Chapter 1

INTRODUCTION

What is MoNETA?

The primary goal of a Modular Neural Exploring Traveling Agent (MoNETA) project is to create an autonomous agent capable of object recognition and localization, navigation,
and planning in virtual and real environments. Major components of the system perform sensory object recognition, motivation and rewards processing, goal selection, all centric representation of the world, spatial planning, and motor execution. MoNETA is based on the real time, massively parallel, Cog Ex Machina environment co-developed by Hewlett-Packard Laboratories and the Neuromorphics Lab at Boston University. The agent is tested in virtual environments replicating neurophysiological and psychological experiments with real rats. The currently used environment replicates the Morris water maze.

Chapter 2

LITERATURE SURVEY

The whole brain system called MoNETA (Modular Neural Exploring Traveling Agent, and its first version has been tested in a virtual Morris Water Maze task. The Morris Water Maze is a task used to probe the navigation skills of a rodent. The rat is placed in a water tank and has to use visual clues in order to locate a submerged platform and swim to it. The rat is motivated to find the platform to get out of the water tank.

Researchers have studied this task at great length, and knows a great the brain areas a rat utilizes in completing the task. Although an apparently simple one, solving the water maze actually requires that the integrated functioning of object recognition and localization, goal selection, and navigation be simulated in several interacting brain areas. To build and simulate MoNETA, new software tools and hardware are needed that that implement these models at biological scale. MoNETA, the brain on a chip is the technology that will lead to a true artificial intelligence. MoNETA the software is designing at Boston University's department of cognitive and neural systems, which will run on a brain-inspired microprocessor under development at HP Labs in California.

It will function according to the principles that distinguish mammals most profoundly from our fast but witless machines. MoNETA (the goddess of memory) will do things no computer ever has. It will perceive its surroundings, decide which information is useful, integrate that information into the emerging structure of its reality, and in some applications, formulate plans that will ensure its survival. In other words, MoNETA will be motivated by the same drives that motivate cockroaches, cats, and humans.

Two years ago, HP built a new class of electronic device called a memristor. Before the memristor, it would have been impossible to create something with the form factor of a brain, having the low power requirements, and the instantaneous internal communications.

Basically, memristors are small enough, cheap enough, and efficient enough to fill the bill. Perhaps most important, they have key characteristics that resemble those of synapses.

That's why they will be a crucial enabler of an artificial intelligence worthy of the term. Animat is said to be successfully built, only when MoNETA is motivated to run, swim, and find food dynamically, without being programmed explicitly to do so.

It should learn throughout its lifetime without needing constant reprogramming or needing to be told a priori what is good for it, and what is bad. This is a true challenge for traditional AI: It is not possible to preprogram a lifetime of knowledge into a virtual or robotic animat. Such wisdom has to be learned from the interaction between a brain, with its large (but not infinite) number of synapses that store memories, and an environment that is constantly changing and dense with information.

The animat will learn about objects in its environment, navigate to reach its goals, and avoid dangers without the need for us to program specific objects or behaviors. Such an ability comes standard-issue in mammals, because our brains are plastic throughout our lives. We learn to recognize new people and places, and we acquire new skills without being told to do so. MoNETA will need to do the same.

Chapter 3

MEMRISTOR

A memristor is a passive two-terminal electronic component for which the resistance (dV/dI) depends in some way on the amount of charge that has flowed through the circuit. When current flows in one direction through the device, the resistance increases; and when current flows in the opposite direction, the resistance decreases, although it must remain positive. When the current is stopped, the component retains the last resistance that it had, and when the flow of charge starts again, the resistance of the circuit will be what it was when it was last active

“The memristor is formally defined as a two-terminal element in which the magnetic flux Φm between the terminals is a function of the amount of electric charge q that has passed through the device.”

Chua defined the element as a resistor whose resistance level was based on the amount of charge that had passed through the memristor

Memristance

Memristance is a property of an electronic component to retain its resistance level even after power had been shut down or lets it remember (or recall) the last resistance it had before being shut down.

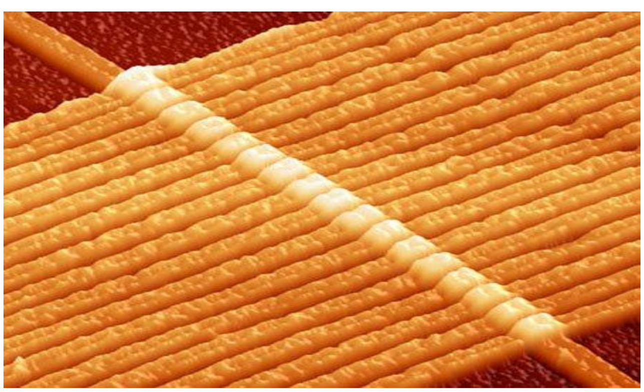


Fig.1.Memristor

Chapter 4

Cog Ex Machina (or Cog)

As a first step in the creation of an animat that could demonstrate visually guided adaptive behavior, researches at BU created an artificial nervous system, MoNETA, based on Cog Ex Machina (or Cog). Cog, built by HP principal investigator Greg Snider, is a neural modeling operating system that lets neural designers interact with the underlying hardware to do neuromorphic computation. Cog abstracts underlying storage hardware and allocates processing resources as required by computational algorithms based on CPU/GPU availability. Cog exposes a programming interface that enforces synchronous parallel processing of neural data encoded as multidimensional arrays (tensors). In this implementation, Cog allows the design of complex brain systems that controls an iRobot Create. To allow the brain models and the neuromorphic hardware to interact, HP built a kind of special-purpose operating system called Cog Ex Machina. Cog, built by HP principal investigator Greg Snider, lets system designers interact with the underlying hardware to do neuromorphic computation. Neuromorphic computation means computation that can be divided up between hardware that processes like the body of a neuron and hardware that processes the way dendrites and axons do.

Cog Ex Machina (Cog), a project lead by Greg Snider at HP Labs, is a software framework the lab uses both as a simulation tool and as a medium for enforcing consistency between modelling and hardware work. As a simulation tool, Cog allows modellers to build their initial prototypes at large scale with minimal difficulty. The dendrite cores in the Cog hardware will be much less flexible than neuron cores, but they will store extraordinary amounts of state information in their massive memristor-based memory banks, and like the tendrils of neurons, they will make up the vast bulk of the system's computational elements. Memristors, finally, will act as the synapses that mediate the information transfer between the dendrites and axons of different neurons. For a programmer, taking full advantage of a machine like this is tremendously challenging, as it has two different core types and complicated memory-storage overlay, the problems need to be properly partitioned across those two radically different types of processors.

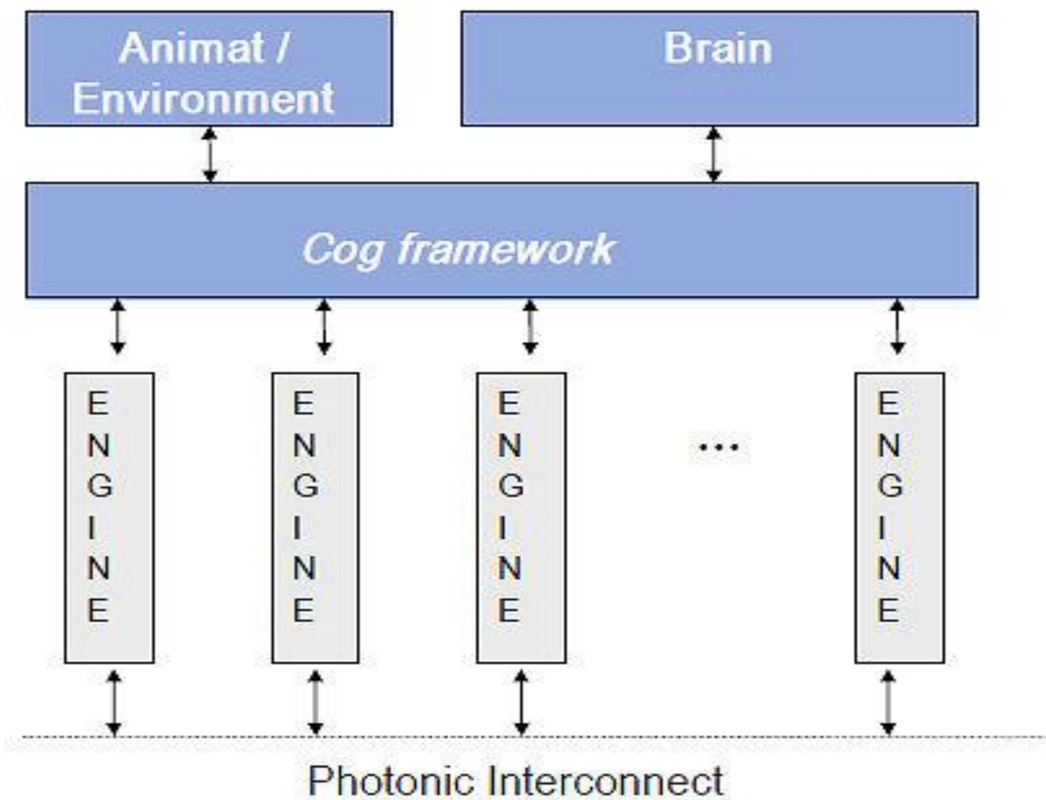


Fig.2. Cog-Ex Machina Architecture

With conventional tools, large-scale simulation requires too much effort to allow rapid exploration**.** Beyond the simplification of large-scale simulation, Cog is a a key product of the joint HP-BU research. Cog captures a year's worth of work towards finding the right abstractions to match modelling needs to hardware capabilities. Both HP and BU have contributed a great deal towards the design assumptions built into the system, so the abstractions are a good fit both for the models we need to build and the hardware we can fabricate. This common set of abstractions is the only way to ensure that the hardware and modelling research are on track to converge to a single physical artefact capable of intelligent behaviour.

Fig.3.Simcity

Large scale simulations, such as the one typical of MoNETA models, will leverage high performance computing resources, such as the new GPU cluster hosted at HP Labs under the direction of Greg Snider and Dick Carter. The cluster, called Simcity, features a total of 144 GPUs, 576 GB of conventional memory, 432 GB of GPU memory, and an infiniband interconnect. A prototype cluster containing three nodes and six GPUs, called Simtown, is also available for testing and debugging.

Chapter 5

Visually Guided Adaptive Robot

The robot, which includes some of MoNETA basics features, explores a world of colour objects. It navigates toward an object if it perceives its color as attractive based on a reward value associated with object color from the animist past interaction history. The animat also learns the locations of objects it has visited in the past in order to avoid these locations in the future exploration of its world.

This behaviour was simulated using Cog 2.0 based brain that communicated with a netbook attached to a robot serial port via WiFi network (see figure 4)

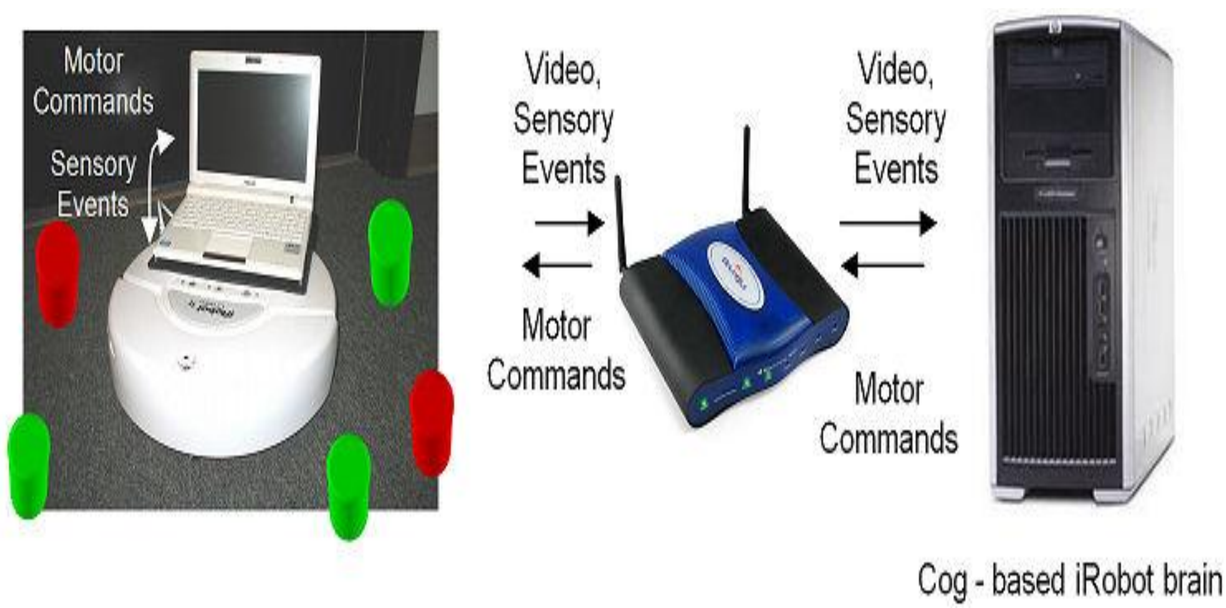


Fig.4. *iRobot Simulation Environment*

A neuromorphic architecture of the Visually Guided Adaptive Robot (ViGuAR) brain is design to support visually guided adaptive navigation in a simplified version of a world that consists of red and green colour objects of fixed size (Fig. 4).

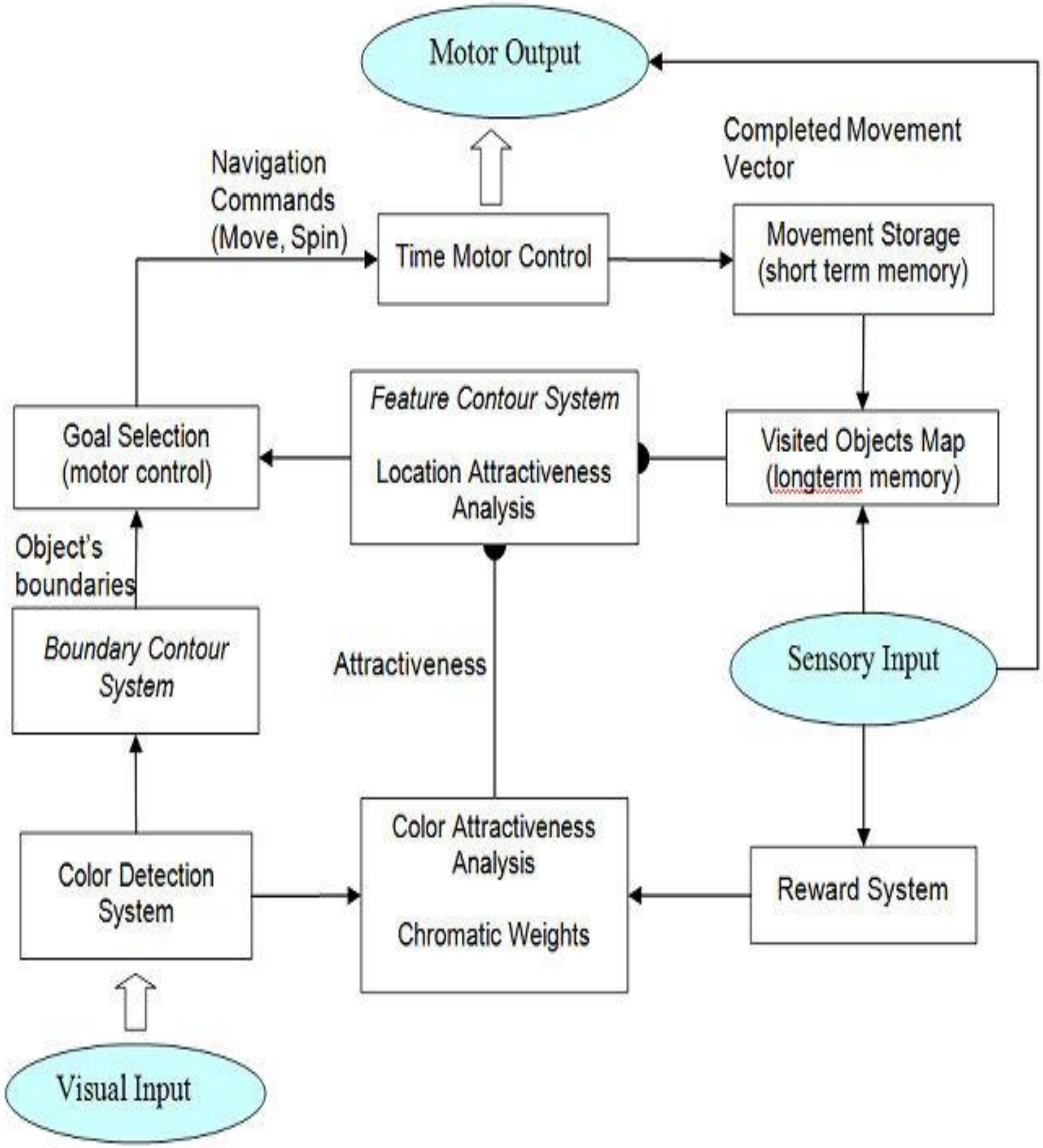


Fig.5. *The ViGuAR brain*

The **Color Detection System** converts RGB input it receives from WEB-camera into chromatic features: redness and greenness for each location.

The **Color Attractiveness Analysis Module** computes the attractiveness for each location based on plastic synaptic weights associated with chromatic features: redness and greenness. This module produces a measure of attractiveness for each spatial location. The synapses associated with chromatic features are adjusted on a signal from the Reward System.

The **Reward System** receives an input from the robot bumper sensors when this contacts an object and produces teaching signals for the color Attractiveness Module to associate chromatic features of the contacted object with the reward.

The **Boundary Contour System** converts the color signal associated with each spatial location into a boundary signal. In a simplified version of ViGuAR brain the discontinuity in color features are recognized as object boundaries, which are sent to the Goal Selection System.

The **Feature Contour System** receives the attractiveness signal for each spatial location. These signals are gated by signals from the **Visited Object Map**, which block the attractiveness for the locations belonging to the objects the animat has already visited.

The **Goal Selection System** analyzes attractiveness at each spatial location in order to find the most attractive goal in its view. It uses the boundary signal to evaluate the distance to a target and generates a proper motion signal. Alternatively, this module may decide to continue collection of attractiveness data by spinning the robot and to collect measure of attractiveness from its surrounding. The Goal Selection System integrates attractiveness data from multiple views in order to select an optimal goal and generate a proper motor command.

The commands are controlled by the **Time Motor Control** unit that maintains and terminates proper motor activity (**Motor Output**) on robot effectors.

The **Movement Storage Unit** updates a vector of movement of the animat upon completion of each movement. Thus, the location of the robot is constantly maintained in a short memory with respect to the initial robot position.

Upon contact with an object, the animat position is used to place object location on the map internally maintained by the **Visited Object Map** module. The signals from the Visited Object Map module are sent back to the Feature Contour System to produce location attractiveness used by the Goal Selection System.

Cog-based Implementation of ViGuAR Brain

The high level diagram of ViGuAR (Fig. 5) was implemented as a Cog-based brain model shown in the (figure 6) . Two independent pathways produce boundary and colour attractiveness information that gets integrated by the Target Selection Module. This model reduces a two-dimensional visual RGB input to a single dimension as the robot seeks to navigate in a horizontal plane. Thus, initial two-dimensional retinomorphic representation of visual information gets squeezed into a single dimension. One-dimensional neural activity is centred to the gaze direction in a robocentric coordinate frame.

ViGuAR maintains two-dimensional space representation memory map. An active entry in this map corresponds to a spatial location that is learned as belonging to an object. This map gets projected into a robot 1D retina to gate visual signal received by the robot in the direction of its head orientation.

ViGuAR performs a search for an attractive object by analyzing attractiveness of a particular space location. This is done independently from the object boundary determination. Once an attractive location is found, the robot orients toward it and integrates attractiveness with object boundaries determined an independent BCS system. This allows the goal selection module to determine the distance that the robot needs to travel to reach the target.

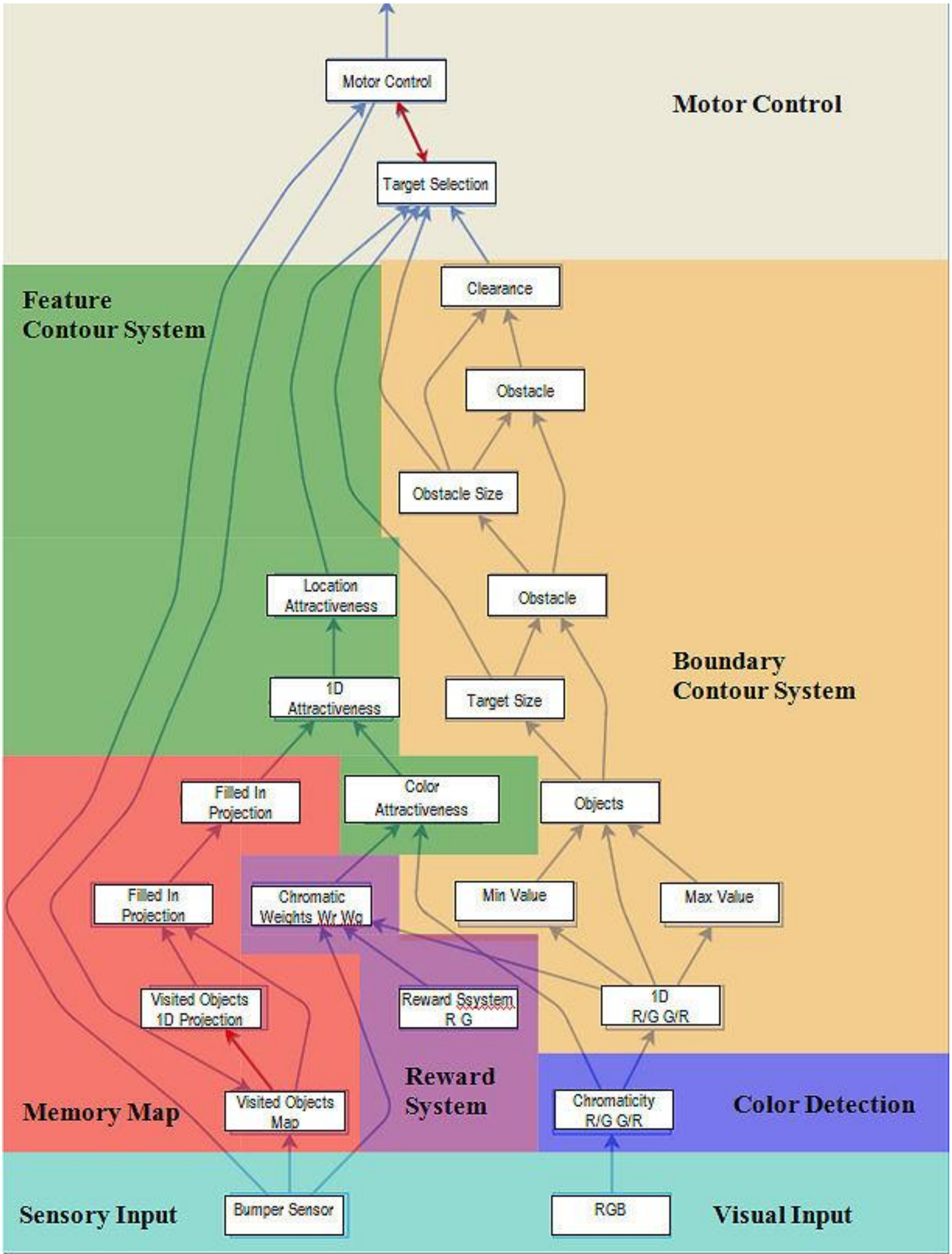


Fig.6. Cog-base ViGuAR Brain

This division of labour is computationally effective in selecting goals evaluation of distance. Its role can be compared with the role attention plays in biological visual systems. Similarly to biological systems, certain locations are primed as not worthy of paying attention. In ViGuAR, this is done by the object map projection to the robot retina. This projection blocks visual signal from the locations iRobot already visited, therefore preventing the robot from reaching them.

Learning occurs upon contact with an object. On contact, the synaptic weights representing colour features forming colour attractiveness learn according to an external reward signal. The reward signal changes the magnitude of the synaptic weights associated with the colour features of the object approached by the robot in positive or negative direction.

Chapter 6

LEARNING IN MoNETA

Single neurons in Cog are implemented based on the following neuron model

$\boldsymbol{y = f(W^T X)}$**..............................1**

where **x** and **w** are the presynaptic input and associated synaptic weight vectors, respectively, f is a scalar-valued activation function, and y is the activation value of the postsynaptic neuron. The product $\boldsymbol{W^T X}$ is referred to as the partial inference.

Cog imposes virtually no restrictions on the choice of the activation function, there by leaving the modeller free to determine the kind of computation performed by each neuron. However, external input is obtained via partial inference only. This restriction ensures that models implemented remain tractable and efficient by removing the need for sophisticated synchronization mechanisms to handle parallelism.

The segregation of computation into a set of partial inferences followed by an activation function is a critical bridge between biological and silicon computation built into Cog. A neural population can maintain relatively little state, but perform potentially highly nonlinear computations. The web of dendrites feeding that population has states stored in each synapse, but computes in a much more rigid manner.

This dual-natured computation maps extremely well to heterogeneous computers. Conventional general-purpose processors can only work on relatively small sets of data efficiently, but include highly robust strategies for handling irregularity and nonlinearity. Special-purpose accelerators like graphics processors are designed for an opposite set of constraints. They require dramatically higher memory bandwidth, but de-prioritize handling of irregular computation. The signal function component of computation is best mapped to a conventional processor, where the partial inference calculation maps efficiently to a graphics processor.

Beyond graphics processors, single-purpose hardware offers multiple additional orders of magnitude in power efficiency. Graphics processors are vastly more efficient than conventional processors for computations like partial inference, but data is still not as physically local as needed for power efficiency rivalling biology. A graphics card includes its own bank of memory with a very wide connection to the processor, but these two components are still physically separated. This separation means data must be shuttled from a memory chip, across a bus, and finally in to the processor. Efficiency would be dramatically higher if memory could be co-located directly on the processor.

Memristive crossbar memory is a viable contender for this unification of processing and memory. Memristor crossbars can be manufactured directly on top of a conventional chip, but with dramatically higher density than existing memory technologies. This means a massively multi-core chip designed to handle partial inference can localize storage of the weights directly on top of the processing core performing calculations .In particular, partial inference and learning - with those weights. Weight data need not move more than a few millimetres, resulting in a massive increase in energy efficiency. Cog's design is in part motivated by such considerations of locality.

In summary, the modelling framework provided by Cog limits communication between neurons to partial inferences and tries to maintain computation related to synaptic weights local to the partial inference processors.

Learning

For efficiently introduce synaptic weight learning in a large-scale model instantiated on a distributed, heterogeneous network ,we restrict learning to a single learning rule which can then be optimized in the same hardware that computes partial inference. From this point of view, one good candidate learning rule might be spike timing- dependent-plasticity since it appears to give the most complete account of biological synaptic modification mechanisms. Indeed, some large-scale models adopt this rule as their only learning mechanism. However, in a system in which biological realism is only secondary to functionality, such a choice is no longer justified.

Moreover, it is fundamentally impossible to determine in advance which learning law is most appropriate without knowing exactly which behavioural task the model needs to perform. Finally, even if the task is known, it is not always clear which learning law will perform best, and if so in what parameter range can it be expected to perform well. Thus, learning in Cog follows the approach of implementing only a generic form of the learning equation which can then be tailored for specific applications. Crucially, the use of a generic form allows for hardware acceleration on the same processors that also compute partial inferences

Current general form of learning laws

Cog currently supports learning laws for which weight changes can be implemented in the following general form

$$\Delta w_{ij} = \lambda s\,(\Delta w_{ij}^{H} + \Delta w_{ij}^{C} + \Delta w_{ij}^{N}) \text{......................................} 2$$

where λ is the learning rate, s is a sign factor (-1 or +1) and Δw_{ij}^{H}, Δw_{ij}^{C}, Δw_{ij}^{N} are weight-change terms related to Hebbian, competitive and normalization operations respectively. Presynaptic and postsynaptic units are respectively denoted in the above equation by indexes *i* and *j*. This general form can encapsulate a number of learning rules performing independent component analysis, and is implemented in Cog as the following sequence of three steps:

$$w_{ij}^{1} \leftarrow w_{ij}^{0} + h_j x_i \text{ ....} 3$$

$$x_i' \leftarrow w_{ij}^{1} h_j \ , \ w_{ij}^{2} \leftarrow w_{ij}^{1} - h_j x_i' \text{ ....} 4$$

$$w_{ij}^{3} \leftarrow w_{ij}^{2} - g_j\, w_{ij}^{2} \text{..} 5$$

Equations 3, 4 and 5 implement the Hebbian, competitive and normalization steps, respectively. Quantities h_j and g_j incorporate the learning rate λ and sign s and are computed by the postsynaptic neuron h_j at each time step. Crucially, the forms of h_j and g_j are determined by the user so as to implement a particular learning rule. Simulation flow in Cog can thus be described as a sequence of two operations performed at each time step. First, all partial inferences are computed and learning is performed based on Equations. 2-5 with feedback termsh_j and g_j returned from postsynaptic neurons computed at the previous time step. Second, all activation functions are computed as well as feedback terms h_j and g_j, which are sent back to the partial inference processors to be used at the next time step.

Toward a generalized learning law equation

In order to allow for a more thorough study of learning in large-scale systems, the general form in Eq. 2 and its associated three-step procedure in Eqs. 3-5 must be generalized to encompass a wider class of learning rules. For example, the *outstar* learning rule ,

$$\boldsymbol{w_{ij}} = \boldsymbol{x_i}\,(\boldsymbol{\lambda y_j} - \boldsymbol{\alpha w_j}(\boldsymbol{t}))\text{................................}\textbf{\textit{6}}$$

where learning is gated by postsynaptic activity y_j and an additional decay rate, α, needs to be specified by the modeller.

It cannot be directly mapped to the existing learning procedure due to the multiplication of weights w_{ij} by the input x_i. As in the case of Hebbian learning with postsynaptic normalization, Equation 6 can be implemented by a suitable modification of the network topology, but this would reduce the efficiency of the framework. Recently a new general form of learning law capable of handling several classes of learning rules is introduced, including Hebb rule derivatives, threshold based rules, input reconstruction-based rules and trace-based rules . Crucially, this generalization is achieved by inserting only one additional postsynaptic feedback term to the already existing two terms (h_j and g_j) of the current procedure, making it a suitable candidate for hardware acceleration. Learning calculations should be local to the partial inference processors, should minimize data transfer to and from signal function processors, and should be implementable in a common general form equation.

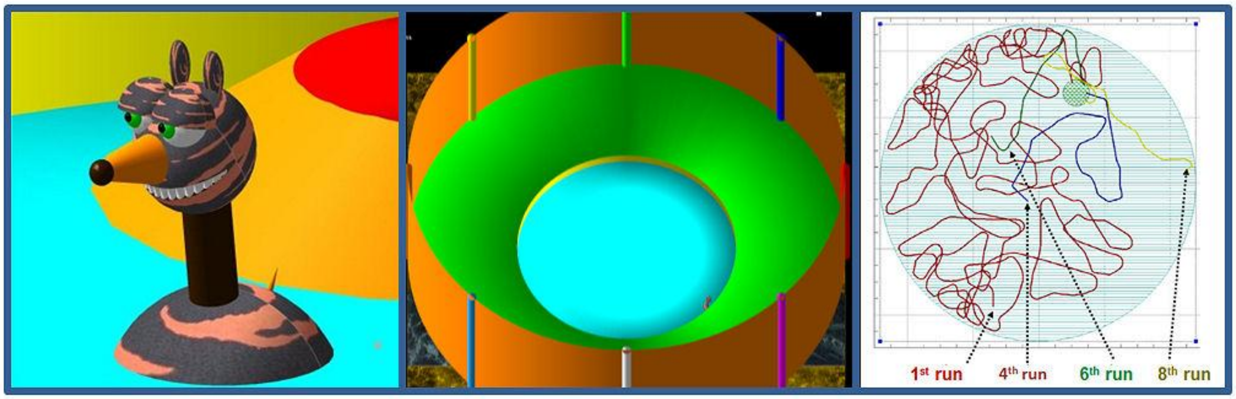
Chapter 7

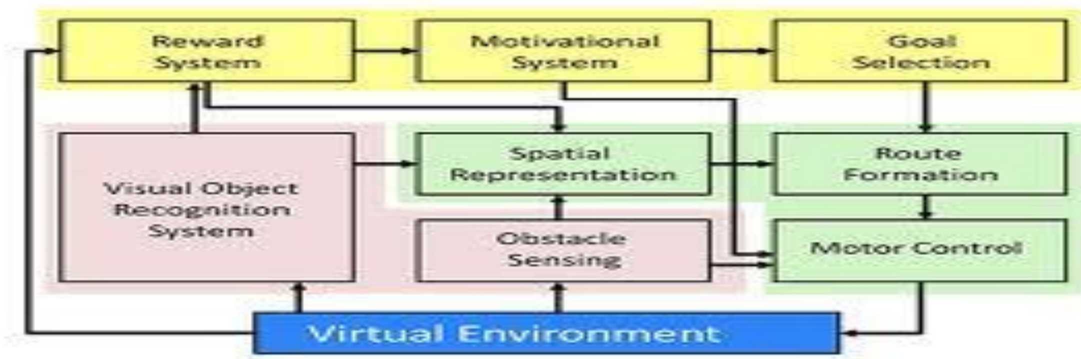
MoNETA Version I

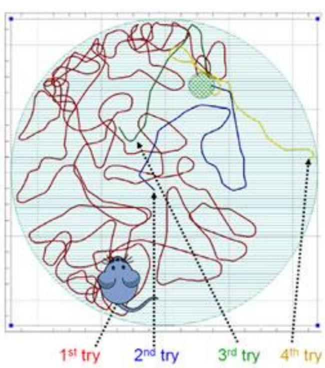
Modular Neural Exploring Traveling Agent (MoNETA) project, supported by DARPA/SyNAPSE via a subcontract with HP, uses Cog to progressively implement complex, whole brain systems able to leverage the power of memristive hardware . MoNETA is the brain of an animat, a neuromorphic agent autonomously learning to perform complex behaviors in a virtual environment. It combines visual scene analysis, spatial navigation, and plasticity.

The system is intended to replicate a rodent’s learning to swim to a submerged platform in the Morris water maze task, a behaviour that involves cooperation among several brain areas. Test animat in a classic trial called the Morris water navigation task. In this experiment, neuroscientists teach a rat to swim through a water maze, using visual clues, to a submerged platform that the rat can't see. That task might seem simple, but it's anything but. To get to the platform, the rat must use many stupendously sophisticated brain areas that synchronize vision, touch, spatial navigation, emotions, intentions, planning, and motor commands. Neuroscientists have studied the water maze task at great length, so they know a great deal about how a rat's anatomy and physiology react to the task. Researchers could train the animat to negotiate this maze, and have taken an important first step toward simulating a mammalian intelligence.

Fig.7.Animat







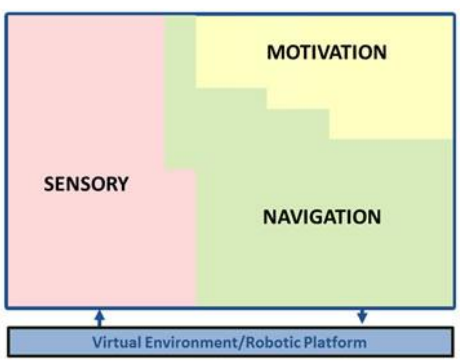


Fig.8. Morris water maze task

The MoNETA brain will eventually implement many cortical and sub cortical areas that will allow an animat or robot to engage with a virtual or real environment. We prepared a proof of concept in a robotic platform (iRobot Create) controlled by a simplified Cog-based MoNETA brain The robot learned to avoid green and approach red objects based on associated reward values. The robot learned not to revisit objects even if they were attractive. This seemingly simple task involved orientation modelling towards a goal, navigation, objects avoidance, sensory processing, motor control, and adaptive learning. It used parallelizable computational threads and tensor data representation results in solutions similar to biological brains, such as layered architecture, parallel processing pathways (for example, what and where pathways), visual-image segmentation, and attentional drive (see Figure 8).

Cog is a scalable, powerful platform for neuromorphic computations that will soon make possible implementation of brain models such as MoNETA comparable in size, power, and behavioural complexity with biological brains. In the context of the SyNAPSE project, researchers will continue developing large-scale, multi-system neural models to be executed on high-density, low-power neuromorphic hardware. These models will be tested in increasingly complex virtual environments as well as on robots with the target of replicating classic experimental results from the rodent behavioral neuroscience literature.

Chapter 8

MoNETA Version II

Researchers at Boston University have completed the first part of MoNETA or Modular Neural Exploring Traveling Agent and are now progressing to version 2.0 of the project. They are going to test the neuromorphic system in a more realistic virtual training ground. This will allow them to see how various organized synthetic neuron configurations function at certain tasks.

Scientists are currently redesigning most of the whole-brain systems components, with a new virtual environment including a powerful physic engine. The MoNETA 2.0 visual system will provide a more powerful system for both classification and navigation in both real and virtual environment. The vision part of MoNETA should support navigation by identifying perceived objects and localizing them within the environment. A distinctive feature of MoNETA vision is that in order to make decisions it uses many principles and inspirations from real animal visual systems.

The academics want to endow their brain with animal like senses and instincts. Some of the operations of the artificial neural networks (ANN) are currently only a shallow mimicry of what organic ones can accomplish. However, perhaps in the future this will change as the devices become more complex.

MoNETA has a number of modules currently under development, including the following:

- space variant vision and eye movements
- color processing
- attentional modulation
- learning and homeostatic mechanisms for synaptic stabilization of all cortical areas
- auditory processing
- basal ganglia,dopamonergic system

- vestibular system

These projects are currently grouped in a new visual, navigation and decision making, and auditory systems, as well as a new virtual environment. Neuromorphic chips won't just power niche AI applications. The architectural lessons we learn here will revolutionize all future CPUs. The fact is, conventional computers will just not get significantly more powerful unless they move to a more parallel and locality-driven architecture. While neuromorphic chips will first supplement today's CPUs, soon their sheer power will overwhelm that of today's computer architectures.

The semiconductor industry's relentless push to focus on smaller and smaller transistors will soon mean transistors have higher failure rates. This year, the state of the art is 22-nanometer feature sizes. By 2018, that number will have shrunk to 12 nm, at which point atomic processes will interfere with transistor function; in other words, they will become increasingly unreliable. Companies like Intel, Hynix, and of course HP are putting a lot of resources into finding ways to rely on these unreliable future devices. Neuromorphic computation will allow that to happen on both memristors and transistors. It won't be long until all multicore chips integrate a dense, low-power memory with their CMOS cores. It's just common sense.

Neuromorphic chips will eventually come in as many flavors as there are brain designs in nature: fruit fly, earthworm, rat, and human. All our chips will have brains.

Chapter 9

CONCLUSION

MoNETA's brain is growing, and like mammalian brains, there are many brain areas that, at one time, fight to take control of a limited set of motor effectors. For instance, curiosity, fear, hunger, visual stimulation, etc. may try to take control of where an animal wants to go, but only one of these representations should take control of driving the animal at a given time. MoNETA v2.0 will integrate mechanisms involving brain areas ranging from sensory, premotor, motor and frontal cortices, to sub cortical areas such as the hippocampus and basal ganglia to implement more complex decision making.

Achieving sentience in a processor is not necessarily the main target of this research. A zombie AI (philosophically speaking) might be able to emulate many emotions or actions of an organism while experiencing no qualia whatsoever. However, building a microchip that could actually perceive things is still the ultimate dream and may be necessary for true artificial general intelligence. Consciousness is a way of driving an organism's behavior in a simple and effective manner. Our brain is constantly carrying out complex calculations that are translated into how we perceive ourselves and the world around us. The main scientists believe that a neuromorphic CPU can do things that aren't possible with conventional architectures.

As a result, an embodied neuromorphic chip would be different than a von Neumann-based chip and, it would be able to generate observable and measurable activity. Such a neural chip would be built to adhere to neurobiological principles of neural structure, organization, and connectivity. A von Neumann chip was not designed to adhere to these biological principles but rather to achieve computational efficiency. Thus, the electrical activity of the actual neuromorphic chip should show some correlation to activity in the human brain. It should be able to self organize into localized regions of activation and show coherent, meaningful oscillations. When work on the brain-inspired microprocessor is complete, MoNETA's first starring role will likely be in the U.S. military, standing in for irreplaceable humans in scout vehicles searching for roadside bombs or navigating hostile terrain.

REFERENCES

1. L. Chua and S. M. Kang, Memristive devices and systems, Proc. IEEE 64 (2), pp. 209–223, 1976.
2. D. B. Strukov, G. S. Snider, D. R. Stewart, and R. S. Williams, The missing memristor found, Nature 453, pp. 80–83, 2008.
3. G. Snider, R. Amerson, D. Carter, H. Abdalla, S. Qureshi, J. L´eveill´e, M. Versace, et al., Adaptive computation with memristive memory, IEEE Comp., 2010. (in press)
4. M. Versace and B. Chandler, The brain of a new machine, IEEE Spectrum 47 (12), pp. 30–37, 2010.

5.http://nl.bu.edu/research/projects/moneta/

6. http://maxversace.com/files/IEEE_Computer_cover_feature_Feb_2011.pdf

7. http://nl.bu.edu/wp-content/uploads/2011/03/Homeostasis_paper.pdf

8. http://maxversace.com/files/Versace_Chandler_IEEE_Spectrum_December_2010.pdf

9. http://www.maxversace.com/files/Silicon-Brains.pdf

10. http://www.ine-news.org/pdf/003500/003500.pdf

11. http://nl.bu.edu/wp-content/uploads/2011/03/IJCNN_2011_-Livitz_et_al_ViGUAR.pdf

12. http://nl.bu.edu/wp-content/uploads/2011/03/IJCNN_2011_Gortchecnikov_et_al_General_form.pdf