Servo-Gaussian Model to Predict Success Rates in Manual Tracking: Path Steering and Pursuit of 1D Moving Target

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

Hiroki Usuba, Haruki Takahashi, Homei Miyashita
Meiji University, Nakano-ku, Tokyo, Japan
{usuba, haruki}@meiji.ac.jp, homei@homei.com

ABSTRACT
We propose a Servo-Gaussian model to predict success rates in continuous manual tracking tasks. Two tasks were conducted to validate this model: path steering and pursuit of a 1D moving target. We hypothesized that (1) hand movements follow the servo-mechanism model, (2) submovement endpoints form a bivariate Gaussian distribution, thus enabling us to predict the success rate at which a submovement endpoint falls inside the tolerance, and (3) the success rate for a whole trial can be predicted if the number of submovements is known. The cross-validation showed $R^2 > 0.92$ and $MAE < 4.9\%$ for steering and $R^2 > 0.95$ and $MAE < 6.5\%$ for pursuit tasks. These results demonstrate that our proposed model delivers high prediction accuracy even for unknown datasets.

Author Keywords
Servo-mechanism model; steering law; manual tracking; moving targets; success rate prediction.

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

INTRODUCTION
In interactive systems such as drawing software and video games, continuous visual feedback and corresponding hand-movement corrections are required to accomplish a task. If the visuo-motor coordination is incorrectly controlled, undesired outcomes arise, such as painting outside the intended area or a player character being damaged. However, even if users try to carefully perform a task, it is inevitable that they will not always finish the task perfectly. Thus, understanding user performance in terms of success rates has attracted the interest of HCI researchers [33, 37, 61, 68].

In this paper, we propose a *Servo-Gaussian model* to predict success rates in manual tracking tasks. We focus on two tasks: path steering and pursuit of a 1D moving target. The former is a well-studied paradigm: the steering law to predict the movement time MT or speed V to pass through a path [1, 17, 51]. However, as yet there is no established success rate prediction model based on the path length and width. Manual pursuit of a moving target is required in hand-gesturing UIs [13], video games, and camera controls. For example, in first-person shooter games, a player may need to continuously shoot a machine gun at a moving tank, or in rhythm games, a player slides a finger inside a target window that moves horizontally at a constant pace (e.g., *Arcaea*). Panning a camera to capture a moving target (walking person, flying ball, etc.) is also a task found in daily activities using smartphones.

The key idea of our model is based on the servo-mechanism model [14, 59]. That is, an operator views (i.e., samples) the cursor position relative to a path/target and then determines which position to aim at in the next submovement. S/he repeatedly performs this action until the task ends. A submovement is performed in an open-loop manner, so no changes of speed or direction are made during a sampling period. Because submovement endpoints would distribute normally [5], the success rate of a submovement can be predicted if the distribution parameters (μ and σ) are known. To successfully accomplish a task, the user must succeed in moving the cursor inside the tolerance every submovement until the trial ends. Hence, assuming that all successes are independent of each other, we obtain the overall success rate for a whole trial as the power function of a single success rate.

Adopting the servo-mechanism model to steering tasks was done by Drury to predict the MT and V, who assumed that a steering action consists of repetitions of ballistic submovements [17]. Also, for pursuit tasks, users have to anticipate the target position a short period later (i.e., sampling interval), and thus, the pursuit tracking should consist of repetitions of feedforward submovements. Readers can refer to [16, 20] for more details on the nature of the servo-mechanism model.

Although success-rate prediction has been examined in recent works, the focus has mainly been limited to target selection tasks [8, 31, 33, 63]. However, interactive systems require different operational styles depending on the application, as mentioned above. If we can overcome this limitation by developing quantitative models, it will contribute to (e.g.) the design of GUI components such as cascaded menus by using the path-steering success-rate model or adjustment of the difficulty level to survive a stage by changing an enemy's speed.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

UIST'20, October 20–23, 2020, Minneapolis, MN, USA

© 2020 Copyright held by the owner/author(s). Publication rights licensed to ACM.

ISBN 978-1-4503-6708-0/20/04...$15.00
DOI: https://doi.org/10.1145/3313831.XXXXXXX

While the two tasks have differences, our experimental results showed a fairly high prediction accuracy, which demonstrates the generalizability of our model. Our contributions include:

- We derived a Servo-Gaussian model to predict success rates in tracking tasks, which require continuous and comparatively longer-term tracking operations than target selection.
- We conducted two user studies to investigate the validity of our model empirically. The results showed that our model could predict the success rates with $R^2 > 0.92$ and $MAE <$ 4.9% for the steering task and $R^2 > 0.95$ and $MAE < 6.5\%$ for the pursuit task through shuffle-split cross-validation using various sizes of training and test datasets.

RELATED WORK

Success or Error Rate Prediction Models for Pointing

For pointing to a static target, a well-known model to predict the MT is Fitts' law [21]. Typically, in a user study, participants are asked to perform "as quickly and accurately as possible" [42]. However, the balance could change towards speed or accuracy [22, 70]. If users aim for a target more rapidly, the error rate increases, and vice versa [70]. This *speed-accuracy tradeoff* is valid in various psychological tasks [60].

In addition to MT, predictive models based on success rates (or error rates) have been derived [43, 61]. For more dynamical user interfaces, deriving models for capturing moving targets has been tackled. Related works have shown that the MT increases as a target's speed V increases [24, 26, 34]. For error rate prediction, naturally, as the V increases, users are more likely to fail a task [30, 32, 33, 39, 50].

Laws of Path Steering Tasks

Rashevsky [51, 52], Drury [17], and Accot and Zhai [1] proposed mathematically similar models to predict the movement speed V to pass through a constant-width (W) path:

$$V = aW. \tag{1}$$

Hereafter, italic lowercase letters *a*–*n* refer to empirical constants. Another form of the steering law is to predict the MT to pass through a path whose length is A [1, 17]:

$$MT = b + c(A/W). \tag{2}$$

These models hold when W is not extremely wide [27, 56]. Similar to pointing, when users try to pass through a path more quickly, they tend to deviate from the path more [72, 73].

Although Drury's derivation to predict V and MT was based on a probabilistic model [17, 20], success rate prediction models based on A and W have never been developed. If we can derive such a model, it would be of benefit to HCI and UI designs, similarly to the claims in related work [9, 31, 33, 61]. In addition, because the steering law is valid for many human-machine interactions, including outside-GUI tasks, our model will potentially contribute to task-difficulty estimation for (e.g.) remote robot controlling [11, 28].

Servo-Mechanism Model and Sampling Interval

In pointing and steering tasks, when an operator perceives the visual feedback on the current position of a probe (hand, cursor, etc.) relative to the tolerance of the target or path, s/he determines the speed and direction of the next submovement. S/he repeatedly performs this action until the task ends. The servo-mechanism model (or an intermittent-acting model [17, 35, 43]) assumes this loop is based on a discrete manner [16, 17]. Hence, even though a complete trial of pointing or steering needs a closed-loop action, each submovement is considered ballistic. This means that once the operator determines the speed and direction of the next submovement and executes it, s/he cannot change these parameters during the period of the *sampling interval* (i.e., the sum of corrective reaction time and the system's latency).

The servo-mechanism model has been used in various derivations of Fitts' [16, 43] and steering laws [17, 45]. In these studies, the corrective reaction time from visual feedback to execution of the next submovement is dealt with as a constant, such as 0.19 to 0.26 sec [35] and shorter than 0.29 sec [7]. Note that these reports are based on a mean result by a pool of participants. Lin and Hsu conducted circular steering tasks and reported that the corrective reaction times by each participant ranged from 0.087 to 0.441 sec with the mean of 0.273 sec [41]. Thus, using the corrective reaction time as a constant (such as 0.26 sec) is a reasonable approximation for performance modeling. On the basis of these related studies, in our data analyses, we first examine the model fitness by using 0.26 sec for the corrective reaction time.

Spatial Variability in a Submovement

There is a consensus that the endpoints of ballistic movements follow a bivariate normal distribution [5, 6, 29, 40, 53]. According to Howarth [29] and Beggs et al. [5, 6], the endpoint variability that is measured as the standard deviation in ballistic aiming movements on the x-axis, σ_x, is modeled as

$$\sigma_\text{x}^2 = \sigma_{0\text{x}}^2 + (\sigma_{\theta\text{x}} D)^2, \tag{3}$$

where σ_{0x} is the *uncontrollable hand tremor factor*, $\sigma_{\theta\text{x}}$ is the angular accuracy relative to the movement distance, and D is the movement distance in a submovement. The same formulation is valid for the perpendicular variability σ_y:

$$\sigma_\text{y}^2 = \sigma_{0\text{y}}^2 + (\sigma_{\theta\text{y}} D)^2. \tag{4}$$

These formulations indicate that the endpoint variabilities increase as the operator aims for a farther position in a submovement. Because the hand tremor constants tend to be small [41, 44, 53, 69], these models are consistent with reports on the relationship being linear [4, 23, 53]:

$$\sigma_\text{x} = \sigma_{\theta\text{x}} D, \;\text{ and }\; \sigma_\text{y} = \sigma_{\theta\text{y}} D. \tag{5}$$

FORMULATING PROBLEM AND MODEL 1: STEERING THROUGH A CONSTANT-WIDTH LINEAR PATH

Revisiting a Derivation of the Steering Law

When Drury derived the steering law, he used the servo-mechanism model with a constant sampling interval T_sampling [17]. He hypothesized that (1) submovement endpoints perpendicular to the movement direction normally distribute with the mean being located at the path center (Figure 1) and (2)

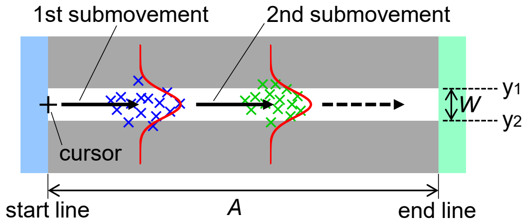


Figure 1. Hypothesized submovement distribution in a steering task. Blue and green Xs indicate the endpoints of first and second submovements, respectively, which normally distribute on the y-axis (red curves).

the variability σ_y is proportional to W:

$$\sigma_y = dW. \tag{6}$$

In steering law experiments, participants try to avoid deviating from the path; however, because they have to pass through the path quickly, it is better to aim for a farther position in every submovement. This explains the speed-accuracy tradeoff: if D increases (i.e., there is faster movement), the variability σ_y also increases (Equation 5), and thus the error rate will increase (Figure 1).

By replacing σ_y in Equation 6 with Equation 5, we obtain

$$\sigma_{\theta y} D = dW, \text{ or } D = eW \text{ (let } e = d/\sigma_{\theta y}). \tag{7}$$

Hence, as the path width increases, users aim for a farther position in a submovement (i.e., they move faster). As the mean speed V for this submovement is defined as the distance divided by the time needed, we have

$$V = D/T_{\text{sampling}}. \tag{8}$$

Replacing D in Equation 8 with Equation 7, we have $V = eW/T_{\text{sampling}}$. Assuming that T_{sampling} is mostly constant with one person [41], Drury derived the "$V = aW$" form of the steering law (Equation 1). Then, MT is defined as the distance divided by the speed ($MT = A/V = A/[aW]$), which is equivalent to Equation 2 with no intercept.

Modeling the Success Rate of a Complete Steering Trial

The key idea of our proposed model is that users have to sequentially succeed in positioning every submovement's endpoint inside the path boundaries until the path end. Otherwise, if even one of the submovements' endpoints falls outside the path, this trial is judged as a failure. Because the probability of a submovement endpoint falling in the path can be predicted, and because the steering law assumes a submovement's success is independent of the others [20, 45], the overall success rate for a whole trial is computed by multiplying the single success rate iteratively. The number of iterations is equivalent to the number of submovements to accomplish the task, and thus this number is computed by the MT divided by T_{sampling}.

First, by replacing D in Equation 4 with Equation 7, we have

$$\sigma_y^2 = \sigma_{0y}^2 + (\sigma_{\theta y} eW)^2. \tag{9}$$

Because σ_{0y}, $\sigma_{\theta y}$, and e are constants, we have

$$\sigma_y = \sqrt{f + gW^2}. \tag{10}$$

When the square of the tremor factor is small ($f = \sigma_{0y}^2 \approx 0$), this becomes $\sigma_y = \sqrt{g}W$ (proportional, Equation 6).

Formally speaking, a submovement's endpoint on the y-axis is a random variable Y that follows a normal distribution:

$$Y \sim N(\mu_y, \sigma_y^2). \tag{11}$$

The probability P of a submovement's endpoint falling inside the path boundaries (upper: y_1, lower: y_2) is derived as

$$P(y_1 \le Y \le y_2) = \frac{1}{2}\left[\operatorname{erf}\left(\frac{y_2 - \mu_y}{\sigma_y\sqrt{2}}\right) - \operatorname{erf}\left(\frac{y_1 - \mu_y}{\sigma_y\sqrt{2}}\right)\right], \tag{12}$$

where erf(x) is the Gauss error function. If we define the origin of the y-coordinate as the path center, μ_y is $\approx$0, as described above. By this definition, we have ($y_1 = -W/2$) and ($y_2 = +W/2$), and thus Equation 12 is simplified as

$$P(-W/2 \le Y \le W/2) = \operatorname{erf}\left[W/(2\sqrt{2}\sigma_y)\right]. \tag{13}$$

The number of submovements for a complete trial N_{submove} is

$$N_{\text{submove}} = MT/T_{\text{sampling}}, \tag{14}$$

where T_{sampling} consists of an operator's corrective reaction time and the system's latency [28]. Assuming that the corrective reaction time is constant (e.g., 0.26 sec), and because the system's latency can be directly measured (e.g., 0.05 sec), N_{submove} can be calculated if the MT to steer through a path is obtained. Fortunately, even for a new path condition, we can predict the MT by the steering law. Note that N_{submove} does not need to be an integer because a trial can finish during a submovement. The success rate for a whole trial, $P(S)$, is

$$\begin{aligned} P(S) &= [P(-W/2 \le Y \le W/2)]^{N_{\text{submove}}} \\ &= \left[\operatorname{erf}\left(W/(2\sqrt{2}\sigma_y)\right)\right]^{(MT/T_{\text{sampling}})}, \end{aligned} \tag{15}$$

where σ_y is computed by Equation 10. The final step to obtain $P(S)$ is to measure the remaining constants of f and g in Equation 10. However, computing these values from the cursor trajectory data is challenging, as detecting the beginning and ending of each submovement is not a naive problem. For pointing, users perform a quick acceleration and deceleration in the ballistic phase and then exhibit several small speed peaks (e.g., [49]). Hence, segmenting the cursor trajectory into each submovement is possible [38, 46].

In contrast, for steering, the steering law assumes that the cursor speed is stable for a fixed path width (Equation 1, $V = aW$), and thus accelerations/decelerations are not clearly observed [55, 65]. Although we apply 0.26 sec as the corrective reaction time, this is the *duration* of a submovement; we cannot detect the spatial positions of the beginning and ending of each submovement.

Another approach would be computing σ_y for the entire trajectory data in a whole trial, i.e., the *SD* of the cursor's y-positions throughout one steering trial [36, 72, 73]. However, the relationship between the σ_y for a whole trial in those studies and the σ_y for each submovement used in our model is unclear.

Therefore, we compute f and g in Equation 10 by parameter optimization. That is, if our model appropriately expresses the success rates, the fitting process between Equation 15 and the observed success rates gives "likely true" values for f and g, which can be used to predict the success rates for new path conditions. A similar approach has been used to indirectly measure parameters in steering tasks, e.g., to compute the corrective reaction time [18].

Discussion 1: Strengths and Limitations of the Model

The positive aspect of our approach is that we need only MT and success rate data, which are commonly measured in steering experiments. This eases the efforts by researchers and UI developers to log the probe trajectory, submovement segmentation, etc. For example, in some steering tasks in the HCI and human-robot interaction fields, it is difficult to measure fine-grained trajectory, such as moving a robot [11, 28], which requires additional costs to track the robot location[1].

One concern is that, while the proposed model builds on related studies, it uses several approximations and simplifications. For example, we assume $\mu_y = 0$ according to [17], but this might have a bias from the path center. Also, we assume that the corrective reaction time and the system's latency are almost constant, which yields T_{sampling} as constant. Yet, the former varies within/between users [41], and the latter has variability [10]. The steering law assumes that the speed is constant for a fixed W, but this is empirically not true [55, 67]. Despite these inconsistencies, much prior literature has shown that the steering law holds to the actual data.

We hypothesized that the success rate for a whole trial can be expressed as the power of each submovement's success rate. This requires that every submovement's success rate be independent from each other. However, because a hand movement has an inertia towards the path end, if a previous submovement's endpoint is located close to the center of the path, the next endpoint will likely fall in the path. Thus, one might suspect that the behaviors between consecutive submovements are not independent. Still, this simplification has also been made to validate the steering law for linear and circular paths [20, 45]. This justifies our use of the power function to compute the success rate of a whole trial for simplicity.

In summary, while our model development is established on existing assumptions, the reliability of the final model needs empirical validation through a user study to determine how well the model works with actual data.

FORMULATING PROBLEM AND MODEL 2: PURSUIT OF A 1D MOVING TARGET

The task is to keep the cursor inside a target, which has a width W and moves at a constant speed V rightwards, for a movement distance A (Figure 2). As the success rate to *point to* a moving target has been studied before [32, 33], our study does not concern itself with the success rate to capture the target. Rather, we focus on the behavior after that: how successfully users keep the cursor inside the target for A. Thus, when we begin to judge the success, the cursor has already moved inside the target. For this formulation, we prepare

[1]Robot controlling along a path is also modeled by the steering law based on the servo-mechanism model [28].

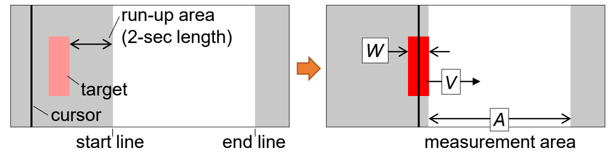


Figure 2. 1D pursuit task. Clicking on the target starts a trial. Users try to keep the cursor inside the target during the measurement.

a *run-up area* before measuring the cursor position, which allows users to adjust the cursor speed appropriately for V.

The basic idea is the same as for steering. Once a submovement is executed with a predetermined speed, this cannot be changed for a sampling interval. Then, as a submovement's success rate can be computed, the success rate for a whole trial can be predicted by iteratively multiplying the single success rate until the target reaches the end line.

Revisiting the Model on Pointing to a Moving Target

We adopt Huang et al.'s success rate model for a 1D moving target [32]. In their study, the cursor is initially located a certain distance from the target with width W. Once a trial begins, the target moves at a constant speed V horizontally. Participants must point to a target and click the mouse button. They assumed that the endpoint on the x-axis when clicking is a random variable X that follows a normal distribution, as

$$X \sim N(\mu_x, \sigma_x^2). \tag{16}$$

Then, they hypothesized that the X is the sum of the three components that follow normal distributions:

$$X = X_a + X_m + X_s \sim N(\mu_x, \sigma_x^2), \tag{17}$$

where $X_a \sim N(\mu_a, \sigma_a^2)$, $X_m \sim N(\mu_m, \sigma_m^2)$, and $X_s \sim N(\mu_s, \sigma_s^2)$ represent the precision of the pointing device, target velocity, and target width, respectively. X_a is the absolute precision uncertainty of a motor system that includes the input device, which is independent from V and W. Thus, μ_a and σ_a are constants. X_m depends on the uncertainty caused by the target motion (i.e., V). A simple assumption is made that μ_m and σ_m are proportional to the V. X_s depends on the desired precision of hitting the target and the corresponding action speed. This variability is controlled by the speed-accuracy tradeoff, and thus the μ_m and σ_m are assumed to be proportional to W.

Huang et al. confirmed that σ_x increased with V, but the positive coefficient decreased as W increased. This interaction suggests that X_m and X_s are dependent on each other. Thus, they set the covariance of X_m and X_s as a term of V/W. By summing the three normal distribution variables, we have

$$\mu_x = \mu_a + \mu_m + \mu_s = h + iV + jW, \tag{18}$$

$$\begin{aligned} \sigma_x &= \sqrt{\sigma_a^2 + \sigma_m^2 + \sigma_s^2 + \mathrm{cov}(X_m, X_s)} \\ &= \sqrt{k + lV^2 + mW^2 + n(V/W)}. \end{aligned} \tag{19}$$

If we define the target center as $x = 0$, $x_1 = -W/2$, and $x_2 = +W/2$, the success rate is predicted as

$$P(x_1 \leq X \leq x_2) = \frac{1}{2}\left[\mathrm{erf}\left(\frac{W/2 - \mu_x}{\sigma_x\sqrt{2}}\right) - \mathrm{erf}\left(\frac{-W/2 - \mu_x}{\sigma_x\sqrt{2}}\right)\right]. \tag{20}$$

This does not include the term of the initial target distance, as it does not affect the μ_x and σ_x [32, 33].

Modeling the Success Rate of a Whole Pursuit Trial

To successfully accomplish a task, operators have to position the cursor inside the target in every frame (i.e., system's sampling timing) until the target reaches the end line. Yet, according to the servo-mechanism model, the cursor speed cannot be dynamically changed during an operator's sampling interval. Thus, if the endpoints of two consecutive submovements are inside the target, it is assumed that the cursor between these two sampled positions also kept on target (Figure 3).

We assume that each submovement in a pursuit task is a moving-target pointing task without manual clicking (or the system automatically clicks every frame) with zero initial target distance. That is, the cursor is already inside the target when a measurement begins, and operators aim for a target position at the next submovement's end timing. Operators know the appropriate cursor speed before a measurement begins, because there is sufficient time to see the target speed in the run-up area. Such a prediction of target position can be performed if a sufficient viewing time is provided [30, 49].

Given that the target speed is V and the sampling interval is T_{sampling}, the distance needed for a submovement D is

$$D = V \times T_{\text{sampling}}. \tag{21}$$

The number of submovements N_{submove} to finish a task is

$$N_{\text{submove}} = A/D = A/(V \times T_{\text{sampling}}) \tag{22}$$

and the success rate for a whole pursuit trial is

$$\begin{aligned} P(S) = & \ [P(-W/2 \le X \le W/2)]^{N_{\text{submove}}} \\ = & \left(\frac{1}{2} \left[\text{erf}\left(\frac{x_2 - \mu_x}{\sigma_x \sqrt{2}} \right) - \text{erf}\left(\frac{x_1 - \mu_x}{\sigma_x \sqrt{2}} \right) \right] \right)^{[A/(V \cdot T_{\text{sampling}})]}. \end{aligned} \tag{23}$$

Because the MT for a trial is A/V, Equation 23 has the same power index as in Equation 15.

Discussion 2: Strengths and Limitations of the Model

While we adopt the success rate model of a 1D moving target [32] for predicting the success rate of a submovement, we do not claim that a whole pursuit task and the repetition of moving-target pointing are the same task. There are various differences in these tasks. For example, clicking timing can be determined by operators in moving-target pointing, which will affect μ_x. Rather, we hypothesize that if the endpoint uncertainty model in [32] is also valid for zero-initial-distance moving-target capturing, the whole success rate of the target pursuit can also be predicted by the power function of a success rate on a submovement.

Similarly to steering tasks, our model does not need to record the cursor trajectory. Furthermore, in contrast to the steering model, only the success rate is needed as a dependent variable, as the MT is fixed. This difference is distinguished as *self-paced tracking* for path steering tasks or *externally paced tracking* for pursuit of a moving target [17, 20]. In spite of the differences in task requirements (e.g., self- or externally paced, and the task constraints given for the y- or x-axes), we hypothesize that the success rate for a whole trial can be predicted for both tasks. If so, it will demonstrate a certain generalizability of our Servo-Gaussian model.

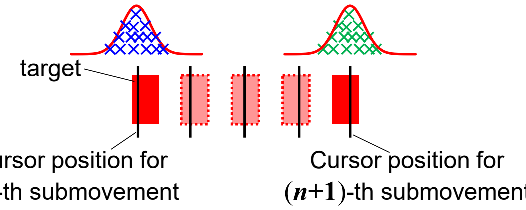


Figure 3. Hypothesized submovement distribution in a pursuit task. Blue and green Xs show the endpoints of n-th and $(n+1)$-th submovements on the x-axis, respectively, which form normal distributions.

We again used several simplifications and assumptions. For example, (1) while Huang et al. confirmed that the initial distance did not affect the endpoint distributions, their model validity has not been empirically shown for zero-initial-distance and no-clicking conditions, and (2) for simplicity in modeling, we assumed that if two consecutive submovements' endpoints fell inside the target, the cursor between those two sampled points kept inside the target. Yet, a cursor could deviate from a target if the speed during a submovement is not perfectly constant. The reliability of our model, again, needs an empirical test.

STUDY 1: CONSTRAINED PATH STEERING

Participants

Five students from a local university participated in this study (all males; ages: M = 21.4, SD = 1.20 years). All were right-handed. Three of them were daily mouse users, and the remaining two could also use mice well.

Apparatus

We used a laptop PC (Dell, 2.8 GHz $\times$ 4 cores, 8 GB of RAM, Windows 10). The display was manufactured by PHILIPS (698.1 $\times$ 392.7 mm, 2560 $\times$ 1440 pixels; 4-msec response time) and 60 Hz refresh. The experimental system was implemented with Hot Soup Processor 3.5 and used in full-screen mode. The system reads and processes input approximately 500 times per second.

The input device was an optical mouse (Logitech, 1000 Hz, 1000 dpi) that had a 2-m cable. The cursor speed was set to the default; the slider in Control Panel was set to the center. The pointer acceleration (*Enhance pointer precision*) was disabled to set the control-display gain to constant for avoiding unintentional speeding-up/down during operations. Because the goal of Study 1 was to validate our model, the acceleration was unneeded and turned off. As the steering law holds both with [2, 64] and without [54, 56] pointer acceleration, our model validity should not be strongly affected by this setting. We used a large mousepad (60 cm $\times$ 30 cm) and participants were asked not to clutch the mouse during a trial.

To measure the cursor latency, we adopted Müller et al.'s method [47]. The mouse was hit with a hard object at high speed, and we counted the number of frames from when the mouse stopped to when the cursor stopped on a movie recorded by a Casio Exilim EX-ZR4000WE camera at 1000 fps. We repeated this action 30 times, and the average latency was 40.5 msec ($SD = 13.5$). This was shorter than typical mouse-to-display latencies (55 to 82 msec [10]), so we assume it would not have a significant negative effect on user performance.

Task

The task was to click on the left blue start area, move along the white path, and then click on the right green end area (see Figure 1). The crosshair cursor left a blue trace in the start and end areas, a green trace in the path, and a red trace in the out-of-path areas to indicate any deviation from the path, which played a friction sound. 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.

Participants were asked to complete each task as quickly and accurately as possible. If the cursor deviated from the path, they were asked to return to the path immediately and then continue the task to the end line. A click was needed in the green end area, but as this area was sufficiently long, no precise pointing was required [55].

Design

Study 1 was a 4×5 within-subjects design. The two independent variables were path length A (400, 600, 800, and 1000 pixels) and width W (8, 11, 15, 20, and 26 pixels). The steering law difficulty (A/W) ranged from 15.4 to 125; these values were higher than 10 so that the W could restrict the speed [57].

In a previous study using mice with $A = 480$–640 and $W = 15$–45 pixels, the error rates ranged from 86 to 99% with $M = 93\%$ (no error-accepting delay condition in [64]). Thus, a model that predicts the error rate being always ~93% has a high prediction accuracy. To avoid obtaining such a model, we chose comparatively more difficult conditions.

We measured two dependent variables: MT and success rate. MT was the time from when the cursor crossed the start line to when it crossed the end line. A steering error was flagged for a trial when one or more deviations were observed, and the success rate was the percentage of the number of trials without error(s) divided by the total number of trials for each $A \times W$ condition.

Procedure

One *session* consisted of four repetitions of a random order of 20 conditions ($= 4_A \times 5_W$). Participants performed seven sessions for data collection following ten trials that were randomly selected from the 20 conditions as practice. In total, we recorded $4_A \times 5_W \times 4_{\text{repetitions}} \times 7_{\text{sessions}} \times 5_{\text{participants}} = 2800$ data points. This task took about 45 min per participant.

Results

We removed 18 outlier data points (0.64%) for trials where the MT was greater than 3σ from the average for each $A \times W \times$ participant condition; i.e., trials done extremely slowly or rapidly. After that, we analyzed the MT data by repeated-measures ANOVA. If a main effect of A or W was found, we performed pairwise comparisons with Bonferroni correction as the p-value adjustment method. For the F statistic, degrees

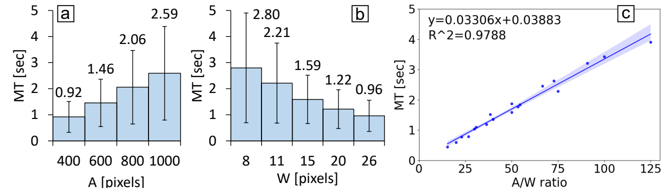


Figure 4. Main effects of (a) A and (b) W on MT with error bars showing 95% CIs. (c) Steering law regression with a 95% CI.

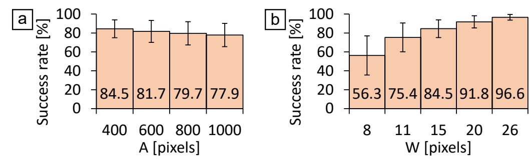


Figure 5. Main effects of A and W on success rate. Error bars: 95% CIs.

of freedom were corrected by the Greenhouse-Geisser method if Mauchly's sphericity assumption was violated ($\alpha = 0.05$).

For success rates, Shapiro-Wilk tests ($\alpha = 0.05$) showed that the data did not normally distribute in 5 out of 20 conditions. Hence, we used non-parametric ANOVAs with Aligned Rank Transform (*ART*) [62] with Tukey's p-value adjustment method for pairwise comparisons.

Movement Time

For the MT analysis, we used error-free data to maintain consistency with previous studies on the steering law [3, 67]. The mean MT was 1.756 sec. We found significant main effects of A ($F_{1.013,4.052} = 13.62$, $p < 0.05$, $\eta_p^2 = 0.77$) and W ($F_{1.006,4.025} = 10.23$, $p < 0.05$, $\eta_p^2 = 0.72$). The interaction of $A \times W$ was significant ($F_{1.283,5.133} = 7.075$, $p < 0.05$, $\eta_p^2 = 0.64$). The MTs increased as A increased (Figure 4a) and as W decreased (b), but pair-wise comparisons showed no significant differences in any pairs for both A and W ($p > 0.05$), possibly because of the small number of participants. The steering law of the MT form showed $R^2 = 0.98$ (Figure 4c).

Success Rate

The mean success rate was 80.1%. We found a significant main effect of W ($F_{4,16} = 61.83$, $p < 0.001$, $\eta_p^2 = 0.94$) but not for A ($F_{3,12} = 2.213$, $p = 0.1393$, $\eta_p^2 = 0.36$). Pairwise comparisons showed $p < 0.001$ for all W pairs except $W = 11$ and 15 pixels ($p < 0.05$) and $W = 20$ and 26 pixels ($p = 0.10$). The interaction of $A \times W$ was not significant ($F_{12,48} = 1.120$, $p = 0.3670$, $\eta_p^2 = 0.22$). While the success rate differences due to A were not statistically significant, we observed that the success rates monotonically decreased as A increased (Figure 5a) and as W decreased (b).

Prediction Accuracy and Discussion of Study 1

Model Fit to All Known Data

As the system's latency was 0.0405 sec on average, and as corrective reaction time is heuristically given as 0.26 sec, we set the sampling interval T_{sampling} to 0.3 sec as an approximation. Again, the MT in our model is computed by the steering law. For example, in a condition of $A = 800$ and $W = 15$ pixels, the actual MT was 1.770 sec, but that predicted by the regression expression ($MT = 0.03883 + 0.03306(A/W)$, see

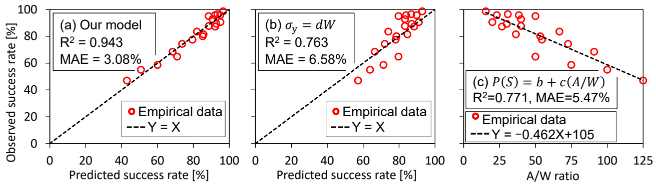


Figure 6. Fitness for three candidate models. (a) Proposed model. (b) Applying $\sigma_y = dW$ to proposed model. (c) A model assuming the success rate decreases for a higher steering difficulty (A/W). Note: only (c) does not include the predicted vs. observed success rates; this shows the correlation between the success rates and task difficulty.

Table 1. Estimated constants for proposed model in Study 1.

b	c	f	g
0.03883	0.03306	1.745	0.04283

Figure 4c) was 1.802 sec. Using such MT values computed from the steering law is needed for predicting the success rates for unknown datasets (described later).

By applying these parameters, Equation 15 computes the success rates for each $A \times W$ condition. We used the `curve_fit` function provided by the `scipy.optimize` library in Python for parameter optimization so as to maximize the correlation between the predicted and observed success rates. For $N = 20$ conditions, the coefficients were computed as listed in Table 1.

As shown in Figure 6a, our proposed model showed $R^2 = 0.943$ between predicted vs. observed success rates with the mean absolute error MAE of 3.08%. Note that the *Wasserstein distance* and *Hellinger distance* used in related work [31, 33, 71] are not used as accuracy indicators here, as we do not directly measure the endpoint distributions.

As no models that compute the success rates in % have been proposed before, for comparison we discuss two other possible model formulations. One is to use Equation 6 ($\sigma_y = dW$) for the endpoint variability in our proposed model of Equation 15. The d in this model was computed as 0.2428, and the result shows that the prediction accuracy was lower than our model (see Figure 6b). This indicates that the hand tremor factor is needed to more accurately predict the success rates. The other model for comparison is to assume that as the steering law difficulty (A/W) increases, it is less likely users will succeed in a task. However, this yielded a lower fit than our model, as shown in Figure 6c.

Prediction Accuracy to Unknown Data

The coefficients b, c, f, and g in Table 1 were optimized to maximize R^2 by using the observed success rates. Thus, the results ($R^2 = 0.943$ and $MAE = 3.08\%$) show the fits to the *known* data. However, model validity, or the "prediction" accuracy, should be judged for the future (unknown) data. For example, by using the optimized coefficients in Table 1, how accurately can we predict the success rate under an untested condition such as $A = 500$ and $W = 18$ pixels?

To investigate this, we ran shuffle-split cross-validation for the candidate models. When the ratio of (training : test) is (70% : 30%), the steps are as follows. (1) Randomly select 70% of the data points (= 14 among 20 conditions) for training, (2) regress the steering law $MT = b + c(A/W)$ to obtain coefficients b and c, (3) optimize parameters f and g to maximize the fit between predicted and observed success rates for those 14 data points, (4) predict success rates for the remaining (test) six data points by using those four coefficients and the test conditions' A and W values, and (5) check the R^2 and MAE between the predicted and actual success rates of the six test data points. To handle the sampling randomness when splitting the training and test datasets, we performed this process over 100 iterations.

Table 2. Average fitness by cross-validation in Study 1. The R^2 and MAE values are averaged over 100 iterations for each ratio of (training : test).

ratio [%]	80 : 20	70 : 30	60 : 40	50 : 50
R^2	0.9226	0.9309	0.9341	0.9352
MAE [%]	3.676	3.667	4.832	4.689

The results of the cross-validation are shown in Table 2. Because the prediction performance can change depending on the sizes of the training and test datasets, we report four ratios. However, we do not include the case of (training : test) = (90% : 10%) because the number of test data points was only 2; R^2 was always 1 and thus is meaningless for cross-validation. This result demonstrates that our model, which uses four free parameters, does not excessively fit to the known data (i.e., overfitting), because the prediction accuracy did not sharply drop even though the dataset sizes changed.

The results show that the prediction accuracy in R^2 increased as the size of the training dataset decreased. This is possibly because the accuracy depends on whether data points that are difficult to predict are included in the test dataset. For example, in Figure 6a, the most biased data point from the $Y = X$ line shows 95.0% for the actual success rate, but our model predicted it as 88.5%. This data point drops the prediction score if it is included in the test dataset. However, if it is included in the training dataset, it may negatively affect the parameter optimization. Hence, it is difficult to analyze the effect of this data-splitting process on the prediction accuracy.

As a comparison, the "$\sigma_y = dW$" model (Figure 6b) showed mean R^2 ranges from 0.71 to 0.74 and mean MAE ranges from 7.5 to 8.4% depending on the training-test ratio over 100 iterations, the same as our proposed model. Similarly, the steering-law difficulty model to predict the success rate (Figure 6c) showed mean R^2 ranges from 0.71 to 0.75 and mean MAE ranges from 5.7 to 6.7%.

In summary, these results show that our model can maintain the prediction accuracy even for unknown data with $R^2 > 0.92$ and $MAE < 4.9\%$. This demonstrates the robustness of our model and overcomes our concern about overfitting causing our model to work well only for the known data.

STUDY 2: PURSUIT OF 1D MOVING TARGET

Participants

Twelve students from a local university participated in this study (two females and ten males; ages: $M = 22.1$, $SD = 1.38$ years). All were right-handed. All of them could use mice well, including four daily mouse users.

Apparatus

We used a desktop PC (3.2 GHz $\times$ 6 cores, 16 GB of RAM, Windows 7). The display was a SHARP PN-K321 (697.9 $\times$ 392.6 mm, 3840 $\times$ 2160 pixels, 8-msec response time). The other conditions were the same as in Study 1: we used a Logitech mouse, a large mousepad, a constant cursor speed, and the same programming language (Hot Soup Processor). The system ran 500 loops per second in full-screen mode. The same method to measure the mouse latency in Study 1 yielded $M = 62.5$ msec ($SD = 28.0$). Thus, we set $T_{\text{sampling}} = 0.32$ sec as an approximation.

Task

When the participants clicked on the pink rectangular target, it began moving rightwards at a constant speed V (Figure 2). Then, they kept the 1-pixel line-shaped cursor within the target until the end line. If the cursor deviated from the target, they had to return the cursor into the target and continue the task.

The *measurement area* was defined between the start and end lines, and its distance was fixed to 1300 pixels. When the right edge of the target reached the start line, a click sound was played and the target turned red to indicate the measurement had begun. A bell was sounded when the right edge of target reached the end line to indicate the trial ended. Before the start line, we prepared a *run-up area* to view V in every trial. The length of this area was $V \times 2$ sec, so the time for the run-up area was always 2 sec. To our knowledge, in mouse-pointing experiments for a moving target, the longest time for grasping the speed of a target is 1.5 sec [30], and thus our run-up area provided a sufficient time to adjust the speed.

In our initial implementation, to give feedback that the cursor is currently outside the target, the system changed the target color to green and played a beep. However, particularly for high V and small W conditions, the cursor repeatedly deviated and entered the target, which resulted in rapid switching of the target color and lots of noise. Therefore, in the actual experiment, we did not give feedback on the cursor state.

Design and Procedure

We checked whether the cursor was kept inside the target until A from the start line; $A = 100$ to 1300 pixels with 100-pixel interval. For example, if the cursor first deviated from the target when the target's right edge was at $x = 820$, conditions with $A = 100$ to 800 pixels were considered successful, but $A = 900$ to 1300 pixels were errors. After running 20 trials for each $V \times W$ condition, if in 16 trials the cursor remained in the target at $x = 100$ pixels, the success rate for $A = 100$ pixels under a $V \times W$ condition was 80%. Similar data processing has already been proposed, i.e., the use of a-posteriori independent-variable values determined for generating data points [12, 48].

This study was a $3 \times 3 \times 13$ within-subjects design. We used three Vs (160, 220, and 340 pixels/sec), three Ws (30, 50, and 90 pixels), and 13 As (100 to 1300 pixels with 100-pixel interval). We referred to [32] for the values of V and W. One *session* consisted of 22 repetitions of three W values appearing in random order with a fixed V. The first two repetitions (= six trials) were considered practice, and the remaining 20 were for data collection. The order of the three V values was counterbalanced among the 12 participants. In total, we recorded $3_W \times 20_{\text{repetitions}} \times 3_V \times 12_{\text{participants}} = 2160$ trials.

Results

Data Processing

To detect outliers, for the x-coordinate of the cursor viewed from the target center, we computed the mean and *SD* in a trial: M_x (signed: positive is on the right side of the target center) and SD_x (unsigned), respectively. For each $V \times W \times$ participant condition, we detected a trial with M_x greater than 3σ from the average as an outlier; i.e., the cursor was always extremely far from the target center. This was also done for SD_x; i.e., a trial with extreme variability (going back and forth many times). For M_x and SD_x, we respectively detected 14 and 9 trials as outliers (3 inclusive), and thus 20 trials were removed (0.93%).

Statistical Analysis

The success rates did not normally distribute for 62% of the conditions according to Shapiro-Wilk tests with $\alpha = 0.05$. We used non-parametric ANOVAs with *Aligned Rank Transform* and Tukey's p-value adjustment method for pairwise comparisons.

We found significant main effects of V ($F_{2,22} = 10.09$, $p < 0.001$, $\eta_p^2 = 0.48$), W ($F_{2,22} = 142.0$, $p < 0.001$, $\eta_p^2 = 0.93$), and A ($F_{11,121} = 70.01$, $p < 0.001$, $\eta_p^2 = 0.86$). As shown in Figure 7, the success rates monotonically decreased as V increased, as W decreased, and as A increased. Pairwise comparisons showed significant differences between $V = 340$ pixels and the other two values ($p < 0.01$), across all W values ($p < 0.001$), and in 52 pairs among 78 ($= {}_{13}C_2$) combinations of A (at least $p < 0.05$). The difference in effect sizes of V and W ($\eta_p^2 = 0.48$ vs. 0.93) is shown in Figure 8ab; a decrease of W affected the success rates more drastically than V. We also found significant interactions of $V \times W$ ($F_{4,44} = 3.21$, $p < 0.05$, $\eta_p^2 = 0.23$), $V \times A$ ($F_{24,264} = 3.01$, $p < 0.001$, $\eta_p^2 = 0.21$), $W \times A$ ($F_{24,264} = 15.88$, $p < 0.001$, $\eta_p^2 = 0.59$), and $V \times W \times A$ ($F_{48,528} = 3.61$, $p < 0.001$, $\eta_p^2 = 0.25$).

In summary, V and W showed a significant interaction in addition to their main effects (Figure 8). Thus, our model will need to consider their covariance to predict the success rates.

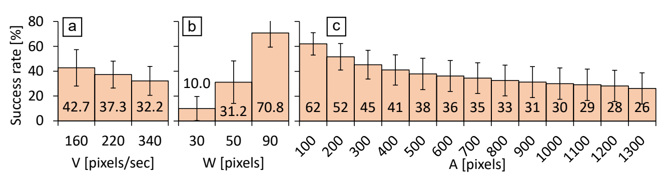


Figure 7. Main effects on success rates. Error bars show 95% CIs.

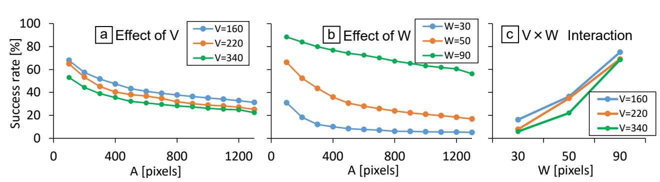


Figure 8. Effects of V and W on success rates.

Prediction Accuracy and Discussion of Study 2

Model Fit to All Known Data

Again, we discuss the necessity of parameters in the model by comparing it with a possible candidate formulation. For static-target pointing, the mean of the endpoint distribution is expected to be close to the center ($\mu_x \approx 0$) [42, 70]. In contrast, for moving-target pointing, the mean is located behind the center ($\mu_x < 0$) [32] due to a delay in the sensory-motor control system [58]. It is not yet evident whether this is also true for the pursuit of moving targets. The cursor is already inside the target when a measurement begins, and no clicking is needed, so it is possible to maintain the cursor ahead of or on the target center ($\mu_x \geq 0$). Hence, as a candidate formulation, we compare a model that applies $\mu_x = 0$ to our model.

The same as with the steering tasks, we could directly measure μ_x and σ_x in a whole trial, but their relationship to those for a single submovement's endpoint distribution was unclear. Hence, we optimize the parameters by using the observed success rates, in the same manner as Study 1.

The results for two candidate models using $N = 117$ ($3_V \times 3_W \times 13_A$) data points are summarized in Table 3. We can see that using non-zero μ_x improved the fitness, which is also shown in Figure 9. Using the coefficients of h–j, the estimated μ_x ranged from 11.7 to 30.0 pixels depending on the V and W values. This means that in a submovement endpoint, the cursor is located ahead of the target center, which is an opposite result from a related work on moving target pointing [25, 32, 33].

Prediction Accuracy to Unknown Data

We ran shuffle-split cross-validation with five training-test ratios. For a training dataset, the rounded-down number of data points was used: e.g., when the training dataset size was 50%, $\texttt{floored}(117/2) = 58$ out of $N = 117$ data points were used for training, and the remaining 59 points were for testing.

The results of the cross-validation are shown in Table 4. The scores did not sharply drop as the size of the training dataset decreased: R^2 ranged from 0.959 to 0.967 and MAE ranged from 6.22 to 6.46%. As a comparison, the "$\mu_x = 0$" model showed mean R^2 ranges from 0.895 to 0.911 and mean MAE ranges from 7.71 to 8.40% depending on the training-test ratio. While this model also maintained a certain prediction accuracy, *not* assuming "$\mu_x = 0$" would be needed to predict the success rate more accurately.

In summary, the results show that our model can maintain the prediction accuracy even for unknown data with $R^2 > 0.95$ and $MAE < 6.5\%$. While this pursuit task is different from moving-target pointing, iteratively applying the success rate prediction model for each submovement's endpoint yielded a fairly high prediction accuracy even for unknown data.

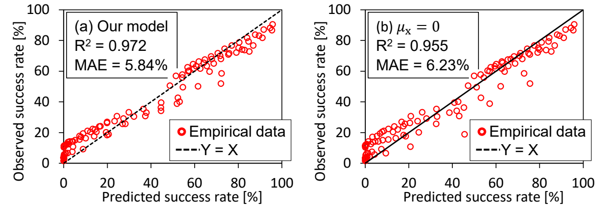


Figure 9. Observed vs. predicted success rates of candidate models.

Table 3. Estimated constants and prediction accuracy in Study 2.

Constants	Our model	"$\mu_x = 0$" model
h	-0.2266	–
i	0.03936	–
j	0.1877	–
k	9.010	-147.8
l	-0.0004097	0.0007050
m	0.01421	0.04903
n	3.741	65.11
R^2	0.9717	0.9554
MAE [%]	5.839	6.255

Table 4. Average fitness by cross-validation in Study 2. The R^2 and MAE values are averaged over 100 iterations for each training-test ratio.

ratio [%]	90 : 10	80 : 20	70 : 30	60 : 40	50 : 50
R^2	0.9672	0.9658	0.9661	0.9633	0.9590
MAE [%]	6.275	6.223	6.273	6.347	6.466

GENERAL DISCUSSION

Model Fit and Effect of Corrective Reaction Time

Although the task requirements in Studies 1 and 2 are quite different in terms of the (e.g.) self/externally paced tracking and perpendicular/collinear constraints to the movement, our concept with the Servo-Gaussian model, namely, to iteratively apply the success rates for each submovement's endpoint falling in the path/target, worked well. The results showed a fairly high prediction accuracy with $R^2 > 0.92$ and $MAE < 4.9\%$ for Study 1 and with $R^2 > 0.95$ and $MAE < 6.5\%$ for Study 2 by cross-validation.

A remaining question is the effect of corrective reaction time (0.26 sec) on prediction accuracy. We set the T_{sampling} values with system latency to 0.3 and 0.32 sec for Studies 1 and 2, respectively. However, as discussed in Related Work, the corrective reaction time could differ within and between persons, and thus it is better to check the effect on prediction accuracy. According to Lin and Hsu [41], the individual corrective reaction times ranged from 0.087 to 0.441 sec, so here we tested $T_{\mathrm{sampling}} = 0.100$ and 0.500 sec as the lower and upper values with the system's latency.

For Study 1, the fitness to all known data was $0.940 < R^2 < 0.945$, and $3.04\% < MAE < 3.16\%$, depending on the T_{sampling}. This indicates that our model is robust against the corrective reaction time differences. However, if we regard the T_{sampling} as another parameter to seek the optimized value, the result shows that $R^2 = 0.954$ and $MAE = 2.76\%$ when $T_{\mathrm{sampling}} = 4.292$ sec. This is obviously inappropriate for a human corrective reaction time with a system's latency. Such a result stems from the characteristic of parameter optimization that all the coefficients balance each other to maximize R^2.

In the same manner, for Study 2, the fitness to all known data was $0.971 < R^2 < 0.975$ and $5.67\% < MAE < 5.86\%$. When we examined the T_{sampling} as an optimized parameter, we obtained $R^2 = 0.9717$ and $MAE = 5.838\%$ when $T_{\mathrm{sampling}} = 0.312$ sec. This T_{sampling} value seems to be within a reasonable range of corrective reaction time with the system's latency. However, compared with the result in Table 3 (fixed T_{sampling}), the differences in R^2 and MAE values were quite slight: improvement for R^2 was 0.00001 and that for MAE was 0.0008%.

In summary, we found no benefit to include the sampling interval as an optimized parameter. Because the corrective reaction time is heuristically known and limited to a certain range, and because the system's latency can be directly measured, it is unnecessary for T_{sampling} to be an optimized parameter. Moreover, neglecting to optimize it also lowers the risk of overfitting to the training dataset.

Evaluating Inter-participant Fitness

For Study 1, when we apply $T_{\text{sampling}} = 0.3$ sec for each of the five participants, the R^2 values ranged from 0.432 to 0.922 ($M = 0.783$), and MAE ranged from 2.05 to 7.00% ($M = 5.36$). Only one participant showed $R^2 < 0.8$. For Study 2, when we apply $T_{\text{sampling}} = 0.32$ sec for each of the 12 participants, the R^2 values ranged from 0.527 to 0.991 ($M = 0.852$), and MAE ranged from 1.67 to 28.0% ($M = 8.37$). Eight participants showed $R^2 > 0.96$, while three showed $R^2 < 0.6$.

This result indicates that our model does not work well for all persons. A reason would be the differences in corrective reaction time for each person. A possible approach to improve the prediction accuracy is to measure the corrective reaction times for each user, as in [41]. This increases the validity of the T_{sampling} values and thus N_{submove}. A limitation of our analyses is that we applied a single heuristic value for the corrective reaction time. Still, even if we apply different T_{sampling} values such as 0.1 or 0.5 sec for participants who showed low fitness values, the R^2 and MAE did not largely change.

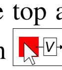
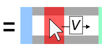
Another reason behind the low fitness is likely due to the number of trials; i.e., one condition was repeatedly tried 28 and 20 times in Studies 1 and 2, respectively, which changed the error rate by 5% for one error in Study 2. As our model is for predicting the central tendency on success rates, evaluating the fitness for each person decreases the number of data points and thus would show poor fits for some users. Hence, the validity of our model was confirmed only for the averaged data, which is common for success-rate modeling [32, 50].

Limitations and Future Work

The findings in this study are limited to our experimental conditions, such as the ranges of task parameters and the use of only a mouse. Also, we asked participants to balance speed and accuracy in Study 1, which means they could spend a long time on each task if needed. However, steering errors may change depending on whether they try to shorten the time or perform more carefully [73]. Our instruction covered only one case among various speed-accuracy tradeoffs. In addition, the numbers of participants were limited, particularly for Study 1. Yet, somewhat small pools of participants have also been seen in other model validity testing (e.g., three users [22] and five [19]). While we are aware that a greater number of participants would strengthen the conclusion, still, the validity of our proposed model was justified via cross-validation.

Each of our studies were conducted within less than an hour, but the validity of our model for users who are sufficiently skilled at some UIs and games is unclear. Although we found no literature that reports the relationship between practice and N_{submove}, it is known that a more practiced task can be performed in a shorter time (cf. learning curve [15]). This means that, for steering tasks, the MT can be reduced, which is equivalent to reducing the N_{submove} in a trial. Further studies are needed to examine the validity of our model after participants are more practiced at a given tracking task.

The purpose of our studies was to test the validity of our Servo-Gaussian model, so we examined the simplest tasks for the steering law (linear path) and moving-target tracking (1D strip). Yet, with the steering law, for example, there are various path shapes [1, 65]. Montazer et al. showed that the servo-mechanism model to derive the steering law is also valid for circular paths [45], and thus we plan to evaluate the model accuracy under non-linear conditions.

As the next step, we plan to investigate the pursuit of a 2D target. Because we confirmed our model's validity for both constraints (y- and x-axes), and because Huang et al.'s endpoint variability model was validated for circular moving targets [33], our model should be effective for predicting success rates with 2D tasks. When the shape of a 2D moving target is square, such a task is a combination of Studies 1 and 2: keeping the cursor inside a target moving horizontally *and* not hitting the top and bottom of the target, as if there is a horizontal path  = . This motivates us to conduct further experiments to test the validity of our model.

Takeaway Notes

Although our model does not suggest specific parameters, such as the optimal height for cascaded menus and the speed of a moving enemy, it contributes to the development of efficient GUIs and video game difficulties by enabling the success rate prediction. Once test users perform a set of $A \times W$ (Study 1) or $V \times W$ (Study 2), designers can predict success rates for new conditions by using our model. This reduces the efforts for conducting additional costly user studies and allows developers to focus on other tasks such as visual design.

Our work introduced an important factor, namely, how often users can successfully accomplish a task, as an indicator of user performance in a manual tracking, in addition to existing ones such as MT. This opens up a new research topic: success rates can be modeled by adequately segmenting an action into smaller submovements. This will inform future studies on success rate prediction for more complicated actions such as the lassoing task, which is a variation of path steering [66].

CONCLUSION

We proposed a Servo-Gaussian model to predict the success rates in manual tracking tasks. Although the task requirements in path steering (Study 1) and pursuit of a 1D moving target (Study 2) are different, once the endpoint distribution in a submovement is obtained, the success rate for a whole trial can be predicted. The results demonstrate a certain generalizability of our model, which worked well for both collinear and perpendicular constraints to the movement direction and for both self and externally paced tracking. This work provides the first evidence of the applicability of success rate models to continuous tracking other than target selection.

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. DOI: http://dx.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. DOI: http://dx.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. DOI: http://dx.doi.org/10.1145/365024.365027

[4] G. H. Begbie. 1959. Accuracy of aiming in linear hand-movements. *Quarterly Journal of Experimental Psychology* 11, 2 (1959), 65–75. DOI: http://dx.doi.org/10.1080/17470215908416293

[5] W. D. A. Beggs, Jacqueline A. Andrew, Martha L. Baker, S. R. Dove, Irene Fairclough, and C. I. Howarth. 1972. The accuracy of non-visual aiming. *Quarterly Journal of Experimental Psychology* 24, 4 (1972), 515–523. DOI: http://dx.doi.org/10.1080/14640747208400311

[6] W. D. A. Beggs, Ruth Sakstein, and C. I. Howarth. 1974. The Generality of a Theory of the Intermittent Control of Accurate Movements. *Ergonomics* 17, 6 (1974), 757–768. DOI: http://dx.doi.org/10.1080/00140137408931422

[7] W. J. Beggs and C. I. Howarth. 1970. Movement Control in a Repetitive Motor Task. *Nature* 225 (1970), 752–753.

[8] 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. DOI: http://dx.doi.org/10.1145/2470654.2466180

[9] Xiaojun Bi and Shumin Zhai. 2016. Predicting Finger-Touch Accuracy Based on the Dual Gaussian Distribution Model. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology (UIST '16)*. ACM, New York, NY, USA, 313–319. DOI: http://dx.doi.org/10.1145/2984511.2984546

[10] 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 and Technology (UIST '15)*. ACM, New York, NY, USA, 629–636. DOI: http://dx.doi.org/10.1145/2807442.2807454

[11] Alan H. S. Chan, Errol R. Hoffmann, and Jeremy C. H. Ho. 2019. Movement time and guidance accuracy in teleoperation of robotic vehicles. *Ergonomics* 62, 5 (2019), 706–720. DOI: http://dx.doi.org/10.1080/00140139.2019.1571246

[12] Olivier Chapuis, Renaud Blanch, and Michel Beaudouin-Lafon. 2007. *Fitts' Law in the Wild: A Field Study of Aimed Movements*. Technical Report. https://hal.archives-ouvertes.fr/hal-00612026 LRI Technical Repport Number 1480, Univ. Paris-Sud, 11 pages.

[13] Christopher Clarke and Hans Gellersen. 2017. MatchPoint: Spontaneous Spatial Coupling of Body Movement for Touchless Pointing. In *Proceedings of the 30th Annual ACM Symposium on User Interface Software and Technology (UIST '17)*. Association for Computing Machinery, New York, NY, USA, 179–192. DOI:http://dx.doi.org/10.1145/3126594.3126626

[14] Kenneth. J. W. Craik. 1947. THEORY OF THE HUMAN OPERATOR IN CONTROL SYSTEMS. *British Journal of Psychology. General Section* 38, 3 (1947), 142–148. DOI: http://dx.doi.org/10.1111/j.2044-8295.1948.tb01149.x

[15] E. R. F. W. Crossman. 1959. A theory of the acquisition of speed skill. *Ergonomics* 2, 2 (1959), 153–166. DOI: http://dx.doi.org/10.1080/00140135908930419

[16] E. R. F. W. Crossman and P. J. Goodeve. 1983. Feedback Control of Hand-Movement and Fitts' Law. *The Quarterly Journal of Experimental Psychology Section A* 35, 2 (1983), 251–278. DOI: http://dx.doi.org/10.1080/14640748308402133

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

[18] Colin G. Drury. 1994. A model for movements under intermittent illumination. *Ergonomics* 37, 7 (1994), 1245–1251. DOI: http://dx.doi.org/10.1080/00140139408964902

[19] C G. Drury and E. B. Daniels. 1975. Performance Limitations in Laterally Constrained Movements. *Ergonomics* 18, 4 (1975), 389–395. DOI: http://dx.doi.org/10.1080/00140137508931472

[20] Colin G. Drury, M. Ali Montazer, and Mark H. Karwan. 1987. Self-Paced Path Control as an Optimization Task. *IEEE Trans. Systems, Man, and Cybernetics* 17, 3 (1987), 455–464. DOI: http://dx.doi.org/10.1109/TSMC.1987.4309061

[21] 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. DOI: http://dx.doi.org/10.1037/h0055392

[22] Paul M. Fitts and Barbara K. Radford. 1966. Information Capacity of Discrete Motor Responses Under Different Cognitive Sets. *Journal of Experimental Psychology* 71, 4 (1966), 475–482. DOI: http://dx.doi.org/10.1037/h0022970

[23] Tovi Grossman and Ravin Balakrishnan. 2005. A Probabilistic Approach to Modeling Two-dimensional Pointing. *ACM Trans. Comput.-Hum. Interact.* 12, 3 (Sept. 2005), 435–459. DOI: http://dx.doi.org/10.1145/1096737.1096741

[24] Abir Al Hajri, Sidney Fels, Gregor Miller, and Michael Ilich. 2011. Moving Target Selection in 2D Graphical User Interfaces. In *Human-Computer Interaction – INTERACT 2011*, Pedro Campos, Nicholas Graham, Joaquim Jorge, Nuno Nunes, Philippe Palanque, and Marco Winckler (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 141–161.

[25] Khalad Hasan, Tovi Grossman, and Pourang Irani. 2011. Comet and Target Ghost: Techniques for Selecting Moving Targets. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '11)*. Association for Computing Machinery, New York, NY, USA, 839–848. DOI: http://dx.doi.org/10.1145/1978942.1979065

[26] Errol R. Hoffmann. 1991. Capture of moving targets: a modification of Fitts' Law. *Ergonomics* 34, 2 (1991), 211–220. DOI: http://dx.doi.org/10.1080/00140139108967307

[27] Errol R. Hoffmann. 2009. Review of models for restricted-path movements. *International Journal of Industrial Ergonomics* 39, 4 (2009), 578–589. DOI: http://dx.doi.org/https://doi.org/10.1016/j.ergon.2008.02.007

[28] Errol R. Hoffmann and Colin G. Drury. 2019. Models of the effect of teleoperation transmission delay on robot movement time. *Ergonomics* 62, 9 (2019), 1175–1180. DOI:http://dx.doi.org/10.1080/00140139.2019.1612954

[29] C.I. Howarth, W.D.A. Beggs, and J.M. Bowden. 1971. The relationship between speed and accuracy of movement aimed at a target. *Acta Psychologica* 35, 3 (1971), 207–218. DOI:http://dx.doi.org/https://doi.org/10.1016/0001-6918(71)90022-9

[30] Jin Huang and Byungjoo Lee. 2019. Modeling Error Rates in Spatiotemporal Moving Target Selection. In *Extended Abstracts of the 2019 CHI Conference on Human Factors in Computing Systems (CHI EA '19)*. Association for Computing Machinery, New York, NY, USA, Article Paper LBW2411, 6 pages. DOI: http://dx.doi.org/10.1145/3290607.3313077

[31] Jin Huang, Feng Tian, Xiangmin Fan, Huawei Tu, Hao Zhang, Xiaolan Peng, and Hongan Wang. 2020. Modeling the Endpoint Uncertainty in Crossing-Based Moving Target Selection. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems (CHI '20)*. Association for Computing Machinery, New York, NY, USA, 1–12. DOI: http://dx.doi.org/10.1145/3313831.3376336

[32] Jin Huang, Feng Tian, Xiangmin Fan, Xiaolong (Luke) Zhang, and Shumin Zhai. 2018. Understanding the Uncertainty in 1D Unidirectional Moving Target Selection. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems (CHI '18)*. Association for Computing Machinery, New York, NY, USA, Article Paper 237, 12 pages. DOI: http://dx.doi.org/10.1145/3173574.3173811

[33] Jin Huang, Feng Tian, Nianlong Li, and Xiangmin Fan. 2019. Modeling the Uncertainty in 2D Moving Target Selection. In *Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology (UIST '19)*. Association for Computing Machinery, New York, NY, USA, 1031–1043. DOI: http://dx.doi.org/10.1145/3332165.3347880

[34] Richard J. Jagacinski, Daniel W. Repperger, Sharon L. Ward, and Martin S. Moran. 1980. A Test of Fitts' Law with Moving Targets. *Human Factors* 22, 2 (1980), 225–233. DOI: http://dx.doi.org/10.1177/001872088002200211

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

[36] 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. DOI: http://dx.doi.org/10.1145/1056808.1056968

[37] Byungjoo Lee, Sunjun Kim, Antti Oulasvirta, Jong-In Lee, and Eunji Park. 2018. Moving Target Selection: A Cue Integration Model. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems (CHI '18)*. Association for Computing Machinery, New York, NY, USA, Article Paper 230, 12 pages. DOI: http://dx.doi.org/10.1145/3173574.3173804

[38] Byungjoo Lee, Mathieu Nancel, Sunjun Kim, and Antti Oulasvirta. 2020. AutoGain: Gain Function Adaptation with Submovement Efficiency Optimization. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems (CHI '20)*. Association for Computing Machinery, New York, NY, USA, 1–12. DOI:http://dx.doi.org/10.1145/3313831.3376244

[39] Injung Lee, Sunjun Kim, and Byungjoo Lee. 2019. Geometrically Compensating Effect of End-to-End Latency in Moving-Target Selection Games. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems (CHI '19)*. Association for Computing Machinery, New York, NY, USA, Article Paper 560, 12 pages. DOI: http://dx.doi.org/10.1145/3290605.3300790

[40] Jui-Feng Lin, Colin G. Drury, Mark H. Karwan, , and Victor Paquet. 2009. A general model that accounts for Fitts' law and Drury's model. In *Proceedings of the 17th Congress of the International Ergonomics*.

[41] Ray F. Lin and Chih-Hsiang Hsu. 2014. Measuring individual corrective reaction time using the intermittent illumination model. *Ergonomics* 57, 9 (2014), 1337–1352. DOI: http://dx.doi.org/10.1080/00140139.2014.933268

[42] I. Scott MacKenzie. 1992. Fitts' law as a research and design tool in human-computer interaction. *Human-Computer Interaction* 7, 1 (1992), 91–139. DOI: http://dx.doi.org/10.1207/s15327051hci0701_3

[43] David E. Meyer, Richard A. Abrams, Sylvan Kornblum, Charles E. Wright, and J. E. Keith Smith. 1988. Optimality in human motor performance: ideal control of rapid aimed movements. *Psychological Review* 95, 3 (1988), 340–370. DOI: http://dx.doi.org/10.1037/0033-295X.95.3.340

[44] David E. Meyer, J. E. Keith Smith, and Charles E. Wright. 1982. Models for the speed and accuracy of aimed movements. *Psychological Review* 89, 5 (1982), 449–482. DOI: http://dx.doi.org/10.1037/0033-295X.89.5.449

[45] 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. DOI: http://dx.doi.org/10.1109/21.23090

[46] Martez E. Mott and Jacob O. Wobbrock. 2014. Beating the Bubble: Using Kinematic Triggering in the Bubble Lens for Acquiring Small, Dense Targets. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '14)*. Association for Computing Machinery, New York, NY, USA, 733–742. DOI: http://dx.doi.org/10.1145/2556288.2557410

[47] Jörg Müller, Antti Oulasvirta, and Roderick Murray-Smith. 2017. Control Theoretic Models of Pointing. *ACM Trans. Comput.-Hum. Interact.* 24, 4, Article 27 (Aug. 2017), 36 pages. DOI: http://dx.doi.org/10.1145/3121431

[48] 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. DOI: http://dx.doi.org/10.1145/3025453.3025951

[49] Eunji Park and Byungjoo Lee. 2018. Predicting Error Rates in Pointing Regardless of Target Motion. (2018).

[50] Eunji Park and Byungjoo Lee. 2020. An Intermittent Click Planning Model. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems (CHI '20)*. Association for Computing Machinery, New York, NY, USA, 1–13. DOI: http://dx.doi.org/10.1145/3313831.3376725

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

[52] Nicolas Rashevsky. 1970. Mathematical biophysics of automobile driving IV. *Bulletin of Mathematical Biophysics* 32, 1 (1970), 71–78. DOI: http://dx.doi.org/10.1007/BF02476794

[53] Richard A. Schmidt, Howard Zelaznik, Brian Hawkins, James S. Frank, and Jr John T. Quinn. 1979. Motor-output variability: A theory for the accuracy of rapid motor acts. *Psychological Review* 86, 5 (1979), 415–451. DOI: http://dx.doi.org/10.1037/0033-295X.86.5.415

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

[55] 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. DOI: http://dx.doi.org/10.1007/s00221-013-3700-4

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

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

[58] Emanuel Todorov. 2005. Stochastic Optimal Control and Estimation Methods Adapted to the Noise Characteristics of the Sensorimotor System. *Neural Computation* 17, 5 (2005), 1084–1108. DOI: http://dx.doi.org/10.1162/0899766053491887

[59] M. A. Vince. 1947. THE INTERMITTENCY OF CONTROL MOVEMENTS AND THE PSYCHOLOGICAL REFRACTORY PERIOD1. *British Journal of Psychology. General Section* 38, 3 (1947), 149–157. DOI: http://dx.doi.org/10.1111/j.2044-8295.1948.tb01150.x

[60] Wayne A. Wickelgren. 1977. Speed-accuracy tradeoff and information processing dynamics. *Acta Psychologica* 41, 1 (1977), 67–85. DOI: http://dx.doi.org/10.1016/0001-6918(77)90012-9

[61] Jacob O. Wobbrock, Edward Cutrell, Susumu Harada, and I. Scott MacKenzie. 2008. An Error Model for Pointing Based on Fitts' Law. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '08)*. ACM, New York, NY, USA, 1613–1622. DOI: http://dx.doi.org/10.1145/1357054.1357306

[62] Jacob O. Wobbrock, Leah Findlater, Darren Gergle, and James J. Higgins. 2011a. The Aligned Rank Transform for Nonparametric Factorial Analyses Using Only Anova Procedures. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '11)*. ACM, New York, NY, USA, 143–146. DOI: http://dx.doi.org/10.1145/1978942.1978963

[63] Jacob O. Wobbrock, Alex Jansen, and Kristen Shinohara. 2011b. Modeling and Predicting Pointing Errors in Two Dimensions. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '11)*. ACM, New York, NY, USA, 1653–1656. DOI: http://dx.doi.org/10.1145/1978942.1979183

[64] Shota Yamanaka. 2019. Steering Performance with Error-accepting Delays. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems (CHI '19)*. ACM, New York, NY, USA, Article 570, 9 pages. DOI:http://dx.doi.org/10.1145/3290605.3300800

[65] 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. DOI:http://dx.doi.org/10.1145/2858036.2858037

[66] Shota Yamanaka and Wolfgang Stuerzlinger. 2019. Modeling Fully and Partially Constrained Lasso Movements in a Grid of Icons. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems (CHI '19)*. ACM, New York, NY, USA, Article 120, 12 pages. DOI: http://dx.doi.org/10.1145/3290605.3300350

[67] 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. DOI: http://dx.doi.org/10.1145/3025453.3025836

[68] Difeng Yu, Hai-Ning Liang, Xueshi Lu, Kaixuan Fan, and Barrett Ens. 2019. Modeling Endpoint Distribution of Pointing Selection Tasks in Virtual Reality Environments. *ACM Trans. Graph.* 38, 6, Article Article 218 (Nov. 2019), 13 pages. DOI: http://dx.doi.org/10.1145/3355089.3356544

[69] Howard N. Zelaznik, Susan Mone, George P. McCabe, and Christopher Thaman. 1988. Role of temporal and spatial precision in determining the nature of the speed-accuracy trade-off in aimed-hand movements. *Journal of Experimental Psychology: Human Perception and Performance* 14, 2 (1988), 221–230. DOI: http://dx.doi.org/10.1037/0096-1523.14.2.221

[70] 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. DOI: http://dx.doi.org/10.1016/j.ijhcs.2004.09.007

[71] Ziyue Zhang, Jin Huang, and Feng Tian. 2020. Modeling the Uncertainty in Pointing of Moving Targets with Arbitrary Shapes. In *Extended Abstracts of the 2020 CHI Conference on Human Factors in Computing Systems (CHI EA '20)*. Association for Computing Machinery, New York, NY, USA, 1–7. DOI: http://dx.doi.org/10.1145/3334480.3382875

[72] 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.

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