Steering Performance with Error-accepting Delays

Shota Yamanaka
Yahoo Japan Corporation
Chiyoda-ku, Tokyo, Japan
syamanak@yahoo-corp.jp

ABSTRACT

In steering law tasks, deviating from the path is immediately considered an error operation. However, in navigating a hierarchical menu item, which is a representative application of the law, a deviation within a short duration is sometimes permitted. We tested the validity of the steering law model with various durations of such *error-accepting delays* and found that it showed high fits for each delay condition ($R^2 > 0.96$) but poor fits if the delay values were not separated ($R^2 = 0.58$). Because the average movement speed linearly increased as the delay increased, we refined the model by taking the delay into account, and the fitness was significantly improved ($R^2 = 0.97$). Our model will help GUI designers estimate the average operational time on the basis of the menu item length, width, and error-accepting delay.

CCS CONCEPTS

• **Human-centered computing → HCI theory, concepts and models**; **Pointing**; **Empirical studies in HCI**;

KEYWORDS

Graphical user interfaces; indirect pointing; delay; menu design; steering law

ACM Reference Format:
Shota Yamanaka. 2019. Steering Performance with Error-accepting Delays. In *CHI Conference on Human Factors in Computing Systems Proceedings (CHI 2019), May 4–9, 2019, Glasgow, Scotland UK*. ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3290605.3300800

ACM ISBN 978-1-4503-5970-2/19/05…$15.00
https://doi.org/10.1145/3290605.3300800

1 INTRODUCTION

Evaluating the applicability of an existing human performance model in a specific task is an important topic in HCI, in addition to deriving the model. One example is testing the validity of Fitts' law [19] in various conditions (e.g., [10, 50]). Such an applicability evaluation is also important for the model that we investigate: the steering law [1, 17, 35], which predicts the movement time MT to navigate a constrained path.

The steering law has many applications such as car driving [18, 35, 49], using a sewing machine [32], and cutting a piece of paper with scissors [22, 46]. A critical application in terms of HCI is navigation in hierarchical menus [1, 5, 16]. Therefore, researchers have endeavored to more accurately model user behaviors when navigating menus and selecting items, e.g., by combining the steering and Fitts' laws [5, 16, 38].

In this paper, we measure user performance for steering law tasks in which users can deviate from a path for a short duration. In many implementations of hierarchical menus, a set of child submenus appears shortly after the cursor hovers over the related parent item. Hence, users do not always have to keep the cursor inside the boundaries of the parent item. The duration, which we call *error-accepting delay*, is typically configured by designers, e.g., using the `setTimeout` (JavaScript) or `delay` (jQuery) functions. Intuitively, as the error-accepting delay increases, users can move the cursor more quickly, as they do not have to carefully steer the cursor through a path. More strictly, (1) users can move so far outside the path as long as they can return to the path within a given duration, thus (2) providing a temporal tolerance outside the path is equivalent to an expanded spatial tolerance because users can stay within the delay constraint, and therefore (3) users can accelerate their speed as if the path width is expanded. However, long error-accepting delays require users to wait a long time for submenus to open after moving the cursor onto the parent item. Thus, the design of error-accepting delay is quite important in terms of usability.

To date, we have found no study on such a delay in steering law tasks. Conventionally, hitting the boundaries of a path is immediately considered an error attempt [1, 17, 35, 44, 47]. This is a natural rule for real-world tasks such as car driving [18, 35], but not necessary for menu navigations. The only exception is a work by Pastel [34], who determined that the percentage of sampled cursor trajectory points outside the path could reach up to 2%. However, in typical menu implementations, the trigger to display submenus is the duration of cursor hovering, and hence the result by Pastel does not directly answer our question. Investigating the effects of error-accepting delays and refining the steering law to estimate *MT* under a given delay will help designers determine an appropriate configuration of the delay for displaying menus; this motivated us to conduct this work. Our key contributions include:

- We conducted an experiment in mouse-steering tasks with error-accepting delays. To our knowledge, the effects of this condition (i.e., users could deviate from the path for a given duration) have never before been empirically tested.
- We showed that the conventional steering law showed very high fits ($R^2 > 0.96$) for each delay condition tested in our study (0 to 800 msec). This result provides evidence that the steering law is empirically valid even when the deviation from the path is permitted for a fixed duration.
- We derived a refined model that showed an excellent fit regardless of delay values ($R^2 = 0.97$ for $N = 48$ data points), while the baseline model showed a poorer fit ($R^2 = 0.58$). Our model helps GUI designers accurately estimate the average operational time to navigate hierarchical menus on the basis of the menu item length, width, and error-accepting delay.

2 RELATED WORK

Laws of Steering

Laws of steering through a restricted path have been proposed by Rashevsky [35] for car driving and by Drury [17] for pen drawing. For GUI operations, Accot and Zhai [1] proposed the global model to steer through a given path C:

$$MT = a + b \int_C \frac{ds}{W(s)} \quad (1)$$

where MT is in msec, s is the cursor position from the path start, $W(s)$ [mm] is the path width at s, and a and b are empirically determined constants. If the path length is A [mm], and the width is constant (W), the model can be simplified:

$$MT = a + b(A/W) \quad (2)$$

Various studies have demonstrated that the model shows good fits to various input devices [2, 37], small and large scales [3], preferred and non-preferred hands [22], various tracking directions [41, 56], various temporal constraints [54], and different priority on speed or accuracy [55].

Although the intercept a is usually close to zero but significant [23], a proportional version of this model is also theoretically valid [17, 30]:

$$MT = b(A/W) \quad (3)$$

Accot and Zhai proposed another form called the local law, which shows that the instantaneous speed v [mm/msec] is linearly related to the path width:

$$v = W(s)/\tau \quad (4)$$

where τ [msec] is an empirical constant. Although the speed in a path gradually increases while a straight path has a constant width [38, 47, 48], this simple relationship between v and W has also been supported by other researchers [17, 30, 35]. Given that the v is stable in a constant-width path [17, 30], the time to pass through an entire path is

$$MT = b\frac{A}{W} = b\frac{A}{\tau v} = b'\frac{A}{v} \ \ (\text{let } b' = b/\tau) \quad (5)$$

This model was also confirmed by Drury in the following form [23]:

$$MT = b(A/V) \quad (6)$$

where V [mm/msec] is the average movement speed throughout the path:

$$V = A/MT \quad (7)$$

This model assumes that the instantaneous speed v is stable, which means that the average speed V can be replaced by v. Thus, the following form is also valid [23]:

$$V = bW \quad (8)$$

Also, the intercept is usually small but significant [23]:

$$V = a + bW \quad (9)$$

In summary, in a constant-width straight path, although there are some variations, such as whether to use an intercept or whether to predict MT, V, or v on the left side of an equation, we have a consensus that the movement speed (V or v) linearly increases as the path width W increases, and thus the time required to navigate the path (MT) linearly decreases. It is suitable for our purpose here to use models that predict MT for estimating an average operational time.

Modifications of Laws of Steering

Models on path shapes other than constant-width linear paths have been also evaluated, including those on circular paths [2, 3, 23], curved paths [33], linearly narrowing paths [1], widening spiral paths [1], paths with a corner [34], successive path segments [47, 48], and those that calculate the MT difference between narrowing and widening paths [44, 45]. Because submovement distances change depending on path shape [31], we are also interested in investigating the effects of error-accepting delays in different path shapes. In this paper, however, we investigate only constant-width linear path segments as a first step to evaluate whether the delay affects user performance in steering motions.

When testing model fitness, Kulikov et al. proposed adjusting the W value on the basis of the observed stroke variability, called the *effective width method* [26]. This method, however, is not suitable for designers who want to know the average operation time under a given task [43, 51]. Hence, we test model fitness with the data of error-free trials.

Facilitating Menu Navigations

There are alternative techniques that enable users to easily navigate hierarchical menus. For example, setting a spatially wider area for a menu item directly eases navigation difficulty [15, 39, 40]. With regard to the steering law, this technique expands an item tolerance W so that users move the cursor less attentively. Another approach is shortening the distance A of the steering law. Kobayashi and Igarashi used gestures for menu navigation, where moving the cursor a short distance to the right opens submenus [25]. In Jumping Menu [6], when a parent menu item is clicked, the cursor jumps to the related child submenus to eliminate any steering operations.

Blocking the cursor from deviating an item tolerance is also effective for menu navigation. Using virtual gravity (or *force-fields* [5]) and a physical force [16] pulls the cursor to the path center. This is a kind of error-accepting technique in motor space, similar to expanding the path tolerance.

Using the marking menu [28] and its related methods (e.g., [8, 52, 53]) or using pie menus [12] helps users select an item without precise cursor positioning. See Bailly et al.'s recent survey [9] for a thorough review on visual menu selection techniques. To the best of our knowledge, no work has been conducted to evaluate the effects of temporal support for hierarchical menu navigation, although such a delay is widely used in many applications.

Effects of Delay on GUI Operations

Our concern with adding a delay to display submenus is that users have to wait to view the submenu items, which could have a negative effect on their subjective feelings. In addition, if a system has a *latency* or *lag* when reacting to a user's action, it directly increases the task completion time. In mouse-pointing tasks, Hoffmann [21] and MacKenzie and Ware [29] proposed models modified from Fitts' law to capture the negative effect of lag. Similarly, Friston et al. [20] conducted a steering law task with transmission latency. Note that their *system lag* is added to every loop from mouse movements to cursor positioning, while under our condition, the *lag* (error-accepting delay) is added only to the final phase of the parent menu item selection to expose the related submenus. While our focus is not on finding the optimal error-accepting delay, our proposed model will be helpful to accurately predict the user performance in terms of MT.

3 EXPERIMENT

Participants

Twelve participants were recruited from a local university (two females and ten males, M = 22.6 and SD = 1.89 years). All had normal or corrected-to-normal vision, were right-handed, and were familiar with mouse operations; six of them were daily mouse users. Each participant received 3,000 JPY (~27 US$) for their time.

Apparatus

The PC we used was a Sony Vaio Z (Core i7-5557U, 3.10 GHz, 4 cores; 16-GB RAM; Windows 10). The display was manufactured by Dell (model 2407WFPb; 24 inches diagonal, 1920 × 1200 pixel resolution, 518.4 × 324.0 mm display area, 3.70 pixels/mm; 16-ms response time; connected by an HDMI-to-DVI cable) and its refresh rate was set at 60 Hz. The input device was an iBuffalo optical mouse (model BSMBU05, Blue LED, 81.6 g, 1000 dpi; 1.5-m USB 2.0 cable). As the steering law has shown excellent fits for mouse operations [2, 26, 27, 37, 38, 41, 42], we assumed a mouse was appropriate to observe the effects of error-accepting delay in steering law tasks. The cursor speed was set to the default, and pointer acceleration, or the *Enhance pointer precision* setting in Windows 10, was enabled to allow the participants to perform mouse operations with higher ecological validity [14]. Using the pointer acceleration does not violate the steering law [2], and it is consistent with consumer OS settings such as Windows and macOS. We used a large mousepad (43 cm × 29 cm). The experimental system was implemented with Hot Soup Processor 3.4 and used in full-screen mode. The system reads and processes input at approximately 1,000 times per second.

Latency was measured with a Casio EXILIM EX-ZR4000WE camera at 1,000 FPS. The mouse was hit with a hard object at high speeds, and the number of frames from when the mouse stopped to when the cursor stopped was counted. We repeated this action 30 times, and the average latency was 57.9 msec (SD = 11.2). This is in the range of typical mouse-display latencies of approximately 55 to 82 msec [13]. Therefore, we assume that the latency of our experimental system would not have a significant negative effect on user performance.

Task

The task was to click on the blue start area, pass through the white path, and then click on the green end area, as shown in Figure 1a. Between the two clicks, if the cursor entered the gray out-of-path areas and the time was over the error-accepting delay T_{dealy}, a beep sounded and the cursor position was marked as a cross ("×") to flag a steering error operation ER_{steer}. If the cursor returned to the path within

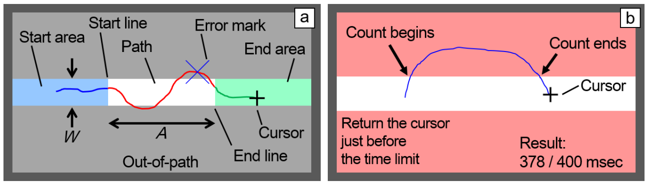

Figure 1: (a) Parameter definitions in the experiment. (b) Exercise system.

a given T_{delay}, ER_{steer} was not marked. Note: the deviation duration was not summed in a trial; the T_{delay} was given to each deviation. For example, in Figure 1a, the first deviation was equal to or less than the given T_{delay}, but the second was over the T_{delay} and thus marked as an ER_{steer}.

Participants were asked to complete the task without restarting even if an ER_{steer} occurred or when they clicked outside the start and end areas (ER_{click}). Although a click was needed after steering through the path, the green end area was sufficiently long along the movement direction (x-axis), and thus the motion in the white path was a steering rather than a combination of steering with pointing [38].

Participants were instructed to minimize the time between the two clicks and to not make any ER_{steer} or ER_{click} errors. Although our instructions were designed to motivate participants during the entire trial, the data to be analyzed were only those between the start and end lines, as in conventional steering law studies; i.e., the time after crossing the start line to crossing the end line (judged on the basis of the cursor's x-position) was measured as MT. After each trial, a large circular button labeled "Next" appeared at a random position and participants clicked it to reveal the next path condition. Movement direction was always to the right.

Design

The experiment was a $6 \times 2 \times 4$ within-subjects design with the following independent variables and levels. Three independent variables were included: error-accepting delay T_{delay}, path length A, and path width W. T_{delay} had six values (0, 100, 200, 400, 600, and 800 msec). Under the T_{delay} = 0 msec condition, deviating from the path immediately caused an ER_{steer}, and this was the baseline condition, as in conventional steering law tasks. The other values were selected on the basis of human reaction time. Because the time to correct hand movements based on visual feedback is longer than 200 msec [36] (approximately 260 [24] or 290 msec [30]), we assumed that participants could return the cursor to the path in ~300 msec (including system latency) after noticing any deviation from the path. Therefore, we set T_{delay} values ranging from less than to sufficiently longer than human reaction time to observe the effects of error-accepting delay. A had two values (480 and 640 pixels; 130 and 173 mm, respectively) and W had four values (15, 23, 33, and 45 pixels; 4.05, 6.21, 8.91, and 12.2 mm) so that the A/W values (10.7–42.7) were beyond 10, which requires continuous visually controlled steering movements [37, 38, 42].

We measured five dependent variables: ER_{steer} rate, temporal ratio of deviation from the path ($Ratio_{out}$), average count of deviations from the path per trial ($Count_{out}$), MT, and V. The ER_{click} rate was not included as a dependent variable because our main focus was on analyzing steering operations.

Procedure

One *block* consisted of a random order of $2A \times 4W \times 7$ repetitions = 56 trials with a fixed T_{delay} value. The first repetition was considered practice. Before each block, the participants used an exercise system (described below) to learn an error-accepting delay of the next block, except for the T_{delay} = 0 msec condition. The order of the six T_{delay} values was balanced among the 12 participants. In total, we recorded $2A \times 4W \times 6$ repetitions $\times\ 6T_{delay} \times 12$ participants = 3,456 data points. This task took about 30 min per participant.

Exercise to Learn a Given Error-accepting Delay

As shown in Figure 1b, participants moved the cursor into the top or bottom pink area, beginning the time measurement. Just before the time surpassed a given T_{delay}, they returned the cursor to the center white area. If the measured time was over T_{delay}, the cursor position was marked as a cross and a beep sound was played, the same as in the main experimental system. Participants repeatedly attempted this task and were expected to learn how to immediately return the cursor after deviating from the path. The session finished when a participant felt that s/he had sufficiently learned the T_{delay}, and the time required was typically 30 sec (~15 or 20 trials). This exercise session enabled us to simulate a situation where a participant was familiar with an error-accepting delay of a certain menu. Omitting this exercise system would simulate participants using a menu for the first time and re-learning the error-accepting delay at every trial. Observing such a learning effect is not part of this study.

Task Revisions based on a Pilot Study

To find issues in task parameters and our experimental system used in the main study, three students who did not participate in the main study joined our pilot study. Apparatus and values of T_{delay}, A, and W were the same as in the main study. The participants performed 8 (= $2A \times 4W$) trials for each T_{delay}; 48 trials per participant. We found three issues. (a) Participants did not know how to immediately return to the path when deviating from it, so we developed the exercise system. (b) The item currently being hovered over (the white path, or the top/bottom out-of-path areas) was immediately highlighted to provide visual feedback, as in typical menu implementations. However, when the cursor moved around the boundaries of the path, the colors of the path and out-of-path areas rapidly switched. This interrupted the operation, therefore, the color changing was not used in the main study. (c) We used an infinite height of start/end areas on the y-axis, as in [26, 47], and we did not require any mouse clicks in the start and end areas. However, the participants found that, even when the T_{delay} was short, they could complete a trial without any precise mouse control simply by moving the cursor from left to right on the screen. This was not appropriate for the purpose of this study (namely, to observe the T_{delay} effects in menu navigation and selection), so we limited the heights and required clicks there.

4 RESULTS

We analyzed the data via repeated-measures ANOVA. If a main effect was found, we performed pair-wise comparisons with Bonferroni correction as the p-value adjustment method. Note: if a significant main effect of T_{delay} is found, it is possible that no significant differences are found for any pair-wise tests of T_{delay} values. In Figure 2b and d, “n.s.” represents the results of pair-wise tests.

Error Rate

We observed 92 ER_{steer} trials and 150 ER_{click} trials, whose error types were inclusive. The remaining number of valid (error-free) trials was 3,227 (93.4%). We analyze the ER_{steer} as the main error result. The overall mean ER_{steer} rate was 2.66%, and a longer T_{delay} tended to decrease the ER_{steer} rate from 6.60% to 0% (Figure 2a). We found significant main effects of T_{delay} ($F_{5,55} = 8.748$, $p < 0.001$) and W ($F_{3,33} = 21.491$, $p < 0.001$) on ER_{steer} rate, but not for A ($F_{1,11} = 0.446$, $p = 0.518$). A significant interaction was found for $T_{delay} \times W$ ($F_{15,165} = 2.582$, $p < 0.01$).

Ratio of Deviation from the Path ($Ratio_{out}$)

We define $Ratio_{out}$ as [(Out of path time)/MT] × 100%. We wanted to determine if a longer error-accepting delay causes users to move the cursor less precisely. Assuming that the system clock is constant, $Ratio_{out}$ is equivalent to *Out of path movement* (*OPM*), which is the percentage of sampled cursor points outside the path [26].

The overall mean $Ratio_{out}$ was 2.21%, and a longer T_{delay} increased the $Ratio_{out}$ from 0.16% to 4.82% (Figure 2b). We found significant main effects of T_{delay} ($F_{5,55} = 4.585$, $p < 0.01$) and W ($F_{3,33} = 28.893$, $p < 0.001$) on $Ratio_{out}$, but not for A ($F_{1,11} = 2.107$, $p = 0.175$). A significant interaction was found for $T_{delay} \times W$ ($F_{15,165} = 3.431$, $p < 0.001$).

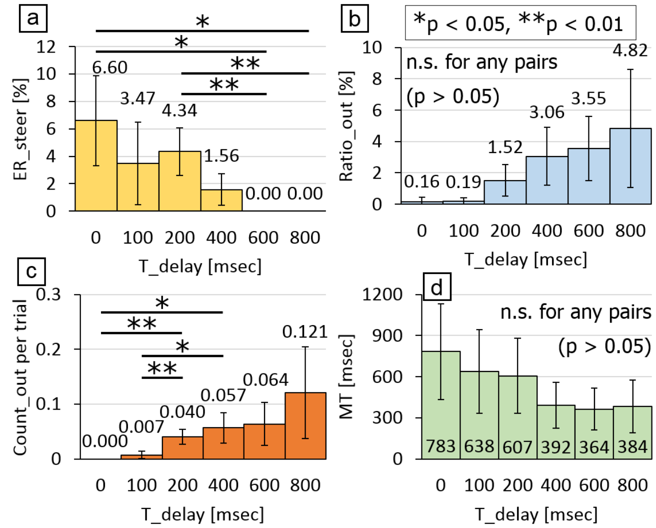

Figure 2: Main effects of T_{delay} on (a) ER_{steer}, (b) $Ratio_{out}$, (c) $Count_{out}$, and (d) MT. Error bars show 95% CIs.

Average Count of Deviation from the Path ($Count_{out}$)

We measured how often the cursor left the path per successful trial as $Count_{out}$. Figure 2c shows that $Count_{out}$ increased from 0 to 0.121 as T_{delay} increased, which means that a longer T_{delay} helped the participants overshoot then return to the path more effectively. The overall mean $Count_{out}$ was 0.0482. We found significant main effects of T_{delay} ($F_{5,55} = 5.996$, $p < 0.001$) and W ($F_{3,33} = 29.000$, $p < 0.001$) on $Count_{out}$, but not for A ($F_{1,11} = 2.467$, $p = 0.145$). Significant interactions were found for $T_{delay} \times W$ ($F_{15,165} = 3.689$, $p < 0.001$) and $A \times W$ ($F_{3,33} = 4.267$, $p < 0.05$).

Movement Time (MT)

Figure 2d shows the effect of T_{delay} on MT. The overall mean MT was 528 msec, which is reasonable compared with that of a related study [2] (mean MT was 768 msec for a linear mouse steering task). MTs for T_{delay} = 0–800 msec were 783, 638, 607, 392, 364, and 384 msec, respectively. T_{delay} reduced MT 53% at most, but any pair-wise tests on T_{delay} showed no significant difference ($p > 0.05$). We found significant main effects of T_{delay} ($F_{5,55} = 8.050$, $p < 0.001$), A ($F_{1,11} = 38.131$, $p < 0.001$), and W ($F_{3,33} = 14.852$, $p < 0.001$) on MT. Significant interactions were found for $T_{delay} \times A$ ($F_{5,55} = 10.739$, $p < 0.001$), $T_{delay} \times W$ ($F_{15,165} = 5.446$, $p < 0.001$), and $T_{delay} \times A \times W$ ($F_{15,165} = 2.679$, $p < 0.01$).

Average Movement Speed (V)

The overall mean V was 0.686 mm/msec. We found significant main effects of T_{delay} ($F_{5,55} = 6.946$, $p < 0.001$) and W ($F_{3,33} = 25.653$, $p < 0.001$) on V, but not for A ($F_{1,11} = 0.133$, $p = 0.722$). No significant interaction was found ($p > 0.05$). The interaction for $T_{delay} \times W$ was not significant ($F_{15,165} = 0.805, p = 0.671$); this demonstrated the independence of T_{delay} from W. Vs for T_{delay} = 0–800 msec were 0.466, 0.614, 0.547, 0.806, 0.823, and 0.861 mm/msec, respectively. So V increased 85% at most, but any pair-wise tests on T_{delay} showed no significant difference ($p > 0.05$).

In summary, the results of $Ratio_{out}$ and $Count_{out}$ show that participants move the cursor less attentively as T_{delay} increased, but the ER_{steer} rate decreased due to the long T_{delay}. This resulted in a shorter MT and thus a higher V.

5 DISCUSSION

Model Fitting and Refinement

We test the fitness of the baseline model (Equation 2, $MT = a + b(A/W)$). Separating the T_{delay} values, each regression line with eight data points ($2A \times 4W$) shows high fits ($R^2 > 0.96$, Figure 3). The average movement times for a given T_{delay} can be accurately predicted, which benefits designers. However, if the T_{delay} values are merged, the fitness is strongly degraded ($R^2 = 0.58$, Figure 4a). Therefore, designers have to conduct a user study to estimate the MT every time they change the T_{delay} value.

To integrate the error-accepting delay in the MT model (Equation 2), we first check the effects of the task parameters on the average movement speed V. Again, we found (a) main effects of T_{delay} and W, and (b) no significant interaction. In particular, because the interaction factor of $T_{delay} \times W$ was not significant, V would be accurately estimated without using interaction factors. Finally, we found that the inclusion of the additive factor T_{delay} significantly improved the fitness of the $V - W$ relationship (Equation 8, $V = bW$):

$$V = 0.0820W, \quad \text{adj. } R^2 = 0.882 \quad (10)$$

$$V = 0.0597W + 0.000574T_{delay}, \quad \text{adj. } R^2 = 0.986 \quad (11)$$

where $V = A/MT$. Here, we use 24 data points ($4W \times 6T_{delay}$) because there was no main effect of A and interactions related to A on V. In Equation 11, both W and T_{delay} are significant contributors ($p < 10^{-10}$). Thus, we empirically obtain the following form, where c and d are constants:

$$V = cW + dT_{delay} \quad (12)$$

Applying this result to Equation 6 ($MT = b(A/V)$), we obtain:

$$\begin{aligned} MT &= b\frac{A}{cW + dT_{delay}} \\ &= b\frac{A}{c(W + d'T_{delay})} \quad (\text{let } d' = d/c) \\ &= b'\frac{A}{W + d'T_{delay}} \quad (\text{let } b' = b/c) \quad (13) \end{aligned}$$

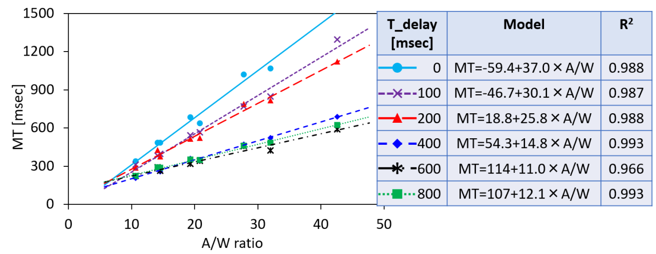

Figure 3: Steering law fitness for each T_{delay}.

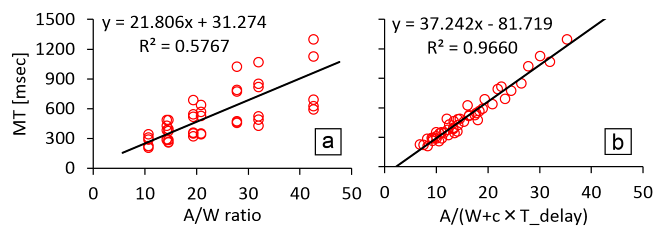

Figure 4: Steering law fitness for (a) baseline and (b) refined models for all data points ($N = 48$).

Table 1: Model fitting results with adjusted R^2 (higher is better) and AIC (lower is better) values for candidate models. a, b, and c are estimated constants with 95% CIs [lower, upper].

Eq.	Model	a	b	c	adj. R^2	AIC
2	A/W	31.3 [-107, 169]	21.8 [16.3, 27.4]		0.577	647
14	$\frac{A}{W + cT_{delay}}$	−81.7 [−123, −40.1]	37.2 [35.1, 39.3]	0.00839 [0.0266, 0.0356]	0.966	528

As suggested by Hoffmann [23], we check this model with an intercept in the following final form:

$$MT = a + b\frac{A}{W + cT_{delay}} \quad (14)$$

Table 1 lists the regression expressions for the baseline model and the refined model considering the delay. Model fitness is improved for both the adjusted R^2 and AIC values. Figure 4b also illustrates the improvement. The only difference between the refined model (Equation 14) and the baseline model (Equation 2) is that the cT_{delay} term acts to expand the path width. For example, when T_{delay} is 200 msec, the width is essentially expanded by ∼1.68 mm (= 0.00839 × 200).

We checked if such an expansion helped steering operations. Because the result of $Count_{out}$ was less than 1 on average (Figure 2c), participants typically left only one side of the path. Thus, the path expansion was essentially $cT_{delay}/2$, resulting in 0, 0.420, 0.840, 1.68, 2.52, and 3.36 mm for T_{delay}

= 0 to 800 msec, respectively. The average values of maximum cursor offset from the path in successful trials were 0, 0.439, 1.789, 3.51, 2.94, and 3.85 mm, respectively. The R^2 for these values was 0.83, which shows that cT_{delay} reasonably represents the path expansion.

Even if we use the data with $Ratio_{out} = 0$, the adjusted R^2 of our model is 0.957. This shows that our model can be robustly used even if users do not deviate from the path. In contrast, the R^2 using only the data with $Ratio_{out} > 0$ cannot be calculated because no deviations from the path occurred in some conditions with large W.

Contribution to HCI and Implication for GUI

Conventionally, the steering law has been used to model menu navigation tasks, but the model itself does not permit deviating from a path. In contrast, in our study, the participants were permitted to deviate from the path as long as the deviation duration was within the given T_{delay}. Although the task requirement was significantly different from the conventional condition, the baseline model of the steering law (Equation 2) showed excellent fits for each given T_{delay} (Figure 3). This is the first evidence to show that the steering law is valid for a condition where users can deviate from the path for a given time. This expands the generality of the steering law for menu navigation in a more realistic implementation where the cursor can deviate from an intended menu item within a short duration. This result demonstrates that, if designers use a fixed error-accepting delay on a Web page or software application, using the conventional steering law is sufficient to accurately predict the MT on the menu.

The fitness improvement achieved by the refined model (Equation 14) can help designers accurately estimate MT for other T_{delay} values that have yet to be tested. For example, after conducting a user study involving two values of T_{delay} (e.g., 0 and 800 msec) with a set of A and W values, we obtain the constants (a, b, and c), and then we can predict MT values for other T_{delay} values such as 200 and 600 msec. To confirm this application, we regressed the MT data for T_{delay} = 0 and 800 msec ($N = 16$ data points), which showed $a = -63.4$, $b = 37.1$, and $c = 0.00669$ with adjusted $R^2 = 0.990$. Using these constants, we predicted the MTs for T_{delay} = 100, 200, 400, and 600 msec ($N = 32$ data points). Figure 5 shows the observed versus predicted MTs; R^2 = 0.954 with slope = 0.978 (both are close to 1), which indicates the high prediction accuracy of our model.

Limitations and Future Work

Our results are somewhat limited due to the experimental conditions, such as testing only two values for A. Although a study with 12 participants is common in CHI [11], further studies with more users in various user pools (e.g., older adults) will strengthen the generality of our model. We used a mouse in this study, but different input devices would result in different user reaction times, so testing the model's validity with other devices such as styli will provide a further contribution. The range of T_{delay} (0 to 800 msec) is also a limitation. If T_{delay} is much longer (e.g., 3 sec), users would not have to focus on the path boundaries any more, and thus the task requirement would only be to point and click the end area after clicking on the start area. This condition is called a *pointing with directional constraint* task tested in the study of the crossing paradigm [4, 7], in which Fitts' law was shown to be an appropriate model for this. Hence, our proposed model, which is based on the steering law, is not suitable for such a task. Our future work will include clarifying the valid T_{delay} range of our proposed model.

Our proposed model is derived on the empirical result that V linearly increases as T_{delay} increases. However, as mentioned above, we assume that this model is valid in a limited range of T_{delay}. If we had tested a wider range of T_{delay} values, other relationships such as a logarithmic function might be better. From this viewpoint, in addition to our empirical derivation, a more theoretical development/justification of the model will provide a further contribution to a better understanding of human motor behavior.

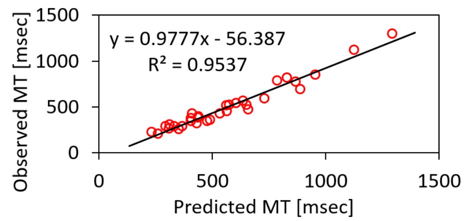

Figure 5: Observed vs. predicted MTs for T_{delay} = 100, 200, 400 and 600 msec by using MTs for T_{delay} = 0 and 800 msec.

6 CONCLUSION

We studied human performance in steering tasks with error-accepting delay. Although the steering law has conventionally considered a deviation from the path as an error, we regarded it as a success as long as the duration was within a given time (T_{delay}), as in typical hierarchical menu implementations. The conventional steering law showed excellent fits ($R^2 > 0.96$) for all T_{delay} conditions ranging from 0 to 800 msec, but the fit was poor if the T_{delay} values were not separated ($R^2 = 0.58$). On the basis of the result that the movement speed linearly increased as the T_{delay} increased, we showed that our model could accurately estimate the steering time ($R^2 = 0.97$). This refined model will help GUI designers estimate average user performance under a given delay. We hope this work expands the tolerance of the steering law by showing the wider applicability of the law in more realistic GUI conditions.

REFERENCES

[1] Johnny Accot and Shumin Zhai. 1997. Beyond Fitts' law: models for trajectory-based HCI tasks. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '97)*. 295–302. https://doi.org/10.1145/258549.258760

[2] Johnny Accot and Shumin Zhai. 1999. Performance Evaluation of Input Devices in Trajectory-based Tasks: An Application of the Steering Law. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '99)*. ACM, New York, NY, USA, 466–472. https://doi.org/10.1145/302979.303133

[3] Johnny Accot and Shumin Zhai. 2001. Scale effects in steering law tasks. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '01)*. 1–8. https://doi.org/10.1145/365024.365027

[4] Johnny Accot and Shumin Zhai. 2002. More than dotting the i's — foundations for crossing-based interfaces. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '02)*. 73–80. https://doi.org/10.1145/503376.50339

[5] David Ahlström. 2005. Modeling and Improving Selection in Cascading Pull-down Menus Using Fitts' Law, the Steering Law and Force Fields. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '05)*. ACM, New York, NY, USA, 61–70. https://doi.org/10.1145/1054972.1054982

[6] David Ahlstrom, Rainer Alexandrowicz, and Martin Hitz. 2006. Improving Menu Interaction: A Comparison of Standard, Force Enhanced and Jumping Menus. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '06)*. ACM, New York, NY, USA, 1067–1076. https://doi.org/10.1145/1124772.1124932

[7] Georg Apitz, François Guimbretière, and Shumin Zhai. 2008. Foundations for Designing and Evaluating User Interfaces Based on the Crossing Paradigm. *ACM Trans. Comput.-Hum. Interact.* 17, 2, Article 9 (May 2008), 42 pages. https://doi.org/10.1145/1746259.1746263

[8] Gilles Bailly, Eric Lecolinet, and Laurence Nigay. 2008. Flower Menus: A New Type of Marking Menu with Large Menu Breadth, Within Groups and Efficient Expert Mode Memorization. In *Proceedings of the Working Conference on Advanced Visual Interfaces (AVI '08)*. ACM, New York, NY, USA, 15–22. https://doi.org/10.1145/1385569.1385575

[9] Gilles Bailly, Eric Lecolinet, and Laurence Nigay. 2017. Visual Menu Techniques. *Comput. Surveys* 49, 4, Article 60 (Dec. 2017), 41 pages. https://doi.org/10.1145/3002171

[10] Xiaojun Bi, Yang Li, and Shumin Zhai. 2013. FFitts Law: Modeling Finger Touch with Fitts' Law. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '13)*. ACM, New York, NY, USA, 1363–1372. https://doi.org/10.1145/2470654.2466180

[11] Kelly Caine. 2016. Local Standards for Sample Size at CHI. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems (CHI '16)*. ACM, New York, NY, USA, 981–992. https://doi.org/10.1145/2858036.2858498

[12] J. Callahan, D. Hopkins, M. Weiser, and B. Shneiderman. 1988. An Empirical Comparison of Pie vs. Linear Menus. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '88)*. ACM, New York, NY, USA, 95–100. https://doi.org/10.1145/57167.57182

[13] Géry Casiez, Stéphane Conversy, Matthieu Falce, Stéphane Huot, and Nicolas Roussel. 2015. Looking Through the Eye of the Mouse: A Simple Method for Measuring End-to-end Latency Using an Optical Mouse. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology (UIST '15)*. ACM, New York, NY, USA, 629–636. https://doi.org/10.1145/2807442.2807454

[14] Géry Casiez and Nicolas Roussel. 2011. No More Bricolage!: Methods and Tools to Characterize, Replicate and Compare Pointing Transfer Functions. In *Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology (UIST '11)*. ACM, New York, NY, USA, 603–614. https://doi.org/10.1145/2047196.2047276

[15] Andy Cockburn and Andrew Gin. 2006. Faster Cascading Menu Selections with Enlarged Activation Areas. In *Proceedings of Graphics Interface 2006 (GI '06)*. Canadian Information Processing Society, Toronto, Ont., Canada, Canada, 65–71. http://dl.acm.org/citation.cfm?id=1143079.1143091

[16] Jack Tigh Dennerlein, David B. Martin, and Christopher Hasser. 2000. Force-feedback Improves Performance for Steering and Combined Steering-targeting Tasks. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '00)*. ACM, New York, NY, USA, 423–429. https://doi.org/10.1145/332040.332469

[17] Colin G. Drury. 1971. Movements with lateral constraint. *Ergonomics* 14, 2 (1971), 293–305. https://doi.org/10.1080/00140137108931246

[18] Colin G. Drury and P. Dawson. 1974. Human Factors Limitations in Fork-Lift Truck Performance. *Ergonomics* 17, 4 (1974), 447–456. https://doi.org/10.1080/00140137408931376

[19] Paul M. Fitts. 1954. The information capacity of the human motor system in controlling the amplitude of movement. *Journal of Experimental Psychology* 47, 6 (1954), 381–391. https://doi.org/10.1037/h0055392

[20] Sebastian Friston, Per Karlström, and Anthony Steed. 2016. The Effects of Low Latency on Pointing and Steering Tasks. *IEEE Transactions on Visualization and Computer Graphics* 22 (2016), 1605–1615. https://doi.org/10.1109/TVCG.2015.2446467

[21] Errol R. Hoffmann. 1992. Fitts' Law with transmission delay. *Ergonomics* 35, 1 (1992), 37–48. https://doi.org/10.1080/00140139208967796

[22] Errol R. Hoffmann. 1997. Movement time of right- and left-handers using their preferred and non-preferred hands. *International Journal of Industrial Ergonomics* 19, 1 (1997), 49 – 57. https://doi.org/10.1016/0169-8141(95)00092-5

[23] Errol R. Hoffmann. 2009. Review of models for restricted-path movements. *International Journal of Industrial Ergonomics* 39, 4 (2009), 578 – 589. https://doi.org/10.1016/j.ergon.2008.02.007 Special issue: Felicitating Colin G. Drury.

[24] Steven W. Keele and Michael I. Posner. 1968. Processing of visual feedback in rapid movements. *Journal of Experimental Psychology* 77, 1 (1968), 155–158. https://doi.org/10.1037/h0025754

[25] Masatomo Kobayashi and Takeo Igarashi. 2003. Considering the Direction of Cursor Movement for Efficient Traversal of Cascading Menus. In *Proceedings of the 16th Annual ACM Symposium on User Interface Software and Technology (UIST '03)*. ACM, New York, NY, USA, 91–94. https://doi.org/10.1145/964696.964706

[26] Sergey Kulikov, I. Scott MacKenzie, and Wolfgang Stuerzlinger. 2005. Measuring the Effective Parameters of Steering Motions. In *CHI '05 Extended Abstracts on Human Factors in Computing Systems (CHI EA '05)*. ACM, New York, NY, USA, 1569–1572. https://doi.org/10.1145/1056808.1056968

[27] Sergey Kulikov and Wolfgang Stuerzlinger. 2006. Targeted Steering Motions. In *CHI '06 Extended Abstracts on Human Factors in Computing Systems (CHI EA '06)*. ACM, New York, NY, USA, 983–988. https://doi.org/10.1145/1125451.1125640

[28] Gordon Kurtenbach and William Buxton. 1993. The Limits of Expert Performance Using Hierarchic Marking Menus. In *Proceedings of the INTERACT '93 and CHI '93 Conference on Human Factors in Computing Systems (CHI '93)*. ACM, New York, NY, USA, 482–487. https://doi.org/10.1145/169059.169426

[29] I. Scott MacKenzie and Colin Ware. 1993. Lag As a Determinant of Human Performance in Interactive Systems. In *Proceedings of the INTERACT '93 and CHI '93 Conference on Human Factors in Computing Systems (CHI '93)*. ACM, New York, NY, USA, 488–493. https://doi.org/10.1145/169059.169431

[30] M. Ali Montazer and Colin G. Drury. 1989. A test of the Beggs' model for self-paced movements. *Ergonomics* 32, 5 (1989), 497–511. https://doi.org/10.1080/00140138908966120

[31] M. A. Montazer, C. G. Drury, and M. H. Karwan. 1988. An optimization model for self-paced tracking on circular courses. *IEEE Transactions on Systems, Man, and Cybernetics* 18, 6 (Nov 1988), 908–916. https://doi.org/10.1109/21.23090

[32] M. Ali Montazer, Suresh K. Vyas, and Ronald N. Wentworth. 1987. A Study of Human Performance in a Sewing Task. *Proceedings of the Human Factors Society Annual Meeting* 31, 5 (1987), 590–594. https://doi.org/10.1177/154193128703100525

[33] Mathieu Nancel and Edward Lank. 2017. Modeling User Performance on Curved Constrained Paths. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems (CHI '17)*. ACM, New York, NY, USA, 244–254. https://doi.org/10.1145/3025453.3025951

[34] Robert Pastel. 2006. Measuring the Difficulty of Steering Through Corners. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '06)*. ACM, New York, NY, USA, 1087–1096. https://doi.org/10.1145/1124772.1124934

[35] Nicolas Rashevsky. 1959. Mathematical biophysics of automobile driving. *Bulletin of Mathematical Biophysics* 21, 4 (1959), 375–385. https://doi.org/10.1007/BF02477896

[36] Richard A. Schmidt, Howard N. Zelaznik, and James S. Frank. 1978. Sources of Inaccuracy in Rapid Movement. In *Information Processing in Motor Control and Learning*, George E. Stelmach (Ed.). Academic Press, 183–203. https://doi.org/10.1016/B978-0-12-665960-3.50014-1

[37] Ransalu Senanayake and Ravindra S. Goonetilleke. 2016. Pointing Device Performance in Steering Tasks. *Perceptual and Motor Skills* 122, 3 (2016), 886–910. https://doi.org/10.1177/0031512516649717

[38] Ransalu Senanayake, Errol R. Hoffmann, and Ravindra S. Goonetilleke. 2013. A model for combined targeting and tracking tasks in computer applications. *Experimental Brain Research* 231, 3 (01 Nov 2013), 367–379. https://doi.org/10.1007/s00221-013-3700-4

[39] Erum Tanvir, Andrea Bunt, Andy Cockburn, and Pourang Irani. 2011. Improving cascading menu selections with adaptive activation areas. *International Journal of Human-Computer Studies* 69, 11 (2011), 769 – 785. https://doi.org/10.1016/j.ijhcs.2011.06.005

[40] Erum Tanvir, Jonathan Cullen, Pourang Irani, and Andy Cockburn. 2008. AAMU: Adaptive Activation Area Menus for Improving Selection in Cascading Pull-down Menus. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '08)*. ACM, New York, NY, USA, 1381–1384. https://doi.org/10.1145/1357054.1357270

[41] Namal Thibbotuwawa, Ravindra S. Goonetilleke, and Errol R. Hoffmann. 2012. Constrained Path Tracking at Varying Angles in a Mouse Tracking Task. *Human Factors* 54, 1 (2012), 138–150. https://doi.org/10.1177/0018720811424743

[42] Namal Thibbotuwawa, Errol R. Hoffmann, and Ravindra S. Goonetilleke. 2012. Open-loop and feedback-controlled mouse cursor movements in linear paths. *Ergonomics* 55, 4 (2012), 476–488. https://doi.org/10.1080/00140139.2011.644587

[43] Charles E. Wright and Francis Lee. 2013. Issues Related to HCI Application of Fitts's Law. *Human-Computer Interaction* 28, 6 (2013), 548–578. https://doi.org/10.1080/07370024.2013.803873

[44] Shota Yamanaka and Homei Miyashita. 2016. Modeling the Steering Time Difference Between Narrowing and Widening Tunnels. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems (CHI '16)*. ACM, New York, NY, USA, 1846–1856. https://doi.org/10.1145/2858036.2858037

[45] Shota Yamanaka and Homei Miyashita. 2016. Scale Effects in the Steering Time Difference Between Narrowing and Widening Linear Tunnels. In *Proceedings of the 9th Nordic Conference on Human-Computer Interaction (NordiCHI '16)*. ACM, New York, NY, USA, Article 12, 10 pages. https://doi.org/10.1145/2971485.2971486

[46] Shota Yamanaka and Homei Miyashita. 2018. Paper-cutting operations using scissors in Drury's law tasks. *Applied Ergonomics* 69 (2018), 32–39. https://doi.org/10.1016/j.apergo.2017.12.018

[47] Shota Yamanaka, Wolfgang Stuerzlinger, and Homei Miyashita. 2017. Steering Through Sequential Linear Path Segments. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems (CHI '17)*. ACM, New York, NY, USA, 232–243. https://doi.org/10.1145/3025453.3025836

[48] Shota Yamanaka, Wolfgang Stuerzlinger, and Homei Miyashita. 2018. Steering Through Successive Objects. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems (CHI '18)*. ACM, New York, NY, USA, Article 603, 13 pages. https://doi.org/10.1145/3173574.3174177

[49] Shumin Zhai, Johnny Accot, and Rogier Woltjer. 2004. Human Action Laws in Electronic Virtual Worlds: An Empirical Study of Path Steering Performance in VR. *Presence: Teleoper. Virtual Environ.* 13, 2 (April 2004), 113–127. https://doi.org/10.1162/1054746041382393

[50] Shumin Zhai, Stéphane Conversy, Michel Beaudouin-Lafon, and Yves Guiard. 2003. Human On-line Response to Target Expansion. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '03)*. ACM, New York, NY, USA, 177–184. https://doi.org/10.1145/642611.642644

[51] Shumin Zhai, Jing Kong, and Xiangshi Ren. 2004. Speed-accuracy tradeoff in Fitts' law tasks: on the equivalency of actual and nominal pointing precision. *International Journal of Human-Computer Studies* 61, 6 (2004), 823–856. https://doi.org/10.1016/j.ijhcs.2004.09.007

[52] Shengdong Zhao, Maneesh Agrawala, and Ken Hinckley. 2006. Zone and Polygon Menus: Using Relative Position to Increase the Breadth of Multi-stroke Marking Menus. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '06)*. ACM, New York, NY, USA, 1077–1086. https://doi.org/10.1145/1124772.1124933

[53] Shengdong Zhao and Ravin Balakrishnan. 2004. Simple vs. Compound Mark Hierarchical Marking Menus. In *Proceedings of the 17th Annual ACM Symposium on User Interface Software and Technology (UIST '04)*. ACM, New York, NY, USA, 33–42. https://doi.org/10.1145/1029632.1029639

[54] Xiaolei Zhou, Xiang Cao, and Xiangshi Ren. 2009. Speed-Accuracy Tradeoff in Trajectory-Based Tasks with Temporal Constraint. In *Human-Computer Interaction – INTERACT 2009*. Springer Berlin Heidelberg, Berlin, Heidelberg, 906–919.

[55] Xiaolei Zhou and Xiangshi Ren. 2010. An investigation of subjective operational biases in steering tasks evaluation. *Behaviour & Information Technology* 29, 2 (2010), 125–135. https://doi.org/10.1080/01449290701773701

[56] Xiaolei Zhou, Xiangshi Ren, and Hue YUi. 2008. Effect of Start Position on Human Performance in Steering Tasks. In *2008 International Conference on Computer Science and Software Engineering*, Vol. 2. 1098–1101. https://doi.org/10.1109/CSSE.2008.1310