

# Introduction to the Natural Anticipator and the Artificial Anticipator

Daniel M. Dubois

*HEC Management School - University of Liege, N1,  
rue Louvrex 14, B-4000 Liège, Belgium  
Daniel.Dubois@ulg.ac.be - <http://www.sia.hec.ulg.ac.be>*

*and  
Centre for Hyperincursion and Anticipation in Ordered Systems,  
CHAOS ASBL, Institute of Mathematics, B37,  
Grande Traverse 12, B-4000 LIEGE 1, Belgium  
<http://www.ulg.ac.be/mathgen/CHAOS>*

**Abstract.** This short communication deals with the introduction of the concept of anticipator, which is one who anticipates, in the framework of computing anticipatory systems. The definition of anticipation deals with the concept of program. Indeed, the word program, comes from "pro-gram" meaning "to write before" by anticipation, and means a plan for the programming of a mechanism, or a sequence of coded instructions that can be inserted into a mechanism, or a sequence of coded instructions, as genes or behavioural responses, that is part of an organism. Any natural or artificial programs are thus related to anticipatory rewriting systems, as shown in this paper. All the cells in the body, and the neurons in the brain, are programmed by the anticipatory genetic code, DNA, in a low-level language with four signs. The programs in computers are also computing anticipatory systems. It will be shown, at one hand, that the genetic code DNA is a natural anticipator. As demonstrated by Nobel laureate McClintock [8], genomes are programmed. The fundamental program deals with the DNA genetic code. The properties of the DNA consist in self-replication and self-modification. The self-replicating process leads to reproduction of the species, while the self-modifying process leads to new species or evolution and adaptation in existing ones. The genetic code DNA keeps its instructions in memory in the DNA coding molecule. The genetic code DNA is a rewriting system, from DNA coding to DNA template molecule. The DNA template molecule is a rewriting system to the Messenger RNA molecule. The information is not destroyed during the execution of the rewriting program. On the other hand, it will be demonstrated that Turing machine is an artificial anticipator. The Turing machine is a rewriting system. The head reads and writes, modifying the content of the tape. The information is destroyed during the execution of the program. This is an irreversible process. The input data are lost.

**Keywords:** anticipator, anticipation, program, genetic code, Turing machine, rewriting system

**PACS:** 82.39.Pj, 87.14.gk, 87.14.gn, 89.20.Ff

## 1. INTRODUCTION

This introduction deals with the definition of concept of anticipator and the comparison between a natural anticipator and an artificial anticipator.

In his paper, THE WISHFUL ANTICIPATOR [7], presented at the first International Conference, CASYS'97, on Computing Anticipatory Systems, Charles François wrote in the abstract:

*Prophecy is a very old dream of mankind: the desire to know the future and, if possible, to twist the course of future events ...*

*There are however many limits to anticipation, and subsequent action. They seem to be basically of two classes:*

- *those inherent to the deep nature of natural evolution of all kinds of situations*
- *those related to our own limitations as observing systems, i.e. as perceivers, learners, modellers and actors.*

Daniel M. Dubois, "Introduction to the Natural Anticipator and the Artificial Anticipator".

In: COMPUTING ANTICIPATORY SYSTEMS: CASYS'09 - Ninth International Conference, edited by D.M. Dubois, American Institute of Physics, AIP CP 1303, pp. 17-24, 2010.

This topic is related to computing anticipatory systems for which the concepts of incursion and hyperincursion were defined [2, 3, 4], in relation to weak and strong anticipations [5].

After the dictionary, the noun, *anticipator, is one who anticipates, and its synonymous is the word, anticipant.*

In a first natural category, the anticipation is embedded in a natural anticipator, and in a second artificial category, the anticipation is programmed in an artificial anticipator.

For an artificial anticipator, it is possible to define weak and strong anticipations [5], because man is the designer of the artificial systems. For a natural anticipator, man is only able to construct a model, which will be always a weak representation of this.

A single artificial agent is not able to become intelligent, even in its environment, because it is not able to use a language for dialoguing, so intelligence can only emerge from a set of agents, what is called a multi-agent system, like a society.

Any intelligence can only be understood with the help of a language.

It exists two types of language, the natural and the artificial languages.

The artificial languages are related to formal logic, and are used in the Turing Machine.

Will a Turing Machine be able to understand natural language?

This is an open question [6].

In my book [1], a section is devoted to the question:

Is the consciousness important for the development of intelligence?

The evolution of the life and species was performed by the Darwinian natural selection. No conscious intelligent entity was at the origin of the creation of conscious intelligent species, as humans. During the evolution, the biological primitive living entities evolved firstly to an intelligence endowed by unconsciousness. The Darwinian Natural Selection is a necessary condition for explaining the evolution of life on earth. But, is it a sufficient condition to explain the emergence of intelligence, language, and consciousness?

This paper deals with an introduction to the capacity of anticipation of natural and artificial systems, what are called natural and artificial anticipators.

The definition of anticipation deals with the concept of program. A program executes a sequence of coded instructions by low-level and high-level languages.

After the dictionary, the word "program" comes from "pro-gram" meaning "to write before" by anticipation, and means a plan for the programming of a mechanism, or a sequence of coded instructions that can be inserted into a mechanism, or a sequence of coded instructions, as genes or behavioural responses, that is part of an organism.

Any natural or artificial programs are thus related to anticipatory rewriting systems, as we will show in this paper.

Recall that all the cells in the body, and the neurons in the brain, are programmed by the anticipatory genetic code, DNA, in a low-level language with four signs. The human cortex is the seat of the natural language, like French, and English languages, which are high-level natural languages.

In computers, there are also low-level and high-level languages. High-levels languages, like C or Pascal languages, are translated in a low-level language that is the machine code.

This paper will show, firstly, that the genetic code DNA is a natural anticipator, and secondly, that the Turing machine is an artificial anticipator.

These DNA and Turing machine are both anticipators, which are based on anticipatory rewriting systems, given by non-algorithmic or algorithmic programs.

## **2. THE GENETIC CODE DNA AS A NATURAL ANTICIPATOR**

DNA, Deoxyribonucleic acid, is a nucleic acid that contains the genetic instructions used in the development and functioning of all known living organisms, after the discover of the double helix by Watson and Crick [10].

The main role of DNA molecules is the long-term storage of information. DNA contains the instructions needed to construct other components of cells, such as proteins and RNA molecules.

The DNA segments that carry this genetic information are called genes, but other DNA sequences have structural purposes, or are involved in regulating the use of this genetic information.

The genetic code consists of 64 triplets of nucleotides, called codons, given in Table 1.

With three exceptions, each codon encodes for one of the 20 amino acids used in the synthesis of proteins.

Despite the minor variations that exist, the genetic code used by all known forms of life is nearly universal.

Daniel M. Dubois, " Introduction to the Natural Anticipator and the Artificial Anticipator".

In: COMPUTING ANTICIPATORY SYSTEMS: CASYS'09 - Ninth International Conference, edited by D.M. Dubois, American Institute of Physics, AIP CP 1303, pp. 17-24, 2010.

**TABLE 1.** The genetic code, with 4 bases T (thymidine), C (cytosine), A (adenosine), G (guanosine), consists of 64 codons. The letter in parentheses is the single-letter code of each amino acid.

	<b>T</b>	<b>C</b>	<b>A</b>	<b>G</b>
<b>T</b>	TTT Phenylalanine (F)	TCT Serine (S)	TAT Tyrosine (Y)	TGT Cysteine (C)
	TTC "	TCC "	TAC "	TGC "
	TTA Leucine (L)	TCA "	TAA <b>Terminator</b>	TGA <b>Terminator</b>
	TTG "	TCG "	TAG <b>Terminator</b>	TGG Tryptophan (W)
<b>C</b>	CTT Leucine (L)	CCT Proline (P)	CAT Histidine (H)	CGT Arginine (R)
	CTC "	CCC "	CAC "	CGC "
	CTA "	CCA "	CAA Glutamine (Q)	CGA "
	CTG "	CCG "	CAG "	CGG "
<b>A</b>	ATT Isoleucine (I)	ACT Threonine (T)	AAT Asparagine (N)	AGT Serine (S)
	ATC "	ACC "	AAC "	AGC "
	ATA "	ACA "	AAA Lysine (K)	AGA Arginine (R)
	ATG <b>Methionine (M)</b>	ACG "	AAG "	AGG "
<b>G</b>	GTT Valine (V)	GCT Alanine (A)	GAT Aspartic acid (D)	GGT Glycine (G)
	GTC "	GCC "	GAC "	GGC "
	GTA "	GCA "	GAA Glutamine (E)	GGA "
	GTG "	GCG "	GAG "	GGG "

NB: Methionine, is the START signal of translation. Terminator, is the STOP signal at the end of the gene.

DNA is a two-stranded molecule. An example is given in Table 2.

**TABLE 2.** The Double Helix and the Messenger RNA

<b>Double Helix</b>															
DNA coding (or sense) strand, with codons	<b>T</b>	<b>T</b>	<b>C</b>	<b>T</b>	<b>C</b>	<b>G</b>	<b>C</b>	<b>T</b>	<b>C</b>	<b>A</b>	<b>T</b>	<b>G</b>	<b>G</b>	<b>A</b>	<b>A</b>
DNA template (or antisense) strand, with anticodons	<b>A</b>	<b>A</b>	<b>G</b>	<b>A</b>	<b>G</b>	<b>C</b>	<b>G</b>	<b>A</b>	<b>G</b>	<b>T</b>	<b>A</b>	<b>C</b>	<b>C</b>	<b>T</b>	<b>T</b>
<b>Messenger RNA (mRNA)</b>															
mRNA made from template strand	<b>U</b>	<b>U</b>	<b>C</b>	<b>U</b>	<b>C</b>	<b>G</b>	<b>C</b>	<b>U</b>	<b>C</b>	<b>A</b>	<b>U</b>	<b>G</b>	<b>G</b>	<b>A</b>	<b>A</b>

Each strand is a polynucleotide composed of A (adenosine), T (thymidine), C (cytosine), and G (guanosine) residues polymerized by dehydration synthesis in linear chains with specific sequences.

At each nucleotide residue along the double-stranded DNA molecule, the nucleotides are complementary, where A forms two hydrogen-bonds with T, and C forms three hydrogen bonds with G.

In most cases, the two-stranded, antiparallel, complementary DNA molecule folds to form a Double Helix.

One strand of DNA, holding the information that codes for various genes, is called the template or antisense strand (containing anticodons). The complementary strand is called the coding or sense strand (containing codons).

Messenger RNA (mRNA) is a molecule of RNA, Ribonucleic acid, which is transcribed from a DNA template, and carries coding information to the sites of protein synthesis, the ribosomes. So, mRNA has the same information as the coding strand, and the code for the mRNA would be identical but for the fact that RNA contains U (uridine) rather than T.

The category of program given by the genetic code that is involved in developmental living systems belongs to non-algorithmic programming systems, like artificial neural networks. The concept of anticipation is not to be confounded with programming of final objective goals. There is no anticipated goal for such a program, because the final result will depend on the input to the program in each step. This means that the result of such living non-algorithmic programs is not anticipated. For living systems input means interaction of the living systems with their environment.

The program defines the process of development of the living systems. Thus this means that the program in living systems includes the coded instructions for learning processes in neural brain and for anticipation.

Daniel M. Dubois, " Introduction to the Natural Anticipator and the Artificial Anticipator".

In: COMPUTING ANTICIPATORY SYSTEMS: CASYS'09 - Ninth International Conference, edited by D.M. Dubois, American Institute of Physics, AIP CP 1303, pp. 17-24, 2010.

As demonstrated by Nobel laureate McClintock [8], genomes are programmed.

The fundamental program deals with the DNA genetic code. The properties of the DNA consist in self-replication and self-modification. The self-replicating process leads to reproduction of the species, while the self-modifying process leads to new species or evolution and adaptation in existing ones. The genetic code DNA keeps its instructions in memory in the DNA coding molecule. The genetic code DNA is a rewriting system, from DNA coding to DNA template molecule. The DNA template molecule is a rewriting system to the Messenger RNA molecule. The information is not destroyed during the execution of the rewriting program.

Let us now study the properties of an artificial anticipator given by the Turing machine.

### 3. THE TURING MACHINE AS AN ARTIFICIAL ANTICIPATOR

Alan Mathison Turing (1912-1954) [9], British mathematician, formalised the concept of algorithm and computability with the Turing machine.

A Turing machine is an abstract model of a computer, with its memory, created by Turing, for giving a precise definition of the concept of algorithm, as a mechanical procedure.

The practical implementation of a Turing machine consists of 4 items:

1. A **TAPE** that is divided into cells, one next to the other. Each cell contains a symbol from any finite alphabet. The alphabet contains a special blank symbol, written, **B**, and one or more other symbols. The tape is assumed to be arbitrarily extendable to the left and to the right, in other words the machine must always have enough length of tape for his execution. All the cells that have been not yet written are marked with a blank, **B**.

2. A **HEAD READ/WRITE** that can read and write symbols on the tape, and move left or right one cell at a time.

3. An **ACTION TABLE** (a finite table or transition function) that tells the machine what symbol to write, how to move the **HEAD READ/WRITE** (**L** for a cell to the **LEFT**, **R** for a cell to the **RIGHT**), and what is the new state, depending on the scanned symbol on the tape and the current state of the Turing machine. If no action exists for a given combination of a scanned symbol and a current state, the machine stops, with a **FINAL STATE**, written, **F**.

4. A **STATE REGISTER** that stores the current state of the Turing machine. The number of possible states is always finite. There is a special state called start condition, which is the initial state of the machine before his execution, with an initial position of the head.

Let us consider the simple algorithm of a binary addition,  $X + 1 = Y$ , where  $X = 0, 1, 2, 3$ , in decimal notation, or in binary notation,  $X = (X1 X0) = (00), (01), (10), (11)$ , and the result  $Y = X + 1 = 1, 2, 3, 4$ , in decimal notation, or in binary notation  $Y = (Y2 Y1 Y0) = (001), (010), (011), (100)$ .

The Table 3 gives the Boolean table of this binary addition.

**TABLE 3.** Binary addition:  $X + 1 = Y$

<b>X</b>		<b>1</b>	<b>Y</b>		
<b>X1</b>	<b>X0</b>		<b>Y2</b>	<b>Y1</b>	<b>Y0</b>
<b>0</b>	<b>0</b>	<b>1</b>	<b>0</b>	<b>0</b>	<b>1</b>
<b>0</b>	<b>1</b>	<b>1</b>	<b>0</b>	<b>1</b>	<b>0</b>
<b>1</b>	<b>0</b>	<b>1</b>	<b>0</b>	<b>1</b>	<b>1</b>
<b>1</b>	<b>1</b>	<b>1</b>	<b>1</b>	<b>0</b>	<b>0</b>

The Table 4 gives the **ACTION TABLE** of the Turing machine that executes the algorithm of the binary addition.

<b>TABLE 4.</b> The <b>ACTION TABLE</b>			
	<b>SYMBOL READ</b>		
<b>CURRENT STATE</b>	<b>0</b>	<b>1</b>	<b>B</b>
1	( <b>0</b> , R, 1)	( <b>1</b> , R, 1)	( <b>B</b> , L, 2)
2	( <b>1</b> , R, 3)	( <b>0</b> , L, 2)	( <b>1</b> , R, 3)
3	( <b>0</b> , R, 3)	( <b>1</b> , R, 3)	( <b>B</b> , L, F)

Daniel M. Dubois, " Introduction to the Natural Anticipator and the Artificial Anticipator".

In: **COMPUTING ANTICIPATORY SYSTEMS: CASYS'09 - Ninth International Conference**, edited by D.M. Dubois, American Institute of Physics, AIP CP 1303, pp. 17-24, 2010.

In this Table 4, of the action table, there are three values of the current state given by, (1, 2, 3), and three symbols to be read, (0, 1, B), and there is a transition function, with three terms, corresponding to the symbol to write, the head to move right or left, and the new current state. For example, for the symbol, 0, read by the head, in the current cell, at the current state, 1, the transition function is given by, (0, R, 1), meaning that the head writes the symbol, 0, then moves to the right, R, and the new current state is, 1.

The initial state of the Turing machine is given as follows: the TAPE, given in Table 5A, contains the symbols, (1, 1), in two successive cells, corresponding to the binary number, 11, or 3 in decimal notation, and the initial condition of the HEAD is on the second symbol, 1. All the other cells of the TAPE are marked by a blank symbol B. The initial STATE of the state register is (1)

**TABLE 5A: TAPE WITH HEAD READ/WRITE**

...	B	B	1	1	B	B	...
				↑			
				HEAD			

**TABLE 5B: STATE REGISTER**

(1)	2	3	F
-----	---	---	---

The successive iterates of the Turing machine are given in the following Tables 6AB to 13AB.

For the conditions of Tables 5AB, the next iterate is given by the transition function, given in Table 4: (1, R, 1)

**TABLE 6A. TAPE WITH HEAD READ/WRITE**

...	B	B	1	1	B	B	...
					↑		
					HEAD		

**TABLE 6B. STATE REGISTER**

(1)	2	3	F
-----	---	---	---

For the conditions of Tables 6AB, the next iterate is given by the transition function, given in Table 4: (B, L, 2)

**TABLE 7A. TAPE WITH HEAD READ/WRITE**

...	B	B	1	1	B	B	...
					↑		
					HEAD		

**TABLE 7B. STATE REGISTER**

1	(2)	3	F
---	-----	---	---

For the conditions of Tables 7AB, the next iterate is given by the transition function, given in Table 4: (0, L, 2)

**TABLE 8A. TAPE WITH HEAD READ/WRITE**

...	B	B	1	0	B	B	...
				↑			
				HEAD			

**TABLE 8B. STATE REGISTER**

1	(2)	3	F
---	-----	---	---

For the conditions of Tables 8AB, the next iterate is given by the transition function, given in Table 4: (0, L, 2)

**TABLE 9A. TAPE WITH HEAD READ/WRITE**

...	B	B	0	0	B	B	...
				↑			
				HEAD			

**TABLE 9B. STATE REGISTER**

1	(2)	3	F
---	-----	---	---

Daniel M. Dubois, " Introduction to the Natural Anticipator and the Artificial Anticipator".  
 In: COMPUTING ANTICIPATORY SYSTEMS: CASYS'09 - Ninth International Conference,  
 edited by D.M. Dubois, American Institute of Physics, AIP CP 1303, pp. 17-24, 2010.

For the conditions of Tables 9AB, the next iterate is given by the transition function, given in Table 4: (1, R, 3)

**TABLE 10A. TAPE WITH HEAD READ/WRITE**

...	<b>B</b>	<b>1</b>	<b>0</b>	<b>0</b>	<b>B</b>	<b>B</b>	...
			↑				
			<b>HEAD</b>				

**TABLE 10B. STATE REGISTER**

1	2	(3)	F
---	---	-----	---

For the conditions of Tables 10AB, the next iterate is given by the transition function, given in Table 4: (0, R, 3)

**TABLE 11A. TAPE WITH HEAD READ/WRITE**

...	<b>B</b>	<b>1</b>	<b>0</b>	<b>0</b>	<b>B</b>	<b>B</b>	...
			↑				
			<b>HEAD</b>				

**TABLE 11B. STATE REGISTER**

1	2	(3)	F
---	---	-----	---

For the conditions of Tables 11AB, the next iterate is given by the transition function, given in Table 4: (0, R, 3)

**TABLE 12A. TAPE WITH HEAD READ/WRITE**

...	<b>B</b>	<b>1</b>	<b>0</b>	<b>0</b>	<b>B</b>	<b>B</b>	...
			↑				
			<b>HEAD</b>				

**TABLE 12B. STATE REGISTER**

1	2	(3)	F
---	---	-----	---

For the conditions of Tables 12AB, the next iterate is given by the transition function, given in Table 4: (B, L, F)

**TABLE 13A. TAPE WITH HEAD READ/WRITE**

...	<b>B</b>	<b>1</b>	<b>0</b>	<b>0</b>	<b>B</b>	<b>B</b>	...
			↑				
			<b>HEAD</b>				

**TABLE 13B. STATE REGISTER**

1	2	3	(F)
---	---	---	-----

The state F is reached, the algorithm stops.

The tape contains the following symbols, (1, 0, 0), corresponding to the binary number, 100, or 4 in decimal notation, which is the result of the addition, in binary notation,  $11 + 1 = 100$ , or  $3 + 1 = 4$ , in decimal notation.

Whatever the number written on the tape originally, the result of executing the instructions of the table will take to write this number + 1 on the tape: the Turing machine performs so well an algorithm.

The Turing machine is not intended as a practical computing technology, but rather as a thought experiment representing a computing machine, in view of understanding the limits of such a mechanical computation.

The Turing machine uses the same tape for data input and output.

The Turing machine is an anticipatory rewriting system.

The head reads and writes, modifying the content of the tape. The information is destroyed during the execution of the program. This is an irreversible process. The input data are lost.

Daniel M. Dubois, "Introduction to the Natural Anticipator and the Artificial Anticipator".

In: COMPUTING ANTICIPATORY SYSTEMS: CASYS'09 - Ninth International Conference, edited by D.M. Dubois, American Institute of Physics, AIP CP 1303, pp. 17-24, 2010.

## 4. CONCLUSION

This short paper deals with the introduction of the concept of anticipator in the framework of computing anticipatory systems.

The definition of anticipation deals with the concept of program.

Indeed, the word program, comes from “pro-gram” meaning “to write before” by anticipation, and means a plan for the programming of a mechanism, or a sequence of coded instructions that can be inserted into a mechanism, or a sequence of coded instructions, as genes or behavioural responses, that is part of an organism.

Any natural or artificial programs are thus related to anticipatory rewriting systems, as shown in this paper.

All the cells in the body, and the neurons in the brain, are programmed by the anticipatory genetic code, DNA, in a low-level language with four signs.

The programs in computers are also computing anticipatory systems.

It is shown, at one hand, that the genetic code DNA is a natural anticipator.

DNA, Deoxyribonucleic acid, is a nucleic acid that contains the genetic instructions used in the development and functioning of all known living organisms, after the discover of the double helix by Watson and Crick.

The main role of DNA molecules is the long-term storage of information. DNA contains the instructions needed to construct other components of cells, such as proteins and RNA molecules.

The DNA segments that carry this genetic information are called genes, but other DNA sequences have structural purposes, or are involved in regulating the use of this genetic information.

The genetic code consists of 64 triplets of nucleotides, called codons.

With three exceptions, each codon encodes for one of the 20 amino acids used in the synthesis of proteins.

Despite the minor variations that exist, the genetic code used by all known forms of life is nearly universal.

As demonstrated by Nobel laureate McClintock [8], genomes are programmed. The fundamental program deals with the DNA genetic code. The properties of the DNA consist in self-replication and self-modification. The self-replicating process leads to reproduction of the species, while the self-modifying process leads to new species or evolution and adaptation in existing ones.

The category of program given by the genetic code that is involved in developmental living systems belongs to non-algorithmic programming systems, like artificial neural networks. The concept of anticipation is not to be confounded with programming of final objective goals. There is no anticipated goal for such a program, because the final result will depend on the input to the program in each step. This means that the result of such living non-algorithmic programs is not anticipated. For living systems input means interaction of the living systems with their environment.

The genetic code DNA keeps its instructions in memory in the DNA coding molecule. The genetic code DNA is a rewriting system, from DNA coding to DNA template molecule. The DNA template molecule is a rewriting system to the Messenger RNA molecule. The information is not destroyed during the execution of the rewriting program.

On the other hand, it is demonstrated that Turing machine is an artificial anticipator.

Alan Mathison Turing, British mathematician, formalised the concept of algorithm and computability with the Turing machine.

A Turing machine is an abstract model of a computer, with its memory, created by Turing, for giving a precise definition of the concept of algorithm, as a mechanical procedure.

The practical implementation of a Turing machine consists of four items. Firstly, a tape that is divided into cells, one next to the other, containing a symbol from any finite alphabet. Secondly, a head read/write that can read and write symbols on the tape, and move left or right one cell at a time. Thirdly, an action table, or transition function, that tells the machine what symbol to write, how to move the head read/write, and what is the new state, depending on the scanned symbol on the tape and the current state of the Turing machine. Fourthly, there is a state register that stores the current state of the Turing machine.

The Turing machine is thus an anticipatory rewriting system. The head reads and writes, modifying the content of the tape. The information is destroyed during the execution of the program. This is an irreversible process. The input data are lost.

These genetic code DNA and Turing machine are both anticipators, which are based on anticipatory rewriting systems, given by non-algorithmic or algorithmic programs.

Daniel M. Dubois, "Introduction to the Natural Anticipator and the Artificial Anticipator".

In: COMPUTING ANTICIPATORY SYSTEMS: CASYS'09 - Ninth International Conference, edited by D.M. Dubois, American Institute of Physics, AIP CP 1303, pp. 17-24, 2010.

## REFERENCES

1. Dubois, Daniel (1990). LE LABYRINTHE DE L'INTELLIGENCE. De l'intelligence naturelle à l'intelligence fractale. InterEditions, Paris. ACADEMIA, Louvain-la-Neuve
2. Dubois, Daniel M. (1999), Hyperincursive McCulloch and Pitts Neurons for Designing a Computing Flip-Flop Memory. *Computing Anticipatory Systems: CASYS'98 – Second International Conference*. Published by The American Institute of Physics, *AIP Conference Proceedings* 465, pp. 3-21.
3. Dubois, Daniel M. (2000), Review of Incursive, Hyperincursive and Anticipatory Systems - Foundation of Anticipation in Electromagnetism. *Computing Anticipatory Systems: CASYS'99 – Third International Conference*. Published by The American Institute of Physics, *AIP Conference Proceedings* 517, pp. 3-30.
4. Dubois, Daniel M. (2001), Incursive and Hyperincursive Systems, Fractal Machine, and Anticipatory Logic. *Computing Anticipatory Systems: CASYS 2000 – Fourth International Conference*. Published by The American Institute of Physics, *AIP Conference Proceedings* 573, pp. 437-451.
5. Dubois, Daniel M. (2003), Mathematical Foundations of Discrete and Functional Systems with Strong and Weak Anticipations. In: Butz Martin et al (ed): *Anticipatory Behavior in Adaptive Learning Systems*, State-of-the-Art Survey. *Lecture Notes in Artificial Intelligence*, LNAI 2684, pp. 110-132, Springer-Verlag, Berlin, Heidelberg
6. Dubois, Daniel M. (2008). New Trends in Computing Anticipatory Systems: Emergence of Artificial Conscious Intelligence with Machine Learning Natural Language. *Computing Anticipatory Systems: CASYS'07 – Eighth International Conference*. Published by The American Institute of Physics, *AIP Conference Proceedings* 1051, pp. 25-32
7. François, Charles (1998), The Wishful Anticipator, *International Journal of Computing Anticipatory Systems*, Volume 2, pp. 44-49.
8. McClintock, B. (1983), *The Significance of responses of the genome to challenge*, Nobel lecture, [http://nobelprize.org/nobel\\_prizes/medicine/laureates/1983/mcclintock-lecture.pdf](http://nobelprize.org/nobel_prizes/medicine/laureates/1983/mcclintock-lecture.pdf)
9. Turing, A.M. (1936). On Computable Numbers, with an Application to the Entscheidungsproblem. *Proceedings of the London Mathematical Society*. 2, **42**: 230–65, 1937, and Turing, A.M. (1938). "On Computable Numbers, with an Application to the Entscheidungsproblem: A correction". *Proceedings of the London Mathematical Society*. 2, **43**: 544–6, 1937, <http://www.scribd.com/doc/2937039/Alan-M-Turing-On-Computable-Numbers>
10. Watson, J. D. and F. Crick (1953), Molecular structure of nucleic acids: A Structure for Deoxyribose Nucleic Acid, *Nature* **171** (4356): 737-738.

Daniel M. Dubois, " Introduction to the Natural Anticipator and the Artificial Anticipator".  
In: COMPUTING ANTICIPATORY SYSTEMS: CASYS'09 - Ninth International Conference,  
edited by D.M. Dubois, American Institute of Physics, AIP CP 1303, pp. 17-24, 2010.