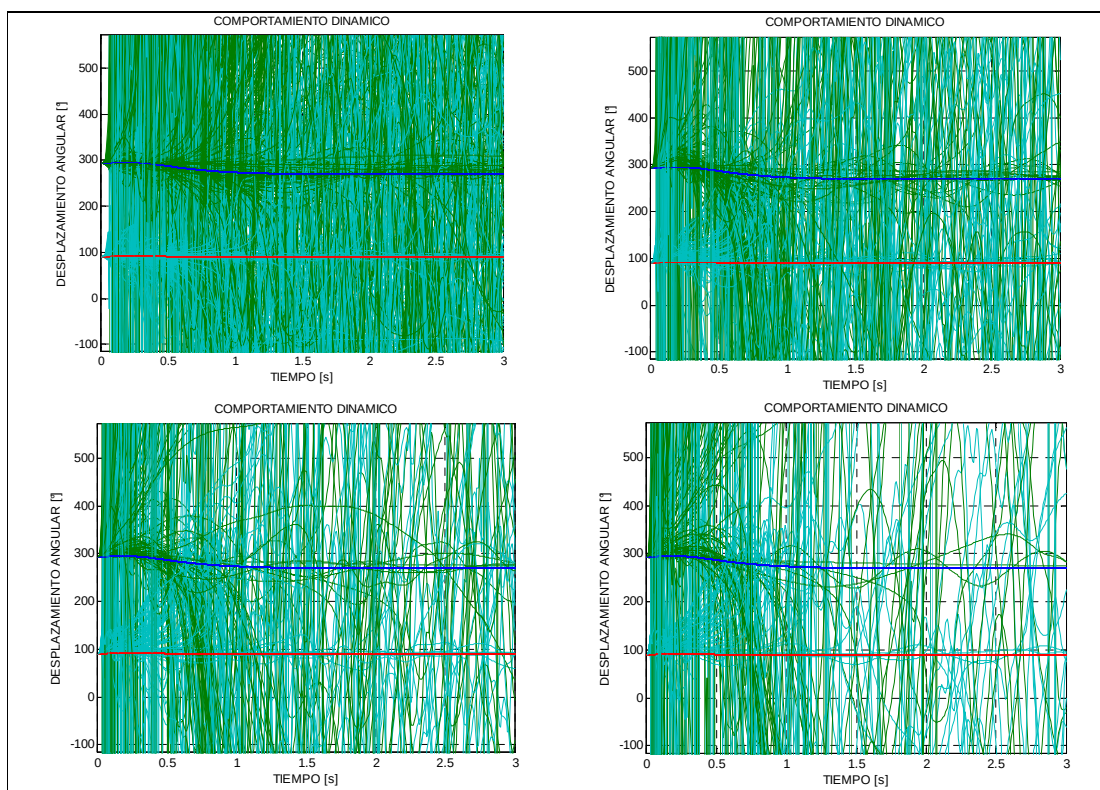


Fig. 5.13. Iteración 1–4 de búsqueda de solución óptima

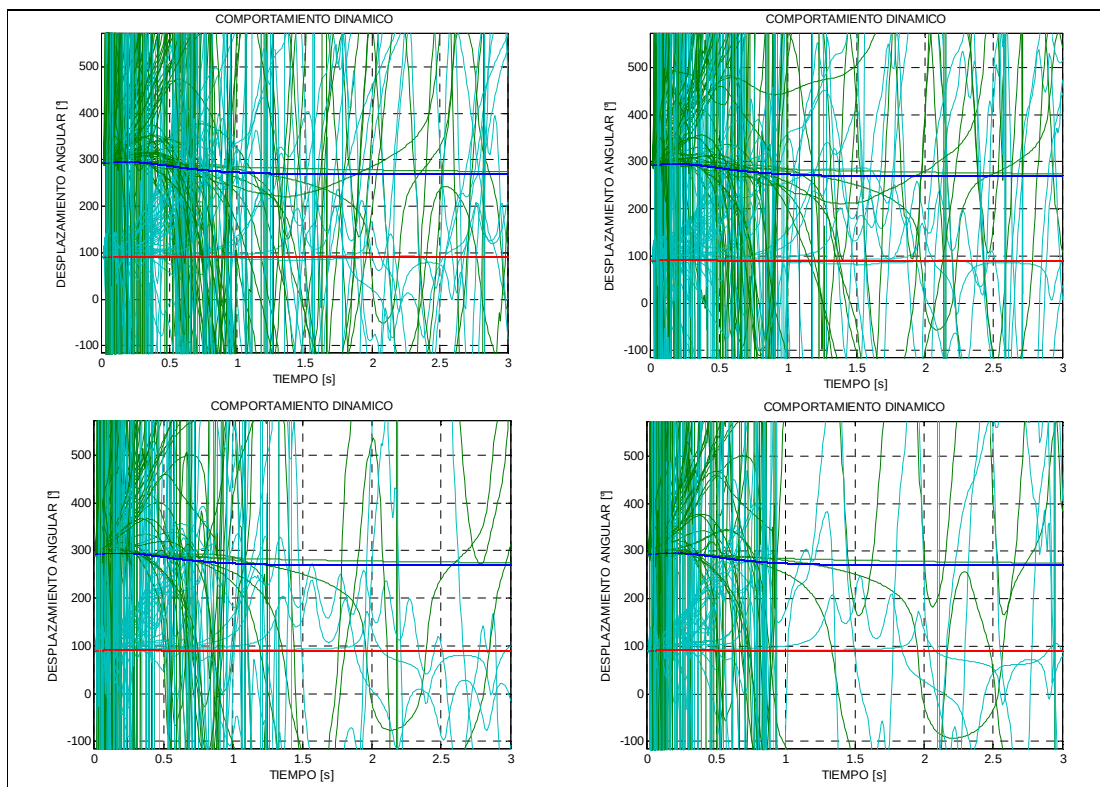
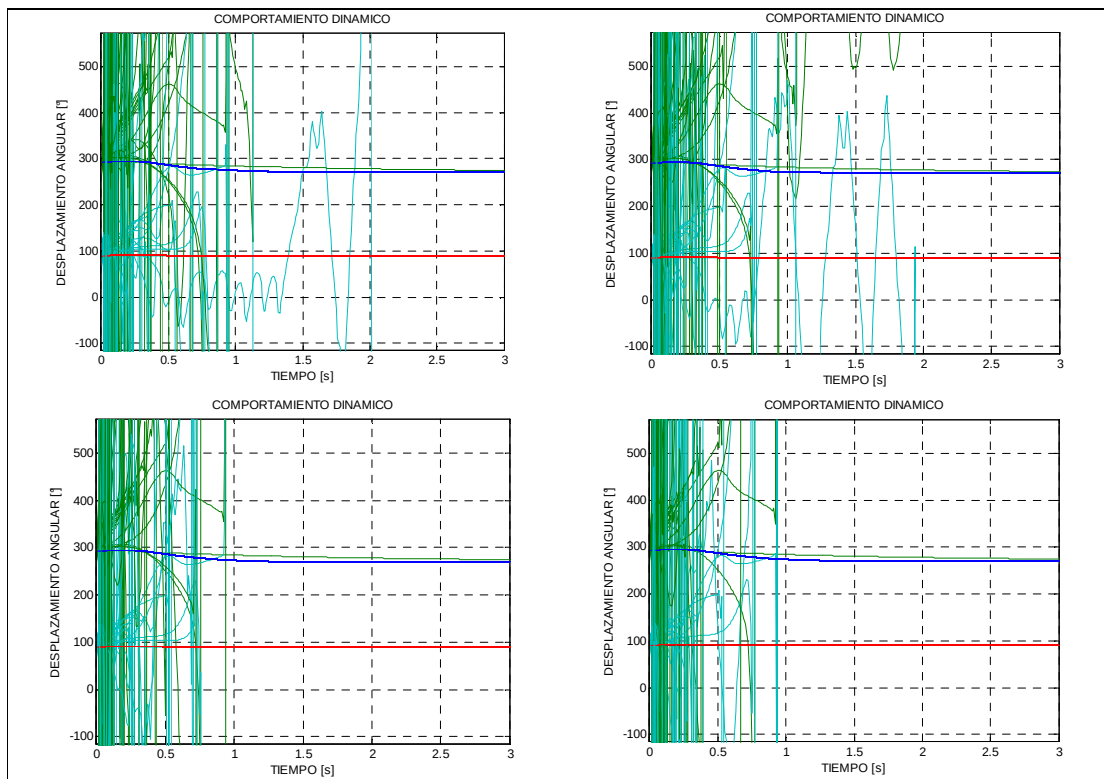
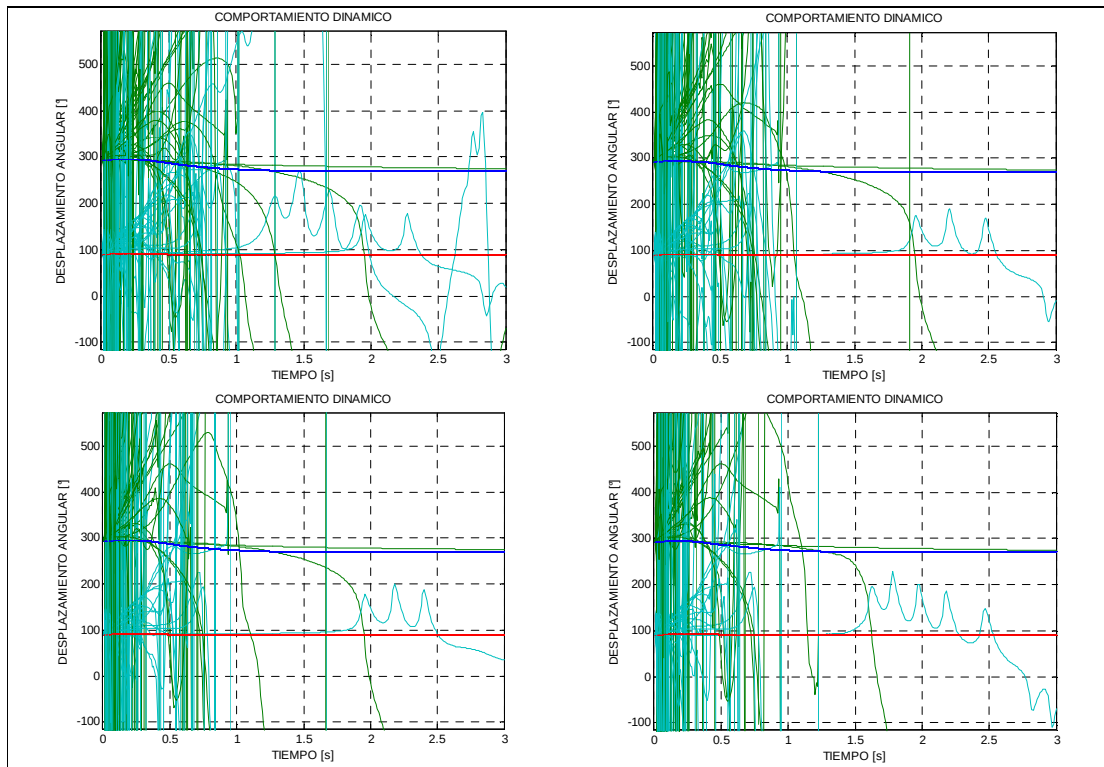


Fig. 5.14. Iteración 5-8 de búsqueda de solución óptima



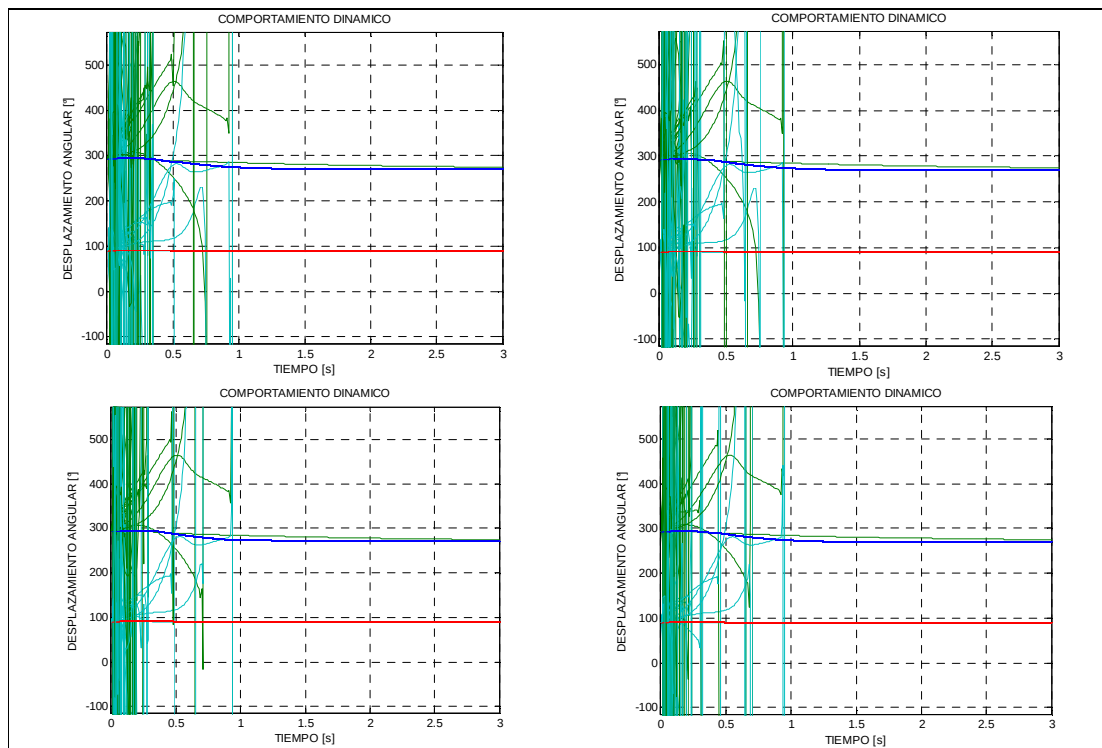


Fig. 5.17. Iteración 17–20 de búsqueda de solución óptima

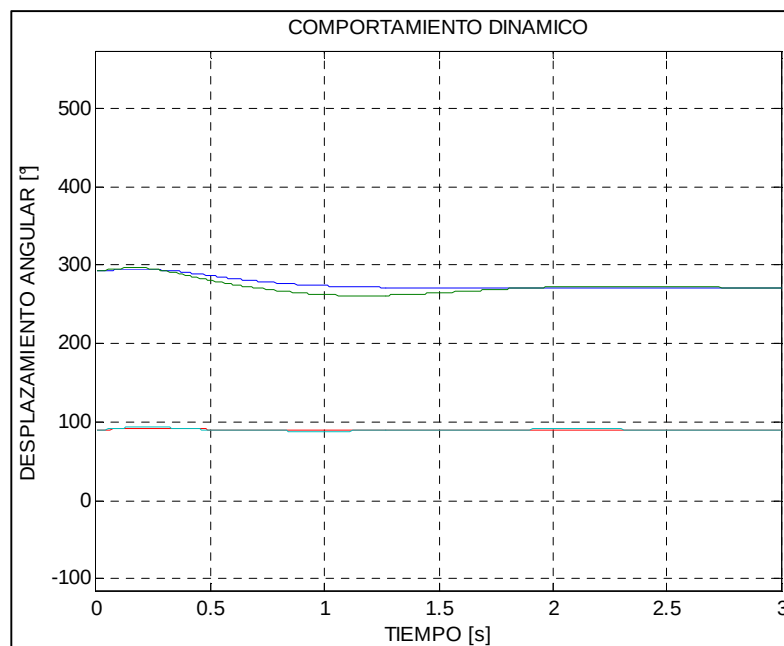


Fig. 5.18. Solución óptima: optimización por enjambre de partículas

Las iteraciones bajo las condiciones anteriormente expuestas son ilustradas en las figuras 5.13 a 5.17. La figura 5.18 ilustra el comportamiento de las ganancias óptimas obtenidas a partir de las iteraciones mostradas en la presente simulación.

Las soluciones óptimas obtenidas después del proceso de iteración de la presente simulación son:

$$k_1 = -0.7671$$

$$k_2 = 2.67780$$

$$k_3 = -0.0870$$

$$k_4 = 0.25850$$

Se puede observar que con base a cada uno de los métodos de sintonización planteados se logró el objetivo de controlar el péndulo doble invertido con un margen de error mínimo, en el capítulo siguiente se analiza la sensibilidad de los algoritmos planteados para los valores establecidos, por el momento se puede observar que los mismos cumplen con los parámetros de sintonización y de búsqueda de ganancias óptimas en el control por retroalimentación de estados del sistema dinámico de estudio.

CAPÍTULO 6

ANÁLISIS DE DATOS

Resumen:

En el presente capítulo se desarrolla el análisis de sensibilidad de las variables presentes en el control del sistema dinámico péndulo doble invertido, se desarrollan varias pruebas en el sistema para determinar el comportamiento de los algoritmos desarrollados con base a la teoría del capítulo 1 y capítulo 4 junto con las pruebas efectuadas en el capítulo 5. Entendiéndose por análisis de sensibilidad a las pruebas efectuadas en cada algoritmo de estudio referentes a algún parámetros de estudio, como es el tiempo de ejecución, la variación referente a las condiciones iniciales, variaciones en las condiciones de las ganancias, etc.

6.1.- INTRODUCCIÓN

Una vez demostrado que los algoritmos basados en computación evolutiva realizan su labor adecuadamente es recomendable observar bajo qué criterios es más eficaz una metodología determinada, por esta razón son efectuadas diversas pruebas para que, en base a las mismas, posteriormente obtener conclusiones sobre cuál es el más adecuado en diversas circunstancias.

Primeramente se analiza el comportamiento de los métodos de sintonización con base a la sensibilidad en las condiciones iniciales, para con ello determinar bajo qué condiciones son similares cada método de sintonización. Posteriormente se presentan los criterios de evaluación de los algoritmos con base a los cuales se obtendrán conclusiones sobre cada una de las metodologías expuestas de métodos no gradientales (algoritmo genético binario, algoritmo de evolución diferencial y optimización por enjambre de partículas) base a:

- Al comportamiento en los límites de las ganancias hasta determinar bajo que límites o fronteras impuestas a las mismas hacen que el sistema permanezca estable.
- Número de individuos y las variaciones referentes a la sensibilidad de los mismos que hacen que el sistema se estabilice y controle.
- La prueba de los algoritmos frente al comportamiento del número de las iteraciones, con ellos se determina con base a las evoluciones si un determinado algoritmo logra obtener el comportamiento deseado.
- El comportamiento del algoritmo frente al sistema de control con respecto al estado transitorio, esto con el fin de determinar cuándo empieza el comportamiento del estado estacionario y saber hasta qué punto es correspondientemente aceptable simular el comportamiento del sistema.
- El comportamiento del algoritmo con respecto al tiempo total de computo del mismo bajo los parámetros establecidos
- Se analiza el parámetro de tasa de mutación en los algoritmos desarrollados y como estos influyen en el comportamiento de los mismos.

Una vez analizados los criterios anteriores se obtienen conclusiones sobre el comportamiento de cada algoritmo estudiado en el presente trabajo y se definen, para el sistema analizado, cual es el mejor algoritmo que cumple con el objetivo planteado de sintonizar el controlador de una manera óptima bajo los criterios de comparación. Posteriormente, se realiza una recomendación sobre la mejor metodología y se presentan posibles trabajos a efectuar sobre los modelos para determinar el mejor algoritmo en otros trabajos de sistemas subactuados mecánicos para corroborar los resultados obtenidos.

6.2.- ANÁLISIS DE SENSIBILIDAD DE LOS ALGORITMOS REFERENTES A LOS LÍMITES DE LAS CONDICIONES INICIALES

En los capítulos anteriores se determinó el comportamiento dinámico del sistema de estudio. En primer lugar, se obtiene el modelo matemático del mismo basado en la formulación Lagrangiana. Una vez planteado el modelo dinámico se procede a obtener la respuesta del mismo sometido primeramente a variaciones en las condiciones iniciales, donde en el capítulo 5 se analizó el comportamiento en lazo abierto del mismo. Posteriormente, con la teoría de control desarrollada en el capítulo 3, se procedió a desarrollar el control en lazo cerrado por retroalimentación de estados por medio de varios métodos, en el presente tema se describen las variaciones obtenidas por cada uno de los métodos de sintonización desarrollados referentes al comportamiento de los límites de las ganancias.

6.2.1.- VARIACIÓN DE LAS CONDICIONES INICIALES DEL CONTROL DEL MODELO LINEALIZADO

La primera prueba para analizar el comportamiento del control se realizó con el sistema linealizado, bajo los teoremas de linealización para sistemas dinámicos no lineales, basado en los métodos clásicos de control y los métodos tradicionales descritos para sintonización del control en el punto de equilibrio deseado, es decir:

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} -\frac{\pi}{2} \\ \pi \\ 0 \\ 0 \end{bmatrix}$$

El modelo dinámico de estudio resultado después de efectuar todo el proceso de linealización es:

$$\begin{aligned} \dot{y}_1 &= y_3 \\ \dot{y}_2 &= y_4 \\ \dot{y}_3 &= \left[\frac{-\theta_4}{\theta_1\theta_2} + \frac{\theta_5\theta_3}{\theta_1\theta_2} \right] [y_1] + \left[\frac{-\theta_5\theta_3}{\theta_1\theta_2} \right] [y_2] + \frac{\tau_1}{\theta_1} \\ \dot{y}_4 &= \frac{\theta_5}{\theta_2} + \left[\frac{\tau_1\theta_3}{\theta_1\theta_2} \right] [y_1] + \left[-\frac{\tau_1\theta_3}{\theta_1\theta_2} \right] [y_2] \end{aligned}$$

Donde se observa el comportamiento de la acción de control en el mismo. Bajo la modelación lineal se puede observar que el control desarrollado es lo bastante robusto como para soportar cualquier desviación del punto de equilibrio, la figura 6.1 ilustra el comportamiento del sistema bajo una desviación del punto de equilibrio de 90°, donde después de una etapa transitoria el control del sistema retoma a la posición original. Por último se observa el comportamiento del sistema

retroalimentado lineal bajo una variación de 180°, en donde, se observa que el control todavía logra estabilizar el sistema en el punto de equilibrio.

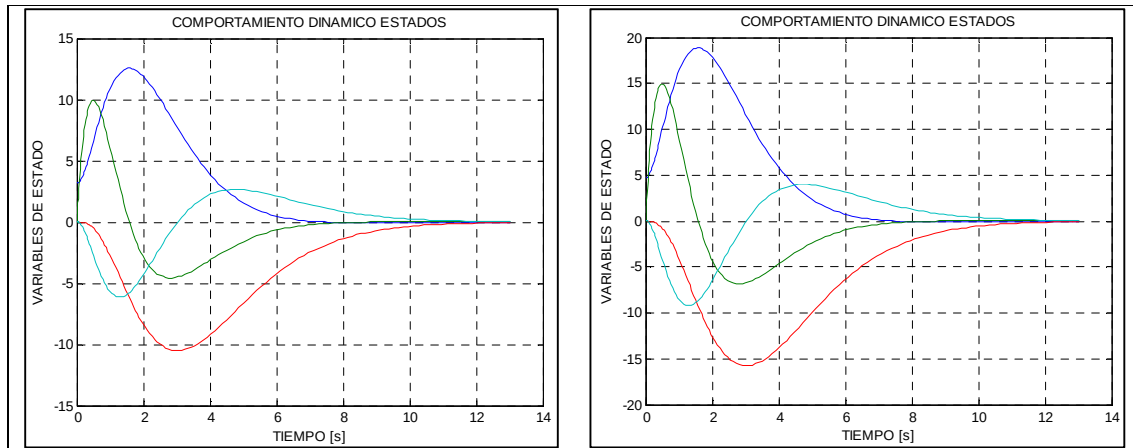


Fig. 6.1. Comportamiento de las variables de estado, variación de 90° (izquierda), variación 180 (derecha)

6.2.2.- VARIACIÓN DE LAS CONDICIONES INICIALES DEL CONTROL DEL MODELO NO LINEAL

La sintonización del sistema dinámico no lineal se desarrolla basados en la teoría lineal alrededor del punto de equilibrio, en teoría el modelo se localiza en el punto de equilibrio, el modelo de aproximación lineal y el modelo no lineal convergerán en comportamientos y en un entorno cercano al mismo el sistema no lineal podrá ser controlado por las ganancias establecidas para el sistema lineal. En este apartado se analiza ese comportamiento teniendo como resultado de acoplamiento las ganancias obtenidas del sistema lineal, el cual bajo una desviación de:

$$\Delta x_1 = 1 \text{ radianes}$$

Desarrollando la acción de control bajo estos sistemas se puede observar que las variables de estado del sistema logran estabilizarse en los puntos de equilibrio establecidos bajo variaciones considerables en las variables de estado, como es el caso de la expuesta anteriormente.

Se define el valor frontera bajo el cual el control deja de ser válido para el sistema no lineal para valores superiores al incremento mostrado en la variable de estado 1 el sistema sale de control. Como se ilustra en la siguiente imagen, el cual toma como valor:

$$\Delta x_1 = 1.053017003$$

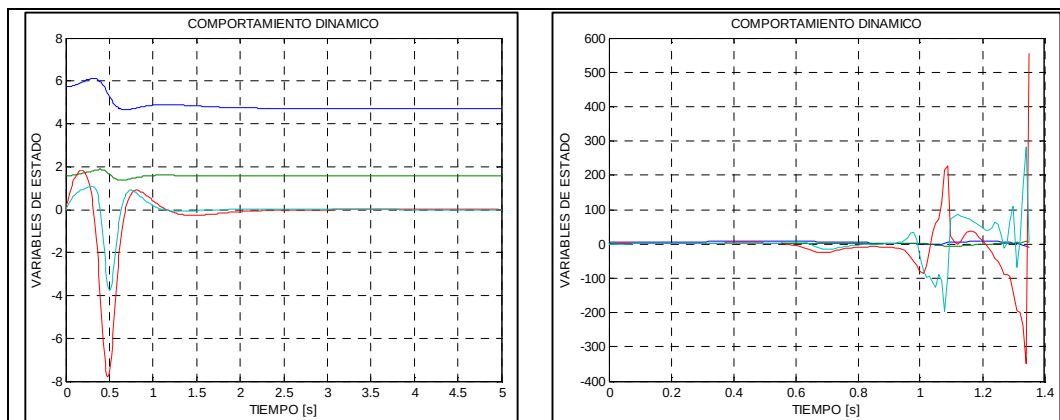


Fig. 6.2. Comportamiento de las variables de estado no lineal, estable (izquierda), inestable (derecha)

Se puede observar en la figura 6.2 que la acción de control para el sistema dinámico no lineal basada en los métodos lineales son válidos para un rango amplio de desviaciones.

6.2.3.- VARIACIÓN DE LAS CONDICIONES INICIALES DEL CONTROL DEL MODELO NO LINEAL BASADO EN ALGORITMOS GENÉTICOS

Con base al modelo no lineal del sistema de péndulo doble invertido se desarrolla el método de sintonización del control no lineal, bajo el cual se puede observar que el sistema dinámico con variaciones considerables en las condiciones iniciales no logra sintonizarse. Basado en este hecho se determina en primer término cual es la variación que permite en la condición inicial referente al estado 1 el sistema de sintonización basado en algoritmos genéticos. Se observa en la figura 6.3 que la desviación máxima permitida es la misma que para el control no lineal basado en el control lineal.

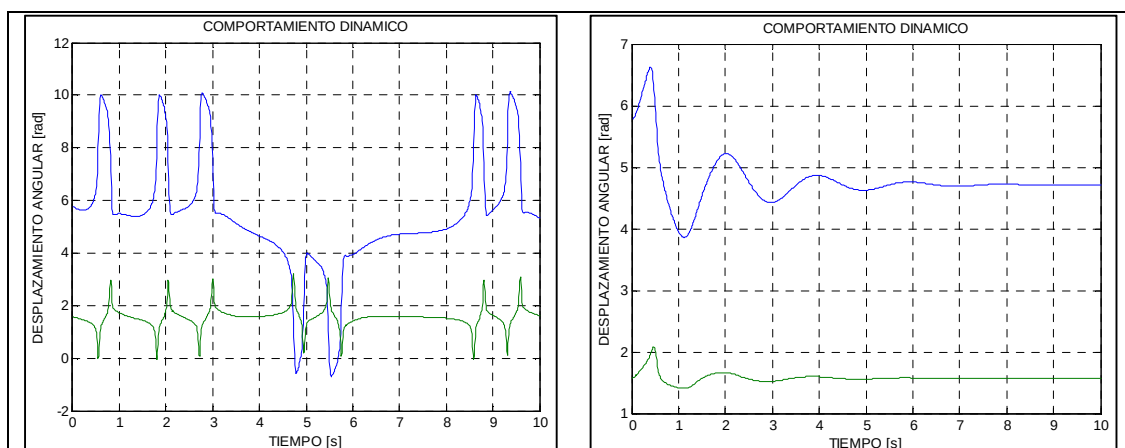


Fig. 6.3. Sintonización de las variables de estado no lineal por método de los algoritmos genéticos valor crítico, estable (derecha), inestable (izquierda)

6.2.4.- VARIACIÓN DE LAS CONDICIONES INICIALES DEL CONTROL DEL MODELO NO LINEAL BASADO EN EVOLUCIÓN DIFERENCIAL

Se analiza a continuación el comportamiento del algoritmo de sintonización bajo el algoritmo de evolución diferencial y todas las condiciones similares al método de sintonización gradiente y de algoritmos genéticos, incluida la variación presente en la condición inicial referente al primer estado. Bajo la cual se puede observar en la figura 6.4 que el sistema tiende a controlarse después de un breve periodo transitorio, se determina que el comportamiento del algoritmo de sintonización por evolución diferencial logra controlar el sistema dinámico más allá de la condición inicial establecida, obteniéndose como valor crítico:

$$\Delta x_1 = 1.05301700374$$

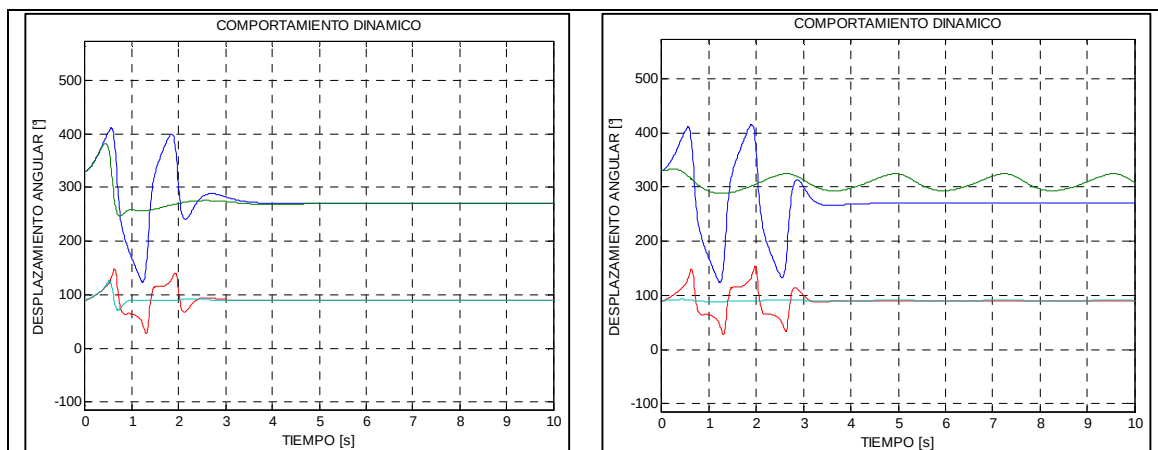


Fig. 6.4. Sintonización de las variables de estado no lineal por método de evolución diferencial valor crítico, estable (izquierda), inestable (derecha)

6.2.5.- VARIACIÓN DE LAS CONDICIONES INICIALES DEL CONTROL DEL MODELO NO LINEAL BASADO EN OPTIMIZACIÓN POR ENJAMBRE DE PARTÍCULAS

El siguiente caso de estudio corresponde a la variación inicial del estado uno, bajo las condiciones impuestas al algoritmo de sintonización de optimización por enjambre de partículas. Se empieza tomando un valor crítico de desviación del punto de referencia igual al definido para el sistema no lineal. Bajo este valor frontera el sistema logra estabilizarse, como se observa en la figura 6.5.

Determinándose como es el caso del algoritmo de evolución diferencial un nuevo valor frontera de:

$$\Delta x_1 = 1.053017003741$$

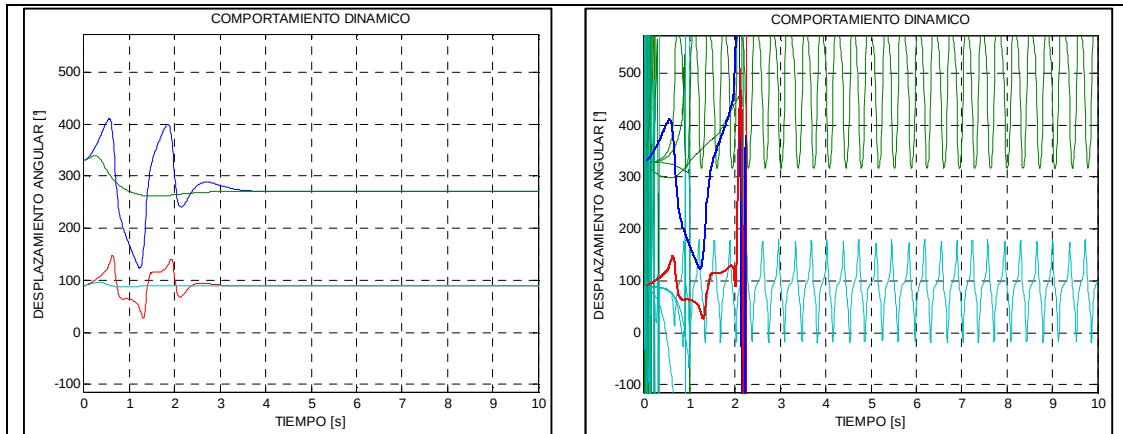


Fig. 6.5. Sintonización de las variables de estado no lineal por método de optimización por enjambre de partículas valor crítico, estable (izquierda), inestable (derecha)

6.3.- SENSIBILIDAD DE ALGORITMOS REFERENTES A LOS LÍMITES DE LAS GANANCIAS

Se estudia el comportamiento de cada algoritmo de sintonización basado en computación evolutiva para el sistema de péndulo doble invertido con respecto a la variación en la frontera para determinar las ganancias de sintonización del control, con lo cual se determina, bajo las condiciones de prueba establecidas en el capítulo 4, que metodología logra ser más eficiente en la sintonización del sistema frente a valores críticos de búsqueda. La variación en cada prueba se efectúa en base a un estrechamiento de los límites, dado que, con ello las posibilidades de búsqueda disminuyen y los algoritmos tienden a forzar su comportamiento para sintonizar el control. Se determina una desviación inicial del estado uno igual a:

$$\Delta x_1 = 1.053$$

Valor inferior al crítico para las pruebas establecidas, esto con el fin de forzar el comportamiento del algoritmo correspondiente de estudio.

6.3.1.- SENSIBILIDAD PRESENTE EN EL ALGORITMO GENÉTICO

Se observa en la figura 6.6 que el algoritmo genético bajo las condiciones impuestas referentes a las condiciones iniciales y los límites de búsqueda de ganancias iniciales de:

$$\begin{aligned} k_1 &= [-1, 1] \\ k_2 &= [0, 3] \\ k_3 &= [-0.5, 1] \\ k_4 &= [0, 0.5] \end{aligned}$$

Tiene un comportamiento de búsqueda estable, logrando sintonizar el mismo.

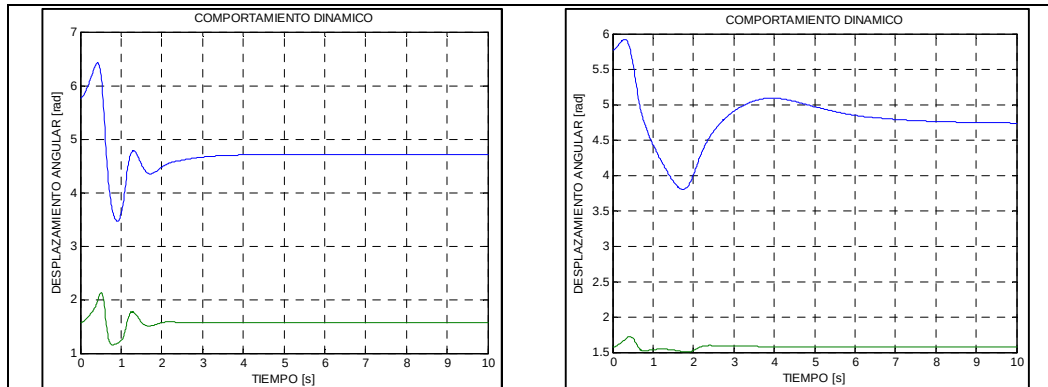


Fig. 6.6. Sintonización AG valores de ganancias estables, ganancias 1 (izquierda), ganancias 2 (derecha)

Se realiza una aproximación a los valores límite frontera bajo los cuales el sistema de control se vuelve inestable estrechando el rango de búsqueda de ganancias bajo los límites de:

$$k_1 = [-0.55, 0.55]$$

$$k_2 = [0.45, 2.55]$$

$$k_3 = [0.1, 0.5]$$

$$k_4 = [0.25, 0.35]$$

Se determina el comportamiento del sistema de control donde se observa que el sistema dinámico en cuestión no logra estabilizarse en el punto de equilibrio del mismo como se ilustra en la figura 6.7.

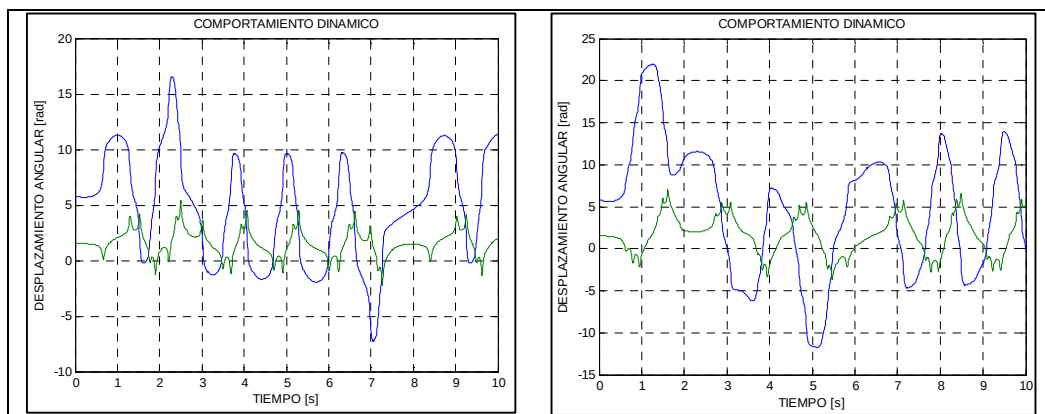


Fig. 6.7. Sintonización AG valores de ganancias inestables, ganancias 1 (izquierda), ganancias 2 (derecha)

Se realiza una nueva búsqueda de límites bajo los cuales el sistema se controle de manera umbral, teniendo como valores de búsqueda:

$$k_1 = [-0.76, 0.76]$$

$$k_2 = [0.24, 2.76]$$

$$k_3 = [-0.27, 0.71]$$

$$k_4 = [0.35, 0.39]$$

Bajo los cuales el sistema toma como comportamiento característico el ilustrado en la figura 6.8. Bajo estos límites el sistema toma un valor umbral de control, son denominados valores frontera de búsqueda del algoritmo genético, valores menores a los mismos muestran un comportamiento en el control inestable. Arriba de los mismos el sistema logra desarrollar valores que controlan perfectamente el sistema dinámico.

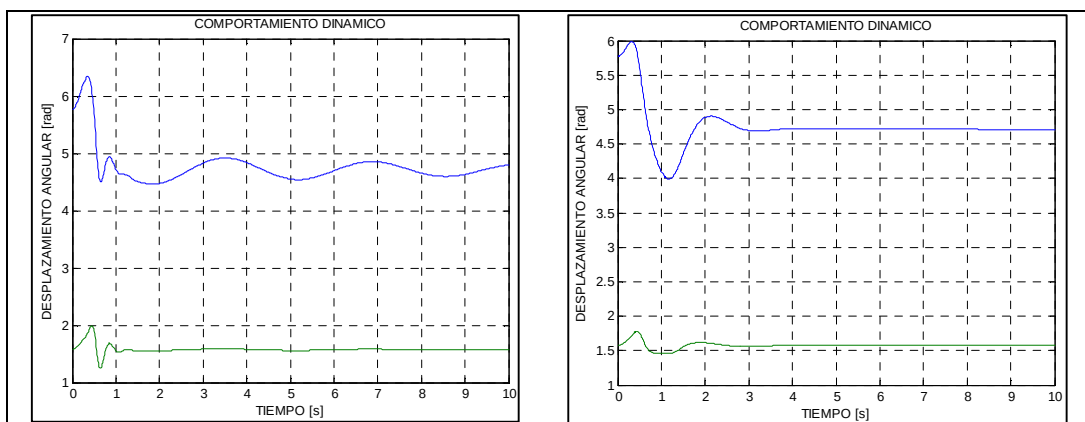


Fig. 6.8. Sintonización AG valores de ganancias estables críticos, ganancias 1 (izquierda), ganancias 2 (derecha)

6.3.2.- SENSIBILIDAD PRESENTE EN EL ALGORITMO DE EVOLUCIÓN DIFERENCIAL

Se realiza ahora la prueba de variación en los límites de búsqueda de ganancias basados en algoritmos de evolución diferencial, bajo los cuales se observa primeramente el comportamiento del sistema bajo los valores de búsqueda de las ganancias definidos, es decir:

$$k_1 = [-1, 1]$$

$$k_2 = [0, 3]$$

$$k_3 = [-0.5, 1]$$

$$k_4 = [0, 0.5]$$

Bajo los cuales se puede observar un comportamiento estable del sistema. Como se muestra en la figura 6.9. Posteriormente se disminuye el intervalo de valores de búsqueda de las ganancias donde el comportamiento del sistema se vuelve inestable como lo ilustra la figura 6.10 con los valores de:

$$k_1 = [-0.5, 0.5]$$

$$k_2 = [0.1, 2.5]$$

$$k_3 = [-0.3, 0.5]$$

$$k_4 = [0, 0.3]$$

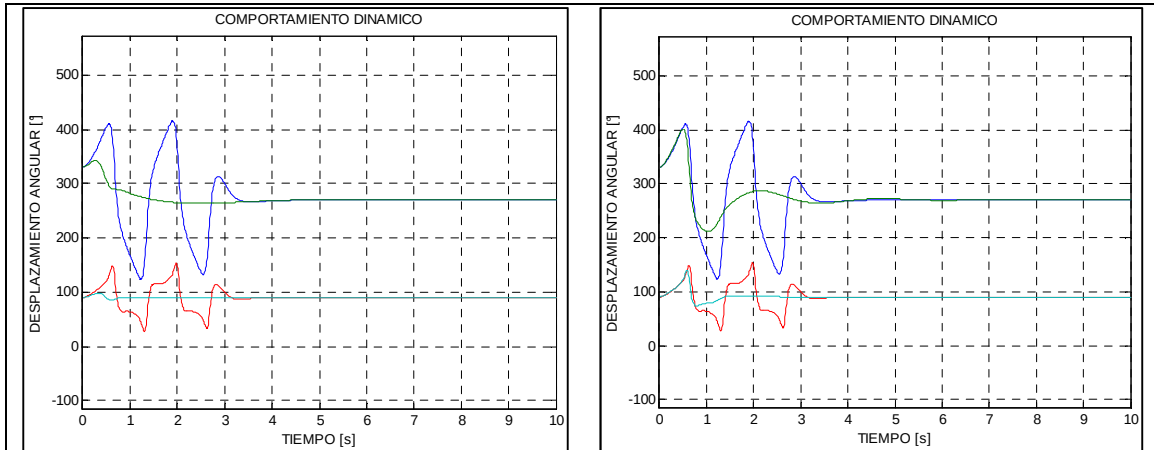


Fig. 6.9. Sintonización ED valores de ganancias estables, ganancias 1 (izquierda), ganancias 2 (derecha)

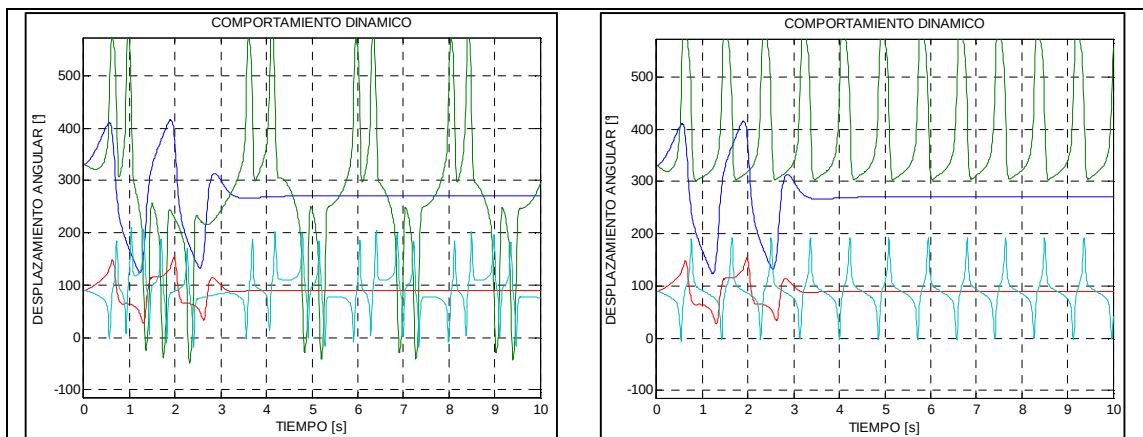


Fig. 6.10. Sintonización ED valores de ganancias inestables, ganancias 1 (izquierda), ganancias 2 (derecha)

Finalmente se presenta el valor obtenido de frontera, bajo el cual el algoritmo de evolución diferencial logra obtener valores inestables y estables en las simulaciones de control del sistema dinámico. Los valores límites definidos para este caso son:

$$k_1 = [-0.55, 0.5]$$

$$k_2 = [0.1, 2.25]$$

$$k_3 = [-0.15, 0.15]$$

$$k_4 = [0, 0.225]$$

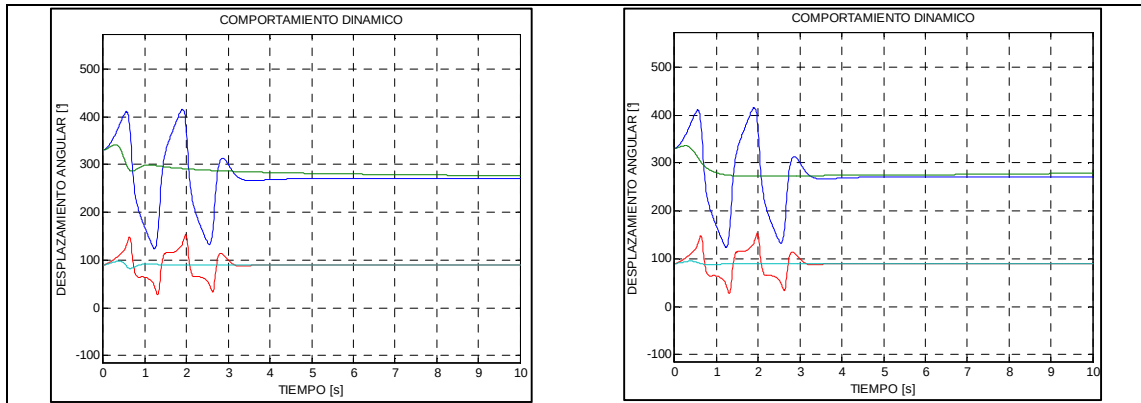


Fig. 6.11. Sintonización ED valores de ganancias estables críticas, ganancias 1 (izquierda), ganancias 2 (derecha)

La figura 6.11 muestra el caso de comportamientos estables que se generaron en el sistema al realizar las búsquedas de ganancias. Es decir el sistema con límites establecidos de búsqueda mayores a los presentados tomara generalmente un comportamiento estable. Para valores menores a los establecidos tomara un mayor esfuerzo computacional y muchas veces no lograra determinar ganancias que logren sintonizar el comportamiento del sistema dinámico de estudio

6.3.3.- SENSIBILIDAD PRESENTE EN EL ALGORITMO DE OPTIMIZACIÓN POR ENJAMBRE DE PARTÍCULAS

Se procede a continuación a analizar el comportamiento del sistema de control del dinámico de estudio sintonizado por medio de optimización por enjambre de partículas (PSO).

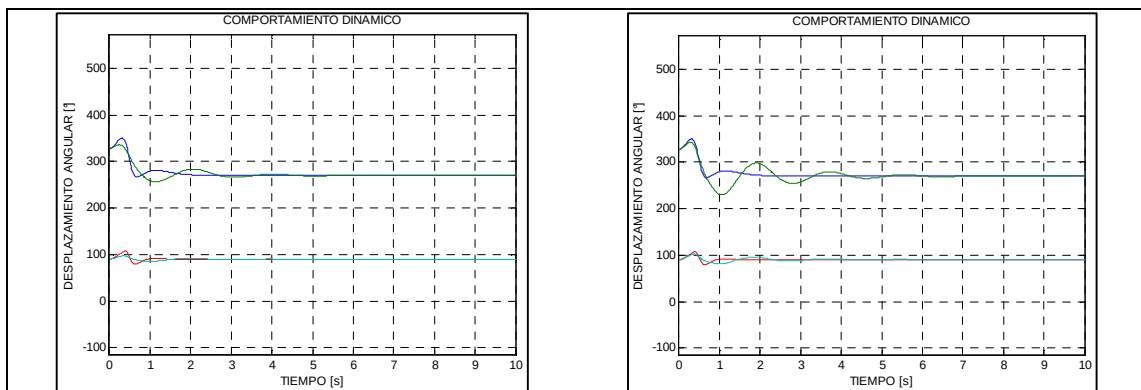


Fig. 6.12. Sintonización PSO valores de ganancias estables, ganancias 1 (izquierda), ganancias 2 (derecha)

Se determina el comportamiento del sistema bajo los mismos parámetros establecidos con anterioridad bajo los límites en los valores de búsqueda de las ganancias de:

$$\begin{aligned}k_1 &= [-1, 1] \\k_2 &= [0, 3] \\k_3 &= [-0.5, 1] \\k_4 &= [0, 0.5]\end{aligned}$$

Donde el comportamiento que toma el sistema dinámico es ilustrado en la figura 6.12, la cual muestra un comportamiento estable del sistema

Posteriormente se realizan variaciones en los límites de búsqueda de las ganancias a los valores definidos de:

$$\begin{aligned}k_1 &= [-0.5, 0.5] \\k_2 &= [0.75, 1.5] \\k_3 &= [-0.1, 0.5] \\k_4 &= [0.1, 0.3]\end{aligned}$$

Bajo estos valores se tiene un comportamiento del algoritmo PSO igual al mostrado en la figura 6.13, donde se puede observar que si bien el sistema es estable, se estabilizo en un nuevo punto de equilibrio o no logro estabilizarse.

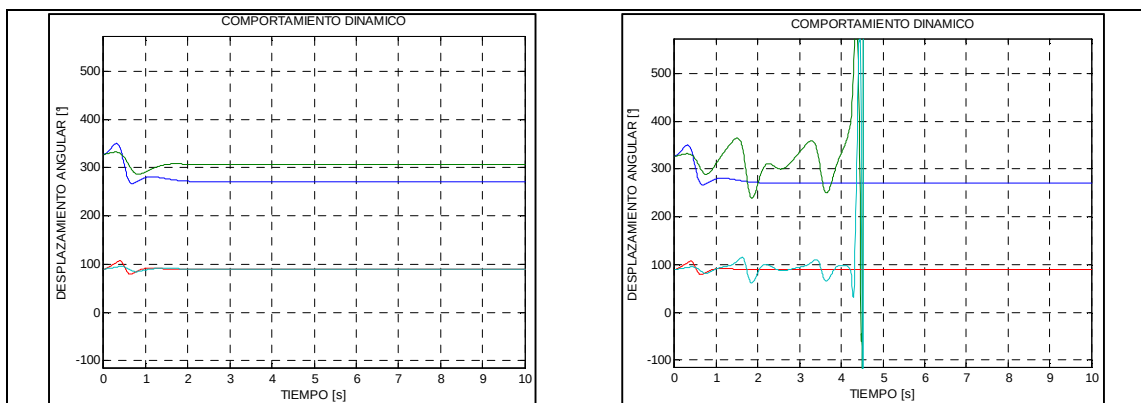


Fig. 6.13. Sintonización PSO valores de ganancias inestables, ganancias 1 (izquierda), ganancias 2 (derecha)

En la figura 6.14 se ilustra el comportamiento del sistema sometido a la búsqueda en el intervalo definido como límite umbral de búsqueda, el cual toma valores de:

$$\begin{aligned}k_1 &= [-0.8, 0.8] \\k_2 &= [0, 2.2] \\k_3 &= [-0.4, 0.7] \\k_4 &= [0, 0.4]\end{aligned}$$

Se puede observar que el comportamiento de este algoritmo es muy sensible a las variaciones en los límites de las ganancias, dado que cualquier perturbación pequeña generaba valores altamente inestables en los estados, no como los

casos anteriores donde la inestabilidad se propagaba a través de movimientos armónicos.

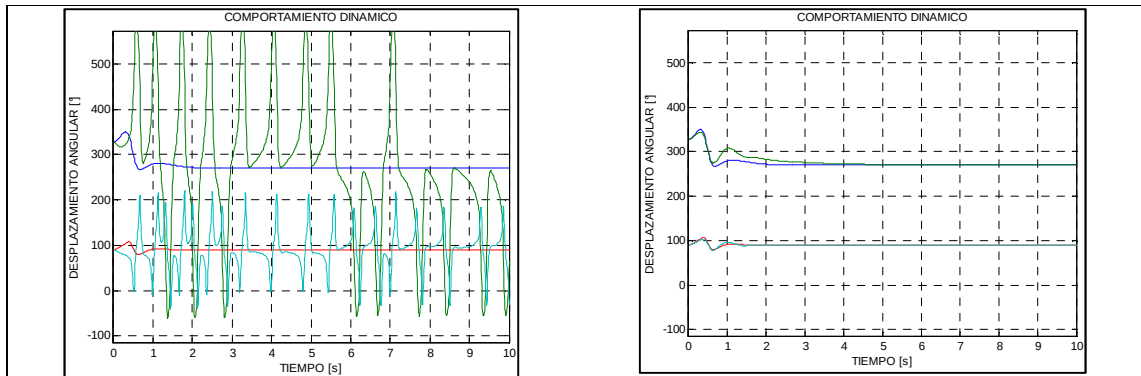


Fig. 6.14. Sintonización PSO valores de ganancias inestables (izquierda), ganancias estables (derecha)

6.4.- VARIACIÓN EN EL TIEMPO DE CÁLCULO DEL ALGORITMO

Se presenta el comportamiento de los algoritmos de evolución artificial desarrollados con base al tiempo de cómputo que consume cada método desarrollado. Esta prueba se define dado que, a causa de la naturaleza de los algoritmos es imposible determinar de manera exacta el tiempo que tarda en desarrollarse cada uno de ellos.

Este tiempo se desarrolla con base a un promedio y al desarrollo de histogramas para determinar el comportamiento en general referente al parámetro tiempo. Las pruebas desarrolladas se efectúan en un total de 100 iteraciones para cada método de sintonización.

La figura 6.15 ilustra el comportamiento del algoritmo genético en general, se puede observar que el mismo tiene una distribución uniforme variando en general en un intervalo de tiempo de [215.11, 259.93]. El valor que más veces tiende a presentarse oscila alrededor de 230 segundos.

La figura número 6.16 ilustra la distribución de tiempo del algoritmo de evolución diferencial, donde se observa que el sistema tiene una variación en un intervalo de [90.68, 124.72] con una tendencia a tomar valores cercanos a los 114 segundos en el desarrollo del algoritmo.

Por último se presenta el comportamiento que toma en general el algoritmo de optimización por enjambre de partículas, donde se observa, con base a las cien iteraciones desarrolladas que estas tienen una distribución en un intervalo de [86.12, 103.55], en el cual se observa una tendencia a querer tomar valores

cercanos a los 95 segundos en tiempo de computo. La figura 6.17 ilustra este comportamiento.

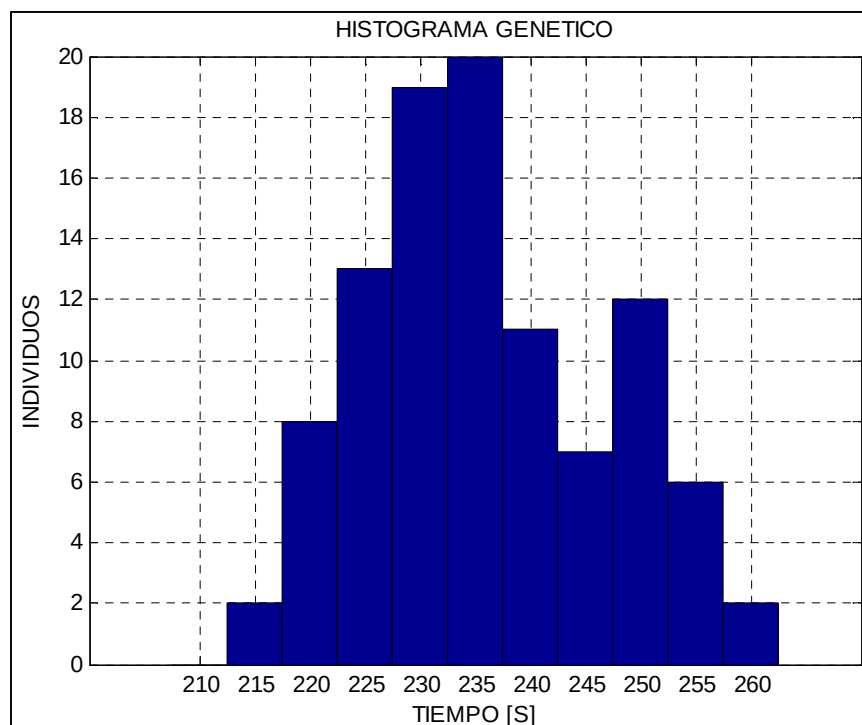


Fig. 6.15. Variación de tiempo de AG

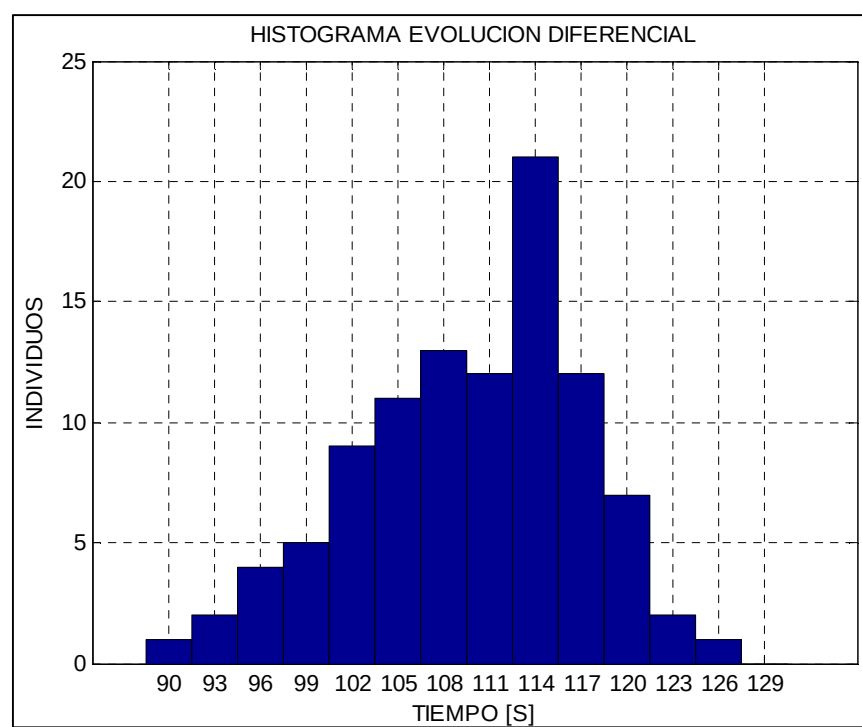


Fig. 6.16. Variación de tiempo de ED.

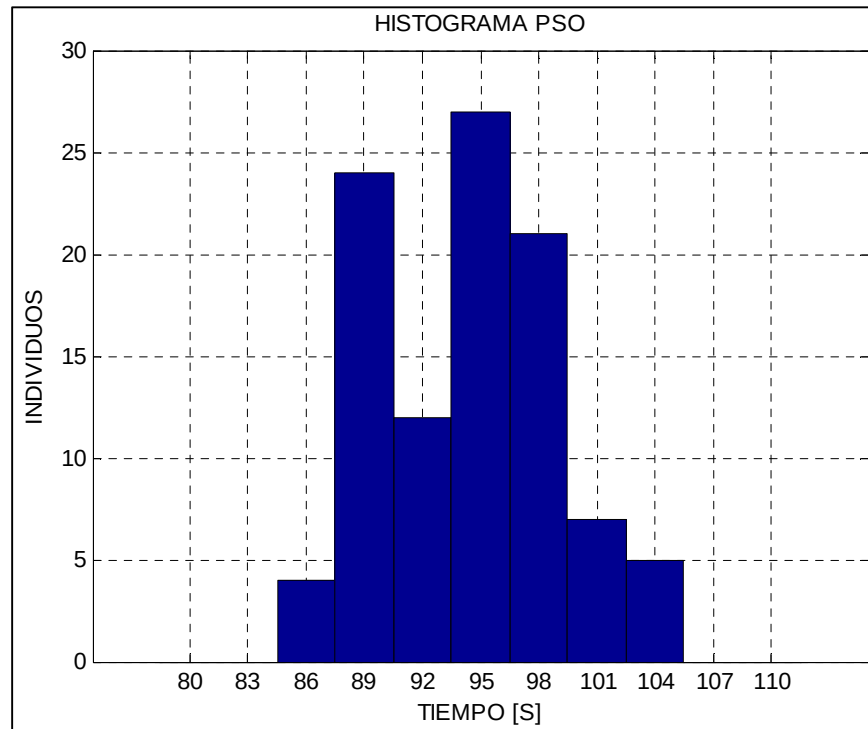


Fig. 6.17. Variación de tiempo de PSO.

6.5.- COMPORTAMIENTO DE LOS ALGORITMOS FRENTE A LA DISMINUCIÓN DE INDIVIDUOS

El parámetro siguiente a medir en cada uno de los algoritmos de prueba de sintonización de las ganancias es la disminución en el número de individuos, esto con el fin de determinar la sensibilidad presente en la sintonización del sistema dinámico cuando los individuos van disminuyendo y hasta que momento cada uno de los métodos desarrollados logra tener precisión en la sintonización del control.

Las pruebas se realizan bajo los mismos parámetros para cada algoritmo de estudio, la única variación presente es el número de individuos.

6.5.1.- COMPORTAMIENTO DEL ALGORITMO GENÉTICO

El primer algoritmo que se analiza referente a las variaciones en el número de individuos es el algoritmo genético, se inicializa el estudio de comportamiento con un número de individuos de 300, posteriormente se realiza una disminución en proporción de 20 individuos para analizar el comportamiento del algoritmo genético con respecto a la disminución de individuos.

Las figuras número 6.18 a 6.20 ilustran el comportamiento del algoritmo, donde se muestran algunos ejemplos del comportamiento de la acción de control del sistema dinámico en cuestión.

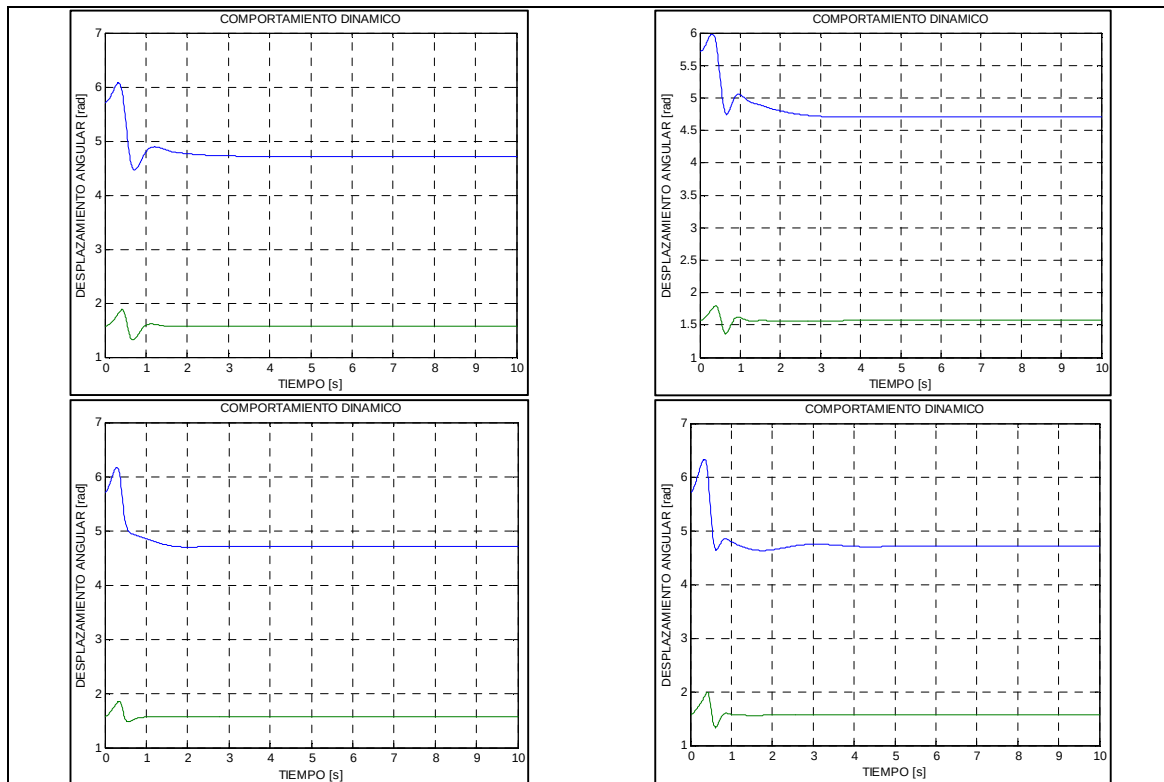


Fig. 6.18. Comportamiento AG 300 individuos

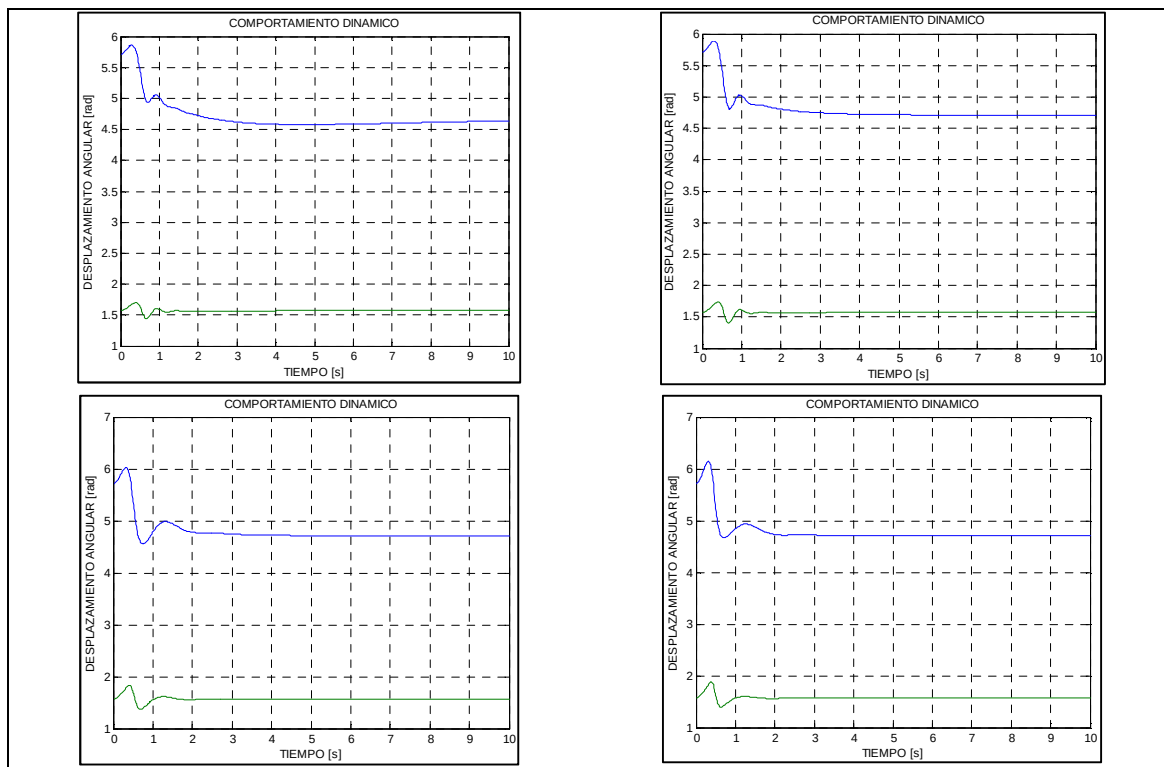


Fig. 6.19. Comportamiento AG 260 individuos

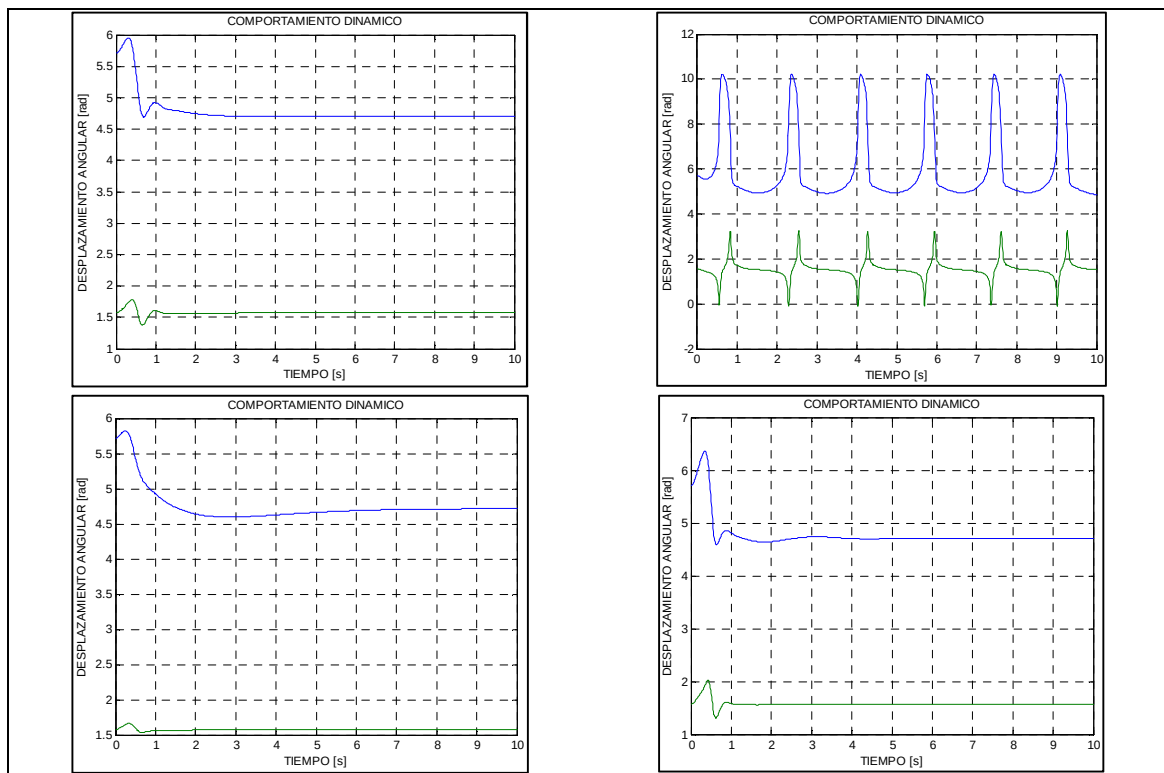


Fig. 6.20. Comportamiento AG 240 individuos

6.5.2.- COMPORTAMIENTO DEL ALGORITMO DE EVOLUCIÓN DIFERENCIAL

Se analiza ahora el comportamiento del sistema dinámico sometido a la acción de sintonización del control por medio del método de evolución diferencial [ED], y el comportamiento que toma el algoritmo con respecto a la disminución de individuos en su entorno de evolución, se mantienen todos los parámetros constantes, para cada prueba, con el fin de medir la sensibilidad solo referente a la disminución de individuos.

Se inicializa el análisis con una población de 300 individuos y posteriormente se va disminuyendo a una razón de 20 individuos como en el caso del algoritmo genético.

Las figuras número 6.21 a 6.29 ilustran el comportamiento del sistema donde todos los parámetros permanecen constantes y existe variación en el número de individuos. Las imágenes describen en general el comportamiento que toma el sistema dinámico de estudio, las cuales ilustran si el algoritmo en cuestión bajo ese número de individuos logra alcanzar la acción de control o no.

Posteriormente se analiza si en general bajo un determinado número de individuos se obtiene un mejor comportamiento de control del sistema o la sensibilidad presente del método de sintonización en cuestión.

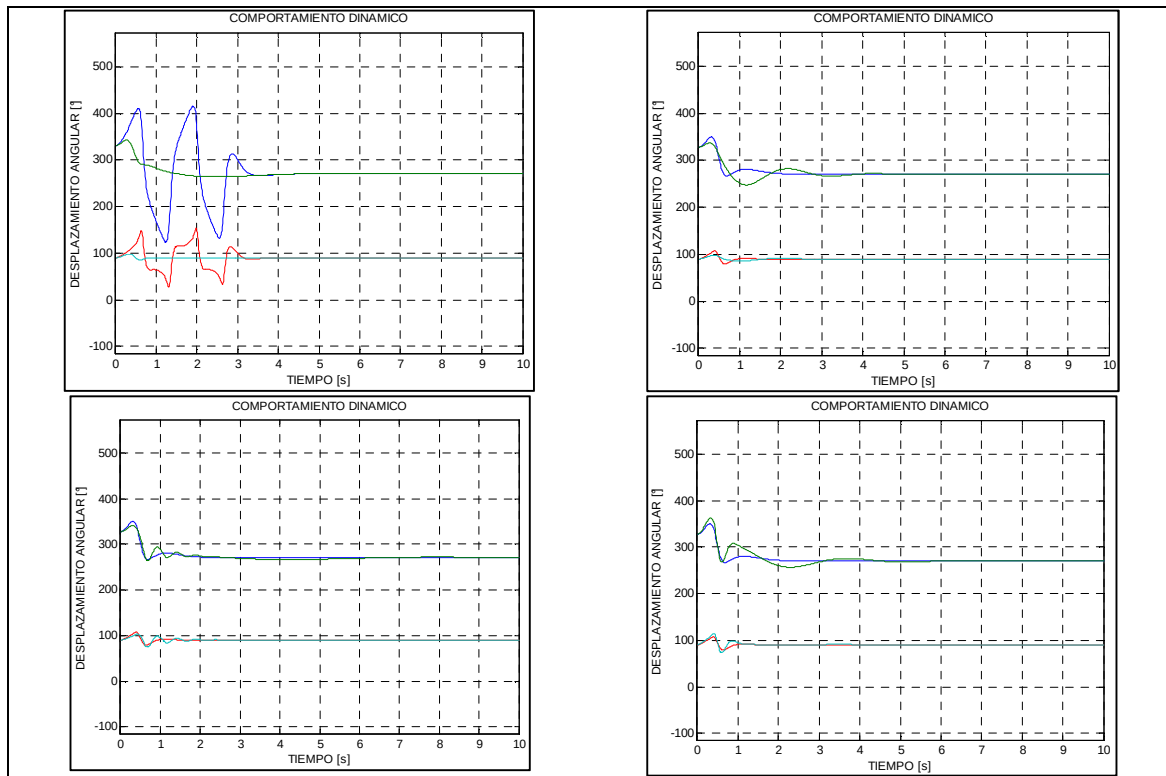


Fig. 6.21. Comportamiento ED 300 individuos

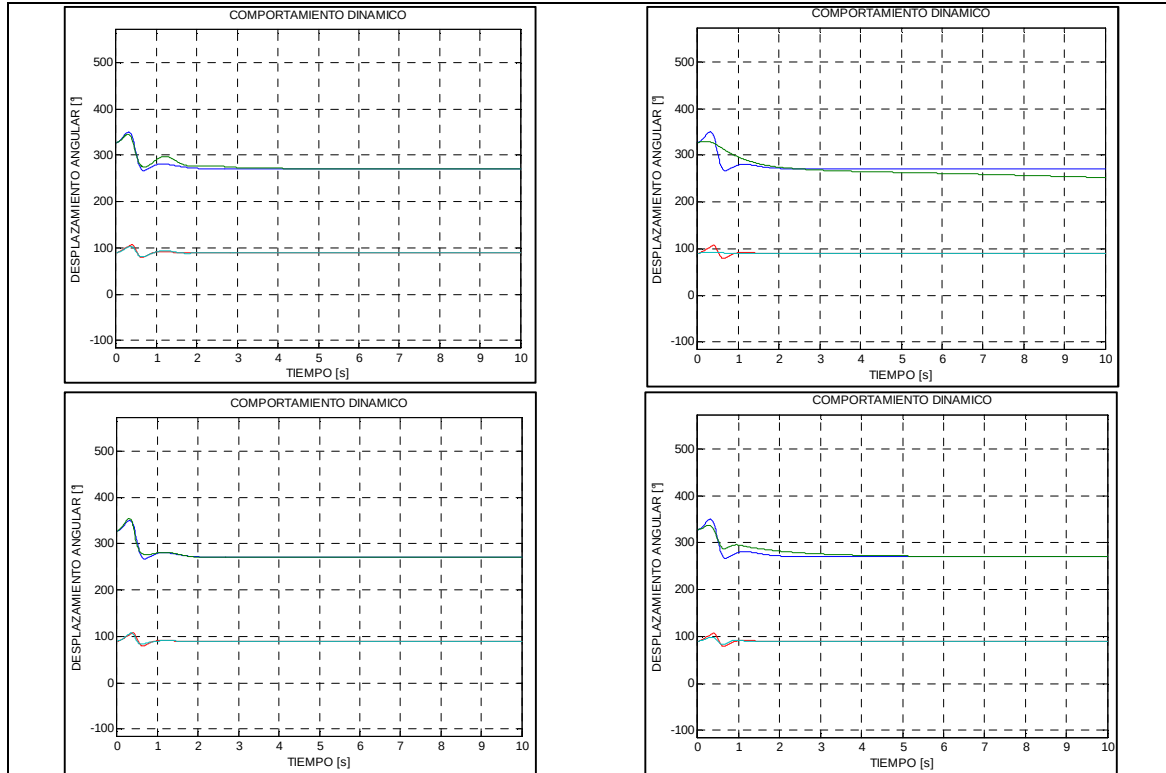


Fig. 6.22. Comportamiento ED 280 individuos

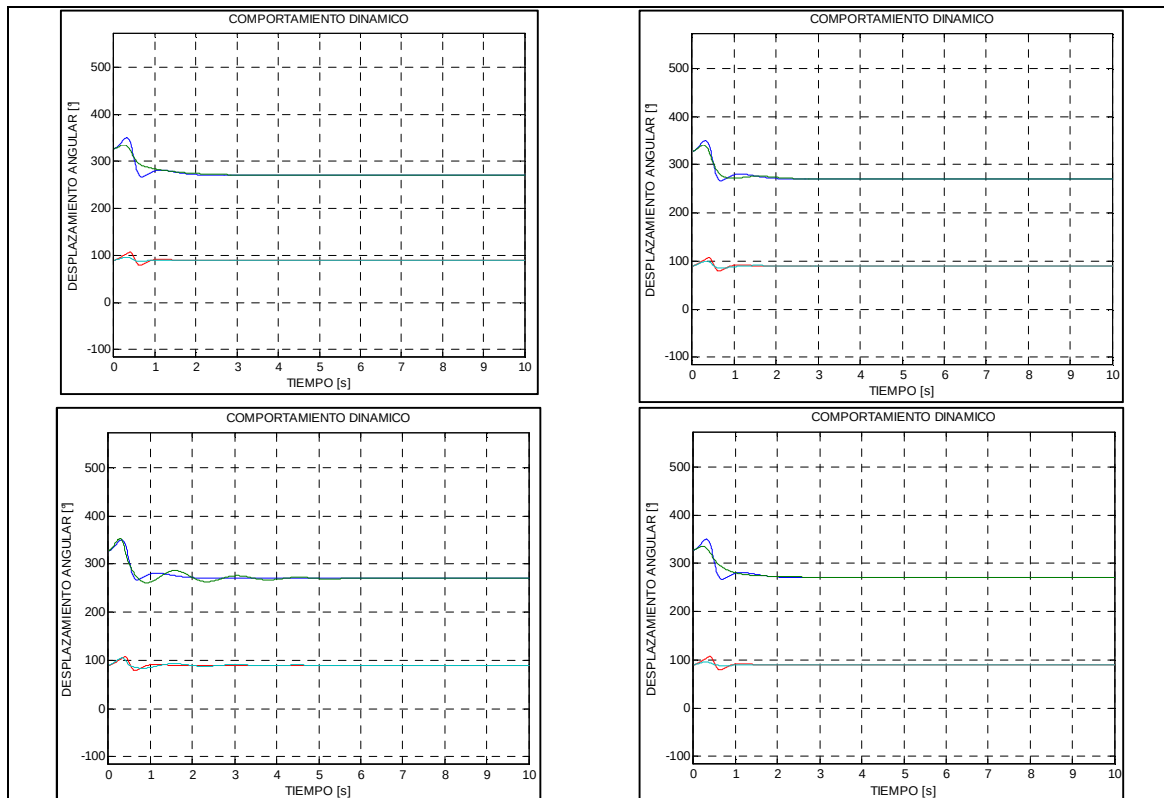


Fig. 6.23. Comportamiento ED 260 individuos

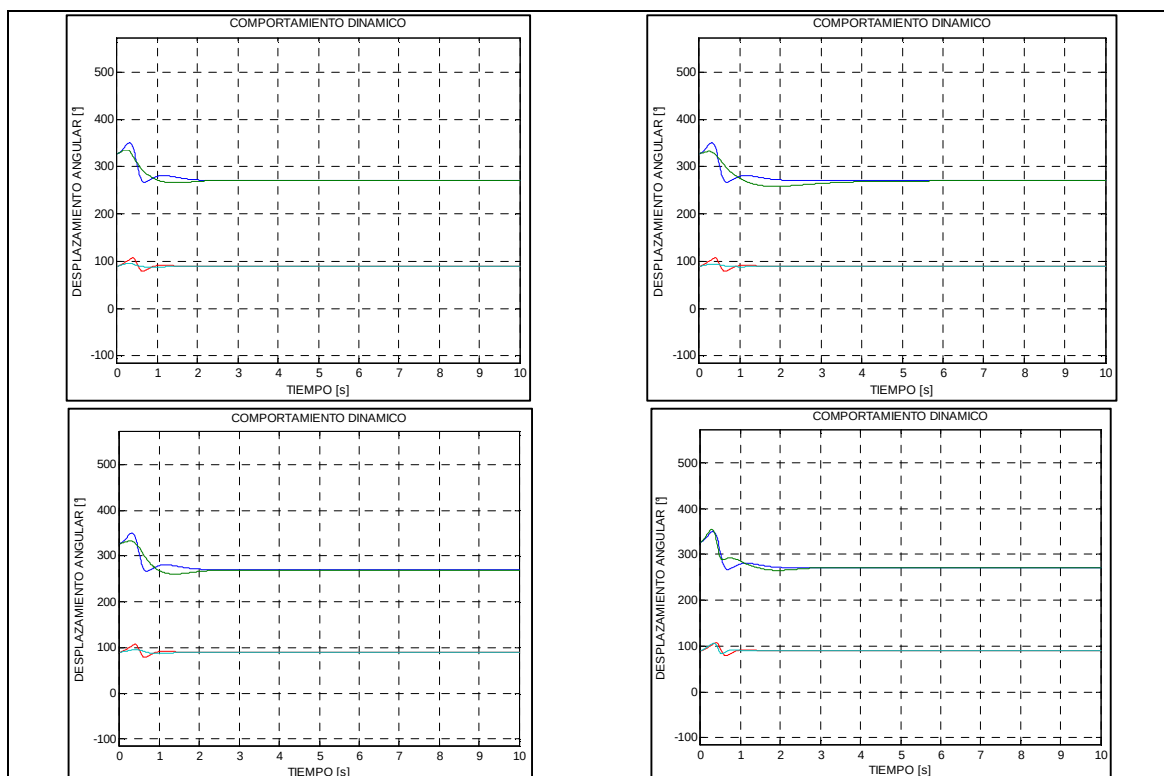


Fig. 6.24. Comportamiento ED 240 individuos

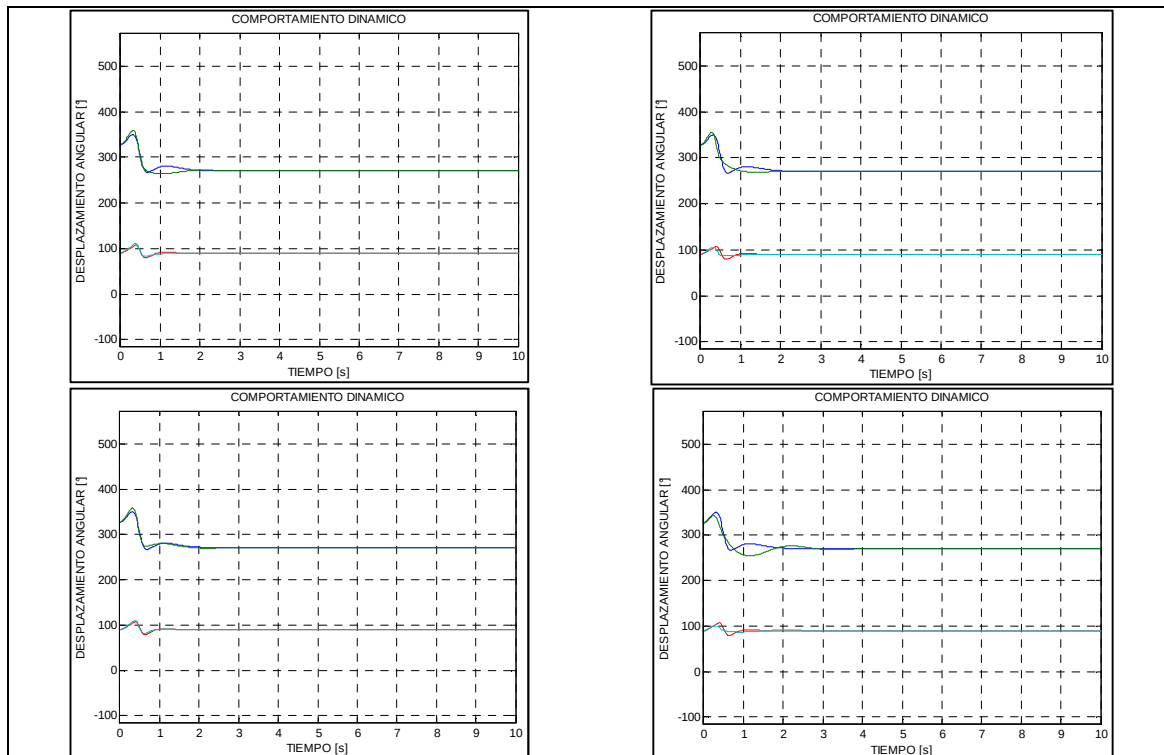


Fig. 6.25. Comportamiento ED 220 individuos

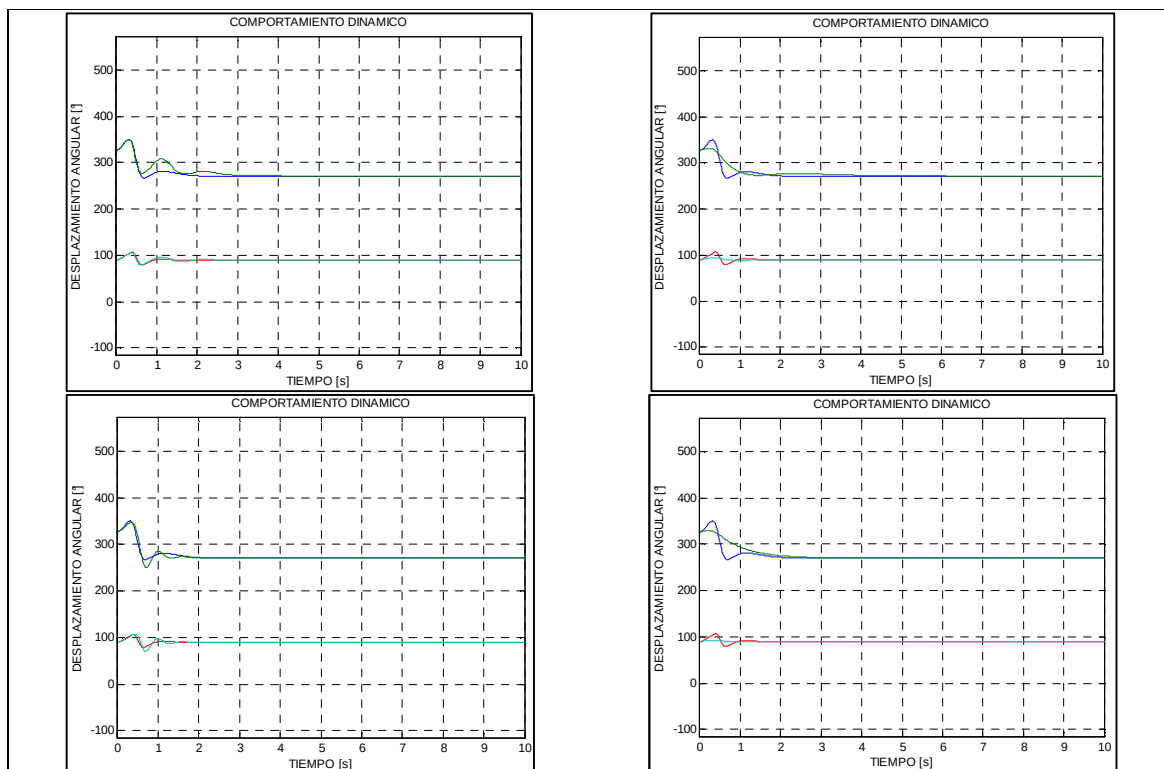


Fig. 6.26. Comportamiento ED 200 individuos

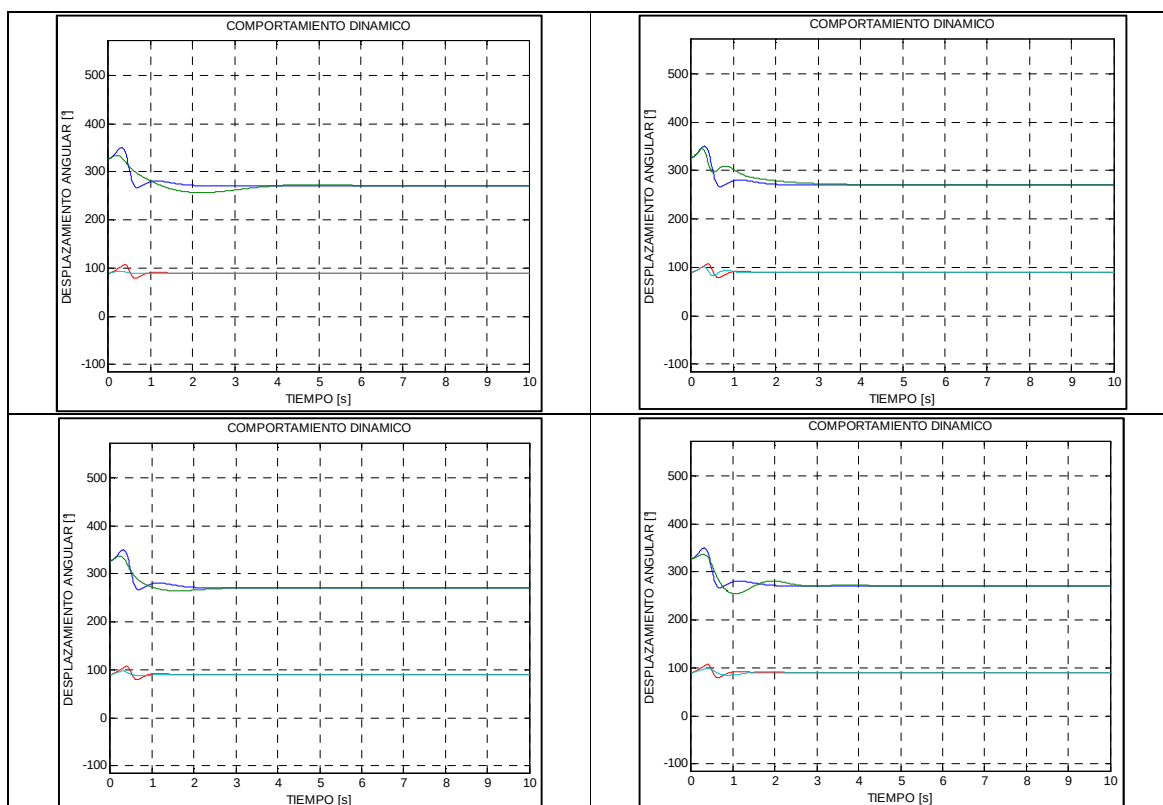


Fig. 6.27. Comportamiento ED 180 individuos

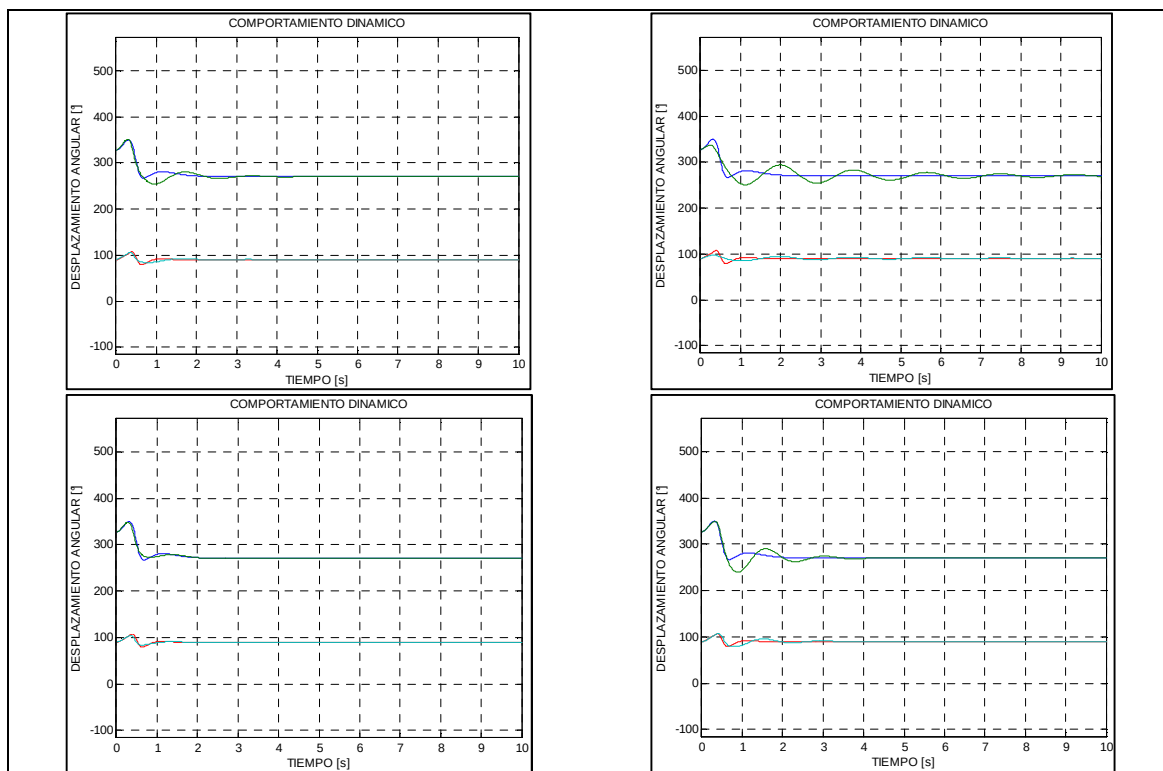


Fig. 6.28. Comportamiento ED 160 individuos

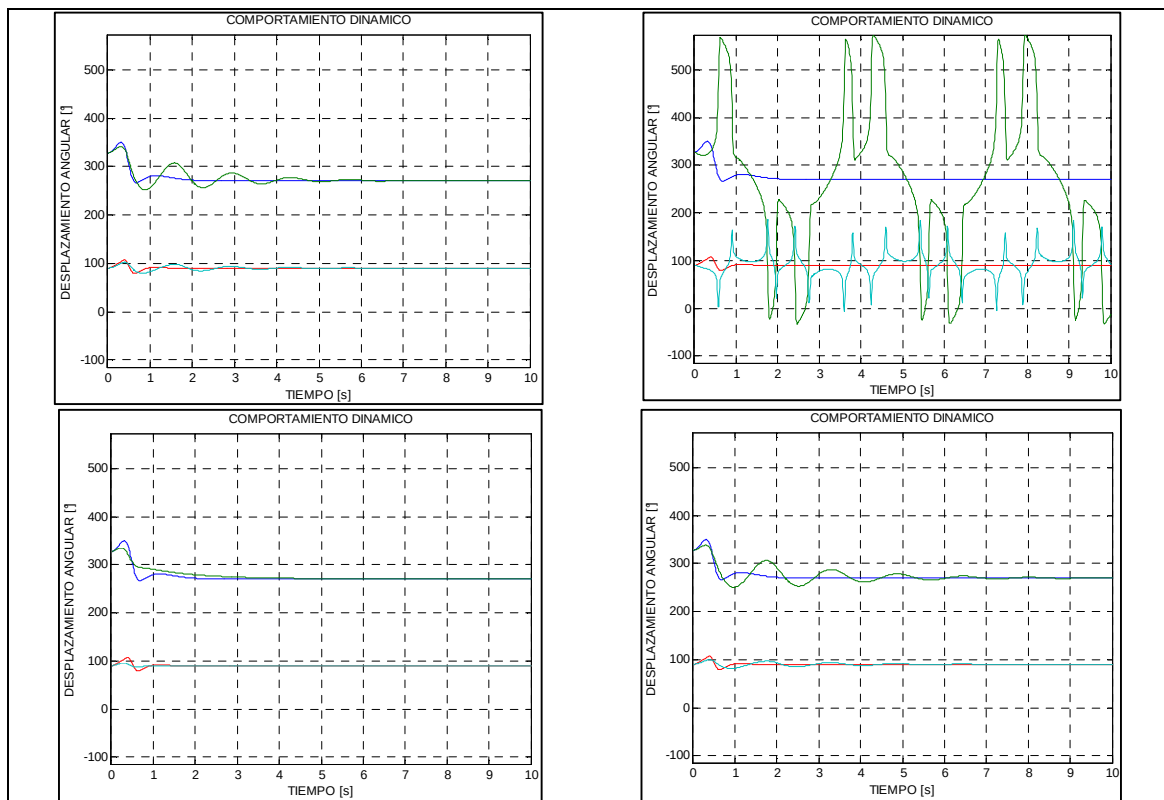


Fig. 6.29. Comportamiento ED 140 individuos

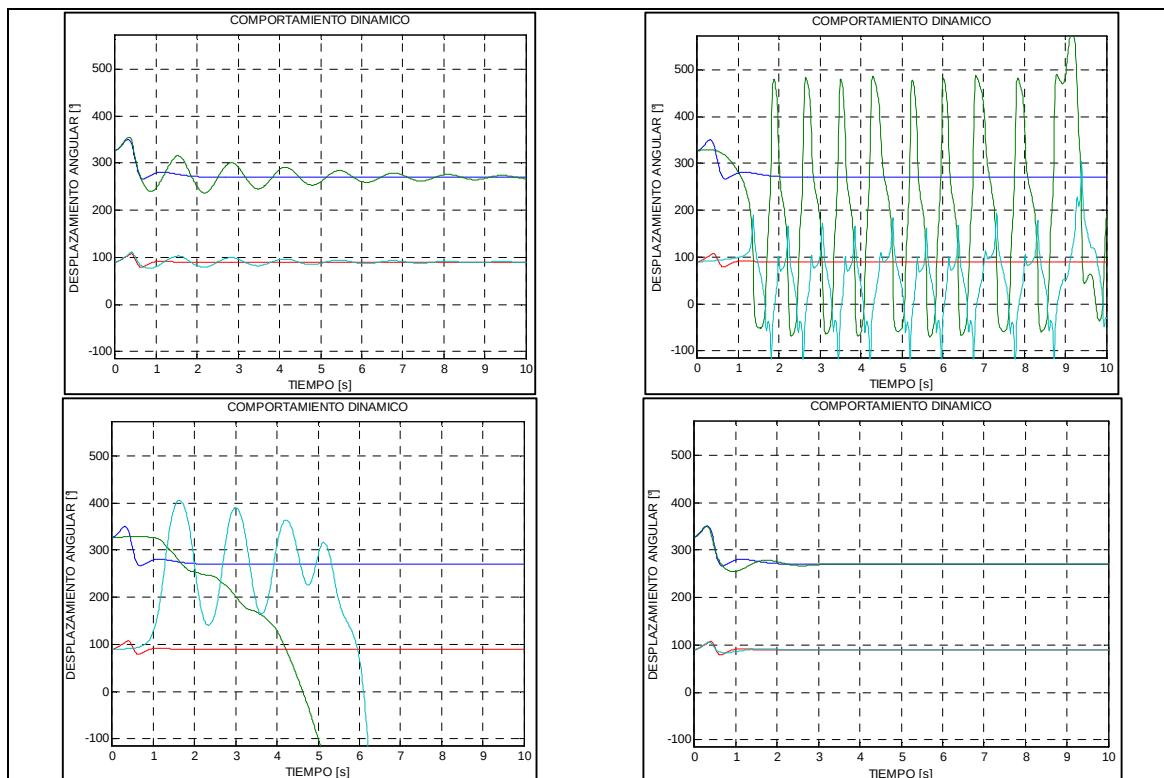
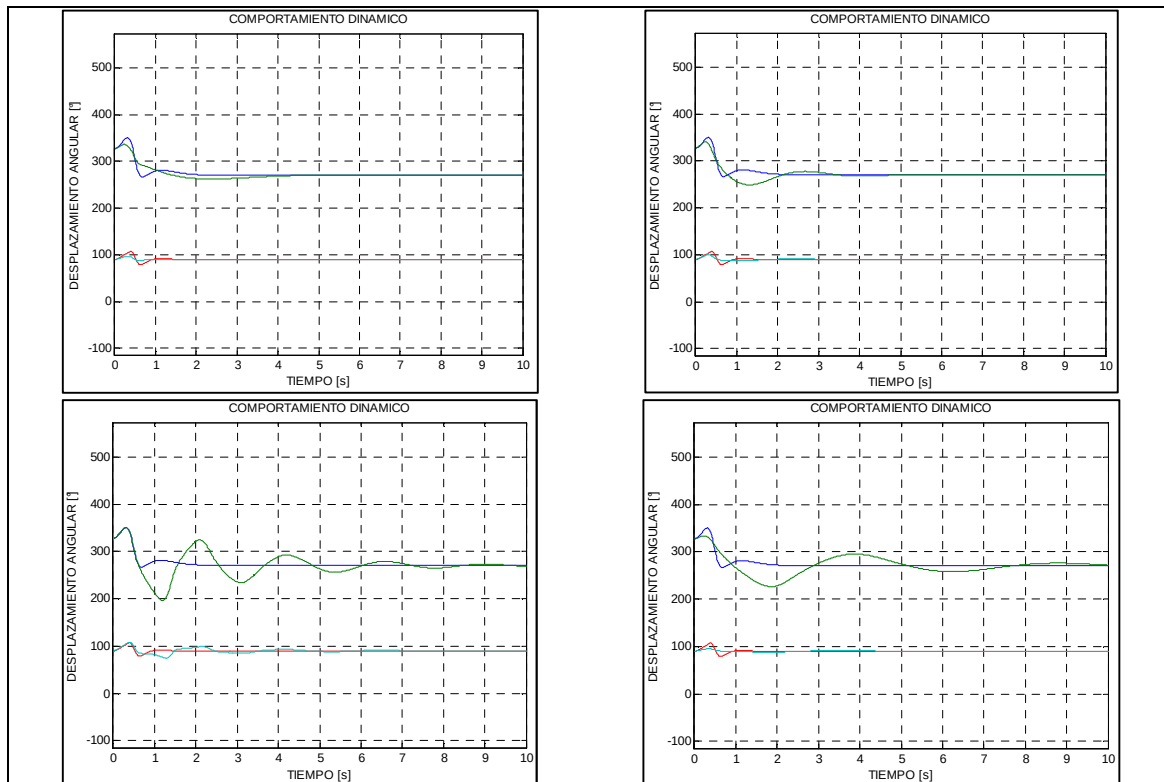
6.5.3.- COMPORTAMIENTO DEL ALGORITMO OPTIMIZACIÓN POR ENJAMBRE DE PARTÍCULAS

Se define a continuación el comportamiento del sistema de control mediante el algoritmo de optimización por enjambre de partículas bajo la variación en el comportamiento referente a la disminución de sus individuos bajo las condiciones constantes de los algoritmos de evolución diferencial y genéticos estudiados anteriormente con el fin de homogeneizar el proceso de análisis.

Se definen variaciones en la cantidad de individuos a una tasa de disminución de los mismos de 20, del mismo modo se inicializa el proceso de análisis en una cantidad de 300 individuos.

Se observa que este método es más sensible a las variaciones en los cambios de poblaciones o disminución de iteraciones dada la misma naturaleza del algoritmo, dado que, este tiene que interactuar con los demás individuos para calcular la mejor población.

Las imágenes 6.30 a 6.31 ilustran el comportamiento del sistema sometido al análisis donde se observa cual es el comportamiento de algunos elementos muestreados para determinar si el sistema de sintonización logra realizar adecuadamente su trabajo con los individuos indicados.



6.6.- SENSIBILIDAD DE LOS ALGORITMOS FRENTE A LA VARIACIÓN EN EL NÚMERO DE EVOLUCIONES

El siguiente parámetro a analizar es el comportamiento de los algoritmos evolutivos frente a una disminución en el número de iteraciones, este parámetro es importante porque es una medida de la cantidad del tiempo que un algoritmo tiende a aprender a localizar los puntos óptimos, si se disminuye la cantidad de evoluciones se está permitiendo que el algoritmo alcance valores máximos en un tiempo relativamente corto. Esta prueba también es un indicativo de la robustez del algoritmo de estudio bajo el comportamiento del sistema dinámico péndulo doble invertido.

6.6.1.- VARIACIÓN EN EL ALGORITMO GENÉTICO

El primer algoritmo de estudio es el algoritmo genético, bajo el cual se inicializa una cantidad de 20 evoluciones como límite máximo para lograr encontrar las ganancias que sintonicen el sistema dinámico en cuestión, la disminución de los individuos se realiza en razón de 2 evoluciones. Todos los parámetros involucrados en el sistema permanecen constantes, el parámetro variable será la cantidad de evoluciones.

Las figuras número 6.32 a 6.37 ilustran el comportamiento de las pruebas realizadas a este algoritmo.

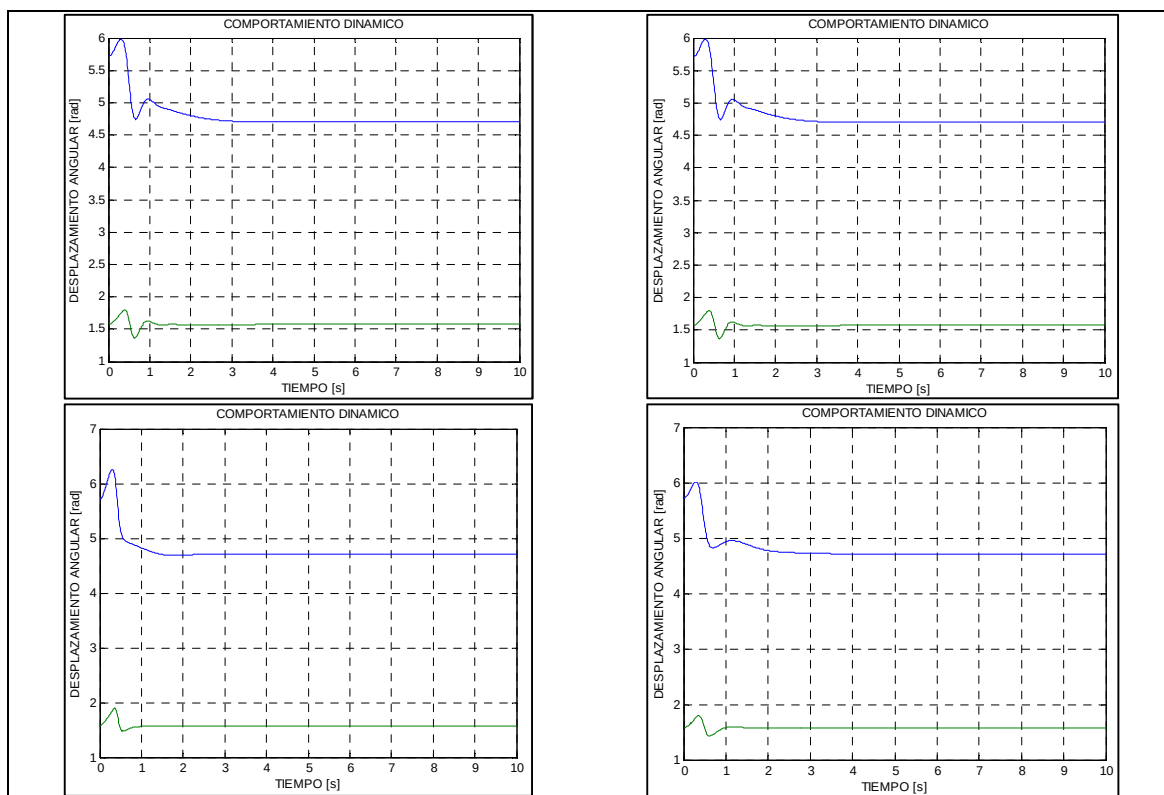
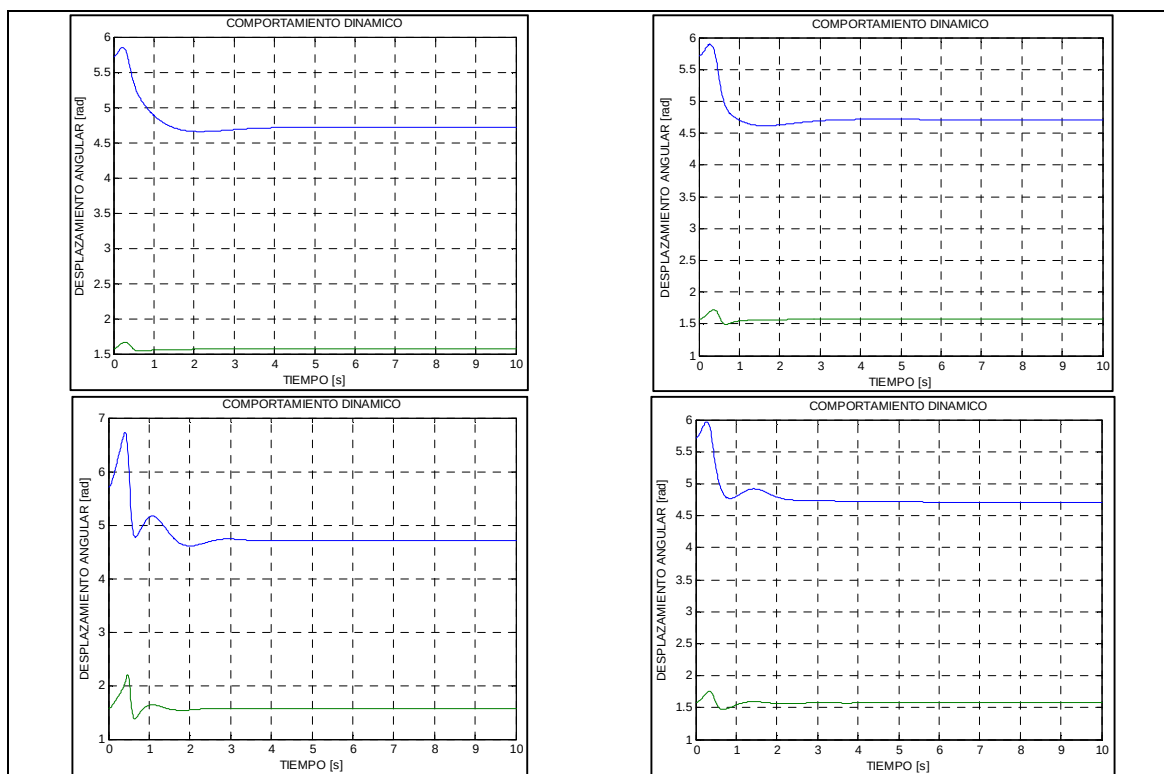
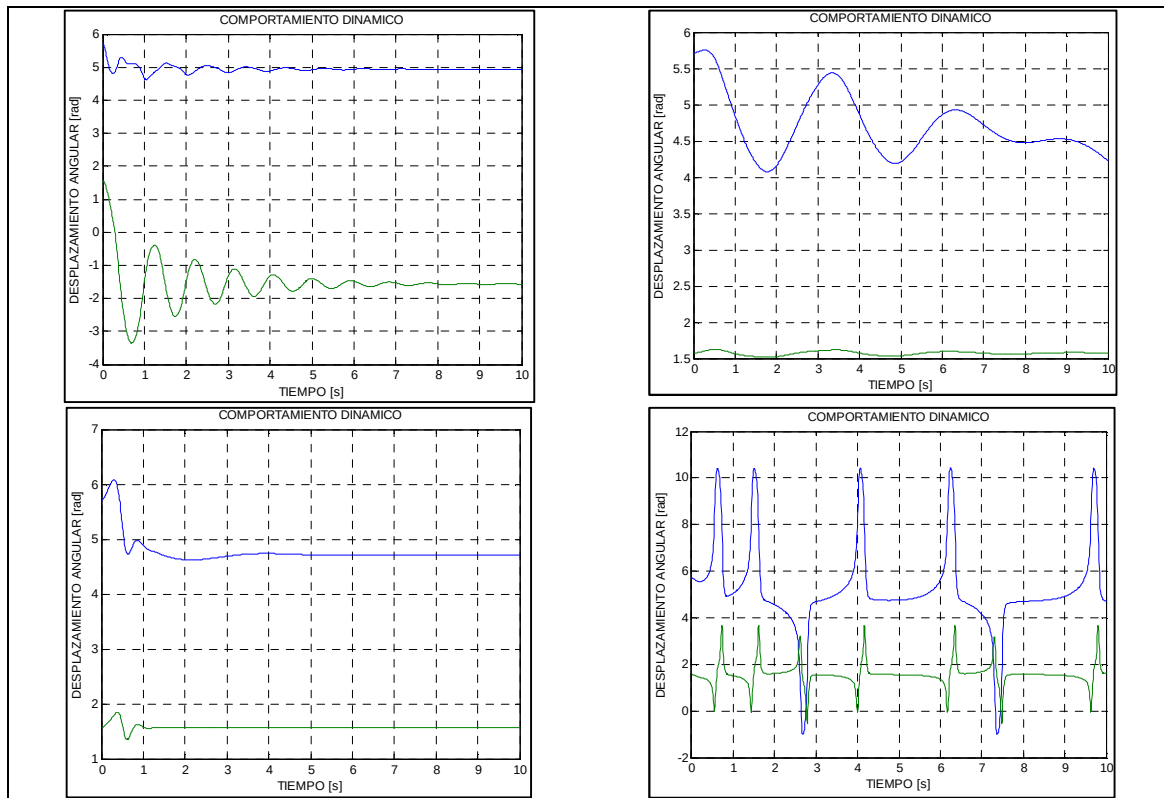


Fig. 6.32. Comportamiento AG 20 evoluciones



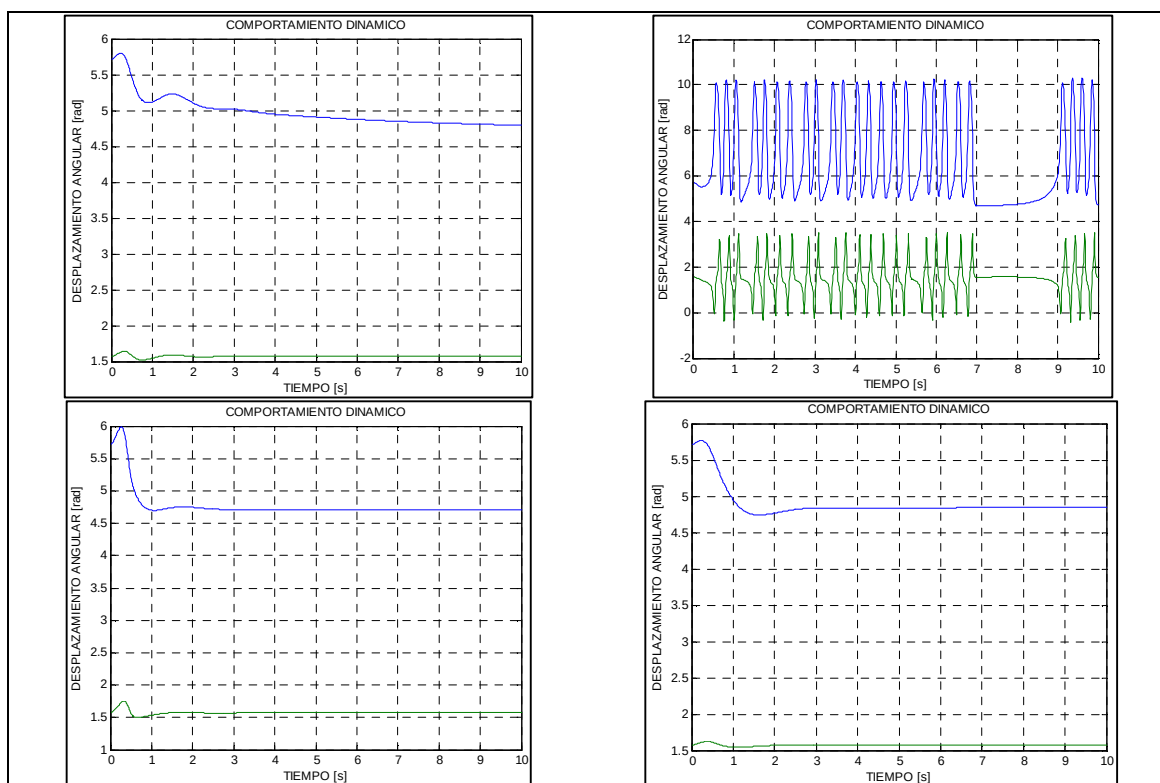


Fig. 6.35. Comportamiento AG 8 evoluciones

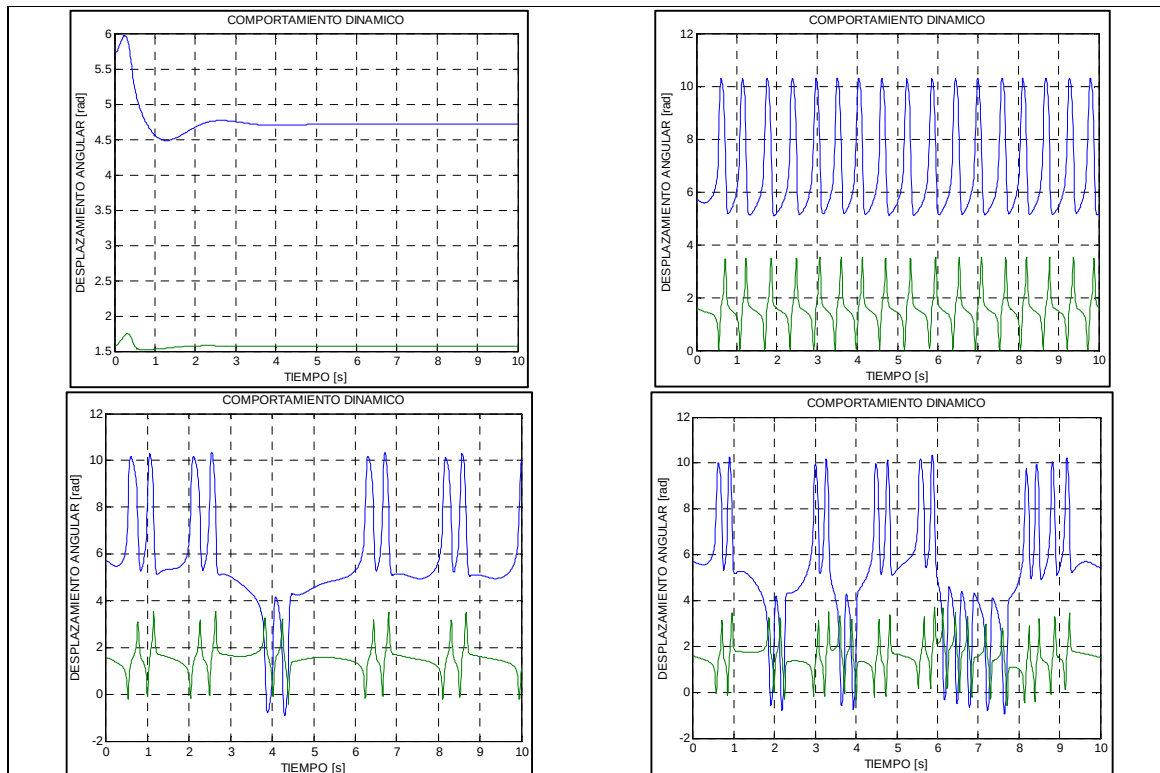


Fig. 6.36. Comportamiento AG 4 evoluciones

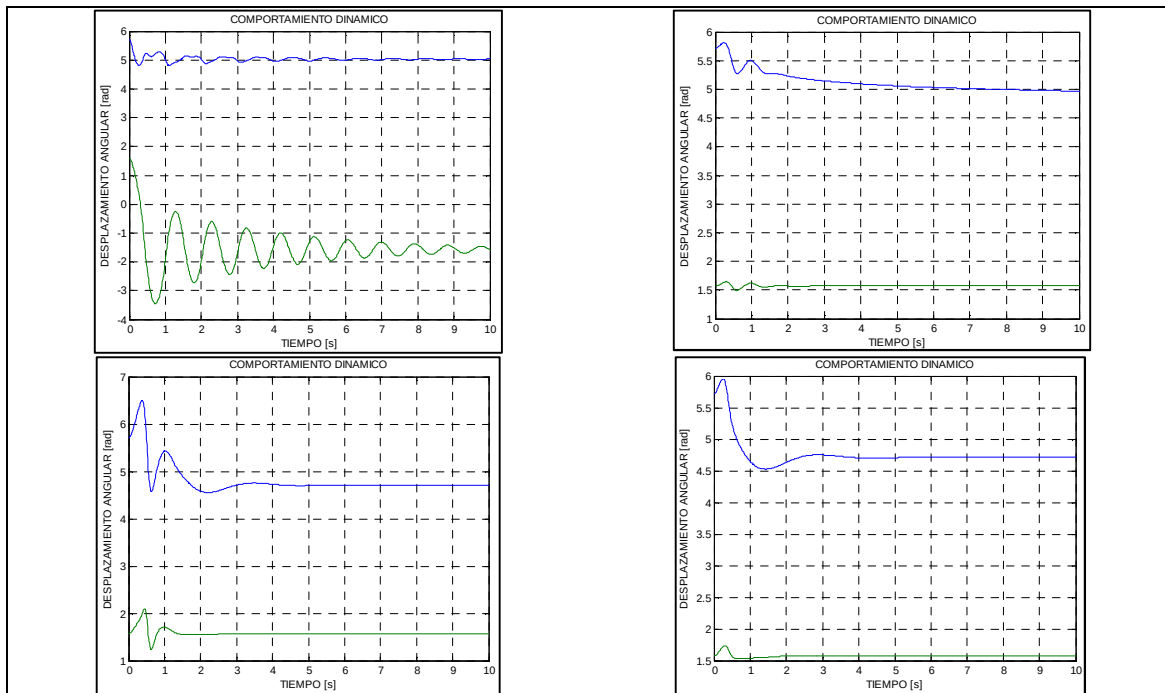


Fig. 6.37. Comportamiento AG 1 evolución

6.6.2.- VARIACIÓN EN EL ALGORITMO DE EVOLUCIÓN DIFERENCIAL

El siguiente sistema a analizar en cuanto a su comportamiento referente a las variaciones en las evoluciones es el algoritmo de evolución diferencial, bajo el cual se inicializa el análisis en 20 evoluciones con una disminución de 2 unidades manteniendo todos los parámetros restantes constantes.

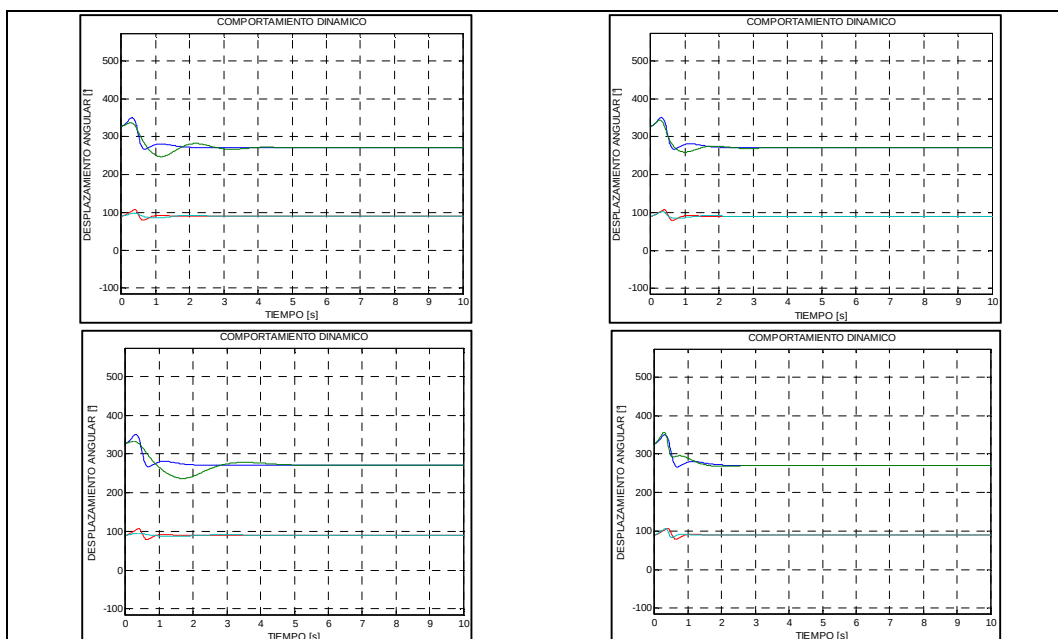


Fig. 6.38. Comportamiento ED 20 evoluciones

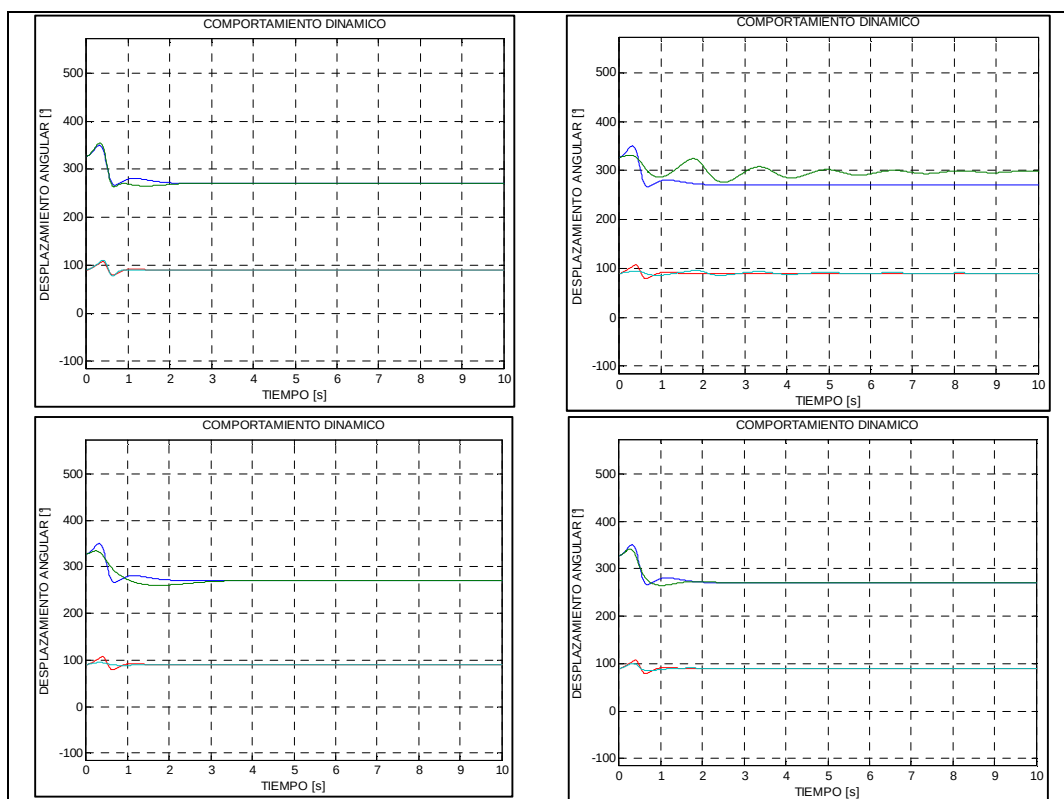


Fig. 6.39. Comportamiento ED 16 evoluciones

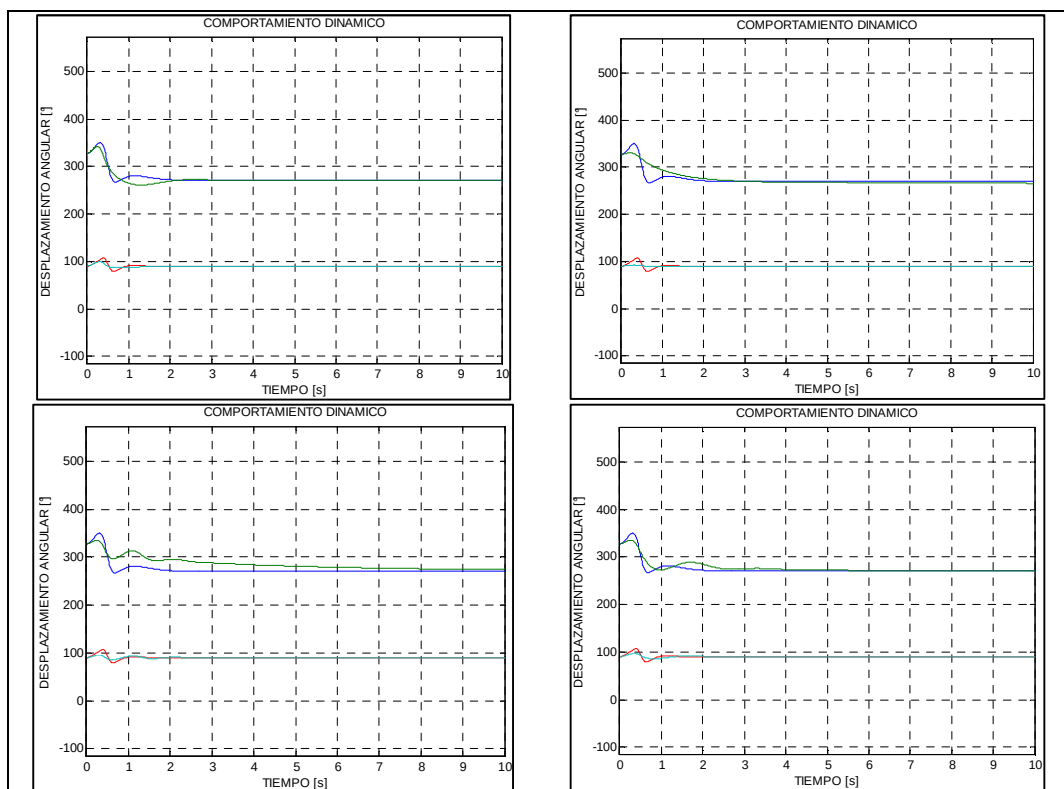


Fig. 6.40. Comportamiento ED 12 evoluciones

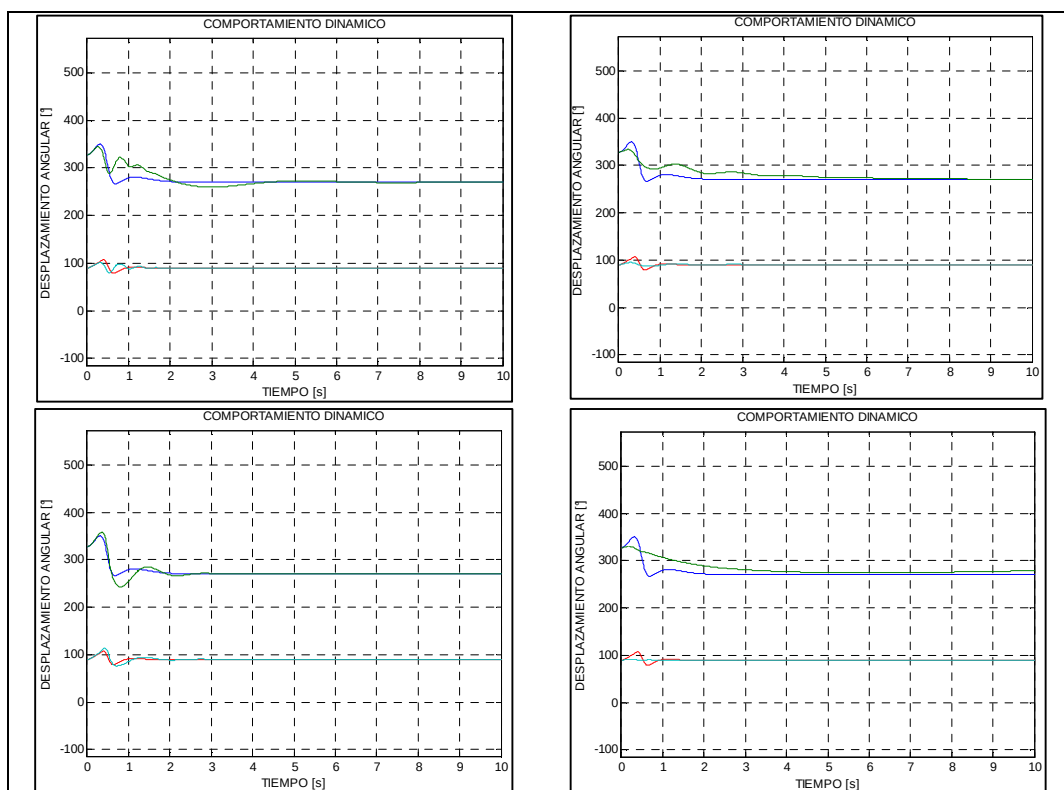


Fig. 6.41. Comportamiento ED 8 evoluciones

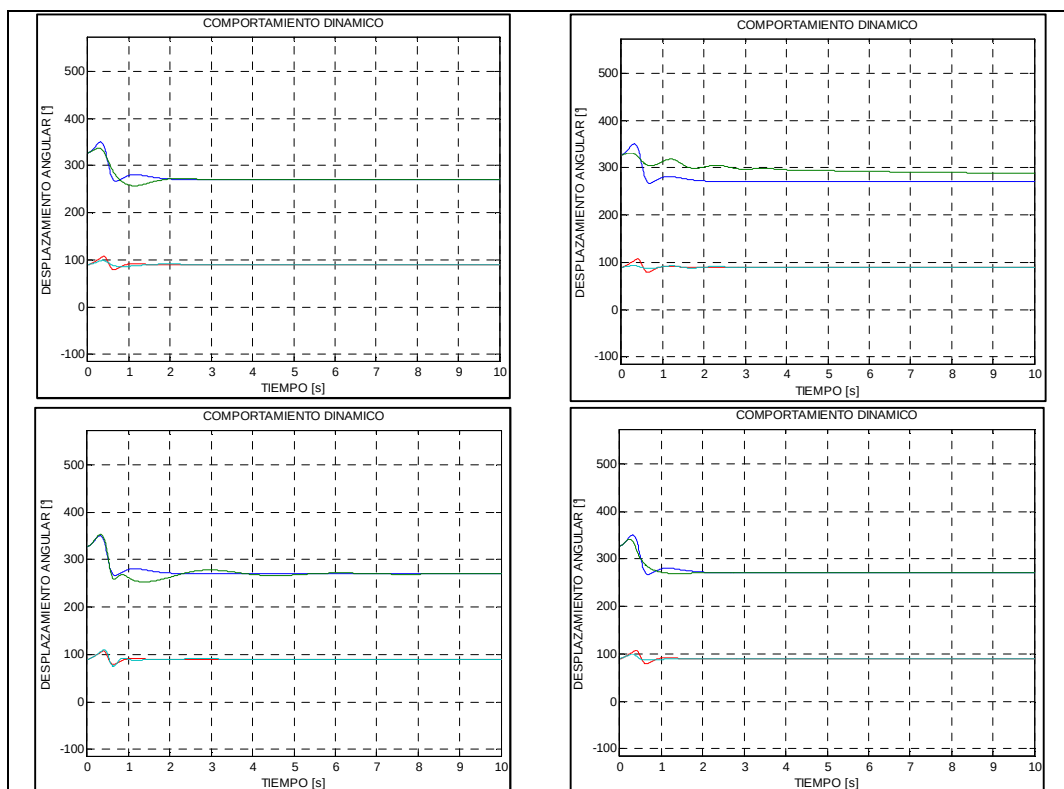


Fig. 6.42. Comportamiento ED 4 evoluciones

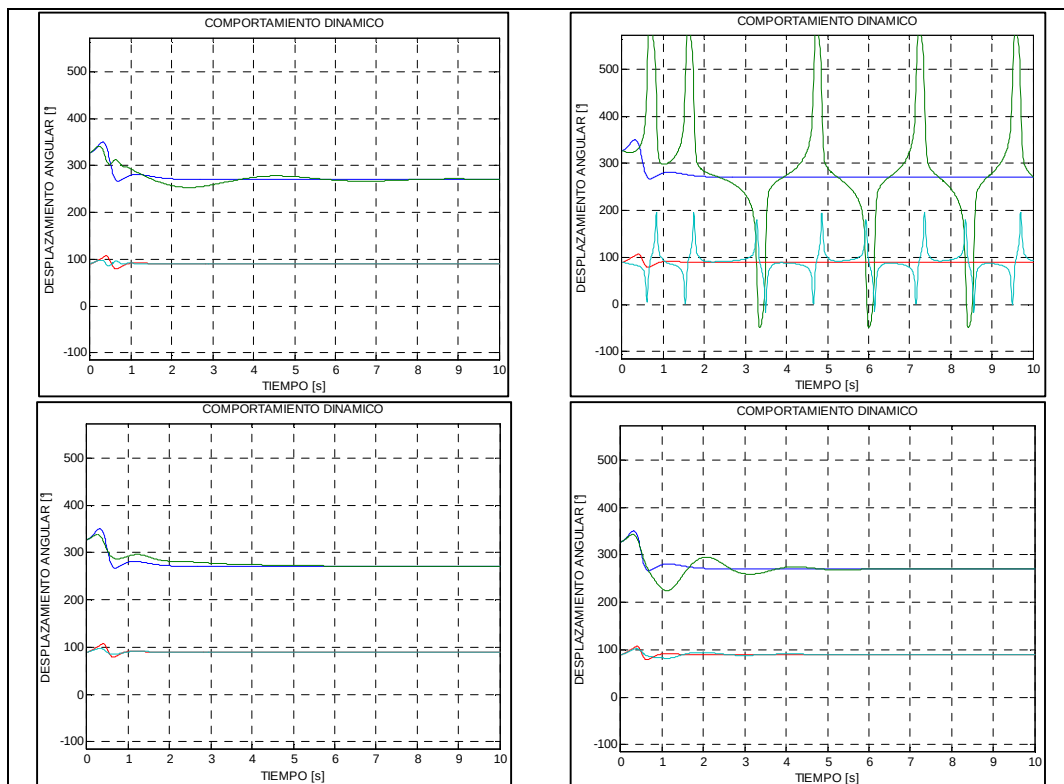


Fig. 6.43. Comportamiento ED 1 evolución

En la serie de figuras número 6.38 a 6.43 se ilustra el comportamiento que toma el algoritmo de evolución diferencial al estar sometido a las variaciones en las evoluciones definidas.

Se puede observar que el algoritmo mantiene un comportamiento estable referente a la variación presente en las evoluciones, es decir el algoritmo sigue controlando el sistema dinámico en cuestión hasta una disminución considerable de las evoluciones.

6.6.3.- VARIACIÓN EN EL ALGORITMO OPTIMIZACIÓN POR ENJAMBRE DE PARTÍCULAS

Se analiza ahora el comportamiento del algoritmo de optimización por enjambre de partículas y su comportamiento frente a la disminución del número de iteraciones. Se realiza la prueba siguiente como en el caso del algoritmo genético y el algoritmo de evolución diferencial manteniendo constantes todos los parámetros que intervienen en el mismo.

Las figuras número de 6.44 a 6.49 ilustran el comportamiento del sistema dinámico de estudio al ser evaluado con un número de iteraciones variables, se ilustran varios ejemplos característicos del comportamiento del sistema, donde se puede observar que el sistema tiende a tomar un comportamiento inestable referente a la disminución de las iteraciones en el algoritmo.

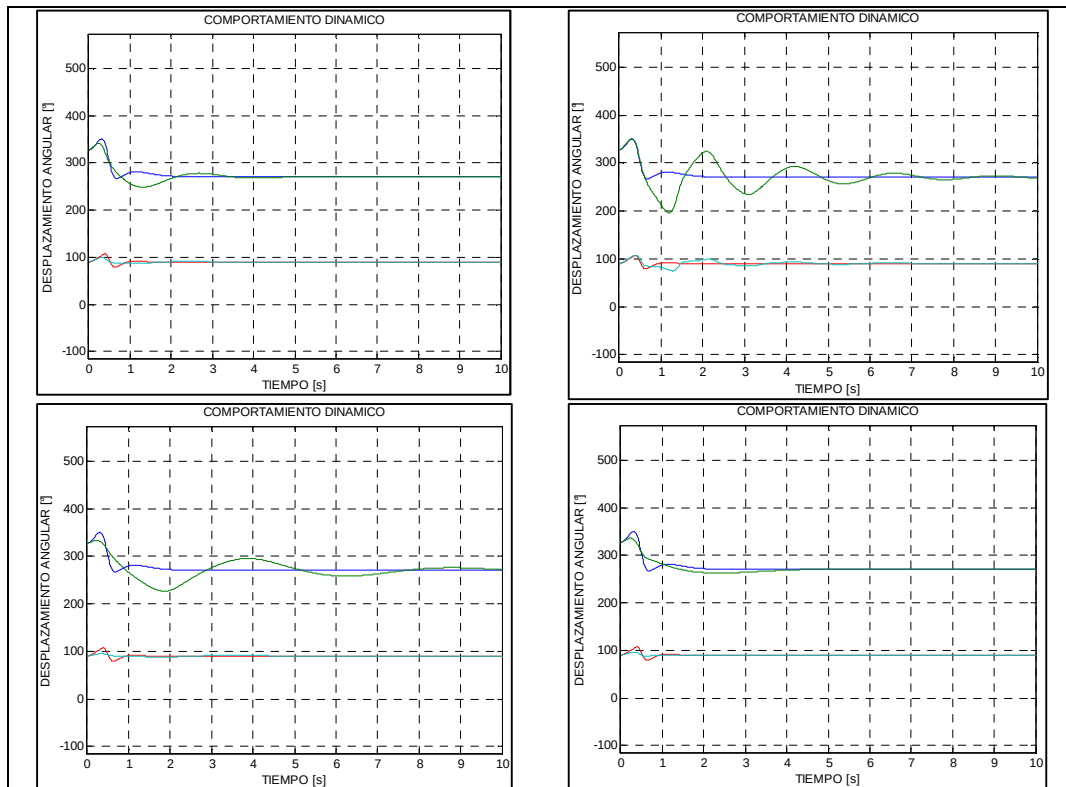


Fig. 6.44. Comportamiento PSO 20 evoluciones

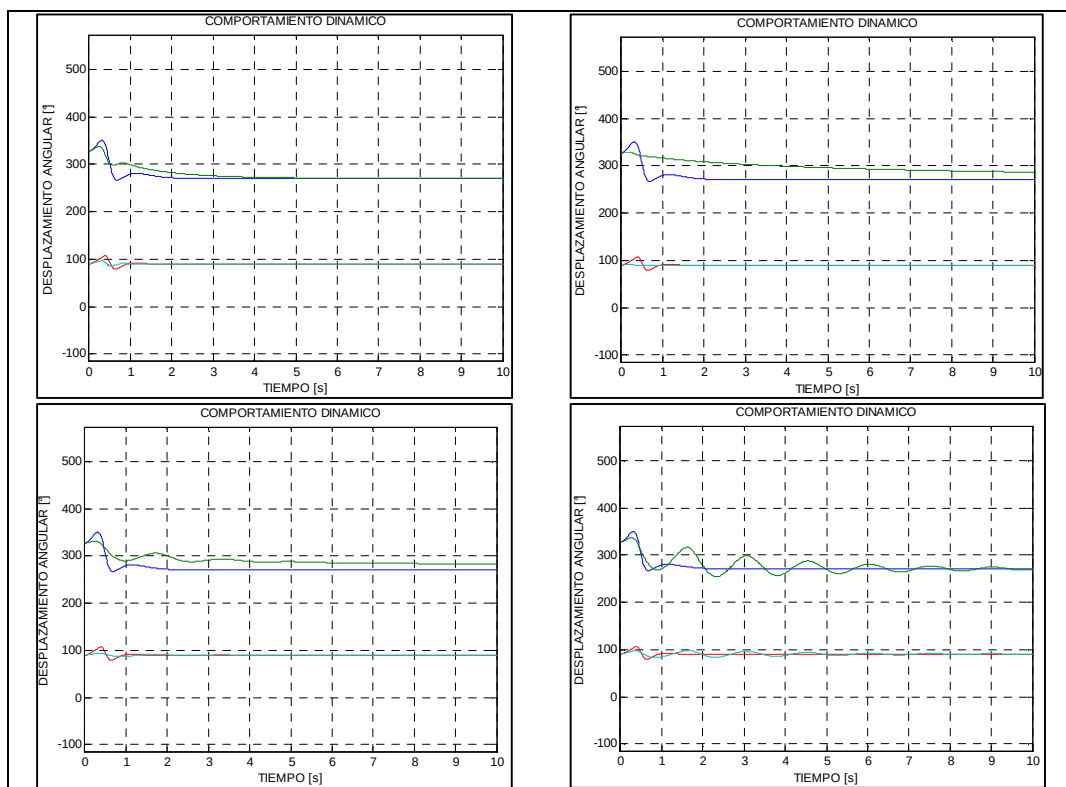


Fig. 6.45. Comportamiento PSO 16 evoluciones

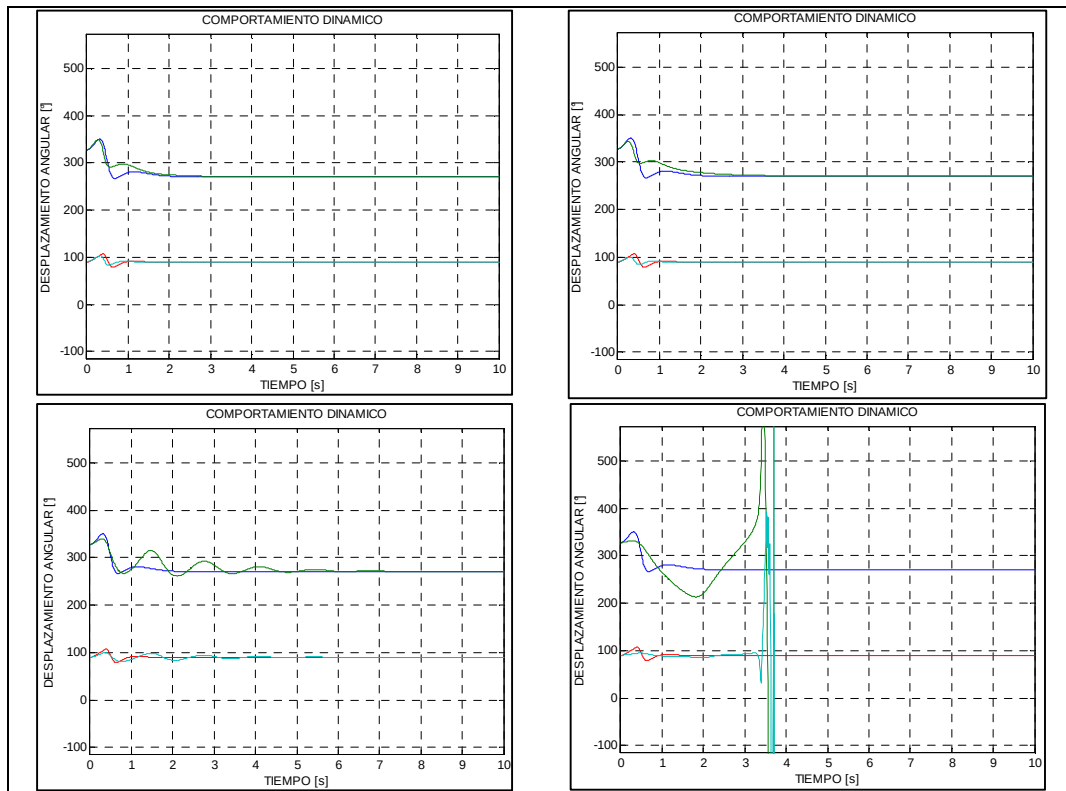


Fig. 6.46. Comportamiento PSO 12 evoluciones

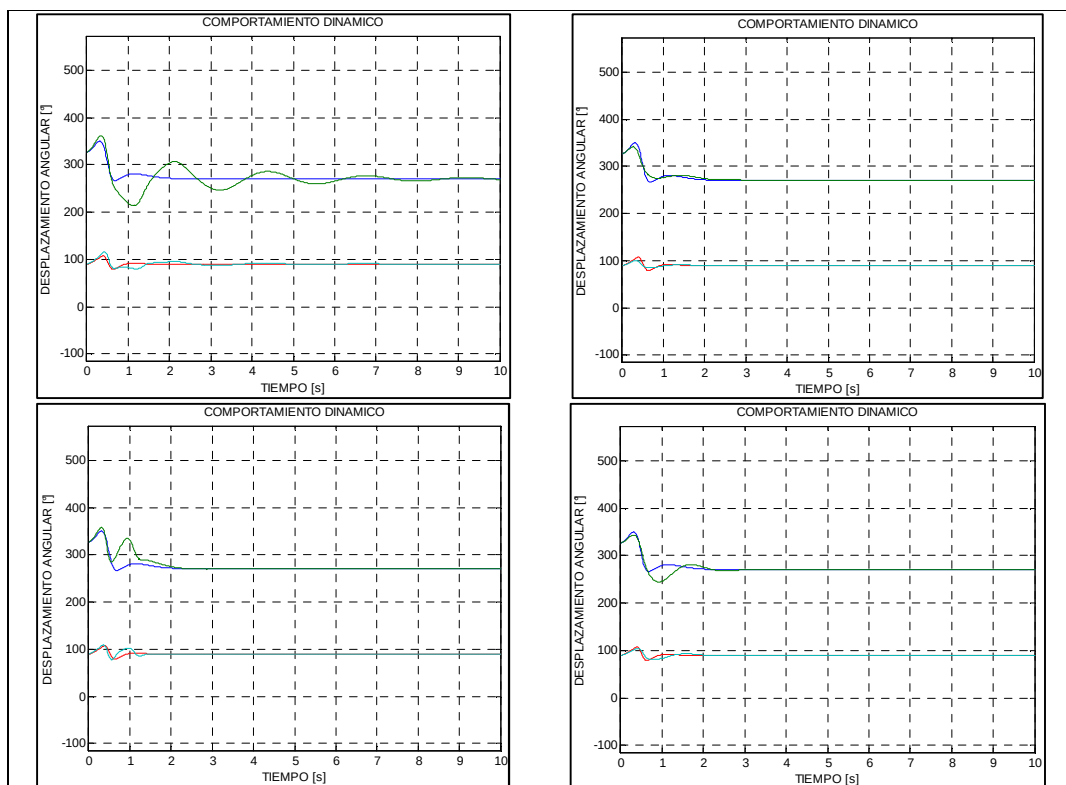


Fig. 6.47. Comportamiento PSO 8 evoluciones

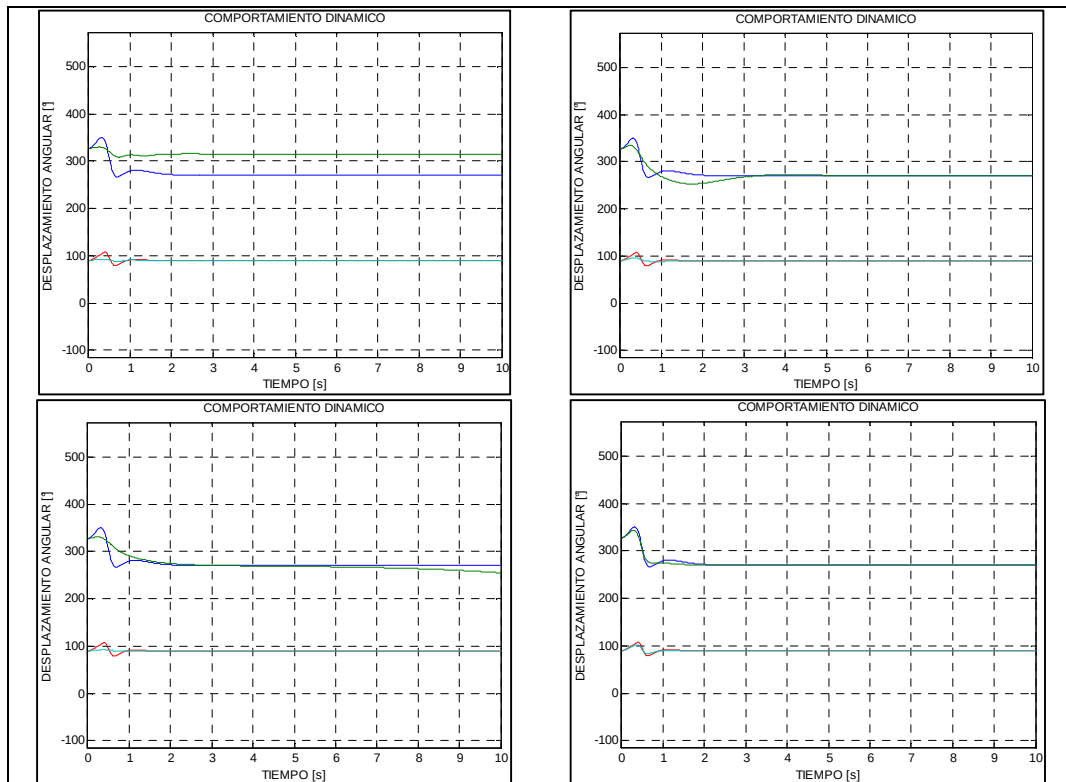


Fig. 6.48. Comportamiento PSO 4 evoluciones

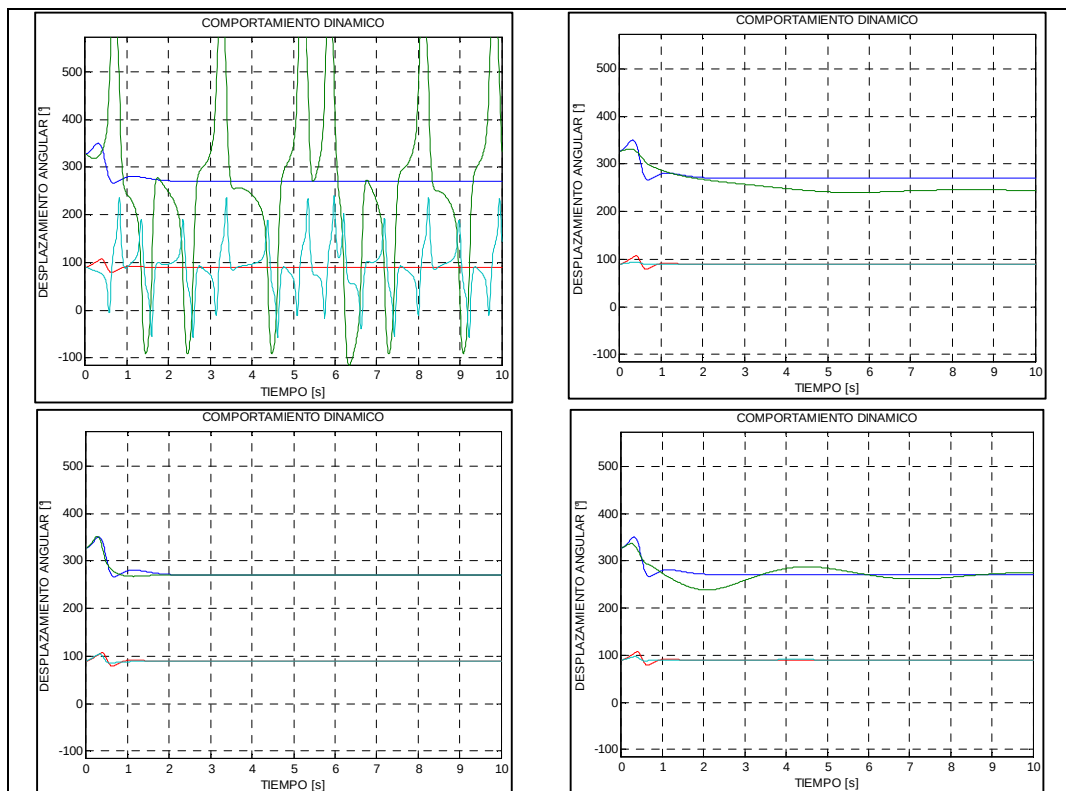


Fig. 6.49. Comportamiento PSO 1 evolución

6.7.-TASA DE MUTACIÓN PRESENTE EN LOS SISTEMAS

Se analiza el comportamiento de la mutación en cada uno de los algoritmos desarrollados, este factor es importante porque brinda al algoritmo diversidad de búsqueda, y analizar el comportamiento del algoritmo bajo estas condiciones, lo que se desea conocer es como este factor afecta en el proceso de localización de ganancias de sintonización del sistema dinámico.

6.7.1.- MUTACIÓN EN EL ALGORITMO GENÉTICO

Se desarrolla la prueba del parámetro mutación del algoritmo genético bajo las características anteriormente expuestas del sistema dinámico en cuestión. La mutación es la segunda forma en que un GA explora una superficie de costo. Las mutaciones aleatorias desarrolladas en el código tienen como finalidad alterar un cierto porcentaje de individuos. Esto con el fin de que el algoritmo converja en un punto óptimo local y no determine realmente el valor deseado. En las imágenes número 6.50 a 6.51 se presentan los resultados del estudio referente a la variación de la mutación y como obtuvo el algoritmo la solución buscada.

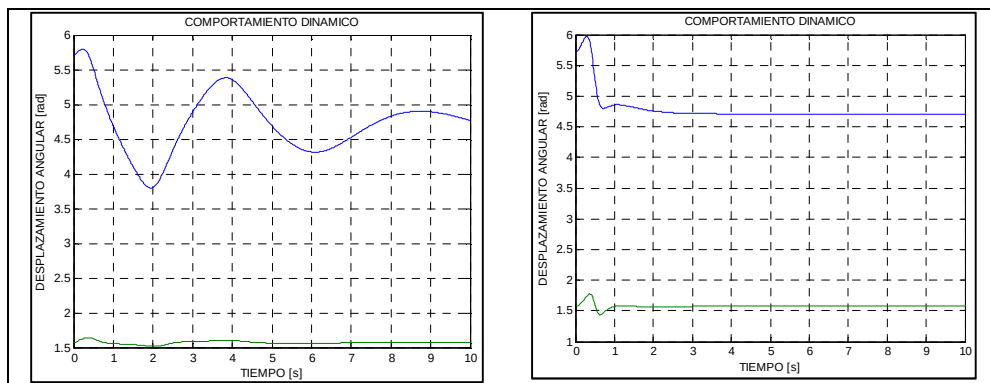


Fig. 6.50. Resultado comportamiento AG. Tasa de mutación 1% (Izquierda) tasa de mutación 5% (derecha)

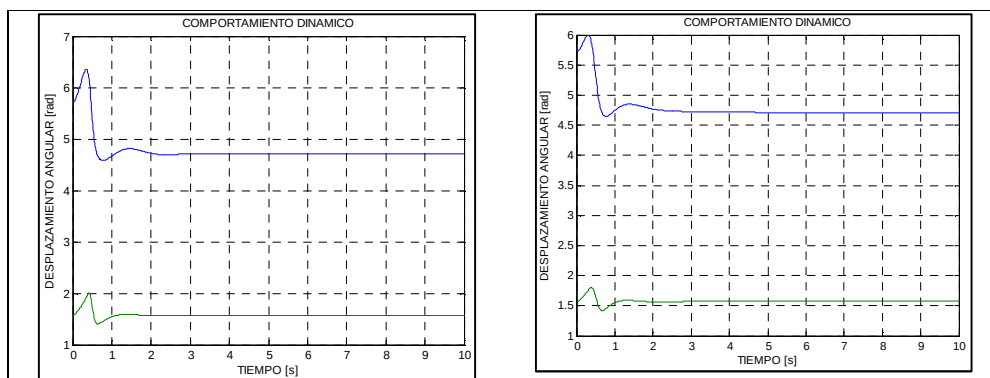


Fig. 6.51. Resultado comportamiento AG. Tasa de mutación 10% (izquierda), tasa de mutación 20% (derecha)

6.7.2.- MUTACIÓN EN EL ALGORITMO DE EVOLUCIÓN DIFERENCIAL

Se desarrollan a continuación las pruebas sobre el efecto de la mutación referente al algoritmo de evolución diferencial, la forma de operar del algoritmo de evolución diferencial es que, una vez inicializado, este muta y se recombina para producir un par de elementos de prueba con la capacidad de explorar el entorno del dominio de solución para que con ello se pueda presentar mayor capacidad de análisis del sistema y no se le permita alcanzar máximos locales como en el caso del algoritmo genético.

En particular, la mutación diferencial añade un parámetro bajo el cual se agrega cierto peso que intercambia el comportamiento de los elementos en la población para obtener mayor diversidad en la misma. En la secuencia de figuras número 6.52 a 6.53 se presenta el desarrollo de las pruebas efectuadas al algoritmo con diferentes tasas de mutación presente en el mismo bajo los parámetros establecidos constantes. Con base al comportamiento del mismo se puede observar que este parámetro añadido al sistema realizan la función en el algoritmo de volverlo muy sensible al comportamiento de la mutación en el sistema.

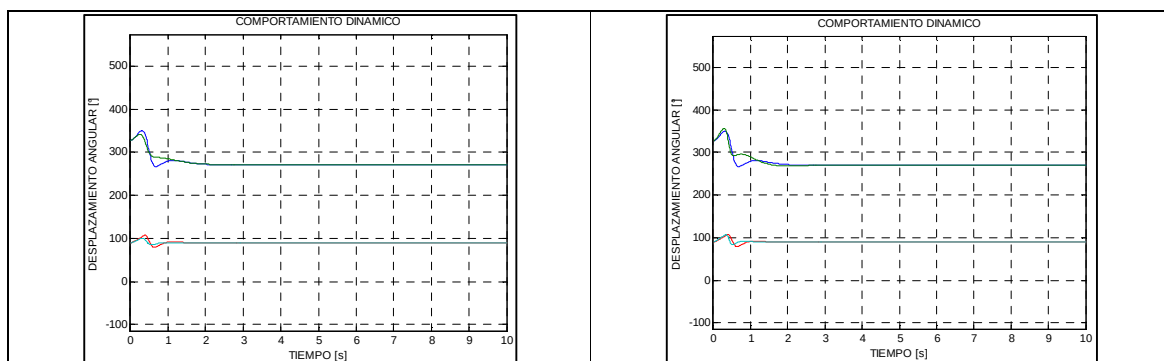


Fig. 6.52. Resultado comportamiento ED. Tasa de mutación 1% (izquierda), tasa de mutación 2% (derecha)

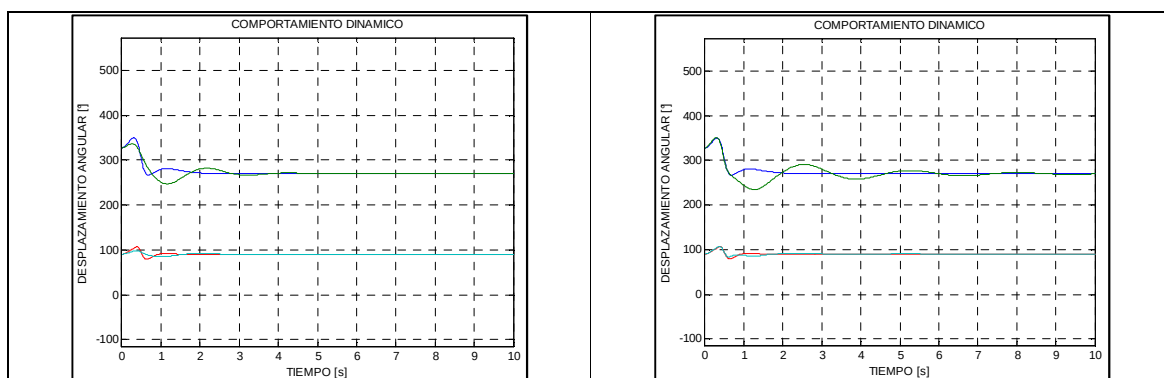


Fig. 6.53. Resultado comportamiento ED. Tasa de mutación 5% (izquierda), tasa de mutación 7% (derecha)

6.7.3.- MUTACIÓN EN EL ALGORITMO DE OPTIMIZACION POR ENJAMBRE DE PARTÍCULAS

Se presentan los resultados del análisis del sistema de control sintonizado mediante algoritmo de optimización por enjambre de partículas con presencia de mutación. Una vez definidos los parámetros constantes de inicialización del algoritmo, el factor mutación tiene la tarea de ampliar el rango de búsqueda de las ganancias para el sistema mecánico mediante la variación presente en las posiciones de los individuos, un comportamiento similar al que toma la mutación presente en los algoritmos genéticos o el de evolución diferencial estudiados con anterioridad. Se muestra a continuación el resultado de las pruebas en las figuras número 6.54 a 6.55, donde se analizan los resultados obtenidos por el sistema, se puede observar que a lo largo de las evoluciones que el sistema genera poblaciones dispersas que analizan todo el comportamiento del sistema logrando que algunas no sintonicen el mismo.

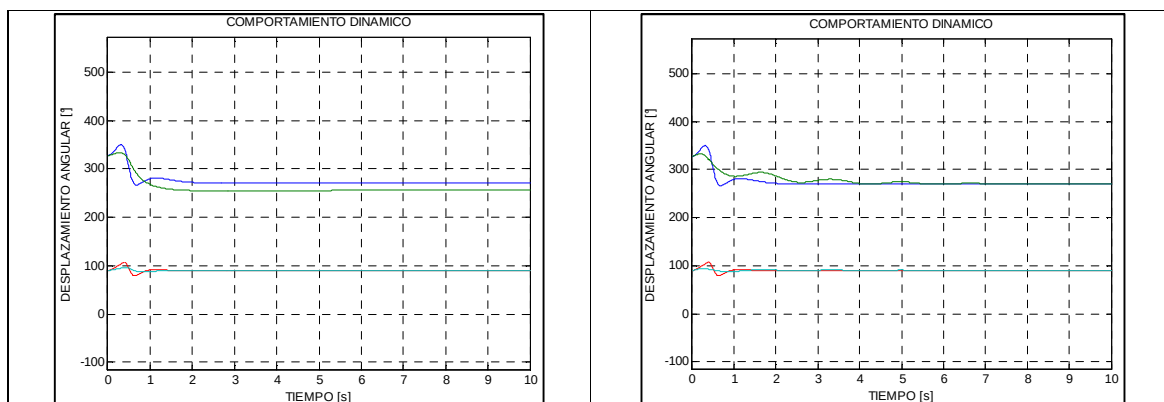


Fig. 6.54. Resultado comportamiento PSO. Tasa de mutación 1% (izquierda), tasa de mutación 2% (derecha)

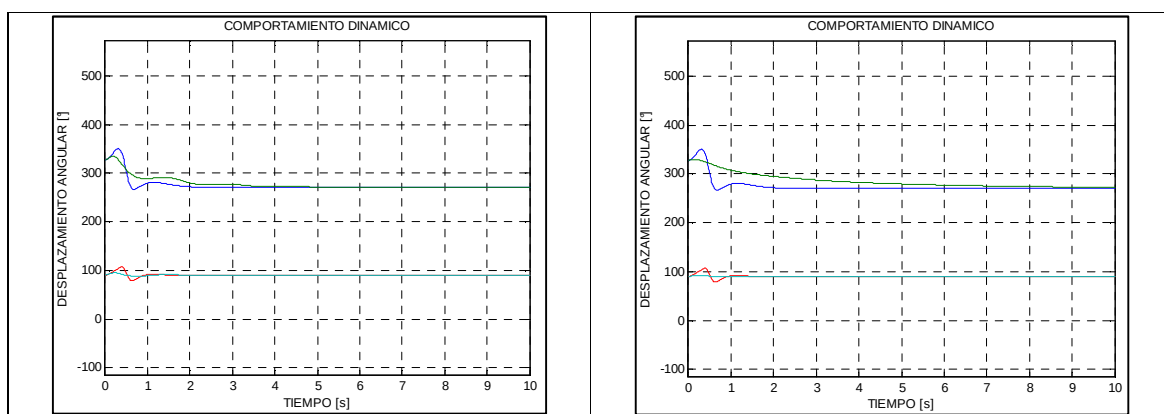


Fig. 6.55. Resultado comportamiento PSO. Tasa de mutación 5% (izquierda), tasa de mutación 7% (derecha)

CAPÍTULO 7

CONCLUSIONES

Resumen:

Finalmente en el presente capítulo se realiza un análisis de los resultados obtenidos en capítulos anteriores sobre el comportamiento de cada algoritmo desarrollado, se obtienen conclusiones de cada medición de efectividad de los algoritmos y se determina en base a estas cual es el más propicio para ser utilizado dependiendo la circunstancia deseada. Al final se presentan recomendaciones y trabajos futuros propuestos.

7.1.- INTRODUCCIÓN

A lo largo del desarrollo del presente trabajo se observó que el sistema dinámico de estudio se logró modelar con base a la formulación de Lagrange para sistemas discretos, bajo la formulación se desarrolla el control por el método de variables de estado, donde la sintonización del sistema se realiza por diversos métodos.

En primer lugar se ocuparon teoremas de linealización del sistema dinámico para efectuar el control lineal, basado en este control lineal se efectúa en el punto de equilibrio inestable en control del sistema no lineal. Bajo la configuración anterior se efectúa el control del sistema no lineal por métodos no gradientales; a saber, algoritmo genético, algoritmo de evolución diferencial y algoritmo de optimización por enjambre de partículas; el análisis de los mismos junto con el comportamiento del sistema bajo estos métodos de sintonización a nivel simulación constituye el principal tema de investigación en el trabajo desarrollado.

7.2.- ANÁLISIS DE SENSIBILIDAD DE LOS ALGORITMOS REFERENTES A LOS LÍMITES DE LAS CONDICIONES INICIALES

Se analizó el comportamiento del sistema dinámico y la acción de control mediante métodos tradicionales (lineal y no lineal) y los algoritmos basados en computación evolutiva bajo las variaciones en las condiciones iniciales, del sistema, esto con el fin de determinar si alguna variación en las mismas generaba resultados variados dependiendo del sistema de sintonización de estudio.

Se observa que el sistema dinámico linealizado y controlado bajo esta premisa logra sintonizarse y estabilizarse en la posición de equilibrio no importando cual sea la condición inicial expuesta, no así el control del sistema dinámico no lineal basado en el control lineal dado que este control solo es válido bajo una variación máxima permisible, la cual es igual a:

$$\Delta x_1 = 1.053017003$$

Sobre este valor el sistema dinámico no lineal no logra controlarse por métodos convencionales.

También se analiza el comportamiento del sistema en cuestión bajo métodos de sintonización basados en computación evolutiva, los métodos de estudio descritos anteriormente muestran un comportamiento similar al sistema no lineal controlado por métodos tradicionales, en general se concluye que las variaciones máximas permitidas para el sistema no lineal independiente del sistema de sintonización está en el orden de la variación definida con anterioridad (1.053 radianes).

7.3.- SENSIBILIDAD DE ALGORITMOS REFERENTES A LOS LÍMITES DE LAS GANANCIAS

Una vez establecido que el comportamiento de los sistemas de sintonización gradienta y no gradiente tienden a tomar el mismo comportamiento con respecto a las variaciones en las condiciones iniciales; el siguiente parámetro a analizar fue el comportamiento del sistema bajo las variaciones en los límites de las ganancias de los sistemas de sintonización basados en computación evolutiva.

Bajo este análisis se determinaron los valores límites de búsqueda de ganancias del sistema, es decir, hasta dónde es sensible el algoritmo, bajo ciertos dominios de búsquedas de las ganancias de sintonización lograr establecer el control del sistema de la manera solicitada, para cada algoritmo de prueba se establecieron los límites solicitados, dando como resultado.

Para el algoritmo genético:

$$\begin{aligned}k_1 &= [-0.76, 0.76] \\k_2 &= [0.24, 2.76] \\k_3 &= [-0.27, 0.71] \\k_4 &= [0.35, 0.39]\end{aligned}$$

Para el algoritmo de evolución diferencial:

$$\begin{aligned}k_1 &= [-0.55, 0.5] \\k_2 &= [0.1, 2.25] \\k_3 &= [-0.15, 0.15] \\k_4 &= [0, 0.225]\end{aligned}$$

Para el algoritmo de optimización por enjambre de partículas:

$$\begin{aligned}k_1 &= [-0.8, 0.8] \\k_2 &= [0, 2.2] \\k_3 &= [-0.4, 0.7] \\k_4 &= [0, 0.4]\end{aligned}$$

Con lo cual se observa que el algoritmo que tiene un desempeño bajo límites más acotados de búsqueda es el algoritmo de evolución diferencial, dado que este permite trabajar bajo un menor intervalo de datos.

El algoritmo que tiene un desempeño más erróneo bajo la disminución del intervalo de búsqueda de ganancias es el algoritmo de optimización por enjambre de partículas.

7.4.- VARIACIÓN EN EL TIEMPO DE CÓMPUTO

Basados en el tiempo que tardan en efectuar la búsqueda y la sintonización del control cada uno de los algoritmos de estudio se realizó una prueba estadística de los mismos basados en cien ciclos de búsqueda para cada metodología de sintonización y bajo los mismos parámetros en cada de una de ellas.

Se presenta el histograma que grafica el comportamiento de cada algoritmo evolutivo y se determinó el promedio de cada uno de ellos dando como resultado.

Para el algoritmo genético:

$$\text{tiempo de cómputo} = 235.70 \text{ segundos}$$

Para el algoritmo de evolución diferencial:

$$\text{tiempo de cómputo} = 109.77 \text{ segundos}$$

Para el algoritmo de optimización por enjambre de partículas:

$$\text{tiempo de cómputo} = 94.29 \text{ segundos}$$

Observando que el mejor desempeño lo obtuvo el algoritmo de optimización por enjambre de partículas, siendo el algoritmo genético el que obtuvo el peor desempeño frente a este parámetro de medición.

7.5.- COMPORTAMIENTO DE LOS ALGORITMOS FRENTE A LA DISMINUCIÓN DE INDIVIDUOS

Con base al número de individuos bajo los cuales cada uno de los algoritmos se efectúa se realizaron pruebas con una tasa de disminución de individuos de 20 inicializando el cálculo con 300 manteniendo constantes el resto de los parámetros involucrados en los mismos.

Bajo lo cual se obtuvieron como resultados para el algoritmo genético que en general este presentada un desempeño medio siendo la tasa de fallo alrededor de 60 individuos, es decir para un número de individuos inferior a este el algoritmo genético en general no lograba sintonizar el mismo.

Para el algoritmo de evolución diferencial se observó un comportamiento muy aceptable, incluso superior al algoritmo genético para valores de 80 individuos, es decir, este algoritmo lograba la sintonización de cada prueba solicitada arriba de este número de individuos, sin embargo bajo este número de individuos el sistema se volvía altamente errático y no lograba sintonizar casi ninguna prueba solicitada.

El caso del algoritmo de optimización por enjambre de partículas demostró ser efectivo para un número de individuos cercanos o superiores a los 300, cuando se disminuyó la cantidad de individuos, este algoritmo mostró ser muy sensible a este parámetro mostrando casi un completo colapso en la tarea de sintonización del control para individuos debajo de 200.

Con base a los resultados anteriormente presentados, se puede observar que el algoritmo genético es el más efectivo en la sintonización del control para un número pequeño de individuos, el algoritmo que obtuvo el peor desempeño es el algoritmo de optimización por enjambre de partículas.

7.6.- SENSIBILIDAD DE LOS ALGORITMOS FRENTE A LA VARIACIÓN EN EL NÚMERO DE EVOLUCIONES

Con base al número de evoluciones permitidas para cada algoritmo y bajo los parámetros fijos demás involucrados en cada uno de ellos se observaron los siguientes comportamientos. Para el algoritmo genético se observa que para valores inferiores a las 6 evoluciones este no logra generar valores que sintonicen el comportamiento del mismo.

Para el algoritmo de evolución diferencial se tiene un número de evoluciones inferior a 2, es decir este algoritmo logró encontrar valores en general desde la primer evolución, no siendo necesarias más para su correcta sintonización. Con valores superiores a este número el algoritmo lograba sintonizar el sistema dinámico sin mayores dificultades. Por último el algoritmo de optimización por enjambre de partículas logró sintonizarse en promedio para un número de evoluciones superiores a 6, bajo un número inferior de evoluciones el algoritmo no lograba tener un desempeño adecuado.

7.7.- TASA DE MUTACIÓN

Con base al parámetro de mutación presente en el sistema se observó que en general para el algoritmo genético este es una pieza fundamental en el correcto funcionamiento del mismo, se determinaron valores bajo los cuales el mismo no puede sintonizar el control del sistema dinámico de manera eficiente. Como lo muestran los resultados obtenidos. Para el algoritmo de evolución diferencial se puede observar que la mutación es un parámetro que en general lograba sintonizar el comportamiento del sistema, pero a mayores tasas de la misma el algoritmo no logró su objetivo siendo este parámetro perjudicial en el mismo, logrando como resultado un sistema caótico.

En el caso del algoritmo de optimización por enjambre de partículas se obtuvo un comportamiento en algunos aspectos similar al comportamiento del algoritmo de evolución diferencial, es decir, el algoritmo presentaba un comportamiento bajo el cual conforme este va evolucionando se observa capacidad de búsqueda de los individuos, pero a mayor tasa de mutación no logra sintonizar el comportamiento del sistema bajo las condiciones constantes establecidas. En general se observa que el algoritmo genético obtiene un mejor desempeño con elevadas tasas de mutación presente en el mismo. No así los algoritmos de evolución diferencial y de optimización por enjambre de partículas.

7.8.- CONCLUSIONES GENERALES

Prueba efectuada	Algoritmo de prueba	Parámetros obtenidos	Conclusión obtenida
Medición de sensibilidad referente a la variación en condiciones iniciales	AG	$\Delta x_1 = 1.05301$	Estable
	ED	$\Delta x_1 = 1.05301$	
	PSO	$\Delta x_1 = 1.05301$	
Medición de sensibilidad referente a la variación en límites de búsqueda de ganancias	AG	$k_1 = [-0.76, 0.76]$ $k_2 = [0.24, 2.76]$ $k_3 = [-0.27, 0.71]$ $k_4 = [0.35, 0.39]$	Adecuado
	ED	$k_1 = [-0.55, 0.5]$ $k_2 = [0.1, 2.25]$ $k_3 = [-0.15, 0.15]$ $k_4 = [0, 0.225]$	
	PSO	$k_1 = [-0.8, 0.8]$ $k_2 = [0, 2.2]$ $k_3 = [-0.4, 0.7]$ $k_4 = [0, 0.4]$	
Medición de la variación referente al tiempo de computo consumido por cada algoritmo	AG	230 segundos	Solucionado
	ED	114 segundos	
	PSO	95 segundos	
Medición de la sensibilidad de los algoritmos referentes a la disminución de individuos.	AG	240 Individuos	Falla
	ED	140 Individuos	
	PSO	280 Individuos	
Medición de sensibilidad referente a la variación en el número de evoluciones de cada algoritmo.	AG	4 Evoluciones	Converge
	ED	2 Evoluciones	
	PSO	8 Evoluciones	
Medición de sensibilidad referente a la variación en la tasa de mutación presente en el algoritmo	AG	20 porciento	Optima
	ED	7 porciento	
	PSO	9 porciento	

Tabla 7.1. Resumen de resultados basados en las pruebas realizadas

Se puede observar que los objetivos planteados se cumplieron, se logró desarrollar la sintonización del control del sistema péndulo doble en el punto de equilibrio inestable de manera eficiente y con ello todos los objetivos particulares

planteados. Con base a los análisis efectuados al sistema se concluye que la hipótesis planteada sobre la sintonización por medio de algoritmos evolutivos es verdadera, dado que estos nos sirven para analizar el comportamiento de sistemas dinámicos de una forma mucho más sencilla a la sintonización por métodos gradientales.

7.9.- TRABAJOS FUTUROS

Con base al presente trabajo y los resultados obtenidos queda como trabajo para futuras investigaciones el desarrollo del modelo físico del péndulo doble invertido y la acción de control del sistema real digital para comprobar los resultados físicos contra los modelos computacionales y matemáticos.

Se desea profundizar en el comportamiento de la mutación de los sistemas de control estudiados, es decir, realizar una formulación más robusta de la mutación para los algoritmos de evolución diferencial y de optimización por enjambre de partículas.

Queda como labor robustecer más el modelo dinámico del sistema de estudio con parámetros como la elasticidad presente en el sistema y el comportamiento del amortiguamiento del mismo.

Determinar el comportamiento del control del sistema dinámico de estudio bajo otros métodos de acción de control como es el caso de control robusto, control adaptativo o control óptimo, por poner un ejemplo.

Queda pendiente analizar el comportamiento de otros sistemas dinámicos subactuados complejos por medio de los lenguajes desarrollados en el presente trabajo y del mismo modo modelar el comportamiento físico de los mismos.

Se desea del mismo modo en un futuro explorar otros comportamientos no gradientales basados en computación evolutiva para evaluar el comportamiento de estos y obtener un mayor rango de conocimiento del funcionamiento de ellos. En la página anterior se muestra la tabla 7.1, la cual resume el comportamiento obtenido en base a cada prueba efectuada para los algoritmos de estudio.

ANEXO 1.- PROGRAMACIÓN EN MATLAB DEL MODELO DINÁMICO DEL SISTEMA PÉNDULO DOBLE INVERTIDO

```
% PROGRAMA PARA DEFINIR EL MODELO DINAMICO DEL PENDULO DOBLE NO LINEAL
%Abraham Manilla García (04 de Abril 2014)
%% definición de parámetros
dx = zeros(4,1); % vector columna
m1 = 0.25; % Masa del eslabon 1
m2 = 0.25; % Masa del eslabon 2
l1 = 0.3; % Longitud del eslabon 2
l2 = 0.3; % Longitud del eslabon 2
lc1 = 0.15; % Distancia del origen al centro de masa del eslabón 1
lc2 = 0.15; % Distancia del origen al centro de masa del eslabón 2
i1 = 1; % Momento de inercia del eslabón 1 alrededor de su centroide
i2 = 1; % Momento de inercia del eslabón 2 alrededor de su centroide
g=9.8; % Aceleración de la gravedad
Tau=0; % Torque aplicado al eslabón 1
%% Definicion de variables de factorizacion
t1 = m1*(lc1)^2+m2*(l1)^2+i1
t2 = m2*(lc2)^2+i2
t3 = m2*l1*l2
t4 = m1*lc1+m2*l1
t5 = m2*lc2
%% Definición de determinante
Det = t1*t2-((t3)^2)*(cos(x(3)))^2;
%% definición de ecuaciones
dx(1)=x(2);
dx(2)=(1/Det)*(t2*t3*((x(2)+x(4))^2)*sin(x(2))+((t3)^2)*((x(2))^2)*cos(x(3))*sin(x(3))-
t2*t4*g*cos(x(1))+t3*t5*g*cos(x(3))*cos(x(1)+x(3))+t2*Tau);
dx(3)=x(4);
dx(4)=(1/Det)*((-t3)*((x(2)+x(4))^2)*(t2+t3*cos(x(3)))*sin(x(3))-
(t1+t3*cos(x(3)))*t3*((x(2))^2)*sin(x(3)))+(t2+t3*cos(x(3)))*(t4*g*cos(x(1))-Tau)-
(t1+t3*cos(x(3)))*t5*g*cos(x(1)+x(3)));
end
```

ANEXO 2.- SOLUCIÓN NUMERICA DEL SISTEMA DINÁMICO PÉNDULO DOBLE INVERTIDO NO LINEAL

```
% -----
% Programa para resolver numéricamente el modelo de péndulo Doble
% Abraham Manilla García (05 Abril de 2014)
% -----
clc
clear all;
% intervalo de tiempo de evaluacion
tf=300;
%
options = odeset('RelTol',1e-4,'AbsTol',[1e-5 1e-5 1e-5 1e-5]);
% Condiciones iniciales y llamada a modelo matematico
[T,Y] = ode45(@Ecuacion_Pendulo_Doble_nl,[0 tf],[0 0 0 0],options);
%% Grafica Theta0
figure(1)
plot(T,Y(:,1))
grid on
title('Comportamiento x1')
xlabel('Tiempo [s]')
ylabel('Theta0 [m]')
%
%% Grafica Theta0_punto
figure(2)
plot(T,Y(:,2))
grid on
title('Comportamiento x2')
xlabel('Tiempo [s]')
ylabel('Theta0_punto [m]')
%
%% Grafica Theta1
figure (3)
plot(T,Y(:,3))
grid on
title ('Comportamiento x3')
xlabel('Tiempo [s]')
ylabel('Theta1 [m/s]')
%
%% Grafica Theta1_punto
figure (4)
plot(T,Y(:,4))
grid on
title ('Comportamiento x4')
xlabel('Tiempo [s]')
ylabel('Theta1_punto [m/s]')
%% Grafica de Fases theta1-theta1_punto
figure (5)
plot(Y(:,1),Y(:,2))
grid on
title ('Fases Theta1-Theta1_punto')
xlabel('Theta1')
ylabel('Theta1_punto [m/s]')
```

ANEXO 3.- PROGRAMACIÓN EN MATLAB PARA OBTENER EL MODELO DINÁMICO EN VARIABLES DE ESTADO

```
%
% *****PROGRAMA PARA DETERMINAR EL COMPORTAMIENTO DEL SISTEMA DINAMICO*****
% *****EN VARIABLES DE ESTADO*****
% *****24 DE ABRIL DE 2014*****
% *****ABRAHAM MANILLA GARCIA*****

%Definición de variables
syms t1 t2 t3 t4 t5 g x1 x2 x3 x4 Tau;
%Definición de la matriz no lineal A de acoplamiento de masa
h1 = t1+t2+2*t3*cos(x3);
h2 = t2+2*t3*cos(x3);
h3 = t2+t3*cos(x3);
h4 = t2;
A = [1 0 0 0; 0 h1 0 h2; 0 0 1 0; h3 0 0 h4];
B=inv(A); % Calculo de la matriz inversa de A
% Definición de la matriz G no lineal
G=[x2; 2*t3*x2*x4*sin(x3)-t4*g*cos(x1)-t5*g*cos(x1+x3); x4; -t3*(x2^2)*sin(x3)-t5*g*cos(x1+x3)];
% Definición de la matriz T no lineal
T= [0; Tau; 0; 0];
% Definición del producto matricial de la inversa de A por G
F=B*G;
% Definición del producto matricial de la inversa de A por T
Tn=B*T;
% Determinación de derivadas
Fx1 = diff(F,x1) % Derivada con respecto a x1 de F;
Fx2 = diff(F,x2) % Derivada con respecto a x2 de F;
Fx3 = diff(F,x3) % Derivada con respecto a x3 de F;
Fx4 = diff(F,x4) % Derivada con respecto a x4 de F;
Tnx1 = diff(Tn,x1) % Derivada con respecto a x1 de Tn;
Tnx2 = diff(Tn,x2) % Derivada con respecto a x2 de Tn;
Tnx3 = diff(Tn,x3) % Derivada con respecto a x3 de Tn;
Tnx4 = diff(Tn,x4) % Derivada con respecto a x4 de Tn;
%*****FIN DE PROGRAMA*****
```

ANEXO 4.- PROGRAMACIÓN EN MATLAB PARA OBTENER EL MODELO DINÁMICO EN VARIABLES DE ESTADO LINEALIZADO

```
% *****LINEALIZACIÓN DEL SISTEMA DINÁMICO NO LINEAL ALREDEDOR DEL PUNTO DE EQUILIBRIO*****
% *****27 DE ABRIL DE 2014*****
% *****ABRAHAM MANILLA GARCIA*****
% **ESTE PROGRAMA ESTA BASADO EN EL CALCULO DEL MODELO DINAMICO DEL PENDULO INVERTIDO**
% *****SI SE REALIZAN CAMBIOS EN EL MISMO ES NECESARIO MODIFICAR ESTE MODELO*****

% Definición de variables
syms t1 t3 t4 t5 g Tau;
% Definición del punto de equilibrio
x1 = pi/2;
x2 = 0;
x3 = 0;
x4 = 0;
% Definición de la función F2 vectorial a linealizar correspondiente a la segunda ecuación de F
F2 = ((t2 + 2*t3*cos(x3))*(t3*sin(x3)*x2^2 + g*t5*cos(x1 + x3)))/(t2*(t1 + t2 + 2*t3*cos(x3))) - (g*t4*cos(x1) + g*t5*cos(x1 + x3) - 2*t3*x2*x4*sin(x3))/(t1 + t2 + 2*t3*cos(x3)) + (x2*(t2 + t3*cos(x3))*(t2 + 2*t3*cos(x3)))/(t2*(t1 + t2 + 2*t3*cos(x3)));
% Definición de la función F4 vectorial a linealizar correspondiente a la segunda ecuación de F
F4 = -(t3*sin(x3)*x2^2 + g*t5*cos(x1 + x3))/t2 - (x2*(t2 + t3*cos(x3)))/t2;
% Definición de la función Tn2 vectorial, correspondiente a la segunda ecuación de Tn
Tn2 = Tau/(t1 + t2 + 2*t3*cos(x3));
% *****DEFINICIÓN DE LAS DERIVADAS DE LAS FUNCIONES A LINEALIZAR*****
% Derivada con respecto a x1 de F2;
F2x1 = (g*t4*sin(x1) + g*t5*sin(x1 + x3))/(t1 + t2 + 2*t3*cos(x3)) - (g*t5*sin(x1 + x3)*(t2 + 2*t3*cos(x3)))/(t2*(t1 + t2 + 2*t3*cos(x3)));
% Derivada con respecto a x1 de F4;
F4x1 = (g*t5*sin(x1 + x3))/t2;
% Derivada con respecto a x2 de F2;
F2x2 = (2*t3*x4*sin(x3))/(t1 + t2 + 2*t3*cos(x3)) + ((t2 + t3*cos(x3))*(t2 + 2*t3*cos(x3)))/(t2*(t1 + t2 + 2*t3*cos(x3))) + (2*t3*x2*sin(x3)*(t2 + 2*t3*cos(x3)))/(t2*(t1 + t2 + 2*t3*cos(x3)));
% Derivada con respecto a x2 de F4;
F4x2 = -(t2+t3*cos(x3))/t2 - (2*t3*x2*sin(x3))/t2;
% Derivada con respecto a x3 de F2;
F2x3 = (g*t5*sin(x1 + x3) + 2*t3*x2*x4*cos(x3))/(t1 + t2 + 2*t3*cos(x3)) - (2*t3*sin(x3)*(g*t4*cos(x1) + g*t5*cos(x1 + x3) - 2*t3*x2*x4*sin(x3)))/(t1 + t2 + 2*t3*cos(x3))^2 + ((t2 + 2*t3*cos(x3))*(t3*cos(x3)*x2^2 - g*t5*sin(x1 + x3)))/(t2*(t1 + t2 + 2*t3*cos(x3))) - (2*t3*sin(x3)*(t3*sin(x3)*x2^2 + g*t5*cos(x1 + x3)))/(t2*(t1 + t2 + 2*t3*cos(x3))) + (2*t3*sin(x3)*(t2 + 2*t3*cos(x3))*(t3*sin(x3)*x2^2 + g*t5*cos(x1 + x3)))/(t2*(t1 + t2 + 2*t3*cos(x3))^2) - (2*t3*x2*sin(x3)*(t2 + t3*cos(x3)))/(t2*(t1 + t2 + 2*t3*cos(x3))) - (t3*x2*sin(x3)*(t2 + 2*t3*cos(x3)))/(t2*(t1 + t2 + 2*t3*cos(x3))) + (2*t3*x2*sin(x3)*(t2 + t3*cos(x3))*(t2 + 2*t3*cos(x3)))/(t2*(t1 + t2 + 2*t3*cos(x3))^2);
F4x3 = (t3*x2*sin(x3))/t2 - (t3*cos(x3)*x2^2 - g*t5*sin(x1 + x3))/t2;
F2x4 = (2*t3*x2*sin(x3))/(t1 + t2 + 2*t3*cos(x3));
F4x4 = 0;
Tn2x1 = 0;
Tn2x2 = 0;
Tn2x3 = (2*Tau*t3*sin(x3))/(t1 + t2 + 2*t3*cos(x3))^2;
Tn2x4 = 0;
% Definición de las funciones evaluadas en el punto de equilibrio
F20 = F2;
F20x1 = F2x1;
F20x2 = F2x2;
F20x3 = F2x3;
F20x4 = F2x4;
F40 = F4;
F40x1 = F4x1;
F40x2 = F4x2;
F40x3 = F4x3;
F40x4 = F4x4;
Tn20 = Tn2;
Tn20x1 = Tn2x1;
Tn20x2 = Tn2x2;
Tn20x3 = Tn2x3;
Tn20x4 = Tn2x4;
% Función de aproximación con respecto a F2
F2a = F20 + F20x1*(x1-pi/2) + F20x2*(x2) + F20x3*(x3) + F20x4*(x4);
% Función de aproximación con respecto a F4
F4a = F40 + F40x1*(x1-pi/2) + F40x2*(x2) + F40x3*(x3) + F40x4*(x4);
% Función de aproximación con respecto a Tn2
Tna = Tn20 + Tn20x1*(x1-pi/2) + Tn20x2*(x2) + Tn20x3*(x3) + Tn20x4*(x4);
% Definición de coeficientes de matriz de aproximación lineal que alimentan al modelo dinámico
a21 = F20x1
a22 = F20x2
a23 = F20x3
a41 = F40x1
a42 = F40x2
a43 = F40x3
a44 = F40x4
% *****FIN DE PROGRAMA*****
```

ANEXO 5.- PROGRAMA EN MATLAB PARA DESARROLLAR PRUEBA DE CONTROLABILIDAD

```
%*****DESARROLLO DE PRUEBA DE CONTROLABILIDAD DEL SISTEMA DINAMICO*****
% *****28 DE ABRIL DE 2014*****
% *****ABRAHAM MANILLA GARCIA*****
%*****ESTE PROGRAMA ESTA BASADO EN EL CALCULO DEL MODELO DINAMICO DEL PENDULO INVERTIDO****
%*****SI SE REALIZAN CAMBIOS EN EL MISMO ES NECESARIO MODIFICAR ESTE MODELO*****
% Definición de variables globales del sistema lineal
syms t1 t2 t3 t4 t5 g;
% Definición de coeficientes de las matrices
a21 = F20x1 = (g*t4 + g*t5)/(t1 + t2 + 2*t3) - (g*t5*(t2 + 2*t3))/(t2*(t1 + t2 + 2*t3));
a22 = F20x2 = ((t2 + t3)*(t2 + 2*t3))/(t2*(t1 + t2 + 2*t3));
a23 = F20x3 = (g*t5)/(t1 + t2 + 2*t3) - (g*t5*(t2 + 2*t3))/(t2*(t1 + t2 + 2*t3));
a41 = F40x1 = (g*t5)/t2;
a42 = F40x2 = -(t2 + t3)/t2;
a43 = F40x3 = (g*t5)/t2;
b21 = (t2)/(t1*t2+((t2)^2)+2*t2*t3);
% Definición de las matrices A y B
A=[0 1 0 0; a21 a22 a23 0; 0 0 0 1; a41 a42 a43 0];
B = [0; b21; 0; 0];
% Definición de la matriz de controlabilidad
AB= [B; A*B; A*A*B; A*A*A*B];
% Definición del determinante de la matriz de controlabilidad
D = det(AB)
% fin de programa
```

ANEXO 6.- PROGRAMA EN MATLAB PARA CONTROLAR EL SISTEMA DINÁMICO LINEAL POR RETROALIMENTACIÓN DE ESTADOS

```
function dx = Ecuacion_Control_Pendulo_Doble_Lineal_2(t,x)
% PROGRAMA PARA DEFINIR EL MODELO DE CONTROL DINAMICO DEL PENDULO DOBLE LINEAL
% MODIFICANDO EL SISTEMA A PARTIR DE PARAMETROS NUEVOS
% SE REALIZAN CAMBIOS EN EL MODELO DE LOS COEFICIENTES, SE EFECTUA LA
% SUSTITUCION y1 = x1-pi/2
%Abraham Manilla García (28 de Abril 2014)

%% definición de parámetros
dx = zeros(4,1); % vector columna
m1 = 0.25; % Masa del eslabón 1
m2 = 0.25; % Masa del eslabón 2
l1 = 0.3; % Longitud del eslabon 2
l2 = 0.3; % Longitud del eslabon 2
lc1 = 0.15; % Distancia del origen al centro de masa del eslabón 1
lc2 = 0.15; % Distancia del origen al centro de masa del eslabón 2
i1 = 1; % Momento de inercia del eslabón 1 alrededor de su centroide
i2 = 1; % Momento de inercia del eslabón 2 alrededor de su centroide
g=9.8; % Aceleración de la gravedad
Tau=0; % Torque aplicado al eslabón 1
%% Definicion de variables de factorizacion
t1 = m1*((lc1)^2)+m2*((l1)^2)+i1;
t2 = m2*((lc2)^2)+i2;
t3 = m2*l1*lc2;
t4 = m1*lc1+m2*l1;
t5 = m2*lc2;
%% Definicion de matriz A
a21 = (g*t4 + g*t5)/(t1 + t2 + 2*t3) - (g*t5*(t2 + 2*t3))/(t2*(t1 + t2 + 2*t3));
a22 = ((t2 + t3)*(t2 + 2*t3))/(t2*(t1 + t2 + 2*t3));
a23 = (g*t5)/(t1 + t2 + 2*t3) - (g*t5*(t2 + 2*t3))/(t2*(t1 + t2 + 2*t3));
a41 = (g*t5)/t2;
a42 = -(t2 + t3)/t2;
a43 = (g*t5)/t2;
A=[0 1 0 0; a21 a22 a23 0; 0 0 0 1; a41 a42 a43 0];
%% Definición de la matriz B
b21 = (t2)/(t1*t2+((t2)^2)+2*t2*t3);
B = [0; b21; 0; 0];
%% Definicion de funcion de ganancia K por metodo de Ackermann
K= acker(A,B,[-1 -1 -1 -1])
%% Definición del producto B*K
Bb=B*K;
% Definición de la matriz de coeficientes del sistema de control
Aa = A-Bb;
%% definición de ecuaciones
dx(1)= (Aa(1,1))*x(1)+(Aa(1,2))*x(2)+(Aa(1,3))*x(3)+(Aa(1,4))*x(4);
dx(2)=(Aa(2,1))*x(1)+(Aa(2,2))*x(2)+(Aa(2,3))*x(3)+(Aa(2,4))*x(4);
dx(3)=(Aa(3,1))*x(1)+(Aa(3,2))*x(2)+(Aa(3,3))*x(3)+(Aa(3,4))*x(4);
dx(4)=(Aa(4,1))*x(1)+(Aa(4,2))*x(2)+(Aa(4,3))*x(3)+(Aa(4,4))*x(4);
end
```

ANEXO 7.- PROGRAMA EN MATLAB PARA CONTROLAR EL SISTEMA DINÁMICO NO LINEAL POR RETROALIMENTACIÓN DE ESTADOS

```
function dx = Ecuacion_Control_Pendulo_Doble_nl_5(t,x,P)
% PROGRAMA PARA DEFINIR EL CONTROL DEL MODELO DINAMICO DEL PENDULO DOBLE NO LINEAL
% BASADO EN LAS GANANCIAS DEL SISTEMA LINEAL
% MODIFICANDO LAS VARIABLES DE ESTADO AL PUNTO y1 =x1-pi/2
% ABRAHAM MANILLA GARCIA (28 de Abril 2014)

%% definición de parámetros
dx = zeros(4,1); % vector columna
m1 = 0.25; % Masa del eslabón 1
m2 = 0.25; % Masa del eslabón 2
l1 = 0.3; % Longitud del eslabón 2
l2 = 0.3; % Longitud del eslabón 2
lc1 = 0.15; % Distancia del origen al centro de masa del eslabón 1
lc2 = 0.15; % Distancia del origen al centro de masa del eslabón 2
i1 = 1; % Momento de inercia del eslabón 1 alrededor de su centroide
i2 = 1; % Momento de inercia del eslabón 2 alrededor de su centroide
g=9.8; % Aceleración de la gravedad
%% Definición de variables de factorización
t1 = m1*((lc1)^2)+m2*((l1)^2)+i1;
t2 = m2*((lc2)^2)+i2;
t3 = m2*l1*lc2;
t4 = m1*lc1+m2*l1;
t5 = m2*lc2;
%% Definición de matriz A
a21 = (g*t4 + g*t5)/(t1 + t2 + 2*t3) - (g*t5*(t2 + 2*t3))/(t2*(t1 + t2 + 2*t3));
a22 = ((t2 + t3)*(t2 + 2*t3))/(t2*(t1 + t2 + 2*t3));
a23 = (g*t5)/(t1 + t2 + 2*t3) - (g*t5*(t2 + 2*t3))/(t2*(t1 + t2 + 2*t3));
a41 = (g*t5)/t2;
a42 = -(t2 + t3)/t2;
a43 = (g*t5)/t2;
A=[0 1 0 0; a21 a22 a23 0; 0 0 0 1; a41 a42 a43 0];
%% Definición de la matriz B
b21 = (t2)/(t1*t2+((t2)^2)+2*t2*t3);
B = [0; b21; 0; 0];
%% Definición de función de ganancia K por método de Ackermann
K= acker(A,B,P);
%% Definición de la función Tau (accion de control)
Tau = -K*[x(1); x(2); x(3); x(4)];
dx(1)=x(2);
dx(2)=((t2+2*t3*cos(x(3)))*(t3*sin(x(3))*x(2)^2+g*t5*cos((x(1)+pi/2)+x(3)))/(t2*(t1+t2+2*t3*cos(x(3))))-(g*t4*cos((x(1)+pi/2)+g*t5*cos((x(1)+pi/2)+ x(3))-2*t3*x(2)*x(4)*sin(x(3)))/(t1+t2 +2*t3*cos(x(3)))+(x(2)*(t2+t3*cos(x(3)))*(t2+2*t3*cos(x(3)))/(t2*(t1+t2+2*t3*cos(x(3))))+Tau/(t1+t2+2*t3*cos(x(3))));
dx(3)=x(4);
dx(4)=-(t3*sin(x(3))*x(2)^2+g*t5*cos((x(1)+pi/2)+x(3)))/t2-(x(2)*(t2 + t3*cos(x(3))))/t2;
end
```


ANEXO 8.- PROGRAMA MATLAB SINTONIZACIÓN POR MEDIO DE ALGORITMOS GENÉTICOS

```

clear all;
clc
%Parametros del AG
tic
N = 300; %Número de individuos
O = 4; %Numero de ganancias
NumEvo = 20;
Mut = 5; % Porcentaje de mutacion 7%
%Valores maximo y minimo de las ganancias
FACTOR = 1.0;
pmin(1) = -1; %minimo de K1
pmin(2) = 0; %minimo de K2
pmin(3) = -0.5; %minimo de K3
pmin(4) = 0; %minimo de K4
pmax(1) = 1; %maximo de K1
pmax(2) = 3; %maximo de K2
pmax(3) = 1; %maximo de K3
pmax(4) = 0.5; %maximo de K4
tfinal = 10;
h = 0.01;
t = 0:h:tfinal;
Xref=[3*pi/2 pi/2]
K=[-0.6099,1.8901,-0.0850,0.2612]
Xci=[3*pi/2+1 pi/2 0 0];
Yd = Pendubot_RE(Xci,Xref,K,h,0,tfinal)
%Inicializamos la poblacion de parametros.
P = rand(N,O);
%Escalamos resepecto al valor m·ximo y minimo.
for i = 1: O
    P(:,i) = (pmax(i) - pmin(i))*P(:,i) + pmin(i);
end
%Para por el número de evoluciones
%o para el un costo menor a 0.1158
contador = 0;
while(contador < NumEvo )
    %Determinamos la función costo
    for i = 1:N
        %Xci=[3*pi/2 pi/2 0 0];
        K = [P(i,1) P(i,2) P(i,3) P(i,4)];
        Y = Pendubot_RE(Xci,Xref,K,h,0,tfinal);
        Costo(i) = sum((Yd(:,1)-Y(:,1)).^2) + sum((Yd(:,2)-Y(:,2)).^2);
        plot(t,Yd(:,1),t,Y(:,1));
        axis([0 tfinal -2 10]);
        grid on
        title('COMPORTAMIENTO DINAMICO')
        xlabel('TIEMPO [s]')
        ylabel('DESPLAZAMIENTO ANGULAR [rad]')
        hold on;
    end
    figure;
    %Ordena de menor a mayor en funcion
    %del costo, en ind se guarda el indice
    [Costo_y, indx]=sort(Costo);
    %Se ordena la poblacion en funcion del indice
    P = P(indx,:);
    %Se selecciona la mitad de la poblacion con buen indice de desempeño.
    P = P(1:N/2,:);
    %Generar los indices para seleccionar
    %de forma aleatoria a las parejas.
    [F C]=size(P);
    vec = randperm(F);
    idx= reshape(vec,F/2,2);
    beta = rand(F/2);
    %Realizar el cruce.
    for i = 1:F/2
        P_Nueva = Cruce(idx(i,:),beta(i),P,pmin,pmax);
    end
    contador = contador + 1;
end

```

```
end
%Realizar la mutacion.
P_Nueva = Mutacion(P_Nueva, Mut, pmin, pmax);
P = [P; P_Nueva];
contador = contador + 1
end
    K = [P(1,1) P(1,2) P(1,3) P(1,4)]
    Y = Pendubot_RE(Xci, Xref, K, h, 0, tfinal);
    Costo(i) = sum((Yd(:,1)-Y(:,1)).^2) + sum((Yd(:,2)-Y(:,2)).^2);
    plot(t, Y(:,1), t, Y(:,2));
    title('COMPORTAMIENTO DINAMICO')
    xlabel('TIEMPO [s]')
    ylabel('DESPLAZAMIENTO ANGULAR [rad]')
    grid on
%Ordena de menor a mayor en funcion
%del costo, en ind se guarda el indice
[Costo_y, indx]=sort(Costo);
%Se ordena la poblacion en funcion del indice
P = P(indx,:);
    k1 = P(1,1)
    k2 = P(1,2)
    k3 = P(1,3)
    k4 = P(1,4)
toc
```

ANEXO 9.- PROGRAMA MATLAB SINTONIZACIÓN POR MEDIO DE ALGORITMO DE EVOLUCION DIFERENCIAL

```
% PROGRAMA PARA OBTENER EL OPTIMO DE LAS GANANCIAS DEL SISTEMA DINAMICO
% ***PENDULO DOBLE INVERTIDO POR EL METODO DE EVOLUCION DIFERENCIAL***
%*****AGREGANDO TASA DE MUTACION*****
% *****02 DE MAYO DE 2014*****
% *****ABRAHAM MANILLA GARCIA*****
function [ output_args ] = GANANCIAS_DE_1(NP, I, K)

% AGREGANDO CONTADOR DE TIEMPO
% EJEMPLO DE INTRODUCCION DE VALORES
%GANANCIAS_DE_1(300,[-1 1; 0 3; -0.5 1; 0 0.5], 20)
% DONDE:
%NP = NUMERO DE INDIVIDUOS
%I = VECTOR DE VALORES MAXIMOS Y MINIMOS DE BUSQUEDA DE SOLUCION
%K = NUMERO DE ITERACIONES
% PARAMETROS DE EVALUACION DE LA FUNCION PENDULO DOBLE
F =0.5; % TASA DE MUTACION
tfinal = 10;
h = 0.01;
t = 0:h:tfinal;
% PARAMETROS DESEADOS DE LA FUNCION PENDULO DOBLE
Xref=[3*pi/2 pi/2];
Kdes=[-0.6099,1.8901,-0.0850,0.2612]; % GANANCIAS DESEADAS
Xci=[3*pi/2+1 pi/2 0 0];
Yd = Pendubot_RE_1(Xci,Xref,Kdes,h,0,tfinal);
plot(t,180/pi*Yd(:,1),t,180/pi*Yd(:,2))
axis([0 tfinal -2*180/pi 10*180/pi]);
grid on
title('COMPORTAMIENTO DINAMICO');
xlabel('TIEMPO [s]');
ylabel('DESPLAZAMIENTO ANGULAR [°]');
% CREACION DE MATRIZ AUMENTADA DE CEROS CON LUGAR PARA LA EVALUACION DE LA
% POBLACION METANTE
Poblacion = zeros(NP,4*3+2);
%CREACION DE LA POBLACION ALEATORIA
Poblacion(:,1:4)=rand(NP,4);
%ESCALAMIENTO DE LA POBLACION CON RESPECTO A LOS VALORES DE BUSQUEDA
for i = 1:4
    Poblacion(:,i) = (I(i,2)-I(i,1))*Poblacion(:,i)+I(i,1);
end
%pause;
k = 1;
while k <=K %| error < epsilon
    %GENERANDO LA POBLACION DE GANANCIAS MUTANTES (Vog)
    for i =1:4
        for j = 1:NP
            index1 = randperm(NP,3);
            Poblacion(j,4+i) = F*(Poblacion(index1(1,1),i)-Poblacion(index1(1,2),i))...
                +Poblacion(index1(1,3),i);
        end
    end
    %CRUZANDO LA POBLACION ORIGINAL X0g CON LA POBLACION MUTANTE V0g
    for i =1:4
        for j = 1:NP
            beta = rand();
            Poblacion(j,4*2+i) = ...
                beta*Poblacion(j,i)+(1-beta)*Poblacion(j,4+i);
        end
    end
    % DEFINICION DE INDIVIDUOS DE LA POBLACION ORIGINAL Y POBLACION CRUZADA
    Kob = [Poblacion(:,1) Poblacion(:,2) Poblacion(:,3) Poblacion(:,4)];
    Kcr = [Poblacion(:,4*2+1) Poblacion(:,4*2+2) Poblacion(:,4*2+3) Poblacion(:,4*2+4)];
    % EVALUACION DE LOS INDIVIDUOS DE LA POBLACION GENERADA Y CRUZADA EN EL MODELO
    % DINAMICO PENDULO DOBLE
    for i=1:NP
        Kasoi = Kob(i,:);
        Yob = Pendubot_RE_1(Xci,Xref,Kasoi,h,0,tfinal);
```

```

Cob(i) = sum((Yd(:,1)-Yob(:,1)).^2) + sum((Yd(:,2)-Yob(:,2)).^2);
plot(t,180/pi*Yd(:,1),t,180/pi*Yob(:,1),t,180/pi*Yd(:,2),t,180/pi*Yob(:,2));
axis([0 tfinal -2*180/pi 10*180/pi]);
grid on;
title('COMPORTAMIENTO DINAMICO')
xlabel('TIEMPO [s]')
ylabel('DESPLAZAMIENTO ANGULAR [°]')
hold on;
Kasci = Kcr(i,:);
Ycr = Pendubot_RE_1(Xci,Xref,Kasci,h,0,tfinal);
Ccr(i) = sum((Yd(:,1)-Ycr(:,1)).^2) + sum((Yd(:,2)-Ycr(:,2)).^2);
plot(t,180/pi*Yd(:,1),t,180/pi*Ycr(:,1),t,180/pi*Yd(:,2),t,180/pi*Ycr(:,2));
axis([0 tfinal -2*180/pi 10*180/pi]);
grid on;
title('COMPORTAMIENTO DINAMICO')
xlabel('TIEMPO [s]')
ylabel('DESPLAZAMIENTO ANGULAR [°]')
end

% MATRICES DE COSTO
Cpo=[Cob]';
Cpc=[Ccr]';
% EVALUACION EL COMPORTAMIENTO DE LA POBLACION DE LAS GANANCIAS
for i = 1:NP
    if Cpo(i) > Cpc(i);
        Poblacion(i,1:4) = Poblacion (i,4*2+1:4*2+4);
    end
end
pause;
hold off;
k = k+1;
end
% DEFINICION DE LA NUEVA POBLACION
Poblacion;
% DEFINICION DE NUEVA POBLACION QUE ALIMENTA EL ALGORITMO
Popt =[Poblacion(:,1) Poblacion(:,2) Poblacion(:,3) Poblacion(:,4)];
% GRAFICANDO LA MEJOR GANANCIA CON RESPECTO A LA ULTIMA POBLACION GENERADA
for i=1:NP
    KaPti = Popt(i,:);
    YPt = Pendubot_RE_1(Xci,Xref,KaPti,h,0,tfinal);
    CPt(i) = sum((Yd(:,1)-YPt(:,1)).^2) + sum((Yd(:,2)-YPt(:,2)).^2);
end
CoPt=[CPt]'; % VECTOR COSTO DE LA POBLACION FINAL
% MATRIZ AUMENTADA DE POBLACION CON RESPECTO AL COSTO
McPt = [CoPt Popt(:,1) Popt(:,2) Popt(:,3) Popt(:,4)];
% ORDENANDO DE MAYOR A MENOR CON RESPECTO AL COSTO
VcCpt =sort(McPt(:,1),'ascend');
% LOCALIZANDO POSICION DEL COSTO MENOR
vectorCpt=find(McPt==VcCpt(1,1));
% OBTENIENDO EL VECTOR COMPLETO DE GANANCIAS CON MENOR ERROR TOTAL
KcPt = McPt(vectorCpt,:);
% DEFINIENDO LAS GANANCIAS MAS APROXIMADAS A LAS OPTIMAS
k1 = Kdes(1,1);
k2 = Kdes(1,2);
k3 = Kdes(1,3);
k4 = Kdes(1,4);
% DETERMINANDO EL ERROR CUADRATICO INDIVIDUAL Y TOTAL CON RESPECTO A LAS
% GANANCIAS OBTENIDAS CON EL ALGORITMO
for i=1:NP
    Kse1i = (k1-Popt(i,1))^2;
    Vkse1(i) = [Kse1i];
    Kse2i = (k2-Popt(i,2))^2;
    Vkse2(i) = [Kse2i];
    Kse3i = (k3-Popt(i,3))^2;
    Vkse3(i) = [Kse3i];
    Kse4i = (k4-Popt(i,4))^2;
    Vkse4(i) = [Kse4i];
    Vrr(i) = [Vkse1(i) + Vkse2(i) + Vkse3(i) + Vkse4(i)];
    Err(i) = [Vrr(i)];
end
% GENERACION DE MATRIZ AUMENTADA DE VALORES CON ERROR TOTAL

```

```

Vgner = [Err' Popt(:,1) Popt(:,2) Popt(:,3) Popt(:,4)];
% ORDENANDO DE MAYOR A MENOR CON RESPECTO AL ERROR
Vecord =sort(Vgner(:,1), 'ascend');
% LOCALIZANDO POSICION DEL ERROR TOTAL MENOR
vector=find(Vgner==Vecord(1,1));
% OBTENIENDO EL VECTOR COMPLETO DE GANANCIAS CON MENOR ERROR TOTAL
Ksel = Vgner(vector,:);
ErrorMenor = Ksel(1,1);
CostoMenor = KcPt(1,1);
% SELECCION DEL VECTOR GANANCIAS CON RESPECTO AL ERROR, CON RESPECTO AL
% COSTO Y EL DESEADO
VGsel = [Ksel(1,2) Ksel(1,3) Ksel(1,4) Ksel(1,5)];
VgCpt = [KcPt(1,2) KcPt(1,3) KcPt(1,4) KcPt(1,5)];
GananciasErrorMenor = [VGsel]
GananciasCostoMenor = [VgCpt]
GananciasDeseadas = [Kdes]
% GRAFICANDO EL COMPORTAMIENTO DEL MEJOR INDIVIDUO CON RESPECTO AL MENOR
% COSTO CON RESPECTO A LAS GANANCIAS OPTIMAS PARA UN TIEMPO DE EVALUACION
% DE 10 SEGUNDOS
tfinal1 = 10;
h1 = 0.01;
t1 = 0:h:tfinal1;
YCpt = Pendubot_RE_1(Xci,Xref,VgCpt(1,:),h1,0,tfinal1);
Yd1 = Pendubot_RE_1(Xci,Xref,Kdes,h1,0,tfinal1);
plot(t1,180/pi*Yd1(:,1),t1,180/pi*YCpt(:,1),t1,180/pi*Yd1(:,2),t1,180/pi*YCpt(:,2))
    axis([0 tfinal1 -2*180/pi 10*180/pi]);
    grid on;
    title('COMPORTAMIENTO DINAMICO')
    xlabel('TIEMPO [s]')
    ylabel('DESPLAZAMIENTO ANGULAR [°]')
    hold on;
end

```

ANEXO 10.- PROGRAMA MATLAB SINTONIZACIÓN POR MEDIO DE ALGORITMO DE OPTIMIZACION POR ENJAMBRE DE PARTICULAS

```
% PROGRAMA PARA OBTENER EL OPTIMO DE LAS GANANCIAS DEL SISTEMA DINAMICO
% ***PENDULO DOBLE INVERTIDO POR EL METODO OPTIMIZACION POR ENJAMBRE***
%*****DE PARTICULAS PSO*****
% *****08 DE MAYO DE 2014*****
% *****ABRAHAM MANILLA GARCIA*****
function [ output_args ] = GANANCIAS_PSO_1(NP, I, K)
% EJEMPLO DE INTRODUCCION DE VALORES
%GANANCIAS_PSO_1(300,[-1 1; 0 3; -0.5 1; 0 0.5],20)
% DONDE:
%NP = NUMERO PARTICULAS
%I = VECTOR DE VALORES MAXIMOS Y MINIMOS DE BUSQUEDA DE SOLUCION
%K = NUMERO DE ITERACIONES
Tic
% NUMERO INICIAL DE PARTICULAS
popsize = NP;
% TAMAÑO DEL ENJAMBRE
npar = 2;
c1 = 4;
c2 = 4-c1;
%LIMITES PARA LA BUSQUEDA DE CADA PARAMETRO DE LAS PARTICULAS
MinRs=I(1,1);      MaxRs=I(1,2);
MinRr=I(2,1);      MaxRr=I(2,2);
MinXs=I(3,1);      MaxXs=I(3,2);
MinXm=I(4,1);      MaxXm=I(4,2);
% PARAMETROS DE EVALUACION DE LA FUNCION PENDULO DOBLE
tfinal = 3;
h = 0.01;
t = 0:h:tfinal;
% PARAMETROS DESEADOS DE LA FUNCION PENDULO DOBLE
Xref=[3*pi/2 pi/2];
Kdes=[-0.6099,1.8901,-0.0850,0.2612]; % GANANCIAS DESEADAS
Xci=[3*pi/2+1 pi/2 0 0];
Yd = Pendubot_RE_1(Xci,Xref,Kdes,h,0,tfinal);
plot(t,180/pi*Yd(:,1),t,180/pi*Yd(:,2));
axis([0 tfinal -2*180/pi 10*180/pi]);
grid on
title('COMPORTAMIENTO DINAMICO')
xlabel('TIEMPO [s]')
ylabel('DESPLAZAMIENTO ANGULAR [°]')
% DEFINICION DE LAS PARTICULAS
par(:,1)=3*((MaxRs-MinRs)*rand(popsize,1)+MinRs);
par(:,2)=3*((MaxRr-MinRr)*rand(popsize,1)+MinRr);
par(:,3)=3*((MaxXs-MinXs)*rand(popsize,1)+MinXs);
par(:,4)=3*((MaxXm-MinXm)*rand(popsize,1)+MinXm);
% DEFINICION DE MATRIZ DE PARTICULAS 'GANANCIAS'
Kini = [par];
%DEFINICION DE MATRIZ DE VELOCIDADES INICIALES
Vini =zeros(NP,4);
% EVALUACION DE LA PRIMER POBLACION DE PARTICULAS EN EL MODELO DINAMICO
for i=1:NP
    Kasoi = Kini(i,:);
    Yob = Pendubot_RE_1(Xci,Xref,Kasoi,h,0,tfinal);
    Cini(i) = sum((Yd(:,1)-Yob(:,1)).^2) + sum((Yd(:,2)-Yob(:,2)).^2);
    plot(t,180/pi*Yd(:,1),t,180/pi*Yob(:,1),t,180/pi*Yd(:,2),t,180/pi*Yob(:,2));
    axis([0 tfinal -2*180/pi 10*180/pi]);
    grid on;
    title('COMPORTAMIENTO DINAMICO')
    xlabel('TIEMPO [s]')
    ylabel('DESPLAZAMIENTO ANGULAR [°]')
end
% MATRIZ DE COSTO INICIAL
MCini = [Cini]';
% ORDENANDO DE MENOR A MENOR EL COSTO DE LA PRIMER POBLACION PARA OBTENER
% EL MEJOR ELEMENTO GLOBAL
[vec iP] =sort(MCini,'ascend');
%OBTENIENDO EL MEJOR INDICE MINIMO CON RESPECTO AL COSTO GLOBAL
```

```

g=iP(1);
%MATRIZ AUMENTADA COSTO GANANCIAS
% BUSQUEDA DEL MEJOR LOCAL
P = zeros(NP,4); % GENERACION DE MATRIZ DE MINIMOS LOCALES
n=NP-2;
for i=1:n
    X1 = MCini(i); %PRIMER ELEMENTO DE COMPARACION
    X2 = MCini(i+1); %SEGUNDO ELEMENTO DE COMPARACION
    X3 = MCini(i+2); %STERCER ELEMENTO DE COMPARACION
    if (X1 < X2 < X3);
        P(i,:) = Kini(i,:);
    end
    if (X2 < X1 < X3);
        P(i,:) = Kini(i+1,:);
    end
    if (X3 < X1 < X2);
        P(i,:) = Kini(i+2,:);
    end
    if (X1 < X3 < X2);
        P(i,:) = Kini(i,:);
    end
    if (X2 < X3 < X1);
        P(i,:) = Kini(i+1,:);
    end
    if (X3 < X2 < X1);
        P(i,:) = Kini(i+2,:);
    end
    if (X1==X2==X3<1e2);
        P(i,:) = Kini(i,:);
    end
end
% ANALISIS DEL PENULTIMO ELEMENTO
X1 = MCini(NP-1); %PRIMER ELEMENTO DE COMPARACION
X2 = MCini(NP); %SEGUNDO ELEMENTO DE COMPARACION
X3 = MCini(NP-2); %STERCER ELEMENTO DE COMPARACION

    if (X1 < X2 < X3);
        P(NP-1,:) = Kini(NP-1,:);
    end
    if (X2 < X1 < X3);
        P(NP-1,:) = Kini(NP,:);
    end
    if (X3 < X1 < X2);
        P(NP-1,:) = Kini(NP-2,:);
    end
    if (X1 < X3 < X2);
        P(NP-1,:) = Kini(NP-1,:);
    end
    if (X2 < X3 < X1);
        P(NP-1,:) = Kini(NP,:);
    end
    if (X3 < X2 < X1);
        P(NP-1,:) = Kini(NP-2,:);
    end
    if (X1 == X2 == X3<1e2);
        P(NP-1,:) = Kini(NP-1,:);
    end
end

% ANALISIS DEL ULTIMO ELEMENTO
X1 = MCini(NP); %PRIMER ELEMENTO DE COMPARACION
X2 = MCini(NP-1); %SEGUNDO ELEMENTO DE COMPARACION
X3 = MCini(1); %STERCER ELEMENTO DE COMPARACION
    if (X1 < X2 < X3);
        P(NP,:) = Kini(NP,:);
    end
    if (X2 < X1 < X3);
        P(NP,:) = Kini(NP-1,:);
    end
    if (X3 < X1 < X2);
        P(NP,:) = Kini(1,:);
    end
end

```

```

    if (X1 < X3 < X2);
        P(NP,:) = Kini(NP,:);
    end
    if (X2 < X3 < X1);
        P(NP,:) = Kini(NP-1,:);
    end
    if (X3 < X2 < X1);
        P(NP,:) = Kini(1,:);
    end
    if (X1 == X2 == X3 < 1e2);
        P(NP,:) = Kini(NP,:);
    end
    Pl = [P]; % DEFINICION DE LA MATRIZ DE MINIMOS LOCALES
    % DEFINICION DE VARIABLES ALEATORIAS
    R1=rand(1);%variables aleatorias uniformes positivos[0 1]
    R2=rand(1);%pag 28 PSO/ tambien pueden ser aleatorio
    evcon=1;
    while (evcon<=K);
        [evcon MCini(g)];
        cont=1;
        while(cont<=NP)
            vn(cont,1)=Vini(cont,1)+c1*R1*(Pl(cont,1)-Kini(cont,1))+c2*R2*(Kini(g,1)-Kini(cont,1));
            vn(cont,2)=Vini(cont,2)+c1*R1*(Pl(cont,2)-Kini(cont,2))+c2*R2*(Kini(g,2)-Kini(cont,2));
            vn(cont,3)=Vini(cont,3)+c1*R1*(Pl(cont,3)-Kini(cont,3))+c2*R2*(Kini(g,3)-Kini(cont,3));
            vn(cont,4)=Vini(cont,4)+c1*R1*(Pl(cont,4)-Kini(cont,4))+c2*R2*(Kini(g,4)-Kini(cont,4));
            %OBTENCION DE VALORES DE LAS SIGUIENTES GANANCIAS
            K1=Kini(cont,1)+vn(cont,1);%actualizacion de S
            K2=Kini(cont,2)+vn(cont,2);%actualizacion de S
            K3=Kini(cont,3)+vn(cont,3);%actualizacion de S
            K4=Kini(cont,4)+vn(cont,4);%actualizacion de S
            Keva = [K1 K2 K3 K4];

            Cc=zeros(NP,1); %MATRIZ DE CEROS PARA COLOCAR COSTOS POR CADA CICLO
            Yan = Pendubot_RE_1(Xci,Xref,Keva,h,0,tfinal);
            Cob(cont) = sum((Yd(:,1)-Yan(:,1)).^2) + sum((Yd(:,2)-Yan(:,2)).^2);
            plot(t,180/pi*Yd(:,1),t,180/pi*Yan(:,1),t,180/pi*Yd(:,2),t,180/pi*Yan(:,2));
            axis([0 tfinal -2*180/pi 10*180/pi]);
            grid on;
            title('COMPORTAMIENTO DINAMICO')
            xlabel('TIEMPO [s]')
            ylabel('DESPLAZAMIENTO ANGULAR [°]')
            hold on;
            % CRITERIO DE SELECCION DE PARTICULAS
            if (Cob(cont)<MCini(cont));%MINIMO
                Kini(cont,:)=Keva;%actualizacion de Kini
                Con(cont)=Cob(cont);
            else
                Kini(cont,:)=Kini(cont,:);
                Con(cont)=MCini(cont);
            end
            Cc(cont) =Con(cont);
            S=[Kini];
            MCini =[Cc]';
            cont=cont+1;
            R1=rand(1);%variables aleatorias uniformes positivos[0 1]
            R2=rand(1);%pag 28 PSO/ tambien pueden ser aleatorios
        end

        % ACTUALIZACION DE VALORES DEL CICLO
        Vini=vn; %actualizacion de velocidades
        Kini = S;
        [vec iP]=sort(Kini);
        g=iP(1);%redefinicion del indice g MINIMO
        %% actualizacion de minimos locales
        P = zeros(NP,4); % GENERACION DE MATRIZ DE MINIMOS LOCALES
    n=NP-2;
    for i=1:n

```



```

X1 = MCini(i); %PRIMER ELEMENTO DE COMPARACION
X2 = MCini(i+1); %SEGUNDO ELEMENTO DE COMPARACION
X3 = MCini(i+2); %STERCER ELEMENTO DE COMPARACION
if (X1 < X2 < X3);
    P(i,:) = Kini(i,:);
end
if (X2 < X1 < X3);
    P(i,:) = Kini(i+1,:);
end
if (X3 < X1 < X2);
    P(i,:) = Kini(i+2,:);
end
if (X1 < X3 < X2);
    P(i,:) = Kini(i,:);
end
if (X2 < X3 < X1);
    P(i,:) = Kini(i+1,:);
end
if (X3 < X2 < X1);
    P(i,:) = Kini(i+2,:);
end
if (X1==X2==X3<1e2);
    P(i,:) = Kini(i,:);
end
end
% ANALISIS DEL PENULTIMO ELEMENTO
X1 = MCini(NP-1); %PRIMER ELEMENTO DE COMPARACION
X2 = MCini(NP); %SEGUNDO ELEMENTO DE COMPARACION
X3 = MCini(NP-2); %STERCER ELEMENTO DE COMPARACION
if (X1 < X2 < X3);
    P(NP-1,:) = Kini(NP-1,:);
end
if (X2 < X1 < X3);
    P(NP-1,:) = Kini(NP,:);
end
if (X3 < X1 < X2);
    P(NP-1,:) = Kini(NP-2,:);
end
if (X1 < X3 < X2);
    P(NP-1,:) = Kini(NP-1,:);
end
if (X2 < X3 < X1);
    P(NP-1,:) = Kini(NP,:);
end
if (X3 < X2 < X1);
    P(NP-1,:) = Kini(NP-2,:);
end
if (X1 == X2 == X3<1e2);
    P(NP-1,:) = Kini(NP-1,:);
end
end
% ANALISIS DEL ULTIMO ELEMENTO
X1 = MCini(NP); %PRIMER ELEMENTO DE COMPARACION
X2 = MCini(NP-1); %SEGUNDO ELEMENTO DE COMPARACION
X3 = MCini(1); %STERCER ELEMENTO DE COMPARACION
if (X1 < X2 < X3);
    P(NP,:) = Kini(NP,:);
end
if (X2 < X1 < X3);
    P(NP,:) = Kini(NP-1,:);
end
if (X3 < X1 < X2);
    P(NP,:) = Kini(1,:);
end
if (X1 < X3 < X2);
    P(NP,:) = Kini(NP,:);
end
if (X2 < X3 < X1);
    P(NP,:) = Kini(NP-1,:);
end
if (X3 < X2 < X1);
    P(NP,:) = Kini(1,:);
end

```

```

end
if (X1 == X2 == X3 < 1e2);
    P(NP,:) = Kini(NP,:);
end
Pl = [P]; % ACTUALIZACION DE MINIMOS LOCALES
%% fin de actualizacion de minimos locales
evcon = evcon + 1;
hold off
%pause
end
Poblacion = Kini; %POBLACION ULTIMA
% DEFINICION DE NUEVA POBLACION QUE ALIMENTA EL ALGORITMO
Popt = [Poblacion(:,1) Poblacion(:,2) Poblacion(:,3) Poblacion(:,4)];
% GRAFICANDO LA MEJOR GANANCIA CON RESPECTO A LA ULTIMA POBLACION GENERADO
for i = 1:NP
    KaPti = Popt(i,:);
    YPt = Pendubot_RE_1(Xci, Xref, KaPti, h, 0, tfinal);
    CPt(i) = sum((Yd(:,1) - YPt(:,1)).^2) + sum((Yd(:,2) - YPt(:,2)).^2);
end
CoPt = [CPt]'; % VECTOR COSTO DE LA POBLACION FINAL
% MATRIZ AUMENTADA DE POBLACION CON RESPECTO AL COSTO
McPt = [CoPt Popt(:,1) Popt(:,2) Popt(:,3) Popt(:,4)];
% ORDENANDO DE MAYOR A MENOR CON RESPECTO AL COSTO
VcCpt = sort(McPt(:,1), 'ascend');
% LOCALIZANDO POSICION DEL COSTO MENOR
vectorCpt = find(McPt == VcCpt(1,1));
% OBTENIENDO EL VECTOR COMPLETO DE GANANCIAS CON MENOR ERROR TOTAL
KcPt = McPt(vectorCpt,:);
% DEFINIENDO LAS GANANCIAS MAS APROXIMADAS A LAS OPTIMAS
k1 = Kdes(1,1);
k2 = Kdes(1,2);
k3 = Kdes(1,3);
k4 = Kdes(1,4);
% DETERMINANDO EL ERROR CUADRATICO INDIVIDUAL Y TOTAL CON RESPECTO A LAS
% GANANCIAS OBTENIDAS CON EL ALGORITMO
for i = 1:NP
    Kse1i = (k1 - Popt(i,1))^2;
    Vkse1(i) = [Kse1i];
    Kse2i = (k2 - Popt(i,2))^2;
    Vkse2(i) = [Kse2i];
    Kse3i = (k3 - Popt(i,3))^2;
    Vkse3(i) = [Kse3i];
    Kse4i = (k4 - Popt(i,4))^2;
    Vkse4(i) = [Kse4i];
    Vrr(i) = [Vkse1(i) + Vkse2(i) + Vkse3(i) + Vkse4(i)];
    Err(i) = [Vrr(i)];
end
% GENERACION DE MATRIZ AUMENTADA DE VALORES CON ERROR TOTAL
Vgner = [Err' Popt(:,1) Popt(:,2) Popt(:,3) Popt(:,4)];
% ORDENANDO DE MAYOR A MENOR CON RESPECTO AL ERROR
Vecord = sort(Vgner(:,1), 'ascend');
% LOCALIZANDO POSICION DEL ERROR TOTAL MENOR
vector = find(Vgner == Vecord(1,1));
% OBTENIENDO EL VECTOR COMPLETO DE GANANCIAS CON MENOR ERROR TOTAL
Ksel = Vgner(vector,:);
ErrorMenor = Ksel(1,1);
CostoMenor = KcPt(1,1);
% SELECCION DEL VECTOR GANANCIAS CON RESPECTO AL ERROR, CON RESPECTO AL
% COSTO Y EL DESEADO
VGsel = [Ksel(1,2) Ksel(1,3) Ksel(1,4) Ksel(1,5)];
VgCpt = [KcPt(1,2) KcPt(1,3) KcPt(1,4) KcPt(1,5)];
GananciasErrorMenor = [VGsel]
GananciasCostoMenor = [VgCpt]
GananciasDeseadas = [Kdes]
% GRAFICANDO EL COMPORTAMIENTO DEL MEJOR INDIVIDUO CON RESPECTO AL MENOR
% COSTO CON RESPECTO A LAS GANANCIAS OPTIMAS Y EL COMPORTAMIENTO DEL SISTEA REAL PARA UN
% TIEMPO MAYOR
tfinal1 = 10;
h1 = 0.01;
t1 = 0:h1:tfinal1;

```

```
YCpT = Pendubot_RE_1(Xci,Xref,VgCpT(1,:),h1,0,tfinal1);
Yd1 = Pendubot_RE_1(Xci,Xref,Kdes,h1,0,tfinal1);
plot(t1,180/pi*Yd1(:,1),t1,180/pi*YCpT(:,1),t1,180/pi*Yd1(:,2),t1,180/pi*YCpT(:,2))
axis([0 tfinal1 -2*180/pi 10*180/pi]);
grid on;
title('COMPORTAMIENTO DINAMICO')
xlabel('TIEMPO [s]')
ylabel('DESPLAZAMIENTO ANGULAR [°]')
hold on;
toc
end
```