PATENT SPECIFICATION 1,184,652

DRAWINGS ATTACHED.
Inventor:—BRIAN RONALD GAINES.


Index at acceptance:—G4 G(2B1, 2B2, 2B14, 2E3, 2F1, 2F2, 2P, 2X, 3A1, 3A2, 3A3B, 3A4, 3A5A, 3A10, 3AX, 3B4, 4A8, 4A10, 4A15, 4AX, 5); G1 N(IA3C1, 3V6P, 3S782, 4A); G3 R(7R, 9A, 9B, 9I, 40, X); G4 A(1D, 1U, 1C3, 1C7, 2D, 2H2, 2AY, 2BY, 2BX, 3W2, 4X, 8C, 10G, 10EX, 13E); G4 D1B3.

International Classification:—G 06 g 7/00; G 06 j 1/00.

COMPLETE SPECIFICATION.

Stochastic Computing Arrangement.

We, STANDARD TELEPHONES AND CABLES LIMITED, a British Company, of STC House, 190 Strand, London, W.C.2, England, do hereby declare the invention, for which we pray that a patent may be granted to us, and the method by which it is to be performed, to be particularly described in and by the following statement:

The present invention relates to a form of computer which has particular application to computation with continuously variable quantities, for example in process control.

According to the invention there is provided a computer including a digital computing element or elements, means for representing input information stochastically by the probability that a level in a clocked sequence of logic levels will be ON, means for applying the information so represented to the computing element or elements wherein computation is performed in a digital manner and means for converting the stochastically represented outputs of the computing element or elements to analogue or digital values.

The invention resides also in the method of representing quantities or events which is herein described.

The phase "computing arrangement" includes devices capable of executing arithmetical operations and also processors for coded signals representing, for example, audio or visual signals (e.g. special purpose speech and pattern recognizers) as well as devices which carry out tests and implement decisions based on numerical computations.

According to a particular aspect of the invention quantities or events are normalised so as to take a value in a standard interval (for example, in an electronic analog computer, the voltage range 0 to 1) and a value occurring in the calculation is presented by the sequence of logic levels or states in which the probability of a particular level being present or of a particular state occurring equals the normalised value.

According to further features of the invention the computing arrangement operates in a ternary logic scheme or in a threshold logic scheme.

In various preferred embodiments hereinafter described and illustrated, the invention provides an analog computer for the solution of differential equations, a non-linear stochastic converter, a control for a distillation column, a random delay, a Bayesian predictor and estimator, a computer using 'steepest descent' methods, a model-reference adaptive controller, a bang-bang or optimal relay control device, a flexible patch-board, an adaptive filter for the detection of signals in noise, an adaptive filter for the detection of randomly phased repetitive signals, a data smoothing and stock control processor, a Laplacian device for solving Laplace's equation with pre-
scribed boundary conditions, and an arrangement of such devices.

The invention can be realised using electronic circuit elements, both analog and digital, which may be for example, conventional macro-circuits, or integrated circuits or other forms of micro-circuit. It can also be realised using pneumatic or hydraulic elements, chemical, mechanical, thermal, radioactive or optical components or other devices which are capable of realising numerical or logical computation or of realising a sequence of states or levels.

The various embodiments of the invention hereinafter described utilise components which for the most part operate in real time, but the general stochastic computing arrangement however utilises elements which take into account behaviour of sequences in past time. If a memory element is used which differs from the usual store elements (e.g. ferrite or magnetic core, delay line, circulating register, charge on a capacitor) and for example from our adaptive element described later with reference to Fig. 1 (e) and Fig. 15, then the circuitry to read out the stored value in a stochastic manner, and the method of calculating with the value which is read out, and the method of representation of the said value, all fall within the scope of our invention.

There is an ineluctable error factor associated with the measurement over only a finite time of the statistical properties of a sequence.

The sequences of states or levels therefore represent the values of quantities to only a finite degree of precision. The accuracy of a measurement can in general be increased by averaging over a larger sample. The theorems of statistical decision theory, as developed by Pearson, Neymann and Wald, provide methods for expressing the degree of accuracy to which a quantity is known, subject to an assigned acceptable probable error, the uncertainty being capable under suitable conditions of being stated as an explicit relationship.

Estimates of and inferences about the values of quantities are then made on the basis of the values observed, the bounds of the error being known in closed form. Configurations of elements operating according to the invention can be used to perform several calculations simultaneously.

A computer wherein the results of calculations in one section are used to control the time-scale of another section may be realised according to the invention.

The invention further resides in the arrangement in which a digital computer provides the clock pulses which control the times of operations in the stochastic computer which is herein described.

The invention also resides in a computing arrangement of the kind described wherein peripheral equipment which includes analog interfaces or digital interfaces is provided, and equally where the computing arrangement is used with no input from the external world (e.g. as an analog computer for the solution of differential equations).

The above mentioned and other features of the invention will become more apparent and the invention itself will best be understood by reference to the following description of various embodiments of the invention taken in conjunction with the drawings accompanying the Complete Specification, in which:

- Fig. 1 is a list of the symbols used;
- Fig. 2 illustrates a ternary logic element together with a table of its values;
- Fig. 3 is a so-called cross-coupled flip-flop (CCFF);
- Fig. 4 is an arrangement which utilises a CCFF and which functions variously as a multiplier or adder;
- Fig. 5 is a multiplier for the affine symmetric representation;
- Fig. 6 illustrates a multiplier for the so-called affine asymmetric representation;
- Fig. 7 is an isolator;
- Fig. 8 is a simple squarer;
- Fig. 9 is a two-input adder for the affine representations;
- Fig. 10 is an analog-to-stochastic converter;
- Fig. 11 is a digital-to-stochastic converter;
- Fig. 12 is a two-input stochastic integrator;
- Fig. 13 is an integrator used with one input;
- Fig. 14 is an integrator used as an adaptive element;
- Fig. 15 is a multiplier having analog inputs and an analog output;
- Fig. 16 is a squarer for analog inputs and has an analog output;
- Fig. 17 is an adder having analog inputs and an analog output;
- Fig. 18 is an adder which utilises a CCFF;
- Fig. 19 is a cross-correlator in which quantities are represented in the affine symmetric representation;
- Fig. 20 is a stochastic computer for the solution of differential equations;
- Fig. 21 is a random delay element;
- Fig. 22 is a non-linear stochastic converter;
- Fig. 23 is a four-state switching function generator;
- Fig. 24 is a stochastic divider;
- Fig. 25 is a stochastic square root solver;
- Fig. 26 is a stochastic subtractor;
- Fig. 27 is a stochastic divider.
Fig. 28 is a stochastic averager and divider.
Fig. 29 illustrates a stochastic computer attached to the control of a distillation column.
Fig. 30 is the estimating section and Fig. 31 the predicting section of a Bayesian predictor and estimator.
Fig. 32 is a computer which uses 'steepest descent' methods.
Fig. 33 is a steepest descent computer used as a divider.
Fig. 34 is a 'bang-bang' or optimal relay control device.
Fig. 35 is an adaptive filter for the detection of randomly phase repetitive signals.
Fig. 36 illustrates the use of adaptive elements as a data smoothing system.
Fig. 37 is a device for solving Laplace's equation with prescribed boundary conditions.
Fig. 38 is an arrangement of the devices of Fig. 37 whereby a solution of Laplace's equation is obtained which is valid over an area and which satisfies prescribed boundary conditions.

In Fig. 1 (a) is an AND gate and will be defined both for ternary logic schemes and for binary logic. In ternary logic three states are given, ON, OFF and OPEN, which in one convention correspond to minus, plus and zero volts respectively. An AND gate has several inputs and a single output, the value of the output being ON if and only if all the inputs have value ON, being OPEN if at least one input is OPEN, and OFF otherwise. In binary logic the operation of the AND gate is the two-valued restriction of the ternary scheme i.e. the output is ON if and only if all the inputs are ON, and is OFF otherwise.

A particular ternary logic element is described later with reference to Fig. 2.

Fig. 1 (b) is an OR gate, that is an element which has several inputs and a single output, the value of which is ON if at least one of the inputs is ON, is OPEN if all the inputs are OPEN, and is OFF otherwise.

(c) is an inverter, that is a single input element whose output is ON if the input is OFF, OFF if the Input is ON, and OPEN if the input is OPEN.

(d) is an isolator and is described later in more detail with reference to Fig. 7.

(e) is a representation as a block-box of a so-called JK flip-flop, and

(f) is a more complete representation of a so-called JK flip-flop.

(g) is a table relating the values of the inputs and outputs of a JK flip-flop in clocked operation.

A JK flip-flop in clocked operation is a device which has two stable states, Q and \( \overline{Q} \). (The bar denotes complementation, i.e. inversion).

Throughout the following paragraph, ON corresponds to 1 and OFF to 0.

A clock pulse is necessary to activate the device, and when it is applied, if there is no signal at J and at K, no change is made in the condition. An OFF at J and an ON at K causes the device to assume state \( Q = 0, \overline{Q} = 1 \), which values are output, an ON at J and OFF at K causes the device to assume state \( Q = 1, \overline{Q} = 0 \), which values are output. Application of a '1' at both inputs causes the state of the device to reverse and gives the output of the complements of the contents before the application of the signals.

As a convention, where an input is shown grounded, it is assumed that it is ON.

In the following diagrams the orientation on the page of for example the flip-flop box is immaterial, it being the number and sense of the inputs that determines the output values.

(h) is an analog-stochastic converter and is described in detail later with reference to Fig. 10,

(i) is a digital-stochastic converter for some representation (where the form of representation is important, this is noted) and is described in detail later with reference to Fig. 11;

(j) is a Schmidt trigger occurring in Figs. 10 and 11, that is an element whose output is ON for each period of time that an input in excess of a certain value is present and whose output is OFF otherwise;

(l) is an integrator having facilitites to read out its count and to set an initial count. It is described in more detail later with reference to Fig. 12;

(m) depicts the box which forms the abbreviation of, or symbol for, 'multiplier'. The constituents of various multipliers are listed in detail with reference to Figs. 4, 5, 6 and 15 of the following drawings;

(n) is an N-input adder, and is a straightforward generalisation of the 2-input adder (k);

(o) is a so-called cross-coupled flip-flop and is described later in more detail with reference to Fig. 3;

(p) is a table which relates the value of the output to the two inputs for symmetric multipliers (these include Fig. 6). The inputs may take any of the values ON, OFF, OPEN, i.e. the logic scheme is ternary

(q) is the restriction of table (p) to the case where the inputs can only be either...
ON or OFF i.e. the restriction of ternary inputs to binary. As the outputs are either ON or OFF i.e. 'well formed' in the binary scheme, and hence meaningful, the restriction is consistent.

Similarly, the restriction of the inverter of Fig. 1 (c) to binary operation is the natural one.

(1) is the standard symbol for a capacitor (these occur for example in Figs. 9, 10, and 11);

(s) is the abbreviation for the switching function generator which is described later in more detail with reference to Fig. 23;

(t) is a Laplacian computing element and is described later in more detail with reference to Fig. 37; and

(u) is a comparator in a servo control, in which an error signal is obtained by subtracting an achieved value from a theoretical value.

Referring now to Fig. 2, the arrangement shown in (a) is a particular ternary logic element.

An NPN transistor and a complementary PNP transistor are connected as shown, the values of all the resistances being equal; if ON is represented by a negative voltage, OPEN by no volts, and OFF by a positive voltage, then application of inputs to A and B yields the output given in table (b) of the figure.

Ternary logic is of particular importance in artificial intelligence, simulation etc. because the OPEN condition may correspond to, for example, the instruction 'go into random search mode' in a learning machine where insufficient information has been supplied for a YES or NO decision to be made.

In the stochastic computer the use of ternary or high-valued logics (having three or more levels) enables analog data to be quantized with more levels and hence to be more accurately represented at any instant.

British Patent Specification 1,099,574 describes an information processing arrangement which includes an adaptive element containing a store the value of which is continually randomly modified and a counter the value of which is repeatedly compared with the instantaneous value of the randomly modified store, being then appropriately modified according to the results of the comparison, the adaptive element having associated therewith a limit store with adjustable setting the maximum setting of which does not exceed the greatest value expressible by the counter, the value of the randomly modified store being maintained below the value of the limit store.

Other logic schemes which might be utilized in stochastic computing are threshold logic schemes with more than three threshold values. Elements using such schemes are important in pattern-recognition and classification schemes where majority decision elements, quorum units, and questions of separability arise.

Digital techniques can be used to handle and process analog data more directly if the analog quantities are represented not as voltage levels nor as n-tuples of binary digits, but as the probability that a particular binary or multi-level event will occur (or more generally as the probability that one configuration representing one of several possible events will occur).

In the conventional analog computer quantities are represented by voltage in a given range from zero up (asymmetric or single-quadrant) or centred about zero (symmetric or four-quadrant), and all quantities must be scaled to be represented within the range of voltages of the computer.

The computing arrangement wherein quantities, e.g. analog variables, or events such as the occurrence of a particular symbol (e.g. on the retina of a pattern recognition device) are represented by statistical properties of states or of logic levels may be referred to as a stochastic computer. Thus in a stochastic computer a quantity for example may be scaled so that the scaled value can be regarded as a probability whose value is not less than zero or greater than one. Scaling for the stochastic computer allows a greater variety of representations than in the conventional computer since non-numeric quantities or events may be represented and no restriction need necessarily be placed upon the range of values of numeric quantities. Each event or quantity is represented in the computer by a sequence of logic levels or states of the inputs and outputs of the elements of the computer, and each sequence of logic levels or states at the input or output of an element of the computer represents an event or quantity. This representation is stochastic in that statistical properties of a sequence rather than the sequence itself are used to define the event or quantity, and hence only the probability that a given sequence represents a given event or quantity is generally known, and conversely only the statistical properties of a sequence are generally constrained by the event or quantity it represents.

The representation e.g. at one input to the computer may differ from that used elsewhere e.g. at an output, and hence the computation performed by a given type of element may vary according to the alternative representations (and hence the interpretations) of its inputs and outputs. Thus
particular representations will be preferred for particular types of calculation.

Corresponding to the symmetric representation of bounded quantities on the conventional analog computer by voltages in the range $-E$ to $+E$, there is the affine symmetric binary representation in which a quantity previously represented by a voltage $V$, is represented by a probability that a two-state device will be in one of its states (here called "on"), this probability being defined by:

$$ p(\text{on}) = \frac{1}{2} \left( 1 + \frac{V}{E} \right) \quad (1) $$

In this representation the maximum voltage $E$ is represented by a certainty that the binary event will be on, and the minimum voltage $-E$, by a certainty that it will not be on (i.e. that it will be 'off'). Zero voltage is represented by the device randomly fluctuating between its on and off states with equal chance (or relative frequency over a period) of being in either. Thus in the affine symmetric binary representation a binary device which over a period takes the state "on" with relative frequency $p_\text{s}$ is taken to represent a corresponding voltage $V$, where

$$ V = (2p_\text{s} - 1)E \quad (2) $$

or, of course, any quantity of the magnitude $V$.

Corresponding to the same representation there is the affine symmetric ternary representation in which the inputs and outputs of stochastic computing elements have three states, here called 'on', 'off' and 'open'. A quantity $V$ in the range $-E$ to $+E$ then does not necessarily have a unique representation and there are a range of statistical properties generating sequences of ternary levels which may represent it. The preferred representation is:

for $V$ greater than or equal to zero,

$$ p(\text{on}) = \frac{V}{E} \quad (3) $$

$$ p(\text{off}) = 0 \quad (4) $$

$$ p(\text{open}) = 1 - \frac{V}{E} \quad (5) $$

(where on, off and open are, of course, mutually exclusive so that only one of them may occur at a time),

for $V$ less than zero,

$$ p(\text{on}) = 0 \quad (6) $$

$$ p(\text{off}) = -\frac{V}{E} \quad (7) $$

$$ p(\text{open}) = 1 + \frac{V}{E} \quad (8) $$

In this representation the maximum and minimum quantities are represented by the level being always on or always off respectively, and the zero level may be represented by a random fluctuation between these levels as before but is preferably represented by the open condition in an element. A sequence of levels of a ternary device with relative frequencies of on and off, $p(\text{on})$ and $p(\text{off})$ respectively will be taken to represent a corresponding quantity, $V$, where:

$$ V = \frac{p(\text{on}) - p(\text{off})}{(1 + aV)} \quad (9) $$

To the asymmetric or single-quadrant bounded representation of quantities by voltages from 0 to $E$, corresponds the affine asymmetric binary representation:

$$ p(\text{on}) = \frac{V}{E} \quad (10) $$

Both in the symmetric and asymmetric representations multi-valued logic of ternary or higher-order may be used. It will be called affine or linear if a sequence of states or logic levels is taken to represent a quantity which is a weighted sum of the relative frequencies of the states or logic levels in the sequence. Multi-level logic is generally capable of greater accuracy and speed than binary or ternary logic but requires circuit elements having greater complexity and hence generally greater cost and less reliability.

The affine representations hitherto described are unsuitable for the representation of unbounded quantities and hence are difficult to use in situations where these may arise, such as division of one quantity by another. In the asymmetric projective binary representation a quantity $V$ greater than or equal to zero is represented by the probability that a two-state device or level will be in one of its states, this probability being defined by:

$$ p(\text{on}) = \frac{V}{(1 + aV)} \quad (11) $$

where $A$ is a scaling factor greater than zero. In this representation zero quantity is represented by certainty that the binary event will be off and infinite quantity by the certainty that it will be on. A binary device which over a period takes the state "on" with relative frequency $p(\text{on})$ is taken to represent a corresponding quantity, $V$, where:

$$ V = \frac{p(\text{on})}{a (1 - p(\text{on}))} \quad (12) $$

This representation is readily extended to quantities which are wholly negative by
multiplying these quantities by $-1$. In the symmetric projective ternary representation as in the previous ternary representation there are a range of statistical properties generating sequences to represent a given level.

The preferred representation is:

for $V$ greater than or equal to zero,

$$p(\text{on}) = \frac{aV}{1 + aV} \quad (13)$$

$$p(\text{off}) = 0 \quad (14)$$

$$p(\text{open}) = \frac{1}{1 + aV} \quad (15)$$

for less than zero,

$$p(\text{on}) = 0 \quad (16)$$

$$p(\text{off}) = \frac{-aV}{1 - aV} \quad (17)$$

$$p(\text{open}) = \frac{1}{1 - aV} \quad (18)$$

In this representation a sequence of states of a ternary device with relative frequencies of on and off $p(\text{on})$ and $p(\text{off})$ respectively will be taken to represent a corresponding quantity $V$, where:

$$V = \frac{p(\text{on})}{a(1-p(\text{on}))} - \frac{p(\text{off})}{a(1-p(\text{off}))} \quad (19)$$

Some properties of stochastic sequences of states or levels should be mentioned. A sequence is said to be almost stationary if its statistics do not vary appreciably over a length of the sequence i.e. if the relative frequencies of events, pairs of successive events etc. remain the same over a length of the sequence. A statistic of a sequence from a set of sequences is said to be an unbiased estimator of a quantity if the mean value of the statistic over the set of sequences tends to that quantity as the set of sequences becomes larger. A sequence is said to be Markovian of zero order or not autocorrelated if the probability of occurrence of a state or level in it is not affected by the events in the sequence previous to that event, i.e. if knowledge of the past history of the sequence does not aid in predicting its future events course; otherwise the sequence is said to be auto-correlated. A sequence is said to be Markovian of order $N$ or autocorrelated to depth of $N$ if the probability of occurrence of a state or level in it depends on the previous $N$ states or levels but is independent of those preceding these. One sequence is said to be cross-correlated with another if the probability of occurrence of a state or level in it depends on the concurrent state or level of the other sequence. A set of sequences is said to be pseudo-random if the relative frequency of occurrence of a level or state in one is independent of the concurrent levels and states in the others. A sequence is said to be pseudo-random to depth $N$ if it is possible to form a set of $N$ sequences by delaying it through zero, one, two etc. events, which set is itself pseudo-random.

In stochastic logic it is preferred to deal with sequences which are not autocorrelated and are unbiased estimators of the quantities they represent. Also when two sequences are made to interact it is preferred that they are not cross-correlated. However for some computations use of such sequences is unavoidable and in others may be desirable, since the conditional probabilities inducing autocorrelating may themselves be used to represent quantities or events, and similarly for reasons of simplicity or economy it may be desirable to use pseudo-random rather than truly stochastic sequences.

For accuracy, the sequences used should be almost stationary, and hence the speed or dynamic range of the processing of these sequences should be much greater than the dynamic range of the quantities they represent. This restricts the dynamic range of the computer.

Parameters whose state or level may be statistically constrained to generate stochastic or pseudo-random sequences are already to be found in many mechanical, hydraulic, pneumatic, thermal, electrical, electronic and radioactive devices or systems involved in process regulation and control, computation, and the system of stochastic representation of quantities herein described may be applied to them directly so that relatively simple elements within these systems may be used in relatively complex calculations, and in particular numerical or quantitative calculations may be carried out by digital or logical elements. The preferred realisations of the computing elements of a stochastic computer are, however, in general electronic, since electronic digital components are available in small size with high speed and high reliability. The mode of operation and logic diagrams herein described might equally apply to other components, in particular to pneumatic logical elements, and these are in no way excluded.

Parameters whose states or levels in their relative frequency of occurrence may be used to represent events or quantities include for example the intensity, frequency, mark-space ratio and pulse width or other pulse-coded parameters, carried in single or multiple lines of electrical, pneumatic or other signals. We distinguish between synchronous or asynchronous forms of computation: in synchronous computation the states or levels of all devices and their inputs and outputs within the computation may change only at given instants of time.
often defined by the occurrence of a clock-
pulse common to all computing elements; in
asynchronous computation the states and
levels of some or all of the devices may
change at instants not common to all the
devices and often determined by the prop-
erties of the devices themselves. Since the
computations effected by a device depend
on the instants at which it may change state
it is necessary for the sequence of instants
at each device to be well defined at least in
its statistical properties. In full synchro-
nous logic this is readily achieved since
a single clock pulse may be used to ac-
tivate the whole computer, but in asyn-
chronous logic this must be done separately for
each asynchronous group of elements within
the computer. Because of this asyn-
chronous logic is often used for computa-
tions which may be done uniformly in time
so that individual elements may run at
their natural rate and synchronous logic is
often used when the time scale of the whole
or part of a computer is to be under con-
trol. In many computations a mixture of
asynchronous and synchronous elements may
be used and in particular one part
of the computer may control the clock
pulses and hence the time scale of another.

In the following description synchronous
computing elements will be described in
detail since these are generally the more
complex. The modifications or omissions
necessary for their asynchronous use are
not given in detail.

The basic elements of a synchronous
logic system are boxes with input and out-
put lines plus a clock pulse line. The in-
put lines are connected to the output lines
of other boxes and have logic levels ap-
plied to them. The output lines are at
logic levels according to the state of the
box. When a clock pulse is applied the
outputs and state take new values depend-
ent upon the inputs and the previous state.

The logical inverter (c) of Fig. 1 gives
the output on when its input is off, the
output off when its input is on and the out-
put open when its input is open. Thus
when used as a stochastic computing ele-
ment it implements the following computa-
tions:—with the affine asymmetric bin-
ary representation:

\[ V' = E - V \] (20)

With the affine symmetric binary and
ternary, projective symmetric ternary, hy-
perbolic binary and trigonometric binary it
implements additive inversion:

\[ V' = -V \] (21)

With the projective binary representation
it implements multiplicative inversion:

\[ V' = 1/V \] (22)

The term 'independent event' is used in
statistics to mean an event whose proba-
bility of occurrence is not affected by the
probability of occurrence of other events.
A probability is a function whose value
lies in the interval zero to one, end points
included.

Consider two independent events with
probabilities \( p_1 \) and \( p_2 \).

The probability of their joint occurrence
is \( p_1 \) times \( p_2 \). If \( p_1 \) and \( p_2 \) represent
the normalised values (in the interval zero to
one) of analog quantities \( V_1 \) and \( V_2 \),
then

\[ \text{times } p_1 \text{ (also in the interval zero to one)} \]

represents the value of the product \( V_1 \)
times \( V_2 \). The output of the binary AND
gate illustrated in Fig. 1(a) is one when both
inputs equal one and zero for any other
combination of inputs. Hence if the proba-
bilities of levels applied to the two inputs
are such that the probabilities that the in-
puts are ON are \( p_1 \) and \( p_2 \) respectively,
then the probability that the output is ON
is \( p_1 \) times \( p_2 \), representing a value \( V_1 \)
times \( V_2 \).

Thus in the affine asymmetric binary re-
presentation multiplication in the stochastic
computer is carried out by a single AND
gate.

The product of three or more quantities
in this representation is obtained as the out-
put of a three or more inputs AND
gate.

In what follows, no confusion will result
if the sign \( + \) is used for the switching con-
nective OR and juxtaposition is used for
AND. Where arithmetical addition and
multiplication are intended, this will be clear from the context. A bar over a
quantity denotes its boolean inverse.

\[ \overline{a} = 0 \text{ if and only if } a = 1, \]

\[ \overline{a} = 1 \text{ if and only if } a = 0. \]

The circuitry for multiplication in the
affine symmetric binary representation is
shown in Fig. 6. If levels \( a \) and \( b \) are ap-
plied as shown, the input to AND gate 1
is \( a \) \( b \) and to AND gate 2 is \( a \) \( b \). The 110
output of OR Gate 3 is then \( a \) \( b \) + \( a \) \( b \),
which is the representation of the scaled
product of the quantities represented by
the sequences containing \( a \) and \( b \).

This multiplier is an 'equality' gate which 115
gives an ON output if and only if its in-
puts are the same.

To multiply three or more quantities in
the affine symmetric binary representation
two or more of the multipliers (equality gates) described may be used in cascade.

The multiplier in the affine symmetric ternary representation which is identical in logical form to that of Fig. 6, will best be understood in conjunction with the truth table (p) of Fig. 1. The output line is on when both inputs are on or both off; it is off if one input is on and the other is off; it is open if either input is open. Care must be taken in the definition of the logical elements used to show the realisation of the multiplier. The AND gate acts as before giving the output on if both its inputs are on, off if one is on and the other off, or if both are off, and the output open if either of its inputs are open, the inverter gives the output on if its input is off, the output off if its input is on and open if its input is open. The OR gate gives the output on if any of its inputs are on, off if none of its inputs are on but one or more are off and open if all its inputs are open.

The multiplier for the projective binary representation will be best understood after consideration of the cross-coupled flip-flop of Fig. 3. At a clock pulse the state of the flip-flop may be changed as already described to a state which depends upon its preceding state and the values of the inputs X and Y. The output Z is equal to the new input X if the flip-flop has output Q on and equals the complemented new input Y if the Q output is ON.

This device realises the transformation

\[ p(Z) = p(X) \] (23)

\[ p(Z) = p(Y) \]

and consists essentially of a flip-flop having an upper input X and a lower input Y.

Y passes also through an inverter and is gated at an AND gate with the output Q. X is gated with Q at another AND gate and the outputs of the two AND gates are combined at an OR gate to give the output Z.

Logical circuity connected to its inputs can turn this device into an adder, multiplier or divider in the appropriate representation. It is possible to use the output of the flip-flop as a direct output of the device and hence remove the two AND gates and the one OR gate, however the output then auto-correlated to depth one which may cause difficulty in later calculation.

Fig. 4 shows the use of a cross-coupled flip-flop CCF as a multiplier for the projective binary representation. Considering equation (23), if we replace X by AB and Y by AB then

\[ \frac{P(Z)}{P(Z)} = \frac{P(AB)}{P(AB)} \]

Therefore

\[ \frac{P(Z)}{P(Z)} = \frac{P(A)P(B)}{P(A)P(B)} \]

or

\[ \frac{P(Z)}{P(Z)} = \frac{P(A)}{P(A)} \times \frac{P(B)}{P(B)} \] (24)

Since the logical inverter gives multiplicative inversion in this representation, the multiplier and inverter may be used together as a divider. The box of Fig. 1(m) can be used to represent the multipliers of Figs. 4—6 and 15.

It will be noted that this is the first example of the use of an element requiring clock pulses for its operation; all previous devices described for inversion and multiplication would work equally well in asynchronous logic. If the CCF is to be used in asynchronous logic then the clock pulses may be obtained from a local oscillator; or if the two inputs are mutually exclusive so that they cannot be on together then the occurrence of a change at the output may be taken to trigger a clock pulse. A combination of these two techniques may also be used in which the flip-flop is arranged to act as its own local oscillator.

The multiplier circuits described rely on lack of cross-correlation between the input sequences and hence if one is to be used as a squarer or several are to be used to raise a quantity to a higher power then it is insufficient to common the input lines since the inputs will then be identical and hence cross-correlated — in practice the quantities represented by input and output would then be the same. However provided the input sequence is not auto-correlated an independent replication of it may be obtained by delaying it through one clock pulse. Thus a flip-flop used as a delay acts as an isolator, as shown for example in Fig. 7. Fig. 8 illustrates squarer circuity in terms of a multiplier and an isolator.

The use of a delay as described above is important and whenever a signal takes mul-
Multiple paths in a stochastic computer statistical independence should be assured using single or multiple delays as isolators if necessary.

Autocorrelated sequences when they arise may be reduced to non-autocorrelated sequences by the introduction of random delays whose maximum depth is greater than or equal to the autocorrelated depth of the sequences. A circuit for doing this with autocorrelation depths up to four is shown in Fig. 21. Signals from random noise sources cause the triggers to change states at random intervals and hence to pass a pulse through the capacitors to the clock pulse lines of the flip-flops FF1 and FF2 whenever the noise increases above a certain level. A single noise source may be used in common to the triggers provided their triggering levels are not the same. At a clock pulse flip-flops one and two are in random states which are transferred to flip-flops FF3 and FF4. The mean rate of random pulses from the triggers should preferably be 10 or more times the clock pulse rate. Flip-flops FF5, FF6 and FF7 are connected as a shift register to the input line and hold the previous states of the input line at unit delay, two delay and three delay, respectively. Flip-flops FF3 and FF4 gate the value of the output of one of these versions of the input or delayed input onto the output line. Thus at each clock pulse a randomly delayed replication of the input is presented on the output line. In alternative realizations of the random delay flip-flops FF1 and FF2 may be connected as a shift-register or binary counter to a single trigger and noise source; greater or lesser depths of random delay may be obtained by extension or reduction in the circuit, in which case feedback may be necessary around the flip-flops connected to the trigger in order to prevent them assuming unwanted configurations of states.

One example of computation with autocorrelated sequences is when a pseudo-random set of sequences are generated to represent quantities by mark-space modulation. In mark-space modulation a binary signal of arbitrary frequency is used to represent a quantity by the ratio of the time on to the total time over an interval or to the time off over that interval. The signal is in the former case an example of the affine representations, symmetric or asymmetric according to the interpretation of the fully off condition, and in the latter an example of the projective representation or of the hyperbolic or trigonometric representations. In fact mark-space modulation is a simple example of asynchronous representation by pseudo-random stochastic sequences. The autocorrelation depth of a mark-space sequence depends on the extent to which its frequency of change of state is regular. It can be made small by using a varying or randomly varying frequency. The cross correlation between two sequences depends on the extent to which their frequencies are the same or integral multiples of one another. Thus two mark-space ratio sequences are pseudo-random and may be used as inputs to the multipliers previously described and other computing elements to be described provided their frequencies are not integral multiples. This form of asynchronous, pseudo-random representation is readily extended to multi-level or multi-state sequences.

In Fig. 9, the outputs of the first flip-flop are applied to a second flip-flop which therefore emits an ON level to gate 4 and OFF to gate 5 or OFF to gate 4 and ON to gate 5 with equal probability. The probability of an output from gate 6, P(Z), is therefore \( \frac{1}{2} \) (sum of probabilities of outputs from gates 4 and 5), i.e. \( \frac{1}{2}P(A) + P(B) \) since the outputs of 4 and 5 never arrive simultaneously at 6, so that Fig. 9 is an adder. There is no difference between adders for the symmetric and asymmetric binary affine representations.

We now describe more circuitry for the implementation of addition in the affine representations and make some remarks on switching theory. The expression \( a + b + ab \), representing a particular network of gates, is a redundant form of \( a + b \), since the switching tables of the two functions are the same and the second function contains fewer terms than the first, i.e. the forms contain the same literals, but the second form contains fewer letters. If the simple OR gate is used for the purposes of addition in the affine representations, the output is not the sum of the input lines but in the affine asymmetric representation is their sum minus their products and a corresponding function in the symmetric representation. The product term is effectively absent, since the signal occurring for inputs on both lines is indistinguishable from that occurring for an input on one of the lines. Thus, although the OR gate is useful in some computations, stochastic logic is required to implement the operation of addition. To sum a number of lines, the unweighted adder for the affine binary representation chooses one of them at random and outputs the value of that line. Thus the output of a k-input adder is \( 1/k \) times the sum of its inputs. If the input lines are chosen at random, they are chosen with equal probability.

Referring again to Fig. 9, which illustrates circuitry for stochastic addition in the case of two inputs (the two input adder \( k \) of Fig. 1), when the signal from the
random noise source exceeds a preset threshold, a trigger pulse is applied to the clock input of a first flip-flop which then changes state because its inputs are grounded and is thus in a random state at the time of a clock pulse, when its state is transferred to a second flip-flop. The mean rate of random pulses for the trigger to the first flip-flop should exceed the clock-pulse rate preferably by a factor of at least ten.

The similarity of the adder of Fig. 9 to the element previously described for removing autocorrelation will be noted — in the latter, delayed versions of the same input were selected at random whereas the adder of Fig. 10 selects one of several different inputs at random. In fact the de-correlator can be constructed from a chain of isolators and an adder of the type described. Alternative realizations of the adder are possible when several inputs are to be summed and the same remarks apply as for the random delay. If equal weights are not to be applied to all inputs then either the inputs may be weighted by passing them through a multiplier one input of which is a sequence generated with constant probability, or, preferably, biased probabilities may be used in the random selection of an input; the output sum is then weighted by the probability of selection of the input. The biased probabilities may be generated through the use of flip-flops with random states gating the input lines or through other forms of generation of multi-state stochastic sequences, or alternatively an adder with a large number of inputs may have these commoned in groups to the incoming lines so weighting the sequences on these lines.

Since the logical inverter performs the operation of additive inversion in the affine symmetric representations it may be used in conjunction with an adder to obtain a subtractor. If the clock pulses to the adder are obtained from a local oscillator it may be used in asynchronous computation, for example with mark-space sequences. In an alternative realization of the asynchronous adder the lower flip-flop may be made a multivibrator, the noise source and trigger may be removed, and one output of the lower flip-flop may be connected through a capacitor to the clock line of the upper flip-flop, the input lines of which are grounded. In another alternative the upper flip-flop may be replaced by a multivibrator and the lower flip-flop, trigger and noise source removed. In this case variable weighting may be obtained by varying the mark-space ratio of the flip-flop. It is desirable if these asynchronous realizations are used with mark-space modulated sequences that the inputs and multivibrators do not have frequencies which are integrally related. Asynchronous adders with more than two inputs may be realised by applying these modifications to the multiple input adders described. If two or more multivibrators are used within a multiple input adder it is desirable that their frequencies are not integral multiples of a common frequency.

Generation of random sequences in the computer may be required for various purposes. The stochastic adder has within it a random element and may generate a random sequence even when its inputs are deterministic. In general, however, the random sequences which carry information in a stochastic computer will be generated either internally as the output of stochastic constants and integrators or externally from the conversion of events or analog or digital quantities into a stochastically coded form. For quantities, one element which does this is a comparator with binary output, one of whose inputs is random or pseudo-random and the other of which is fixed (stochastic constant) or the stochastic integrator state. The only requirement on inputs to the comparator is that it should be possible to say which one is greater in magnitude and thus two voltages or two digitally coded numbers or even two pressures might be used. The random input is generally required to take all its possible levels with equal probability (though specific random distributions may be very useful for nonlinear conversion) and this may be achieved by random selection of its levels in a similar way to that in which an input was randomly selected by the adder of Fig. 9.

Fig. 10 illustrates one generator of random sequences, the analog-stochastic converter (a) of Fig. 1. When the random noise from the noise source exceeds a certain threshold, a trigger pulse is applied to the clock line of the appropriate flip-flop. (There will normally be more than four flip-flops in a practical circuit). The flip-flops are in a random state and hence give a random level to the comparator via the digital-to-analog convertor. If at a clock pulse the analog input is greater than this random input, then the output flip-flop goes on, otherwise it is off. The sequence of outputs is a stochastic representation of the analog input in the affine binary representation if the digital-to-analog convertor is linear.

Fig. 11 illustrates the digital-stochastic convertor (i) of Fig. 1. Here the digital input is fed directly to a digital comparator where it is compared with the random digital output of the flip-flops. The output of the flip-flop is obtained in the same way
Many alternative realizations of binary stochastic converters are possible;—as for the
factor previously described in the above equations (11) and (13). These may be
built up from a number of the stochastic switches (1) and (2) and of
a pair of binary comparators (3) and (4). The combination of these switches
and comparators, together with binary counters (5), may be used to
realize the symmetric projective and unsymmetric projective representations which
have been described in the previous section. A possible configuration is shown
in Fig. 24, which is used to generate the hyperbolic and trigonometric functions
represented in fields (5) and (6).
put to feed other stochastic computing elements. This digital output may also be used at the output interface to feed-out the state of the integrator which information may be used directly or converted to analog or other form. In the affine representation the digital-stochastic converter will generally be linear so that the probability that the output will be on after the next clock pulse is n/K when a counter with K+1 states is in its n'th state, in which case the output sequence represents the integral of the sum of the quantities represented by the inputs provided the counter does not 'limit', that is provided it does not receive increment or decrement signals at a clock pulse when in its K'th or zeroth states respectively. The state of an integrator at the beginning of a computation is not necessarily specified by the circuit and may be set by external sources or by another section of the computer, and similarly it may be changed at any stage of the computation. Thus an integrator may be used as an input interface from other apparatus, for example a digital computer, and an integrator without inputs may be used to hold a constant during computation. In some uses of the integrator, particularly when it is used to 'track-or-store', gates may be placed in the increment and decrement lines so that an input, the 'hold' input, may be used to prevent changes of state. In other realizations the number of states of the counter, the 'time-constant', may be set by limiting the counter to a smaller or larger part of its range, for example by use of another counter whose state determines the maximum and minimum states of the integrating counter. The random generator which determines the stochastic output of the integrator is generally but not necessarily required to generate the random output to one side or the other in such a way that the range of the counter feeding the other side, and hence if the range of the counter is changed it may be desirable to change the range of the random generator; this is certainly so if the digital to stochastic converter is required to be linear in the sense described above.

The integrator may have many forms of output but they will be controlled to some extent by its state which can for example control a stochastic converter to generate binary or multi-level stochastic sequences, or set-up the state of another integrator, or feed-out information from the stochastic computer to other devices, to processes, operators and so on. The interpretation which is placed upon the output of an integrator will depend upon its purpose and upon the representation being used.

Many schemes for feedback of the output or state of an integrator to its input to control the effect of its input on its state are possible and are especially useful when the integrator is used to convert the stochastic representation of quantity or events back to its original or some other form. If the linear stochastic output of an integrator is fed back to one of its inputs via an inverter as in Fig. 14 then the configuration is an "ADDIE" or adaptive digital element of the kind described in British Patent Specification No. 1,069,159. This configuration is particularly attractive in that limiting cannot occur and the state of the counter may be used to estimate the probability that the input is on at a clock pulse and hence may be used to estimate the event or quantity represented by a stochastic sequence, for example an K+1 state ADDIE in its n'th state may be taken as representing:

\[ V = \frac{nE}{K} \]  
(26)
in the asymmetric affine binary representation;

\[ V = \frac{(2n-K)E}{K} \]  
(27)
in the symmetric affine binary representation;

\[ V = \frac{n}{a(K-n+1)} \]  
(28)
in the projective binary representation;

the latter estimate and similar ones for the hyperbolic and trigonometric representations are biased and their accuracy depends upon the extent to which the K'th state of the counter is not attained, whereas the two former estimates are unbiased. In other schemes for feedback around the integrator, gates in the increment and decrement lines may be fed by outputs dependent upon the state of the integrator so that the probability of an increment or decrement is determined not only by the inputs but also by the state of the integrator.

The integrator with or without feedback may be used as a function generator in which case the reversible counter will generally have far less states than when it is used for integrator or as an interface. Particular interest is attached to those integrators in which the output is determined exactly rather than statistically by the integrator state since these are capable of realization with fewer components. An example of this is the switching function in which an integrator with 2j states is used, the output being on when it is in one of its upper j states and off otherwise. As j increases this approximates more and more to a discontinuity known as a switching or threshold function. Fig. 25 illustrates a switching function with a four-state counter (j=2).
The output \( p = \frac{a^2}{(a^2 + \overline{a^2})} \), or
\[
\frac{r^2}{r^2 + (1+r)^2}
\]
where \( r \) is the scaled value representing \( a \).
Recall that the symbol for this element is shown at (s) of Fig. 1.

Fig. 15 illustrates a multiplier for analog inputs and incorporating an ADDIE which is used as a stochastic to digital converter. A plurality of levels in the adaptive element corresponds to the several digits in a word.

Fig. 16 is the multiplier of Fig. 15 used as a squarer, isolation therefore being provided for the second input.

Fig. 17 is an adder incorporating an ADDIE as a stochastic to digital converter.

Fig. 18 is an adder utilising a CCFF with interconnected logical elements forming gates. Considering equation (23), if we replace \( X \) by \( AB \lor AB \) and \( Y \) by \( AB \), then we can rewrite (23) as:
\[
P(Z) = \frac{P(AB) \lor AB}{P(AB)}
\]
Therefore
\[
P(Z) = \frac{P(A)P(B) + P(A)P(B)}{P(A)P(B)}
\]
\[
P(Z) = \frac{P(A)}{P(A)} + \frac{P(B)}{P(B)}
\]
(29)

Fig. 19 is a cross-correlator which calculates the covariance of inputs \( X \) and \( Y \), taking the time over which averaging takes place into account. The output \( G \) has value
\[
G(t) = G(o) + \frac{t}{0} \int \exp(s-t) \]
(X - \( \overline{X} \))(Y - \( \overline{Y} \) ) ds where \( X, Y, \overline{X}, \overline{Y} \) denote the expected values (average or mean) of \( X, Y \) respectively, where \( \exp \) denotes the exponential function, and where \( \overline{t} \) is the period of time during which the inputs are applied and where \( G(o) \) is a constant.

Fig. 20 illustrates a stochastic computer with analogue inputs and outputs in the symmetric affine representation for solving the differential equation
\[
\frac{1}{N} \bigg( 1 + \bigg( \frac{1}{N} \bigg) Z + \frac{1}{K} \bigg) \bigg( Z + \frac{1}{N} \bigg) Z + \frac{1}{KN} Z = \frac{X}{KN} \bigg)
\]
where a dot above a variable denotes differentiation with respect to the clock-pulse time scale. (This equation arises in the calculation of the transfer function of a system with natural frequency \( \sqrt{KN} \) and \( K \).

If \( X \) is the input to the converter, \( Y \) the quantity represented by the output of the first integrator, and \( Z \) the final output, and if the first integrator has \( K + 1 \) states, and the second integrator has \( N + 1 \) states, then
\[
N \overline{Z} = Y - Z \text{ and }
\]
\[
\overline{K} \overline{Y} = X - Z
\]
\[
\overline{KNZ} = \frac{Y - Z}{Z - X} - K \overline{Z}
\]
\[
\frac{1}{N} \bigg( 1 + \bigg( \frac{1}{N} \bigg) Z + \frac{1}{KN} Z = \frac{X}{KN} \bigg)
\]
(31)

Some computations in some representations cannot be realised by logical elements operating on a finite section of the input sequence to produce an output sequence which is an unbiased representation of the result of the computation, for example division in the affine representations, square-roots in affine and projective representations, subtraction in the projective representations. These computations may be realised or approximated by conversion between representations and in many applications different representations will be used in different parts of the computer, or by descent techniques for minimization, or by feedback around an integrator, or by special-purpose function-generators. In feedback around an integrator the technique is to use an integrator as an element with high-gain over many clock-pulses and place the function whose inverse is required in its feedback loop. This is an approximate technique in that the output sequences are generally not unbiased estimators of the
Figs. 24, 25 and 26 are examples of three elements using feedback around an integrator. The stochastic divider Figs. 24 has a multiplier and inverter in the feedback loop of an integrator and the quantity represented by Z approximates to that represented by Y divided by that represented by X. In the symmetric affine representation this system is unstable if X represents a negative quantity and hence it is only a half-scale divider; division by negative quantities may be realized by removal of the inverter. The stochastic square-root divider Fig. 25 has a multiplier with isolator used as a squarer together with an inverter in the feedback loop of an integrator and the quantity represented by Z approximates to the square root of that represented by Y. The stochastic subtractor for the projective representation Fig. 26 has a projective adder and an inverter in the feedback loop of an integrator and the quantity represented by Z approximates that represented by Y minus that represented by X. Alternative forms of feedback around integrators may be used to compute or approximate functions not readily obtained by the straightforward application of logical elements.

Fig. 27 illustrates a divider for positive analog quantities. Positive quantities X and Y are converted into the projective binary representation and the two streams representing them gated through AND gates into a cross-coupled flip-flop the output of which is estimated by an ADDIE and converted to its equivalent analog form by a digital to analog converter in the projective representation. The output Z equals X divided by Y.

Fig. 28 illustrates division of one variable by the average of another. The positive quantity Y is converted to the affine asymmetric binary mode and its representation is averaged by an ADDIE. This representations of the averaged or exponentially smoothed input Y is gated together with the projective representations of the input X into a cross the output of which is estimated by an ADDIE and converted to its equivalent analog form by digital to analog converter in the projective representation. The first ADDIE will generally have far more states than the second ADDIE.

The output Z equals J Y divided by \( \exp(\delta t) \) ds, where \( \exp \) denotes the exponential function.

Fig. 29 shows a stochastic computer used in process control with particular application to the control of a distillation column. Tanks T1-T4 deliver the ingredients to the distillation column, the proportions being varied by opening and shutting valves V1-V4. P is a pump and PH a pre-heater for the mixture. SI - S6 are different stages in the process and provide monitored information which is fed to an analog to digital converter A--D. Coolant is injected at IN and emerges at OUT. Some of the circulating reflux is diverted through a flow meter and a composition meter whose readings are fed to A -- D. The level of the composition in the base of the column is measured and fed to A -- D, as is the level in the BOTTOM storage tank and the temperature of the heating jacket. The various outputs of A -- D pass through an analog or digital-to-stochastic interface, such as has been described with reference to Figs. 10 and 11, and the derived values are then processed by the stochastic computer. Signals from the computer pass through a stochastic to digital interface, such as has been described with reference to Fig. 14, and are then applied either after digital-to-analog conversion or directly to the control of valves V1 -- V4, the reflux, to the controls governing the temperature to be maintained in the base or to the levelling valve L. In this example the stochastic computer may contain a simulation of the distillation column by stochastic elements, such as that shown in Fig. 20, simulating each plate and may also contain a predictor, such as that shown in Figs. 30 and 31, to predict the result of varying valves V1 -- V4. The mode of control may be steepest descent optimization, such as described with reference to Fig. 32, to keep the contents of T1 to T4 at prescribed ratios.

Figs. 30 and 31 illustrate the use of the stochastic computer as a Bayesian predictor in which the occurrence of an event Ei is used as evidence to predict the probability of occurrence of an event A. The computation which is implemented by the predictive section (Fig. 31) is that of multiplication of likelihood ratios:--

\[
\text{if } E_i \ldots E_N \text{ are known to have occurred then}
\]
required output and also under certain conditions the loop may be unstable and oscillatory or indeterminate outputs be represented.

Figs. 24, 25 and 26 are examples of three elements using feedback around an integrator. The stochastic divider Fig. 24 has a multiplier and inverter in the feedback loop of an integrator and the quantity represented by \( Y \) divided by that represented by \( X \). In the symmetric affine representation this system is unstable if \( X \) represents a negative quantity and hence it is only a half-scale divider; division by negative quantities may be realized by removal of the inverter. The stochastic square-root solver Fig. 25 has a multiplier with isolator used as a squarer together with an inverter in the feedback loop of an integrator and the quantity represented by \( Z \) approximates to that represented by \( Y \). The stochastic subtractor for the projective representation Fig. 26 has a projective adder and inverter in the feedback loop of an integrator and the quantity represented by \( Z \) approximates to the square root of that represented by \( Y \). The stochastic subtractor for the projective representation Fig. 26 has a projective adder and inverter in the feedback loop of an integrator and the quantity represented by \( Z \) approximates to that represented by \( Y \). Alternative forms of feedback around integrators may be used to compute or approximate functions not readily obtained by the straightforward application of logical elements.

Fig. 27 illustrates a divider for positive analog quantities. Positive quantities \( X \) and \( Y \) are converted into the projective binary representation and the two streams representing them gated through AND gates into a cross-coupled flip-flop the output of which is estimated by an ADDIE and converted to its equivalent analog form by a digital to analog converter in the projective representation. The output \( Z \) equals \( X / Y \).

Fig. 28 illustrates division of one variable by the average of another. The positive quantity \( Y \) is converted to the affine symmetric binary mode and its representation is averaged by an ADDIE. This representation of the averaged or exponentially smoothed input \( Y \) is gated together with the projective representations of the input \( X \) into a ccf the output of which is estimated by an ADDIE and converted to its equivalent analog form by digital to analog converter in the projective representation. The first ADDIE will generally have far more states than the second ADDIE.

The output \( Z \) equals \( X / Y \) divided by \( \exp (s-t) \), where \( \exp \) denotes the exponential function.

Fig. 29 shows a stochastic computer used in process control with particular application to the control of a distillation column. Tanks TI-T4 deliver the ingredients to the distillation column, the proportions being varied by opening and shutting valves VI—V4. \( P \) is a pump and \( PH \) a pre-heater for the mixture. SI — S6 are different stages in the process and provide monitored information which is fed to an analog to digital converter A—D. Coolant is injected at IN and emerges at OUT. Some of the circulating reflux is diverted through a flow meter and a composition meter whose readings are fed to A — D. The level of the composition in the base of the column is measured and fed to A — D, as is the level in the BOTTOM storage tank and the temperature of the heating jacket. The various outputs of A — D pass through an analog or digital-to-stochastic interface, such as has been described with reference to Figs. 10 and 11, and the derived values are then processed by the stochastic computer. Signals from the computer pass through a stochastic to digital interface, such as has been described with reference to Fig. 14, and are then applied either after digital-to-analog conversion or directly to the control of valves VI — V4, the reflux, to the controls governing the temperature to be maintained in the base or to the levelling valve \( L \). In particular the stochastic computer may contain a simulation of the distillation column by stochastic elements, such as that shown in Fig. 20, simulating each plate and may also contain a predictor, such as that shown in Figs. 30 and 31, to predict the result of varying valves VI — V4. The mode of control may be steepest descent optimization, such as described with reference to Fig. 32, to keep the contents of TI 105 to T4 at prescribed ratios.

Figs. 30 and 31 illustrate the use of the stochastic computer as a Bayesian predictor in which the occurrence of an event \( E \) is used as evidence to predict the probability of occurrence of an event \( A \). The computation which is implemented by the predictive section (Fig. 31) is that of multiplication of likelihood ratios:

\[
p(A | E_1 \ldots E_n) = \prod_{i=1}^{n} p(A | E_i)
\]

\[
p(E_1 \ldots E_n | A) = \prod_{i=1}^{n} p(E_i | A)
\]

The output \( Z \) equals \( X / Y \) divided by \( \exp (s-t) \), where \( \exp \) denotes the exponential function.
\[
\frac{p(A/E_1 \ldots E_N)}{p(A/E_1)} = \frac{p(A)}{p(A/E_1) p(A)} \cdots \frac{p(A/E_N)}{p(A/E_1) p(A)}
\]

\[
P(Z) = \frac{p_A}{1 - P_A} \times \frac{P_A/E_1}{1 - P_A/E_1} \times \ldots
\]

so that from equation (33)

\[
P(Z) = L \times L_1 \times L_2 \times \ldots L_N
\]

which, for equation (32) is equal to

\[
P(A/E_1 \ldots E_N)
\]

so that

\[
P(Z) = P(A/E_1 \ldots E_N)
\]

The output of the ADDIE in Fig. 30 estimates \(P(Z)\) and hence becomes equal to \(P(A/E_1 \ldots E_N)\) which is the predicted probability of occurrence of \(A\) given the events \(E_1 \ldots E_N\). This may be read out digitally or through a digital-analog converter, or the stochastic output of the integrator may be used for further stochastic computation.

Although both sections of the Bayesian estimator and predictor have been here shown as realized by stochastic logic alternative realizations part or whole of the systems may be realized by other forms of computation such as the digital computer, in particular the output of the estimating section is very suitable for digital computation. The output of this section gives estimates of normalized likelihood ratios in a convenient symmetric form and has an advantage over Bayesian predictors using logarithmic transformations. In another alternative many such estimator/predictor systems may be combined to give conditional probability computers and maximum likelihood predictors. The events \(E_i\) need not be independent of the predicted events such as \(A\) but can be logically combinations of past occurrences of these. In a further alternative the events \(E_i\) are weighted with the confidence that can be placed in them and the output is also weighted according to its reliability.
the correctness of past predictions. In a further alternative the outputs or states of integrators 1 to N are used to assess the usefulness of events \( E_1 \) to \( E_N \) respectively as predictors (since the evidence is useless if the states of the integrators do not deviate appreciably from mid-range) and may be used to change the content of the evidence.

This predictor may be applied in any situation where evidence is available which may be eventually dichotomized into the occurrence or non-occurrence of events. It is of particular use however in medical diagnosis where \( E_a \) may be symptoms and other evidence of the patient's condition and A may be a particular disease or condition; meteorological forecasting where the \( E_i \) may be information about meteorological variables such as barometric pressure in various localities and A may be a particular aspect of weather to be expected; automatic control where \( E_i \) may represent conditions of the plant or controlled system and past control actions and A may represent a future condition of the plant or controlled system; pattern recognition and the detection of signals in noise where \( E_i \) may represent features of the pattern or signal plus noise and A may be a particular classification of the pattern or signal.

The arrangement of CCFF's and integrators is duplicated for each input \( E_i \) to \( E_N \) of Fig. 30, and the corresponding gate sections of Fig. 31 are similarly duplicated for the signals \( P_A/P_B \) and the signals \( E_i \).

One realization of a variable processor which is of great importance is the steepest descent computer for determining or enforcing a relationship between a set of variables. In this computer variables whose value is to be determined are represented as the outputs of integrators whose inputs are functions of some or all of the variables. With appropriate functional relationships the integrator outputs may be forced to vary so as to satisfy the required relationship. The computer is then said to 'model' the relationship and may be used to compute other sets of variables satisfying it, for example for purposes of pattern recognition or prediction. In alternative realizations the appropriate functional relationship may be unknown and the computer may itself determine it by parameter perturbation or other techniques.

Fig. 32 illustrates a stochastic computer circuit in the affine symmetric binary representation for the steepest descent approach to a linear relationship. It is required to find 'weights', \( w_1, \ldots, w_N \), for the inputs \( x_1, \ldots, x_N \), such that the output \( z \) is equal to the required output \( y \), that is to determine the \( w_i \) such that:

\[
z = y = \frac{1}{N} (w_1x_1 + w_2x_2 + \ldots + w_Nx_N) (35)
\]

where

\[
x = -(w_1x_1 + w_2x_2 + \ldots + w_Nx_N)
\]

The input representing \( x_i \) passes through the multiplier \( M^1 \), together with the output of \( I_{m1} \), representing \( w_i \), and their product goes into the adder together with similar products to produce \( z \). The input \( x_i \) also passes through an isolator I to a second multiplier \( M^2 \), together with the inverted output representing \(-z\) into one input of the integrator, \( I_{m2} \), and to a third multiplier \( M^3 \), together with the input representing \( y \), the required output.

When the 'adapt' line is on the outputs of the integrator change so as to minimize the difference between \( z \) and \( y \). To use the device as a predictor, in the absence of the signal \( y \) the adapt line may be put off and the output \( z \) will predict the signal \( y \) corresponding to the inputs \( x_1 \) to \( x_N \). \( y \) itself may often be zero in which case it is required to find a linear relationship between the variables \( x_i \) themselves. Some of the integrators may be used to hold constant weights rather than be placed in the adapt mode (i.e. their adapt lines are not on) in which case the variable weights are then required to satisfy the constraints set by the fixed weights. In application to pattern recognition the inputs \( x_1, \ldots, x_N \) may be binary signals representing the presence or absence of features of patterns and the output may be a classification of patterns. Further transformation may be applied to the output of the adder before it is fed out and back; the computation is not then necessarily steepest descent but it is convergent if the transformation applied is monotone. Stochastic elements have an especial advantage over other variable weighting devices with limited storage in that they may be shown to converge for a large class of training sequences. The arrangement of integrators and multipliers is duplicated for each \( X_i \).

Fig. 33 illustrates a particular application of the device of Fig. 32 applied to the division of two quantities of the affine symmetric binary representation. This is a four-quadrant divider as opposed to the two-quadrant divider of Fig. 24. The adder of Fig. 32 is unnecessary since only one weight, \( W \), and one variable, \( X \), are in use. The output \( W \) equals \( Y/X \).

Fig. 34 illustrates relay control of a motor using stochastic elements in the binary symmetric affine representation. The position and velocity of the motor shaft given by a positional pick-off and tacho-
Thus if each value of \( u_{ij} \) is represented as the output of an ADDIE placed at the grid point \((a_i, a_j)\) we have the following relationships between the output of one ADDIE and its neighbours:

\[ u_{i-1,j} - 2u_{ij} + u_{i+1,j} + u_{i,j-1} - 2u_{ij} + u_{ij} + 1 = 0 \]

This equation may be enforced by the computing element of Fig. 37 which is for the affine binary symmetric representation and

\[ 4u_{ij} = u_{i-1,j} + u_{i+1,j} + u_{i,j-1} + u_{i,j+1} \]
consists of an integrator connected as an ADDIE with its other input fed by a four-input, equally weighted adder. The ADDIE realizes the function

\[
U = \frac{1}{4} (x_1 + x_2 + x_3 + x_4)
\]

so that if the output of the ADDIE is \( u \), then its input should be \( u_{-1}, u_0, u_1 \), etc., which are the outputs of the neighbouring four ADDIES. A symbol for an ADDIE connected in this way is shown in Fig. 11(1) and a network of ADDIES interconnected as described to solve equation 26 is shown in Fig. 38. These units are interconnected in a two-dimensional rectangular array with output of every element going to the inputs of its four neighbours. At the edges of the array there will be inputs not connected to other Laplacian elements and these are connected to stochastic constants representing the boundary condition BC in the affine binary symmetric representation.

Stochastic computing elements lend themselves particularly to the solution of partial differential equations because they may be realized in a compact form and are readily interconnected, thus providing a covering for the area for which the solution of the equation is required. The boundary conditions BC on the mesh which subdivides the area correspond to the values of the inputs at the interconnections between elements.

Well-known mathematical methods such as relaxation techniques are programmable in a straightforward way for the stochastic elements.

The adaptive facilities of the ADDIE are particularly suitable for implementing the iterative minimisation utilised in relaxation methods.

Thus a stochastic computer may utilise various forms of computing element such as binary, ternary or multiple-level AND-gates, OR-gates, inverters and so on, or threshold logic in which the logical output depends upon the number and weight attached to the inputs, or stochastic logic in which the statistical properties of the output rather than the output itself are determined by the input, or non-logical elements such as amplifiers, together with binary and multi-level storage elements such as flip-flops, ferrite-cores, capacitors, inductors, pneumatic and hydraulic cylinders, optical devices and so on. Coupled to the stochastic computer may be conventional analog or digital computers and interfaces with industrial processes, human operators and so on. In some part of this configuration one or more computations will be carried out which involve the representations of an event or quantity by a sequence of logic levels or states whose statistical properties are determined by that level or quantity.

WHAT WE CLAIM IS:

1. A computer including a digital computing element or elements, means for representing input information stochastically by the probability that a level in a clocked sequence of logic levels will be ON, means for applying the information so represented to the computing element or elements wherein computation is performed in a digital manner and means for converting the stochastically represented outputs of the computing element or elements to analogue or digital values.

2. An arrangement according to claim 1 wherein items of information applied to the computing element or elements are normalised so that each item takes a value in a standard interval.

3. An arrangement according to claim 2 wherein the standard interval is the range 0 to 1 and in which the probability of a particular level being ON is arranged to equal the normalised value of the item of information being applied to the computing element or elements.

4. An arrangement according to any one of the preceding claims in which at least one of the computing elements utilises binary logic devices.

5. An arrangement according to any one of the preceding claims in which at least one of the computing elements utilises ternary logic devices.

6. An arrangement according to any one of the preceding claims in which at least one of the computing elements utilises threshold logic devices.

7. An arrangement according to any one of the preceding claims wherein two computing elements are arranged to operate in parallel so as to compute information separately.

8. An arrangement as claimed in any one of the preceding claims which includes an adaptive element as hereinbefore defined.

9. An arrangement as claimed in any one of the preceding claims wherein a clock signal is arranged to be applied to a synchronous logic device in a computing element of the arrangement so as thereby to control the timing of a computing operation.

10. An arrangement as claimed in any one of the preceding claims which includes an integrating device.

11. An arrangement as claimed in any one of the preceding claims which includes a memory wherein results of computations and values of quantities are arranged to be stored.

12. An arrangement as claimed in claim...
10 wherein the value of a constant quantity is arranged to be held in the integrating device.
13. An arrangement as claimed in any one of the preceding claims having interconnections through external sensors and information channels to an environment.
14. An arrangement as claimed in any one of the preceding claims wherein a first sequence is arranged to be de-correlated from another sequence by being delayed through one or more time delays relative to the said other sequence.
15. An arrangement as claimed in claim 14 wherein a flip-flop acts as an isolating unit-delay.
16. An arrangement as claimed in any one of the preceding claims wherein the biased probability of selection of one of a number of logic levels is generated by the gating of a number of flip-flops in random states to appropriate ones of a number of lines carrying the said logic levels.
17. An arrangement as claimed in any one of the preceding claims wherein asynchronous information computation is arranged to be carried out with an ordinarily synchronous element, quantities being represented by mark-space sequence and a local oscillator being arranged to provide the clock pulses.
18. An arrangement as claimed in any one of the preceding claims which includes an analog-or-digital-to stochastic converter which is arranged to be fed a constant input so as to generate a stationary sequence.
19. An arrangement as claimed in any one of the preceding claims which includes an integrator having increment and decrement lines having gates therein with facility for applying a “hold” signal thereto so as to prevent change of state of the integrator.
20. An arrangement as claimed in any one of the preceding claims which includes a delay element substantially as hereinbefore described with reference to Fig. 7 of the drawings accompanying the complete specification.
21. An arrangement as claimed in any one of the preceding claims which includes an element substantially as hereinbefore described with reference to Figs. 2(a) and 2(b) of the drawings accompanying the complete specification.
22. An arrangement as claimed in any one of the preceding claims which includes a cross-coupled flip-flop substantially as hereinbefore described with reference to Fig. 3 of the drawings accompanying the complete specification.
23. An arrangement as claimed in any one of the preceding claims wherein the device of Fig. 4 in the drawings accompanying the complete specification is arranged to function as a multiplier for the projective binary representation.
24. An arrangement as claimed in any one of the preceding claims which includes a delay element substantially as hereinbefore described with reference to Fig. 21 of the drawings accompanying the complete specification.
25. A computing device including a multiplier substantially as hereinbefore described with reference to Fig. 6 of the drawings accompanying the complete specification.
26. A computing device including a squarer substantially as hereinbefore described with reference to Fig. 8 of the drawings accompanying the complete specification.
27. A computing device including an adder substantially as hereinbefore described with reference to Fig. 9 of the drawings accompanying the complete specification.
28. A computing device including an analog-to-stochastic converter substantially as hereinbefore described with reference to Fig. 10 of the drawings accompanying the complete specification.
29. A computing device including a digital-to-stochastic converter substantially as hereinbefore described with reference to Fig. 11 of the drawings accompanying the complete specification.
30. A computing device including an integrator substantially as hereinbefore described with reference to Fig. 12 of the drawings accompanying the complete specification.
31. A computing device including an integrating arrangement substantially as hereinbefore described with reference to Fig. 13 of the drawings accompanying the complete specification.
32. A computing device including an adaptive element substantially as hereinbefore described with reference to Fig. 14 of the drawings accompanying the complete specification.
33. A computing device including a multiplier substantially as hereinbefore described with reference to Fig. 15 of the drawings accompanying the complete specification.
34. A computing device including a squarer substantially as hereinbefore described with reference to Fig. 16 of the drawings accompanying the complete specification.
35. A computing device including an adder substantially as hereinbefore described with reference to Fig. 17 of the drawings accompanying the complete specification.
adder substantially as hereinbefore described with reference to Fig. 18 of the drawings accompanying the complete specification.

37. A computing device including a correlator substantially as hereinbefore described with reference to Fig. 19 of the drawings accompanying the complete specification.

38. A computing device including a computer for the solution of differential equations substantially as hereinbefore described with reference to Fig. 20 of the drawings accompanying the complete specification.

39. A computing device including a non-linear stochastic converter substantially as hereinbefore described with reference to Fig. 22 of the drawings accompanying the complete specification.

40. A computing device including a switching function generator substantially as hereinbefore described with reference to Fig. 23 of the drawings accompanying the complete specification.

41. A computing device including a divider substantially as hereinbefore described with reference to Fig. 24 of the drawings accompanying the complete specification.

42. A computing device including a square-root solver substantially as hereinbefore described with reference to Fig. 25 of the drawings accompanying the complete specification.

43. A computing device including a subtractor substantially as hereinbefore described with reference to Fig. 26 of the drawings accompanying the complete specification.

44. A computing device including a divider substantially as hereinbefore described with reference to Fig. 27 of the drawings accompanying the complete specification.

45. A computing device including an averaging and dividing arrangement substantially as hereinbefore described with reference to Fig. 28 of the drawings accompanying the complete specification.

46. A computing device including a control system substantially as hereinbefore described with reference to Fig. 29 of the drawings accompanying the complete specification.

47. A computing device estimating and predicting arrangement for a Bayesian predictor as hereinbefore described with reference to Figs. 31 and 32 of the drawings accompanying the complete specification.

48. A computing device substantially as hereinbefore described with reference to Fig. 33 of the drawings accompanying the complete specification.

49. A computing device including a control device substantially as hereinbefore described with reference to Fig. 34 of the drawings accompanying the complete specification.

50. A computing device including a control arrangement substantially as hereinbefore described with reference to Fig. 35 of the drawings accompanying the complete specification.

51. A computing device including an adaptive filter substantially as hereinbefore described with reference to Fig. 36 of the drawings accompanying the complete specification.

52. A computing device including a data smoothing arrangement substantially as hereinbefore described with reference to Fig. 37 of the drawings accompanying the complete specification.

53. A computing device substantially as hereinbefore described with reference to Fig. 38 of the drawings accompanying the complete specification.

54. A computing arrangement substantially as hereinbefore described with reference to Fig. 39 of the drawings accompanying the complete specification.

S. R. CAPSEY,
Chartered Patent Agent,
For the Applicants.
Fig. 1
Fig. 2 (a)

(b)

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
</tr>
</thead>
<tbody>
<tr>
<td>ON</td>
<td>OFF OPEN</td>
</tr>
<tr>
<td>OFF</td>
<td>OPEN ON ON</td>
</tr>
<tr>
<td>OPEN</td>
<td>OFF ON ON OPEN</td>
</tr>
</tbody>
</table>
This drawing is a reproduction of the original on a reduced scale Sheet 3
Fig. 23.

Fig. 24.

Fig. 25.
This drawing is a reproduction of the original on a reduced scale.

Sheet 9

Fig. 26

Fig. 27

Fig. 28
Fig. 29
Fig. 33

Fig. 34

REQUIRED SHAFT POSITION
Fig. 35
Fig. 36.
Fig. 37.

Fig. 38.