Forward and Inverse Models in the Cerebellum

Computational Models of Neural Systems

Lecture 2.3

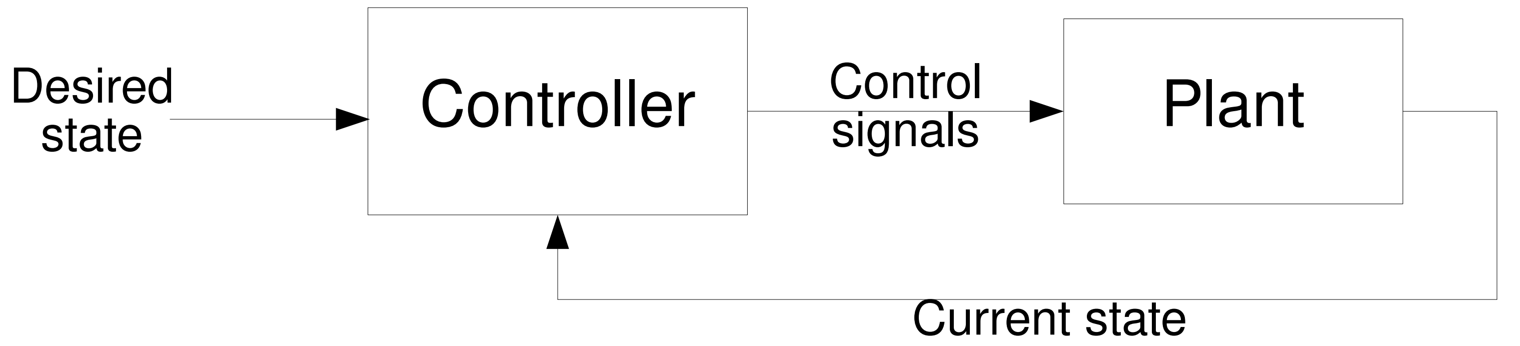
David S. Touretzky

September, 2023

Dynamical Control

- The Marr-Albus models are static models that map a single input pattern to a corresponding output pattern. They don't address dynamics at all.
- How can we provide smooth control of a physical thing (like a limb) that has nontrivial dynamics, e.g., velocity and inertia?
- The "setpoint" theory of control (e.g., E. Bizzi):
  - Cortex/cerebellum specifies a series of positions for the limb
  - Reflexes in the spinal cord cause the motor system to behave like a "<u>spring</u>" and smoothly move each time the <u>setpoint</u> changes.
  - Problem: this only works for "stiff" (high gain) actuators.
  - Experiments show that the motor system is not stiff.
- Alternative approach: use an <u>inverse</u> <u>dynamics</u> <u>model</u>.

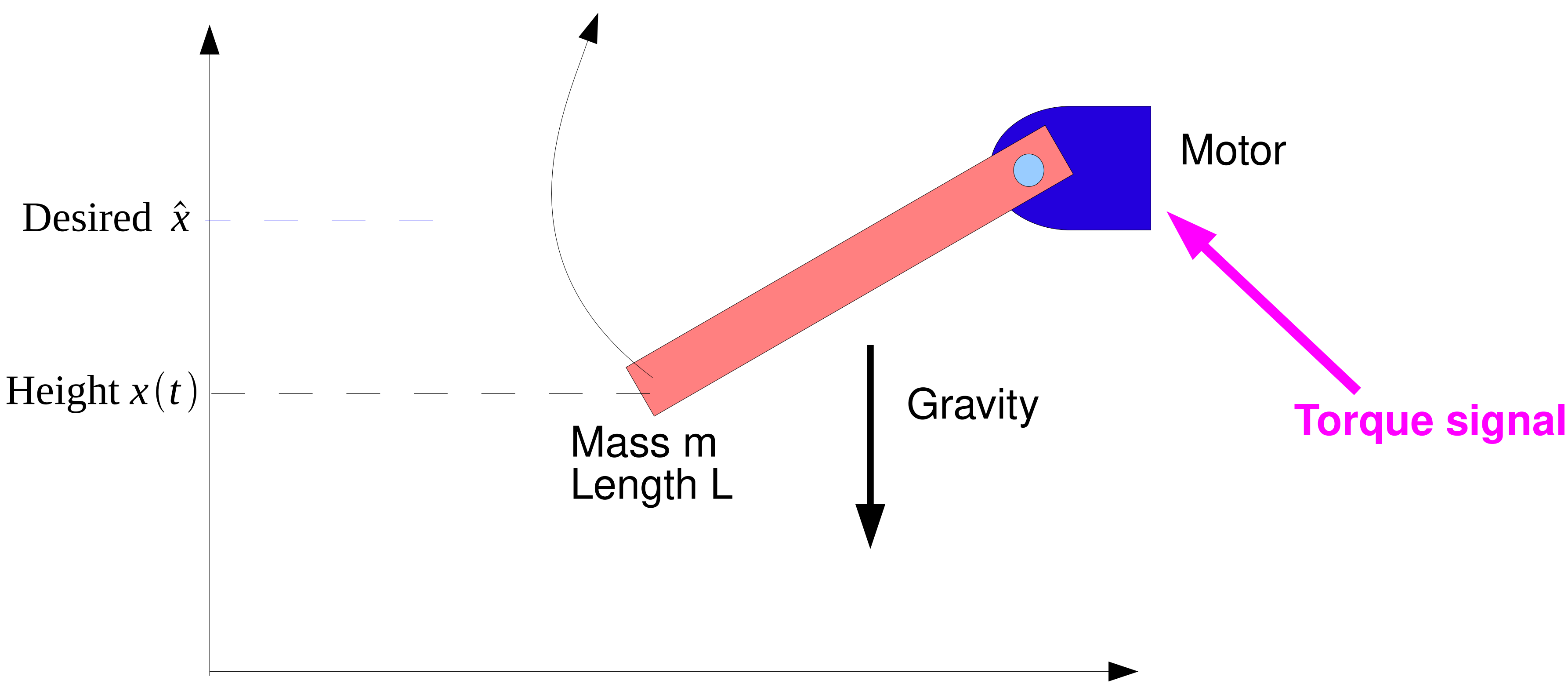
Basics of Control Theory



- The “**plant**” is the thing being controlled.
- The controller translates desired states into **control signals**.
- Control signals might be motor torques or muscle activations.
- The current state could be just the joint positions, or it could include joint velocities, accelerations, load signals, etc.
- Complications: actuators may be slow to respond; feedback may be delayed.

Feedback Control

- A simple way to control a plant is to try to continuously reduce the difference between its current state and the desired state.
- Simple example: control the height of a swinging arm by varying the torque on a motor.

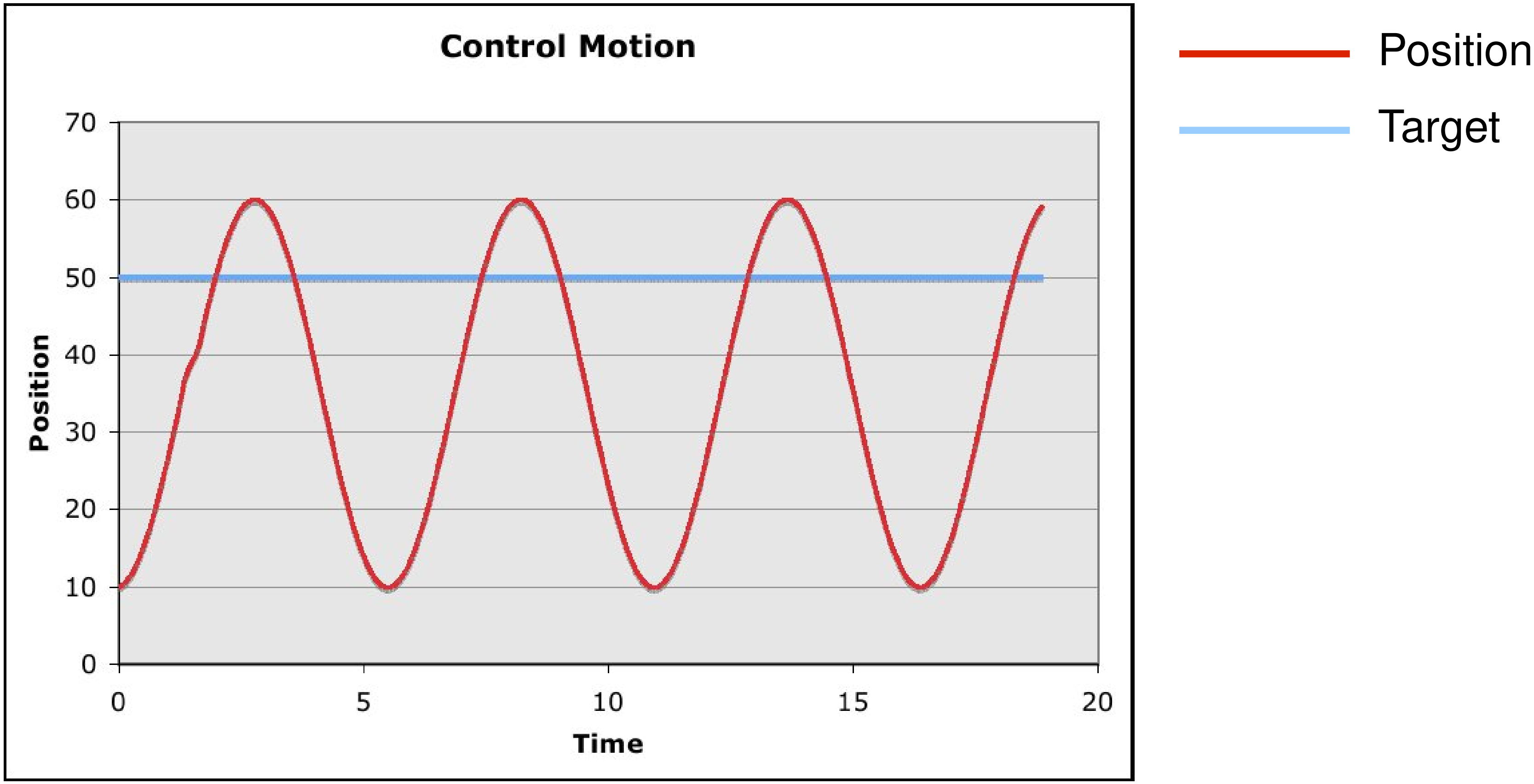


Proportional Control

$$
\begin{aligned}
x(t) &= \text{current position} \\
\hat{x} &= \text{desired position} \\
e(t) &= x(t) - \hat{x} \quad \text{error signal} \\
torque &= -k_p \cdot e(t)
\end{aligned}
$$

- Larger error will generate more torque, proportional to k_p.
- This is a spring model: F = -kx
- When error is zero, torque is zero.
  - But error won't stay zero due to gravity pulling the arm down.

Proportional Control Is Unstable



- Position oscillates and never converges
- Doesn't even oscillate around the target value.

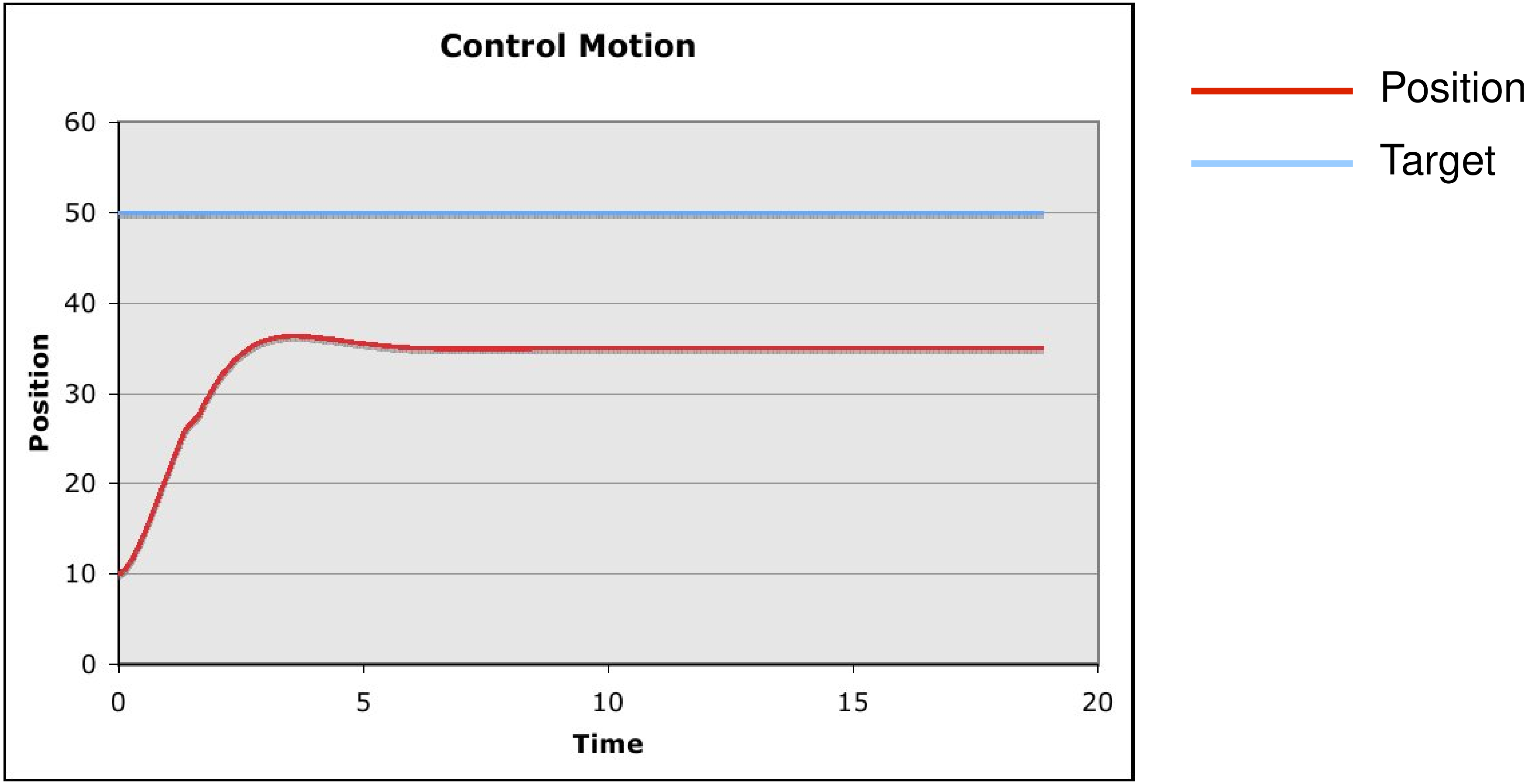
Proportional-Derivative Control

- Oscillation occurs because inertia keeps the arm moving even as the error (and applied torque) are reduced.
- Solution: introduce a braking factor k_d multiplied by the derivative of the error.
  - If error is falling rapidly, apply the brakes so we don't overshoot.

$$torque = -k_p \cdot e(t) - k_d \cdot \frac{\partial e(t)}{\partial t}$$

PD Control Undershoots

- The arm asymptotes at a position where the force of gravity exactly balances the torque from the residual error.

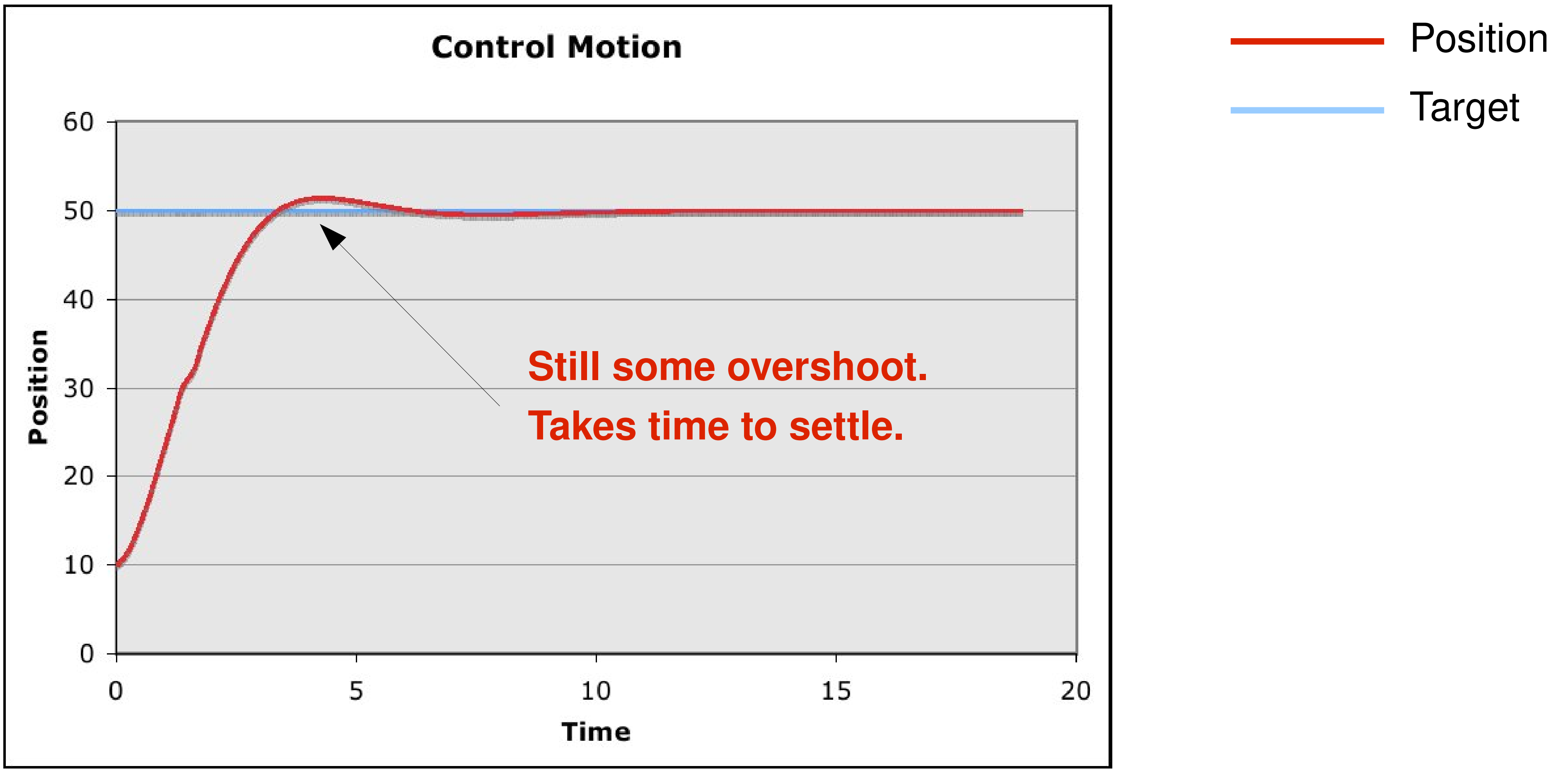


Proportional-Integral-Derivative Control

- Need another term to counteract constant inputs to the system, such as gravity pulling the arm down.
- Use an integral of the error term, so persistent error will gradually be met with increasing force.

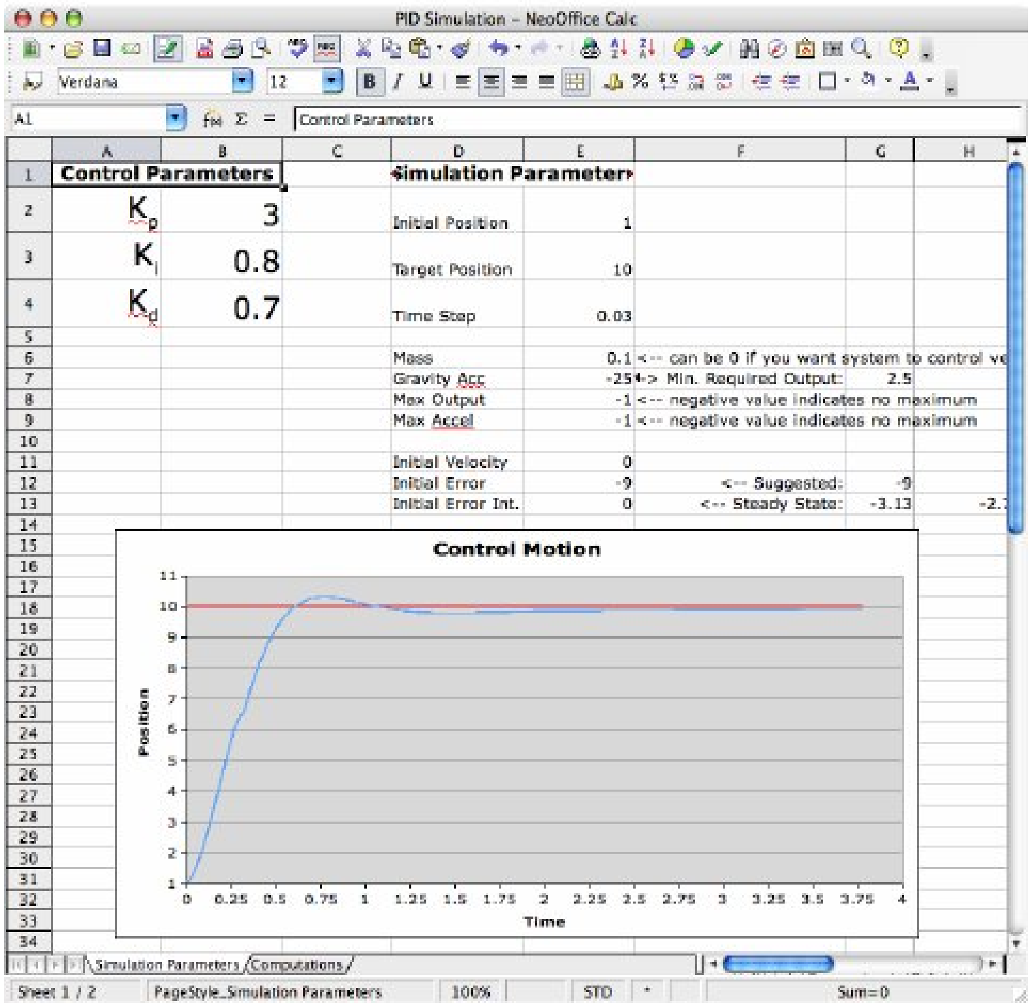
$$torque = -k_p \cdot e(t) - k_i \cdot \int e(t)\,dt - k_d \cdot \frac{\partial e(t)}{\partial t}$$

PID Control Works Better



Demos

- Excel spreadsheet for PID control:



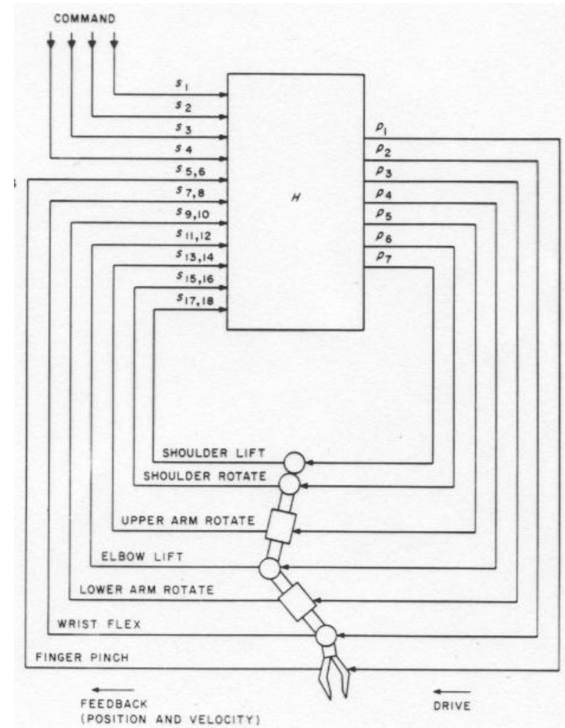
- Video of P vs. PID control of a wheeled cart
- Video of 2-dof inverse pendulum controller.

Control Theory: General

- Branch of engineering and mathematics dealing with dynamical systems.
- If we have a complete description of the system (mass distribution, torques, friction) we can derive controllers for it mathematically.
    - Differential equations describe the system.
    - Many control strategies possible: linear, nonlinear, adaptive, …
- Model identification: learning the system description through observation.
- Machine learning can be used to learn an efficient controller from experience.

Plants With Complex Dynamics

Simple PID controllers won't work well for plants where the actuators can interact and the dynamics are complex.



Instead, we need a **model** of the plant that captures these complex dynamics.

Forward model: maps control signals to predicted plant behavior.

Inverse model: maps desired behavior to control signals that will produce that behavior.

Wolpert et al.

- Simple feedback controllers (setpoint) won't work for animals because biological feedback loops are slow and have small gains (not stiff).

- Proposal: use an inverse model to anticipate what the plant will do and generate appropriate control signals.

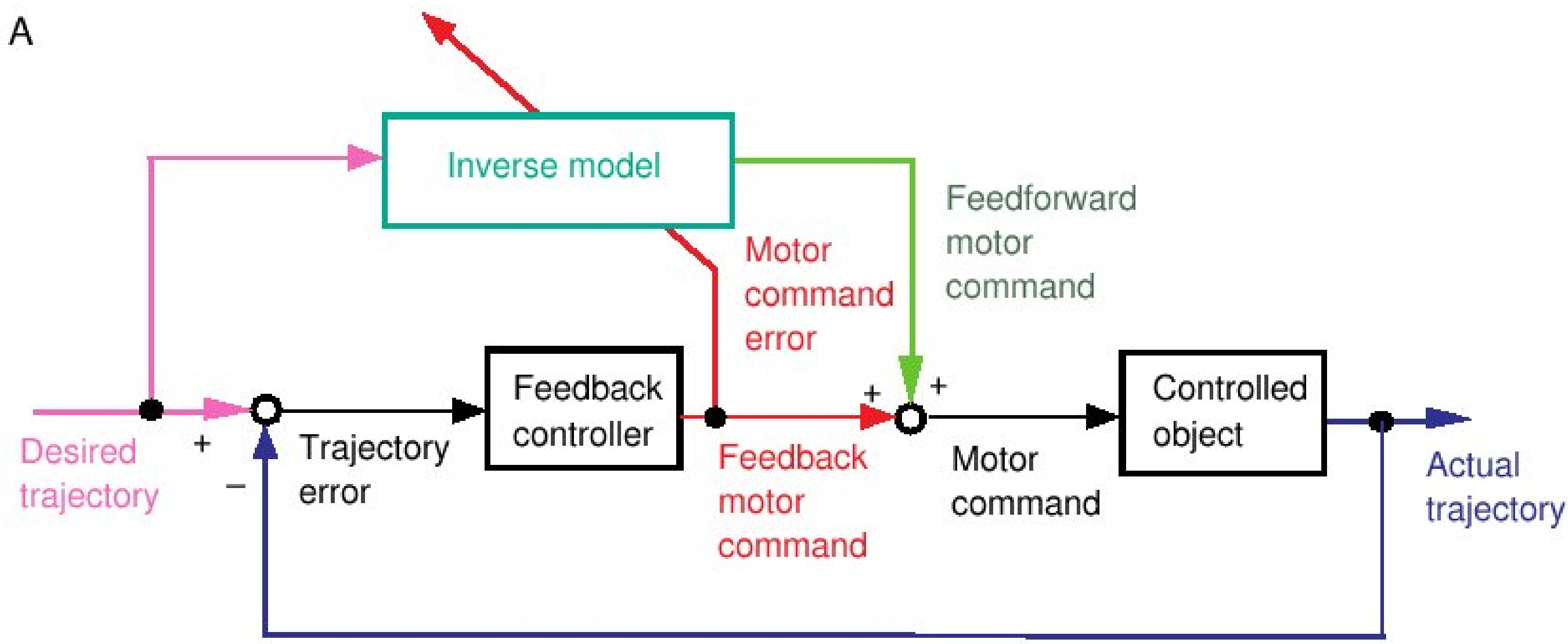
- But how do we train an inverse model?
  - We don't know the correct control signals to start with.
  - So how do we correct errors in the inverse model's output?

Representations in Arm Control

- Sensory space
  - Perceived location of the hand
  - Could be in retinal coordinates (x,y), or body coordinates (x,y,z)
- Joint or motor command space
  - Joint angles (shoulder, elbow, wrist, etc.) or …
  - Motor commands: one dimension per muscle
- Trajectory space
  - Desired limb trajectory to accomplish an action (e.g., grasping)

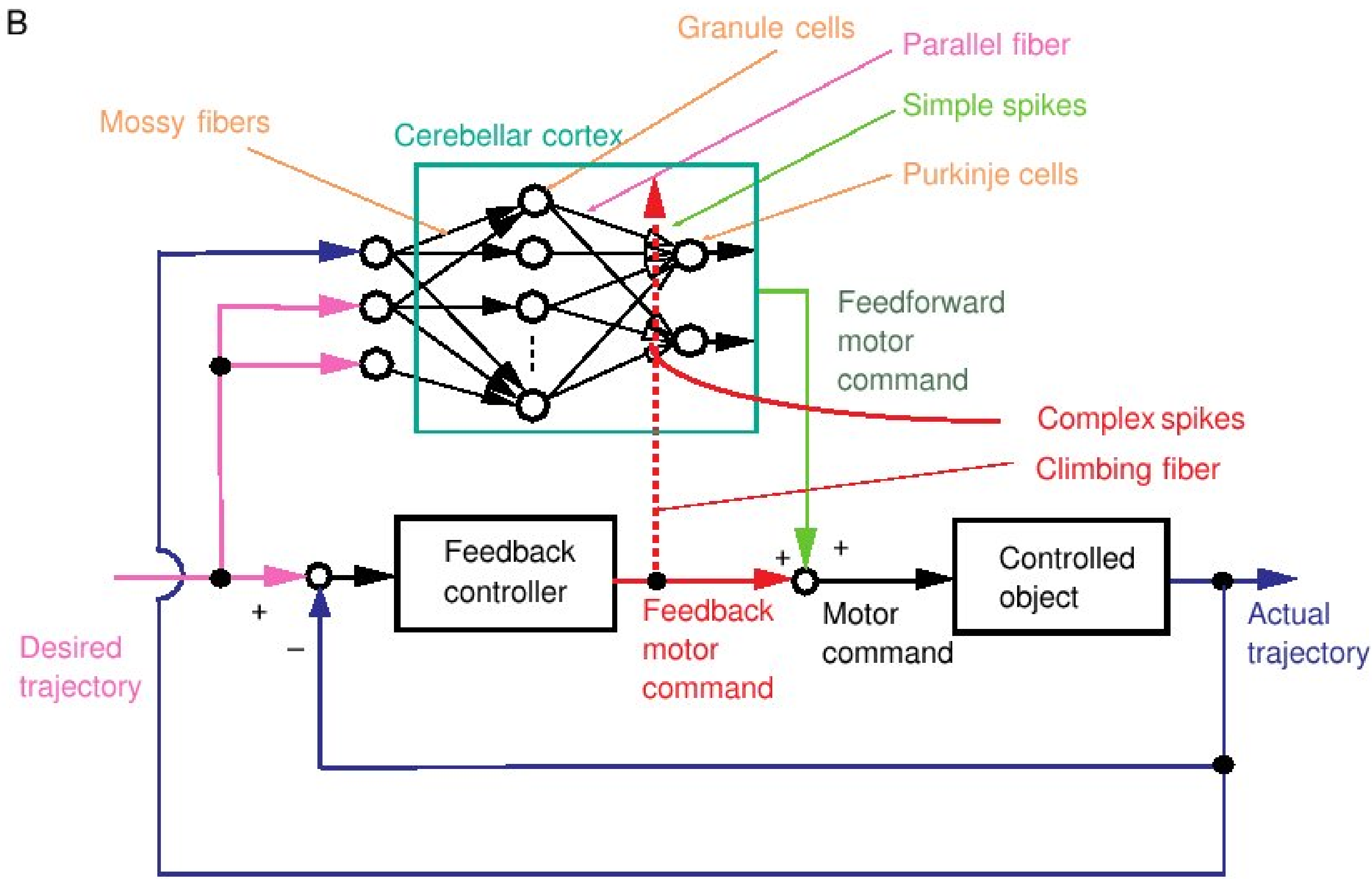
Training the Inverse Model

- Assume a feedback controller that can convert sensory signals to control signal error.
- Use this error to train the inverse model.



Does the Cerebellum Contain Inverse Models?

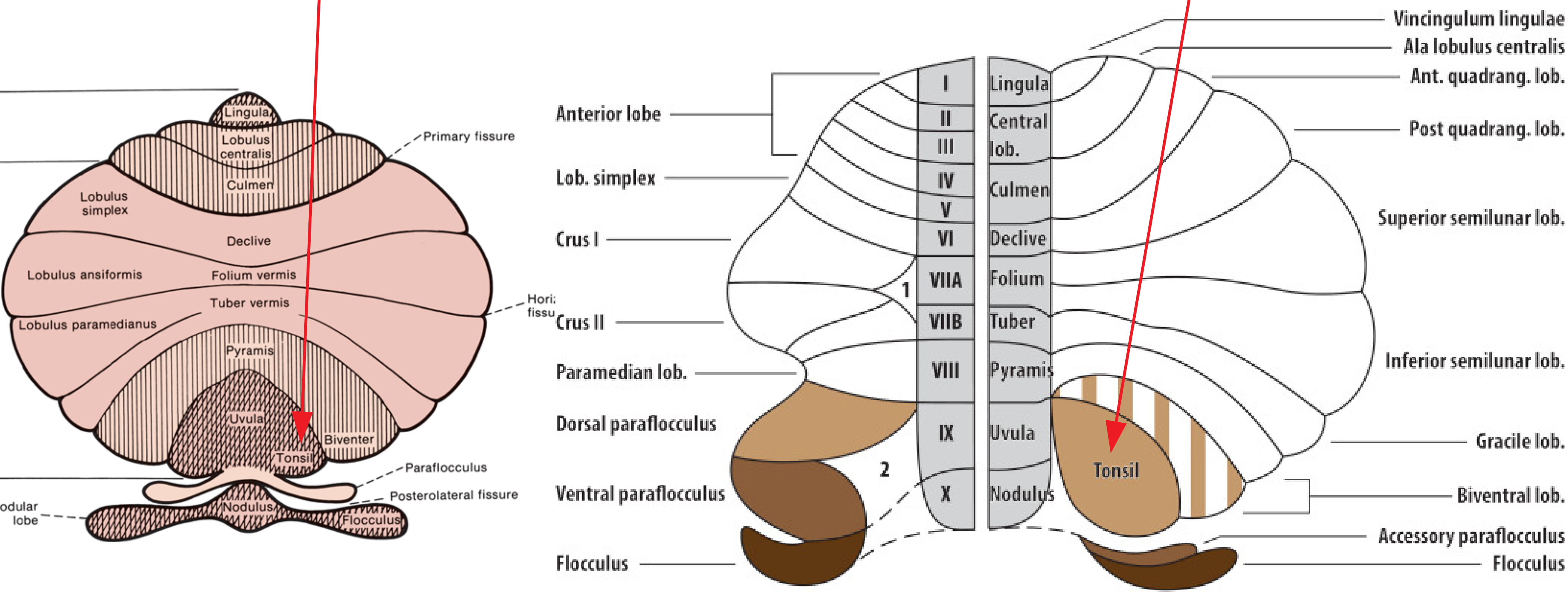
Kawato's CFBELM (Cerebellar Feedback-Error Learning Model)



Apply this model to OFR (Optical Following Response).

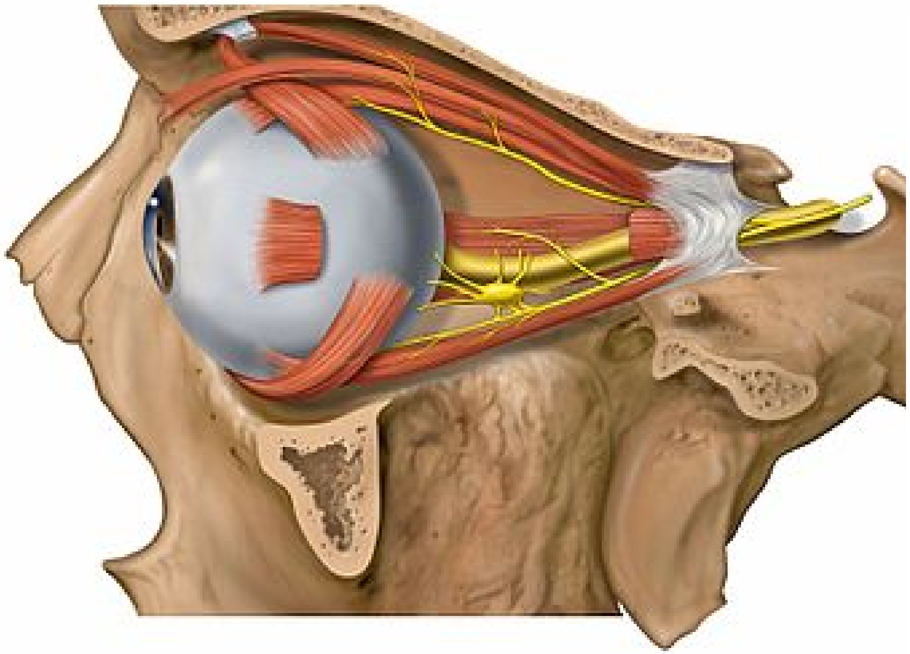
Cerebellar Control of Eye Movements

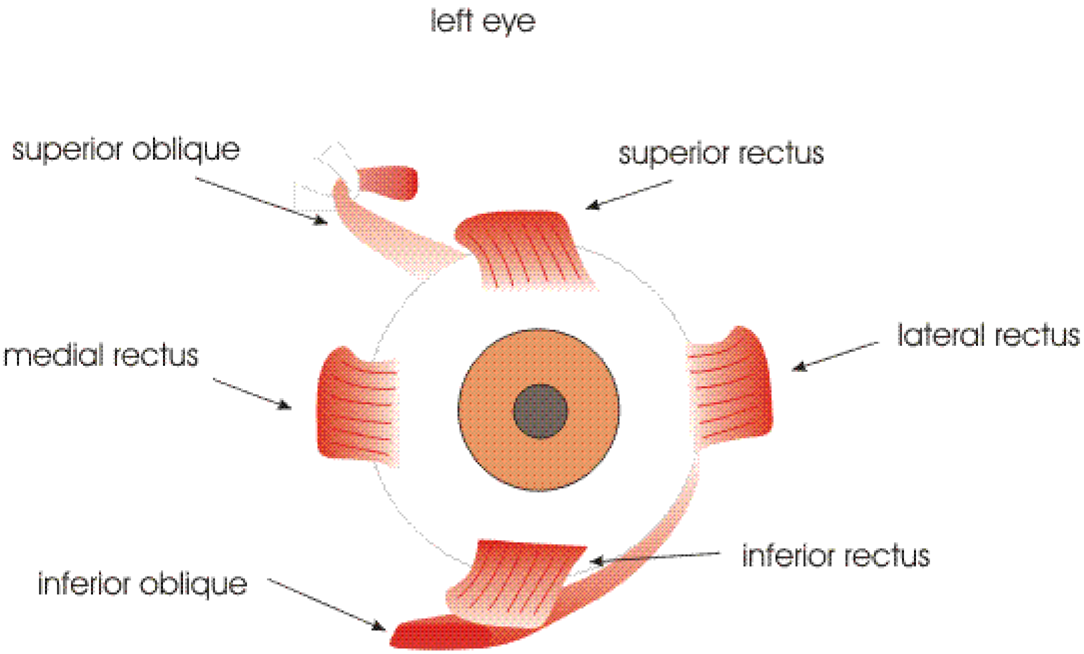
- Assume each cerebellar “microzone” contains a separate inverse model for some part of the body.
- Optical following response (OFR) generated in the tonsil (ventral paraflocculus).

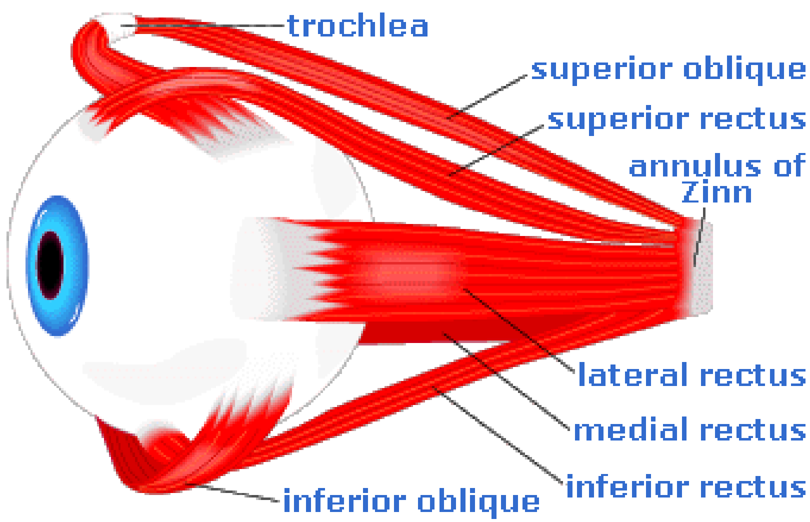


Non-human mammalian cerebellum
Human cerebellum

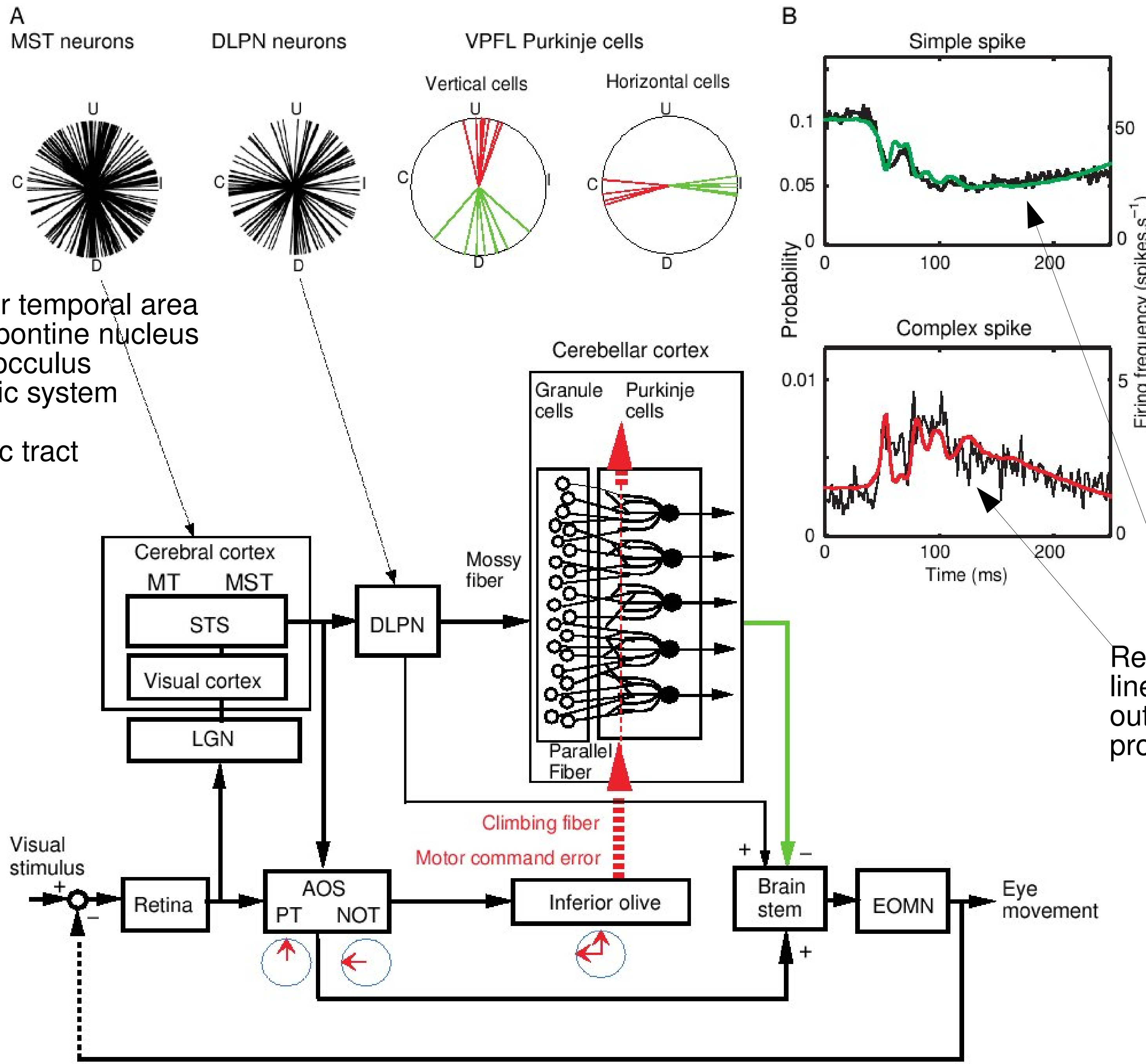
Musculature of the Eye







Ocular Following Response (OFR)

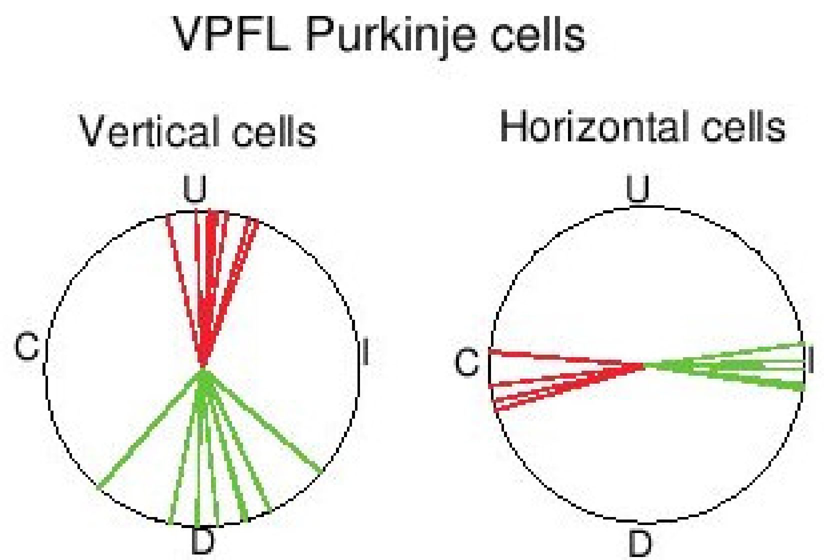


MST: Medial superior temporal area
DLPN: Dorsolateral pontine nucleus
VPFL: ventral paraflocculus
AOS: Accessory optic system
PT: Pretectum
NOT: nucelus of optic tract
EOMN: extra-ocular motor neurons

Red and green lines = model output: firing probability

Measured Purkinje Cell Responses

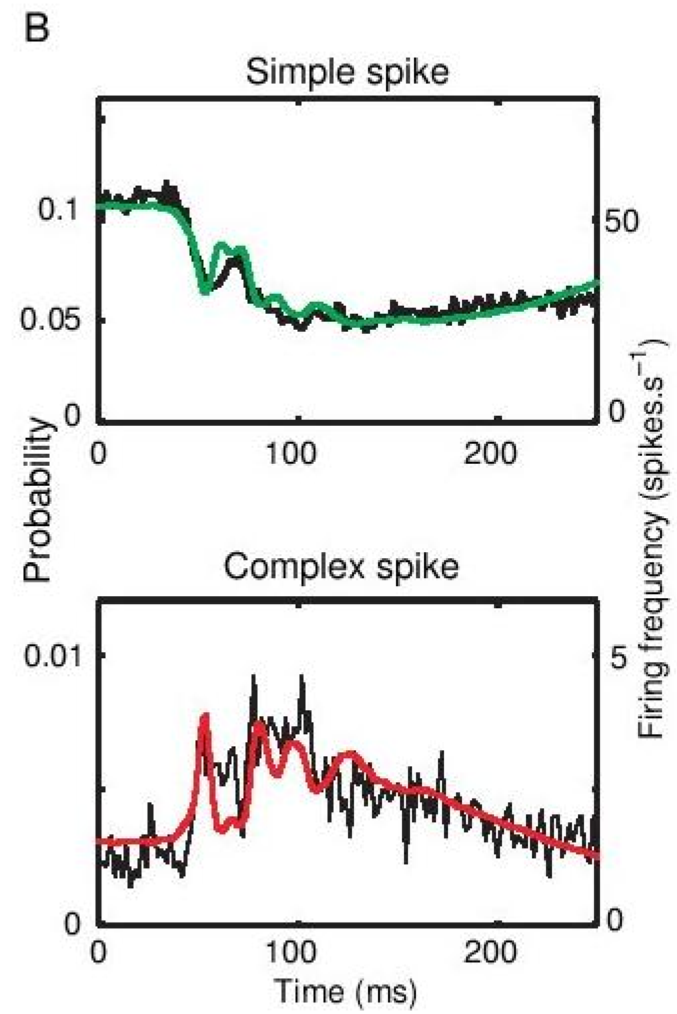
- Radial plot: angle = direction of moving stimulus.
  - U = up, D = down, C = contralateral, I = ipsilateral



- Simple spike responses (parallel fiber inputs).
- Complex spike responses (climbing fiber input).

Modeling Purkinje Cell Responses

- Model used linear combination of eye acceleration, velocity, and position.
- Quantities were measured 10 ms after simple spike measurement (accounts for conduction delay).
- Good fit for Purkinje cells in VPFL.
- So VPFL may be the inverse model for ocular following response.
- Not so good fit for neurons in MST or DLPN, which provide the input to VPFL. Do they encode trajectories (input to inverse model)?



What Do The Input Fibers Encode?

Parallel fibers:

- Eye movements : motor representation
- Retinal slip: sensory representation

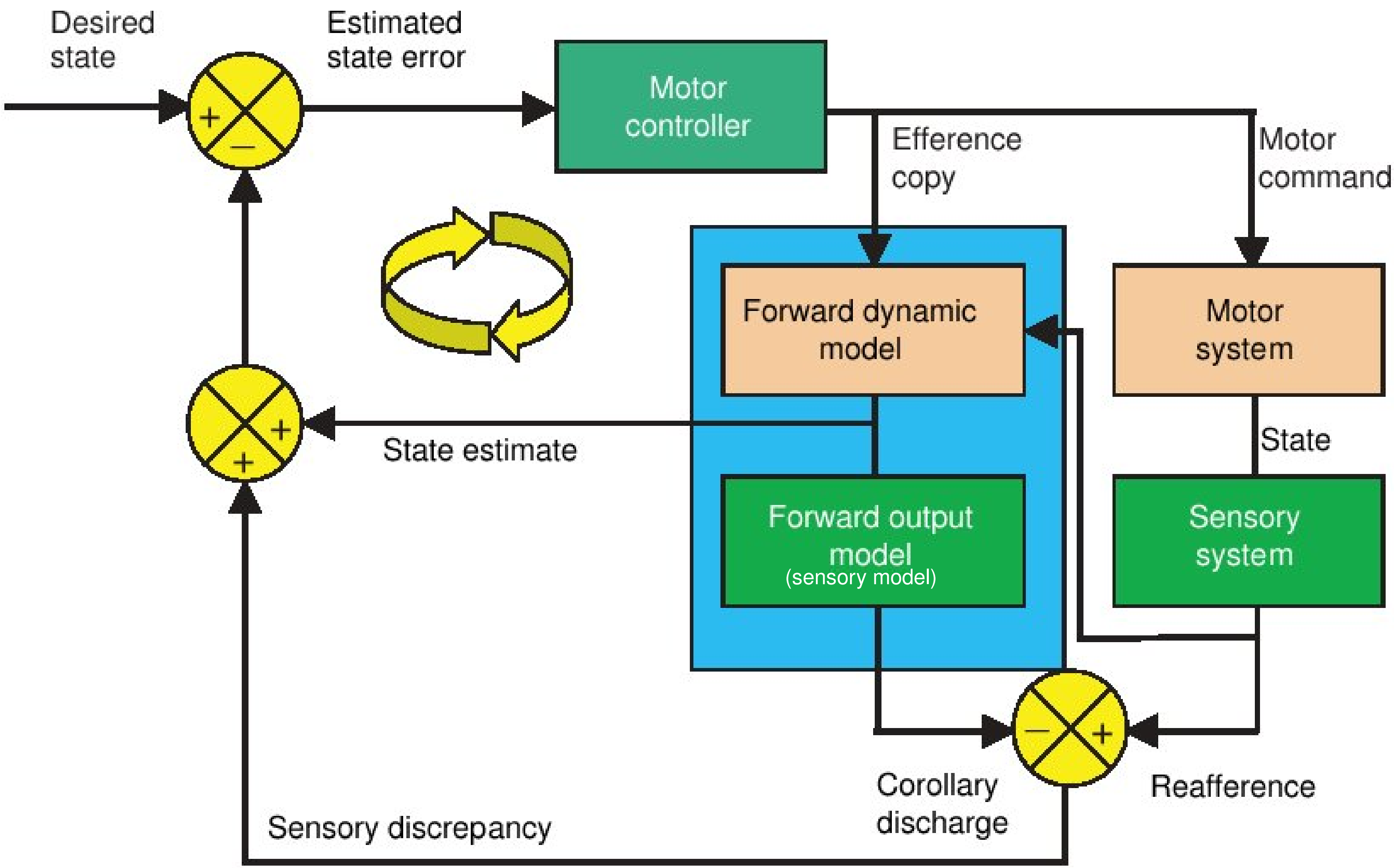
Climbing fibers

- Motor error?

Forward Models in the Cerebellum?

- Why are forward models useful here?
  - Sensory feedback has long time delays, so ...
  - A forward model can allow us to make faster corrections.
- A Smith predictor is a type of controller useful when there are delays in:
  - Sensory processing
  - Sensory-motor coupling
  - Motor execution
- The Smith predictor has <u>two</u> forward models:
  - Forward dynamic model predicts future state of the plant
  - Forward output model predicts future delayed sensory inputs
- Wolpert proposes that the forward dynamic model has a faster adaptation rate than the forward output model.

Smith Predictor Model



Arguments for Multiple Controllers in Cerebellum

1. Human motor behavior is rich and complex.
   - Unreasonable to expect everything to be captured by a single inverse or forward model.

2. Assigning different behaviors to different modules allows them to be learned independently, avoiding mutual interference.

3. If we have multiple controllers, we can take weighted combinations of them to synthesize new control regimes.
   - Controllers could serve as motor primitives.

4. Prism glasses de-adaptation and re-adaptation are faster than adaptation, suggesting that there is switching going on.

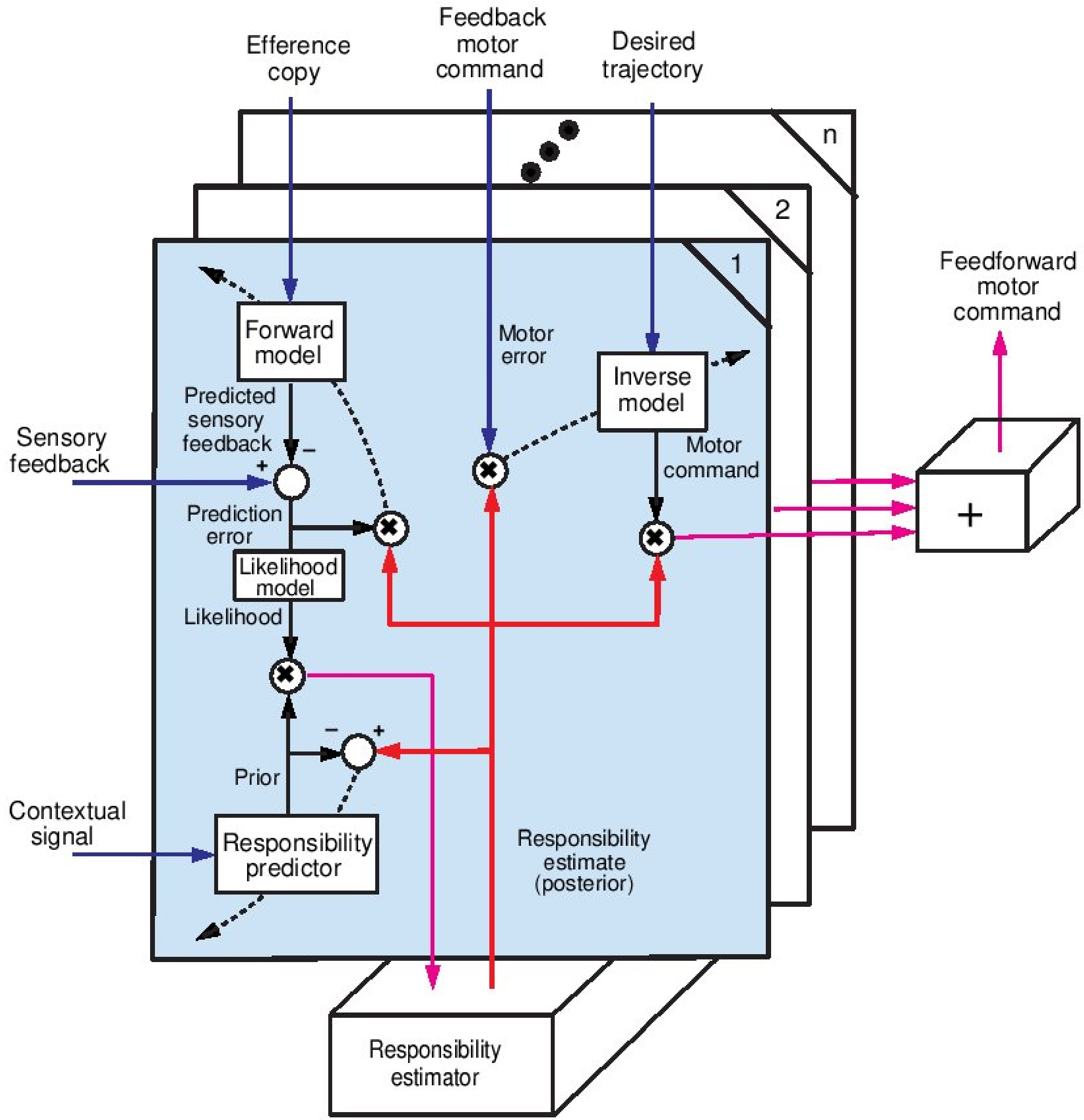
But how do we decide which model(s) to apply?

Multiple Paired Forward and Inverse Models?

Inverse model specialized for a particular behavioral context.

Forward models help determine “responsibility” for their associated inverse model in the current context, based on the goodness of their sensory predictions.

Prior estimate comes from a separate responsibility predictor.



Summary

- Biological motor control is difficult due to sensory and motor delays, and complex dynamics of the plant.

- Eye movement is a good control problem to study because it's relatively simple compared to reaching tasks.
  - But there are actually several types of eye movements:
    OFR, VOR, saccades, …

- We know that cerebellum learns, but <u>what</u> is it learning?
  - Inverse model? Forward model? Something else?

- Cerebellar circuitry appears to be uniform throughout. So how does this theory account for cerebellar contributions to:
  - Motion planning (cerebrocerebellum)
  - Classical conditioning (timing of responses)
  - Cognitive phenomena, including language tasks

Essential Concepts

- A **feedback error controller** generates control signals based on the difference between the current state and a desired state.
- An **inverse model** suggests control signals that should produce a specified desired state.
- A **forward model** predicts future state based on current control signals.
- Feedback control doesn’t work well when there is a long delay in the feedback signal. To accommodate this, we can use forward models to *predict* the feedback: either future state or future sensory signals, or both (Smith predictor).