A QUANTITATIVE PERFORMANCE MEASURE AND ITS APPLICATIONS TO SYSTEM DYNAMICS

Raj Kumar Bapna
International Software India Limited
SDF Block 1, Units 5 & 6, M.E.P.Z.
Madras-600 045 India

Sushil K. Sharma
Department of Mechanical Engineering
Institute of Technology, B.H.U.
Varanasi-221 005 India

ABSTRACT

This paper presents a general quantitative performance measure to analyze the performance of System Dynamics models for the desired model behavior. The advantages of using a quantitative performance analysis facility are discussed. Some possible applications including optimization and sensitivity analysis are discussed. Also discussed are the implementation aspects.

1. INTRODUCTION

System Dynamics (SD) and DYNAMO (Pugh 1976) were developed over 25 years ago. In spite of the vast scope for further developments, very limited research has been done to improve the SD simulation languages as well as applications like optimization (Bapna 1985, Bapna, Ghose, Sharma 1987a and 1987b, Ghose et al. 1987b, Keloharju 1987, Rohrbaugh 1983, Wolstenholme 1985a and 1985b)

At present, no techniques or tools are available which allow the SD practitioner to quantitatively measure the model performance for the desired model behavior. In the current situation, it is necessary to study graphical or tabular outputs to determine how good the model performance is.

The ability to quantitatively analyze the performance of a SD model would permit, among others, development of Optimization and Sensitivity Analysis facilities.

To calculate the quantitative performance measure, the user is required to mathematically express the desired model behavior, and this will help the user understand the desired system behavior more precisely.

2. QUANTITATIVE PERFORMANCE MEASURE (QPM)

The desired behavior of an SD model can be represented as the values of various system parameters (and variables) over the simulation time period. The QPM is based on the use of error functions to compute the error between the desired model behavior and the simulated model behavior. When the desired model behavior and the simulated model behavior are identical, the value of QPM is zero.

The QPM is defined as follows.

$$QPM = \sum_{i=1}^{n} w_i * f_i \ (p_i , p_{i0})$$

where

p_i - the simulated value of the i th parameter over the simulation period, as function of time.

p_{i0} - the desired value of the i th parameter over the simulation period, as function of time.

f_i - a non linear function representing the integration, over the simulation period, of some function expressing difference between p_i and p_{i0} as below.

$$f_i (p_i , p_{i0}) = \sum e_i (p_i , p_{i0}) * DT$$

(Summation over simulation period at DT intervals)

Here e_i is error function. Appendix shows some of the possible functions for e_i

w_i - weightage associated with $f_i (p_i , p_{i0})$.

n - total number of parameters used in QPM.

To compute QPM once, the SD model has to be simulated over the desired period of time. To calculate QPM, all the system parameters or only some important parameters may be used.

The concept of quantitative performance analysis is illustrated in Fig. 1. The diagram shows the desired inventory changing from i_{10} to i_{20} as a result of consumption changing from c_1 to c_2 at the time T1. The simulated inventory is shown superimposed over the desired inventory. The shaded area shows the deviation of the simulated inventory from . the desired inventory. The deviation or error can be represented by QPM as below.

$$QPM = f(i_2 , i_{20})$$

where $f(i_2 , i_{20}) = \sum e(i_2 , i_{20}) * DT$

(Summation over simulation period at DT intervals)

where the function e is a non linear function of the desired inventory (i_{20}) and the simulated inventory (i_2). DT is the time increment for simulation. The function e can be chosen to be one of the functions described in the Appendix, say $(i_2 - i_{20})^2$.

On the basis of the above description, QPM gives a figure of merit for the model performance. Once the QPM is defined for an SD model, it is easy to simulate the model for different policies.

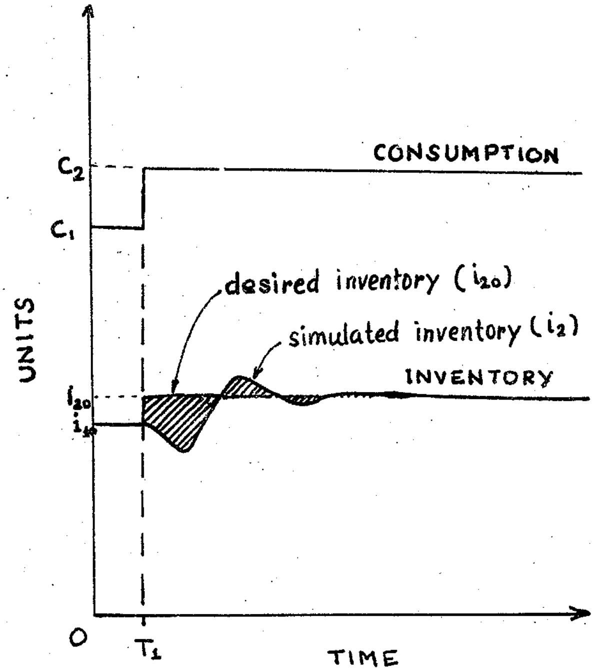

Fig. 1 Illustration of Quantitative Performance Measure in System Dynamics

(or parameters) and achieve optimization. Similarly it is easy to do sensitivity analysis. Both optimization and sensitivity analysis are described in detail in the next section.

The QPM presented here is general enough to allow the user to express any desired model behavior of an SD model. The QPM and its applications can be implemented easily in languages like DYMOSIM (Mohapatra and Bora 1983) or any general purpose programming languages like FORTRAN, BASIC. Software can be developed for DYNAMO (Pugh 1976), DYSMAP (Coyle 1977) to offer facilities like QPM, optimization, sensitivity analysis etc. in a very easy-to-use way.

3. APPLICATIONS

The facility to measure the model performance quantitatively can be used to perform sensitivity analysis and optimization as described below.

3.1 Sensitivity Analysis

Sensitivity analysis is the study of the variation in model performance with the variation in the model parameters (or variables).

One possible method of its implementation is as follows. The sensitivity of an SD model to certain variable X can be represented in several ways, four of which are described below.

(1) $S = [QPM(X+\Delta X) - QPM(X)] / [\Delta X]$

(2) $S = [QPM(X+\Delta X) - QPM(X)] / [\Delta X/X]$

(3) $S = [QPM(X+\Delta X) - QPM(X)] / [QPM(X) * \Delta X]$

(4) $S = [QPM(X+\Delta X) - QPM(X)] / [QPM(X) * \Delta X/X]$

The sensitivity (S) in item (4) represents the ratio of percent variation in model performance to the peroent variation in the model parameter.

3.2 Optimization

Three possible types of optimization for SD studies are described below.

Optimization Type 1: Here the user is required to specify the names and ranges of parameters. The number of parameters is assumed to be n, thus the n-dimensional space is searched for optimization. Two simple implementation methods are described below. In the first one the data points are selected at random, and in the second one the data points are selected in a regular fashion.

(A) Here, the non linear programming technique is used. First, a set of n random numbers are generated. Each random number determines the value of a corresponding parameter. In this manner, a set of parameters for one simulation run are generated. This randomization process is done several times giving different simulation runs, and thus sub-optimal solution is found along with the associated parameter values.

(B) In this implementation, all the parameter ranges are divided into several equidistant discrete values, and thus a definite number of equidistant points in the n-dimensional space are generated. QPM is determined for each of them and thus the sub-optimal solution is found along with the associated parameter values.

In most of the cases, as the model performance varies smoothly over a parameter range (Keloharju 1987) for SD problems , the method (B) is likely to give a solution quite close to the optimal solution.

The optimization method (A) or (B) can be used to optimize parameters locally (over DT or multiples of DT period). This local optimization can be done thorough-out the simulation time, thus generating dynamically changing policies (parameters).

Optimization Type 2: Here the user is required to specify the alternate policies (equations or parameters). The QPM for each policy option is computed and thus optimal solution is found.

Optimization Type 3: In this optimization, a model base is used. The model base may contain several SD models for the same system. These models may differ in one or more modules. But the parameters present in the QPM should be common to each of the alternative models. The QPM for each SD model is computed and thus the optimal model is chosen.

As a variation, some of the best solutions obtained by the above types of optimization can be presented to the user with the final choice left to him or her.

The above optimizations require QPM to be determined for each alternative choice. The best choice at any time can be remembered by the program, so that a simulation run may be stopped as soon as the performance is found to be not as good as current best. Thus, not all of the simulation runs will have to cover the full length of simulation time period.

4. CONCLUSIONS AND FURTHER RESEARCH

This paper presented a quantitative performance measure. This measure makes possible quantitative analysis of the SD models quantitatively. The applications of the measure to optimization and sensitivity analysis of the SD models were described. Implementation aspects of optimization and sensitivity analysis were also discussed.

There is a vast scope for further research on quantitative performance analysis and its applications to SD models. The QPM as presented in this paper serves only as a beginning. The following are the possible directions for further research.

(1) Various error functions for QPM in the context of SD problems can be studied and suitable ones identified.

(2) The QPM may consider factors like frequency, raise time, settling time etc to simplify the QPM function description for certain specific SD applications.

(3) Optimization and sensitivity analysis, using QPM method, can be carried out for real life problems as well as text book problems. Such experience with QPM would help improve the QPM methodology.

(4) Different sensitivity analysis formulae including those presented in this paper can be studied, and suitable ones identified for use with the SD models.

(5) Algorithmic control modules (Wolstenholme 1985b) can be analyzed for sensitivity to parameter variation. Such information can be of use while developing an SD model in modular fashion (Bapna, Ghose, Sharma 1987b).

(6) The techniques of Artificial Intelligence and Decision Support Systems , with the usage of the QPM, can be used to result significant. developments in SD methodology, techniques and tools.

ACKNOWLEDGMENT

The authors thank Abburi Rajasekhar for his help in discussion and presentation of ideas, and also in preparation of the paper.

REFERENCES

Bapna, R. K. (1985) Design and Implementation of a Simulation Language for System Dynamics, Master's Thesis, Indian Institute of Technology, Kharagpur, India.

Bapna, R. K., Ghose, S., Sharma, S. K. (1987a) "Development of a System Dynamics Simulation Language", Proc. of the Second National Conference on System Dynamics, India.

Bapna, R. K., Ghose, S., Sharma, S. K. (1987b) "Beyond DYNAMO: Future System Dynamics Simulation Programming", Proc. of the 1987 International Conference of the System Dynamics Society, China.

Birta, L. G. (1977) "A Parameter Optimization Module for CSSL-Based Simulation Software", Simulation, Vol 28 No 4.

Coyle, R. G. (1977) Management System Dynamics, John-Wiley.

Ghose, S., Chakrabarti, P. P., Bapna, R. K. (1987a) "Intelligent Decision Support Systems Using System Dynamics Models", Proc. of the Second National Conference on System Dynamics, India.

Ghose, S. et al. (1987b) "Qualitative Optimization - An AI Approach to Policy Design in System Dynamics", Proc of the Second National Conference on System Dynamics, India.

Keloharju, R. (1987) "Use of Optimization in SD", Orientation Course Lecture Notes of the Second National Conference on System Dynamics, India.

Mohapatra, P. K. J. and Bora M. C. (1983) DYMOSIM User's Manual, Indian Institute of Technology, Kharagpur, India.

Pugh, A. L. (1976) DYNAMO III User's Manual, MIT Press.

Rohrbaugh, J. and Andersen, D. F (1983) "Specifying Dynamic Objective Functions: Problems and Possibilities", Dynamica, 9.

Wolstenholme, E. F. (1985a) "Algorithmic Control Modules for System Dynamics Models", Proc. of the 1985 International Conference of the System Dynamics Society.

Wolstenholme, E. F. (1985b) "A Methodology for Qualitative System Dynamics", Proc. of the 1985 International Conference of the System Dynamics Society.

APPENDIX - ERROR FUNCTIONS FOR QPM

As described in Section 2, the QPM is represented as

$$\text{QPM} = \sum_{i=1}^{n} w_i * f_i(i,io)$$

$$\text{where } f_i(p_i, p_{io}) = \sum e_i(p_i, p_{io}) * DT$$

(Summation over simulation period at DT intervals)

Some of the error functions $e_i(i,io)$ that can be used for QPM and their implications are described below.

(1) Difference (i-io)

This error function gives equal importance to deviation on either side of the desired value, but errors with opposite sign cancel each other. It also gives equal importance to deviation of small magnitudes as well as large magnitudes.

(2) Square of difference ($[i-io]^2$)

This error function adds up errors of both the positive and the negative signs. The errors of large magnitudes are magnified. This error function is likely to be useful in most cases.

(3) Negative difference only ($1/2\ ([i-io] - |[i-io]|)$)

This error function penalizes deviations only of negative sign. This may be useful in modeling parameters such as actual or notational profit.

(4) Positive difference only ($1/2\ ([i-io] + |[i-io]|)$)

This error function penalizes deviations only of positive sign. This may be useful in modeling parameters such as actual or notational loss.

(5) Even power of difference $(i-io)^n$

This is similar to the error function in item (2), but amplifies the large errors to a greater extent than the function in item (2). May be useful in modeling those parameters such as risk or parameters whose large value will have catastrophic effect on model performance.