A Causal Model Approach to Dynamic Control

Zachary J. Davis, Neil R. Bramley, Bob Rehder, Todd Gureckis
{**zach.davis, neil.bramley, bob.rehder, todd.gureckis**} **@nyu.edu**
Department of Psychology, New York University
6 Washington Place, New York, NY, USA 10003 USA

Abstract

Acting effectively in the world requires learning and controlling dynamic systems, that is, systems involving feedback relations among continuous variables that vary in real time. We introduce a novel class of dynamic control environments using Ornstein-Uhlenbeck processes connected in causal Markov graphs that allow us to systematically test people's ability to learn and control various dynamic systems. We find that performance varied across a range of test environments, roughly matching with complexity defined by a set of models trained on the task (an optimal model, a deep Reinforcement Learning agent, and a PID controller). The testbed of dynamic environments and class of models introduced in this paper lay the groundwork for the systematic study of people's ability to control complex dynamic systems.

Keywords: dynamic control, causal learning, dynamic decision making, reinforcement learning, control theory

Introduction

> The principles of information, computation, and control bridge the chasm between the physical world of cause and effect and the mental world of knowledge, intelligence, and purpose.
>
> Steven Pinker (2018, p. 20)

Humans are daily met with the need to navigate and manipulate complex dynamic systems. Anyone who has been involved in a particularly engaging conversation, or one that escalates into a fight based on a shift in tone of voice, knows that we regularly deal with systems that are highly sensitive and involve complex dynamics. Our aim in this paper is to provide tools for studying continuous goal-directed control in complex dynamic natural environments. We do this by offering (1) a class of environments with appealing formal features, and (2) three candidate models of control behavior with widely diverging assumptions about cognitive processes. We hope that these environments, combined with candidate models, will aid researchers interested in this important topic in cognitive science.

Previous studies of human control have have generally not included an analysis of the formal properties of the dynamic system under investigation, such as its learnability or controllability. We build on this literature by introducing a class of dynamic systems composed of multiple components related by an underlying causal Markov structure. The explicit definition of underlying structure has two key favorable properties. Firstly, it enables us to systematically vary the environment that subjects interact with and so build an understanding of the conditions under which people succeed or fail at dynamic control. Secondly, as pointed out by Pinker (2018), successful control involves learning and harnessing causality to accomplish one's goals. Uniting causal learning and complex dynamic control in a single task makes explicit the connection between these key components.

We begin by briefly reviewing the literature on dynamic control tasks, highlighting where we depart from previous studies. We then provide a formal description of the "language" we use to describe system dynamics. We next report a novel experiment that investigates the extent to which people learn and exploit knowledge of the causal structure of a system to maximize reward. Finally, we introduce three candidate models that, when trained to optimize performance, generally agree on the difficulty of different environments. We conclude that people are capable controllers, but exhibit significant deviations from optimality that may be fruitful in guiding future research into the strategies and limitations of their control behavior.

Past research

Complex dynamic control (CDC) tasks have come under a variety of names. We follow Osman (2010) in grouping, under the umbrella of CDC, tasks described as complex problem-solving tasks, dynamic decision-making tasks, and process control tasks.

Static control. Control of dynamic systems has been heavily studied in static contexts (i.e., those where participants have clearly delimited trials with time to plan their next action). For example, Berry and Broadbent (1984) introduced the sugar factory task, where participants attempted to maintain a level of production in a sugar factory by controlling the number of workers employed. Hagmayer et al. (2010) tested control behavior in environments defined by a simple causal structure, finding that people responded adaptively when parts of the structure were lesioned, suggesting that they had learned the causal model and were anticipating the downstream effects of this intervention.

In the previous two studies, the dynamics simply involved a decay in the variable participants were trying to control. Other studies have introduced more complex dynamics, for example Gureckis and Love (2009) put short- and long-term rewards in competition in a task where the short-term choice was had a larger immediate reward but decreased potential future rewards. Schulz, Klenske, Bramley, and Speekenbrink (2017) introduced a navigation task in which a ship is pushed about in a nonlinear fashion depending on its value, finding that people explore environments strategically, planning forward to maximize information specifically relevant to future control actions.

We differ from these previous studies in several respects.

First, we present variables that change value, and can be controlled, in continuous rather than discrete time. Second, while a range of dynamics have been studied, they have all been with respect to a single variable's value. Our incorporation of structure between multiple variables allows for complex behavior such as oscillations, chains of influence, and feedback loops that have not been studied. Finally, by design, our environments are easy to model formally, allowing us to explore learnability and controllability at the computational level.

Micro-worlds. An important alternative approach to understanding control behavior are experiments on micro-worlds, which involve asking participants to stabilize a complex dynamic system as it unfolds in real time (for summary, see Brehmer, 2005). Unfortunately, the high ecological validity of their systems results in unconstrained environments where "one cannot be sure about the demands that a given microworld makes" (Brehmer, 2005, p. 87). We hope to contribute to this literature by having a model that can optimally infer the underlying structure, allowing us to identify which tasks are more difficult by their nature, and which are more difficult due to cognitive constraints.

Ornstein-Uhlenbeck Process

An Ornstein-Uhlenbeck (OU) process is a stationary Gauss-Markov process in continuous time that reverts to a stable mean (Uhlenbeck & Ornstein, 1930). It has been used to model phenomena in physics (Lacko, 2012) and finance (Barndorff-Nielsen & Shepard, 2001), and has also been studied in perception (Vul, Alvarez, Tenenbaum, & Black, 2009). Because these processes are able to capture dynamic natural phenomena across a wide variety of domains, we believe that the OU process is a reasonable formalism for modeling causal relationships between continuous variables in continuous time.

Generative model. In our environments, Δx_t—the change in x from time t to t+1—is defined as follows:

$$\Delta x_t = \theta\Big[\sum_{i=1}^{n} \beta_{Y^iX} \cdot y_t^i - x_t\Big] + N(0,\sigma) \qquad (1)$$

where x_t is the value of the process at time t, σ is the variance, and θ is a parameter greater than 0 that determines how sharply the process reverts to the mean. That mean is determined by the direct causal parents of X by summing, over each parent Y^i, the product of the strength of the causal relation between Y^i and X, β_{Y^iX}, and Y^i's current value, y_t^i. In this study, we constrain the values β_{Y^iX} so as to define three types of causal relationships: "regular" (β_{Y^iX}=1), "none" (β_{Y^iX}=0), and "inverse" (β_{Y^iX}=-1). For more extensive treatment of OU processes including how to optimally infer structure, see Davis, Bramley, and Rehder (2018).

Experiment: Control Task

Method

Participants. 36 participants (20 female, mean age=33) were recruited from Amazon Mechanical Turk using the psiTurk framework (Gureckis et al., 2016), which has been shown to produce comparable results to lab experiments in cognitive science (Crump, McDonnell, & Gureckis, 2013). They were paid $3.50 for approximately 25 minutes, with additional bonus based on performance (M=$0.52). Of the 36 participants gathered, 6 were excluded because they did not use the arrow keys on more than 25% of the phases.

Materials and procedure. See Figure 1 for illustration of a trial[1]. Each of the three variables was represented by a vertical slider constrained to be between -100 and 100. The handles of each slider presented a rounded integer value. One slider, the "control" slider, could be intervened on with three keys 'o', 'k', and 'm'. As is intuitive on a QWERTY keyboard, the 'o' key increased the control slider (by 10), the 'k' key held the value steady, and the 'm' key decreased the control slider by 10. If the participant did not press a key, the control slider would move according to the dynamics of the OU system.

The other two sliders could not be directly controlled by participants. One of these sliders, the "target" slider, had 20% of its area colored green to indicate the reward region. For each time step (100ms) that the target slider was in the green region, $0.01 was added to their score (displayed at the bottom of the screen). The top of the screen presented a timer counting down from 20 seconds, at which point the phase finished.

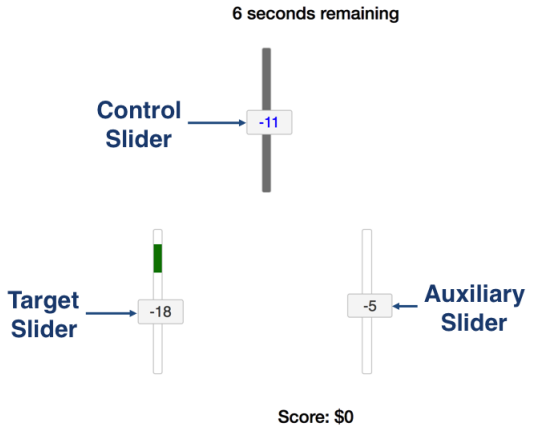

Figure 1: Example of the interface used by participants. Participants moved the control slider while observing the effects on the target/auxiliary slider. The goal was to exert forces on the control slider to keep the target slider within the green region.

[1]For a demo of the experiment see https://zach-davis.github.io/publication/dynamic_control/

In the instructions phase, participants were shown four videos of an agent interacting with the structures to familiarize them with the interface (participants were told the structure that the agent was interacting with). They were shown examples of (1) a network with no connections, (2) one with a single, direct "regular" relationship between controller and target, (3) a single "inverse" causal relationship, and (4) an indirect connection through the auxiliary slider (with both links being regular). Participants were then presented with a four question comprehension check. Questions established that participants understood that the connections, but not reward regions, stayed the same between Phase 1 and Phase 2, that the 'o' key moved the control slider up and the 'm' key moved it down, and that the reward would be a randomly selected phase of the experiment. Participants could not continue without answering all questions correctly. The parameters used during training and the control task were $\theta = .1$, $\sigma = 5$, and βs were either -1, 0, or 1.

In the control task, participants initiated the trial by pressing the "Start" button and the sliders started jittering according to an OU process, with unknown β weights driving the movement (there were no causes outside the network). The values of the sliders updated every 100ms, and each phase lasted for 20 seconds. At any time, participants were free to manipulate the control slider. After the first 20 second phase for a structure, participants received a pop-up inviting them to begin the second phase, and were reminded that the reward region but not the connections would change. After the second phase, the participants moved onto the next environment, repeating this process for each of the 12 structures (see Figure 2). After controlling in all environments, participants completed a brief questionnaire.

Models

In order to guide future research, we conducted preliminary analyses to assess the controllability of these systems by different models. Each model we consider has a different representation of the problem, from a Bayesian optimal learner (CMBC), to a highly flexible learner of state-action reward functions (DQN), to a limited agent that only learns whether they are in a direct or reverse acting system (PID). The data gathered in this paper are insufficient to distinguish between these models, but we introduce these models as candidates for how people approach complex systems, and in the discussion propose possible future experiments using our paradigm to more aptly distinguish between the models.

Causal Model Based Controller. The goal of the Causal Model Based Controller (CMBC) agent is to use its best estimate of the causal structure of the environment to act flexibly to maximize reward[2]. The CMBC agent, then, must estimate the probability of there being causal connections between sliders. For the current environment, causal strengths are defined as β weights between sliders (see the generative model section). Given some movement of slider x, and the values of other sliders y_i, the likelihood of causal strengths $\boldsymbol{\beta}$ is the density of Δx for the following PDF:

$$P(\boldsymbol{\beta}|\Delta x_t, y_t) \sim N\Big(\theta\Big[\sum_{i=1}^{n} \beta_{Y^iX} y_t^i - x_t\Big], \sigma\Big) \tag{2}$$

For each observation, we jointly estimate the full space of beta values for possible edges. For example, for three variables there are six possible edges, $\boldsymbol{\beta} = \{\beta_{XY}, \beta_{XZ}, \beta_{YX}, \beta_{YZ}, \beta_{ZX}, \beta_{ZY}\}$. Multiplying by the (initially uniform) prior probability of each hypothesis and normalizing yields the posterior over hypotheses.

The CMBC uses its online estimate of the causal structure of the environment to act. In particular, it imagines taking each of the four possible choices available to it ('o', 'k', 'm', or nothing). A given choice at time t will impact the controlled variable's state at time $t+1$. The CMBC then projects forward the effects that this choice would have over time. For this study, we project forward the impact of a choice for three time steps from the time of the decision. Because the process is stochastic, the impact of a choice will yield a probability distribution over possible states of the target variable. For each possible causal structure, it takes the integral of the expected distribution within the target range over all projected time steps. The expected value of a choice (C) given a structure (S) is:

$$\mathrm{EV}(\mathrm{C}|\mathrm{S}) = \int_{\text{range min}}^{\text{range max}} N(\mu_T, \sigma) dx \tag{3}$$

where μ_T is the mean of the target variable's distribution. The CMBC then weights the expected value of some choice given a structure by the probability of that structure:

$$\mathrm{EV}(\mathrm{C}, \mathrm{S}) = \mathrm{EV}(\mathrm{C}|\mathrm{S}) \cdot P(\mathrm{S}) \tag{4}$$

Marginalizing over structures gives the EV(C). The CMBC agent chooses the action that maximizes expected value.

Deep Reinforcement Learning. To compare to the CMBC, we considered a model-free reinforcement learning agent based on a deep-Q learning network (DQN). This model represents the state of the art for sequential decision making in complex environments similar to those studied here. Recently DQNs have been used to push the limits of what reinforcement learning algorithms can accomplish (e.g., learning to play Atari at near human levels, Mnih et al., 2015).

This model is interesting to compare for a number of reasons. First, a DQN is explicitly non-causal in that it has no direct represention of the environment and is unable to counter-factually plan future states and actions. At the same time, such models are powerful tools for dynamic control because they can learn forward-looking policies by approximating the solution to Bellman's equation. Related approaches have been used to successfully model human performance in discrete-time control and learning problems (e.g., Gureckis & Love, 2009).

[2]It is important to note that the learning is all passive, reducing uncertainty is not factored into the choice the CMBC agent makes.

To evaluate the ability of these networks to learn in the OU environment, we constructed a neural network in pyTorch (Paszke et al., 2017) with three layers. The input layer was made up of 6 inputs, representing the current location of each of the three sliders, the upper and lower bounds of the current reward region of the target slider, and the distance of the target slider's value from the mid-point of the target region. A fully connected hidden layer with 256 (rectified linear) units was in turn fully connected to an output layer with 4 linear units representing the estimated Q-values of moving the control slider up, down, hold it steady, or do nothing.

The target objective for training was the standard "on policy" Q-learning algorithm that learns how an action might effect future states of the system as well as the value of each action at the current time (Watkins & Dayan, 1992). For the first 1000 trials of learning the the model choose actions based on a linearly decaying epsilon-greedy choice rule, there after using softmax (Sutton & Barto, 1998). For each time step the target slider was maintained in the target reward region the network earned 10 units of reward. Furthermore, to punish extreme deviations from the target the reward was the negative of the absolute value of the distance between the target slider and the center of the reward region anytime the slider moved outside the target window. Learning was accomplished via gradient descent on the *smooth_l1_loss* of the difference between predicted and actual Q-values for each action. To speed learning, the network maintained a buffer of past state-action-reward-next state transitions and randomly sampled 64 of these each trial for use in a single batch gradient update. Specific network parameters were set as follows: discount rate ($\gamma = 0.98$), learning rate ($\eta = 0.001$), and softmax temperature ($\tau = 8$).

Even with these powerful learning features, the DQN is at a disadvantage in learning the task because it comes with randomly intialized weights and can only learn the objective of the task by experiencing certain state-action transitions paired with reward. As a result it cannot learn to perform the task in real-time (e.g., using the same number of time steps as human participants). However, it can still provide insight into the relative difficulty of learning each environment. To evaluate this, we ran the DQN 200 times on each structure with a given reward region, froze the weights, and had the network try to maximize reward on the trained reward region as well as a new reward region (identical to phase 1 and phase 2 for participants).

Proportional-Integral-Derivative. The Proportional-Integral-Derivative (PID) controller adjusts its actions based on a proportion of its error—the difference between desired and observed outcomes. It has been found to be a good model of how people generate predictions in dynamic environments (Ritz, Nassar, Frank, & Shenhav, in press), but has yet to be applied as a model of people's decision-making in dynamic control tasks. It operates by computing and storing a history of prediction errors, and performing some simple operations over this history. The form we use for the PID controller is:

$$u_t = K_P e_t + K_I \sum_{n=1}^{t} e_n + K_D(e_t - e_{t-1}) \quad (5)$$

where e_i is the error at time i and K_P, K_I, and K_D refer to the proportional, integral, and derivative components of the controller, respectively. The first component—$K_p e(t)$—is known as the "proportional" term. It is identical to the delta-rule (Widrow & Hoff, 1960), adjusting the state some proportion of the way towards the desired setpoint as a function of the currently observed difference between observed and desired states. The second component—$K_i \sum_{n=0}^{t} e_n$—is known as the "integral" term, computing a (signed) sum of previous errors. This component corrects for the system consistently over- or under-shooting the target. The final, "derivative" term, is rarely used and so we set K_D to 0 for our purposes.

Error is the key operator in a PID controller. The controller is given a setpoint (the mean of the target region), and at each time point subtracts the value of the target variable from the setpoint to get e_t. This is stored in a buffer, and the desired value for the control state (u_t) is computed as in Equation 5. The control action ('o', 'k', or 'm') that moves the control variable closest to u_t is chosen as the action. Note that, because it has no capacity to project forward the OU dynamics, the PID does not have the option to not act.

In practice, PID controllers are built with the knowledge of whether increasing the control variable will result in an increase or decrease in the target variable. Of course, in our task agents must learn this relationship. For this reason we build an additional component on top, to estimate whether the control variable is positively or negatively related to the target variable. This involves a simple timelagged correlation between control and target variables, of the form $\rho(Control_{t-1}, Target_t)$. The agent begins by randomly assuming a positive or negative connection, and if a correlation is found to be significant ($p<.05$) then it sets the sign of K_P, K_I, and K_D to be the same as the sign of the correlation coefficient.

Results

A paired-samples t-test did not reveal a significant difference in performance between phases of the study (M=4.25, SD=14.12); t(29)=1.65, p=.11, so we collapse over phase in figures and analyses.

To compare participant judgments to model predictions, we had the models perform the task. Figure 2 shows the reward curves for each agent (including participants). It is important to note that the models were fit to maximize performance on the task, not to most closely match participants[3]. Also worth noting is that the reward curves for participants, CMBC, and PID are on their first experience of a new structure, whereas the reward curve for the DQN is after extensive training on that structure (200 play-throughs).

[3]Parameters chosen to maximize task performance were γ, η, and τ for the DQN; K_P and K_I for the PID; and no parameters for the CMBC

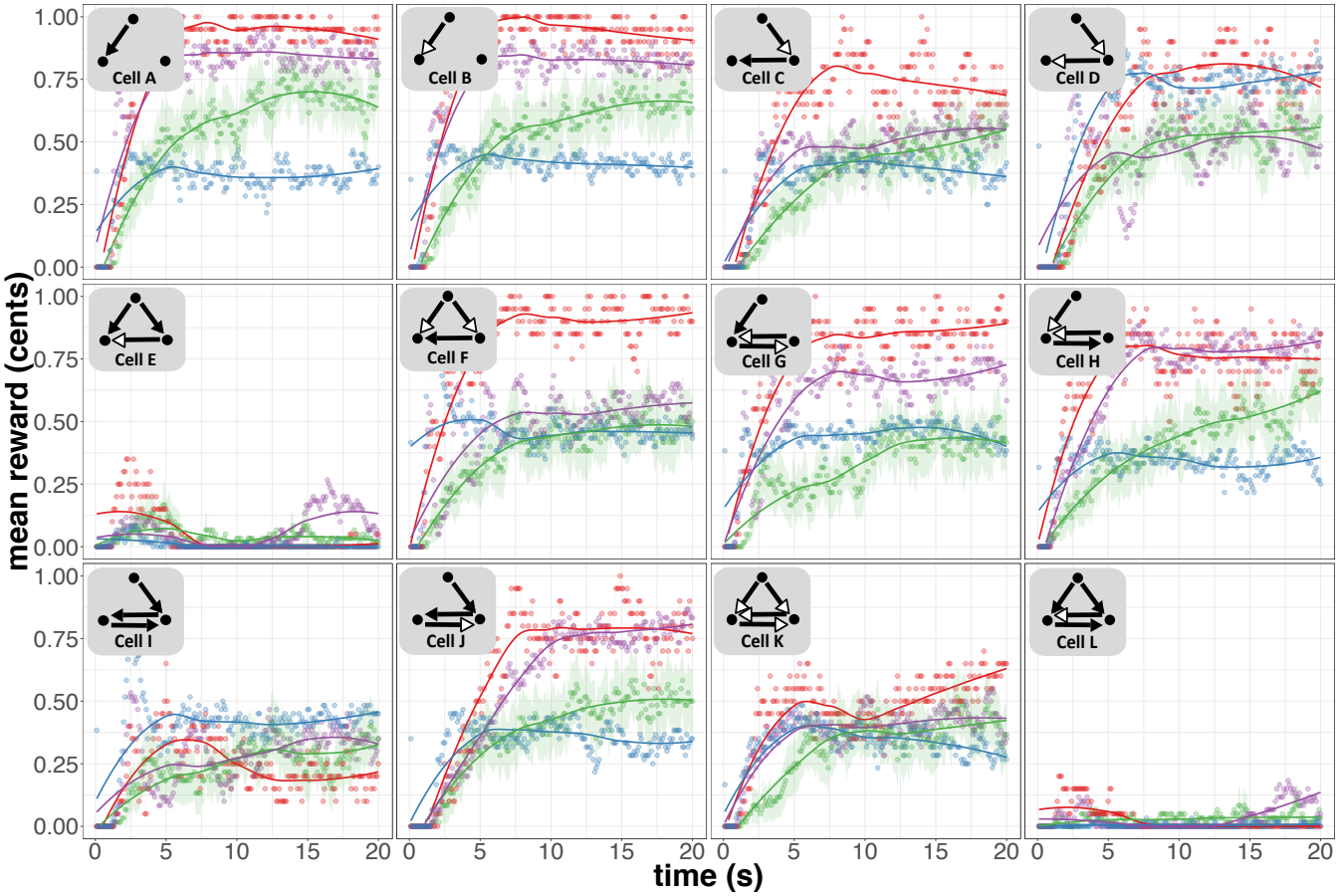

Figure 2: Proportion of possible reward received per-time step over the course of a trial for each agent. Green lines represent participants; red lines the CMBC; purple lines the PID controller; and blue lines the Reinforcement Learning agent. Error bars for participants denote 95% confidence intervals (from normal approximation to the binomial). The graphs in the corner of each plot label the causal structure that determined the dynamics of the environment with the node at the top of the triangle mapped to the control slider and the node on the left mapped to the target slider. Solid arrowheads denote regular connections ($\beta = 1$), white arrowheads denote inverse connections ($\beta = -1$).

The first thing to note is the surprisingly poor performance of the DQN. In fact, in phase 1 the DQN was at or even exceeded the performance of the CMBC. However, its performance collapsed in phase 2, suggesting that its state-action representations were not flexible enough to accommodate new goals. None of the CMBC, PID, or participants exhibited such dramatic differences between phases. While interesting, this idiosyncrasy clouds analyses of the controllability of different environments, and thus we exclude the DQN from the following discussion on their formal properties.

As can be seen, the models and participants generally agree on the difficulty of different structures. For example, for participants, the CMBC, and the PID, reward curves in Cells A and B (direct links) have higher plateaus than Cells C and D (indirect links). This is because noise propagates through causal links. In an environment with a direct control-target relationship, a control action has a direct influence on the probability distribution of the target. For an indirect control-target relationship, a control action influences the probability distribution of an intermediate variable, which then further spreads the distribution of the target.

Of course, the most dramatically different environments are cells E and L. In these environments, holding the control variable at any point trends the target toward 0, because the control variable exhibits a direct influence on the target, but also an indirect (and hence time-lagged) influence of the opposite sign. The mean that the control variable trends to, then, is the function $Control_t - Control_{t-1}$. Learning this function, and planning far enough ahead to exploit a strategy that maximizes reward, would be an interesting problem in hierarchical planning that we do not investigate here.

While there are qualitative similarities in performance between the models and participants, there are also significant departures. For example, people underperform the CMBC and PID in cells G, H, and J. These environments all involve exogenous influences—in the form of feedback loops and

oscillations—interfering with intended control actions. Future models of heuristics, strategies, or processing limitations may shed light on the reason(s) for this underperformance.

To test the extent to which the models and participants agreed on the relative difficulty of environments, we correlated participant reward curves with the models. Participant judgments correlated with predictions from the CMBC (r=.86, p<.001), the PID controller (r=.84, p<.001), and the DQN agent (r=.63, p<.001). Note that these models are highly correlated with one another, so future experiments are needed to pull the models apart.

Discussion

In this paper, we presented a new class of environments that can be systematically varied in order to test people's ability to learn and control dynamic systems. We found that people and our models generally found the same environments easy or difficult, although differences exist that may be informative about people's strategies or cognitive limitations. We expect that this new class of environments will be useful to the field as a test bed, and to draw links between the formal analyses in the causal literature and the sophisticated but black-box style learning of contemporary control tasks.

Although the data gathered in this experiment were not sensitive enough to distinguish between the models' ability to predict human behavior, the incorporation of causality in our dynamical system allows for a diverse range of future experiments to further test people's flexibility in control. Future experiments could test people's sensitivity to switching reward variables, counterfactuals, or multiple target or control variables. These studies would allow for a deeper investigation into the conditions under which people are model-based controllers—learning the structure of the environment and using it to plan actions—or doing something more model-free (akin to the DQN or PID agents). In a slightly different vein, the system described in this paper could be used to study a type of real-time 'systems programming', where dynamical systems are learned individually and then linked up into a larger structure.

Problem-solving, here operationalized as the ability to manipulate one's environment in service of some goal, is fundamental to higher-level cognition (Newell & Simon, 1972). Problems that we have to solve in our everyday lives do not often come pre-packaged or with clearly delimited trials. Rather, we must deal with control problems of unknown structure, learning through noisy feedback as we attempt to gather rewards. Impressively, people are able learn how these systems work, and can leverage this knowledge to be robust and flexible in controlling complex systems.

References

Barndorff-Nielsen, O. E., & Shephard, N. (2001). Non-Gaussian Ornstein-Uhlenbeck-based models and some of their uses in financial economics. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 63(2), 167-241.

Berry, D. C., & Broadbent, D. E. (1984). On the relationship between task performance and associated verbalizable knowledge. *The Quarterly Journal of Experimental Psychology*, *36*(2), 209-231.

Brehmer, B. (2005). Micro-worlds and the circular relation between people and their environment. *Theoretical Issues in Ergonomics Science, 6*(1), 73-93.

Crump, M. J., McDonnell, J. V., & Gureckis, T.M. (2013). Evaluating Amazon's Mechanical Turk as a tool for experimental behavioral research. *PloS one*, 8(3), e57410.

Davis, Z. D., Bramley, N. B., & Rehder, B. E. (2018). Causal Structure Learning with Continuous Variables in Continuous Time. In *Proceedings of the 40th Annual Conference of the Cognitive Science Society*. Austin, TX: Cognitive Science Society.

Gureckis, T. M. & Love, B. C. (2009) Learning in Noise: Dynamic Decision-Making in a Variable Environment. *Journal of Mathematical Psychology*, 53, 180-193.

Gureckis, T. M., Martin, J., McDonnell, J., Rich, A. S., Markant, D., Coenen, A., ... & Chan, P. (2016). psiTurk: An open-source framework for conducting replicable behavioral experiments online. *Behavior research methods, 48*(3), 829-842.

Hagmayer, Y., Meder, B., Osman, M., Mangold, S., & Lagnado, D. (2010). Spontaneous causal learning while controlling a dynamic system. *The Open Psychology Journal*, *3*, 145-162.

Lacko, V. (2012). Planning of experiments for a nonautonomous Ornstein-Uhlenbeck process. *Tatra Mountains Mathematical Publications*, 51(1), 101-113.

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., ... & Petersen, S. (2015). Human-level control through deep reinforcement learning. *Nature*, *518*(7540), 529.

Newell, A., & Simon, H. A. (1972). *Human problem solving* (Vol. 104, No. 9). Englewood Cliffs, NJ: Prentice-Hall.

Osman, M. (2010). Controlling uncertainty: a review of human behavior in complex dynamic environments. *Psychological bulletin*, 136(1), 65.

Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., ... & Lerer, A. (2017). Automatic differentiation in PyTorch.

Pinker, S. (2018). *Enlightenment Now: The Case for Reason, Science, Humanism, and Progress*. Penguin.

Ritz, H., Nassar, M. R., Frank, M. J., & Shenhav, A. (in press). A control theoretic model of adaptive learning in dynamic environments. *Journal of Cognitive Neuroscience*.

Schulz, E., Klenske, E., Bramley, N., & Speekenbrink, M. (2017). Strategic exploration in human adaptive control. In *Proceedings of the 39th Annual Conference of the Cognitive Science Society*. Austin, TX: Cognitive Science Society.

Sutton, R. S., & Barto, A. G. (1998). *Reinforcement learning: An introduction* (Vol. 1, No. 1). Cambridge: MIT press.

Uhlenbeck, G. E., & Ornstein, L. S. (1930). On the theory of the Brownian motion. *Physical review*, *36*(5), 823.

Vul, E., Alvarez, G., Tenenbaum, J. B., & Black, M. J. (2009). Explaining human multiple object tracking as resource-constrained approximate inference in a dynamic probabilistic model. In *Advances in neural information processing systems* (pp. 1955-1963).

Watkins, C. J., & Dayan, P. (1992). Q-learning. *Machine learning*, *8*(3-4), 279-292.

Widrow, B., & Hoff, M. E. (1960). *Adaptive switching circuits*. Stanford Electronics Labs. (No. TR-1553-1).