Prediction Driven Behavior: Learning Predictions that Drive Fixed Responses

Joseph Modayil and **Richard S. Sutton**
Reinforcement Learning and Artificial Intelligence Laboratory
University of Alberta

Abstract

We introduce a new method for robot control that combines prediction learning with a fixed, crafted response—the robot learns to make a temporally-extended prediction during its normal operation, and the prediction is used to select actions as part of a fixed behavioral response. Our method is inspired by Pavlovian conditioning experiments in which an animal's behavior adapts as it learns to predict an event. Surprisingly the animal's behavior changes even in the absence of any benefit to the animal (*i.e.* the animal is not modifying its behavior to maximize reward). Our method for robot control combines a fixed response with online prediction learning, thereby producing an adaptive behavior. This method is different from standard non-adaptive control methods and also from adaptive reward-maximizing control methods. We show that this method improves upon the performance of two reactive controls, with visible benefits within 2.5 minutes of real-time learning on the robot. In the first experiment, the robot turns off its motors when it predicts a future over-current condition, which reduces the time spent in unsafe over-current conditions and improves efficiency. In the second experiment, the robot starts to move when it predicts a human-issued request, which reduces the apparent latency of the human-robot interface.

Introduction

Certain aspects of a robot's behavior are commonly specified as a simple fixed response to a fixed stimulus, for example turning the motors off when they are stalling, or responding to commands from a human operator. These fixed stimulus-response pairings express domain knowledge for the behavior desired from the system, such as obeying human commands and not destroying the motors. However, responding only to a fixed stimulus can lead to overly conservative or risky behavior, as the robot and its environments typically change over time. Tire treads wear down, a battery's capacity decreases with use, floor textures change, and motor loads vary. In addition to environmental variability, the fixed response can have an effect after some time delay, in this case it is beneficial to start the response before observing the stimulus. If the robot could predict when a motor will stall and in response turn off the motor in advance, then the long-term damage to the motor could be reduced. Moreover, if the robot can improve its predictions by learning from ongoing experience, then the robot's behavior would also be continually adapted to its current operating environment.

Pavlovian conditioning is an extensively-studied mechanism of prediction learning in animals. Pavlovian conditioning is typically visualized as a dog in a laboratory learning to salivate on hearing a bell that signals the onset of food. In Pavlovian conditioning there is no reward signal—the animal modifies its behavior even when there is no benefit observed from the response (such as when the salivary glands are removed) and even when a punishment is given for the behavior modification (Mackintosh 1974). The simplest examples of Pavlovian conditioning are when the animal has an unconditional fixed response to a learned prediction of the stimulus.

This paper's main contribution is a new method for robot control that combines a fixed control response with a temporally-extended prediction that is learned online during the robot's normal operation. The control response is a fixed function that is freely crafted by the human designer to specify the behavior desired from the robot. We call this Pavlovian control due to its similarity with Pavlovian conditioning. This method is different from methods that perform no learning during deployment (including reactive controls, control policies learned offline, and control methods that make predictions with a fixed model). This method is also different from control methods in reinforcement learning that learn during deployment to maximize an explicit reward signal.

We demonstrate Pavlovian control on two tasks with physical robots that show how this method is able to improve the robot's behavior within minutes. The first demonstration is of safety, in which the robot anticipates wheel stalls and turns off the motors in advance. This response reduces harm to the motors, and improves the efficiency of the robot's movement. The second demonstration is an anticipatory human-robot interface, in which the robot learns to predict a human-provided command and then responds in advance of the command. The prediction-driven responses reduce the apparent latency of the robot with few errors.

Background

Although Pavlovian conditioning has been a topic of study for over a century, the underlying biological mechanisms are still an active topic of study. Before examining the modern biological understanding of this phenomenon, we introduce

some of the basic terminology, by means of the canonical example of a dog learning to associate a bell with food (as measured by its salivation). There are three components. The first is the unconditioned response (UR), which corresponds to an aspect of the dog's behavior, in this case salivation. The second is the unconditioned stimulus (US), which corresponds to presenting the dog with food. The third is the conditioned stimulus (CS), which is the bell ringing for some time before the food is presented. After some number of presentations of the bell preceding the food, the dog will start salivating before the food is presented.

Although it seems as first glance that the dog can be learning to salivate using standard reward-seeking mechanisms, the evidence points against this, as the dog will still learn to salivate even when there is no benefit accrued from salivation (when the salivary ducts are removed). This suggests that another mechanism must be involved. As described in a modern review of Pavlovian conditioning (Rescorla 1988), the primary phenomenon involved in Pavlovian conditioning is "the learning that results from exposure to relations among events in the environment".

For Pavlovian responses, the animal is not wired to emit the UR to the US, but rather the animal is wired to respond to the *prediction* of the US. Although initially this seems to be an unusual way to specify behavior, this explains much of the data on Pavlovian conditioning.

The time course of Pavlovian conditioning has been carefully studied for the eyeblink of a rabbit's nicitating membrane (UR) when irritated by a puff of air (US), and this event is preceded by a tone (CS). Under a variety of stimulation profiles, the rabbit is able to learn precisely how much time will elapse after hearing a tone, before the irritating stimulus will arrive at its eye. Moreover, the rabbit keeps its eye open until just before the irritating puff of air, and then closes its eyelid just before the puff of air arrives (with an accuracy in the tens of milliseconds), which means that the nerve impulses to close the eyelid must precede the event with precise timing. Indeed, experiments have shown this to be the case. Experiments have carefully timed the physical motion of these events under a variety of tone stimulus configurations (Keyhoe and Macrae 2002). Experiments have measured the time course of this phenomenon via electrical activity in the muscles (Lepora et al. 2007). The underlying circuit for this learned reflex resides in cerebellum, and recent work has shown such learning directly on the relevant cerebellum cells of live ferrets (Jirenhed and Hesslow 2011). In the experiment, the animal is curarized, which prevents any skeletal muscle motion, and the cerebellum is disconnected from the cerebrum to prevent influences from higher brain functions. They demonstrated that learning the timing relationships between CS, US, and UR happens within the cerebellar neural circuitry, by stimulating the mossy fibers (CS) and the climbing fibers (US), and measuring the output of a Purkinje cell (UR). Their experiment showed a single Purkinje cell can adapt to different CS-US timing intervals. The cerebellum is understood to be a uniform parallel architecture of this cellular pattern, and the cerebellum is the basis of motor co-ordination in many animals. The broad utility of this general control learning mechanism in animals suggests that there could be many uses for similar mechanisms in robots.

Although Pavlovian conditioning presents a promising direction for robot control, to the authors' knowledge there has been little study of Pavlovian conditioning as a general tool for engineering robot systems. Instead the focus of prior studies has largely been on developing computational models of the biological systems. One study (Ludvig, Sutton, and Keyhoe 2008) has demonstrated how temporal-difference learning algorithms can match the experimentally-measured learning curves for the rabbit-eyeblink, showing that temporal-difference learning as a computational model can fit the empirical data on Pavlovian conditioning. Work by Balkenius and Morén (1999) compared various computational models of Pavlovian conditioning, and related it to accelerating reward-based learning in mazes. Work by Mannella and colleagues (2009) attempted to recreate Pavlovian responses, but largely to emulate the underlying biology. Although many have followed David Marr and James Albus in considering the computational role of the cerebellum as providing a model for feedback or feedforward control (see van der Smagt (2000) for an overview), these works are largely hypothetical proposals considering the cerebellum to be computing a full model of a biological system's dynamics that is used as part of computing an optimal response, unlike the approach proposed here of coupling a fixed response with prediction learning.

Formalism

The robot interacts with the environment at discrete time steps, $t = 1, 2, 3, \ldots$, and at each time the robot receives a sensor observation o_t and then emits an action a_t. The observation is a function of the environmental state, but the robot does not observe this state. Instead, the robot's information about the state of the environment is acquired from its stream of observations and actions and is expressed as a feature vector $x_t \in \mathbb{R}^n$. A prediction task for the robot is specified with some time-varying observable signal of interest R, and a time-varying discount rate γ:

$$G_t \equiv \sum_{k=0}^{\infty} (\prod_{j=1}^{k} \gamma_{t+j}) R_{t+k+1}.$$

The quantity G_t is called the *return*, and its value depends on the robot's future experience. The term γ_t limits contributions to the infinite sum when it goes to zero. For example, γ_t can be 0.9 when the robot's bump sensor is inactive and 0 when the bump sensor is active; this causes any bump to be a terminating event for predictions starting from earlier time steps. A discount rate can also be interpreted as an expected timescale for the prediction (Modayil, White, and Sutton 2014), for a timescale of T steps the discount rate is set to $\gamma = 1 - \frac{1}{T}$.

For the predictions in this paper, we restrict our attention to the setting where the signal $R_t = \mathbb{I}\{E\}$ is the indicator function of a stimulus event E; this function is one when the stimulus is present and zero when it is absent. Similarly, we set

$$\gamma_t = (1 - \mathbb{I}\{E\})(1 - \frac{1}{\tau}),$$

which terminates the prediction on observing the stimulus event E or in τ time steps in expectation.

We define a Pavlovian control to be a fixed response to one or more predictions that are being learned continually during the robot's normal operation. The fixed response specifies the behavior desired of the robot as a function of the predictions and other state variables. For this paper we restrict our attention to Pavlovian controls that are a modification of a standard reactive control with one prediction. For concreteness, let π denote the default policy being followed by the robot (a policy is any function from the robot's state to a probability distribution over the actions it can take from that state). Let ϱ be the policy of the fixed response. If the reactive control is activated on a stimulus E, the action selected by the behavior policy b of the robot is expressed as

$$b = \mathbb{I}\{E\}\varrho + (1 - \mathbb{I}\{E\})\pi.$$

We refer to the above behavior policy as a reactive control.

A Pavlovian control can be constructed by modifying a reactive control to use a prediction v about E. This modification is to use a mixing function $f_E : \mathbb{R} \rightarrow [0, 1]$ that selects the response based on the observation of E or the prediction v,

$$b = f_E(v)\varrho + (1 - f_E(v))\pi.$$

The prediction is computed as a function of the robot's state at each time step. When the prediction function is adapted by an online learning process, we refer to the above behavior policy as a Pavlovian control.

If the latency of the response is known, then the mixing function f_E can be used to synchronize the response and unconditional stimulus. Namely, if the response takes K steps, then an appropriate mixing function is the following,

$$f_E(v) = \mathbb{I}\{E \ \vee \ v > (1 - \frac{1}{\tau})^K\}.$$

For example, consider the bell preceding the food for Pavlov's dog. Let $R = \mathbb{I}\{Food\}$ be the indicator function for the presence of food, and let $\tau = 50$. If the latency between requesting salivation (ϱ) and having saliva reach the mouth is $K = 10$ time steps, then the dog should start salivating when the prediction v is greater than $(1 - \frac{1}{\tau})^K \approx 0.82$.

We used the TD(λ) algorithm to learn the predictions during the robot's normal operation. The TD(λ) algorithm learns to approximate the return, with features available to the robot. The algorithm approximates the return with a linear function of the feature vector x_t, sometimes transformed by a monotonic differentiable function $\sigma : \mathbb{R} \rightarrow \mathbb{R}$ (which can be the identity function). The algorithm learns the weight vector $w_t \in \mathbb{R}^n$,

$$v_t \equiv \sigma(w_t^\top x_t) = \sigma(\sum_{j=1}^{n} w_t(j)x_t(j)),$$

to approximate the expected value of the return,

$$v_t \approx \mathbb{E}[G_t].$$

The TD(λ) algorithm has linear complexity per time step in the length of the feature vector which enables it to function in real-time even for large values of n. The two algorithm parameters are α (the step-size which controls the magnitude of the update, and is often set to $\frac{0.1}{||x||_2}$) and λ (the bootstrapping parameter, which can accelerate learning). The weights are updated at each time step using the algorithm shown below (in which σ' is the derivative of σ).

$$\delta_t = R_{t+1} + \gamma_{t+1}\sigma(w_t^\top x_{t+1}) - \sigma(w_t^\top x_t)$$
$$e_t = \gamma_t\lambda e_{t-1} + \sigma'(w_t^\top x_t)x_t$$
$$w_{t+1} = w_t + \alpha\delta_t e_t$$

The limiting accuracy of the predictions learned by TD(λ) are primarily a function of the expressiveness of the features. Under some common formal assumptions, the algorithm will converge to the best predictor that can be linearly represented with the given features. Moreover, the weight vector can be initialized to any desired vector (for example the zero vector or the weight vector with the least squared error over a batch of data).

Reducing Motor Stalls

Our first experiment shows how Pavlovian control can be applied to reduce motor stalls. Motor stalls occur when a mobile robot pushes against a larger object than it can move. An extended stall condition damages the motor and drains the batteries. The net damage to the motor is typically not directly measurable by the robot's sensors. However, the current drawn by the motor can be measured, and the motor can be turned off to prevent damage. A prototypical scenario for a reactive safety control is to make a temporary corrective response when a particular signal exceeds a threshold; here it is to turn off the motors when excessive current is drawn. The Pavlovian control is a modification of the reactive safety control that turns off the motors when it predicts that excessive current will be drawn.

The robot was an iRobot Create, a small two-wheeled mobile robot that is the research version of a commercial robot vacuum cleaner. The robot communicated by a Bluetooth dongle to a laptop. The robot sent observation packets from the sensors and received motor commands at a rate of 50 Hz. The corresponding duration of a time step for the learning algorithm was 20 milliseconds. The observation packet contained the values of all 44 robot sensors, which included contact sensors in the bumper, wheel-drop sensors, infrared drop-off sensors, a battery voltage sensor, and many sensors that reported constant values during the experiment. The most important sensors for this experiment were the over-current sensors. The robot sensed the current drawn by the each wheel and reported a binary observation for each wheel to indicate if it was drawing an excessive amount of current.

The sensor readings were stochastic, and isolated observations of an over-current condition occurred regularly in normal operation, such as when the robot overcame friction or small irregularities on the floor. However, a persistent over-current condition can damage the robot, either by partially melting the robot's shell, or causing the tires to become worn down (both have occurred on our robots).

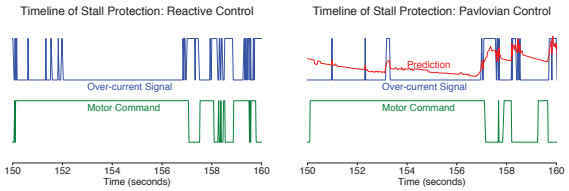

Figure 1: After 2.5 minutes of learning, the Pavlovian control had a substantial reduction in motor stalls compared to the reactive control. The graphs show relevant signals in the stall experiment for the two control policies over 10 seconds, which both end with the robot pushing against a wall. When the reactive control was pushing, the behavior rapidly cycled between the default policy (pushing against the wall with a high motor command) and the fixed response to the over-current signal that stopped the motors (the low motor command). In comparison, the Pavlovian control used the learned prediction that the stall would persist, and spent more time with the motors stopped.

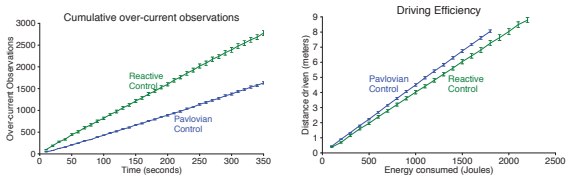

Figure 2: (left) The Pavlovian control achieved a 41% reduction of the over-current condition in the course of a 6 minute trial, and thus provided more safety than the reactive control. Learning provided almost immediate performance gains. (right) The Pavlovian control had a 10% gain in driving efficiency over the reactive control, so the robot's safety gains were not at the cost of functionality. The results are shown with standard error bars from 10 runs of each control policy.

We implemented a simple reactive control to turn off the motors when stalls occurred. We defined a stall event (E) to have occurred when a leaky integrator of the over-current sensor readings exceeded 3 (using a leaky integrator that incremented on an over-current signal, decremented otherwise, and was clamped to values between 0 and 5). The response of the control, ϱ, was to turn off the motors with a zero velocity command.

This reactive safety control interrupted action commands provided by a default policy π. The motor commands sent to the robot are expressed as

$$b = \mathbb{I}\{E\}\varrho + (1 - \mathbb{I}\{E\})\pi,$$

where b denotes the policy used to select actions for the robot at each time step.

The default policy π was a deterministic policy that rotated right for 3 seconds, then drove forward for 7 seconds, rotated right for 3 seconds, and then drove backward for 7 seconds. This policy was designed to repeatedly invoke the reactive safety control in a small testing environment, an enclosure in the shape of right angle triangle, where the short walls were one meter in length. In addition, a small weight (approximately two kilograms) was placed on the back of the robot so that the wheels would not spin on the wooden floor when the robot was pushing against a wall.

This policy caused the robot to push up against one of walls during the 7 seconds of driving. The motors quickly stalled and the reactive safety response was started. The reactive safety response turned off the motors, which after some latency caused the over-current condition to become false. After the over-current condition stopped, the default policy started to push on the wall again, which led the robot back to the over-current condition.

This reactive control method presented an opportunity for Pavlovian control. Although the information about the over-current condition was binary, another sensor on the robot measured the current drawn from the battery. The current drawn from the battery varied with the motor outputs, the type of floor, and the force with which the robot pushed against an obstacle. In particular, the battery current provided information relevant to predicting whether the robot was pushing against something. The robot's bump sensors also contained information about whether the robot was pushing against something when it drove forward. The information in these sensors suggested that the robot could use its sensor observations to predict whether a stall would be observed soon.

We constructed a prediction about an upcoming over-current condition at a timescale of 0.5 seconds, by setting $R_t = \mathbb{I}\{E\}$ and $\gamma_t = (1-\mathbb{I}\{E\})(1-\frac{1}{\tau})$ where $\tau = 25$ time steps. For the feature vector x, we used a single tiling of 10 tiles for each of the 44 dimensions of the observation vector. This resulted in a sparse 440 dimensional vector where 44 elements were one, and the remainder were zero. A tiling is a standard mechanism for converting continuous variables into sparse binary features in reinforcement learning (Sutton and Barto 1998). A tiling covers the domain of a continuous variable with tiles of uniform size and shape, and each binary feature is an indicator function which is one exactly when the value of the input variable falls on the tile. Here, each dimension of the robot's observation vector was scaled to between [0,1] and the tiles had a width of 0.1.

Using the learned prediction, the robot's behavior is specified with a mixing function f_E that activates if the prediction is above 0.5,

$$f_E(v) = \mathbb{I}\{E \ \vee \ v > 0.5\},$$
$$b = f_E(v)\varrho + (1 - f_E(v))\pi.$$

Converted into durations, the response is activated if a stall is predicted in $K = \log(1 - 1/25)/\log(0.5) \approx 17$ time steps (0.34 seconds),

We collected data from 20 interleaved runs, where the robot first used the reactive control policy and then the Pavlovian control policy. Each run lasted 6 minutes, and 10 runs were collected for each policy. For the learning algorithm, we set σ to the identity, $\lambda = 0.5$, and $\alpha = \frac{1}{44}$.

The Pavlovian control had visibly fewer over-current events than the original safety control. An example of the

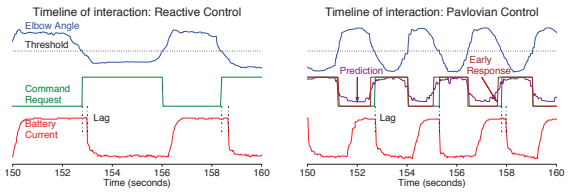

Figure 3: Timing of signals in the interaction experiment after 2.5 minutes. A human commanding the robot was observed by a stationary Xtion-Pro sensor. The reactive control sent desired motor velocities to the robot when the person's sensed elbow joint angle crossed a threshold. There was a lag from when the command was requested to when the robot started to move (as measured by the large negative current flow from the robot's batteries). The Pavlovian control reduced by apparent lag by starting an early response using the prediction of the command.

individual time course of a push against a wall is shown in Figure 1, where the reactive control experienced many more over-current observations than the Pavlovian control. The averages across the 10 runs for each policy is shown in Figure 2. The results show a 41% reduction in the number of over-current observations. However, an even more impressive decrease could be achieved by not letting the robot move at all. One measure of the progress enabled by the two controls is to measure the efficiency of the robot, namely how far it travels for a given amount of battery energy. An overly conservative control policy will cause the robot to not move at all and still consume power while idle, but the reactive control can waste power by pushing against walls. We evaluate the efficiency of the Pavlovian control against the reactive control in Figure 2. The graph shows a measurable performance improvement of the Pavlovian control over the reactive control. The Pavlovian control travels 10% farther than the reactive control for the same amount of energy consumed (the distance, battery voltage and battery current were measured by the robot's sensors, and the energy consumed was computed as the sum across time of the product of the instantaneous battery voltage and battery current). These results demonstrate that Pavlovian control can improve a robot's behavior.

Reducing Lag By Predicting the Next Command

Fixed reactive controls are often used to implement command interfaces to robots, in which a person issues a command request via a physical gesture, and the robot moves to follow the command. However, there are delays in the process from when the robot receives the command to when it begins to move. This situation can benefit from a Pavlovian control by having the robot initiate movement before the person requests it using a gesture.

A potential challenge for this scenario is that starting the wrong command could be dangerous. However, a dog can anticipate a command from a person and will commence movement when it has enough information to proceed, without necessarily waiting for the person to finish with the command. Analogous auto-completion capabilities exist for text entry boxes on web browsers. Thus preemptively initiating movement is only dangerous in certain settings, and the errors from recognizing gesture-based commands would be a more pressing concern in such settings.

For this experiment, we used a stationary Xtion Pro RGBD depth sensor (a device similar to the Microsoft Kinect sensor). This depth sensor has active light sensing to infer a depth for every pixel. The associated OpenNI software libraries separated the pixels associated with human bodies from the background scene, and fitted an approximate three dimensional skeleton to the visible bodies. The libraries provided the computed joint positions of a person's hands, elbows, and shoulders in a Cartesian reference frame.

We used these joint positions to define a simple command interface to the Create. If the observed left elbow joint was bent more than 90 degrees, then the robot spun left. If the observed right elbow was bent more than 90 degrees, then the robot spun right. When both elbows were bent beyond 90 degrees, the robot drove forward. This interface provided a simple, reactive control interface to the robot. However for simplicity, the following experiment only tested the left spin motion.

In this control interface, there was a measurable lag between the time that a request was observed via the elbow joint passing the threshold to when the robot started to move. The sensor measurements indicated a lag of 0.1 seconds before drawing battery current, and 0.3 seconds before the robot's odometry reported subsequent motion.

We implemented a Pavlovian control to replace the reactive control. The stimulus event E was to observe a command request from the left elbow, $R = \mathbb{I}\{LeftElbow < 90\}$. The timescale was set to 0.5 seconds ($\tau = \frac{.5}{.02} = 25$ time steps), with $\gamma = (1 - \mathbb{I}\{E\})(1 - \frac{1}{\tau})$. The threshold for the mixing function was set for 0.3 seconds ($K = 15$ time steps), with $f_E(v) = \mathbb{I}\{E \vee v > (1 - \frac{1}{\tau})^K\}$. The response ϱ was the left spin command.

The feature vector used to learn the prediction contained 500 binary features, with 5 active features. The sensed elbow angles were augmented with an exponentially weighted moving average of the elbow angles ($\mu_{t+1} = 0.8\mu_t + 0.2LeftElbow_t$). These continuous values were used to generate 5 two-dimensional tilings, each with 100 tiles, by splitting the range of each input dimension into 10 bins of equal size. Two tilings were used for the left elbow with its average (offset from each other), two tilings were for the right elbow with its average (also offset), and the final tiling was for the left elbow with the right elbow. The parameters were set to standard values $\alpha = 0.1/5$, and $\lambda = 0.8$. For the prediction algorithm, we used TD(λ) with a sigmoid nonlinearity, $\sigma(y) = (1 + e^{-y})^{-1}$, to keep the predictions in the range $[0, 1]$. To prevent the predictions from impacting the control initially, every component of the weight vector was initialized to -1, corresponding to a initial prediction of $(1+\exp(-5))^{-1}$ over the state space. The parameters of the learning algorithm were $\lambda = 0.8$, and $\alpha = \frac{0.1}{5}$.

The Pavlovian control was able to make anticipatory predictions in less than 3 minutes, as seen in Figure 3. To an-

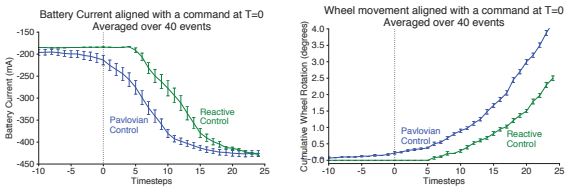

Figure 4: The graphs show averaged sensor observations from 40 commands, that are aligned to the onset of the command request. The reactive control took 5 time steps (0.1 seconds) before any additional current is consumed, and had a longer delay for the odometry to report wheel rotation. The Pavlovian control had an increased current draw at the time the command was requested, and had the wheels rotate earlier.

alyze the system's ability to initiate motion preemptively, we aligned the sensor readings at the moment the left elbow command was observed. The left graph of Figure 4 shows that the battery current began to change when the command was observed under Pavlovian control, approximately 0.1 seconds before the reactive control showed a change. The right graph of Figure 4 shows that the wheels began to move earlier with Pavlovian control. The data comes from the last 40 commands over a single run of each policy (each run lasted less than 10 minutes), and the gap in performance at most points is statistically significant ($p < .05$).

Discussion

Reactive controls are often deployed in situations with an asymmetric cost, where it is costly to not perform an response in the event of a stimulus, but not so costly to occasionally perform the activity in the absence of a stimulus. The experiments show that Pavlovian control can be used to improve on reactive controls, by decreasing the latency between the stimulus and the response, while still limiting the cost of unnecessary responses. Moreover, the prediction learning was fast in two ways; the learning algorithm operated within the 20 millisecond time step of the robot and the benefits of learning were measurable within minutes.

The method described in this paper generalizes in certain natural ways. One extension is to learn predictions off-policy, and thus enable the robot to make responses that prevent the occurrence of the event. Such predictions would improve the performance of the motor stall control. Another extension would be to consider more than a simple fixed response to a single prediction, which is not a complete characterization of Pavlovian conditioning in animals. In animals, the same information can elicit different responses depending on the modality by which the information arrives (Rescorla 1988). Studies of latent learning in animals have shown examples of how information gained by Pavlovian conditioning can be later used to accelerate the learning of reward-seeking behavior. Related ideas could be used to improve learning reward-seeking policies on robots.

The use of a prediction in a fixed response is a constrained use of the idea of predictive state representations (Littman, Sutton, and Singh 2002), in that the learned prediction is used as a state variable for selecting actions. Results for predictive state representations have shown that a large enough set of predictions can serve as a sufficient statistic for the environment. However, a complete representation of the environment is often not necessary for control, and this work shows that even a single well-chosen prediction can be useful. More complex decisions could be made with several carefully selected predictions.

Our experiments show how a single prediction can be thought of as a piece of knowledge that is both functional and learnable. The robot uses its prediction to select actions, and the robot learns more accurate predictions with time. Knowledge represented in a predictive form could be suitable foundation for intelligent and flexible robot behavior. Previous work has shown that very general forms of predictions can be learned online on a robot (Sutton et al. 2011), that many predictions can be learned in parallel on a robot (Modayil, White, and Sutton 2014; White, Modayil, and Sutton 2012), and that collections of predictions can be used as models for planning (Sutton et al. 2008). A robot requires a substantial amount of knowledge to demonstrate intelligent behavior (e.g. robot navigation (Kuipers 2000)), and learning predictions is a scalable mechanism for a robot to acquire knowledge.

Summary

We introduced a method that combines online prediction learning with a fixed crafted response to produce an adaptive behavior. The robot learns to make temporally-extended predictions during its normal operation, and the fixed response uses the prediction at each time step to select a task-appropriate action. We applied this method to two tasks, one was to reduce motor stalls and the other was a human-command interface. Learning was fast on both tasks, and led to better performance than reactive controls within minutes.

Acknowledgments

This research was supported by Alberta Innovates—Technology Futures through the Alberta Innovates Centre for Machine Learning, and the Reinforcement Learning and Artificial Intelligence Laboratory.

References

Balkenius, C., and Morén, J. 1999. Dynamics of a classical conditioning model. *Autonomous Robots* 7:41–56.

Jirenhed, D.-A., and Hesslow, G. 2011. Learning stimulus intervals—adaptive timing of conditioned Purkinje cell responses. *Cerebellum* 10:523–535.

Keyhoe, E. J., and Macrae, M. 2002. Fundamental behavioural methods and findings in classical conditioning. In *A Neuroscientist's Guide to Classical Conditioning*. Springer. 171–231.

Kuipers, B. J. 2000. The Spatial Semantic Hierarchy. *Artificial Intelligence* 119:191–233.

Lepora, N.; Mavritsaki, E.; Porrill, J.; Yeo, C.; Evinger, C.; and Dean, P. 2007. Evidence from retractor bulbi

EMG for linearized motor control of conditioned nictitating membrane responses. *Journal of Neurophysiology* 98:2074–2088.

Littman, M. L.; Sutton, R. S.; and Singh, S. 2002. Predictive representations of state. In *Advances in Neural Information Processing Systems 14*, 1555–1561.

Ludvig, E. A.; Sutton, R. S.; and Keyhoe, E. J. 2008. Stimulus representation and the timing of reward-prediction errors in models of the dopamine system. *Neural Computation* 20:3034–3054.

Mackintosh, N. J. 1974. *The Psychology of Animal Learning*. Academic Press.

Mannella, F.; Koene, A.; and Baldassarre, G. 2009. Navigation via Pavlovian conditioning: A robotic bio-constrained model of autoshaping in rats. In *The 9th International Conference on Epigenetic Robotics*.

Modayil, J.; White, A.; and Sutton, R. S. 2014. Multi-timescale nexting in a reinforcement learning robot. *Adaptive Behavior* 22(2):146–160.

Rescorla, R. 1988. Pavlovian conditioning: It's not what you think it is. *American Psychologist* 43(3):151–160.

Sutton, R. S., and Barto, A. 1998. *Reinforcement Learning: An Introduction*. MIT Press.

Sutton, R. S.; Szepesvári, C.; Geramifard, A.; and Bowling, M. H. 2008. Dyna-style planning with linear function approximation and prioritized sweeping. In *UAI*, 528–536.

Sutton, R. S.; Modayil, J.; Delp, M.; Degris, T.; Pilarski, P. M.; and Precup, D. 2011. Horde: A scalable real-time architecture for learning knowledge from unsupervised sensorimotor interaction. In *Proceedings of the 10th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*.

van der Smagt, P. 2000. Benchmarking cerebellar control. *Robotics and Autonomous Systems* 32:237–251.

White, A.; Modayil, J.; and Sutton, R. S. 2012. Scaling life-long off-policy learning. In *Second Joint IEEE International Conference on Development and Learning and on Epigenetic Robotics (ICDL-Epirob)*.