Transactions on Information and Communications Technologies vol 19, © 1997 WIT Press, www.witpress.com, ISSN 1743-3517

Mathematical modelling and simulation of robotic dynamic systems using an intelligent tutoring system based on fuzzy logic and fractal theory

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

Abstract

We describe in this paper a computer program for Mathematical Modelling and Simulation of Robotic Dynamic Systems using Fuzzy Logic Techniques and Fractal Theory. The computer program combines Artificial Intelligence (AI) techniques with mathematical methods and can be considered an Intelligent Tutoring System (ITS) for the domain of modelling and simulation of robotic systems. This domain is quite complex because robotic systems can be viewed as non-linear dynamical systems, and it is a well known fact that even very simple non-linear dynamical systems can exhibit "chaotic" behavior. The computer program simulates the reasoning of a human expert in the process of teaching how to develop mathematical models of Robotic Dynamic Systems (RDS). The program contains the knowledge of the human experts expressed as fuzzy rules (in the knowledge base) for Mathematical Modelling and Simulation (MMS) of robotic systems. The ITS also contains knowledge about teaching methodologies for this domain (in the knowledge base). The ITS uses efficiently AI techniques to teach MMS of RDS, and also to monitor the learning process of students of this domain. Mathematical Modelling and Simulation of Robotic Systems is very important because it can help in the control of an actual system or in the design of a new system using the results of the simulations.

1 Introduction

We describe in this paper an Intelligent Tutoring System (ITS) for the domain of Mathematical Modelling and Simulation (MMS) of Robotic Dynamic Systems (RDS). Intelligent Tutoring Systems are very sophisticated software systems based on Artificial Intelligence (AI) techniques and Cognitive Science, see Takeuchi and Otsuki [14]. An ITS can be used to teach a specific domain of application because it simulates the reasoning process of human experts as they teach their domain of expertise. In this case, an ITS was developed for the domain of Dynamic Systems in Robotics using Fuzzy Logic techniques and Fractal Theory. The study of Non-Linear Dynamical Systems in general is very interesting because their behaviors can range from very simple periodic solutions to the very complicated "chaotic" behavior, see Devaney [9]. For the case of Robotic Systems in particular, realistic dynamic mathematical models are always non-linear and then the corresponding behaviors of the Robots can be quite complex, see Vukobratovic' [15]. For this reason, the problem of obtaining the right mathematical model representing the dynamical behavior of RDS is very important. Having a mathematical model of the RDS enables the simulation of the system for the prediction of future behavior and also for the design of the "best" control possible for the robotic system.

An ITS is similar to an Expert System (ES) in that it contains the knowledge about a specific domain of application. However, an ITS is different with respect to an ES in that it also contains knowledge about teaching methodologies and student modelling. This difference is very important because a common mistake done by many researchers is to believe that a plain ES can be used as a teaching/training tool for a specific domain. However, is easy to show that a plain ES can not evaluate or monitor the learning process of their students. Once this fact is recognized an ITS can be build using as a foundation the knowledge of the corresponding ES. In this, the authors developed before an ES for Automated Mathematical Modelling and Simulation of Dynamical Engineering Systems, see Castillo and Melin [8]. This Expert System has been used as a starting point in building the ITS described in this paper. The method for MMS is based on the use of Fuzzy Logic techniques, see Badiru [2], to select the appropriate mathematical model for a given problem. Also, the method for MMS uses the concept of the fractal dimension, see Mandelbrot [11], to classify the components of the time series (data for the problem). This method for MMS is contained in the knowledge base of the ITS and is the basis to obtain the mathematical models for Robotic Systems. The method for Simulation of a mathematical model is based on an algorithm developed by the authors, see Castillo and Melin [7]. This algorithm enables the selection of the right parameters of the model by using heuristics from the experts of the domain. Once the parameters are selected the numerical simulations are performed and then the dynamical behaviors are identified.

The ITS described in this paper has the goal of teaching graduate students or engineers in the task of modelling Robotic Systems and then performing numerical simulations on the models, to identify the corresponding dynamical behaviors for the system. The process of teaching this domain is difficult, and one of the reasons for this it that it requires a strong background by the students in differential equations and difference equations. In this ITS we are assuming that the student has the required prior knowledge mentioned above. When a student uses the Intelligent Tutoring System, he or she enters in an individualized learning process of the domain of RDS. The ITS monitors the actual learning process of the students and also stores all the relevant information about the learning status of the students. In this ITS, the learning process is considered as consisting of six phases:

1.- Acquiring the basic concepts and theory about RDS.
2.- Learning the solution of well known dynamical systems in the literature (for robotic systems)
3.- Learning to obtain mathematical models for given problems in the area of RDS.
4.- Learning to perform the right numerical simulations for a given mathematical model of a robotic system
5.- Learning to identify dynamical behaviors from the numerical simulations of the models.
6.- Learning to explore the future behavior of the system and the possible ways to control it.

The ITS gives the theory for each phase and then gives examples. At the end of each phase, the student is given an examination to evaluate his level of understanding of the theory and also to adapt the level of future teaching according to the performance of the student.

The reasons why we consider that an ITS is justified for this domain are the following:

1.- There are not many experts in Dynamical Systems Theory and Robotic Systems, because both of this areas of study are relatively new in Computer Science and Mathematics.
2.- The domain of Dynamical Systems is a complex one, because of the very different types of behaviors possible (they can range from simple periodic orbits to very complicated chaotic orbits). Then to teach this domain becomes necessary to explain very clearly each logical or numerical step done in the exploration of a given dynamical system.

We believe that an ITS like ours can help solve both problems. First of all, the ITS contains the knowledge of the experts in RDS (within the scope the computer program). Also, the ITS has the ability to explain to the user each of the steps performed in the exploration of a given dynamical system.

2 Modelling and simulation of robotic systems

The adequate study of robotic systems starts with the development of computer-oriented methods for building the mathematical models of kinematics and dynamics of spatial active mechanisms. The rising state of development of robotic mechanism mathematical models went the way from the numerical-iterative computer methods, to the numeric-symbolic ones, and finally, to the forming of mathematical models in symbolic form, see Vukobratovic' [16].

Mathematical models of robot kinematics and dynamics are not the ultimate goal, but tools for the synthesis of dynamic control of this mechanisms.

However, mathematical modelling and simulation are very important tasks in achieving the ultimate goal of "control" because they provide useful information about the dynamical behavior of the robotic system.

In the last several years, many papers have been published, rendering an important contribution to the development of computer methods for the mathematical modelling of robotic systems. The modelling methods may be classified with respect to the laws of mechanics on the basis of which motion equations are formed. One may distinguish methods based on Lagrange, Newton-Euler, D'Alembert, and other formalisms for dynamic modelling of interconnected multibody systems. The dynamic model of the robot consists of the model of the mechanical part of the robot (mechanism) and the model of the actuators that are driving the robot joints. The model of the mechanical part of the robot is usually assumed, see Vukobratovic' [16], in the following form:

$$P = H(q)\, q'' + h(q, q') \qquad (1)$$

where P = n x 1 vector of driving torques in the joints, H = n x n inertia matrix of the mechanism, h = n x 1 vector of centrifugal, coriolis, and gravity moments (forces) around the axes of the joints. A specific mathematical model for a particular robot can be obtained by specifying the values for P, H and h in equation (1). But in any case the resulting model is a complex non-linear dynamical system.

Various types of actuators are applied to drive robots: dc motors, ac motors, hydraulic actuators, pneumatic actuators and so on. The models of actuators are in general non-linear, but for the dc motors (which are still most often applied for industrial robots) a linear state model may be used:

$$(X^i)' = A^i X^i + b^i u^i + f^i P_i, \quad i = 1, 2, ..., n \qquad (2)$$

where: $X^i = (q_i, q_i', i_{Ri})^T$ = 3 x 1 state vector of ith actuator model,
i_{Rj} = rotor current of ith dc motor
u^i = scalar input to ith actuator
P_i = driving torque (load) in ith joint
A^i = 3 x 3 matrix, b^i, f^i = 3 x 1 vectors

The connection between the models (1) and (2) (through the state coordinates q_i, q_i', and driving torques P_i) are evident. Certain constraints upon the actuators input u amplitude as well as on the allowable driving torques should be also added to these models.

The ITS enables the modelling and simulation of the dynamic models of various robots of arbitrary type and structure. The ITS shows to the user/student how to develop the dynamic model in symbolic form (like in equation 1) and then shows to the user/student how to perform simulations to identify all the possible dynamical behaviors of the robotic system.

3 Method for automated mathematical modelling

In this section a method for automated modelling developed by the authors, see Castillo and Melin [5,6], is described briefly. 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 [13]. 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 ε $(0,\sigma)$ 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 mathematical models for continuous dynamical systems can be one-dimensional, two-dimensional or three-dimensional. We show below some sample models [12] 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 [12] 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:

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

<u>STEP 2</u>: Time Series Analysis of the set D to find the components.

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

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

4 Method for automated simulation of dynamical systems

In this section a method for automated simulation developed previously by the authors, see Castillo and Melin [4,7], is described briefly. 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:

$$X' = \sigma(Y-X)$$
$$Y' = rX - Y - XZ \tag{3}$$
$$Z' = XY - bZ$$

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 [12] 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.(3), 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.

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. 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.

We are considering in this algorithm the "best" set of parameter values for simulation, the ones that enable the correct identification of all the dynamical behaviors for a particular mathematical model.

5 Description of the intelligent tutoring system

We describe below the general architecture of the ITS, and then we give a detailed description for each of the modules, except for the Inference Engine and the Numerical Module. The Inference Engine for the System is based on the Prolog Programming Language, see Bratko [3], and the numerical module is not our main concern in this paper.

5.1 Architecture of the intelligent tutoring system

The architecture of an ITS is very similar to that of an Expert System (ES) for a specific domain of application. However, there are also some main differences between an ITS and an ES. One of the main differences is in the knowledge base (KB), because for an ITS the KB needs to contain Expert Knowledge about teaching methodologies and not only Expert knowledge about the domain of application. The Knowledge Base of the ITS consists of three parts: Expert Module, Teaching/Learning Module and Student Module. For an ITS the last two modules are the ones that enables the teaching of new students for the domain and also enable monitoring of their learning process. We will explain in the next section how this is accomplish in our ITS for the domain of RDS.

5.2 Description of the knowledge base

The knowledge base of the ITS consists of three modules: the Expert Module, the Teaching/Learning Module and the Student Module. The Expert Module contains the knowledge about the domain of RDS, i.e., the knowledge about which numerical methods can be used for the identification of possible behaviors of a dynamical system and activates the numerical module. The expert module also has the knowledge to generate problems for the user/student to solve. The Teaching/Learning Module contains the knowledge about teaching methodologies and the knowledge for diagnosis of student's attributes. The Student Module is a data base of the student's performance history and a catalog of student's attributes.

5.2.1 Description of the expert module

The Expert Module contains the knowledge about the method for automated mathematical modelling described in section three and also the knowledge about the method for automated simulation described in section four. Accordingly the Expert Module is divided in two submodules: 1) submodule of automated modelling and 2) submodule for automated simulation. In the following lines we will describe briefly each of these modules.

5.2.1.1 Submodule of automated modelling

Automated mathematical modelling consists in the process of model discovery described by steps 2 to 4 in the algorithm of section 3: Time Series Analysis, Model Selection and Best Model Selection. We describe briefly each of this steps in the following lines.

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)$\in$[1.6,1.7)	Periodic_part = difficult
Fractal_dimension(D)$\in$[1.7,1.8)	Periodic_part = very_difficult
Fractal_dimension(D)>1.8	Periodic_part = chaotic

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.

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.

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

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.

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.

5.2.1.2 Submodule for automated simulation

The knowledge for simulation of the Intelligent Tutoring 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:

$$X' = Y$$

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

Transactions on Information and Communications Technologies vol 19, © 1997 WIT Press, www.witpress.com, ISSN 1743-3517

where the parameters α and β are positive and $\alpha < 1$ and $\beta < 25$. Ueda [17] 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 $a = -\text{tr}J$, $b=\det J$. 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.

5.2.2. Description of the teaching/learning module

The teaching expertise consists of knowledge for planning global teaching sequences and local teaching methods. The global teaching sequence is determined according to the domain knowledge structure and the readiness.

In the planning process, the teaching expertise should select a new teaching objective which is not too easy nor too difficult for a learner. The degree of how well a learner will understand a teaching objective is evaluated by "readiness". The readiness, see Takeuchi and Otsuki [14], is defined as follows:

$$\text{readiness} = (p + (1 - p) \cdot c \cdot a) \cdot f(d)$$

where "p" is the mean value of understanding level of all prerequisite knowledge of the teaching objective, "d" expresses the difficulty level of the teaching objective, which is defined by the course designer, "a" is the overall understanding level of a learner and is equal to the mean value of understanding level of all objectives throughout the learner's past learning process, and "f" is a monotonically decreasing function ("c" is a constant, $0 < c < 1$).

The teaching/learning module selects a teaching objective which has the readiness falling within some fixed range. Then, the module selects the local teaching paradigm and a topic for generating appropriate messages. The teaching paradigm is a style of actions such as praising a learner, asking a sub-problem, giving counter-examples, explaining a part of a problem solving process, etc. The teaching/learning module selects an appropriate paradigm and a topic according to both the student model and history of learning, such as frequency of the same error occurrence, success/failure records of teaching paradigms, etc. The teaching/learning module also performs examinations on the student to evaluate the level of understanding for each of the topics of the course on MMS of RDS.

5.2.3. Description of the student module

User modelling in this ITS is done by means of a technique developed by Díaz-Ilarraza et al [10] that uses a record of the most recent and most characteristic errors. Students are categorized into three general classes according to their known experience: "novice", "medium" and "expert". besides the student's learning characteristic, the history of the learning process, the more common errors made, etc. are also taken into account. This information is recorded and handled by dynamically creating and updating the "student Profile".

The "Student Profile" is organized around two main knowledge bases: "student-representation" and "session-history". The first one is a static model (updated only at the end of a session) including both the student's background with his/her general learning characteristics and the knowledge acquired in previous teaching sessions; the second one is a dynamic model (updated during the session) representing the current teaching task: the history of the development of the current session with a record of the session protocol.

6 Validation of the intelligent tutoring system

The validation process for an ITS can be divided in two parts: (1) validating the knowledge of the domain ("Expert Module") and 2) Validating the knowledge of teaching methodologies ("Teaching Module"). For validating the Expert Module, we performed an extensive comparison between the results of this module (computer program) and the results given by the real human experts. the results of the Expert Module were accurate in approximately 90 % of the cases considered in the validation process. This in fact can be considered very good for this domain, considering the complexity of the area of Robotic Systems. With this results, the validation of the Domain Knowledge for the ITS was considered as appropriate for our goals (at the moment). On the other hand, to validate the knowledge of teaching (for the ITS), we have tested this computer program with several groups of students to see if the learning process of the domain was enhanced as a result of using this intelligent educational tool. The results of the students were able to learn the domain of RDS (at a certain level of difficulty) in less time that without the ITS. Also, the quality of learning of the domain seems to be better, since the students can explain in greater detail the concepts of this domain. Of course, we recognize that even more tests are needed to have a complete validation of the ITS. However, we can conclude, for the moment, with this results that the ITS can be considered an efficient educational tool for the domain of Mathematical Modelling and Simulation for Robotic Dynamic Systems.

7 Conclusions

We have developed an Intelligent Tutoring System for mathematical modelling and simulation of Robotic Systems using fuzzy logic and fractal theory. This ITS teaches the methods of automated mathematical modelling and simulation described in this paper. The idea in this paper is to show how AI techniques can be applied to capture the dynamics of robotic systems by using non-linear mathematical models. The models represent (to some extent) the robotic system and enable the simulation of the system to explore all of its possible dynamical behaviors. The ultimate goal in robotic systems is to find the optimum "control" for a specific application, and this can be achieved by modelling the system and then performing simulations to explore and predict its future behavior. The ITS has the goal of teaching to the students of this domain how to perform the tasks of modelling and simulation of RDS. The ITS also monitors the actual learning process for each of the students, until they achieve the desired level of understanding of the domain.

The importance of having an ITS for teaching students of this domain can be seen from several points of view. First of all, from the economical point of view the impact of this ITS in is reducing the costs of instruction (or training, if in industrial setting), because of the more efficient teaching method resulting from the application of AI and educational techniques to this domain. Second of all, from the educational point of view the impact of this ITS is in the contribution to the advance of the research in applications and methodologies of AI in education.

Keywords:

Modelling, Simulation, Robot Dynamics, Knowledge-Based Systems

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. Castillo, O. and P. Melin (1996). "An intelligent system for automated mathematical modelling and simulation of dynamical engineering systems", Proceeding of AIENG'96, pp. 9-10, Computational Mechanics Publications.
9. Devaney, R. (1989). "An Introduction to Chaotic Dynamical Systems", Addison Wesley Publishing Company.
10. Díaz-Ilarraza, A., J.A. Elorriaga, I. Fernández-Castro, J. Gutierrez-Serrano and J.A. Vadillo-Zorita (1992). "User Modeling and Architecture in Industrial ITS's" Proceedings of ITS'92, Springer Verlag, pp. 661-668.
11. Mandelbrot, B. (1983). "The Fractal Geometry of Nature", W. H. Freeman and Company.
12. Rasband, S. (1990). "Chaotic Dynamics of Non-Linear Systems", John Wiley & Sons.
13. Rao, R. B. and S. Lu (1993). "An Knowledge-Based Equation Discovery System for engineering Domains, IEEE Expert, pp. 37-42.
14. Takeuchi, A. and S. Otsuki (1992). "EXPITS: An Experimental Environment on ITS", Proceedings of ITS'92, Montreal, Springer Verlag, pp. 124-131.
15. Vukobratovic', K.M. (1989). "Applied Dynamics of Manipulation Robots: Modelling, Analysis and Examples", Springer-Verlag, Berlin.
16. Vukobratovic', K.M. (1991). "Robotics and Flexible Automation Simulator/Integrator", in Neural and Intelligent Systems Integration, Edited by Branko Soucek, Wiley Interscience Publisher.
17. Ueda, Y. and N. Akamatsu (1981). "Chaotically Transitional phenomena in the Forced Negative Resistance Oscillator", IEEE Trans. CS, Vol. 28, pp. 217-224.