

• Supplementary File •

A Unified User Behavior Model for Trajectory-based Tasks with Different Types of Path Constraints

Hao ZHANG^{1,2}, Jin HUANG^{1,3*}, Huawei TU⁴, Feng TIAN^{1,3}, Guozhong DAI¹ & Hongan WANG¹

¹Beijing Key Laboratory of Human-Computer Interaction, Institute of Software Chinese Academy of Sciences, Beijing 100190, China;

²School of Computer Science and Technology, University of Chinese Academy of Sciences, Beijing 101408, China;

³School of Artificial Intelligence, University of Chinese Academy of Sciences, Beijing 101408, China;

⁴Department of Computer Science and Information Technology La Trobe University Melbourne, Victoria 3086, Australia

Appendix A The Method Description

Appendix A.1 Task Space

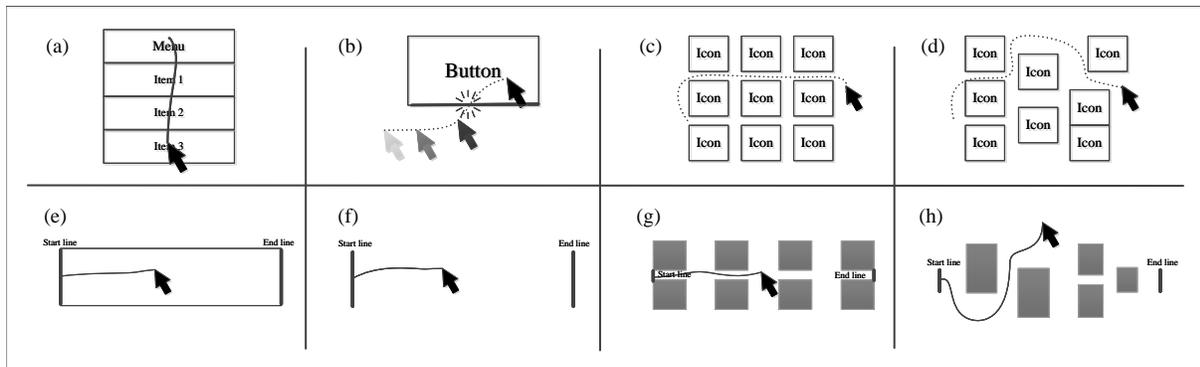


Figure A1 (a-d) Various fundamental interaction behavior on graphical user interface (GUI): (a) navigating through menu, (b) crossing the boundary of a graphical object. (c) lassoing icons which are arranged at regular intervals. (d) lassoing randomly arranged icons. (e-h) Various trajectory-based tasks: (e) Steering task, (f) Goal-Crossing task, (g) Steering through obstacles with regular intervals (Steering-Crossing task), (h) Steering through random obstacles (Randomized Steering task).

As shown in Figure A1, when interacting with the objects on graphical user interface (GUI), users often implicitly perform various trajectory-based interactions, such as navigating through menus (Figure A1 a), entering the boundary of a button (Figure A1 b) and lassoing regularly arranged icons (Figure A1 c). Based on these fundamental interactions on GUI, researchers explored three typical trajectory-based tasks: the Steering task [1], the Goal Crossing task [1, 2], and the Steering-Crossing task [3]. The Steering task (Figure A1 e) needs users to navigate from one end of a tunnel to the other without touching the boundaries of the tunnel. Thus it is a task with continuous path constraints. The Goal Crossing task (Figure A1 f) asks users to stroke through two targets at a certain distance successively. It has only two discrete constraints at the beginning and the end of the movement, we consider it a task with discrete constraints. The Steering-Crossing task is a combination of the above two tasks, it requires users to navigate through obstacles at regular intervals (Figure A1 g). So it can be viewed as a task with interval constraints. Besides, considering interactions with irregularly arranged objects on GUI (Figure A1 d), we further introduce one randomized task (i.e., randomized steering task Figure A1 h).

The task space in this paper is the area in which the previously mentioned task is performed by the user. The cursor controlled by users is represented by *Particle*. The goal is a destination to which the cursor is heading. The model has two types of goals: *Final Goal* and *Temporary Goal*. The *Final Goal* is the final target area of the trajectory-based tasks. The *Temporary Goal* is a goal in a Visual Field (which will be specified later) that represents a local destination toward which the cursor tends to move. *Obstacle* refers to boundaries in the task space. *Visual Field* is a fan-shaped area centered on the cursor and faces the cursor's moving direction. The *Visual Field* generates a *Temporary Goal* based on the geometry of the local boundaries, which directs the local movement of the *Particle*.

* Corresponding author (email: huangjin@iscas.ac.cn)

Appendix A.2 Artificial Potential Field

APF consists of two parts, one is the *Attractive Potential* and the other is the *Repulsive Potential*. The *Attractive Potential* is a potential that is centered on a *Goal* and drives the *Particle* moves towards the *Goal*. The *Attractive Potential* from the center of the *Goal* to the *Particle* can be written as:

$$U_{att}(t) = \frac{1}{2}k\rho_{goal}^2(p) = \frac{1}{2}k(p_t - p_{goal})^2, \quad (A1)$$

where $\rho_{goal}(p) = \|p - p_{goal}\|$ is the Euclidean distance between the *Particle* and the center of the *Goal*, $p_t = (x(t), y(t))^T$ is the *Particle*'s position at current time t , and k is a positive scaling factor of *Attractive Potential*. Eq. A1 shows that $U_{att}(t)$ monotonically increases with the current distance towards the center of the *Goal*. The attractive force $F_{att}(t)$ in current *Attractive Potential* can be obtained by calculating the partial derivative:

$$F_{att}(t) = -\nabla U_{att}(t) = -\frac{1}{2} - \nabla \rho_{goal}^2(p) = -k(p_t - p_{goal}). \quad (A2)$$

Attractive Potential describes the desire of the *Particle* to move towards the center of the *Goal*. The farther away from the center of the *Goal*, the more eager the *Particle* is to move towards it. As shown in Eq. (A2), $F_{att}(t)$ converges linearly toward zero as the *Particle* approaches the *Goal*.

Repulsive Potential generates high potential at the center of an *Obstacle* that can repel the *Particle* from the *Obstacle*. The *Repulsive Potential* can be written as follows:

$$U_{rep}(t) = \begin{cases} 0 & , \rho(p) > \rho_0 \\ \frac{1}{2}\eta \left(\frac{1}{\rho(p)} - \frac{1}{\rho_0} \right) & , \rho(p) \leq \rho_0 \end{cases} \quad (A3)$$

Where η is a positive constant of the *Repulsive Potential*, $\rho(p) = \|p - p(p)\|$ denotes the minimal distance from the *Particle* to one certain point on *Obstacle*. ρ_0 is a positive constant denoting the influence range of the *Repulsive Potential*.

In practice, we found that *Particle* may stop approaching to the *Goal* when there are *Obstacles* nearby. To address this problem, we adopt a modified *Repulsive Potential* introduced by [4], which generates forces in two directions: one from the *Particle* to the *Obstacle*, the other from the *Particle* the *Goal*. The modified *Repulsive Potential* takes the following form:

$$\begin{aligned} F_{rep}(t) &= \frac{\partial U_{rep}(t)}{\partial \mathbf{n}_p} + \frac{\partial U_{rep}(t)}{\partial \mathbf{n}_{goal}} \\ &= \begin{cases} 0 & , \rho(p) > \rho_0 \\ \|F_{rep_p}(t)\| \mathbf{n}_p + \|F_{rep_g}(t)\| \mathbf{n}_{goal} & , \rho(p) \leq \rho_0 \end{cases} \end{aligned} \quad (A4)$$

where \mathbf{n}_p is the direction vector from the *Particle* to the *Obstacle*, \mathbf{n}_{goal} is the direction vector from the *Particle* to the *Goal*. In these two directions, $F_{rep}(t)$ is divided into two sub-force: $F_{rep_p}(t)$ and $F_{rep_g}(t)$. The two forces are calculated by the following formulae:

$$\|F_{rep_p}(t)\| = \eta \left(\frac{1}{\rho(p)} - \frac{1}{\rho_0} \right) \frac{\rho_{goal}^2(p)}{\rho^2(p)} \quad (A5)$$

$$\|F_{rep_g}(t)\| = \eta \left(\frac{1}{\rho(p)} - \frac{1}{\rho_0} \right)^2 \rho(p) \quad (A6)$$

Where η is a positive constant of the *Repulsive Potential*, $\rho(p) = \|p - p(p)\|$ denotes the minimal distance from the *Particle* to one certain point on *Obstacle*. ρ_0 is a positive constant denoting the influence range of the *Repulsive Potential*.

Appendix A.3 Modified Artificial Potential Field Model

Following the basic mechanism of Artificial Potential Field (APF), our model describes the movement of the particle in trajectory-based tasks with *Attractive Potential* and *Repulsive Potential*. The state of the particle X at time t can be expressed as follows:

$$X(t) = [F(t), v(t), x(t), y(t)], \quad (A7)$$

where $F(t)$ is the combined force of the particle, $F(t)$ consists of two parts, one is the force generated by the whole potentials $F_U(t)$, another is the perturbing force $F_{per}(t)$ which perpendicular to the moving direction of the particle. $v(t)$ is the speed of the particle which can be calculated by the combined force $F(t)$ using Newton's second law. $(x(t), y(t))$ is the position of the particle.

The framework of our model is shown in Fig. A2. The framework is divided into two parts: Task Space Modeling and Movement Status Update. In the Task Space Modeling, attractive and repulsive potential fields are used to build the overall distribution of potentials caused by boundaries, non-target objects, and the *Final Goal* in the *Task Space* (Fig. A2 (a)). The potentials in the *Task Space* produce forces $F_U(t)$ that cause the particle to move (Fig. A2 (b)). Then we introduce an uncertainty model (the perturbing force $F_{per}(t)$) to account for the uncertainty in human behavior. Given the resultant force from force analysis, Newton's second law is used to calculate the *Particle*'s movement status in a specific time delta (Fig. A2 (c)). Finally, as the *Particle* moves in the *Task Space*, it changes the potential fields in the space, causing the model to send feedback (the updated state of the particle $X(t+1)$) to the *Task Space*, update the potential fields, and repeat the loop. Our model can calculate the movement of the *Particle* until it reaches the *Final Goal* by iterating the above process.

In the following sections, we introduce the parts of Task Space Modeling and Movement Status Update respectively.

Appendix A.3.1 Task Space Modeling

This section describes how we use the *Attractive Potential* and *Repulsive Potential* to model the potentials in the *Task Space*. According to different purposes of movement simulation, we divide the potentials of the *Task Space* into three parts: potentials in overall task space, potentials in the space near the *Particle*, and potentials in the space near the *Final Goal*.

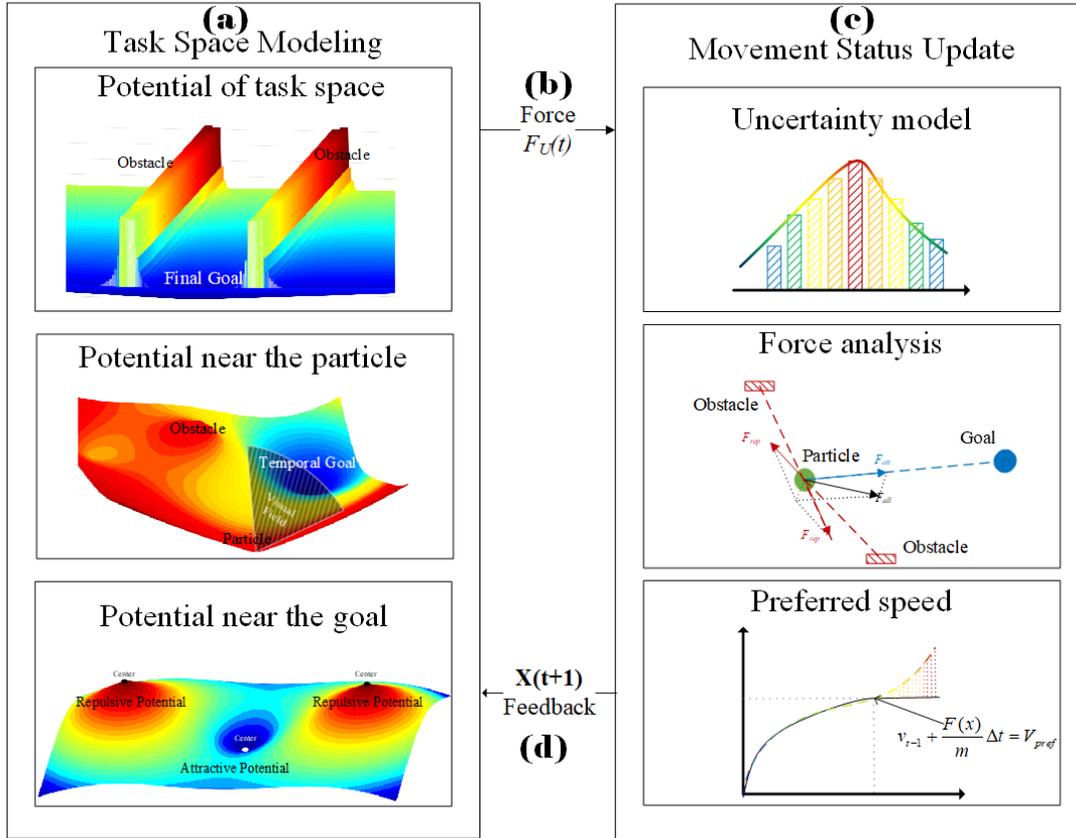


Figure A2 Framework of Modified Artificial Potential Field Model

Potentials in the Overall Task Space To model the potentials in the overall task space, we first place an *Attractive Potential* U_{Gatt} (calculated by Eq.A1) at the location of the *Final Goal* to drive the *Particle* to the target area of the task. We then rasterize the boundaries (obstacles) of the task and attach a repulsive potential to each of the rasterized “obstacle grid” in the task space. This will result in numerous repulsive potentials and significantly reduce the simulation efficiency. To simplify the calculation, inspired by the work of Kala [5], we use six sources of potentials along the moving direction of the *Particle* to describe *Repulsive Potential* U_{Orep} (calculated by Eq.A8) generated by *Obstacle* around the *Particle*. These six sources of potentials include one in the front of the *Particle* (i.e., forward potential) and one in the back (i.e., backward potential), four on the left and right sides of the *Particle* (i.e., diagonal and side potential). Then, the combined of *Repulsive Potential* generated by *Obstacle* U_{Orep} can be represented by:

$$U_{Orep} = U_{Orep}(p_{fro}) + U_{Orep}(p_{back}) + U_{Orep}(p_{diat}) + U_{Orep}(p_{diab}) + U_{Orep}(p_{top}) + U_{Orep}(p_{bot}). \quad (A8)$$

Each source of potential is calculated according to Eq.A4.

Potentials Near the Particle Instead of moving optimally to the target area, users’ movements can be influenced by the layout of a specific local space within a certain distance. Users, for example, tend to move toward the farthest visible destination in front of them. We use the *Visual Field* to simulate this phenomenon. The *Visual Field* is defined as an area that can be seen from a *Particle*. It is commonly used in robot path planning [6, 7], where the robot must first locate a *Temporary Goal* within a *visual Field*, and then safely avoid obstacles by constantly updating the *Temporary Goal*. Because of its working principle, this method is highly adaptable, making the model suitable for a variety of scenarios.

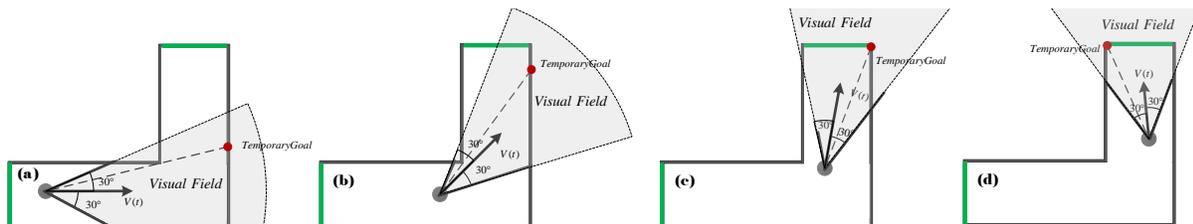


Figure A3 A simple diagram illustrating how to use the visual field to navigate a curved tunnel.

As shown in Figure A3, the *Visual Field* is a fan-shaped area with the *Particle* as the center and the *Particle*'s movement direction as the central axis. The center angle θ and radius ρ determine the range of *Visual Field*. When there is an *Obstacle* O in *Visual Field*, the *Particle* will select a point on *Obstacle*, which is farthest from the *Particle* as the *Temporary Goal*. As the *Final Goal*, we attach a *Attractive Potential* U_{Tatt} (calculated by Eq. A1) at the *Temporary Goal*, so it generates attractive force to help the particle move to the edge of *Obstacle*. Figure A3 shows how the *Visual Field* helps the *Particle* in navigating through a curved tunnel. In a curved tunnel, the tunnel boundary is viewed as an obstacle. The *Temporary Goal* always causes the *Particle* to move to the farthest *Obstacle*, causing it to bypass the tunnel. We set the center angle θ as 60° and radius ρ as 350 px (53.67mm).

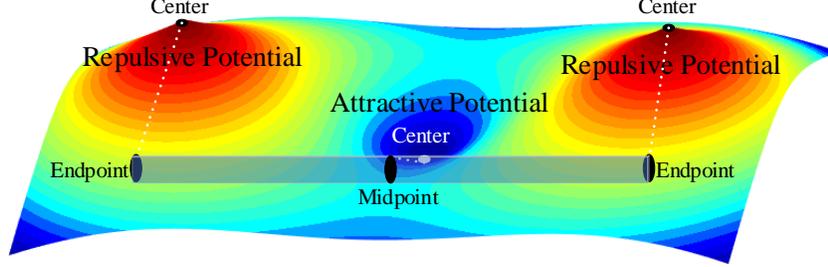


Figure A4 The potentials near a strip-like goal.

Potentials Near the Final Goal In previous APF studies, the goal was usually regarded as a point-like object, so the particle's movement usually ended when it came into contact with an adjacent area of the goal. However, strip-like goals [1] are more common in trajectory-based tasks, which result in completely different user behavior relative to point-like goals. To finish the task, users must control the cursor to cross (but not reach) the center of the strip-like goal. This movement is referred to as the "goal-crossing" behavior in crossing user interfaces [2]. To approximate the goal-crossing behavior, we added two *Repulsive Potential* U_{Grep} (calculated refer to Eq. A4) at the two endpoints of the strip-like goal, as well as an *Attractive Potential* at the goal's midpoint as shown in Figure A4. In this way, the *Particle* will avoid touching the two endpoints of the strip-like goal and try to pass through the middle of the goal.

Appendix A.3.2 Movement Status Update

This section describes how we update and adjust the movement of the *Particle* based on the forces generated by the potentials in the *Task Space*. First, an uncertainty model is used to adjust the resultant force to approximate the uncertainty in the human musculoskeletal system; next, a force analysis is performed to calculate the resultant force on the *Particle*; and finally, a preferred speed model is employed to limit the maximum moving speed of the *Particle*.

Uncertainty Model Through the above calculation of force generated by the whole potential field, we can obtain the status of the *Particle* in a specified time and iterate the simulation process to obtain a unique trajectory until the *Particle* reaches the end. However, the unique trajectory determined by the above process is quite different from actual human behavior in performing HCI tasks which contains uncertainty in movement, speed, or decision-making. In this section, we introduce a perturbing force to the APF model to account for the uncertainty in human behavior.

According to neuroscience research, the uncertainty of human behavior is caused by control signal noise in the human body. The noise in the control signal causes force perturbation in the musculoskeletal system, which finally leads to uncertain hand movement [8]. Specifically, evidence [8–10] shows that the variability in the force produced is linearly related to the amount of force produced. As the user uses more force to achieve a faster cursor movement speed, the uncertainty of hand movement increases with increased acceleration and speed [11].

As a result, we introduced a perturbing force ($F_{per}(t)$) to approximate the uncertainty of hand movement. The perturbing force is a force perpendicular to the moving direction of the hand, and the perturbing force follows a Gaussian distribution (N_{per}) with a mean value of 0 and a standard deviation (SD) of σ . The SD of σ of the perturbing force is linearly related to particle's current speed ($v(t)$) as follows:

$$\sigma = av(t) + b. \quad (A9)$$

Force Analysis The overall potential can be obtained by calculating the sum of attractive and repulsive potentials. Then we can get the force $F_U(t)$ generated by the whole potentials by calculating the partial derivative:

$$\begin{aligned} F_U(t) &= -\nabla U(t) = -\nabla U_{att}(t) + \nabla U_{rep}(t) \\ &= -\nabla U_{Tatt} - \left(1 - \frac{\rho_{G-goal}(p)}{\rho_{G-goal}(p_0)}\right) \nabla U_{Gatt} + \nabla U_{Orep} + \nabla U_{Grep}, \end{aligned} \quad (A10)$$

where $\rho_{G-goal}(p_0)$ is the distance between *Particle*'s initial position and *Final Goal*, $\rho_{G-goal}(p)$ is the distance between *Particle*'s current position and *Final Goal*. Figure A5 shows an example of the combined force of *Particle* generated by the whole potentials in steering task.

As shown in Figure A5, an *Attractive Potential Field* (U_{Gatt}) located at the center of the *Final Goal* generates *Attractive Force* to drive the *Particle* to the target area. The six sources of potentials along the moving direction of the *Particle* describe the *Repulsive Potential Field* (U_{Orep}) generated by the boundaries of the tunnel in the *Task Space*. Another *Attractive Potential Field* (U_{Tatt}) is placed at a *Temporary Goal* within the *VisualField*, generating attractive force to lead the *Particle* move to the

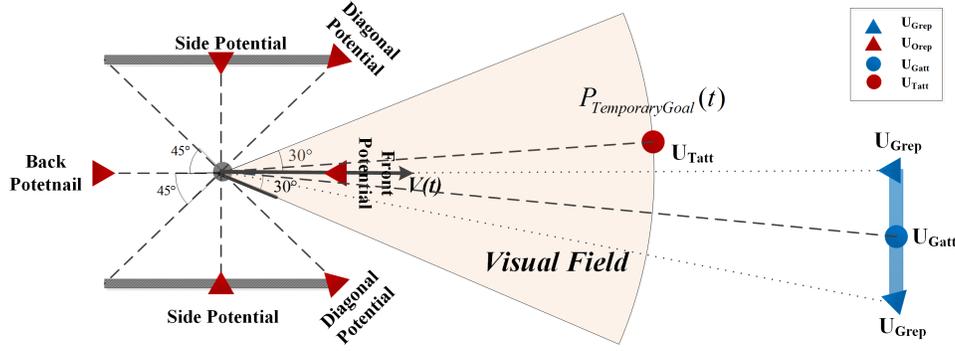


Figure A5 An example of the force of the particle generated by the whole potentials.

Temporary Goal. The two *Repulsive Potential Field* (U_{Grep}) placed at the two endpoints of the *Final Goal* generate repulsive forces to ensure that the *Particle* cross through the center of the goal. Each potential produces a force at the *Particle*.

Finally, by adding the uncertainty adjustment to Eq. A10, we can obtain the final combined force $F(t)$ on the *Particle*:

$$F(t) = F_U(t) + F_{per}(t) = F_U(t) + N_{per} \mathbf{n}_v, N_{per} \sim N(0, \sigma), \quad (\text{A11})$$

where vector \mathbf{n}_v denotes the normal direction of the present movement. The combined force of these forces finally determines the exact movement of the *Particle*.

Preferred Speed Model According to Newton's second law and Eq. A11, the *Particle*'s speed at next frame ($v(t+1)$) can be calculated as follows:

$$v(t+1) = v(t) + \frac{F(t)}{m} \Delta t, \quad (\text{A12})$$

where $F(t)$ is combined force calculated by Eq. A11, $v(t)$ is the current speed of the *Particle*

Different from simple physical simulation without any restrictions, simulation of human movement should consider its inherent behavior pattern. More relevant to the trajectory-based tasks in this paper, users tend to reduce the acceleration and speed of their hands to improve accuracy in narrower spaces according to the speed-accuracy trade-off principle [12]. As a result, we developed a Preferred Speed Model to control the maximum moving speed of the *Particle* according to the width of the tunnel. The preferred speed (V_{pref}) of the *Particle* is determined by as follows:

$$V_{pref} = cW + d, \quad (\text{A13})$$

where c and d are two constants, W is the width of the tunnel. Eq. A12 is finally defined as follow:

$$v(t+1) = \min(v(t) + \frac{F(t)}{m} \Delta t, V_{pref}), \quad (\text{A14})$$

where $F(t)$ is combined force calculated by Eq. A11, $v(t)$ is the speed at time t , Δt is the time difference between the current and next states. Finally, the position of the *Particle* in the next frame $p_{t+1} = (x_{t+1}, y_{t+1})^T$ can be calculated as follow:

$$p_{t+1} = p_t + v(t+1) \Delta t, \quad (\text{A15})$$

where $p_t = (x(t), y(t))^T$ is the *Particle*'s position at current time t , $v(t+1)$ is the *Particle*'s speed calculated by Eq. A14. In this way, we can gradually update the position of the *Particle* until the *Particle* reaches the target area.

In the next two sections, we evaluated our model in four different trajectory-based tasks including a Steering task, a Goal-Crossing task, a Steering-Crossing task and a Randomized Steering task.

Appendix B Study 1: Modeling Evaluation in Steering Task

We begin by evaluating the Steering task because it is a standard trajectory-based task, and thus we can use the data from this study to estimate model parameters for subsequent evaluations.

Appendix B.1 Task Description

As shown in Figure B1, the Steering task requires participants to control a cursor to pass through a straight tunnel with boundaries on top and bottom. The participants must navigate the cursor from one side of the tunnel to the other without touching the boundaries of the tunnel or lifting their hands off the screen.

Appendix B.2 Participants and Apparatus

All the experiments in this paper were conducted on a Dell M4800 laptop computer with an Inter Core *i7* 4810 *MQ* CPU at 2.8 *GHz*. The computer was connected to a Wacom pen display with a stylus as input. The pen display is a direct interactive screen that can only be operated by the stylus. The pen display is 294 *mm* \times 165 *mm* (13.3 *inches*) in size and 1920 \times 1080 *px* in resolution. Each pixel on the screen is 0.153 *mm* wide. The stylus is 15.4 *cm* in length and 10 *g* in weight. The system ran with a sampling frequency of 100 *Hz*. The pen display was positioned on a table in "stand" mode (40° angle). The experiment programs in this study were developed using Unity 3D with C# code. We recruited 12 right-handed participants (8 males and 4 females, with an average age of 26.83) in this study. All of them were familiar with computer and stylus use, and all of them had normal or corrected-to-normal vision.



Figure B1 Layout of the interface used for Steering Task

Appendix B.3 Procedure and Design

We tested the combinations of five goals' width (W) and three goals' length (A): $A = 540, 760, 980$ px; $W = 46, 69, 92, 115, 138$ px. All the movement directions were from left to right. The total number of recorded trials was 15 conditions \times 15 repeats \times 12 participants = 2700 trials. For all the trials in the Steering task, and the following the Goal-Crossing task, the Steering-Crossing task, and Randomized Steering task, once the participants failed to complete a trial, they had to do it again. We only analyzed successful trials in this study.

Appendix B.4 Parameter-Setting

The following set of parameters needs to be optimized in our model [$k, \eta, N_{per}, V_{pref}$]. We used a genetic algorithm with a population size of 50 to optimize the parameters k and η . To produce better simulations, the parameters k and η for each task are optimized individually. To estimate parameters N_{per} in the Movement Uncertainty Model (Eq. A9), we performed a linear regression analysis between instantaneous speed and corresponding SD σ of the perturbing force. The parameters of the V_{pref} in the preferred speed model (Eq. A13) was obtained by performing a linear regression analysis between tunnel widths and the average speed of the particle in each tunnel.

Appendix B.4.1 Parameters of APF

There are two parameters k and η needed to be optimized for APF. To optimize the parameters k and η , we used a cost function J , which is defined as the Mean Absolute Error (MAE) between the actual trajectories and simulated trajectories:

$$J = MAE_{mean} = \frac{\sum_{i=1}^n (\|P_{actu}(i) - P_{sim}(i)\|)}{n}, n = 1, 2, \dots, 100, \quad (B1)$$

where $\|P_{actu}(i) - P_{sim}(i)\|$ is the Euclidean distance between sampling points of the actual and simulated trajectories, n is the number of MT% sampling points. We used a genetic algorithm with a population size of 50 to optimize the parameters k and η . The final estimated parameters k, η in Study 1 are shown in Table B1.

Appendix B.4.2 Parameters of Perturbing Force

We used the collected data in Steering task to estimate parameters a and b in the Movement Uncertainty Model (Eq. A9). To do so, we need actual data of instantaneous speed V_i and corresponding SD of the perturbing force σ . To obtain such data, we resampled the 2700 trials, converting each trajectory to 100 MT% sampling points. The MT% sampling points were obtained by interpolating the original trajectory data at the time from 1 to 100% of the total movement time. Then we calculated instantaneous speed (V_i) of each sampling point as:

$$V_i = \frac{\|P_{i+1} - P_{i-1}\|}{t_{i+1} - t_{i-1}}, (i = 2, 3, \dots, 99), \quad (B2)$$

where $\|P_{i+1} - P_{i-1}\|$ is the distance between the previous sampling point and the next sampling point.

To estimate the SD σ of the perturbing force in a certain instantaneous speed, we need enough perturbing force data in a specified speed condition to estimate the SD of the force data. Because we can not obtain the actual perturbing force from participants, we used the second derivative of small displacement perpendicular to the moving direction to approximate the perturbing force (F_{per}):

$$F_{per} = ma = m \frac{d^2 s}{dt^2}, \quad (B3)$$

where ds is a small displacement, m is the quality of the Particle. Because m is a constant, the perturbing force (F_{per}) is proportional to the second derivative of the displacement ds , thus can be directly obtained by calculating ds . The small displacement ds is calculated by getting the deviation between actual position P_{i+1} and desired position $P_{d(i+1)}$ at time t_{i+1} in the axis perpendicular to the moving direction.

$$ds = \frac{\overrightarrow{P_{i+1}P_{d(i+1)}} \cdot \mathbf{n}_{pv}}{\left\| \overrightarrow{P_{i+1}P_{d(i+1)}} \right\|}, \quad (B4)$$

where $\overrightarrow{P_{i+1}P_{d(i+1)}}$ is a vector from P_{i+1} to $P_{d(i+1)}$, \mathbf{n}_{pv} is the normal vector of the current moving direction, $\left\| \overrightarrow{P_{i+1}P_{d(i+1)}} \right\|$ is the norm of vector $\overrightarrow{P_{i+1}P_{d(i+1)}}$. The desired position $P_{d(i+1)}$ is given by multiplying the instantaneous speed V_i of the sampling point at time t_i by the time interval:

$$P_{d(i+1)} = V_i(t_{i+1} - t_i), (i = 2, 3, \dots, 99). \quad (B5)$$

Through the above steps, we obtained 232055 data of (*InstantaneousSpeed, PerturbingForce*). We then binned the data with an instantaneous speed interval of 10 px/s. This process yielded a dataset that includes 251 instantaneous speed conditions, each of which contains a number of perturbing force data (range from 81 to 1848). By calculating the SD of the perturbing forces for each instantaneous speed condition, we obtained 251 pairs of $\sigma \times V$ data. Finally, we fit Eq. A9 with the $\sigma \times V$ data and got a R^2 of 0.8772 as shown in Figure B2 (a). The final estimated parameter N_{per} is shown in Table B1.

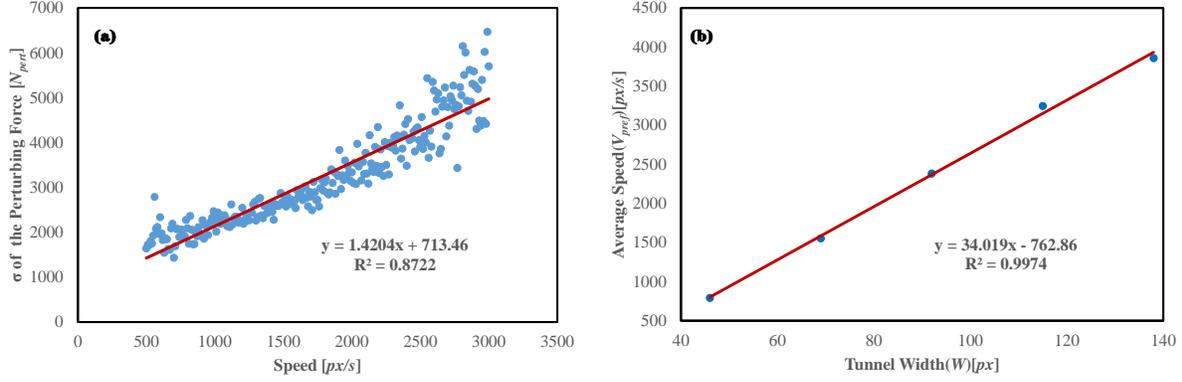

Figure B2 Fitting function of empirical models ((a) N_{pert} , (b) V_{pref})

Table B1 Parameter-Setting estimated in Steering Task

Parameters	Value	Methods of Computation	Ref.
k	9.5	Genetic Algorithm	[4, 13]
η	2.5×10^3	Genetic Algorithm	[4, 13]
N_{per}	$N_{per} = 1.4204V + 713.46$	Eq.(A9)(B2)(B3)(B4)(B5)	[10, 11]
V_{pref}	$V_{pref} = 34.019W - 762.86$	Eq.(A13)	[12]

Table B2 The MAE between simulated and actual movement time in Steering Task (MAE)[ms]

	$W = 46 \text{ px}$	$W = 69 \text{ px}$	$W = 92 \text{ px}$	$W = 115 \text{ px}$	$W = 138 \text{ px}$	Average
$A = 540 \text{ px}$	31.00	49.27	6.12	28.44	24.71	27.91
$A = 760 \text{ px}$	48.65	0.66	12.52	22.52	35.80	24.03
$A = 980 \text{ px}$	168.20	24.26	50.02	18.42	30.98	58.38
Average	82.62	24.73	22.89	23.13	30.50	36.77

Appendix B.4.3 Parameters of the Preferred Speed Model

We used the collected data in Steering task to estimate parameters of the preferred speed model (Eq. A13). We performed a linear regression analysis between tunnel widths and the average speed of the particle in each tunnel. The average speed of the particle is calculated by dividing the tunnel's length by the time it takes to pass through it. The fitted model V_{pref} that reflects the relation between the preferred speed and different tunnel widths is illustrated in Figure B2 (b). The final estimated parameter V_{pref} is shown in Table B1.

Appendix B.5 Results

We simulated the trajectories for the Steering task using the optimized parameters in Table B1. The first status of the actual data is used as the initial status of our model, and the model then updates the particle's movement status frame by frame until the particle encounters obstacles or successfully crosses the goal. The update interval was empirically set to 40 ms. We simulated 2700 trajectories and then tested our model by comparing the simulated data to the actual data in terms of movement time, mean trajectory, and movement uncertainty.

Appendix B.5.1 Movement Time

Since our model can simulate the whole movement, we can easily obtain simulated movement time of the model from its simulation. We used mean absolute error (MAE) between simulated movement time and actual movement time to evaluate the model's predictive power for movement time. As shown in Table B2, the MAE of movement time for Steering task is 36.77 ms, accounting for 6.92% of the total movement time. The minimum value of the offset occurs when $W = 69 \text{ px}$ and $A = 760 \text{ px}$ with an average offset of 0.66 ms (0.12%), while the maximum offset obtained when $W = 46 \text{ px}$ and $A = 980 \text{ px}$ with an average offset of 168.20 ms (14.73%). Our model shows the best movement time fitting when A and W are at a moderate level.

By comparing the goodness of fits of our model ($R^2 = 0.952$) and Steering Law ($R^2 = 0.962$), we found that our model is competitive with Steering Law in movement time prediction.

Table B3 The MAE between simulated and actual mean trajectories in Steering Task (MAE)[px]

	$W = 46 px$	$W = 69 px$	$W = 92 px$	$W = 115 px$	$W = 138 px$	<i>Average</i>
$A = 540 px$	7.96	20.30	22.93	19.60	25.80	19.32
$A = 760 px$	33.28	28.46	26.56	35.99	39.68	32.79
$A = 980 px$	63.82	43.64	45.11	59.74	74.29	57.32
<i>Average</i>	35.02	30.80	31.53	38.44	46.59	36.48

Table B4 The MAE of movement variability between simulated and actual trajectories in Steering Task (MAE_{MV})[px]

	$W = 46 px$	$W = 69 px$	$W = 92 px$	$W = 115 px$	$W = 138 px$	<i>Average</i>
$A = 540 px$	2.32	3.23	3.60	2.38	3.41	2.99
$A = 760 px$	2.58	4.41	3.81	3.56	5.73	4.02
$A = 980 px$	1.37	3.71	2.52	5.10	6.38	3.81
<i>Average</i>	2.09	3.78	3.31	3.68	5.17	3.61

Appendix B.5.2 Mean Trajectory

We used mean absolute error (MAE, Eq. B1) between simulated mean trajectory and actual mean trajectory to evaluate the performance of our model in predicting mean movement trajectory. The MAE in each $A \times W$ condition can be seen in Table B3. The average MAE is 36.48 px (5.59 mm), accounting for 4.93 % of the total length of actual trajectories. The minimum value of the offset occurs when $W = 46 px$ and $A = 540 px$ with an average offset of 7.96 px (1.56 %). And the maximum offset is obtained when $W = 138 px$ and $A = 980 px$ with an average offset of 74.29 px (8.07 %). These results indicate that the proposed model can well predict the mean movement trajectory in the Steering task.

Appendix B.5.3 Movement Uncertainty

Movement Variability (MV) is obtained by calculating the distance between the sample point and the horizontal axis [14–16], and is used to describe the extent to which the sample points lie in a straight line along the axis. In this paper, considering different types of scenarios, we used the mean trajectory instead of the axis as a reference and calculated MV by computing the Euclidean distance of each sampling point between the mean trajectory and the trajectories (actual and simulated). And then, we use the MAE of movement variability (MAE_{MV}) between the actual (MV_{actu}) and simulated trajectories (MV_{sim}) as a criterion for evaluating movement uncertainty.

$$\begin{aligned}
 MAE_{MV} &= |MV_{actu} - MV_{sim}| \\
 &= \left| \frac{1}{m} \sum_{i=1}^m \frac{\sum_{i=1}^n (|P_{actu}(i) - \bar{P}_{actu}(i)|)}{n} \right. \\
 &\quad \left. - \frac{1}{m} \sum_{i=1}^m \frac{\sum_{i=1}^n (|P_{sim}(i) - \bar{P}_{sim}(i)|)}{n} \right|, n = 1, 2, \dots, 100, m = 1, 2, \dots, m,
 \end{aligned} \tag{B6}$$

where $|MV_{actu} - MV_{sim}|$ is the absolute error between the MV of sampling points of the actual and simulated trajectories. $P_{actu}(i)$ and $P_{sim}(i)$ are the position of the sampling points on the actual and simulated trajectories, respectively. $\bar{P}_{actu}(i)$ and $\bar{P}_{sim}(i)$ are the position of the sampling points on the actual and simulated mean trajectories, respectively. $|P_{actu}(i) - \bar{P}_{actu}(i)|$ and $|P_{sim}(i) - \bar{P}_{sim}(i)|$ are the Euclidean distance between the mean trajectory and the trajectories (actual and simulated), respectively. n is the number of MT% sampling points. m is the number of trajectories.

Figure B3 shows mean trajectories for all $W \times A$ conditions and trajectory variabilities in these conditions. The absolute error of movement variability (MAE_{MV}) between the actual (MV_{actu}) and simulated (MV_{sim}) trajectories is 3.61 px (0.62 mm), the movement variability of the actual trajectories is 8.13 px (1.40 mm), the movement variability of the simulated trajectories is 11.74 px (2.02 mm). The MAE_{MV} in each $A \times W$ conditions can be seen in Table B4.

Appendix B.5.4 Repeated K -fold Cross-validation

To test the generalization ability of our model, we performed repeated k -fold cross-validation ($k = 2$, repeated times $N = 10$) for movement time prediction and mean trajectory prediction. The process of the validation is as follows: 1) Shuffle the dataset randomly; 2) Split the dataset into 2 groups; 3) For each unique group: i) Take the group as a hold out or test dataset; ii) Take the remaining groups as a training dataset; iii) Fit the model on the training set and evaluate it on the test set; iv) Retain the evaluation scores or both movement time prediction and mean trajectory simulation (will be described below) and discard the model; 4) Summarize the performance of the model using the sample of model evaluation scores.

For movement time prediction, the evaluation score is the MAE between simulated movement time and actual movement time, and the average MAE is 64.66 ms ($SD = 2.51$). For mean trajectory prediction, the evaluation score is the MAE between simulated mean trajectory and actual mean trajectory (Eq. B1), and the average MAE is 37.66 px ($SD = 0.68$). These results indicate that our model is generalizable across the conditions.

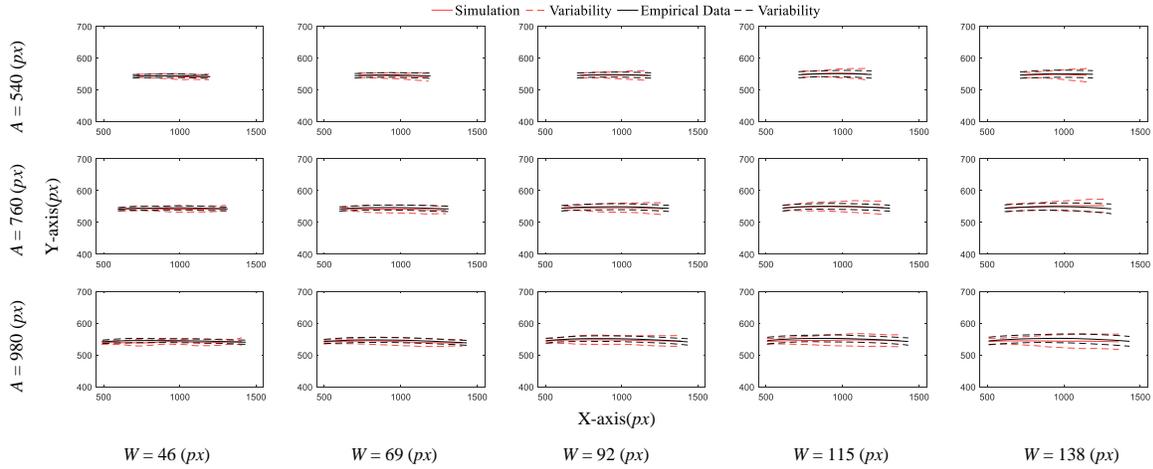


Figure B3 Actual (black) and simulated (red) mean trajectories and trajectory variabilities for the 15 $W \times A$ conditions in Steering Task. Solid lines represent the mean trajectories while the trajectory variabilities are plotted as dashed lines. The trajectory variabilities were determined by 95% CI of the trajectories.

Appendix C Study 2: Model Evaluation in Goal-Crossing, Steering-Crossing and Randomized Steering Tasks

In this section, we examined our model's performance in the Goal Crossing, Steering-Crossing, and Randomized Steering tasks.

Appendix C.1 Goal Crossing Task

Appendix C.1.1 Task Description

Goal Crossing tasks (Figure C1) require users to cross two goals in a certain distance. Participants start the task by crossing the left goal and complete it by crossing the right goal. Compared to Steering tasks, Goal crossing tasks have no tunnel boundaries. Since there is no risk of errors caused by touching boundaries, users tend to move the input device faster when performing Goal Crossing tasks.

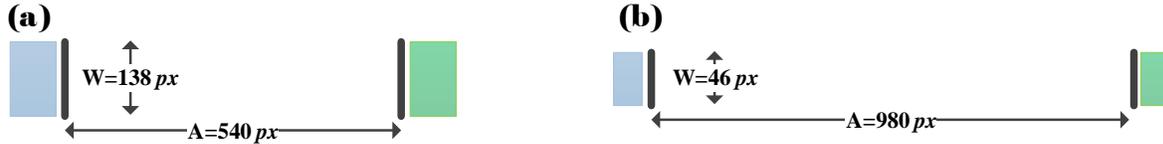


Figure C1 Layout of the interface used for Goal Crossing Task

Appendix C.1.2 Experiment Design

The apparatus in Study 2 was the same as Study 1. We recruited 20 right-handed participants (10 males and 10 females, with an average age of 23.35) in Study 2. All of the participants were familiar with using the computer and stylus. All of them had normal or corrected-to-normal vision. None of them took part in Study 1.

We tested the combinations of five values of goals' width (W) and three values of goals' distance (A): $A = 540, 760, 980$ px; $W = 46, 69, 92, 115, 138$ px. The total number of recorded trials was 3 widths \times 5 distance \times 15 repeats \times 20 participants = 4500 trials. We used the same parameters of the movement uncertainty model and preferred speed model in Study 1, while the parameters (Eq. B1) of APF were optimized with newly collected data in this task ($[k = 26, \eta = 3 \times 10^4]$). We simulated 4500 trajectories using our model, and evaluated our model by comparing the simulated data with the actual data in terms of movement time, mean trajectory and movement uncertainty. Measures of model performances in movement time prediction, mean trajectory and movement uncertainty simulations are the same as Study 1.

Appendix C.1.3 Movement Time

As shown in Table C1, the average MAE of movement time is 17.20 ms, accounting for 5.65% of the total movement time. Simulation results show that the minimum value of the offset occurs when $W = 69$ px and $A = 760$ px with an average offset of 0.55 ms (0.12 %). The maximum offset is obtained when $W = 92$ px and $A = 540$ px with an average offset of 44.80 ms (19.70 %). Our model fits the movement time data ($R^2 = 0.954$) slightly better than Fitts' Law ($R^2 = 0.948$).

Appendix C.1.4 Mean Trajectory

As shown in Table C2, The average MAE of mean trajectory is 68.28 px (10.16mm), accounting for 9.60 % of the total length. The minimum value of the offset occurs when $W = 69$ px and $A = 540$ px with an average offset of 30.76 px (6.53 %), while the maximum offset is obtained when $W = 138$ px and $A = 980$ px with an average offset of 110.02 px (12.32 %).

Table C1 The MAE between simulated and actual movement time in Goal Crossing Task (MAE)[ms]

	$W = 46 px$	$W = 69 px$	$W = 92 px$	$W = 115 px$	$W = 138 px$	<i>Average</i>
$A = 540 px$	10.41	9.59	44.80	7.61	16.90	17.86
$A = 760 px$	9.61	0.55	5.69	6.94	42.48	13.05
$A = 980 px$	44.46	5.72	6.32	25.24	21.63	20.69
<i>Average</i>	21.49	5.29	18.94	13.26	27.02	17.20

Table C2 The MAE between simulated and actual mean trajectories in Goal Crossing Task (MAE)[px]

	$W = 46 px$	$W = 69 px$	$W = 92 px$	$W = 115 px$	$W = 138 px$	<i>Average</i>
$A = 540 px$	34.04	30.76	38.60	39.97	42.45	37.16
$A = 760 px$	62.77	65.16	61.65	60.06	75.83	65.09
$A = 980 px$	105.95	111.28	97.00	88.64	110.02	102.58
<i>Average</i>	67.59	69.07	65.75	62.89	76.10	68.28

Table C3 The MAE of movement variability between simulated and actual trajectories in Goal-Crossing Task (MAE_{MV})[px]

	$W = 46 px$	$W = 69 px$	$W = 92 px$	$W = 115 px$	$W = 138 px$	<i>Average</i>
$A = 540 px$	14.57	9.68	8.11	5.34	5.38	8.61
$A = 760 px$	26.56	24.01	16.78	14.45	18.49	20.06
$A = 980 px$	41.03	39.60	29.57	23.66	29.78	32.73
<i>Average</i>	27.38	24.43	18.15	14.48	17.89	20.47

Appendix C.1.5 Movement Uncertainty

The mean absolute error (MAE) of movement variability (MAE_{MV}) between the actual (MV_{actu}) and simulated (MV_{sim}) trajectories in this task is 20.47 px (3.52 mm), the movement variability of the actual trajectories is 40.41 px (6.95 mm), the movement variability of the simulated trajectories is 19.95 px (3.43 mm). The MAE_{MV} in each $A \times W$ conditions can be seen in Table C3. Figure C2 shows mean trajectories for all $W \times A$ conditions and trajectory variabilities in these conditions.

Appendix C.1.6 Repeated K-fold Cross-validation

Similar to Study 1, we performed repeated k-fold cross-validation ($k = 2$, repeated times $N = 10$) for movement time prediction and mean trajectory simulation. For movement time prediction, the average MAE is 36.12 ms ($SD = 1.46$). For trajectory simulation, the average MAE is 73.44 px ($SD = 0.39$).

Appendix C.2 Steering-Crossing Task

Appendix C.2.1 Task Description

Steering-Crossing tasks require users to steer through multiple constrained path segments. Such motions can be used to lasso objects in illustration or presentation software. In the work of [3], researchers found that user performance varies by path segment width (W), segment size (S), segment interval (I), and the layout of the segments. Generally, users behave similarly to the Steering tasks when passing through the obstacles, and more like Crossing tasks when passing through the interval. Furthermore, the Steering-Crossing task is relatively easy to expand upon, which means that it can be used to simulate different interaction scenarios by changing the settings of the interval and the size of obstacles.

We evaluated our model in the Steering-Crossing Task with 3 typical settings: 1) Task 3 (a) (Figure C3 (a)): 3 path segments and 2 intervals with $S = 390 px$ and $I = 293 px$; 2) Task 3 (b) (Figure C3 (b)): 2 path segments and 3 intervals with $S = 390 px$ and $I = 293 px$; 3) Task 3 (c) (Figure C3 (c)): 40 path segments and 39 intervals with $S = 32 px$ and $I = 13 px$. The path segment width was fixed at $W = 46 px$.

We did not discuss the impact of width (W) in this study because it has been extensively examined in previous work [3]. Meanwhile, evidence suggests that the size of the segment interval (I) influences the navigating strategy (i.e., Goal-Crossing strategy or Steering strategy) used to complete the task. To evaluate the effectiveness of our model for simulating different motion processes, we designed scenarios with large I (Figure C3 (a) and (b)) and small I (Figure C3 (c)) at a specific W . Besides, as segment sizes (S) generally do not have significant effects on the motion process [3], we only tested whether our model can handle different segment layouts (Figure C3 (a) and (b)).

Appendix C.2.2 Experiment Design

The recorded data for this task were 3 settings \times 15 repeats \times 20 participants = 900 trials. We used the same parameters of the movement uncertainty model and preferred speed model in Study 1, while the parameters (Eq. B1) of APF were optimized with newly collected data in this task ($[k = 1, \eta = 6 \times 10^4]$). We simulated 900 trajectories using our model. We also evaluated our model by comparing the simulated data with the actual data in terms of movement time, mean trajectory and movement uncertainty.

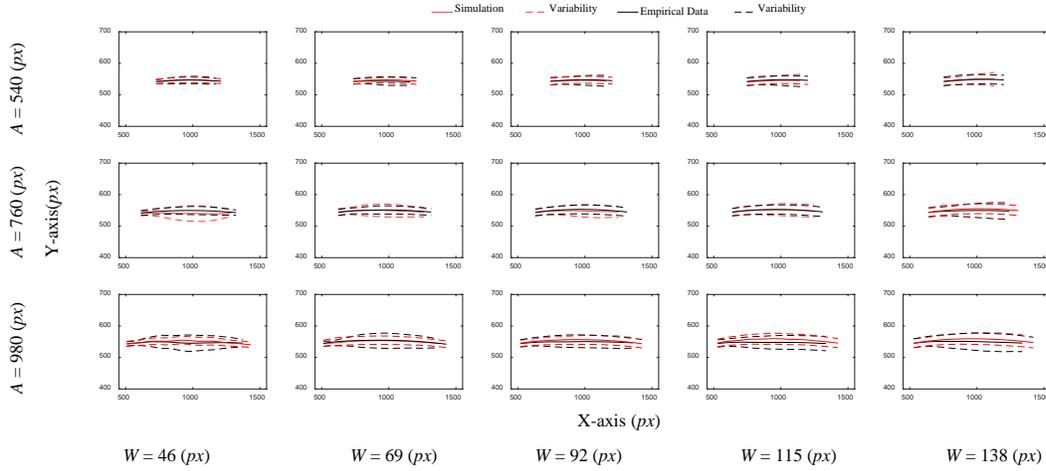


Figure C2 Actual (black) and simulated (red) mean trajectories and trajectory variabilities for the 15 $W \times A$ conditions in Goal Crossing Task. Solid lines represent the mean trajectories while the trajectory variabilities are plotted as dashed lines. The trajectory variabilities were determined by 95% CI of the trajectories.

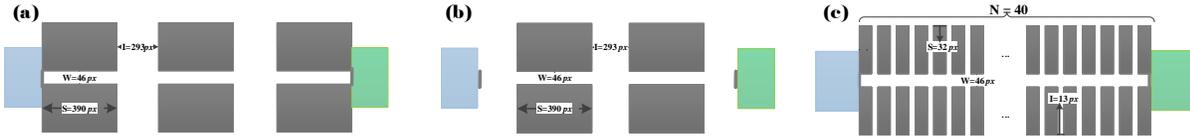


Figure C3 Layout of the interface used for Steering-Crossing Task

Table C4 The MAE of movement time and mean trajectory between simulated and actual in Steering-Crossing Task

	$Task3(a)$	$Task3(b)$	$Task3(c)$	$Average$
Movement Time (MAE)[ms]	75.38	129.32	55.07	86.59
Mean Trajectories (MAE)[px]	61.28	128.03	207.39	132.23
Movement variability MAE_{MV} [px]	2.88	2.46	3.29	2.88

Measures of model performances in movement time prediction, mean trajectory and movement uncertainty simulations are the same as Study 1.

Appendix C.2.3 Movement Time

As shown in the first row of Table C4, the average MAE of the movement time for Steering-Crossing task is 86.59 ms, accounting for 4.42% of the total movement time. Among the three conditions, we obtained the best time fitting in the third condition with an average offset of 55.07 ms (2.53 %).

We compared our model with Steering Law, Crossing Law [1], and the model introduced by Shota Yamanaka [3] in movement time prediction. The Shota Yamanaka’s model gained the highest R^2 with a value approaching 1, followed by our model with a comparable R^2 value of 0.969. The Steering Law and Crossing Law had a relatively low fit in the Steering-Crossing task. The reason for the high R^2 of the Shota Yamanaka’s model is that, by combining Steering Law and Fitts’ Law, the Shota Yamanaka’s model [3] introduced additional ID and parameters, which helps the model to obtain high fits when there are less experimental conditions in the experiment.

Appendix C.2.4 Mean Trajectory

As shown in the second row of Table C4, the MAE of mean trajectory is 132.23 px (20.28 mm), accounting for 7.64 % of the overall length of the trajectory. Unlike the results of time fitting, the MAE of mean trajectory of the third condition is the largest (207.39 px, 31.81 mm), accounting for 11.75 % of the total length of the actual mean trajectory.

Appendix C.2.5 Movement Uncertainty

As shown in the third row of Table C4, the mean absolute error (MAE) of movement variability (MAE_{MV}) between the actual (MV_{actu}) and simulated (MV_{sim}) trajectories in this task is 2.88 px (0.57 mm), the movement variability of the actual trajectories is 5.13 px (0.85 mm), the movement variability of the simulated trajectories is 8.01 px (1.42 mm). Figure C4 shows mean trajectories for the 3 settings in Figure C3 and trajectory variabilities in these conditions.

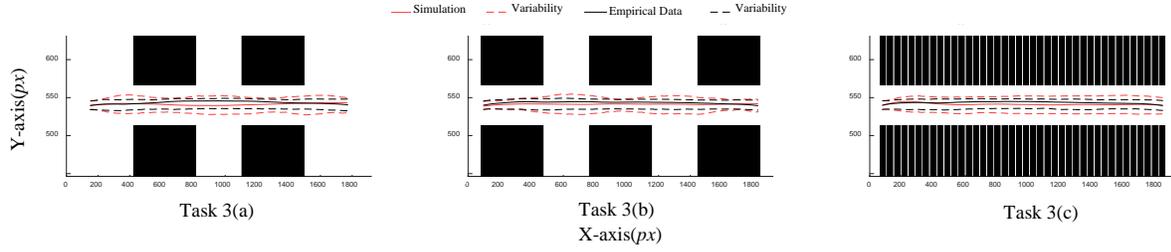


Figure C4 Simulation results of movement uncertainty in Steering-Crossing Task.

Appendix C.2.6 Repeated K-fold Cross-validation

Similar to Study 1, we performed repeated k-fold cross-validation ($k = 2$, repeated times $N = 10$) for movement time prediction and mean trajectory simulation. For movement time prediction, the average MAE is 141.91 ms ($SD = 3.02$). For trajectory simulation, the average MAE is 202.74 px ($SD = 1.11$).

Appendix C.3 Randomized Steering Task

Appendix C.3.1 Task Description

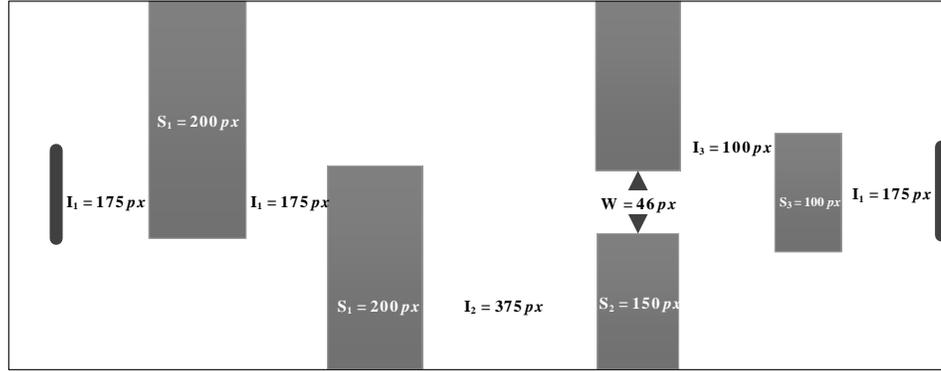


Figure C5 Layout of the interface used for Randomized Steering Task

We created a scenario with randomly arranged obstacles to test our model’s ability in dealing with complicated trajectory-based tasks. This task is similar to lassoing haphazardly organized icons on a user interface. In this task, we set up three different Interval ($I_1 = 175\text{ px}$, $I_2 = 375\text{ px}$, $I_3 = 100\text{ px}$), three different Obstacle Size ($S_1 = 200\text{ px}$, $S_2 = 150\text{ px}$, $S_3 = 100\text{ px}$) and one Width ($W = 46\text{ px}$). Overall, the straight line distance from the start bar to the end bar is 1650 px , and the narrowest tunnel on the path has a straight line width of 46 px . The specific task settings are shown in Figure C5.

The final obstacle on the right splits the path to the final destination into two paths: one goes above the obstacle, while the other goes below it. We intentionally designed this type of path to test if our model can simulate user behavior for route choice.

Appendix C.3.2 Experiment Design

20 participants repeated the task 15 times, resulting 300 trials in all. We used the same parameters of the movement uncertainty model and preferred speed model in Study 1, while the parameters (Eq. B1) of APF were optimized with newly collected data in this task ($[k = 35, \eta = 9 \times 10^6]$). We simulated 300 trajectories using our model and evaluated our model by comparing the simulated data with the actual data using the same methods as previous tasks.

Appendix C.3.3 Movement Time

Unexpectedly, when circumventing the final obstacle near the goal, participants tended to choose one path randomly, as shown in Figure C6. This may be mainly because the cost of the two routes to the goal is similar. This uncertain choice of route splits the actual trajectory data into two clusters: upper trajectories and lower trajectories. Therefore, we analyzed the two clusters of trajectories respectively.

As shown in Table C5, the MAE of movement time for upper and lower trajectories is 245.33 ms and 273.93 ms respectively, accounting for 8.75 % and 9.77% of the total time cost. There is no previous HCI model which can predict the movement time for this task, thus we do not conduct horizontal comparison between models.

Appendix C.3.4 Mean Trajectory

Simulated upper and lower mean trajectories are shown by red and yellow solid lines in Figure C6. The MAE of the mean trajectory for upper and lower trajectories is 241.11 px (36.97 mm) and 295.75 px (45.35 mm) respectively, accounting for 9.58 % and 11.30 % of the overall length of the actual mean trajectories.

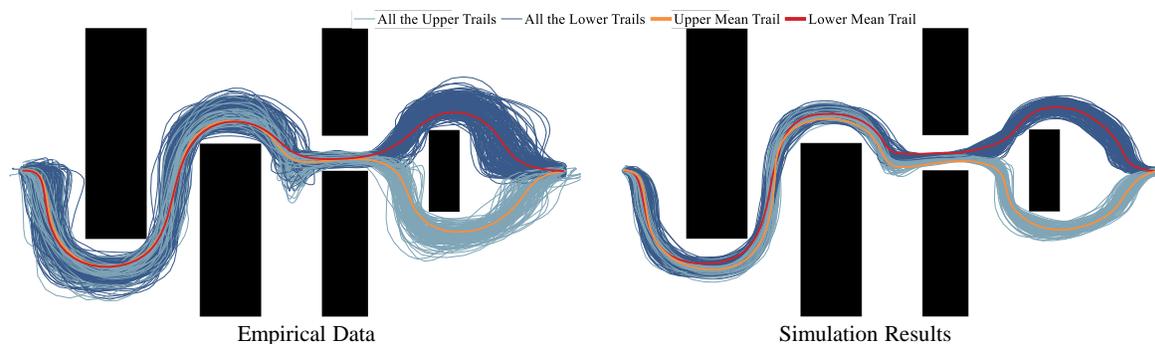


Figure C6 Simulation results in Randomized Steering Task

Table C5 The MAE between simulated and actual mean trajectories in Randomized Steering Task

	Upper Trials in Randomized Steering Task	Lower Trials in Randomized Steering Task
Movement Time (MAE)[ms]	245.33	273.93
Mean Trajectories (MAE)[px]	241.11	295.75
Movement Variability (MAE_{MV})[px]	122.32	83.49

Appendix C.3.5 Movement Uncertainty

The mean absolute error (MAE) of movement variability (MAE_{MV}) for upper and lower trajectories between the actual (MV_{actu}) and simulated (MV_{sim}) trajectories in this task is 122.32 px (21.04 mm) and 83.49 px (14.36 mm), respectively. The movement variability for upper and lower trajectories of the actual trajectories is 296.07 px (50.92 mm) and 231.84 px (39.87 mm), respectively. The movement variability for upper and lower trajectories of the simulated trajectories is 173.75 px (29.88 mm) and 148.35 px (25.51 mm), respectively. The simulated upper and lower trajectories are represented by dark blue and light blue lines in Figure C6. As shown in Figure C6, the simulated trajectories have a similar pattern of movement uncertainty compared with the actual trajectories. Also, the movement uncertainty is larger in relatively open spaces (e.g., the beginning and ending spaces of the task) while it becomes smaller in narrow tunnels (e.g., the second column of obstacles on the right). Meanwhile, our model successfully simulates the uncertainty of human behavior in choosing the way to the final goal. As the actual data, the simulated data also split into two clusters of trajectories and both finally reach the ending area of the task.

We also found that the simulated trajectories have generally smaller uncertainty compared to the actual data. The smaller vibration amplitude of simulated trajectories could be due to the relatively large η for *Repulsive Potential* in this task. A larger η could limit vibration of the trajectory in a smaller range.

Appendix C.3.6 Repeated K-fold Cross-validation

Similar to Study 1, we performed repeated k-fold cross-validation ($k = 2$, repeated times $N = 10$) for movement time prediction and mean trajectory simulation. For movement time prediction, the average MAE is 323.11 ms (SD = 2.25). For trajectory simulation, the average MAE is 365.19 px (SD = 1.90).

References

- Accot J, Zhai S. Beyond Fitts' Law: Models for Trajectory-Based HCI Tasks. CHI' 97, 1997: 295-302.
- Accot J, Zhai S. More than dotting the i's—foundations for crossing-based interfaces. CHI '02, 2002: 73-80.
- Yamanaka S, Stuerzlinger W, Miyashita H. Steering through Successive Objects. Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems - CHI '18, 2018: 1-13.
- Liu J, Li C. A new method to solve the local minimum problem of an artificial potential field (in Chinese). In The 15th china mechanical design convention. 2009:76-78,85.
- Kala R, Warwick K. Planning autonomous vehicles in the absence of speed lanes using lateral potentials. IEEE Intelligent Vehicles Symposium, 2012: 597-602
- Burt P J. Computer simulations of a dynamic visual perception model. International Journal of Man-Machine Studies, 1975,7 (4): 529-546.
- Xin Q, Hong L, Hao Z, Baoxi L. A collective motion model based on two-layer relationship mechanism for bi-direction pedestrian flow simulation. Simulation Modelling Practice and Theory, 2018, 84: 268-285
- Schmidt R A, Zelaznik H, Hawkins B, et al. Motor-output variability: a theory for the accuracy of rapid motor acts. Psychological review, 1979, 86(5): 415
- Sutton G G, Sykes K. The variation of hand tremor with force in healthy subjects. The Journal of physiology, 1967, 191(3): 699-711
- Todorov E. Optimality principles in sensorimotor control. Nature neuroscience, 2004, 7(9): 907-915
- Todorov E, Jordan M I. Optimal feedback control as a theory of motor coordination. Nature neuroscience, 2002, 5: 1226-1235
- Zhou X, Cao X, Ren X. Speed-accuracy tradeoff in trajectory-based tasks with temporal constraint. IFIP Conference on Human-Computer Interaction. Springer, Berlin, Heidelberg, 2009: 906-919
- Das M S, Sanyal S, Mandal S. Navigation of Multiple Robots in Formative Manner in an Unknown Environment using Artificial Potential Field Based Path Planning Algorithm[J]. Ain Shams Engineering Journal, 2022, 13(5): 101675.
- MacKenzie I S, Kauppinen T, Silfverberg M. Accuracy measures for evaluating computer pointing devices. CHI '01, 2001: 9-16
- Kim J, Wichmann T, Inan O T, et al. Fitts' Law Based Performance Metrics to Quantify Tremor in Individuals With Essential Tremor[J]. IEEE Journal of Biomedical and Health Informatics, 2021, 26(5): 2169-2179.

- 16 Yamanaka S, Usuba H, Miyashita H. Bivariate Effective Width Method to Improve the Normalization Capability for Subjective Speed-accuracy Biases in Rectangular-target Pointing[C]//CHI Conference on Human Factors in Computing Systems. 2022: 1-13.