SENSITIVITY ANALYSIS METHODS FOR SYSTEM DYNAMICS MODELS

by

J.A. Sharp

Lecturer

System Dynamics Research Group

University of Bradford

BD7 1DP

England

ABSTRACT

System Dynamics (SD) may be viewed as a process of designing ROBUST systems. The concept of ROBUSTNESS leads to a need for analyzing the effects on SD models of both parameter changes and stochastic inputs. It is demonstrated that the effects of large parameter changes can be measured by the use of hill climbing techniques given efficient computation. The paper describes the traditional ways of assessing sensitivities in SD models, together with methods based on perturbation techniques which unify the parameter and stochastic sensitivity problems. The computational characteristics of the various methods are analysed and the factors that affect their computational efficiency are discussed.

The paper discusses the results of experiments to determine the accuracy and speed of the various methods on a 7 state variable, 16 parameter model and on a 70 state variable, 160 parameter model derived from it. The perturbation methods yield acceptable accuracy and for the models described reduce computer time by a factor of between 9 and 25. Compiler changes discussed in the paper would make sensitivity analysis easier and quicker and would improve techniques elsewhere in System Dynamics.

Contents

I. THE NEED FOR SENSITIVITY ANALYSIS

The philosophy of system design by System Dynamics (SD) methods can be interpreted as the use of feedback control methods to attain ROBUST system performance, (Sharp, 1975), (Coyle, 1975). The concept of ROBUSTNESS has at least 3 elements:

a) that the system show a satisfactory response when subjected to a wide variety of inputs. Thus the production planning system of a company manufacturing for stock would be expected to maintain adequate stocks and change production in a way that was compatible with company policy for redundancy and recruitment when subjected to seasonality or a business cycle in orders for its products. In this paper we are not concerned with this aspect, except indirectly through its bearing on the choice of objective functions.

b) that the system perform satisfactorily over the range of parameter values considered plausible; since however system parameters (Constant and Initial Values)are estimated they will usually be subject to error and similar errors may arise through aggregation, (Sharp, 1974).

c) that the system be relatively unaffected by the fairly considerable amounts of noise usually found in socioeconomic systems as a result of structural error, input noise and to errors in measuring variables on which system policies are based, (Sharp, 1976a).

In the analysis and redesign of systems using SD methods sensitivity analysis should play a large part. Furthermore sensitivity analysis has two separate, though from certain viewpoints interlinked, facets, which this paper will examine, viz Parameter Sensitivity and Stochastic Sensitivity.

To assess whether he has been successful in designing a ROBUST system the modeller needs to measure the impact of parameter (Constant and Initial Value) changes and the effects of noise on the system. In most studies certain model constants on initial values may have only been estimated rather crudely because of lack of data or because the modeller has decided that precise estimation of them is unnecessary since the behaviour of the system is expected to be insensitive to changes in them. The paper by Graham in this volume illustrates the practicalities of parameter estimation and in particular how in most cases a Constant such as Average Housing Life can only be fixed within a certain range.

In addition certain parameters that appear in the model as Constants may be slowly changing over time. Again other parameters such as smoothing times are capable of being changed by managerial decision and it is desirable to know the impact on system behaviour of changes to them.

In practice both modeller and client are usually well aware of the imprecision in many parameter estimates. This leads naturally to the question: Is the policy that appears best as a result of the model experiments really the best for all plausible parameter values or is it possible to find equally acceptable parameter values for which another policy is better? To give a satisfactory answer to this question, the modeller needs to examine the sensitivity of system behaviour to these uncertain parameters.

Since in many situations for one or other of the reasons just given there are few parameters for which we do not wish to assess sensitivity it is natural to search for methods of parameter sensitivity analysis that can deal with variations in all system parameters.

A similar problem arises with regard to the effects of noise on a system. Most experienced modellers will generally examine the effects of noise in the driving input (very often Sales Rate) on the behaviour of the system. There are, however, many other sources of noise in the system.

Any model equation will, in practice, only contain the variables that are considered of major importance in determining the behaviour it describes. There will of course be many other factors that, from time to time, affect this behaviour. Rather than ignore them completely, it is better to follow econometric tradition and approximate their effects by adding a random noise term (structural error) to the equation. The results, when this is done, are often quite dramatic. Figure 1 represents the performance of a simple production planning system operating deterministically without noise terms in the equations. Figure 2 shows the behaviour of the same system when noise terms are added to the equations. Not only is there little resemblance between the two systems but in some respects, e.g. Production Completion Rate, that of Figure 2, is unsatisfactory.

The modeller should therefore also attempt to satisfy himself that the policies that appear best with deterministic experiments will also perform satisfactorily under the noisy conditions likely to prevail in the real system. In other words he must measure the sensitivity of the system to noise.

Strikingly little attention has been given to Sensitivity Analysis in the SD literature since the publication of 'Industrial Dynamics' in 1961. Most published studies have been concerned with specific aspects of particular models, particularly the World 2 model. There have been relatively few attempts to consider the general problems of sensitivity analysis.

On the whole the sensitivity analysis aspect in most SD studies appears somewhat unsatisfactory which suggests that even if they recognise its importance, most analysts find sensitivity analysis a chore.

It might be argued that the classical approach as outlined by Jay Forrester in 'Industrial Dynamics' is not sufficiently systematic to determine the effects of uncertainties about parameters or the effects of noise on a system. The method adopted there and subsequently followed by most other workers is essentially the analysis of the sensitivity of the system to certain changes which their experience suggests might have considerable effects. Such a procedure would seem to have at least two drawbacks: firstly, even experienced analysts may wrongly assume that the system is insensitive to changes that in fact have a large effect; and secondly, it offers little help to the novice in developing the system understanding that might enable him to apply it successfully. This paper will therefore be concerned with exhaustive methods of sensitivity analysis that permit the effects of changes in all parameters and all stochastic effects to be measured.

Sensitivity analysis methods should satisfy a number of criteria, if they are to be useful in practice. Firstly, they should be systematic, that is they should be capable of assessing the effects of all uncertainties on the system. They should be simple for the modeller to use so that sensitivity testing is not skipped because of the effort required. They should also be computationally efficient, since tests of large models they may require substantial amounts of computer time.

Interestingly enough however, the ROBUSTNESS concept suggests that very accurate estimates of the effect of a change are unnecessary, as long as the order of magnitude of it is correctly estimated. If the effect is large, the system will require redesign to reduce it, whilst if it is small quite large percentage errors in estimating it are unimportant. This suggests that there may be considerable scope for sensitivity testing methods that give a reasonable approximation to the correct result and reduce the amount of computing required.

II PARAMETER SENSITIVITY ANALYSIS

Behaviour and Objective Functions

The behaviour of an SD model depends on its structure and its parameters. (Constants and Initial Values). The previous section discussed the reasons why the modeller must study the effects of parameter changes on system behaviour. The behaviour of interest to the modeller will vary depending on the situation. It will however be assumed that the behaviour is quantifiable, that is it can be defined by numerical measures or objective functions. The reason for this assumption is purely pragmatic; Sensitivity analysis yields a great deal of information. For example, a full parameter sensitivity analysis of the small model described later in this paper produced 130 pages of PLOTS and PRINTS. Such volumes are wasteful of computer time and little help to the analyst. In practice, the analyst is forced to select a few key variables which his experience tells him are the most important indicators of system behaviour and examine the effects of parameter changes on the behaviour of these variables. Thus in the World 2 model Jay Forrester was clearly primarily concerned with the behaviour of population - in particular the avoidance of catastrophic decline - and natural resource consumption. Gil Low in his paper in this volume is

concerned with the presence or absence of cycles in GNP.

In essence the modeller must construct objective function(s) for the system, which, in principle, can be formulated explicitly. Assuming that this has been carried out, the analyst has at his disposal a relatively small number of measures of system performance to assess sensitivity.

In the World 2 model Jay Forrester formulated an explicit objective function, which he called Quality of Life. In addition he also considered implicitly the value of other variables such as Natural Resources remaining in the year 2100. De Jong and Dereksen, (1975) have given explicit objective functions, of the type engisaged in this paper, for the World 2 model. Barnett (1973) has shown how objective functions can be set up to reflect the financial, engineering and political considerations involved in determining how best to exploit a new oilfield.

Exploring the effect of changes in model structure by parameter changes.

Though DYNAMO makes a clear distinction between parameters (C and N variables) and structure (L, A and R equations) this distinction is, mathematically, artificial. This fact can usefully be exploited to enable the effects of changes in uncertain parts of the model structure.

For example, by regarding the equation:

```
A   X.K = Y.K/C
```

as a particular case of the equation

```
A   X.K = Y.K/C + D*Y.K**2
```

with D = 0 the analyst can explore the effects of changing model structure by introducing a quadratic term into the expression for X.K, by determining the effect of changing the value of D to a non-zero value.

The control engineering literature that deals with sensitivity analysis distinguishes two types of sensitivity analysis: Local Sensitivity Analysis and Global Sensitivity Analysis.

Local Sensitivity Analysis is concerned with the way in which system behaviour is changed by small changes in system parameters. The changes should strictly be infinitesimally small for accurate estimates of sensitivity (Sharp, 1974). In practice, however, it is normal to make 1% changes and observe how much system behaviour is changed.

For insensitive systems that show small changes (significantly less than 1%) in the behaviour of interest such an analysis usually gives accurate answers for much bigger changes in parameters. Local parameter sensitivity analysis has three uses:

a) Since it is straightforward to carry out and as has already been shown most model parameters are uncertain, it provides a basis for assessing how model performance would be affected by parameter changes. It thus represents a desirable standard of SD practice that the results of such an analysis be published as evidence that the question of parameter sensitivity has been considered and as a guide for those who wish to evaluate the results of the study as to ways in which model behaviour might be changed.

b) In the mathematical study of sensitivity analysis Local Parameter Sensitivity is a natural starting point because the restriction that changes in parameters be very small greatly simplifies the analysis. With this restriction it is possible to show that the effects of parameter changes is additive, i.e. the effect of say a 0.1% change in parameters A and B simultaneously is simply the sum of the effects of a 0.1% change in each on its own. This simplification makes possible the development of new and potentially powerful methods of sensitivity analysis, (Sharp, 1974).

Global Sensitivity Analysis is concerned with the question of how system behaviour is affected by finite changes in parameters. Since most parameters can only be fixed within a certain range the model sensitivity analysis should ideally be global in nature. Thus the value of any particular parameter (p) will only be fixed within a certain range, i.e. lower limit $\leq$ p $\leq$ upper limit. In practice the limits are derived on the basis of the modeller's knowledge of the system. The range of values capable of being taken is however usually quite large - a range equal to the chosen parameter value being not uncommon. Since these uncertainties exist for most model parameters the Global Sensitivity problem is to find how system behaviour is affected by simultaneous changes to all uncertain parameters within their range of uncertainty.

This problem is far more complex than that of local sensitivity analysis. Whereas in the latter case the effects of changes are additive, in global sensitivity analysis synergistic effects arise, so that the effect of finite changes to several parameters simultaneously is not merely the sum of the effects of individual changes. The mathematical theory of global sensitivity analysis is difficult and incomplete so no general methods exist for the determination of global sensitivity. Nevertheless it is possible to develop practical methods that enable global sensitivity to be examined and these will be discussed later.

Computation of Local Parameter Sensitivities by the Conventional Method

The Conventional Method of determining parameter sensitivities is that given by Jay Forrester (1961). The model parameters are changed one at a time from their initial values, each parameter change generating a run. Though it is usual to change the parameters by a finite amount this approach is essentially a local sensitivity method.

The method is simple to use but tedious to apply in practice, since each parameter change requires the generation of a C or N card depending on whether we are changing a Constant or an Initial Value and a RUN card. In large models the total number of parameters is normally about the same as the total number of variables so this procedure may require the modeller to set up several runs. For this reason few modellers seem to carry out sensitivity testing of all model parameters. Thus existing DYNAMO compilers do not encourage systematic sensitivity testing.

It can be shown (Sharp, 1974) that if v is the number of L,A and R variables in the SD model then the time taken to carry out a full sensitivity analysis is given by $K(1)\ v^2/DT$ where DT is the length of the simulation time step and K(1) is a constant that depends on the computer being used. The quadratic nature of this formula implies that the time to carry out a full sensitivity analysis depends on the square of the number of model variables and hence increases rapidly with model size.

If good programming practice is followed the actual computations in DYNAMO models are very efficient because complex expressions are normally defined as auxiliaries. It follows that little reduction in the time for sensitivity analysis can be gained by reformulating the model itself. For many business systems however the effects of changes in initial values

quickly die out (Sharp, 1974), 1976b) and therefore the time required for sensitivity analysis by the conventional method can be reduced by merely carrying it out only for model constants that appear in L,A and R equations. Even if the scope of parameter sensitivity testing is reduced in this way however the effect is merely to reduce the factor K(1) by 20-30% and for models such as World 2 that may be sensitive to certain initial values such a simplification of parameter testing is undesirable.

Statistical Methods of Local Parameter Sensitivity Analysis

Another approach to local sensitivity testing is the use of a statistical approach, i.e. the modeller makes random changes to the parameter and initial condition vectors, carries out model runs and measures their effects. Since locally the effects of parameter changes are additive the effects of individual parameter changes can be estimated by either regression or correlation methods. However that such an approach requires more than p runs for p parameters to give a reasonable estimate of the sensitivities. Statistical approaches are, consequently inferior to the Conventional method and will not be considered further.

Local Parameter Sensitivity Analysis By Perturbation Methods

The control engineering approach to local parameter sensitivity analysis, (Tomovićand Vukobratović, 1972) is very different to the Conventional SD method. Since the mathematics are complex only a brief outline of the method will be given here. Full details are given in Sharp, (1976b) and Sharp (1974).

To illustrate the method the following SD model will be considered:

$$\text{L} \quad X.K = X.J - DT*C*X.J \qquad (1)$$

$$\text{N} \quad X = 1 \qquad (2)$$

This may be rewritten mathematically in accordance with the way the DYNAMO compiler actually simulates as:

$$X(N+1) = (1-DT*C)*X(N) \qquad (3)$$

$$X(0) = 1 \qquad (4)$$

where X(N) denotes the value of X at time N*DT.

Suppose that the value of C were changed by a small amount δC to $C+\delta C$ and the initial value is changed by a small amount δI. Corresponding to the equations (3) and (4) new equations can be derived for the variable Y(N) corresponding to X(N) in the original simulation

$$Y(N+1) = Y(N)*(1-DT*(C+\delta C))*Y(N) \qquad (5)$$

$$Y(0) = 1+\delta I \qquad (6)$$

The obvious measurement of parameter sensitivity for this system is simply the difference between corresponding values of X and Y, that is the value of $D(N) = Y(N)-X(N)$

Subtraction of equation (3) from equation (5) gives

$$\begin{aligned} D(N+1) &= (Y(N)-X(N)) * (1-DT*C) + C*DT*Y(N) \\ &= D(N)*(1-DT*C)+ C*DT* X(N)+D(N) \\ &= D(N)*(1-DT*C)+ C*DT*X(N)+ C*DT*D(N) \qquad (7) \end{aligned}$$

If C is small it is easy to see that $C*DT*D(N)$ is small. As an approximation the last term in equation (7) is therefore dropped to give:

$$D(N)+1) = (1-DT*C)*D(N)+ C*DT*X(N) \qquad (8)$$

From equations (4) and (6) the initial conditions for this equation are

$$D(0) = \delta I \qquad (9)$$

Since the values of X(N) are known from the original model runs equations (8) and (9) can now be solved to give the value of D(N). Writing for notational convenience $(1-C*DT) = a$, $CxDT = b$,

$$\begin{aligned} D(N+1) = b.\ X(N) + aX(N-1)+a^2X(N-2) \ .....a^N X(0) \\ + a^{N+1}D(0) \qquad (10) \end{aligned}$$

Note that in computing the effect of the change in the constant C, the effect of the change in initial value δI is obtained as a by-product. Note also that the approximations are strictly valid only for very small changes δC and δI.

The procedure just outlined can be generalized to the case of a model with many variables. The general case, however, involves the use of partial differentiation and matrices in order to obtain the formulae for the sensitivities(Sharp, 1976b). This method is therefore by no means as easy to understand and in fact has the disadvantage that, before it can be applied to any particular model, the equations of the model must be partially differentiated. This requires mathematical expertise and is a tedious and error-prone process. To counter balance these substantial disadvantages the perturbation approach offers the following advantages:

a) it provides a theoretical basis for the Conventional Method of assessing local parameter sensitivities

b) the method is applicable to many other SD problems. The papers by Peterson and Thissen in this volume, for example, discuss methods which involve the use of this same perturbation approach

c) the approach is easily adapted to assessing the effects of noise on the system with very little additional computational effort.

d) the approach lends itself to a variety of computational tricks for the reduction of computation time that cannot be applied to the Conventional Method (Sharp, 1976b). It can be shown that the computational time required for a model with v variables and time step DT is given by K(2) v^2/DT. Though the dependence on model size is quadratic as with the Conventional Method the constant K(2) can be considerably smaller than the constant K(1) associated with the Conventional Method. Therefore, as will be shown later in this paper the perturbation approach offers the possibility of substantial reductions in computing time that are highly desirable.

Computational Methods for Global Sensitivity Analysis

Though no general Global Sensitivity Analysis method exists it is possible to carry out global sensitivity analysis by using local sensitivity analysis in conjunction with a hill-climbing or minimization program. Such a program will minimize the value of a nonlinear function of several or many variables. Standard programs for this purpose are available at most installations, (ICI, 1969).

These methods either use local sensitivity analysis methods to guide the direction in which variables should be changed to reduce the value of the function being minimized and are suitable for use with either method of local sensitivity analysis or operate in an equivalent way by observing

the effects of parameter changes on the value of the function and using this information to guide the minimization process, which in effect carries out directly the conventional method of local parameter sensitivity analysis. In effect, they use information about the local sensitivity of the function in a systematic way to seek out the global minimum.

The way in which they can be used is best illustrated by an example. Assume, for simplicity, that the model behaviour of interest is Cumulative profit over the run. Assume that two policies (P1 and P2) have been considered and that policy P1 gives a higher profit(PR1)than the profit (PR2) attained with policy P2. Assume further for simplicity, that the model depends only on one parameter (C) whose value lies in the range $C1 \leqslant C \leqslant C2$.

The question then arises as to whether for any plausible value of C the profit generated by Policy 2 would be higher than that generated by Policy 1. If it cannot then obviously Policy 1 is to be preferred to Policy 2 for any plausible value of C - the preferred policy is insensitive to variations in C.

To set this up as a global sensitivity problem it is necessary to construct a double model that can generate the profits under both policies simultaneously, that is supply the values PR1 and PR2. This is a straightforward matter. This model is now used to supply the values for the computation of the function to be minimized by the minimization program by varying the value of C.

This function is defined by the equations

$$
\begin{aligned}
F &= PR1 - PR2 + Penalty \\
PENALTY &= 1000000*(C-C2) \qquad \text{if} \quad C \geq C2 \\
&= 1000000*(C1-C) \qquad \text{if} \quad C \leq C1 \\
&= 0 \qquad\qquad\qquad\quad\ \ \text{if} \quad C1 < C < C2
\end{aligned}
$$

The purpose of PENALTY is to ensure that the minimum will be found inside the range of plausible C values. The choice of the weighting factor - here 1000000 - depends on the likely value of (PR1-PR2) since it should give value of PENALTY larger than this difference. With this function F the minimization routine will find the minimum value of the difference (PR1-PR2) for values of C in the range $C1 \leq C \leq C2$. If the minimum value that is found turns out to be negative then for the corresponding value of C we have that $PR1 < PR2$, in other words Policy 1 is not superior for all values of C. If the minimum value of F is positive, on the other hand, we can only conclude that the minimization has not found a value of PR1 less than PR2. Because such a procedure cannot be guaranteed to attain the global minimum of the function the modeller cannot be absolutely certain that a C value cannot be found for which Policy 2 is superior to Policy 1. Nevertheless the use of a minimization procedure of the type outlined represents a much more systematic and thorough attempt to determine global sensitivities than the modeller can usually undertake by a series of ad hoc parameter changes.

Provided a suitable compiler is available that enables the DYNAMO program to be used together with say a FORTRAN program - for example the DYSMAP language (Ratnatunga, 1975) used at the University of Bradford the use of a minimization program in

the way outlined is quite straightforward and requires little mathematical expertise. Since the minimization process is likely, however, to require the equivalent of several hundred runs it is necessary to ensure efficient computation if large models are to be tackled. The minimization process is best broken off from time to time so that the modeller can guide the progress towards the minimum. The time required of the analyst is however much less than where the more usual strategy of ad hoc changes on the basis of experience is used to explore global sensitivity.

General Computational Considerations in Parameter Sensitivity Analysis

If the computer time required for parameter sensitivity analysis is to be minimized efforts should be made to make the process computationally efficient no matter which method of sensitivity analysis is adopted.

In part the analyst must formulate the model in a form that will reduce the computing required. Doing so is a matter of mundane programming skills, such as not computing a complex expression twice. It is also worth paying attention to variables defined through TABLE functions, since these functions merely consist of an array of parameters, exhaustive sensitivity analysis accordingly would require the variation of all TABLE parameters used in the simulation. Such a procedure would, in general, be far more time consuming than replacing the TABLE with a mathematical function and testing the parameters of this function.

III SENSITIVITY OF A SYSTEM TO NOISE

In Section I reasons were given for studying the effects of noise on systems. Few published models however systematically examine these effects. Most studies, following Jay Forrester (1961), consider the effect of noise in the driving input but few consider the effect of noise terms representing Structural Error or imperfections in the way individual equations describe reality. This is unfortunate since the effects of such noise can, as discussed in Section 1, be dramatic.

Sharp (1974) has argued that under ideal conditions SD and Econometric methods would represent different ways of obtaining the same model. One of the ways in which SD differs most from Econometrics at present, however, is the lack of attention paid to structural error. This is a natural consequence of the way SD models are built, certain difficulties associated with modelling noise that will be discussed later, and of the fact that structural error terms appear explicitly in Econometric models.

In this section we shall be concerned with the effects of random noise with zero mean and no serial correlation (correlation between a noise term from one time step to the next), on the system. These restrictions are unimportant, since control engineering methods for the modelling of more complex noise processes are well developed, (Kwakernaak and Sivan, 1972).

The discussion will focus on computational methods for evaluating the sensitivity of a model to noise. The question of the form and size of noise terms that should be incorporated into the model will not be considered. For a discussion of these the reader is referred to Peterson's paper in this volume. A cruder approach was given by Sharp (1974).

The Perturbation Approach to computing the Effects of Noise on the System

As an illustration of the type of analysis under discussion it is convenient to consider a simple example. Comprehensive discussions have been given by Sharp (1976b), (1974).

We consider the simple model of equation (1) to which a random noise term R(N) with zero mean and no serial correlation has been added to represent the effects of structural error. To make things more definite we now take the value of C as -0.1.

The system we wish to consider is therefore

$$X(N+1) = X(N) - DT*X(N)/10 \qquad (11)$$

or with the addition of the random noise term

$$Y(N+1) = Y(N) - DT*Y(N)/10 + R(N) \qquad (12)$$

both with initial condition

$$X(0) = Y(0) = 1$$

If as before, we define D(N) = Y(N) -X(N), then D(N) represents the effects of noise on the system up until time N. In statistical terms D(N) of course a random variable since different noise sequences will give rise to different values of D (N).

Subtraction of equation (11) from equation (12) gives:

$$D(N+1) = D(N)\ (1-DT/10) + R(N) \qquad (13)$$

Equation (13) is of no use as it stands because D(N+1) is a random variable. It is therefore necessary to derive from this equation the statistical characteristics of D(N+1). If we denote the statistical expectation operator by E( ), then we require the mean of D(N+1) = E(D(N+1)) and its variance E(D(N+1)**2) - E(D(N+1))**2.

From equation (13) we have

$$E(D(N+1)) = E(D(N))(1-DT/10) + E(R(N)) \qquad (14)$$

Since by assumption the mean of R(N) is zero $E(R(N)) = 0$. Similarly $D(0) = 0$. Equation (14) therefore gives $E(D(N+1)) = E(D(N)) = \ldots E(D(1)) = E(D(0)) = 0.$
To compute the variance of D(N+1) we therefore need only compute the term E(D(N+1)**2). From equation (13) this is

$$E(D(N+1)**2) = E((1-DT/10)**2*D(N)**2 + 2*(1-DT/10)*D(N)*R(N) + R(N)**2) \qquad (15)$$

Since the R(N) are by assumption not serially correlated equation (15) reduces to:

$$E(D(N+1)**2) = (1-DT/10)^2*E(D(N)**2) + E(R(N)**2) \qquad (16)$$

The value of E(R(N)**2) - the variance of R(N) - i , of course, fixed by the modeller at some value V. Equation (16) therefore becomes:

$$E(D(N+1)**2) = E(D(N)**2)*(1-DT/10)**2 + V \qquad (17)$$

Since $E(D(0)**2) = 0$ by assumption equation (17) enables us to successively compute E(D(1)**2), then E(D(2)**2), etc.

In this simple case it is easy to show that as N increases E(D(N+1)**2) fairly rapidly approaches a limiting value of

$$E(D(N+1)**2) = \frac{V}{(2DT/10 + DT**2/100)} \qquad (18)$$

or since the term DT**2/100 will be negligible in practice:

$$E(D(N+1)**2) = \frac{5V}{DT} \qquad (19)$$

The approach just described is easily generalized to more complex systems and to deal with noise in inputs such as SALES RATE or uncertainties in initial conditions. As in the comparable case for local parameter sensitivity the computations involve the use of matrices. The matrices are in fact the same as those used in parameter sensitivity analysis. The perturbation approach therefore allows parameter and noise sensitivities to be determined in a single run using less computer time than the sum of the times taken for either type of sensitivity analysis singly. As will be shown later, no such advantages exist with the Conventional methods of determining parameter and noise sensitivities.

The perturbation method of determining of noise sensitivities is an approximate method. Its computer time requirements are given by $K(3)\dot{V}^2/DT$, where, as before, V is the number of model variables, DT the time step and K(3) is a constant that depends on the computer used, (Sharp, 1974).

The Conventional Method of Determining Noise Sensitivities

The Conventional Method (Monte Carlo) of determining the sensitivity of the system to noise is best illustrated by reference to the example of equations (11) and (12). The model is set up in DYNAMO form as:

```
L   X.K =X.J-DT*(X.J/10)+R.J
N   X=1
A   R.K=NORMRN(Ø,√V)*NS
```

A run is first carried out on the noise free system by setting NS=0. NS is then set equal to 1 and a number of runs carried out with noise. The value of D(N) for a particular run with noise is computed by subtracting from the value of X(N) for that run the corresponding X value from the noise free run. Means and variances are computed by the usual numerical formulae. This process is easy to carry out with a compiler such as DYSMAP that generates a FORTRAN program that can be suitably modified. With ordinary DYNAMO compilers it is not quite so straightford since they provide no mechanism for using the X - value generated by the first noise free in the later runs nor for storing the sums of squares, etc. necessary for the computation of variances.

This approach involves no approximations. The computer time requirements can be shown to be equal to K(4) V/DT, (Sharp, 1974). This is therefore the only method for computing some aspect of model sensitivity that has been discussed that does not involve computer time requirements increasing with the square of model size. Unfortunately the efficiency of the method is more apparent than real. Firstly computer random number generators as used by DYNAMO function NORMRN required a lot of computation. Secondly, it can be shown, (Sharp, 1976b) that even to estimate the standard deviation of thê effects of noise on the value of X to within 50% requires about 70 runs and therefore, as will be shown later, even for quite large models the computer time requirements are a lot higher than for the perturbation method. Only for models involving thousands of variables is it even likely to be attractive with regard to computer time. For models of this size however the computations for the perturbation method could be speeded up by various approximate methods so even for them it is somewhat doubtful.

It is worth noting however, that is we do not wish to estimate accurately the standard deviation of D(N+1) (and hence its variance) but merely wish to confirm that it is different from some chosen value then the Conventional Method may be attractive as far as computer time requirements are concerned.

Suppose, for example, that we are concerned with a library system and an important measure of service is the time a would-be borrower must wait for a book that he has requested and which the library does not own. Suppose that with a certain set of policies the average waiting time is 15 days. In many circumstances it might be considered satisfactory if in the noisy conditions that prevail in the real system the standard deviation of this waiting time were less than 2 days. It is therefore necessary to use the model to test the hypothesis that the standard deviation of the waiting time

is less than 2 days. Where the actual standard deviation is significantly (say 50%) less or greater than a value of 2 days this can be done using a t-test with only a few model runs.

The Effect of DT in Examining Noise Sensitivities

In computing noise effects by the conventional method care must be taken to choose the variances of noise terms with reference to the size of DT. If this is not done a halving of DT will produce a dramatic - and totally spurious - apparent change in the effects of noise on the system. Since the usual modelling practice is to choose a suitably small time step that the analyst varies, if necessary, to ensure numerical accuracy but otherwise to ignore the actual size of DT completely, it seems worth drawing attention to this fact. The reasons for it are buried deep in the somewhat counterintuitive theory of stochastic differential equations, (Kwakernaak and Sivan, 1972).

The point however is easily illustrated for the model discussed earlier in this section by reference to equation (19). If the value of V chosen is 1/640 and DT is chosen as 1/32 the variance of D(N+1) tends to $5 \times 32/640 = \frac{1}{4}$. If the modeller now changes DT to 1/64 without changingV the variance of D(N+1) will magically change to ½! While DT is kept constant during the modelling activity no problem arises - provided, of course, that the noise variances appropriate for the system under study and the time step being used were properly chosen initially. Very often, however, after certain redesign work on the system the analyst finds he can increase the size of DT and does so to reduce computing costs. If he does not at the same time adjust noise variances accordingly, he is in danger of drawing erroneous conclusions about the sensitivity of the improved system to noise.

IV MODEL EXPERIMENTS

To gain a feeling for the actual performance of sensitivity analysis methods there is no substitute for actual tests. Various experiments were accordingly carried out on the model given in the appendix. These experiments are described in detail in Sharp, 1976b. The purpose of the present discussion is to summarize the conclusions of that more detailed study.

Reasons for Choice of Model

A number of factors dictated the choice of model.

a) Most SD applications involve the use of a model that is driven by one or more inputs, e.g. Order Rate. A model of a production planning system was therefore chosen as being more 'typical' of actual applications than the World 2 model with which most workers seem to have experimented.

b) A small model was required so that a number of experiments could be carried out quickly and so that the partial differentiation of the model equations required by the perturbation methods would not be too burdensome.

c) The model should contain nonlinearities to test how well various methods could cope with them and a performance measure of a type that might be used in practice.

The model chosen contained 7 state variables (LEVELS), F,S,I,B,C,M,G. Most of the LEVEL equations are straightforward. That for M represents a quadratic performance measure of a type frequently used in control engineering. A similar type of measure has been applied to production planning systems by Holt et al., 1961.

The measure M contains 3 terms; the first penalizes deviations of inventory from desired inventory; the second pelalizes rapid changes in production rate and the third high order backlogs. The constant W2 determines the weighting of production change penalties relative to those for deviations of inventory from desired inventory and the constant W3 the weighting of backlog penalties relative to those for deviations of inventory from desired inventory.

The equation for G demonstrates a way in which it is possible to introduce complicated noise terms into an SD model. Its purpose is to ensure that there is a degree of serial correlation and intercorrelation between the noise effects as might frequently be found in practice. Thus a random event such as a change in quality of incoming raw materials will have effects that may be expected to persist for some time (serial correlation) and also to impinge on various parts of the system (intercorrelation). This approach to the modelling of noise is common in control engineering (c.f. Kwakernaak and Sivan, 1972). It was introduced into System Dynamics by Jay Forrester (1961) but unfortunately seems to have received little attention since then.

The RATE and AUXILIARY equations are again fairly standard in form except that Despatch Rate (D). This was chosen to be a fairly complex nonlinear function of a form that ensures that Despatch Rate becomes zero when Inventory reaches some minimum level greater than zero. The input to the system Order Rate was taken as the sum of a constant, a term representing seasonal variation and a term representing a longer term business cycle variation. Such test inputs are very useful in assessing the performance of systems of this type.

Two types of sensitivity analysis were carried out on the model, local parameter sensitivity analysis in which the effects of varying each parameter by 1% were studied; and a stochastic sensitivity analysis to determine the effects of setting the noise switch (NS) in the model equal to 1.

The local parameter sensitivity analysis was carried out for the CONSTANTS, TAU, KAPPA, LAMBDA, ALPHA, RHO, NU, MU, SIGMA, THETA, W2 and W3. The first 9 constants need little explanation. The last 2 are perhaps more interesting. They enable us to answer the question: if we weight production rate changes or order backlogs more heavily, how will this affect our perfprmance measure.

Local parameter sensitivity analysis was also applied to the initial values FINV, IINV, BINV, SINV and CINV. No sensitivity analysis was carried out on the initial values of M or G. The initial value of M is arbitrary and may as well be set to zero by the analyst. The variable G is a random variable with zero mean and the natural initial value is this mean value.

In all then, local parameter sensitivity analysis was carried out for 16 parameters (11 constants plus 5 initial values).

Assessing Performance for larger models

As mentioned earlier there were good practical reasons for experimenting with a small model. Nonetheless most models used in actual applications are likely to be of the order of ten times the size of the one described. A 'pseudo-model' ten times the size of the one described here was constructed from the model in the appendix.

This involved running the model in a suitable way to reproduce the computational effort required to deal with a problem ten times bigger. The procedure is best illustrated by the process of local sensitivity analysis. The problem described has seven state variables and eleven parameters. To simulate the process of local sensitivity analysis for a model with seventy state variables and one hundred and sixty parameters, the sensitivity analysis for each of the eleven parameters was carried out ten times and at each time step of the model run each model equation was computed ten times.

The pseudo-models made it possible to assess the computer time requirements for the different methods on bigger models.

Computer Used for the Experiments

The experiments were carried out on an ICL 1904S. This is a medium speed scientific machine. Basic instructions such as multiply two numbers together typically require several microseconds. A large scientific computer such as a CDC 7600 would probably have carried out the calculations described some fifty times faster.

Computational Aspects of Experiments

In the experiments that were carried out a number of technical 'hygiene factors'of good computing practice were found to be very important in reducing the time required for both types of sensitivity analysis whatever method was used to carry them out. These are well known to computer scientists but in the author's experience are unknown to most SD practitioners. These will therefore be treated in some detail.

In the earlier discussions the reader will have noted that the time required for any sensitivity analysis method is universely proportional to the size of the simulation time step (DT). Therefore to reduce computer time requirements it is necessary to select the largest value of DT compatible with reasonable numerical accuracy. Most practitioners on the other hand tend to select a time step that they know will not give rise to accuracy problems and this is generally much smaller than necessary. Thus for the model described the time step originally selected was DT = 0.125. Experiments showed however that DT = 1 gave satisfactory results. In other words a couple of simple model runs enabled the time for sensitivity analysis to be reduced by 87.5% ! (The reader who is unconvinced by this may care to confirm that Forrester's World 2 model runs quite satisfactorily with DT = 1 instead of DT = 0.25 as used by Forrester).

In theory there are more sophisticated methods of computing a model than the Euler method used by DYNAMO. One such method is the Runge Kutta method (Henrici, 1962). This method was tried on the model described but took twice as long as the usual Euler method. Though it may, of course, prove quicker for some models it was not so in this case.

Once the maximum feasible time step is determined the process of sensitivity analysis can begin. At this stage however the software of the computer being used begins to have an important effect.

A DYNAMO model has to be translated or compiled before the computer can run it. For the DYSMAP language used at Bradford, 2 compilers are available. The compiler that is usually used is designed to compile a program very quickly but to produce computer code that is rather slow to run. In most university computer systems the reality is that the majority of computer programs fail during compilation which is why most installations generally use such compilers. On the other hand most installations generally have available an optimizing compiler that compiles the program in such a way as to minimize the computer time taken to run it. Such compilers take longer to compile a program which is why in general it is necessary to specify that they be used. In sensitivity analysis however the extra compilation time is handsomely compensated for by the reduction in time required for the sensitivity analysis. It was found that the time required for sensitivity analysis by the conventional method was reduced by about 25% when the optimizing compiler was used. For technical reasons the use of an optimizing compiler was even more beneficial with the perturbation methods reductions in computing time of 50% being achieved.

Once the program has been compiled another factor comes into play. It is usual to run DYNAMO programs in conjunction with an error-trapping package that detects, for example, where the user's program has attempted to divide by zero. The use of such packages increases computer run time considerably. Thus for the seven state variable model and one of the perturbation methods a local sensitivity analysis took 32 seconds with the error-trapping package and five seconds without it.

Finally in trying to increase computational efficiency it is necessary to bear in mind that the setting up of PRINT and PLOT output requires a great deal of computer time. A full local sensitivity analysis for parameters and initial conditions together with PLOT and PRINT information took 510 seconds and produced 15000 lines of output. The PLOT and PRINT commands were replaced by output of a summary of the information generated during the run that occupied only 50 lines of lineprinter output. The time required for this run was only 8 seconds.

To summarize the discussion of this section the modeller should, if he wants to minimize the computer time required for sensitivity analysis:

i) use the largest time step compatible with reasonable accuracy.

ii) use an optimizing compiler to compile the program

iii) not use an error-trapping package

iv) reduce the amount of line printer output as far as possible.

Comparison of Sensitivity Analysis Methods - Accuracy

The discussion of section II shows that the perturbation method is accurate for Local Parameter Sensitivity Analysis. Comparison of the results obtained from it with those obtained by using the Conventional method showed this was the case.

For stochastic sensitivity analysis the Perturbation method gives only approximate answers whereas the Monte Carlo method is capable of giving as accurate an answer as desired, if enough runs are carried out. The question therefore arises as to whether the approximate answers given by the Perturbation method are good enough to be useful in practice.

Comparing the results of the perturbation method and the Monte Carlo method, the former gave excellent results apart from the variables B and M, where the standard deviations calculated were about 50% and 40% of the true values, respectively. However, the noise signals have deliberately been chosen to be large. A comparison of Figures 1 and 2 shows that the introduction of these noise signals has a drastic effect on system performance, in that the graphs of any variable, for example, C differ markedly. Indeed it is difficult to believe that the time series of Figure 1 and those of Figure 2 are produced by the same system.

The final value of M in the noise free run, for example, is about 2.25E08. Its mean value for the runs with noise is about 2.ØEØ9 - - very far from the value of zero assumed by the perturbation methods and 8 times the value attained in the noise free run. Its computed standard deviation is extremely large relative to the mean and, since M cannot be negative, implies a very skewed distribution. The standard deviations of C, I and B are very large. In fact, in many runs these variables actually take on negative values. In other words, the performance of the system with these noise inputs is quite unacceptable. The results of the perturbation methods are accurate enough to indicate the need for redesign in this case, that is, they are quite acceptable for measurement of system robustness.

Comparison of Sensitivity Analysis Methods - Time Requirements

The experiments carried out involved a number of different versions of the perturbation method and various combinations of sensitivity analysis, (Sharp, 1976b). The results are summarized in Table 1 which shows the best results attained for each method of sensitivity analysis. All the times given are for the 70 state variable, 160 parameter pseudo model.

Table 1 Computer time required for various sensitivity analyses

	Method of Sensitivity Analysis	
Type of Sensitivity Analysis	Conventional	Perturbation
Local Parameter Sensitivity (170 parameters)	705 seconds	74 seconds
Stochastic Sensitivity (Compute variances of effects noise terms of model in appendix)	1720 seconds	+
Stochastic and Local Parameter Sensitivities combined	2420 seconds	100 seconds

+ This experiment not carried out. Note however figure for both types of analysis combined.

Choice of Sensitivity Analysis Method

As Table 1 shows, if we wish to carry out local parameter sensitivity analysis for all parameters or a stochastic sensitivity analysis the perturbation methods can substantially reduce the computer time required.

The conventional method of parameter sensitivity analysis has however certain advantages compared with the perturbation method. The perturbation method is an all or nothing method. It takes almost as much time to carry out a sensitivity analysis for 1 parameter as for 170. To carry out local parameter sensitivity analysis for 1 parameter by the conventional method requires only 1/170th of the time required for all 170 parameters.

The conventional method is therefore preferable where local parameter sensitivity analysis is only required for a few parameters. Furthermore, the utility of the perturbation method is much reduced at present by the fact that partial differentiation of the model equations must be carried out before it can be used.

As was noted in earlier discussions, the conventional method for stochastic sensitivity analysis should be quicker than the perturbation method for models beyond a certain size. If, as here, it is desired to compute variances of the effects of noise terms on the model then this point has clearly not been reached. Of course if we only wish to test whether these variances are significantly different from a certain value then as was pointed out in section III the conventional method may be preferable particularly since it is at present easier to apply.

For either type of sensitivity analysis the perturbation methods work best where we wish to evaluate the sensitivities of a relatively small number of performance measures. The figures given above are for a model in which only 7 out of the 70 state variables are performance measures. The conventional method is relatively insensitive to the number of performance measures (providing lineprinter output volumes are kept down).

The factors that favour the choice of one method of computing sensitivities rather than the other are shown in Table 2.

Table 2 Factors affecting choice of Sensitivity Analysis Method

Factors favouring Conventional Method	Factors favouring Perturbation Method
Small number of parameters whose sensitivity is to be determined	Parameter sensitivities required for large number parameters
Large number of performance measures	Small number of performance measures
Required to test effects noise different from specified value	Required to compute variance noise effects
Method needs to be simple to apply	Combined Parameter and Noise Sensitivities required
Very accurate measurement of stochastic sensitivities required	

V CONCLUSIONS

The following conclusions to this paper appear reasonable:

a) The time required for full sensitivity analyses of major problems by the conventional and Monte Carlo methods is probably far shorter than assumed by most analysts. This insight opens up the prospect for using sensitivity analyses in conjunction with hill-climbing programs on a wider scale than at present. In view of the results of Table 1, it would seem worth while to design a special efficient compiler for sensitivity analysis purposes.

As far as stochastic sensitivity is concerned, the biggest problem would seem to be determining the size and form of the noise signals. Peterson's paper in this volume indicates how this can be done.

b) The perturbation methods appear to yield results that are accurate enough for the measurement of robustness in practice. If further effort were devoted to improving their computational efficiency, they should be capable of producing very rapid sensitivity analyses. Perturbation methods are particularly well-suited to cases where large numbers of parameters and stochastic effects are present, and offer considerable benefits if both parameter and stochastic sensitivities must be assessed together. They are probably capable of further refinement to produce rapid approximate answers. Indeed for large systems that break up into a number of weakly connected subsystems, the perturbation methods seem to lend themselves to the construction of approximate methods based on block diagonal matrices (Sharp 1974). The biggest obstacle to their use is likely to be the need for partial differentiation of the model equations. It would therefore seem well worth exploring the use of a programming language, such as ALTRAN (Brown), to carry out the necessary partial differentiations. This facility would be useful elsewhere in System Dynamics, for instance, for some of the calculations described in the papers by Peterson and Thissen in this volume.

c) With suitable developments to compilers, sensitivity analyses should become as natural a part of the model building and policy development as normal runs are at present. The extent to which system robustness had been attained would then be apparent not only to the analyst, but also to those responsible for assessing his recommendations.

References

Barnett A.B., 1973, A system dynamics model of oilfield development, Unpublished Ph.D. thesis, University of Bradford.

Brown W.S., 1973 ALTRAN User's Manual, Bell Telephone Laboratories, Holmdel, New Jersey.

Coyle R.G., 1975 Management System Dynamics, System Dynamics Research Group, University of Bradford.

De Jong, J.L. & Derksen J.W. 1975 The application of gradient algorithms to the optimization of controlled versions of the World 2 Model of Forrester, IFIP Conference, Nice, September 8th-14th.

Doerer H.T., 1974 A study of analytic methods for the analysis and design of Decision Policies in I.D. models, Ph.D., Arizona State.

Forrester J.W. 1961 Industrial Dynamics, MIT Press

Henrici P, 1962 Discrete Variable Methods in Ordinary Differential Equations, Wiley.

Holt C.C., Modigliani F., Muth J.F., Simon H.A., 1960. Planning Production Inventories and Workforce, Prentice Hall.

I.C.I. 1969 Nonlinear Optimization Techniques, I.C.I. Monograph No. 5. Oliver & Boyd

Kwakernaak H & Sivan, R. 1972 Linear Optimal Control Systems, Wiley 1972

Low G.W., 1976 The principle of convervation and the multiplier accelerator theory of business cycles, This Volume.

Naylor T. et al., 1969. Spectral analysis of data generated by simulation experiments, Econometrica, Vol. 37, pp 333-352

Peterson D.W.1976, 'Are there cookery book procedures for establishing the 'goodness' of model structures and parameters. This Volume.

Rademaaker O., 1973 On understanding complicated models. Simple methods. U.S.- Soviet Conf. on methodological aspects of Social Systems simulation.

References Cont'd

Ratnatunga A.K., 1975 DYSMAP User's Manual, System Dynamics Research Group, University of Bradford.

Sharp J.A., 1974 A study of some methodological problems of system dynamics. Unpublished Ph.D. thesis, University of Bradford.

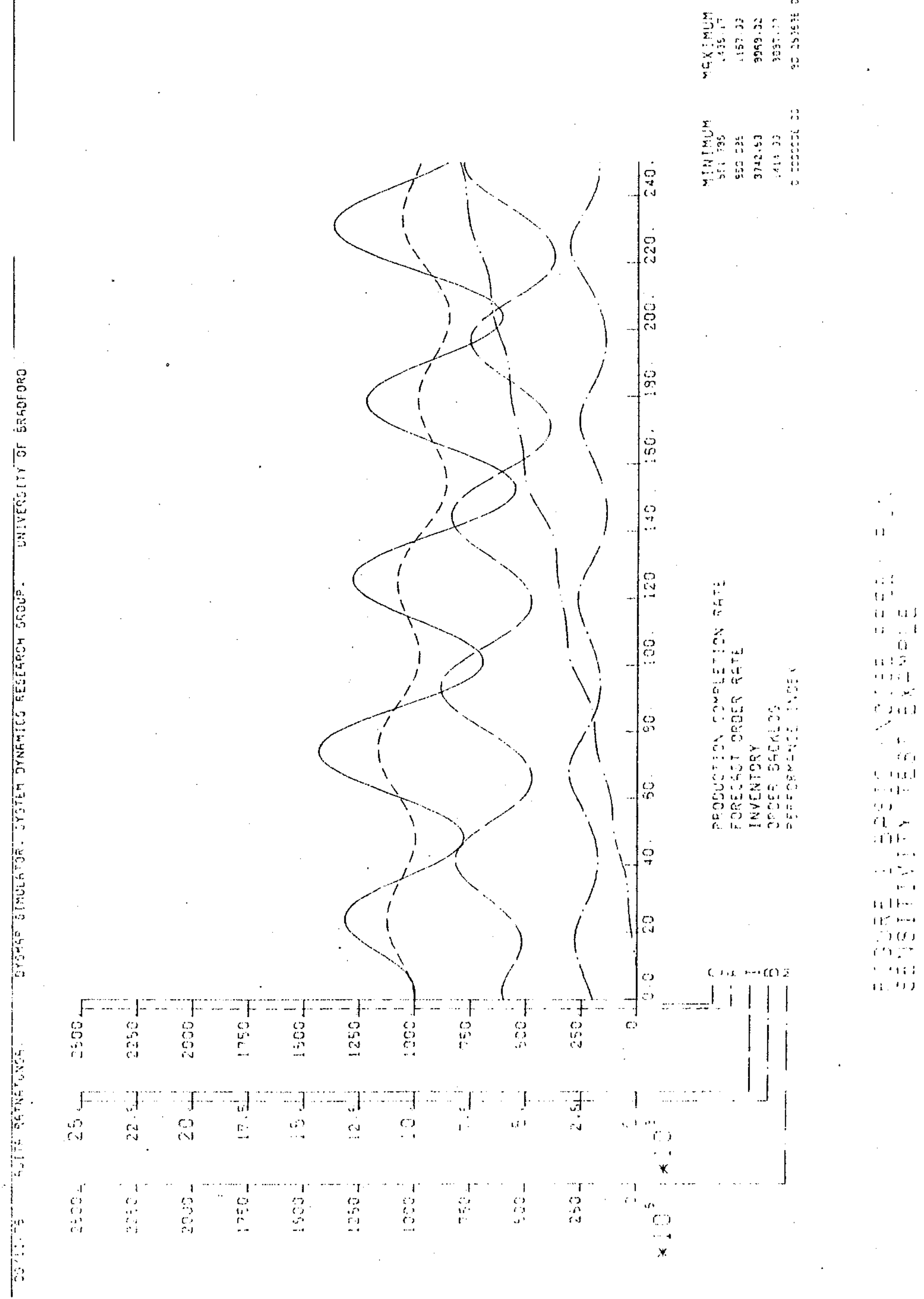
" 1975 The work of the System Dynamics Research Group at the University of Bradford, O.R. Society Conference.

" 1976a) The role of forecasts in System Dynamics models, Dynamica, System Dynamics Research Group, University of Bradford. Vol. 2, pp 50-61

" 1976b) Some comparisons of Sensitivity Testing Methods for System Dynamics models, Working Memorandum, System Dynamics Research Group, University of Bradford.

Thissen W., Some methods of acquiring insight into the working of complicated mathematical models, This Volume.

Tomovic R. & Vukobratovic M., 1972 General Sensitivity Theory, Elsevier.



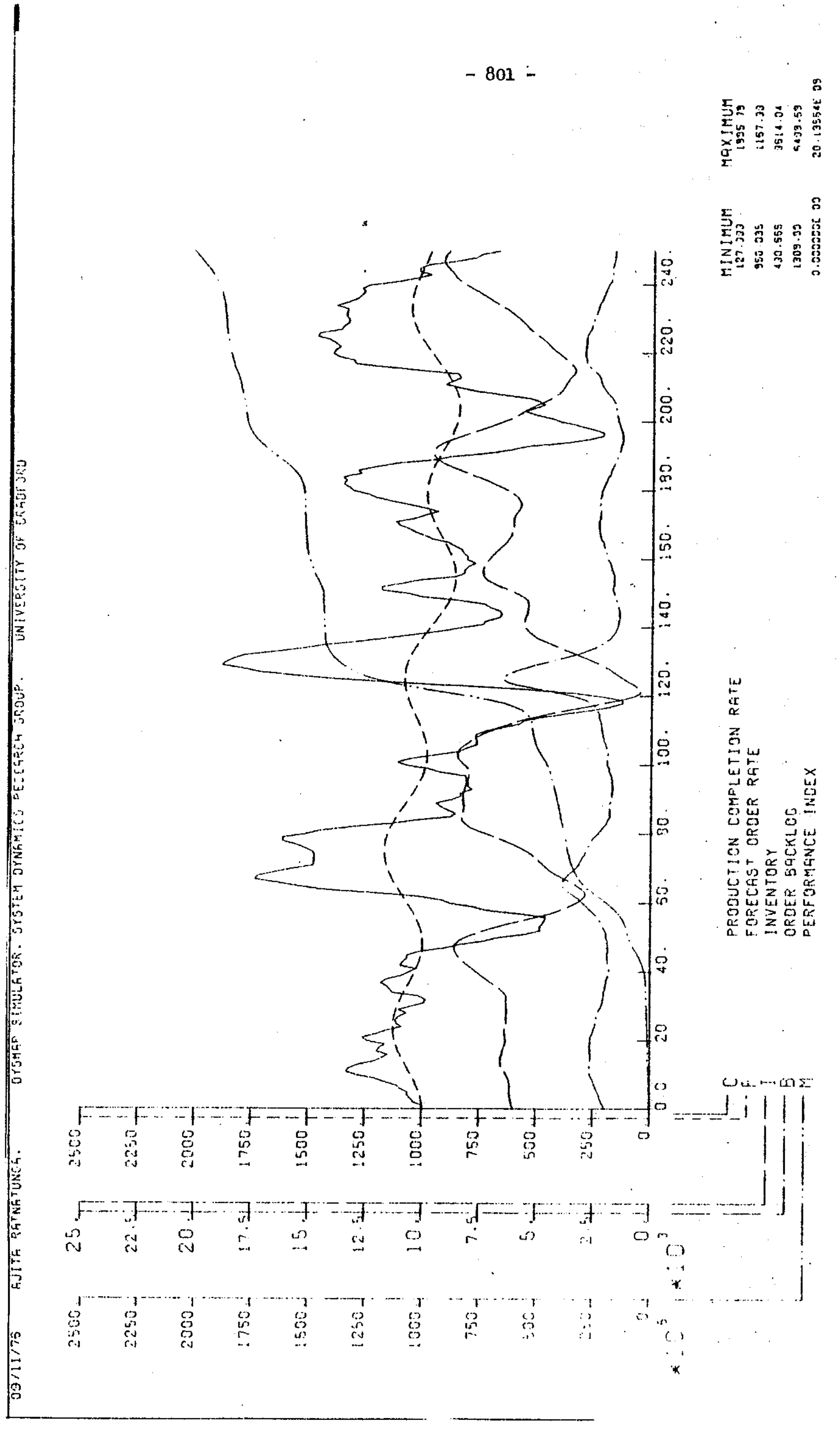


FIGURE 2 BASIC SYSTEM WITH NOISE
SENSITIVITY TEST EXAMPLE

<u>Appendix Model used for Tests</u>

```
 0 *   SENSITIVITY TEST EXAMPLE
 1 A   E1.K=NORMRN(0.,6.)*NS
 2 A   E2.K=NORMRN(0.,4.)*NS
 3 A   E3.K=NORMRN(0.,8.)*NS
 4 A   E4.K=NORMRN(0.,9.)*NS
 5 A   E5.K=NORMRN(0.,5.)*NS
 6 A   E7.K=NORMRN(0.,4.)*NS
 7 R   O.KL=1000+250*SIN(6.28*TIME.K/220)
 8 D   O=(UNITS/WK)  ORDER RECEIPT RATE
 9 R   P.KL=F.K+((R.K-I.K)+(B.K-E.K))/TAU
10 D   P=(UNITS/WK) PRODUCTION START RATE
11 A   R.K=KAPPA*F.K
12 D   R=(UNITS) REQUIRED INVENTORY
13 A   E.K=NU*F.K
14 D   E=(UNITS) EXPECTED BACKLOG
15 R   D.KL=B.K*(1-EXP(-ALPHA*I.K/S.K))/NU+RHO*B.K*(I.K-THETA)
16 D   D=(UNITS/WK) DESPATCH RATE
17 L   C.K=C.J+DT*(P.JK-C.J)/MU+DT*(E1.J+50*G.J)
18 N   C=CINV
19 D   C=(UNITS/WK) PRODUCTION COMPLETION RATE
20 L   F.K=F.J+DT*(O.JK-F.J)/LAMBDA
21 N   F=FINV
22 D   F=(UNITS/WK) DEMAND FORECAST
23 L   I.K=I.J+DT*(C.J-D.JK)+DT*(E3.J+30*G.J)
24 N   I=IINV
25 D   I=(UNITS/WK) FINISHED GOODS INVENTORY
26 L   B.K=B.J+DT*(O.JK-D.JK)+DT*(E4.J+30*G.J)
27 N   B=BINV
28 D   B=(UNITS) ORDER BACKLOG
29 L   S.K=S.J+DT*(O.JK-S.J)/SIGMA+DT*E5.J
30 N   S=SINV
31 D   S=(UNITS/WK) SMOOTHED ORDER RATE
32 A   A1.K=(R.K-I.K)*(R.K-I.K)
33 A   A2.K=(P.KL-C.K)*(P.KL-C.K)
34 L   M.K=M.J+DT*(A1.J+U2*A2.J+U3*B.J**2)
35 N   M=0
36 D   M=(1) MEASURE SYSTEM PERFORMANCE
37 L   G.K=G.J+DT*(E7.J-G.J)/4
38 N   G=0
39 D   G=(1)  NOISE TERM
```

Model Continued

```
40 NOTE CONSTANTS
41 C  TAU=8
42 D  TAU=(WK) PRODUCTION SMOOTHING TIME
43 C  KAPPA=6
44 D  KAPPA=(WK) DESIRED NUMBER OF WEEKS ORDERS IN STOCK
45 C  NU=2
46 D  NU=(WK) EXPECTED NUMBER WEEKS ORDERS IN BACKLOG
47 C  LAMBDA=20
48 D  LAMBDA=(WK) FORECAST SMOOTHING TIME
49 C  ALPHA=0.50
50 D  ALPHA=(1/WK) EFFECT INVENTORY SHORTAGE ON DESPATCHES
51 C  RHO=0.000005
52 D  RHO=(1/UNITS/WK)  EFFECT BACKLOG AND INVENTORY ON DESPATCHES
53 C  THETA=6000
54 D  THETA=(UNITS)  BREAKPOINT FOR CHANGE IN INVENTORY AND BACKLOG EFFECT
55 C  MU=6
56 D  MU=(WK)  PRODUCTION COMPLETION DELAY
57 C  SIGMA=4
58 D  SIGMA=(WK) SMOOTHING TIME
59 C  W2=20
60 D  W2=(WK**2/UNITS**2) WEIGHTING CONSTANT
61 C  W3=0.05
62 D  W3=(1/UNITS**2) WEIGHTING CONSTANT
63 C  CINV=1000
64 D  CINV=(UNITS/WK) INITIAL VALUE C
65 C  FINV=1000
66 D  FINV=(UNITS/WK) INITIAL VALUE F
67 C  IINV=6000
68 D  IINV=(UNITS) INITIAL VALUE I
69 C  BINV=2000
70 D  BINV=(UNITS) INITIAL VALUE B
71 C  SINV=1000
72 D  SINV=(UNITS/WK) INITIAL VALUE S
```