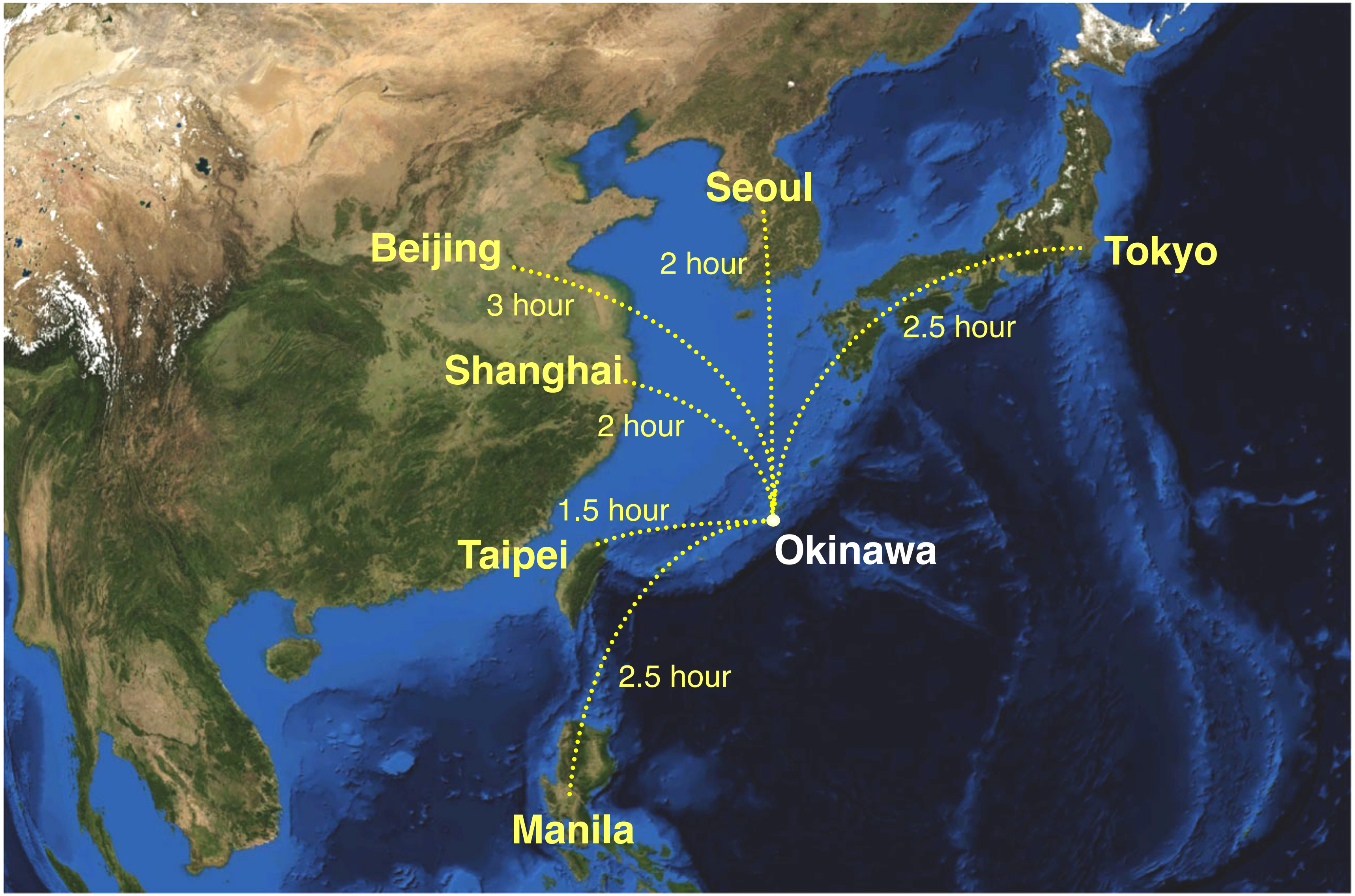
ISSA Summer School 2017 at CiNet
2017.5.30

Neural Circuit for Mental Simulation

Kenji Doya
doya@oist.jp
Neural Computation Unit
Okinawa Institute of Science and Technology

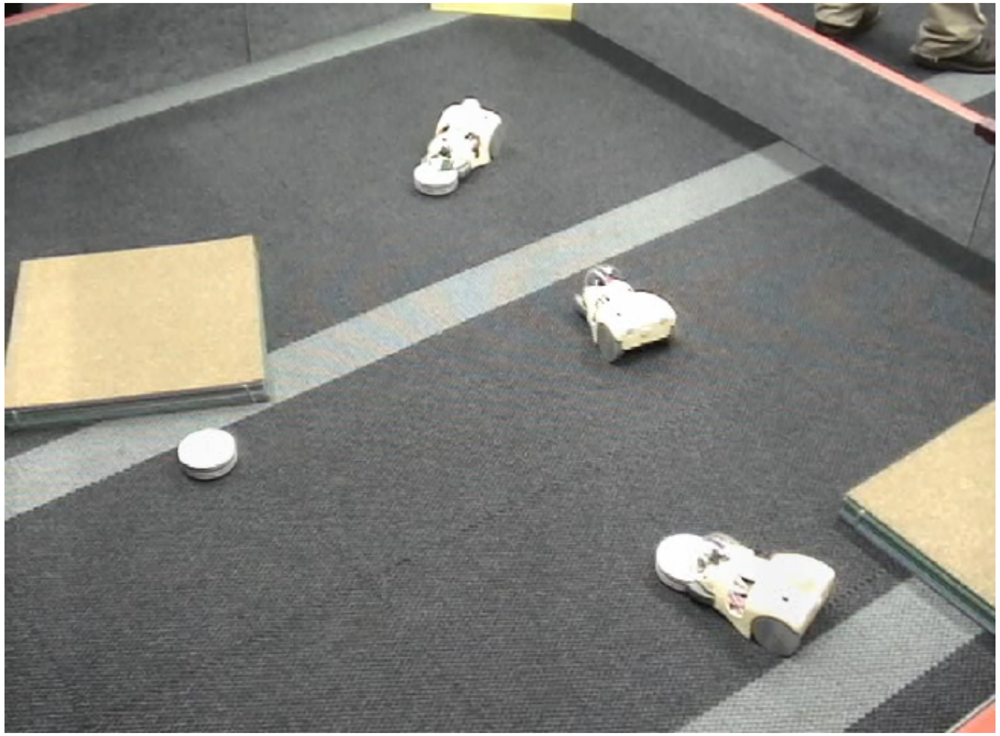
Location of Okinawa



OIST Neural Computation Unit

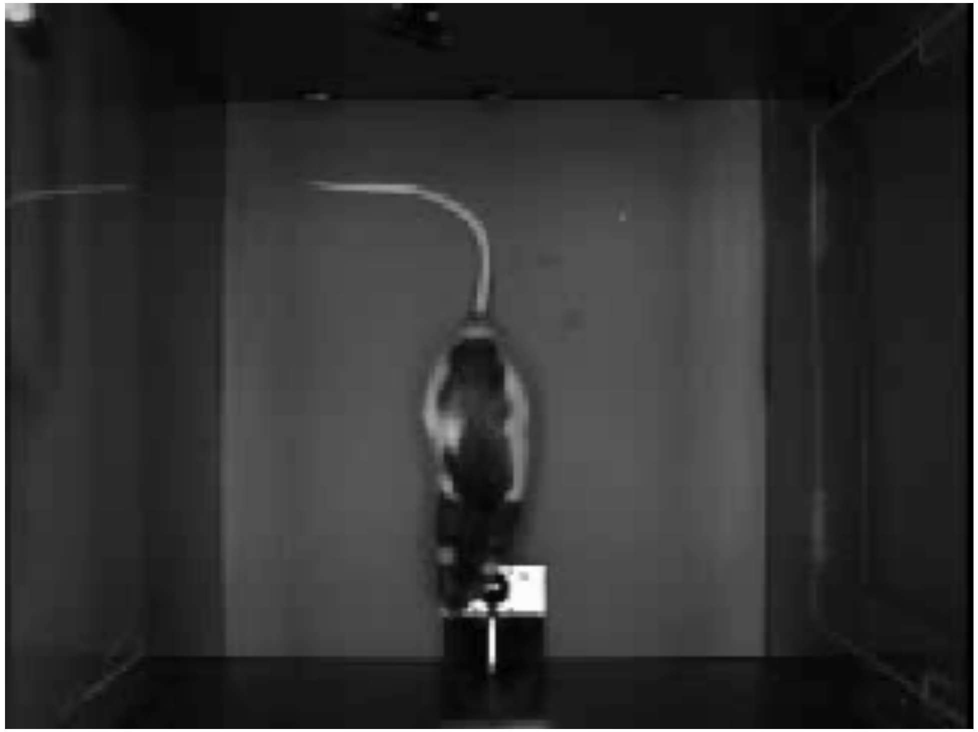
How to build adaptive, autonomous systems

- robot experiments



How the brain realizes robust, flexible adaptation

- neurobiology



Outline

Reinforcement Learning

- Can robots create their own reward function?
- Value function and basal ganglia

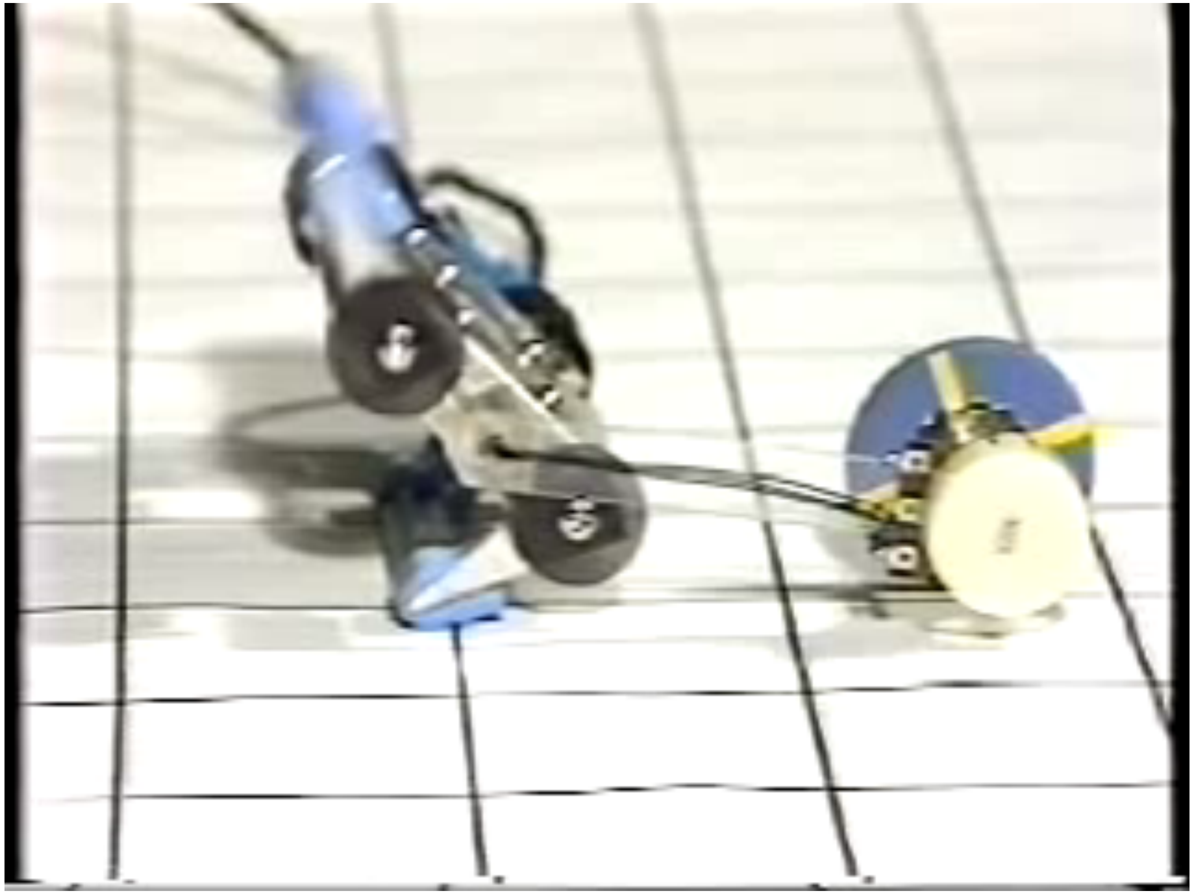
Mental Simulation

- Model-based action planning
- Dynamic Bayesian inference
- Patience, confidence and serotonin

Learning to Walk

(Doya & Nakano, 1985)

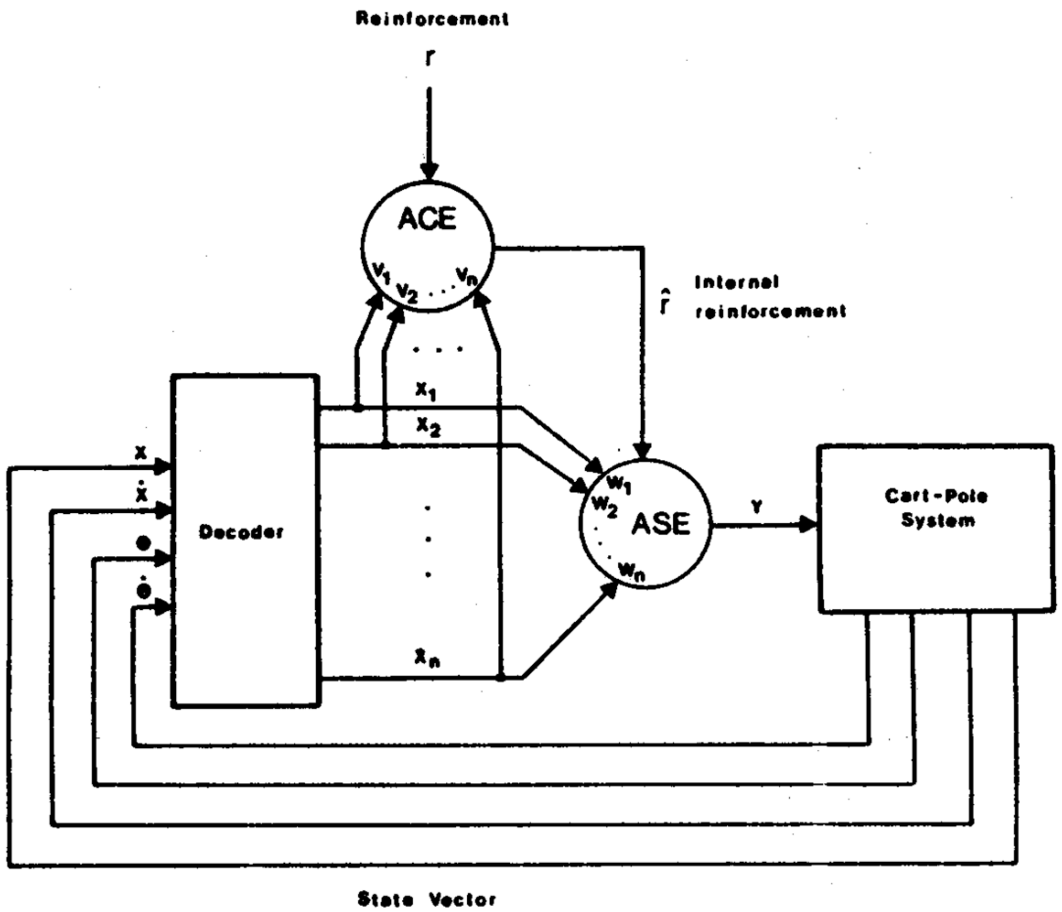
- Explore actions (cycle of 4 postures)
- Learn from performance feedback (speed sensor)

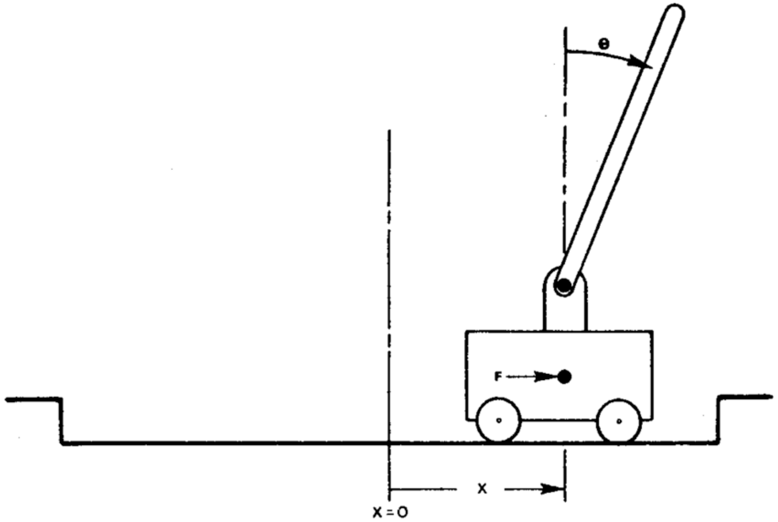


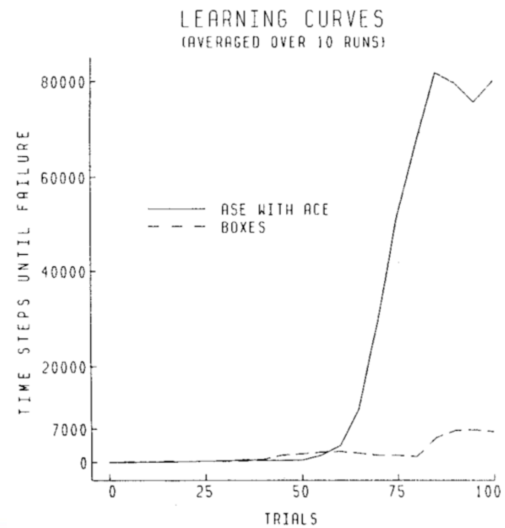
Neuronlike Adaptive Elements That Can Solve Difficult Learning Control Problems

ANDREW G. BARTO, MEMBER, IEEE, RICHARD S. SUTTON, AND CHARLES W. ANDERSON

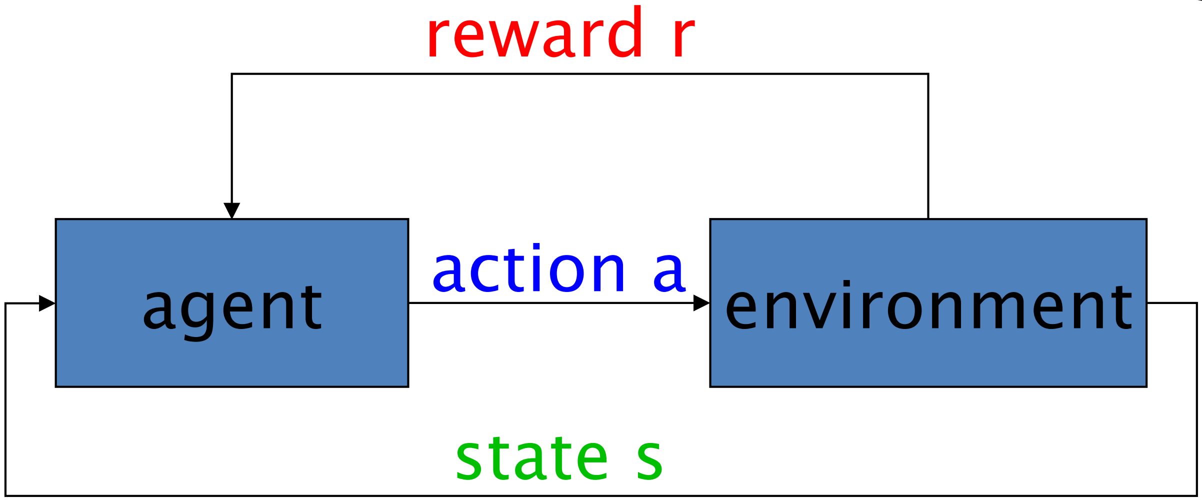
(1983)







Reinforcement Learning

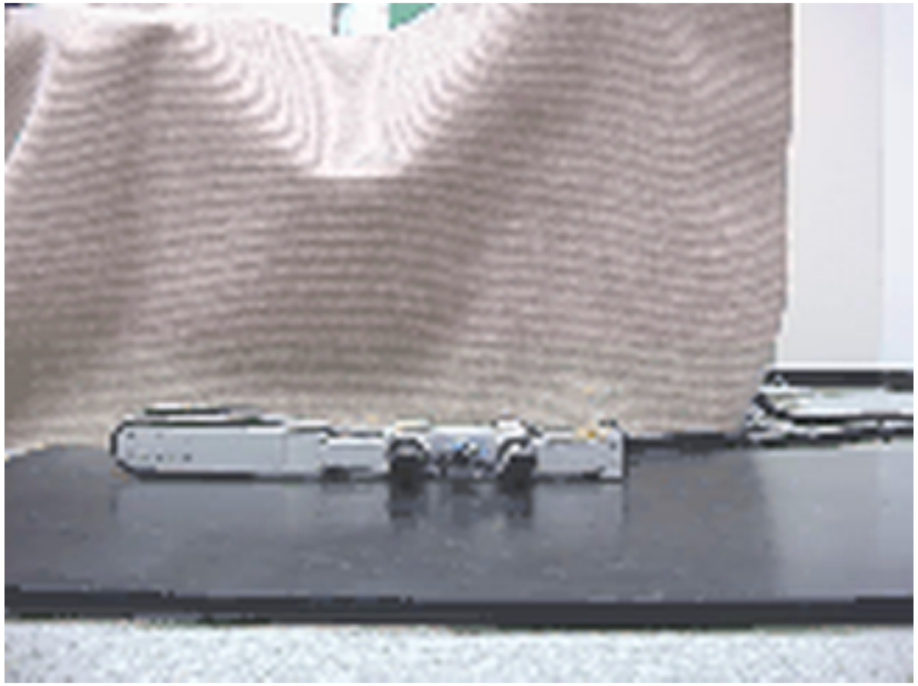


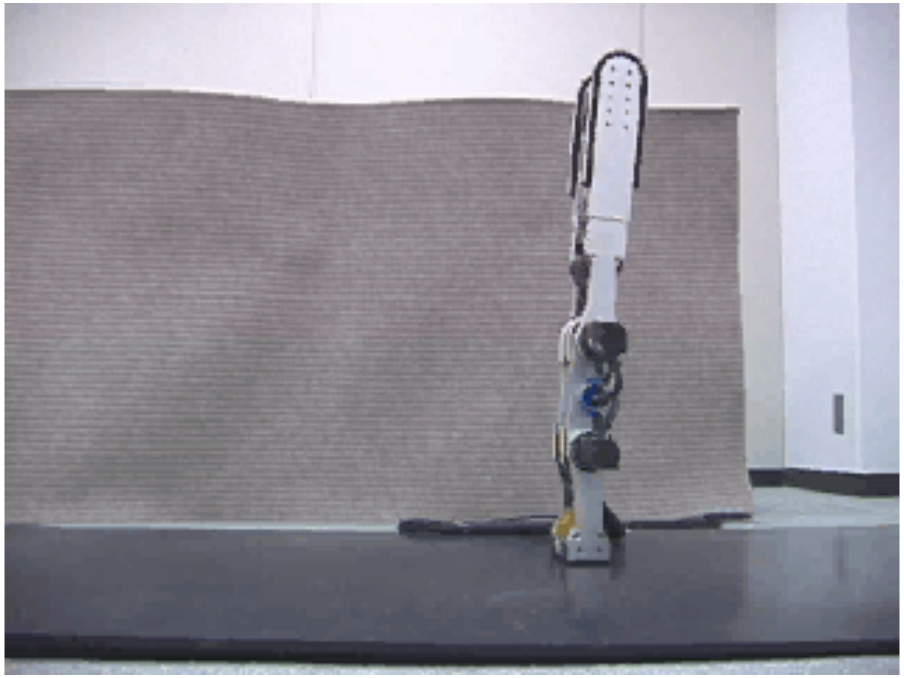
- Learn action policy: $s \rightarrow a$ to maximize rewards
- Value function: expected future rewards
  - $V(s(t)) = E[\, r(t) + \underbrace{\gamma r(t+1) + \gamma^2 r(t+2) + \gamma^3 r(t+3) + \ldots}_{\gamma V(s(t+1))}]$

    $0 \le \gamma \le 1$: discount factor
- Temporal difference (TD) error:
  - $\delta(t) = r(t) + \gamma V(s(t+1)) - V(s(t))$

Reinforcement Learning

(Morimoto & Doya, 2000)





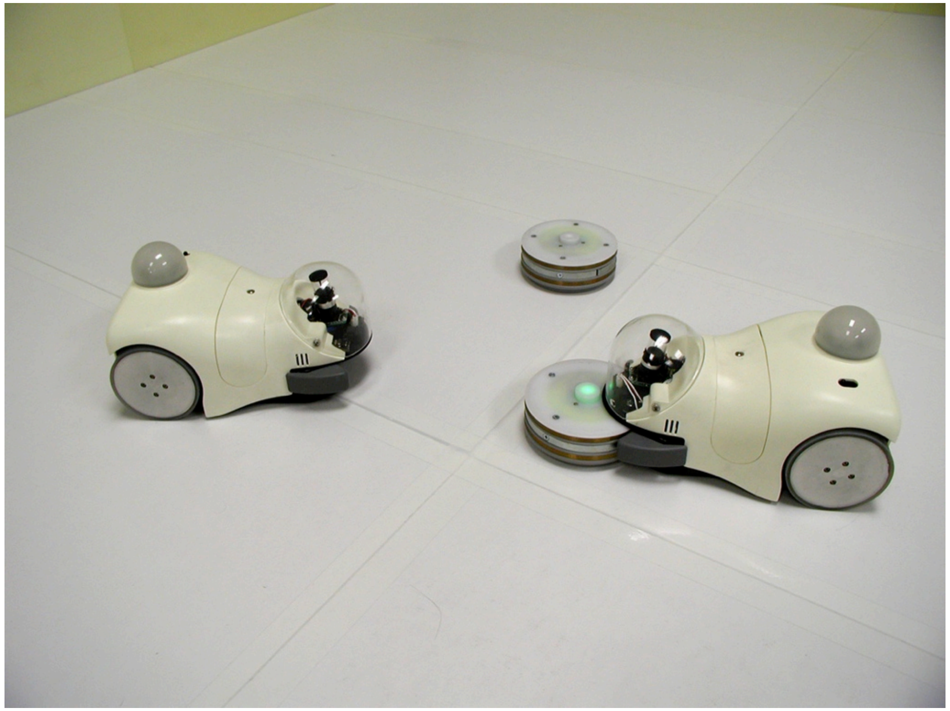
- Learning from reward and punishment
  - reward: height of the head
  - punishment: bump on the floor

Cyber Rodent Project (Doya & Uchibe, 2005)

What is the origin of rewards?

Robots with same constraint as biological agents

- Self-preservation
  - capture batteries
- Self-reproduction
  - exchange programs through IR ports

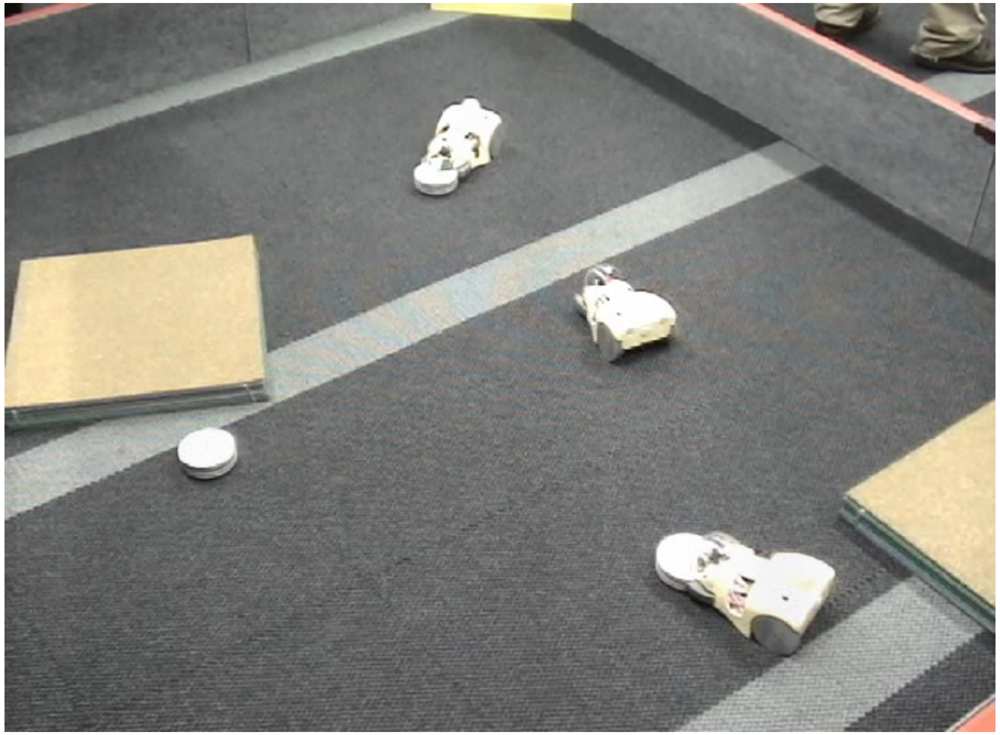


Vision of Cyber Rodents

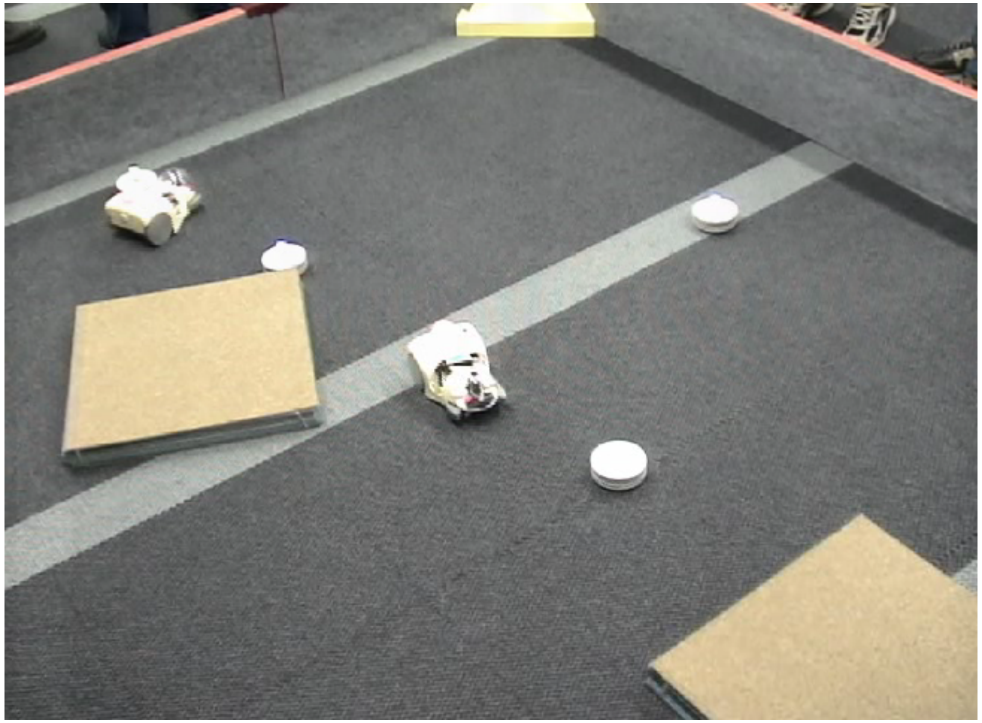
- Robot eye view

Learning to Survive and Reproduce

- Catch battery packs
  - survival



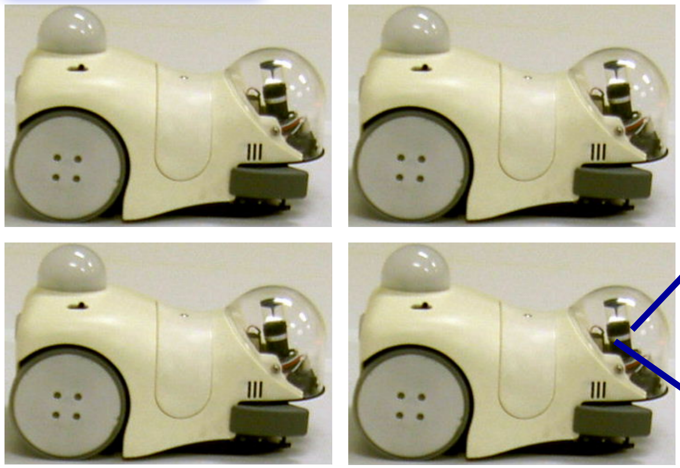
- Copy ‘genes’ by IR ports
  - reproduction, evolution



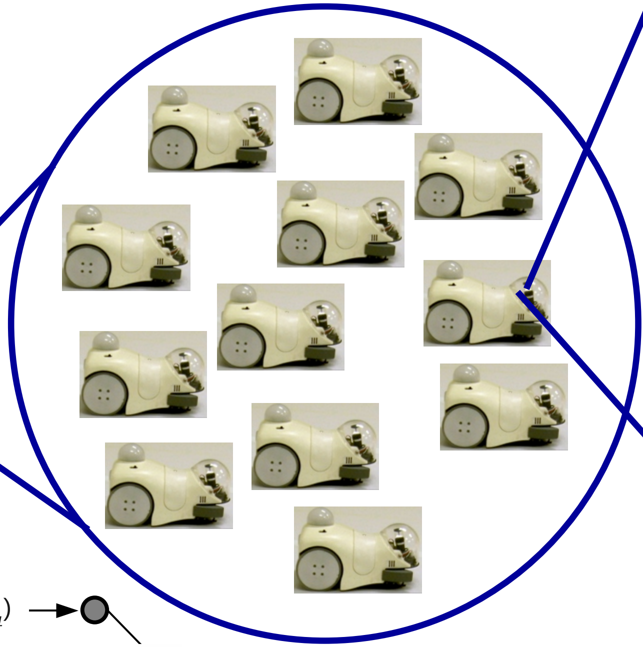
Embodied Evolution (Elfwing et al., 2011)

Population

Robots



Virtual agents
15-25



Genes

Weights for top layer NN

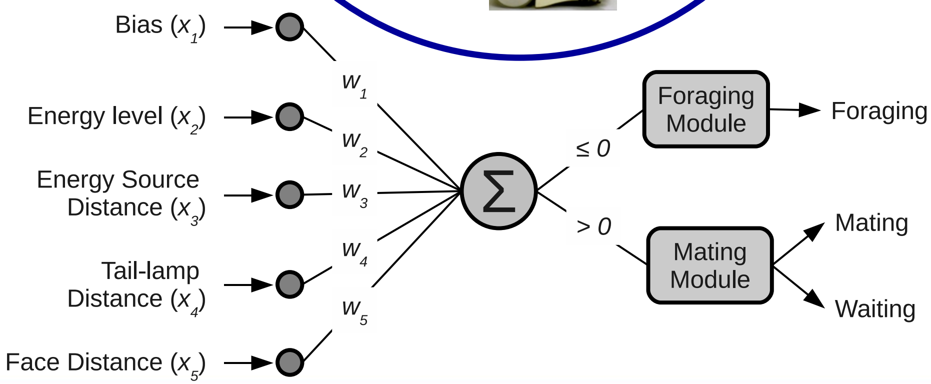
$w_1, w_2, \ldots, w_n$

Weights shaping rewards

$v_1, v_2, \ldots, v_n$

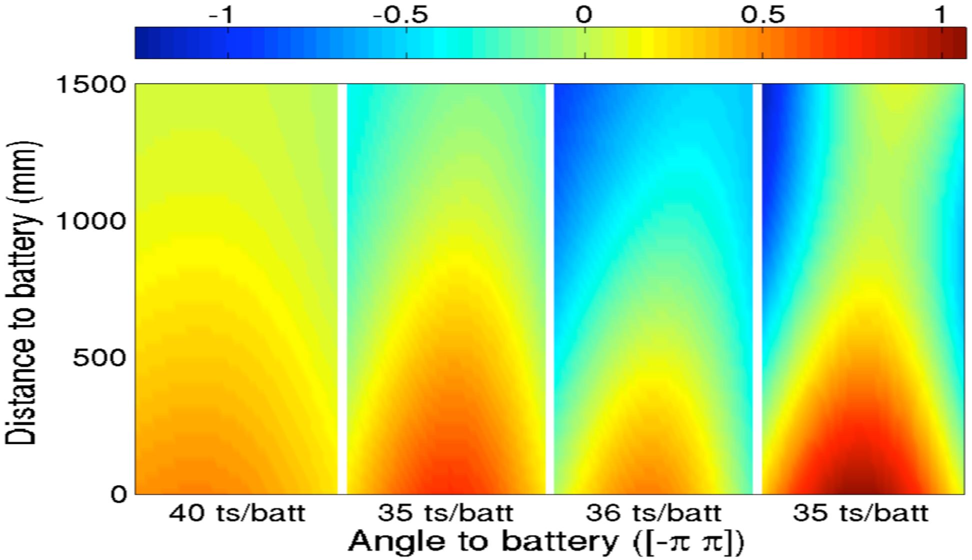
Meta-parameters

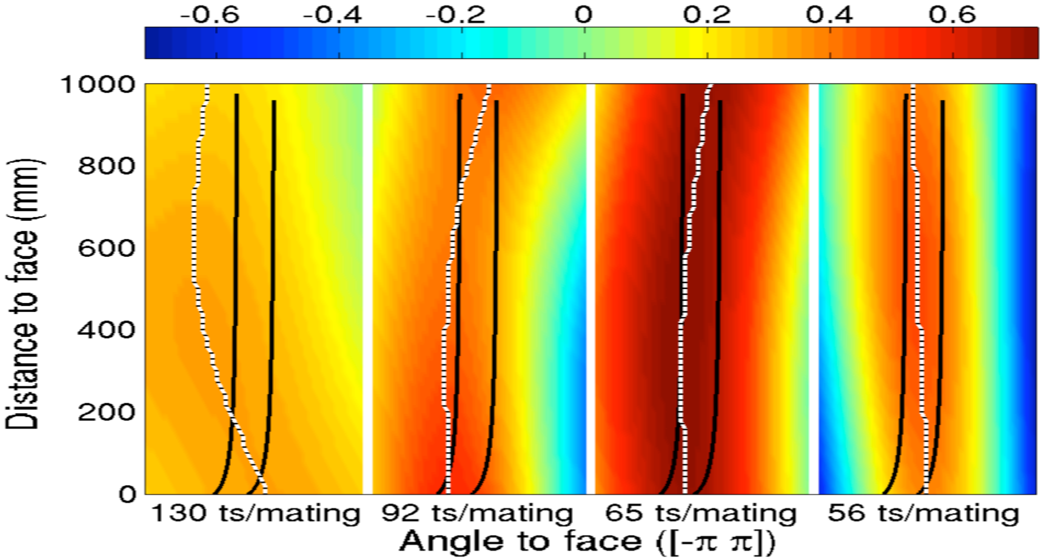
$\alpha\gamma\lambda\tau_k\tau_0$



Evolution of Shaping Rewards

- Vision of battery
- Vision of face

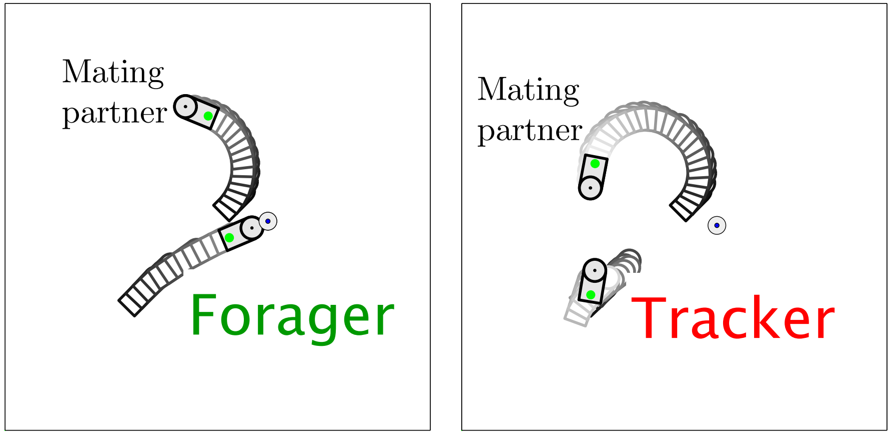




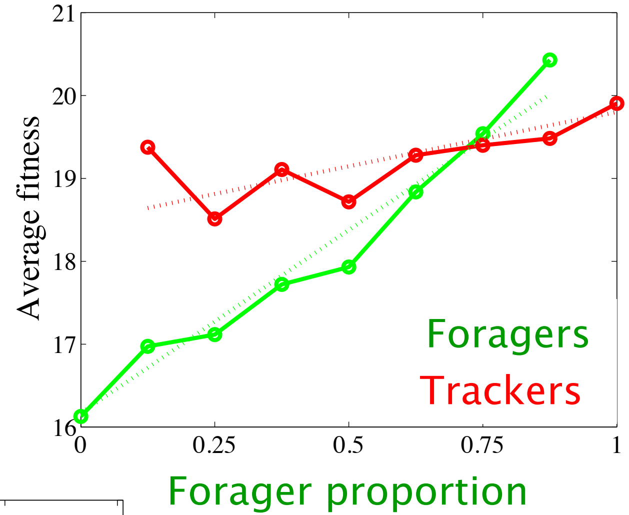
Polymorphism within Colony

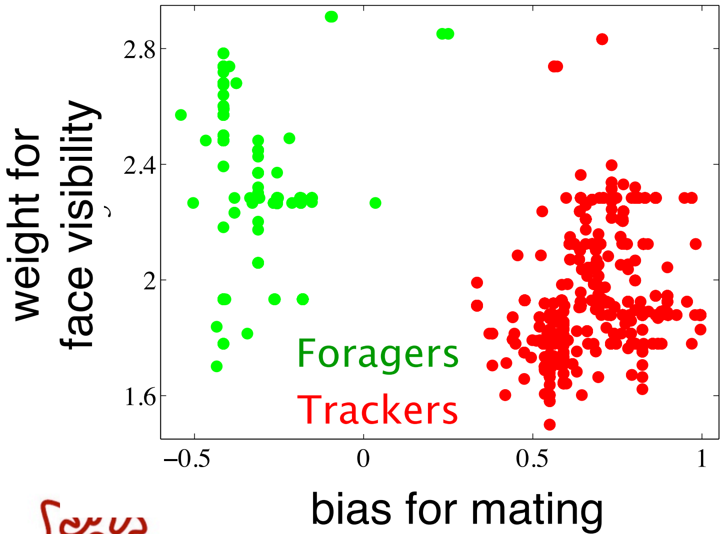
(Elfwing et al. 2014)

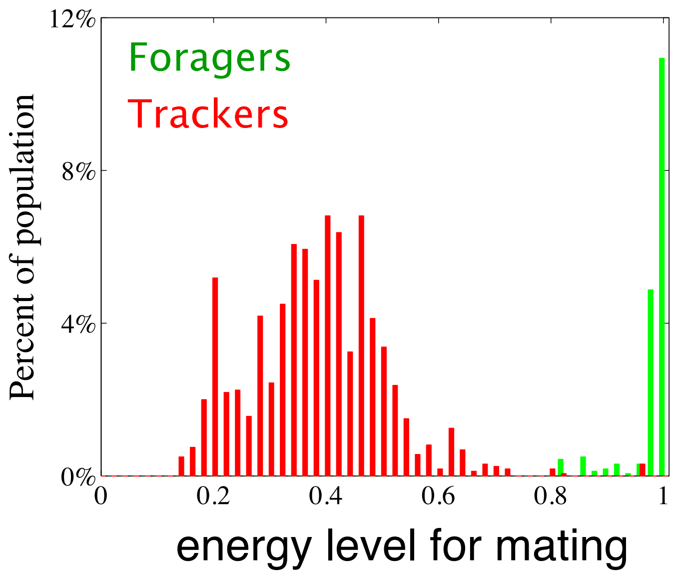
- Foragers and Trackers



- Evolutionaly stability







Reinforcement Learning

- Predict reward: *value function*
  - $V(s) = E[\, r(t) + \gamma r(t+1) + \gamma^2 r(t+2)\ldots \mid s(t)=s]$
  - $Q(s,a) = E[\, r(t) + \gamma r(t+1) + \gamma^2 r(t+2)\ldots \mid s(t)=s,\, a(t)=a]$
- Select action

How to implement these steps?

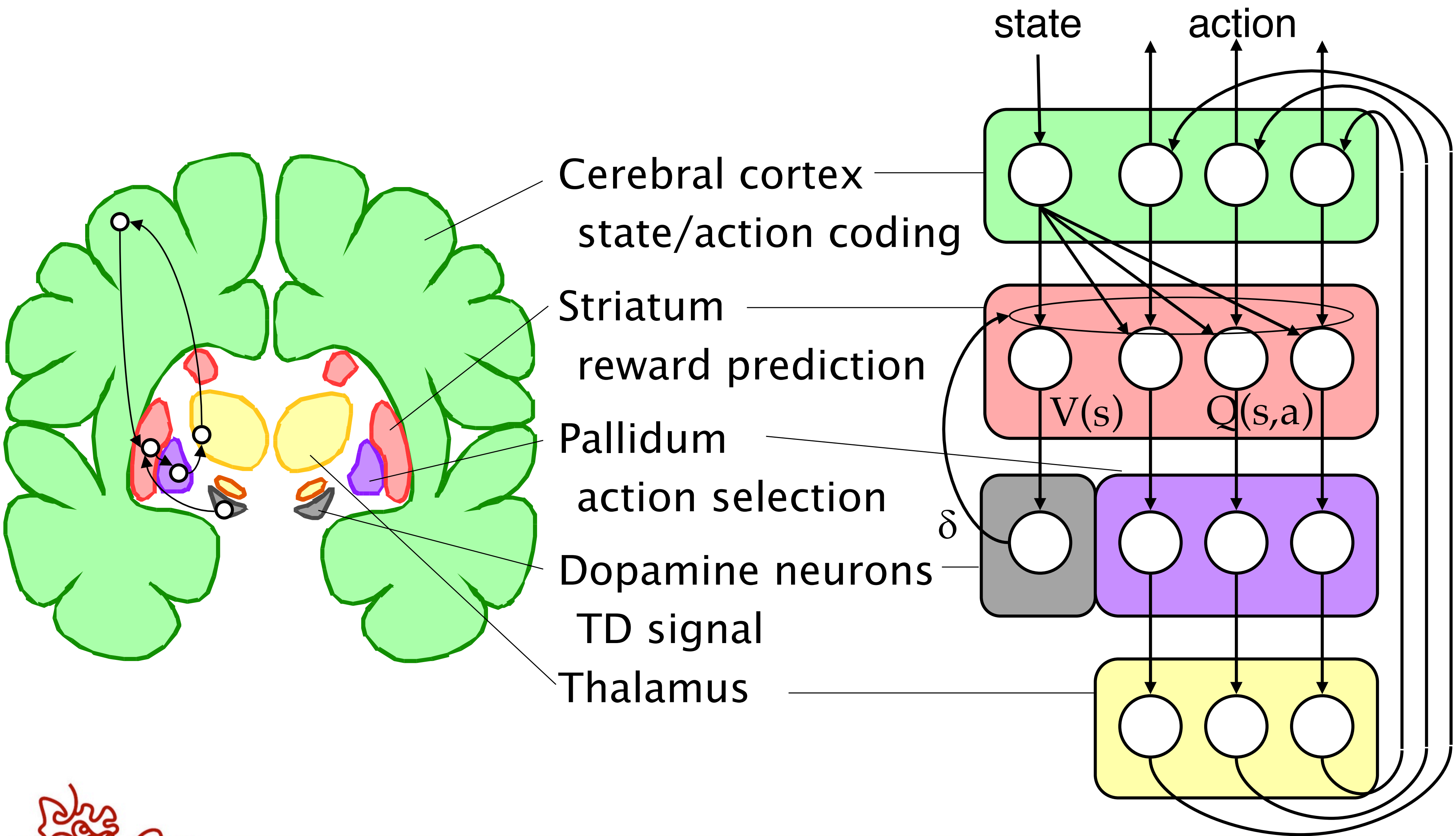
  - *greedy*: $a = \text{argmax } Q(s,a)$
  - *Boltzmann*: $P(a|s) \propto \exp[\, \beta\, Q(s,a)]$
- Update prediction: *TD error*
  - $\delta(t) = r(t) + \gamma V(s(t+1)) - V(s(t))$
  - $\Delta V(s(t)) = \alpha\, \delta(t)$

How to tune these parameters?

  - $\Delta Q(s(t),a(t)) = \alpha\, \delta(t)$

Basal Ganglia for Reinforcement Learning?

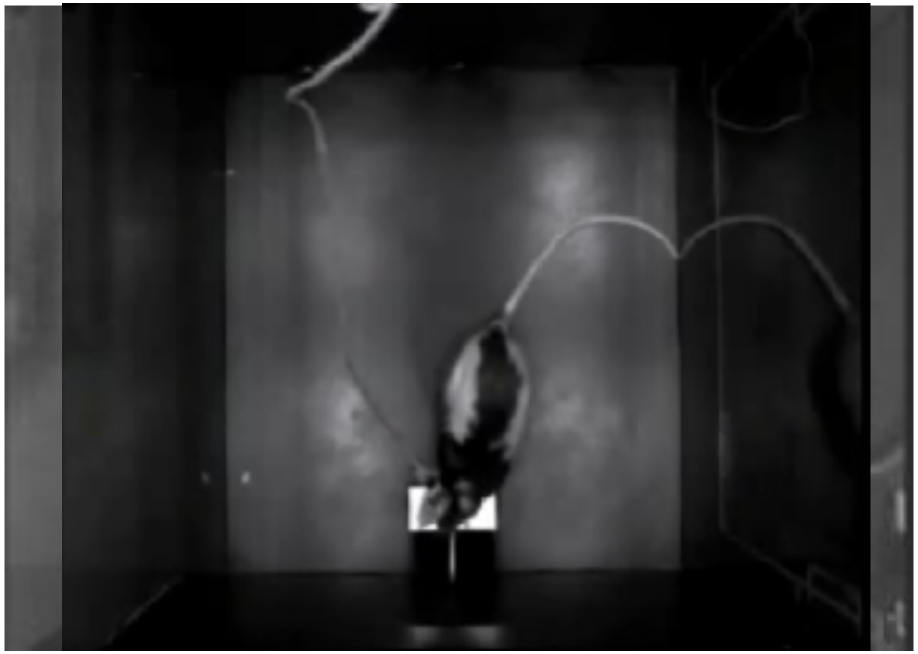
(Doya 2000, 2007)

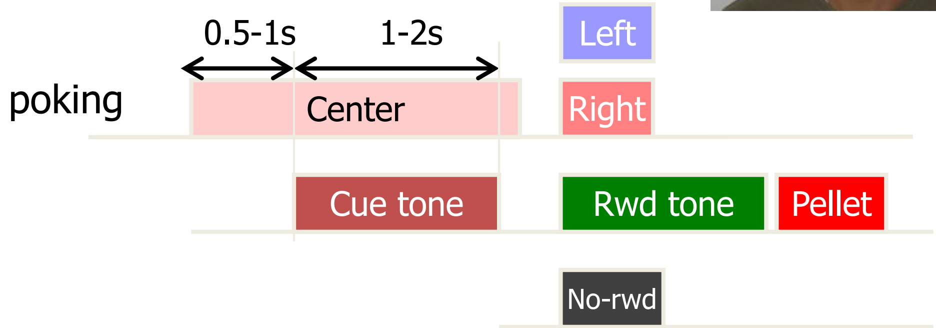


Fixed and Free Choice Task

(Ito & Doya, 2015, J Neuroscience)

Left Center Right

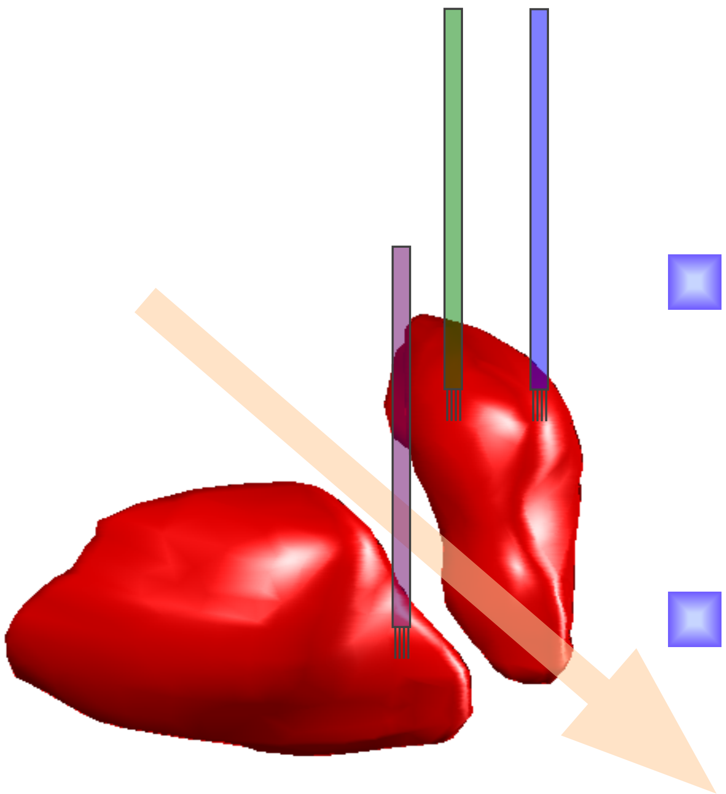




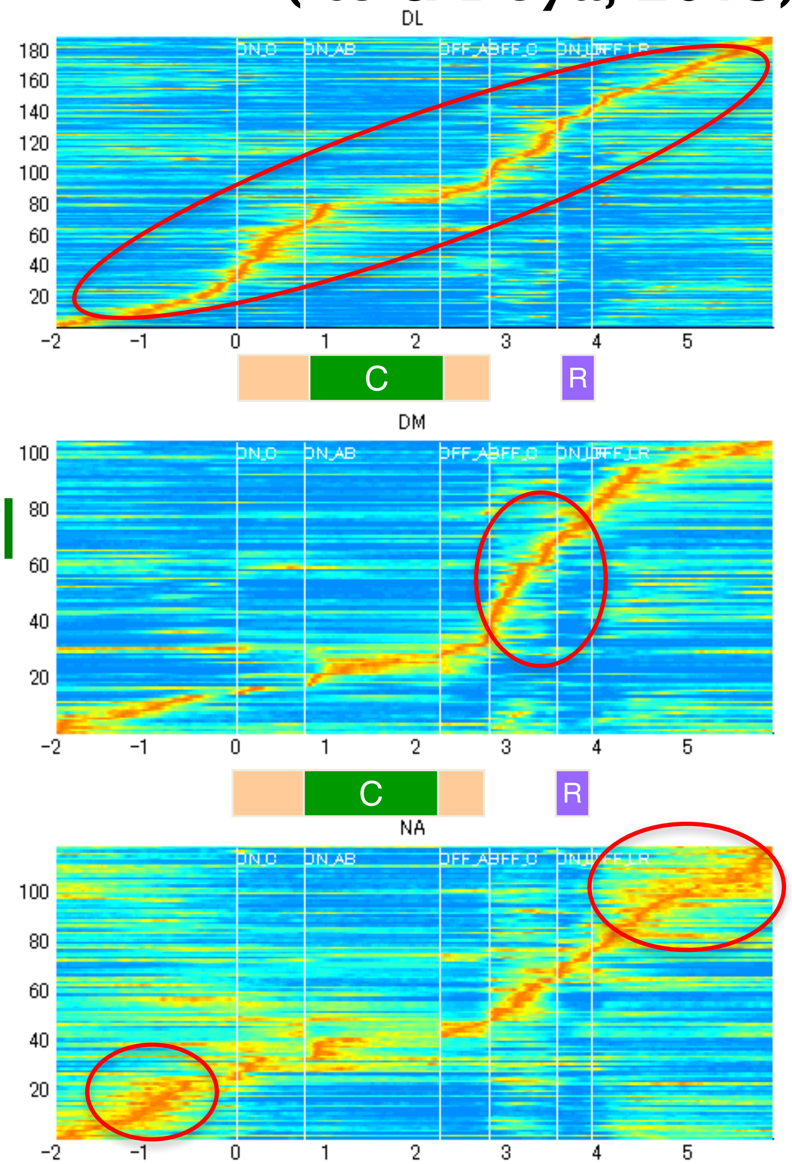
Cue tone	Reward prob. (L, R)
Left tone (900Hz)	Fixed (50%,0%)
Right tone (6500Hz)	Fixed (0%, 50%)
Free-choice tone (White noise)	Varied (90%, 50%) (50%, 90%) (50%, 10%) (10%, 50%)

Neural Activity in the Striatum

(Ito & Doya, 2015)



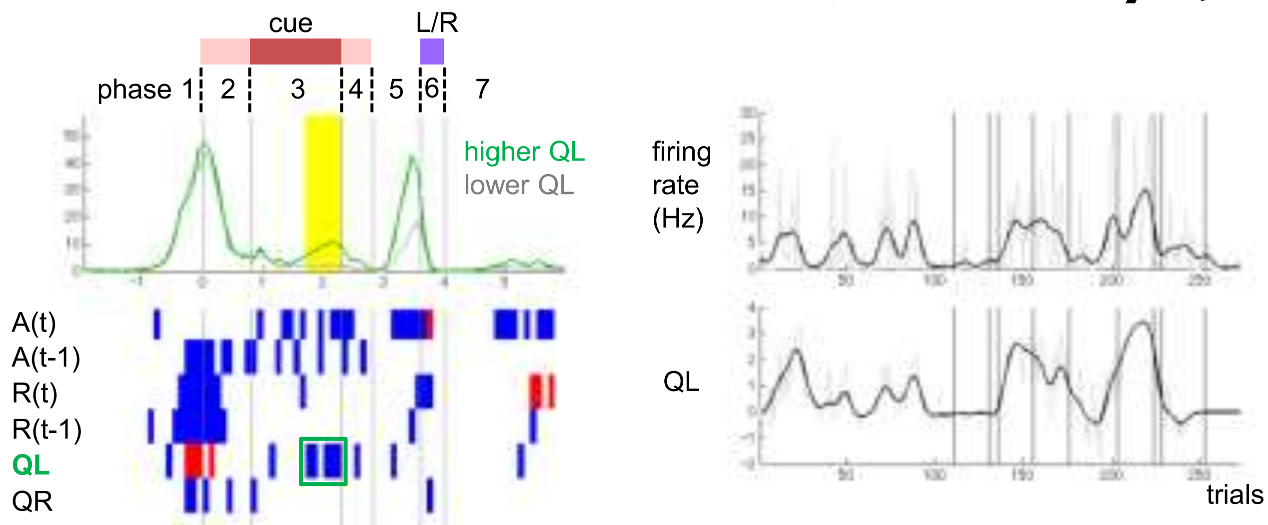
- Dorsolateral
- Dorsomedial
- Ventral



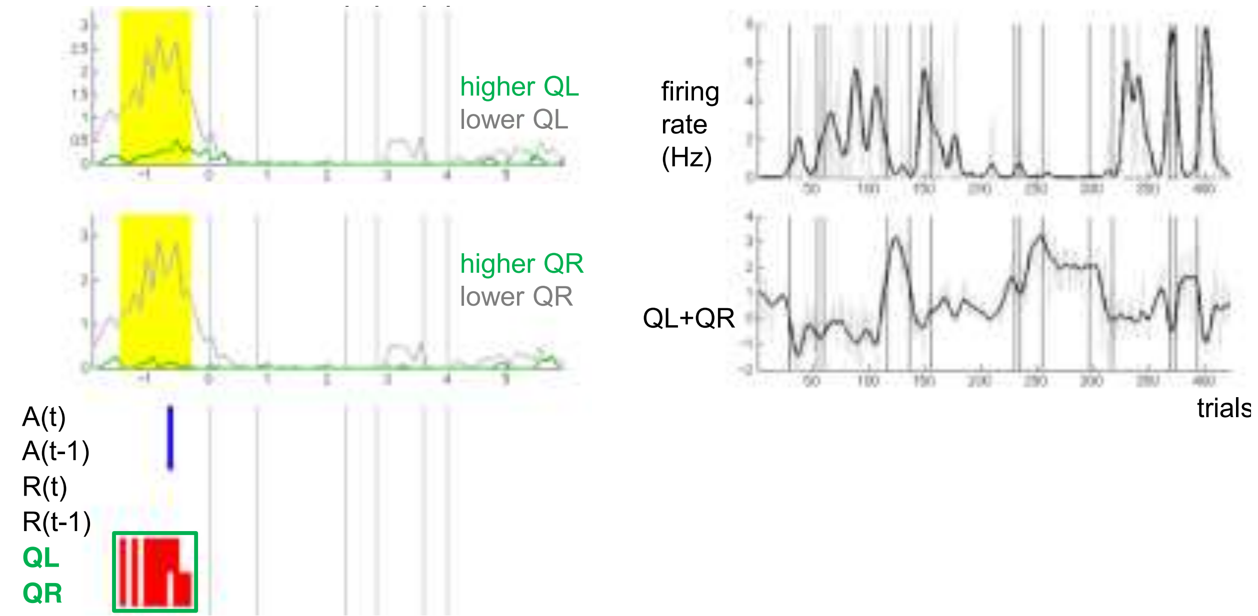
Action/State Value Coding Neurons

(Ito & Doya, 2015)

- Action value
  - DLS
  - DMS

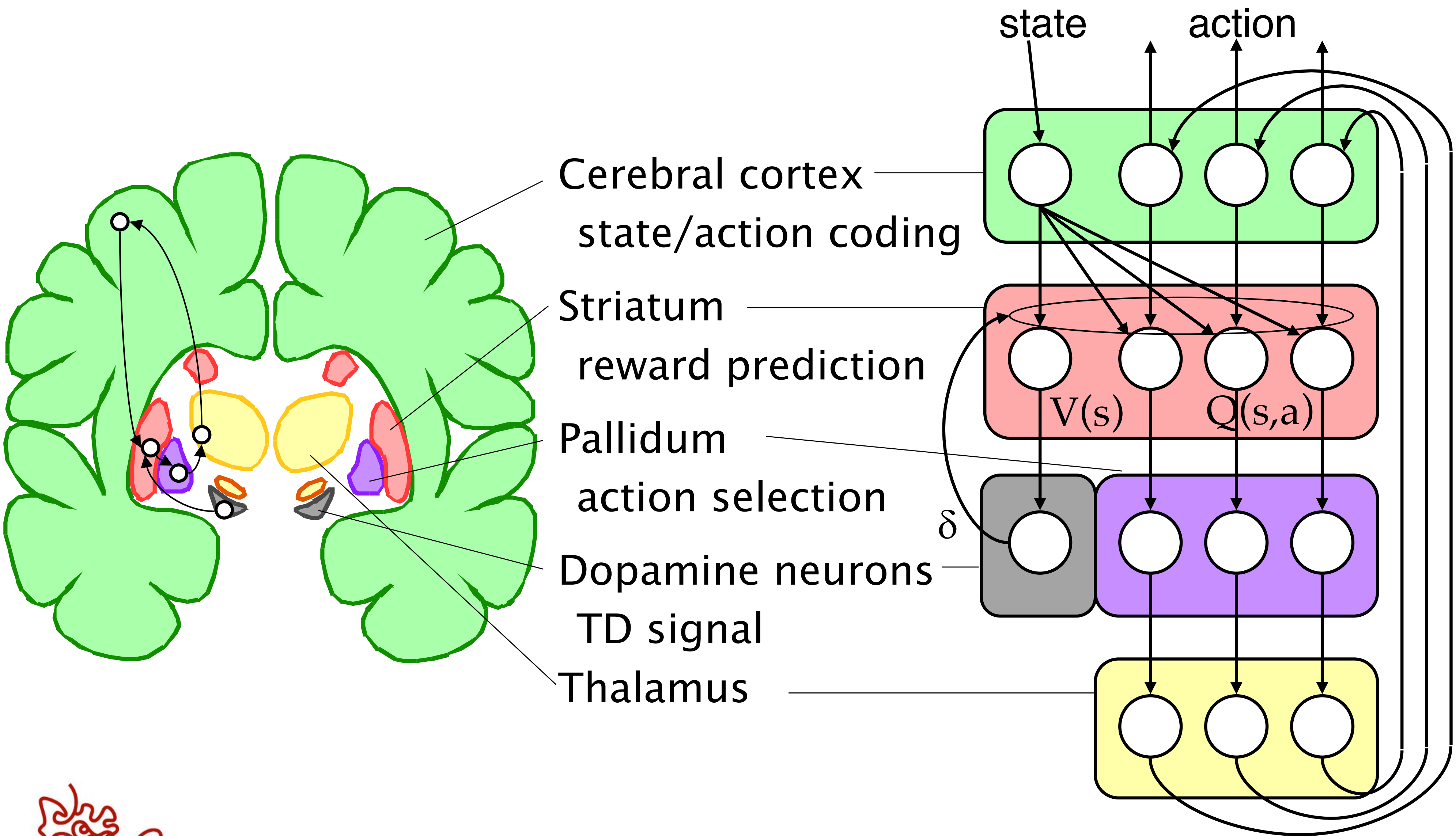


- State value
  - VS



Basal Ganglia for Reinforcement Learning?

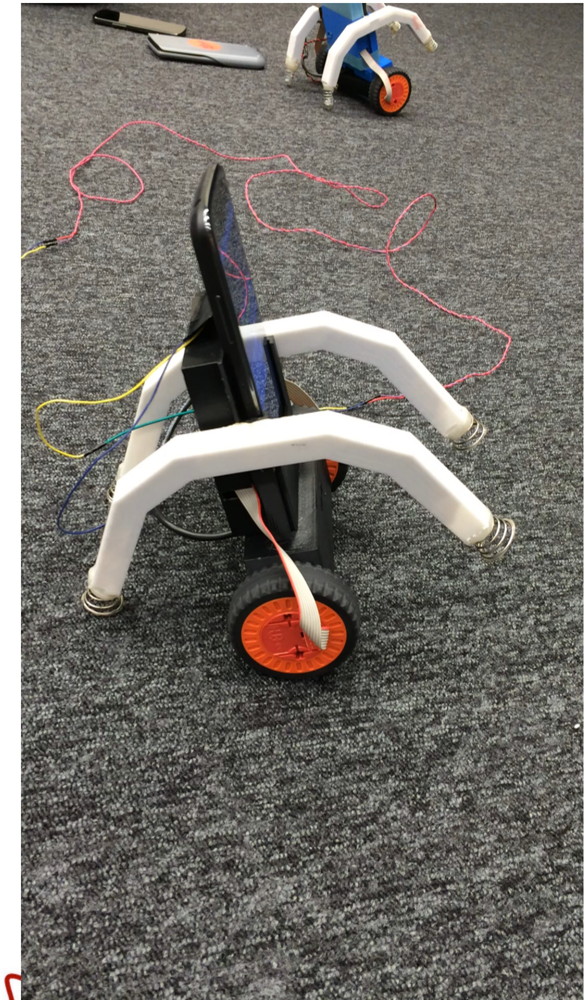
(Doya 2000, 2007)

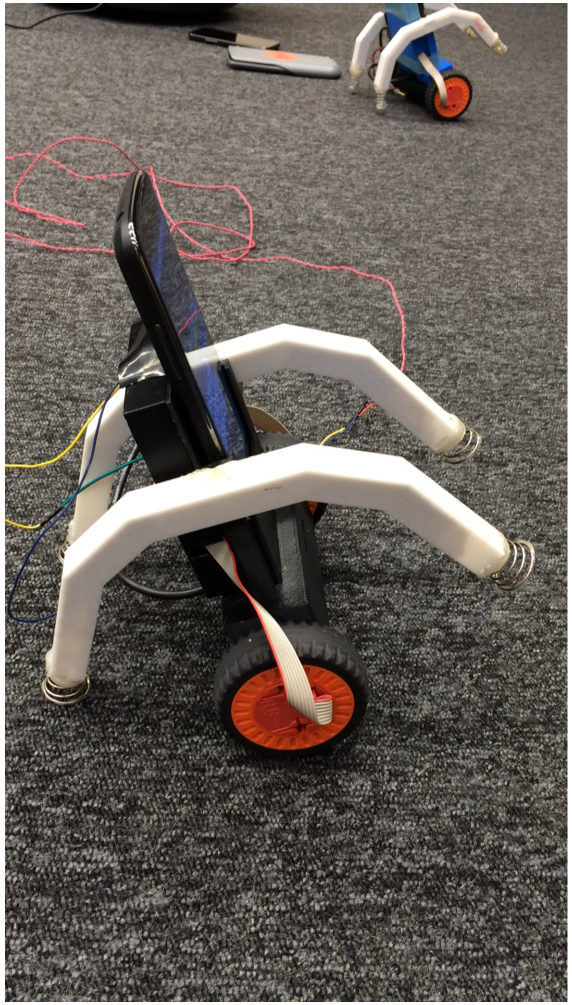


Bounce Up and Balance by PILCO

(Paavo Parmas)

1st try	2nd try	8th try





Model-free/Model-based Decisions

Model-free

- No prior knowledge
- Learn from experience
  - state–action–reward
  - values of states/actions

Simple, but slow learning

Model-based

- Internal model of the world
  - state, action $\rightarrow$ new state
  - state, action $\rightarrow$ reward
- Mental simulation
  - action planning
    find the best action sequence
  - hidden state estimation
    cope with noisy observation

Flexible, but heavy load