Available online at www.sciencedirect.com
SCIENCE DIRECT
C. R. Biologies 329 (2006) 137–147
COMPTES RENDUS BIOLOGIES
http://france.elsevier.com/direct/CRASS3/

Biological modelling / Biomodélisation

Adaptive properties of living beings: Proposal for a generic mechanism. (Self-Programming Machines III)

Jean-Paul Moulin

Département d'information et de recherche médicale, Centre hospitalier interdépartemental de Clermont-de-l'Oise, 2, rue des Finets, 60607 Clermont-de-l'Oise cedex, France

Received 7 December 2004; accepted after revision 6 December 2005

Available online 18 January 2006

Presented by Michel Thellier

Abstract

Living systems are capable to have appropriate responses to unpredictable environment. This kind of self-organization seems to operate as a self-programming machine, i.e. an organization able to modify itself. Until now the models of self-organization of living beings proposed are functions solutions of differential systems or transition functions of automata. These functions are fixed and these models are therefore unable to modify their organization. On the other hand, computer science propose a lot of models having the properties of adaptive systems of living beings, but all these models depend on the comparison between a goal and the results and ingenious choices of parameters by programmers, whereas there are no programmer's intention nor choice in the living systems. From two best known examples of adaptive systems of living beings, nervous system and immune system that have in common that the external signals modify the rewriting of their organization and therefore work as self-organizing machines, we devised machines with a finite set of inputs, based upon a recurrence, are able to rewrite their organization (Self-programming machines or $\mathbf{m_{sp}}$) whenever external conditions vary and have striking properties of adaptation. $\mathbf{M_{sp}}$ have similar properties whatever the operation defining the recurrence maybe. These results bring us to make the following statement: adaptive properties of living systems can be explained by their ability to rewrite their organization whenever external conditions vary under the only assumption that the rewriting mechanism be a deterministic constant recurrence in a finite state set. ***To cite this article: J.-P. Moulin, C. R. Biologies 329 (2006).***

Keywords: Adaptive systems; Self-programming machines; Self-organizing machines; Self-reproducing machines; Ultra stable systems

1. Introduction

According to recent definitions [1], self-organization is a basic emergent behaviour. Plants and animals assemble and regulate themselves independent of any hierarchy for planning or management.

Emergence describes the way unpredictable patterns arise from innumerable interactions between independent parts.

Let us quote some historical main publications about this topic:

– feedback loops described in Cybernetics [2,3] which give one of the first models of the living beings' organization. Wiener himself [4] makes the paral-

E-mail address: j-paul.moulin@chi-clermont.fr (J.-P. Moulin).

1631-0691/$ – see front matter
doi:10.1016/j.crvi.2005.12.002

lelism between the cerebellar tremor and an unadjusted retroactive loop;
- the works [5,6] studying the entropy of self-organizing systems. Prigogine defined dissipative structures systems which continuously export entropy in order to maintain their organization [7];
- modelling the organization of living systems by dynamical systems. René Thom proposes catastrophe theory [8] as a model of embryological differentiation. This model uses continuous differentiable dynamical systems;
- automata theory that allows us to very easily represent interactions between parts of systems. Von Neumann self-reproducing automata [9] is the main example of machines building themselves. Stuart Kaufman models genetic regulatory networks with networks of combinatorial Boolean automata [10,11];
- Von Foerster and later Henri Atlan [12] try to explain the self-organizing properties of the living beings by the paradigm of the complexity by the noise;
- Maturana, Varela [13] and Zeleny [14] develop the concepts of the closure property of all the autonomous systems and autopoiesis.

Yet Ashby [15] argues that self-organization cannot be the result of a function because this function ought to be the result of self-organization and so forth.

On the other hand, computer science proposes to this date a lot of models to design adaptive systems, but all these models depend on the comparison between a goal and the results and ingenious choices of parameters by programmers, whereas there are no programmer's intention nor choice in the living systems. Let us mention some examples:

- for the simulated annealing model, choice of parameters: initial temperature, how many iterations are performed and how much the temperature is decremented at each step [16–18];
- in neural networks, back propagation of the difference between the target value and the output is used for learning the target [19,20];
- in evolutionary programming, each individual in the population receives a measure of its fitness to the environment [21,22];
- the adaptive filter, using the least mean-square algorithm, which is the most widely used adaptive filtering algorithm, measures the difference between the output of the adaptive filter and the output of the unknown system [23,24].

One cannot simulate with all these previous models the adaptive properties of the nervous system and the immune system such as those found in the following examples:

(1) if we cut the opposite muscles of a monkey and reattach them in a crossed position of one eyeball [25] or of the limb [26], after some weeks, the eyeballs again move together and the movements of the limb are co-ordinated;
(2) in an experiment of rats switching between two different environments (morph square and morph circle), it was shown that neurons states got stabilized into two different configurations [27];
(3) B cells can distinguish the change of a single amino acid in the epitope of an antigen [28] and elaborate new antibodies directed against the new epitope.

Nervous system and immune system are different [29]. The immune system comprises different types of cells and there are no equivalents of the Hodgkin, Huxley equations giving a model of the neuron [30,31].

Yet these systems have in common the property to be able to rewrite their organization whenever the external conditions vary.

All present theories of synaptic learning refer to the general rule introduced by the Canadian psychologist Donald Hebb in 1949 [32]. He postulated that repeated activation of one neuron [33] by another, across a particular synapse, increases its strength.

B cells make specific immuno globulins during their differentiation by recombination, splicing and removal of V, D and J gene segments [34].

These systems having 'the rewriting property' operate as self-organizing machines, which self-organize by modifying their internal functions, i.e. self-programming machines.

This problem was raised in genetics: "Molecular biology itself must acknowledge that there is a fundamental difficulty, since it is obliged to admit that the famous 'genetic program' is a 'program which programs itself' or a program which needs the result of its reading and execution so as to be read and executed" [35].

So through this very fast review of literature, we see that there are fundamental difference between the classical model approach and self-programming machines. Table 1 summarizes these differences.

We will define, in this paper, an Adaptive System as a system that changes its behaviour in response to its environment, by varying its own parameters or even, in our case, its own structure: it thus retains information about the environment in a form that can influence its fu-

Table 1
Differences between the classical model approach and self-programming machines

	Model	Self-programming machines
Implementation	Is a fixed function solution of differential system or a transition function of automata network	Is an organization modified by environmental data
Properties of the organization	Distribution of transient and cycles length. Diameter of attraction basins	Only one trajectory can be studied since the trajectory modifies the organization as it goes. Fixed points are pervasive
Properties of the cycle	Each point of a cycle is gone through only once	The self-programming machine can go through a point more than once during a cycle (in case of n-cycle) since the internal function will be different on the next occurrence
Probability to stabilize into a fixed point when the input set increases	Less probable [36]	Very probable

ture behaviour [37]. However, an Adaptive System does not have a stated or implicit goal or objective, and thus does not aim at improving its performance. Yet, adaptation results in a seemingly goal-directed behaviour, if the environment repeats patterns of inputs.

We will show that the mechanism proposed here and the adaptive systems of living beings share similar properties:

- all functions that compare results to a goal (feedback, gradient, fitness to the environment, etc.) and all parameters choices are not to be allowed;
- initial states and functions are randomly chosen at the beginning and once and for all;
- ability to stabilize into a fixed point if the environment is fixed, this stabilization being more probable when the input set size increases or the machines are interconnected;
- ability to find another stabilization (ultra stability) [38] if the conditions vary, in case of small perturbations or intentional massive breakdowns;
- ability to run in parallel, each machine having its local clocks, not studied in this paper.

2. Self-programming machines. Description and functioning

We define here self-programming machine formalism and show the properties of their dynamics.

2.1. Components

Let be given:

- $\frac{\mathbb{Z}}{p\mathbb{Z}}$, with p prime,
- $P_A = \{f_\alpha, \alpha \in \{0, \ldots, p^p - 1\} \mid f_\alpha : \frac{\mathbb{Z}}{p\mathbb{Z}} \mapsto \frac{\mathbb{Z}}{p\mathbb{Z}}\}$, the set of polynomials which map $\frac{\mathbb{Z}}{p\mathbb{Z}}$ into itself, $|P_A| = p^p$,
- $* : P_A \times P_A \mapsto P_A$,
- $\mathscr{C}_P = \{\mathscr{C} : \frac{\mathbb{Z}}{p\mathbb{Z}} \times \frac{\mathbb{Z}}{p\mathbb{Z}} \mapsto P_A\}$, a set of functions that map $\frac{\mathbb{Z}}{p\mathbb{Z}} \times \frac{\mathbb{Z}}{p\mathbb{Z}}$ into P_A,
- an index $\alpha \in \mathbb{N}$.

2.2. Definition and functioning

Definition. A self-programming machine (or $\mathbf{m_{sp}}$) is a sextuplet $(Inp, F, Out, S, \delta, R)$ where:

(1) Inp is the set of *inputs* of $\mathbf{m_{sp}}$: $Inp = \frac{\mathbb{Z}}{p\mathbb{Z}} \times \frac{\mathbb{Z}}{p\mathbb{Z}}$,
(2) $F = \{\mathscr{F} \mid \mathscr{F} : Inp \mapsto Out\}$,
(3) Out is the set of *outputs* of $\mathbf{m_{sp}}$: $Out \subseteq \frac{\mathbb{Z}}{p\mathbb{Z}} \times \frac{\mathbb{Z}}{p\mathbb{Z}}$,
(4) S is the set of *internal states* of $\mathbf{m_{sp}}$: $S = P_A \times Inp$,
(5) $\delta : S \mapsto S$ is the *transition function* of $\mathbf{m_{sp}}$ (change state), $S_\alpha = (f_\alpha, (e_\alpha, s_\alpha)) \overset{\delta}{\to} S_{\alpha+1} = (f_{\alpha+1}, (e_{\alpha+1}, s_{\alpha+1}))$,
(6) $(\mathscr{C}_\alpha)_{\alpha\in\mathbb{N}}$, a family of functions defined by a recursion R.

Functioning: $\mathbf{M_{sp}}$ performs the following operations (Fig. 1):

$$inp_\alpha = (e_\alpha, s_\alpha), \quad e_\alpha, s_\alpha \in \frac{\mathbb{Z}}{p\mathbb{Z}} \tag{1}$$

$$\mathscr{C}_\alpha(inp_\alpha) = f_\alpha \tag{2}$$

(1) and (2) define the state $S_\alpha = (\mathscr{C}_\alpha(inp_\alpha), inp_\alpha) = (f_\alpha, inp_\alpha)$

$$s_{\alpha+1} = f_\alpha(e_\alpha) = e_{\alpha+1} \tag{3}$$

$$out_\alpha = (e_{\alpha+1}, s_{\alpha+1}) = \big(f_\alpha(e_\alpha), f_\alpha(e_\alpha)\big) = inp_{\alpha+1} \tag{4}$$

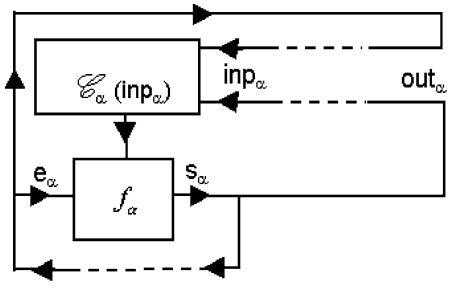

Fig. 1.

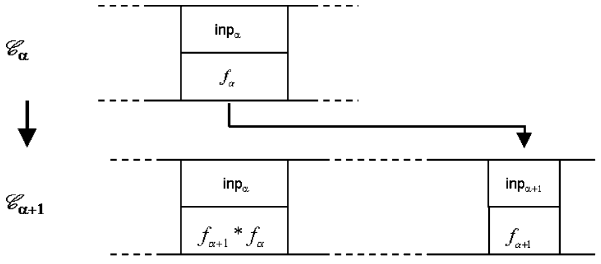

Fig. 2.

(4) defines

$$\mathscr{F}_\alpha(inp_\alpha) = out_\alpha = inp_{\alpha+1} \tag{5}$$

$$\mathscr{C}_\alpha(inp_{\alpha+1}) = f_{\alpha+1} \tag{6}$$

The equalities (5) and (6) define the transition function:

$$\begin{aligned}\delta(f_\alpha, inp_\alpha) &= \big(\mathscr{C}_\alpha(inp_{\alpha+1}), \mathscr{F}_\alpha(inp_\alpha)\big)\\ &= (f_{\alpha+1}, inp_{\alpha+1})\end{aligned} \tag{7}$$

The recursion $R : \mathscr{C}_\alpha \mapsto \mathscr{C}_{\alpha+1}$ is defined by:

$$\mathscr{C}_{\alpha+1}(input) = \begin{cases} \mathscr{C}_\alpha(\mathscr{F}_\alpha(input)) * f_\alpha & \text{if } input = inp_\alpha \quad (8)\\ \mathscr{C}_\alpha(input) & \text{if } input \neq inp_\alpha \quad (9)\end{cases}$$

A machine $\mathbf{m_{sp}}$ is thus entirely characterized by the dimensionality of its input space p and its operator $*$.

From (5) and (6) and (8), we obtain (Fig. 2):

$$\begin{aligned}\mathscr{C}_{\alpha+1}(inp_\alpha) &= \mathscr{C}_\alpha\big(\mathscr{F}_\alpha(inp_\alpha)\big) * f_\alpha = \mathscr{C}_\alpha(inp_{\alpha+1}) * f_\alpha\\ &= f_{\alpha+1} * f_\alpha\end{aligned} \tag{10}$$

2.3. Initialization and trajectory

Initial conditions of a machine $\mathbf{m_{sp}}$ $(p, *)$ are given by initial inputs inp_0 and initial function $\mathscr{C}_0$:

Initialization. $inp_0 = (e_0, s_0)$ and $\mathscr{C}_0$ are chosen at random at the beginning and once for all from two laws of probability defined respectively by its sample spaces $Inp = \frac{\mathbb{Z}}{p\mathbb{Z}} \times \frac{\mathbb{Z}}{p\mathbb{Z}}$ for inp_0 and P_A for $\mathscr{C}_0$ and its uniform distributions:

$$\wp\big(\{\alpha\}\big) = \frac{1}{p^2}, \quad \alpha \in \frac{\mathbb{Z}}{p\mathbb{Z}} \times \frac{\mathbb{Z}}{p\mathbb{Z}}$$

$$\wp\big(\{f\}\big) = \frac{1}{p^p}, \quad f \in P_A$$

Trajectory. The trajectory of $\mathbf{m_{sp}}$ corresponds to successive points $(inp_\alpha, out_\alpha) \in I \times O$ the machine goes through, starts at (inp_0, out_0) and will be denoted:

$$\begin{aligned}Trajectory = \big\{&(inp_0, out_0), \ldots, (inp_\alpha, out_\alpha),\\ &(inp_{\alpha+1}, out_{\alpha+1}), \ldots\big\}\end{aligned}$$

Theorem 1. *A self-programming machine is a deterministic device, and the cardinality of state set is finite. Therefore such a machine necessarily converges.*

Definitions. A $\mathbf{m_{sp}}$ *converges* whenever a point of its trajectory $(inp_{\tau+n}, out_{\tau+n})$ is identical to a previous one (inp_τ, oup_τ). It then indefinitely goes through a limit cycle or n-cycle composed of the n points:

$$\begin{aligned}&(inp_\tau, out_\tau), (inp_{\tau+1}, out_{\tau+1}), \ldots,\\ &\quad (inp_{\tau+n-1}, out_{\tau+n-1}).\end{aligned}$$

Thus dynamics of $\mathbf{m_{sp}}$ is characterized by the transient length τ and the length n of the limit cycle (i.e. its period).

2.4. Connection to external processes g and h: Machine $\mathbf{m_{sp1}}$ and $\mathbf{m_{sp2}}$

In order to show the $\mathbf{m_{sp}}$ adaptive properties:

(1) the $\mathbf{m_{sp}}$ is connected (Fig. 3) to a combinatorial automaton defined by some external function $g \in P_A$. Let us call this machine $\mathbf{m_{sp1}}$.

1. The equality (3) $e_{\alpha+1} = s_{\alpha+1}$ defining the $\mathbf{m_{sp}}$ is replaced by $e'_{\alpha+1} = g(s'_{\alpha+1})$ for the $\mathbf{m_{sp1}}$.

Let us study under which conditions a $\mathbf{m_{sp}}$ that we call $m'_{sp}$is identical to $\mathbf{m_{sp1}}$, i.e. conditions under which a $\mathbf{m_{sp1}}$ is a self-programming machine $\mathbf{m_{sp}}$.

The equality (2) $\mathscr{C}_\alpha(inp_\alpha) = f_\alpha$ defining $\mathbf{m_{sp}}$ is replaced by $\mathscr{C}'_\alpha(inp_\alpha) = g \circ f_\alpha$ for the m'_{sp}.

The equality (6) $\mathscr{C}_\alpha(inp_{\alpha+1}) = f_{\alpha+1}$ by $\mathscr{C}'_\alpha(inp_{\alpha+1}) = g \circ f_{\alpha+1}$.

The equality (10) $\mathscr{C}_{\alpha+1}(inp_\alpha) = \mathscr{C}_\alpha(\mathscr{F}_\alpha(inp_\alpha)) * f_\alpha = f_{\alpha+1} * f_\alpha$ by $\mathscr{C}'_{\alpha+1}(inp_\alpha) = \mathscr{C}'_\alpha(\mathscr{F}_\alpha(inp_\alpha)) * (g \circ f_\alpha) = \mathscr{C}'_\alpha(inp_{\alpha+1}) * (g \circ f_\alpha) = (g \circ f_{\alpha+1}) * (g \circ f_\alpha)$.

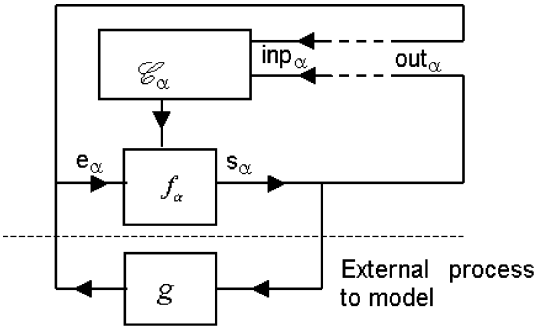

Fig. 3.

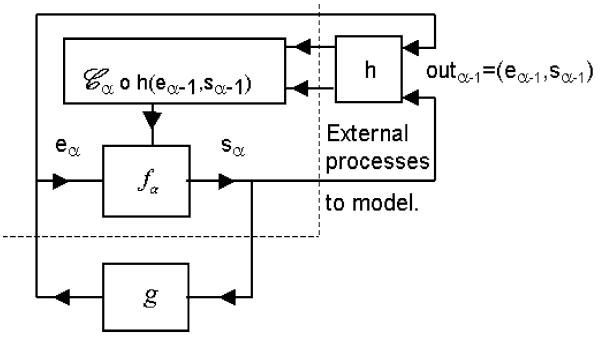

Fig. 4.

Operation '$\circ$' is left distributive with '$*$', so we can write:

$$\mathscr{C}'_{\alpha+1}(inp_\alpha) = g \circ (f_{\alpha+1} * f_\alpha) \tag{11}$$

In this case:

Theorem 2. *m'_{sp} and $\mathbf{m_{sp1}}$ are identical and hence have the same trajectory.*

2. $\mathbf{M_{sp1}}$ is connected (Fig. 4) to another combinatorial automaton defined by any external function $h \in P_A \times P_A$. Let us call this machine $\mathbf{m_{sp2}}$.

The equality (4) $inp_\alpha = out_{\alpha-1}$ defining the $\mathbf{m_{sp}}$ is replaced by $inp''_\alpha = h(out''_{\alpha-1})$ for the $\mathbf{m_{sp2}}$.

Let us study under which conditions a m_{sp1} that we call m''_{sp} identical to $\mathbf{m_{sp2}}$:

Then the equality (2): $\mathscr{C}'_\alpha(inp_\alpha) = g \circ f_\alpha$ defining m_{sp1} is replaced by $\mathscr{C}''_\alpha \circ h(out_{\alpha-1}) = g \circ f_\alpha$ for the machine m''_{sp}, the equality (6): $\mathscr{C}'_\alpha(inp_{\alpha+1}) = g \circ f_{\alpha+1}$ is replaced by $\mathscr{C}''_\alpha \circ h(out_\alpha) = g \circ f_{\alpha+1}$, and the equality (11) $\mathscr{C}'_{\alpha+1}(inp_\alpha) = g \circ (f_{\alpha+1} * f_\alpha)$ is replaced by $\mathscr{C}''_{\alpha+1} \circ h(out_{\alpha-1}) = g \circ (f_{\alpha+1} * f_\alpha)$.

The function h is injective. In this case:

Theorem 3. *m''_{sp} and $\mathbf{m_{sp2}}$ are identical and hence have the same trajectory.*

From now on, we will only study $\mathbf{m_{sp2}}$ and we call $\mathbf{m_{sp}}$ this machine.

3. Cyclic groups

Let us suppose a $\mathbf{m_{sp}}$ is going through a n-cycle points of which are: $\{(inp_\alpha, out_\alpha)\}_{\alpha\in I}$, $I = \{\tau, \tau+1, \ldots, \tau+n-1\}$.

Cyclic group of the states (G_S)

From the equality (7), we can write $S_\alpha = (f_\alpha, inp_\alpha)$, $\alpha \in I$, $\delta(S_\alpha) = (S_{\alpha+1})$, therefore $\exists s$ such as

$$\underbrace{\delta \circ \cdots \circ \partial}_{s \text{ times}}(S_\alpha) = (S_\alpha)$$

and $(\partial, \circ)$ generates a cyclic group G_S order of which is s.

Cyclic group of the n-cycles (G_n)

The function g being fixed at the beginning, one point (inp_α, out_α) of the cycle is only defined by the function f_α: $(inp_\alpha, out_\alpha) = ((e_\alpha, f_\alpha(e_\alpha)), (g \circ f_\alpha(e_\alpha), f_\alpha(e_\alpha))$. If we have a s-cycle for $(S_\alpha)_{\alpha\in I}$, we obtain an n-cycle for $((inp_\alpha, out_\alpha))_{\alpha\in I}$ with $n \leqslant s$ (hence $n \mid s$).

It therefore depends uniquely on the membership of f_α to the equivalence class of polynomials having the same image for the value e_α. Therefore the order n of G_n is a divisor of s.

4. Conditions of *n*-cycles

4.1. Linear operator

Definition. In the case of a linear operator $*$, equality (9) becomes: $\mathscr{C}_{\alpha+1}(inp_\alpha) = f_{\alpha+1} * f_\alpha = \lambda f_{\alpha+1} + \mu f_\alpha$. For the sake of simplicity, we will simplify this to $\mathscr{C}_{\alpha+1}(inp_\alpha) = \lambda f_{\alpha+1} + f_\alpha$, which not does influence the distribution of n-cycles or transient length.

$$\lambda \in \frac{\mathbb{Z}}{p\mathbb{Z}} - \{0\}$$

$$\mathscr{C}_{\alpha+1}(inp_\alpha) = \lambda \mathscr{C}_\alpha(inp_{\alpha+1}) + \mathscr{C}_\alpha(inp_\alpha) = \lambda f_{\alpha+1} + f_\alpha$$

Let m be the number ($m \leqslant n$) of different input values of the n-cycle. The matrix corresponding to transition from state $(f_{\tau+v}, inp_{\tau+v})$ to state

$(f_{\tau+v+1}, inp_{\tau+v+1})$ is the $(m \times m)$ matrix:

$$\begin{bmatrix} 1 & \dots & 0 & 0 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & \dots & 1 & \lambda & \dots & \dots \\ 0 & \dots & 0 & 1 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & \dots & 0 & 0 & \dots & 1 \end{bmatrix} \begin{bmatrix} f_\tau \\ \dots \\ f_{\tau+v} \\ f_{\tau+v+1} \\ \dots \\ f_{\tau+m-1} \end{bmatrix} = \begin{bmatrix} f_\tau \\ \dots \\ f_{\tau+v} + \lambda f_{\tau+v+1} \\ f_{\tau+v+1} \\ \dots \\ f_{\tau+m-1} \end{bmatrix} \quad (12)$$

The n transitions of the n-cycle give a product $[P]$ of n matrices such as:

$$[P] \begin{bmatrix} f_\tau(e_\tau) \\ \dots \\ f_{\tau+v}(e_{\tau+v}) \\ \dots \\ f_{\tau+m-1}(e_{\tau+m-1}) \end{bmatrix} = \begin{bmatrix} f_{\tau+n}(e_\tau) \\ \dots \\ f_{\tau+n+v}(e_{\tau+v}) \\ \dots \\ f_{\tau+n+m-1}(e_{\tau+m-1}) \end{bmatrix} = \begin{bmatrix} f_\tau(e_\tau) \\ \dots \\ f_{\tau+v}(e_{\tau+v}) \\ \dots \\ f_{\tau+m-1}(e_{\tau+m-1}) \end{bmatrix} \quad (13)$$

Equality (13) implies that the conditions for a $\mathbf{m_{sp}}$ to go through a n-cycle, defined as some vector $[f_\alpha(e_\alpha), \alpha \in \{\tau, \dots, \tau+m-1\}]$, is that this vector is an eigenvector associated to eigenvalue 1 of $[P]$.

Inversely, if we have a product of n $(m \times m)$ matrices $[P] = P_n.P_{n-1}.\cdots.P_2.P_1$ where $[P]$ has eigenvalue 1 and P_i are matrices such as in the left-hand side of Eq. (12), then we can find a linear $\mathbf{m_{sp}}$ such as these matrices are transition matrices of $\mathbf{m_{sp}}$ and $\mathbf{m_{spc}}$ goes through a n-cycle corresponding to the 1-eigenvector and to the n transition matrices P_i.

Theorem 4. *A* $\mathbf{m_{sp}}$ *with linear operator goes through a n-cycle* $[f_\alpha(e_\alpha), \alpha \in \{\tau, \dots, \tau+m-1\}]$ *if 1 is an eigenvalue of* $[P]$ *and* $[f_\alpha(e_\alpha), \alpha \in \{\tau, \dots, \tau+m-1\}]$ *its associated eigenvector.*

4.2. Polynomial operator

In preliminary unpublished work [39], we have studied the case: $\mathscr{C}_{\alpha+1}(inp_\alpha) = f_{\alpha+1} \circ f_\alpha$, where $\circ$ is the composition of functions, thus corresponding to the case where $* = \circ$. We found that, in this case, trajectories always ended in fixed points. We have introduced before the linear case, we now introduce a more general polynomial case.

Definition. A polynomial operator $*$ is defined, for some polynomial $f \in P_A$, by:

$$\mathscr{C}_{\alpha+1}(inp_\alpha) = f_{\alpha+1} * f_\alpha \quad (14)$$

$$\mathscr{C}_{\alpha+1}(inp_\alpha) = C_\alpha(inp_\alpha) + f \circ \mathscr{C}_\alpha(inp_{\alpha+1}) \quad f \in P_A \quad (15)$$

$$C_{\alpha+1}(inp_\alpha) = f_\alpha + f \circ f_{\alpha+1} \quad (16)$$

We do not have theoretical results in that case and can only give simulations results for such $\mathbf{m_{sp}}$: all theoretical results presented in the following Sections 5 and 6 only apply to linear machines. Yet, simulations show very similar results for both linear and polynomial operators (see results in §6.1).

5. Association of machines

5.1. Association with a combinatorial automaton

5.1.1. Definition

A combinatorial automaton defined by a function $p_{\alpha \in N} \in P_A$ is modified by a $\mathbf{m_{sp}}$ (Fig. 5).

The functioning of this machine ($\mathbf{m_{spc}}$) differs from a $\mathbf{m_{sp}}$ uniquely by the equality (3) replaced by $s_\alpha = p_\alpha(e_\alpha)$ with $p_\alpha = p_{\alpha-1} + f_{\alpha-1}$.

In this paper, we consider for simplicity that the functions h and g are identity functions (which implies $e_{\alpha+1} = s_\alpha$ and $inp_{\alpha+1} = out_\alpha$) and the operation '$*$' is linear. Apparently, in general case, we cannot obtain theoretical results.

5.1.2. Conditions of n-cycle

Theorem 5. *A* $\mathbf{m_{spc}}$ *goes through a n-cycle if* $\forall i, j \in I = \{\tau, \dots, \tau+n-1\}$

- $(f_i)(e_j) = 0$
- $p_i(e_i) \neq p_j(e_j)$

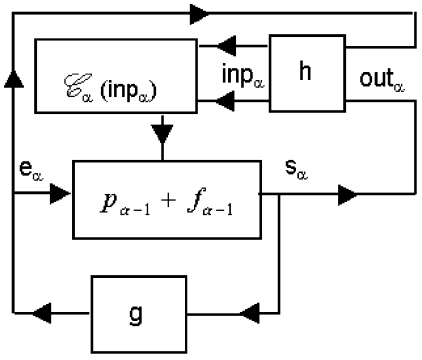

Fig. 5.

Let us suppose this machine goes through a n-cycle, i.e. $\alpha \geqslant \tau$.

We have:

$$\forall i \in \{0, \ldots, n-1\},$$
$$p_{\alpha+i}(e_{\alpha+i}) = p_{\alpha+n+i}(e_{\alpha+n+i}) = p_{\alpha+2n+i}(e_{\alpha+2n+i}) = \cdots \quad (17)$$

We make the demonstration for $i = 0$.

Let P be the matrix defined by the equality (13), let L_i be the line vector such as $L_i.P$ is the line vector equal to the ith line of P, let V the column vector elements of which are $f_\alpha(e_\alpha), f_{\alpha+1}(e_\alpha), \ldots, f_{\alpha+n-1}(e_\alpha)$, from the equality (17):

$p_\alpha(e_\alpha) = p_{\alpha+n}(e_\alpha)$ gives
$p_\alpha(e_\alpha) = p_\alpha(e_\alpha) + [L_1 \cdot P].V$
$p_\alpha(e_\alpha) = p_{\alpha+2n}(e_\alpha)$ gives
$p_\alpha(e_\alpha) = p_\alpha(e_\alpha) + \left[L_1 \cdot P^2 + L_1 \cdot P\right].V$
$\ldots$
$p_\alpha(e_\alpha) = p_{\alpha+n^2}(e_\alpha)$ gives
$p_\alpha(e_\alpha) = p_\alpha(e_\alpha) + \left[L_1 \cdot P^{n-1} + \cdots + L_1 \cdot P^2 + L_1 \cdot P\right].V$

which implies

$$(L_1 \cdot P) \cdot V = (L_1 \cdot P^2) \cdot V = \cdots = (L_1 \cdot P^{n-1}) \cdot V = 0 \quad (18)$$

These n equalities give the following homogenous system:

$$\begin{bmatrix} L_1 \cdot P \\ L_1 \cdot P^2 \\ \cdots \\ L_1 \cdot P^{n-1} \end{bmatrix} V = 0$$

One obtains by recurrence:

$$\begin{bmatrix} L_1 \cdot P \\ L_1 \cdot P^2 \\ \cdots \\ L_1 \cdot P^{n-1} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & \cdots \\ 1 & \lambda & 0 & 0 & 0 & \cdots \\ 1 & 2\lambda & \lambda^2 & 0 & 0 & \cdots \\ 1 & 3\lambda & 3\lambda^2 & \lambda^3 & 0 & \cdots \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 1 & \binom{n-1}{1}\lambda & \binom{n-1}{2}\lambda^2 & \cdots & \cdots & \lambda^{n-1} \end{bmatrix} \quad (19)$$

the determinant of which is equal to $\prod_{i=0}^{n-1} \lambda^i = \lambda^{n(n-1)/2} \neq 0$, therefore the only solution of this system is the trivial solution: $f_\alpha(e_\alpha) = f_{\alpha+1}(e_\alpha) = \cdots = f_{\alpha+n-1}(e_\alpha) = 0$.

Repeating the same reasoning for $i = 1, \ldots, n-1$, the determinant of the system is equal to $\pm\lambda^{n(n-1)/2} \neq 0$ and we obtain

$$(f_i)(e_j) = 0 \quad (20)$$

The equalities $(f_i)(e_j) = 0$ and the equality $e_{\alpha+1} = [p_{\alpha-1} + f_{\alpha-1}](e_\alpha)$ implies that

$$\forall i, j \geqslant \tau, \quad p_i(e_i) \neq p_j(e_j) \quad (21)$$

The equality $p_i(e_i) = p_j(e_j)$ would imply a cycle of length $n' < n$ contrary to the hypothesis (21).

In addition,

$$p(e_{\tau+n-1}) = e_{\tau+n} = e_\tau \quad (22)$$

5.1.3. Distribution of τ and n

Let be given identical machines $\mathbf{m_{spc}}$ differing uniquely by their initial conditions.

Remark. If $\wp(\{e\}) = \frac{1}{p}$, $e \in \frac{\mathbb{Z}}{p\mathbb{Z}}$, a sequence with elements chosen at random $((e_\alpha, e_\alpha), (e_{\alpha+1}, e_{\alpha+1}), \ldots, (e_{\alpha+n-1}, e_{\alpha+n-1}))$ can be a trajectory of some $\mathbf{m_{spc}}$ if all the values $e_{i\in\alpha,\ldots,\alpha+n-1}$ are different: $P_{\alpha-1}$ and e_α being given, we have only to choose a function $f_{\alpha-1}$ that satisfies the equality $f_{\alpha-1}(e_\alpha) = e_{\alpha+1} - P_{\alpha-1}(e_\alpha)$. We make the same reasoning for the following points of the sequence.

But, if two points in the sequence are equal (for example, (e_α, e_α) and $(e_{\alpha+l}, e_{\alpha+l})$), if the value $e_{\alpha+l+1}$ is such that $e_{\alpha+l+1} \neq P_{\alpha+l}(e_\alpha) + [f_\alpha + \lambda f_{\alpha+1}](e_\alpha)$, the sequence cannot be the trajectory of a $\mathbf{m_{spc}}$.

The probability that an arbitrary sequence has all its values different is:

$$\frac{p-1}{p}\frac{p-2}{p}\cdots\frac{p-n}{p} = \frac{(p-1)!}{(p-n-1)!}\frac{1}{p^n},$$

if n is fixed, this probability tends towards 1 when $p \to \infty$, therefore an arbitrary sequence chosen at random is almost always a trajectory of a $\mathbf{m_{spc}}$ if $p \gg n$.

Lemma. $\wp[f_i(e_j) = x] = \frac{1}{p}$.

The following matrix given in Fig. 6 establishes a one-to-one correspondence between the coefficients of polynomial $f \in P_A$ and the values of this polynomial.

The determinant of this matrix is a Vandermonde determinant equal to $(p-1)!(p-2)!\cdots 2!1!$ and different from 0 since p is prime.

The number of functions $f \in P_A$ is p^p, each function has p coefficients, the number of coefficients of the

$$\begin{bmatrix} 0 & 0 & 0 & \dots & 0 & 1 \\ 1 & 1 & 1 & \dots & 1 & 1 \\ 2^{p-1} & 2^{p-2} & 2^{p-3} & \dots & 2^1 & 2^0 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ (p-2)^{p-1} & (p-2)^{p-2} & (p-2)^{p-3} & \dots & (p-2)^1 & (p-2)^0 \\ (p-1)^{p-1} & (p-1)^{p-2} & (p-1)^{p-3} & \dots & (p-1)^1 & (p-1)^0 \end{bmatrix} \begin{bmatrix} a_{p-1} \\ a_{p-2} \\ a_{p-3} \\ \dots \\ a_1 \\ a_0 \end{bmatrix} = \begin{bmatrix} f(0) \\ f(1) \\ f(2) \\ \dots \\ f(p-2) \\ f(p-1) \end{bmatrix}$$

Fig. 6.

functions of P_A is p^{p+1} and $\forall x \in \frac{\mathbb{Z}}{p\mathbb{Z}}$, p^p coefficients are equal to x.

The fact that p^p coefficients are equal to x and the correspondence one to one between the coefficients of polynomials P_A and the values of these polynomials imply that it exists p^p pairs (i, j) such as $f_i(e_j) = x$.

Therefore, if the random variable f_i and e_j have respectively its sample space equal to P_A and $\frac{\mathbb{Z}}{p\mathbb{Z}}$, and have uniform distributions:

$$\wp\big[f_i(e_j) = x\big] = \frac{p^p}{p^{p+1}} = \frac{1}{p}.$$

In the following, we try to find a way to calculate the probability of the event that a $\mathbf{m_{spc}}$ goes through a n-cycle. For that, we should work on the space of all the possible trajectories of $\mathbf{m_{spc}}$ differing by its initial conditions, but it is not a finite space. To get rid of this problem, for fixed n and $p \gg n$, we consider just the set T_n of all the n-sequence of elements $T_n = \{((e_1, e_1), \dots, (e_n, e_n)) \mid e_i \in \frac{\mathbb{Z}}{p\mathbb{Z}}\}$ which is a finite set. In this space, we are looking for the n-cycles. I mean to find a $\mathbf{m_{spc}}$ that goes through these n-elements like a n-cycle. From now on, for n fixed and $p \gg n$, as shows our remark, we consider that all elements of T_n are on a trajectory of a $\mathbf{m_{spc}}$ which allows us to give the following results:

Let us consider the sequence which elements are chosen at random $((e_\alpha, e_\alpha), (e_{\alpha+1}, e_{\alpha+1}), \dots, (e_{\alpha+n-1}, e_{\alpha+n-1}))$. This sequence corresponds to a n-cycle of $\mathbf{m_{spc}}$ if the three events $f_i(e_j) = 0$ (20), $p(e_i) \neq p(e_j)$ (21) and $p(e_{\tau+n-1}) = e_\tau$ (22) occur:

$$\wp\big[f_\alpha(e_\alpha) = 0\big] = \frac{1}{p} \quad \text{implies}$$

$$\wp\big[f_i(e_j) = 0\big] = \frac{1}{p^{n^2}}, \quad i, j \in I,\ |I| = n.$$

We have demonstrated that [35]:

$$\wp\big[\big[p(e_i) \neq p(e_j)\big] \text{ and } \big[p(e_{\tau+n-1}) = e_\tau\big]\big] = \frac{p!}{(n-p)!} \cdot \frac{n}{p^{n+1}}$$

The probability that a point in the trajectory of this machine belongs to a n-cycle is the probability that a point belongs to a cycle is:

$$\begin{aligned} \wp[E_{n\text{-cycle}}] &= \frac{1}{n}\left[\frac{1}{p^{n^2}}\right] \cdot \left[\frac{p!}{(n-p)!} \cdot \frac{n}{p^{n+1}}\right] \\ &= \wp[E_{1\text{-cycle}}] + \wp[E_{2\text{-cycle}}] \\ &\quad + \wp[E_{2\text{-cycle}}] + \cdots \\ &= \frac{1}{p^2} + \frac{p-1}{4p^6} + \frac{(p-1)(p-2)}{9p^{12}} + \cdots \end{aligned}$$

Now, let us evaluate the probability for a point to be a transient point.

Theorem 6. *Under the assumption that the point is in the trajectory of some machine, then obviously*:

$$\wp[\textit{transient-point}] \simeq 1 - \frac{1}{p^2}$$

and the probability a $\mathbf{m_{spc}}$ *converges after* τ *transient points (the* $(\tau+1)$*th point is on a cycle) is* $\wp(\tau) \simeq (1 - \frac{1}{p^2})^\tau \frac{1}{p^2}$.

τ is a geometric random variable that has expected value $E(\tau) = p^2$ and variance

$$E(\tau^2) - \big[E(\tau)\big]^2 = 2p^4 - p^2$$

The fast convergence of $\mathbf{m_{spc}}$ explains the concordance between the computations and the simulations (Tables 3 and 4).

Because of the remark in §5.1.3, notice that for m small enough so that we can find $m-1$ different points, these $m-1$ points together with one mth point will be on a trajectory of some $\mathbf{m_{spc}}$ machine. Our assumption is thus very light.

5.2. Association with another m_{sp}

Fig. 7a and b shows the two ways of binding together two m_{sp}.

In this paper, we focus on a sequential updating scheme: machines change states one at a time. Other updating regimes (parallel, asynchronous) could also be studied but will not be reported here.

The method to demonstrate the equivalence between one $\mathbf{m_{sp}}$ and a network of n $\mathbf{m_{sp}}$ consists in replacing each term defining one $\mathbf{m_{sp}}$ by a n-tuplet.

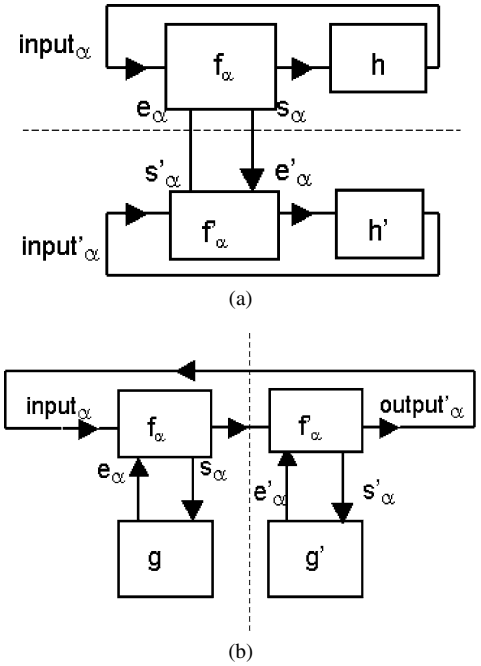

Fig. 7.

For example, in the case of Fig. 7a (Fig. 7b would be similar), equations can be rewritten as:

$$inpn_\alpha = (inp_\alpha, inp'_\alpha) = \big((e_\alpha, s_\alpha), (s_\alpha, e_\alpha)\big) \tag{23}$$

$$\mathscr{C}n_\alpha(inp_\alpha) = \big(\mathscr{C}_\alpha(inp_\alpha), \mathscr{C}'_\alpha(inp'_\alpha)\big) = (f_\alpha, f'_\alpha) \tag{24}$$

$$outn_\alpha = \big(\big(f'_\alpha(s_\alpha), f_\alpha(e_\alpha)\big), \big(f_\alpha(e_\alpha), f'_\alpha(s_\alpha)\big)\big) \tag{25}$$

$$inpn_{\alpha+1} = \big(h(inp_\alpha), h'(inp'_\alpha)\big) \tag{26}$$

$$\mathscr{C}n_{\alpha+1}(inpn_{\alpha+1}) = (f_{\alpha+1} * f_\alpha, f'_{\alpha+1} * f'_\alpha) \tag{27}$$

which shows the equivalence.

6. Results

We now present the results of simulations we have run for m_{sp} machines, linear and polynomial. Each result is obtained by 32 000 simulations of the same machine with different initial conditions chosen at random. $|Input| = p^2$. For each value of p, the machine is chosen at random, i.e. parameter λ for the linear case and polynomial $f \in P_A$ are chosen at random.

6.1. $\mathbf{M_{sp}}$ *with operation '$*$' linear and polynomial*

The similarity between $\mathbf{m_{sp}}$ with operations '$*$' linear and polynomial is striking: we show in Table 2 results of simulations, both for the linear and polynomial cases, indicating that both linear and polynomial cases seem to be in agreement with the theoretical value.

Table 2
Results of simulations for linear and polynomial cases

$\lvert Input\rvert$	'$*$' linear		'$*$' polynomial	
	% n-cycle $n \neq 1$	Expected value of τ	% n-cycle $n \neq 1$	Expected value of τ
9	3.28	8.28	3.01	11.82
25	2.63	85.50	1.84	32.92
49	0.55	188.99	0.21	149.46
121	0.052	299.98	0.056	302.34

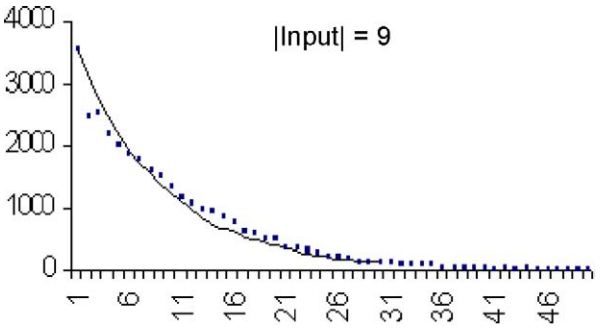

Fig. 8.

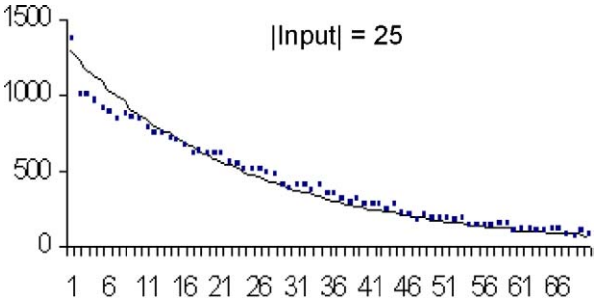

Fig. 9.

6.2. $\mathbf{M_{sp}}$ *associated with a combinatorial automaton*

6.2.1. *Transient length*

Each figure (Figs. 8–11) compares simulations (points) and computation (theoretical value) from $\wp(\tau) \simeq (1 - \frac{1}{p^2})^\tau \frac{1}{p^2}$ (smooth curve) for a given value of $|Input|$ and for 32 000 simulations, τ is on the X axis, the number of initial conditions on the Y axis.

In Table 3, we compare the mean and variance of τ for different values of $|Input|$:

6.2.2. *Frequency of cycles of length n*

Table 4 shows the results of simulations and comparison to the theoretical result obtained in §6.1 for $p_1 = \wp(E_{1\text{-cycle}})$, $p_2 = \wp(E_{2\text{-cycle}})$. The results show that fixed points (1-cycles) are the most common cycles, and increasingly so with increasing p. Again simulation

Table 3
Mean and variance of τ for different values of $|Input|$

$\|Input\|$	Expected value of τ: $E(\tau)$			Variance of τ: $E(\tau^2) - [E(\tau)]^2$		
	Simulations (linear)	Theoretical value	Simulations (polynomial)	Simulations (linear)	Theoretical value	Simulations (polynomial)
9	9.5	9	8.2	68.8	72	57.5
25	26.6	25	32.9	621.3	600	183.3
49	51.8	49	60.7	2467.6	2352	1214.72
121	125.2	121	132.4	14974.1	14520	5528.4

Table 4
Results of simulations and comparison to the theoretical result for $p_1 = \wp(E_{1\text{-cycle}})$, $p_2 = \wp(E_{2\text{-cycle}})$

T	$\|Input\| = 9$	$\|Input\| = 25$	$\|Input\| = 49$	$\|Input\| = 121$
Nb of 1-cycles	31,310	31,809	31,916	31,977
Nb of 2-cycles	686	191	84	23
Nb of 3-cycles	4	0	0	0
Frequency of n-cycle $n > 1$	2.15%	0.59%	0.26%	0.07%
p_1/p_2 (simulation)	45.6	166.5	379.9	1390.3
p_1/p_2 (theoretical value)	40.5	156.2	400.1	1464.1

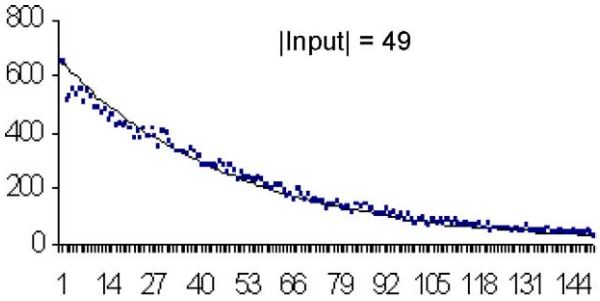

Fig. 10.

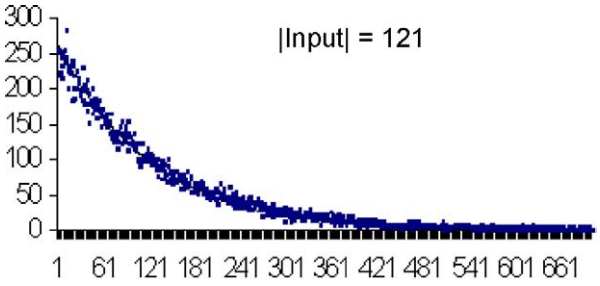

Fig. 11.

results are in agreement with the expected theoretical value.

7. Conclusion

We have shown in the two well-known examples of adaptive systems in living beings, the nervous system and the immune system that they share a common property: External signals modify the rewriting of their organization and therefore work as stimuli to modify the 'program' of the system.

To get better understanding of this very common feature in the living beings, we have devised the concept of **Self-programming machine**s or $\mathbf{m_{sp}}$.

These machines have a finite set of inputs, are based on a recurrence and are able to rewrite their internal organization whenever external conditions vary. They have striking adaptation properties:

- they stabilize almost exclusively into 1-cycles;
- the proportion of 1-cycles to all possible n-cycles goes to 1 as the number of inputs increases (i.e. 1-cycles are almost certain);
- they have similar properties whatever the operation defining the recurrence may be (linear or polynomial);
- they will stabilize again in case of small perturbations or intentional massive breakdowns and have proportion of 1-cycles to all possible n-cycles closer to 1 when they are connected in networks.

These results bring us to make the following statement: adaptive properties of living systems can be explained by their ability to rewrite their internal organization whenever external conditions vary under the only assumption that the rewriting mechanism be a deterministic constant recurrence in a finite state set.

References

[1] C. Meyer, Living machines, The New Facts of Life, Wired Magazine, February 2004.

[2] H. Von Foerster, Cybernetics in circular causal and feedback mechanisms, in: H. Von Foerster, M. Mead, H. Teuber, L.J. Macy (Eds.), Biological and Social Systems, Transactions of the Sixth Conference, 24–25 March 1949, Josiah Macy Jr. Fundation, New York, 1949, p. 1.

[3] D. Stanley Jones, K. Stanley Jones, The Kybernetics of Natural Systems, Pergamon Press, Oxford, UK, 1960.

[4] N. Wiener, Cybernetics, Hermann, Paris, 1948.

[5] P. Glanssdorf, I. Prigogine, Structures, Stabilité et Fluctuations, Masson, Paris, 1971.

[6] M. Eigen, Self-organization of matter and evolution of biological macromolecules, Die Naturwissenschaft 58 (1971) 463–520.

[7] G. Nicolis, I. Prigogine, Self-Organization in Nonequilibrium Systems (From Dissipative Structures to Order Through Fluctuations), John Wiley and Sons, New York, 1977.

[8] R. Thom, Catastrophe Theory, its Present State and Future Perspectives, Dynamical Systems, Lecture Notes in Mathematics, vol. 468, Springer-Verlag, Warwick, 1974.

[9] J. von Neumann, A.W. Burks, Theory of Self-Reproducing Automata, University of Illinois Press, Urbana, 1966.

[10] S. Kaufmann, Behavior of randomly constructed genetics nets, in: C.H. Waddington (Ed.), Toward a Theoretical Biology, vol. 3, Edinburgh University Press, Edinburgh, UK, 1970.

[11] S. Kaufmann, Metabolic stability and epigenesis, in: Randomly Constructed Genetic Nets, Theor. Biol. 22 (1969).

[12] H. Atlan, L'Organisation biologique et la Théorie de l'information, Hermann, Paris, 1970.

[13] F. Varela, Principles of Biological Autonomy, Elsevier/North-Holland, New York/Oxford, 1979.

[14] M. Zeleny (Ed.), Autopoiesis, A Theory of Living Organization, vol. 3, Elsevier/North-Holland, New York/Oxford, 1981.

[15] W.R. Ashby, Principles of self-organizing systems, in: Proc. Western Joint Computer Conf., 1961, pp. 255–278.

[16] B.P. Flannery, S.A. Teukolsky, W.T. Vetterling, in: Numerical Recipes, Cambridge University Press, New York, 1986, pp. 326–334.

[17] S. Kirkpatrick, C.D. Gelatt, M.P. Vecchi, Optimization by simulated annealing, Science 220 (4598) (1983) 671–680.

[18] S. Geman, D. Geman, Stochastic relaxation, Gibbs distribution and the Bayesian restoration of images, IEEE Trans. PAMI 6 (1984) 721–741.

[19] D.D. Rumelhart, G.E. Hinton, R.J. William, R.J. Learning, Representations by back-propagating errors, Nature 323 (1986) 533–536.

[20] P. Gallinari, S. Thiria, F. Badran, F. Fogelman-Soulié, On the relations between discriminant analysis and multilayer perceptrons, Neural Networks 4 (1991) 349–360.

[21] L.J. Fogel, P.J. Angeline, D.B. Fogel, An evolutionary programming approach to self-adaptation on finite states machines, Evol. Program. (1995) 355–365.

[22] R. Forsyth, A Darwinian approach to pattern recognition, Kibernetes 10 (1981) 159–166.

[23] S. Haykin, Adaptive Filter Theory, third ed., Prentice-Hall, Neurol. Zentralbl. 34 (1996) 338.

[24] A. Benvéniste, M. Goursay, G. Ruget, Robust identification of a non-minimum phase system: Blind adjustement of a linear equalizer in data communications, IEEE Trans. Autom. Control 3 (1996) 385–399.

[25] A. Marina, Die Relationnen des Palaeencephalons sind nicht fix, Neurol. Zentralbl. 34 (1915) 338.

[26] R.W. Sperry, Effects of crossing nerves to antagonistic limbs muscles in the monkey, Arch. Neurol. Psychiatry 58 (1947) 542.

[27] T.J. Wills, C. Lever, F. Cacucci, N. Burgess, J. O'Keefe, Attractor dynamics in the hippocampal representation of the local environment, Science 308 (5723) (2005) 873–876.

[28] H. Eisen, Immunology, Harperand Row, New York, 1980.

[29] D. Dasgupta, Artificial Neural Networks and Artificial Immune Systems: Similarities and Differences, in: Proc. Int. Conf. on Systems, Man and Cybernetics, Orlando, FL, USA, 12–15 October 1997.

[30] A. Perelson, G. Weisbuch, Immunology for physicists, Rev. Mod. Phys. 69 (4) (1997).

[31] L. Hodgkin, A.F. Huxley, A quantitative description of membrane current and its application to conduction and excitation in nerve, J. Physiol. 117 (1952) 500–544.

[32] D.O. Hebb, The Organization of Behavior, A Neuro-Psychological Theory, Wiley, New York, 1949.

[33] W.S. McCulloch, W. Pitts, in: A Logical Calculus of the Ideas Immanent in Nervous Activity, in: Bulletin of Mathematical Biophysics, vol. 5, 1943, pp. 115–133.

[34] K. Abbas, A.H. Lichtman, Basic Immunology, Functions and Disorders of the Immune System, Saunders imprint of Elsevier, 2004.

[35] P. Dumouchel, J.-P. Dupuy, in: Colloque de Cerisy, L'auto-organisation, Seuil, Paris, 1983.

[36] J.-P. Moulin, Modifiable automata, Self-modifying automata, Acta Biotheor. 40 (2/3) (1992) 195–204.

[37] Y. Bar-Yam, Dynamics of Complex Systems, Addison-Wesley, 1997.

[38] W.R. Ashby, Design for a Brain, Chapman and Hall, London, 1952.

[39] J.-P. Moulin, Self-Programming Machines I, Unpublished, 2000.