Nested Q-Learning of Hierarchical Control Structures

Bruce L. Digney[*]
Defence Research Establishment Suffield

ABSTRACT

The use of externally imposed hierarchical structures to reduce the complexity of learning control is common. However it is clear that the learning of the hierarchical structure by the machine itself is an important step towards more general and less bounded learning. Presented in this paper is a nested Q-learning technique that generates a hierarchical control structure as the robot interacts with its world. These emergent structures combined with learned bottom-up reactive reactions result in a flexible hierarchical control system.

1. Introduction

The use of Q-learning [1] and other related reinforcement learning techniques is common for the control of autonomous robots [2]. However, long learning times combined with the slow speed (when compared to simulations) and the frailties of robot hardware present considerable problems when scaling up to real applications. Often, to reduce the problem to a more tractable size, the control task is hand decomposed into a hierarchical structure [3]. This abstracts the problem into many smaller, more easily learned control problems. However, this hand decomposition imposes the designer's preconceived notions on the robot which, from the robot's own sensors and actuators point of view, may be inefficient or incorrect. Furthermore, it is acknowledged that for truly general learning and full autonomy in the face of unknown and changing environments, the structure of the hierarchical control system must be learned.

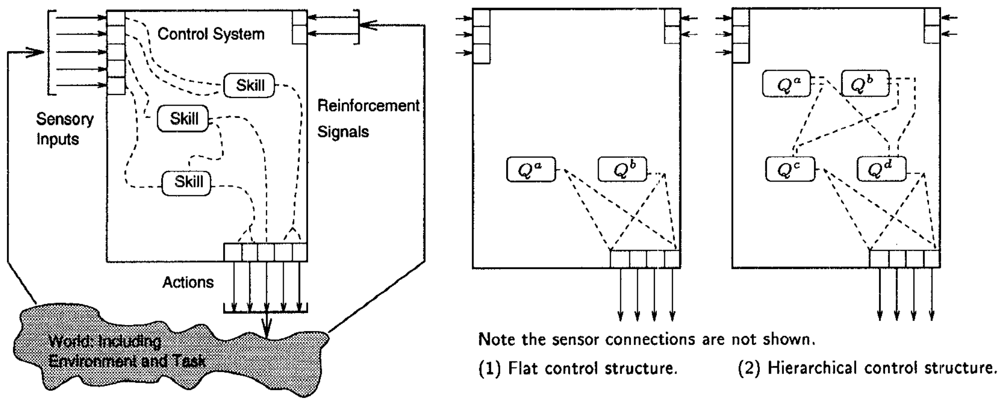
Once the structure has been learned, skills that have been mastered in previous situations can be used in future tasks and environments. This continual carrying forward of learned information is called life long learning and provides the robot a head start at learning new tasks [4] [5]. As the robot moves from task to task and environment to environment it will have the accumulated information of its past experiences available to it as skills. These transportable skills will allow the robot to learn progressively more complex tasks. Eventually this continual learning will allow for the learning of tasks that would be impossible in a simple monolithic network. The use of pre-training nested Q-Learning controlled robots using scaffolding actions and staged learning as well as online skill transfer between task/environment settings is discussed further elsewhere [4] [5].

2. Nested Q-learning for Emergent Control Structures

Consider the example of a robot receiving sensations from its sensors and internal and external reinforcements as pictured schematically in Figure 1 (a). This robot is capable of acting on its world using a number of primitive actuator movements. These primitive actions are at the simplest level of the robot's actuators and although they may utilize feedback control mechanisms, they do not embody any higher intelligence. This robot also receives reinforcement signals which are a critical indicator of how the robot is progressing with respect to the completion of some desired task(s). These critical signals are all the direction that can be assumed for an autonomous robot. The critical error signals simply provide negative reinforcement when the actions of the robot do not achieve the task and positive (favorable) reinforcement whenever the actions achieve the goal.

In the field of intelligent control, the control strategies connecting the sensors to the actions are either hardwired by a designer, they are taught to the agent via a teacher or are left to be learned by the agent. There are many variations of architectures in which the control systems maybe implemented. The two main variations are flat and hierarchical as shown schematically in Figure 1 (b) (1) and (2), respectively. Note that the hierarchical control system relies upon higher level skills being built upon lower level skills while the flat architecture contains only skills at a single level interacting directly with the actuators.

[*] The author may be contacted via Phone (403) 544-4854, FAX (403) 544-4704, E_mail bdigney@dres.dnd.ca or at DRES Box 4000, Medicine Hat, Alberta, CANADA, T1A 8K6. The author acknowledges support from the Canadian Department of National Defence (DND) and the Natural Sciences and Engineering Research Council (NSERC).


Control System
Skill
Skill
Skill
Sensory Inputs
Reinforcement Signals
Actions
World: Including Environment and Task
Q^a
Q^b
Q^a
Q^b
Q^c
Q^d
Note the sensor connections are not shown.
(1) Flat control structure.
(2) Hierarchical control structure.

(a) Control schematic. (b) Flat and Hierachical control structures.

Fig. 1: Schematic of control architectures: (a) sensory, action and reinforcement signal configuration for a learning control system and (b) control architectures: (1) flat structure and (2) hierarchical structure.

A method will now be presented that will allow for the autonomous generation of hierarchical control structures as pictured in Figure 1 (b) (2). Consider the control system schematic of Figure 1 (a). Each incoming sensor has a finite number of perceivable distinct states. These distinct states can be imposed to be evenly distributed or exist as some fuzzy quantity. Eventually, it is planned to have these features refined using some continuous adaption scheme *piggy-backed* upon this structural learning method. In this paper an even distribution of features over the range of the sensor will be assumed. For sensor s_n these features are represented by the distinct values, $s_n \in \left[\ s_n^1,\ \ s_n^2, \cdots\ s_n^m,\ \ \cdots\ \ s_n^{M_n}\ \right]$, where s_n is the n^{th} sensor, s_n^m, is the m^{th} distinct value and M_n is the number of distinct values for sensor n. For any number of incoming sensors their distinct values or recognizable sensory conditions will constitute *features*. These features may or may not prove useful in controlling the agent. These features are defined over all distinctive values for all sensors, $f_i \in \left[\ \cdots\ \ s_{n-1}^1,\ \ s_{n-1}^2, \cdots\ \ s_n^1,\ \ \cdots\ \ s_N^{m_N}\ \right]$, where f_i is the i^{th} distinct feature and N is the number of sensors.

A Q-learning evaluation function is now defined for each distinct feature, Q^{f_i}. With features defined as recognizable sensory conditions, skills become the control strategies required to provide the actions to reach each feature. Eventually the lowest level primitive actions are needed to be invoked by the learned control strategies in an attempt to reach the desired feature. These primitive actions are the simple low level actuator movements that act on the world. They may themselves contain some form of feedback control mechanisms, but remain the simple building blocks out of which complex control strategies (skills) can be built. These actions are represented as $a_n \in \left[\ a_1,\ \ a_2,\ \ \cdots\ \ a_n,\ \ \cdots\ \ a_N\ \right]$, where a_n is a primitive action and N is the number of primitive actions.

Each feature, f_i, is by definition the endpoint of a skill with its control strategy represented by the evolving evaluation function, Q^{f_i}. While attempting to reach feature f_i, the control strategy can invoke any of the primitive actions, a_n or any of the skills represented by all Q^{f_i}. The choice of possible actions is now $u_n \in \left[\ Q^{a_1},\ \ \cdots\ \ Q^{a_n}\ \ \cdots\ \ Q^{a_N},\ \ Q^{f_i}\ \ \cdots Q^{f_I}\ \right]$, where u_n is a possible action or skill choice, Q^{a_n} is a non-adaptive primitive action and Q^{f_i} is an adaptive skill. The state x of the agent is established by the state of all the incoming sensors, $x_n \in \left[\ x_1,\ \ x_2,\ \ \cdots\ \ x_N\ \right]$, where x_n is a distinct state of the agent. The evaluation function for each feature is, $Q^{f_i} = f(x, u)$ or more specifically, $Q^{f_i} = f(x_1, \cdots x_n, \cdots x_N \cdots Q^{a_n}, \cdots Q^{f_i}, \cdots Q^{f_I})$, where both x_N and Q^{f_I} are open ended and subject to initial discovery and to increases and decreases due to ongoing changes. It is clearly seen that this evaluation function becomes nested and possibly recursive. That is evaluation function, Q^{f_i}, can invoke other skills including itself while attempting to reach feature f_i. It is this nested nature that will allow hierarchical control structures to emerge. As the agent interacts with the environment it receives an external reinforcement signal(s), r_{EXT}. It is through these signals that the agent is driven to perform tasks of external benefit. This reinforcement signal is defined as

$$r_{EXT} = \begin{cases} 0 & \text{if external task is achieved} \\ -R_{\mathrm{EXT}} & \text{otherwise} \end{cases} \tag{1}$$

where r_{EXT} is an external reinforcement signal and R_{EXT} is a positive constant.

In addition, there are various internal reinforcement signals, r_{INT}. These usually originate at sensors monitoring fragile components and drive the agent to perform tasks of internal benefit such as *avoid damage* and *avoid low fuel*.

In the nested Q-learning algorithm there is also a reinforcement signal that effects each currently active skill(s). This reinforcement signal drives the action of the agent to reach its defining and desired feature.

$$r_{\text{FEAT}} = \begin{cases} 0 & \text{if at desired feature} \\ -R_{\text{FEAT}} & \text{otherwise} \end{cases} \tag{2}$$

where r_{FEAT} is the feature's reinforcement signal and R_{FEAT} is a positive constant.

For the top-down goal directed action/skill selection the action chosen, u^*, is determined by,

$$\max_u \{Q^{f_i}_{x,u} + E_{x,u}\} \Rightarrow u^* \tag{3}$$

where max_u is the maximum function taken over all possible primitive actions and skills, $Q^{f_i}_{x,u}$ is an adaptive skill and $E_{x,u}$ is the exploration (a recency based exploration policy is used) strategy.

Upon performing the chosen action, u^* be it a primitive action or a skill, the robot advances from state x_v to the next state x_w and incurs a total reinforcement signal, r_{TOTAL}. Also included in this total reinforcement signal is the cost of performing the selected action. This cost is designated as r_{LOW} and includes all costs incurred by lower level skills and ultimately the actuators as control signals cascade downward in the hierarchy.

$$r_{\text{LOW}} = \begin{cases} -C & \text{if } u^* \text{ is a primitive action} \\ \sum_{k=0}^{K} r^{u*}_{\text{TOTAL}}(k) & \text{if } u^* \text{ is an adaptive skill} \end{cases} \tag{4}$$

where $\sum_{k=0}^{K} r^{u*}_{\text{TOTAL}}(k)$ is the total reinforcement signal from the invoked skill summed over the number of steps, K, required to perform the skill, u^* and C is a constant that reflects the cost of performing the primitive action.
The total reinforcement signal for the invoking skill becomes

$$r_{\text{TOTAL}} = r_{\text{EXT}} + r_{\text{INT}} + r_{\text{FEAT}} + r_{\text{LOW}} \tag{5}$$

This total reinforcement is used to construct the evaluation function that represents the expected reinforcement surface, from which useful top-down control strategies will emerge. The error, e_Q is defined to be,

$$e_Q = \gamma \cdot \max_u \{Q_{x_w,u}\} - Q_{x_v,u^*} + r_{\text{TOTAL}} \tag{6}$$

where γ is the temporal discount factor $0 < \gamma < 1$ and $\max_u\{Q_{x_w,u}\}$ is the current prediction of the maximum total future reinforcement remaining when agent leaves state x_w.

This error is used to adapt the evaluation functions.

$$Q_{x_v,u=u^*}(k+1) = Q_{x_v,u=u^*}(k) + \eta_Q \cdot e_Q \tag{7}$$

where η_Q is the rate of adaptation and k is the index of adaptation.

The preceding derivation described a nested Q-learning technique though which top down goal action selection mechanism that will propagate goal seeking commands down though an emergent hierarchical control system. Each skill, whenever invoked, will in turn invoke other skills and/or primitive actions in an attempt to fulfil the desired goals of higher skills. Next, the bottom up or sensory based action selection mechanism and how it interacts with the top down action selection mechanism will be described.

Consider the agent at state x_v and the skill Q^{f_i} is invoked. Upon the arrival at the sensory conditions of the defining feature, f_i, two things can occur: one, an uneventful arrival at f_i with only nominal reinforcements occurring, or two, the arrival at f_i coincides with high negative or high positive reinforcement signals. Such would represent possible reactive and opportunistic behaviors, respectively. The invocation of that same skill from another state may not necessarily result in such non-typical results or useful correlation. For example, the invocation of the *avoid obstacle* skill would not yield any benefit if an obstacle was not present. Similarly, the invocation of the *begin feeding* skill would not be of any benefit if food is not near. For these bottom-up relationships an evolving reactive function, $B^{f_i}(x_{invoked})$ is defined for each feature f_i. This function learns to predict the reinforcement outcomes of invoking each skill from different states.

Effectively these functions evolve into the bottom-up triggering mechanism for reactive and opportunistic skills. Within an emergent hierarchy, such a bottom-up action selection mechanism will be in control at the very top of the hierarchy as there are no higher levels to invoke them in a top-down manner. They will also be able to subsume the current top-down flow of commands should a reactive or opportunistic situation present itself. Eventually, some of the bottom-up reactions will be absorbed by the top-down control strategies. That is, a bottom-up triggering situation presents itself with enough regularity it will eventually be included in the evolving top-down structure. These bottom-up reactive behaviors are most important when the agent is in new situations where the top-down structure has not yet formed. They represent chunks of important information learned in the past that might be useful in these new situations. As encapsulated reactive/opportunistic skills that will likely contain their own top-down structures, these bottom-up contributions will be important in transferring these learned top-down skills between tasks and environments. The error, e_B for these functions is determined using the eventual reinforcement, r_{EVT}, that occurs at the defining feature.

$$e_B = r_{EVT} - B^{f_i}(x_{invoked}) \tag{8}$$

where $B^{f_i}(x_{invoked})$ is the bottom-up reactive function for skill Q^{f_i}, $x_{invoked}$ is the state from which Q^{f_i} was invoked and r_{EVT} is the total reinforcement when the feature has been reached.

The bottom-up predictive function is then adapted.

$$B^{f_i}(x_{invoked})(k+1) = B^{f_i}(x_{invoked})(k) + \eta_B \cdot e_B \tag{9}$$

where η_B is the learning rate.

As the agent gains experience with its environment and tasks it becomes able to predict opportunistic/reactive situations from state information. When B^{f_i} is included in the top down action selection mechanism of Equation 3 it will favour the selection of proven opportunistic and reactive skills above others, even those actions that it may have learned to take. This prevents irregular occurrences from distorting the learned evaluation function as well as allowing for transfer of skills to new situations. This allows for previously discovered and learned important information, of immediate benefit or danger, to be transfered between tasks and environments without having to rediscover it. A more detailed simulation study of information transfer can be found elsewhere [4].

3. Simulation

To evaluate the nested Q-learning generation of hierarchical control systems, the simple two dimensional robot

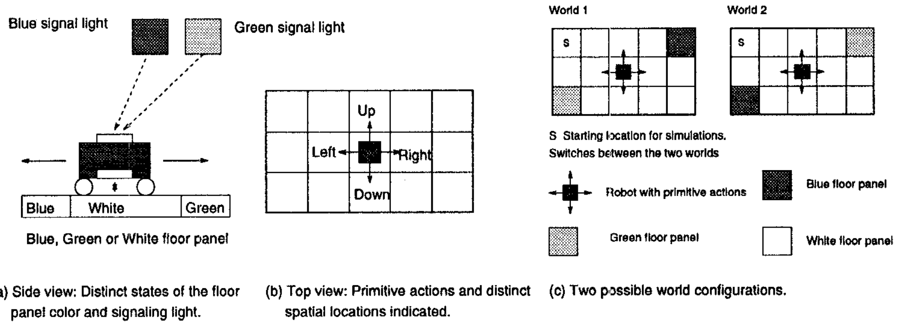

Blue signal light
Green signal light
Blue
White
Green
Blue, Green or White floor panel
(a) Side view: Distinct states of the floor panel color and signaling light.
Up
Left
Right
Down
(b) Top view: Primitive actions and distinct spatial locations indicated.
World 1
World 2
S
S Starting location for simulations. Switches between the two worlds
Robot with primitive actions
Blue floor panel
Green floor panel
White floor panel
(c) Two possible world configurations.

Fig. 2: Simulated robot's sensory systems: (a) Floor color and signal light sensors, (b) distinct spatial locations and possible movements of the robot. Simulated world: (c) Blue and green floor panels in two possible configurations as indicted. The locations of blue and green floor panels can change to that of $World_1$ or that of $World_2$.

and world of Figure 2 was used. The robot's primitive actions were capable of moving it to one of four adjacent spatial locations. The robot's sensors sensed the color of the floor panel below it, the color of a signalling light and the robot's spatial location within the world. The world is shown in Figure 2 (c). It had white colored floor panels except for one blue and one green floor panel at the locations indicated. It was possible for the location of these blue and green panels to change, moving to locations indicated by either $World_1$ and $World_2$ in Figure 2 (c). The robot itself remains unaware of these changes as it can only sense what is happening locally at its current spatial location within the world. Tasks for the robot are externally specified using external reinforcement signals and could entail anything from movement to a desired spatial location to matching sensed floor panel color with the color of the signalling light. Summarized in Figure 3 are the features found to be relevant by the robot. Although these features are subject to random discovery, they are presented in an orderly list for the readers benefit in the following analysis. To evaluate the capabilities of nested Q-learning to generate control hierarchies the agent was rewarded

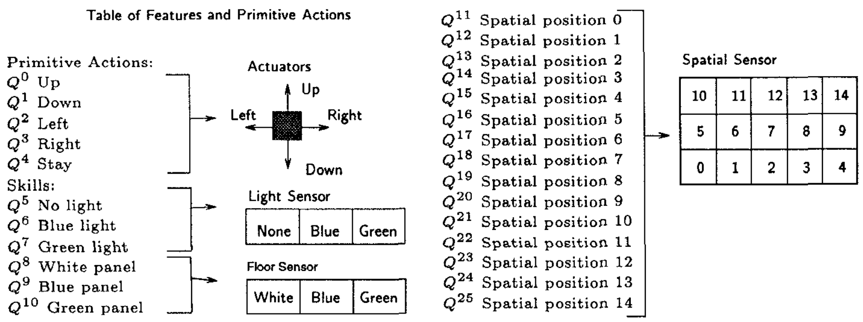


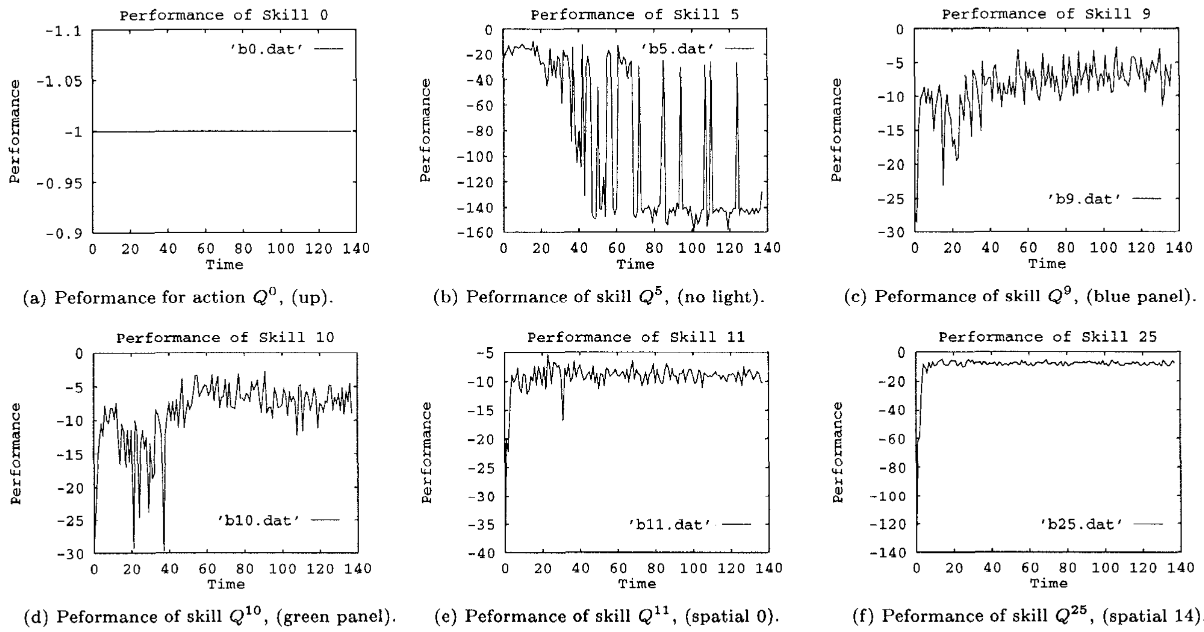
Table of Features and Primitive Actions
Primitive Actions:
Q^0 Up
Q^1 Down
Q^2 Left
Q^3 Right
Q^4 Stay
Skills:
Q^5 No light
Q^6 Blue light
Q^7 Green light
Q^8 White panel
Q^9 Blue panel
Q^{10} Green panel
Actuators
Up
Left
Right
Down
Light Sensor
None
Blue
Green
Floor Sensor
White
Blue
Green
Q^{11} Spatial position 0
Q^{12} Spatial position 1
Q^{13} Spatial position 2
Q^{14} Spatial position 3
Q^{15} Spatial position 4
Q^{16} Spatial position 5
Q^{17} Spatial position 6
Q^{18} Spatial position 7
Q^{19} Spatial position 8
Q^{20} Spatial position 9
Q^{21} Spatial position 10
Q^{22} Spatial position 11
Q^{23} Spatial position 12
Q^{24} Spatial position 13
Q^{25} Spatial position 14
Spatial Sensor
10 11 12 13 14
5 6 7 8 9
0 1 2 3 4

Fig. 3: Summary of skills and features.

with the external reinforcement signal of Equation 10.

$$r_{\text{EXT}} = \begin{cases} +2 & \text{if light is blue and floor is blue.} \\ +4 & \text{if light is green and floor is green.} \\ -1 & \text{otherwise.} \end{cases} \tag{10}$$

The locations of the blue and green floor panels were set randomly switching between the two possible configurations as shown in in Figure 2 (c). The signal light was set alternating between blue and green. A robot with no prespecified information was placed in this world and allowed to attempt to learn how to maximize its rewards over time. The performance plots for all skills and primitive actions were too large to be included here, but selected skills are shown in Figure 4. The vertical axis shows the performance of the skill or primitive action. The performance is taken to be the total reinforcement signal that the skill responds with whenever it is invoked. From these figures it is seen that the primitive actions respond consistently with a performance of -1.0 while all the adaptive skills performed change over time and usually improve. The first skills to be mastered are the ones defined by the spatial locations, skills Q^{11} through Q^{25}. What proved to be two higher level skills, Q^9, *find blue panel* and Q^{10}, *find green panel* were subsequently mastered using the two skills of Q^{11} and Q^{25}. The structure that emerged and the actions taken by the robot are shown in Figure 5 (b) and (c).



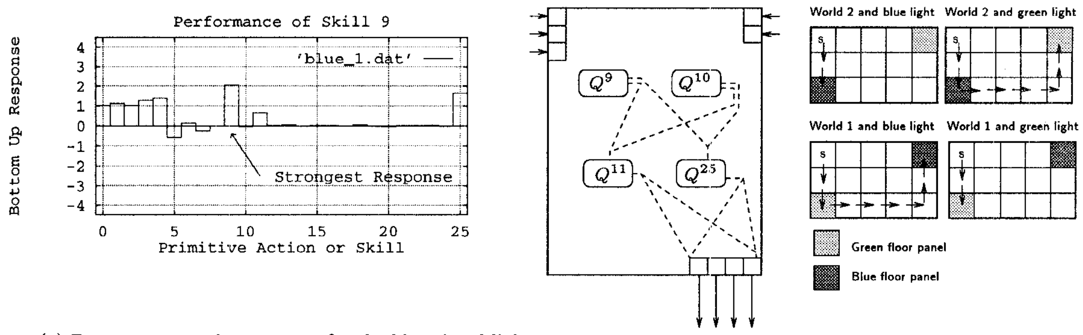
(a) Peformance for action Q^0, (up). (b) Peformance of skill Q^5, (no light). (c) Peformance of skill Q^9, (blue panel).

(d) Peformance of skill Q^{10}, (green panel). (e) Peformance of skill Q^{11}, (spatial 0). (f) Peformance of skill Q^{25}, (spatial 14).

Fig. 4: Detailed performance plots for selected skills.

Sensations or features over which the robot has no control were never learned, nor could they be. For instance,

Q^5, *find no signal light* could never be learned, as the signal light was always either blue or green and was never off. Interestingly enough, it was originally expected that a similar situation would occur for the other two signal light related behaviors, *find green signal* while the blue light was on and *find blue signal* while the green light was on, because they were not directly controlled by the agent. However, the light was alternating blue, then green, then blue, etc., changing upon task completion. The robot soon discovered that if it wanted to see a green signal light, all it had to do was go to the blue panel and once the task triggered by the blue light was completed, the green light would come on in place of the blue light. Examples of the agent finding novel solutions and taking advantage of unintentional loopholes in the simulation were common during these trials.



(a) Bottom up reactive response for the blue signal light on. Note the strength of response for skill Q^9.

(b) Emergent structure.

(c) Actions taken by the robot.

Fig. 5: (a) Bottom up reactive responses for all skills and primitive actions for blue signal light on sensory condition, (b) the structure that emerged with two bottom up driven skills, Q^9 and Q^{10}, at the top and (c) actions taken by the robot.

Figure 5 (a) shows the bottom-up responses of all the possible skills for an invoking state in which the blue signal light is on. The other details of that state were discovered by the control system to be irrelevant, as would be expected. As indicated, the predominant bottom-up response for the blue signal it corresponded to Q^9, or *find the blue panel.* These bottom-up responses were shown to respond high for the skill that was capable of fulfilling the task communicated by the signalling light. In these simulations, the green light triggered skill Q^{10} and the blue light triggered Q^9. These high level skill then utilized other skills and eventually primitive actions to fulfill the tasks. It was clear that a hierarchical control system emerged which, when invoked by a bottom-up opportunistic drive, began to search for the locations of the correctly colored panels. It should be noted that as this search requires two way backtracking through the state space and the continual disruption of a single monolithic evaluation function would make learning this problem in a non-hierarchical architecture difficult.

4. Conclusions

The nested Q-learning technique developed in this paper generated hierarchical control structures for the control of a simple simulated robot. These control structures resulted from having the controls strategies represented as many nested evaluation functions, rather than a single monolithic structure. The skills encapsulated by these control strategies could be invoked by other skills (to-down) or invoke themselves (bottom-up). As a bottom-up reactive/opportunistic drive was triggered, it was fulfilled by a cascade of top-down invoked skills.

References

[1] Barto, A.G., Sutton, R.S. and Watkins C.H. (1989), Learning and Sequential Decision Making, *COINS Technical Report.*

[2] Digney B. L. (1994), A Distributed Adaptive Control System for a Quadruped Mobile Robot, *From animals to animats 3: The third conference on the Simulation of Adaptive Behavior SAB 94,* Brighton UK, August 1994, pp 344-354, MIT Press-Bradford Books, Massachussets.

[3] Albus, J.S. (1991), Outline of a Theory of Intelligence. *IEEE Transactions on Systems, Man and Cybernetics,* 21, 3, pp 473-509.

[4] Digney, B.L. (1995), The Operator's Apprentice: Shaping Control Structures, *The Department of National Defence and Canadian Space Agency Conference on Robotics,* October 1995, St. Hubert, PQ., CANADA.

[5] Digney, B.L. (1996), Skill Transfer and Training in Nested Q-learning, *ICNN-96,* June 1996, Washington DC, (submitted).