

SYDE 556/750
Simulating Neurobiological Systems
Lecture 7: Temporal Basis Functions

Andreas Stöckel

Based on lecture notes by
Chris Eliasmith and Terrence C. Stewart

February 13, 2020



Accompanying Readings: See References

Contents

| | |
|---|----------|
| 1 Introduction | 1 |
| 2 Representing Functions | 2 |
| 2.1 Sampling | 2 |
| 2.2 Basis Functions | 2 |
| 3 Representing the Past: The Delay Network | 3 |

1 Introduction

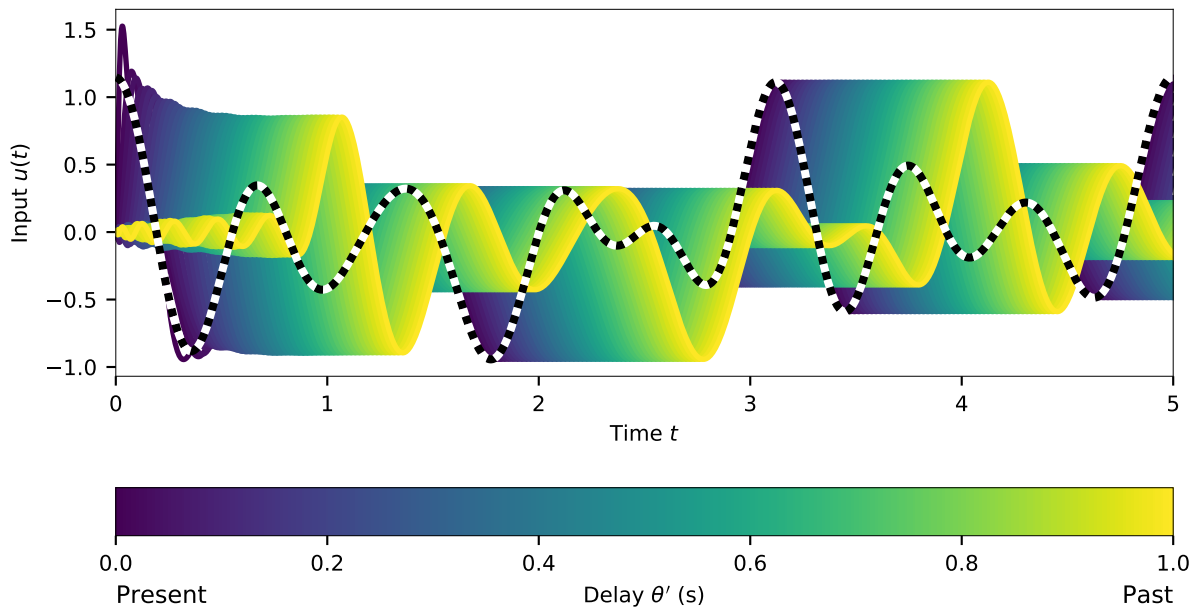


Figure 1: Example of the “delay network” we are going to discuss in the lecture. The dotted line is the input $u(t)$. The coloured lines correspond to delayed versions of the input signal. These delayed versions are all decoded from the same function representation $f_{[t-\theta, t]}(\theta')$. This diagram has been generated by decoding from a mathematically perfect implementation of the delay network. [Code](#)



Note: In this lecture we are going to discuss a very recent addition to the Neural Engineering Framework, the so called Delay Network developed by Aaron Voelker, first presented in [1], and discussed in more detail in his PhD thesis [2]. The Delay Network allows us to represent a function over time in a neural population. A more general version of the Delay Network has been presented at NeurIPS 2019 as the “Legendre Memory Unit” (LMU) [3]. The LMU can be used as a component within artificial Deep Neural Networks and vastly outperforms other recurrent architectures such as LSTMs in a variety of tasks.

As humans, we often feel as if we have a good recollection of events that happened in the immediate past, where “immediate” refers to events happening within the last few seconds. As events move into the more distant past, it becomes increasingly harder to recall details. In this lecture, we are going to discuss a system that similarly memorizes stimuli, remembers them for a certain period, and then gradually forgets them.

Mathematically speaking, what we would like to have a function $f_{[t-\theta, t]}(\theta')$ which allows us to access stimuli $u(t)$ in a time-interval from $[t - \theta, t]$, that is

$$f_{[t-\theta, t]}(\theta') = u(t - \theta'), \text{ where } 0 \leq \theta' \leq \theta.$$

Put differently $f_{[t-\theta, t]}(0)$ will return the present stimulus $u(t)$, whereas $f_{[t-\theta, t]}(\theta)$ will return the input θ seconds from the past.

We would like to build a biologically plausible version of this system. That is, we would like to represent information about the immediate past in a population of neurons $f_{[t-\theta, t]}$. In order to do this, we have to solve two problems

1. **Function representation.** So far we have seen how we can represent vectorial quantities in a neural population. But how can we represent an entire *function* of the form $f_{[t-\theta, t]}(\theta')$ in a neural ensemble?
2. **Updating the function representation.** If we were somehow able to represent functions in our neural population we still need to know how exactly to update this representation over time to get the de

We discuss these two issues separately.

2 Representing Functions

- **Idea:** We know how to represent vectors within a neuron population \mathbf{x} . Can we represent a function as a vector?
- Two approaches:
 - Sampling/discretisation
 - Basis functions

2.1 Sampling

- **Idea:** Measure the value of the function $f(t)$ at discrete points $x_1 = f(t_1), \dots, x_N = f(t_N)$, where $t_i = \Delta t i$. Represent x_1, \dots, x_N as a vector $\mathbf{x} \in \mathbb{R}^N$.
- **Problem:** For a general function $f(t)$ in an interval we need infinitely many samples, even for a time-slice $[0, T]$. We cannot guarantee being able to perfectly reconstruct $f(t)$ with fewer samples.
- **The Nyquist-Shannon Sampling Theorem:** If $f(x)$ contains no frequencies greater than B then it is *completely* determined by samples spaced $\Delta t = \frac{1}{2B}$ apart ($N = 2BT$ equally spaced samples for a time-slice $[0, T]$). There is a *one-to-one* mapping between the samples \mathbf{x} and the function $f(x)$.

2.2 Basis Functions

- **Idea:** $\hat{f}(t) = \sum_{i=1}^q x_i \phi_i(t)$, where the $\phi_i(t)$, are a set of *basis functions*.
- The coefficients $\mathbf{x} = (x_1, \dots, x_q)$ form a vector that could be represented in a neural ensemble.

- We can use a linear transformation to compute $f(t)$ at a single point t

$$f(t) = \langle \mathbf{a}, \mathbf{x} \rangle, \quad \text{where } a_i = \phi_i(t).$$

- We can compute the \mathbf{x} approximating a certain $f(t)$ by discretizing $f(t)$, $\phi_1(t), \dots, \phi_q(t)$ and using least squares.
- Example Function Spaces
 - Discrete Fourier Basis
 - Discrete Cosine Basis
 - Legendre Polynomials

$$\phi_i(t) = \tilde{P}_i(t) = (-1)^{i-1} \sum_{k=0}^{i-1} \binom{n}{k} \binom{n+k}{k} (-t)^k$$

- Neural Tuning Curves (this is exactly what we have done so far)
- **Remark:** Sampling can be seen as a special case where $\phi_i(t) = \delta(t - \Delta ti)$.

3 Representing the Past: The Delay Network

- **Motivation:** Implement a perfect delay of a time θ in the Neural Engineering Framework.
- This can be seen as a dynamical system. In the Laplace Domain, a perfect delay is $e^{-s\theta}$.
- Use Padé approximants up to a degree q to compute an LTI system approximating $e^{-s\theta}$. This system has a one-dimensional input $u(t)$ and an internal state $\mathbf{x}(t)$ of dimension q :

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

$$\theta \mathbf{A} = a_{ij} \in \mathbb{R}^{q \times q}, \quad a_{ij} = \begin{cases} (2i+1)(-1) & i < j, \\ (2i+1)(-1)^{i-j+1} & i \geq j, \end{cases}$$

$$\theta \mathbf{B} = b_i \in \mathbb{R}^q, \quad b_i = (2i+1)(-1)^i.$$

- We can implement this LTI system as a neural ensemble using the transformation

$$\mathbf{A}' = \tau \mathbf{A} + \mathbf{I},$$

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

- The state $\mathbf{x} \in \mathbb{R}^q$ represents more information than just the input θ seconds ago. It represents the state at every point in time up to θ seconds.
- In fact, \mathbf{x} represents the function $f_{[t-\theta, t]} \left(\frac{\theta'}{\theta} \right) \approx u(t' - \theta')$ in the Legendre basis

$$u(t' - \theta') \approx f_{[t-\theta, t]} \left(\frac{\theta'}{\theta} \right) = \sum_{i=1}^q \tilde{P}_i \left(\frac{\theta'}{\theta} \right) x_i(t).$$

- When representing \mathbf{x} in neurons, we can not only decode delays but any function using information from the past θ seconds.
- This network implements an optimal recurrent neural network remembering a slice of the past, as so called “reservoir”.

References

- [1] Aaron R. Voelker and Chris Eliasmith. “Improving Spiking Dynamical Networks: Accurate Delays, Higher-Order Synapses, and Time Cells”. In: *Neural Computation* 30.3 (Mar. 2018), pp. 569–609. DOI: 10.1162/neco_a_01046. URL: https://www.mitpressjournals.org/doi/abs/10.1162/neco_a_01046.
- [2] Aaron R. Voelker. “Dynamical Systems in Spiking Neuromorphic Hardware”. PhD thesis. Waterloo, ON: University of Waterloo, 2019. URL: <http://hdl.handle.net/10012/14625>.
- [3] Aaron R. Voelker, Ivana Kaji, and Chris Eliasmith. “Legendre Memory Units: Continuous-Time Representation in Recurrent Neural Networks”. In: *Advances in Neural Information Processing Systems*. 2019.