Cognitive Psychology 67 (2013) 26–54

Contents lists available at ScienceDirect

Cognitive Psychology

journal homepage: www.elsevier.com/locate/cogpsych

Timing in multitasking: Memory contamination and time pressure bias

Jungaa Moon *, John R. Anderson

Department of Psychology, Carnegie Mellon University, Pittsburgh, PA 15213, USA

ARTICLE INFO

Article history:
Accepted 29 June 2013
Available online 27 July 2013

Keywords:
Time estimation
Memory
Time pressure
Cognitive model
Multitasking

ABSTRACT

There can be systematic biases in time estimation when it is performed in complex multitasking situations. In this paper we focus on the mechanisms that cause participants to tend to respond too quickly and underestimate a target interval (250–400 ms) in a complex, real-time task. We hypothesized that two factors are responsible for the too-early bias: (1) Memory contamination from an even shorter time interval in the task, and (2) time pressure to take appropriate actions in time. In a simpler experiment that was focused on just these two factors, we found a strong too-early bias when participants estimated the target interval in alternation with a shorter interval and when they had little time to perform the task. The too-early bias was absent when they estimated the target interval in isolation without contamination and time pressure. A strong too-late bias occurred when the target interval alternated with a longer interval and there was no time pressure to respond. The effects were captured by incorporating the timing model of Taatgen and van Rijn (2011) into the ACT-R model for the Space Fortress task (Bothell, 2010). The results show that to properly understand time estimation in a dynamic task one needs to model the multiple influences that are occurring from the surrounding context.

1. Introduction

The ability to accurately estimate time intervals is important for organisms in making decisions and executing various actions. Time interval estimation enables animals to maximize net energy

* Corresponding author.
E-mail address: jungaam@andrew.cmu.edu (J. Moon).

0010-0285/$ - see front matter © 2013 Elsevier Inc. All rights reserved.
http://dx.doi.org/10.1016/j.cogpsych.2013.06.001

intake while minimizing time spent foraging (Bateson, 2003). Learning of time intervals between events and event rates has been one of the mechanisms proposed for conditioning of animals (Gallistel & Gibbon, 2000). Time interval estimation underlies skills such as motor control (Ivry, Spencer, Zelaznik, & Diedrichsen, 2002; Michon, 1967), musical performance (Jones, 1990), and speech processing (Schirmer, 2004). Millisecond-to-second interval timing is critical in performing real-time dynamic tasks that require rapid adaptive responses to a changing environment. For instance, when driving it is necessary to estimate how long one can attend to a GPS navigator before switching back to attending to the road and driving control (Salvucci, Taatgen, & Kushleyeva, 2006).

Time estimation has been studied in various paradigms (Zakay, 1990). Two major time estimation paradigms are a retrospective paradigm and a prospective paradigm. In the retrospective paradigm, participants are not aware of the need to judge duration until after the time duration has ended. In the prospective paradigm, participants are aware that a duration judgment has to be made while they experience the duration (Zakay & Block, 2004). In addition to producing a verbal estimate of a time interval or comparing two intervals, participants can also be asked to reproduce an interval. In a version of the reproduction method called the peak-interval (PI) procedure, participants attend to a target interval (e.g., a square that changes its color after the criterion duration), then later try to reproduce the same duration of time (e.g., by pressing a key when they expect the square to change its color). Studies with this procedure typically find that response distributions are (1) centered at the real time criteria, (2) roughly symmetrical,[1] and (3) standard deviations increase in proportion to the mean intervals (e.g., Rakitin et al., 1998).

In most studies of time estimation, time estimation is an isolated task performed in a static environment. It is the primary task on which the participants focus, even when a secondary task is given in order to discourage counting during time estimation (Rakitin et al., 1998) or to test the effect of the secondary task (e.g., Fortin, Rousseau, Bourque, & Kirouac, 1993). One standard procedure is to have participants prospectively estimate a time interval while doing a secondary nontemporal task (e.g., card-sorting task), then reproduce the interval without the secondary task. A common result is that the reproduction estimate decreases as a function of the demands of the secondary task (Block & Zakay, 1997; Brown, 1985; Zakay, Nitzan, & Glicksohn, 1983). The negative relationship between the reproduction estimate and the secondary task demand has been accounted for by the attentional-gate model (Zakay & Block, 1995, 1997). The model assumes that time estimation is a cognitive activity that competes for the limited attentional resources with other tasks (Zakay, 1990). As the secondary task demand increases, it will draw attentional resources away from time estimation resulting fewer time signals accumulated (i.e., decreased estimate).

One may wonder to what extent time estimation performed in the standard procedures reflects time estimation that people perform in everyday multitasking situations. In various multitasking situations, time estimation is often an implicit secondary task that one performs in order to flexibly coordinate primary tasks. Someone who tries to attend to a navigator while driving a car (Salvucci et al., 2006) may not even realize that they are estimating the time spent on those tasks. In addition, multitasking situations often require estimating two or more time intervals in the same context. For instance, a cook at a busy restaurant would need to keep an eye on everything on the grill so that things that need different lengths of cooking time can be flipped at the appropriate times. Studies showed that time estimation accuracy is lower when two overlapping time estimates are made compared to when only a single interval is estimated (Brown & West, 1990; van Rijn & Taatgen, 2008). Among the multiple factors that can influence time estimation in multitasking situations, the current study focused on the effects of memory contamination and time pressure.

Memory plays a major role in time estimation performance (e.g., Baudouin, Vanneste, Pouthas, & Isingrini, 2006; Brown, 1997; McCormack, Brown, Smith, & Brock, 2004). The need for a mechanism that stores temporal representations is obvious given that time estimation usually involves comparing time intervals at different points of time. According to the scalar timing model (Gibbon, Church, & Meck, 1984), time estimation is a function of pacemaker, memory, and decision processes. The pacemaker generates pulses (temporal signals) with certain intervals while

[1] See Church, Miller, Meck, and Gibbon (1991) for the discussion of small amount of asymmetry observed in the PI procedure.

one is estimating time. Those pulses are accumulated in working memory, which represents the experienced time duration. The count of the pulses in working memory is then stored in the long-term reference memory for later comparison. When one needs to reproduce a time interval experienced in the past as in the peak-interval (PI) procedure, the current count of the pulses accumulated in working memory is compared with the value sampled from reference memory. When the two values become close enough, a decision is made to terminate the reproduction of the interval.

Vierordt's law (Gu & Meck, 2011; Lejeune & Wearden, 2009) is one of the most robust phenomena in time perception. When multiple durations are experienced in the same context, participants tend to overestimate shorter intervals and underestimate longer intervals, moving their mean estimate towards "central tendency". It is as if memories of the two durations mix when they are experienced in the same context. This so-called "memory-mixing" effect has been shown in various time estimation paradigms. When participants reproduced time intervals that were drawn from different prior distributions, their estimates were systematically biased toward the mean of the prior distribution (Jazayeri & Shadlen, 2010). In a prospective time estimation paradigm in which participants were exposed to an induction stimulus context (sequence of brief tones) and subsequently presented with a standard interval followed by a comparison interval, temporal judgments (whether the comparison interval was shorter, equal to, or longer than the standard interval) were influenced by the distributional properties of local (the rate of the stimulus context) as well as global (the rates of the other sequences within a session) temporal contexts (Barnes & Jones, 2000; Jones & McAuley, 2005). Such results challenge models of time estimation because they deviate from the linear relationship between the subjective estimation and the real-time criterion (e.g., Rakitin et al., 1998). One candidate explanation is that individual representations of time intervals in reference memory are not independent from each other and create contamination when more than a single interval is estimated in the same context (Taatgen & van Rijn, 2011).

Time pressure, another factor common in multitasking situations, has been a popular topic of research in decision-making. Studies have focused on how time pressure affects cognitive strategies. According to Maule and Hockey (1993), if it is impossible to implement the preferred strategy due to a time constraint, people tend to adopt the best strategy given the time constraint. Zakay (1993)'s model provides an attention-based account for decision-making under time pressure. According to this model, a person engaged in decision-making under time pressure is essentially in the dual-task paradigm in which two concurrent cognitive tasks compete for the limited attentional resources. An increased awareness to the passage of time under time pressure increases attentional resources allocated to temporal information processing. Less attentional resources left for decision-making induces adoption of simpler strategies resulting in suboptimal decision-making performance. While the negative consequences of time pressure in decision-making seems clear, it is less clear how it will influence time estimation performed under time pressure. One might predict that time pressure will increase attention to temporal processing and thus will facilitate time estimation performance. On the other hand, if one focuses on the decisional aspect of time estimation as assumed in scalar timing model (Gibbon et al., 1984), one might predict that time pressure will negatively affect time estimation performance.

In summary, it seems plausible that time estimation in many real-world multitasking environments will exhibit properties not seen when time estimation is performed as an isolated task in a static environment. We decided to investigate the push and pull of factors affecting time estimation in a fast-paced task. We chose the Space Fortress task (Donchin, 1989), a computer-based video game that simulates real-time complex tasks performed in dynamic environments (e.g., piloting an aircraft). It has the advantage of a long history of study as well as some well-defined subtasks in which time estimation is critical. We found that dynamic multitasking environment could introduce strong biases into the time estimation process. After documenting the existence of such a bias in the Space Fortress task, we will describe an experiment that investigates the multiple origins of this bias. Finally, we will present a simulation model (based on ideas in Taatgen & van Rijn, 2011) that shows how the effects of those multiple factors can be integrated into a cognitive architecture.

2. Time interval estimation in the Space Fortress task

2.1. The IFF (Identify Friend or Foe) tapping task

The Space Fortress task (Fig. 1) was originally developed for the learning strategy program initiated by DARPA (Defense Advanced Research Projects Agency) to investigate the effectiveness of various learning strategies in complex tasks. The game requires coordination of cognitive, perceptual, and motor activities in real time. It involves navigating a ship in a frictionless space and firing missiles to destroy a central fortress and peripheral mines, all while simultaneously protecting the ship from the fortress and the mines. The goal of the game is to accumulate as many points as possible while performing those activities. The participant navigates the ship by rotating left or right (using the A and D keys, respectively) or thrusting (using the W key) to make the ship 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 (using the spacebar) at least ten times and then make a rapid double-shot to destroy it. One also has to monitor symbols regularly flashing underneath the fortress and collect bonuses when the "$" symbol appears twice in a row. The bonus collection task is similar to a 1-back task that requires judging whether an item matches the item one back in a sequentially presented list of items (McElree, 2001).

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 (referred to as "foe letters") and asked to remember them. During the game, a mine appears at a random location on the screen 5 s 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

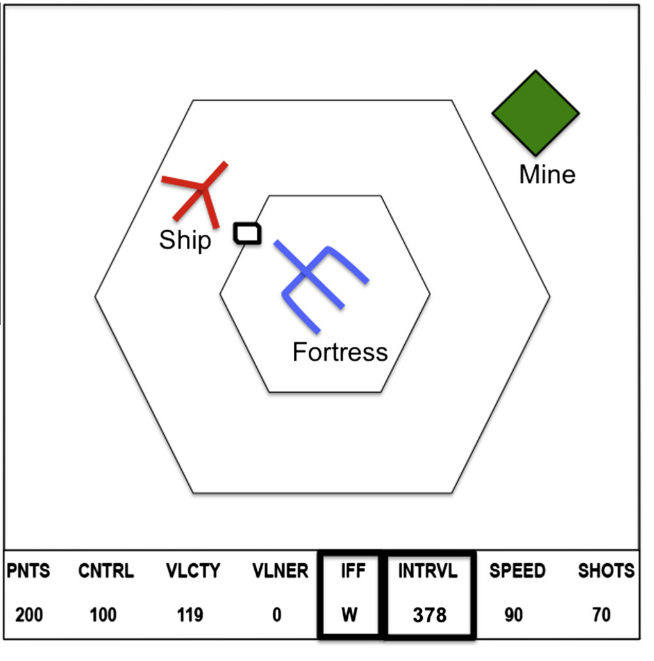


Fig. 1. A schematic representation of the screen for Space Fortress task. The participant has to navigate the ship (indicated in red) within the area enclosed by two hexagons while destroying the fortress (in blue) stationed in the center of the screen. The panel in the bottom of the screen displays critical information such as game scores. When a mine (in green) appears, a letter (e.g., W) associated with the mine is displayed in the IFF box (the first thick-lined box in the panel). Immediately after the participant produces an IFF interval, the produced interval (e.g., 378) is displayed in the INTRVL box (the second thick-lined box). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

letter that appears in the IFF box in the bottom panel of the screen (see Fig. 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 IFF tapping task, which involves tapping the J key twice with a 250–400 ms interval between the two key presses. Once an acceptable time interval has been generated, aiming the ship at the mine and firing a missile can destroy the mine. A missile can be fired even after a wrong IFF interval, but the missile will not destroy the mine. 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.

2.2. Too-early bias in the IFF tapping task

As a time interval estimation task, the IFF tapping task has three notable characteristics. First, it is a prospective time estimation task. Participants are told the target interval in written instructions ("you must identify it as a foe by pressing the IFF button twice at a moderate speed: 250 and 400 ms between each push"). During a game, participants have to produce the target interval whenever a foe mine appears. Immediately after each IFF tapping, the produced interval is displayed as feedback (e.g., "378") in the INTRVL box in the bottom panel (see Fig. 1). Presumably, participants start with a vague idea of the target interval, and then learn to produce the correct interval over the course of practice by receiving feedback and observing the outcome of their responses. Second, both the initiation and the termination of the interval are under the control of participants. Finally, and most importantly, the time estimation is performed not as an isolated task but rather 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 at the mine and firing a missile) the IFF tapping task, all of which need to be completed within a brief period of time, usually 2–3 s.

A study (unpublished work) previously conducted in our laboratory revealed an interesting pattern of performance in the IFF tapping task. Fig. 2 displays the percentage of responses within each of three categories: Correct responses (the produced interval was between 250 and 400 ms), too-early responses (the produced interval was shorter than 250 ms), and too-late responses (the produced interval was longer than 400 ms). It shows the average of 100 participants over 300 attempts (partitioned into 10 bins, with 30 attempts per bin). Trials with no response (neither the first nor the second key press was made) or single response (the second key press was not made) were excluded from the analysis. 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, which is especially strong at the beginning (representing approximately 55% of all responses), is maintained until the end despite a fair amount of improvement over practice.

This too-early bias deviates from the roughly symmetrical responses observed in time interval estimation studies (e.g., Rakitin et al., 1998). We suspected that two factors were responsible for the too-early bias. The first possibility is that estimating a shorter time interval contaminated estimation of the target interval. The IFF tapping task associated with mines occurs in the context of other tasks in the game, such as attempting to destroy the fortress. As mentioned earlier, after the fortress has been hit at least ten times, it can be destroyed with a rapid double-shot, which involves pressing the spacebar twice with an interval between key presses shorter than 250 ms. The results of several 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 and van Rijn's (2011) study learned two intervals (a short interval of 2 s and a long interval of 3.1 s) and repeatedly reproduced them in an alternating order. For each response, they received feedback (too-short, correct, or too-long) based on the accuracy of the reproduced interval. When the feedback criterion for the long interval was shifted unbeknownst to the participants (the feedback criterion for the short interval remained constant), not only did the estimate of the long interval shift toward the shifted feedback criterion, but the estimate of the short interval also changed in the same direction. Their

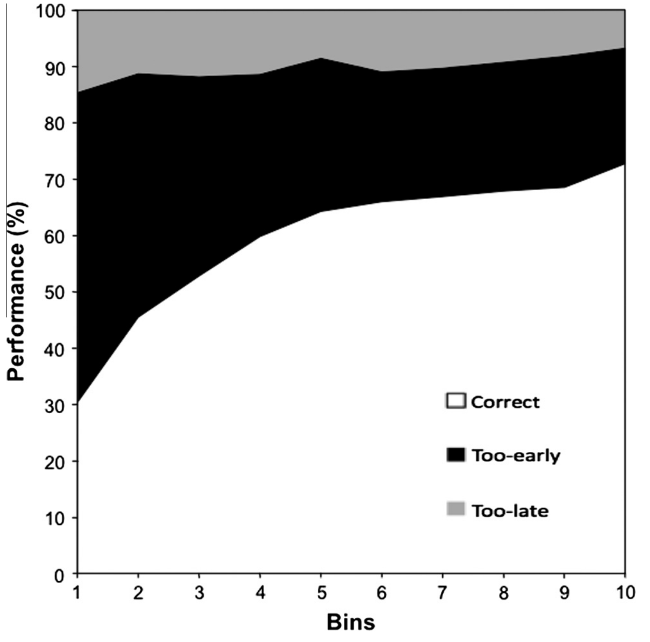


Fig. 2. Change in the percentages of IFF tapping response categories over 300 trials. *X*-axis: bins (each bin corresponds 30 trials of the IFF tapping task). *Y*-axis: percentages of response categories (correct, too-early, and too-late) in the IFF tapping task. The white area represents the correct (250–400 ms) responses, the black area represents too-early (<250 ms) responses, and the grey area represents too-late (>400 ms) responses.

results show that contamination among different time interval representations occur when they are estimated in the same context, similar to the memory-mixing effect (e.g., Gu & Meck, 2011; Jazayeri & Shadlen, 2010). Thus, estimating the shorter interval for the double-shot to destroy the fortress might have influenced estimating the target interval for the IFF tapping task.

A second possibility is that participants might be more likely to commit too-early errors when there is high time pressure (i.e., little time to avoid a collision as the mine gets closer to the ship). Note that the mine task consists of multiple demanding activities that have to be completed within a short time period. Thus, those activities are in competition with each other for the limited time available to perform the mine task. One might hypothesize that participants cope with the time constraint by adjusting the length of the IFF interval based on their estimation of how much time is left for the mine task. The need for this adjustment seems clear given that after producing the IFF interval one still needs to fire a missile to destroy the mine. The consequence of shortening the IFF interval length will be positive as long as the produced interval meets the target range since it affords more time for firing a missile. However, shortening the IFF interval length also increases the chances of being too early. If this hypothesis were true, one would expect that participants tend to prematurely terminate the IFF interval (i.e., too-early bias) as less time is allowed for the mine task. The decrease in the too-early bias with practice (see Fig. 2) may reflect participants learning to perform other aspects of the task more rapidly, thereby allowing more time for generating the IFF interval.

Both of these possibilities are not just true of Space Fortress but reflect factors that complicate timing in many real-world tasks. People may experience multiple timing patterns and one pattern might contaminate another. People are also often under pressure to get everything done within a fixed amount of time and this might bias timing. While the Space Fortress task is representative of the general issues of timing in real-world multitasking situations, there are so many aspects to it that it is hard to be sure whether both of these factors (memory contamination and time pressure) are responsible for the bias in time estimation, or just one, or perhaps some other factor. Therefore, we stripped down the original task into a simpler experimental paradigm that offered greater control but still allowed us to assess the degree to which one timing task was contaminating the other and the degree to which timing was being distorted by time pressure.

3. The IFF tapping experiment

3.1. Method

3.1.1. Participants

Twenty participants (5 males, mean age: 19 yrs) from Carnegie Mellon University participated for course credit and bonus money (Mean: $3.5, Standard Deviation: $1.05) based on their performance. Informed consent approved by the Carnegie Mellon University Institutional Review Board (IRB) was obtained from each participant.

3.1.2. Apparatus

Participants performed the task using a computer keyboard for input while attending stimuli displayed on a 17-in. monitor. The task was a simplified version of Pygame Space Fortress (Destefano, 2010) written in the Python programming language.

3.1.3. Design

We tested the two variables of interest (memory contamination and time pressure) in a within-subjects design by manipulating (1) the speed of tapping (fast/slow) that alternated with the IFF (intermediate) tapping, and (2) the distance (short/long) between ship and mine at mine onset. We created three types of games: fast-tap, slow-tap, and intermediate-tap-only games. These games were based on a simplified version of the original Space Fortress task but they still had the components essential for testing our hypotheses in the context of a dynamic task. For example, navigation and aiming tasks were eliminated in order to simplify the response requirements for the mine task. In the fast-tap and slow-tap games, a static ship was fixed at the bottom left of the screen always correctly aimed toward the mine that appeared from the other side. The bonus collection task and the fortress task were also eliminated in these games.

Fig. 3a shows a sample sequence of trials in the fast-tap game. During the game, 8 red static mines 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 (using the spacebar). This red static mine trial requires a time interval equal to the double-shot interval required to destroy the fortress in the original Space Fortress game. In the following trial, a green mine containing a letter appeared and approached the ship.[2] For the green moving mine, participants had to (1) check the letter and determine its identity, (2) produce the IFF interval (250–400 ms) using an appropriate key (F key for friend and J key for foe), and (3) fire a missile (by pressing the spacebar). If all three steps were successfully completed, the mine was destroyed (immediately disappeared from the screen) and the trial ended. If any of the three steps were missed or performed incorrectly (e.g., a friend was identified as a foe, or a wrong interval was produced), the mine became invincible (a missile could not destroy the mine) and eventually destroyed the ship. The slow-tap games (see Fig. 3b) were identical to the fast-tap games except that they had blue static mines instead of red static mines. Participants produced a slow (400–650 ms) double-tap (using the spacebar) for the blue static mines. The distance manipulation was applied to the green moving mines in the fast-tap and slow-tap games. The distance between the ship and a 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, 3.72 s). In each game, four of the green moving mines appeared from the short distance and four from the long distance.

[2] In the original Space Fortress game, the letter associated with the mine appears in the bottom panel (see Fig. 1). Given that navigation and aiming were eliminated in our simplified games, there was a possibility that participants would ignore the main part of the screen and respond by attending to the bottom panel only. To prevent this from happening, the letter was placed inside the mine so that participants had to process the main part of the screen.

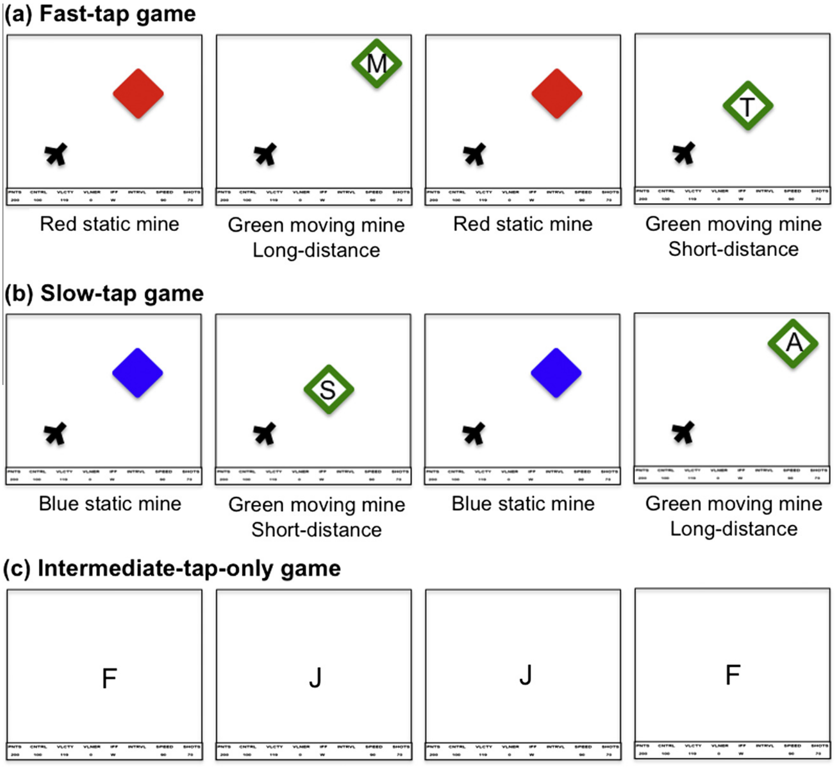


Fig. 3. Sample sequence of trials in (a) fast-tap, (b) slow-tap, and (c) intermediate-tap-only games. In (a) and (b), a mine (diamond) appears at the top right and a ship is located at the bottom left of the screen.

The intermediate-tap-only games (Fig. 3c) were intended to test whether the too-early bias would still be present when participants produced the target interval of 250–400 ms without the demands of the mine task and without estimating different time intervals. These games were very simple. In each trial, when a letter (either F or J) appeared in the center of the screen, participants produced the IFF interval (250–400 ms) using the corresponding key. If the produced interval was correct, the letter disappeared from the screen immediately. If not, the letter remained for another couple of seconds until the next trial started. Each intermediate-tap-only game had eight trials.

The experiment consisted of 12 blocks. Each block had one intermediate-tap-only game, one fast-tap game, and one slow-tap game. The order of games was randomized within each block. Participants read instructions (Appendix A) at the beginning of the experiment and had access to the instructions during the experiment. At the end of each game, participants were presented with a summary of their performance in the game (e.g., number of correct responses) and the bonus money they earned based on their performance in the tapping tasks. Each participant took approximately 50 min to finish the 36 games (3 games * 12 blocks).

3.2. Results

3.2.1. Performance on the IFF-taps

Fig. 4 (left) presents the IFF tapping performance in intermediate-tap-only games over 12 blocks. Compared to performance in the original Space fortress task (see Fig. 2), participants performed fairly well (mean accuracy: 86%). The percentage of correct responses does not show much fluctuation except for the improvement in the initial blocks. Importantly, the too-early bias is not present,

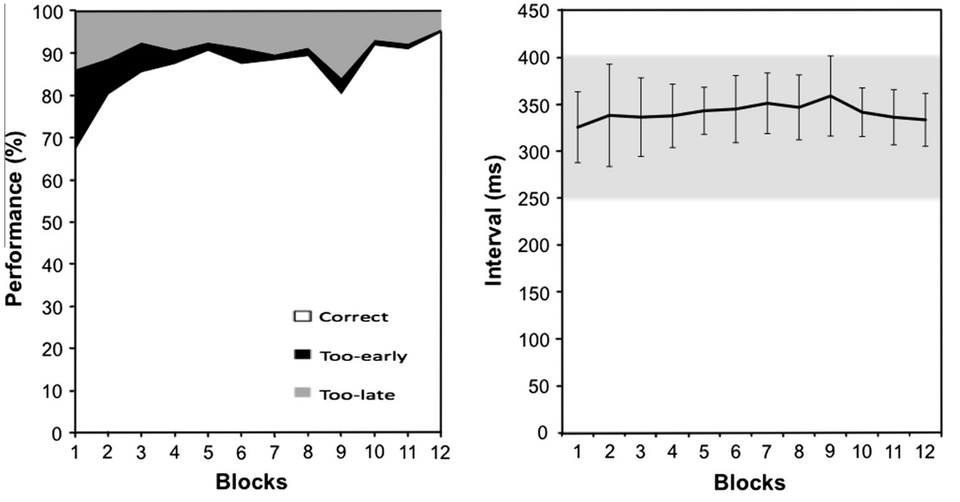


Fig. 4. IFF tapping performance of participants in intermediate-tap-only games over 12 blocks: Percentages of response categories (left) and mean produced IFF intervals (right). In the right figure, error bars represent the standard deviations, and the grey area represents the target interval range (250–400 ms).

confirming our prediction that the bias would be absent when time estimation was the only task participants had to perform. 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. Given that this was a quite brief interval pushing the lower bounds of timing, this bias may just reflect a floor effect of the shortest intervals participants could produce. Fig. 4 (right) shows that the mean produced IFF interval falls within the targeted 250–400 ms range and does not fluctuate much over blocks.

The results from the fast-tap and slow-tap games confirmed both the contamination and the distance hypotheses. Fig. 5 displays the performance in the IFF tapping task in the four conditions defined by crossing the fast/slow tap speed and the short/long distance manipulations: fast-short, fast-long, slow-short, and slow-long. The response percentages were calculated by dividing the number of responses in each response category (correct, too-early, and too-late) by the total number of attempts across all categories, discarding trials with no responses or incomplete responses (i.e., single taps). 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 slow-long condition showed a too-late bias. Note that the fast-short condition best reflects the original Space Fortress game in which participants handle both mines (requiring IFF-taps) and the fortress (requiring fast-taps), and have only a short time for the mine task (short-distance).

Fig. 6 (top) plots the mean produced IFF intervals averaged across participants in different conditions. Except for the intermediate-tap-only condition, which is overall flat over blocks, all other conditions share a similar trend: the mean intervals increase in the first three or four blocks, then remain stable thereafter. The fast-short condition produces the fastest tapping, indicating the strong too-early bias while the slow-long condition produces the slowest tapping. The fast-long and slow-short conditions, each with only one source of the too-early bias (contamination or distance), are similar to each other and between the fast-short and the slow-long conditions. The mean intervals in the last 8 blocks were 344 ms for intermediate-tap-only (SD: 20 ms), 294 ms for fast-short (SD: 26 ms), 320 ms for fast-long (SD: 23 ms), 324 ms for slow-short (SD: 21 ms), and 365 ms for slow-long (SD: 23 ms). Fig 6 (bottom) shows the distributions of the produced IFF intervals from the last 8 blocks. The fast-short curve is on the far left, the slow-long curve is on the far right, and the three curves with

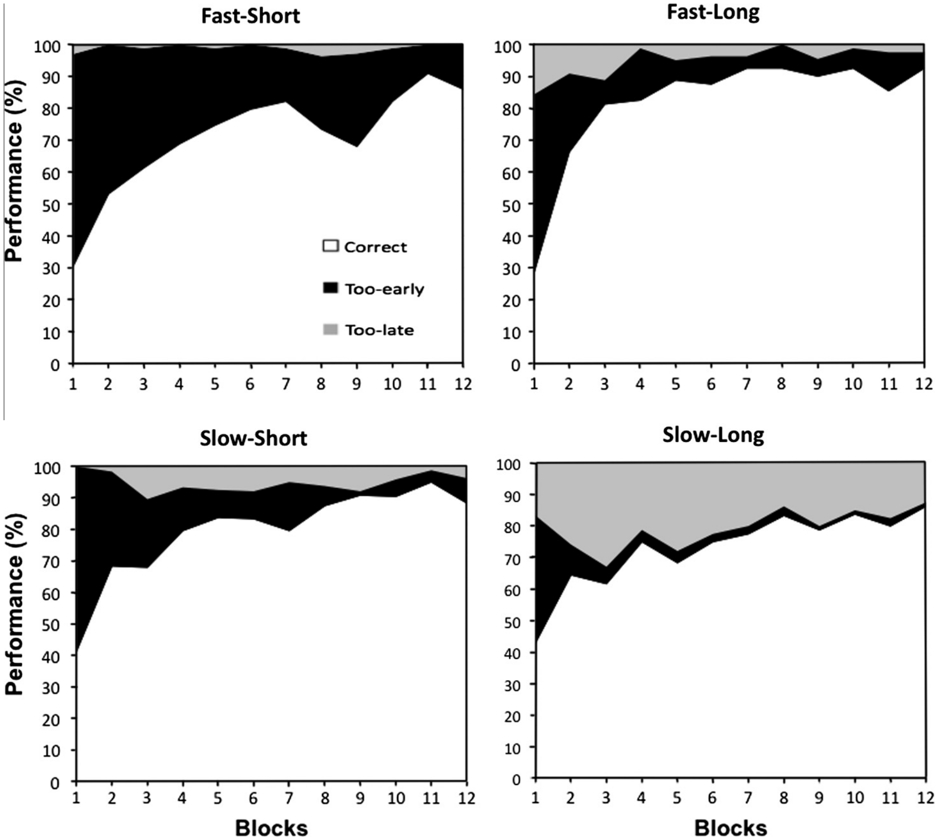


Fig. 5. Change in the percentages of IFF tapping response categories in fast/slow-tap games as a function of distance (short/long).

fewer sources of bias are sitting in the middle. The curves are similar to one another, but shifted according to the tap speed and distance effects.

A repeated measures analysis of variance was performed with tap speed (fast/slow) and distance (short/long) as within-subjects factors and the mean produced IFF interval in the last 8 blocks as the dependent measure. Intervals longer than 1000 ms and the blocks with mean intervals 3 inter-quartile-ranges below the first quartile or above the third quartile were excluded from analysis. These criteria resulted in the exclusion of 0.78% of data. The effects of tap speed ($F(1,19) = 72.22$, $p < .001$, $\eta_\rho^2 = .792$) and distance ($F(1,19) = 42.53$, $p < .001$, $\eta_\rho^2 = .691$) were both significant. The interaction between tap speed and distance was significant ($F(1,19) = 18.45$, $p < 0.01$, $\eta_\rho^2 = .493$). The interaction reflects the fact that the slow-long condition is even slower than that would be predicted by the main effects of tap speed and distance, as shown in Fig. 6 (top). In summary, the results support our hypotheses regarding the too-early bias. Estimating a shorter time interval contaminated the estimates of the intermediate IFF interval, making them shorter. A shorter distance between the mine and the ship (high time pressure) was another source of the too-early bias.

3.2.2. Performance on the fast-taps and slow-taps

Participants showed fairly good performance in producing fast-taps (<250 ms) and slow-taps (400–650 ms). The mean percentage of correct responses on fast-taps and slow-taps in the last 8 blocks was 87.7%. The high performance level indicates that participants followed the instructions,

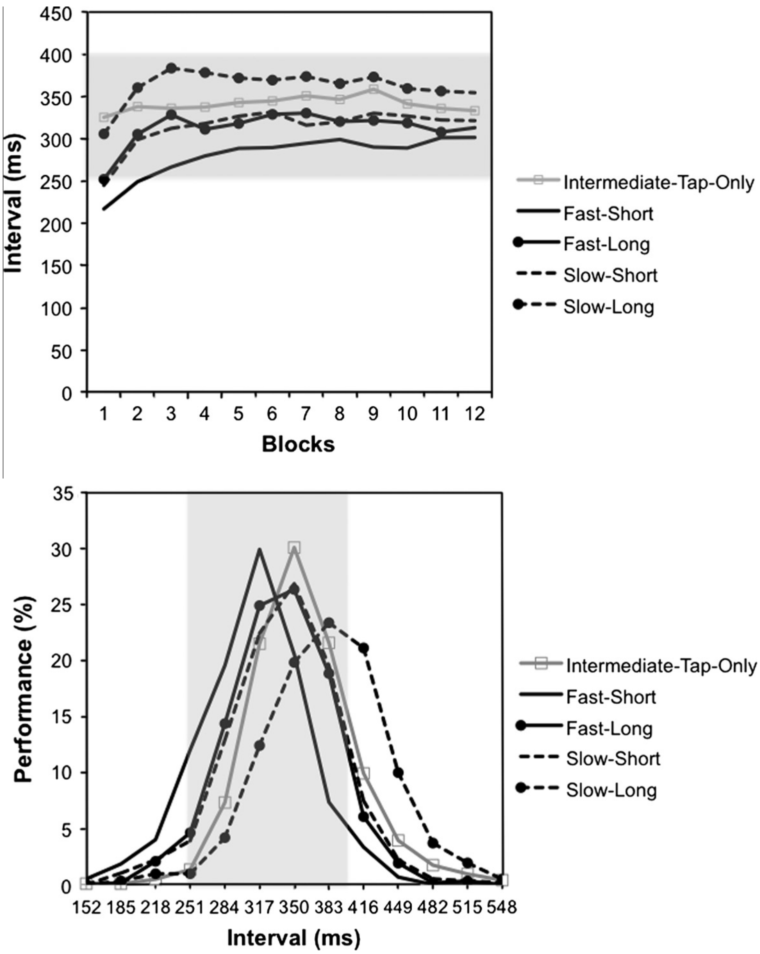


Fig. 6. Mean produced IFF intervals over 12 blocks (top) and distributions of the produced IFF intervals in the last 8 blocks (bottom).

and that these time interval ranges had moderate levels of difficulty. Participants performed significantly better on fast-taps (93.5%) than on slow-taps (81.9%): $t(19) = 5.52$, $p < .001$. This difference presumably reflects the different requirements for fast-taps and slow-taps. That is, whereas participants only had to meet an upper limit for fast-taps (250 ms), they had to meet both lower and upper limits for slow-taps (400–650 ms). One can simply execute two taps as quickly as possible for fast-taps, but one cannot use any such a simple strategy for slow-taps. For slow-taps, too-early errors (<400 ms) were significantly more frequent than too-late errors (>650 ms), $\text{Mean}_{\text{too-early}} = 11.9\%$, $\text{Mean}_{\text{too-late}} = 5.6\%$, $t(19) = 3.43$, $p < .01$. This result reinforces the contamination hypothesis. In the context in which participants alternated between the intermediate-tap and the slow-tap, the former contaminated the latter producing more frequent too-early errors.

3.2.3. Intervals associated with the IFF task

To investigate how the distance manipulation influenced the way participants executed the IFF tapping we analyzed the relationship between three successive intervals: (1) pre-IFF interval: the time between mine onset and the first IFF key press, (2) IFF interval: the time between the first IFF key press

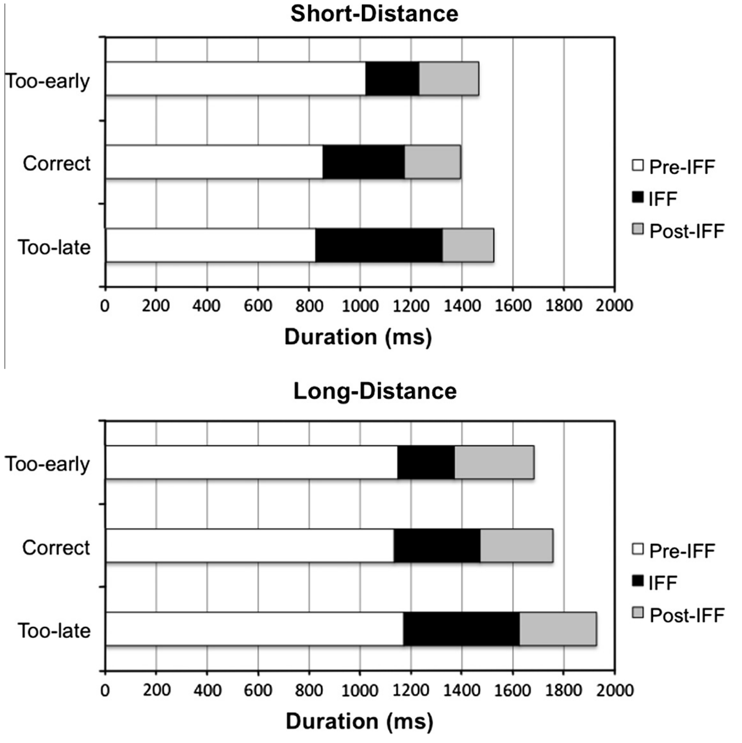


Fig. 7. Durations of time allocated to the pre-IFF, IFF and post-IFF activities under each response category (too-early, correct, and too-late) in the short-distance condition (top) and the long-distance condition (bottom) in the last 8 blocks.

and the second IFF key press, and (3) post-IFF interval: the time between the second IFF key press and the firing of a missile (spacebar key press). Fig. 7 displays the mean time allocated to the pre-IFF, IFF and post-IFF activities under each response category (too-early, correct, and too-late) in the short-distance condition (averaged across fast-short and slow-short conditions) and the long-distance condition (averaged across fast-long and slow-long conditions) in the last 8 blocks.[3] Perhaps not surprising, participants took 300 ms longer in the long-distance conditions reflecting the greater time they had to respond. There were no significant differences within the long-distance condition in terms of the time in the pre-IFF or post-IFF intervals. However, an interesting pattern did appear in the short-distance condition where participants were under much more time pressure. When participants were too early with their IFF, they averaged over 180 ms longer in their pre-IFF interval. All 18 of the participants who produced too-early responses in the short-distance condition had average pre-IFF interval times that were longer on these too-early trials than on correct trials, $t(17) = 7.58$, $p < .001$. In contrast, when participants were too long with their IFF, their pre-IFF interval was 40 ms shorter. There were only 9 participants who had too-long IFFs in the short-distance condition and this effect was not significant, $t(8) = 1.63$, $p = .142$. There were no significant differences involving the post-IFF interval. In summary, there was a strong tendency for participants to make too-early responses when their pre-IFF interval was long and they were in the short-distance condition.

The results suggest that when participants were under a tight time constraint, they adjusted the IFF interval based on how much time had been spent on pre-IFF activities and how much time was left for firing a missile. The pre-IFF interval is subject to factors such as how well one remembers the foe

[3] The total time allowed for the mine task was 1860 ms for the short-distance condition and 3720 ms for the long-distance condition.

letters and how quickly one determines the appropriate key for the IFF tapping. Spending too much time on the pre-IFF activities leaves little time for the IFF and post-IFF activities. This can result in a shortening of the IFF interval to avoid having too little time to execute a key press (to fire a missile at the mine) after completing the IFF tapping. On the other hand, when participants had a plenty of time for the mine task, the duration of the pre-IFF activities was not a critical determinant of the IFF interval.

Based on these analyses, one might further expect that even within the same distance condition, one's level of practice can influence the degree of time pressure. In the earlier phase of learning, performing the pre-IFF activities will take longer, which will induce higher time pressure and a stronger tendency to shorten the IFF interval. With more practice, speedup of the pre-IFF activities will induce lower time pressure, which will weaken the too-early bias. We found a shortening of the pre-IFF interval with practice. The mean pre-IFF interval was 1089 ms in earlier blocks (block 1–6) and 973 ms in later blocks (block 7–12). The shortening of the pre-IFF interval explains why the too-early bias weakened over practice (Fig. 5) and why the mean intervals increased in the first three or four blocks (Fig. 6, top). Changes in the pre-IFF and IFF intervals in the fast-short condition, a condition with high time pressure, demonstrate this point. Compared with the earlier blocks, the mean pre-IFF interval became shorter by 54 ms in the later blocks ($\text{Mean}_{\text{early}} = 894$ ms, $\text{Mean}_{\text{late}} = 840$ ms, $t(19) = 2.73$, $p < .05$). In conjunction with this change, the IFF interval became longer by 30 ms ($\text{Mean}_{\text{early}} = 265$ ms, $\text{Mean}_{\text{late}} = 296$ ms, $t(19) = -6.02$, $p < .001$). In contrast, such dynamics between the pre-IFF and IFF intervals over practice were absent in the intermediate-tap-only games. Neither the pre-IFF interval ($\text{Mean}_{\text{early}} = 625$ ms, $\text{Mean}_{\text{late}} = 640$ ms, $t(19) = -.51$, $p = .616$) nor the IFF interval ($\text{Mean}_{\text{early}} = 337$ ms, $\text{Mean}_{\text{late}} = 344$ ms, $t(19) = -1.34$, $p = .195$) changed significantly with practice.

3.2.4. Individual differences in timing accuracy

We wondered whether there were individual differences in the degree to which participants allowed one interval to contaminate another and what effect this would have on their overall accuracy. We computed the difference between the mean IFF interval produced in the slow-tap condition and the mean in the fast-tap condition. This measure reflects the degree of contamination in the intermediate interval representation under different tapping conditions, with a smaller value indicating a more robust representation of the intermediate interval. We also computed the performance accuracy averaged across intermediate as well as fast/slow tapping trials in the last 8 blocks. A significantly negative correlation was found between the average accuracy on these trials and the degree to which the intermediate tap was contaminated ($r = -.593$, $p < .01$). The interpretation is that the ability to minimize the contamination from different timing intervals partly explains better performance in the time estimation task.

4. An ACT-R model of the time estimation task

4.1. Time estimation in ACT-R

As a way of articulating and testing our understanding of the results, we investigated whether we could model our data by incorporating ideas from the Taatgen and van Rijn (2011) timing model into a task model based on the ACT-R (Adaptive Control of Thought – Rational) model for Space Fortress (Bothell, 2010). The model[4] was implemented in the ACT-R architecture (Anderson et al., 2004), an integrated theory in which activities of multiple modules are coordinated by a production system to produce coherent cognition. Modeling in ACT-R allowed 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, & Anderson, 2007) and its interaction with the rest of the system. In the temporal module, which is based on the internal clock model (Matell & Meck, 2000), a pacemaker keeps incrementing pulses once a start signal is given. Those pulses are accumulated in temporal buffer. A request made to the temporal buffer can reset the pulse count to 0, put a chunk (element of declarative

[4] The model is available at the models page of the ACT-R website (act-r.psy.cmu.edu/models/) under the title of this paper.

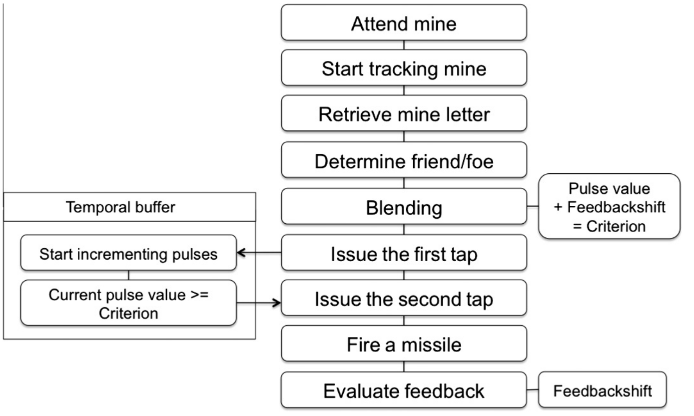


Fig. 8. The ACT-R model of the IFF tapping task.

knowledge in ACT-R) into the temporal buffer holding the count of 0, and start incrementing that pulse count. The pulse value stored in the temporal buffer corresponds to the estimated time interval. More details of the temporal module can be found in Appendix B. In addition to this temporal module, other relevant components of the ACT-R architecture include a visual module for reading the characters and tracking the mine, a motor module for issuing the taps, a retrieval model for keeping track of foe letters and different time lengths, and goal and procedural modules for coordinating these activities.

The procedural module is implemented as a production system that 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. Even if the temporal module has accurate timing, performance may not be perfect because the production system may be preoccupied with its interactions with other modules. For instance, the production system may fail to read the estimated pulse count while it is engaged in other tasks.

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).

4.2. Model description

4.2.1. IFF interval estimation

The flowchart in Fig. 8 displays the series of steps in which the model performs the IFF tapping task. When a mine appears, the model attends to the letter and starts tracking the mine as it moves. The model determines the mine's identity by retrieving the letter from memory, and decides which key is appropriate for producing the IFF interval (F key for a friend mine, J key for a foe mine). The model then starts retrieving a criterion pulse value for the IFF interval. The retrieval of the criterion value is based on the blending mechanism discussed later and in Appendix B. If blending is successful, the model uses the blended result as the criterion. If blending fails, the model uses a default value (set to 16 pulses). Once the criterion is determined, the model issues the first IFF tap and starts incrementing the pulse value in the temporal buffer. When the pulse value (a positive integer) in the temporal buffer is greater than or equal to the criterion value (a positive real number), the model issues the second IFF tap that terminates the interval. As soon as the manual module is free, the model issues another tap (spacebar) to fire a missile. After firing a missile, the model attends to the feedback and evaluates the outcome as too-early, correct, or too-late. Finally, the model uses the feedback to adjust the criterion by storing a feedbackshift value (positive for too-early responses, zero for correct responses, and negative for too-late responses) that is added to the pulse value so that the criterion can be appropriately adjusted in the next trial.

Although a missile can destroy the mine only after the correct IFF interval has been produced, participants had a tendency to fire a missile even after producing a wrong IFF interval. According to our data from 12 blocks, approximately 90% of the wrong IFF intervals were followed by a missile firing. We interpret this result as indicating that participants tended to execute the entire sequence of key presses (two IFF key presses followed by the spacebar key press) as a unit rather than interrupting the sequence after the IFF tapping to attend to feedback. After each IFF tapping sequence, the feedback remained in the bottom panel of the screen until the end of the trial, which makes this strategy possible. Thus, the model only attends to the feedback after firing a missile.

4.2.2. Fast/slow interval estimation

Besides the intermediate (IFF) tapping for moving mines, the model also performs the fast-taps and slow-taps for static mines. We built the model to adopt different strategies for the fast-taps and slow-taps. We assumed that memory retrieval was not strongly involved in performing the fast-taps. Considering there is only an upper criterion for the fast-taps (250 ms), one can adopt a strategy of executing two key presses as fast as possible instead of retrieving a time interval from memory. Thus, the model executes the first tap without retrieving an interval from memory, and then executes the second tap as soon as the motor module is ready. We found that some of the participants were consistently faster than others in the fast-tap trials. To accommodate such individual differences, the model had a conservative and a liberal strategy for executing the second tap. The conservative strategy involved initiating preparation of the second tap only when the manual module is entirely free (i.e., the first tap has been initiated). The liberal strategy was to initiate preparation of the second tap when the first tap has been prepared and it just has to be initiated. The conservative strategy produces a 200-ms interval while the liberal strategy produces a 150-ms interval for the fast-taps. We mixed those two strategies so that each strategy was used in half of the total model runs.

For the slow-taps (400–650 ms), we assumed that memory retrieval is involved because one cannot use the straightforward execution strategy afforded by the fast-taps. Slow tapping is executed similar to intermediate tapping. The model starts retrieving a blended time interval before it executes the first tap, using the retrieved interval as the criterion value. If blending fails, a randomly selected integer between 18 and 20 is used as the criterion pulse value. When the current pulse value of the temporal buffer exceeds the criterion value, the model executes the second tap. As with the IFF tapping, feedback is displayed after each instance of slow tapping. The model evaluates the feedback and assigns an appropriate feedbackshift value.

4.2.3. The blending mechanism

The ACT-R blending mechanism (Lebiere, Gonzalez, & Martin, 2007) was adopted to model the contamination from representations of different time intervals. Rather than retrieving a specific chunk as in standard ACT-R retrieval, blending produces 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.

Fig. 9 illustrates blending for a trial in a fast-tap game in which the model alternates between intermediate-taps and fast-taps. For the example shown in the figure, the model is in an intermediate-tap trial in which it has to deal with a green moving mine (as in the second trial in Fig. 3a). The upper left box shows the request made to the blending module, which to find a pulse value that would be appropriate for executing the second IFF tap. The blending request has four slots (pulse, feedbackshift, type, and outcome). Pulse refers to the total count of pulses accumulated in the temporal buffer during a previous instance of interval estimation. Feedbackshift refers to the adjustment (positive/zero/negative value) made to the pulse value at the end of the trial based on the outcome of using that pulse value. Type refers to the tap speed, which can be fast, slow, or intermediate. Outcome refers to the evaluation the model made based on the timing feedback; it can be too-early, too-late, or correct. The blending request in Fig. 9 specifies type (intermediate-tap) and outcome (correct) but does not specify pulse or feedbackshift. Blending is performed to fill in those unspecified slots. The request

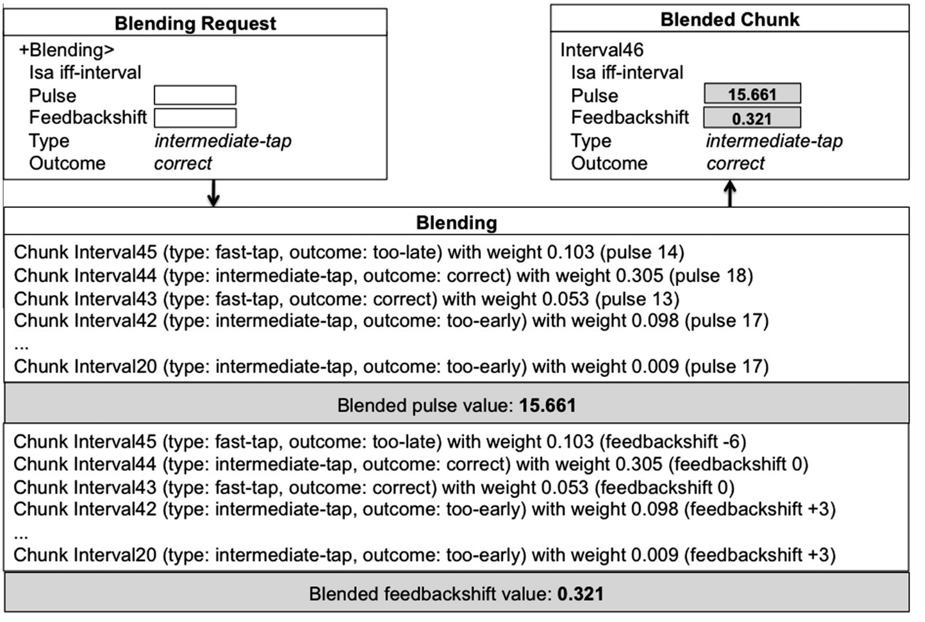


Fig. 9. An example of blending for IFF interval in a fast-tap game.

blends candidate chunks that perfectly match the request (e.g., chunks with correct intermediate-tap) and imperfectly matching chunks (e.g., chunks with fast/slow-tap type or too-early/too-late outcome), with the latter penalized according to their degree of mismatch with the blending request.

The bottom box in Fig. 9 shows an example of the computation of the blended values for the pulse and feedbackshift slots. The weight associated with each chunk represents the degree to which the chunk contributes to blending and is determined by its recency, the match with the request, and activation noise. For instance, consider the first four chunks represented in Fig. 9, which are from a block where fast-tap and intermediate-tap trials alternated:

- Interval45 mismatches both type and outcome but is most recent: Weight 0.103.
- Interval44 is a perfect match and second-most recent: Weight 0.305.
- Interval43 mismatches type and is third-most recent: Weight 0.053.
- Interval42 mismatches outcome and is fourth-most recent: Weight 0.098.[5]

Chunks of the intermediate-tap type and correct outcome (e.g., interval44) tend to have a higher contribution than those of the fast-tap type (e.g., interval43). However, due to the contribution of fast-tap chunks, the final blended value is smaller than it would have been if only intermediate-tap chunks had contributed to blending, which explains the too-early bias in the fast-tap conditions. The same mechanism applies to the slow-tap games, only in the opposite direction. In slow-tap games, due to the contribution of slow-tap chunks, the blended value for the intermediate interval becomes larger than it is supposed to be.

The model performs blending separately to produce the pulse value and the feedbackshift value. The top-right box in Fig. 9 shows the outcome of the blending computations, which is a chunk with blended pulse and feedbackshift values (15.661 and 0.321). If a match score (computed based on the activation of chunks in the set) exceeds the retrieval threshold, then the model uses the sum of the two blended

[5] Note that even though Interval43 and Interval42 both mismatch on one feature and Interval43 is more recent, Interval42 receives greater weighting due to activation noise.

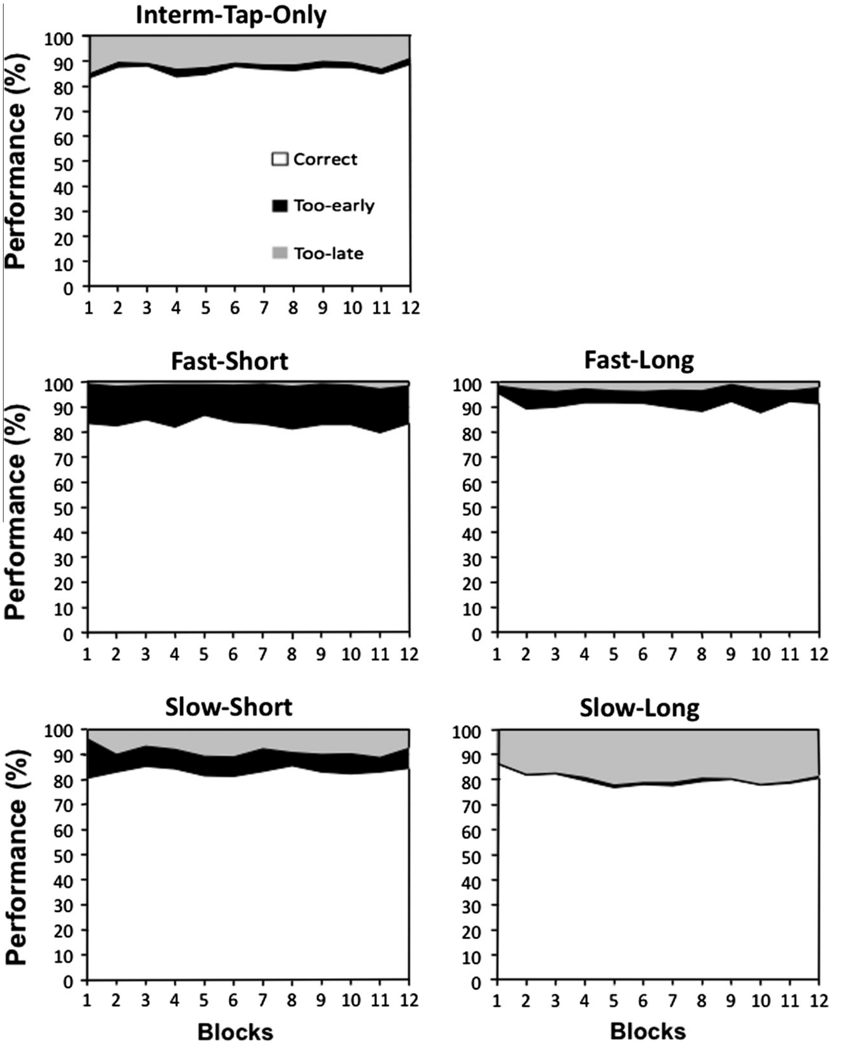


Fig. 10. IFF tapping performance of the model. Change in the percentages of IFF tapping response categories in intermediate-tap-only games and fast/slow-tap games as a function of distance (short/long).

values to set the criterion for the IFF interval (in this example, 15.661 + 0.321 = 15.982). The model executes the second IFF tap when the pulse value in the temporal buffer is greater than or equal to the criterion (in this example, when it reaches 16). More details regarding the blending mechanism can be found in Appendix B.

4.2.4. 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. 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 shoot at the mine before it hits the ship. The model ignores the pulse value

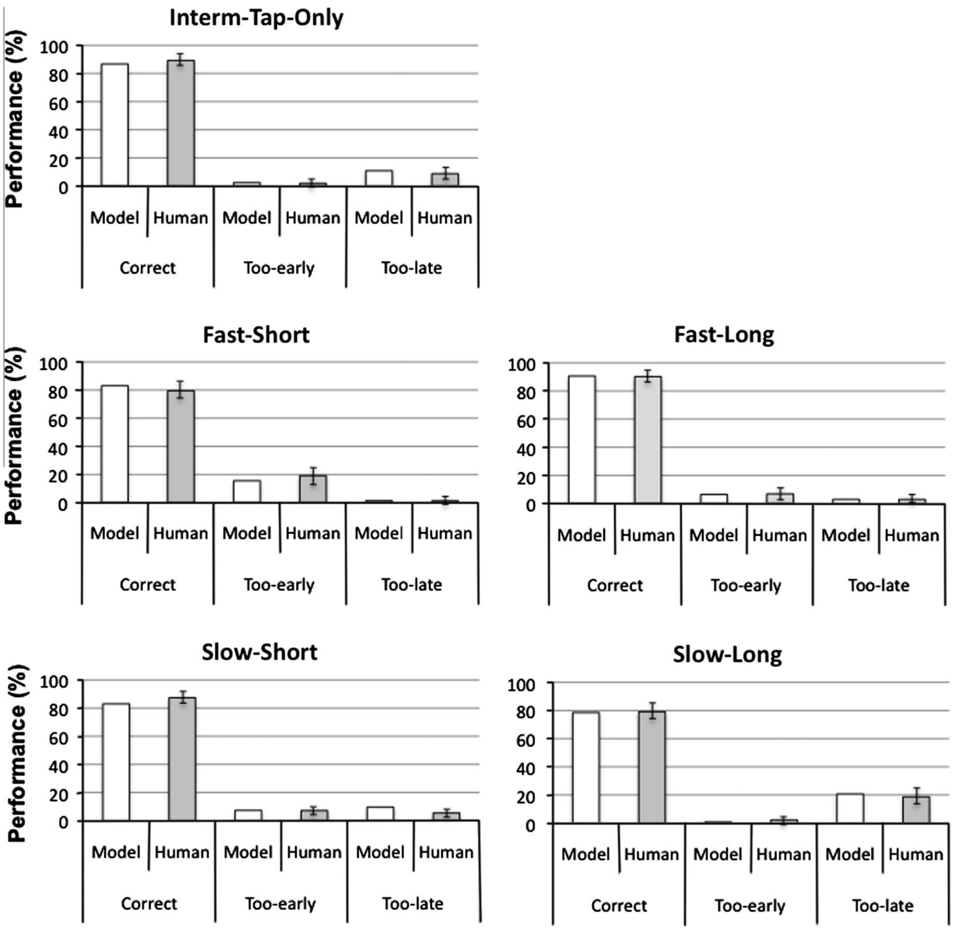


Fig. 11. Comparison of the model and participants in correct/too-early/too-late responses in the last 8 blocks. Error bars represent standard errors of the means.

in the temporal buffer when this production fires. Similar to participants' tendency to adjust the IFF interval based on how much time was left, the model adaptively decides when to terminate the interval based on the changing state of the environment. This emergency production never needs to fire in the long-distance trials.

4.3. Model results

Fig. 10 shows the model's performance in the IFF tapping task over 12 blocks. In contrast to humans (see Figs. 4 and 5), it starts with a relatively high accuracy level. 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 such as failures to make any response at all. 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.

The performance of the model depends on many parameters associated with the various modules. Most of these parameters have pre-established values in ACT-R and were not changed for this

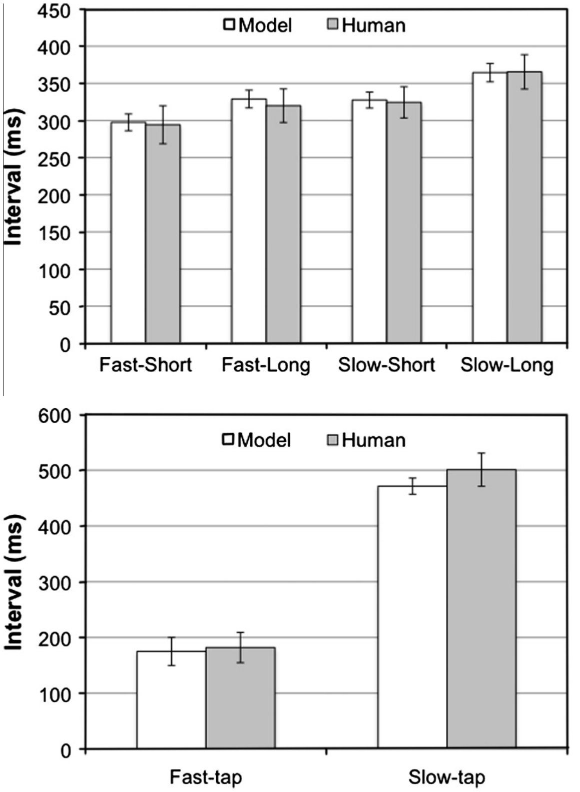


Fig. 12. Comparison of the model and participants in the produced IFF intervals (top) and in the produced fast/slow-tap intervals (bottom) in the last 8 blocks. Error bars represent standard deviations.

experiment. For instance, the motor parameters were taken from the EPIC (Executive-Process/Interactive Control) model of Meyer and Kieras (1997). Table C1 in Appendix C gives the critical parameters for the retrieval and temporal modules, which do not have established default values in ACT-R.

Fig. 11 (see Table C2 in Appendix C for exact values) 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.

The model's mean produced IFF intervals (Fig. 12, top) are reasonably close to those of participants. The average IFF interval is shortest in the fast-short condition (297 ms), longest in the slow-long condition (364 ms), and intermediate in the fast-long (328 ms) and the slow-short (327 ms) conditions. The model also produces fast-tap and slow-tap intervals (Fig. 12, bottom) that are close to those of participants. The distribution of the IFF intervals of the model in the last 8 blocks can be found in Fig. 13 (see the bottom of Fig. 6 for comparison with participants). The model and human distributions show similar shift in their peaks. The correlations between the 5 corresponding distributions in Figs. 6 and 13 range from .974 to .989, with an overall correlation of .98.

5. Discussion

Research in time estimation has mostly focused on timing behaviors performed as a primary task in relatively simple, static environments. Although this approach, studying timing behaviors in isolation,

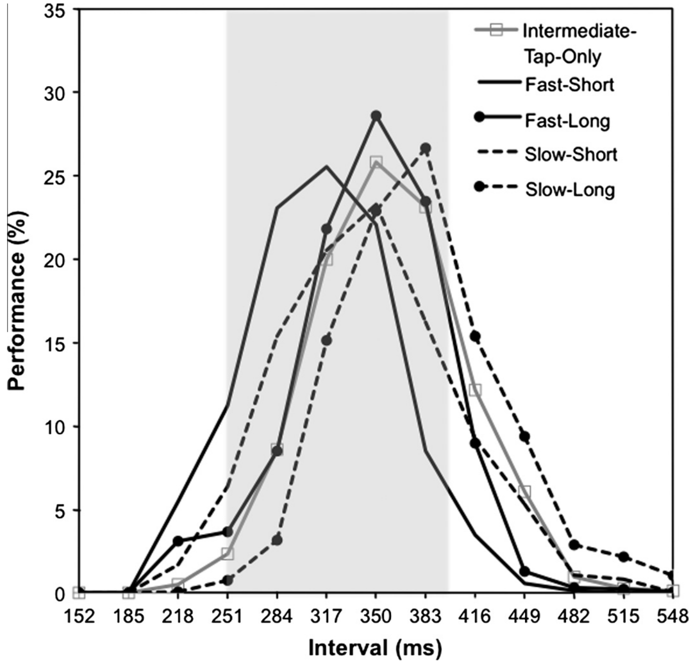


Fig. 13. Distributions of the IFF intervals produced by the model in the last 8 blocks.

has made significant contributions to advancing theories of time estimation, one may need a different approach to expand understanding of temporal cognition to a wider range of behaviors performed outside the standard time estimation paradigms. Time estimation embedded in complex multitasking situations has received little attention in timing research despite its significance in everyday activities. The current study aimed to understand multiple factors responsible for timing behaviors in multitasking by modeling them in an integrated framework of cognition.

5.1. Contamination and time pressure explain too-early bias

Two factors appear to be responsible for the too-early bias in time estimation that occurs in the context of multitasking. First, producing different time intervals contaminated estimation of the target interval. 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 and our model was able to produce the contamination effects seen in the data. The contamination effect suggests that different time intervals can interfere with each other when those intervals are estimated in the same context, consistent with the memory-mixing effect in time estimation (e.g., Gu & Meck, 2011).

Second, the time allowed for the task influenced time estimation. Our behavioral results indicated that executing a set of multiple responses under high time pressure impaired performance in the target interval estimation. Participants under time pressure showed a tendency to adjust the IFF interval depending on how much time was left for completing the task. This shows that time estimation can be sensitive to one's knowledge of what is about to happen, consistent with Church et al. (1991) in which the asymmetric responses of rats in the peak procedure reflected anticipation of the end of a trial and the conditions of the next trial. Our model captures this by having a procedural rule (the 'emergency' production) that overrides the outcome of the internal temporal estimate based on its perceptual processing of the external environment. Zakay's (1993) attention-based account for decision-making assumes division of attentional resources between temporal task and nontemporal task, and predicts an adoption of a simpler strategy in the nontemporal task under time pressure. In the current study, the assumption is that attentional resources were divided between two tasks that both involve time estimation: (1) Estimating the time progressed since the mine onset, and (2) the mine task that involves estimating the IFF interval. Time pressure presumably increased attention to the former task. Less attentional resources allocated to the latter mine task resulted in the adoption of a simpler strategy in the decision when to terminate the IFF interval.

The interaction between the two factors (Fig. 6) suggests that the contamination effect was larger in the long-distance condition than in the short-distance condition. The model also produced an interaction between tap speed and distance ($F(1,99) = 11.06$, $p = .001$, $\eta_\rho^2 = .10$). This interaction is produced in the model by the emergency production in the short-distance condition. The emergency production fires most of the trials (about 80%) in the short-distance condition. Because the outcome of blending is used only in about 20% of the trials in which the emergency production does not fire, the contamination effect is only partially observed in the short-distance condition. On the other hand, because the emergency production never fires in the long-distance condition, one can see the full rather than partial effect of the contamination effect. In other words, the contamination effect is weighed higher in the long-distance condition than in the short-distance condition.

Regardless of the conditions, participants showed a strong too-early bias in the early blocks (see Fig. 5). 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 IFF tapping across blocks. In early blocks, these other processes might have been so slow as to increase the use of the emergency rule. For example, possibly there was a speedup of how fast participants selected an appropriate key depending on the mine's identity. We showed earlier that there was a speedup of the pre-IFF activities over practice. 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, which has been associated with the demands of cognitive processing (Kahneman, 1973). Studies suggest that arousal can affect the subjective duration of intervals by speeding up the rate at which a pacemaker produces pulses (e.g., Burle & Casini, 2001; Penton-Voak, Edwards, Percival, & Wearden, 1996). For example, participants in Burle and Casini's (2001) study, who produced a target interval while hearing click trains that differed in intensity, produced shorter estimates under strong intensity than under weak intensity. Those studies predict that accumulation of pulses would complete in a shorter time in the aroused status, which shortens the produced interval. A brain imaging study by Anderson et al. (2011) in Space Fortress task showed phasic activity during the first two seconds after mine onset that indicated resource competition, which could possibly be related to a higher arousal level. Such arousal would be particularly high in the early blocks.

The contamination effect was present in both fast-tap trials and slow-tap trials. One might argue that the contamination hypothesis does not apply to fast-tap trials if participants were adopting a simplistic motor strategy (i.e., tap twice as fast as one can) without explicitly timing the interval. While this is the strategy adopted by our ACT-R model for executing fast-taps, the model remembers the intervals retrospectively. The question of how temporal information is remembered retrospectively has not been completely understood. Retrospective time estimation tends to be less accurate than prospective time estimation (Block & Zakay, 1997). Based on evidence that some variables differently influence experienced and remembered duration, different mechanisms have been claimed to underlie prospective and retrospective time estimation (Zakay & Block, 2004). On the other hand, the structural remembering approach (Boltz, 1998; Boltz, Kupperman, & Dunne, 1998) claims that people are able to incidentally learn and remember event durations in retrospective situations with high accuracy when event structures are highly predictable (i.e., temporal and nontemporal information bear a lawful relationship to one another).[6] This account predicts that time duration can be incidentally learned without additional attentional effort when people learn to temporally coordinate sequence of actions (e.g., learn a new motor skill) or attend to melodies with a hierarchical arrangement of melodic structure. We suspect that the repeated execution of motor sequences for the fast-tap created a highly predictable event structure in which participants were able to represent the durations retrospectively.

5.2. Modeling time estimation in ACT-R cognitive architecture

Block (1989) argued, "A complete understanding of any kind of temporal experience is possible only if we consider complex interactions among all of these factors". Inspired by Newell (1990), cognitive architecture aims to explain all aspects of cognition within a coherent framework. In line with previous efforts (e.g., Taatgen & van Rijn, 2011), we suggest that modeling complete processes of time estimation from perception to action in the context of the entire cognitive system can lead towards a better understanding of timing behaviors. Our model's interaction with the environment through its perceptual/motor module and the memory retrieval mechanism provided quantitative accounts of the too-early bias in the data. The model's time estimation was based on the pacemaker–accumulator internal clock model (Matell & Meck, 2000). However, more critical aspects of our modeling work are: (1) The contributions of the declarative (memory contamination) and procedural (time pressure) components to time estimation, (2) the ability of the model to disambiguate between the two contributions, and (3) modeling behaviors outside the standard time estimation paradigms.

The blending mechanism is a part of ACT-R's declarative memory mechanism and explains how time intervals are represented in memory and interact with each other. It was originally developed as an account for general memory process yet it has been shown to explain the memory-mixing effect in interval timing (Taatgen & van Rijn, 2011). One of the limitations of the scalar timing model (Gibbon et al., 1984) is the ability to account for how time estimation is influenced by representations of other durations in the same context. We suggest that modeling time estimation based on general memory mechanisms developed outside the domain of timing is one way toward a better understanding of time estimation. This approach can further contribute to addressing theoretical questions about memory mechanisms of time estimation, for example, whether memory mechanisms underlying processing temporal information are fundamentally different from those underlying processing nontemporal information (Brown, McCormack, Smith, & Stewart, 2005).

The contribution of the procedural component lies in the ability to model the effect of nontemporal aspects of cognition on time estimation. The role of the procedural system in ACT-R is to coordinate behaviors of multiple modules to achieve coherent behavior (Anderson et al., 2004). In our model, the procedural system prioritized the outcome of the visual module over the outcome of the temporal module when there was high time pressure. This illustrates that when internal temporal estimation is accurate, timing errors can still occur due to other cognitive processes. Models in ACT-R produce precise quantitative predictions on how long each process will take, which allows one to assess relative contributions of the declarative component (e.g., latency for blending) and the procedural component (e.g., latency for encoding visual input) to timing performance. Although a similar degree of too-early bias was present in the fast-long condition and the slow-short condition, the model was able to disambiguate between different sources of error in the two conditions: The declarative contribution in the former and the procedural contribution in the latter. We suggest that modeling in cognitive architecture allows one to understand complex interactions among multiple factors in timing behaviors because one can separately examine different contributions of those factors.

[6] In Boltz et al. (1998), when participants learned a novel motor activity consisting of several independent steps (e.g., building a model car) with varying number of trials, their retrospective verbal estimation of task duration became more accurate in the later stage of learning (i.e., when event structure became highly predictable) than in the earlier stage. In (Boltz, 1998) in which participants attended to tunes with coherent melodic structure, the reproduction of the total duration of the tunes was highly accurate regardless of whether participants attended to temporal information, nontemporal information, or both during the tune presentation suggesting that both kinds of information can be jointly encoded without additional attentional effort. The predictions of the structural remembering approach were supported in retrospective estimation of minute-range (Boltz et al., 1998) as well as second-range (approximately 10 s in Boltz, 1998) durations.

Table C1
The critical parameters for the retrieval and temporal modules.

Name	Description	Value
:rt	The retrieval threshold. The minimum activation a chunk must have to be retrieved	1.0
:lf	The latency factor, which determines the time it takes the declarative module to respond to a request for a chunk	1.1 s
:ans	The activation noise parameter used to generate the instantaneous noise added to the activation of a chunk. This value is typical of ACT-R models where the noise parameter is less than 1 and usually less that .5	.385
:mp	The mismatch penalty	2.25
similarity	The similarity between the value in the retrieval specification and the value currently in the corresponding slot of the chunk in the buffer. The default range is from 0 (highest similarity) to -1 (lowest similarity)	fast:interm (−0.5) slow:interm (−0.2) too-early:correct (−0.7) too-late:correct (−0.7)
:time-mult	The multiplier for increasing the pulse length	1.1
:time-noise	The noise added to the pulse lengths	.0015
:time-master-start-increment	The length of the initial pulse (the time between the pulse count of 0 and the count of 1) in seconds	11 ms

Table C2
Model and human performance in correct, too-early, and too-late responses.

	Model			Human		
	Correct (%)	Too-early (%)	Too-late (%)	Correct (%)	Too-early (%)	Too-late (%)
Interm-Tap-Only	86.7	2.4	10.9	89.3	2.0	8.8
Fast-Short	83.1	15.6	1.4	79.5	19.2	1.3
Fast-Long	90.6	6.4	3.0	90.2	6.9	2.9
Slow-Short	83.0	7.3	9.7	87.3	7.2	5.5
Slow-Long	78.5	0.8	20.7	79.1	2.2	18.7

Finally, the current work illustrates that modeling in cognitive architecture allows one to study a wider range of timing behaviors without being bounded by standard time estimation paradigms. Performance in the intermediate-tap-only condition was remarkably different from performance in the other conditions. Participants did not show the too-early bias when they estimated the same interval without interference from a different temporal task or the mine task. This clear contrast demonstrates that time estimation performed in a dynamic task can exhibit properties different from those observed from standard paradigms in which time estimation is performed as an isolated task in static environments. Furthermore, time estimations in multitasking tend to involve more complex procedures than assumed in the standard paradigms. In the IFF task, the target interval is never presented, and time estimation is interleaved in a number of other attention-demanding tasks. The large procedural differences between timing in multitasking and timing in standard paradigms suggest that a better way to understand timing in multitasking is to model the complete processes of timing in cognitive architecture rather than oversimplifying the task in an attempt to fit it into the standard paradigms.

5.3. Implications for other fields of research

Besides the contributions of the current approach to the theoretical understanding of timing mechanisms, our results also provide implications to research in complex skill performance in which timing plays an important role. The performance difference between the intermediate-only condition and other conditions is relevant to the question of what is the optimal training to improve time estimation performance. In the skill acquisition literature, one of the well-known instructional strategies involves part-task training, which decomposes a complex task into multiple part-tasks and gives training on each part-task individually before practicing the whole task. In Frederiksen and White's (1989) application of part-task training to Space Fortress, training on the IFF interval estimation was similar to our intermediate-tap-only condition in that their participants simply practiced producing the target interval in a game isolated from the whole context of the Space Fortress task. An alternative training approach is the integrated emphasis-change strategy (Gopher, Weil, & Bareket, 1994; Gopher, Weil, & Siegel, 1989) which has participants perform the full task but manipulates the relative emphasis given to different components of the task. Comparison of the two approaches (Fabiani, Buckley, Gratton, Coles, & Donchin, 1989) showed that although the part-task approach led to better final performance, the integrated approach was more resistant to interference when concurrent tasks were introduced. Our results showed that good performance in the intermediate-tap-only condition did not transfer to good intermediate timing in the more complex games. While this does not necessarily mean that part-task training is less effective, it suggests that in order for the part-task approach to be successful, the part-task should be large enough to include the critical sequence of activities (e.g., pre-IFF, IFF, and post-IFF activities) so that they can be learned as a unit. If this condition is met, one might find that more learning opportunities offered in the easier condition facilitate successful transfer to the harder condition as in Taatgen et al. (2007).

Human factors researchers have studied timing performance and patterns of timing errors in dynamic multitasking situations (Levinthal & Rantanen, 2004; Rantanen & Xu, 2001; Xu & Rantanen, 2003). Participants in Rantanen and Xu's (2001) study were presented with a simulated busy traffic environment and let a pedestrian cross the street between successive vehicles in order to avoid collision by estimating a sufficient interval and timing the initiation of the action. They found that the proportion of too-early errors in release of the pedestrian increased when the temporal gap between successive vehicles was reduced. They also found that increasing the perceived accuracy demand (faster speed of the traffic) shifted participants toward earlier release times, suggesting that the visual element of the task dominated time perception. Such a situation is common in multitasking situations (e.g., driving a car or piloting an aircraft) in which one has to predict the future trajectories of moving objects in order to avoid collisions (Xu & Rantanen, 2003). Similar to those studies, our results can be potentially applied to addressing human factors issues such as improving safety and reducing errors in various time-critical multitasking situations. Identifying patterns of timing errors and investigating the underlying causes may suggest changes in work procedures, work environments, or training. For instance, work procedures can be organized such that a timing-critical task is separated from other tasks that involve less critical timing, avoiding memory contamination.

6. Conclusions

The time estimation mechanism in ACT-R has successfully captured human time estimation performance in attention-demanding dual-task conditions (Taatgen et al., 2007), estimation of multiple overlapping time intervals (van Rijn & Taatgen, 2008), as well as in dynamic multitasking situations such as driving (Salvucci et al., 2006), piano playing (Nguyen & Salvucci, 2006), and conversation (Trafton, Bugajska, Fransen, & Ratwani, 2008). The current study explored millisecond-level time estimation embedded in a complex real-time task that imposes especially high perceptual-motor demands. The model built in the ACT-R architecture 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 to the domain of complex skills.

Acknowledgments

This work was supported by ONR Grant N00014-09-1-0402 to Wayne Gray & John Anderson. The authors thank Shawn Betts, Dan Bothell, and Darryl Schneider for providing helpful comments on this work.

Appendix A. Instructions

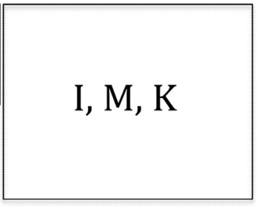
In this experiment, you will play games in which you need to respond to different stimuli by producing finger taps with fast/intermediate/slow intervals. There are 3 types of games (intermediate-tap-only, fast-tap, and slow-tap).

A.1. Intermediate-tap-only game

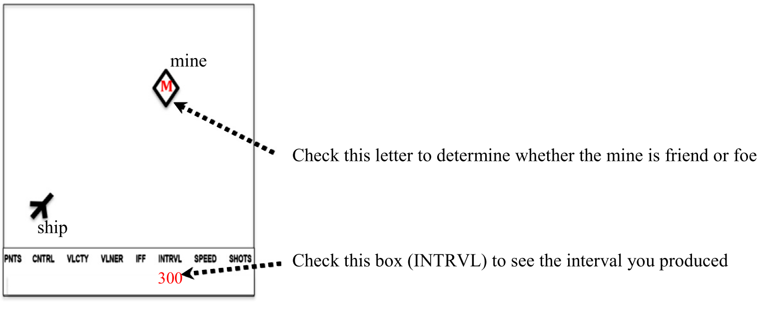
In this game, a series of letter stimuli (F or J) will appear in the screen. When the letter F appears, tap the F key twice with an **intermediate (between 250 and 400 ms)** interval. When the letter J appears, tap the J key twice with an **intermediate** interval. If you make a successful interval, the letter will disappear immediately (you will gain points). If you fail, you will need to wait a few seconds until it disappears (you will lose points).

A.2. FAST-TAP game

In the beginning of this game, you will see three alphabet letters in the screen. Make sure you remember those letters during the game. You are encouraged to use mnemonic strategy to remember the letters (e.g., "I am the king" for the "I, M, K").



During the game, you will see a ship stationed at the bottom left of the screen and a mine that appears every couple of seconds after the previous mine disappears.



There are two types of mines: red stationary mine and green moving mine.

When a **red stationary mine** appears, tap the **spacebar** twice with a **fast (< 250ms)** interval. If you make a successful interval, the mine will disappear immediately (you will gain points). If you fail, you will need to wait a few seconds until it disappears (you will lose points).

When a **green moving mine** appears, check the letter on the mine. If the mine contains any of the three letters shown in the beginning (as in the figure above), the mine is a **foe**. Otherwise the mine is a **friend**.

In order to destroy foe mine, (1) tap the **J key** twice with an **intermediate** interval and (2) shoot a missile (tap **spacebar**). In order to destroy friend mine, (1) tap **F key** twice with an **intermediate** interval and (2) shoot a missile (tap **spacebar**). If you successfully make the correct interval and shoot a missile, the mine will disappear immediately (you will gain points). If you fail to make the correct interval OR fail to shoot a missile, the mine will destroy your ship (you will lose points).

A.3. SLOW-TAP game

This game is identical to the Fast-tap game except that you will see blue stationary mines instead of red stationary mines.

For the blue stationary mine, tap the spacebar twice with a slow (between 401 and 650 ms) interval.

The response to green moving mines should be the same as the Fast-tap game.

After each double-tap, the interval you produced will appear in the INTRVL box (on the bottom panel of the screen) for a few seconds. In all games, you have only one chance for producing the interval. If you fail to produce the right interval, you cannot try again. For any stimulus (letter or mine), make the appropriate response as fast as you can. Your bonus money will depend on how successfully you handle the mines AND produce the correct intervals.

Appendix B. Blending and time estimation mechanism in ACT-R

B.1. Blending

In the subsymbolic structure of ACT-R, each chunk in declarative memory is associated with an activation value that reflects the degree to which past experiences indicate that the chunk will be useful at any particular moment. The activation value of the chunk is the sum of base-level activation (reflecting the recency and frequency of use of the chunk), a partial matching value (reflecting the degree to which the chunk matches the specification requested), and noise, as described in the following equation:

$$A_i = \ln\left(\sum_{k}^{n} t_k^{-d}\right) + \sum_{m} PM_{mi} + noise \tag{B.1}$$

n, the number of past presentations of chunk i; t_k, the time since the kth presentation; d, the decay parameter. We used the default value of 0.5; P, the match scale parameter (set with :mp in Table C1) that determines the weight given to similarity; M_{mi}, the similarity between the value m in the retrieval specification and the value in the corresponding slot of chunk i (set with similarity in Table C1).

The activation value determines how likely the chunk is to be retrieved and how long the retrieval takes. With the standard retrieval mechanism, the chunk that has the highest activation has the highest likelihood of retrieval if the activation value is above a threshold. With the blending mechanism, however, the chunk returned will reflect the weighted average of the chunks in memory. When the blending request is made, the matching set (i.e., the set of chunks that match the request) is found regardless of the activation of those chunks (e.g., interval45 through interval20 in Fig. 9).[7] The weight (P_i) given to each chunk in the matching set (chunk A_1 through chunk A_j) is determined based on the following Boltzmann equation:

[7] The size of the matching set grows as the model repeatedly generates chunks over the trials. By default blending can use all chunks that match the blending request. However, because older chunks have very little impact on blending outcome, we limited the size of the matching set such that only relatively recent chunks participate in the blending. The :min-bl parameter (a minimum base-level activation a chunk has to have to be considered in the matching-set) was set to −2.5.

$$P_i = \frac{e^{\frac{A_i}{t}}}{\sum_j e^{\frac{A_j}{t}}} \tag{B.2}$$

A_i, the activation of chunk i; A_j, the activation of chunk j in the matching set; t, temperature.

Chunks with better matches to the request and those experienced more recently are assigned a higher weight (P_i).[8] A blended chunk is created that has values that are weighted averages of the individual chunk values:

$$V = \sum_j P_j V_j \tag{B.3}$$

P_j, the weight for chunk j in the matching set; V_j, the value for chunk j in the matching set.

In Fig. 9, the blended pulse value 15.661 is the weighted average of individual pulse values: 0.103 * 14 + 0.305 * 18 + 0.053 * 13 + 0.098 * 17 + ⋯ + 0.009 * 17. The blended feedbackshift value 0.321 is the weighted average of individual feedbackshift values: 0.103 * (−6) + 0.305 * 0 + 0.053 * 0 + 0.098 * 3 + ⋯ + 0.009 * 3.

The blended chunk is given a match score M, the log of the sum over the chunks in the matching set of e to the power of A_j:

$$M = \ln\left(\sum_j e^{A_j}\right) \tag{B.4}$$

A_j, the activation of chunk j in the matching set.

Similar to the retrieval process, if the match score is equal to or greater than the retrieval threshold (:rt in Table C1), then the blending succeeds and the retrieval latency is computed using the match score. If the match score is lower than the retrieval threshold, then the blending results in failure and latency is determined by the retrieval threshold.

B.2. Time estimation

Time estimation in ACT-R occurs via the temporal module (Taatgen, van Rijn, & Anderson, 2007) and its interaction with the rest of the system. The temporal module is based on the internal clock model (Matell & Meck, 2000) that assumes a pacemaker generates pulses at certain intervals. Once the accumulator is reset to zero (by a request made to the temporal buffer), it starts counting pulses and automatically keeps accumulating the pulse counts as time progresses. The total pulse count accumulated during the interval indicates the estimate of the time interval. The production system can access the current pulse value via a chunk in the temporal buffer. When time estimation finishes, the temporal buffer can be cleared (stop the pulse accumulation).

The temporal module produces a logarithmic representation of time. The initial pulse length (t_0) is .011 s. The pulse length keeps increasing as time progresses. The following equation describes how the nth pulse length (t_n) is computed. Due to this logarithmic property, the time estimates are more accurate for shorter intervals than for longer intervals.

$$t_0 = start + \varepsilon_1 t_n = a * t_{n-1} + \varepsilon_2$$

Start, value of the :time-master-start-increment parameter; a, value of the :time-mult parameter; b, value of the :time-noise parameter; ε_1, noise generated with the act-r-noise command with an s of b * 5 * start; ε_2, noise generated with the act-r-noise command with an s of $b * a * t_{n-1}$.

[8] In our model, the same temperature value (t = 0.544) was set for all chunks throughout all blending computations, thus differences in weights are mostly explained by differences in chunk activation. The temperature parameter was set to its default value, which is the square root of 2 times :ans (instantaneous noise in Table C1).

Appendix C. Tables

See Tables C1 and C2..

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.

Anderson, J. R., Bothell, D., Fincham, J. M., Anderson, A. R., Poole, B., & Qin, Y. (2011). Brain regions engaged by part and whole-task performance in a video game: A model-based test of the decomposition hypothesis. *Journal of Cognitive Neuroscience, 23*, 3983–3997.

Barnes, R., & Jones, M. R. (2000). Expectancy, attention, and time. *Cognitive Psychology, 41*, 254–311.

Bateson, M. (2003). Interval timing and optimal foraging. In W. H. Meck (Ed.), *Functional and neural mechanisms of interval timing* (pp. 113–141). Boca-Raton, FL: CRC Press.

Baudouin, A., Vanneste, S., Pouthas, V., & Isingrini, M. (2006). Age-related changes in duration reproduction: Involvement of working memory processes. *Brain and Cognition, 62*, 17–23.

Block, R. A. (1989). Experiencing and remembering time: Affordances, context and cognition. In I. Levin & D. Zakay (Eds.), *Time and human cognition: A life span perspective* (pp. 333–363). Amsterdam: North Holland.

Block, R. A., & Zakay, D. (1997). Prospective and retrospective duration judgments: A meta-analytic review. *Psychonomic Bulletin & Review, 4*, 184–197.

Boltz, M. G. (1998). The processing of temporal and nontemporal information in the remembering of event durations and musical structure. *Journal of Experimental Psychology: Human Perception and Performance, 24*, 1087–1104.

Boltz, M. G., Kupperman, C., & Dunne, J. (1998). The role of learning in remembered duration. *Memory & Cognition, 26*, 903–921.

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

Brown, S. W. (1985). Time perception and attention: The effects of prospective versus retrospective paradigms and task demands on perceived duration. *Perception & Psychophysics, 38*, 115–124.

Brown, S. W. (1997). Attentional resources in timing: Interference effects in concurrent temporal and nontemporal working memory tasks. *Perception & Psychophysics, 59*, 1118–1140.

Brown, G. D., McCormack, T., Smith, M., & Stewart, N. (2005). Identification and bisection of temporal durations and tone frequencies: Common models for temporal and nontemporal stimuli. *Journal of Experimental Psychology: Human Perception and Performance, 31*(5), 919–938.

Brown, S. W., & West, A. N. (1990). Multiple timing and the allocation of attention. *Acta Psychologica, 75*, 103–121.

Burle, B., & Casini, L. (2001). Dissociation between activation and attention effects in time estimation: Implications for internal clock models. *Journal of Experimental Psychology: Human Perception and Performance, 27*, 195–205.

Church, R. M., Miller, K. D., Meck, 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.

Fabiani, M., Buckley, J., Gratton, G., Coles, M. G. H., & Donchin, E. (1989). The training of complex task performance. *Acta Psychologica, 71*, 259–299.

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.

Gallistel, C. R., & Gibbon, J. (2000). Time, rate, and conditioning. *Psychological Review, 107*, 289–344.

Gibbon, J., Church, R. M., & Meck, W. H. (1984). Scalar timing in memory. *Annals of the New York Academy of Sciences, 423*, 52–77.

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

Gopher, D., Weil, M., & Siegel, D. (1989). Practice under changing priorities: An approach to the training of complex skills. *Acta Psychologica, 71*, 144–177.

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

Gu, B.-M., & Meck, W. (2011). New perspectives on Vierordt's law: Memory-mixing in ordinal temporal comparison tasks. In A. Vatakis, A. Esposito, M. Giagkou, F. Cummins, & G. Papadelis (Eds.), *Time and time perception* (pp. 67–78). Berlin: Springer-Verlag.

Ivry, R., Spencer, R. M., Zelaznik, H. N., & Diedrichsen, J. (2002). The cerebellum and event timing. In S. M. Highstein & W. T. Thach (Eds.). *The cerebellum: Recent developments in cerebellar research* (Vol. 978, pp. 302–317). New York: New York Academy of Sciences.

Jazayeri, M., & Shadlen, M. N. (2010). Temporal context calibrates interval timing. *Nature Neuroscience, 13*, 1020–1152.

Jones, M. R. (1990). Musical events and models of musical time. In R. Block (Ed.), *Cognitive models of psychological time* (pp. 207–240). Hillsdale, NJ: Lawrence Erlbaum.

Jones, M. R., & McAuley, J. D. (2005). Time judgments in global temporal contexts. *Perception & Psychophysics, 67*, 398–417.

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

Kahneman, D. (1973). *Attention and effort*. Englewood Cliffs, NJ: Prentice-Hall.

Lebiere, C., Gonzalez, C., & Martin, M. K. (2007). Instance-based decision making model of repeated binary choice. In *Proceedings of the 8th international conference on cognitive modeling* (pp. 77–82), Ann Arbor, MI.

Lejeune, H., & Wearden, J. H. (2009). Vierordt's the experimental study of the time sense (1868) and its legacy. *European Journal of Cognitive Psychology, 21*, 941–960.
Levinthal, B. R., & Rantanen, E. M. (2004). Measurement of taskload and performance in a dynamic multi-task experiment. In *Proceedings of the 48th annual meeting of the human factors and ergonomics society* (pp. 567–570), Santa Monica, CA.
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.
Maule, A. J., & Hockey, G. R. J. (1993). State, stress and time pressure. In O. Svenson & A. J. Maule (Eds.), *Time pressure and stress in human judgment and decision making* (pp. 83–102). New York: Plenum.
McCormack, T., Brown, G. D. A., Smith, M. C., & Brock, J. (2004). A timing-specific memory distortion effect in young children. *Journal of Experimental Child Psychology, 87*, 33–56.
McElree, B. (2001). Working-memory and focal attention. *Journal of Experimental Psychology: Learning, Memory, and Cognition, 27*, 817–835.
Meyer, D. E., & Kieras, D. E. (1997). A computational theory of executive control processes and human multiple-task performance: Part 1. Basic mechanisms. *Psychological Review, 104*, 3–65.
Michon, J. A. (1967). *Timing in temporal tracking*. Assen: Van Gorcum.
Newell, A. (1990). *Unified theories of cognition*. Harvard University Press.
Nguyen, T. M., & Salvucci, D. D. (2006). Piano playing: A model of sight-reading and rhythmic timing. In *Proceedings of the 7th international conference on cognitive modeling* (pp. 208–212), Trieste, Italy.
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. In *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. In *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*, 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. A., 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.
Trafton, J. G., Bugajska, M. D., Fransen, B. R., & Ratwani, R. M. (2008). Integrating vision and audition within a cognitive architecture to track conversations. In *Paper presented at the proceedings of the 3rd ACM/IEEE international conference on human robot interaction*.
van Rijn, H., & Taatgen, N. A. (2008). Timing of multiple overlapping intervals: How many clocks do we have? *Acta Psychologica, 129*, 365–375.
Xu, X., & Rantanen, E. M. (2003). Timing of discrete actions: Evidence for perceptual dominance in human performance. In *Proceedings of the 47th annual meeting of the human factors and ergonomics society* (pp. 581–585), Denver, CO.
Zakay, D. (1990). The evasive art of subjective time measurement: Some methodological dilemmas. In R. A. Block (Ed.), *Cognitive models of psychological time* (pp. 59–84). Hillsdale, NJ: Erlbaum.
Zakay, D., & Block, R. A. (1997). Temporal cognition. *Current Directions in Psychological Science, 6*, 12–16.
Zakay, D., & Block, R. A. (2004). Prospective and retrospective duration judgments: An executive-control perspective. *Acta Neurobiologiae Experimentalis, 64*, 319–328.
Zakay, D., & Block, R. A. (1995). An attentional-gate model of prospective time estimation. In M. Richelle, V. D. Keyser, G. d'Ydewalle, & A. Vandierendonck (Eds.), *Time and the dynamic control of behavior* (pp. 167–178). Liège, Belgium: Universite de Liege.
Zakay, D., Nitzan, D., & Glicksohn, J. (1983). The influence of task difficulty and external tempo on subjective time estimation. *Perception & Psychophysics, 34*, 451–456.
Zakay, D. (1993). The impact of time perception processes on decision making under time stress. In O. Svenson & A. J. Maule (Eds.), *Time pressure and stress in human judgment and decision making* (pp. 59–72). New York: Plenum Press.