

## Application of fuzzy tools to the automatic analysis of system dynamics models: an example of World3

Margarita Mediavilla\*, Luis J. Miguel\*, Pedro Retortillo\*,  
Carlos de Castro\*\*, L. Felipe Blázquez\*\*\*

\**Department of Automatic Control and Systems Engineering, University of Valladolid, E.I.I., Paseo del Cauce s/n, 47011 Valladolid, Spain. (Tel: 34 983423545; e-mail: {marga,ljmiguel}@eii.uva.es).*

\*\**Department of Applied Physics, University of Valladolid, E.I.I., Paseo del Cauce s/n, 47011 Valladolid, Spain. (Tel: 34 983423136; e-mail: ccastro@termo.uva.es).*

\*\*\**Department of Electrical and Automatic Control and Systems Engineering, University of León, E.I.I.I., Campus de Vegazana s/n, 24071 León, Spain. (Tel: 34 987293471; e-mail: diefbq@unileon.es).*

---

**Abstract:** Even the simple run of a medium size system dynamics model can be a cumbersome process, since the uncertainty of the parameters forces the modeler to consider many runs before being confident of how the model behaves. System dynamics simulation packages include some analysis tools, but there is a lack of a commonly used analysis tool such as fuzzy clustering. In this paper, we explore the possibilities of a programming language, Matlab, and its simulation tool, Simulink, for the above mentioned possibilities. These languages enable the development of customized analysis tools at a very low programming cost. World3 model is a widely known example of a large model whose analysis becomes a difficult task. We have programmed this model in Simulink and, afterwards, some tools, such as fuzzy clustering and pathway participation metric (PPM), have been applied to show their potential capacity to facilitate model simulation analyses.

**Keywords:** System dynamics, World3, fuzzy logic, fuzzy clustering, pathway participation metric.

---

### 1. INTRODUCTION

Systems thinking (Checkland, 1981) is a process for understanding how things, or subsystems, influence each other as parts of a set within a whole or integrated system. It is an approach for problem solving by viewing problems as parts of an overall system rather than reacting to a specific part (Ackoff, 2010). Systems thinking examples include ecosystems and socioeconomic organizations. System dynamics (Forrester, 1961) is a methodology based on systems thinking to understand and model the behavior and relations of complex dynamic systems. System dynamics uses various control factors, such as feedback loops and time delays, to represent how the system reacts and behaves towards exogenous inputs. It can be a helpful tool to assist policy and decision makers when the behavior of a system is complex and dynamic.

System dynamics modeling and simulation is an exciting process that helps the modeler to get a better understanding of the problems involved. However, when the model is built and run, the user does not have so many instruments to test what that model is really showing. A handful of simulation runs do not give much information when dealing with large models with many stock variables, nonlinear dynamics and a high degree of uncertainty in the parameters; while the tools

for analyzing large scale models are not very well developed. Some rely on intuitive approaches based on experience or, for example, on the notion of system archetypes (Güneralp, 2006); others focus on bounding the structure of the model with the observed behavior, using methods like the eigenvalue elasticity analysis described by Forrester and refined recently in Kampmann and Oliva (2006). This approach uses linear systems theory to break the behavior down into simple modes, each of which corresponds to an eigenvalue. Measuring how much a given eigenvalue changes, using a small change in a link in the model, gives an indication of how much that link contributes to the said behavior mode. The main drawback of this method is the fact that eigenvalues are meaningful only for linear models or valid linearizations, which, on many occasions, are not possible. The pathway participation metric (PPM) (Mojtahedzadeh *et al.*, 2004) identifies the structure that is most influential in affecting the qualitative time path of a given variable. The main strength of PPM is that it does not require eigenvalues to be calculated.

These two approaches focus on linking the temporal behavior observed in the simulations of the model with its structure, focusing on what part of the system's structure contributes most to a particular pattern of behavior. However, they do not pay so much attention to the uncertainties and parameter variations of the models, which may significantly change the

system's behavior. Ford and Flynn (2004) use a pragmatic approach called screening that also focuses on detecting what part of the model structure contributes to the observed behavior, but does not use eigenvalues or dominant loops, and pays more attention to the uncertainty of parameters. Part of the screening process is done in Vensim using the sensibility tool, but another part must be done with customized software.

Knowing what part of the model structure contributes to the observed behavior is interesting because it helps, as Ford describes, to "create system stories" or correct explanations of how influential pieces of structure give rise to behavior, and helps managers to understand the systems they manage. However, prior to the construction of such explanations, the modeler would need to know how his/her system really does behave. This is not an easy task, since the uncertainty of the parameters is high and the possible variations in the values of the parameters multiplied by the number of parameters gives rise to an enormous number of possible behaviors that the modeler would have to test in order to really know the model.

In this paper, we have used some tools taken from artificial intelligence and control engineering to test system dynamics models. We decided to use a simulation program widely used in engineering: the Matlab-Simulink platform; and a highly complex model: the World3-2003 (Meadows *et al.*, 2004). The fact that Simulink, which is a simulation language, works with the Matlab programming platform, gives rise to an enormous number of possibilities for designing customized software at a very low programming cost with very interesting results, as we show in this paper. Our aim is to make the model analysis easier by automating the process via the programming ability of Matlab and using some well known analysis tools in artificial intelligence.

The typical analysis of a system dynamics models would be as follows: the modeler makes the model, decides the most adequate parameters based on the available data, and takes this as the reference run. Then, he/she changes the policies, for example, one at a time, and observes the results via plots of the output variables. The analysis of the World3 model that Meadows *et al.*, (2004) carried out is made that way: first, a base scenario is described fitting the parameters to the observed tendencies; then, the second scenario is run changing one policy, then the first and the second policies are introduced and run, and this is called the third scenario. The process continues by introducing one more policy in each one of the 13 scenarios. In this process, it is the modeler who decides the combination of policies using his/her own criteria.

A more sophisticated approach to the model analysis would involve trying all the policies at the same time and varying the parameters randomly inside some confidence intervals. Some software distributor (SD) packages do this, and the results are plots which offer interesting information: we can see the range of variation of the results, but all the results are plotted together and we do not know exactly what plot corresponds to what parameter. Moreover, in the case where several parameters or policies are tested at the same time, it is difficult to disaggregate the effects of each one of them

without a more careful analysis of the data. It would be very desirable to count on a programming language that would enable all kinds of manipulations of these data, so that some relevant information could be extracted, such as a statistical treatment of the output, or a correlation analysis that would respond to the following questions: If I take into account several parameters which might vary and several policies which might be applied, what are the four or five most common results of the simulations? Which parameter most influences the model output, or a particular characteristic of the output, such as the maximum value, the slope, standard deviation, etc.?

In addition, the extraction of all those data requires the performance of many simulation runs and treatment of many data. The programming ability of the Matlab Simulink platform enables us to program the multiple runs of the simulations. This is very important because making many manual runs of a simulation with different parameters is a cumbersome process that often confuses the programmer instead of helping him/her extract a conclusion. We would also like the computer to automatically inspect the graphs for us, since we normally use them as results, and a human evaluates a graph and extracts a conclusion such as "the result is good or bad" based on heuristic criteria. It would be nice if the computer could analyze the characteristics of the curves for us, saying, for example: "out of 1000 simulations, 15% were considered bad results".

All these features: automatic programming of runs, automatic inspection of plots and numerical treatment of data, are obtained using the techniques described in this paper. The paper starts with a brief description of the World3 model and its translation into Simulink-Matlab. In Section 3, we describe the results of using screening. In Section 4, we describe more sophisticated techniques taken from artificial intelligence, such as fuzzy logic and clustering, and show the results of the analysis of World3. Section 5 gathers the conclusions and, finally, some references are provided.

## 2. THE WORLD3 MODEL

### 2.1 World3 and its scenarios

The World3 model described in Meadows *et al.*, (2004) is similar to the model of the previous book presented by Meadows *et al.*, (1992), which was an extension of the much simpler World3 model described in Meadows *et al.*, (1972). The 2004 model is a lot more complex than the 1972 one, including 41 levels, more than 60 parameters and several basic loops: population, agriculture, service capital, industry, non-renewable resources, persistent pollution, etc. The scenarios described in the book are based on the application of several worldwide policies (such as the population stagnation, the attenuation of pollution, etc.) in a particular year. Some have criticized this introduction of policies, which assumes a worldwide authority capable of enforcing them (Acharya and Saeed, 1996), but we are not entering that debate, since our main objective is to prove our analysis methods on a complex and known model.

We work with the first seven scenarios of Meadows *et al.*, (2004) and compare our results with them. The scenarios and parameters of table 1 that define them are the following:

Scenario 1. The standard run, with the parameters that follow the observed tendencies and the known natural resources to date.

Scenario 2. The natural resources are doubled (parameter 1 changes).

Scenario 3. More accessible non-renewable resources and pollution control technology. In this scenario, they assume the same ample resource supply as in scenario 2, as well an increasingly effective pollution control technology, which can reduce the amount of pollution generated by unit of output by 4% per year (parameters 1 and 6 change).

Scenario 4. More accessible non-renewable resources, pollution control and land yield enhancement. A set of technologies is added to pollution control technology to greatly increase the food per unit of land (parameters 1, 10 and 6 change).

Scenario 5. More accessible non-renewable resources, pollution control technology, land yield enhancement and land erosion protection. Land preservation technology is added (parameters 1, 10, 6 and 4 change).

Scenario 6. More accessible non-renewable resources, pollution control technology, land yield enhancement, land erosion protection and resource efficiency technology. Developing powerful technologies for pollution abatement, land yield, enhancement, land protection and conservation of non-renewable resources all at once. All these things are assumed to involve cost and take 20 years to be implemented fully (parameters 1, 10, 6, 5, 4 and 3 change).

Scenario 7. Added to the previous policies, after 2002, all couples decide to limit their family size to 2 children, but because of the momentum, the population continues to grow for another generation. The delay time in the application of policies goes down from 20 to 10 (parameters 1, 10, 5, 6, 4, 3, 12, 8 and 9 change).

### 2.2 World3 in Simulink

System dynamics software packages, i.e., Stella, Vensim and Powesim, are, without a doubt, the easiest tools for system dynamics modeling. Other simulation languages, such as those used in control engineering, do the same simulations with different languages. Those packages might be a bit less clear, but they have other advantages, as we aim to show in this paper. For those who would like to explore the possibilities of these tools, it could be useful to look at Fig. 1, where we compare the graphical implementation of a stock with two flows in Vensim and Simulink. The main difference is in the representation of stocks, i.e., state variables. In the block diagram, the variables are operated as they “flow” between the blocks. We can compare the representation of a simple stock-flow model in a system dynamics language (a) and in Simulink (b).

Table 1. Parameter values and ranges of the World3 model that have been used in the scenarios

Parameters	Value of base scenario	Range in scenarios 1 to 7
1 nr resources init	$1 \cdot 10^{12}$	$1 \cdot 10^{12} - 2 \cdot 10^{12}$
2 in x146	1.5	1.5
3 agr inp 2	2	2 -3
4 y LYCM	0	0 - 0.05
5 y NRCM	0	0 - -0.05
6 y ppoll	0	0 - -0.05
7 in x159	400	400
8 in TDD	20	10

Application times of policies		
9 t_policy_year	4000	1995 - 2020
10 t_air poll_time	4000	1995- 2020
11 t_land_fert_time	4000	1995-2020
12 t_fcaor_time	4000	1995-2020
13 t_zero_pop_grow_time	4000	4000
14 t_ind_equil_time	4000	4000

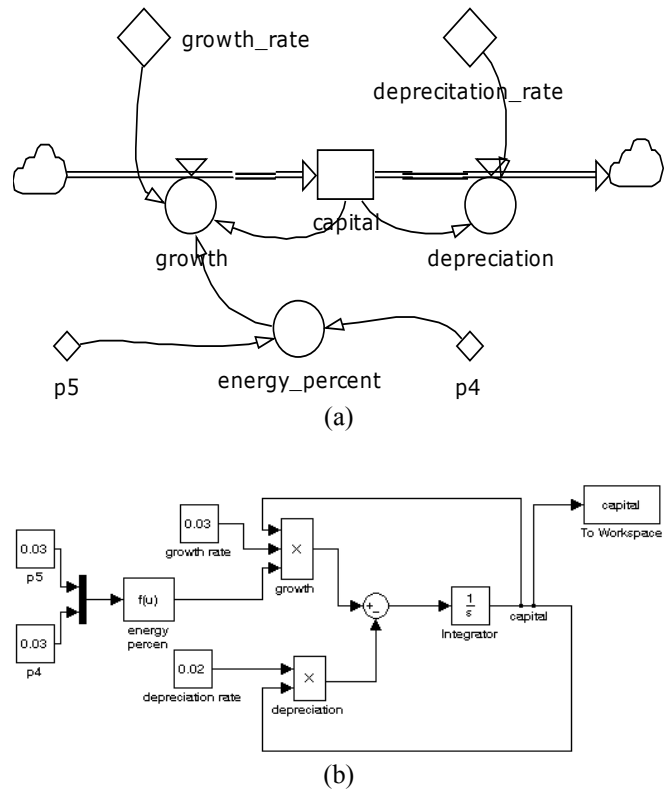


Fig. 1: comparison of a Forrester diagram (a) and the equivalent block diagram (b) implemented in Simulink.

Figure 2 compares the results of the original World3 simulation given by Stella, and the results of our translation of this model into Simulink. Both simulations correspond with scenario 1, and we can see that the differences between the results of our model and the original one are negligible.

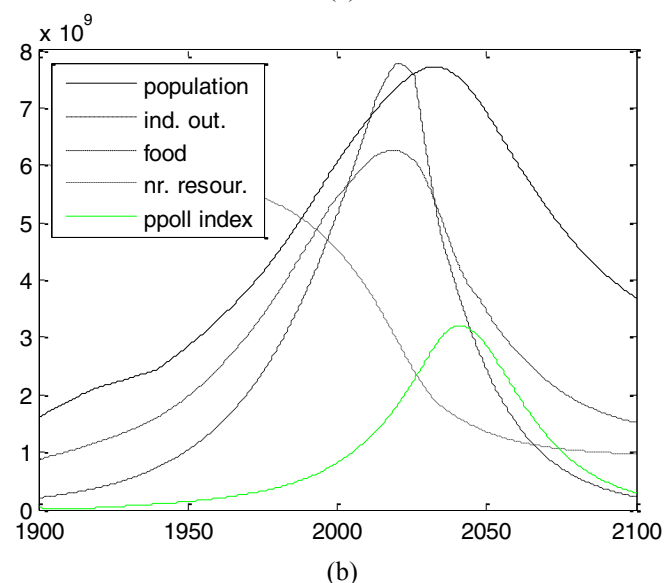
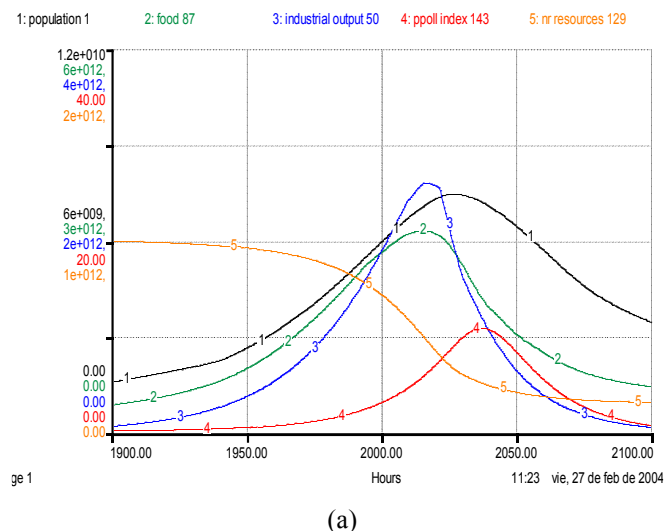


Fig. 2. Stella graph of World3 simulation (a), and Matlab-Simulink graph of World3 simulation (b).

### 3. ANALYSIS OF SYSTEM DYNAMICS MODELS USING SCREENING

The screening technique, described by Ford and Flynn (2004), can also be implemented at minimum programming cost using Matlab, since most of the mathematical calculations are already programmed as toolboxes. This technique consists of calculating the correlation between the parameters and the output at each time of the simulation. We can see the results of applying this technique to World3 in Fig. 3. We have used the application times of the policies listed in table 1 as parameters. The output is the ecological footprint. The result of the screening is an estimation of how much each of the parameters influences the output at each time instant. If the value of the correlation is positive, it means that the later the policy is applied, the bigger the ecological footprint will be. Therefore, if the value is positive, that particular policy is good for reducing the ecological footprint.

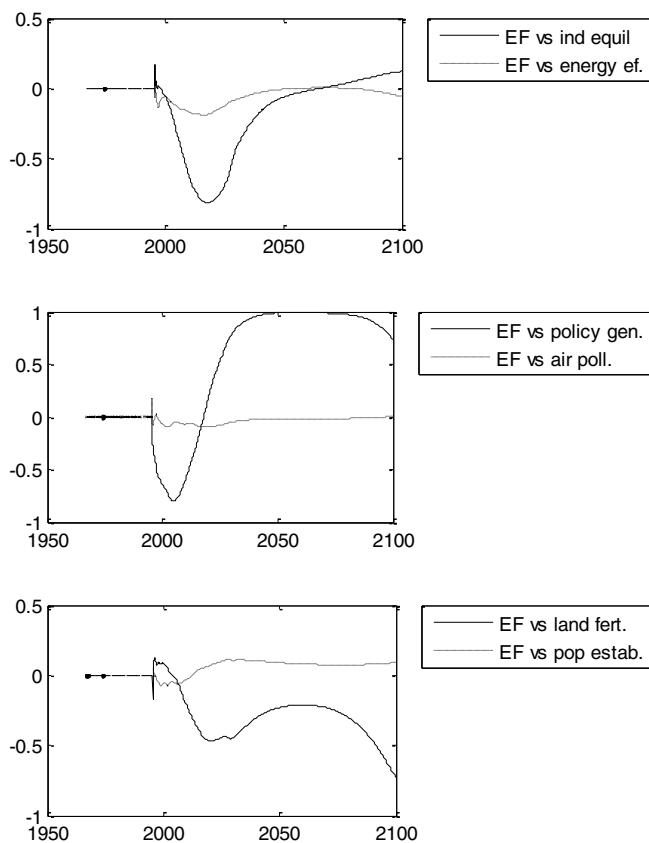


Fig. 3: Results of the screening process.

The results of Fig. 3 show that the policy best correlated to reduce the ecological footprint is the general policy ( $t_{policy\_year}$ ), but this policy is bad, or counter-productive, in the short term. Other favourable policies are the equilibrium of the industrial production and air pollution control, but both have bad effects in the short term, especially the industrial equilibrium policy. The population stability, the energy efficiency and the fertility of land all have adverse effects in the long run.

The results are, in some ways, the same as those for other techniques: the best correlated policy to reduce the ecological footprint is the general policy, which includes all the others (a global focus), but which, in the short term, increases the ecological footprint. Then, in the long run, the next best policies are those of industrial production equilibrium and that of air pollution, although both of these are counter-productive in the short term, giving negative values, in particular that of industrial equilibrium. Screening is a good technique for analyzing models. It gives more information about correlations than the simple inspection of output-variable plots, and it is remarkable how easy it is to program using Matlab tools, compared to the information it reveals.

Ford and Flynn (2007) describe the analysis of the World3 model with this screening technique, and find similar results to counter-productive policies in the short term. Unfortunately, these results cannot be compared with ours, since they use the previous version of the model, the World3-72 (Meadows *et al.*, 1992), which is a lot simpler than the

one we use, the World3-2004. Besides, they analyze the effects of two parameters: “fraction of industrial output allocated to consumption” and “industrial capital output ratio”, which are not directly manipulated in the scenarios of World3-2004, and it is difficult to know if equivalent parameters can be found.

#### 4. USE OF ARTIFICIAL INTELLIGENCE IN WORLD3

##### 4.1 Fuzzy logic in World3

We have used fuzzy logic to evaluate the response of the output values of some simulations of World3. The population curve is the output that we select. We would say that the curve is a bad result (catastrophe) if the final value of the population is low, this feature being the most significant. However, the result is also bad if the peak of the population curve occurs very soon or if the final slope of the population is steep, which means that the population tends to decrease even more.

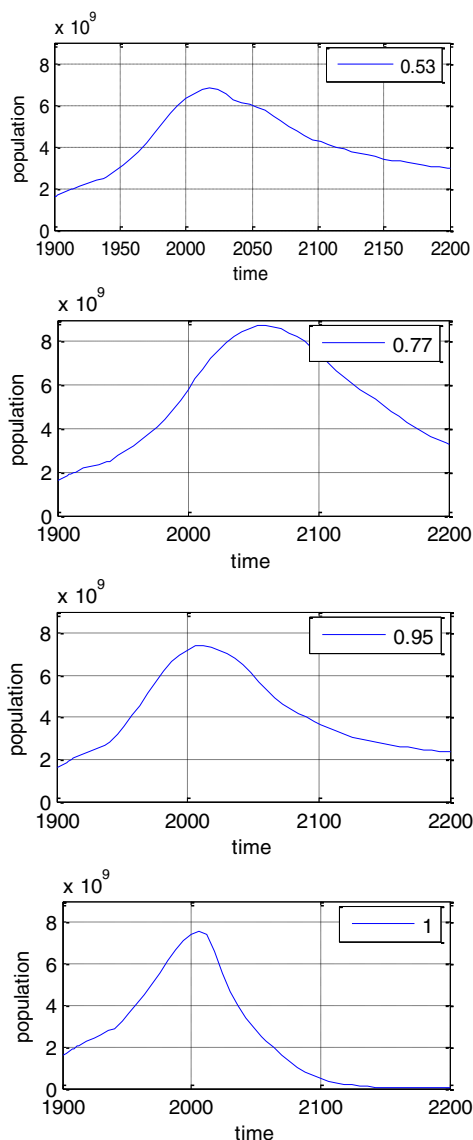


Fig. 4: Results of the use of fuzzy logic for the analysis of the population curve.

Therefore, the following values of the population curve are dealt with: the final value, the year when the maximum is reached and the final slope of the curve. These values would be assigned to the following fuzzy sets: “final population low”, “peak year of population low” and “final slope of population low”. Therefore, the fuzzy logic we use is as follows:

IF the final population is low  
OR 0.5 \* peak year of population is low  
OR 0.7 \* final slope of population is low  
THEN  $\rightarrow$  result is a collapse

In Fig. 4, we can see several population curves and the results of the application of this sentence. We can see that these results have different degrees of “badness”, which reflect how close they are to the characteristics expressed in the fuzzy sentence. The curve whose result of 0.53 is clearly the best result, the final population is not too low and it is more stable than in other results. The curve with the result 0.77 has a higher maximum value and a greater fall, the curve with 0.95 has a low final value and the curve with 1 is clearly the worst of all.

##### 4.2 Fuzzy clustering in World3

Clustering of numerical data forms the basis of many classification and system modeling algorithms. The purpose of clustering is to identify natural groupings of data from a large data set to produce a concise representation of a system's behavior. Fuzzy c-means (FCM) is a data clustering technique wherein each data point is characterized by a membership grade to each cluster. This technique is an improvement on earlier clustering methods. Fuzzy classification algorithms start with an initial guess for the cluster centers, which are intended to mark the mean location of each cluster. They also assume an initial number of clusters. The initial guess for these cluster centers is most likely incorrect. Additionally, the algorithm assigns every data point a membership grade for each cluster. By iteratively updating the cluster centers and the membership grades for each data point, it iteratively moves the cluster centers to the right location within a data set. This iteration is based on minimizing an objective function that represents the distance from any given data point to a cluster center weighted by that data point's membership grade. If we want to use fuzzy clustering to analyze the results of our simulations of World3, we must first choose those data that we want to group. The data we have are the random parameters of our simulations and the corresponding results (final value of population, ecological footprint at medium term, “catastrophe” fuzzy indicator, etc.). Those are many data, untraceable unless we manipulate them, as we are doing with clustering.

We might focus on the results of the simulations, for example: run a big number of simulations (say 200) with randomly chosen parameters, calculate some values of interesting results in all those simulations, and then run the clustering algorithm with those data to find the centers of three or four clusters. In this way, all the possible results of a big number of simulations are summarized in those three or four cluster centers. These centers give us the following

conclusions about the results of the simulations: *Firstly*, if the medium term ecological footprint is low, then the welfare index in the long term is high. If the ecological footprint is high, the long-term welfare is low. *Secondly*, at medium term, the fact that the footprint is high does not make a big difference, but in the long run it does. A high ecological footprint would decrease the long term human welfare. This result coincides with the spirit of the whole Limits to Growth study and fits the philosophy of the World3 model.

The real interest of the clustering technique is the inspection of data in high dimensional spaces, where a visual inspection of results is not possible. In this section, we present an experiment where the number of variables is 10: six parameters and four results; the value of the ecological footprint (*HEF*) in 2050 and in 2100 and the value of the human welfare index (*HWI*) in 2050 and 2100. The results of the *HWI* and *HEF* for 50 simulation runs show that, in groups 1 and 3, the result is quite positive: a high final welfare index and low ecological footprint, while those of clusters 2 and 4 are worse. The main difference between the parameters of those clusters is in the *t\_policy* (the general policy), which is high in the two clusters of bad results. In those clusters where this combined policy is applied before 2005-2010, the result is a high final human welfare.

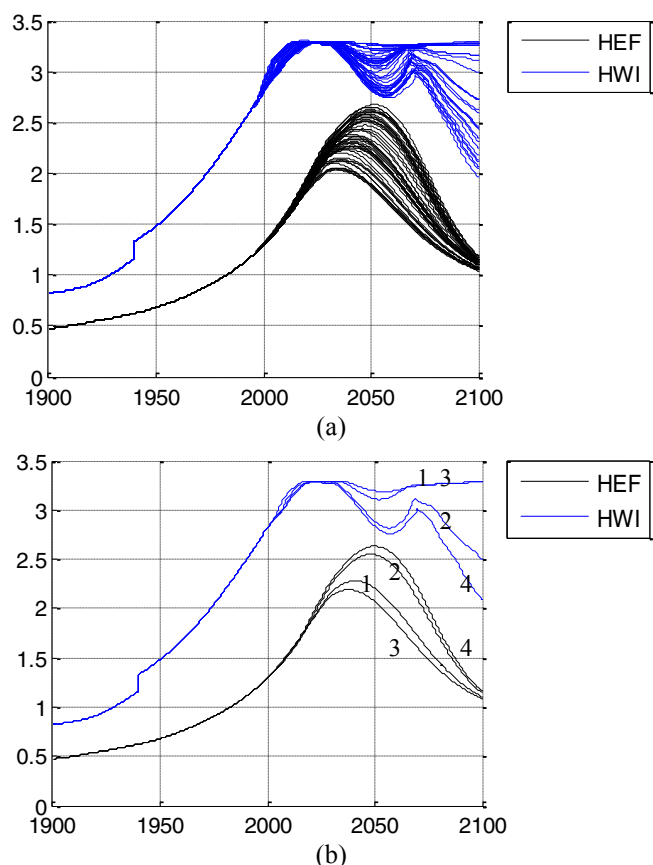


Fig. 5: Ten dimensional examples of data clustering.

We cannot see these ten dimensional clusters in a simple figure, but Fig. 5 shows the results of several simulations with randomly changing parameters (a) and the results of the simulations with the four centers found with the clustering

technique (b). In this figure, we can see that the four clusters represent quite well all the possible results that can be obtained in all the simulations.

## 5. CONCLUSIONS

This paper has explored the possibilities that a programming language (Matlab-Simulink) gives us for the programming and analysis of system dynamics models. It has also shown the possibilities that some tools taken from artificial intelligence offer for the analysis of these models. We have applied several techniques to the analysis of a known and very complex model, the World3: clustering, fuzzy logic, correlation analysis, etc. All of them enable the programmer to automate the inspection of the results and give more insight into the relationship between the changing parameters and the results.

All these tools make the analysis of complex models a lot more profound than simple sensitivity analysis and help the modeler to extract information from it. It would also be nice if the companies that create software for system dynamics programming could develop packages to interact with math programming environments such as Matlab.

## REFERENCES

- Acharya, S.R., and Saeed, K. (1996). An attempt to operationalize the recommendations of the 'limits to growth' study to sustain the future of mankind. *System Dynamics Review*, 12(4), 281–304.
- Ackoff, R.L. (2010). *Systems thinking for curious managers*. Triarchy Press Limited.
- Checkland, P. (1981) *Systems Thinking, Systems Practice*. John Wiley & Sons, Ltd. England.
- Ford, A., and Flynn, H. (2004). Statistical screening of system dynamics models. *System Dynamics Review*, 21(4), 273–303.
- Forrester, J.W. (1961). *Industrial dynamics*. The M.I.T. Press. Cambridge, MA.
- Güneralp, B. (2006). Towards coherent loop dominance analysis: progress in eigenvalue elasticity analysis. *System Dynamics Review*, 22(3), 263–289.
- Kampmann, C. E., and Oliva, R. (2006). Loop eigenvalue elasticity analysis: three case studies. *System Dynamics Review*, 22(2), 141–162.
- Meadows, D.H., Meadows, D.L., Randers, J., and Berhens, W.W. (1972). *The Limits to Growth*. New York: Universe Books. [Republished by Productivity Press, Portland, OR].
- Meadows, D.H., Meadows, D.L., and Randers, J. (1992). *Beyond the Limits*. Post Mills, Vermont: Chelsea Green Publishing Company. USA.
- Meadows D.H. Meadows, Randers J. and Meadows, D.L. (2004). *Learning Environment: Limits to Growth-The 30 year Update*. Sustainability Institute.
- Mojtahedzadeh M., Andersen D. and Richardson GP. (2004). Using *Digest* to implement the pathway participation method for detecting influential system structure. *System Dynamics Review*, 20(1), 1–20.