Weitzenfeld A., Journal of Intelligent and Robotics Systems, Springer, Vol 51, Num 2, Feb 2008, ISSN 0921-0296

A Prey Catching and Predator Avoidance Neural-Schema Architecture for Single and Multiple Robots

Alfredo Weitzenfeld
Instituto Tecnológico Autónomo de México
Computer Engineering Department
Río Hondo #1, San Angel Tizapán, México DF 01000, MEXICO

Abstract – The paper presents a biologically-inspired multi-level *neural-schema* architecture for prey catching and predator avoidance in single and multiple autonomous robotic systems. The architecture is inspired on anuran (frogs and toads) *neuroethological* studies and wolf pack group behaviors. The single robot architecture exploits visuomotor coordination models developed to explain anuran behavior in the presence of preys and predators. The multiple robot architecture extends the individual prey catching and predator avoidance model to experiment with group behavior. The robotic modeling architecture distinguishes between higher-level schemas representing behavior and lower-level neural structures representing brain regions. We present results from single and multiple robot experiments developed using the NSL/ASL/MIRO system and Sony AIBO ERS-210 robots.

Index Terms – Biorobotics; Biologically-inspired Robotics; Neural Networks; Schemas; Behaviors; Prey Catching; Predator Avoidance; Swarms.

I. INTRODUCTION

In recent years researchers have taken a special interest in biological studies of animal behavior (*ethology*) as a basis for robotic systems [8][13]. Some researchers have gone even further by trying to provide a linkage between animal behavior and underlying brain mechanisms (*neuroethology*) [2]. Among these, there has been substantial work in studying the underlying neural mechanisms responsible for *sensorimotor* coordination in living animals [3]. In this paper we extend upon this work by describing a neural-schema architecture for robotics systems inspired on animal studies that builds upon a cycle of biological experimentation, computational modeling and robotics experimentation as depicted in Figure 1. This general research framework not only contributes to the development of new robotic architectures but also provides a platform to further study and experiment with animal behaviors [43].

The prey catching and predator avoidance sensorimotor models described in this work are inspired on visually-guided *neuroethological* studies in anurans (frogs and toads) [26] as well as praying mantis [16]. These models are further extended by *ethological* studies in wolf pack hunting [33] to inspire new studies, experiments and robotic architectures for single and multiple robot systems. The neuroethological models described by this work are based on *schema* and *neural network* multi-level architectures developed using the Abstract Simulation Language ASL [44] and the Neural Simulation Language NSL [47]. We integrate the ASL/NSL distributed system [50] with the Mobile Internet Robotics MIRO [49] to perform actual robot experiments using the Sony AIBO ERS-210 platform. This work is presented in the paper as follows: Section II describes a neural-schema architecture for biologically-inspired robotic systems; Secion III presents a prey catching and predator avoidance model for single robots; Section IV presents experiments and results for prey catching and predator avoidance on single robots; Secion V presents a prey catching and predator avoidance model for multiple robots; Section VI presents experiments and results for prey catching and predator avoidance on multiple robots; and Section VII presents the conclusions and discusses future work.

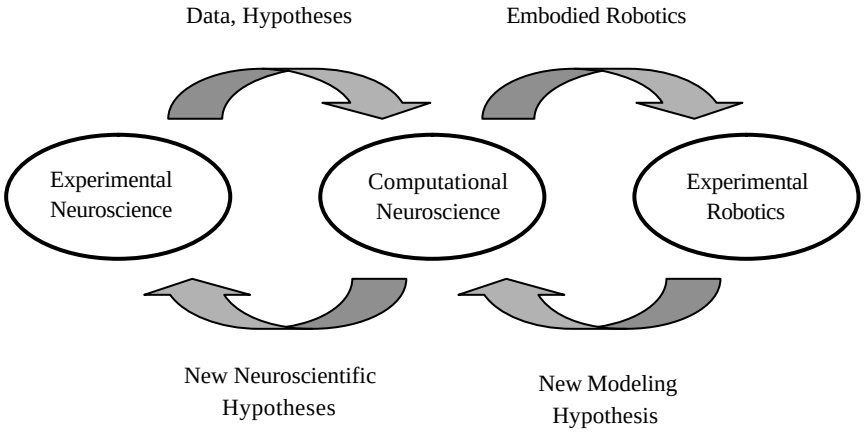

Figure 1. The above diagram depicts a framework for the study of living organisms through cycles of biological experimentation, computational modeling, and robotics experimentation serving as inspiration to the design of biologically-inspired autonomous robotics systems.

II. A NEURAL-SCHEMA ARCHITECTURE FOR BIOLOGICALLY-INSPIRED ROBOTICS SYSTEMS

The study of animals has provided an invaluable source of information for the understanding of neurobiological systems and the inspiration of new robotic architectures [7]. To address the underlying complexity in building such systems we distinguish between two different levels of modeling: behavior and neural structure [1]:

1. At the behavior level, ethological and neuroethological data from living animals is gathered to study the spatial-temporal relationship between single and multiple living entities and their environment, giving emphasis to aspects such as cooperation and competition between them. Examples of behavioral models include prey catching and predator avoidance in frogs and toads [6, 19, 20, 21], and praying mantis [9]. A *schema* computational model describes behaviors in terms of perception, sensorimotor integration and motor action [4, 5].
2. At the neural level, neuroanatomical and neuronphysiological data are used to generate perceptual and motor neural network models corresponding to schemas developed at the behavioral level [46]. These models are intended to explain the underlying mechanisms for sensorimotor integration involved in behaviors such as habituation [32, 42], prey catching and predator avoidance [48], and learning to detour around a barrier [23]. Examples of neural network models include the frog's tectum and pretectum-thalamus responsible for discrimination among preys and predators [17, 18].

A. Behavior

Animals depend on their ability to perform certain behaviors in response to external stimuli. In Figure 2 we show a prey catching behavior when a prey becomes visible (left) and a predator avoidance behavior when a predator appears (right).

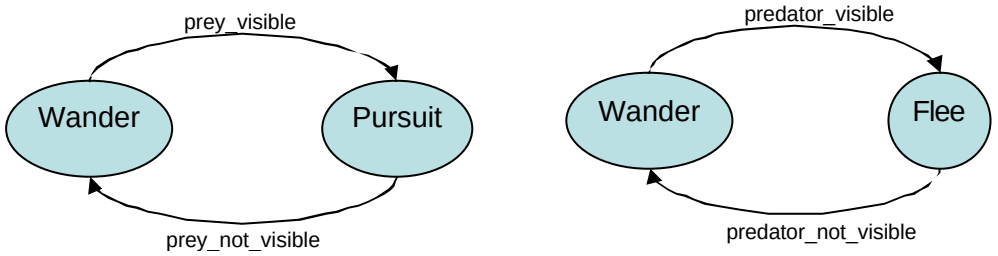

Figure 2. State machine diagram describing an animal pursuing a visible prey (left) and an animal fleeing away from a visible predator (right).

B. Schemas and ASL

Schemas define a distributed computational hierarchy for action-perception in animals. In schema architectures, as depicted in Figure 3, we distinguish among multiple levels of decomposition to better describe simpler units of processing in the brain. In the top portion of the diagram a higher-level schema is shown decomposed into two lower level schemas where the three schemas together form what is known as a schema *aggregate* or *assemblage*. Note how the multi-level schema decomposition provides a top-down approach where higher level schemas are initially described at a more functional way followed by a more detailed lower level schema specification, and a bottom-up data-driven

approach where individual schemas are specified in detail and then assembled in creating higher level comprehensive schema systems.

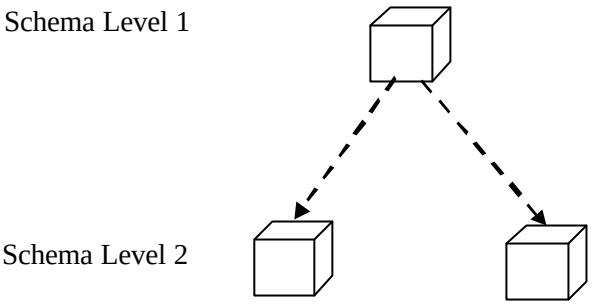

Figure 3. A schema hierarchy decomposes (dashed arrows) higher level schemas (level 1) into more detailed lower level schemas (level 2).

Communication between schemas can be of a cooperative or competitive nature. When schema activity surpasses certain threshold it produces output indicating enough confidence on that particular schema, also known as schema *assertion*. Figure 4 shows a schema architecture for a *prey catching* and *predator avoidance* behavior involving *Visual Perception* (PS - perceptual schema), *Motor Action* (MS - motor schema), and *Prey Catching* and *Predator Avoidance* (S – sensorimotor schemas). In the diagram *Prey Catching* is further decomposed into *Prey Recognition*, *Prey Selector* and *Prey Approach* schemas.

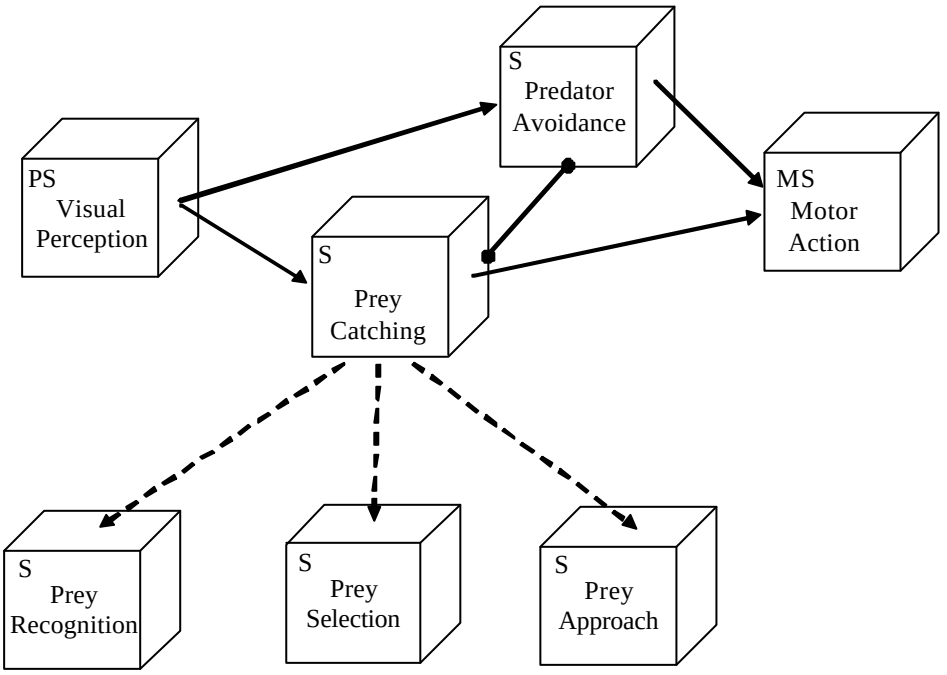

Figure 4. Schema architecture showing *Visual Perception* (PS - perceptual schemas), *Motor Action* (MS - Motor Schema), and *Prey Catching* and *Predator Avoidance* (S - sensorimotor schemas). *Prey Catching* is further decomposed into lower level *Prey Recognition*, *Prey Selection* and *Prey Approach* schemas. Solid arrows represent information flow from schemas at the same level; dashed arrows represent schema decomposition between multiple levels; and solid circles at the end of lines represent schema competition.

The Abstract Schema Language ASL [44] can be used to model schema architectures where each schema incorporates its own structure and control mechanisms. At the higher functional levels, schemas are only specified, leaving their detailed implementations to 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 5.

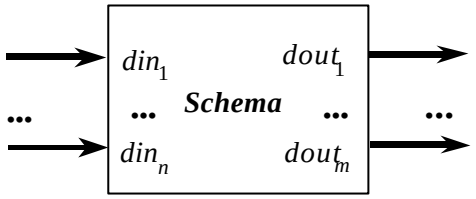

Figure 5. Each schema may contain multiple input, $din_1,...,din_n$, and output, $dout_1,...,dout_m$, ports for unidirectional communication.

Schema connectivity is inspired in *port automata* with activity variables indicating the degree of confidence [9]. Communication between schemas in ASL 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*. In Figure 6 we show an example of two level schema hierarchy with interconnected and relabeled ports. Schemas are interconnected by matching schema interfaces, in other words, *connections* (solid arrows) are done by linking *output ports* from one schema to *input ports* in other schemas. On the other hand, *relabelings* (dashed arrows) are done by linking ports of similar type (input or output) among different schemas usually at different levels in the hierarchy. In such a multi-level hierarchy, we consider that higher-level schemas to *delegate* the task to lower-level ones. The hierarchical port management framework enables the development of distributed architectures where schemas may be designed in a top-down or bottom-up fashion implemented independently and without prior knowledge of the complete model or their final execution environment, thus encouraging component reusability.

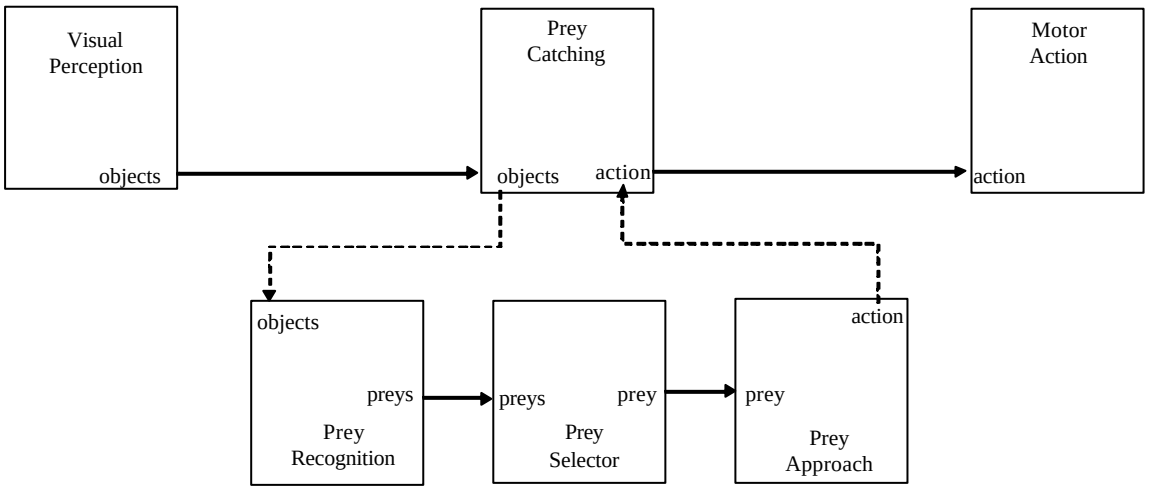

Figure 6. Example of a two-level schema model showing port interconnections and relabelings. Note how the *objects* input port in *Prey Catching* is relabeled to *objects* in *Prey Recognition* while *action* output port in *Prey Approach* is relabeled to *action* in *Prey Catching*. The solid lines show connections from output ports in one schema to input ports in another schema.

In the following piece of code we show portions of the schema specification in ASL corresponding to the diagram described in Figure 6. ASL uses a Java-like syntax where schemas are declared and instantiated (not seen in the code) and have their ports interconnected inside the *makeConn* using the *nslConnect* method connecting an output port to an input port, e.g. *vp.objects* with *pc.objects* and *pc.action* with *ma.action*, as shown below.

```
nslModule SchemaModel extends NslModule
{
        …
        public VisualPerception vp;
        public PreCatching pc;
        public MotorAction ma;
        …

        public voidmakeConn(){
                ….
                nslConnect(vp.objects, pc.objects);
                nslConnect(pc.action, ma.action);
                ….
        }
}
```

In the following piece of code we show a portion of *PreyCatching* schema where it *relabels* some of its ports to lower-level ports defined in *PreyRecognition* and *PreyApproach* schemas. Note the additional connection between the *preys* and *prey* ports in the lower level schemas.

```
nslModule PreyCatching extends NslModule
{
        public PreyRecognition pr;
        public PreySelection ps;
        public PreyApproach pa;
        …

        public voidmakeConn(){
                nslRelabel(objects, pr.objects);
                nslRelabel(pa.action, action);

                nslConnect(pr.preys, ps.preys);
                nslConnect(ps.prey, pa.prey);
        }
}
```

C. Neural Networks and NSL

In a neural-schema architecture, schemas may be specified at a higher level and then implemented by lower level neuroethological mechanisms corresponding to *neural schemas*, as shown in Figure 7.

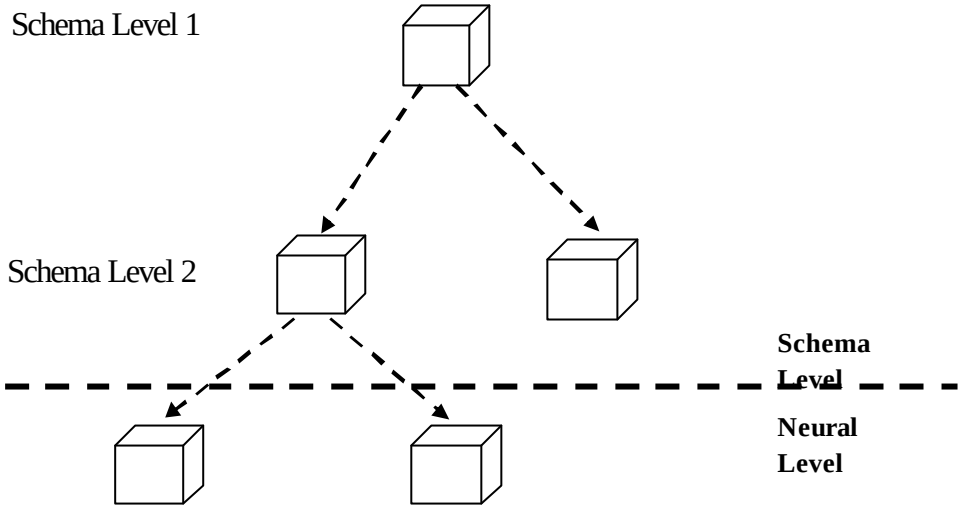

Figure 7. In a neural schema architecture, higher level schema may be implemented by lower level neural mechanisms in the form of *neural schemas*.

A *MaxSelector* neural schema example is shown in Figure 8. This schema is responsible for selecting among multiple preys where more than one is present. At the ASL description level, neural schemas are treated as regular schemas having ports, connections and relabels as any other schema.

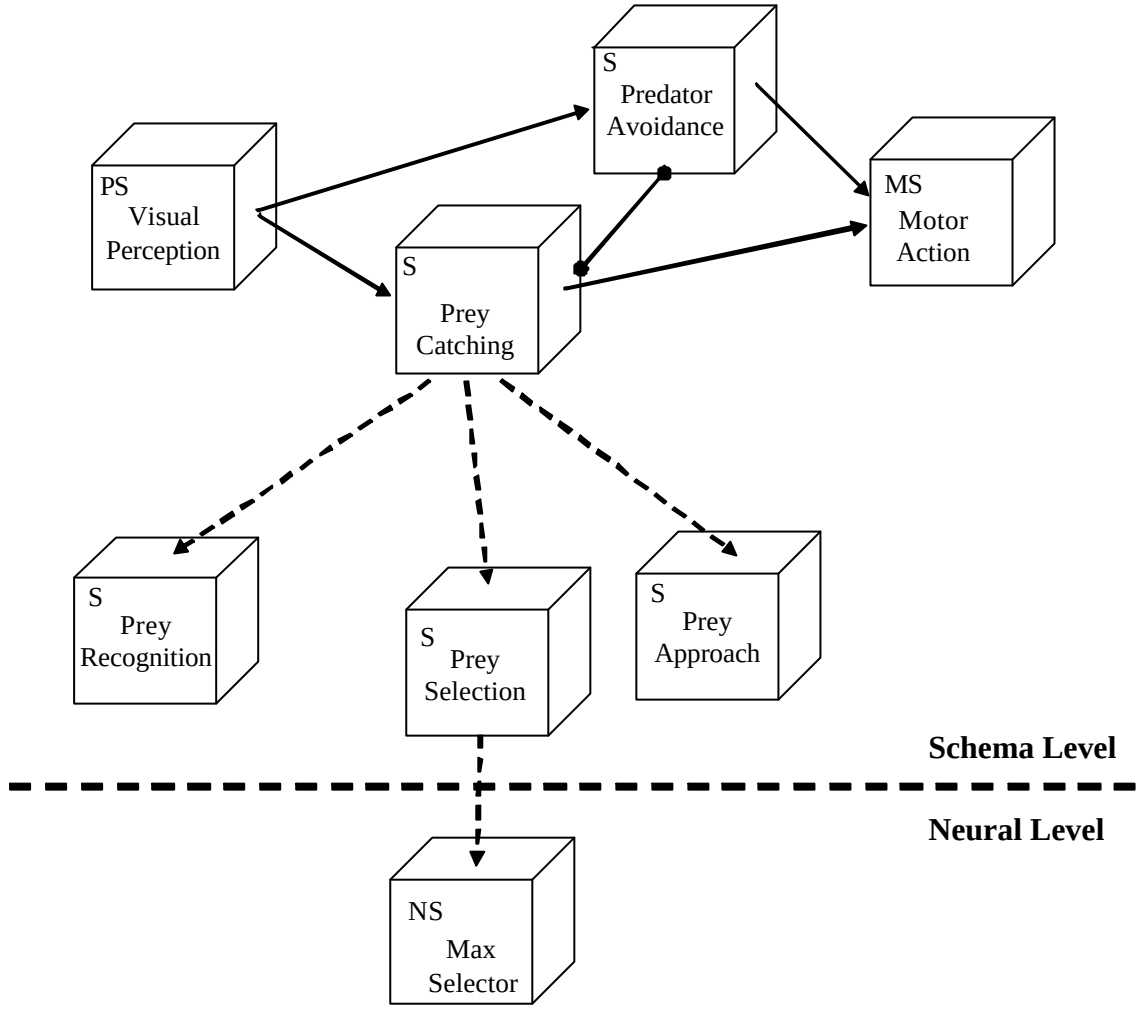

Figure 8. A *MaxSelector* neural schema implementation of *Prey Selection* schema.

A neural network implementation of a neural schema is shown in Figure 9 where neurons are described by simpler elements, each represented by a sphere with connections among neurons represented by solid arrows. In general, *neural schemas* may be implemented by neural networks described at any levels of detail, from very simple neurons to complex and detailed neurons involving electrochemical mechanisms responsible for phenomena like synaptic plasticity. While there is no restriction to what degree of detail a neural schema may be described, it is usually the case where simpler neural models are used when building larger neural networks.

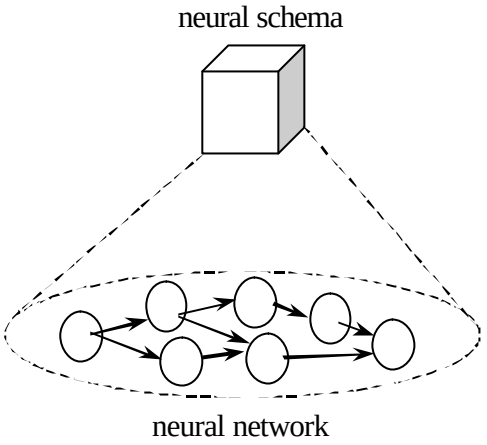

Figure 9. Neural schema implementation by a simple neural network model. Spheres represent neurons while solid arrows represent neural connections.

Simpler neuron models, such as the *leaky integrator* neural model [2], are best suited for large-scale computation, where each neuron is defined in terms of a *membrane potential* with value *mp* representing its previous history, input *s* and output value *mf* represented by a non-linear threshold function over its membrane potential, as shown in Figure 10.

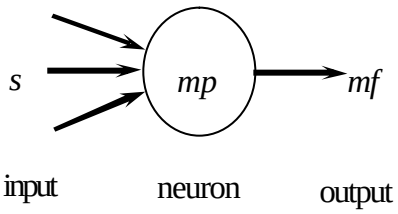

Figure 10. Simple neural model having an input *s*, a neuron body or soma *mp*, and output *mf*.

The *leaky integrator* neural model is described by:

$$d\mathbf{mp}(t)/dt = f(\mathbf{s},\mathbf{mp},t) \quad (1)$$

$$\mathbf{mf}(t) = s(\mathbf{mp}(t)) \quad (2)$$

$$t\ d\mathbf{m}(t)/dt = -m(t) + s \quad (3)$$

Equation 1 describes the membrane potential *mp* in terms of a first order differential equation function *f* that varies in time and depends on its input *s* and previous *mp* value. Equation 2 describes the cell firing *mf* in terms of a threshold function *s* that also varies in time and depends on its membrane potential. Equation 3 describes the leaky integrator model with dependence on an integration constant *t*.

For example, the Maximum Selector [25] neural network, shown in Figure 11, is described using the following leaky integrator equations:

$$t_u \frac{dup_i}{dt} = -up_i + w_u uf_i - w_m vf - h_u + s_i \quad (4)$$

$$uf_i = \begin{cases} 1 & up_i > 0 \\ 0 & up_i \le 0 \end{cases} \tag{5}$$

$$t_v \frac{dvp}{dt} = -vp + w_n \sum_{i=i}^{n} uf_i - h_v \tag{6}$$

$$vf = \begin{cases} vp & vp > 0 \\ 0 & vp \le 0 \end{cases} \tag{7}$$

Equation 4 represents the leaky integrator equation for neurons *u* with weighted input from *u* and *v*, visual input *s*, in addition to parameter *h* and time constant t. Equation 5 describes the output of neurons *u* as a "step" threshold function. Equation 6 represents the leaky integrator equation for neuron *v* with weighted input from *v* and *u*, in addition to parameter *h* and time constant t. Equation 7 describes the output of neuron *v* as a "ramp" threshold function.

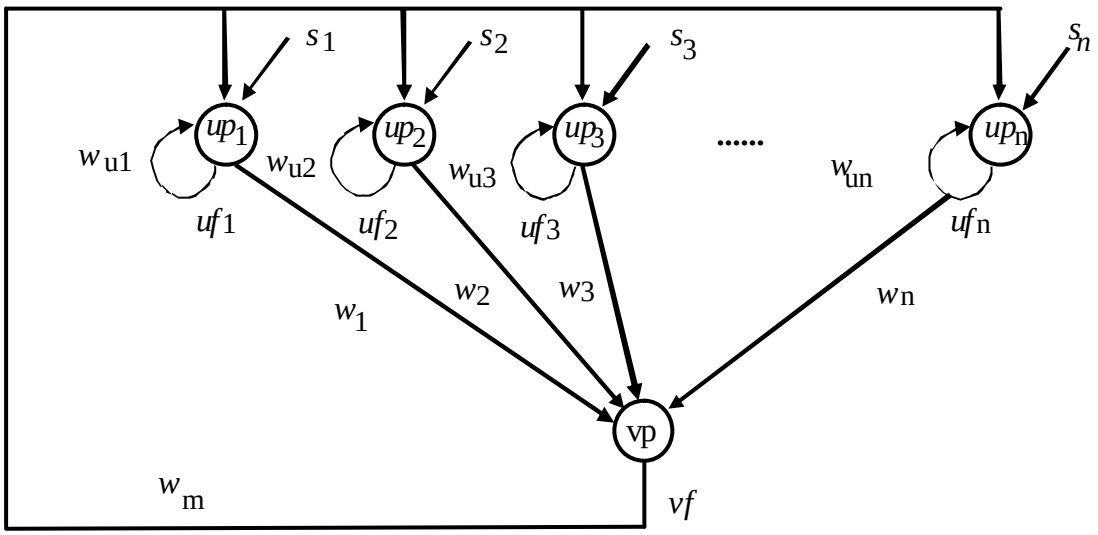

Figure 11. The neural network shown corresponds to the architecture of the Maximum Selector model, where up_i and *vp* represent neural membrane potentials, uf_i and *vf* represent neural firing rates, s_i represent inputs to the network, and w_i represent connection weights. The network is initialized 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.

The Neural Simulation Language NSL [47] provides the foundations to describe neural schemas in terms of neural networks. For example, the *MaxSelector* neural schema code is described next and includes instantiation variables, *sizeX* and *sizeY* where the actual number of neurons is specified. The schema also contains an *in* "NslDinDouble2" input port and an *out* "NslDoutDouble2" output ports. Additionally, there are declarations for internal variables, *up*, *uf*, *vp*, *vf*, *hu*, *hv* and *tau*, used in the *simRun* method for neural processing. Note the correspondence with equations 4 to 7. The other

method, *initRun*, is used for constant and variable initializations. (Documentation and model code is available for download from the NSL web sites [34].)

```
nslModule MaxSelector (int sizeX, int sizeY) extends NslModule
{
        public NslDinDouble2 in(sizeX,sizeY);
        public NslDoutDouble2 out(sizeX,sizeY);

        public NslDouble2 up(sizeX,sizeY);
        public NslDouble2 uf(sizeX,sizeY);
        public NslDouble0 vp, vf;
        public NslDouble0 hu, hv, tau;

        public void initRun(){
                up = 0; uf = 0; vp = 0; vf = 0;
                hu = 0.1; hv = 0.5; tau = 1.0;
        }
        public void simRun(){
                up = nslDiff(up,tau, -up + uf  - vf – hu + in);
                uf = nslStep(up,0.1,0.1.0);
                vp = nslDiff(vp,tau,-vp + nslSum(uf) – hv);
                vf = nslRamp(vp);
                out = uf;
        }
}
```

D. Robotics and MIRO

The ASL and NSL systems have been *embedded* into the MIRO [49] robotics architecture into a unified simulation and robotics experimental environment. In the integrated system, shown in Figure 12, MIRO controls robot sensory input and motor output while ASL/NSL performs expensive schema and neural network model processing. MIRO performs preliminary visual processing such as blob formation and segmentation from either the simulated or real camera, and performs motor actions from simulation or real robot commands. Although the MIRO architecture would make it possible to share robot "intelligence" among multiple robots, we keep robots fully independent and autonomous. The drawbacks in such an architecture are communication delays between robot and computer considered negligible considering the expensive processing of neural networks.

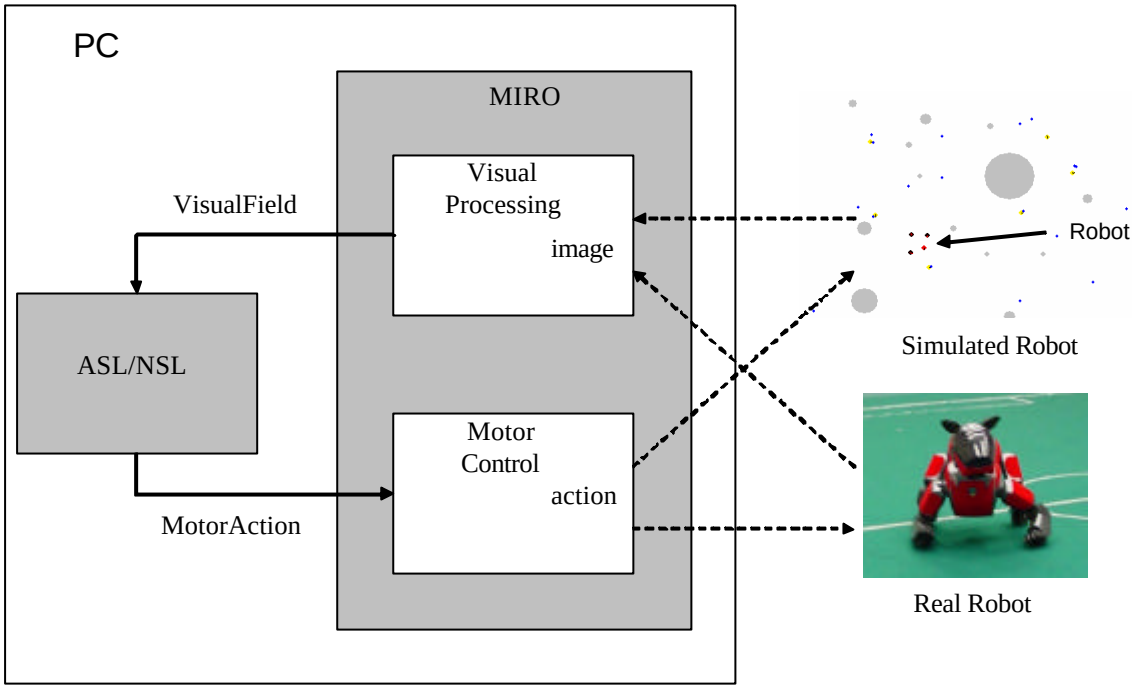

Figure 12. Block diagram for the *embedded* ASL/NSL and MIRO robot architecture. ASL/NSL is responsible for schema and neural network processing while MIRO provides robotic control from the simulated or physical robot environment.

For the work presented in this paper we have used a number of Sony AIBO ERS-210 four-legged robots having a local camera communicating with the ASL/NSL/MIRO system via wireless networking. In Figure 13 we show a sample cycle of computation of the NSL/ASL/MIRO robotics architecture. The camera in the robot captures video and sends it to the remote computer for video processing by MIRO. Afterwards model processing is carried out in the computer by NSL/ASL using as input the processed images. NSL/ASL generates model output in the form of action control commands, i.e. robot walking, robot and camera headings. These commands are sent by MIRO to the robots via wireless communication. The following cycles repeats themselves indefinitely or until behaviors are completed:

- *Video Capture* - Images are obtained from the simulated or real robot camera via wireless transmission.
- *Video Processing* - Objects are recognized according to some intrinsic characteristic such as color, e.g. blue corresponding to a prey and green to a predator.
- *Model Processing* - Graphs represent prey catching and predator avoidance schema fields.
- *Model Output* - Model output is specified in terms of displacement d, robot orientation q_r and camera orientation q_c.
- *Robot Control*. Robot is controlled remotely via wireless transmission.

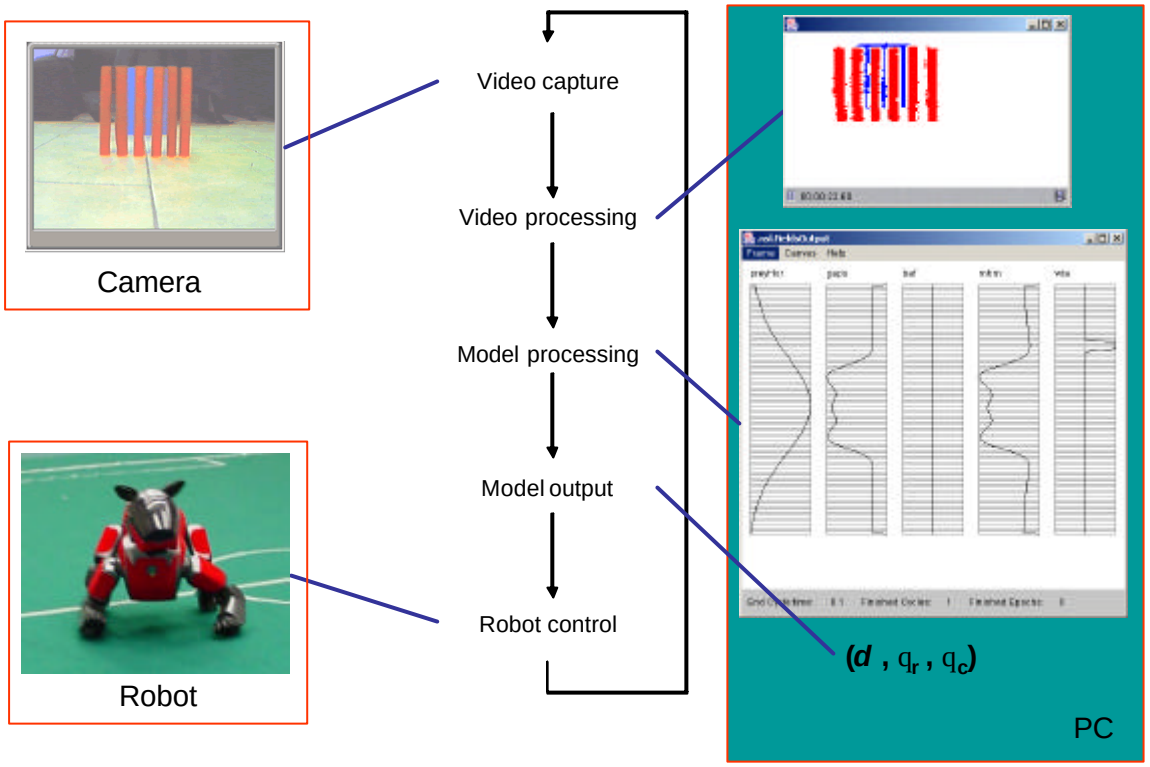

Figure 13. Cycle of computation for robotic experiments using the ASL/NSL and MIRO architecture. The model output consists of robot displacement *d*, robot orientation *qr* and camera orientation *qc*.

III. Prey Catching and Predator Avoidance – Single Robot Architecture

Anurans (frogs and toads) and praying mantis display similar visuomotor coordination prey catching and predator avoidance. Both animals respond to far away moving stimuli in their monocular lateral visual field, exploiting their binocular visual field for closer stimuli distance computation. While their overall behaviors may overlap, such as orienting and then approaching a prey, there are several differences such as their response to predators. Initially both animals may try to flee away, yet, frogs will display a *ducking* behavior when closer to predators while praying mantis will display a *deimatic* behavior where the insect stands up and opens its wings and forearms to appear larger than it really is [16]. In this section we describe the generalities in anuran and praying mantis prey catching and predator avoidance behaviors where particular anuran behaviors are included in the model when differing in their response. Prey catching behavior in anurans and praying mantis (to be generalized as "frogs" in the rest of the paper) are summarized as follows:

- *Pursuit* - The frog will move towards a prey using either its lateral or binocular visual field depending on its distance to the stimulus. The frog will move in the same direction as the prey, trying to keep it within reaching distance.
- *Attack* - When the stimulus is within reaching distance, the frog will snap at the prey after performing a binocular fixation to estimate its three-dimensional relative position.
- *Eat* – After successfully snapping at the prey the frog will swallow and ingest it.

In the case of predator avoidance, frogs will detect large snake-like or bird-like moving stimuli by reorientation and fleeing away from them.

- *Flee* - The frog will orient away in an opposite direction to the predator.
- *Duck* – When too close to escape, the frog will duck in an attempt to avoid being seen by the predator.

In the case of multiple stimuli, a frog will select and react to the largest prey or predator although in some cases it will respond to their average [22]. Figure 14 shows a stimulus-response diagram for the frog prey catching (left) and predator avoidance (right) behaviors.

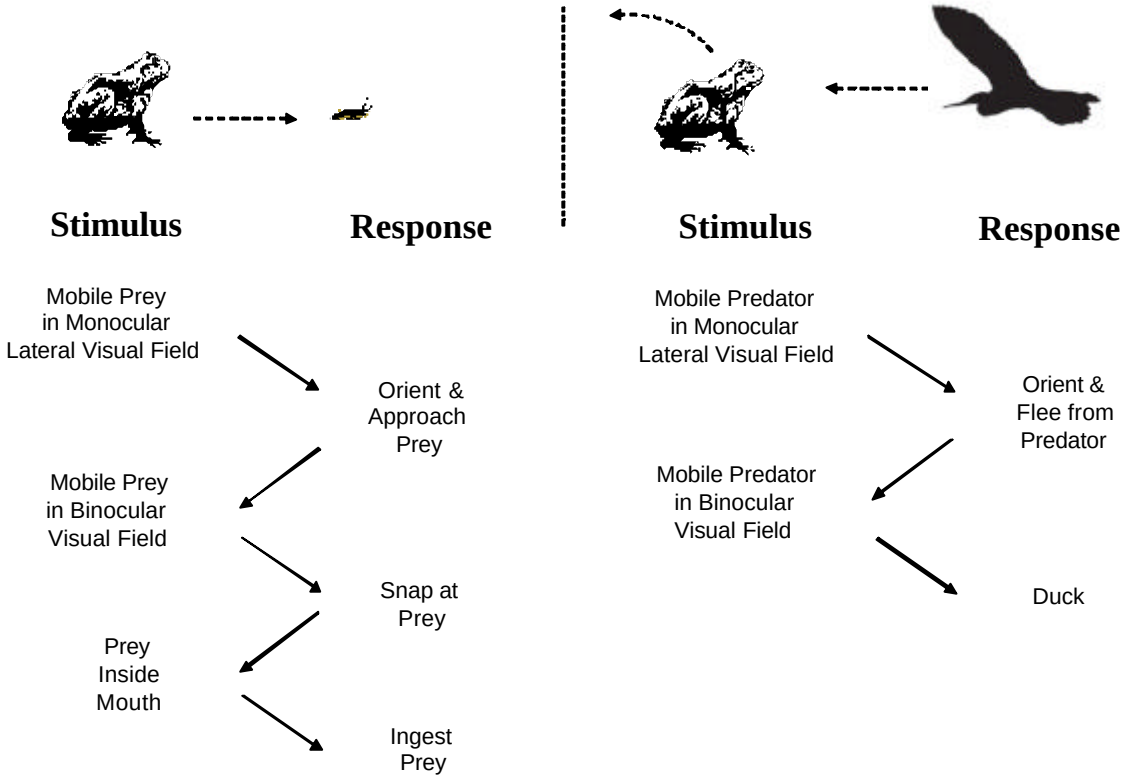

Figure 14. Stimulus-response diagrams for a frog (left) prey catching and (predator) predator avoidance behaviors.

A. Behaviors

The prey catching behavior is described in terms of a state diagram that includes *Pursuit*, *Attack*, *Eat* and *Wander* states as shown in Figure 15:

- *Wander* - In the *Wander* state the frog explores the environment in search for a prey. When it detects one, the *prey_visible* condition is activated indicating a change to the *Pursuit* state.
- *Pursuit* - In the *Pursuit* state there are two possible transitions: one towards the *Wander* state in case the prey is outside its range of vision caused by the *prey_not_visible* condition; the second

transition occurs when the prey is detected at a close distance, activated by the *prey_near* condition.

- *Attack* - In the *Attack* state the animal snaps at the prey activating the *prey_catch* condition. If the animal gets far from the prey the behavior goes back to the *Pursuit* state activated by the *prey_far* condition.
- *Eat* - In the *Eat* state the animal digests the prey after successfully snapping at it. If the prey has been eaten the condition *prey_not_visible* is activated returning to the *Wander* state. If the prey suddenly escapes, the condition *prey_not_catch* is activated returning to the *Attack* state.

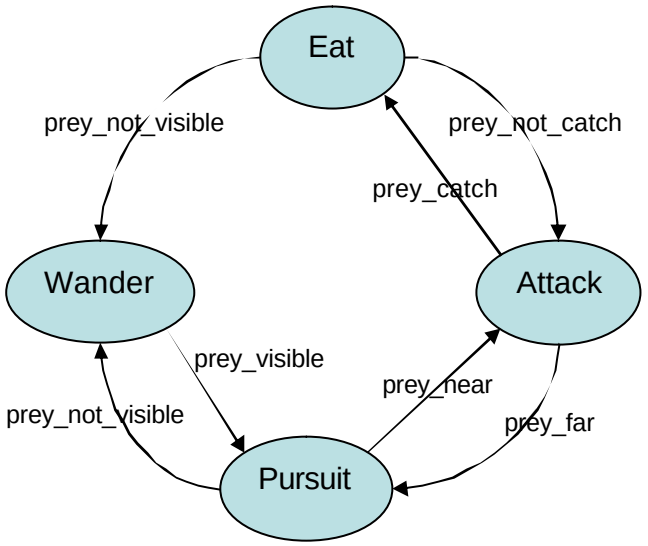

Figure 15. Prey catching behavior consisting of *Wander*, *Pursuit*, *Attack* and *Eat* states.

The predator avoidance behavior is described in terms of a state diagram that includes *Flee*, *Duck* and *Wander* states as shown in Figure 16:

- *Wander* – The *Wander* state is analogous to that in the Prey Catching behavior. If the animal detects a predator during wandering, the *predator_visible* condition is activated indicating a change to the *Flee* state.
- *Flee* – In the *Flee* state the animal will get away in direction opposite to the predator. If the predator gets too close the *predator_near* condition is activated changing to a *Duck* state avoid being perceived by the predator. If the predator is not visible, the *predator_not_visible* condition is activated going back to a *Wander* state.
- *Duck* – In the *Duck* state the frog gets either eaten or is able to escape. If the predator is visible but not too near, the *predator_far* condition is activated going back to the *Flee* state. If the predator is not visible, the *predator_not_visible* condition is activated changing to a *Wander* state.

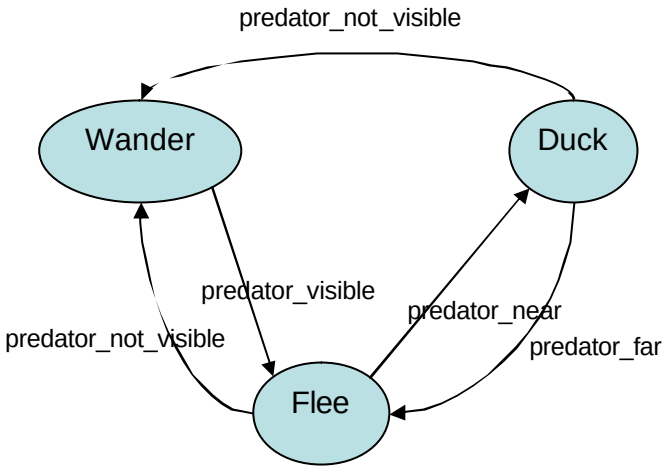

Figure 16. Predator Avoidance behavior consisting of *Wander*, *Flee* and *Duck* states.

Additionally, the animal will change from *Prey Catching* to *Predator Avoidance* state depending if a predator becomes visible with the activation of the corresponding *predator_visible* condition and alternatively the state will switch from *Predator Avoidance* to *Prey Catching* with the activation of the corresponding *predator_not_visible* condition as long as a prey is in the visual field (otherwise the frog will switch to a *Wander* state), as shown in Figure 17:

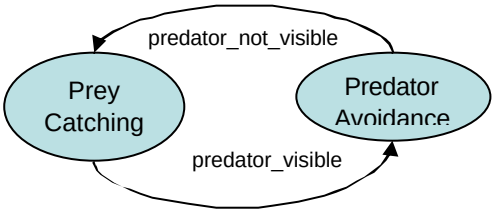

Figure 17. When in the *Prey Catching* state the animal will change to the *Predator Avoidance* state if the condition *predator_visible* is activated. Alternatively, when in the *Predator Avoidance* state the animal will change to the *Prey Catching* state if the condition *predator_not_visible* is activated.

B. Schemas

The prey catching and predator avoidance behaviors are mapped to a set of schemas, shown in Figure 18. The schema architecture includes perceptual schemas: *Visual*, *Prey Recognizer*, *Predator Recognizer*, *Max Prey Selector* and *Max Predator Selector*; sensorimotor schemas: *Prey Approach*, *Predator Avoid* and *Motor Heading Map*; and motor schemas: *Orient*, *Forward* and *Backup*. Note that *Orient* can be combined with *Forward* or *Backward* movement to elicit e.g. a diagonal motion. Note the competitive "dotted" line between *Prey Approach* and *Predator Avoid* schemas.

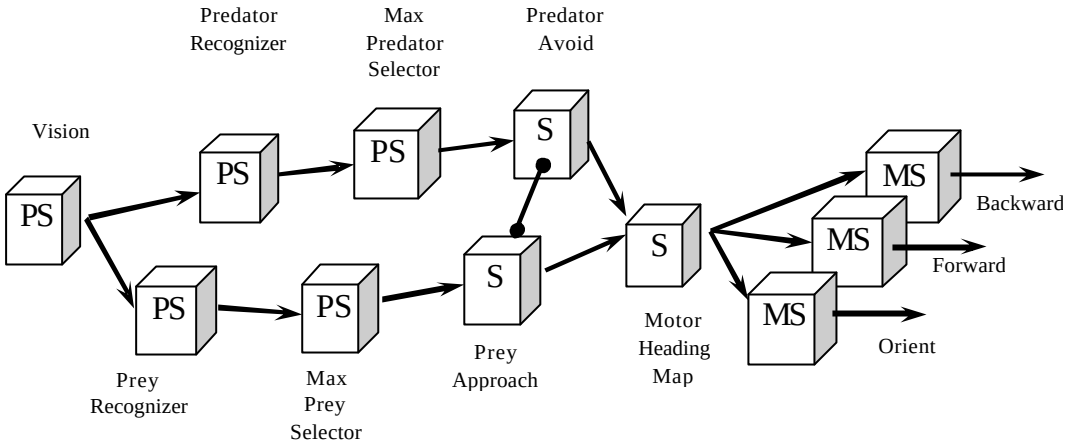

Figure 18. Prey acquisition and predator avoidance schema architecture consisting of perceptual schemas (PS): *Visual*, *Prey Recognizer*, *Predator Recognizer*, *Max Prey Selector* and *Max Predator Selector*; sensorimotor schemas (S): *Prey Approach*, *Predator Avoid* and *Motor Heading Map*; and motor schemas (MS): *Orient*, *Forward* and *Backup*.

Perceptual schemas describe sensor and recognition processes:

- *Vision* - Perceives moving stimuli mapped to either preys or predators.
- *Prey Recognizer* - Recognizes prey-like stimuli.
- *Predator Recognizer* - Recognizes predator-like stimuli.
- *Max Prey Selector* - Selects a single prey-like stimulus from multiple ones.
- *Max Predator Selector* - Selects a single predator-like stimulus from multiple ones.

Sensorimotor schemas integrate sensory and motor action processes:

- *Prey Approach* – Obtains input from the *Max Prey Selector* schema generating as output an attractant field whose strength decays proportional to the distance to the prey.
- *Predator Avoid* - Obtains input from the *Max Predator Selector* schema generating as output a repellent field whose strength decays proportional to the distance to the predator.
- *Motor Heading Map* (*MHM*) – Obtains input from both *Prey Approach* and *Predator Avoid* schemas generating output that combines their weighed activities. A winner-take-all dynamics over MHM assures the selection of the strongest target angle for prey attraction or predator repulsion. The MHM schema provides input to the different motor schemas.

Motor schemas describe motor actions representing intrinsic motor patterns or muscle activations:

- *Orient* - Obtains a direction to reorient, either forwards or backwards.
- *Backward* - Performs backward movements.
- *Forward* - Performs forward movements.

C. Neural Networks

In order to develop a multi-level neural schema model it is necessary to have a mapping between behavior, schemas and the underlying frog brain regions. Figure 19 illustrates the most important prey catching and predator avoidance brain regions involved, mainly: *Retina*, *Optic Tectum* and *Pretectum/Thalamus* [37].

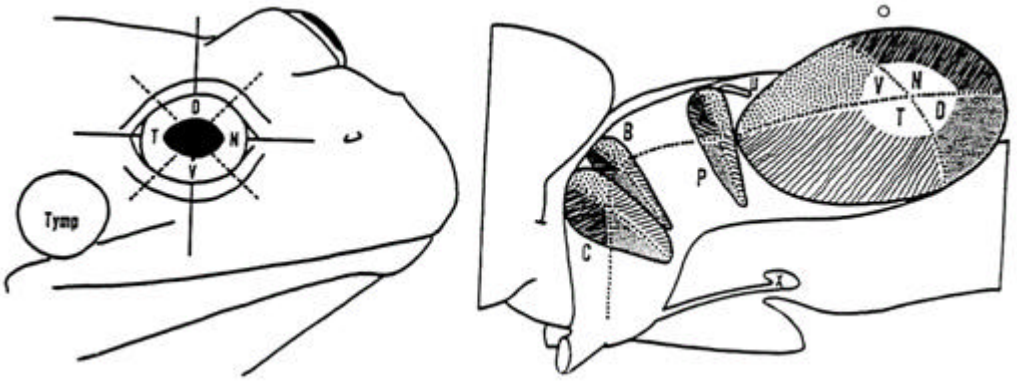

Figure 19. The two illustrations show the most important neural areas involved in the frog prey catching and predator avoidance model: Optic Tectum (O), divided into four regions: Temporal (T), Dorsal (D), Nasal (N) and Ventral (V); Thalamic Pretectal Neuropil (P); and Nucleus of Belonci (B), Lateral Geniculate Nucleus (C) and Basal Optic Root (X) [37].

Based on neurophysiological and neuroanatomical studies researchers have identified the structures of the Central Nervous System (CNS) that possibly conform the base of neural activity during prey catching and predator avoidance [27, 28]: *retina* (R), *optic tectum* (O), and *prethalamic tectum* (P). Many electrophysiological tests have shown how the optic tectum and its cells (*T5_2*) are linked to the prey recognition process and, therefore, to the prey catching behavior. On the other hand, the prethalamic tectum region and its cells (*TH3*) are related to predator recognition and, therefore, to predator avoidance. The discrimination between preys and predators, and the selection of the most important stimuli (if both are present), depend on the interaction between the T5_2 and TH3 cells. If there is a prey stimulus, the optic tectum, in addition to some prethalamic tectum signals, activates the catching motor responses. Meanwhile, when a predator stimulus is present, the prethalamic tectum sends inhibitory signals to the optic tectum to initiate the avoidance behavior. Physiological evidence shows that separate classes of retinal ganglion cells are sensitive to prey and predator stimuli supporting the hypothesis that tectal and thalamic visual mechanisms can operate somewhat independently [31]. In Figure 20 we show mappings between schemas and the underlying neural networks:

- *Retina* - Processes visual stimuli in anurans [24, 40].
- *MaxSelector* - Chooses among multiple stimuli, sometimes responding to an "average" stimulus [22, 25].
- *Tectum* – Recognizes preys as well as mates in anurans [14, 26].
- *PreTectum/Thalamus* - Recognizes stationary or predator-like objects in anurans [29, 30].

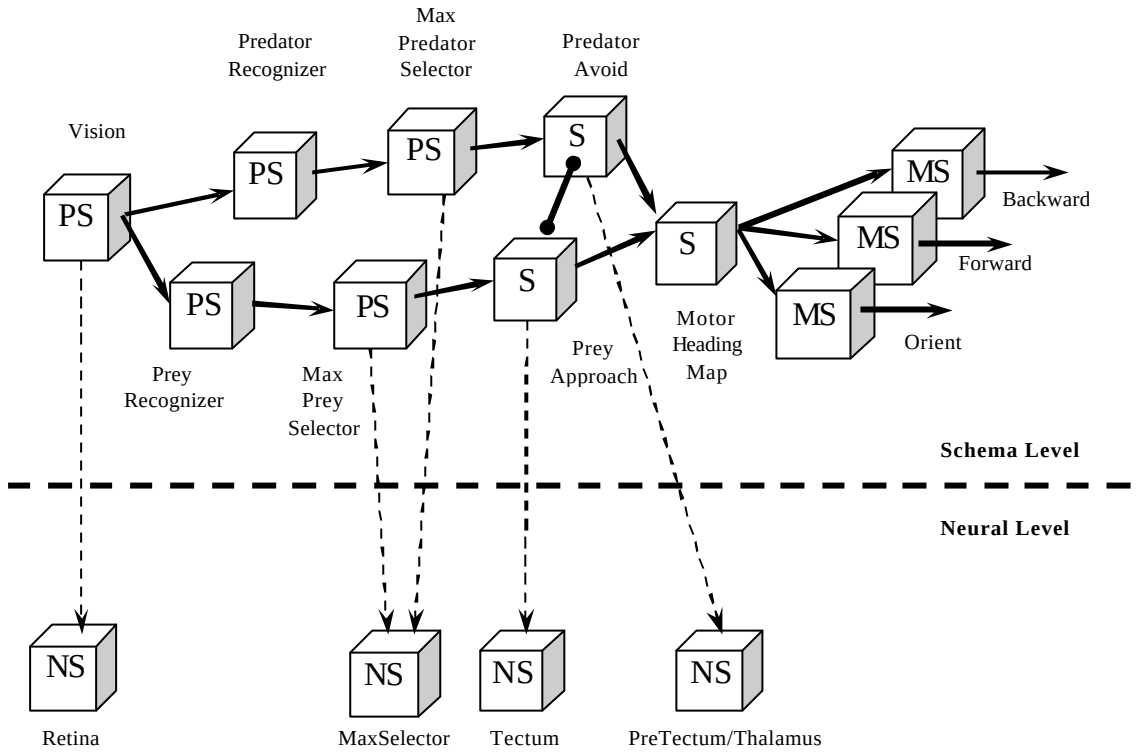

Figure 20. Prey catching and predator avoidance multi level neural schema architecture. The schema level consists of perceptual schemas (PS): *Visual*, *Prey Recognizer*, *Predator Recognizer*, *Max Prey Selector* and *Max Predator Selector*; sensorimotor Schemas (S): *Prey Approach*, *Predator Avoid*, and *Motor Heading Map*; and motor schemas (MS): *Orient*, *Forward*, and *Backup*. The neural level consists of four neural schemas: *Retina*, *MaxSelector*, *Tectum* and *PreTectum/Thalamus*.

IV. Prey Catching and Predator Avoidance – Single Robot Experiments and Results

In this section we describe prey catching and predator avoidance experiments on single robots inspired in anuran studies as described in the previous section. We initially test the experiments under a simulation-only environment and then perform physical experiments with the real robot.

A. Prey Catching

We tested a set of basic prey catching experiments involving a frog and prey as shown in Figure 21. The simulation shows, Figure 21 (left), a frog (square box) pursuing a moving prey (smaller horizontal rectangle). The frog moves towards the prey with steps numbered 1 through 4 where dashed arrows represent frog's visual field. Figure 21 (right) shows different neural schema field activities during the simulation: *predator_hor* represents predator activity in *PreTectum/Thalamus*, *prey_hor* represents prey activity in *Tectum*, *mhm* represents combined prey and predator activities in *Motor Heading Map* (*MHM*) and *wta* (winner-take-all) represents frog movement direction in *MHM*. Note that activities are modeled as Gaussians with the exception of the *wta* step function.

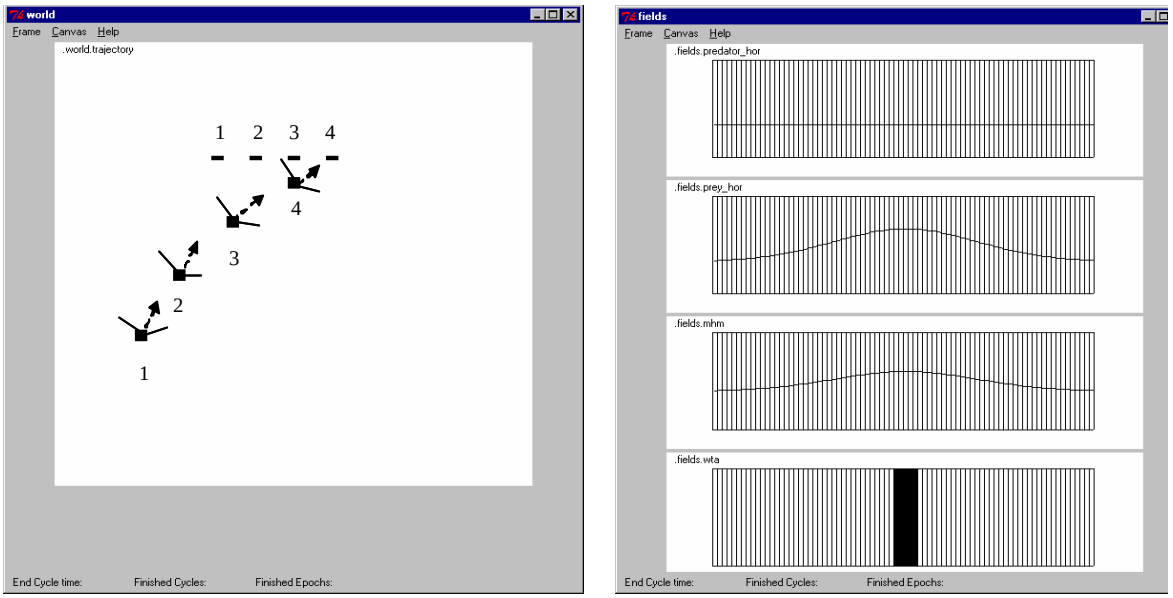

Figure 21. The figure shows (left) the frog (square box) trajectory during prey catching behavior in response to prey (smaller horizontal rectangle) position. The prey moves horizontally independent of actual frog movement. The trajectory shows numbered positions "1" through "4" for frog (dashed arrows) and prey, with frog visual field shown as a "v" shape. Note how the frog gets closer to the prey adjusting its orientation at each time step. The figure (right) shows different activity fields in the neural schema model. From the top: (i) *predator_hor* represents predator activation in *PreTectum/Thalamus* neural schema showing no activity since no predator is present, (ii) *prey_hor* represents prey activation in *Tectum* neural schema with activation centered at the prey location, (iii) *mhm* represents combined prey and predator activation in *Motor Heading Map* (*MHM*), and (iv) *wta* represents winner-take-all activity in *MHM* resulting in movement towards maximum MHM activity.

The physical experiment results are shown in Figure 22. Figure 22 (left) shows the robot trajectory towards a static blue colored cylinder corresponding to prey (a moving prey experiment is described in the multiple robot experiment section). Note that we are simplifying perception by recognizing a blue colored cylinder as the prey. The *Retina* schema calculates distances to the stimulus by computing the number of blue pixels segmented from the visual field of the camera. Due to the absence of obstacles the robot moves towards the prey (*Pursuit* state) stopping at a short distance before the prey (*Attack* and *Eat* states). Figure 22 (right) shows field activities (left portion of the display window) similar to those previously shown in Figure 21. The right portion of the display window shows *visualField* activity in the *Retina* schema displaying the segmented blue cylinder corresponding to prey.

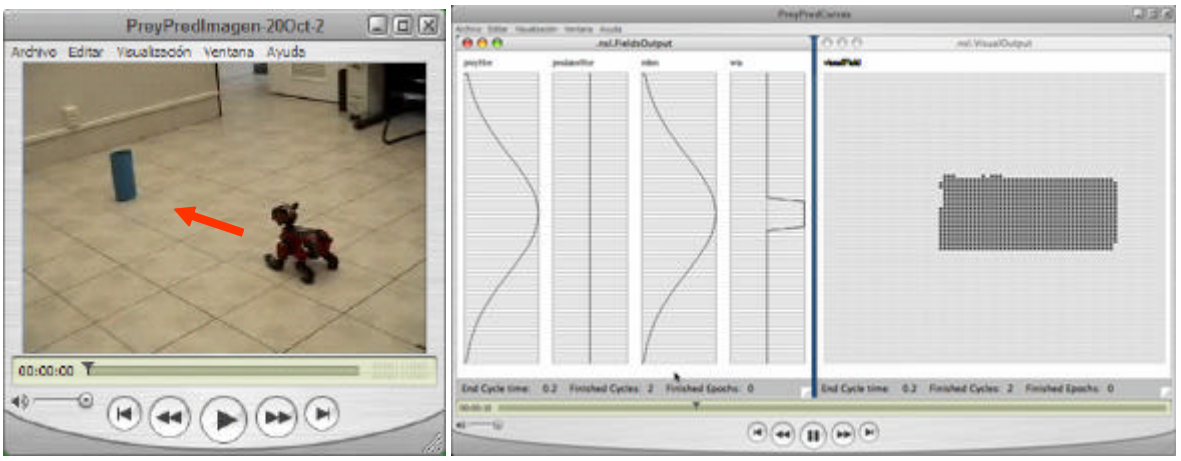

Figure 22. The figure shows (left) the robot trajectory during prey catching behavior in response to prey position. The figure (right) shows a window with two displays. The left display shows the following activity fields from the left: (i) *preyHor* represents prey activation in *Tectum* neural schema with activation centered at the prey location, (ii) *predatorHor* represents predator activation in *PreTectum/Thalamus* neural schema showing no activity since no predator is present, (iii) *mhm* represents combined prey and predator activation in *Motor Heading Map* (*MHM*), and (iv) *wta* represents winner-take-all activity in *MHM* resulting in movement direction towards maximum combined MHM activity. The right display shows *visualField* activity in the *Retina*.

B. Predator Avoidance

We tested a set of basic predator avoidance experiments involving a frog and predator as shown in Figure 23. The simulation shows a frog (square box) avoiding a moving predator (larger vertical rectangle). The frog moves opposite predator direction with step, numbered 1 through 3, where dashed arrows represent frog's visual field. Figure 23 (right) shows different neural schema field activities during the simulation: *predator_hor* represents predator activity in *PreTectum/Thalamus*, *prey_hor* representing prey activity in *Tectum*, *mhm* represents combined prey and predator activities in *Motor Heading Map* (*MHM*) and *wta* (winner-take-all) represents frog movement direction in *MHM*. In real frogs, obstacles correspond to any static objects while preys and predators are recognized only if they move. In particular, preys have a larger horizontal to vertical ratio while predators have a larger vertical to horizontal ratio. To simplify visual processing, preys and predators will be recognized in these experiments by color where blue is used for preys and green for predators.

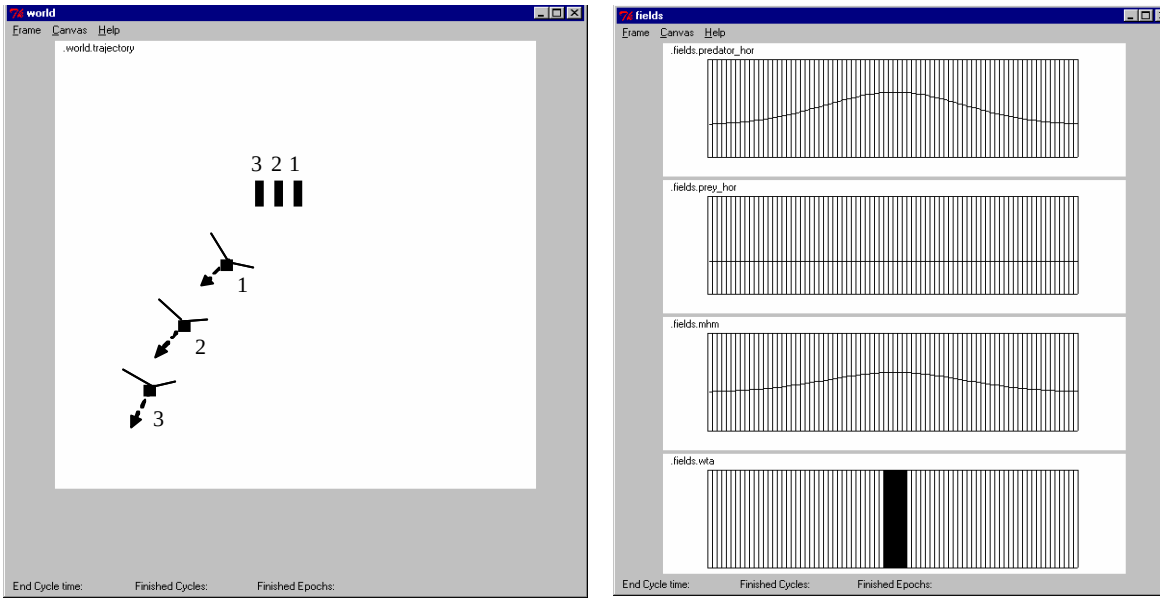

Figure 23. The figure shows (left) the frog (square box) trajectory during predator avoidance behavior in response to predator (larger vertical rectangle) position. The predator moves horizontally independent of actual frog movement. The trajectory shows numbered positions "1" through "3" for frog (dashed arrows) and predator, with frog visual field shown as a "v" shape. Note how the frog gets away from the predator adjusting its orientation at each time step. The figure (right) shows different activity fields in the neural schema model. From the top: (i) *predator_hor* represents predator activation in *PreTectum/Thalamus* neural schema with activity centered at the predator location, (ii) *prey_hor* represents prey activation in *Tectum* neural schema showing no activity since no predator is present, (iii) *mhm* represents combined prey and predator activation in *Motor Heading Map* (*MHM*), and (iv) *wta* represents winner-take-all activity in *MHM* resulting in movement towards maximum MHM activity.

In Figure 24 we show a combination of prey catching and predator avoidance behaviors in a frog in the presence of both moving prey and predator. At first, positions 1 and 2, the predator is outside the visual field of the frog and the frog pursuits the prey. Once the predator enters the frog visual field, position 3, the frog flees away from the predator in direction opposite to the predator position without reacting any longer to the prey. In Figure 24 (left) note how at step 3, the frog visual field points towards the predator. In Figure 24 (right) note the different prey and predator activities, *predator_hor* and *prey_hor* respectively. The *mhm* field shows the combination of both fields giving more weight to the predator, thus resulting in a *wta* pointing towards the predator.

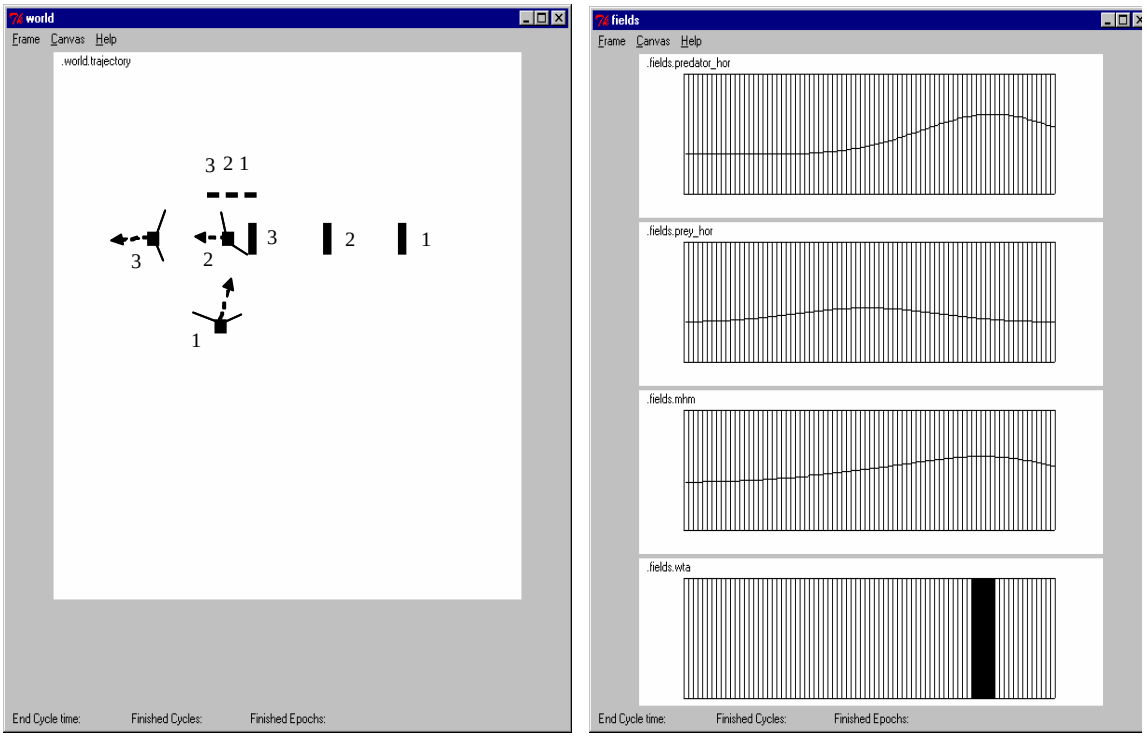

Figure 24. The figure shows (left) the frog (square box) trajectory during combined prey catching and predator avoidance behavior in response to prey (smaller horizontal rectangle) and predator (larger vertical rectangle) position. Both prey and predator move horizontally and independent of frog. The trajectory shows numbered positions starting with "1" ending with "3". Dashed arrows represent frog movement direction while "v" shape represents visual field orientation. Note how the frog initially moves towards the prey (positions 1 and 2) and then gets away from the predator when it is inside its visual field (position 3). The figure (right) shows different activity fields in the neural schema model. From the top: (i) *predator_hor* represents predator activation in the *PreTectum/Thalamus* neural schema with activity centered at the predator location, (ii) *prey_hor* represents prey activation in the *Tectum* neural schema with activity centered at the prey location, (iii) *mhm* represents combined prey and predator activation in the *Motor Heading Map* (*MHM*), and (iv) *wta* represents winner-take-all activity in *MHM* resulting in movement towards maximum MHM activity.

The physical experiment results are shown in Figure 25. Figure 25 (left) the robot trajectory towards a blue colored cylinder corresponding to prey and then away from a green colored cylinder corresponding to predator. Note again that we are simplifying perception by recognizing a blue colored cylinder as prey and a green colored cylinder as predator. The *Retina* schema calculates distances to the stimulus by computing the number of blue pixels segmented from the visual field of the camera. Due to the absence of obstacles the robot moves towards the prey (*Pursuit* state), locations 1 and 2, changing orientation in response to predator presence (*Flee* state). Figure 25 (right) shows field activities (left portion of the display window) similar to those previously shown in Figure 24. The right portion of the display shows *visualField* activity in the *Retina* schema displaying the segmented blue cylinder corresponding to prey and green cylinders corresponding to predator.

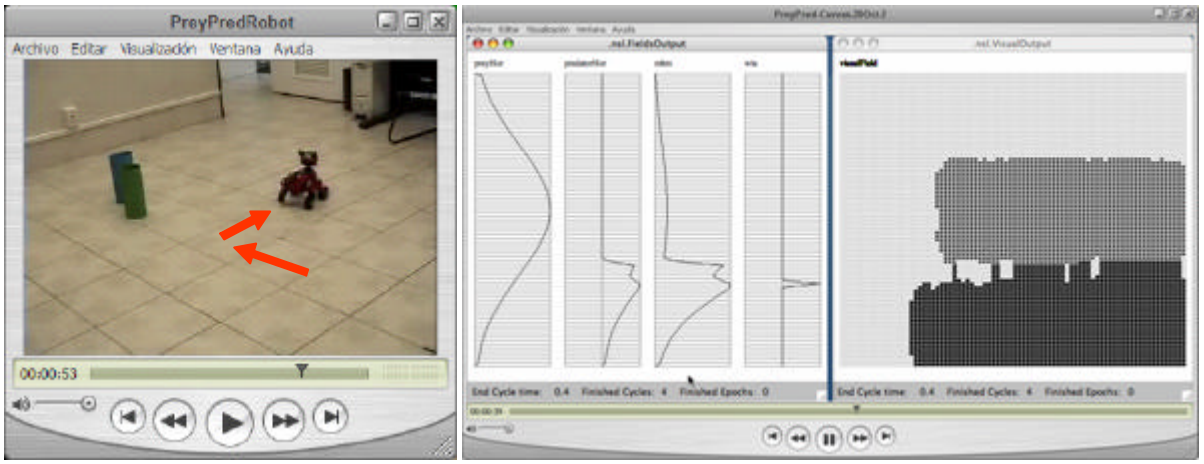

Figure 25. The figure shows (left) the robot trajectory during prey catching and predator avoidance behavior initially in response to prey and then predator. The figure (right) shows a window with two displays. The left displays shows the following activity fields from the left: (i) *preyHor* represents prey activation in *Tectum* neural schema with activity centered at prey location, (ii) *predatorHor* represents predator activation in the *PreTectum/Thalamus* neural schema with activity centered at predator location, (iii) *mhm* represents combined prey and predator activation in *Motor Heading Map* (*MHM*), and (iv) *wta* represents winner-take-all activity in *MHM* resulting in movement direction towards maximum combined MHM activity. The right display shows *visualField* activity in the *Retina* with segmented prey (light rectangle) and predator (dark rectangle).

V. Prey Catching and Predator Avoidance – Multiple Robot Architecture

While neither anurans nor praying mantis display multi-group hunting, there exist other animals with advanced social behaviors. Among these, wolves show advanced hunting by organizing themselves as small packs [51]. This organization enables wolves to hunt animals that are larger and sometimes even faster than themselves, as well as be more compact in response to attacks from outside predators. While very limited neurobiological studies exist on wolves, it is possible in robots to combine studies from different animals into a single comprehensive model. In this section we take the individual anuran model for prey catching and predator avoidance and extend it to multiple robots taking inspiration on wolf pack hunting [41]. The goal of this extended architecture is two fold: (1) To analyze the reusability and scalability of the underlying anuran neural-schema architecture when applied to collaborative robots; and (2) to evaluate the extensibility of the higher level prey catching and predator avoidance individual behavior model to multiple robots. This approach differs from "classical" behavior models developed for multiple robots [11, 15, 35, 36, 38].

The prey catching and predator avoidance model for multiple robots considers a team of wolf predators, i.e. a wolf pack, comprising an *alpha* wolf and several *beta* wolves. Behavioral studies have shown that wolves hunt in packs of about 5 to 20 members keeping a social hierarchy during hunting as well as eating with the stronger *alpha* wolf leading the pack. The wolf pack hunting model described in this work includes the following generalizations:

- Wolf teams are conformed by a group leader (*alpha* wolf) and at least one follower (*beta* wolf).
- *Beta* wolves group around the *alpha* wolf keeping a certain distance from their leader and among themselves.
- Wolves receive only visual information from the environment, using this input to calculate their positions and distances to stimuli. There is no direct communication between wolves.

- Visual fields are limited to a single camera recognizing objects by their color. If an alpha wolf is outside a beta wolf visual field, then the beta wolves loses track of leader.
- Walking speeds are kept constant for all wolves at all times.
- Head direction is kept constant relative to body motions, similar to the anuran model.

We extend the individual anuran prey catching and predator avoidance model from previous section by incorporation formations consisting of one or more beta wolves in addition to the alpha wolf. In the model, beta wolves will try to keep a radial distance r behind the alpha wolf effectively forming a circumference centered at the alpha wolf as shown in Figure 26. In the figure, movement direction d is shown for all wolves. Lines in blue represent beta wolf visual field having angular size $2a$. If q represents the angle between beta wolf moving direction and visual sight of alpha wolf; then when $a < q$ the beta wolf loses track of alpha wolf.

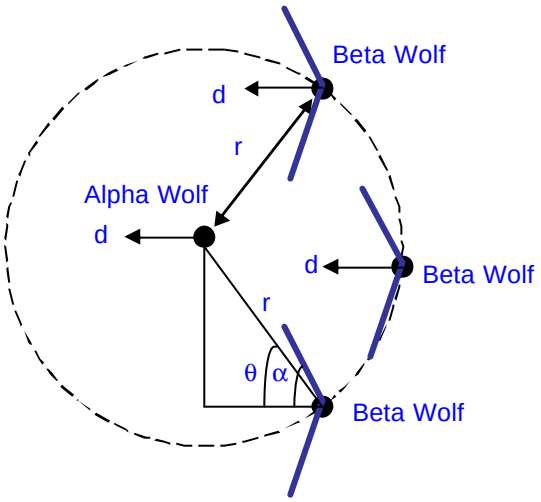

Figure 26. Wolf pack formation diagram. Beta wolves keep behind alpha wolf following a circumference formation centered at the alpha wolf using their visual field to track the alpha wolf.

A. Behavior

The *alpha* wolf prey catching behaviour is the same as previously described for anurans. The *beta* wolf prey catching behavior is described in terms of a state diagram that includes *Formation*, *Pursuit*, *Attack*, *Eat*, and *Wander* as shown in Figure 27:

- *Wander* - In the *Wander* state the wolf explores the environment in search for a prey. The alpha wolf behavior is similar to the individual frog previously described. The beta wolf behaviour considers both prey and leader. If the wolf perceives the leader, *leader_visible* is activated continuing to the *Formation* state. If the wolf perceives a prey (but not the leader), *prey_visible* is activated continuing to the *Pursuit* state.
- *Formation* - As long as the beta wolf continues perceiving the alpha wolf, it will stay to it. If visible contact is lost with the leader then the condition *leader_not_visible* is activated continuing to the

Wander state. If the beta wolf detects a prey (but not the leader) it will change to the *Pursuit* state in response to *prey_visible* activation.

- *Pursuit* - In the *Pursuit* state the goal is to approach the prey without separating from the formation. In the *Pursuit* state there are three possible transitions: one towards the *Formation* state in case the prey is outside its range of vision caused by the *prey_not_visible* condition; the second transition occurs when the prey is detected at a close distance, activated by the *prey_near* condition and continuing to the *Attack* state; and, the third transition occurs when the beta wolf loses track of the alpha wolf, activating the *leader_not_visible* condition and continuing to the *Wander* state.
- *Attack* – In the *Attack* state the animal snaps at the prey activating the *prey_catch* condition. If the animal gets far from the prey the behavior goes back to the *Pursuit* state activated by the *prey_far* condition. The beta wolf will follow the alpha wolf cue in attacking the prey. If the *prey_far* condition is activated the beta wolf returns to the *Pursuit* state. If the prey is caught, the condition *catch_prey* is activated continuing to the *Eat* state.
- *Eat* - In the *Eat* state the wolf eats the animal where beta wolves eat only after the alpha wolf has done so. After the prey has been eaten the condition *prey_not_visible* is activated returning to the *Wander* state. If the prey escapes, the condition *prey_not_catch* is activated returning to the *Attack* state.

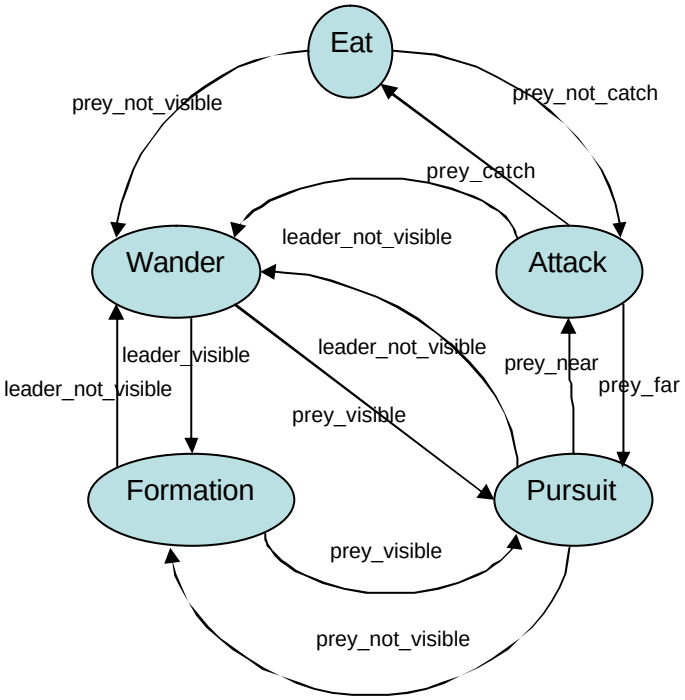

Figure 27. Beta wolf prey catching behavior consisting of *Wander*, *Formation*, *Pursuit*, *Attack* and *Eat* states. Alpha wolf behavior is similar to the individual behavior described in Figure 15.

The predator avoidance behavior is described in terms of a state diagram that includes *Flee*, *Duck* and *Wander* states as shown in Figure 16:

The predator avoidance behavior is similar to the one described in Figure 16 and 17.

B. Schemas

The multiple animal prey catching and predator avoidance behavior is mapped to an extended schema architecture where a new "same species" set of schemas has been added, as shown in Figure 28. The extended schema architecture includes, in addition to the previous schemas shown in Figure 18, a *Same-Species Recognizer*, a *Max Same-Species Selector*, and a *Same-Species Approach* schema.

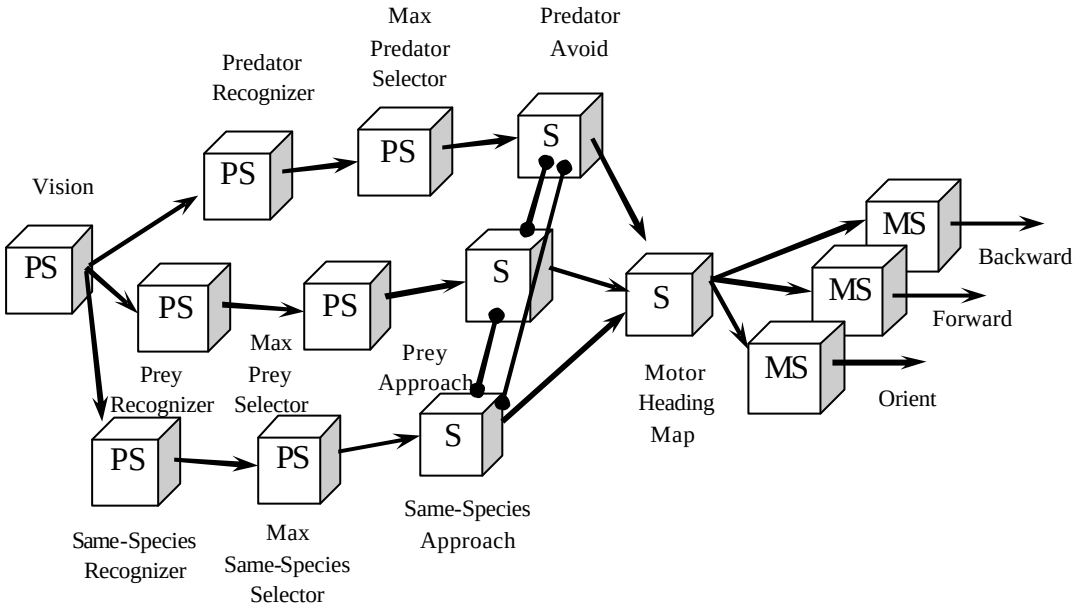

Figure 28. Extended prey acquisition and predator avoidance schema architecture consisting of perceptual schemas (PS): *Visual*, *Prey Recognizer*, *Predator Recognizer*, *Same-Species Recognizer*, *Max Prey Selector*, *Max Predator Selector* and *Max Same-Specie Selector*; sensorimotor schemas (S): *Prey Approach*, *Predator Avoid*, *Same-Specie Approach* and *Motor Heading Map*; and motor schemas (MS): *Orient*, *Forward* and *Backup*.

Perceptual schemas include *Vision*, *Prey Recognizer*, *Predator Recognizer*, *Max Prey Selector*, *Max Predator Selector* in addition to:

- *Same-Species Recognizer* – Recognizes a stimulus from the same species corresponding to the animal leader.
- *Max Same-Species Selector* - Selects among multiple *Same-Specie* stimuli.

Sensorimotor schemas include *Prey Approach* and *Predator Avoid* in additional to:

- *Same-Specie Approach* – Obtains input from the *Max Same-Specie Selector* schema generating as output an attractant field whose strength decays proportional to the distance to the same-specie stimulus.

Motor schemas include *Orient*, *Backward* and *Forward* as before.

C. Neural Networks

The neural schema architecture is similar to the previous architecture with the addition of a link between *Same-Species Approach* and *Tectum* inspired in studies that map same species recognition (*mates*) to the *Tectum* [14]. In Figure 29 we show mappings between schemas and the underlying neural networks:

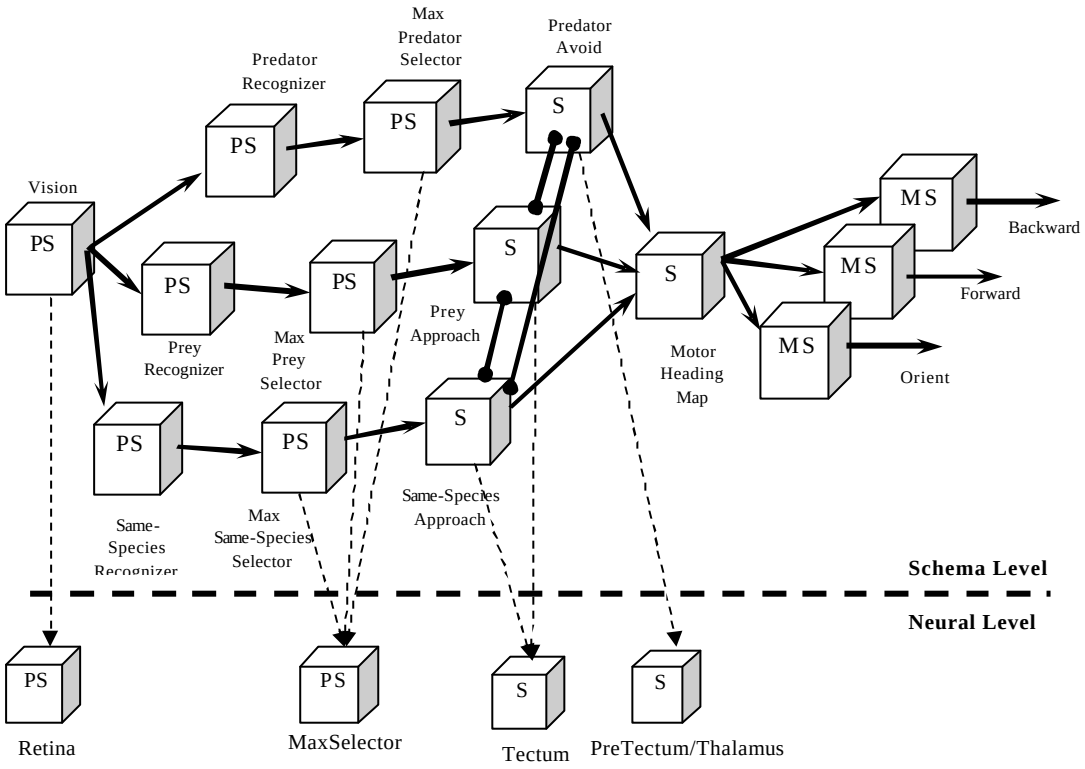

Figure 29. Prey catching and predator avoidance extended multi level neural schema architecture. The schema level consists of perceptual schemas (PS): *Visual*, *Prey Recognizer*, *Predator Recognizer*, *Same-Species Recognizer*, *Max Prey Selector*, *Max Predator Selector* and *Max Same-Species Selector*; sensorimotor Schemas (S): *Prey Approach*, *Predator Avoid*, *Same-Species Approach*, and *Motor Heading Map*; and motor schemas (MS): *Orient*, *Forward*, and *Backup*. The neural level consists of four neural schemas: *Retina*, *MaxSelector*, *Tectum* and *PreTectum/Thalamus*.

VI. Prey Catching and Predator Avoidance – Multiple Robot Experiments and Results

In this section we describe prey catching and predator avoidance experiments on multiple robots inspired on wolf pack studies as described in the previous section. We initially test the experiments under a simulation-only environment and then perform physical experiments with three real robots.

A. Prey Catching

We tested a set of basic prey catching experiments involving a prey, an alpha robot and several beta robots. The simulations shown in Figure 30 were developed using the TeamBots software [10]. Three

beta wolves in black maintain a formation behind an alpha robot in red pursuing a prey in yellow. The beta wolves respond to the presence of the alpha wolf leader.

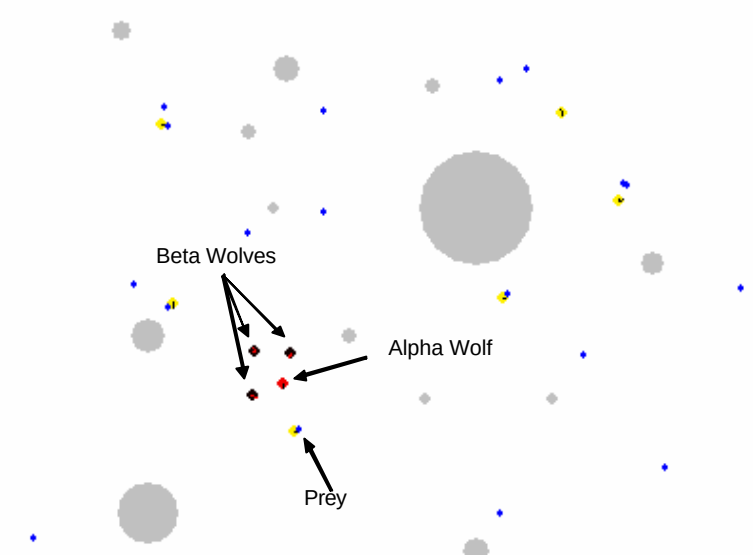

Figure 30. A 4-robot pack formation consisting of one alpha robot in red and three beta robots in black. The robots are pursuing a prey in yellow.

The physical experiment results are shown in Figure 31. Figure 31 (left) shows the robot pack pursuing a blue colored cylinder corresponding to a prey. Alpha wolf is in red and beta wolves in blue. Figure 31 (center) shows the robot pack pursuing the prey after it has moved. All wolves move at the same speed and there are no obstacles in the field. To keep formation the alpha wolf has to be inside the visual field of the beta wolves at all times. Figure 31 (right) shows the alpha and beta wolves surrounding (attacking) the prey when close enough to it. Note that the alpha robot is the first to attack.

Figure 31. Left: Alpha wolf in red and beta wolves in dark blue. Robots pursue the blue colored cylinder corresponding to prey. Beta wolves keep a formation behind their leader. Center: Prey is manually moved with pack tracking it. Beta wolves maintain a formation behind their leader. Right: Eventually the robots get close to the prey attacking it by a surrounding motion.

Figure 32 shows a window with three displays: (bottom) beta wolf segmented view of prey and alpha wolf; (top left) segmented objects in visual field; and (top right) different field activities: (i) the left four graphs represent prey (*preyHor*), predator (*predatorHor*), and alpha (*alphaHor*) recognition

maps; (ii) the next graph represents the motor heading map (*mhm*) integrated movement showing stronger alpha recognition; (iii) the graph to the right shows a step activation corresponding to a winner-take-all (*wta*) dynamics representing output robot orientation. As shown in the figure, in the current stage the beta wolf will follow the alpha wolf and not the prey.

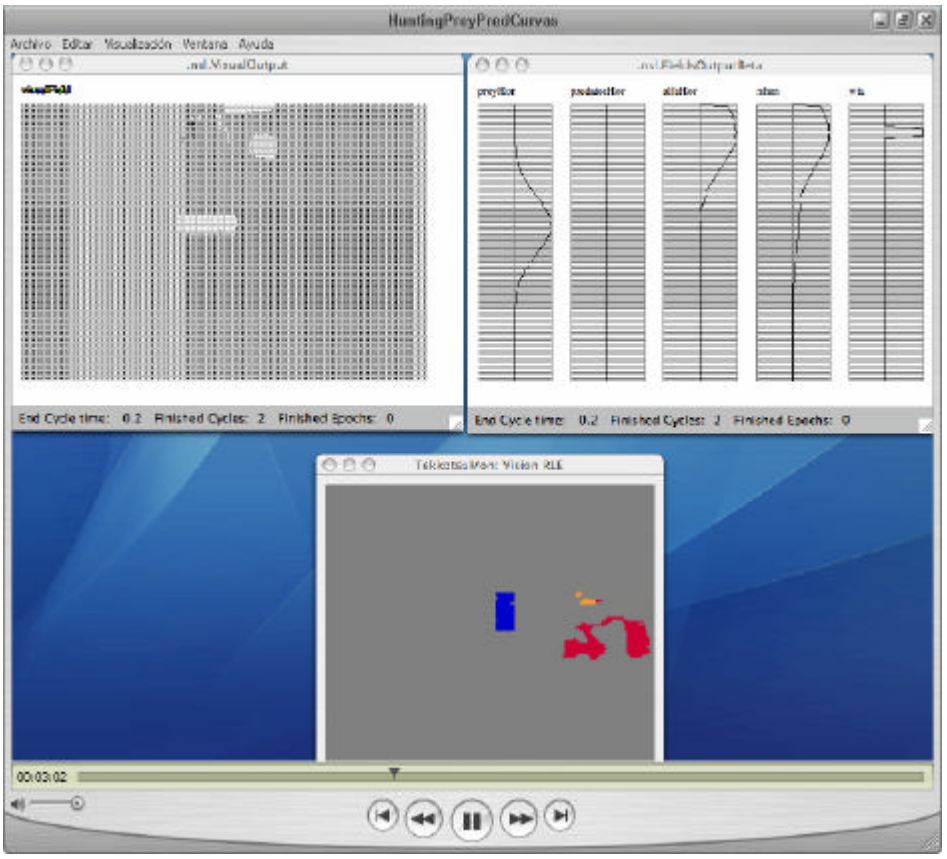

Figure 32. The figure shows (bottom) beta wolf view of prey segmented in blue and alpha wolf segmented in red. Upper graphs show segmented objects in gray (left) and field activities graph (right). From left to right these field activities correspond to: *preyHor*, *predatorHor*, *alphaHor*, *mhm* and *wta*.

B. Predator Avoidance

We tested a set of basic of predator avoidance experiments involve a prey, an alpha robot and several beta robots. These simulations shown in Figure 33 were developed using the TeamBots software [10]. Three beta wolves in black break away from a formation behind an alpha robot as the predator in brown. As opposed to the prey catching behavior where robots react to the leader and the prey, once a predator is perceived robots flee away from the intruder breaking up with the formation.

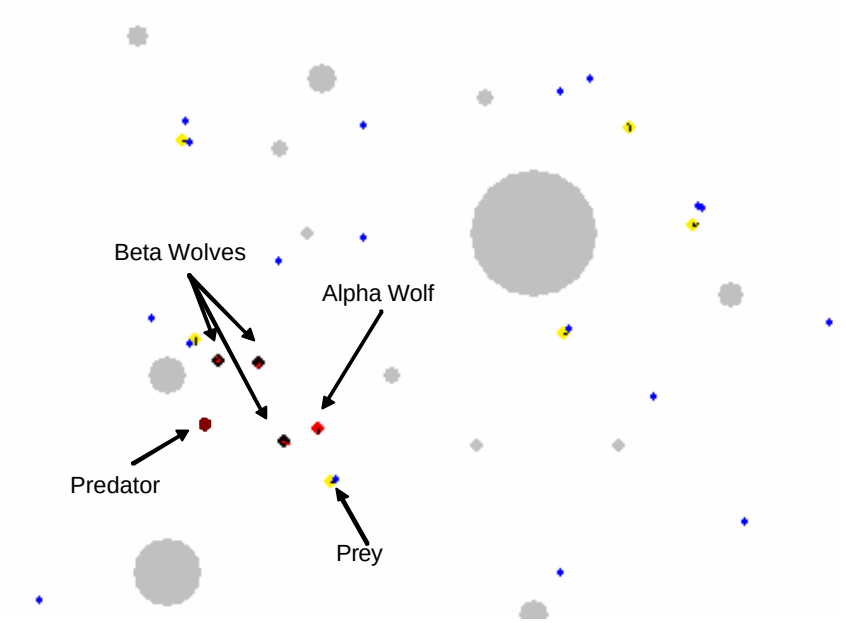

Figure 33. Left: A 4-robot pack formation consisting of one alpha robot in red and three beta robots in black. The robots initially pursuing a prey in yellow start fleeing once the predator in brown appears.

The physical experiment results are shown in Figure 34. Figure 34 (left) shows the robot pack, initially pursuing a blue cylinder corresponding to a prey, now fleeing away from a green cylinder representing a predator. Alpha wolf is in red and beta wolves in blue. Similarly to the individual robot predator avoidance behavior, alpha and beta wolves break away from the pack independent of pack formation in direction opposite to the predator location. Figure 34 (right) shows a window with three displays: (bottom) beta wolf segmented view of prey, predator and alpha wolf; (top left) segmented objects in visual field; and (top right) different field activities: (i) the left four graphs represent prey (*preyHor*), predator (*predatorHor*), and alpha (*alphaHor*) recognition maps; (ii) the next graph represents the motor heading map (*mhm*) integrated movement showing stronger alpha recognition; (iii) the graph to the right shows a step activation corresponding to a winner-take-all (*wta*) dynamics representing output robot orientation. As shown in the figure, beta wolves will react towards predator disregarding the alpha wolf and the prey.

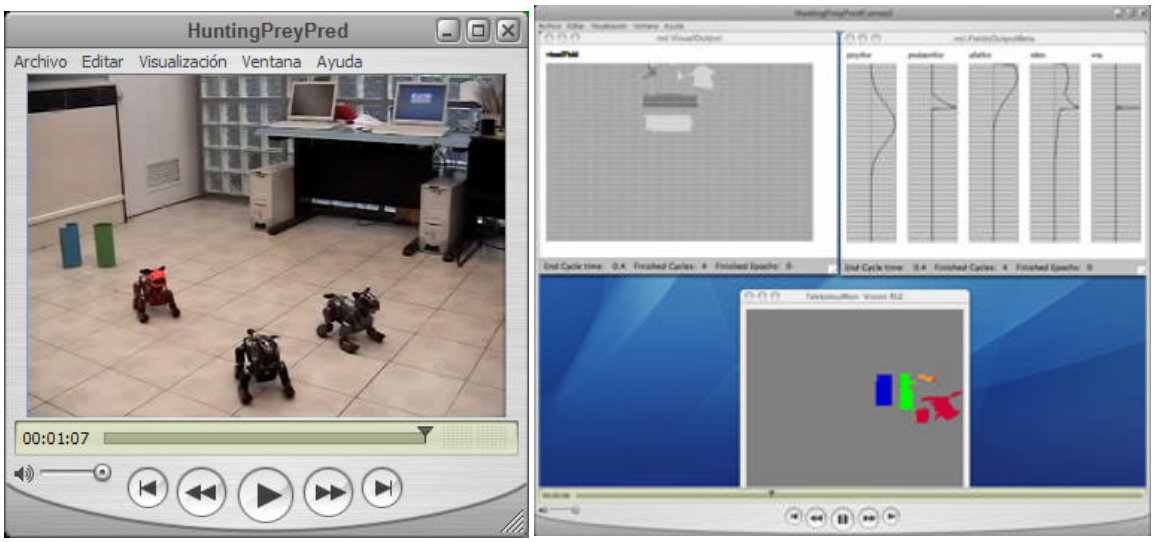

Figure 34. Left: Alpha wolf in red and beta wolves in dark blue. Robots flee away from green colored cylinder corresponding to predator. Right: The figure shows (bottom) beta wolf view of prey segmented in blue, predator segmented in green and alpha wolf segmented in red. Upper graphs show segmented objects in gray (left) and field activities graph (right). From left to right these field activities correspond to: *preyHor*, *predatorHor*, *alphaHor*, *mhm* and *wta*.

VII. CONCLUSIONS

The goal of this work has been to develop a multi-level neural-schema robotic architecture inspired in behavior and brain studies in real animals. We have modeled, based on anuran and praying mantis neuroethological studies, a prey catching and predator avoidance single robot architecture. We also extended the model and architecture to multiple robots taking inspiration in wolf pack ethological studies. While there has been some prior work in prey catching and predator avoidance robotic models and robotic architectures, the work presented in this paper is unique in its multi-level neural schema approach and its extension from single to multiple robots.

The methodology used in this work involves the development of simulations to test behaviors and neural schema models in single and multiple animals followed by its implementation in real robotic systems. We use as single animal simulation platform the NSL/ASL multi-level neural schema systems, while we use as multiple animal simulation platform the TeamBots behavioral modeling system. We execute all robotic models using the integrated NSL/ASL/MIRO system connected in a wireless fashion to a Sony AIBO ERS-210 robot for physical experimentation. Although TeamBots provides linkage to real robots, we only use this environment for preliminary behavior-only multiple robot simulation. Models tested in TeamBots are then integrated with NSL/ASL single animal models in developing multi-level neural schema architectures for multiple robots.

The NSL/ASL arquitecture has been crucial to manage the complexity involved in modeling multi-level neuroethological systems and corresponding neural schema architectures. Under the NSL/ASL architecture schemas represent behavioral patterns while neural networks represent the dynamic and computational properties of the underlying brain mechanisms. As the complexity of the biological systems being modeled grows, it becomes necessary to have powerful simulation tools and robotic systems that let the developer follow good software practices including modularity and top-town and bottom-up designs.

In terms of actual models, prey catching and predator avoidance is inspired primarily in anuran and praying mantis studies with extensions to multiple robots inspired in wolf pack hunting behavior. The behaviors modeled are much more complex in nature than those modeled in the robots. For example, anurans and praying mantis perform a more complex visual object recognition based on motion detection while wolves use both smell and vision to keep pack hierarchy. In the robotic experiments we used only color to detect objects in order to simplify perception limiting robot coordination to sight only. The limited robot visual field directly affects pack tightness. Other aspects that will make behaviors more realistic include: use of more robots during experimentation, addition of motivational variables like fatigue and hunger, variation of robot speeds and turning angles, the inclusion of more than one prey and predator, use of robots for both preys and predators, and the extension to new behaviors. In terms of neural mechanisms in anurans and praying mantis we plan to analyze the effect of additional factors in prey catching and predator avoidance behavior including additional neural schemas, extension of existing ones to adapt to later studies and the inclusion of motivational factors such as hunger (internal) and season and time (external) at the neural level. These aspects become critical in robot architectures that intend to exhibit adaptation and learning [12].

Finally, all robot experiments can be retrieved from our web site [45].

ACKNOWLEDGMENT

This work has been supported in part at USF by ARO and SPAWAR; at ITAM by CONACYT: UC-MEXUS, LAFMI, and NSF-42440; and by "Asociación Mexicana de Cultura".

REFERENCES

1. Arbib, M.A., Levels of Modeling 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.: Neural Mechanisms of Visuomotor Coordination: The Evolution of Rana computatrix. In: Visual Structures and Integrated Functions. Springer–Verlag, Berlin Heidelberg (1991).
4. Arbib, M.A., Schema Theory, in the *Encyclopedia of Artificial Intelligence*, 2nd Edition, Editor Stuart Shapiro, 2:1427-1443, Wiley (1992).
5. Arbib, M.A., Schema Theory, in *The Handbook of Brain Theory and Neural Networks*, 2nd Edition, Editor Michael Arbib, 993-998, MIT Press (2002).
6. Arbib, M. A., Cobas, A., Prey-catching and predator avoidance 1: Maps and Schemas. In: Visual Structures and Integrated Functions. Springer – Verlag, Berlin Heidelberg (1991).
7. Arkin, R.C., "Neuroscience in motion: the application of Schema Theory to Mobile Robotics". In, *Visuomotor Coordination: Amphibians, Comparisons, Models, and Robots*, eds. J.P. Ewert and M.A. Arbib. New York: Plenum Press, pp. 649-672 (1989).
8. Arkin, R.C., *Behavioral based Robotics*, MIT Press (1998).
9. 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).
10. Balch, T., Teambots 2.0, http://www.cs.cmu.edu/~trb/TeamBots/ (2000).

11. Balch, T. and Arkin, R.C., Motor Schema-based Formation Control for Multiagent Teams, *IEEE Transactions Robotics and Automation* (1999).
12. Barrera, A., and Weitzenfeld A., Bio-inspired Model of Robot Adaptive Learning and Mapping, *Proceedings IROS 2006 – International Robots and Systems Conference*, Beijing, China, Oct 9-13 (2006).
13. Bekey, G., *Autonomous Robots: From Biological Inspiration to Implementation and Control*, MIT Press (2005).
14. Betts, B., The toad optic tectum as a recurrent on-center off-surround neuralnet with quenching threshold, ICNN, IEEE International Conference and Neural Networks, Vol 2, pp 47-54, July 24-27 (1988).
15. Cao, Y.U., Fukunaga, A.S., and Kahn, A.B., Cooperative mobile robotics: antecedents and directions, *Autonomous Robots*, 4(1) pp 7-27, Kluwer Academic Publisher (1997).
16. 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).
17. Cervantes-Pérez, F., Guevara-Pozas, D. and Herrera, A. "Modulation of prey-catching behavior in toads: data and modeling". In: *Visual structures and integrated functions*, eds. Arbib, M.A. and Ewert, J.P., Springer Verlag Research Notes in Neural Computing, vol. 3, pp. 397-415 (1991).
18. 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. Rudoman, M.A. Arbib, F. Cervantes-Perez, and R. Romo, Springer Verlag Research Notes in Neural Computing, Vol 4, pp. 426-449 (1993).
19. 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).
20. Cobas, A., Arbib, M. A.: Prey-catching and predator avoidance 2: Modeling the medullary hemifield deficit. In: Visual Structures and Integrated Functions. Springer – Verlag, Berlin Heidelberg (1991).
21. 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).
22. 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).
23. Corbacho, F., and Arbib M. Learning to Detour, *Adaptive Behavior*, Volume 3, Number 4, pp 419-468 (1995).
24. Corbacho, F., and Weitzenfeld, Retina, in *The Neural Simulation Language NSL, A System for Brain Modeling*, pp 189-206, MIT Press, July 2002.
25. Didday, R.L., A model of visuomotor mechanisms in the frog optic tectum, *Math. Biosci.* 30:169-180, (1976).
26. Ewert, J.P. *Neuroethology: an introduction to the neurophysiological fundamentals of behavior*. Berlin, Heidelberg, New York, Springer Verlag (1980).

27. Ewert, J.P. "Tectal mechanisms that underlie prey-catching and avoidance behaviors in toads". In: *Comparative neurology of the optic tectum*, ed. Vanegas, H., Plenum Press, pp.247-416 (1984).
28. Ewert, J. P.: Commands Neurons and Command Systems. In Arbib, M. A (ed.): The handbook of brain theory and neural networks. Cambridge, Mass. The MIT Press (1995).
29. Ewert, J.P. and Kehl, W. "Configural prey-selection by individual experience in toad *Bufo-Bufo*. *J. Physiol*., 126:105-114 (1978).
30. Grüsser, O.J. and Grüsser-Cornehls, U. "Neurophysiology of the anuran visual system". In: *Frog neurobiology*, eds. Llinás, R., and Precht, W. Berlin, Heidelberg, New York, Springer Verlag, pp. 297-385 (1976).
31. 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).
32. Lara, R, and Arbib, M.A. "A model of the neural mechanisms responsible for pattern recognition and stimulus–specific habituation in toads". *Biological Cybernetics*, 51:223-237 (1985).
33. Mech, D.L., *The Wolf. The Ecology and Behaviour of Endangered Species*. The Natural History Press (1970).
34. NSL Web Sites, http://www.neuralsimulationlanguage.org/ and http://nsl.usc.edu (2002).
35. Parker, L., Current State of the Art in Distributed Autonomous Mobile Robotics, *Proceedings International Symposium on Distributed Autonomous Robotic Systems*, pages 3-12, Knoxville, TN (2000).
36. Reynolds, C.W., Flocks, Herds, and Schools: A Distributed Behavioral Model, *ACM SIGGRAPH '87 Conference Proceedings*, Anaheim, California (1978).
37. 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).
38. Sgorbissa, A., and Arkin, R.C., Local Navigation Strategies for a Team of Robots, *Robotica* 21(5) pp 461-473, Cambridge University Press (2003).
39. Steenstrup, M., Arbib, M.A., Manes, E.G., Port Automata and the Algebra of Concurrent Processes, J. Computer Syst. Sci., Vol. 27, no. 1, pp. 29-50, Aug (1983).
40. 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).
41. Vallesa, A., and Weitzenfeld, A., Multi-Agent Formations in a Pack of Wolves Hunting Model, *Proceedings 1st Latin American Robotics Symposium, LARS 2004*, Mexico City (2004).
42. Wang, D. "A neural model of synaptic plasticity underlying short-term and long-term habituation". *Adapt. Behav.*, 2:111-129 (1993).
43. Webb, B., What does robotics offer animal behaviour?, *Animal Behaviour*, 60, 545-558 (2000).
44. 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).
45. Weitzenfeld, A., Prey Approach and Predator Avoidance Single and Multiple Robot Videos, ftp://ftp.itam.mx/pub/alfredo/videos/PreyPred (2007).

46. Weitzenfeld, A., From Schemas to Neural Networks: A Multi-level Modeling Approach to Biologically-Inspired Autonomous Robotic Systems, Journal of Robotics and Autonomous Systems, Elsevier (2007).
47. Weitzenfeld, A., Arbib, M., Alexander, A., *NSL - Neural Simulation Language: A System for Brain Modeling*, MIT Press (2002).
48. 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).
49. 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).
50. Weitzenfeld, A., Peguero, O., Gutiérrez, S., NSL/ASL: Distributed Simulation of Modular Neural Networks, MICAI 2000 Mexican International Conference on Artificial Intelligence, Acapulco, Mexico, April 14-18, LNAI 1793, Springer-Verlag (2000).
51. Weitzenfeld A., Vallesa, A., and Flores, H., A Biologically-Inspired Wolf Pack Multiple Robot Hunting Model, Proc of Latin American Robotics Symposium LARS 2006, Santiago Chile, Oct 26-27, (2006).