The 12th World Congress on Intelligent Control and Automation
Steve and Rosalind Hsia Best Biomedical Paper Award

Model-free Robust Optimal Feedback Mechanisms of Biological Motor Control

Tao Bian
Advisor: Zhong-Ping Jiang

Control and Networks Lab, Department of Electrical and Computer Engineering, Tandon School of Engineering, New York University, 5 Metrotech Center, Brooklyn, NY 11201, USA.

NYU

TANDON
SCHOOL OF
ENGINEERING

Outline

- What is biological motor control?
- Recent developments in biological motor control
- Our contributions
- Simulation validation
- Conclusions and future work

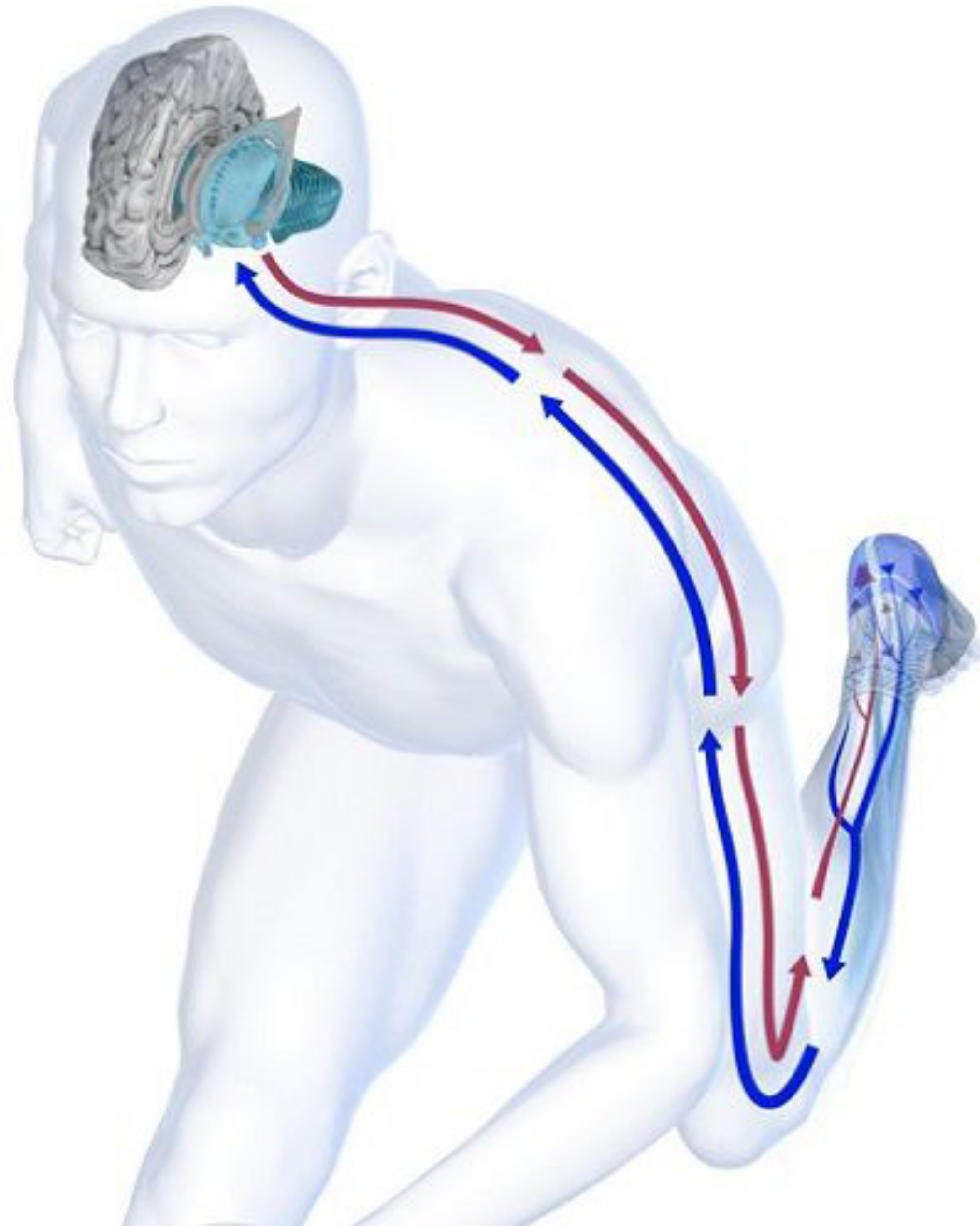
Outline

- What is biological motor control?
- Recent developments in biological motor control
- Our contributions
- Simulation validation
- Conclusions and future work

What is biological motor control?

What is biological motor control?

How does the motor system
control and learn
the motor movement?



Applications

Parkinson's disease

- Over 700,000 patients in the world
- Disorder of the motor system

Robotic prostheses

- Improved ambulation safety, and quality

Humanoid robot

- Robots performing human tasks

Outline

- What is biological motor control?
- Recent developments in biological motor control
- Our contributions
- Simulation validation
- Conclusions and future work

Key developments

Key developments

- Minimum jerk/torque-change model
  (Flash & Hogan, 1985, The Journal of Neuroscience; Uno, et.al., 1989, Biological Cybernetics)

Key developments

- Minimum jerk/torque-change model
(Flash & Hogan, 1985, The Journal of Neuroscience; Uno, et.al., 1989, Biological Cybernetics)
- Minimizing endpoint variance model
(Harris & Wolpert, 1998, Nature; Haruno & Wolpert, 2005, Journal of Neurophysiology)

Key developments

- Minimum jerk/torque-change model
  (Flash & Hogan, 1985, The Journal of Neuroscience; Uno, et.al., 1989, Biological Cybernetics)
- Minimizing endpoint variance model
  (Harris & Wolpert, 1998, Nature; Haruno & Wolpert, 2005, Journal of Neurophysiology)
- Optimal impedance control
  (Burdet, et.al., 2001, Nature)

Key developments

- Minimum jerk/torque-change model
  (Flash & Hogan, 1985, The Journal of Neuroscience; Uno, et.al., 1989, Biological Cybernetics)
- Minimizing endpoint variance model
  (Harris & Wolpert, 1998, Nature; Haruno & Wolpert, 2005, Journal of Neurophysiology)
- Optimal impedance control
  (Burdet, et.al., 2001, Nature)
- Optimal control-theoretic approach
  (Todorov & Jordan, 2002, Nature Neuroscience; Todorov, 2004, Nature Neuroscience)

Problems of interest

Problems of interest

- Huang, et.al., 2011, Neuron.
  - Model-based or Model-free?
  - Instead of control after building a model, historical movement experience is directly used for learning and control.
  - No existing theory can explain this.

Problems of interest

- Huang, et.al., 2011, Neuron.
  - Model-based or Model-free?
  - Instead of control after building a model, historical movement experience is directly used for learning and control.
  - No existing theory can explain this.
- Wu, et.al., 2014, Nature Neuroscience.
  - Motor variability: Uncertainties (noise, errors, …) is useful for learning!
  - No existing theory can explain this either.

Outline

- What is biological motor control?
- Recent developments in biological motor control
- Our contributions
- Simulation validation
- Conclusions and future work

Contributions

Contributions

1. A new computational principle for biological motor control
   - Explaining model-free motor learning
   - Proving the contribution of motor variability

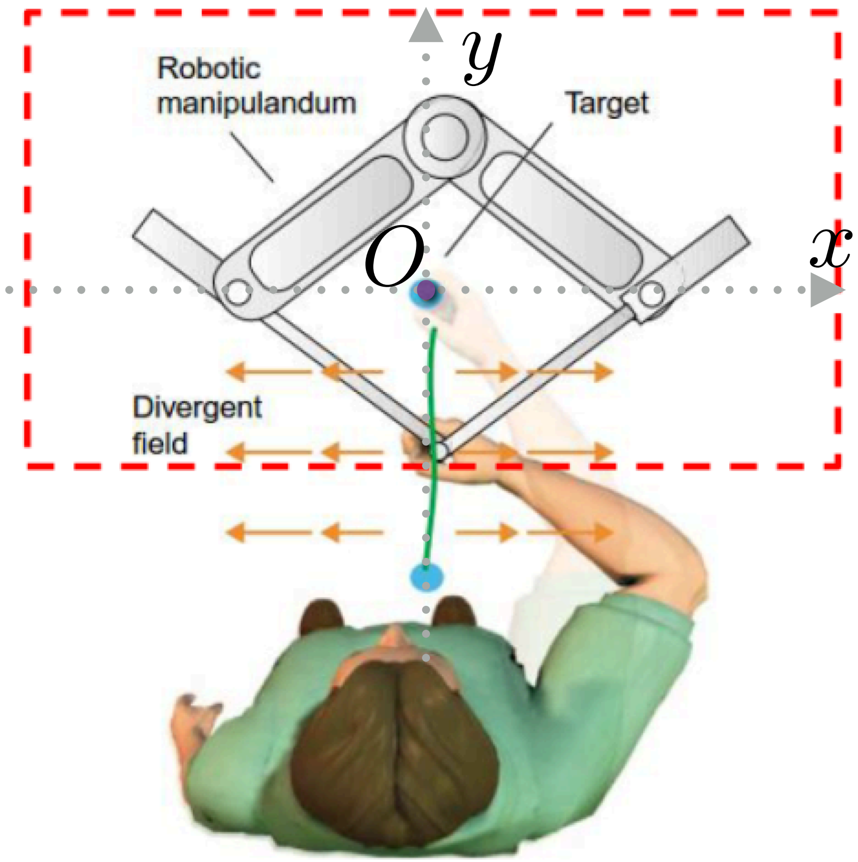
Contributions

1. A new computational principle for biological motor control
    - Explaining model-free motor learning
    - Proving the contribution of motor variability
2. A new contribution to adaptive optimal control theory
    - The first value iteration-based stochastic adaptive dynamic programming

Motor learning model

Consider the following model
(Todorov, 2005, Neural Computation):

$$dp = vdt,$$
$$mdv = (a - bv + f)dt,$$
$$\tau da = (u - a + e + \Delta(\varsigma, \xi))dt + d\zeta,$$
$$T\dot{\varsigma} = -\varsigma + a - \gamma_1 \tanh(a), \quad \Delta = \gamma_0 \varsigma.$$



Suboptimal inference (Beck, et.al., 2012, Neuron).

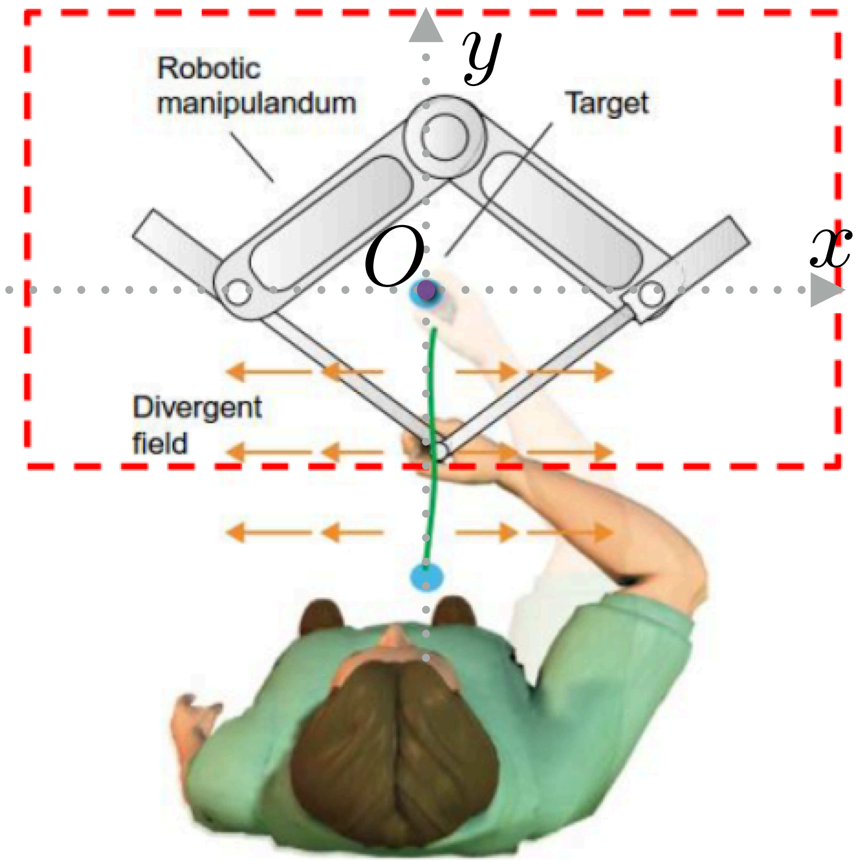
Signal-dependent noise:

$$d\zeta = \begin{bmatrix} c_1 & 0 \\ c_2 & 0 \end{bmatrix} \begin{bmatrix} u_x \\ u_y \end{bmatrix} dw_1 + \begin{bmatrix} 0 & -c_2 \\ 0 & c_1 \end{bmatrix} \begin{bmatrix} u_x \\ u_y \end{bmatrix} dw_2$$

Motor learning model

Consider the following model
(Todorov, 2005, Neural Computation):

$$dp = vdt,$$
$$mdv = (a - bv + f)dt,$$
$$\tau da = (u - a + e + \Delta(\varsigma, \xi))dt + d\zeta,$$
$$T\dot{\varsigma} = -\varsigma + a - \gamma_1 \tanh(a), \quad \Delta = \gamma_0 \varsigma.$$



Suboptimal inference (Beck, et.al., 2012, Neuron).

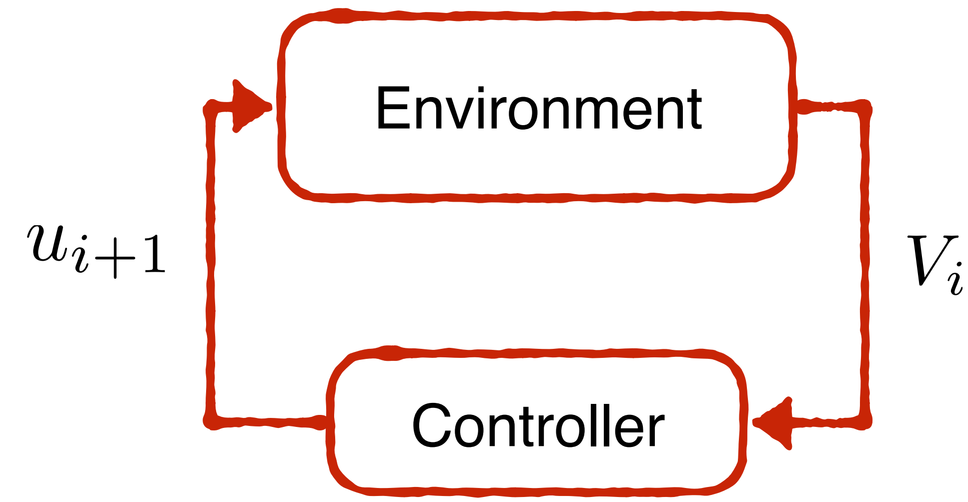
Signal-dependent noise:

$$d\zeta = \begin{bmatrix} c_1 & 0 \\ c_2 & 0 \end{bmatrix} \begin{bmatrix} u_x \\ u_y \end{bmatrix} dw_1 + \begin{bmatrix} 0 & -c_2 \\ 0 & c_1 \end{bmatrix} \begin{bmatrix} u_x \\ u_y \end{bmatrix} dw_2$$

1. How to develop model-free learning?
2. How to guarantee the robustness?

Adaptive dynamic programming

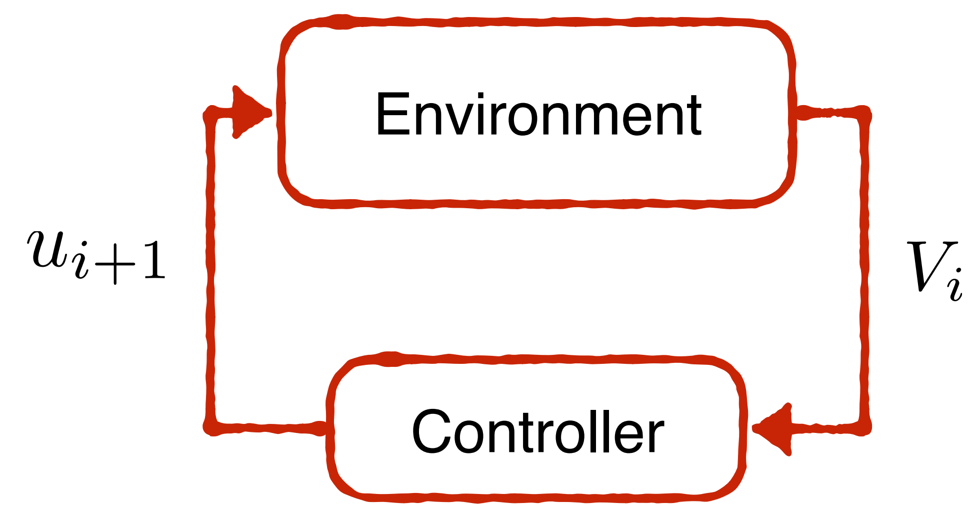
Adaptive dynamic programming (ADP) aims at finding a stabilizing optimal control policy for feedback control systems via online learning.



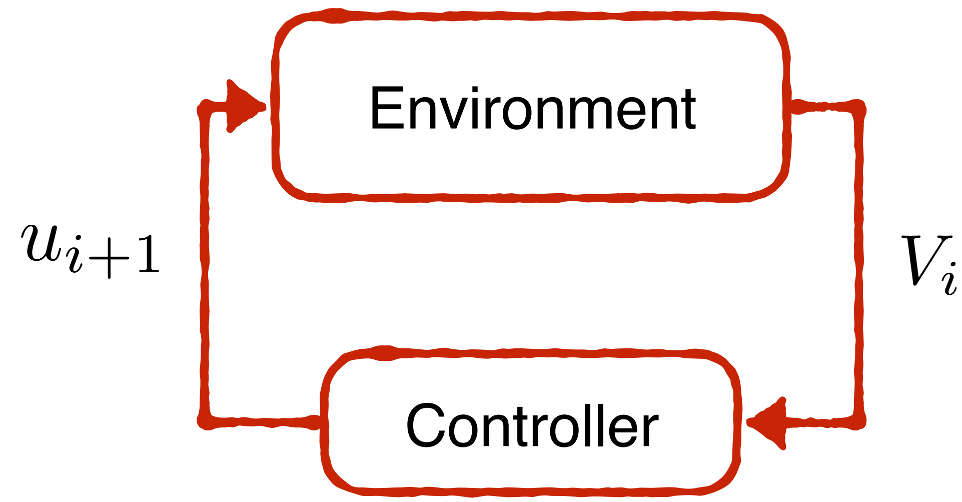
Adaptive dynamic programming

Adaptive dynamic programming (ADP) aims at finding a stabilizing optimal control policy for feedback control systems via online learning.

ADP is ideal for studying the model-free learning.



Adaptive dynamic programming



Adaptive dynamic programming (ADP) aims at finding a stabilizing optimal control policy for feedback control systems via online learning.

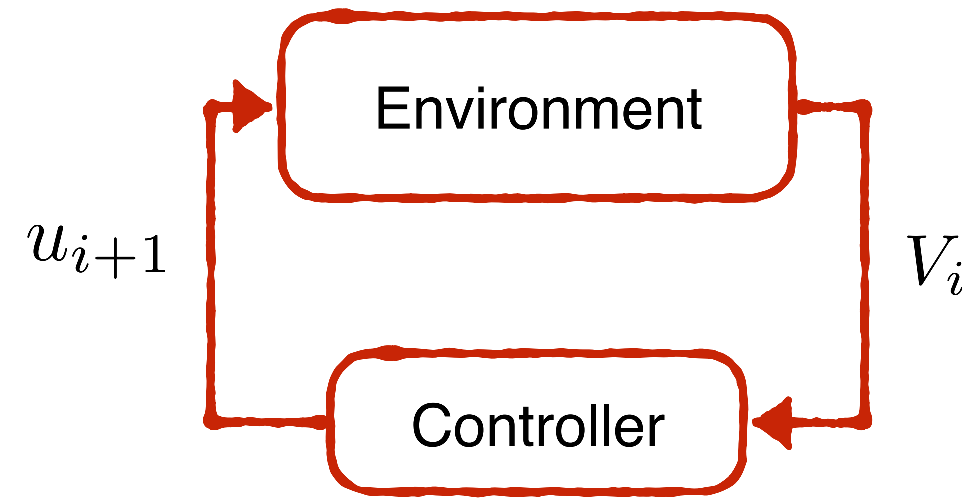
ADP is ideal for studying the model-free learning.

ADP has been studied by different research groups:

D. Vrabie, K. G. Vamvoudakis, and F. L. Lewis (2013). Optimal Adaptive Control and Differential Games by Reinforcement Learning Principles, IET, London, UK.
F. L. Lewis and D. Liu, editors (2013). Reinforcement Learning and Approximate Dynamic Programming for Feedback Control. John Wiley & Sons, Inc., Piscataway, NJ.
H. Zhang, D. Liu, Y. Luo, and D. Wang (2013). Adaptive Dynamic Programming for Control: Algorithms and Stability. Springer London.
T. Bian, Y. Jiang, and Z. P. Jiang (2014). Adaptive dynamic programming and optimal control of nonlinear nonaffine systems. Automatica 50(10), 2624–2632.
T. Bian, Y. Jiang, and Z. P. Jiang (2015). Decentralized adaptive optimal control of large-scale systems with application to power systems. IEEE Transactions on Industrial Electronics 62(4), 2439–2447.
T. Bian, and Z. P. Jiang (2016e). Value iteration and adaptive dynamic programming for data-driven adaptive optimal control design. Automatica (Accepted).

Adaptive dynamic programming

Adaptive dynamic programming (ADP) aims at finding a stabilizing optimal control policy for feedback control systems via online learning.



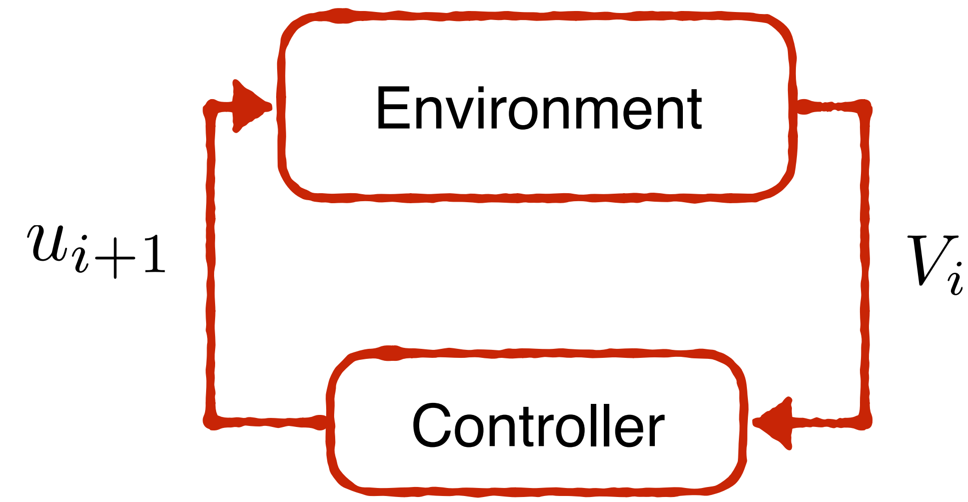
Existing ADP result for CT linear deterministic systems:

$$\begin{cases} \dot{\xi} = A\xi + Bu, \\ \mathcal{J}(\xi(0); u) = \int_0^\infty (\xi^T Q \xi + u^T R u) dt, \end{cases} \quad \begin{aligned} u^* &= -K^* \xi = -R^{-1} B^T P^* \xi, \\ 0 &= A^T P^* + P^* A - P^* B R^{-1} B^T P^* + Q. \end{aligned}$$

If K_0 is stabilizing, $\{K_k\}$, $\{P_k\}$: $K_k \to K^*$, $P_k \to P^*$.

Adaptive dynamic programming

Adaptive dynamic programming (ADP) aims at finding a stabilizing optimal control policy for feedback control systems via online learning.



Existing ADP result for CT linear deterministic systems:

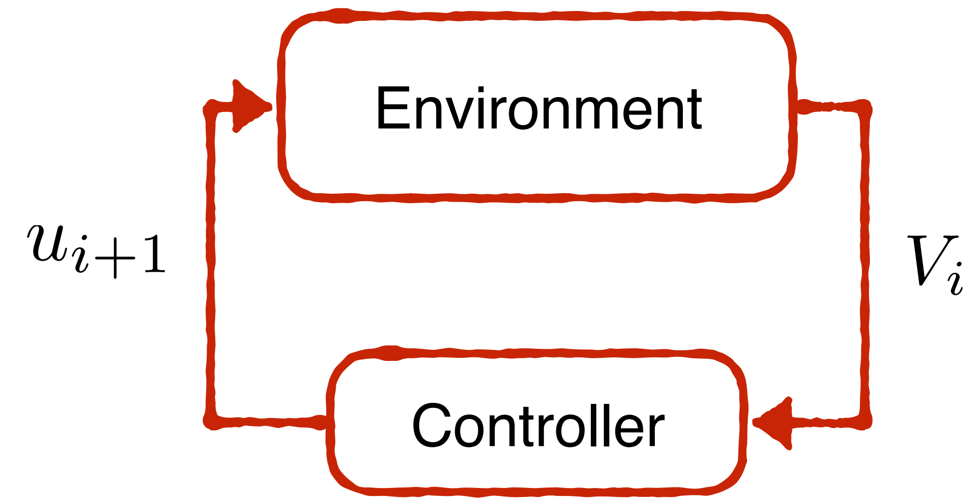
$$\begin{cases} \dot{\xi} = A\xi + Bu, \\ \mathcal{J}(\xi(0); u) = \int_0^\infty (\xi^T Q \xi + u^T R u) dt, \end{cases} \quad \begin{aligned} u^* &= -K^*\xi = -R^{-1}B^T P^* \xi, \\ 0 &= A^T P^* + P^* A - P^* B R^{-1} B^T P^* + Q. \end{aligned}$$

If K_0 is stabilizing, $\{K_k\}$, $\{P_k\}$: $K_k \to K^*$, $P_k \to P^*$.

Remaining issues with CT ADP:
1. ADP for stochastic systems?
2. ADP in unstable environment (unknown K_0)?

Adaptive dynamic programming

Adaptive dynamic programming (ADP) aims at finding a stabilizing optimal control policy for feedback control systems via online learning.



Existing ADP result for CT linear deterministic systems:

$$\begin{cases} \dot{\xi} = A\xi + Bu, \\ \mathcal{J}(\xi(0); u) = \int_0^\infty (\xi^T Q \xi + u^T R u) dt, \end{cases} \quad \begin{aligned} u^* &= -K^* \xi = -R^{-1} B^T P^* \xi, \\ 0 &= A^T P^* + P^* A - P^* B R^{-1} B^T P^* + Q. \end{aligned}$$

If K_0 is stabilizing, $\{K_k\}$, $\{P_k\}$: $K_k \to K^*$, $P_k \to P^*$.

Remaining issues with CT ADP:

1. ADP for stochastic systems?
2. ADP in unstable environment (unknown K_0)?

CT value iteration-based stochastic ADP

Stochastic ADP algorithm

CT value iteration:

$$\begin{cases} P_{k+1} = P_k + \epsilon_k(A^T P_k + P_k A - P_k B R^{-1} B^T P_k + Q), \\ K_{k+1} = R^{-1} B^T P_k, \end{cases}$$

- Instead of starting from a stabilizing K_0,
  we start from a symmetric and positive definite P_0.
- In value iteration, K_1 may not be stable.

Stochastic ADP algorithm

CT value iteration:

$$\begin{cases} P_{k+1} = P_k + \epsilon_k (A^T P_k + P_k A - P_k B R^{-1} B^T P_k + Q), \\ K_{k+1} = R^{-1} B^T P_k, \end{cases}$$

- Instead of starting from a stabilizing K_0,
  we start from a symmetric and positive definite P_0.
- In value iteration, K_1 may not be stable.

How to develop the stochastic ADP algorithm?

Stochastic ADP algorithm

Rewrite the motor system model in the following form:

$$
\begin{aligned}
d\xi &= A\xi dt + Bd\nu, \\
d\nu &= (u + e + \Delta(\varsigma, \xi))dt + G(u + e + \Delta(\varsigma, \xi))dw, \\
d\varsigma &= f(\varsigma, \xi)dt + g(\varsigma, \xi)d\eta.
\end{aligned}
$$

Stochastic ADP algorithm

Rewrite the motor system model in the following form:

$$
\begin{aligned}
d\xi &= A\xi dt + B d\nu, \\
d\nu &= (u + e + \Delta(\varsigma, \xi))dt + G(u + e + \Delta(\varsigma, \xi))dw, \\
d\varsigma &= f(\varsigma, \xi)dt + g(\varsigma, \xi)d\eta.
\end{aligned}
$$

$$
d(\xi^T P_k \xi) = \xi^T (A^T P_k + P_k A)\xi dt + 2\xi^T K_{k+1}^T R d\nu + d\xi^T P_k d\xi
$$

Stochastic ADP algorithm

Rewrite the motor system model in the following form:

$$
\begin{aligned}
d\xi &= A\xi dt + B d\nu, \\
d\nu &= (u + e + \Delta(\varsigma, \xi))dt + G(u + e + \Delta(\varsigma, \xi))dw, \\
d\varsigma &= f(\varsigma, \xi)dt + g(\varsigma, \xi)d\eta.
\end{aligned}
$$

$$
d(\xi^T P_k \xi) = \xi^T (A^T P_k + P_k A)\xi dt + 2\xi^T K_{k+1}^T R d\nu + d\xi^T P_k d\xi
$$

Denote $H_k = A^T P_k + P_k A$

$$
\begin{aligned}
\xi^T P_k \xi \,\big|_t^{t+\delta t} = & \left(\int_t^{t+\delta t} \xi \otimes \xi ds \right)^T \mathrm{vec}(H_k) + \int_t^{t+\delta t} d\xi^T P_k d\xi \\
& + 2 \left(\int_t^{t+\delta t} \xi \otimes R d\nu \right)^T \mathrm{vec}(K_{k+1}).
\end{aligned}
$$

Stochastic ADP algorithm

Rewrite the motor system model in the following form:

$$
\begin{aligned}
d\xi &= A\xi dt + B d\nu, \\
d\nu &= (u + e + \Delta(\varsigma, \xi))dt + G(u + e + \Delta(\varsigma, \xi))dw, \\
d\varsigma &= f(\varsigma, \xi)dt + g(\varsigma, \xi)d\eta.
\end{aligned}
$$

$$
d(\xi^T P_k \xi) = \xi^T (A^T P_k + P_k A)\xi dt + 2\xi^T K_{k+1}^T R d\nu + d\xi^T P_k d\xi
$$

Denote $H_k = A^T P_k + P_k A$

$$
\xi^T P_k \xi \mid_t^{t+\delta t} = \left(\int_t^{t+\delta t} \xi \otimes \xi ds \right)^T \mathrm{vec}(H_k) + \int_t^{t+\delta t} d\xi^T P_k d\xi
$$

$$
+ 2 \left(\int_t^{t+\delta t} \xi \otimes R d\nu \right)^T \mathrm{vec}(K_{k+1}).
$$

Stochastic ADP algorithm

Continuous-time VI:

$$\begin{cases} P_{k+1} = P_k + \epsilon_k (A^T P_k + P_k A - P_k B R^{-1} B^T P_k + Q), \\ K_{k+1} = R^{-1} B^T P_k, \end{cases}$$

Stochastic ADP algorithm:

$$\begin{bmatrix} \mathrm{vecs}(H_k) \\ \mathrm{vec}(K_{k+1}) \end{bmatrix} = [I_{\xi\xi}, 2I_{\xi v}]^{\dagger} (\delta_{\xi\xi} - I_{d\xi\xi}) \mathrm{vec}(P_k)$$

$$P_{k+1} \leftarrow P_k + \epsilon_k (H_k + Q - K_{k+1}^T R K_{k+1})$$

Stochastic ADP algorithm

Continuous-time VI:

$$\begin{cases} P_{k+1} = P_k + \epsilon_k (A^T P_k + P_k A - P_k B R^{-1} B^T P_k + Q), \\ K_{k+1} = R^{-1} B^T P_k, \end{cases}$$

Stochastic ADP algorithm:

Online data

$$\begin{bmatrix} \mathrm{vecs}(H_k) \\ \mathrm{vec}(K_{k+1}) \end{bmatrix} = [I_{\xi\xi}, 2I_{\xi v}]^\dagger (\delta_{\xi\xi} - I_{d\xi\xi}) \mathrm{vec}(P_k)$$

$$P_{k+1} \leftarrow P_k + \epsilon_k (H_k + Q - K_{k+1}^T R K_{k+1})$$

Stochastic ADP algorithm

Continuous-time VI:

$$\begin{cases} P_{k+1} = P_k + \epsilon_k (A^T P_k + P_k A - P_k B R^{-1} B^T P_k + Q), \\ K_{k+1} = R^{-1} B^T P_k, \end{cases}$$

Stochastic ADP algorithm:

$$\begin{bmatrix} \mathrm{vecs}(H_k) \\ \mathrm{vec}(K_{k+1}) \end{bmatrix} = [I_{\xi\xi}, 2I_{\xi v}]^{\dagger} (\delta_{\xi\xi} - I_{d\xi\xi}) \mathrm{vec}(P_k)$$

$$P_{k+1} \leftarrow P_k + \epsilon_k (H_k + Q - K_{k+1}^T R K_{k+1})$$

Well defined due to the motor variability

Stochastic ADP algorithm

Continuous-time VI:

$$\begin{cases} P_{k+1} = P_k + \epsilon_k (A^T P_k + P_k A - P_k B R^{-1} B^T P_k + Q), \\ K_{k+1} = R^{-1} B^T P_k, \end{cases}$$

Stochastic ADP algorithm:

$$\begin{bmatrix} \mathrm{vecs}(H_k) \\ \mathrm{vec}(K_{k+1}) \end{bmatrix} = [I_{\xi\xi}, 2I_{\xi v}]^{\dagger} (\delta_{\xi\xi} - I_{d\xi\xi}) \mathrm{vec}(P_k)$$

$$P_{k+1} \leftarrow P_k + \epsilon_k (H_k + Q - K_{k+1}^T R K_{k+1})$$

Step size

Stochastic ADP algorithm

Continuous-time VI:

$$\begin{cases} P_{k+1} = P_k + \epsilon_k(A^T P_k + P_k A - P_k B R^{-1} B^T P_k + Q), \\ K_{k+1} = R^{-1} B^T P_k, \end{cases}$$

Stochastic ADP algorithm:

$$\begin{bmatrix} \text{vecs}(H_k) \\ \text{vec}(K_{k+1}) \end{bmatrix} = [I_{\xi\xi}, 2I_{\xi v}]^\dagger (\delta_{\xi\xi} - I_{d\xi\xi})\text{vec}(P_k)$$

$$P_{k+1} \leftarrow P_k + \epsilon_k (H_k + Q - K_{k+1}^T R K_{k+1})$$

Step size

If the step size is small, then

$$\lim_{k\to\infty} K_k = K^*, \ \lim_{k\to\infty} P_k = P^*.$$

Stochastic ADP algorithm

Continuous-time VI:

$$\begin{cases} P_{k+1} = P_k + \epsilon_k (A^T P_k + P_k A - P_k B R^{-1} B^T P_k + Q), \\ K_{k+1} = R^{-1} B^T P_k, \end{cases}$$

Stochastic ADP algorithm:

$$\begin{bmatrix} \mathrm{vecs}(H_k) \\ \mathrm{vec}(K_{k+1}) \end{bmatrix} = [I_{\xi\xi}, 2I_{\xi v}]^\dagger (\delta_{\xi\xi} - I_{d\xi\xi}) \mathrm{vec}(P_k)$$

$$P_{k+1} \leftarrow P_k + \epsilon_k (H_k + Q - K_{k+1}^T R K_{k+1})$$

Step size

If the step size is small, then

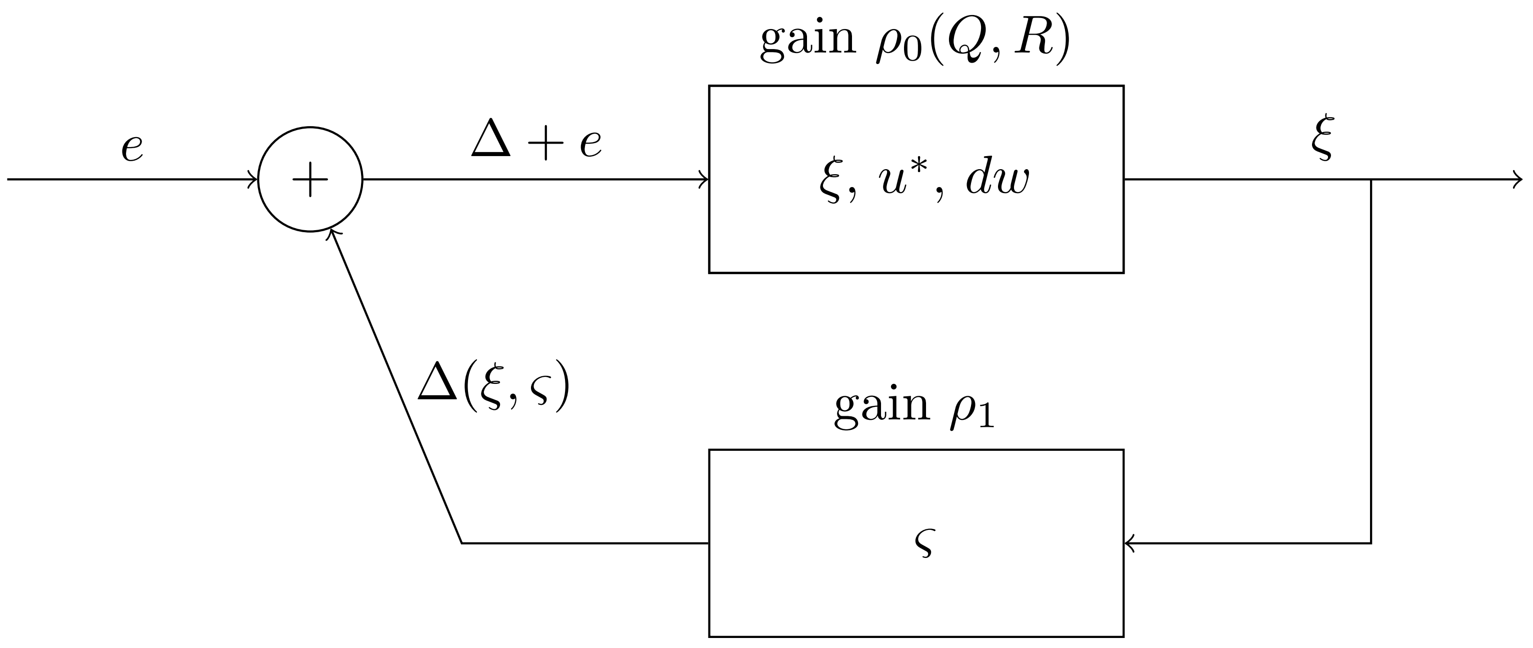
$$\lim_{k\to\infty} K_k = K^*, \ \lim_{k\to\infty} P_k = P^*.$$

How about robustness?

Small-gain theorem

$$d\xi = A\xi dt + B d\nu,$$
$$d\nu = (u + e + \Delta(\varsigma, \xi))dt + G(u + e + \Delta(\varsigma, \xi))dw,$$
$$d\varsigma = f(\varsigma, \xi)dt + g(\varsigma, \xi)d\eta.$$

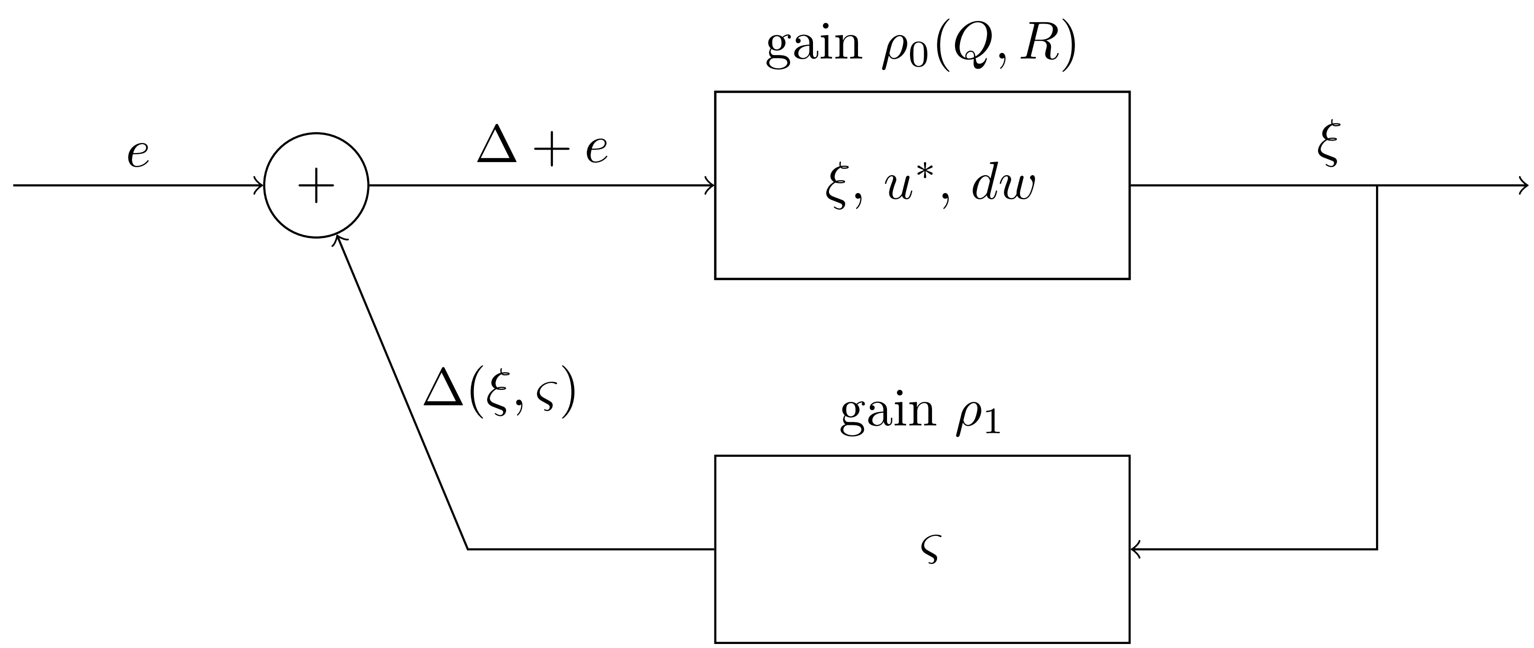
If the gain ρ_0 is small, then the interconnected system is stable.



Small-gain theorem

$$d\xi = A\xi dt + B d\nu,$$
$$d\nu = (u + e + \Delta(\varsigma, \xi))dt + G(u + e + \Delta(\varsigma, \xi))dw,$$
$$d\varsigma = f(\varsigma, \xi)dt + g(\varsigma, \xi)d\eta.$$

If the gain ρ_0 is small, then the interconnected system is stable.



Stochastic gain assignment:

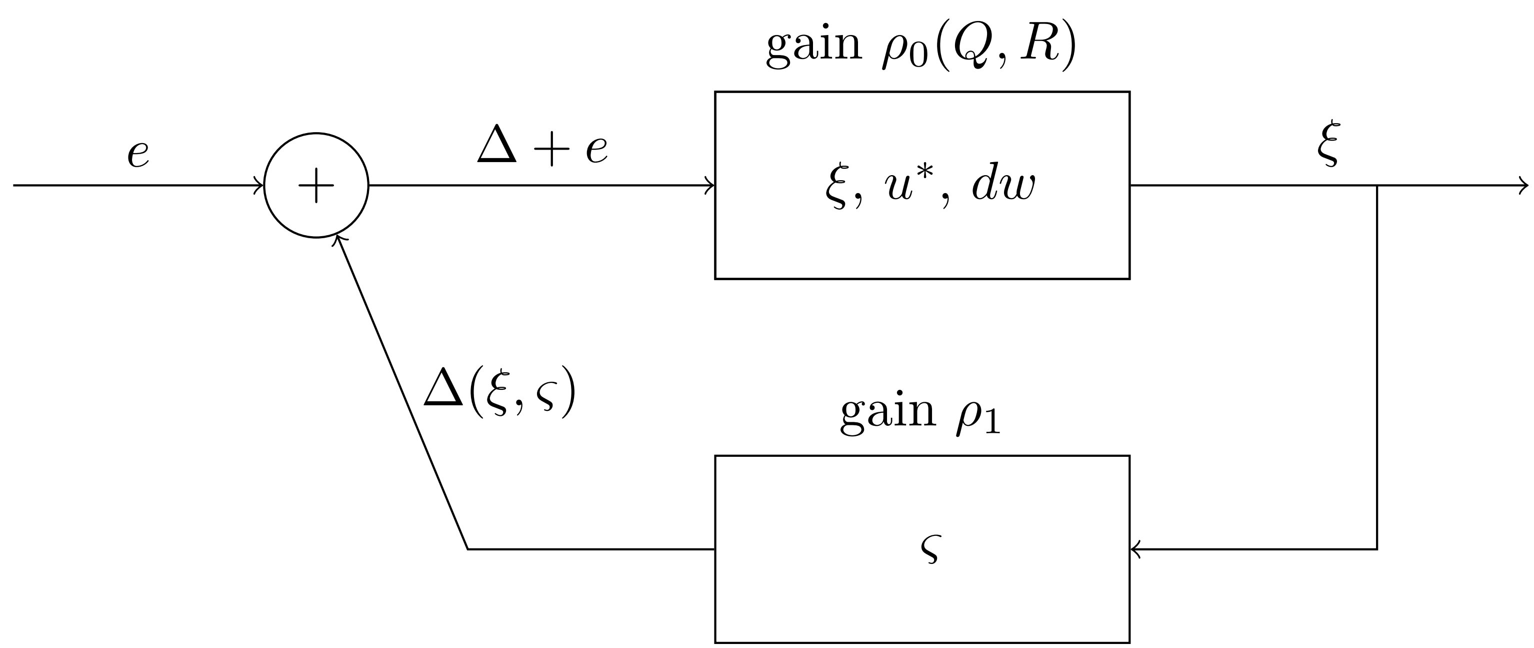
$$2\rho_1\alpha_1 I < Q$$

α_1 dependents on system parameters.

Small-gain theorem

$$d\xi = A\xi dt + B d\nu,$$
$$d\nu = (u + e + \Delta(\varsigma, \xi))dt + G(u + e + \Delta(\varsigma, \xi))dw,$$
$$d\varsigma = f(\varsigma, \xi)dt + g(\varsigma, \xi)d\eta.$$

If the gain ρ_0 is small, then the interconnected system is stable.



Stochastic gain assignment:

$$2\rho_1 \alpha_1 I < Q$$

α_1 dependents on system parameters.

Robustness is guaranteed in the presence of the suboptimal inference

Outline

- What is biological motor control?
- Recent developments in biological motor control
- Our contributions
- Simulation validation
- Conclusions and future work

Motor learning model

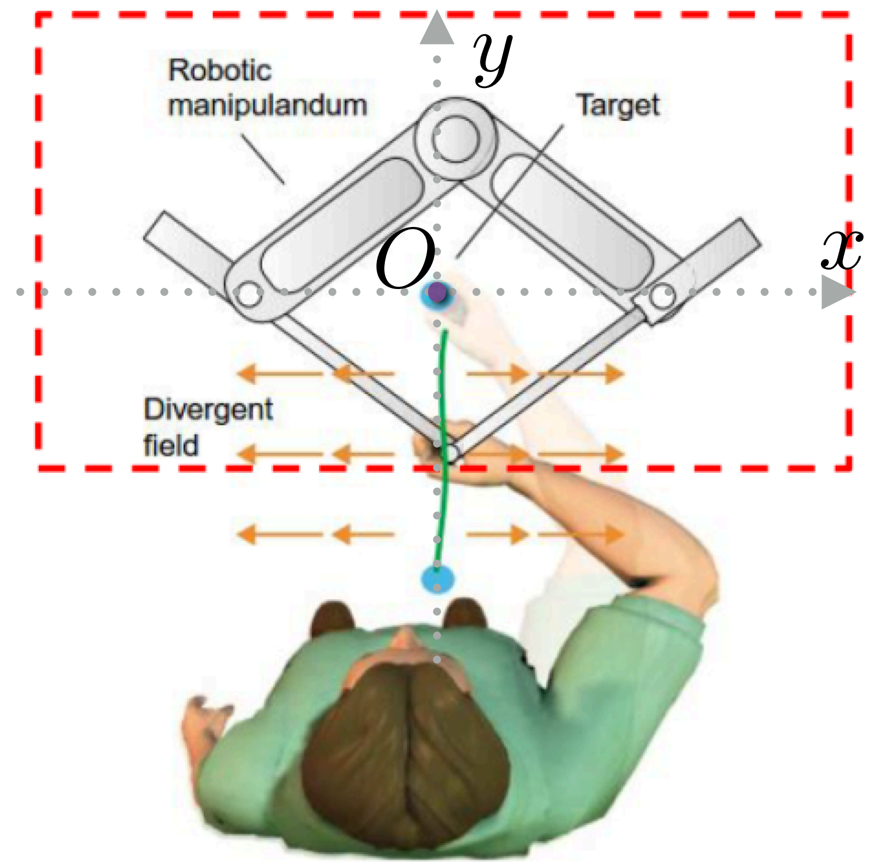
Consider the following model
(Todorov, 2005, Neural Computation):

$$dp = v dt,$$

$$m dv = (a - bv + f) dt,$$

$$\tau da = (u - a + e + \Delta(\varsigma, \xi)) dt + d\zeta,$$

$$T\dot{\varsigma} = -\varsigma + a - \gamma_1 \tanh(a), \quad \Delta = \gamma_0 \varsigma.$$



Divergent field (DF):

$$f = \beta p_x$$

Signal-dependent noise:

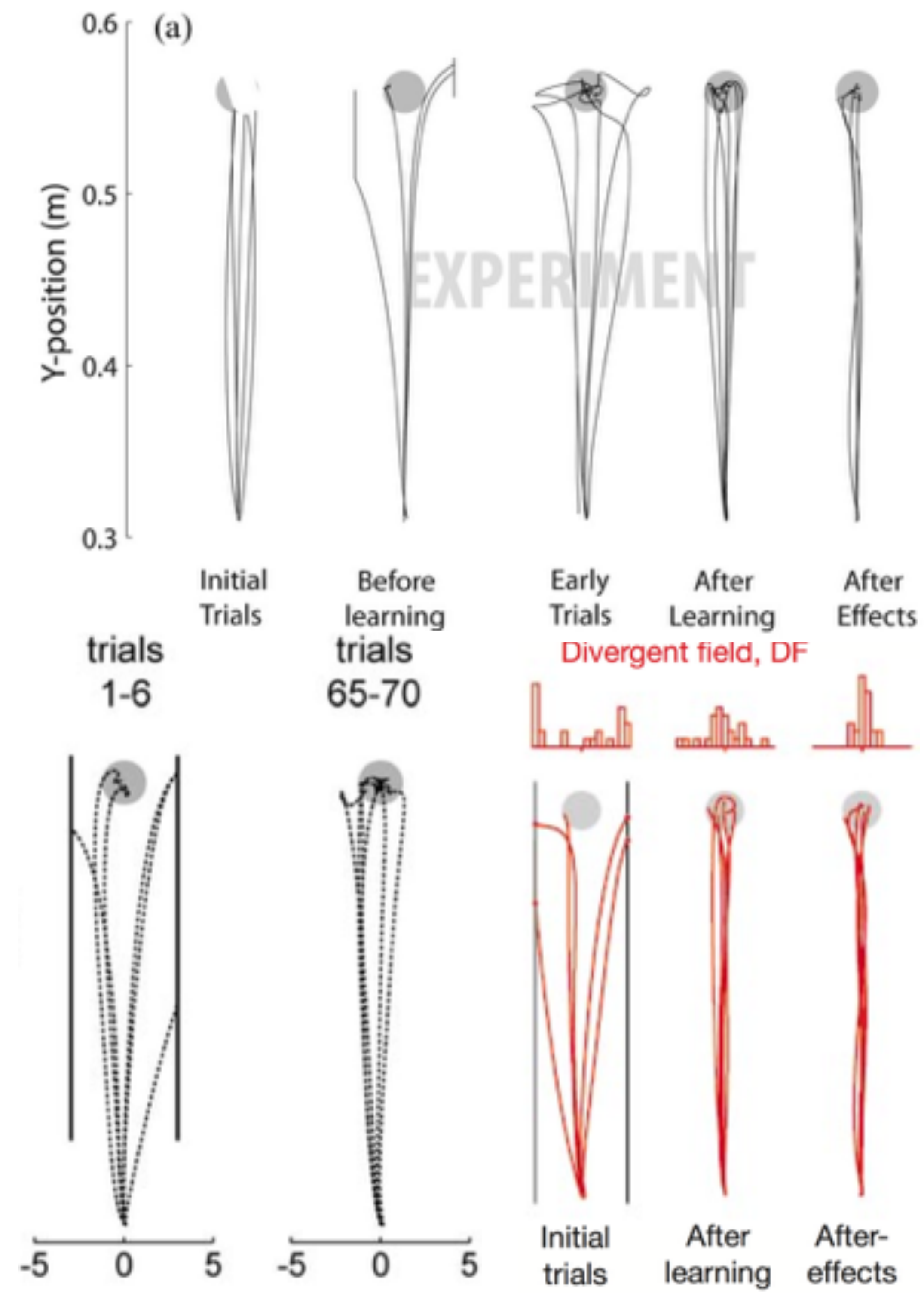
$$d\zeta = \begin{bmatrix} c_1 & 0 \\ c_2 & 0 \end{bmatrix} \begin{bmatrix} u_x \\ u_y \end{bmatrix} dw_1 + \begin{bmatrix} 0 & -c_2 \\ 0 & c_1 \end{bmatrix} \begin{bmatrix} u_x \\ u_y \end{bmatrix} dw_2$$

Parameter	Description	Value
m	Hand mass	1.3kg
b	Viscosity	10Ns/m
tau	Time constant	0.05s
c1	Noise magnitude	0.075
c2	Noise magnitude	0.025
beta	Force magnitude	150

Validation — learning in DF

Experiment data:

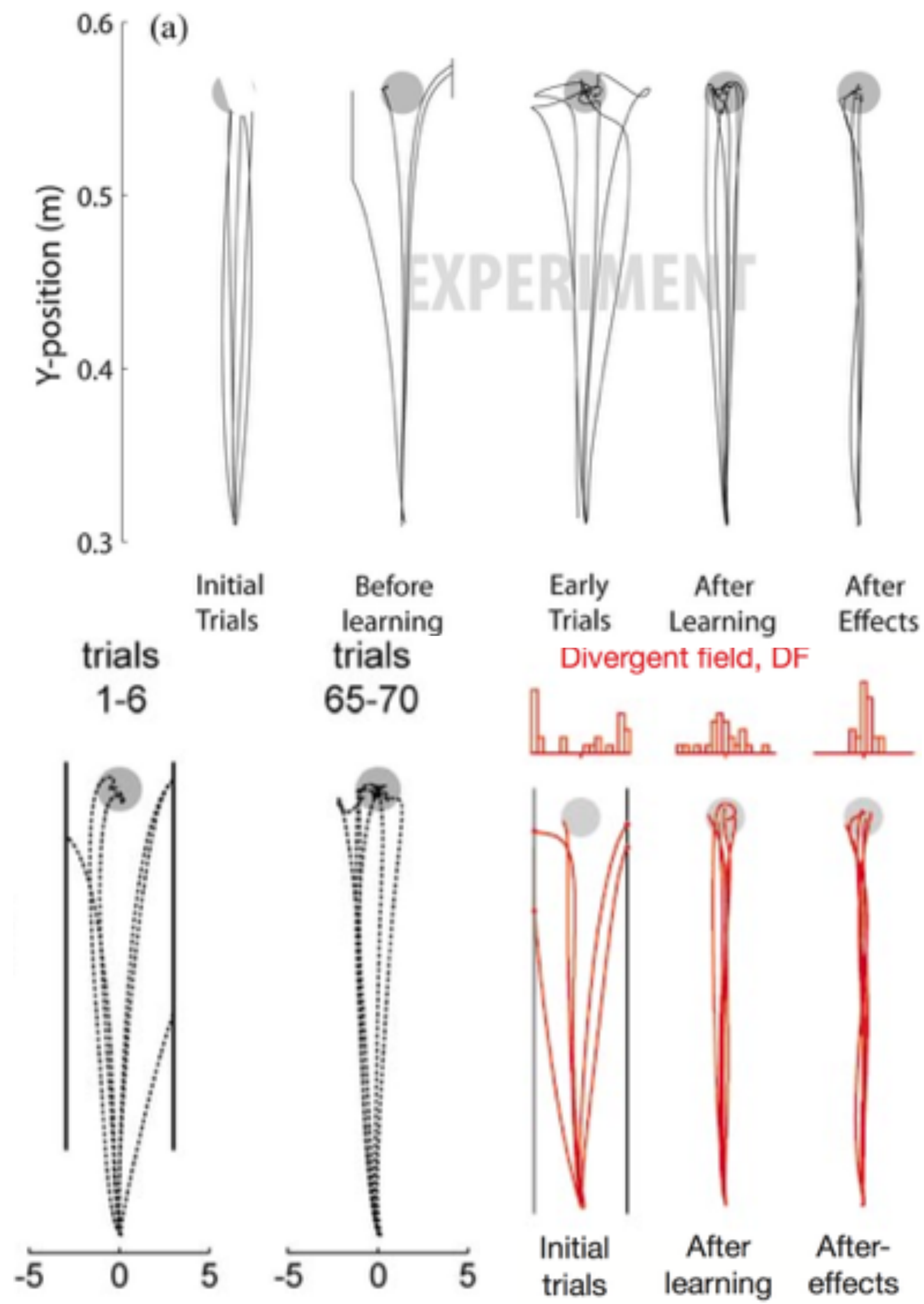
Burdet, et.al., 2001, Nature;
Franklin, et.al., 2003, Experimental Brain Research;
Zhou, et.al., 2012, IEEE Transactions on Biomedical Engineering



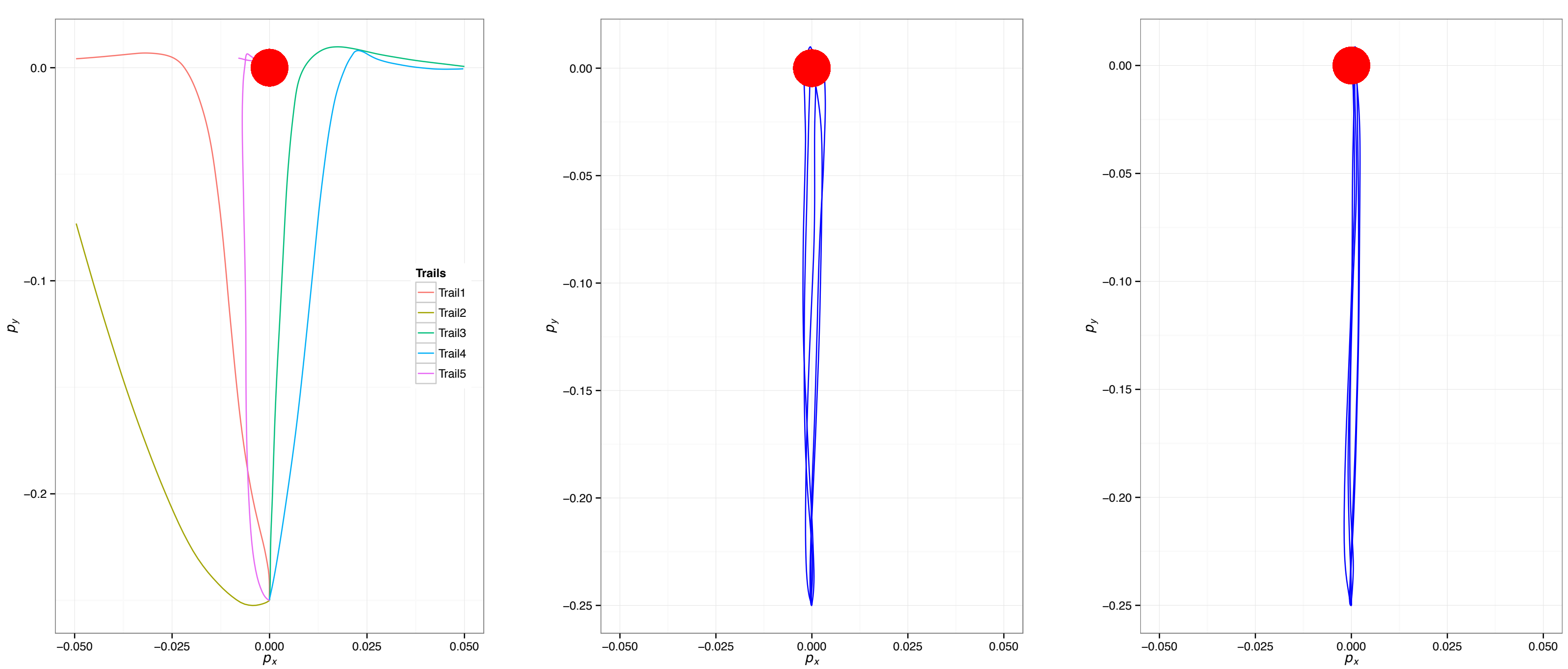
Validation — learning in DF

Experiment data:

Burdet, et.al., 2001, Nature;
Franklin, et.al., 2003, Experimental Brain Research;
Zhou, et.al., 2012, IEEE Transactions on Biomedical Engineering



Simulation from our theory



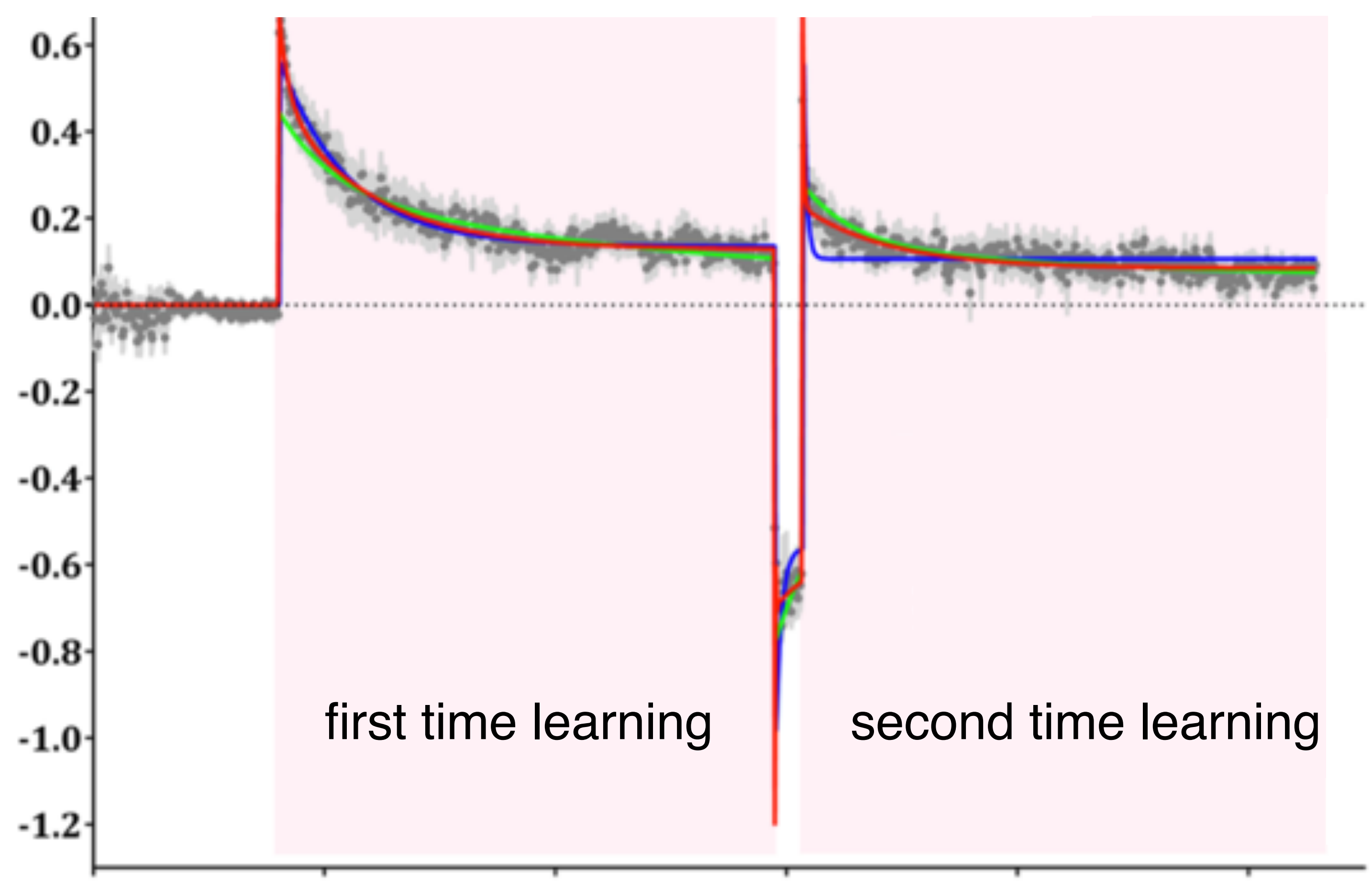
(a) First five trials in the DF.

(b) Five independent trials after ADP learning in the DF.

(c) Five independent trials after ADP learning in the NF.

Validation — savings

Savings: Prior learning to speed subsequent relearning even after washout.



(Mawase, 2014)

Validation — savings

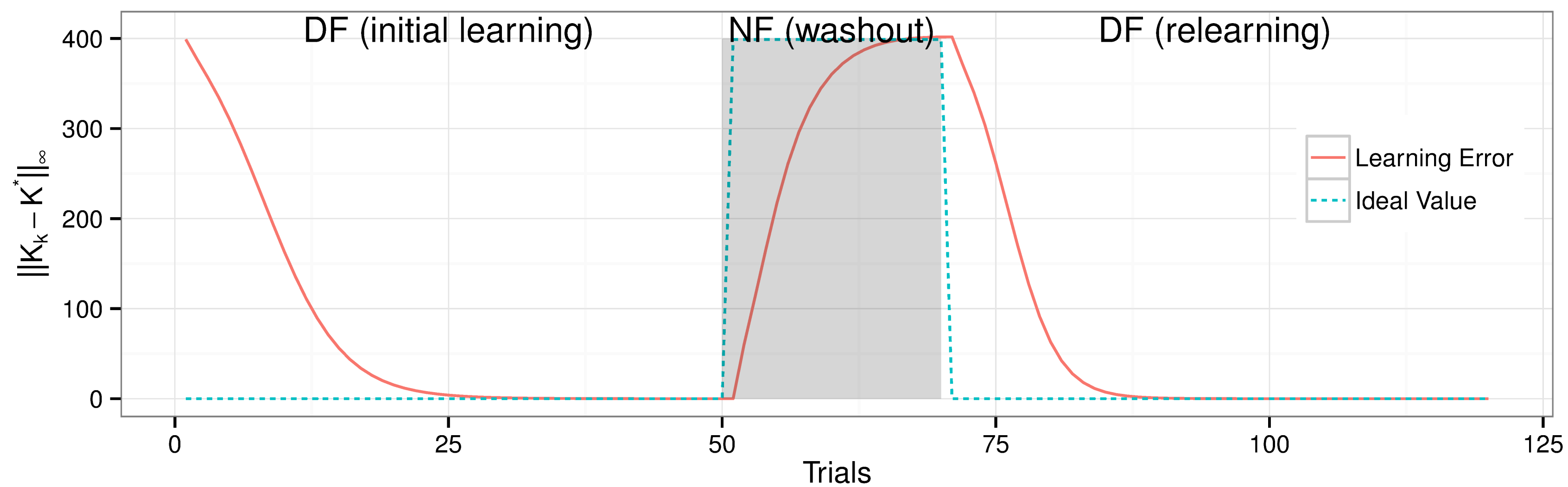
Two theories:

Validation — savings

Two theories:

In the relearning phase, the recall of the motor memory is accelerated (larger learning rate (Zarahn, et.al., 2008, Journal of Neurophysiology)).

Simulation from our theory



Accelerated recall of the reinforced action

Validation — savings

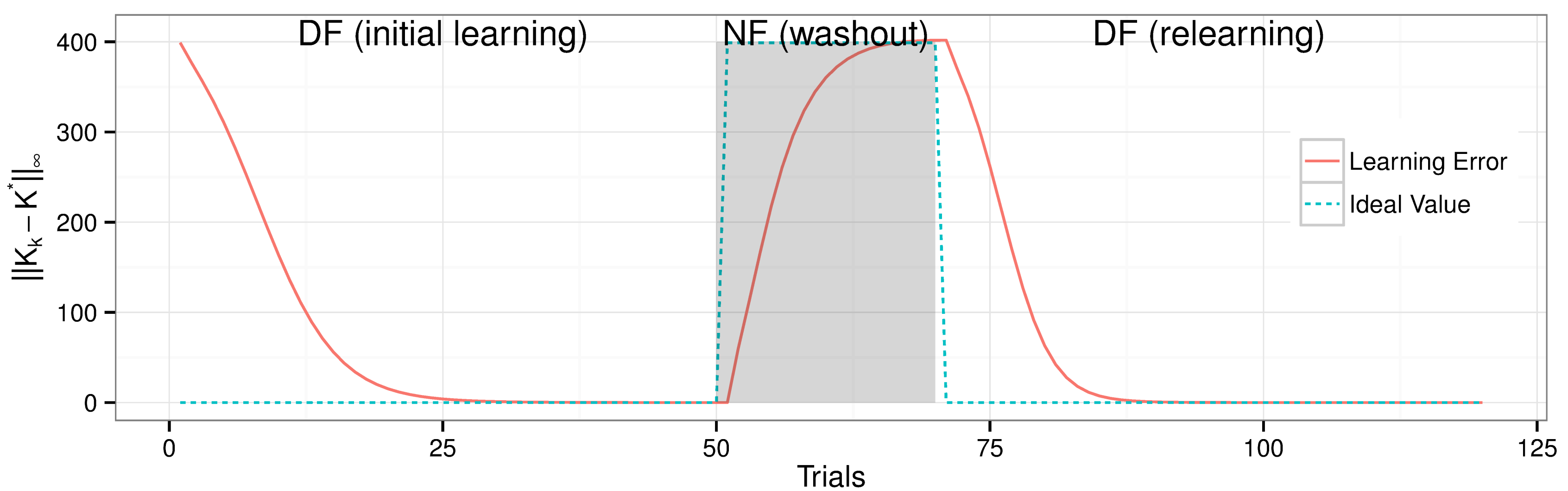
Two theories:

In the relearning phase, the recall of the motor memory is accelerated (larger learning rate (Zarahn, et.al., 2008, Journal of Neurophysiology)).

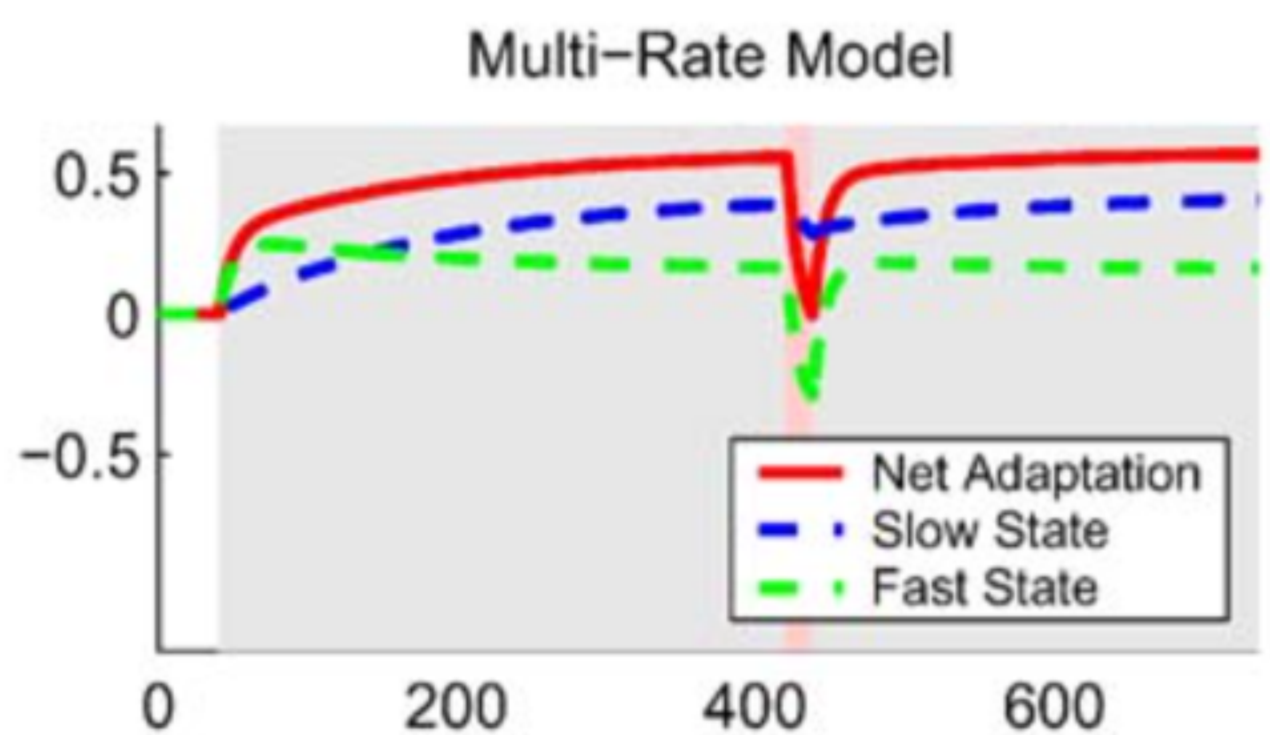
Savings is due to a multi-rate model (the fast state and the slow state) (Smith, et.al., 2006, PLoS Biology).

Model-free learning is related to the slow state model.

Simulation from our theory



n

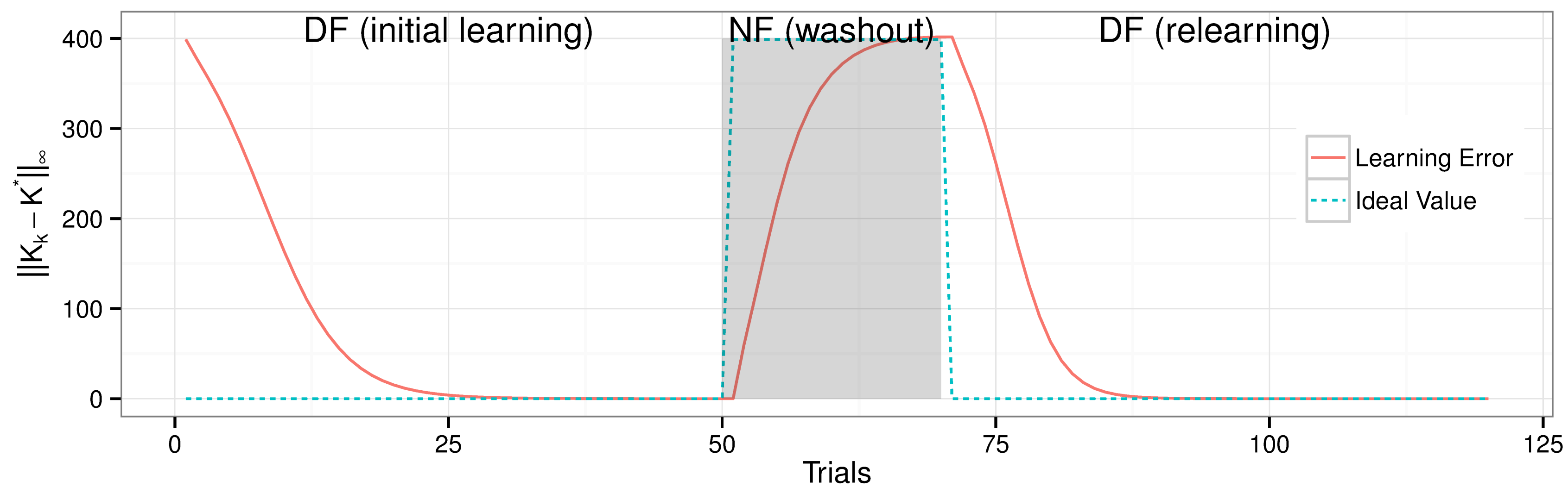


Validation — savings

Two theories:

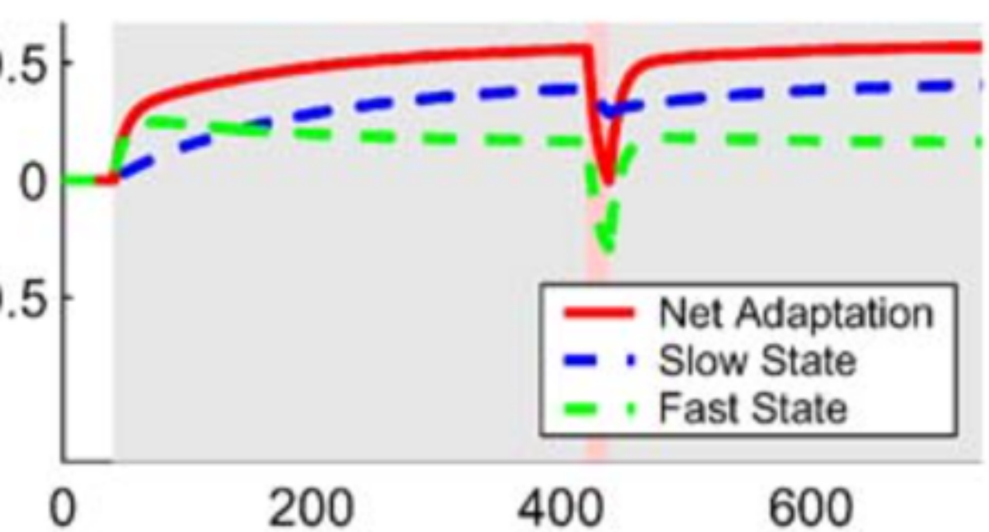
In the relearning phase, the recall of the motor memory is accelerated (larger learning rate (Zarahn, et.al., 2008, Journal of Neurophysiology)).

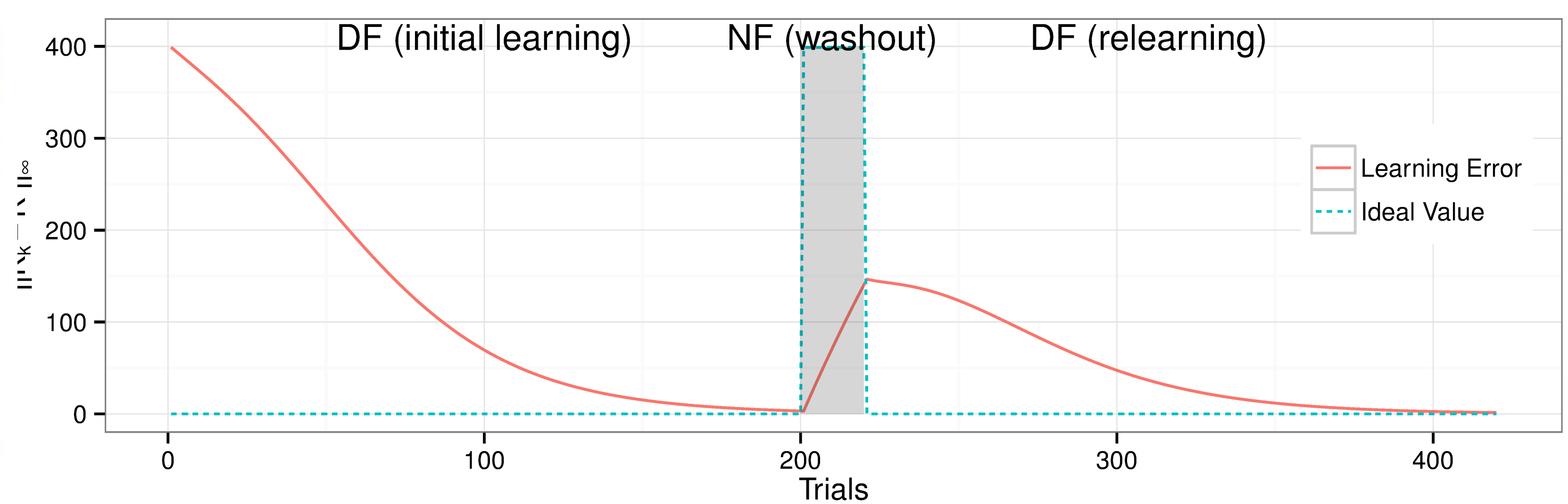
Simulation from our theory



Accelerated recall of the reinforced action

Multi−Rate Model



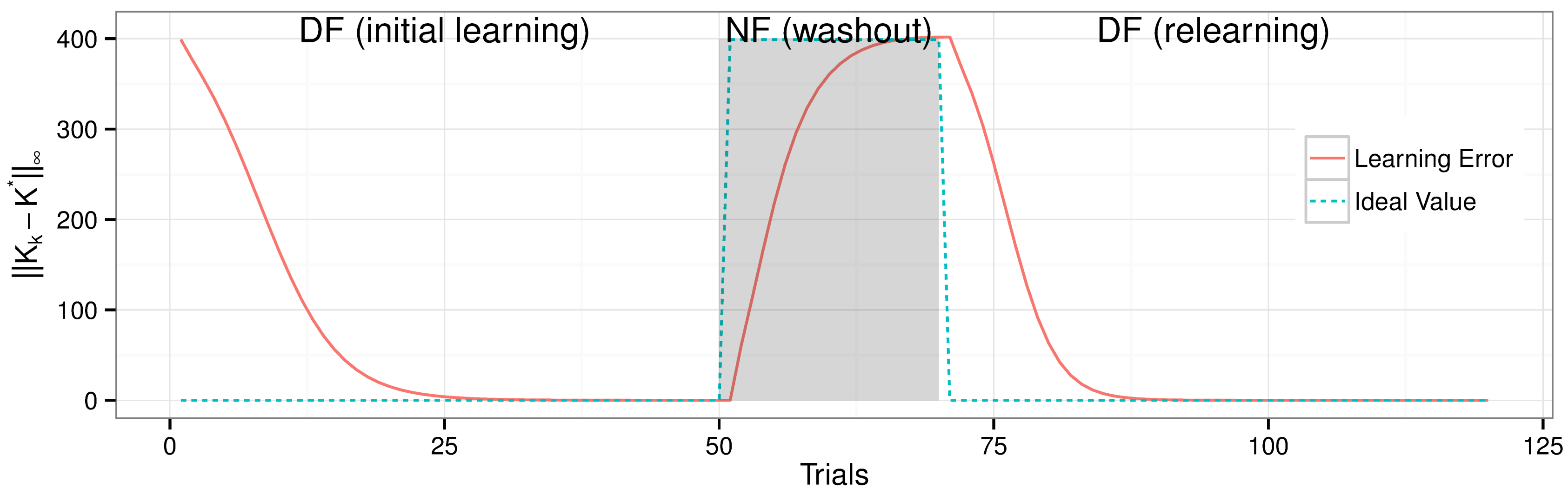


Validation — savings

Two theories:

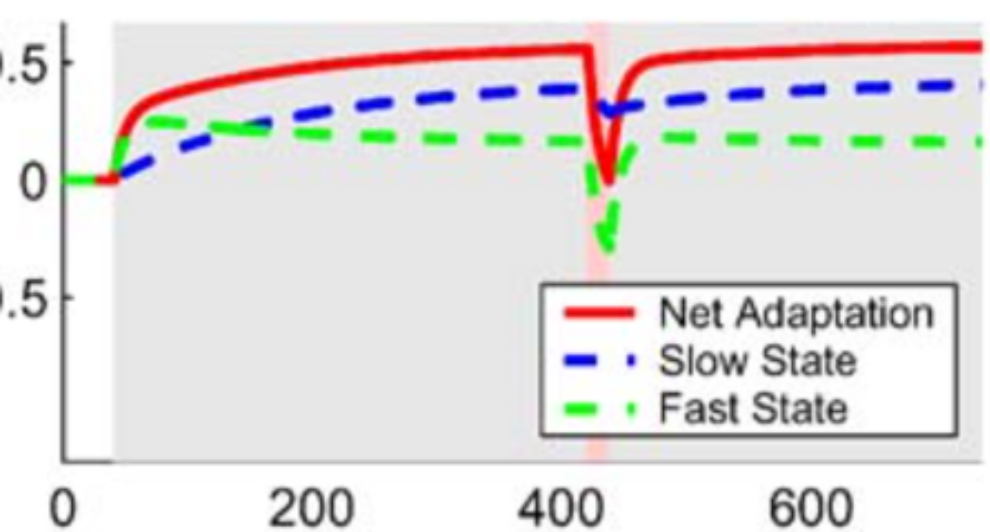
In the relearning phase, the recall of the motor memory is accelerated (larger learning rate (Zarahn, et.al., 2008, Journal of Neurophysiology)).

Simulation from our theory

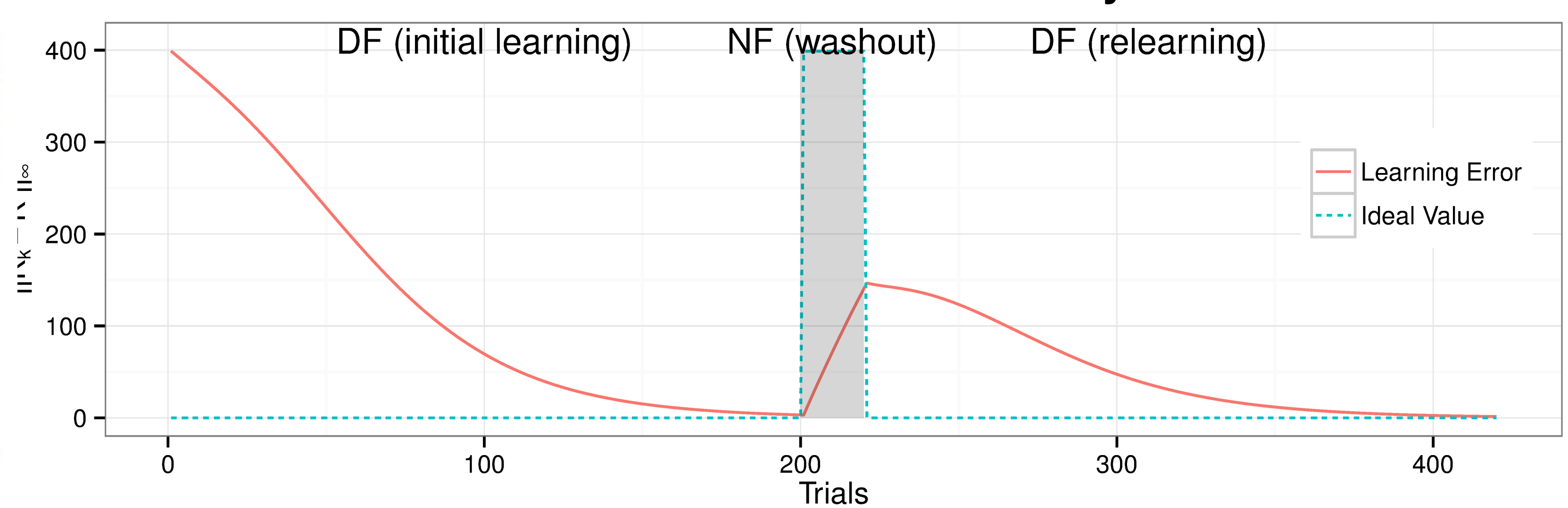


Accelerated recall of the reinforced action

Multi−Rate Model



Simulation from our theory



Slow rate model

Outline

- What is biological motor control?
- Recent developments in biological motor control
- Our contributions
- Simulation validation
- Conclusions and future work

Conclusions and future work

Conclusions and future work

Conclusions:

- We have developed a new theory of biological motor control based on ADP
- This theory can explain both old and new experimental phenomena
- Simulations are consistent with existing experimental results
- On the level of control theory, we have proposed a new theory of stochastic ADP

Conclusions and future work

Conclusions:

- We have developed a new theory of biological motor control based on ADP
- This theory can explain both old and new experimental phenomena
- Simulations are consistent with existing experimental results
- On the level of control theory, we have proposed a new theory of stochastic ADP

Future work

- We will conduct experiments with the University of Cambridge to fully extend the proposed theory to other biological problems.

Thank you!