# Getting the Job Done in a Hostile Environment

Steven Versteeg and Alan Blair

Department of Computer Science
University of Melbourne
Parkville 3010 Australia
scv@cs.mu.oz.au
blair@cs.mu.oz.au

**Abstract.** In nature animals going about their daily routine need to avoid predators in order to survive. Many animals have evolved some kind of startle response, which enables them to escape from dangerous situations. As robots are required to operate in more hostile environments, mechanisms analogous to startle responses may be critical to building robust systems. This paper presents some preliminary work exploring (1) how some reactive evasive behaviours can be added to an agent operating in a hostile environment, and (2) how evasive measures can be integrated with the agent's other activities.

## 1 Introduction

Almost every animal is required to escape from predators from time to time in order to survive and reproduce. Animats may also need to escape from sources of danger, such as malevolent passers by and curious children [18]. The key is not only to have effective escape mechanisms but also to integrate escape with the animat's other activities. An animal that spends all its time running away without stopping to eat and find food will not survive very long.

A startle response for escaping quickly in critical situations is an adaptation which has arisen in many different animals from disparate parts of the evolutionary tree [7]. The startle responses vary greatly. In common they usually have simple reliable triggers, very fast activation and often produce a stereotyped response [3]. The fact that almost all animals have some kind of startle response suggest that there is an advantage to having a dedicated subsystem for detecting and responding to hazardous situations. Hoy [13] argues that startle responses will have a significant role in designing robust robot architectures.

A relatively 'simple' invertebrate animal that is particularly well studied by neuroscientists is the crayfish. (This is because they have relatively few neurons, some of the neurons are very large and crayfish make a delicious meal at the end of the experiment.) The crayfish is equipped with a variety of evasion techniques including (but not limited to): spending large amounts of time in hiding; retreating to safety if it notices a far away predator; and an escape response [19]. The escape response is a last resort mechanism that is used in extreme conditions. It is activated by a sharp tap to the abdomen or sudden visual stimuli. The

M. Brooks, D. Corbett, and M. Stumptner (Eds.): AI 2001, LNAI 2256, pp. 507–518, 2001.

response is a stereotyped tailflip that rapidly propels the crayfish away from the source of danger. The trigger for the escape reflex is controlled by giant command neurons in the abdomen. These neurons are responsible for integrating a large group of stimuli and making a snap decision. For a detailed reference to the neural organisation of the crayfish escape mechanism refer to [25,24,17,15, 16].

Some previous studies that have drawn inspiration from invertebrate neural circuits have yielded promising results. Beer [1] successfully modelled the neurons controlling insect walking with an R-C network. The circuit was fully distributed, efficient and robust; later it was used to control a real hexapod robot. There has also been work on modelling the escape response of the cockroach. [2]

There have been many previous studies of evasion in isolation. A couple of examples of evolving optimal strategies in scenarios with fixed predator behaviour include [14] and [12]. Miller and Cliff [18] co-evolved pursuer and evader tactics using noisy neural network controllers. The pursuer-evader problem has been reformulated as a one-dimensional, time-series prediction game. [10] There has been exploration of the evolution of evasion strategies when the game is made slightly asymmetric between the pursuer and evader. [23]

This paper presents some preliminary work in exploring how some simple reactive evasive behaviours can be added to an agent operating in a hostile environment. The mechanisms used are loosely inspired by the evasive tactics and reactions of the crayfish.

## 2 The Scenario

The scenario is a simple predator-prey simulation. There is one predator and one prey. The prey has the task of collecting enough food to survive while being hunted by a predator. The predator has a greater maximum velocity and a further seeing distance than the prey. The prey has superior acceleration over the predator and may choose to hide in a shelter where it is safe from the predator.

The environment is a continuous two-dimensional plane of $n \times m$ units. It has wraparound edges (this is to avoid the artifact of the prey being trapped in a corner.) The world contains pieces of food located at random locations. New pieces of food are added and old ones are removed at random time intervals. Also situated in the environment is a shelter. When the prey is in the shelter the predator is unable to see it and unable to kill it.

The predator and prey are able to move within the world and are able to make some limited interactions with the other entities in the world. The prey is able to eat a piece of food if it is close enough. The predator is able to kill the prey if it is close enough. The simulation is updated in discrete time steps. At each time step the predator and prey are queried by the simulation engine about their intended movements and other actions they want to take. All updates to positions and interactions are executed simultaneously. If the predator or prey elect to change their velocity, they are not able to do it instantly but instead

accelerate to new velocities. It may take several time steps for the predator or prey to reach their new velocity. The predator and prey each have maximum rates of acceleration.

The predator follows a very simple behaviour pattern. This behaviour pattern is fixed for all the experiments. The predator roams around at half-speed travelling in a straight line. At random time intervals it changes to a new random direction. The predator is continuously looking for the prey. If at any point the predator spots the prey it will immediately change its direction to head directly toward it and accelerate to its maximum velocity. When the predator gets within a distance of $k_{kill}$ units of the prey, the predator kills the prey and then eats its. The predator is present somewhere in the world for the entire duration of the simulation.

The prey has an internal energy level. To avoid starving it must maintain its energy level above zero. At each time interval the prey consumes an amount of energy determined by equation 1.

$$\Delta E = -(B + Av^2) \tag{1}$$

The base energy consumption (determined by $B$) forces the prey to occasionally go and collect food. The other term is dependent on the square of the prey's velocity to penalise travelling at high speeds. The prey replenishes its energy level by eating food. The prey needs to move close to a piece of food before it can eat it. When a piece of food is eaten the food is removed and the prey gains $k$ units of energy. If $E$ drops below zero the prey starves to death.

The simulation was written in Java. There are two interfaces: a graphical applet interface and a command line interface. The applet interface can be accessed on the web at `http://www.cs.mu.oz.au/~scv/botsim/`

## 3 Architecture of the Prey

The prey uses a layered architecture which we build up incrementally [20], [4]. The prey operates in distinct behavioural modes. It monitors the level of some simple stimuli, such as 'hunger', to determine which behavioural mode to operate in. At the most basic stage the prey ignores the predator completely and is solely focused on collecting enough food to avoid starving. At each stage, another behavioural mode or stimulus is added to the prey's repertoire to assist it in avoiding the predator. New behaviours are able to subsume or suppress behaviours introduced at previous levels.

### 3.1 The Hiding Bot

First we consider a very simple bot. The hiding bot looks for food when it is hungry and hides in the shelter when it is not. It does not detect an approaching predator. The hiding bot's survival strategy is basically to spend as much time in the shelter as possible, while avoiding starvation.

The hiding bot uses one stimulus with which to make its decisions: the internal energy level ($E$). It uses this information to choose between one of two behavioural modes in which it can operate:

**Forage.** The prey searches for food and eats it. The prey follows the odour gradient emitted by the food until it reaches an item of food which it then eats. The prey travels at half speed to conserve energy.

**Hide.** The prey moves to the shelter and hides there when it arrives. The prey travels at half speed to conserve energy.

Figure 1a shows the hiding bot's control architecture.

### 3.2 The Running Away Bot

The running away bot actively keeps an eye out for the predator while it is outside of the shelter. The bot operates in the same way as the hiding bot in that it ventures out of the shelter for food when it is hungry, but if while the bot is out of the shelter it detects the predator it will scurry back to the shelter for safety.

The running away bot may use the behavioural modes of the hiding bot: hide and forage, and in addition may use: *run away* mode.

**Run Away.** The prey runs to the shelter at maximum velocity.

To determine when to run away, the bot uses the stimulus *predator fear*. It is dependent on the distance ($d_{pred}$) between the prey and the predator as shown in equation 2.

$$P = A_P e^{-\frac{d_{pred}}{L}} \tag{2}$$

The algorithm uses to control the running away bot is shown in figure 1b.

### 3.3 The Memory Bot

The hiding bot and the running away bot are stateless. The memory bot explores what advantage a simple piece of state information can give an agent. [6] demonstrated that adding even a few memory bits can give a significant improvement to the performance of a reactive object tracking system.

The memory bot remembers when it last saw the predator. This affects the stimulus *memory fear*. When the predator is seen memory fear instantly rises to the maximum. It decays exponentially with time ($t_{pred}$) from when the predator was last seen as shown in equation 3

$$M = A_M e^{-\frac{t_{pred}}{\tau}} \tag{3}$$

The memory bot operates in the same way as the running away bot but if its memory fear is still above a threshold $T_M$ then it will continue to hide. Figure 1c shows the control algorithm.

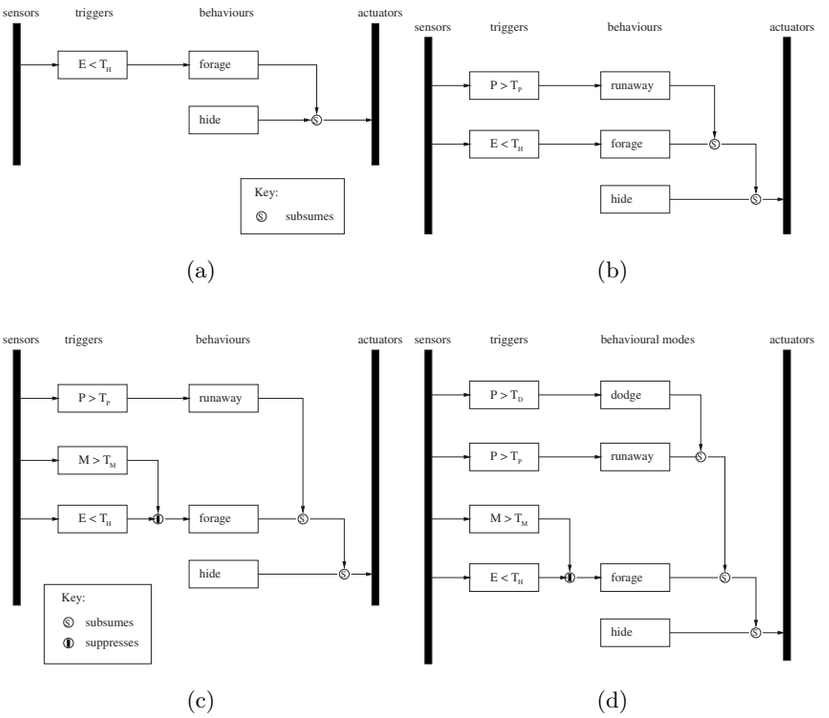


**Fig. 1.** The architecture of the prey. (a) Hiding bot. By default the bot hides; if the bot is hungry, the forage behaviour takes control of the actuators. (b) The running away bot. *Run away* takes control of the actuators if the predator is seen. (c) The memory bot. If the predator was seen recently the forage behaviour is suppressed. (d) The dodging bot. Dodge takes precedence over all other behaviours.

### 3.4 The Dodging Bot

The dodging bot has a reflex action with which it attempts to evade the predator if it gets too close.

If the predator fear stimulus crosses a threshold $T_D$ then the prey will go into *dodge* mode:

**Dodge.** The prey immediately changes to a new direction which is orthogonal to the direction of the approaching predator. The prey very rapidly accelerates to maximum velocity. (It is in effect a jump to the side.)

This manœuvre is somewhat analogous to the escape reflex in the crayfish in that (1) a simple test is used to determine when to activate the response, and (2) to be effective a special piece of hardware is needed. The crayfish makes use of special flexors in the abdomen which are only used in an escape tailflip; this gives it extremely rapid acceleration. The dodging bot uses a higher amount of acceleration in the dodge manœuvre than it normally has available to it. The dodge manœuvre is also somewhat similar to the zig-zagging used by the evasive agents in the [18] simulations.

The dodge behaviour takes precedence over all other behaviours. Figure 1d shows the control layer diagram of the dodging bot.

## 4 Results

The four bots were placed in the environment to see how well they survived. Different configurations were tried for each bot by varying the thresholds used to make the decisions.

Each configuration was tested a thousand times. For each configuration the following statistics were recorded:

1. The mean number of time steps that the bot survived.
2. The median number of time steps that the bot survived.
3. The standard deviation of the survival time.
4. A tally of the causes of death, i.e., on how many runs the bot ran out of energy (*starved*), on how many runs the bot was caught by the predator (*killed*) and on how many runs the bot reached the end of the simulation still alive (*survived*).

Refer to Appendix A for the values of constants used in the simulation.

### 4.1 The Hiding Bot

Since the hiding bot is unable to detect the predator its behaviour is governed only by the hunger threshold, which determines when it will hide in the shelter and when it will venture out to look for food. Figure 2a shows how the survival time and cause of death vary as the hunger threshold ($T_H$) changes. If the hunger threshold is very low, then the bot will wait until it is almost completely out of energy before venturing out to look for food. There is a very high chance that the bot will starve to death before finding the food. If the hunger threshold is very high, then the bot will spend most of its time out of the shelter looking for food and there is a greater chance of it being eaten by the predator. The bots that do the best are the ones that stay in the shelter as long as possible while still having a good chance of finding food in time before starving.

Figure 2a shows how the survival rate of the hiding bot varies by adjusting the hunger threshold. A curious feature of the graph in figure 2a is that as the hunger threshold goes from 0 to about 2, the median declines slightly but the mean rises. The explanation of this phenomenon is at very low hunger thresholds

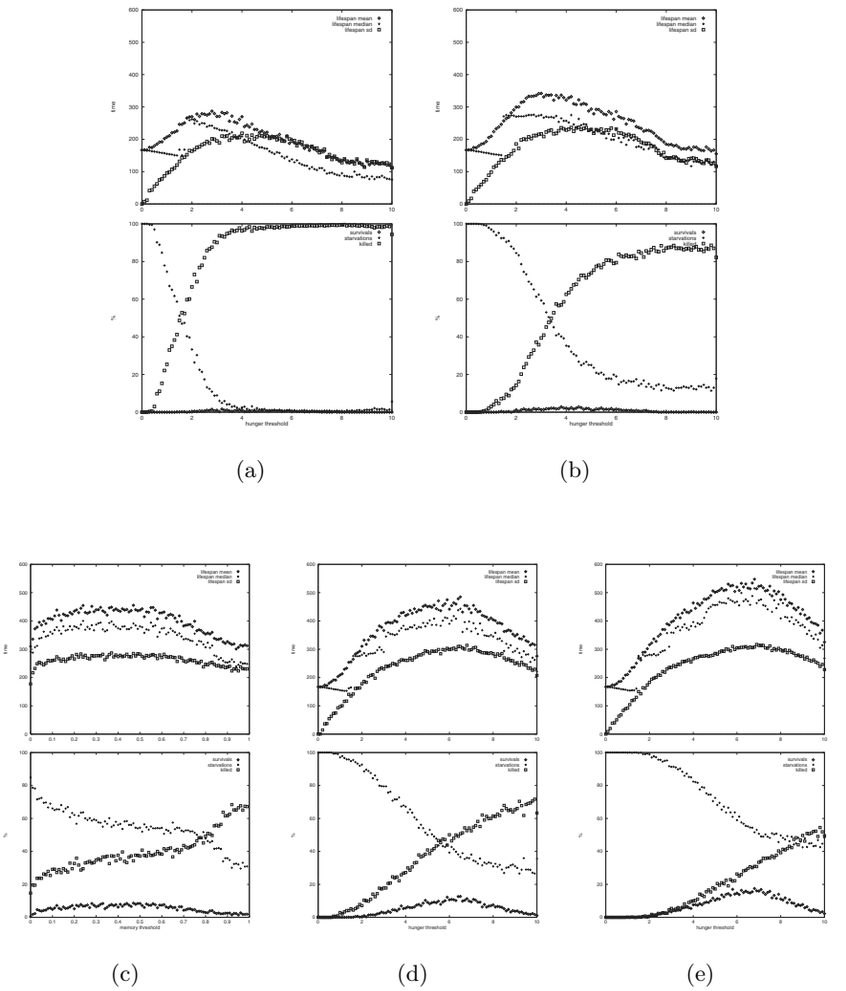


**Fig. 2.** Plots of the survival times and the causes of death versus the hunger threshold for each bot: (a) the hiding bot, (b) the running bot, (d) the memory bot and (e) the dodging bot. (c) The survival times and cause of death versus memory threshold for the memory bot ($T_H = 6$).

when the bot ventures out it is more likely that the bot will starve than that it will find food. While most of the bots die of this cause the median will stay

low. However the lucky few who find food live significantly longer so push up the mean.

### 4.2 The Running Bot

The running bot survives significantly better than the hiding bot as can be seen in figure 2b.

After seeing the predator the running bot is generally effective at running back to the shelter in time before the predator overhauls it. Provided that the hunger threshold is set at a reasonable level the death of the running bot generally occurs under one of three circumstances: (1) the bot is too far away from the shelter when the predator sees it and is therefore rundown by the predator in pursuit, (2) the predator approaches so that it is between the bot and the shelter; in this case the bot is cut off while running to safety, and (3) the bot is chased before it reaches any food so it is even shorter of energy when it gets back to the shelter; this causes the bot to die of exhaustion rather than being killed by the predator directly. Furthermore, the running bot suffers from having no memory and a shorter seeing distance than the predator. If the bot is chased to the shelter and is still hungry it will venture out as soon as it can no longer see the predator. But it is very likely that the bot will be still within the predator's seeing distance and therefore is immediately chased again.

### 4.3 The Memory Bot

The simple state information provided by the memory fear stimulus gives the memory bot a big boost in survival chances. The memory bot is able to run back to the shelter and stay there long enough until the predator has passed. Since the predator moves randomly, the longer it stays in the shelter the greater the chance that the predator will have passed. Balanced against this is that staying longer in the shelter reduces the chance of having enough energy to reach new food before it starves. The bot will stay in the shelter until the memory fear (see equation 3) drops below the threshold $T_M$. If the bot has a very high threshold it will stay in the shelter only a short time. If the bot has a very low threshold it will stay in the shelter a very long time. Figure 2c shows how the survival rate is affected by the value chosen for the memory threshold. Note that the survival rate is fairly even for thresholds between 0.2 and 0.6, reflecting that the tradeoff between waiting out the predator and risk of starvation is fairly evenly balanced.

Figure 2d shows how the survival rate is affected by the hunger threshold for the memory bot. The optimal hunger threshold is greater for the memory bot than for the running bot. This can be explained by two factors: (1) the memory bot has got a better chance of surviving an encounter with the predator so it can risk spending more time out of the shelter and (2) if the memory bot collects more energy then it can spend more time in the shelter after it has been chased by a predator, giving it higher chance of the predator having left.

The memory bot is still killed if it is too far away from the shelter when pursuit begins or if the predator is in between it and the shelter. It is fairly

successful in waiting in the shelter until the predator has passed. It sometimes starves while waiting for the predator to pass and sometimes is unlucky and finds the predator still waiting outside after it has waited.

### 4.4 The Dodging Bot

The dodge manœuvre of the dodging bot allows it to survive in one of the situations where the memory bot is frequently killed. If the predator attacks the prey coming from the direction of the shelter, the dodging bot is able to evasively side-step the predator. The predator coming at full speed is unable to adjust its direction in time to catch the prey. The dodge manœuvre is less successful in evading the predator chasing from behind. However as with the running bot and the memory bot, in most cases when the prey is being chased from behind it will be able to reach the shelter in time. It is usually only when the prey is very far from the shelter that it is caught in pursuit.

Figure 2e shows how the survival rate of the dodging bot varies with the hunger threshold. The dodging bot has a higher optimal hunger threshold again compared to the memory bot.

## 5 Discussion and Conclusions

A trend that emerges is the prey spends more time outside of the shelter foraging for food as it gets better at avoiding the predator. This phenomenon may be caused by two factors: (1) for the evasion tactics to be advantageous the prey needs enough energy to do the evasive manœuvres and still collect food, and (2) because the prey is more likely to survive an encounter with the predator it can afford to spend more time outside the shelter foraging for food, thereby reducing its risk of starvation.

To optimise its survival time the prey needs to balance its primary task, collecting food, with taking evasive action. If the predator were nonexistent the optimal survival strategy for the prey would be to spend all its time collecting food to nullify any risk of starvation. In the presence of a predator, collecting food becomes a risky task. This causes the bots with poor predator avoidance to wait until they are really hungry before they will venture out to look for food. As the prey is equipped with more evasive capabilities the risk in collecting food diminishes. This increased confidence allows the prey to act more like it would if there were no predator. As more evasive capabilities are added, the prey's behaviour pattern (when it is not taking evasive action) approaches what it would be if the predator did not exist.

A more general implication of this result is that having separate subsystems to deal with dangerous situations allows an agent to be less obstructed in undertaking its primary activities. Robots presumably have a set of primary tasks which they have to perform. However some robots while undertaking their work robots may periodically have to face obstructions or even dangers. A hypothetical example is a rescue robot sent into a burning building that may have

to dodge falling debris. Having separate subsystems to deal with obstructions may allow robots to be minimally affected in the way they achieve their primary tasks. Animals have dedicated neural circuits to deal with unexpected hazardous situations. [3,13]

One of the biggest improvements given to the prey in this simulation is the addition of some simple memory information. In this memory model the time that the predator was last seen is used implicitly as a predictor of the predator's present proximity.

The fixed hierarchical model used for the agents seems suboptimal for this scenario. One kind of behaviour should not always have precedence over another. For example because hiding in a shelter after seeing a predator has a higher precedence than foraging, in some cases the bot would starve to death in the shelter. A better model would be more flexible: wait longer in the shelter if energy reserves are relatively high, and shorter if energy reserves are low. Different actions need different precedence at different times.

There is biological evidence that in animals the precedence of actions is much more flexible. [21] review biological findings about parts of the vertebrate brain and argue that the basal ganglia acts as a central decision making point for arbitrating between conflicting actions. They argue that a similar specialised switching mechanisms might be employed in layered robot architectures (such as [4]) to provide more flexible action selection.

In the crayfish the giant command neurons responsible for triggering the escape response are modulated by other parts of the nervous system [22,11,9]. The trigger threshold adjusts according to various circumstances, such as during feeding and restraint [24] and also adjusts according to longer term conditions such as the mating cycle and social dominance [26].

Edwards [8] proposes a model for behavioural choice in crayfish that uses mutual inhibition amongst the neural command centres. In Edwards' model there is one command neuron for seven different behavioural modes. Each neuron receives excitatory stimuli from sensors. Each neuron is able to inhibit other command neurons and also receives inhibitory signals from the excited command neurons. After summing the excitations and the inhibitions, the command neuron with the greatest excitation wins. Edwards' model is able to give actions different precedence at different times. An attempt was made to write a bot based on Edwards' architecture for the scenario described in this paper. Preliminary results indicate that in this scenario it performs slightly better, but the results are inconclusive.

The scenario examined in this paper is very specific. Previous pursuit-evasion experiments [5] have shown that effective evasion strategies are often very sensitive to the parameters of the environment. Future work may consider what kind of escape measures work best when faced with different kinds and variable sources of danger. The prey currently uses 'magic' perception to get the position of the predator. This is unrealistic. In future simulations the prey will have to infer the presence of a predator from noisy sensors.

The work presented in this paper is a preliminary step in exploring the role of integrating evasive actions in the context of doing other activities. As robots move out of the laboratory into more hostile environments, handling evasive actions may prove an important component of the robot architecture.

## A Constants Used in Simulation

| Constant | Value | Equations used |
|---|---|---|
| $A$ | 0.001 | (1) |
| $B$ | 0.03 | (1) |
| $A_P$ | 1 | (2) |
| $L$ | 40 | (2) |
| $A_M$ | 1 | (3) |
| $\tau$ | 15 | (3) |

## References


1. Randall D. Beer. Intelligence as adaptive behaviour: An experiment in computational neuroethology. In *Perspectives in Artificial Intelligence*, volume 6. Academic Press, 1990.
2. Randall D. Beer and Hillel J. Chiel. Simulations of cockroach locomotion and escape. In *Biological Neural Networks in Invertebrate Neuroethology and Robotics*, pages 267–285. Academic Press, San Diago, 1993.
3. Michael V. L. Bennett. Escapism: Some startling revelations. In *Neural Mechanisms of Startle Behavior*, pages 353–363. New York: Plenum Press, 1984. edited by R. C. Eaton.
4. Rodney A. Brooks. A robust layered control system for a mobile robot. *IEEE Journal of Robotics and Automation*, RA-2(1):14–23, Marth 1986.
5. Dave Cliff and Geoffrey F. Miller. Co-evolution of pursuit and evasion ii: Simulation methods and results. In *Animals to Animats IV: Proceedings of Fourth International Conference on Simulation of Adaptive Behaviour*, pages 506–515. MIT Press, 1996.
6. M Dorigo and M Colombetti. Robot shaping: developing autonomous agents through learning. *Artifical Intelligence*, 71:321–370, 1994.
7. R. C. Eaton, editor. *Neural mechanisms of startle behaviour*. Plenum Press, New York, 1984.
8. Donald H. Edwards. Mutual inhibition among neural command systems as a possible mechanism for behavioral choice in crayfish. *Journal of Neuroscience*, 11(5):1210–1223, May 1991.
9. Donald H. Edwards, William J. Heitler, Esther M. Leise, and Russell A. Fricke. Postsynaptic modulation of rectifying synaptic inputs to the lg escape command neuron in crayfish. *Journal of Neuroscience*, 11(7):2117–2129, July 1991.
10. Sevan G. Ficici and Jordan B. Pollack. Coevolving communinative behavior in a linear pursuer-evasion game. In *Animals to Animats V: Proceedings of Fifth International Conference on Simulation of Adaptive Behaviour*, pages 557–561. MIT Press, 1998.

11. David L. Glanzman and Franklin B. Krasne. Serotonin and octopamine have opposite modulatory effects on the crayfish's lateral giant escape reaction. *Journal of Neuroscience*, 3(11):2263–2269, November 1983.
12. J. J. Grefenstette, C. L. Ramsey, and A. C. Schultz. Learning sequential decision rules using simulation models and competition. *Machine Learning*, 5(4):355–391, 1990.
13. Ronald R. Hoy. Acoustic startle: an adaptive behaviour act in flying insects. In *Biological neural networks in invertebrate neuroethology and robotics*, pages 139–158. Academic Press, San Diago, 1993.
14. J. Koza. Evolution and co-evolution of computer programs to control independently-acting agents. In *Animals to Animats: Proceedings of First International Conference on Simulation of Adaptive Behaviour (SAB92)*, pages 366–375. MIT Press, 1991.
15. Franklin B. Krasne and Sunhee Cho Lee. Response-dedicated trigger neurons as control points for behavioural actions: selective inhibition of lateral giant command neurons during feeding in crayfish. *Journal of Neuroscience*, 8(10):3703–3712, October 1988.
16. Franklin B. Krasne and Terri M. Teshiba. Habituation of an invertebrate escape reflex due to modulation by higher centers rather than local events. In *Proceedings of the National Academy of Science, USA*, volume 92, pages 3362–3366, 1995.
17. Franklin B. Krasne and Jeffrey J. Wine. The production of crayfish tailflip escape response. In *Neural Mechanisms of Startle Behaviour*, pages 179–211. New York: Plenum Press, 1984. edited by R. C. Eaton.
18. Geoffrey F. Miller and Dave Cliff. Protean behaviour in dynamic games: co-evolution of pursuit-evasion tactics. *From Animals to Animats 3*, pages 411–420, 1994.
19. P. Olszewski. *A Salute to the humble yabby*. London Angus & Robertson, 1980.
20. Rolf Pfeifer and Christian Scheier. *Understanding Intelligence*. MIT Press, 1999.
21. Tony J. Prescott, Peter Redgrave, and Kevin Gurney. Layered control architectures in robots and vertebrates. *Adaptive Behaviour*, 7(1):99–127, 1999.
22. Eric T. Vu and Franklin B. Krasne. Crayfish tonic inhibition: Prolonged modulation of behavioural excitability by classical gabaergic inhibition. *Journal of Neuroscience*, 13(10):4394–4402, October 1993.
23. Mattias Wahde and Mats G. Nordahl. Evolution of protean behavior in pursuit-evasion contests. In *Animals to Animats V: Proceedings of Fifth International Conference on Simulation of Adaptive Behaviour*, pages 557–561. MIT Press, 1998.
24. Jeffrey J. Wine. The structural basis of an innate behavioural pattern. *Journal of Experimental Biology*, 112:283–319, 1984.
25. Jeffrey J Wine and Franlin B Krasne. The cellular organisation of crayfish escape behaviour. In *The Biology of Crustacea*, volume 4, pages 241–292. Academic Press, 1982.
26. S-R Yeh, B. E. Musolf, and D. H. Edwards. Neuronal adaptations to changes in the social dominance status of crayfish. *Journal of Neuroscience*, 17(2):697–708, January 1997.