MC2019 Part 2 Optimal control for stochastic systems.

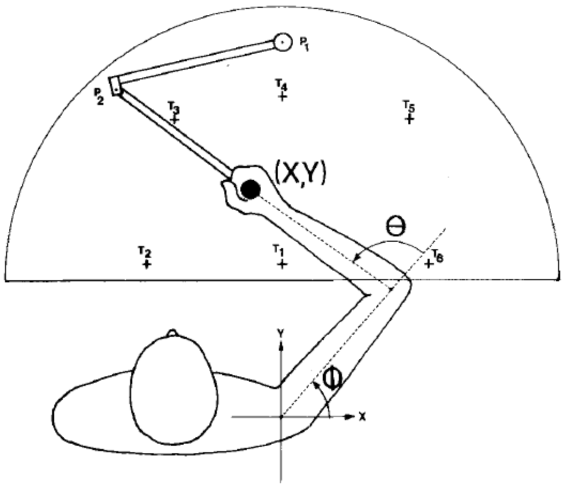
Hirokazu Tanaka

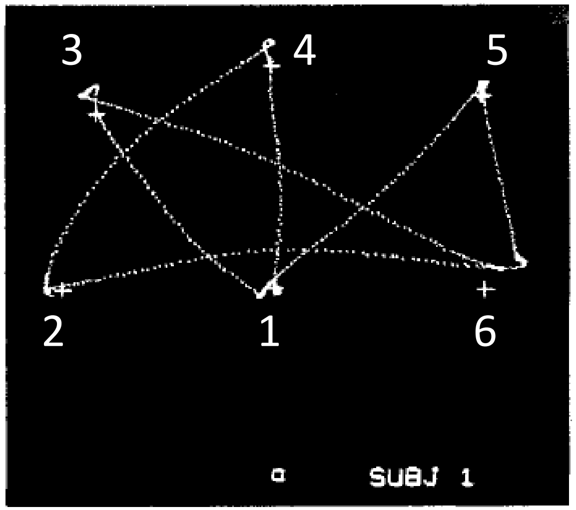
Optimal control for stochastic systems.

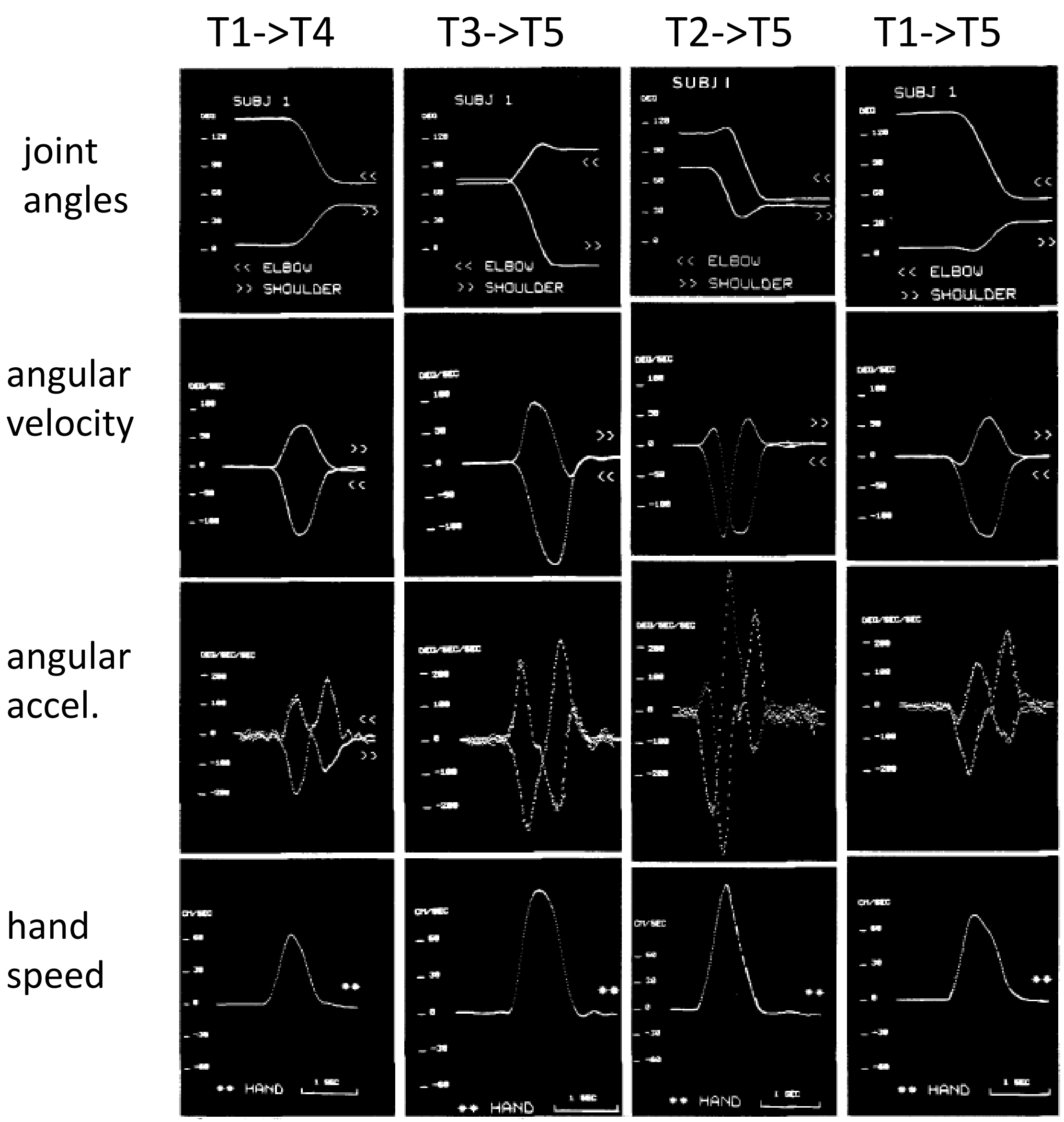
In this lecture, we will learn:

- Feedforward and feedback control
- Trial-by-trial variability
- Signal-dependent noise
- Dynamic programming
- Bellman’s optimality equation
- Linear-quadratic-Gaussian (LQG) control
- Optimal feedback control (OFC) model
- Human psychophysics

Straight paths and bell-shaped velocity in reaching.

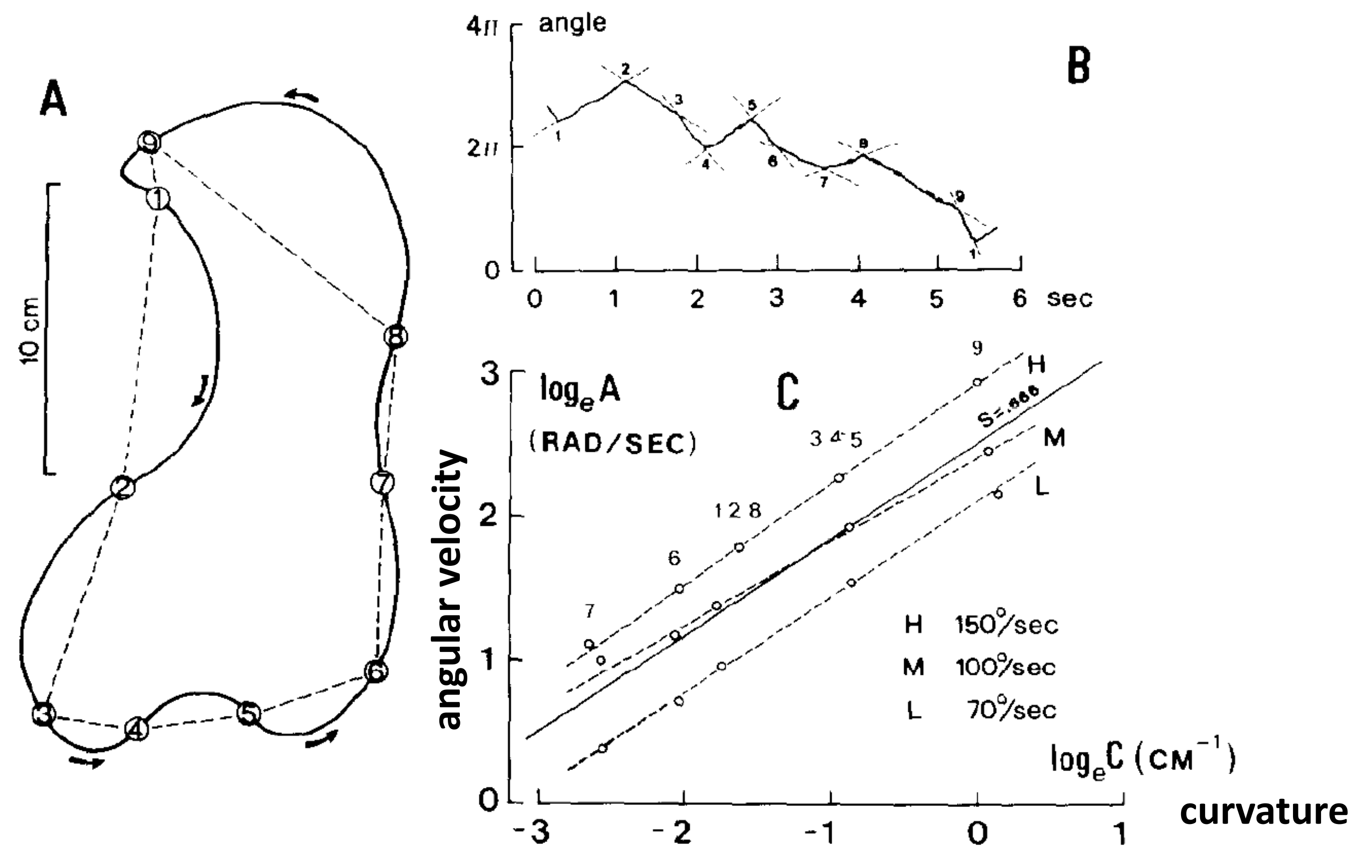






Morasso (1981) Exp Brain Res

Power laws for curved movements.



$\dot{\theta}$: angular velocity

v : hand speed

κ : curvature

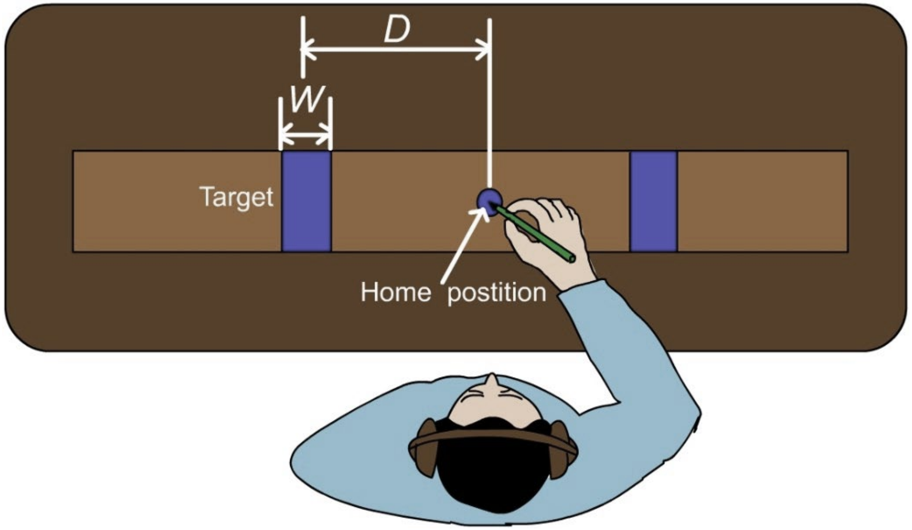
$$\boxed{\dot{\theta} \propto \kappa^{\frac{2}{3}}}$$

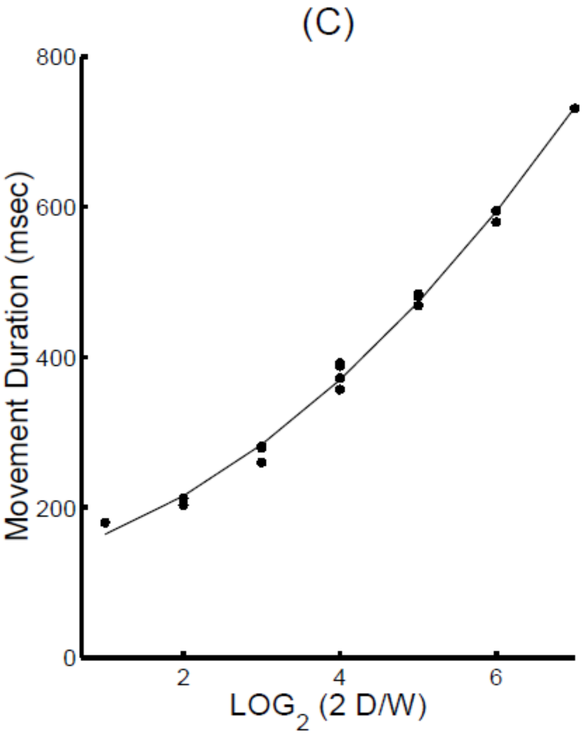
or

$$\boxed{v \propto \kappa^{-\frac{1}{3}}}$$

Lacquaniti et al. (1983) Acta Psychologica

Fitts' law for movement duration in rapid pointing movements.

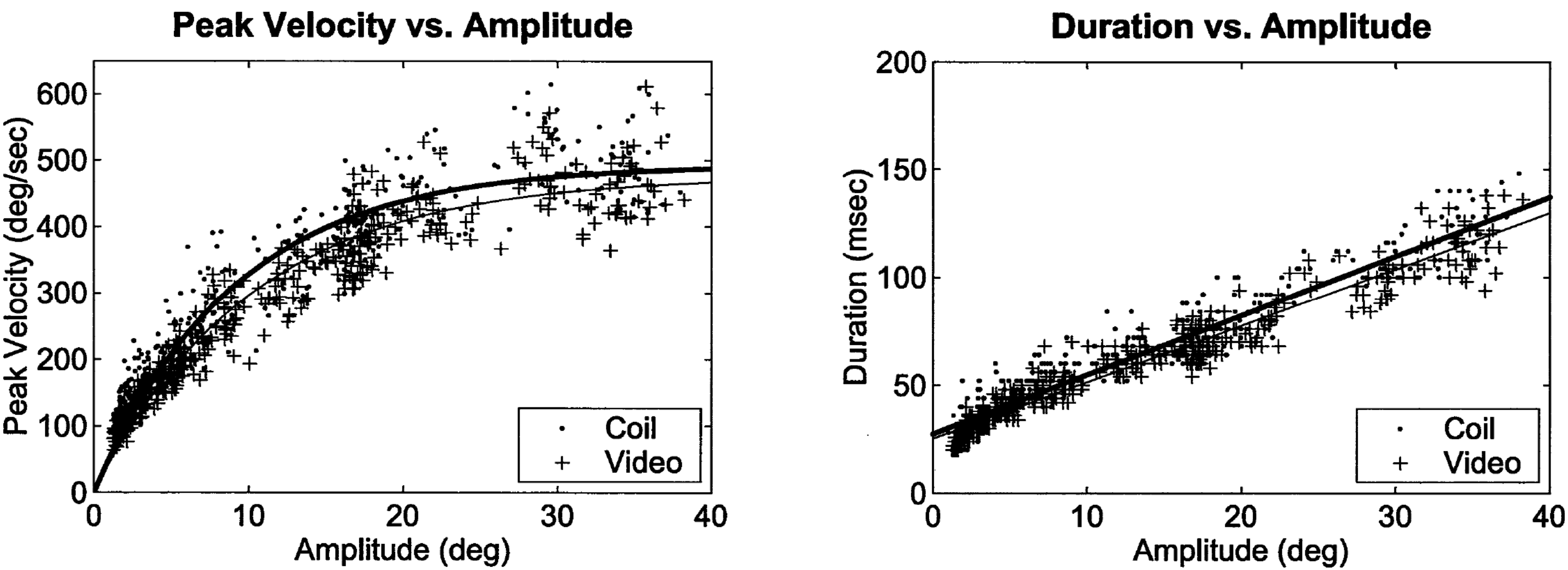




$$t_f = a + b\log_2 \frac{2D}{W}$$

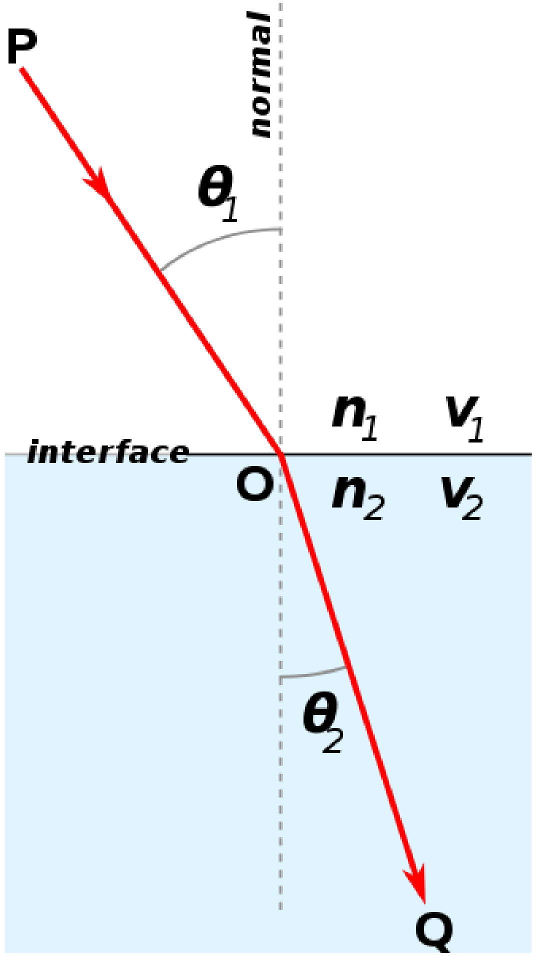
Fitts (1954) J Exp Psychol

“Main sequence” for saccadic movements



Bahill et al. (1975) Math Biosci

Optimality principle: how a unique trajectory is chosen.



Snell's law

$$\frac{\sin\theta_1}{\sin\theta_2}=\frac{v_1}{v_2}=\frac{1/n_1}{1/n_2}$$

Fermat's principle of least time

$$T\left[s(\cdot)\right]=\int_0^T dt=\int_{\mathrm{P}}^{\mathrm{Q}}\frac{ds}{v(s)}=\frac{1}{c}\int_{\mathrm{P}}^{\mathrm{Q}}n(s)\,ds$$

$$\delta T\left[s(\cdot)\right]=\frac{1}{c}\delta\int_{\mathrm{P}}^{\mathrm{Q}}n(s)\,ds=0$$

http://en.wikipedia.org/wiki/Snell%27s_law

Feedforward and feedback control.

$$\mathbf{x}_{k+1} = \mathbf{A}\mathbf{x}_k + \mathbf{B}\mathbf{u}_k$$

Feedforward control as a function of time step:

$$\mathbf{u}_k = \mathbf{u}_k\left(k\right)$$

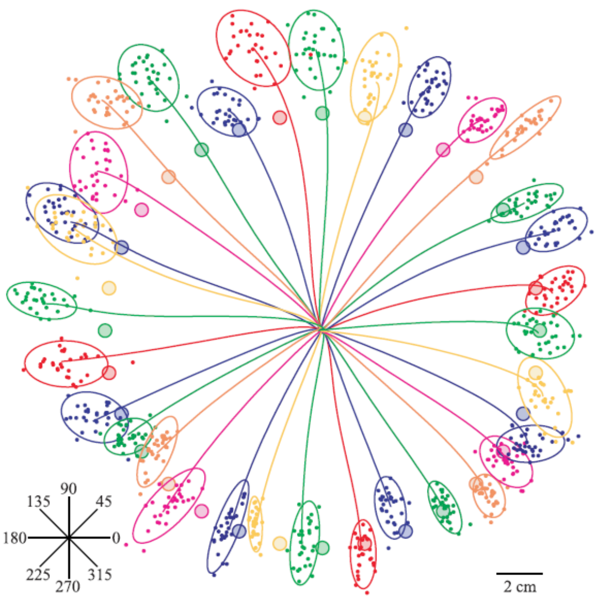
Feedback control as a function of state:

$$\mathbf{u}_k = \mathbf{u}_k\left(\mathbf{x}_k\right)$$

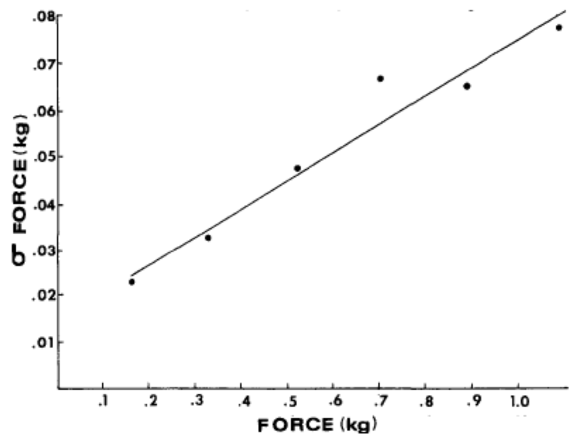
For a deterministic system, feedforward and feedback control are equivalent. On the other hand, for a stochastic system, feedforward and feedback control are NOT equivalent.

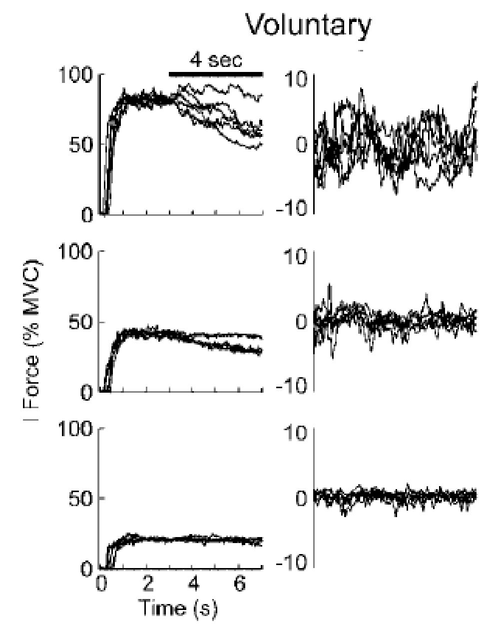
Signal-dependent noise: trial-by-trial variability.

Variability in reaching endpoints



Variability in force production





Signal dependent noise

$$\mathbf{u}^{\text{noise}} = \mathbf{u}(1+\xi), \quad \xi \sim \mathcal{N}(0,1)$$

$$\mathrm{E}\left[\mathbf{u}^{\text{noise}}\right] = \mathbf{u}, \ \ \mathrm{Cov}\left[\mathbf{u}^{\text{noise}}\right] = \mathbf{u}\mathbf{u}^{\mathrm{T}}$$

van Beers et al. (2004) J Neurophysiol;
Schmidt et al. (1979) Psychol Rev; Jones et al. (2002) J Neurophysiol

Minimum-variance control predicts smooth trajectory.

$$\boxed{\mathbf{x}_{n+1} = \mathbf{A}\mathbf{x}_n + \mathbf{B}\mathbf{u}_n\left(1+\xi_n\right)}$$

$$\begin{aligned}
\mathbf{x}_n &= \mathbf{A}\mathbf{x}_{n-1} + \mathbf{B}\mathbf{u}_{n-1}\left(1+\xi_{n-1}\right) \\
&= \mathbf{A}^2\mathbf{x}_{n-2} + \mathbf{A}\mathbf{B}\mathbf{u}_{n-2}\left(1+\xi_{n-2}\right) + \mathbf{B}\mathbf{u}_{n-1}\left(1+\xi_{n-1}\right) \\
&= \cdots \\
&= \mathbf{A}^n\mathbf{x}_0 + \sum_{k=0}^{n-1}\mathbf{A}^{n-k-1}\mathbf{B}\mathbf{u}_k\left(1+\xi_k\right)
\end{aligned}$$

$$\mathrm{E}\left[\mathbf{x}_n\right] = \mathbf{A}^n\mathbf{x}_0 + \sum_{k=0}^{n-1}\mathbf{A}^{n-k-1}\mathbf{B}\mathbf{u}_k$$

$$\mathrm{Cov}\left[\mathbf{x}_n\right] = \sum_{k=0}^{n-1}\mathbf{A}^{n-k-1}\mathbf{B}\mathbf{u}_k\mathbf{u}_k^{\mathrm{T}}\mathbf{B}^{\mathrm{T}}\left(\mathbf{A}^{n-k-1}\right)^{\mathrm{T}}$$

Harris & Wolpert (1998) Nature

Minimum-variance control predicts smooth trajectory.

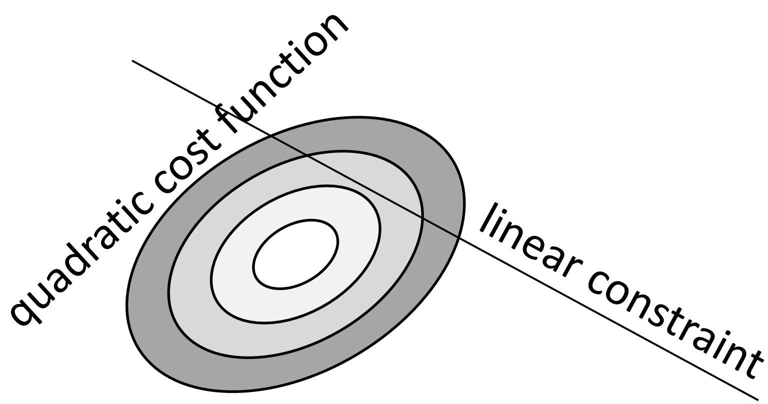
Minimize the variance of final position (quadratic with respect to $\mathbf{u}$)

$$\sum_{t=t_f}^{t_f+T} \mathrm{Cov}\left[\mathbf{x}_n\right]_{11} = \sum_{n=t_f}^{t_f+T} \left[\sum_{k=0}^{n-1} \mathbf{A}^{n-k-1} \mathbf{B} \mathbf{u}_k \mathbf{u}_k^{\mathrm{T}} \mathbf{B}^{\mathrm{T}} \left(\mathbf{A}^{n-k-1}\right)^{\mathrm{T}} \right]_{11}$$

under constraints ($T+1$ constraints with respect to $\mathbf{u}$)

$$\mathrm{E}\left[\mathbf{x}_n\right] = \mathbf{A}^n \mathbf{x}_0 + \sum_{k=0}^{n-1} \mathbf{A}^{n-k-1} \mathbf{B} \mathbf{u}_k = \mathbf{x}_f \quad \left(t_f \le t \le +T\right)$$

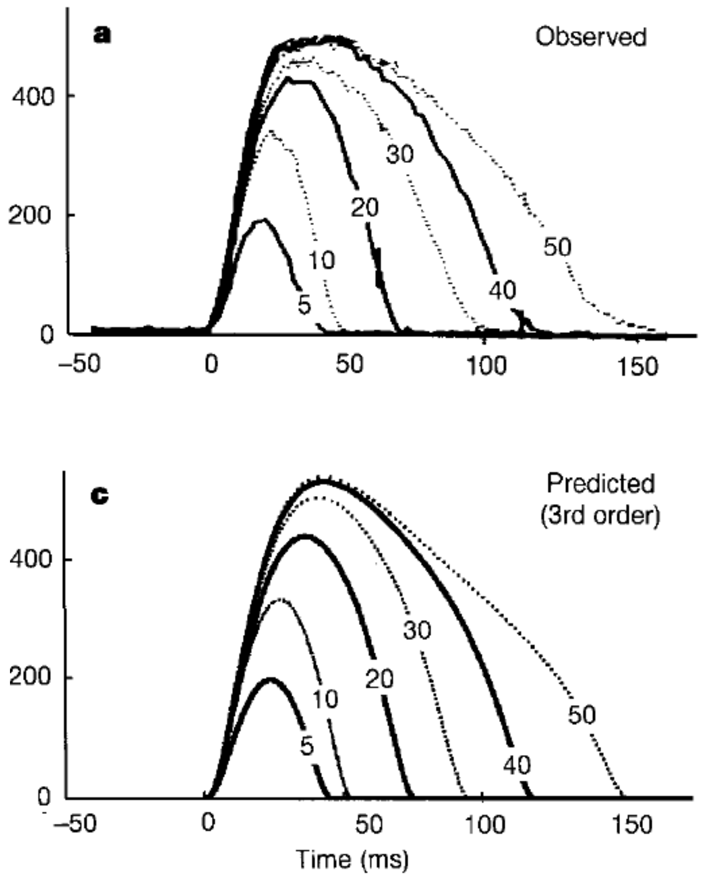
This problem can be solved with quadratic programming (quadprog command).



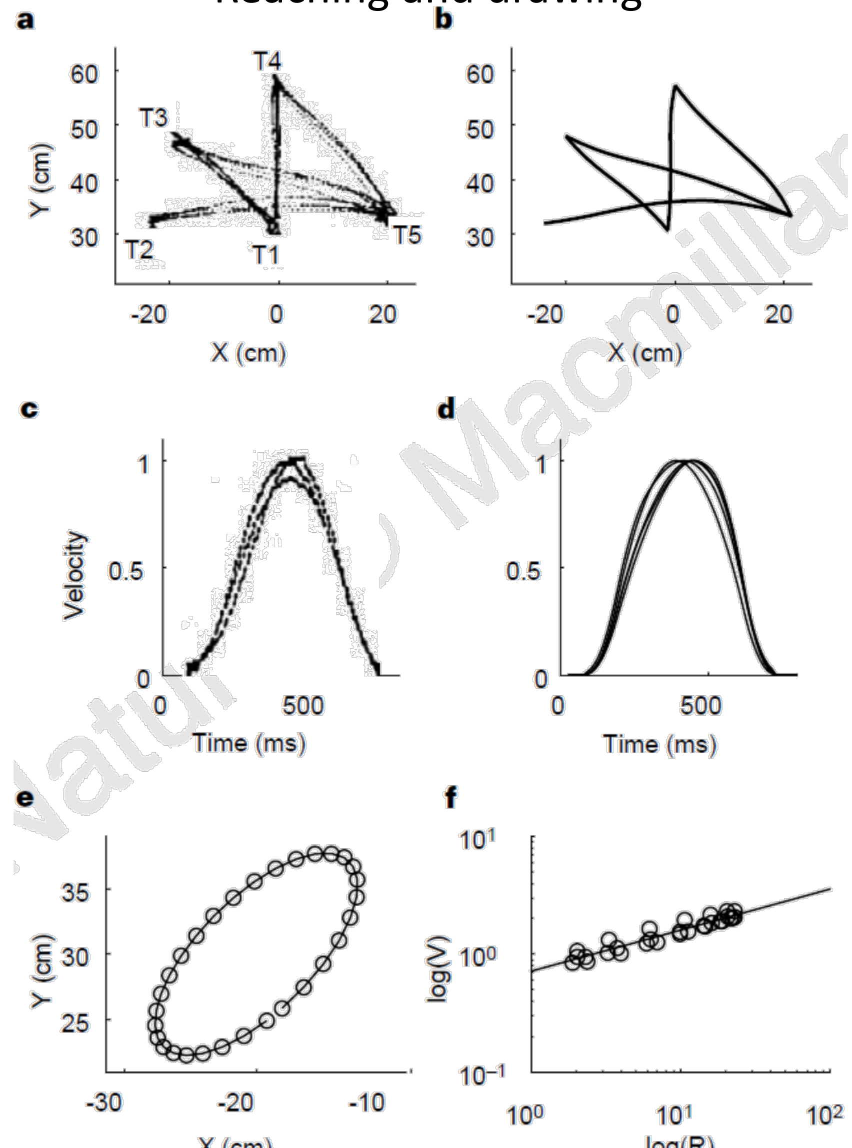
Harris & Wolpert (1998) Nature

Minimum-variance control predicts smooth trajectory.

Saccade velocity



Reaching and drawing



Harris & Wolpert (1998) Nature

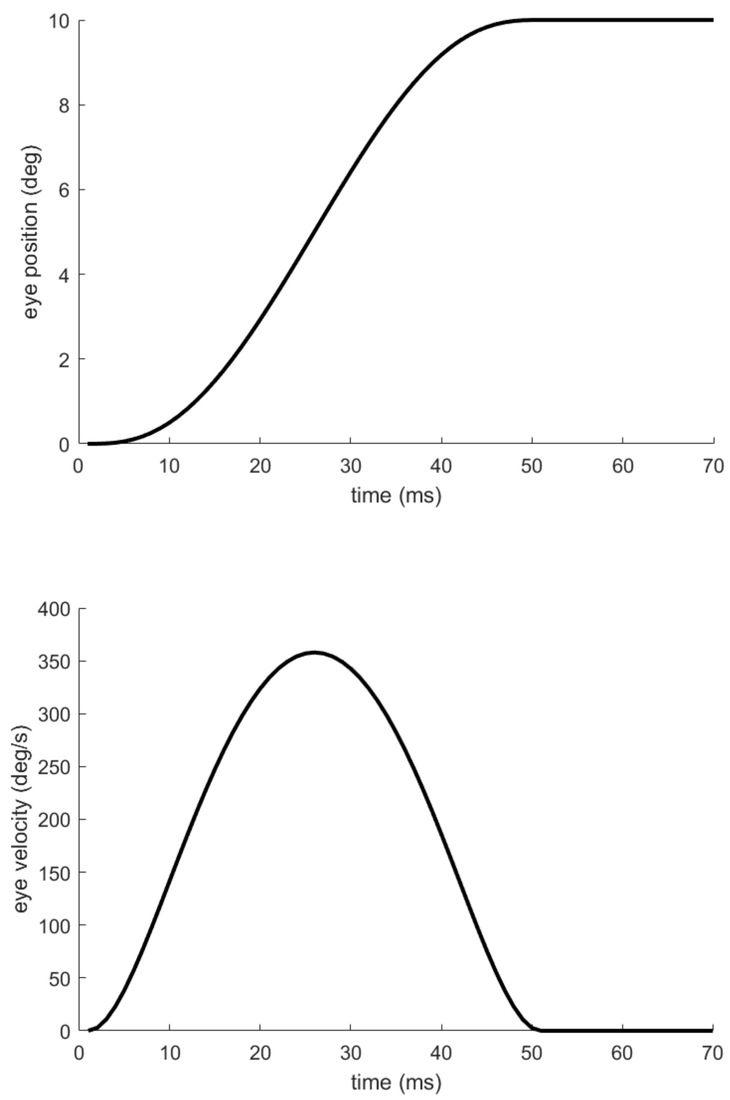
Matlab code for the minimum-variance model.

```
t1 = 224/1000;  % time const of eye dynamics
t2 = 13/1000;   % another time const of eye dynamics
tm = 10/1000;

dt = 1/1000;    % simulation time step

tf = 50/1000;   % movement duration
tp = 20/1000;   % post-movement duration
K = tf/dt;
L = tp/dt;

x0 = [0; 0; 0]; % initial state
xf = [10; 0; 0];    % final state

Ac = [0 1 0; ...
    0 0 1; ...
    -1/(t1*t2*tm) -1/(t1*t2)-1/(t1*tm)-1/(t2*tm) -1/t1-1/t2-
1/tm];
Bc = [0; 0; 1];
A = expm(Ac*dt);
B = inv(Ac)*(eye(3)-expm(Ac*dt))*Bc;

Q = zeros(K+L);

% calculation of Q
for ell=0:K+L-1
    if ell<K
       for k=K:K+L-1
           tmpQ = A^(k-ell-1)*B*B'*A'^(k-ell-1);
           Q(ell+1, ell+1) = Q(ell+1, ell+1) + tmpQ(1,1);
       end
    else
        for k=ell+1:K+L-1
           tmpQ = A^(k-ell-1)*B*B'*A'^(k-ell-1);
           Q(ell+1, ell+1) = Q(ell, ell) + tmpQ(1,1);
        end
    end
End
```

```
% calculation of C
C = [];
for p=1:L+1
    Ctmp = [];
    for q=1:K+L
        if K-1-(q-1)+(p-1)>0
            Ctmp = [Ctmp A^(K-1-(q-1)+(p-1))*B];
        elseif K-1-(q-1)+(p-1)==0
            Ctmp = [Ctmp B];
        else
            Ctmp = [Ctmp zeros(3,1)];
        end
    end
    C = [C; Ctmp];
end

% calculation of d
d = [];
for ell=0:L
    d = [d; xf-A^(K+ell)*x0];
end

% solution by quadratic programming
u = quadprog(Q, [], [], [], C, d);

% forward solution
x = zeros(3, K+L);
x(:,1) = x0;
for k=1:K+L-1
    x(:,k+1) = A*x(:,k) + B*u(k);
end

figure(1);
subplot(211); hold on; plot(x(1,:), 'k', 'linewidth', 2);
set(gca, 'plotboxaspectratio', [1.5 1 1]);
xlabel('time (ms)'); ylabel('eye position (deg)');
subplot(212); hold on; plot(x(2,:), 'k', 'linewidth', 2)
set(gca, 'plotboxaspectratio', [1.5 1 1]);
xlabel('time (ms)'); ylabel('eye velocity (deg/s)');
```

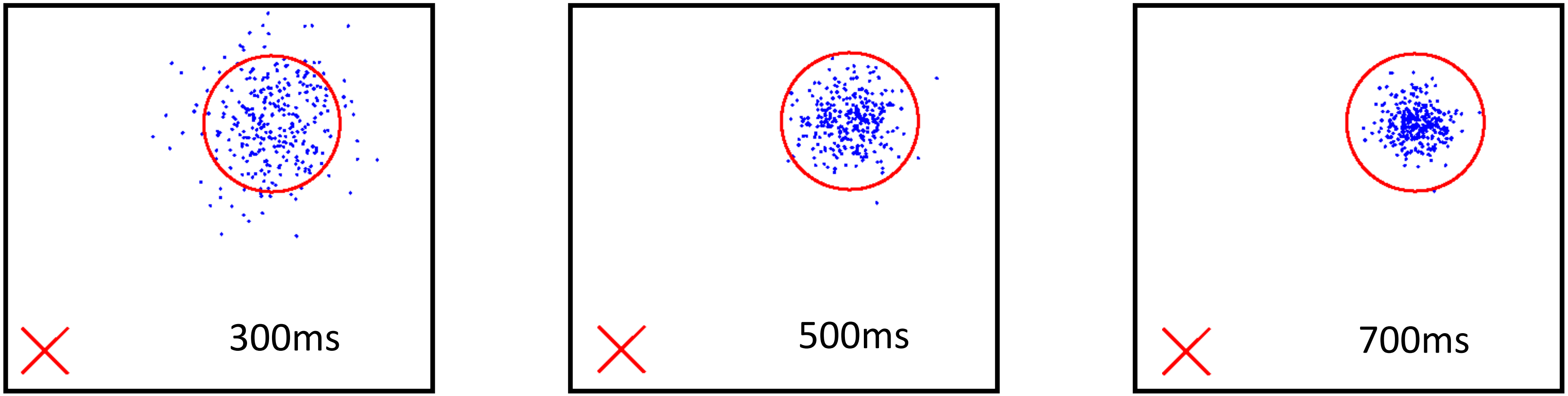
Harris & Wolpert (1998) Nature

Matlab code for the minimum-variance model.



Harris & Wolpert (1998) Nature

Minimum-time control predicts Fitts' law and main sequence.



Faster, but less accurate

Unique movement time can be determined

More precise, but slower

$$S_{\mathrm{MT}}\left[t_f, \{\mathbf{u}\}\right] = t_f + \lambda\left[V_f - \frac{1}{t_p}\int_{t_f}^{t_f+t_p} dt\ \mathrm{Var}\left[\mathbf{x}(\mathrm{t})\right]\right] + \int_{t_f}^{t_f+t_p} dt\ \boldsymbol{\mu}^{\mathrm{T}}(t)\left[\mathbf{x}_f - \mathrm{E}\left[\mathbf{x}(t)\right]\right]$$

minimization of movement time

final variance constraint

final position constraint

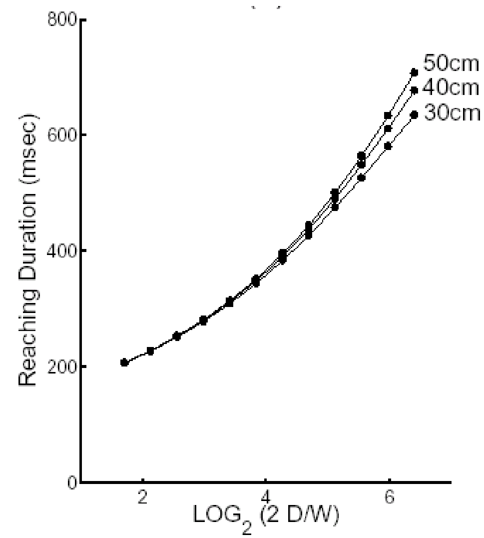
Tanaka et al. (2006) J Neurophysiol

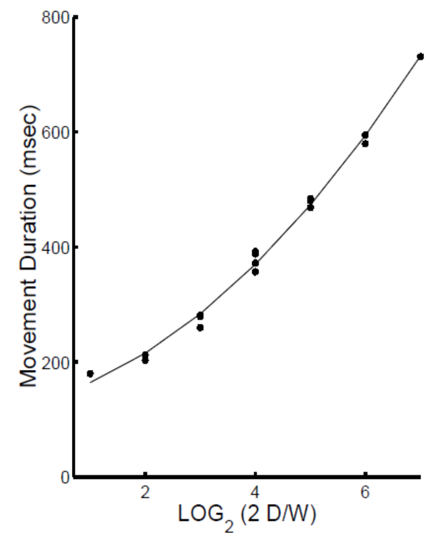
Minimum-time control predicts Fitts' law and main sequence.

Model

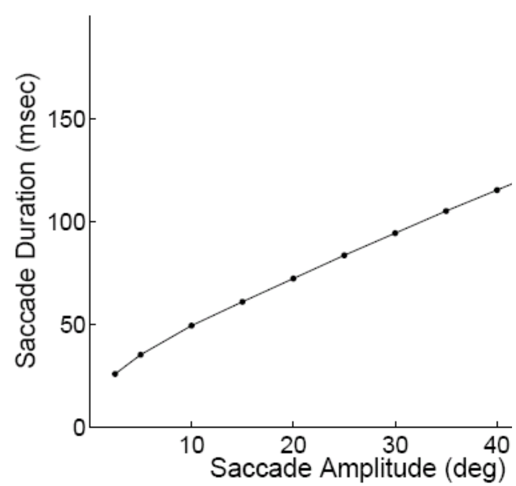
Experiment

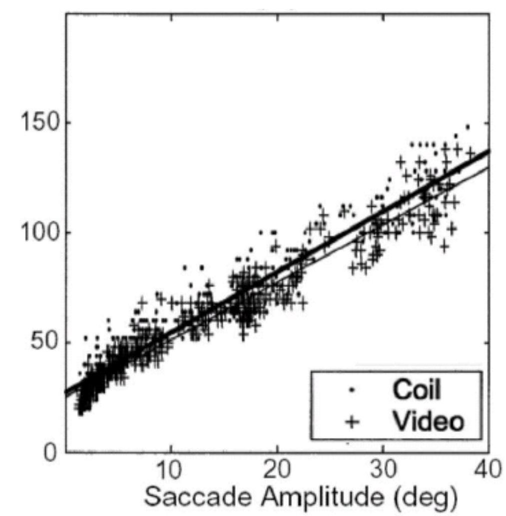
Fitts' law





Main sequence





Tanaka et al. (2006) J Neurophysiol

Optimal control: Bellman's optimality equation.

Deterministic (i.e., noiseless) dynamics:

$$\mathbf{x}_{k+1} = \mathbf{A}\mathbf{x}_k + \mathbf{B}\mathbf{u}_k$$

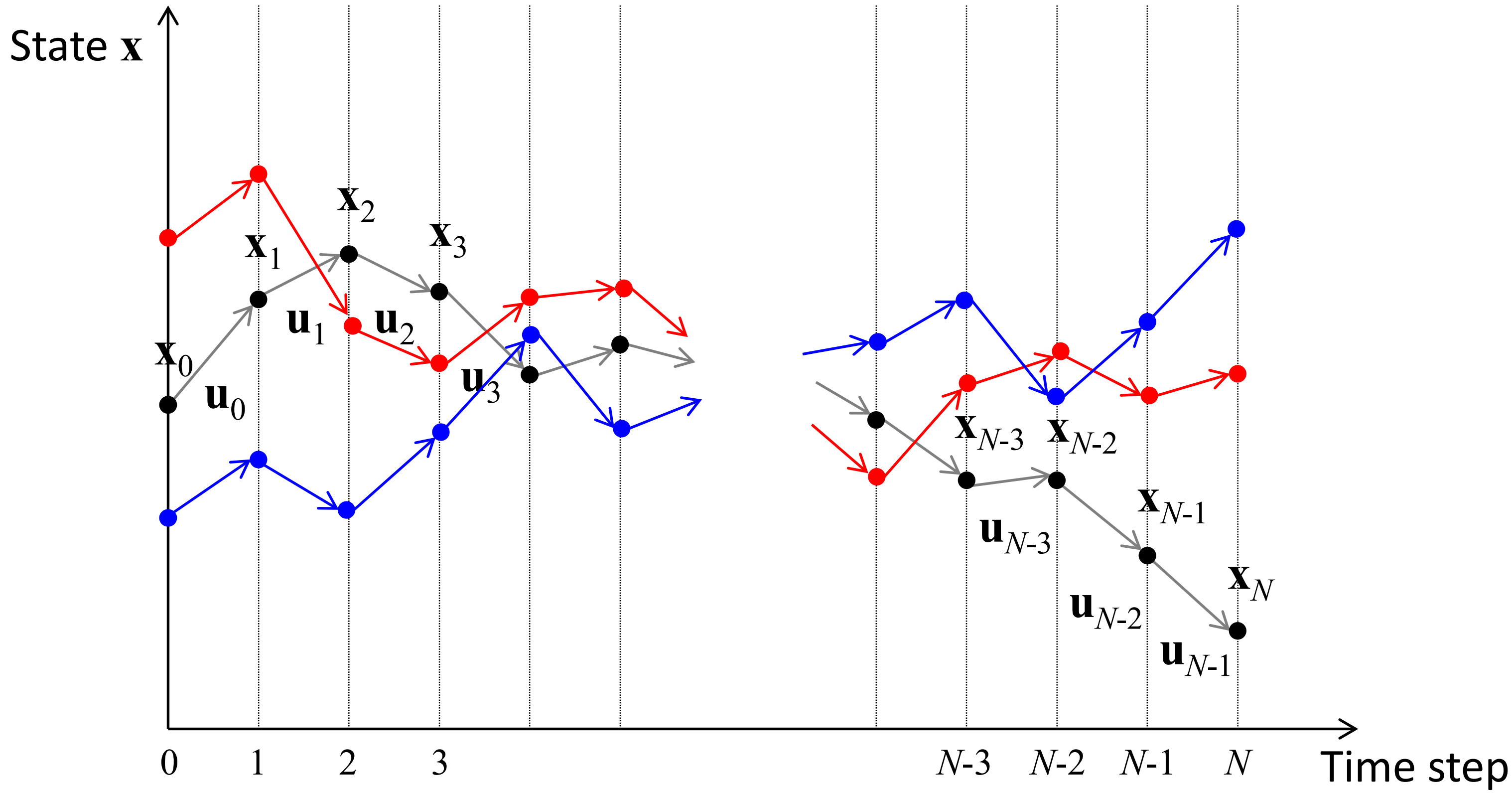
Cost function:

$$J\left(\mathbf{u}_0, \mathbf{u}_1, \cdots, \mathbf{u}_{N-1}; \mathbf{x}_0\right) = \frac{1}{2}\sum_{k=0}^{N-1}\left(\underbrace{\mathbf{x}_k^T \mathbf{Q}_k \mathbf{x}_k}_{\text{Error during movement}} + \underbrace{\mathbf{u}_k^T \mathbf{R} \mathbf{u}_k}_{\text{Control cost}}\right) + \underbrace{\frac{1}{2}\mathbf{x}_N^T \mathbf{Q}_N \mathbf{x}_N}_{\text{Endpoint error}}$$

Find control signals $\{\mathbf{u}_0, \mathbf{u}_1, \ldots, \mathbf{u}_{N\text{-}1}\}$ that minimize the cost function.

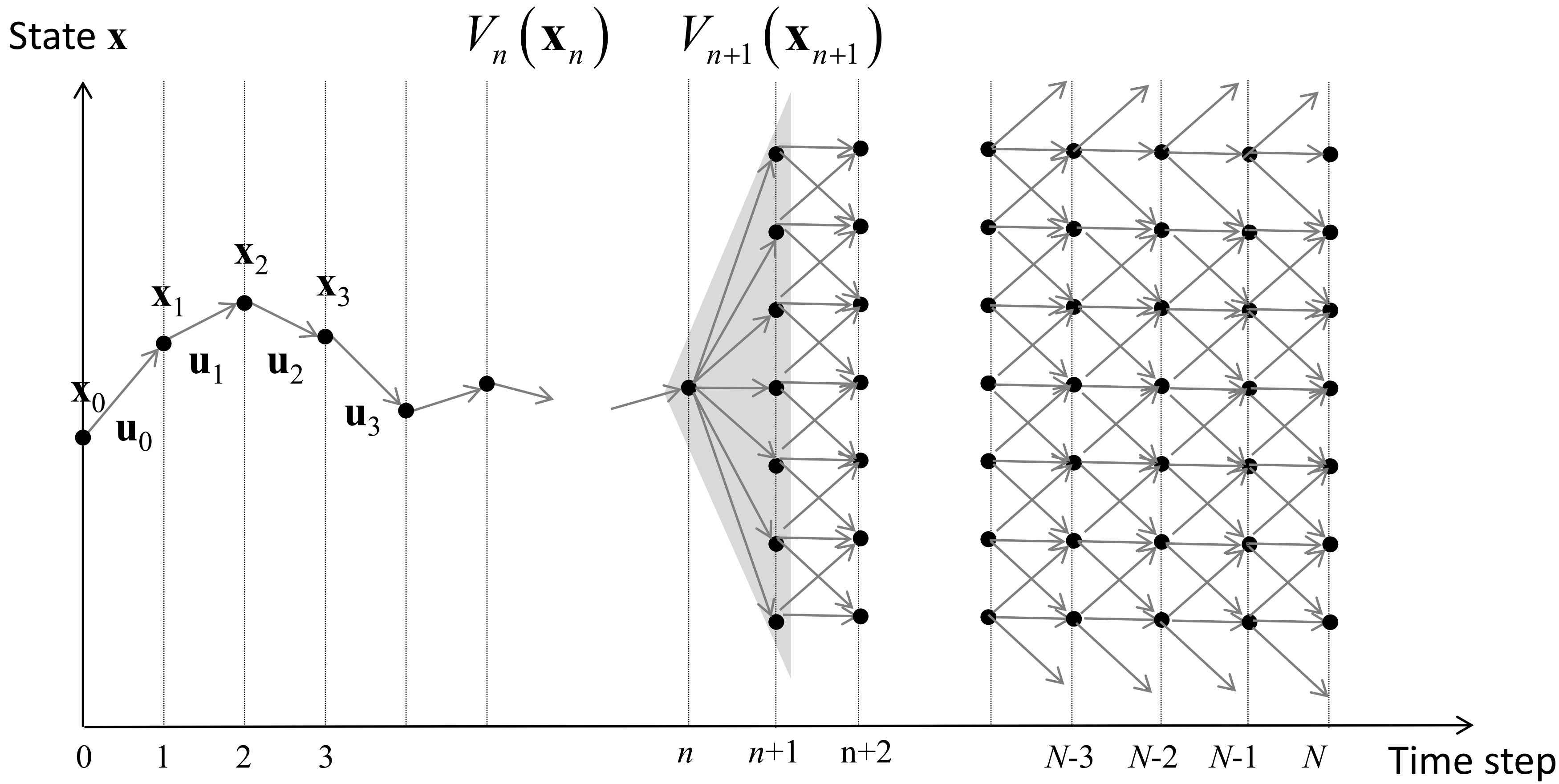
Optimal control: Pontryagin's minimum principle.

$$J\left(\mathbf{u}_0,\mathbf{u}_1,\cdots,\mathbf{u}_{N-1};\mathbf{x}_0\right)=\frac{1}{2}\sum_{k=0}^{N-1}\left(\mathbf{x}_k^T\mathbf{Q}_k\mathbf{x}_k+\mathbf{u}_k^T\mathbf{R}\mathbf{u}_k\right)+\frac{1}{2}\mathbf{x}_N^T\mathbf{Q}_N\mathbf{x}_N$$



The control signals $\{\mathbf{u}_0, \mathbf{u}_1, \ldots, \mathbf{u}_{N-1}\}$ are optimized as a whole.

Optimal control: Bellman's optimality equation.



(cost at state $\mathbf{x}_n$ at step n)
= (cost of moving $\mathbf{x}_n$ to $\mathbf{x}_{n+1}$) + (cost at state $\mathbf{x}_{n+1}$ at step $n+1$)

Optimal control: Bellman's optimality equation.

$$\boxed{V_n(\mathbf{x}_n) = \min_{\{\mathbf{u}_n, \mathbf{u}_{n+1}, \cdots, \mathbf{u}_{N-1}\}} \left[\frac{1}{2} \sum_{k=n}^{N-1} \left(\mathbf{x}_k^T \mathbf{Q}_k \mathbf{x}_k + \mathbf{u}_k^T \mathbf{R} \mathbf{u}_k \right) + \frac{1}{2} \mathbf{x}_N^T \mathbf{Q}_N \mathbf{x}_N \right]}$$

$$\begin{aligned}
V_n(\mathbf{x}_n) &= \min_{\{\mathbf{u}_n, \mathbf{u}_{n+1}, \cdots, \mathbf{u}_{N-1}\}} \left[\frac{1}{2} \sum_{k=n}^{N-1} \left(\mathbf{x}_k^T \mathbf{Q}_k \mathbf{x}_k + \mathbf{u}_k^T \mathbf{R} \mathbf{u}_k \right) + \frac{1}{2} \mathbf{x}_N^T \mathbf{Q}_N \mathbf{x}_N \right] \\
&= \min_{\mathbf{u}_n} \left[\frac{1}{2} \left(\mathbf{x}_n^T \mathbf{Q}_n \mathbf{x}_n + \mathbf{u}_n^T \mathbf{R} \mathbf{u}_n \right) + \underbrace{\min_{\{\mathbf{u}_{n+1}, \cdots, \mathbf{u}_{N-1}\}} \left[\frac{1}{2} \sum_{k=n+1}^{N-1} \left(\mathbf{x}_k^T \mathbf{Q}_k \mathbf{x}_k + \mathbf{u}_k^T \mathbf{R} \mathbf{u}_k \right) + \frac{1}{2} \mathbf{x}_N^T \mathbf{Q}_N \mathbf{x}_N \right]}_{V_{n+1}(\mathbf{x}_{n+1})} \right] \\
&= \min_{\mathbf{u}_n} \left[\frac{1}{2} \left(\mathbf{x}_n^T \mathbf{Q}_n \mathbf{x}_n + \mathbf{u}_n^T \mathbf{R} \mathbf{u}_n \right) + V_{n+1}(\mathbf{x}_{n+1}) \right]
\end{aligned}$$

$$\boxed{V_n(\mathbf{x}_n) = \min_{\mathbf{u}_n} \left[\frac{1}{2} \left(\mathbf{x}_n^T \mathbf{Q}_n \mathbf{x}_n + \mathbf{u}_n^T \mathbf{R} \mathbf{u}_n \right) + V_{n+1}(\mathbf{x}_{n+1}) \right]}$$

Solvable example: Linear-Quadratic-Regulator control.

Assume that the cost-to-go function has a quadratic form:

$$V_{k+1}(\mathbf{x}) = \frac{1}{2}\mathbf{x}_{k+1}^T \mathbf{S}_{k+1} \mathbf{x}_{k+1}$$

Substituting this into the Bellman equation gives

$$\begin{aligned} V_k(\mathbf{x}) &= \min_u \left[\frac{1}{2}\left(\mathbf{x}_k^T \mathbf{Q}_k \mathbf{x}_k + \mathbf{u}_k^T \mathbf{R} \mathbf{u}_k \right) + \frac{1}{2}\mathbf{x}_{k+1}^T \mathbf{S}_{k+1} \mathbf{x}_{k+1} \right] \\ &= \min_u \left[\frac{1}{2}\left(\mathbf{x}_k^T \mathbf{Q}_k \mathbf{x}_k + \mathbf{u}_k^T \mathbf{R} \mathbf{u}_k \right) + \frac{1}{2}\left(\mathbf{A}\mathbf{x}_k + \mathbf{B}\mathbf{u}_k \right)^T \mathbf{S}_{k+1} \left(\mathbf{A}\mathbf{x}_k + \mathbf{B}\mathbf{u}_k \right) \right] \end{aligned}$$

Then, by minimizing with respect to $\mathbf{u}$,

$$\begin{aligned} 0 &= \frac{\partial}{\partial \mathbf{u}} \left[\frac{1}{2}\left(\mathbf{u}_k^T \mathbf{R} \mathbf{u}_k + \mathbf{u}_k^T \mathbf{B}^T \mathbf{A} \mathbf{B} \mathbf{u}_k + \mathbf{x}_k^T \mathbf{A}^T \mathbf{S}_{k+1} \mathbf{B} \mathbf{u}_k + \mathbf{u}_k^T \mathbf{B}^T \mathbf{S}_{k+1} \mathbf{A} \mathbf{x}_k \right) \right] \\ &= \left(\mathbf{R} + \mathbf{B}^T \mathbf{S}_{k+1} \mathbf{B} \right) \mathbf{u}_k + \mathbf{B}^T \mathbf{S}_{k+1} \mathbf{A} \mathbf{x}_k \end{aligned}$$

A feedback control law is obtained:

$$\boxed{\mathbf{u}_k = -\left(\mathbf{R} + \mathbf{B}^T \mathbf{S}_{k+1} \mathbf{B} \right)^{-1} \mathbf{B}^T \mathbf{S}_{k+1} \mathbf{A} \mathbf{x}_k = -\mathbf{L}_k \mathbf{x}_k}$$

Solvable example: Linear-Quadratic-Regulator control.

Substituting the control law

$$\mathbf{u}_k = -\left(\mathbf{R} + \mathbf{B}^T \mathbf{S}_{k+1} \mathbf{B}\right)^{-1} \mathbf{B}^T \mathbf{S}_{k+1} \mathbf{A} \mathbf{x}_k = -\mathbf{L}_k \mathbf{x}_k$$

into the Bellman equation gives

$$V_k\left(\mathbf{x}_k\right) = \frac{1}{2} \mathbf{x}_k^T \mathbf{S}_k \mathbf{x}_k = \frac{1}{2} \mathbf{x}_k^T \left[\mathbf{Q}_k + \mathbf{A}\left(\mathbf{S}_{k+1}^{-1} + \mathbf{B}\mathbf{R}^{-1}\mathbf{B}^T\right)^{-1} \mathbf{A}\right] \mathbf{x}_k$$

Therefore a backward recursive equation for the matrix S is obtained.

$$\mathbf{S}_k = \mathbf{Q}_k + \mathbf{A}^T \left(\mathbf{S}_{k+1}^{-1} + \mathbf{B}\mathbf{R}^{-1}\mathbf{B}^T\right)^{-1} \mathbf{A}$$

$$\begin{tikzcd}
0 & 1 & 2 & & k\text{-}1 & k & k+1 & & N\text{-}1 & N \\
\mathbf{S}_0 & \mathbf{S}_1 \arrow[l] \arrow[d] & \mathbf{S}_2 \arrow[l] \arrow[d] & \arrow[l] & \mathbf{S}_{k\text{-}1} \arrow[l] \arrow[d] & \mathbf{S}_k \arrow[l] \arrow[d] & \mathbf{S}_{k+1} \arrow[l] \arrow[d] & \arrow[l] & \mathbf{S}_{N\text{-}1} \arrow[l] \arrow[d] & \mathbf{S}_N \arrow[l] \arrow[d] \\
 & \mathbf{L}_0 & \mathbf{L}_1 & & \mathbf{L}_{k\text{-}2} & \mathbf{L}_{k\text{-}1} & \mathbf{L}_k & & \mathbf{L}_{N\text{-}2} & \mathbf{L}_{N\text{-}1}
\end{tikzcd}$$

Solvable example: Linear-Quadratic-Regulator control.

Backward recursion equation for the matrix $\mathbf{S}$:

$$\mathbf{S}_k = \mathbf{Q}_k + \mathbf{A}^T \left(\mathbf{S}_{k+1}^{-1} + \mathbf{B}\mathbf{R}^{-1}\mathbf{B}^T \right)^{-1} \mathbf{A}$$

with the terminal condition $\mathbf{S}_N = \mathbf{Q}_N$

Feedback control law:

$$\mathbf{u}_k = -\mathbf{L}_k \mathbf{x}_k$$

where $\mathbf{L}_k = \left(\mathbf{R} + \mathbf{B}^T \mathbf{S}_{k+1} \mathbf{B} \right)^{-1} \mathbf{B}^T \mathbf{S}_{k+1} \mathbf{A}$

Therefore, once the matrices $\{\mathbf{S}_0, \ldots, \mathbf{S}_N\}$ and $\{\mathbf{L}_0, \ldots, \mathbf{L}_{N\text{-}1}\}$ are obtained, the optimal feedback control signals $\{\mathbf{u}_0, \ldots, \mathbf{u}_{N\text{-}1}\}$ can be computed.

Deterministic and stochastic optimal control.

Deterministic Optimal Control

$$J\left(\mathbf{u}_0,\mathbf{u}_1,\cdots,\mathbf{u}_{N-1};\mathbf{x}_0\right)=\frac{1}{2}\sum_{k=0}^{N-1}\left(\mathbf{x}_k^T\mathbf{Q}_k\mathbf{x}_k+\mathbf{u}_k^T\mathbf{R}\mathbf{u}_k\right)+\frac{1}{2}\mathbf{x}_N^T\mathbf{Q}_N\mathbf{x}_N$$

$$\begin{cases}\mathbf{x}_{k+1}=\mathbf{A}\mathbf{x}_k+\mathbf{B}\mathbf{u}_k\\ \mathbf{y}_k=\mathbf{C}\mathbf{x}_k\end{cases}$$

Stochastic Optimal Control

$$J\left(\mathbf{u}_0,\mathbf{u}_1,\cdots,\mathbf{u}_{N-1};\hat{\mathbf{x}}_0,\Sigma_0\right)=\mathrm{E}\left[\frac{1}{2}\sum_{k=0}^{N-1}\left(\mathbf{x}_k^T\mathbf{Q}_k\mathbf{x}_k+\mathbf{u}_k^T\mathbf{R}\mathbf{u}_k\right)+\frac{1}{2}\mathbf{x}_N^T\mathbf{Q}_N\mathbf{x}_N\right]_{\mathbf{w},\mathbf{v}}$$

$$\begin{cases}\mathbf{x}_{k+1}=\mathbf{A}\mathbf{x}_k+\mathbf{B}\mathbf{u}_k+\mathbf{w}_k\\ \mathbf{y}_k=\mathbf{C}\mathbf{x}_k+\mathbf{v}_k\end{cases}$$

Stochastic optimal control combines Kalman filter and feedback control.

Stochastic Optimal Control

$$J\left(\mathbf{u}_0,\mathbf{u}_1,\cdots,\mathbf{u}_{N-1};\hat{\mathbf{x}}_0,\Sigma_0\right)=\mathrm{E}\left[\frac{1}{2}\sum_{k=0}^{N-1}\left(\mathbf{x}_k^T\mathbf{Q}_k\mathbf{x}_k+\mathbf{u}_k^T\mathbf{R}\mathbf{u}_k\right)+\frac{1}{2}\mathbf{x}_N^T\mathbf{Q}_N\mathbf{x}_N\right]_{\mathbf{w},\mathbf{v}}$$

$$\begin{cases}\mathbf{x}_{k+1}=\mathbf{A}\mathbf{x}_k+\mathbf{B}\mathbf{u}_k+\mathbf{w}_k\\ \mathbf{y}_k=\mathbf{C}\mathbf{x}_k+\mathbf{v}_k\end{cases}$$

Optimal state estimation (Kalman Filter)

$$\hat{\mathbf{x}}_{k+1}=\mathbf{A}\hat{\mathbf{x}}_k+\mathbf{B}\mathbf{u}_k+\mathbf{K}_k\left(\mathbf{y}_k-\mathbf{C}\hat{\mathbf{x}}_k\right)$$

Feedback controller

$$\mathbf{u}_k=-\mathbf{L}_k\hat{\mathbf{x}}_k$$

Optimal feedback control as a model of motor control.

$$\mathbf{x}_{k+1} = \mathbf{A}\mathbf{x}_k + \mathbf{B}\mathbf{u}_k + \mathbf{C}\mathbf{u}_k \varepsilon_k$$
$$\mathbf{y}_k = \mathbf{H}\mathbf{x}_k + \boldsymbol{\omega}_k$$

$$J\left(\mathbf{u}_0, \mathbf{u}_1, \cdots, \mathbf{u}_{N-1}; \hat{\mathbf{x}}_0, \Sigma_0\right) = \mathrm{E}\left[\frac{1}{2}\sum_{k=0}^{N-1}\left(\mathbf{x}_k^T \mathbf{Q}_k \mathbf{x}_k + \mathbf{u}_k^T \mathbf{R}\mathbf{u}_k\right) + \frac{1}{2}\mathbf{x}_N^T \mathbf{Q}_N \mathbf{x}_N\right]_{\varepsilon, \boldsymbol{\omega}}$$

Todorov & Jordan (2002) Nature Neurosci

Optimal feedback control as a model of motor control.

Kalman filter:

$$\boxed{\hat{\mathbf{x}}_{k+1} = \mathbf{A}\hat{\mathbf{x}}_k + \mathbf{B}\mathbf{u}_k + \mathbf{K}_k\left(\mathbf{y}_k - \mathbf{H}\hat{\mathbf{x}}_k\right)}$$

$$\mathbf{K}_k = \mathbf{A}\Sigma_k^{\mathbf{e}}\mathbf{H}^{\mathrm{T}}\left(\mathbf{H}\Sigma_k^{\mathbf{e}}\mathbf{H}^{\mathrm{T}} + \mathbf{\Omega}^{\boldsymbol{\omega}}\right)^{-1}$$

$$\Sigma_{k+1}^{\mathbf{e}} = \left(\mathbf{A} - \mathbf{K}_k\mathbf{H}\right)\Sigma_k^{\mathbf{e}}\mathbf{A}^{\mathrm{T}} + \mathbf{C}\mathbf{L}_k\Sigma_k^{\mathbf{x}}\mathbf{L}_k^{\mathrm{T}}\mathbf{C}^{\mathrm{T}}; \quad \Sigma_1^{\mathbf{e}} = \Sigma_1$$

$$\Sigma_{k+1}^{\hat{\mathbf{x}}} = \mathbf{K}_k\mathbf{H}\Sigma_k^{\mathbf{e}}\mathbf{A}^{\mathrm{T}} + \left(\mathbf{A} - \mathbf{B}\mathbf{L}_k\right)\Sigma_k^{\hat{\mathbf{x}}}\left(\mathbf{A} - \mathbf{B}\mathbf{L}_k\right)^T; \quad \Sigma_1^{\hat{\mathbf{x}}} = \hat{\mathbf{x}}\hat{\mathbf{x}}^T$$

Feedback control:

$$\boxed{\mathbf{u}_k = -\mathbf{L}_k\hat{\mathbf{x}}_k}$$

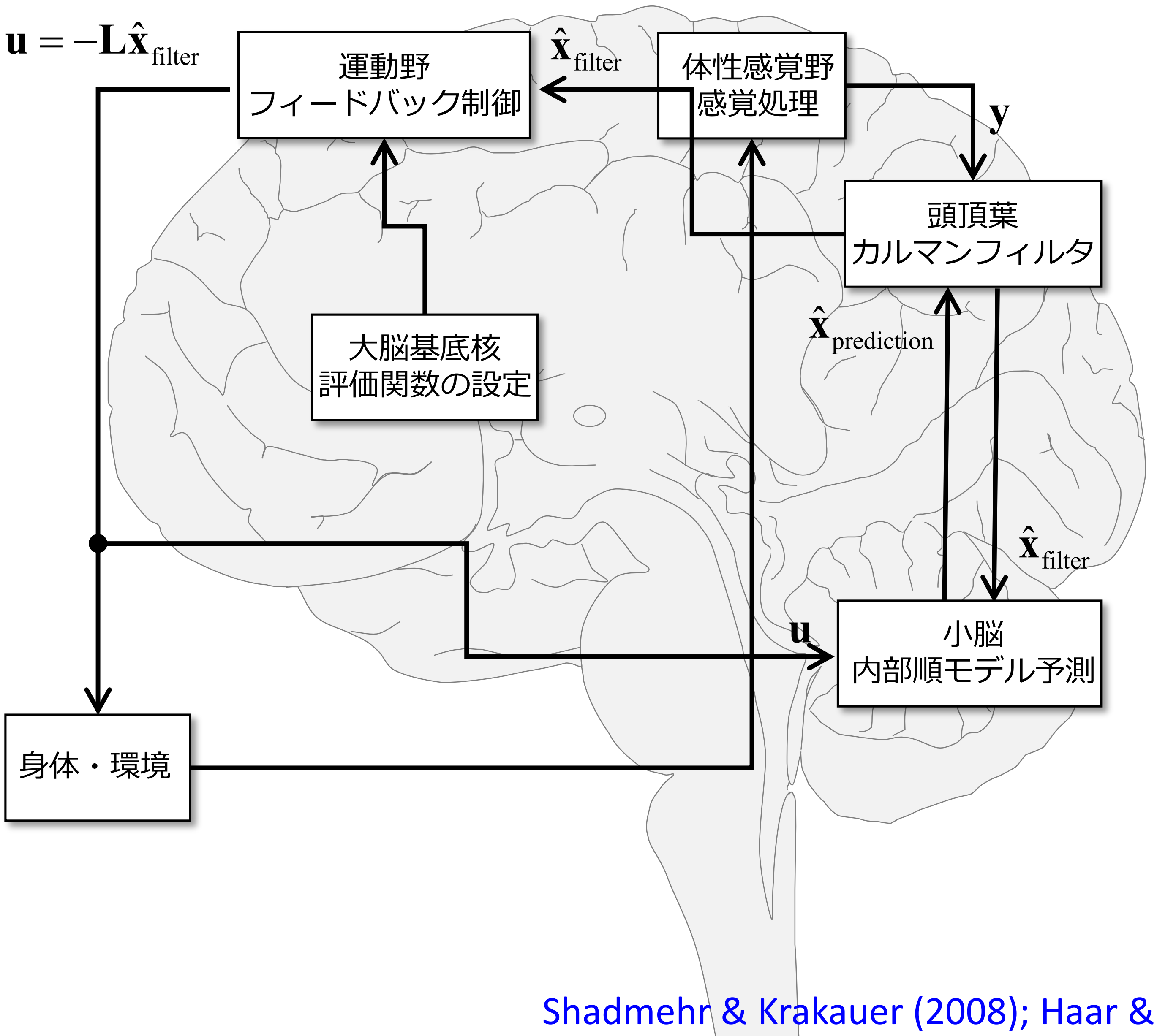
$$\mathbf{L}_k = \left(\mathbf{B}^{\mathrm{T}}\mathbf{S}_{k+1}^{\mathbf{x}}\mathbf{B} + \mathbf{R} + \mathbf{C}^{\mathrm{T}}\left(\mathbf{S}_{k+1}^{\mathbf{x}} + \mathbf{S}_{k+1}^{\mathbf{e}}\right)\mathbf{C}\right)^{-1}\mathbf{B}^{\mathrm{T}}\mathbf{S}_{k+1}^{\mathbf{x}}\mathbf{A}$$

$$\mathbf{S}_k^{\mathbf{x}} = \mathbf{Q}_k + \mathbf{A}^{\mathrm{T}}\mathbf{S}_{k+1}^{\mathbf{x}}\left(\mathbf{A} - \mathbf{B}\mathbf{L}_k\right); \quad \mathbf{S}_N^{\mathbf{x}} = Q_N$$

$$\mathbf{S}_k^{\mathbf{e}} = \mathbf{A}^{\mathrm{T}}\mathbf{S}_{k+1}^{\mathbf{x}}\mathbf{B}\mathbf{L}_k + \left(\mathbf{A} - \mathbf{K}_k\mathbf{H}\right)^{\mathrm{T}}\mathbf{S}_{k+1}^{\mathbf{e}}\left(\mathbf{A} - \mathbf{K}_k\mathbf{H}\right); \quad \mathbf{S}_N^{\mathbf{e}} = \mathbf{0}$$

Todorov & Jordan (2002) Nature Neurosci; Todorov (2005) Neural Comput

Computational neuroanatomy for motor control.



Shadmehr & Krakauer (2008); Haar & Donchin (2019)

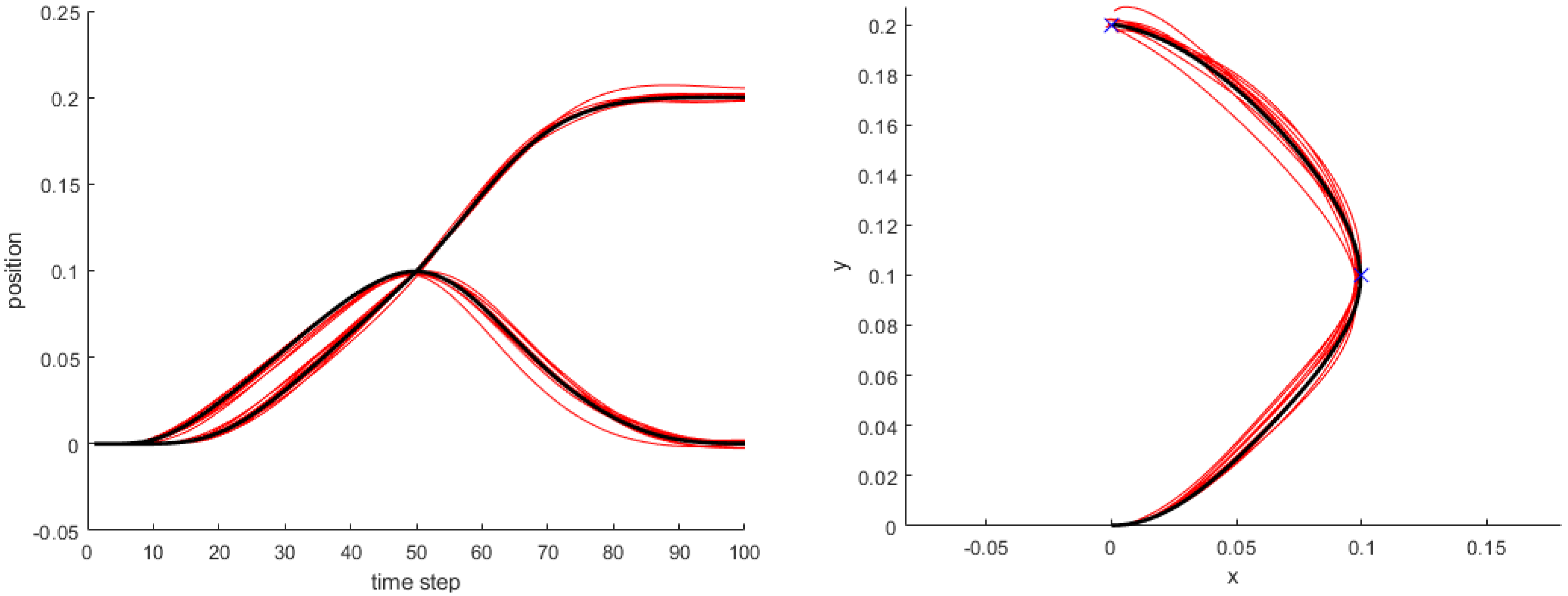
Matlab code: optimal feedback control model.

```
for iter = 1:MaxIter

   % initialize covariances
   SiE = S1;
   SiX = X1*X1';
   SiXE = zeros(szX,szX);

   % forward pass - recompute Kalman filter
   for k = 1:N-1

      % compute Kalman gain
      temp = SiE + SiX + SiXE + SiXE';
      if size(D,2)==1,
          DSiD = diag(diag(temp).*D.^2);
      else
          DSiD = zeros(szY,szY);
          for i=1:szD
              DSiD = DSiD + D(:,:,i)*temp*D(:,:,i)';
          end;
      end;
      K(:,:,k) = A*SiE*H'*pinv(H*SiE*H'+D0*D0'+DSiD);

      % compute new SiE
      newE = E0*E0' + C0*C0' + (A-K(:,:,k)*H)*SiE*A';
      LSiL = L(:,:,k)*SiX*L(:,:,k)';
      if size(C,2)==1,
         newE = newE + B*diag(diag(LSiL).*C.^2)*B';
      else
         for i=1:szC
            newE = newE + B*C(:,:,i)*LSiL*C(:,:,i)'*B';
         end;
      end;

      % update SiX, SiE, SiXE
      SiX = E0*E0' + K(:,:,k)*H*SiE*A' + (A-B*L(:,:,k))*SiX*(A-B*L(:,:,k))' + ...
          (A-B*L(:,:,k))*SiXE*H'*K(:,:,k)' + K(:,:,k)*H*SiXE'*(A-B*L(:,:,k))';
      SiE = newE;
      SiXE = (A-B*L(:,:,k))*SiXE*(A-K(:,:,k)*H)' - E0*E0';
   end;


   % first pass initialization
   if iter==1,
      if Init==0,          % open loop
         K = zeros(szX,szY,N-1);
      elseif Init==2,      % random
         K = randn(szX,szY,N-1);
      end;
   end;
```

```
   % initialize optimal cost-to-go function
   Sx = Q(:,:,N);
   Se = zeros(szX,szX);
   Cost(iter) = 0;

   % backward pass - recompute control policy
   for k=N-1:-1:1

      % update Cost
      Cost(iter) = Cost(iter) + trace(Sx*C0*C0') + ...
         trace(Se*(K(:,:,k)*D0*D0'*K(:,:,k)' + E0*E0' + C0*C0'));

      % Controller
      temp = R + B'*Sx*B;
      BSxeB = B'*(Sx+Se)*B;
      if size(C,2)==1,
         temp = temp + diag(diag(BSxeB).*C.^2);
      else
         for i=1:size(C,3)
            temp = temp + C(:,:,i)'*BSxeB*C(:,:,i);
         end;
      end;
      L(:,:,k) = pinv(temp)*B'*Sx*A;

      % compute new Se
      newE = A'*Sx*B*L(:,:,k) + (A-K(:,:,k)*H)'*Se*(A-K(:,:,k)*H);

      % update Sx and Se
      Sx = Q(:,:,k) + A'*Sx*(A-B*L(:,:,k));
      KSeK = K(:,:,k)'*Se*K(:,:,k);
      if size(D,2)==1,
          Sx = Sx + diag(diag(KSeK).*D.^2);
      else
          for i=1:szD
              Sx = Sx + D(:,:,i)'*KSeK*D(:,:,i);
          end;
      end;
      Se = newE;
   end;


   % adjust cost
   Cost(iter) = Cost(iter) + X1'*Sx*X1 + trace((Se+Sx)*S1);

   % progress bar
   if ~rem(iter,10),
      fprintf('.');
   end;

   % check convergence of Cost
   if (Niter>0 & iter>=Niter) | ...
      (Niter==0 & iter>1 & abs(Cost(iter-1)-Cost(iter))<Eps) | ...
      (Niter==0 & iter>20 & sum(diff(dist(iter-10:iter))>0)>3),
      break;
   end;
end;
```

Todorov & Jordan (2002) Nature Neurosci; Todorov (2005) Neural Comput

Matlab code: optimal feedback control model.



Todorov & Jordan (2002) Nature Neurosci; Todorov (2005) Neural Comput

Infinite-horizon optimal feedback control.

Stochastic dynamics with signal- and state-dependent noises:

$$d\mathbf{x} = \left(\mathbf{Ax} + \mathbf{Bu}\right)dt + \mathbf{Fx}d\beta + \mathbf{Yu}d\gamma + \mathbf{G}d\omega,$$
$$d\mathbf{y} = \mathbf{Cx}dt + \mathbf{D}d\xi$$

Infinite-horizon cost:

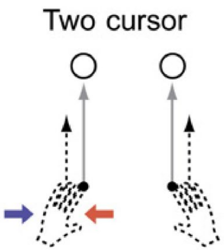
$$J = J_1 + J_2 = \lim_{t\to\infty} \mathrm{E}\left[\tilde{\mathbf{x}}^{\mathrm{T}}\mathbf{U}\tilde{\mathbf{x}}\right] + \mathrm{E}\left[\lim_{t\to\infty}\frac{1}{t}\int_0^t \left(\mathbf{x}^{\mathrm{T}}\mathbf{Qx} + \mathbf{u}^{\mathrm{T}}\mathbf{Ru}\right)dt\right]$$

$$d\hat{\mathbf{x}} = \left(\mathbf{A}\hat{\mathbf{x}} + \mathbf{Bu}\right)dt + \mathbf{K}\left(d\mathbf{y} - \mathbf{C}\hat{\mathbf{x}}dt\right),$$
$$\mathbf{u} = -\mathbf{L}\hat{\mathbf{x}}.$$

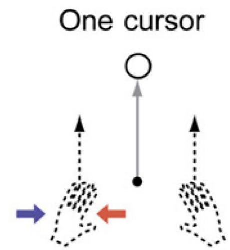
In infinite-horizon optimal feedback control, the Kalman gain ($\mathbf{K}$) and the feedback gain ($\mathbf{L}$) are time invariant.

Phillis (1985) IEEE Trans Auto Contr; Qian et al. (2013) Neural Comput

Bimanual coordination explained by OFC model.



$$J = \sum_{t=1}^{T} w_{p,t}\left(\|p_{L,t}-g_L\|^2+\|p_{R,t}-g_R\|^2\right) + w_{v,t}\left(v_{L,t}^2+v_{R,t}^2\right)+w_u\left(u_{L,t}^2+u_{R,t}^2\right).$$

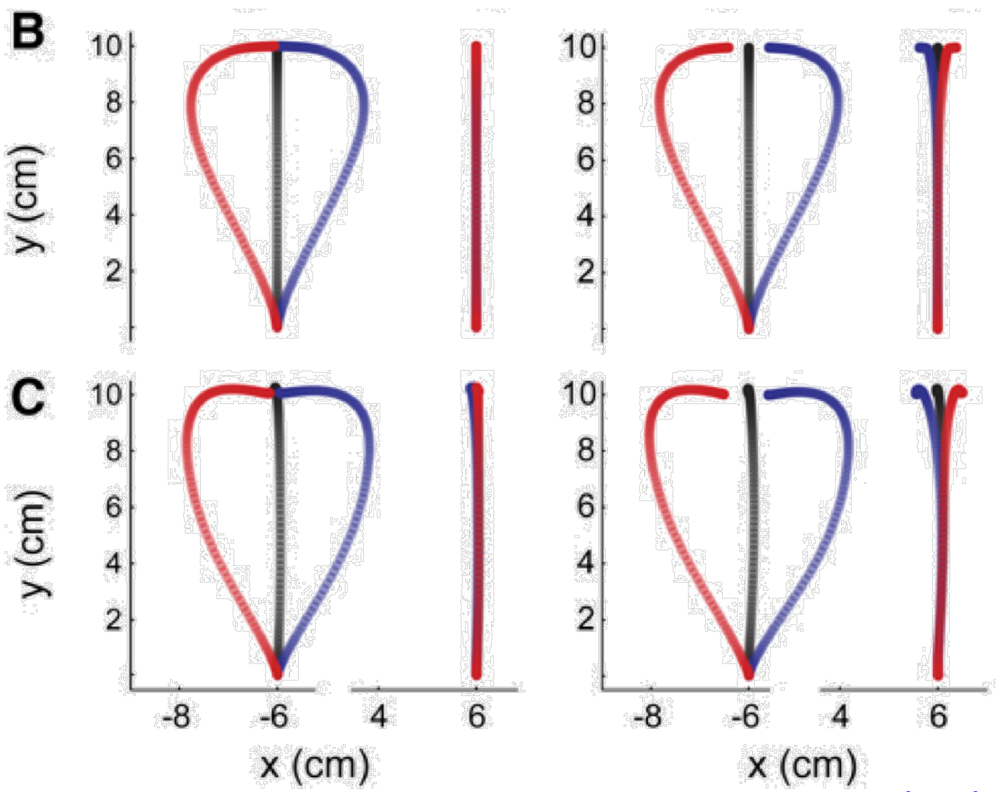


$$J = \sum_{t=1}^{T} w_{p,t}\left\|\frac{x_{L,t}+x_{R,t}}{2}-g\right\|^2+\dots$$

Two cursor condition | One cursor condition

Model

Experiment

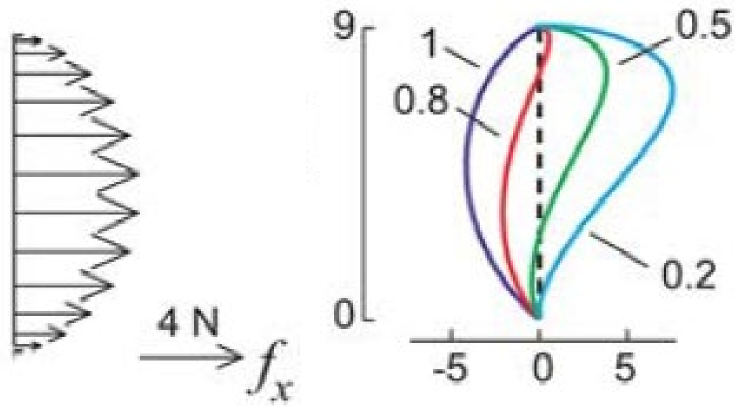


Dietrichsen (2007) Curr Biol

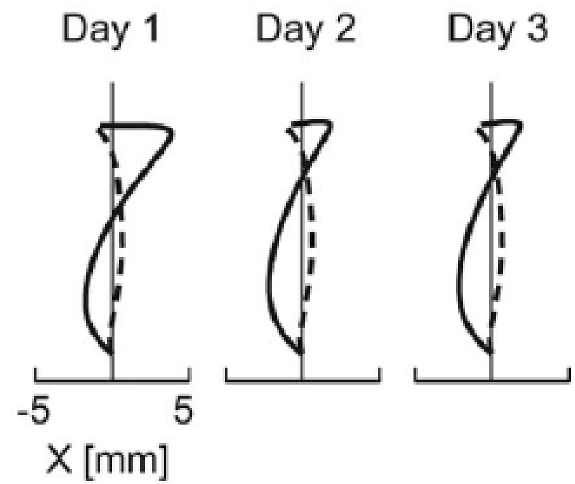
Motor adaptation as reoptimization.

Sigmoidal paths

Model

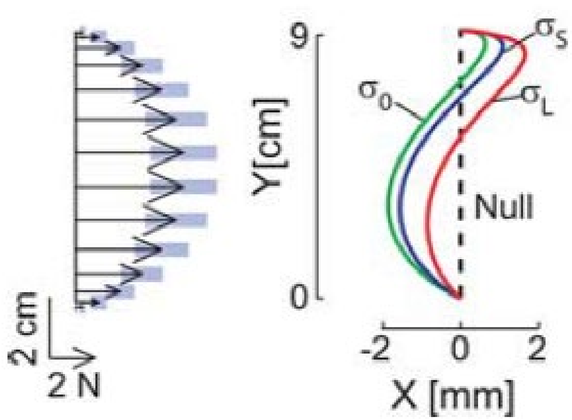


Experiment

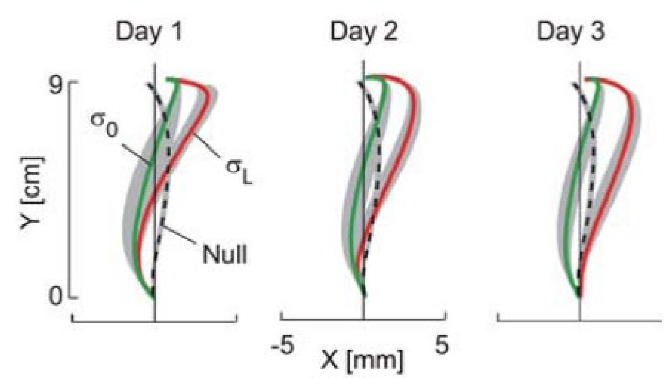


Field uncertainty

Model

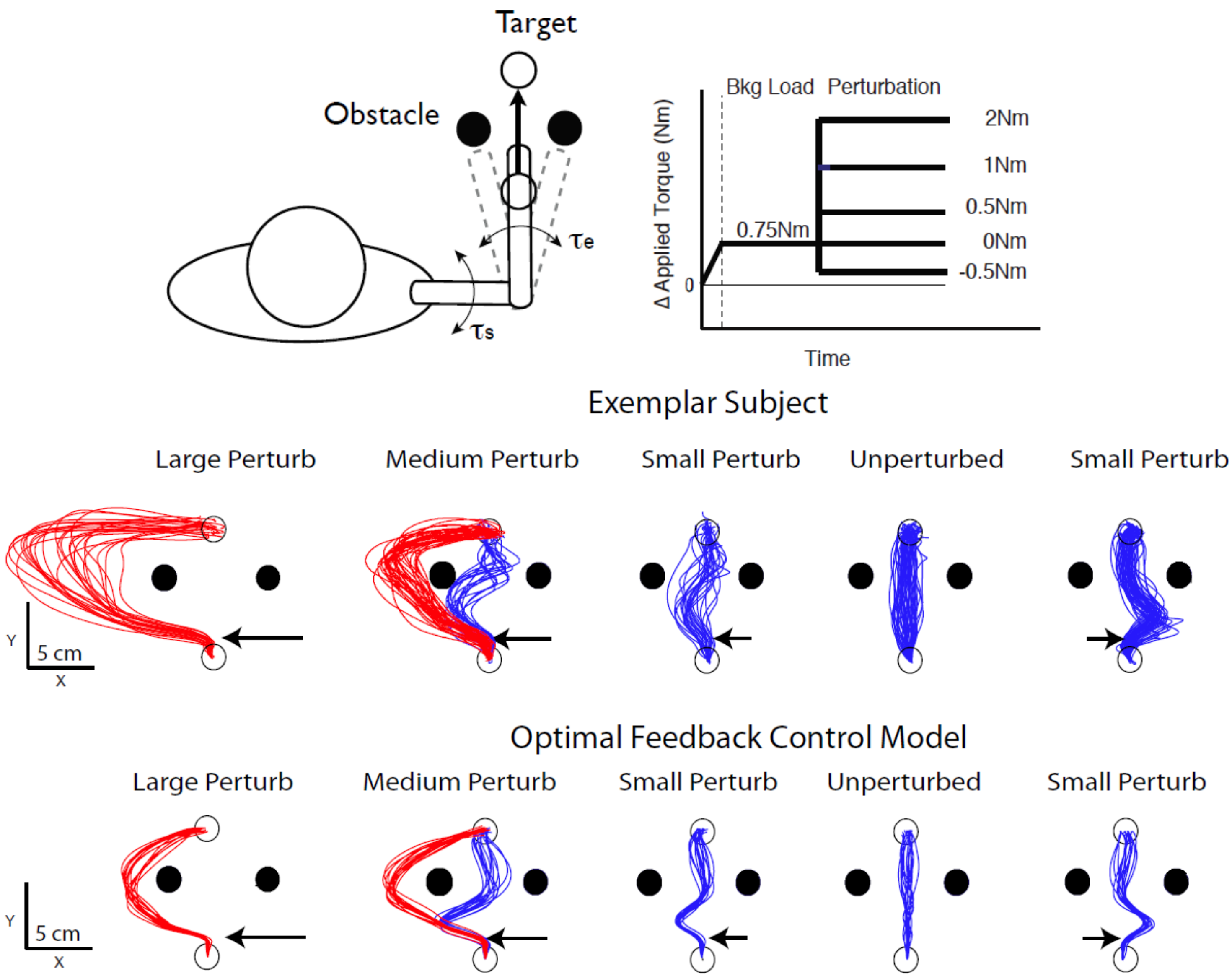


Experiment



Izawa et al. (2008) J Neurosci

Movement as a real-time decision making process.



Nashed et al. (2014) J Neurosci

Learning optimal control policy without knowing dynamics.

$$d\mathbf{x} = \mathbf{A}\mathbf{x}dt + \mathbf{B}\mathbf{u}dt + \mathbf{B}\sum_i \mathbf{C}_i \mathbf{u}\,\eta_i$$

$$J = \int_0^\infty \left(\mathbf{x}^T \mathbf{Q}\mathbf{x} + \mathbf{u}^T \mathbf{R}\mathbf{u}\right) dt$$

$$\mathbf{u} = -\mathbf{K}\mathbf{x}$$

Algebraic Riccati equation:

$$0 = \mathbf{A}^T \mathbf{P} + \mathbf{P}\mathbf{A} - \mathbf{Q} + \mathbf{P}\mathbf{B}\mathbf{R}^{-1}\mathbf{B}\mathbf{P}$$

$$\mathbf{K} = \mathbf{R}^{-1}\mathbf{B}^T \mathbf{P}$$

The Riccati equation for the optimal feedback gain requires explicit knowledge of the forward model (the matrices $\mathbf{A}$ and $\mathbf{B}$)!

Jiang & Jiang (2014) Biol Cybern

Learning optimal control policy without knowing dynamics.

Algebraic Riccati equation:

$$0 = \mathbf{A}^T\mathbf{P} + \mathbf{P}\mathbf{A} + \mathbf{Q} - \mathbf{P}\mathbf{B}\mathbf{R}^{-1}\mathbf{B}\mathbf{P}$$

Iterative solution to the Riccati equation:

$$0 = \left(\mathbf{A} - \mathbf{B}\mathbf{K}_k\right)^T \mathbf{P}_k + \mathbf{P}_k\left(\mathbf{A} - \mathbf{B}\mathbf{K}_k\right) + \mathbf{Q} - \mathbf{K}_k\mathbf{R}\mathbf{K}_k$$

$$\mathbf{K}_{k+1} = \mathbf{R}^{-1}\mathbf{B}^T\mathbf{P}_k$$

1) $\mathrm{A} - \mathrm{BK}$ is Hurwitz.
2) $P^* \leq P_{k+1} \leq P_k$
3) $K_k \rightarrow K^*$ and $P_k \rightarrow P^*$

Jiang & Jiang (2014) Biol Cybern

Learning optimal control policy without knowing dynamics.

Algebraic Riccati equation:

$$0 = \mathbf{A}^T\mathbf{P} + \mathbf{P}\mathbf{A} - \mathbf{Q} + \mathbf{P}\mathbf{B}\mathbf{R}^{-1}\mathbf{B}\mathbf{P}$$

Iterative solution to the Riccati equation:

$$0 = \left(\mathbf{A} - \mathbf{B}\mathbf{K}_k\right)^T \mathbf{P}_k + \mathbf{P}_k\left(\mathbf{A} - \mathbf{B}\mathbf{K}_k\right) - \mathbf{Q} + \mathbf{K}_k\mathbf{R}\mathbf{K}_k$$

$$\mathbf{K}_{k+1} = \mathbf{R}^{-1}\mathbf{B}^T\mathbf{P}_k$$

1) $\mathrm{A} - \mathrm{BK}$ is Hurwitz.
2) $P^* \le P_{k+1} \le P_k$
3) $K_k \to K^*$ and $P_k \to P^*$

Jiang & Jiang (2014) Biol Cybern

Learning optimal control policy without knowing dynamics.

Problem: In determining optimal policy, the system matrices need to be known:

$$0 = \mathbf{A}^T\mathbf{P} + \mathbf{P}\mathbf{A} + \mathbf{Q} - \mathbf{P}\mathbf{B}\mathbf{R}^{-1}\mathbf{B}\mathbf{P}$$

How can optimal policy be derived without assuming the direct knowledge of the system matrices?

Solution: Note that

$$\begin{aligned} 0 &= \mathbf{x}^T\left[\mathbf{A}^T\mathbf{P} + \mathbf{P}\mathbf{A} + \mathbf{Q} - \mathbf{P}\mathbf{B}\mathbf{R}^{-1}\mathbf{B}\mathbf{P}\right]\mathbf{x}dt \\ &= d\left(\mathbf{x}^T\mathbf{P}\mathbf{x}\right) + \mathbf{x}^T\left[\mathbf{Q} + \mathbf{K}^T\mathbf{R}\mathbf{K}\right]\mathbf{x}dt \end{aligned}$$

Therefore, the knowledge of $(\mathbf{A}, \mathbf{B})$ is replaced with the knowledge of $d\mathbf{x}$.

Jiang & Jiang (2014) Biol Cybern

Learning optimal control policy without knowing dynamics.

By integrating over an infinitesimal interval:

$$0 = d\left(\mathbf{x}^T \mathbf{P} \mathbf{x}\right) + \mathbf{x}^T \left[\mathbf{Q} + \mathbf{K}^T \mathbf{R} \mathbf{K}\right] \mathbf{x} dt$$

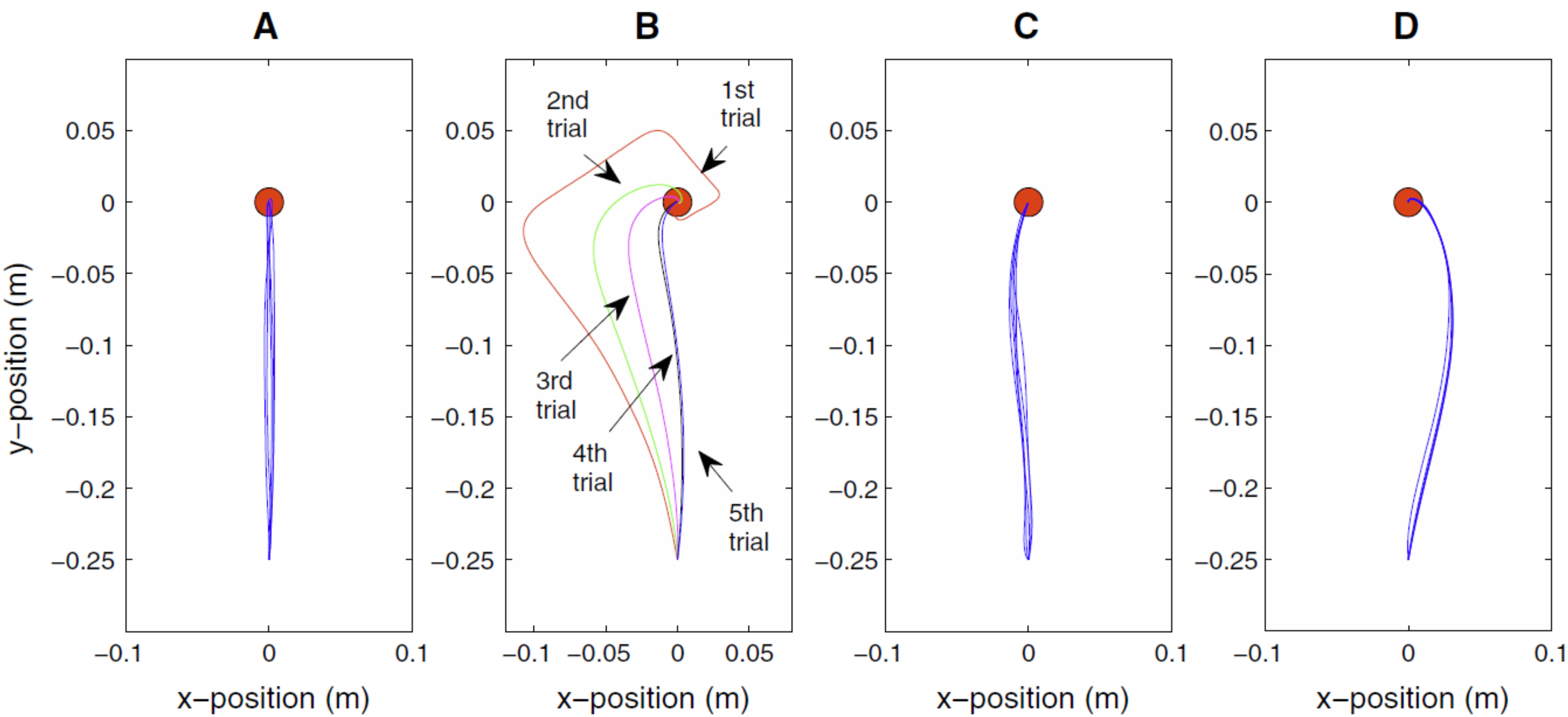
By integrating over an infinitesimal interval:

$$\mathbf{x}^T\left(t+\delta t\right) \mathbf{P} \mathbf{x}\left(t+\delta t\right) - \mathbf{x}^T\left(t\right) \mathbf{P} \mathbf{x}\left(t\right) = -\int_t^{t+\delta t} \mathbf{x}^T \left[\mathbf{Q} + \mathbf{K}^T \mathbf{R} \mathbf{K}\right] \mathbf{x} dt$$

Therefore, without knowing the system dynamics, the matrices **P** and **K** are determined.

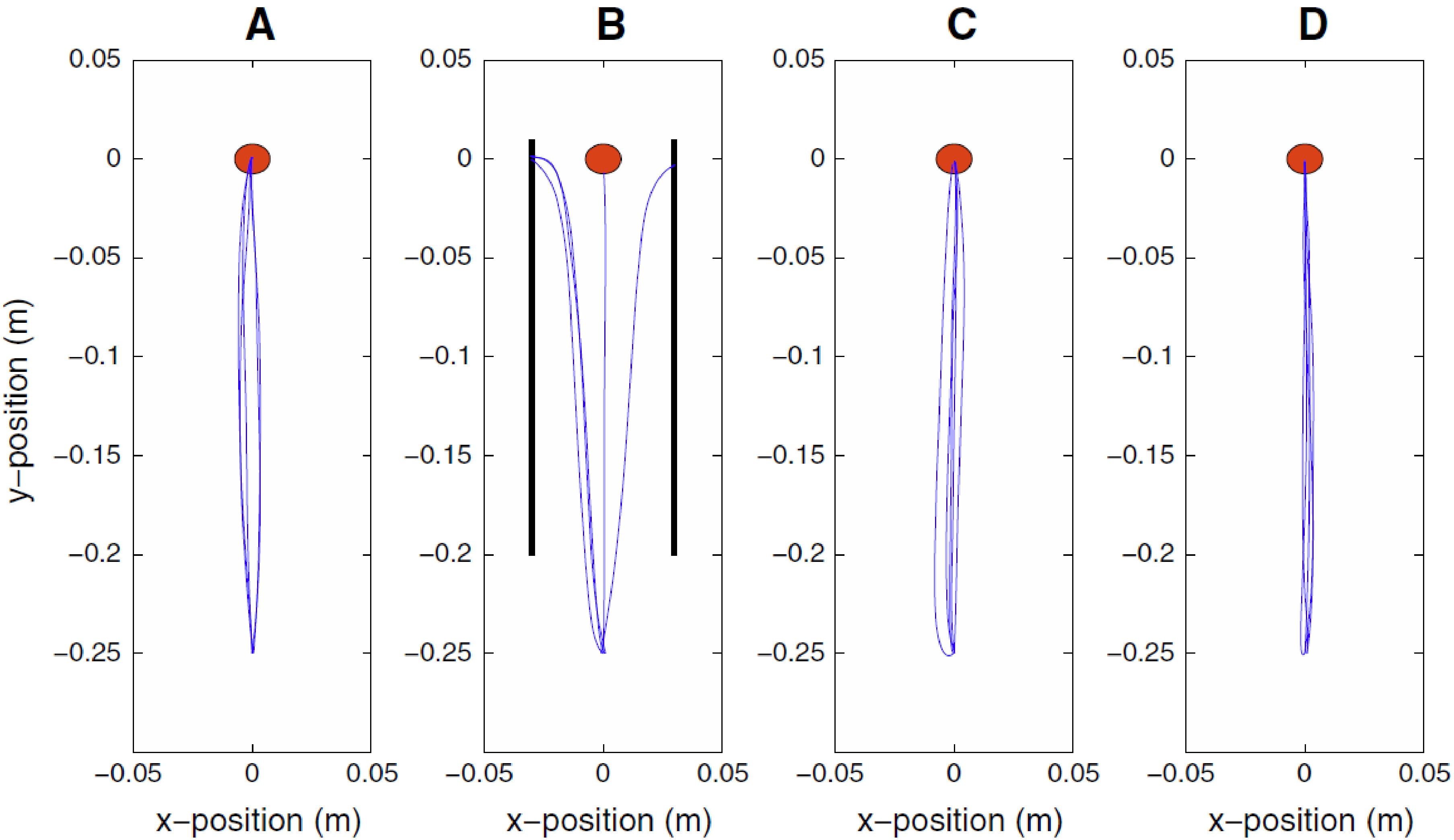
Jiang & Jiang (2014) Biol Cybern

Learning optimal control policy without knowing dynamics.



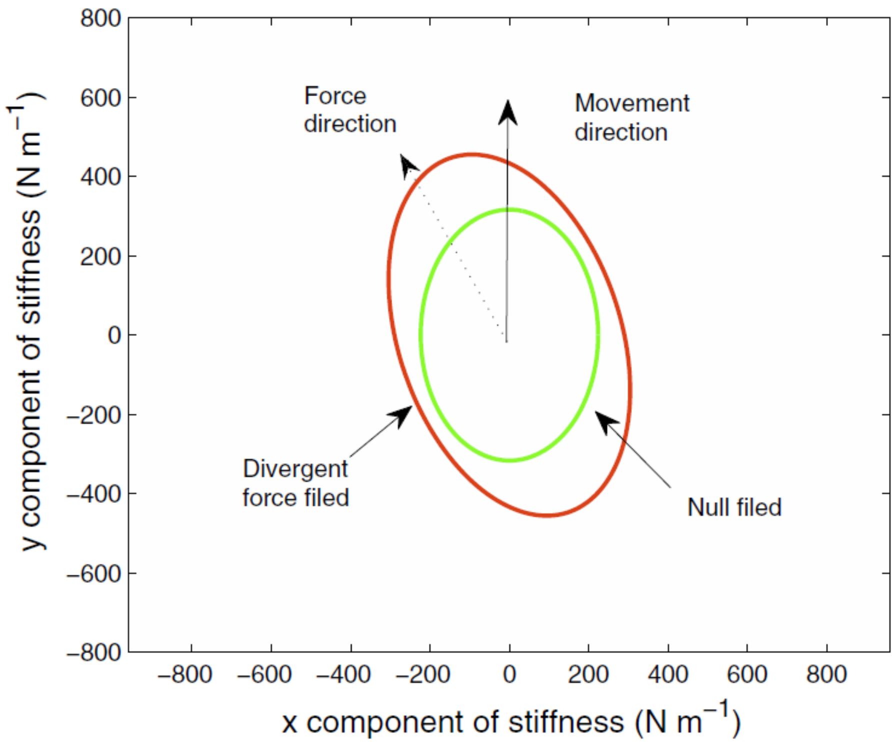
Jiang & Jiang (2014) Biol Cybern

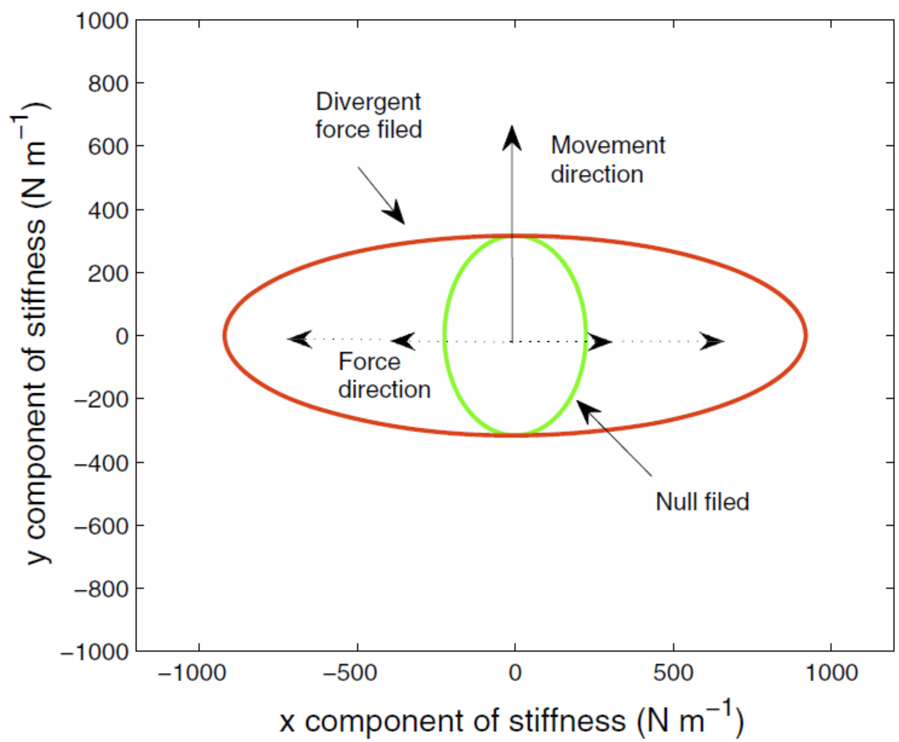
Learning optimal control policy without knowing dynamics.



Jiang & Jiang (2014) Biol Cybern

Learning optimal control policy without knowing dynamics.





Jiang & Jiang (2014) Biol Cybern

Summary

- Feedforward and feedback control differ in stochastic dynamics.
- Signal-dependent control noise contributes to trial-by-trial motor variability.
- Movement accuracy under signal-dependent noise models human movements.
- Optimal feedback control (OFC) integrates state estimation and motor control in a single computational framework.
- A number of psychophysical experiments can be explained by the OFC model.

References

- Schmidt, R. A., Zelaznik, H., Hawkins, B., Frank, J. S., & Quinn Jr, J. T. (1979). Motor-output variability: a theory for the accuracy of rapid motor acts. Psychological review, 86(5), 415.
- van Beers, R. J., Haggard, P., & Wolpert, D. M. (2004). The role of execution noise in movement variability. Journal of Neurophysiology, 91(2), 1050-1063.
- Harris, C. M., & Wolpert, D. M. (1998). Signal-dependent noise determines motor planning. Nature, 394(6695), 780-784.
- Tanaka, H., Krakauer, J. W., & Qian, N. (2006). An optimization principle for determining movement duration. Journal of neurophysiology, 95(6), 3875-3886.
- Todorov, E., & Jordan, M. I. (2002). Optimal feedback control as a theory of motor coordination. Nature neuroscience, 5(11), 1226-1235.
- Todorov, E. (2005). Stochastic optimal control and estimation methods adapted to the noise characteristics of the sensorimotor system. Neural computation, 17(5), 1084-1108.
- Athans, M. (1967). The matrix minimum principle. Information and control, 11(5), 592-606.
- Phillis, Y. A. (1985). Controller design of systems with multiplicative noise. Automatic Control, IEEE Transactions on, 30(10), 1017-1019.
- Qian, N., Jiang, Y., Jiang, Z. P., & Mazzoni, P. (2013). Movement duration, fitts's law, and an infinite-horizon optimal feedback control model for biological motor systems. Neural computation, 25(3), 697-724.
- Diedrichsen, J. (2007). Optimal task-dependent changes of bimanual feedback control and adaptation. Current Biology, 17(19), 1675-1679.
- Izawa, J., Rane, T., Donchin, O., & Shadmehr, R. (2008). Motor adaptation as a process of reoptimization. The Journal of Neuroscience, 28(11), 2883-2891.
- Nashed, J. Y., Crevecoeur, F., & Scott, S. H. (2014). Rapid online selection between multiple motor plans. The Journal of Neuroscience, 34(5), 1769-1780.
- Jiang, Y., & Jiang, Z. P. (2014). Adaptive dynamic programming as a theory of sensorimotor control. Biological cybernetics, 108(4), 459-473.

Exercise

- Write a Matlab code of the minimum-variance model for a trajectory with given initial and final positions.
- Write a Matlab code of the optimal feedback control for a trajectory with given initial and final positions.
- Investigate how the minimum-variance model and the optimal feedback model differ.