

# We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

6,900

Open access books available

186,000

International authors and editors

200M

Downloads

Our authors are among the

154

Countries delivered to

TOP 1%

most cited scientists

12.2%

Contributors from top 500 universities



WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index  
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?  
Contact [book.department@intechopen.com](mailto:book.department@intechopen.com)

Numbers displayed above are based on latest data collected.  
For more information visit [www.intechopen.com](http://www.intechopen.com)



---

# Autonomous Robots and Behavior Initiators

---

Oscar Chang

Additional information is available at the end of the chapter

<http://dx.doi.org/10.5772/intechopen.71958>

---

## Abstract

We use an autonomous neural controller (ANC) that handles the mechanical behavior of virtual, multi-joint robots, with many moving parts and sensors distributed through the robot's body, satisfying basic Newtonian laws. As in living creatures, activities inside the robot include behavior initiators: self-activating networks that burn energy and function without external stimulus. Autonomy is achieved by mimicking the dynamics of biological brains, in resting situations, a default state network (DSN), specialized set of energy burning neurons, assumes control and keeps the robot in a safe condition, where other behaviors can be brought to use. Our ANC contains several kinds of neural nets trained with gradient descent to perform specialized jobs. The first group generates moving wave activities in the robot muscles, the second yields basic position/presence prediction information about sensors, the third acts as timing masters, empowering sequential tasks. We add a fourth category of self-activating networks that push behavior from the inside. Through evolutive methods, the composed network share clue information along a few connecting weights, producing self-motivated robots, capable of achieving noticeable self-level of competence. We show that this spirited robot interacts with humans and, through appropriate interfaces, learn complex behaviors that satisfy unknown, sub-jacent human expectative.

**Keywords:** autonomous robot, behavior initiators, deep learning

---

## 1. Introduction

To reach the degree of complexity required in human-robot interaction, artificial neural networks (ANNs) seem to be good nominees. The new powerful training algorithms known globally as deep learning have made possible massively trained ANN capable of recognizing a specific human face in a blink. However, these powerful neural processors lack a key component of life: self-motivation. What internal force moves a fruit fly? Important research [1, 2]

has found that in ultimate navigating lifesaving situations, the decisions in the fly's brain, with about 250,000 neurons, are taken by a reduced set of neurons that consume energy and originate an inner noisy output that in turn fires a massive body response (change in flying direction, for instance). So, at this scale, by using its own onboard neural processor, a self-motivated behavior initiator situation occurs inside the fly, causing noticeable changes in the activity of the individual.

As a significant consequence, this internal capacity converts the fly into a free-running autonomous living creature. One of our objectives in the chapter is thus to develop design methods that bring this genuine spontaneity and autonomy to our robots.

At the bigger scale of human brains with about eighty billion of neurons, the function of autonomous behaviors initiator is a much more elaborated matter, well documented by Raichel and its research team by using modern functional magnetic resonance imaging (fMRI) [3, 4]. From these studies, one noticeable finding is that the human brain never really rests but stays always in constant activity, burning a substantial amount of energy that seems to go nowhere. Raichel called this phenomenon "the brain dark energy," and his discovery changes every previous concept about brain functioning. This energy-burning attitude seems to be the common way of living brains, and signs of constant burning energy have been reported in bees [5] and submillimeter worms [6].

### 1.1. Previous works

The use of artificial neural nets to control robots represents a promising activity, and recent research has been published. In [7], the authors develop an autonomous robot with the application of neural network and apply it for monitoring and rescue activities in case of natural or man-made disaster. In [8], the use of an artificial neural network to improve the estimation of the position of a mobile node in indoor environment using wireless networks is studied. In [9], the author focuses on deep convolutional neural networks, capable of differentiating between thousands of objects by self-learning from millions of images. In [10], the authors study the design of a controlling neural network using adaptive resonance theory. In [11], the authors developed a new method based on neural networks that allows learning multichain redundant structure configuration during grasping.

In our previous work, we proposed a method where the capacities of two specific kinds of neural processors [12], vehicle driving and path planning, were stacked as to control mobile robots. Each processor behaves as an independent trained agent that, through simulated evolution, is encouraged to socialize through low-bandwidth, asynchronous channels. Under evolutive pressure, agents develop communication skill and cooperative behaviors that raise the level of competence of vision-guided mobile robots, allowing a convenient autonomous exploration of the environment. In [13], a neural behavior-initiating agent (BIA) was proposed to integrate relevant compressed image information coming from other cooperating and specialized neural agents. Using this arrangement, the problem of tracking and recognizing a moving icon was solved by three simpler and separated tasks. Neural agents associated proved to be easier to train and show a good general performance. The obtained neural

controller handled spurious images, solved acute image-related tasks, and, as a distinctive feature, in prolonged deadlock conditions showed traces of genuine spontaneity.

In this chapter, we have taken these ideas further and propose an all-neural controller specialized in governing the functioning of a multi-joint robot, with many joints, muscles, sensors, and specialized sub-processors. The general problem is to find an ideal balance that guarantees self-motivation and maximizes the learning capacity of the robot in human-robot interaction situations.

Our methodology contemplates the partially supervised training with backpropagation of shallow networks inside explicit scenarios, with specialized tasks, where information about the environment is available and is used as targets for local neural training. The objective at this basal level is to produce reliable abstract representations of the environment, including both short-term and long-term influences and wave-time-related information. This neural set is then stacked together with other internal and external neural signals, creating a fertile ground for the robot to learn new behaviors related with human's interaction. Our macro-objective is to build robots supported by self-motivated, multipart, robust neural controllers.

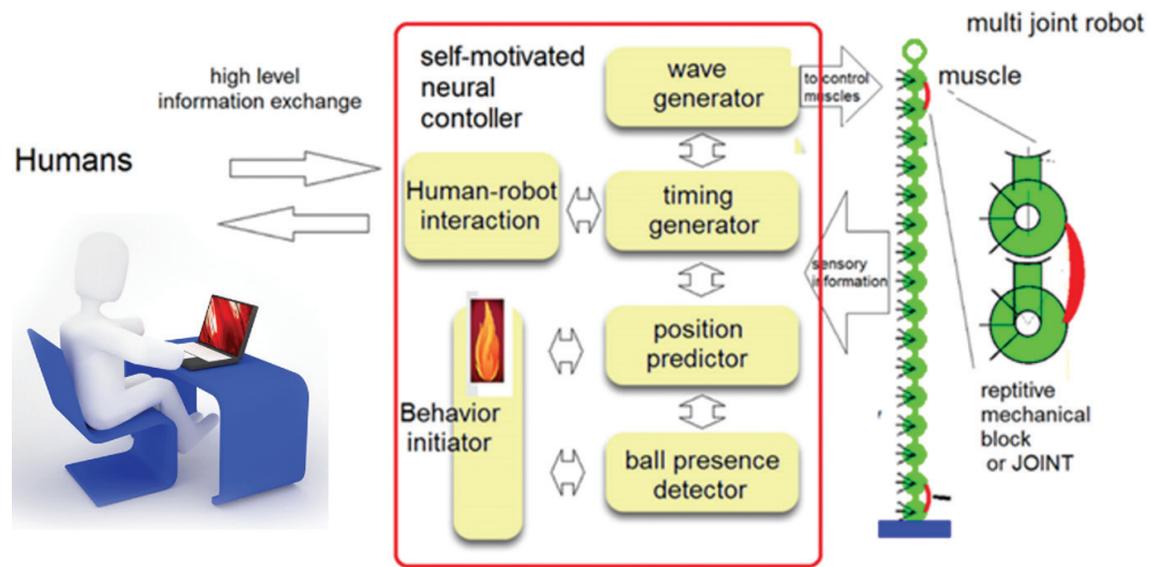
## 1.2. Research methods

We have constructed neural models written in C++ that behave or can be trained to behave as different kinds of neural sub-processors including self-activated behavior initiators, wave generator, timing generator, and general purpose predictive units. We also develop C++ model for an expandable mechanical universe where neuro-mechanical nodes composed by muscles, sensors, joints, mechanical structures, and mechanical links can be connected together, creating wormlike robots extrapolable to many components. The robot universe includes other items such as ball, floor, fixed walls, and one flexile moving wall that can be manipulated by humans.

Through evolutive methods, the neural subcontroller learns to share clue information along a few low-bandwidth channels producing a self-motivated robot with a high level of competence. We show that this proactive robot is capable of interacting with humans through appropriate interfaces and learning complex behaviors that satisfy unknown, subjacent human purposes.

## 1.3. Autonomous neural controller

The proposed self-activated neural controller is developed around the ambient in **Figure 1**. The mechanical assembly is defined by a set of repetitive, neural-mechanical blocks called joints, snapped together to form long chains. The wave generator block is a shallow network directly connected to the actuators, one neuron per muscle. Its function is to massively move the muscle in a coordinate way. The timing generator, position detector, and ball detector blocks are all shallow, three-layer neural networks, trained with backpropagation to do robotic tasks related to sensor activities. The behavior initiator block is an energy-consuming network that satisfied a local, weight-encoded syntactic rule. By evolutive algorithm and through a



**Figure 1.** Autonomous neural controller. Modular neural-mechanical blocks called joints are snapped together to form long chains. The resulting structure is controlled by an arrangement of stacked neural controllers. A wave generator massively moves robot muscle in a coordinate way. Other shallow neural networks are trained with backpropagation do specific robotic tasks: handle sensor activities, timing generation, position detection, and ball sensing. The behavior initiator is an energy-consuming, self-activating network that satisfied syntactic rules and pushes behaviors by itself. The genetic combination of all these elements produces a self-motivated robot capable of learning, through human-robot interaction, behaviors that satisfy human's expectations.

convenient interface, human-robot interaction triggers a learning process where some weights are modified, and the robot learns behaviors that satisfy human's expectations. After training, the behavior initiator network behaves as a default mode network (DMN) that assumes the control, burns energy, and uses other subjacent resources to initiate new behaviors, if required.

## 2. Biological brains

### 2.1. Brain's wiring diagram

The basic building block of brains is the neuron, which by itself has a very especial nature in terms of energy consumption, higher than any other kind of cell in living creatures [14]. From human to rotifers and very simple worms, neurons group themselves into elaborated networks called brains, where the common factor seems to be carefully knitted structuring complexity, with high job specializations [15, 16].

### 2.2. A default mode of brain function

"Whilst part of what we perceive comes through our senses from the object before us, another part (and it may be the larger part) always comes out of our own head." William James (1890)

In classical studies of brain function, the main accepted model is based in task-evoked responses. In general, the used experiments encourage a reflexive view of how the brain

works, ignoring that brain functions may be mainly intrinsic, connecting by themselves information and processing it to respond to environmental demands. By carefully analyzing the allocation of the brain's energy resources, Raichel [4] argues that the essence of brain function is indeed mainly intrinsic and components of signal transduction and metabolic pathways are all in a continuous state of flux.

Consider this functional aspect of human brain, described in Raichel's research [3, 4]:

*"In the mid-1990s we noticed quite by accident that, surprisingly, certain brain regions experienced a decreased level of activity from the baseline resting state when subjects carried out some task. These areas—in particular, a section of the medial parietal cortex (a region near the middle of the brain involved with remembering personal events in one's life, among other things)—registered this drop when other areas were engaged in carrying out a defied task such as reading aloud. Befuddled, we labeled the area showing the most depression MMPA, for "medial mystery parietal area. This cuing — among the visual and auditory parts of the cortex, for instance—probably ensures that all regions of the brain are ready to react in concert to stimuli. Further analyses indicated that performing a particular task increases the brain's energy consumption by less than 5 percent of the underlying baseline activity. A large fraction of the overall activity—from 60 to 80 percent of all energy used by the brain—occurs in circuits unrelated to any external event."*

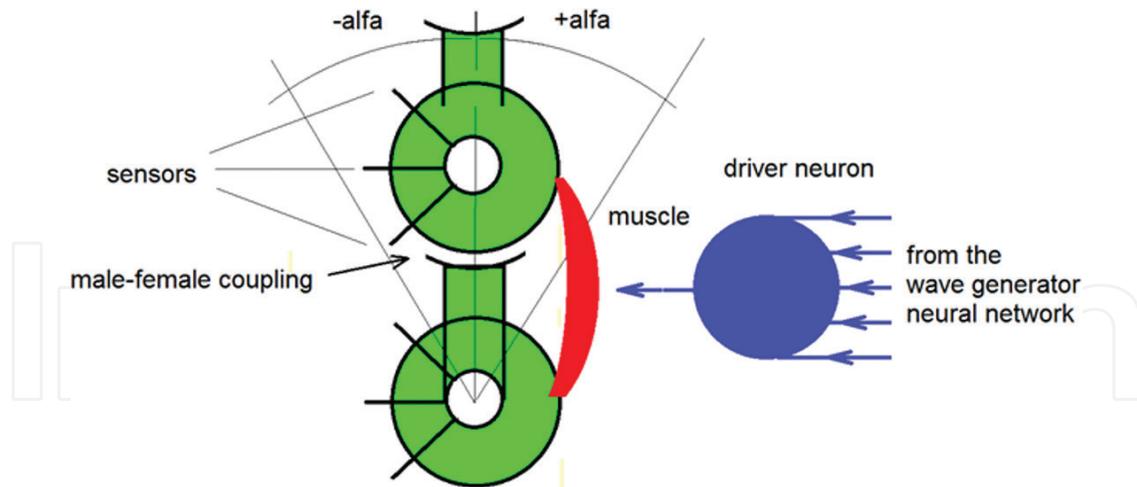
According to [3, 4], the human brain has a default mode of function controlled by a default mode network (DMN) which serves as a master organizer of its dark energy. The DMN is thought to behave like an orchestra conductor, issuing timing signals, much as a conductor waves a baton, to coordinate activity among different brain regions. This orchestrated way of doing things is described in a neat story in [4] where during a quiet beach afternoon a placid tourist does daydreaming watching nowhere. In his lap rests a magazine that he's been reading for a while, suddenly a weird looking insect lands in its naked leg, firing a cascade of stimulus. The point is that during the following chains of events, where the human tries to get rid of a potential danger, the brain in fact consumes less energy during daydreaming. Raichel found that the default mode network burns energy and maintains the control of the whole body, while many other powerful neural processors (vision, sense of touch, etc.) return to the borderline of activity and keep on burning energy, ready to actuate.

The lesson about this biological brain story is that to survive in a complex physical world, our robots and robot controllers should have a safe default mode that keeps itself in charge, burns energy, preserves the mechanical structure in a safe condition, and is ready to evocate other behaviors under stimuli.

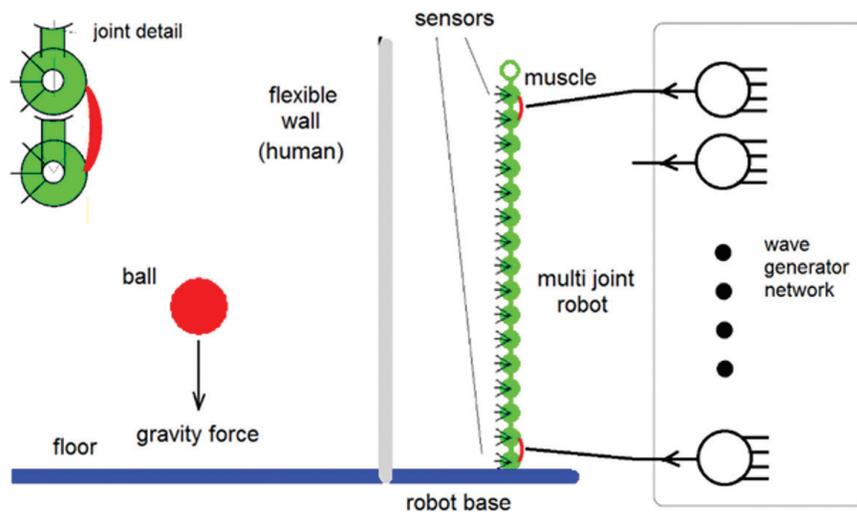
### 3. The robot and its environment

The robot is assembled with elements that contain sensors, muscles, rigid joints, and a male-female coupling (**Figure 2**). Each joint has a dedicated neuron that activates the corresponding muscle which, for the sake of simplicity, has both contraction and expansion capacities.

Joints are snapped together to form arbitrarily long wormlike robots (**Figure 3**).



**Figure 2.** The basic robotic joint with sensors, muscles, and neural driver. The joint can be snapped into long chains. Muscles are driven by the output neurons of an associate network.



**Figure 3.** Modular robot with 16 joints and dedicated driven neurons in its working environment.

## 4. Autonomy

Our approach to autonomous robots is based in autonomous controllers, constructed with artificial neural nets, and incorporating basic rules of living brains in terms of energy usage, default mode network (DMN), and the orchestrated, autonomous transitions forms default mode to other operative behaviors in reaction to stimuli.

### 4.1. Artificial neural networks

The neural networks used in this chapter are sigmoidal neurons trained with backpropagation [17]. Other functioning details are given in [12, 13].

In both biology and circuit complexity theory, it is maintained that deep architectures can be much more efficient (even exponentially efficient) than shallow ones in terms of computational power and abstract representation of some functions [18, 19]. Unfortunately, well-established gradient descent methods such as backpropagation that have proven effective when applied to shallow architectures do not work well when applied to deep architectures. Our method uses shallow nets trained with backpropagation, but these networks are thereafter stacked with other networks, thus becoming deep architectures.

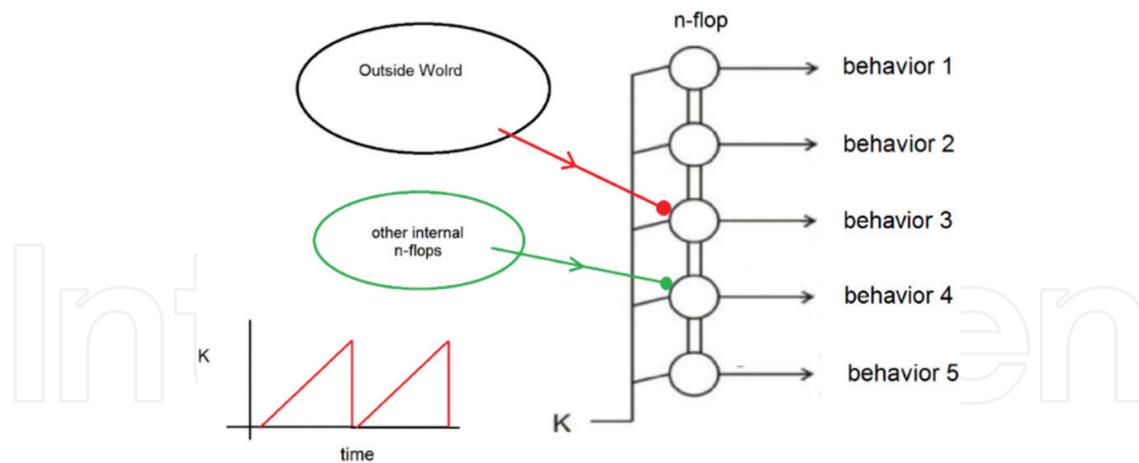
## 5. Neural controllers

### 5.1. Autonomous behavior initiators

As mentioned in the introduction, the *Drosophila* brain involves nonlinearity and the competence of only a few neurons in the final fly's behavior-initiating mechanism, deep buried in its brain. So, we are interested in neural structures with few neurons and genuine spontaneity. In the previous work [12], we presented a solution where the term behavior is defined as a finite sequence of events distributes in the space time. The initiation of these sequences is fired by using an n-flop, a robust network constructed with sigmoidal-type neurons sharing a common self-activating excitatory input  $K$  [20]. Being robust, it serves as foundation for other large-scale optimization structures that solve difficult jobs, such as the travel salesman problem (TSP). The n-flop is the basic building block beyond the concept of programming with neurons [20], and the term is derived from the flip-flop, a computer circuit that has only two stable states. n-flops have  $n$ -stable states and the rooted capacity to solve high-dimensional problems [21].

In an n-flop, neurons are programmed by their weight interconnections to solve the constraint that only one of them will be active when the system is in equilibrium. To this end, the output of each neuron is connected with an inhibitory input weight (-1) to each of the other  $n-1$  neuron inputs (lateral inhibition). In addition, each neuron receives a common excitatory input  $K$  which, on controlled situations, tends to force all neuron outputs toward 1. A solution or desired output is self-activated by rising  $K$  and forcing all neurons to some near-equilibrium but unstable "high-energy" state. At this point,  $K$  is set to almost zero, forcing the network to seek a low-energy or equilibrium-stable state. The solution given by a non-biased n-flop is a unique but unpredictable winner, which may be used as a behavior initiator, where "behavior" corresponds to a finite sequence of events, distributes in the space time. A unique winner guarantees a conflict-free operation in terms of robotic conduct. A good stabilized n-flop will always satisfy the syntactic rule "only one winner," even when neurons in the n-flop community share input weights with outside-world neurons, including other n-flops. This conduces toward a proactive scenery where it is possible to control, with events that happen inside or outside the robot, the initiation of behaviors that are being pushed from the inside (**Figure 4**).

Like in biological brains, the proposed behavior initiator constantly consumes energy, and since it controls behaviors, it can affect the whole information processing of the robot.



**Figure 4.** The n-flop as a behavior pusher. A set of sigmoidal-type neurons inhibit each other with  $-1$  value weights and share a common self-activating excitatory input  $K$ . Under controlled situations,  $K$  raises and forces all neuron outputs toward a 1 output. Once in this high-energy position,  $K$  comes down to almost zero, forcing the network to seek a low-energy or equilibrium state. For each  $K$  cycle, a new, unique, random winner is generated. Neurons in the n-flop can share input weights with other insider or outsider neurons. This results in a proactive scenery where behavior initiation is being pushed from the inside and all behaviors can be eventually awakened, but it is still possible for events inside or outside of the robot to modify the statistical occurrence of behaviors.

## 5.2. The wave generator

Waves are important in living creatures, and some forms of contraction waves are always used for locomotion and other activities [22].

We use a wave generator network to control the robot's muscles through a one-to-one assignment so that one output neuron controls one muscle. The net is pre-trained with a set of inputs that produce output values in the range 0–1. This moves the joint associated to the muscle in the range  $(\alpha, -\alpha)$  where  $\alpha$  is a target angle measured in degrees. With the appropriated targets, the net learns to reproduce moving wave forms in its outputs (**Figure 5**).

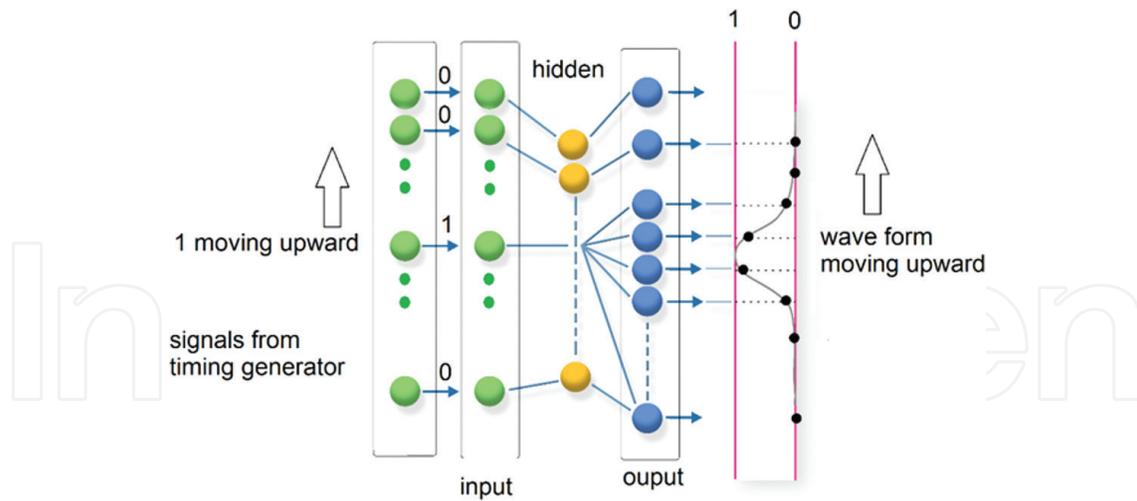
The training objective is to create a neural moving wave that in turn produces a mechanical moving wave through the robot's body.

## 5.3. The timing generator

Timing is important in living creatures making it possible to control complex thing, from walking to sleeping [23]. We use a neural timing generator trained with backpropagation so that its output vector behaves as a programmable shift register with left, stop, right commands and a winner-takes-all output; the winner stays near 1, while all other  $m-1$  outputs stay off (near to 0). The training objective is to create neural timing signals that produce mechanical timing through the robot's body (**Figure 6**).

## 5.4. The position predictor

The position predictor is trained to indicate the position of the ball when it touches the sensors. The predictor receives sensor signals in the range 0–1 as input and predicts the mean position of the detected ball in terms of one joint number. For example, in **Figure 7**, due to

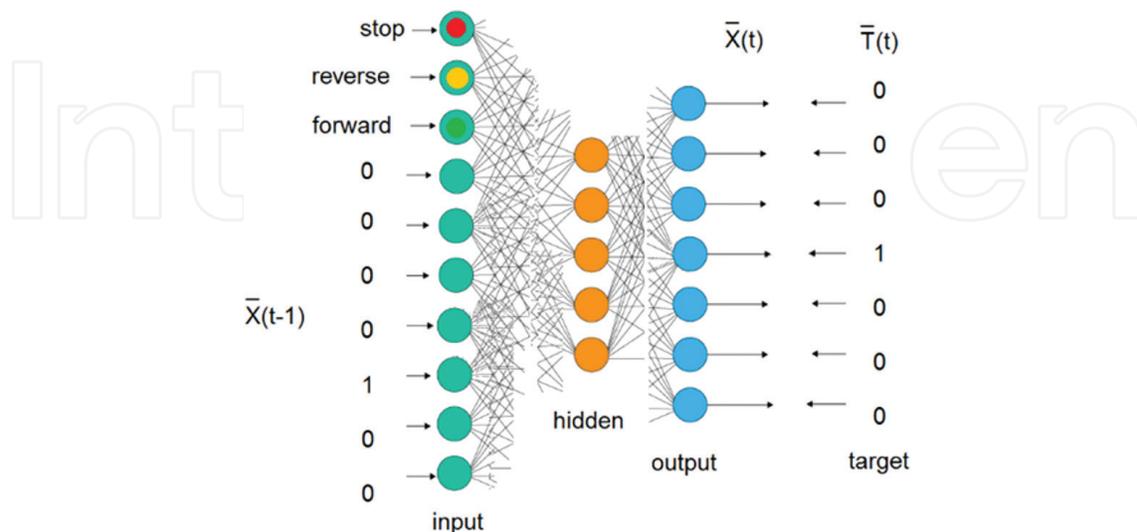


**Figure 5.** The wave generator. A shallow network receives as input 16 decoded timing signals produced by a timing generator. By using a moving wave form as target, the net is trained to generate the chosen waveform and to move it, when the input timing signal moves. Since output neurons directly drive robot’s muscles, moving mechanical waves can be created in the robot’s joint structure. The chosen hidden layer has nine neurons so data compression occurs.

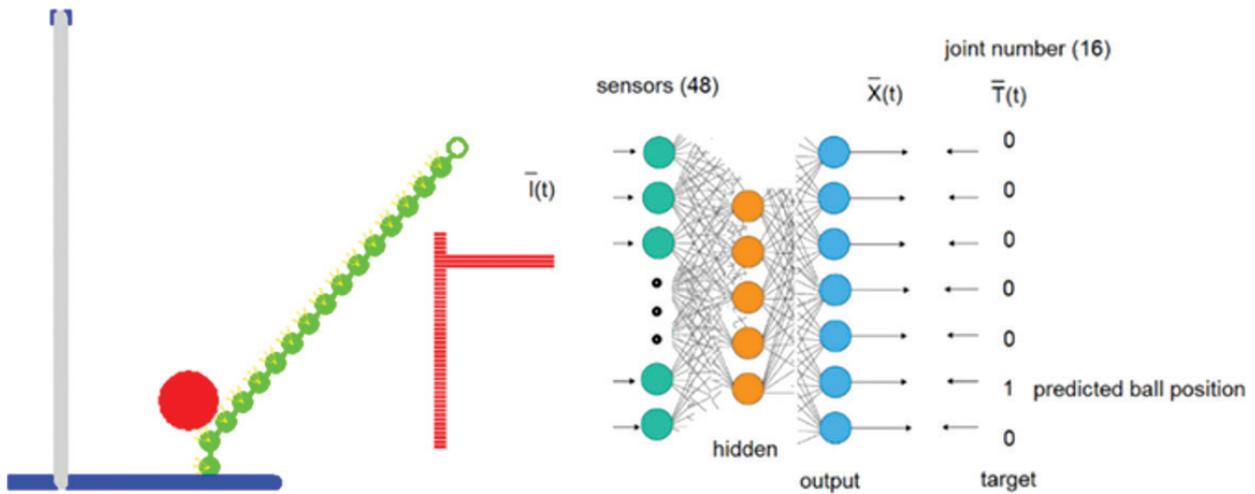
spatial interaction between the space sensors and the ball, a complex input pattern is produced. Applying a mass center algorithm, the position of the ball can be estimated in terms of a unique joint number and given as target to the net.

### 5.5. The ball presence detector

The ball presence is trained to indicate that the ball is touching the sensors somewhere along the robot’s body. It is a first front detector, and its training includes white noise as counter example so that the net learns to distinguish the specific sensor excitation pattern produced by the ball (**Figure 8**).



**Figure 6.** The timing generator. After training, the output vector behaves as a programmable shift register, with left, stop, right, commands and a winner-takes-all output, where only one of n-defined output stays on (near to 1), while all other m-1 outputs stay off (near to 0).

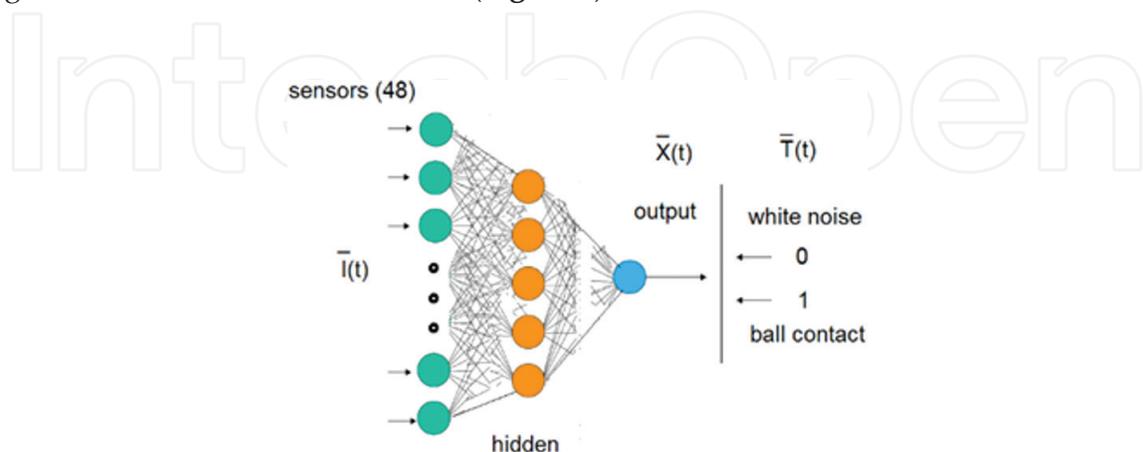


**Figure 7.** The position predictor is trained to indicate the position of the sensed ball in terms of a joint number. From 48 sensors, 16 outputs are derived.

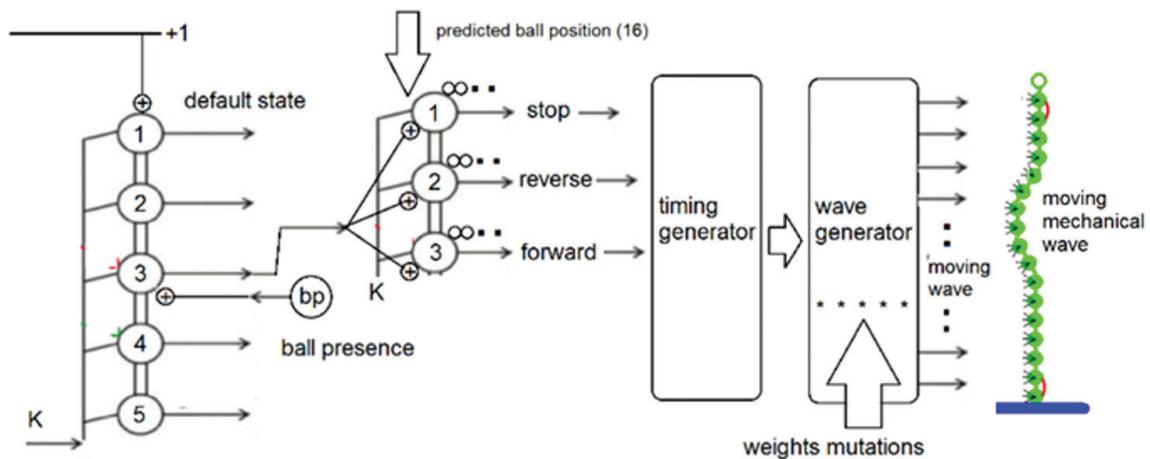
### 5.6. The autonomous neural controller

Our next step is to assemble the above-defined neural devices into an integrated autonomous neural controller (ANC) that combines the different capacities of the participant networks. As shown in **Figure 9**, a five-flop is established as a basal behavior initiator, where neuron 1 is connected with a positive weight to a constant output, becoming the neuron with the highest probability to win, assuming the role of a default network. When the ball presence network becomes active, the state 3 may become a winner activating the three-flop, which feeds random values to the wave generator producing a random moving wave. Notice that a single event (ball presence) activates a complex assembly of neural devices that work by themselves, creating a mechanical wave running through the robot's body.

To promote complex behaviors, a set of selected outputs are allowed to have connecting weights with a selected set of neurons (**Figure 9**).



**Figure 8.** The ball presence detector is trained to indicate that the ball is touching the sensors somewhere along the robot's body.



**Figure 9.** The autonomous neural controller. A 5-flop acts as basal behavior initiator; its number 1 neuron is connected with a positive weight value to the constant 1 line, making the neuron the highest probability to win (default state). When the ball presence network (bp) becomes active, the state 3 becomes a likely winner that in turn activates the 3-flop. Next, random values are fed to the wave generator, producing a random moving mechanical wave in the robot. To promote complex behaviors, selected outputs are allowed to develop connecting weights with other neurons, for instance, the 3-flop neurons have connective input weights with the position predictor network. Through weight mutations, complex behaviors emerge.

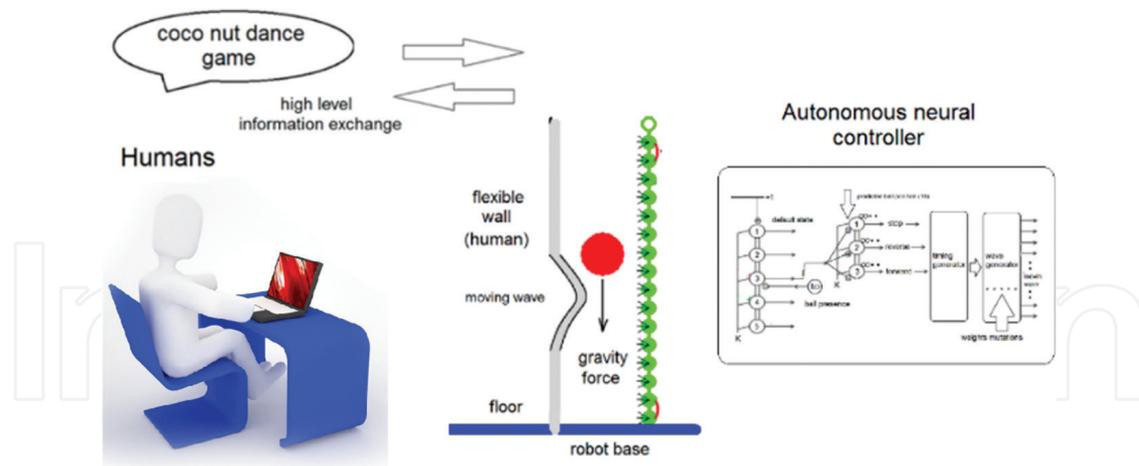
Specifically, the outputs of the ball position predictor (16) are weight connected with the neurons of the 3-flop, making possible for the ball position to control the direction of the moving wave. This comprises  $16 \times 3 = 48$  weights.

The hidden weight in the wave generator (144 weights) is also allowed to mutate, opening opportunities for different wave forms and wave movements to appear. The overall behavior of the free-running autonomous neural controller is thus governed by these 192 variables.

## 6. The human-robot interaction

Once the autonomous neural controller (ANC) begins to behave like an orchestra conductor, issuing timing signals to coordinate activities among different control regions, humans begin to interact with the robot through a keyboard and a visual interface (**Figure 10**). Humans are asked to play the coconut dance game, in which a couple tries, without using their hands, to move upward a coconut placed between them at their waist level; in our case, the human player uses the robot as dancing partner. We choose this activity because it requires a close, coordinate interaction between the two participants, and it doesn't have a trivial solution. The coconut (ball) is subjected to gravity force and is released somewhere between the dancers. The human, represented by a flexible wall, must use the keyboard to move toward the robot and trap the ball between the two bodies; he/she then uses the keyboard to manipulate a moving body bending that pushes the ball up. The game is won when the ball is pushed up, out of the body's reach.

Animated by internal n-flops, the robot behaves proactively, burning energy and initiating behaviors independently of the outside world.



**Figure 10.** Human-robot interaction. Humans play the coconut dance game by using the robot as dancing partner. This activity requires a close, coordinate interaction between the two participants, and it doesn't have a trivial solution. The coconut, pulled down by gravity, is released somewhere between the dancers. The human (flexible wall) must use the keyboard to trap the ball between the two bodies and then manipulate a moving body bending, to push the ball up and out of the body's height.

## 7. Genetic algorithms

Genetic algorithms are search algorithms used to find near-optimal solutions in arbitrarily created search spaces [24]. Applications in robot control have been reported in [25]. In this work, the search space is defined by a chromosome formed with the 192 weights defined in Section 5.1.

The 144 weight values obtained in the trained process in Section 5.2, corresponding to the wave generator's hidden layer, are left untouched but subjected to possible future changes. The 48 weight values corresponding to the ball position predictor and the 3-flop are given initial random value between +0.5 and -0.5.

Genetic algorithms have three main operators: selection, crossover, and mutation.

For the purposes of this chapter, we will use an evolutive approach where only mutation and selection are put to work. This kind of process plays a dominant role in bacterial evolution [26] and in pseudo-code can be written as:

```

get fitness
{
    timer=p;
    store initial coconut vertical position hi
do{
    use stored move
    play

```

```

    timer--
  } until timer>0
get coconut final vertical position  $h_f$ 
}
if (  $h_f - h_{i>0}$  ) fitness=  $h_f - h_i$ 
else > fitness=0

```

Mutation is implemented by iterating all bits in the chromosome and randomly adding a small value (positive/negative) to them. The probability of changing one weight is called the mutation rate and is here maintained in 10%.

## 8. Results: the quick evolution

### 8.1. Experiment 1. High human activity

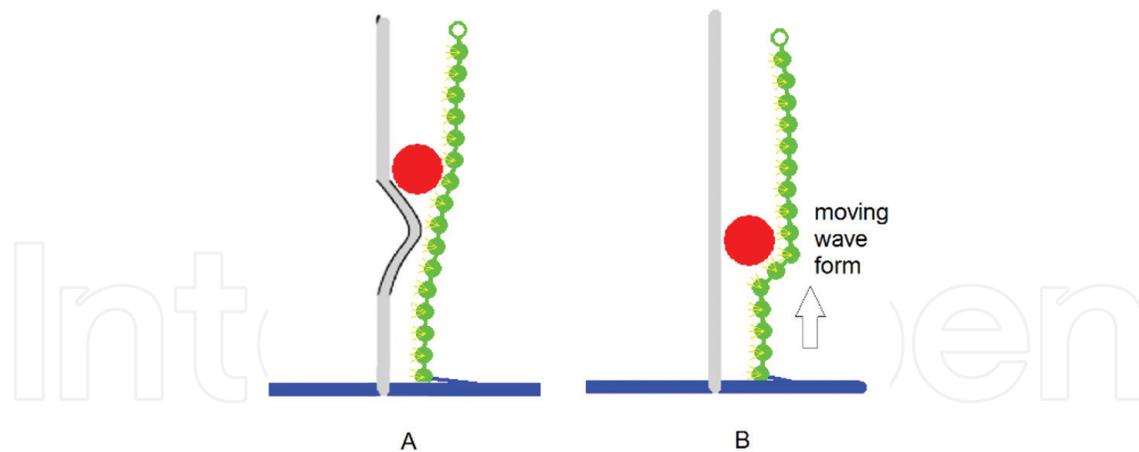
Several human players interact with the robot. His/her moves (keyboard inputs) are stored in a vector with fixed time sampling. The fitness is measured in how much the coconut raises in a given time period, in pseudo-code:

```

get fitness
{
  timer=p;
  store initial coconut vertical position  $h_i$ 
  do{
    use stored move
    play
    timer--
  } until timer>0
get coconut final vertical position  $h_f$ 
}
if (  $h_f - h_{i>0}$  ) fitness=  $h_f - h_i$ 
else > fitness=0

```

After using this fitness formula with the genetic algorithm of Section 7 and after about 5000 accepted mutations, the kind of individual shown in **Figure 11a** evolved. Since humans do most of the active part, the evolved robots learn to stay upright, facilitating the human actions, but show little or null body wave activity.



**Figure 11.** Evolved autonomous robots. (A) When human do most of the active part of the game, the evolved robots learn to stay upright, facilitating the human actions, but show little or null body wave activity. (B) When the human provides little game action and puts the coconut-lifting responsibility in the robot, evolution teaches the robot the connotation of the game. The evolved robot develops an autonomous dynamic response that learned to produce its own mechanical moving body bending and uses it to push the coconut all the way up.

## 8.2. Experiment 2. Low human activity

For this setting, the human players stay mostly inactive, as a passive wall whose only function is to get the coconut pressed against the robot.

By using the same fitness formula of experiment 1, a quite different outcome is obtained. Although the human provides little action to the game, the fitness formula put all the coconut-lifting responsibility in the robot. In other words, evolution teaches the robot the connotation of the game. The final evolutive result, after about 19,000 mutations, is a robot with an autonomous dynamic response that learned to produce its own moving body bending and uses it to push the coconut all the way up, out of the body's gap (**Figure 11b**).

## 9. Conclusions

By coupling two self-activated n-flops, we end up with an autonomous behavior initiator system that mimics the functioning of a living brain, in the sense that a default network consumes energy and is ready to initiate other behaviors under specific stimulus. Due to n-flops activity, all behaviors are constantly self-pushed from the inside.

With behaviors pushing from the inside, the robot is quite ready to face the real world and quickly learn new tricks. This is corroborated by the relative small number of mutation required to evolve reliable robots.

Our model incorporates some basic aspect of biological brains: (a) a fraction of the overall activity of all energy used by the autonomous neural controller (ANC) occurs in circuits unrelated to any external event. (b) In terms of structure, the components of the ANC are separated, carefully knitted constructions with pronounced job specializations.

Complex behaviors are codified in one single chromosome with 198 genes.

This satisfies one of the basic rules of evolution: Few genetic information unravels into complex things.

It seems reasonable to conclude that in a compact gene, small mutations produce enormous changes in the mutated individual, which enriches the search for solutions.

At least for our model of ANC, a successfully interaction with humans depends on the human attitude, if the humans put too much emphasis on the robot to learn to stay quiet. On the other hand, if human stays quiet but the basic rules of the game (lift the ball) is passed on to the robot learning, then the robot will pick up to the hard part of the job.

As in biology our robots, concerning behavior initiation, do throw the dice, but they keep and attractively control over when, where, and how this random event will be put into effect.

## Author details

Oscar Chang<sup>1,2,3\*</sup>

\*Address all correspondence to: [ochang@yachaytech.edu.ec](mailto:ochang@yachaytech.edu.ec)

1 Yachay Tech. School of Mathematical Sciences and Information Technology, Republic of Ecuador

2 Universidad Central de Venezuela (UCV), School of Electrical Engineering, Graduate Studies Program, Venezuela

3 Prometeo Project, SENESCYT, Republic of Ecuador

## References

- [1] Maye A, Chih-hao H, Sugihara G, Brembs B. Order in spontaneous behavior. *PLoS One*. 2007;**10**(1371). (Online). Available: <http://www.plosone.org/article/info:doi/10.1371/journal.pone.0000443>
- [2] Brembs B. Genetic Analysis of Behavior in *Drosophila*. *Cognition and Behavioral Neuroscience*. Online Publication. Date: Feb 2017. DOI: 10.1093/oxfordhb/9780190456757.013.37
- [3] Ichle M, Marcus E, Snyder AZ. A default mode of brain function: a brief history of an evolving idea. *Neuroimage*. 2007;**37**(4):1083-1090
- [4] Raichle ME. Two views of brain function. *Trends in Cognitive Sciences*. 2010;**14**(4): 180-190

- [5] Eban-Rothschild A, Bloch G. Circadian rhythms and sleep in honey bees. *Honeybee Neurobiology and Behavior: A Tribute to Randolph Menzel*. 2012: Springer Netherlands. pp. 31-45. [https://doi.org/10.1007/978-94-007-2099-2\\_3](https://doi.org/10.1007/978-94-007-2099-2_3)
- [6] Palyanov A, Khayrulin S, Larson SD, Dibert A. Towards a virtual *C. elegans*: A framework for simulation and visualization of the neuromuscular system in a 3D physical environment. In *Silico Biology*. August 2012;**11**(3-4):137-147. DOI: 10.3233/ISB-2012-0445
- [7] Sharique H, Mall RN. Development of autonomous aero-robot and its applications to safety and disaster prevention with the help of neural network. *International Journal of Engineering Research & Technology (IJERT)*. August – 2013;**2**(8). IJERT ISSN: 2278-0181
- [8] Pessin G, Osório FS, Ueyama J, Wolf DF, Braun T. Mobile robot indoor localization using artificial neural networks and wireless networks. In: *Proc. of First Brazilian Conference on Critical Embedded Systems (I CBSEC)*. 2011. pp. 89-94
- [9] Khan M. *Imitating the Brain: Autonomous Robots Harnessing the Power of Artificial Neural Networks*. Computer Science Honors Papers. 8. 2017. <http://digitalcommons.conncoll.edu/comscihp/8>
- [10] Bartona A, Volnab E. Control of autonomous robot using neural networks. *AIP Conference Proceedings*. 1863. 070002. 2017. DOI: <http://dx.doi.org/10.1063/1.4992224>. July 2017
- [11] Nasser R, Philippe G. *Robotic Grasping: A Generic Neural Network Architecture, Mobile Robots: towards New Applications*. Lazineca A. ed: InTech; 2006. DOI: 10.5772/4687. Available from: [https://www.intechopen.com/books/mobile\\_robots\\_towards\\_new\\_applications/robotic\\_grasping\\_\\_a\\_generic\\_neural\\_network\\_architecture](https://www.intechopen.com/books/mobile_robots_towards_new_applications/robotic_grasping__a_generic_neural_network_architecture)
- [12] Chang O. Evolving cooperative neural agents for controlling vision guided mobile robots. 2010 IEEE 9th International Conference on Cybernetic Intelligent Systems, Reading. 2010:1-6
- [13] Chang O, Campoy P, Martinez C, Olivares-Mendez M. A robotic eye controller based on cooperative neural agents. *Neural Networks (IJCNN), The 2010 International Joint Conference on*. pp. 1-6. 2010. <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5898127&isnumber=5898083>
- [14] Harris J, Jolivet R, Attwell D. Synaptic energy use and supply. *Neuron*. 2012;**75**(5):762-777. ISSN: 0896-6273, <http://dx.doi.org/10.1016/j.neuron.2012.08.019>
- [15] Collins F. Making the Connections: Study Links Brain's Wiring to Human Traits. *National Institutes of Health*; 2015. <https://directorsblog.nih.gov/2015/10/06/making-the-connections-study-links-brains-wiring-to-human-traits/>
- [16] Hochberg R. Three-dimensional reconstruction and neural map of the serotonergic brain of *Asplanchna brightwellii* (Rotifera, Monogononta). *Journal of Morphology*. **270**: pp. 430-441

- [17] Suliman A, Zhang Y. A review on back-propagation neural networks in the application of remote sensing image classification. *Journal of Earth Science and Engineering*. 2015;5:52-65. DOI: 10.17265/2159-581X/2015. 01. 004
- [18] Bengio Y, Courville A, Vincent P. Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*. 2013;35(8):1798-1828
- [19] Bengio Y. Learning Deep Architectures for AI. *Foundations and Trends in Machine Learning*. 2009;2(1):1-127
- [20] Shackelford JB. Neural data structures: programming with neuron-technical. *Hewlett-Packard Journal*. June, 1989:69-78
- [21] Serpen G. Managing spatio – temporal complexity in Hopfield Neural Network simulation for large-scale static optimization. *Mathematics and Computers in Simulation*. 2004;64(2) January 2008:279-293
- [22] Organismic Biology course (Bi 11), Notes on Animal Phylogeny. <http://www.cco.caltech.edu/~brokawc/Bi11/AnimalPhylogeny1.html>
- [23] Rensing L, Meyer-Grahlé U, Ruoff P. Biological timing and the clock metaphor: Oscillatory and hourglass mechanisms. *Chronobiology International*. 2001;18(3):329-369
- [24] David E. Goldberg. Genetic Algorithm, — in Search, Optimization & Machine Learning. Addison Wesley; 1989. Google Scholar
- [25] Gill MAC, Zomaya AY. Genetic algorithms for robot control. *Proceedings of 1995 IEEE International Conference on Evolutionary Computation, Perth, WA, Australia, 1995*, 462 p. DOI: 10.1109/ICEC.1995.489192. <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=489192&isnumber=10439>
- [26] [https://www.researchgate.net/publication/20715998\\_Bacterial\\_EvolutionBacterial\\_Evolution\\_Algorithm\\_for\\_rapid\\_adaptation](https://www.researchgate.net/publication/20715998_Bacterial_EvolutionBacterial_Evolution_Algorithm_for_rapid_adaptation)

IntechOpen

