Transactions on Information and Communications Technologies vol 16, © 1996 WIT Press, www.witpress.com, ISSN 1743-3517

An intelligent system for automated mathematical modelling and simulation of dynamical engineering systems

O. Castillo[a] & P. Melin[b]
[a]Dept. of Computer Science, Instituto Tecnologico de Tijuana,
EMail: ocastillo@besttj.tij.cetys.mx
[b]School of Engineering, CETYS University,
EMail: emelin@besttj.tij.cetys.mx
P.O. Box 4207 Chula Vista CA 91909, USA

Abstract

We describe in this paper a computer program for Automated Mathematical Modelling (AMM) and Automated Simulation (AS) of dynamical engineering systems using Artificial Intelligence (AI) techniques. This computer program is an implementation of a new method for AMM using Fuzzy Logic techniques and Fractal Theory, and a new method for AS using Expert Systems technology. Our new method for AMM consists of three main parts: Time Series Analysis, Developing a set of Admissible Models and Selecting the "Best" model. Our method for Time Series Analysis consists in the use of the fractal dimension of a set of points as a measure of the geometrical complexity of the time series. Our method for developing a set of admissible models is based on the use of Fuzzy Logic techniques to simulate the reasoning process of the human experts in mathematical modelling of engineering systems. The selection of the "best" model for the engineering systems is done using heuristics from the experts and statistical calculations. The simulation of the "best" model can be done by using a new method developed by the authors, that enables automated computer exploration of all the dynamical behaviors of the engineering system. Given a mathematical model for a specific Dynamical Engineering System (DES), this method will automatically select (using a rule base) the "best" set of parameter values to perform numerical simulations of the system. This simulations will in turn enable the identification of all the dynamical behaviors of the engineering system.

1 Introduction

We describe in this paper a new method to perform automated mathematical modelling for DES using Fuzzy Logic techniques, Dynamical Systems and Fractal theory. The idea of using Dynamical Systems Theory and Fractal Theory as alternative approaches for modelling can be justified if we consider that traditional statistical methods only have limited success in real world engineering applications, and this is mainly because engineering problems show very complicated dynamics in time. Traditional statistical methods assume that the erratic behavior of a time series is mainly due to a external random error (that can not be explained). However, a Dynamical Systems approach, using non-linear mathematical models, can explain this erratic behavior because "chaos" is an intrinsic part of this type of models. It is a well known fact from Dynamical Systems, see Devaney [8], that even very simple non-linear mathematical models can exhibit the behavior known as "chaos" for certain parameter values, and therefore are good candidates to use as equations for modelling. Fractal Theory, see Mandelbrot [9], also offers a way to explain the erratic behavior of a time series, but the method is geometrical in the sense that the fractal dimension is used to describe the complexity of the distribution of the data points.

We describe a prototype implementation of our new method for Automated Mathematical Modelling (AMM) as a computer program written in the PROLOG programming language. This computer program can be considered an intelligent system for the domain of DES because it uses Artificial Intelligence (AI) techniques to obtain the "best' mathematical model for a given engineering problem. The use of AI techniques is to achieve the goal of automated modelling of engineering problems by simulating (in the computer) how human experts in this domain obtain the "best" model for a given problem. Given an engineering time series the intelligent system develops mathematical models based on the geometry of the data. The method for AMM consists of three main parts: Time Series Analysis, Developing a Set of Admissible Models and Selecting the Best Model. First, the computer program uses the fractal dimension to classify the components of the time series over a set of qualitative values, then the program uses this information to decide (using a fuzzy rule base) which dynamical models are the most appropriate for the data, and finally the program decides which model is the "best' one using heuristics and statistical calculations. In a prior prototype intelligent system developed by the authors, see Castillo & Melin [6], the method used didn't consider using the fractal dimension as a measure of classification and also didn't consider a fuzzy rule base for the simulation of the decision process. The use of Fuzzy logic in engineering applications has been well recognized and many applications have been developed. In this case we came to the conclusion that the best way to convey the information of engineering modelling problems was using fuzzy sets

[2]. Also, we think that the best way to reason with uncertainty in this case is using Fuzzy Logic.

We also show in this paper how Artificial Intelligence (AI) techniques can be used to make the task of computer simulation of non-linear dynamical systems more easy and less time consuming. The task of computer simulation of the non-linear mathematical model of a particular engineering system, is dependent on the appropriate selection of the parameter values to perform the best exploration possible of the dynamical behaviors of the system. The problem of selecting the "best" set of parameter values for the exploration is not a simple one, since it requires using heuristics to develop a set of admissible parameters AP, considered to be appropriate for the given mathematical model, and then using heuristics and statistical calculations to make the final decision of which parameter values are considered to be the "best" set BP. The idea in this paper is that, using AI methodology [3], the process of parameter value selection described above can be simulated by a computer program, in this way obtaining a useful software tool for the simulation of non-linear dynamical engineering systems. The computer program can be considered an intelligent system, because it shows some intelligent features in simulating the real human experts in the process of parameter value selection described above.

We have developed before several prototype intelligent systems, see Castillo & Melin [4, 7], for the exploration of dynamical systems, however they didn't consider the task of simulating real engineering systems. This is very important in the applications, because if we can simulate an engineering system very well, we can predict its future behavior very accurately or we can simulate different scenarios to see how we can control the future dynamical behavior of the system. Controlling the dynamical behavior of a real engineering system is the ultimate goal for many engineers, then a computer program like ours that can help in this regard can be a very useful tool for them.

2 Method for automated mathematical modelling

The problem of achieving automated mathematical modelling can be defined as follows:

Given: A data set (time series) with n data points, $D = \{d_1 ..., dm\}$ where $di \in R^n$, $i = 1,...,m$, $n = 1, 2...$.

Goal: From the data set D, discover automatically the "best" mathematical model for the time series.

This problem is not a simple one, because in theory there can be an infinite number of mathematical models that can be build for a given data set [12]. So the problem lies in knowing which models to try for a data set and then to select the "best" one. We can state the problem more formally in the following lines.

Let M be the space of mathematical models defined for a given data set D. Let $MA = \{M_1, ..., Mq\}$ be the set of admissible models that are considered to be appropriate for the geometry of the data set D. The problem is to find automatically the "best" model Mb for time series prediction.

We consider mathematical statistical models of the following form:

$$Y = F(X) + \varepsilon\ (0,\sigma)$$

where $\varepsilon\ (0,\sigma)$ represents a 0-mean Gaussian noise-process with standard deviation σ. F(X) is a polynomial equation in X, where the predictor variables are in the vector:

$$X = (X_1, X_2, ..., Xp)$$

We consider mathematical models as "dynamical systems" of the following form:

$$dY/dt = F(Y)$$

where Y is a vector of variables of the form: $Y = (Y_1, Y_2, ..., Yp)$ and F(Y) is a non-linear function of Y. Other kind of mathematical models are the discrete "dynamical systems" of the following form:

$$Y_t = F(X)$$

where $X = (Y_{t-1}, Y_{t-2}, ..., Y_{t-p})$ and F(X) is a non-linear function of X. Note that in this case we have deterministic models expressed as differential or difference equations.

The models for the statistical methods can be linear as well as non-linear equations. We show below some sample statistical models [10] that the intelligent system explores:

a) linear regression: $Y_t = a + bt$

b) quadratic regression: $Y_t = a + bt + ct^2$

c) logarithmic regression: $\ln Y_t = a + b\ln t$

d) autoregression: $Y_t = a + bY_{t-1}$

The mathematical models for continuous dynamical systems can be one-dimensional, two-dimensional or three-dimensional. We show below some sample models [11] that the intelligent system explores:

a) Logistic differential equation:

$$dY_1/dt = a\ Y_1(1-Y_1)$$

b) Lotka Volterra two dimensional:

$$dY_1/dt = aY_1 - bY_1Y_2$$
$$dY_2/dt = bY_1Y_2 - cY_2$$

c) Lorenz three dimensional:

$$dY_1/dt=aY_2-aY_1$$
$$dY_2/dt=-Y_1Y_3+bY_1-Y_2$$
$$dY_3/dt=Y_1Y_2-cY_3$$

The mathematical models for discrete dynamical systems can also be one, two or three dimensional. We show below some sample models [11] that the intelligent system explores:

a) Logistic difference equation:

$$Y_{t+1} = aY_t(1-Y_t)$$

b) Lotka Volterra two dimensional:

$$Y_{t+1} = aY_t - bY_tX_t$$
$$X_{t+1} = bY_tX_t - cX_t$$

c) Henon map two dimensional:

$$Y_{t+1} = X_t$$
$$X_{t+1} = a - X_t^2 + bY_t$$

In all of the above mathematical models a, b and c are parameters that need to be estimated using the corresponding numerical methods.

The algorithm for automated mathematical modelling for prediction can be stated as follows:

STEP 1: Read the data set D = $\{d_1, d_2, ..., dm\}$.

STEP 2: Time Series Analysis of the set D to find the components.

STEP 3: Find the set of Admissible models MA = $\{M_1, M_2, ..., Mq\}$, using the qualitative values of the time series components.

STEP 4: Find the "Best" mathematical model Mb from the set MA using the measures of "goodness" of each of the models from the set MA.

In section 4 we will show how this algorithm can be implemented to achieve the goal of AMM for Engineering Systems.

3 Method for automated simulation of dynamical systems

We will consider the problem of performing automated simulation of dynamical systems and how this problem can be solved using AI techniques. First, we will consider some of the main ideas involved in the process of simulation of a mathematical model. Then, we will consider how this ideas can be implemented as a computer program.

3.1 Simulation of mathematical engineering models

The problem of performing an efficient simulation for a particular engineering system can be better understood if we consider a specific mathematical model. Let us consider the following model:

Transactions on Information and Communications Technologies vol 16, © 1996 WIT Press, www.witpress.com, ISSN 1743-3517

$$X' = \sigma(Y-X)$$
$$Y' = rX - Y - XZ \quad (1)$$
$$Z' = XY - bZ$$

where X, Y, Z, σ, r, b $\in$ R, and σ, r and b are three parameters which are normally taken, because of their physical origins, to be positive. The equations are often studied for different values of r in $0< r <\infty$. This mathematical model has been studied by Rasband [11] to some extent, however there are still many questions to be answer for this model with respect to its very complicated dynamics for some ranges of parameter values.

If we consider simulating eq.(1), for example, the problem is of selecting the appropriate parameter values for σ, r, b, so that the interesting dynamical behavior of the model can be extracted. The problem is not an easy one, since we need to consider a three-dimensional search space σ r b and there are many possible dynamical behaviors for this model. In this case, the model consisting of three simultaneous differential equations, the behaviors can range from simple periodic orbits to very complicated chaotic attractors. Once the parameter values are selected then the problem becomes a numerical one, since then we need to iterate an appropriate map to approximate the solutions numerically.

3.2 Method for automated simulation using AI

The problem of performing automated simulation for a particular engineering system is then of finding the "best" set of parameter values BP for the mathematical model. Here is where AI methodology comes to be very useful. The main idea in AI is that we can use certain techniques to simulate human experts in a particular domain of application. In this case then, we use heuristics extracted from experts in this domain and statistical calculations to limit the search space for the computer program. The algorithm for selecting the "best" set of parameter values can be stated as follows:

Step 1 Read the mathematical model M.

Step 2 Analyze the model M to "understand" its complexity.

Step 3 Generate a set of admissible parameters AP using the initial "understanding" of the model. This set is generated using heuristics (expressed as rules in the knowledge base) and solving some mathematical relations that will be defined later.

Step 4 Perform a selection of the "best" set of parameter values BP. This set is generated using heuristics (expressed as rules in the knowledge base).

Step 5 Perform the simulations by solving numerically the equations of the mathematical model. At this time the different types of dynamical behaviors are identified.

The implementation of this algorithm is described in section 5 of this paper. The result of this implementation is a computer program that can be considered an intelligent system for the simulation of dynamical engineering systems.

4 Implementation of the method for automated modelling

The implementation of the method for AMM as a computer program was done using the PROLOG programming language. The choice of PROLOG is because of its symbolic manipulation features and also because it is an excellent language for developing Prototypes, see Bratko [3]. The computer program was developed using an architecture very similar to that of an intelligent system (knowledge base, inference engine and user interface) with the addition of a numerical module for parameter estimation. We will focus our description of implementation details only to the knowledge base of the intelligent system. In the computer program, the knowledge base is the part that simulates the process of model discovery described by step 1 to 4 in the algorithm of section 2. Accordingly, the knowledge base consists of three Expert Modules: Time Series Analysis, Expert Selection and Best Model Selection.

4.1 Description of the time series analysis module

This module is the implementation of Step 2 of the algorithm and contains the knowledge necessary for time series analysis, i.e., the knowledge to extract from the data the time series components. Our method for time series analysis consist in the use of the fractal dimension of the set of points D as a measure of the geometrical complexity of the time series. We use the value of the fractal dimension to classify the time series components over a set of qualitative values. Our classification scheme was obtained by a combination of expert knowledge and mathematical modelling for several samples of data. To give an idea of this scheme we show in Table 1 some sample rules of this module.

Table 1.- Sample rules for time series analysis

IF	THEN
Fractal_dimension(D)$\in$ (0.8,1.2)	Trend = linear, Time_series = smooth
Fractal_dimension(D)$\in$ [1.2,1.5)	Trend = non_linear, Time_series = cyclic
Fractal_dimension(D)$\in$ [1.5,1.8)	Time_series = erratic
Fractal_dimension(D)$\in$ [1.2,1.4)	Periodic_part = simple
Fractal_dimension(D)$\in$ [1.4,1.6)	Periodic_part = regular

Fractal_dimension(D)∈ [1.6,1.7)	Periodic_part = difficult
Fractal_dimension(D)∈ [1.7,1.8)	Periodic_part = very_difficult
Fractal_dimension(D)>1.8	Periodic_part = chaotic

We performed several experiments with real data sets to decide on the classification needed for this "Time Series Analysis Module" and we found that for the moment classifying the periodic components in "simple", "regular", "difficult" and "chaotic" was sufficient. Also, we only classify the "trend" component in two Kinds: "linear" and "no-linear". Of course, it is possible that we may need better classification in the future, for a more accurate implementation of this Module, but now we are only showing how the method can be implemented.

In conclusion our method for time series analysis can be viewed as a one to one mapping between the fractal dimension of the set D and the qualitative values of the time series components.

4.2 Description of the expert selection module

This module is the implementation of the step 3 of the algorithm and contains the knowledge necessary to select the kind of mathematical models more appropriate for the type of data given, i.e., given the qualitative values of the time series components decide which models are more likely to give a good prediction. Our method for selecting the models consists of a set of fuzzy rules (heuristics) that simulates the human expert decision process of model selection. In our approach the qualitative values of the time series components are viewed as fuzzy sets (using the fractal dimension as a classification variable). We have membership functions for each of the qualitative values of the time series components. Also, the qualitative values of the "Type_Model" variable are considered as fuzzy sets and we have membership functions for each of this values. To give and idea of the way this Expert knowledge is structured, we show in Table 2 some rules of this module.

Table 2.- Sample fuzzy rules for model selection

IF			**THEN**
Dim	Trend	Periodic	Type_Model
one	non_linear	simple	logistic_differential_equation
two	non_linear	simple	lotka_volterra_differential_equation
three	non_linear	regular	lorenz_differential_equation
one	non_linear	simple	logistic_difference_equation
two	non_linear	regular	lotka_volterra_difference_equation

The rules in Table 2 show how this Expert Module selects the appropriate models for a given engineering problem, using as information the dimensionality of the problem and the qualitative values of the time series components. Each rule of this Expert Module contains a piece of Knowledge about the problem of model selection in engineering domains.

We have to mention here that the role of Fuzzy Logic is very important because it enables the simulation of the expert reasoning process under uncertainty for our problem. We came to a conclusion that the rules, for deciding which models are appropriate for a given time series, can't be categorical because the complexity of engineering modelling problems is very high. Since it is well known that Fuzzy Logic has been applied successfully to many engineering problems, see Badiru [2], and our problem requirements needed reasoning under uncertainty, we decided to use Fuzzy Logic techniques. In the following lines we will explain how the knowledge of the Experts is contained in the fuzzy rules with a complete example.

Suppose that a Time Series Analysis on a particular time series for a one-dimensional problem results in a Trend component valued as "non-linear" with a fractal dimension of 1.35, and a Periodic component valued as "simple" with the same fractal dimension, then the logical conclusion is that the "Logistic Map" is the best model for this problem with a 90% certainty. Of course, other mathematical models have a lower degree of certainty for this particular example. The reasoning behind this rule is that a time series that exhibits a non-linear trend and simple periodicity can be modeled by a logistic map with relatively good accuracy.

4.3 Description of the best model selection module

This module is the implementation of step 4 of the algorithm and contains the knowledge to select the "best" mathematical model for prediction, i.e., given the set of selected models generated by step 3, decide which model is the "best" one to predict the time series. Our method for selecting the "best' model consists of comparing the Sum of Squares of Errors (SSE) for all the models and selecting the one that minimizes SSE. This criteria has the advantage of been valid for all the types of models that we consider for the intelligent system (statistical models and non-linear dynamical systems models).

The reasoning behind this criteria is that the value of the SSE is a measure of how well a particular mathematical model fits the data (time series) for a given problem.

To give an idea of this method, suppose that the set of admissible models is $MA = \{M_1, M_2, ..., M_q\}$ for a given problem, then the corresponding set of Sums of Squares of Errors is: $SA = \{SSE_1, SSE_2,, ..., SSE_q\}$.

Let M_2 be the mathematical model that satisfies the following relationship:

$$SSE_{min} = SSE_2 = \min\{SSE_1, SSE_2, ..., SSE_q\}$$

then the "best" model for the problem is M_2 because it minimizes the distance from the model to the data.

We have to say here that this method for selecting the "best" model for a given problem can be improved in several ways to consider other factors that relate to this decision process. For example, one may like to consider the "type" of the model or the "simplicity" of the model as other factors of importance in the process of "best" model selection. In this case, a set of if-then rules would be required to make the decision and the module would be then considered a real "Knowledge Base". For the moment, we have only a method for "Best" Model Selection that uses statistical measures and "Knowledge" about the process of mathematical modelling.

5 Implementation of the method for automated simulation

The implementation of the method for AS as a computer program was done using the PROLOG programming language. The choice of PROLOG is because of its symbolic manipulation features and also because it is an excellent language for developing prototypes [3]. The computer program was developed using an architecture very similar to that of an expert system (knowledge base, inference engine and user interface) with the addition of a Machine Learning Module and a Numerical Module. The inference engine and user interface are based mainly in PROLOG language and will not be described in this paper because they do not relate directly to the simulation and modelling problem. We will describe in the following sections the Knowledge Base and the Machine Learning Module of the Intelligent System.

5.1 Knowledge acquisition and the machine learning module

We use the Machine Learning Module to construct the knowledge base of the Intelligent System. The Machine Learning Module consist of a Computer program that "learns" to simulate dynamical engineering systems. The method of learning is known as "Learning from examples" or "Induction". We show below this learning algorithm.

Let (M1, F1), (M2, F2), ..., (Mn, Fn) be a set of Training examples for the algorithm, where M1, ..., Mn are Mathematical Models and F1, F2, ..., Fn are mappings that give us the dynamical behavior for the corresponding parameters. The Function Fi represent the expert knowledge about selecting the "best" set of parameters for discovering the full range of dynamical behaviors. The steps of the algorithm are:

Algorithm:

1) Consider the first example as the initial hypothesis of the "understanding" of the algorithm.

Mathematically: $H1 = (M1, F1)$

2) For each of the following examples develop the new hypothesis H2, H3,..., Hn. The final hypothesis will be the "final understanding" about the problem.

After the computer program has processed the examples the output will be a set of rules that can be considered the desired knowledge base for the Intelligent System.

5.2 Knowledge base

The knowledge base (KB) for simulation of the Intelligent System consist of a set of rules (in Prolog) containing heuristics and mathematical knowledge about the problem of computer simulation of non-linear mathematical dynamical models, in particular for engineering systems. To give an idea of how this knowledge is contained in the KB we will show below some sample rules for several types of dynamical systems:

1) Ueda and Akamatsu Model: This mathematical model of a sinusoidally non-linear electronic oscillator consist of two simultaneous differential equations:

$$X' = Y$$

$$Y' = \alpha\,(1-X^2)Y - X^3 + \beta\cos(ft)$$

where the parameters α and β are positive and $\alpha < 1$ and $\beta < 25$. Ueda [13] has presented an extensive gallery of periodic and chaotic motions for this model. In this case the equilibria (X*,Y*) is stable if and only if the real parts of the eigenvalues are negative and this is equivalent to the rule:

IF $a > 0$ **THEN** Equilibria = stable

where a is defined by the characteristic equation: $\lambda^2 + a\lambda + b = 0$, with $a = -\,trJ$, $b=detJ$. Where "trJ" is the trace and "detJ" is the determinant of the Jacobian Matrix.

Another rule of the knowledge base is the following (for $\beta = 0$):

IF $\alpha\,(1-X^2) < 0$ **THEN** Equilibria = asimptotically_stable

Another rule is (for $\beta = 0$):

IF $\alpha\,(1-X^2) = 0$ **THEN** Hopf_Bifurcation

which gives us the condition for a Hopf Bifurcation to occur.

2) Other Bi-dimensional Models: Similar bi-dimensional autonomous models can be written in the following manner:

$$X' = \alpha\, f(X,Y)$$

$$Y' = \beta\, g(X,Y)$$

In this case, the Equilibria (X*,Y*) is stable if: $\alpha f_x + (g_y - \beta) < 0$, where f_x and g_y are partial derivatives. In AI language we have the rule:

IF $[\alpha f_x + (g_y - \beta) < 0]$ **THEN** Equilibria = stable

Also we have the following rule for a Hopf Bifurcation:

IF $\alpha_0 = (\beta - g_y)/f_x$ **THEN** Hopf_Bifurcation

3) Firth's Model of a single-mode laser: The basic equations for a single-mode (unidirectional) homogeneously broadened laser in a high-finesse cavity, tuned to resonance, may be written as a system of three differential equations, see Abraham & Firth [1]:

$$X' = \gamma_c (X + 2C_p)$$

$$P' = - \Gamma (P-XD)$$

$$D' = - \gamma (D + XP - 1)$$

Here X is a scaled electric field (or Rabi frequency), γ_c is a constant describing the decay of the cavity field and C is the cooperativity parameter.

In this case, the equilibria (X*, P*, D*) is stable if a, b, c>0 and (ab-c)>0, where a, b and c are defined by the characteristic equation of the system. We also have more complicated rules for other types of dynamical behaviors.

4) Other Three-dimensional Models: A three-dimensional system of differential equations can be written in the following form:

$$X' = \alpha f(X,Y,Z)$$

$$Y' = \beta g(X,Y,Z)$$

$$Z' = \gamma h(X,Y,Z)$$

In this case, the Equilibria (X*,Y*,Z*) is stable if a, b, c > 0 and (ab - c) > 0, where a, b and c are defined by the characteristic equation for the system:

$$\lambda^3 + a \lambda^2 + b \lambda + c = 0$$

In AI language we have the rule:

IF a,b,c >0 **AND** (ab-c) >0 **THEN** Equilibria = stable

other rules follow in the same manner for all the types of dynamical behaviors possible for this class of mathematical models.

We have to note here that the computer program can obtain the symbolic derivatives for the functions in the conditions of the rules. This is critical for the problem of simulation, since we require this derivatives to obtain the values of the parameters α, β and γ.

6 Validation of the intelligent system

The process of validation for the prototype intelligent system consisted of comparing the results of the computer program against the known results by the human experts, for a set of engineering problems. The comparison was done considering both parts of the problem, i. e., modelling and simulation. The results of this validation are encouraging, the computer program has a 90% accuracy with respect to the human experts in this domain. However, there is still much work to be done, because we are only considering at the moment a relatively small variety of mathematical models for this kind of problems.

We have to say that the results of the validation of the intelligent system can be considered good evidence that the algorithms presented in this paper are "sound" an that our work as a whole can be considered a promising line of future research in the areas of Mathematical Modelling and Simulation using AI techniques.

We are now going to show some of the results of the validation process performed on the intelligent system. In Table 3 we show the comparison between the Mathematical Models selected by the computer program and the Models given by the human Experts for a set of Examples.

Table 3.- Sample Comparison between the Expert and the System.

No.	**Engineering Problem (time series)**	**Model given by the System**	**Model given by the Experts**	**Comparison Equal (1) Different (0)**
1	Non-linear electronic oscillator	Ueda and Akamatsu	Ueda and Akamatsu	1
2	Duffing's oscillator	Duffing's equation	Duffing's equation	1
3	Van der Pol oscillator	Van der Pol equation	Van der Pol equation	1
4	Turbulent fluid flow	Lorenz equation	Lorenz equation	1
5	Single-mode homogeneous Laser	Lorenz equation	Firth's model	0

From the results of the validation we can see that the intelligent system predicts very well the right model for many engineering problems. However, there are still some cases where the system runs into trouble, i.e., the problem of the single-model laser where the system should predict Firth's model [1]. The reason why this happens is that both the Lorenz equation and Firth's model are very similar. We still have to do some work in the knowledge base of the intelligent system to refine the process of selection of the mathematical models.

The results in Table 3 show only part of the validation process for the modelling algorithm. A similar validation process was also performed for the simulation algorithm. The results for both parts of the validation are good in general. However, there is still much work to be done, because we have to extend the set of mathematical models that the intelligent system can explore and also we have to test the system with respect to the task of time series prediction. We think that the task of time series prediction is going to be a more difficult challenge for our methods and also for the implementations.

7 Conclusions

We described in this paper a new method to perform automated mathematical modelling for dynamical engineering systems. Also, we described the implementation of this new method as a prototype intelligent system for the domain of engineering problems. We have tested this computer program with real engineering data with encouraging results. In this paper we have successfully generalized our previous work on this matter, see Castillo & Melin [5] by using non-linear dynamical systems models.

We also described in this paper a new method for automated simulation of non-linear dynamical engineering systems and its implementation as a computer program. This program selects the appropriate parameter values to perform the numerical exploration of the engineering system and then decides which dynamical behaviors are found for different parameter intervals. This information can be used to perform a real-time simulation of the engineering system and then can be used to predict its future behavior or to control it, which ever is desired.

We have tested the prototype intelligent system with several non-linear models of real known engineering systems with encouraging results. The intelligent system performs "good" simulations and obtains "good" predictions of the dynamical behaviors of the engineering systems. More tests with more complicated models are needed to continue the development of the intelligent system.

Acknowledgments

The authors would like to thank the COSNET research grant committee for supporting this project, the Department of Computer Science of Tijuana Institute of Technology for the time and resources given to this project and the School of Engineering of CETYS University for the collaboration in this research and for providing the Computing facilities needed for this research work.

References

1. Abraham, E. and W. J. Firth, (1984). "Multiparameter universal route to chaos in a Fabry-Perot resonator" Optical Bistability, Vol. 2, pp. 119-126.

2. Badiru, A.B. (1992). "Expert Systems Applications in Engineering and Manufacturing", Prentice-Hall.

3. Bratko, I. (1990). "Prolog Programming for Artificial Intelligence", Addison Wesley.

4. Castillo, O. and P. Melin, (1994a). "An Intelligent System for Research and Education of Non-Linear Dynamical Systems", Proceedings of IMACS'94, Vol. 1, pp. 82-85, Georgia Institute of Technology.

5. Castillo, O. and P. Melin (1994b). "An Intelligent System for Discovering Mathematical Models for financial time Series Prediction", Proceedings of the IEEE Region 10's Ninth Annual International Conference, pp. 217-221, IEEE Computer Society of Singapore.

6. Castillo, O. and P. Melin (1995a). "An Intelligent System for Financial time Series Prediction Combining Dynamical Systems Theory, Fractal Theory and Statistical methods", Proceedings of the IEEE/IAFE 1995 Conference on Computational Intelligence for financial Engineering (CIFER), pp. 151-155.

7. Castillo, O. and P. Melin, (1995b). "An Intelligent System for the Simulation of Non-Linear Dynamical Economical Systems", Journal of Systems, Analysis and Simulation, Vol. 19, pp. 767-770, Gordon and Breach Publishers.

8. Devaney, R. (1989). "An Introduction to Chaotic Dynamical Systems", Addison Wesley Publishing Company.

9. Mandelbrot, B. (1983). "The Fractal Geometry of Nature", W. H. Freeman and Company.

10. Mendenhall, W. and J. E. Reinmuth (1981). "Statistics for management and Economics", Wadsworth International.

11. Rasband, S. (1990). "Chaotic Dynamics of Non-Linear Systems", John Wiley & Sons.

12. Rao, R. B. and S. Lu (1993). "An Knowledge-Based Equation Discovery System for engineering Domains, IEEE Expert, pp. 37-42.

Transactions on Information and Communications Technologies vol 16, © 1996 WIT Press, www.witpress.com, ISSN 1743-3517

13. Ueda, Y. and N. Akamatsu (1981). "Chaotically Transitional phenomena in the Forced Negative Resistance Oscillator", IEEE Trans. CS, Vol. 28, pp. 217-224.