Journal of Experimental Psychology: Learning, Memory, and Cognition

A Decay-Based Account of Learning and Adaptation in Complex Skills

Roderick Yang Terng Seow, Shawn A. Betts, and John R. Anderson

Online First Publication, November 15, 2021. http://dx.doi.org/10.1037/xlm0001071

CITATION

Seow, R. Y. T., Betts, S. A., & Anderson, J. R. (2021, November 15). A Decay-Based Account of Learning and Adaptation in Complex Skills. *Journal of Experimental Psychology: Learning, Memory, and Cognition*. Advance online publication. http://dx.doi.org/10.1037/xlm0001071

AMERICAN PSYCHOLOGICAL ASSOCIATION

Journal of Experimental Psychology: Learning, Memory, and Cognition

ISSN: 0278-7393

https://doi.org/10.1037/xlm0001071

A Decay-Based Account of Learning and Adaptation in Complex Skills

Roderick Yang Terng Seow[1, 2], Shawn A. Betts[1], and John R. Anderson[1, 2]
[1] Department of Psychology, Carnegie Mellon University
[2] Center for the Neural Basis of Cognition, Carnegie Mellon University

OPEN DATA

How do humans adapt to parametric changes in a task without having to learn a new skill from scratch? Many studies of memory and sensorimotor adaptation have proposed theories that incorporate a decay on prior events, which leads the agent to eventually forget old experiences. This study investigates if a similar decay mechanism can account for human adaptation in complex skills that require the simultaneous integration of cognitive, motor, and perceptual processes. In 2 experiments, subjects learned to play a novel racing video game while adapting to parametric changes in the physics of the game's controls. Human learning and performance were modeled using the ACT–R cognitive architecture, which has been used successfully to model learning and fluency across a wide range of skills in prior research. Anderson et al. (2019) introduced the Controller module, a new component of the architecture that learns the setting of control parameters for actions and allows the agent to execute the rapid and precise actions that are necessary for good performance on complex tasks. Model simulations support including a moderate time–based decay on the weight of the experiences that the Controller uses. This is implemented in the Controller module by discounting the influence of older observations which helps the agent to focus on recent experiences that better reflect the current relationship between different settings of a control parameter and the rate of payoff from using that setting.

Keywords: decay, adaptation, skill acquisition, modeling, ACT-R

Supplemental materials: https://doi.org/10.1037/xlm0001071.supp

Heraclitus said, "Change is the only constant in life." Despite this ever-changing nature of our lives, our success as humans depends on using our past to adapt to the present situation. Psychology has studied transfer in a wide range of domains such as mathematical problem solving (Speelman & Kirsner, 2001), perceptual categorization and discrimination (McGovern et al., 2012), and sensorimotor learning (Goodwin et al., 1998). (Thorndike's (1906) identical elements theory can be applied to many transfer situations – the proposal that transfer occurs to the extent that the same elements are exercised in a new situation as were learned in a prior situation. The challenge in applying this principle can be determining what those elements are. For instance, Taatgen (2013) argued that demonstrations of far transfer from working memory training, which seemed to defy Thorndike's principle, could be explained by transfer of low-level information-processing elements.

Complementary to the broad range of studies on transfer, this paper examines how people adapt existing skills when the situation changes. The need for such adaption pervades our lives. One example, which is close to the experimental task we will investigate, is driving a new car. The car might well have different brake and gas pedal sensitivities. If we performed the exact same motor movements as in our old car we would not succeed. We need to adapt to the changes of the new car. We can do so with some success, but it is also the case that we need some practice with the new car to become as fluent as we were with the old car. Rather than characterizing our adaption as learning new elements, a better characterization seems that we are tuning the existing elements that control our behavior in the new car.

To our knowledge, most studies on such parametric adaptation have focused on sensorimotor perturbations (Braun et al., 2009) or abstract associative tasks (Narain et al., 2014). Within the sensorimotor adaptation literature, one theory that has been gaining traction is that sensorimotor perturbations facilitate structural learning, such that exposure to a wide range of parameter values helps one to abstract a higher–level task structure that can be applied to

Roderick Yang Terng Seow https://orcid.org/0000-0003-0158-8160
Shawn A. Betts https://orcid.org/0000-0002-9143-6269
John R. Anderson https://orcid.org/0000-0002-4977-6630

This research was supported by ONR Grant N00014-15-1-2151 and AFOSR/AFRL Award FA9550-18-1-0251. We would like to thank Dan Bothell for his assistance with setting up the ACT-R software. Data, analyses, and model code are available at https://github.com/roderickseow/adaptation_skill_acquisition/

The data are available at https://github.com/roderickseow/adaptation_skill_acquisition/

Correspondence concerning this article should be addressed to Roderick Yang Terng Seow, Department of Psychology, Carnegie Mellon University, 5000 Forbes Avenue, Pittsburgh, PA 15213, United States. Email: yseow@andrew.cmu.edu

reduce learning time when faced with new parameter values (Braun et al., 2009; Turnham et al. 2012). Similar theories of meta–learning in the motor domain have also been proposed, such as the variability of practice hypothesis (Newell & Shapiro, 1976). Current theories of sensorimotor adaptation also include decay–based processes, allowing an agent to eventually fully adapt to a new environment. For example, the dual–rate theory proposes two simultaneous learning processes, each with a different rate of decay on motor memories (Smith et al., 2006).

The question remains about how these results apply to more complex skills, like driving a new car, that require the integration of many different components. Unlike the tasks in previous studies, success in complex tasks such as driving or playing tennis relies on more than just tuning the control parameters of key actions. One needs to learn how to embed them in a complex interactive task structure. Indeed, to understand how to tune these parameters one needs to understand how they relate to the relevant internal and external states that occur in the game.

To understand the mechanisms that underlie transfer in complex skills, it is useful to begin with a model of complex skill acquisition. Anderson et al. (2019) presented one such model and implemented it within the ACT-R cognitive architecture (Anderson et al., 2004; Anderson & Lebiere, 2014). Their model simulated human learning and performance across two related video games that had different goals, successfully predicting the learning of each but a lack of transfer between them. Critical to performance in either task was learning a fairly complex structure of parameters to that controlled successful performance. Each task stayed constant and so, once the parametric structure had been learned, it could be used for the rest of the experiment. Complementary to their study, this study asks participants to transfer among parametric variations of a video game that maintain the same task goal structure. This allows us to study how people adapt their control parameters to changes within a task and investigate whether the model in Anderson et al. (2019) can be extended to apply to this sort of adaptation.

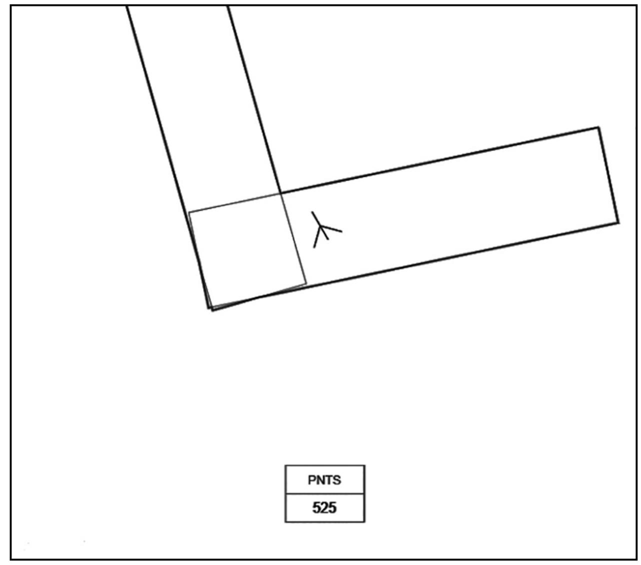
Task—Space Track

Space Track was originally a video game developed by Anderson et al. (2019) as part of their study on the transfer of complex skills. Just like driving, mastering Space Track is a complex skill because it requires one to integrate perceptual, motor, and cognitive components. Expertise arises from having gained an intuitive understanding of the physics of the game and the ability to use that knowledge toward planning sequences of key presses to overcome various situations.

In Space Track, players control a spaceship in a frictionless environment using three keys: thrust (W), rotate clockwise (A), and rotate counterclockwise (D). Rotation only changes the orientation of the ship and does not affect its trajectory. Upon thrusting, the ship's trajectory will be the vector resulting from the sum of the ship's previous velocity and the acceleration in the direction that the ship is oriented toward. The ship's previous velocity significantly influences the resulting trajectory because of the lack of friction; in the real world, friction quickly cancels out the previous velocity once acceleration is no longer provided in that direction.

Figure 1 displays the task screen. Players earn 25 points by successfully navigating the ship along each rectangular track segment and lose 100 points when the spaceship crashes into the walls of the track. Rectangular track segments overlap to form intersections of varying angles (30 to 150 degrees) and directions (left or right facing). The unpredictable nature of the upcoming path adds another layer of difficulty, forcing participants to truly learn to master the controls of the game instead of learning simple perceptual-motor associations. Further details about Space Track are available in Appendix A.

Figure 1
An Example Space Track Screen Rendered in Black and White



Note. Players control the spaceship to cover as much distance as possible along the racetrack while avoiding crashing into the track walls (the bolded edges of the rectangular segments). The overlap between two segments is marked by the lighter colored lines and are not relevant for gameplay. PNTS = points earned in the current trial.

Finding a good speed is crucial for performance – one needs to fly fast enough to cover as much distance as possible but also slow enough to avoid losing control of the ship and crashing. The speed of the ship is determined by pressing a thrust key which adds velocity in whatever direction the ship is moving. The amount of velocity is controlled by how long the key is held down. A unit of acceleration is added each game tick the thrust key is held down. Our experiment changes the amount of acceleration between certain games in a sequence of 40 games. As we will see, this single manipulation requires adapting the complex set of parameters that subjects are learning to play the game.

Below we describe two experiments that involve variations on the same basic design. Simultaneously fitting two experiments gives us generality in our conclusions and add power to our modeling. We will describe the results of the experiments followed by an exploration of how to understand those results within a computational modeling framework.

Experiment 1

Participants in all experiments would participate only after giving online consent, and all procedures and methods were approved by the Carnegie Mellon Institutional Review Board.

Design

To create parametric changes in the task environment, we manipulated the amount of thrust the ship receives for the same duration that the thrust key is depressed. When the thrust key is depressed, a vector of x pixels in the current direction of the ship is added each game tick, which is 1/30th of a second. For the same duration of key press, a game with higher thrust would cause the ship to fly faster than a game with lower thrust. We created three game types, each with a different acceleration. High acceleration games (H) added .6 pixels/tick to the ship's velocity vector for each tick that the thrust key was depressed. Medium (M) and low (L) acceleration games added .4 and .2 pixels/tick, respectively. As mastery of the game relies on adequately predicting and controlling the motion of the spaceship, players would have to retune their control parameters when faced with a different acceleration.

With the three game types, we created five conditions as follows: LLLLM, HHHHM, LHLHM, and HLHLM for training, and MMMMM as a control, where each letter stands for one block of 8 × three-minute games (see Figure 2). For instance, a player in the LLLLM condition would play 4 blocks (32 games) of low thrust followed by 1 block (8 games) of medium thrust. This allows us to investigate how subjects adapt to changes after different amounts of experience in different conditions.

Note that all conditions end with a block of medium games. One of these conditions (MMMMM) involves medium games throughout. Comparing the final block between conditions allows us to assess how much transfer there is from these other sequences to a constant condition. Two of the other conditions involve varied training (LHLHM and HLHLM) while the other two (LLLLM and HHHHM) do not. Within the domain of sensorimotor learning, practicing on varied task parameters has been shown to facilitate adaptation to new task parameters as compared to consistent practice. For instance, when the transfer task is to toss a beanbag to a target at a set distance, Kerr and Booth (1978) reported that participants who trained on different target distances excluding the transfer target perform better than those who trained on just one target distance. Given that the type of practice affects transfer in simple skills, we can also investigate if practice type has a significant impact on transfer in complex skills.

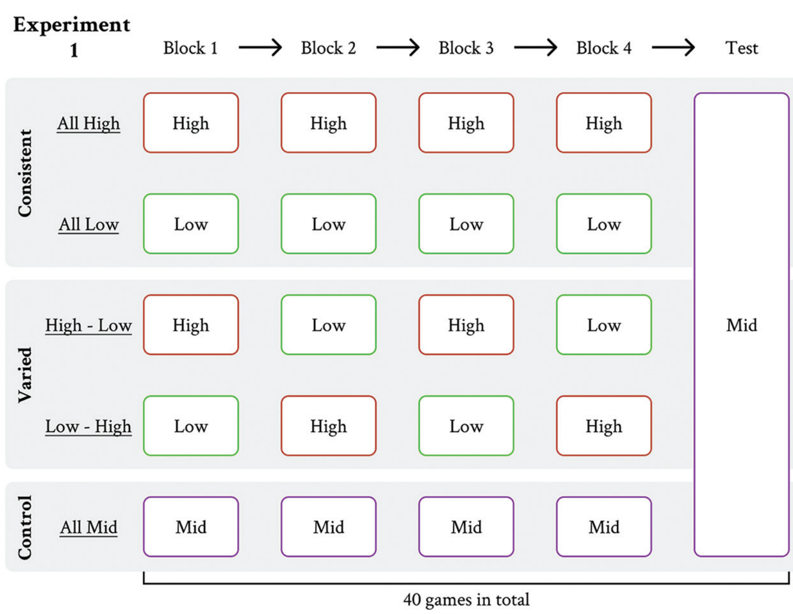
Participants

One hundred participants (73 males), aged 19–65 years (M = 30.8 years), completed the experiment through Amazon Mechanical Turk. Participants were paid $5 for the first session and $10 for the second session plus a bonus of $0.03 per 100 points.

Procedure

The experiment consisted of two sessions. In the first session, participants filled out a demographic questionnaire and read instructions regarding the controls and goals of the game. The instructions also alerted participants to the possibility that they might encounter ships with a different "feel" as indicated by a different color. The participants then proceeded to complete the first 20 games. Before each game, participants would be shown the color of the upcoming ship. Note that no information about the nature of the differences between ships (i.e., magnitude of acceleration) was provided at any time. At the end of the first session, participants that passed a set of inclusion criteria were then invited to the second session, which consisted of another 20 games. During a game, if 20 seconds elapse without the participant pressing a key, a pop up with a "ready to restart" button will appear. These are inclusion criteria for the second session:

Figure 2
Task Design of Experiment 1



Note Each row represents one condition, and each box represents one block of 8 games. See the online article for the color version of this figure.

1. No more than 3 resets due to inactivity.
2. Either at least 500 points in at least 3 out of the 20 games, or that the average of games 17 to 20 is at least 100 points higher than the average of game 1 to 4.

These criteria were put in place to maximize recruiting only players who were sufficiently attentive and showed signs of learning. Using those criteria, 66 players who finished the first session were excluded from participating in the second session (MMMMM: 11; LLLLM: 9; HHHHM: 16; LHLHM: 17; HLHLM: 13). At the end of the first session (game 20), players that were excluded scored −1728.41 on average whereas players that completed both sessions scored an average of 478 on game 20, indicating that the exclusion criteria did distinguish between two distinct populations of players. Recruitment continued until 20 participants per condition successfully completed both sessions.

Experiment 2

Design

The purpose of the second experiment was to examine if varied practice would have a different effect on transfer when the transfer target was beyond the range of practiced values. For experiment 2, we used the High, Medium, and Low thrust game types to create five conditions as follows: LLLLH, MMMMH, LMLMH, and MLMLH for training, and HHHHH as a control. Unlike experiment 1, all training conditions now transfer to the High thrust after training, which is an acceleration outside the range of practiced thrusts for those in a varied training condition. Furthermore, participants in a varied training condition also experienced a smaller range of thrusts (Low to Medium) as compared to those in experiment 1 (Low to High). Figure 3 provides a pictorial representation of the new task design.

Participants

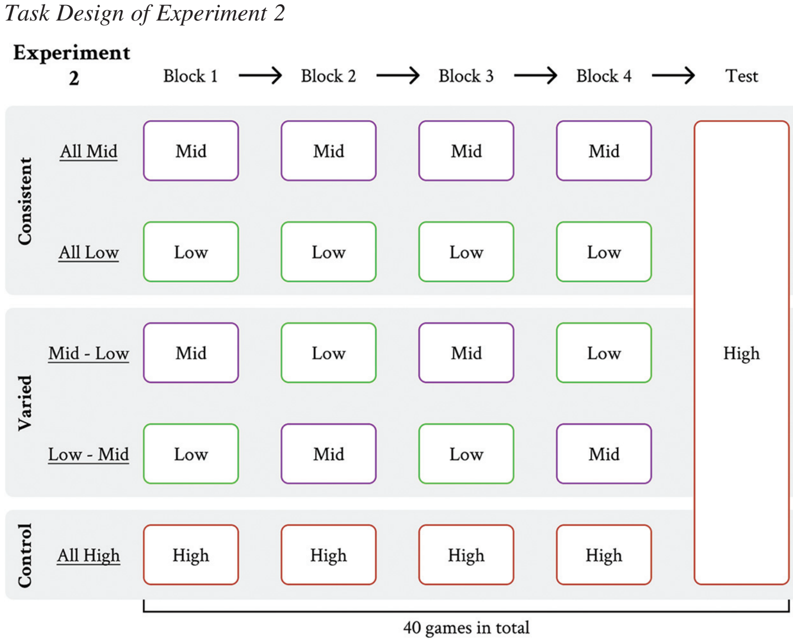
One hundred participants (71 males), aged 19–61 years (M = 32.1 years), completed the experiment through Amazon Mechanical Turk. Participants were paid \$5 for the first session and \$10 for the second session plus a bonus of \$0.03 per 100 points. 63 players who finished the first session were excluded from participating in the second session using the same criteria as before (HHHHH: 15; LLLLH: 11; MMMMH: 11; LMLMH: 15; MLMLH: 11). Similar to experiment 1, excluded players scored −1581.35 on average, whereas players that completed both sessions scored an average of 576 on game 20, again indicative of two distinct populations of players. Recruitment continued until 20 participants per condition successfully completed both sessions.

Behavioral Results

Figure 4 displays the points per game for each condition. On average, human participants (in red) in all conditions improve their performance with increased practice on the task, but also show some temporary drop in performance in some cases where there is a change in acceleration between blocks (For a breakdown of points into the number of crashes and segments cleared per game, refer to Figure 5 and Figure 6, respectively.)

Our first question of interest is whether varied as compared to consistent training facilitates transfer performance. Table 1 presents the mean points across participants and all eight games in the final transfer block. We factorized the four training conditions per experiment (excluding MMMMM and HHHHH) into two factors: varied versus consistent training and the acceleration immediately preceding the transfer block. For experiment 1, a multiple regression

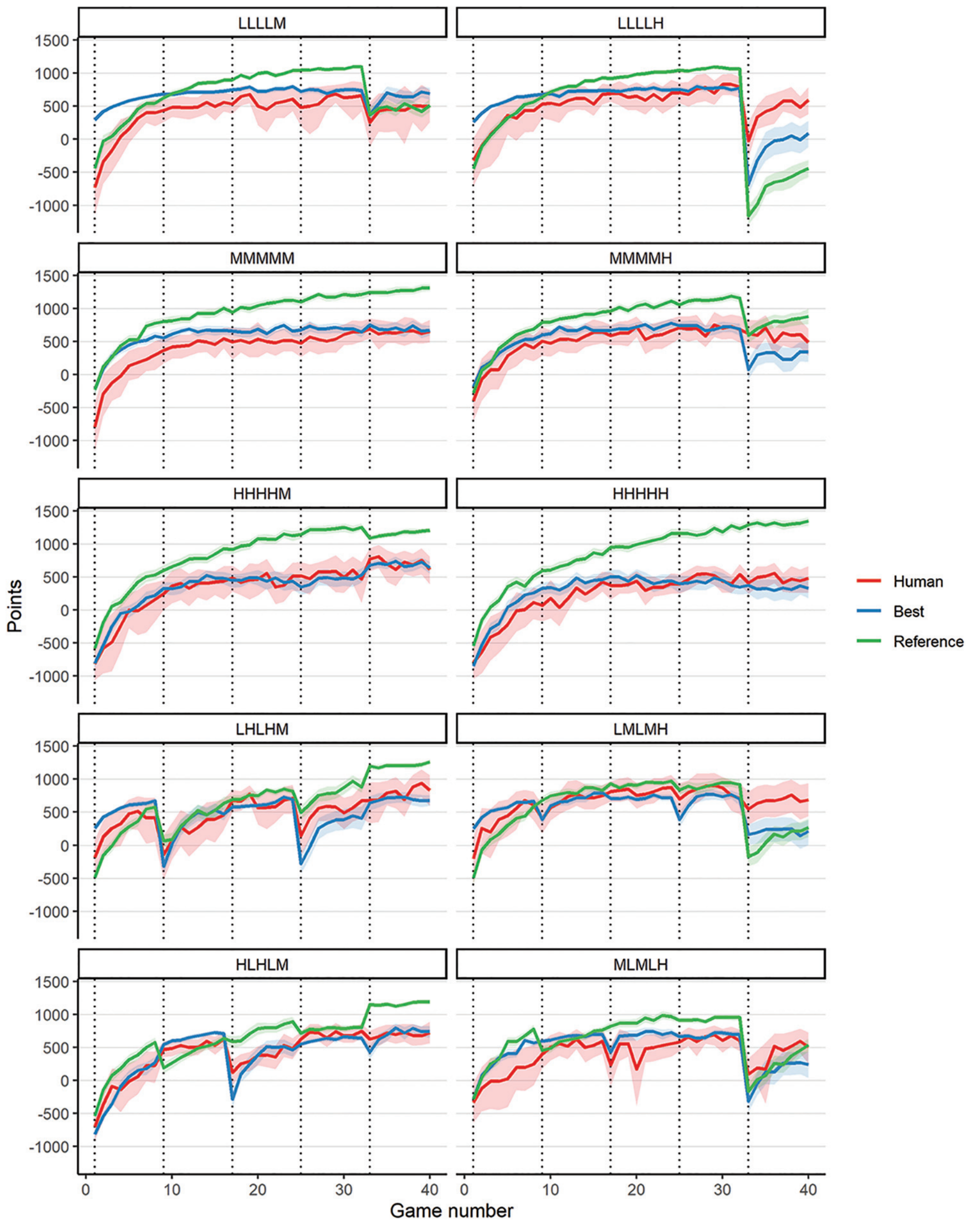
Figure 3
Task Design of Experiment 2



Note. See the online article for the color version of this figure.

Figure 4

Points per Game Across All Conditions for Human Players (in Red), the Best Fitting Model (in Blue), and the Reference Model From Anderson et al. (2019) (in Green)



Note. Conditions from experiment 1 are presented on the left, and those from experiment 2 are presented on the right. Dotted lines indicate start of a new block. Shaded areas reflect bootstrapped 95% confidence limits. M = Medium; L = Low; H = High. See the online article for the color version of this figure.

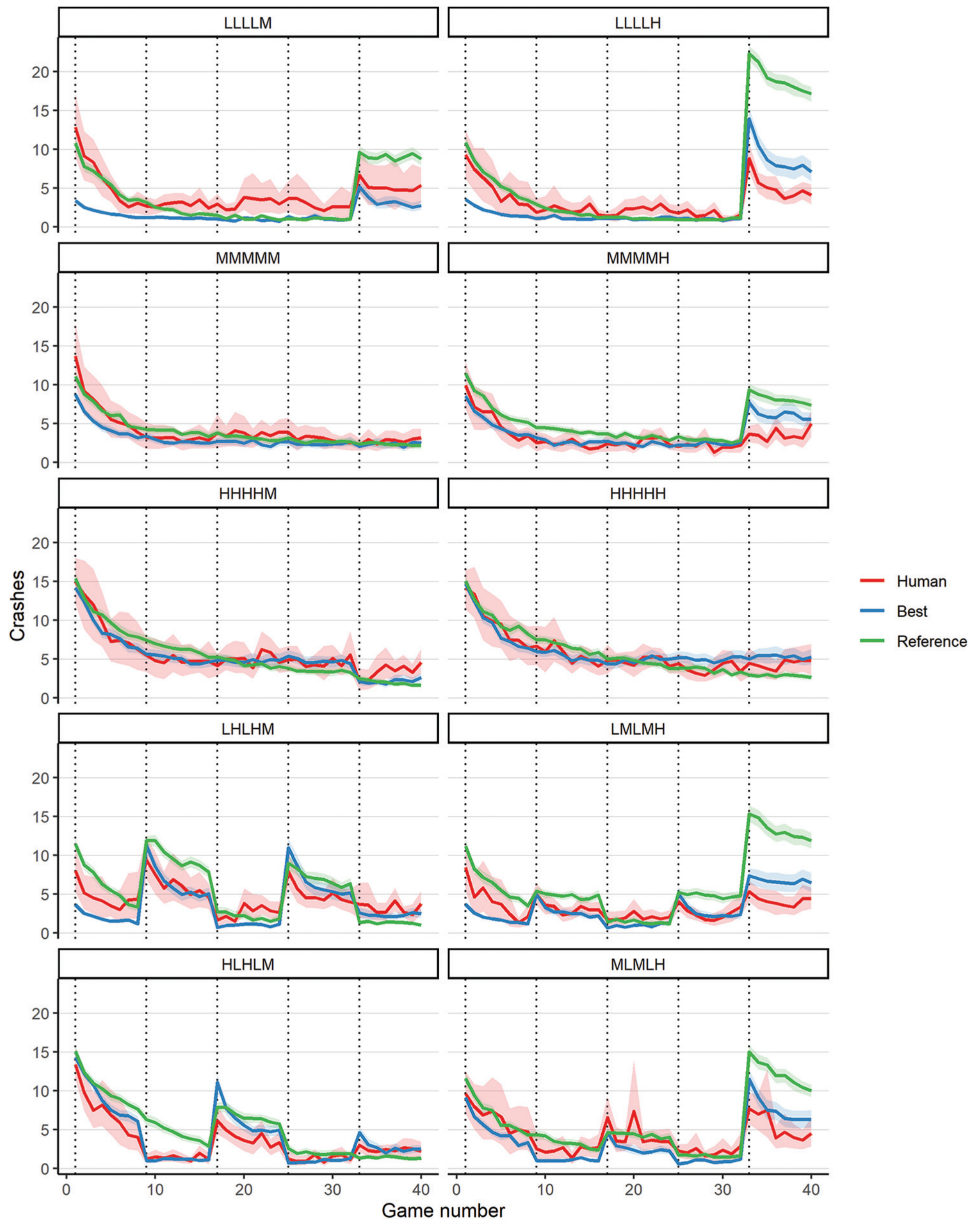
with these two factors revealed that the main effect of varied training was not statistically significant (Varied: $\beta = -84.69$, $SE = 156.27$, $p > .05$). The main effect of preceding acceleration was also not statistically significant, but the negative term suggests that switching from a low block to the final medium block could result in worse transfer than switching from a high block to a final medium block (From–low: $\beta = -266.88$, $SE = 156.27$, $p > .05$).

Experiment 2 yielded similar results. The main effect of varied training was statistically insignificant (Varied: $\beta = 75.62$, $SE = 152.35$, $p > .05$). The main effect of preceding acceleration was also negative but statistically insignificant (From–low: $\beta = -162.66$, $SE = 152.35$, $p > .05$). These results are presented in Table 2.

Taken together, these results suggest that in the context of complex skill acquisition, the level of varied training in our experiment does not facilitate transfer performance compared to consistent training. We are hardly the first to report a small to null effect of varied training on transfer performance. Previous studies have also reported a null effect (Van Rossum, 1990; Tuitert et al. 2017), or a

Figure 5

Crashes per Game Across All Conditions for Human Players (inRed), the Best Fitting Model (in Blue), and the Reference Model (Iin Green)



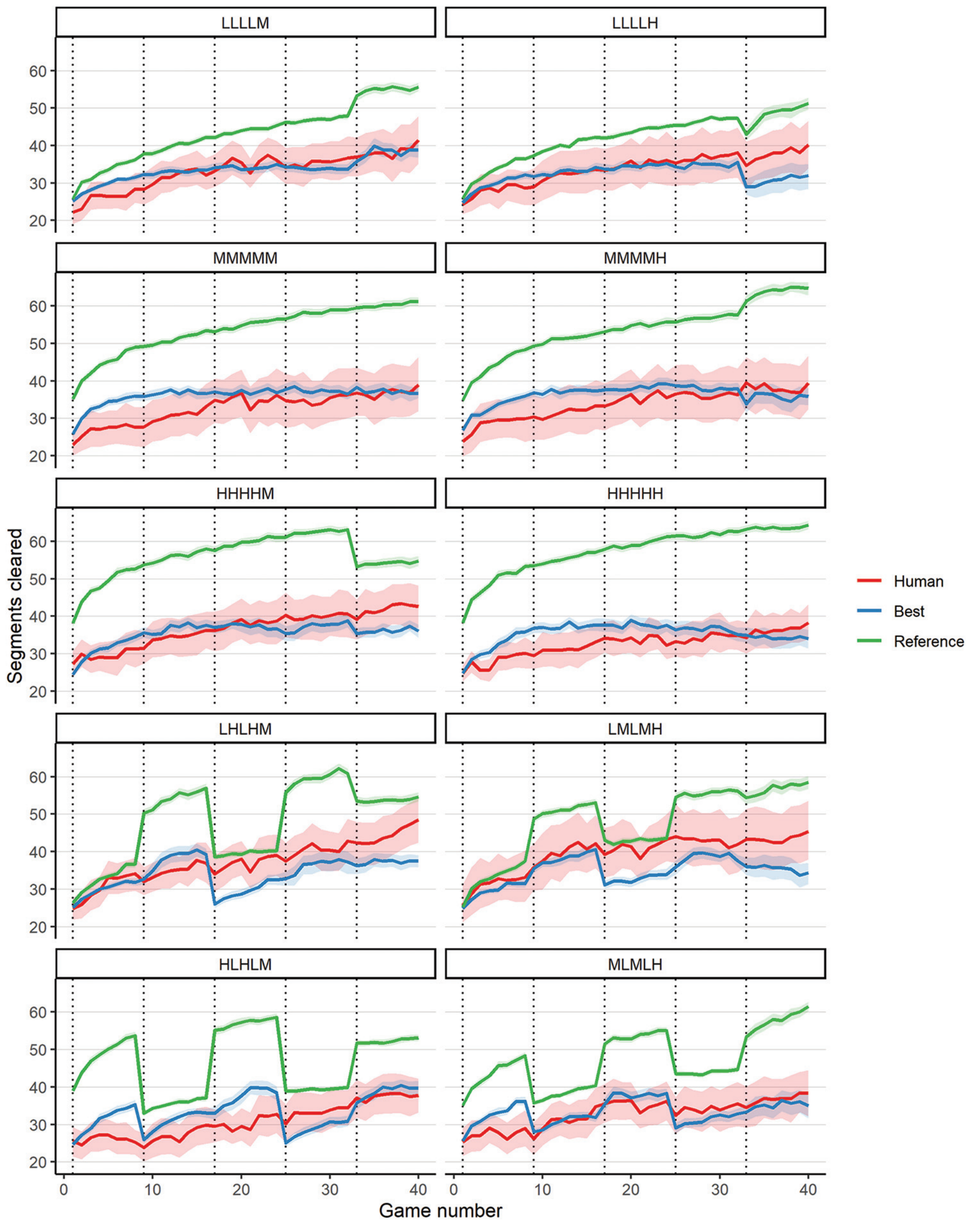
Note. Conditions from experiment 1 are presented on the left, and those from experiment 2 are presented on the right. Dotted lines indicate start of a new block. Shaded areas reflect bootstrapped 95% confidence limits. M = Medium; L = Low; H = High. See the online article for the color version of this figure.

limited effect of varied training on transfer (Brady, 2008; Cardis et al. 2018; Newell, 2003; Willey & Liu, 2018). However, it is possible that the amount of variation in acceleration is overshadowed by the inherent amount of variability that players experience even when playing games with the same acceleration, thus nullifying any benefit from being having practiced multiple accelerations. For instance, track corner angles per game are drawn at random from a range of 30 to 150 degrees.

Another possibility is that the participants in the varied conditions did not experience a sufficient degree of interleaving. According to Schmidt's schema theory, the development of a schema (an abstract representation of a class of related motor actions) is facilitated by an increased degree of variability in prior experiences (Schmidt, 1975). Empirically, in some studies that found an advantage for varied training, the key environment variable would be changed between individual trials (Catalano & Kleiner, 1984; McCracken & Stelmach, 1977).

To address this, we conducted a follow–up experiment with four conditions: LHLHM, HLHLM (block–level varied training), and two conditions that alternated every trial between high and

Figure 6
Segments Cleared Per Game Across All Conditions for Human Players (in Red), the Best Fitting Model (in Blue), and the Reference Model (in Green)



Note. Conditions from Experiment 1 are presented on the left, and those from Experiment 2 are presented on the right. Dotted lines indicate start of a new block. Shaded areas reflect bootstrapped 95% confidence limits. M = Medium; L = Low; H = High. See the online article for the color version of this figure.

low accelerations for the first 32 trials before switching to medium acceleration for the final block of 8 trials (trial–level varied training). The recruitment and experimental procedures were the same as those of experiment 1. Participants who received trial–level varied training (*M* = 531.72, *SD* = 1095.51) did not perform differently from those who received block–level varied training (*M* = 543.51, *SD* = 602.81; *t[60.63]* = .060, *p* = .95). These results reinforce our conclusion that at least within the context of our experiment, it appears that varied training has little benefit for transfer performance.

Aside from the null findings regarding the effect of varied training, the previous analyses suggest that entering the transfer block from low acceleration might negatively impact points as compared to entering from medium or high acceleration. Thus, we examined in greater detail the effect of switching accelerations between blocks and other task–related factors that could account for the wide range of performance observed such as the total amount of practice and the current acceleration of the ship.

Using performance data from both experiments, we fit a linear mixed model to the average points per block for each participant

Table 1
Mean Points on Final Transfer Block

	Experiment 1		Experiment 2	
Training variability	From Low	From High	From Low	From Mid
Consistent	LLLLM 440 (695)	HHHHM 707 (421)	LLLLH 430 (431)	MMMMH 593 (432)
Varied	HLHLM 692 (276)	LHLHM 791 (491)	MLMLH 382 (508)	LMLMH 669 (546)

Note. M = Medium; L = Low; H = High. Standard deviations are displayed in parentheses.

using the natural logarithm of the block number *(log[block])* and acceleration. We hypothesized that performance would improve with more practice, but at a decreasing rate as players approach a performance ceiling. As higher accelerations entail more sensitive controls and more difficult games, we also hypothesized that performance would decrease with increasing acceleration. To account for possible effects of prior skill and other individual differences on performance, we included participant ID as a random effect. Consistent with our hypotheses, we observed both a practice effect of block number *(log[block]:* $\beta = 395.53$, $SE = 13.11$, $p < .01$) and a difficulty effect of acceleration ($\beta = -871.58$, $SE = 58.89$, $p < .01$; refer to the left column of Table 3).

Given these practice and difficulty effects on performance, is there an additional effect of switching between games of different acceleration? In domains where optimal performance depends on precise sensorimotor coordination, studies have found support for error–based learning mechanisms that gradually updates one's action parameters with experience (Berniker & Kording, 2011; Izawa & Shadmehr, 2011; He et al. 2016). In the context of Space Track, these theories entail that a player's actions in a new acceleration would be initially based on their experiences with the previous acceleration. Actions that were optimal in one acceleration are likely to be suboptimal for a different acceleration, which might cause players to suffer a temporary hit in performance when switching accelerations. We further hypothesize that this cost to performance is asymmetrical, where switching to a higher acceleration would impair performance, while switching to a lower acceleration would have little to no effect on performance. A player used to a low acceleration would likely end up with too fast a ship when switching to a high acceleration game and thus experience more crashes. In contrast, while a player used to a high acceleration might find themselves with too slow a ship when switching to a low acceleration game, it is much simpler and less costly to speed the ship by issuing more thrusts.

To investigate the additional effect of switching accelerations on performance, we conducted a linear regression on points after adjusting for the previously estimated effects of practice, acceleration, and individual prior skill (the adjusted points are displayed in Figure 7). To test our hypothesis we used two predictors, one to measure the effect of switching faster and one to measure the effect of switching slower. The predictor values for switch–faster were .4 for a switch from L to H, .2 for a switch from L to M or from M to H, and 0 otherwise. The predictor values for switch–slower were .4 for a switch from H to L, .2 for a switch from M to L or from H to M, and 0 otherwise. While averaging over the games within each block provides a sufficient summary for estimating practice and difficulty effects, it has the disadvantage of smoothing over the performance changes observed within a block. After a change in performance that follows from switching to a new acceleration, players recover relatively quickly within the span of 8 games. Consequently, performance changes attributable to switch effects might become diluted or confounded when using block–average data. For this analysis, we also excluded data from block 1 as switch type is undefined for the first block. In sum, only points earned during the first game of blocks 2 to 5 (i.e., games 9, 17, 25, and 33) were used to estimate the effects of switching acceleration on performance.

Table 2
Regression Models on Transfer Block Points With Varied Training and Preceding Acceleration Effects

Factor	Experiment 1 (*SE*)	Experiment 2
Varied	84.69 (156.27)	75.62 (152.35)
From–low	–266.88* (156.27)	–162.66 (152.35)
Varied:From–low	167.97 (221.00)	–124.22 (215.45)
Intercept	706.56*** (110.50)	593.13*** (107.73)
Observations	80	80
R^2	0.07	0.06
Adjusted R^2	0.03	0.02
Residual *SE*	494.2 (df = 76)	481.8 (df = 76)
F statistic	1.88 (df = 3; 76)	1.567 (df = 3; 76)

* $p < .1$. ** $p < .05$. *** $p < .01$.

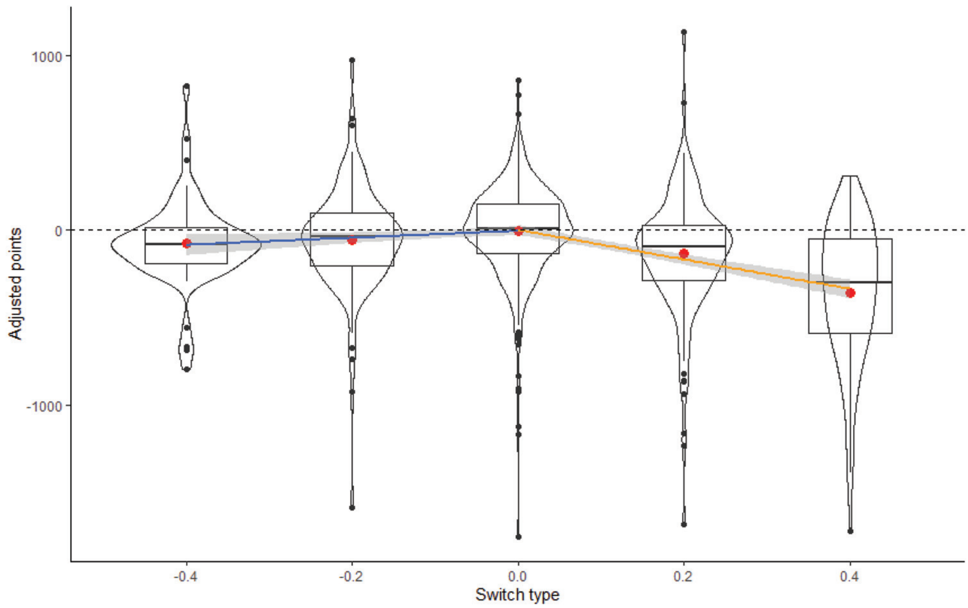
Table 3
Linear Mixed Models on Average Points per Block With Practice and Acceleration Effects

Factor	Human players (*SE*)	Model players
log(block)	395.93*** (13.11)	193.47*** (8.37)
Acceleration	–871.58*** (58.89)	–1,241.806*** (30.44)
Intercept	423.93*** (38.46)	811.69*** (14.40)
Observations	1,000	5,000
Log likelihood	–7,139.0	–36,204.7

* $p < .1$. ** $p < .05$. *** $p < .01$.

Figure 7

Violin and Box Plots of the Adjusted Points on the First Game of Blocks 2 to 5 by Switch Type



Note. Increasing acceleration corresponds to a positive switch type, while decreasing acceleration corresponds to a negative switch type. Box plot whiskers are drawn up to the data points that fall within 1.5*IQR from the upper and lower quartiles. Red circles indicate the average for a given switch type. The blue and orange lines indicate the predicted values for games with decreased and increased accelerations, respectively. The dashed line marks an adjusted point value of 0, which is expected if performance on a game is entirely accounted for by the practice and acceleration effects estimated from the previous regression model. See the online article for the color version of this figure.

Consistent with our hypothesis, switching to a higher acceleration predicts an additional decrease in performance after accounting for the practice and difficulty effects (Switch–faster: $\beta = -838.69$, $SE = 80.66$, $p < .01$). Switching to a lower acceleration also predicts an additional, albeit lesser, decrease in performance (Switch–slower: $\beta = -218.42$, $SE = 95.95$, $p < .05$; refer to the left column of Table 4). A comparison between this regression model and a restricted one (switch–faster = switch–slower) confirms that the two coefficients are different ($F[1, 797] = 33.87$, $p < .001$). Concretely, this implies that players switching from a low to a high thrust game would earn 335.48 points less than players who had been playing high thrust games. In contrast, players switching from high to low thrust would only earn 87.37 points less than players who had been consistently playing low thrust games.

Table 4

Linear Models on Block–Initial Residual Points With Switch Effects

Factor	Human players (*SE*)	Model players
Switch–faster	–838.69***	–1,741.43***
	(80.66)	(55.94)
Switch–slower	–218.428*	–673.89***
	(95.95)	(66.55)
Intercept	0.9,478	77.81***
	(14.61)	(10.14)
Observations	800	4,000
Adjusted R^2	0.118	0.195

While there are variations in how well participants adapt to a change in acceleration, the overwhelming effect is one of positive transfer. For instance, the 40 participants experiment 1 who played high acceleration in the first block scored an average of −170 points, while the 80 participants in experiment 2 who first played high in their last block scored 519 points. Likewise, the 40 participants in experiment 2 who played medium acceleration in the first block scored 86.6 points on average while the 80 participants in experiment 1 who first played medium in their last block scored 657 points. We now turn to a model that explains this overall positive transfer plus the fixed effects described in the regression analyses.

Modeling Learning

Having characterized human learning and performance on the task, we now turn to our work on modeling the processes by which such learning occurs, with a particular focus on how people adapt to parametric changes in the task environment. In a recent study, Anderson et al. (2019) demonstrated that an ACT-R model produced a learning trajectory similar to what humans do ($r = .96$) in a Space Track task where players played 40 games at a acceleration of .3. Given the success of this approach, we used that model as a reference model for our investigations. In the following section, we will briefly describe the key features of the reference model and the ACT-R cognitive architecture.

One advantage of testing theories of any psychological process within a cognitive architecture such as ACT-R is the constraints that the architecture imposes on the theory. In skill acquisition, there are limits on various human cognitive processes such as attention and response times that constrain how humans learn and perform on a task. The ACT-R cognitive architecture incorporates realistic performance constraints on the speed and accuracy of perception and action, which are based on past literature and successfully used by many models developed within the ACT-R architecture.

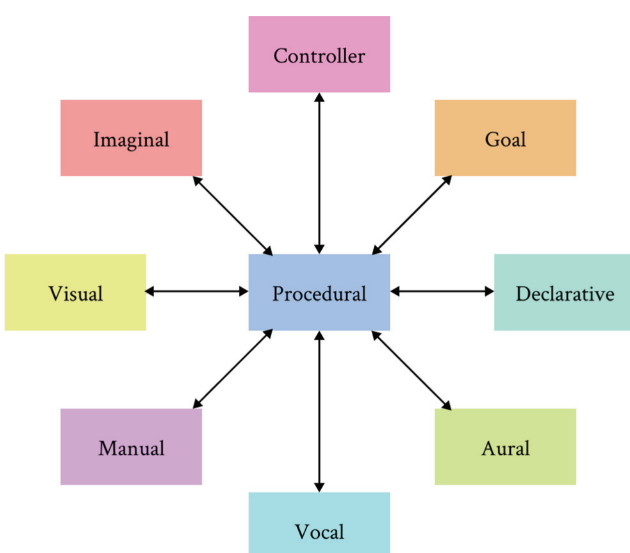
ACT-R's architecture comprises of modules and buffers (see Figure 8). Modules are independent computational systems that can run asynchronously, and they communicate with each other via buffers that can hold a limited amount of information. A central piece of the architecture is the Procedural module. The Procedural module consists of production rules, which are general pattern-action associations that are built up over experience. It takes as input information from the buffers of other modules, pattern matches those pieces of information to production rules, and subsequently issues commands to the other modules as per the matched production rule. Thus, the Procedural module is heavily involved in the coordination of activity across modules and buffers.

The Declarative module stores the game instructions as operators and places the most appropriate operator into the Retrieval buffer when it receives a request from the Procedural module. A single retrieval takes approximately 100ms in the Anderson et al. (2019) model.

Other modules and buffers serve more specialized functions. The Game State buffer is part of the Visual module, and stores information about the visual game state – for example, the position and the orientation of the ship. The Goal buffer stores information about various control parameters, like the desired speed to fly at. ACT-R also possesses a Manual module that is based on the EPIC model (Kieras & Meyer, 1997); the Manual module is responsible

Figure 8

The Module Structure of ACT-R



Note. Boxes represent independent modules that process information in parallel. This is updated from the corresponding figure in Anderson et al. (2019) to reflect the addition of the Controller module. ACT-R = Adaptive Control of Thought - Rational. See the online article for the color version of this figure.

for executing keypresses and is constrained with an average of 250ms lag between consecutive keypresses.

Human participants do not begin learning from scratch but are informed by explicit instructions about the controls and goals of the task. Through *instruction following,* ACT-R models also utilize task knowledge to accelerate learning in the initial stages. Such task knowledge is represented in ACT-R's declarative memory as operators, which are triplets of state-action-newstate. Together, these operators represent the overall strategy a player might use, constituting a decision tree where each node corresponds to a state and each branch is the outcome of an action.

As instructions are represented as declarative knowledge, they need to be interpreted by production rules. In ACT-R, it takes roughly 100ms for retrieval and 50ms for the execution of a production, which totals to 150ms for the interpretation of a step of declarative instruction. Multiple operators need to be retrieved to determine what action to execute next (see Appendix B). To execute a physical action (e.g., a keypress), additional time is needed for motor preparation and execution. Thus, if the model is interpreting these operators, the total time to get from the start of a strategy to a physical action could be more than half a second. While a model that relies solely on instruction following is able to play Space Track from the start, it will perform poorly due to its slow response time.

For the model to successfully earn points on a fast-paced task, it needs to improve its response time. Human players demonstrate improvement in performance with experience and this is partially governed by increased automaticity and faster deployment of knowledge. Instead of having to retrieve declarative knowledge at each step of the strategy, people are able to eventually respond directly to the situation. ACT-R models capture this by *production compilation* (Taatgen & Anderson, 2002), a process that gradually eliminates the retrieval of declarative instruction and replaces it by rules that directly respond to the state of the environment.

Control Tuning

Figure 8 shows a new module (the Controller module) that was added to the ACT–R architecture in Anderson et al. (2019). This module was responsible for learning various parameters that control actions such as how long to thrust. Since this module is new and critical to understanding the results in our experiment, we will describe it in some detail. More information on how the Controller module interacts with the other ACT–R modules can be found in Appendix D.

The basic strategy of the model (see Figure B1) is to fly as fast as is safe down a rectangle. While doing so it prepares to make a "hard turn" at the intersection by turning the ship to an angle that bisects the angle between the two rectangles. When it gets to the new rectangle it presses thrust until it is flying down the new rectangle. Then it performs any flight adjustments before preparing for the next turn. Successfully executing this strategy requires learning the optimal values for 5 control parameters:

1. *Ship speed.* The ideal speed of the ship when flying down a rectangle.

2. *Press point.* To make a hard turn, the model needs to learn where the ship should be in the overlap between two rectangular segments before initiating the thrust.

3. *Stop angle.* As the thrust key is held down during a hard turn, the ship changes its flight direction toward that of the new rectangle. The model cannot wait until the directions match up because it takes time to lift its finger. Hence, it needs to learn how close to the desired angle the flight direction should be to lift its finger.

4. *Tap thrust duration.* At various points, the model must issue brief taps to adjust the ship speed or the flight direction and it needs to learn the duration for such thrusts.

5. *Aim.* The model does not have to be precise in achieving a desired aim for the ship, but it needs to learn how close is good enough.

For each control parameter, the model samples values within a preset range and evaluates the payoff for the sampled values according to relevant feedback. As point-related events in Space Track are temporally sparse, the model samples a particular control value for some period of time (n) and divides the total payoff by the period to arrive at the average rate of payoff (y_i):

$$y_i = \frac{payoff_{total}}{n} \tag{1}$$

While it is possible to sample enough and collect highly accurate average payoff rates for all possible values of a control parameter, this is a highly inefficient method of learning the optimal control value. By generalizing across past experiences, an agent can converge upon the optimal value without such exhaustive sampling by positing a relationship between values and payoffs. Humans have demonstrated the ability to perform such generalization, as revealed by relatively successful extrapolation and interpolation on function learning tasks (DeLosh et al., 1997; Kalish et al. 2004).

Lucas et al. (2015) argue that humans tend to assume simple functions when positing relationships between two continuous dimensions. The ACT-R controller assumes a quadratic function relating payoff to reward, as this is a simplest polynomial function that possesses a unique maximum, which provides an estimate of the optimal control value. The quadratic function is estimated via quadratic regression (more details can be found in Appendix C), and is then transformed into a probability distribution using the softmax equation:

$$P(x_i) = \frac{e^{V(x_i)/T}}{\sum_j e^{V(x_j)/T}} \tag{2}$$

where x_i and x_j refer to control values, V refers to the quadratic function which predicts the rate of return for a given control value, and T refers to a temperature parameter that decreases over time. At high temperatures the model tends to choose a wide range of values and at low temperatures it tends to concentrate its choices on the best performing values. Many models from reinforcement learning have successfully varied the temperature from high to low to capture the tendency to explore less with increasing experience (Christakou et al., 2013; Decker et al. 2016). In the ACT-R implementation T is an inverse function of experience: $T = A/time$. Thus, the model initially explores a range of values updating its estimate of their payoff. The model ideally eventually converges to a true estimate of the relationship between payoff and control values and just uses the highest performing value. (See Figure 9 for an illustration of the process described above.) Note that this is a strategy designed for a constant environment. When the environment changes its former estimate of an optimal value may no longer be correct.

Ratio of Source Weights

For all the models that will be discussed, the rate of payoff for each sampled control value is computed as a weighted combination of the number of segments cleared and the number of crashes:

$$payoff = W_{good} segments + W_{bad} crashes \tag{3}$$

where W_{good} is a positive weight given to segments cleared and W_{bad} is a negative weight given to crashes. The size of these two weights determines the contribution of each source of feedback to the estimated payoff.

While Space Track implements a 1:4 ratio for points earned per cleared segment (+25) to points lost per crash (−100), players could plausibly deviate from this ratio when calculating the payoff for control values. For example, different ratios of source weights potentially relate to differences in risk attitudes; a 1:1 ratio player could favor higher ship speeds, which allows them to clear more track segments at the risk of crashing more often than a more cautious player.

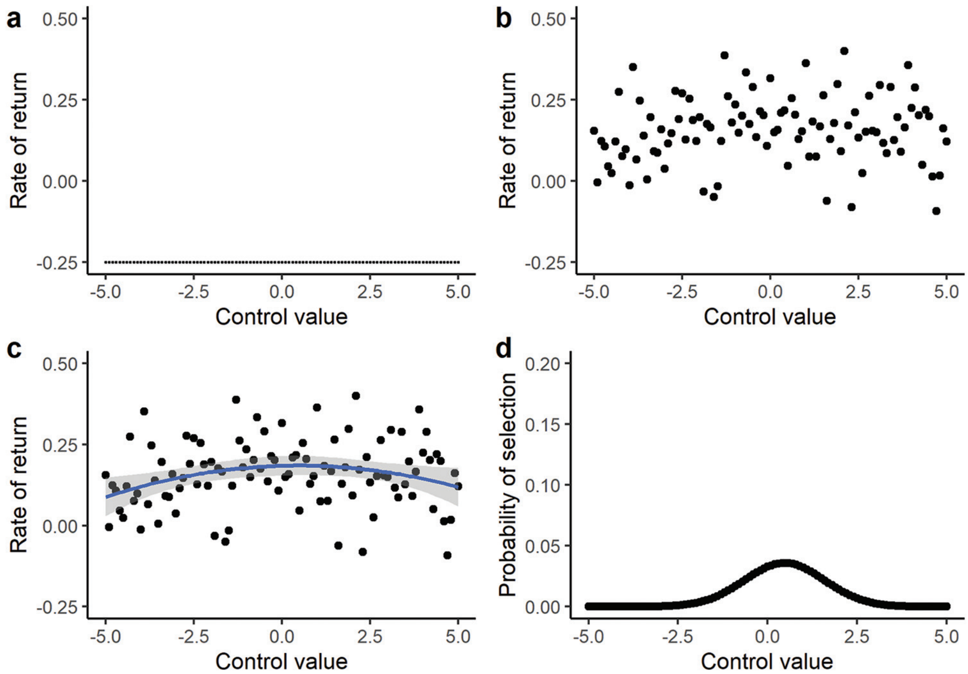
The reference model (from Anderson et al., [2019]) weights both sources equally (+1 per cleared segment, −1 per crash). To investigate how the average human player weights these sources, we tested two other ratios: a 1:4 ratio that reflects the point allocation according to the instructions, and a 1:8 ratio that reflects a greater penalty on crashing than warranted by the point allocation. This increased emphasis on crashes is aligned with the more general principle of loss aversion - in uncertain situations, people prefer to avoid losses over acquiring gains of equivalent value (Kahneman & Tversky, 1979).

Decay and Maintenance

The mechanisms described above assume that all experiences are factored equally into computing the optimal control value. However, it is plausible that the influence of older experiences decay with time, thus leading the agent to prioritize information from recent experiences. Decay has been studied and described in various human faculties. For one, gradual decay is characteristic of human memory; as (a translation of) Ebbinghaus (1885/1913) puts it, "(left) to itself, every mental content gradually loses its capacity for being revived." Decay has also been studied in the motor domain; studies of human motor skill have demonstrated that error-driven adaptation gradually decays back to baseline when feedback is removed (Codol et al., 2018; Ingram et al. 2013; Smith et al. 2006).

It is important to note that our use of the term "decay" does not entail a specific mechanism that underlies the declining influence of older experiences; our theory does not distinguish between decay as the disappearance of experience traces from memory storage (Loftus & Loftus, 1980) or as the decreased capacity for

Figure 9
How the Controller Module Learns



Note. (a) Choices are initially uniformly distributed between a range of preset values. (b) While trying out a chosen value, the Controller module records the associated weighted number of segments cleared and number of crashes experienced. It does this for all sampled values. (c) Quadratic regression over samples. (d) The Controller module converts the quadratic estimated from the sampled values into a probability distribution via the softmax equation. The next choice is drawn from the new distribution. See the online article for the color version of this figure.

memory retrieval (Averell & Heathcote, 2011). Both accounts of decay would be consistent with our theory.

While there are multiple possible ways to implement decay in an agent, one commonly used function is to discount experiences as an exponential function of time (Heathcote et al. 2000; Rubin et al., 1999):

$$m = \theta^{t} \tag{4}$$

where θ is a maintenance parameter that determines the rate of decay (ranges from 0 to 1), t is the number of seconds since the start of the game, and m is a weighting factor applied to average rate of payoff associated with a prior experience. Smaller values of maintenance (θ) entail that less information is maintained from prior experience at every subsequent time step, which reflect faster rates of decay.

The reference model has no decay and so effectively sets θ to 1, which reflects an agent that equally weights all experiences regardless of how much time has passed since an observation. Given the lack of acceleration changes in the experiment in Anderson et al. (2019), it is reasonable to expect that there is little to no advantage to setting θ to a value less than 1.

However, when there are changes in the task environment, it is likely that the same control value results in different payoffs as time progresses. Thus, when there are changes in the environment, less maintenance of older experiences possibly endows an agent with greater adaptability. Less relevant experiences from a previous environment would be weighted less, which allows the agent to focus on learning what is best for the current state of the system.

In our experiment, players have to adapt to changes in accelerations, potentially by adjusting their estimates of what the optimal control values are. For instance, a tap thrust of .1 seconds at a low acceleration increases the the ship's velocity by a smaller amount than a tap thrust of .1 seconds at a high acceleration. To maintain the same average speed when switching from a low to a high acceleration, a player would then need to reduce their estimate of the optimal tap thrust duration.

Approach

To investigate the mechanisms that underlie adapting to changes in acceleration, we compared the performance of models that varied along multiple dimensions to the performance of the average human player. While we mainly focused on the ratio of source weights and maintenance, we also tested variations on a pair of task-specific components of the model.

The first variation concerned a strategy for slowing the ship down if it was flying too fast. The ship can be slowed down by rotating the ship by 180° with respect to its current velocity, and then thrusting in the new orientation to reduce its speed. An analysis of frame by frame changes in ship speeds and keypresses revealed that participants would indeed sometimes execute such a maneuver to sharply decrease their ship speed, particularly when approaching track corners. We call this alternative strategy "slowdown."

The second variation concerned the range of tap thrust durations. The original model searched for an ideal value in the range from 67 ms to 167 ms. However, at our higher accelerations very brief tap thrusts were optimal. Therefore, the alternative involved making the minimum duration 10 ms and having the model instead explore the range from 10 to 167 ms.[1]

We tested variations across these four dimensions: source weight ratio, maintenance, slowdown, and minimum thrust time. For each combination of values, we simulated 100 runs for each of the 10 experimental conditions. These simulations were compared participant behavior for every condition (i) and game number (j), yielding 400 squared differences between the average model measure and the corresponding average participant measure:

$$\epsilon_{i,j}^2 = (p_{M,i,j} - p_{H,i,j})^2 \tag{5}$$

$$SSE = \sum_{j=1}^{10} \sum_{i=1}^{40} \epsilon_{i,j}^2 \tag{6}$$

The sum of these squared differences was then divided by the average squared standard error of the participant mean to produce a measure of goodness-of-fit:

$$SE_{total}^2 = \sum_{j=1}^{10} \sum_{i=1}^{40} \frac{Var(X_{i,j})}{20} \tag{7}$$

$$SE_{avg}^2 = \frac{SE_{total}^2}{400} \tag{8}$$

$$fit = \frac{SSE}{SE_{avg}^2} \tag{9}$$

While points are the primary indicator of performance on the task, two players could conceivably achieve the same total points through different strategies. For instance, a player might adopt a riskier approach to the game, flying faster to clear more track segments but also crashing more often than a more cautious player. We thus chose to compare the fits between models and participants using a composite measure computed from both the number of segments cleared and the number of crashes:

$$fit_{total} = fit_{rectangles} + fit_{crashes} \tag{10}$$

Model Comparisons

We fit the 36 models created by crossing the three weightings of crashes by 3 decay choices by 2 slowdown options by 2 minimum thrust times. Figure 10 provides a characterization of where the best model lies by showing how various choices improved the fit over the reference model (for a full comparison across all 36 models, see Appendix F). From the reference model, the largest improvement in fit came from including the ability for the model to slow down (7135). At the next step, changing the ratio of source weights provided the greatest improvement. While the 1: 8 model yielded a better fit than the 1: 4 model (3782 to 3823), the relatively small difference in fit warranted further investigations along both branches of modifications. Along the *weight* = 8 branch, the most effective improvement came from adding an exponential decay with the maintenance parameter set to .995 (2596). Note that both the presence of decay and the value of the maintenance parameter are important; setting the parameter to .5 drastically impairs the fit (45559). From the θ = .995 model, the remaining unexplored modification was to reduce the minimum thrust time available to the controller from 0.067s to 0.01s. However, this modification slightly impaired the model's fit (2972).

Following the *weight* = 4 branch, the most effective improvement also resulted from adding θ = .995 (2644). Unlike the *weight* = 8 branch, further reducing the minimum thrust time to 0.01s improved the model's fit (2358). This model also has the best fit of all 36 models (the next best is also shown in Figure 10, the one with −8 wt, .995 decay, slowdown, 0.067s minimum thrust time), which suggests that the best fitting model thus far possesses the following features:

1. The ability to slow the ship down if the ship speed is higher than some threshold

2. Weighting feedback from segments cleared to crashes in a 1 to 4 ratio which corresponds to their weighting in determining score

3. Applying an exponential decay to past experiences at the rate of 0.995^{time}

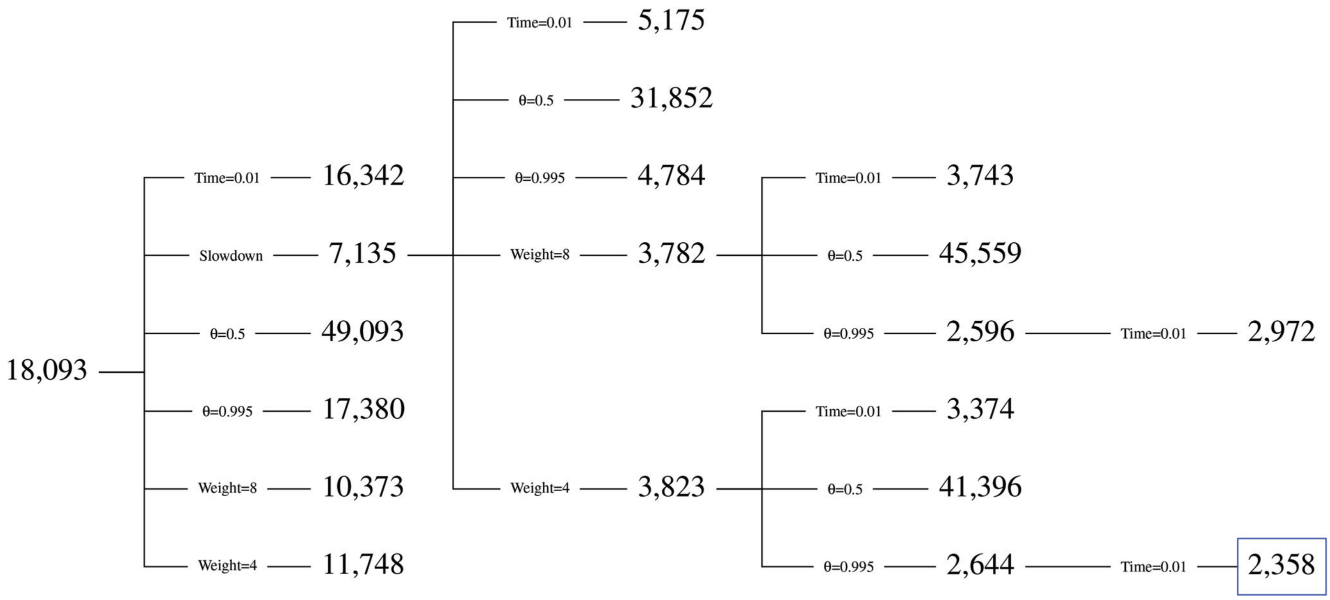
4. The possibility of issuing a tap thrust of 10ms + baseline motor processing time costs

While the best model includes exponential decay, the comparison between the θ = .5 and θ = .995 models suggest that too much discounting is both detrimental to performing well on the task and uncharacteristic of the mechanisms that underlie human adaptation. Further model simulations over a range maintenance values confirms this hypothesis.

Referring to Figure 11, models with maintenance values at .5 or even .9 provide fits that are an order of magnitude worse than the model without decay (θ = 1). Models with maintenance values in the range of .99 to .999 all demonstrate better fits than the reference model, but the improvement is relatively smaller. Of the models tested, the best fitting model set θ to .992, which is lower than the model we identified previously (θ = .995). While this appears to be a minute difference in parameter values, this difference has large implications for the weighting of experiences. For a model where θ = .995, an experience that starts with a weight of 1 when it is first observed would be weighted at .995180 = .406 by

[1] A recent unrelated experiment using the Space Track task but run at 60 frames per second revealed that participants were indeed capable of pressing and releasing the thrust key in the same game tick (∼16ms). While we did not record finger movements, we assume that for a recorded key press duration of one game tick, the participant's finger was making some movement with a downward trajectory just sufficient to engage the key for less than one game tick. It should also be noted that when the key press duration is set to 10 ms in ACT–R, because of motor noise, the key can fail to be depressed (like a finger movement failing to engage the key) if the final duration is 0 ms or less.

Figure 10
Model Fits to Participant Data

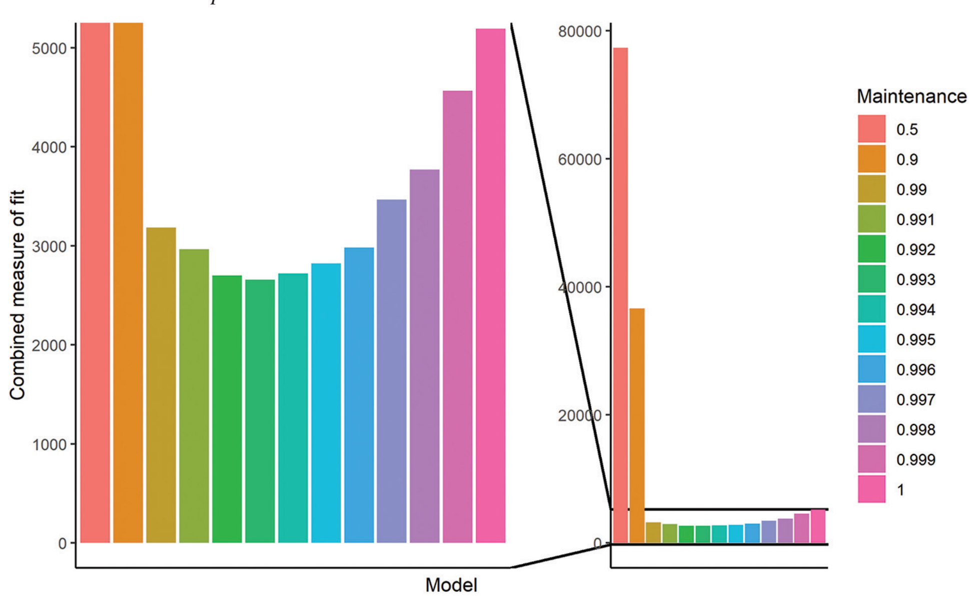


Note. Fits were computed using the squared difference between models and humans adjusted by the squared standard error of the mean. Smaller numbers reflect better fits. Each branch describes the modification made to the parent model. The best model is indicated in a blue box. See the online article for the color version of this figure.

the end of a three-minute game. When θ = .992, the same experience would only be weighted at .992180 = .236 at the end of the same period.

A visual inspection of the performance of the best–fitting model as compared to the reference model (refer to Figures 4, 5, and 6) confirms that the θ = .992 model does a much better job of simulating the group level variations observed in human performance, particularly following a switch in acceleration between blocks. Further discussion of the model's ability to capture within and across participant variability can be found in Appendix G. Applying the regression analyses used with the empirical data to the best model's points, we found practice, acceleration, and switch effects that were similar to those estimated from the empirical data (refer to Tables 3 and 4), which further suggests that human adaptation

Figure 11
Model Fits to Participant Data



Note. Models only varied in the degree of maintenance. Note that lower fit scores entail a closer fit to participant data. See the online article for the color version of this figure.

and complex skill acquisition is described by the learning mechanisms like those implemented in our model.

Decaying Past Experiences Facilitates Adaptation

Having established that the behavior of our human players is best described by a learning mechanism that applies a moderate decay to experiences, we now examine the implications decay has for adaptation and task performance by comparing the model with decay ($\theta = .992$) and the model without decay ($\theta = 1$). Note that both models in this comparison are identical in all other model dimensions (i.e., sharing the same optimal choices for source weight ratio, slowdown, and minimum thrust time) except for the degree of maintenance/decay.

Compared to a player without decay, a player with decay adapts faster to changes in the task environment. This is especially apparent in the LLLLH condition. Both models perform similarly during the extended low thrust period (games 1 to 32), but differ in their behavior following a drastic change from low to high thrust on game 33 (refer to Figures 12a, b, and c). While both models, like subjects, are adversely affected by the increase in acceleration on game 33, as indicated by the sharp decrease in points between games 32 and 33, the model with decay adapts faster to the change and scores higher than the model without decay immediately following the switch.

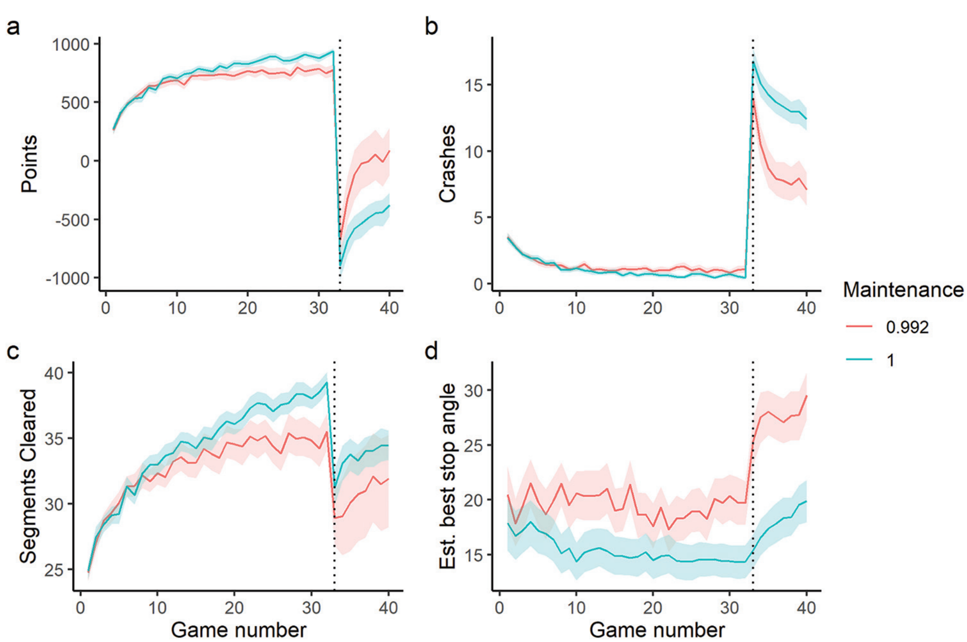
This increased rate of adaptation is reflected in the dynamics of the control parameters. For the purposes of illustrating these dynamics, we will focus on the Stop angle parameter (refer to Appendix E for details on all 5 control parameters across the 10 conditions). Recall that the stop angle is the angle at which the player decides to stop thrusting when making a hard turn around corners. As there is a lag between that decision and actually lifting the finger off the thrust key, the ship's trajectory will be off if the decision is issued only when the angle of the ship's trajectory matches the desired angle. As the acceleration increases, each additional tick of thrust changes the ship's trajectory more, and thus the stop angle needs to increase to achieve the same target trajectory. As Figure 12d shows, following game 32, the model with decay increased its estimate of the optimal stop angle at a faster rate than the model without decay did.

While discounting past experiences does facilitate faster adaptation to a new environment, decay also causes the player to be oversensitive to random fluctuations in the rate of return for various settings of control parameters (see Figure 9). Even when the acceleration remains unchanged for a long period, the rate of payoff for a particular control value (e.g., stop angle) continues to vary between observations; for example, noise inherent in the motor system affects how precisely each keypress is executed, which in turn affects the likelihood of the ship crashing in a given situation. Due to the stochasticity of the processes that underlie such variation, it is impossible to deterministically predict the effect of these sources of variation on the rate of payoff. Thus, an optimal controller should learn to average over these sources of variation to estimate the optimal control value for its current acceleration.

For the decay model, the bias toward learning from recent experiences that facilitated adapting to new environments is now a disadvantage. This disadvantage is most evident in the control conditions (MMMMM and HHHHH). Although both the decay

Figure 12

(a) Points per Game in the LLLLH Condition for the Model With Decay (Maintenance = 0.992, In Red) and the Model Without Decay (Maintenance = 1, in Blue); (b) Number of Crashes per Game; (c) Number Of Segments Cleared Per Game; (d) Estimated Optimal Stop Angle Per Game



Note. Shaded areas reflect bootstrapped 95% confidence limits. the dotted lines indicate a switch from low to high thrust. L = Low; H = High. See the online article for the color version of this figure.

and non–decay models rapidly decrease in crashes early on (games to 1 to 6), the two models diverge thereafter. For the HHHHH condition, the decay model asymptotes on points, crashes, and number of segments cleared around game 6, but the non–decay model continues improving until about game 20 (see Figures 13a, b, and c).

Similar to above, these differences are reflected in the dynamics of the stop angle controller, particularly in the HHHHH condition. As Figure 13d shows, the model without decay quickly settles on a larger optimal angle than the model with decay.

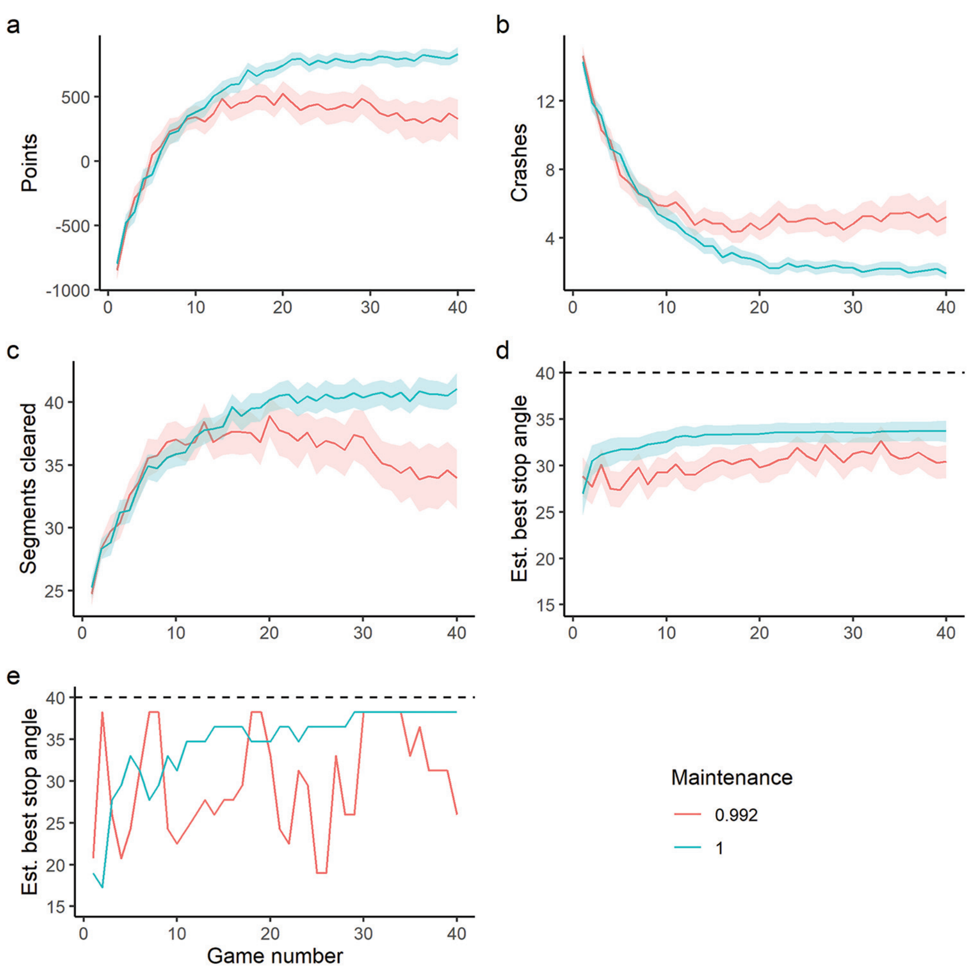
The dynamics of the model's control parameters also provide a possible account for the asymmetric costs of switching accelerations. Recall that for human players, switching to a new block from a lower acceleration impairs performance while switching to a new block from a higher acceleration has little effect. Model players also demonstrate this asymmetry between switch–faster and switch–slower effects ($F[1, 3997] = 208.6$, $p < .001$); refer to Table 4). In terms of control learning, a larger change in acceleration correlates with a larger amount of change required in the optimal stop angle, press point, and tap thrust duration (refer to Appendix E). For the same magnitude of change required, switching to a higher acceleration hurts performance because applying the control values learned in a lower acceleration results in over–executing key actions. For example, if a player currently in a high acceleration game used a long tap thrust duration that was optimal for previous low acceleration games, they would over–thrust and have to turn the ship around to slow it down. Switching to a lower acceleration is much less punishing, since it is easier to compensate for under–executing key actions; if a player executed too short of a thrust, they could simply tap the thrust key again to increase the speed of the ship.

General Discussion and Future Work

The current study investigated how human players adapted to parametric changes in the context of learning and mastering a

Figure 13

(a) Points per Game in the HHHHH Condition for the Model With Decay (Maintenance = 0.992, in Red) and the Model Without Decay (Maintenance = 1, in Blue); (b) Number of Crashes per Game; (c) Number of Segments Cleared per Game; (d) Estimated Optimal Stop Angle per Game; (e) Etimated Optimal Stop Angle per Game for Two Individual Models



Note. Shaded areas reflect bootstrapped 95% confidence limits. The dotted lines indicate a switch from low to high thrust. See the online article for the color version of this figure.

complex skill – navigating the spaceship in Space Track. Notably, most players were able to quickly adapt to changes in the ship's acceleration, and the amount of adaptation was affected by the acceleration on the game before the change.

Of the ACT–R models tested, the model that incorporated a constant time-based decay on past experiences best captured how human players come to master Space Track and adapt to these parameteric changes in acceleration. As different accelerations likely result in different payoffs for the same control setting, a player that discounts old experiences from a previous acceleration would update their estimated payoffs for the same control value faster when adapting to a new acceleration. This in turn allows the player to converge faster on the new optimal control value which facilitates adaptation to changes in the task environment.

More generally, adaptation to changes in the environment is facilitated by prioritizing information learned from recent experiences as recent experiences would better reflect the state and reward structure of the current environment. This principle is well-established in models of learning and adaptation across a wide range of domains: retention and forgetting (Anderson & Milson, 1989; Anderson & Schooler, 1991), category learning (Elliott & Anderson, 1995), sensorimotor learning (Berniker & Kording, 2011; Trewartha et al. 2014), and reinforcement learning (Niv et al., 2015).

However, decay may not always be adaptive. Due to the large degree of freedom in Space Track, two different instances of an action with the same control values could result in two different payoffs. The relationship between control values and payoffs is probabilistic rather than deterministic such that changing the acceleration entails shifting the means of probability distributions. In such a system, a player with a high rate of decay on experiences would fail to abstract away from the specifics of each instance, and consequently fail to settle on a stable estimate of the mean payoff for each control value.

The decay rate in our framework bears a close relationship with the the learning rate parameter in reinforcement learning models. While a high learning rate entails adapting faster to a change in the action-reward contingencies, it also entails being oversensitive to sources of irrelevant variability. An agent with a high learning rate would excessively increase the estimated value of a bad option following just one rewarding outcome or overdevalue a good option after one particularly punishing experience. In fact, some reinforcement learning models of human learning and choice behavior do report low to moderate values of the learning rate parameter (Collins et al., 2014; Pedersen et al., 2017). These findings parallel the results of our model simulations, where human learning and adaptation is best described with a moderate decay rate.

Instead of maintaining the same learning rate across the entire experiments, participants might modulate their learning rates in response to changes in the environment. Within the reinforcement learning framework, human learners have been found to increase their learning rate when the task environment is volatile (Behrens et al., 2007; Cook et al., 2019). When the agent is exposed to unchanging environments, the learning rate might be expected to decrease, allowing the agent converge on a stable, optimal parameter setting. Conversely, when the agent detects a change, or is primed to expect a change, the learning rate would increase, allowing the agent to become sensitive to the change and adapt quickly to it. Research from reversal learning further suggests that learners incorporate prior expectations about when such changes might occur with their recent experiences to increase learning rates even before a significant change occurs (Costa et al., 2015; Izquierdo et al. 2017).

Although the ability to vary learning (and decay) rates appears to be both advantageous and already present in human learners, it is not immediately apparent what the role of such an ability should be in complex tasks like Space Track or driving. Even when practicing only with one level of acceleration (or type of car), these experiences are definitely less "stable" as compared to those typically studied in the reversal and reinforcement learning literatures. A potential line of research is to investigate the role of variable learning rates in complex tasks by extending the single–decay model proposed in this article.

Another potential line of research concerns the phenomenon of savings. Savings is a motor learning phenomena whereby people display faster readaptation to a previously experienced situation as compared to a completely new environment (Huang et al., 2011; Krakauer et al. 2005; Smith et al. 2006). Our current model does not directly account for such a phenomenon; the weight of an experience will eventually decay to a value close to zero after a few minutes of gameplay regardless of whether a switch in acceleration occurs. If the learner could instead create a separate controller for each context (i.e., acceleration) and pause the decay on experiences in one context when switching to another context, then when the learner switches back to the old context, it can reactivate the old controller without having to relearn the optimal control settings.

While the study primarily investigates if and how a decay-based mechanism facilitates adapting to changes during complex skill acquisition, our model comparisons also reveals that players tend to weight negative events more severely than positive ones. As players are rewarded depending on how many points they earn, it is reasonable that some players would weight avoiding crashes over clearing track segments in a ratio that reflects their relative contribution to points. Alternatively, as humans have been shown to demonstrate loss aversion in the face of equally valued gambles (Kahneman & Tversky, 1979), it is also reasonable that some players would place an even greater emphasis on avoiding crashes. Future work could explore whether subjects' weighting of crashes and rectangles cleared would respond to changes in how the two are weighted.

The study also identifies a potential role for control tuning in explaining performance costs that arise from shifting between different task dynamics. The nature of these costs differ from those often discussed in the task–switching literature, where switches occur between tasks with different goals (Allport et al., 1994; Wylie & Allport, 2000), or tasks with different complexity of procedures (Luwel et al. 2009; Schneider & Anderson, 2010). Rather, these switches are more similar to those discussed in the sensorimotor adaptation literature, where a gradually adapting motor system tunes its output according to the error it experiences (Berniker & Kording, 2011; He et al. 2016). These costs in Space Track arise from the discrepancy between actions that were tuned to a previous task dynamic and what is optimal for the current task dynamic. A larger change in task dynamics incurs greater performance costs because previously learned actions are more suboptimal in the new environment. This penalty is amplified when switching to faster and more sensitive task dynamics as players have less reaction time (RT) and need more corrective actions to compensate for over–executions as compared to under–executions of key actions.

References

Allport, D. A., Styles, E. A., & Hsieh, S. (1994). Shifting intentional set: Exploring the dynamic control of tasks. In C. Umilta, & M. Moscovitch (Eds.), *Attention and performance XV* (pp. 421–452). MIT Press.

Anderson, J. R., Betts, S., Bothell, D., Hope, R., & Lebiere, C. (2019). Learning rapid and precise skills. *Psychological Review*, *126*(5), 727–760. https://doi.org/10.1037/rev0000152

Anderson, J. R., Bothell, D., Byrne, M. D., Douglass, S., Lebiere, C., & Qin, Y. (2004). An integrated theory of the mind. *Psychological Review*, *111*(4), 1036–1060. https://doi.org/10.1037/0033-295X.111.4.1036

Anderson, J. R., & Lebiere, C. J. (2014). *The atomic components of thought*. Psychology Press.

Anderson, J. R., & Milson, R. (1989). Human memory: An adaptive perspective. *Psychological Review*, *96*(4), 703–719. https://doi.org/10.1037/0033-295X.96.4.703

Anderson, J. R., & Schooler, L. J. (1991). Reflections of the environment in memory. *Psychological Science*, *2*(6), 396–408. https://doi.org/10.1111/j.1467-9280.1991.tb00174.x

Averell, L., & Heathcote, A. (2011). The form of the forgetting curve and the fate of memories. *Journal of Mathematical Psychology*, *55*(1), 25–35. https://doi.org/10.1016/j.jmp.2010.08.009

Behrens, T. E., Woolrich, M. W., Walton, M. E., & Rushworth, M. F. (2007). Learning the value of information in an uncertain world. *Nature Neuroscience*, *10*(9), 1214–1221. https://doi.org/10.1038/nn1954

Berniker, M., & Kording, K. P. (2011). Estimating the relevance of world disturbances to explain savings, interference and long-term motor adaptation effects. *PLOS Computational Biology*, *7*(10), Article e1002210. https://doi.org/10.1371/journal.pcbi.1002210

Brady, F. (2008). The contextual interference effect and sport skills. *Perceptual and Motor Skills*, *106*(2), 461–472. https://doi.org/10.2466/pms.106.2.461-472

Braun, D. A., Aertsen, A., Wolpert, D. M., & Mehring, C. (2009). Motor task variation induces structural learning. *Current Biology*, *19*(4), 352–357. https://doi.org/10.1016/j.cub.2009.01.036

Cardis, M., Casadio, M., & Ranganathan, R. (2018). High variability impairs motor learning regardless of whether it affects task performance. *Journal of Neurophysiology*, *119*(1), 39–48. https://doi.org/10.1152/jn.00158.2017

Catalano, J. F., & Kleiner, B. M. (1984). Distant transfer in coincident timing as a function of variability of practice. *Perceptual and Motor Skills*, *58*(3), 851–856. https://doi.org/10.2466/pms.1984.58.3.851

Christakou, A., Murphy, C. M., Chantiluke, K., Cubillo, A. I., Smith, A. B., Giampietro, V., Daly, E., Ecker, C., Robertson, D., Murphy, D. G., & Rubia, K., & MRC AIMS Consortium. (2013). Disorder-specific functional abnormalities during sustained attention in youth with attention deficit hyperactivity disorder (adhd) and with autism. *Molecular Psychiatry*, *18*(2), 236–244. https://doi.org/10.1038/mp.2011.185

Codol, O., Holland, P. J., & Galea, J. M. (2018). The relationship between reinforcement and explicit control during visuomotor adaptation. *Scientific Reports*, *8*(1), 1–11. https://doi.org/10.1038/s41598-018-27378-1

Collins, A. G., Brown, J. K., Gold, J. M., Waltz, J. A., & Frank, M. J. (2014). Working memory contributions to reinforcement learning impairments in schizophrenia. *Journal of Neuroscience*, *34*(41), 13747–13756. https://doi.org/10.1523/JNEUROSCI.0989-14.2014

Cook, J. L., Swart, J. C., Froböse, M. I., Diaconescu, A. O., Geurts, D. E., Den Ouden, H. E., & Cools, R. (2019). Catecholaminergic modulation of meta-learning. *Elife*, *8*, Article e51439. https://doi.org/10.7554/eLife.51439

Costa, V. D., Tran, V. L., Turchi, J., & Averbeck, B. B. (2015). Reversal learning and dopamine: A bayesian perspective. *Journal of Neuroscience*, *35*(6), 2407–2416. https://doi.org/10.1523/JNEUROSCI.1989-14.2015

Decker, J. H., Otto, A. R., Daw, N. D., & Hartley, C. A. (2016). From creatures of habit to goal-directed learners: Tracking the developmental emergence of model-based reinforcement learning. *Psychological Science*, *27*(6), 848–858. https://doi.org/10.1177/0956797616639301

DeLosh, E. L., Busemeyer, J. R., & McDaniel, M. A. (1997). Extrapolation: The sine qua non for abstraction in function learning. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, *23*(4), 968–986. https://doi.org/10.1037//0278-7393.23.4.968

Ebbinghaus, H. (1885/1913). *Memory: A contribution to experimental psychology* (Translated by H. A. Ruger & C. E. Bussenues). Teachers College, Columbia University.

Elliott, S. W., & Anderson, J. R. (1995). Effect of memory decay on predictions from changing categories. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, *21*(4), 815–836. https://doi.org/10.1037//0278-7393.21.4.815

Goodwin, J. E., Eckerson, J. M., Grimes, C. R., & Gordon, P. M. (1998). Effect of different quantities of variable practice on acquisition, retention, and transfer of an applied motor skill. *Perceptual and Motor Skills*, *87*(1), 147–151. https://doi.org/10.2466/pms.1998.87.1.147

He, K., Liang, Y., Abdollahi, F., Fisher Bittmann, M., Kording, K., & Wei, K. (2016). The statistical determinants of the speed of motor learning. *PLoS Computational Biology*, *12*(9), Article e1005023. https://doi.org/10.1371/journal.pcbi.1005023

Heathcote, A., Brown, S., & Mewhort, D. J. (2000). The power law repealed: The case for an exponential law of practice. *Psychonomic Bulletin & Review*, *7*(2), 185–207. https://doi.org/10.3758/bf03212979

Huang, V. S., Haith, A., Mazzoni, P., & Krakauer, J. W. (2011). Rethinking motor learning and savings in adaptation paradigms: model-free memory for successful actions combines with internal models. *Neuron*, *70*(4), 787–801. https://doi.org/10.1016/j.neuron.2011.04.012

Ingram, J. N., Flanagan, J. R., & Wolpert, D. M. (2013). Context-dependent decay of motor memories during skill acquisition. *Current Biology*, *23*(12), 1107–1112. https://doi.org/10.1016/j.cub.2013.04.079

Izawa, J., & Shadmehr, R. (2011). Learning from sensory and reward prediction errors during motor adaptation. *PLOS Computational Biology*, *7*(3), Article e1002012. https://doi.org/10.1371/journal.pcbi.1002012

Izquierdo, A., Brigman, J. L., Radke, A. K., Rudebeck, P. H., & Holmes, A. (2017). The neural basis of reversal learning: an updated perspective. *Neuroscience*, *345*, 12–26. https://doi.org/10.1016/j.neuroscience.2016.03.021

Kahneman, D., & Tversky, A. (1979). Prospect theory: An analysis of decisions under risk. *Econometrica*, *47*(2), 263–292. https://doi.org/10.2307/1914185

Kalish, M. L., Lewandowsky, S., & Kruschke, J. K. (2004). Population of linear experts: Knowledge partitioning and function learning. *Psychological Review*, *111*(4), 1072–1099. https://doi.org/10.1037/0033-295X.111.4.1072

Kerr, R., & Booth, B. (1978). Specific and varied practice of motor skill. pdf. *Perceptual and Motor Skills*, *46*(2), 395–401. https://doi.org/10.1177/003151257804600201

Kieras, D. E., & Meyer, D. E. (1997). An overview of the epic architecture for cognition and performance with application to human-computer interaction. *Human–Computer Interaction*, *12*(4), 391–438. https://doi.org/10.1207/s15327051hci1204_4

Krakauer, J. W., Ghez, C., & Ghilardi, M. F. (2005). Adaptation to visuomotor transformations: consolidation, interference, and forgetting. *Journal of Neuroscience*, *25*(2), 473–478. https://doi.org/10.1523/JNEUROSCI.4218-04.2005

Loftus, E. F., & Loftus, G. R. (1980). On the permanence of stored information in the human brain. *American Psychologist*, *35*(5), 409–420. https://doi.org/10.1037/0003-066X.35.5.409

Lucas, C. G., Griffiths, T. L., Williams, J. J., & Kalish, M. L. (2015). A rational model of function learning. *Psychonomic Bulletin & Review*, *22*(5), 1193–1215. https://doi.org/10.3758/s13423-015-0808-5

Luwel, K., Schillemans, V., Onghena, P., & Verschaffel, L. (2009). Does switching between strategies within the same task involve a cost? *British Journal of Psychology*, *100*(4), 753–771. https://doi.org/10.1348/000712609X402801

McCracken, H. D., & Stelmach, G. E. (1977). A test of the schema theory of discrete motor learning. *Journal of Motor Behavior*, *9*(3), 193–201. https://doi.org/10.1080/00222895.1977.10735109

McGovern, D. P., Webb, B. S., & Peirce, J. W. (2012). Transfer of perceptual learning between different visual tasks. *Journal of Vision*, *12*(11), 4. https://doi.org/10.1167/12.11.4

Narain, D., Smeets, J. B., Mamassian, P., Brenner, E., & van Beers, R. J. (2014). Structure learning and the occam's razor principle: A new view of human function acquisition. *Frontiers in Computational Neuroscience*, *8*, 121. https://doi.org/10.3389/fncom.2014.00121

Newell, K. M. (2003). Schema theory (1975). Retrospectives and prospectives. *Research Quarterly for Exercise and Sport*, *74*(4), 383–388. https://doi.org/10.1080/02701367.2003.10609108

Newell, K., & Shapiro, D. C. (1976). Variability of practice and transfer of training: Some evidence toward a schema view of motor learning. *Journal of Motor Behavior*, *8*(3), 233–243. https://doi.org/10.1080/00222895.1976.10735077

Niv, Y., Daniel, R., Geana, A., Gershman, S. J., Leong, Y. C., Radulescu, A., & Wilson, R. C. (2015). Reinforcement learning in multidimensional environments relies on attention mechanisms. *Journal of Neuroscience*, *35*(21), 8145–8157. https://doi.org/10.1523/JNEUROSCI.2978-14.2015

Pedersen, M. L., Frank, M. J., & Biele, G. (2017). The drift diffusion model as the choice rule in reinforcement learning. *Psychonomic Bulletin & Review*, *24*(4), 1234–1251. https://doi.org/10.3758/s13423-016-1199-y

Rubin, D. C., Hinton, S., & Wenzel, A. (1999). The precise time course of retention. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, *25*(5), 1161–1176. https://doi.org/10.1037/0278-7393.25.5.1161.

Schmidt, R. A. (1975). A schema theory of discrete motor skill learning. *Psychological Review*, *82*(4), 225–260. https://doi.org/10.1037/h0076770

Schneider, D. W., & Anderson, J. R. (2010). Asymmetric switch costs as sequential difficulty effects. *Quarterly Journal of Experimental Psychology*, *63*(10), 1873–1894. https://doi.org/10.1080/17470211003624010

Smith, M. A., Ghazizadeh, A., & Shadmehr, R. (2006). Interacting adaptive processes with different timescales underlie short-term motor learning. *PLOS Biology*, *4*(6), Article e179. https://doi.org/10.1371/journal.pbio.0040179

Speelman, C. P., & Kirsner, K. (2001). Predicting transfer from training performance. *Acta Psychologica*, *108*(3), 247–281. https://doi.org/10.1016/s0001-6918(01)00039-7

Taatgen, N. A. (2013). The nature and transfer of cognitive skills. *Psychological Review*, *120*(3), 439–471. https://doi.org/10.1037/a0033138

Taatgen, N. A., & Anderson, J. R. (2002). Why do children learn to say "broke"? a model of learning the past tense without feedback. *Cognition*, *86*(2), 123–155. https://doi.org/10.1016/S0010-0277(02)00176-2

Thorndike, E. L. (1906). *The principles of teaching: Based on psychology*. AG Seiler.

Trewartha, K. M., Garcia, A., Wolpert, D. M., & Flanagan, J. R. (2014). Fast but fleeting: adaptive motor learning processes associated with aging and cognitive decline. *Journal of Neuroscience*, *34*(40), 13411–13421. https://doi.org/10.1523/JNEUROSCI.1489-14.2014

Tuitert, I., Bootsma, R. J., Schoemaker, M. M., Otten, E., Mouton, L. J., & Bongers, R. M. (2017). Does practicing a wide range of joint angle configurations lead to higher flexibility in a manual obstacle-avoidance target-pointing task? *PLoS ONE*, *12*(7), Article e0181041. https://doi.org/10.1371/journal.pone.0181041

Turnham, E. J. A., Braun, D. A., & Wolpert, D. M. (2012). Facilitation of learning induced by both random and gradual visuomotor task variation. *Journal of Neurophysiology*, *107*(4), 1111–1122. https://doi.org/10.1152/jn.00635.2011

Van Rossum, J. H. (1990). Schmidt's schema theory: The empirical base of the variability of practice hypothesis: A critical analysis. *Human Movement Science*, *9*(3-5), 387–435. https://doi.org/10.1016/0167-9457(90)90010-B

Willey, C. R., & Liu, Z. (2018). Limited generalization with varied, as compared to specific, practice in short-term motor learning. *Acta Psychologica*, *182*, 39–45. https://doi.org/10.1016/j.actpsy.2017.11.008

Wylie, G., & Allport, A. (2000). Task switching and the measurement of "switch costs. *Psychological Research*, *63*(3–4), 212–233. https://doi.org/10.1007/s004269900003

Appendix A

Space Track

Here is a pointer to the version of Space Track that was used in the two experiments: http://andersonlab.net/demos/spacetrack-v4/.

The in–game track is composed of rectangles. The skeleton of the track comprises straight line segments of 363 pixels in length connected from end to end. Rectangle segments are drawn by extending the line segments such that each line segment lies in the middle of a rectangle. Rectangles are 473 pixels long and 110 pixels wide.

There are always two rectangles on the screen: the "current" rectangle that the ship occupies, and the "next" rectangle along the track. No points are earned when the ship enters the overlapping area between the two rectangles. 25 points are awarded when the ship has exited the "current" rectangle and is in the "next" rectangle, which then becomes the new "current" rectangle. If the game detects that the ship is not in either rectangle, the ship is determined to have crashed and the player loses 100 points.

When the ship exits the "current" rectangle and is in the "next" rectangle, the game creates a new "next" rectangle. A new line segment is generated that forms an angle between 30° to 150° or -30° to -150° with the current line segment. The new rectangle is then drawn around the new line segment.

In addition to the track layout, our implementation of Space Track also shares the following key features with the version described in Anderson et al. (2019):

1. The game runs at 30 frames per second, and each game tick is 1/30th of a second (or ~33ms)

(*Appendices continue*)

2. The task screen's dimensions are 710 x 626 pixels.

3. At the start of each game, the ship moves along the the first rectangle in the direction of the *x*-axis at a speed of 1 pixel per tick.

4. Each game tick a turn key is held for rotates the ship by 6°.

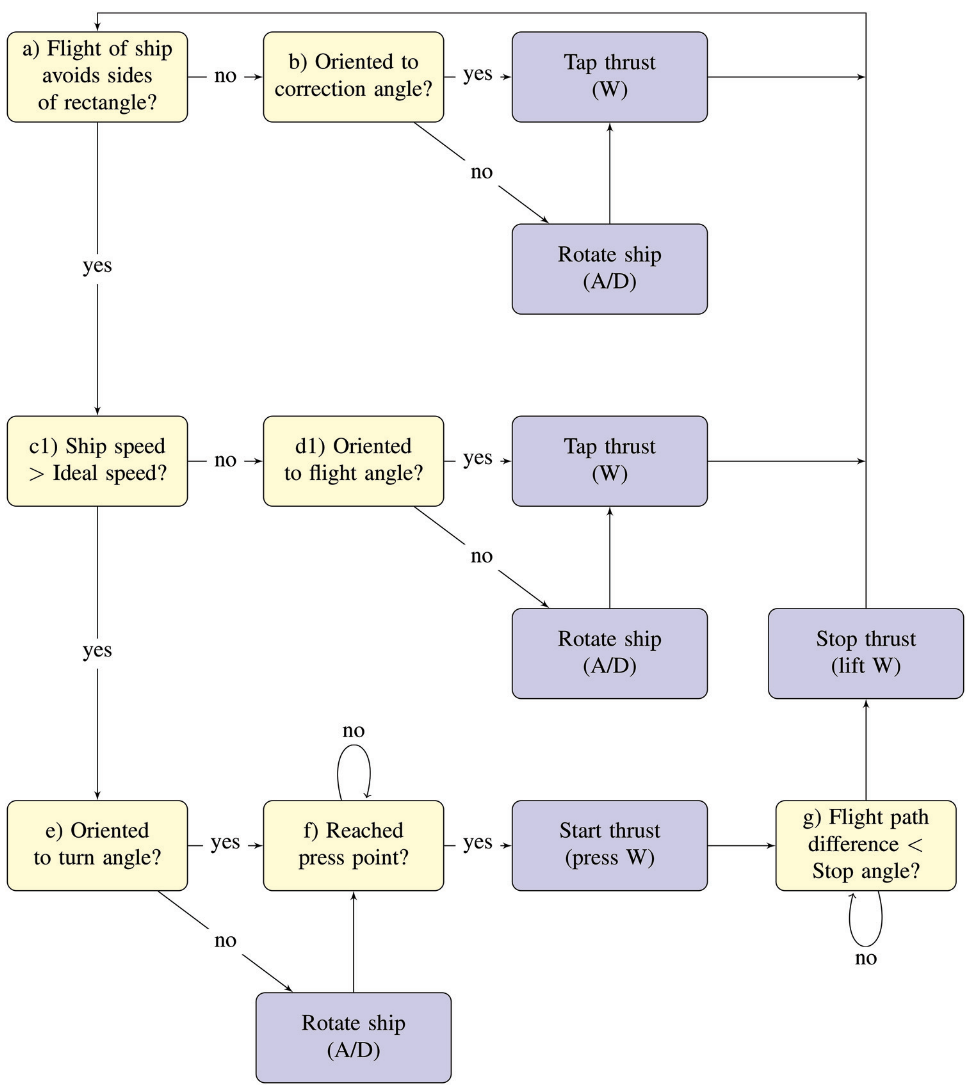
The key difference between our implementation and the one in Anderson et al. (2019) pertains to the acceleration. In the original implementation, each game tick the thrust key was held for increased the ship's velocity vector by .3 pixels per tick. In the current study, the accelerations were set to .2, .4, or .6, depending on the acceleration of the block (low, medium, or high, respectively).

Appendix B

Model Operators

Figure B1

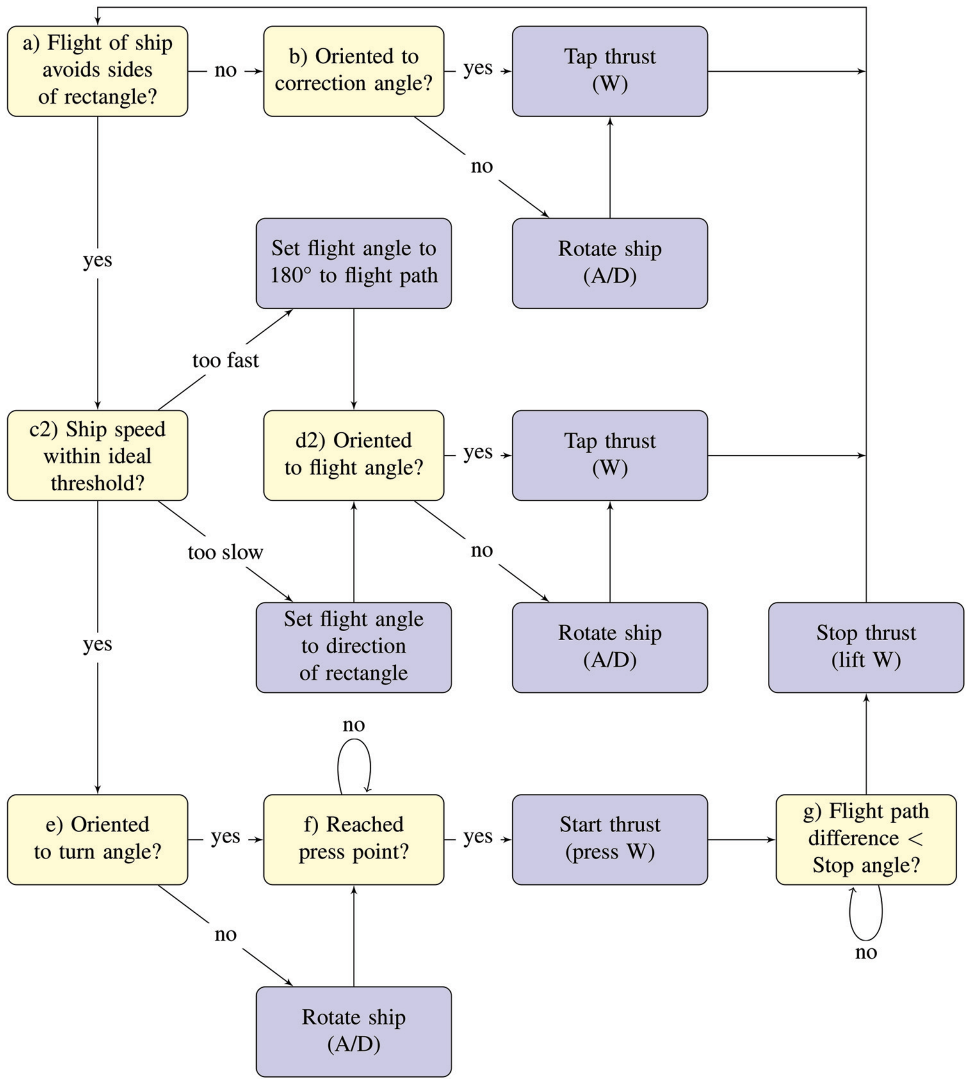
Flowchart of Operators for a Model Without the Ability to Slow Down



Note. Arrows represent transitions between states. Yellow operators test conditions to determine the next state. Blue operators perform actions. See the online article for the color version of this figure.

(*Appendices continue*)

Figure B2
Flowchart of Operators for a Model With the Ability to Slow Down



Note. Arrows represent transitions between states. Yellow operators test conditions to determine the next state. Blue operators perform actions. See the online article for the color version of this figure.

(*Appendices continue*)

Table B1
Branching Operators in Figures B1 and B2

(a)	**Flight of ship avoids sides of rectangle?** While flying down a rectangle, players need to monitor whether the ship's current flight path will result in crashing into one of the rectangle's lengths. This operator checks for this condition and branches to making a correction to the path if necessary.
(b)	**Oriented to correction angle?** If a correction is necessary, the player computes a target angle at which a thrust should be issued to change the trajectory of the ship. This operator checks if the difference between the ship's orientation and the target correction angle exceeds the aim threshold. If so, the player corrects the ship's orientation before issuing a tap thrust.
(c1)	**Ship speed > Ideal speed?** For models that cannot slow down, this operator checks if the ship speed exceeds some speed threshold. If it is, the operator branches to preparing for a hard turn at the corner between two rectangles. If the ship is too slow, the operator branches to speeding the ship up.
(d1)	**Oriented to flight angle?** If the ship is too slow, the player will prepare to increase the ship speed. Before issuing a thrust, the desired flight angle is set to the direction of the rectangle. This operator checks if the difference between the ship's orientation and the target flight angle exceeds the aim threshold. If so, the ship's orientation is corrected before a tap thrust is issued.
(c2)	**Ship speed within ideal threshold?** For models that can slow down, this operator checks if the ship is flying within some speed threshold. If it is, the operator branches to preparing for a hard turn at the upcoming corner. Otherwise, the operator branches to making adjustments to the ship's speed.
(d2)	**Oriented to flight angle?** If the ship is too slow, the flight angle is set to the direction of the rectangle. If the ship is too fast, the flight angle is set to 180° with respect to the current flight path to prepare for a slow down thrust. This operator checks if the difference between the ship's orientation and the target flight angle exceeds the aim threshold. If so, the ship's orientation is corrected before a tap thrust is issued.
(e)	**Oriented to turn angle?** To make a hard turn at the corner, the ship needs to be oriented to the angle bisector of the two rectangles. This operator checks if the difference between the ship's orientation and the target angle exceeds the aim threshold. If so, the player will correct the ship's orientation.
(f)	**Reached press point?** This operator checks if the ship has traveled far enough into the overlapping area between the two rectangles. If so, the player will begin to hold the W key for a sustained thrust.
(g)	**Flight path difference < Stop angle?** During a hard thrust, holding the thrust key changes the ship's flight path towards the direction of the new rectangle. As there is some lag between the decision to lift a finger and the actual release of the key, the decision to release the thrust key needs to occur before the ship's flight path is aligned with the direction of the new rectangle. This operator checks if the angular difference between the ship's flight path and the direction of the new rectangle is less than a stop angle threshold. If so, the player issues the command to lift the finger off the thrust key.

Note. Bold text referers to operators that correspond to the flowchart Figures B1 and B2.

Appendix C

Quadratic Regression

For a finite range of control values, we assume that there exists an optimal value and that the rate of payoff decreases with increasing distance from that optimal value. The quadratic function is a simple function that satisfies these two assumptions, and thus we chose to model the process of learning a relationship between control values (x) and their rates of payoffs (y) using quadratic regression.

In quadratic regression, the goal is to estimate parameters a, b, and c that define the best-fitting quadratic function, and to use that function to determine the control value with the highest estimated rate of payoff:

$$\hat{y} = ax^2 + bx + c \tag{C.1}$$

The parameters are estimated using the following matrix equation:

$$\begin{bmatrix} a \\ b \\ c \end{bmatrix} = \begin{bmatrix} \sum x_i^4 & \sum x_i^3 & \sum x_i^2 \\ \sum x_i^3 & \sum x_i^2 & \sum x_i \\ \sum x_i^2 & \sum x_i & n \end{bmatrix}^{-1} \begin{bmatrix} \sum x_i^2 y_i \\ \sum x_i y_i \\ \sum y_i \end{bmatrix} \tag{C.2}$$

where $[x_i, y_i]$ represent a control value–rate of payoff dyad during the ith experience. Recall that the model samples a particular control value for a period of n ticks, and divides the total payoff observed by n to arrive at the average rate of payoff (y_i):

$$y_i = \frac{payoff_{total}}{n} \tag{C.3}$$

For each sample, information about the sampling period, the sampled control value, and the average rate of payoff are stored as a set of summary statistics:

$$stats = \left\{ n, \sum x_i, \sum x_i^2, \sum x_i^3, \sum x_i^4, \sum y_i, \sum x_i y_i, \sum x_i^2 y_i \right\} \tag{C.4}$$

which are then used in equation C.2. Such a method is computationally simple, as it only requires storing eight terms per sample regardless of the length of the period. Furthermore, only one set of statistics need to be stored at any given point in time; the updated statistics are computed by adding the current sample's statistics to that of the previous set of statistics:

$$stats_{updated} = m * stats_{sample} + stats_{previous} \tag{C.5}$$

where m is the weighting factor from Equation 4. When $m = 1$, all experiences are weighted equally. When $m < 1$, past experiences are decayed as a function of time.

(*Appendices continue*)

Appendix D

How ACT–R Uses Control Values

This section briefly describes how the Controller module interacts with the central production system. For clarity, the productions described here are simplified versions of those actually used in the ACT–R model.

At the start of the experiment, Production 1 makes a request to the Controller module to set up 5 controllers in the Controller buffer, one for each control parameter.

Production: 1 Set up controllers

if The goal is to start playing **then**
  for each control parameter **do**
    Request Controller module to create controllers where
    CONTROL-SLOT = control parameter
    GOOD-SLOT = type of good events (i.e., segments cleared)
    BAD-SLOT = type of bad events (i.e., crashes)
    MIN = minimum control value
    MAX = maximum control value
    BAD-WEIGHT = relative weight of bad versus good outcomes
  end for
end if

The good and bad events are recorded in the Imaginal buffer. The count of these events are passed to the controllers at the end of each update interval independently of the central production system. The control values determined by the controllers are then passed to the Goal buffer. When the value of a state variable (in the Game State buffer) fulfills certain conditions with the corresponding control value in the Goal buffer, a production will fire to move the model into the next state according to Appendix B.

The example production below fires when the ship's current speed is below the threshold set by the ship speed controller. Upon firing, it prepares the ship for a speed up thrust by setting a target flight angle that is parallel to direction of the current track segment and changing the goal state to turn the ship to face that target angle.

Production 2: Ship is too slow, align ship with segment direction

if The goal is to do a retrieved step
and The retrieved action is to check the ship's speed
and The ship's speed is less than the controller's ship speed
then
  Set the target direction as the direction of the current track segment
  Change goal to turn to direction
end if

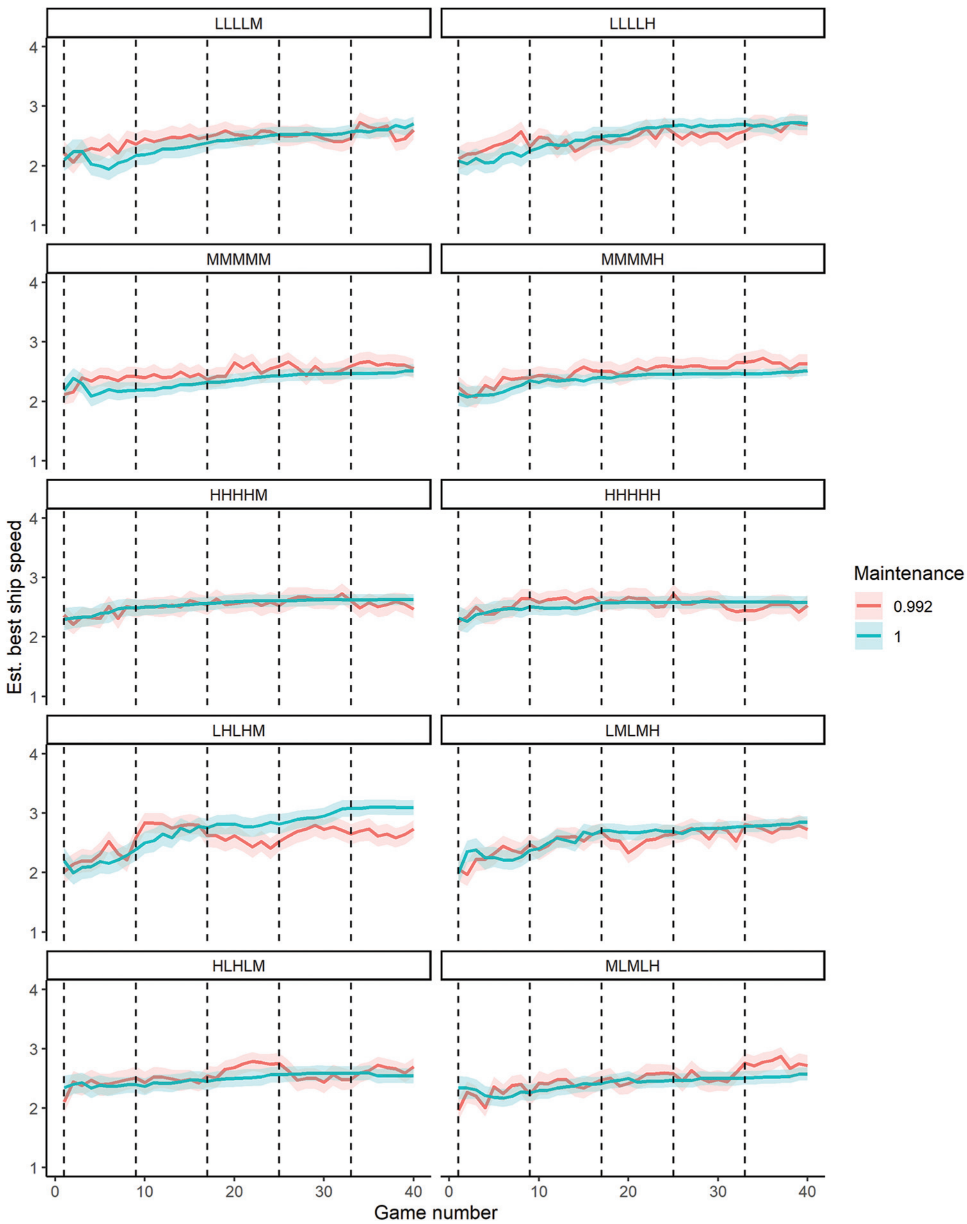
(*Appendices continue*)

Appendix E

Control Parameters

Figure E1

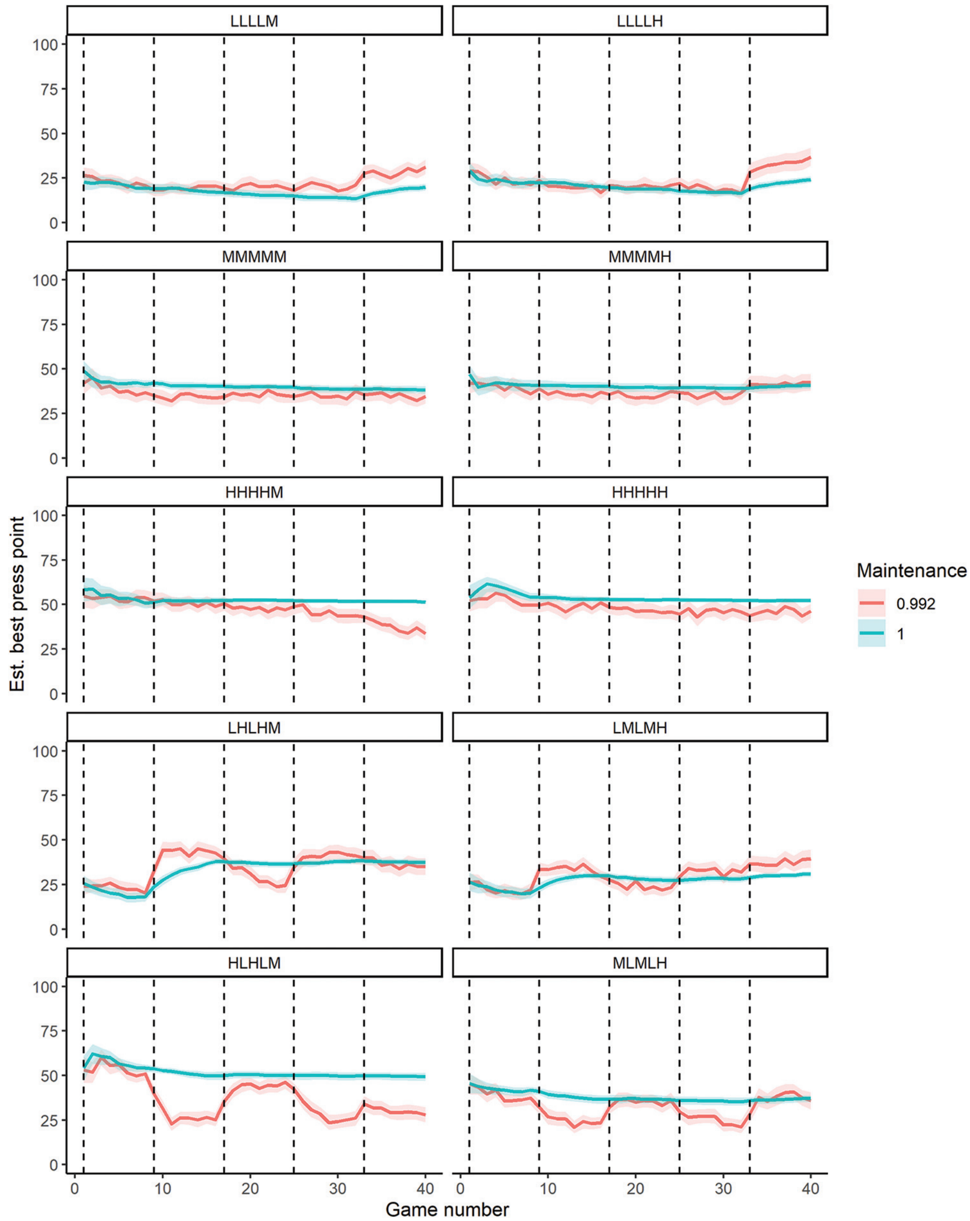
Best Values of Ship Speed Learned by the Model With (in Blue) and Without Decay (in Red) on Past Experiences



Note. Conditions from Experiment 1 are presented on the left and those in Experiment 2 are presented on the right. M = Medium; L = Low; H = High. See the online article for the color version of this figure.

(*Appendices continue*)

Figure E2
Best Values of Press Point Learned by Models

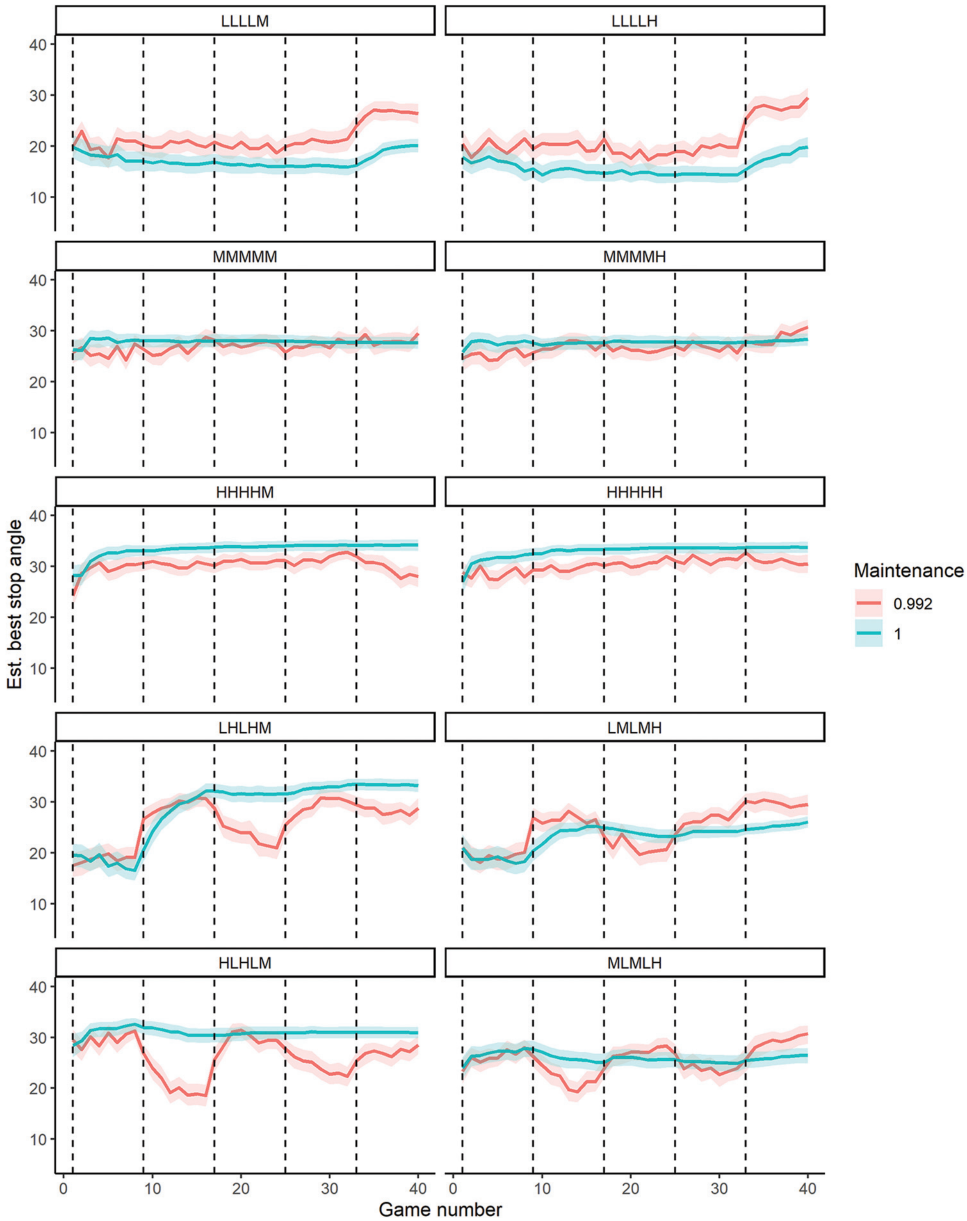


Note. M = Medium; L = Low; H = High. See the online article for the color version of this figure.

(*Appendices continue*)

Figure E3

Best Values of Stop Angle Learned by Models

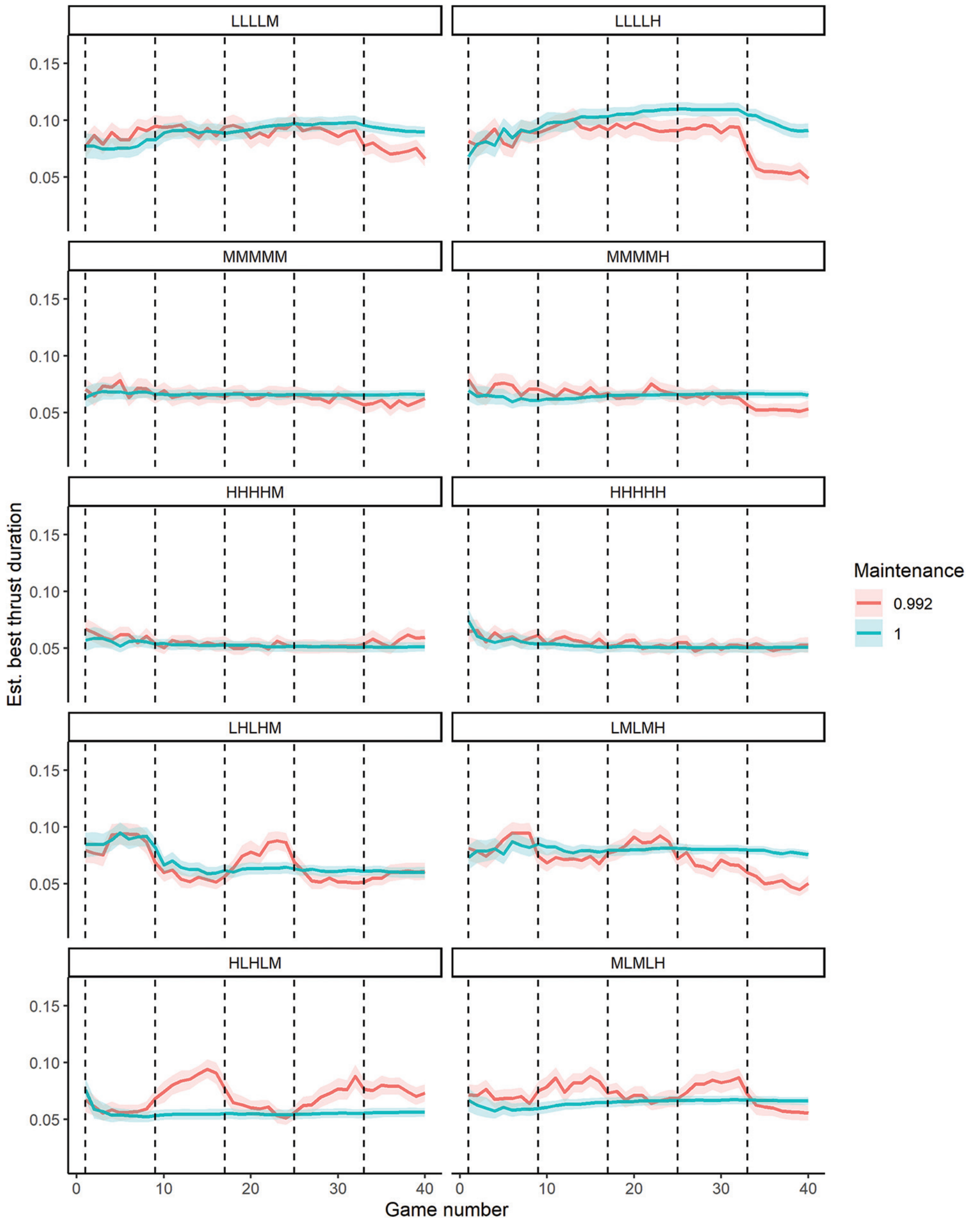


Note. M = Medium; L = Low; H = High. See the online article for the color version of this figure.

(*Appendices continue*)

Figure E4

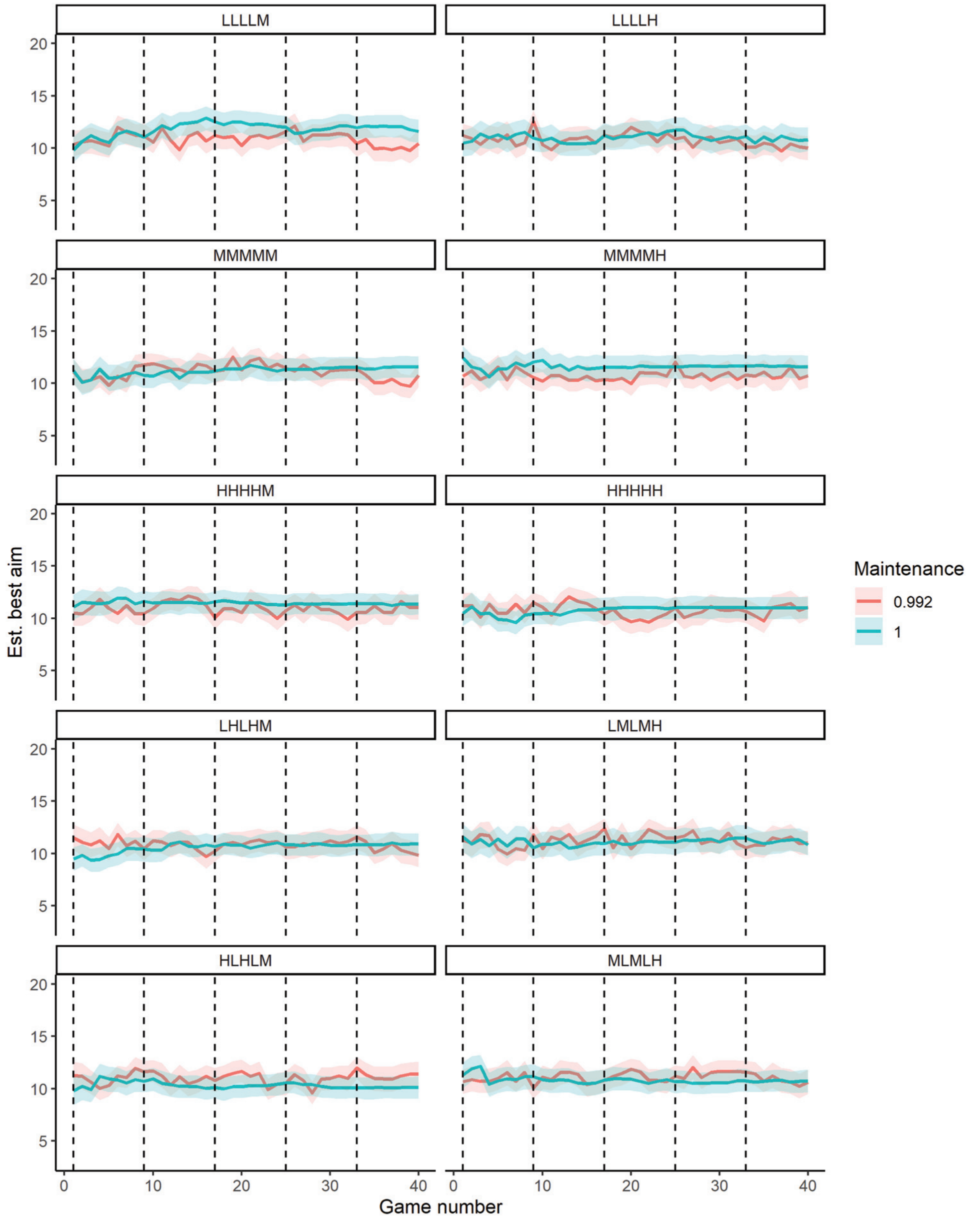
Best Values of Tap Thrust Duration Learned by Models



Note. M = Medium; L = Low; H = High. See the online article for the color version of this figure.

(*Appendices continue*)

Figure E5
Best Values of Aim Learned by Models



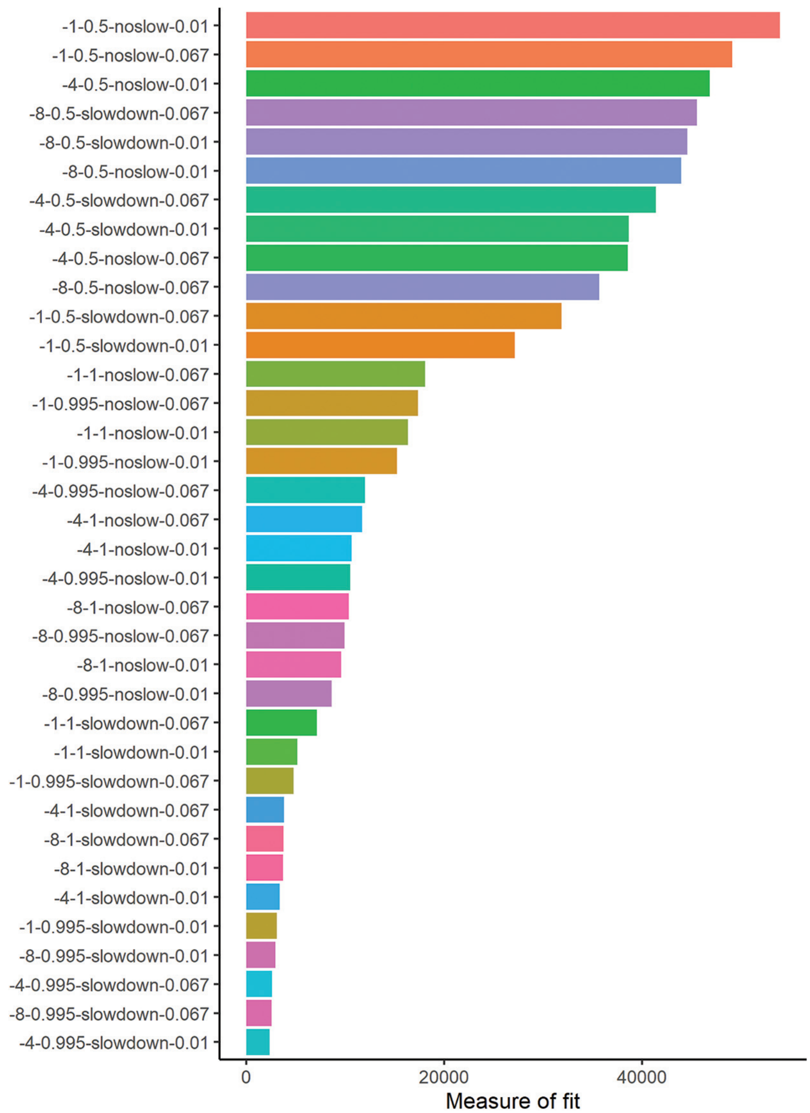
Note. M = Medium; L = Low; H = High. See the online article for the color version of this figure.

(*Appendices continue*)

Appendix F

Comparison of Model Fits

Figure F1
Model Fits to Participant Data



Note. This figure expands upon Figure 10 and presents fits for all 36 combinations of values across source weight ratio (3 values), maintenance (3 values), slowdown (2 levels), and tap thrust duration (2 levels). See the online article for the color version of this figure.

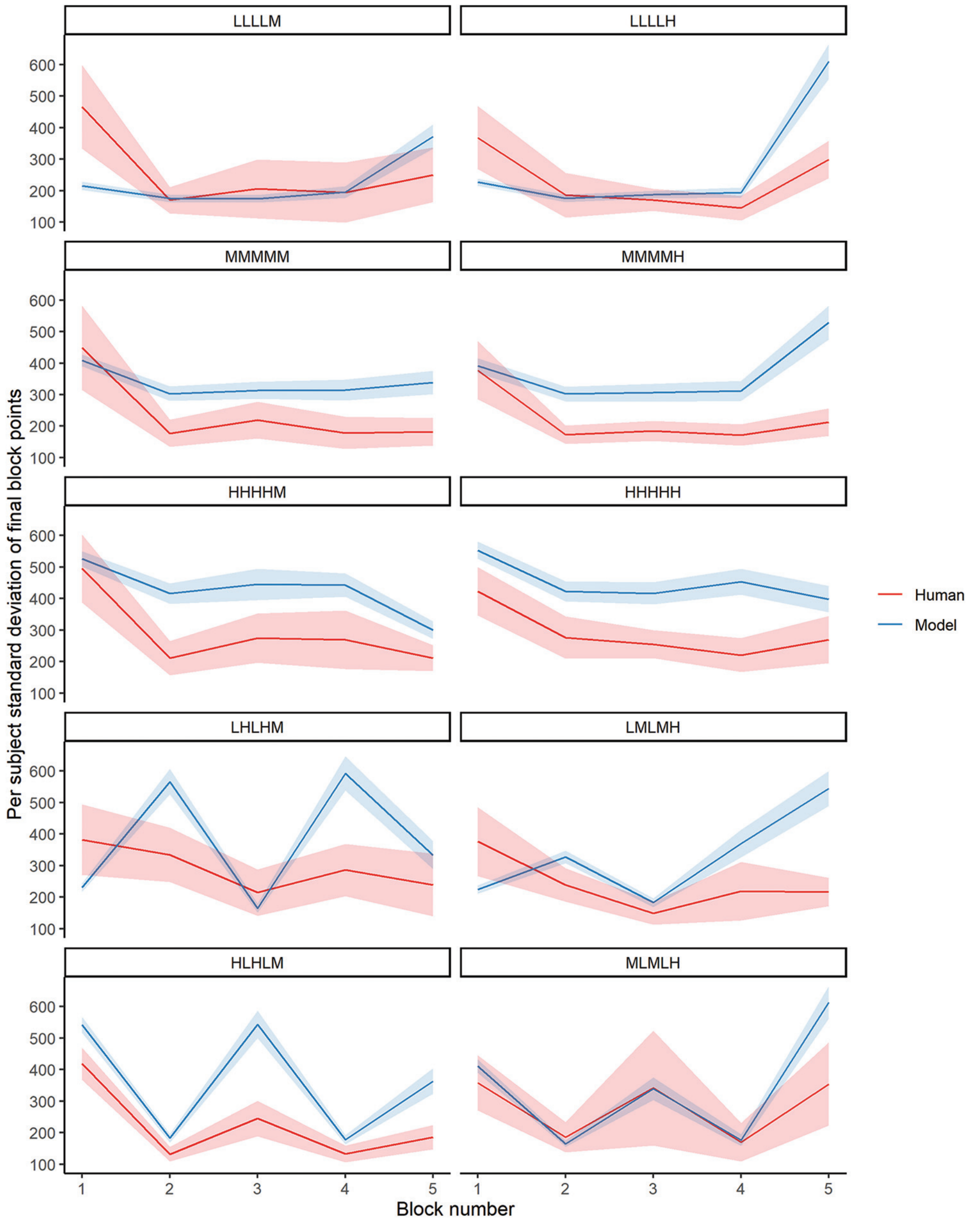
Appendix G

Comparisons of Within and Across participant Variability

While the model displays a close fit to the group level performance of human players, it is interesting to examine if the model is also able to capture the range of variability in performance observed in the human players. Note that the model simulations presented here were all generated with the same set of model parameters. This differs from the modeling approach of Anderson et al. where model performance was an average of simulations using different sets of model parameters. For instance, their models spanned five different rates of production learning, which enabled a wide range of performance. In

(*Appendices continue*)

Figure G1
Plot of Standard Deviations of Points Within Each Block of 8 Games



Note. Shaded regions indicate 95% confidence intervals. M = Medium; L = Low; H = High. See the online article for the color version of this figure.

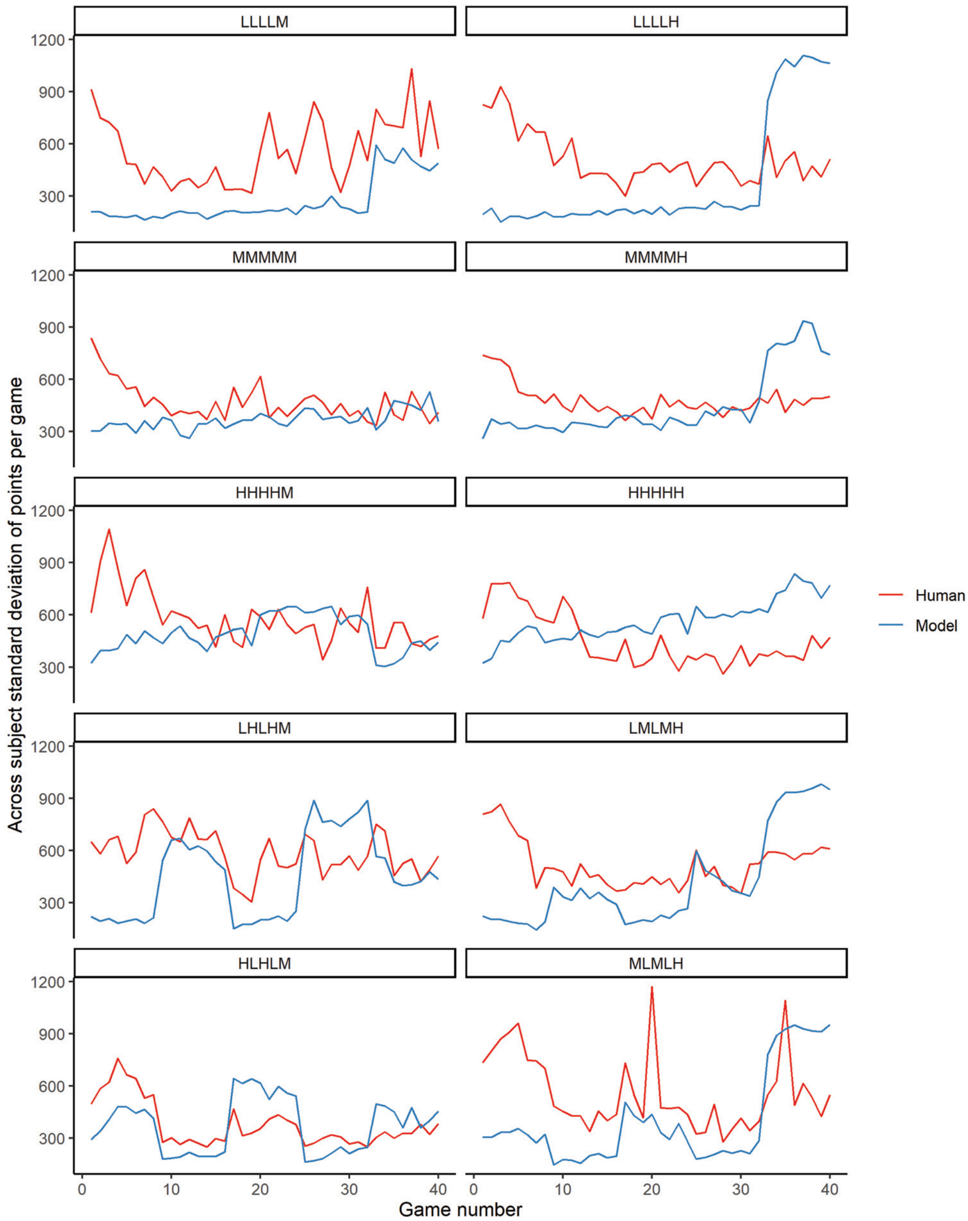
contrast, the variability across simulations in this study results only from random processes inherent to the model (e.g., motor noise) and randomness in the game environment (e.g., random order of track angles).

Figure G1 compares the within participant variability between human and model players; for each participant (and simulation), we computed the standard deviation across points per block. Recall that players show a period of adjustment when switching accelerations, and especially when switching to a higher acceleration. From the figure, this is reflected in both the human and model curves as an increase in the standard deviation measure when transitioning from a lower to a higher acceleration block. This figure also shows that model performance within each block does appear to be more variable than that of human participants, particularly during the high acceleration blocks.

Figure G2 compares the across participant variability between human and model players; for each game, we computed the standard deviation across the points scored by each player. One might imagine that both model and human participant variability would decrease with game number because the slower learners would be able to catch up with the faster learners in the later games. From

(*Appendices continue*)

Figure G2
Plot of Standard Deviations of Points per Game Across Participants



Note. M = Medium; L = Low; H = High. See the online article for the color version of this figure.

the figure, models do show a slight increase in across participant variability as game number increases, but that contrasts with the slight decreasing trend observed in human participants. Both human and model players also display an increase in across participant variability when switching to a higher acceleration, but the effect appears to be more drastic for models than it is for humans. Both discrepancies could be due to the consistently poor performance of a few model simulations that get stuck in a nonoptimal selection of control values. The performance gap between these poor performing models and the other models only widens with experience or when forced to adapt to increased accelerations. With human participants however, it appears that even those who perform poorly manage to get better with experience.

Received November 19, 2020
Revision received May 8, 2021
Accepted June 13, 2021 ■