MODELING DYNAMIC SYSTEMS OF VARIABLE STRUCTURE

Alexander Umnov

March, 1984
WP-84-19

Working papers are interim reports on work of the International Institute for Applied Systems Analysis and have received only limited review. Views or opinions expressed herein do not necessarily represent those of the Institute or of its National Member Organizations.

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

PREFACE

This paper is devoted to the development of applied software for strategic regional policy-making. Based on the first use of the methodology, suggested by the author in the Skane case study (see WP-82-66), we expect it to be used for other case studies as well.

Andrei Kochetkov

Scientific Supervisor of the
Task: 'Concepts and Tools for
Strategic Regional Policy'.

CONTENTS

Abstract.

Reliable software tools for mathematical modeling have been developed for simulation models but for constrained optimization models progress has been slower.

The following approach examines the possibility of working with large-scale mathematical models in a *compact form* permitting the direct analysis of input-output relations where information is transformed into a compressed mode (as opposed to aggregation procedures) without loss of accuracy.

The article also describes the approach when applied to discrete dynamic models.

Modeling Dynamic Systems of Variable Structure

Alexander Umnov,
International Institute for
Applied Systems Analysis,
A-2361, Laxenburg, Austria

Introduction

This paper examines the mathematical modeling of discrete dynamic systems, *i.e.* those states which are defined for specific fixed points in time.

Usually this problem is solved by building an integrated model, which describes both the states for all points in time under consideration and the links between these states. This leads to an efficient use of computers but a restricted application, for example, when the number of time points considered is unknown or variable, as in minimal-time control problems.

The static approach to modeling a discrete dynamic system requires that the model structure be changed with the time points, which causes computer difficulties.

This method builds a set of consistent submodels, each of them describing the state of the modeled system for a fixed point in time. This requires the solution of a system of problems concerned with finding states of the modeled object with the required properties by using solution procedures which are linked in information systems and synchronized and which may be solved independently. The effect of changing the number of time points considered is reduced to changes in the number of problems included in the system and, therefore, the number of independently-run computer processes. It would be better to use physical as well as logical disaggregation processes (i.e. to use parallel working processors or computers).

The main question is how to organize the information exchange between the submodels, which will direct the whole set to the required states. Consider that all the processes of solving subproblems are imbedded in a common information 'environment', the state of which is described by a vector. Let the components of this vector be exogenous parameters of the linked submodels. An appropriate change in the values of these parameters should to bring all submodels to the required states.

In this situation each of the submodels interacts only with the 'environment'; they receive exogenous information from it and initiate changes in its state. Information links between different submodels perform their function implicitly via the information 'environment'. This makes it easier to change the number of submodels .

Next, we need to define a mathematical form for the interaction. We can include the components of the vector of the 'environmental' state in the description of the submodels to be linked. This enables the 'environment' to influence the states of the submodels. The level of consistency of the submodels and the state of the 'environment' will be interdependent, and the interdependence may be used for iterative changes of this state to achieve better consistency. A method called 'compact modeling' will be applied here to construct a computer procedure to handle the iterative change.

These ideas are developed in this paper.

1. Statement of the Initial Problem

Consider a mathematical model of a discrete dynamic system for the following fixed points in time

$$t=1,2,3,\dots,T.$$

Let the state of the model be described by a vector x^t, which has n components defined for all t, and by a vector u^t, which has l components defined for $t=[1,T-1]$. The formal difference between vectors x^t and u^t is that the first depends on x^{t-1} (*i.e.* pre-history) in an explicit way, but the second does not.

Thus a system of equations

$$x^{t+1}=f^t(x^t,u^t) \tag{1}$$

exists for all $t=[1,T-1]$ describing the dynamic links of the object under investigation.

Moreover, the components of x^t and u^t must satisfy the following set of constraints

$$y_s^t(x^t,u^t)\geq 0,\ s=[1,m]. \tag{2}$$

In particular, these constraints may include boundary conditions, if these exist.

Finally, a functional is assumed to exist which defines the quality of the synthesized sequence of the vectors x^t and u^t

$$\Phi(x^1,x^2,\dots,x^T,u^1,u^2,\dots,u^{T-1})\rightarrow\min. \tag{3}$$

The initial problem (1)-(3) consists of finding sequences $\{ x^t,t=[1,T] \}$ and $\{ u^t,t=[1,T-1] \}$, which satisfy the constraints (1) and (2) and which define extremes for the functional (3) .

Conditions (1)-(3) describe the statements of a wide class of problems which are of practical value and, in some cases, the statements of mathematical models investigated may be reduced to the form of (1)-(3) by means of various artificial transformations.

2. The Information 'Environment' and Parallel Processes

The idea is to change an integrated model of the discrete dynamic system into a set of linked submodels, the number of which can be easily varied.

The original statement (1)-(3) is not suitable for this purpose. Therefore, let us introduce an auxiliary vector V, the components of which may be considered as exogenous parameters of the linked submodels

$$V^{t+1}=f^t(x^t,u^t),\ t=[1,T-1] \tag{4}$$

$$V^t=x^t,\ t=[1,T] \tag{5}$$

$$y_s^t(x^t,u^t)\geq 0,\ s=[1,m] \tag{6}$$

for all $t=[1,T]$. If the system of constraints (4)-(6) is feasible for a fixed V, then it is also possible to evaluate the functional of the model as a dependence of V.

Relations (4)-(6) may be considered as an implicit definition of the dependences $x^t(V^t,V^{t+1})$ and $u^t(V^t,V^{t+1})$ and we can try to reduce the process of solving the initial problem (1)-(3) to a two-level scheme; components of V will be changed on the first level and x^t and u^t will be calculated on the second one. All submodels can now be brought to the desired states by appropriate variations of the components of V. Thus we have constructed an information 'environment' which is single-valued as described by the vector V and in which all submodels are supposedly imbedded.

An optimal state of the 'environment' where all subproblems have solutions with the required properties must be found. In the present case, these solutions have to be considered for different points in time and the functional has to have an extreme value.

In practice this optimal state of the information 'environment' is found through a step-by-step movement towards optimality. It is necessary to formulate rules for finding the local direction of the improvement and for determining the value of the step in this direction.

To demonstrate how this scheme makes it possible to change the number of considered points in time, we solve subproblems (4)-(6) independently when V is fixed. We must start a set of parallel computer processes, each of them solving the subproblem for a given t. We also need a process that finds the optimal state of the information 'environment'. All these processes have to be linked according to principles of informatics and must be synchronized. This means that the main process should be be able to initiate all required processes whenever they become necessary and wait for the results.

In this scheme, changing the number of subproblems T is reduced to changing the number of these parallel processes. The number of components of vector V and lengths of auxiliary arrays for the main process also need to be changed. All these changes are easily made by standard programming practices.

Commands from the operating system can also implement the synchronization of the processes and information exchange between them.

More detailed discussion follows.

3. The Method of Compact Modeling

The obvious approach for finding the optimal state of the information 'environment' is to try to use an iterative scheme which converges to the desired state. We first need a mathematical statement of this searching problem in terms of components of the vector V. The simplest way to formulate this is to exclude $\boldsymbol{x}^t$ and $\boldsymbol{u}^t$ from the statement of (4)-(6), but this is only possible in the case of some trivial problems.

The *compact modeling* method designed for the analysis of systems described by complex mathematical models is more appropriate here.

This method is based on the assumption that, even in the case of a very complex model, the user is interested in the interdependence of only a small number of input and output data and wants to reformulate the model in terms of these data alone.

Assume that there is a high-dimensional vector X, describing a state of a model M. Further, let vector X^* also be a solution of a problem solved by means of the model M and having the required properties. Then the solution process may be formulated as the relation

$$M(X^0)\rightarrow X^*$$

where X^* is an initial state of the model.

Assume that the low-dimensional vectors V and W describe the input and output data respectively. We shall examine the dependence between W and V. We also postulate two conversion processes. The first permits the generation of the initial state of the model by using the initial input data

$$G(V)\rightarrow X^0$$

and the second converts the final state X^* into the output data

$$S(X^*)\rightarrow W.$$

Finally, the dependence between the input and the output data can be written as

$$W=S(M(G(V))).$$

The operator $\hat{M}=S(M(G(\bullet)))$ will be referred to as *the compact image of the model M*.

As the explicit form of this compact image cannot be built for the majority of mathematical models we must find a form which is of more practical use.

One way might be to build this image locally for the immediate vicinity of a current V, rather than globally for the whole set of V under consideration. The following analogy is relevant. The use of the local compact image may be compared with using a part of a power series as a local approximation of a function. This part has a convenient and simple description in comparison with the description of the original function, but this simple description is different for different points. Simply stated, the coefficients of the power series are different.

It could be said, that the compact modeling approach is a method for the modeling of models. The approximating model must be low-dimensional and as simple as possible due to the local nature of the approximation.

But this analogy is limited. The dependence $W=\hat{M}(V)$ is not correctly described by, for example, Taylor approximations, even if the model $X^*=M(X)$ is described by smooth enough functions.

There are three main reasons for this inconvenient property of the dependence $W=\hat{M}(V)$

- a feasible state X of the model M for a given vector V of the input data may not exist,

- the solution x^* of the model M may be non-unique,

- the dependence of X^* on V may be nondifferentiable, even if the functions in (4)-(5)-(6) are themselves differentiable.

These properties are present even in the case of such simple models as linear programming problems or systems of linear inequalities.

To use the standard Taylor approximations, it is necessary to transform the original model M into a new one $\overline{M}$ possessing the following properties:

- the $\overline{M}$ has a solution $\overline{X}$ for any vector V ,

- this solution $\overline{X}$ must be close (in the sense of a metric) to the original solution X^{*} , wherever the latter exists,

- the dependence of W on V must be smooth enough for the use of standard Taylor approximations.

Local approximations of this new form of the model $\overline{M}$ are the same compact images of the original mathematical model, which make it possible to investigate the interdependence of input and output data.

The *practical* effectiveness of the compact modeling approach depends on the specific method used to transform the original model M into $\overline{M}$. Here this problem is considered only for finite-dimensional optimization mathematical models, to which the model (4)-(6) belongs.

4. The Method of Smooth Penalty Functions

The compact modeling approach is now applied to find consistent solutions of the submodels described by relations (4)-(6). To simplify the description we will not define the functional of the whole model.

The compact modeling approach enables us to link together *the compact images* of the submodels, even if the submodels in their original form cannot be linked. The input data are described by the exogenous parameters V. The output data measure the degree of inconsistency of the solutions found for different submodels.

The description of the submodel for a point in time t is

$$\begin{cases} V^{t+1} & = f^{t}(x^{t},u^{t}) \\ V^{t} & = x^{t} \\ y_{s}^{t}(x^{t},u^{t}) & \geq 0,\ s=[1,T] \end{cases} . \tag{7}$$

If $X^t = \begin{bmatrix} x^t \\ u^t \end{bmatrix}$ is the solution of system (7), then the relation $\begin{bmatrix} V^t \\ V^{t+1} \end{bmatrix} \to \begin{bmatrix} x^t \\ u^t \end{bmatrix}$ describes the operator $M(G(\bullet))$ for the submodel (7).

A norm of vector $V^t - x^t$ may be taken as a measure of the inconsistency of the optimal states of the linked submodels. This norm may be calculated by a Euclidean or some other metric. Therefore, the vector W of the output data will have components $W_t = || V^t - x^t ||, t = [1, T]$.

It is not difficult to demonstrate that the dependence between X and V possesses all the undesirable properties mentioned in the previous section. For example, it is possible to choose input vector V, such that the system (7) will be infeasible. In other words, this dependence is not defined for all V.

The dependence may also be non-unique because the system (7) may have a non-unique solution for a given V.

Finally, even if this dependence exists and is unique-valued, the function $X^*(V)$ may be nondifferentiable, because of the fact that system (7) has inequality-type constraints.

This means that all of these submodels must be transformed into the new form $\overline{M}$.

As discussed earlier, the model $\overline{M}$ has to reproduce exactly all the properties of the original model M as well as having an input-output dependence possessing "good" properties,*i.e.* permitting the use of Taylor approximations.

We propose here that an auxiliary function, created for system (7) according to the rules of the smooth version of the Penalty Functions Method, be taken as the transformed model $\overline{M}$ [A.Fiacco, G.McCormick, 1968].

This auxiliary function is

$$E^t = P_=(\tau, || V^{t+1} - f^t(x^t, u^t) ||) + P_=(\tau, || V^t - x^t ||) + \sum_{s=1}^{m} P_\geq(\tau, y_s), \qquad (8)$$

where the so-called *penalty function* $P_{\geq}(\tau,\alpha)$ is defined, is smooth enough for all α and $\tau>0$ and satisfies the following relation

$$\lim_{\tau\to+0} P_{\geq}(\tau,\ \alpha)=\begin{cases}0\ , & \alpha\geq 0\\ +\infty, & \alpha<0\end{cases}$$

The function $P_=$ is usually defined as

$$P_=(\tau,\alpha)=\frac{1}{2}[\ P_{\geq}(\tau,\alpha)\ +\ P_{\geq}(\tau,-\alpha)].$$

Now we will demonstrate that the auxiliary function (8) may be used as the transformed submodel $\bar{M}$.

Firstly, the value of E^t may be used as a measure of the degree of the inconsistency of the submodels because of the equivalence of $E^t=0$ and $W_t=0$.

Secondly,

$$\begin{bmatrix}\bar{x}^t\\ \bar{u}^t\end{bmatrix}=\underset{\begin{bmatrix}x^t\\ u^t\end{bmatrix}}{\arg\min}\, E^t(\tau,x^t,u^t)$$

exists for any vector of the input data as a minimum point for a continuous function which is bounded below.

Thirdly, $\bar{E}^t=E^t(\tau,\bar{x}^t,\bar{u}^t)$ exists and is unique-valued for any V.

Finally, $\bar{E}^t$ will be a differentiable function of V. This is because the *implicit functions theorem* can be applied to the condition of the existence of the stationary state of auxiliary function (8)

$$\nabla_{\begin{bmatrix}x^t\\ u^t\end{bmatrix}} \bar{E}^t(\tau,x^t,u^t) = 0, \tag{9}$$

which defines the implicit functions $\bar{x}^t(V)$ and $\bar{u}^t(V)$, subject to all functions P, f^t, and y_s being smooth enough.

Thus, the vector $\begin{bmatrix}\bar{x}^t\\ \bar{u}^t\end{bmatrix}$ can be considered as a smoothed and predetermined image of the vector $X^{t*}(V)$. Consequently, we may use local Taylor approximations of $\bar{E}^t(V)$ to analyse the properties of the submodels (7).

5. Searching for the Optimal State of the Information 'Environment'

Now we may formulate a mathematical statement of the problem of finding the optimal state of the information 'environment'. To ensure that the values of the components of V guarantee consistency between the linked submodels, it is sufficient to have a situation where the minimum of

$$\varepsilon = \sum_{t=1}^{T} \bar{E}^t(V) \tag{10}$$

equals zero. This value will differ from zero only if there is no consistency between the submodels, or if their internal constraints are contradictory. Therefore, the problem of searching for the optimal state of the 'environment' is equivalent to the minimization of (10).

This minimization procedure may be carried out according to any numerical scheme. To illustrate the applicability of 'compact modeling' we will consider the use of the Newton method, where quadratic approximations of the function to be minimized are used. This requires the calculation of all partial derivatives up to and including the second order.

Using the standard method to calculate the gradient and hessian of the function ε we have

$$\nabla_V \varepsilon = \frac{\partial \varepsilon}{\partial V} + H_{xV} \nabla_x \varepsilon + H_{uV} \nabla_u \varepsilon,$$

where the following notation is used

$$\frac{\partial \varepsilon}{\partial V} = \left|\left| \frac{\partial \varepsilon}{\partial V_i^t} \right|\right|; H_{xV} = \left|\left| \frac{\partial \bar{x}_j^t}{\partial V_i^t} \right|\right|; \quad H_{uV} = \left|\left| \frac{\partial \bar{u}_j^t}{\partial V_i^t} \right|\right|$$

for all t, i, j .

Substituting (10) and taking into account that $\nabla_{x^t} \bar{E}^t = 0$ and $\nabla_{u^t} \bar{E}^t = 0$ from (9) we find

$$\nabla_V \varepsilon = \frac{\partial \varepsilon}{\partial V} + \sum_{t=1}^{T} (H_{xV}^t \nabla_{x^t} \bar{E}^t + H_{uV}^t \nabla_{u^t} \bar{E}^t) = \frac{\partial \varepsilon}{\partial V}$$

This means that it is not necessary, in calculating the gradient, to know the elements of the sensitivity matrices H_{xV} and H_{uV} and also that formulae for the gradient may be written in an explicit form as a function of vectors $V, \bar{x}^t$ and $\bar{u}^t$

$$\nabla_V \varepsilon = \sum_{t=1}^{T} \frac{\partial \bar{E}^t}{\partial V}. \tag{11}$$

Analogously, let us find the second partial derivatives of the minimized function ε . For the hessian of E we have

$$\begin{gathered}
\nabla_V^2 \varepsilon = \frac{\partial^2 \varepsilon}{\partial V^2} + H_{xV} \frac{\partial^2 \varepsilon}{\partial V \partial x} + H_{uV} \frac{\partial^2 \varepsilon}{\partial V \partial u} + \\
H_{xV}(\nabla_x \varepsilon)'_V + H_{uV}(\nabla_u \varepsilon)'_V + \\
(H_{xV})'_V \nabla_x \varepsilon + (H_{uV})'_V \nabla_u \varepsilon
\end{gathered}$$

According to (9), only the first three terms will be nonzero here. Finally we have

$$\begin{aligned}
\nabla_V^2 \varepsilon &= \frac{\partial^2 \varepsilon}{\partial V^2} + H_{xV} \frac{\partial^2 \varepsilon}{\partial V \partial x} + H_{uV} \frac{\partial^2 \varepsilon}{\partial V \partial u} \\
&= \sum_{t=1}^{T} \left(\frac{\partial^2 \bar{E}^t}{\partial V^2} + H_{xV}^t \frac{\partial^2 \bar{E}^t}{\partial V \partial x} + H_{uV}^t \frac{\partial^2 \bar{E}^t}{\partial V \partial u} \right),
\end{aligned} \tag{12}$$

where matrices $\frac{\partial^2 \varepsilon}{\partial V^2}$; $\frac{\partial^2 \varepsilon}{\partial V \partial x}$ and $\frac{\partial^2 \varepsilon}{\partial V \partial u}$ have the components $\frac{\partial^2 \varepsilon}{\partial V_j^t \partial V_i^q}$; $\frac{\partial^2 \varepsilon}{\partial V_i^t \partial x_j^q}$ and $\frac{\partial^2 \varepsilon}{\partial V_i^t \partial u_j^q}$ for all feasible indexes.

The quadratic approximation

$$\bar{E}^t (V_0) + (V - V_0) \nabla_V \bar{E}^t + \frac{1}{2} (V - V_0) \nabla_V^2 \bar{E}^t (V - V_0)$$

may be constructed if all components of $\bar{E}^t(V)$, $\nabla_V \bar{E}^t$ and $\nabla_V^2 \bar{E}^t$ have been calculated for a given point V_0 . This approximation is the *compact local image* of the submodels (7) for the vicinity of the point V_0 .

There is a theoretical case where an explicit form for the function $E(V)$ can be found; however, in practice we will use a numerical algorithm for the minimization so a local approximation is sufficient.

Searching for the optimal state of the information 'environment' is an iterative process, at each step of which the following elements are calculated:

- vectors $\bar{x}^t, \bar{u}^t$ and , if necessary, sensitivity matrices H^t_{xV} and H^t_{uV}.
- a direction ω, along which there is a trend to a decrease in the auxiliary function E,
- ρ - the length of a step along this direction ω, which guarantees the convergence of the whole process.

Finally, vector V^{k+1} describes a new 'improved' state of the information 'environment'

$$V^{t+1} = V^t + \rho\omega$$

after the kth iteration .

6. A Linear Dynamic Model.

The efficiency of this approach depends on the volume of required calculation, which will mainly build the compact images of the submodels. In its turn, this volume depends on the concrete type of the functions (7), so it seems pointless to try to elaborate a scheme effective for all cases. It is more practical to take into account specific features of the submodels to be linked.

For example take the following linear dynamic model

$$x^{t+1} = A^t x^t + B^t + c^t \tag{14}$$

for all $t = [1, T-1]$,

$$D^t + G^t u^t + f^t \geq 0, \;\; s = [1, m] \tag{15}$$

with a functional

$$\sum_{t=1}^{T} p^t x^t + \sum_{t=1}^{T-1} q^t u^t \;\rightarrow min, \tag{16}$$

where $A^t,B^t,D^t,g^t,c^t,f^t p^t,q^t$ are constant matrices of appropriate dimensions. As in the general case, consider dynamic linear models without the functional (16).

The submodel (4)-(5)-(6) in this case will be

$$A^t x^t + B^t u^t + c^t = V^{t+1}, \tag{17}$$

$$x^t = V^t \tag{18}$$

$$D^t x^t + G^t u^t + f^t \geq 0. \tag{19}$$

The penalty functions P will be chosen in the following form

$$P_{=}(\tau, \alpha) = \frac{\alpha^2}{2\tau}$$

$$P_{\geq}(\tau, \alpha) = \frac{1}{2\tau}\left[\frac{\alpha - |\alpha|}{2}\right]^2.$$

For the auxiliary function (8) we have the scalar form

$$\begin{aligned} E^t = {} & \frac{1}{2\tau}\sum_{i=1}^{n}\Big(\sum_{j=1}^{n} A_{ij}^t x_j^t + \sum_{j=1}^{l} B_{ij}^t u_j^t + c_i^t - V_i^{t+1}\Big)^2 \\ & + \frac{1}{2\tau}\sum_{i=1}^{n}(x_i^t - V_i^t)^2 \\ & + \frac{1}{8\tau}\sum_{s=1}^{m}\Big(\sum_{j=1}^{n} D_{sj}^t x_j^t + \sum_{j=1}^{l} G_{sj}^t u_j^t + f_s^t - \Big|\sum_{j=1}^{n} D_{sj}^t x_j^t + \sum_{j=1}^{l} G_{sj}^t u_j^t + f_s^t\Big|\Big)^2 \end{aligned} \tag{20}$$

where $A_{ij}^t, B_{ij}^t, D_{sj}^t, G_{sj}^t, c_i^t, f_s^t$ are elements of the matrices A^t,B^t,D^t,G^t,c^t,f^t respectively.

Minimizing the piece-wise quadratic function (20) with respect to x^t and u^t we find $\begin{bmatrix}\bar{x}^t \\ \bar{u}^t\end{bmatrix}$ and $\bar{E}^t(\bar{x}^t, \bar{u}^t)$.

This problem may be solved with standard software .

Calculating the elements of sensitivity matrices is reduced to solving a set of systems of linear equations built directly by differentiation of the equations (9) with respect to components of the vector V

$$\begin{cases} \displaystyle\sum_{i=1}^{n}\frac{\partial^2 E^t}{\partial x_i^t \partial x_j^t}\frac{\partial x_j^t}{\partial V_r^t} + \sum_{i=1}^{l}\frac{\partial^2 E^t}{\partial x_i^t \partial u_j^t}\frac{\partial u_j^t}{\partial V_r^t} = -\frac{\partial^2 E^t}{\partial x_i^t \partial V_r^t}, & i=[\,1,n\,] \\ \displaystyle\sum_{i=1}^{n}\frac{\partial^2 E^t}{\partial x_i^t \partial u_j^t}\frac{\partial x_j^t}{\partial V_r^t} + \sum_{i=1}^{l}\frac{\partial^2 E^t}{\partial u_i^t \partial u_j^t}\frac{\partial u_j^t}{\partial V_r^t} = -\frac{\partial^2 E^t}{\partial u_j^t \partial V_r^t}, & j=[\,1,l\,] \end{cases}$$

Taking (20) into account, we get

$$\frac{\partial E^t}{\partial x_k^t \partial x_j^t} = \frac{1}{\tau}\sum_{i=1}^{n} A_{ik}^t A_{ij}^t + \frac{1}{\tau}\delta_{kj} + \frac{1}{\tau}\sum_{s=1}^{m} D_{sk}^t D_{sj}^t \vartheta_s^t$$

$$\frac{\partial E^t}{\partial x_k^t \partial u_j^t} = \frac{1}{\tau}\sum_{i=1}^{n} A_{ik}^t B_{ij}^t + \frac{1}{\tau}\sum_{s=1}^{m} D_{sk}^t G_{sj}^t \vartheta_s^t$$

$$\frac{\partial E^t}{\partial u_k^t \partial u_j^t} = \frac{1}{\tau}\sum_{i=1}^{n} B_{ik}^t B_{ij}^t + \frac{1}{\tau}\delta_{kj} + \frac{1}{\tau}\sum_{s=1}^{m} G_{sk}^t G_{sj}^t \vartheta_s^t$$

where

$$\delta_{kj} = \begin{cases} 1, \text{ if } k = j \\ 0, \text{ if } k \neq j \end{cases}$$

and

$$\vartheta_s^t = \begin{cases} 1, \text{ if } s\textit{-th constraint is active} \\ 0, \text{ if } \textit{otherwise} \end{cases}$$

To learn whether the s-th constraint is active or not, substitute values of $\bar{x}_i^t$ and $\bar{u}_j^t$ directly into (18).

Let us derive formulae for components of the gradient and hessian of the function ε . Starting from (11) and taking (20) into account, we find

$$\varepsilon(V)'_{V_r^t} = -\frac{1}{\tau}(\sum_{j=1}^{n} A_{rj}^{t-1}\bar{x}_j^{t-1} + \sum_{B_{rj}^{t-1}} \bar{u}_j^{t-1} + c_r^{t-1} - V_r^t) + \frac{1}{\tau}(V_r^t - \bar{x}_r^t).$$

Analogously, we can find other nonzero elements of (12)

$$\frac{\partial^2 \varepsilon}{\partial V_r^t \partial V_q^t} = \frac{2}{\tau}\delta_{rq} \ ; \quad \frac{\partial^2 \varepsilon}{\partial V_r^t \partial x_i^t} = -\frac{1}{\tau} ; \quad \frac{\partial^2 \varepsilon}{\partial V_r^t \partial x_i^{t-1}} = -\frac{A_{ri}^{t-1}}{\tau} ; \quad \frac{\partial^2 \varepsilon}{\partial V_r^t \partial u_i^{t-1}} = -\frac{B_{ri}^{t-1}}{\tau} \ .$$

Substituting the above formula and components of sensitivity matrices in (12) gives values of elements of the hessian of ε .

To summarize these results, we can say that the building of a quadratic approximation of a compact image for the submodels (17)-(18)-(19) consists of the following elementary problems:

- minimization of a piece-wise quadratic function of n + 1 variables, which could be done by standard software,

- defining the subset of active constraints by means of direct substitution,

- calculation of components of the sensitivity matrices by solving a set of systems of linear equations.

Note that there have been no limitations introduced to the possible methods of optimization of the state of the information 'environment'. The only difficulty is the need to regularize the hessian of ε when its singularity takes place.

7. An Illustrative Example.

In this section the approach under consideration is applied to a linear dynamic model and solved using the VAX-11/780 computer working under the UNIX operation system .

The MINOS mathematical programming system [Murtagh B., 1980] has been used to find an optimal state of the information 'environment' and to build the compact images of linked submodels . In this system both linear and non-linear problems can be solved, but it is especially effective in the case of non-linear problems with linear constraints.

The optimization of the auxiliary function (20) consists of two stages. The first stage, the systems of linear inequalities (17)-(18)-(19), are solved by a modified version of the simplex-method. As a rule these systems are infeasible. The final basis of the first stage was used as an initial basis for the second stage.

The second stage consists of minimizing the following piece-wise quadratic function with respect to α, β, γ, x and u

$$\underline{E}^t = \frac{1}{2}\sum_{i=1}^{n}(\ \alpha^2 + \beta^2\ ) + \frac{1}{2}\sum_{s=1}^{m}(\ \frac{\gamma - |\gamma|}{2})^2,$$

subject to linear equality-type constraints

$$\alpha_i = \sum_{j=1}^{n} A_{ij}^t x_j^t + \sum_{j=1}^{l} B_{ij}^t u_j^t + c_i^t - V_i^t$$

$$\beta_i = x_i^t - V_i^t$$

$$\gamma_s = \sum_{D_{st}^t} x_j^t + \sum_{j=1}^{l} G_{sj}^t + f_s^t\ .$$

where none of the variables have any constraints on their sign.

As the MINOS system solves nonlinear problems using the derivatives of the first order only, it is not necessary to calculate sensitivity matrices here.

The whole problem is solved by T+1 processes. The main process of optimizing the state of the information 'environment' initiates optimization processes of functions (20), whenever values of ε and components of its gradient are required.

The MPS-files for the submodels are prepared and corrected by specially written subroutines.

The optimal number of the considered periods of time is found by a one-dimensional search according to the following rule: if for a given T the system of the submodels is consistent (in this case ε is zero), then T has been decreased by one. If the consistency is not reached (ε is different to zero), then T is replaced by T+1. The solution is the minimal T, when the consistency still takes place.

Finally, an illustrative problem.

Let us find the minimal number of periods required to bring the system described by vector $|| x_1^t, x_2^t, x_3^t ||$ from the initial state $|| 1., 1., 1. ||$ to the final one satisfying $x_1^T \geq 4.$, $x_2^T \geq 2.5$ and $x_3^T \geq 1.2$, if

$$\begin{bmatrix} x_1^{t+1} \\ x_2^{t+1} \\ x_3^{t+1} \end{bmatrix} = \begin{bmatrix} 0.8 & 0.2 & 0.1 \\ 0.1 & 0.7 & -0.2 \\ -0.3 & 0.0 & 0.8 \end{bmatrix} \begin{bmatrix} x_1^t \\ x_2^t \\ x_3^t \end{bmatrix} + \begin{bmatrix} 1.0 & 0.2 \\ 0.5 & 0.5 \\ 0.3 & 0.8 \end{bmatrix} \begin{bmatrix} u_1^t \\ u_2^t \end{bmatrix} .$$

subject to

$$|x_1^t - x_2^t| \leq 1.5$$
$$|x_2^t - x_3^t| \leq 1.5$$
$$|u_1^t| \leq 1., |u_2^t| \leq 1.$$

It is easy to verify that $T=1$ is not the solution. Therefore, the process starts from $T=2$. $T=9$ shows a desirable level of consistency but the exact solution is $T=10$ when the value of ε appears small.

Table 1 gives all the information.

Table 1.

T	e
2	574.02
3	41.50
4	13.16
5	8.27
6	4.20
7	2.71
8	1.50
9	0.007
10	0.0

Figure 1 shows the graph of the dependence of ε on T .

Table 2 gives the optimal solution for $T=10$.

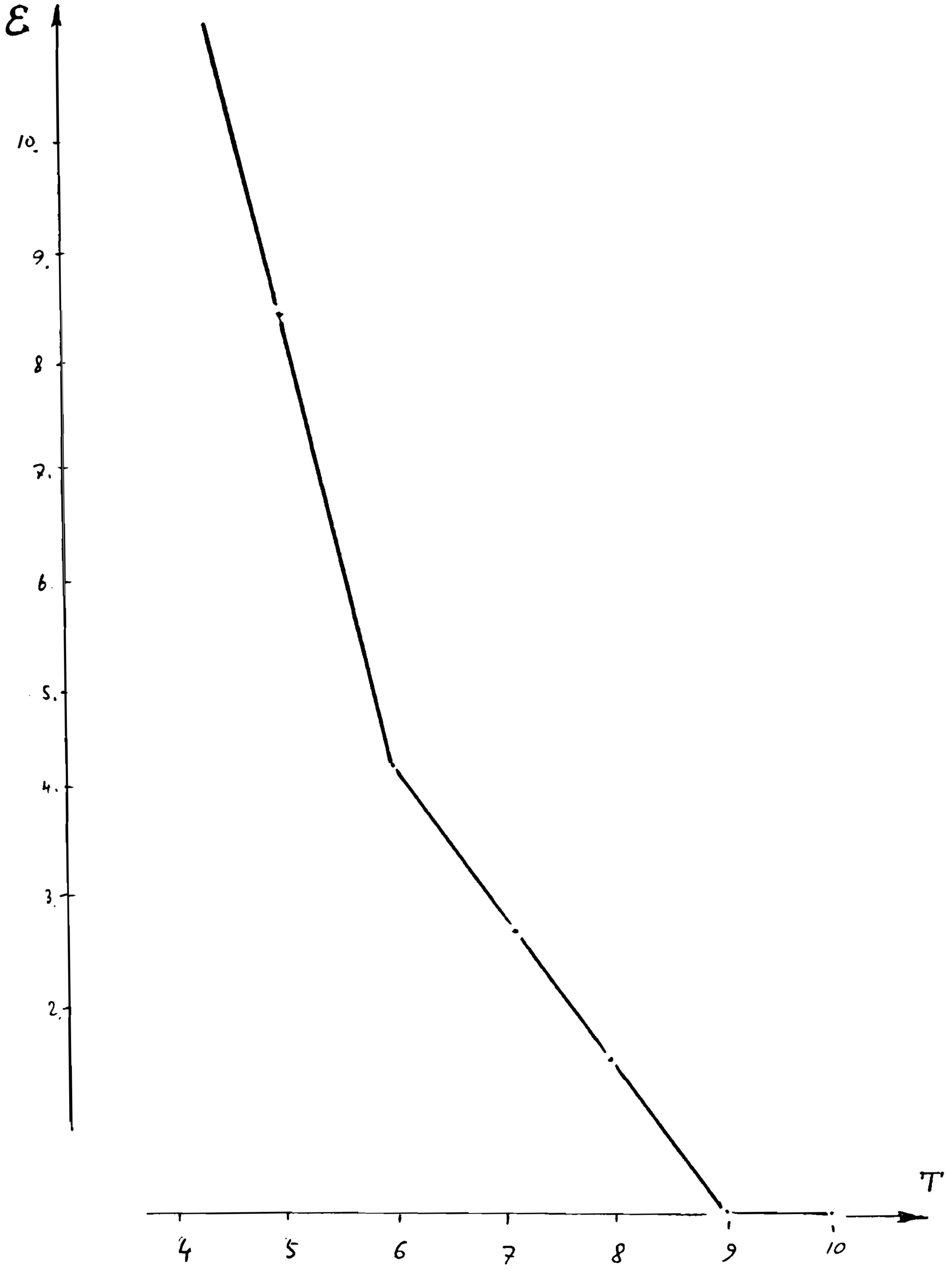

Figure 1 .

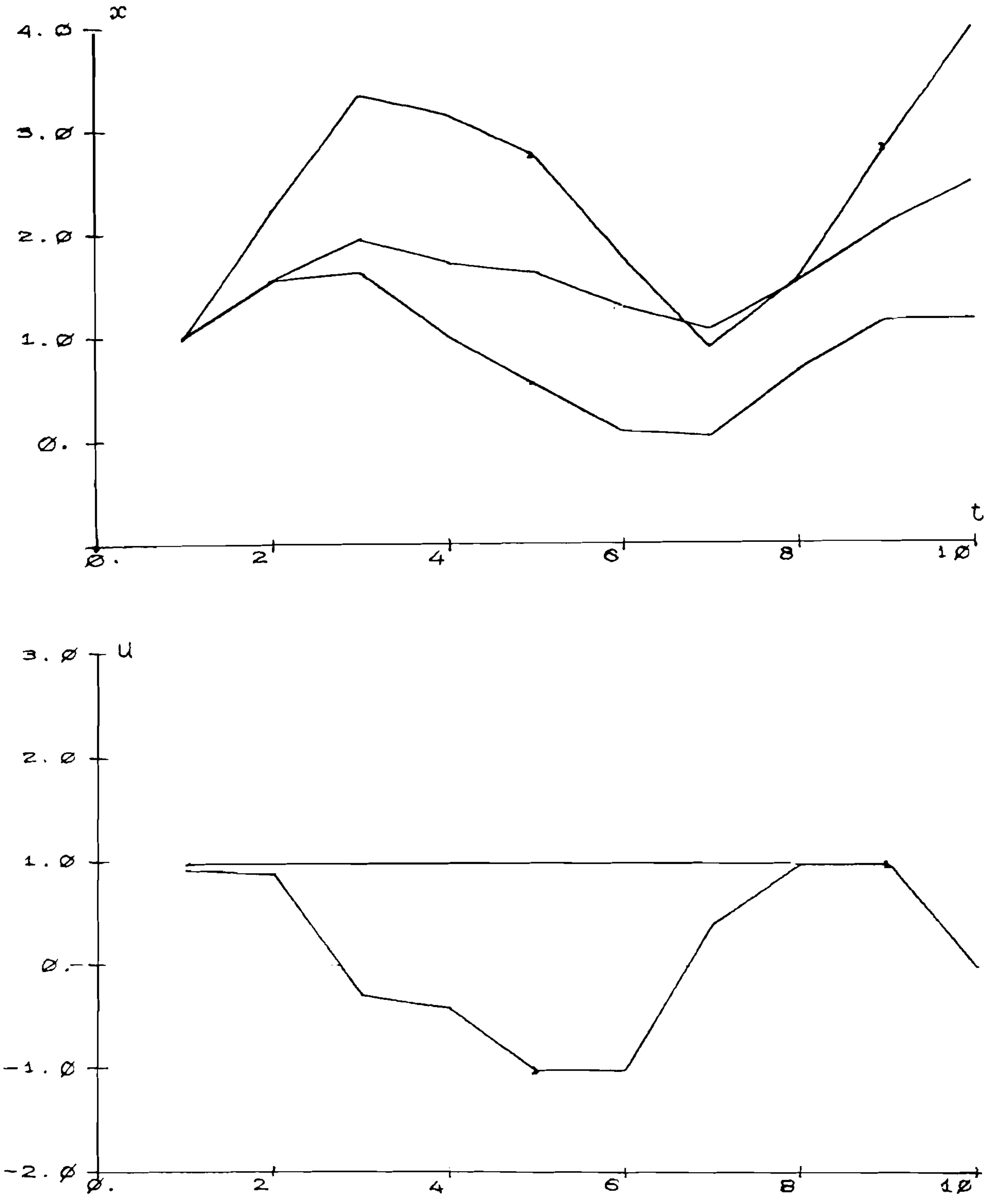

Figure 2 .

Table 2.

t	x(1)	x(2)	x(3)	u(1)	u(2)
1	1.00000	1.00000	1.00000	0.93114	0.97850
2	2.22684	1.55482	1.56214	0.89817	0.99986
3	3.34679	1.94764	1.65100	-0.27609	0.99996
4	3.15597	1.72977	1.03390	-0.39998	0.99944
5	2.77403	1.61938	0.55988	-0.99994	0.99985
6	1.79911	1.29895	0.11560	-0.99961	0.99999
7	0.91103	1.06624	0.05285	0.41783	0.99984
8	1.56515	1.53574	0.69419	0.99994	1.00000
9	2.82863	2.09266	1.18579	1.00000	1.00000
10	4.00000	2.51054	1.20000	0.0	0.0

Figure 2 shows these dependencies in graph form .

The initial basis is taken from the solution with $T=9$. The vector V corresponding to this state of the information 'environment' has components given in Table 3 .

Table 3 .

t	V(1)	V(2)	V(3)
1	1.00000	1.00000	1.00000
2	1.90013	0.06004	0.00000
3	2.35903	1.16179	0.00001
4	2.01153	1.39581	0.00000
5	1.43099	1.35071	-0.00005
6	0.06140	1.09019	0.00701
7	1.49853	1.60526	0.08477
8	2.80415	2.10948	1.33219
9	4.00000	2.50000	1.33219

The value of auxiliary function (10), while searching for the optimal state of the information 'environment' for $T=10$, is given in Table 4 .

Table 4 .

# of iteration	ε	Mean inconsistency
0	1.673 e -1	0.129
5	9.290 e -2	0.096
10	1.531 e -3	0.012
15	4.046 e -4	0.002
20	5.010 e -5	0.0007
25	8.031 e -8	0.00003

The optimal values of the components V_i^t do not differ from the values of x_i^t given in Table 2 .

Related Work

In this paper the compact modeling approach has been applied to the analysis of discrete dynamic models. It has also been applied to other problems involving the interdependence of a small number of inputs and outputs of a complex model. Examples include linkage procedures for optimization mathematical models [Umnov A., Albegov M., 1981] and optimization of the share of the Pareto set in multicriteria mathematical models [Umnov ,1982]. The most detailed description of practical aspects for using the compact modeling approach in the case of large-scale models can be found in [Umnov 1983].

Acknowledgments

The author would like to thank Ing. Miloslav Lenko for all his work on coding the algorithm and carrying out numerical experiments.

References

Fiacco, A.V. and McCormick, G.P. 1968. Nonlinear Programming: Sequential Unconstrained Minimization Techniques. New York. Wiley.

Umnov, A.E. and Albegov, M.M. 1981. An Approach to Distributed Modeling. Behavioral Science, Vol. 26, pp. 354-365.

Umnov, A.E. 1982. Software for Regional Studies: Analysis of Parametrical Multicriteria Models, WP-82-66, International Institute for Applied Systems Analysis, Laxenburg, Austria.

Umnov, A.E. 1983. Impacts of Price Variations on the Balance of World Trade,PP-83-03,International Institute for Applied Systems Analysis, Laxenburg, Austria.