

Linearization and Order Reduction in System Dynamics Models

By

A.K. Ratnatunga & J.A. Sharp
System Dynamics Research Group
University of Bradford

Abstract

The utility of linearization and order reduction of System Dynamics models is discussed.

Two methods are presented for linearizing a System Dynamics model and the results of implementing them on an actual model are given.

Introduction

The objective of carrying out a System Dynamics study is to redesign the system to improve its behaviour. In the redesign process it is of particular interest to produce a ROBUST system, (Sharp, 1975), i.e. one that functions satisfactorily whatever inputs it is subjected to and whose performance is insensitive to likely parameters and structural errors. Because of the non-linearity of System Dynamics models however no analytical procedure exists for designing ROBUST systems. Such methods are though available for linear systems so an approximate linear model could be extremely useful in the early stages of redesign in that standard analytical techniques to produce control laws that lead to suitable behaviour in the linear system that could then be tested on the non-linear model. The non-linear systems of System Dynamics often show behaviour that approximate closely to that of a linear system so that linearization is not perhaps as ambitious as at first appears. In particular many of the nonlinearities that occur, e.g. desired production exceeding capacity so that production has to be set equal to the minimum of these two terms, tend to be inoperative once a proper control system has been devised.

Furthermore it seems plausible that if the linear model is a reasonable approximation to the nonlinear one and a suitable ROBUST set of control laws can be devised for it, then these laws will be relatively insensitive to the effects of the approximation and should therefore stand a good chance of working satisfactorily for the nonlinear system also.

The aim of such linearization is, of course, to provide a basis for the rapid and systematic design of policies that will serve as a starting point for further experiments with the nonlinear system.

Previous work (e.g. Thillainathan, 1975) suggests that such an approach can often provide useful insights and relatively good control rules. In practice however the linearization of System Dynamics models is a tedious and error-prone process if carried out by means of partial differentiation. The aim of this paper is to demonstrate a method whereby linearization of a System Dynamics model can be carried out by numerical techniques which are easily implemented with a precompiler such as DYSMAP or some versions of DYNAMO that generates a FORTRAN program.

The linearization process outlined is also suitable for other purposes, e.g. for the generation of the interaction matrices used by McLean and Shepherd, 1976.

It can also be used to map a nonlinear system on to one of lower order. It is observed in many models (c.f. Rademaaker, 1973) that certain of the State Variables are effectively constant over the simulation range. It can therefore be useful to construct a lower order linear model that is easier to work with, which does not include such state variables explicitly.

Again in the interests of securing a lower order model that is easier to analyze it is often desirable to replace a RATE generated by say, a DELAY3 macro by a linear combination of other state variables, thereby eliminating the hidden state variables of the macro.

Mathematical Procedure for Obtaining the Approximate Linear Model

System Dynamics models can be expressed in the form, (Sharp, 1974)

$$\underline{x}_{n+1} = \underline{x}_n + DT * f(\underline{x}_n, \underline{i}_n) \quad (1)$$

Assume that we wish to represent the output of some subset \underline{x} of these variables by a linear system

$$\underline{x}_{i,n+1} = \underline{x}_{i,n} + DT * (A_{ij} \underline{x}_{j,n} + B_{ik} \underline{i}_{k,n} + c_i) \quad 1 \leq j \leq J \quad 1 \leq k \leq K \quad (2)$$

In fact the variables $\underline{x}_{i,n}$ and the inputs $\underline{i}_{k,n}$ will vary about some average operating level over the simulation (\underline{x}_{av} and \underline{i}_{av})

$$\begin{aligned} \underline{x} &= \underline{x}' + \underline{x}_{av} \\ \underline{i} &= \underline{i}' + \underline{i}_{av} \end{aligned}$$

and attempt to identify the system relative to its operating trajectory as

$$\underline{x}'_{i,n+1} = \underline{x}'_{i,n} + DT * (\sum_j A_{ij} \underline{x}'_{j,n} + \sum_k B_{ik} \underline{i}'_{k,n} + c'_i) \quad (3)$$

where

$$c'_i = A_{ij} x_{j,av} + B_{ik} c_{i,av}$$

returning to (1) and writing $\underline{X} \equiv \{x, y\}$ and $y \equiv y_{av} + y'$ we have

$$x'_{i,n+1} = x'_{i,n} + DT * f((x'_n + x_{av}), (i'_n + i_{av}), (y'_n + y_{av})) \quad (4)$$

expanding (4) as a Taylor series and ignoring terms in y'

$$x'_{i,n+1} = x'_{i,n} + DT * \left[\left\{ \frac{\partial f_i}{\partial x_j} \right\}_{x_{av}, i_{av}, y_{av}} x'_{j,n} + \left\{ \frac{\partial f_i}{\partial i_k} \right\}_{x_{av}, i_{av}, y_{av}} i'_{k,n} + f_i(x_{av}, i_{av}, y_{av}) \right] \quad (5)$$

thus from (3) and (5)

$$A_{ij} \approx \left\{ \frac{\partial f_i}{\partial x_j} \right\}_{x_{av}, i_{av}, y_{av}} \quad (6)$$

$$B_{ik} \approx \left\{ \frac{\partial f_i}{\partial i_k} \right\}_{x_{av}, i_{av}, y_{av}}$$

$$c'_i \approx f_i(x_{av}, i_{av}, y_{av})$$

Estimation of A_{ij}

To estimate the matrix A_{ij} it is necessary to:

- i) Run model (1)
- ii) Decide which state variables should appear in the linear model
- iii) Decide by examining the nonlinear model which - if any - of the A_{ij} are zero
- iv) Run model with uncorrelated and non-serially correlated noise terms ξ_j (PRBS) added to each state variable of interest

If we assume that the state vector \underline{X}' corresponding to this output is written as

$$\underline{X}' \equiv \left(\{x'' + x_{av}\}, \{y' + y_{av}\}, \{i' + i_{av}\} \right) \quad (7)$$

then it satisfies the equation

$$\underline{X}'_{n+1} = \underline{X}'_n + DT * f(\{x'' + x_{av} + \underline{x}\}, \{y' + y_{av}\}, \{i' + i_{av}\}) \quad (8)$$

We now assume that the variables \underline{X} in (1) are generated by a linear system of the form (3).

It then follows that the equations for the subset of state variables \underline{x}'' can be written

$$x'_{i,n+1} = x'_{i,n} + DT * (\sum_j A_{ij} (x''_{j,n} + \xi_{j,n}) + \sum_k B_{ik} i'_{k,n} + c'_i) \quad (9)$$

subtraction of (3) from (9) gives:

$$x'_{i,n+1} - x'_{i,n+1} = x'_{i,n} - x'_{i,n} + DT * \sum_j A_{ij} (x'_{j,n} - x'_{j,n} + \xi_{j,n}) \quad (10)$$

or

$$x'_{i,n+1} - x'_{i,n} - x'_{i,n+1} + x'_{i,n} = DT * (\sum_j A_{ij} (x'_{j,n} - x'_{j,n} + \xi_{j,n})) \quad (11)$$

Clearly the left hand side of (11) represents the output of a stationary process.

Cross-correlation of (11) with $\xi_{j,n}$ gives, assuming that $\xi_{j,n}$ are uncorrelated random variables (PRBS), that:

$$E(\xi_{j,n}^2) * A_{ij} = \frac{1}{DT} * E\left\{\xi_{j,n} (x'_{i,n+1} - x'_{i,n} - x'_{i,n+1} + x'_{i,n})\right\} \quad (12)$$

or

$$\hat{A}_{ij} = \frac{1}{E(\xi_{j,n}^2)} * E\left\{\xi_{j,n} (x'_{i,n+1} - x'_{i,n} - x'_{i,n+1} + x'_{i,n})\right\} \quad (13)$$

It should be noticed that unlike certain other applications of correlation methods there is no problem in this case of obtaining the necessary number of data points since all the data is computer generated.

Estimation of B_{ik}

(1) The value B_{ik} could be estimated by use of the correlation technique described above. They can however also be derived by an alternative method described below. It should be noted that this method can easily be adapted to the determination of the A_{ij} .

The method consists of running the model (1) with each of K input pairs

$$i^{k+} = (i'_1(t), i'_2(t), \dots, i'_k(t), +I_k, i'_m(t))$$

$$i^{k-} = (i'_1(t), i'_2(t), \dots, i'_k(t), -I_k, i'_m(t))$$

where I_k (which is constant) represents the size of a test input perturbation applied to the input i_k .

(2) If we assume that the outputs are generated by a linear model of form (3) then the values \underline{x}^{k+} and \underline{x}^{k-} corresponding to the inputs \underline{i}^{k+} and \underline{i}^{k-} are given by

$$\underline{x}_{i,n+1}^{k+} = \underline{x}_{i,n}^{k+} + DT * \sum_j A_{ij} \underline{x}_{j,n}^{k+} + \sum_{r \neq k} B_{ir} \underline{i}_{r,n}^{k+} + B_{ik} (\underline{i}_{k,n}^{k+} - I_k) + \underline{c}_i \quad (14)$$

and

$$\underline{x}_{i,n+1}^{k-} = \underline{x}_{i,n}^{k-} + DT * \sum_j A_{ij} \underline{x}_{j,n}^{k-} + \sum_{r \neq k} B_{ir} \underline{i}_{r,n}^{k-} + B_{ik} (\underline{i}_{k,n}^{k-} - I_k) + \underline{c}_i \quad (15)$$

subtraction of (15) from (14) gives

$$\underline{x}_{i,n+1}^{k+} - \underline{x}_{i,n+1}^{k-} - \underline{x}_{i,n+1}^{k+} + \underline{x}_{i,n}^{k-} + DT * \sum_j A_{ij} (\underline{x}_{j,n}^{k+} - \underline{x}_{j,n}^{k-}) = 2 * DT * B_{ik} * I_k \quad (16)$$

which once the A_{ij} have been estimated gives an estimate of B_{ik} at each step of the simulation.

The obvious approach to obtaining an estimate of B_{ik} if there are N steps in the simulation is to take

$$\hat{B}_{ik} = \frac{1}{2 * DT * I_k * N} \sum_{n=0}^N \left\{ (\underline{x}_{i,n+1}^{k+} - \underline{x}_{i,n+1}^{k-}) - (\underline{x}_{i,n}^{k+} - \underline{x}_{i,n}^{k-}) + DT * \sum_j A_{ij} (\underline{x}_{j,n}^{k-} - \underline{x}_{j,n}^{k+}) \right\} \quad (17)$$

or

$$\hat{B}_{ik} = \frac{1}{2 * DT * I_k * N} \left\{ \underline{x}_{N+1,i}^{k+} - \underline{x}_{N+1,i}^{k-} - (\underline{x}_{0,i}^{k+} - \underline{x}_{0,i}^{k-}) + DT * \sum_j A_{ij} \sum_{m=0}^N (\underline{x}_{j,m}^{k-} - \underline{x}_{j,m}^{k+}) \right\} \quad (18)$$

Estimation of \underline{c}_i

Addition of (14) and (15) gives

$$(\underline{x}_{i,n+1}^{k+} + \underline{x}_{i,n+1}^{k-}) - (\underline{x}_{i,n}^{k+} + \underline{x}_{i,n}^{k-}) - DT * \sum_j A_{ij} (\underline{x}_{j,n}^{k+} + \underline{x}_{j,n}^{k-}) - 2 * DT * \sum_{r \neq k} B_{ir} \underline{i}_{r,n}^{k+} = 2 * DT * \underline{c}_i \quad (19)$$

whence once the A_{ij} and B_{ik} have been estimated, an estimate of \underline{c}_i can be obtained,

so:

$$\hat{\underline{c}}_i = \frac{1}{2 * DT * N} \sum_{n=0}^N \left\{ (\underline{x}_{i,n+1}^{k+} + \underline{x}_{i,n+1}^{k-}) - (\underline{x}_{i,n}^{k+} + \underline{x}_{i,n}^{k-}) - DT * \left\{ \sum_j \hat{A}_{ij} \sum_{n=0}^N (\underline{x}_{j,n}^{k+} + \underline{x}_{j,n}^{k-}) - 2 * \sum_r \hat{B}_{ir} \underline{i}_{r,n}^{k+} \right\} \right\} \quad (20)$$

or

$$\hat{\underline{c}}_i = \frac{1}{2 * DT * N} \left\{ (\underline{x}_{i,N+1}^{k+} + \underline{x}_{i,N+1}^{k-}) - (\underline{x}_{i,0}^{k+} + \underline{x}_{i,0}^{k-}) - DT * \left\{ \sum_j \hat{A}_{ij} \sum_{n=0}^N (\underline{x}_{j,n}^{k+} + \underline{x}_{j,n}^{k-}) - 2 * \sum_r \hat{B}_{ir} \sum_{n=0}^N \underline{i}_{r,n}^{k+} \right\} \right\} \quad (21)$$

It should be noted that the values of the coefficients of the c_i matrix represent roughly estimates of the average degree of influence which the state variables not included in the linear model have upon the system, while the values A_{ij} represent the influence of the state variables of interest on the system. Clearly if the neglected state variables show considerable variation the c_i may be relatively poor estimates of their effect at any particular point in time.

Under these circumstances the linear model obtained may be unsatisfactory.

Similarly the B_{ik} represent the average effects of the inputs on the system.

Again if the variables that determine these effects (in general a set of nonlinear equations) show considerable variation we may again expect that the linear system will be a poor approximation.

Computational Aspects of Methods

The methods described above are easily applied where a FORTRAN version of the original DYNAMO model is available, e.g. as generated by the DYSMAP compiler. Indeed it would be a relatively simple matter to produce a version of the DYSMAP compiler that would generate the necessary estimates automatically. The facilities that are necessary are:

- (i) the ability to run the nonlinear model in its original form and again with various perturbations such as noise inputs
- (ii) the ability to run in parallel the basic nonlinear model without noise and the same model with noise terms added by dimensioning the state variables as 2 - vectors
- (iii) a method of scaling the perturbations to the state variables and inputs, e.g. as 1% of their initial value.

Example of an Application

The method described above was applied to the model below:

```

R  PROJSR.KL=0.04*RDSIZE.K
L  RDSIZE.K=RDSIZE.J+DT*RDCHGE.JK
R  RDCHGE.KL=DELAY3(DESCS.JK,RECDEL)
C  RECDEL=10
R  DESRD.KL=(DESCS.K*1.4-CS.K-POTP.K*2.5)/(12*2.5)
L  POTP.K=POTP.J+DT*(PROJSR.JK-PROJCR,JK)+DT*(DESCR.DJ-RDCHGE,JK)*.04
A  DESCS.K=TABHL(TAB,TIME,K,0,120,12)
T  TAB=170/175/186/218/237/255/285/326/365/418/497
L  CS.K=CS.J+DT*(PROJCR.JK-DECAY.JK)*2.5
R  DECAY.KL=DELAY3(PROJCR,JK,ML)

C  ML=100
R  PROJCR,KL=DELAY3(PROJCR,KL,DD)
C  DD=50
C  DT=.125
C  LENGTH=60
C  PLTPER=1
A  PRTPER.K=1
N  RDSIZE=8.5
N  CS=170
N  PROJSR=.34
N  POTP=17.2
N  PROJCR=.34
N  DESRD=.4

```

As formulated this contains 12 state variables:

RDSIZE, POTP, CS and 9 hidden state variables corresponding to the 3 DELAY3 macros. The method was used to determine a 3rd order linear model containing only the state variables RDSIZE, POTP and CS and the input DESCS corresponding to this system.

- In fact this model constituted a fairly severe test of the method since it is unstable. Any differences between the actual model and its linear approximation would therefore tend to increase over the course of the simulation. Application of the approximation method using runs to determine the A_{ij} gave the following linear model:

$$\begin{Bmatrix} \text{RDSIZE} \\ \text{POTP} \\ \text{CS} \end{Bmatrix} = \begin{Bmatrix} .0008 & .0001 & -.0001 \\ .0396 & -.0029 & -.0012 \\ .0000 & .0000 & .0000 \end{Bmatrix} \begin{Bmatrix} \text{RDSIZE} \\ \text{POTP} \\ \text{CS} \end{Bmatrix} + \begin{Bmatrix} .0121 \\ -.0028 \\ .0104 \end{Bmatrix} \text{DESCS} + \begin{Bmatrix} -1.483 \\ 0.419 \\ -1.820 \end{Bmatrix}$$

A comparison of selected values of the 3 state variables is given below. As can be seen agreement between the 2 models is good with the only significant error being of the order of 5% in POTP. In view of the unstable nature of the model this result is very satisfactory and suggests that for the stable systems encountered in the majority of business applications the techniques described should give satisfactory linear approximations.

Comparison of State Variable Values for Nonlinear Model
and Linear Approximation

	ORIGINAL MODEL	LINEAR MODEL	ORIGINAL MODEL	LINEAR MODEL	ORIGINAL MODEL	LINEAR MODEL
TIME	RDSIZE	RDSIZE	POTP	POTP	CS	CS
0.0000E 00	8,5000	8,5000	17.200	17.200	170.00	170.00
10.00	13,506	14.384	18.245	18.554	170.00	169.79
20.00	21.668	20.919	21.794	22.116	170.14	170.19
30.00	31.374	28.878	28.675	28.011	170.96	171.85
40.00	42.999	39.730	39.114	36.690	173.51	176.06
50.00	56.771	52.661	53.237	49.238	179.22	182.13
60.00	67.392	67.352	70.261	66.319	189.73	189.79

References

- McLean M. (1976) 'The Importance of Model Structure,' Futures, February, pp40 - 50
& Shepherd P.
- Rademaaker O. (1973) 'On Understanding complicated Models , Simple Methods', Technische Hogeschool, Eindhoven,
- Sharp J.A. (1974) A Study of some Problems of System Dynamics Methodology, Ph.D. thesis, University of Bradford
- Sharp J.A. (1975) The Work of the System Dynamics Research Group at the University of Bradford, O.R. Society Conference.
- Thillainathan V. (1975) Capacity Acquisition for Continuous Production Systems, DYNAMICA, I, pp 11-18.