

IDENTIFICACIÓN DE SISTEMAS USANDO PROGRAMACIÓN GENÉTICA

TESIS

Que para obtener el grado de
MAESTRÍA EN CIENCIAS DE LA INGENIERÍA ELÉCTRICA

presenta

Mario Graff Guerrero

Juan José Flores Romero

Director de Tesis

Universidad Michoacana de San Nicolás de Hidalgo

Agosto 2005

Resumen

La identificación de sistemas es el proceso de obtener un modelo matemático a partir de las observaciones del sistema. Este proceso es de vital importancia para una gran variedad de problemas de ingeniería. Por ejemplo, para controlar un sistema se requiere primero tener el modelo de éste.

La computación evolutiva, como Programación Genética y Gene Expression Programming, es una herramienta basada en el principio de Darwin de la evolución; esto es, la supervivencia del más apto. Tanto Programación Genética como Gene Expression Programming son herramientas que permiten la creación de programas o funciones siguiendo un criterio de optimización.

En este trabajo se desarrolla un programa llamado “Evolutionary Computation based System Identification” (ECSID). ECSID crea un modelo matemático a partir de las observaciones del sistema, un conjunto de terminales y un conjunto de funciones.

ECSID crea modelos en forma de sistemas de ecuaciones diferenciales ordinarias, lineales o no lineales. Además tiene la capacidad de crear modelos usando los valores pasados de la serie de tiempo, lo cual da la posibilidad de crear modelos tipo ARIMA (Autoregressive Integrated Moving Average).

ECSID se ha probado con el péndulo lineal, péndulo no lineal, masa resorte con amortiguador, circuito eléctrico, ecuación de Van der Pol, serie de tiempo de velocidad del viento, etc. ECSID ha logrado obtener modelos que se comportan de manera similar al modelo real.

Abstract

System identification is the process of obtaining a mathematical model from observations of the behavior of a system. This process is important in a wide variety of engineering problems. For example, in order to make a controller, it is needed to know the model of the system.

Gene Expression Programming and Genetic Programming are evolutionary tools inspired in the Darwinian principle of natural selection and survival of the fittest individuals. Both tools evolve programs or expressions.

In this work, we present a program called “Evolutionary Computation based System Identification” (ECSID). ECSID builds a mathematical model using the input-output of the system, a terminal set, and a function set.

ECSID can produce systems of ordinary differential equations, linear or non-linear. Besides, ECSID has the ability to evolve models that use previous values of the time series, this gives the possibility to produce ARIMA (Autoregressive Integrated Moving Average) models.

ECSID has been proved with the following examples: linear pendulum, non-linear pendulum, coupled mass-spring, electrical circuit, Van der Pol Equation, wind speed, etc. ECSID has obtained models that exhibit behaviors similar to the real models.

Contenido

Resumen	III
Abstract	V
Contenido	VI
Lista de Figuras	IX
Lista de Tablas	XI
Lista de Algoritmos	XIII
Lista de Símbolos	XV
Lista de Publicaciones	XVII
1. Introducción	1
1.1. Objetivo	3
1.2. Antecedentes	3
1.3. Descripción de la Tesis	5
2. Computación Evolutiva	7
2.1. Algoritmos Genéticos	10
2.2. Programación Genética	12
2.3. Gene Expression Programming	17
3. ECSID	25
4. Resultados	33
4.1. Regresión Simbólica	34
4.1.1. Seno	34
4.1.2. Parábola	36
4.2. Sistemas Dinámicos	37
4.2.1. Masa Resorte con Amortiguador	37
4.2.2. Masas y Resortes Acoplados	39
4.2.3. Péndulo	40
4.2.4. Circuito Eléctrico	44
4.2.5. Ecuaciones lineales con Coeficientes Variables	45
4.2.6. Ecuación de Van der Pol	48
4.3. Motor de Corriente Directa	49
4.4. Series de Tiempo	51

4.4.1. Manchas Solares	52
4.4.2. Velocidad del Viento	58
5. Conclusiones	67
5.1. Trabajo Futuro	68
A. Manual de Usuario de ECSID	69
A.1. Método de ventana	71
A.2. Método de EDOs	73
Referencias	75

Lista de Figuras

1.1. Sistema con salida y y entrada u	1
2.1. Cromosoma de Algoritmos Genéticos	11
2.2. Mutación en Algoritmos Genéticos	11
2.3. Apareamiento en Algoritmos Genéticos	12
2.4. Cromosoma de PG	12
2.5. Árbol balanceado	13
2.6. Árbol desbalanceado	13
2.7. Mutación	14
2.8. Apareamiento	15
2.9. Conversión de GEP a árbol.	17
2.10. Cromosoma multigen	18
2.11. Cromosoma multigen	19
3.1. EDO representada en ECSID.	27
4.1. Seno.	35
4.2. Seno $F = \{+, -, *, /\}$	35
4.3. Parábola	36
4.4. Masa Resorte con Amortiguador	37
4.5. Sistema masa resorte amortiguador	38
4.6. Sistema masa resorte con amortiguador y excitación externa	39
4.7. Diagrama de un sistema masa resortes acoplados	39
4.8. Sistema masa resortes acoplados	40
4.9. Diagrama de un péndulo	41
4.10. Péndulo lineal	42
4.11. Péndulo no lineal	43
4.12. Péndulo no lineal con fricción	43
4.13. Diagrama de un circuito eléctrico	44
4.14. Circuito eléctrico.	45
4.15. Ecuación (4.27).	46
4.16. Ecuación (4.30)	47
4.17. Ecuación (4.33)	48
4.18. Ecuación (4.36)	48

4.19. Ecuación de Van der Pol	49
4.20. Motor de corriente directa	51
4.21. Manchas solares sin errores de predicción (GEP)	53
4.22. Porcentaje de error de manchas solares Ec. (4.40).	54
4.23. Manchas solares con errores de predicción (PG)	55
4.24. Manchas solares sin errores de predicción (GEP)	55
4.25. Manchas solares sin errores de predicción (PG)	57
4.26. Porcentaje de error de manchas solares Ec. (4.43).	57
4.27. Viento (ARIMA)	59
4.28. Porcentaje de error de velocidad del viento Ec. (4.45).	60
4.29. Viento con errores de predicción (GEP)	60
4.30. Viento con errores de predicción (PG)	61
4.31. Viento sin errores de predicción (GEP)	62
4.32. Porcentaje de error de velocidad del viento Ec. (4.48).	63
4.33. Viento sin errores de predicción (PG)	64

Lista de Tablas

3.1. Parámetros de PG y GEP	30
4.1. Parámetros de los experimentos	34
4.2. Resultados de regresión simbólica	37
4.3. Resultados de sistemas dinámicos	50
4.4. Resultados de Manchas Solares	58
4.5. Resultados del Viento	64

Lista de Algoritmos

1.	Algoritmo Evolutivo General	8
2.	Selección proporcional a la función de aptitud	9
3.	Corta el último elemento de la población	10
4.	Operadores Genéticos de PG	16
5.	Convierte un gen a una estructura de árbol	20
6.	Operadores Genéticos de GEP	23
7.	Evaluación	28
8.	Evalúa modelo ARIMA	29
9.	Evalúa un modelo autoregresivo.	29
10.	Evalúa una EDO o un sistema de EDOs	30

Lista de Símbolos

AG	Algoritmos Genéticos
ARIMA	Autoregressive Integrated Moving Average
BFS	Breath First Search
ECSID	Evolutionary Computation based System Identification
EDO	Ecuación Diferencial Ordinaria
F	Conjunto de funciones
GEP	Gene Expression Programming
M	Número de individuos
NARMAX	Non-linear AutoRegressive Moving Average with eXogenous inputs
PG	Programación Genética
r	Coefficiente de correlación
T	Conjunto de Terminales
τ	Tamaño de la ventana

Lista de Publicaciones

■ Publicados

Juan J. Flores, Mario Graff and Erasmo Cadenas. “Wind Prediction Using Genetic Algorithms and Gene Expression Programming” Proceedings of the International Conference on Modeling and Simulation, AMSE 2005. Morelia, México. Abril 2005.

Juan J. Flores and Mario Graff. “Lessons Learned in Modeling Dynamic Systems using Genetic Programming” Proceedings of the Genetic and Evolutionary Computation Conference, GECCO 2005. Aceptado para presentación en la conferencia. No publicado por no poder asistir a la conferencia. Será publicado en una revista internacional del área.

Juan J. Flores and Mario Graff. “System Identification using Genetic Programming and Gene Expression Programming” Lecture Notes in Computer Science (LNCS), Springer-Verlag. Proceedings of the International Symposium on Computer and Information Sciences, ISCIS 2005. Estambul, Turquía. Octubre 2005

■ En preparación

Juan J. Flores and Mario Graff. “Improving a Regression Model Using a Function Set Reduction”

Juan J. Flores, Mario Graff and Francisco Madrigal Moreno. “Modelling a Company - Forecasting Behavior”. Publicación como capítulo del libro “Handbook of Research on Nature Inspired Computing for Economy and Managment”. Editor Jead Phillipe Renard. Francia. 2005

Capítulo 1

Introducción

La identificación de sistemas es el proceso de deducir un modelo matemático a partir de las observaciones de un sistema. Un sistema es un objeto en donde las variables de diferentes tipos interactúan y producen señales visibles. Las señales visibles son llamadas salidas del sistema. Un sistema puede ser manipulado por estímulos externos los cuales son llamados entradas del sistema (ver [Ljung99]). La figura 1.1 muestra un sistema f el cual es estimulado por una entrada u y se observa la salida y .

El modelado de un sistema es uno de los primeros pasos en varios problemas de ingeniería. Por ejemplo, para diseñar el control de un brazo de robot, un ingeniero hace algunas pruebas al sistema y observa su respuesta; a partir de estas respuestas, crea un modelo matemático y lo valida contra el modelo físico. Crear un buen modelo es una tarea compleja, dado que el modelo no solo debe de ser preciso, simple y debe capturar el comportamiento importante del sistema. Las habilidades necesarias para realizar un buen modelo, dada su complejidad, ha llevado a clasificar al modelado como “intuitivo” y “artístico” [Morrison91].

En la identificación de sistemas se tienen tres tipos diferentes de identificación. Estos dependen de la cantidad de conocimiento que se tiene del sistema. Los tres diferentes tipos de identificación son:

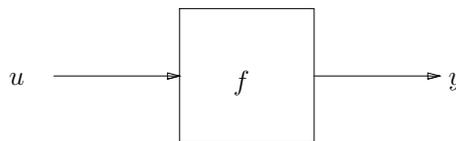


Figura 1.1: Sistema con salida y y entrada u .

Caja blanca. En este caso el modelo se conoce perfectamente y es posible construir el modelo a partir del conocimiento a priori y físico del sistema.

Caja gris. En este tipo de modelo se tiene algún conocimiento físico sobre el sistema, pero muchos parámetros todavía necesitan ser descubiertos. Estos se obtienen observando las entradas y salidas del sistema.

Caja negra. En este modelo no se tiene ningún conocimiento físico del sistema. La estructura del modelo se escoge de estructuras que han probado ser eficaces en el pasado.

Otra clasificación que se hace a la identificación de sistemas es paramétrica o estructural. En la identificación paramétrica se tiene todo el conocimiento de la estructura del modelo y lo único que desconoce son los parámetros de esta estructura. En la identificación estructural se desconoce tanto la estructura del modelo como los parámetros de que tiene la estructura.

Programación Genética (PG) [Koza92] y Gene Expression Programming (GEP) [Ferreira01] son herramientas evolutivas inspiradas en la teoría de la evolución de Darwin. Éstas permiten la evolución de programas o de entidades de estructura variable, las cuales pueden representar funciones. Este proceso simula la evolución en donde solo el conjunto de individuos más fuertes son seleccionados para reproducirse. El resultado de un programa genético es el mejor individuo en todas las generaciones.

En este trabajo se desarrolla un programa de computadora llamado “Evolutionary Computation based System Identification” (ECSID), el cual utiliza PG y GEP para obtener un modelo matemático a partir de las entradas del sistema, de un conjunto de terminales T y de un conjunto de funciones F .

ECSID es un identificador tipo caja negra; dado que el sistema no tiene ningún conocimiento físico del sistema. Lo único que observa son las entradas del sistema y construye un modelo a partir del conjunto de terminales T y del conjunto de funciones F .

El conjunto de terminales T representa las entradas y constantes del sistema, las cuales pueden ser las variables de estado de la ecuación diferencial o los valores pasados de la predicción. El conjunto de funciones F está formado por funciones aritméticas, trigonométricas, exponenciales, etc.

Los modelos obtenidos por ECSID pueden ser representados en forma de una ecuación diferencial ordinaria (EDO), un sistema de EDOs o como una modelo autoregresivo.

Los modelos producidos pueden ser lineales o no lineales.

1.1. Objetivo

El objetivo de este trabajo es desarrollar un programa (ECSID) que haga una identificación de sistemas tipo caja negra. El sistema utiliza herramientas evolutivas (PG o GEP) para la construcción del modelo y las entradas del sistema son: las observaciones del modelo, los estímulos que recibe el sistema, las variables de estado y un conjunto de funciones para construir el modelo.

Los objetivos particulares de este trabajo son:

- El programa debe de utilizar una EDO, un sistema de EDOs y/o modelos autoregresivos para representar el sistema.
- Debe de ser capaz de producir modelos lineales.

1.2. Antecedentes

Las herramientas evolutivas han sido utilizadas para la identificación de sistemas, por ejemplo los algoritmos genéticos fueron utilizados para hacer una identificación estructural (ver [Kristinsson92]). Tanto PG como GEP han sido utilizados para crear identificadores tipo caja negra. Estos identificadores no solo obtienen los parámetros del sistema sino también su estructura.

Existen dos tendencias en la forma del modelo que se obtiene utilizando PG y GEP en la identificación de sistemas. La primera es crear un modelo utilizando EDO y la segunda es creando modelos autoregresivos. Estas dos metodologías han sido eficaces en la generación de modelos que exhiben un comportamiento similar del sistema a modelado.

La forma general de un modelo que utiliza EDO se muestra en la Ecuación (1.1).

$$\begin{aligned}y_1' &= f(t, y_1, \dots, y_n) \\ &\vdots \\ y_n' &= f(t, y_1, \dots, y_n)\end{aligned}\tag{1.1}$$

Un modelo autoregresivo utiliza los τ valores pasados de la predicción para hacer una nueva predicción, τ representa el tamaño de la ventana. La forma general de un modelo autoregresivo se muestra en la Ecuación (1.2). Donde $y_{1,k-1}$ es el valor pasado de la predicción de la primera variable y $y_{n,k-1}$ representa el valor pasado de la predicción para la n -ésima variable.

$$y_k = f(y_{1,k-1}, \dots, y_{1,k-\tau}, y_{n,k-1}, \dots, y_{n,k-\tau}) \quad (1.2)$$

Gray et al. [Gray96a] usaron programación genética para hacer la identificación estructural y para la estimación de parámetros utilizaron un algoritmo de optimización. La metodología fue crear la estructura del modelo a partir de los bloques que tiene el paquete SIMULINK [Mat92] y la optimización de parámetros fue hecha mediante un método que combina simulated annealing y Nelder simplex [Press92]. Este procedimiento fue verificado utilizando modelos simulados.

Gray et al. [Gray96b] usaron programación genética para encontrar una expresión algebraica perteneciente a un sistema de EDOs y un método combinado de simulated annealing y Nelder simplex para hacer la identificación paramétrica. Este procedimiento fue utilizado para modelar un sistema de dos tanques conectados.

Weinbrenner [Weinbrenner97] usó programación genética para encontrar el modelo de la máquina de un helicóptero. Programación genética fue utilizada para encontrar la estructura del sistema y el método simulated annealing y Nelder simplex para encontrar los parámetros de la estructura. La estructura general de los modelos obtenidos se muestra en la Ecuación (1.3).

$$f(t, y, y', \dots, y^{(n)}) = 0 \quad (1.3)$$

Szpiro [Szpiro97] usó una variación de algoritmos genéticos para encontrar el modelo de una serie de tiempo. Esta variante usa un cromosoma variable. El cromosoma esta formado por variables, constantes, operadores y paréntesis. Szpiro utiliza los valores pasados de la predicción y los errores de predicción para hacer la nueva predicción. Este procedimiento fue probado con la serie de manchas solares del año 1900 a 1992.

Rodríguez-Vázquez et al. [Rodriguez-Vázquez98, Rodríguez-Vázquez04], Arkov et al. [Arkov00] y Evans et al. [Evans01] utilizaron programación genética para encontrar un sistema utilizando un modelo NARMAX (Non-linear AutoRegressive Moving Average with

eXogenous inputs) [Leontaritis85]. Este procedimiento fue utilizado en la identificación de una turbina de gas.

López et al. [López99] utilizaron programación genética para encontrar la estructura del sistema y algoritmos genéticos para encontrar sus parámetros. La estructura del sistema fue modelada mediante diagramas de bloques y los parámetros son las ganancias que tienen estos bloques. Este procedimiento fue validado con modelos simulados.

Cao et al. [Cao99b, Cao99a, Cao00a, Cao00b] usaron programación genética para encontrar la estructura del sistema y algoritmos genéticos para encontrar sus parámetros. La estructura fue modelada mediante un sistema de EDOs de la forma de la Ecuación (1.4). Este procedimiento fue validado con la serie de tiempo de la población de leopardos en Canadá de 1831 a 1944.

$$y^{(n)} = f(t, y, y', y'', \dots, y^{(n-1)}) \quad (1.4)$$

Babovic et al. [Babovic00] y Sakamoto [Sakamoto01] utilizaron programación genética para encontrar la estructura y los parámetros del sistema. Utilizaron EDO para representar el sistema. Estos procedimientos fueron validados con datos simulados.

Jäske [Jäske96], Angeline y Fogel [Angeline97], Angeline [Angeline98], Hinchliffe [Hinchliffe01], Varadan y Leung [Varadan02, Leung02], Chen [Chen01], Aguilar [Aguilar01], Rodríguez-Vázquez [Rodriguez-Vazquez01], Kaboudan [Kaboudan00], [Kaboudan03], Duan y Povinelly [Duan01], Hui [Hui03] y Winkler et al. [Winkler04] utilizaron los valores pasados de la predicción para crear el modelo del sistema.

Lie et al. [Jie04] usó gene expression programming [Ferreira01] para crear el modelo del sistema. Lie creó los modelos usando tanto EDOs como modelos autoregresivos.

1.3. Descripción de la Tesis

En este capítulo se dio una introducción a los diferentes tipos de identificación de sistemas, a las tecnologías utilizadas en la realización de ECSID y se mencionó el trabajo relacionado.

El capítulo 2 presenta una introducción a la computación evolutiva, específicamente a los algoritmos genéticos, programación genética y gene expression programming.

El capítulo 3 explica en pseudocódigo el funcionamiento de ECSID.

El capítulo 4 presenta los resultados obtenidos por ECSID. En ese capítulo se presentan experimentos con sistemas masa resorte, péndulo lineal y no lineal, circuito eléctrico, motor eléctrico, ecuación de Van der Pol, serie de tiempo de manchas solares y del viento.

El capítulo 5 presenta las conclusiones y las aportaciones del presente trabajo.

El apéndice A presenta el manual de usuario de ECSID. En este manual se explica la forma de obtener ECSID, así como la manera de especificar las observaciones del sistema, el conjunto de terminales T y el conjunto de funciones F .

Capítulo 2

Computación Evolutiva

La computación evolutiva está basada en el principio de Darwin de la selección natural. Es un método de optimización que aplica los mismos principios que la evolución biológica; los cuales son selección basada en la población, reproducción sexual y mutación.

En este capítulo se da una introducción a los siguiente algoritmos evolutivos: Algoritmos Genéticos AG (Sección 2.1), Programación Genética PG (Sección 2.2) y Gene Expression Programming GEP (Sección 2.3).

Los algoritmos evolutivos que se presentan en este trabajo comparten un algoritmo general, por lo cual, se propone el siguiente algoritmo 1 (*AlgoritmoEvolutivo*). Éste recibe el número de generaciones *generaciones*, el número de individuos de la población *númeroIndividuos*, el criterio de terminación *terminación* y el método *método*. Existen procedimientos que son particulares para cada método. Por ejemplo, la función *CreaPoblación* es diferente para AG, PG y GEP por lo cual *AlgoritmoEvolutivo* recibe una estructura con las diferentes funciones.

El procedimiento de *AlgoritmoEvolutivo* se presenta a continuación. En la línea 2 se crea la población inicial; esta función es particular para cada tipo de algoritmo evolutivo. La línea 3 evalúa a cada individuo de la población y ordena la población del mejor individuo al peor, de acuerdo a la función de aptitud *Evaluación*. *Evaluación* es también particular para cada algoritmo evolutivo. La línea 4 es el ciclo el cual termina cuando se cumple el número de generaciones o se cumple el criterio de terminación. Para todos los ejemplos supondremos que el mejor individuo es aquel que tiene una función de aptitud cero. En la línea 6 se seleccionan *Selección* (ver Algoritmo 2) aquellos individuos de la población a los cuales se les pueden aplicar los operadores genéticos *OperadoresGenéticos*. En la línea

7 se aplica la función de aptitud y se ordena la población. La línea 8 elimina los peores individuos para que el número de individuos dentro de la población se mantenga constante. La línea 9 regresa el mejor individuo encontrado.

Algoritmo 1 Algoritmo Evolutivo General

```

ALGORITMOEVOLUTIVO(generaciones, númeroIndividuos, terminación, método)
1   $i \leftarrow 1$ 
2   $población \leftarrow método[CREAPOBLACIÓN](númeroIndividuos)$ 
3   $población \leftarrow método[EVALUACIÓN](población)$ 
4  mientras  $i \leq generaciones$  y  $MEJOR(población) > terminación()$ 
5     $i \leftarrow i + 1$ 
6     $tmp \leftarrow método[OPERADORESGENÉTICOS](SELECCIÓN(población))$ 
7     $tmp \leftarrow método[EVALUACIÓN](tmp)$ 
8     $población \leftarrow CORTACOLA(población, tmp)$ 
9  regresar  $MEJOR(población)$ 

```

Existen varias formas de elegir a los individuos (ver [Holland92], [Thierens94], [Blickle95] y [Whitley89]); por ejemplo se pueden seleccionar solo los mejores individuos o se pueden seleccionar al azar n individuos y escoger los mejores de ellos o se puede usar la selección proporcional a la función de aptitud, donde todos tienen una probabilidad ser seleccionados.

El algoritmo 2 (*Selección*) muestra la selección proporcional a la función de aptitud la cual es utilizada en este trabajo, en la línea 6 se crea un número aleatorio de punto flotante de 0 a 1. El ciclo de la línea 9 a la 11 termina cuando el valor acumulado es mayor que el número aleatorio. En la línea 12 se añade el elemento escogido j a la lista. El operador $+$ es utilizado para añadir elementos al final de una lista. La línea 13 regresa la lista de individuos seleccionados.

Una vez aplicados los operadores genéticos a la población es necesario mantener constante el número de individuos de la población. Existen varias maneras para hacer esto; una es ordenar los nuevos individuos y los viejos de acuerdo a su aptitud y desechar los

Algoritmo 2 Selección proporcional a la función de aptitud

SELECCIÓN(*población*)

- 1 $sumaAptitud \leftarrow 0$
- 2 $lista \leftarrow []$
- 3 **para** i **en** *población*
- 4 $sumaAptitud \leftarrow sumaAptitud + \frac{1}{1+APTITUD(i)}$
- 5 **para** $i \in poblacion$
- 6 $sw \leftarrow \text{NÚMEROALEATORIO}(0, 1)$
- 7 $suma \leftarrow 0$
- 8 $j \leftarrow 1$
- 9 **mientras** $sw > suma$ **y** $j \leq \text{TAMAÑO}(población)$
- 10 $suma \leftarrow suma + \frac{\frac{1}{1+APTITUD(población[j])}}{sumaAptitud}$
- 11 $j \leftarrow j + 1$
- 12 $lista \leftarrow lista + población[j]$
- 13 **regresar** $lista$

elementos que se encuentren al final de la lista. Otra forma es ordenar a los nuevos individuos con el mejor individuo de la población anterior y eliminar el último individuo de la lista (ver [Ferreira01]). El algoritmo 3 (*CortaCola*) muestra este último procedimiento.

Algoritmo 3 Corta el último elemento de la población

```

CORTACOLA(población, nuevaPoblación)
1  lista ← []
2  lista ← lista + población[1] + nuevaPoblación
3  lista ← ORDENA(lista)
4  lista ← QUITAÚLTIMO(lista)
5  regresar lista

```

2.1. Algoritmos Genéticos

Los algoritmos genéticos son un método de optimización, los cuales tratan de encontrar una serie de parámetros (x_1, \dots, x_n) tales que maximicen la función $F(x_1, \dots, x_n)$. Para tal efecto, las variables (x_1, \dots, x_n) son codificadas en un cromosoma. Los cromosomas en los algoritmos genéticos tienen una longitud constante. Esto hace que los problemas tengan que ser codificados para poder encontrar su solución.

Para utilizar algoritmos genéticos, el problema tiene que ser codificado en un cromosoma de longitud constante. El cromosoma puede estar formado por un arreglo de números reales o por una representación binaria de este arreglo.

En identificación de sistemas, primero se tendría que conocer la estructura del sistema para poder aplicar algoritmos genéticos. En este caso, los algoritmos genéticos son utilizados para optimizar los parámetros de la estructura del sistema. Los algoritmos genéticos en identificación de sistemas solo pueden hacer una identificación paramétrica.

La Figura 2.1 muestra la ecuación $y_t = ay_{t-2} + by_{t-1} + c$ representada en un cromosoma. Se observa que lo que se codifica en el cromosoma solo son las variables a , b y c y en ningún momento se codifica la ecuación anterior. Este tipo de codificación hace que el algoritmo genético optimice los parámetros a , b y c para minimizar la función y_t .

$$y_t = ay_{t-2} + by_{t-1} + c$$

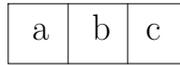
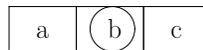


Figura 2.1: Cromosoma de Algoritmos Genéticos

La forma de operar de un algoritmo genético es creando aleatoriamente una población inicial. Los individuos de esta población son ordenados de acuerdo a una función de aptitud; los más aptos son seleccionados para aplicarles operadores genéticos (mutación, apareamiento y reproducción). Una vez aplicados los operadores genéticos, los nuevos individuos y los viejos son ordenados de acuerdo a su aptitud. Finalmente los individuos menos aptos son desechados y el proceso continúa hasta que se encuentra un individuo que satisface un criterio de terminación o se excede el número de generaciones.

La mutación es un operador genético que selecciona una subcadena del cromosoma y la cambia aleatoriamente por otra. La Figura 2.2 muestra este proceso, donde la subcadena b es cambiada por e . Este operador genético por lo general tiene poca probabilidad de ocurrir, pero es un factor importante, dado que proporciona variedad genética [Luke97].

Antes de mutación



Después de mutación

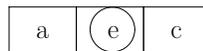


Figura 2.2: Mutación en Algoritmos Genéticos

El apareamiento es otro operador genético. Este es el más utilizado y se comporta de la siguiente manera; dos individuos de la población son procesados y aleatoriamente se selecciona una subcadena de cada individuo para después intercambiarlas. La Figura 2.3 muestra este proceso donde las subcadenas b y f del primer y segundo individuo respectivamente son seleccionadas e intercambiadas.

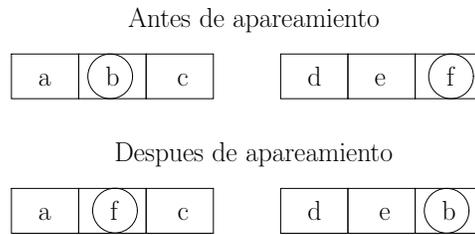


Figura 2.3: Apareamiento en Algoritmos Genéticos

La reproducción, otro operador genético, copia un individuo sin modificaciones a la nueva población. Note que el operador reproducción encuentra su analogía biológica más en la clonación que en el cruzamiento.

2.2. Programación Genética

La programación genética es una generalización de los algoritmos genéticos, donde los cromosomas son de tamaño variable, dando la posibilidad de encontrar funciones y no solo evolucionar los parámetros de las funciones. La Figura 2.4 muestra un cromosoma en PG. El cromosoma representa la ecuación completa y no solo sus parámetros; el cromosoma tiene una estructura de árbol, mientras que un cromosoma en algoritmos genéticos es una cadena.

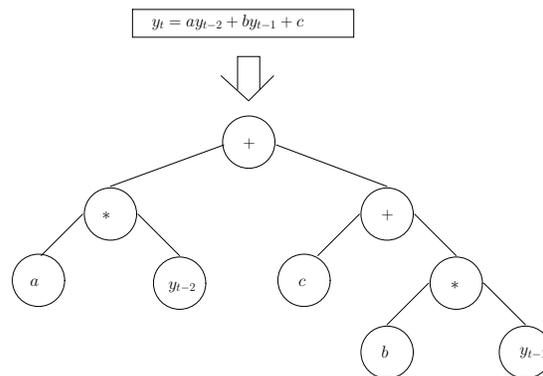


Figura 2.4: Cromosoma de PG

PG usa un conjunto de terminales T y un conjunto de funciones F . El conjunto de terminales T representan las variables del sistema, en el caso de identificación de sistemas puede ser las variables de estado o los valores pasados de la predicción. El conjunto de

funciones F son las operaciones que pueden ser usadas en el cromosoma. En la Figura 2.4 se observa como una estructura de árbol puede representar una ecuación.

Usando F y T se crea la población inicial. Para crearla, Koza [Koza92] propone dos métodos: el primero es crear árboles balanceados y el segundo es crear árboles desbalanceados. Koza también propone que la altura del árbol sea variable. La altura del árbol no puede ser mayor a un parámetro dado. Estas formas de crear la población inicial hace que exista diversidad morfológica en la población inicial.

La Figura 2.5 muestra un árbol creado con el método balanceado. El conjunto de funciones es $F = \{+, -, *, /\}$ y el conjunto de terminales es $T = \{x, \Re\}$. Donde \Re es un número flotante dentro de 0 al 100. Para crear un árbol balanceado se siguen los siguientes pasos: Se selecciona aleatoriamente un elemento del conjunto de funciones F siempre y cuando no se halla llegado a la altura del árbol. Cuando se llega a la altura del árbol se seleccionan un elemento del conjunto de terminales T .

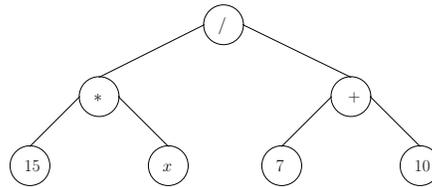


Figura 2.5: Árbol balanceado

En la Figura 2.6 se observa un árbol creado con el método desbalanceado. El procedimiento para crear un árbol desbalanceado es el siguiente: En la altura 0 se selecciona una función del conjunto F , mientras no llegue a la altura final se hace la unión del conjunto F y el conjunto T . Se selecciona un elemento de este conjunto. Si se selecciona una función se siguen añadiendo elementos a esa rama, de lo contrario se deja la rama como está. Si se llega a la altura máxima del árbol se seleccionan solo elementos de T .

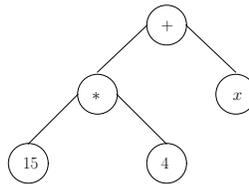


Figura 2.6: Árbol desbalanceado

Los operadores genéticos usados en PG son la mutación y el apareamiento. La mutación recibe un individuo y regresa el individuo modificado. El apareamiento recibe dos individuos de la población y crea dos nuevos individuos, destruyendo a los padres.

La mutación selecciona aleatoriamente un nodo y lo cambia aleatoriamente por otro que tenga las mismas características; las características se refieren al conjunto que pertenece el nodo y también a la cardinalidad de éste. Por ejemplo si el nodo contiene una terminal se cambia por otro elemento del conjunto de terminales T , pero si el nodo es una función se selecciona por otra función que tenga la misma cardinalidad.

La Figura 2.7 muestra un ejemplo de mutación. El individuo de la parte superior es el individuo seleccionado y se le denomina padre. El individuo de la parte inferior es el nuevo individuo y se le denomina hijo. El nodo seleccionado para mutarse se encuentra resaltado por un cuadro. Se observa que la función $+$ fue seleccionada y ésta es aleatoriamente intercambiada por otra del conjunto F que tenga la misma cardinalidad.

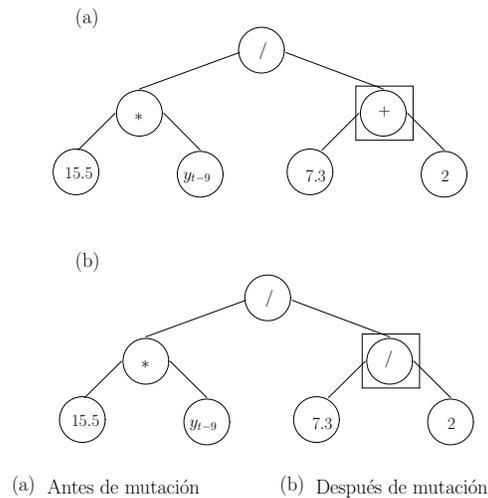


Figura 2.7: Mutación

El apareamiento recibe dos individuos de la población, selecciona aleatoriamente un subárbol de cada individuo e intercambiar estos subárboles. La Figura 2.8 muestra esta operación, los subárboles seleccionados se encuentran resaltados por un cuadro. Los individuos de la parte superior fueron seleccionados para aparearse y se les denomina padres. El resultado del apareamiento son los individuos que se encuentran en la parte inferior y se les denomina hijos. Se observa que el subárbol del primer hijo pertenecía al segundo padre y que el subárbol del segundo hijo pertenecía al primer padre.

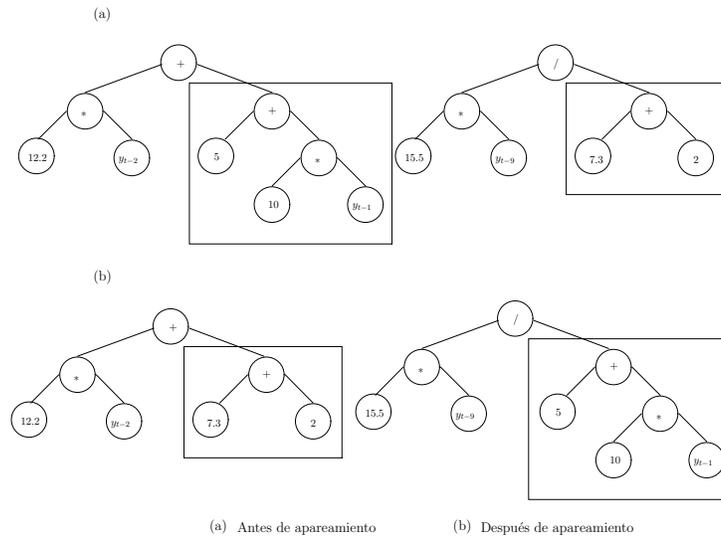


Figura 2.8: Apareamiento

El algoritmo 4 (*OperadoresGenéticosPG*) muestra la función que aplica los operadores genéticos en PG. Éste recibe los individuos seleccionados *población* e itera por todos los elementos de *población* (de la línea 2 a la 15). En la línea 3 se selecciona un número flotante del 0 al 1. De la línea 4 a la 7 es un caso especial que se ejecuta cuando ya no se tienen individuos para ser apareados, por lo cual sólo se puede hacer mutación. De la línea 8 a la 17 se aplican los operadores genéticos, p_m es la probabilidad de mutarse y p_a es la probabilidad de aparearse.

Este algoritmo tiene una operación genética no estándar que es *IntercambioGen*. Este operador solo funciona cuando los individuos de la población tienen una lista de cromosomas en lugar de tener sólo uno. *IntercambioGen* selecciona un cromosoma de cada individuo y los intercambia.

Durante el proceso evolutivo los individuos en la población tienden a crecer, lo cual hace que evaluarlos sea más costoso y que los operadores genéticos sean menos efectivos. Existen varias maneras para reducir este problema; una es crear una función que castigue al individuo por el número de nodos que contiene, otra forma es crear una función que elimine todos aquellos individuos que su tamaño sobrepase un cierto límite. En este trabajo se optó por la segunda opción y se eliminan aquellos individuos que están constituidos por más de 100 nodos.

Se han hecho estudios para determinar los valores de p_m y de p_a . Luke et al [Luke97]

Algoritmo 4 Operadores Genéticos de PG

OPERADORES GENÉTICOS PG(*población*)

```
1   $i \leftarrow 1$ 
2   $p_m \leftarrow 0.2$ 
3   $p_a \leftarrow 0.7$ 
4  mientras  $i \leq \text{TAMAÑO}(\textit{población})$ 
5       $sw \leftarrow \text{NÚMERO ALEATORIO}(0, 1)$ 
6      si  $i = \text{TAMAÑO}(\textit{población})$ 
7          entonces MUTACIÓN( $\textit{población}[i]$ )
8               $i \leftarrow i + 1$ 
9              continuar
10     si  $p_m > sw$ 
11         entonces MUTACIÓN( $\textit{población}[i]$ )
12              $i \leftarrow i + 1$ 
13             continuar
14     si  $p_a > sw$ 
15         entonces APAREAMIENTO( $\textit{población}[i], \textit{población}[i + 1]$ )
16              $i \leftarrow i + 2$ 
17             continuar
18     INTERCAMBIO GEN( $\textit{población}[i], \textit{población}[i + 1]$ )
19      $i \leftarrow i + 2$ 
```

hizo una serie de experimentos tratando de comparar la efectividad de la mutación y del apareamiento; los resultados que obtuvo no son concluyentes. Comenta que la mutación es más efectiva cuando se tienen poblaciones pequeñas y el apareamiento es más efectivo en lo contrario, pero no logró encontrar cual operador es más efectivo. Banzhaf et al [Banzhaf96] encontró que cambiar el balance entre la probabilidad de mutación y apareamiento puede cambiar significativamente el comportamiento de PG. En su trabajo encontró que una probabilidad alta de mutación puede mejorar considerablemente el desempeño de PG pero este caso solo se da para problemas considerados “difíciles”.

2.3. Gene Expression Programming

Gene Expression Programming (GEP) es similar a PG dado que puede crear programas o en el caso de identificación de sistemas funciones de estructura variable. GEP usa una codificación diferente a PG; usa una cadena que representan genes. Aunque el cromosoma en GEP está compuesto por una cadena, éste representa una estructura de árbol. La figura 2.9 muestra un ejemplo de un cromosoma de un gen y su representación en estructura de árbol. Se puede observar que el gen es una cadena formada por funciones y terminales; este gen representa un árbol, el cual se construye recorriendo en forma BFS el árbol representado por el gen (ver Algoritmo 5). El gen tiene longitud constante y se interpreta hasta donde se completa el árbol que representa. El resto de los caracteres en él no tienen significado alguno.

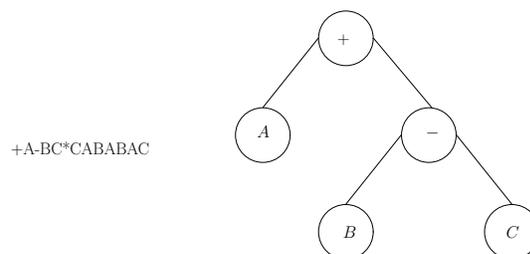


Figura 2.9: Conversión de GEP a árbol.

Los genes en GEP están compuestos por una cabeza H y una cola T . En la parte de la cabeza se pueden incluir funciones y terminales pero en la parte de la cola solo se pueden incluir terminales. El tamaño de la cabeza es puesta por el usuario. El tamaño de la cola $t(T)$ está en función de $t(H)$. La Ecuación (2.1) (ver [Ferreira01]) muestra la relación

que tiene el tamaño de $t(T)$ con el tamaño $t(H)$, donde n es la máxima cardinalidad que tienen las funciones de F . Por ejemplo si $F = \{+, -, /, *, \sqrt{\cdot}\}$ entonces $n = 2$.

$$t(T) = 1 + t(H) * (n - 1) \tag{2.1}$$

La Ecuación (2.2) muestra un cromosoma donde la cabeza es $+ * -/b$ y la cola es $aababb$. La cola es identificada con una barra horizontal encima de ésta.

$$+ * -/\overline{baababb} \tag{2.2}$$

En GEP cada cromosoma puede estar formado por múltiples genes; la Ecuación (2.3) muestra un cromosoma formado por dos genes. Este cromosoma representa dos árboles, la figura 2.10 muestra estos árboles. La cola de cada gen es identificada por una barra horizontal encima de ésta.

$$+ * -/\overline{baababb}/ + \overline{-abbaaabb} \tag{2.3}$$

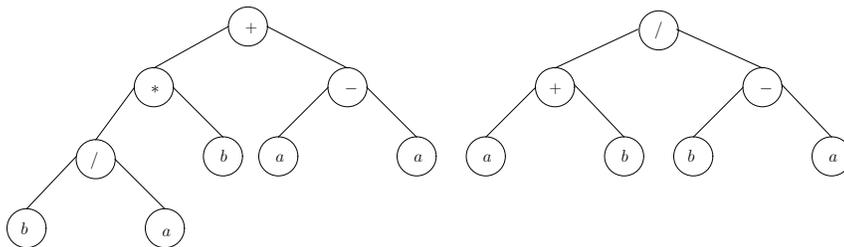


Figura 2.10: Cromosoma multigen

Cuando un cromosoma tiene múltiples genes se requiere de una función que una estos genes. Se puede usar una función para unir éstos, como por ejemplo la suma o se puede usar un nuevo cromosoma para que evolucione una función de unión. La Ecuación (2.4) muestra un individuo que tiene dos cromosomas el primero $(+ * -/\overline{baababb}/ + \overline{-abbaaabb})$ es multigen y el segundo $(+ * 0\overline{1001})$ es la función que une los genes. El conjunto T del segundo cromosoma es $\{0, 1\}$ donde 0 representa el primer gen del primer cromosoma y 1 representa el segundo gen del primer cromosoma. La figura 2.11 muestra este individuo.

$$+ * -/\overline{baababb}/ + \overline{-abbaaabb} + * 0\overline{1001} \tag{2.4}$$

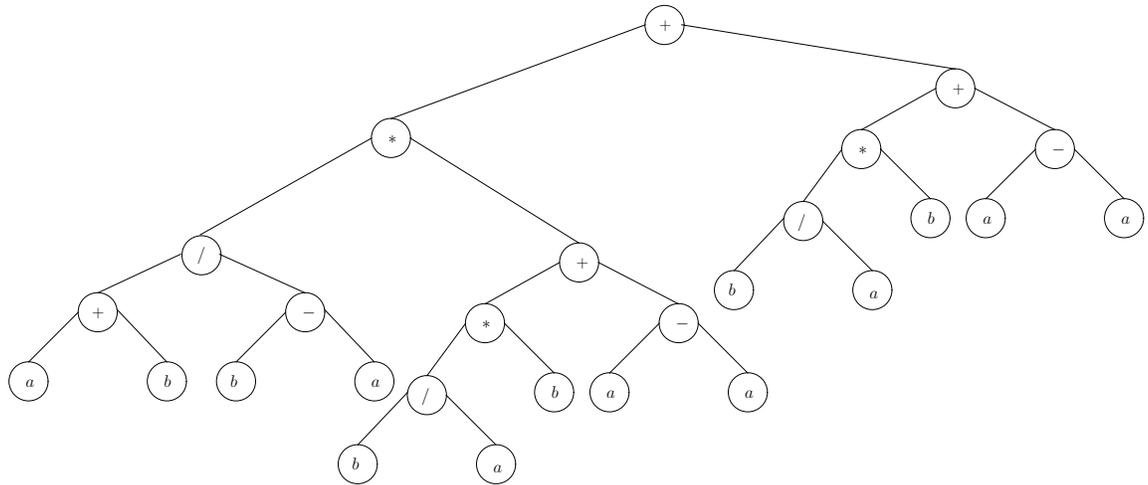


Figura 2.11: Cromosoma multigen

El algoritmo 5 (*GenÁrbol*) se muestra el algoritmo para convertir un gen a una estructura de árbol. En la línea 2 se crea un nodo, este nodo tiene la capacidad de aceptar tantos hijos como sean necesarios. En la línea 3 se añade este nuevo elemento a la lista *lista*, el operador + hace la concatenación de sus operandos. El ciclo de la línea 5 a la 13 se repite hasta que *lista* se encuentre vacía. En la línea 6 se toma el primer elemento de *lista*. La línea 7 elimina el primer elemento de *lista*. El ciclo de la línea 8 a 13 se hace tantas veces como el número de argumentos de la función *elemento*. En la línea 10 se añade el nodo que está en la variable *tmp* a *elemento*. Si el elemento que se acaba de añadir *tmp* es una función se añade a *lista* (líneas 12 y 13). La línea 14 regresa el árbol creado.

Las operaciones genéticas en GEP se aplican a los cromosomas, esto hace que todas las operaciones sean sencillas de implementar. Las operaciones genéticas usadas en GEP son: mutation, is-transposition, ris-transposition, gene-transposition, 1-point-recombination, 2-point-recombination, gene-recombination.

La mutación selecciona aleatoriamente un caracter del cromosoma y lo intercambia por otro. Si el caracter seleccionado está dentro de la cabeza lo puede cambiar por cualquier elemento de T o F , pero si se encuentra en la cola es cambiado por elementos que se encuentran en el conjunto T .

La Ecuación (2.5) muestra un ejemplo. El elemento para ser mutado se encuentra subrayado. Se observa que el carácter b se encuentra dentro de la cabeza del segundo gen y es intercambiado por un $+$ que es un elemento del conjunto de funciones F .

Algoritmo 5 Convierte un gen a una estructura de árbol

GENÁRBOL(*gen*)

- 1 $lista \leftarrow []$
- 2 $árbol = \text{CREANODO}(gen[0])$
- 3 $lista \leftarrow lista + árbol$
- 4 $c \leftarrow 1$
- 5 **mientras** TAMAÑO(*lista*) $\neq 0$
- 6 $elemento \leftarrow lista[0]$
- 7 ELIMINA(*lista*[0])
- 8 **para** $i \leftarrow 1$ **hasta** CARDINALIDAD(*elemento*)
- 9 $tmp \leftarrow \text{CREANODO}gen[c]$
- 10 AÑADEHIJO(*elemento*, *tmp*)
- 11 $c \leftarrow c + 1$
- 12 **si** $gen[c] \in F$
- 13 **entonces** $lista = lista + tmp$
- 14 **regresar** *árbol*

$$\begin{array}{c}
+ * -/\overline{baababb}/ + -\overline{abbaaabb} \\
\Downarrow \\
+ * -/\overline{baababb}/ + -\overline{a*baaabb}
\end{array} \tag{2.5}$$

Is-transposition (Insertion sequence transposition) selecciona una subcadena de cualquier parte del gen y lo inserta en cualquier parte de la cabeza excepto en el primer elemento. Ris-transposition (Root insertion sequence transposition) es similar al anterior pero la subcadena seleccionada es colocada en el primer elemento de la cabeza. Las Ecuaciones (2.6) y (2.7) muestran un ejemplo de estas operaciones.

En la Ecuación (2.6) se observa que la subcadena $-ab$ es seleccionada y es insertada en el primer gen. Se observa que los caracteres donde va a ser insertada la cadena son desplazados hacia la izquierda. Como el tamaño de la cabeza es fijo se desechan los últimos caracteres. En este ejemplo los caracteres $-/b$ son desechados.

$$\begin{array}{c}
+ * -/\overline{baababb}/ + -\overline{abbaaabb} \\
\Downarrow \\
+\underline{-ab} * \overline{aababb}/ + -\overline{abbaaabb}
\end{array} \tag{2.6}$$

$$\begin{array}{c}
+ * -/\overline{baababb}/ + -\overline{abbaaabb} \\
\Downarrow \\
\underline{-ab} + * \overline{aababb}/ + -\overline{abbaaabb}
\end{array} \tag{2.7}$$

Gene-transposition selecciona un gen del individuo y lo coloca como primer gen haciendo que todos se recorran un lugar. La Ecuación (2.8) muestra un ejemplo de esta operación. En este ejemplo el segundo gen es seleccionado y se coloca en el lugar del primer gen y todos se recorren a la izquierda.

$$\begin{array}{c}
+ * -/\overline{baababb}/ + -\overline{abbaaabb} \\
\Downarrow \\
/\underline{+ -\overline{abbaaabb}} + * -/\overline{baababb}
\end{array} \tag{2.8}$$

1-point-recombination recibe dos elementos de la población y selecciona un punto al azar en el cromosoma. El cromosoma de cada elemento es cortado en este punto e intercambiado entre ellos. La Ecuación (2.9) muestra un ejemplo de esta operación. Los individuos se encuentran separados por el caracter |. La subcadena seleccionada para intercambiarse se encuentra subrayada.

$$\begin{aligned}
 & + * - / \overline{baababb} / + - \overline{abbaaabb} | - a / b / \overline{bbaba} * \underline{+aa / \overline{aaabb}} \\
 & \quad \Downarrow \\
 & + * - / \overline{baababb} / \underline{+aa / \overline{aaabb}} | - a / b / \overline{bbaba} * \underline{+ - \overline{abbaaabb}}
 \end{aligned} \tag{2.9}$$

El operador 2-point-recombination recibe dos elementos de la población, selecciona dos puntos del cromosoma e intercambia la información que está entre estos puntos. La Ecuación (2.10) muestra un ejemplo de esta operación.

$$\begin{aligned}
 & + * - / \overline{baababb} / + - \overline{abbaaabb} | - a / b / \overline{bbaba} * \underline{+aa / \overline{aaabb}} \\
 & \quad \Downarrow \\
 & + * - / \overline{baababb} / \underline{+aa \overline{baaabb}} | - a / b / \overline{bbaba} * \underline{+ - a / \overline{aaabb}}
 \end{aligned} \tag{2.10}$$

Gene-recombination recibe dos individuos y selecciona aleatoriamente un gen de cada individuo para intercambiarlo. La Ecuación (2.11) muestra un ejemplo.

$$\begin{aligned}
 & \underline{+ * - / \overline{baababb} / + - \overline{abbaaabb}} | \underline{- a / b / \overline{bbaba} * +aa / \overline{aaabb}} \\
 & \quad \Downarrow \\
 & \underline{- a / b / \overline{bbaba} / + - \overline{abbaaabb}} | \underline{+ * - / \overline{baababb} * +aa / \overline{aaabb}}
 \end{aligned} \tag{2.11}$$

En el algoritmo 6 (*Operadores Genéticos GEP*) se observa la forma en que se aplican los operadores genéticos en GEP. Se observa que se puede aplicar más de un operador genético a un mismo individuo. La probabilidad de mutación es p_m . La probabilidad de is-transposition es p_{is} . La probabilidad de ris-transposition es p_{ris} . La probabilidad de aplicar gene-transposition es p_{gene} . La probabilidad de 1-point-recombination y 2-point-recombination es $p_{1-point}$ y $p_{2-point}$ respectivamente. La probabilidad de aplicar gene-recombination es p_{gene-r} .

Algoritmo 6 Operadores Genéticos de GEP

OPERADORES GENÉTICOS GEP(*población*)

```

1   $i \leftarrow 1$ 
2   $p_m \leftarrow 0.2$ 
3   $p_{is} \leftarrow p_{ris} \leftarrow p_{gene} \leftarrow g_{gene-r} \leftarrow 0.1$ 
4   $p_{1-point} \leftarrow p_{2-point} \leftarrow 0.3$ 
5  mientras  $i \leq \text{TAMAÑO}(\textit{población})$ 
6     $sw \leftarrow \text{NÚMERO ALEATORIO}(0, 1)$ 
7    si  $p_m > sw$ 
8      entonces MUTACIÓN(población[i])
9    si  $p_{is} > sw$ 
10     entonces ISTRANSPOSITION(población[i])
11   si  $p_{ris} > sw$ 
12     entonces RISTRANSPOSITION(población[i])
13   si  $p_{gene} > sw$ 
14     entonces GENETRANSPOSITION(población[i])
15   si  $p_{1-point} > sw$  y  $\text{TAMAÑO}(\textit{población}) \neq i$ 
16     entonces 1-POINTRECOMBINATION(población[i], población[i + 1])
17   si  $p_{2-point} > sw$  y  $\text{TAMAÑO}(\textit{población}) \neq i$ 
18     entonces 2-POINTRECOMBINATION(población[i], población[i + 1])
19   si  $p_{gene-r} > sw$  y  $\text{TAMAÑO}(\textit{población}) \neq i$ 
20     entonces GENERECOMBINATION(población[i], población[i + 1])

```

En este capítulo se hizo introducción a los algoritmos genéticos, programación genética y gene expression programming. Los tres algoritmos evolutivos presentados en este capítulo comparten un algoritmo general (ver Algoritmo 1).

Estos tres algoritmos (AG, PG y GEP) son diferentes en la estructura de su cromosoma. Los algoritmos genéticos tienen un cromosoma fijo, la programación genética tiene un cromosoma variable que tiene una estructura de árbol y gene expression programming tiene un cromosoma fijo que representa una estructura variable tipo árbol.

En el siguiente capítulo se presenta ECSID, el cual es un software tipo caja negra desarrollado para la identificación de sistemas. ECSID utiliza todas las herramientas descritas en este capítulo para hacer la identificación de sistemas.

Capítulo 3

ECSID

ECSID es un sistema que utiliza AG, PG y GEP para hacer identificación de sistemas. ECSID crea un modelo a partir de las observaciones del sistema. Usando PG y GEP, ECSID usa tres diferentes formas para representar un modelo; puede crear modelos autoregresivos, modelos con una ecuación diferencial ordinaria (EDO) y con un sistema de EDOs. Utilizando AG ECSID puede evolucionar un modelo tipo ARIMA (Autoregressive Integrated Moving Average).

ECSID utiliza las observaciones del sistema, un conjunto de terminales T y un conjunto de funciones F .

El conjunto de terminales T está formado por las variables de estado del sistema, por ejemplo, si se está modelando el péndulo lineal (ver 4.2.3) el conjunto de terminales está formado por θ' , θ , por el tiempo t y por un número generador de números aleatorios \mathfrak{R} ($T = \{\theta', \theta, t, \mathfrak{R}\}$).

El conjunto de funciones F depende del sistema que se quiera modelar. Por ejemplo, en el modelo del péndulo lineal, el conjunto de funciones F está formado por las funciones aritméticas básicas ($F = \{+, -, *, /\}$). Este conjunto puede incluir funciones logarítmicas, exponenciales o funciones particulares para un problema.

Tanto el conjunto de funciones F como el conjunto de terminales T es específico para cada sistema a modelar. En el conjunto de terminales se incluyen tantas variables como tenga el sistema. En el caso de que se modele mediante una EDO, se deben incluir las variables del sistema y las derivadas del mismo. Solamente es necesario especificar una cota superior en el orden de las derivadas, ECSID es capaz de crear modelos de orden inferior. Por ejemplo si se quiere evolucionar un sistema de cuarto orden el conjunto de terminales

sería $T = \{x^{(3)}, x^{(2)}, x', x, t, \mathfrak{R}\}$. En el conjunto de funciones F se incluyen las funciones que puedan crear una ecuación que modele al sistema. Por ejemplo si se esta haciendo una regresión simbólica y el sistema presenta oscilaciones sería muy recomendable incluir en el conjunto de funciones F las funciones trigonométricas o si el sistema es lineal y de coeficientes constantes sería recomendable que el conjunto de funciones F solamente tuviera las funciones aritméticas. En el caso de modelos autoregresivos en el conjunto de funciones se incluyen funciones que hacen referencia a los valores pasados en la predicción.

El conjunto de funciones F debe de ser mínimo pero suficiente para poder representar el sistema que se esta modelando. Si F no es suficiente es muy probable que el individuo encontrado sea complejo de comprender o ECSID no sea capaz de modelar el sistema. De lo contrario si F el espacio de búsqueda se amplia lo cual hace que ECSID sea más lento para encontrar una solución.

ECSID puede construir modelos ARIMA, la forma general de un modelo ARIMA se muestra en la Ecuación (3.1). Para modelar este tipo de sistemas ECSID utiliza un cromosoma formado por valores reales, cada uno de éstos representa los valores anteriores de la predicción o los errores de predicción.

$$f(t) = \sum_{i=t-\tau}^{t-1} a_i f(i) + \sum_{i=t-\tau}^{t-1} b_i e(i) \quad (3.1)$$

Uno de los tipos de modelos que evoluciona ECSID es una EDO. La Ecuación (3.2) muestra el modelo general de una EDO. ECSID solamente evoluciona la parte derecha de la Ecuación (3.2). El orden del sistema se obtiene con la derivada de más alto orden que tiene el individuo más uno. La figura 3.1 muestra un la siguiente EDO $y^{(2)} = 7y' + 10y + 12$ representada en ECSID, en esta figura se observa, que se evoluciona solamente la parte derecha de la ecuación $7y' + 10y + 12$. Además se puede observar que el sistema es de segundo orden porque la derivada de más alto orden del individuo es de primer orden.

$$y^{(n)} = f(t, y, y', y'', \dots, y^{(n-1)}) \quad (3.2)$$

Para integrar la Ecuación (3.2) con un método numérico, se requiere transformar la Ecuación (3.2) a la forma de la Ecuación (3.3). Esto se hace con una simple sustitución de variables $y_1 = y, y_2 = y', y_3 = y'', \dots, y_n = y^{(n-1)}$.

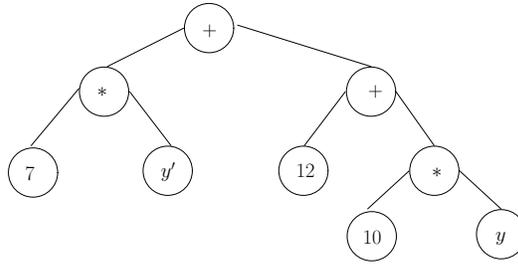


Figura 3.1: EDO representada en ECSID.

$$\begin{cases} y'_1 = y_2 \\ y'_2 = y_3 \\ \vdots \\ y_n = f(t, y_1, y_2, \dots, y_n) \end{cases} \quad (3.3)$$

El otro método que se utiliza para crear modelos es utilizando los valores pasados de predicción. La Ecuación (3.4) muestra la ecuación general de este modelo. En este método se puede tener accesos a los τ valores pasados; al conjunto de valores pasados se les denomina la ventana de regresión. ECSID utiliza una ventana fija, determinada por el usuario. ECSID solo evoluciona la parte derecha de la Ecuación (3.4).

$$y_k = f(y_{1,k-1}, \dots, y_{1,k-\tau}, y_{n,k-1}, \dots, y_{n,k-\tau}) \quad (3.4)$$

ECSID usa el factor de correlación (Ecuación 3.5) para poder comparar resultados de diferentes dominios. Este factor regresa un valor entre -1 y 1 siendo 1 cuando x y y tienen los mismos valores.

$$r = \frac{n \sum xy - \sum x \sum y}{\sqrt{[n \sum x^2 - (\sum x)^2][n \sum y^2 - (\sum y)^2]}} \quad (3.5)$$

El algoritmo 7 (*Evaluación*) muestra el procedimiento que tiene ECSID para evaluar a cada individuo de la población. En la línea 3 se llama a *Evalúa*, este método depende del tipo de modelo que se está evolucionando. La línea 4 regresa la *población* ordenada de acuerdo a su función de aptitud.

El algoritmo 8 (*Evalúa*) muestra el procedimiento para evaluar un individuo de modelo ARIMA. El ciclo de la línea 4 calcula todos los valores pasados y el ciclo de la línea

Algoritmo 7 Evaluación

```

EVALUACIÓN(población)
1  para  $i \in \textit{población}$ 
2     $\textit{árboles} \leftarrow \text{CREAARBOL}(i)$ 
3    EVALÚA( $\textit{árboles}$ )
4  regresar ORDENA( $\textit{población}$ )

```

7 calcula todos los errores en la predicción. El resultado de este procedimiento se guarda en *resultado*. La línea 11 regresa *resultado*.

El algoritmo 9 (*Evalúa*) muestra el código evaluar un individuo de modelo auto-regresivo. Este código es el mismo que se utiliza para evaluar cualquier regresión simbólica. *Evalúa* recibe una lista de árboles *árboles*. De la línea 1 a la 3 se prueba si *árboles* es una lista, si lo es se llama de nuevo a la función *Evalúa*. En el ciclo de las líneas 5 y 6 se itera por todos los hijos de *árboles* llamando en cada hijo a la función *Evalúa*. En la línea 7 se llama a la función que representa *árboles* pasándole como argumentos los *operandos*.

El método para evaluar un individuo que usa una EDO o un sistema de EDOs es muy similar al anterior. La diferencia se encuentra en que éste requiere de un método de integración. El algoritmo 10 (*Evalúa*) presenta el método para evaluar un sistema que evoluciona EDOs. La línea 1 calcula el orden del sistema, la función *Orden* regresa una lista. La condición de la línea 2 verifica que el orden del sistema sea correcto. Verificar el orden del sistema es necesario porque se puede generar sistemas de EDOs los cuales pueden no ser correctos. La línea 4 ejecuta el runge-kutta para integrar *árbol*. La condición de la línea 6 verifica si el sistema está evolucionando ecuaciones lineales; si *árbol* no es lineal regresa la media de la población mas la *aptitud* como el resultado de la evaluación de *árbol*. En la línea 7 se regresa la *aptitud* de *árbol*.

En la tabla 3.1 se observan los parámetros que se utilizaron en ECSID para todos los experimentos.

Se observa que ECSID es una implementación de algoritmos evolutivos cuya diferencia se encuentra, en la función de evaluación. La función de aptitud debe de ser modi-

Algoritmo 8 Evalúa modelo ARIMA

EVALÚA(*cromosoma*)
 1 *resultado* \leftarrow 0
 2 *contador* \leftarrow 1
 3 *i* \leftarrow 1
 4 **mientras** *contador* \leq LONGITUD(*cromosoma*)/2
 5 *resultado* \leftarrow *resultado* + *cromosoma*[*contador*] * F(*n* - *contador*)
 6 *contador* \leftarrow *contador* + 1
 7 **mientras** *contador* \leq LONGITUD(*cromosoma*)
 8 *resultado* \leftarrow *resultado* + *cromosoma*[*contador*] * E(*n* - *i*)
 9 *contador* \leftarrow *contador* + 1
 10 *i* \leftarrow *i* + 1
 11 **regresar** *resultado*

Algoritmo 9 Evalúa un modelo autoregresivo.

EVALÚA(*árboles*)
 1 **si** *árboles* = **Lista**
 2 **entonces para** *i* \in *árboles*
 3 **regresar** EVALÚA(*i*)
 4 *operandos* \leftarrow []
 5 **para** *i* \leftarrow 1 **hasta** CARDINALIDAD(*árboles*)
 6 *operandos* \leftarrow *operandos* + EVALÚA(HIJO(*árboles*, *i*))
 7 **regresar** APLICA(*árboles*, *operandos*)

Algoritmo 10 Evalúa una EDO o un sistema de EDOs

EVALÚA(*árbol*)

- 1 *orden* ← ORDEN(*árbol*)
 - 2 **si** VERIFICAORDEN(*árbol*, *orden*) = NULO
 - 3 **entonces regresar** ∞
 - 4 *aptitud* ← RUNGEKUTTA(*árbol*, *orden*)
 - 5 **si** NOLINEAL(*árbol*) y SYSTEMASLINEALES()
 - 6 **entonces regresar** *aptitud* + *media*
 - 7 **regresar** *aptitud*
-

Tabla 3.1: Parámetros de PG y GEP

Operador Genético	Probabilidad
Mutación	0.2
Apareamiento	0.7
is-transposition	0.1
ris-transposition	0.1
gene-transposition	0.1
one-point recombination	0.3
two-point recombination	0.3
GP and GEP gene-recombination	0.1

ficada para aceptar los valores pasados de predicción o utilizar un método de integración para encontrar el valor numérico de un sistema de ecuaciones diferenciales.

ECSID (ver [Graff05]) fue desarrollado en common lisp utilizando la implementación de CMUCL [CMU80]. La razón para escoger este lenguaje es la facilidad para desarrollar prototipos; el tiempo de desarrollo es corto, comparado con algunos otros lenguajes como C o Java [Sun96]. De acuerdo a Anderson [Anderson94] es razonable pensar, que un programa codificado en Lisp puede estar dentro del 20% o menos del tiempo de ejecución que el mismo programa implementado en C. Otra de las ventajas es la portabilidad, el código desarrollado puede ser portado a otras implementaciones de common lisp. Una ventaja muy importante que ofrece lisp es la facilidad para evaluar funciones generadas.

Capítulo 4

Resultados

En este capítulo se muestran los resultados obtenidos con ECSID. El sistema fue capaz de obtener modelos en problemas de regresión simbólica (seno y parábola), sistemas dinámicos (péndulo lineal, péndulo no lineal, etc.) y series de tiempo (viento y manchas solares).

Los resultados fueron obtenidos mediante 20 corridas independientes; estas corridas permiten encontrar una aproximación a la cantidad de individuos necesarios para obtener un modelo cuya correlación sea $r \geq 0.99$ [Koza92].

La cantidad de procesamiento requerido es determinada con el número de corridas independientes, z , necesarias para asegurar con una probabilidad $p = 0.99$ el criterio de terminación. Una vez obtenido z , la cantidad de procesamiento se calcula multiplicando z por el número de individuos M y el número de generaciones i .

Para obtener el número de corridas independientes z se usa la Ecuación (4.1) [Koza92] en donde $P(M, i)$ es la probabilidad acumulativa de satisfacer la condición de terminación en la generación i con una población de tamaño M . $P(M, i)$ se obtiene experimentalmente mediante corridas independientes.

$$z = \frac{\log 0.01}{\log(1 - P(M, i))} \quad (4.1)$$

El número de individuos procesados, para encontrar con una probabilidad p un resultado aceptable, se denomina $I(M, i, z)$ (Ecuación 4.2).

$$I(M, i, z) = Mzi \quad (4.2)$$

En la Tabla 4.1 se muestran los parámetros usados para realizar los experimentos.

Tabla 4.1: Parámetros de los experimentos

Tamaño de la cabeza H	4
Altura máxima	4
Número máximo de nodos	100
Número de generaciones	500
Número de individuos	500
Criterio de terminación	$r \geq 0.99$

En la Sección 4.1 se presentan ejemplos de regresión simbólica. La Sección 4.2 muestra sistemas dinámicos. En la sección 4.4 se modelan series de tiempo.

4.1. Regresión Simbólica

En esta sección se muestran dos ejemplos de regresión simbólica. El primero es encontrar una función sinusoidal y el segundo una parábola.

El problema de regresión simbólica se define como sigue: dada una serie de tiempo x , encontrar una función que se comporte de manera similar a x .

4.1.1. Seno

En este ejemplo se modela una función senoidal (Ecuación 4.3). El conjunto de terminales utilizado en este experimento es $T = \{x, \Re\}$ y el conjunto de funciones es $F = \{+, -, *, /, \sin, \cos, \sqrt{\cdot}, \ln\}$.

ECSID fue capaz de obtener la misma función usando tanto PG y GEP. El coeficiente de correlación en ambos casos es $r = 1$. La gráfica 4.1 muestra este modelo. En la gráfica *PG*, *GEP* y *Real* son las gráficas de los modelos obtenidos por PG, GEP y el sistema real respectivamente.

$$f(x) = \sin(x) \tag{4.3}$$

En este ejemplo se realizó otro experimento cambiando el conjunto de funciones a $F = \{+, -, *, /\}$. Este experimento fue desarrollando utilizando programación genética. La Ecuación (4.4) muestra el modelo obtenido. Este modelo tiene un coeficiente de correlación de $r = 0.99286$. Se observa que el modelo es mucho más complejo que el obtenido por la

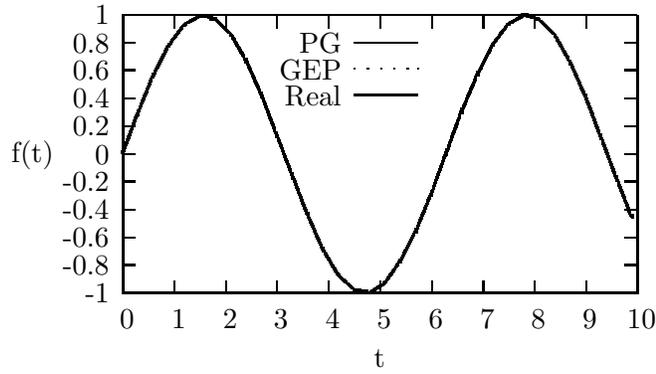


Figura 4.1: Seno.

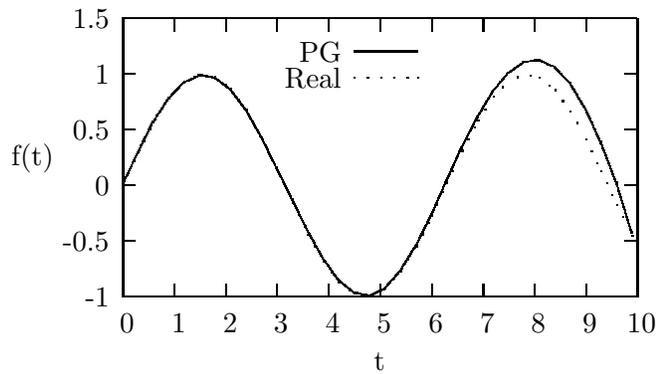


Figura 4.2: Seno $F = \{+, -, *, /\}$

Ecuación (4.3) esto es porque en el conjunto de funciones no se encuentra ninguna función trigonométrica.

$$f(x) = \frac{x}{5.1854 x^3 \left(\frac{x+85.4666}{x} - x + \frac{130.5233}{26.8886 \left(\frac{-x + \frac{5.1854}{x} + 101.8516}{x} - x \right)} + 5.1854 \right)} + 1 \quad (4.4)$$

$$\frac{x^2}{(51.88818 - 5.1854 x^2) (0.0724 (-0.1928 x^3 - 6.1854 x^2 + 1.0) + 53.7275)}$$

La figura 4.2 muestra el modelo obtenido por PG y el modelo real. Se observa que el modelo exhibe un comportamiento similar al sistema buscado. El modelo se empieza a desviar de los datos reales hasta $t = 7$.

De este ejemplo podemos concluir que cuando el conjunto de funciones F no contiene las funciones óptimas para representar el sistema se obtiene un modelo más complejo

y menos preciso.

4.1.2. Parábola

Este ejemplo modela una parábola (Ecuación 4.5). El conjunto de terminales es $T = \{x, \Re\}$ y el conjunto de funciones es $F = \{+, -, *, /, \sin, \cos\}$.

ECSID usando GEP encontró la Ecuación (4.6) la cual tiene un coeficiente de correlación de $r = 0.99984$ y usando PG encontró la Ecuación (4.7) con $r = 0.99975$. La Figura 4.3 muestra estas ecuaciones. En la figura *GEP* es la gráfica del modelo obtenido utilizando GEP, *PG* es la gráfica del modelo de PG y *Real* es la gráfica del sistema.

$$f(x) = x^2 + 5x \quad (4.5)$$

$$f(x) = x^2 + x + 47.255413 \quad (4.6)$$

$$f(x) = x^2 \quad (4.7)$$

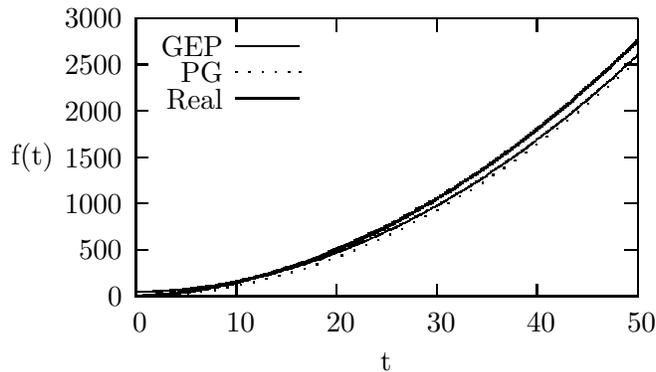


Figura 4.3: Parábola

En este ejemplo se observa que los modelos obtenidos no contienen ninguna función trigonométrica aún y cuando estas fueron incluidas en el conjunto de funciones F . Siempre es más recomendable tener funciones que puedan no ser necesarias para construir el modelo a no tenerlas en F .

En la tabla 4.2 se muestran los resultados de estos experimentos. De esta tabla se puede concluir que los experimentos presentados pueden ser considerados como sencillos,

dado que no se requiere evaluar una gran cantidad de individuos para encontrar un resultado satisfactorio. En el experimento del seno (Ecuación 4.4) no se logró obtener información para calcular $I(M, i, z)$. Dado que de las 20 corridas independientes solo una logró obtener un modelo con coeficiente de correlación $r \geq 0.99$. En esta tabla se observa que todos los modelos obtenidos tienen un coeficiente de correlación superior a 0.99 lo cual los hace exhibir un comportamiento muy similar al sistema que está modelando.

Tabla 4.2: Resultados de regresión simbólica

Nombre	Método	Mejor individuo	r	$I(M, i, z)$	Generación
Seno	GEP	Ec. (4.3)	$r = 1$	500	1
Seno	GP	Ec. (4.3)	$r = 1$	500	1
Seno $F = \{+, -, *, /\}$	GP	Ec. (4.4)	$r = 0.99286$	N/A	N/A
Parábola	GEP	Ec. (4.6)	$r = 0.99984$	500	1
Parábola	GP	Ec. (4.7)	$r = 0.99975$	500	1

4.2. Sistemas Dinámicos

En esta sección se muestran los resultados obtenidos en sistemas dinámicos. ECSID fue capaz de modelar: un sistema masa resorte con amortiguador, masa resorte acoplados, el péndulo lineal, no lineal, no lineal con fricción, ecuaciones diferenciales lineales con coeficientes variables, etc.

4.2.1. Masa Resorte con Amortiguador

En este apartado se encuentra el modelo de un sistema masa-resorte-amortiguador con y sin excitación. La Figura 4.4 muestra el diagrama del sistema.

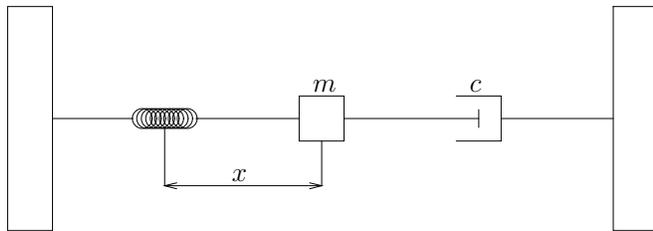


Figura 4.4: Masa Resorte con Amortiguador

La EDO que describe un sistema masa resorte con amortiguador es:

$$x^{(2)} = -\frac{m}{c}x' - kx + F(t) \quad (4.8)$$

Para los siguientes ejemplos se consideran los siguientes parámetros $m = 1$, $c = 1$ y $k = 3$, sustituyendo éstos en la Ecuación (4.8) se obtiene la Ecuación (4.9) que es el modelo a identificar, las condiciones iniciales son: $f(0) = 1$ y $f'(0) = 0$. El conjunto de terminales es $T = \{x', x, t, \Re\}$ y el conjunto de funciones es $F = \{+, -, *, /\}$.

ECSID obtuvo el modelo exacto usando GEP y PG (Ecuación 4.9). En la Figura 4.5 se puede observar estas ecuaciones.

$$x^{(2)} = -x' - 3x \quad (4.9)$$

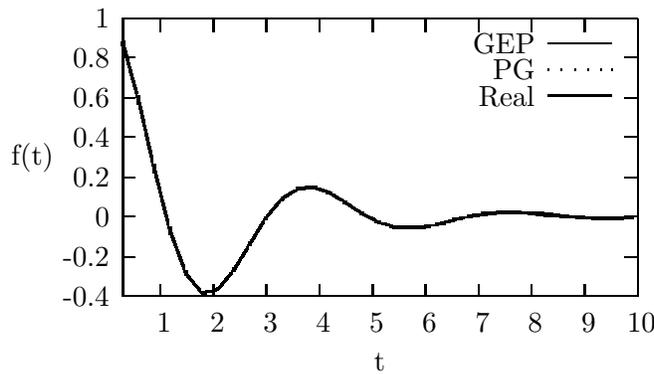


Figura 4.5: Sistema masa resorte amortiguador

A continuación se modela el sistema masa resorte con amortiguador y excitación externa $F(t) = \sin(t)$, la Ecuación (4.10) muestra el sistema.

En este ejemplo se incorpora a ECSID la excitación senoidal externa. Esto hace que los modelos evolucionados siempre tengan el término $\sin(t)$. El conjunto de terminales es $T = \{x', x, t, \Re\}$ y el conjunto de funciones es $F = \{+, -, *, /\}$.

El modelo encontrado por ECSID usando GEP y PG es el mismo que la Ecuación (4.10). La Figura 4.6 muestra estos modelos.

$$x^{(2)} = \sin(t) - x' - 3x \quad (4.10)$$

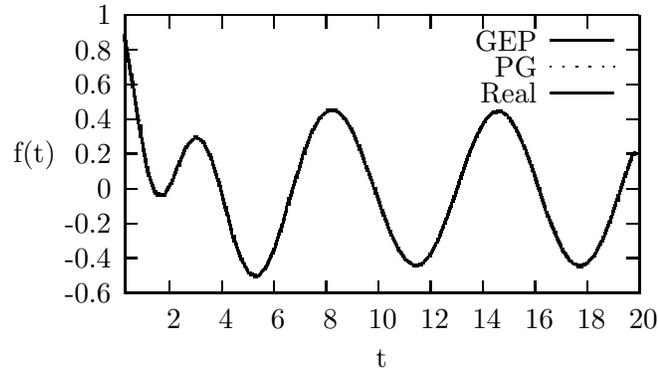


Figura 4.6: Sistema masa resorte con amortiguador y excitación externa

4.2.2. Masas y Resortes Acoplados

En este ejemplo se modela un sistema de masas y resortes acoplados; este sistema puede ser modelado por una ecuación diferencial de cuarto orden o dos ecuaciones de segundo orden. En este ejemplo se optó por usar un sistema de dos ecuaciones. La Figura 4.7 muestra un diagrama de este sistema.

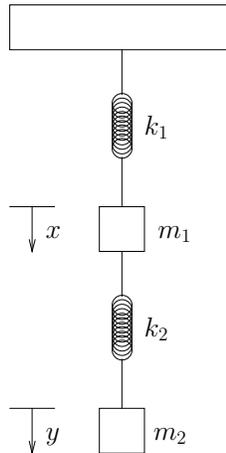


Figura 4.7: Diagrama de un sistema masa resorte acoplados

El modelo a encontrar se muestra en la Ecuación (4.11) esta sistema de ecuaciones son obtenidas cuando se sustituyen los siguientes valores: $m_1 = m_2 = 1$, $k_1 = 3$ y $k_2 = 2$. Las condiciones iniciales son: $f(0) = 1$, $f'(0) = 0$, $g(0) = 2$ y $g'(0) = 0$. El conjunto de terminales es $T = \{x', x, y', y, t, \Re\}$ y el conjunto de funciones es $F = \{+, -, *, /\}$.

$$\begin{aligned}x^{(2)} &= -5x + 2y \\y^{(2)} &= 2x - 2y\end{aligned}\tag{4.11}$$

El modelo encontrado por ECSID usando GEP y PG se muestra en la Ecuación (4.12). Como se observa, este modelo no tiene la misma estructura que la Ecuación (4.11), pero sus coeficientes de correlación son $r_1 = 1$ y $r_2 = 1$. La Figura 4.8 muestra estos modelos. En la gráfica *GEP-1* y *GEP-2* representan las gráficas de las ecuaciones 1 y 2, respectivamente, del modelo obtenido por GEP. *PG-1* y *PG-2* son las gráficas de las ecuaciones 1 y 2, respectivamente, del modelo obtenido por PG. *Real-1* y *Real-2* son las gráficas de las ecuaciones 1 y 2, respectivamente, del modelo buscado.

$$\begin{aligned}x^{(2)} &= -x \\y^{(2)} &= -y\end{aligned}\tag{4.12}$$

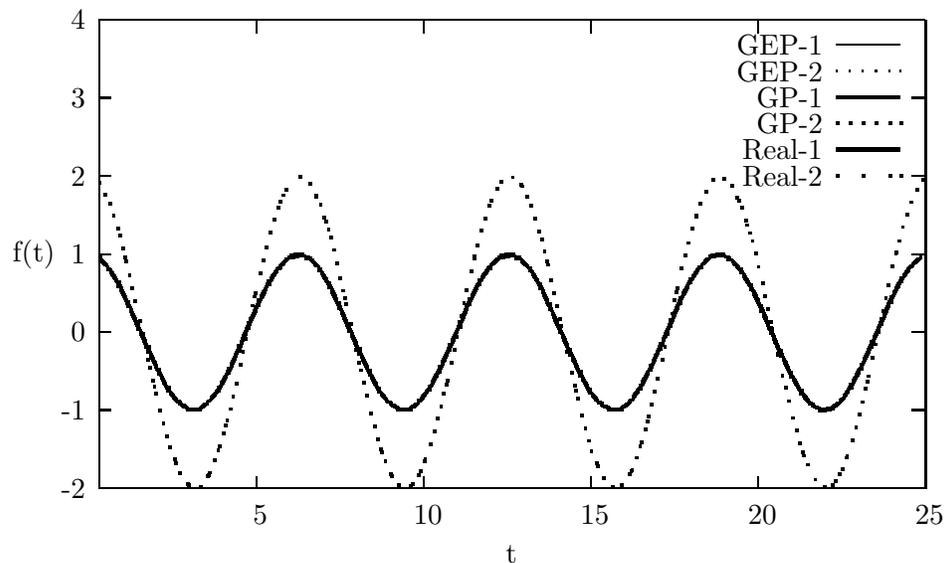


Figura 4.8: Sistema masa resortes acoplados

4.2.3. Péndulo

En este apartado se modela un péndulo lineal, un péndulo no lineal y un péndulo no lineal con fricción. El diagrama del péndulo es mostrado en la Figura 4.9.

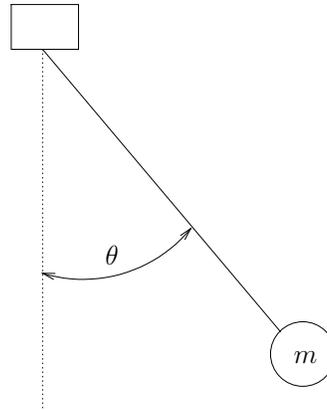


Figura 4.9: Diagrama de un péndulo

La Ecuación (4.13) muestra el modelo general de un péndulo lineal; si sustituimos $L = 0.5$ obtenemos la Ecuación (4.14). Éste es el modelo a simular, las condiciones iniciales son: $f(0) = 1$ y $f'(0) = 0$. El conjunto de terminales es $T = \{\theta', \theta, t, \Re\}$ y el conjunto de funciones es $F = \{+, -, *, /\}$.

$$\theta^{(2)} = \frac{g}{L}\theta \quad (4.13)$$

$$\theta^{(2)} = -19.6\theta \quad (4.14)$$

Las Ecuaciones (4.15) y (4.16) muestran los modelos obtenidos por ECSID usando GEP y PG respectivamente. Se observa que los modelos obtenidos tienen la misma estructura que el modelo real y solamente tienen los parámetros ligeramente diferentes. El modelo obtenido por GEP tiene un coeficiente de correlación $r = 0.99635$ y el de PG tiene un coeficiente de correlación $r = 0.99996$. Se observa que el modelo obtenido por PG es muy similar al sistema buscado. La figura 4.10 muestra el modelo obtenido y el real.

$$\theta^{(2)} = -19.33501\theta \quad (4.15)$$

$$\theta^{(2)} = -19.62675\theta \quad (4.16)$$

En el siguiente ejemplo se simula un péndulo no lineal, la Ecuación (4.17) muestra este modelo, las condiciones iniciales y el conjunto de terminales son los mismos que en el ejemplo pasado. El conjunto de funciones es $F = \{+, -, *, /, \sin, \cos, \exp\}$.

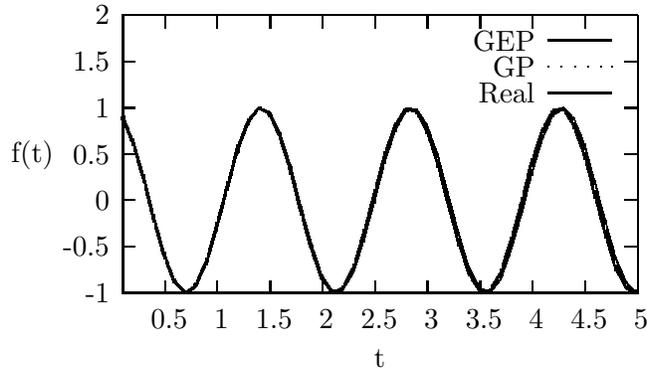


Figura 4.10: Péndulo lineal

$$\theta^{(2)} = -19.6\sin(\theta) \quad (4.17)$$

Las Ecuaciones (4.18) y (4.19) muestran los modelos obtenidos por ECSID usando GEP y PG, respectivamente. Los coeficiente de correlación de los modelos obtenidos por GEP y PG son $r = 0.9999$ y $r = 0.99863$, respectivamente. Los coeficientes de correlación indican que los dos modelos obtenidos tienen un comportamiento similar al sistema modelado. Aun y cuando su comportamiento es similar, ninguno de los modelos tiene la misma estructura que el modelo real. La Figura 4.11 muestra los dos modelos obtenidos y el modelo real.

$$\theta^{(2)} = -14.53186166\theta \quad (4.18)$$

$$\theta^{(2)} = -21.96932 \sin(\theta) \cos(\sin(3.47683 \cos(\sin(22.69132 \cos(\cos(\sin(\theta') + \theta)))))) \quad (4.19)$$

En el siguiente ejemplo se modela un péndulo no lineal con fricción. la Ecuación (4.20) muestra el modelo general de éste, sustituyendo $c = 1$, $m = 1$ y $L = 0.5$ en (4.20) se obtiene la Ecuación (4.21), la cual es el modelo a simular. Las condiciones iniciales y los conjuntos T y F son los mismo que en el ejemplo pasado.

$$\theta^{(2)} = -\frac{c}{mL}\theta' - \frac{g}{L}\sin(\theta) \quad (4.20)$$

$$\theta^{(2)} = -2\theta' - 19.6\sin(\theta) \quad (4.21)$$

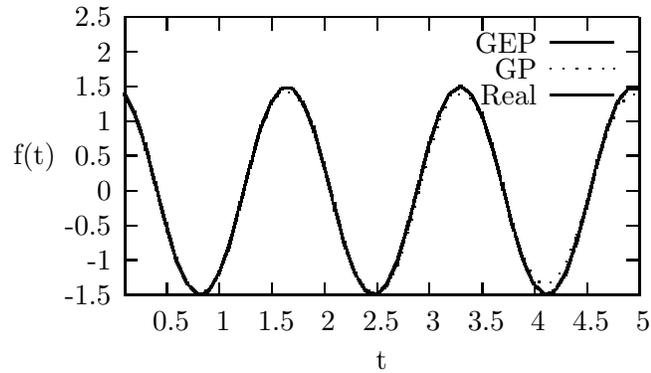


Figura 4.11: Péndulo no lineal

Los modelos encontrados por ECSID usando GEP y PG se muestran en las ecuaciones (4.22) y (4.23) respectivamente. El coeficiente de correlación del modelo obtenido utilizando GEP es $r = 0.99822$ y del modelo de PG es $r = 0.99697$. La Figura 4.12 muestra los modelos obtenidos y el modelo real, se observa que los modelos obtenidos tienen un comportamiento similar que el modelo real. Aun y cuando los modelos obtenidos tienen un comportamiento similar, ninguno presenta la misma estructura que el modelo real.

$$\theta^{(2)} = -2\theta\theta' - 2\theta' - 20\theta \quad (4.22)$$

$$\theta^{(2)} = 4.25972(-0.54108(\theta' + \theta) - 4\theta) \quad (4.23)$$

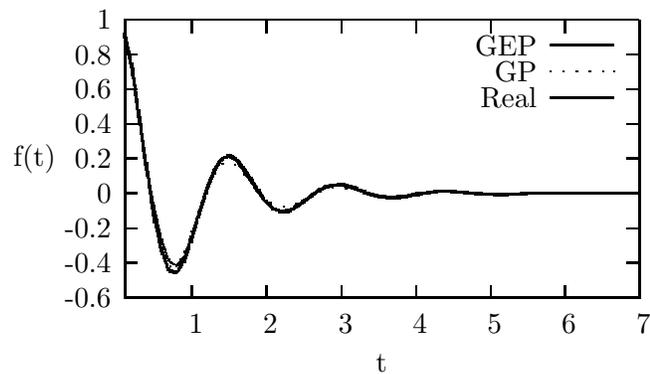


Figura 4.12: Péndulo no lineal con fricción

4.2.4. Circuito Eléctrico

En esta sección se simula un circuito eléctrico lineal, formado por dos inductores, tres resistencias y una fuente de voltaje. La Figura 4.13 muestra su diagrama. ECSID también puede modelar circuitos eléctricos no lineales.

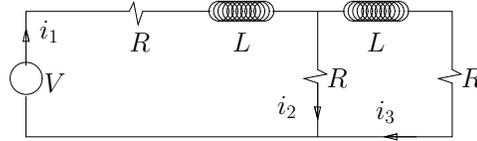


Figura 4.13: Diagrama de un circuito eléctrico

La Ecuación (4.24) se obtiene cuando sustituyendo los siguiente valores: $E = 100$, $R = 10$ y $L = 1$, las condiciones iniciales son $f(0) = 0$ y $g(0) = 0$. El conjunto de terminales es $T = \{i_1, i_3, t, \mathfrak{R}\}$ y el conjunto de funciones es $F = \{+, -, *, /\}$.

$$\begin{aligned} i_1' &= -20i_1 + 10i_3 + 100 \\ i_3' &= 10i_1 - 20i_3 \end{aligned} \quad (4.24)$$

Los modelos obtenidos por ECSID usando GEP y PG se muestran en las ecuaciones (4.25) y (4.26), respectivamente. En estas ecuaciones se puede observar que el modelo obtenido por GEP tiene una estructura diferente que el modelo real y el modelo obtenido por PG tiene la misma estructura que el modelo real. Los coeficientes de correlación del modelo obtenido con GEP es $r_1 = 0.98623$ y $r_2 = 0.99753$. El coeficiente de correlación del modelo de PG es $r_1 = 0.99579$ y $r_2 = 0.99491$.

$$\begin{aligned} i_1' &= \frac{1424}{25} - 8i_1 \\ i_3' &= \frac{1399}{50} - 4i_1 \end{aligned} \quad (4.25)$$

$$\begin{aligned} i_1' &= -19.09362i_1 + 4i_3 + 109.37688 \\ i_3' &= 6i_1 - 11i_3 \end{aligned} \quad (4.26)$$

En la Figura 4.14 se muestran los modelos obtenidos y el modelo real; se observa que los modelos obtenidos son similares al modelo real pero ambos pueden mejorarse. En el modelo obtenido por PG se pueden mejorar los parámetros.

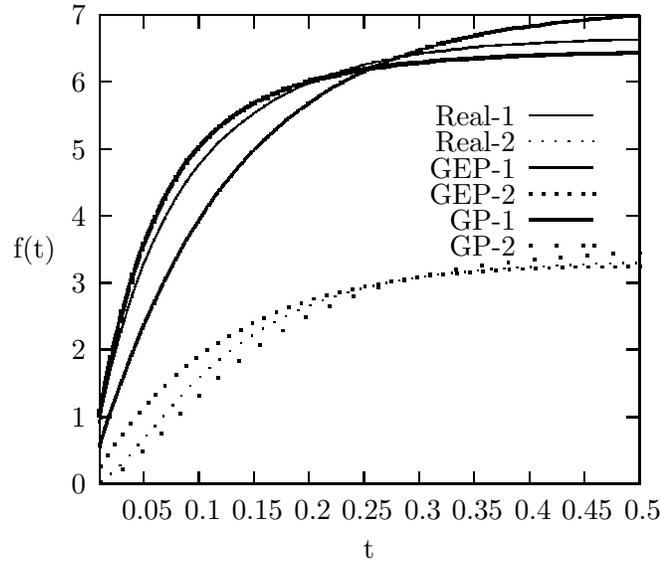


Figura 4.14: Circuito eléctrico.

4.2.5. Ecuaciones lineales con Coeficientes Variables

En esta sección se simulan ecuaciones diferenciales lineales con coeficientes variables. Este tipo de ecuaciones no pueden ser evolucionadas por ninguno de los trabajos relacionados, dado que ellos no usan ningún módulo para identificar si una ecuación es lineal o no.

La Ecuación (4.27) muestra el modelo a simular, las condiciones iniciales son $f(1) = 2$. El conjunto de terminales es $T = \{x, t, \Re\}$ y el conjunto de funciones es $F = \{+, -, *, /\}$.

$$x' = -\frac{2x}{t} + 4t \quad (4.27)$$

Las ecuaciones (4.28) y (4.29) fueron obtenidas por ECSID usando GEP y PG respectivamente. Los coeficientes de correlación utilizando GEP y PG son $r = 0.99972$ y $r = 0.99962$ respectivamente. Como se observa, ninguna de las ecuaciones tiene la misma estructura que el modelo original, pero los dos modelos se comportan de manera similar al sistema original. Además se observa que los modelos obtenidos son lineales con coeficientes variables.

$$x' = 2t - 0.270554 \quad (4.28)$$

$$x' = 20.17977(t^2 - x) \quad (4.29)$$

En la Figura 4.15 se muestran los modelos obtenidos por ECSID y el modelo real, se observa que los modelos obtenidos tienen un comportamiento similar al modelo real.

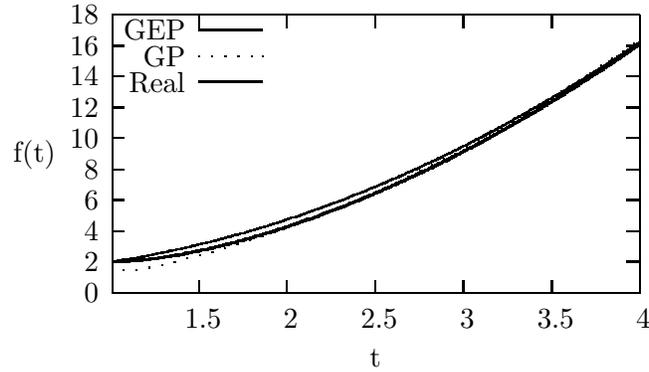


Figura 4.15: Ecuación (4.27).

En la Ecuación (4.30) se muestra el siguiente sistema a modelar, las condiciones iniciales son $f(4) = 2$. El conjunto de terminales es $T = \{x, t, \Re\}$ y el conjunto de funciones es $F = \{+, -, *, /, \sin, \cos, \ln, \exp\}$.

$$x' = \frac{2t - \ln tx}{t - 3} \quad (4.30)$$

Las ecuaciones (4.31) y (4.32) muestran los modelos obtenidos por ECSID usando GEP y PG, respectivamente. Como se puede observar en la gráfica 4.16, estos modelos no son similares al modelo buscado pero son modelos que tienen un comportamiento similar al modelo real. Los coeficientes de correlación de los modelos obtenidos usando GEP y PG son $r = 0.99624$ y $r = 0.9953$.

$$x' = -2x + 2 \log(\log t) + t + \sin\left(\frac{29.3542}{t} + 92.40376\right) \frac{2t}{t-3} \quad (4.31)$$

$$x' = + \frac{2t}{t-3} \quad (4.32)$$

La Ecuación (4.33) muestra el siguiente modelo a simular, las condiciones iniciales son $f(\pi) = 0$. Los conjuntos T y F son los mismos que en el ejemplo anterior.

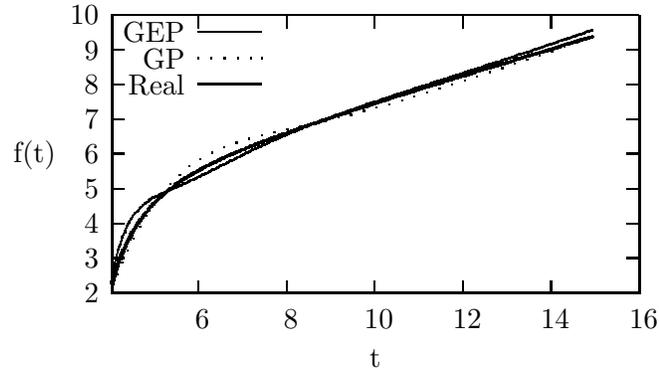


Figura 4.16: Ecuación (4.30)

$$x' = \sin(t) - \tan(t)x \quad (4.33)$$

Las ecuaciones (4.34) y (4.35) muestran los modelos obtenidos por ECSID usando GEP y PG. El coeficiente de correlación de GEP es $r = 0.99976$ y de PG es $r = 0.99446$. Se observa que ninguno de los dos modelos obtenidos tiene la misma estructura del modelo real. El modelo obtenido por PG no es lineal, debido a que no se encontró un mejor modelo lineal que el modelo de la Ecuación (4.35).

$$x' = x - 0.18950707 + \sin(t) \quad (4.34)$$

$$x' = 2 \sin(t) \cos^2(x) \quad (4.35)$$

La Figura 4.17 muestra estas ecuaciones, se puede observar que el modelo obtenido usando PG se desvía en $t = 4.4$ del modelo real, pero el modelo obtenido por GEP exhibe un comportamiento más similar al modelo real.

La Ecuación (4.36) muestra el siguiente modelo a simular, las condiciones iniciales son $f(0) = 0$ y $f'(0) = 0$. El conjunto de terminales es $T = \{x', x, t, \Re\}$ y el conjunto de funciones es $F = \{+, -, *\}$.

En este ejemplo también se incluyó la $F(t) = 0.5 \cos(0.8t)$ en ECSID. Por lo cual todos los individuos evolucionados tienen ese término.

$$x^{(2)} = 0.5 \cos(0.8t) - x \quad (4.36)$$

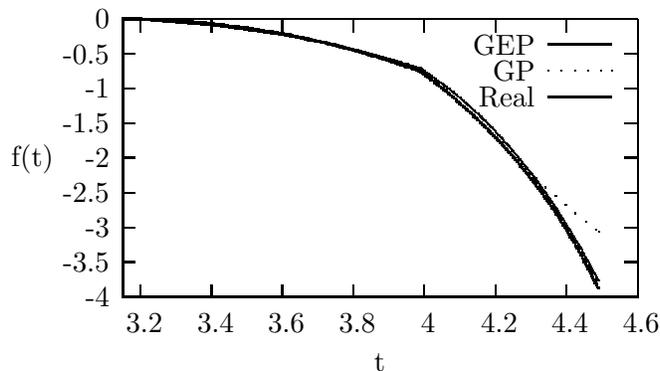


Figura 4.17: Ecuación (4.33)

ECSID encontró el mismo modelo (Ecuación 4.36) usando tanto GEP como PG, la Figura 4.18 muestra estos modelos.

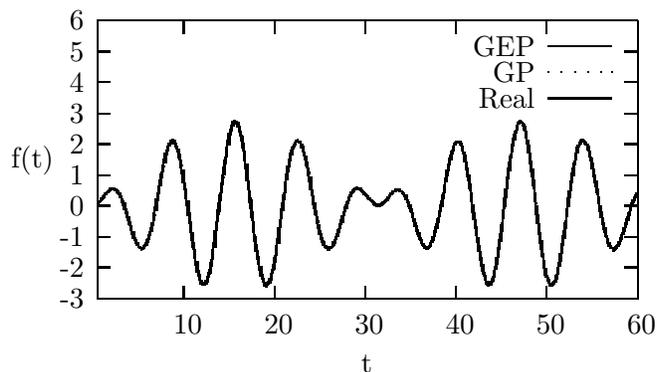


Figura 4.18: Ecuación (4.36)

4.2.6. Ecuación de Van der Pol

En esta sección se simula la ecuación de Van der Pol (Ecuación 4.37), se observa que este sistema es no lineal. Las condiciones iniciales son $f(0) = 1.5$ y $f'(0) = 0$. El conjunto de terminales es $T = \{x', x, t, \Re\}$ y el conjunto de funciones es $F = \{+, -, *, /\}$.

$$x^{(2)} = (1 - x^2)x' - x \quad (4.37)$$

ECSID encontró el mismo modelo usando tanto GEP como PG; en la Figura 4.19 se muestran los modelos.

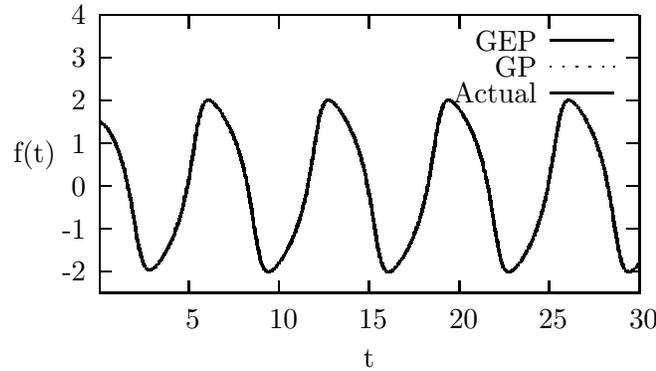


Figura 4.19: Ecuación de Van der Pol

La tabla 4.3 muestra un resumen de los resultados obtenidos por ECSID, ésta muestra el método utilizado, la ecuación obtenida, el coeficiente de correlación, el número de individuos necesarios para obtener el individuo y el número de generaciones.

En esta tabla se observa que PG evaluó menos individuos en todos los casos excepto en las Ecuaciones (4.23) y (4.35). En la Ecuación (4.35) la diferencia de individuos evaluados entre PG y GEP es poca por lo cual se puede atribuir a errores en la forma que se calcula $I(M, i, z)$. En el último ejemplo presentado en la tabla se observa una gran diferencia entre el número de individuos evaluados por GEP y PG.

La tabla 4.3 también da idea de la complejidad del problema. Mientras más individuos sean necesario evaluar, más complejo es el problema. Además la columna *Generación* indica el número de generaciones necesarias, se observa que ningún experimento de PG requiere más de 80 generaciones por lo cual el parámetro que se tiene de 500 generaciones en ECSID está muy sobrado. Los experimentos que requieren solamente una generación para encontrar la solución, son experimentos muy sencillos que no requieren ningún procedimiento para encontrarlos, esto quiere decir que solamente es necesario crear individuos al azar.

4.3. Motor de Corriente Directa

En este ejemplo se modelan las fuerzas que actúan en la flecha de un motor de corriente directa. La Ecuación (4.38) muestra este modelo, donde w es la velocidad de la flecha, b es un coeficiente de pérdidas mecánicas, J es el momento de inercia y T es el torque.

Tabla 4.3: Resultados de sistemas dinámicos

Método	Mejor individuo	r	$I(M, i, z)$	Generación
GEP	Ec. (4.9)	$r = 1$	110,163	77
GP	Ec. (4.9)	$r = 1$	9,000	18
GEP	Ec. (4.10)	$r = 1$	148,750	33
GP	Ec. (4.10)	$r = 1$	27,643	11
GEP	Ec. (4.12)	$r_1 = r_2 = 1$	41,276	4
GP	Ec. (4.12)	$r_1 = r_2 = 1$	34,664	9
GEP	Ec. (4.15)	$r = 0.99635$	N/A	N/A
GP	Ec. (4.16)	$r = 0.99996$	413,165	64
GEP	Ec. (4.18)	$r = 0.9999$	262,012	137
GP	Ec. (4.19)	$r = 0.99863$	48,001	48
GEP	Ec. (4.22)	$r = 0.99822$	154,783	15
GP	Ec. (4.23)	$r = 0.99697$	278,609	27
GEP	Ec. (4.25)	$r_1 = 0.98623$ y $r_2 = 0.99753$	N/A	N/A
GP	Ec. (4.26)	$r_1 = 0.99579$ y $r_2 = 0.99491$	N/A	N/A
GEP	Ec. (4.28)	$r = 0.99972$	769	1
GP	Ec. (4.29)	$r = 0.99962$	500	1
GEP	Ec. (4.31)	$r = 0.99624$	44,891	1
GP	Ec. (4.32)	$r = 0.9953$	30,344	25
GEP	Ec. (4.34)	$r = 0.99976$	1,001	1
GP	Ec. (4.35)	$r = 0.99446$	1,661	1
GEP	Ec. (4.36)	$r = 1$	3000	6
GP	Ec. (4.36)	$r = 1$	1000	2
GEP	Ec. (4.37)	$r = 1$	2,872,996	64
GP	Ec. (4.37)	$r = 1$	197,770	37

$$w' = \frac{b}{J}w + \frac{T}{J} \quad (4.38)$$

Los datos usados para la identificación se tomaron directamente de un motor de corriente directa, sin aplicarles ningún filtro para eliminar el ruido. Las condiciones iniciales del sistema son: $w(0) = 0$. El conjunto de terminales es $T = \{w, t, \Re\}$ y el conjunto de funciones es $F = \{+, -, *, /\}$. El modelo que se busca en este experimento es un modelo lineal.

El modelo encontrado por ECSID usando PG tiene un coeficiente de correlación de $r = 0.99390$, el modelo se puede observar en la Ecuación (4.39).

$$w' = 6075.11126 - 83.58453w \quad (4.39)$$

La Figura 4.20 se muestra el modelo obtenido y los datos reales.

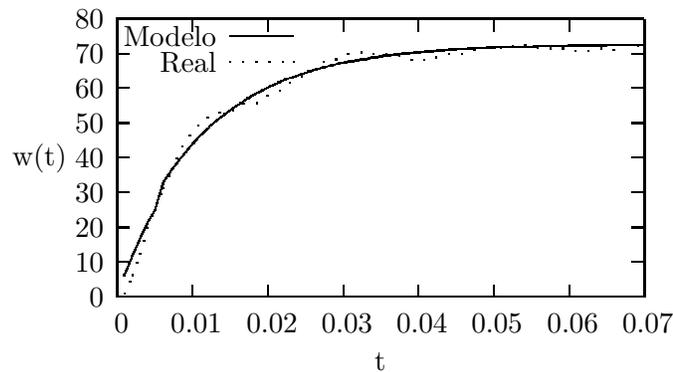


Figura 4.20: Motor de corriente directa

4.4. Series de Tiempo

En esta sección se muestran dos series de tiempo, la primera es una serie de manchas solares y la segunda es una serie de tiempo de velocidad del viento. En estos dos experimentos el tamaño de la ventana es fijo y tiene un valor de 16. La ventana es el número de valores pasados que se pueden incluir en el modelo.

El modelado de las series de tiempo tiene una aplicación importante dado que sirve para poder predecir como se va a comportar un fenómeno en el futuro. Por ejemplo, la serie

de tiempo de la velocidad de viento ayuda a predecir como se va a comportar el viento en los siguiente 12 meses.

En el modelado de las series de tiempo se tienen dos periodos, el primero llamado “entrenamiento”, este es utilizado para evolucionar un modelo; el segundo es llamado “predicción” es utilizado para evaluar el modelo a la hora de predecir.

Los modelos pueden tener coeficientes de correlación altos en el entrenamiento y esto no necesariamente hace que el modelo tenga un coeficiente de correlación alto en el periodo de predicción. Esto puede ser, porque se tuvo un entrenamiento excesivo y se aprendió el ruido de la serie de tiempo. Lo importante de un modelo de una serie de tiempo es el periodo de predicción; éste debe de acercarse lo más posible a la realidad.

4.4.1. Manchas Solares

Las manchas solares son regiones del sol las cuales marcan una temperatura menor a sus alrededores y una actividad magnética intensa. El número de manchas solares está directamente relacionada con la intensidad de la radiación solar. A mayor número de manchas solares se incrementa la intensidad de radiación del sol.

El tamaño de la ventana de las dos series es de 16. El conjunto de terminales es $T = \{t\}$ y el conjunto de funciones es $F = \{+, -, *, /, \sin, \cos, \sqrt{\cdot}, \ln, f(n-1), \dots, f(n-16)\}$. En algunos ejemplos se añade al conjunto de funciones los errores de predicción los cuales son: $\{e(n-1), \dots, e(n-16)\}$. Cuando no se tiene información para calcular los errores de predicción se dice que el error es 0.

La Ecuación (4.40) fue obtenida por ECSID usando GEP tomando en cuenta los errores en la predicción. Se puede observar que el modelo final no toma en cuenta ningún error en la predicción. El coeficiente de correlación en la parte de entrenamiento es $r = 0.89221$; este coeficiente no es tan alto como en los ejemplos anteriores, pero el sistema que se está modelando es más complejo, dado que tiene datos atípicos. Por ejemplo, en $t = 251$ se tiene un pico que no está presente en ningún otro momento de la serie. El coeficiente de correlación en la parte de predicción es $r_v = 0.65272$, lo cual significa que el modelo obtenido es bueno en el periodo de entrenamiento pero no es un buen modelo para predecir.

La Figura 4.21 muestra los datos y el modelo obtenido. Se puede observar que los primeros puntos de validación son buenos pero los últimos ya no predicen adecuadamente.

La Figura 4.22 muestra el porcentaje de error en el modelo de la Ecuación (4.40). Se observa que algunos puntos tienen un porcentaje de error superior al 800 %. En general solo el 39 % de los datos tienen un porcentaje de error inferior al 20 %, el 73 % de los datos tienen un porcentaje de error inferior al 50 % y solamente un 15 % de los datos tienen un porcentaje de error superior al 100 %. El máximo porcentaje de error es de 871 %. Este modelo es el peor modelo obtenido para modelar esta serie de tiempo y esto se observa en el porcentaje de error.

$$f(n) = -0.35666f(n - 3) + f(n - 1) + 16.49205 \quad (4.40)$$

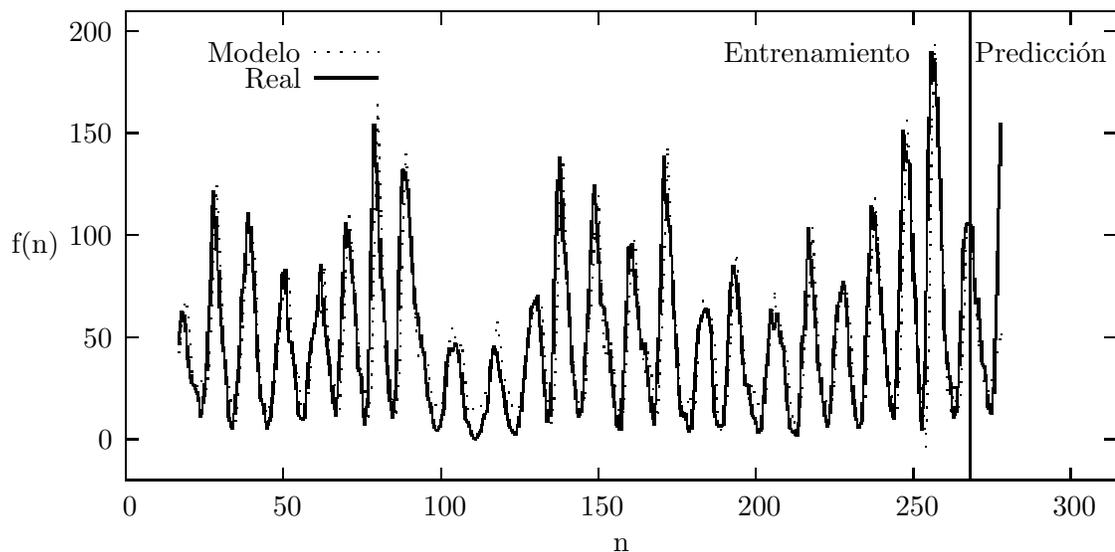


Figura 4.21: Manchas solares sin errores de predicción (GEP)

La Ecuación (4.41) muestra el modelo obtenido por ECSID utilizando PG cuando se toman en cuenta los errores de predicción e . La figura 4.23 muestra este modelo y la serie de tiempo. El coeficiente de correlación en el periodo de predicción es $r = 0.91767$, este coeficiente es más alto que en el modelo anterior pero aún así el coeficiente de correlación en el periodo de predicción ($r_v = 0.64245$) es menor, por lo cual lo hace un peor modelo para predecir. De la figura podemos observar que en el periodo de predicción algunos puntos se ven desplazados hacia la izquierda y los últimos puntos de la predicción no predicen adecuadamente.

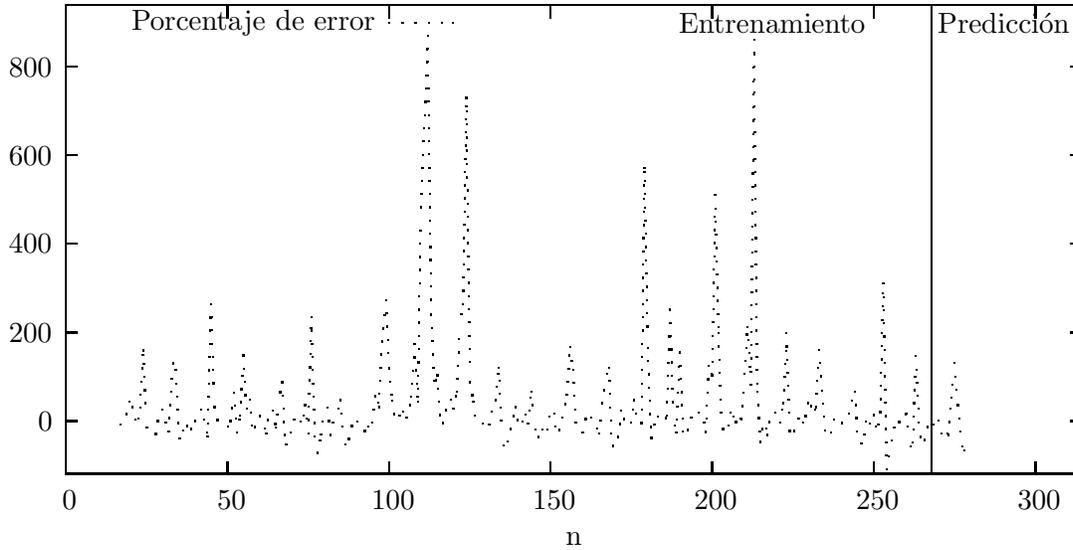


Figura 4.22: Porcentaje de error de manchas solares Ec. (4.40).

$$\begin{aligned}
 f(n) = & -0.17218861196177 \left(-\frac{0.02964891808932 f(n-3) f(n-9)}{f(n-1) - f(n-9)} \right. \\
 & - f(n-9) + f(n-2) \left. - 0.01875328421916 \left(\frac{5.807585}{f(n-2) - f(n-1)} \right. \right. \\
 & \left. \left. - \frac{f(n-3)}{\frac{5.807585 (f(n-3) - e(n-14))}{-f(n-9) - e(n-6) + 6.7097645} - f(n-1)} \right) f(n-9) \right. \\
 & \left. - 0.17218861196177 f(n-3) + f(n-1) + 6.709764502555219 \right)
 \end{aligned} \tag{4.41}$$

La Ecuación (4.42) muestra el modelo obtenido por ECSID utilizando GEP sin utilizar los errores de predicción en el conjunto de funciones F . La figura 4.24 muestra el modelo obtenido y la serie de tiempo. El coeficiente de correlación en el periodo de entrenamiento es $r = 0.87369$, este es el menor coeficiente de todos los experimentos de la serie de manchas solares. En el periodo de predicción se tiene también el coeficiente más bajo de todas las series que es $r_v = 0.25076$. De la figura se observa que en el periodo de predicción el modelo no sigue al último pico del sistema real.

$$f(n) = -0.21871 f(n-3) + f(n-1) + 7.48676 \tag{4.42}$$

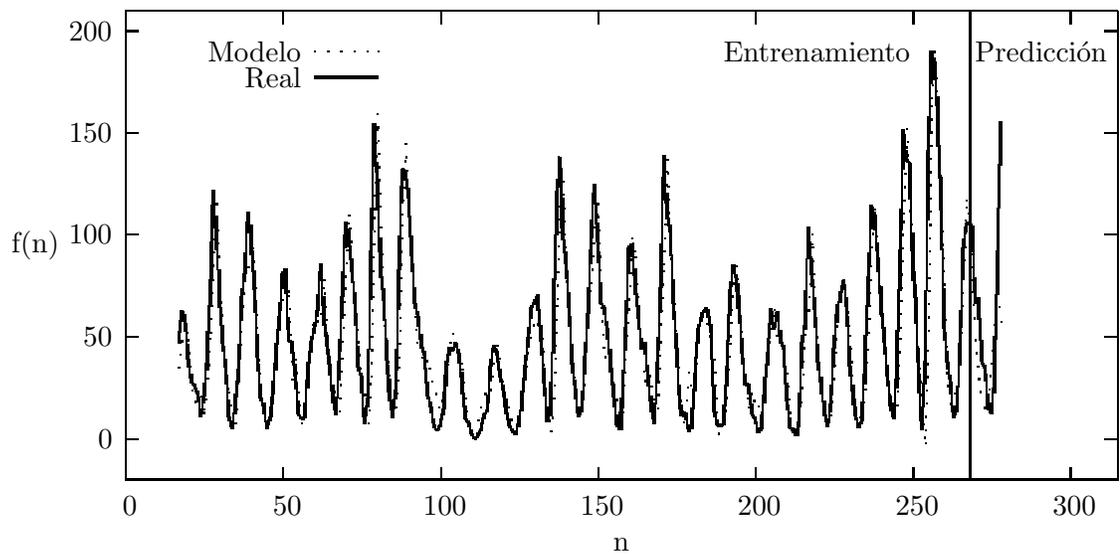


Figura 4.23: Manchas solares con errores de predicción (PG)

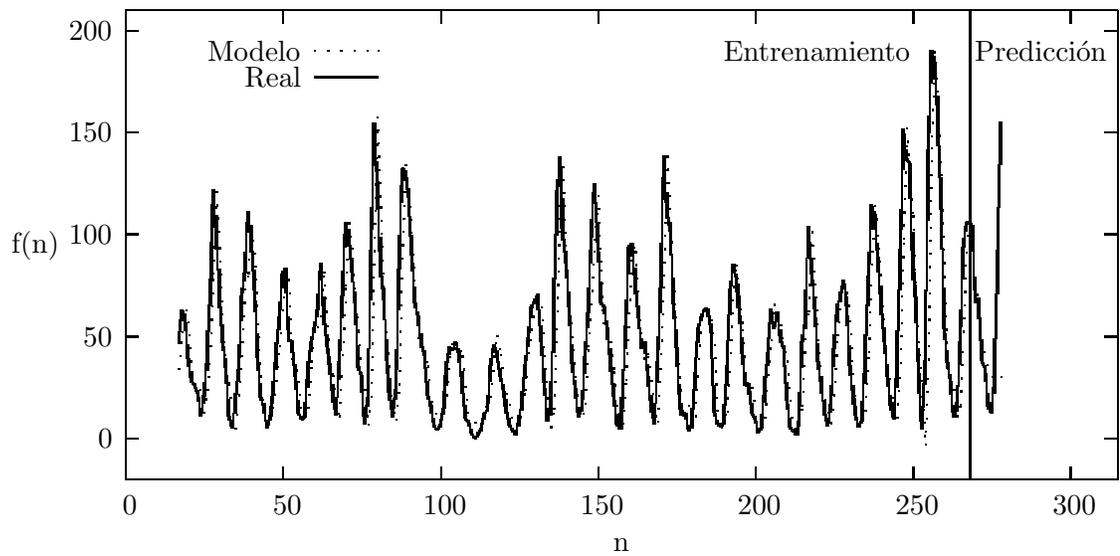


Figura 4.24: Manchas solares sin errores de predicción (GEP)

La Ecuación (4.43) muestra el modelo obtenido por ECSID utilizando PG sin tomar en cuenta los errores de predicción. La figura 4.25 muestra este modelo y la serie de tiempo. Este modelo tiene los coeficientes más altos que son $r = 0.91868$ y $r_v = 0.88482$ para el periodo de entrenamiento y predicción respectivamente. De la figura podemos observar en el periodo de entrenamiento que la gran mayoría de los puntos se acercan a la serie real existiendo algunos pico y valles que son pronosticados por el modelo y no se encuentran en la serie real. En el periodo de predicción que los primeros predicen de manera adecuada y que los últimos puntos no predicen tan bien sin embargo este modelo es el que más se acerca de todos los modelos, es el mejor modelo obtenido de la serie de tiempo y su predicción es aceptable para los primeros 6 años.

La Figura 4.26 muestra el porcentaje de error de la Ecuación (4.43). En este modelo se tiene que el 51 % de los datos tienen un porcentaje de error inferior al 20 %, el 78 % de los datos tienen un porcentaje de error inferior al 50 % y solamente un 10 % de los datos tienen un margen de error superior al 100 %. El máximo porcentaje de error es de 805 %. De la gráfica se observa que los máximos errores de porcentaje se encuentra cuando el modelo real es casi cero por lo cual este dato no debe de ser tomado en cuenta, dado que la diferencia el modelo real y el obtenido es menor de 10. Este modelo tiene porcentajes de error inferior a los de la Ecuación (4.40), pero aún así, solo sería utilizado cuando la aplicación pueda aceptar errores superiores al 100 %, aunque estos, solo se presenten en el 10 % de los datos.

$$\begin{aligned}
f(n) = & -(\sin \sin \cos f(n-1) - \sqrt{f(n-1) - f(n-8)} - f(n-7) \\
& + 2f(n-3) + 2f(n-2) - 2\sqrt{f(n-2)} - 2\sqrt{f(n-1) - f(n-2)})^{\frac{1}{2}} \\
& + (-\cos(f(n-1) - \sqrt{f(n-1)}) - \cos f(n-1)) \\
& + f(n-11) + 3f(n-10) + 3f(n-7) - f(n-5) - f(n-4) \\
& - 5f(n-3) - 7f(n-2) + 5f(n-1))^{\frac{1}{2}} + f(n-1)
\end{aligned} \tag{4.43}$$

La tabla 4.4 muestra los resultados obtenidos por ECSID, ésta muestra el método utilizado, el modelo obtenido, el coeficiente de correlación en el periodo de entrenamiento r , el coeficiente en el periodo de predicción r_v , la suma de los errores absolutos $\Sigma|e|$ en el periodo de entrenamiento y la suma de los errores absolutos $\Sigma|e_v|$ en el periodo de predicción.

Se hicieron dos tipos de experimentos, los que toman en cuenta los errores de predicción y los que no los toman en cuenta. En los experimentos de PG (4.41) y (4.43) se observan que los dos modelos tienen un coeficiente de correlación en el periodo de entrena-

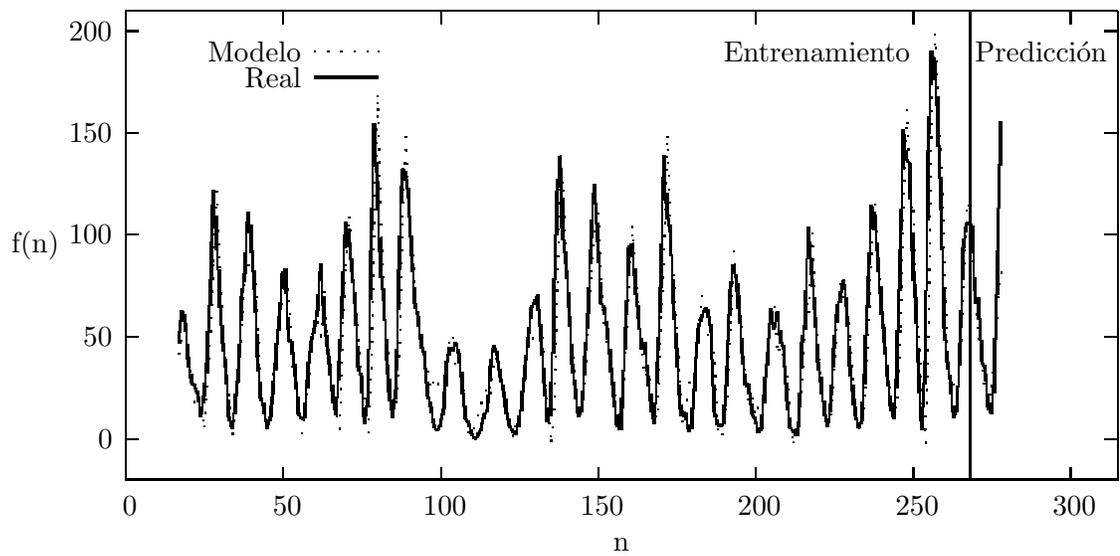


Figura 4.25: Manchas solares sin errores de predicción (PG)

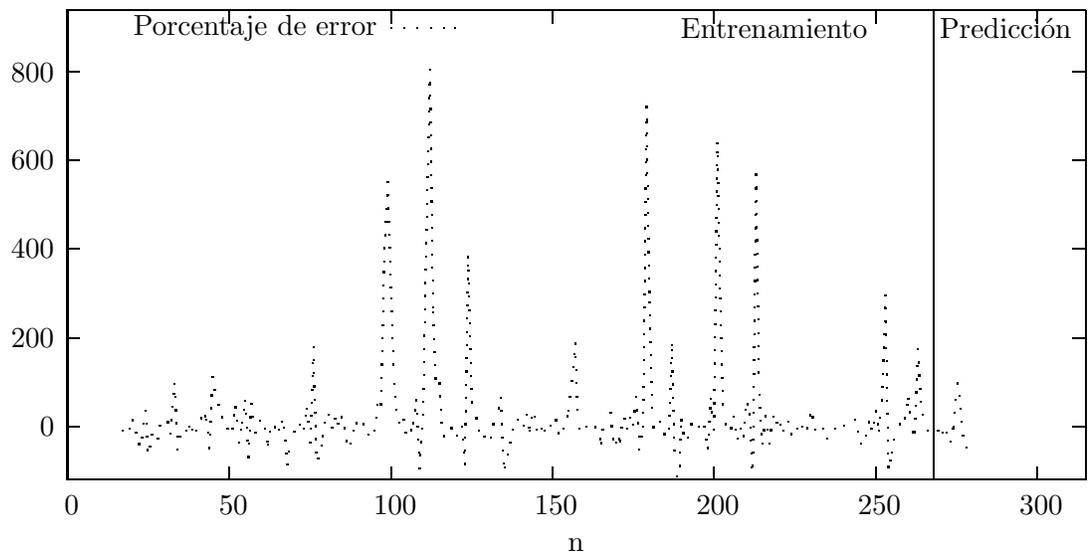


Figura 4.26: Porcentaje de error de manchas solares Ec. (4.43).

miento muy similar, pero que el modelo que no toma en cuenta los errores de predicción e se comporta de una mejor manera en el periodo de predicción. El mejor modelo obtenido en esta serie de tiempo es el de la Ecuación (4.43), este modelo se comporta mejor tanto en el periodo de entrenamiento como de predicción.

Tabla 4.4: Resultados de Manchas Solares

Método	Ecuación	r	r_v	$\Sigma e $	$\Sigma e_v $
GEP con e	(4.40)	$r = 0.89221$	$r_v = 0.65272$	3157.6309	242.79767
PG con e	(4.41)	$r = 0.91767$	$r_v = 0.64245$	2728.5073	305.14488
GEP sin e	(4.42)	$r = 0.87369$	$r_v = 0.25076$	3369.247	270.37188
PG sin e	(4.43)	$r = 0.91868$	$r_v = 0.88482$	2717.894	184.51397

4.4.2. Velocidad del Viento

En la serie de tiempo del viento se desarrolló un modelo ARIMA (ver [Makridakis92]) usando el procedimiento estándar. Este modelo fue desarrollado con el propósito de comparación. El modelo fue obtenido usando el programa Minitab [Min].

En la Ecuación (4.44) se muestra el modelo general de ARIMA donde $f(i)$ representa el valor de la serie en el instante i y $e(i)$ es el vector de los errores de predicción.

$$f(t) = \sum_{i=t-\tau}^{t-1} a_i f(i) + \sum_{i=t-\tau}^{t-1} b_i e(i) \quad (4.44)$$

La Ecuación (4.45) muestra el modelo obtenido usando Minitab: la Figura 4.27 muestra el modelo obtenido y la serie de tiempo. En la figura se observa que el modelo tiene un comportamiento pobre tanto en el periodo de entrenamiento como en el de predicción. También se observa que el modelo se encuentra desplazado hacia abajo en el periodo de predicción. Los coeficientes de correlación son $r = 0.51933$ y $r_v = 0.896761$ para el periodo de entrenamiento y de predicción, respectivamente.

La Figura 4.28 muestra el porcentaje de error de la Ecuación (4.45). En este modelo se tiene que el 13% de los datos tienen un porcentaje de error menor al 20% y el 90% de los datos tienen un porcentaje de error menor al 50%. El máximo porcentaje de error presentado es del 67%.

$$f(n) = f(n-1) + f(n-2) - f(n-13) - 0.997e(n-1) - 0.7976e(n-12) + 0.7956e(n-13) \quad (4.45)$$

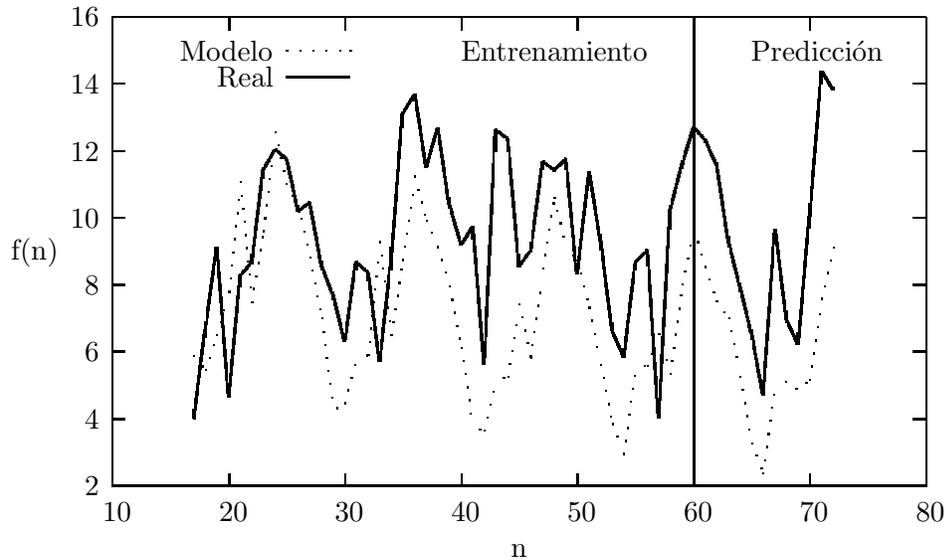


Figura 4.27: Viento (ARIMA)

La Ecuación (4.46) fue obtenida utilizando GEP, usando los errores de predicción en el conjunto de funciones. La figura 4.29 muestra los datos reales y el modelo obtenido. Los coeficientes de correlación para los dos periodos son $r = 0.59052$ y $r_v = 0.23194$. El coeficiente en el periodo de entrenamiento es mayor que el modelo anterior y éste se observa en la figura 4.29 donde el modelo tiene un comportamiento similar a la serie real. En el periodo de predicción el modelo falla en todos los puntos y los últimos cuatro puntos están muy alejados de la serie real.

$$\begin{aligned}
 f(n) = & \sqrt{(\sin(f(n-15)))} \sqrt{(\sin(e(n-13)))} \\
 & (\sin(\sin(\sin(e(n-13)))) + \sqrt{(\sin(e(n-13)))}) \\
 & + e(n-13) + 9.1094265
 \end{aligned} \tag{4.46}$$

La Ecuación (4.47) muestra el modelo obtenido por ECSID usando PG, utilizando los errores de predicción en el conjunto F . El modelo encontrado es un modelo que no nos proporciona una idea de las dinámicas que gobiernan a esta serie de tiempo. La figura 4.30 muestra el modelo y la serie real de tiempo. Los coeficientes de correlación en los dos periodos son $r = 0.59084$ y $r_v = 0.72386$. De la figura se observa que en el periodo de entrenamiento el modelo se comporta de manera similar a la serie real, pero falla en casi todos los puntos y no logra aprender todos los puntos inferiores. En el periodo de predicción

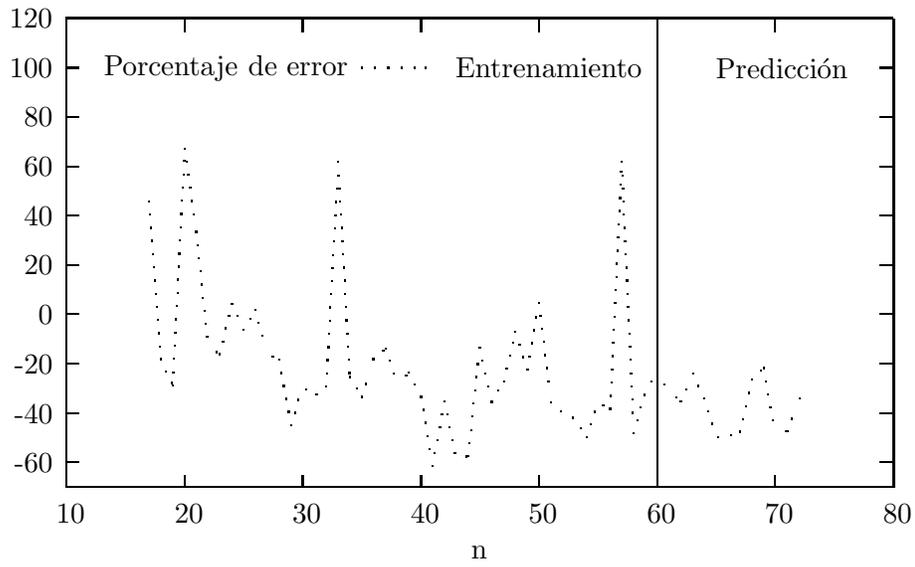


Figura 4.28: Porcentaje de error de velocidad del viento Ec. (4.45).

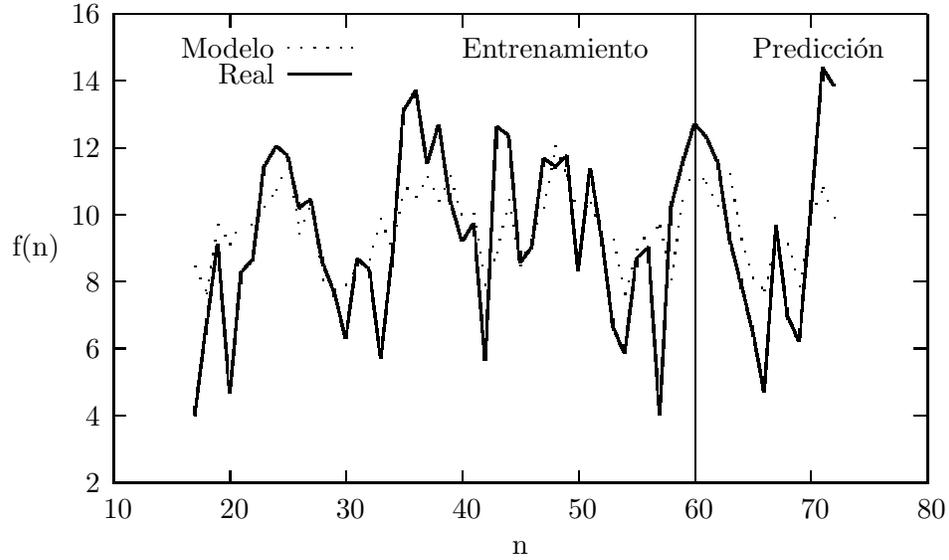


Figura 4.29: Viento con errores de predicción (GEP)

falla en casi todos los puntos y tampoco logra predecir los puntos inferiores de la serie real.

$$\begin{aligned}
 f(n) &= f(n-1)^{\frac{1}{4}} f(n-11)^{\frac{1}{4}} f(n-12)^{\frac{3}{32}} f(n-13)^{\frac{1}{4}} f(n-15)^{\frac{1}{8}} \\
 &= (f(n-1)^{\frac{25}{32}} f(n-12)^{\frac{1}{8}} ((f(n-5) - f(n-1)f(n-3))f(n-15))^{\frac{1}{4}})^{\frac{1}{32}} \\
 &= (\sqrt{(f(n-13) - \sqrt{f(n-15)})f(n-16)} \\
 &+ f(n-7)^{\frac{1}{4}} \sqrt{f(n-12)} f(n-1)^{\frac{1}{8}} e(n-16)^{\frac{21}{32}} f(n-12)^{\frac{43}{32}})^{\frac{1}{64}}
 \end{aligned} \tag{4.47}$$

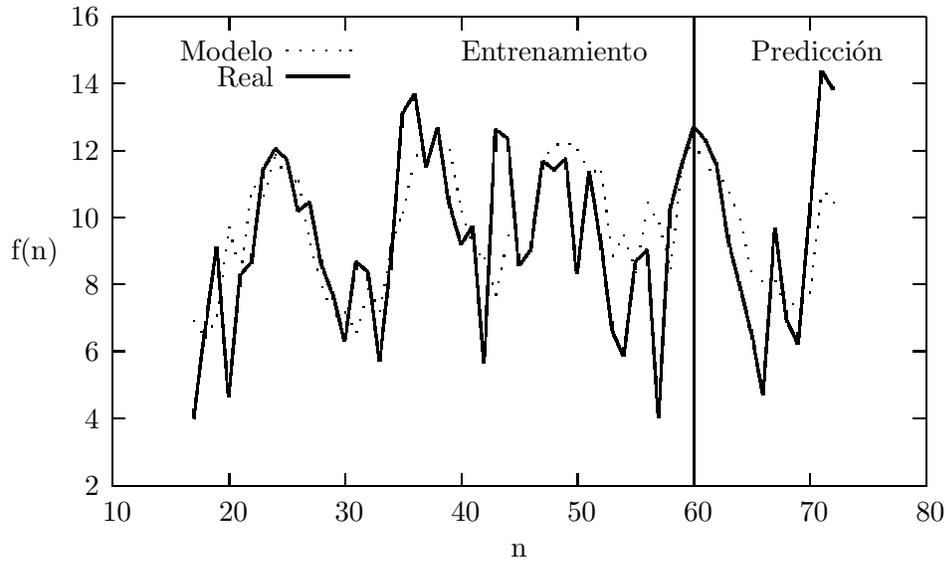


Figura 4.30: Viento con errores de predicción (PG)

La Ecuación (4.48) muestra el modelo obtenido utilizando GEP sin usar los errores de predicción en el conjunto de funciones. Los coeficientes de correlación de los dos periodos son $r = 0.73137$ y $r_v = 0.90111$. De acuerdo a sus coeficientes de correlación este modelo es el mejor modelo encontrado para esta serie de tiempo. La figura 4.31 muestra los datos reales y el modelo obtenido. De la figura se observa que en el periodo de entrenamiento algunos puntos no se apegan a la serie real pero el modelo logra pronosticar tanto los puntos superiores como los puntos inferiores. En el periodo de predicción el modelo se comporta de manera muy similar al modelos real. Los primeros punto no lograron modelar el primer pico de la serie real pero si se lograron modelar los siguientes puntos. Al final tiende a predecir valores menores cuando la serie real sigue creciendo.

La Figura 4.32 muestra el porcentaje de error de la Ecuación (4.48). En este modelo

se tiene que el 76 % de los datos tienen un porcentaje de error inferior al 20 % y un 96 % de los datos tienen un porcentaje de error inferior al 50 %. Este modelo tiene al 96 % de sus datos con un porcentaje menor al 50 % lo cual es mayor que lo que tiene el modelo utilizado para comparación (Ecuación 4.45). El modelo puede ser utilizado siempre y cuando sea aceptable tener un porcentaje de error del 114 % el cual fue el máximo porcentaje de error presentado.

$$\begin{aligned}
 f(n) = & 0.1408 \cos\left(\frac{\sin(\ln(f(n-12)))}{\sin(f(n-14))} + f(n-2)\right) \\
 & + \cos\left(\frac{8.5209}{\frac{6.8594}{\ln(\ln(f(n-3)))} - 5.6555} + f(n-12) - 6.8962\right) \\
 & + \frac{(0.7323f(n-12) + 4.1514) \cos(2 \sin(f(n-12)))}{\frac{f(n-9)}{f(n-14)} - f(n-2)} \\
 & + 0.5986f(n-12)
 \end{aligned} \tag{4.48}$$

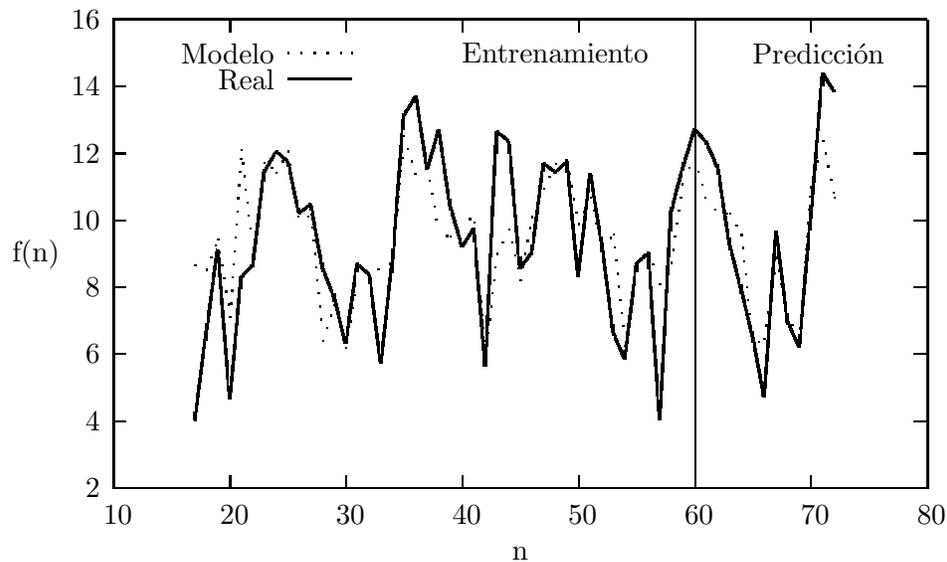


Figura 4.31: Viento sin errores de predicción (GEP)

La Ecuación (4.49) muestra el modelo obtenido por ECSID usado PG sin usar los errores de predicción. La figura 4.33 muestra este modelo y la serie de tiempo. Los coeficientes de correlación son $r = 0.69898$ y $r_v = 0.78821$ del periodo de entrenamiento y predicción. De la figura se observa, en el periodo de predicción, que el modelo falla en

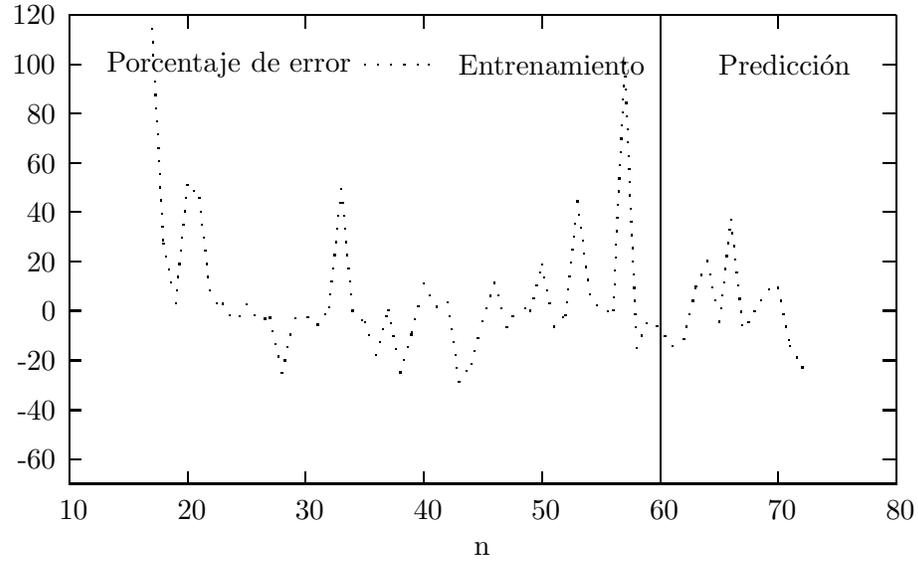


Figura 4.32: Porcentaje de error de velocidad del viento Ec. (4.48).

predecir los primeros puntos pero a diferencia de todos los modelos anteriores este si logra predecir el último pico.

$$\begin{aligned}
 f(n) = & \sin(\sin((\sin(\log(\sin((\sin(\log(2 \sin(\sin f(n-12) + f(n-14) + f(n-2))) \\
 & + \sin f(n-12) + f(n-12) + (-\frac{f(n-1)}{f(n-12)})^{\frac{1}{4}} + f(n-2))) \\
 & + \sin(\log f(n-10) + f(n-12))))^{\frac{1}{8}} + f(n-12)) \\
 & + \sqrt{\sin(\sin \sin(2 f(n-12)) - \log f(n-10) + f(n-6))} \\
 & + f(n-12) + \sin f(n-12))^{\frac{1}{8}} + 2 f(n-12) + f(n-11)) \\
 & + \sin(f(n-13) + f(n-12)) + f(n-14) + f(n-2)) \\
 & + \sqrt{-\sin f(n-1) - f(n-12) + f(n-1) + f(n-12)}
 \end{aligned} \tag{4.49}$$

La tabla 4.5 muestra los resultados obtenidos por ECSID en la serie de tiempo del viento. Esta tabla contiene el coeficiente de correlación en el periodo de entrenamiento r , en el periodo de predicción r_v y la suma de los errores absolutos en el periodo de entrenamiento y predicción, $\Sigma|e|$ y $\Sigma|e_v|$, respectivamente.

En esta tabla se observa que todos los modelos obtenidos por ECSID son mejores al modelo ARIMA. Son mejores porque tienen un valor menor tanto en $\Sigma|e|$ como en

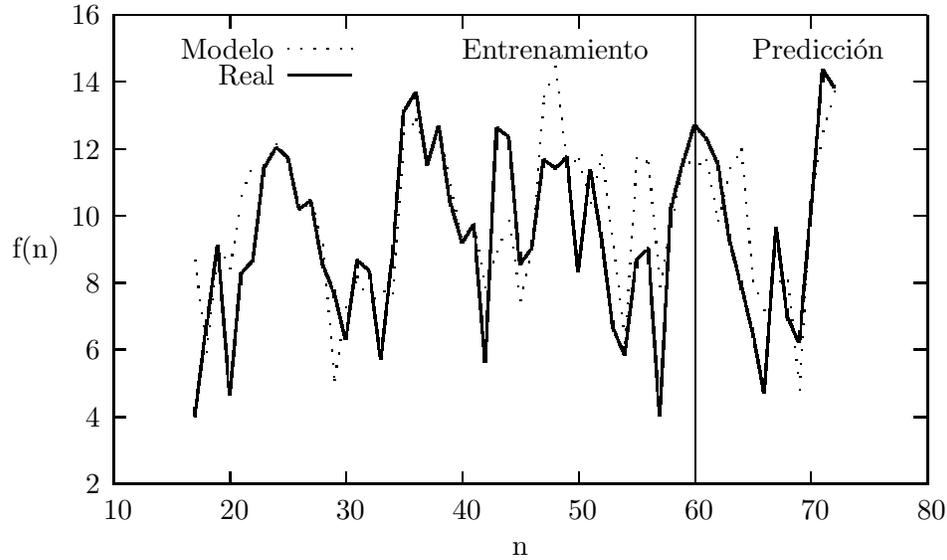


Figura 4.33: Viento sin errores de predicción (PG)

$\Sigma|e_v|$. Además se puede observar que se comportan mejor los modelos en el conjunto de validación cuando son evolucionados sin los errores de predicción. Esto es porque en el conjunto de validación no tenemos información para calcular los errores de predicción y estos se convierten rápidamente en 0.

Tabla 4.5: Resultados del Viento

Método	Ecuación	r	r_v	$\Sigma e $	$\Sigma e_v $
ARIMA	(4.45)	$r = 0.51933$	$r_v = 0.896761$	121.22153	42.58549
GEP con e	(4.46)	$r = 0.59052$	$r_v = 0.23194$	66.12773	34.69025
PG con e	(4.47)	$r = 0.59084$	$r_v = 0.72386$	66.56554	21.99036
GEP sin e	(4.48)	$r = 0.73137$	$r_v = 0.90111$	50.99468	15.09404
PG sin e	(4.49)	$r = 0.69898$	$r_v = 0.78821$	61.42284	19.09754

En este capítulo se presentaron los resultados obtenidos por ECSID. ECSID modeló sistemas lineales y no lineales utilizando EDOs. También creó modelos ARIMA mediante algoritmos genéticos y modelos autoregresivos.

Utilizando EDOs casi todos los modelos obtenidos tienen una estructura simple, lo cual puede servir de guía para entender las características del sistema que se está modelando. Los modelos autoregresivos obtenidos por ECSID, en su mayoría, fueron ecuaciones complejas pero buenos modelos. En el caso de la serie de la velocidad del viento todos los

modelos obtenidos por ECSID fueron mejores que el obtenido por Minitab.

Los modelos de ECSID en sistemas lineales cumplieron la característica de ser lineales. Este tipo de modelos no se pueden desarrollar con ninguno de los trabajos relacionados.

El tiempo de corrida que requiere ECSID para encontrar un modelo adecuado es dependiente de la complejidad del problema y de la cantidad de datos que se tienen. Por ejemplo en los problemas considerados difíciles (ver Sección 4.2) ECSID duró más de un día en las 20 corridas independientes. En el ejemplo del seno (ver Sección 4.1.1) ECSID duró solo unos 6 segundos en las 20 corridas independientes. La máquina utilizada fue un Pentium III dual a 900Mhz.

Capítulo 5

Conclusiones

ECSID ha sido capaz de modelar sistemas dinámicos ya sea como una sola EDO o como un sistema de EDOs, además se puede modelar usando un método de ventana.

En este trabajo se ha mostrado experimentalmente que la programación genética y gene expression programming pueden ser utilizados para el modelado de sistemas; se ha mostrado que este procedimiento es automático y que el usuario no tiene que proporcionar ninguna estructura del modelo buscado. La única entrada del sistema es el conjunto de terminales, de funciones y las observaciones del sistema.

ECSID ha sido capaz de modelar sistemas lineales variantes e invariantes en el tiempo. En toda la bibliografía revisada hasta el momento no se ha encontrado ningún sistema que modele a estos sistemas lineales. Los sistemas lineales son de gran interés para los ingenieros dado que una gran variedad de sistemas son modelados con ecuaciones lineales.

ECSID no solo ha sido capaz de modelar sistemas, sino además, en casi todos los ejemplos presentados en este trabajo, se han encontrado modelos que le pueden dar al ingeniero una idea de la dinámica que gobierna al sistema.

ECSID no solo ha sido capaz de encontrar modelos simulados. Se hicieron tres experimentos reales. El primero, de un motor de corriente directa el cual encontró el modelo satisfactoriamente; el segundo, la serie de tiempo de manchas solares; el último, la serie de tiempo del viento en la cual se obtuvo un mejor modelo que el realizado por el procedimiento ARIMA.

Dentro de los trabajos relacionados existe la tendencia de usar a PG para obtener la estructura del sistema y usar otro método para optimizar sus parámetros. Dentro de todos los experimentos realizados no se ha encontrado la necesidad de utilizar un método

externo para optimizar parámetros dado que tanto PG como GEP han encontrado modelos cuyos coeficientes de correlación son superiores a 0.9 y en algunos casos han encontrado el mismo modelo que la ecuación buscada.

5.1. Trabajo Futuro

ECSID ha probado ser eficaz en la identificación de sistemas, pero necesita ser mejorado. A continuación se presenta una lista del trabajo futuro.

- Explorar las diferencias entre PG y GEP para identificar cual se adecua más al problema. Se ha encontrado que PG requiere de menos esfuerzo computacional que GEP pero los experimentos no han sido suficientes como para poder generalizar esta suposición.
- Mejorar la usabilidad de ECSID para que persona ajenas al área de computación lo puedan utilizar de manera intuitiva.
- Hacer a ECSID multiobjetivo. Aquí se propone optimizar dos objetivos. El primer objetivo sería encontrar el mejor modelo y el segundo objetivo sería que este modelo fuera mínimo.
- Investigar métodos paralelos para reducir el tiempo de modelado y con un mejor modelo.
- Implementar un método de reducción de conjunto de funciones. En el caso de serie de tiempo de varias variables el número de elementos que tiene el conjunto de funciones F es grande, por lo cual se propone crear un algoritmo que investigue cuales de estas funciones son útiles para poder reducir este conjunto, esto traería consigo una disminución en el espacio de búsqueda.

Apéndice A

Manual de Usuario de ECSID

ECSID es un paquete creado en lisp para la identificación de sistemas. Éste usa PG y GEP para encontrar el modelo de una serie de datos observados.

ECSID se puede obtener de [Graff05].

ECSID tiene cuatro módulos para encontrar el modelos. El primero y segundo utilizan GEP para encontrar un modelo mediante el método de ventana o una EDO. El tercero y cuarto utilizan GP para encontrar el modelo usando el método de ventana o EDO.

Para poder utilizar ECSID se requiere crear un archivo de configuración o de teclear el código en el intérprete de lisp. En el caso de que se cree un archivo de configuración este se debe de cargar al intérprete de lisp.

Las siguientes líneas indican donde se encuentra ECSID y además le indican al intérprete que debe de cargar el sistema.

```
(load "ev/gep.asdf")
(require :asdf)
(asdf:operate 'asdf:load-op :gep)
```

Una vez que se ha cargado ECSID en el intérprete de lisp es necesario crear la instancia de la clase principal de ECSID. Esto se hace de la siguiente manera.

```
(defvar *gep* (gep::make-gep :terminal-set '(t (0.0 100.0 1.0))
  :function-set '(,gep::*suma* ,gep::*resta* ,gep::*producto*
,gep::*division* ,gep::*seno* ,gep::*raiz*
,gep::*ln* ,gep::*negacion* ,gep::*coseno*)
  :objective "sunspot/datos.txt"
  :independent-variable "sunspot/tiempo.txt"))
```

El conjunto de terminales T está definido por *:terminal-set*. Éste es definido por el usuario. El primer elemento del conjunto de terminales debe de ser siempre el tiempo t . Para poder definir \mathfrak{R} se usa el siguiente formato (0.0 100.0 1.0), el cual significa crear un número aleatorio flotante entre 0.0 y 100.0 con paso de 1.0. El primer elemento de la lista es la cota inferior, el segundo elemento es la cota superior y el último elemento es el paso.

El último elemento de \mathfrak{R} solamente tiene sentido cuando se hace la siguiente especificación (01001) lo cual quiere decir que se cree un número aleatorio entero entre 0 y 100. Si se quisiera crear un elemento con un solo decimal se diría (0 100 10) y así hasta el número de decimales que se quisieran. Una recomendación es usar la primera formato (0.0 100.0 1.0) donde se crean número aleatorios de número flotante.

El conjunto de funciones F está definido por *function-set*. Las funciones definidas en ECSID son:

- **suma**.
- **resta**.
- **division**.
- **producto**.
- **exponente**.
- **seno**.
- **coseno**.
- **negacion**.
- **raiz**.
- **ln**.
- **producto – restringido**.

Los archivos *:objective* e *:independent-variable* son explicados en las secciones posteriores.

ECSID tiene dos métodos para crear los modelos a partir de un conjunto de observaciones del sistema. Éstas son creando un modelo de ventana o creando una sistema de EDOs.

A.1. Método de ventana

Este método fue presentado en la Sección 3. El método de ventana requiere de varios archivos de datos, estos son:

:objective En este archivo se pone la serie de tiempo.

:independent-variable Este archivo indica cual es el tiempo relacionado con la serie de tiempo.

Validación Es el archivo de validación.

Tiempo validación Éste indica el tiempo en el archivo de validación.

El archivo *:objective* tiene la siguiente sintaxis. Los siguientes datos fueron extraídos del archivo objetivo usado en la serie de tiempo de las manchas solares.

```
(47.0)
(63.0)
(60.0)
(39.0)
(28.0)
```

El siguiente fragmento es tomado del archivo *:independent-variable*.

```
(17)
(18)
(19)
(20)
(21)
```

La forma de especificar el archivo validación y el tiempo de validación es.

```
(setf (gep::validation *gep*) "sunspot/val.txt")
(setf (gep::validation-time *gep*) "sunspot/tiempo-1.txt")
```

Los siguientes datos fueron tomados del archivo validación de las manchas solares.

(104.5)

(66.6)

(68.9)

(38.0)

(34.5)

Un fragmento del archivo de tiempo relacionado con el archivo de validación se presenta a continuación. Hay que observar que la sintaxis de este archivo es diferente a los anteriores.

269

270

271

272

273

Una vez creada la instancia de **gep** se requieren ajustar algunos parámetros de ECSID. Estos son el tamaño de la ventana, si el modelo a evolucionar debe de contener a t o no, si el modelo debe de incluir los errores de predicción y el método para evolucionar el modelo (GEP o PG).

El tamaño de la ventana se especifica de la siguiente manera.

```
(setf (gep::size-window *gep*) 16)
```

Los errores de predicción y el incluir a t en el modelo se especifica como se muestra a continuación. En las siguientes líneas se está diciendo que el modelo no debe de incluir a t y que tampoco debe de incluir los errores de predicción en el conjunto F .

```
(setf (gep::ode-homogenea *gep*) t)
(setf (gep::prediction-errors *gep*) nil)
```

El método para evolucionar el modelo se especifica a continuación. Si se quiere usar programación genética se llama a la siguiente función.

```
(gep::set-gp-regression *gep*)
```

Si lo que se quiere es utilizar GEP se llama a la siguiente función.

```
(gep::set-gep-regression *gep*)
```

Una vez que todos los parámetros han sido puestos se requiere de llamar a la siguiente función para que el algoritmo comience.

```
(funcall (gep::gep-loop *gep*) *gep*)
```

Las siguientes dos funciones son para imprimir y graficar el mejor modelo obtenido por ECSID. Las gráficas se guardan con el nombre de *grafica-0.txt* y *grafica-validation.txt*.

```
(funcall (gep::print-best-of-all *gep*) *gep*)
(funcall (gep::graph-best-of-all *gep*) *gep*)
```

A.2. Método de EDOs

Este método fue presentado en la Sección 3.

Este método requiere solo de dos archivos de datos, los cuales son *:objective* y *:independent-variable*.

El formato de *:objective* es igual al presentado en la sección A.1. El formato de *:independent-variable* se presenta a continuación. Este es un fragmento del problema de masa resorte amortiguador.

```
0.3
0.6
0.90000004
1.2
1.5
```

Para usar este método se requiere de una modificación en el conjunto de terminales T . En el método de ventana, el conjunto T solo contiene a t y a \mathfrak{R} . Para este método el conjunto T debe de contener además las variables de estado. A continuación se muestra el código para crear la instancia de **gep**.

```
(defvar *gep* (gep::make-gep :terminal-set '(t x0 x1 (0.0 100.0 1.0))
  :function-set '(,gep::*suma* ,gep::*resta*
,gep::*producto* ,gep::*division*)
  :objective "resorte-amortiguador/datos.txt"
  :independent-variable "resorte-amortiguador/tiempo.txt"))
```

Se puede observar que el conjunto T es diferente y que contiene a las variables x_0 y x_1 , éstas representan x y x' respectivamente.

Una vez creada la instancia de **gep** se requiere de configurar algunos parámetros

como son: Las condiciones iniciales, el orden del sistema, si el modelo debe de contener a t o no y el método que se desea utilizar para encontrar el modelo.

El orden del sistema y las condiciones iniciales se especifican de la siguiente manera. En el siguiente código se especifican las siguientes condiciones iniciales $x(0) = 1$ y $x'(0) = 0$.

```
(setf (ode-order *gep*) 2)
(setf (gep::ode-t0 *gep*) '(0))
(setf (gep::ode-initial-values *gep*) '((1 0)))
```

La forma de especificar si el modelo debe de contener t o no es de la siguiente manera.

```
(setf (ode-homogenea *gep*) t)
```

Una vez que ya se ha especificado todos los parámetros solo falta decir cual método se desea utilizar, llamar a la función que inicia el proceso, imprimir y graficar los mejores individuos.

Si se quiere utilizar PG se llama a la siguiente función.

```
(gep::set-gp-ode *gep*)
```

De lo contrario si se quiere utilizar GEP se llama a la siguiente función.

```
(gep::set-gp-ode *gep*)
```

Para finalizar se llama a la función que inicia el proceso y las que imprimen y grafican el mejor individuo obtenido por ECSID.

```
(funcall (gep::gep-loop *gep*) *gep*)
(funcall (gep::print-best-of-all *gep*) *gep*)
(funcall (gep::graph-best-of-all *gep*) *gep*)
```

Referencias

- [Aguilar01] Aguilar, J. y Cerrada, M. Fuzzy classifier system and genetic programming on system identification problems. *En* L. Spector, E. D. Goodman, A. Wu, W. B. Langdon, H.-M. Voigt, M. Gen, S. Sen, M. Dorigo, S. Pezeshk, M. H. Garzon, y E. Burke, eds., *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2001)*, págs. 1245–1251. Morgan Kaufmann, San Francisco, California, USA, 7-11 jul. 2001. ISBN 1-55860-774-9.
- [Anderson94] Anderson, K. R. y Rettig, D. Performing lisp analysis of the fannkuch benchmark. *SIGPLAN Lisp Pointers*, VII(4):2–12, 1994. ISSN 1045-3563. doi:<http://doi.acm.org/10.1145/382109.382124>.
- [Angeline97] Angeline, P. J. y Fogel, D. B. An evolutionary program for the identification of dynamical systems. *En* S. Rogers, ed., *Application and Science of Artificial Neural Networks III*, tomo 3077, págs. 409–417. 1997.
- [Angeline98] Angeline, P. J. Evolving predictors for chaotic time series. *En* S. Rogers, D. Fogel, J. Bezdek, y B. Bosacchi, eds., *Proceedings of SPIE: Application and Science of Computational Intelligence*, tomo 3390, págs. 170–80. 1998.
- [Arkov00] Arkov, V., Evans, C., Fleming, P. J., Hill, D. C., Norton, J. P., Pratt, I., Rees, D., y Rodriguez-Vazquez, K. System identification strategies applied to aircraft gas turbine engines. *Annual Reviews in Control*, 24(1):67–81, 2000.

- [Babovic00] Babovic, V. y Keijzer, M. Evolutionary algorithms approach to induction of differential equations. International Conference on Hydroinformatics, 2000.
- [Banzhaf96] Banzhaf, W., Francone, F. D., y Nordin, P. The effect of extensive use of the mutation operator on generalization in genetic programming using sparse data sets. *En PPSN IV: Proceedings of the 4th International Conference on Parallel Problem Solving from Nature*, págs. 300–309. Springer-Verlag, London, UK, 1996. ISBN 3-540-61723-X.
- [Blickle95] Blickle, T. y Thiele, L. A comparison of selection schemes used in genetic algorithms. Inf. téc., Computer Engineering and Communication Networks Lab, 1995.
- [Cao99a] Cao, H., Kang, L., y Chen, Y. Evolutionary modeling of ordinary differential equations for dynamic systems. *En W. Banzhaf, J. Daida, A. E. Eiben, M. H. Garzon, V. Honavar, M. Jakiela, y R. E. Smith, eds., Proceedings of the Genetic and Evolutionary Computation Conference*, tomo 2, págs. 959–965. Morgan Kaufmann, Orlando, Florida, USA, 13-17 jul. 1999. ISBN 1-55860-611-4.
- [Cao99b] Cao, H., Yu, J., Kang, L., Chen, Y., y Chen, Y. The kinetic evolutionary modeling of complex systems of chemical reactions. *Computers & Chemistry*, 23(2):143–152, 30 mar. 1999. doi:doi:10.1016/S0097-8485(99)00005-4.
- [Cao00a] Cao, H., Kang, L., Chen, Y., y Yu, J. Evolutionary modeling of systems of ordinary differential equations with genetic programming. *Genetic Programming and Evolvable Machines*, 1(4):309–337, oct. 2000. ISSN 1389-2576. doi:doi:10.1023/A:1010013106294.
- [Cao00b] Cao, H.-Q., Kang, L.-S., Guo, T., Chen, Y.-P., y de Garis, H. A two-level hybrid evolutionary algorithm for modeling one-dimensional dynamic systems by higher-order ODE models. *IEEE Transactions*

- on Systems, Man and Cybernetics – Part B: Cybernetics*, 40(2):351–357, abr. 2000. ISSN 1083-4419.
- [Chen01] Chen, Y. *Hybrid Soft Computing Approach to Identification and Control of Nonlinear Systems*. Tesis Doctoral, Department of Computer Science, Kumamoto University, Japan, mar. 2001.
- [CMU80] 1980. [Http://www.cons.org/cmuc/](http://www.cons.org/cmuc/).
- [Duan01] Duan, M. y Povinelli, R. Estimating time series predictability using genetic programming. En C. H. Dagli, ed., *Intelligent Engineering Systems Through Artificial Neural Networks (ANNIE 2001)*, págs. 215–220. St. Louis, Missouri, USA, 4-7 nov. 2001.
- [Evans01] Evans, C., Fleming, P. J., Hill, D. C., Norton, J. P., Pratt, I., Rees, D., y Rodriguez-Vazquez, K. Application of system identification techniques to aircraft gas turbine engines. *Control Engineering Practice*, 9(2):135–148, 2001.
- [Ferreira01] Ferreira, C. Gene expression programming: A new adaptive algorithm for solving problems. En *Complex Systems*, 2, págs. 87–129. 2001.
- [Graff05] Graff, M. y Flores, J. J., January 2005. [Http://sourceforge.net/projects/ecs/](http://sourceforge.net/projects/ecs/).
- [Gray96a] Gray, G., Li, Y., Murray-Smith, D., y Sharman, K. Structural system identification using genetic programming and ablock diagram oriented simulation tool. *Electronics Letters, IEEE*, 1996.
- [Gray96b] Gray, G. J., Murray-Smith, D. J., Li, Y., y Sharman, K. C. Nonlinear model structure identification using genetic programming. En J. R. Koza, ed., *Late Breaking Papers at the Genetic Programming 1996 Conference Stanford University July 28-31, 1996*, págs. 32–37. Stanford Bookstore, Stanford University, CA, USA, 28–31 1996. ISBN 0-18-201031-7.
- URL citeseer.ist.psu.edu/article/gray96nonlinear.html

- [Hinchliffe01] Hinchliffe, M. *Dynamic Modelling Using Genetic Programming*. Thesis Doctoral, University of Newcastle upon Tyne, 2001.
- [Holland92] Holland, J. H. *Adaptation in Natural and Artificial Systems 2nd ed.* University of Michigan Press, 1992.
- [Hui03] Hui, A. Using genetic programming to perform time-series forecasting of stock prices. En J. R. Koza, ed., *Genetic Algorithms and Genetic Programming at Stanford 2003*, págs. 83–90. Stanford Bookstore, Stanford, California, 94305-3079 USA, 4 dic. 2003.
URL <http://www.genetic-programming.org/sp2003/Hui.pdf>
- [Jäske96] Jäske, H. One-step-ahead prediction of sunspots with genetic programming. En J. T. Alander, ed., *Proceedings of the Second Nordic Workshop on Genetic Algorithms and their Applications (2NWGA)*, Proceedings of the University of Vaasa, Nro. 13, págs. 79–88. University of Vaasa, Vaasa (Finland), 19.-23. ago. 1996.
- [Jie04] Jie, Z., Changjie, T., Chuan, L., Anlong, C., y Chang'an, Y. Time series prediction based on gene expression programming. International Conference for Web Information (WAIM04), Springer Verlag, 2004.
- [Kaboudan00] Kaboudan, M. A. Evaluation of forecasts produced by genetically evolved models. En *Computing in Economics and Finance*. Universitat Pompeu Fabra, Barcelona, Spain, 6-8 jul. 2000.
- [Kaboudan03] Kaboudan, M. A. Genetic programming software to forecast time series. En *Computing in Economics and Finance*. University of Washington, Seattle, USA, 11-13 jul. 2003.
- [Koza92] Koza, J. R. *Genetic Programming: On the Programming of Computers by Means of Natural Selection (Complex Adaptive Systems)*. The MIT Press, 1992.
- [Kristinsson92] Kristinsson, K. y Dumont, G. System identification and control

- using genetic algorithms. *IEEE Transactions on Systems, Man and Cybernetics*, 22:1033–1046, 1992.
- [Leontaritis85] Leontaritis, I. J. y Billings, S. A. Input-output parametric models for nonlinear systems. *Int. J. Control*, 1985.
- [Leung02] Leung, H. y Varadan, V. System modeling and design using genetic programming. Computer Society, IEEE, 2002. ICCI 2002.
- [Ljung99] Ljung, L. *System Identification Theory for the User*. Ptr Prentice Hall and System Sciences Series, 1999.
- [Luke97] Luke, S. y Spector, L. A comparison of crossover and mutation in genetic programming. Second Annual Conference on Genetic Programming, 1997.
- [López99] López, A. M., López, H., Ojea, G., y González, V. System inductive modeling using genetic programming with a geneticalgorithm for parameter adjustment. *Emerging Technologies and Factory Automation*, IEEE, 1999.
- [Makridakis92] Makridakis, S., Whellwright, S. C., y Hyndamn, R. J. *Forecasting Methods and Applications*. John Wiley & Sons, Inc., 3^a ed^{ón}., 1992.
- [Mat92] The MathWorks, Inc., Mass. *SIMULINK User's guide*, 1992.
- [Min] Minitab. <http://www.minitab.com/>.
- [Morrison91] Morrison, F. *The Art of Modelling Dynamic Systems*. Wiley, 1991.
- [Press92] Press, W., Teukolsky, S., Vetterling, W., y Flannery, B. *Numerical Recipes in C*. Cambridge University Press, 2^a ed^{ón}., 1992.
- [Rodriguez-Vazquez01] Rodriguez-Vazquez, K. Genetic programming in time series modeling: An application to meteorological data. *En Proceedings of the 2001 Congress on Evolutionary Computation CEC2001*, págs. 261–266. IEEE Press, COEX, World Trade Center, 159 Samseong-dong, Gangnam-gu, Seoul, Korea, 27-30 2001. ISBN 0-7803-6658-1.

- [Rodríguez-Vázquez98] Rodríguez-Vázquez, K. y Fleming, P. Multiobjective genetic programming for gas turbine engine model identification. International Conference on CONTROL, IEE, 1998.
- [Rodríguez-Vázquez04] Rodríguez-Vázquez, K., Fonseca, C. M., y P. J. Fleming, M. Identifying the structure of nonlinear dynamic systems using multiobjective genetic programming. *IEEE Transactions on Systems, Man and Cybernetics – Part A: Systems and Humans*, 34(4), 2004.
- [Sakamoto01] Sakamoto, E. y Iba, H. Inferring a system of differential equations for a gene regulatory network by using genetic programming. *Evolutionary Computation*, 2001.
- [Sun96] Java. <http://java.sun.com>, 1996.
- [Szpiro97] Szpiro, G. G. Forecasting chaotic time series with genetic algorithms. *Physical Review E.*, 1997.
- [Thierens94] Thierens, D. y Goldberg, D. E. Convergence models of genetic algorithm selection schemes. *En PPSN III: Proceedings of the International Conference on Evolutionary Computation. The Third Conference on Parallel Problem Solving from Nature*, págs. 119–129. Springer-Verlag, London, UK, 1994. ISBN 3-540-58484-6.
- [Varadan02] Varadan, V. y Leung, H. Chaotic system reconstruction from noisy time series measurements using improved least squares genetic programming. *Circuits and Systems, IEEE*, 2002. ISCAS 2002.
- [Weinbrenner97] Weinbrenner, T. Genetic programming techniques applied to measurement data. Diploma Thesis, 1997.
URL citeseer.ist.psu.edu/weinbrenner97genetic.html
- [Whitley89] Whitley, D. The genitor algorithm and selection pressure: why rank-based allocation of reproductive trials is best. *En Proceedings of the third international conference on Genetic algorithms*, págs. 116–121. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1989. ISBN 1-55860-006-3.

- [Winkler04] Winkler, S., Affenzeller, M., y Wagner, S. Identifying nonlinear model structures using genetic programming techniques. págs. 689–694. Austrian Society for Cybernetic Studies, 2004.