Classical Conditioning in Social Robots

Rony Novianto[1], Mary-Anne Williams[1],
Peter Gärdenfors[2], and Glenn Wightwick[3]

[1] Quantum Computation and Intelligent Systems,
University of Technology Sydney, Australia
rony@ronynovianto.com, mary-Anne.williams@uts.edu.au
[2] Lund University, Sweden
peter.gardenfors@lucs.lu.se
[3] University of Technology Sydney, Australia
glenn.wightwick@uts.edu.au

Abstract. Classical conditioning is important in humans to learn and predict events in terms of associations between stimuli and to produce responses based on these associations. Social robots that have a classical conditioning skill like humans will have an advantage to interact with people more naturally, socially and effectively. In this paper, we present a novel classical conditioning mechanism and describe its implementation in ASMO cognitive architecture. The capability of this mechanism is demonstrated in the Smokey robot companion experiment. Results show that Smokey can associate stimuli and predict events in its surroundings. ASMO's classical conditioning mechanism can be used in social robots to adapt to the environment and to improve the robots' performances.

Keywords: Classical Conditioning, Maximum Likelihood Estimation, ASMO Cognitive Architecture.

1 Introduction

Classical conditioning is a cognitive skill crucial to learn and predict events in terms of associations between stimuli, and to produce responses based on these associations. People are expected to develop a classical conditioning when a right condition is presented repeatedly. If social robots have a similar cognitive skill to develop a classical conditioning like people, then people will know how they behave and can interact with them more naturally, socially and effectively. Social robots require cognitive skills that support the necessary social intelligence to engage with people and other robots effectively [15].

In this paper, we present a novel classical conditioning mechanism for social robots. This mechanism is implemented in ASMO cognitive architecture [6]. Section 2 first describes a definition of classical conditioning. Section 3 discusses existing computational models of a classical conditioning proposed in the literature and how they are different to this work. Section 4 describes the design and implementation of the classical conditioning mechanism in ASMO cognitive architecture. Section 5 evaluates ASMO's classical conditioning mechanism in the 'Smokey robot companion' experiment and shows that the robot can predict

M. Beetz et al. (Eds.): ICSR 2014, LNAI 8755, pp. 279–289, 2014.

users' requests. Finally, Section 6 summarises the benefit and future work of ASMO's classical conditioning mechanism.

2 Definition of Classical Conditioning

Classical conditioning (or Pavlovian conditioning) [10, p.109–110] [9] is an association of a neutral stimulus that does not elicit a response (called the *conditioned stimulus* or CS) with another stimulus that elicits a response (called the *unconditioned stimulus* or US), such that the presence of the CS will elicit the same response that would be elicited by the US, despite the US not actually being present. For example, if John repeatedly asks Mary to cook him the same dish every time he visits Mary, then Mary may develop the association between his visit and the dish, such that his presence will trigger Mary to *accidentally* start cooking the dish, even though John had asked Mary to go to a restaurant.

A classical conditioning is different to an operant conditioning. They are both a form of associative learning. However, a classical conditioning creates an association between *involuntary* behaviours and a stimulus before the behaviours are performed, whereas an operant conditioning creates an association between *voluntary* behaviours and their consequences after the behaviours are performed [2, pp. 141–142].

3 Existing Computational Models

Computational models of classical conditioning can be divided into models based on neural network and models that are not based on neural network. They can also be divided into trial-level and real-time models [3]. In trial-level models, the association between the stimuli is computed after all relevant stimuli have been observed and terminated. In real-time models, the association between stimuli is computed at every time-frame and the computation can cope with those frames being arbitrarily small.

In Furze's dissertation [3], he has reviewed a large number of trial-level and real-time computational models of classical conditioning (for both neural network and non-neural network models):

- The trial-level neural network models reviewed were the Pearce and Hall model and the Kehoe model.
- The trial-level non-neural network models reviewed were the Stimulus Substitution model, the Rescorla–Wagner model and the Mackintosh's Attention model.
- The real-time neural network models reviewed were the Grossberg model, the Grossberg–Schmajuk (G.S) model, the Klopf model (also called the drive-reinforcement model), the Schmajuk–DiCarlo (S.D) model and the Schmajuk–Lam–Gray (S.L.G) model.
- The real-time non-neural network models reviewed were the Sometimes-Opponent-Process (SOP) model, the Temporal Difference (TD) model and the Sutton–Barto (S.B) model.

In this paper, we focus more details on the real-time non-neural network models: SOP, TD and S.B models. This is because robots are required to operate in real-time. In addition, non-neural network models allow robots to learn without the need to be trained based on some prior input stimuli. Thus, they allow robots to predict stimuli that have not been trained previously.

The SOP model [14] represents a stimulus in one of three states: A1 (high activation), A2 (low activation) or I (inactive). A stimulus in the A1 state will elicit a primary A1 response (observed as an unconditioned response) whereas a stimulus in the A2 state will elicit a secondary A2 response. Two stimuli that are both in the A1 state will become associated and cause the strength of their association to increase. A stimulus that is either in the A1 or A2 state will induce its associated stimuli to enter their A2 states, which will then elicit their A2 responses (observed as conditioned responses). This inducement occurs in proportion to the strength of the association between the two stimuli. This model supports different phenomena of classical conditioning. However, it requires a stimulus to be represented in one of the three states and it is not implemented in robots.

The Temporal Difference (TD) model [13] is an extension of the Sutton–Barto model [12] proposed by the same authors. These two models rely on reinforcement (or rewards) and eligibility to determine the association strength of a stimulus (1). They have the same operations and equations, except that the reinforcement is determined by R_{TD} for the TD model (2) or R_{SB} for the SB model (3). Unconditioned stimuli have a starting association strength of a positive value. Other stimuli have a starting association strength value of zero.

$$\begin{aligned} \Delta V_t(i) &= \beta R \times \alpha(i)\overline{X}(i) \\ \overline{X}_{t+1}(i) &= \delta\overline{X}_t(i) + (1-\delta)X_t(i) \end{aligned} \tag{1}$$

$$R_{TD} = \lambda_t + \gamma Y_t - Y_{t-1} \tag{2}$$

$$R_{SB} = \dot{Y}_t = Y_t - Y_{t-1} \tag{3}$$

Where:
$R \in R_{TD}, R_{SB}$, $0 < \beta < 1$, $0 < \alpha < 1$
$V(i)$ and $\Delta V(i)$ are the association strength and the change of the association strength of stimulus i respectively
β and α are the constant reinforcement and eligibility learning rates respectively
$X_t(i)$ and $\overline{X}_t(i)$ are the strength and the weighted average strength (called eligibility trace) of conditioned stimulus i at time t respectively
δ is the decay rate of the eligibility trace
λ_t is the strength of the unconditioned stimulus at time t
γ is the discount factor
Y_t is the prediction made at time t of the unconditioned stimulus being associated

This paper presents the novel ASMO's classical conditioning mechanism based on attention and manipulation of memory. This mechanism differs from previous works in the following: (i) it does not require reinforcement values to learn and does not require specific representations of stimuli and responses, (ii) it is embedded in a cognitive architecture, (iii) it is not based on neural network, (iv) it is a real-time model and (v) it is implemented in a robot.

4 Design and Implementation in ASMO Cognitive Architecture

In this section, we describe the design and implementation of ASMO's classical conditioning mechanism based on the inspiration of human classical conditioning. We first review the overview of ASMO cognitive architecture. We follow by describing the mechanism and how it fits in the architecture.

4.1 Overview of ASMO Cognitive Architecture

ASMO [4,5,7] is a flexible cognitive architecture that orchestrates and integrates a diversity of artificial intelligence components based on bio-inspired model of attention. It can be used to explain and understand human cognition, however it does not aim to *imitate* the human cognitive architecture (i.e. it is bio-inspired rather than biomimetic).

ASMO cognitive architecture contains a set of self-contained, autonomous and independent processes (also called modules) that can run concurrently on separate threads (see Fig.1). Each module requests 'actions' to be performed. An action can be a low-level command to actuators, such as move head to a ball or walk to a specific location, or it can be a high-level function, such as store data to a memory, recognise objects (i.e. percept) or find the shortest path (i.e. plan).

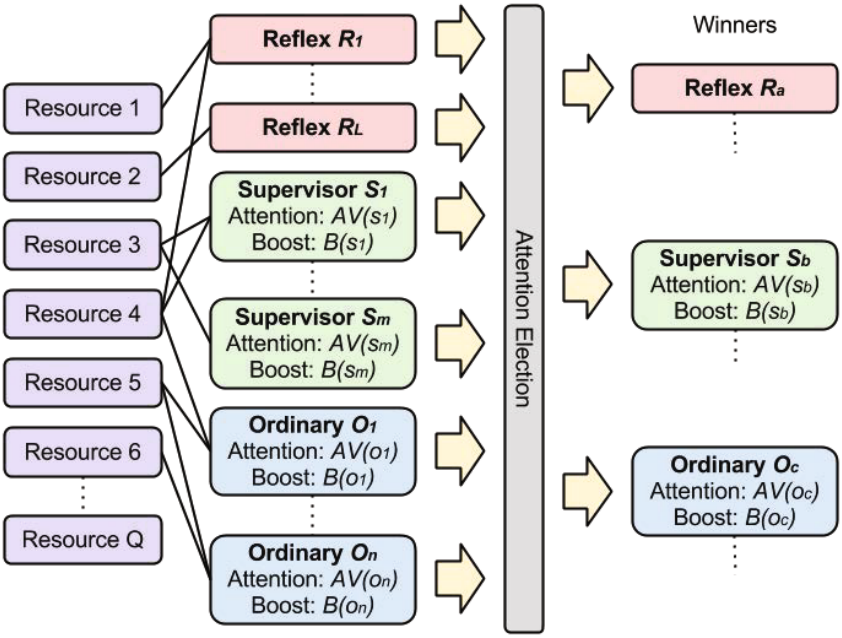


Fig. 1. Attention election in ASMO cognitive architecture

Actions can only be performed if the resources required by the actions are available (e.g. hand, leg, CPU, memory, virtual resource, etc). They can be performed simultaneously when there is no conflict in using the resources. Otherwise, modules with actions that require the same resources have to compete for 'attention' to use the resources. The winner of the competition is chosen based on the modules' types and attention levels.

Modules are divided into few types, including supervisor and ordinary modules. The supervisor and ordinary modules are non-reflex modules that have a 'attention value' attribute and 'boost value' attribute to determine their total attention levels (used to compete for attention). The 'attention value' attribute captures the degree of attention the module seeks based on the demand of the tasks whereas the 'boost value' attribute represents the bias associated with the module as a result of learning [6] or subjective influence [8]. Supervisor and ordinary modules are similar, except that supervisor modules can influence the total attention levels of ordinary modules but not vice versa.

Currently, modules that have the highest total attention levels will win the competition. The total attention level is given by the sum of boost and attention values. Under ordinary operation, attention values are (by convention) bounded between 0.0 and 100.0, equivalent to scaled values between 0.0 and 1.0. Modules with attention values of 0 demand the least attention whereas modules with attention values of 100 demand full or maximum attention. The boost value will bias this demand in the competition.

4.2 ASMO's Classical Conditioning Mechanism

ASMO's classical conditioning mechanism is created to trigger non-reflex modules to propose actions when the *conditional* stimulus is present even though the *unconditional* stimulus is not actually present. This mechanism is implemented in a supervisor module and its algorithm is described in the following five steps:

1. **Capture sequences of stimuli**
   ASMO's classical conditioning mechanism will capture sequences of stimuli. It represents each sequence of stimuli using a Markov chain where each node represents a stimulus.
2. **Calculate probabilities of stimuli will occur given an occurring stimulus**
   For every occurring stimulus, ASMO's classical conditioning mechanism will calculate the probabilities of other stimuli will occur (i.e. called 'candidates') given this occurring stimulus. In other words, it will calculate the probabilities of unconditioned stimuli being associated with a given conditioned stimulus. It calculates these probabilities by using the maximum likelihood estimation algorithm [1, p. 615]. These probabilities represent the strengths (or rather the confidences) of the associations between stimuli.
3. **Pick associated stimuli**
   ASMO's classical conditioning mechanism will pick the candidates that have significant probabilities as the stimuli being associated with the occurring

stimulus (i.e. pick the likely unconditioned stimuli). A candidate is significantly different if its root mean square deviation is above a threshold (4).

$$RMSD(c) = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(P_c - P_i)^2}$$
$$Significance(c) = \begin{cases} True & \text{if } RMSD(c) \geq T_{\text{RMSD}} \\ False & \text{otherwise} \end{cases} \tag{4}$$

Where:
$Significance(c)$ is the significance function of candidate c
$RMSD(c)$ is the root-mean-square deviation of candidate c
T_{RMSD} is the threshold of a candidate being significant
n is the number of candidates
P_i is the probability of candidate i

4. **Trigger modules to propose actions**
ASMO's operant conditioning mechanism will add the likely unconditioned stimuli to ASMO's memory as if these stimuli are currently occurring. This addition will cause non-reflex modules to believe that these stimuli are present, despite the fact that these stimuli are not physically present. As a result, it will trigger non-reflex modules to compete and propose actions in order to respond to these stimuli. Hence, the conditioned stimulus has triggered actions that are associated with the unconditioned stimuli without the unconditioned stimuli being physically present. This implementation allows a conditioned stimulus to be paired with a single unconditioned stimulus or multiple unconditioned stimuli.
5. **Repeat step 2 to step 4 for other occurring stimuli**
ASMO's operant conditioning mechanism will repeat step 2 to step 3 if there are other stimuli that are currently occurring.

The Markov chain model used by ASMO's operant conditioning mechanism may require many observations to provide an accurate estimation of reality. However, many observations are often not available and can be difficult to obtain. Thus, this mechanism uses a smoothing technique, such as the *Laplace smoothing* (also called *additive smoothing*) [11], to smoothen the observations in order to provide a better estimation.

5 Evaluation

ASMO's classical conditioning mechanism is experimented in Smokey robot companion project using a bear-like robot called Smokey [6,8,7]. This project aims to bring Smokey to 'life' and explores the meaning of life by interacting socially with people. It has potential applications in nursing, healthcare and entertainment industries by providing companionship to people with disabilities, people with autism, the elderly and children.

As part of the experiment, in a simplified scenario, Smokey has to accompany or entertain a person (i.e. the target user) while simultaneously regulating the person's rest. Smokey can play either a red ball game or drums to accompany the user. It can also go to sleep to encourage the user to rest (since it will not interact with the user when it is sleeping). When playing, Smokey will also pay attention to any motion in the environment from people other than the user.

Smokey can receive a request from the user through a graphical user interface to either play the red ball game, play the drums or go to sleep. It will consider this request, but does not necessarily have to perform this request. In addition, Smokey is desired to learn to predict the request that the user tends to ask and to perform this request before the user asks (i.e. to be conditioned by the appearance of the user so as to perform his/her request). Conditioning to the appearance of the user is similar to the example of classical conditioning described in Section 2. It will make Smokey more personalised to the user, which results in better companionship.

In summary, our hypothesis in this experiment was that ASMO's classical conditioning mechanism could model a classical conditioning: it could learn the association between the appearance of a user and the user's request. The methodology to validate this hypothesis was to show that after learning Smokey would perform the request that a user tended to ask (if any) when the user was seen. In addition, we would show the probability of the request compared to other requests. This experiment involved five users (i.e. participants) with different requests.

There were four ordinary modules and two supervisor modules created in this experiment to govern Smokey's behaviours:

- **The 'attend_motion' ordinary module**
  The 'attend_motion' module proposed an action when Smokey was not sleeping to look at the fastest motion in the environment. Its attention value was set to the average speed of the motion scaled between 0.0 and 100.0. The faster the motion, the more attention demanded by the module to look at the motion.
- **The 'play_ball' ordinary module**
  The 'play_ball' module proposed an action when Smokey was not sleeping either to track or to search for the ball depending on whether the location of the ball was known or not respectively. Its attention value was set to a constant value of either 60.0 when the user preferred Smokey to play the ball than to do other things, or 50.0 when the user preferred Smokey to do other things than to play the ball.
- **The 'play_drums' ordinary module**
  The 'play_drums' module proposed an action when Smokey was not sleeping either to play, track or search for the drums depending on whether the location of the drums was known and within reach, known but not within reach or unknown respectively. Similar to the 'play_ball' module, its attention value was set to a constant value of either 60.0 when the user preferred Smokey to play drums than to do other things, or 50.0 when the user preferred Smokey to do other things than to play the drums.

- **The 'go_sleep' ordinary module**
  The 'go_sleep' module proposed an action to go to sleep and wake up in every defined period. Its attention value was linearly increased until either it won an attention competition or its attention value reached 100.0 (i.e. maximum value of attention). This module then reset back its attention value to 0.0 after Smokey had enough sleep (i.e. predefined time).
- **The 'attend_request' supervisor module**
  The attend_request module proposed an action to increase the boost value of the play_ball, play_drums or go_sleep module when Smokey was not sleeping and requested to play the red ball game, play the drums or go to sleep respectively. It increased these boost values proportionally to the probability of the request (5). This probability was set to 1.0 when a request was received through a graphical user interface, or set to a value calculated by ASMO's classical conditioning mechanism when an associated stimulus was determined.
  The attend_request module did not require any resource. Its attention value was set to a constant arbitrary value of 10.0. This value does not hold any significant meaning. It does not have to be 10.0 and could be any value between 0.0 to 100.0. The reason is because the attend_request module did not need to compete for attention to gain access to resources since this module did not require any resource. Thus, this module will always be selected regardless of its attention value.

$$\begin{aligned} BV(pb) &= P(b) \times 20.0 \\ BV(pd) &= P(d) \times 20.0 \\ BV(gs) &= P(s) \times 20.0 \end{aligned} \tag{5}$$

  Where:
  $BV(pb)$ is the boost value of the play_ball module
  $BV(pd)$ is the boost value of the play_drums module
  $BV(gs)$ is the boost value of the go_sleep module
  $P(b)$ is the probability that the request to play the ball is received
  $P(d)$ is the probability that the request to play the drums is received
  $P(s)$ is the probability that the request to go to sleep is received
- **The 'classical_conditioning' supervisor module**
  The classical_conditioning module performed the five steps described in the previous section to learn the associations between the appearance of a user and his/her request. This module was specified by developers to observe users' requests. It calculated the probability of a user requesting Smokey to play the ball, to play the drums and to go to sleep. It determined requests with significant probabilities and added these requests into ASMO's memory every time the user was appear. This addition caused the attend_request module to believe that the user had made a request even though the user did not ask. As a result, the attend_request module increased the boost value of either the play_ball, play_drums or go_sleep module as if the user made an actual request.

Table 1 shows the requests received when interacting with the five users where -, b, d and s denote no request, play the ball request, play the drums request and go to sleep request respectively. Table 2 shows the probability of each request that might be asked by each user given when the user was seen. These probabilities were calculated based on the users' requests in Table 1 using the expectation maximization algorithm and Laplace smoothing with k of 1.0. Note that the probability of the request (or no request) that a user tended to ask was higher than other requests.

Table 1. Users' Requests

User	Requests
Anshar	b,s,d,d
Ben	d,d,d,d,d
Evelyn	-,-,-
Michelle	b
Xun	s,d,b,-,s

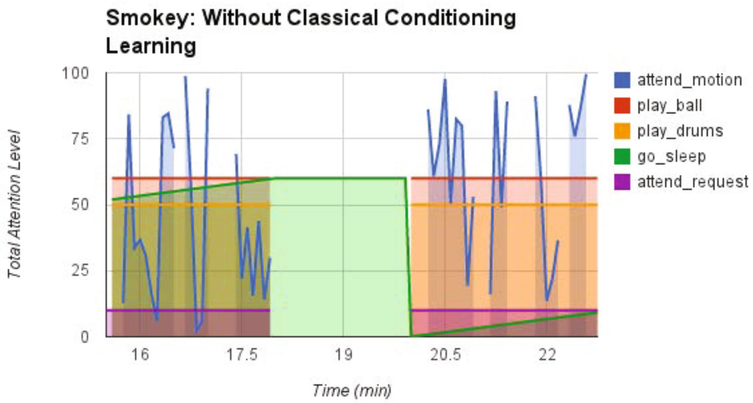
Table 2. Probability of Requests Asked by Users

User	Probability of Request Given User is Seen			
	Play Ball	Play Drums	Go to Sleep	No Request
Anshar	0.25	0.375	0.25	0.125
Ben	0.1111	0.6666	0.1111	0.1111
Evelyn	0.1429	0.1429	0.1429	0.5714
Michelle	0.4	0.2	0.2	0.2
Xun	0.2222	0.2222	0.3333	0.2222

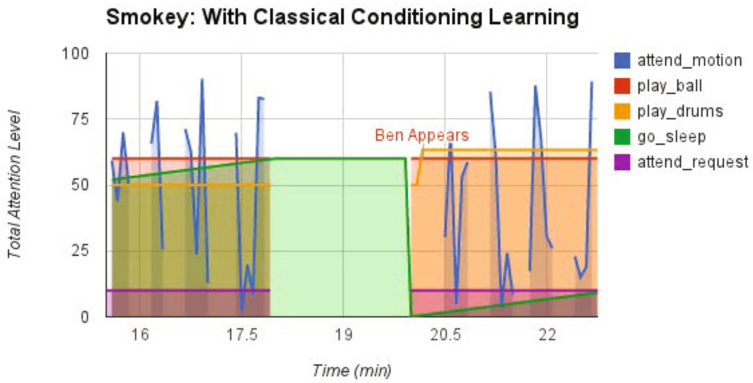
Figure 2 shows the result of the experiment without and with ASMO's classical conditioning learning mechanism when Smokey was interacting with Evelyn and then replaced by Ben. Both Evelyn and Ben preferred Smokey to play the ball rather than the drums. Thus, the total attention level of the play_ball module was initially higher than the total attention level of the play_drums module.

Without ASMO's classical conditioning mechanism, the total attention level of the play_drums module did not change when Smokey saw Ben. Thus, Smokey still chose to play the ball instead of the drums when interacting with Ben (i.e. no change of behaviour).

With ASMO's classical conditioning mechanism, the total attention level of the play_drums module was increased when Smokey saw Ben. This increase caused the total attention level of the play_drums module to be higher than the total attention level of the play_ball module. Thus, Smokey chose to play drums instead of the ball when interacting with Ben (i.e. change of behaviour). This change of behaviour showed that Smokey was classically conditioned to the appearance of Ben: it could learn the association between Ben's appearance and his requests.



(a) Without Classical Conditioning Learning



(b) With Classical Conditioning Learning

Fig. 2. Smokey's Classical Conditioning Learning

6 Conclusion

This paper has demonstrated the capability of ASMO's classical conditioning mechanism to learn in real-time in a physical robot without requiring reinforcement values. This mechanism is not based on neural network and has been embedded in ASMO cognitive architecture. It allows social robots to learn and predict events in the environment and to respond to those events.

For future work, ASMO's classical conditioning mechanism can be extended to further match the characteristics of human classical conditioning (with the aim to improve the mechanisms instead of imitating human classical conditioning). In addition, it can be extended to accommodate different types of learning, such as operant conditioning.

References

1. Bishop, C.: Pattern Recognition and Machine Learning. Information Science and Statistics. Springer (2006)
2. Chance, P.: Learning and behavior. Wadsworth (2013)
3. Furze, T.A.: The application of classical conditioning to the machine learning of a commonsense knowledge of visual events. Ph.d. dissertation, University of Leeds, United Kingdom (2013)
4. Novianto, R.: Flexible Attention-based Cognitive Architecture for Robots. Ph.d. dissertation, University of Technology Sydney, Sydney, Australia (2014)
5. Novianto, R., Johnston, B., Williams, M.A.: Attention in the ASMO cognitive architecture. In: Samsonovich, A.V., Jóhannsdóttir, K.R., Chella, A., Goertzel, B. (eds.) Proceedings of the First Annual Meeting of the BICA Society. Frontiers in Artificial Intelligence and Applications, vol. 221, pp. 98–105. IOS Press (November 2010)
6. Novianto, R., Johnston, B., Williams, M.-A.: Habituation and sensitisation learning in ASMO cognitive architecture. In: Herrmann, G., Pearson, M.J., Lenz, A., Bremner, P., Spiers, A., Leonards, U. (eds.) ICSR 2013. LNCS, vol. 8239, pp. 249–259. Springer, Heidelberg (2013)
7. Novianto, R., Williams, M.A.: The role of attention in robot self-awareness. In: The 18th IEEE International Symposium on Robot and Human Interactive Communication, RO-MAN 2009, pp. 1047–1053 (2009)
8. Novianto, R., Williams, M.A.: Innate and learned emotion network. In: Samsonovich, A.V., Jóhannsdóttir, K.R. (eds.) Proceedings of the Second Annual Meeting of the BICA Society. Frontiers in Artificial Intelligence and Applications, vol. 233, pp. 263–268. IOS Press (November 2011)
9. Pavlov, I., Anrep, G.: Conditioned Reflexes. Dover Publications (2003)
10. Powell, R.A., Honey, P.L., Symbaluk, D.G.: Introduction to learning and behavior, 4th edn. Cengage Learning (2013)
11. Russell, S.J., Norvig, P.: Artificial intelligence: a modern approach. Prentice Hall series in artificial intelligence. Prentice Hall (2010)
12. Sutton, R.S., Barto, A.G.: Toward a modern theory of adaptive networks: expectation and prediction. Psychological Review 88(2), 135–170 (1981)
13. Sutton, R.S., Barto, A.G.: Time-derivative models of pavlovian reinforcement. In: Gabriel, M., Moore, J. (eds.) Learning and Computational Neuroscience: Foundations of Adaptive Networks, pp. 497–537. The MIT Press, Cambridge (1990)
14. Wagner, A.R.: Sop: A model of automatic memory processing in animal behavior. In: Spear, N.E., Miller, R.R. (eds.) Information Processing in Animals: Memory Mechanisms, pp. 5–47. ch. 1. Lawrence Erlbaum Associates, Hillsdale (1981)
15. Williams, M.-A.: Robot social intelligence. In: Ge, S.S., Khatib, O., Cabibihan, J.-J., Simmons, R., Williams, M.-A. (eds.) ICSR 2012. LNCS, vol. 7621, pp. 45–55. Springer, Heidelberg (2012)