Theory and Methodology

Some problems with the design of self-learning Open-loop control systems

Ziny Flikop *

NYNEX Science and Technology, Inc., 500 Westchester Avenue, White Plains, NY 10604 USA

Received October 1992; revised Septemb er 1993

Abstract: In this paper some problems in the design of open-loop control systems for complex objects are discussed. Considering the absence of adequate models and the fact that human expertise in the control of non-stationary objects becomes obsolete quickly, the use of self-learning together with a two-step optimization for the development of on-line open-loop control rules is suggested. To prepare for the object analysis, a set of definitions has been proposed. Traditional and fuzzy sets (see Zadeh and Sugeno) approaches are used in the analysis.

Keywords: Control processes; Decision theory; Fuzzy sets; Optimization

1. Introduction

Automatic and semi-automatic control systems for the complex objects usually are based on sets of control rules. The development of such rules requires either comprehensive human expertise or an adequate object model or both. However, human expertise in the control of the complex non-stationary objects becomes obsolete with time. In addition, commonly used closed-loop control has a prolonged reaction time. These problems can be partially avoided if open-loop control together with a self-learning can be used. In this paper, which is but another drop in the sea of control literature, we too are proposing and studying some variant of such an approach.

The methodology for synthesizing control systems depends on the complexity of the controlled object. In this paper we discuss the control of complex objects with multiple inputs and outputs. As an example of a controlled object we use a telecommunication network proposed in [3] (Fig. 1) and its model developed in correspondence with [l].

The numbers displayed in Fig. 1 represent line capacities C^i in packets-per-second, i = 1,. . . ,16, is a line number. In this network, traffic transmitted between any source and destination nodes can be split among different paths.

In general, the .object transformation function (mapping) is defined by an object organization (structure) and by the values of the object element set-up parameters (control variables). In our example, network nodes, their connectivity and link capacities represent network structure. The traffic routing tables defined for each network node are control variables. Often object perfor mance is evaluated by multiple criteria (via

* Corresponding author.

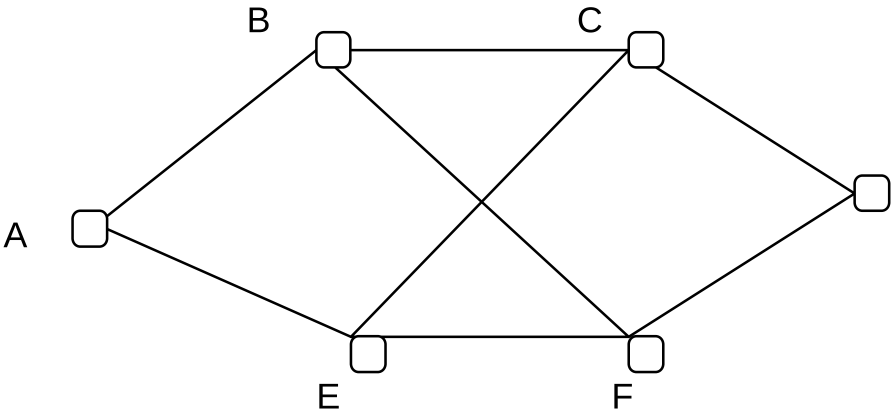

Fig 1

multiple controlled variables). We evaluate performance of our network via average delay that packets undergo during traversing a network. In other words, delay is our controlled parameter (controlled output). Control system performance can be evaluated by the control system's ability to maintain outputs at the predefined level (the simple control task) and by its ability to minimize the `cost' required for the control (the optimized control task). In our example, we evaluate performance of the network control system by its ability to maintain an average delay below predefined level (the simple control task) and execute this control upon minimal network cost (the optimized control task). Control can be executed by varying either control variables and/or object structure. In the development of control systems, one should consider that control system reaction time must be much shorter than input drift, and processes of environmental and structural changes. This requirement is particularly important for the telecommunications network control systems.

The ability of control systems to control and optimize objects depends on the efficiency of the algorithms used for these purposes. In turn, selection of these algorithms depends on continuity, separability, and monotonicity of controlled object mapping functions. In general, the behavior of mapping functions depends on the nature of controlled objects. If, for example, simple physical devices often have mapping functions that are continuous, separable, and monotonic, this is not always the case for more complex controlled objects, such as heterogeneous telecommunication networks, for example. When mapping functions are continuous, separable, and monotonic, relatively simple control and optimization algorithms can be applied. However, when mapping functions are not continuous, separable, and monotonic, then significantly more powerful algorithms are needed. Because the behavior of mapping functions imposes limitations on the selection of control and optimization algorithms, and since proposed self-learning methodology is based on such algorithms, we define what we mean by continuity, separability, and monotonicity. We also define several other terms used in this paper.

2. Definitions

From a control point-of-view, a general object can be defined via the following mapping functions:

$$F(V, H): X \rightarrow Y, \;\; F(X, H): V \rightarrow Y \text{ and } F(X, H): V \rightarrow Q, \qquad (1)$$

where:

$X = (x^1, x^2, ..., x^m)$ is an input vector, $m = |M|$, M is a set of input variables;

$Y = (y^1, y^2, ..., y^n)$ is an output vector, $n = |N|$, N is a set of output variables;

$V = (v^1, v^2, ..., v^p)$ is a vector of control variables , $p = |P|$, P is a set of control variables;

$Q = (q^1, q^2, ..., q^b)$ is a vector of controlled variables, $b= |B|$, B is a set of controlled variables;

$H = (h_l)$ is a controlled object structure, h_l is an object element (link), $l \in L$ In our example L is a set of network elements and links connecting such elements. We define some properties of these functions:

a. Let's consider a function $F(V, H) : (X + \mathrm{l}) \to (Y + \mathrm{m})$ with any fixed V and H as continuous into space C if when $\mathrm{l} \to 0$, then $\mathrm{m} \to 0$ for $\forall X \in C, (X + \mathrm{l}) \in C$. (2)

b. Let's consider a function $F(X, H) : (V + \mathrm{h}) \to (Y + \mathrm{m})$ with any fixed X and H as continuous into space A if when $\mathrm{h} \to 0$, then $\mathrm{m} \to 0$ for $\forall V \in A, (V + \mathrm{h}) \in A$. (3)

c. Let's consider a function $F(X, H) : (V + \mathrm{h}) \to (Q + \mathrm{r})$ with any fixed X and H as continuous into space D if when $\mathrm{h} \to 0$, then $\mathrm{r} \to 0$ for $\forall V \in L, (V + \mathrm{h}) \in L$. (4)

d. Let's consider a function $F(V, H) : X \to Y$ with any fixed V and H, and $X \in C$ as monotone if when X is changing in one direction along same monotone trajectory into C, then Y is also changing in one direction along a monotone trajectory into output space. (5)

e. Let's consider a function $F(X, H) : V \to Y$ with any fixed X and H and $V \in A$ as monotone if when V is changing in one direction along same monotone trajectory into A, then Y is also changing in one direction along a monotone trajectory into output space. (6)

f. Let's consider a function $F(X, H) : V \to Q$ with any fixed X and H and $V \in D$ as monotone if when V is changing in one direction along same monotone trajectory into L, then Q is also changing in one direction along monotone trajectory into controlled variable space. (7)

g. Let's consider a function $F(V, H) : X \to Y$ as separable if $F : (x^1, x^2, ..., x^i + \Delta^i, ..., x^m) \to (Y + k^i)$ and $F : (x^i, x^2, ..., x^j + \Delta^j, ..., x^m) \to (Y + k^j)$ then

$$F : (x^i, x^2, ..., x^{i} + \Delta^i, ..., x^j + \Delta^j, ..., x^m) \to (Y + k^i + k^j) \quad (8)$$

h. Let's consider a function $F(X, H) : V \to Y$ as separable if $F : (v^1, v^2, ..., v^i + \Delta^i, ..., v^p) \to (Y + k^i)$ and $F : (v^1, v^2, ..., v^j + \Delta^j, ..., v^p) \to (Y + k^j)$ then $F : (v^1, v^2, ..., v^i + \Delta^i, ..., v^j + \Delta^j, ..., v^p) \to (Y + k^i + k^j)$ (9)

i. Let's consider a function $F(X, H) : V \to Q$ as separable if $F : (v^i, v^2, ..., v^i + \Delta^i, ..., v^p) \to (Q + \mathrm{j}^i)$ and

$F : (v^i, v^2, ..., v^j + \Delta^j, ..., v^p) \to (Q + \mathrm{j}^j)$ then $F : (v^1, v^2, ..., v^i + \Delta^i, ..., v^j + \Delta^j) \to (Q + \mathrm{j}^i + \mathrm{j}^j)$ (10)

We can define the fluctuation range of $i,(i \in M)$ input variable by an ordered A^i set that consists of real numbers x^i representing possible measured values of this variable. The whole input space is:

$$A = A^1 \times A^2 \times ... A^M \ . \qquad (11)$$

We can define the reference (desired) value y_r^i for every output variable $i,(i \in N)$ and the reference output vector $Y(y_r^1, y_r^2,..., y_r^n)$ for the whole object. As a particular case, we also can define the permissible output space:

$$\Phi = [y_r^1 + \mathrm{a}^1, y_r^1 - \mathrm{a}^1] \times [y_r^2 + \mathrm{a}^2, y_r^2 - \mathrm{a}^2] \times ... \times [y_r^n + \mathrm{a}^n, y_r^n - \mathrm{a}^n], \qquad (12)$$

where a^i is an accuracy of tracking *i* variable; $[y_r^i + \mathrm{a}^i, y_r^i - \mathrm{a}^i]$ is a permissible interval of *i* output variable.

Each permissible interval also can be represented by a normalized fuzzy set U^i with a membership function $b_{v^i}(y), y \in \sup U^i$. In this set, y_r^i has a maximal possible grade $b_{v^r}(y_r^i) = 1$. In this case, the permissible output space q can be defined as:

$$\mathrm{q} = \text{supp } U^1 \times \text{supp } U^2 \times ... \times \text{supp } U^n \ . \qquad (13)$$

The actual output vector Y_t, at moment *t* usually differs from Y_r. This difference is the result of either x_t, drift or mapping function changes caused by *V* instability and environmental and *H* changes. Deviation of Y_t from Y_r, is a control error for which the control system must compensate. Compensation can be done either by varying only *V* or only *H* or by simultaneous changes *V* and *H*.

The quality of control is evaluated either by an output error vector that at moment *t* is:

$$\Xi_t = (\mathrm{e}_t^1, \mathrm{e}_t^2,..., \mathrm{e}_t^n), \text{ where } \mathrm{e}_t^i = y_t^i - y_t^r, (i \in N), \qquad (14)$$

or by $F = \sum_{i \in N} (1 - b_{v^i}(y_t^i))$ that can be used for estimating a degree that $Y_t \notin \mathrm{q}$. (15)

We will consider that the controlled object is working within a required accuracy if either of the following conditions are satisfied:

$$\forall i,(i \in N)(y_r^i + \mathrm{a}^i) \leq y_t^i \geq (y_r^i - \mathrm{a}^i) \text{ or } y_t^i \in \text{ supp } U^i \qquad (16)$$

$$\text{or } Y_t \in \Phi \text{ or } Y_t \in \mathrm{q} \ . \qquad (17)$$

All *X*, for which conditions (16, 17) are satisfied for some combination of *VH*, are permissible input vectors for this combination. We can propose a definition of permissible input subspace Ψ^z for *z*-th combination of *VH*: $\Psi^z = \{X \mid X \rightarrow Y \in \Phi$ or $Y \in \Theta$ for z-th combination of *VH*$\}$.

We can define for each output variable i at any time t a. distance of y_t^i from the border of a permissible output interval either via:

$\mathrm{d}_t^i = \min(| y_t^i + \mathrm{a}^i - y_r^i |, | y_r^i + \mathrm{a}^i - y_t^i |)$ or via membership grades as $1 - b_{v^i}(y_t^i)$. (18)

For the output vector we can use either:

$$D_t = \sqrt{\sum_{i\in N} (\mathrm{d}_t^i)^2}, \tag{19}$$

or (15). The efficiency with which the control system executes its controlling functions can be different for different combinations of V and H. We can introduce a, multivalued goal function G that can be used for evaluating the efficiency of the control system and optimizing the object:

$$G_t = \sum_{f\in B} \mathrm{g}^f q_t^f \tag{20}$$

where g^f is a weight coefficient of a controlled variable $f, f \in B$.

In our telecommunication network example control can be presented via:

$$F(V_{net}, H) : \Lambda_{imp} \to R, \tag{21}$$

where $V_{net} = (v^{AB_1}, ..., v^{DC_j}, ..., v^{DC_j}, ..., v^{DC_z}, ..., v^{FK_k})$ is the control vector that is implemented in the network via node routing tables and defines sets of paths for every source-destination pair of nodes. It also defines in what *proportion* traffic must be split between paths. For example, v^{DC_j} is the portion of the traffic transmitted from source node D to destination node C via path j. (Control vectors are presented in Table 1, columns "Curve A", "Curve B", and "Curve C".)

$H = \{H\}$ is a set of network nodes and links.

$\Lambda_{imp} = (\Lambda^{AB}, ..., \Lambda^{DC}, ..., \Lambda^{EF})$ is an input vector, where Λ^{DC} for example, is traffic that is entering the network via node D and destined for node C. When traff;c between nodes D and C is split, then

$\Lambda^{DC} = \sum_{i\in j^{DC}} \mathrm{l}^j; J^{DC}$ is a set of paths between which traffic from D to C is split; $\mathrm{l}^j = \Lambda^{DC} v^{DC_j}$ is a traffic in the path j.

One can see examples of splitting in Table 1, `Path ' column. $\Lambda_{out} = (\tilde{\Lambda}^{AB}, \tilde{\Lambda}^{AC}, ..., \tilde{\Lambda}^{BC}, ..., \tilde{\Lambda}^{EA})$ is a vector representing network output traffic (it is a network performance constrain, not a controlled output from our network performance control point of view). $\tilde{\Lambda}^{AB}$ is, for example, traffic that is entering the network via node A, is transmitted by the network to node B, and is leaving successfully (without packet loss) the network via node B.

Table 1
Network traffic routing

Source	Destination	Traffic (pkts/sec)	Path	Curve A Traffic (%)	Curve B Traffic (%)	Curve C Traffic (%)
A	B	9.0	AB	100.0	100.0	100.0
A	C	4.0	ABC AEC	40.0 60.0	47.5 52.5	16.25 83.75
A	D	1.0	ABCD AECD ABFD	90.0 10.0	100.0	10 20 70
A	E	7.0	AE	100.0	100.0	100.0
A	F	4.0	AEF ABF	100.0	100.0	65.0 35.0
B	A	9.0	BA	100.0	100.0	100.0
B	C	8.0	BC	100.0	100.0	100.0
B	D	3.0	BCD BFD	50.3 49.7	100.0	89.7 10.3
B	E	2.0	BFE	100.0	100.0	100.0
B	F	4.0 and vary	BF	100.0	100.0	100.0
C	A	4.0	CBA CEA	51.25 48..75	32.5 67.5	71.25 28.75
C	B	8.0	CB	100.0	100.0	100.0
C	D	3.0	CD	100.0	100.0	100.0
C	E	3.0 and vary	CE	100.0	100.0	100.0
C	F	2.0	CEF	100.0	100.0	100.0
D	A	1.0	DCBA DCEA DFBA DFEA	30.0 60.0 10.0	60.0 5.0 35.0	5.0 60.0 35.0
D	B	3.0	DCB DFB	56.7 43.3	46.7 53.3	90.0 10.0
D	C	3.0	DC	100.0	100.0	100.0
D	E	3.0	DFE DCE	60.0 40.0	40.0 60.0	93.3 6.7
D	F	4.0	DF	100.0	100.0	100.0
E	A	7.0	EA	100.0	100.0	100.0
E	B	2.0	EFB	100.0	100.0	100.0
E	D	3.0	EFD ECD	57.7 42.3	72.3 27.7	75.0 25.0
E	C	3.0	EC	100.0	100.0	100.0
E	F	5.0	EF	100.0	100.0	100.0
F	A	4.0	FEA FBA	100.0	90.0 10.0	100.0
F	B	4.0	FB	100.0	100.0	100.0
F	C	2.0	FEC	100.0	100.0	100.0
F	D	4.0	FD	100.0	100.0	100.0
F	E	5.0	FE	100.0	100.0	100.0

(When traffic approaches link capacity, then the network can drop traffic to avoid congestion. In this case $\Lambda_{out} \leq \Lambda_{inp}$

. $R = \sum_{i=1}^{16} \mathrm{l}^i / (C^i - \mathrm{l}^i)$ is a network performance; (22)

$\mathrm{l}^i / (C^i - \mathrm{l}^i)$ is an average traffic delay in the link i, C^i is a value of traffic in link i ,

C^i is a capacity of link *i.* (23)

Since task of the network control consists in maintaining value of *R* below some predefined level and without packet loss (that is a constrain), *R* is a controlled output of our network performance control system. In this case we will use $F(H, \Lambda_{inp}) : V_{net} \rightarrow R$ mapping for the control system development. For simplicity, let's evaluate an efficiency *G* of the network according to the following:

$$G = \sum_{i}^{16} C^i \qquad (24)$$

3. Control

Control systems designed in correspondence with a proposed terminology should be able to work in two interrelated modes:

1. *Simple control mode.*

This is either a process of minimizing an output error vector:

$$\underset{V,HV,H}{\min} \quad \Xi_t = (\mathrm{e}_t^1, \mathrm{e}_t^2, ..., \mathrm{e}_t^n) \quad \text{or} \quad \underset{V,HV,H}{\min} \quad \sum_{i \in N} (1 - b_{v^i}(y_t^i)) \qquad (25)$$

or a process of confining Y_t , to permissible output space (12, 13), which is executed by varying either only *V* or *V* together with *H*. For successful control, conditions (2) and (3) must be satisfied. Controlling algorithms can be relatively simple if conditions (5, 6, 8, 9) are also satisfied.

2. *Optimized control mode*.

This is also a process of the object control that corresponds (25). However, here the object performance is *optimized* by varying either only *V*, or *V* and *H*:

$$\underset{V,H}{\max} G = \sum_{f \in B} \mathrm{g}^f q^f \qquad (26)$$

Relatively simple algorithms can be used for this optimization if conditions (4, 7, and 10) are satisfied. If conditions (5-10) are not satisfied, then the algorithm proposed in [l] can be recommended. The optimization

(25, 26) that is executed by varying V is based, in general, on nonlinear programming. During such an optimization, conditions (16, 17) can be preserved relatively easily. However, the optimization that is executed via controlled object structural changes often is based on the combinatorial approach. During combinatorial optimization, conditions (16, 17) can be unexpectedly violated, since any changes of H create a significant destabilization effect on the object mapping function. To decrease the possibility of violations of (16, 17), the control system should, before changes of H are made, try to drive Y_t into the center of $\Phi(\Theta)$. This can be done, for example, by thorough optimization made via V variations. The occurrence of Y_t near the center of $\Phi(\Theta)$ is an indication that the object has an excess of stability. As a result, the optimization executed via changes of H becomes possible.

When, in our network example, V_{net} provides traffic in each network link which is below link capacitie, then with high level of probability $\Lambda_{out} = \Lambda_{unp}$, i.e. network quality constrains are satisfied. If a routing optimization algorithm allows for the continuous changing of V_{net}, then mapping $F(H, \Lambda_{imp}) : V_{net} \to R$ is a continuous. However, this mapping function is non-separable because traffic in every link consists of traffic flows originated by different sources and destined to different destinations and also because (23) `is a non-linear function.

Different control system modes require the use of different models. Namely, the simple control mode requires the input-control variables-object structure-output mapping model. This model reflects $F(V,H) : X \to Y$ (Fig. 2a) and $F(X,H) : V \to Y$ (Fig. 2b). For the optimized control mode, a control variables-controlled variables-mapping model that reflects $F(X,H) : V \to Q$ (Fig. 2c) should also be used.

For the simple control mode two approaches based either on $V_t H_t = f(Y_t)$ or $V_t H_t = f(X_t)$ can be used. The first approach uses a feedback, i.e., the control system constantly monitors conditions (16, 17)

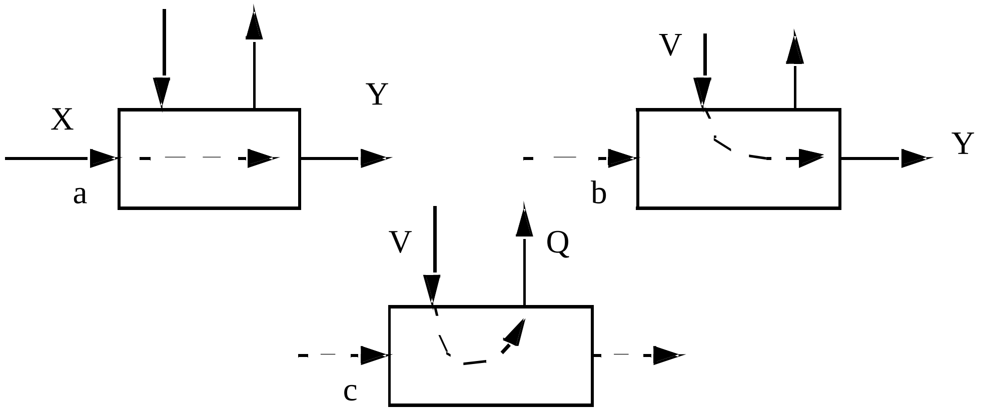

Fig. 2

and corrects output, if necessary, by varying V and H. This approach is relatively accurate. However, it is slow, since control decisions are delayed by an object $X \to Y$ transformation time and by a decision process that requires CPU time. The second $V_t H_t = f(X_t)$ approach (open-loop control) does not use a feedback all the time. Instead, the control system continuously monitors condition $X_t \in \Psi^z$ and makes decisions either that the current $V_t H_t$ combination has to be changed and thus what has to be done to satisfy (16, 17), or that no change has to be made. In other words, the second approach uses rules of this kind: If $X_t \in \Psi^z$ then do nothing. If $X_t \notin \Psi^z$, then find other Ψ^r to which X_t, belongs and change the object in correspondence with $V^r H^r$. This approach is faster than the feedback approach, but requires that *VH=f(X)* (reactions) for $\forall X$ will be prepared in advance. We will study the possibility of using a second (open-loop) controlling approach in combination with self-learning and adaptation.

4. Self-learning and adaptation

The purpose of self-learning is the development and modification of control rules based on cause- and-effect information received via trials. Trials can be made as on analytical models as on the real objects.

Self-learning consists of three phases. Namely:

4.1. The preliminary cause -and-effect trials phase

This phase is dedicated to the analysis of linearity of $F(V, H : X \to Y, F(X, H) : V \to Y$, and $F(X, H) : V \to Q$ and studies where conditions (2-7) are satisfied. Analysis of conditions (8-10) must also be considered.

For the analysis of $F(V, H) : X \to Y$ with fixed *V* and *H*, we will either observe natural *X* fluctuations on the real object, or actively change *X* on the model or on the real object. For each *X*, a value of *Y* is defined. This process is repeated for different *V* and *H*. Similarly, for the analysis of $F(X, H) : V \to Y$ with fixed *X* (if it is possible) and *H*, we will vary *V* and define for each *V* a value of *Y*. This process is repeated for different *X* and *H*. The purpose of this is to check (2, 3, 5, 6, 8, 9).

For the analysis of $F(X, H) : V \to Q$ with fixed *X* (if it is possible) and *H*, we will vary *V* and define for each *V* value of *Q*. This process is repeated for different *X* and *H*. The purpose of this is to check (4, 7, and 10).

The number of such trials is dictated by the desired accuracy of verification of (2-10) and it should be held to the minimum. The results of the first phase are needed for the selection of optimization algorithms used in the control rule development phase.

4.2. The development of the control rules phase

This phase is implemented via a two-step object optimization. During the first step, an *VH* combination (feasible solution) is received for the analyzed *X* in correspondence with (25) (Fig. 3). The second step is the selection of the optimal (in correspondence with (26) *VH* combination for the same *X*.)

Control rule development starts from some *X*. When the optimal *VH* combination is received for this *X*, then a permissible input space Ψ^z for this combination is defined by varying either model or real object inputs (Fig 4)

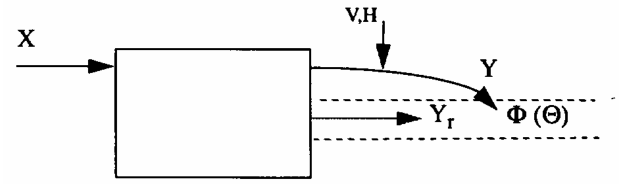

Fig 3

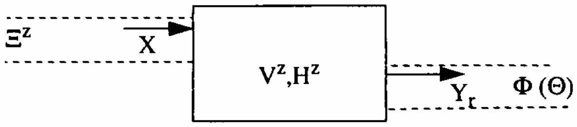

Fig 4

Consequently, we will formulate a rule: "If $X \in \Psi^Z$, then $V^Z H^Z$ combination has to be chosen." Then the two-stage optimization process is repeated for some other $X \notin \Psi^Z$. As a result, we will find other $V^Z H^Z$ combinations and other input subspaces Ψ^R (Fig. 5).

After this, the next *X* (neither $X \notin \Psi^z$ nor $X \notin \Psi^r$) is selected and the process is repeated. Input vectors are selected for the analysis until either the whole input space *A* becomes decomposed or a feasible solution for the same *X* is impossible to find. As a result of decomposition, some input subspaces can intersect; i.e., more than one feasible solution exists for some *X*; i.e., $X \in \Psi_g, \Psi_g = \mathrm{I}_{w \notin W} \Psi_g^W \neq 0$, where Ψ_g represent a *g*-th intersection, *W* is a number of intersected subspaces. If $X_t \in \Psi_g$ and if conditions (3) and (6), or (9) are satisfied, then a combined weighted rule [41 can be executed to provide $Y_t \in \mathrm{q}$.

When the controlled object is non-linear, then it is possible that $|\Psi^z| \neq |\Psi^r|$. As a rule, we should try to avoid intersections because the more input space belongs to the intersections, the more *VH* combinations have to be analyzed. If Ψ^g represents an object stability input space for the *z*-th combination of *VH*, then $\Psi = \mathbf{|}_{z \in Z} \Psi^z$ represents a total object stability input space (*Z* is a number of subspaces received during input space decomposition).

When Ψ =*A*, then the object is stable. If $\Psi \in A$, then the object is only partially stable. Results consisting of optimal *input subspace-object structure-control vector* rules should be stored in the input-reaction table. This table becomes a basis for the open-loop control.

Now, for the illustration purposes, let's try to apply proposed above methodology to the development of control rules for our network control system. Let's also define the network control task as one that provides $R \leq 14.0$ and *min G* when traffic between nodes *BF* and *CE* varies and is fixed for all othe source-destination pairs of nodes. For the developing network control rules, we use the model an optimization algorithm proposed in [l]. This model and optimization algorithm allow for the continuous changing of V_{net}, and they do not require monotonicity or separability of the optimized function. Since we have already assumed the continuity of the $F(\Lambda_{inp} h): V_{net} \rightarrow R$ mappings, we can turn directly to the second phase of the control rules development.

We start the development with traffic values for *B-F* and. *C-E* source-destination pairs which are proposed in [3]; namely *4.0 pkts/sec* for *B-F* and *3.0 pkts/sec* for *C-E*. During the first step of the optimization (which is based on varying V_{net} upon fixed network structure *H*), the procedure described in [l] allows us to find a control vector that provides conditions $\Phi \leq$ 14.0. Such vector `*A*' is presented in Table 1 in the column, `*Curve A traffic (%)*. The value of *R* is 13.56 for this vector, analyzed network structure, traffic and corresponding G = 425. Since R < 14.0, the network has an excess of capacity and its structure can be optimized.

During the second step of optimization, the network structure (link capacities) was changing. However, control vector V_{net} was fixed. As a result of this optimization, the capacities of the *EF* and *FE* lines were decreased from 62.5 to 42.63 pkts/sec. That corresponds to G = 385.26 and R = 14.0.

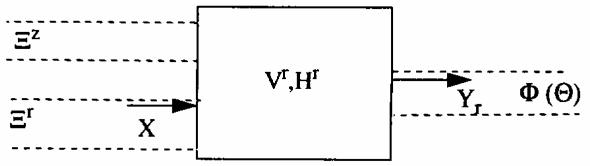

Fig. 5

Then permissible input spaces were defined for the original network structure and for the optimized one. The first was done on a fixed network structure by varying *B-F* and *C-E* traffic, while monitoring $R \le 14.0$ conditions. We started with the control vector that is optimal for *B-F* traffic that is equal to 4.0 pkts/sec and *C-E* traffic that is equal to 3.0 pkts/sec. Modelling allows us to plot curve "*A*" on Figure 6. The zone under this curve is a permissible input space and it defines possible combinations of *B-F* and *C-E* traffic for which the analyzed control vector provides $R \le 14.0$. In other words, a control rule *`Until B-F and C-E traffic is in the zone below curve `A', the set of routing tables, corresponding to `Curve A traffic (%I' of the Table 1, should be used'* can be applied.

We can see from curve `*A*' that the maximal *B-F* traffic is limited to *9.66 pkts/sec* and *C-E* traffic is limited to *8.12 pkts/sec*. To analyze the network's ability to absorb more *B-F* traffic, we can choose a value of *B-F* traffic which is above *9.66 pkts/sec*. Then we can find the other control vector that provides *R* s 14.0 condition. This vector is presented in Table 1, column `*Curve B traffic (%)*'. Analysis of the network with a new control vector allows us to plot curve `*B*' in Figure 6. We repeated similar procedures for the *C-E* traffic that exceeds *8.12 pkts/sec*. This give us one more control vector (Table 1, column `C) and another curve ` *C*'.

As a result of these studies, the following rules can be created: *`Until B-F and C-E traffic is in the zone below curve `A', the set of routing tables corresponding to `Curve A traffic (70)' should be used. If traffic is in zone I, then use the set of routing tables corresponding to `Curve B traffic (o/o)`. If traffic is in zone II, then use the set of routing tables corresponding to `Curve C traffic (%I'*. These rules apply to the initial *H* that corresponds to *G* = 425.

A similar study was done with control vector `A' and the network in which the capacities of lines *EF* and *FE* were decreased to *42.63 pkts/sec*. As a result a curve $A_{opt}$ was plotted. The previous rule can be modified by adding the following: *`If B-F and C-E traffic is in the zone below the curve Aopt, then routing tables corresponding to `Curve A traffic (%)' should be used and the capacity of lines EF and FE can be decreased to 42.63 pkts/sec'*.

4.3. The adaptation phase

Because an object and/or object environment usually are non-stationary, the control system's ability to control and its performance efficiency degrade with time. To maintain controlled object performance at the predefined level, the control system should constantly monitor the validity of the developed rules and the values of *G*. The purpose of monitoring is to detect a moment when control becomes inefficient. When control system inefficiency is detected, another self-learning process can be repeated.

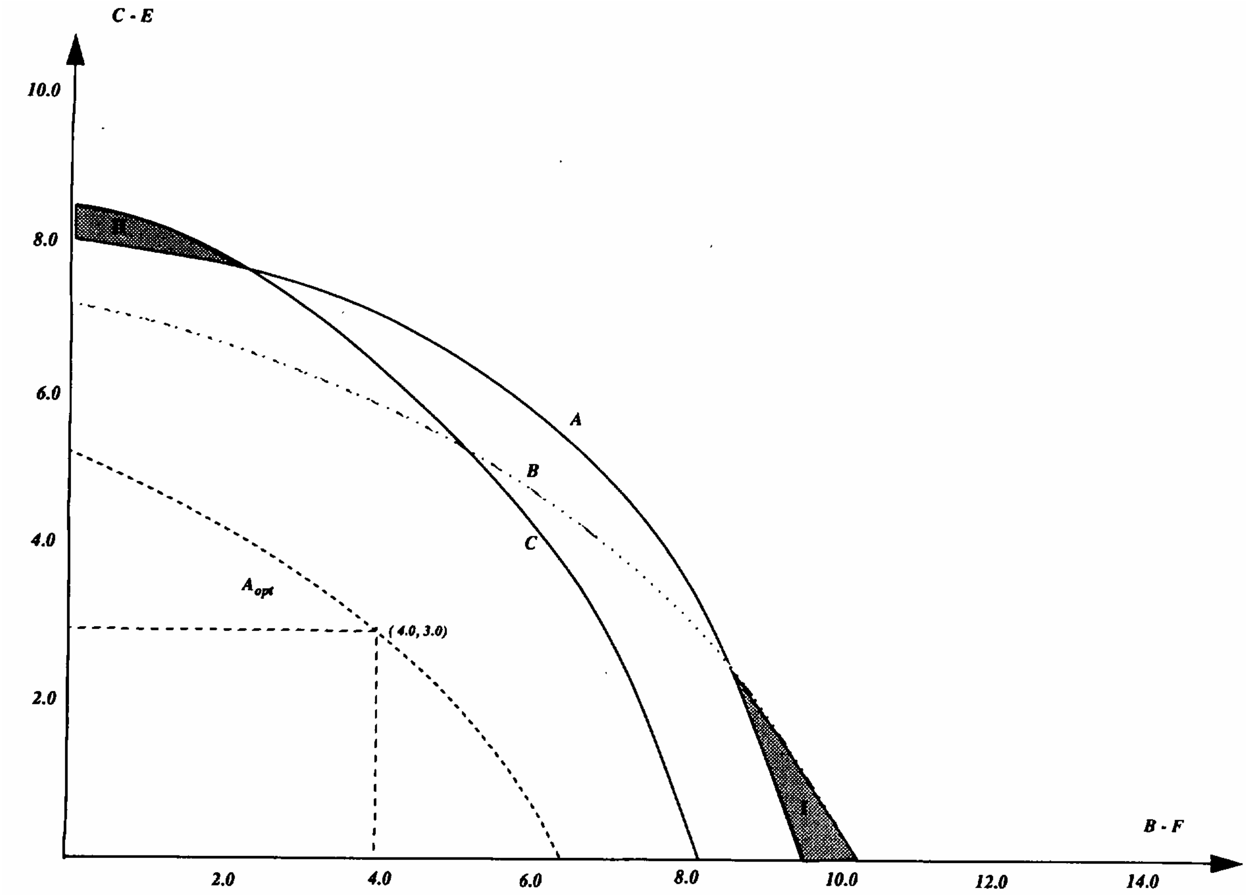

Fig. 6

5. Conclusion

Usage of proposed approach to the telecommunications network model and received results demonstrate feasibility of design of the open-loop control systems for objects with multiple inputs, outputs and complex structures. Development of such systems can significantly decrease control reaction time, i.e. improve quality of the control.

References

[l] Flikop, Z., "Routing optimization in packet switching communication networks," European Journal of Operational Research 19 (1985) 262-267.
[2] Sugeno, M., "An introductory survey of fuzzy control", .Inforrnation Sciences (1985) 59-83.
[3] Tanenbaum, A.S., Computer Networks, Second edition, Prentice Hall, New Jersey, 1988.
[4] Yager, R.R. and Filev, D.P., "Identification of nonlinear systems by fuzzy models" (forthcoming).
[5] Zadeh, L.A. "The concept of a linguistic variable and its application to approximate reasoning", Information Sciences 8