AUTOMATED DYNAMIC PATTERN TESTING, PARAMETER CALIBRATION AND POLICY IMPROVEMENT[1]

Suat Boğ
Koc University, Dept of Industrial Engineering
34450 Sariyer, Istanbul – Turkey
Tel: ++90 212 338 1676
sbog@ku.edu.tr

Yaman Barlas
Bogazici University, Dept. of Industrial Eng.
34342 Bebek, Istanbul - Turkey
Tel: ++(90) 212 359 7073, Fax: ++(90) 212 265 18 00
ybarlas@boun.edu.tr

System Dynamics (SD) is a special type of simulation modeling where output validity refers to validating the patterns of dynamic behaviors, such as oscillations, growth or decline. The developers and users of these models (the decision makers and people affected by decisions based on such models) are all rightly concerned with whether a model and its results are "valid." Structural model validity and validation have long been recognized as one of the main issues in system dynamics. This concern is addressed through pattern recognition and testing in this paper. Another issue in dynamic simulation methodology is parameter calibration; assuming that the structure of simulation model constructed by the user is valid. Parameter calibration is the minimization of an error function which is a measure of the correspondence between numerically calculated output patterns and the respective real behavior patterns. We offer a software that does automated parameter calibration with respect to a given (desired) dynamic pattern. This particular feature can also be used in policy improvement design.

Keywords: Dynamic Pattern Recognition, Structure validity Testing, Parameter Calibration

INTRODUCTION

Some earlier reviews of System Dynamics methodology criticized it for not having formal / objective validity tools (Ansoff and Slevin, 1968; Nordhaus, 1973). Articles by Forrester (1968) and Forrester et al. (1974) are responses to these criticisms. Forrester attempts at establishing formal tools for model validation (Forrester and Senge, 1980). Forrester outlines the "formal" validation concept and introduces a set of tests for structure and behavior validation.

The importance of model validation in the development of System Dynamics discipline is emphasized in literature (Forrester and Senge 1980; Sterman 1984; Barlas 1989a; Barlas 1996, Barlas and Carpenter, 1990; Richardson, 1996). However, despite its importance, Barlas (1996) states that only three of the articles published in System

[1] Supported by Boğaziçi University Research Grant 02R102

Dynamics Review (1985-1995) deal with model validity/validation. It is observed that the majority of the literature on general model validation discusses behavior validation due to lack of recognition of the importance of structure validity, as well as the relative ease of developing tools for behavior validation. Barlas (1989a) and Barlas et al. (1997) provide a set of tools ("BTS" and BTS II) for testing the behavior validity of the model. Structural aspects are discussed in (Forrester, 1961, ch. 13; Forrester and Senge, 1980; Barlas, 1989b; Peterson and Eberlein, 1994).

There are some system dynamics validation tools and tests previously developed. But the existing tests/tools are *not integrated* to the available simulation software packages and they are *platform-dependent.*

We design a validity testing and parameter calibration Software (SiS) that does pattern recognition/testing/calibration with full integration with the simulation software VENSIM. 'Validity Testing' part of the software takes the dynamic behavior generated by the model, "recognizes" it and tests if it belongs to the class hypothesized by the modeler. It is also possible to set values for some parameters of the model and perform the test afterwards.

Second contribution of the software is that by assuming the structure of simulation model constructed by the user is valid; SiS software finds those parameter value sets for the model that best fit the real data patterns. In order to make repetitive parameter adjustments, the simulation package Vensim is run automatically by the aid of internal commands. There is no existing simulation packages which automatically achieve this calibration process.

In classical context, parameter calibration is done by the minimization of an error function which is a measure of the correspondence between numericaly calculated output values and respective measurements. However, in the software developed, Hidden Markov Models (HMMs) are used for pattern recognition. Each pattern class is characterized by a nonstationary HMM which is built using a set of training samples. As a result of the training procedure, one HMM is obtained for each class. The classification is based on the state-optimized likelihood function which is a measure of how well the input signal is representative of a given class. (Barlas and Kanar 1999) In parameter calibration, the likelihood values obtained for the specified class are used to find best parameter set, instead of using minimization of an error function. In order to make repetitive parameter adjustments, Vensim is run automatically by the aid of internal commands.

INDIRECT STRUCTURE TESTING SOFTWARE (ISTS)

ISTS is a computerized algorithm that seeks to automate the structure-oriented behavior testing in model validation. (Barlas and Kanar 1999; Kanar 1999) In such validity tests, the modeler makes a claim of the form: "if the system operated under condition C, the behavior B would result". The model is then run under condition C and is said to "pass" the test, if the resulting behavior is similar to the expected behavior. In the automated structure-oriented behavior testing environment, the modeler hypothesizes a dynamic behavior by choosing a dynamic pattern from a template of all basic patterns (shown in Figure 1). The computerized algorithm takes the dynamic behavior generated by the

modeler, “recognizes” it and tests if it belongs to the pattern class hypothesized by the modeler.

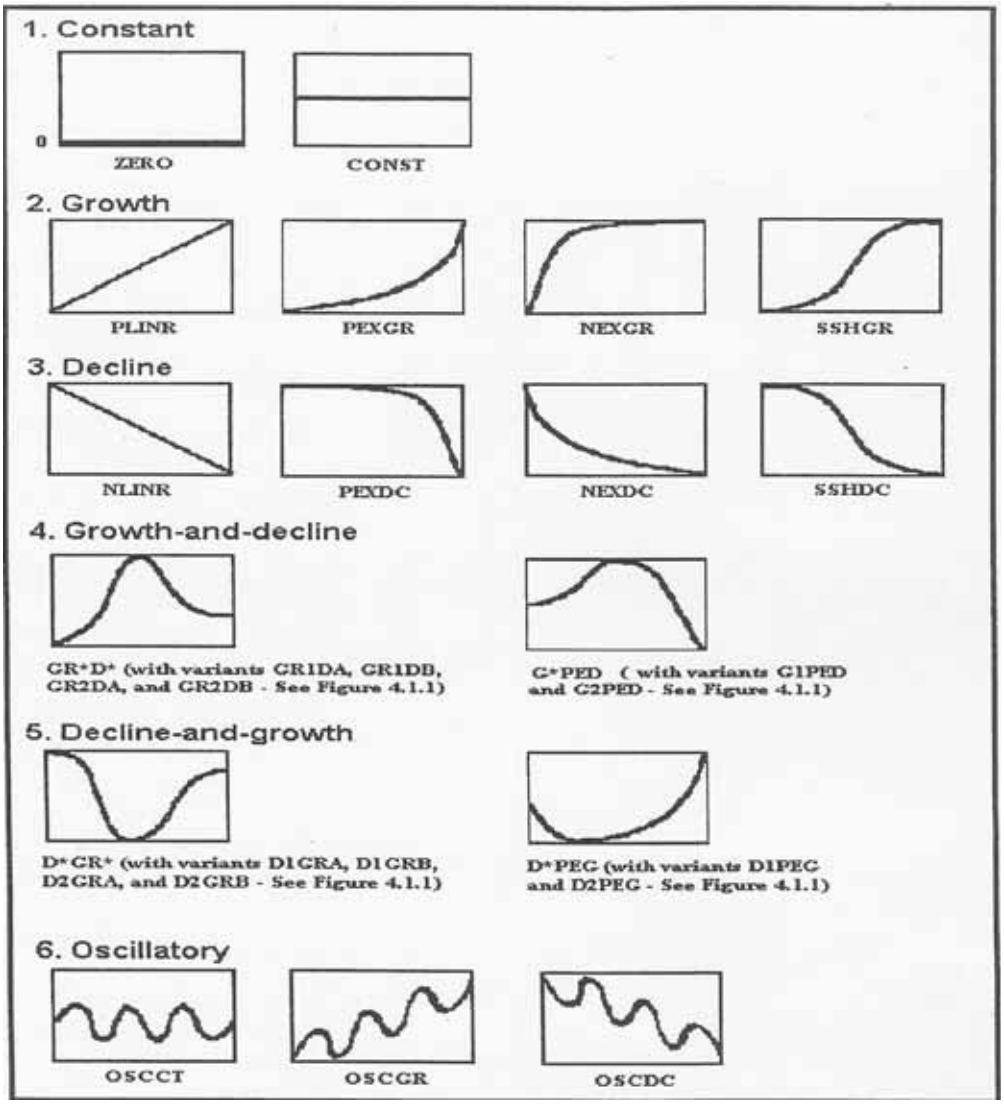

Figure 1. Basic Dynamic Patterns

The algorithm used in the implementation is a pattern recognizer/classifier based on Hidden Markov Models. The general flowchart of ISTS algorithm is seen in the Figure 3. (Barlas and Kanar 1999; Kanar 1999)

Abbreviation	Description
ZERO	Zero
CONST	Constant
PLINR	Linear with positive slope
NLINR	Linear with negative slope
NEXGR	Negative exponential growth
SSHGR	S-shaped growth
PEXGR	Positive exponential growth
GR1DA	Growth with decreasing rate followed by decline to equilibrium (growth level is less than decline level)
GR1DB	Growth with decreasing rate followed by decline to equilibrium (growth level is greater than decline level)
GR2DA	S-shaped growth and decline to equilibrium (growth level is less than decline level)
GR2DB	S-shaped exponential growth and decline to equilibrium (growth level is greater than decline level)
D1PEG	Decline with increasing rate followed by positive exponential growth
D2PEG	S-shaped decline followed by positive exponential decline
NEXDC	Negative exponential decline
SSHDC	S-shaped decline
PEXDC	Positive exponential decline
D1GRA	Decline with increasing rate followed by growth to equilibrium (decline level is less than growth level)
D1GRB	Decline with decreasing rate followed by decline to equilibrium (growth level is less than decline level)
D2GRA	S-shaped decline and growth to equilibrium (decline level is less than growth level)
D2GRB	S-shaped decline and growth to equilibrium (decline level is greater than growth level)
G1PED	Decline with decreasing rate followed by positive exponential decline
G2PED	S-shaped growth followed by positive exponential decline
OSCCT	Oscillation around constant mean
OSCGR	Oscillation around linearly growing trend
OSCDC	Oscillation around linearly declining trend

Figure 2. List of dynamic behavior pattern classes

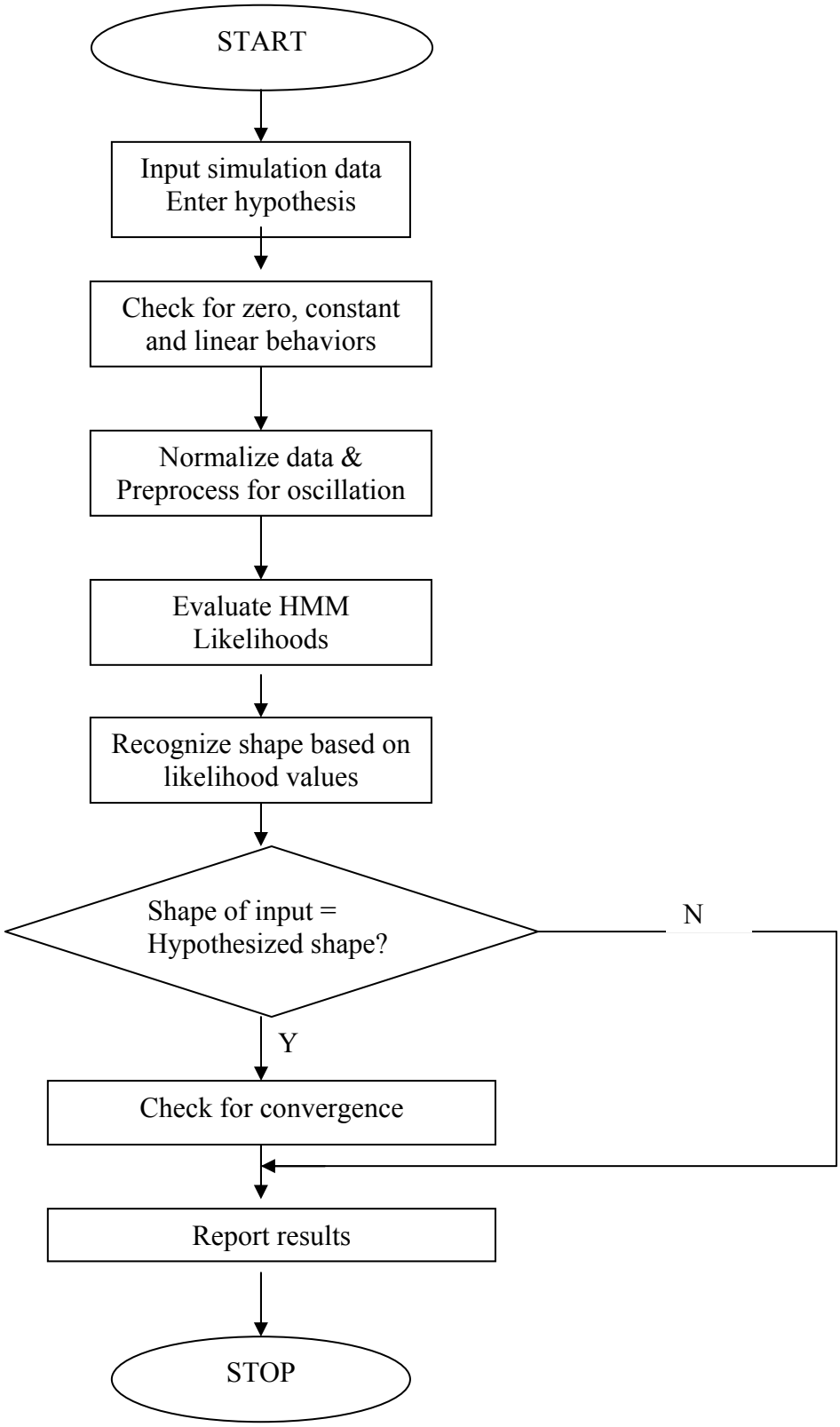

Figure 3. General Flowchart of ISTS Algorithm

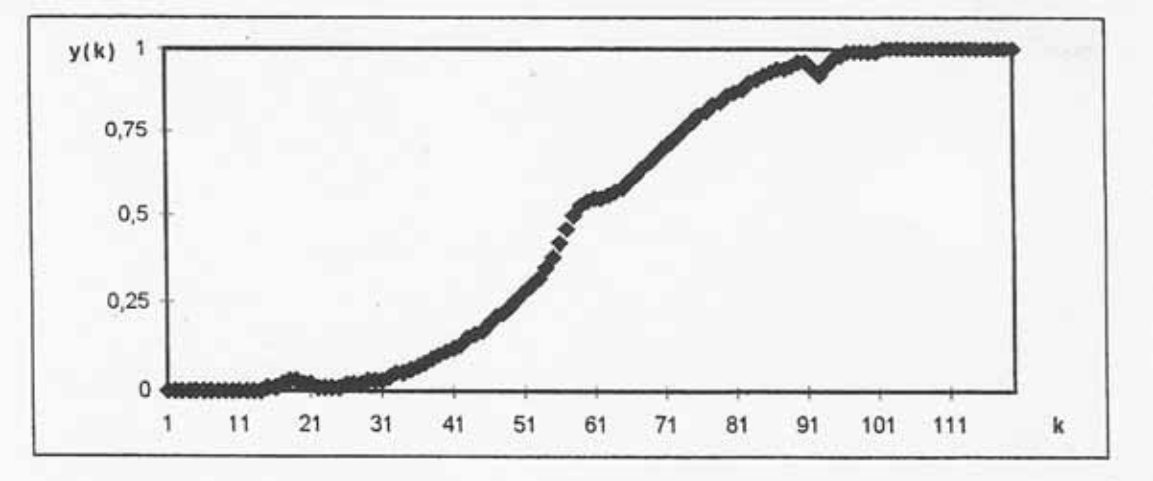

Figure 4. A dynamic pattern example.

Hidden Markov Models

Statistical pattern recognition is based on the classification of feature vectors extracted from the data. In most shape recognition problems, representation of the whole data with a single set of feature vectors is not desirable. It is difficult to model the variations in the characteristics along the shape with a single vector (He and Kundu, 1991). Hidden Markov Models do not represent the whole data with a single feature vector. In HMM based pattern recognition, one dimensional data is divided into segments and a sequence of feature vectors is extracted.

Selection and Extraction of Features

In the dynamic behavior recognition problem, a dynamic behavior can be denoted by a sequence *y(k), k= 1, 2, ….., K*, where *K* is the number of data points. As depicted in Figure 4. , such a signal would be a somewhat distorted (or "noisy") version of one of the patterns given in the template of Figure 1. The procedure starts with dividing the sequence

$y_t(l)$, $t= 1, 2, \ldots.. T$:
$y_t(l)=y[(t-1)L+l]$, $l=1, 2, \ldots..L$.

here, the choice of value T is one of the decisions to be made in the design of the recognition system.

The next step is to extract features from each data segment. Basic dynamic patterns are characterized by successive time segments of growth or decline and their trends (as growing or declining rates). Therefore, it is reasonable to form the feature vector using the slope and 2nd derivative("curvature") information of the data in each segment. The features can be obtained by fitting polynomials to each segment data. The slope of the first order polynomial provides trend information, which is either growth, decline or constant. The second order polynomial can be used to obtain the second derivative, which will yield the curvature information. In addition to slope and

curvature, the level of the state variable also provides useful information. Thus, the segment mean becomes the third element of the feature vector.

In summary, each feature vector is M=3 dimensional and are given by three components; slope, curvature, and mean:

$$o_t = \begin{bmatrix} \varphi_{t1} \\ c_t(\kappa_t) \\ m_t \end{bmatrix} \qquad \text{for } t = 1, 2, ..., T.$$

Therefore, the result of the segmentation and feature extraction process for a pattern sample using T number of segments is a sequence $\{o_1, o_2, \ldots..,o_T\}$. For the example signal in Figure 4 -and specially for its 5th segment –feature extraction yields

$$O = \left\{ o_1, o_2, o_3, o_4, \begin{bmatrix} 0.6711 \\ -2.5308 \\ 0.9440 \end{bmatrix}, o_6 \right\}.$$

It is quite flexible in the sense that new pattern classes can be added without having to re-train for all the existing classes. Also by training it with more and more samples, the classification power of the algorithm can be improved. (See Barlas and Kanar 1999; Kanar 1999 for more information)

SOFTWARE IMPLEMENTATION and ILLUSTRATIVE EXAMPLES

A computer program, Validity Testing and Calibration Software (SiS), has been developed for the purposes of integrating existing ISTS algorithm into the existing VENSIM dynamic system simulation software and automating the validity testing and parameter calibration of System Dynamics models. The software is written in JAVA programming language by using Sun One Studio Compiler Version 5.

"General Pictures" for 2 Main Fuctions of the Software

This section presents the overview for the two main processes which are the 'Validity Testing' and 'Parameter Calibration'.

Validity Testing

This function provides user to check the validity of a model output behavior according to the hypothesized pattern. Furhermore, user has the chance to change parameter values without any change on model and to make validity check by these desired parameter values.

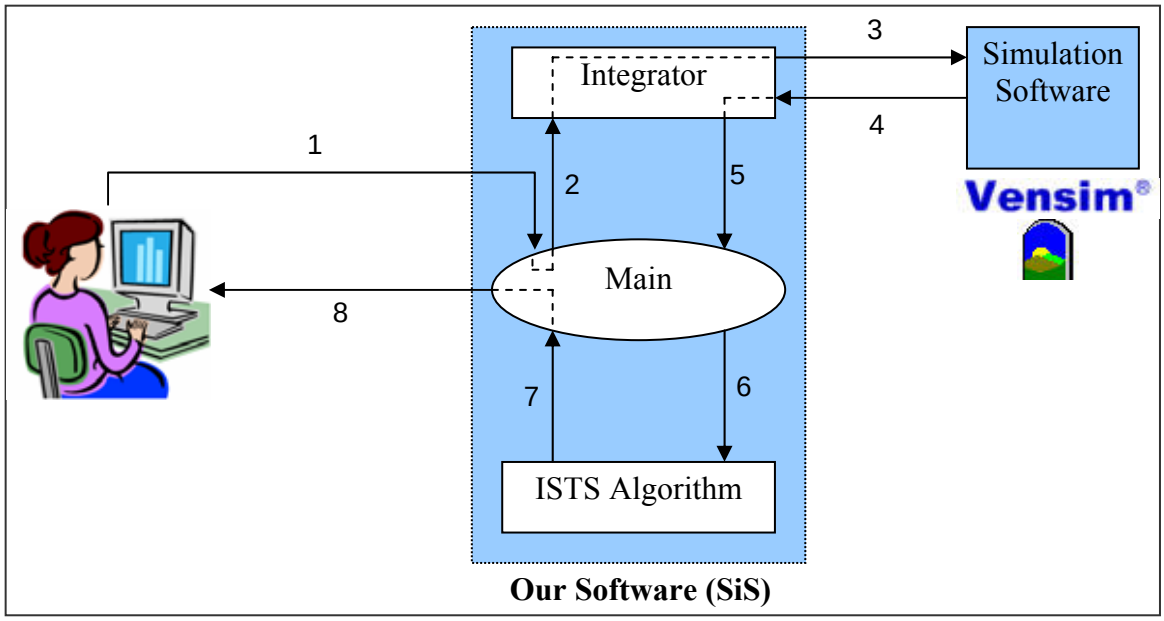

Figure 5. General Picture of the Processes in Validity Testing function.

- **Process Flow for "Validity Check":**

1. Browse a Model: User selects a simulation model which was previously created to check the validity of its output pattern.
Hypothesize Output Pattern: User select an output pattern among 25 previously defined patterns to make hypothesis for the model output.

2. "Main" part of our software takes these inputs and sends them to "integrator" to communicate with VENSIM.

3. "Integrator" part loads the model to VENSIM and give the command to start simulation.

4. "Simulation Output" that is created by VENSIM is taken by "integrator" part.

5. & 6. "Main" part takes the simulation output pattern and sends it to "ISTS Algorithm" to check its validity.

6. ISTS Algorithm checks the validity of this pattern according to the hypothesized pattern. Likelihood values of the simulation output to each previously defined pattern is obtained.

7. Result: Likelihood values and the result, whether the hypothesis passes or fails, are given to the user.

Parameter Calibration

This function simply provides user to make parameter calibration within a model. User has the option to choose a pattern type or to load real data. By choosing one of these options, the desired pattern that the model output will follow is determined.

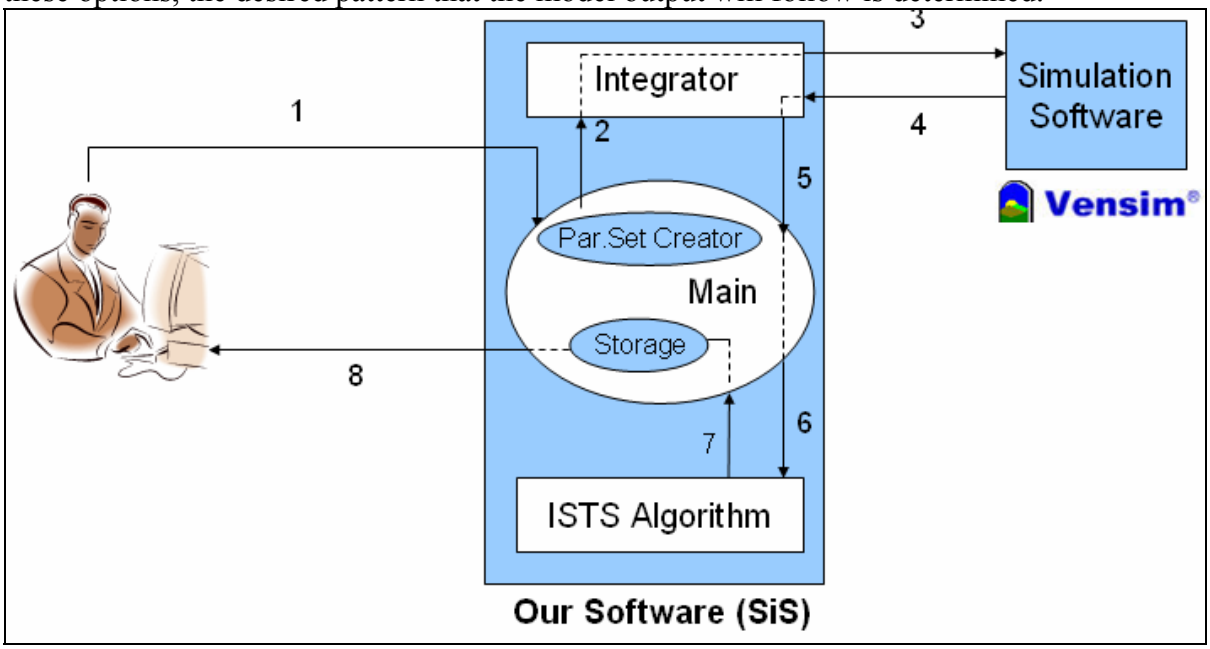

Figure 6. General Picture of the Processes in "Parameter Calibration".

- **Process Flow for "Parameter Calibration" function:**

1. User selects a previously created simulation model, enters maximum and minimum values and trial number for each parameter that he desires to calibrate. User also specifies a pattern according to which the parameter calibration will be processed.

2. "Parameter Set Creator" part of our software takes these inputs from user and creates parameter sets for all combinations of the parameter value levels. Then it starts to send one set at each time to "integrator" part.

3. "Integrator" part send this parameter set to VENSIM and make it to start simulation with these parameter values.

4. Simulation output data is taken by "integrator" part of our software.

5. & 6. Output data of VENSIM is sent to ISTS algorithm to obtain its likelihood value for the specified pattern.

7. Likelihood value of the modern output for the specified pattern is taken by ISTS Algorithm and stored in "storage" part for future comparison.

A new parameter set which was created in 2 step is sent to integrator and goes into similar processes.

8. After processing of all parameter sets, our software compare all stored likelihood values. Parameter set that gives the best likelihood value for the specified pattern is found and then sent to user as output, "best parameter set".

Illustrative Examples

Several examples that illustrate the usages of the software are presented in this section. There are four usages mode of the software which is asked from the user at the start. The usages are *Validity Testing with Default Parameters, Validity Testing by Setting Parameters, Parameter Calibration with Specified Pattern and Parameter Calibration with Input Data*

Validity Testing with Default Parameters

Figure 7. shows output of a Vensim model for a selected variable. The pattern is S-shaped exponential growth and decline to equilibrium (Growth level is greater than decline level). Hypothesis pattern selected from Figure 1 and 2 is D2GRB with convergence to nonzero. (Note that the user makes the hypothesis without seeing the graph.)

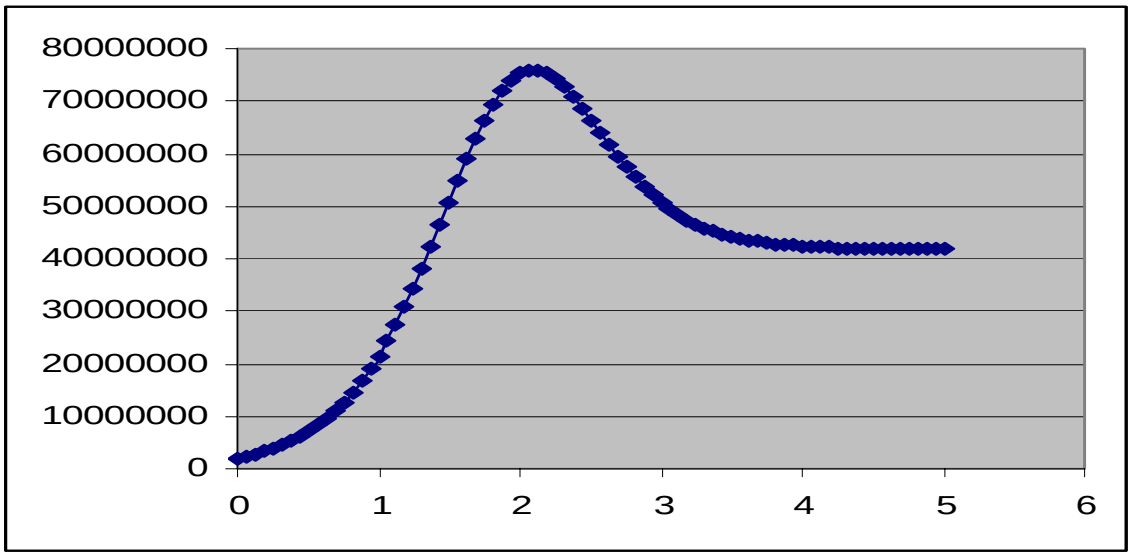

Figure 7. Simulation Output (with default base paraameters)

The state-optimized likelihood values for the pattern are tabulated in Table 1. The maximum value *1,020065* is found to be for the hypothesized class. The decision of the program is "PASSED" as expected.

ZERO0	-10	GR1DA	-16,7075	SSHDC	-25,4027	G2PED	-8,20289
CONST	-10	GR1DB	-7,26676	PEXDC	-34,4607	OSCCT	-21,6809
PLINR	-10	GR2DA	-7,08969	D1GRA	-13,9766	OSCGR	-21,6809
NLINR	-10	**GR2DB**	**1,02006**	D1GRB	-10,9594	OSCDC	-10
NEXGR	-15,7189	D1PEG	-11,0744	D2GRA	-9,18981		
SSHGR	-15,8379	D2PEG	-15,9738	D2GRB	-8,76851		
PEXGR	-29,1582	NEXDC	-17,7455	G1PED	-10,4806		

Table 1 . Likelihood Values of simulation behavior in Figure 7 compared to the D2GRB pattern (shown in Figures 1 and 2).

Validity Testing by Setting Parameters

The user changes the values of the two parameters of a Vensim model. Figure 8 shows the output of the model with default parameters and Figure 9. shows the output of the model after changing the parameters. If the user has mastery over the model, he/she expects that the pattern would be NEXGR (Negative Exponential Growth) after parameters are set to new values and without seeing the second figure, she makes the hypothesis that the pattern is NEXGR. With this usage of the software, extreme condition tests can be done easily.

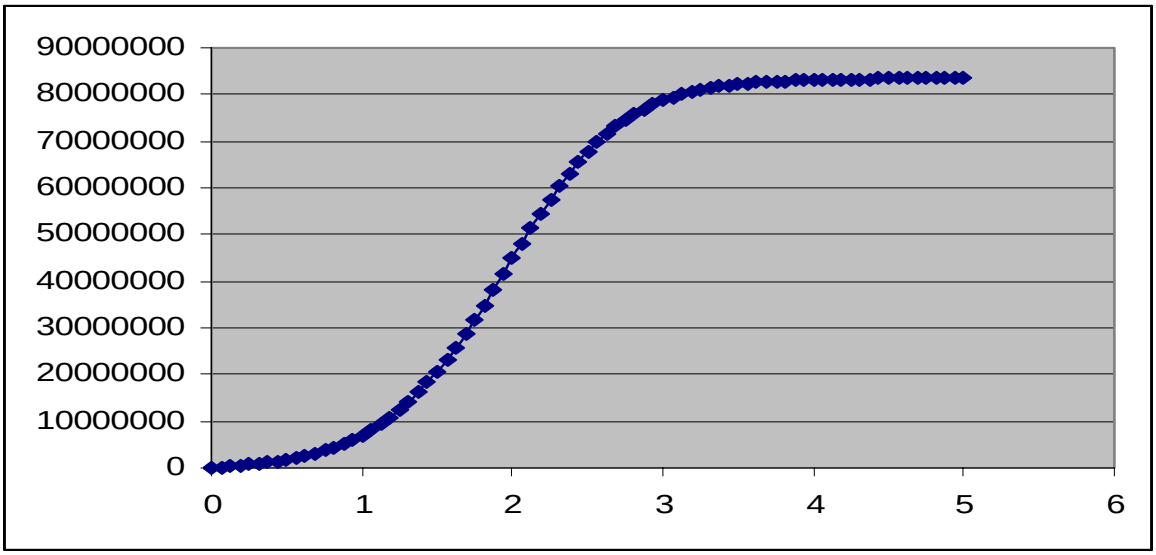

Figure 8. Simulation Output (with base parameters)

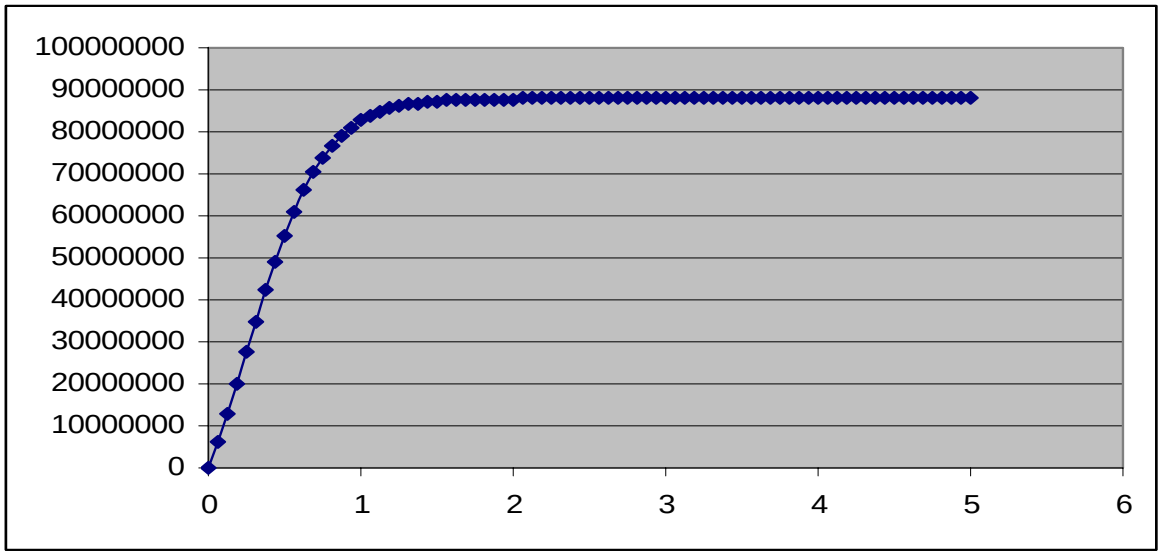

Figure 9. Simulation Output (with changed parameters)

The state-optimized likelihood values for the pattern are tabulated in Table 2 below. The maximum value *1,5198* is found to be for the hypothesized class. The decision of the program is "PASSED" as expected.

ZERO0	-10,0000	GR1DA	-7,0232	SSHDC	-22,5683	G2PED	-1,7966
CONST	-10,0000	GR1DB	-3,4047	PEXDC	-20,1837	OSCCT	-21,6809
PLINR	-10,0000	GR2DA	-4,1902	D1GRA	-4,4541	OSCGR	-21,6809
NLINR	-10,0000	GR2DB	-3,1932	D1GRB	-7,9637	OSCDC	-10,0000
NEXGR	**1,5198**	D1PEG	-14,4001	D2GRA	-2,5036		
SSHGR	-2,3409	D2PEG	-18,2043	D2GRB	-7,4449		
PEXGR	-31,9219	NEXDC	-18,2829	G1PED	-2,4259		

Table 2. Likelihood Values of simulation behavior in Figure 9 compared to the NEXGR pattern (shown in Figures 1 and 2).

Parameter Calibration with Specified Pattern

Figure 10 shows output of a Vensim model for a selected variable. The pattern is PEXGR (Positive Exponential Growth) The user specifies the calibration pattern as SSHGR (S-shaped Growth) and selects three parameters of the model and gives the range and the number of values that will be tried in the interval.

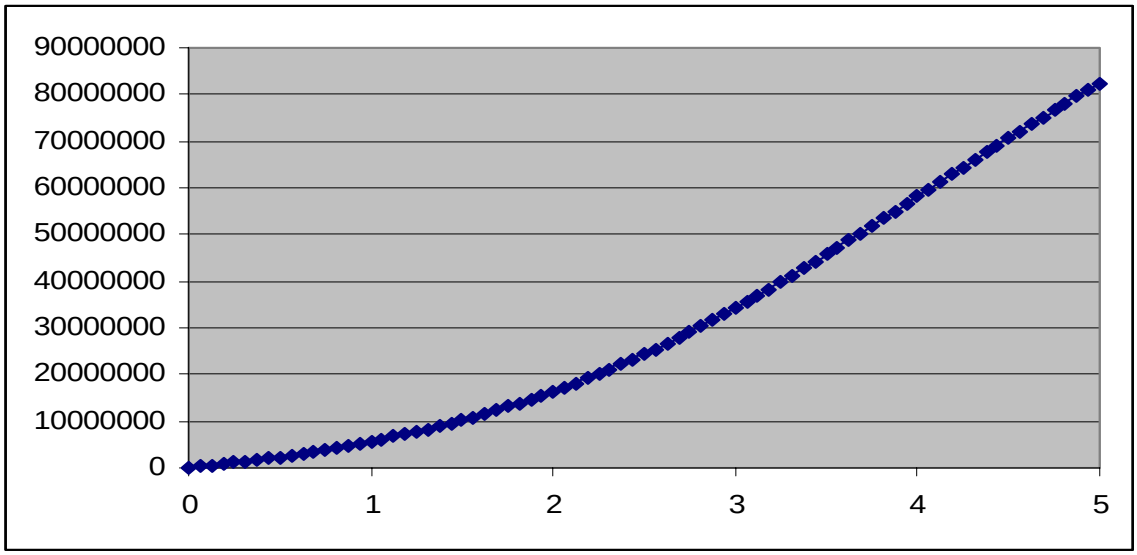

Figure 10. Simulation Output (with base parameters)

Table 3. The ranges and number of values tried for each parameter.

Selected Parameters	Min	Max	Number of Values In the Interval
1. advertising effectiveness	0	1	5
2. customer sales effectiveness	0	8	5
3. sales size	1	5	5

- **Result of the Parameter Calibration**

After simulating and automatic testing for each parameter value set (125 sets formed in this example), the best parameter set is found and Figure 11 below shows the output after the parameters are set to their best values. The pattern is SSHGR as the user specified.

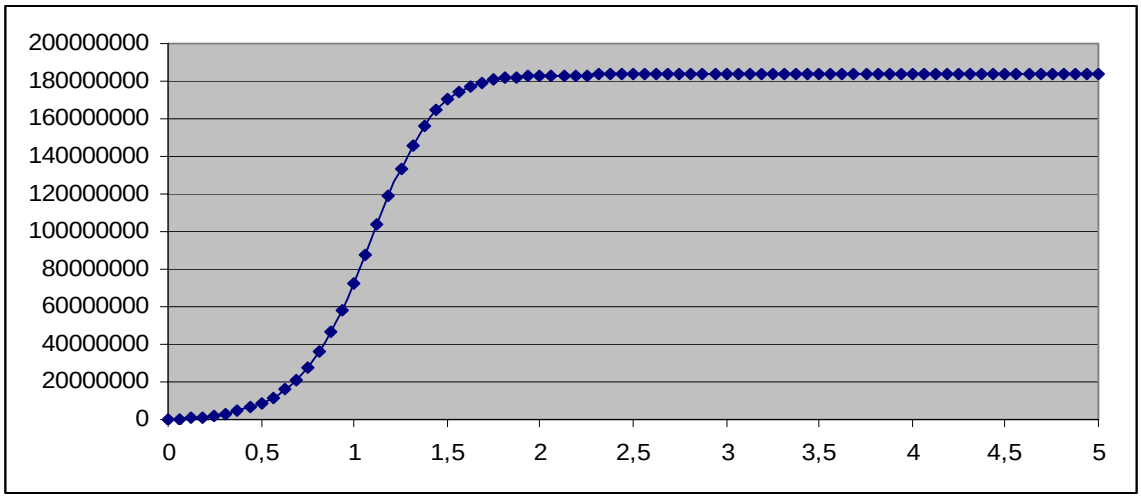

Figure 11. Simulation Output as Desired (after automated parameter calibration).

Best parameter set is 41
Best Likelihood Result: 1.2119776136254248
Best Parameter Set:
1. advertising effectiveness: 0.25
2. customer sales effectiveness: 6.0
3. sales size: 1.0

Parameter Calibration with Input Data

The user browses a real data pattern input as shown in the Figure 12. The ouput of the Vensim model is to be calibrated to the class of this input data pattern.

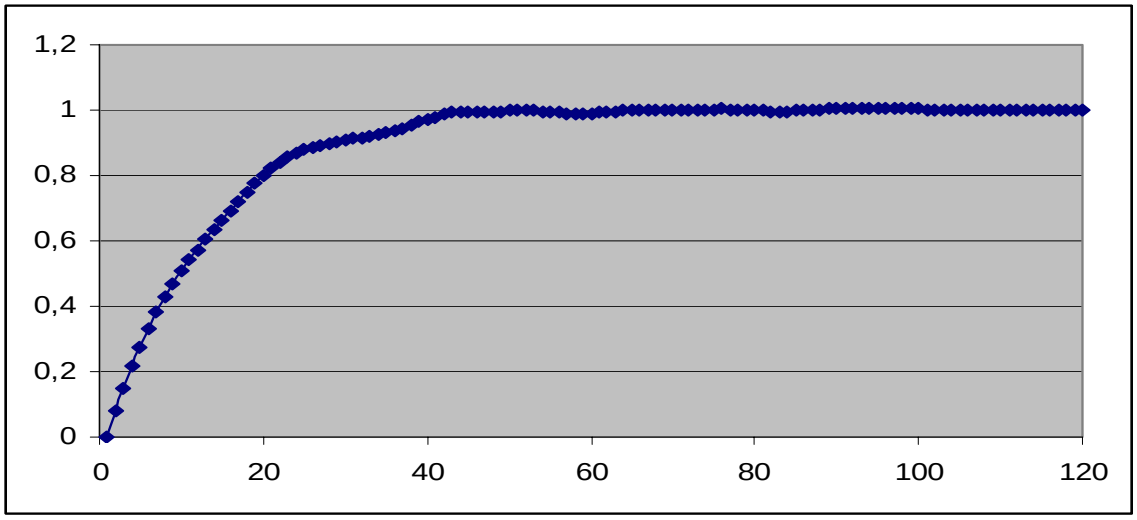

Figure 12. Input Data Pattern

Figure 14 shows the output of a Vensim Model in the base run. The output will be calibrated to NEXGR, which is the class of the input data pattern as seen on the results screen of the software in Figure 13.

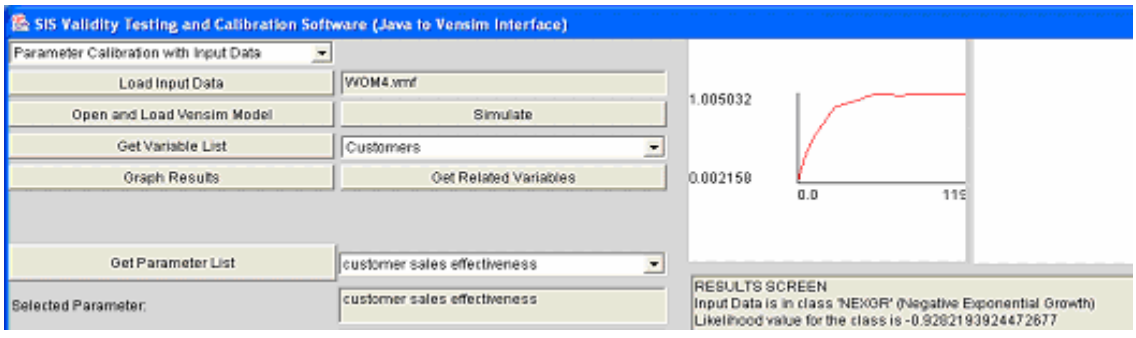

Figure 13. A view of the SiS interface during parameter calibration.

RESULTS SCREEN
Input Data is in class 'NEXGR' (Negative Exponential Growth)
Likelihood value for the class is -0.9282193924472677

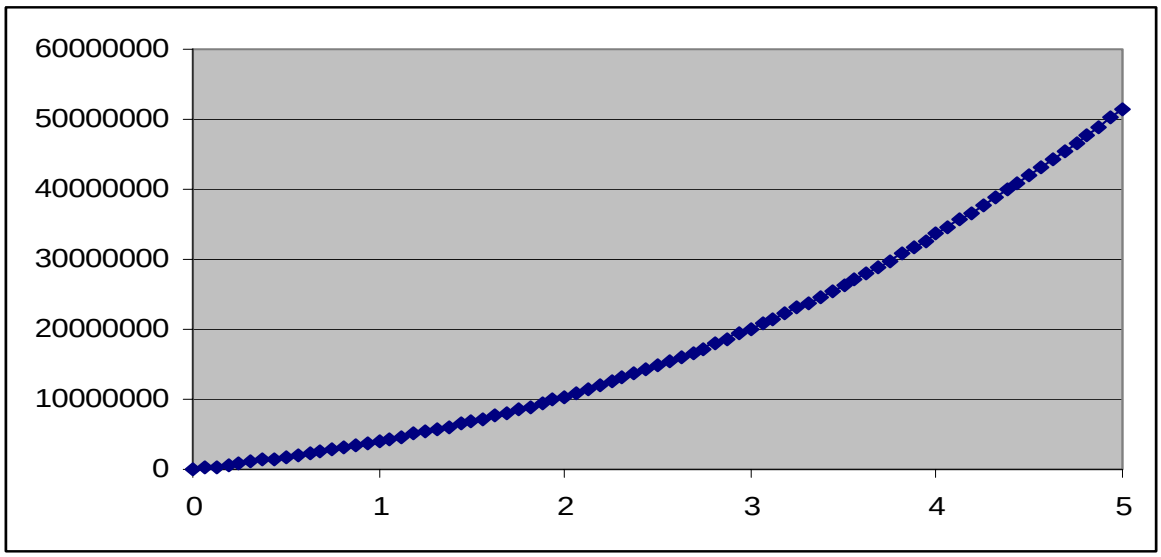

Figure 14. Simulation Output (with base parameters)

- **Result of the Parameter Calibration**

After automated testing for each parameter set (25 sets formed in this example), the best parameter set is found and the Figure 15 shows the output after the parameters are set to their best values. The pattern is NEXGR which is the class of the input data pattern.

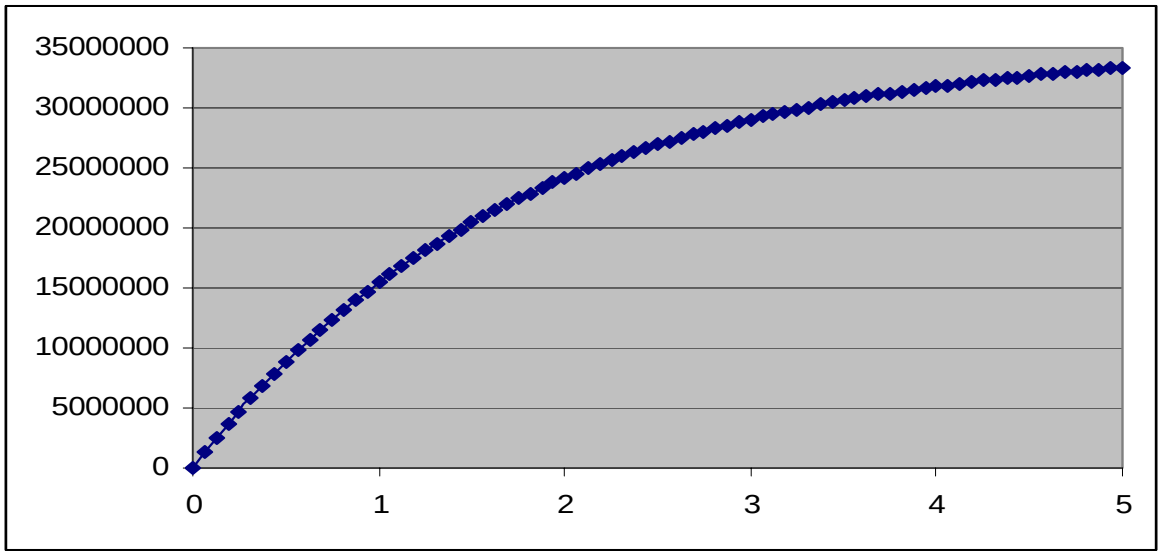

Figure 15. Simulation Output (after parameter calibration to match the input pattern)

Best parameter set is 21
Best Likelihood Result: -3.7109428620957883
Best Parameter Set:
1. advertising effectiveness: 5.0
2. customer sales effectiveness: 0.0

CONCLUSIONS

In this paper a software that seeks to partially automate validation and calibration of system dynamics models is presented. 'Validity Testing' part of the software takes the dynamic behavior generated by the model, "recognizes" it and tests if it belongs to the class hypothesized by the modeler. It is also possible to set values of selected parameters of the model and perform the test afterwards. Software is flexible in the sense that new pattern classes can be added without having to train for all the existing classes. A similar advantage is the ever improving the classification power of the software, by training it with more and more samples.

In the software developed, Hidden Markov Models (HMMs) are used for pattern recognition. Tests show that the performance of our validity testing algorithm is satisfactory, but it naturally depends on the parameters of the available HMMs. Especially oscillatory, growth-and-decline and decline-and-growth patterns is to be improved furtherby adding new samples for these classes. Training the classes with more and representative samples should improve the performance of the algorithm.

'Parameter Calibration' part of the software can be done in two different modes. In the first one, the user specifies a calibration pattern out of the existing basic classes and model output is calibrated to fit the specified class. This mode can also be used for policy improvement: the user may specify a desirable behavior pattern, select a set of policy parameters to be calibrated and the software will provide a set of policy parameter values that will yield the desired output behavior pattern. In the second mode, user browses real (input) data pattern and the model output is calibrated to fit to the class of the input data pattern. The advantage of the second mode is that the user is not required to have mastery over the model since he/she is not expected to specify a pattern. This mode in a sense offers an 'optimal' parameter estimation procedure.

REFERENCES

Balci, O., "Validation, Verification and Testing throughout the Life Cycle of a Simulation Study", *Annuals of Operations Research*, No. 53, pp. 121-173, 1994.

Barlas, Y., "Formal Aspects of Model Validity and Validation in System Dynamics", *System Dynamics Review*, Vol. 12, No. 3, pp. 183-210, 1996.

Barlas, Y., "Multiple Tests for Validation of System Dynamics Type of Simulation Models", *European Journal of Operational Research*, Vol. 42, No. 1, pp. 59-87, 1989a.

Barlas, Y., "Tests of Model Behavior that can Detect Structural Flaws: Demonstration with Simulation Experiments", In P. M. Milling and E. O. K. Zahn (Eds.), *Computer-Based Management of Complex Systems*, pp. 246-254, Berlin: Springer-Verlag, 1989b.

Barlas, Y. and Kanar, K., "Structure-oriented Behavior Tests in Model Validation", *Proceedings of 4th Systems Science European Congress*, Valencia, Spain, pp. 269-286, 1999.

Forrester, J.W., Low, G.W. and Mass, N.J., "The Debate on World Dynamics: A response to Nordhaus", *Policy Sciences 5*: pp. 169-190, 1974.

Forrester, J.W. and Senge, P.M., "Tests for Building Confidence in System Dynamics Models", *System Dynamics,* North-Holland, 1980.

He, Y. and Kundu, A. "2-D Shape Classification Using Hidden Markov Model", *IEEE Trans. Patt. Anal. Machine Intell.* Vol. PAMI-13, No. 11, pp. 1172-1184, 1991.

Kanar, K., *Structure-oriented Behavior Tests in Model Validation, M.S. thesis,* Department of Industrial Engineering, Bogaziçi University, 1999.