Dedicated to
Professor Jakub Gutenbaum
on his 70th birthday

Control and Cybernetics

vol. **29** (2000) No. 1

Optimization of operative decisions; computer analysis

by

K. Malinowski

Warsaw University of Technology,
Institute of Control and Computation Engineering,
e-mail: k.malinowski@ia.pw.edu.pl

Abstract: When entering the age of ubiquitous computing we observe a rapidly increasing number of applications in which – in real time – operative (control or management) decisions are made based on optimization. To implement repetitive optimization of these decisions we need to use process models, models or directly available forecasts of free, uncontrolled, inputs, methods for state and parameter estimation, model adaptation techniques, and, finally, suitable optimization techniques. Analysis and validation of such optimization based systems require – prior to actual real-life implementation – the carefully planned and performed computer experiments. Theoretical methods are always welcome to assist in this analysis – if possible – but it is mainly the computer based simulation experiment which allows us to investigate, verify and tune a complicated decision mechanism and then control system as a whole before this system is authorized to be put to real-life test. Success in design of a system with on-line computed optimization-based operative decisions is therefore dependent upon many factors, in particular – on having good models, efficient and reliable optimization techniques, good software tools for computer based simulation and on proper, real-life-like conditions under which the computer experiments are performed. These key issues concerning development of such control systems are discussed in this paper.

Keywords: operative control, management, operative decisions, forecast, optimization, computer simulation, computer analysis.

1. Introduction

One could perhaps feel puzzled when being asked the following question: what can be common, what in fact is common, to such problems as: launching a multistage rocket so as to place a satellite on an orbit, management of water releases from a system of reservoirs during flood, defending an important object against aerial attack, introducing a new product – perhaps a new motion picture – on the market? What these activities have in common with month by month or day by day operations, under changing external conditions, of a division of a refinery or of a group of petrol stations? It appears that in all these, and in many, many other situations it is necessary to conceive, usually repetitively, operative decisions, which are needed to achieve the possibly best results – there is a need to introduce optimization of these decisions. The reasons for this are many, but first of all it is the computer revolution, which both makes possible and forces us to use optimization within operative decision making. The rapid growth of computer technology allows to transmit, store, rapidly access and process vast amounts of data. Available information, as measured by various methods – like the number of bytes, doubles every few years (every seven years according to Cross, 1997, and even more frequently according to others). In production and marketing situations one faces ever-growing competition and ever increasing entropy (disorder). Fast and smart decisions are needed to avoid commercial disasters and the competitors are thus very likely to use optimization of their decisions. Hence, one is willing or even forced to use any relevant information available in the best possible way. In this new reality the basic assumption about a control system, the basic control paradigm, is changed. For many years it was assumed that an operative control law or a management rule had to be simple, similar to PI or PID control algorithms, or based upon another parametric decision rule; more recently, perhaps, on the use of a small artificial neural network. Yet, now there is both a need and the means to use much more advanced techniques to make operative decisions, for example decisions concerning launching of a rocket, protective measures during flood, actions – like weapon assignment and commitment – during air defense or decisions regarding next day prices of numerous products in a megastore. The use of such techniques, techniques of optimization and system analysis, has for many years been perceived and advocated by many researchers, in particular by Professor Jakub Gutenbaum (Gutenbaum, 1988). It is only recently, however, that actual design and implementation of the optimization based mechanisms for operative decision making became a real-life fact. It appeared that the introduction of such mechanisms created a need to develop and to master new specialized computing techniques as well as software tools enabling both verification and implementation of complex, possibly hierarchical, control systems. In particular, crucial role of computer-based experiment became apparent (Kleiber, 1999). While theoretical analysis can provide valuable assistance in understanding and evaluation of the systems concerned, it is only through computer simulation-based experiment that one can obtain both qualitative and quantitative overall evaluation of a complex control or management system prior to its actual implementation. Computer simulation experiment has, however, to be very carefully planned and executed if one wishes to obtain meaningful and reliable outcomes.

It has to be well structured, based upon good models and relevant data used to evaluate possible results of the proposed decisions. It also should be convenient to perform; there is a need for advanced software tools supporting development and execution of computer simulations.

2. Repetitive optimization of operative decisions

Consider a control or management system depicted in Fig. 1. An object of our interest – process P – evolves under the influence of decisions m formed by a decision maker or an automated procedure, and the free, uncontrolled inputs z. Goals, control objectives, are related to outcomes w. These outcomes may be expressed through constraints put on specified process variables or through various performance indices which should be, if possible, minimized or maximized. Objectives to be met by the decision maker, or the controller, have to be set when taking into account the uncertainty about future and – often – about present values of the free inputs. When building a decision mechanism one should consider on-line usage of available information concerning process behavior – measurements y_p – and information related to the environment behavior, observed through y_e. These external measurements, being relevant to free inputs generation, allow to formulate operative models of those inputs – the forecasting models. Observations of y_p and y_e provide for information I ($I = I_k$ at time k, i.e. at the beginning of the k-th stage of the process operation), which can be used for the operative decision making.

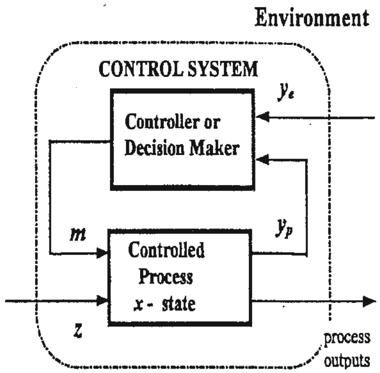

Figure 1. Control system; m – operative decisions, z – free (uncontrolled) inputs, w – effects of process operation

2.1. The repetitive optimization problem

As it was explained in the introduction, in a growing number of diverse applications there is a need to introduce on-line optimization of control or management decisions. One has to work out these decisions in the presence of meaningful, often very considerable, uncertainty concerning future free inputs and process parameters. It should be observed that, for the purpose of compact notation, the free inputs may represent, and will represent within this paper, both external uncertainty – uncontrolled inputs from the process environment – and internal uncertainty related to process behavior.

Repetitive decision making, based upon the use of process model and free input forecasting for decision optimization, can be in a discrete-time case – obtained by dividing the operation interval into stages between specified time instants – formulated as follows:

- at a given time k a forecast of future free inputs is provided in the form of a single or a multiple scenario, or in the form of a probability distribution, concerning possible values of these inputs during stages $j = k, k+1, \ldots, k+L_k-1$, where L_k denotes the prediction horizon at time instant k,
- after the above described forecast is made available, an optimization problem concerning operation stages $i = k, k+1, \ldots, k+K_k-1$ is formulated and solved; K_k denotes the optimization horizon at time k (obviously, there must be $L_k \geq K_k$),
- the computed decision vector m_k, for stage k, from time instant k to time $k+1$, is then implemented, and so on.

The basic decision mechanism belonging to the above generic type consists of an open-loop optimization of a sequence of decisions, with one single forecasted scenario of future free inputs. Decision algorithm – Basic Predictive Controller – is defined, for each consecutive time instant k ($k = 0, \ldots, K-1$), as follows:

1. Based upon the currently available information I_k about process and its environment's behavior till time k compute the estimated value $\bar{x}_k$ of the process state x_k at this time and a *a single scenario forecast* of future free inputs: $\bar{z}_k^{k,k+L_k-1} = \{\bar{z}_{k,k}, \bar{z}_{k,k+1}, \ldots, \bar{z}_{k,k+L_k-1}\}$, where $\bar{z}_{k,j}$ denotes the forecasted value of z_j, at time k, $k < j$.
2. Solve the Deterministic Decision Optimization Problem:
   DDOP$_k$:
   find a sequence of decisions (controls) $\bar{m}_{k,k}, \bar{m}_{k,k+1}, \ldots, \bar{m}_{k,k+K_k-1}$ such that
   $$\bar{m}_{k,k}, \bar{m}_{k,k+1}, \ldots, \bar{m}_{k,k+K_k-1} = \\ = \arg \min_{m_k, m_{k+1}, \ldots, m_{k+K_k-1}} \sum_{j=k}^{k+K_k-1} W_{j+1}\left(x_{j+1}, m_j, \bar{z}_{k,j}\right)$$
   where $x_{j+1} = f_j\left(x_j, m_j, \bar{z}_{k,j}\right), x_k = \bar{x}_k$ and $m_j \in M_j, j = k, \ldots, k+K_k-1$
3. Apply the decision for the k-th stage $m_k = \bar{m}_{k,k}$; during this stage, then, at time $k+1$, repeat steps 1, 2, 3, and so on.

It can be immediately observed that in the above mechanism the optimization performed in Step 2 makes use only of the process model given in the form of the state transition function $f_i(.)$, while forecasting, at Step 1, of the future free inputs may be based upon any available model of those inputs; it may in particular be based directly upon their past values. To compute estimated state of the process one should either use jointly the process and free inputs model or – when, for instance, good measurements of z_i, for $i < k$, are available – use only the process model on its own. In the above formulation of DDOP$_k$ an additive performance function, to be minimized, is assumed. There are different possible forms of such function – also in the form of vector performance criteria – as well as different constraints imposed on decision and state values.

Other possible formulations of multi-stage optimization of operative decisions at time k, or of decision policy considered at this time, could be:

a) Stochastic Closed Loop Optimization Problem – it is in fact a problem of optimal decision rule design under probabilistic, forecasted at time k, representation of future uncertainty; this rule is then implemented until time $k+1$,

b) Stochastic Open Loop Optimization Problem – in which expected value of a given performance function is minimized with respect to a sequence of decision values $\bar{m}_{k,k}, \bar{m}_{k,k+1}, \ldots, \bar{m}_{k,k+K_k-1}$ – as in DDOP$_k$; probabilistic distribution of future free inputs, forecasted at time k is required as in case of approach a) above,

c) Optimization problem based on the use of forecasted probabilistic distribution, or on multiple-scenario prediction of future free inputs, and assuming one (or more) future interventions of the decision making mechanism (limited look-ahead policy, Bertsekas, 1987, Niewiadomska et al., 1996),

d) Decomposed optimization problem – represented as a collection (to be coordinated if so required) of smaller optimization problems.

With respect to the decision mechanisms mentioned in point c) above, it is worth noting that the main disadvantage of the open-loop feedback policies is that, while computing the future control inputs, e.g. by solving DDOP$_k$, one does not take into account *any* future measurements nor interventions; this is in contrast with the optimal control design by stochastic dynamic programming, where *all* future interventions are accounted for. The intermediate, reasonable, solution is offered by the limited look-ahead schemes. To explain the basic idea behind these schemes consider, at time k, the following general look-ahead decision procedure:

Solve the Stochastic Look-Ahead Decision Optimization Problem:

SLADOP$_k$: find control input $\bar{m}_{k,k}$ such that

$$\bar{m}_{k,k} = \arg \min_{m_k \in M_k} \mathop{E}_{z_k, y_{k+1}, y_{e,k+1}} \left\{ W_k\left(x_{k+1}, m_k, z_k\right) + \tilde{J}_{k,k+1}\left(I_{k+1}\right) \right\}$$

where $x_{k+1} = f_k(\bar{x}_k, m_k, z_k), I_k = (I_k, y_{k+1}, y_{e,k+1}, m_k)$, and $\tilde{J}_{k,k+1}(I_{k+1})$ is – at time k – the approximate value of – cost-to-go from time $k+1$ till the end of the assumed – at time k – optimization horizon.

Instead of using the estimated value $\bar{x}_k$ of the state x_k at time k in the above problem formulation, it is possible to compute the expectation with respect to

z_k, $y_{k+1}, y_{e,k+1}$ and x_k to define the performance function value in SLADOP$_k$. Yet, to simplify the control scheme it is more convenient to separate the state estimation from the control computation; such enforced 'separation property' is common in practical control algorithms.

The crucial question is then how to define and, then, to compute at time k the approximate cost-to-go $\tilde{J}_{k,k+1}(I_{k+1})$. There are many possible ways to define $\tilde{J}_{k,k+1}(I_{k+1})$, some of them are discussed in Bertsekas (1987). Let us consider an important case, in which $y_{e,k+1} = z_k$. In such a situation this measurement is the only relevant new information at time $k+1$ – as perceived at time k – and so $\tilde{J}_{k,k+1}(I_{k+1})= \tilde{J}_{k,k+1}(x_{k+1})$, since – once we know $\bar{x}_k$ – the knowledge of z_k and the value of m_k is sufficient to compute x_{k+1}. $\tilde{J}_{k,k+1}(x_{k+1})$ can now be defined, for example, as a solution of an open-loop stochastic optimization problem, with x_{k+1} given and future free inputs – starting from time $k+1$ – considered as the sequence of random variables. If such approach is adopted to define $\tilde{J}_{k,k+1}(I_{k+1}) = \tilde{J}_{k,k+1}(x_{k+1})$, then, to be able to solve the SLADOP$_k$, it may be necessary to pre-compute the function $\tilde{J}_{k,k+1}(x_{k+1})$, prior to implementation of the considered scheme, as the on-line computation of this function values may be very time consuming. To avoid massive computation required for this purpose it might be useful to define $\tilde{J}_{k,k+1}(x_{k+1})$ in a different way.

Consider the case in which, at time k, one takes into account the set of forecasts of future free input values:

$$\left\{{}^s\bar{z}_k^{k,k+L_k-1} = \{{}^s\bar{z}_{k,k}, {}^s\bar{z}_{k,k+1}, \dots {}^s\bar{z}_{k,k+L_k-1}\}, \quad s = 1, \dots, S\right\},$$

with associated weights ${}^s w_k, s = 1, \dots, S$, and then *makes the assumption* that one of these forecasted trajectories will actually occur, and that at time $k+1$ it will be known which of them have had occurred. Then, $\tilde{J}_{k,k+1}(x_{k+1})$ can be defined, for $x_{k+1} = {}^s x_{k+1}$, $s = 1, \dots, S$, and for given $m_k \in M_k$, as the solution of the following optimization problem ^{s}DOP$_{k+1}$:
^{s}DOP$_{k+1}$: find $\tilde{J}_{k,k+1}$, such that

$$\tilde{J}_{k,k+1}\left({}^s x_{k+1}\right) = \min_{m_{k+1},\dots,m_{k+K_k-1}} \sum_{j=k+1}^{k+K_{k-1}} W_{j+1}\left({}^s x_{j+1}, m_j, {}^s\bar{z}_{k,j}\right)$$

where

$${}^s x_{j+1} = f_j\left({}^s x_j, m_j, {}^s\bar{z}_{k,j}\right), {}^s x_k = \bar{x}_k \quad \text{and} \quad m_j \in M_j, j = k, \dots, k+K_k+1.$$

Then, SLADOP$_k$ is transformed into the following Multi-Forecast-One-Step-Look-Ahead Controller (MFOSLAC$_k$) decision problem:
MFOSLAC$_k$: find control input $\bar{m}_{k,k}$ such that

$$\bar{m}_{k,k} = \arg\min_{m_k \in M_k} \sum_{s=1}^{S} {}^s w_k \left[W_k\left({}^s x_{k+1}, m_k, {}^s\bar{z}_{k,k}\right) + \tilde{J}_{k,k+1}\left({}^s x_{k+1}\right)\right]$$

where ${}^{s}x_{k+1} = f_k\left(\bar{x}_k, m_k, {}^{s}\bar{z}_{k,k}\right), s = 1, \ldots, S$ and where $\sum_{s=0}^{S} {}^{s}w_k = 1$.

Again, the practical possibility of using this controller depends on how efficiently one is able to solve – for each considered value of m_k – the collection of S open-loop optimization problems: ${}^{s}\mathrm{DOP}_{k+1}$, $s = 1, \ldots, S$. It is useful to observe that if a parallel computer is available then those optimization problems may and can be solved simultaneously – in parallel.

An answer to the question whether the use of the above MFOSLAC scheme is worth the considerable effort required is definitely application–dependent; it may be hoped that in most cases of practical interest it will be sufficient to apply simpler predictive controllers of the BPC type. In Niewiadomska et al. (1996) the example is provided, in which MFOSLAC is used to determine the outflows from the flood protection reservoirs during flood; it appears that in that case MFOSLAC allows for slightly better results than those obtained by using open-loop-feedback controllers.

In case when a composite controlled process is partitioned into N subprocesses it is natural to introduce – for purpose of optimization of operative decisions – a hierarchical structure consisting of local units and of a coordinating unit. Such a structure is depicted in Fig. 2. One can propose numerous decision strategies, which could be implemented within this structure.

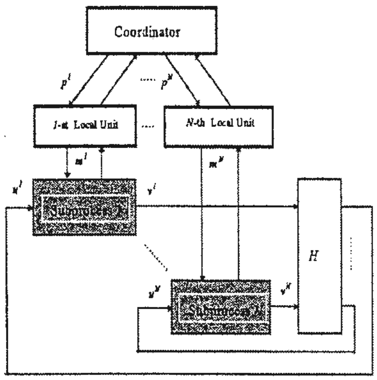

Figure 2. The hierarchical two-level structure

Most important, from the practical point of view, are hierarchical structures with periodic coordination. In brief, the operation of such a structure, with a collection of decision mechanisms at local and coordinator level, is the following (Malinowski, 1992):

- Lower level: local decision units compute their decisions m^i, concerning sub-processes, independently; these decisions are recomputed frequently at times $k, k+1$ etc., and one can possibly try to avoid using complicated decision procedures; simple parametric decision rules may be introduced, and re-adjusted by the coordinator at less frequent instants k_l, k_{l+1}, with the use of the coordinating instruments $p^i, i = 1, \ldots, N$,

- Upper level: the coordinator modifies at times k_l, k_{l+1}, etc., the local decision mechanisms so as to achieve satisfactory (or even best possible) overall process behavior – *periodic coordination* is realized through repetitive optimization of coordinating instruments p^i, set at time k_l until next intervention of the coordinator at time k_{l+1} etc.

On the other hand, two-level (or multi-level) hierarchical methods with decomposition and iterative coordination, like Direct Method or Price Method (Findeisen et al., 1980), can be used to solve DDOP_k, or similar optimization problems, in a complex process case. Successful applications of those techniques depend on various factors. Yet, as far as the operation of the controlled process is concerned, it is not so much important which technique is applied to solve optimization problems posed at subsequent time instants – provided that one can solve this problem sufficiently fast and accurately.

2.2. Examples of applications of control or management structures with optimization of operative decisions

It is useful now to describe a few examples of processes, which, together with accompanying operation goals, may require introduction of decision mechanisms with repeated optimization of control or other decisions.

Regulatory control system. Consider a typical control design problem, where the task is to find a controller which will drive the outputs of a given process to desired set-points or will make them follow a desired trajectory. Such design problem may arise in a straightforward way from direct design specifications related to a particular process – or it may result as *a direct control layer problem* from vertical decomposition of a complex control problem (Findeisen, 1974, 1997). In case of industrial process control the direct regulatory functions are often performed by traditional controllers like PI or PID. Such controllers do not involve on-line usage of process models and complicated numerical algorithms and can even be tuned using simple experiment based rules, like, for instance, Ziegler-Nichols rules. Yet, to improve performance of regulatory control one may be willing now to build a predictive controller (Camacho and Bordons, 1995), which operates according to the above presented Basic Predictive Controller scheme; linear model is used together with the – usually very simple – deterministic forecast of future free inputs, representing in this case the process and measurement disturbances.

Set-point control of industrial plant. Repeated optimization of a vector of set-point values for plant operating at steady-state or in periodic regime, is now used in a large variety of applications within the Manufacturing Execution Systems (MES). This long advocated approach (Findeisen, 1974, 1997) became quite common in chemical process industry and is introduced in environmental protection and other industries.

It is worth to observe that the current computer technology and available computing tools may make it useful, in a number of cases, to replace the classical two-layer controller – involving regulatory control at the direct, lower, layer, and set-point specification at the second, upper, layer – with a single layer predictive controller with repeated model and forecast-based optimization of control decisions. Such controller may allow for better dynamic performance of the controlled process. It is interesting also that this approach would be compatible with, now often observed, tendency to reduce the number of decision levels in management systems.

Management of a system of reservoirs during flood. An efficient usage of reservoir capacity during flood, especially during flood due to heavy rainfall, is one of the most important measures which can be considered for protection of people and property against floods (Malinowski, Żelaziński, 1990). Decision mechanisms, needed to determine the releases from the reservoirs, should make proper usage of available information, in particular – of weather and inflow forecasts. Many years of research into that matter and numerous computer experiments concerning introduction of centralized, decentralized and hierarchical flood management structures and algorithms for reservoirs of the Upper Vistula River System (Malinowski, 1984; Niewiadomska et al., 1996), allowed to demonstrate that the use of optimization based decision rules should provide for significantly better results than the use of the traditional fixed reservoir operation rules. In case of a single reservoir the mechanism with optimization of decisions consists of repetitive planning of future water releases – using single or multiple inflow scenario forecasting – so as to minimize the peak release from reservoir during the entire flood period. In case of a system of reservoirs and river reaches the objective is to minimize the damages associated with peak water levels at important points – damage centers. In this case it was demonstrated that the best possible results could be obtained when introducing a hierarchical control structure with periodic coordination. Each local decision mechanism of this structure is concerned with a particular reservoir and computes water releases from this reservoir by minimizing performance function which is periodically re-parameterized by the coordinator. The coordinator adjusts, over longer time intervals, its decisions by performing optimization with the use of simulation of the lower level operation.

Revenue management. One of the quickly developing areas of applications of modern decision mechanisms is 'control' of behavior of different segments of a market. Quantitative marketing can be achieved through operative decisions concerning various market instruments, in particular prices and availability characteristics of the offered products. The decisions are taken repetitively in view of local conditions. In this case the controlled process is represented by the considered market segment and the operative decisions are concerned with choosing such values of marketing instruments which would – in the best possible measure – lead to a bigger market share and increased profit (Singh and Bennavail, 1993; Cross, 1997). As an example one can mention the process of setting ticket prices for different flights by an airline, together with dynamic adjustment of numbers of seats offered in various categories (business, economy). Another example concerns petrol retail, where – in view of increasing competition – a network of sites, which wants both to retain (or to increase) its market share and to maximize profits, has to use sophisticated market analysis and decision support techniques to set both pump prices of petrol grades and prices of products offered at the site shops. On-line decision support requires identification of a behavioral model of the considered market segment, describing expected reaction of this segment to both ours and our competitors' actions, and repeated optimization of our prices. When a group of petrol stations is considered a hierarchical decision mechanism can be used. Within last years these ideas became exploited and implemented for actual decision support tools used by several major companies involved in petrol production and retail. The results obtained so far are very encouraging. Similarly, quantitative, forecast and optimization based, decision support techniques provided for surprisingly large profit and market share enhancements of airline carriers and car renting companies. Decision mechanisms for operative pricing and shelf space allocation are now being tested by supermarket chains; significant profits are expected.

3. Key issues

Introduction of a control or management system with centralized or hierarchical mechanism consisting of repetitive optimization of operative decisions can be successful, provided careful preparation of such mechanism and its verification and tuning is performed prior to actual real-life implementation. The key issues are:

- Formulation of a decision optimization problem,
- Choice (preparation) and testing of computational methods used to identify process model parameters and process state, to compute forecasts of free inputs and, finally, to perform optimization of decisions; all this must be done in real time,
- Theoretical analysis of the proposed control structure, its elements and important aspects of operation,
- Simulation experiment: setting up this experiment, verification of all components of the control system as well as of the entire system; tuning of forecasting and decision mechanisms.

3.1. Formulation of a decision optimization problem.

Correct formulation of a decision optimization problem requires the following, properly defined and mutually compatible, elements:

- the operation (control) objectives, in the form of performance criteria and constraints on relevant quantities,

- the model of controlled process and, if needed, due to formulation of the optimization problem, a model of the measurement system,
- the model of free inputs and a representation of free input forecasts used for repetitive optimization of decisions.

It must be stressed that the design stage at which formulation of the decision optimization problem is done plays the crucial role. There are, usually, many degrees of freedom. In particular, existing data allow to introduce various forms of description (models) of uncertainty, i.e. of the free inputs. One can use probabilistic models in the form of Markov processes or in the form of dynamic generators driven by white noise input (e.g. ARMA model). It is also possible to characterize uncertain quantities by set membership models or by fuzzy sets. Such models have to be compatible with the objectives to be assessed and optimized. For example, if a priori evaluation of a given performance index is defined as an expected value of this performance, then the probabilistic models – allowing for computation of this expectation – must be used (Bertsekas, 1987). If a priori performance evaluation is in form of the worst case value, then set membership models of uncertain quantities are needed. When a priori performance is defined through a fuzzy constraint (Zimmerman, 1996), then compatible, fuzzy, description of free inputs is required.

A natural, in majority of applications, a priori performance evaluation in the form of expected value is, therefore, often hampered by the accompanying necessity of introducing good probabilistic model of the process operation and free input behavior. One can in fact find in many texts the argumentation supporting the view that such probabilistic models are difficult to obtain and verify when based on insufficient data, and that it is easier and, perhaps, more natural to introduce and use other models, for example based on possibility concepts or fuzzy sets. In author's opinion such reasoning is not fully substantiated. Similarly as a fuzzy model, a probabilistic model can be introduced with the use of the available, even quite poor information. To put trust in this model one should only avoid embedding in this model any information which, in reality, is not available; in other words one should observe the maximum uncertainty principle. This can be achieved by using the entropy optimization principles as explained in Kapur and Kesavan (1992).

Building models of uncertainty in the form of fuzzy sets, with the use of fuzzy logic, is reasonable in situations in which one is in possession of a knowledge base – provided by experts – consisting of a number of linguistic rules. Then, fuzzy sets and fuzzy logic offer tools to construct a decision mechanism; this mechanism requires then parametric tuning – which, in turn, implies a need to have either a probability model of uncertain factors or a sufficient amount of data.

For practical purpose it is often sufficient to model uncertain inputs, at a given time, in the form of forecasted – and, if possible, generated by simple algorithm – single scenario of these inputs. Such approach to deterministic modeling of uncertainty is used within the Basic Predictive Controller scheme presented in the previous section; it is commonly adopted in predictive regulatory control algorithms (Camacho, Bordons, 1995) and in supervisory control or management mechanisms (Findeisen et al., 1980). It is always necessary, however, to perform a number of simulation experiments to be able to find whether the BPC scheme is sufficiently good to be accepted.

3.2. Computational techniques

Practical realization of a mechanism with repetitive optimization of operative decisions is possible only when sufficiently fast and reliable computational techniques, in particular optimization methods needed to solve problems like DDOP_k or more complicated problems, are available. To solve large optimization problems, which allow for problem partitioning, one can use hierarchical methods with iterative coordination (Findeisen et al., 1980; Malinowski, 1992).

It could appear that the existing range of optimization methods, constantly improved and developed, given in the form of computer procedures, allows in almost any practical case to choose and implement, without too much hassle, a method satisfying the specified requirements. Unfortunately, this is not the case. Apart from the algorithms used to solve linear programming problems (the Simplex Method and its variants) there do not exist sufficiently fast and reliable *universal* methods for nonlinear optimization problems, especially such problems in which many constraints are imposed on the decision and other variables. Almost always, when having to cope with a real-life practical problem, one must provide a specialized computational method – an algorithm taking into account characteristic features of a particular decision optimization task. Obviously, such specialized algorithm may be created based upon some existing general purpose optimization method, after this method has been adapted for particular application, and proved, by tests, to be sufficiently reliable. Yet, quite often it is necessary to build a new specialized algorithm. Such was the case when developing decision mechanisms for flood control in a multiple reservoir system (Niewiadomska et al., 1996), and when developing algorithms for operative price optimization in marketing applications. Simultaneously, it must be stressed that knowledge concerning theory of optimization and existing optimization techniques is always very useful, often necessary, to build specialized, effective, dedicated methods.

It is interesting to observe that rapid development of computer technology allows to use computational techniques for on-line applications, which need a lot of computer power. The emphasis then, when developing such techniques for a given application, is upon the reliability of the method and its ability to handle all relevant constraints, instead of trying to decrease the number of mathematical operations or to reduce memory requirements. In particular, for process state and model parameter estimation one might be willing now to use optimization of an error function in the presence of known constraints on these state or parameter values, instead of applying the Kalman Filter (or Extended

Kalman Filter) which does not allow to observe such constraints.

3.3. Theoretical analysis of overall control (management) system

In view of considerable complexity of a control or management system with repetitive optimization of operative decisions, theoretical analysis of this system is possible only with respect to some aspects of its operation, such as: convergence properties of the computational methods, dynamic stability – if stability is an issue – and bounds on possible performance values. Most interesting, from the practical point of view, quantitative aspects of system operation are usually extremely hard to examine on an analytical level.

In general, theoretical results are possible to be achieved only under simplifying, often far reaching assumptions (Malinowski, 1992). Nevertheless, such analytical results can be very important for a better understanding of system operation and properties of proposed decision mechanisms. An obvious advantage of analytical results is their generality; these results are not concerned with particular values of decision variables, free inputs, etc.; no experiment, real-life or simulation based, has this property. Yet, simulation experiments may be performed when theoretical analysis is impossible or gives only partial answers to the questions posed. In most cases of interest such experiments are fundamental for evaluation of the considered control mechanism, especially for evaluation of quantitative effects of repetitive optimization of decisions, and are followed then by the real-life experiments (pilot implementations).

3.4. Simulation experiment

The objective of simulation experiments is to tune forecasting and decision mechanisms, to verify operation of all components of a control system and, finally and most important, to examine the operation of this system as a whole. In view of the important role of simulation experiments in design and analysis of control (management) systems with repetitive optimization of operative decisions the following section is devoted to discussion of objectives, components and other relevant aspects of Computer Analysis of Control (CAC).

4. Computer analysis of control

4.1. CAC; objectives

Based upon computer simulation, i.e. on a computer experiment (Cellier, 1991), computer analysis of control (CAC) allows, at the pre-implementation stage, to examine the properties of the decision mechanisms and the possible effects which may accompany real-life implementation of these mechanisms. CAC involves numerical simulation experiments needed to evaluate quality of control algorithms in view of their effectiveness, robustness, unitary and averaged quantitative measures of performance.

Computer supported design is still relatively young; its development has been stimulated by appearance and proliferation of powerful personal workstations and PC's, possessing large computational and graphical capabilities. These computers are usually networked, which allows for data exchange and distributing processing; in particular – for parallel computing. Computer supported design and analysis of control or of management systems makes use of achievements of several technical disciplines, in particular control engineering, management science, computer science and information engineering. CAC allows for taking into account all phenomena relevant to real system operation, including mutual interaction between the controlled process and its environment. CAC plays the most important part in cases when one has to investigate control or management structures and decision mechanisms for complex processes, when it is not possible to obtain quantitative results from theoretical analysis. Computer experiment requires a lot of commitment on the part of a designer. To make his or her efforts fruitful and to decrease the amount of time spent on setting and performing the experiment it is necessary to provide both good methodology and software tools.

The objectives of CAC are:

- evaluation and comparison of various possible control (decision) strategies with respect to their effectiveness, robustness and other factors like speed of computing,
- examination of the influence of forecasting quality on behavior of decision strategy and the overall control system,
- evaluation of the influence of various external and internal factors on system operation (e.g. effects of delays in computing decisions and transmitting data),
- proposals concerning modifications of estimation, forecasting and decision algorithms.

It is very important to provide the decision maker with a convenient interface to CAC system, to automatize as many steps leading to performing a computer experiment as possible, and to ensure that each essential step of the experiment is well documented.

CAC requires, in most cases, a large number of simulation experiments to get meaningful results. These simulations should be based on reliable data, for example on realistic scenarios of free inputs and on the identified process model. As far as the input scenarios are concerned, historical real data are preferred to artificially generated data. If it becomes necessary to generate new data for simulation a lot of care must be taken to make sure that these data conform to investigated operational conditions. When control system with repetitive optimization of operative decisions is examined, one needs also to simulate forecasting mechanisms to solve, within simulation, decision problems like DDOP_k or $\mathrm{MFOSLAC}_k$ (in the latter case multiple forecasted scenarios are required). Also, other factors which may affect the real-time operation, like the already mentioned delays in computing and transmitting decisions (Malinowski, 1992), have to be taken into account.

4.2. Elements of CAC

In view of the above presented role and objectives of CAC, it is useful to distinguish several elements needed to set and then to perform a simulation experiment. The most important are:

- Simulators of processes and phenomena which are considered as external with respect to the decision maker (control agent), in particular: simulator of the controlled process, simulator (generator) of the values (trajectories) of free inputs, simulator (forecast dummy) of any forecasts used by the decision mechanism if such forecasts are prepared by units being external with respect to the decision maker,
- Simulator of behavior of the operator (operators) if there is a need to simulate activities and decisions of the human being involved in process operation, in particular – in decision making,
- Identification and estimation algorithms used for on-line adaptation of all models and for state estimation,
- Decision mechanisms; either explicit control rules or optimization based decision procedures,
- Time structure: time scales specifying when and what information is obtained, when and what decisions are made and when the simulation is advanced, what are decision and transmission delays, etc.,
- Software environment under which the computer experiment is organized, including tools to create user defined modules, graphical interfaces to monitor experiment, facilities to store experimentation data, and providing convenient means for user-computer interaction.

To illustrate the above points consider hierarchical two-level structure, as depicted in Fig. 2, in case when this structure represents two-level flood management described briefly in Section 2.2, presented in detail in Niewiadomska et al. (1996). Local decision mechanisms are concerned with decisions made by reservoir operators and are represented by simple decision rules or optimization based procedures of low complexity using forecasts of future local inflows. These rules are from time to time modified by the control center (coordinator), through coordinating parameters p^i. Local reservoir problems are solved at time intervals ΔT_L. The control center solves a much more complicated problem, at longer intervals ΔT_C. As a result of solving this coordinator problem, new parameters for local mechanisms are established. Decision mechanisms used to solve the coordinator problem are based upon the minimization of a performance function, which is defined through simulations of the lower level operation. These simulations are performed for given scenarios of external inputs (inflows); such scenarios can be different from those that are used for the lower level decision making. Computer experiment involves all elements of this structure: models of the water system, including reservoirs and river reaches, forecasting mechanisms for the local level and for the control center, decision rules of reservoir operators, and the complete decision mechanism at the upper level. Both historical inflow scenarios during flood periods and other, feasible, scenarios of these inflows can be used.

It must be stressed that for the successful computer analysis it is absolutely necessary to make sure that within a computer simulation experiment one may only use data to simulate forecasting and decision making of such kind and quality as will be used in real-life implementation of these mechanisms, i.e. which will be available to the decision maker under real operating conditions. One should not use for this purpose other information, for example, without making it explicitly clear, or use more accurate – than actually available, simulated forecasts; e.g. to use overoptimistic forecast dummies.

4.3. Software environments for CAC

It has been already said that effective computer experiment can be set and performed in a reasonable time only if good software tools are available. These tools, in the form of a software environment for CAC, have to support the decision maker – alleviate tedious tasks and enable him or her to concentrate on important issues.

One may distinguish the following types of software environments for CAS:

1. Specialized systems: designed and developed for particular application. As an example let us mention: the FC-VS system (Flood Control Vistula System) to be used for simulation and support the real-time operation at the center (coordinator) level of the hierarchical two-level management structure described above (Niewiadomska-Szynkiewicz et al., 1992),
2. General purpose (universal) systems: software systems with open architecture, like SIMULINK-MATLAB, which can be used as support tools for a wide variety of applications. Such systems are usually too general to make a convenient environment to investigate complex control (management) systems,

Intermediate solution could be to develop software environments with open architecture, yet oriented towards analysis of complex systems with many decision and process units and capable of distributing the computing tasks during simulation between several processors (workstations). System CSA&S (Complex Systems Analysis and Simulation) belongs to that category and was developed in several versions (Niewiadomska-Szynkiewicz et al., 1995; Niewiadomska-Szynkiewicz and Malinowski, 1999); this software tool can be also used to organize computer experiments involving complex optimization mechanisms and information exchange patterns. It has already been used to perform computer aided analysis of flood control systems, routing mechanisms in telecommunication networks, and water management in protected agricultural systems.

References

BERTSEKAS, D. P. (1987) *Dynamic Programming: Deterministic and Stochastic Models.* Prentice-Hall, Englewood Cliffs, N.J.

CAMACHO, E. F., C. BORDONS (1995) *Model Predictive Control in the Process Industry.* Springer, London.

CELLIER, F. E. (1991) *Continuous Process Modeling.* Springer, New York.

CROSS, R. (1997) *Revenue Management; Hard-Core Tactics for Profit-Making and Market Domination.* Orion Business Books, London.

FINDEISEN, W., F.N. BAILEY, M. BRDYŚ, K. MALINOWSKI, P. TATJEWSKI, A. WOŹNIAK (1980) *Control and Coordination in Hierarchical Systems.* J. Wiley, London.

FINDEISEN, W. (1974) *Wielopoziomowe układy sterowania.* PWN (Polish Scientific Publishers), Warszawa (in Polish).

FINDEISEN, W. (1997) *Struktury sterowania dla złożonych systemów*, Oficyna Wydawnicza Politechniki Warszawskiej (Warsaw Univ. of Technology Press), Warszawa (in Polish).

GUTENBAUM, J. (1988) Dynamic Multistage Processes. *Systems and Control Encyclopedia*, Pergamon Press, 1272–1276.

GUTENBAUM, J. (1999) Sterowanie optymalne procesami wieloetapowymi. In: K. Mańczak, ed. *Analiza Systemowa i Zarządzanie*, Instytut Badań Systemowych PAN, Warszawa, 97–106 (in Polish).

KAPUR, J. N., KESAVAN, H.K. (1992) *Entropy Optimization Principles with Applications.* Academic Press, Boston.

KLEIBER, M. (1999) Modelowanie i symulacja komputerowa – moda czy naturalny trend rozwojowy nauki. *Nauka Polska*, **VIII (XXXIII)**, Rocznik kasy im. Józefa Mianowskiego Fundacji Popierania Nauki, Warszawa, 41-52 (in Polish).

MALINOWSKI, K. (1984) Hierarchical structure for real-time control in a multireservoir system. *1st European Workshop on Real-Time Control of LLS*, Patras, Greece.

MALINOWSKI, K. (1992) Practical Issues of Coordination in Control of Large-Scale Stochastic Systems. In: S.G. Tzafestas and K. Watanabe, eds., *Stochastic Large-Scale Engineering Systems*, Marcel Dekker, 195–227.

MALINOWSKI, K., ŻELAZIŃSKI, J. (1990) Reservoir systems: Operational flood control. *Systems and Control Encyclopedia*, supplementary vol. 1, 495–503, Pergamon Press.

MALINOWSKI K., NIEWIADOMSKA-SZYNKIEWICZ, E. (1995) Computer-Based Analysis of Complex Systems; Basic Concepts and Software Environments. *7-th IFAC Symposium on Large Scale Systems*, London, July 1995.

NIEWIADOMSKA-SZYNKIEWICZ, E., KARBOWSKI, A., MALINOWSKI, K. (1992) FC-VS a Decision Support System for Flood Control in Multireservoir System. *Environmental Software*, 7, 217–228.

NIEWIADOMSKA-SZYNKIEWICZ, E., MALINOWSKI, K., KARBOWSKI, A. (1996)

Predictive methods for real-time control of flood operation of a multireservoir system: Methodology and comparative study. *Water Resources Research*, **32**, 9, 2885–2895.

NIEWIADOMSKA-SZYNKIEWICZ, E., MALINOWSKI, K. (1999) Computer-Based Analysis and Design of Complex Control Systems. Research Report, Institute of Control and Computation Engineering, Warsaw University of Technology. Submitted for publication.

PYTLAK, R. (1999) *Numerical Methods for Optimal Control Problems with State Constraints.* Springer Verlag, Berlin-Heidelberg.

SINGH, M. G., BENNAVAIL, J-C. (1993) PriceStrat: a knowledge support system for profitable decision making during price wars. *J. Information and Decision Technologies*, **19**, 227–246.

ZIMMERMANN, H. J. (1996) *Fuzzy Set Theory and Its Applications.* Kluwer Academic Publishers, Boston.