

An Adaptive Neuro-Endocrine System for Robotic Systems

Jon Timmis*[†], Mark Neal[‡] and James Thorniley*

*Department of Electronics, University of York, Heslington, York, UK

Email: {jt517}@ohm.york.ac.uk

[†]Department of Computer Science, University of York, Heslington, York, UK

[‡]Department of Computer Science, University of Aberystwyth, Aberystwyth, Wales, UK

Abstract—We present an adaptive artificial neural-endocrine (AANE) system that is capable of learning “on-line” and exploits environmental data to allow for adaptive behaviour to be demonstrated. Our AANE is capable of learning associations between sensor data and actions, and affords systems the ability to cope with sensor degradation and failure. We have tested our system in real robotic units and demonstrate adaptive behaviour over prolonged periods of time. This work is another step towards creating a robotic control system that affords “homeostasis” for prolonged autonomy.

I. INTRODUCTION

Homeostasis in biology is the quality of living cells or organisms that allows them to maintain a stable internal state [1]. Work in [2] argues that a system of internal state regulation modelled on homeostasis can lead to robot autonomy. They describe three primary systems in the human body that control homeostasis: the nervous system, the endocrine system and the immune system. These systems are capable of controlling or influencing organs of the body as well as each other. Work in [2] argue that by capitalising on the interactions of these systems, it should be possible to design architectures of interacting neural and endocrine systems that afford significant long term autonomy. There has been a limited amount of work investigating the interaction of the neural and endocrine system using the architecture presented in [2]. For example [3] develop the ideas further in terms of biological inspiration and apply the architecture to a simple seeking problem, and work in [4] extend the ideas of gland cells by introducing a “pool and release” mechanism for the glands. However, no previous work in this area has examined the idea of introducing adaptivity into the neural-endocrine system. To date, once a neural-endocrine system had been trained, then once on-line, the system could not adapt to new situations and make acquire new knowledge. This presents a problem for the goal of long-term autonomy when a system will be required to adapt over its operational life time. This paper presents the first studies into creating an adaptive neural-endocrine system. Our system is capable of learning associations between sensor data and actions during the operational life-time of the system. Based on a simple Hebbian approach, we show how it is possible to introduce feedback into the endocrine system that can affect neural behavior. This paper is structured as follows. In section II we provide background information on the neural-

endocrine system and previous work in the area, in section III we present our architecture for adaptive neural-endocrine system and some initial results, in section IV we present a slightly modified version of our system with further analysis and results, and we conclude in section V

II. BACKGROUND MATERIAL

Whilst much recent work in robotics has focussed on various “behaviour-based” architectures and methods [5], the long-running parallel thread of biologically inspired control systems has continued to develop. Where behaviour-based systems use carefully engineered interactions and suppressions to allow the more complex behaviours to emerge, many biologically inspired techniques attempt to evolve and/or learn to generate these behaviours. The vision of the work in this paper falls squarely into this latter category and lays the foundations for genuinely self-organizing autonomous systems.

A. Artificial Endocrine Systems

The biological inspiration for the artificial endocrine system, particularly in the context of artificial homeostasis, is given in [3]. In their initial work, [2] the authors establish a method by which a hormone can be used to regulate behaviour, even though it is a particularly simple example. The endocrine system is modeled by adding a gland cell to an ANN system. The gland generates a time-varying hormone concentration in the system based on stimulation from sensory inputs and an exponential decay. The effect of the hormone binding to the neural cells is modeled by multiplying the weighted inputs of each neuron by a value based on hormone concentration, sensitivity of the synapse to the hormone, and a match term between the synapse and the hormone.

Simple object avoidance rules were trained into an ANN on a robot with 16 proximity sensors. The authors observe that when the machine was placed in a highly populated environment the ANN would sometimes fail and the robot would bump into walls and obstacles. Implementing a more cautious algorithm is argued to be an inadequate solution as the robot would then be inefficient when working in less cluttered environments.

Results show that the hormone applies a level of cautiousness to the ANN, and that the time-varying hormonal system makes closer approaches to obstacles than the overly-cautious

constant hormone system, but retreats much faster than the pure ANN system.

What is limiting in this approach is the ability of the system to adapt over time. Once the neural-endocrine system has been placed “on-line” there is no adaptation at all. This presents a problem for building systems that are supposed to have long-term autonomous behaviour. Ideally, one would have a system that can learn “on-line” and adapt to new situations and data. To this end, the rest of this paper is concerned with developing such an architecture.

III. ARCHITECTURE DESIGN AND DEVELOPMENT

In order to allow the system to learn “on-line”, our proposed architecture uses two signals and attempts to associate them online. The first signal, the “collision signal”, is active whenever the robot bumps into a wall or object, and will always activate an object-avoidance behaviour. The second signal is the “proximity signal”, and initially is activated only by small, random amounts when the proximity sensors provide input data. As the robot explores the environment, it should learn to associate proximity signals and collision signals (since they are likely to occur at the same time), and thus to activate the avoidance behaviour from stimulation of the proximity signal.

A. An Artificial Neuroendocrine System

As previously discussed, a neuroendocrine system will be used here to regulate the behaviour of an artificial neural network.

Using [2] as a starting point, the first component we include in our system is an endocrine gland. This controls the production and “secretion” of a particular hormone into the system. By this we mean that the gland maintains a hormone concentration that influences the behaviour of the neural network. The hormone concentration at a discrete time point, $c_g(t)$, is given by Equation 1.

$$c_g(t) = \beta_g c_g(t-1) + R_g(t) \quad (1)$$

Here β_g is a decay constant, and R_g is a stimulation value, the calculation of which will be discussed later. The time t values used here are assumed to be discrete, integer values, where the value of t increases by one for each processing step.

1) *Neuron Dynamics*: The output of the neuron is the weighted sum of the input values passed through a threshold function. In the general case, the activation u (before the application of the threshold function) is given by Equation 2, where w_i is a weight and x_i is an input value (note that w_0 is generally used as a *bias* value and x_0 is set to -1 in such cases).

$$u = \sum_i w_i x_i \quad (2)$$

[2] now introduces the idea of a “membrane receptor” on each neuron. This models the ability of hormones to stimulate or inhibit neurons. Equation 3 shows the new activation calculation, which is dependent on the hormone concentration

and a “sensitivity” term¹. Not every hormone in the system must affect every neuron. In fact, as will be shown below, this architecture “pairs up” two sets of one gland and one neural network.

$$u = \sum_i w_i x_i \sum_g c_g S_{gi} \quad (3)$$

$g \in$ the set of glands connected or “matched” to this neuron

S_{gi} = the sensitivity of synapse i to hormone/gland g

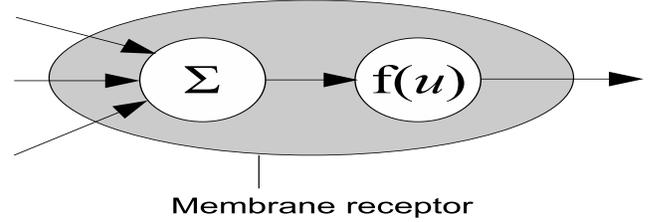


Fig. 1. Artificial neuron with membrane receptor

B. Architecture Overview

A high-level block diagram of a simple neuroendocrine architecture is shown in Figure 2. We have two behaviours (wander and avoid) and there is one gland, or “desire”, associated with each.

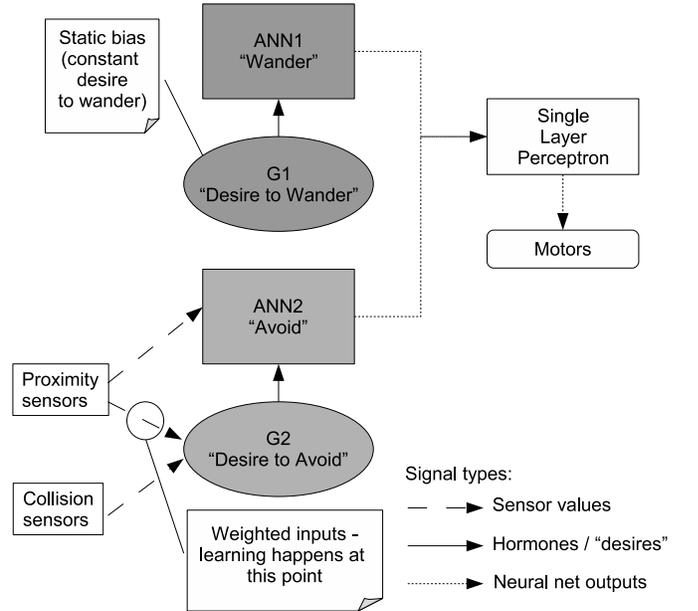


Fig. 2. Basic behavioural architecture

The two behaviours are implemented by artificial neural networks. The ANNs are multi-layer perceptrons whose weights have been chosen by manual design. The first network “wanders”, i.e. is simply biased to move the robot forward

¹In [2] a “match” term is also suggested, but this will not be incorporated into this architecture, except to say that each gland is either “connected” to a network or it is not – this could be viewed as a binary 1 or 0 match term.

constantly. The second network avoids obstacles by using input from the proximity sensors to decide on a sensible direction to turn. There are two glands, one associated with each network. The synapses in each network are sensitive to the associated gland (or if you prefer, the hormone that that gland secretes) with a sensitivity value of 1 in all cases. The concentrations of each hormone thus regulate the extent to which the associated behaviour dominates. The hormone concentrations lend themselves to being interpreted as “desires”, for example “desire to wander” or “desire to avoid”.

The “single layer perceptron” simply combines the outputs of the two ANNs and limits them to the range $-1/+1$, and uses this to control the motor speeds.

The architecture shown can be implemented using the neuron models given above and the standard feedforward network design, but the one area that remains undefined is how the stimulation of the glands (R_g) is calculated. In the case of the “wander” gland, the hormone concentration was initially set to a constant value of 1, so that the robot can be thought to have a static desire to wander. The “avoid” gland is to be made adaptive, and thus the calculation of R_g here is more complex.

1) *Gland Stimulation*: We define two vectors representing input groups, \vec{x}_1 for the collision inputs, and \vec{x}_2 for the proximity inputs. Elements in the vectors are the numeric outputs of the sensors (in all cases, limited to the range 0.0 to 1.0). Now consider Equation 4, which gives us a signal value A_i through the dot product of an input vector \vec{x}_i and a weight vector \vec{w}_i ². This formula can be used to obtain two signal values, A_1 for collision and A_2 for proximity, that will later be used in calculating gland stimulation. Note that hereafter, the two sets of input values (collision and proximity) will be subscripted with an i , and the individual sensors within those sets will be subscripted with j where necessary, such that x_{ij} is the output value of the j th sensor in the i th input set.

$$A_i(t) = \vec{w}_i(t) \cdot \vec{x}_i(t) \quad (4)$$

Or, equivalently:

$$A_i(t) = \sum_j w_{ij}(t)x_{ij}(t) \quad (5)$$

The collision signal, which has already been defined as a “hard-wired” signal, can be generated with a fixed weight vector. If we define the collision weight vector \vec{w}_1 to have an equal number of elements to the sensor vector (\vec{x}_1), all with unity value, then Equation 4 simplifies to Equation 6 – it becomes a summation.

$$A_1(t) = \sum_j x_{1j}(t) \quad (6)$$

²This weight vector is entirely unrelated to the weight values w_i used in the neuron activation equations (2 and 3). The symbol has been re-used as there is no intention to revisit the neuron dynamics and it is convenient to think of these new vectors as weights also.

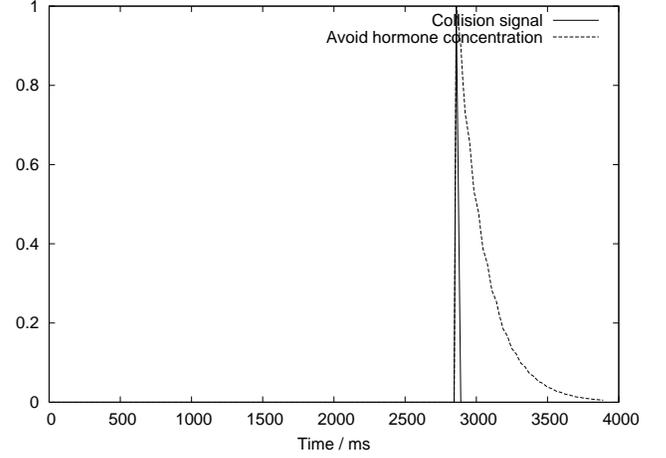


Fig. 3. “Desire to avoid” hormone concentration values in response to collision signal

In the case of the proximity signal, the inclusion of a weight vector provides a useful point to insert adaptivity. By intelligently updating the weight vector on the basis of previous inputs the robot should be able to learn useful information about the nature of proximity sensor values. Thus the vector \vec{w}_2 will be a set of values that may vary during the course of the experiment, precisely how this happens will be covered below.

The simple summation in Equation 7 provides a method for calculating R_g (the “stimulation” value passed to the gland in Equation 1). Here α_g is simply a constant stimulation rate.

$$R_g(t) = \alpha_g \sum_i A_i(t) \quad (7)$$

We can now see in Figure 3 the dynamics of the hormone produced by the “desire to avoid” gland. In this example, run in the simulator, only the collision signal A_1 was used in calculating R_g . The hormone is secreted rapidly in response to the collision signal, then decays exponentially.

2) *Adaptive Stimulation*: The adaptive part of the gland stimulation comes from a learning algorithm applied to the weight vector that is used in the proximity signal. The j th weight in the vector \vec{w}_2 will be updated in every processing step according to Equation 8.

$$w_{2j}(t+1) = w_{2j}(t) + \Delta w_{2j}(t) \quad (8)$$

It is now possible to vary the learning method by defining Δw_{2j} . The intention is to learn an association between the collision signal and the proximity signal, thus the first technique explored was to increase the weights when the collision signal, A_1 , and the proximity signal due to the input of interest (i.e. $w_{2j}x_{2j}$) were activated together.

The learning rule in Equation 9, uses both the collision signal and the proximity signal to mediate the association. The aim is that the proximity signal will be self-reinforcing, so that as long as the proximity signal is being stimulated the

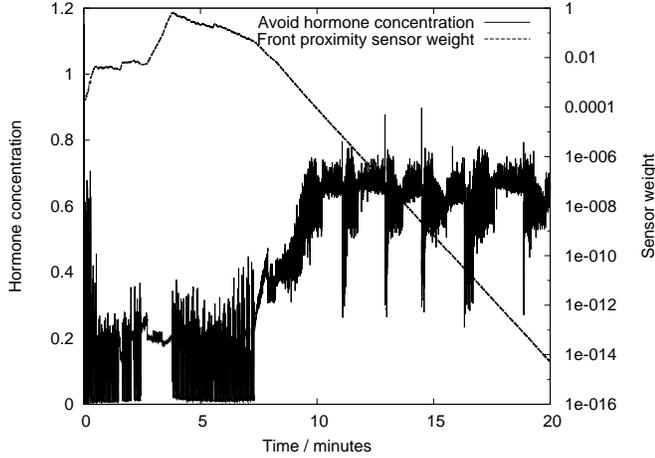


Fig. 7. Hormone concentration and sensor weight in a simulated robot using the neuroendocrine plus feedback architecture

concentration of the main hormone, and has an exponential decay as per the established gland dynamics. For clarity, Equation 11 shows the complete formula for the concentration of feedback hormone. Table I summarises the equations that will be used from this point on to calculate the hormone concentrations in the two glands.

$$R_g(t) = \frac{\alpha_g}{1 + c_f(t-1)} \sum_i A_i(t) \quad (10)$$

$$c_f(t) = \beta_f c_f(t-1) + \alpha_f c_g(t) \quad (11)$$

This should limit the rapid increase in the hormone concentration, but recall that we also have a learning system, where the weight update is given by Equation 9. This rule is dependent on the input signals and the current weight values, neither of which is properly limited. However, if we replace the input signals in Equation 9 with the stimulation R_g (Equation 12), the feedback mechanism is now incorporated in the learning rule, and can help to keep the weights within bounds.

$$\Delta w_{2j} = \eta w_{2j} x_{2j} R_g - \epsilon w_{2j} \quad (12)$$

Figure 7 shows some results for this controller in simulation. For the first few minutes, the controller works well. It gets stuck at one point (around 3 minutes in), but escapes without causing an uncontrolled increase in either hormone concentration or sensor weight. However, later on it becomes trapped and unable to escape. The sensor weights still decrease exponentially, and it appears they will never return to “normal” levels.

At this point it was decided that the learning rule should be modified again. In Equation 13, the weight is removed from the first term of the Δw calculation. This should prevent a low weight from inhibiting itself from learning. Also, this final equation allows us to think of the system in terms of Hebbian learning: consider a simple system consisting of two artificial neurons and a weighted synapse between them. The

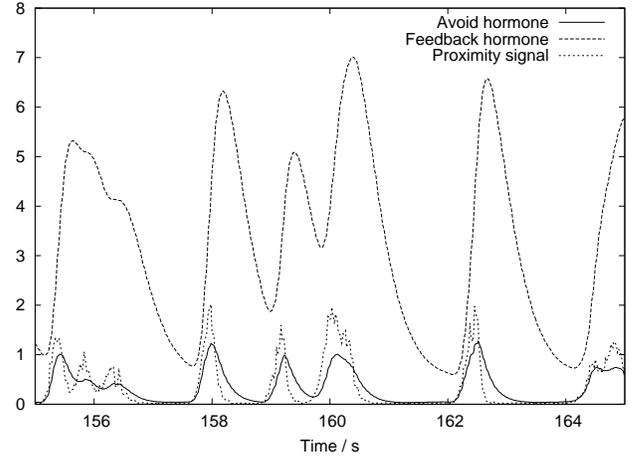


Fig. 8. Interaction of primary and feedback hormones in a short time period

pre-synaptic neuron is a proximity sensor, its output being x_{2j} . The post-synaptic neuron stimulates a gland with its output value R_g . This new learning mechanism simply increases the weighting of the synapse when both neurons are active at the same time.

$$\Delta w_{2j} = \eta x_{2j} R_g - \epsilon w_{2j} \quad (13)$$

In simulations using this learning rule it was found that if the robot got stuck, the weight stayed constant indefinitely - which means that the robot controller will at least be able to recover with some external help (e.g. repositioning). To prevent the robot from getting stuck at all, it was necessary to redesign the avoidance neural network a little. The robot was then able to run in simulation for over an hour without any problems. Figure 8 shows the hormone concentrations from a short period in this experiment. Notice how the “desire to avoid” hormone is prevented from rising to very high levels. Consider the peak at 160s, the hormone concentration stops rising once the feedback hormone has built up, even though the proximity signal is still present.

IV. FURTHER TESTING OF ARCHITECTURE

This section presents a further improvement to the architecture, based on a value-dependent learning scheme is proposed and compared to the initial results.

A. Experiment set-up

The environment for all the experiments was a rectangular pen 60cm by 80cm with wooden barriers as walls. The pen is filled with eight cylindrical obstacles of about 10cm diameter.

1) *Control experiment:* Three 45 minute runs of the control experiment were conducted. In each case, the robot was observed to collide with walls and obstacles several times in the first few minutes of the run. Later, the system would generally avoid obstructions from a distance of three to five centimetres, though there would still be the occasional collision. The system appears to demonstrate the associative learning behaviour that was intended.

Gland type	Stimulation	Hormone concentration
Generic gland	$R_g(t) = \frac{\alpha_g}{1+c_f(t-1)} \sum_i A_i(t)$	$c_g(t) = \beta_g c_g(t-1) + R_g(t)$
Feedback gland	$R_f(t) = \alpha_f c_g(t)$	$c_f(t) = \beta_f c_f(t-1) + R_f(t)$

TABLE I
GLAND DYNAMICS INCORPORATING FEEDBACK

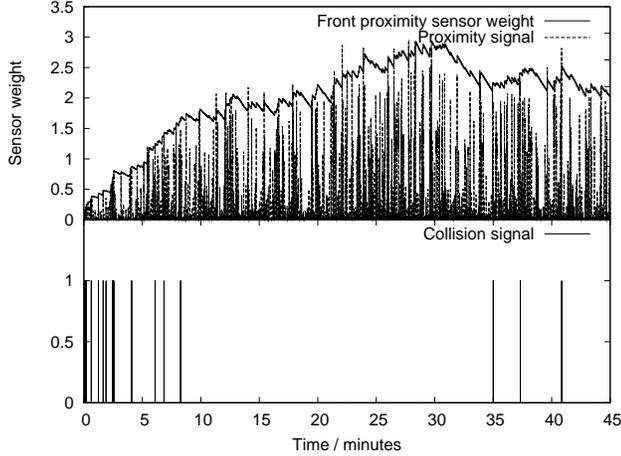


Fig. 9. Measurements from a typical control experiment (with no artificial errors)

Some significant variables from one of the control experiments are plotted in Figure 9. The top graph shows one of the weight values (in this case the value associated with sensor number 3, one of the forward sensors), and the overall proximity signal. The weight increases rapidly to begin with, mainly due to the collisions (bottom graph), and then levels off at an equilibrium as predicted. The proximity signal rises and falls rapidly as the robot approaches the walls and then avoids them, but the overall envelope rises in proportion to the sensor weight. This correlation is to be expected due to the dependence of the proximity signal on the sensor weights.

Here, only the front proximity weight (from sensor number 3) has been plotted. This is taken as indicative of the weight values in general, though unsurprisingly they all vary from each other as they are all controlled by their own update calculations. Figure 10 shows four of the weight values during the course of the experiment. Sensor weight 3 (the front sensor), is the one used as an indicator in the graphs in this section. Weight 4 (the front-right sensor) follows the same approximate shape, but is often a little higher than weight 3. This is possibly due to the tendency of the system to approach obstacles at an angle, so the angled sensor would generally receive more stimulation. Weight 5 (on the right hand side) again follows a similar shape but is considerably lower in value, and weight 6 does not really increase as it is not generally active when collisions are occurring.

B. Improving the system – A value-dependent learning scheme

The version presented to far, encounters serious problems when readings from the sensors are not reliable.

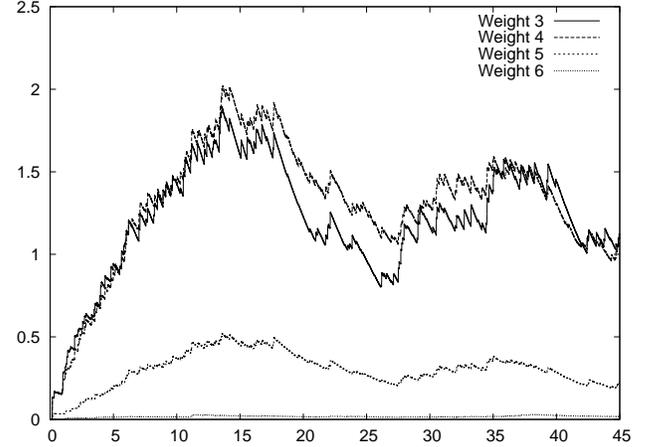


Fig. 10. The right hand side sensor weights from a control experiment

The argument given in section III was that, for example in the case of under-sensitivity, the collision signal would be activated more often and thus the proximity weights would increase to a higher value, cancelling out the error. This argument is in fact invalid because, although the collision signal is indeed present more often in the under-sensitive condition, an equilibrium (where the weight update Δw equals 0) is reached at a much lower weight, and thus the error is in fact exaggerated.

An improvement to the system, inspired by [6], was made by modifying the weight update rule using the principle of value-based learning. Previous research has shown that value-based learning can improve the stability of basic Hebbian systems. Here a value signal is used to try and improve the systems robustness to sensor errors.

The principle is to add a multiplicative term to the weight update that modulates the learning, limiting the updates to only taking place when they are deemed useful. If we say that the “task” of the robot is to avoid walls without colliding with them, then it follows that when the system is not colliding with anything, it is succeeding. Logically, when the robot is in this successful state, learning serves no useful purpose, since the current configuration is already good enough. It follows that learning *should* occur when the robot is *failing* its task, as this is the only way it will ever be able to stop failing.

If, then, we define “failing the task” as bumping into walls, the collision signal, A_1 provides a good indicator of when this is happening. Consider then equation 14

$$\Delta w_{2j} = A_1(\eta R_g x_{2j} - \epsilon R_g w_{2j}) \quad (14)$$

Parameter	Setting
avoid gland stimulation rate	0.4
avoid gland decay constant	0.9
feedback stimulation rate	0.04
feedback decay constant	0.99
avoid gland learning rate	0.01
avoid gland forgetting rate	0.00005

TABLE II

PARAMETER SETTINGS FOR EXPERIMENTAL RESULTS SHOWN IN FIGURE

11

The controller was re-run with the value based learning system, including with the same sensitivity errors that were used before. Figure 11 shows a plot of the results from the new value-based learning controller against the equivalent results from the previous controller. Here the collision signals have been plotted using a 5000 step moving average, which makes it easier to see how the frequency of collisions changes over longer periods. Parameter settings for these experiments can be found in Table II.

First, consider the top two graphs in Figure 11. With the original learning rule, the sensor weight rises sharply to about 1.6 within around 15 minutes. After this point, it falls and rises slowly. It appears to vary between about 0.8 and 1.6 – that is to say the rate appears to rise to about twice the level that is necessary to avoid collisions. By comparison, when the value signal is introduced, the sensor weight seems to rise much more slowly, only reaching 1.0 after 35 minutes. However, the weight is much more stable, varying from roughly 0.9 to 1.0 between 20 minutes and 45 minutes – a change of approximately 11% compared to 100% using the original learning rule. We can also see from the graph how the collision signal decays over time. With the value-based learning rule, it is clear that it takes longer in terms of time and in terms of number of collisions for the robot to complete its learning phase. After 45 minutes, the value-based system seems to have learnt to prevent the majority of collisions, but not quite all. This can be seen from the decay in the collision signal – since the environment is constant the only way this could be achieved is by activating the avoidance behaviour without waiting for the collision signal. This implies that the weight increase is converging to a point where it is only just high enough to prevent collisions, and no higher.

In summary, the value-dependent learning scheme learns more slowly, but seems to be less susceptible to the overgeneralisation problem (excessively high weight values) of the basic Hebbian scheme.

1) *Performance under simulated errors*: Two further experiments designed to assess whether the architecture could deal with partial sensor failures were also carried out, one with and under-sensitive sensor and one with an over-sensitive sensor.

When under-sensitivity error is introduced the sensor weight in the value dependent learning scheme (middle graph, Figure 11b) increases slowly compared to when the control. However it is not clear that the weight is converging within the 45 minute experiment. The collision signal seems to be falling

at a lower rate, and it is not clear what is happening in this experiment, since it appears that the run was not long enough to incorporate the whole learning period. The learning is much slower than the original learning rule although the number of collisions after 45 minutes does not seem to have been substantially decreased by using the value system (the collision signal is at around 0.01 in both cases).

In the over-sensitivity experiments (bottom graph), the ability of the value-dependent learning rule to avoid over-generalisation is emphasized. The initial weight increases happen much more quickly (bottom graph Figure 11b) but after 15 minutes the weight does not increase significantly, and collisions are kept below the 0.004 mark. This compares extremely well with the pure Hebbian learning scheme, where the weight increases well beyond the needed value. With such large weights applied to the proximity signal, the robot was avoiding objects from a distance of 4cm or more, and its behaviour could be described as “confused” – i.e. a lot of its movement was erratic and unnecessary. With the value-based learning scheme, the behaviour was hard to tell apart from the error-free system.

Whilst far more work is needed on this, these results show two potential improvements offered by the value based learning scheme:

- The need for “re-learning”, that is, the need for constant stimulation from the collision signal, is reduced. This is implied but not conclusively demonstrated in the results from the under-sensitive condition.
- The problem of overgeneralisation is definitely mitigated, as shown in the over-sensitive condition.

V. CONCLUSIONS

Our ultimate goal is to create a combined immune-neural-endocrine behaviour based robotic control system for long-term autonomy that affords “homeostasis” on the robotic unit. As one further step towards that goal, we have presented a novel adaptive artificial endocrine system (AAES) for the modulation of neural networks. The new AAES allows us to incorporate learning into a combined AAES-ANN behaviour based control system, which previously was unable to learn. Through the use of a value-based learning system, we have demonstrated the ability of the AAES to adapt “on-line” and afford a certain amount of *fault tolerance* to the control system, with the AAES being able to cope with faulty sensor data. Our on-going work is examining the scalability of the combined AAES-ANN system with respect to the number of behaviors that are possible to include. Initial work suggests that we can include up to 8 behaviors within a single control system, and reliably perform the required tasks over prolonged periods of time.

ACKNOWLEDGEMENTS

This research is funded by EOARD, grant number FA-8655-07-3061.

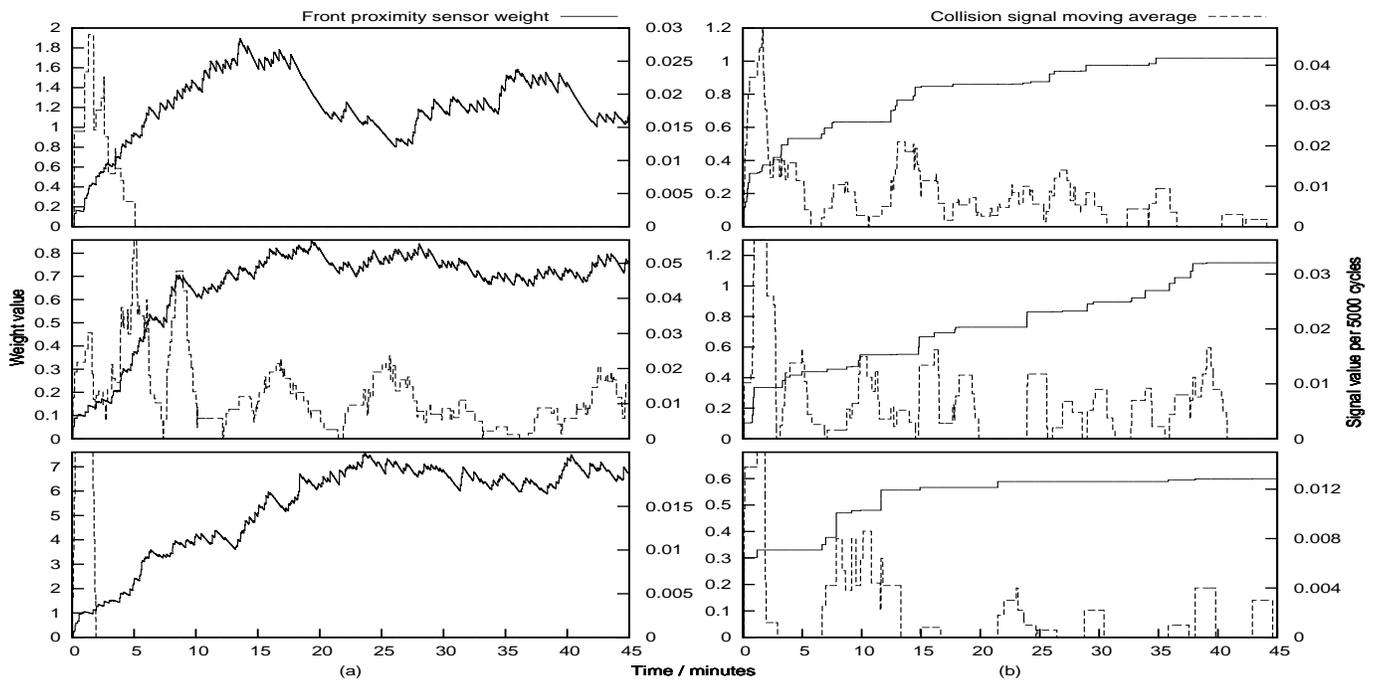


Fig. 11. Comparison of (a) original and (b) value-dependent learning schemes. The top two graphs are the control condition (with no error). The middle graphs are the under-sensitive condition and the bottom graphs are the over-sensitive condition

REFERENCES

- [1] A. J. Vander, J. Sherman, and D. Luciano, *Human Physiology: The Mechanisms of Body Function*, 5th ed. McGraw-Hill, 1990.
- [2] M. Neal and J. Timmis, "Timidity: A Useful Mechanism for Robot Control?" *Informatica*, vol. 27, no. 4, pp. 197–204, 2003. [Online]. Available: <http://www.cs.kent.ac.uk/pubs/2003/1635>
- [3] —, *Recent Developments in Biologically Inspired Computing*. Idea Group Publishing, 2005, ch. Once More unto the Breach: Towards Artificial Homeostasis, pp. 340–365.
- [4] P. Vargas, R. Moioli, L. N. de Castro, J. Timmis, M. Neal, and F. J. Von Zuben, "Artificial homeostatic system: A novel approach." in *Proc. Advances in Artificial Life: 8th European Conference*, ser. LNAI, no. 3630, 2005, pp. 754–763.
- [5] R. Brooks, "A robust layered control system for a mobile robot," *IEEE Journal of Robotics and Automation*, vol. 2, no. 1, pp. 14–23, 1986.
- [6] P. F. M. J. Verschure, J. Wray, O. Sporns, G. Tononi, and G. M. Edelman, "Multilevel analysis of classical conditioning in a behaving real world artifact," *Robotics and Autonomous Systems*, vol. 16, pp. 247–265, 1995.