

Lessons Learned in Modeling Dynamic Systems using Genetic Programming

Juan J. Flores and Mario Graff

División de Estudios de Posgrado
Facultad de Ingeniería Eléctrica
Universidad Michoacana
Mexico

Abstract. This article describes a set of experiences in modeling dynamic systems using Genetic Programming and Gene Expression Programming. We have made experiments in linear systems, non-linear systems and times series. The experiments in linear systems include linear pendulum, coupled mass-spring, electrical circuits, etc. In non-linear systems we have modeled the Van der Pol Equation, non-linear pendulum with friction, etc. The models have been represented as an ordinary differential equation, system of ordinary differential equations, and using a time series approach.

1 Introduction

Modeling a dynamic system is a process that generally has been made by an engineer. This work describes the ability of Genetic Programming (GP) [6] and Gene Expression Programming (GEP) [2] to model a dynamic system.

It is presented a set of examples where GP and GEP have found good models. Examples included in this work are: a mass spring shock absorber, a coupled mass-spring system, the Van der Pol Equation, a non-linear pendulum with friction, and a wind time series.

This article shows three different ways of representing a model in GP and GEP. It is used an ordinary differential equation, a system of ordinary differential equations and a time series approach.

Section 2 is an introduction to system identification. Section 3 shows the representation of a dynamic system in GP. Section 4 is the experiments scenarios. Section 5 presents examples in of linear systems. Section 6 shows examples of non-linear systems. Section 7 presents an example of time series. Section 8 presents a summary of the obtained results. Section 9 presents the conclusions.

2 System Identification

System identification (SID) is the process of deriving a mathematical model from the internal dynamics of a system of observations of its outputs. Modeling is the essential first step in a variety of engineering problems. For example, if an

engineer is going to control a dc-motor, first he needs to model it. The model is made by applying a given input to the system and observing its behavior. From this behavior, a mathematical model is built and tested against the dc-motor. Based on the obtained mathematical model, the controller is made.

Linear system identification methods have been widely studied (see [7]). However, these methods involve a complicated process that usually can only be followed by an expert. Nonlinear system identification remains a difficult task, because frequently there is not enough information about the system (i.e. the structure system's is unknown).

3 Representation of Dynamic Systems

In order to make the experiments, a computer program called ECSID (Evolutionary Computation based System Identification) was implemented. ECSID uses GP or GEP to build a mathematical model from observed data. ECSID has three different ways to represent a model; it can represent the model as an ODE (Ordinary Differential Equation), a System of ODEs, or using a time series approach.

All the models represented as ODE have the general form of Equation 1.

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

ECSID evolves only the right part of Equation 1. The order of the system is determined by the order of the highest order element whose coefficient is not zero. Figure 1 shows an ODE represented in ECSID. The ode represented in the Figure is $\frac{d^2x}{dt^2} = 7\frac{dx}{dt} + 10x + 12$. We can observe that this system is a second order system, because it has a first order element.

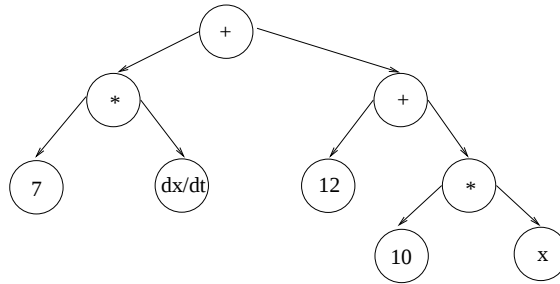


Fig. 1. ODE represented in ECSID.

In order to get the behavior of the system, Equation 1 is integrated. First this equation is transformed to Equation 2 and a 4th order Runge-Kutta method is used to integrate the latter.

$$\begin{cases} y'_1 &= y_2 \\ y'_2 &= y_3 \\ \vdots & \\ y_n &= f(t, y_1, y_2, \dots, y_n) \end{cases} \quad (2)$$

Equation 2 is formed by replacing the following variables $y_1 = y, y_2 = y', y_3 = y'', \dots, y_n = y^{(n-1)}$.

ECSID can evolve a system of ODEs. It uses a multi-gene chromosome to represent the system, where each gene has the same structure of the right part of Equation 1. An operation similar to crossover was implemented; it is called gene-recombination. Gene-recombination receives two individuals and randomly chooses one gene from each individual and swaps them.

ECSID can also represent its models using a time series approach, where the output is modeled as a function of past values of the input(s) and output(s). Equation 3 shows this approach where k is the current time and τ is the maximum time shift.

$$y_k = f(y_{1,k-1}, \dots, y_{1,k-\tau}, y_{n,k-1}, \dots, y_{n,k-\tau}) \quad (3)$$

Another characteristic of ECSID is that it can evolve linear systems. In order to evolve linear systems we need to disable product and division. Product and division can receive any s-expression and multiply or divide these s-expressions. For example, product can give an individual whose form is $\frac{dx}{dt} = x^2$ which is non-linear. An operation called “coefficient” is implemented to replace product. Coefficient receives any s-expression and a constant, and multiplies the constant with the s-expression. It was also implemented a function that punishes individuals that are non-linear. One way of punishing a non-linear individual is to set its fitness to its original fitness plus the average fitness of the population.

4 Experiments Scenarios

In order to compare experiments from different domains, we use the correlation coefficient (Equation 4). The correlation coefficient gives a number between -1 and 1 where 1 means that the curves are equal.

$$r = \frac{n \sum xy - \sum x \sum y}{\sqrt{[n \sum x^2 - (\sum x)^2][n \sum y^2 - (\sum y)^2]}} \quad (4)$$

ECSID uses the standard evolutionary computation procedure, random initial population, fitness proportional selection and elitism. The fitness function is the absolute difference of the errors $\sum |e|$. Each experiment had a population of 500 individuals and was run for 500 generations. The termination criteria is met when the correlation coefficient is ≥ 0.99 . All models presented here are selected from the best individuals of 20 independent runs. Table 1 shows the parameters used in ECSID.

Genetic Operator	Probability
Mutation	0.2
Crossover	0.8
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

Table 1. Genetic operators' parameters

5 Linear Systems

There are a lot of problem that are modeled as linear systems. E.g. Electrical Circuits, Electrical Machines, Mechanical Systems, etc. If it is known that the system is linear, it is important that ECSID finds a linear model. A system is linear if it can be represented as Equation 5.

$$a_n(t)\frac{d^n x}{dt^n} + \dots + a_1(t)\frac{dx}{dt} + a_0(t)x = g(t) \quad (5)$$

In order to obtain a linear model, we use two methods: the first one uses the operation “coefficient” and disables the operations “product” and “division”; the second one uses a function that punishes those individuals that are non-linear. The function set used to obtain the following two examples is $\{+, -, *, /\}$ and the terminal set is $\{x, \frac{dx}{dt}, \mathfrak{R}\}$. The symbol $\mathfrak{R}$ means a random floating point number in the range $[0, 100]$. In the following example a function that punishes those individuals that are non-linear is used.

The first example is a mass spring shock absorber. Equation 6 shows its model. Its initial conditions are $x(0) = 1$ and $x'(0) = 0$.

$$\frac{d^2 x}{dt^2} = -\frac{dx}{dt} - 3x \quad (6)$$

GP and GEP evolved the same equation that Equation 6. Figure 2 shows the graphics of Equation 6.

In the second example we model a coupled mass-spring system. Equation 7 shows this system. The initial conditions for the system are $x(0) = 1, x'(0) = 0, y' = 2, y'(0) = 0$.

$$\begin{aligned} \frac{d^2 x}{dt^2} &= -5x + 2y \\ \frac{d^2 y}{dt^2} &= 2x - 2y \end{aligned} \quad (7)$$

In this example GEP and GP obtained the same model. Equation 8 shows this model. It can be observed that the obtained model is linear.

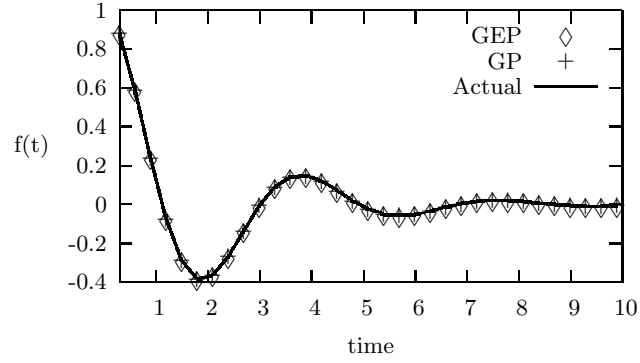


Fig. 2. Mass Spring Shock-absorber

$$\begin{aligned}\frac{d^2x}{dt^2} &= -x \\ \frac{d^2y}{dt^2} &= -y\end{aligned}\tag{8}$$

Figure 3 shows the behavior exhibited by the obtained models and the real system (there are four graphics in the figure because each model has two equations).

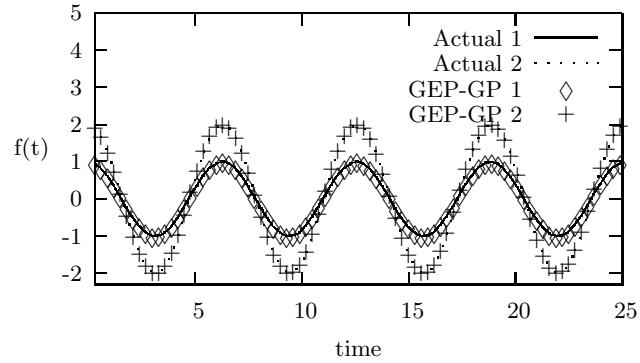


Fig. 3. Mass springs coupled.

ECSID has obtained models for a linear pendulum, an electric circuit (two branches), and a DC machine; all of models obtained exhibit a similar behavior that the ones presented in this section. All the models that have been obtained

have a similar structure than the real models. Therefore this methodology builds good models and also the models obtained can be understood by an engineer.

Hinchliffe [4] states that it is unlikely to evolve models that can provide any insight into the underlying physical processes of a dynamic system. In all experiments done so far we have found models that have a similar structure than the real models. Therefore this procedure can produce not only good models but also models that provide some insight into the underlying physical processes.

The results obtained using both methods (“coefficient” and punish function) were comparable.

We have not found any work that can evolve linear systems.

6 Non-linear Systems

Non-linear systems are those systems that cannot be represented using Equation 5. These kind of systems are very important; e.g. mechanical systems, chemical processes, electrical circuits, etc.

This example evolves the Van der Pol Equation, which is shown in Eq. 9. The experiment was done with initial conditions $x(0) = 1.5$ and $x'(0) = 0$. The function set is $\{+, -, *, /\}$ and the terminal set is $\{\frac{dx}{dt}, x, \Re\}$

$$\frac{d^2x}{dt} = (1 - x^2) \frac{dx}{dt} - x \quad (9)$$

Both GEP and GP obtained the same model that Equation 9. Figure 4 shows the real model and the obtained by GEP and GP.

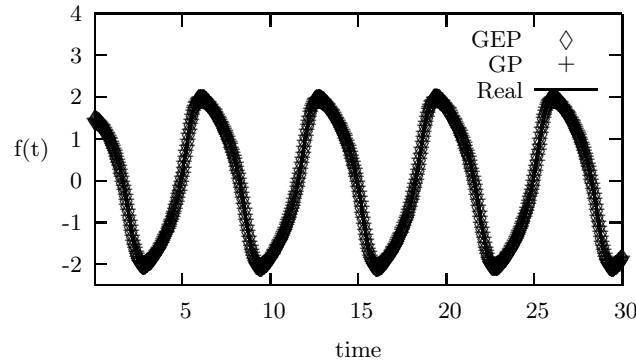


Fig. 4. Van der Pol Equation

The next example is a non-linear pendulum with friction; equation 10 non-linear pendulum shows the model. The initial conditions for the experiment are $\theta(0) = 1$ and $\theta'(0) = 0$, the function set is $\{+, -, *, /, \sin, \cos, \exp\}$ and the terminal set is $\{\frac{d\theta}{dt}, \theta, \Re\}$.

$$\frac{d^2\theta}{dt^2} = -2\frac{d\theta}{dt} - 19.6\sin(\theta) \quad (10)$$

Equations 11 and 12 show the result using GEP and GP, respectively. It is observed that the models obtained are linear but it is not implemented any restriction about non-linearity. Figure 5 shows the behavior exhibited by those models.

$$\frac{d^2\theta}{dt^2} = -2\theta\frac{d\theta}{dt} - 2\frac{d\theta}{dt} - 20\theta \quad (11)$$

$$\frac{d^2\theta}{dt^2} = -2.3048492976\frac{d\theta}{dt} - 19.3437292976\theta \quad (12)$$

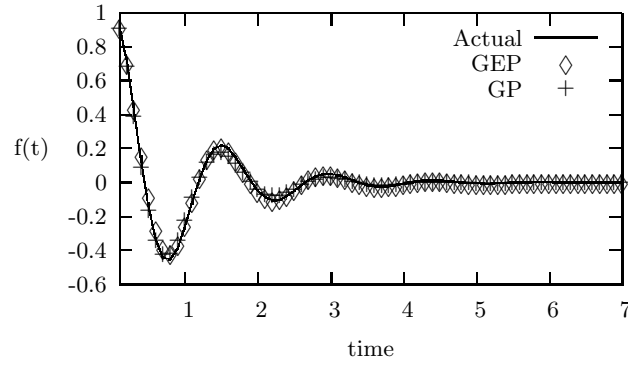


Fig. 5. Pendulum with friction.

Gray et al [3], Weinbrenner [11] and Cao et al [1] use a similar procedure to evolve non-linear systems. All of them represent the system as an ODE. Their procedure uses GP to find the structure of the system and another one to find the parameters of the system.

In all the experiments done we have not found evidence that it is necessary to use a different procedure to optimize the parameters of the ODE. We have found that GP does a good job in finding a model that behaves close to the observed data. Furthermore all the models obtained are simple enough to be understood by an engineer.

7 Time Series Prediction

This example models a wind time series using a slide window prediction method. In order to compare the obtained model, we developed an ARIMA model (see

[8]) using the standard ARIMA procedure. This model was obtained using the software Minitab [9]. Equation 13 shows the model obtained using the ARIMA procedure. Where e means the prediction errors.

$$\begin{aligned} f(n) = & f(n-1) + f(n-2) - f(n-13) - 0.997e(n-1) \\ & - 0.7976e(n-12) + 0.7956e(n-13) \end{aligned} \quad (13)$$

The function set is $\{+, -, *, /, \sin, \cos, \log\}$ and the terminal set is $\{y_{n-1}, \dots, y_{n-16}, \Re\}$

Equation 14 shows the best model. Figure 6 shows the time series and the obtained model.

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

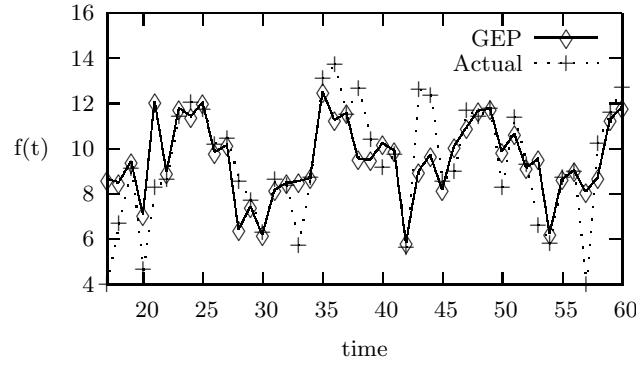


Fig. 6. Wind Time Series

The correlation coefficient of Equation 13 is $r = 0.51933$ meanwhile for Equation 14 is $r = 0.7313$. It is clear that GP found a better model than the one obtained by the ARIMA procedure.

Lie et al [5], and Szpiro [10] use a slide window prediction method (Equation 3) to model a time series. Hinchliffe [4] used this method to find a model of a dynamic system.

8 Results

Table 2 presents a summary of the results of the experiments. ECSID found good models in all the the experiments represented as ODEs and slide window regressors. In the last experiment the model found is no as good as the others but the observed data is more complex.

Method	Problem	r
GEP	Mass Spring Shock-absorber (Eq. 6)	1
GP	Mass Spring Shock-absorber (Eq. 6)	1
GEP	Coupled mass-spring (Eq. 8)	1
GP	Coupled mass-spring (Eq. 8)	1
GEP	Van der Pol (Eq. 9)	1
GP	Van der Pol (Eq. 9)	1
GEP	Pendulum with friction (Eq. 11)	0.9982
GP	Pendulum with friction (Eq. 12)	0.9969
GEP	Wind Prediction (Eq. 14)	0.7313

Table 2. Results

Table 3 shows the computational resources needed to find a good model with a probability of 0.99 (see [2] Chapter 8). This information was acquired experimentally. Each experiment was run 20 times. $I(M, i, z)$ is the number of individuals that needs to be processed in order to obtain a model with a correlation coefficient ≥ 0.99 . The last column is maximum number of generations needed to set a correlation coefficient ≥ 0.99 .

Column $I(M, i, z)$ and *Gen.* give us an idea of the problem complexity and the computational resources needed by GP or GEP. It is observed that the most complex problem is the “Pendulum with friction” for both cases and the simplest experiment is the “Coupled mass-spring” again for both cases.

From this table we can say that GP generally needs to process less individuals than GEP, therefore it also needs less generations. Ferreira [2] states that GEP is better than GP. We have not found any evidence that supports her assertion. Another characteristic that can be inferred from this table is the number of generations needed, we can say that 100 generations are acceptable.

The wind time series model is not included in this table because there is not information to calculate $I(M, i, z)$, instead an ARIMA model was used to compare the GP model.

9 Conclusions

In this work we have found experimentally that GP is better than GEP because it processes less individuals to obtain comparable results. ECSID has found good models and those models can provide an insight into the underlying physical process of a dynamic system.

Method	Problem	$I(M, i, z)$	Gen.
GEP	Mass Spring Shock-absorber	110, 162	77
GP	Mass Spring Shock-absorber	9, 000	18
GEP	Coupled mass-spring	41, 275	4
GP	Coupled mass-spring	42, 760	8
GEP	Van der Pol	2, 872, 995	64
GP	Van der Pol	197, 769	37
GEP	Pendulum with friction	262, 011	137
GP	Pendulum with friction	48, 000	48

Table 3. Computational effort

We provide the reader with a set of experiences that show evidence about what results a scientist or engineer might expect from using GP for modeling Dynamic Systems.

ECSID is a free software, written in Lisp, and can be downloaded from <http://sf.net/projects/ecsidi>.

References

1. H. Cao, L. Kang, Y. Chen, and J. Yu. Evolutionary modeling of systems of ordinary differential equations with genetic programming. *Genetic Programming and Evolvable Machines*, 1(4):309–337, Oct. 2000.
2. C. Ferreira. Gene expression programming: A new adaptive algorithm for solving problems. In *Complex Systems*, number 2, pages 87–129, 2001.
3. G. J. Gray, D. J. Murray-Smith, Y. Li, and K. C. Sharman. Nonlinear model structure identification using genetic programming. In J. R. Koza, editor, *Late Breaking Papers at the Genetic Programming 1996 Conference Stanford University July 28-31, 1996*, pages 32–37, Stanford University, CA, USA, 28–31 1996. Stanford Bookstore.
4. M. Hinchliffe. *Dynamic Modelling Using Genetic Programming*. PhD thesis, University of Newcastle upon Tyne, 2001.
5. Z. Jie, T. Changjie, L. Chuan, C. Anlong, and Y. Chang'an. Time series prediction based on gene expression programming. International Conference for Web Information (WAIM04), Springer Verlag, 2004.
6. J. R. Koza. *Genetic Programming: On the Programming of Computers by Means of Natural Selection (Complex Adaptive Systems)*. The MIT Press, 1992.
7. L. S. Ljung. *System Identification: Theory for the User*. Prentice Hall, 1987.
8. S. Makridakis, S. C. Whellwright, and R. J. Hyndamn. *Forecasting Methods and Applications*. John Wiley & Sons, Inc., 3 edition, 1992.
9. Minitab. <http://www.minitab.com/>.
10. G. G. Szpiro. Forecasting chaotic time series with genetic algorithms. *Physical Review E*, 1997.
11. T. Weinbrenner. Genetic programming techniques applied to measurement data. Diploma Thesis, 1997.