NEF and Nengo

Centre for Theoretical Neuroscience

Nengo Summer School

Chris & Terry

Waterloo

UNIVERSITY OF WATERLOO

CONCORDIA CUM VERITATE

Three Principles of the NEF

1. *Representation*

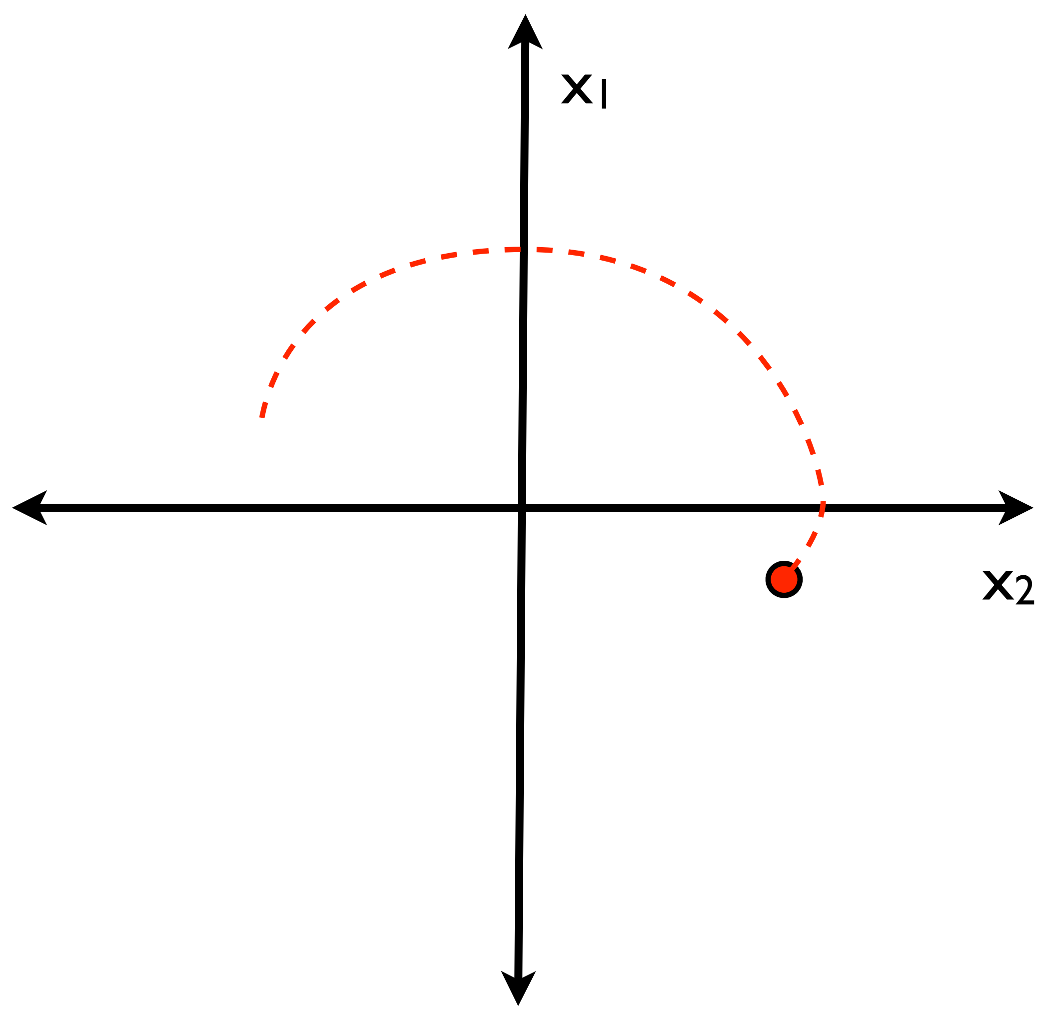
2. *Computation/Transformation*

3. *Dynamics*

- Neural representations (1) are control theoretic state variables in a nonlinear (2) dynamical system

Principle 3: Dynamics

- Control theory is one general way to write descriptions of time-varying phenomena



neural dynamics

neural transformation

$$\dot{\mathbf{x}} = f(\mathbf{x}, t)$$

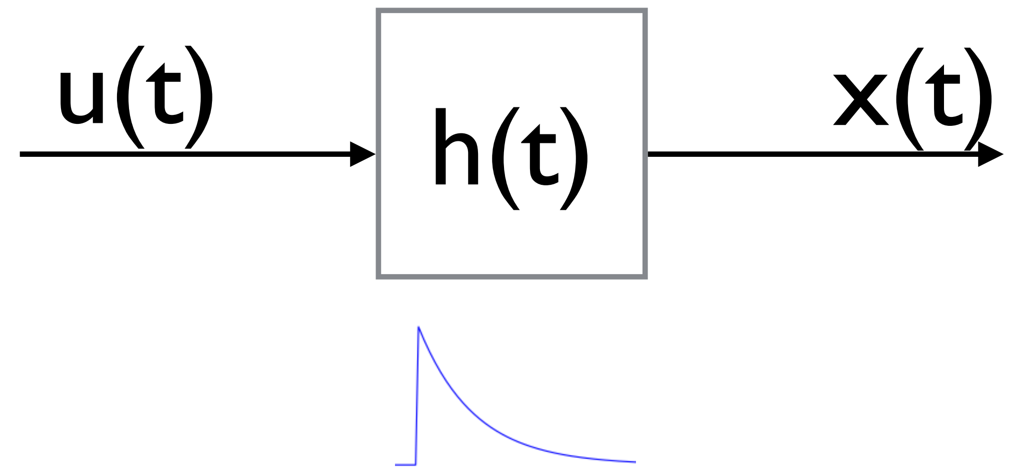
neural representation

Dynamics

- We’ve already seen (feedforward) dynamics

$$x(t) = u(t) * h(t)$$

$$\Rightarrow \quad X(s) = U(s)H(s)$$



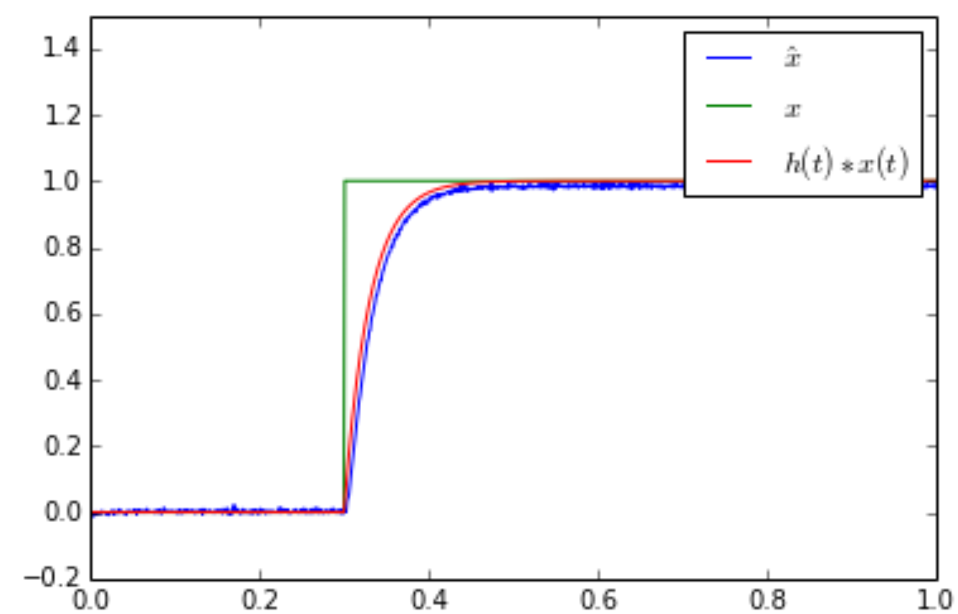
- Let $$H(s) = \frac{1}{(1 + s\tau)}$$

- Then $$X(s) = U(s)\frac{1}{(1+s\tau)}$$

$$X(s)(1+s\tau) = U(s)$$

$$X(s) + s\tau X(s) = U(s)$$

$$sX(s) = (U(s) - X(s))/\tau$$

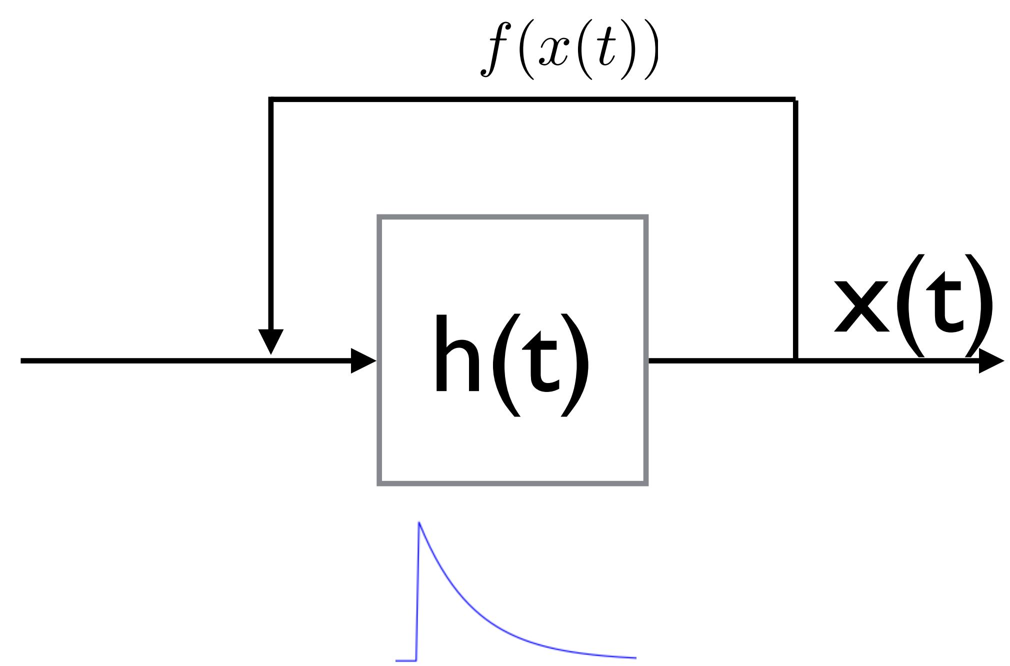


$$\Rightarrow \quad \dot{x} = -\frac{1}{\tau}(x(t) - u(t))$$

Dynamics

- Same is true for a recurrent connection:

$$x(t) = f(x(t)) * h(t) \Rightarrow X(s) = F(s)H(s)$$



$$X(s) = F(s)\frac{1}{1+s\tau}$$

$$X(s)(1+s\tau) = F(s)$$

$$X(s) + X(s)s\tau = F(s)$$

$$sX(s) = \frac{1}{\tau}(F(s) - X(s))$$

$$\Rightarrow \quad \dot{x} = -\frac{1}{\tau}\left(x(t) - f(x(t))\right)$$

Dynamics

- Often want to go the ‘other direction’
- We want: $\dot{x} = f(x) + g(u)$
- We’ll get: $\dot{x} = \frac{1}{\tau}(f(x) + g(u) - x)$
- We change what we want to:

$$f' + g' = \tau(f(x) + g(u) + x)$$

- We get:

$$\begin{aligned}\dot{x} &= \frac{1}{\tau}\tau(f(x) + g(u) + x - x)\\ &= f(x) + g(u)\end{aligned}$$

Dynamics

- Therefore, if we want:

$$\dot{x} = f(x) + g(u)$$

- We set our 'implemented' dynamics to:

$$f' = \tau f(x) + x$$

$$g' = \tau g(u)$$

- So that:

$$\begin{aligned} \dot{x} &= \frac{1}{\tau}(f' + g' - x) \\ &= f(x) + g(u) \end{aligned}$$

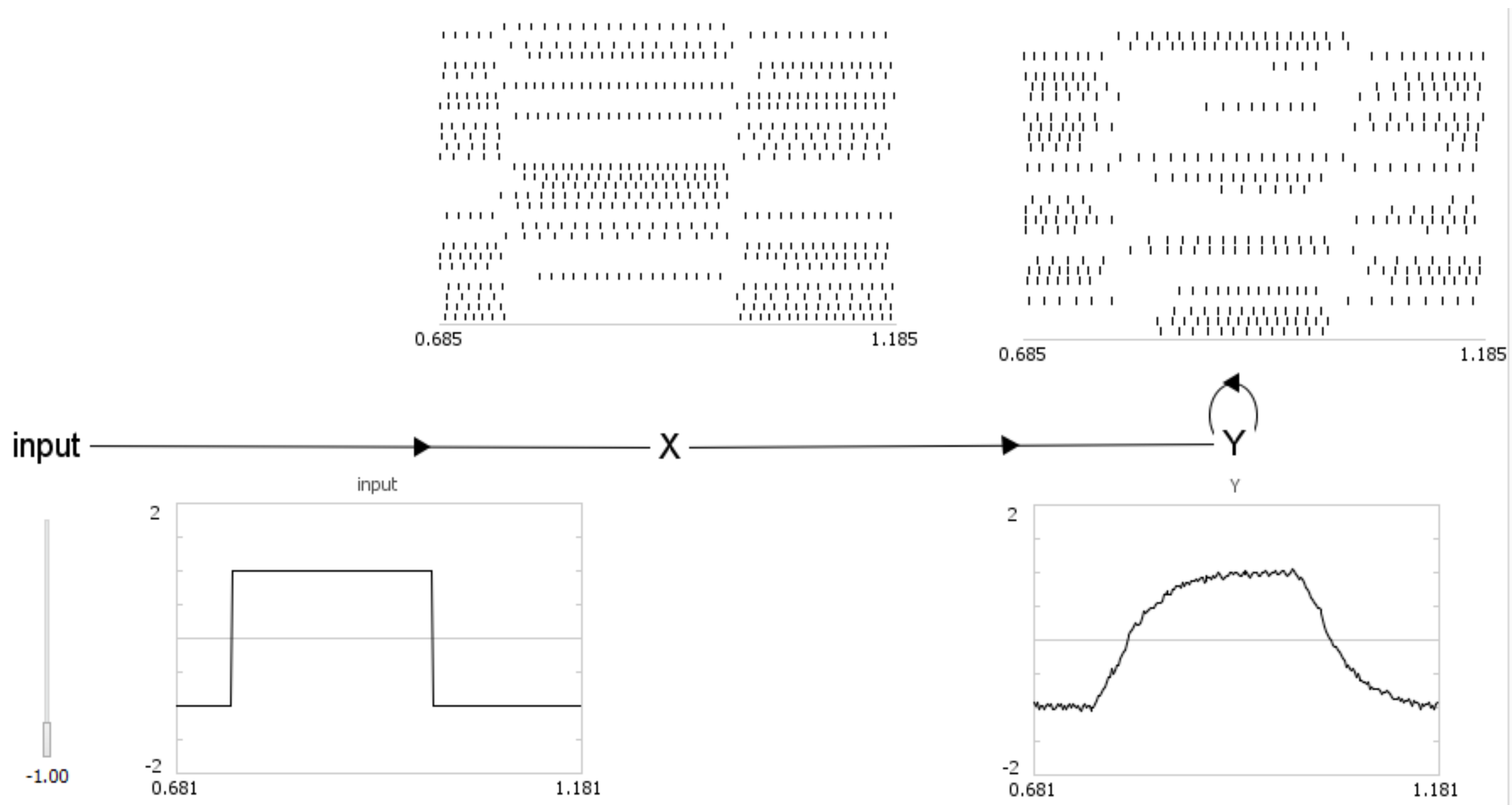
Filtering example

- Therefore, if we want: $\dot{x} = -\frac{1}{0.05}(x(t) - u(t))$

- We set our 'implemented' dynamics to:

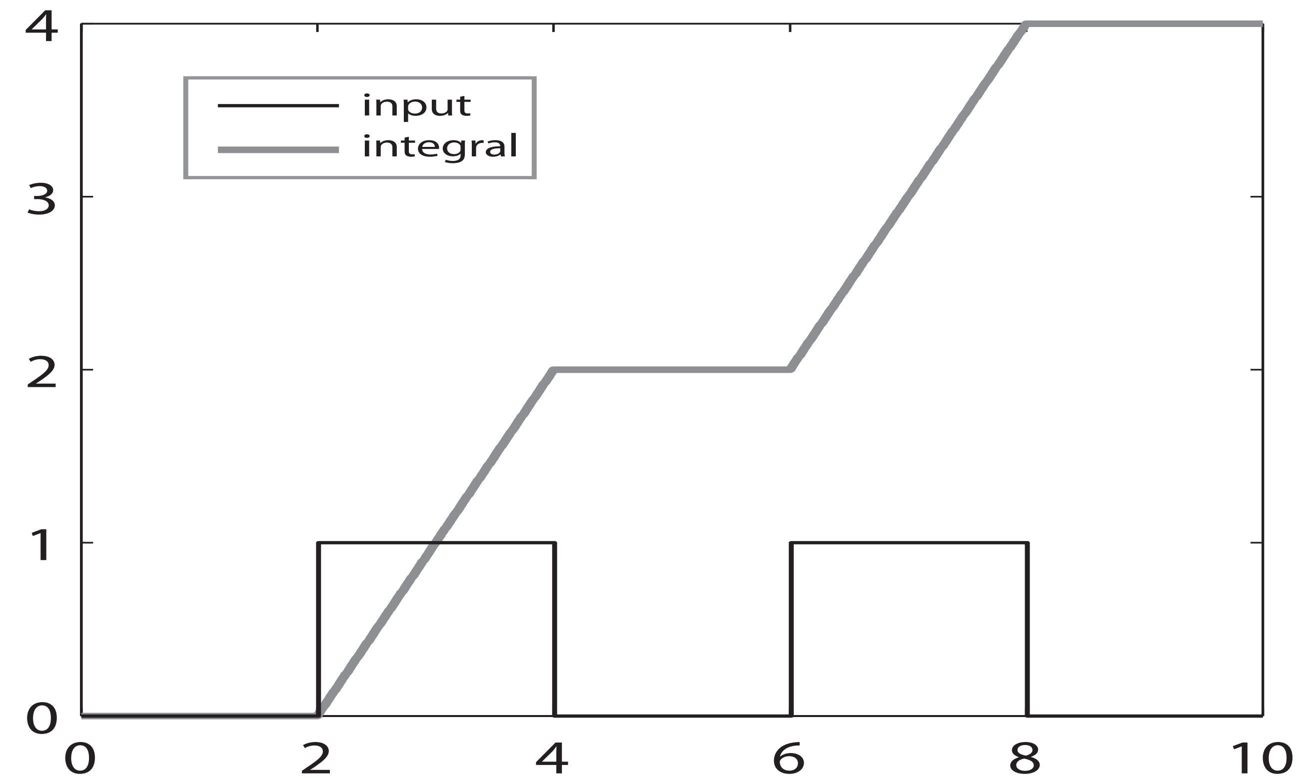
$$f' = -\frac{\tau}{0.05}x + x = (-\frac{\tau}{0.05} + 1)x$$

$$g' = \frac{\tau}{0.05}u$$



Neural integrator

- NPH & VN turn velocity signals into eye position commands



- Difficult problem to solve, but simple to formulate:

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u}$$

$$\mathbf{A} = 0$$

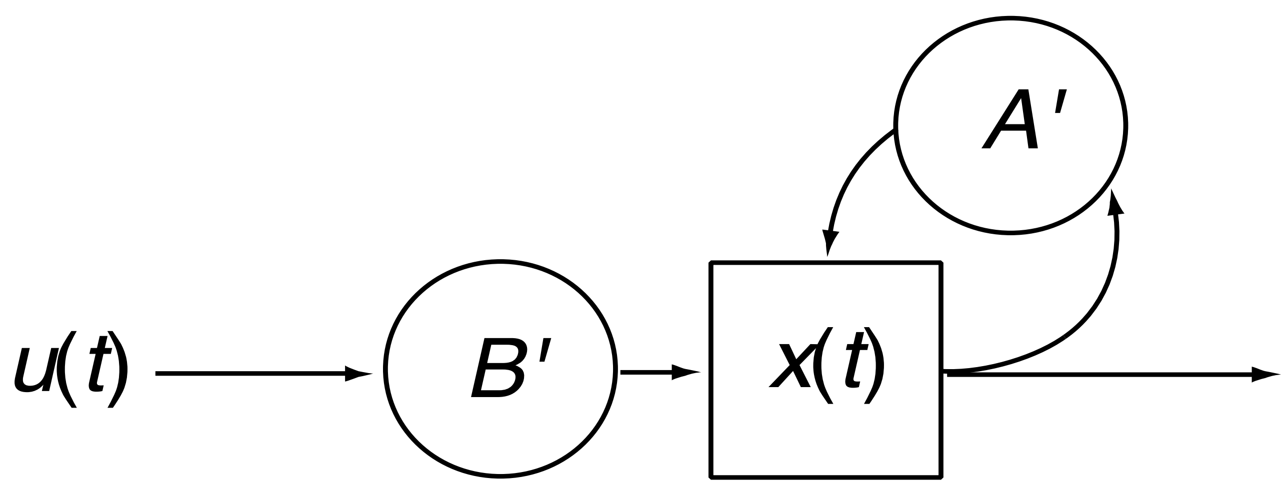
$$\mathbf{B} = \mathbf{I}$$

Neural integrator

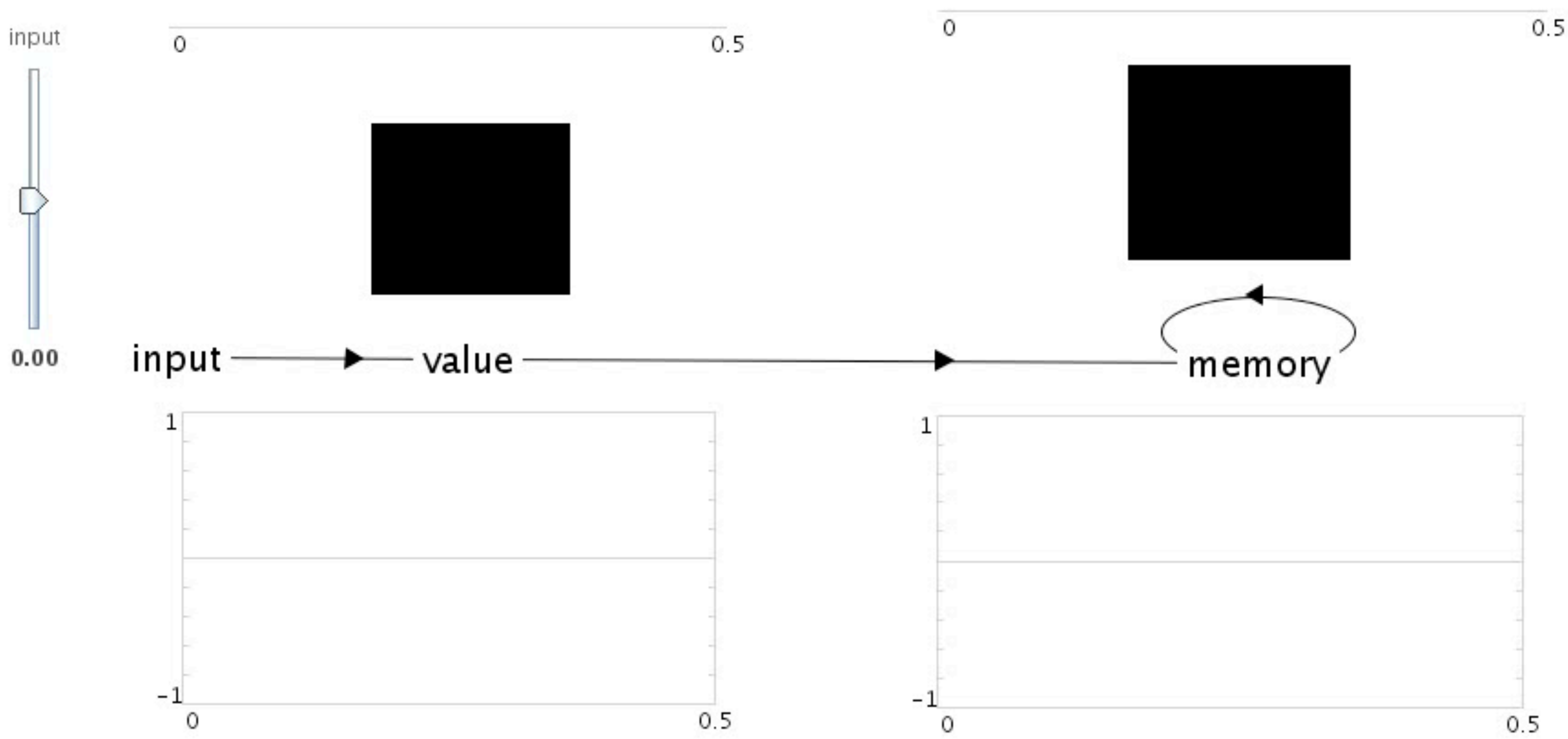
$$\mathbf{A}' = 1$$

$$\mathbf{B}' = \tau$$

- So, in 'neural control' we have



Integrator



Example: Working memory

- We want a neural circuit that:
  - stores an input state
  - can be cleared (controlled storage)
- Dynamical equation (i.e., high-level program):

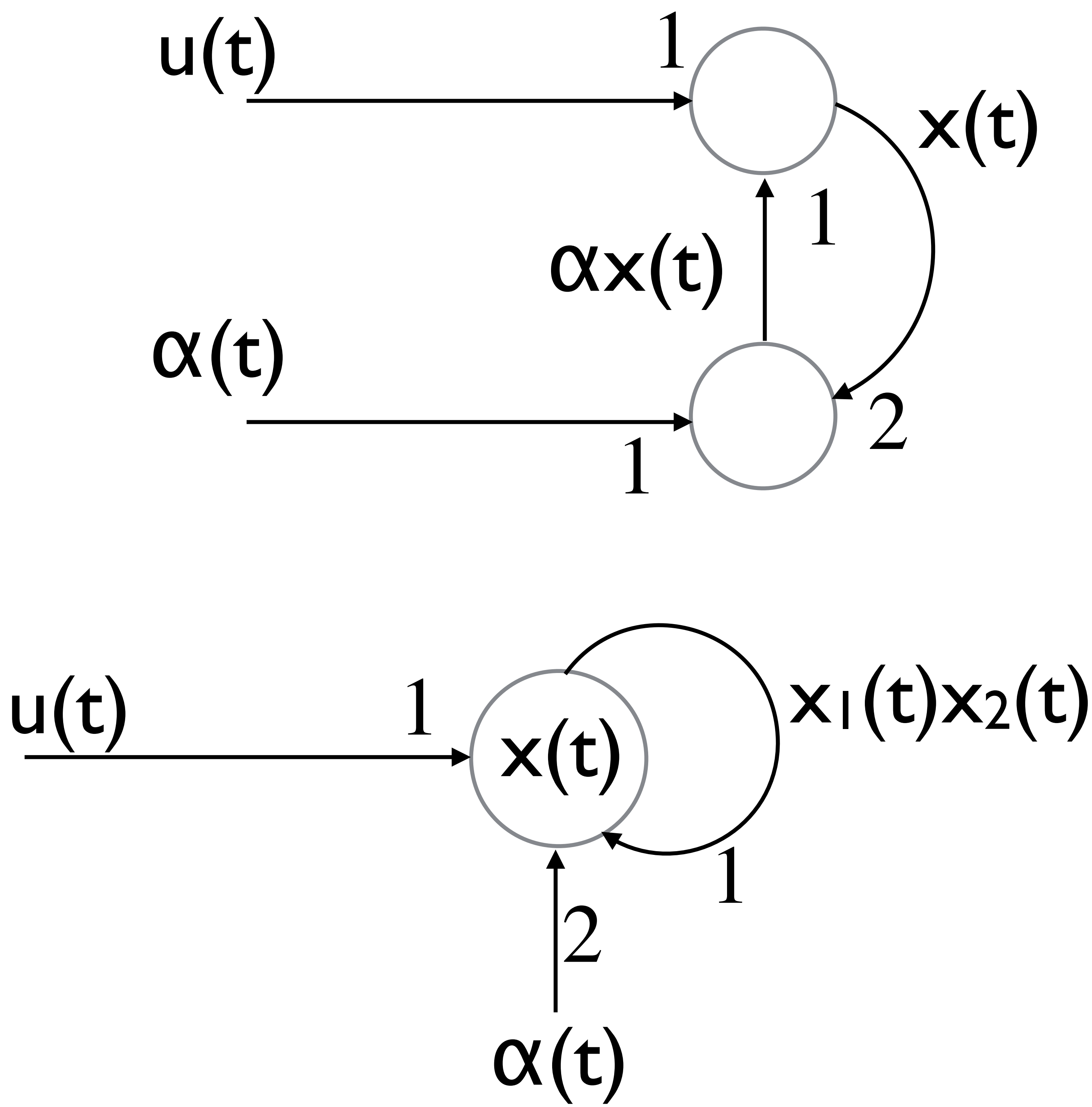
Input

Memory

$$\dot{\mathbf{x}} = \alpha \mathbf{I}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t)$$

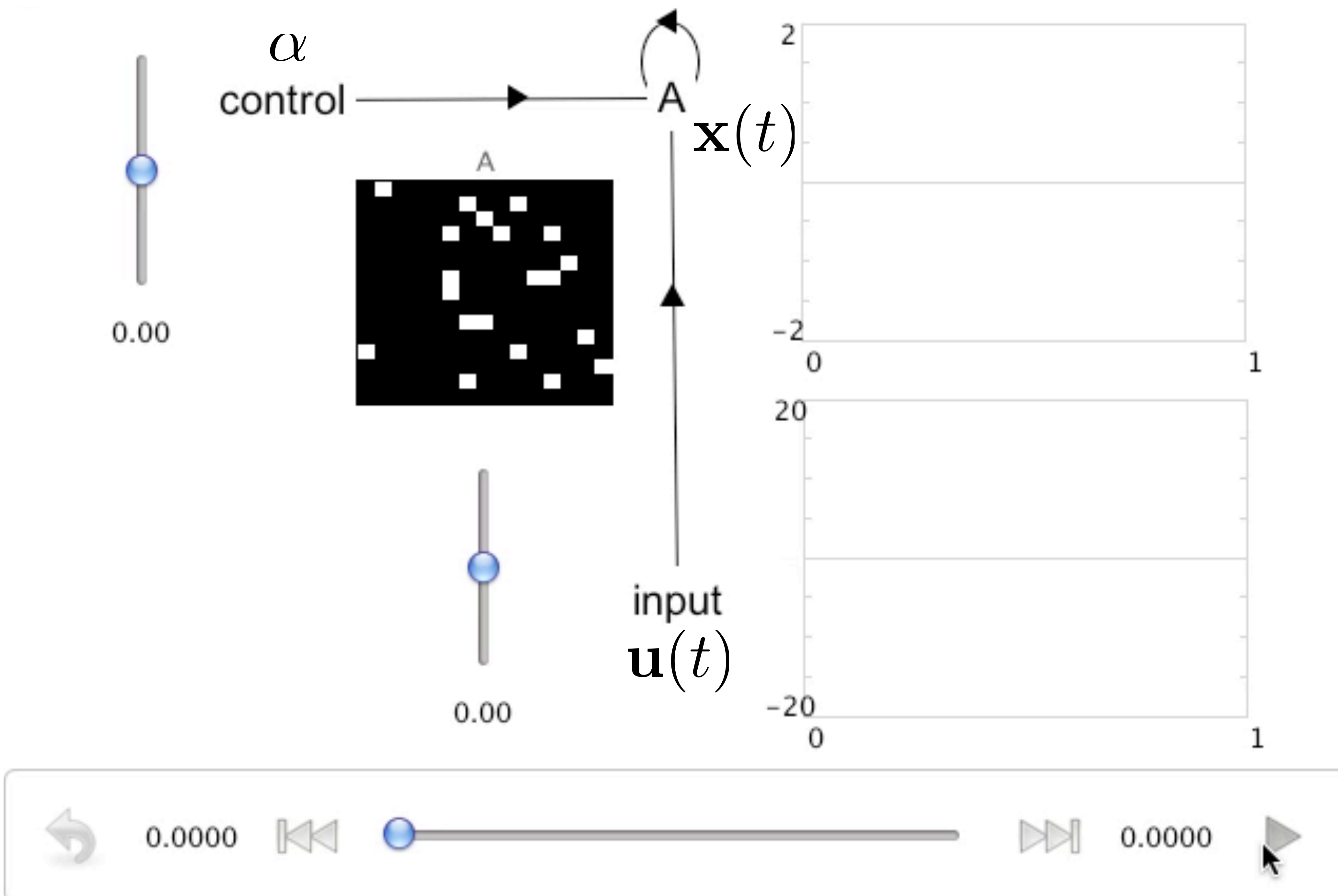
Control

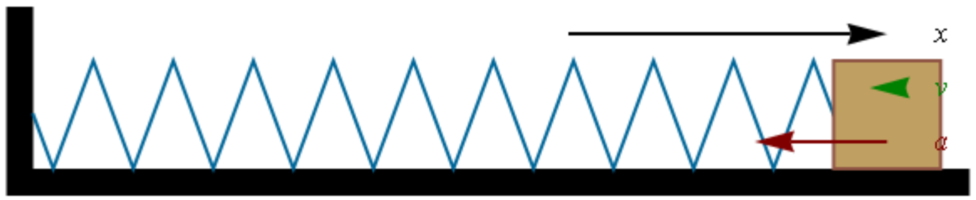
Architectures



Principle 3: Dynamics

$$\dot{\mathbf{x}} = \alpha \mathbf{I}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t)$$





Simple Oscillator

- An oscillator $F = -kx = ma = m\ddot{x}$
- That is $\ddot{x} + \frac{k}{m}x = 0$ let $\omega = \sqrt{\frac{k}{m}}$

- Which can be written

$$\frac{d}{dt}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 0 & \omega \\ -\omega & 0 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

Controlled oscillator

- Control the speed of a neural oscillator

Fun with dynamics

- Chaotic attractor (Lorenz)
- Oddly shaped oscillators (heart? square?)

Summary

- Built a controlled nonlinear dynamical system in a spiking network
- Principles used are very general (vector space representation, nonlinear computation, nonlinear dynamics)
- Dealt with heterogeneity, nonlinearities, noise
- ... scaling... later

Build a Critter

NEF and Nengo

Centre for Theoretical Neuroscience
Nengo Summer School

Chris & Terry

Build a Critter

- Two inputs:
  - A desired velocity
  - A “fear” indicator
- Behaviour:
  - If “fear” is low, move with the desired velocity
  - Otherwise, move back to the starting location