Neural circuits for cognition

MIT 9.49/9.490/6.S076

Instructor: Professor Ila Fiete
TA: Gregg Heller
Senior Instruction Assistant: Adnan Rebei

Today

- Analysis of symmetric linear networks
- Oculomotor control, oculomotor integrator circuit
- Oculomotor integrator circuit model: a highly tuned linear network

Brief review from last class

The rate-based network equation (vector-matrix form)

$$\frac{ds_i}{dt} + \frac{s_i}{\tau} = f(\sum_j W_{ij} s_i + b_i(t))$$

$$\frac{d\mathbf{s}}{dt} + \frac{\mathbf{s}}{\tau} = f(\mathbf{W}\mathbf{s} + \mathbf{b})$$

Linearizing the network equations about a point

$$\frac{ds_i}{dt} + \frac{s_i}{\tau} = f(\sum_j W_{ij} s_j + b_i)$$

Linearized dynamics in the vicinity of some state $\bar{\mathbf{S}}$: $\mathbf{s} = \bar{\mathbf{s}} + \delta \mathbf{s}$

$$\frac{d\delta s_i}{dt} + \frac{\delta s_i}{\tau} = (\frac{\partial f}{\partial g_i}\bigg|_{\bar{\mathbf{s}}}) \sum_j W_{ij} \delta s_j$$

$$\frac{d\delta \mathbf{s}}{dt} + \frac{\delta \mathbf{s}}{\tau} = \mathbf{D}\mathbf{W}\delta \mathbf{s}$$

$$\mathbf{D}_{ij} = (\frac{\partial f}{\partial g_i}\bigg|_{\bar{\mathbf{s}}})\delta_{ij}$$

Linear and linearized neural networks

Linearized dynamics of a *nonlinear neural network* around a point $\bar{\mathbf{s}}$:

$$\frac{d\delta\mathbf{s}}{dt} + \frac{\delta\mathbf{s}}{\tau} = \mathbf{D}\mathbf{W}\delta\mathbf{s}$$

$$\mathbf{D}_{ij} = \left(\frac{\partial f}{\partial g_i}\bigg|_{\bar{\mathbf{s}}}\right)\delta_{ij}$$

A linear neural network:

$$\tau\frac{d\mathbf{s}}{dt} + \mathbf{s} = \mathbf{W}\mathbf{s} + \mathbf{b}$$

$$(\mathbf{D} = \mathbb{I})$$

Why do we care about linear(ized) networks?

- Any network's dynamics can be approximated as linear if we want to analyze its properties very locally around some point. Tool: linearize the network using Taylor expansion.
- Some networks might closely approximate linear networks: e.g. the oculomotor integrator network in the brain, and also some theoretical models.
- Piecewise-linear neurons (e.g. ReLUs) have piecewise-linear dynamics.
- Linear networks can exhibit rich behaviors that are simpler to analyze.

Linear and linearized networks, relationship to linear systems

Linear(ized) dynamical system fixed points correspond to the roots of corresponding linear systems

Fixed points of $$\frac{d\mathbf{x}}{dt} = W\mathbf{x}$$

$$\Updownarrow$$

Solutions of $$W\mathbf{x} = 0$$

Linear systems review

n equations in *m* unknowns *(v_1,…v_m):*

$$a_{11}v_1 + \cdots + a_{1m}v_m = b_1$$
$$a_{21}v_1 + \cdots + a_{2m}v_m = b_2$$
$$\cdots\cdots\cdots$$
$$a_{n1}v_1 + \cdots + a_{nm}v_m = b_n$$

$$\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1m} \\ a_{21} & a_{22} & \cdots & a_{2m} \\ \cdots & \cdots & \cdots & \cdots \\ a_{n1} & a_{n2} & \cdots & a_{nm} \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_m \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix}$$

(n x m) *(m x 1)* *(n x 1)*

$$A\mathbf{v} = \mathbf{b}$$

System of equations: when does unique solution exist?

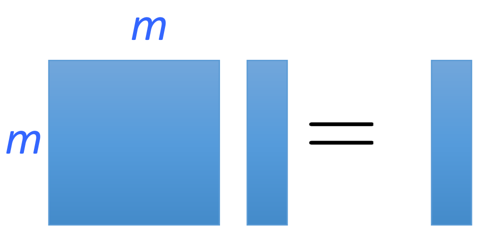
n equations (constraints) in *m* unknowns: *generically (though not exactly always!)*, a unique solution exists when, *n=m or A* is square.

$$\begin{bmatrix} a_{11} & \cdots & a_{1m} \\ a_{21} & \cdots & a_{2m} \\ \cdots & \cdots & \cdots \\ a_{m1} & \cdots & a_{mm} \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_m \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{bmatrix}$$

(m x m) *(m x 1)* *(n x 1)*

$$A\mathbf{v} = \mathbf{b}$$

(m x m) (m x 1) *(m x 1)*



For a square matrix, when is a unique solution guaranteed to exist? Time for some geometric insight.

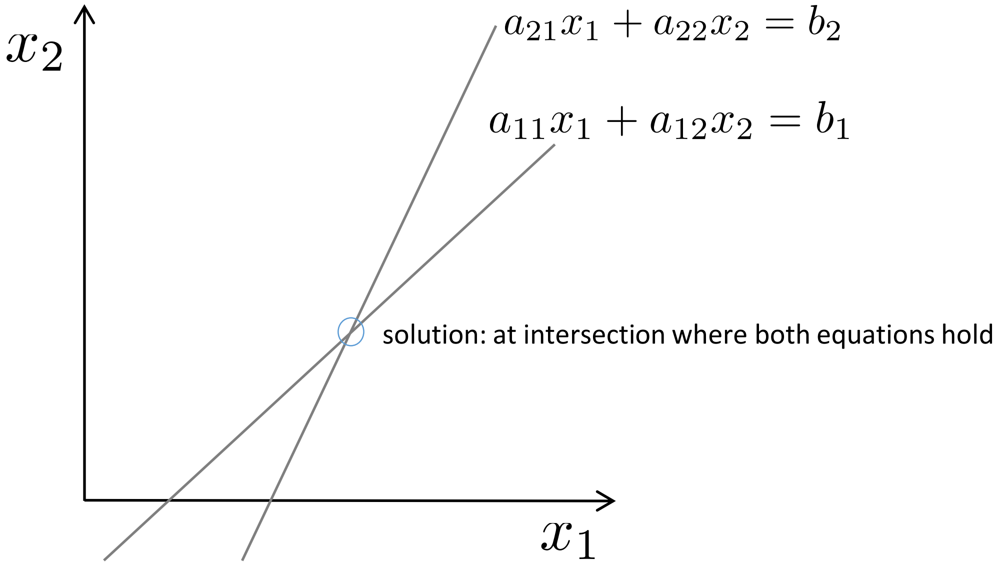
Geometric view: when does a unique solution exist?

E.g. 2-dimensional problem: 2 unknowns, 2 equations

equation of a line

$$a_{11}x_1 + a_{12}x_2 = b_1$$
$$a_{21}x_1 + a_{22}x_2 = b_2$$

unknowns x_1, x_2



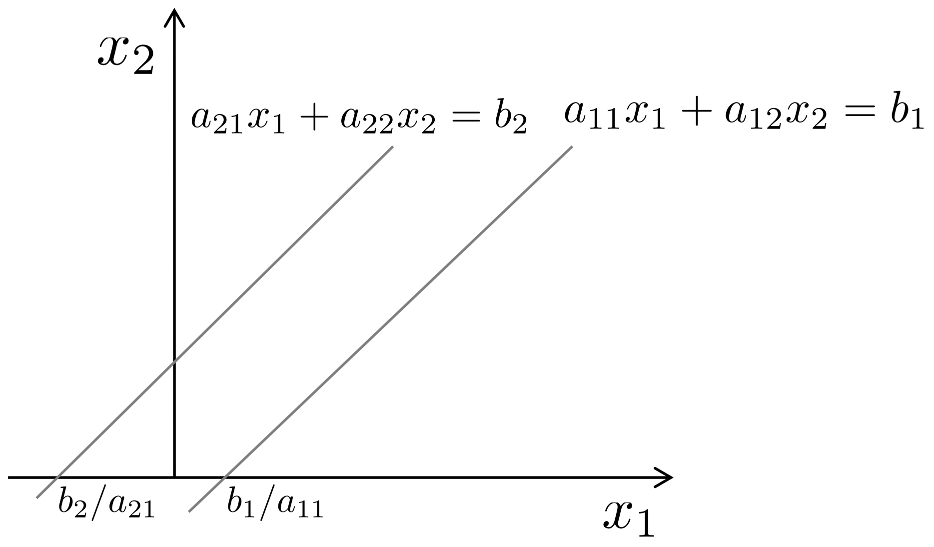
Two lines in 2D *generically* intersect at a (single) location thus generically a unique solution exists.

Geometric view: Two ways that a unique solution *does not* exist in 2D

What are these?

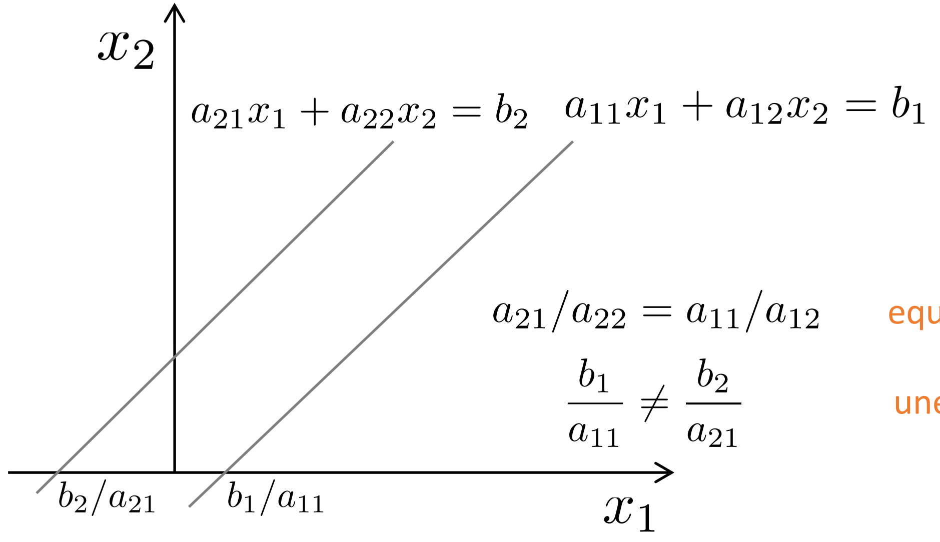
Geometric view: Two ways a unique solution *does not* exist in 2D

1. Offset parallel lines: **no solution**



Algebra: when does a unique solution *not* exist?

1. Offset parallel lines: **no solution**



$$a_{21}/a_{22} = a_{11}/a_{12}$$ equal slopes

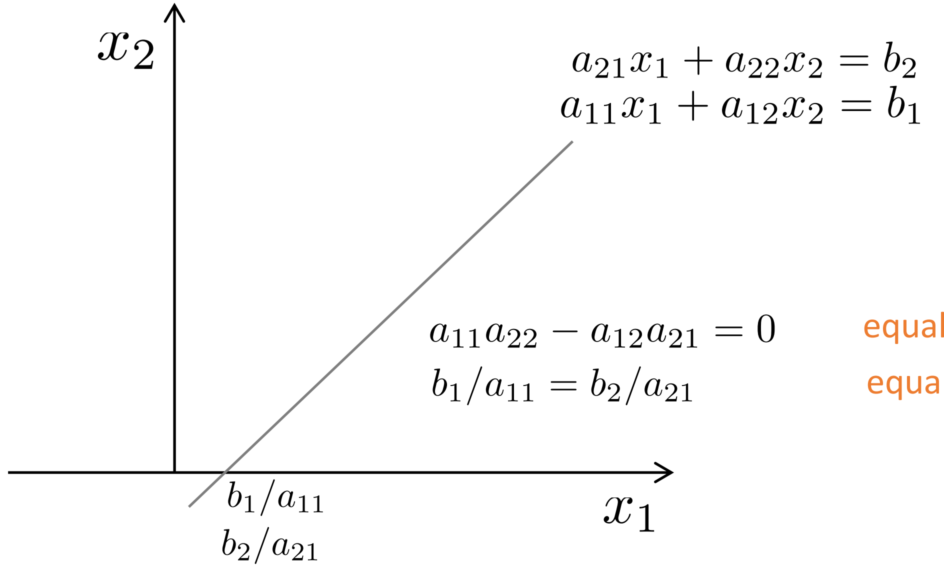
$$\frac{b_1}{a_{11}} \neq \frac{b_2}{a_{21}}$$ unequal intercepts

$$a_{11}a_{22} - a_{12}a_{21} = 0$$

$$\det(A) \equiv a_{11}a_{22} - a_{12}a_{21}$$

Algebra: when does a unique solution *not* exist?

2. Aligned parallel lines**: infinitely many solutions**



$a_{11}a_{22} - a_{12}a_{21} = 0$ equal slopes

$b_1/a_{11} = b_2/a_{21}$ equal intercepts

Back to algebraic view: existence of unique solution in terms of coefficient matrix *A*

$$A = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}$$

determinant: $$\det(A) \equiv a_{11}a_{22} - a_{12}a_{21}$$

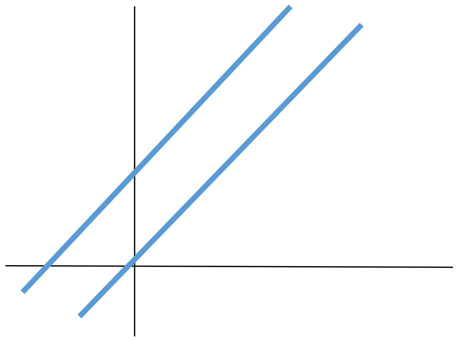
2-dim system of equations with square coefficient matrix *A* has a unique solution when:

$$\det(A) \neq 0$$

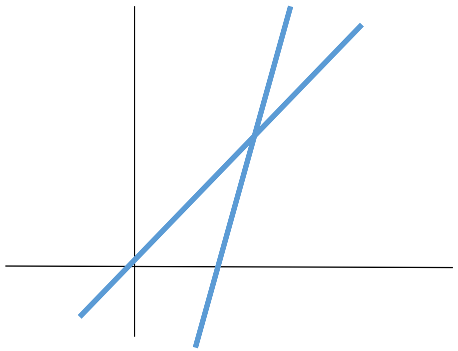
Same condition for *m*-dim system of equations with square coefficient matrix: need non-singular determinant.

Fixed points of any linear dynamical system

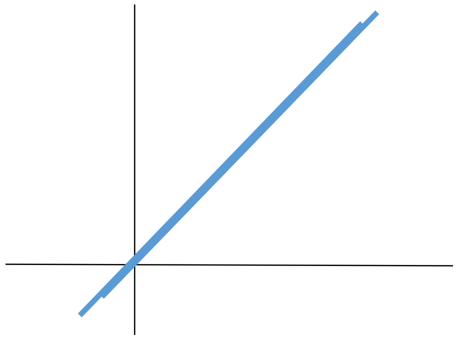
- A linear system admits exactly 0, 1, or infinitely many fixed points.



0 solutions
NOT generic



1 solution
(generic case)
square matrix, non-zero determinant

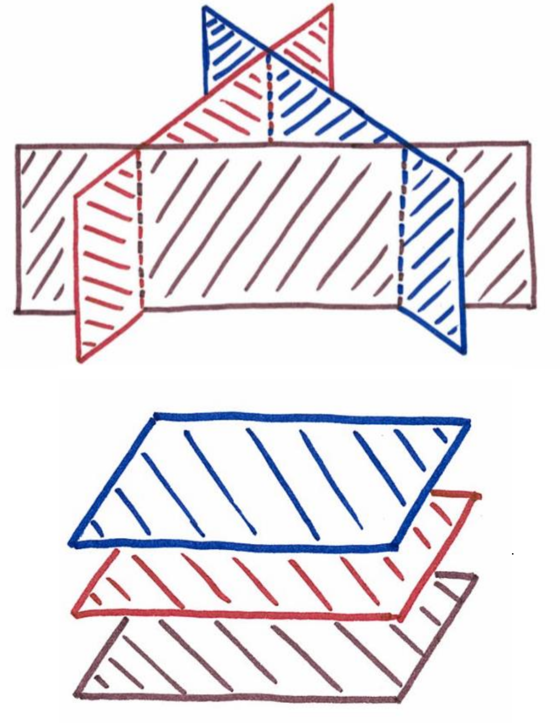


Infinitely many solutions
NOT generic

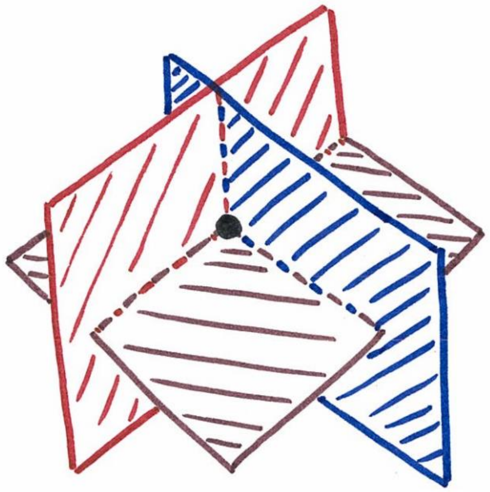
Corollary: A linear system cannot exhibit a finite number >1 of fixed points (cf. our bistable switch)

Linear dynamical systems: all possibilities

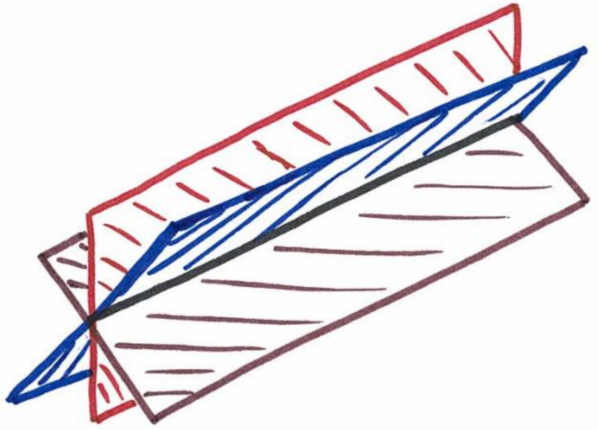
- A linear system admits 0, 1, or infinitely many fixed points.
- **Regardless of system dimension**: these are the only possibilities.



0 solutions
NOT generic



1 solution
(generic case)
square matrix, non-zero determinant



Infinitely many solutions
NOT generic

Image credit: https://www.math.utah.edu/~wortman/1050-text-lei3v.pdf

Summary

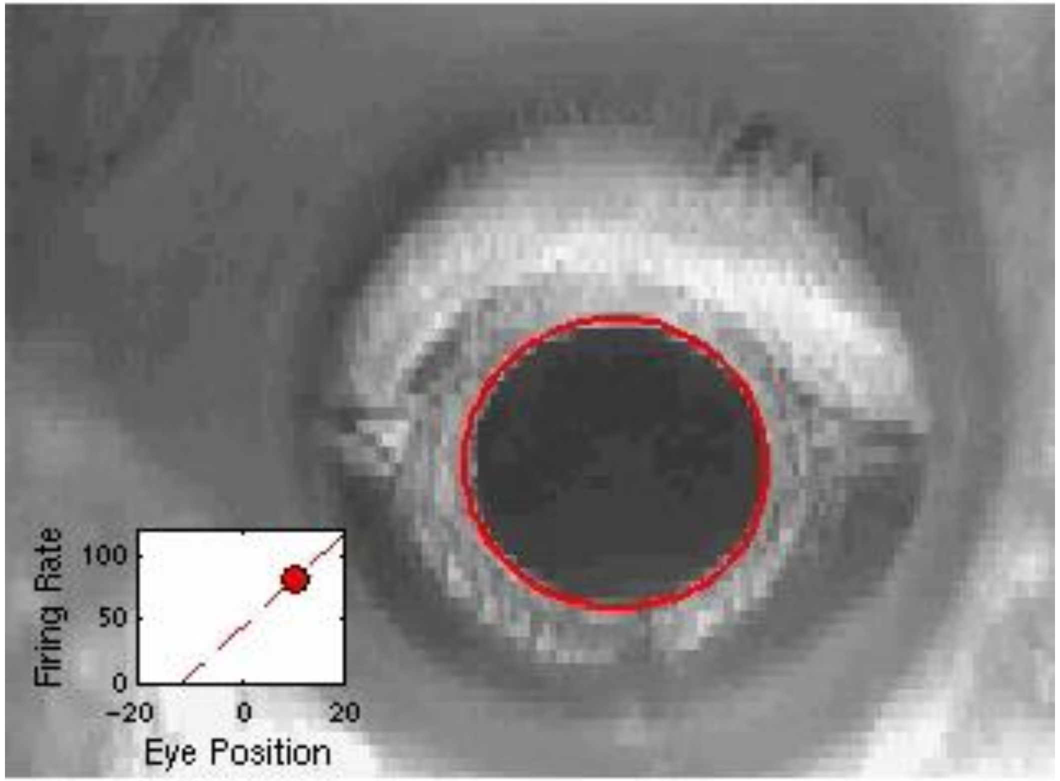
- Global and linear stability analysis
- Accelerating positive feedback + saturation → bistability
- Linear dynamical systems and relationship with linear systems of equations: fixed points of dynamical system are roots of linear system
- Linear dynamical systems admit 0,1, or infinitely many fixed points

The oculomotor integrator: highly tuned near-linear memory networks

The oculomotor integrator

The oculomotor integrator: stabilizing gaze

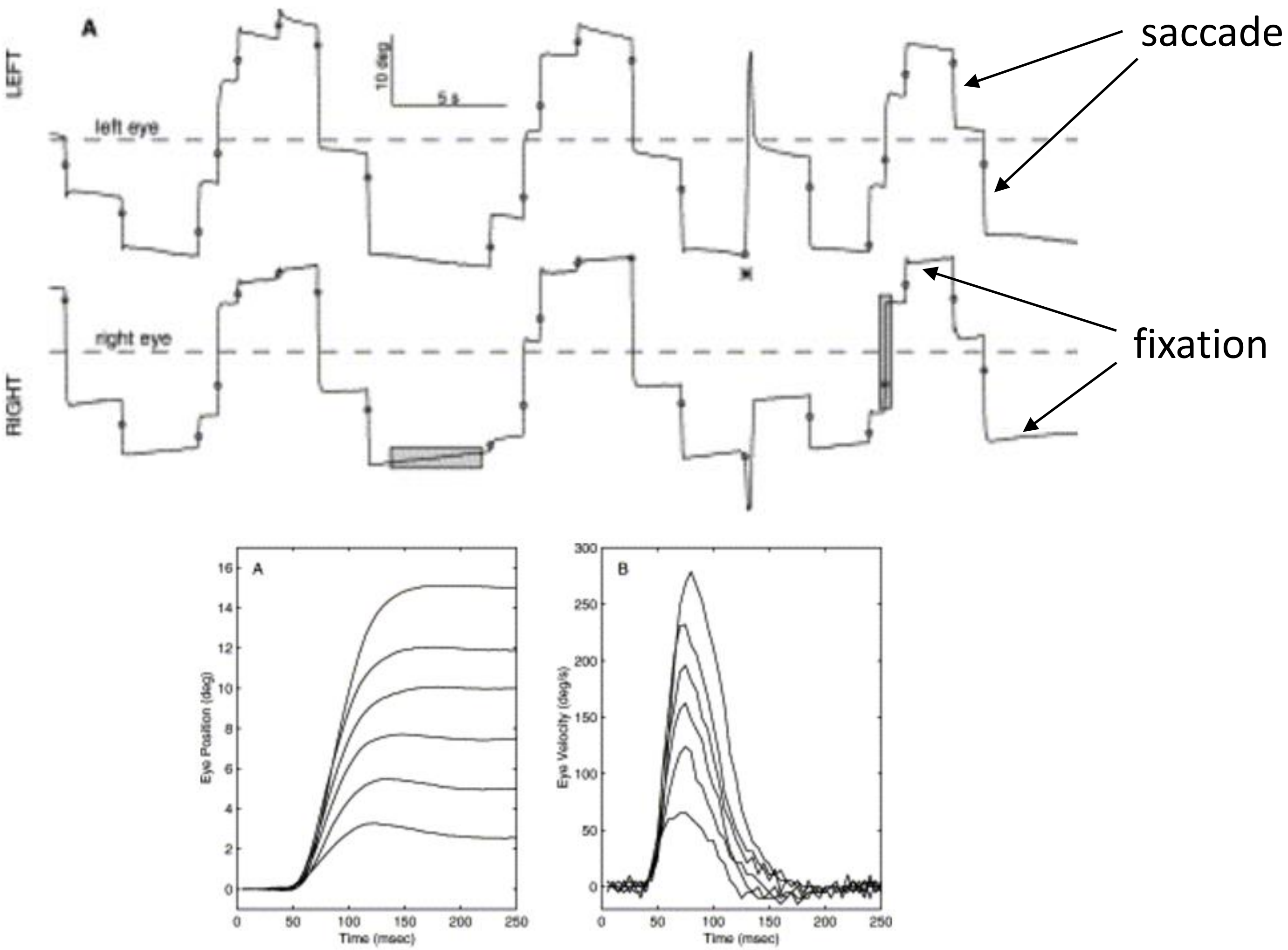
The oculomotor integrator



Body of work:
Seung, Baker, Tank, 2000’s

Ocolumotor behavior

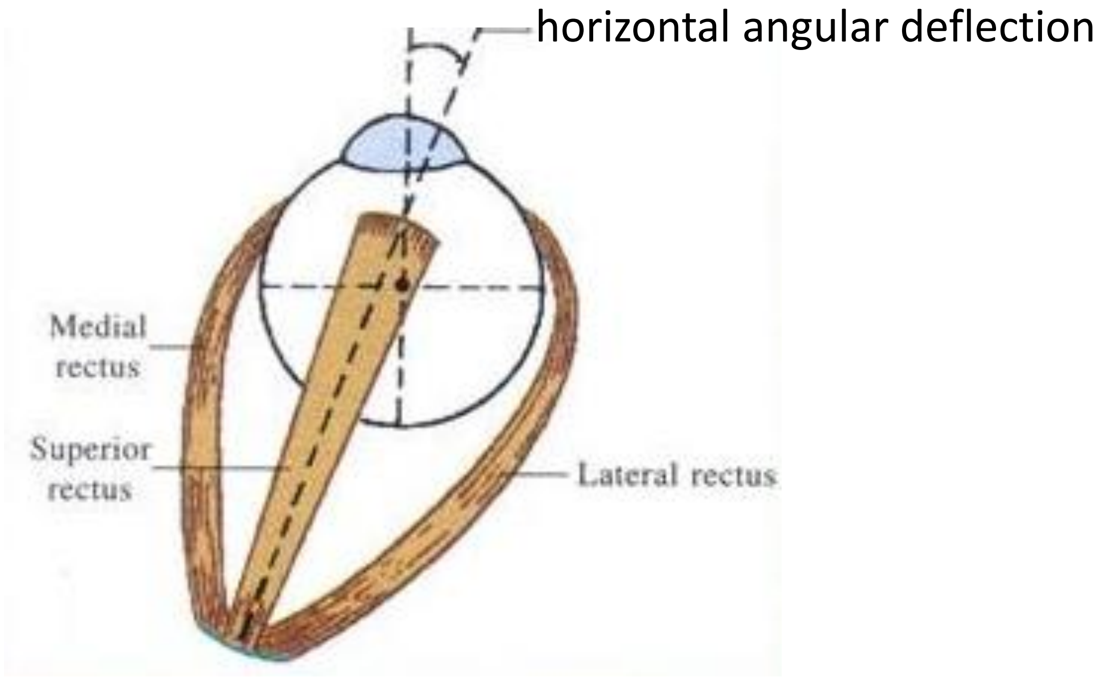
Horizontal eye position:



Mensh et al. 2004.

Oculomotor control

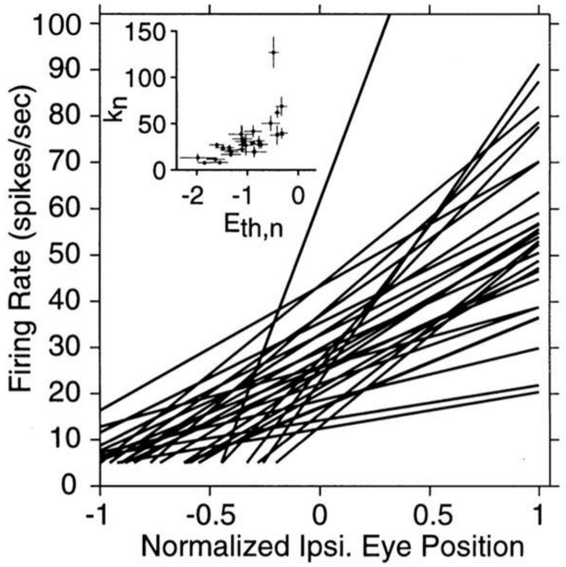
Eye muscles are (dampled) springs and require a constant drive to maintain a constant angular deflection

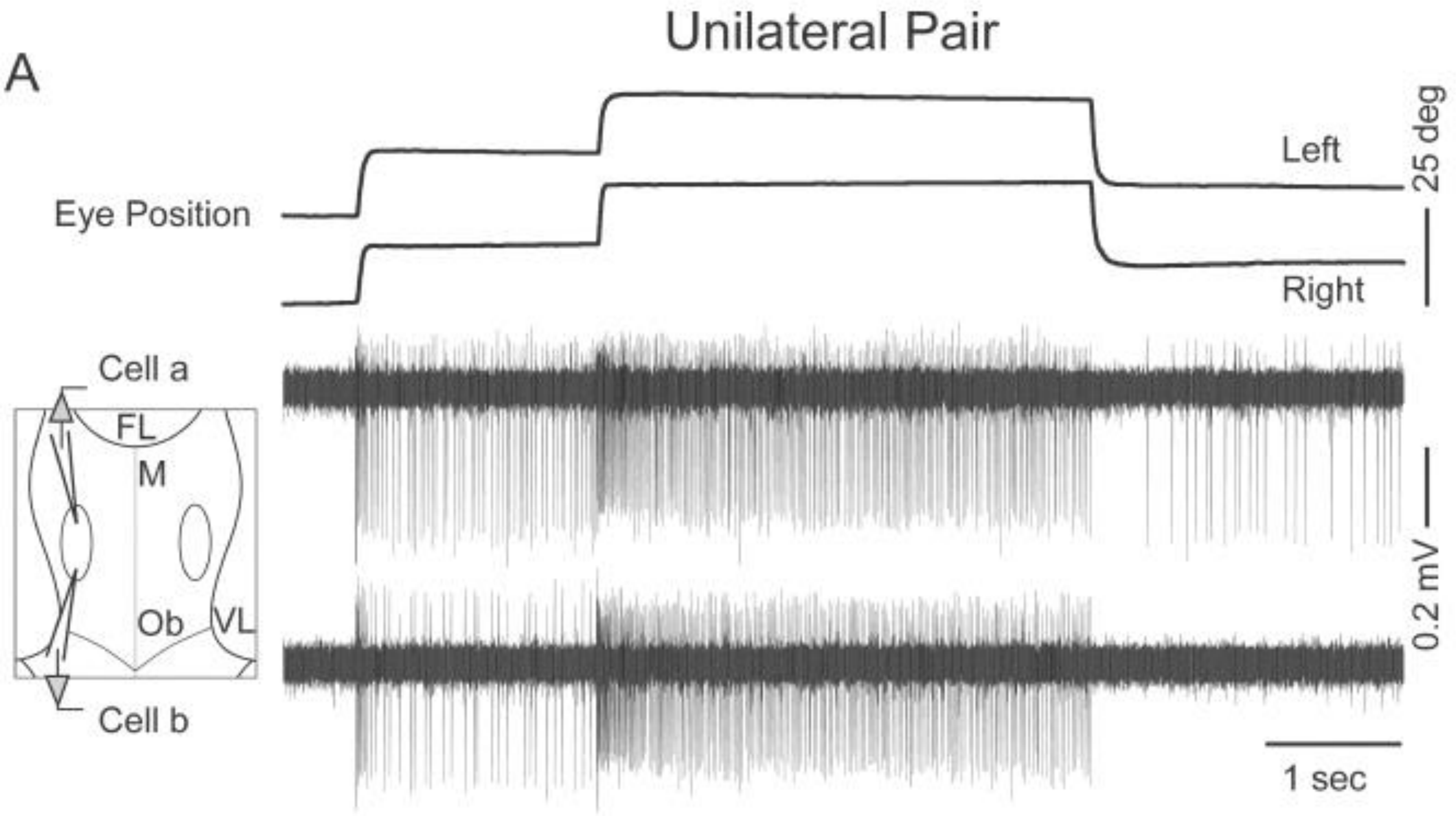


Neural drive to oculomotor muscles

Oculomotor integrator neurons provide different constant levels of drive to maintain muscle deflections for different horizontal eye positions

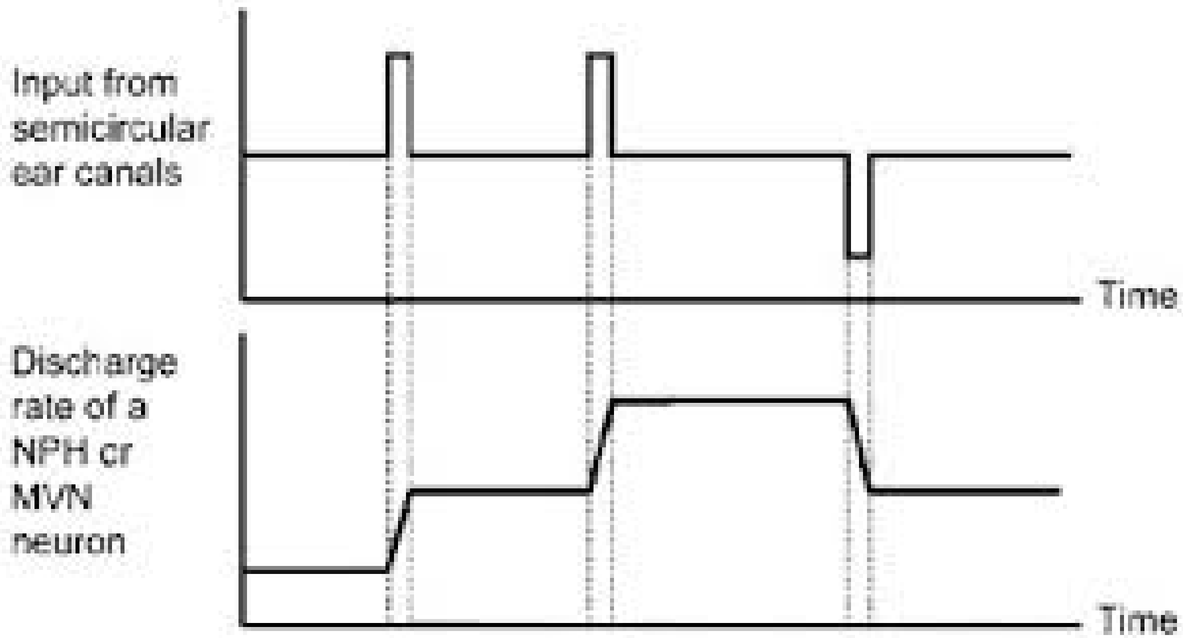
Escudero, Cheron, Godaux 1996



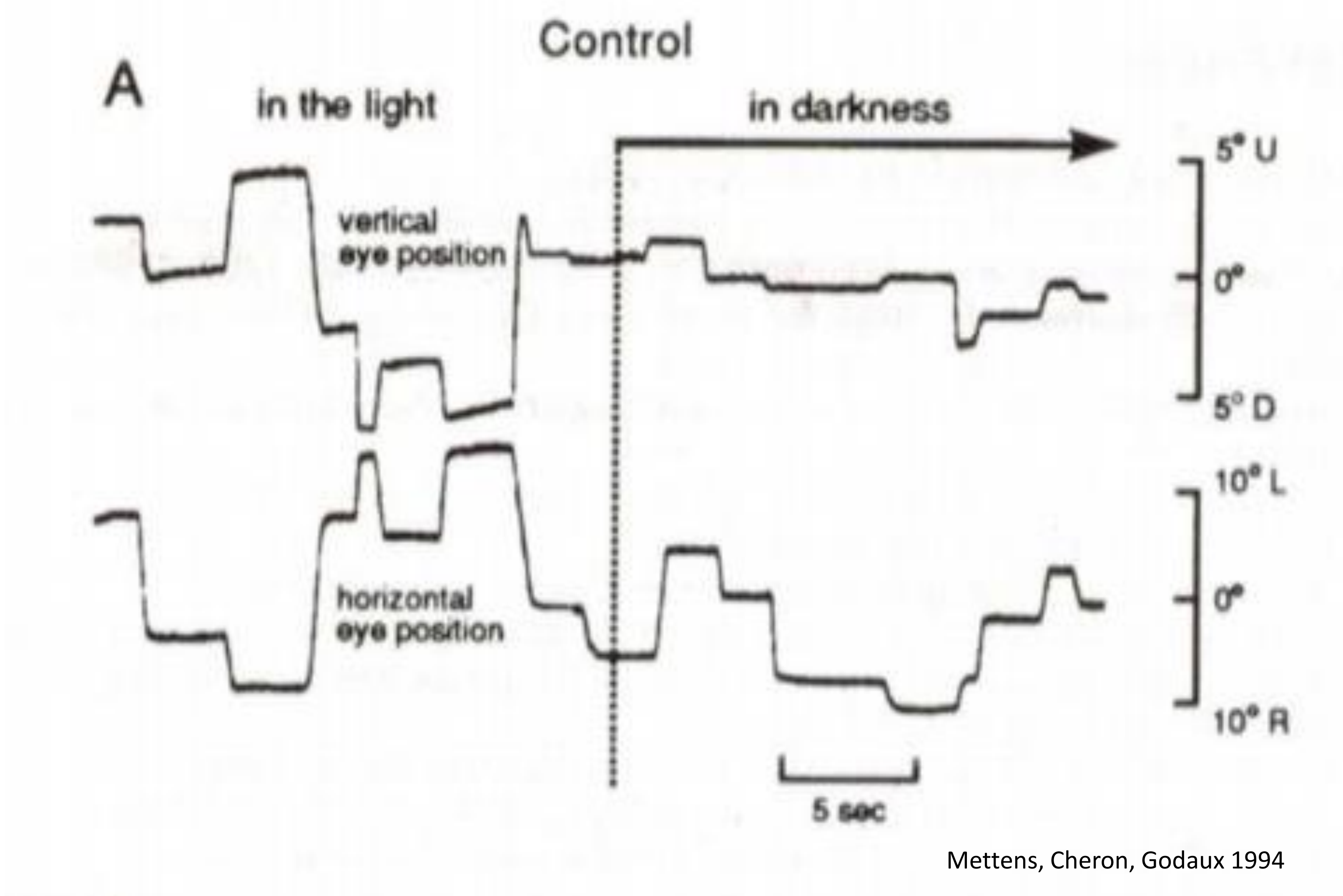


Aksay et al. 2003

The oculomotor integrator neurons receive only transient input



The oculomotor integrator supports stable eye position at different values even in the dark



Mettens, Cheron, Godaux 1994

General behavior of linear *symmetric* neural networks

Further assume that **W** is symmetric:

$$\tau \frac{d\mathbf{s}}{dt} + \mathbf{s} = \mathbf{W}\mathbf{s} + \mathbf{b}$$

Linear symmetric neural networks

$$\tau \frac{d\mathbf{s}}{dt} + \mathbf{s} = \mathbf{W}\mathbf{s} + \mathbf{b}$$

Symmetric weight matrix $\mathbf{W}$ → orthogonal eigenvectors $\mathbf{v}_\alpha$ that span the space.
Without loss of generality, assume eigenvectors are normalized.
Write time-varying state as linear combination of eigenvectors, with time-varying coefficients:

$$\mathbf{s}(t) = \sum_\beta c_\beta(t)\mathbf{v}_\beta$$

$$\rightarrow \tau \frac{d}{dt} \sum_\beta c_\beta \mathbf{v}_\beta + \sum_\beta c_\beta \mathbf{v}_\beta = \mathbf{W} \sum_\beta c_\beta \mathbf{v}_\beta + \mathbf{b}$$

This will allow us to go from N coupled equations in N variables to N uncoupled equations in a single variable each.

Linear symmetric neural networks

Use the eigenvector property $\mathbf{W}\mathbf{v}_\beta = \lambda_\beta \mathbf{v}_\beta$ and left-multiply both sides by one eigenvector, $\mathbf{v}_\alpha$ to get:

$$\mathbf{v}_\alpha^T\left(\tau \frac{d}{dt}\sum_\beta c_\beta \mathbf{v}_\beta + \sum_\beta c_\beta \mathbf{v}_\beta\right) = \mathbf{v}_\alpha^T\left(\mathbf{W}\sum_\beta c_\beta \mathbf{v}_\beta + \mathbf{b}\right)$$

$$\rightarrow \quad \tau \frac{d}{dt}\sum_\beta c_\beta \mathbf{v}_\alpha^T \mathbf{v}_\beta + \sum_\beta c_\beta \mathbf{v}_\alpha^T \mathbf{v}_\beta = \sum_\beta \lambda_\beta c_\beta \mathbf{v}_\alpha^T \mathbf{v}_\beta + \mathbf{v}_\alpha^T \mathbf{b}$$

Linear symmetric neural networks

Use the eigenvector property $\mathbf{W}\mathbf{v}_\beta = \lambda_\beta \mathbf{v}_\beta$ and left-multiply both sides by one eigenvector, $\mathbf{v}_\alpha$ to get:

$$\mathbf{v}_\alpha^T\big(\tau \frac{d}{dt}\sum_\beta c_\beta \mathbf{v}_\beta + \sum_\beta c_\beta \mathbf{v}_\beta\big) = \mathbf{v}_\alpha^T\big(\mathbf{W}\sum_\beta c_\beta \mathbf{v}_\beta + \mathbf{b}\big)$$

$$\rightarrow \quad \tau \frac{d}{dt}\sum_\beta c_\beta \mathbf{v}_\alpha^T \mathbf{v}_\beta + \sum_\beta c_\beta \mathbf{v}_\alpha^T \mathbf{v}_\beta = \sum_\beta \lambda_\beta c_\beta \mathbf{v}_\alpha^T \mathbf{v}_\beta + \mathbf{v}_\alpha^T \mathbf{b}$$

Finally, use the orthonormality property $\mathbf{v}_\alpha^T \mathbf{v}_\beta = \delta_{\alpha\beta}$ for symmetric **W**, to get the decoupled equations:

$$\tau \frac{dc_\alpha}{dt} + c_\alpha = \lambda_\alpha c_\alpha + b_\alpha,$$

Decoupled equations for the network

where $b_\alpha = \mathbf{v}_\alpha^T \mathbf{b}$ is the projection of the network input $\mathbf{b}$ onto the eigenvector ("mode") $\mathbf{v}_\alpha$ and $\mathbf{s}$ can be recomposed from its coefficients as: $\mathbf{s}(t) = \sum_\beta c_\beta(t) \mathbf{v}_\beta$

The decoupled dynamics

$$\tau \frac{d\mathbf{s}}{dt} + \mathbf{s} = \mathbf{W}\mathbf{s} + \mathbf{b}$$

N *coupled* (vector-matrix) equations for the activities of N *neurons*

$$\rightarrow \tau \frac{dc_\alpha}{dt} + c_\alpha = \lambda_\alpha c_\alpha + b_\alpha$$

N *uncoupled* (scalar) equations for the activities of N *modes*

$$\tau \frac{dc_\alpha}{dt} = -(1 - \lambda_\alpha) c_\alpha + b_\alpha$$

simple exponentials
Stability: all $\lambda_\alpha \leq 1$

I.e. all eigenvalues
of W should be <=1

Dynamics of symmetric linear networks

$$\tau \frac{dc_\alpha}{dt} = -(1-\lambda_\alpha)c_\alpha + b_\alpha$$

simple exponentials

Stability: $\lambda_\alpha \leq 1$

Complete solution for the linear symmetric network:

$$\mathbf{s}(t) = \sum_\beta c_\beta(t)\mathbf{v}_\beta$$ where

$$c_\alpha(t) = \big(c_\alpha(0) - \frac{b_\alpha}{1-\lambda_\alpha}\big)e^{-t(1-\lambda_\alpha)/\tau} + \frac{b_\alpha}{1-\lambda_\alpha}$$

$$c_\alpha(0) = \mathbf{v}_\alpha^T \mathbf{s}(0)$$

$$b_\alpha = \mathbf{v}_\alpha^T \mathbf{b}$$

s is a simple sum of simple exponentials: it exponentially decays and/or blows up along the different eigenvectors

Dynamics of general (not necessarily symmetric) linear networks

$$\frac{d\mathbf{s}}{dt} = -\mathbf{s} + \mathbf{W}\mathbf{s} = (-\mathbb{I} + \mathbf{W})\mathbf{s} \equiv \mathbf{A}\mathbf{s}$$

$$\mathbf{s}(t) = \sum_{\beta} a_{\beta} e^{\nu_{\beta} t} \mathbf{u}_{\beta} + \mathbf{s}(0)$$

$\mathbf{u}_{\beta}$ eigenvector of A
ν_{β} eigenvalue of A

Real eigenvalues:
Stability: *All* eigenvalues of A < 0 (of W < 1)
Instability: *Any* eigenvalue of A > 0 (of W > 1)
Neutral stability along a dimension:
that eigenvalue of A = 0 (of W = 1)

Complex eigenvalues: $\nu_{\beta} = p_{\beta} + iq_{\beta}$
Stability: *All real parts* of eigenvalues of A < 0 (of W < 1)
Instability: *Real part of any* eigenvalue of A > 0 (of W > 1)
Neutral stability along a dimension: that eigenvalue of A = 0
Imaginary part: leads to oscillations of frequency q_{β}

Behavior/uses of linear symmetric networks: attenuation and amplification

Recall: in symmetric network, all eigenvalues are real

Steady-state value for α th mode: $$\bar{c}_\alpha = \frac{b_\alpha}{(1-\lambda_\alpha)}$$

For $\lambda_\alpha \neq 1$

Network time-constant for α th mode: $$\tau_\alpha = \frac{\tau}{(1-\lambda_\alpha)}$$

Attenuating mode

$$\lambda < 0$$

$$\bar{c_\alpha} < b_\alpha, \tau_\alpha < \tau$$

Fast but **low-amplitude**/suppressed input response

Amplifying mode

$$0 < \lambda < 1$$

$$\bar{c_\alpha} > b_\alpha, \tau_\alpha > \tau$$

Slow but **large-amplitude**/amplified input response

Behavior/uses of linear symmetric networks: memory

If $\lambda = 1$

$$\tau_\alpha = \frac{\tau}{1 - \lambda_\alpha} \to \infty$$

Creation of a long time constant

If inputs set only initial condition (no additional input *b(t)*):

$$\tau \frac{dc_\alpha}{dt} = 0 \implies c_\alpha(t) = c_\alpha(0)$$

Perfect analog memory: remember ANY initial condition

Leaky units can together create a long-lived analog memory/persistent state

Behavior of linear symmetric networks: integration

If $\lambda = 1$ and time-varying inputs *b(t)* along that eigenmode:

$$\tau \frac{dc_\alpha}{dt} = b_\alpha \quad \rightarrow \quad c_\alpha(t) = c_\alpha(0) + \int^t b_\alpha(t')dt'$$

Perfect (non-leaky) Integration along this mode

Leaky units can together perform perfect, non-leaky integration (calculus)!

Can view analog memory as special case of integration

$$b_\alpha(t) = 0 \quad \text{over} \quad t \in [t_0, t_0 + T]$$

$$\rightarrow \; c_\alpha(t) = c_\alpha(0)$$

Analog memory:
c_α **can hold any value**

Fine-tuning for memory and integration

Leaky units can collectively perform non-leaky integration and hold analog memory.
BUT: *require fine-tuning:* $\lambda = 1$

Quantifying the degree of fine-tuning: $\tau_\alpha = \dfrac{\tau}{(1-\lambda_\alpha)} \quad \rightarrow \infty \text{ as } \lambda_\alpha \rightarrow 1$

To get a >= 200x increase in time-constant (from 50 ms to 10 s): $\lambda_\alpha = 0.995$

Parameters set to within 0.5% of the tuned value of 1.
Is this possible in biology? Do linear integrator systems exist?

Summary: different modes in a linear network

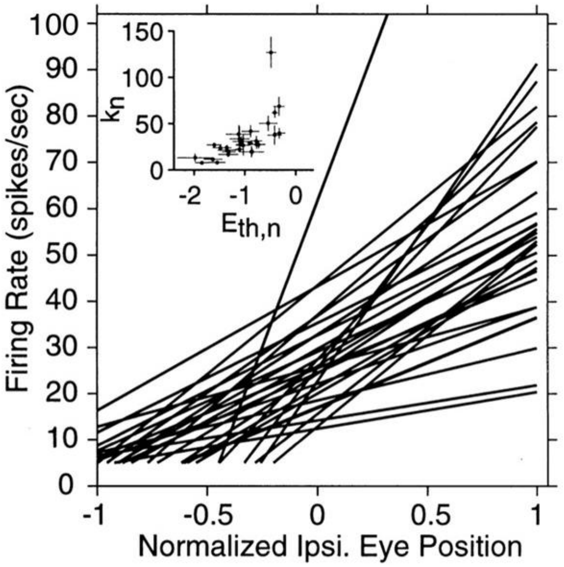
- Attenuation $\bar{c_\alpha} < b_\alpha$
  $\lambda < 0$
  $\tau_\alpha < \tau$
- Amplification $\bar{c_\alpha} > b_\alpha$
  $0 < \lambda < 1$
  $\tau_\alpha > \tau$
- Integration/memory (marginally stable) $\tau_\alpha \uparrow \infty$
  $\lambda = 1$
- Instability: activity diverges unbounded
  $\lambda_\alpha > 1$

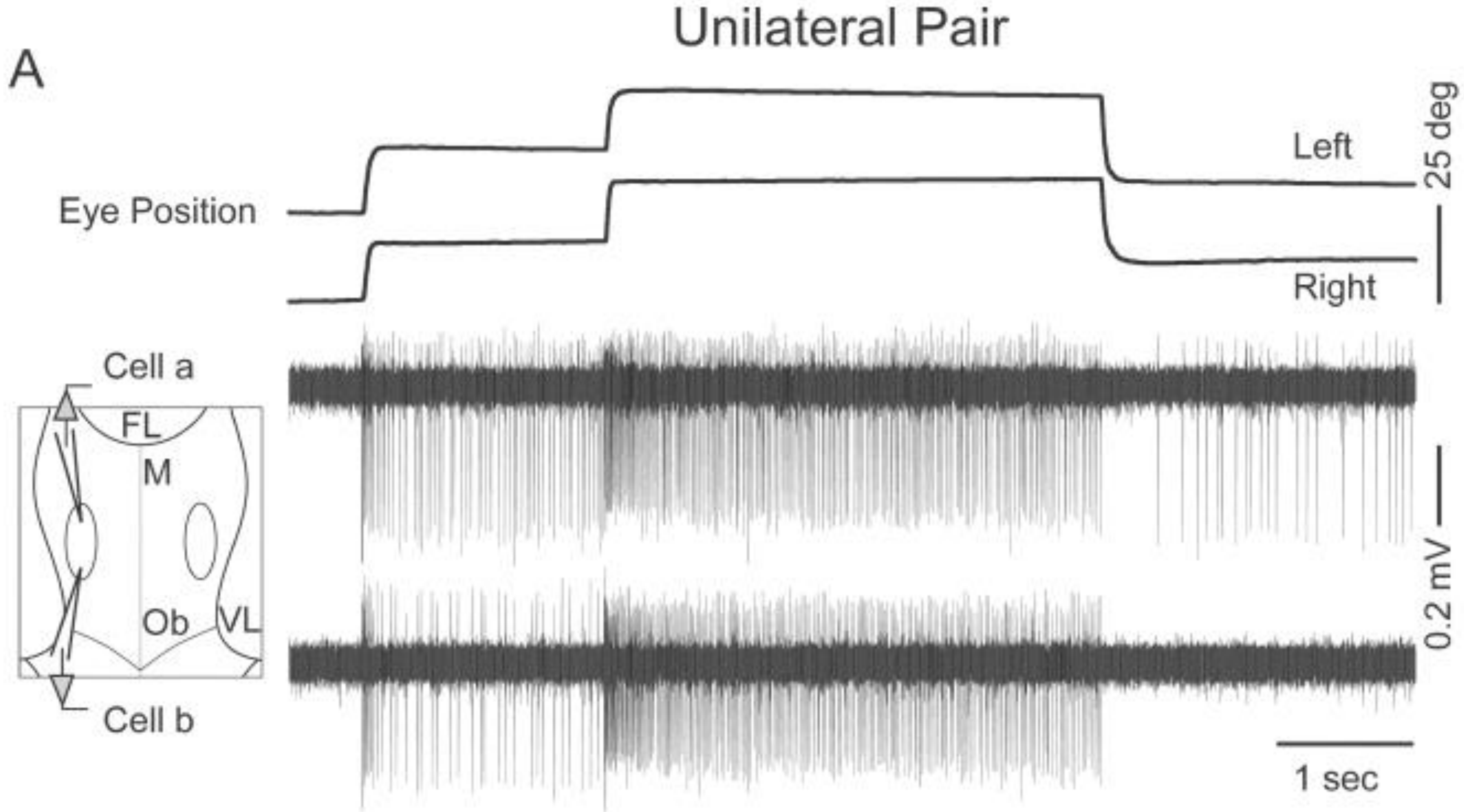
See Seung in Arbib, 2002
http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.8.5776&rep=rep1&type=pdf

Neural drive to oculomotor muscles

Oculomotor integrator neurons provide different constant levels of drive to maintain muscle deflections for different horizontal eye positions

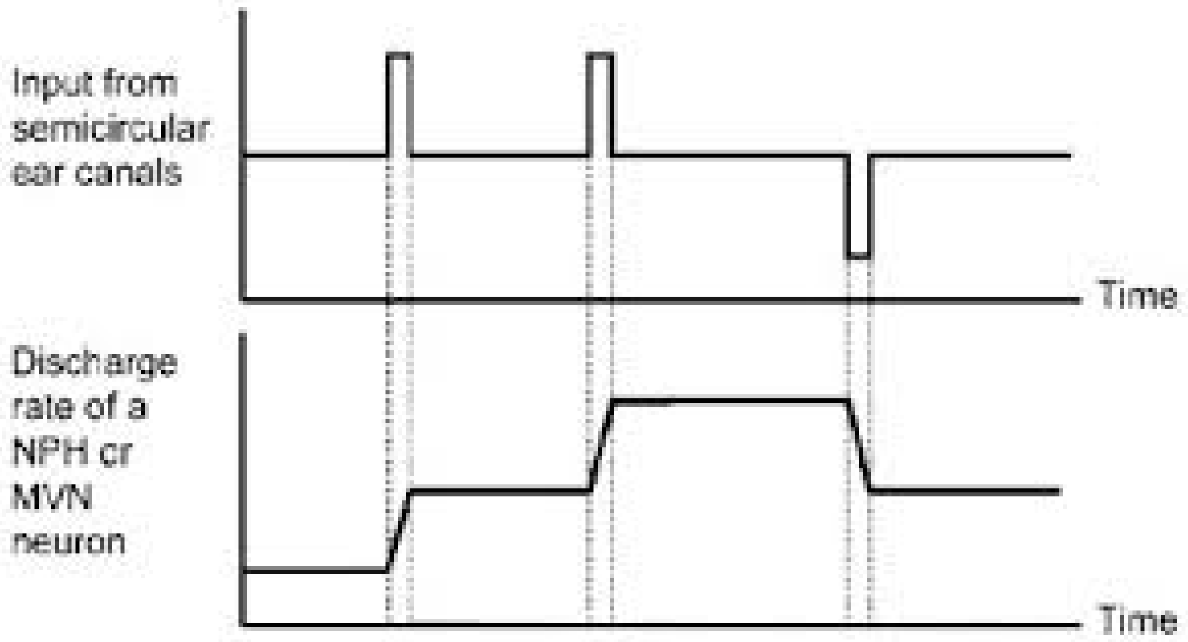
Escudero, Cheron, Godaux 1996





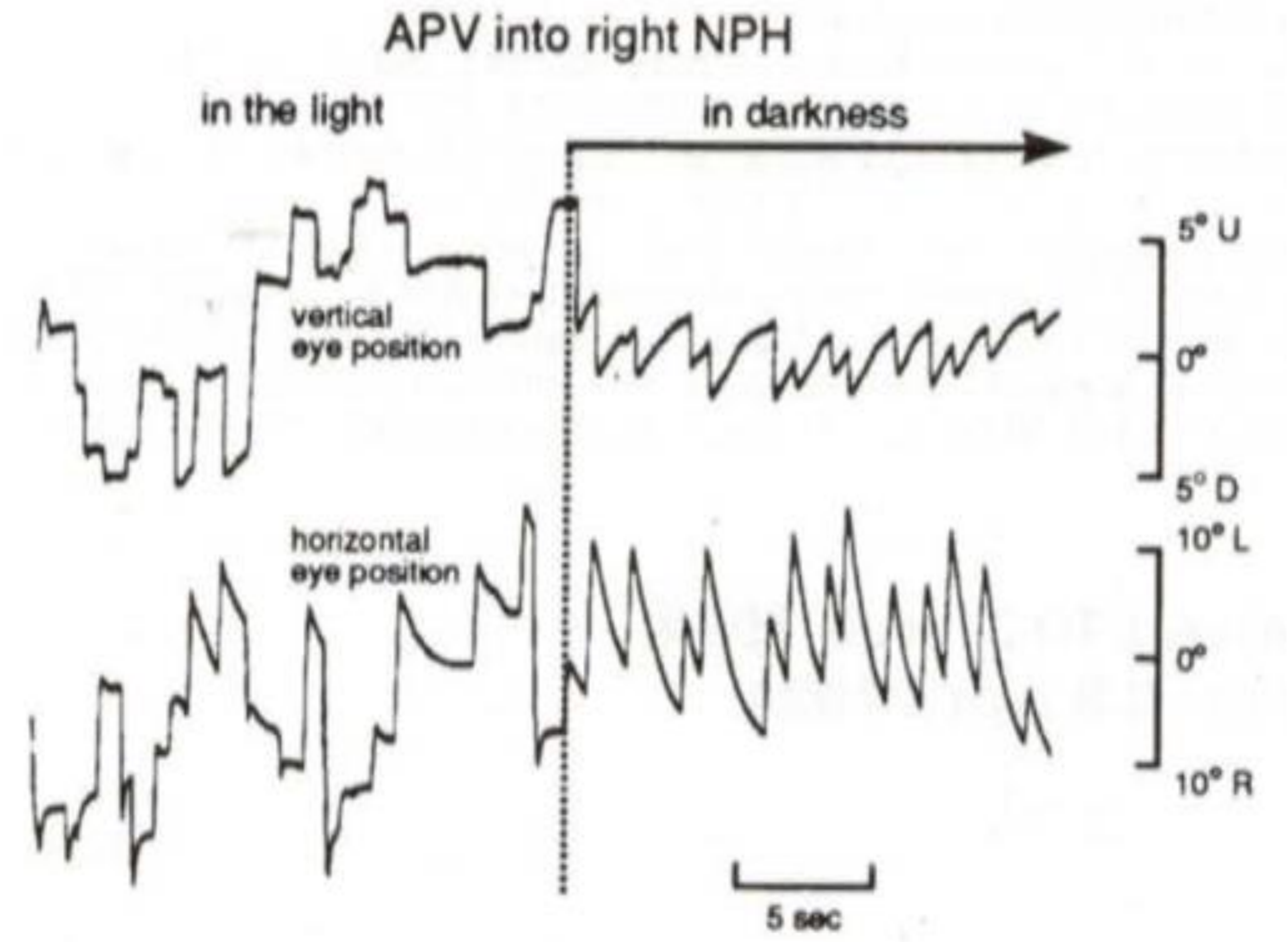
Aksay et al. 2003

The oculomotor integrator neurons receive only transient input



Integration requires synaptic feedback

Reduction of network feedback results in leaky integration

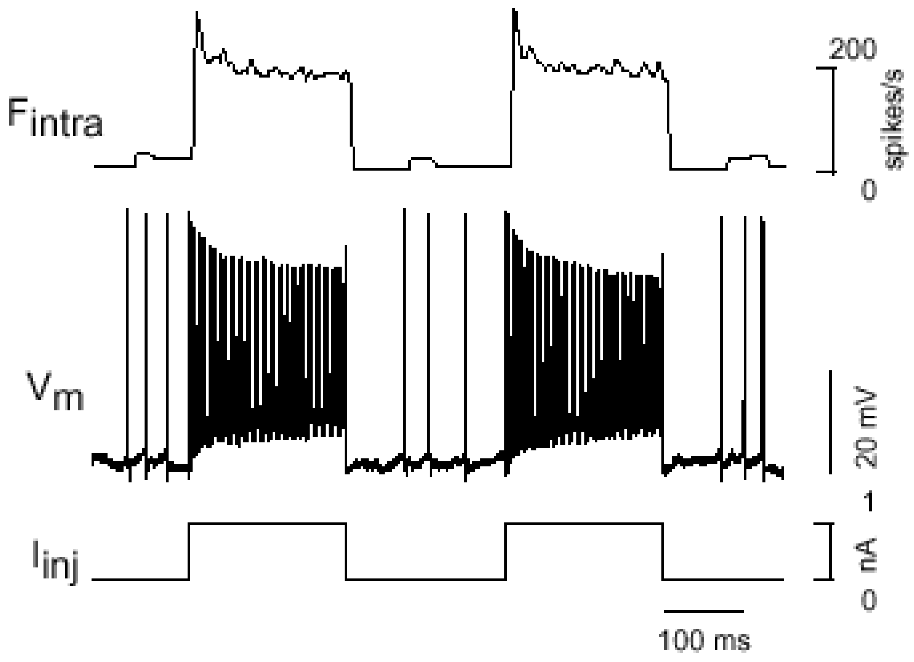


Mettens, Cheron, Godaux 1994

APV (also called AP5) is an NMDA receptor antagonist: blocks slow excitatory neurotransmission.

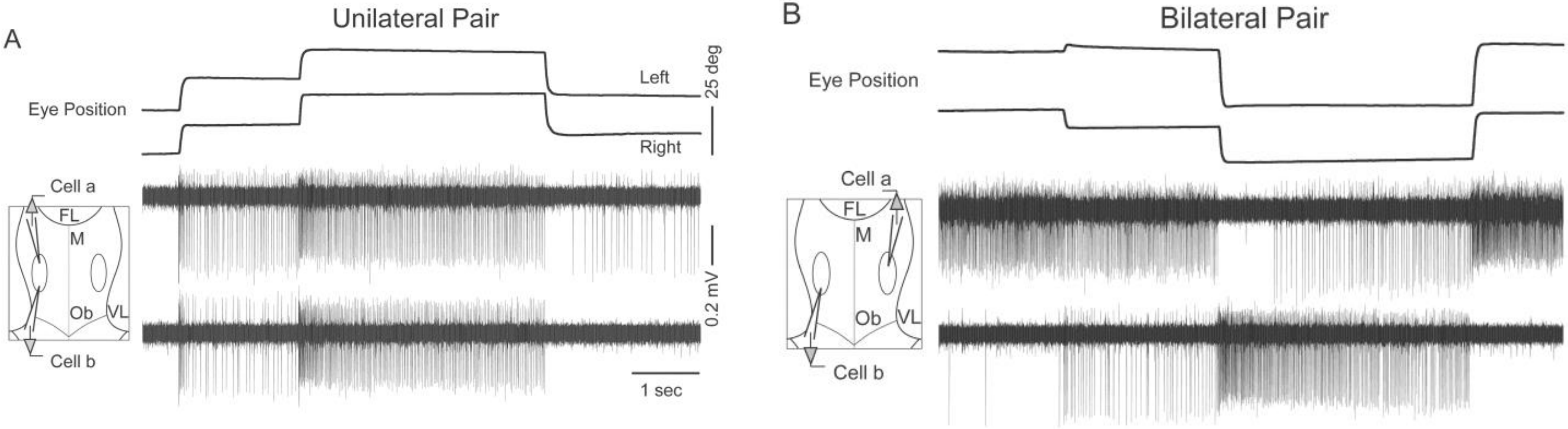
Evidence of network rather than single-cell dynamics

Perturb single cell; response NOT persistent



The oculomotor integrator neurons are bilaterally arranged

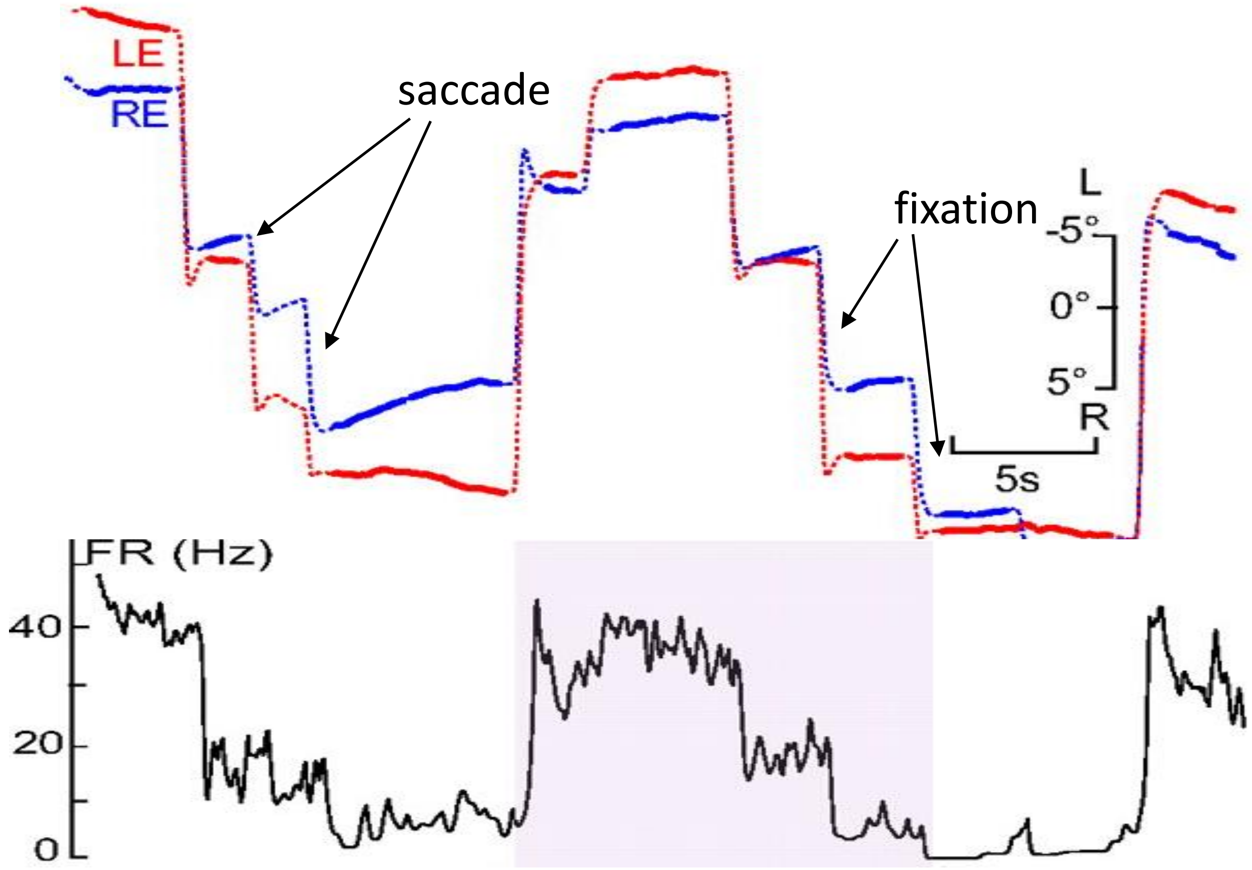
Neurons on contralateral sides do opposing things

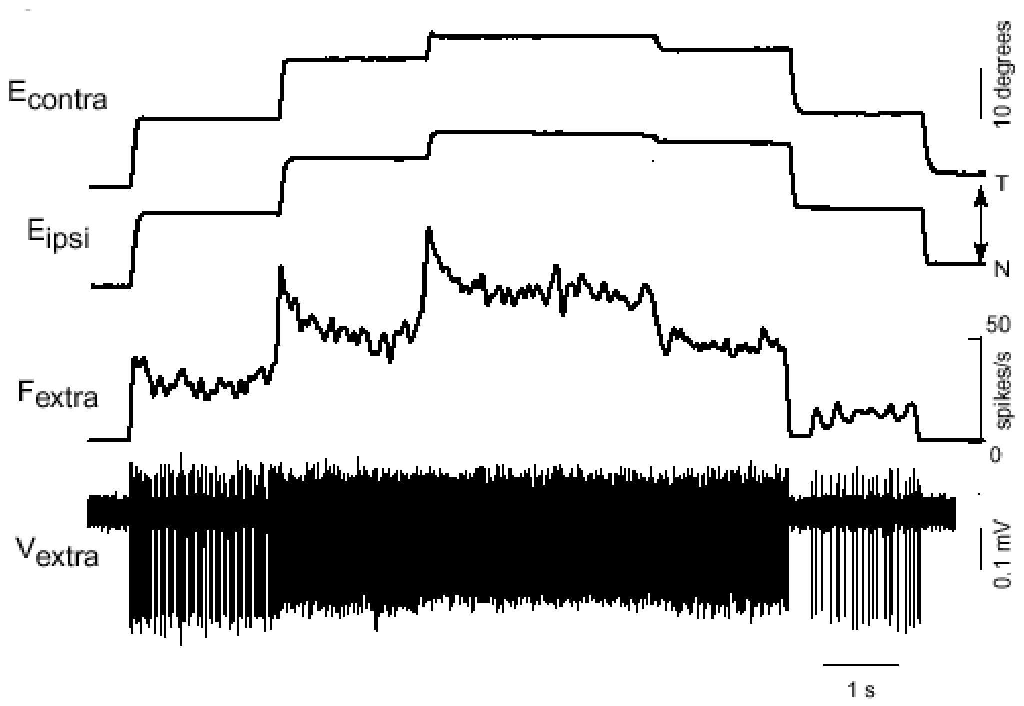


Aksay et al. 2003

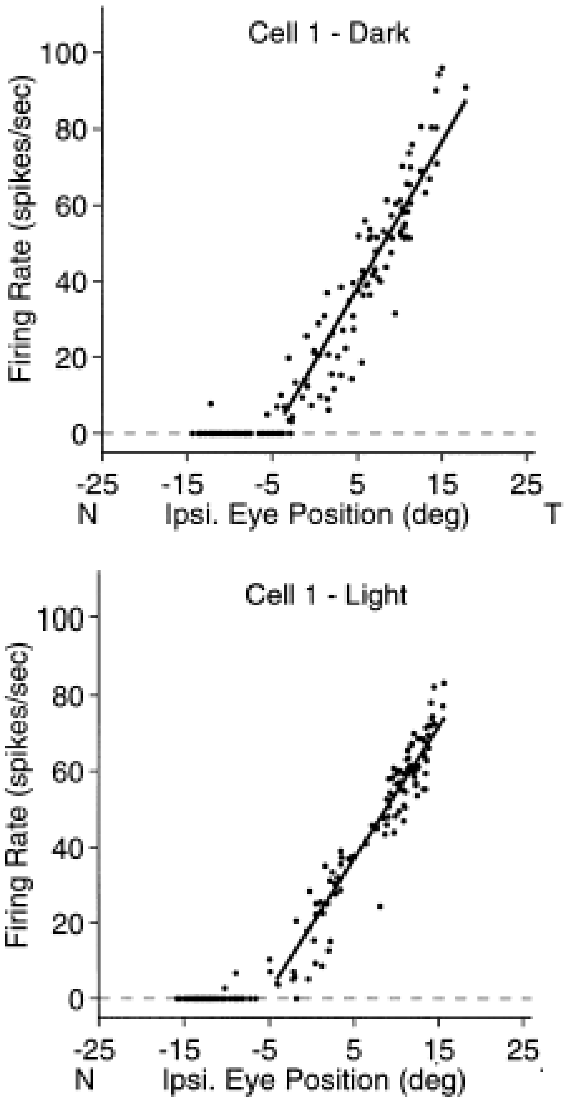
The oculomotor integrator

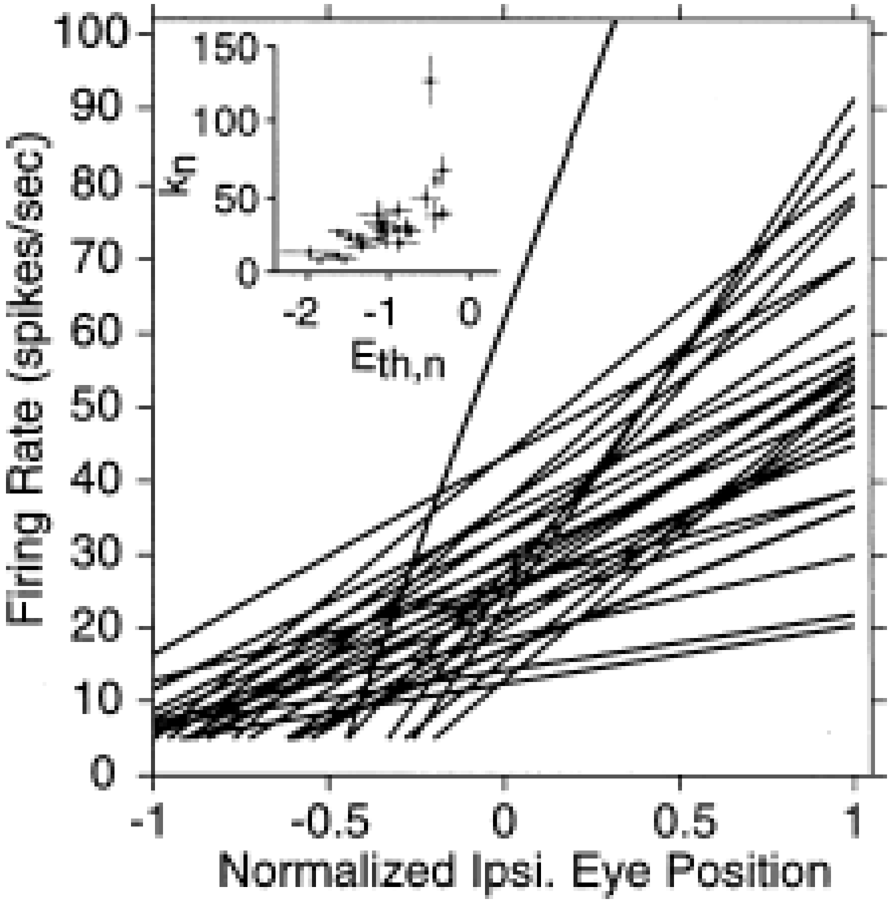
Horizontal eye position:





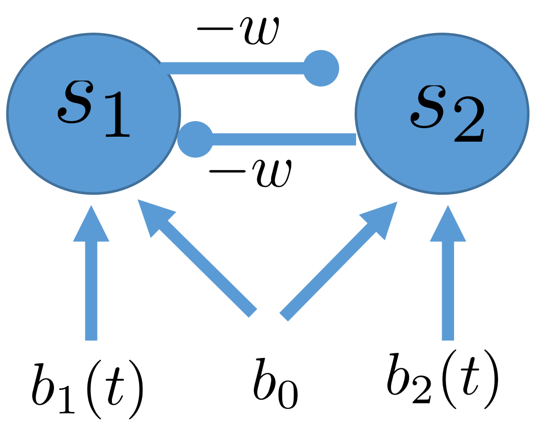
Quantification and population data



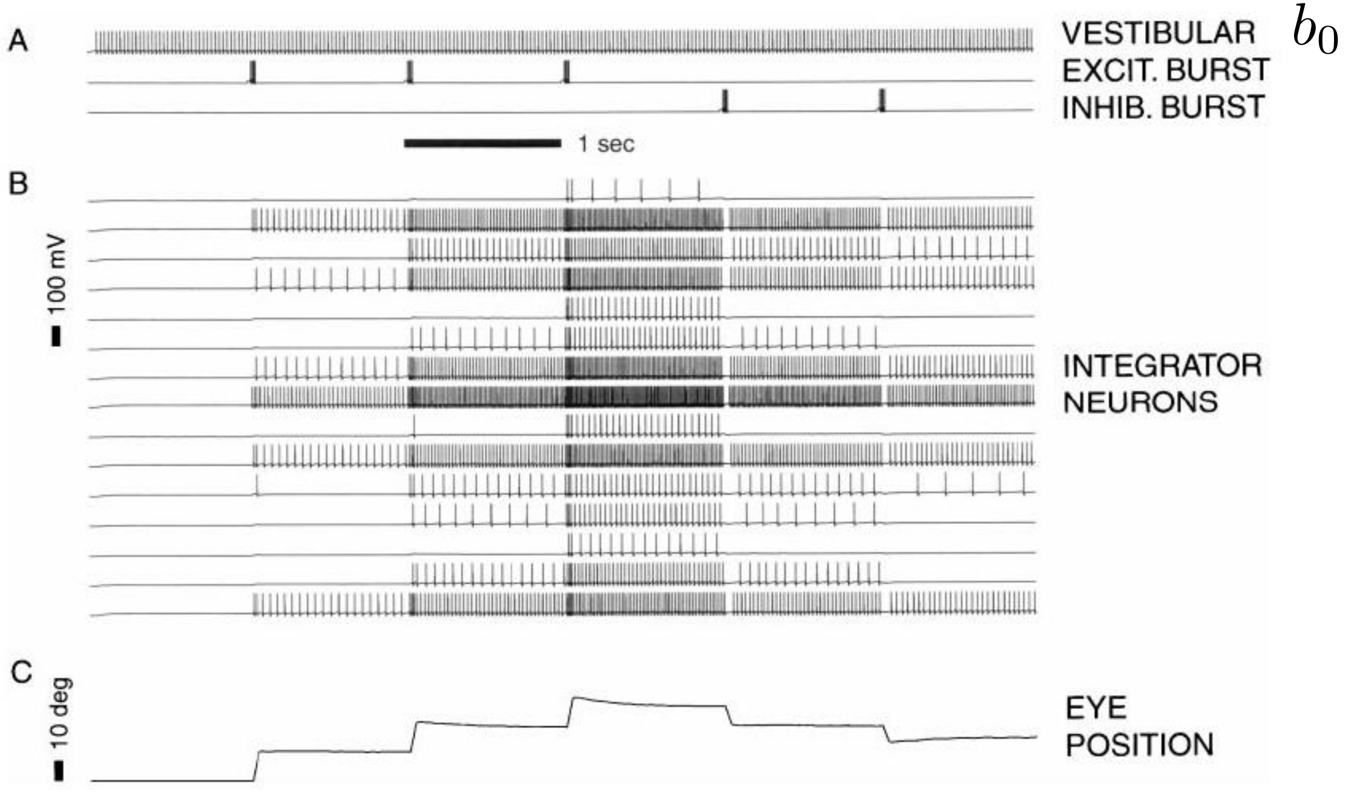


Model

Simple model: two mutually inhibitory populations



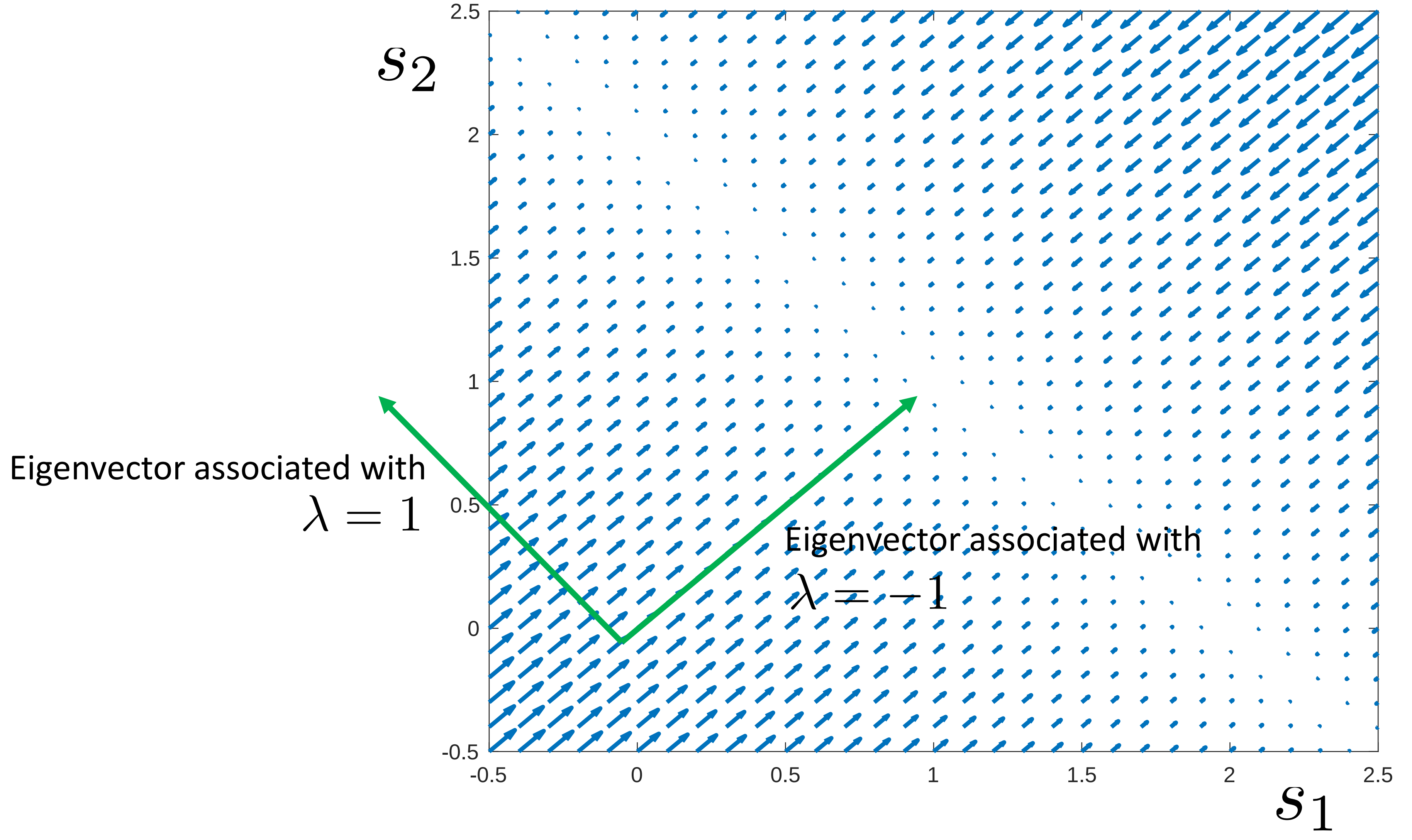
More complex model: multiple neurons, inclusion of saturating nonlinearity compensated by recruitment



→ Matlab demo of model.

(Homework: solve the dynamics of this circuit.)

State-space view: flows in the system



Arrows: flows (d**s**/dt), with magnitude given by arrow length

Linear symmetric networks summary

- Symmetric networks have only real eigenvalues.
- Symmetric linear networks have either a: single fixed point, no fixed points, or infinitely many fixed points.
- Stable single fixed-point system: amplification (slow) or attenuation (fast) of inputs.
- Continuum of fixed points along some dimension(s), stable dynamics in all others: “continuous attractor”
  - Analog memory
  - Integration over time of inputs
- Oculomotor integrator: biological example of system that operates analogously to a linear attractor, and exhibits line-attractor-like dynamics.