Overview

CNS*2003 Workshop

Unifying Neural Coding, Computation and Dynamics

C. Eliasmith & C. H. Anderson

Purpose

- Given
  1. Basic neural data (tuning curves); and
  2. An hypothesis about what a neural system is doing (computationally and/or over time)
- Provide
  - A fully spiking (recurrent) neural model appropriately connected to compute the given function using neurons with the given properties

Theoretical framework

- Three principles:
  1. *Representation*: Neural representations are defined by nonlinear encoding and linear decoding
  2. *Computation*: Neural transformations are determined by an alternate linear decoding.
  3. *Dynamics*: Neural representations are control theoretic state variables.

Theoretical framework

- Principle 1 – Representation
  - *Encoding*: determined by neuron tuning curves

$$\sum_n \delta_i(t - t_n) = G_i\left[\alpha_i \left\langle \tilde{\phi}_i \cdot \mathbf{x}(t) \right\rangle + J_i^{bias}\right]$$

  - *Decoding*: determined by ‘optimal’ weights

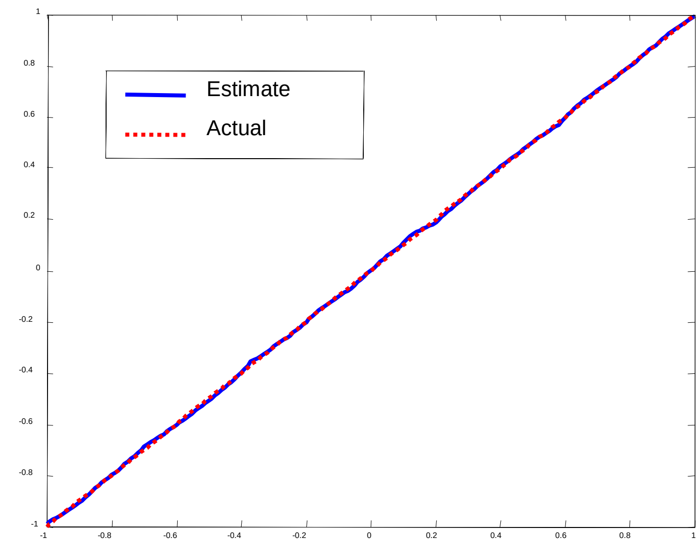
$$\begin{aligned} \hat{\mathbf{x}}(t) &= \sum_{in} \delta_i(t - t_n) * h_i(t)\phi_i^{\mathbf{x}} \\ &= \sum_{in} \delta_i(t - t_n) * \phi_i^{\mathbf{x}}(t) \\ &= \sum_{in} \phi_i^{\mathbf{x}}(t - t_n) \end{aligned}$$

Theoretical framework

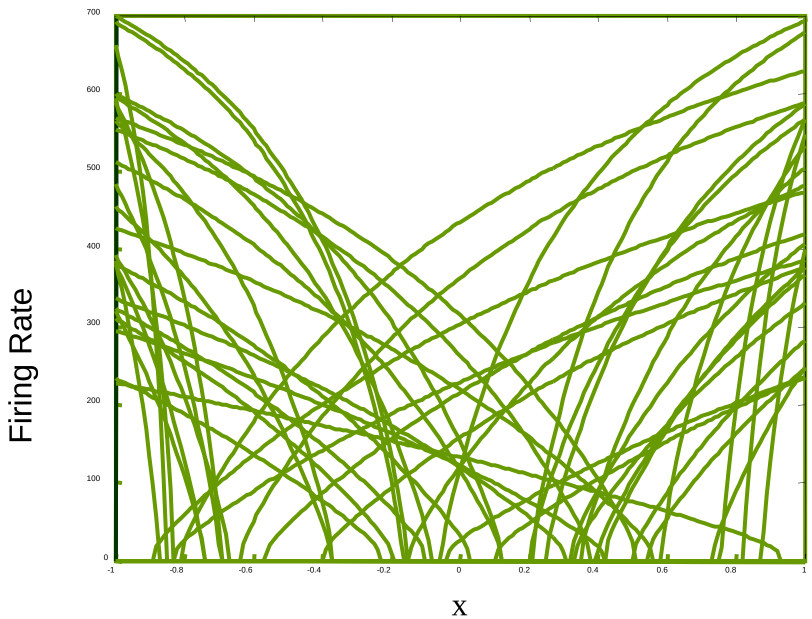
- Notes on neural representation:
  - Temporal decoders are PSCs
    - Huge increase in plausibility
    - Small, easily compensated drop in info transfer
  - Population and temporal coding can be analyzed separately but results apply to population-temporal coding. E.g.,
    - Error and number of neurons (1/N)
    - Computable functions

Neural representation

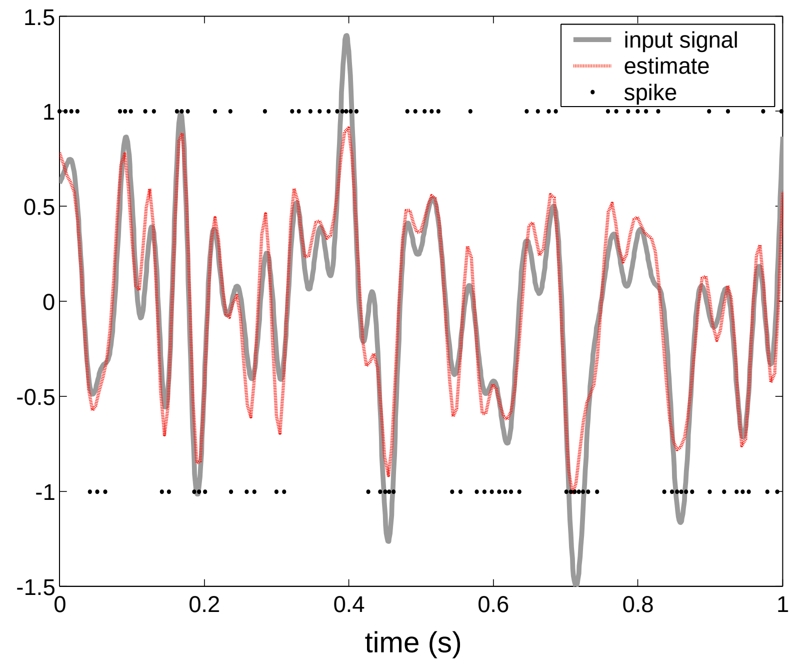
Population Decoding: Estimate of x



Population Encoding: Neuron Tuning Curves



Temporal Encoding (spikes) and Decoding (signal)



Theoretical framework

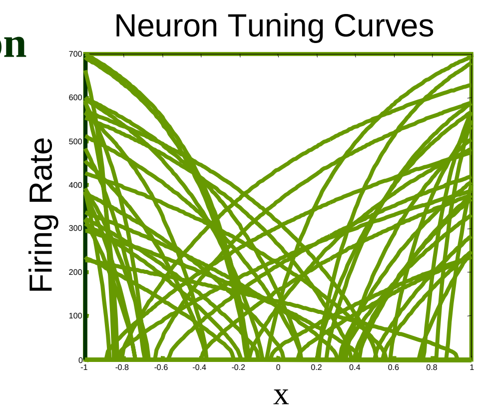
- Principle 2 – Computation
  - *Encoding*: same as principle 1
  - *Decoding*: different from principle 1 (transformational decoding) but still linear and optimal

$$\hat{f}(\mathbf{x};t) = \sum_i a_i(\mathbf{x}(t))\phi_i^{f(\mathbf{x})}$$

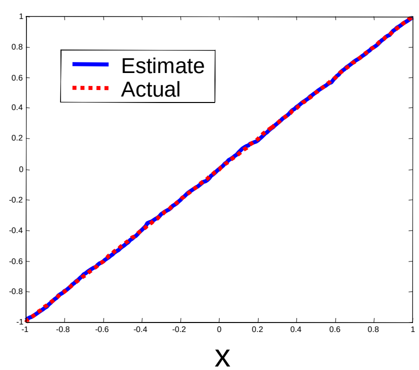
Neural computation

Representation

Encoding

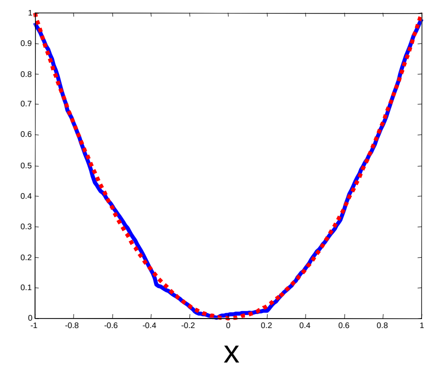


Representational
Decoding

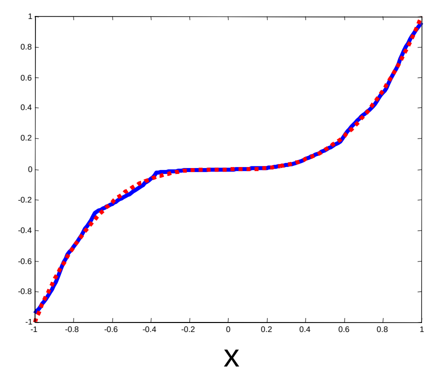


Computation

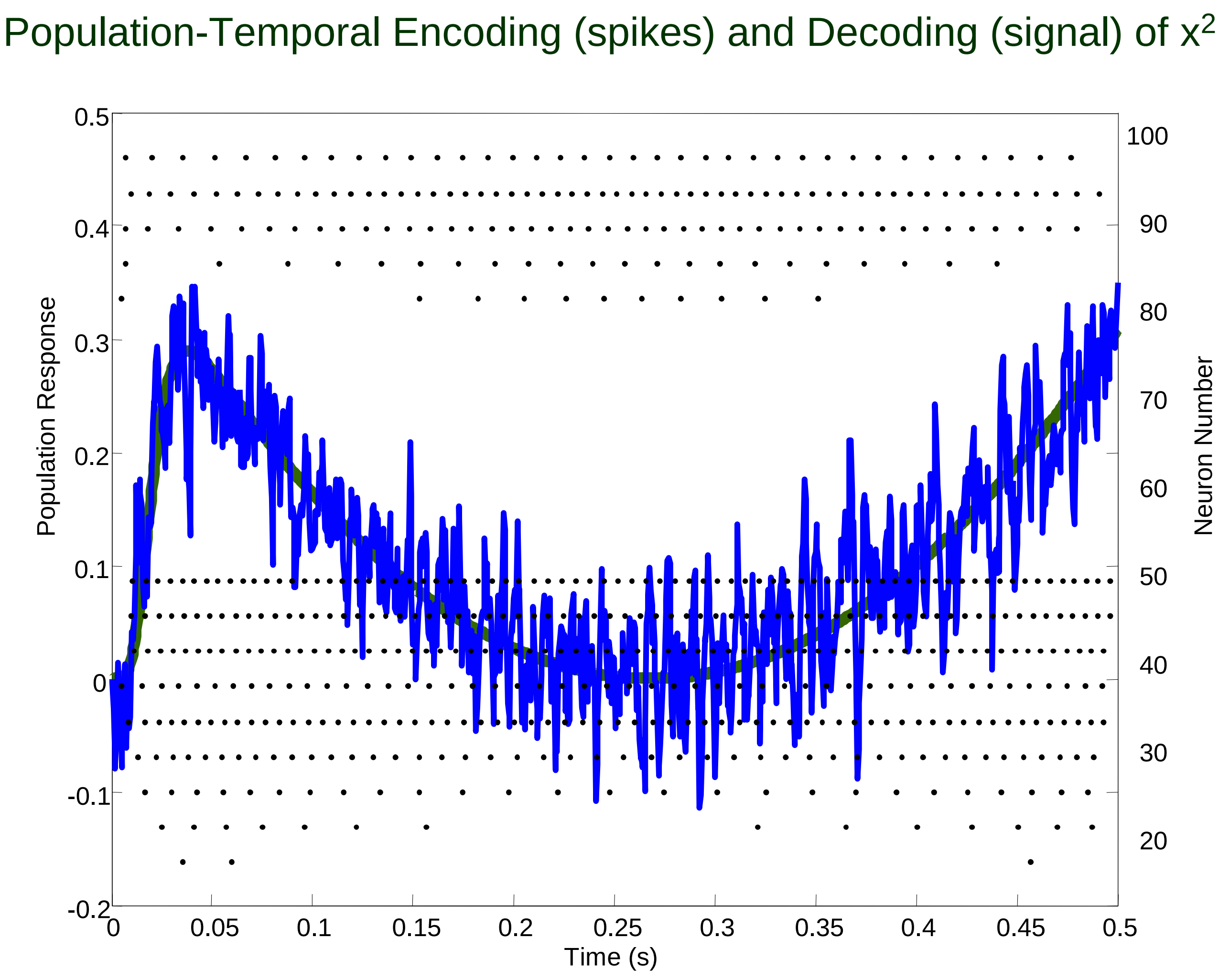
Quadratic
Decoding



Cubic
Decoding



Neural computation



Theoretical framework

- Principle 3 – Dynamics
  - *Neural representations are control theoretic state variables.*

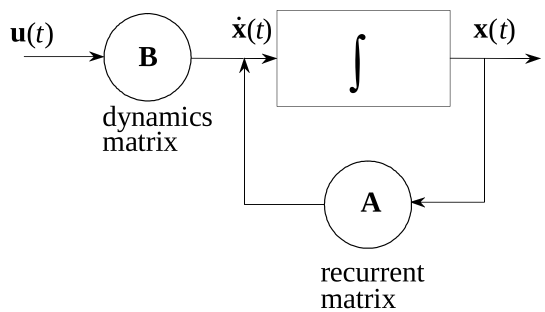
$$\mathbf{x}(s) = h(s)\left[\mathbf{A}\mathbf{x}(s) + \mathbf{B}\mathbf{u}(s)\right]$$

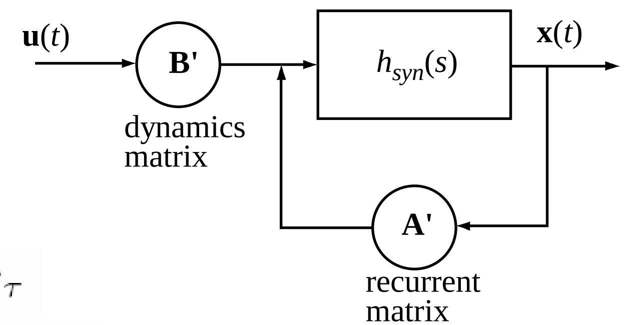
$$\sum_n \delta_i(t - t_n) = G_i\left[\alpha_i\left\langle\tilde{\phi}_i\left(h_i(t) * \left[\mathbf{A}'\mathbf{x}(t) + \mathbf{B}'\mathbf{u}(t)\right]\right)\right\rangle_m + J_i^{bias}\right]$$

  - Uses representation from principle 1 and computations from principle 2 to implement a given control structure.

Theoretical framework

- Need to translate between standard and neural control structures





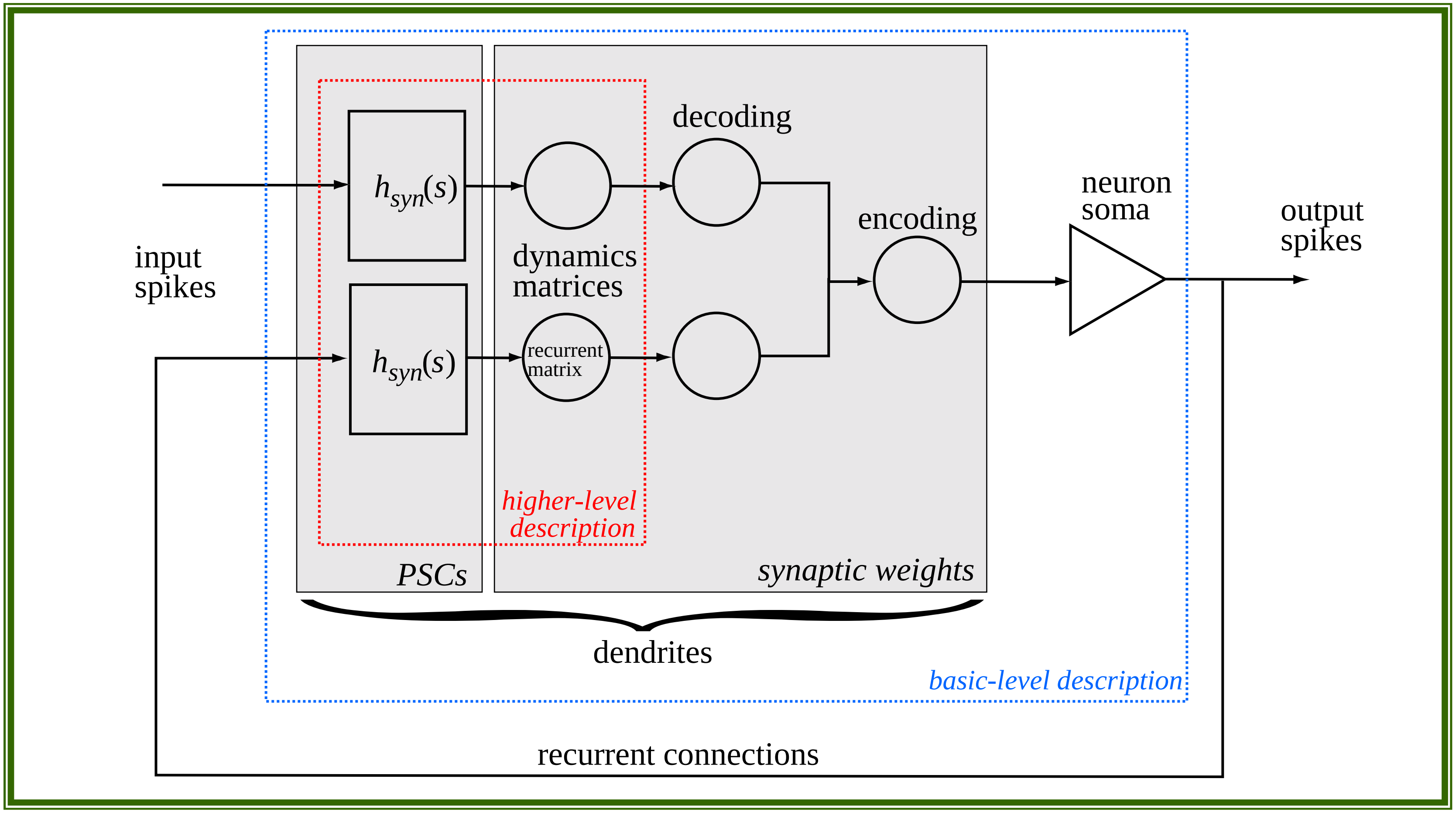
For $h(t) = \frac{1}{\tau} e^{-t/\tau}$

$$\begin{aligned} \mathbf{A}' &= \tau \mathbf{A} + \mathbf{I} \\ \mathbf{B}' &= \tau \mathbf{B}. \end{aligned}$$

Note: $h_{syn}(s)$ is determined by intrinsic synaptic properties. A' and B' by system dynamics.

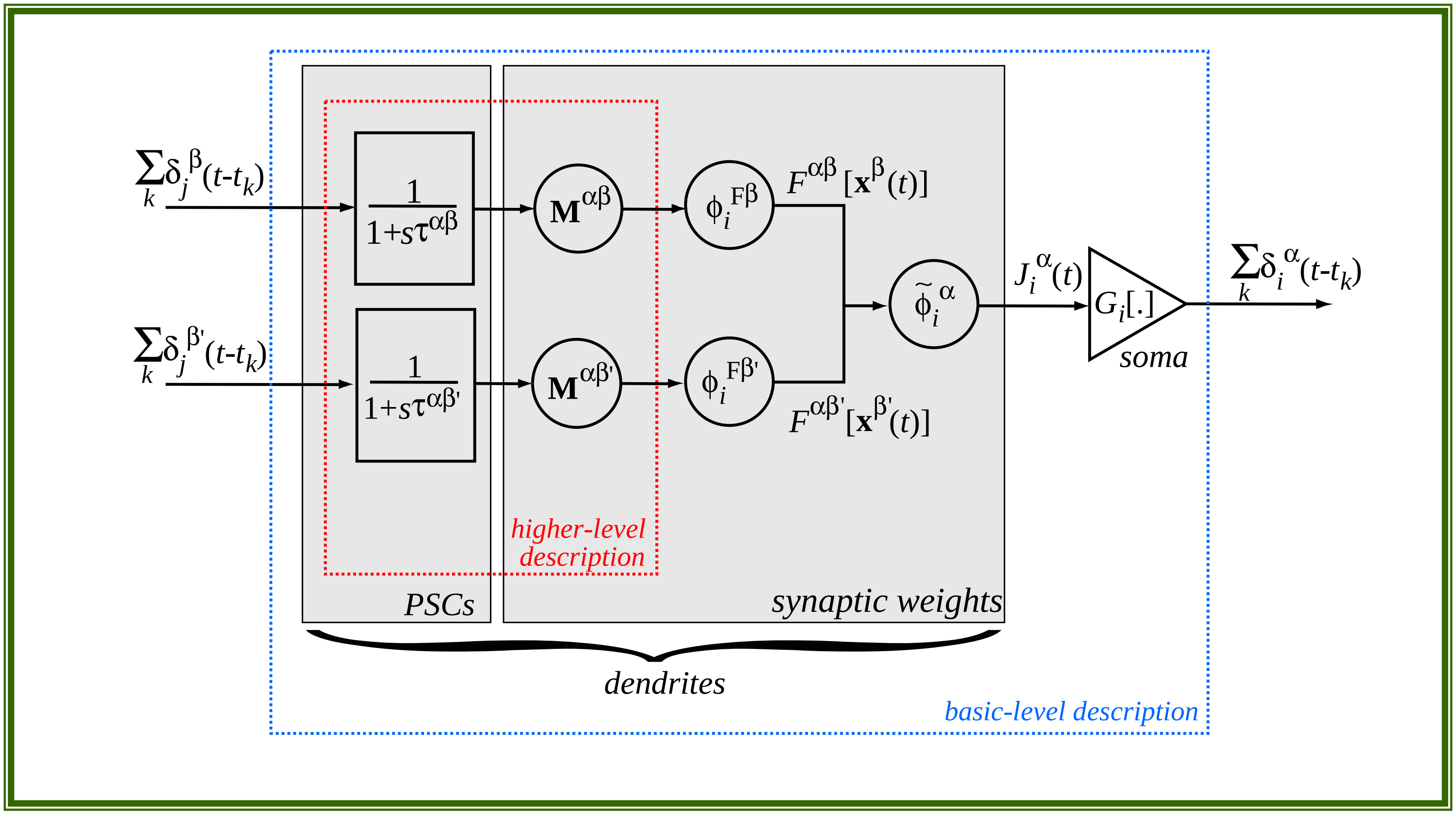
Theoretical framework

- *Synthesis*: A generic neural subsystem



Theoretical framework

- *Synthesis*: A generic neural subsystem



Some specific examples

- Vestibular system
  - Large-scale model, nonlinear computation (cross product) and control
- Eye control (NPH-VN)
  - Line attractor, nonlinear control
- Working memory (LIP)
  - Plane attractor, nonlinear control
- Zebrafish swimming
  - Cyclic attractor, guaranteed dynamics

Blatant plugs

- Book:
  Eliasmith and Anderson (2003) Neural Engineering: Computation, Representation, and Dynamics in Neurobiological Systems, MIT Press.

- Web Site:
  http://compneuro.uwaterloo.ca

- NESim:
  Free Matlab/Java based simulation package