



Tecnura

ISSN: 0123-921X

tecnura@udistrital.edu.co

Universidad Distrital Francisco José de Caldas  
Colombia

Martínez Sarmiento, Fredy Hernán; Delgado R., Alberto  
Seres vivos: fuente de inspiración para el diseño artificial  
Tecnura, vol. 17, núm. 37, julio-septiembre, 2013, pp. 121-137  
Universidad Distrital Francisco José de Caldas  
Bogotá, Colombia

Disponible en: <http://www.redalyc.org/articulo.oa?id=257028383012>

- ▶ Cómo citar el artículo
- ▶ Número completo
- ▶ Más información del artículo
- ▶ Página de la revista en redalyc.org

redalyc.org

Sistema de Información Científica

Red de Revistas Científicas de América Latina, el Caribe, España y Portugal  
Proyecto académico sin fines de lucro, desarrollado bajo la iniciativa de acceso abierto

# Seres vivos: fuente de inspiración para el diseño artificial

*Living beings: source of inspiration for artificial design*

## Fredy Hernán Martínez Sarmiento

Ingeniero Electricista, candidato a Doctor en Ingeniería. Docente de la Universidad Distrital Francisco José de Caldas. Bogotá, Colombia. Contacto: fhmartinezs@udistrital.edu.co

## Alberto Delgado R.

Ingeniero Electricista, Magíster en Ingeniería Eléctrica, Doctor en Cibernética. Docente de la Universidad Nacional de Colombia. Bogotá, Colombia.  
Contacto: jadelgador@unal.edu.co

Fecha de recepción: 2 de abril de 2012

Clasificación del artículo: Revisión

Fecha de aceptación: 12 de febrero de 2013

Financiamiento: Universidad Distrital Francisco José de Caldas - Universidad Nacional de Colombia

**Palabras clave:** emulación por hardware, sistemas inteligentes, tecnologías biológicamente inspiradas.

**Key words:** hardware emulation, intelligent systems, biologically inspired technologies.

## RESUMEN

Muchas de las cosas “inteligentes” que se desean en robots, máquinas y en general en los sistemas construidos por el hombre, son muy bien resueltas por la naturaleza. Por tanto, es razonable esperar que imitando a la naturaleza se puedan lograr “soluciones inteligentes” a los problemas humanos. En este artículo se presenta un análisis del estado del arte con respecto al desarrollo del diseño artificial inspirado en la naturaleza, con especial interés en los seres vivos. El estudio toma como punto de partida el modelo POE (Phylogeny, Ontogeny and Epigenesis) estableciendo un marco de referencia tridimensional con tres ejes de desarrollo biológico: filogenético, ontogenético y epigenético. Sobre cada uno de ellos se realiza un estudio de las implementaciones hardware existentes, de sus características, ventajas y debilidades, a fin de es-

tablecer un mapa general del desarrollo tecnológico actual. Al final del artículo, se concluye que la implementación de este tipo de algoritmos inspirados en la naturaleza no solo es importante, sino también indispensable en la solución de muchos problemas complejos hoy en día.

## ABSTRACT

Many of the “intelligent” features that we want in robots, machines and man-made systems, are already well implemented by nature. Therefore, it is reasonable to expect that, by imitating nature, “intelligent solutions” to human problems can be found. In this paper, we present a survey on the analysis of current developments connected to artificial design inspired by nature. Particular interest is devoted to living organisms. The study takes the POE model as the starting point, establishing a three-dimensional

framework with three axes of biological development, namely phylogenetic, ontogenetic and epigenetic axes. For each of these axes, a survey of existing hardware implementations is conducted in order to gather their characteristics, strengths and

weaknesses and establish a general map of the latest technological developments. Finally, we conclude that implementing this type of nature inspired algorithms is not only important but also indispensable in solving many complex problems today.

\* \* \*

## 1. INTRODUCCIÓN

Computadores “humanos”, sistemas hardware y software con la posibilidad de percibir e interpretar nuestro mundo bajo las mismas condiciones del ser humano. Capaces de ver, oír, sentir, gustar y oler. Capaces de interactuar con el ser humano. Estas son posibilidades concretas para un futuro no muy lejano, en el que el oriente de desarrollo para los sistemas artificiales como herramientas de apoyo para el ser humano se inspira en sistemas biológicos homólogos, considerados como robustos, eficientes y óptimos [1]. Estas funciones probablemente se incorporen exitosamente en sistemas muy diversos, pero íntimamente ligados al hombre.

Los seres vivos son máquinas complejas. Son sistemas compuestos por una gran variedad de herramientas especializadas que le permiten, como unidad autónoma, detectar estímulos, medirlos, analizarlos, producir una respuesta y ajustarla de acuerdo con su desempeño. Así, por ejemplo, a un felino no le cuesta ningún esfuerzo identificar y evaluar a su presa, y tomar la decisión de acuerdo a su capacidad y experiencia de atacar en el momento adecuado.

El campo de investigación en el cual se busca crear máquinas artificiales (hardware, software o ambos) que reproduzcan comportamientos equivalentes a los observados en sistemas biológicos, con capacidad para aprender y generar nuevo conocimiento a partir del conocimiento previo, y enfrentar exitosamente problemas tanto conocidos como desconocidos, es conocido como inteligencia artificial

(AI). Dentro de ese comportamiento inteligente, se incluyen tanto los procesos relacionados con el raciocinio como los relacionados con la conducta humana, así como aquellos procesos relacionados con la percepción y el reconocimiento de señales (imágenes, sonidos, olores, etc.) [2].

Un sistema artificial bioinspirado nace cuando se aplica para el diseño y desarrollo de sistemas analíticos, comportamientos, estructuras o procesos biológicos, plasmados en un algoritmo. Finalmente, lo que se busca es imitar la forma en que los sistemas biológicos son capaces de aprender y evolucionar para resolver sus problemas. En el diseño de estos sistemas se pueden caracterizar de forma más o menos clara tres enfoques diferentes: la evolución y adaptación de las especies, la evolución y adaptación de un individuo, y el desarrollo del individuo a lo largo de su vida.

La revisión presentada a continuación, sin aspirar a ser completa, pretende plasmar un marco general de referencia en cuanto a la aplicación de conceptos biológicos en el diseño de sistemas hardware. La estructura del texto está orientada hacia la concepción, diseño y desarrollo soportados en procesos observados en los seres vivos: (1) evolución biológica, la cual da lugar a los algoritmos evolutivos, (2) desarrollo celular, el cual conduce al diseño de sistemas con características autorreparadoras, y (3) los sistemas de aprendizaje del individuo, los cuales sirven como motivación subyacente para los sistemas de aprendizaje artificiales.

## 2. BIOLOGÍA ARTIFICIAL

Es posible afirmar que la biología artificial está soportada por dos tipos básicos de profesionales. El primer grupo es liderado por biólogos y químicos. Los primeros tienen un gran interés en diseñar y construir sistemas biológicos sintéticos, ya que ellos les proporcionan un método directo y completo para evaluar la comprensión actual de los sistemas biológicos naturales [3]. Por otro lado, para los químicos la biología es química, y la biología sintética es por tanto una extensión de la química sintética [4].

El segundo grupo se encuentra liderado por ingenieros, para quienes la biología es una tecnología, un conjunto de sistemas con un alto nivel de diseño, que a lo largo de miles de años de evolución y adaptación ha permitido decantar estructuras eficientes y robustas codificadas en genomas, que pueden ser rescritas y reinterpretadas con la idea de reproducir sustitutos que eventualmente remplacen sistemas actuales [5][6].

## 3. MODELO POE

Atendiendo la aproximación propuesta en 1997 por el Swiss Federal Institute [1], en la que se plantea el llamado modelo POE (Phylogeny, Ontogeny and Epigenesis) como sistema estructurado que se soporta en tres niveles de organización de la vida en la tierra desde sus orígenes —filogenia o filogenética (evolución de las especies), ontogenia (desarrollo y evolución de los organismos individuales), y epigenesis (desarrollo del individuo a lo largo de su vida a través del aprendizaje)—, es posible organizar y caracterizar coherentemente los diferentes desarrollos actuales en sistemas artificiales bioinspirados (figura 1).

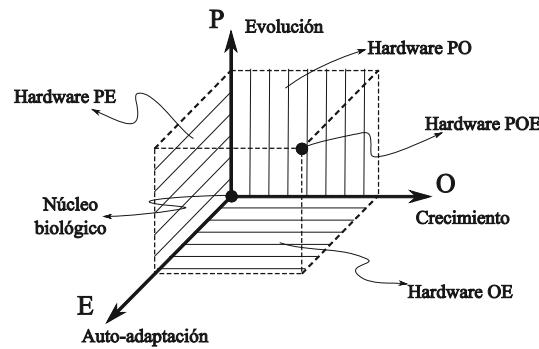


Figura 1. Modelo POE.

Fuente: tomada de [1].

La filogenética o filogenia es la parte de la biología que estudia la evolución de las especies de forma global, en contraposición a la ontogenia, que estudia la evolución del individuo. La ontogenia (también llamada morfogénesis u ontogénesis) describe el desarrollo de un organismo, o de alguna de sus partes, desde su concepción hasta su forma adulta. En especial, se refiere al periodo embrionario, por lo que también se le conoce como embriología.

En biología, la epigénesis es el proceso de desarrollo de un individuo, a través del cual su estructura se diferencia y se hace más compleja. El caso paradigmático es el del crecimiento, en el que a partir de un cigoto se desarrolla una compleja estructura celular y orgánica. Por extensión, en teoría de sistemas se incluyen los mecanismos que permiten a un determinado individuo modificar ciertos aspectos de su estructura interna o externa como resultado de la interacción con su entorno inmediato. La epigénesis representa por tanto el proceso de “sintonización” final mediante el cual cada individuo se adapta de forma eficiente a su entorno a partir de las capacidades contenidas en su código genético.

### 3.1 Sistemas artificiales basados en la filogenia

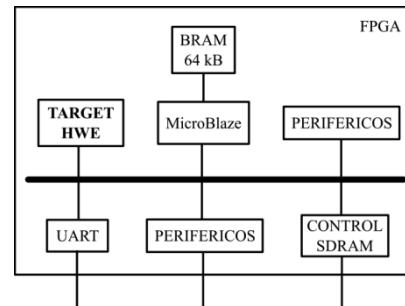
Este eje de desarrollo se conoce comúnmente como evolución artificial, y se centra en el desarrollo de hardware evolutivo (HWE). Introducido hace más de cuatro décadas, su principal motivación

es la construcción de sistemas hardware que sean capaces de auto-adaptarse en tiempo real, reconfigurando su arquitectura de manera dinámica y autónoma mediante una interacción con su ambiente, copiando técnicas evolutivas. Es decir, algoritmos basados en el mecanismo de selección natural para la síntesis de los circuitos (algoritmos genéticos, estrategias evolutivas, programación evolutiva y programación genética) [7].

Pese a que a finales de los años cincuenta (cien años después de que Charles Darwin enunciara su teoría sobre el origen de las especies) el biólogo Alexander Fraser publicó una serie de trabajos sobre la evolución de sistemas biológicos en un computador digital, y habida cuenta de los experimentos fallidos de R. M. Friedberg por evolucionar programas de computador, es a John R. Koza, profesor asesor de informática médica en la Universidad de Stanford, a quien se le considera hoy en día como el inventor de la programación genética. Su trabajo *Genetic Programming* presentado en 1992 [8], en donde muestra la aplicación de la programación genética para evolucionar un multiplexor de tres variables y un sumador de dos bits, se considera como la primera aproximación de aplicación de algoritmos evolutivos en la síntesis de sistemas. Muy a pesar de que Koza posteriormente trabajó con sistemas análogos [9], luego de examinar los trabajos reportados por la comunidad científica es claro que la mayoría de las aplicaciones que se desarrollan en este campo se orientan a la implementación de sistemas digitales [10] - [15].

De acuerdo con el trabajo de Koza [8], virtualmente todos los problemas en inteligencia artificial, aprendizaje de máquina (o maquinal), sistemas adaptables y aprendizaje automatizado, se pueden formular como una búsqueda para un programa de computador. Por tanto, la programación genética proporciona una forma de conducir con éxito la búsqueda de un determinado programa de computador en el espacio de programas de computador. Este principio se ha extendido al diseño de hardware.

En cuanto a estos sistemas embebidos para aplicaciones finales, recientemente se dio a conocer el que se considera a la fecha como el primer prototipo funcional de HWE. Desarrollado por un equipo de investigadores noruegos [16], este hardware evoluciona como lo harían las especies animales o vegetales, y se adapta al medio sin que sufra su funcionalidad. En su trabajo, se implementó por software un procesador MicroBlaze sobre una FPGA de Xilinx (Virtex-II Pro XC2VP30), en el cual el HWE opera como un módulo periférico esclavo (figura 2). La aplicación de prueba para la arquitectura hardware fue un problema de reconocimiento facial. En sus pruebas de laboratorio, lograron reducir el tiempo de operación 2,2 veces frente a su contraparte software corriendo sobre un PC Pentium M de 1,7 GHz.



**Figura 2.** Arquitectura de la aplicación incluyendo HWE  
Fuente: tomada de [16].

Como ejemplo característico de sistemas artificiales análogos, con Koza a la cabeza, un gran grupo de investigadores convencido de las posibilidades del esquema de diseño evolutivo ha logrado sintetizar sistemas completos de forma exitosa. Tales sistemas incluyen: circuitos eléctricos análogos [17][18], sistemas de lentes ópticos [19][20], antenas [21], esquemas de control análogos [20][22], sistemas mecánicos [23], y circuitos de computación cuántica [24]. De hecho, a partir de tales trabajos, Koza logró identificar cinco características comunes en los diseños [9]:

- Las representaciones nativas son suficientes cuando se está trabajando con programación genética. Es decir, para desarrollar productos competitivos con los diseños tradicionales del ser humano, basta el uso de información básica sobre el campo involucrado.
- La programación genética, luego de algunas pocas generaciones, produce rápidamente individuos viables que estructuralmente cumplen las condiciones mínimas exigidas por el simulador. Todos los desarrollos reportados utilizan un simulador para evaluar la adaptación de los individuos, sin embargo, no todos los individuos generados cumplen los requisitos mínimos para ser simulados (no son sistemas viables), razón por la cual normalmente se les asigna el peor valor de adaptación, con lo cual en pocas generaciones su número se reduce drásticamente.
- Al comienzo de la ejecución de un proceso de optimización por programación genética, para un sistema con múltiples partes, los mejores individuos tienden a tener muy pocas partes operativas.
- La programación genética generalmente explota las suposiciones razonables de un simulador. Los simuladores utilizados en estos procesos de optimización han sido concebidos y diseñados para evaluar diseños realizados por el hombre como parte de un proceso diseño/prueba y error, esto significa que no están preparados para evaluar estructuras del mundo real.
- Los ingenieros que trabajan con programación genética son capaces de producir diseños de sistemas viables, competitivos con los existentes, y no patentados con anterioridad. Al respecto, se debe tener claro que una patente es solo una solución a un problema, pero no la única o más óptima.

### 3.1.1 Características del hardware filogenético

La evolución artificial persigue un único objetivo predefinido. El diseño de un circuito electrónico se encuentra amarrado a un conjunto específico y preciso de especificaciones.

La población no tiene existencia material real. En el mejor de los casos, en lo que se ha llamado evolución *online* (se dice que la evolución es *offline* si se realiza por simulación, y *online* si implementa hardware), existe un circuito sobre el cual se implementa a los individuos de la población, uno a la vez, para evaluar su desempeño.

La ausencia de una población real en la que los individuos coexisten simultáneamente plantea grandes problemas a la hora de implementar las interacciones entre individuos.

Las implementaciones reales de los sistemas artificiales evolutivos, plantean grandes dudas sobre su verdadera justificación. En la mayoría de los casos de implementaciones de hardware evolutivo para sistemas digitales, se utilizan tablas de consulta para evaluar la adaptación de los individuos, tablas que contienen la descripción completa del circuito buscado, y que deben ser almacenadas en alguna memoria. Así las cosas, ¿no resulta una solución más rápida y económica la implementación directa de estas tablas en una memoria?

En cuanto al proceso evolutivo, se observa cómo los mecanismos son ejecutados de forma externa al circuito resultante, esto incluye tanto a los operadores como al cálculo de la aptitud. En cuanto a esta última, mientras que se ha avanzado en la evolución *online*, los valores de aptitud son normalmente almacenados en otro lugar.

Las diferentes etapas del proceso evolutivo son ejecutadas de forma secuencial, controladas por una unidad central. En caso de falla en esta unidad central, todo el proceso evolutivo falla.

# revisión |

La mayoría de diseños hardware basados en evolución, en cuanto a desempeño se presentan a la fecha como sub-óptimos frente a sus contrapartes tradicionales.

## 3.1.2 Tipos de hardware filogenético

De acuerdo a las formas y estructuras utilizadas para la implementación en hardware de estos sistemas, es posible caracterizar cuatro tipos específicos de hardware filogenético:

Evolución *online*: en este tipo de implementación, todas las operaciones genéticas se desarrollan dentro del hardware objetivo. A pesar de esto, en las aplicaciones reportadas aún muchas de las operaciones se simulan por software (poblaciones almacenadas en unidades externas, unidades externas para el control del proceso, etc.) (figura 3) [10], [12], [15].

Evolución *offline*: en este tipo de implementación, todas las operaciones genéticas se desarrollan fuera del hardware objetivo. Cuando tal simulación entrega una estructura factible, esta se implementa finalmente por hardware (figura 3) [11], [13], [14].

Evolución guiada o dirigida: en este tipo de implementación el criterio de adaptabilidad se impone como condición de diseño de acuerdo a la tarea a resolver. Todas las referencias de aplicación artificial consultadas caen dentro de esta categoría (figura 3). Evolución no guiada o abierta: este es el tipo de evolución que se verifica en la naturaleza, en la cual no existen criterios de adaptación externos impuestos más allá que la supervivencia en el ambiente. A la fecha, no fue posible identificar en la literatura reportada una aplicación artificial que caiga en este tipo, categoría que podría considerarse como correspondiente al verdadero hardware evolutivo.

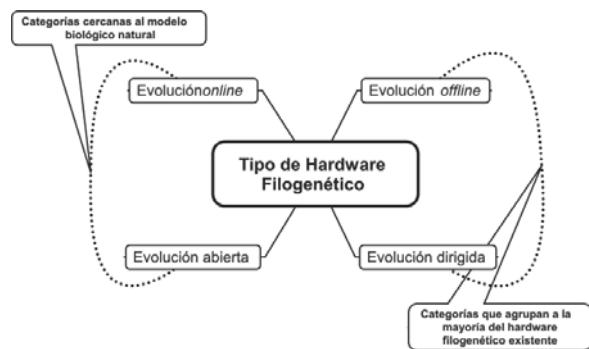


Figura 3. Tipos de hardware filogenético

Fuente: elaboración propia.

## 3.2 Sistemas artificiales basados en la ontogenia

El eje ontogenético, también conocido como morfogénesis u ontogénesis, involucra el desarrollo de un individuo (o de alguna de sus partes) desde su propio material genético, sin contemplar interacción directa con el medio ambiente. Este desarrollo ontogenético transcurre durante el tiempo de vida del ser (a diferencia de la filogenia que abarca siglos o milenarios), y debe verificar dos funciones básicas:

- Generar la diversidad celular (diferenciación celular) a partir del huevo fecundado (cigoto), y organizar los diversos tipos de células en tejidos y órganos (crecimiento).
- Asegurar la continuidad de la vida de una generación a la siguiente (reproducción).

Los organismos biológicos muestran una sorprendente habilidad durante su desarrollo ontogenético para desarrollar estrategias y soluciones a una gran cantidad de problemas de supervivencia. El ácido desoxirribonucleico (ADN) es el material biológico que sirve de archivo e instructivo. Todas las células de un organismo poseen una copia del mismo código, pero este se expresa en cada una de ellas de forma diferente dando lugar a la especialización en diferentes tejidos [27]. El proceso más importante en este eje es el de construcción celular. El hardware artificial debe por tanto implementar procesos de

replicación y regeneración, procesos que le darían la habilidad al sistema artificial de auto-repararse [25], [26], [28], [29].

El proceso de replicación celular puede verse como un caso particular de construcción celular, en el cual se crea un organismo idéntico por medio de la duplicación de su material genético. Es posible ubicar el inicio de la investigación del hardware ontogenético en los años cuarenta y cincuenta, con el trabajo de Von Neumann alrededor de las máquinas autorreplicantes [30], años antes de que el ADN fuera descubierto como el material genético de la naturaleza. En su visión, conocida como máquina de Von Neumann, este matemático proyecta una máquina dentro de un mar de recursos, con una cinta de memoria que le dicta instrucciones sobre las partes que debía tomar y cómo armarlas, y cuya instrucción final era la de copiar la memoria en la nueva máquina.

Posteriormente en 1956, el matemático Edward F. Moore sugirió una aplicación práctica de la máquina de Von Neumann [31]. En su modelo, propone plantas artificiales que toman todos los recursos naturales de una región, elementos con los cuales la máquina haría cables, engranajes, tornillos, etc., y los ensamblaría en una réplica de sí misma. Estos replicantes flotarían en los océanos, y cubrirían los desiertos.

Otras visiones, que extienden las ideas de Moore, incluyen el desarrollo de sistemas a fin de adecuar áreas inhabitables como desiertos y regiones árticas, ideas que también se extienden a territorios fuera de nuestro planeta. Este es el planteamiento que condujo a la NASA (National Aeronautics and Space Administration) en 1980 a conducir un estudio en conjunto con la ASEE (American Society for Engineering Education) denominado Advanced Automation for Space Missions, con el fin de desarrollar una propuesta detallada para el desarrollo de fábricas lunares auto-replicantes [29], [32].

En 1995, inspirado por las ideas de sembrar desiertos con máquinas auto-replicantes para desarrol-

ustrial, Klaus Lackner y Christopher Wendt desarrollaron un esquema detallado para tal sistema [33], [34]. Ellos propusieron una colonia de robots móviles cooperativos de 10 a 30 cm sobre una red electrificada alrededor de un equipo de fabricación fijo y con campos de celdas solares. Esta propuesta fue popularizada en Discovery Magazine, donde se mostró el equipo de desalación con alimentación solar [35].

Los trabajos más recientes (desde el 2000 hasta nuestros días) se han orientado más al desarrollo de técnicas de “robótica evolutiva” dentro de un mundo real, que al desarrollo de sistemas complejos auto-sostenidos [36]. Al respecto, conviene mencionar algunos de los desarrollos más destacados. En el arreglo electrónico planteado en 1997 por el Swiss Federal Institute [37], [38], por ejemplo, se pretendía duplicar estas características a nivel celular bajo el marco de lo que denominaron Embryonics (electrónica embrionaria). El objetivo final era la obtención de circuitos integrados de gran escala (VLSI) dotados con características de auto-reparación (cicatrización) y auto-replicación.

En este enfoque, se plantea la imitación de tres características del proceso de desarrollo celular: la organización multicelular, la diferenciación celular y la división celular. La primera característica (organización multicelular) se basa en un organismo artificial dividido en un número finito de células (células artificiales); la segunda (diferenciación celular) se fundamenta en el concepto de genoma y la tercera opera sobre el arreglo celular. Inicialmente se define una célula madre, la cual mantiene una copia del genoma; luego el genoma de la célula madre se copia en las células vecinas. El proceso continúa hasta que el espacio 2D se programa totalmente.

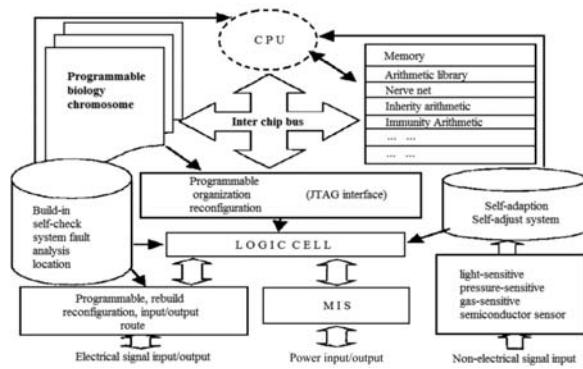
Otro enfoque propuesto en 2007 por el Departamento de Ingeniería Electrónica de la Universidad Politécnica de Cataluña [39][40], mantiene los lineamientos generales propuestos por el Swiss Federal Institute en cuanto al arreglo homogéneo de elementos programables. Sin embargo, ellos

plantean la implementación de las “células” dentro de la FPGA, no una FPGA por célula. Para ello, utilizan los bloques programables de la FPGA como células, conectadas por las conexiones programables. Elementos más complejos se implementan agrupando células para formar un componente, y agrupando componentes se obtiene un SANE (Self-Adaptive Network Element), elemento básico de procesamiento con la capacidad de monitorear el ambiente y sus procesos internos, y que se crea para desarrollar la tarea del sistema.

Otros sistemas sobre este eje que merecen ser tenidos en cuenta incluyen a los Sistemas-L y el modelo BIOSOC. Básicamente, un Sistema-L (L-System) o Sistema Lindenmayer, es una gramática formal (un conjunto de reglas y símbolos) principalmente utilizada para modelar el proceso de crecimiento de las plantas, pero que bien puede utilizarse para modelar también la morfología de una variedad de organismos y sistemas. Los sistemas-L fueron introducidos y desarrollados en 1968 por el biólogo y botánico húngaro Aristid Lindenmayer de la Universidad de Utrecht [41], [42]. La idea es la de emplear un modelo de desarrollo basado en Sistema-L para diseñar una estructura de crecimiento, con la interpretación gráfica dada por un autómata celular (Cellular Automaton - CA) [43], [44].

En 2002, la School of Infomation Science and Technology de la Shan Dong University (China) presentó el modelo Bionic SOC (BIOSOC), a través del cual se explora la posibilidad de imitar la actividad de un organismo mediante la automatización del diseño electrónico (EDA) sobre un circuito integrado [45]. En el modelo, las actividades relacionadas con pensamiento y memoria (actividad cerebral) se estructuran sobre una CPU y una memoria RAM. La red nerviosa, es decir, el canal de transferencia de señales fisiológicas, se implementa sobre un bus interno en el circuito integrado, separando un canal para datos y otro para comunicaciones. Los genes y cromosomas con la información hereditaria se asimilan a flujos de datos programables como series de bits estructurados. Las habilidades de crecimiento

y auto-reparación del organismo se planean sobre un sistema de re-configureación del modelo. La supervivencia frente a un ambiente cambiante se traza sobre el sistema de control auto-adaptable (figura 4). Es de aclarar, sin embargo, que la proyección de este modelo no cuenta a la fecha con un circuito integrado capaz de implementarlo.



**Figura 4.** Modelo BIOSOC

Fuente: Tomada de [45].

Existen sobre este eje otras aplicaciones hardware en las cuales se emulan algunos de los procesos concretos de construcción celular, tales son los casos del Chip ADN y del Gene Digital [46][47]. En estas aplicaciones, se desarrolla concretamente un proceso de hibridación electrónica, proceso de gran eficiencia en tareas de búsqueda no informada.

### 3.2.1 Características del hardware ontogenético

Los ejes filogenético y ontogenético pueden ser considerados ortogonales. Si se ubica una posición sobre el eje filogenético, y se desplaza esta posición a lo largo del eje (a lo largo de diferentes generaciones), el desplazamiento contemplará el paso de varias generaciones de individuos.

En coherencia con la característica anterior, hay que puntualizar que estos genes artificiales actúan en un único nivel. Participan en el proceso de desarrollo celular definiendo tan solo el fenotipo del sistema artificial; es decir, estos genes artificiales no

participan en el proceso genético de reproducción, no existen generaciones con variaciones genéticas.

La mayoría de los estudios sobre estructuras o máquinas auto-replicantes ha estado motivado por el deseo de entender los principios fundamentales y algoritmos involucrados en la auto-replicación, de forma independiente a su realización física. La fabricación de máquinas artificiales auto-replicantes a lo largo de la historia ha despertado gran interés para su uso en una gran variedad de aplicaciones, las cuales van desde la exploración espacial y la robótica hasta la propia supervivencia del ser humano en un planeta con recursos limitados.

Uno de los modelos centrales utilizados para el estudio de la auto-replicación artificial es el de los autómatas celulares. La razón de esto radica en un antecedente histórico, consecuencia de los trabajos de Von Newmann, quien los utilizó por su simplicidad y rigurosidad matemática. Estas ventajas siguen siendo válidas hoy en día.

Muchos de los desarrollos actuales basados en modelos ontogenéticos implementan esquemas de desarrollo celular sobre un mar de recursos predefinidos, sobre los cuales se programan las nuevas células (no existe la construcción celular). En estos modelos, normalmente existe una unidad central encargada de supervisar y controlar los procesos.

### **3.2.2 Tipos de hardware ontogenético**

De acuerdo a las formas y estructuras utilizadas para la implementación en hardware de éstos sistemas, es posible caracterizar seis tipos específicos de hardware ontogenético:

- Computación universal: sistemas concebidos para emular una máquina universal de Turing.
- Computación finita (o máquinas de estados finitas): sistemas concebidos para operar de acuerdo a reglas locales fijas, es decir, el estado de llegada está únicamente determinado

por el estado inicial y el carácter leído por el autómata.

- Construcción universal: sistemas concebidos con la habilidad de construir cualquier autómata descrito por un genoma artificial.
- Auto-reparación: sistemas concebidos específicamente para emular las características de auto-reparación y cicatrización.
- Organización multicelular: sistemas concebidos con diferenciación celular (normalmente espacial) en un arreglo de células artificiales idénticas.
- Organización unicelular: sistemas que, a diferencia del tipo anterior, se organizan en un arreglo de células idénticas, todas desarrollando la misma tarea para una determinada entrada.

### **3.3 Sistemas artificiales basados en la epigenésis**

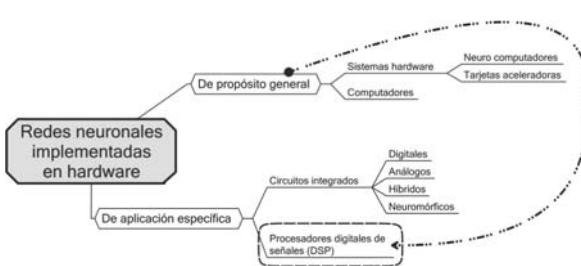
El eje epigenético involucra el aprendizaje mediante la interacción con el medio ambiente, que tiene lugar luego de la formación del individuo. Tomando nuevamente como referente a los organismos vivos, es posible identificar tres tipos concretos de sistemas epigenéticos:

- Sistemas basados en el sistema nervioso central: el de mayor desarrollo desde el punto de vista de la ingeniería de sistemas hardware, dado el gran interés que han despertado desde que McCulloch y Pitts formularon un primer modelo conexionista [48]. Se soporta en un elemento básico simple, la neurona artificial, modelo simplificado de la neurona biológica, el cual en un arreglo o red es capaz de duplicar dinámicas complejas al ajustar sus conexiones por medio de entrenamiento [49].
- Sistemas basados en el sistema inmunológico: el sistema inmunológico de un organismo

vivo superior es también una organización que aprende, es por ello que una persona no se contagia del mismo resfriado dos veces, y es la razón por la cual las vacunas funcionan. Una característica notoria de estos sistemas es la capacidad de reconocimiento altamente distribuido, con una gran diversidad de componentes en todo el organismo [50].

- Sistemas basados en el sistema endocrino: el sistema endocrino, particularmente el humano, está compuesto por un gran número de tejidos glandulares que tienen en común el hecho de que secretan hormonas directamente en la corriente sanguínea, que regulan e integran las funciones corporales del individuo. Este sistema se asemeja en algunas de sus funciones al sistema nervioso central, sin embargo, su velocidad de respuesta es más lenta que la transmisión neuronal, aunque más persistente [51].

Los sistemas neuronales implementan dinámicas difíciles de simular, debido a que están compuestas por un gran número de elementos no lineales, y a que involucran un gran rango de constantes de tiempo. Un modelo de red neuronal puede ser implementado o simulado de tres formas básicas: (1) programando un computador de propósito general, (2) programando una red neuronal sobre una máquina dedicada (un neuro computador), y (3) construyendo un dispositivo de propósito especial para implementar una red en particular y su algoritmo de aprendizaje (figura 5).



**Figura 5.** Redes neuronales implementadas por hardware  
Fuente: elaboración propia.

Una máquina de propósito general puede simular cualquier modelo de red neuronal y cualquier algoritmo de aprendizaje; sin embargo, la velocidad de tales simulaciones tiende a ser lenta. Máquinas dedicadas para redes neuronales pueden trabajar mucho más rápido, pero son normalmente más difíciles de programar. Hardware de propósito especial suministra bastante desempeño para aplicaciones de tiempo real pero demanda una considerable inversión.

La primera iniciativa por construir un hardware especial para redes neuronales artificiales data de 1951, cuando Marvin Minsky creó SNARC (Stochastic Neural-Analog Reinforcement Computer), computador en el cual con potenciómetros se simulaba la adaptabilidad de los pesos de la red. Posteriormente, a finales de los años 50, Rosenblatt implementó el primer perceptrón utilizando resistencias variables de la misma forma en que lo hizo Minsky [52].

A principios de los sesenta Widrow y Hoff [49] utilizaron elementos especialmente desarrollados llamados memistors, resistencias ajustables eléctricamente, para implementar redes tipo Adaline y Madaline. Ya a comienzos de los ochenta, justo antes de que John Hopfield reviviera el interés por las redes neuronales, Aleksander implementó el WISARD (Wilkie, Stonham and Aleksander's Device), una red de simple implementación basada en RAM [53] de una técnica de muestreo de n-tuplas. Aleksander aseguraba que un dispositivo de memoria tipo RAM puede funcionar como una neurona en la que un conjunto de entradas  $x_1$  a  $x_n$ , aprende a duplicar una salida deseada ajustando los valores de la RAM. Este tipo de modelos son llamados ingravidos o sin peso (*weightless*).

Las implementaciones análogas tienen la gran ventaja de explotar las propiedades físicas para operar la red neuronal artificial, con lo cual se obtiene una alta velocidad de procesamiento y una gran densidad. Sin embargo, el diseño puede ser bastante complejo. Otro inconveniente está relacionado con

el almacenamiento de los pesos. Esta alternativa análoga ha sido particularmente exitosa en el diseño de sistemas neuromórficos como las retinas y cócleas de silicio [54].

Los neuro computadores digitales son básicamente sistemas multi-procesadores, aunque en muchos casos se han diseñado procesadores personalizados para redes neuronales artificiales [55][56] [57], lo que normalmente toma bastante tiempo de desarrollo. Los sistemas multi-procesadores, con procesadores comerciales, tienen la ventaja de un corto tiempo de diseño, bajo costo y buen soporte de software. En los trabajos de investigación más recientes, se encuentran referencias a varios algoritmos de gran velocidad; desafortunadamente, el hardware específico que se encuentra en el mercado para neuro computadores [54] no los utiliza, encontrándose reporte de aplicaciones de algoritmos lentos sobre hardware de gran velocidad.

Dos arquitecturas dominan el diseño de los neuro procesadores: instrucción sencilla con múltiple dato SIMD (Single Instruction with Multiple Data) [58] y arreglos sistólicos (*systolic arrays*) [59]. Estas arquitecturas ejecutan la misma instrucción de forma paralela pero sobre datos diferentes. Los esquemas híbridos combinan la tecnología digital y analógica. Las variaciones incluyen procesamiento interno analógico para velocidad y pesos ajustados digitalmente, y redes de pulsos que utilizan una rata o ancho de pulso para emular la amplitud de las E/S y de los pesos.

El segundo enfoque epigenético se basa en el sistema inmunológico de los organismos. Este sistema es uno de los mayores campos de investigación en biología; se ha demostrado que es capaz de aprender, reconocer y eliminar cuerpos extraños que invaden al organismo. Visto desde el punto de vista de la ingeniería, es de gran interés duplicar un sistema que mantenga su inmunidad frente a cambios en el ambiente.

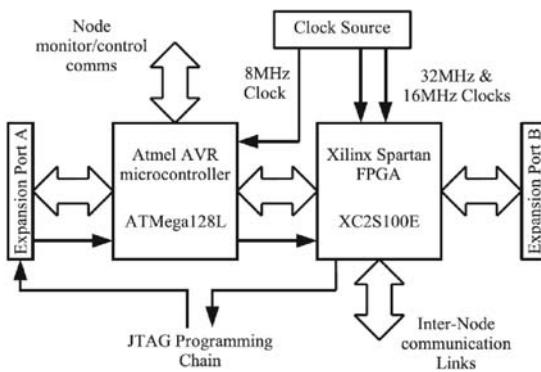
Dentro de las primeras aplicaciones que se dieron a los Sistemas Inmunológicos Artificiales (SIA) está la detección y corrección de fallas causadas por virus, siendo Kephart quien la propuso por primera vez. Hoy en día, los sistemas inmunes artificiales han extendido su rango de acción a la eliminación de basura, reparación de daños, protección de sistemas y detección de intrusos [60]. La respuesta inmune en el sistema inmunitario humano aprende a través de un proceso centralizado de maduración, en el cual crea una colección de anticuerpos capaz de detectar la invasión de un elemento extraño. Aplicando esta analogía al campo del hardware electrónico, es posible desarrollar un sistema de detección de errores integrado en el ciclo de desarrollo del hardware, a fin de permitir que el sistema final soporte un gran número de fallas no con un sistema adicional, sino como parte de su propio diseño [61], [62].

Finalmente, en cuanto a esquemas hardware inspirados en el sistema endocrino se encuentran algunos pocos desarrollos [63][64]. Dentro de los animales superiores opera un sistema de comunicación intercelular que se encarga de regular el equilibrio fisiológico de su anfitrión. El sistema endocrino utiliza mensajes hormonales para controlar de forma remota grupos de células, e invoca reacciones para mantener el equilibrio químico y físico. El funcionamiento y la estructura del sistema endocrino exhiben robustez y tolerancia a fallos.

Tales son los casos del modelo emocional implementado en el robot japonés WAMOEBA-2R (Waseda Amoeba, Waseda Artificial Mind on Emotion Base) y del sistema o red de comunicación intercelular Bionode (Bionode System). En el primero, el objetivo fue el de desarrollar un robot móvil que sirviera de interface con el ser humano, para lo cual se desarrolló un modelo emocional sin la necesidad de definir y clasificar los estados emotivos, sino introduciendo un modelo basado en el sistema endocrino. El software del robot basado en el sistema endocrino, se implementó utilizando teoría de conjuntos difusos con cuatro tipos de parámetros hormonales que controlan sus condi-

ciones internas, usando en la arquitectura hardware sensores internos equivalentes a los presentes en el sistema endocrino.

Salta la vista en este trabajo que el modelo endocrino se ha implementado por software (pese a tratarse de un robot), caso contrario a los trabajos desarrollados por Greensted y Tyrrell [64]. En este segundo caso, estos investigadores crearon un sistema de 30 nodos individuales conectados en una red débilmente acoplada. Cada Bionode contiene un microcontrolador y una FPGA, que pueden ser configurados para modelar la funcionalidad de una célula, concretamente el sistema de comunicación intercelular (figura 6). Esta arquitectura multiprocesador implementa entonces tal esquema de comunicación emulando la operación de las hormonas como un sistema paralelo altamente robusto, en el cual las células que fallan son enmascaradas, removidas y remplazadas *online*, asegurando la operación del sistema en el evento de falla.



**Figura 6.** Diagrama de bloques de un módulo Bionode  
Fuente: Tomada de [64].

### 3.3.1 Características del hardware epigenético

De acuerdo a las estructuras utilizadas para la implementación en hardware de éstos sistemas, es posible identificar algunas características específicas:

- Paralelismo y robustez (tolerancia a fallos): el diseño paralelo de estos sistemas artificiales

(información distribuida) les permiten, frente a una destrucción parcial del sistema, tener como consecuencia una degradación de su estructura (pérdida de alguna información), pero no la falla del sistema, el cual puede seguir operando para la mayoría de las condiciones normales.

- Aprendizaje adaptativo: la capacidad de aprender a realizar tareas basada en un entrenamiento o experiencia inicial, es una característica fundamental de estos sistemas.
- Generalización: la capacidad de generar una salida correcta frente a una situación nueva o desconocida, es una de las características más sobresalientes de los sistemas epigenéticos biológicos. En los sistemas artificiales, tal característica se logra emular mediante la identificación de características o parámetros en la información de entrada, identificación que en algunos casos es realizada por el mismo sistema artificial, pero en muchos otros obedece a criterios del diseñador.
- Auto-organización: los sistemas epigenéticos biológicos son capaces de crear su propia organización o representación de la información que reciben. En los sistemas artificiales, esto normalmente se trata de emular mediante una etapa de aprendizaje, que a diferencia de su contraparte biológica, normalmente está predefinida en operación y alcances.
- Operación en tiempo real: la operación en paralelo de estos sistemas brinda la posibilidad de obtener sistemas complejos operando en tiempo real, si son implementados sobre hardware especial (tal como ocurre en los sistemas biológicos). Sin embargo, existen muchas aplicaciones de tales modelos biológicos sobre hardware de uso general en los que se realiza una simulación de la operación, lo cual claramente los diferencia de los organismos biológicos.

- Inserción dentro de la tecnología existente: para la mayoría de estos modelos biológicos, es relativamente simple obtener hoy en día circuitos integrados (microcontroladores, DSPs, CPLDs o FPGAs) capaces de simular el comportamiento de estos sistemas, implementación que bien puede mejorar el desempeño en ciertas tareas.

#### 4. CONCLUSIONES

Los trabajos presentados permiten concluir que la implementación de modelos de diseño basados en seres vivos para el desarrollo de hardware electrónico es una posibilidad real que ha estado en la cabeza de los investigadores desde hace mucho

tiempo. La gran complejidad alcanzada hoy en día en muchos diseños tradicionales de equipo electrónico no llega a competir en complejidad con los organismos vivos, y lo que es aún más interesante, no logra competir con ellos en cuanto a confiabilidad, adaptabilidad y desempeño. Si bien es posible encontrar aproximaciones interesantes de diseño bio-inspirado, tan solo son aproximaciones iniciales, que difícilmente se pueden utilizar en usos reales. En estas aproximaciones, es posible identificar tendencias de diseño, de las cuales es posible afirmar que aún no se ha logrado el desarrollo artificial en tiempo real de los procesos biológicos de los seres vivos.

---

#### REFERENCIAS

---

- [1] D. Mange, and M. Tomassini, “Bio-Inspired Computing Machines”, *Towards Novel Computational Architectures*, Presses Polytechniques et Universitaires Romandes, Suiza, 1998.
- [2] S. J. Russell y P. Norvig, *Inteligencia artificial, un enfoque moderno*, Segunda edición, Madrid: Pearson Prentice Hall, 2004, pp. 2-35.
- [3] D. Sprinzak, and M.B. Elowitz, “Reconstruction of Genetic Circuits”, *Nature*, Nov. 2005, vol. 438, pp. 443-448.
- [4] S. Benner, and A. Sismour, “Synthetic Biology”, *Nature*, Jul. 2005, vol. 6, pp. 533-543.
- [5] L.Y. Chan, S. Kosuri and D. Endy, “Refactoring Bacteriophage T7”, *Molecular Systems Biology*, published online, 13 Sept. 2005, Article number: 2005.0018 doi:10.1038/msb4100025.
- [6] A. Stauffer, D. Mange and J. Rossier, “Design of Self-organizing Bio-inspired Systems”, *Second NASA/ESA Conf. on Adaptive Hardware and Systems*, IEEE Computer Society, 2007, pp. 413-419.
- [7] G. Greenwood and A. Tyrrell, *Introduction to Evolvable Hardware: A Practical Guide for Designing Self-Adaptive Systems*, IEEE Press Series on Computational Intelligence, Wiley: IEEE Press, 2007.
- [8] J.R. Koza, *Genetic Programming—On the Programming of Computer by Means of Natural Selection*, Cambridge, Massachusetts: The MIT Press, Massachusetts Institute of Technology, 1992.
- [9] J.R. Koza, S.H. Al-Sakran and L.W. Jones, “Cross-Domain Features of Runs of Genetic Programming Used to Evolve Designs for Analog Circuits, Optical Lens Systems, Controllers, Antennas, Mechanical Systems, and Quan-

- tum Computing Circuits”, *Conference NASA/DoD of Evolution Hardware (EH-05)*, 2005, pp. 205-212.
- [10] J. Wang, Q. Chen and C. Lee, “Design and implementation of a virtual reconfigurable architecture for different applications of intrinsic evolvable hardware”, *IET Computers & Digital Techniques*, vol. 2, 2008, pp. 386-400.
- [11] T. C. Oliveira y V. Pilla, “An Implementation of Compact Genetic Algorithm on FPGA for Extrinsic Evolvable Hardware”, *4th Southern Conf. on Programmable Logic*, 2008, pp. 187-190.
- [12] Q. Wang, “ATM Cell Scheduling and Learning by Function-Level Evolvable Hardware”, *Pacific-Asia Workshop on Computational Intelligence and Industrial Application*, vol. 2, 2008, pp. 727-729.
- [13] Ch. Jie, Z. Qiang, D. Guo-liang and Y. Liang, “The Implementation of Evolvable Hardware Closed Loop”, *Inter. Conf. on Intelligent Computation Technology and Automation*, vol. 2, 2008, pp. 48-51.
- [14] S. M. Cheang, K. Hong Lee y K. Sak Leung, “Applying Genetic Parallel Programming to Synthesize Combinational Logic Circuits”, *IEEE Trans. on Evolutionary Computation*, vol. 11, 2007, pp. 503-520.
- [15] M. Iwata, I. Kajitani, Y. Liu, N. Kajihara and T. Higuchi, “Implementation of a Gate-Level Evolvable Hardware Chip”, *Evolvable Systems: From Biology to Hardware*, Lecture Notes in Computer Science, Springer Berlin, vol. 2210/2001, 2001.
- [16] K. Glette, J. Torresen, M. Yasunaga and Y. Yamaguchi, “On-Chip Evolution Using a Soft Processor Core Applied to Image Recognition”, *Conf. on Adaptive Hardware and Systems AHS 2006*, First NASA/ESA, 2006, pp. 373-380.
- [17] J. R. Koza, F.H. Bennett III, D. Andre and M. A. Keane, *Genetic Programming III: Darwinian Invention and Problem Solving*, San Francisco: Morgan Kaufmann, 1999.
- [18] J. R. Koza, M. A. Keane, M. J. Streeter, W. Mydlowec, J. Yu and G. Lanza, *Genetic Programming IV: Routine Human-Competitive Machine Intelligence*, Kluwer Academic Publishers, 2003.
- [19] S.H. Al-Sakran, J.R. Koza and W. Lee, “Automated Re-invention of a previously Patented Optical Lens System Using Genetic Programming”, *8th European Conf. Genetic Programming EuroGP*, Lecture Notes in Computer Science 3447, Switzerland, 2005, Springer Verlag, pp. 25-37.
- [20] J. R. Koza, S. H. Al-Sakran and W. Lee, “Automated Re-invention of Six Patented Optical Lens Systems Using Genetic Programming”, *Genetic and Evolutionary Computation Conf. GECCO*, 2005.
- [21] J. D. Lohn, G.S. Hornby and D. S. Linden, An Evolved Antenna for Deployment on NASA’s Space Technology 5 Mission, *O’Reilly Genetic Programming Theory and Practice II*, Dordrecht: Kluwer, 2004, Ch 18.
- [22] M.A. Keane, J. R. Koza and M. J. Streeter, *Aparatus for Improved General Purpose PID and Non-PID Controllers*, U. S. Patent 6,847,851, Issued January 25, 2005.

- [23] H. Lipson, "How to Draw a Straight Line Using a GP: Benchmarking Evolutionary Design Against 19th Century Kinematic Synthesis", *Genetic and Evolutionary Conference 2005*, Late-Breaking Papers, USA, International Society for Genetic and Evolutionary Computation, 2005.
- [24] L. Spector, *Automatic Quantum Computer Programming: A Genetic Programming Approach*, Boston: Kluwer Academic Publisher, 2004.
- [25] M. Sipper, E. Sánchez, D. Mange, M. Tomassini, A. Pérez and A. Stauffer, "A Phylogenetic, Ontogenetic, and Epigenetic View of Bio Inspired Hardware Systems", *IEEE Transactions on Evolutionary Computation*, vol. 1, no. 1, pp. 83-97, 1997.
- [26] M. Sipper, D. Mange and A. Stauffer, "Ontogenetic Hardware", *BioSystems*, vol. 44, pp. 193-207, 1997.
- [27] A. Y. Zomaya, *Parallel Computing for Bioinformatics and Computational Biology: Models, Enabling Technologies, and Case Studies*, New Jersey: John Wiley & Sons Inc., 2006.
- [28] D. Mange, M. Goeke, D. Madon, A. Stauffer, G. Tempesti and S. Durand, "Embryonics: A New Family of Coarse-Grained Field Programmable Gate Array with Self-Repair and Self-Reproducing Properties", *Towards Evolvable Hardware*, vol. 1062 LN in Computer Science, pp. 197-220, Springer-Verlag, 1996.
- [29] R. A. Freitas and W. P. Gilbreath, "Advanced Automation for Space Missions", *Proceedings of the NASA/ASEE Summer Study*, Ch.5: Replicating Systems Concepts: Self-Replicating Lunar Factory and Demostration, NASA, Scientific and Technical Information Branch, 1980.
- [30] J. Von Neumann, *Theory of Self-Reproducing Automata*, Edited and completed by Arthur W. Burks, Urbana: University of Illinois Press, 1966.
- [31] E. F. Moore, "Artificial Living Plants", *Scientific American*, vol. 195, pp. 118-126, 1956.
- [32] R. A. Freitas and W. B. Zachary, "A Self-Replicating, Growing Lunar Factory", *Proceedings of the Fifth Princeton/AIAA Conference*, American Institute of Aeronautics and Astronautics, 1981.
- [33] K. S. Lackner y C. H. Wendt, "Exponential Growth of Large Self-Reproducing Machine Systems", *Mathematical and Computer Modelling*, Elsevier Science Ltd., vol. 21, Issue 10, 1995, pp. 55-81.
- [34] K.S. Lackner and C. H. Wendt, "Self-Reproducing Machine Systems for Global Scale Projects", *4th International Conference and Exposition on Engineering, Construction and Operations in Space/Conference and Exposition/Demonstrations on Robotic for Challenging Environments*, Albuquerque, 1994.
- [35] T. Bass, "Robot, Build Thyself", *Discover Magazine: Science, Tech. and the Future*, 1995, pp. 64-72.
- [36] R.A Freitas and R. Merkle, "Kinematic Self-Replicating Machines", *Landes Bioscience*, Georgetown, 2004.
- [37] D. Mange, E. Sánchez, A. Stauffer, G. Tempesti, S. Durand, and C. Piguet,

- «Embryonics: A New Methodology for Designing Field-Programmable Gate Arrays with Self-Repair and Self-Reproducing Properties», *IEEE Trans. on Very Large Scale Integration (VLSI) Systems*, vol. 6, pp. 387-399, 1998.
- [38] D. Mange and A. Stauffer, Introduction to Embryonics: Towards New Self-Repairing and Self-Reproducing Hardware Based on Biological-Like Properties, *Artificial Life and Virtual Reality*, England: John Wiley, Thalmann Editors, 1994, pp. 61-72.
- [39] J. A. Casas, J. M. Moreno, J. Madrenas and J. Cabestany, “A Novel Hardware Architecture for Self-Adaptive Systems”, *Conference on Adaptive Hardware and Systems 2007*, AHS 2007. Second NASA/ESA, 5-8 Aug. 2007, Edinburgh, pp. 592-599.
- [40] J. A. Casas, J. M. Moreno, J. Madrenas y J. Cabestany, “Arquitectura Hardware para Sistemas Autoadaptativos”, *Sимпозиум VII Jornadas sobre Computación Reconfigurable y Aplicaciones 2007*, Zaragoza - España, Septiembre 12-14 de 2007.
- [41] A. Lindenmayer, “Mathematical Models for Cellular Interaction in Development”, Parts I and II, *Journal of Theoretical Biology*, vol. 18, 1968, pp. 280-315.
- [42] P. Prusinkiewicz and A. Lindenmayer, *The Algorithmic Beauty of Plants*, Heidelberg: Springer-Verlag, 1990.
- [43] M. Sipper, *Evolution of Parallel Cellular Machines: The Cellular Programming Approach*, Heidelberg: Springer-Verlag, 1997.
- [44] D. Mange, A. Stauffer, L. Peparaolo and G. Tempesti, “A Macroscopic View of Self-Replication”, *Proceedings of the IEEE*, vol. 92, Issue 12, Dec. 2004, pp. 1929-1945.
- [45] F. Zeng, Q. Wang and M. Zeng, “The Method of Bionic System on a Chip”, *Proceedings 5th International Conference on ASIC 2003*, vol. 1, 21-24 Oct. 2003, pp. 45 - 48.
- [46] J. Prieto, O. Ramos y A. Delgado, “Diseño de un Gene Digital en FPGA Y Matlab con Aplicaciones en Robótica móvil”, *XIII Taller Iberchip*, Lima - Perú, marzo 14-16, 2007.
- [47] D. Perlaza y A. Delgado, “Detección de Fallas en Sistemas de Potencia con Chip ADN en FPGA”, *XI Taller Iberchip*, Salvador de Bahía - Brasil, marzo 28-30, 2005.
- [48] W. S. McCulloch y W. Pitts, “A Logical Calculus of the Ideas Immanent in Neurons Activity”, *Bulletin of Mathematical Biophysics*, vol. 5, 1943, pp. 115-133.
- [49] M. Arbib, *The Handbook of Brain Theory and Neural Networks*, Wisconsin: The MIT Press, 2003.
- [50] L. Wai-Yin, *Computer Immunology*, Tesis Departament of information Technology and Electrical Engeneering University of Queensland, October 2002, New York.
- [51] C. G. Morris y M. E. Ortiz, *Introducción a la Psicología*, México D.F.: Pearson Educación, Edición 12, 2005.
- [52] E. Kussul, T. Baidyk, L. Kasatkina and V. Lukovich, “Rosenblatt Perceptrons for Handwritten Digit Recognition”, *Inter-*

*Joint Conf. on Neural Networks IJCNN '01*, vol. 2, 2001, pp. 1516-1520.

- [53] J. Austin, "A Review of RAM based Neural Networks", *Proceedings of the Fourth International Conference on Microelectronics for Neural Networks and Fuzzy Systems*, 1994, pp. 58-66.
- [54] C. Mead, *Analog VLSI and Neural Systems*, 1st edition, Michigan: Addison Wesley Publishing, 1989.
- [55] N. Avellana, A. Strey, R. Holgado, J. A. Fernandes, R. Capillas, and E. Valderama, "Design of a Low-Cost and High-Speed Neurocomputer System", *Proceedings of Fifth International Conference on Microelectronics for Neural Networks*, 12-14 Feb. 1996, pp. 221-226.
- [56] U. Ramacher, "Synapse-X: A General-Purpose Neurocomputer Architecture", *IEEE International Joint Conference on Neural Networks*, vol. 3, pp. 2168-2176, 18-21 Nov., 1991.
- [57] W. Cao, Y. Wan and S. Wang, "Greeting Speech Recognition Based on Semiconductor Neurocomputer", *Inter. Conf. on Neural Networks and Brain*, vol. 3, 2005, pp. 1453-1458.
- [58] D. C. Hendry, A. A. Duncan and N. Lightowler, "IP Core Implementation of a Self-Organizing Neural Network", *IEEE Transactions on Neural Networks*, vol. 14, Issue 5, pp. 1085-1096, 2003.
- [59] I. Manolakos and E. Logaras, "High Throughput Systolic SOM IP Core for FPGAs", *Processing IEEE Inter. Conf. on Acoustics, Speech and Signal ICASSP 2007*, vol. 2, pp. II-61, II-64, 2007.
- [60] M. Burgess, "Computer Immunology", *Proceedings of the Twelfth Systems Administration Conference*, Dec. 6-11, 1998, Boston, Massachusetts, pp. 282-298.
- [61] D. Bradley and A. Tyrrell, "Immunotronics - Novel finite-state-machine architectures with built-in self-test using self-nonslef differentiation", *IEEE Trans. on Evolutionary Computation*, vol. 6, pp. 227-238, 2002.
- [62] D. W. Bradley and A. M. Tyrrell, "Hardware fault tolerance: an immunological solution", *IEEE International Conf. on Systems, Man, and Cybernetics 2000*, vol. 1, pp. 107-112, 2000.
- [63] T. Ogata and S. Sugano, "Robot Hardware which Built in the Endocrine Model", *Journal of the Society of Biomechanisms*, vol. 23, no. 2, pp. 106-111, 1999.
- [64] A. J. Greensted and A. M. Tyrrell, "Implementation Results for a Fault-Tolerant Multicellular Architecture Inspired by Endocrine Communication", *Conf. on Evolvable Hardware NASA/DoD*, 2005, pp. 253 - 261.