

Modulatory Influence of Motivations on a Schema-Based Architecture: a Simulative Study

Giovanni Pezzulo and Gianguglielmo Calvi

ISTC-CNR, Via S. Martino della Battaglia, 44 - 00185 Rome, Italy
giovanni.pezzulo@istc.cnr.it gianguglielmo.calvi@noze.it

Abstract. We analyze the role of motivations in living organisms, and the nature of their influences on behavior with the aim to propose a design methodology for schema-based agent architectures. We propose that motivations have a modulatory influence on behavior, and in our design methodology they regulate the allocation of resources to the sensorimotor system and schemas. We describe an agent architecture incorporating this principle and we highlight its performance in a simulative study¹.

1 Introduction

Neurobiological evidence suggest that the organism’s motivational system is its first and foremost source of activity, that it determines the architecture and organization of its brain, and that an organism without feelings is severely limited in its adaptivity, survivability, and autonomy [4]. In a living organism the motivational system is shaped for maintaining in acceptable bounds bodily variables that are crucial for survival: internal signals, feelings and drive stimuli, carry on information related to the needs of the organism, and motivate the organism to look for appropriate external stimuli. We can say then that in a living system actions are selected in order to satisfy motivations, and not to respond to external stimuli as such: a machine is run by stimuli, an agent acts according to its motivations. Since in order to fully understand living organisms and their autonomy it is crucial to focus on their internal context, here we propose that this methodology should be adopted in the design of artificial systems, too. Contrary to the typical emphasis on inputs and outputs, we then focus first and foremost on the motivational dimension of agents, we then investigate the relations between motivations and behavior with the aim to provide a design methodology for incorporating motivations in a schema-based agent architecture.

How Do Motivations Impact on Behavior? What is special about the “processing” of motivations? why do feelings and drives stimuli are different from external stimuli and affordances? Here we suggest that the motivational system has a way of influencing the sensorimotor system that is distinct from the effects of the stimuli from the environment: it *modulates* behavior in a very broad, often

¹ Work supported by the EU-funded project **MindRACES** (FP6-511931).

unselective way (‘energizing’ effect in [8]), and produces qualitative effects only indirectly. According to [9]: “The manner in which the nervous system responds to the inputs from the external environment can be called neuro-transmissory whereas the manner in which it responds to influences from the rest of the body can be called neuro-modulatory.” This fact has rarely been acknowledged in AI (with exceptions, see later); but treating motivational and perceptual influences in the same way we risk to have ‘hyper-rational’ models of emotions and motivations, and ultimately to reduce them to appealing labels.

Including Motivations in Agents’ Design Our design methodology consists in introducing in a schema-based agent architecture [1, 2] a motivational component which interacts that other components of the architecture in an energizing, modulatory and unselective way. This means that while specific behavioral responses exist for stimuli and affordances in the environment, this is not the case for feelings and motivational influences. Motivations and emotions have directing and energizing effects, and produce behavioral consequences such as selecting stimuli, orienting attention, facilitating action, etc. However, these specific behavioral responses are not designed but emerge from the interaction between the motivational and the sensorimotor system (either immediately, or indirectly, by learning the value of actions and objects). Different responses such as ‘fight, flee or freeze’ do not depend on three specific modules or schemas, but on the influences of different internal contexts over the same sensorimotor system.

Our work is related to similar schema-based architectures (e.g., [1, 2]; see also [14]). However, in our design methodology the motivational system influences the sensorimotor system by *regulating the allocation of resources* for action selection and execution. Similar approaches are ‘hormonal modulation of behavior’ in [3], and the idea that emotions are ‘selectors of resources’ [7]. We also emphasize the distributed and decentralized nature of resources allocation, and the fact that by manipulating the allocation of resources in the sensorimotor system qualitatively different behaviors can be produced. In this perspective, the main influence of motivations on behavior is ‘covert’, but indirectly it results in ‘overt’ actions such as the selection of a specific behavior or an attentional switch.

2 The Computational Model: the Mantis Architecture

We have implemented a schema-based agent architecture in which the sensorimotor and motivational systems interact. We have used the cognitive modeling framework AKIRA (<http://www.akira-project.org/>), the AKIRA Schema Library (AKSL) [12], and the 3-D engine Irrlicht (<http://irrlicht.sourceforge.net/>). The architecture is based on earlier work [10] and is inspired by the ethological model of the praying mantis reported in [2].

Fig. 1 shows the main components of the model: the **motivational system** (the drives) and the **sensorimotor system** (Perceptual and Motor Schemas). The architecture also includes several **routines** for preprocessing of data, two **sensors** and two **actuators**, the camera and wheels.

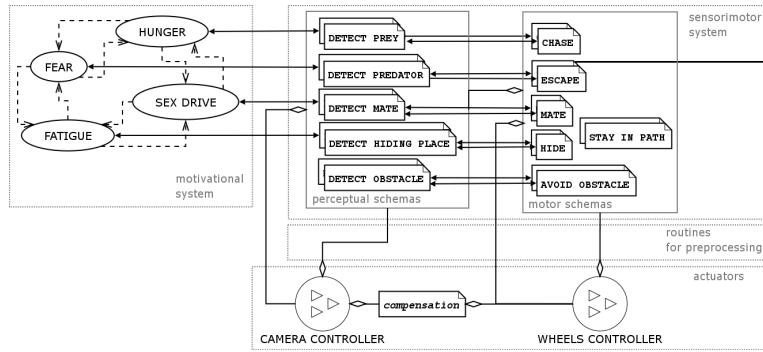


Fig. 1. The components of the mantis architecture.

2.1 The Sensorimotor System: Schemas and Actuators

The architecture includes two kinds of schemas: perceptual schemas realize behaviors such as *escape from predator* and *avoid obstacle*, while motor schemas realize behavior such as *find prey* and *detect obstacle*. Each schema has also an activity level that represents its reliability and usefulness with respect to the external context, as well as its relevance with respect to the motivational context. Depending on the activity level, each schema can influence more the other schemas, by sending them activation or by inhibiting them, receive more or less promptly stimuli from the sensors, and send motor commands with higher or lower firing rate to the actuators.

The whole mantis architecture includes five kinds of perceptual schemas: *detect prey*, *detect predator*, *detect mate*, *detect hiding place*, *detect obstacle*; and six kinds of motor schemas: *stay in path* (the default behavior), *chase*, *escape*, *mate*, *hide*, *avoid obstacle*. For each kind of schema, there are five schema instances, that are specialized for different contextual conditions. For example, two *find prey* schemas can be specialized for dealing with slow or fast preys.

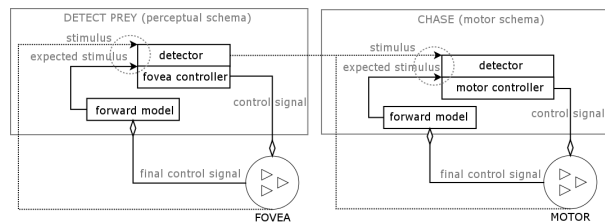


Fig. 2. A coupled perceptual and motor schema.

Perceptual and motor schemas can be *coupled*. The functioning of two sample coupled schemas, *detct prey* and *chase prey*, is illustrated in Figure 2. The perceptual schema receives as input perceptual information from the camera. As indicated by the dotted circle, sensed stimuli are compared with sensory information that is predicted by the forward model, and the error is used for setting the reliability value of the schema (explained later on). The detector thus sends sensory stimuli to the controller (inverse model), which in turn generates a motor command and sends it to the camera motor (via motor routines), and (optionally) sensory information (e.g. the position of the detected prey) to the coupled motor schema. The motor schema receives as input the activity level of the coupled perceptual schema, proprioceptive information about the current state of the wheels' motor, and optionally additional sensory information from the perceptual schema. Like in the perceptual schema, sensed and predicted stimuli are compared and reliability values are assigned. Sensory information is conveyed to the controller, which sends motor commands to the wheels' motor. Notice that in both schemas, in order to generate predictions, the forward models receive an efference copy of the (final) motor commands received by the camera or wheel motors, and learn to predict their sensory effects.

Activity Level. Schemas have an *activity level* reflecting their relevance with respect to external and motivational context. Activity level results in fact from the sum of four parameters. **(1) Absolute value.** Schemas have a varying level of activation that depends on their absolute relevance. For example, schemas such as *escape from predator* are more relevant in absolute than *avoid obstacle* and, ceteris paribus, influence more the agent's conduct. **(2) Reliability.** The expectations produced by forward models are compared with actual stimulus, and schemas predicting well gain activation, like in other schema-based architectures [15]. The rationale is that schemas predicting well are well attuned with the current context; for example, if a slow prey is being chased, schemas specialized for finding and chasing slow prey will generate appropriate predictions. It is important to remark that success in prediction also entails success in action (the converse is not always true); in the above example, if the active schemas continue to generate good predictions, eventually the slow prey will be successfully chased (otherwise they would have generated bad predictions). **(3) Learned links.** Energetic links are learned between schemas with differential Hebbian learning [6], that favor connections between components that are active in the same time span. Schemas (for the same or for different behaviors) that are active in the same context are likely to evolve strong links and are thus able to prime each other. For example, it is likely that during escaping behavior schemas for avoiding obstacles are useful while ones for chasing preys are not. **(4) Motivational context.** Schemas receive activation from related drives via links that are learned with differential Hebbian learning.

Another factor impacts on activation level: there is a limited amount of resources that can be accessed by schemas, called the *energy pool*. Since schemas compete for accessing resources, currently active schemas also inhibit other ones (e.g. *escape* inhibits *chase* and vice versa).

The Actuators Commands received by the schemas are fused by the actuators, the camera and wheels controllers. We adopted an emergent action selection scheme, which favors the most active schemas of the same kind, see [12]. Since more active schemas send motor commands more frequently (with an higher firing rate), they influence more the overall behavior of the agent. This does not mean that other active schemas do not have an influence. For example, when the agent is chasing a prey, occasionally (if *detect predator* is active) it can do a saccade in search of a predator. Our action selection mechanism favors *clean switching*: it permits to switch among behaviors, but avoids to activate ‘intermediate’ ones (e.g., do a saccade in the middle between a predator and a prey). The final behavior is emergent, since it is not encoded in any schema, but depends on the continuous interactions between schemas, the internal context (drives) and the external one (stimuli).

2.2 The Motivational System: Drives

The agent model includes a motivational system composed of four drives: *hunger*, *sex drive*, *fatigue* and *fear*²; see Fig. 1. All drives have an activity level that changes dynamically and is inversely proportional to their degree of satisfaction, that ranges from zero to one for each drive. Drives have also inhibitory links to each other, representing the fact that the motivational system has limited resources and drives compete.

Hunger receives three kinds of feelings. The first represents the need for food of the organism. This is provided by a biological clock (a shortcut for the fact that the body monitoring system notifies the need of food at regular intervals). When such feeling is received, the level of hunger increases. The second represents the anticipated pleasure of receiving food. This is received when the activity level of *detect prey* schemas increases, and slightly raises hunger. This mechanism is adaptive, since it permits to maintain persistence during a chase. The third represent the satisfaction of the need of food. It is notified (by the body monitoring system) when the prey is successfully chased (i.e., when a *chase* schema succeeds) and then food is obtained.

Sex drive functions like *hunger*, with a different biological clock. It is related to *detect mate* and *mate*, not to *detect prey* and *chase*.

Fatigue receives two kinds of feelings. The former represents the consummation of resources for moving, and augments the level of fatigue whenever the agent moves (i.e., whichever schema is used for moving). The second represents the recovery, and restores the initial level of fatigue when the agent successfully rests. The level of fatigue impacts also on the amount of computational resources available to the schemas. In order to act effectively, the agent has to rest (possibly in hiding places) when it is fatigued.

Fear receives two kinds of feelings. The former represents the anticipated pain of being captured by a predator; it is received when the activity level of

² In our schema-based model, like in similar ones [2], *fear* is considered as a motivational variable that influences predator avoidance. An alternative view is that fear is the feeling originating from the drive to avoid predators and other dangers.

detect predator increases and increases the level of fear. The latter represents the anticipated pleasure of escaping the predator; it is received when the activity level of *detect predator* decreases (or remains low) and decreases the level of fear.

Drives send activation to schemas satisfying their conditions; for example, *hunger* makes schemas such as *detect prey* and *chase* more active. Drives that are far from satisfaction are able to send more activation to related schemas, and vice versa. At the same time, schemas activation level feedbacks on related drives (e.g. *detect prey* and *chase* send activation to *hunger*). This characteristic, called *hysteresis* [13] or persistence, permits to maintain behavior elicited by a stimulus after it has disappeared (otherwise an organism that loses visual contact with a predator would stop escaping it).

3 Experimental Set-Up

We have tested the mantis architecture in a simulated predator-and-prey scenario designed with the 3D simulator Irrlicht. The scenario consists in a 3D open space, with hills and valleys that partially occlude vision, with coordinates that vary between -10000 and +10000 in the three axes. We have introduced from one to five predators, preys, mates, obstacles and hiding places. Each of these entities is defined by three features: color, size, and shape, that range between zero and one. As an average, entities have 30% of overlapping features.

We have implemented the agent architecture described here by using the AKSL library [12], which is based on the multi-thread framework AKIRA [11]. Each component, schema or drive, is implemented by using a single thread of execution whose activity level is dynamically set. The amount of resources available to schemas is limited; since schemas compete for gaining activation, this means that the most active ones also inhibit the other ones. For all the simulations described here, we have set this limit to 5.0.

Drives Drives have an activity level that varies between zero and one. A Fuzzy Cognitive Map [6], a hybrid neural network - fuzzy system, calculates dynamically the amount of resources assigned to each drive (and to its thread of execution) on the basis of the input they receive from the body monitoring system and the sensorimotor system. All their inhibitory links are set to -0.6. The weights of the links from and to the schemas are learned with differential Hebbian learning. As described earlier, two drives (*hunger* and *sex drive*) depend on a ‘biological clock’. *Hunger* is increased by 0.05 every seconds, while *sex drive* is set to one after exactly one and two minutes from the beginning of the simulation. Fatigue is increased by 0.01 every ten seconds.

Sensors and Actuators. A simulated camera which can monitor an angle of 90° in front of the agent collects the estimated position in the three axis ($\langle x_e, y_e, z_e \rangle$) of one feature per time in the visual field and conveys it to the perceptual schemas. The input was modified by randomly adding or subtracting 5% of its value as noise. Preprocessing units, the routines, are used for conveying sensory

information only to perceptual schemas which are specialized in the appropriate subregion of the feature (e.g., to *detect prey* schemas that are specialized for small preys). Each schema only receives information by a subset of routines, which is pre-determined. The camera and wheels motors collect motor commands sent asynchronously and with different firing rates from perceptual and motor schemas respectively, and fuse them for obtaining the next fixation point of the camera $\langle x_n f, y_n f, z_n f \rangle$ and the next position of the agent $\langle x_n p, y_n p, z_n p \rangle$.

Schemas. Each schema has a fixed cycle which is however executed at a speed which depends on its current activity level (calculated anew at the beginning of each cycle). The activity level is set by summing up the four parameters introduced above: absolute value, reliability, learned links, and motivational context. The values are then normalized, and so each schema has an activity level which varies between zero and one. For each cycle, the three components of each schema illustrated in fig. 2 perform one single operation.

The *detector* collects the estimated position $\langle x_e, y_e, z_e \rangle$ of the features it is specialized to deal with (from one to three) from the camera (the case of perceptual schemas) or the coupled perceptual schema (the case of motor schemas). It also collects the predicted position generated by the forward model $\langle x_p, y_p, z_p \rangle$. It compares them, and the error in prediction ($1 - \|\langle x_e, y_e, z_e \rangle - \langle x_p, y_p, z_p \rangle\|$) is used for determining the reliability value of the schema. The detector of the perceptual schemas fed $\langle x_e, y_e, z_e \rangle$ to the detector of the coupled motor schema, too.

The *inverse model* receives the estimated and predicted position of the feature(s), and generates a motor command to send to the effector, representing the next fixation point of the camera $\langle x_n f, y_n f, z_n f \rangle$ (the case of perceptual schemas) and the next position of the agent $\langle x_n p, y_n p, z_n p \rangle$ (the case of motor schemas). For eight schemas (*detect prey*, *detect predator*, *detect mate*, *detect hiding place*, *detect obstacle*, *chase*, *mate*, *hide*) the inverse model generates the appropriate motor command for minimizing the distance between $\langle x_e, y_e, z_e \rangle$ and $\langle x_p, y_p, z_p \rangle$. For two schemas (*escape* and *avoid obstacle*) it generates the appropriate motor command for maximizing their distance.

The *forward model* receives an efference copy of the final motor command executed by one actuator and generates the prediction $\langle x_p, y_p, z_p \rangle$ ³.

³ For example, the *detect prey* schema specialized for small preys (i.e., a subregion of the size feature) receives the information that a small entity is in front of the agent, and predicts that a small entity will be next sensed on the left. The sensed and expected position (generated by the forward model) are then used for generating a command to the camera to move left, in order to maintain the prey in sight. *Detect prey* feeds the sensed position to the coupled *chase prey* schema, too, which predicts the next position of the prey and generates appropriate motor commands to reach it. At the same time, other perceptual schemas such as *detect obstacle* receive the information that a small entity is in front of the agent (and, indirectly, other motor schemas, too). Arguably, they generate different predictions about the next position of the small entity (if it is an obstacle, it remains in place). Schemas' reliability, that is their accuracy in predicting, serves to calculate their activity level and then their

Learning. The drives were entirely designed by hand, as above described. We have also designed by hand about one hundred schemas specialized for processing different subregions of the features of the entities. For example, we designed several *detect prey* schemas detecting and predicting different subregions of color and/or size and/or shapes of preys. All the schemas had the same absolute value: 0.1. The mantis architecture, shown in figure 1, was then developed through a two-stages learning process. In the former phase the objective is to learn the inverse and forward models of each schema, and to store only the most reliable ones. In the latter phase the objective is to permit to all the schema to operate together, and to learn the schema-schema and drive-schema links.

First stage. In the former stage each schema learned its inverse and forward models individually in a simple environment having a limited number of entities (from one to five preys, predators, mates, obstacles or hiding places of the same kind). Basically the task of a forward model is to learn to predict the trajectory of an entity which is defined by one to three (subregions of) features. Each predator, prey and mate moves with a specific trajectory in the environment; circular and oval trajectories having different amplitudes were used. Notice that several factors contribute to make these trajectories difficult to learn. First of all, the environment is 3D: trajectories that are circular in 2D are much more complex with hills and valleys since even the values in the z axis varies over time. Moreover, the agent has a limited visual field, the target can be partially occluded by obstacles, and there is noise. The task of the inverse model is instead to learn to generate appropriate motor commands for minimizing or maximizing the distance between the sensed and predicted positions. Again, several factors contribute to make this task more complex, the most significant one being the complexity of the 3D environment.

The inverse and forward models inside each schema are realized by using feed-forward neural networks. The inverse models have three input nodes for each feature the schema is specialized to deal with, representing the position in the three axis of the features, plus three input nodes representing the predicted position of the entity in the three axis, calculated by the forward model. They have three output nodes, representing the motor commands to send to the effector: the next fixation point of the camera (the case of perceptual schemas) and the next position of the agent (the case of motor schemas) in the three axis. The forward models have three input nodes for each feature the schema is specialized to deal with, representing the position in the three axis of the features, plus three input nodes representing the final motor command executed by the effector. They have three output nodes, representing the predicted position of the entity in the three axis. The networks were trained with the cascade-correlation algorithm [5] that estimates autonomously during learning the best number of neurons in the hidden layer. They learned via a free interaction of the agent with the entities in the environment. One example was sampled every twelve,

influence on the final agent's behavior, since more active schemas read more quickly from the sensors and send more motor commands to the effectors.

with a total of thirty-six; the learning stopped when the error of at least one forward model (the euclidean distance between the actual and predicted position in 3D, $0.1 * 10^{-6}$) was less than 0,0000001 (positions vary between -10000 and +10000 in the three axes). After learning we selected only the five more reliable perceptual and motor schemas of each kind (i.e., those predicting better): five *detect prey*, five *chase*, five *detect predator*, etc. In total we selected 55 schemas.

Second stage. In the second stage all the previously selected schemas were integrated in an unique architecture having also the motivational system (the drives). The challenge is now to coordinate them in a complex environment. The agent architecture interacted freely with all the entities of the environment in 3-minutes simulations. In this phase the inverse and forward models did not learn any more, but drives and schemas which were active in the same span of time evolved energetic links (in addition to those shown in Fig. 1) with differential Hebbian learning [6]; see [11] for the details. As a result, schemas which operate on the same entities learns to prime each other, and activation is conveyed in a context-sensitive way from less to more relevant schemas.

Notice that the learned schema-schema and drive-schema links have weights which concur to determine the activity level of the schemas (they correspond to the ‘learned links’ and ‘motivational context’ parameters). This means that the same schemas can have different roles and importance in different agent architectures depending on the arrangement of the whole system.

3.1 Study 1: Adaptive Advantages of the Motivational System

Which are the adaptive advantages of having drives? We have argued that motivations make pragmatic activity more efficient, and we tested this prediction in a set of simulations in the above described scenario.

We have compared in the same environmental conditions two agent architectures. The former (Mantis) is the previously described one, having a full fledged motivational and sensorimotor systems. In the latter (Mantis-wf) the energetic links from the motivational system to the sensorimotor system are removed, then they can not influence one another –in a sense, AG2 is deprived of its feelings.

We have conducted 100 real-time, 3-minutes simulations for each of three experimental conditions: (1) the environment only contains one entity of each kind (one predator, one prey, one mate, one obstacle, and one hiding place); (2) the environment contains three entities of each kind; (3) the environment contains five entities of each kind. We have then estimated the performance and adaptivity of each agent (Mantis vs. Mantis-wf) in the environment by measuring how much their four drives are satisfied throughout the simulations. The value of each drive ranges from zero (totally unsatisfied) to one (totally satisfied), and represents a poor or a good level of vital bodily variables. Three analysis of variance (ANOVA) with mean drives satisfaction (calculated as the mean value of $1 - (fatigue + fear + hunger + sex\ drive) / 4$) during the simulations) as dependent variable were carried out, one for each experimental condition.

In the first experimental condition *Mantis* had a mean satisfaction of 0,818, while *Mantis-wf* had 0,427. In the second experimental condition *Mantis* had 0,835 and *Mantis-wf* had 0,415. In the third experimental condition *Mantis* had 0,843 and *Mantis-wf* had 0,403. In all conditions *Mantis* performs significantly better than *Mantis-wf* ($p < .00001$ with ANOVA in all cases). An agent deprived of its feelings can not select the right affordances in its environment, and its performance is very poor. This fact derives from many converging factors. First of all, *Mantis-wf* is almost totally driven by the presence of stimuli and affordances, which can appear quite randomly. This could be appropriate if the agent had only one need, but it is extremely maladaptive if its drives and necessities change with time, as it is the case in living organisms. Another relevant factor is that drives make pragmatic activity much more efficient, for example by providing persistence: even if the visual contact is lost, the agent does not change objective too often (i.e. energetic resources are not transferred too often among schemas of different kinds). We illustrate this phenomenon in a second study.

3.2 Study 2: Persistence provided by motivations

In this study we have compared three identical agent architectures, AG1, AG2 and AG3, all having full fledged motivational and sensorimotor systems, like *Mantis*. Our aim is to analyze the role of motivations in the selection of alternative actions and in determining persistence. We have tested AG1, AG2 and AG3 individually in the same task, illustrated in Figure 3 (for the sake of simplicity, we have illustrated it in 2D, although the environment is 3D). Each agent has two tasks: catching a static prey (square, center) and, at the same time, escaping a moving predator (double circle, top left) that enters in their visual field shortly after the beginning of the simulation. We have set their levels of *hunger* differently. AG1 and AG2 have respectively an initial value of 0.6 and 0.3, and these values are allowed to vary during the simulation (as previously explained). AG3 has instead an unrealistically high (and maladaptive) value, since it is allowed to recruit all the activity level of the motivational system, and its value is not allowed to change during the simulation.

As Figure 3 shows, only AG1 captures the prey and escapes the predator. AG2 only escapes the predator, while AG3 is captured. Both AG1 and AG2 detect the predator at instant 1, and they switch accordingly their behavior, avoiding to go in its direction. The difference in behavior between AG1 and AG2 depends on their different motivational levels, that provide different levels of persistence. While AG2 loses persistence after the detour caused by the predator, AG1 is able to come back again to the prey, and to fulfill both the tasks. After the detour, AG1 and AG2 lose visual contact with the prey. The stimulus can not be used to trigger *detect prey* and *chase*. If the energy provided by the motivational state is poor, as in AG2, there is no way to remain committed; and for this reason AG2 continues going in the opposite direction. On the contrary, in AG1 the value of *hunger* is high, and it continues fueling the *detect prey* and *chase* schemas even in the period in which they are unsuccessful, since they lack visual stimuli. AG1 remains committed to the initial motivation of capturing the prey, and when

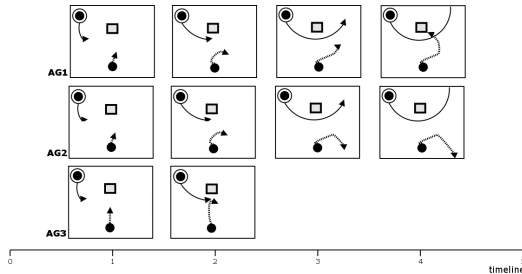


Fig. 3. Different trajectories of AG1, AG2 and AG3 (full circle, bottom) when capturing a prey (square, center) and escaping a predator (double circle, top left).

the danger represented by the predator is passed, it can fulfill it. In this case, it is possible for AG1 to maintain a ‘virtual’ visual contact with the prey since *detect prey* contains a predictive component (the forward model) that continues to simulate the position of the prey even when the visual contact is lost. In order to do that, however, energetic resources have to be assigned to the schema even if it is temporarily not relevant for the external context (since there are no prey-like stimuli). The energizing influence of *hunger* produces covert behavior (instant 2), and later on overt effects (instant 3). The difference in behavior between AG3 and the other two agents depends instead on the fact that, since it is totally focused on detecting the prey, it detects the predator only at instant 2, when it is too late; and it is captured. Our study shows that qualitatively different responses to similar stimuli, chase, escape or freeze, can be obtained without three specialized schemas by varying the internal, motivational context. Here AG3 freezes because of the contradictory motor commands received from *chase* and *escape*. But the most important aspect is that the epistemic activity of AG3 is channeled by its motivations, and a maladaptive value of *hunger* makes AG3 unable to deal with the dangers in its environment [4].

4 Conclusions

We have proposed that motivations have a way of influencing behavior that can be called modulatory, since they do not trigger specific behavioral responses (as stimuli do) but influence indirectly the sensorimotor system by governing the allocation of its resources, and motivations determine which affordances in the environment are exploited. As suggested by [8], motivations have a *directing* and an *energizing* effect on behavior, but in our view the directing effect is a byproduct of the energizing one. We have then sketched a methodology for designing artificial agents that focuses on modulatory influences of motivations on behavior. In order to illustrate the effects of this design methodology, we have implemented an agent architecture which has a sensorimotor system, including perceptual and motor schemas, and a motivational system, including drives. We

have then analyzed its behavior in two studies. In study 1 we have reported the significant adaptive advantages of having a fully functioning motivational apparatus that interacts with the sensorimotor system and channelizes the agent's behavior. Motivations provide *values* to the agent: an agent acts (and learns to act) for satisfying its drives. The affordances in the environment are selected and used only if they are relevant for the agent's current needs. In study 2 we have highlighted two points. (1) Motivations can provide persistence (hysteresis). Thanks to its motivational state, AG1 is able to capture the prey even after losing perceptual contact with it. (2) Epistemic activity is channelized by motivations. The motivational state of AG3 do not only determine its behavior (going quickly toward the prey). Since its *detect prey* perceptual schemas have almost total control over the camera, only information relative to the prey is gathered and processed, and AG3 detects the predator only when it is too late.

References

1. M. Arbib. Schema theory. In S. Shapiro, editor, *Encyclopedia of Artificial Intelligence, 2nd Edition*, volume 2, pages 1427–1443. Wiley, 1992.
2. R. Arkin, K. Ali, A. Weitzenfeld, and F. Cervantes-Prez. Behavioral models of the praying mantis as a basis for robotic behavior. *Robotics and Autonomous Systems.*, 32(1):39–60, 2000.
3. O. Avila-Garcia and L. Canamero. Hormonal modulation of perception in motivation-based action selection architectures. In L. Canamero and D. Evans, editors, *Agents that Want and Like: Motivational and Emotional Roots of Cognition and Action. Workshop of the AISB05 Convention*, 2005.
4. A. Damasio. *Looking for Spinoza: Joy, Sorrow, and the Feeling Brain*. Harcourt, 2003.
5. S. E. Fahlman and C. Lebiere. The cascade-correlation learning architecture. pages 524–532, 1990.
6. B. Kosko. *Neural Networks and Fuzzy Systems*. Prentice Hall, 1992.
7. M. Minsky. The emotion machine. In preparation.
8. Y. Niv, D. Joel, and P. Dayan. A normative perspective on motivation. *Trends in Cognitive Science*, 8:375–381, 2006.
9. D. Parisi. Internal robotics. *Connection Science*, 16(14):325–338, 2004.
10. G. Pezzulo and G. Calvi. A schema based model of the praying mantis. In S. Nolfi et al., editors, *Proceedings of SAB 2006*, LNAI 4095, pages 211–223, 2006. Springer.
11. G. Pezzulo and G. Calvi. Designing modular architectures in the framework akira. *Multiagent and Grid Systems*, 3(1):65–86, 2007.
12. G. Pezzulo and G. Calvi. Schema-based design and the akira schema language: An overview. In M. Butz et al., editors, *Anticipatory Behavior in Adaptive Learning Systems: Advances in Anticipatory Processing*, LNAI 4520. Springer, 2007.
13. M. Snaith and O. Holland. An investigation of two mediation strategies suitable for behavioural control in animals and animats. In *Proceedings SAB 2006*, pages 255–262, 1990. MIT Press.
14. T. Tyrrell. *Computational Mechanisms for Action Selection*. PhD thesis, University of Edinburgh, 1993.
15. D. M. Wolpert and M. Kawato. Multiple paired forward and inverse models for motor control. *Neural Networks*, 11(7-8):1317–1329, 1998.