

Self-synchronizing Signal Parsing with Spiking Feature-detection Filters

Hans-Andrea Loeliger, Sarah Neff, and Christoph Reller

ETH Zurich

Dept. of Information Technology & Electrical Engineering

{loeliger, neff}@isi.ee.ethz.ch, christoph.reller@gmail.com

Abstract—Following an earlier suggestion, the concept of a hierarchical network of feature-detection filters is developed. The individual filters are derived from a localized least-squares approach based on non-generative state space models, which results in simple forward-only recursions for the actual computations. It is demonstrated that such filters can naturally cope with spiking signals, and the use of spiking signals in such networks is advocated. The feasibility of the approach is demonstrated with a four-layer network that understands Morse code.

I. INTRODUCTION

In [1], it was proposed to use likelihood filters (first proposed in [2]) in a layered network as in Figure 1 for multi-scale pattern detection in time signals. Each block in such a network is looking for some feature in its multichannel-input signal and produces some sort of a score signal (or likelihood signal). The proposal in [1] was not very concrete, though, and no example of such a network was given.

In this paper, we develop this approach further and we will present a nontrivial example of such a network: a four-layer network that “understands” Morse code [3].

As for the individual filter blocks, we follow the general recipe of [1] and use state space models with 2nd-order (i.e., sinusoidal) components, which leads to simple forward-only recursions. However, we develop these filters from a least-squares perspective as in [4] rather than from a statistical perspective as in [1]. The relation between these two perspectives is discussed in [4].

Note that these filters are model-based (cf. Section II), but the models are not generative. Also, these filters could, in principle, be implemented in clockless continuous-time analog circuits. By their very nature, the proposed networks are self-synchronizing and they tolerate small variations in the timing of the input signal.

As it turns out, a key issue with the structure of Figure 1 is the format of the intermediate signals (score signals or likelihood signals). Experiments suggest that the format of these intermediate signals is critical, and some obvious approaches (including those proposed in [1]) do not seem to work well. However, we find that spiking signals—unit pulses separated by spaces of at least some minimum duration—work well, which comes as a surprise to us and is a main point of this paper.

Networks as in Figure 1 may be viewed as a new kind of artificial neural network, and indeed, many connections

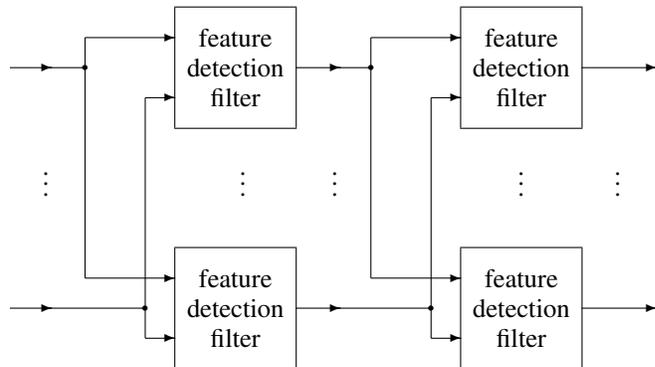


Fig. 1. Layered network of feature-detection filters.

to the vast neural-networks literature are becoming apparent. For example, Figure 1 is vaguely reminiscent of deep convolutional neural networks [5]–[7], which have recently received much interest. Also, analog computation with pulses was studied in [8], and neural network models with spiking neurons are a subject of intense current research [9]–[12]. However, the approach of this paper is new and it offers a new perspective on pulse-based information processing that is firmly rooted in classical signal processing concepts.

This paper does not address learning at all. In particular, the mentioned Morse code network is handcrafted, not learned.

The paper is structured as follows. The class of feature models used in this paper is described in Section II. A Hilbert space perspective on the corresponding filters is given in Section III. The actual computations in these filters are addressed in Section IV. The application of such filters to pulse patterns is demonstrated in Section V.

While Sections II–V are about individual feature-detection filters, Section VI returns to networks as in Figure 1 and presents the case for spiking intermediate signals. Section VII, finally, describes (in outline) the design of a network that parses Morse code.

II. LOCAL FEATURE MODELS

As in [4], we use local feature models of the following form. Let $y_1, \dots, y_N \in \mathbb{R}^L$ (with $N \gg 1$) be the signal that is to be analyzed. (Only $L = 1$ was considered in [4], but the generalization to $L > 1$ is obvious.) For $k = 0, 1, \dots, N$,

let $x_k \in \mathbb{R}^m$ be a vector that evolves according to

$$x_{k+1} = Ax_k \quad (1)$$

where $A \in \mathbb{R}^{m \times m}$ is a non-singular square matrix. Note that the state x_k at any time k completely determines the whole state trajectory x_0, x_1, \dots, x_N . A corresponding output signal $\tilde{y}_1, \dots, \tilde{y}_N \in \mathbb{R}^L$ is defined by

$$\tilde{y}_k = Cx_k \quad (2)$$

for some matrix $C \in \mathbb{R}^{L \times m}$. At any given time $n \in \{1, 2, \dots, N\}$, we locally fit this model to the given signal y_1, \dots, y_N by forming an estimate

$$\hat{x}_n = \operatorname{argmin}_{x_n \in \mathcal{S}} \sum_{k=1}^n \gamma^{n-k} \|y_k - \tilde{y}_k(x_n)\|^2, \quad (3)$$

where γ is a real parameter with $0 < \gamma < 1$, where

$$\tilde{y}_k(x_n) = CA^{k-n}x_n \quad (4)$$

is the output signal determined by x_n according to (1) and (2), and where $\mathcal{S} \subset \mathbb{R}^m$ is an admissible set for \hat{x}_n . We will mainly be interested in the case where $n \gg 1$ so that boundary effects can be neglected.

The *score signal* of such a model is s_1, \dots, s_N where

$$s_n \triangleq 1 - \frac{\sum_{k=1}^n \gamma^{n-k} \|y_k - \tilde{y}_k(\hat{x}_n)\|^2}{\sum_{k=1}^n \gamma^{n-k} \|y_k\|^2} \quad (5)$$

with \hat{x}_n as in (3). The fraction in (5) is the local normalized squared error. Note that $s_n \leq 1$. If the admissible set \mathcal{S} contains the origin, we also have $s \geq 0$ (because for $\hat{x}_n = 0$, the numerator in (5) coincides with the denominator).

The actual computation of the score signal will be addressed in Section IV.

In this paper, the state transition matrix A of all such feature models is a block diagonal matrix

$$A = \begin{pmatrix} J_1 & 0 & 0 & \dots & 0 \\ 0 & J_2 & 0 & \dots & 0 \\ \vdots & & & & \vdots \\ 0 & 0 & \dots & 0 & J_{m/2} \end{pmatrix} \quad (6)$$

with 2×2 rotation matrices

$$J_\ell \triangleq \begin{pmatrix} \cos(\Omega_\ell) & -\sin(\Omega_\ell) \\ \sin(\Omega_\ell) & \cos(\Omega_\ell) \end{pmatrix} \quad (7)$$

($\ell = 1, \dots, m/2$) on the diagonal. For any fixed time- n state x_n , every component of the vector signal $\tilde{y}_1(x_n), \dots, \tilde{y}_n(x_n)$ is thus a superposition of undamped sinusoids with amplitude and phase determined by x_n .

Example 1 (Sinusoid) Let $L = 1$, $m = 2$, and $\mathcal{S} = \mathbb{R}^m$ (i.e., no restrictions on \hat{x}_n). In this case, the score signal (5) indicates the local presence of a sinusoid with frequency Ω_1 , as illustrated in Figure 2. \square

This example is unimpressive in itself, of course, and more-interesting examples will be described later. We will mostly use models with $L > 1$ and with admissible sets \mathcal{S} of the form

$$\mathcal{S} = \{\beta s : \beta \geq 0\} \quad (8)$$

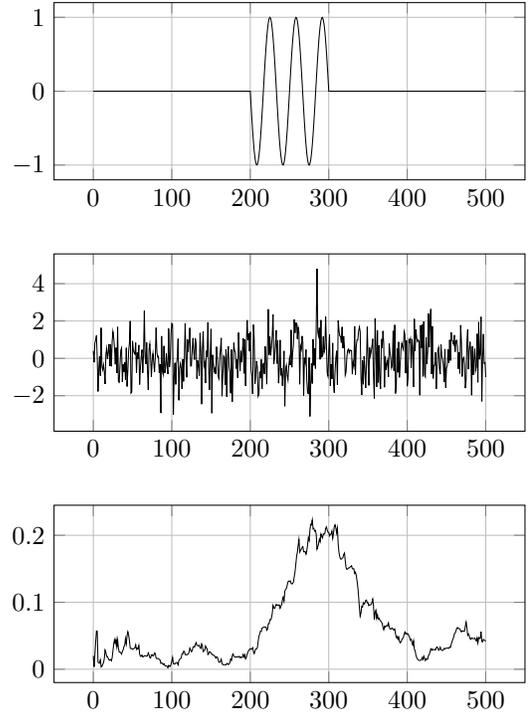


Fig. 2. Detecting a sinusoid in noise as in Example 1. Top: clean signal; middle: same signal with additive white Gaussian noise; bottom: score signal (5) and (19).

for some vector $s \in \mathbb{R}^m$. Note that (8) implies that the signal $\tilde{y}_1, \dots, \tilde{y}_n$ is fully determined up to a nonnegative amplitude.

In this paper, each feature-detection filter in Figure 1 is based on such a model, and the output signal of each feature-detection filter is a thresholded or pulsed version of the corresponding score signal (5) as will be discussed in Section VI.

III. HILBERT SPACE PERSPECTIVE

It is instructive to consider the local model fitting of Section II also from a Hilbert space perspective. Specifically, for any time $n \in \{1, \dots, N\}$, let \mathcal{H}_n be the Hilbert space of discrete-time vector signals $y'_1, \dots, y'_n \in \mathbb{R}^L$ with the inner product

$$\langle y', y'' \rangle \triangleq \sum_{k=1}^n \gamma^{n-k} (y'_k)^\top y''_k. \quad (9)$$

If the set $\mathcal{S} \subset \mathbb{R}^m$ of admissible states x_n is a subspace of \mathbb{R}^m , then the corresponding set of signals

$$\tilde{\mathcal{H}}_n \triangleq \{(\tilde{y}_1(x_n), \dots, \tilde{y}_n(x_n)) : x_n \in \mathcal{S}\} \quad (10)$$

is a subspace of \mathcal{H}_n . It then follows from basic Hilbert space theory [13] that the signal $\tilde{y}_1(\hat{x}_n), \dots, \tilde{y}_n(\hat{x}_n)$ (with \hat{x}_n as in (3)) is the projection of the signal y_1, \dots, y_n to $\tilde{\mathcal{H}}_n$. It follows, in particular, that the mappings

$$y_1, \dots, y_n \mapsto \tilde{y}_1(\hat{x}_n), \dots, \tilde{y}_n(\hat{x}_n) \quad (11)$$

and

$$y_1, \dots, y_n \mapsto \hat{x}_n \quad (12)$$

are linear.

We also remark that, with suitably spaced frequencies $\Omega_1, \dots, \Omega_{m/2}$ and with $\mathcal{S} = \mathbb{R}^m$, the mapping (12) may be viewed as a short-time Fourier transform in the spirit of [14].

Admissible sets \mathcal{S} of the form (8) can be handled by first projecting the signal y_1, \dots, y_n to the one-dimensional space determined by $\hat{x}_n \in \{\beta s : \beta \in \mathbb{R}\}$ and then clipping the projection $\hat{x}_n = \hat{\beta}s$ to $\hat{x}_n = 0$ if $\hat{\beta} < 0$.

IV. RECURSIVE COMPUTATION OF THE SCORE SIGNAL

The score signal (5) can be computed from the quantities $\vec{\kappa}_n \in \mathbb{R}$ (the signal energy), $\vec{\xi}_n \in \mathbb{R}^m$, and $\vec{W}_n \in \mathbb{R}^{m \times m}$ (the precision matrix), which in turn are easily computed by the following recursions from [4] (which are very similar to, but not identical with, classical Kalman filtering and recursive least-squares algorithms [15], [16]):

$$\vec{\kappa}_n = \gamma \vec{\kappa}_{n-1} + \|y_n\|^2 \quad (13)$$

$$\vec{\xi}_n = \gamma (A^{-1})^T \vec{\xi}_{n-1} + C^T y_n \quad (14)$$

$$= \gamma A \vec{\xi}_{n-1} + C^T y_n \quad (15)$$

$$\vec{W}_n = \gamma (A^{-1})^T \vec{W}_{n-1} A^{-1} + C^T C \quad (16)$$

$$= \gamma A \vec{W}_{n-1} A^{-1} + C^T C. \quad (17)$$

For $\mathcal{S} = \mathbb{R}^m$ (i.e., no constraints on \hat{x}_n), we then have

$$\hat{x}_k = \vec{W}^{-1} \vec{\xi}_k \quad (18)$$

and

$$s_n = \frac{\vec{\xi}_n^T \vec{W}_n^{-1} \vec{\xi}_n}{\vec{\kappa}_n}, \quad (19)$$

which can be derived from (55)–(58) in [4]. For \mathcal{S} as in (8), we have

$$s_n = \begin{cases} \frac{(s^T \vec{\xi}_n)^2}{\vec{\kappa}_n s^T \vec{W}_n s}, & \text{if } s^T \vec{\xi}_n > 0 \\ 0, & \text{otherwise,} \end{cases} \quad (20)$$

which can be derived from (61) in [4].

For $n \rightarrow \infty$ (which is the case of primary interest), the matrix \vec{W}_n converges to some constant \vec{W} . The term $s^T \vec{W}_n s$ in (20) is then a mere constant scale factor that may be omitted. In this case, the matrix \vec{W}_n and the corresponding recursion (17) are not needed at all.

V. APPLICATION TO PULSE PATTERNS

We now come to the heart of this paper. We first demonstrate the application of local feature models (as in Section II) to sparse pulse patterns. We will then argue (in Section VI) that sparse-pulse signals are advantageous for the internal feature signals in feature-detection networks as in Figure 1.

For some of the illustrations, it will be helpful to rewrite (3) as

$$\hat{x}_n = \operatorname{argmin}_{x_n \in \mathcal{S}} \sum_{k=1}^n \left\| \gamma^{\frac{n-k}{2}} y_k - \gamma^{\frac{n-k}{2}} \tilde{y}_k(x_n) \right\|^2, \quad (21)$$

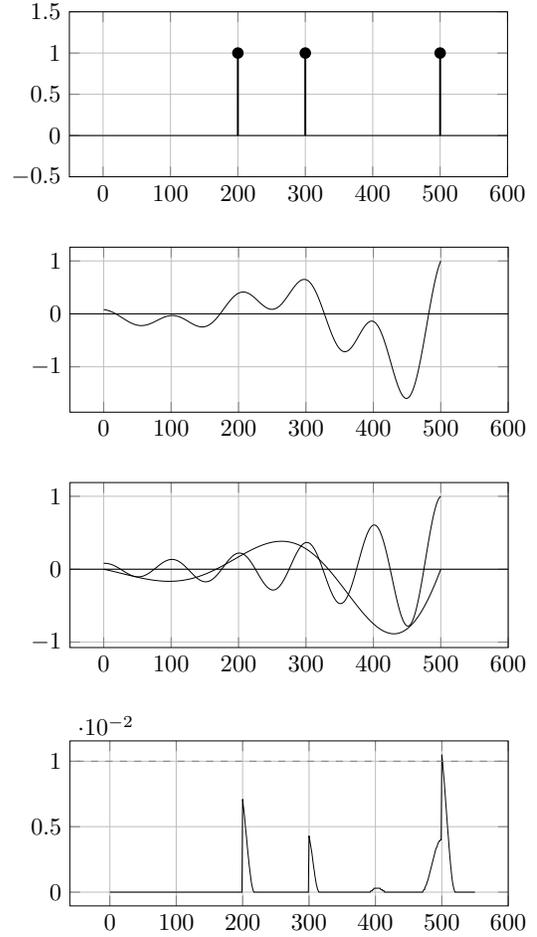


Fig. 3. Pulse pattern detection filter of Example 2. Top: the pattern to be detected. Next two: the signal $\tilde{y}_1, \dots, \tilde{y}_n$ (not to scale) and its decomposition into the two sinusoids, for $n = 500$ and exponentially weighted as in (21). Bottom: the score signal (20) of the clean pulse pattern for $n = 1, \dots, 550$.

i.e., with exponentially damped signals rather than with a damped cost function.

Example 2 (Single-Channel Pulse Pattern) Suppose we want to detect the particular pulse sequence shown in Figure 3 (top). A simple local feature model can be built with $L = 1$, $m = 4$, $\Omega_1 = 2\pi/100$, $\Omega_2 = 2\pi/333.3$, $\gamma = 0.99$, and an admissible set \mathcal{S} as in (8). The two sinusoids and the resulting signal $\tilde{y}_1, \dots, \tilde{y}_n$ (for $n = 500$) are also shown in Figure 3, both exponentially weighted as in (21).

Figure 3 (bottom) shows the resulting score signal when the input signal is the clean pulse pattern. The dashed horizontal line is a suitable threshold for detection: no subset of unit pulses at positions 200, 300, 400, and 500 except the desired one (Figure 3 top) produces a score signal that exceeds this threshold. \square

This example, like all examples in this paper, is hand-crafted (with a little experimentation) and has no claims on optimality of any kind.

Example 3 (Multi-Channel Pulse Train) Suppose we want to detect the particular 3-channel pulse sequence shown in Figure 4: three consecutive pulses (separated by 100 samples) arrive in three different channels. A simple local feature model can be built with $L = 3$, $m = 2$, $\Omega_1 = 2\pi/400$, $\gamma = 0.995$, and an admissible set \mathcal{S}

as in (8). Figure 5 shows the corresponding single-frequency signal $\tilde{y}_1, \dots, \tilde{y}_n$ (for $n = 400$ and exponentially weighted as in (21)) in all three channels.

Figure 6 shows the score signal when the input signal is the clean pulse train of Figure 4. The dashed horizontal line is a suitable threshold for detection. \square

It should be obvious from these examples that short sparse pulse patterns may be detected with quite simple feature-detection filters. Also, small deviations of the pulses from their nominal positions will typically not substantially damage the detectability of the pattern.

While these examples are typical for the Morse code network in Section VII, they may be misleading in several ways. First, good discrimination between different pulse patterns may require many frequencies, i.e., $m \gg 1$. Second, we expect $L \gg m$ to hold for the most interesting feature-detection filters. Third, any such feature-detection filter (with a fixed detection threshold on the score signal) will be deceived by multiple pulses very close together (in the same channel): in both Examples 2 and 3, two pulses at positions k and $k + 1$ (for any k) will make the score signal exceed the threshold indicated in Figure 3 (bottom) and Figure 6, respectively. We will therefore argue in the next section that pulses should be separated by a suitable guard space.

VI. THRESHOLDING AND SPIKING FEATURE SIGNALS

As mentioned in the introduction, the format of the intermediate signals in Figure 1 is an issue. Using the score signal (5) directly does not work well (and the statistical proposals in [1, eq. (10) and (12)] have similar difficulties): variations of the score signals at one layer of the network tend to cause larger variations of the score signals at the next layer, which makes it difficult to achieve robust functionality.

It turns out that the desired robustness can be achieved with thresholded and pulsed versions of the score signal (5) as illustrated in Figure 7. The former amounts to converting the score signal into a $\{0, 1\}$ -valued signal depending on the sign of $s_n - \theta$ where $\theta \in \mathbb{R}$ is a threshold parameter. The latter turns the score signal into a sparse sequence of unit pulses as follows: if the score signal exceeds some threshold, a unit pulse is generated; thereafter, any subsequent pulses are only permitted after a fixed guard space. Note that the pulsed score signal can be obtained from the thresholded score signal, but not vice versa.

The choice between a thresholded score signal and a pulsed score signal depends on the feature: a timeless condition (like the presence of a tone in Example 1) suggests a thresholded score function while a localized event (as in Examples 2 and 3) suggests a pulsed score signal with a suitable guard space such that only one pulse is generated per event. These two types of features are likely to be expressed by different admissible sets \mathcal{S} as illustrated by the mentioned examples.

However, we also note that the thresholded score signal can be approximately simulated by a pulsed score signal with a short guard space (so that typical on-periods of the thresholded signal are simulated by many pulses). By contrast, the pulsed score signal cannot be approximated by

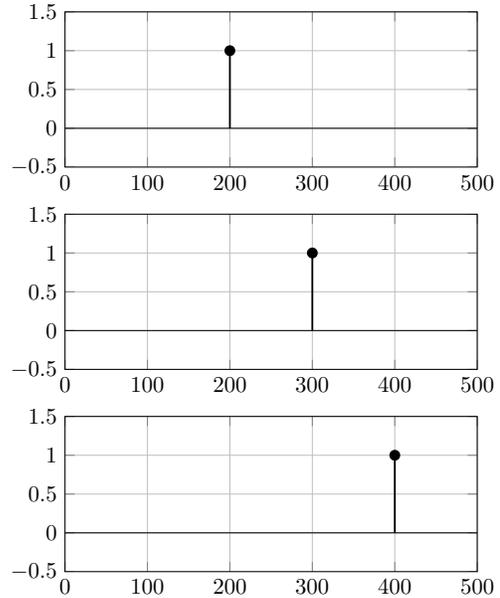


Fig. 4. Three-channel pulse train of Example 3.

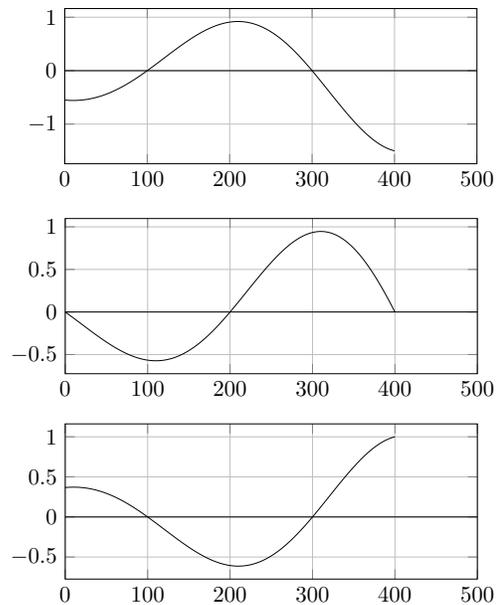


Fig. 5. Example 3: the signal $\tilde{y}_1, \dots, \tilde{y}_n$ (for $n = 400$) in all three channels, up to a common nonnegative scale factor and exponentially weighted as in (21).

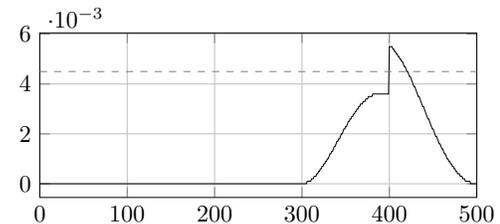


Fig. 6. Example 3: the score signal (20) (for $n = 1, \dots, 500$) of the clean pulse train in Figure 4.

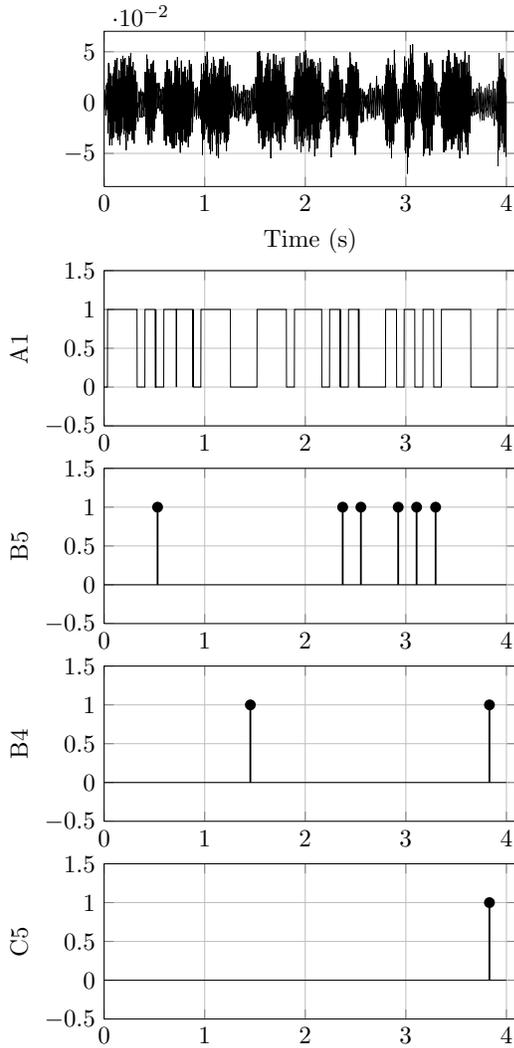


Fig. 7. Morse code network. Top: raw data from microphone. Second: thresholded score signal of first-layer tone detector (A1). Subsequent: pulsed score signal of features B5, B4, and C5.

a thresholded score signal. In this sense, the pulsed version is arguably more fundamental.

VII. EXAMPLE: MORSE CODE DETECTION

Using the principles of the previous sections, we built a four-layer network for parsing Morse code [3]. For this example, Morse signals are represented by on-off keying of a 400-Hz tone, which is transmitted acoustically. A typical waveform as picked up by the microphone in the receiver is shown in Figure 7 (top).

The first layer of our network is simply a tone detector as in Example 1 with a thresholded score signal (referred to as A1) as shown in Figure 7 (second from top). Note that this thresholded signal need not be a perfect Morse code signal since the second layer can correct small glitches in this signal.

Layers 2–4 all produce pulsed score signals as described in Section VI. Three examples of such pulsed score signals (features B4, B5, and C5) are given in Figure 7. The

TABLE I
SECOND-LAYER FEATURE-DETECTION FILTERS

id	feature, letter	input	freq.
B1	pause – dit	A1	$1/3.5T$
B2	pause – dah	A1	$1/8T$
B3	dit – pause	A1	$1/2.5T$
B4	dah – pause	A1	$1/8T$
B5	dit	A1	$1/2.5T$
B6	dah	A1	$1/6T$
B7	E	A1	$1/3T$
B8	T	A1	$1/6T$
B9	pause between words	A1	$1/8T$

TABLE II
THIRD-LAYER FEATURE-DETECTION FILTERS

id	letter	feature	input	freq.
C1			B1, B6	$1/9T$
C2			B1, B5	$1/9T$
C3			B2, B5	$1/6T$
C4			B2, B6	$1/18T$
C5			B5, B4	$1/12T$
C6			B5, B3	$1/2T$
C7			B6, B3	$1/6T$
C8			B6, B4	$1/4T$
C9	A		B1, B4	$1/12T$
C10	I		B1, B3	$1/6T$
C11	N		B2, B3	$1/9T$
C12	M		B2, B4	$1/15T$

TABLE III
FOURTH-LAYER FEATURE-DETECTION FILTERS

id	letter	input	freq.
D1	U	C2, C5	$1/9T$
D2	S	C2, C6	$1/6T$
D3	R	C1, C7	$1/6T$
D4	W	C1, C8	$1/12T$
D5	K	C3, C5	$1/9T$
D6	D	C3, C6	$1/6T$
D7	G	C4, C7	$1/6T$
D8	O	C4, C8	$1/9T$
D9	Q	C4, C5	$1/12T$
D10	Z	C4, C6	$1/9T$
D11	L	C1, C6	$1/9T$
D12	P	C1, C7	$1/12T$
D13	V	C2, C5	$1/15T$
D14	H	C2, C6	$1/9T$
D15	F	C2, C7	$1/12T$
D16	X	C3, C5	$1/15T$
D17	B	C3, C6	$1/9T$
D18	C	C3, C7	$1/12T$
D19	J	C1, C8	$1/16T$
D20	Y	C3, C8	$1/16T$

network's guess of the transmitted letters are also represented by such pulses.

The feature-detection filters in layers 2, 3, and 4 are roughly described in Tables I, II, and III, respectively. All these filters are single-frequency filters (i.e., $m = 2$) with frequencies as indicated in the tables, where the symbol T is the duration of a dit (or dot) or the space between symbols of the same letter.

The filters in Layer 2 are similar to Example 2, except that they use only a single frequency and work on the thresholded signal A1. The filters in Layers 3 and 4 are all very similar to a two-channel version of Example 3.

Note the different time scales in the different layers. The two shortest letters ('E' and 'T') are already detected in Layer 1, but most letters are detected only in Layer 4.

We observe that this network works quite robustly (but without any claim on optimality). It would be easy to add a fifth layer to detect some short sequences of letters like "SOS", and additional layers for longer words.

VIII. CONCLUSION

We have further developed the concept of a hierarchical multichannel feature-detection filter network (as proposed in [1]) and demonstrated its feasibility by a nontrivial example. In contrast to the statistical setting of [1], we developed the individual filters from a least-squares perspective, and we gave explicit formulas ((19) and (20)) for the score signal in two important cases. We demonstrated that such feature-detection filters can easily cope with spiking signals, and we advocate the use of spiking signals inside the network. By their very nature, the proposed networks are self-synchronizing and tolerate small variations in the timing of the input signal.

The important subject of learning such networks from data was not addressed and is left for future research.

REFERENCES

- [1] Ch. Reller, M. V. R. S. Devarakonda, and H.-A. Loeliger, "Glue factors, likelihood computation, and filtering in state space models," *Proc. 50th Annual Allerton Conference on Communication, Control, and Computing*, Monticello, Illinois, USA, Oct. 1–5, 2012.
- [2] H.-A. Loeliger, L. Bolliger, S. Korl, and Ch. Reller, "Localizing, forgetting, and likelihood filtering in state-space models," 2009 Information Theory & Applications Workshop, UCSD, La Jolla, CA, Feb. 8-13, 2009.
- [3] "Morse code" on *Wikipedia*.
- [4] L. Bruderer, H.-A. Loeliger, and N. Zalmi, "Local statistical models from deterministic state space models, likelihood filtering, and local typicality," *Proc. 2014 IEEE Int. Symp. on Information Theory (ISIT)*, Honolulu, Hawaii, June 29 – July 4, 2014, to appear.
- [5] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.
- [6] P. Simard, D. Steinkraus, and J. C. Platt, "Best practices for convolutional neural networks applied to visual document analysis," in *ICDAR*, vol. 3, pp. 958–962, 2003.
- [7] Y. Bengio. 2009. *Learning Deep Architectures for AI*. Foundations and Trends in Machine Learning, vol. 2, no. 1, January 2009, pp. 1–127.
- [8] R. W. Brockett, "Dynamical systems and their associated automata," in *Systems and Networks: Mathematical Theory and Applications* (U. Helmke, R. Mennicken and J. Saurer eds.), vol. 77, pp. 49–69, Akademie-Verlag, Berlin, 1994.
- [9] W. Maass and C. Bishop (eds.), *Pulsed Neural Networks*. Cambridge, MA: MIT Press, 1999.
- [10] E. N. Brown, R. E. Kass, and P. P. Mitra, "Multiple neural spike train data analysis: state-of-the-art and future challenges," *Nature Neuroscience*, vol. 7, no. 8, May 2004, pp. 456–461.
- [11] H. Paugam-Moisy and S. Bohte, "Computing with Spiking Neuron Networks," *Handbook of Natural Computing*, (G. Rozenberg, T. Bäck, and J. N. Kok, eds.) chapt. 10, Springer Verlag, 2011.
- [12] T. Sejnowski and T. Delbruck, "The language of the brain," *Scientific American*, vol. 307, no. 4, pp. 54–59, 2012.
- [13] M. Vetterli, J. Kovačević, and V. Goyal, *Foundations of Signal Processing*. Cambridge Univ. Press, 2014.
- [14] E. Terhardt, "Fourier transformation of time signals: Conceptual revision," *Acustica*, vol. 57, pp. 242–265, 1985.
- [15] T. Kailath, A. H. Sayed, and B. Hassibi, *Linear Estimation*. Prentice Hall, NJ, 2000.
- [16] H.-A. Loeliger, J. Dauwels, Junli Hu, S. Korl, Li Ping, and F. R. Kschischang, "The factor graph approach to model-based signal processing," *Proceedings of the IEEE*, vol. 95, no. 6, pp. 1295–1322, June 2007.