LGBIO2060: Modelling of biological systems

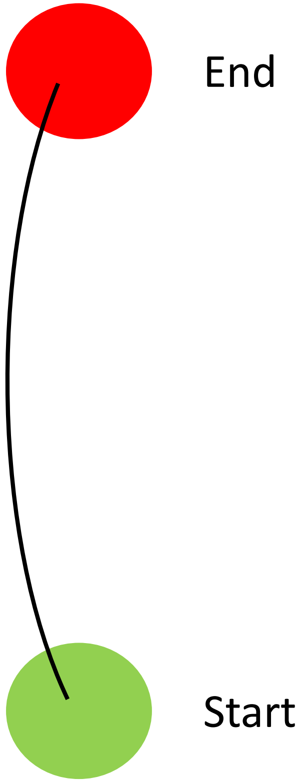
Session 6 : Control theory and applications to motor control

Professor
P. Lefèvre

Teaching assistants
S. Vandergooten
C. Vandamme

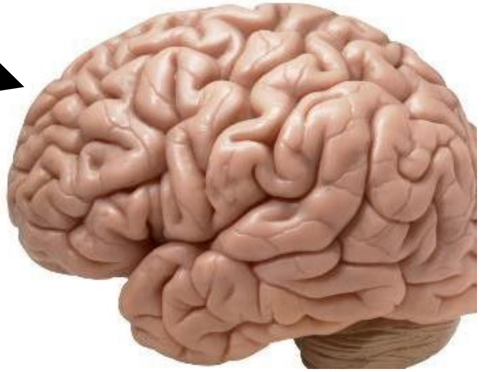
Introduction : Reaching movements

Introduction : Reaching movements



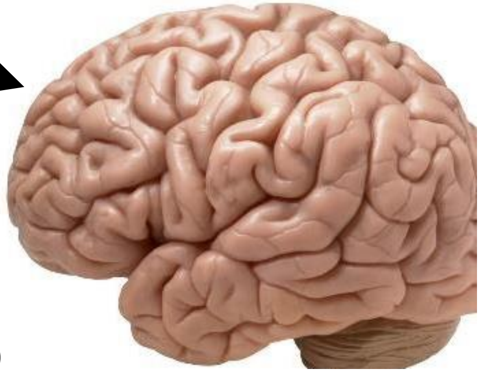
How does the brain implement this ?

Determine the task goal



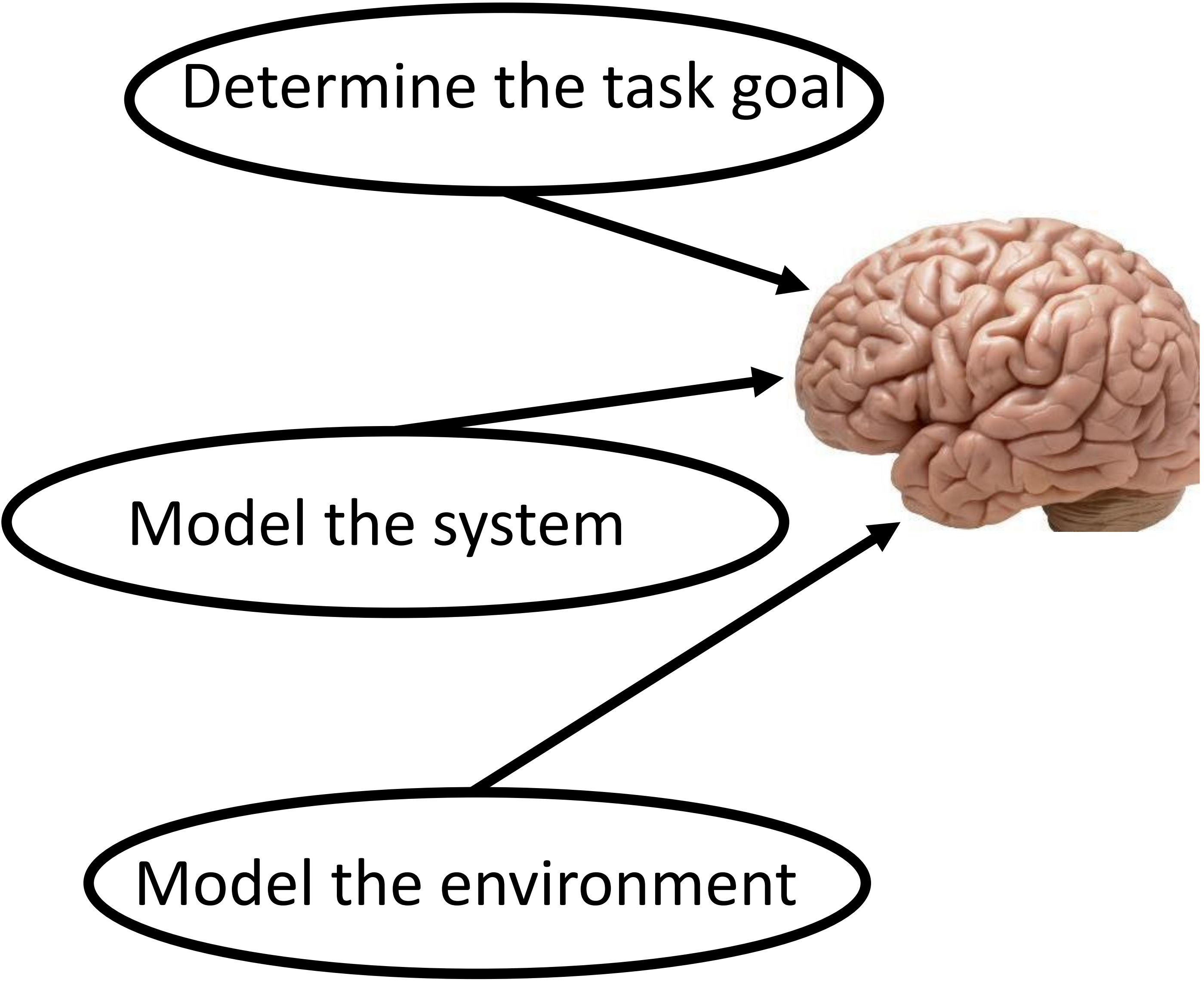
How does the brain implement this ?

Determine the task goal

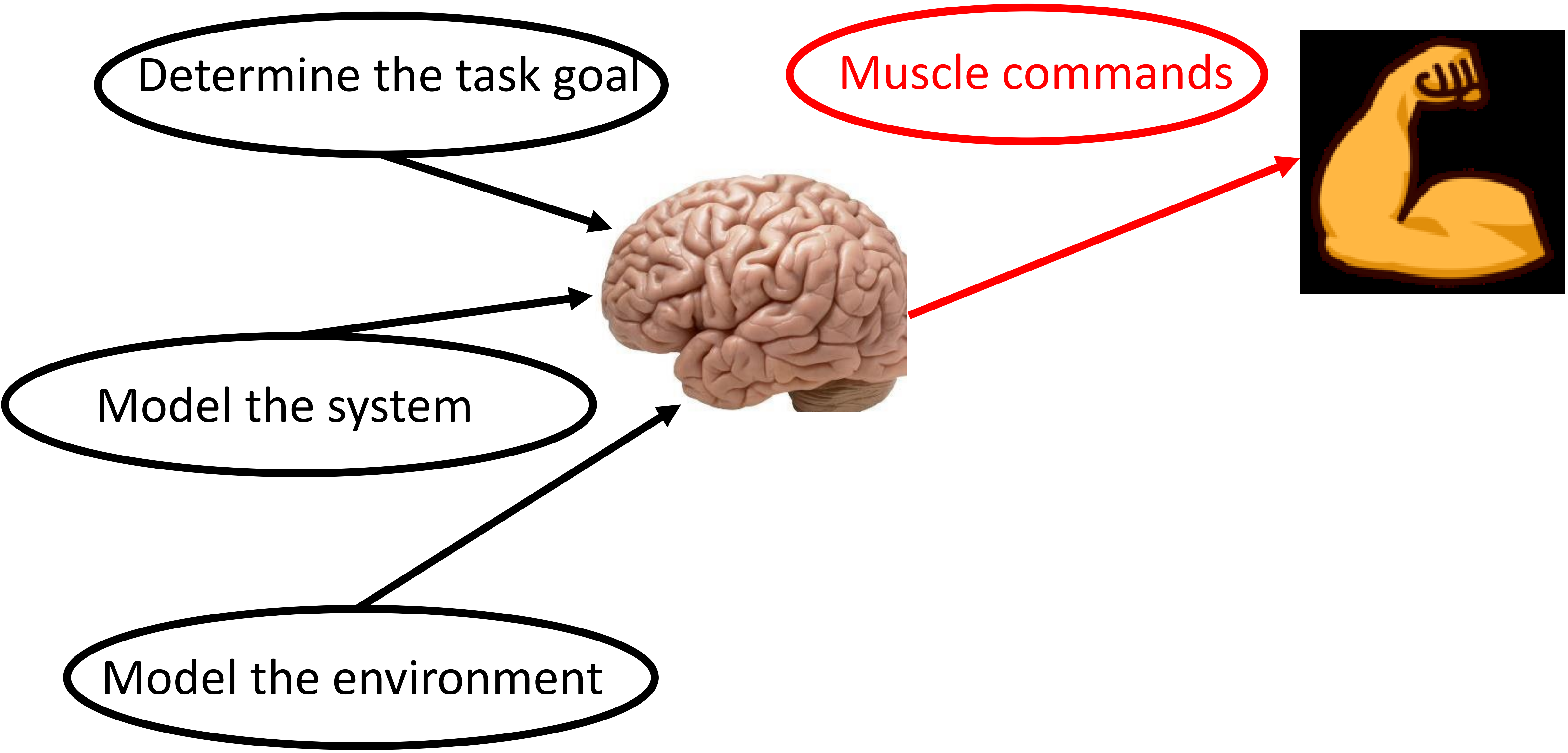


Model the system

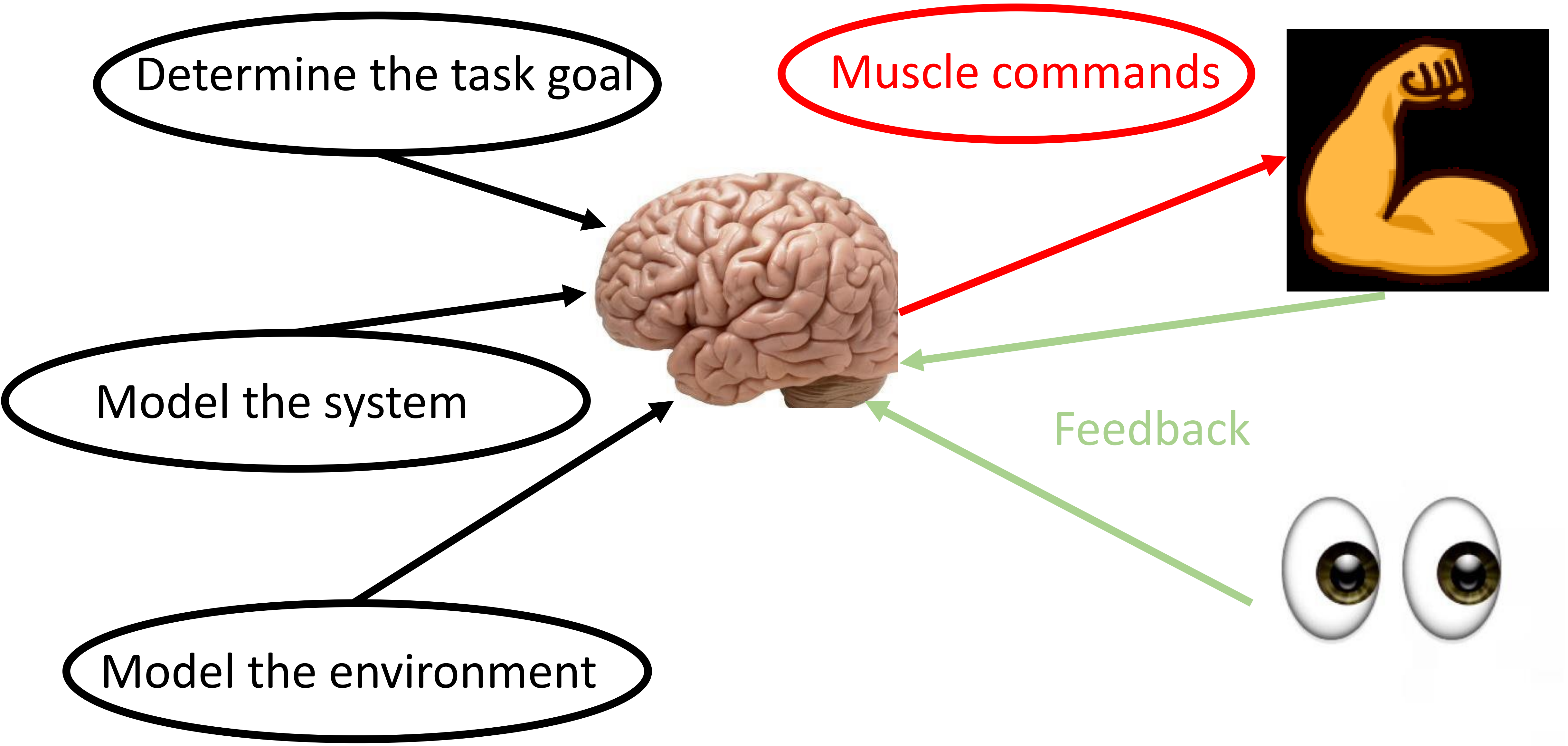
How does the brain implement this ?



How does the brain implement this ?



How does the brain implement this ?



Outline

- How to model the system?
- How to determine and model the task goal and environment?
- How to obtain the muscle commands?
- How to integrate the feedbacks ?

How to model the system ?

System = hand and forearm

Dynamics defined by : $x_{k+1} = Ax_k + Bu_k$

State of the system at time k+1:

- Position of the hand
- Speed of the hand
- Force on the hand
- …

How to model the system ?

System = hand and forearm

Dynamics defined by : $$x_{k+1} = Ax_k + Bu_k$$

Matrix that grasps the kinematics of the movement

State of the system at time k+1:

- Position of the hand
- Speed of the hand
- Force on the hand
- …

How to model the system ?

System = hand and forearm

Dynamics defined by : $x_{k+1} = Ax_k + Bu_k$

Muscle commands at time k

Matrix that grasps the kinematics of the movement

State of the system at time k+1:

- Position of the hand
- Speed of the hand
- Force on the hand
- …

How to model the system ?

System = hand and forearm

Muscle commands at time k

Dynamics defined by : $$x_{k+1} = Ax_k + Bu_k$$

Matrix that grasps the kinematics of the movement

State of the system at time k+1:

- Position of the hand
- Speed of the hand
- Force on the hand
- …

How to obtain this equation ?

Equations of movement

Using Newton's law… (mainly the second)

$$\sum \vec{F} = m\,\vec{a}$$

In both directions!!! (if working in 2D)

This leads to that kind of system :

$$\ddot{x} = -k_v\,\dot{x} + F_x$$
$$\ddot{y} = -k_v\,\dot{y} + F_y$$

$$\tau\,\dot{F}_x = u_x - F_x$$
$$\tau\,\dot{F}_y = u_y - F_y$$

How to discretise this?

Equations of movement

Continuous

$$\ddot{x}(t) = -k_v\,\dot{x}(t) + F_x(t)$$
$$\ddot{y}(t) = -k_v\,\dot{y}(t) + F_y(t)$$

$$\tau\,\dot{F}_x(t) = u_x(t) - F_x(t)$$
$$\tau\,\dot{F}_y(t) = u_y(t) - F_y(t)$$

Finite difference schemes

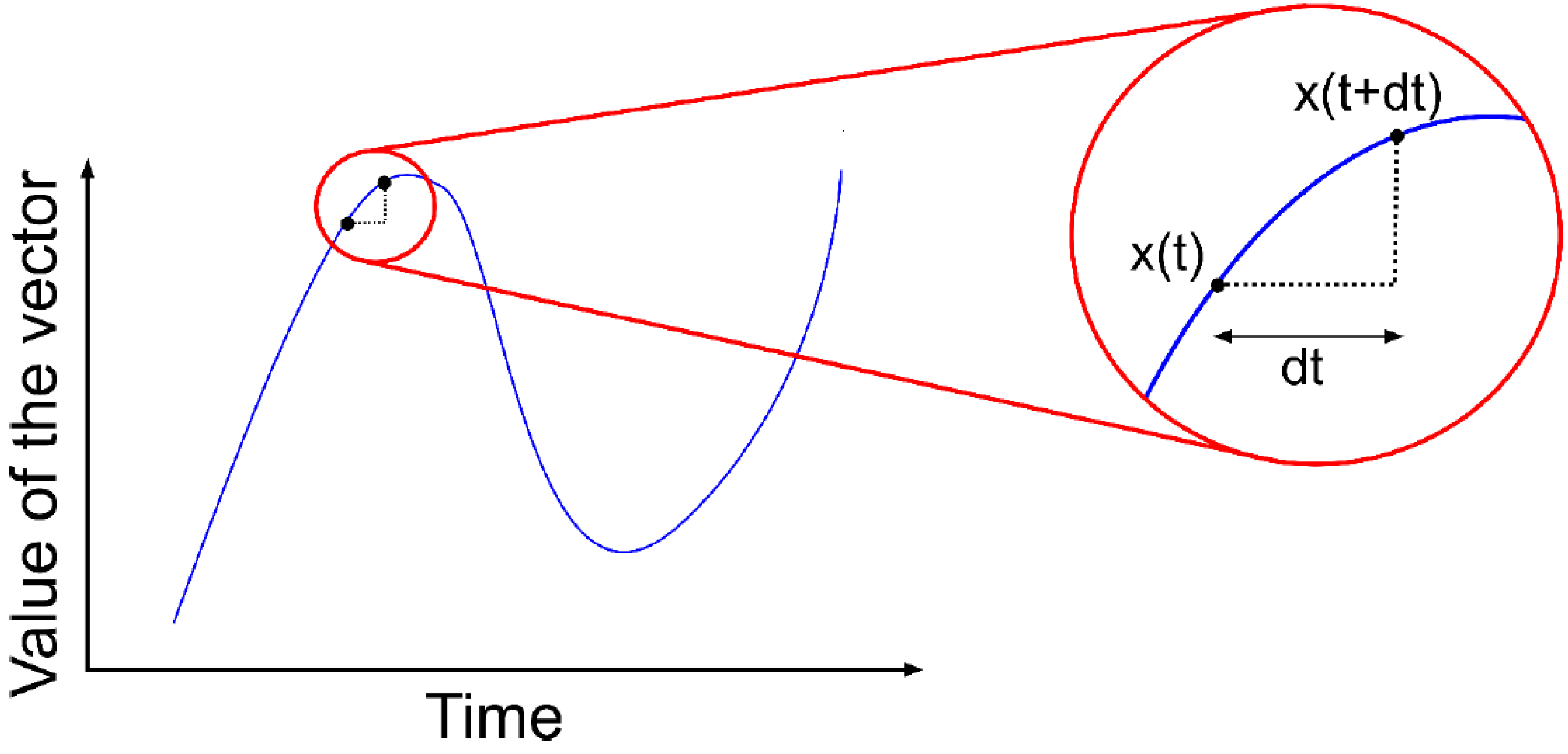
Discrete

$$x[n+1] = Ax[n] + Bu[n]$$

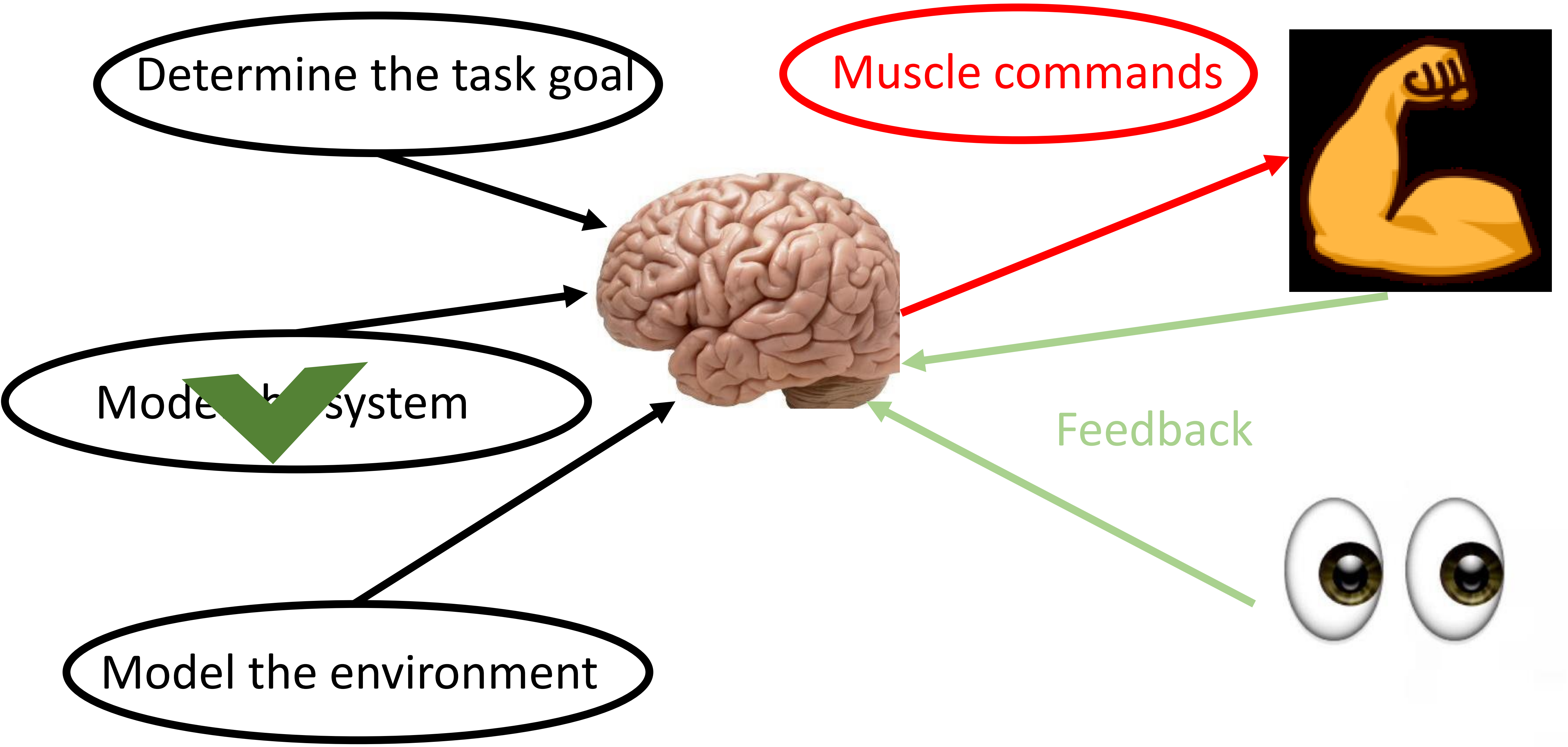
Reminder : Euler implicit (see LFSAB1104)

Allows to approximate a derivative by a finite difference

$$\dot{x}(t) \cong \frac{x(t+\delta t) - x(t)}{\delta t}$$



How does the brain implement this ?



How to determine and model the task goal ?

Task goal is associated by cost function :

Cost function = end-point error + energetic cost

$$J(x,u) = x_N^T Q_N x_N + \sum u_k^T R u_k$$

How does this take the position of the target into account?

What’s inside the state?

Let’s take a look at the state of the system…

Position and speed of the hand

Forces exerted by the hand

Forces exerted on the hand

x_k

Cost function

End-point error :

! For simplicity the target is 0 !

$$(x_{hand} - x_{target})^2 + (y_{hand} - y_{target})^2 + (v^x_{hand} - v^x_{target})^2 + (v^y_{hand} - v^y_{target})^2$$

Matricial form

$$J(x, u) = x_N^T Q_N x_N + \sum u_k^T R u_k$$

→ See section 2

Cost function

End-point error :

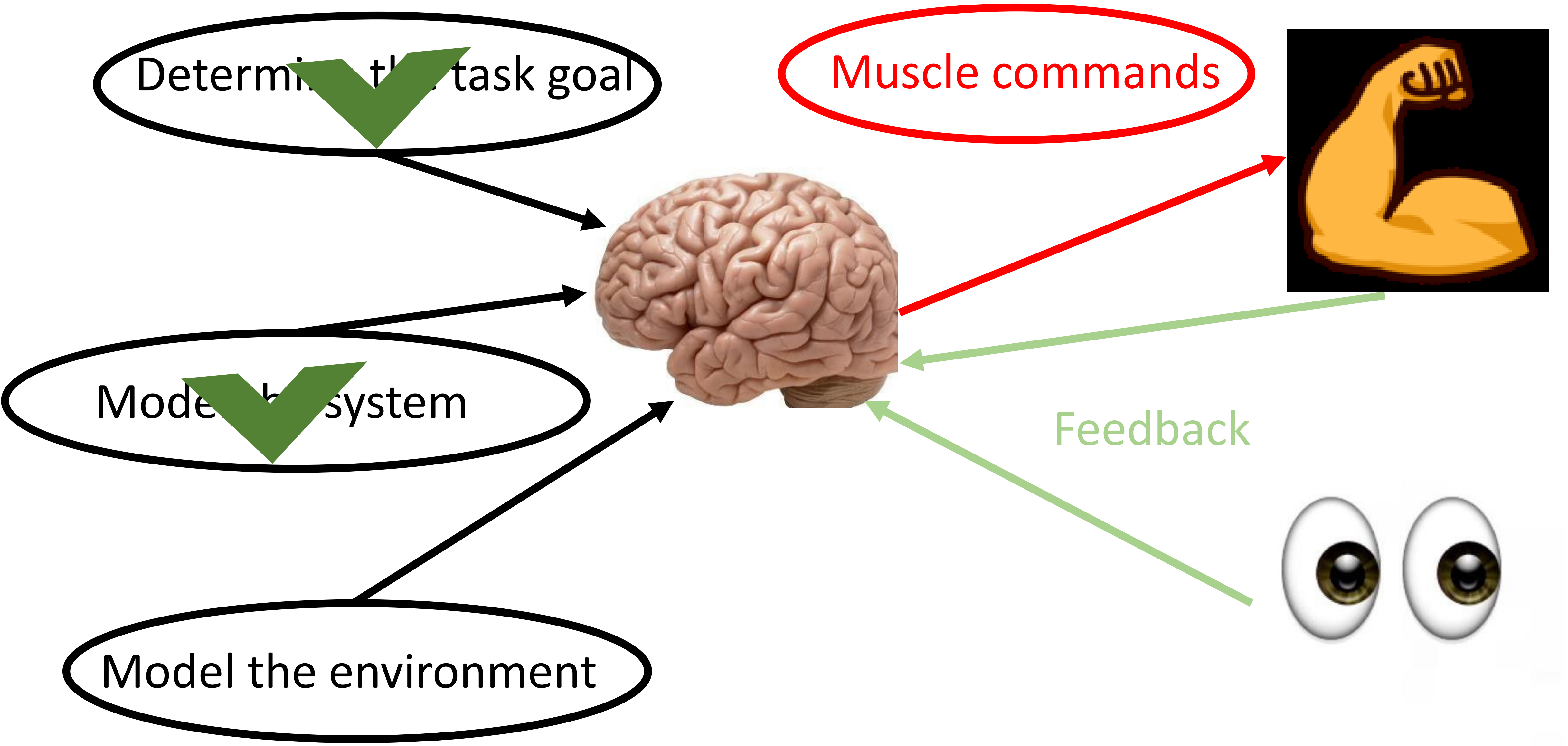
$$(x_{hand})^2 + (y_{hand})^2 + (v^x_{hand})^2 + (v^y_{hand})^2$$

Matricial form

$$J(x,u) = x_N^T Q_N x_N + \sum u_k^T R u_k$$

→ See section 2

How does the brain implement this ?



How does the brain compute the muscle commands ?

The solution is optimal in the sense that it **minimises the objective function**

In the fully observable case, we have the following recurrence

$$L_k = (R + B^T S_{k+1} B)^{-1} B^T S_{k+1} A$$
$$S_k = Q_k + A^T S_{k+1} (A - B L_k)$$

$$S_N = Q_N,$$

$$x_{k+1} = A\, x_k + B\, u_k + \xi_k$$

Where $u_k = -L_k x_k$

$$x_{k+1} = (A - B L_k) x_k + \xi_k$$

Implementation of the backward recurrence ?

1. Determine the matrices Q and R (cfr section 2)

2. Implement the recursion starting **from the end** for the S_k terms

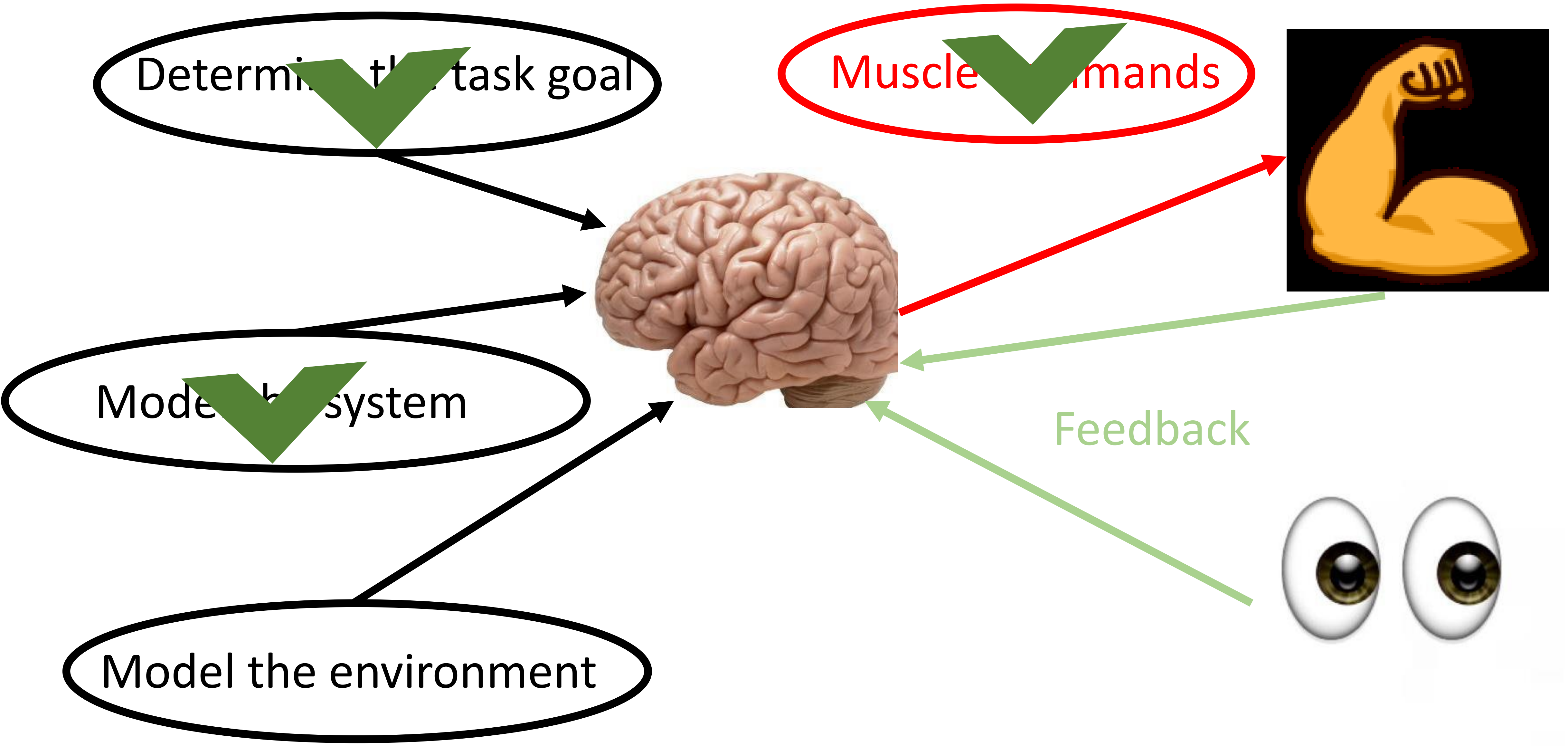
3. Add the recursion for the gains L_k

4. Plug the gains in the closed loop control system thanks to:

$$x_{k+1} = (A - BL_k)x_k + \xi_k$$

SEE SECTION 3

How does the brain implement this ?



Problem ?

No feedback…

Perturbations?

Modification of target?

…

Closed loop control = Blind control

Human body has feedbacks…

Feedbacks from the human body

Eyes and proprioception

$$y_k = Hx_k + \omega_k$$

Noisy observations of the state

Combination of priors and feedback :

$$\hat{x}_{k+1} = (1 - K) \, * prior + K * feedback$$
$$\hat{x}_{k+1} = A\,\hat{x}_k + B\,u_k + K(y_k - H\,\hat{x}_k)$$

How to find the coefficient K?

Feedbacks from the human body

Eyes and proprioception

$$y_k = Hx_k + \omega_k$$

Noisy observations of the state

Combination of priors and feedback :

$$\hat{x}_{k+1} = (1 - K) \; * prior + K * feedback$$
$$\hat{x}_{k+1} = A\,\hat{x}_k + B\,u_k + K(y_k - H\,\hat{x}_k)$$

How to find the coefficient K?

Optimal estimation of the state

Ponderate source by their 'accuracy'

Computation of the different gains

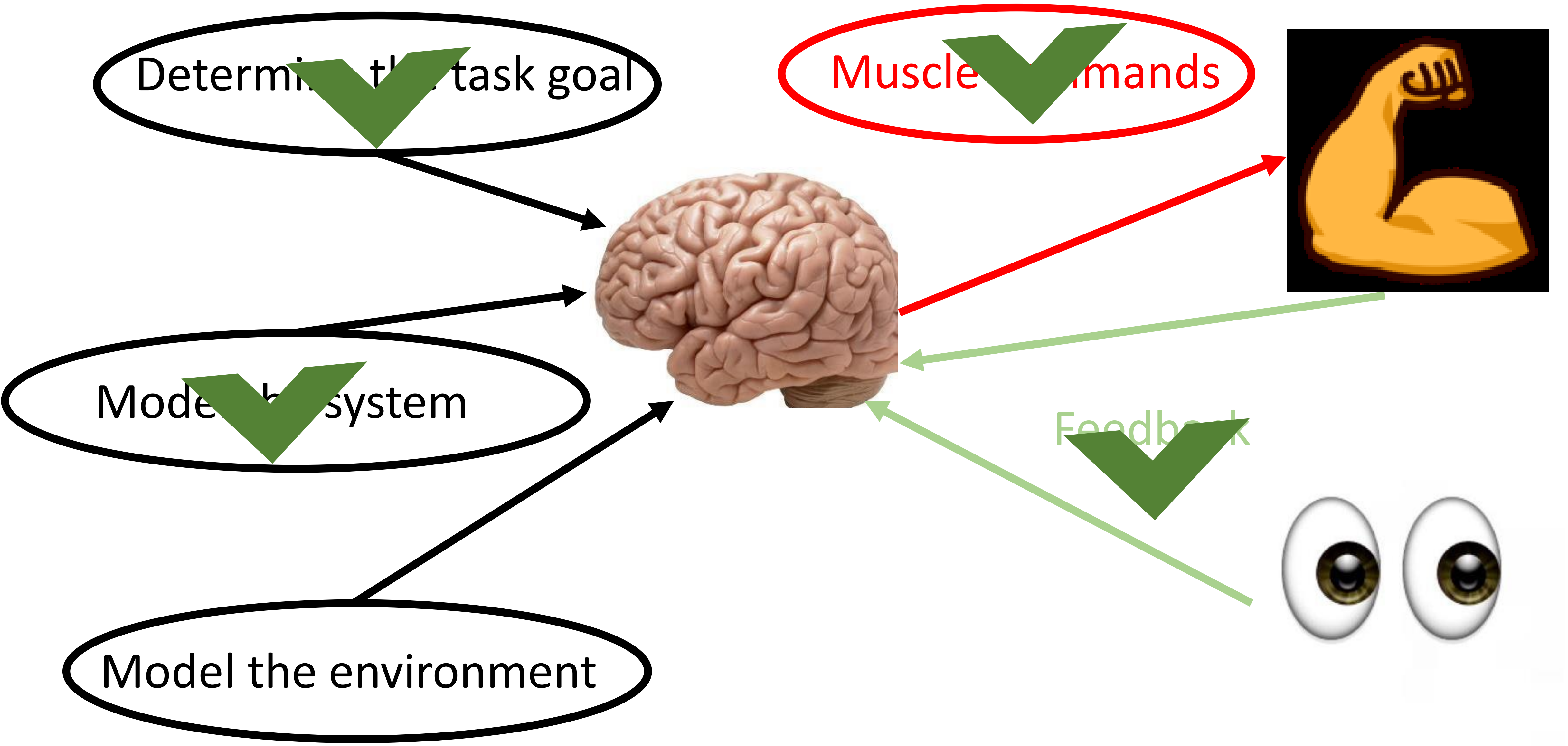
$$\hat{x}_{k+1} = A\,\hat{x}_k + Bu_k + K_k(y_k - H\ \hat{x}_k)$$
$$K_k = A\,\Sigma_k H^T (H\Sigma_k H^T + \Omega_\omega)^{-1}$$
$$\Sigma_{k+1} = \Omega_\xi + (A - K_k H)\Sigma_\mathrm{k}\mathrm{A}^\mathrm{T}$$

The command becomes :

$$u_k = -L_k\,\hat{x}_k$$

SEE SECTION 4

How does the brain implement this ?



What you must have?

A closed-loop **feedback controller** that models simple **reaching movements**

How could it be improved ?

- Adding target in the state vector
- Adding delays in sensory feedbacks
- Adding online modification of targets
- Adding perturbations
- Adding signal dependent noise
- …