

Self-inhibiting modules can self-organize as a brain of a robot: A conjecture

doi:10.1533.abbi.2005.0061

J. Negrete-Martínez

Departamento de Biología Celular y Fisiología. IIBm. Unidad periférica en Xalapa, UNAM, and Facultad de Física e Inteligencia Artificial, UV, Calle Sebastián Camacho # 5, Colonia Centro 91000 Xalapa, Veracruz, Mexico

Abstract: In this article we describe a new robot control architecture on the basis of self-organization of self-inhibiting modules. The architecture can generate a complex behaviour repertoire. The repertoire can be performance-enhanced or increased by modular poly-functionality and/or by addition of new modules. This architecture is illustrated in a robot consisting of a car carrying an arm with a grasping tool. In the robot, each module drives either a joint motor or a pair of wheel motors. Every module estimates the distance from a sensor placed in the tool to a beacon. If the distance is smaller than a previously measured distance, the module drives its motor in the same direction of its prior movement. If the distance is larger, the next movement will be in the opposite direction; but, if the movement produces no significant change in distance, the module self-inhibits. A self-organization emerges: any module can be the next to take control of the motor activity of the robot once one module self-inhibits. A single module is active at a given time. The modules are implemented as computer procedures and their turn for participation scheduled by an endless program. The overall behaviour of the robot corresponds to a reaching attention behaviour. It is easily switched to a running-away attention behaviour by changing the sign of the same parameter in each module. The addition of a “sensor-gain attenuation reflex” module and of a “light-orientation reflex” module provides an increase of the behavioural attention repertoire and performance enhancement. Since scheduling a module does not necessarily produce its sustained intervention, the architecture of the “brain” is actually providing action induction rather than action selection.

Key words: Modular robot control, self-inhibiting modules (SIMs), self-organization, action induction, brain theory.

INTRODUCTION

The words “brain”, “brained”, “brain-like”, “brain architecture” and “attention” are used in this article within the context of Brain Theory.

We describe a brain architecture with self-organized modules so as to produce at least a complex attention repertoire that can be performance-enhanced or increased by modular multifunctionality and/or by addition of new modules. Behavior Oriented Systems and Recurrent Modular Systems are candidates for self-organization, generating the aforementioned brain-like characteristics.

In some Behavior Oriented (BO) systems, self-organization emerges (BO.1) by the interference of higher hierarchy modules with the inputs or outputs of “subsumed” modules (Brooks 1986) and (BO.2) by spreading activation among behaviour-generating modules (Maes 1990).

In some Recurrent Modular (RM) systems, self-organization emerges (RM1) by the modulation of some reciprocally oscillating modules (Beers 1990) and (RM2) by scheduling recurrent self-inhibiting modules (SIMs), as we propose (Negrete-Martínez and Cruz 2002).

To test the possibility of a brain-like system with a self-organized control based on SIMs, a simple robot “RM2-ROB” was built.

IMPLEMENTATION OF AN RM2-ROB

Body and hardware

We implemented a hybrid RM2-ROB that is both *ad hoc* hardware and computer software. The main mechanical

Corresponding Author:

J. Negrete-Martínez

Departamento de Biología Celular y Fisiología

Calle Sebastián Camacho #5, Colonia Centro 91000 Xalapa, Veracruz, Mexico

Tel: 228-8172957; Fax: 228-8172855

Email: jnegrete@uv.mx

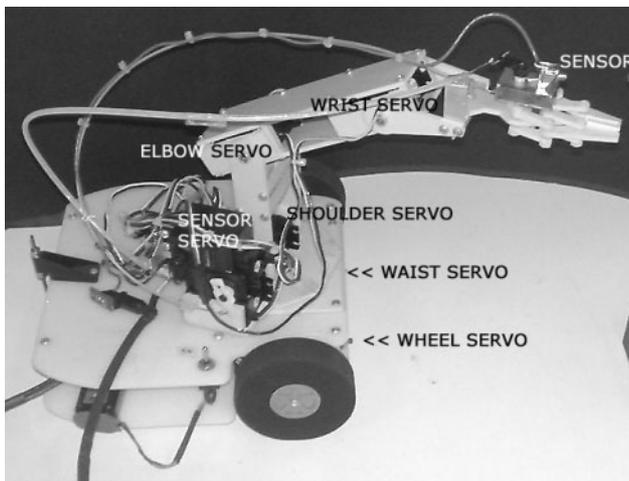


Figure 1 Body and motors. Denotation of servos and parts.

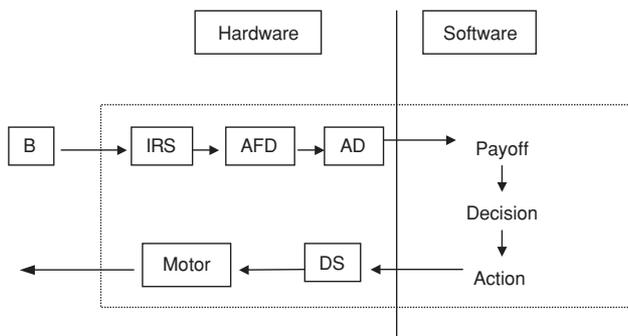


Figure 2 Implementation of a generic SIM module. The module has an ad hoc hardware part and a software part. The hardware part is composed of an infrared sensor IRS that receives the light signal from a beacon B; an A-F-D amplifier-filter-detector that receives the IRS output signal; AD is an analog-to-digital transducer to the computer; DS is a digital-to-servomotor position transducer from the computer. The software part has a payoff function, a decision function and an action function.

part is a three-wheeled car that carries, on its deck, an articulated arm, which has a grasping tool, and a mount for a sensor at its free end (Figure 1). Other parts include five servo motors, which move the five joints of the arm, and two modified servo motors, each one driving a frontal wheel of the car; a computer-servos interface (DS); a single-infrared sensor (IRS); an Amplifier-Filter-Detector Circuit (A-F-D) for the IRS output signal and the AD interface between the A-F-D and a PC computer (Figure 2).

There is a beacon in the robot's environment that produces a flickering IR light. This light generates a corresponding square wave voltage in the IRS. This voltage wave is amplified and then filtered (to eliminate the high-frequency noise) finally, the resulting wave is filtered again (Detection) leaving only its DC envelope as the output of the A-F-D.

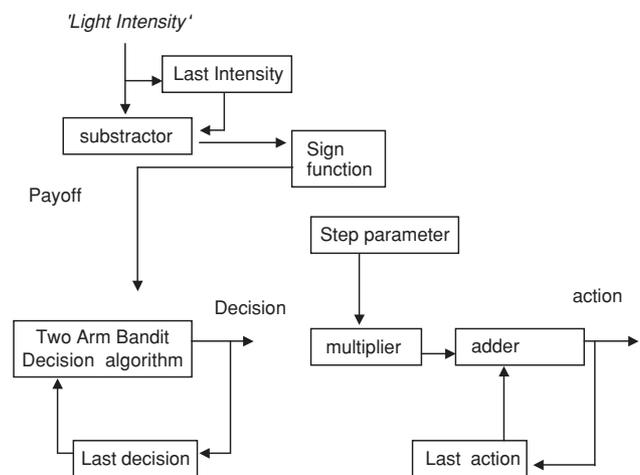


Figure 3 Detail of the payoff, decision and action functions of a SIM. PAYOFF FUNCTION: The light intensity (as digital voltage) feeds a subtractor and actualizes a “last intensity” variable. The output of the subtractor or the payoff of the last movement of the robot is transformed into a payoff value by a sign function. The output of this last function is +1 or -1. A zero value output is produced when the payoff of the movement is in a range near zero difference. DECISION FUNCTION: The payoff value is fed to a “Two arm bandit algorithm”. This algorithm has this input and the last value decided by the algorithm. The combination of the two input values produces a decision value output identical to the last decision value when both inputs are the same; otherwise, the algorithm switches to the opposite last decision. ACTION FUNCTION: The output decision value is +1 or -1. This value is multiplied by a *step* value and fed to an adder that produces the signal to the corresponding servo and actualizes the last action value. The three functions are nested into a *while* function that re-enters when the payoff value is non-zero.

Software

Implementing the basic self-inhibiting module

Figure 3 shows the basic software organization scheme of any SIM. The process is implemented with a “while” function of a “C” language in a PC. Each module, during its activation, reads the “light intensity” coming from the IR beacon. A payoff function calculates the difference of IR light intensity before and after the module’s motor action and maps the difference into a payoff value -1, +1 or 0. The case zero is when the difference is within a pre-defined range near 0. There is also a Decision function that uses the payoff value to determine the direction of the next movement. The module repeats the movement or its opposite until the payoff is near zero (in “zero range”). If the payoff value is in zero range, the module self-inactivates. The movement generated by this module is actually a step movement added to the present position of the joint in one direction or the other, or a step rotation of the car clockwise or counterclockwise, or a fixed rolling time movement of the car either forward or backward.

In every module the Decision function makes use of an algorithm related to the “Two Arm Bandit” algorithm (Kaelbling 1993), inputting the payoff value and its own previous output value. The Decision Function output is the same as the last output when the payoff is positive, but it switches to the opposite last output if the payoff value changes to negative. The output value of the function (1 or -1) calculated by the aforementioned algorithm is multiplied by a *step* value and then added to the *Last Action* value stored in the function (Figure 3). However, if the payoff value of the movement is in the zero range, the function self-inactivates before proceeding to calculate anything else. Every SIM receives the name of the joint that it moves: WAIST, SHOULDER, ELBOW, WRIST. Figure 1 shows the labels on a picture of the robot. The names CAR_ROTATION and WHEELS F/B stand for the modules that produce the movement of the car.

The organization among the modules

The set of modules is called in an endless cycle by the main program of the computer. However, the turn of the modules was changed in several experiments.

Changing parameters

For performance-enhancement purposes, we changed several parameters of each module. However, a remarkable change in performance occurred when the sign of the Step parameter was modified.

Adding a searching-light module

To avoid the stopping of movements due to lack of light stimulation, a new module was added to the already-described set of modules. This module produces rotation of the sensor mount in the absence of light and self-inhibits when a non-significant light signal is found. We dubbed this module SENSOR_JOINT. This module acts on an *ad hoc* mount of the IR sensor (Figure 4). The mount rotates the IR in a plane parallel to the “hand”. The SENSOR_JOINT differs from the rest of the modules in that it self-inhibits when the payoff passes from positive to negative and it changes the sign of the Step every time the sensor-joint servo reaches its extreme position. Before the SENSOR_JOINT self-inhibits, it calls another module, CAR_ROTATION, which rotates the car to the left or to the right by acting on the modified servos of each wheel. The module “reads” the last angle taken by the sensor joint and rotates the car in an angle similar to that angle and then unconditionally self-inhibits. The SENSOR_JOINT is called by the main program after every joint’s module calling.

Adding hardware and a new module

The A-F-D was modified so that the gain of the “A” stage could be attenuated by the action of an added Gain Attenuating Module (GAM). The module samples the output voltage of the A-F-D and its activation moves a relay that attenuates the gain in the hardware of the A stage. The activation is produced only when the output voltage of the

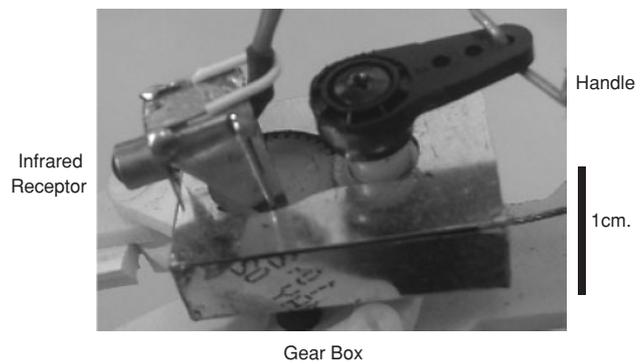


Figure 4 Detail of the sensor mount on top of the “hand”. It is a pull-to-rotate gear box that rotates the infrared sensor, parallel to the hand.

A-F-D reaches the saturation level of the AD. The module is not self-inhibiting. GAM is scheduled by the main program after the call of every joint module.

RM2-ROB PERFORMANCE

The scheduling of modules in a different order in each of several trials of the first implementation of the RM2-ROB produced an approaching movement of the arm and car toward the beacon. In all of the experiments the grasping tool with its sensor eventually reached the beacon from any initial position and attitude of the robot, provided the sensor started picking up light from the beacon.

Irrespective of the scheduling and long before the robot gets near the beacon, the arm’s attitude is already a grasping attitude. The final docking of the sensor’s tip to the beacon is frequently touching it (Figure 5).

With the addition of the searching light module, we started experimenting with the robot in any initial attitude or position, even facing away from the beacon. We would momentarily interrupt the light or move the position of the beacon; but, the robot eventually turned to face the beacon.

When we changed the sign of the Step parameter in the Decision functions, we could drastically switch the reaching behaviour to a running-away behaviour. When this parameter change was implemented, an incomplete running-away behaviour was produced: the robot running-away movement stopped when the sensor was no longer stimulated by the light.

Without the GAM, the robot, when near the beacon, searches for the non-saturated regions of the emitted light, producing a side-wise arm approach. When the GAM is incorporated, the robot reaches the beacon in a more direct way.

When the searching-light module was included in the endless calling cycle in a running-away configuration, the very interesting behaviour of turning towards the beacon alternating with the running away behaviour was observed.

The relative autonomy of the modules suggests that external hindering of the movements of part of the robot

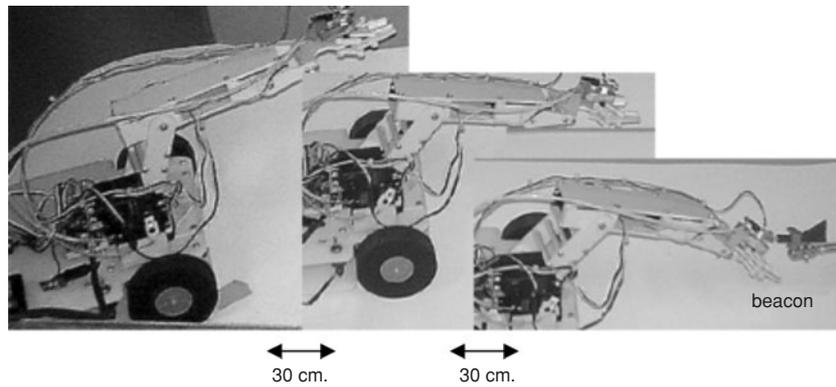


Figure 5 Three photos of the experimental sequencing: {WAIST, SHOULDER, ELBOW, WRIST, WHEELS_F/B}. On the left extreme is the attitude and position of the robot at the start. The photo in the middle is the position and grasping attitude of the robot long before reaching the beacon. To the right is the position and attitude of the robot at the beacon. The sensor is nearly touching the beacon. The space rolled by the robot between photos is 30 cm.

would not suppress the reaching action completely. Thus, when we hindered the robot’s advance with an obstacle, the arm still stretched out towards the beacon. A similar arm extension was observed when the beacon was placed too high for the arm to reach.

DISCUSSION

If we assign the term “burst” to the sequence of movements produced in a given joint without interruption, the robot produces a series of bursts. The order of the joint bursts as well as the duration changed during the observed behaviours (see the table in Figure 6). This is a clear manifestation of an implemented self-organization.

CALL\CYCLE	1	2	3	4	5	6	7	8
SENSOR_JOINT	94	3	3					
CAR_ROTATION	1	1	1					
WAIST	3		7	3	6	5	5	1
SHOULDER		3		4	5	2	2	1
ELBOW				4		2	2	1
WRIST				4	3	6	5	
WHEELS_F/B			16		2	2	1	

Figure 6 RN2-ROB with the orienting-reaction module. The files of the table are the calling order of the endless program and the columns, the program cycle turn. The numbers in the body of the table are the number of times of successive participations of the corresponding module in its turn. SENSOR_JOINT and CAR_ROTATION are the two modules that produce the orienting-reaction and the rest of the modules are responsible of a reaching behaviour. WHEELS_F/B stands for the module producing the straight-line rolling movement of the car. Notice that in each cycle some modules did not contribute to the robot’s movement. This qualifies the module’s participation as self-organized.

This self-organization also explains why there was an unexpected pre-grasping attitude of the arm long before it reached the beacon and the hyperextension found when the environment was manipulated.

In the RM2-ROB, self-organization generates a behaviour that is more than a simple reaching-light behaviour. It should be classed as an open-reaching neurological attention, as can be concluded from previous simulations of a similar robot (Negrete-Martínez and Cruz 2002).

The SENSOR_JOINT is an Attentive Module *per se*. Its implementation and addition are the solution of the problem implicit in the need to find a general location of the object of attention. The result is equivalent to having implemented the Orienting-reaction component of attention (Parasuraman 2000).

The module SENSOR_JOINT is a modified copy of any of the SIMs. As mentioned before, the module inverts the sign of the parameter *Step* every time the servo reaches its extreme position. This makes the servo oscillate in the absence of light. The addition of this module can be seen as a case of the RM2-ROB’s growth and evolution through a copy-modify-add policy.

The implementation of a module that attenuates the amplification gain in the A-F-D is conceptually the case of the addition of a reflex “on top” of the whole attention behaviour.

The elicitation of a run-away behavior through a change of sign in the *Step* parameter in the joint-modules can be seen as an instance of multi-functionality (at least bi-functionality).

CONCLUSIONS

Since the calling of a module in RM2 does not necessarily produce its sustained intervention, the Main Program can be considered as a primitive brain architecture actually

carrying out an Action Induction rather than an Action Selection.

We postulate that combining BO and RM organizations should lead to the implementation of more realistic brains for robots, that is, the calling by program of the SIMs can subsume (BO1) the timing of the sensor joint scanning and this scanning can be implemented by a reciprocally inhibiting (RM1) organization; and finally the switching between reaching and running-away behaviours can be accomplished by spreading activation among the modules (BO2).

In our endeavour to implement an RM2 primarily made out of SIMs, we have built a structure that must be conceptualized as the seed of a robotic brain because it is modular, self-organized, openly-attending, multifunctional and potentially evolving.

A designer can evolve the robot from openly-attending to fetch objects, to navigating, to escape-predation, etc. This can be brought about by adding new SIMs—these modules being modified copies of old modules.

Conceptually, a “more brain-like” mechanism for a robot should be built out of *ad hoc* hardware modules. Robots, so built, would provide the opportunity to explore the behavioural restrictions imposed by the *ad hoc* hardware, harnessed to their electro-mechanics. More important, however, is the fact that these robots would enable us to discover unsuspected emergent behaviours and would lead us to the implementation of desirable unplanned reflexes and behaviours. The orienting-reaction conflict with the running-away behaviour in the experiments reported here is a case of the emergence of a desirable variant of the running-away behaviour appearing after the implementation of a new behaviour.

FUTURE WORK

We think that the Payoff Function in the RM2-ROB can be seen as a simple Elman’s neural network (Elman 1990) and the Decision Function as a simple Jordan’s neural network (Hertz *et al.* 1991). Thus, in a future work the SIM module will be implemented as an Elman-Jordan neural network. The auto-inhibition of the modules in this possible implementation and their organization under subsumption should also be investigated. This future implementation will lower the granularity level to a neural level, a more recognizable feature of brain architecture in the living context.

REFERENCES

- Beers RD. 1990. Intelligence as Adaptive Behavior. An Experiment in Computational Neuroethology. Boston: Academic Press.
- Brooks RA. 1986. A robust layered control system for a mobile robot. *IEEE J Robotics Automat*, Ra-2(1) 14–23.
- Elman J. 1990. Finding structure in time. *Cogn Sci*, 14 179–211.
- Hertz J, Krog A, Palmer R. 1991. Introduction to the Theory of Neural Computation. Redwood City, CA: Addison-Wesley.
- Kaelbling LP. 1993. Learning in Embedded Systems. Cambridge: The MIT Press.
- Maes P. 1990. A bottom-up mechanism for action selection in an artificial creature. From Animals to Animats. In Proceedings of the First International Conference on the Simulation of Adaptive Behavior S Wilson, J-A Meyer Cambridge: The MIT Press, p. 238–46.
- Negrete-Martinez J, Cruz R. 2002. Self-organized multi-modular Robotic Control. In Proceedings of the 3rd International Symposium on Robotics and Automation, ISRA’2002, p. 421–5.
- Parasuraman R. 2000. The attentive brain: issues and prospects. In The Attentive Brain. R Parasuraman Cambridge. The MIT Press, p. 3–15.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

