

# Automation of Loop Eigenvalue Elasticity Analysis

Esneyder Gonzalez Ponzon<sup>1</sup>, René Amaya<sup>2</sup>, Rubén Yie Pinedo<sup>3</sup>

<sup>1</sup> Faculty Member - Industrial Engineering Department. Universidad del Norte. Km. 5 vía Puerto Colombia. Barranquilla. Colombia. [eponzon@uninorte.edu.co](mailto:eponzon@uninorte.edu.co).

<sup>2</sup> Faculty Member - Industrial Engineering Department. Universidad del Norte. Km. 5 vía Puerto Colombia. Barranquilla. Colombia [ramaya@uninorte.edu.co](mailto:ramaya@uninorte.edu.co)

<sup>3</sup> Faculty Member - Industrial Engineering Department. Universidad del Norte. Km. 5 vía Puerto Colombia. Barranquilla. Colombia [ryie@uninorte.edu.co](mailto:ryie@uninorte.edu.co)

## ABSTRACT

In this paper, the automation of the loop eigenvalue elasticity analysis (LEEA) is proposed. LEEA is a methodology for generating a formal analysis of system dynamics models and relies on the analysis of the structure of a linearized model to infer from its eigenvalues and eigenvectors the model portions most related to a given behavior of interest. In order to apply the LEEA so far has been necessary to get involved in a manual and lengthy process, that includes among its several steps the decoding of complex functions into simplest versions and deleting some parts of the model and adding some others that are used as auxiliary variables in order to have the model generator simulation suite (such as VENSIM®) interact with numerical computation software (such as MATLAB®) back and forth to provide the LEEA analysis. This research provides an efficient tool which integrates the LEEA methodology without modifying the created models. Three models were tested and computational experiments indicate a good behavior of the tool compared with the results obtained in the existing literature.

**Keywords** eigenvalue analysis; loop eigenvalue elasticity analysis

## 1. Introduction

Lack of formal analysis to relate structure and behavior of models have long been an issue in the SD domain (Ford, 1999; Saleh, 2002; Kampmann C.E. and Oliva R, 2008).

An important approach to cover for this is provided by the loop eigenvalue elasticity analysis (LEEA), which is a method for analyzing formal models based on the interpretation of eigenvalues.

Despite the fact of being applied in many of contexts (e.g. Gonçalves et to., 2000; Saleh and Davidsen, 2001a, b; Gonçalves, 2003; Abdel-Gawad et al., 2005; Güneralp, 2006; Kampmann and Oliva, 2006; Saleh et al., 2008), LEEA remains as a tool used only by specialists in fundamental

research and has not even been incorporated in a standard software (Kampmann C.E. and Oliva R, 2008).

The two existing developed tools on dynamic systems focus on the implementation of the LEEA are: The First tool “Tool set for Loop Eigenvalue Elasticity Analysis”, Developed by the Dr. Rogelio Oliva. This tool incorporates the model designed in Vensim to after convert it in a Mathematica® format (this part is based on the Web), then the user open a file, process the information together with the base file in Mathematica® format that is downloaded from the web site and finally an excel template is used for the revision of the results. The Limitations are: (1) the special characters in variable names are not allowed. (2) The special functions are not supported and should be replaced manually with the notation of Mathematica®. (3) The dynamic functions are not supported. (4) The statements IF THEN ELSE are not allowed. (5) Macros are forbidden. (7) Arrays are not allowed.

The second tool is known like “Guide for the Implementation of the EEA”, Developed by the Dr. Burak Güneralp. Güneralp introduce a methodology based on 10 steps (Güneralp, 2005). The user must derive the model, obtain gains from pathways, build matrices, and so on. The Limitations are: (1) Despite of its applicability of this methodology to any models, the tool is not totally automated. The user has to spend long time in setting the appropriate entries of the model. (2) The desktop software is configured for three specific models. The user should take them as a guide to configure all required input files. (3) Due to the inputs of the tool it is necessary for the user to have knowledge in math partial derivatives, and management tools such as C and Matlab.

The implementation guide of the tool is good but is not fully automated and requires a lot of user intervention.

## **2. Problem and Proposed solution**

As far as we know, there are no works addressing fully automated software that allows the formal analysis of dynamic systems considering the LEEA method.

In the methodology proposed by Guneralp (2005) it is necessary to find the gain matrix manually; that is, to determine the partial derivatives that compose the gain matrix. It was also necessary to design specific software for each model in order to find the elasticities for the gain matrix. On the other hand it is necessary to fully define the direct matrix. The problem with this is that find derivatives manually implies an intensive time-consuming task.

Oliva (Kampmann and Oliva, 2006) designed a software that allows you to apply LEEA methodology. In this software, the user must manually intervene (direct manipulation of model constructs in a software suite such as Vensim ®) the model to make several changes. In addition, some models cannot be analyzed because some functions are not supported.

The handling model is necessary because the functions must adapt to a software that derives symbolically. This is a problem because there are functions whose translation into a language understood symbolically derived software is too complex or impossible.

Our main objective is to design software that minimizes the user data manipulation. The existing software's have as a main disadvantage that the users have to be experts in the specific system in order to find the state equations of it. In order to solve this problem, our software is based on numerical approximation of the state equation. To find these equations, we based our core on the determination of the derivatives numerically using simulation. Yie (2012) propose the following: For use the approximate derivative is necessary to obtain the corresponding simplest equation from a variable before replacing the value of tolerance. Numerical partial derivative of a function is defined as follows:

$$\frac{\partial f}{\partial x_i}(\bar{x}) = \frac{1}{2h} (f(\bar{x}_1, \dots, \bar{x}_{i-1}, \bar{x}_i + h, \bar{x}_{i+1}, \dots, \bar{x}_n) - f(\bar{x}_1, \dots, \bar{x}_{i-1}, \bar{x}_i - h, \bar{x}_{i+1}, \dots, \bar{x}_n)) + O(h^2)$$

Where  $h$  is the tolerance.

E.g. For the equations from industrial structures model presented in the Appendix, if we take the partial derivative  $\frac{\partial WR}{\partial IS}$ , we obtain the simplest expression for the WR (Water Reserves) variable as follows:

$$\dot{WR} = -WC$$

$$\dot{WR} = -WD * EWA$$

$$\dot{WR} = -IS * wdpi * (p1/(1 + EXP(p2 * (WR/WD/rwrc - p3))))$$

$$\dot{WR} = -IS * wdpi * (p1/(1 + EXP(p2 * (WR/(IS * wdpi)/rwrc - p3))))$$

Now, we can take the partial derivative, with respect to IS (Industrial Structures) variable:  $\frac{\partial WR}{\partial IS} =$

$$\frac{\left( \left( - (IS+h) * wdpi * \frac{p1}{1 + \text{EXP} \left( p2 * \left( \frac{WR}{rwrc} - p3 \right) \right)} \right) \right) - \left( \left( - (IS-h) * wdpi * \frac{p1}{1 + \text{EXP} \left( p2 * \left( \frac{WR}{rwrc} - p3 \right) \right)} \right) \right)}{2h}$$

### 3. Software design and implementation

The software is developed in the programming language C # on IDE Microsoft Visual Studio 2012. A project in Matlab is considered for defining the functions and for the computationally intensive calculations. The Matlab project generates a DLL file that is then referenced in the main project in .NET and from which one can invoke functions defined in Matlab. Similarly, the connection to Vensim software is made as well via DLL (we use vendll32.dll file provided by VENTANA SYSTEMS, inc).

The software receives an extension .mdl model and itself receives the .vpm extension. The .mdl model is used to overwrite the numerical partial derived and .vpm for definitions of the variables (by simulation) without passing through the .mdl file.

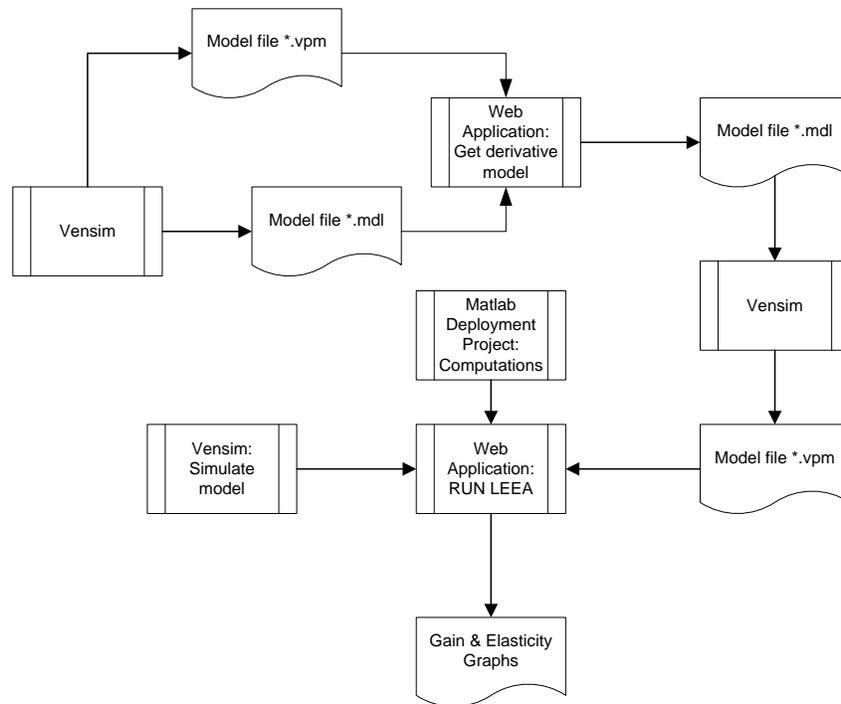


Figure 1. Software operating structure

Once the model has been derived, the user can download the new model (with the definitions of the derivatives and the gains of the pathways) to be published (using Vensim DSS software) and loaded to the system.

When loading the new model, the user selects the state variable. Then, the gain matrix is read and the SILS is calculated altogether with the elasticities loops; the direct array is set, and corresponding charts are generated by Matlab for the analysis of the user.

The implementation of the software is given under the client-server scheme. There is a computer with an IIS Web server (Internet Information Services) and the .NET Framework 4.5 to allow the access of customers from a Web browser application.

Interested persons can directly access the application from internet.<sup>1</sup>

<sup>1</sup> <http://c2c.uninorte.edu.co/LEEA/Default.aspx>

## 4. Testing tool

We selected three models in the literature to perform testing of the tool. Models correspond to those presented by Rogelio Oliva and Eric Kampmann in his own application of LEEA: Three case studies (Kampmann C. E. & R. Oliva 2006). For each of these models there are records of the results of LEEA against which we can make comparisons.

### 4.1 Industrial structures model

The first model is a simple industrial structures model. Vensim equations are provided in the appendix. Below is the SILS found, that exactly matches the feedback loops in the literature.

Loop no.	Variable sequence
1	IS,DEM
2	IS,NI
3	WR,EWA,WC
4	IS,WD,EWS,NI
5	IS,WD,WC,EWS,NI
6	IS,WD,EWA,WC,EWS,NI

Table 1. Feedback loops in the SILS in Industrial structures model

In figure 2 you can see the progress eigenvalues over time. To find out if two series are statistically equal, a test for paired comparisons is made.

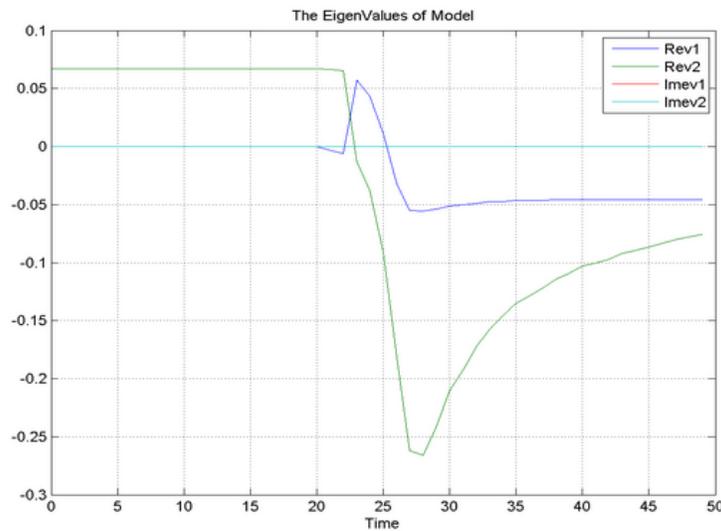


Figure 2. Progress Eigenvalues in Industrial structures model

We define statistical hypothesis as follows:  $H_0: \mu_1 = \mu_2$  y  $H_1: \mu_1 \neq \mu_2$ . The values obtained for the differences of the series for the first eigenvalue are:  $\bar{d}_1 = 0.00281$ ,  $s_d = 0.01508$ ,  $t_0 = 1.31621$ ,

$t_{0.025,49} = 2.01$ . This test is done for a significance level of  $\alpha = 0.05$ . Then, as the absolute value of  $t$  it is not greater than  $t_{\alpha/2, n-1}$  we don't reject the null hypothesis. The series are equals.

The values obtained for the differences of the series for the second eigenvalue are:  $\bar{d}_2 = 0.003034$ ,  $s_d = 0.01537$ ,  $t_0 = 1.396$ ,  $t_{0.025,49} = 2.01$ . Then, as the absolute value of  $t$  it is not greater than  $t_{\alpha/2, n-1}$  we don't reject the null hypothesis. The series are equals.

#### 4.2 Lorenz model

The Lorenz model is an example of deterministic chaos. This model has three stock variables. Below the SILS found.

Loop no.	Variable sequence
1	X,dx
2	Y,dy
3	Z,dz
4	X,dy,Y,dx
5	Y,dz,Z,dy
6	X,dz,Z,dy,Y,dx

Table 2. Feedback loops in the SILS in Lorenz Model

These loops exactly match the feedback loops in the literature. In figure 3 you can see the progress eigenvalues over time

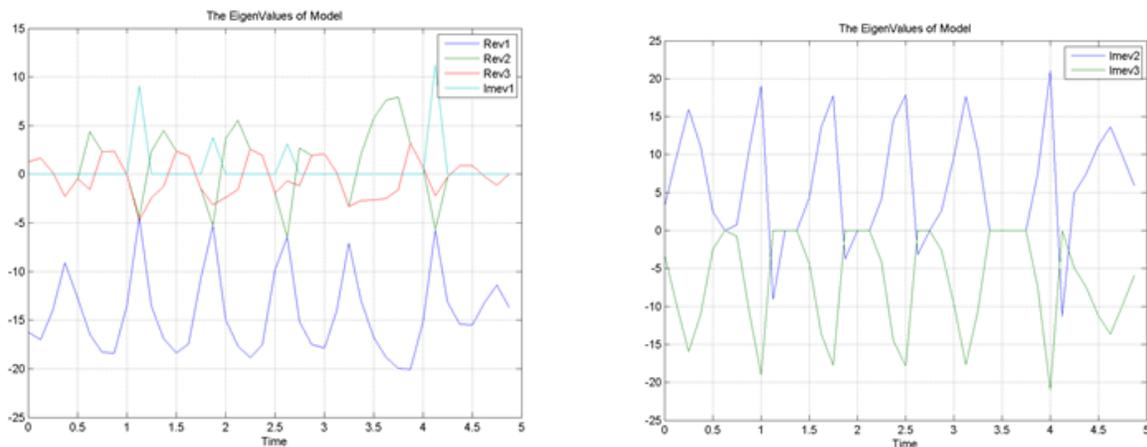


Figure 3. Progress Eigenvalues in Lorenz model

The values obtained for the differences of the series for the first eigenvalue are:  $\bar{d}_1 = 0.232$ ,  $s_d = 1.463$ ,  $t_0 = 1.001$ ,  $t_{0.025,39} = 2.02$ . This test is done for a significance level of  $\alpha = 0.05$ , we don't reject the null hypothesis.

The values obtained for the differences of the series for the second eigenvalue are:  $\bar{d}_2 = 0.0000417$ ,  $s_d = 0.00041$ ,  $t_0 = 0.634$ ,  $t_{0.025,39} = 2.02$ .for the third eigenvalue we have:  $\bar{d}_3 = 0.231$ ,  $s_d = 1.463$ ,  $t_0 = 1.001$ ,  $t_{0.025,39} = 2.02$ . We don't reject the null hypothesis.

#### 4.3 Mass Model

The third model used to validate the tool is important because it is fairly large model and presents many oscillations. Below the SILS found:

Loop no.	Variable sequence
1	ANVC,chANVC
2	AOK,chAOK
3	APR,chAPR
4	Vac,HR
5	K,KD
6	KOO,KA
7	L,TR
8	ANVC,NVC,chANVC
9	AOK,OK,chAOK
10	ANVC,RDOL,MLDOL,NVC,chANVC
11	AOK,RDKO,MKDOOFC,OK,chAOK
12	KOO,IKOC,RDKO,MKDOOFC,OK
13	AOK,DKOO,IKOC,RDKO,MKDOOFC,OK,chAOK
14	KOO,KA,K,IKOC,RDKO,MKDOOFC,OK
15	Vac,IVC,RDOL,MLDOL,NVC
16	ANVC,DVac,IVC,RDOL,MLDOL,NVC,chANVC
17	L,IVC,RDOL,MLDOL,NVC,Vac,HR
18	BL,BLR,BLMS,SR
19	Inv,INVR,IMS,SR
20	L,LR,MLDDE,ADE,TR
21	K,KR,MKDLC,ALK,KD
22	APR,PRR,KUF,EK,Normz EK,PR,chAPR
23	APR,DProd,PRR,KUF,EK,Normz EK,PR,chAPR
24	Inv,DProd,PRR,KUF,EK,Normz EK,PR
25	APR,SR,Inv,DProd,PRR,KUF,EK,Normz EK,PR,chAPR
26	APR,SR,BL,DProd,PRR,KUF,EK,Normz EK,PR,chAPR

27	APR,DBL,DProd,PRR,KUF,EK,Normz EK,PR,chAPR
28	APR,DBL,BLR,BLMS,SR,BL,DProd,PRR,KUF,EK,Normz EK,PR,chAPR
29	APR,DInv,DProd,PRR,KUF,EK,Normz EK,PR,chAPR
30	APR,DInv,INVR,IMS,SR,BL,DProd,PRR,KUF,EK,Normz EK,PR,chAPR
31	K,EK,Normz EK,PR,Inv,DProd,DK,KR,MKDLC,ALK,KD
32	KOO,KA,K,EK,Normz EK,PR,Inv,DProd,DK,IKOC,RDKO,MKDOOFC,OK
33	K,EK,Normz EK,PR,EGRP,MGDK,DK,KR,MKDLC,ALK,KD
34	APR,EGRP,MGDK,DK,KR,MKDLC,ALK,KD,K,EK,Normz EK,PR,chAPR
35	L,EL,Normz EL,PR,Inv,DProd,DL,LR,MLDDE,ADE,TR
36	Vac,HR,L,EL,Normz EL,PR,Inv,DProd,DL,IVC,RDOL,MLDOL,NVC
37	L,EL,Normz EL,PR,EGRP,MGDL,DL,LR,MLDDE,ADE,TR
38	APR,PRR,MLD OT,MHY,RLWW,EL,Normz EL,PR,chAPR

Table 3. Feedback loops in the SILS in Mass Model

The tool found 38 loops in the SILS, and these loops exactly matches the feedback loops in the literature. In figure 4 you can see the progress eigenvalues over time

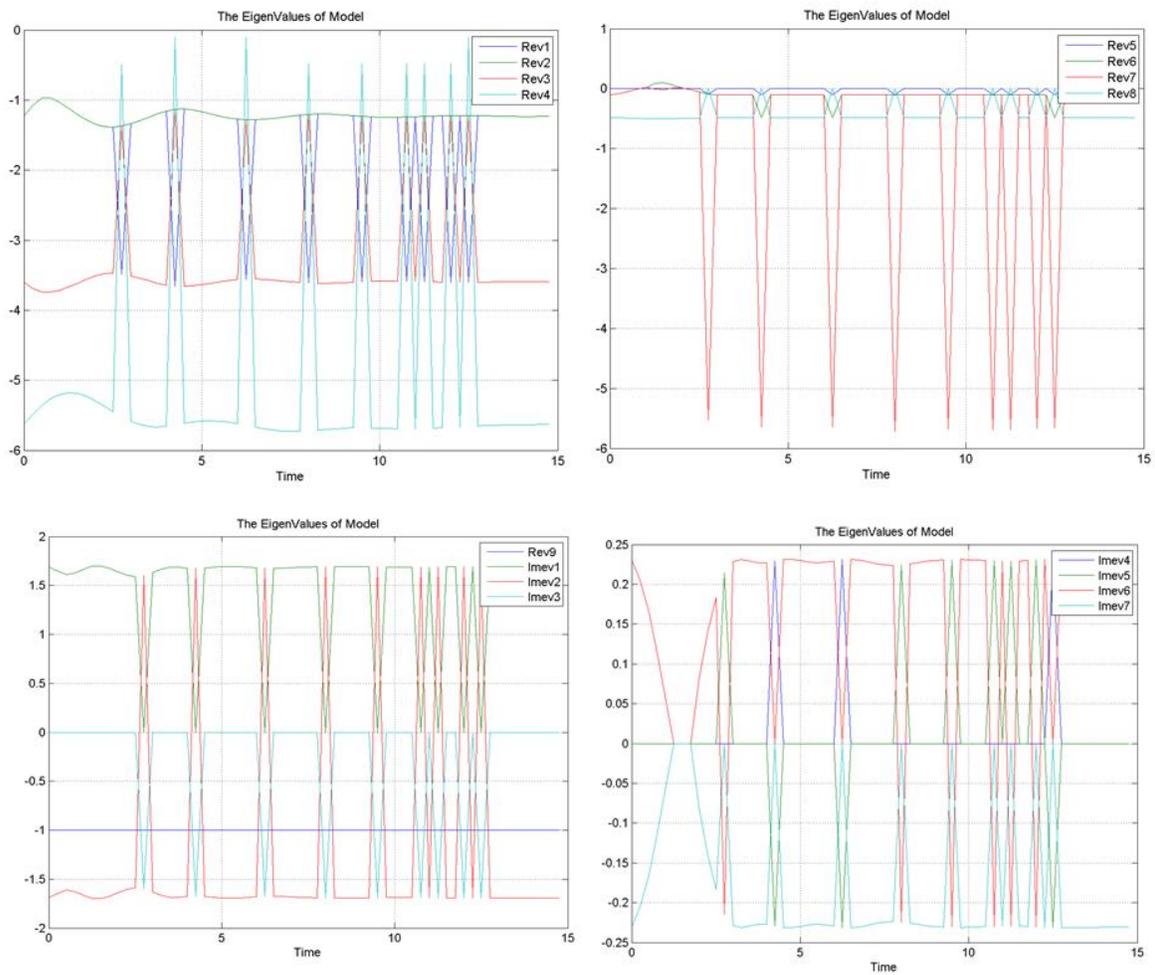


Figure 4. Progress Eigenvalues in Mass model

The values obtained for the differences of the series for the first eigenvalue are:  $\bar{d}_1 = -0.054$ ,  $s_d = 0.143$ ,  $t_0 = -1.194$ ,  $t_{0.025,9} = 2.26$ . This test is done for a significance level of  $\alpha = 0.05$ . Then, as the absolute value of  $t$  it is not greater than  $t_{\alpha/2, n-1}$  we don't reject the null hypothesis. The series are equals.

The values obtained for the differences of the series for the first eigenvalue are:  $\bar{d}_2 = 0.131$ ,  $s_d = 0.731$ ,  $t_0 = 0.567$ ,  $t_{0.025,9} = 2.26$ . For the third eigenvalue we have:  $\bar{d}_3 = -0.431$ ,  $s_d = 0.769$ ,  $t_0 = -1.77$ ,  $t_{0.025,9} = 2.26$ . For the fourth eigenvalue:  $\bar{d}_4 = -0.521$ ,  $s_d = 1.592$ ,  $t_0 = -1.04$ ,  $t_{0.025,9} = 2.26$ . In each case we don't reject the null hypothesis. The series are equals.

Previous tests correspond to the values with different values over time.

#### 4.4 Computing Times

We present computational times for models with different numbers of variables and simulation time. The Computing time was measured on a PC Intel® Core™ i7 2.20 GHz (8 GB RAM) running with Windows.

The number of state variables defines the order of model, which, in turn increase the size of gain matrix. The number of total variables in most cases it involves more pathways and perhaps more loops.

Number of state variables	Number of total variables	Number of loops in the SILS	Time (minutes)
2	15	6	0.23
3	9	6	0.2
3	25	16	2.3
9	81	38	4.42
17	227	37	15.1

Table 4. Computing Times

The above results show that computational time increase meanly by both numbers of state and total variables. For smaller models the computational time is less than a minute. For large models is greater than then minutes.

## Conclusions

This research was focused on the implementation of a software tool that applies the LEEA (Loop Elasticity Eigenvalue Analysis) methodology for automated large-scale models. The existing tools have required from the users depth knowledge on the model, deriving expressions, or limiting the model with some specific functions to be used in a specialized software, etc.

The software is developed in C # language. It is connected by DLL with a project in Matlab for computational intensive calculations and receives from Vensim (for DLL) the simulated values for the model.

For the validation of the tool, three existing models in the literature have been tested and statistical and graph theory validation was made. The results were good, even for larger models.

Such as contribution, this work gives a more accessible tool for the scientific community. The software receives a model from Vensim without constraints (it can be used all functions and settings) allowing the user to modify or delete the model functions due to effects of compatibility. Then, the major contribution of this research corresponds to the complete automation of LEEA and the generation of an open source tool for users.

## References

- Abdel-Gawad A, Abdel-Aleem B, Saleh M, Davidsen P. 2005. Identifying dominant behavior patterns, links and loops: automated eigenvalue analysis of system dynamics models. In Proceedings of the Int. System Dynamics Conference. System Dynamics Society: Boston.
- Ford D. 1999. A behavioral approach to feedback loop dominance analysis. *System Dynamics Review* 15(1): 3-36.
- Gonçalves P, Lerpattarapong C, Hines JH. 2000. Implementing formal model analysis. In Proceedings of the Int. System Dynamics Conference. System Dynamics Society: Bergen, Norway.
- Gonçalves P. 2003. Demand bubbles and phantom orders in supply chains. PhD Thesis. Sloan School of Management, Mass. Inst. of Technology: Cambridge, MA.

- Güneralp B. 2005. Progress in eigenvalue elasticity analysis as a coherent loop dominance analysis tool. In *Proceedings of the 2005 International System Dynamics Conference*, Boston, MA. System Dynamics Society: Albany, NY
- Leithold Louis. 1998. *El cálculo, 7ª ed.* Oxford University Press, México.
- Kampmann C. E. & Oliva R. 2006. Loop eigenvalue elasticity analysis: three case studies. *System Dynamics Review* 22(2): 141–162.
- Saleh M, Davidsen P. 2001a. The origins of business cycles. In *Proceedings of the Int. System Dynamics Conference*. System Dynamics Society: Atlanta.
- Saleh M, Davidsen P. 2001b. The origins of behavior patterns. In *Proceedings of the Int. System Dynamics Conference*. System Dynamics Society: Atlanta.
- Saleh MM. 2002. The characterization of model behavior and its causal foundation. PhD dissertation, University of Bergen, Bergen, Norway.
- Saleh M, Oliva R, Davidsen P, Kampmann CE. 2008. A comprehensive analytical approach for policy analysis of system dynamics models. Working Paper. Mays Business School, Texas A&M University.
- Yie Pinedo, R. (2012). *Herramienta computacional para el análisis de sensibilidad de sistemas de simulación continua usando el método LEEA (loop eigenvalue elasticity analysis)* Tesis de Magíster en Ingeniería Industrial, Universidad del Norte, Barranquilla, Colombia.

## Appendix

### Equations for simple industrial structures model

$$EWS = p1 / (1 + \text{EXP}(p2 * (WC/WD - p3)))$$

Effect of water resources on new structures

$$EWA = p1 / (1 + \text{EXP}(p2 * (WR/WD/rwrc - p3)))$$

Effect of water availability on consumption

$$p2 = -6.98405$$

Coefficient in function

$$p1 = 1$$

Coefficient in function

$$rwrc = 10$$

df= 0.05 Reference water resource coverage  
 Demolition fraction  
 p3= 0.4715 Coefficient in function  
 WR= INTEG ( -WC, 10000)  
 WC= WD\*EWA  
 DEM= IS\*df Demolition  
 IS= INTEG (+NI-DEM, 10) Industrial structures  
 NI= IS\*EWS\*ng New industries  
 ng= 0.12 Normal growth rate  
 WD= IS\*wdpi Water demand  
 wdpi= 10 Water demand per industry