SENSITIVITY ANALYSIS OF TECH1--A SYSTEMS DYNAMICS MODEL FOR TECHNOLOGICAL SHIFT

P. Markowich

September 1978
WP-79-78

INTERNATIONAL INSTITUTE FOR APPLIED SYSTEMS ANALYSIS
A-2361 Laxenburg, Austria

PREFACE

This paper deals with the sensitivity analysis of TECH1--a system dynamics model, which describes the technological shift from an old technology to a new one, within a specific scenario. However, its goal is not to describe the model, which was done by Robinson (1979), in this case the paper's goal is threefold:

1. To show with mathematical tools which factors are important for an invention to become an innovation, by interpreting in an economic sense the results of the performed analysis.

2. To make it possible for a broader range of people to understand system dynamics models--especially TECH1 and consequently to improve them.

3. To show what kind of mathematical analysis is useful for a class of economic models represented by differential equations.

Although TECH1 has not yet been applied to the real world, the author hopes that this paper will help to produce a better understanding of the innovation process in the real world, as well as of system dynamics models and their limits.

CONTENTS

SENSITIVITY ANALYSIS OF TECH1--A SYSTEMS DYNAMICS MODEL FOR TECHNOLOGICAL SHIFT

P. Markowich

GENERAL ASPECTS OF SENSITIVITY ANALYSIS

Why Sensitivity Analysis?

The behaviour of most models--including TECH1*--is sensitive to certain parameters, initial values, constants, or time varying functions. The values we assign to the model simulating the real world by computation on a machine, called actual values, are in most cases different from the real parameter values, called nominal. The reasons for this might be:

1. The parameter values are estimated by a statistical method from historical data, so that an error is unavoidable, for example, least-square-estimation (LSE), maximum-likelihood-estimation (MLE).
2. The data used for the computing of parameter values are imprecise. This may be because of measurement errors or round-off errors.
3. For simplicity, parameters are assumed to be of a simple form. For example, nominal-time varying parameters are assumed to be constant (the model builder takes the mean value, for example).

The second difference between the model and the real world is that the model builder assumes imprecise or simplified relationships

*The model TECH1 is still in development and a rather different version appears in Robinson (1979).

between the variables, which describe states in the real world, called state variables. An example of this is when the model builder builds a linear model, knowing that the realistic relationship is slightly nonlinear. He will be interested in the model's behaviour where the linear relationships change by a small or random-nonlinear term. He can only accept his simplified model if the behaviour does not change very much in a way which must be specified in connection with the example.

Assuming this realistic situation, it is obvious that we need a way to measure the sensitivity of a model in respect to small changes of the parameter values. Because in most cases we cannot avoid errors 1, 2, and 3, and therefore the sensitivity is only one point of view of the reliability of the model. For example, if we have a model depending on a parameter computed by LSE using imprecise data, and if we find that the model is very sensitive to small changes in this parameter, let us say an error of 1% in the parameter causes an error of 1000% in the model's state variables, then this model is worthless in describing the real world.

This was the first reason for sensitivity analysis, the second is quite different. In many cases a model will give advice to decision makers, and because of that, the model builder has to have a good overview of its behaviour in order to be able to talk with the decision makers who want to know how to influence exogenous variables (parameters) to obtain desired results such as higher profits, etc. For this qualitative and in some cases quantitative sensitivity analysis is useful.

The next question is how to perform sensitivity analysis for a given model. The simplest method is to have many runs with systematically changed parameter values. The disadvantages of this are:

1. High cost of computing time.
2. No comprehensive impression of the sensitivity-behaviour of complex multi-parameter models.
3. A lot of data, so that the model builder has to be very careful not to loose the overview.

The one advantage of this method is that it can be used for all kinds of parameter-dependent computer driven models. Another advantage of this method--compared with sensitivity analysis--is, that one can also investigate parameter changes of large amounts, which is important if the model parameters are very imprecise.

For our system dynamics model, represented by a system of nonlinear first order differential equations with given initial values, there is an efficient method of sensitivity analysis. This will be described in the next section.

Sensitivity Analysis of Differential Equation Models

Consider the following parameter dependent initial value problem

$$y' = f(t,y,\alpha_1,\ldots,\alpha_m,\beta_1(t),\ldots,\beta_1(t),\ldots,\beta_k(t)) \qquad (1)$$

$$y(0) = y_o \qquad (2)$$

where

$y = (y_1,\ldots,y_n)^T$ is the vector of state variables,

$\alpha = (\alpha_1,\ldots,\alpha_m)^T$ is the vector of constant parameters, and

$\beta(t) = (\beta_1(t),\ldots,\beta_k(t)^T$ is the vector of time varying parameters

y_o is the initial value vector

t = time.

In the terminology of the last chapter α,β are actual parameters. Further we assume that the Jacobians $\frac{\partial f}{\partial y}$, $\frac{\partial f}{\partial \alpha}$ and $\frac{\partial f}{\partial \beta}$ exist in a great enough domain around $(y,\alpha,\beta)^T$, and that they are continuous (as well as f) so that a unique solution exists for $t\varepsilon[0,T]$. For sensitivity testing with respect to α we change the value of α_i to $\alpha_i^o = \alpha_i + \Delta\alpha_i$, so we define $\tilde{\alpha}^i = \alpha + \vec{e}_i\Delta\alpha_i$, where $\vec{e}_i$ is the i-th canonical unity vector. Therefore we get the perturbed system

$$\tilde{y}' = f(t,\tilde{y},\tilde{\alpha}^i,\beta(t)) \qquad (3)$$

$$\tilde{y}(0) = y_o \qquad (4)$$

By subtracting we derive at the equation for the error $e^i = \tilde{y} - y$.

$$\dot{e}^i = \frac{\partial f}{\partial y}(t,y,\alpha,\beta(t))e^i + \frac{\partial f}{\partial \alpha_i}(t,y,\alpha,\beta(t))\cdot\Delta\alpha_i \qquad (5)$$

$$e^i(0) = 0 \qquad (6)$$

This is only valid for the first approximation because we neglected all terms in the Taylor expansion of f except the first and the second.

Performing the substitution $e^i(t) = S(\alpha_i,t,y)\cdot\Delta\alpha_i$ where we define $S_i = S(\alpha_i,t,y)$, we get:

$$\dot{S}_i = \frac{\partial f}{\partial y}(t,y,\alpha,\beta(t))S_i + \frac{\partial f}{\partial \alpha_i}(t,y,\alpha,\beta(t)) \qquad (7)$$

$$S_i(0) = 0 \qquad (8)$$

Or in matrix notation with $S(\alpha,t,y) = [S_1,\ldots,S_m]$

$$\dot{S} = \frac{\partial f}{\partial y}(t,y,\alpha,\beta(t))S + \frac{\partial f}{\partial \alpha}(t,y,\alpha,\beta(t)) \qquad (9)$$

$$S(0) = 0 \qquad (n \times m \ \text{ zero-matrix}) \qquad (10)$$

assuming that $\Delta\alpha_1 = \ldots = \Delta\alpha_m = \Delta\alpha$ and $\tilde{\alpha} = \alpha + \Delta\alpha\cdot\vec{e}$, $\vec{e} = (1,\ldots,1)^T$.

This matrix-differential equation, called a first order sensitivity equation of (1),(2) with respect to small changes in α is a linear first order differential equation which can be solved knowing the actual parameter vector α and the corresponding solution y (the parameter vector β is assumed to be fixed).

In order to get a better understanding of what the elements of the matrix S mean, let us derive a second way of reaching the sensitivity equation (9) and (10):

The solution y of the system (1) and (2) is of course dependent on α, and the theory of differential equations stresses that y is a differentiable function of α, if f is differentiable in α. That means:

$$y(t) = y(t,\alpha) \quad .$$

We can conclude: $y(t,\tilde{\alpha}) - y(t,\alpha) = \text{grad}_{\alpha} y(t,\alpha)\cdot(\tilde{\alpha} - \alpha)$. That means, that the element S_{ij} in the i-th row and j-th column of S is the partial derivative $\frac{\partial y_i}{\partial \alpha_j}(t,\alpha)$. This function is called sensitivity function of y_i with respect to α_j and it is the amplification of the error $\tilde{\alpha}^j - \alpha$ to the corresponding error in the solutions.

The next step is to consider changes in the time-varying parameter vector $\beta(t)$. So $\tilde{\beta} = \beta(t) + \Delta\beta\cdot g(t)\cdot\vec{e}$, where $\Delta\beta$ and $g(t)$ are scalars, and $\tilde{\beta}^j = \beta(t) + \Delta\beta\cdot g(t)\cdot\vec{e}$. By the same means we get the first order sensitivity equation with respect to $\beta(t)$:

$$\dot{S} = \frac{\partial f}{\partial y}(t,y,\alpha,\beta)S + \frac{\partial f}{\partial \beta}(t,y,\alpha,\beta)g(t) \tag{11}$$

$$S(0) = 0 \qquad (n \times k \quad \text{zero matrix}) \tag{12}$$

Now $S = S(\beta,t,y)$. Again the elements S_{ij} are the amplification functions of $\Delta\beta$ to the error in the solutions:

$$y_i(t,\tilde{\beta}^j(t)) - y_i(t,\beta(t)) = S_{ij}\cdot\Delta\beta \quad .$$

The equation for initial-value sensitivity is derived in a similar way:

$$\dot{S} = \frac{\partial f}{\partial y}(t,y)S \tag{13}$$

$$S(0) = I \qquad (n \times n \quad \text{unity matrix}) \tag{14}$$

The parameters α,β do not play a role here.

The last changes to be considered here are changes in the right hand side of the differential equation. We consider:

$$y' = f(t,y,g(y)) \tag{15}$$

$$y(0) = y_o \tag{16}$$

where

$$g(y) = (g_1(y),\ldots,g_e(y))$$

and

$$g_e: \mathbb{R}^n \to \mathbb{R}$$

continuously. The Jacobian $\frac{\partial f}{\partial y}$ and $\frac{\partial f}{\partial g}$ are assumed to exist and to be continous locally around y.

Now we change g(y) to $\tilde{g}(y) = g(y) + \varepsilon h(y)\vec{e}$, ($\varepsilon$ fixed). We again achieve the perturbed system:

$$y' = f(t,\tilde{y},g(\tilde{y}) + \varepsilon h(\tilde{y})\vec{e}) \tag{17}$$

$$\tilde{y}(0) = y_o \quad . \tag{18}$$

By subtracting we get $e = \tilde{y} - y$ and $e = \varepsilon.S$

$$S' = \frac{\partial f}{\partial y}(t,y,g(y))S + \frac{\partial f}{\partial g}(t,y,g(y))h(y) \tag{19}$$

$$S(0) = 0 \qquad (n \times 1 \quad \text{zero matrix}) \tag{20}$$

This is the first order sensitivity equation for structural changes. More features of sensitivity are worked out in Tomovic (1970) and Frank (1978).

Definition of a Sensitivity Measure

What have we really achieved now? We are able to compute to each state variable y_i and to each parameter value $\alpha_j, \beta_j(t)$ (initial value y_{oj} or function of state-variables $g_j(y)$), a function $S_{ij}(t)$ which relates the absolute error of the parameter value $\tilde{\alpha}_j - \alpha_j$ (initial values $\tilde{y}_{oj} - y_o$), or ε-errors in the case of time dependent parameters or function of state-variables to the absolute errors of the state-variables in the following way:

$$\tilde{y}_i - y_i = S_{ij}(\alpha)(\tilde{\alpha}_j - \alpha_j) \tag{21}$$

$$\tilde{y}_i - y_i = S_{ij}(\beta)\varepsilon \tag{22}$$

$$\tilde{y}_i - y_i = S_{ij}(y_o)(\tilde{y}_{oj} - y_{oj}) \tag{23}$$

$$\tilde{y}_i - y_i = S_{ij}(g)\varepsilon \tag{24}$$

where $\tilde{y}_i$ is the i-th component of the perturbed solution vector.

Let us now consider time invariant parameters. We compute $n \times m$ functions s_{ij} (in-time invariant parameters α_j) which does not really allow a good overview of the sensitivity behaviour of the system, and we cannot say the system is more sensitive to α_j than to α_k.

The second disadvantage is that we are not yet able to deal with percental change, because we can only look at absolute errors. In order to get rid of these disadvantages, we have to introduce norms and relative errors.

Firstly, we choose a norm $\|\cdot\|$, which expresses well the desired measurement of changes of state variables y_i to $\tilde{y}_i$, defined on a linear space, and which contains the space of continuous functions on the interval $[0,T]$, where T is the endpoint for the integration, because our solutions y_i and $\tilde{y}_i$ are continuous on $[0,T]$ and define the relative error.

$$r(y_i,\tilde{y}_i) = \frac{\|\tilde{y}_i - y_i\|}{\|y_i\|} \quad \text{(for } y_i \text{ not equivalent to the zero element of the chosen linear space)} \tag{25}$$

assuming that we know the unperturbed solution y. Now we can compute (for $\alpha_j \neq 0$):

$$\frac{\| \tilde{y}_i - y_i \|}{\| y_i \|} = \left[\frac{|\alpha_j| \cdot \| s_{ij} \|}{\| y_i \|} \right] \cdot \frac{|\alpha_j - \tilde{\alpha}_j|}{|\alpha_j|} \qquad (26)$$

We call the quantity $\frac{|\alpha_j| \; \|s_{ij}\|}{\| y_i \|} = \tilde{S}_{ij}$ relative sensitivity measure of y_i according to α_j. That means:

$$r(y_i, \tilde{y}_i) = \tilde{S}_{ij} \; r(\alpha_j, \tilde{\alpha}_j) \quad . \qquad (27)$$

In this way $\tilde{S}_{ij}$ relates the relative error of α_j to the relative error of y_i.

If we are interested in the effect of parameter changes to r groups of state variables $y_1, \ldots, y_{r_1}, \ldots, y_{r_{r-1}+1}, \ldots, y_{r_r}$ where $\sum_{i=1}^{r} r_i = n$ we can compute (r_o set to 0)

$$\sum_{l=r_i+1}^{r_{i+1}} \tilde{S}_{lj} = \sum_{l=r_i+1}^{r_{i+1}} \frac{\| s_{lj} \|}{\|y_l\|} \; |\alpha_j| =: \bar{S}_{ji} \qquad (28)$$

and

$$\sum_{l=r_i+1}^{r_{i+1}} r(y_l, \tilde{y}_l) = \bar{S}_{ji} \cdot r(\alpha_j, \tilde{\alpha}_j) \qquad i = 1(1)r$$

The quantity $\bar{S}_{ji}$ shows the effect of α_j changes to the group $(y_{r_{i-1}+1}, \ldots, y_{r_i})$.

Now we can compare influences of changes of α_j to influences of changes of α_l showing that the group $(y_{r_{i-1}+1}, \ldots, y_{r_i})$ is more sensitive with respect to α_j than to α_l if

$$\bar{S}_{ji} > \bar{S}_{li}$$

or the whole model is more sensitive to α_j than to α_r if

$$\sum_{i=1}^{n} S_{ij} > \sum_{i=1}^{n} S_{il} \quad .$$

The last definition is the most important (9): The number $\hat{S}_{ji} = \frac{1}{(|r_i - r_{i-1} - 1| + 1).T} \bar{S}_{ji}$ is called the average relative sensitivity measure of the group $(y_{r_{i-1}+1},\ldots,y_{r_i})$ in one time unit changing α_j for one percent under the scenario $\alpha,\beta(t)$. The same method is possible in the case of other parameter changes (initial values, time dependent parameters, etc.) if the parameter perturbation is constant (no function of time or of state variables). The last problem is to choose the norm. "Appropriate" is a very loose expression, but this problem can only be solved according to the actual model. For example, choose the maximum norm if you are interested in the greatest possible error, or choose the L_2-norm if you are interested in the error over the whole interval $[0,T]$.

Now let us consider an example: consider the wave equation:

$$y = \begin{bmatrix} 0 & 1 \\ -\omega^2 & 0 \end{bmatrix} y \ , \ y(o) = y_o = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \ y = \begin{pmatrix} y_1 \\ y_2 \end{pmatrix}$$

where ω is a real parameter. The solution of this equation is

$$y = \begin{pmatrix} \cos\omega t \\ -\ \omega\ \sin\omega t \end{pmatrix}$$

and the sensitivity equation:

$$\dot{S} = \begin{bmatrix} 0 & 1 \\ -\omega^2 & 0 \end{bmatrix} S + \begin{pmatrix} 0 \\ -2\omega y_1 \end{pmatrix}$$

or

$$\dot{S} = \begin{bmatrix} 0 & 1 \\ -\omega^2 & 0 \end{bmatrix} S + \begin{pmatrix} 0 \\ -2\omega \cos\omega t \end{pmatrix}$$

$$S(0) = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

We can easily obtain the solution

$$= \begin{pmatrix} -t \sin\omega t \\ -\sin\omega t - t\omega \cos\omega t \end{pmatrix}$$

We are interested in $\hat{S}_{\omega,1}$ where $r_1 = r_2 = 1$, that means that we want to know the percental change of y_1 in one time unit, changing ω for one percent, we get:

$$\hat{S}_{\omega,1} = \frac{|\omega|}{T} \frac{\| t \sin\omega t \|}{\| \cos\omega t \|}$$

For the sake of simplicity we choose $\|\cdot\| = \|\cdot\|_{\infty}^{[0,T]}$, that is the maximum norm on the interval [0,T]; and further we choose $T > \frac{\pi}{\omega}$ (so that $\cos\omega t$ reaches 1 in [0,T]). We conclude

$$\hat{S}_{\omega,1} = \frac{|\omega|}{T} \| t \sin\omega t \|_{\infty}^{[0,T]} \leq \frac{1}{T} \cdot |\omega| \cdot T = |\omega| \quad .$$

That means that y_1 does not change for more than $|\omega|$ percent in one time unit (in the average and max-norm), if ω changes for one percent.

TRANSLATION OF TECH1 INTO MATHEMATICAL TERMS

DYNAMO--A Simulation Language

TECH1 is a system dynamics model, and because of this it is necessary to translate DYNAMO-statements, which is the computer language of system dynamics models, into mathematical expressions. In the following we are always speaking about the extended DYNAMO/ NDTRAN version.

DYNAMO is a language which enables the user to solve initial value problems numerically by using either EULER, RUNGE KUTTA or ADAMS-BASHFORTH methods as determined by an integration option. The usual way is to construct the DYNAMO program having built a flow-chart, which represents the system's dynamics. We, however, take a different way. We just have a DYNAMO program which we translate into the corresponding differential equations and then we perform sensitivity analysis.

The most important types of equation in DYNAMO are the rate- and level equations. Their meaning is well demonstrated by an example:

```
r rate.kl = a.k

l level2.k = integral (rate.jk)
```

That means, that the computer is going to perform the chosen numerical procedure for the differential equation:

$$\frac{d}{dt}\,\text{level2}(t) = \dot{\text{level2}}(t) = a(t)$$

Rates are derivatives of state (level) variables which they influence. Of course a(t) may be a function of level2(t) defined by auxiliary equations, and so we get an ordinary first order differential equation. Because of uniqueness we have to assign an intial value to this equation, and this has the following form in DYNAMO:

```
n level2 = 2.5
```

In mathematical terms

level2(0) = 2.5

The next statement which has to be declared are table function statements. Consider the following sequence:

```
a atef.k = tabhl(afc,log(atco.k),0,20,2)
t afc = 1/2/5/7/9/10/10/10/10/10/3
```

That means that atef is a function of log(atco(t)) in the following called atef(log(atco(t)), which is defined by a piecewise linear function (Figure 1).

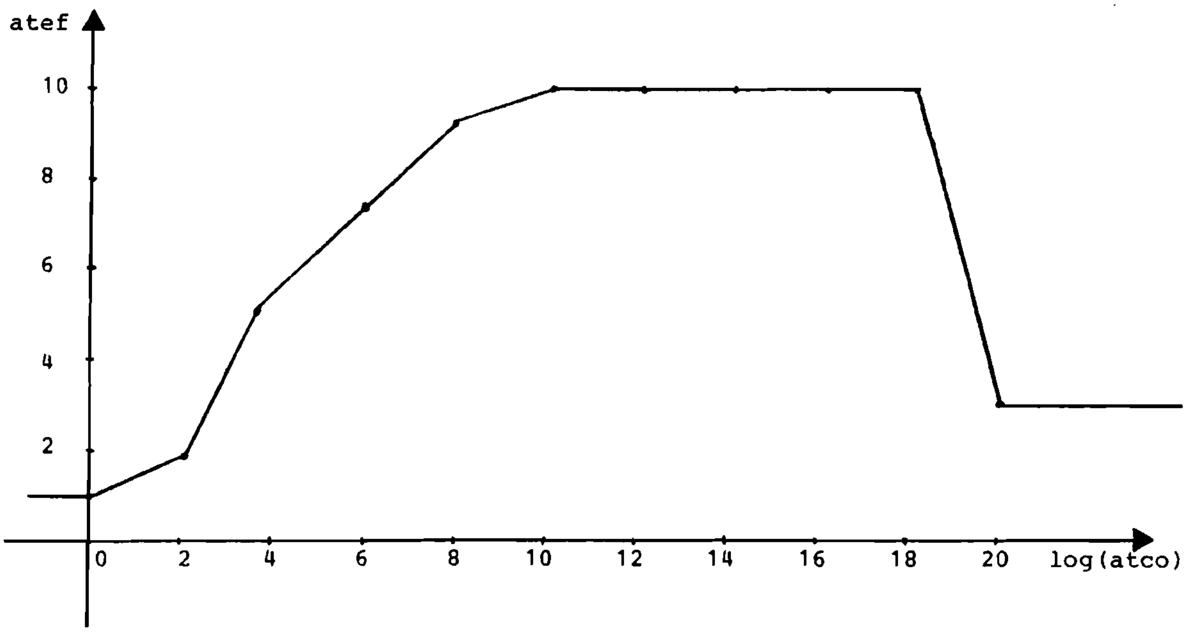


Figure 1. A table function.

Table functions are one method of making the model (differential equations) nonlinear, the others are to use standard functions, like exp, log etc., or to use ratios or product of state (level variables. In our example atco may of course be a function of levels, constants and auxiliary variables. In DYNAMO there are some more forms of table-functions, but in TECH1 only the tabhl-statement is used.

Another standard function, which has been used in TECH1 is the clip-function. Its general form is:

```
a.k = clip(value1.k,value2.k,control.k,test.k) .
```

This means that the value1.k (value2.k) is assigned to a.k if control.k is greater or equal to test.k (if control.k is smaller than test.k). Of course some arguments of the clip-function may be constants or functions of state (level) variables.

Last but not least the delay macros have to be declared. In TECH1 DELAY as well as DLINF macros of order one and three are used. The call of the DELAY1 (first order) has the following form:

```
expnd delay1(out1,in1,del,init1) .
```

In mathematical terms you can value:

$$\dot{out1}(t) = \frac{1}{del}\,(in1(t) - out1(t))$$

$$out1(0) = \frac{init1}{del}$$

The DELAY1 macro gives you the function out1(t) (solution of the first order initial value problem). The DELAY3(out3,in,del.init) For the task of solving

$$\dot{out1}(t) = \frac{3}{del}\,(in1(t) - out1(t)),\quad out1(0) = \frac{init}{\left(\frac{del}{3}\right)}$$

$$\dot{out2}(t) = \frac{3}{del}\,(out1(t) - out2(t)),\quad out2(0) = \frac{init}{\left(\frac{del}{3}\right)}$$

$$\dot{out3}(t) = \frac{3}{del}\,(out2(t) - out3(t)),\; out3(0) = \frac{init}{\left(\frac{del}{3}\right)}$$

for the function out3(t).

The DLINF1 and DLINF3 macros represent the same mathematical equations, only the initial values are changed.

In DLINF1: out1(0) = init1

In DLINF3: out1(0) = init
out2(0) = init
out3(0) = init.

Readers more interested in DYNAMO are advised to look at W.I. Davisson and J. Uhran Jr. (1977).*

TECH1--A Differential Equation Model

Looking at the DYNAMO program for TECH1, our model for a technological shift from an old to a new technology, we notice that it consists of three parts (see Appendix A):

1. The part describing the behaviour of the old technology.
2. The new technology's market share (in terms of sales).
3. The new technology's behaviour.

Part 2 links part 1 and part 3. The levels defined in part 1 are:

ot: old technology production capacity in real units

otk: old technology monetary capital in monetary units

otco: old technology cumulative output

ioto: old technology inventory output.

In Delay macros the following levels are created:

esoto: sales expectation formation time of old technology

otli: old technology's last investment

otim: old technology investment maturation.

*An updated version is in preparation.

The levels in part 2 are:

fmnt: new technology market share in percentage.

Levels in part 3 are the same as in part 1 but concerned with new technology.

The parameters and table functions used are particularly described in Robinson (1979).

We can now define the vector of states:

$$y = (ot, otk, otco, ioto, y_5, y_6, otli, esoto, y_9, y_{10}, otim)^T \qquad (29)$$

$$z = fmnt \qquad (30)$$

$$x = (nt, ntk, ntco, into, x_5, x_6, ntli, esnto, x_9, x_{10}, ntim)^T \qquad (31)$$

where

y_5, y_6, y_9, y_{10}; x_5, x_6, x_9, x_{10} are auxiliary state level variables defined in delay-macros.

The whole vector of states:

$$u = (y^T, z, x^T)^T \qquad (32)$$

(x^T is the transposed vector to x).

Further we define:

$$\begin{aligned}\bar{y} = {} & pmdot(ot\$(\frac{y_4}{y_8}))(1 - z) \cdot mkt(t) \cdot ot\$(\frac{y_4}{y_8}) \\ & - (y_1 \cdot otfc + y_1 \cdot otef(ln y_3) \cdot otpo \cdot otcu(\frac{y_4}{y_8}) \cdot otvc(t) \\ & + mdc \cdot y_2) \quad . \end{aligned} \qquad (33)$$

$$\begin{aligned}\bar{x} = {} & pmdnt(nt\$(\frac{x_4}{x_8})) \cdot z \cdot mkt(t) \cdot nt\$(\frac{x_4}{x_8}) \\ & -(x_1 \cdot ntfc + x_1 \cdot ntef(ln x_3) \cdot ntpo \cdot ntcu(\frac{x_4}{x_8}) \cdot ntvc(t) \\ & + mdc \cdot x_2) \quad , \end{aligned} \qquad (34)$$

in the DYNAMO program (Appendix A) $\bar{y}$ is otre and $\bar{x}$ is ntre.
With these definitions we get:

$$\dot{y}_1 = y_{11} - \frac{y_1}{alt} \qquad y_1(0) = otz$$

$$\dot{y}_2 = \bar{y}\cdot pfiot\left(\frac{\bar{y} - y_7}{ear.y_2}\right) - mdc\cdot y_2 \qquad y_2(0) = otkz$$

$$\dot{y}_3 = y_1\cdot otef(lny_3)\cdot otpo\cdot otcu\left(\frac{y_4}{y_8}\right) \qquad y_3(0) = otcoz$$

$$\dot{y}_4 = y_1\cdot otef(lny_3)\cdot otpo\cdot otcu\left(\frac{y_4}{y_8}\right) - pmdot\left(ot\$\left(\frac{y_4}{y_8}\right)\right)\cdot \cdot(1 - z)\cdot mkt(t) \qquad y_4(0) = iotoz$$

$$\dot{y}_5 = \frac{3}{ii}\left(- y_5 + \bar{y}\cdot pfiot\left(\frac{\bar{y} - y_7}{ear\cdot y_2}\right)\right) \qquad y_5(0) = otiz$$

$$\dot{y}_6 = \frac{3}{ii}(- y_6 + y_5) \qquad y_6(0) = otiz$$

$$\dot{y}_7 = \frac{3}{ii}(- y_7 + y_6) \qquad y_7(0) = otiz$$

$$\dot{y}_8 = \frac{1}{seft}\left(y_8 - pmdot\left(ot\$\left(\frac{y_4}{y_8}\right)\right)(1 - z)\cdot mkt(t)\right) \qquad y_8(0) = sez$$

$$\dot{y}_9 = \frac{3}{otid}\left(- y_9 + \bar{y}\cdot pfiot\left(\frac{\bar{y} - y_7}{ear\cdot y_2}\right)\right) \qquad y_9(0) = \frac{otiz}{3}$$

$$\dot{y}_{10} = \frac{3}{otid}(- y_{10} + y_9) \qquad y_{10}(0) = \frac{otiz}{3}$$

$$\dot{y}_{11} = \frac{3}{otid}(- y_{11} + y_{10}) \qquad y_{11}(0) = \frac{otiz}{3}$$

$$\dot{z} = \left(afms(t) + pfms\left(\frac{nt\$\left(\frac{x_4}{x_8}\right)}{ot\$\left(\frac{y_4}{y_8}\right)}\right)\right) clip\left(z, 1 - z, \frac{nt\$(t)}{ot\$(t)}, 1\right)$$

$$z(0) = fmntz$$

$$\dot{x}_1 = x_{11} - \frac{x_1}{alt} \qquad x_1(0) = ntz$$

$$\dot{x}_2 = \bar{x}\cdot pfint\left(\frac{\bar{x} - x_7}{earx_2}\right) upmi(lnx_3) - mdc\cdot x_2 \qquad x_2(0) = ntkz$$

$$\dot{x}_3 = x_1 \cdot \text{ntef}(\ln x_3) \cdot \text{ntpo} \cdot \text{ntcu}(\frac{x_4}{x_8}) \qquad x_3(0) = \text{ntcoz}$$

$$\dot{x}_4 = x_1 \cdot \text{ntef}(\ln x_3) \cdot \text{ntpo} \cdot \text{ntcu}(\frac{x_4}{x_8}) - \text{pmdnt}(\text{nt\$}(\frac{x_4}{x_8}))z \cdot \text{mkt}(t)$$

$$x_4(0) = \text{intoz}$$

$$\dot{x}_5 = \frac{3}{\text{ii}}(-x_5 + \bar{x} \cdot \text{pfint}(\frac{\bar{x} - x_7}{\text{ear} \cdot x_2}) \cdot \text{upmi}(\ln x_3)) \qquad x_5(0) = \text{ntiz}$$

$$\dot{x}_6 = \frac{3}{\text{ii}}(-x_6 + x_5) \qquad x_6(0) = \text{ntiz}$$

$$\dot{x}_7 = \frac{3}{\text{ii}}(-x_7 + x_6) \qquad x_7(0) = \text{ntiz}$$

$$\dot{x}_8 = -\frac{1}{\text{seft}}(x_8 - \text{pmdnt}(\text{nt\$}(\frac{x_4}{x_8}))z \, .\text{mkt}(t)) \qquad x_8(0) = \text{ntsez}$$

$$\dot{x}_9 = \frac{3}{\text{ntid}}(-x_9 + \bar{x}.\text{pfint}(\frac{\bar{x} - x_7}{\text{ear} \cdot x_2})\text{upmi}(\ln x_3)) \qquad x_9(0) = \frac{\text{ntiz}}{3}$$

$$\dot{x}_{10} = \frac{3}{\text{ntid}}(-x_{10} + x_9) \qquad x_{10}(0) = \frac{\text{ntiz}}{3}$$

$$\dot{x}_{11} = \frac{3}{\text{ntid}}(-x_{11} + x_{10}) \qquad x_{11}(0) = \frac{\text{ntiz}}{3}$$

We have achieved a system of ordinary differential equations with given initial values which we can write as:

$$\begin{aligned} \dot{u} &= g(u,t) \\ u(0) &= u_o \end{aligned} \qquad (35)$$

where

$$u_o = (y^T(0), z(0), x^T(0))^T$$

and

$$g: \mathbb{R}^{23} \times \mathbb{R} \to \mathbb{R}^{23} \quad .$$

It is obvious that it is impossible to obtain an analytic solution of a 23-dimensional nonlinear system of differential equations with given initial values, so sensitivity analysis and testing are the best ways of getting information about the behaviour of the solutions.

Application of the Sensitivity Theory to TECH1

We have represented TECH1 by the initial value problem (35)

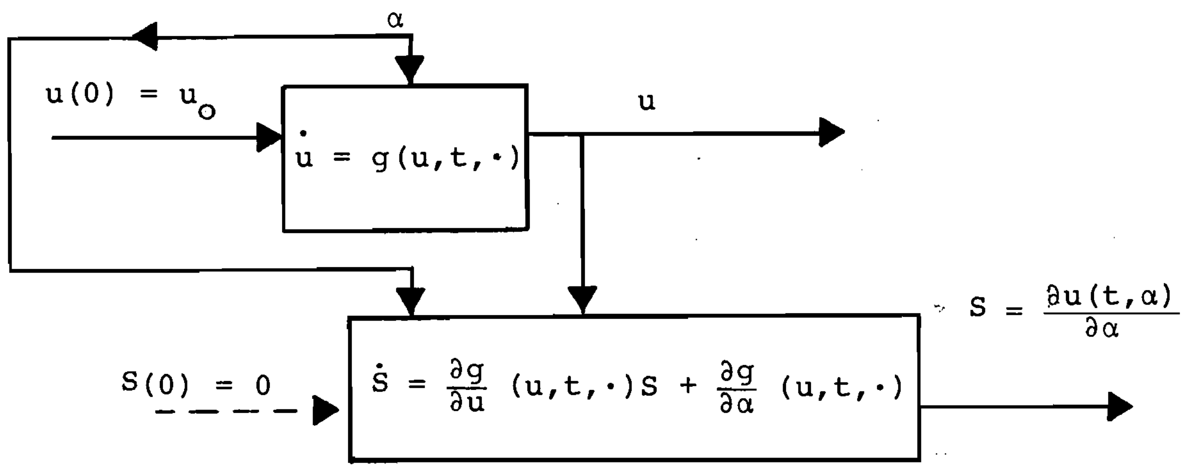
$$\dot{u} = g(u,t) \qquad 0 \leq t \leq T$$

$$u(0) = u_o$$

where g in the sense of sensitivity analysis is also a function of the parameters we are interested in. Let α be such a parameter, and we therefore have to compute the solution of the combined system

$$\begin{aligned} \dot{u} &= g(u,t,\alpha) \\ \dot{S} &= \frac{\partial g}{\partial u}(u,t,\alpha)S + \frac{\partial g}{\partial \alpha}(u,t,\alpha) \qquad 0 \leq t \leq T \\ u(0) &= u_o \\ S(0) &= 0 \end{aligned} \qquad (36)$$

The structure of this problem can be shown in the following way:



The first problems we are faced with are to compute $\frac{\partial g}{\partial u}$ and $\frac{\partial u}{\partial \alpha}$. Describing the DYNAMO-language we saw that there are a table function and a clip function which are not differentiable because they are a linear interpolation function and a jump-function respectively. However, we can solve these problems in the following way: in the case of table functions we compute the derivative using the formula

$$f'(x) \approx \frac{f(x + h) - f(x)}{h}$$

for small h, for example, consider the table function

```
a pmdot.k = tabhl(pmdc,ot$.k,0,3,.5) .
```

We compute the derivative $\frac{d(pmdot)}{d(ot\$)}$ called dpmdot in the DYNAMO-program:

```
c h = 0.001
a hpmdot.k = tabhl(pmdc,h + ot$.k,0,3,.5)
a dpmdot.k = (hpmdot.k - pmdot.k)/h   .
```

That means that we have smoothed the function on the "dangerous" points. Let us now show the procedure with clip-functions using an example:

```
a sg.k = clip(fmnt.k,(1 - fmnt.k),nt$.k/ot$.k,1)
```

We compute $\frac{d(sg)}{d(fmnt)}$ called dsg:

```
a dsg.k = clip(1, - 1,nt$.k/ot$.k,1)
```

assuming that the clip-function is an approximation for a smooth function we compute the "derivative" of the clip-function as an approximation to the derivative of the smooth function.

In the DYNAMO program the sensibility vector S is named by the following format

$$\frac{d(\text{level})}{d\alpha} = \text{slevel}$$

The parameter α is specified by the additive term $\frac{\partial g}{\partial \alpha}$. The Jacobians are not listed here because they can be reconstructed easily from the DYNAMO-program in Appendix A.

RESULTS AND ECONOMIC CONCLUSIONS

Specification of Investigated Parameters

The analyzed parameters are shown in Table 1.

Table 1. Parameters characterising the two technologies, in decreasing importance.

Parameter	Variable name*	Character-istics**
Attractiveness factor	afms	tv
New tech's market share initial value	fmntz	c
Potential capital/output ratio	.tpo	c
Learning curve, relationship between experience and technical efficiency	.tef	lf
Variable costs	.tvc	tv
Fixed costs	.tfc	c
Novelty factor in investment	upmi	lf
Economywide average return	ear	c
Market size	mkt	tv
Monetary depreciation	mdc	c
Physical depreciation	alt	c

*In the variable name column "." is a dummy letter which would be replaced by "n" for the new technology and by "o" for the old.

**In the characteristic column "c" means constant parameter, "tv" means time varying (function of time only) parameter and "lf" means level-function (function of level variables).

The purpose of the following paragraph is to show the results of the sensitivity analysis, and as mentioned in previous chapters, because of the nonlinearity of our system of differential equations, this is only possible locally, that means with regard to the computed vector of state (level) variables, because the sensitivity equations depend on this solution vector. However, having performed some runs it becomes noticeable that it is necessary to look at two different cases:

1. The success case, which is determined by the fact that the new technology reaches--in the observed time period--a high enough market share to compete with the old one and afterwards to substitute it.

2. The failure case. The market share of the new technology remains too small to give the possibility of a substitution process.

Within these two cases the behaviour of the sensitivity curves remains rather constant, and therefore it is sufficient for understanding the system to look at some simulation runs belonging to case 1 and 2.

Interpretation of the Results

In both cases the results will be shown using the same data (parameter) configuration (see Appendix A), except for the parameter ntpo: We chose ntpo = 3.0 to gain a success run, and on the other hand we chose ntpo = 0.667 to gain a failure run. The observed time interval is in all cases [0,40] (in years). It is further obvious that the computed figures should not be accepted too strictly because of round-off errors, procedure errors, and in some cases because of the bad condition of the sensitivity equations. Thereby we notice a disadvantage of DYNAMO: there is no procedure available for integrating "stiff" (badly conditioned) differential equations, all available procedures--Euler, Runge-Kutta of fourth order and Adams-Bashforth of fourth order are explicit, so it would be very useful to introduce an implicit method, for example an implicit Euler method into the DYNAMO-system. However, despite this disadvantage the computed numbers are reliable in their meaning because several simulation runs with the same data configuration were done with different integration methods and different step-sizes before the result was accepted.

Before the chapter concerning the translation of TECH1 into mathematical terms we introduced the concept of the relative sensitivity measure of a parameter to a group of state variables, for one time unit. For our purposes it would seem to be useful to look at three groups:

Group 1: State variables of old tech.
ot, otk, otco, ioto

Group 2: Market share of new tech.
fmnt

Group 3: State variables of new tech.

These relative sensitivity measures are called seno (group 1), senf (group 2) and sen (group 3), and the regarded parameter must be specified. The norm is the $L_2[0,T]$-norm. This is a norm for square-summable functions on the time interval [0,T] (our solutions are continuous, and therefore square-summable) which is defined by:

$$\| f \|_2 := \left(\int_0^T f^2(t)dt \right)^{1/2} .$$

The advantage of this norm is that it does not measure the function only in some time points (as the maximum norm), but it measures the development of the state variable over the whole time interval, which is quite useful for our purposes because we want to know the "history" of our sensitivity functions over the whole time period.

The first parameters we shall investigate are those which the system is very sensitive to:

<u>Parameter fmntz:</u> initial value of new tech's market share

<u>Parameter afms:</u> new tech's attractiveness (or market efficiency).

Let us start the discussion with fmntz performing a failure run, as this is the most interesting case here. Old tech's sensitivity functions are mostly negative and new tech's including sfmnt are positive and sfmnt is an increasing function of t in [0,40]. The max-norm of sfmnt is 14327.0. This indicates that the sensitivity analysis with regard to fmntz is only valid for very small changes of fmntz and that these small changes have a great influence on new tech's market share and further it indicates that an <u>increased initial value for fmnt increases new tech's market share, new tech's production</u> and so on very much. In this context the relative sensitivity function $\frac{sfmnt}{fmnt}$ is more interesting and it is about 128000.0 in the max-norm. Therefore we see that a slightly better starting position of new tech's market share gives a much better success for new technology in the future and considerably damages old tech's market share. The relative sensitivity measures are:

seno	senf	sen
0.025	0.5	0.39

These numbers seem to be rather small, but the reason for this is that fmntz was chosen to be 1.3×10^{-5} so that a one percent change of fmntz is of the absolute amount 1.3×10^{-7}, and because of that the numbers which relate the absolute error of fmntz to the relative error of the state variables, may show the sensitivity of the system to fmntz better:

$\overline{seno}$	$\overline{senf}$	$\overline{sen}$
192.1	3124.0	3063.0

Doing a success run almost nothing in the qualitative and quantitative behaviour of the sensitivity functions changes. The sign situation is still the same, but one thing is different: sfmnt is no longer an increasing function of t, it reaches its maximum at t = 20 and then it decreases, which means that the starting position of a new successful technology is more important in the short term than in the long.

All these things show us that the starting position of new tech (regarding the market share in term of sales) determines the development of new tech greatly.

We shall now discuss the system's sensitivity behaviour with regard to afms--new tech's attractiveness: In Robinson (1979) we can see that the increase of afms from constant 0.0 to constant 0.08 makes a failure run into a success run. This fact makes us think that the system is very sensitive to small (0.08) changes of afms. But we shall see what we could work out by sensitivity analysis. Firstly, we consider the success case; the sensitivity functions for old tech are mostly negative, those for new tech (including those for new tech's market share) are mostly postive. In relative terms the impact for changes of afms is approximately equal for new tech and old tech, and it differs only in sign. The maximum norm of the relative sensitivity function of new tech's market share is approximately 240.0. That means that an absolute change of afms by 0.01 causes a 2.4% change in new tech's market share. Relative sensitivity measures cannot be computed because afms is chosen to be constant 0 in the whole interval. The thing we are most interested in, is how increases of afms influence new tech's market share in a failure run.

The sign-situation is now the following: old tech and new tech's sensitivity functions oscillate in value and sign, but those for old tech are negative and those for new tech are positive over the greatest part of the time period. The max-norms of all sensitivity functions vary from 6.8×10^6 to 4.2×10^8 and especially the max-norm of sfmnt is approximately 10^4. The relative sensitivity functions for old tech have smaller max-norms than those for new tech. The max-norm of the relative sensitivity function for new tech's market share is approximately 1.8×10^8. The (changed) sensitivity measures are:

$\overline{seno}$	$\overline{senf}$	$\overline{sen}$
270.5	8614000	77690

Here we have found the reason why small changes of afms greatly alter the values and the shape of the state variables.

All that means that increase of market efficiency increase new tech's market share very much over the whole time period. These great increases indicate (as observed by testing) that a small percential change of market efficiency can make the new technology penetrate the market very well.

On the other hand, this analysis, especially the analysis of the sensitivity, with regard to fmntz show that the first part of the diffusion process is the hardest. The new technology has to attract investment, customers and has to keep its price low in order to remain competitive and its market efficiency must be increased (by a learning process).

The next parameters we turn to are:

<u>Parameter otpo</u>: old technology's potential output, and

<u>Parameter ntpo</u>: new technology's potential output.

The first thing that should be considered here is that a change of ntpo from 0.667 to 3.0 turns a failure run into a success run, using the data in Appendix A. Of course, this fact cannot be observed by a sensitivity analysis because the change from 0.667 to 3.0 is too large. However, we can figure out some other aspects. Let us first consider changes of otpo in the success case. All sensitivity functions of old tech are positive over the greatest time period, and all sensitivity functions of new tech including sfmnt are negative over the whole period. The relative sensitivity functions of new tech have a greater maximum norm than those of old tech (for example the max-norm of sntco is about 8 times as big as that of sotco). This means that increases of otpo have more negative influences on the potential output of new technology than they have positive influences on old tech's potential output. The relative sensitivity measures (with respect to otpo) are

seno	senf	sen
0.0291	0.058	0.0708

This says that the relative influence in the L_2-norm of otpo to the old tech is smaller than to the new.

Altogether, we found out that increases of otpo damage more of new tech than they are of use to old tech. Now let us remain in the success case and look at ntpo. The sensitivity functions for old tech as well as for fmnt and new tech are changing in sign (oscillatory) and the relative function for new tech's market share has a max-norm of about 32.0. The measures are:

seno	senf	sen
0.0141	0.486	0.429

Increases in otpo damage new tech slightly more than they are of use to old tech.

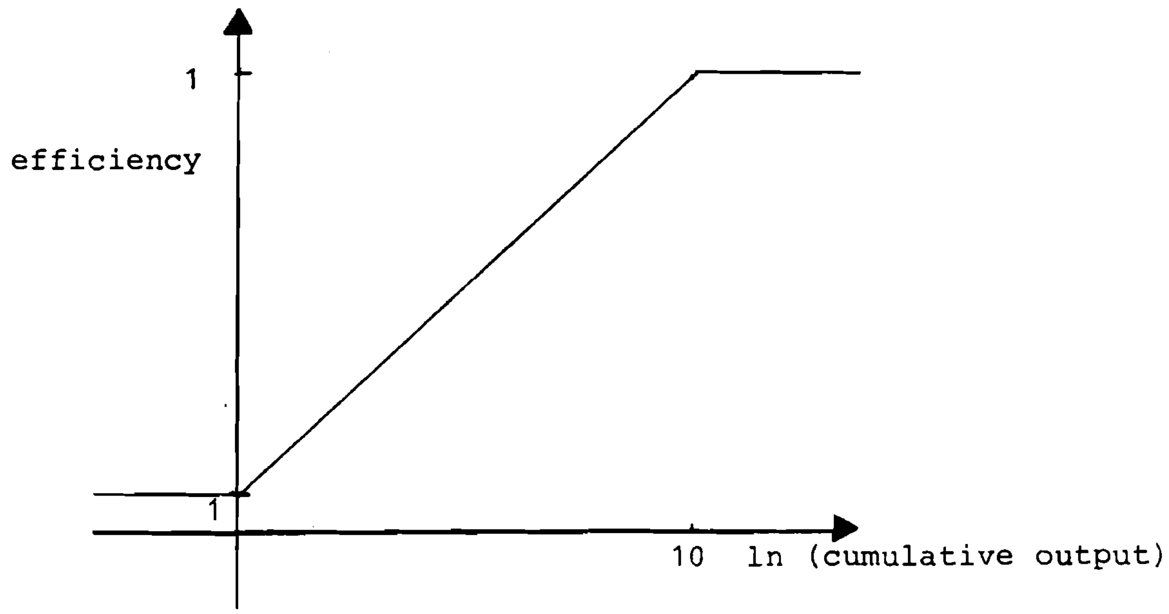
Let us now consider ntpo. The sensitivity functions for old tech are strictly negative, and those for new tech and its market share are stictly positive. Increases of ntpo are useful for the development of new tech and decreases the success of the old technology. The max-norm of the relative sensitivity function referring to new tech's market share is about 28281.0 (but consider fmnt is very small in this run). The relative measures are:

seno	senf	sen
0.727	73.47	4.87

These numbers show (especially senf) that ntpo is very important for the development of the market share of new technology, the model is very sensitive to small changes of ntpo, because a one percent increase of ntpo (in a failure run) gives a 73 percent increase in new tech's market share.

Parameter otef: learning curve for old tech
Parameter ntef: learning curve for new tech

Learning curves are structural relationships between the cumulative output of a technology and the efficiency (for further information see Devendra Sahal 1978). In TECH1 the following shape of curve was assumed:



What features does the sensitivity analysis show with respect to the learning curves? Performing a success run it shows that the sensitivity functions of old tech are postive and those for new tech, including new tech's market share, are negative. This is when the sensitivity analysis is performed with respect to otef. It is realistic. Increases of the efficiency of old tech should cause increases of old tech's market share and therefore decreases of new tech's market share.

The max-norm of the relative sensitivity functions for new tech is greater than for old tech's relative sensitivity functions. Sensitivity measures having the same meaning then for the parameter upmi were computed

$\overline{\text{seno}}$	$\overline{\text{senf}}$	$\overline{\text{sen}}$
0.021	0.09	0.11

The influence of absolute changes of otef to new tech is greater than to old tech.

Now we look at ntef still remaining in the success case. The first thing we notice is, that in contrast to otef all sensitivity functions for old tech are negative, and positive for new tech (including sfmnt) that means that increases of ntef favour (of course!) new technology and help to slow down old technology. The max-norm for new tech's relative sensitivity functions is greater than for old tech's (compare with otef).

The sensitivity measures are:

$\overline{\text{seno}}$	$\overline{\text{senf}}$	$\overline{\text{sen}}$
0.178	0.8	1.04

Absolute changes of ntef have greater relative influence to new technology than to the old one, and the influence of ntef to old, new tech and new tech's market share is, performing a success run, about 10 times larger than that of otef.

What about the failure case? Looking at the sensitivity functions regarding otef we see that they are all oscillatory in value and in sign. But these for new technology are positive over a greater period of time than those for new technology.

sfmnt(t) is negative in the interval [14,40]. The relative sensitivity functions for new technology are much greater than those for old tech (in the max-norm) and the sensitivity measures are:

$\overline{seno}$	$\overline{senf}$	$\overline{sen}$
0.282	8871	84.38

Maybe that the great number for $\overline{senf}$ does not give the right impression of otef's impact on new tech's market share, because it gives the percentage change of fmnt (in the L_2-norm), changing otef for 1.0 in absolute value, and in that case fmnt is very small.

Looking at the sensitivity functions of ntef we see that they are all oscillatory in sign and the relative sensitivity functions for new tech have much greater maximum norms than those of old tech. The sensitivity measures for new tech are much bigger than those for old tech. Altogether we came to the conclusion that one reason for the failure of new technology is an insufficient learning process.

The next parameters we look at are:

<u>Parameter otfc</u>: fixed costs of old tech

<u>Parameter ntfc</u>: fixed costs of new tech

<u>Parameter otvc</u>: variables costs of old tech

<u>Parameter ntvc</u>: variable costs of new tech.

Having a success run it shows (realistically) that increasing otfc and otvc has a negative influence on old technology and favours new technology (including new tech's market share) also increasing ntfc and ntvc has the opposite influence on both technologies. In this case the system is much more sensitive to the variable costs than to the fixed costs (negotiating the relative sensitivity measures). For example:

	seno	senf	sen
otfc	0.015	0.031	0.038
otvc	0.0611	0.1287	0.156

Small changes of otfc and otvc influence old technology less than new technology and changes of ntfc and ntvc influence new tech also more than old technology (consider we are speaking about the success case!) regarding the relative sensitivity measures.

The results in the failure case are quite different. Let us first consider the parameter otvc. The most important factor is, that sfmnt, which is the absolute sensitivity function of fmnt with respect to otvc is positive over the greatest part of the time period [0,40], but mostly very small in absolute value. This means that small changes of otvc (increases) cannot turn a failure run into a success run, they can only help to gain a slightly greater market share. The other state variables are varying in signs. In terms of the relative sensitivity measure for fmnt:

$$\text{senf} = 2311.8$$

which could make one assume that fmnt is very sensitive to changes of otvc, but the real reason for this is that fmnt is, in this case, very small (10^{-4} in the max-norm). seno and sen are rather small.

A further important observation is that in the failure case also, the system is more sensitive to small changes of variable costs than to fixed costs, but also none of them can make a failure run into a real success run.

We shall now consider the

Parameter upmi: novelty factor of investment

The factor upmi is chosen in TECH1 as a non-increasing function of new tech's cumulative output. It shows that performing a success run increases of upmi cause increases in new tech's market share and that they cause decreases in old tech's state variables except in old tech's inventory output. The sensitivity function belonging to old tech's inventory output is oscillatory in value and in sign too. The relative sensitivity functions are in absolute value less than 1.0 (except sfmnt which is less than 3.0). therefore changes of upmi have only a small impact on all state variables performing a success run. This situation changes completely when we look at a failure run. All sensitivity functions except the functions referring to new tech's market share are oscillatory in value and in sign. This exception remains positive over the greater part of the observed time period, but takes its maximum (approximately -3.8) while negative. Sensitivity measures which relate the average relative error of state variables in one time unit to the absolute error of the parameter function (which is assumed to be constant) were computed.

$\overline{\text{seno}}$	$\overline{\text{senf}}$	$\overline{\text{sen}}$
0.00402	128.1	1.21

Changes of upmi influence new tech more than they influence old tech, which is very realistic. Incidentally, it seems useful to mention here that parameters which are only involved with either old or new technology can influence the other--new or old technology--only by the market share of new tech, because it is the only connection between the two parts of the model.

The next parameters to be discussed are

Parameter ear: economywide average return
Parameter mkt: market size

Peforming a success run, the sensitivity functions regarding ear have the following features:

sot, sofk, sotco:	oscillatory in value and sign
sioto:	negative
snt, sntco, sinto, sfmnt:	negative
sntk:	oscillatory in value and sign.

The max-norms of the relative sensitivity functions for new tech are greater than those for old tech. The max-norm of $\frac{\text{sfmnt}}{\text{fmnt}}$ is approximately 3.5. The relative sensitivity measures are:

seno	senf	sen
7.0×10^{-4}	11240.0×10^{-7}	2.1×10^{-3}

The relative impact of ear to old and new tech is about the same. In a failure run the sensitivity functions for old tech are oscillatory in sign, sioto is negative, and the sensitivity functions for new tech including sfmnt are positive in the interval [14,40]. Comparing these results with the success run we see that increases of ear favour that technology which is in the worse situation (this is rather realistic!). The relative sensitivity functions for new tech have again greater max-norm than those for new tech and the relative sensitivity measures are:

seno	senf	sen
0.0011	0.136	0.1124

ear's relative impact to new tech is greater than that on old tech.

Let us now discuss the system's senstivity to changes of mkt(t). In a success run old tech's sensitivity functions are oscillatory in sign and new tech's are positive. All relative sensitivity functions are less than 10^{-1} in the max-norm and the relative sensitivity measures are

seno	senf	sen
0.23	0.5	0.93

Therefore the whole system is not very sensitive to mkt in a success run.

In the failure case the situation does not change very much except the sign-situation. Every sensitivity function is positive over the greatest part of the time period [0,40]. We see that in both cases increases of mkt favour the new technology (only slightly!).

Now let us continue the discussion with

Parameter alt: physical depreciation factor

and

Parameter mdc: monetary depreciation factor.

These two parameters are chosen equally for both technologies, but looking at the program in Appendix A we notice that alt is in the denominator of the term for real depreciation, and mdc is multiplied with otk in order to give monetary depreciation. That means that we have to compare increasing values of alt with decreasing values of mdc and vice versa. Increasing alt forces the real depreciation to decrease and increasing mdc forces the monetary depreciation to increase.

The computed (absolute) sensitivity functions with respect to alt showed the following properties:

1. In the successful case all sensitivity functions except sotk = $\frac{\partial otk}{\partial alt}$ are greater 0 over the greatest part of the time interval. That means that all state variables except otk (= old tech's capital) increase if the real depreciation decreases. The relative sensitivity functions $\frac{slevel}{level}$ are very small in the max norm (10^{-4} to 10^{-2}) and

seno	senf	sen
8.7×10^{-3}	5.9×10^{-3}	11.3×10^{-3}

   Every group is approximately equally influenced by changes of alt and this influence is not very strong.

2. In the failure case the same behaviour of absolute sensitivity functions referring to old tech can be watched, but the sign of the sensitivity functions for new tech changes. The size of old tech's relative sensitivity function is less than 10^{-1} and of new tech is less than 8.0. The market share of new technology is almost not influenced. The relative sensitivity measures show other sizes than those in the success case.

seno	senf	sen
7.1×10^{-3}	2.2	2.0

   (The high value of senf comes from the very small scale of fmnt!). Changes of alt influence the new technology about 200 times more than they influence the old technology.

What does this all mean for the real world? Firstly, it means that the physical depreciation factor has no strong impact on the behaviour of new and old technology. Changing it for a small amount can never make the new technology succeed if the common "climate" is against this success even if it means that in the failure case a decrease of physical depreciation must not result in a better situation for the new technology over the whole time period, although the "relative" impact of these small changes to the new technology is high (because the state variables of new technology are rather small).

What does one say now about mdc--the monetary depreciation factor?

1. In the success case the sensitivity functions for old technology are changing in sign, only sotk (capital of old tech) remains negative over the whole time period. All sensitivity functions for new tech and for the market share of new tech are less than 0 over the whole period of observation. The greatest maximum-norm of old tech's relative sensitivity function has sotk = $\frac{\partial otk}{\partial mdc}$ which is about 30.275. The others, for new tech as well, remain in the max-norm less than 6.0. This is rather realistic, because monetary depreciation must have its greatest impact on capital. It is also realistic that increasing monetary depreciation depresses the growth, or at least does not favour it.

   The sensitivity measures are

seno	senf	sen
0.005491	0.003902	0.008527

   mdc's impact on old technology and to the market share is almost equal, the impact on new tech is greater (maybe because new tech is favoured?)

2. In the failure case sotk is strictly negative and (in absolute value) increasing, sfmnt is positive and decreasing, all other sensitivity functions are oscillatory (in sign too). This fact also stresses that monetary depreciation hits the favoured technology more than that in a worse situation. The relative sensitivity functions for new tech have greater maximum norms than those for the old technology and the measures are:

seno	senf	sen
0.00587	0.521	0.4246

The relative impact to old tech is much smaller than to new tech, which is comparable to the behaviour of the impact of alt.

Looking at the results, we can give a proposal for the improvement of the whole model: in order to get better flexibility it would be of use to introduce different parameters for old tech and for new tech (in the case of alt).

CONCLUSIONS AND FINAL REMARKS

Having discussed all the parameters analyzed we can now draw the following conslusions. The model TECH1 is, especially under failure conditions for new technology, very sensitive to all parameters which influence the equation for new tech's market share. Its sensitivity to other investigated parameters is small compared with this fact in relative as well as in absolute terms, and because of this the question is again raised (Robinson 1979) as to whether the attractiveness is (as assumed in the model) an exogenous parameter. it is obvious that the parameter "attractiveness" must be investigated further in the real world as well as in this and in other models. But assuming the model's situations, a good aid for the decision maker is to use all possible tools for increasing new tech's attractiveness in order to gain a success for the recently established technology.

Of course a great problem is that attractiveness is not very well measured because it is influenced by different aspects like marketing by the managers of the new technology (advertisement and so on) and by personal taste and preference of the consumers. Naturally it depends strongly on the kind of the innovation we are faced with. Personal taste will play a much greater role in the case of an improvement innovation than in the case of a basic innovation* (for the explanation of these terms see H.D. Haustein and H. Maier 1979). Altogether this analysis shows that the initial phase of the diffusion process determines the success or failure of a new technology for a great deal, because it is more difficult to get a few percent market share starting from zero than to enlarge the market share after being rather well established.

---

*This is the opinion of the author, because there are differing opinions further empirical studies are necessary.

Inevitably some questions have been raised (particularly after discussions with J. Robinson):

1. Increasing old tech's potential output causes a flooding of the market and decreasing prices. How can this be observed using sensitivity analysis?

2. What sort of technologies are favoured by market contraction (decrease of mkt)?

3. What are the relations between the behaviour of the model with respect to changes of ear and mkt and to the theory that economic fluctuation stimulates or discourages the innovation process (see G. Mensch 1975).

Another possibility for further research on that model is to perform structural sensitivity analysis in order to try to get rid of some state variables by substituting them by algebraic equations. Doing this could gain some more insights into the diffusion process of new technologies.

REFERENCES

Davisson, W.I., and J.J. Uhran (1977) A Primer for NOTRAN: A Systems Dynamics Interpreter. University of Notre Dame. Indiana.

Frank, P.M. (1978) Introduction to System Sensitivity Theory. Academic Press. New York, San Francisco, London.

Haustein, H.-D. and H. Haier (1979) Stimuli and Barrier of Managing Innovations within Industrial Firms. Draft paper. Laxenburg, Austria: International Institute for Applied Systems Analysis.

Mensch, G. (1975) Das technologische Patt, Umschau Verlag, Frankfurt.

Robinson, J. (1979) Technological Shift. WP-79-00. Laxenburg, Austria: International Institute for Applied Systems Analysis.

Sahal, D. (1978) A Theory of Progress Functions. dp/78-10. Berlin. International Institute for Management.

Tomovic, R. and M. Vukobratovic (1970) General Sensitivity Theory, in Richard Bellman (ed) Modern Analytic and Computational Methods in Science and Mathematics. Elsevier.

APPENDIX A

The program listing of TECH1 and of the sensitivity analysis is shown below. In the listing you find $\frac{\partial g}{\partial \alpha}$ (derivative of g after the parameter α) below the comment "parameters". The name indicates where the derivative belongs to, if the character before the last one is a "d" then the derivative should be added to the according delay (below the comment "auxiliary sensitivity variables") which is indicated by the last number (after the "d") either of old or of new technology (according to the sense of the parameter or indicated by the character "o" for old or "n" for new technology in the name of the parameter. This is shown in the program with the parameter "ntpo" in order to give an example.

```
* TECH1
*abint
*nostats
note            OLD TECH CAPACITY, CAPITAL, DEPRECIATION, INVESTMENT
note
l ot.k=integral(otim.jk-otd.jk)                  production capacity (ru)
n ot=otz                                         in'tl prod'n capacity (ru)
c otz=6200                                       "
a otrd.k=ot.k/alt                                real depreciat'n, (ru/yr)
r otd.kl=otrd.k                                  "
c alt=15                                         avg. capacity lifetime  (yrs)
l otk.k=integral(oti.jk-ot$d.jk)                 monetary capital ($k)
n otk=otkz                                       in'tl monetary capital ($k)
c otkz=4200                                      "
r ot$d.kl=otmd.k                                 monetary depreciation ($k/yr)
a otmd.k=otk.k*mdc
c mdc=.10                                        "             rate (%/yr)
expnd dlinf3(otli,otqi,ii,otiz)                  last investm'nt (ru)
c ii=1                                           investment interval (yrs)
a otqi.k=otre.k*pfiot.k                          investment, (ru/yr)
a pfiot.k=tabhl(pfic,otr.k/ear,0,2,.25)*.1       profit factor investment (%)
t pfic=0/1/2.5/5/10/12.5/13.7/14/15              "
a otr.k=(otre.k-otli.k)/otk.k
c ear=.06                                        economywide avg return (%/yr)
r oti.kl=otqi.k                                  investment as a rate ($/yr)
expnd delay3(otim,oti,otid,otiz)                 investment maturat'n (ru/yr)
c otid=5                                         investment delay (yrs)
c otiz=375                                       in'tl investment (ru)
note            OLD TECH OUTPUT, COSTS, RETURNS, EXPERIENCE
note
a otc.k=ot.k*otfc+oto.k*otvc.k+otmd.k            costs ($/yr)
a otre.k=otqd.k*ot$.k-otc.k                      residual earnings ($/yr)
c otfc=.5                                        fixed costs ($/yr)
a otvc.k=tabhl(otvcc,time.k,0,40,10)*.1          variable costs ($/yr)
t otvcc=3/3/3/3/3                                "
a oto.k=ot.k*otef.k*otpo*otcu.k                  output (ru/yr)
c otpo=.667
a otcu.k=tabhl(cuc,ioto.k/esoto.k,0,2,.5)        capacity utilization (%)
t cuc=1.1/1.07/1/.9/.85                          "
expnd dlinf1(esoto,otqd,seft,sez)
c seft=2                                         sales exp'tn formation time (yrs
c sez=4550                                       in'tl sales exp'tn (ru)
a otef.k=tabhl(efc,log(otco.k),0,20,2)*.1        experience factor (%)
t efc=1/3/5/7/9/10/10/10/10/10/10
l otco.k=integral(otoa.jk)                       cumulative output  (ru)
n otco=otcoz                                     in'tl  "
c otcoz=40000                                    "
r otoa.kl=oto.k                                  output accumulation (ru/yr)
note            NEW TECH CAPACITY, CAPITAL, DEPRECIATION, INVESTMENT
note
l nt.k=integral(ntim.jk-ntd.jk)                  real production capacity (ru)
n nt=ntz                                         in'tl  "
c ntz=1                                          "
```

```
a ntrd.k=nt.k/alt                                real depreciat'n, ru/yr
r ntd.kl=ntrd.k                                  real depreciat'n, ru/yr
l ntk.k=integral(nti.jk-nt$d.jk)                 monetary capital ($)
n ntk=ntkz                                       in'tl monetary capital ($)
c ntkz=1                                         "
r nt$d.kl=ntmd.k                                 monetary depreciation ($/yr)
a ntmd.k=ntk.k*mdc                               "
expnd dlinf3(ntli,ntqi,ii,ntiz)                  last investment ($/yr)
a ntqi.k=ntre.k*pfint.k*upmi.k                   investment, (ru/yr)
a pfint.k=tabhl(pfic,ntr.k/ear,0,2,.25)*.1       profit factor investment (%)
a upmi.k=tabhl(upmic,log(ntco.k),0,20,2)         unrealized pot'l mult on investmen
t upmic=8/5/3/1.5/1/1/1/1/1/1                    "
a ntr.k=(ntre.k-ntli.k)/ntk.k                    returns (%/yr)
r nti.kl=ntqi.k                                  investment as a rate (ru/yr)
expnd delay3(ntim,nti,ntid,ntiz)                 investment maturation (ru/yr)
c ntid=5                                         maturat'n delay time (yrs)
c ntiz=0.067                                     in'tl investment (ru)
note            NEW TECH OUTPUT, COSTS, RETURNS, EXPERIENCE
note
a ntre.k=ntqd.k*nt$.k-ntc.k                      residual earnings ($/yr)
a ntc.k=ntfc*nt.k+ntvc.k*nto.k+ntmd.k            costs ($/yr)
c ntfc=.4                                        fixed costs ($/yr)
a ntvc.k=tabhl(ntvcc,time.k,0,40,10)             variable costs ($/yr)
t ntvcc=.2/.2/.2/.2/.2                           "
a nto.k=ntef.k*nt.k*ntpo*ntcu.k                  output (ru/yr)
c ntpo=3                                         pot'l output/capacity (ru/yr)
a ntcu.k=tabhl(cuc,into.k/esnto.k,0,2,.5)        capacity utilizat'n (%)
expnd dlinfl(esnto,ntqd,seft,ntsez)              expected sales (ru/yr)
c ntsez=1                                        init'l sales expectat'n (ru/yr)
a ntef.k=tabhl(efc,log(ntco.k),0,20,2)*.1        experience factor (%)
l ntco.k=integral(ntoa.jk)                       cum output  (ru)
n ntco=ntcoz                                     in'tl  "
c ntcoz=1                                        "      "
r ntoa.kl=nto.k                                  output accumulation (u/yr)
note             MARKET SHARE, INVENTORY, PRICE, DEMAND
note
l fmnt.k=integral(ms.jk)                         fract'n of mkt to nt (%)
n fmnt=fmntz                                     in'tl "
c fmntz=0.0000129                                "
r ms.kl=(afms.k+pfms.k)*(sg.k)                   mkt switching (%/%/yr)
a sg.k=clip(fmnt.k,(1-fmnt.k),nt$.k/ot$.k,1)     switching group (%)
a pfms.k=tabhl(pmmsc,nt$.k/ot$.k,0,3,.5)*.01     price factor mkt switching (%)
t pmmsc=25/10/0/-5/-10/-13/-15                   "
a afms.k=tabhl(afmsc,time.k,0,40,10)*.01         attractiveness factor in switching
t afmsc=0/0/0/0/0
a mkt.k=5000*exp(mgr*time.k)*mv.k                market size (ru/yr with price = $1
c mgr=0                                          market growth (%/yr)
a mv.k=(noise(49204)-.5)*na+1                    market variation (%)
c na=0                                           noise amplitude (%)
l into.k=integral(ntoa.jk-ntqdr.jk)              inventory output  (ru)
n into=intoz                                     in'tl "
c intoz=1                                        "
a ntqd.k=pmdnt.k*fmnt.k*mkt.k                    quantity demanded (ru)
r ntqdr.kl=ntqd.k                                "
a pmdnt.k=tabhl(pmdc,nt$.k,0,3,.5)               price mult demand for nt (%)
t pmdc=1.5/1.2/1/.8/.7/.65/.62
a nt$.k=tabhl(o$c,into.k/esnto.k,0,3,.5)         price ($/ru)
t o$c=10/3/1/.7/.55/.50/.48                      "
l ioto.k=integral(otoa.jk-otqdr.jk)              inventory ot output  (ru/yr)
```

```
n iotc=iotoz                                               in'tl "
c iotoz=4550                                               "
a otqd.k=pmdot.k*(1-fmnt.k)*mkt.k                          quantity demanded (ru/yr)
r otqdr.kl=otqd.k                                          "
a pmdot.k=tabhl(pmdc,ot$.k,0,3,.5)                         price mult demand for ot (%)
a ot$.k=tabhl(o$c,ioto.k/esoto.k,0,3,.5)                   price ($/ru)
note
note SENSITIVITY ANALYSIS OF TECH1
note
c h=.00001
note
note derivatives of table functions
note
a hpfiot.k=tabhl(pfic,h+(otr.k/ear),0,2,.25)*.1
a dpfiot.k=(hpfiot.k-pfiot.k)/h
a hotcu.k=tabhl(cuc,h+(ioto.k/esoto.k),0,2,.5)
a dotcu.k=(hotcu.k-otcu.k)/h
a hotef.k=tabhl(efc,log(otco.k)+h,0,20,2)*.1
a dotef.k=(hotef.k-otef.k)/h
a hpfint.k=tabhl(pfic,h+(ntr.k/ear),0,2,.25)*.1
a dpfint.k=(hpfint.k-pfint.k)/h
a hupmi.k=tabhl(upmic,log(ntco.k)+h,0,20,2)
a dupmi.k=(hupmi.k-upmi.k)/h
a hntcu.k=tabhl(cuc,h+(into.k/esnto.k),0,2,.5)
a dntcu.k=(hntcu.k-ntcu.k)/h
a hntef.k=tabhl(efc,log(ntco.k)+h,0,20,2)*.1
a dntef.k=(hntef.k-ntef.k)/h
a hpfms.k=tabhl(pmmsc,h+(nt$.k/ot$.k),0,3,.5)*.01
a dpfms.k=(hpfms.k-pfms.k)/h
a hpmdnt.k=tabhl(pmdc,h+(nt$.k),0,3,.5)
a dpmdnt.k=(hpmdnt.k-pmdnt.k)/h
a hnt$.k=tabhl(o$c,h+(into.k/esnto.k),0,3,.5)
a dnt$.k=(hnt$.k-nt$.k)/h
a hpmdot.k=tabhl(pmdc,(ot$.k)+h,0,3,.5)
a dpmdot.k=(hpmdot.k-pmdot.k)/h
a hot$.k=tabhl(o$c,h+(ioto.k/esoto.k),0,3,.5)
a dot$.k=(hot$.k-ot$.k)/h
note
note
note derivatives of otqi
note
a auxl.k=pfiot.k+(otre.k*dpfiot.k/(otk.k*ear))
a dotqil.k=(-otfc-otef.k*otpo*otcu.k*otvc.k)*auxl.k
a dotqi2.k=-mdc*pfiot.k+(otre.k*dpfiot.k*
x (-mdc*otk.k-otre.k+otli.k)/(otk.k*otk.k*ear))
a dotqi3.k=-ot.k*dotef.k*otpo*otcu.k*otvc.k*auxl.k/otco.k
a aux2.k=(1-fmnt.k)*mkt.k*dot$.k
a aux3.k=dpmdot.k*ot$.k+pmdot.k
a aux4.k=ot.k*otef.k*otpo*otvc.k*dotcu.k
a dotqi4.k=(aux2.k*aux3.k-aux4.k)*auxl.k/esoto.k
a dotqi7.k=-otre.k*dpfiot.k/(otk.k*ear)
a dotqi8.k=-ioto.k*dotqi4.k/esoto.k
a dotqiz.k=-auxl.k*pmdot.k*mkt.k*ot$.k
note
```

```
note
note derivatives of ntqi
note
a helpl.k=pfint.k+(ntre.k*dpfint.k/(ntk.k*ear))
a dntqil.k=upmi.k*helpl.k*(-ntfc-ntef.k*ntpo*ntcu.k*ntvc.k)
a dntqi3.k=(-nt.k*dntef.k*ntpo*ntcu.k*ntvc.k*helpl.k
x *upmi.k+ntre.k*pfint.k*dupmi.k)/ntco.k
a dntqi2.k=upmi.k*(-mdc*pfint.k+ntre.k*dpfint.k*
x (-mdc*ntk.k-ntre.k+ntli.k)/(ntk.k*ntk.k*ear))
a help2.k=dnt$.k*fmnt.k*mkt.k
a help3.k=pmdnt.k+dpmdnt.k*nt$.k
a help4.k=nt.k*ntef.k*ntpo*dntcu.k*ntvc.k
a dntqi4.k=helpl.k*upmi.k*(help2.k*help3.k-help4.k)/esnto.k
a dntqi7.k=-ntre.k*upmi.k*dpfint.k/(ntk.k*ear)
a dntqi8.k=-dntqi4.k*into.k/esnto.k
a dntqiz.k=pmdnt.k*mkt.k*nt$.k*upmi.k*helpl.k
note
note
a hilfo.k=dotqil.k*sot.k+dotqi2.k*sotk.k+dotqi3.k*sotco.k
x +otto.k
a otto.k=dotqi4.k*sioto.k+dotqi7.k*sotli.k+
x dotqi8.k*sesoto.k+dotqiz.k*sfmnt.k
note
a hilfn.k=dntqil.k*snt.k+dntqi2.k*sntk.k+dntqi3.k*sntco.k
x +ntto.k
a ntto.k=dntqi4.k*sinto.k+dntqi7.k*sntli.k+
x dntqi8.k*sesnto.k+dntqiz.k*sfmnt.k
note
note
note
note jacobian*sensitivityvector
note
a vl.k=(sot.k/alt)
a v2.k=hilfo.k-(mdc*sotk.k)
a ol.k=otef.k*otpo*otcu.k
a o2.k=ot.k*dotef.k*otpo*otcu.k/otco.k
a o3.k=ot.k*otef.k*otpo*dotcu.k/esoto.k
a o4.k=ot.k*otef.k*otpo*dotcu.k*ioto.k/(esoto.k*esoto.k)
a v3.k=ol.k*sot.k+o2.k*sotco.k+o3.k*sioto.k-o4.k*sesoto.k
a o5.k=dpmdot.k*dot$.k*(l-fmnt.k)*mkt.k/esoto.k
a ho5.k=o5.k*(sioto.k-(ioto.k*sesoto.k/esoto.k))
a hoa.k=-pmdot.k*mkt.k*sfmnt.k
a hol.k=ho5.k+hoa.k
a v4.k=v3.k-hol.k
a coefl.k=afms.k+pfms.k
a coef2.k=clip(1,-1,nt$.k/ot$.k,1)
a coef3.k=dpfms.k*dnt$.k/(ot$.k*esnto.k)
a coef4.k=-coef3.k*into.k/esnto.k
a coef5.k=-dpfms.k*nt$.k*dot$.k/(ot$.k*ot$.k*esoto.k)
a coef6.k=-coef5.k*ioto.k/(esoto.k)
a u.k=coefl.k*coef2.k*sfmnt.k+sg.k*(coef3.k*sinto.k+
x coef4.k*sesnto.k+coef5.k*sioto.k+coef6.k*sesoto.k)
note
```

```
a wl.k=(snt.k/alt)
a w2.k=hilfn.k-(mdc*sntk.k)
a nl.k=ntef.k*ntpo*ntcu.k
a n2.k=nt.k*dntef.k*ntpo*ntcu.k/ntco.k
a n3.k=nt.k*ntef.k*ntpo*dntcu.k/esnto.k
a n4.k=-nt.k*ntef.k*ntpo*dntcu.k*into.k/(esnto.k*esnto.k)
a w3.k=nl.k*snt.k+n2.k*sntco.k+n3.k*sinto.k+n4.k*sesnto.k
a n5.k=dpmdnt.k*dnt$.k*fmnt.k*mkt.k/esnto.k
a hn.k=n5.k*(sinto.k-(into.k*sesnto.k/esnto.k))
a hna.k=pmdnt.k*mkt.k*sfmnt.k
a hnl.k=hn.k+hna.k
a w4.k=w3.k-hnl.k
note
note
note rates of the sensitivityvector
note
note
r opl.kl=vl.k
r op2.kl=v2.k
r op3.kl=v3.k
r op4.kl=v4.k
r or.kl=u.k
r npl.kl=wl.k
r np2.kl=w2.k+ntpn2.k
r np3.kl=w3.k+ntpn3.k
r np4.kl=w4.k+ntpn4.k
note
note
note sensitivityvector as levels
note
note
l sot.k=integral(sotim.jk-opl.jk)
l sotk.k=integral(op2.jk)
l sotco.k=integral(op3.jk)
l sioto.k=integral(op4.jk)
l sfmnt.k=integral(or.jk)
l snt.k=integral(sntim.jk-npl.jk)
l sntk.k=integral(np2.jk)
l sntco.k=integral(np3.jk)
l sinto.k=integral(np4.jk)
note
note initial values for the sensitivityvector
note
n sot=0
n sotk=0
n sotco=0
n sioto=0
n sfmnt=0
n snt=0
n sntk=0
n sntco=0
n sinto=0
note
```

```
note computation of the sensitivitymeasures
note
r hsot.kl=sot.k*sot.k
r hotk.kl=sotk.k*sotk.k
r hotco.kl=sotco.k*sotco.k
r hioto.kl=sioto.k*sioto.k
r hfmnt.kl=sfmnt.k*sfmnt.k
r hsnt.kl=snt.k*snt.k
r hntk.kl=sntk.k*sntk.k
r hntco.kl=sntco.k*sntco.k
r hinto.kl=sinto.k*sinto.k
l nsot.k=integral(hsot.jk)
l nsotk.k=integral(hotk.jk)
l nsotco.k=integral(hotco.jk)
l nsioto.k=integral(hioto.jk)
l nsfmnt.k=integral(hfmnt.jk)
l nsnt.k=integral(hsnt.jk)
l nsntk.k=integral(hntk.jk)
l nsntco.k=integral(hntco.jk)
l nsinto.k=integral(hinto.jk)
n nsot=0
n nsotk=0
n nsotco=0
n nsioto=0
n nsnt=0
n nsntk=0
n nsntco=0
n nsinto=0
n nsfmnt=0
r aot.kl=ot.k*ot.k
r aotk.kl=otk.k*otk.k
r aotco.kl=otco.k*otco.k
r aioto.kl=ioto.k*ioto.k
r afmnt.kl=fmnt.k*fmnt.k
r ant.kl=nt.k*nt.k
r antk.kl=ntk.k*ntk.k
r antco.kl=ntco.k*ntco.k
r ainto.kl=into.k*into.k
l bot.k=integral(aot.jk)
l botk.k=integral(aotk.jk)
l botco.k=integral(aotco.jk)
l bioto.k=integral(aioto.jk)
l bfmnt.k=integral(afmnt.jk)
l bnt.k=integral(ant.jk)
l bntk.k=integral(antk.jk)
l bntco.k=integral(antco.jk)
l binto.k=integral(ainto.jk)
note
note sensitivitymeasures
note
a seno.k=ntpo*(sqrt(nsot.k/bot.k)+sqrt(nsotk.k/botk.k)
x +sqrt(nsotco.k/botco.k)+sqrt(nsioto.k/bioto.k))/160
a senf.k=ntpo*sqrt(nsfmnt.k/bfmnt.k)/40
a sen.k=ntpo*(sqrt(nsnt.k/bnt.k)+sqrt(nsntk.k/bntk.k)
x +sqrt(nsntco.k/bntco.k)+sqrt(nsinto.k/binto.k))/160
note
```

```
n bot=.0000001
n botk=.0000001
n botco=.00000001
n bioto=.00000001
n bnt=.00000001
n bntk=.00000001
n bntco=.00000001
n binto=.00000001
n bfmnt=.00000001
note
note relative sensitivityfunctions
note
a rot.k=sot.k/ot.k
a rotk.k=sotk.k/otk.k
a rotco.k=sotco.k/otco.k
a rioto.k=sioto.k/ioto.k
a rfmnt.k=sfmnt.k/fmnt.k
a rnt.k=snt.k/nt.k
a rntk.k=snt.k/nt.k
a rntco.k=sntco.k/ntco.k
a rinto.k=sinto.k/into.k
note
note
note auxiliary sensitivityvariables
note
a otdel1.k=hilfo.k
expnd dlinf3(sotli,otdel1,ii,sot)
a otdel2.k=hol.k
expnd dlinf1(sesoto,otdel2,seft,sot)
r otdel3.kl=hilfo.k
expnd delay3(sotim,otdel3,otid,sot)
note
a ntdel1.k=hilfn.k+ntpnd1.k
expnd dlinf3(sntli,ntdel1,ii,snt)
a ntdel2.k=hnl.k
expnd dlinf1(sesnto,ntdel2,seft,snt)
r ntdel3.kl=hilfn.k+ntpnd3.k
expnd delay3(sntim,ntdel3,ntid,snt)
note
note
note parameters
note
note alt
note
a altol.k=-ot.k/(alt*alt)
a altnl.k=-nt.k/(alt*alt)
note
note mdc
note
a mdo2.k=-otk.k*pfiot.k-(otre.k*dpfiot.k/ear)-otk.k
a mdod1.k=mdo2.k+otk.k
a mdod3.k=mdod1.k
a mdn2.k=(-ntk.k*pfint.k-(ntre.k*dpfint.k/ear))*upmi.k-ntk.k
a mdnd1.k=mdn2.k+ntk.k
a mdnd3.k=mdnd1.k
note
```

```
note otpo
note
a dototp.k=-ot.k*otef.k*otcu.k*otvc.k
a otpo2.k=dototp.k*auxl.k
a otpo3.k=ot.k*otef.k*otcu.k
a otpo4.k=otpo3.k
a otpodl.k=otpo2.k
a otpod3.k=otpo2.k
note
note ntpo
note
a dntntp.k=-nt.k*ntef.k*ntcu.k*ntvc.k
a ntpn2.k=dntntp.k*helpl.k*upmi.k
a ntpn3.k=nt.k*ntef.k*ntcu.k
a ntpn4.k=ntpn3.k
a ntpndl.k=ntpn2.k
a ntpnd3.k=ntpn2.k
note
note afms
note
a afmsor.k=sg.k
note
note ear
note
a eo2.k=-otre.k*dpfiot.k*otr.k/(ear*ear)
a eodl.k=eo2.k
a eod3.k=eo2.k
a en2.k=-ntre.k*dpfint.k*ntr.k*upmi.k/(ear*ear)
a endl.k=en2.k
a en3d.k=en2.k
note
note otfc
note
a otfco2.k=-ot.k*auxl.k
a otfcodl.k=otfco2.k
a otfcod3.k=otfco2.k
note
note ntfc
note
note ntfc
note
a ntfcn2.k=-nt.k*helpl.k*upmi.k
a ntfcndl.k=ntfcn2.k
a ntfcnd3.k=ntfcn2.k
note
```

```
note otvc
note
a dot.k=-ot.k*otef.k*otpo*otcu.k
a otvc2.k=dot.k*auxl.k
a otvcdl.k=otvc2.k
a otvcd3.k=otvc2.k
note
note ntvc
note
a dnt.k=-nt.k*ntef.k*ntpo*ntcu.k
a ntvc2.k=dnt.k*upmi.k*helpl.k
a ntvcdl.k=ntvc2.k
a ntvcd3.k=ntvc2.k
note
note mkt
note
a domkt.k=pmdot.k*(l-fmnt.k)*ot$.k
a mkto2.k=domkt.k*auxl.k
a mkto4.k=-pmdot.k*(l-fmnt.k)
a mktodl.k=mkto2.k
a mktod2.k=-mkto4.k
a mktod3.k=mkto2.k
a dnmkt.k=pmdnt.k*fmnt.k*nt$.k
a mktn2.k=dnmkt.k*upmi.k*helpl.k
a mktn4.k=-pmdnt.k*fmnt.k
a mktndl.k=mktn2.k
a mktnd2.k=-mktn4.k
a mktnd3.k=mktn2.k
note
note otef
note
a otef3.k=ot.k*otpo*otcu.k
a otef4.k=otef3.k
note
note ntef
note
a ntef3.k=nt.k*ntpo*ntcu.k
a ntef4.k=ntef3.k
note
note upmi
note
a upmi2.k=ntre.k*pfint.k
a upmidl.k=upmi2.k
a upmid3.k=upmi2.k
note
note
note
note
note                    PROGRAM SPECIFICATIONS, PERUNS
spec dt=.35,start=0,stop=40,prtper=2
print ot,otk,otco,ioto
print nt,ntk,ntco,into,fmnt
print sot,sotk,sotco,sioto
print snt,sntk,sntco,sinto,sfmnt
print rot,rotk,rotco,rioto
print rnt,rntk,rntco,rinto,rfmnt
print seno,senf,sen
```