International Journal of Combinatorial Optimization Problems and Informatics, 13(1), Jan-April 2022, 33-49. ISSN: 2007-1558.

www.editada.org

Acetyl-modulated architecture for evolutionary robotics

Fernando Aldana-Franco [1], Fernando Montes-González[1], Alberto Ochoa-Zezzatti[2]

1 Artificial Intelligence Research Center. University of Veracruz, Xalapa, Mexico.
2. City of Juarez Autonomous University, City of Juarez, Mexico.

faldana@uv.mx

Abstract. Control modularity is an important tool for improving the organisation and potential task resolution in Evolutionary Robotics. Neuro-controllers are divided into many sections or many coordinated networks to solve a particular task. In this paper, we present a model for control evolutionary robots inspired by the effects of acetylcholine neurotransmitters, chemical synapse, renshaw cells, and based on artificial neural networks. The performance of our model is compared to the other two implementations: a) a system controlled by one single neural network and b) a control system based on one neural network divided into different sections. A garbage collection and room swap task were used for the experimentation to validate the proposed model. As for the simulator, we decided to use a commercial platform as Webots, which includes a virtual e-puck robot in a 3D environment.
Keywords: Evolutionary Robotics, Neural Networks, Modularity, Acetyl-Coline, chemical synapses, E-puck.

Article Info
Received May 8, 2021
Accepted Sep 12, 2021

1 Introduction

Evolutionary Robotics (ER) is a Robotics field where control and morphological components are set under artificial evolutionary pressure that optimises the desired components. Hence, ER is a field where software and hardware components are modelled. Robust solutions are obtained after the optimisation process, and the worst solutions are discarded.

The most popular object of optimization in this field are Artificial Neural Networks (ANNs) neural network controllers or neuro-controllers (Montes & Aldana [47], Loula, Gudwin & Queiroz [37], Mitri, Floreano & Keller [44] & [45]). Amongst other solutions that can be combined using this approach, it is possible to find tools such as Fuzzy Logic (Mendel [42], Jamil, Jalani & Ahmad [31]); Based Rules Classification Systems (Kodjabachian & Meyer [35]); and Bayesian Networks (Inamura, Inaba e Inoue [28], Calandra, Seyfarth, Peters & Deisenroth [9]).

The weights of the ANNs are codified in a straightforward manner to be optimised using Evolutionary Computation (EC) algorithms; a commonly used algorithm is a Genetic Algorithm (GA) (Nolfi, Bongard, Husbands & Floreano [50]). Therefore, robots sensors are the inputs of the neural controller and actuators are linked to neurons at the output layer. Mainly optimisation works on neural weights, but it is possible to evolve morphological components of the ANNs.

When a neuro-controller is optimised, the work realised by the EC algorithm is to search for a robust configuration, which facilitates robots to solve a particular task. Depending on the complexity of the task, neuro-controllers can be simple or elaborated selection mechanisms. This adds complexity to the search space of potential solutions.

It is widely known that when an ANN has to solve more than two classification or control tasks, the search stalls because of the complexity of search space. In other words, suppose we have two control tasks: A and B. The optimal weights to solve task A and task B are in different sections of the complete search space. The algorithm adjusts weights to find an optimal solution for task A, but the solution for task B is missed. Therefore, the solution for task A blocks the task B solution. In Artificial Vision, this problem is identified as Neuronal Blocking (Jordan, Jordan & Barto [30]). In ER, a similar problem is named Search Space Blocking. In the literature (Calabretta, Nolfi, Parisi & Wagner [8]), it has been identified that the task of cleaning environments produces this problem when robots have to deposit and collect garbage. Robots tend to collect garbage but are stuck depositing garbage when one single ANN controls both sub-tasks (collection and depositing).

Proposed solutions to this problem consider differentiating ANNS mechanisms to employ a divide and conquer strategy, which is similar to living beings dividing functions through organs, apparatus, and systems (Newman [48]). In particular, this solution can be roughly compared to division in the human brain, which cooperatively solves different tasks through different specific and neighbour regions. In humans, modularity is the product of evolution with specialised regions that can interact with each other (Cho & Shimohara [12]).

Modularity in the human brain is the product of large number of neurons compared to a limited number of connections, with 100,000 million neurons having 7,000 connections for neurons (Ballard [3]). Focused specialisation in particular regions of the human brain is the consequence of interactions between neurons and neighbour regions (Geschwind & Galaburda [21]). This propriety is fundamental for successful multi-task systems (Potter, Meeden & Schultz [53]).

In biology, modularity can be used as a form of organisation to describe simple organisms (Bolker [4]). In computer sciences, modularity is considered an operation that facilitates the decomposition of the search space where each part is as complex to explore as the set of the altogether components (Hart, Kammeyer & Belew [24]). Therefore, an ER module is considered a part of a whole solution (Khare, Yao, Sendhoff, Jin & Wersing [34]) where interaction mechanisms are a basic tool for improving artificial evolution.

The modularisation of a control system in ER depends on the complexity of tasks (Di Fernando, Calabretta & Parisi [14]). When robots have to solve a simple task, a modular control system may not be necessary. On the other hand, a complex control system has to be employed when a complex task is evolved. The possibility of finding a locking condition in the search space is increased with complex tasks and environments (Jacobs & Jordan [29]). It has been identified that this kind of interference happens at the beginning of the search process (Calabretta, Di Fernando, Parisi & Keil [6]).

Modularity in ER is based on implementing fixed ANNs as blocks that process sensorial information together with system values whereas other mechanisms work with action selection arbitration between different modules (Floreano, Dürr & Mattiussi [17]). Thus, we identified two possible approaches: internal and external modularity. Then, we divide ANNs in an internal way where groups of neurons of one ANN are forced to specialise for attending a specific part of the whole problem (internal modularity). On the other hand, more than one ANN control system can be used to solve a global task divided into sub-tasks (external modularity).

Our research proposes to test an external modules architecture that offers a solution for the locking search space problem. Our architecture was based on bio-inspired elements such as the neuro-modulation carried out by acetylcholine, chemical synapses, and Renshaw cells. To validate the proposed architecture, we developed two experiments for evolutionary robots where we compared three different architectures.

This paper is structured as follows. Section 1 introduces the problem and presents the purpose of our research. In section 2, some work related to the ANNs modular architecture is presented. Section 3 presents all bio-inspired concepts where our architecture is based. The details around our model are explained in section 4. In section 5, the methodology and experiments are described in section 5. Section 6 contains the results of the experimentation. Next, the discussion of our findings is presented in section 7. Finally, Section 8 shows the conclusion of our research.

2 Related Works

Initial research in modularity and ANNs optimises morphology and modules weights in the same evolutionary process (Gruau [22]). In this case, the architecture is based on parallelism, and results show that, as in nature, the evolutionary process tends to generate regularity in the topology. Later research shows that less complex structures have better performance than complex structures (NourAshrafoddin, Vahdat & Ebadzadeh [51]).
In terms of internal modularity, one of the initial models uses inhibitory and excitatory connections in order to produce competition between regions of one single ANN. Complex structures appear at the beginning of the evolutionary processes but tend to disappear (Cho [11]).

One of the most important research in internal modularity is proposed by Nolfi [49]. Five architectures for control systems of evolutionary robots are tested and compared in this work. The setup is based on an experiment for cleaning a squared arena using the Khepera robots. Each robot has to collect cylinders and deposit them outside the arena. In this work, five different ANNs topologies are tested (see figure 1): two-layered ANN; three-layered ANN; two-layered ANN with retro-connections; external modules ANNs (two ANNs in parallel); and the Emergent Modular Architecture (EMA), which is an internal module ANN.

For instance, the external modules ANNs is a divided control system where two nets compete for accessing the actuators every step of the simulation. In this case, no explicit mechanism mediates the competition between modules. On the other hand, the EMA is composed of two modules for each actuator; thus, output modules are groups of two neurons at the output layer. One neuron is used to codify motor velocity, and the other is the activation neuron. Therefore, two modules compete to access the associated actuator, and control is granted to the highest activation level module.

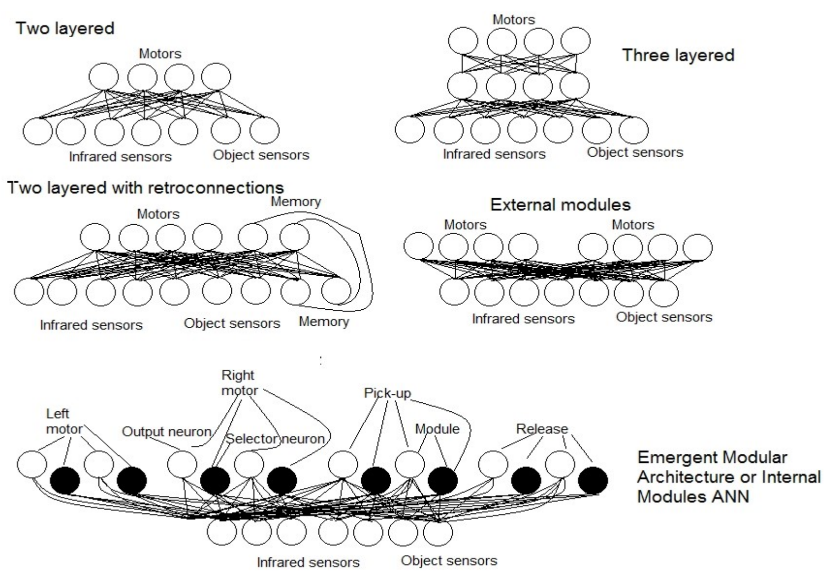

Fig. 1. Architectures tested by Nolfi [49]: A two-layered ANN, a three-layered ANN, a two-layered ANN with retro-connections, the external modules ANNs, and the internal modules ANN. The retro-connected ANN uses memory neurons to remember the last state. The external modules control is composed of two ANNs that compete to solve the task. The internal modules ANN is formed by two modules that compete for accessing the actuators.

Results show that the EMA has the best performance in terms of fitness and cylinders collected. The reason for these results is associated with the design of the ANN and the competition between modules. In addition, it has been showed that just one pair of modules is always active for selection. This constrains the variety of potential solutions to one group of neurons and does not reproduce what happens in the human brain where module monopolisation does not occur. Module specialisation is the consequence of the evolutionary pressure exercised by competition between modules. External module representation has average results, because two ANNs compete to access the resources and do not collaborate to achieve the general solution. In order to alleviate this situation, is necessary to include a mechanism that regulates competition and promotes a cooperative process.

For the first three architectures (a two-layered ANN, a three-layered ANN, and a two-layered ANNS with retro-connections), evolution does not optimise module selection, and individuals are not capable of collecting and depositing garbage. This evidences that the cleaning task presents a blocking in search space where the ANN has to be optimised to collect and deposit objects.

Further research supports the claim of the EMA having a better performance compared to an architecture based on the original EMA where a hidden layer and retro-connections are included (Ziemke, Carlsonn & Bodén [65]). The best competitor of the different implementations corresponds to the model with retro-connections and the reason is the capability of these connections to maintain information from previous steps (Ziemke [63]).

An important variation of the EMA is proposed using new evolutionary components (Calabretta *et al.* [7], which is named the Emergent Modular Architecture Based on Duplications (EMABD). Additionally, it has been used to explain evolutionary mechanisms for the evolution of modules. This architecture considers that modularity is included in the genetic information. Therefore, there are many available evolutionary methods for the optimisation of module morphology. One method employs assembler Encoding (Praczyk [54]), which tries to concentrate efforts on evolving module morphology instead of neural weights. Their results show that this methodology produces both simple and complex modules (Praczyk [55]).

As for the external modularity, one of the first implementations used sequential components, which means that the output one ANN is the input for a next stage ANN (Cao, Ahmadu & Shridhar [10]). A later approach implemented a control system for a four-legged robot based on external and sequential modularity (Manoonpong, Pasemann & Fischer [38]). Thus, two independent ANN modules work as the input for a third module that controls most of the actuators. As a result, a robust system controls an avoiding robot that has to explore the environment ((Manoonpong, Paseman & Roth [39]). Furthermore, the model is also implemented for multi-legged robots (Manoonpong, Paseman & Wörgötter [40]). Additionally, incremental neural evolution is used for an external modular system for controlling helicopters (Rice [56]).

3 Bio-inspiration in Evolutionary Robotics

Clearly, from its conception, ER is a highly bio-inspired algorithm. Computational optimisation processes tried to mimic evolution in nature. Therefore, solutions that are tested and do not have enough abilities to solve a task, tend to disappear, and they are replaced with new solutions. Then, solutions that have better adaption to the environment, and the task to be solved, have the highest possibilities to remain in the evolutionary process (Darwinian Evolution).

The use of ANNs adds an extra path to the research of bio-inspired methods that are highly related to neurosciences. In this way, an evolved neuro-controller represents an artificial nervous system (Gurkiewicz & Kornegreen [23]). Thereby, ER is considered a standard platform that facilitates proving and conducting experiments in areas such as neuroscience, and computation and biology (computational neurobiology), and their implementation is analogous to the evolution of the nervous systems (Floreano, Husbands & Nolfi [18]).

ER has the capability to generate computational models of neural components (Webb [61]). These models try to imitate the way nature solves problems that are optimised by billions of years through evolution (Mayer [41], Pasemann [52], Jeason & White [32], Fernando, Szathmary & Husbands [16]). For this reason, is very important to study and apply evolutionary mechanisms to ER (Husbands *et al.* [26]).

In terms of external modularity, the main problem consists in creating inter-modulator systems that provide an efficient solution in terms of which specialised module can solve a problem after is presented. In nature, chemical synapses and neurotransmitters are the solutions to this problem.

Neurons have two different ways of transmitting information: electrical and chemical. All models of ANNs replicate electrical synapses. Electrical synapses, in the brain, happens when two physically proximal neurons (pre-synaptic and post-synaptic) transmit an electrical current that generates a rapid voltage in form of a spike. On the other hand, chemical synapses happen when pre-synaptic neurons liberate a chemical substance (neurotransmitter) close to a post-synaptic neuron. There is not direct contact between axons, and there is a space between neurons named the synaptic cleft. Post-synaptic neuron captures molecules of neurotransmitter using receptors and then produces a depolarization. Chemical synapses do not produce an instantaneous effect on the post-synaptic neuron because chemical substances take time to travel between neurons. Neurotransmitters amplify their effects through an increasing number of receptors in the post-synaptic cells. In addition, a special mechanism recovers neurotransmitter molecules, which are not captured by the post-synaptic neurons. The aforementioned mechanism allows recovering the neuro-transmitter to convey further chemical processes in the post-synaptic neurons. Consequently, neurotransmitters regulate interactions between neurons and brain regions (Katz [33] & Spitzer [60]).

Different types of neurotransmitters are available in the brain: acetylcholine, dopamine, serotonin, noradrenalin, endorphin, and glutamate. Each of these has a particular effect under different brain regions and interact between them in particular cases. Many neurotransmitters are produced in the basal ganglia (neuronal groups that belong to the primitive brain); and the transportation mechanisms from production zones to stock zones are named pathways. When a neurotransmitter arrives to pre-synaptic neurons, specialised vesicles store them for posterior release purposes.

In the nervous system, the aforementioned substances have four characteristics that distinguish them from other similar substances (Seiggelbaum & Hudspeth [58]):

- Neurons produced them.
- They are stored in the presynaptic neurons, and then they are liberated for post-synaptic purposes creating a specific effect.
- In most cases, their effect can be replicated by administrating them in reasonable concentrations near the synaptic cleft (except for acetylcholine).
- There is a mechanism in the synaptic cleft for their recollection.

Under these conditions, is possible to create virtual neurotransmitters using ANNs models. Thus, virtual neurotransmitters can be used to communicate internal states, process status, or promote specific outputs in the ANNs (Smith & Philippides [59]). However, the most important application is the modulation of complex control systems for evolutionary robots (Husbands *et al.* [27]). A neurotransmitter that deserves particular attention is Acetylcholine (Ach), which modulates the action of motor neurons. Additionally, is considered a primitive neurotransmitter because is contained in all organisms around the world.

The composition of ACh is based on Choline and Acetylcoenzyme type A that is acquired from glycolysis consuming energy molecules. Catalization is made by choline acetyltransferase. Production centers are located in Renshaw neurons, Pedunculopontine nucleus, Meynert Basal Nucleus, and other basal ganglia nucleus in the cholinergic pathway. Acetylcholine molecules are captured by two types of receptors: Muscarinic and Nicotinic. The first type of receptors is associated with the inhibition of the parasympathetic system. The second type is associated with excitation of neurons in the muscular and skeletal apparatus, and involuntary movements of the heart. Nicotinic N1 receptors regulate synapses involuntary movement muscles. When acetylcholine is recaptured, a substance named Acetylcholinesterase assimilates those molecules of neurotransmitter that are not used in chemical synapses.

ACh is present in all organisms, but is produced exclusively by neurons. After humans eat, butryrylcholinesterase decomposes acetylcholine molecules to disallow excitatory effects in the human body that include involuntary movements. For this reason, acetylcholine is not based on food intake and cannot be ingested. Deficiencies of ACh are associated to Alzheimer and Parkinson. Nicotine is an agonist substance of ACh, which produces strong addiction in the human brain.

In renshaw neurons, ACh acts in an inhibitory way that promotes right movements and avoids excitation from motor neurons on the contrary muscles. When a motor neuron is excited, renshaw neurons, which are considered inter-neurons of the spinal cord (Aznavurian & Aguilar Rebolledo [2]), send inhibitory signals to lateral motor neurons and produce the desired movement. This mechanism is named lateral inhibition (Alvarez & Fyffe [1]). In other words, renshaw neurons allow muscle contraction when another one is extended (Kravos & Malesic [36]).

4 Proposed Architecture

The Acetyl-Modulated Architecture (AChMA) is composed of two blocks (see figure 2). The first block corresponds to all modules or Artificial Neural Networks, decomposing or dividing the proposed general problem. The second block is represented by an ANN that produces artificial levels of acetylcholine to other modules and is named network producer (similar to the basal ganglia).

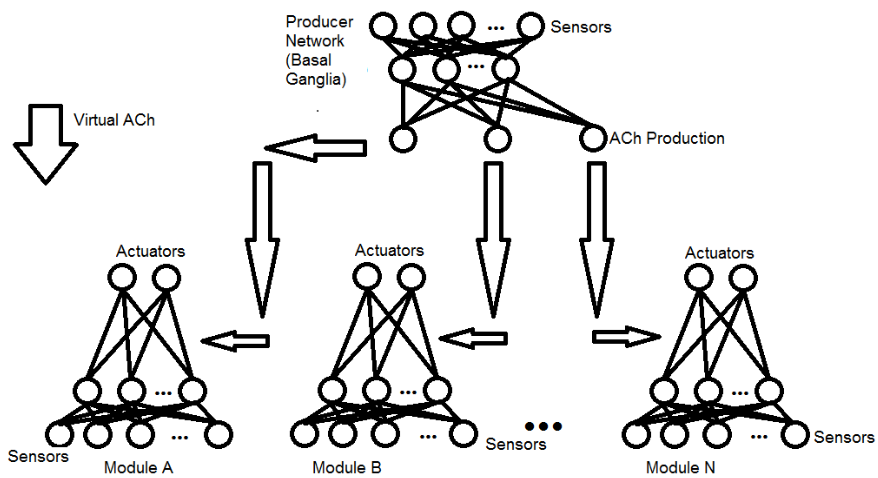

Fig. 2. Acetyl-Modulated Architecture. Ganglia groups of modules compete for accessing the motor system of an evolutionary robot. A producer network sends levels of virtual ACh to the inter-neurons of each module to produce synapses between neurons of the hidden layer and neurons of the output layer. Consequently, resource access and participation in the solution are regulated.

The network producer senses all variables of the environments where robots are situated. Additionally, this ANN receives the information of which module was binarily activated in the previous step (internal state sensor). Hence, when a module was active in the previous step, the internal state sensor receives a binary value of one. When the module is not active, the virtual sensor has a binary value of zero.

In order to improve discrimination in the producer network, a hidden layer was included. The number of neurons in this layer is less than neurons in the input layer, and more than the number at the output layer. Next, the output layer of the net is a virtual value of acetylcholine. This value is associated with each producer neuron, which is an activation neuron. Moreover, its function is to inhibit the Ach production of other modules by means of inhibitory connections. In this way, when a producer neuron reaches the maximum value of activation, the neuron sends an inhibitory signal to other producer neurons in an off-center off-surround manner. Therefore, only one of all producer networks send their artificial neuro-transmitter to their representative module. This mechanism is inspired by renshaw cells.

Virtual levels of ACh are sent to the interneurons (hidden layer) of each module. When a module receives the virtual substance, neurons in the hidden layer produce synapses (replicating a chemical synapsis) with the output neurons and produce changes on robot actuators. If inter-neurons do not receive virtual ACh, they cannot produce synapses between hidden and output layers.

On the other hand, modules networks are composed of sensor neurons at the input, hidden layer, and finally actuator neurons in the output layer. Each module is evolved independently and each one solves a part of the general problem (a divide and conquer strategy). In order to split the original problem into sub-problems, we used a division of the fitness function to reduce the influence of human design in the evolutionary process. After all modules are evolved, a final evolutionary process adjusts just the producer network leaving the rest of the modules intact. This evolutionary process uses all fitness function terms, which results in an alternative for reducing the influence of human designers during evolution (Frenken, Marengo & Valente [20], Frenken [19], and Weitzenfield [62]).

5 Methodology

In order to validate the proposed architecture, we designed two experiments where the performance of the architecture was compared with two control systems: a Single Net-Based Control (SNBC), which is one ANN that controls all robots operations, and the Emergent Modular Architecture (Nolfi [49]). The first experiment consists of a cleaning arena task where a single robot has to collect and deposit trash. The second experiment is a simple maze. Robots have to interchange the room where they are situated. The arena, right in the middle, has a door that is open before changing room.

In all cases, a generational Genetic Algorithm was used for weight optimisation of the artificial neural networks employed as robot controllers. The parameters utilised in the evolutionary process were 100 generations of 20 individuals, a mutation rate of 2%, one random crossover point, tournament selection, 80% of substitution, 20% individuals from the previous generation, and finally elitism (the best individual of the previous generation). Evolution run in a virtual environment modelled in Webots (Michel [43]). This simulator includes 3D models of various commercial robots and is possible to program controllers using programming tools such as Matlab, Java, C, and C++, among others.

The e-puck robot (Mondada et al [46]) was selected as the virtual platform for our experimentation. This robot is an open hardware platform, built for educational purposes; thus, users can add new hardware components. The robot is equipped with several actuators such as infrared sensors, light sensors, a VGA camera (640x480 pixels), step motors, a ring of LEDs, one microphone and a speaker, in its basic version gyroscope, Bluetooth communication, and a serial port. All devices are controlled by a TTL component, the dsPic30F6014A, which can be programmed using C or assembler language.

5.1 Experiment 1: An Environmental Cleaning Task

In this experiment, a robot has to clean a square arena delimited by four walls. In the original experiment, a Khepera robot uses its gripper to collect cylinders that represent trash. Thus, the robot deposits the cylinder outside of the arena. Taking into consideration the limitations and differences of the e-puck robot, we adapted the original experiment.

Our experiment cylinders were replaced by virtual garbage in a circular white target, and the walls were coloured in blue (see figure 3). The e-puck robot was rewarded with one garbage unit for each step that it remained in the white zone. Therefore, one point was accumulated every step that the robot spent at the white zone until the maximum load capacity of 200 units was reached. Additionally, for each step that the robot stayed in the black zone, it was equivalent to depositing one garbage unit. Furthermore, in the centre, the two target areas included a cylinder that was used as an additional visual aid for the robot.

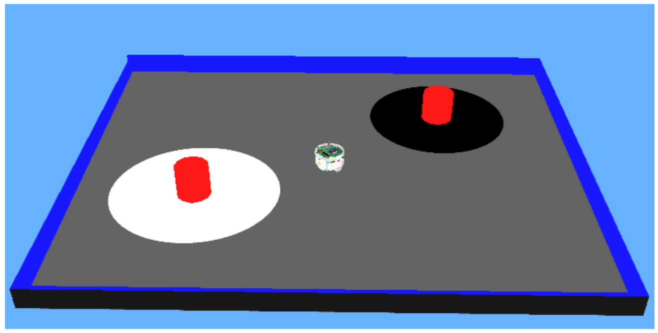

Fig. 3. Environmental setup for experiment 1. A robot is placed in a square arena delimited by four blue walls. There is a white circular target where the robot collects garbage and a black circular target where the virtual garbage is deposited.

In relation to the sensory system, the robot uses eight infrared sensors to avoid obstacles. Additionally, three infrared sensors located at the bottom of robot are used for locating target areas. When an obstacle are detected, IR sensors send a binary value of 1 (target detected); otherwise, the IR sends an output value of 0 (undetected target). These kinds of sensors were used to detect colour variations on the floor, which are white, grey, and black. When white colour is detected, the first sensor has a binary value of 1 and the other two sensors remain inactive. In the case that the gray color is detected, the second sensor has an active status, and the rest of the sensors have an inactive value. Finally, when black colour is detected the third sensor is active, and the rest are inactive. Next, a virtual garbage sensor was created to acquire information about the status of the garbage load. This sensor is a coefficient of the actual load of garbage divided into the maximum amount of garbage (200 U). Additionally, one point was added to the fitness function for each step that the robot remained at the black target until the minimum of zero units was reached.

The red and blue components of the VGA camera were used, and four sectors of 72 pixels were delimited. Thus, four binary virtual sensors were generated by each colour component. When the colour component was detected within a colour segment, the virtual colour sensor sent and active value; otherwise, sensors outputted an inactive value.

Three different types of control were evolved for this experiment. The first control system was a Single Network-Based Control (SNBC). It was represented by one ANN formed by 27 neurons, 20 neurons at the input layer, eight for the infrared sensors, three for the ground sensors, one for the virtual garbage sensor, four for the red VGA component, and four for the blue VGA component. In this experiment, and the rest of the experiments, the green VGA component was not used to avoid influence from synaptic connections related to the green colour. Particularly, because nothing is coloured in green, in the 3D scene, and these extra connections may unbalance the evolution of the control system. As for the neural topology, we added five neurons at the hidden layer, and two neurons in the output layer for controlling motors that encoded the robot's speed. The ANN had a feed-forward architecture and the sigmoid transfer function.

The second control was an example of internal modularity named the Emergent Modular Architecture and was composed of a total of 38 neurons. A the input layer 20 neurons, eight infrared sensors, three ground sensors, one garbage sensor, four red colour components, and four blue colour components; next, ten at the hidden layer; and eight at the output layer. Four modules were built at the output layer, and each one was composed of two neurons. One output neuron was used to codify motor speed and the other was used as an activation neuron. A pair of modules competed to control the right wheel; and the other pair of modules controlled the left wheel. The maximum level of the activation neurons was used to select, at every simulation step, which pair gain motor control. The implemented ANN had a feed-forward architecture and the sigmoid transfer function.

The third control system was a representation of external modularity (ACh modulated) composed of three ANNs: two modules and one producer. Modules were composed of a total of 32 neurons. 20 neurons formed the input layer: eight infrared sensors, three ground sensors, one garbage sensor, four VGA red components, and four VGA blue components; ten neurons at the hidden layer; and two neurons at the output layer. Next, the producer network was composed of 36 neurons in total. The input layer was formed by 22 neurons as follows:

eight infrared sensors, three ground sensors, one garbage sensor, four VGA red components, four VGA blue components; and as a form of short memory, one neuron for the previous binary state of the module A, and an extra one for the previous binary state of neuron B. Next, ten neurons are at the hidden, and four are at the output layer. The first module (module A) was optimised for controlling the collection task, and the second module (module B) was optimised for the depositing task. The fitness function is composed of two terms that correspond to modular division, and respectively reward the individual with one point for each unit of garbage collected and deposited.

Each control system was represented as an experimental group. Hence, each group had 18 replications of the evolutionary process where the best individual of each replication was used for experimentation in a post-evolution test. In order to find statistical differences between the experimental groups, we used a one-way ANOVA for ranks (Kruskal Wallis) and a post-hoc Student Newman-Keuls test. As the dependent variable, we used the amount of trash collected and deposited for a post-evolution test of 6 trails of 1000 steps. Next, the best chromosome of the last generation of the current evolutionary process was used. As for the independent variable, the type of control system was used.

5.2 Experiment 2: Interchange Room Task

The interchange room task is an experiment developed for robots to acquire the ability to solve mazes. Also, can be used to test the architecture with more modules. The arena was built with two square rooms delimited by a red wall with a door in the center (see figure 4). There was a rectangular black target area in the first room. This target was used as a switch that allowed opening the door between rooms. The door opened, after the robot stayed at least one step at the black rectangle. A surrounding white area was set below the red wall to add extra information about the proximity of the border wall to help and as a visual aid for finding the aisle between rooms.

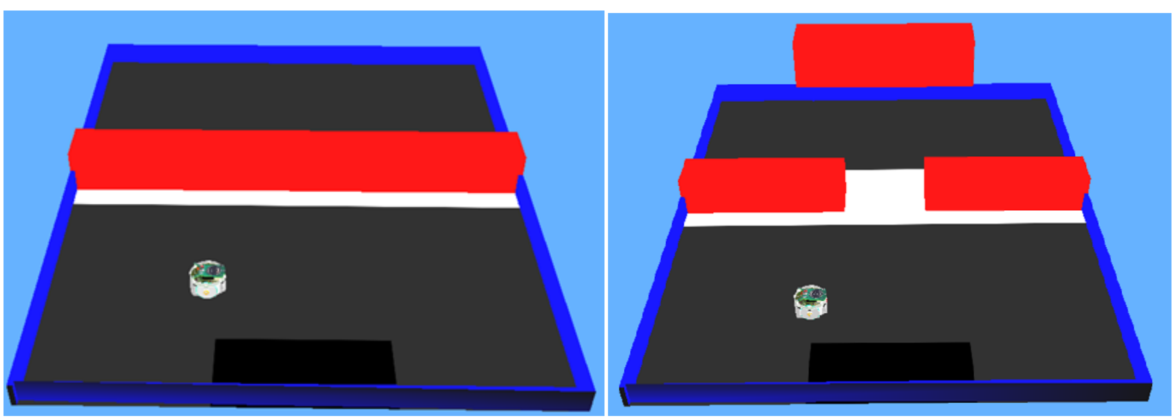

Fig. 4. An environment of experiment 2. A robot is placed in a room, and has to find a black rectangular target area to open the door. Then, the robot has to find aisle between rooms to move to the next room.

A robot with an initial random position had to find the black target, open the door and change to other room. The fitness function was rewarded with 3.0 points when the robot opened the door, 11.0 points when the robot found the aisle and 29.0 points when the robot changed the room.

Eight infrared sensors integrated the sensory system, three ground sensors, four VGA red colour components, and four VGA blue colour component sensors representing four sectors of 72 pixels. Additionally, three virtual sensors were created representing the actual state of the task (door open, aisle find, and change room). These sensors operated in a binary way and had an active value when the robot completed a partial task. Next, the motors were used as actuators. All ANNs used as control systems had a feed-forward architecture and the sigmoid transfer function.

Three ANNs control systems were tested as in experiment 1. A single control network was configured with one ANN composed of 35 neurons: 22 at the input layer (eight infrared sensors, three ground sensors, three task status sensors, four VGA red colour components, and four VGA blue component sensors); 11 neurons at the hidden layer; and 2 at the output layer that controlled the motor speeds.

One ANN integrated the Emergent Modular Architecture (internal modularity representation) with 41 neurons: 22 at the input layer (eight infrared sensors, three ground sensors, three task status sensors, four VGA red colour components, and four VGA blue colour component sensors); 11 neurons at the hidden layer; and eight at the output layer. Four output modules were built, two competing modules for controlling the right wheel and two competing modules for the left wheel.

The Acetyl-modulated control system (external modularity representation) was composed of four ANNs: three modules and one producer. Each module was composed of 35 neurons: 22 at the input layer (eight infrared sensors, three ground sensors, three task status sensors, four VGA red colour components, and four VGA colour blue component sensors); 11 neurons at the hidden layer; and two neurons at the output layer. The producer network was composed of 42 neurons: 22 neurons at the input layer, eight infrared sensors, three ground sensors, three task status sensors, four VGA red colour components, four VGA blue colour components, one neuron for storing the previous binary state of module A, one for storing the previous state of neuron B, and one for the previous state of neuron C. Next, 11 neurons at the hidden layer, and six neurons at the output layer.

The third control system is an example of external modularity (ACh modulated) composed of three ANNs: two modules and one producer. Modules were composed of 32 neurons: 20 at the input layer (eight infrared sensors, three ground sensors, one garbage sensor, four VGA red colour components, four VGA blue colour components); ten neurons at the hidden layer; and two neurons at the output layer. The producer network was composed of 36 neurons: 22 neurons at the input layer (eight infrared sensors, three ground sensors, one garbage sensor, four VGA red colour components, four VGA blue colour components, one neuron for storing the previous binary state of the module A, and one for the previous state of neuron B). Next, ten neurons at the hidden layer, and four neurons are at the output layer. The first module (module A) was optimised for controlling the collection task, whereas the second module (module B) was optimised for the depositing task. The fitness function was designated focusing on modular division by means of its fitness function components. Thus, module A was optimised for finding and opening the door, module B was optimised for finding the aisle, and module C was optimised for changing to the next room.

Three experimental groups represent the proposed control architectures. 18 replications of the evolutionary process integrated all experimental groups. The best individual of the last generation for each replication was tested. A one-way ANOVA for ranks (Kruskal Wallis) was used to find statistical differences, complemented with a post-hoc Student Newman Keuls test. A post-evolution test of six trials of 1000 steps measured the fitness levels of the individuals in each experimental group.

6 Results

6.1 Environmental Cleaning Task

The one-way ANOVA for Ranks analysis (see table 1) showed significant differences between all groups ($H_{(13.622)}$, $p=0.001$, 2df). Next, the post-hoc test revealed that all groups (ACh = 307.66 ± 33.812, Control = 133.33 ± 21.737, EMA = 204.00 ± 37.660) were different from each other ($p<0.05$).

Table 1. One-way ANOVA for ranks (Kruskal Wallis) of experiment 1.

	AChMA	SNBC	EMA
Median	307.660 *	133.330 *	204.00*
Mean + standard error	318.617± 33.812	133.579± 21.737	250.553 ± 37.660
Standard Deviation	143.453	92.223	159.777
H=13.622, p=0.001, n=18 *SNK comparison between groups			

The maximum expected fitness was 700 unities of collected (350 unities) and deposited garbage (350 unities). The ACh experimental group had the best fitness of the post evolution test (see figure 5). Individuals that belonged to this group had enough abilities for solving the task. In contrast, the fitness of individuals in the EMA group (able to solve the task) was lower concerning the ACh group but greater than the control group. The latter corresponds to the findings in the work of Nolfi [49] where the control group obtained a lower fitness compared with the EMA. The fitness levels also showed that the control group could not complete the garbage collection task during the evolutionary process. These individuals tended to learn how to collect garbage, but not to deposit it. The collection sub-task locked the posterior optimisation of the depositing sub-task.

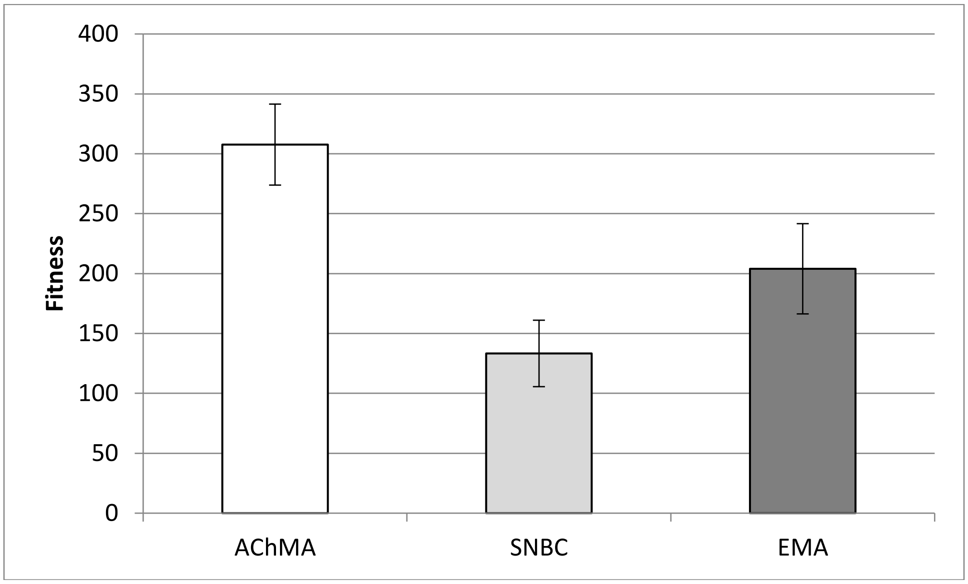

Fig. 5. The Median values and standard error in experiment 1. AChMA had the best performance, followed by the EMA and SNBC. The level of the SNBC group shows that robots are more efficient for collecting and depositing garbage.

6.2 Interchange Room Task

The one-way ANOVA on ranks (see table 2) exhibited significant between-groups differences ($H_{(13.890)}$, $p<=0.001$, 2df) for the interchange room task. This was verified with the posthoc test where was showed that the ACh group (14 ± 4.693) was statistically different ($p<0.05$) than the EMA (3 ± 2.238) and control (0 ± 0.79) groups.

Table 2. The one-way ANOVA for ranks (Kruskal Wallis) for experiment 2.

	AChMA	SNBC	EMA
Median	14*	0	3
Mean + standard error	21.994 ± 4.693	1.944 ± 0.79	5.056 ± 2.238
Standard deviation	19.910	3.351	9.496
H=13.890, p≤0.001, n=18 *SNK comparison between groups			

The highest possible score for this task was 43. The ACh experimental group had the best score, producing individuals able to open the door, find the corridor, and change room (see figure 6). Individuals in the EMA group were able to open the door and, in some cases move to the next room. Finally, individuals in the control group were not able to open the door and, consequently, could not change to the next room. In other words, the evolutionary process locked the search space for the developing of later sub-tasks.

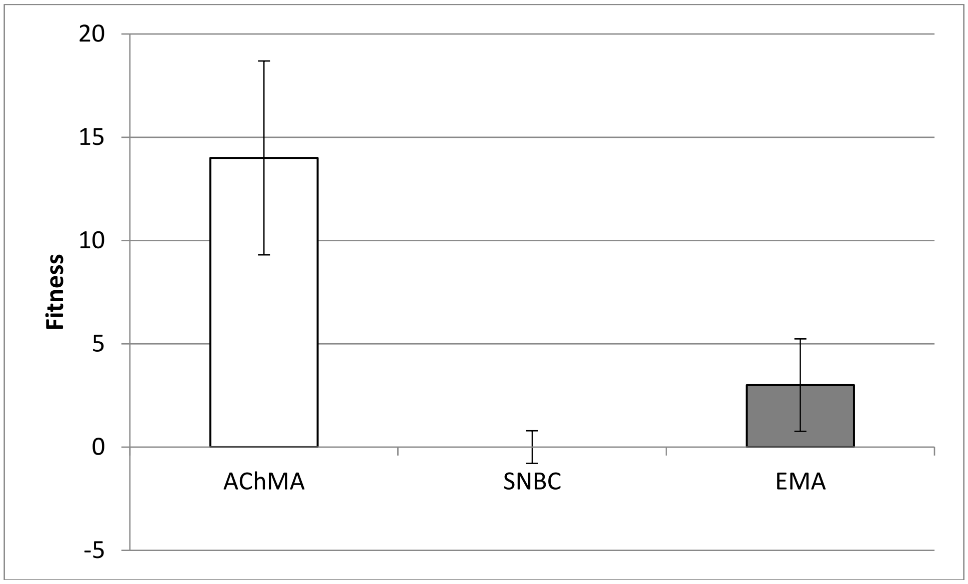

Figure 6. The Median and standard error values for experiment 2. In this experiment, the number of modules of the AChMA group increased. The best performance correspond to the AChMA group.

7 Discussion

7.1 The environmental Cleaning Task

The first experiment had the purpose to test and compare the optimisation of a task with a lock in the search space when using artificial evolution such as the environmental cleaning task. Therefore, we propose the development of the AChMA and two more control systems.

Our results confirm that the cleaning environmental task is a valid platform to study the problem of locked search spaces (Ziemke *et al.* [64]). A Single Network-Based Control has the worst performance with a median of 133 points. The SNBC had the best performance with robots that can collect an average number of garbage and deposit it in a lesser proportion. However, in the worst scenario, the robot neither collects nor deposits any garbage unit. The argument for such a poor performance is related to the complexity of the search space where the optimal weights for collection may be located in a different space in comparison to those of the depositing task (Jacobs *et al.* [30]).

On the other hand, the EMA produces specialised individuals for a single task with a median of 200 units. In this case, the evolutionary process obtains better individuals than in the SNBC. These results are consistent with reports from previous research (Nolfi [49]). The reason for this improvement is related to internal modularity.

Because each ANN is divided into internal modules, the discrimination level for each neuron at the output layer is relaxed. As a result, some competitive modules in the EMA monopolise decisions participating in all steps of the solution tasks. However, this characteristic leads to a locking condition of the search space that reduces the variety of the solutions.

The AChMA is a highly bio-inspired model, and its most important component is the ANN producer module. This module allows gradually controlling other modules. In other words, the virtual acetylcholine production avoids the monopolisation, of a particular resource, by a single module. Virtual ACh mimics the behaviour of ACh in nature; thus, its main effect is to control actuators similar to the movement of volunteer muscles. To facilitate good performance of the external modularity control systems is necessary, but not sufficient, to have a good inter-modular mechanism optimised by evolution. As shown in experiment 1, the evolutionary process of the AChMA model is decomposed into two parts: the first one where modules are optimised independently and the second one, where the ANN producer is optimised. The module search space is affected by the environmental configuration and sensory-motor system (Bongard [5]), and in general by the dimensionality and complexity of the search space (Huizinga, Mouret & Clune [25]). The simpler the modules, the less convoluted the search space is.

Hence, the AChMA models the external modularity function. It is common in the human brain that different regions or zones cooperate in the solution of a general problem. In our work, this characteristic facilitates searching subspaces of the overall problem. As a result, we are adding robustness to the space search using artificial evolution. Particularly, we use this characteristic to produce better solutions and complex control systems through evolution pressure (Corucci [13]). These modules, in ER, have the possibility to produce, organise, and represent thought, reasoning, and ultimately memory elements (Scheper, Tijmons, De Visser & De Croon [57]).

Next, the lateral inhibition used in the neural net, which is similar to the effect of Renshaw cells, is another important component of our architecture. This kind of inhibition guarantees that just one of the modules takes part at every step of the evolution. The latter is combined with the model of chemical synapses that inhibits, in the absence of virtual Ach, synapses from neurons of the hidden layer into neurons of the output layer.

The virtual substance in the model acts as a virtual neurotransmitter. Next, this substance is produced in specialised neurons and transported to pre-synaptic neurons (interneurons at the hidden layer). At certain levels of the neurotransmitter, the pre-synaptic neurons affect the post-synaptic neurons. At the end of each step, all levels of neurotransmitters are eliminated. In nature, ACh cannot be induced to an organism because is related to muscles and can cause noxious effects. The modelled Virtual ACh is exempt from these effects.

Our results confirmed that all the properties and characteristics of the bio-inspired AChM are combined to produce a solution that unlocks the search space. The performance of this architecture shows the best solution for this experiment. Robots are ready to collect and deposit garbage units during all the evolutionary processes. The use of modules facilitates the division of the general search space into simple search spaces. The two independent evolutionary processes optimise the weights of the two ANNs that execute the collection and depositing sub-tasks. The optimisation of the network producer becomes a decision problem where it acts as a switch for the modules participating in the solution of the general problem. In order to improve the performance, the producer has to choose the right action from the behavioural repertoire (collection, depositing, or no action).

7.2 The Interchange Room Task

With the aim to know if the AChMA is expandable to more than three modules, experiment 2 was developed. This experiment aims to analyse the modular capability of reducing the effects of locking the search space when the number of modules increases.

The interchange room task was discomposed into three modules related to three different sub-tasks: Open the door, look for the corridor, and swap rooms. Although, we use evolution for the optimisation, actions have to be executed in a specific sequence that adds an extra parameter for searching the space problem. The sequence is as follows, the robot has to find the black rectangular target area for opening the door, then to explore the initial room until it finds the white target area and locates the door, finally to cross the door and move to the next room. This sequence produces a sequential interference that in some cases causes the evolutionary algorithm to fail.

Our results show that the best performance is associated with the AChM, followed by the EMA and then the SNBC. In the latter, the evolutionary process produced individuals that we're able to solve, at least partially, the global task.

This result is significant because the interchanging room task presents a locking of the search space because of the added complexity and dimensionality of the ANN. Extra complexity is added because of the sequential nature of the task. Thus, individuals explore the arena, but cannot cross the corridor because the door is closed.

As a result, an optimal point of the search space is reached when individuals are able to open the door, though this does not imply that they are near the rest of the optimal points for completing the general task.

The interchange room task is a good experimental platform to study modular architectures and the locking search space problem. A single ANN did not produce adequate results; hence, to improve the results the control system is divided and used as a modular architecture. In the case of a modular control system, it is important to have the proper module selection for solving the general task.

The EMA produced better fitness results than the SNBC, but the evolutionary processes did not produce fitted individuals for solving the general task. Evolution spans individuals who can open the door but not complete the rest of the global task. The latter is the effect of module monopolisation that causes one particular module to take over the motor control most of the time. In the case of the cleaning environment task, the internal modules had more than one choice. Although there are three selection options in the interchange room task, sequential interference complicates selection evolution. Moreover, the EMA is affected by sequential interference. A potential solution for this problem is the use of incremental evolution where internal modules are evolved at different stages to facilitate robots learning the right sequential order. Therefore, new fitness terms are added (Faíña, Jacobsen & Risi [15]).

The AChMA results showed that most of the individuals are able to solve two-thirds of the task. Additionally, some individuals are able to solve the whole task. This improvement is due to the architecture of the external module, particularly the mentioned AChMA, which is able to avoid both the locking condition and sequential interference in the search space. Independent processes isolate chunks of the sequence that are evolved to solve in a partial way the general task. In experiment 2, individuals in the first module are evaluated for their abilities to find the black target area and open the door. The initial conditions for the second module start with the door open, thus evolution focuses on evolving the task of locating the door. As for the third module, individuals focus on moving across the nearby corridor. Then, the producer network is optimised for module selection in both the SNBC and EMA systems.

Finally, the AChMA demonstrated that is able to overcome the locking search space problem, and avoid sequential interference. Furthermore, this expandable architecture allows the addition of extra modules.

8 Conclusions

The Acetyl-Modulated Architecture (AChMA) offers a good solution for the search space-locking problem. This solution uses a divide and conquers strategy and modular division derived from the global fitness function equation. The AChMA roughly replicates living organisms because regions, organs, apparatus, and systems can be considered modules. Control on this architecture facilitates searching the general search focusing on smaller sub-spaces guided through the included elements in the fitness function. In other words, the AChMA represents a re-dimension of the search space. Therefore, each module represents a part of the general problem. Furthermore, an important component is the producer network that regulates the interaction between modules to solve the main task. A virtual neurotransmitter is included to regulate module selection.

As for the Environmental cleaning task, our experiments presented a comparison between the SNBC, EMA, AChMA architectures. This task presented a locking in the search space produced by the collection and garbage deposit tasks. The AChMA architecture overcame the locking condition using modular components and a robust selection system.

Finally, the interchange room experiments showed that the AChMA preserved its functionality even though the number of modules increased. A decompositional task methodology facilitated the optimisation of independent modules, and at a later stage, the producer network was evolved for producing the right module interaction.

9 References

[1] Alvarez, F.J., & Fyffe, R.E. (2007). The continuing case for the Renshaw cell. The Journal of physiology, 584(1), 31-45.

[2] Aznavurian, A., & Aguilar-Rebolledo, F. (2006). Espasticidad ¿Qué es y qué no es? Nuevos Horizontes en la Restauración Neurológica. Plasticidad & Restauración Neurológica, 5(2).

[3] Ballard, D.H. (1986). Cortical Connections and Parallel Processing. Structure and Function. The Behavioral and Brain Science, 9(1). 67-90.

[4] Bolker, J.A. (2000). Modularity in Development and Why It Matters to Evo-Devo. American Zoologist, 40(5), 770-776.

[5] Bongard, J. (2015). Using robots to investigate the evolution of adaptive behavior. Current Opinion in Behavioral Sciences, 6, 168-173.

[6] Calabretta, R., Di Fernando, A., Parisi, D., & Keil, F.C. (2008). How to learn multiple tasks. Biological Theory, 3(1), 30-41.

[7] Calabretta, R., Nolfi, S., Parisi, D., & Wagner, G.P. (1998). Emergence of Functional Modularity in Robots. From Animals to Animats, 5, 497-504.

[8] Calabretta, R., Nolfi, S., Parisi, D., & Wagner, G.P. (1998). A case of the evolution of modularity: towards a bridge between evolutionary biology, artificial life, neuro- and cognitive science. Proceedings of Sixth International Conference on Artificial Life, (6), 275-284.

[9] Calandra, R., Seyfarth, A., Peters, J., & Deisenroth, M.P. (2016). Bayesian optimisation for learning gaits under uncertainty. Annals of Mathematics and Artificial Intelligence, 76(1-2), 5-23.

[10] Cao, J., Ahmadi, M., & Shridhar, M. (1997). A Hierarchical Neural Network Architecture for Handwritten Numeral Recognition. Pattern Recognition, 30(2), 289-294.

[11] Cho, S. (1997). Combining Modular Neural Networks Developed by Evolutionary Algorithm. (1997). In IEEE International Conference on Evolutionary Computation 1997, 647-650.

[12] Cho, S.B., & Shimohara, K. (1997). Emergence of Structure and Function in Evolutionary Modular Neural Networks. In Proceedings of Fourth European Conference on Artificial Life, 197-204.

[13] Corucci, F. (2017). Evolutionary Developmental Soft Robotics: towards adaptive and intelligence soft machines following nature's approach to design. In Soft Robotics: Trends, Applications and Challenges, 111-116. Springer International Publishing.

[14] Di Fernando A., Calabretta, R., & Parisi, D. (2001). Evolving Modular Architectures for Neural Networks. Proceedings of the Sixth Neural Computation and Psychology Workshop: Evolution Learning and Development, 253-262. Springer Verlag.

[15] Faíña, A., Jacobsen, L.T., & Risi, S. (2017). Automating the incremental Evolution of Controllers for Physical Robots. Artificial Life.

[16] Fernando, C.T., Szathmary, E., & Husbands, P. (2012). Selectionist and evolutionary approaches to brain function: a critical appraisal. Frontiers in computational neuroscience, 6, 24.

[17] Floreano, D., Dürr, P., & Mattiussi, C. (2008). Neuroevolution: From architectures to learning. Evolutionary Intelligence, 1(1), 47-62.

[18] Floreano, D., Husbands, P., & Nolfi, S. (2008). Evolutionary Robotics. In Springer handbook of robotics, 1432-1451. Springer Berlin Heidelberg.

[19] Frenken, K. (2006). A fitness landscape approach to technological complexity, modularity, and vertical disintegration. Structural Change and Economic Dynamics, 17(3), 288-305.

[20] Frenken, K., Marengo, L., & Valente, M. (1999). Interdependencies near-decomposibility and adaption. Computational techniques for modeling learning in economics, 209-244.

[21] Gecshwind, N., & Galaburda, A.M. (1985). Cerebral Lateralisation: Biological Mechanism, Association, and Pathology: I.A. hypothesis and a program for search. Archives of neurology, 42(5), 428-459.

[22] Gruau, F. (1994). Automatic Definition of Modular Neural Networks. Adaptive Behavior, 3(2), 151-183.

[23] Gurkiewicz, M., & Kornegreen, A. (2007). A numerical approach to ion channel modelling using whole-cell voltage-clamp recordings and genetic algorithm. PLoS Comput Biol, 3(8), e169.

[24] Hart, W.E., Kammeyer, T.E., & Belew, R.K. (1994). The role of development in genetic algorithms. Doundations of genetic algorithms, 3, 315-332.

[25] Huizinga, J., Mouret, J.B., & Clune, J. (2016). Does Aligning Phenotypic and Genotypic Modularity Improve the Evolution of Neural Networks? In Proceedings of the 2016 on Genetic and Evolutionary Computation Conference, 125-132. ACM.

[26] Husbands, P., Moioli, R., Shim, Y., Philippides, A., Vargas, P., & O'shea, M. (2014). Evolutionary robotics and neuroscience. The horizons of evolutionary robotics, 17-63.
[27] Husbands, P., Smith, T., O'shea, M., Jakobi, N., Anderson J., & Philippides A. (1998). Brains, Gases and Robots. In Proceeding ICANN, (98).
[28] Inamura, T., Inaba, M., e Inoue, H. (2000) User adaption of human-robot interaction model based on Bayesian network and introspection of interaction experience. In Proceedings of the 2000 IEE/JRS International Conference on Intelligent Robots and Systems, (3), 2139-2144.
[29] Jacobs, R. & Jordan, M. (1992). Computational Consequences of a bias toward short connections. Journal of Cognitive Neuroscience, 4(4), 323-335.
[30] Jacobs, R., Jordan, M., & Barto, A. (1991). Task decomposition through competition in a modular connectionist architecture: The what and where vision task. Cognitive Science, 15(2), 219-250.
[31] Jamil, M.F.A., Jalani, J., & Ahmad, A. (2016). A new approach of active compliance control via fuzzy logic control for multifingered robot hand. In First International Workshop on Pattern Recognition, (10011), id 1001111 9.
[32] Jeason, F., & White, A. (2012). Evolving Axonal Delay Neural Networks for Robot Control. In Proceedings of the 14th annual conference on Genetic and Evolutionary Computation, 121-128. ACM.
[33] Katz, B. (1969). The release of neural transmitter substances. Liverpool University Press.
[34] Khare, V., Yao, X., Sendhoff, B., Jin, Y., & Wersing, H. (2005). Co-evolutionary Modular Neural Network for Automatic Problem Decomposition. In Proceedings of the 2005 IEEE congress on Evolutionary Computation, (3), 2691-2698.
[35] Kodjabachian, J., & Meyer, J. (1995). Evolution and Development of Control Architectures in Animats. Robotics and Autonomous Systems, 16(2-4), 161-182.
[36] Kravos, M., & Malesic, I. (2017). The Role of Glutamate Dehydrogenase Activity in Development of Neurodegenerative Disorders. World Journal of Neuroscience, 7, 181-192.
[37] Loula, A., Gudwin, R., & Queiroz, J. (2010). On the emergence of indexical and symbolic interpretation in artificial creatures, or What Is this I Hear? In ALIFE 2010, 862-868.
[38] Manoonpong, P., Pasemann, F., & Fischer, J. (2005). Modular Neural Control for a Reactive Behavior of Walking Machines. IEEE International Symposium on Computational Intelligence in Robotics and Automation 2005, 403-408.
[39] Manoonpong, P., Paseman, F., & Roth, H. Modular Reactive Neurocontrol for Biologically-Inspired Walking Machines. The International Journal of Robotics Research, 26(3), 301-331. 2007.
[40] Manoonpong, P., Paseman, F. & Wörgötter, F. (2008). Sensor-driven neural control for omnidirectional locomotion and versatile reactive behaviors of walking machines. Robotics and Autonomous Systems, 56(3), 265-288.
[41] Mayer, H. (2011). Evolution of Robotic Neurocontrollers with Intrinsic Noise and their Behavior in Noisy Environments. In the 2011 International Joint Conference on Neural Networks (IJCNN), 1975-1980. IEEE.
[42] Mendel, J. Fuzzy Logic Systems for Engineering: a tutorial. (1995). Proceedings of the IEEE Transactions on Fuzzy Systems, 83(3), 345-377.
[43] Michel, O. (1998). Webots: Symbiosis between virtual and real mobile robots. In International Conference on Virtual Works, 254-263. Springer, Berlin, Heidelberg.
[44] Mitri, S., Floreano, D., & Keller, L. (2009). The evolution of information suppression in communicating robots with conflict of interests. Proceedings of the National Academy of Sciences, 106(37), 15786-15790.
[45] Mitri, S., Floreano, D., & Keller, L. (2010). Relatedness influences signal reliability in evolving robots. Proceedings of The Royal Society of London B: Biological Science, (278), 378-383.
[46] Mondada, F., Bonani, M., Raemy, X., Pugh, J., Cianci, C., Klaptocz, A., Magnenat, S., Zufferey, J.C., Floreano, D., & Martinoli, A. (2009). The e-puck robot, designed for education in engineering. In Proceedings of the 9th conference on autonomous robot systems and competitions, (1), 59-65. IPCB: Instituto Politectino de Castelo Branco.
[47] Montes-Gonzalez, F., & Aldana-Franco, F. (2011). The evolution of signal communication for the e-puck robot. Advances in Artificial Intelligence, 466.477.
[48] Newman, M. (2006). Modularity and community structure in networks. Proceedings of the National Academy of Science, 103(23), 8577-8582.
[49] Nolfi, S. (1997).Using emergent modularity to develop control systems for mobile robots. Adaptive Behavior, 5(3-4), 343-363.
[50] Nolfi, S., Bongard, J., Husbands, P., & Floreano, D. (2016). Evolutionary Robotics. In Springer Handbook of Robotics, 2035-2068. Springer International Publishing. Springer-Verlag.

[51] NourAshrafoddin, N., Vahdat, A., & Ebadzadeh, M. (2007). Automatic Design of Modular Neural Networks Using Genetic Programming. Artificial Neural Networks-ICANN 2007, 788-798.
[52] Pasemann, F. (2013). Self-regulating Neurons in the Sensor motor Loop. In International Work-Conference on Artificial Neural Networks, 481.491. Springer Berlin Heidelberg.
[53] Potter, M., Meeden, L., & Schultz, A. (2001). Heterogeneity in the Coevolved Behaviors of Mobile Robots: The Emergent of Specialists. In Proceedings of International joint conference on artificial intelligence, (17), 1337-1343. Lawrence Erlbaum Associates LTD.
[54] Praczyk, T. (2008). Modularity networks in Assembler Encoding. Computational Methods in Science and Technology, 14(1), 27-38.
[55] Praczyk, T. (2013). Modularity and Regularity in Neural Networks Produced with Assembler Code. Computational Methods in Science and Technology, 19(3), 145-155.
[56] Rice, S. (2000). The evolution of developmental interactions: epitasis, canalisation and integration. In Epitasis and The Evolutionary Process, 82-98.
[57] Scheper, K. Y., Tijmons, S., De Visser, C. C., & De Croon, G.C. (2016). Behavior trees for evolutionary robotics. Artificial life.
[58] Seiggelbaum, S.A., & Hudspeth, A.J. (2000). Principles of neural science. E.R. Kandel, J.H. Schwart, & T.M. Jessell (eds). New York: McGraw-Hill.
[59] Smith, T. & Philippides, A. (2000). Nitric Oxide in Real and Artificial Neural Networks. BT Technology Journal, 18(4), 140-149.
[60] Spitzer, N.C. (2017). Neurotransmitter Switching in the Developing and Adult Brain. Annual Review of Neuroscience, (0).
[61] Webb, B. (2009). Animals versus animats: or why not model the real iguana? Adaptive Behavior, 17 (4), 269-286.
[62] Weitzenfield, A. (2000). From Schemas to Neural Networks: A Multi-level Modeling Approach to Biologically-Inspired Autonomous Robotic Systems. Robotics and Autonomous Systems, 56(2), 177-197.
[63] Ziemke, T. (2000). On 'Parts' and 'Wholes' of Adaptive Behavior: Functional Modularity and Diachronic Structure in Recurrent Neural Robot Controllers. From Animals to Animats, 6, 171-180.
[64] Ziemke, T., Bergfeldt, N., Buason, G., Susi, T., & Svensson, H. (2004). Evolving Cognitive Scaffolding and Environment Adaption: A new research direction for Evolutionary Robotics. Connection Science, 16(4), 339-350.
[65] Ziemke, T., Carlsonn, J., & Bodén, M. (1999). An Experimental Comparison of Weight Evolution in Neural Control Architectures for a 'Garbage-Collecting' Khepera Robot. In Proceedings of the First International Khepera Workshop, 31-40.