Time-related Effects of Speed on Motor Skill Acquisition

Pierre G. Gianferrara (pgianfer@andrew.cmu.edu)
Shawn Betts (sabetts@andrew.cmu.edu)
John R. Anderson (ja0s@andrew.cmu.edu)
Department of Psychology, Carnegie Mellon University, 5000 Forbes Ave
Pittsburgh, PA 15213 USA

Abstract

Anderson *et al*. (2019) present an ACT-R model of how humans learn to play rapid-action video games. To further test this model, we utilized new measures of action timing and sequencing to predict skill acquisition in a controlled motor task named *Auto Orbit*. Our first goal was to use these measures to capture time-related effects of speed on motor skill acquisition, operationalized as a performance score. Our second goal was to compare human and model motor skill learning. Our results suggest that humans rely on different motor timing systems in the sub- and supra-second time scales. While our model successfully learned to play *Auto Orbit*, some discrepancies in terms of motor learning were noted as well. Future research is needed to improve the current model parameterization and enable ACT-R's motor module to engage in rhythmic behavior at fast speeds.

Keywords: timing; sequencing; motor control; speed; rhythm; variability; ACT-R; motor skill acquisition

Introduction

Many everyday tasks such as typing, chopping, and playing a musical instrument require one to acquire complex motor skills. Motor skill learning has been defined as the neuronal changes that enable an organism to execute a motor task better, faster, and more accurately over time (Diedrichsen & Kornysheva, 2015). Evidence from the motor control literature suggests that humans may rely on chunking strategies over the course of motor skill learning resulting in more predictable motor action sequences (Verstynen *et al*., 2012; Beukema, Diedrichsen & Verstynen, 2019).

In addition to the chunking of motor actions, one needs to incrementally estimate the correct timing of upcoming motor actions (Decety, Jeannerod & Prablanc, 1989; Palmer & Pfordresher, 2003). One common model of timing is the attentional gate model developed by Zakay and Block (1996) which relies on the generation of regular pulses to keep track of time. However, prior research on the neuroscience of timing provides evidence suggesting that this model may not be sufficient to capture the full complexity of motor timing (Coull, Cheng & Meck, 2011; Breska & Ivry, 2016).

According to Lewis and Miall (2003), motor timing may differ depending on the time scale of the motor actions that are executed. On the one hand, motor actions in the second range were shown to depend on a 'cognitively controlled' timing system heavily dependent on prefrontal and parietal neural processing. On the other hand, motor actions in the sub-second range were shown to depend on an 'automatic' timing system mostly dependent on motor circuits (Lewis & Miall, 2003).

In this study, we used the ACT-R production system, which successfully modeled skill acquisition in a previous complex task (Anderson *et al*., 2019), to assess potential time-related effects of speed on motor skill acquisition. To do so, we designed a novel video game, *Auto Orbit,* inspired by *Space Fortress* (Mané & Donchin, 1989). Our main goal was to determine to what extent ACT-R is currently able to capture the detail of human motor learning in such games. In this study, we manipulated the game speed, such that agents would play the same video game at a faster or slower speed. Our principal analysis compared ACT-R's motor learning to human learning and strove to capture the different elements of motor skill acquisition in a time-dependent statistical framework.

Auto Orbit Video Game

In *Auto Orbit,* a spaceship is rotating in an orbit at a fixed speed around a balloon (circle-shaped target) placed in the middle of the screen (see Figure 1). The player must periodically adjust the ship's aim and regularly shoot missiles within a specific firing interval. Each successful shot triggers a quick electronic sound and results in the balloon being inflated by 1/10 of its full size. Once the balloon is fully inflated, the player needs to execute a quick double shot shorter than 250 ms to burst the balloon and complete a game cycle. Balloon bursts were each rewarded by a fixed number of points dependent on the game speed, while misses (unsuccessful missiles) resulted in a penalty of 2 points. Each game was broken down into a series of game cycles that started with a "balloon respawn" game event and ended with a "balloon burst" game event. For each game, a log file was recorded with 16-ms temporal resolution.

Controlling the space ship in *Auto Orbit* involved three actions: rotating clockwise by 15 degrees ("D" key), rotating counterclockwise by 15 degrees ("A" key), and launching a missile ("L" key). During the game, the specified firing interval was learned through balloon resets as the lower bound, and balloon deflations as the upper bound. Each shot that was faster than the lower bound resulted in a reset characterized by the balloon popping on the screen. Conversely, the player's failure to hit the balloon before the upper bound resulted in a balloon deflation characterized by the balloon dwindling at a constant deflation rate of 1 % of the balloon's full size or 0.18 pixels per game tick. Finally, in order to add some noise in the video game, random ship rotations of 60 to 120 degrees occurred with 1/3 probability at the beginning of every game cycle, and the players were then given additional time to re-adjust the ship's aim. Rotation onset time was randomly generated according to a uniform distribution with 1 s. as the minimum rotation onset time and 4 s. as the maximum rotation onset time. After each ship rotation, agents had 2 s. to adjust the ship's aim and continue firing before the balloon started deflating. An illustration of the *Auto Orbit* interface is depicted on Figure 1. One can play the *Auto Orbit* video game using the following link: http://andersonlab.net/orbit/signin.html

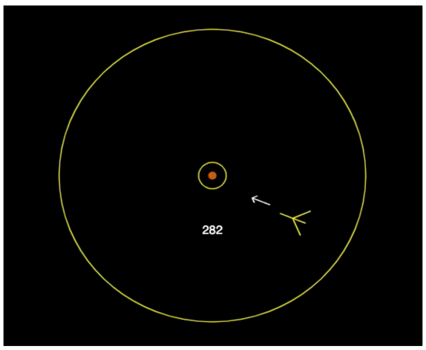

Figure 1: Visualization of the *Auto Orbit* video game interface.

ACT-R Model of Skill Acquisition

The ACT-R model was adapted from past modelling work by Anderson *et al*. (2019). First, operators were stored in declarative memory to represent the model's strategy for playing the video game. Operators were set up such that the model would first adjust the ship's aim (angular orientation relative to the balloon center) and would then monitor the timing of its shots (time that needed to elapse after a shot before the model could fire another missile). Over the games, the model became faster in executing these operators through a process called production compilation, which converts operators into direct action rules (Anderson *et al*., 2019).

Specifically, production compilation involves the integration of two productions into a novel one, thus bypassing the retrieval of operators from declarative memory. Each newly learned production is initially assigned a utility of zero and starts competing with its original parent production. Every time the new production is selected, its utility gets progressively updated until it reaches its true value. Utility values are incrementally adjusted based on the difference learning equation, which is shown on equation 1 (see Anderson (2007) and Anderson *et al*. (2019) for further details):

$$U_i(n) = U_i(n-1) + \alpha[R_i(n) - U_i(n-1)] \quad (1)$$

where $U_i(n)$ corresponds to the nth update of the ith production utility, $R_i(n)$ corresponds to the reward at the nth update, $U_i(n-1)$ corresponds to the ith production utility at the n-1th update, and α is the learning rate. Note that α was set to 0.2 in our model.

Second, monitoring time was a crucial aspect of skill acquisition. The model kept track of time in its temporal module through a pacemaker-accumulator internal clock timing system that generated regular pulses (Taatgen, van Rijn & Anderson, 2007). Pulses were monitored in the form of time ticks whose value could be accessed via a temporal buffer (Taatgen *et al*., 2007). In the ACT-R model, time ticks were reset at the start of the game, and every time the model fired a missile.

Finally, the controller module monitored and refined the estimation of a number of key game parameters including the ship's aim and the shot timing threshold (Anderson *et al.,* 2019). The controller module progressively adjusted these parameters via a control tuning mechanism starting with greater tolerance and narrowing the tolerance over time, so that values could be progressively tuned (Anderson *et al*. 2019; Seow, Betts & Anderson, 2019). One key parameter that impacted the speed of control tuning was the temperature (see equation 2).

$$T(t) = A/(1 + B.t) \quad (2)$$

where A is the initial temperature which was set to 1.0, B is a scaling factor which was set to its default value of 1/180, and t is the time in seconds that elapsed since the start of the game. In our model, control tuning critically involved two control values: First, the model needs to turn the ship so it will be aimed at the fortress when it later fires. Since the ship is moving, it searches for an offset in its aim from -18 degrees (lower bound) to 0 degrees (upper bound). Second, the ship needs to pace its fires to avoid both resets and deflations. The controller considers a range from 8 time ticks or 126 ms (lower bound) to 28 time ticks or 1476 ms (upper bound).

The range for shot timing in this model was much larger than in Anderson *et al*. (2019) because there was no information about what the appropriate time was whereas in the original *Space Fortress*, subjects were told that the lower bound is 250 ms. In *Auto Orbit*, the model narrowed the firing range to search as it gained information. Specifically, the "detect-reset" and "detect-deflate" productions were responsible for adjusting the firing time threshold range of parameters. While the time threshold's *upper* bound progressively *decreased* during deflations, the time threshold's *lower* bound progressively *increased* during resets.

Methods

Experimental Design

In this experiment, all human subjects and ACT-R models played a total of 15 games that were 3 minutes in duration (45 minutes in total). Each subject was randomly assigned to one of three possible conditions corresponding to the *fast,* *medium* and *slow* game speed (see Table 1). In the fast game speed condition, the ship's orbital speed was 1.0 pixel per game tick (16 ms), and the missile speed was 10 pixels per game tick. Agents assigned to that condition needed to fire within the [250 ms – 600 ms] interval. Each fired missile that resulted into a miss was penalized by a loss of 2 points, and each balloon burst was rewarded by a gain of 100 points.

In the medium game speed condition, all aspects of the game happened at half the speed of the fast condition including the timing of shots. This was halved again for the slow game speed condition. Individuals earned points as a way to get monetarily compensated for their gameplay. The total number of points per game was our main measure of skill level which was assessed independently in each game speed. We designed a point system that met the two following criteria: First, participants should earn the greatest amount of points per balloon burst in the slowest conditions so that all players get fairly compensated for the same game length (3 minutes); Second, the point system should be adjusted such that participants in easy conditions (e.g., slow speed) may not earn significantly more than participants in hard conditions (e.g., fast speed). Each balloon burst thus led to a reward that increased by 100 points each time the game parameters were halved (see Table 1). Note that we did not compare performance scores across speeds.

Table 1: Description of the three game speed conditions

Game Speed	Speed Multiplier	Resets	Deflations	Points
Fast	1.0	250 ms	600 ms	100/burst
Medium	0.5	500 ms	1,200 ms	200/burst
Slow	0.25	1,000 ms	2,400 ms	300/burst

Agents

Human Participants We are reporting data from 60 human participants randomly assigned to each of the 3 game speed conditions. Participants were aged 21 to 40 years-old (M = 30.5, SD = 4.7). Forty were male and 20 were female. All participants were recruited on the Amazon Mechanical Turk (mTurk) online platform. Subjects earned a base pay of $4 for completing the experiment, in addition to a bonus which was proportional to their performance (in points) as specified on Table 1. On average, participants earned a bonus of $5.

ACT-R Models Ninety ACT-R model simulations were conducted in each of the 3 game speed conditions (270 model runs in total). All models were initialized with the same parameters.

Procedure

The mTurk experiment consisted of four main steps. First, participants filled out a short background questionnaire including questions about the participants' demographics. Second, they read a short description of the *Auto Orbit* video game including game play instructions. Third, participants were randomly assigned to one out of the three experimental conditions (see Table 1) and completed 15 games that lasted 3 minutes each. Finally, they filled out some additional questionnaires where they provided feedback and wrote about strategies that they used during the experiment.

Experimental Measures

In this study, we were interested in a number of experimental measures pertaining to motor skill acquisition. Our main dependent variable was *performance,* which was operationalized as points earned per game (see Table 1). The design comprised a total of 4 independent variables: the keypress sequence *entropy*, the *inter-shot-interval (ISI) coefficient of variation logarithm,* and the shot *periodicity* and *regularity*. All measures were computed across game cycles ("balloon respawn" to "balloon burst") without random rotations for every agent and every game.

Entropy The entropy was our measure of keypress sequence regularity in *Auto Orbit*. We focused on the relative frequency of various keypress triples. With three keys ('F': fire, 'L': left, 'R': right) there are 3^3 = 27 triples. We computed the proportions of each keypress triple per game by using a non-overlapping counting method (Python count() function) with Laplace smoothing for each keypress triple in all game cycles. We used the Shannon entropy measure, which quantifies unpredictability of information content in a probability distribution (Shannon & Weaver, 1949). Shannon entropy's formula is given in equation 3:

$$H(X) = -\sum_{i=1}^{27} p_i \,.\, log_2\, p_i \qquad (3)$$

where X refers to a game number and p_i refers to the probability of the ith triple. This entropy measure could vary from 0 (only 1 triple throughout) to 4.75 (all triples equally likely). We expected the entropy measure to decrease as subjects developed a systematic approach to the game.

Log CV Inter-Shot-Interval (ISI) In order to measure shot timing variability, we extracted the time interval between consecutive shots in milliseconds, named inter-shot-interval (ISI) within game cycles. For each game of every agent, we then made use of the coefficient of variation (CV), which is defined as the standard deviation divided by the mean of the ISIs, consistent with previous research (Loehr & Palmer, 2009). An average CV of the ISIs was computed across game cycles within each agent's game. Because our measure of CV and our performance variable were not linearly related, we carried out data transformation on CV and calculated its logarithm instead. We expected this measure to decrease as subjects became more skilled.

Periodicity and Regularity The shot regularity measure was computed based on the shots autocorrelation function within games. This method has been used in the music information retrieval literature to perform meter extraction (Brown, 1993). For each game cycle, we first re-preprocessed agents' log files such that we would get a single discrete time series of fire events, where individual entries corresponded to successive game ticks of 16 ms. At every tick, a 1 indicated a fire keypress hold event and a 0 indicated a fire keypress release event. We could then compute the correlation coefficient (Box & Jenkins, 1976) of keypress actions at a particular lag (see equation 4) using the 'acf' function from the statsmodels time series analysis ('tsa') library in Python (McKinney, Perktold & Seabold, 2011):

$$r_l = \frac{\sum_{i=1}^{100-l}(x_i - \bar{x})(x_{i+l} - \bar{x})}{\sum_{i=1}^{100}(x_i - \bar{x})^2} \tag{4}$$

where l refers to the current time lag. A total of 100 time lags of 16 ms were investigated in each autocorrelation function. We averaged the autocorrelation function across game cycles of each agent's game. As a result, for each agent, we obtained 15 autocorrelation functions corresponding to each of the 15 games. Figure 2 displays an example of a game autocorrelation function in a subject. Positive peaks in this function reflect lags at which the fire keys tended to be pressed.

Finally, we used each game autocorrelation function to extract our two measures of interest: *periodicity* and *regularity.* To do so, we identified the first non-zero lag positive peak of the autocorrelation function. Periodicity was the lag for this peak at which fires tended to be pressed. Regularity was the height of this function and reflected how consistently keypresses occurred at this lag. In our example, the first non-zero autocorrelation positive peak has been identified with a red bar (see Figure 2). We expected that better play would be associated with decreased periodicity as subjects got their shots closer to the threshold, and increased regularity as they got more consistent in their timing.

Data Analysis

Our data set consisted of the above measures, each recorded once per game and per agent (i.e., in humans and models).

Linear Mixed-Effects Model In order to assess each measure's individual effect on skill acquisition, we fit a linear mixed-effects model (LMEM) across game speeds. Our main dependent variable was performance, operationalized as points earned per game. Our predictors were the four measures described earlier, each modeled as a fixed factor. In addition to our fixed factors, we added two random factors to account for some of the variability in our performance measure that was not explained by our four linear predictors. The first random factor accounted for differences across participants' skill levels and was modeled as a random intercept. The second random factor accounted for residual variance in performance related to individual game numbers that could not be captured by our four fixed factors.

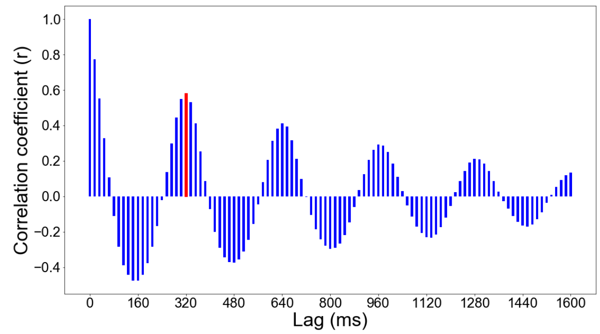

Figure 2: Autocorrelation function in game 11 of one of the subjects in the fast speed condition. The red bar indicates the periodicity (lag) and regularity (correlation coefficient) of this subject's shot timings in that game.

In R, we used the 'lme4' (Bates *et al*., 2014) package to fit our linear mixed-effects model. The model was written as *lmer*(*Performance* ~ *Entropy* + *logCV* + *Periodicity* + *Regularity* + (1|*Subject*) + (1|*GameNb*)). For each model, the 95 % confidence interval was computed for each estimate using bootstrapping with resampling ('bootMer' function in R). A total of 1000 simulations were run for each bootstrapped 95 % confidence interval.

Results

Behavioral Results

We hereby present the results from human games and ACT-R model simulations across the three game speed conditions (fast, medium, and slow). For each measure of interest, we report the mean within games across agents along with the standard error. Because the ACT-R models were all initialized with the same parameter values, there is lower variability among models than humans for each measure.

We first report the performance results across games in humans and models (see Figure 3). Humans and models achieved similar numbers of points, both showing rapid initial improvement approaching an asymptote by 15 games. However, some differences are worth noting: In the fast speed condition, humans performed somewhat better than the models. In the medium and slow speed conditions, models had a somewhat steeper slope than humans.

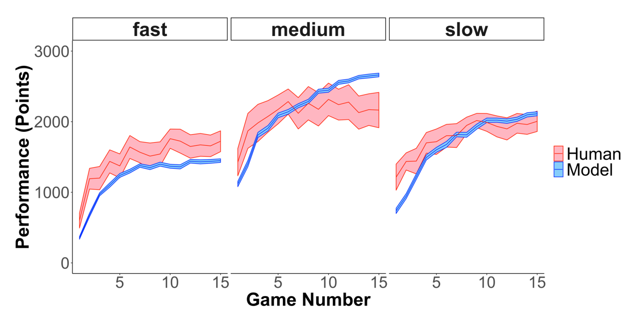

Figure 3: Performance scores over the games. Human performance is indicated in red (N = 20 per speed); model performance is indicated in blue (N = 90 per speed). Shaded areas indicate the standard error of the mean (S.E.M).

In terms of keypress sequencing, both models and humans showed similar levels of entropy with an increase for slower games. (Figure 4a), but humans' entropy progressively decreased in early games towards their asymptote whereas models' entropy was constant across all games. We think this reflects the fact that models have a constant strategy, whereas subjects' strategies are evolving during the early games and they only settle down to a constant strategy in later games.

The results with respect to ISI variability (Figure 4b), are similar to entropy. While models show some decrease over games, subjects show a more drastic decrease with a large early drop in the 2 first games. We think this reflects subjects' evolving strategy and progressive adaptation to the game's shot timing constraints. As such, early changes will produce large changes in motor timing.

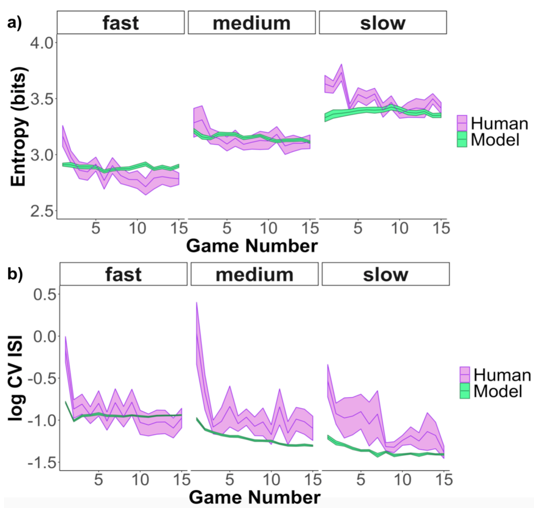

Figure 4: a) Entropy over the games in humans and models. b) Logarithm of the inter-shot intervals (ISI in ms) coefficient of variation (CV). Shaded areas indicate the standard error of the mean (S.E.M).

Analysis of shot timing autocorrelations provided periodicity and regularity measures (Figure 5). With regards to the shot periodicity, humans and models were quite similar. Both had shot timings within the assigned game speeds' firing intervals (see Figure 5a), and both models and subjects showed an increase in periodicity in the slow speed condition. (see Figure 5b). That increase reflected an early tendency to fire too soon, which both models and humans had to learn to change.

Both models and humans showed an increase in regularity over games, but human regularity increased as the game speed got faster whereas model regularity did not increase with speed. The regularity in the model reflects its timing mechanism, which is a variant of the attentional gate model. That model produces a variability that scales with duration, so that regularity will not change much. In contrast, subjects may be changing their timing processes as they move to timing actions well under a second.

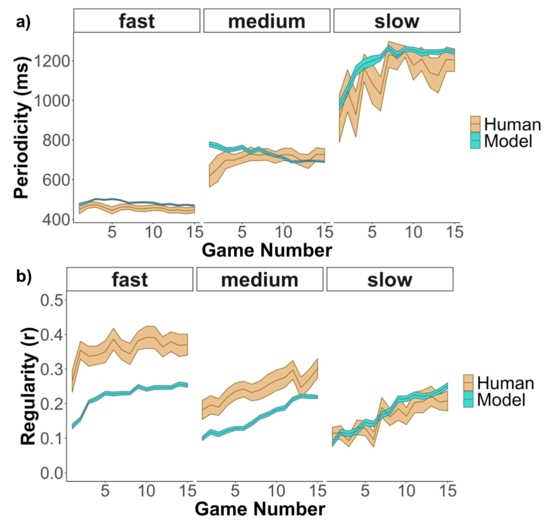

Figure 5: a) Periodicity over the games in humans and models. b) Regularity over the games. Shaded areas indicate the standard error of the mean (S.E.M).

Linear Mixed-Effects Model

We hereby report the LMEM results pertaining to humans and ACT-R models separately. Table 2 displays the human LMEM results. Three models were fitted to each game speed data set. Entropy and the logarithm of the ISI coefficient of variation were both predictive of performance across all speeds. A decrease in entropy was reliably predictive of performance, and both lower and upper bounds of the confidence interval were negative. Similarly, a decrease in ISI variability was also reliably predictive of performance across all speeds. With respect to the shots autocorrelation measures, a positive effect of regularity (more regular) on performance and negative effect of periodicity (faster firing rate) on performance were found in the fast and medium game speeds, but *not* in the slow game speed. All significant effects are in line with our expectations about these factors. The fact that regularity and periodicity are not predictive in the slow game speed reflects the fact that it is not time pressured.

Table 2: Human LMEM results across game speeds

	Game Speed - Human					
	Fast		*Medium*		*Slow*	
	Estimate	95 % CI	Estimate	95 % CI	Estimate	95 % CI
Entropy	-490^{***}	$(-638, -312)$	-641^{***}	$(-844, -421)$	-295^{***}	$(-452, -136)$
Log CV ISI	-164^{***}	$(-242, -84)$	-84^{*}	$(-161, -12)$	-146^{***}	$(-206, -83)$
Regularity	1171^{***}	$(732, 1650)$	1062^{**}	$(469, 1677)$	378	$(-141, 943)$
Periodicity	-1.31^{***}	$(-2.25, -0.38)$	-0.52^{***}	$(-0.87, -0.15)$	0.10	$(-0.03, 0.23)$
Adjusted R^2	0.88		0.88		0.79	

$^{***}p < .001$; $^{**}p < .01$; $^{*}p < .05$

Table 3 displays the ACT-R models' LMEMs. We found a consistent positive effect of shot regularity on performance across all three game speeds. In contrast to

Table 2, there are a number of significant effects in the opposite direction of expectation. Increased periodicity is associated with more points in the slow condition. We think this is associated with the initial too-fast firing in the slow condition (Figure 5) and is partly responsible for the lack of an effect for humans in Table 2. The other discrepancy is that greater ISI variability was predictive of higher performance in the fast and slow game speeds. This is puzzling, because in terms of simple correlation, there is no correlation between points and this measure in the fast game (r = .005) and a weak negative correlation in the slow games (r = -.196). The direct correlation for medium speed, where Table 3 shows the expected negative effect, is .581.

Table 3: ACT-R models LMEM results across game speeds

	Game Speed - ACT-R models					
	Fast		*Medium*		*Slow*	
	Estimate	95 % CI	Estimate	95 % CI	Estimate	95 % CI
Entropy	-451^{***}	$(-531, -372)$	-15	$(-118, 97)$	53	$(-22, 127)$
Log CV ISI	1034^{***}	$(896, 1177)$	-372^{***}	$(-531, -205)$	604^{***}	$(477, 734)$
Regularity	974^{***}	$(745, 1201)$	1043^{***}	$(754, 1348)$	558^{***}	$(326, 822)$
Periodicity	-0.05	$(-0.40, 0.27)$	-0.62^{***}	$(-0.82, -0.43)$	0.32^{***}	$(0.21, 0.44)$
Adjusted R^2	0.73		0.66		0.65	

$^{***}p < .001$; $^{**}p < .01$; $^{*}p < .05$

Discussion and Conclusion

ACT-R models and humans showed similar improvement scores in the *Auto Orbit* video game. Motor skill acquisition was characterized by a fast performance increase in early games, and a slower learning rate in later games where performance progressively plateaued towards an asymptote. In terms of motor behavior, humans and models both learned to be more regular and less variable in terms of keypress sequential patterns and shot timing. One challenge in the *Auto Orbit* video game was for players to learn how to shoot within a firing interval bounded by resets and deflations. Shot timing autocorrelation analyses revealed that humans and models learned to fire missiles within their assigned game speed firing interval (periodicity) with increasing rhythmicity (regularity).

Nevertheless, our analyses also revealed a number of motor learning differences between humans and models which are worth discussing. First, we found that humans' keypress patterns and shot timing were more variable than models, particularly in early games, but quickly converged towards models' variability levels as performance increased. These variability patterns in humans fit with previous neuroscience (Wu *et al.,* 2014) and motor skill learning research (Caramiaux *et al*., 2018) suggesting that motor and timing variability may predict performance over the course of motor skill acquisition.

Second, while shot regularity increased in a similar fashion at the slow speed across humans and models, we found that human subjects' shot regularity levels were higher at faster speeds, whereas models' regularity levels remained constant across speeds. This result may be due to a higher reliance on motor circuits (Lewis & Miall, 2003) and subcortical modulation (Ivry & Spencer, 2004; Koch *et al*., 2007) under fast speed constraints. Specifically, past research suggests that motor timing tasks involving fast discrete actions such as repetitive keypresses may heavily recruit the cerebellum through dynamic sensorimotor learning and motor error correction (Koch *et al*., 2007; Breska & Ivry, 2016).

While model performance and motor learning were relatively close to humans, improvements can be made to optimize the ACT-R model and better simulate human motor skill acquisition. One option would be to change the parameterization of ACT-R to make the model more variable, both within- and between-models. In terms of within-models variability, one could vary the noise levels and learning rates related to different components of the model. One example is the temporal module noise parameter whose increase may lead to further shot timing variability in ACT-R. Another example is the utility learning and production compilation learning rate (α in equation 1), which controls the speed at which newly formed productions replace their original parent productions. Specifically, high values of α typically lead to a faster rate of production compilation and skill acquisition whereas low values of α typically lead to a lower rate of production compilation and skill acquisition. One last example is the initial temperature (A in equation 2), whose value assignment may lead to different degrees of randomness in control tuning. Generally, lower initial temperatures enable the model to incorporate more of its learning experience into its game play, but they also increase the risk of converging towards non-optimal values if the initial temperature is too low.

As to between-models variability, past research by Anderson *et al*. (2019) explored potential performance fluctuations related to the adjustment of a few selected parameters. Specifically, the authors explored the effects of α (see equation 1) in the [0.025, 0.3] range, and the initial temperature (A in equation 2) in the [0.1, 2.0] range. It would be of interest to further explore the stochastic initialization of these parameters in our current ACT-R model to determine whether one could replicate humans' inter-subject variability patterns.

A second option would be to modify the initialization of operators to enable the model to adjust its behavior to a fast vs. slow game speed depending on shot timing threshold information in the controller module. Alternatively, one could vary the order in which operators are retrieved at different phases of the game such that the model would initially prioritize shooting over aiming, but would progressively switch to a more optimal strategy that prioritizes aiming over shooting.

Last but not least, our current results strongly suggest that ACT-R's current motor module needs the ability to adjust its motor behavior according to the speed at which it executes motor actions. One striking result was that human shot timing became increasingly rhythmic as the game speed got faster. One way of modeling this time-related effect of speed on motor behavior would be to augment the current motor module with its own timing component, such that the model would fire missiles with increasing rhythmicity at faster speeds. This novel addition would fit with ACT-R being a template of human behavior.

In sum, we have shown that human motor skill learning was characterized by time-independent and time-dependent effects of speed. On the one hand, variability in keypress sequencing and motor timing were shown to predict skill acquisition regardless of the game speed. On the other hand, motor timing regularity and periodicity were shown to only be predictive of performance in the sub-second range. As a way to model these effects in ACT-R, we suggest a number of improvements which include incorporating a timing component into ACT-R's motor module.

Acknowledgements

This research was supported by the Office of Naval Research Grant N00014-15-1-2151 and AFOSR/AFRL award FA9550-18-1-0251. We would like to thank Dan Bothell for his help with the implementation of the *Auto Orbit* video game and the ACT-R model.

References

Anderson, J. R. (2007). *How Can the Human Mind Occur in the Physical Universe?*. New York, NY: Oxford University Press.

Anderson, J. R., Betts, S., Bothell, D., Hope, R., & Lebiere, C. (2019). Learning rapid and precise skills. *Psychological Review, 126*, 727–760.

Bates, D., Mächler, M., Bolker, B., & Walker, S. (2014). Fitting linear mixed-effects models using lme4. *J Stat Softw* arXiv:1406.5823.

Beukema, P., Diedrichsen, J., & Verstynen, T. D. (2019). Binding during sequence learning does not alter cortical representations of individual actions. *Journal of Neuroscience, 39*, 6968-6977.

Box, G. E. P., & Jenkins, G. M. (1976). *Time Series Analysis: Forecasting and Control*. San Francisco, CA: Holden-Day.

Breska, A., & Ivry, R. B. (2016). Taxonomies of timing: where does the cerebellum fit in?. *Current opinion in behavioral sciences, 8*, 282-288.

Brown, J. C. (1993). Determination of the meter of musical scores by autocorrelation. *The Journal of the Acoustical Society of America, 94*, 1953-1957.

Caramiaux, B., Bevilacqua, F., Wanderley, M. M., & Palmer, C. (2018). Dissociable effects of practice variability on learning motor and timing skills. *PloS one, 13*: e0193580.

Coull, J. T., Cheng, R. K., & Meck, W. H. (2011). Neuroanatomical and neurochemical substrates of timing. *Neuropsychopharmacology, 36*, 3-25.

Decety, J., Jeannerod, M., & Prablanc, C. (1989). The timing of mentally represented actions. *Behavioural brain research, 34*, 35-42.

Diedrichsen, J., & Kornysheva, K. (2015). Motor skill learning between selection and execution. *Trends in cognitive sciences, 19*, 227-233.

Ivry, R. B., & Spencer, R. M. (2004). The neural representation of time. *Current opinion in neurobiology, 14*, 225-232.

Koch, G., Oliveri, M., Torriero, S., Salerno, S., Gerfo, E. L., & Caltagirone, C. (2007). Repetitive TMS of cerebellum interferes with millisecond time processing. *Experimental brain research, 179*, 291-299.

Lewis, P. A., & Miall, R. C. (2003). Distinct systems for automatic and cognitively controlled time measurement: evidence from neuroimaging. *Current opinion in neurobiology, 13*, 250-255.

Loehr, J. D., & Palmer, C. (2009). Sequential and biomechanical factors constrain timing and motion in tapping. *Journal of Motor Behavior, 41*, 128-136.

Mané, A., & Donchin, E. (1989). The space fortress game. *Acta psychologica, 71*, 17-22.

McKinney, W., Perktold, J., & Seabold, S. (2011). Time series analysis in Python with statsmodels. *Jarrodmillman. Com*, 96-102.

Palmer, C., & Pfordresher, P. Q. (2003). Incremental planning in sequence production. *Psychological Review, 110*, 683-712.

Seow, R. Y. T., Betts, S., & Anderson, J. R. (2019). Transfer effects of varied practice and adaptation to changes in complex skill acquisition. In Stewart, T.C. (Ed.). *Proceedings of the 17th International Conference on Cognitive Modelling* (pp. 222-227). Waterloo, Canada: University of Waterloo.

Shannon, C. E., & Weaver, W. (1949). *The mathematical theory of communication*. Urbana, IL: University of Illinois Press.

Taatgen, N. A., Van Rijn, H., & Anderson, J. (2007). An integrated theory of prospective time interval estimation: The role of cognition, attention, and learning. *Psychological Review, 114*, 577-598.

Verstynen, T., Phillips, J., Braun, E., Workman, B., Schunn, C., & Schneider, W. (2012). Dynamic sensorimotor planning during long-term sequence learning: the role of variability, response chunking and planning errors. *PLoS One, 7*: e47336.

Wu, H. G., Miyamoto, Y. R., Castro, L. N. G., Ölveczky, B. P., & Smith, M. A. (2014). Temporal structure of motor variability is dynamically regulated and predicts motor learning ability. *Nature neuroscience, 17*, 312-321.

Zakay, D., & Block, R. A. (1996). The role of attention in time estimation processes. *Advances in Psychology*, *115,* 143–164.