Attention Focus in Curious, Reconfigurable Robots

Kathryn Merrick
School of Information Technology and Electrical Engineering
University of New South Wales
Australian Defence Force Academy
kathryn.merrick@adfa.edu.au

Elanor Huntington
School of Information Technology and Electrical Engineering
University of New South Wales
Australian Defence Force Academy
elanor.huntington@adfa.edu.au

Abstract

The study of computational models of motivation – such as curiosity and interest – has opened the way for new types of artificial agents that can self-select their own goals and focus of attention. In robotic systems, curious agents that can select their own goals have a range of potential applications. These include support for tool use, fault tolerance and robot reconfigurability. This paper considers the design of curious, reconfigurable robots that can explore new behaviour in response to changes in their physical structure. A computational model of curiosity is combined with neural-network reinforcement learning to create curious, reconfigurable robots on the *Lego Mindstorms NXT* platform. Results show that the curious robot can shift its attention focus to learn new behaviours in response to changes in its structure.

1 Reconfigurable Robots

Reconfigurable robots comprise sets of modules that can be re-arranged to achieve different structures, behaviours and functions. Reconfiguration may be instigated by an entity external to the robot, or as a result of internally motivated re-structure [Jantapremijit and Austin, 2001; Unsal and Khosla, 2000; Yim et al., 2000].

Reconfigurable robots promise future functional, economic and creative advantages. Functionally, reconfigurable robots permit engineering versatility, flexibility, robustness and the ability to self-repair through redundancy [Yim et al., 2000]. Economic advantage can be gained by designing complex machines as reconfigurable sets of modules. Likewise, self-reconfigurable robots have the ability to adapt their structure in response to unexpected changes in their environment. This makes them potentially more flexible than robots with a fixed physical structure. From a creative viewpoint, reconfigurable robot technologies such as *Lego*[1] and *Meccano*[2] can encourage creative play, creative thinking and creative design [Merrick, 2008].

The remainder of this section reviews the literature describing reconfigurable robots and identifies a need for new behavioural algorithms to control behaviour after reconfiguration. Section 2 presents an architecture for curious, reconfigurable robots using motivated, neural-network reinforcement learning. This is implemented on the *Lego Mindstorms NXT* platform. In Section 3 we discuss how the behaviour of a curious robot can be characterised using a type of bifurcation diagram to study their shifting attention focus in response to changes in their structure. This is followed by a demonstration and brief analysis of the behaviour of a curious reconfigurable robot in Section 4, using this metric. The paper concludes in Section 5 with a discussion of the lessons learned and directions for future work.

1.1 Tool Using Robots

Tool use can be thought of as a specific way that robots can either be reconfigured or can self-reconfigure. In the first case, robots may be provided with tools to perform a particular task. In the second case, robots may self-select and learn about objects that can extend their set of sensors and actuators, and thus the sets of problems they can solve. Existing work with tool use in robots has considered postures for using and grasping tools [Wood et al., 2005], exploratory learning [Stoytchev, 2005] and learning tool manuipulation models with STRIPS planning [Brown and Sammut, 2007].

While tool use can be thought of as a form of recofiguration, much existing work with tool using robots has focused on using tools to achieve certain predefined goals. In contrast, the work in this paper looks at ways that robots can self-select learning goals using curiosity.

1.2 Fault Tolerant Robots

Another area of robotic research related to reconfiguration is the design of fault tolerant robots. Fault tolerant robots

are engineered systems that are resilient to damage [Bongard et al., 2006; Mahdavi and Bentley, 2006]. For example, Bongard et al. [2006] describe a model for a legged robot that can recover from damage autonomously by continuous self modelling. The robot infers a model of its own structure then uses the model to generate forward locomotion. If damaged, the robot can adapt its inferred model and generate alternative gaits.

While fault tolerance is an important property of robots, approaches to fault tolerance do not generally extend to adaptation to additive reconfiguration of a robot's structure. This is because fault tolerant reasoning processesses have tended to be limited to reasoning about damaging subtractive variations of the robot's original structure. In contrast, the model for curious robots presented in this paper provides a way for agents to explore addative changes to their structure, as well as unexpected changes in their environment.

1.3 Curious Robots

Existing research has focused on developing hardware modules for reconfigurable and self-reconfigurable robots, and the software required for those modules to communicate [Yim et al., 2000; Jantapremijit and Austin, 2001; Unsal and Khosla, 2000]. However, the problem of attributing behaviour to reconfigurable robots remains a challenge. Traditional artificial intelligence techniques tend to assume a fixed set of inputs and outputs (sensors and actuators) and a fixed set of goals based on these inputs and outputs [Russel and Norvig, 1995]. This is in contrast to the needs of reconfigurable robots, which may have changing sensors and actuators and thus changing goals. In addition, for robots engaged in life-long learning, or robots in environments that humans cannot yet go, it may be infeasible to predict or predefine all goals the robot may need to achieve.

Recent work has focused on the design of developmental learning algorithms that can generate goals online in response to the changing experiences of a robot. These approaches uses computational models of novelty [Marsland, 2000] or intelligent, adaptive, curiosity [Kaplan and Oudeyer, 2003; Oudeyer and Kaplan, 2004] to develop new goals. In Marsland's [2000] model the search for novelty drives a simultaneous localisation and mapping algorithm. In Oudeyer and Kaplan's [2003; 2004] work, curious goals are achieved using a reward based learning module. While these approaches permit robots to adapt to unexpected changes in their environment, they do not provide a way for robots to adapt to changes in their sensors or actuators or other physical characteristics.

In other work Merrick and Maher [2007] proposed motivated reinforcement learning (MRL) as an approach to the design of adaptive virtual agents. MRL agents use context-free grammars (CFGs) [Merceron, 2001] to represent variable-length, attribute based sensation and action spaces. This permits agents to adapt to unexpected changes in their sensors or actuators. Merrick and Maher [2007] studied MRL agents in non-player characters in simulated virtual environments and concluded that structured behaviour can emerge from the combination of curiosity and reinforcement learning.

This paper extends MRL to design an architecture for curious, reconfigurable robots that can develop new behaviour as a response to their changing structure. The aim is to achieve robots that are self-motivated to explore changes in their structure and develop new behaviour that is relevant to their current structure. The robot may then use this behaviour to manipulate tools, overcome faults or simply solve problems it computes to be interesting.

2 Architecture for a Curious, Reconfigurable Robot

Existing work with MRL has focused on the design of intrinsic reward systems that interact with flat and hierarchical reinforcement learning [Merrick and Maher, 2007; Barto et al., 2004]. These models use table-based representations to store behavioural policies mapping sensations to actions and utility values. A utility values represents the value the agent places on performing a given action in response to a given sensation. However a table-based representation becomes problematic in continuous, real world domains. In this type complex environment, function approximation is generally required to limit the time and memory needed for learning.

The architecture presented in this section extends the MRL model from table-based MRL to neural-network MRL. The architecture is based on that proposed by Merrick [2008] and comprises two layers: a device layer and an agent layer. The device layer manages communication between a software agent and a *Lego Mindstorms NXT* robot platform running the Lejos[3] Java firmware. The main difference between the work in this paper and the Merrick [2008] model is that the agent layer uses motivated, neural network reinforcement learning to respond to curiosity. The architecture is shown in Figure 1. While the *Lego Mindstorms NXT* platform is used in this paper, the concept of a reconfigurable robot using curiosity and MRL is general enough to be applied to other reconfigurable robot hardware.

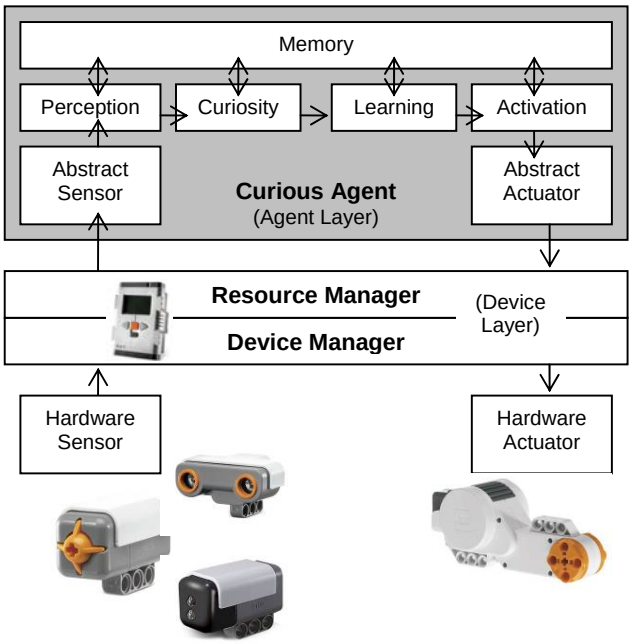

Figure 1. System architecture for a curious, reconfigurable robot. Diagram adapted from [Merrick, 2008].

[1] http://www.lego.com
[2] http://www.meccano.com
[3] http://lejos.sourceforge.net

2.1 Device Layer

Lego Mindstorms NXT robots can be designed with a range of different sensors and actuators, some of which are shown in Figure 1. These include sensors for colour, touch and distance to objects (bottom left) and actuators for servo motors (bottom right). Other sensors include compass sensors, sound sensors, gyroscopic sensors and infrared sensors. NXT sensors and actuators are heterogeneous and return different numbers and types of data. The role of the device layer is thus to construct a standardised, CFG representation of sensor data and available actions and communicates this to the agent layer.

Device Manager

At each time-step t, the device manager identifies the sensor devices S_1, S_2, S_3 … attached to the robot. Data from all sensors is encapsulated as a single sensation **S**=(s_1, s_2, s_3, …) represented as a string from a CFG of the form:

```
S              →    <device_data>
<device_data>  →    <Sᵢdata><device_data> | ε
<Sᵢdata>       →    <s><Sᵢdata> | ε
<s>            →    (<label>:<value>)
<label>        →    A unique identifier
<value>        →    Real number
```

At each time step t, the device manager also identifies the actuators A_1, A_2, A_3 … attached to the robot. A list **A** of available actions from these devices is encapsulated as a string from a CFG of the form:

```
A              →    <actions>
<actions>      →    <AⱼActions><actions> | ε
<AⱼActions>    →    A unique identifier
```

The current, normalised sensation and action set are then sent to the resourc manager.

Resource Manager

The resource manager communicates sensations and actions between the agent and the devices via Bluetooth. In the demonstration in Section 4, the agent layer is run on a computer separate from the NXT brick. The Lejos Java firmware does not currently offer sufficient support for the complex programming constructs required to implement a curious agent. In future it is envisaged that it may be possible to run the agent layer on the NXT brick (or equivalent device) and the resource manager will no longer be required.

2.2 Agent Layer

The basic structure of the agent layer is that of a MRL agent [Merrick and Maher, 2007]. The agent uses a computational model of curiosity as the motivation function. Neural-network reinforcement learning is used as the learning component. The agent has four processes for perception, curiosity, learning and activation. These communicate with the physical sensors and actuators via the device layer. An overview of the agent algorithm is given in Figure 2. Its steps are explained in detail in the following sections.

Abstract Sensor

The abstract sensor communicates with the resource manager via Bluetooth. At each time-step t it receives a sensation **S** and set **A** of available actions. This information is passed to the perception process.

```
Repeat forever:
      Sensors return sensation S
      Perception computes event E
      Curiosity computes reward R
      Learning updates behavior policy
            with S, R and previous A
      Activation selects next action A
      Actuators execute the action
```

Figure 2. Agent layer algorithm for the curious robot.

Perception

The perception process computes an event E=(e_1, e_2, e_3, …) representing the change between the current sensation and the previous sensation stored in memory. Events are also represented as strings from a CFG. The value of each event element e is computed as the real number difference of sensation elements with the same label:

```
E              →    <dev_events>
<dev_events>   →    <SᵢEvents><dev_events> | ε
<SᵢEvent>      →    <e><SᵢEvent> | ε
<e>            →    (<label>:<value>)
<label>        →    A unique identifier
<value>        →    Real number
```

The current sensation is stored in memory and the computed event is passed to the curiosity process.

Curiosity

The curiosity process computes a curiosity value for the current event. Curiosity is computed as a function of the novelty of an event. Novelty is calculated using a real-time novelty detector [Marsland et al., 2000]. The real-time novelty detector classifies events using an Habituated, Self-Organising Map (HSOM). Each neuron in a SOM is connected to an habituating neuron updated according to the equation:

$$\tau \frac{dN_{(t)}}{dt} = \beta\,[N_{(0)} - N_{(t)}] - \varsigma_{(t)}$$

Where $N_{(t)}$ is the novelty of the current stimulus, τ is the rate of habituation and β is the rate of recovery. The Wundt [1910] curve is applied to compute curiosity reward, as proposed by Saunders [2001]:

$$R_{(t)} = \frac{F^{+}_{max}}{1 + e^{-\rho^{+}(2N_{(t)} - F^{+}_{min})}} - \frac{F^{-}_{max}}{1 + e^{-\rho^{-}(2N_{(t)} - F^{-}_{min})}}$$

Figure 3(a) shows how a stimulus event habituates and recovers over time in response to repeated presentation or removal of the event. Figure 3(b) shows that the most curious events are those that are moderately novel in the agent's experience.

The curiosity value $R_{(t)}$ is passed to the learning process as the intrinsic reward value for learning. τ=14.3, β=1.05, $F^{+}_{max} = F^{-}_{max} = 1$, $\rho^{+} = \rho^{-} = 10$, $F^{+}_{min} = 0.5$ and $F^{-}_{min} = 1.5$ are used in this paper.

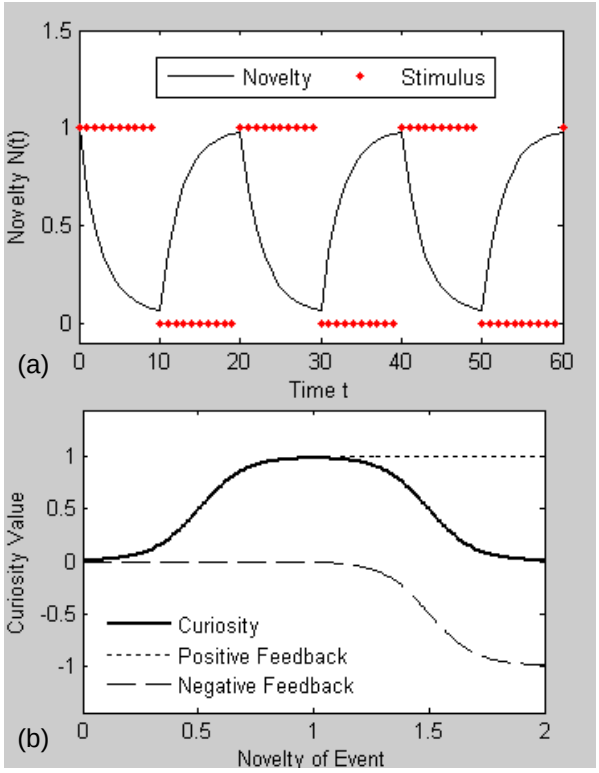

Figure 3. (a) Modelling novelty as habituation and recovery in response to presentation or removal of an event stimulus (b) Modelling curiosity as moderate novelty using the Wundt [1910] curve.

Learning

The learning process uses neural-network Q-learning. While various approaches to function approximation in reinforcement learning exist – including neural network reinforcement learning [Coulom, 2002], decision tree reinforcement learning [Pyaett and Howe, 2001], relational reinforcement learning [Dzeroski et al, 1998] and gradient descent approaches [Aberdeen, 2003] – MRL requires an approach in which the structure representing the learned policy can be developed incrementally. This is required so that the set of sensations and actions does not need to be known prior to learning. Additionally, an approach is required that assumes an attribute based representation of sensations.

In neural-network function approximation for reinforcement learning, the behavioural policy is represented by a neural network mapping sensations to actions. For agents using an attribute based representation of sensations $\mathbf{S}=(s_1, s_2, s_3, \ldots)$ and having a set $\mathbf{A}$ of actions, a network of perceptrons of the form shown in Figure 4 can be used to represent the utility of each action with respect to each sensation. Each sensation element is represented by an input neuron. Each action is represented by an output neuron. Each input neuron is connected to each output neuron by a weight. In our model, initially, the network has no neurons or weights. Input and output neurons are added progressively as the robot is reconfigured and new sensations and actions are encountered. This means that knowledge of the environment or configuration of the robot is not required prior to learning.

The neural network has just one neuron for each element of the sensation, rather than entries for every combination of values for sensation elements needed in a table-based representation. This significantly reduces the memory requirements of learning, while providing a way for the robot to adapt its learned behaviour policy.

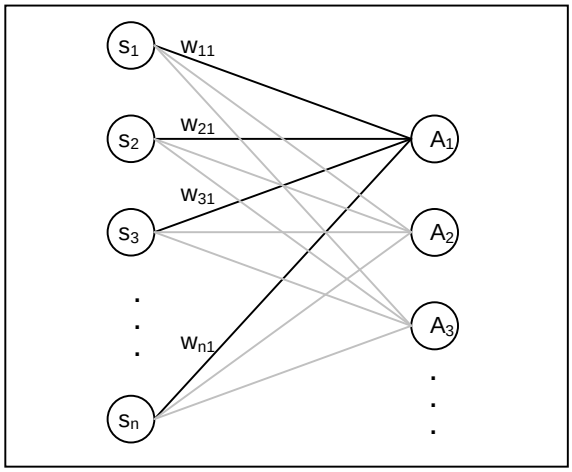

Figure 4. A neural-network is used to represent the learned mapping from sensations to actions and utility values. Input nodes represent elements of sensations. Output nodes represent actions. The network is constructed incrementally in response to the robot's experiences in its environment.

At each time t, the agent receives a sensation $\mathbf{S}_{(t)}$ and a curiosity reward value $R_{(t)}$. The perceptron update rule for neural networks [Russel and Norvig, 1995] can be adapted for the Q-learning [Watkins, 1989] variant of reinforcement learning as follows. First, the predicted utility of the current action $A_{(t)a}$ is computed as:

$$Q(\mathbf{S}_{(t)}, A_{(t)a}) = g(s_1 w_{1a} + s_2 w_{2a} + s_3 w_{3a} + \ldots)$$

$$\text{where } g(in) = \frac{in + 1}{2}$$

In this paper we use a linear activation function, as the utility function to be represented has continuous outputs in the range [0, 1]. This is not accurately represented by a non-linear step function or its sigmoid approximation. Some of the implications and limitations of this kind of perceptron learning are discussed in Section 5.1.

Next, the actual utility $T_{(t)}$ of the current action is computed as:

$$T_{(t)} = R_{(t)} + \gamma \max_a Q(\mathbf{S}_{(t+1)}, A_a)$$

The first term is the current intrinsic reward (curiosity) value $R_{(t)}$. The second term is the expected future reward value. γ is the discount factor for expected future reward. We used γ=0.9 in this paper. The error in prediction is:

$$X_{(t)} = T_{(t)} - Q(\mathbf{S}_{(t)}, A_{(t)a})$$

Finally, the weights connecting the neuron representing the current action to elements of the current sensation is updated to bring the predicted utility closer to the actual utility. The update equation is:

$$w_{(t)na} \leftarrow w_{(t-1)na} + \alpha X_{(t)} g'(in) s_{(t)n} \quad \forall n$$

α is the learning rate of the neural network. We used α=0.5 in this paper. The derivative of the activation is calculated stepwise as:

$$g_{(t)}'(in) = \frac{g_{(t)}(in) - g_{(t-1)}(in)}{in_{(t)} - in_{(t-1)}}$$

Because the agent will compute high curiosity for different events at different times based on its experiences and its current sensors and actuators, the agent will learn different behaviours that are adapted to its current sensors and actuators and current structure.

Activation

The activation process uses an ε-greedy algorithm to select the action with the highest utility value 90% of the time and a random action 10% of the time. This means that the agent follows its learned behavioural policy 90% of the time and experiments 10% of the time. This encourages the robot to explore and adapt to changes in its structure.

Abstract Actuator

The abstract actuator communicates the unique identifier of the action selected by the activation process to the resource manager. The resource manager passes this to the device manager, which in turn triggers the relevant hardware device.

3 Characterising the Behaviour of Curious, Reconfigurable Robots

Previous work with curious MRL agents has characterised their behaviour numerically in terms of the variety and complexity of behaviour cycles learned [Merrick, 2007]. Other work has analysed MRL agents qualitatively using case studies in specific domains [Merrick and Maher, 2007; Merrick, 2008]. However, existing metrics for behavioural variety and complexity rely on the exact repetition of sensations or events in the agent's experience trajectory. This becomes infeasible in robots in real-world environments where actions may result in subtly different sensations or events each time they are executed. In this paper we use a form of bifurcation diagram to study the shifting focus of attention of the robot over time.

3.1 Characterising Attention Focus using Modified Bifurcation Diagrams

A particular point of interest in these experiments is the change in behaviour of the reconfigurable robot when sensors or actuators are added to the system. In the terminology of nonlinear systems analysis, these additions are likely to induce bifurcations or abrupt changes in the dynamics of the system. There exists a suite of standard techniques used to analyse the dynamics of nonlinear systems [Kantz and Schreiber, 2003]. As the name suggests, bifurcation diagrams are particularly well suited to the characterisation of abrupt changes in system dynamics. We thus use this technique to characterise the changing attention focus of the robots in this paper.

Bifurcation diagrams are formally defined as Poincare maps whose fixed points represent limit cycles with a single intersection with the Poincare plane [Kantz and Schreiber, 2003]. In practice, this means that the time series of a system variable is sampled once during a behaviour cycle and that sample plotted in a map. Such a map is generated for each value of a control parameter – a parameter which induces changes in system dyanmics – to produce a bifurcation diagram. As the dynamics of the system changes with the control parameter, so too will the map, thereby allowing analysis of the bifurcations. In our case, the control parameter is time and the system variable under consideration is the action performed by the robot.

The density of points in the map is sufficient to identify equilibria in situations where the system variable under analysis is continuous or quasi-continuous and stochastic variations are relatively small. Such is not the case here where the system variable under consideration (the action) can take on one of a small, discrete set of values. Furthermore, the system under consideration here shows a high degree of intermittency by design, and so equilbria would be difficult to identify in a time-averaged sense. Thus a modified bifurcation diagram is proposed here. The purpose of the modified bifurcation diagram is to afford analysis of the focus of attention rather than an analysis of the topology of equilibria.

Rather than sub-sampling the time series data at the period of a previously identified behaviour cycle, the full set of time-series data is used to generate the modified bifurcation diagram. A histogram of that data is generated for each value of the control variable and the entire data set plotted in a false-colour image, normalised to the largest value in the image. An example is shown in Figure 6. Pixels that appear "hot" (red) in the image represent a strong focus of attention while "cold" (blue) pixels indicate no attention. Thus, the modified bifurcation diagram allows analysis of the focus of attention as a function of the control parameter time.

In the experiments described here, the time-series data were broken into sections of fixed lengths of 200 time steps, and a histogram of the data generated over that section. Subsequent sections are analysed independently to generate the modified bifurcation diagram. The exact choice of the length of the analysis sections was made heuristically for this work.

4 Demonstration of a Curious, Reconfigurable Robot

This section describes a simple experiment to demonstrate and characterise the behaviour of a curious, reconfigurable robot in response to changes in its structure.

4.1 Structure of the Robot

The robot used in demonstration is a jointed arm with the structure shown in Figure 5. The robot was run for 10000 time steps or approximately two hours. During this time its structure was modified several times, without resetting the robot software. Initially the robot was equipped with a compass sensor and turntable controlled by a motor, as shown in Figure 5(a). The robot initially had three actions available, A_1 for moving the motor forwards, A_2 for moving the motor backwards and A_3 for stopping the motor. An example of a sensation received by the agent during this period was:

```
S((degs:178)(speed4:180)(mov4:0))
```

The agent can sense its angle from north, the speed setting of its motor and whether the motor is moving.

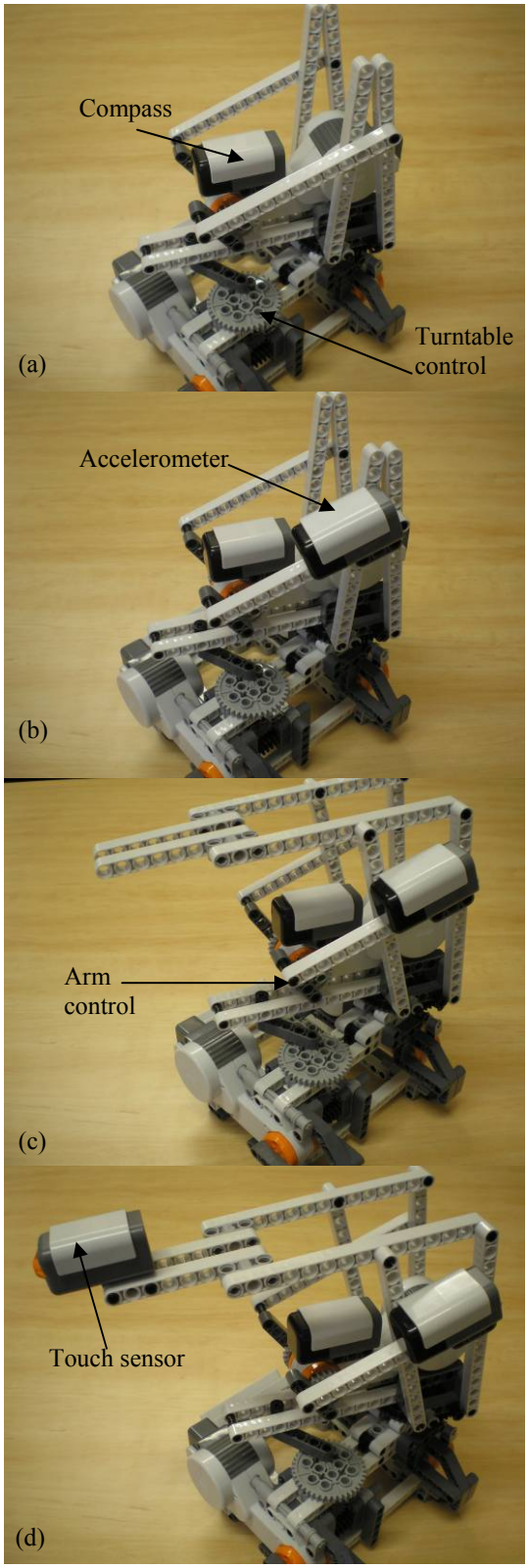

Figure 5. A curious, reconfigurable robot. (a) Structure at t≈0-2000 (b) Structure at t≈2000-4000 (c) Structure at t≈4000-6000 (d) Structure at t≈6000-10000.

After around 20 minutes, an accelerometer was added to the arm as shown in Figure 5(b). An example of a sensation during this period was:

```
S((xa:90)(degs:178)(speed4:180)(xt:22)(za:1
    6)(mov4:0)(zt:4)(ya:0)(yt:0))
```

After a further 20 minutes a motor was connected to the arm, as shown in Figure 5(c). This introduced three new actions: A_4 for moving the second motor forwards, A_5 for moving the second motor backwards and A_6 for stopping the second motor. The final modification was the addition of a touch sensor, as shown in Figure 5(d). An example of a sensation after this modification was:

```
S((xa:92)(speed5:180)(degs:178)(speed4:180)
(ad1:18)(xt:23)(za:17)(mov5:0)(mov4:0)(zt:4
          )(ya:102)(yt:0))
```

One important aspect of the design of this robot is that the turntable and arm are built such that infinite forward or backward motion of the motors results in limited oscillation of the robot arm. This is a requirement for curious robots as there is no intrinsic motivation for the robot to cease exploring its structure beyond the physical limits of the structure. This can cause the robot to damage itself if it not carefully designed.

4.2 Results and Discussion

Figure 6 charts the change in attention focus of the robot against time. In the first 2000 time steps, the focus of attention is shared between A_1, A_2 and A_3, with a slight preference for A_3. This represents the agent experimenting with different actions for moving the turntable and changing the value of the compass sensor. In contrast, for t=2000-4000 the robot displays a clear preference for A_1 in response to the addition of the accelerometer. That is, the agent focuses only on keeping the motor moving forwards as this results in repeated change to the values returned by the compass and accelerometer.

When the additional motor is added at t=4000, initially this motivates a shift in attention focus to a preference for A_2 at t=4000-4500. That is, the reconfiguration has triggered the robot to find another way to achieve changes in the compass and accelerometer readings. This highlights one of the advantages of curious agents, that they can find different solutions to a problem by exploring and focusing attention on different actions or actuators.

Around t=4500 the agent identifies the availablitiy of new actions assoicated with the addition of the second motor. It then begins to experiment with these actions. The red squares on the bifurcation diagram show the agent's focus of attention shifts from A_5 and A_4 (the second motor) back to A_2 (the first motor), then back to A_4 (second motor), then again to A_1 (first motor) and finally between A_5 (second motor), A_1 and A_2 (first motor) and A_5 (second motor) near the end of the period shown. This represents repeated shifts back and forth between curiosity about the new motor controlling the arm and the original motor controlling the turntable. This can be thought of as repeated habituation and recovery in response to the new stimuli introduced by the motor.

Thus, while action-by-action cycles have not been evidient in the behaviour of the curious robots , the results do suggest that some cyclic periods of habituation and recovery are present in the robot's behaviour.

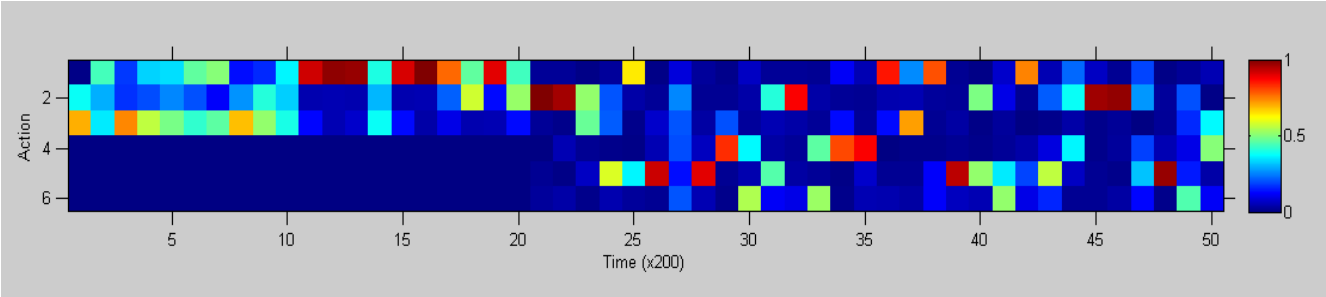

Figure 6. Change in attention focus over time by the curious, reconfigurable robot.

However, further work is required with additional experiments to understand this more clearly. This and other directions for future work are considered in the next section.

5 Conclusion and Future Work

This paper has presented an architecture for curious, reconfigurable robots using motivated, neural-network reinforcement learning. This architecture was implemented on the *Lego Mindstorms NXT* platform. We demonstrated how the behaviour of a curious robot can be characterised using bifurcation diagrams to study their shifting attention focus in response to changes in their structure.

5.1 Strengths and Limitations

This paper has explored an initial demonstration of a curious, reconfigurable robot. Analysis of the behaviour of a curious, reconfigurable robot showed that shifting attention focus is evident as the robot explores new behaviour, both in response to changes in its structure and shifts in its own intrinsic curiosity. Despite this, the robot is limited in the types of behaviour it learns. Essentially, learning is focused on low level joint attention, as the robot builds a model of how different actions change its perception of its environment. Unlike previous work with agents in virtual environments [Merrick and Maher, 2007] where structured behaviour cycles were evident, higher level behaviour that combines basic joint movement into more complex sequences was not evident during this experiment. Further work is required to understand how the curiosity and learning are interacting in this, more complex, environment. For example, changing the way events are represented to limit the number of event elements in a single event may improve the agent's ability to focus on learning to control individual stimuli.

The combination of motivation and neural network reinforcement learning using perceptrons also raises a number of issues. Perceptron learning provided a starting point for function approximation in MRL, where incremental development of the learned policy structure is required. However, neural network reinforcement learning in general suffers from the fact that Q-learning relies on local learning updates, but updates to a neural network can cause non-local changes to the behavioural policy. Our use of a linear activation function in this paper also limits the types of functions and thus the types of behaviour that can be learned by the network. One direction for future work is to consider how other variants of function approximation, including other types of neural networks, can be adapted to the incremental learning required in MRL.

The model presented in this paper is also limited in the way the agent remembers learned behaviour. In this paper, a single neural network is used to represent the agent's behavioural policy. This means that each time the agent shifts its focus of attention to a new behaviour, the old behaviour is forgotten. If the robot were reconfigured into a previous structure, it would have to learn about that structure again. The advantage of the single neural network is that it reduces memory usage and also produces the most adaptable robot as obsolete behaviour is rapidly forgotten. This is useful for robots with structure that is changing rapidly. However, robots that change structure slowly, or return often to old structures, may benefit from a form of hierarcical learning. This is another area for future work.

In Section 4 we also noted that the structure of a curious robot must be carefully designed so that the robot cannot damage itself while it is exploring its structure. Future work may also study other models of motivation, as alternatives to curiosity, to reduce the need for constraints on the robot's structure. This may include biological motivations such as pain when parts of the robot are pushed past their structural limits. Additional sensors that could monitor and detect such behaviour would be required to achieve this.

Finally, the metrics used to evaluate curious robots are also an area for further development. The metrics used in this paper characterise the robot in terms of its ability to focus attention on different actions. This demonstrates the curiosity function as a mechanism for attention focus. However the metric presented in this paper does no measure the variety, complexity or structure of the robot's behaviour. Further work is required, not only to achieve structured behaviour in curious robots, but to provide a way to characterise and measure that structure.

5.2 Future

In summary, this paper has considered the design of curious, reconfigurable robots that can explore new behaviour in response to changes in their physical structure. An initial demonstration and brief analyis was performed.

In the long term, curious agents that can select their own goals may have a range of potential applications in robots beyond reconfigurability. This may include a role in self-reconfigurable robots, tool using and fault tolerant robots. For example, a curiosity module might provide a generic impetus for exploration of objects that

extend the robot's sensory range, and its ability to change its environment, in interesting ways.

References

[Aberdeen, 2003] Aberdeen, D., *Policy gradient algorithms for partially observable Markov Decision Problems*, Australian National University, 2003.

[Barto et al., 2004] Barto, A., Singh, S., Chentanex, N., Intrinsically motivated learning of hierarchical collections of skills, *International Conference on Developmental Learning*, pp 112-119, 2004

[Bongard et al., 2006] Bongard, J., Zykov, V., Lipson, H., Resilient machines through continuous self-modelling. *Science*, 314(5802):1118-1121

[Brown and Sammut, 2007] Brown, S., and Sammut, C., An architecture for tool use and learning in robots. *Australian Conference on Robotics and Automation*, 2007.

[Coulom, 2002] Coulom, R., *Reinforcement Learning Using Neural Networks, with Applications to Motor Control.* PhD Thesis, Institut National Polytechnique de Grenoble, 2002.

[Dzeroski et al., 1998] Relational reinforcement learning. *International Conference on Inductive Logic Programming,* Lecture Notes in Artificial Intelligence, Springer, pp 11-22

[Jantapremjit and Austin, 2001] Design of a modular, self-reconfigurable robot. *Australian Conference on Robotics and Automation*, pp 38-43, 2001.

[Kantz and Schreiber, 2003] *Nonlinear Time Series Analysis*: 2nd Ed., Cambridge University Press, Cambridge, 2003.

[Mahdavi and Bently, 2006] Mahdavi, S. H., Bentley, P. J., Innately adaptive robotics through embodied evolution. *Autonomous Robots*, 20(2):149-163, 2006.

[Marsland et al., 2000] Marsland, S., Nehmzow, U, Shapiro, J.: A real-time novelty detector for a mobile robot, *EUREL European Advanced Robotics Systems Masterclass and Conference*, 2000.

[Merceron, 2001] Merceron, A. Languages and logic. Pearson Education Australia.

[Merrick, 2007] Merrick, K.: (2008) Modelling motivation for adaptive non-player characters in dynamic computer game worlds, *ACM Computers in Entertainment,* Newton Lee (Ed.), Vol 5(4).

[Merrick, 2008] Merrick, K.: Designing toys that come alive: curious robots for creative play, *The Seventh International Conference on Entertainment Computing (ICEC 2008),* September 25-27, Carnegie Mellon University, Springer, pp 149-154.

[Merrick and Maher, 2007] Merrick, K., Maher, M-L.: Motivated reinforcement learning for adaptive characters in open-ended simulation games, *ACM SIGCHI International Conference on Advances in Computer Entertainment Technology*, (ACE 2007), Salzburg, Austria, pp 127-134.

[Oudeyer and Kaplan, 2004]: Intelligent adaptive curiosity: a source of self-development, *In Proceedings of the Fourth International Workshop on Epigenetic Robotics*, pp 127-130, 2004.

[Pyaett and Howe, 2001] Pyaett, L., Howe, A., Decision tree function approximation in reinforcement learning. *In Proceedings of the Third International Symposium on Adaptive Systems: Evolutionary Computation and Probabilistic Graphical Models*, Havana Cuba, pp 70-77.

[Russel and Norvig, 1995] Russel, S., Norvig, P.: *Artificial intelligence a modern approach*, Prentice Hall, 1995.

[Saunders, 2001] Saunders, R., 2001. *Curious design agents and artificial creativity*. PhD Thesis, University of Sydney, Sydney.

[Stoytchev, 2005] Stoytchev, A., Behaviour-grounded representation of tool affordances. *In Proceeings of IEEE International Conference on Robitics and Automation* (ICRA 2005).

[Unsal and Khosla, 2000] Unsal, C., Khosla, P., Mechatronic design of a modular self-reconfigurable robotics system. *Proceedings of the 2000 IEEE International Conference on Intelligent Robots and Systems*, pp 1742-1747, 2000.

[Watkins, 1989] Watkins, C., *Learning from delayed rewards*. PhD Thesis, Cambridge University, Cambridge, England, 1989.

[Wood et al., 2005] Wood, A. B., Horton, T. E., and Amant, R., Effective tool use in a bahile agent. *In Systems and Information Engineering Design Symposium*, pp75-81. IEEE, April 2005.

[Wundt, 1910] Wundt, W., *Principles of physiological psychology*. Macmillan, New York, 1910.

[Yim et al., 2000] Yim, M., Duff, D., Roufas, K.: Polybot: a modular, reconfigurable robot, *In Proceedings of the 2000 IEEE International Conference on Robotics and Automation*, CA, pp 514-520, 2000