Online State Space Filtering of Biosignals using Neural Network-Augmented Kalman Filter

Yu Yao
Translational Neuromodeling Unit
University of Zurich & ETH Zurich
Zurich, Switzerland
yao@biomed.ee.ethz.ch

Guanghao Sun and Tetsuo Kirimoto
Graduate School of Informatics and Engineering
The University of Electro-Communications
Tokyo, Japan
guanghao.sun@uec.ac.jp

Takemi Matsui
Graduate School of System Design
Tokyo Metropolitan University
Tokyo, Japan

Michael Schiek
ZEA-2 – Electronic Systems
Forschungszentrum Jülich
Jülich, Germany

***Abstract*—Since its inception, the Kalman filter, which represents the optimal estimator for linear, Gaussian state space models, has been adopted for a wide array of practical applications due to its efficient, recursive formulation. At the same time, the field of nonlinear time series analysis has produced powerful methods for filtering time series with deterministic dynamics. These nonlinear filters differ significantly from linear filters, because they transform the time series via delay embedding and exploiting geometric features in delay space. However, they are not capable of operating recursively like the Kalman filter. In this paper, we propose a state space filter based on the delay embedding principle, but capable of online estimation. This is achieved by formulating the nonlinear delay space filter as a state estimation problem, which can be solved using the extended Kalman filter. In order for this reformulation to work, it is necessary to approximate the dynamics of the time series. For this purpose, we use a feed-forward neural network. By embedding the neural network weights in the Kalman filter state, we are able to simultaneously estimate the hidden dynamics of the time series and perform online state space filtering. We present preliminary performance estimates of our online state space filtering approach obtained from tests with artificial biomedical time series.**

***Index Terms*— Kalman filter; artificial neural network; state space filtering; delay embedding**

I. INTRODUCTION

In 1960, a paper published by R.E. Kalman led to the development of what is today known as the Kalman filter (KF), the optimal estimator for linear, Gaussian state space models [1]. Due to its computational efficiency and applicability to a wide range of applications, the KF enjoyed enormous success.

About one decade later, the work of Lorenz, Ruelle and Takens on strange attractors set off a wave of interest in the field of nonlinear dynamics and chaos, which eventually culminated in the creation of a wide array of techniques for nonlinear time series analysis [2], including methods for nonlinear noise reduction in delay space [3].

This parallel development of efficient linear filters with widespread adoption in technical domains on one side and powerful nonlinear filters with unique properties on the other side continues until the present day [2]. In this paper, we attempt to combine the advantages from both fields with the aim of developing an efficient, online state space filtering method for biomedical time series.

Building on the concepts of delay embedding and delay space, we reformulate the nonlinear filtering problem as nonlinear state estimation, which allows us to apply the extended Kalman filter (EKF) for efficient, online state estimation. In order to bridge the delay embedding idea with the EKF formalism, we make use of artificial neural networks as a flexible regression method. This provides us with an algorithm which performs online state space filtering and simultaneously models the dynamics of the time series.

This paper is organized as follows. In section II, we first introduce the three topics of nonlinear filtering, extended Kalman filter and artificial neural networks. Due to space constraints, we will restrict ourselves to a basic introduction and provide pointers to the literature. Then, we describe the details of our state space filtering framework and highlight the differences between our approach and related work. Section III presents experimental results obtained with our state space filtering framework. Finally, we close with a conclusion and summary in section IV.

II. METHODS

In this section, we describe our approach to online state space filtering. First, basic introductions are provided for nonlinear filtering, extended Kalman filters and neural networks, which form the fundamental building blocks of our state space filtering framework introduced thereafter. A discussion of the differences between our approach and related work concludes this section.

978-1-5386-0882-1/17/$31.00 ©2017 IEEE

A. Nonlinear Filtering

Time series generated from deterministic systems have the property that future samples can be predicted from past samples:

$$x_{k+1} = f(x_{k-m}, \dots, x_{k-1}, x_k) \tag{1}$$

This characteristic, which Takens formally described in his famous delay embedding theorem [4], has been exploited by several groups to derive nonlinear noise reduction schemes [3].

These nonlinear filtering methods are based on the observation that as a consequence of (1), delay vectors of a deterministic time series lie on a low-dimensional manifold. Therefore, by identifying this low-dimensional manifold, any perturbation to the time series, e.g. due to measurement noise, can be corrected in delay space by projecting the perturbed delay vectors back onto the manifold. This approach results in a nonlinear filter which differs significantly from linear filters based on Fourier theory (see [2] for an in-depth discussion).

Although, these nonlinear filters were designed for strictly deterministic time series, practical experience has shown that they can be successfully applied to time series generated by biophysical systems such as ECG [5], Ballistocardiography [6] or speech [7].

B. The Extended Kalman Filter

The Kalman filter was originally designed as the optimal estimator for the hidden state of a linear, Gaussian state space model [1]. However, many practical applications are best modelled using nonlinear evolution and observation equations of the form:

$$\begin{aligned} x^{(k)} &= f\left(x^{(k-1)}\right) + n \\ y^{(k)} &= g\left(x^{(k)}\right) \quad + e \end{aligned}$$

where $f(x)$ denotes the system evolution equation, $g(x)$ the observation equation, n the system noise and e the measurement noise. One possibility would be to derive a linear approximation to the nonlinear state space model such that it becomes possible to apply the Kalman filter again. This procedure is known as the extended Kalman filter (EKF) [8] and the linear approximation typically consists of a Taylor expansion:

$$\begin{aligned} x^{(k)} &\approx f\left(\hat{x}^{(k-1)}\right) + F\left(x^{(k-1)} - \hat{x}^{(k-1)}\right) + n \\ y^{(k)} &\approx g\left(\hat{x}^{(k-1)}\right) + G\left(x^{(k)} - \hat{x}^{(k-1)}\right) \quad + e \end{aligned}$$

where $F = \partial f / \partial x$ and $G = \partial g / \partial x$ denote the Jacobian matrices of $f(x)$ and $g(x)$ evaluated at the current state estimate $\hat{x}^{(k-1)}$. Based on this approximation, the EKF time update equations are given by:

$$\begin{aligned} \tilde{x}^{(k)} &= f\left(\hat{x}^{(k-1)}\right) \\ \tilde{P}^{(k)} &= FP^{(k)}F^T + \Sigma_n \end{aligned} \tag{2}$$

and the measurement update equations by:

$$\begin{aligned} K \quad &= \tilde{P}^{(k)}G^T\left(G\tilde{P}^{(k)}G^T + \Sigma_e\right)^{-1} \\ \hat{x}^{(k)} &= \tilde{x}^{(k)} + K\left(y^{(k)} - g\left(\tilde{x}^{(k)}\right)\right) \\ P^{(k)} &= (I - KG)\tilde{P}^{(k)} \end{aligned} \tag{3}$$

Here, K denotes the Kalman gain, $\hat{x}^{(k)}$ the new state estimate and $P^{(k)}$ the state covariance. Σ_n and Σ_e denote the covariance matrices of system and measurement noise, respectively.

For a more detailed introduction to Kalman filtering and a derivation of the update equations provided above, we refer the interested reader to [8].

C. Artificial Neural Networks

Neural networks originated from mathematical models of biological information processing. However, for the purpose of signal processing, they can be viewed as very flexible models for function approximation. One common form of artificial neural networks is given by the so called feed-forward neural network (FNN), which consists of input x, output f_{NN} and hidden units h. The hidden units in a FNN obtain activations by applying a nonlinear activation function to a weighted sum of input units, while each output unit calculates a weighted sum over hidden unit activations. This process can be described by the following set of equations:

$$\begin{aligned} f_{NN}(x; w) &= w_o^T h(x; w) \\ h_i(x; w) \quad &= \tanh\left(w_{h,i}^T x\right) \end{aligned}$$

where $w_{h,i}$ denotes the weights associated with the i[th] hidden unit and w_o the weights associated with the output unit. Furthermore, w is the vector comprising all weights, while h is a vector comprising all hidden unit activations h_i:

$$h(x; w) = \left(h_1(x; w), \dots, h_i(x; w), \dots, h_I(x; w)\right)^T$$

In our example, the nonlinear activation function is given by the hyperbolic tangent, but other choices are possible.

During training of the network, an important quantity is the derivative of the FNN output with respect to the weights, which for an FNN described by the equations above is given by:

$$\begin{aligned} \frac{\partial f_{NN}}{\partial w} &= \left(\frac{\partial f_{NN}}{\partial w_0}, \frac{\partial f_{NN}}{\partial w_{h,1}}, \dots, \frac{\partial f_{NN}}{\partial w_{h,i}}, \dots, \frac{\partial f_{NN}}{\partial w_{h,I}}\right) \\ \frac{\partial f_{NN}}{\partial w_0} &= h^T(x; w) \\ \frac{\partial f_{NN}}{\partial w_{h,i}} &= w_{o,i}\left(1 - h_i^2(x; w)\right)x \end{aligned} \tag{4}$$

The numerical values of the derivative can also be obtained recursively using error backpropagation, which is a popular algorithm for FNN training. A more detailed introduction to neural networks and the backpropagation algorithm is provided in [9].

In addition to the weight derivatives given above, our state space filtering approach also requires the derivative of the FNN output with respect to the input, which is given by:

$$\frac{\partial f_{NN}}{\partial x} = w_o^T diag\left(1 - h(x; w)\right)\left(w_{h,1}, \dots, w_{h,I}\right)^T \tag{5}$$

D. Online State Space Filtering

Having introduced the fundamental building blocks, we now describe the main contribution of this paper.

Based on the ideas behind nonlinear filtering, one can attempt noise reduction by treating the delay vectors of a time series as the state of an EKF, with $f(x)$ from (1) as the state evolution equation. However, in practice one often faces two problems. First, the exact form of $f(x)$ might be unknown. Second, in order to predict the next sample using (1), past samples of the noise free time series x_k are needed as input. But in general, only noisy observations $y_k = x_k + e_k$ are available.

A solution to the first problem is to approximate $f(x)$ with an FNN $f_{NN}(x; w)$, where the weights can be learned online by embedding them into the EKF state. The second problem can be tackled by not taking the FNN input from the observed time series directly, but from the EKF state estimate instead. Thus, the EKF state consists of the concatenation of delay embedded time series vector x and FNN weights w, with system equations given by:

$$
\begin{aligned}
x_1^{(k)} &= f_{NN}\left(x^{(k-1)}; w^{(k-1)}\right) + n_1 \\
x_2^{(k)} &= x_1^{(k-1)} \qquad\qquad\qquad + n_2 \\
&\vdots \\
x_m^{(k)} &= x_{m-1}^{(k-1)} \qquad\qquad\quad\; + n_m \\
w^{(k)} &= w^{(k-1)} \qquad\qquad\quad\;\; + n_w
\end{aligned}
$$

Here, superscripts refer to the time step, while subscripts index the position in the state vector. The variable n denotes the system noise. The observation function is simply given by the first element of the state corrupted by measurement noise e:

$$y^{(k)} = x_1^{(k)} + e.$$

Here, we assume that both system and measurement noise terms are white and Gaussian. In order to apply the EKF, we need to calculate the Jacobians of the state evolution and observation functions, which are given by:

$$
\begin{aligned}
F &= \begin{pmatrix} \partial f_{NN}/\partial x & \partial f_{NN}/\partial w \\ J & 0 \\ 0 & I_W \end{pmatrix} \\
J &= \begin{pmatrix} I_{m-1} & 0 \end{pmatrix} \\
G &= \begin{pmatrix} 1 & 0 & \cdots & 0 \end{pmatrix}
\end{aligned}
$$

Here, I_{m-1} denotes the $m-1 \times m-1$ identity matrix, where m is the length of the delay vector x. Similarly, I_W denotes the $W \times W$ identity matrix, where W is the length of the weight vector w. Additionally, $\partial f_{NN}/\partial x$ and $\partial f_{NN}/\partial w$ denote the derivatives of the FNN output with respect to input and weights given in (5) and (4), respectively. The zeros denote zero matrices of appropriate size.

With F and G defined above, we can now apply the EKF update equations (2) and (3), which provides us with estimates of the noise free time series and simultaneously trains the FNN weights to predict the time series. To start off the estimation process, the delay vector part of the EKF state can be initialized with the first m observations, while the FNN weights part of the state can be initialized with random values similar to standard FNN training procedures. The principle of our online state space filtering approach laid out above is illustrated in Fig. 1 in the form of a flowchart.

E. Difference to Existing Approaches

The work presented in this paper is based on ideas from two distinct fields: nonlinear delay space filtering and EKF based FNN training. However, several key differences distinguish our work from existing approaches in both fields. These differences are highlighted in the following.

Current nonlinear delay space filtering approaches do not attempt to explicitly model the dynamics of the time series (cf. (1)). Instead, they operate directly on the set of all samples from the time series, which implicitly contains all available information on the dynamics [2, 3]. This non-parametric approach has the advantage of being very flexible, but suffers from the drawbacks that (i) it has to store the entire time series in memory and (ii) it can only be applied batch-wise or post-hoc after completion of data acquisition. In contrast, our approach uses a parametric approximation (FNN) to the dynamics of the time series, which allows us to apply efficient, online estimation techniques (i.e. the EKF), eliminating the need for batch processing and reducing the amount of memory required.

Previous attempts to combine Kalman filters with neural networks for time series analysis have either focused on using the FNN to approximate the observation function [10, 11] or have relied on offline training of the FNN weights [12]. Compared to these approaches, our work uses the FNN to model the state evolution, while embedding both the delay vector x and the FNN weights w into the EKF state. This allows us to achieve two goals simultaneously: (i) estimating the noise free time series and (ii) capturing the time series dynamics by training the FNN weights from noisy observations in an online fashion. Although [10-12] solve the first problem, none of these methods can achieve both goals simultaneously.

Another interesting approach is presented in [13], where a nonlinear transformation is applied to the state space in order to render the state evolution linear and to allow the use of the regular Kalman smoother. The parameters of the nonlinear transformation are learned with a method called expectation maximization, which leads to a multi-pass algorithm. This approach differs fundamentally from our work, since we embed the nonlinearity into the state evolution equation and use the EKF to obtain a single-pass, online algorithm.

Finally, it is worth noting that the methods proposed in [10-13] mostly focus on data generated from chaotic time series models such as the Lorenz equations [2], while we developed our method for application to biomedical time series. In order to evaluate its effectiveness in a biosignal processing scenario, we tested the state space filter with artificial data generated from a biomedical time series model.

III. Results

In this section, we present results obtained from preliminary performance tests of the state space filter based on simulated respiration data. Using artificial data offers two important advantages: (i) the ground truth data is available to assess the effectiveness of the filter and (ii) the amount of test data is virtually unlimited. We used a model for mechanically measured respiratory time series to generate 1000s of artificial respiration data [6]. White Gaussian noise was added to the original time

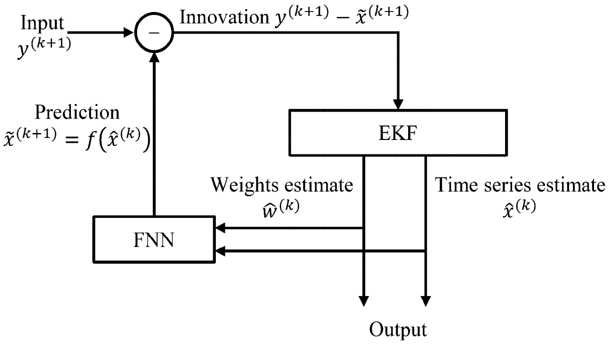

Fig. 1: Flowchart illustrating the working principle behind the proposed online state space filter

series to simulate measurement noise. The noise level was chosen, such that the resulting noisy time series had a signal-to-noise ratio (SNR) of 10dB. A section from the original noise-free time series and the corresponding section from the noisy time series is shown in Fig. 2.

An FNN with 10 inputs, 10 hidden units and one output (10-10-1) was chosen to approximate the state evolution equation. Applying our state space filter with these settings to the noisy time series, an error reduction of 14dB could be observed, which corresponds to a reduction in noise variance of 96%. The online state space filter, required 181.6s to perform recursive, sample-by-sample processing of the noisy time series with a length of 1000s. This is a marked improvement compared to nonlinear filters, which have to rely on batch processing to achieve online performance [6]. Also, since our filter was implemented in Matlab, an additional speed up can be expected when porting the code to a more efficient language like C. Figure 2 compares a section from the original noise-free time series to the same section from the noisy time series before and after filtering.

In Fig. 3, the time evolution of a subset of the FNN weights is shown along with the original noise-free time series. Note how the weights go through a period of strong fluctuation immediately after initialization of the filter and then transition into a more stable phase after about 5s. The onset of the stable phase coincides approximately with the completion of the first respiration cycle. One interpretation of this observation is that in the first phase, the FNN has yet to observe a full respiration cycle. Thus, every sample contributes new information to the FNN training process, leading to high volatility of the weights. With the passage of time and especially after observing one full respiration cycle, the information carried by new samples become more and more redundant. Hence, the weights begin to settle down. However, due to noise and non-stationarity of the signal the fluctuations in the FNN weights will never fully die out.

IV. Conclusion

In this paper, we have introduced a novel concept for online state space filtering of biomedical time series using extended Kalman filter (EKF) augmented with feed-forward neural network (FNN). Our approach is based on the same principles as the delay space filtering approach of [3]. However, by casting delay space filtering into a state estimation framework, the EKF can be used to obtain an efficient, online solution to the filtering problem.

At the same time, our framework captures the dynamics of the observed time series, by also embedding the FNN weights in the EKF state and estimating them from noisy observations. Additionally, if the system and measurement noise, as well as the distribution of the initial estimates are Gaussian, as is the case in most EKF applications, this approach implicitly defines shrinking priors [9] on the FNN weights. In effect, our approach corresponds to FNN training with noisy samples, which stands in contrast to previous attempts to combine Kalman filters (KF) with FNN that use the FNN to approximate the observation function and only include the weights in the KF state [10]. These methods need access to noise free training samples.

However, the use of FNN also gives rise to a number of problems. Due to their nonlinear nature, the error surface of FNN is non-convex, making the training procedure susceptible to local optima [9]. Recent research on deep neural networks suggest that this can be partially mitigated by carefully tuning the initialization procedure [14]. Another potential problem lies in the high number of parameters of typical FNN. This means that (i) a large number of training samples are necessary to train these parameters and (ii) convergence might be slow, leading to a bad approximation during the initial phase right after starting the filtering process. On the one hand, a fairly high number of training samples is available with our approach, since the use of delay embedding means that the number of training samples (corresponding to delay vectors) is approximately equal to the number of samples in the original time series. On the other hand, this does not mitigate slowness of convergence, as subsequent delay vectors are highly correlated [2].

To improve convergence speed and reduce the influence of local optima, offline pretraining of the FNN may be considered. Although, our algorithm is designed for online application, offline pretraining can help to initialize the FNN weights to a more favourable starting point than it is possible with the current random initialization procedure. In particular, the signals used for pretraining could be chosen similar to the signals which the algorithm will be applied to. E.g., in the case of respiratory time series, the FNN can be pre-trained on sinusoidal time series, yielding FNN weights which represent better initial starting points than random weights.

Finally, we would like to note that future work will include testing the state space filter on real world biomedical signals and exploring which types of biosignals our method can be applied to. Given the flexibility of neural networks, we imagine that with the right choice of FNN architecture, the state space filter can be adapted to many different modalities. Candidate modalities include (but are not limited to) PPG, ECG, medical radar or Ballistocardiography.

Acknowledgment

This work was funded in part by the Central Institute ZEA-2—Electronic Systems at Research Center Jülich, Germany (to YY).

References

[1] R. E. Kalman, "A New Approach to Linear Filtering and Prediction Problems," *Journal of Basic Engineering,* vol. 82, no. 1, p. 35, 1960.

[2] H. Kantz and T. Schreiber, *Nonlinear Time Series Analysis*, 1st paperback ed. with. corr, repr. ed. (Cambridge Nonlinear Series, no. 7). Cambridge: Cambridge Univ. Press, 2002, pp. XVI, 304 S.

[3] P. Grassberger, R. Hegger, H. Kantz, and C. Schaffrath, "On noise reduction methods for chaotic data," *Chaos,* vol. 3, no. 2, pp. 127-141, 1993.

[4] F. Takens, "Detecting strange attractors in turbulence," in *Dynamical Systems and Turbulence, Warwick 1980*, vol. 898, D. Rand and L.-S. Young, Eds. (Lecture Notes in Mathematics: Springer Berlin Heidelberg, 1981, pp. 366-381.

[5] T. Schreiber and D. T. Kaplan, "Nonlinear noise reduction for electrocardiograms," *Chaos,* vol. 6, no. 1, pp. 87-92, 1996.

[6] Y. Yao, C. Brüser, U. Pietrzyk, S. Leonhardt, S. v. Waasen, and M. Schiek, "Model-Based Verification of a Non-Linear Separation Scheme for Ballistocardiography," *IEEE Journal of Biomedical and Health Informatics,* vol. 18, no. 1, pp. 174-182, 2014.

[7] R. Hegger, H. Kantz, and L. Matassini, "Noise reduction for human speech signals by local projections in embedding spaces," *IEEE Transactions on Circuits and Systems I-Fundamental Theory and Applications,* Article vol. 48, no. 12, pp. 1454-1461, Dec 2001.

[8] P. Zarchan and H. Musoff, *Fundamentals of Kalman filtering : a practical approach*, 4th ed. (Progress in astronautics and aeronautics). Reston, VA : American Institute of Aeronautics and Astronautics, 2015, p. 876 S.

[9] C. Bishop, *Pattern Recognition and Machine Learning* (Information Science and Statistics). Cambridge: Springer, 2006.

[10] X. Wu, J. Huang, and Z. Song, "Generalized extended Kalman filter for prediction of chaotic time-series with intermittent failures," in *Proceedings of the 7th World Congress on Intelligent Control and Automation (WCICA)*, 2008, pp. 2355-2359.

[11] A. Saptoro, "Extended and unscented kalman filters for artificial neural network modelling of a nonlinear dynamical system," *Theoretical Foundations of Chemical Engineering,* vol. 46, no. 3, pp. 274-278, 05/01/ 2012.

[12] J. Ma and J.-F. Teng, "Predict chaotic time-series using unscented Kalman filter," in *Proceedings of the 2004 International Conference on Machine Learning and Cybernetics*, 2004, vol. 2, pp. 687-690.

[13] L. Ralaivola and F. d'Alche-Buc, "Time series filtering, smoothing and learning using the kernel Kalman filter," in *Proceedings of the 2005 IEEE International Joint Conference on Neural Networks*, 2005, vol. 3, pp. 1449-1454.

[14] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proceedings of the 13th International Conference on Artificial Intelligence and Statistics (AISTATS)*, Sardinia, Italy, 2010, vol. 9, pp. 249-256.

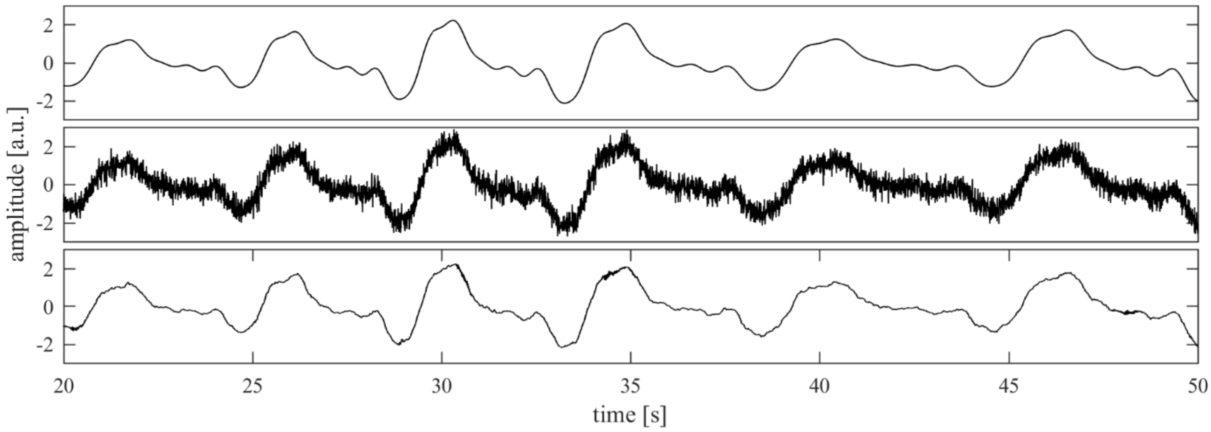

Fig. 2: Samples of the simulated time series. Top: original noise-free signal, middle: noisy signal, bottom: restored signal after filtering.

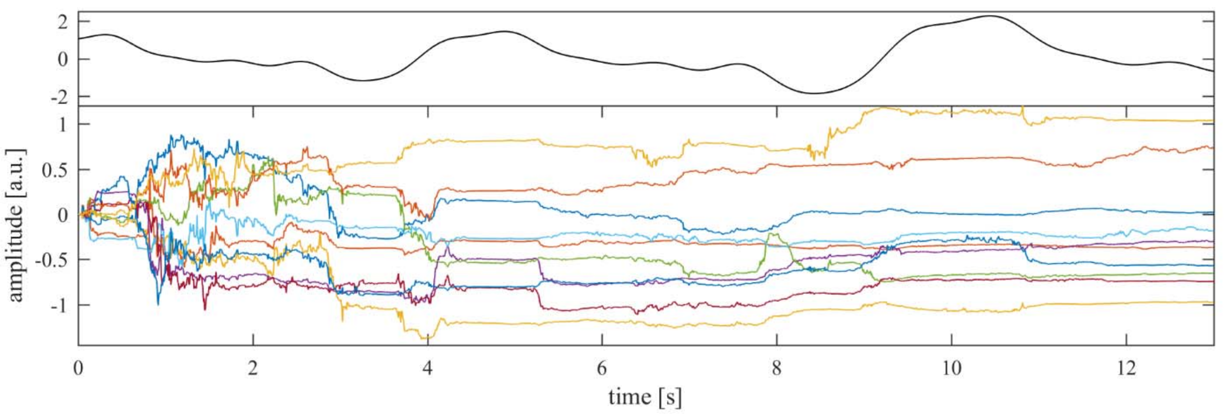

Fig. 3: Top: noise-free signal. Bottom: Time evolution of FNN weight estimates. To avoid cluttering the figure, only a subset of all weights are shown.