Representing and Combining Dynamics in Biologically Plausible Neurons

Sean R. Aubin

September 8, 2016

Abstract

Conceptors, although biologically implausible, admirably capture high dimensional dynamical patterns. This report contains a concise overview of Conceptors and describes how the same dynamical pattern approximation and combination can be achieved in a biologically plausible manner using the Neural Engineering Framework. Two methods were compared to Conceptors with this goal in mind: Rhythmic Dynamic Movement Primitives, both with and without point attractors. In terms of representing a dynamic signal, Dynamic Movement Primitives implemented with point attractors were better. In terms of blending between dynamic signals, Conceptors are distinct from Dynamic Movement Primitives, but this usefulness of this is unclear.

1 Introduction

The intent of this project was to replicate the results of Conceptors in the domain of representing and combining dynamic signals using a neural population [5], but instead doing so in a biologically plausible manner by leveraging the Neural Engineering Framework (NEF) [4]. All code used in this report is available at `https://github.com/Seanny123/nef-conceptors`.

1.1 Conceptors

The Conceptor approach to representing dynamics is inspired by Reservoir Computing (RC), where a randomly connected population of neurons (in this implementation tanh rate-based activation functions and a leaky rate memory) are fed back on themselves to create a dynamic system. The update equation for this system is shown in Equation 1.

$$state_i(t+1) = act(W_i^{rec} + W_i^{in} p_{in}(t+1) + b_i) \tag{1}$$

Where t is time-steps, act is the activation function of the neuron, i is the neuron index, $state_i$ is the neuron state, W^{in} are the input weights to the neuron population, W^{rec} are the recurrent weights, b_i is the bias term for the neuron and p_{in} is the input pattern. Note that the magnitude W^{rec} is less than their eigenvalues to maintain a stable system.

In RC, the output weights are set to minimize the error between output signal p_{out} and the intended signal which is some function of the input signal $f(p_{in})$, as shown in Equation 2.

$$\begin{aligned} p_{out}(t) &= W^{out} state(t) \\ err &= (f(p_{in}) - W^{out} state(t))^2 \end{aligned} \tag{2}$$

This equation has many solutions, however in this report ridge regression was used.

The RC approach to dynamic signal representation can only represent one pattern at a time. Conceptors attempt to rectify this. Conceptors also use a recurrently connected group of neurons. However, instead of the recurrent weights being selected randomly and limited in their magnitude, the weights are determined by linear regression such that the system is oscillates without exterior input for all K input patterns. This linear regression problem is phrased as follows:

For stable oscillation, the state with input should approximate the state without input.

$$act(W^{rec} x^j(t) + W^{in} p^j(t+1) + b) \approx act(W^{rec} + b)$$

Thus the recurrent weights can be found from this minimization which is solved with linear regression.

$$W^{rec} = argmin_{W^{rec}} \sum_{j=1,\ldots,K} \sum_{t=1,\ldots,t^{max}} (W^{rec} + W^{in} p^j(t+1) + b)$$

This dynamic system can then reproduce specific dynamic patterns by the modification of the recurrent weight matrix by various the Conceptor matrix C^j, where j is still the pattern index, such that the update rule becomes:

$$state(t+1) = C^j act(W^{rec} state(t) + b)$$
$$\text{such that, } p_{out} = W^{out} state(t) = p^j_{in}$$

Thus, C^j could also be considered as a "projection matrix" such that the activities of the reservoir are projected onto the correct output pattern.

To find C^j, let $R^j = X^j(X^j)'/L$ be the reservoir state correlation matrix where X^j is the state of the neurons for pattern j for all time steps. Let $\alpha \in (0, \inf)$ be the "aperture", which the fidelity to the original signal that the output should achieve.

Given that C still projects back onto the reservoir, we want to minimize the deviations of the actual state from the projected state.

$$C = E\left[\|x - Cx\|^2\right]$$

As mentioned before, the degree we want C to influence the state of the reservoir is scaled by α. This influence is taken into account by adding a term

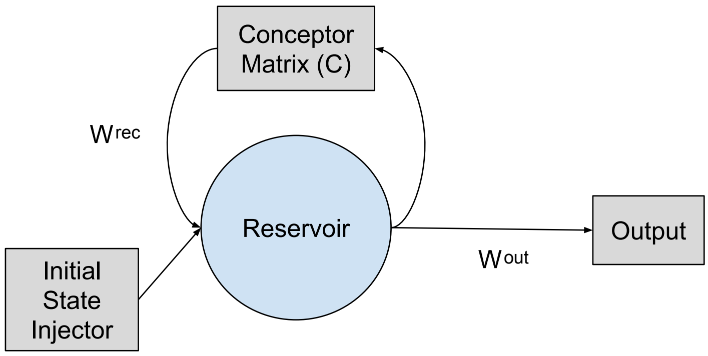


Figure 1: Illustration of Conceptor system. Grey boxes are non-neural components, blue circles are neural components and unlabeled arrows are direct connections to or from neurons. The "Initial State Injector" is required to initialize the neural system so it does not remain inactive, but this value can be any arbitrary value that activates the majority of the neurons.

$$C = E\left[\|x - Cx\|^2\right] + \alpha^{-2}\|x - Cx\|_{fro}^2$$

Where $\|x - Cx\|_{fro}^2$ is the Frobenius norm.

Using a derivation contained in Section 5.1 of [5] it is proven that C and R have the same principal component vector orientations. Thus,

$$\begin{aligned} \mathrm{SVD}\,(R) &= U\Sigma U' \\ \mathrm{SVD}\,(C) &= USU' \end{aligned} \tag{3}$$

Additionally, the singular values of C are related to the singular values of R by

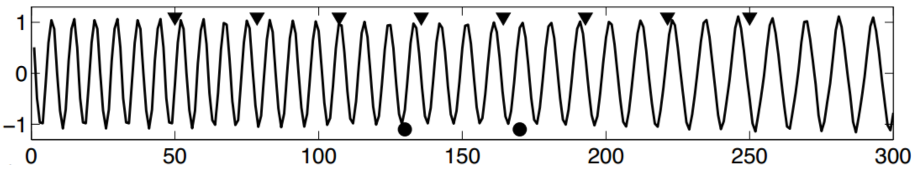
$$s_i = \frac{\sigma_i}{\sigma_i + \alpha^{-2}} \tag{4}$$

Equations 3 and 4 provides us a way to derive C, since R is known. An illustration of the formulated Conceptor system is shown in Figure 1.

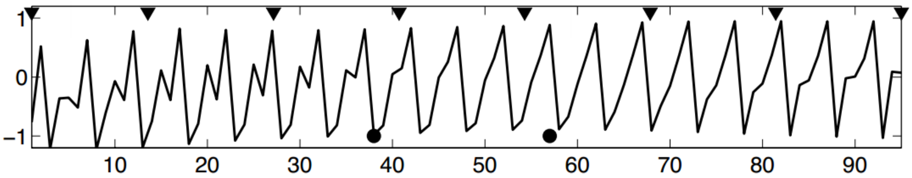
To switch between signal, different values of C must be swapped out. This constant dramatic modification of the recurrent weights, which are mapped onto synaptic weights, makes this approach biologically implausible, since synaptic weights generally undergo gradual changes.

1.1.1 Similarity to Principle 3 of the NEF

Aside from the aforementioned swapping out of C, the formulation of Conceptors is remarkably similar to Control Theoretic formulation of Principle 3 of the NEF. Specifically, a C can be thought of as the stand-in for A in the standard control equation $\frac{dx}{dt} = Ax$, where A is the transform that dictates the next neural state $\frac{dx}{dt}$ based off the current neural state x such that the output pattern can continue to be maintained. Finding A is a System Identification problem which



(a) Blending between two sinusoids.



(b) Blending between non-sinusoidal signals.

Figure 2: Blending between two signals by modifying μ. The black dots are points were $\mu = 1$ and $\mu = 0$, thus the signals appear in their unmodified form. The x axis is time steps, while the y-axis is the output. Figures reproduced from the original Conceptor publication [5].

can be formulated as a least-squared problem. Additionally, one of the possible solutions to the least-squares problem to find A is SVD, which is the same method to find C. Consequently, it seems justifiable to say that Conceptors are a reformulation of Principle 3 of the NEF, but without the usual use of axonal low-pass filters to create system dynamics.

1.1.2 Blending Between Conceptors

In addition to replicating signals, Conceptors are able to blend between signals by mixing Conceptors using a scaling factor μ, such that:

$$C = ((1 - \mu)C^a + \mu C^b)$$

For sin-waves this gives the effect of blending between frequencies, while more irregular signals have less elegantly describable transitions, as shown in Figure 2.

The ability to reliably recreate and blend dynamic signals in a biologically plausible manner is essential for a complete cognitive system. This report describes how this aforementioned goal was achieved using the Neural Engineering Framework (NEF).

Note that the other features of Conceptors, such as their relation to Boolean algebra, online adaptation and classification, are outside of the scope of this report.

2 Methods

A modified reference implementation of Conceptors implemented in Matlab [1] was compared to two approaches using the NEF in Nengo [2].

2.1 Biologically Plausible Alternatives to Conceptors

The two approaches investigated using the NEF and Nengo were Rhythmic Dynamic Primitives, either decoded directly from an oscillator or decoded using forcing function taking into account a point attractor.

2.2 Rhythmic Dynamic Movement Primitives

Dynamic Movement Primitives (DMPs) are a way of planning movement using dynamics. To achieve this, weighted Gaussian basis functions approximating a path dictate the forces on a moving point as it goes from the starting point to the finishing point implemented by a point attractor [3]. This can be described in mathematical terms, as a point attractor modified by a force:

$$\ddot{y} = \alpha_y(\beta_y(g - y) - \dot{y}) + f \tag{5}$$

Where is y is the position of the moving point in response to the point attractor forces, g is the position of the point attractor, α_y and β_y are gain terms of the point attractor, and f is the forcing function that modifies the path that the moving point follows on it's way to the point attractor.

The forcing function is typically implemented by weighted Gaussian basis function that are sequentially activated by a ramp function. In a neural implementation, the basis functions are leaky-integrate-fire neuron tuning curves and the weights are the decoding weights that correspond to the aforementioned neurons.

To imitate a desired path, which is the goal of this DMP use case, the forcing function needs to be solved in terms of the given path y and the point attractor location and gains by using Equation 5.

Rhythmic Dynamic Movement Primitives (rDMPs) are a variation of DMPs, where instead of having a discrete goal, a repeating path is followed. Consequently, instead of a ramp function activating the basis functions, a repeating ramp is used by taking the $arctan$ of an oscillator moving around the circle and spacing the basis functions between $-\pi$ and π.

For a simpler implementation of rDMPs, it is possible to remove the point attractor from Equation 5. This allows for a path to still be defined via a forcing function which can be calculated by the usual NEF method of decoding a transform given an input signal and a desired output signal. However, it should be noted that since the forcing dynamics are arbitrary, there is no guarantee for this system to converge and there is no ability to change these dynamics on the

[1] Code that was originally used to generate the results seen in `https://youtu.be/DkS_Yw1ldD4` was stripped down to it's basic components and various bugs were corrected. Thus I refer to the code I used as a 'modified' reference implementation.

[2] Note that a reference implementation of Conceptors in Nengo was attempted, but ultimately failed due to being unable to implement a population that would sustain a repeating pattern via oscillation. The cause of this failure is unknown.

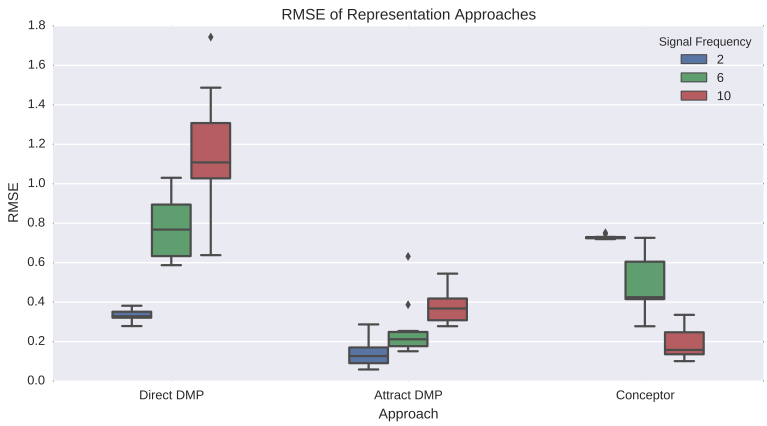


Figure 3: RMSE of various signal representation approaches across different frequencies of a sinusoid signal. Note how the Conceptor RMSE is inversely proportional to the signal frequency, which is the opposite of the DMP based approaches

fly, as there is when using the system defined by Equation 5. This implementation will be refered to as Direct DMPs (dDMPs) while the aforementioned attractor-based implementation will be referred to as (aDMPs).

2.3 Switching Between Dynamic Movement Primitives

To switch between the rDMPs, two methods were attempted. The first was simply to switch between two output patterns by inhibiting one while releasing the inhibition on the other. The second was to multiply the output of each pattern by a scalar in a similar manner as μ in the Conceptor case.

3 Comparison of Results

The representational ability and blending ability of each were the metrics for comparison. For representational ability, the Root Mean Square Error (RMSE) from the original signal, after phase shifts, was the evaluation metric. For blending, the transitions between two signals ($\sin(2\pi 6t)$ and $0.5\cos(2\pi 10t)$), given a fixed time of 1 second to transition, were compared qualitatively in terms of smoothness of transition.

3.1 Representing Signals

For sinusoidal signals, the ideal method for representation depended on the frequency of the signal. For low frequency signals, aDMPs were far superior. For high frequency signals, Conceptors were much better at representing sinusoidal signals, as shown in Figure 3. This is partially because Conceptors fail to reach the desired amplitude at low frequencies, as can be seen in the comparison in Figure 4. The reason for this is unknown.

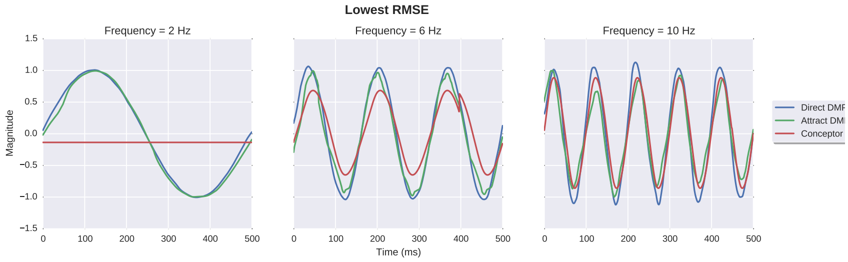


Figure 4: Generated signals at each frequency with the lowest RMSE for each representation method. Notice how although the dDMP is smoother, because of it's phase errors, it actually has a higher RMSE than the aDMP approach.

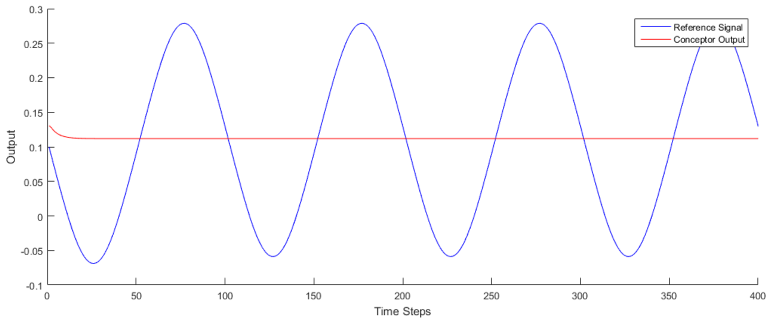


Figure 5: Conceptor failing to replicate the input signal of a low-pass filtered sawtooth wave.

Despite the results shown in Figure 2. I found no evidence for Conceptors being able to approximate jagged signals in any way, regardless of number of neurons, sparsity of recurrent weights, aperture values, frequency of signal, neuron memory or signal magnitude range. I tested this by setting the input signal to a saw-tooth wave. Even when filtered with a low-pass filter, the Conceptor still failed to replicate the oscillation, as shown in Figure 5. This is probably because the activities of the neurons were too limited. Given that the saw-tooth wave can only be expressed as an infinite series of sinusoids, the frequencies required were not found during the Conceptor calculation and the approximation failed.

3.2 Blending Between Signals

The blending, shown in Figure 6, accomplished by Conceptors by changing μ linearly from 1 to 0, is distinct from the blending accomplished with dDMPs. Additionally, it does seem to replicate the results in Figure 2, other than the aforementioned magnitude problems. However, once the frequency of the two target signals was increased, blending also caused a reduction in signal magnitude at the interim frequencies to the point that no signal is output (Figure 7), showing that the description of blending being frequency control is too simplistic.

It is unclear whether this blending behaviour is actually useful. When scaled

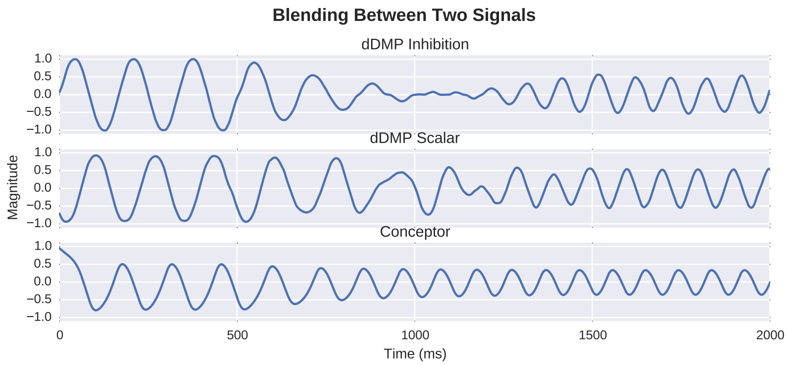


Figure 6: Comparison of approaches to blending between two signals.

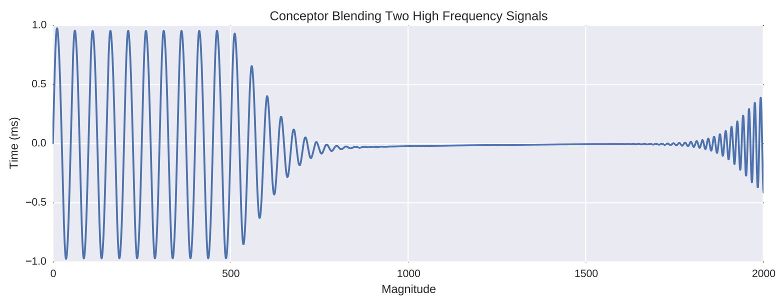


Figure 7: Drastic loss of magnitude when using Conceptors to blend between two high frequency signals.

up to 61D to represent a human skeleton [5], where each Conceptor controls a dimension which represents an angle between joints with no external forces acting on the skeleton, the transition between various movements patterns appear jerky and hesitant [3]. This is even after much fine manual tuning of the transitions between different patterns. Consequently, final judgment on what an ideal blending looks like should be withheld until an application is chosen, such as human transitions between movement patterns [1]. The investigation of these various applications and comparison to the various blending methodologies are outside of the scope of this report.

4 Discussion

This report demonstrated that the dynamic signal representation and combination, previously accomplished Conceptors, could be replicated to a certain degree in a biologically plausible manner using the NEF. It also highlighted the limitations of the various approaches when representing signals in the NEF. Ultimately, Conceptors far surpassed the NEF in representation of high-frequency sinusoidal waves, but were neither to represent non-sinusoidal periodic signals, nor low-frequency periodic signals. Future work should focus on transferring Conceptors to Nengo to see if switching between Conceptor matrices can be accomplished via inhibition and with spiking neurons, as well as exploring Conceptors using Systems Characterization to diagnose the cause of their various deficiencies.

The preliminary biological plausibility allows for many advantages, including comparison with neuroanatomical mapping. For example, the DMP populations of this model could map onto the Central Pattern Generators [2], which are neural circuits in the human spinal chord that assist with the creation of rhythmic movements. Obviously, a sophisticated model of human movement needs to be consulted, to match neurological data exactly, to store the patterns more realistically and to match the dynamics. That being said, at least the approach with the NEF offers this possibility, whereas Conceptors do not.

References

[1] Bruce Abernethy. *Biophysical foundations of human movement.* Human Kinetics, 2013.

[2] Simon M Danner, Ursula S Hofstoetter, Brigitta Freundl, Heinrich Binder, Winfried Mayr, Frank Rattay, and Karen Minassian. Human spinal locomotor control is based on flexibly organized burst generators. *Brain*, 138(3):577–588, 2015.

[3] Travis DeWolf. A neural model of the motor control system. 2015.

[4] Chris Eliasmith and Charles H. Anderson. *Neural engineering: Computation, representation, and dynamics in neurobiological systems.* MIT Press, Cambridge, MA, 2003.

[3] To examine the output, see the video `https://www.youtube.com/watch?v=DkS_Yw1ldD4`

[5] Herbert Jaeger. Controlling recurrent neural networks by conceptors. *arXiv preprint arXiv:1403.3369*, 2014.