



(72) Owens, Aaron James, US

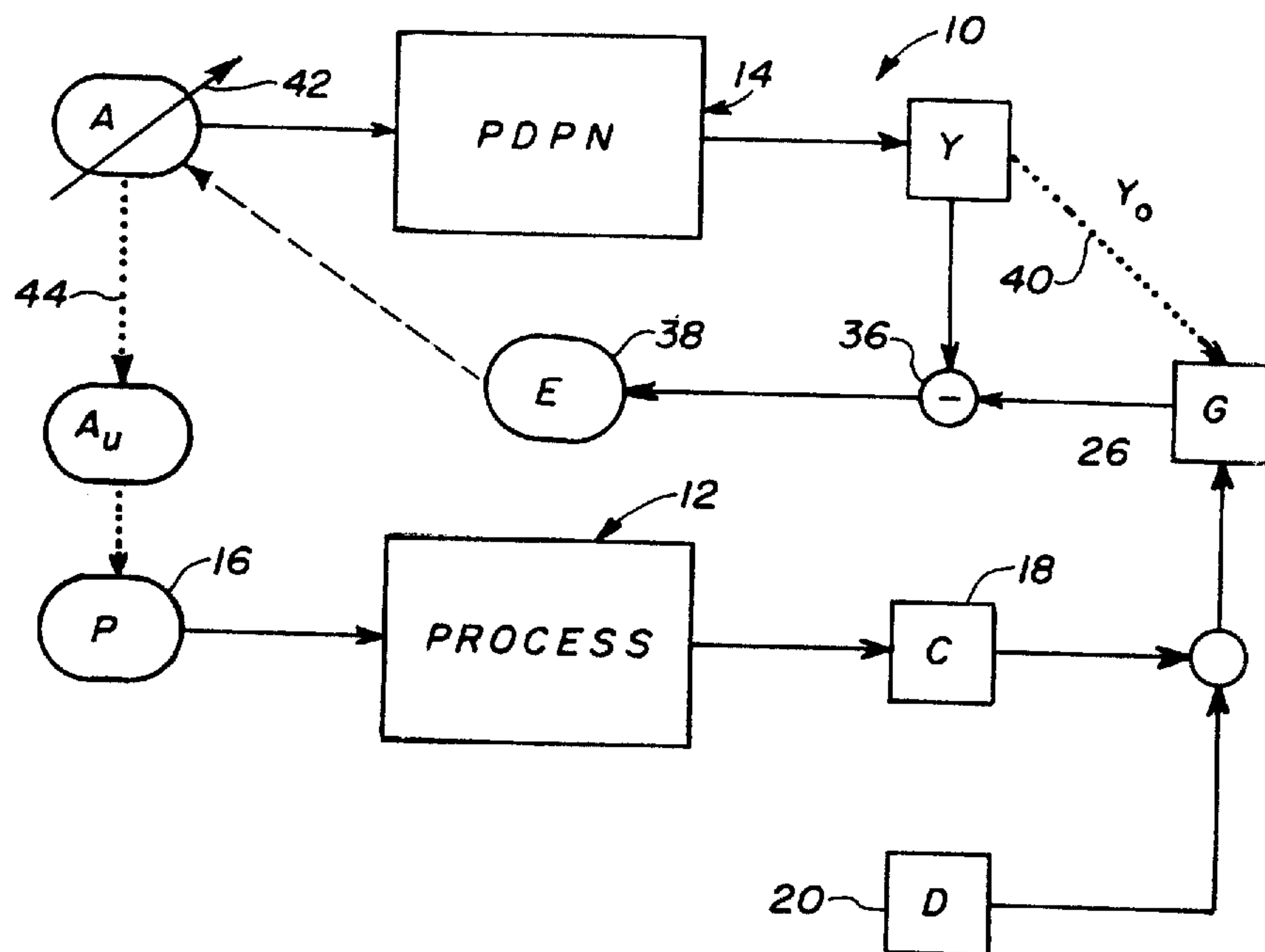
(73) E.I. DU PONT DE NEMOURS AND COMPANY, US

(51) Int.Cl.⁵ G06F 15/46

(30) 1989/02/28 (316,717) US

(54) **APPAREIL ET METHODE POUR CONTROLER UN
PROCESSUS AU MOYEN D'UN RESEAU DE TRAITEMENT
REPARTI PARALLELE**

(54) **APPARATUS AND METHOD FOR CONTROLLING A PROCESS
USING A TRAINED PARALLEL DISTRIBUTED PROCESSING
NETWORK**



(57) A parallel distributed processing network, previously trained to simulate a process, is responsive to input activations A to produce outputs Y. The activations are representative of the input parameters P of the process and the outputs are representative of the characteristics C of the product of the process. A set of goal signals G corresponding to the desired physical characteristics D of the product are produced and used to generate a signal representative of the error E between the goal signals G and the outputs Y of the network. The values of the updated inputs A_U of the network that are needed to drive the error E to a predetermined minimum are determined, and the actual physical input parameters P of the process changed to correspond to the updated inputs values A_U producing the minimum error. The present invention can be used in both a batch initialization situation and/or in an on-going process control situation.

5

TITLE

APPARATUS AND METHOD FOR CONTROLLING A PROCESS USING
A TRAINED PARALLEL DISTRIBUTED PROCESSING NETWORK

10

ABSTRACT OF THE INVENTION

A parallel distributed processing network, previously
trained to simulate a process, is responsive to input activations
A to produce outputs Y. The activations are representative of
15 the input parameters P of the process and the outputs are
representative of the characteristics C of the product of the
process. A set of goal signals G corresponding to the desired
physical characteristics D of the product are produced and used
to generate a signal representative of the error E between the
20 goal signals G and the outputs Y of the network. The values of
the updated inputs AU of the network that are needed to drive
the error E to a predetermined minimum are determined, and
the actual physical input parameters P of the process changed
to correspond to the updated inputs values AU producing the
25 minimum error. The present invention can be used in both a
batch initialization situation and/or in an on-going process
control situation.

2011135

1

ED-0373

TITLE5 APPARATUS AND METHOD FOR CONTROLLING A PROCESS USING
A TRAINED PARALLEL DISTRIBUTED PROCESSING NETWORK

10 A portion of the disclosure of this patent document
contains material which is subject to copyright protection. The
copyright owner has no objection to the facsimile reproduction
by anyone of the patent document or patent disclosure, as it
appears in the Patent and Trademark Office patent file or
records, but otherwise reserves all copyright rights
whatsoever.

15

BACKGROUND OF THE INVENTIONField of the Invention

20 The present invention relates to a method and an
apparatus for controlling a process, and, in particular, to such a
method and an apparatus that makes use of the information
stored in a parallel distributed processing network that is
previously trained to simulate the process.

25

Description of the Prior Art

30 Control of a manufacturing process may be broadly
divided into two classes of activity: (1) determining the initial
parameter settings or initial constituent concentrations for a
process ("batch initialization"); and (2) ongoing adjustment of
parameter settings ("on-going process control"). The latter
class of activity may itself be subdivided into those
adjustments that are able to be made without reference to the
35 history of the process adjustments (termed "static"
adjustments) and those adjustments which are dependent upon

2011135

the preceding process adjustments (termed "dynamical" adjustments). These activities are interrelated in the sense that an initial setting of a parameter or initial concentration of a constituent may be viewed as an adjustment of the variable
5 from its random position or from an undetermined concentration whereas an ongoing adjustment may be viewed as a timed sequence of initial parameter settings.

There is a prevalent attitude that a so-called parallel
10 distributed processing network ("PDPN") may have utility in supervisory control of physical processes. For example B. Widrow discussed the control of two physical processes with feed-forward neural networks at the December 1988 meeting "Neural Information Processing Systems", Denver, Colorado.

15 In such a role the parallel distributed processing network is used as a conceptual model that relates the process input parameters to the output characteristics of the physical process in a feed forward manner. This arrangement is satisfactory if it is desired merely to know how certain predetermined
20 alterations of the inputs would affect the characteristics of the output product. In process control situations, however, what is desired is to know *a priori* is exactly what the input parameters should be in order to produce an output product having desired characteristics.

25 Accordingly, in view of the foregoing it is believed desirable to provide a control arrangement using a parallel distributed processing network that would enable an operator to determine what input parameters of a physical process are needed in order to obtain predetermined characteristics in the
30 product of the process, whether used in batch initialization situation or on-going process control.

2011135

3

SUMMARY OF THE INVENTION

5 The present invention relates to a method and apparatus
for controlling a physical process of the type in which
predetermined inputs having predetermined physical
parameters P are used to produce a product having physical
characteristics C associated therewith. A parallel distributed
processing network, previously trained to simulate the process,
10 is responsive to input activations A to produce outputs Y. The
activations are representative of the input parameters P of the
process and the outputs are representative of the
characteristics C of the product of the process.

In accordance with the invention a set of goal signals G
15 corresponding to the desired physical characteristics D of the
product of the process are produced and used to generate a
signal representative of the error E between the goal signals G
and the outputs Y of the network. The values of the updated
inputs A_U of the parallel distributed processing network that
20 are needed to drive the error E to a predetermined minimum
are determined, and the actual physical input parameters P of
the process are changed to correspond to the updated inputs
values A_U producing the minimum error.

The present invention can be used in both a batch
25 initialization situation and/or in an on-going process control
situation, whether for static or dynamical adjustments.

2011135

4

BRIEF DESCRIPTION OF THE DRAWINGS

The invention will be more fully understood from the following detailed description thereof, taken in connection with the accompanying drawings, which form a part of this application and in which:

Figure 1 is a block diagram of a process control arrangement in accordance with the present invention for controlling a physical process using a parallel distributed processing network which has been previously trained to simulate the process;

Figure 2 is a block diagram generally similar to Figure 1 of a process control arrangement in accordance with the present invention for controlling a time dependent physical process using a parallel distributed processing network which has been previously trained to simulate the process;

Figure 3 is a block diagram illustrating the training phase of a parallel distributed processing network to simulate the time dependent physical process illustrated in Figure 2;

Figure 4 is a stylized diagram of a parallel distributed processing network using in the preferred implementation of the present invention shown in Figures 1 and 2;

Figure 5 is a generalized representation of a processing element used in the parallel distributed processing network shown in Figure 4;

Figures 6 and 7 are, respectively, stylized diagrams of physical processes used in the Examples of the present invention discussed herein;

Figures 8 and 9, respectively, show the time evolution of the controlled output (temperature) and the input parameter (voltage) for an Example based on the process illustrated in Figure 7.

The Appendix containing Listing 1 (Pages A-1 to A-8) and Listing 2 (Pages A-9 to A-10) is attached to and forms part of this application.

2011135

5

Listing 1 (Pages A-1 to A-8) is the source code, in the Fortran language, of a computer program which implements the method of this invention. Listing 2 (Pages A-9 to A-10) gives the data files and a sample run for the implementation of an example application using the process shown in Figure 6.

DETAILED DESCRIPTION OF THE INVENTION

Throughout the following detailed description similar reference numerals refer to similar elements in all figures of the drawings.

Shown in Figure 1 is a block diagram of a process control arrangement generally indicated by the reference character 10 in accordance with the present invention for controlling a physical process using a parallel distributed processing network ("PDPN") which has been previously trained to simulate the process. In Figure 1 the physical process is generally indicated by the reference character 12 while the parallel distributed processing network is indicated by the character 14. The network 14 is responsive to a set of one or more inputs, or activations A, which produce a corresponding set of one or more responses Y predicting the actual response of the process 12 to various sets of activations.

The process 12 can, within the contemplation of the present invention, be any physical process acting upon any predetermined number of physical input parameters P, indicated in the block 16, to produce corresponding physical output products 18. The outputs 18 have actual physical characteristics C associated therewith.

In practice the actual physical characteristics C associated with the products 18 of the process 12 may differ in one or more respects from the set 20 of desired physical characteristics D. If signals representing the actual physical characteristics C and the desired physical characteristics D are not equal a set 26 of goal signals G may be generated. These

goal signals G represent the target responses for the network 14 which correspond to the desired characteristics D of the products of the process 12, in a manner to be described below.

5 The parallel distributed processing network 14 contains at least two layers of processing elements. The processing elements are interconnected in a predetermined pattern and with predetermined connection weights therebetween. The feed forward network 14 has been previously trained to simulate the response of the process 12 to variation of the
10 inputs 16 thereto. When trained, the connection weights between the processing elements in the network 14 contain information regarding the process in distributed form throughout the network. The network 14 may be trained using any one of various training techniques, such as the Generalized
15 Delta Rule described in Rumelhart, Hinton and Williams, "Learning Internal Representations by Error Propagation", Parallel Distributed Processing, Volume 1, Foundations, Rumelhart and McClelland, editors, MIT Press, Cambridge, Massachusetts (1986). This rule is a generalization of the
20 "Delta Rule" given by Widrow, "Generalization and Information Storage In Networks of ADELINe Neurons", Self-Organizing Systems, Yovitts, editor, Spartan Books, New York (1962). Alternatively, the network may be trained using the technique disclosed and claimed in copending application Serial Number
25 07/285,534, filed December 16, 1988 and assigned to the assignee of the present invention.

In accordance with this invention means generally indicated by reference character 36 is provided to generate a set 38 of error signals E representative of the error between
30 the set of goal signals G and the responses Y of the network 14. Symbolically,

$$E = (Y - G)^2 \quad (1A)$$

35 where $G = D + (Y_0 - C) \quad (1B)$

2011135

7

where Y_0 is the output of the network 14 when the activations A_0 are set equal to the input parameters P . The dotted connection line 40 between Y and G is meant to emphasize the fact that it is only the initial values Y_0 that are used in setting the goal G . If the network 14 simulates the process 12 accurately, then effectively $Y_0 = C$, and the goal G is equal to the desired process characteristics D .

The activations A are then altered as illustrated diagrammatically at 42, to determine the updated set of activations A_U of the network 14 that are needed to drive the error E to a predetermined minimum, typically zero or some predetermined threshold. It should be understood that some of the activations A may be constrained within predetermined intervals.

The actual physical parameters P of the process 14 are then set equal to the activations A_U that produce the minimum error E . This step is indicated by the dotted connection 44. The dotted line between A and P is meant to emphasize the fact that it is only the updated values A_U that are used to set the input process parameters P .

Although the use of the updated activations A_U in the manner set forth above to an on-going process control situation may, perhaps, be more easily comprehended, changing the actual physical input parameters of the process to correspond to the updated activations A_U that produce the minimum error E should be understood to apply as well to the various facets the batch initialization situation.

In one facet of a batch initialization situation, each control device of a system (e.g., a valve) must, of necessity, have some pre-existing setting. The pre-existing setting is typically randomly chosen, and usually has no relationship whatsoever to the particular manufacturing control environment in which the device is being used. In another facet of a batch initialization situation, e.g., determining the

2011135

8

initial concentration of color constituents as in the formulation of paint, the initial concentrations of constituents of the mixture is undetermined. Nevertheless, changing the setting of a control device from its pre-existing random setting to some
5 initialization setting or determining the initial concentrations of constituents to form a mixture should be understood as uses of the updated activations A_U and are accordingly encompassed within the scope of the present invention.

The activations A_U may be determined by any
10 predetermined error minimization technique, for example, the Levenberg-Marquardt algorithm (e.g., the subroutine ZXSSQ from the IMSL Library, sold by IMSL, Inc. Houston, Texas), or the conjugate gradient method (e.g., the subroutine UMCGG from the IMSL Library).

15 An alternative technique to determine the activations A_U , to be discussed in more detail herein, involves "training" the inputs A using stiff differential equations. In other words, the network is, in effect, trained on its inputs to send the response Y within some predetermined range of the goals G . If the goal
20 G is feasible within the constraints on the activations A the error converges toward zero. If, in addition, the network simulation of the process is highly accurate so that the output of the network 14 is equal to the output of the physical process 12, setting the input process parameters P equal to updated
25 activations A_U results exactly in the desired output characteristics D . If the network simulation model is less accurate and Y_0 does not equal C , setting the input process parameters P equal to the updated activations A_U will move the process characteristics C toward the desired output D .

30 Once the parameters P have been set equal to A_U , the characteristics C of the products 18 of the process 12 are checked. If the actual characteristic C are within a predetermined acceptable range of the desired characteristics D then the control cycle is complete. If, on the other hand, after
35 alteration the difference between the actual characteristics C

2011135

9

and the desired characteristics D is still above some predetermined amount, the control loop process is iterated as necessary to bring the actual characteristics into an acceptable range of difference with the desired characteristics.

5

-o-o-o-

In the process control arrangement shown in Figure 1, it is implicitly assumed that there is a direct and immediate (i.e., static) one-to-one correspondence between the process control parameters P and the output properties C , regardless of the time history of the process. In some processes, there are dynamical (i.e., time-dependent) relationships between P and C , so that the output C depends upon the past time history of the process parameters P . It should be understood that the present invention is equally applicable to the control of dynamical as well as static systems. Figure 2 illustrates a block diagram of a process control arrangement generally indicated by the reference character 10' in accordance with the present invention for controlling a time dependent physical process 12'.

Figure 3 is a block diagram illustrating the training phase of a parallel distributed processing network 14' to simulate the time dependent physical process 12' illustrated in Figure 2. For a dynamical system, the process parameters $P[t]$ and outputs $C[t]$ at the present time are augmented by those for the previous time $t-1$ (denoted $P[t-1]$ and $C[t-1]$, respectively) and up to m other previous times are all used as activations to the network 14'. The outputs to which the network 14' is trained are the process properties $C[t+1]$ at the next time. If prediction past the next time step is desired, additional outputs $C[t+2]$, $C[t+n]$ are added to the training set. Each available sample time t provides one exemplar in the training data set.

During training, the augmented time-dependent data set is presented to the network 14', and the connection weights W of the network 14' are adjusted to minimize the error in the

35

2011135

10

prediction of the outputs $C[t+1]$ as usual. When training is complete, the weights W contain the information for a fully dynamical simulation of the time dependent physical system 12'.

5 During the process control phase illustrated in Figure 2, the connection weights W of the network 14' are fixed, and in order to minimize the difference E between the model output $Y[t+1]$ and the goal G at the next time step, the process control parameters $P[t]$ at the present time are varied. The other
10 activations to the network ($C[t]$, $P[t-1]$, $C[t-1]$, ... $P[t-m]$, $C[t-m]$) are fixed at their observed values during the variation of the parameters $P[t]$.

 Once the updated process inputs parameters $P_U[t]$ which minimize the error E are determined, the process parameters P
15 are set equal to $P_U[t]$, as shown by 44' in Figure 2. In this way, the network 14' can be used for the supervisory control of any dynamical process.

-o-0-o-

20 With reference to Figure 4 shown is a stylized diagram of a preferred form of the distributed parallel processing network 14 which the present invention shown in Figures 1 and 2 uses. The network 14 is configured as a back propagation network and comprises individual processing elements 52 configured
25 with at least a first layer 54 and a second layer 58. The layer 54 is termed the "input layer", and the layer 58 is the "output layer". In addition there may zero or more intermediate layers called "hidden layers". In Figure 4 shown is one hidden layer 56. The processing network 14 is organized such that the input
30 layer 54 contains a set of at least one to i processing elements, the hidden layer 56 has a set of at least one to k processing elements, and the output layer 58 has a set of at least one to j processing elements therein. When implementing the network an operator would specify the values i , k and j .

2011135

11

Figure 5 is a generalized representation of a processing element 52 used throughout the network 14. Each processing element 52 has an input port P_{in} and an output port P_{out} . The processing element is responsive to one or a plurality of excitation signal(s) I_1 through I_m presented at the input port P_{in} and is operative to produce an activation signal Q_{out} carried on a line L_{out} connected at the output port P_{out} . Each of the excitations I_1 through I_m is connected to the input port P_{in} of the element 52 over a respective line L_1 through L_m that has a predetermined respective connection weight W_1 through W_m . The activation signal Q_{out} that is produced on the line L_{out} at the output port P_{out} of the processing element 52 is a function of the input signal Q_{in} applied at the input port P_{in} . The input signal Q_{in} is the summation, over all of the input lines to the processing element, of the inner product of the strength of a given excitation signal I and the connection weight W of the line L carrying that signal to the processing element, scaled by the squashing function S for that processing element. Symbolically, the activation signal Q_{out} at the output port P_{out} is functionally related to the input signal Q_{in}

$$\text{Signal on line } L_{out} = Q_{out} = S(Q_{in}), \tag{2}$$

$$\text{where } Q_{in} = \sum_{z=1}^m W_z I_z + T \tag{3}$$

where T is the threshold of the processing element 52. The threshold T can be considered as an additional weight from an input line $(m + 1)$ having a constant excitation signal of one (1), similar to the prior art. The squashing function S can take the form of the sigmoid function,

$$S = [1 + \exp(-Q_{in})]^{-1} \tag{4}$$

2011135

12

although any other desired monotonically nondecreasing squashing function may be used. If some other squashing function is used the detailed forms of the Equations (6) and (7) would, of course, be different from those set forth hereinafter. 5 However, the teachings of the present invention would have equal applicability thereto.

As may be seen from Figure 4 each of the processing elements 54-1 through 54-i in the input layer 54 is connected to at least one of a predetermined number i of network input signals on a respective input line 62. The signals on the input lines 62 correspond to the activations of the network 14 discussed earlier. Each of the lines 62 may have a predetermined weight associated therewith, similar to the situation discussed above in the case of the generalized processing element. Each of the processing elements in the input layer 54 is operative, in a manner similar to that described above for the generalized processing element, to produce a respective activation signal $A1[1]$ through $A1[i]$ at the output port thereof. 10 15

The output port of each of the i processing elements in the input layer 54 is connected to at least one of the k processing elements 56-1 through 56-k in the intermediate layer 56 of processing elements by a respective line 64. In a typical case, the output port of each of the processing elements in the input layer 54 is connected to the input port of each of the processing elements in the intermediate layer 56. Each such interconnection line 64 has a connection weight $W1[i,k]$ associated therewith. 20 25

In a fashion similar to that discussed above the processing elements in the intermediate layer 56 respond to the inputs applied thereto to produce an activation signal $A2[k]$ at the output port thereof. In similar fashion, the output port of each of the k processing elements in the intermediate layer 56 is connected to the input port of at least one of the j processing elements in the output layer 58 of processing 30 35

2011135

13

elements on a line 66. In the preferred case the output port of each element in the intermediate layer 56 is connected to the input port of one of the processing elements in the output layer 58. The lines 66 each have a connection weight $W2[k,j]$

5 associated therewith.

The processing elements in the output layer 58 respond to the inputs thereto to produce an activation signal $A3[j]$ at the output port thereof. These output signals are carried over respective output lines 68. The output signals on the lines 68
10 correspond to the outputs Y discussed earlier.

-o-0-o-

With such a network as that shown in the Figure 4, the
15 error signal E in Figure 1 is given by

$$E = \sum_j (Y[j] - G[j])^2 \quad (5)$$

20 where $G[j]$ is element j of the goal vector. The symbol A in Figure 1 corresponds to $A1$ in Figure 4 and similarly Y in Figure 1 corresponds to $A3$ in Figure 4.

If the required activations $A1[i]$ of the network 14 are obtained by training using a method similar to the conventional
25 Generalized Delta Rule using the method of steepest descent, then

$$\Delta A1[i] = \sum_k W1[i,k] A2[k](1 - A2[k]) \times$$

$$30 \sum_j (A3[j] - G[j]) A3[j] (1 - A3[j]) W2[k,j] \quad (6)$$

and training is accomplished iteratively by setting

$$35 \quad A1[i]^{n+1} = \text{ETA} \Delta A1[i] + A1[i]^n$$

2011135

14

where ETA is the training rate.

Alternatively the required activations A1[i] can be obtained by using standard optimization techniques such as a conjugate gradient technique or the Levenberg-Marquardt algorithm to find the values of A1[i] yielding the minimum error E in Equation (5).

In the preferred implementation the required activations A1[i] are obtained from the differential equations (7):

$$\begin{aligned}
 10 \quad d/dt(A1[i]) = & \sum_k W1[i,k] A2[k](1 - A2[k]) \times \\
 & \sum_j (A3[j]-G[j]) A3[j] (1 - A3[j]) W2[k,j] \quad (7)
 \end{aligned}$$

15

Equation (7) is a set of nonlinear ordinary differential equations for the continuous variables A1[i]. The initial conditions are $A1 = A_0 = P$. This system of equations may be integrated numerically to solve the equations, such that, as time increases, the activations A1 converge to steady state values corresponding to those giving a local minimum least-squares prediction error. In the preferred instance, the equations are numerically solved using the stiff integration package DGEAR from the IMSL Library. The original version of the algorithm used in the package DGEAR is disclosed in Gear, C. W., Numerical Initial Value Problems in Ordinary Differential Equations, Prentice-Hall, Englewood Cliffs, New Jersey (1971).

Because the differential Equation (7) are formulated to decrease the mean square prediction error, in the direction of steepest descent, the prediction error will always decrease.

-o-0-o-

2011135

15

It is often desirable or sometimes necessary to constrain some or all of the activations A to remain within a fixed range of values,

$$\text{Min}[i] < A[i] < \text{Max}[i] \quad (8)$$

5

Regardless of what method is used to minimize the error E, the constraint (8) can be generated as follows.

In minimization the variable Z (which has no limits) is used, where

10

$$Z[i] = -\log \frac{\text{Max}[i] - \text{Min}[i]}{A[i] - \text{Min}[i]} - 1 \quad (9A)$$

15

and log is the natural logarithm. As Z[i] varies over the entire range from - infinity to + infinity the activation

$$A[i] = \text{Min}[i] + \frac{\text{Max}[i] - \text{Min}[i]}{(1 + e^{-Z[i]})} \quad (9B)$$

20

only varies within the range given in Equation (8). In this manner range constraints on the activations can be maintained.

25

-o-O-o-

A specific implementation of the method for finding the input activations AU which minimize the error E between the goal G and the outputs Y of a three-layer back propagation network is shown in the VAX Fortran PROGRAM SET of Listing 1. This program determines the input activations AU corresponding to a desired set of output characteristics G. For a batch initialization situation, as shown by the program in

30

2011135

17

initial conditions with the minimum RMS error are used. In lines 151-160, the activations are scaled to the unrestricted variables Z used internally via Equation (9A).

5 Lines 162-176 initialize some values to be used in the subroutine DGEAR which solves the differential Equations (7). Lines 177 to 202 are a loop that is repeatedly executed until the solution reaches the specified tolerance TOLER or maximum number of effective training presentations TMAX. Inside this loop, subroutine DGEAR is called to advance the solution the
 10 differential equations (7) from time T to time TEND, with both T and TEND being increased on each pass through the loop. The IMSL subroutine DGEAR calls the subroutine DIFFUN which supplies the time derivatives dZ/dt , as discussed below. Information on the RMS error after each call to DGEAR in
 15 printed to the computer screen (in line 193) for monitoring purposes.

Once a solution has been obtained to specified tolerance, the program SET does a forward pass through the network in lines 205-242 to compute the outputs Y corresponding to the
 20 updated activations AU. These outputs Y, the output goal G, and the updated activations AU are printed to the screen in lines 243-260, and the activations are written to the file SET.OUT in line 261 before the program stops.

The subroutine DIFFUN in lines 270-412 is called by
 25 DGEAR to evaluate the time derivatives $DZ = dZ/dt$ as a function of the time T and the variables Z. Recall that the variables Z are the values of the activations scaled by Equation (9A). Lines 271-291 give the parameters, vectors, arrays, and common blocks used in the subroutine. In lines 292-312, the internal
 30 variables Z are converted using Equation (9B) to activations A, denoted in the program by TRIN. Lines 213-389 are a computation of the activations (A2, A3 in Figure 4) in the network based on the input activations TRIN, along with an evaluation of the time derivatives of the input activations
 35 given in Equation (7). Finally, lines 394-408 convert from

dA/dt to dZ/dt, which is the output of DIFFUN required by DGEAR.

-o-o-o-

5

The invention may be more fully appreciated from the following examples.

EXAMPLE 1

10

The first example under consideration involves a paint formulation process, shown in Figure 6, in which plural input lines, each carrying a predetermined different color of paint, are fed into a mixing vat. A single output line carries the output product, the paint mixture. One of the input lines carries white paint, a second black paint, a third violet paint and a fourth magenta paint. When a panel is coated with the resulting output mixture, the properties of brightness (L^*), red-to-blue color (a^*) and yellow-to-green color (b^*) can be measured. The example, including Tables 1 and 2 set forth herein, is derived from the article by Alman and Pfeifer in Color Research and Applications, Volume 12, Number 4, pages 210 to 222, August 1987. The entry in sample 10 is believed to have been printed with the incorrect sign, which has been corrected in Table 1, as noted by the asterisk.

15

20

25

2011135

19

TABLE 1

		Column Concentrations				Color response		
Sample	White (C _w)	Black (C _b)	Violet (C _v)	Magenta (C _m)	L*	a*	b*	
5	Estimation samples							
	1	1.00	0.00	0.00	0.00	95.29	-.75	1
	2	0.75	0.25	0.00	0.00	68.28	-.85	-2
	3	0.50	0.50	0.00	0.00	54.85	-.87	-2
	4	0.25	0.75	0.00	0.00	39.58	-.79	-3
10	5	0.00	1.00	0.00	0.00	2.91	-.05	-2
	6	0.75	0.00	0.25	0.00	58.48	19.59	-32
	7	0.50	0.00	0.50	0.00	44.62	25.32	-38
	8	0.25	0.00	0.75	0.00	29.92	29.75	-41
	9	0.00	0.00	1.00	0.00	2.73	7.23	-4
15	10	0.75	0.00	0.00	0.25	59.15	43.27	-15
	11	0.50	0.00	0.00	0.50	47.59	53.13	-13
	12	0.25	0.00	0.00	0.75	36.60	57.13	-7
	13	0.00	0.00	0.00	1.00	13.34	39.94	11
	14	0.00	0.75	0.25	0.00	2.38	1.94	-3
20	15	0.00	0.50	0.50	0.00	2.28	3.42	-3
	16	0.00	0.25	0.75	0.00	2.38	4.95	-4
	17	0.00	0.75	0.00	0.25	2.38	7.30	-3
	18	0.00	0.50	0.00	0.50	3.08	13.71	-2
	19	0.00	0.25	0.00	0.75	5.16	20.71	.
25	20	0.00	0.00	0.75	0.25	2.53	10.96	-5
	21	0.00	0.00	0.50	0.50	3.34	14.72	-4
	22	0.00	0.00	0.25	0.75	4.58	20.21	-2
	23	0.50	0.25	0.25	0.00	48.47	11.43	-21
	24	0.25	0.50	0.25	0.00	34.57	8.14	-16
30	25	0.25	0.25	0.50	0.00	31.52	17.49	-28
	26	0.50	0.00	0.25	0.25	42.45	31.97	-29
	27	0.25	0.00	0.50	0.25	28.19	31.11	-33
	28	0.25	0.00	0.25	0.50	29.43	36.71	-25
	29	0.50	0.25	0.00	0.25	46.31	26.08	-15
35	30	0.25	0.50	0.00	0.25	33.15	19.05	-12
	31	0.25	0.25	0.00	0.50	31.99	34.59	-14
	32	0.00	0.50	0.25	0.25	2.19	8.09	-3
	33	0.00	0.25	0.50	0.25	2.33	8.80	-4
	34	0.00	0.25	0.25	0.50	2.76	13.35	-4
40	35	0.25	0.25	0.25	0.25	30.19	23.64	-23
	36	0.25	0.25	0.25	0.25	30.11	23.66	-23
	37	0.25	0.25	0.25	0.25	30.05	23.45	-22
	38	0.75	0.12	0.13	0.00	62.19	9.54	-18
	39	0.12	0.75	0.13	0.00	25.23	3.12	-8
45	40	0.12	0.13	0.75	0.00	19.82	22.73	-32
	41	0.75	0.12	0.00	0.13	60.28	22.52	-13
	42	0.12	0.75	0.00	0.13	23.85	8.33	-8
	43	0.12	0.13	0.00	0.75	23.69	41.21	-9
	44	0.75	0.00	0.12	0.13	57.33	26.52	-24
50	45	0.12	0.00	0.75	0.13	18.73	28.53	-34
	46	0.12	0.00	0.13	0.75	22.08	40.28	-14
	47	0.00	0.75	0.12	0.13	2.20	4.84	-3

20 2011135

TABLE 1. cont'd

Colorant Concentrations					Color response		
Sample	White (C _w)	Black (C _b)	Violet (C _v)	Magenta (C _m)	L*	a*	b*
Estimation samples							
48	0.00	0.12	0.75	0.13	2.19	8.01	-5
49	0.00	0.12	0.13	0.75	4.62	19.82	-2

TABLE 2

Colorant Concentrations					Color response		
Sample	White (C _w)	Black (C _b)	Violet (C _v)	Magenta (C _m)	L*	a*	b*
Checkpoint samples							
50	0.50	0.17	0.17	0.16	45.28	22.09	-22
51	0.17	0.50	0.17	0.16	26.32	14.92	-16
52	0.17	0.17	0.50	0.16	23.79	23.52	-28
53	0.17	0.17	0.16	0.50	24.53	31.31	-19
54	0.34	0.33	0.33	0.00	38.98	12.56	-22
55	0.00	0.34	0.33	0.33	2.29	10.05	-4
56	0.34	0.00	0.33	0.33	33.98	33.25	-29
57	0.34	0.33	0.00	0.33	37.46	27.13	-15

2011135

The first 49 paint mixtures in Table 1 are used to train a three-layer back propagation network (i.e., one hidden layer) used as the PDPN in Figure 1. The PDPN has four input nodes ($i = 4$), one each for the input concentrations of the colors white, black, violet, and magenta. The three output nodes ($j = 3$) represent the color indices L^* , a^* , and b^* corresponding to the mixed paint. Eight hidden nodes ($k = 8$) were used in the PDPN model. Scaled units were used internally in the training and process control models, with each of the inputs and outputs individually scaled by linear transformation to place their minimum value in the training set at 0.1 and their maximum value in the training set at 0.9. As with Example 2, training was conducted on a CRAY X-MP/28 supercomputer.

The complete data files and a sample run of the process control program SET are shown in Listing 2 (pages A-9 and A-10). In the listing, normal face type corresponds to the actual contents of the data files and printer output, while the text in bold is explanatory annotation. The data file SET.PRM specifies the configuration of the back propagation model (4 input nodes, 3 output nodes, 8 hidden nodes), the error tolerance and maximum number of iterations used in determining the process input parameters A , and the number of random trials to be used in choosing the initial values of A . The data file TRAIN.W contains the 57 weights W of the PDPN simulation model.

In this example, begin with the paint mixture number 51, with a composition of 0.17 white, 0.50 black, 0.17 violet, 0.16 magenta. The object is to move from this operating condition to the one with the output color given in paint mixture number 52, by adjusting the concentrations of only the black and violet pigments. The file CONSTRAINTS.DAT shows that the first (white) and fourth (magenta) colorants have their concentrations fixed at the scaled values corresponding to concentrations of 0.17 and 0.16, respectively, while the

2011135

22

concentrations of the second and third colorants (black and violet) are permitted to vary over a wide range. The file OUTPUT.DAT specifies the goal values G for the output Y: a color with $L^* = 23.79$, $a^* = 23.52$, and $b^* = -28.38$.

5 The sample run gives the information which the PROGRAM SET prints to the screen. There are 4 input process parameters A, corresponding to the two fixed (white, magenta) and the two randomly selected (black, violet) initial concentrations. As the black and violet concentrations are varied, the output RMS
10 error E between the goal color G and modeled color components Y decreases monotonically to a small amount (0.03%). The final result is a model determination of the concentrations which should be used to produce the desired color. The program generated a color with concentrations of 0.161 for
15 black and 0.515 for violet, very close to the nominal concentrations of 0.17 for black and 0.50 for violet in the mixture actually produced.

20 This foregoing example also serves to illustrate the use of the invention in a batch initialization situation. The problem presented in a batch initialization situation is as follows: What should be the initial settings of the control parameters P (colorant concentrations) if it is desired to formulate a target color, such as that given by paint mixture number 52 ?

25 Similar to the aspect of the Example discussed above, the program generates one thousand (1000) sets of input concentrations at random (page A-2, lines 107 to 160). The set of input concentrations giving the color response closest to
30 the desired one was used for the input parameter settings. From this randomly generated starting point the program gave the results summarized in Table 3.

2011135

23

TABLE 3

sample	colorant	actual concentration	predicted concentration
5			
52	white	0.17	0.17
	black	0.17	0.16
	violet	0.50	0.51
	magenta	0.16	0.16
10			

It may be seen that the predicted concentrations are extremely close to those which actually produced the desired color.

15

EXAMPLE 2

An example of dynamical process control is the heat exchanger illustrated in Figure 7, which appears in problem 2.15 of Graham C. Goodman and Kwai Sang, Adaptive Filtering Prediction and Control, Prentice-Hall Englewood Cliffs, New Jersey (1984). The objective of this example is to provide dynamical control of the output temperature y of the heat exchanger by varying the voltage u of the heater, in the presence of known but uncontrolled changes in the input temperature T and the flow rate F .

For this example, both the volume V of the heat exchanger and the time constant RC were set to the value of one. For any time interval in which the temperature (T), flow rate (F), and voltage (u) are kept constant, the exact analytic solution to the differential equation shown in Figure 7 for the output temperature y is

$$y = y_0 \exp(-F*t) + (T + u*u/F)*[1 - \exp(-F*t)]$$

30

2011135

Here y_0 represents the output temperature at the beginning of the time interval, $t = 0$. Output temperatures using this exact solution will be called "Actual" results.

5 To generate a data set used for training the PDPN simulation model, input parameters F, T, and u were selected from the values given in this table:

	<u>Parameter</u>	<u>Values Used in Training</u>		
10	F	0.5	1.0	2.0
	T	0.0	1.0	2.0
	u	0.0	0.5	1.0

15 For each of the 27 possible combinations of the three parameters (F, T, and u), the actual solution was obtained for 9 consecutive time intervals separated by the constant time increment $dt = 0.1$. The next combination of input parameters (F, T, and u) was then selected, and the actual solution was continued. This resulted in a total of 243 consecutive samples
 20 of the exact solution to the differential equation describing the process shown in Figure 7.

For the data set used in training the PDPN, as guided by Figure 3, the following 8 inputs and one output were used:

	<u>Inputs</u>	<u>Output</u>
25	F(t)	y[t+dt]
	T(t)	
	u(t)	
	y(t)	
30	F[t-dt]	
	T[t-dt]	
	u[t-dt]	
	y[t-dt]	

2011135

The inputs and output were individually scaled by a linear transformation to place their minimum value in the training set at 0.1 and their maximum value in the training set at 0.9. A back propagation network with 4 hidden nodes was trained using these data to produce 41 connection weights embodying the PDPN simulation model of the dynamical process.

As an example of process control for this dynamical system, the input parameters $F = 1$ and $T = 0.5$ were set and fixed at nominal values. For a voltage $u = 0.5$, the steady-state output temperature $y = 1.0$ (giving 0.3142 in scaled units). From this starting point, the desired output temperature was changed to a value of about 1.3 (precisely 0.3772 in scaled units), and only the input voltage u was allowed to vary. The program SET was used with the trained PDPN simulation model to determine the value of the input voltage u for 20 successive time intervals, and the actual process output temperature y was updated using the analytic solution given above.

Figure 8 shows the smooth transition of the actual output temperature y (solid line) from the scaled value of 0.3142 to 0.3772, under control of the input voltage u specified by the method of this invention. The circles show the accuracy of the PDPN model prediction of the output temperature, computed one step at a time from the actual present and previous process inputs. The control values for the input voltage u produced by the PDPN are show in Figure 9. Beginning with the (scaled) steady-state value of 0.6656 for the voltage at the initial time, the voltage was immediately driven to its maximum allowed level (0.9) at the next time step. The voltage remained at the maximum level until the output temperature had almost reached its goal (at time interval 9), and then it quickly (about 6 time steps) was reduced to the scaled voltage (0.8127) required to maintain the scaled output temperature at the new value of 0.3772. This example demonstrates the efficient and

2011135

26

stable control strategy generated by the invention for this example problem in dynamical control.

Those skilled in the art, having the benefit of the teachings of the present invention as hereinabove set forth
5 may effect numerous modifications thereto. It should be understood that such modifications are to be construed as lying within the contemplation of the present invention as defined by the appended claims.

2011135

27

ED-0373

WHAT IS CLAIMED:

5

1. A control system for both batch initialization and for on-going control of a physical process, the process being of the type in which inputs having physical parameters P are used to produce a product having physical characteristics C associated therewith, the control system comprising:

10

a parallel distributed processing network previously trained to simulate the process, the parallel distributed processing network being responsive to input activations A to produce outputs Y , the activations being representative of the input parameters P of the process and the outputs being representative of the characteristics C of the product of the process;

15

means for producing a set of goal signals G corresponding to the desired physical characteristics D of the product of the process;

20

means for generating a signal representative of the error E between the goal signals G and the outputs Y of the network;

25

means for determining the values of the updated inputs A_U of the parallel distributed processing network that are needed to drive the error E to a predetermined minimum, and

30

means for changing the actual physical input parameters P of the process to correspond to the updated inputs values A_U producing the minimum error.

2011135

28

2. A method of both batch initialization and on-going control of a physical process using a parallel distributed processing network previously trained to simulate the process, the process being of the type in which inputs having
- 5 predetermined physical parameters P produce a product having corresponding physical characteristics C , the parallel distributed processing network having activations A and producing outputs Y , the method comprising the steps of:
- 10 (a) producing a set of goal signals G corresponding to the desired physical characteristics D of the product of the process
- (b) determining the error E between the goal signals G and the outputs Y of the network
- 15 (c) determining the values of the updated inputs A_U of the parallel distributed processing network that are needed to drive the error E to a predetermined minimum, and thereafter
- 20 (d) changing the actual physical parameters P of the process to correspond to the updated inputs values A_U producing the minimum error.

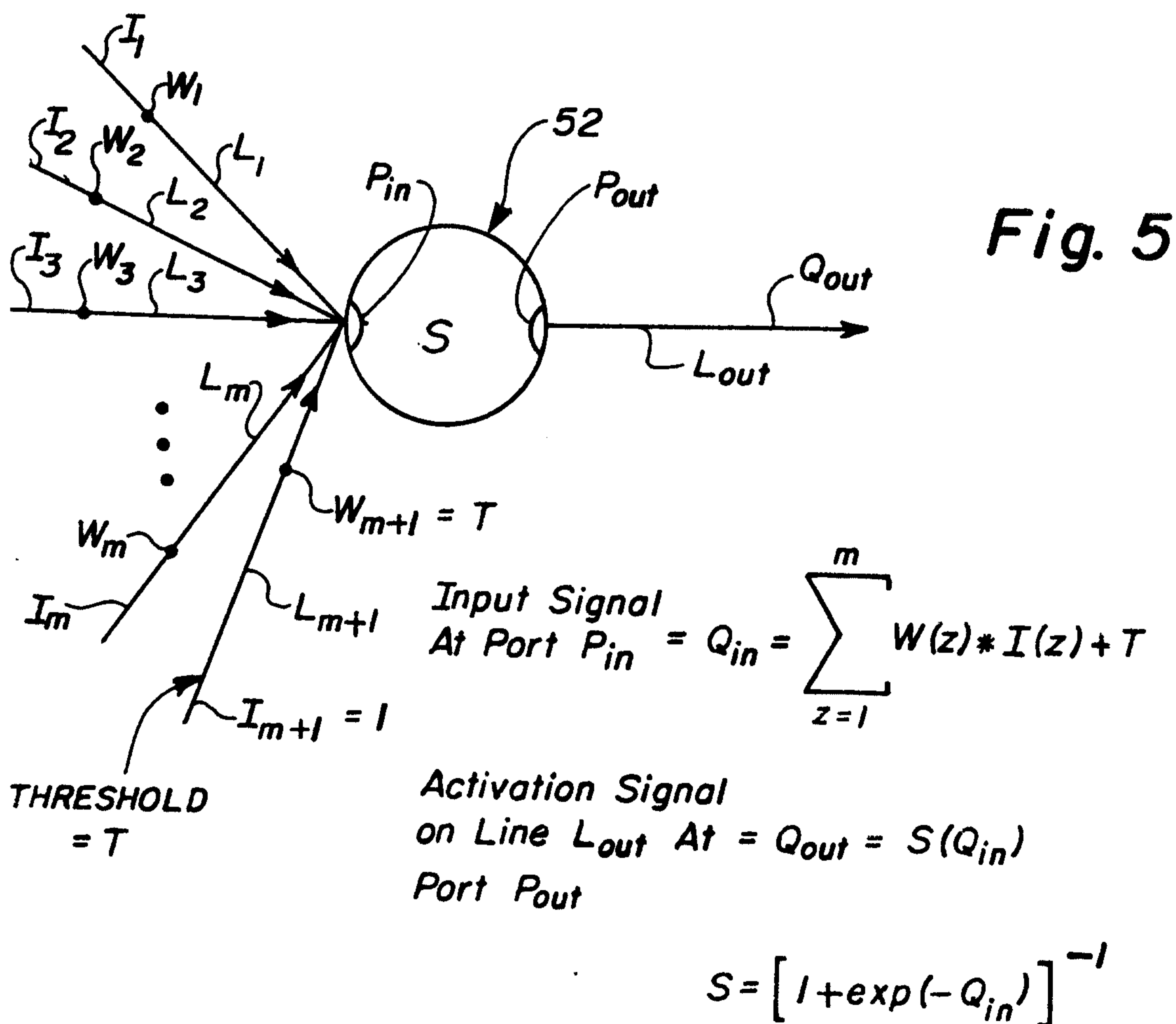
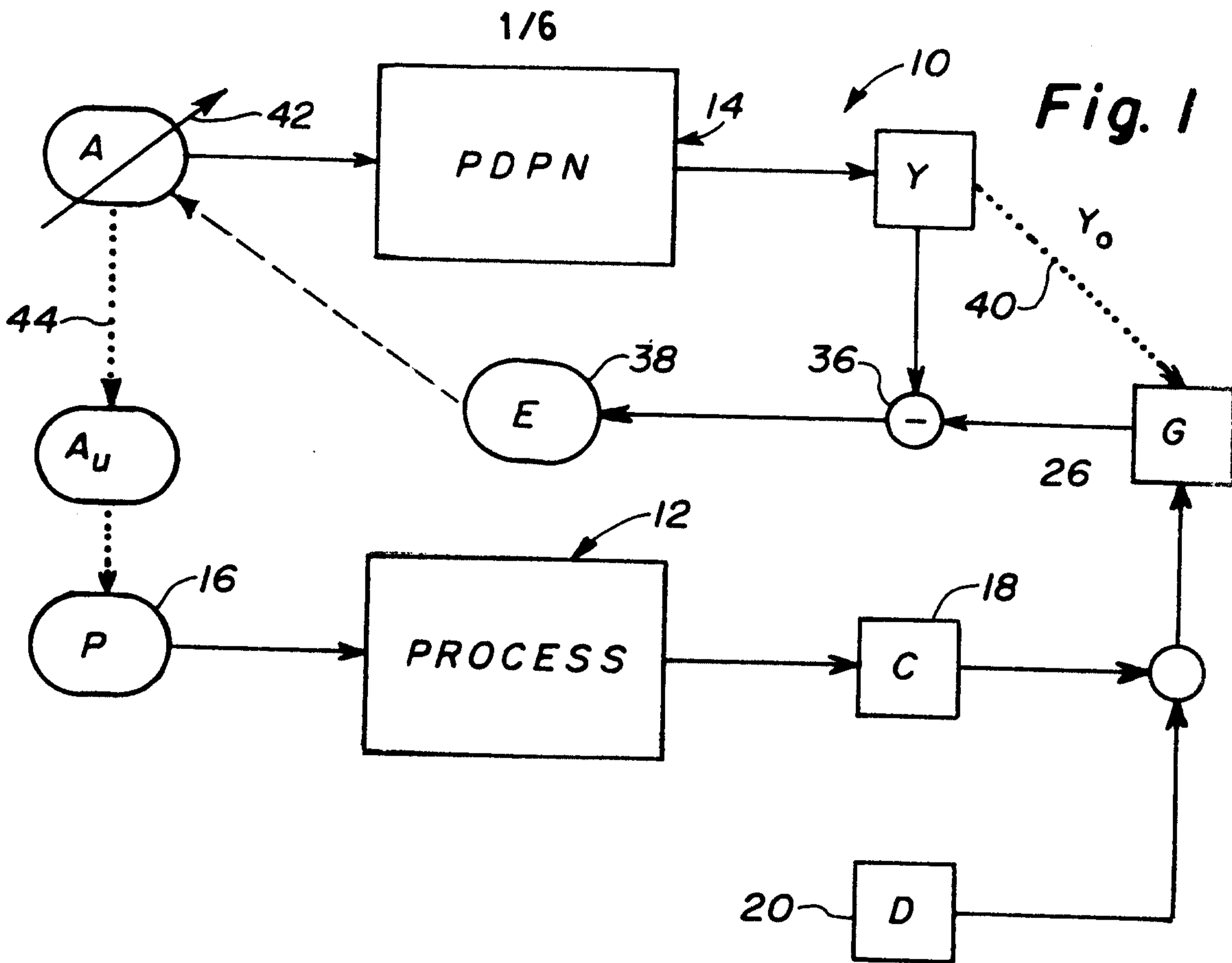
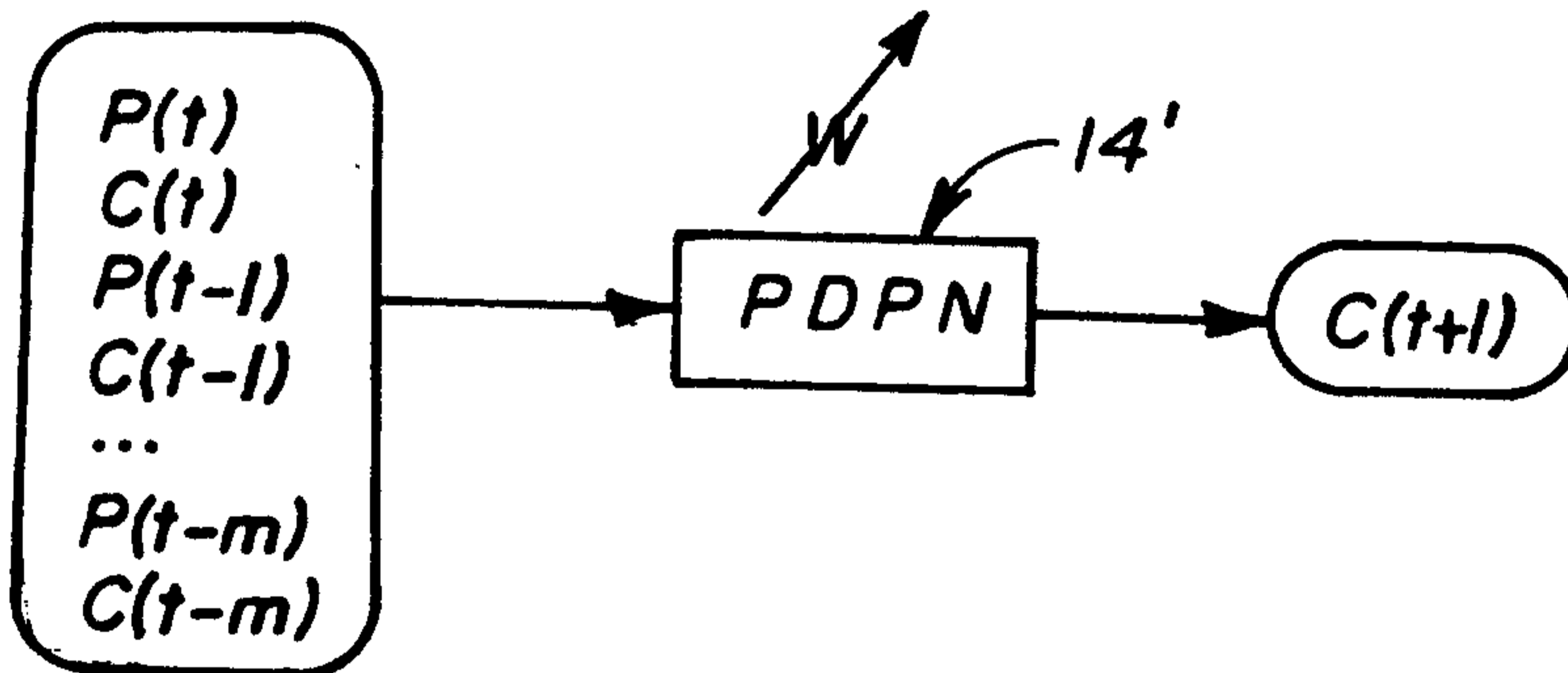


Fig. 3



TIME-DEPENDENT PROCESSES: LEARNING PHASE

Fig. 6

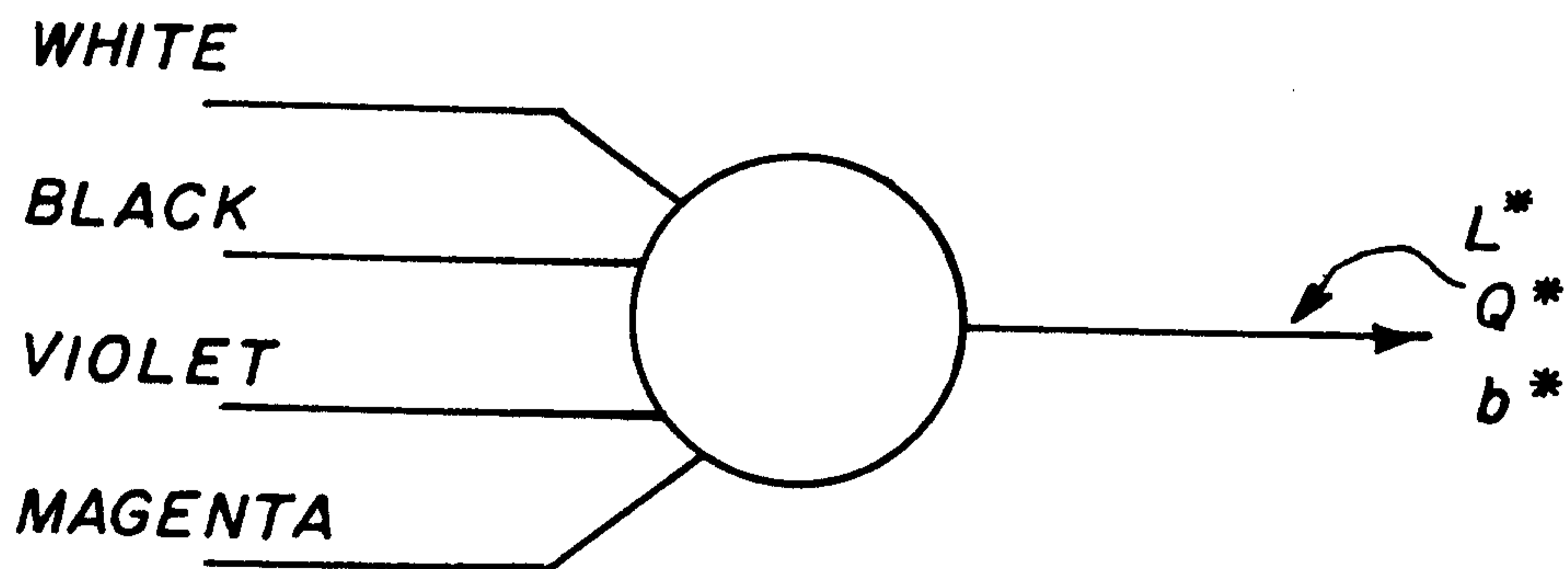


Fig. 4

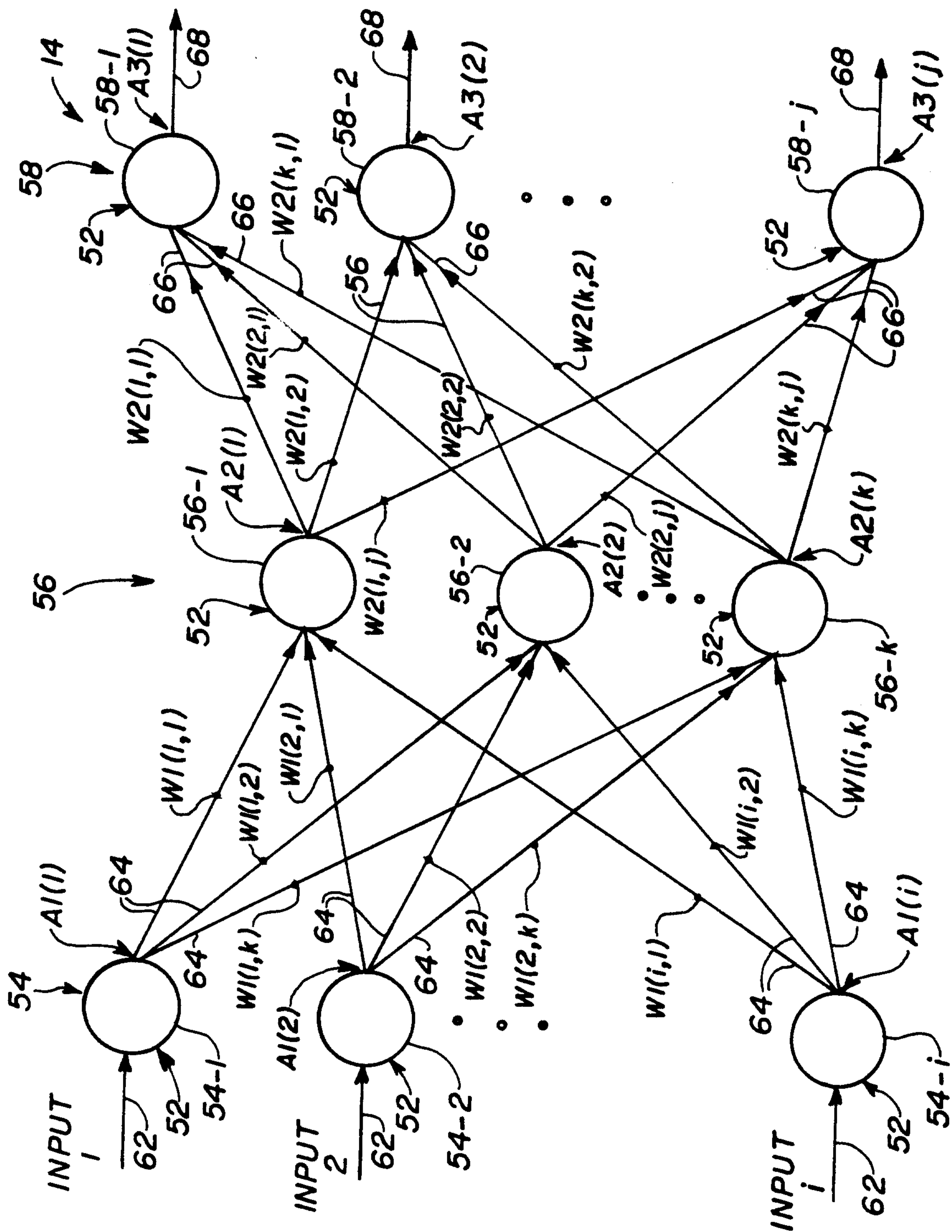
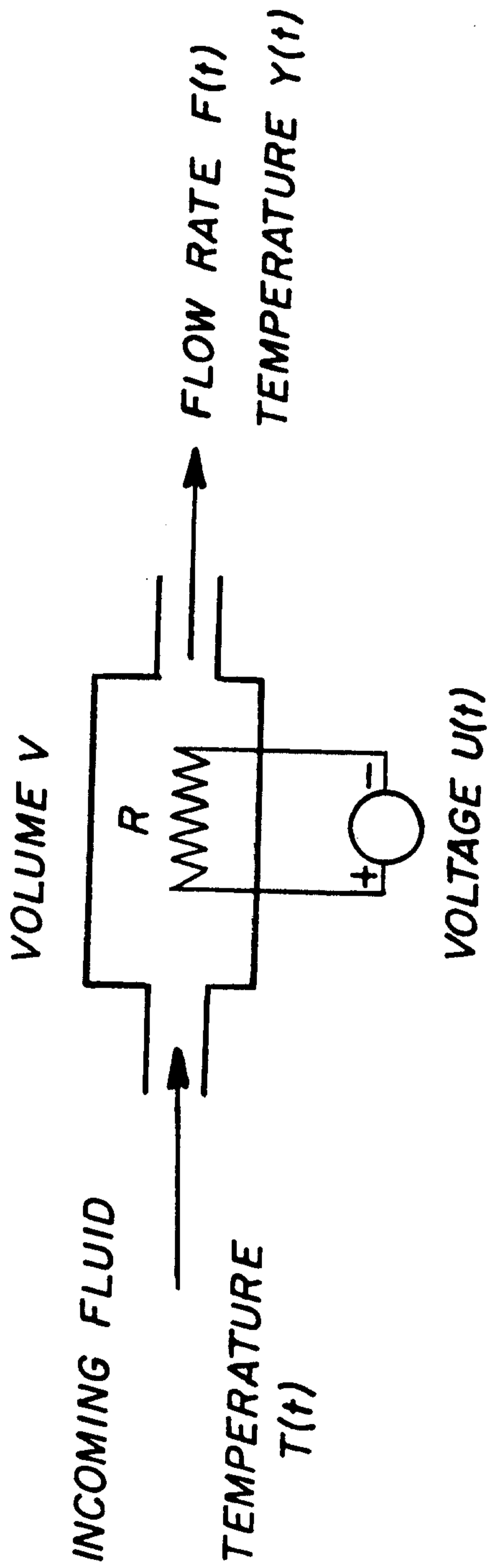


Fig. 7



5/6

$$\frac{dy}{dt} = -\frac{1}{V}F(t)y(t) + \frac{1}{V}F(t)T(t) + \frac{1}{RCV}u(t)^2$$

6/6

Fig. 8

HEAT EXCHANGER EXAMPLE

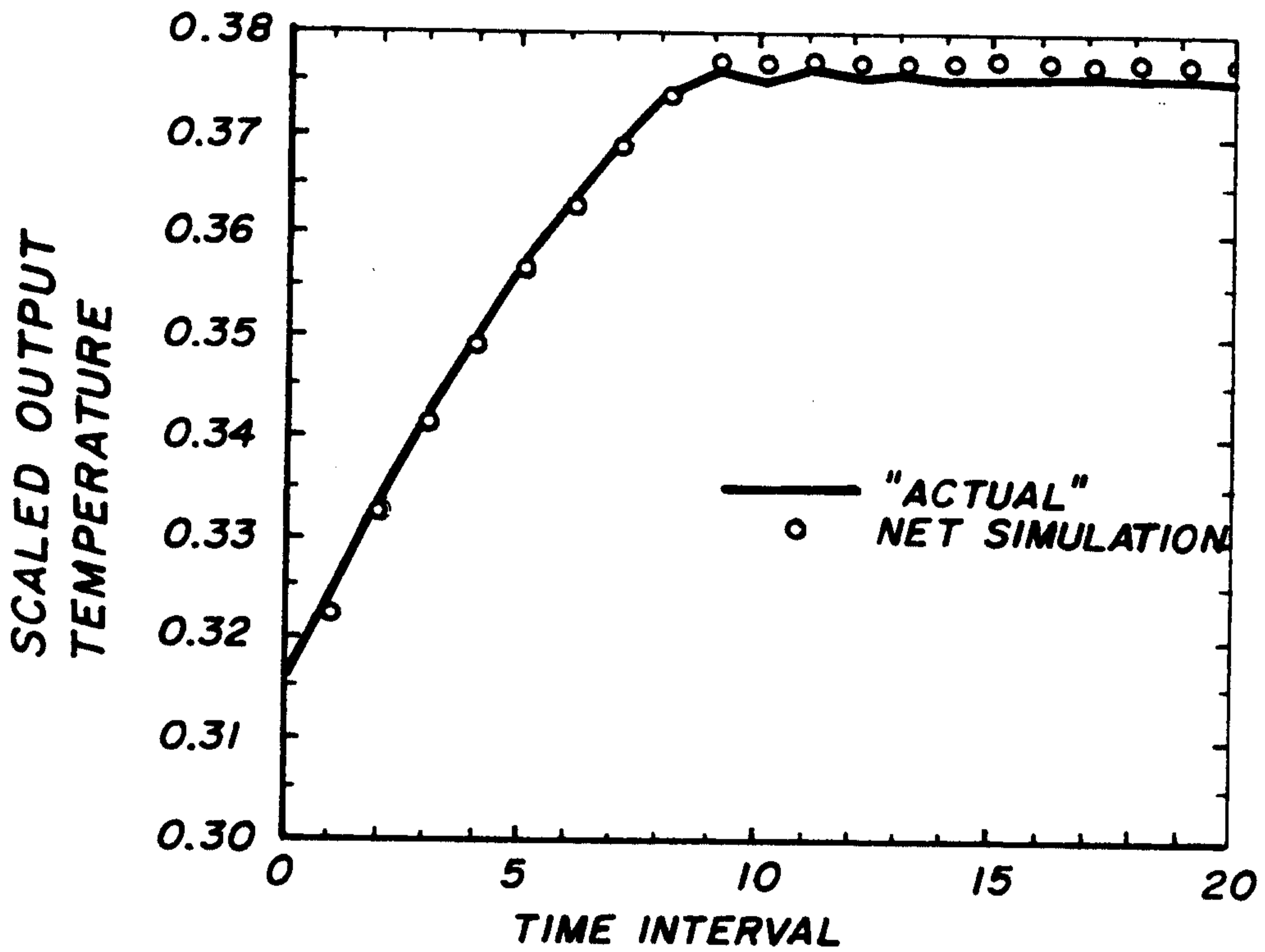


Fig. 9

HEAT EXCHANGER EXAMPLE

