A Comparative Study of Value Systems for Self-Motivated Exploration and Learning by Robots

Kathryn Elizabeth Merrick

Abstract—**A range of different value systems have been proposed for self-motivated agents, including biologically and cognitively inspired approaches. Likewise, these value systems have been integrated with different behavioral systems including reflexive architectures, reward-based learning and supervised learning. However, there is little literature comparing the performance of different value systems for motivating exploration and learning by robots. This paper proposes a neural network architecture for integrating different value systems with reinforcement learning. It then presents an empirical evaluation and comparison of four value systems for motivating exploration by a *Lego Mindstorms NXT* robot. Results reveal the different exploratory properties of novelty-seeking motivation, interest and competence-seeking motivation.**

Index Terms—**Competence, developmental robotics, interest, motivated reinforcement learning, novelty, value system.**

I. Introduction

SELF-MOTIVATED robots have an embedded value system that mediates the saliency of environmental stimuli. This allows the robot to self-supervise and self-organize its own exploratory and learning activities. The value system signals the occurrence of important stimuli and triggers the formation of goals. These goals are then acted on by a behavioral system. The behavioral system may use reflexes, learning, planning or other processes to achieve the goals generated by the value system.

Value systems play an important role in the design of robots with adaptive, lifelong learning behavior, because they provide a way for robots to behave autonomously through spontaneous, self-generated activity. This is in contrast to robots without value systems, which often rely on instructions provided by their human designer, or a human teacher, to determine the goals they will pursue.

Many different value systems have been proposed for robots and other artificial systems [1]–[9]. However, there is little literature comparing the performance of these techniques for motivating exploration and learning in complex environments, such as those inhabited by robots. Making such a comparison is currently difficult for a number of reasons. First, a range of different system architectures have been used to model different value systems. Secondly, these value systems have been combined with different behavioral systems including reflexive behaviors [8], reinforcement learning [2], [9] and supervised learning [10]. Thirdly, a range of different metrics have been used to evaluate self-motivated agents and robots [2].

This paper addresses some of these challenges, specifically in a reinforcement learning setting. First, an integrated neural network architecture is proposed as a framework for combining different cognitive value systems with function approximation reinforcement learning. Secondly, an empirical evaluation is made of four value systems using this architecture, to compare the exploratory and learning capabilities of each value system.

The remainder of this paper is organized as follows. Section II begins with a brief review of theories of motivation from natural systems and, where they exist, corresponding artificial value systems. The range of system architectures identified by the review demonstrates the need for a flexible, integrated architecture that can support a number of different value systems. Section III proposes such an architecture for motivated reinforcement learning (MRL), based on a neural network formalism. Four variations of the architecture are presented for novelty-seeking behavior, interest, competence-seeking behavior and random exploration. The experimental evaluation is presented in Section IV using a *Lego Mindstorms NXT* robot to compare the stability, variety and complexity of the robot's behavior using different value systems. Results reveal the different exploratory properties of novelty-seeking motivation, interest and competence-seeking motivation.

The paper concludes by discussing a number of variations of the proposed architecture to demonstrate the flexibility of the model and its capacity to support more complex variants of MRL in future.

II. Motivation Theories and Value Systems

In natural systems research, neuroscientists, psychologists and ethologists have proposed different theories describing the forces that motivate action. These can be loosely classified as biological theories that work within the biological system of a behaving organism; cognitive theories that cover theories of the mind abstracted from the biological system of the behaving organism; social theories concerned with what individuals do when they are in contact with one another; and combined theories that synthesize ideas from several or all of the previous categories [2]. Value systems for artificial agents have been

Manuscript received January 26, 2010; revised April 13, 2010 and May 07, 2010; accepted May 10, 2010. Date of publication May 27, 2010; date of current version June 30, 2010. This work was supported in part by a UNSW@ADFA Silverstar Grant: UNSWA OP002 Z6300 0000 00 PS19105.

The author is with the University of New South Wales, Australian Defence Force Academy (e-mail: k.merrick@adfa.edu.au).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TAMD.2010.2051435

considered primarily in the biological and cognitive categories, although some social models have been proposed.

In this paper, the term “value system” is used to describe a subsystem that mediates the saliency of environmental stimuli on behalf of an artificial agent or robot. A value system may thus model a motivation theory from natural systems literature or it may incorporate concepts relevant only to artificial systems. The following sections review existing value systems for robots and other artificial systems, and their parallels in natural systems.

A. Biological Motivation Theories and Value Systems

Biological motivation theories explain motivation in terms of the processes that work at a biological level in natural systems. Examples include neuromodulatory theories [4], [5], [11], drive theory [12], [13], motivational state theory [14] and arousal theory [15], [16].

Biological value systems for robots and other artificial systems have tended to focus on neuromodulatory and drive-based approaches. Neuromodulatory systems in robots are based on computational models of neurons in the brain. They define properties such as how neurons influence each other, how long neurons activate for, and the regions of the brain that are affected. Existing work with neuromodulatory value systems in robots has focused on areas such as the adaptation of appetitive and aversive behavior [4] and adaptation of the visual system [5].

Drive theory [12], [13] holds that homeostatic requirements drive an individual to restore some optimal biological condition when stimulus input is not congruous with that condition. Drive-based value systems have been studied in the artificial life community as an approach to building action-selection architectures [17]–[19]. Action-selection architectures make decisions about what behaviors to execute in order to satisfy internal goals and guarantee an agent’s continued functioning in a given environment.

B. Cognitive Motivation Theories and Value Systems

In contrast to biological value systems, cognitive value systems are based on psychological theories of the mind, abstracted from the physical organism. Examples include curiosity [20], incentive motivation [21], operant theory [22], achievement motivation [23], attribution theory [24] and intrinsic motivation [25].

Oudeyer *et al.* [1] classify cognitive value systems for robots in three categories: error maximization (EM), progress maximization (PM) and similarity-based progress maximization (SBPM). These categories reflect the idea that robots using value systems try to choose actions that will maximize the value of a reward signal, and that this reward signal may be calculated in different ways.

Robots using EM techniques focus on actions that permit them to learn about stimuli for which they currently have a high prediction error. Examples of EM approaches [3], [6], [26], [27] can often be thought of as modeling the ‘novelty’ of a stimulus and seeking out stimuli of high novelty. The main criticism of EM approaches is that random occurrences often result in a high prediction error, but there is little to be learned from such occurrences. Alternative approaches that can filter out random occurrences are thought to be required in robotics domains, where random occurrences may be a result of sensor noise.

PM techniques focus attention on stimuli for which the robot “predicts that it will have a high prediction error.” This more indirect method of computing reward overcomes some of the difficulties associated with EM techniques in environments that may contain random occurrences. Examples include work by Kaplan and Oudeyer [28] and Herrmann *et al.* [29].

SBPM techniques are like PM techniques, but take into account the similarity of observations when making predictions. These approaches often incorporate an unsupervised learning algorithm or other mechanism to cluster similar experiences, before computing a “novelty,” “interest,” or “curiosity” value for the learned cluster [1], [3], [30].

Alternative computational models of curiosity for applications other than robots include Saunders and Gero’s curious social force model for design agents [31], Schmidhuber’s curious neural controller [32], and Lenat’s AM [33].

C. Social Motivation Theories and Value Systems

Social motivation theories are psychological theories concerned with what individuals do when they are in contact with one another. They include conformity [21], creativity [34], cultural effect [21], and evolutionary theories [35].

One example of a social value system for software agents is proposed by Saunders and Gero [36]. They present a computational model of creativity that captures the social aspects of an individual’s search for novelty. Incorporation of evolutionary forces in self-motivated agents has been considered by Singh *et al.* [37].

D. Combined Motivation Theories

A small number of psychological motivation theories synthesize biological, cognitive and social motivation theories. These include Maslow’s Hierarchy of Needs [38], Existence-Relatedness-Growth (ERG) theory [39] and Stagner’s steady state model [40]. The development of combined artificial value systems remains an open research challenge.

III. An Integrated Model for Motivated Reinforcement Learning

While this paper does not claim to present an architecture for a combined artificial value system, it does make a step towards a generic platform for the integration, and potential combination, of different cognitive motivation functions with reinforcement learning. In particular, features of the proposed architecture include:

- a combined memory model for the value system and reinforcement learning components, so the robot has a single, consistent, shared representation of long-term memory;
- a uniform notation for both the value system and the learning module, based on the idea of an artificial neuron.
- computational models of motivation as the only driver for exploration and learning;
- capacity for either exemplar learning or function approximation, as required by a given application.

A. Generic Architecture

The proposed integrated MRL architecture is a multilayer neural network with a generic structure shown in Fig. 1. The ar-

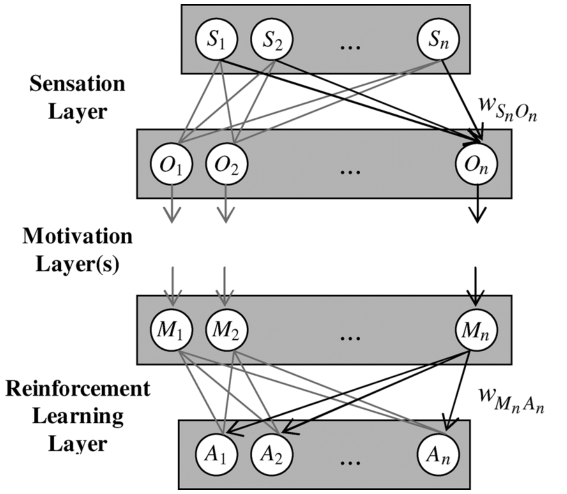

Fig. 1. Generic network structure for motivated reinforcement learning.

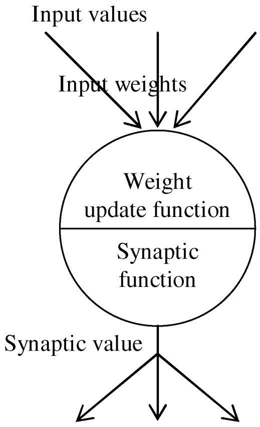

Fig. 2. Generic neuron structure and processes for motivated reinforcement learning.

chitecture uses four main types of neurons organized in layers: sensory neurons (input neurons), observation and motivation neurons (hidden neurons) and activation neurons (output neurons).

In each layer, one or more "winning" neurons:

- inputs values from connected neurons in the previous layer;
- updates connected input weights;
- computes and outputs a synaptic value to one or more neurons in the next layer.

The main components of a generic MRL neuron are illustrated in Fig. 2. They are input values passed from neurons in the previous layer; input weights representing the strength of the connection between the current neuron and neurons in the previous layer; a weight update functions which calculates how those connections change over time; and a synaptic function which calculates the output synaptic value to be passed to connected neurons in the next layer.

While all neurons have these components, different neurons play different roles in the network. Sensory neurons S_n are responsible for inputting raw data from the robot's sensors to the network. There is one sensory neuron for each piece of sensor data generated at a given time. This data then passes through three types of layers made up of different types of neurons: the sensation layer with weights connecting sensory neurons to observation neurons, one or more motivation layers with weights connecting motivation neurons, and the reinforcement learning layer with weights connecting motivation neurons to activation neurons. Activation neurons A_n output data from the network to the robot's physical actuators. There is one activation neuron for each primitive action available to the robot.

The following paragraphs describe the flow of data through the network in detail, to show how an action is selected. This paper assumes a Q-learning [41] approach to reinforcement learning. Q-learning is particularly appropriate as an approach to MRL because it is an incremental, online algorithm that can learn at each step of the robot's interaction with its environment. Using Q-learning, at each time t, the robot reasons about its current sensory data $S_{1(t)}, S_{2(t)}, S_{3(t)}, \ldots S_{n(t)}, \ldots$, the last action it performed $A_{(t-1)}$ and its previous sensory data $S_{1(t-1)}, S_{2(t-1)}, S_{3(t-1)}, \ldots S_{n(t-1)}, \ldots$.

This paper makes two main theoretical contributions: 1) adaptation of neural network notation to describe motivated reinforcement learning problems and 2) translation of three motivation functions and the Q-learning update and action-selection functions into the new notation. A third practical contribution of the paper is the demonstration, evaluation and comparison of four specific instances of the generic approach.

1) Sensation Layer: The first layer of the network is the sensation layer that generalizes over the state space encountered by the robot. This layer clusters raw sensory data $S_{1(t)}, S_{2(t)}, S_{3(t)}, \ldots S_{n(t)}, \ldots$ at a given time t, to an observation neuron $O_{(t)}$, that best represents the data. This layer can incorporate common clustering algorithms such as K-Means clustering, Self-Organizing Maps (SOMs) [42], Adaptive Resonance Theory (ART) networks [43], or Growing Neural Gases (GNGs) [44]. Depending on the algorithm used, there may be any number of observation neurons and, additionally, this number can be fixed or variable. Observation neurons can be organized as a set or have a topological relationship.

All of the models in this paper use a Simplified ART (SART) network [43] in the sensation layer. This approach has had demonstrated success in robotics applications with MRL [30]. Using this approach, there are initially no observation neurons. When sensory data is presented to the network, either a new observation neuron is created (with associated weights) or an existing observation neuron is selected with weights that best describe the sensory data.

More formally, at each time t, raw sensory data $S_{1(t)}, S_{2(t)}, S_{3(t)}, \ldots S_{n(t)}, \ldots$ are the input values to the observation neurons. The observation neuron $O_{min(t)}$ with the minimum Euclidean distance to the sensory data is identified as follows:

$$O_{\min(t)} = \arg\min_{O} \sqrt{\sum_{n} \left(S_{n(t)} - w_{S_nO_n(t-1)}\right)^2}.$$

If $O_{\min(t)}$ is within some distance ρ of the sensed state (called the vigilance constraint), the weights connecting $O_{\min(t)}$ to the sensory neurons are updated. That is, each of the connected weights $w_{S_nO_n}$ is modified using the update function:

$$w_{S_nO_n(t)} = w_{S_nO_n(t-1)} + \eta\left(S_{n(t)} - w_{S_nO_n(t-1)}\right)$$

where η is the learning rate of the SART network. We denote the updated neuron $O_{\text{min-updated}(t)}$

If $O_{min(t)}$ does not satisfy the vigilance constraint, a new observation neuron $O_{\text{new}(t)}$ is created with associated weights. $O_{\text{new}(t)}$ uses the sensory data $S_1, S_2, S_3, \ldots S_n, \ldots$ to initialize its connected weights. This means that the sensation layer is never randomized.

In summary the sensation layer identifies a winning observation neuron $O_{(t)}$ with weights that best describe the current sensory data as follows:

$$O(t)=\begin{cases} O_{\text{min-updated}(t)}, & \text{if } \sqrt{\sum_n \left(S_{n(t)}-w_{S_nO_n(t-1)}\right)^2} \leq \rho \\ O_{\text{new}(t)}, & \text{otherwise} \end{cases}$$

Using a vigilance constraint $\rho > 0$, function approximation ensures that the SART network remains stable enough to guard against expansion caused by noisy sensor data, or simply the size and complexity of the state space, but flexible enough to generate new observation neurons when required.

If $\rho = 0$ then there is no function approximation and each unique combination of sensory data will trigger the creation of a new observation neuron. This is equivalent to instance-based, exemplar learning. In reinforcement learning terms it would be equivalent to using a table-based approach. However, the advantage of the proposed approach is that the layered neural network can integrate complex motivation layers in series or in parallel. This would not be easily achieved using an extended table-based approach. Further examples of the flexibility of the layered neural network model are discussed in Section V.

The synaptic values O_n output by each observation neuron are computed using the following synaptic function:

$$On = \begin{cases} 1, & \text{if } O_n = O_{(t)} \text{ (i.e., if } O_n \text{ is a winning neuron)} \\ 0, & \text{otherwise.} \end{cases} \quad (1)$$

In this way, the continuous-valued raw sensory data input to the network is converted to a series of binary action potentials. These action potentials trigger activity in other parts of the network. Specifically, in the models in this paper, the synaptic values O_n are input to the first motivation layer.

2) Motivation Layer(s): The motivation layers each have one neuron for every observation neuron in the sensation layer, as shown in Fig. 1. As such, they may also have a fixed or variable number of neurons, depending on the structure of the sensation layer. For example, using the SART-based function approximation described above, motivation neurons and associated weights will be added to the network when new observation neurons are created.

There may be several layers of motivation neurons, depending on the complexity of the motivation function used. Motivation neurons in the first motivation layer receive their input values from observation neurons according to (1). Their associated weights are updated using a computational model of motivation as the update function. Three specific examples of this process are described in Parts B, C, and D later. Motivation neurons in the last motivation layer, output synaptic values M_n to the reinforcement learning layer.

Depending on the nature of the motivation function all motivation neurons may be updated at each time-step, or some subset may be updated. For example, some motivation functions may require only the motivation neuron connected to the winning observation to be updated. We refer to such a neuron as the "winning motivation neuron."

3) Reinforcement Learning Layer: Weights in the reinforcement learning layer are updated to reinforce actions that are highly motivating, according to the activities of the motivation layer. We use a Q-learning approach in this paper so only the weight connecting the winning motivation neuron to the action performed at the previous time-step is updated. For example, if a network with one motivation layer is assumed, the update equation is

$$\begin{aligned} w_{M_{(t-1)}A_{(t-1)}(t)} &= w_{M_{(t-1)}A_{(t-1)}(t-1)} \\ &\quad + \beta \Big[w_{O_{(t)}M_{(t)}(t)} + \gamma \max_n w_{M_{(t)}A_n(t-1)} \\ &\qquad - w_{M_{(t-1)}A_{(t-1)}(t-1)} \Big]. \end{aligned}$$

This update is comparable to the standard Q-learning update with the weight-based notation $w_{M_{(t-1)}A_{(t-1)}(t)}$ in place of the table-based notation $Q_{(t)}(M_{(t-1)}, A_{(t-1)})$. $M_{(t-1)}$, can be thought of as the current motivational state of the agent. In addition, reward $R_{(t)}$ is defined by the motivation weight $w_{O_{(t)}M_{(t)}(t)}$. An action is reinforced by incorporating a percentage of both the current motivation and future expected motivation for performing that action in response to a given observation. β is the reinforcement learning rate and γ is the discount factor for future expected motivation.

Using the modified notation, a winning activation neuron is selected according to the synaptic function:

$$A(t) = \arg\max_n w_{M_{(t)}A_n(t-1)}.$$

This function greedily selects the action with the highest weight. Note that there is no additional exploration component such as e-greedy exploration incorporated into this action-selection equation. Rather, exploration and exploitation are both controlled by the motivation function embedded in the motivation layer. The following sections describe a number of such motivation functions.

B. Novelty-Seeking Exploration

Novelty-seeking robots find unfamiliar stimuli the most highly motivating. Stimuli may be unfamiliar because the robot has never encountered them before or because the robot has not encountered them for a long time. To model this latter property, a model of habituated novelty [45] is used in this paper.

The network for novelty-seeking exploration and learning is shown in Fig. 3. This network follows the generic structure described above, but specifically incorporates a novelty layer as the motivation layer.

The sensation layer is implemented as described above using a SART network. Novelty neurons and weights are added progressively as observation neurons are added in the sensation layer. Weights are initialized to $w_{O_nN_n(0)} = 1$ and $w_{N_nA_n(0)} = 1$ meaning that all observations are initially considered to be highly novel. The novelty layer takes the synaptic values output by the observation neurons as input. Weights in the novelty layer

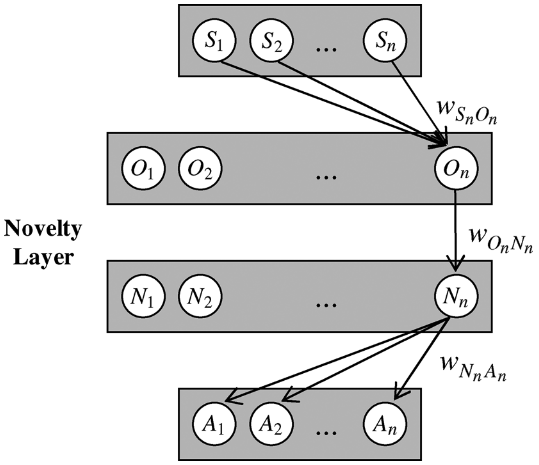

Fig. 3. Network structure for novelty-seeking exploration.

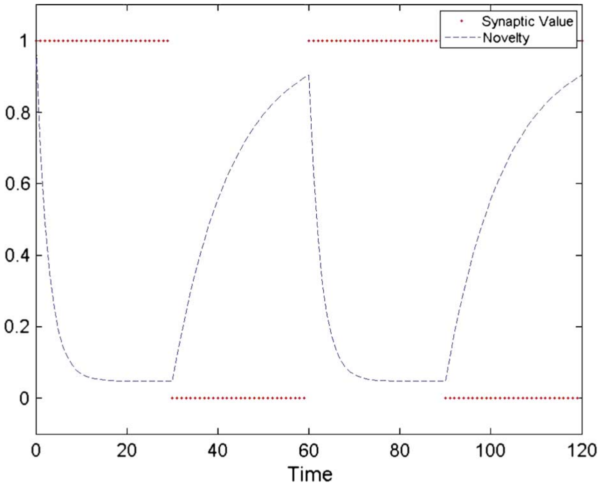

Fig. 4. Change in novelty over time [see (2)] in response to changing synaptic values from the observation layer [see (1)].

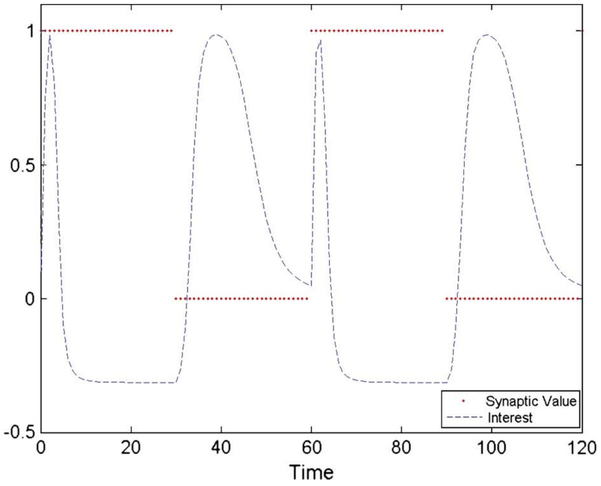

Fig. 5. Change in interest over time in response to changing synaptic values from the observation layer. Compare to Fig. 4. to see the difference between novelty and interest.

are updated using a step-wise approximation of Stanley's habituation function [45] so

$$w_{O_nN_n(t)} = w_{O_nN_n(t-1)} + \frac{\mathrm{d}w_{O_nN_n}}{\mathrm{d}t}$$

where

$$\frac{\mathrm{d}w_{O_nN_n}}{\mathrm{d}t} = \frac{\alpha\left[w_{O_nN_n(0)} - w_{O_nN_n(t-1)}\right] - O_n}{\tau}$$

α is a constant governing the rate of recovery of novel observations and τ is a constant governing the rate of habituation such that:

$$\tau = \begin{cases} \tau_1, & \text{if } O_n = 1 \\ \tau_2, & \text{otherwise} \end{cases}$$

τ_1 governs the rate at which novelty decreases for winning observations, while τ_2 governs the rate at which novelty increases for losing observations.

Novelty neurons output a synaptic value N_n computed using the following synaptic function:

$$Nn = \begin{cases} w_{O_nN_n(t)}, & \text{if } O_n = 1 \\ 0, & \text{otherwise} \end{cases} \tag{2}$$

This means that the action potentials computed by novelty neurons [see (2)] are influenced by those computed by observation neurons in (1). An example of the novelty curve is shown in Fig. 4. This curve shows how novelty changes in response to changing binary synaptic values passed from the observation layer.

C. Interest-Seeking Exploration

Novelty and interest differ in that interest is highest for observations of moderate novelty and lowest for observations of very low or very high novelty. This precludes both very familiar observations and very unfamiliar observations from being highly motivating. In particular, the interest curve in this paper generates the lowest interest values for stimuli of very low novelty, as shown in Fig. 5. This models an aversive response to stimuli of very low novelty.

Fig. 5. shows the change in interest in response to changing synaptic values from the observation neurons. The interest layer takes the synaptic values output by the novelty neurons (see (2) and Fig. 4.) as its direct input. Weights in the interest layer are updated using two sigmoid functions $F^+(N_n)$ and $F^-(N_n)$ to provide positive and negative feedback for very high and very low novelty, respectively [8]:

$$\begin{aligned} w_{N_nI_n(t)} &= F^+(N_n) - F^-(N_n) \\ &= \frac{F^+_{\max} - F^+_{\text{avn}}\mathrm{e}^{-10\left(2N_n - F^+_{\min}\right)}}{1 + \mathrm{e}^{-10\left(2N_n - F^+_{\min}\right)}} \\ &\quad - \frac{F^-_{\max} - F^-_{\text{avn}}\mathrm{e}^{-10\left(2N_n - F^-_{\min}\right)}}{1 + \mathrm{e}^{-10\left(2N_n - F^-_{\min}\right)}} \end{aligned}$$

N_n is the synaptic value passed from the novelty layer, as shown in (2). $F^+_{\min}$ is the minimum novelty to receive positive feedback and $F^-_{\min}$ is the minimum novelty to receive negative feedback $F^+_{\max}$ is the maximum positive feedback and $F^-_{\max}$ is the maximum negative feedback. F^+_{avn} controls the level of aversion to low novelty while F^-_{avn} controls the level of aversion to high novelty. Weights are initialized to $w_{O_nN_n} = 1$,

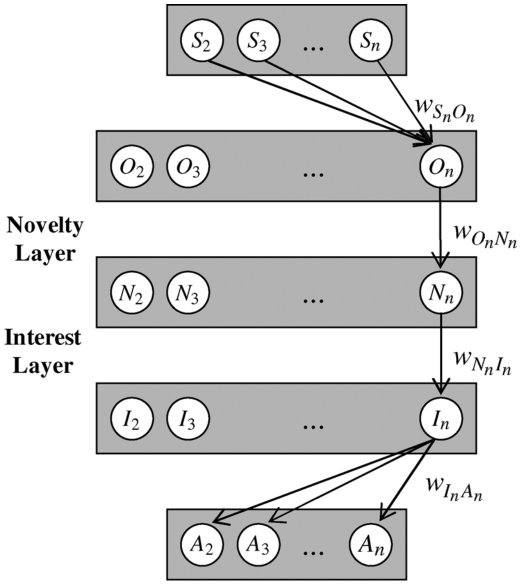

Fig. 6. Network structure for interest-seeking exploration.

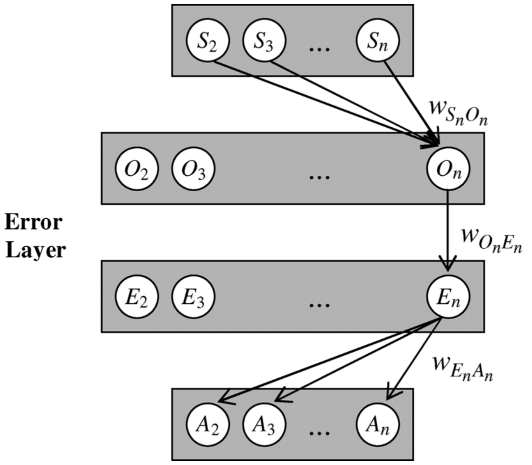

Fig. 7. Network structure for competence-seeking exploration.

$w_{O_nI_n(0)} = 0$ and $w_{I_nA_n(0)} = 0$, meaning that all observations are initially highly novel and thus of low interest to the robot.

Interest neurons output a synaptic value of $I_n = w_{N_nI_n(t)}$.

The network for interest-seeking exploration and learning is shown in Fig. 6. This network is based on the network for novelty-seeking exploration described above, but incorporates an additional motivation layer for interest.

D. Competence-Seeking Exploration

Competence-seeking exploration differs from novelty and interest-based exploration in that stimuli specifically maintain high motivation values until the robot can repeat the stimuli with a small, but positive, learning error. A small, positive learning error indicates that the learned network weights are an accurate prediction of the robot's self-motivation for the associated observation and action. The robot is thus internally competent at fulfilling its current motivational needs.

The network for competence-seeking exploration and learning is shown in Fig. 7. This network follows the generic structure described in Part A above, but specifically incorporates an error layer to permit the robot to compute its competence at various tasks.

The sensation layer is implemented as described in Part A using a SART network. Error neurons and weights are added progressively as observation neurons are added in the sensation layer. Weights are initialized to $w_{O_nE_n(0)} = 0$ and $w_{E_nA_n(0)} = 0$, meaning that the robot initially models itself as incompetent at all tasks. Weights in the error layer are updated according to

$$w_{O_{(t-1)}E_{(t-1)}(t)} = \beta \Big[C_{(t)} + \gamma \max_n w_{E_{(t)}A_n(t-1)} - w_{E_{(t-1)}A_{(t-1)}(t-1)} \Big]$$

$w_{O_{(t-1)}E_{(t-1)}(t)}$ is the learning error for the reinforcement learning update (often referred to as ΔQ using traditional notation). $C_{(t)}$ is a competence-based reward signal that has two rules as follows:

$$C_{(t)} = \begin{cases} 1, & \text{if } \exists \tau > 1 \text{ such that } O_{(t-\tau)} = O_{(t-1)} \text{ and} \\ & \quad w_{E_{(t-1)}A_{(t-1)}(t-1)} > \varepsilon \text{ or} \\ & \quad 0 > w_{E_{(t-1)}A_{(t-1)}(t-1)} > -\varepsilon) \\ -1, & \text{otherwise} \end{cases}$$

The first rule assigns the highest reward of 1 to observation neurons that win repeatedly and cause learning. This rule contains a component to continue rewarding winning observation neurons will continue to improve the robot's prediction of its current motivational needs (i.e. continue to have $w_{E_{(t-1)}A_{(t-1)}(t-1)} > \varepsilon$) and a component to switch to rewarding observation neurons that the robot predicts will soon start to improve the its prediction of its motivational needs (i.e. those with $0 > w_{E_{(t-1)}A_{(t-1)}(t-1)} > -\varepsilon$). The second rule assigns other winning observation neurons a punishment of -1.

So that the robot learns faster than it forgets, in this model, the reinforcement leaning rate β is split into two parameters β_1 and β_2 governing the rates of learning and forgetting respectively.

$$\beta = \begin{cases} \beta_1, & \text{if } C_{(t-1)} = 1 \\ \beta_2, & \text{otherwise} \end{cases}$$

The competence-based reward function is shown diagrammatically in Fig. 8. Because competence is linked to the reinforcement learning error, it can only be computed when learning occurs. In other words, competence motivation is computed only in motivation neurons attached to the winning observation neurons [which generate a synaptic value of one, as shown in (1)]. This means that a robot maintains its competence with observations even if it has not repeated that observation for some time. This is in contrast to novelty-based approaches where the novelty of an observation rises if the observation has not been repeated for some time.

Error neurons output a synaptic value of $E_n = w_{O_nE_n(t)}$ so the update equation for the activation layer becomes

$$w_{E_{(t-1)}A_{(t-1)}(t)} = w_{E_{(t-1)}A_{(t-1)}(t-1)} + E_n.$$

E. Random Exploration

This network, shown in Fig. 9, has the same structure as the network for novelty-seeking exploration. However instead of a motivation layer, a randomized layer is used to generate random

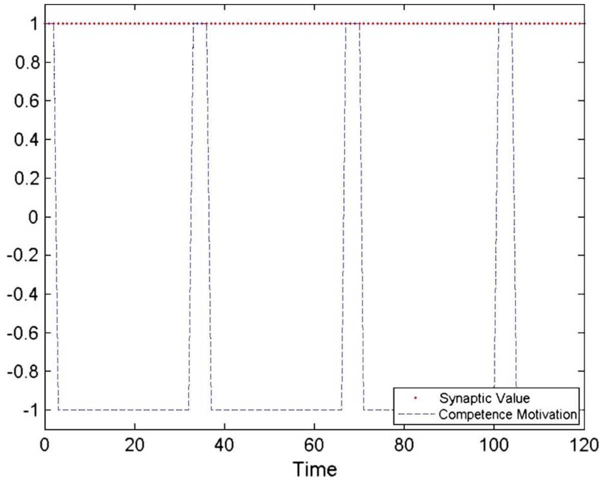

Fig. 8. Change in competence motivation over time. Note that competence motivation is only updated for winning observation neurons, that have an associated learning error. Winning neurons have a synaptic value of one [see (1)].

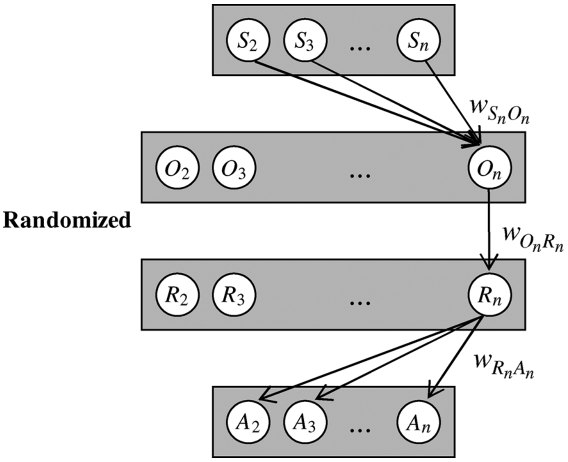

Fig. 9. Network structure for random exploration.

exploration. This model acts as a naïve baseline against which to measure and understand the other models.

The sensation layer is implemented as described in Part A using a SART network. Neurons and weights in the randomized layer are added progressively as observation neurons are added in the sensation layer. To provide a comparison with the other models, weights are initialized to $w_{O_nR_n(0)} = 0$ and $w_{R_nA_n(0)} = 0$.

At each time t, the weight linked to the winning observation neuron is updated according to

$$w_{O_{(t)}R_{(t)}(t)} = \text{rand}(-1, 1)$$

That is, weights are assigned a random number between -1 and 1. Randomized neurons output a synaptic value of $R_n = w_{O_nR_n(t)}$ to the reinforcement learning layer.

IV. Experiments

This section describes an experimental evaluation of the four models described above on a *Lego Mindstorms NXT* robot. Part A describes the robot. Part B describes the strategies used to characterize the performance of the robot. Finally, Part C discusses the results of the experiments.

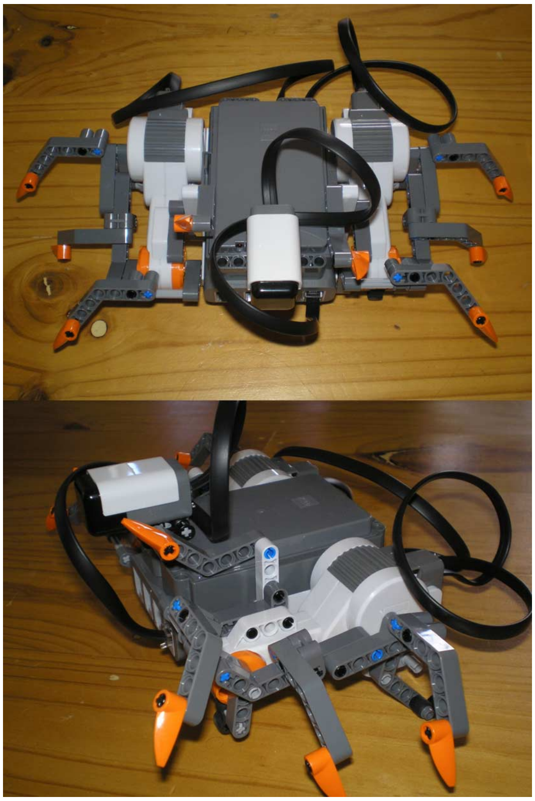

Fig. 10. A “crab” robot using the *Lego Mindstorms NXT* platform.

A. The “Crab” Robot

The physical structure of the robot, which roughly resembles a crab, is shown in Fig. 10. The robot has two servo motors controlling the left and right sets of legs. The robot can move each motor forwards or backwards or stop a motor. The robot can sense whether the motors are moving or not, and in which direction. In addition, the robot can sense the position (rotation) of each motor using the motors’ built in tachometers. The robot is also equipped with an accelerometer permitting it to sense its acceleration and tilt in three dimensions.

Table I summarizes the sensory neurons needed by this robot and the range of values produced by their associated sensors. Table II summarizes the activation neurons.

While this robot is relatively simple in comparison to other systems, it provides the basic structure for a robot that can potentially learn to walk: motors to control the action of the legs and an accelerometer to monitor the movement of the body. Previous experiments with a similar but simpler “Ant” robot [30] shown in Fig. 17 demonstrated a self-motivated walking behavior using a cycle-based motivation function. This paper extends that work with a comparison of different value systems on a more complex mobile robot.

To permit a fair comparison of the four value systems described in Section III, common parameters use the same values. All parameters and their values are summarized in Table III. Each value system was run five times on the robot for 4000 time-steps (approximately 30 min). Where appropriate, the results in Part C show the 95% confidence interval. The measurement strategies used in Part C are discussed in the following section.

TABLE I
SENSORY NEURONS AND THE RANGE OF VALUES THEY PRODUCE

Neuron	Senses	Range
S_1	Whether the left motor is moving or not.	0 → stopped 100 → moving backwards 200 → moving forwards
S_2	Whether the right motor is moving or not	0 → stopped 100 → moving backwards 200 → moving forwards
S_3	Rotation of left motor in degrees	Integers (0, 360]
S_4	Rotation of right motor in degrees	Integers (0, 360]
S_5	Acceleration in the x-dimension	Integers (0, 255)
S_6	Acceleration in the y-dimension	Integers (0, 255)
S_7	Acceleration in the z-dimension	Integers (0, 255)
S_8	Tilt in the x-dimension	Integers (0, 255)
S_9	Tilt in the y-dimension	Integers (0, 255)
S_{10}	Tilt in the z-dimension	Integers (0, 255)

TABLE II
ACTIVATION NEURONS

Neuron	Description
A_1	Left motor forwards
A_2	Left motor backwards
A_3	Stop left motor
A_4	Right motor forwards
A_5	Right motor backwards
A_6	Stop right motor

TABLE III
MODEL PARAMETERS AND THEIR EXPERIMENTAL VALUES

Parameter	Description	Value
η	SART learning rate	0.1
ρ	SART Validation threshold	0.2
β	Q-learning rate	0.9
β_1	Q-learning rate (competence motivation)	0.9
β_2	Q-forgetting rate (competence motivation)	0.1
γ	Q-learning discount factor	0.9
α	Novelty recovery rate	1.05
τ_1	Novelty decrease rate for winning observation neurons	3.3
τ_2	Novelty increase rate for losing observation neurons	14.3
F^+_{min}	Minimum novelty to receive positive interest	0.5
F^-_{min}	Minimum novelty to receive negative interest	1.5
F^+_{max}	Maximum positive interest feedback	1
F^-_{max}	Maximum negative interest feedback	1
F^+_{avn}	Aversion to low novelty	0.33
F^-_{avn}	Aversion to high novelty	0
ε	Competence motivation switch threshold	0.01

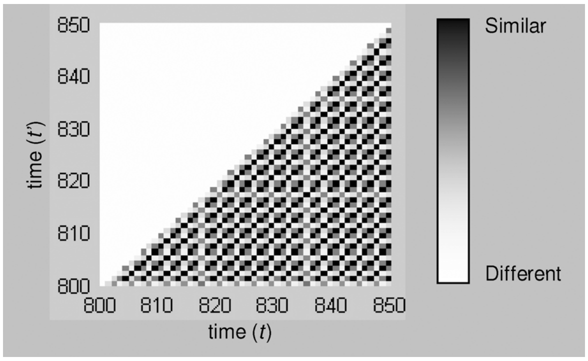

Fig. 11. Point-cloud matrix for a fragment of robot data. Diagonals indicate cycle behavior. Image from [30].

B. Measurement Strategy

This paper uses a number of existing, generic metrics [30] for MRL as well as one metric specific to the integrated approach presented in this paper. These approaches are summarized here.

1) Posture and Point Cloud Matrices: A robot's posture at any time t can be characterized by its sensory data $S_{1(t)}, S_{2(t)}, S_{3(t)}, \ldots, S_{n(t)}, \ldots$. Using such an attribute-based state representation, a point-cloud matrix can be constructed to visualize a robot's behavior by computing the Euclidean distance d between pairs of postures at all times t and t'. That is

$$d = \sqrt{\sum_n \left(S_{n(t)} - S_{n(t')}\right)^2}.$$

The intensity of a pixel (t, t') on the point-cloud diagram is determined by d. A darker color indicates more similar postures as shown in Fig. 11. Dark diagonals on the point-cloud matrix indicate that the robot is cycling through a sequence of similar postures.

Cycles are an important behavioral structure for both natural [46] and artificial systems [30]. As such they provide an approach to evaluating emergent, developmental behavior.

2) Identifying Cyclic Behavior: In Fig. 11, cycles can be identified by analyzing unbroken sequences of "dark" pixels that form diagonals. A "dark" pixel is defined as one where $d < \rho'$. Formally, using a point-cloud matrix, a behavior cycle B is a sequence of posture-pairs (t, t') such that for all $t_1 \leq t \leq t_2$, $d < \rho'$ and $t' = t - N_B$. N_B is the length of the cycle. The duration of the cycle is $D_B = t_2 - t_1 + 1$. The sequence of posture-pairs must be repeated in its entirety at least once (i.e., $D_B > N_B$) and cannot be a multiple of another shorter cycle (i.e., there is no $N < N_B$ for which $\text{dist}(S_{(t)}, S_{(t')}) < \rho'$ and $t' = t - N$ for all $t_1 \leq t \leq t_2$). $\rho' = 0.1$ is used in this paper.

3) Behavioral Stability: The stability ζ of a robot's behavior over a time period of length T is the total duration of all behavior cycles in the period, divided by the length of the period:

$$\zeta_T = \frac{\sum\limits_{B \in T} D_B}{T}.$$

This gives a stability value normalized between zero (fewer cycles/shorter durations) and one (many cycles/longer durations). A higher stability value is generally desirable as it

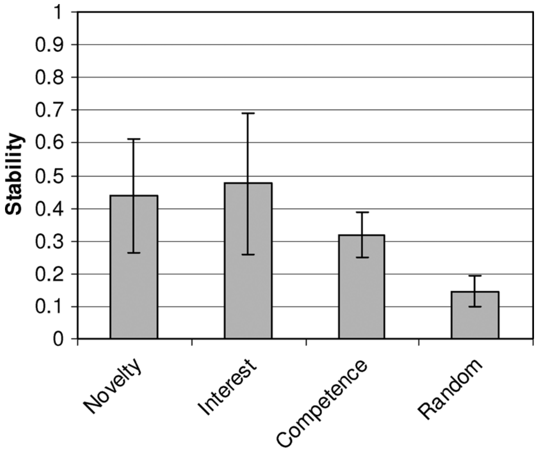

Fig. 12. Average behavioral stability attained by robots using each value system after 4000 time steps (30 min).

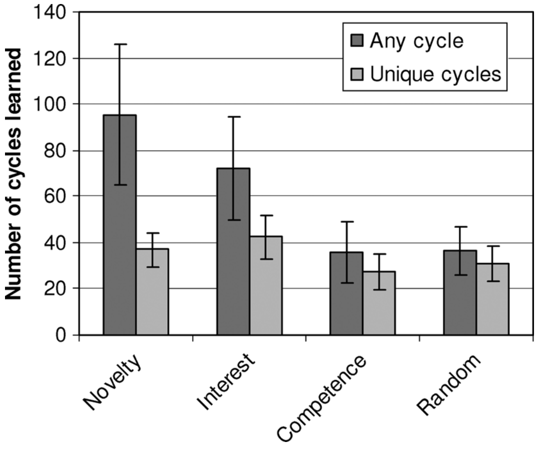

Fig. 13. Average number of behavior cycles learned by robots using each of the value systems.

indicates that the robot is exploiting learned behavioral cycles more productively than a robot with a lower stability value.

4) Exploration: A measure of the robot's capacity for exploration can also be obtained by analyzing the number of observation neurons created. The creation of more observation neurons suggests that the robot has explored more of its environment.

C. Results and Discussion

Fig. 12 shows stability values for robots using the novelty-seeking, interest-seeking, competence-seeking, and randomized value systems. The first important result evident from this chart is that the robots using the three motivation functions (novelty, interest, and competence) show behavior that is significantly more stable than that exhibited by the robot using random exploration.

While the behavioral stability of the robots using the motivation-based value systems is significantly higher than the robot using random exploration, Fig. 12 shows that the stability results for the three motivated robots are still relatively low. Results indicate that all of these robots spend, on average, more than 50% of their lifetime exploring rather than exploiting learned behavior cycles. In addition, no statistical difference can be claimed between the three motivated techniques in terms of their behavioral stability.

Further analysis of the behavior of the robots, does however, show some significant differences. The similar behavioral stability of the three motivated approaches indicates that robots using these algorithms spend similar proportions of their lifetime exploiting learned behavior. However, Fig. 13 and Fig. 15. show that this exploitation is distributed differently throughout the robots' lifetimes. Robots using the novelty and interest-based approaches tend to exploit learned behavior cycles for short periods and return to the same behaviors up to three times during their life. In contrast, robots using the competence-based motivation function exploit learned behaviors for longer periods. These robots tend to focus attention on each behavior cycle only once.

Fig. 13 shows that the difference between the number of unique cycles learned by each of the robots is statistically ambiguous. That is, we cannot claim a statistical difference in the number of unique cycles learned. This is not the case, however, if an interest function without an aversion to low novelty is used. If such an interest function, shown in Fig. 16, is used, a robot will maintain its starting posture for the duration of its life. That is, it will never move. This results in a very high behavioral stability but a very low number of behaviors learned (one) of very short length (one posture long only). Stability values should thus not be considered in isolation, but rather in conjunction with statistics describing the number and length of cycles. This gives an additional indication of the variety and complexity of the robot's behavior.

When a robot has no aversion to low novelty, interest in the starting posture will drop to a very small positive number, but this is still enough for the posture to be reinforced in the learning layer and remain more interesting than other unexplored postures with weights initialized at zero. Without a secondary exploration function, such as e-greedy exploration, the robot will never explore other postures.

The use of the aversion constant is more desirable than using a secondary randomized exploration algorithm as it provides a structured approach to exploration, based on motivational theory.

Fig. 14 illustrates the other main difference between the value systems described in Section III. Fig. 14 shows that robots using random exploration tend to learn cycles of length one action the most. That is, they tend to learn cycles that maintain a given posture for some time. Generally, this happens when the robot randomly generates a sequence of relatively high, positive reward values when performing a stop-motor action.

In contrast, robots using interest and competence motivation learn the most cycles of length two actions. Robots motivated to seek novelty learn the most cycles of length three actions.

In general, Fig. 14 indicates that the robots learn very simple behaviors. Inspection of the log files for the runs shows that these include lifting and lowering their legs. Generally these behaviors involve stopping one motor then moving the other motor backwards and forwards in some sequence. The emergence of simple structured behaviors is important, but indicates that the limits of these motivation functions and basic MRL is being reached, even on this relatively simple robot structure.

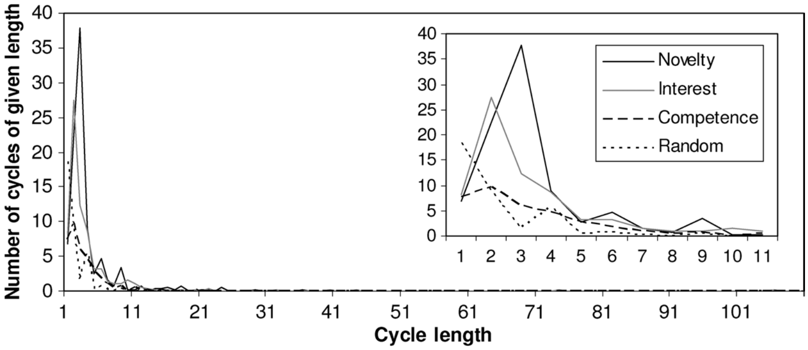

Fig. 14. Average number of cycles learned of a given length by robots using each of the value systems.

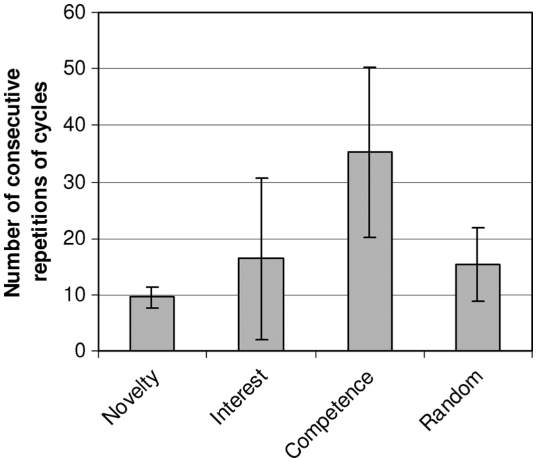

Fig. 15. Average number of consecutive repetitions of a behavior cycle.

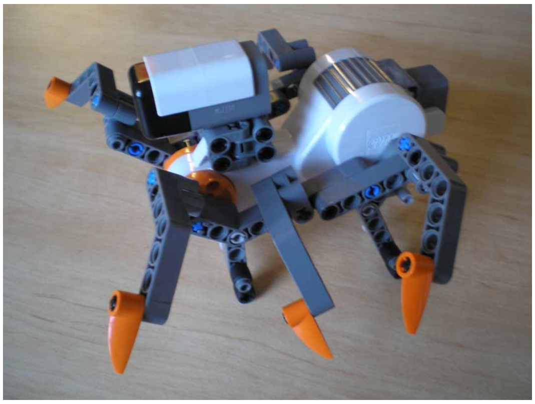

Fig. 17. An "Ant" robot using the *Lego Mindstorms NXT* platform. This robot has one motor controlling all legs and an accelerometer. Image from [30].

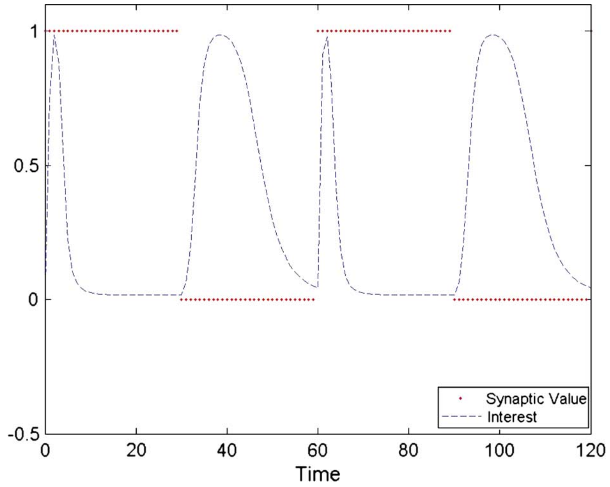

Fig. 16. Change in interest over time without an aversion to low novelty.

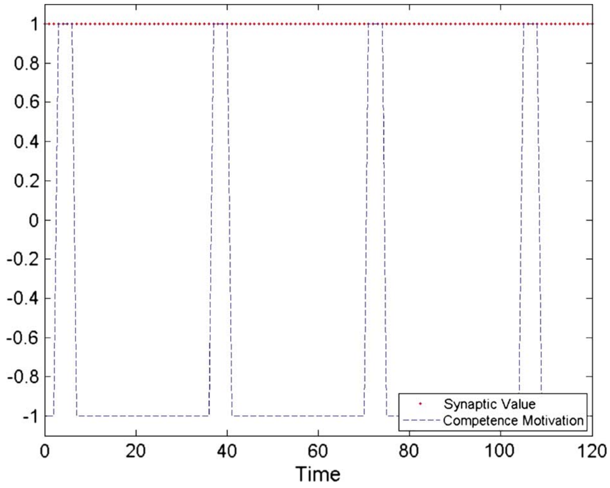

Fig. 18. Change in competence motivation over time with optimistic initialization of utility values.

Fig. 14 does show the emergence of some longer behavior cycles, particularly by the interest-motivated agents. These cycles of up to 106 postures are generally repeated 2–3 times, but have no clear function in practice.

While the emergence of structured behavior cycles, including long cycles, is encouraging because it shows that structure can be generated using generic motivation functions, the cycles emerging in these experiments are not particularly compelling. None of the algorithms was able to motivate the Crab robot to learn to walk, for example. This is disappointing as a variation on the competence-based technique [30] has been shown to produce an emergent walking behavior in the simpler "Ant" robot shown in Fig. 17. This "Ant" robot is quite similar to the Crab, but all legs are controlled with a single motor. This robot, when motivated specifically to learn behavior cycles, is able to learn a ten-posture long behavior cycle for walking. This variation on competence-seeking behavior is shown in Fig. 18. In this variant, Q-values are initialized to one. This means that the robot initially believes itself to be universally competent. It then progressively reduces its prediction of its

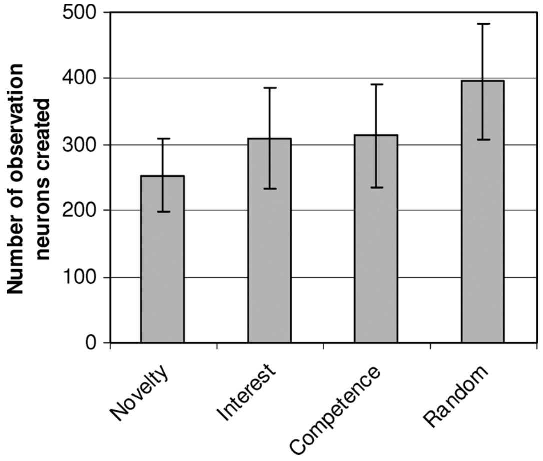

Fig. 19. Average number of observation neurons created by robots using each of the value systems.

own competence. This approach causes the robot to explore rapidly early in its life to reduce the large error in its perception of its competence. In addition, a negative reward (not shown in Fig. 18.) is assigned to actions that cause the robot to maintain its previous posture. This further motivates the robot towards behaviors that cause change, such as walking.

The problem with this approach on the more complex Crab robot is that behavioral stability is very low early in the robot's life because exploration is so high. The Crab using such a motivation function creates around 600 observation neurons. In contrast, Fig. 19 shows that robots using the novelty, interest and competence based approaches described in this paper generate only around 300 observation neurons. Because their focus of attention is narrower, they learn more quickly.

In reinforcement learning terms, several hundred states should describe a plausible learning problem. However, in practice none of the value systems evaluated in this paper focus attention consistently or coherently enough for compelling, functional behavior cycles (such as walking) to emerge.

The overall similarity in performance of the novelty and interest-based approaches tested in this paper is also noteworthy because interest-based approaches are often favored over novelty-seeking approaches for noisy applications such as robots. This is because novelty-based approaches are believed to be weaker than interest-based approaches in the presence of noise. Random stimuli such as sensor noise, from which little can be learned, tend to be unfamiliar and thus generate high novelty values. In practice, however, it would appear that this weakness is not apparent in a MRL setting because random stimuli by nature do not appear consistently enough to be reinforced. In other words, the frequency of stimuli is as important as their familiarity for determining novelty-based reward.

V. Conclusion and Future Directions

This paper makes a step towards a generic platform for the integration, and potential combination, of different cognitive motivation functions with reinforcement learning. The proposed architecture includes the following:

- a uniform notation for both the value system and the learning module, based on the idea of an artificial neuron;
- a combined memory model, in the form of observation neurons, for the value system and learning components, so the robot has a single, consistent, shared representation of long-term memory;
- the capacity for exemplar learning (equivalent to a table-based approach) or function approximation.

This paper makes two main theoretical contributions: (1) adaptation of neural network notation to describe motivated reinforcement learning problems; and (2) translation of three motivation functions and the Q-learning update and action-selection functions into the new notation. The paper also presents an empirical evaluation of four value functions within the proposed framework, including three based on computational models of motivation—novelty, interest and competence—without the need for secondary random exploration functions.

Results show the following:

- the three motivation-based approaches outperform random exploration for generating behavior comprising stable cycles;
- robots using the novelty and interest-based approaches tend to exploit learned behavior cycles for multiple short periods, but robots using the competence-based motivation function exploit learned behaviors for fewer, longer periods;
- the motivation-based approaches inspire longer behavior cycles than random exploration;
- novelty-based reward is at least as effective as interest-based reward, even in an application with high sensor noise.

The emergence of structured behavior cycles is encouraging because it shows that structure can be generated using generic motivation functions, but the value functions compared in this paper demonstrate only modest success in motivating the formation of compelling, functional, cyclic behavior. Overall, the experiments in this paper suggest that the limits of these simple value systems and MRL variants are reached by the Crab robot platform. While the novelty-, interest-, and competence-based motivation functions are capable of motivating structured behavior, there is a need for more expressive MRL architectures to permit the emergence of more complex and compelling behavior. The integrated architecture presented in this paper provides a basis for such architectures. Two variants are discussed in the following sections.

A. Hierarchical Learning Models

The basic architecture described in Section III has capacity for remembering only a single behavioral policy representing one type of behavior cycle. While this means that the robot's behavior is always adapting, in practice it is likely that robots will need to be able to remember and reuse learned behaviors [47]. The integrated model can be adapted to include multiple policies or options [48] using parallel reinforcement learning layers. This is shown in Fig. 20.

The approach in Fig. 20 suggests a number of new ways that motivation can be considered. In particular, motivation may not only act as a reward signal for learning, but it may act as a trigger for creating options by freezing and duplicating the weights in the current reinforcement learning layer, forgetting options, or for activating a previously learned option.

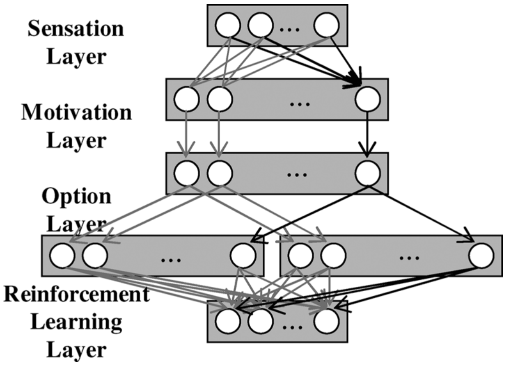

Fig. 20. Integrated approach using parallel reinforcement learning layers to remember more than one learned behavior cycle.

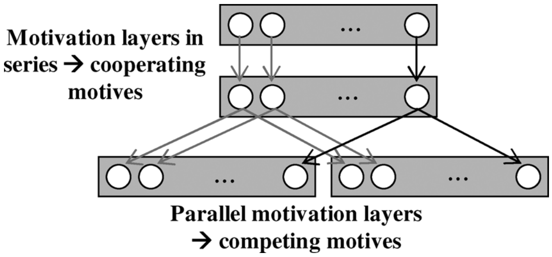
Motivation layers in series → cooperating motives
Parallel motivation layers → competing motives
...

Fig. 21. Multiple motivation layers may be combined in series or in parallel.

When the robot has integrated memory of a number of behavior cycles as options, motivation can also play a role in reasoning about potential highly motivating situations the robot may strive towards, beyond those that it has actually encountered. Existing approaches to MRL assume that the robot must first experience a particular situation before it can motivate itself to achieve that situation. That is, motivation is a function only of the concrete experiences the robot has of its environment, such as states, observations or events. However, in robots with an integrated memory of their current skill set, the motivation function can potentially reason about and generate goals for things the robot "might" be able to do, based on its previous experiences of things it "can" do. For example, the robot may generate a goal to achieve a sequence of observations that have previously only been experienced individually.

This has a number of implications. First, that motivation can be a basis for creativity in robots by permitting them to construct models of possible situations they have not actually encountered. This kind of speculative reasoning also provides a way for a motivated robot to actively direct its behavior towards progressively more complex cycles. This is important for building robots that can combine simple skills to develop more complex ones.

B. Combined Motivation Models

The experiments in this paper also use only one or two motivation functions in each robot. The novelty-seeking robot, for example used a single motivation layer. The interest-seeking robot used two motivation layers in series. The framework, however, is general enough to permit the use of an arbitrary number of motivation modules, either in series or in parallel, as shown in Fig. 21. This provides a basis for future development of simple, combined value systems in which different kinds of motives can cooperate or compete to control the robot. In particular, this model may be used to ground the knowledge of the robot. Grounding is one of the main principles that helps structure behavior, by permitting individual interpretation of the effect of behavior, One of the problems artificial robots is that they lack internal physiology, unless it is explicitly modeled. As a result, they do not have internal drives instructing them to act to maintain their internal resources within the range tolerable to remain alive. The behavioral structure of robots such as the Crab or Ant reflects this absence, since the effect of their behavior does not affect their internal state, except in terms of the custom designed motivation function.

Although artificial robots lack physiology, it is still possible to inspire an inner–outer world relationship to provide structure to behavior by explicitly modeling relevant internal variables such as energy, heat, mechanical balance and so on. These can be modeled in parallel motivation layers, as shown in Fig. 21, so that biological motivations can compete with cognitive motives to control the robot. This is likely to be fundamental for a successful scale-up of the models presented in this paper.

References

[1] P.-Y. Oudeyer, F. Kaplan, and V. V. Hafner, "Intrinsic motivation systems for autonomous mental development," *IEEE Trans. Evol. Computat.*, vol. 11, pp. 265–286, April 2007, 2007.

[2] K. Merrick and M. L. Maher, *Motivated Reinforcement Learning: Curious Characters for Multiuser Games*. Berlin: Springer, 2009.

[3] S. Marsland, U. Nehmzow, and J. Shapiro, "A real-time novelty detector for a mobile robot," in *Proc. EUREL European Adv. Robot. Syst. Master Class Conf.*, 2000.

[4] O. Sporns, N. Almassy, and G. Edelman, "Plasticity in value systems and its role in adaptive behaviour," *Adapt. Behav.*, vol. 8, pp. 129–148, 2000.

[5] K. Friston, G. Tononi, G. Reeke, O. Sporns, and G. Edelman, "Value-dependent selection in the brain: Simulation in a synthetic neural model," *Neuroscience*, vol. 59, pp. 229–243, 1994.

[6] X. Huang and J. Weng, "Inherent value systems for autonomous mental development," *Int. J. Humanoid Robot.*, vol. 4, pp. 407–433, 2007.

[7] J. Schmidhuber, "Curious model building control systems," in *Proc. Int. Joint Conf. Artif. Neural Netw.*, Singapore, 1991, pp. 1458–1463.

[8] R. Saunders, "Curious Design Agents and Artificial Creativity," Ph.D. dissertation, University of Sydney, Sydney, Australia, 2001.

[9] S. Singh, A. G. Barto, and N. Chentanez, "Intrinsically motivated reinforcement learning," in *Proc. Adv. Neural Inf. Processing Syst. 17 (NIPS)*, 2005, pp. 1281–1288.

[10] K. Merrick, M. L. Maher, and R. Saunders, "Achieving adaptable behaviour in intelligent rooms using curious supervised learning agents," in *Proc. CAADRiA 2008 Beyond Computer Aided Design*, Chiang Mai, Thailand, 2008, pp. 185–192.

[11] M. Lungarella, G. Metta, R. Pfeifer, and G. Sandini, "Developmental robotics: A survey," *Connection Sci.*, vol. 15, pp. 151–190, Dec. 2003.

[12] C. L. Hull, *Principles of Behaviour*. New York: Appleton-Century-Crofts, 1943.

[13] C. L. Hull, *A Behaviour System: An Introduction to Behaviour Theory Concerning the Individual Organism*. New Haven, CT: Yale University Press, 1952.

[14] D. McFarland, *Animal Behaviour*, 3rd ed. London, U.K.: Longman, 1995.

[15] W. Wundt, *Principles of Physiological Psychology*. New York: Macmillan, 1910.

[16] D. E. Berlyne, *Conflict, Arousal and Curiosity*. New York: McGraw-Hill, 1960.

[17] O. Avila-Garcia and L. Canamero, "Comparison of behaviour selection architectures using viability indicators," in *The EPSRC/BBSRC International Workshop on Biologically Inspired Robotics: The Legacy of W. Grey Walter*, Bristol, UK, 2002, pp. 86–93, HP Labs.

[18] L. Canamero, "Modelling motivations and emotions as a basis for intelligent behaviour," in *Proc. First Int. Symp. Autonom. Agents*, New York, 1997, pp. 148–155.

[19] C. Gershenson, “Artificial societies of intelligent agents,” in *Engineering*. Mexico: Fundacion Arturo Rosenblueth, 2001.

[20] D. E. Berlyne, “Exploration and curiosity,” *Science*, vol. 153, pp. 25–33, 1966.

[21] D. G. Mook, *Motivation: The Organisation of Action*, 1st ed. New York: W. W. Norton, 1987.

[22] E. C. Tolman, *Purposive Behaviour in Animals and Men*. New York: Century, 1932.

[23] J. W. Atkinson and N. T. Feather, *A Theory of Achievement Motivation*. New York: Wiley, 1966.

[24] F. Heider, *The Psychology of Interpersonal Relations*. New York: Wiley, 1958.

[25] E. Deci and R. Ryan, *Intrinsic Motivation and Self-Determination in Human Behaviour*. New York: Plenum Press, 1985.

[26] J. Marshall, D. Blank, and L. Meeden, “An emergent framework for self-motivation in developmental robotics,” in *Proc. Third Int. Conf. Develop. Learning*, San Diego, CA, 2004, pp. 104–111.

[27] S. Thrun, “Exploration in active learning,” in *Handbook of Brain Science an Neural Networks*. Cambridge, MA: MIT Press, 1995.

[28] F. Kaplan and P.-Y. Oudeyer, “Motivational principles for visual know-how development,” in *Proc. 3rd Int. Workshop Epigenetic Robot.: Modelling Cogn. Develop. Robot. Syst.*, 2003, pp. 73–80, Lund University Cognitive Studies.

[29] J. Herrmann, K. Pawelzik, and T. Geisel, “Learning predictive representations,” *Neurocomputing*, vol. 32–33, pp. 785–791, 2000.

[30] K. Merrick, “Modeling behavior cycles as a value system for developmental robots,” *Adapt. Behav.*, 2010, to be published.

[31] R. Saunders and J. S. Gero, “Curious agents and situated design evaluations,” in *Agents in Design*, J. S. Gero and F. M. T. Brazier, Eds. Sydney, Australia: University of Sydney, 2002, Key Centre of Design Computing and Cognition, pp. 133–149.

[32] J. Schmidhuber, “A possibility for implementing curiosity and boredom in model-building neural controllers,” in *Proc. Int. Conf. Simul. Adapt. Behav.: From Animals to Animats*, 1991, pp. 222–227.

[33] D. Lenat, “AM: An artificial intelligence approach to discovery in mathematics,” in *Computer Science*. Stanford, CA: Stanford University, 1976.

[34] M. Csikszentmihalyi, *Creativity: Flow and the Psychology of Discovery and Invention*. New York: Harper Collins, 1996.

[35] C. Darwin, The Origin of the Species 1859.

[36] R. Saunders and J. S. Gero, “The digital clockwork muse: A computational model of aesthetic evolution,” in *Proc. AISB’01 Symp. AI Creativity Arts Sci., SSAISB*, 2001.

[37] S. Singh, R. L. Lewis, A. G. Barto, and J. Sorg, “Intrinsically motivated reinforcement learning: An evolutionary perspective,” *IEEE Trans. Autonom. Mental Develop.*, vol. 2, Special Issue on Active Learning and Intrinsically Motivated Exploration in Robots, no. 2, pp. 70–82, Jun. 2010.

[38] A. Maslow, *Motivation and Personality*. New York: Harper Collins, 1954.

[39] C. Alderfer, *Existence, Relatedness and Growth*. New York: Free Press, 1972.

[40] R. Stagner, “Homeostasis, discrepancy, dissonance: A theory of motives and motivation,” *Motiv. Emotion*, vol. 1, pp. 103–138, 1977.

[41] C. Watkins and P. Dayan, “Q-learning,” *Machine Learn.*, vol. 8, pp. 279–292, 1992.

[42] T. Kohonen, *Self-Organisation and Associative Memory*. Berlin, Germany: Springer, 1993.

[43] A. Baraldi and E. Alpaydin, Simplified ART: A New Class of ART Algorithms International Computer Science Institute, Berkley, CA, Tech. Rep., TR 98-0041998.

[44] B. Fritzke, “Incremental learning of local linear mappings,” in *Proc. Int. Conf. Artif. Neural Netw.*, 1995, pp. 217–222.

[45] J. C. Stanley, “Computer simulation of a model of habituation,” *Nature*, vol. 261, pp. 146–148, 1976.

[46] A. Ahlgren and F. Halberg, *Cycles of Nature: An Introduction to Biological Rhythms*. Washington, DC: National Teachers Association, 1990.

[47] C. Vigorito and A. G. Barto, “Intrinsically motivated hierarchical skill learning in structured environments,” *IEEE Trans. Autonom. Mental Develop.*, vol. 2, Special Issue on Active Learning and Intrinsically Motivated Exploration in Robots, no. 2, pp. 132–143, Jun. 2010.

[48] R. Sutton, D. Precup, and S. Singh, “Between MDPs and semi-MDPs: A framework for temporal abstraction in reinforcement learning,” *Artif. Intell.*, vol. 112, pp. 181–211, 1998.

Kathryn Elizabeth Merrick received the Bachelor of Computer Science and Technology degree (with Advanced, Honours I, University Medal) from the University of Sydney, NSW, Australia, in 2002 and the Ph.D. degree in computer science from the National ICT Australia and University of Sydney, NSW, Australia, in 2007.

Currently, she is a lecturer in information systems at the University of New South Wales, Australian Defence Force Academy, Canberra, ACT. Her research interests lie in the broad areas of artificial intelligence and machine learning with applications in virtual characters, robotics, intelligent environments, and network security. Her research is principally concerned with the development of algorithms for self-motivated learning agents. She is coauthor of the book *Motivated Reinforcement Learning: Curious Characters for Multiuser Games* (Berlin, Germany: Springer-Verlag, 2009) and over 20 refereed conference and journal papers.