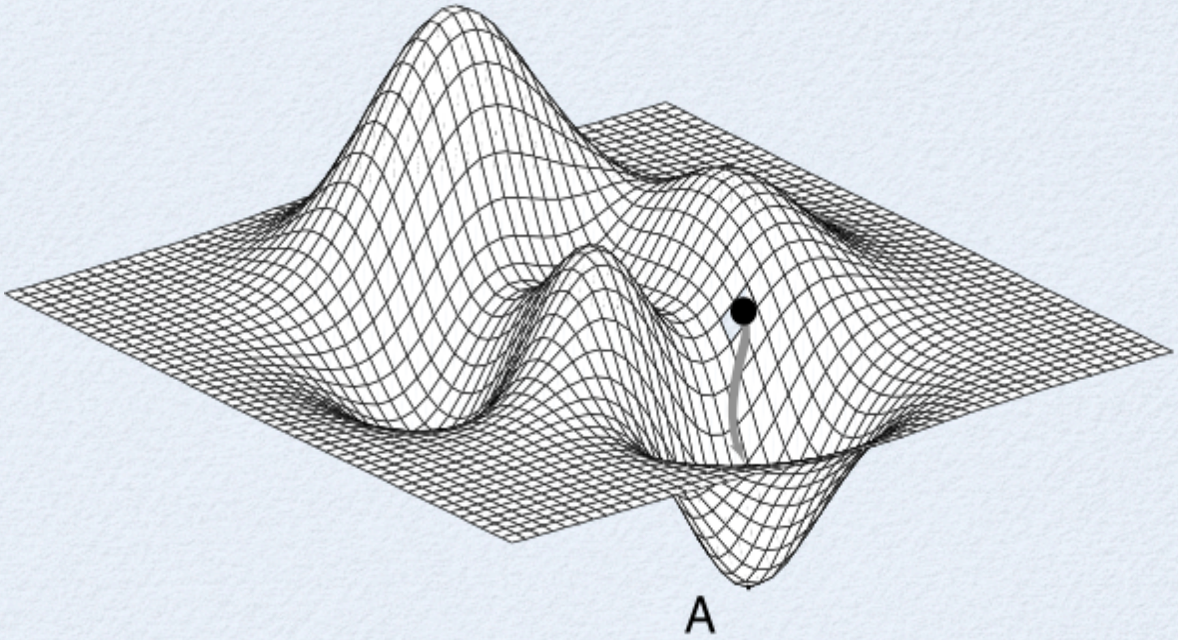
Dynamics: Attractor Networks

Attractor

- An attractor in dynamical systems theory is a system state (or states) towards which other states tend over time
- The standard analogy is to imagine the state space as a 'hill-like' topology which a ball travels through (tending downhill).

Attractor

- Point attractor at A

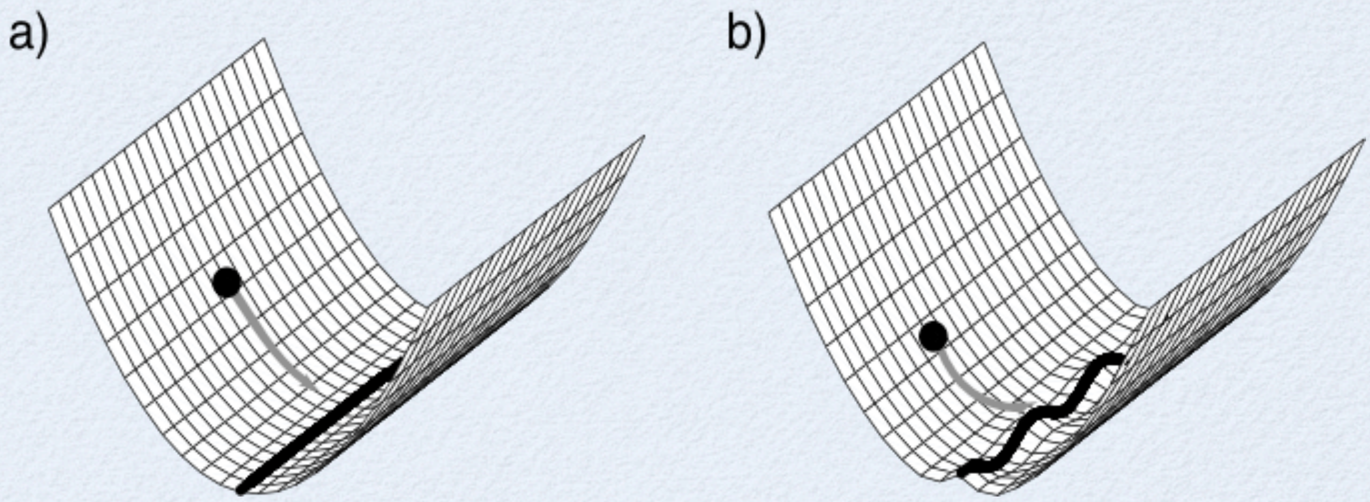


Attractors

- In neural network research, attractor networks (networks with dynamical attractors) have long been thought relevant for various behaviours
    - e.g., memory, integration, off-line updating of representations, repetitive pattern generation, noise reduction, etc.
- The neural integrator and working memory are both examples of attractor networks

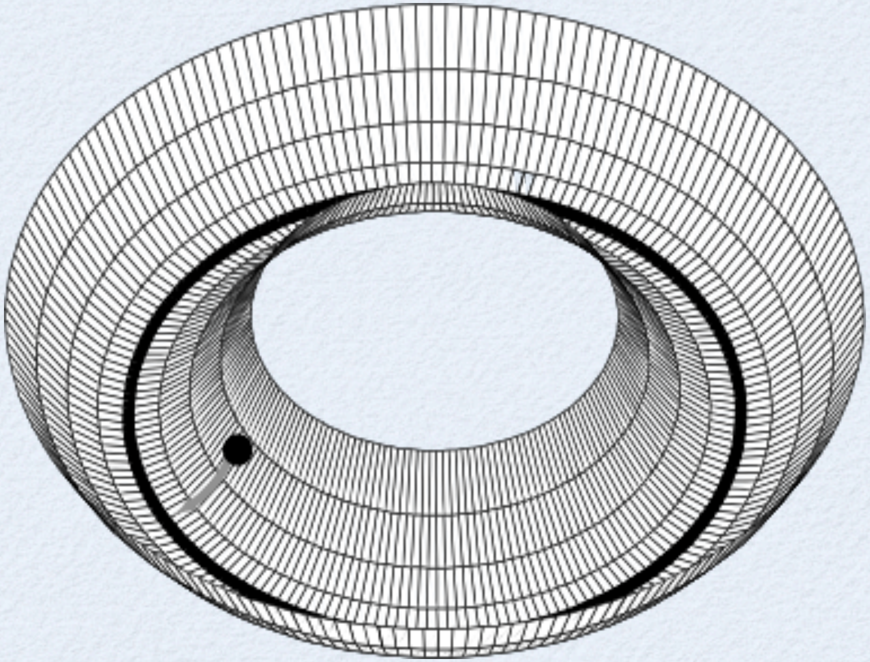
Line attractor

- The neural integrator can be thought of as a *line attractor* (approximately)



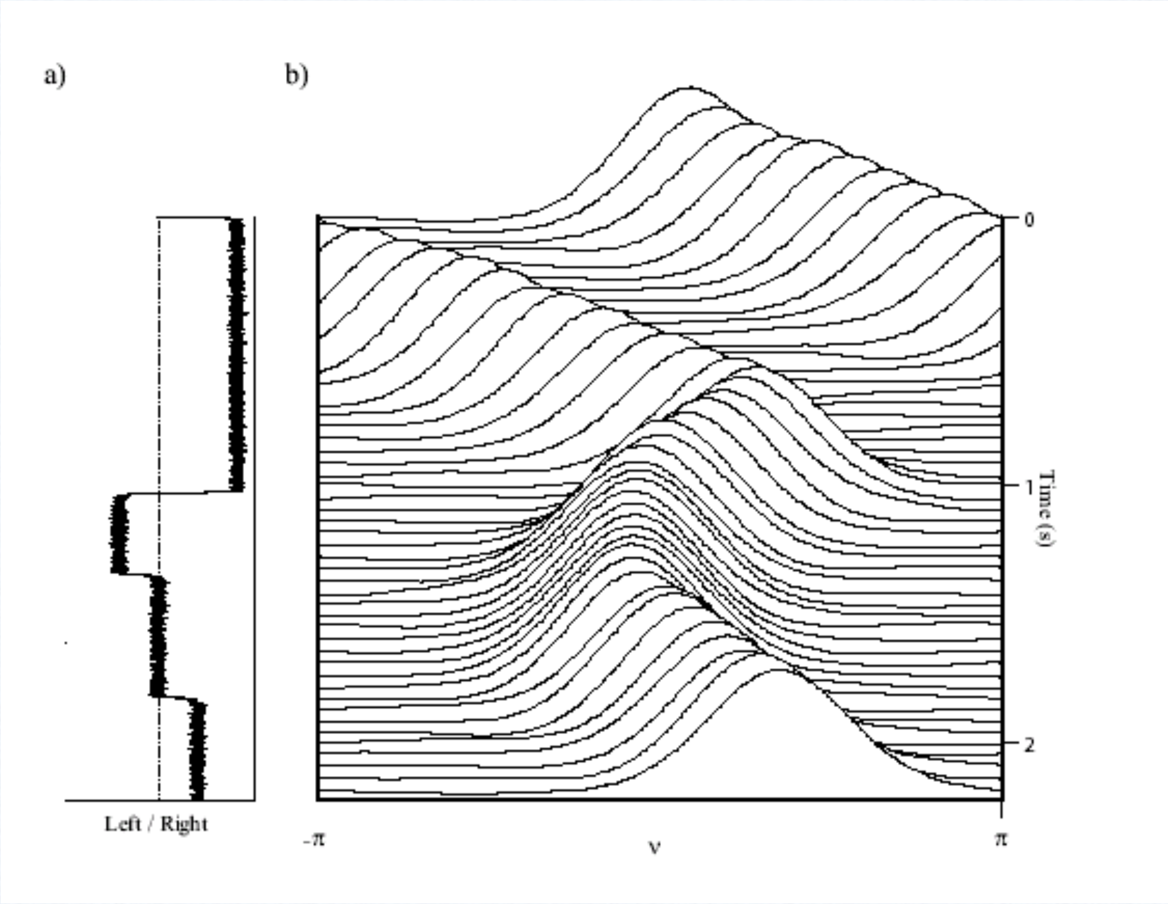
Ring attractor

- A line attractor that's wrapped in a circle is a *ring attractor* (describes systems that encode and hold positions over a repeating axis (e.g., head direction in hippocampus))



Ring attractor

- A ring attractor over time



Attractors

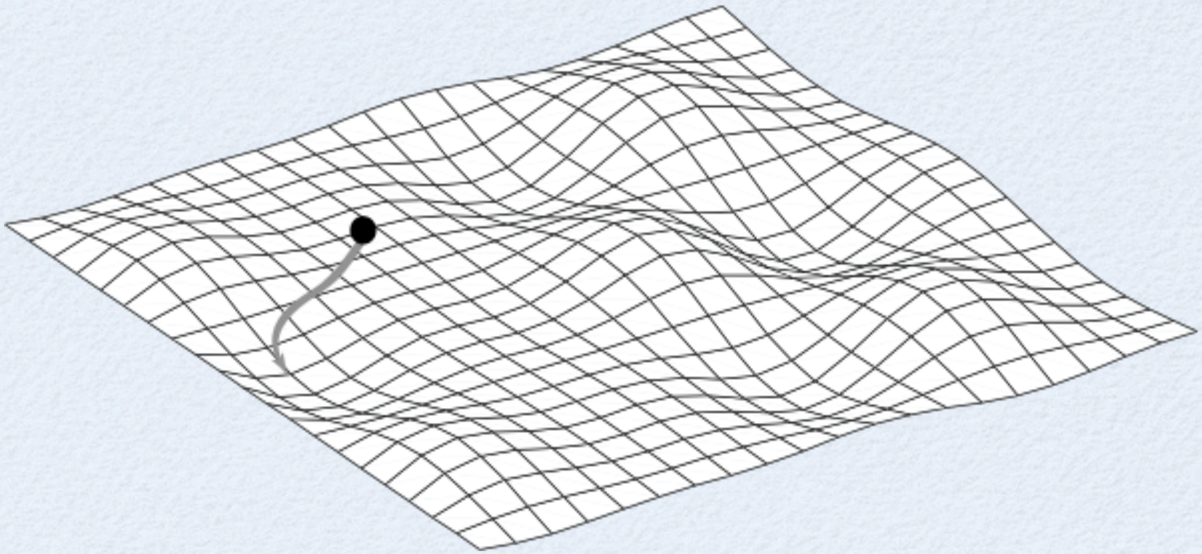
- Attractor networks were extensively examined in the ANN community (e.g. hopfield nets). Amit suggested that persistent activity could be associated with recurrent biological networks
    - Persistent activity is found in motor, premotor, parietal, prefrontal, frontal, hippocampal, and inferotemporal cortex; and basal ganglia, midbrain, superior colliculus, and brainstem
- Focus to date is on simple attractors, let's generalize...

Attractors

- Can generalize representation, or dynamics
- Representational gives more complex memory circuits (next slide)
- As a result, we can see the effects of e.g., heterogeneity on the dynamic properties of such systems (e.g. number of fixed points)

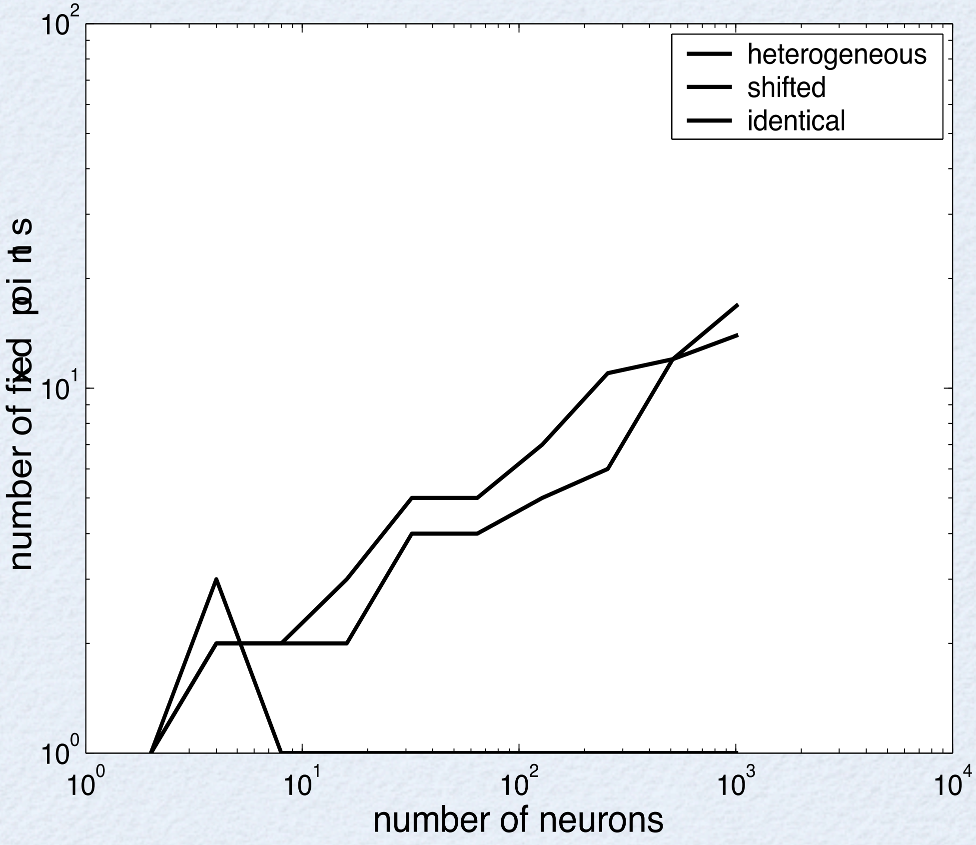
Plane attractor

- The memory network can be thought of as a *plane attractor* (approximately)



Fixed points

- Fixed points vs various x-intercept distributions



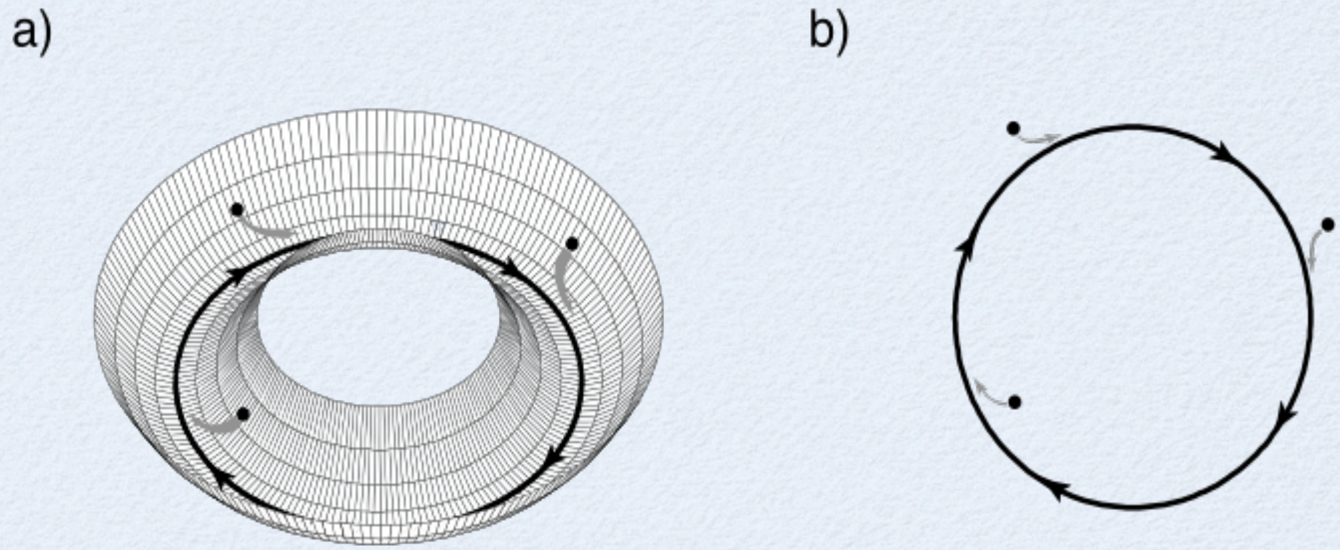
Generalizing attractors

- We can also generalize the dynamics, because we can relate dynamics to the control dynamics eqn
- So, e.g., we can determine the effects of interesting choices of **A**
- Consider a simple harmonic oscillator:

$$\mathbf{A} = \begin{bmatrix} 0 & \omega \\ -\omega & 0 \end{bmatrix}$$

Cyclic attractor

- NB: The ball analogy no longer applies



Cyclic attractor

- This isn't a simple cyclic attractor, since the amplitude depends on the initial conditions.
    - Just as we control the stable point of the line attractor by adjusting $u(t)$, we can control the cycle (i.e., amplitude) at which the oscillator attractor operates by adjusting $\mathbf{u}(t)$.
- The oscillator also includes the variable ω which controls the speed around the attractor.
    - Useful for describing repetitive behaviours

Lamprey

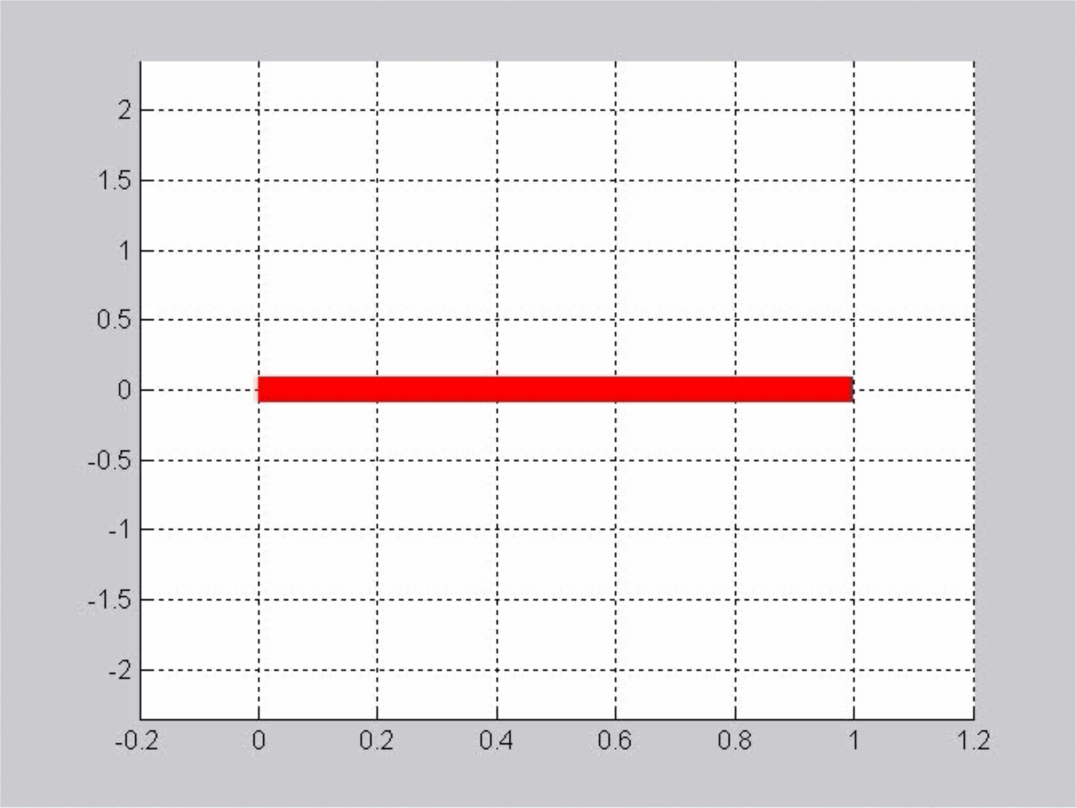
- The lamprey example implements a cyclic attractor (simple harmonic oscillator).
- There are two techniques of interest in this example: ‘damping’ and ‘leveling’
- Damping is just the insight that because neural representation is inaccurate, you sometimes have to write dynamical eqns that explicitly damp the introduced error.

Lamprey

- Leveling is simply taking advantage of the representational hierarchy we defined long ago
- In this case, we can introduce an intermediate level of representation (between the harmonic oscillator and neural characterizations), which is a ‘neural group’ level of representation that maps to lamprey spinal organization.

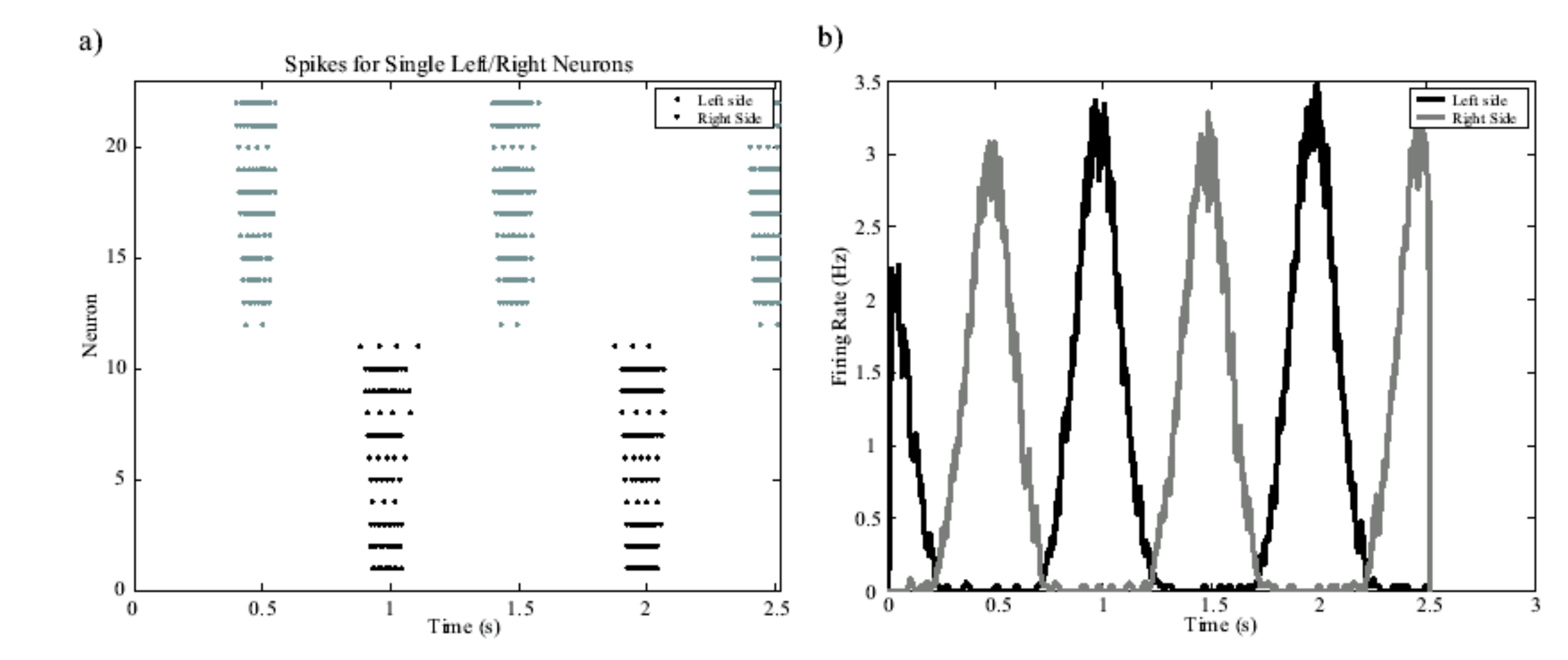
Lamprey swimming

- (lampswim.avi in MPlayer)



Lamprey swimming

- Single cell behaviour



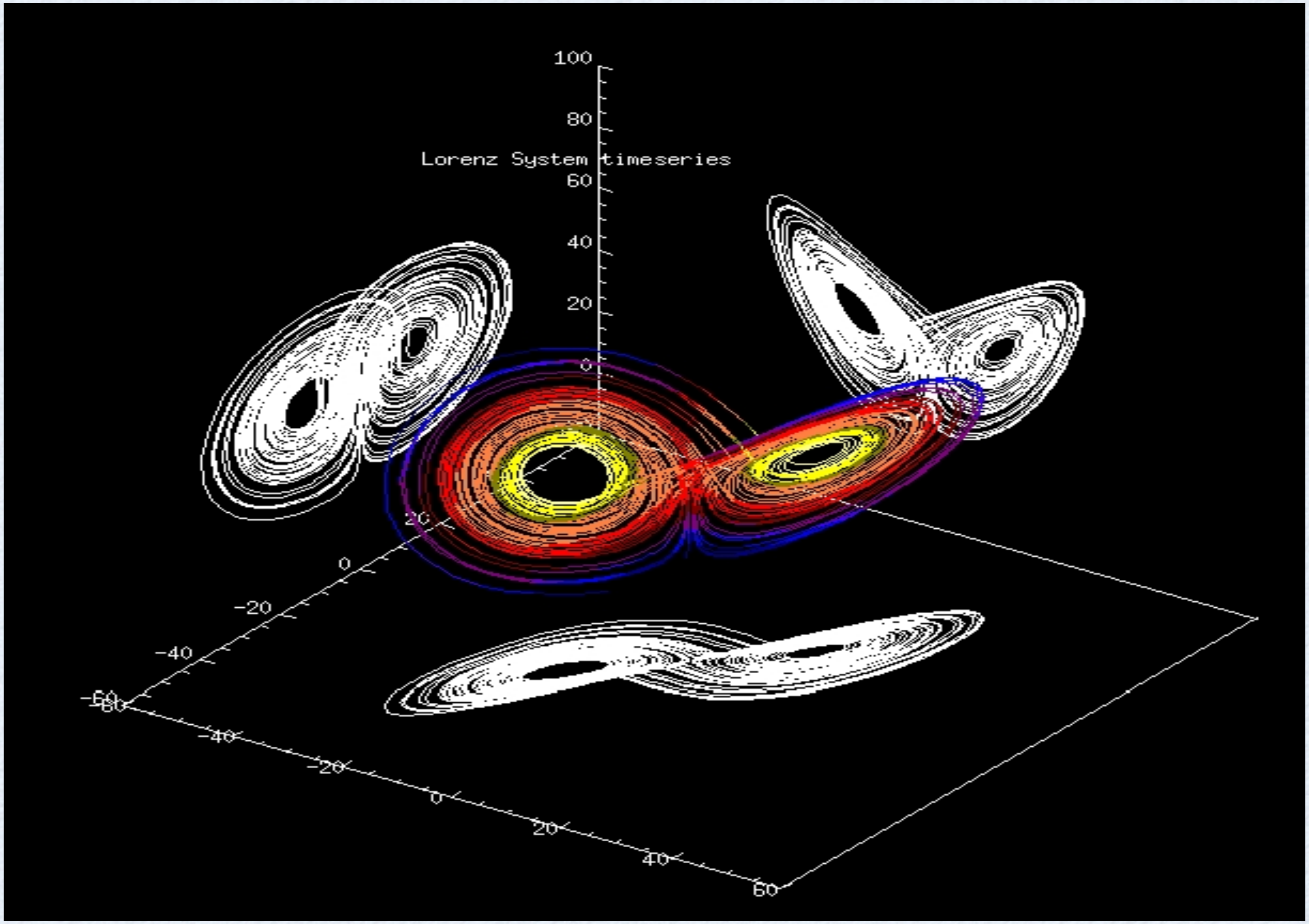
Lamprey

- This approach is unique in CPG modelling
- Advantages:
  - Accounts for observed dynamic behaviour
  - Can explicitly enforce stability and control
  - Consistent with neurophysiology and anatomy

Lorenz Attractor

- The Lorenz attractor is the simplest continuous chaotic attractor.
- Here we show how to move between kinds of attractors in a single network
- This kind of control is ideal for understanding various behaviours that have distinct regimes of operation (walking vs running, etc.)

Chaotic attractor (Lorenz)



Lorenz equations

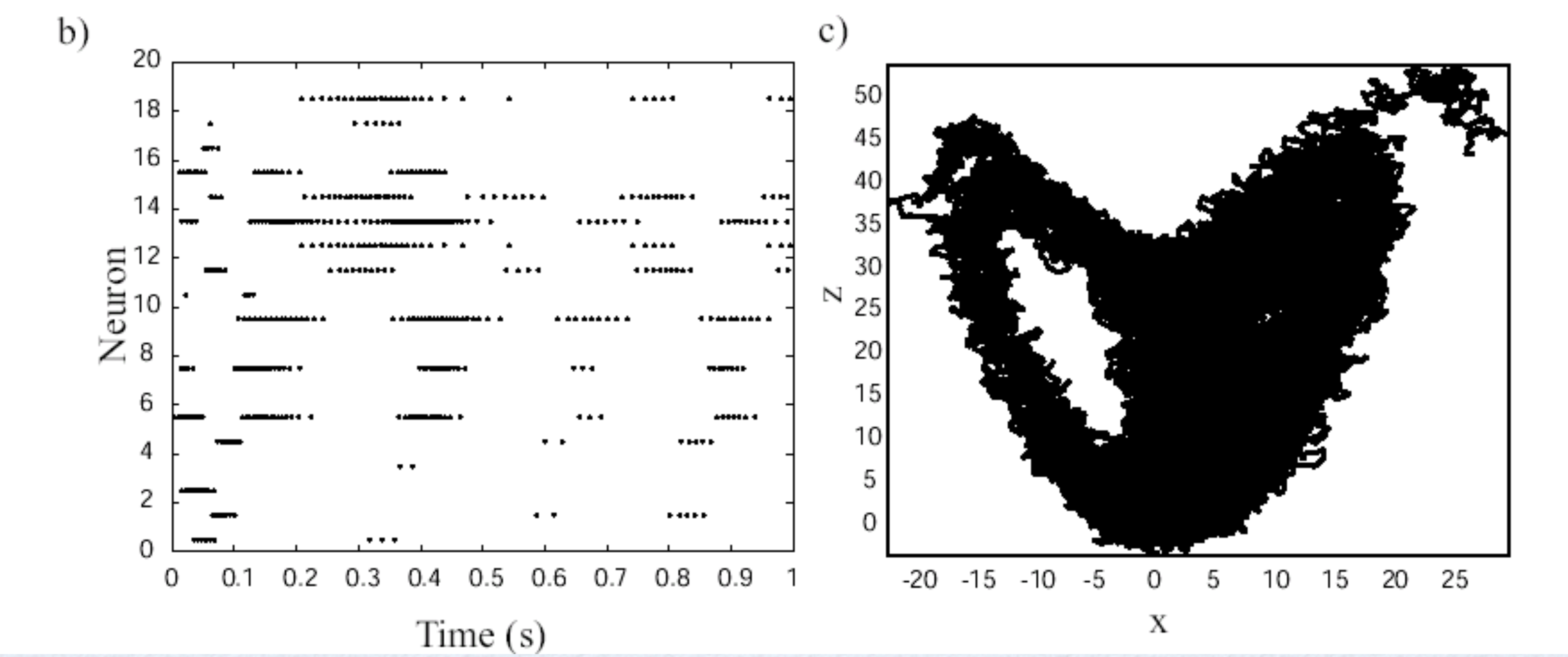
- a=10, b=28, and c=8/3; changing b over the range [1, 300] traverses different attractor types

$$\begin{aligned}
\dot{x}_1 &= a(x_2 - x_1) \\
\dot{x}_2 &= bx_1 - x_2 - x_1x_3 \\
\dot{x}_3 &= x_1x_2 - cx_3
\end{aligned}$$

- (Nonlinear) control version

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{bmatrix} = \begin{bmatrix} -a & a & 0 \\ b & -1 & x_1 \\ x_2 & 0 & -c \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$$

Chaotic attractor (Lorenz)



Lorenz equations

- This wide range of b, and coupling to x_3, means we need big dynamic range (hence lots of neurons). More efficiently we can rearrange:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{bmatrix} = \begin{bmatrix} -a & a & 0 \\ 0 & -1 & -x_1 \\ x_2 & 0 & -c \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ -(c+1) & 0 & 0 \end{bmatrix} \begin{bmatrix} b \\ 0 \\ 0 \end{bmatrix}$$

- The control is now not even multiplicative

Chaotic attractor (Lorenz)