Title
Modeling Millisecond Time Interval Estimation in Space Fortress Game

Journal
Proceedings of the Annual Meeting of the Cognitive Science Society, 34(34)

ISSN
1069-7977

Authors
Moon, Jungaa
Anderson, John

Publication Date
2012

Peer reviewed

Modeling Millisecond Time Interval Estimation in Space Fortress Game

Jungaa Moon (jungaam@andrew.cmu.edu)
John R. Anderson (ja+@cmu.edu)
Department of Psychology, Carnegie Mellon University
Pittsburgh, PA 15213 USA

Abstract

We investigated sources of the asymmetric bias found in estimation of a time interval (250-400 ms) embedded in the Space Fortress task (Donchin, 1989). Two hypotheses to explain this bias were tested in a behavioral experiment: 1) contamination from a different time interval representation, and 2) pressure to complete the task in time. Participants alternated between producing the target interval and producing either a shorter or a longer interval while the total time allowed for the task was manipulated. The results showed that the target interval estimate was significantly influenced by both manipulations. The effects were captured by incorporating the timing model of Taatgen and Van Rijn (2011) into the ACT-R model for Space Fortress (Bothell, 2010). Time estimation performed in a dynamic task requires understanding the influence of external temporal tasks as well as the procedural demands of performing multiple tasks under time pressure.

Keywords: Time estimation; cognitive model; multitasking.

Introduction

Time interval estimation underlies various skills such as motor control (Ivry, Spencer, Zelaznik, & Diedrichsen, 2002), musical performance (Jones, 1990), and speech processing (Schirmer, 2004). Millisecond-to-second interval timing is critical in real-time dynamic tasks that require adaptive responses to the changing environment. For instance, when driving it is necessary to estimate how long one can attend to a navigator before switching back to attending to the road and driving control (Salvucci, Taatgen, & Kushleyeva, 2006).

Time estimation can be studied under various paradigms (Zakay, 1990). Participants can be asked to retrospectively generate verbal estimation of an interval, to judge whether a presented interval is the same length as a target interval, or reproduce a target interval. Studies using the reproduction paradigm typically show response distributions that are 1) centered at the real-time criteria, 2) symmetrical, and 3) have a standard deviations that increase in proportion to the mean interval (e.g. Rakitin, Gibbon, Penney, Malapani, Hinton, & Meck, 1998).

In most studies under those paradigms, time estimation is often an isolated task performed in a static environment. It is the primary task on which participants focus, even when a secondary task is given for various purposes (Fortin, Rousseau, Bourque, Kirouac, 1993; Rakitin, et al., 1998). However, one may wonder to what extent the time estimation performed in those paradigms reflects the time estimation that people usually perform in various multitasking situations. As in the driving example, time estimation is often an implicit secondary task that one performs to coordinate primary tasks. In addition, people sometimes need to estimate multiple time intervals concurrently (e.g., cooking breakfast). It seems plausible that time estimation in those circumstances will exhibit properties not seen when it is performed as an isolated task in a static environment. We investigated this question in the Space Fortress task (Donchin, 1989), a video game that simulates real-time complex tasks performed in dynamic environments (e.g., piloting an aircraft).

Time Interval Estimation in Space Fortress Task

The goal of the Space Fortress task (Figure 1) is to maximize the total scores by navigating a ship in a frictionless space, destroying a fortress multiple times and handling mines while protecting the ship from the fortress and mines. The participant navigates the ship by rotating left or right (A/D keys) or thrusting (W key) to make it fly within an area enclosed by two hexagons. A fortress stationed in the center rotates like a turret, tracking the ship's trajectory and firing shells at it. The participant has to shoot the fortress with a missile (spacebar) at least ten times and then make a rapid double-shot to destroy it.

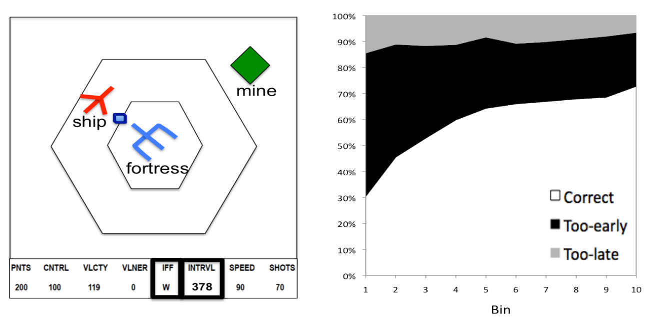

Figure 1: Schematic representation of the Space Fortress task (left) and performance in the IFF tapping task (right).

The mine task, which is the focus of the current study, consists of a series of activities in a specific order. At the beginning of the game, the participant is presented with a screen with three alphabetic letters ('foe letters') and asked to remember them. During the game, a mine appears at a random location on the screen 5 seconds after the destruction of the previous mine and starts pursuing the ship with the intent of crashing into the ship. When a mine appears, the participant has to check a letter that appears in the IFF box in the bottom panel (see Figure 1). The mine is a foe if the letter matches one of the foe letters; otherwise, it is a friend. Mine identification is a version of the Sternberg memory-scanning task (Sternberg, 1966). If the mine is a foe, one has to perform an Identify Friend/Foe (IFF) tapping task, which involves tapping the J key twice with a 250-400 ms interval ('IFF interval') between the two key presses. Once a correct interval has been generated, the mine can be destroyed by aiming the ship at the mine and firing a missile. A missile can be fired even after a wrong IFF interval, but the missile can destroy the mine only after a correct IFF interval. If the mine is a friend, then the IFF tapping task should not be performed and the mine can be destroyed by a missile shot. If all steps are completed successfully before the mine reaches the ship, then the mine is destroyed and points are earned. Otherwise, the mine eventually collides with the ship and points are lost.

As a time interval estimation task, the IFF tapping task has three notable characteristics. First, it is a prospective time estimation task. Participants are initially told the target interval in written instructions, and then produce the interval whenever a foe mine appears during a game. Immediately after each attempt, the produced interval is displayed as feedback (e.g., "378") in the INTRVL box in the bottom panel. Second, both the initiation and the termination of the interval are under the control of participants. Finally, and most importantly, it is performed not as an isolated task but as part of a real-time complex task. The game requires time-sharing multiple tasks such as navigating the ship while dealing with the fortress and the mines. Even within the mine task, a series of activities precede (checking the letter and determining the mine's identity) and follow (aiming the ship and firing a missile) the IFF tapping task, all of which need to be completed within a brief period of time, usually 2-3 seconds.

A study previously conducted in our laboratory revealed an interesting pattern of performance in the IFF tapping task. Figure 1 (right) displays the percentage of responses within each of three categories: correct (the produced interval was between 250-400 ms), too-early (<250 ms), and too-late (>400 ms) responses. The figure shows the average percentages from 100 participants over 300 attempts (30 attempts per bin). Participants improved with practice, as indicated by the percentage of correct responses reaching almost 70% accuracy by the end. More notable is the error pattern, with participants making too-early responses more often than too-late responses.

This too-early bias deviates from the roughly symmetrical responses observed in time interval estimation studies (e.g. Rakitin, et al., 1998). We suspected two factors might be responsible for the too-early bias. The first possibility is that estimating a shorter time interval contaminated performance in the target interval. In the Space Fortress task, the fortress task involves shooting a fast double-shot (<250 ms interval). Studies (Grondin, 2005; Jones & Wearden, 2004; Taatgen & Van Rijn, 2011) suggest that representations of different time intervals are not independent of each other. Participants in Taatgen & Van Rijn (2011) study alternated between producing a short interval and a long interval. When the feedback criterion for the long interval was shifted unbeknownst to the participants, not only did the estimate of the long interval change, but the estimate of the short interval also changed. Thus, estimating the shorter interval for the fortress task might have influenced estimating the target interval for the mine task.

A second possibility is that participants might be more likely to commit too-early errors as less time is allowed for the mine task. Note that the mine task consists of multiple demanding activities that are in competition with each other for the limited length of time available for the mine task. One might hypothesize that participants adjust the IFF interval based on their estimation of time remaining to fire a missile before the mine crashes into the ship.

The IFF Tapping Experiment

We tested those two hypotheses in a within-subjects design by manipulating 1) the speed of tapping (fast/slow) alternated with the IFF (intermediate) tapping, and 2) the distance between ship and mine (short/long) at mine onset. We created three types of games: fast-tap, slow-tap, and intermediate-tap-only games. Those games were a simplified version of the original Pygame Space Fortress task (Destefano, 2010).

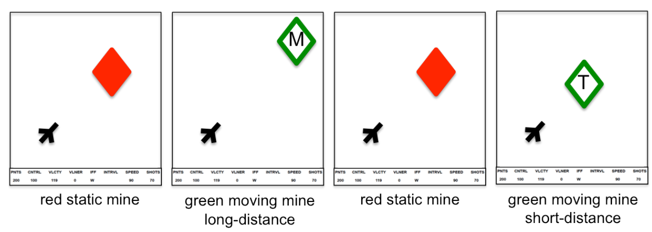

Figure 2: Sample sequence of trials in the fast-tap game.

Figure 2 shows a sample sequence of trials in the fast-tap game. The game had a static ship fixed at the bottom left of the screen always correctly aimed toward the mine that appeared from the other side. During the game, 8 red static and 8 green moving mines appeared in a strictly alternating order. For a red static mine, participants simply had to produce a fast (<250 ms) double-tap (spacebar). In the following trial, a green mine containing a letter appeared and approached the ship. For the green moving mine, participants had to 1) check the letter and determine its identity, 2) produce the IFF interval using an appropriate key (F key for friend and J key for foe), and 3) fire a missile (space bar). If all steps were successfully completed, the mine was destroyed. If any of the three steps were missed or performed incorrectly, the mine became invincible and eventually destroyed the ship. The slow-tap games were identical to the fast-tap games except that they had blue static mines (instead of red static mines) for which participants produced a slow (400-650 ms) double-tap. The distance manipulation was applied to the green moving mines in the fast-tap and slow-tap games. The distance between ship and mine at the moment of mine onset was randomized to be either short (283 pixels, corresponding to 1.86 s until mine collision) or long (566 pixels, 3.73 s). The intermediate-tap-only games were intended to test whether the too-early bias would still be present when participants produced the target interval without the demands of the mine task and without estimating different time intervals. In each trial, when a letter (either F or J) appeared in the center of the screen, participants simply produced the IFF interval using the corresponding key. Each intermediate-tap-only game had eight trials.

Twenty participants (5 males, mean age: 19 yrs) from Carnegie Mellon University participated in the experiment. The experiment consisted of 12 blocks of games. Each block had one intermediate-tap-only game, one fast-tap game, and one slow-tap game in a randomized order.

Behavioral Results

Figure 3 (left) presents the IFF tapping performance in intermediate-tap-only games over 12 blocks. Participants overall performed very well (mean accuracy: 86%). Importantly, the too-early bias was not present confirming our prediction. Participants committed too-early and too-late errors with roughly equal frequencies in the first block, but they quickly reduced their too-early errors. Thus, there was a small too-late bias on later trials, which may reflect a floor effect on the shortest intervals participants could produce. Figure 3 (right) shows that the mean produced IFF interval fell within the targeted 250-400 ms range and did not fluctuate much over blocks.

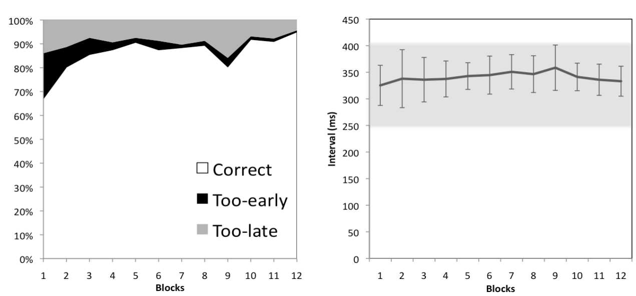

Figure 3: IFF tapping performance in intermediate-tap-only games: percentages of correct/too-early/too-late responses (left) and mean produced IFF intervals (right).

The results from the fast-tap and slow-tap games confirmed both the contamination and the distance hypotheses. Figure 4 displays the performance in the IFF tapping task in the four conditions defined by crossing the tap speed (fast/slow) and the distance (short/long) manipulations: fast-short, fast-long, slow-short, and slow-long. The percentage of correct responses increased over practice in all conditions. In all conditions the too-early responses dominated the first couple of blocks, but afterwards the bias stabilized at a lower level. The largest too-early bias was present in the fast-short condition, whereas the smallest too-early bias (and the largest too-late bias) was found in the slow-long condition. Note that the fast-short condition best reflects the original Space Fortress game in which participants handle both mines (IFF taps) and the fortress (fast-taps), and have only a short time for the mine task (short-distance).

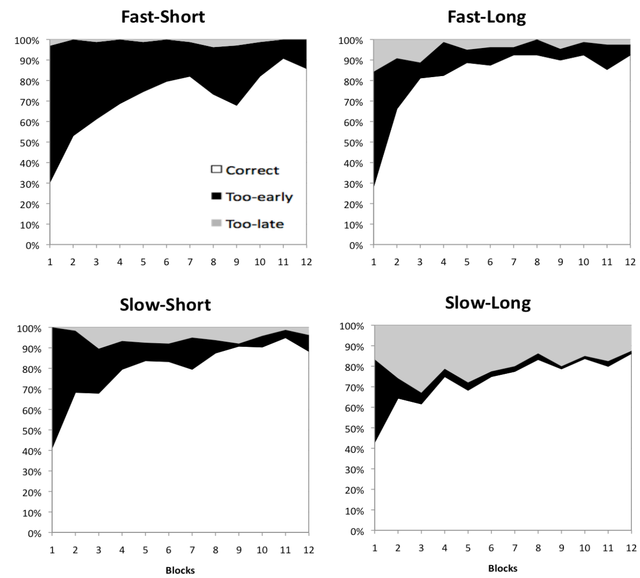

Figure 4: IFF tapping performance in fast/slow-tap games.

A repeated measures analysis of variance was performed with tap speed (fast/slow), distance (short/long), and practice (early: block 1-6 vs. late: block 7-12) as within-subjects factors and the mean produced IFF interval as the dependent measure. There were significant interactions between tap speed and distance ($F(1,19) = 10.23$, $p < 0.01$), tap speed and practice ($F(1,19) = 5.13$, $p < .05$), and distance and practice ($F(1,19) = 11.62$, $p < 0.01$). The interaction between tap speed and distance reflects the larger distance effect in the slow-tap condition compared with the fast-tap condition. The interactions between tap speed and practice, and distance and practice reflect that those effects were larger in earlier blocks than in later blocks. The three-way interaction was not significant.

The ACT-R Model

We developed a simulation model of our time estimation task, incorporating ideas from the Taatgen and Van Rijn (2011) timing model into a task model based on the ACT-R model for Space Fortress (Bothell, 2010). The model[1] was implemented in the ACT-R architecture (Anderson, Bothell, Byrne, Douglass, Lebiere, & Qin, 2004), which allows us to simulate all aspects of the task, not just the timing component. In ACT-R, time estimation is achieved through the processing in the temporal module (Taatgen, Van Rijn,

[1] Model parameters: :rt 1.0, :lf 1.1, :ans 0.385, :mp 2.25, :time-master-start-increment 0.011, :time-multi 1.1, :time-noise 0.0015.

& Anderson, 2007) and its interaction with the rest of the system. The temporal module, based on the internal clock model (Matell & Meck, 2000), assumes a pacemaker keeps accumulating pulses as time progresses. The production system can access the current pulse value through the temporal module's buffer and compare it with a criterion (e.g., a value retrieved from memory) to determine if the target interval has elapsed.

The model uses an instance-based approach to learn the required tapping times. When the model produces a time interval (e.g., 15 pulses) and observes its outcome (e.g., too-early), the specific instance of that experience is stored in declarative memory as a chunk. This chunk can be retrieved later to serve as a basis for deciding how long to wait the next time the model has to produce the interval. As such chunks are added to memory, the speed of retrieval increases and the accuracy of the retrieved result improves (similar to Logan's 1988 instance theory)

Figure 5 displays the series of steps in which the model performs the IFF tapping task. When a mine appears, the model attends the letter and determines its identity by retrieving the letter from memory. The model then starts retrieving a criterion value for the IFF interval. The retrieval of the criterion value is based on the blending mechanism discussed in the next section. If blending is successful, the model uses the blended result as the criterion. If blending fails, the model uses a default value. Once the criterion is determined, the model issues the first tap and starts incrementing the pulse value in the temporal buffer. When the current pulse value exceeds the criterion, the model issues the second tap that terminates the interval. The model then taps the spacebar to fire a missile. After completing both IFF tapping and the missile firing, the model attends the feedback[2], evaluates the outcome (too-early, correct, or too-late), and assigns a feedbackshift value (positive for too-early, zero for correct, and negative for too-late) so that the criterion could be appropriately adjusted in the next trial.

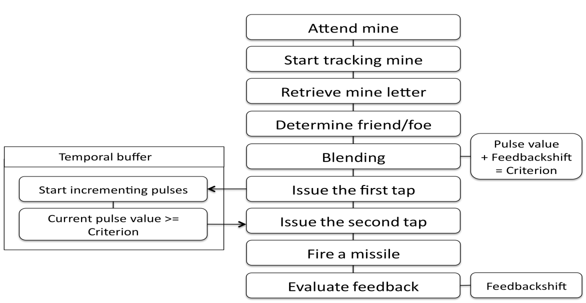

Figure 5: The ACT-R model of the IFF tapping task.

[2] According to our data, approximately 90% of the wrong IFF intervals were followed by a missile firing. We interpret this as indicating that participants tended to execute the entire sequence of key presses as a unit rather than interrupting the sequence after the IFF tapping to attend to feedback.

Blending

The ACT-R blending mechanism (Lebiere, Gonzalez, & Martin, 2007) was adopted to model the contamination from representations of different time intervals. In the standard retrieval mechanism of ACT-R, a retrieval request results in retrieval of a single chunk with the highest activation that exceeds the retrieval threshold. Blending is an alternative mechanism that allows retrieval of a weighted aggregation of all candidate chunks available in memory. Each candidate chunk is given a different weight based on how recently the chunk has been created and how closely it matches the retrieval request.

Figure 6 illustrates how contamination occurs during the blending in the fast-tap game in which the model alternates between the intermediate-tap and the fast-tap. When the blending request is made for pulse value (the value that was previously used for the 'intermediate-tap' and its outcome was 'correct'), blending is performed for candidate chunks that perfectly match the request (e.g., interval44 with 'intermediate-tap' type and 'correct' outcome) and imperfectly matching chunks (e.g., interval45 with 'fast-tap' type and 'too-late' outcome), with the latter penalized according to their degree of mismatch with the blending request[3]. Due to the contribution of fast-tap chunks (e.g. interval45), the final pulse value (15.551) is smaller than it is supposed to be had only intermediate-tap chunks contribute to blending. The model performed blending separately for pulse value and feedbackshift value, then used the sum of those two (15.982=15.661+0.321) as the criterion.

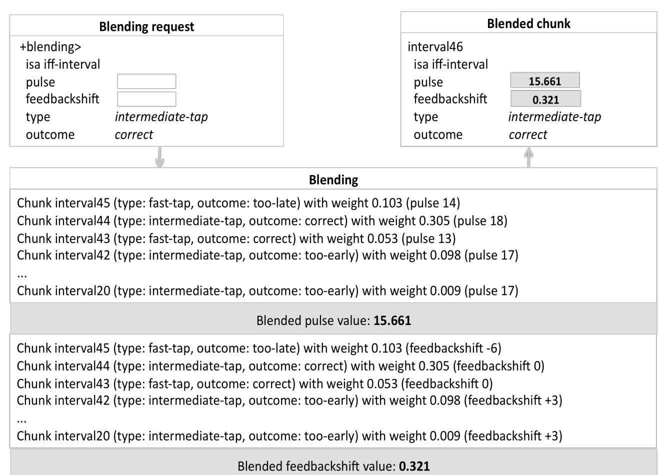

Figure 6: Blending for intermediate-tap.

Modeling the Distance effect

The model has a production rule that issues the second IFF tap when the current pulse value is greater than or equal to the criterion. We modeled the distance effect by adding an additional 'emergency' production for the second IFF tap.

[3] Partial matching was enabled for the tap type (fast/intermediate/slow) and outcome (too-early/correct/too-late).

During the trial, the model tracks the mine's trajectory by updating the visual-location buffer with the mine's current location. The emergency production specifies a threshold value in pixels that forces the model to issue the second tap such that it will have enough time remaining to fire a missile before it hits the ship. The model ignores the pulse value in the temporal buffer when this production fires.

Model Results

Contrary to human, we found that the model does not show the burst of too-early responses in early blocks. This is not surprising because the model starts out with a perfect representation of the task instructions, whereas participants have to work out any misunderstandings. Thus, participants show many more start up errors (e.g., no response). Since our goal is not to model this skill learning, we decided to focus on modeling the stable effects in the last 8 blocks, where participants and the model have both mastered the basic task requirements. Figure 7 offers comparisons of human and model performance in the last 8 blocks based on 100 model runs. The model successfully captures not only the lack of a too-early bias in the intermediate-tap-only condition, but also the distance and contamination effects in the other conditions. The overall correlation between model and participants equals .992.

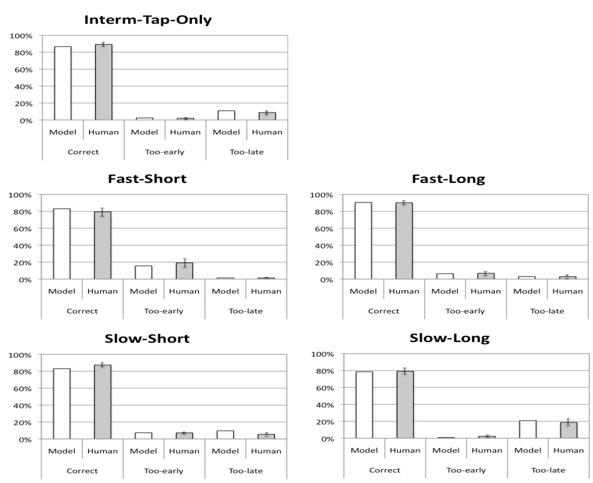

Figure 7: The model fit.

Discussion

Two factors appear to be responsible for the too-early bias in time estimation that occurs in the context of a dynamic task. First, the representation of the shorter or longer interval shifted the representation of the intermediate interval, supporting the claim (Taatgen & Van Rijn, 2011) that more than a single experience determines the representation of the target interval. The blending mechanism of ACT-R offers quantitative descriptions of interference among time interval representations in declarative memory. Second, the time remaining until the end of the task influenced time interval production. It shows that time estimation can be sensitive to one's knowledge of what is about to happen, consistent with animal literature (e.g., Church, Miller, Meek, and Gibbon, 1991). Our model captures this by having a procedural rule that overrides the outcome of the internal temporal estimate based on its perceptual processing of the environment.

Regardless of the conditions, participants showed a strong too-early bias in the early blocks (see Figure 4). There are a number of possible explanations for this result. First, participants were likely learning how to speed up other aspects of the task besides the blending process across blocks. In early blocks, these other processes might have been so slow as to increase use of the emergency rule. Second, participants might not even have been trying to time the target interval in the early blocks; instead they may have just practiced the sequence of responses in the task and focused on time estimation only when they had become proficient at responding. The third possible explanation is arousal. Studies (e.g., Penton-Voak, Edwards, Percival, & Wearden, 1996), suggest that arousal can affect the subjective duration of intervals by speeding up the rate at which a temporal pacemaker produces pulses.

The clear contrast between performance in the intermediate-tap-only condition and the other conditions demonstrates that time estimation performed in a dynamic task can exhibit properties different from when it is performed as an isolated task in static environments. External temporal or nontemporal tasks can influence production of the target time interval not only when those tasks are performed concurrently (Van Rijn & Taatgen, 2008) but also when performed in the same context in an alternating order (Taatgen & Van Rijn, 2011). Those results emphasize the virtue of modeling time estimation in the integrated cognitive architecture of ACT-R. The critical aspect of our model is not just the module's internal temporal processing, but also the contributions of the declarative and procedural components. This integrated approach of modeling time estimation in cognitive architecture can be especially useful in understanding time estimation in multitasking situations.

One possible application of our results concerns the training of time estimation tasks. In skill acquisition literature, two instructional strategies, part-task training (e.g., Frederiksen & White, 1989) and integrated training (e.g., Gopher, Weil, & Bareket, 1994), have been compared. The contrast between the intermediate-tap-only condition and the other conditions in our study suggests that a greater training emphasis can be directed to integrating timing with other subtasks (whole-task approach) rather than drilling on timing alone (part-task approach). Good performance in the intermediate-tap-only condition did not transfer to good intermediate timing in the more complex games.

Human factors researchers have studied timing performance and patterns of timing errors in dynamic multitasking situations (e.g., Rantanen & Xu, 2001). Similar to those studies, another potential application of our results regards to improving safety and reducing errors in time-critical multitasking situations (e.g., traffic environments). Identifying patterns of timing errors and investigating the underlying causes may suggest changes in work procedures. For instance, a timing-critical task can be separated from other tasks that involve less critical timing, putting it under lower time pressure.

The time estimation mechanism in ACT-R has successfully captured time estimation performance in dual-task conditions (Taatgen et al., 2007) as well as in dynamic multitasking situations (Salvucci, et al., 2006). We explored millisecond-level time estimation embedded in a complex real-time task that imposes especially high perceptual-motor demands. The model provided an integrated account of why time estimation performed in this context exhibited different properties than when it was performed in an isolated context. This study further supports the need to model time estimation in the broader context of cognition as we attempt to expand our understanding of human temporal cognition in the domain of complex skills.

Acknowledgments

This work was supported by ONR grant N00014-09-1-0402 to Wayne Gray & John Anderson.

References

Anderson, J. R., Bothell, D., Byrne, M., Douglass, D., Lebiere, C., & Qin, Y. (2004). An integrated theory of mind. *Psychological Review*, *111*, 1036–1060.

Bothell, D. (2010). *Modeling Space Fortress: CMU Effort* [PowerPoint slides]. Retrieved from http://act-r.psy.cmu.edu/workshops/workshop-2010/schedule.html

Church, R. M, Miller, K. D, Meek, W. H, & Gibbon, J. (1991). Symmetrical and asymmetrical sources of variance in temporal generalization. *Animal Learning and Behavior*, *19*, 207-214.

Destefano, M. (2010). *The mechanics of multitasking: The choreography of perception, action, and cognition over 7.05 orders of magnitude.* Unpublished doctoral dissertation, Rensselaer Polytechnic Institute, Troy, NY.

Donchin, E. (1989). The learning strategies project. *Acta Psychologica*, *71*, 1-15.

Fortin, C., Rousseau. R., Bourque. P., & Kirouac. E. (1993). Time estimation and concurrent nontemporal processing: Specific interference from short-term–memory demands. *Perception & Psychophysics*, *53*, 536-548.

Frederiksen, J. R., & White, B. Y. (1989). An approach to training based upon principled task decomposition. *Acta Psychologica, 71*, 89-146.

Gopher, D., Weil, M., & Bareket, T. (1994). Transfer of skill from a computer game trainer to flight. *Human Factors, 36*, 387-405.

Grondin, S. (2005). Overloading temporal memory. *Journal of Experimental Psychology: Human Perception and Performance*, *31*, 869-879.

Ivry, R., Spencer, R. M., Zelaznik, H. N., & Diedrichsen, J. (2002). The cerebellum and event timing. In S.M. Highstein and W.T. Thach (Eds.), The cerebellum: Recent developments in cerebellar research. *Annals of the New York Academy of Sciences*. New York: New York Academy of Sciences.

Jones, L. A., & Wearden, J. H. (2004). Double standards: Memory loading in temporal reference memory. *Quarterly Journal of Experimental Psychology, 57B*, 55-77.

Jones, M. R. (1990). Musical events and models of musical time. In R. Block (Ed.), *Cognitive models of psychological time*. Hillsdale, NJ: Lawrence Erlbaum.

Lebiere, C., Gonzalez, C. & Martin, M.K. (2007). Instance-based decision making model of repeated binary choice. *Proceedings of the 8th International Conference on Cognitive Modeling* (pp. 77-82). Ann Arbor, MI.

Logan, G. D. (1988). Toward an instance theory of automatization. *Psychological Review*, *95*, 492–527.

Matell, M. S., & Meck, W. H. (2000). Neuropsychological mechanisms of interval timing behavior. *BioEssays*, *22*, 94–103.

Penton-Voak. I. S., Edwards. H., Percival. A., Wearden. J. H. (1996). Speeding up an internal clock in humans? Effects of click trains on subjective duration. *Journal of Experimental Psychology: Animal Behavior Processes*, *22*, 307-320.

Rakitin, B. C., Gibbon, J., Penney, T. B., Malapani, C., Hinton, S. C., & Meck, W. H. (1998). Scalar expectancy theory and peak-interval timing in humans. *Journal of Experimental Psychology: Animal Behavior Processes, 24*, 15–33.

Rantanen, E. M. & Xu, X. (2001). Human performance in timing of discrete actions. *Proceedings of the 45th Annual Meeting of the Human Factors and Ergonomics* Society (pp. 527-531). Santa Monica, CA.

Salvucci, D., Taatgen, N., & Kushleyeva, Y. (2006). Learning when to switch tasks in a dynamic multitasking environment. *Proceedings of the Seventh International Conference on Cognitive Modeling* (pp. 268-273). Trieste, Italy.

Schirmer, A. (2004). Timing speech: A review of lesion and neuroimaging findings. *Cognitive Brain Research*, *21*, 269-287.

Sternberg, S. (1966). High-speed scanning in human memory. *Science*, *153(3736)*, 652-654.

Taatgen, N.A. & Van Rijn, H. (2011). Trace of times past: Representations of temporal intervals in memory. *Memory & Cognition*, *39*, 1546-1560.

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

Van Rijn, H., & Taatgen, N. A. (2008). Timing of multiple overlapping intervals: How many clocks do we have? *Acta Psychologica*, *129*, 365-375.

Zakay, D. (1990). The evasive art of subjective time measurement: Some methodological dilemmas. In R. A. Block (Ed.), Cognitive models of psychological time. Hillsdale, NJ: Erlbaum.