Appendix A Method

Appendix A.1 Optimal Feedback Control System

The linear-quadratic-Gaussian OFC system with state $\boldsymbol{x}_t \in \mathbb{R}^m$, control $\boldsymbol{u}_t \in \mathbb{R}^n$, and feedback $\boldsymbol{y}_t \in \mathbb{R}^k$ in discrete time $t \in [1, N]$:

$$
\begin{aligned}
&Dynamics && \boldsymbol{x}_{t+1} = A\boldsymbol{x}_t + B\boldsymbol{u}_t + \boldsymbol{\xi}_t + \varepsilon_t C\boldsymbol{u}_t, \\
&Feedback && \boldsymbol{y}_{t+1} = H\boldsymbol{x}_t + \boldsymbol{\omega}_t, \\
&Cost && \sum_{t=1}^{n} \boldsymbol{x}_t^T Q_t \boldsymbol{x}_t + \boldsymbol{u}_t^T R \boldsymbol{u}_t.
\end{aligned}
$$

At each time step t, an optimal control $\boldsymbol{u}_t$ (simulating the user's control command) must be found that minimize the expected total cost in Eq.1 over the movement based on the currently obtained feedback $\boldsymbol{y}_t$. This process simulates the phenomenon in which a user continually adjusts their action while considering consumption, speed, and accuracy. After finding and sending a control command, the controlled force (simulating the force of the user's hand) changes to follow the command and pushes the mass point (simulating the pointing device) to move to the desired target. The terms $\boldsymbol{\xi}_t \in \mathbb{R}^m$ and $\boldsymbol{\omega}_t \in \mathbb{R}^k$ are independent multidimensional normal random variables with the mean 0 and covariances Ω^ξ and Ω^ω, which respectively simulate white noise in the control and observation processes. The other random variable ε_t is an independent 0-mean Gaussian with standard deviation, which is multiplied by the control vector to simulate control-dependent noise. R and Q_t are coefficient matrixes that define the system cost related to the state and control, respectively, which reflect tradeoffs between consumption, speed, and accuracy. The solution to this optimal control problem is based on the assumption that the estimate state is a summary of the entire history of the control and feedback signals [4]. We adopted this solution in this paper.

For a target-selection motion, we represent the cursor's state by the following ten-dimensional vector:

$$\boldsymbol{x}_t = [p_x(t); p_y(t); \dot{p}_x(t); \dot{p}_y(t); f_x(t); f_y(t); g_x(t); g_y(t); p_x^*(t); p_y^*(t)],$$

where $p_x(t)$, $p_y(t)$ represent the cursor's position, and $\dot{p}_x(t)$, $\dot{p}_y(t)$ represent its velocity on x-axis and y-axis, respectively. $f_x(t)$, $f_y(t)$ represent the force of the hand, and $g_x(t)$, $g_y(t)$ are auxiliary state needed to implement the second-order muscle filter [4], respectively. $p_x^*(t)$ and $p_y^*(t)$ represent the target's position, and we note that the state is defined in time step t, so we can change the position in each time step to formulate the moving target. The initial state $\boldsymbol{x}_1$ has a multivariate normal distribution with the mean 0 and covariance $\sum_1$.

The control signal that drives the cursor to the target is defined as follows:

$$\boldsymbol{u}_t = [u_x(t); u_y(t)].$$

The cursor's movement is simulated by the particle motion in a Newtonian mechanics system with the following discrete-time form ($d = x/y$):

$$
\begin{aligned}
&p_d(t+1) = p_d(t) + \dot{p}_d(t)\Delta, \\
&\dot{p}_d(t+1) = \dot{p}_d(t) + f_d(t)\Delta/m, \\
&f_d(t+1) = f_d(t)(1-\Delta/\tau_2) + g_d(t)\Delta/\tau_2, \\
&g_d(t+1) = g_d(t)(1-\Delta/\tau_1) + u_d(t)(1+\sigma_c\varepsilon_t)\Delta/\tau_1.
\end{aligned}
$$

This system can be transformed into Eq.1 using the following matrixes:

$$
A = \begin{bmatrix}
1 & . & \Delta & . & . & . & . & . & 0_{8\times 2} \\
. & 1 & . & \Delta & . & . & . & . & \\
. & . & 1 & . & \Delta/m & . & . & . & \\
. & . & . & 1 & . & \Delta/m & . & . & \\
. & . & . & . & 1-\Delta/\tau_2 & . & \Delta/\tau_2 & . & \\
. & . & . & . & . & 1-\Delta/\tau_2 & . & \Delta/\tau_2 & \\
. & . & . & . & . & . & 1-\Delta/\tau_1 & . & \\
. & . & . & . & . & . & . & 1-\Delta/\tau_1 & \\
0_{2\times 8} & & & & & & & & I_{2\times 2}
\end{bmatrix},
\quad
B = \begin{bmatrix} 0_{6\times 2} \\ \begin{bmatrix} \Delta/\tau_1 & . \\ . & \Delta/\tau_1 \end{bmatrix} \\ 0_{2\times 2} \end{bmatrix},
\quad C = B\sigma_c,
\quad H = \begin{bmatrix} I_{6\times 6} & 0_{6\times 4} \end{bmatrix}.
$$

As we can see, the control signal in the system is affected by the control-dependent noise ε_t weighted with parameter σ_c. We have assumed that user sensory feedback (including visual and kinesthetic) includes information about position, velocity, and force as follows:

$$\boldsymbol{y}_t = [p_x(t); p_y(t); \dot{p}_x(t); \dot{p}_y(t); f_x(t); f_y(t)].$$

We set the covariances of white noise $\boldsymbol{\xi}_t$ and $\boldsymbol{\omega}_t$ in Eq.1 as follows:

$$\Omega^\xi = 0;\ \Omega^\omega = (diag[\sigma_p; \sigma_p; \sigma_v; \sigma_v; \sigma_f; \sigma_f])^2\ ,$$

which assumes that the white noise in the control process is ignorable and that in the observation process has the standard deviations σ_p, σ_v and σ_f with respect to position, velocity, and force.

The state cost in each time step is defined as follows:

$$(p_x(t)-p_x^*(t))^2+(p_y(t)-p_y^*(t))^2+(\dot{p}_x(t)w_v)^2+(\dot{p}_y(t)w_v)^2+(f_x(t)w_f)^2+(f_y(t)w_f)^2,$$

and the control cost is:

$$r({u_x}^2+{u_y}^2).$$

By obtaining the sum of the state and control costs, we obtain the total system cost, which can be transformed into Eq.1 by the matrixes $R=r$ and Q_t, as follows:

$$Q_t=\tfrac{1}{N}\boldsymbol{q}^T\boldsymbol{q};\ \boldsymbol{q}=\begin{bmatrix} -1 & . & . & . & . & . & . & . & 1 & . \\ . & -1 & . & . & . & . & . & . & . & 1 \\ . & . & w_v & . & . & . & . & . & . & . \\ . & . & . & w_v & . & . & . & . & . & . \\ . & . & . & . & w_f & . & . & . & . & . \\ . & . & . & . & . & w_f & . & . & . & . \end{bmatrix}.$$

The parameters r, w_v and w_f weight the penalties of consumption, velocity, and force, and balance the speed, accuracy, and effort expended in the target acquisition process.

Appendix A.2 Parameter Estimations

We define the cost function $J(\theta)$ for estimating the key parameters as follows:
For a trajectory set containing M trajectories $\boldsymbol{traj}=\{traj_j|j=1,2...M\}$, the i_{th} point in the j_{th} trajectory is denoted by $p_{ij}=(x_{ij},y_{ij})$, and we can compute the mean point position of all i_{th} points in the trajectory set as follows:

$$\overline{p_i}=(\overline{x_i},\overline{y_i})=(\frac{1}{M}\sum_{j=1}^{M}x_{ij},\frac{1}{M}\sum_{j=1}^{M}y_{ij}).$$

We can also compute the variability of all the i_{th} points in the trajectory set as follows:

$$var_i=(\sqrt{\frac{1}{M}\sum_{j=1}^{M}(x_{ij}-\overline{x_i})^2},\sqrt{\frac{1}{M}\sum_{j=1}^{M}(y_{ij}-\overline{y_i})^2}).$$

We resampled all the simulated and experimental trajectories to have them contain the same number of points. Let s and e denote the simulated and experimental trajectory sets, respectively. Then, we can define the cost function as follows:

$$J(\theta)=trE\times vaE,$$

$$trE=\frac{1}{n}\sum_{i=1}^{n}||\overline{p_i^s}-\overline{p_i^e}||,$$

$$vaE=\frac{1}{n}\sum_{i=1}^{n}||var_i^s-var_i^e||.$$

As presented above, trE and vaE are the average Euler distances between the mean point position and variability along the trajectory, and we must find a θ^* value that minimizes product of trE and vaE.
We developed a random search algorithm to obtain θ^* by the following major steps:
(a) Given a searching region $\mu_i-\Delta_i<\theta_i<\mu_i+\Delta_i(i=1,2,...n)$,
(b) find T θ in this region that satisfies $J(\theta^{(1)})>J(\theta^{(2)})>...J(\theta^{(T)})$.
(c) Determine the new μ_i and Δ_i as follows:

$$\mu_i=\frac{\sum_{t=1}^{T}W^{(t)}\theta_i^{(t)}}{\sum_{t=1}^{T}W^{(t)}},\ \Delta_i=C\sqrt{\frac{\sum_{t=1}^{T}W^{(t)}{(\theta_i^{(t)}-\mu_i)}^2}{\sum_{t=1}^{T}W^{(t)}}},\ W^{(t)}=\frac{J(\theta^{(T)})}{J(\theta^{(t)})}\text{ , then}$$

(d) search in the new region until satisfying the following convergence condition or reaching the maximum number of iterations.

$$\left|\frac{J(\theta^{(1)})-J(\theta^{(T)})}{J(\theta^{(T)})}\right|<\varphi,$$

where, in our case, n is 7 and we empirically set T and C to 5 and 2.
For the other parameters in the OFC system, m, we set Δ, τ_1 and τ_2 to $m=1$kg, $\Delta=0.01\,\mathrm{sec}$ and $\tau_1=\tau_2=0.04\,\mathrm{sec}$ as the authors did in [4]. We empirically set $\sum_1$ to $(diag[10\mathrm{pixel},10\mathrm{pixel},10\mathrm{pixel/sec},10\mathrm{pixel/sec},1\mathrm{N},1\mathrm{N},0,0,0,0])^2$. Note that the unit pixel here is a logical pixel distance unit since the movement we are simulating is in a logical space in the device display. We set the number of time steps N to 1000.

Appendix B Results

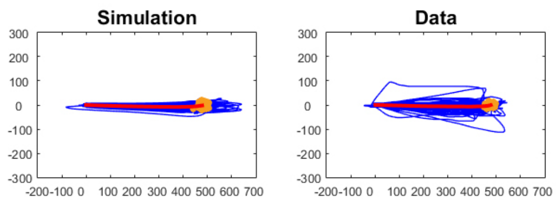

Figure B1 Static-target selection simulation results and empirical data. The red line represents the mean trajectory of the movements, the blue lines represent the trajectories of each trial, and the orange points represent the movement end-points.

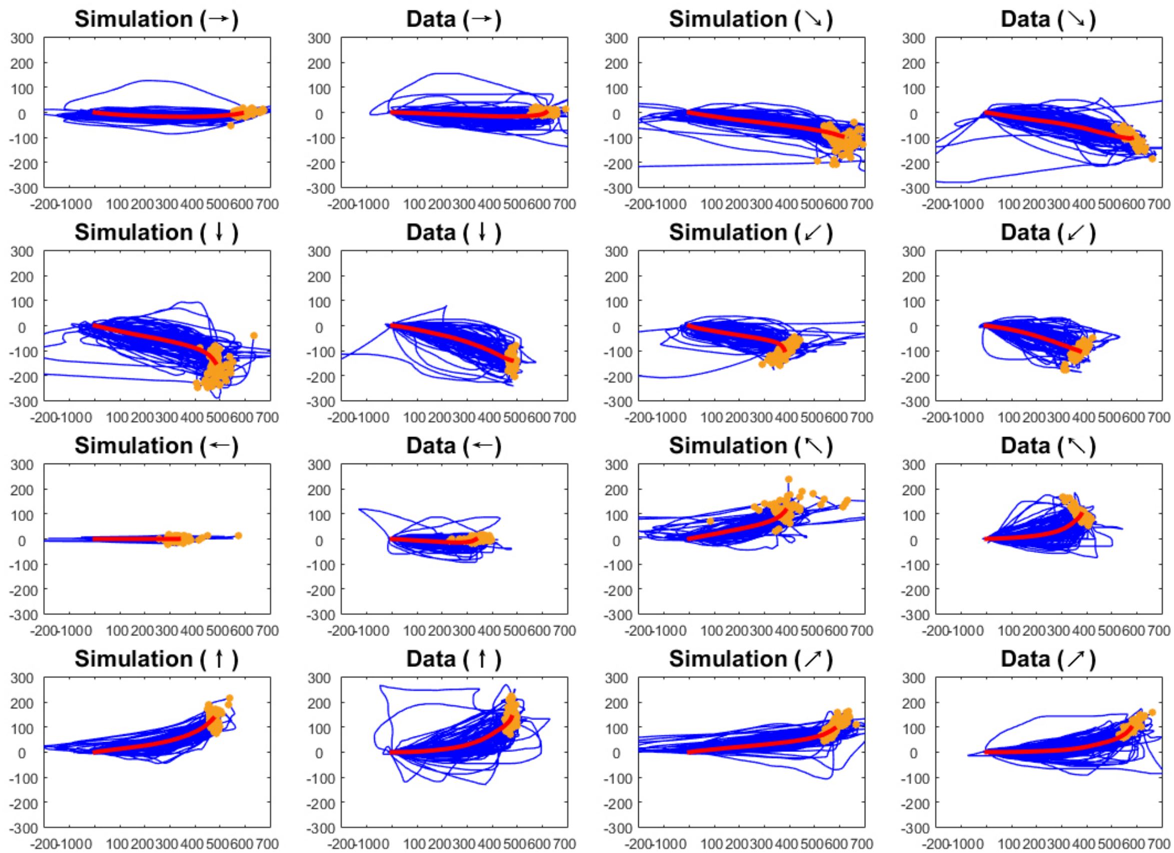

Figure B2 Moving-target selection simulation results and empirical data in eight directions. The red line represents the mean trajectory of the movements, the blue lines represent the trajectories for each trial, and the orange points represent the movement end-points.

Table B1 Estimated parameters for static- and moving-target selection

parameters	σ_c	σ_p	σ_v	σ_f	r	w_v	w_f
θ_s	5.4132	68.05	474.68	381.10	690.90	820.35	592.20
θ_m	10.3310	147.29	241.49	147.82	1149.71	646.96	176.89

Table B2 The errors of mean trajectory and trajectory variability for all tasks

measures	static	moving (→)	moving (↘)	moving (↓)	moving (↙)	moving (←)	moving (↖)	moving (↑)	moving (↗)
trE	9.86	30.51	45.02	52.07	22.14	23.13	37.04	38.86	44.93
vaE	17.89	46.02	40.32	30.98	27.03	27.14	28.80	49.75	29.82