A Multi-level Approach to Biomimetic Robots: From Schemas to Neural Networks [1]

Alfredo Weitzenfeld
Computer Engineering Department

Rio Hondo #1, San Angel Tizapan, CP 01000, Mexico DF, MEXICO

Instituto Tecnologico Autonomo de Mexico
email: alfredo@itam.mx

Abstract

Biology has been an important source of inspiration in building adaptive autonomous robotic systems. Through experimentation and simulation, scientists are better able to understand the underlying mechanisms, both structural and behavioral, involved in living organisms. Experimentation, in the form of data gathering (ethological, physiological and anatomical), feed theoretical models that, through simulation, generate predictions to be validated by further experimentation in both robots as well as living organisms. Due to the inherent complexity of these systems and the resulting architectures, most biologically inspired robotic systems are ethological, i.e. behavior is described by higher-level processes; as opposed to neuroethological, i.e. behavior mapped to underlying biological neural dynamics. Yet, neural mechanisms are crucial in modeling biological mechanisms such as adaptation and learning. The work presented here describes a multi-level schema and neural network based approach to modeling biomimetic robotic systems.

Key words: robot, biomimetic, schema, neural.

1. INTRODUCTION

The study of biological systems comprises a cycle of biological experimentation, computational modeling and robotics experimentation as depicted in Figure 1. This cycle serves as framework for the study of the underlying neural mechanisms responsible for animal behavior.
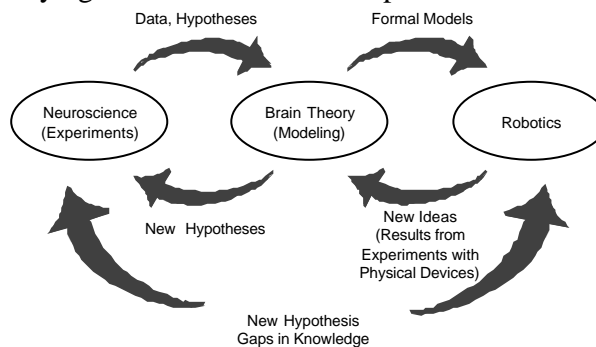

Figure 1. Framework for the study of living organisms through cycles of biological experimentation, computational modeling, and robotics experimentation.

While many living organisms have been studied as inspiration to robotic architectures, the most common approach involves ethological robotics [6], intended to imitate animal behavior without any linkage to neural structure. On the other hand, neuroethological architectures are intended to imitate both biological neural structure as well as behavior. To address the underlying complexity in simulating and building neuroethological robotics systems we usually distinguish between two different levels of modeling, behavior (*schemas* [3]) and structure (*neural networks* [2]). It is important to highlight that while many robot architectures do incorporate some kind of neural processing, most of these are of the artificial neural kind involving non-biological training algorithms, such as back-propagation or reinforcement learning [35].

2. BIOLOGICALLY-INSPIRED ROBOTICS: A MULTI-LEVEL APPROACH

To address the underlying complexity in building biologically inspired robotic systems we have developed a multi-level analysis approach integrating across three primary modeling and simulation levels: (1) embodiment, (2) behavior, and (3) neural networks.

1. At the highest level, embodiment is designed to provide robots with the ability to interact with the real world, mainly in the form of sensors and actuators. The types of sensors and actuators help define the arenas and tasks where the robot may perform. Biologically inspired robots are exemplified by the computational frog (*rana computatrix*) [1], the computational praying mantis [7][13], the computational cockroach [10], and the computational hoverfly [15].
2. At the behavioral level, neuroethological data from living animals is gathered to study the relationship between embodied entities and their environment, giving emphasis to aspects such as cooperation and competition between them. Examples of *ecological* behaviors include the praying mantis *Chantlitaxia* ("search for a proper habitat") [8], and the frog and toad prey acquisition and predator avoidance models [16]. We describe behavior in terms of perceptual and motor schemas decomposed and refined in a recursive fashion in such a way that complex behaviors can be described as the composition of simpler ones. Behaviors and schemas can be modeled with the Abstract Simulation Language ASL [37].
3. At the neural network level, neuroanatomical and neuronphysiological data are used to generate perceptual and motor neural network models corresponding to schemas developed at the behavioral level. These models try to explain the underlying mechanisms for sensorimotor integration in, for example, visually guided animals [25][41]. Examples of neural network models are tectum and pretectum-thalamus responsible for discrimination among preys and predators [12], corresponding to behaviors such as prey acquisition and predator avoidance in toads [14][18] and higher-level models such as the monkey oculomotor system controlling eye saccades in monkeys [21]. Neural networks can be modeled with the Neural Simulation Language NSL [39].

2.1 Embodiment: Animals and Robots

To simplify the development process, models are first simulated in a virtual world and then embodied and executed in the real world. This process starts by specifying a set of sensors and actuators that should correspond as close as possible to those found in animals. For example, frogs and toads use vision and tact as primary sensors while legs and tongue are their primary actuators. In some cases actuators such as wheels can be used in building robots without

affecting the resulting behavioral interaction with the environment. In Figure 2 we show an illustration of a frog scene corresponding to a prey acquisition with detour behavior consisting of a frog and a prey (worm) interposed by a fencepost.
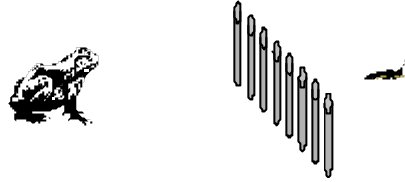

Figure 2. Computational frog in a prey and barrier setup.

2.2 Behavior: Schemas and ASL

There are a number of ways of describing animal behavior. In Figure 3 we show *rana computatrix* ethograms for mating, prey acquisition, predator and static object avoidance.
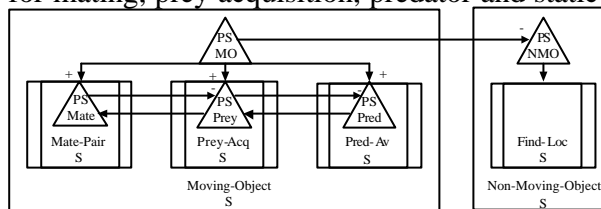

Figure 3. *Rana Computatrix* ethogram: Mating, Prey Acquisition and Predator Avoidance schemas (moving and non-moving objects) [12]. The diagram shows feedback between perceptual schemas (triangles) and regular schemas (rectangles). Note the hierarchical schema organization. (Acronyms are as follows: PS - Perceptual Schema, MO - Moving Object, NMO - Non-Moving Object, S - Schemas)

Behavior can also be described in terms of stimulus-response diagrams. In Figure 4 we show a typical frog prey acquisition behavior using a stimulus-response diagram.
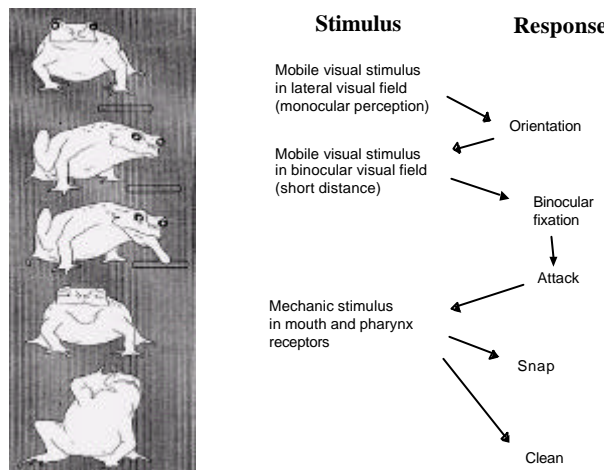

Figure 4. Frog's prey acquisition behavior involving a worm as shown on the left-hand side. The right-hand side describes the frog's response in relation to the stimulus [24].

Schemas

In order to model complex biological systems linking behavior with neural structure a schema computational model is defined in terms of schema hierarchies representing a distributed model for action-perception control [34][40]. For example, diagram in Figure 5 represents a schema

computational model for prey acquisition, predator and static object avoidance in frogs, where blocks correspond to schemas and arrows represent data flow between schemas.
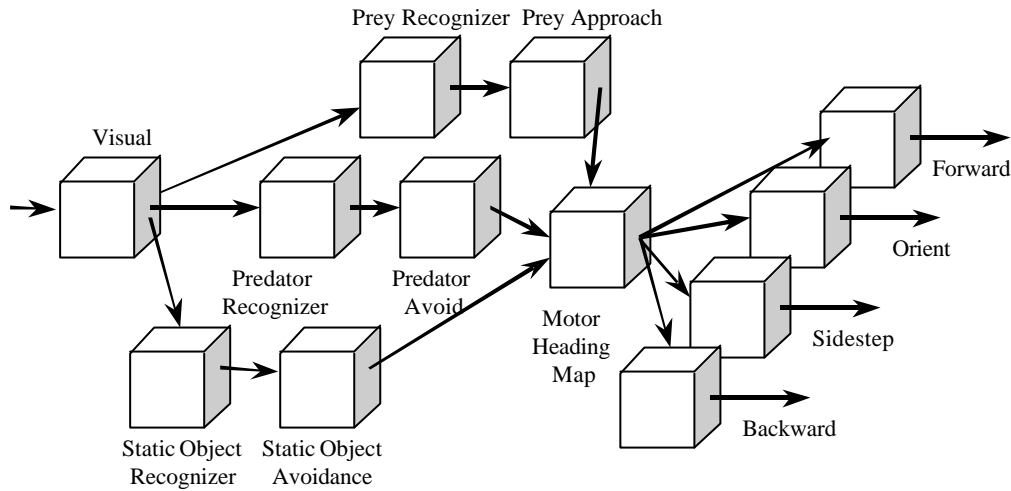

Figure 5. The diagram shows a schema computational model for frog behavior, including prey acquisition, predator and static object avoidance. Each block in the diagram represents a schema, where arrows correspond to data flow between schemas.

   Communication between schemas can be of a cooperative or competitive nature and may involve data transmission as well as *schema assertion* [18], in other words, weighing schemas to discriminate between opposed behaviors, such as prey acquisition and predator avoidance. Schema assertion takes place when schema activity surpasses certain threshold hence indicating enough confidence on that particular schema in the particular context. Once asserted, schema output is produced.

ASL – Abstract Schema Language
The schema computational model is specified using the ASL – Abstract Schema Language defining a hierarchical distributed computational model, where each schema incorporates its own structure and control mechanisms, as shown in Figure 6. In the top portion of the diagram, a higher-level schema is decomposed into two lower level schemas where the three schemas together are known as a schema *aggregate*, or *assemblage.* When at the same level, schemas can be *interconnected* (solid arrows), or when at different levels, schemas can be *relabeled* having their task *delegated* (dashed arrows).
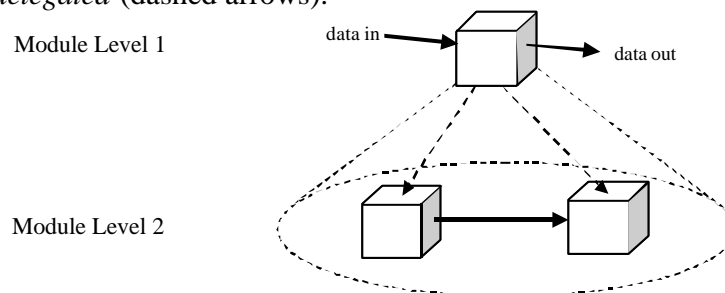

Figure 6. The ASL computational model is based on hierarchical interconnected schemas. A schema at a higher level (level 1) is decomposed (dashed lines) into additional interconnected (solid arrow) subschemas (level 2).

   The schema computational model follows a tree-like structure (schemas may also communicate between layers making the structure a directed graph). At the higher abstraction levels, detailed schema implementations are left unspecified, only specifying what is to be

achieved by lower level schemas. The schema interface consists of multiple unidirectional control/data, input and output ports together with a schema implementation body, as shown in Figure 7. Schemas correspond formally to *port automata* with activity variables indicating the degree of confidence [1].
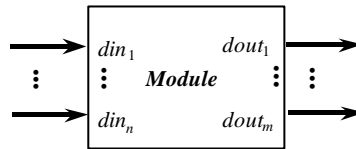

Figure 7. Each schema may contain multiple input, $din_1,...,din_n$, and output, $dout_1,...,dout_m$, ports for unidirectional communication.

Communication is in the form of asynchronous message passing, hierarchically managed, internally, through anonymous port reading and writing, and externally, through dynamic port *connections* and *relabelings*. Schemas are interconnected by matching schema interfaces, in other words, *connections* are done by linking *output ports* from one schema to *input ports* in other schemas. On the other hand, *relabelings* are done by linking ports of similar type (input or output) among different schemas usually at different levels in the hierarchy. The hierarchical port management methodology enables the development of distributed architectures where schemas may be designed in a top-down and bottom-up fashion implemented independently and without prior knowledge of the complete model or their final execution environment, encouraging component reusability. In the top-down approach a complete system is first described at the schema level, as shown in Figure 5. In the bottom-up approach individual schema is first developed and then integrated in creating more complete schema systems.

2.3 Neural Networks: Neuroethology and NSL

Biologically inspired neural networks are based on physiological and anatomical neural mappings. For example, Figure 8 shows a diagram of different neural areas involved in the frog's prey acquisition, predator avoidance and static object avoidance.
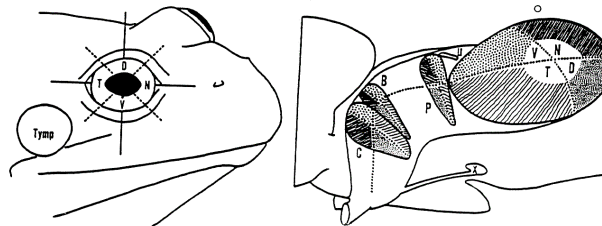

Figure 8. The two illustrations show the most important areas in the frog's prey acquisition model. These are the Optic Tectum (O) (divided in four regions: Temporal (T), Dorsal (D), Nasal (N) and Ventral (V)), the Thalamic Pretectal Neuropil (P), together with other regions: Nucleus of Belonci (B), Lateral Geniculate Nucleus (C) and Basal Optic Root (X) [32].

Neuroethology

In order to develop a multi-level neural schema model, it is necessary to have a correspondence in animal behavior to neural network structure. In Figure 9 a number of neural networks are incorporated: *Retina* [36], *Stereo* [29], *Maximum Selector* [20], *Tectum* and *PreTectum-Thalamus* [12], together with neural *motor heading maps*. These neural schemas correspond each to a different higher-level schema, in particular, *prey approach*, *predator avoidance* and *static object avoidance*. Additional schemas in the model include *visual* and *tactile* input, *depth* and *moving stimulus selector* (when more than one prey exists), *prey*, *predator* and *static object*

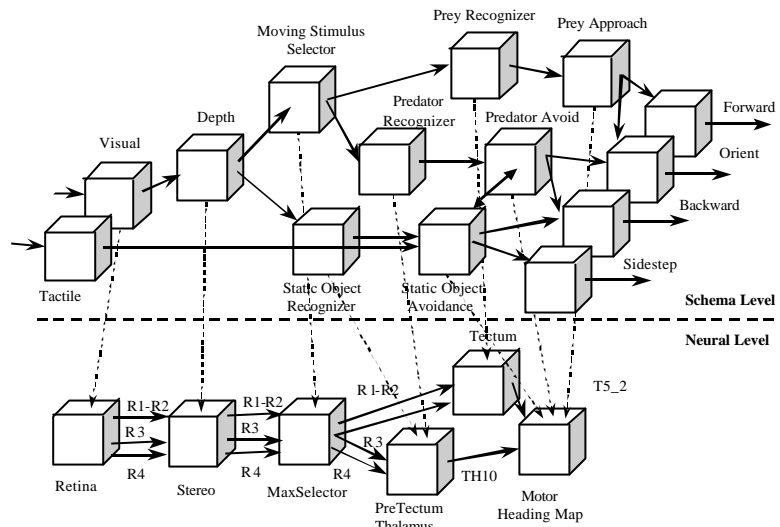recognizers together with the four types of motor actions: *forward*, *orient*, *sidestep* and *backward*.

Figure 9. Toad's prey-predator visuomotor coordination model architecture with schema and neural level modules. The schema architecture consists of two levels: a schema level and a neural networks level. The schema level consists of Perceptual Schemas: Visual and Tactile, Prey Recognition, Static Object Recognition (*SOR*); Sensorimotor Schemas: Prey Approach and Static Object Avoidance; and Motor Schemas Orient, Forward, Sidestep and Backup. The neural level consists of four modules: Retina, T5_2layer, TH10layer and the Motor Heading Map (*MHM*).

NSL – Neural Simulation Language

The NSL – Neural Simulation Language is integrated as part of the ASL system with, where NSL is used to model individual neural schemas. Each schema in the ASL model may have its task delegated to a neural schema or other process, as shown in Figure 10.
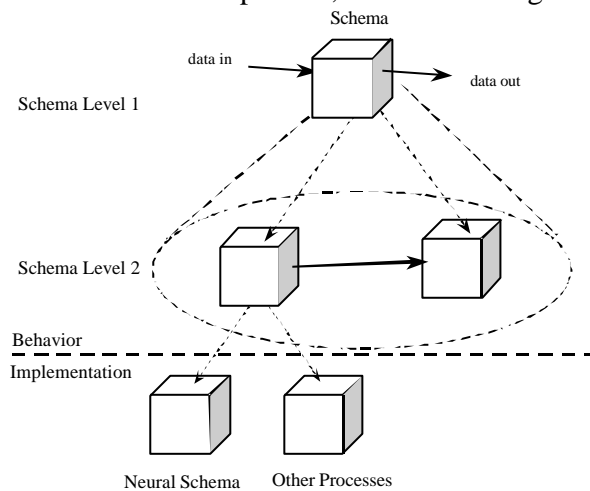
Figure 10. The ASL/NSL computational model refines lower level schemas into neural networks corresponding to neural schemas or other processes.

Each neural schema may be described at different levels of detail, from simpler neuron models to very detailed ones [38], as shown in Figure 11, such as electrochemical mechanisms responsible for phenomena such as synaptic plasticity, compartmental models where a single axon is divided in compartments [31], or ion kinetics model where chemical concentrations responsible for electric current are described [28]. These models are simulated with systems

such as GENESIS [11] and NEURON [27].

Figure 10. Neural schema hierarchy showing task delegation to neural networks processing.
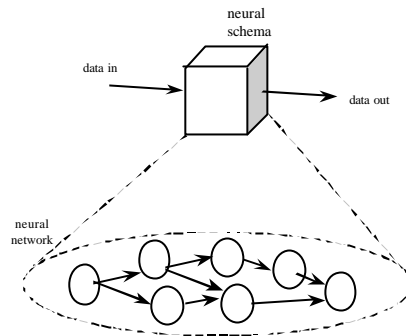
   Simpler neuron models, such as the *leaky integrator* neural model [2], are best suited for large-scale computation, where each neuron is defined by a *membrane potential* with value m representing its previous history, input s_m and output value M defined by a non-linear threshold function over its membrane potential, as shown in Figure 11.
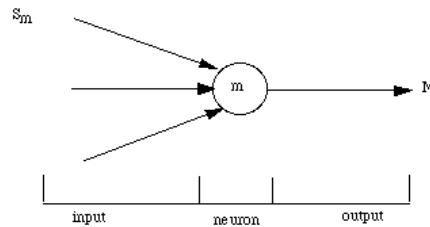

Figure 11. Simple neural element as basic component at the neural network level.

   For example, Figure 12 shows the MaxSelector [20] neural model described in terms of multiple leaky integrator neurons.
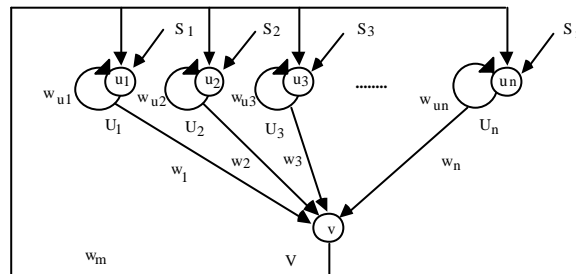
Figure 12. The neural network shown corresponds to the architecture of the Maximum Selector model, where u_i and v represent neural membrane potentials, U_i and V represent neural firing rates, S_i represent inputs to the network, and w_i represent connection weights. The network is initialized with a number of positive inputs assigned to different cells. After multiple iterations the network stabilizes producing a single "winner", i.e. a single active cell.

3. LEARNING TO DETOUR: A MULTI-LEVEL NEURAL-SCHEMA MODEL

Anurans, such as frogs and toads, show quite flexible behavior when confronted with stationary objects on their way to prey or when escaping from a threat. *Rana computatrix* [1], a biomimetic system representing anuran visuomotor coordination, models complex behaviors such as detouring around a stationary barrier to get to a prey by appropriate barrier recognition, depth perception, and motor pattern generation mechanisms based on sensory perception [18].

3.1 Model Background

Ingle [30] and Collett [17] have observed that a frog or toad approaching a prey or avoiding a predator is affected by the stationary objects surrounding the animal. A frog or toad, viewing a vertical fence barrier through which it can see a worm, may either approach directly to snap at the worm, or detour around the barrier. However, if no worm is visible, the animal does not move. Thus, the worm triggers the animal's response but, when the barrier is present, the animal's trajectory to the worm changes in a way that reflects the relative spatial configuration of the worm and the barrier. Different behavioral responses to different barrier configurations described in [18] are shown in Figure 13.
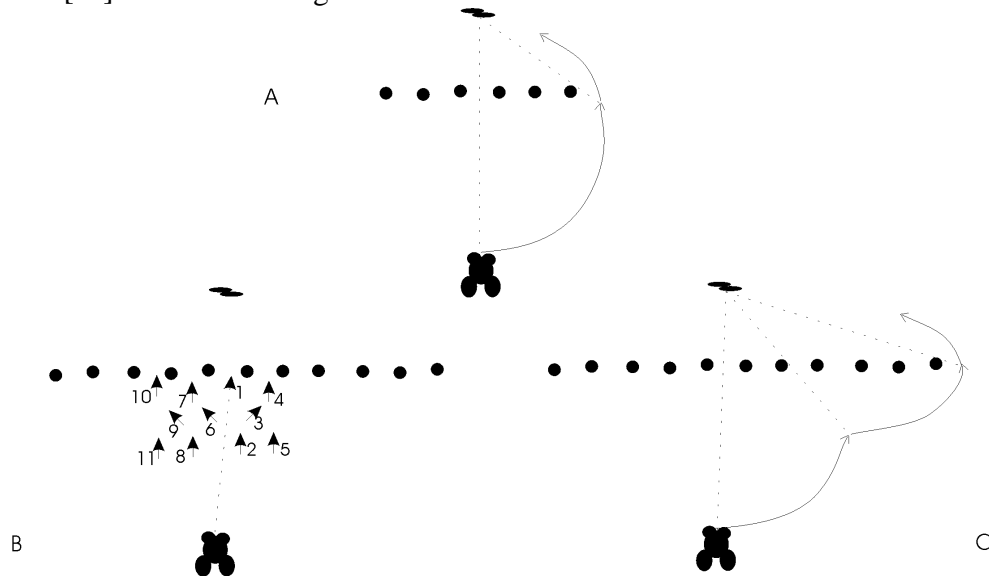

Figure 13. **A.** Approach to prey with single 10cm barrier with immediate detour. **B.** Approach to prey with single 20 cm barrier: first trial with frog in front of 20cm barrier (numbers indicate the succession of the movements). The toad directly approaches de center of the barrier requiring successive trials to manage the detour around it. **C.** Approach to prey with single 20cm barrier. After 3 trials the frog detours directly around the 20cm barrier. Arrowheads indicate the position and orientation of the frog following a single continuous movement after which the frog pauses.

The following experiments are performed [18]:
- **Experiment I: Barrier 10cm Wide.** A 10cm wide barrier with the toad starting from a long enough distance (15-25cm) in front of the barrier and the worm 10cm behind the barrier. The experiment shows (in 95% of the trials) reliable detour behaviors from the first interaction with the 10cm barrier producing an immediate approach towards one of the edges of the barrier.
- **Experiment II: Barrier 20 cm wide.** A 20cm wide barrier where the "naïve" toad (a toad that has not been yet exposed to the barrier) tends to go towards a fencepost gap in the direction of the prey (this was the case for 88% of the trials). The toad initially approaches the fence trying to make its way through the gaps. During the first trials the toad goes straight towards the prey thus bumping into the barrier. Since the toad is not able to go through a gap it backs-up about 2cm and then reorients towards one of the neighboring gaps.
- **Experiment III: Barrier 20 cm wide. After learning.** A 20cm wide barrier where the "trained" toad, after 2 (43%) or 3 (57%) trials, is already detouring around the barrier

without bumping into the barrier. The behavior involves a synergy of both forward and lateral body (sidestep) movements in a very smooth and continuous single movement.

3.2 Model Architecture

In order to define the model architecture it is necessary to define the robot-environment synergy. In our example, the robot embodiment consists of visual and tactile input in the form of a 2d visual field projected upon the retina of the robot.

The detour model incorporates a number of schemas and neural modules as implemented in NSL [19] and shown in Table 1, in correspondence to schemas previously shown in Figure 9.

Function	Schema Level Modules	Neural Level Modules
Perceptual	Visual, Depth, Tactile, PreyRec, SoRec	Retina, T5_2layer, TH10layer
Sensorimotor	PreyApproach, SoAvoid	Motor Heading Map (MHM)
Motor	Forward, Orient, Sidestep, Backup	

Table 1. Frog schemas according to their functional and neural structure organization.

Perceptual Schemas

Perceptual schemas involve sensor and recognition components:

- **Visual**. Obtains visual information from the environment in the form of single or multiple eyes or cameras. It performs basic visual field processing, discriminating between visual elements such as prey and barrier. Has linkage to the Retina neural schema.
- **Depth**. Generates a depth map for objects of interest in the environment. Depth perception is important in avoiding hitting objects and generating appropriate responses according to their closeness.
- **Tactile**. Gets triggered when robot hits an object, in particular, to detect bumping of the barrier.
- **PreyRec**. The presence of a prey within the visual field of the animal produces a 2d pattern of activity in the prey recognition system, while absence of prey leaves the system at rest. The PreyRec schema uses feature detectors to approximate the functionality of a detailed neural network model for prey recognition in amphibia as described in [12], where cells in the toad's pretectum continuously discharge in the presence of large dark stationary objects [23]. Has linkage to the *T5_2layer* neural schema.
- **SoRec**. The SorRec schema model for stationary object recognition in anurans is based on th10 cells [5]. Has linkage to the TH10layer neural schema.

Sensorimotor Schemas

Sensorimotor schemas integrate sensory perception and motor action:

- **Prey Approach** This schema involves a prey attractant field whose strength decays gradually with the distance from the prey [29], projecting this excitatory field onto the *MHM* (motor heading map).
- **Static Object Avoid**. The model includes a repellent vector field associated with each fence post. Its effect is more localized to its point of origin than that of the prey. It projects this inhibitory field onto the *MHM* (motor heading map).
- **Bump Avoid**. The *BumpAvoid* schema produces a reorientation that triggers the projection of an activity pattern to the *MHM*. This field gives rise to excitation on neighbor regions resulting in reorientation under bumping.
- **Motor Heading Map**. The motor heading map (*MHM*) determines the direction to jump [16]. Projections to the *MHM* must differ depending on whether a visual stimulus is identified as prey, predator or obstacle. In the latter, the sensory map and the motor map

must be distinguished, in other words, direction of a prey and prey catching are the same, but directions of a predator and escape are different. Winner-take-all dynamics over *MHM* assure the selection of the strongest target angle, upon which a transformation from retinotopic to motor coordinates takes place. This is the input to the different motor schemas [18][20].

Motor Schemas

In the model, motor schemas (sidestep, orient, approach, snap, etc.) are implemented as functional components schematizing the neural interactions underlying behavior. The intrinsic motor patterns or muscle activations are not simulated. Detour behavior can be seen as the coordination of motor schemas. Ingle [30] offered some clues as to possible neural correlates of the various schemas. Apparently, thalamic and tectal visual mechanism can operate somewhat independently. Monocular frogs without a contralateral optic tectum can quite accurately localize barriers, and while visual input to the pretectal region of the caudal thalamus mediates barrier avoidance behavior, caudal thalamic lesions produce an inability to sidestep stationary barriers set in the frog's path during pursuit of prey. The motor pattern selection is obtained from the interaction of motor schemas[16]. *MHM* contains target location but motor schema selection is the result of competition of many maps.

3.3 Model Simulation

The following experiments were simulated varying the barrier size (10cm and 20cm) as well as applying learning to the 20cm barrier experiment.

Experiment I

A number of internal fields are shown in Figure 14 for experiment I, consisting of a 10 cm wide barrier.
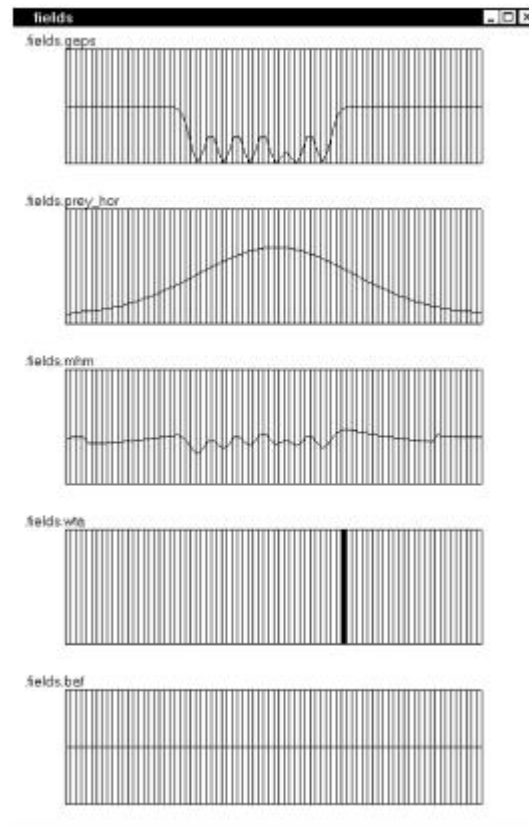
Figure 14. Different activity fields for the 10cm barrier experiment due to *visual_field* processing in the frog with the exception of the bottom one processed after the *tactile* field. The top display (*gaps*) shows the repulsive field generated from the barrier (note that it is negative). The next display down (*prey_hor*) represents the attraction field generated from the prey (note that it is positive). The next display down (*mhm*) represents the combined *gaps* and *prey_hor* fields. The next display down (*wta*) represents the winner-take-all element from the above *mhm* field. This winning element results in the heading or frog's orientation when moving forwards. The last display (*baf*) is currently empty and represents activity due to bumping against the barrier.

The movement direction results from the combination of the prey attraction and barrier repulsion fields. In this experiment the direction of movement is towards the side of the barrier, heading towards the right since the frog was positioned just a bit to the right from the axis joining the center of the prey and barrier. The resulting path motion is shown in Figure 15.
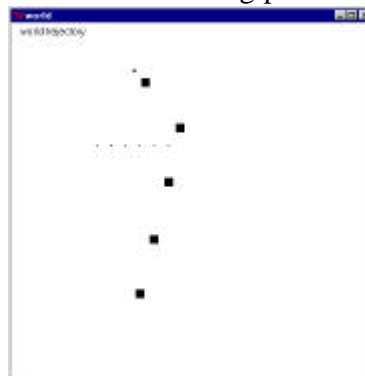

Figure 15. *Rana Computatrix* interacting with the 10 cm wide barrier. The different dots correspond to the frog's trajectory from its initial location as it finally reaches the prey. Note how the frog heads itself towards the side of the barrier.

Experiment II

A number of internal fields are shown in Figure 16 for experiment II, consisting of a 20 cm wide barrier.
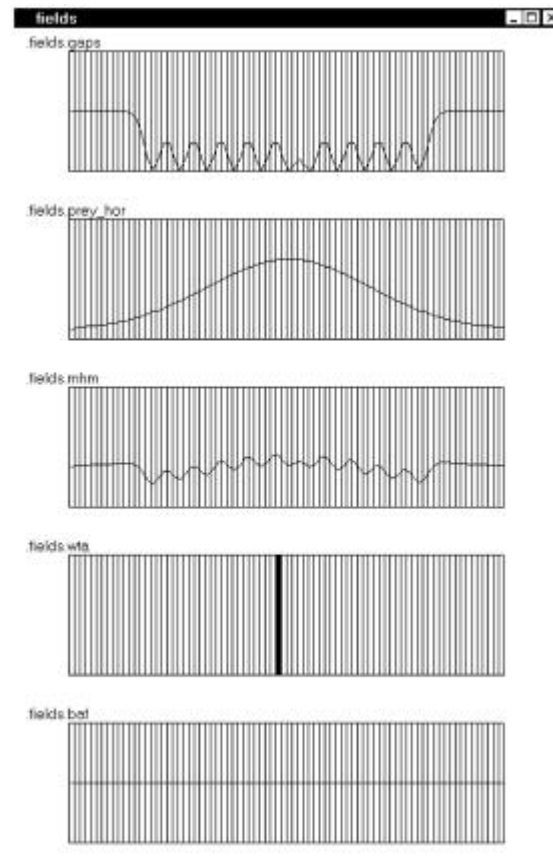

Figure 16. Different activity fields for the 20cm barrier experiment before bumping due to *visual_field* processing in the frog with the exception of the bottom one processed after the *tactile* field. The top display (*gaps*) shows the repulsive field generated from the barrier (note that it is negative). The next display down *prey_hor*) represents the attraction field generated from the prey (note that it is positive). The next display down (*mhm*) represents the combined *gaps* and *prey_hor* fields. The next display down (*wta*) represents the winner-take-all element from the above *mhm* field. This winning element results in the heading or frog's orientation when moving forwards. The last display (*baf*) is currently empty and represents activity due to bumping against the barrier.

Again, the most important factor in the frog movement direction results from the combination of the prey attraction and barrier repulsion fields. In this experiment the direction of movement before bumping into the barrier is towards the middle of the barrier. Once the frog hits the barrier a bumping field is generated. The purpose of this field is to redirect the movement towards a different heading. Before that occurs the frog will backup. The resulting field after bumping is shown in Figure 17.
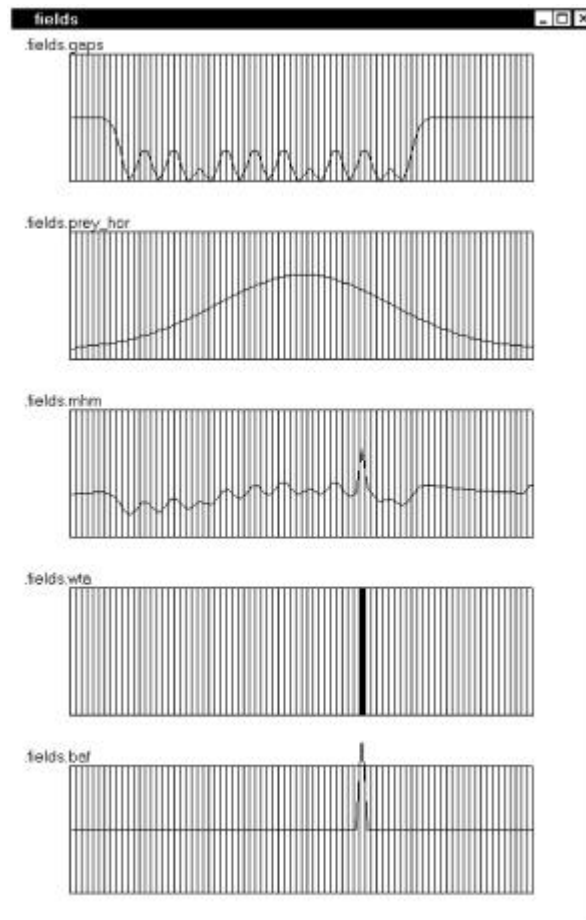
Figure 17. Different activity fields for the 20cm barrier experiment after bumping due to *visual_field* processing in the frog with the exception of the bottom one processed after the *tactile* field. The top display (*gaps*) shows the repulsive field generated from the barrier (note that it is negative). The next display down (*prey_hor*) represents the attraction field generated from the prey (note that it is positive). The next display down (*mhm*) represents the combined *gaps* and *prey_hor* fields. The next display down (*wta*) represents the winner-take-all element from the above *mhm* field. This winning element results in the heading or frog's orientation when moving forwards. The last display (*baf*) is represents activity due to bumping against the barrier.

The resulting path motion after hitting the barrier several times is shown in Figure 18.

Figure 18. *Rana Computatrix* interacting with the 20 cm barrier before learning. The different dots correspond to the frog's trajectory from its initial location as it finally reaches the prey. We have added numbers corresponding to the frogs position in time. In this experiment the frog hits the barrier three times before perceiving the side of the barrier.

Experiment III

A number of internal fields are shown in Figure 19 for experiment III, consisting of a 20 cm wide barrie. after learning.
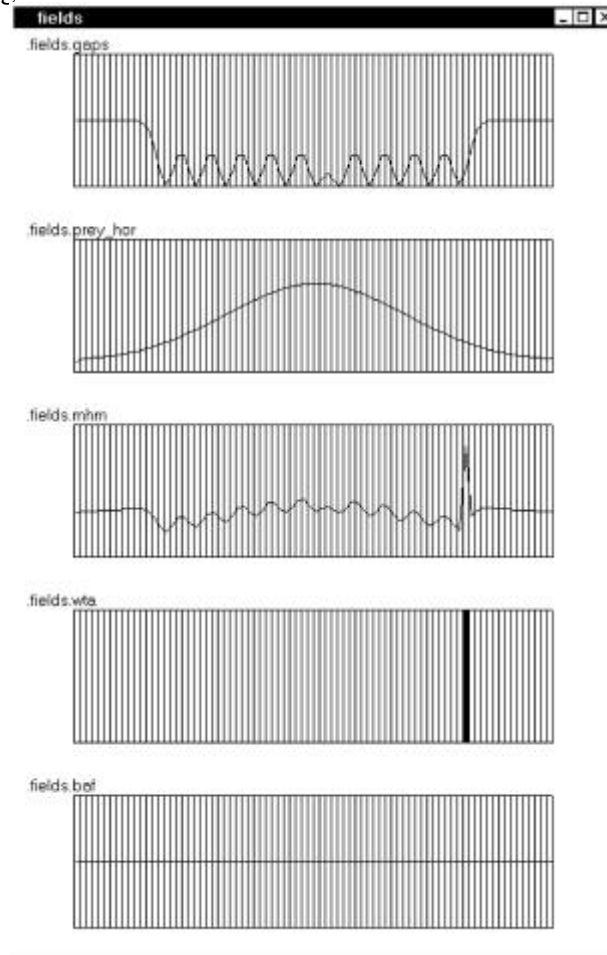

Figure 19. Different activity fields for the 20cm barrier experiment after learning due to *visual_field* processing in the frog with the exception of the bottom one processed after the *tactile* field. The top display (*gaps*) shows the repulsive field generated from the barrier (note that it is negative). The next display down (*prey_hor*) represents the attraction field generated from the prey (note that it is positive). The next display down (*mhm*) represents the combined *gaps* and *prey_hor* fields. The next display down (*wta*) represents the winner-take-all element from the above *mhm* field. This winning element results in the heading or frog's orientation when moving forwards. The last display (*baf*) is currently empty and represents activity due to bumping against the barrier.

Note that although no bumping occurs, the *mhm* field involves a similar integration where heading is explicitly generated, in this case by learning. The resulting behavior is shown in Figure 20.
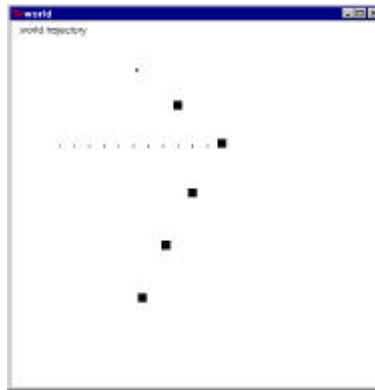
Figure 20. *Rana Computatrix* interacting with the 20 cm wide barrier after learning.

3.4 Robot Embodiment

The experimental setup consists of a mobile robot with wireless trans mission to a remote computer where model computation takes place. The system itself is completely autonomous, although physically distributed to restrict robot size, power consumption and hardware capabilities, making it simpler to move between simulated or embodied systems. The distributed Internet-based robot architecture, known as MIRO - Mobile Internet Robots [43], is shown in Figure 21. Note that multiple robots can execute concurrently, although no sharing of information is allowed between remote computer systems.
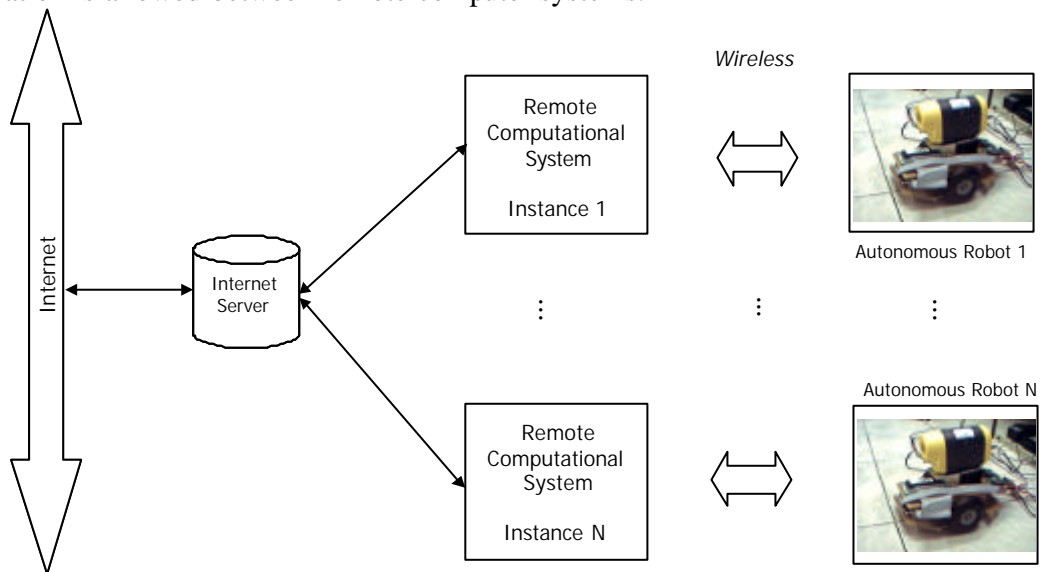
Figure 21. MIRO [43] embedded robotic architecture consisting of multiple autonomous robots linked to their own instance of the distributed neural computational system. All such instances are connected to Internet for remote monitoring alt hough no sharing of information is allowed between systems.

The distributed architecture consists of a remote computational system linked to a PC and a robot having a wireless camera transmitting video to the remote frame grabber in the computer where the model executes. The robot incorporates a separate transceiver for sensor reading and motor control controlled by a local OOPIC processor doing very limited computation. The architecture is shown in Figure 22.
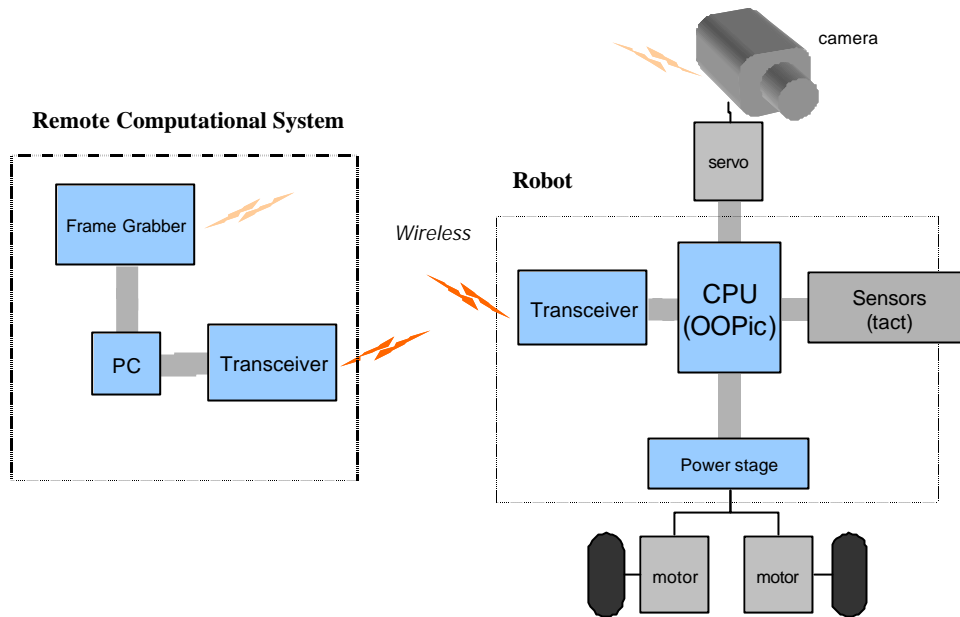
Figure 22. Distributed robot architecture consisting of a PC-based frame grabber and transceiver and a wireless mobile robot incorporating a wireless video camera and separate transceiver for sensor reading and motor control being managed by an OOPIC processor.

Two robot configurations were developed, initially a Lego-based configuration and later an OOPIC based robot were used, as shown in Figures 23.

LEGO   OOPIC

Figure 23. Two different robot configurations were used for these experiments, one Lego-based and later an OOPIC based.

In Figure 24, we show a sample display of how scenes are visualized directly from Internet, in this case consisting of an aerial camera and local robot camera. User interaction includes real-time graphic displays corresponding to the different neural schemas in the model.
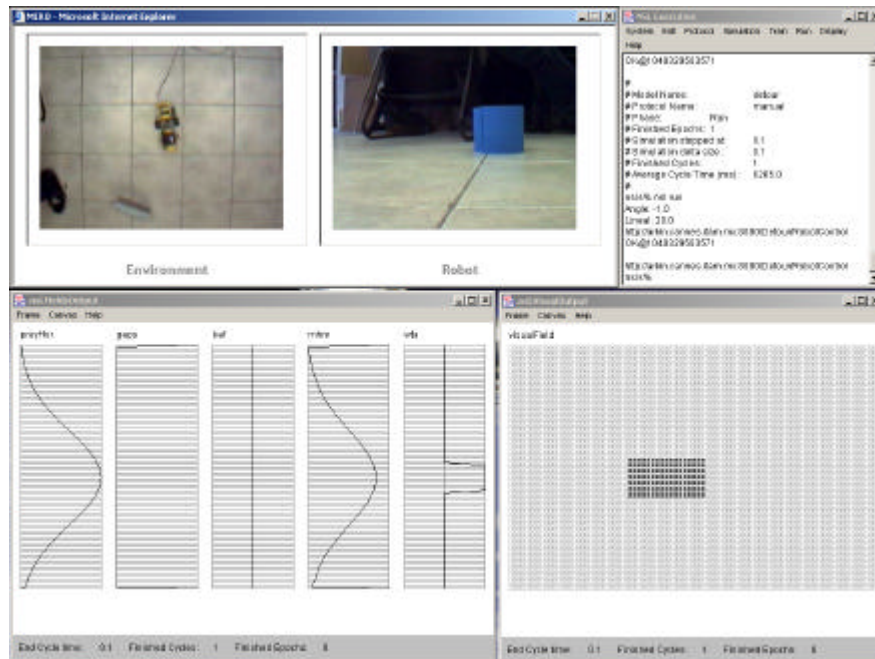
Figure 24. Internet aerial view of autonomous robot and robot's camera view of "blue" prey-like stimulus with NSL/ASL frames showing results from different visual and neural modules in a basic prey acquisition robot experiment.

In Figure 25, we show a sample cycle of computation where objects are recognized by color, where blue corresponds to a prey and red to a fencepost.

Figure 25. Cycle of computation for the different experiments where a blue object represents a prey and a red object represents a fencepost. The model output consists of a distance d to move, a robot orientation q_r and a camera orientation q_c.

Experiment I

In Figure 26, a sequence of pictures is shown for experiment I. Note how the camera rotates independently from the robot body in keeping a view of the prey at all times.

(A)   (B)

(C)   (D)
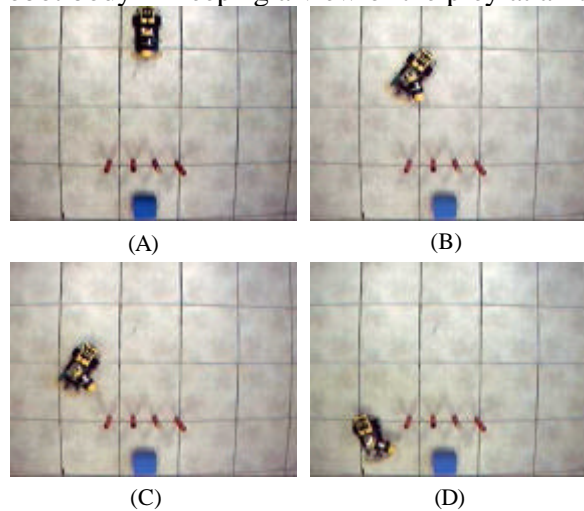
Figure 26. Results from prey acquisition experiment for 10cm barrier with direct detour around barrier.

Experiment II

In Figure 27, a sequence of pictures is shown for experiment II. Again, note how the camera rotates independently from the robot body in keeping a view of the prey at all times. In this experiment the robot bumps into the barrier a couple of times before being able to circle around the barrier.
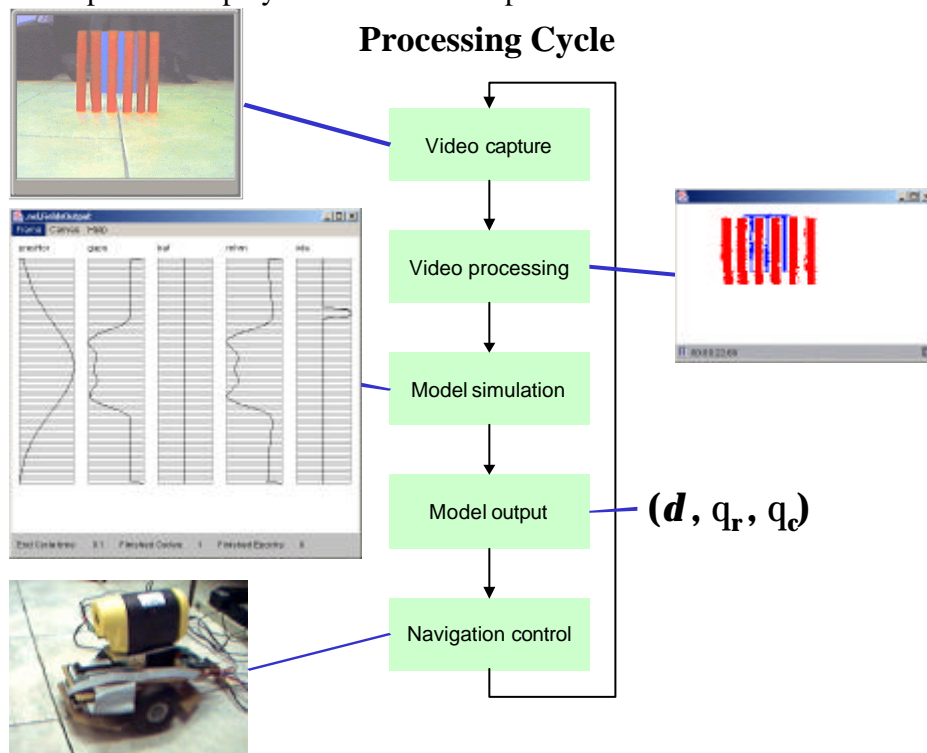

(A)   (B)   (C)   (D)

(E)   (F)   (G)   (H)
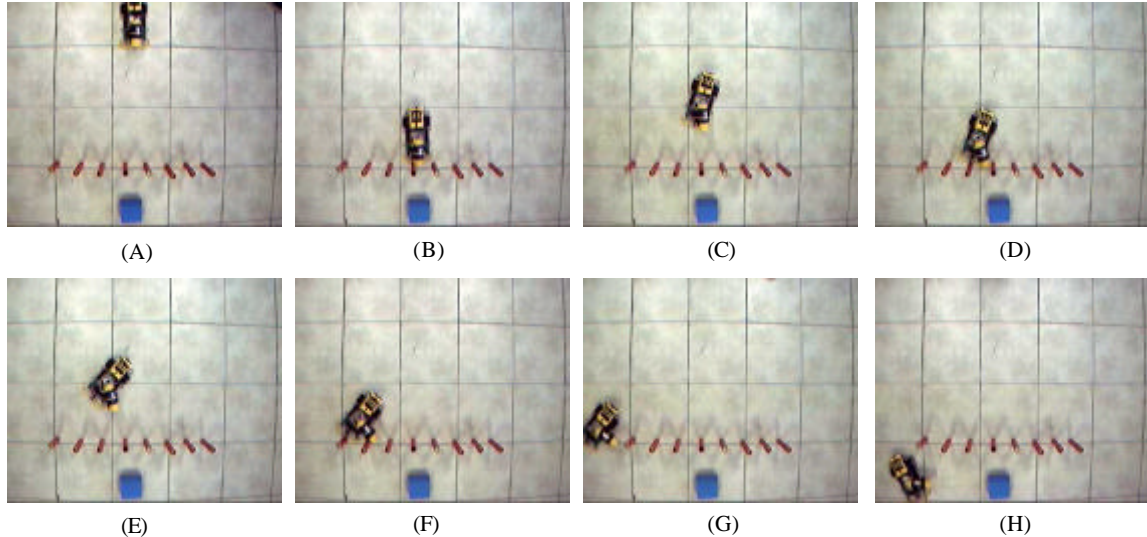
Figure 27. Results from prey acquisition experiment for 20cm barrier with direct detour around barrier.

4.   CONCLUSIONS AND DISCUSSION

The work presented in this paper overviews the challenges and complexity in modeling robotic systems inspired by neuroethological animal models. The motivation behind this work has been to provide neuroscientists a testbed for robotic experimentation and to provide roboticists with new tools and architectures in the development of biologically inspired systems.

- **Neuroethological modeling**. In terms of neuroethological modeling, complexity is managed by taking a multi-level approach emphasizing both top-town and bottom-up designs through different levels of granularity. At the top-level a robot is defined in terms of sensors and actuators in defining the robot interaction environment. Next level down, behaviors are described in terms of schema models such as the frog's prey acquisitions, predator and static object avoidance. Schemas may be refined in a hierarchical manner until reaching lower-level neural networks representing schema implementations. Neural implementation may include simpler or more detailed neuron models, such as those requiring synaptic plasticity. The main challenge with neural schemas is the need to link independently developed neural models, such as those shown in Figure 9, where input and output specifications do not necessarily match. For example, the original neural models for *Retina*, *Tectum* and *Pretectum* incorporated coarse visual input instead of more faithful *R2*, *R3*, and *R4* retina class cells. This was done in order to obtain quicker results and make them more independent from other neural models. At this time we are reexamining the different neural models in order to: (1) separate what relates to actual visual input from specialized module processing and (2) modify these models to accept R2, R3, and R4 output coming from actual retinal visual input. To complicate matters further, the logic of one module may be based on different assumptions different from those of other modules, e.g. different experiments, parameters or time frequencies. Yet, if we do not manage this integration, it will not be possible to "reuse" neural modules in more comprehensive neuroethological based robotic architectures [25].
- **Prey acquisition with detour**. The toad prey acquisition with static object avoidance model presented in this paper explains basic facts about detour behavior. In experiment I, if the retinotopic representation of the edge of the barrier falls within the prey-attractant-field, then the summation of activity from the prey-attractant-field and the barrier-repellent map at the retinotopic position just beyond the barrier's edge is stronger then the summation at the center of the barrier where the prey is located. Hence, the winner-take-all dynamics will select the cluster of activity corresponding to the retinotopic position at the edge of the barrier, resulting in a detour of the frog around narrow barriers. In experiment II, in the case of wide barriers, the prey-attractant-field extent falls within a much wider barrier field. Hence, at the retinotopic position corresponding to the barrier's edge there will be no input activity from the prey map. On the other hand, there will be a great projection of activity at the retinotopic position of the prey, thus triggering an approach to a point within the barrier map so long as the peak of prey attraction exceeds the barrier inhibition. Thus, the model predicts that the naive frog would approach wide barriers rather than detour around them. In experiment III, an artificial detouring field is added at the edge of the barrier to make the frog move around the barrier without the need to perceive the actual edge.
- **Simulation versus robot embodiment**. Historically, most brain model development has been accomplished through simulation, but simulation is not quite the same as real-world robotic experimentation. In particular, many shortcuts are taken in simulation. For example, simulated cameras and world objects are made quite ideal, where cameras have large visual fields with "perfect" images and object sizes. Once models are experimented under real world conditions objects become less "perfect" harder to recognize. As part of our model experimentation with real robots, an interesting problem appeared in our prey acquisition with detour experiment, the problem of "losing" the prey once the robot orients towards one of the edges of the barrier. In the simulated version the robot always perceived the prey as

well as barrier by adjusting the size of the visual field. While toads do take care of this problem the actual model had to be modified in dealing with such issues. This is an example where simulated models may do fine under simulated environments but do not address specific issues originating from actual embodied robot experimentation. A simple solution to this problem was to add a new motor to control the camera independently from robot movement. Another alternative we are starting to explore gets its inspiration from other neurobiological models, in particular the oculomotor system in monkeys [21]. An interesting function of the oculomotor system is the control of "memory" saccades where the eye's fovea redirects itself to a stimulus from information previously recorded, something of particular interest to the prey acquisition and predator avoidance models. Yet, it is not simply a matter of integrating across the two models. The prey acquisition and predator avoidance models are based on toad and frog studies, while the oculomotor system previously mentioned is based on monkey studies, varying quite a bit in terms of the involved neurobiological systems. To neurobiologist this is quite significant. On the other hand, to robotic designers this is not necessarily important.

- **Distributed Neural Processing.** Large neural network models produce and consume great amounts of data and take a very large number of processing cycles to obtain meaningful results. A typical computation cycle starts by obtaining sensory input (visual and tactile) and ends by producing motor output. In between, schemas process sensory data in order to produce motor behavior. Cycles continue indefinitely or until some specific task is completed, such as reaching the prey. For example, a "typical" retina model [36] may consist of more than 100,000 neurons and half a million interconnections requiring many hours of simulation to complete these cycles. The expensive nature of neural computation is further exacerbated by the fact that a comprehensive schema-neural model includes multiple neural networks. This becomes even worse in the case of higher-level animals involving extensive behaviors and other brain regions [4]. By taking advantage of the parallel and distributed nature of neural network computation [38], we have extended the original NSL/ASL simulation system into a distributed architecture [42].

- **Distributed Robot Architecture.** In recent years a number of research efforts have been carried out to embed mobile robotic systems into computer networks via wireless communication [22]. This approach makes it possible not only to control and monitor remote robots, such as with the Mars explorers, but also enhance its capabilities by linking the robot to remote computational resources, such as image processing or neural processing. These efforts have highlighted the benefits of embedded systems in robotics via Internet [26][33]. As part of our current work in the design of embedded robotic systems we have developed the MIRO (Mobile Internet Robotics) architecture shown in Figure 21. The architecture consists of multiple robots each one connected to its own particular instance of the neural computational system. In such a way, processing is distributed among the robotic hardware and the remote computational system. Although it is possible to share robot "intelligence" among multiple robots where applications could easily take advantage of information sharing (see [9] for a discussion on distributed versus centralized robotic systems), we are particularly interested in keeping a truly autonomous robot architecture where neuroethological experimentation can be conducted. Under the MIRO architecture: (i) time-consuming processes are carried out in the (neural) computational system, implemented using the NSL/ASL system while (ii) sensory input, motor output and other limited tasks are carried out in the robot hardware. In such a way, the computational system

provides the robot's "intelligence", while the robot does limited processing. The most important challenge in this type of architectures is how to achieve real-time performance. We are at this time assessing the efficiency of the MIRO embedded architecture. A number of interesting questions have arisen, for example, what happens when communication between the robot and computational system actually fails or becomes extremely slow or unreliable. When such situation arises, the robot can respond in a number of ways, do nothing until communication is restored, end its mission, or perform limited tasks that may put it back in action such as actively searching for a location where communication can be reestablished.

- **Multi-robot systems**. Additionally, most of our experiments until now have involved single robots. The reason for this has been mostly due to the underlying complexity of neuroethological models. Our next goal is to experiment with multiple robots, where each robot will represent a different animal, such as a prey, toad or predator.

5. REFERENCES

[1]   Arbib, M.A., Levels of Modelling of Mechanisms of Visually Guided Behavior, *Behavior Brain Science* 10:407-465, 1987.

[2]   Arbib, M.A., *The Metaphorical Brain 2*, Wiley, 1989.

[3]   Arbib, M.A., Schema Theory, in the *Encyclopedia of Artificial Intelligence*, 2nd Edition, Editor Stuart Shapiro, 2:1427-1443, Wiley, 1992.

[4]   Arbib, M.A., Erdi, P. and Szentagothai, J., *Neural Organization: Structure, Function and Dynamics*, MIT Press, 1998.

[5]   Arbib, M.A. and Lee, H. B., Anuran visuomotor coordination for detour behavior: from retina to motor schemas, Proceedings From Animals to Animats III, Editors Cliff, D., Husbands, P., Meyer, J.M. and Wilson, S.W., MIT Press, 1994.

[6]   Arkin, R.C., Behavioral based Robotics, MIT Press, 1998.

[7]   Arkin, R.C., Ali, K., Weitzenfeld, A., and Cervantes-Perez, F., Behavior Models of the Praying Mantis as a Basis for Robotic Behavior, *in Journal of Robotics and Autonomous Systems*, 32(1), pp 39-60, Elsevier, 2000.

[8]   Arkin, R.C., Cervantes-Perez, F., and Weitzenfeld, A., "Ecological Robotics: A Schema-Theoretic Approach", "Intelligent Robots: Sensing, Modelling and Planning", eds. R.C.Bolles, H.Bunke, and H.Noltemeier, pp 377-393, World Scientific, 1997.

[9]   Balch, T. and Arkin, R.C., Communication in Reactive Multiagent Robotic Systems, Autonomous Robots, 1, pp 1-25, 1994.

[10]   Beer, R. D., Intelligence as Adaptive Behavior: An Experiment in Computational Neuroethology, San Diego, Academic Press, 1990.

[11]   Bower, J.M., and Beeman, D., The Book of GENESIS, Exploring Realistic Neural Models with the GEneral NEural SImulation System, Telos, Springer-Verlag, 2nd Edition, 1998.

[12]   Cervantes-Perez, F., Lara, R., and Arbib, M.A., A neural model of interactions subserving prey-predator discrimination and size preference in anuran amphibia, Journal of Theoretical Biology, 113, 117-152, 1985.

[13]   Cervantes-Perez, F., Franco, A., Velazquez, S., Lara, N., A Schema Theoretic Approach to Study the 'Chantitlaxia' Behavior in the Praying Mantis, *Proceeding of the First Workshop on Neural Architectures and Distributed AI: From Schema Assemblages to Neural Networks*, USC, October 19-20, 1993.

[14]   Cervantes-Perez, F., Herrera, A., and García, M., Modulatory effects on prey-recognition in amphibia: a theoretical 'experimental study', in *Neuroscience: from neural networks to artificial intelligence*, Editors P. Rudomin, M.A. Arbib, F. Cervantes-Perez, and R. Romo, Springer Verlag Research Notes in Neural Computing vol 4, pp. 426-449, 1993.

[15]   Cliff, D., Neural Networks for Visual Tracking in an Artificial Fly, in *Towards a Practice of Autonomous Systems: Proc. of the First European Conference on Artificial Life (ECAL 91)*, Editors, F.J., Varela and P. Bourgine, MIT Press, pp 78-87, 1992.

[16]   Cobas, A., and Arbib, M.A., Prey-catching and Predator-avoidance in Frog and Toad: Defining the Schemas*, J. Theor. Biol* 157, 271-304, 1992.

[17]   Collett, T., Picking a route; Do toads follow rules or make plans? *(Advances in Vertebrate Neuroethology,* J.P. Ewert, R.R. Capranica and D.J. Ingle, Eds), pp.321 – 330, 1983.

[18]   Corbacho, F., and Arbib M. Learning to Detour*, Adaptive Behavior*, Volume 3, Number 4, pp 419-468, 1995.

[19]   Corbacho, F., and Weitzenfeld, Learning to Detour*, in *The Neural Simulation Language NSL, A System for Brain Modeling*, MIT Press, July 2002.

[20]   Didday, R.L., A model of visuomotor mechanisms in the frog optic tectum, *Math. Biosci.* 30:169-180, 1976.

[21]   Dominey, P., and Arbib, M.A., A cortico-subcortical model for generation of spatially accurate sequential saccades, Cerebral Cortex, 2, pp 153-175, 1992,

[22]   Estrin, D., Govidian, R., and Heidemann, J. (Eds.) Special issue on Embedding the Internet, Communication of the ACM, 43(5), May 2000

[23]   Ewert, J. P., Single unit response of the toad's (*Bufo americanus*) caudal thalamus to visual objects. *Z. vergl. Physiol.* 74, 81-102, 1971.

[24]   Ewert, J.P, *Neuroethology, an introduction to the neurophysiological fundamentals of behavior*, Springer-Verlag, 1980.

[25]   Fagg, A., King, I., Lewis, A., Liaw, J., Weitzenfeld, A., A Testbed for Sensorimotor Integration, Proceedings of IJCNN '92, Baltimore, MD, 1:86-91, 1992.

[26]   Goldberg, K., and Siegwert, R., (eds), Beyond Webcams: An Introduction to Online Robots, MIT Press, 2002.

[27]   Hines, M., and Carnevale, T., The NEURON Simulation Environment, *Neural Computation*, 9:1179-1209, 1997.

[28]   Hodgkin, A.L. and Huxley, A.F., A quantitative description of membrane current and its application to conduction and excitation in nerve, *Journal of Physiology*, 117, 500-544, 1952.

[29]   House, D., Depth Perception in Frogs and Toads: A study in Neural Computing, Lecture Notes in Biomathematics 80, Springer-Verlag, 1985.

[30]   Ingle, D., Brain mechanisms of visual localization by frogs and toads. (*Advances in Vertebrate Neuroethology*, J. -P. Ewert, R. R. Capranica and D. J. Ingle, Eds), 177 – 226, 1983.

[31]   Rall, W., Branching dendritic trees and motoneuron membrane resistivity, *Exp. Neurol.*, 2:503-532, 1959.

[32]   Scalia, F., and Fite., K.V., A retinotopic analysis of the central connections of the optic nerve in the frog, J. Comp. Neurol., 158:455-478, 1974.

[33]   Sukhatme, G.S., and Mataric, M.J., Embedding Robots Into the Internet, Communication of the ACM, 43(5) pp 67-73, Special issue on Embedding the Internet, D. Estrin, R. Govidian, and J. Heidemann, eds., May 2000.

[34]   Sun, R., 1995, On schemas, logics, and neural assemblies, Applied Intelligence, 5(2):83-102.

[35]   Sutton, R., and Barto, A., Reinforcement Learning: An Introduction, MIT Press, 1998.

[36]   Teeters, J.L., and Arbib, M.A., A model of the anuran retina relating interneurons to ganglion cell responses, *Biological Cybernetics*, 64, 197-207, 1991.

[37]   Weitzenfeld, A., ASL: Hierarchy, Composition, Heterogeneity, and Multi-Granularity in Concurrent Object-Oriented Programming, *Proceedings of the Workshop on Neural Architectures and Distributed AI: From Schema Assemblages to Neural Networks*, USC, October 19-20, 1993.

[38]   Weitzenfeld, A., Arbib, M., A Concurrent Object-Oriented Framework for the Simulation of Neural Networks, *Proceedings of ECOOP/OOPSLA '90 Workshop on Object-Based Concurrent Programming*, *OOPS Messenger*, 2(2):120-124, April 1991.

[39]   Weitzenfeld, A., Arbib, M., Alexander, A., *NSL - Neural Simulation Language: A System for Brain Modeling*, MIT Press, July 2002.

[40]   Weitzenfeld A., "A Multi-level Approach to Biologically Inspired Robotic Systems", *Proc of NNW 2000 10th International Conference on Artificial Neural Networks and Intelligent Systems* , Prague, Czech Republic, Juy 9-12, 2000.

[41]   Weitzenfeld, A., Cervantes, F., Sigala, R., NSL/ASL: Simulation of Neural based Visuomotor Systems, in *Proc. of IJCNN 2001 International Joint Conference on Neural Networks*, Washington DC, July 14-19, 2001.

[42]   Weitzenfeld A., Gutierrez-Nolasco, S., "ASL/NSL: A Multi-level Computational Model for Distributed Neural Simulation", in Proc of SCSC 2000 Summer Computer Simulation Conference, Vancouver, Canada, July 16-20, 2000.

[43]   Weitzenfeld A., Gutierrez-Nolasco S., and Venkatasubramanian N., MIRO: An Embedded Distributed Architecture for Biologically inspired Mobile Robots, Proc ICAR-03, 11th International Conference on Advanced Robotics, June 30 – July 3, Coimbra, Portugal, 2003.