Computer-assisted Modeling of Ill-defined Systems

FRANCOIS E. CELLIER
Institute for Automatic Control
The Swiss Federal Institute of Technology Zurich
ETH-Zentrum, CH-8092 Zurich, Switzerland

ANDREAS FISCHLIN
Department of Entomology
The Swiss Federal Institute of Technology Zurich
ETH-Zentrum, CH-8092 Zurich, Switzerland

This paper shows that the highly sophisticated simulation software as it is available of today is not too well suited for the analysis of socalled ill-defined systems. In a first section of this paper we shall identify the difficulties inherent in ill-defined system modeling. The second part then deals with a discussion of possible adaptations and enhancements of simulation software for improving its modeling capabilities. The resulting enhanced simulation languages which may be more appropriately called "model-oriented" languages are expected to be much better suited for the derivation of models for ill-defined systems than the currently available simulation software.

For the illustration of our statements we shall cite examples from biological modeling (larch bud moth), from social modeling (Forrester's WORLD1 model), as well as from psychological modeling (conformity of individuals with the judgement of groups).

1. INTRODUCTION

Continuous simulation software has originally been developed to fit the needs of the aircraft industry. In this domain (as this is the case with almost all electro/mechanical systems), the equations for the description of models are well known, and even their parameters can often be identified with relative ease. What was needed, was an easy means to integrate differential equations, produce graphical representations of state variables versus time and versus one each other, design simulation experiments, means which are supposed to be easier in comparison with the facilities offered in any general purpose computer software (like FORTRAN-IV, ALGOL-60, or PASCAL). In recent years, the simulation software offered on the "software market" became highly sophisticated and extremely well suited for this purpose.

This is no longer the case, as soon as we are confronted with simulating socalled ill-defined systems, that is, systems of which we hardly know the physical laws governing the basic relations between different state variables, or even find it difficult to select adequate state variables. The model finding procedure becomes here the essential part of the research, whereas coding a once developed model into any available simulation program is now the -- rather trivial -- final step of the investigation. For this task, we require software able to help us in the model finding procedure itself.

In a first step, we shall try to describe more clearly, where the difficulties of ill-defined system modeling can be located, what makes these systems more difficult to tackle.

2. BIOLOGICAL MODELING

How are models developed in general? First, we should distinguish between two principally different types of modeling efforts:

a) Derivation of compound models, and
b) Derivation of basic models.

In the former case, we assume that we can decompose the system under investigation into subsystems for which appropriate models have already been found. The modeling effort is, thus, restricted to combining previously determined models to larger units. Most of the modeling efforts for electrical and/or mechanical systems fall nowadays under this former category. This can be shown at a very simple example of an electrical network. Previous experiments have shown that a resistor can be modeled by the simple law U=R*I under a large variety of experimental conditions. If we now want to model the behaviour of an electrical network consisting of many resistors, this can easily be accomplished by simply applying the previously found basic model describing the behaviour of one single resistor again and again. We can assume such a model to be valid, even without being forced to build the compound system itself. Such models can, thus, be used for design purposes, e.g. to determine beforehand which realization of a proposed equipment out of a variety of alternatives shall suit our purposes best (layout).

Models in biology, on the other hand, fall primarily under the latter of the two categories. A decomposition into previously known subsystems is not possible. For such systems, we require the system under investigation to be already existing and accessible for experimentation. Model derivation is done by first applying a set of different experimental conditions to the real system, and measuring the input-output behaviour of that system, before postulating a model able to reproduce this input-ouput behaviour sufficiently well under the same set of experimental conditions.

Why is it such that many more basic models exist for electrical and mechanical systems than this is the case for biological systems? Two major reasons may be mentioned:

a) The eigen-frequencies inherent in electrical

systems are usually in the order of kHz or even MHz. Mechanical systems show mostly eigen-frequencies of 10 Hz to 100 Hz. On the other hand, biological systems have in most cases eigen-frequencies of μHz or even nHz. This makes it extremely difficult to execute experiments which contain a sufficient amount of the dynamic behaviour of the system to allow appropriate models to be identified.

b) Any system under investigation is, in reality, a multi-input system. This is even true for the simple resistor mentioned above. The value of R shall, for instance, certainly be depending on the temperature of the surrounding air. However, those other influences, beside of the electromagnetic phenomena, are by several orders of magnitude smaller, and may, therefore, be neglected. This does not hold for biological systems. They are always subject to many different influences in the same order of magnitude. The growing speed of a flower in spring depends not only on temperature, but as well on photoperiod, air and soil humidity, nutrients in the soil, solar radiation, wind velocity, rain, microbial activity of the soil, and so on. All those influencing factors usually have effects of the same order of magnitude. A model which does not account for all of those influencing factors (e.g. by referencing them as inputs), cannot be properly validated because repeated (real-life) experiments must then certainly produce data streams which cannot be reproduced by the model. On the other hand, if we happen to be able to reproduce experiments by applying very accurate laboratory conditions, we may derive models validatable under those laboratory conditions, but which are not at all applicable to the really interesting field situation.

These two properties make the modeling of biological systems a very tough task as we shall illustrate at the example of the larch bud moth.

The larch bud moth is an insect attacking, in the European Alps, regularly all larch trees at an altitude between 1800 and 2000 m above sea level. Fig.1 shows how the average population density of the larch bud moth was changing over the past 30 years in the Upper Engadine Valley in Switzerland. After having taken measurements over this long time span, we may now hope to be able to postulate a model which can predict the population density with sufficiently high precision. A second more intricate aim of the model could be to determine in which way the forest engineer could intervene to prevent future outbreaks of the larch bud moth without disturbing the ecological equilibrium of the valley too much.

Fig.1 illustrates drastically the difficulty (a) expressed above. Data collection has not been done in an afternoon's experiment, rather it required to sample over three entire decades which is a most laborious and expensive undertaking involving tree climbing and examination of hundreds of kilograms of larch branches each year! Looking at Fig.1, it seems obvious that a limit cycle exists in this system. We could, thus, postulate a simple Lotka-Volterra type model consisting of one consumer and one prey to start with.

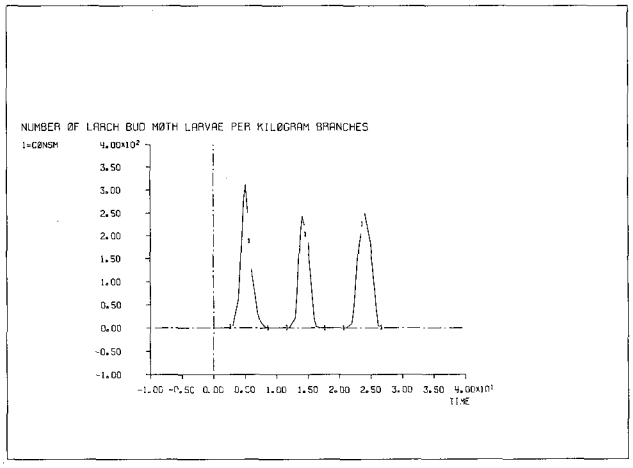

Fig.1: Population density of the larch bud moths in the Upper Engadine Valley (data from [2])

This can be modeled as:

$$\dot{x}_1 = a*x_1 - b*x_1*x_2 \quad ; \quad x_1(t_0) = x_{10}$$
$$\dot{x}_2 = -c*x_2 + d*x_1*x_2 \quad ; \quad x_2(t_0) = x_{20}$$

where x_1 describes the pupulation of the prey (larch), and where x_2 describes the population of the herbivorous consumer (larch bud moth). The parameters a and c describe the regeneration rate of the trees and of the insects, whereas b and d describe the interaction between the two species. We can now use an identification procedure to determine the four parameters a, b, c, and d, as well as the two initial conditions for the state variables. The insects are expressed in number of larvae per kilogram branches, whereas the prey is expressed in kilograms of needle biomass.

This is a very simple model, and we can expect the identification not to create any particular problems. Unfortunately, such an assumption would be incorrect, as we are going to show.

First, we must ask ourselves how many parameters are identifiable from available measurements. For this purpose, we apply a linear variable transformation:

$$x_1 = (c/d)*y_1 \quad ; \quad x_2 = (a/b)*y_2 \ .$$

This leads to the following set of state equations:

$$\dot{y}_1 = a*(1.0 - y_2)*y_1$$
$$\dot{y}_2 = -c*(1.0 - y_1)*y_2 \ .$$

In this new formulation, two parameters are eliminated. Obviously, the parameters b and d do not influence the shape of the state trajectories. They are just multiplying factors of the output equations for computation of the physical variables $\underline{x}$ out of the internal state variables $\underline{y}$. As a result of this, we cannot hope to identify the parameter d through measurements of the consumer alone, and, vice versa, we shall not be able to identify b from measurements of the prey, that is, the system is not fully observable without measurements of both state variables. This has

been shown in [13].

Next, we have to determine an appropriate performance index. Due to the tremendous variations in x_2 (as shown on Fig.1), a simple ISE criterion (integral squared error) may not be appealing. Instead, we prefer to use a logarithmic scale, leading to:

$$PI = \int_{t_0}^{t_f} (\log(x_2) - \log(\hat{x}_2))^2 dt = \text{min!}$$

under the assumption that we are only interested to fit the consumer. Values with a cap denote measured quantities. This choice for the performance index creates, however, additional difficulties since we now have to guarantee that x_2 never takes negative values. For this reason, we apply a second variable transformation:

$$z_1 = \log(x_1) \quad ; \quad z_2 = \log(x_2)$$

leading to the new set of differential equations:

$$\dot{z}_1 = a - b*\exp(z_2)$$
$$\dot{z}_2 = d*\exp(z_1) - c \ .$$

Lucky enough, we do not have to compute logarithms in this new formulation any longer. By this transformation, the first sector of the $\underline{x}$-plane corresponds to the entire $\underline{z}$-plane. Since there exist no limitations on acceptable values of the parameters a, b, c, and d in the $\underline{z}$-plane, no trajectory starting within the first sector of the $\underline{x}$-plane ever leaves this sector. Therefore, we can either solve the problem in the $\underline{x}$-variables by adding two inequality constraints for the initial conditions:

$$x_{10} > 0.0 \quad ; \quad x_{20} > 0.0$$

or, alternatively, we can solve the entire optimization study directly in the $\underline{z}$-variables without additional constraints.

This approach works neatly as can be seen from Fig.2 in which the measured and the simulated (average) population density are plotted together after a curve fitting has been carried out.

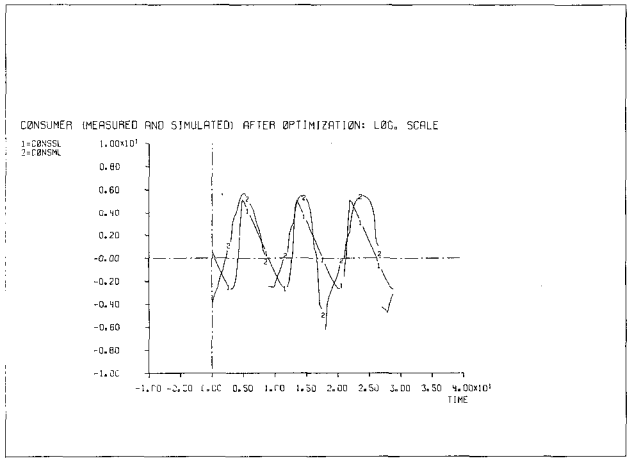

Fig.2: Curve fitting with Lotka-Volterra model. Insect is assumed to be the consumer.

The identification led to the values:

$$a = 0.3005 \ ; \ b = 0.01159 \ ; \ c = 2.8679$$

together with the initial conditions:

$$x_{20} = 0.2378 \ .$$

This simple model seems to represent the influence of the most dominating factor in the system to a very good extent, and we may be inclined to conclude that the interaction between insects and trees is the major responsible factor for the limit cycle in the population density of the insects. In fact, this may be true, however, the simulation results shown in Fig.2 do not <u>prove</u> anything. The reasons against hasty acceptance of models which fit reality well instead of using them as a good working hypothesis only shall be shown at a somewhat simpler example.

Somebody is supposed to model the tramway traffic between two stations "A" and "B". Two different tramways ("1" and "2") are used between these two stations. Since the modeler has to travel between "A" and "B" anyhow, he designs the following experiment: Reaching station "A" any time of the day, he waits for the next tramway to come. While traveling by that street car, he writes down in his note book:

a) The time he spent while waiting, and
b) the tramway type he travels with.

He recognizes after some days that he has to wait between zero and nine minutes (uniformally distributed), and that he uses tramway "1" in nine out of ten cases. From that he concludes that tramways arrive at station "A" with a uniform distribution between zero and nine, and that type "1" circulates nine times more often than type "2". This is certainly a valid model <u>under the given experimental conditions</u>. However, we cannot be sure that the internal structure of that model is correct. More likely seems to be, as it is indeed the case, that both types of tramways follow a time table. Each type passes through station "A" every ten minutes, but type "2" follows always one minute behind type "1" which would lead to the same input-output behaviour under the given experimental conditions. The first model is nevertheless not to be called incorrect if our aim was to describe the way somebody is traveling from "A" to "B". Indeed, it is very temptating to jump to conclusions from misinterpreted modeling efforts.

Coming back to our population dynamics example: If we try to identify as well the prey with the available measurements, we find that no acceptable fit results. We could, therefore, assume that our postulated model was incorrect, and look for another possible reason for the oscillatory behaviour of the insect population.

It is known that the larch bud moth suffers from parasites (up to 80% infection). We could, therefore, assume that, using the same model as before, the insects represent the prey rather than the consumer, whereas now the parasites act as "consumer". Fig.3 shows the result of an identification in which the prey has been fitted to the available measurements of insect population, using the performance index:

$$PI = \int_{t_0}^{t_f} (\log(x_1) - \log(\hat{x}_1))^2 dt = \text{min!}$$

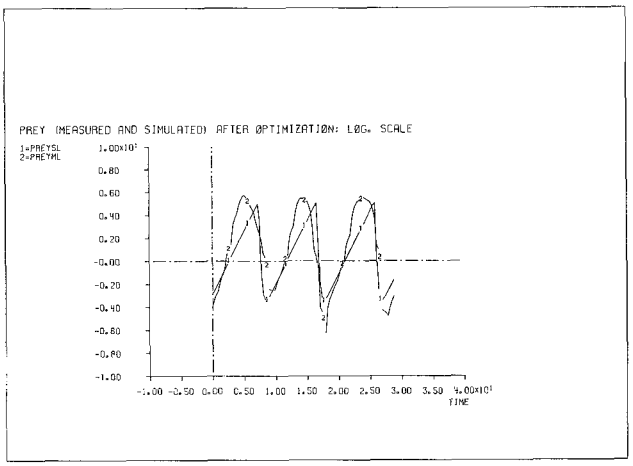

Fig.3: Curve fitting with Lotka-Volterra model. Insect is assumed to be the prey.

This time, we obtained the following values for the parameters:

$$a = 1.0934 \; ; \; c = 3.4272 \; ; \; d = 0.1782$$

together with the initial condition:

$$x_{10} = 0.05766 \; .$$

The achievable fit is now somewhat less appealing than before. However, it may be still acceptable. Unfortunately, an identification of the other state variable with available measurements of the consumer was not successful as this was the case before.

Since the first model looks somewhat better, we could come back to our previous assumption, but modify the model in order to improve the curve fit attainable for the prey. What may be the reasons for the bad fit obtained? We may notice that Fig.1 is somewhat misleading, since it conceals the fact that all measurements were taken during a rather short period of the year only, namely while the insect is actively grazing larch foliage, i.e. during June and July. In winter, the population density of larch bud moths is much higher because every female has multiplied its number a hundredfold by producing offspring. Subsequently, during the following seasons, the population of the larch bud moths will decline until the next reproduction phase in autumn. Only the larval life stage which lasts about two months does live on the trees. The eggs, the pupae, and the moths are unable to cause any harm to the larch trees. Hence, in reality, there does not take place a continuous interaction over time between the larch tree and the insect, as it is assumed by the Lotka-Volterra model.

Moreover, we could include other known influencing factors into the model as well.

Fig.4 shows the variations found by taking measurements at different locations within the Upper Engadine Valley. Though there is still a significant trend identifiable, variations between population densities of 70 larvae per kg branches and of several hundreds of them make a lot of a difference! One could assume that the trend is a

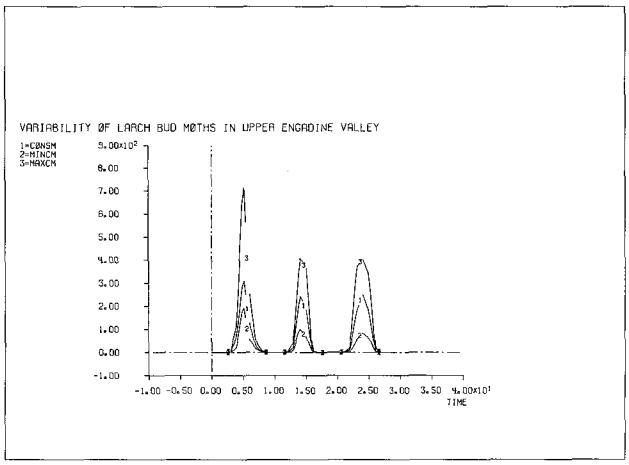

Fig.4: Variations in the population density of the larch bud moth as a function of the location within the Upper Engadine Valley (data taken from [2])

result of the (modeled) major influencing factor, whereas the variation results from other, non-identified, influencing factors of at least comparable importance. To reduce those variations, we could, for instance, divide the valley into several sectors, and model the effect of moth's migration between those different sectors. For this task, we require independent measurements of the involved species as a function of space and time. Such refined models have been postulated and identified as well. The results of those efforts have been published in [6,7].

This was to show how delicate the modeling of biological systems can be, even if appropriate measurements are available which show a large variability in the data as in our case. We must, therefore, be extremely cautious in the interpretation of modeling efforts. In fact, jumping to conclusions has, in the past, often misled decision makers to try to control ecological systems by introducing control variables into the real systems which seemed to have the desired effect in a simulation study, that is, changing the experimental conditions and assuming that the model still holds. Such experiments have mostly ended disasterously! (E.g. usage of pesticides in US agricultural production has increased twelvefold during the last 30 years while crop losses have doubled [15]! New problems arose like resistence of pests towards many pesticides, which e.g. has led to a dramatic come back of the so feared malaria.) We should have learnt by now that the "ecological equilibrium" is a very delicate matter which should not be willingly modified as long as we have not gained a much deeper understanding of the physical laws which rule the interrelationship between different species of the ecosystem. Currently, our efforts must, in most cases, be restricted to measure the variations in the "natural equilibrium" to <u>prevent</u> alterations to the "equilibrium" resulting from human activities of which we are not a priori aware, or to <u>minimize</u> the effects of unavoidable alterations (e.g. alterations resulting from urbanization).

3. SOCIAL MODELING

We have now discussed the difficulties, as they can be found in modeling biological systems. All these difficulties are also inherent in the modeling of social and economic systems. However, in this new field of application, again two additional complications can be identified:

a) If human beings play part of the system under investigation, there exists an unavoidable interaction between the system itself and the model representing that system. If a dentist asks his patient to demonstrate how he brushes his teeth at home, he is confronted with the difficulty that the patient is certainly brushing his teeth differently from the way he would do at home, be it for no other reason than that he is brushing his teeth consciouly whereas at home he would do it in a pure routine. We call this effect "keep-smiling effect".

b) There can be found indications that some state variables in social system models take subjective rather than objective values (although some scientists debate this statement). What is, for instance, the "value" of gold? For this reason, new methodologies (like cross-impact modeling) have sometimes been applied to the modeling of social systems, methodologies which should take this subjectiveness of some model components better into account than the previously used techniques (like system's dynamics).

All these four complicating difficulties make the usefullness of modeling efforts for social systems debatable. We must be especially careful in how to interpret results of such efforts.

In this light, we shall now discuss one of the best known, most discussed, and most controversial socio-economic models: Forrester's world model [8] which is often referred to as "WORLD1 model". ("WORLD2" by Meadows [14] has even become more popular, but we prefer to discuss "WORLD1" since the model is much simpler, and since all equations involved in the model have been fully presented in [8] which is not the case in [14].)

This system shows all the difficulties mentioned above:

a) The eigen-frequencies are very small, and data are accordingly difficult to obtain.

b) There exist certainly many competing influencing factors of equal importance, which are hardly separable since laboratory experiments cannot be executed.

c) The keep-smiling effect is not neglectable (the ongoing discussions of the effects of environmental pollution, as well as the effects of utilizing atomic energy have certainly been influenced to some extent by the modeling efforts of Forrester and of the Club of Rome).

d) Subjectiveness of parameters especially in the capital investment sector of the model leads to large sensitivities of the model behaviour with respect to those parameters, as has been shown in [18].

The model finding procedure is, therefore, very difficult, and the approach invented by Forrester must be considered genious. He first tried to identify the most important state variables in the world (levels) which he decided to be population, pollution, natural irrestorable resources, capital investment, and percentage of capital investment in the agricultural sector.

Some equations involving those state variables can be written down immediately. For instance, it is obvious that the first time derivative of the population must be equal to the difference between birth rate and death rate at any instant of time:

$$P' = BR - DR \ .$$

Other relations are less evident, and we must be glad if we can at least identify the influencing factors, though the physical law how these influences act on the relation is unknown. Forrester developed a special technique (system's dynamics) which allows to write down equations in a rather easy way. This shall be illustrated at the relation governing birth rate. The birth rate is assumed to depend on the material standard of living (MSL), the crowding ratio (CR), the food supply (FR), and the pollution (POL). Forrester assumed that each of these influencing factors have effects independently of each other, which can be described by variations from "normal" values (that is, from the values measured in 1970). The relation can, thus, be written as:

$$BR = BRN * f_1(MSL) * f_2(CR) * f_3(FR) * f_4(POL) \ .$$

BRN represents the "normal" birth rate, and each of the functions must take a value of 1.0 for its input being equal to "normal". The precise graph of each function can be found by comparing the different values of the output (birth rate) and of the influencing term at different places in the world, or during different years of this century. f_1, for example, which is depicted in Fig.5, has been found by comparing living standard and birth rate in industrial countries with those in the third world.

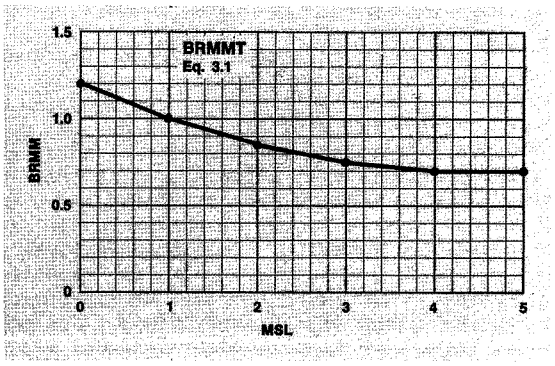

Fig.5: Birth rate as a function of the material standard of living

Finally, there exist some parameters which cannot be determined in this way since no statistical data were available. Those have been identified by curve fitting techniques as demonstrated in the previous example.

As we said before, the approach taken was genious since it allows to derive a model for a system which is otherwise extremely difficult to handle. However, there are so many assumptions hidden behind these equations that we must be ter-

ribly careful in how to interpret the results obtained by experimenting with this model. Forrester was so confident of his model that he succumbed to the temptation of drawing illegitimate conclusions not in one but in many instances in his book [8].

Some of the possible criticisms and shortcomings involve the model itself. Those are:

a) The model is of fifth order, whereas the world is certainly not a fifth order system. It is quite normal that the model of a system is of lower order than the system itself, but order reduction is only feasible as long as a sufficiently high degree of homomorphism is preserved.

b) The structure of the model has simply been preset, though other relations than linear regression would have been as easily defendable. The different influencing factors are, in reality, not independent of each other.

c) The reasoning behind the determination of influencing factors is debatable. The model suggests, for instance, that lowering the material standard of living in Europe would imply an immediate increase in the birth rate. There is, however, no evidence of that fact. It seems reasonable to assume that the high birth rate in India, for example, is (at least partly) motivated by the insufficient social security which, together with the existing family structures, makes it attractive to raise many children to guarantee that there shall be somebody who cares in the autumn of life. This is, of course, indirectly related to the material standard of living, but it is certainly incorrect to assume a simple functional relation between those two terms. It would be similarly plausible to correlate the decreasing birth rate in Switzerland with the continuously shrinking number of storks in our country to include in the model another influencing factor of the variable birth rate!

Other (more stringent) criticisms concern the designed experiments:

d) For such weak relations between different variables, as we face them in socio-economic models, we must be glad if we can use simulation to <u>interpolate</u> within the range of measured variations of all variables. Given two different (real-life) experiments, it may be acceptable to believe in a simulated curve representing the behaviour of the system if the obtained trajectories lie somewhere between the two real experiments. It is, however, something entirely different to <u>extrapolate</u> beyond the range of our experiments. For construction of the WORLD1 model, measured values of pollution have been used which show some dynamic behaviour over the last decades. However, the simulated experiments show a behaviour where the pollution (in the future) takes values being 100 to 1000 times larger than any value available from measurements. It requires a good amount of self-assurance to assume that the model shall produce valid trajectories for such values of the pollution as well!

e) The limiting factor in WORLD1 is the depletion of natural irrestorable resources. When this state has reached a level of about 0.4E12, the population starts decreasing. This result in mind, Forrester suggested to reduce the amount of usage of those resources to 25% starting from 1970. Although Forrester does not give any recipe on how to achieve this without influencing the other control variables of the system (like production), this assumption seems reasonable since its effects are almost equivalent to detecting new resources (like new oil fields). Under these modified conditions, the limiting factor is no longer resource depletion. Before this could happen again, the pollution has increased to such a high level, that (around 2030) a "pollution shock" takes place which reduces the population from about 5.6 billion people in 2030 to 800 million people in 2060. To avoid this disaster, Forrester suggested to reduce also the pollution rate to 25% starting from 1970. Under those new assumptions, he obtained wonderful state trajectories in which the population increases smoothly and, finally, levels off at about 8 billion people shortly prior to the year 2100 when the simulation ends. However, to our understanding of system theory, a steady state is reached when <u>all</u> (and not just one) state derivatives (rates) have become zero! Simulating further from 2100 to 2500, the natural resources are again depleted (due to the many consumers), and the population decreases smoothly to approximately the same value of 2 billion people as in the original case. Due to the reduced pollution, again resource depletion takes preference over the pollution shock, and the model behaviour is very similar to the original case, except that everything happens some decades later.

After all these depressing comments, we must ask ourselves, what conclusions can be drawn from such modeling efforts, or whether there are eventually none. A complete sensitivity analysis of WORLD1 has been carried out and published [18]. The authors found that, if some of the parameters are modified by just 2%, entirely different state trajectories result which do, for instance, no longer predict a pollution shock. So, their conclusion was that either the model is correct (and, thus, also the sensitivities as they show), in which case the model may not be used to predict the behaviour of the world, or the model is incorrect, in which case we know precisely as much as before!

However, we feel that some conclusions may nevertheless be drawn:

a) No model whatsoever shall predict unlimited growth. From this we can conclude that there exists a limit to growth, though we cannot predict where precisely this shall be. A continued increase of neither population nor welfare can be assumed to take place for long. Although we might not have required a computerized model to prove this truism, it is nevertheless nice to have one! More generally spoken, if we detect modes of behaviour in a model which are insensitive to parameter and structural variations, these modes can be believed to be correct. Hence, sensitivity analysis can, indeed, tell us a lot about model validity.

b) WORLD1 suggests that unlimited pollution is much worse than resource depletion in its ef-

fects to the dynamics of the world. Although we cannot be sure that this conclusion really holds (e.g. due to criticism (d)), we should not bear the risk to execute a real-life experiment to determine whether this conclusion was correct or incorrect! Immediate action should be taken to prevent (or at least reduce the risk of) a pollution shock to happen. From smaller real-life experiments we know that ecosystems tend to fall suddenly into very stable, undesired states (e.g. lakes getting eutrophicated or agricultural crop production depending entirely on an ever increasing pesticide usage). This should better not happen to the world ecology as a whole. Free competition (capitalism) will not solve this problem since each competitor has to maximize his own (and not the global) profit to survive in the competition. Monopolism (as nowadays communism), on the other hand, does not promise an attractive alternative either since there the regulating control loops are broken. A possible contribution to find a way out may be by introducing penalty functions, a technique well known from constraint optimization. Full freedom of decision is still preserved, but unfavourite decisions are punished. (If a company produces plastic bottles which are not collected for reuse, it has to pay an "environmental tax". If this tax happens to be sufficiently high, the production and recycling of glass bottles becomes more profitable again. If oil tankers which are caught while cleaning their tanks in the open sea have to pay a sufficiently high fine, the risk of being caught does no longer justify the attainable money savings.) By these means, the control loops of a free competition are not broken while granting the government a possibility to intervene. If there is only a certain realistic chance, and we believe there is, that such a relatively cheap and easily implementable measure may diminish the risks of a pollution shock to happen, it is worthwhile considering.

4. PSYCHOLOGICAL MODELING

Psychological models, dealing with the description of the behaviour of human beings, are a subclass of the previously discussed social models. In those systems, the first difficulty (small eigen-frequencies) cannot be observed, whereas the other three complicating factors (number of competing inputs, keep-smiling effect, subjectiveness) become the dominating factors of the system!

Many articles have been published in recent years on this exciting subject. Again, we shall concentrate on one particular study which shall serve for illustration. This analysis [9] discusses the effect of conformity of individuals to (prepared) groups when those individuals are confronted with the conflict to choose between an answer to a given problem in which they are conform with the unanimous meaning of a group though they must believe this answer to be incorrect versus an answer in which they believe but with which they are in opposition to the whole group.

A major difficulty lies in the fact that meaningful measurement units for social behaviour cannot easily be determined. Only under laboratory conditions, behaviour can be expressed in a numerical manner by exposing many individuals to the same problem and counting the percentage of people acting in the same way as compared to those acting differently, or, which is equivalent, computing the probability of an individual to act in a prescribed manner. Again, the way of behaviour depends on many different influencing factors, like current mood, alertness, sympathy (or antipathy) with members of the group, personality, education, and many more. To solve this problem, the laboratory conditions must be kept very rigorously with the obvious implication that results of such modeling efforts are not easily expandable to other situations outside the precise experimental conditions. This is evident since we can never be sure that there do not exist other equally important influencing factors which have not been identified in the experiment since they were kept constant (which is most likely to be the case!).

Hirsig [9] made use of the previously published experiments by Asch [1] and Cohen [5] in which the length of a straight line had to be determined by comparing it to a set of three different reference lines. The problem was posed such that, under "neutral" conditions, 95% of individuals should be able to solve the posed problems without difficulties. In this experiment, groups of prepared persons were asked to answer the problem incorrectly prior to letting the test person express his opinion in public. The test person was, thus, in conflict whether he should oppose to the unanimous meaning of the group as a whole or go conform with the group. Hirsig felt that this experiment was disturbed by too many uncontrollable factors (like sympathy of the test person with one particular member of the group, and the necessity to replace the members of the group between test persons). He, therefore, modified the experiment in that he replaced the group members by a CAI (computer-aided instruction) program to exclude other than the desired (controllable) influencing factors. In this way, all test persons faced equal conditions. The important effect of "public" exposure was achieved by letting the test persons believe that the opinion of the "group" (as it was displayed to him at his terminal) represented the meaning of other participants sitting at different terminals at the same time which would in reverse get his opinion displayed at their terminals. In this way, each of the 40 test persons had to solve a series of 20 similar experiments (of which he believed that they served to test his visual capabilities).

Hirsig now assumed that the dynamics of the process can be expressed by a discrete-time model:

$$\underline{x}(k+1) = \underline{f}(\underline{x}(k),\underline{u}(k),\underline{w}(k))$$

where k denotes the index of the experiment in the series, $\underline{x}$ is a set of observable state variables (e.g. $x_1(k)$ could denote the probability of a test person to be in agreement with the group during the k-th experiment), $\underline{u}$ is a set of controllable influencing factors, and $\underline{w}$ is another set of influencing factors which are observable, but not

controllable.

Contrary to Forrester's approach, Hirsig did not try to determine the structure of his model. He just assumed that the functional relations $\underline{f}$ are expressable by Taylor series expansions as polynomials of infinite order in $\underline{x}$, $\underline{u}$, and $\underline{w}$. In his model, he then approximated $\underline{f}$ by neglecting higher order terms than n, where n was determined during the identification by the quality of the obtained curve fit. This approach seems extremely useful for such systems since the only decision which must be taken beforehand concerns the number of state variables to be included, and their meaning. For all state variables, measurements are required which makes this choice extremely simple since we are anyway in most cases short of measurements. As in the case of the world model, the modeling efforts taken look very promising. Again, there are just the experiments and the conclusions drawn by Hirsig which are debatable (though Hirsig was much more careful in this respect as compared to Forrester).

We shall try to identify some of those illegitimate conclusions. One of the aims of the modeling effort was to determine whether there exists a time dependency in the behaviour of the test persons. Previous experiments had led to contradictory conclusions in this respect. To decide this question, Hirsig used once the model:

$$\underline{x}(k+1) = \underline{f}(\underline{x}(k),\underline{u}(k))$$

and once the model:

$$\underline{x}(k+1) = \underline{f}(\underline{x}(k),\underline{u}(k),k)$$

to fit the same streams of experimental data. $\underline{f}$ was approximated in both cases by second order polynomials. k describes the time dependency. (k is here used as an observable but not controllable input). For identification, Hirsig tried to minimize an ISE criterion. In the former case, he obtained a performance index of 743 units, whereas the index was reduced to 589 units in the latter case. Therefore, Hirsig concluded that the time dependency had been shown. In reality, since the model describes the system only approximately, we even must expect to obtain a better curve fit when introducing additional parameters into the model. A smaller value of the performance index cannot be taken as a proof that a time dependency exists, just that the model approximates the system somewhat better! To draw further conclusions, we would have to show that the improvement of the fit is significant. During the whole study, Hirsig operates on mean values only. He seems not to have realized that also mean values have their variances which must be remarkably high since only 40 test persons have been doing the experiment. This means that the better curve fit which was achieved by applying the second model cannot be taken as a proof for time dependency. To obtain better results, statistical tests should be carried out to determine the significance of the performance index reduction obtained. Beside of such statistical tests, again sensitivity analysis may be used to discuss the validity of the model.

Hirsig tried to identify five different inputs (size of group, percentage of group answering incorrectly, percentage of group with a higher "social" status, percentage of group with a lower "social" status, percentage of group with a higher "professional" status (like geometers, or art teachers)). Those inputs were varied stochastically in the sequence of experiments to observe the system's reaction. (This is a well known technique in identification). However, it seems to us that a series of 20 experiments may not suffice to identify the effects of those five inputs. Moreover, if the extreme situation (like entire group answering unanimously during the whole series) has not been measured, we cannot expect the model to be valid for such an experiment.

Let us assume for the moment that modified experiments including many more test persons have answered all the questions we may dream off. How can these results prove useful? All test persons have (on purpose) been selected from the same class of people (students during their first or second term). How can the results then be expanded to other classes of people (like workers, house wifes, or university professors) which presumably may react in a different way? How can these results be applied to other situations than selecting one stick out of three, situations which are more likely to happen in real life?

These (unanswered) questions may show that we are still far from useful applications of modeling efforts in psychological systems.

5. GENERAL REMARKS TO MODELING

Karplus [11] has once presented a "rain bow" (Fig.6) in which the relation between different types of models for different fields of applications were summarized.

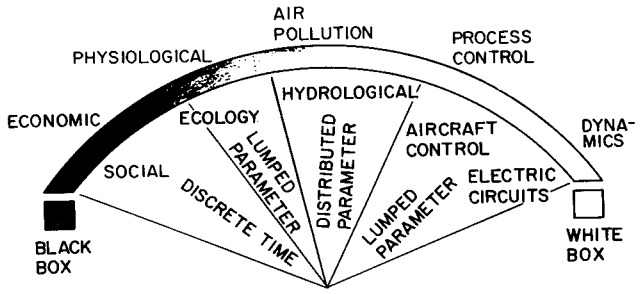

Fig.6: Model types versus system types

He mentioned in his talk that the aim of modeling efforts in the different topics cannot be same. Models at the right end of the bow (white box systems) can be used for system design, models somewhat more to the left may be used for system prediction, again to the left models can be used for a qualitative analysis to gain a better understanding, whereas models to the very left end of the bow (black box systems) may be used to raise public opinion! With this remark, Karplus had certainly efforts like WORLD1 in mind.

We can add an additional dimension to the rain bow: As one can see, there is really a lot of commonality in the models for different system types. Given a general purpose simulation program being able to handle all types of models (partial differential equations, ordinary differential equations, difference equations, discrete events), as

there exist [3,4], such a program may be used for models in all fields of application. However, at least the output module of that program should "realize" the well/ill-definition of the problem, and produce 14 digits when applied to white box systems, two or three digits when applied to gray box systems, and zero digits when applied to black box systems (that is, a standard message could be displayed stating that the available data were so unreliable and irreproducible that no output whatsoever makes any sense)! Confronted with ill-defined models, we must learn that out of 14 digits displayed to the user by the computer (out of which none is zero), as many as 14 may be insignificant.

This final conclusion sounds very pessimistic indeed. However, in the case of the biological (gray box) model we were able to produce quite useful results, although the data acquisition effort was tremendous). Even in the case of WORLD1, which is certainly a black box model, we could draw some useful (if not vital!) conclusions.

We have shown that with increased weakness of the models, there goes hand in hand an increased difficulty in model validation, and an increased danger of drawing illegitimate conclusions. Software which could help in the validation and interpretation process itself would, therefore, be an enormous improvement over the current state. The second part of this paper shall now discuss to which extent the currently available simulation software can be enhanced to meet these requirements, that is, to make simulation software more suitable for ill-defined system modeling.

6. INTERACTIVENESS

The model finding procedure is, in the case of ill-defined systems, not a straight forward matter. Usually, one has to postulate several models and reject them again, until one finds a model which shows satisfactory results. Even if a model has been determined which looks promising, usually several steps of refinement are still required to make it finally appropriate. In the last step, parameters have to be adjusted in order to fit some measured curves. Only at this stage of the modeling process, the modeler can start to experiment with his model. To do all this in an efficient way, the simulation software should run interactively to allow easy manipulations of models and data from a terminal.

Most of the currently available CSSL's run in a batch mode only. This is partly due to historical reasons (interactive programming became widely used only with the newer generation of minicomputers and intelligent terminals). However, also another constraint to the interactiveness of simulation software may be mentioned. Even the available simulation systems are, in general, quite large. As we shall see, the additional requirements to make this software appropriate for ill-defined system modeling will demand even larger software systems to be coded. Highly interactive programs have to run on minicomputers since most of the multi-user and multi-tasking main frames (computing centers) do not allow the single user to access the system in a highly interactive mode. Unfortunately, there is a certain limit to the size of programs which can be efficiently implemented on computers with a very restricted word length (most minis are 16 bit machines). Only the very recent new generation of 32 bit process computers (e.g. VAX-780) shall allow an efficient implementation of the type of software we advertize in this paper. Such computers are available now for slightly more than a year, and it can be expected that appropriate software tools shall become available soon.

One of the best interactive simulation systems available of today is DARE-ELEVEN [12]. Fig.7 shows the different levels of user-program interaction.

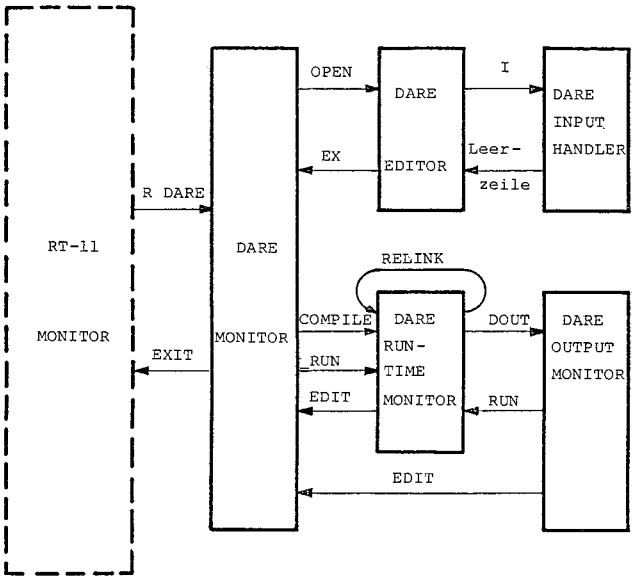

Fig.7: Structure of the interactive DARE-ELEVEN simulation system

On the MONITOR level, the user can access a set of (about 20) commands mostly for file handling purposes (storage and retrieval of problem files). Each problem consists of a series of different files for the description of models, experiments, output descriptions, data, FORTRAN subroutines, assembly macros, and tabular functions. The EDITOR level allows to manipulate data within files (equations are also considered to be "data" in this context). Once a problem has been fully prepared, it can be compiled. The RUN-TIME MONITOR shall then detect whether any data (constants) are missing, and prompts the user to supply those data which it, subsequently, collects into a data file. Now, the user can execute simulation experiments which have been described in an experiment description (LOGIC) block. Each experiment may involve many simulation runs to be executed (e.g. an entire optimization study). Between experiments, the user may modify model parameters, or replace the integration rule to be used. During simulation runs, the user may watch simulation trajectories plotted "on-line" on a graphical terminal. During execution of experiments, data can also be stored for later retrieval by the OUTPUT MONITOR. This final level of user-program communication allows to specify many different output commands or to execute previously coded output description blocks to look at simulation results in more detail than

this is possible by use of the run-time display.

DARE-ELEVEN is available for PDP 11 systems with VT-11 graphic terminals (GT 40 system) running under either DOS/BATCH or RT/11. Due to the restrictions of 16 bit machines mentioned above, DARE-ELEVEN is not very well suited for large scale problems. Furthermore, no attempt has been made to make DARE-ELEVEN at least partly portable. Finally, the test status of this software is still not satisfactory which makes it an educational rather than a production tool.

7. FILE HANDLING

As we have previously shown, files need to be manipulated. It must be possible to store models for later reuse, retrieve previously stored models, and execute some operations on those models. For this purpose, we require a data-base management system. In the future, it may be possible to execute some analytical operations on models other than compilation, like model linearisation, model reduction, model decomposition, and others more. Such features would be very valuable especially for large scale systems (which ill-defined systems often are). Currently, research in this field has just started. When this shall become available, data-base management will be even more important than today.

Beside of model files, there exist also data files, that is, files in which data are collected during the execution of simulation experiments. It is important to realize that not all simulation trajectories need to be printed or plotted while the simulation is running, as this is implemented in most of the available CSSL's. In fact, the run-time display is rather meant to give the user an oportunity to watch that everything goes as expected. It is much more effective to store data during the simulation runs into data files, and use an independent program (which may easily be as large or even larger than the run-time system) to retrieve data from those data files lateron to produce the required graphical data representations. The best file structure available has been implemented in DARE-P [12]. There exist four different types of data files. During simulation runs, data can be stored either on the TIME-file or on the SAVE-file. Data stored on the TIME-file are overwritten during each new simulation run, whereas a new logical record is opened for each run on the SAVE-file. The output variables are stored on the CROSS-file each time the user explicitely asks for. This type of file is mostly used to store simulation parameters between different simulation runs, e.g. to obtain a graphical representation of the performance index of an optimization study as a function of optimization parameters. The fourth file (STASH-file), finally, is not used during the execution of simulation experiments, but may be addressed in the output module to "stash" data streams away for later reuse, e.g. in another problem. This data philosophy is also used in GASP-V [3] and in COSY [4].

One of the reasons to use the STASH-file is to compare measured and simulated data. In a first "run", measured data are read from a file (in any format available), and a dummy simulation study is carried out in which those data are interpolated and stored on the TIME-file at each communication instant. In the output module, the TIME-file is then stashed. In a second (true) simulation run, the model is used to produce data streams of simulated variables which are again stored on the TIME-file. In the output module, we can now plot the simulated variable stored on the TIME-file together with the measured variable retrieved from the STASH-file versus the simulation clock by specifying:

```
PLOT X, X(S-1) .
```

It would be very valuable (although this is not implemented in DARE-P) if we could use measured and previously stashed input variables as inputs to the simulation model as well. For this task, we would require to access data on the STASH-file at run-time, either by specifying:

```
U = STASH (UMEAS, 1)
```

in the equation section of the model, or by expressing something like:

```
USE UMEAS(S-1) AS U
```

in the experiment description block of the problem. The CSSL-type function table is only a very primitive replacement for such a feature.

8. OPTIMIZATION AND IDENTIFICATION

One of the most common operations in the model finding procedure, is curve fitting. This was done in all three of the presented examples. It is, therefore, astonishing that so far only very few simulation systems have a nonlinear programming package associated with them. Most CSSL's allow the user to code his own optimization strategy (e.g. CSMP allows this by letting the user "CALL RERUN" from within the TERMINAL-section of the model), but coding an intelligent optimization strategy for multi-dimensional optimization subject to sets of equality and inequality constraints is a task maybe as difficult as coding a numerical integration algorithm for stiff systems. It is, therefore, bare nonsence to let the user do this. Hence, it is very important that future simulation systems offer such a facility as a standard feature. The structuring capabilities implemented in most CSSL's (INITIAL, DYNAMIC, TERMINAL section) is, unfortunately, not at all appropriate for this enhancement. To implement this feature efficiently, we require an experiment description block (like the LOGIC-block of DARE-ELEVEN or DARE-P, or like the EXPERIMENT block of COSY).

In COSY [4] we can, for instance, express a curve fitting experiment by writing:

```
EXPERIMENT
  IDENTIFY
    LOTKAMODEL: PAR1:=0.0, PAR2:=1.0; TOL=1.0E-4
                END;
    FITTING X TO XMEAS(STASHF[1]);
    FITTING Y TO YMEAS(STASHF[1]);
    FITTING Z TO ZMEAS(STASHF[1]);
    USING UMEAS(STASHF[1]) AS U;
    USING VMEAS(STASHF[1]) AS V;
    SIMULATE FROM 0.0 TO 30.0 COMINT=0.5
  END  (* IDENTIFY *)
END  (* EXPERIMENT *);
```

in which case the two parameters PAR1 and PAR2 are modified in order to minimize an ISE criterion:

$$PI = \int_{0.0}^{30.0}((x-\hat{x})^2 + (y-\hat{y})^2 + (z-\hat{z})^2)dt = \text{min!}$$

by utilizing the conjugate direction method of Fletcher for the unconstraint multi-dimensional optimization study together with the golden section algorithm for the uni-dimensional search. The starting values for the two parameters are zero and one, resp.. The model is fed by the two measured input signals UMEAS and VMEAS, referred to as U and V in the model.

If the standard algorithm fails to converge, or if a more general optimization study has to be carried out, the user can write:

```
EXPERIMENT
  MINIMIZE  (* OR MAXIMIZE *)
    PERFINDEX: A:=1.0, B:=0.0, C:=0.0; TOL=1.0E-5
               END;
    EQUCONST  A*A + B*B + C*C = 1.0 END;
    INEQUCONST  A>=0.0; B>=0.0; C>=0.0 END;
    OPTMETHOD CONSTR = EXTPENAL; MULTIDIM
              = DAVIDON; UNIDIM = CUBIC END;
    SIMULATE FROM 0.0 TO FINISH COMINT = 0.1 END;
    PERFINDEX := PICOMP
  END  (* OPTIMIZATION *)
END  (* EXPERIMENT *);
```

which would minimize the performance index (PICOMP) computed somewhere in the model description section by modifying the three parameters subject to one equality constraint and three inequality constraints. The uni-dimensional search is now done by using cubic interpolation, the multi-dimensional optimization is done by using the algorithm of Davidon-Fletcher-Powell, whereas constraints are handled by exterior penalty functions. The nonlinear programming package associated with COSY has been described in [16,17].

It is evident that the user is not relieved from thinking by the introduction of such features into a simulation language. For instance, all variable transformations which were required to identify the biological system discussed previously must be done by the user as before.

9. SENSITIVITY ANALYSIS

After a model has been determined, we must discuss its validity. As the presented examples have shown, the sensitivity of the model behaviour with respect to changes in the parameter values can tell us something about model validity. Let us look once more at the Lotka-Volterra model:

$$\dot{x}_1 = a*x_1 - b*x_1*x_2 \; ; \quad x_1(t_0) = x_{10}$$
$$\dot{x}_2 = d*x_1*x_2 - c*x_2 \; ; \quad x_2(t_0) = x_{20} \; .$$

We can compute the sensitivity of the two state variables with respect to any parameter (e.g. a) by differentiating the state equations with respect to this parameter, for instance:

$$\frac{d\dot{x}_1}{da} = x_1 + a*\frac{dx_1}{da} - b*\frac{dx_1}{da}*x_2 - b*x_1*\frac{dx_2}{da} \; ;$$

$$\frac{dx_1}{da}(t_0) = 0.0$$

which can be written as:

$$\dot{v}_{1a} = x_1 + a*v_{1a} - b*v_{1a}*x_2 - b*x_1*v_{2a} \; ;$$
$$v_{1a}(t_0) = 0.0$$

where:

$$v_{1a} = dx_1/da \; ; \quad v_{2a} = dx_2/da \; .$$

Computing the sensitivity model, thus, means to increase the number of differential equations from n to n*(k+1) where n is the order of the system (number of differential equations), and k denotes the number of parameters. (In the case of linear systems, the number of additional differential equations can be reduced.)

Although the sensitivities could also be computed numerically at run-time by computing the n-th order model k+1 times for each integration step while modifying each parameter by a small amount. The sensitivities can then be computed as:

$$\frac{dx_1}{da} = \frac{x_1(a+\Delta a) - x_1(a)}{\Delta a} \; .$$

However, it seems a promising approach to compute the sensitivity model once for ever analytically at compile time, and to simulate the enlarged n*(k+1)-th order model. This is not so unfeasible as it may look at a first glance. The COSY preprocessor, for instance, generates analytically the Jacobian matrix of any given model during compilation. This matrix is required by many integration algorithms, and the achievable savings in computing time are remarkable. This program bases on an algorithm by Joss for <u>algorithmic differentiation</u>. The procedure is best explained at an example. Let us compute the derivative of:

$$f(x) = x*\sin(x^2)$$

with respect to x. This expression is first separated into its primitives:

$$t_1 = x^2 \; ; \quad t_2 = \sin(t_1) \; ; \quad f(x) = x*t_2 \; .$$

Now, each primitive is handled independently by expanding each of the primitive equations to:

$$dt_1 = 2*x \; ; \; dt_2 = \cos(t_1)*dt_1 \; ;$$
$$f'(x) = t_2 + x*dt_2 \; .$$

The original program developed by Joss was ALGOL coded, and was able to compute the derivative of any ALGOL procedure with respect to any variable or array of variables, resulting in another (expanded) ALGOL procedure. A modified version has been coded meanwhile, in which a PASCAL coded program computes the derivative of any FORTRAN subroutine/function or even of sequences of them with respect to any variable or array of variables, resulting in another (expanded) FORTRAN subroutine. A specialized version of this program forms part of the COSY preprocessor, and computes the Jacobian matrix ($d\underline{x}/d\underline{x}$) of any model. Again, another specialized version of this program could easily compute the sensitivity model. Another version could be used for model linearization along any trajectory, and a final version could compute the sensitivity model of this linearized model (by taking advantage of the linearity of the model). All these operations on models should be executable by MONITOR commands.

The same sensitivity model could, by the way, also be used during optimization, since many optimization algorithms require the gradients of the performance index which can be expressed in terms of the state variables of the sensitivity model.

Very often, we are most interested to know the largest sensitivities in the system. For this purpose, we can add an array of state conditions of the form:

```
CONDIT MAXSENS[1..K]: VDOT[1..K] CROSSES 0.0
       TOL=1.E-4 END;
```

to the model description. This would then determine those instants of time when the sensitivity derivatives cross through zero (corresponding to maxima and minima of the sensitivities themselves).

10. SENSITIVITIES IN THE LARGE

A sensitivity analysis, as we have discussed it so far, is justified by the assumption, that we can write:

$$x(a+\Delta a) = x(a) + v_{1a}(a)*\Delta a \ .$$

This assumption is not necessarily true in the case of nonlinear models and for a large variablity of parameter values.

Let us assume that the parameter a has a nominal value of 100.0, but may vary between 80.0 and 120.0. The notation:

```
PARAMETER A = (100.0 MIN=80.0 MAX=120.0);
```

would lead to the computation of three different simulation runs, once with the nominal value of a=100.0 to obtain a nominal trajectory, and twice with the two extreme values a=80.0 and a=120.0 to obtain a band in which each state variable is expected to be (although this need not be correct either since we cannot assure that the worst case is at the boundaries). If ranges are specified for several of the parameters, one simulation run is executed for each combination of maximum variations, resulting in 2^k+1 simulation runs where k denotes the number of considered parameters. For each output variable we then store its nominal trajectory and the maximum positive and negative variations from this trajectory (envelope). These are then plotted together with the nominal trajectory.

This procedure is quite expensive since the computational effort grows exponentially in the number of considered parameters. For this reason, it is recommended to compute the previously discussed sensitivity model first to determine those parameters which show the largest sensitivities, and to compute the here proposed bands only for a limited number of not more than five or six parameters.

11. RANGE TESTS

We have explained that models are never valid as such, but only under given experimental conditions. In the discussion of the WORLD1 model, we have shown how dangerous it can be to extrapolate beyond available measurements. For this reason, it seems important that the user may specify ranges for state and algebraic variables, e.g. by coding:

```
TYPE
  STATESUBRANGE = STATE  0.0 .. 1.0E10;
VAR
  STATESUBRANGE  POL;
```

Pollution will then be checked versus the specified ranges during the execution of each simulation run. This is accomplished by generating additional state conditions with associated precoded state events at compile time. Each time, one of the variables leaves its allowed range, the user is informed of this fact by an informative message.

More complicated range tests can, of course, be user handled by coding appropriate state conditions (using the CONDIT-statement), in which case corresponding state event must also be user coded.

Previous papers have also suggested the introduction of range tests, e.g. for automated scaling of simulation programs which are to execute on hybrid hardware, or to prevent variables from taking impossible values (like negative population). However, the additional justification, mentioned here, seems to be very important for improvement of the robustness of models.

12. LIMITS TO CURVE FITTING

In the discussion of the psychological model, we have seen that an improved curve fit does not necessarily imply an improved model. Given a trajectory band as discussed in section X, it is obvious that any trajectory within that range is equally acceptable, and we cannot prefer one over the other. As a solution to the problem of Hirsig, we would recommend to compute the standard deviations for all measured curves, and draw similar ranges around the measured curves within which each curve fit must be considered equal. We can then modify the performance index in that we do no longer compute the error as the difference between the simulated and the measured trajectory, but as the distance of the simulated trajectory from the

closer of the two boundaries of the measured curve. If the simulated trajectory lies within the band, the error is to be taken equal to zero. This modified performance index can be minimized in precisely the same way as before, resulting in a value which may be larger or even equal to zero. Let us assume that the value of the performance index with inclusion of the time dependency will still be smaller than without time dependency. What would this imply? It would even now be too early to argue that time dependency has been shown since the model is still an approximation of the real system only, in that polynomial terms higher than second order have been neglected. It could well be that, including those higher order terms as well, no difference would result between the two considered models any longer. However, such a modified performance index shall certainly allow for better judgements of the obtained results.

13. STOCHASTIC MODELS

Most of the available CSSL's offer facilities to draw samples from pseudo random number generators, and to generate uniformally and Gaussian distributed random number streams. This allows to formally describe stochastic models. Especially in the case of the psychological system, a stochastic model might have given more promising results that the deterministic model being used by Hirsig. Although this feature introduces additional numerical difficulties (like failure of the step size control of the numerical integration algorithms), such models may be extremely useful. However, it is not done with a pure simulation of such stochastic models. Additional software features should be provided to collect statistical information, apply significance tests to statistical data, and other more. Such features are currently available in many of the discrete event software programs, whereas they are almost constantly missing in CSSL's.

14. CONCLUSIONS

Although we are still far from really convincing application of ill-defined system modeling, enhancements of the available simulation software may pave the way for a better applicability of modeling techniques to the analysis of ill-defined systems in the future.

REFERENCES

1. ASCH, S. E., Social Psychology, New York, 1952.
2. AUER, C., "Dynamik von Laerchenwickler-Populationen laengs des Alpenbogens," Mitteilungen der Eidgenoessischen Anstalt fuer forstwirtschaftliches Versuchswesen, Vol.53, pp. 71 - 105, 1977.
3. CELLIER F. E., and A. E. BLITZ, "GASP-V: A Universal Simulation Package," Proceedings of the 8th AICA Congress on Simulation of Systems, L. Dekker (ed.), North-Holland Publishing Company, pp. 391 - 402, 1976.
4. CELLIER F. E., and A. P. BONGULIELMI, "The COSY Simulation Language," Proceedings of the 9th IMACS Congress on Simulation of Systems, L. Dekker, G. Savastano, G. C. Vansteenkiste (eds.), North-Holland Publishing Company, 1979.
5. COHEN B. P., Conflict and Conformity: A Probability Model and its Application, Cambridge (Massachusetts).
6. FISCHLIN A., "Systems Analysis of the Larch Bud Moth System: Part II: Spatial Dynamics," in prep.
7. FISCHLIN A., and W. BALTENSWEILER, "Systems Analysis of the Larch Bud Moth System. Part 1: The Larch - Larch Bud Moth Relationship," Mitteilungen der Schweizerischen Entomologischen Gesellschaft, Vol. 52, pp. 272 - 289, 1979.
8. FORRESTER J. W., World Dynamics, Wright-Allen Press, 1971.
9. HIRSIG R., Menschliches Konformitaetsverhalten - am Computer simuliert, Birkhaeuser Verlag, Interdisciplinary Systems Research (ISR 1), 1973.
10. JOSS J., "Algorithmisches Differentieren"," Ph.D. Thesis, ETH Zurich, Diss. ETH 5757, 1976.
11. KARPLUS W.J., "The Spectrum of Mathematical Modeling and Systems Simulation," Proceedings of the 8th AICA Congress on Simulation of Systems, L. Dekker (ed.), North-Holland Publishing Company, pp. 5 - 13, 1976.
12. KORN G. A., and J. V. WAIT, Digital Continuous-System Simulation, Prentice Hall, 1978.
13. LEKKAS G., D. F. RUFER, W. SCHAUFELBERGER, "Identifikation dynamischer Systeme durch nichtlinearen Modellabgleich," Internal Report, AIE 77-04, Institute for Automatic Control, The Swiss Federal Institute of Technology Zurich, ETH Zentrum, CH-8092 Zurich, Switzerland, 1977.
14. MEADOWS D., The Limits to Growth, Universe Books, New York, 1972.
15. PIMENTEL D., et al., "Pesticides, Insects in Food, and Cosmetic Standards," Bioscience Vol. 27, pp. 178-185, 1977.
16. RUFER D. F., "General Purpose Nonlinear Programming Package," Proceedings of the 8th IFIP Conference on Optimization Techniques, Wuerzburg, FRG, Springer Verlag, Lecture Notes in Control and Information Science, Vol. 7, Part 2, pp. 495 - 506, 1978.
17. RUFER D. F., "User's Guide for NLP: A Subroutine Package to Solve Nonlinear Optimization Problems". Internal Report: AIE 78-07, Institute for Automatic Control, The Swiss Federal Institute of Technology Zurich, ETH - Zentrum, CH-8092 Zurich, Switzerland, 1978.
18. VERMEULEN P. J., and D. C. J. de JONGH, "Dynamics of Growth in a Finite World - A Comprehensive Sensitivity Analysis," Proceedings of the Symposium on Large Scale Systems Theory and Applications, Instrument Society of America, 400 Stanwix Street, Pittsburgh PA 15222, U.S.A. Editors: G. Guardabassi, A. Locatelli.