# High Throughput Architecture for Inpainting-Based Recovery of Correlated Neural Signals

Sebastian Schmale, Jochen Rust, Nils Hülsmeier, Heiner Lange, Benjamin Knoop, Steffen Paul
Institute of Electrodynamics and Microelectronics (ITEM.me)
University of Bremen, Otto-Hahn Allee 1, 28359 Bremen, Germany
Email: {schmale, rust, lange, knoop, steffen.paul}@me.uni-bremen.de
Web: http://www.me.item.uni-bremen.de/

***Abstract*—This paper presents the first hardware architecture for compressing and reconstructing correlated neural signals using structure-based inpainting. This novel methodology is especially important for the realization of implantable neural measurement systems (NMS), which are subject to strict constraints in terms of area and energy consumption. Such an implant only requires a defined controlling of the electrode activity to compress neural data. To achieve an efficient implementation with high throughput at the data recovery, approximately computation of arithmetic operations and elementary functions is proposed by using the logarithmic number system (LNS). Because of the digital quantization effects of the LNS conversions, an inherent thresholding operation arises. The proposed hardware realization significantly reduces the required iteration of inpainting computations. This inherent zero forcing in conjunction with the algorithmic error correction results in a speed-up in terms of neural signal recovery, which results in a throughput of 32 961 parallel reconstructions per second.**

## I. Introduction and Related Work

Long term recording of brain activity is utilized in several fields of research and medical diagnostics like the detection of epileptic seizures [1] or controlling prosthetics [2]. In order to constantly monitor the brain activity, implantable systems are getting more and more important. In contrast to traditional measurement equipment, these implants achieve a higher spatial resolution by placing the multi electrode array (MEA) directly on the brain surface. By removing all physical connections through the skull, wireless communication eliminates the risk of an infection and enables remote diagnostics during everyday life of the patient.

The wireless energy supply results in a tight energy constraint for the implant. Transceivers for this highly constrained environment approved for bio-medical applications do not provide enough bandwith to cope with the amount of information generated by a MEA of up to 100 [3] and even 1000 [4] electrodes, which range into several $\mathrm{Mbit/s}$.

To alleviate this problem, compression techniques are a welcome approach at the implant. Data recovery at the external receiver (ex vivo) provides accurate approximations of the original neural signals. Standard methods, like JPEG [5], are unfeasible due to their high algorithmic complexity leading to an unacceptable energy and area consumption. However, more suited compression methods, like Compressed Sensing (CS) [6] or inpainting [7], have proven to be very feasible in this scope. The latter has been shown to outperform CS, e.g.

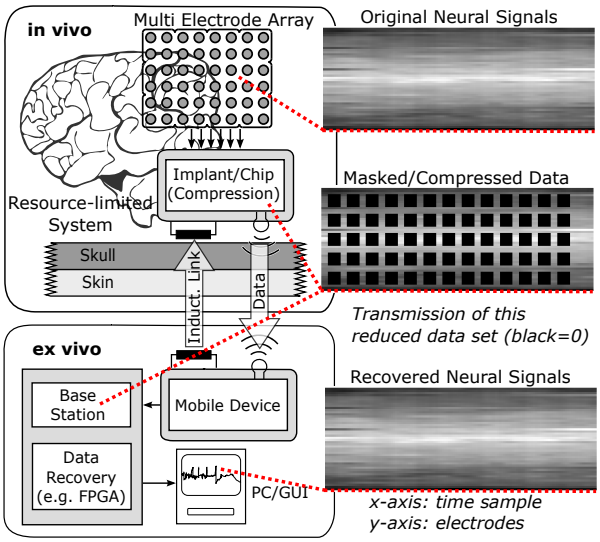



Fig. 1. Schematical design of the inpainting-based compression and reconstruction methodology for brain monitoring. The process of compressing the original neural signals due to the inpainting mask on implant and reconstructing the brain activity e.g. on a FPGA.

in terms of signal recovery quality (see tab. I). These methods also offers the opportunity of computational complexity reduction (CR) [7]. Therefore, it can be implemented within the highly constraint environment of neurological implants because of their low computational load.

This paper presents the hardware architecture of the data reconstruction method using structure-based inpainting [8], which is able to exploit correlation (EC) of neural signals. This novel compression and reconstruction methodology [7], [9], schematical shown in fig. 1, is based on spatial inter-electrode correlation of the neural signals [10]. These signals exhibit an image-related structure, which can be optionally enhanced by correlation-based sorting [7]. The major key of this new data compression technique is the inpainting mask, which corresponds to an electrode activity controller to manage the recording behavior of the MEA [7]. It switches specified electrodes on and off for several time (and spatial) parts of the neural activity recording, as shown in fig. 1. This results in a data compression, as only the remaining signal fragments have to be transmitted. To constantly monitor the brain activity a high performance hardware architecture for data reconstruction is introduced that uses the logarithmic number system (LNS)

978-0-9928-6265-7/16/$31.00 ©2016 IEEE

TABLE I
ERROR, COMPRESSION RATIO, PROPERTY OF COMPUTATIONAL COMPLEXITY REDUCTION AND EXPLOITATION OF SIGNAL CORRELATION OF RECOVERED NEURAL DATA ARE ILLUSTRATED, MODIFIED FROM [7].

| Approach | NMSE | $\sigma_{\text{NMSE}}$ | $\eta$ in % | CR | EC |
|---|---|---|---|---|---|
| **Inpainting** | **0.078** | **±0.012** | **62** | **Yes** | **Yes** |
| CS | 0.170 | ±0.037 | 62 | Yes | No |
| JPEG | 0.017 | ±0.003 | 78 | No | Yes |

[11] for nontrivial function computation. Its major advantage refers to the simplification of several arithmetic operations and elementary functions by processing the corresponding exponential concatenations. Hence, logarithmic (LOG) and antilogarithmic (ALOG) converters are implemented for number system transformation between LNS and fixed-point number system (FPNS). In order to minimize the processing time and the hardware complexity on the target FPGA, high performance function approximation techniques are considered.

## II. NEURAL INPAINTING PROCESSING

The neural raw signals applied in this paper were recorded invasively from the human brain of a male patient by $M$ electrodes of a surface MEA at the *Epilepsy Centre of Erlangen* (EZE) [12]. Here, local field potentials (LFP) are utilized, which are sampled at $f_{\text{S}} = 1024\,\text{Hz}$ by a resolution of $16\,\text{bits}$. Typically, $N$ time samples long data blocks in form of a matrix $\mathbf{A} \in \mathbb{R}^{M\times N}$ are utilized in the inpainting processing [7].

The major key of the recent inpainting-based data compression technique is the logical matrix $\mathbf{\Omega} \in \{0,1\}^{M\times N}$, which controls the recording behavior of the MEA [7]. Therefore, logical ones denote the recorded parts of the neural activity in time and spatial domain, respectively. All logical zeros in $\mathbf{\Omega}$ remark the constellation of deactivated electrodes, which regularized the data reduction. Hence, a compression ratio $\eta = 100\%(1 - |\mathbf{\Omega}|/(MN))$ can be defined, where the cardinality $|\cdot|$ of the mask $\mathbf{\Omega}$ corresponds to the number of retained components of the neural signal recording. The data transmission on implant side comprises only these reduced signal parts. In order to relocate the received signal fragments to the original time and spatial positions in the matrix $\mathbf{A}(\mathbf{\Omega}) \in \mathbb{R}^{M\times N}$, the mask has to be known for the data recovery procedure.

The structure-based inpainting algorithm [8] is an iterative method $1 \le t \le T_{\max}$, which composes the fundament of the *isophotes*-focused reconstruction of the correlated neural signals. Isophotes are lines of equal magnitude values, which serve as information in the recovery procedure and arrive at the boundary of the masked parts in $\mathbf{A}(\mathbf{\Omega})$. Optionally, in order to increase the spatial correlation between the neural signals a electrode-based rearrangement [7] can be applied as a calibration step and to choose a suitable mask $\mathbf{\Omega}$.

The inpainting recovery alternates between $B_{\max}$ anisotropic diffusion $\mathcal{D}$, which acts as a smoothing operator, and $A_{\max}$ recursive update computations,

$$\mathbf{A}^{a+1}(\mathbf{\Omega}(i,j)) = \mathbf{A}^{a}(\mathbf{\Omega}(i,j)) + \Delta u \cdot \mathbf{A}_u^a(\mathbf{\Omega}(i,j)), \quad (1)$$

in order to recover the missing parts of the signal array $\mathbf{A}(\mathbf{\Omega})$. In eq. (1) the exponent $a \in \{1,\ldots,A_{\max}\}$ indicates the iterative recovery computation, the parameter $\Delta u$ the update step size and the arguments $i$ and $j$ the vertical and horizontal coordinates in the zero-marked parts of $\mathbf{\Omega}$, respectively. By calculating the gradient $\mathcal{G}$ (in general, *central difference*: $\mathcal{G}_i = \mathbf{A}(i+1,j) - \mathbf{A}(i-1,j)$) and Laplacian $\mathcal{L}_i$ (standard kernel: $\mathcal{L} = \mathbf{A}(i+1,j) + \mathbf{A}(i-1,j) + \mathbf{A}(i,j+1) + \mathbf{A}(i,j-1) - 4\mathbf{A}(i,j)$) operators for the vector projection

$$\beta(i,j) = \frac{1}{\sqrt{\mathcal{G}_i^2 + \mathcal{G}_j^2}} \cdot \langle \begin{pmatrix} -\mathcal{G}_j \\ +\mathcal{G}_i \end{pmatrix}, \begin{pmatrix} \mathcal{L}_{i+1} - \mathcal{L}_{i-1} \\ \mathcal{L}_{j+1} - \mathcal{L}_{j-1} \end{pmatrix} \rangle \quad (2)$$

the update term $\mathbf{A}_u(\mathbf{\Omega}(i,j)) = \beta(i,j) \cdot |\nabla \mathbf{A}(i,j)|$ can detect and further recover isophotes in the masked neural array $\mathbf{A}(\mathbf{\Omega})$, while the so called slope limiter

$$|\nabla \mathbf{A}(i,j)| = \begin{cases} \sqrt{\mathcal{G}_{ibm}^2 + \mathcal{G}_{ifM}^2 + \mathcal{G}_{jbm}^2 + \mathcal{G}_{jfM}^2} & \beta > 0 \\ \sqrt{\mathcal{G}_{ibM}^2 + \mathcal{G}_{ifm}^2 + \mathcal{G}_{jbM}^2 + \mathcal{G}_{jfm}^2} & \beta < 0 \end{cases} \quad (3)$$

ensures the stability of the structure-based inpainting algorithm. In contrast to (2), the gradients in (3) are implemented as forward and backward differences labeled by the indices $b$ and $f$ for the coordinates $i$ and $j$, respectively. The letter indices $m$ and capital $M$ denote the minimum and maximum of the backward or forward differences to zero value.

After $A_{\max}$ recursive update computations $\mathbf{A}_u(\mathbf{\Omega})$, $B_{\max}$ anisotropic diffusions $\mathcal{D}$ are applied to the masked part of the neural array $\mathbf{A}(\mathbf{\Omega})$ in order to fill and smooth the result of the inpainting recovery. Therefore, appropriate smoothing operators could be exponential or fractional functions [13]:

$$\mathcal{D}_e(\mathbf{A}(\mathbf{\Omega}(i,j))) = \sum_{u,v} p_{u,v} \cdot e^{-(p_{u,v}/K)^2}, \quad (4)$$

$$\mathcal{D}_f(\mathbf{A}(\mathbf{\Omega}(i,j))) = \sum_{u,v} p_{u,v}/(1 + (p_{u,v}/K)^2), \quad (5)$$

where the partial differences are defined by the following expression: $p_{u,v} = \mathbf{A}(\mathbf{\Omega}(i+u, j+v)) - \mathbf{A}(\mathbf{\Omega}(i,j))$, here, all combinations of $\{u,v\} \in \{-1,0,+1\}$ have to be inserted, under the prerequisite that $u \neq v$ holds. Of course, such nontrivial functions like (2) and (3) as well as (4) or (5) are difficult to implement into hardware. Therefore, the logarithmic number system (LNS) approach is applied to this work in order to design a high performance hardware architecture for inpainting recovery computation.

## III. IMPLEMENTATION

### A. Number Format Transformation

As described in sec. II, nontrivial numeric functions are calculated in the LNS that decrease the signal processing effort evidently. Therefore, LOG and ALOG converter modules are required realizing the number format transformation. For an efficient realization of the elementary functions in (2), (3) and (4) as well as the arithmethic operation in (5), an automated piecewise linear function approximation method is proposed.

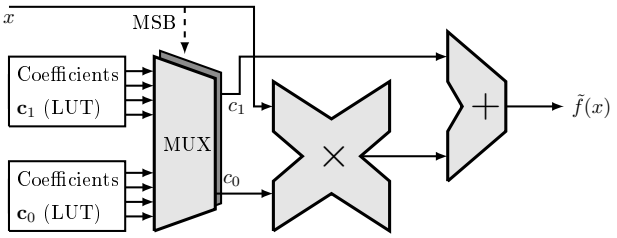



Fig. 2. Hardware structure for the conversion from FPNS to LNS.

This technique is about imitating the slope of a given original function by linear equations which are easy to compute by digital architectures [14]. Though, this will reduce the path delay and the latency, this technique also leads to small calculation errors. However, in the scope of (the comparatively low-precision) application-specific fixed-point computations, this method has proven to achieve good-enough results (e.g., [15]). By using a piecewise function approximation technique, the function will be split up into several pieces of sub-functions, so-called segments. Thus, finding an optimal trade-off between hardware performance and calculation error is one of the main goals of function approximation hardware design. In general, linear equations are described by $\tilde{f}(x) = c_1 \cdot x + c_0$, where $x$ denotes the input data, $\tilde{f}(x)$ the approximation of the original function $f(x)$ and $c_1$ and $c_0$ the gradient and offset, respectively. In order to realize a non-uniform segmentation scheme, the original function is split up into several sub-functions of variable input range each. Quick access to a sub-function is enabled by a restricted segmentation scheme. Each segment must fulfill the constraint

$$\mathrm{seg}(i) = \mathrm{seg}(i-1) + \frac{C_{\text{start}} - C_{\text{end}}}{2^{h_i}}, \tag{6}$$

with $C_{\text{start}}$, $C_{\text{end}}$ as start and end point of the function, seg as sub-function start point, $i$ as segment index ($\mathrm{seg}(0) = C_{\text{start}}$) and $h_i \in \mathbb{N}^+$ as the interval exponent of the $i$th segment. Note, that $h_i$ may differ for each segment which may cause a varying number of MSB that must be considered for each segment. The input range of the original function is set to $C_{\text{end}} - C_{\text{start}} = 2^{h_\mathrm{k}}$, with k as number of all segments. Hence, the sub-function selection can be carried out by taking only the most significant bits (MSB) of the input value $x$ into account. Thus, the approximation is expressible

$$\tilde{f}(\mathbf{x}) = (\boldsymbol{c}_1 \cdot \mathbf{x} + \boldsymbol{c}_0) \cdot \boldsymbol{\kappa}(x) \tag{7}$$

with $\boldsymbol{c}_1$ and $\boldsymbol{c}_0$ as the vector of gradients and offsets, respectively, and $\boldsymbol{\kappa}$ as the blanking function realizing the mapping of the linear equation to corresponding segments (see also [15]).

In order to leverage the segmentation and, consequently, minimize the calculation error, an automated accuracy-driven scheme is used. By the utilization of Remez-Algorithm, best-case approximations are achieved, minimizing the maximum error $\tilde{\epsilon}$ [14]. Hence, function approximation are generated (starting with the entire range of the original function) and $\tilde{\epsilon}$ is compared to a specified error constraint $\epsilon_{\max}$. If the constraint is not met ($\tilde{\epsilon} > \epsilon_{\max}$), bisection is performed and the heuristic starts over in the leftmost segment. Otherwise, if $\tilde{\epsilon} \leq \epsilon_{\max}$ is fulfilled, all coefficients and segment data are stored and the next segment to the right is taken into account. When the entire function has been processed this way, the stored parameters are mapped onto hardware using a corresponding VHDL-StringTemplate [16]. A more sophisticated description of this method (except for the multiplier-less design technique) can be found in [15]. An overview of the general hardware structure is given in fig. 2.

## IV. The logarithmic number format

In order to reduce the signal processing effort, e.g., of the smoothing functions given in (4) and (5), LNS-based calculation is taken into account. In general, this technique is used to simplify numeric operations and/or elementary functions by processing the exponential equivalents of the operands [11]. As the trivial arithmetic addition and subtraction cannot be performed in the LNS, number format transformation are utilized in this paper to switch between the LNS and the (ordinary) fixed-point number system (FPNS). According to [17], a value $V_{\text{LNS}}$ in the LNS can be expressed as

$$V_{\text{LNS}} = (1-z) \cdot (-1)^s \cdot 2^{-e} \cdot 2^l \ , \tag{8}$$

with $s$, $z$ as the sign and zero bit, $e$ as the exponent and $l$ as the logarithmic mantissa. Note, that the LNS is very similar to the floating-point number system except for the logarithmic expression of the mantissa. In contrast to this, the $V_{\text{FPNS}}$ of the FPNS is usually defined as

$$V_{\text{FPNS}} = (-1)^s \cdot o \cdot 2^{-r} \ , \tag{9}$$

where $s$ is the sign bit, $o$ the value offset and $r$ the radix point. The logarithmic (LOG) and anti-logarithmic (ALOG) mantissa transformation are expressed by

$$L = \log_2(O_n + 1);\ O_n = O \cdot 2^e;\ 0 \leq O_n < 1 \ , \tag{10}$$

$$O = 1 + (2^{L_n} - 1);\ L_n = L \cdot e;\ 0 \leq L_n < 1 \ , \tag{11}$$

with $O_n$ and $L_n$ as the normalized values shifted by $2^e$. Thus, the transformation requires the calculation of (non-linear) elementary functions which is realized in this paper by the use of piece-wise linear function approximation (sec. III-A), as this approach has proven to be a powerful solution in many application areas, e.g., mobile communication [15].

### A. LNS-based signal processing optimizations

In oder to simplify arithmetic operations and elementary functions, the LNS is established to realize a high performance hardware architecture. For the hardware implementation perspective, the LNS approach replaces multi-digit multiplication like $uv$ by table look-ups and simpler addition because of the fact that the logarithm of a product is the sum of the logarithms of the factors $\log_a(uv) = \log_a(u) + \log_a(v)$.

Fig. 2 presents the used LNS converter circuit in order to determine nontrivial mathematical expressions. This converter consists of two LUTs for the coefficient sets $\{\mathbf{c}_0, \mathbf{c}_1\} \in \mathbb{R}^k$, a multiplexer (MUX) controlled by the MSB of the input $x$ and a single multiplier as well as adder to obtain the

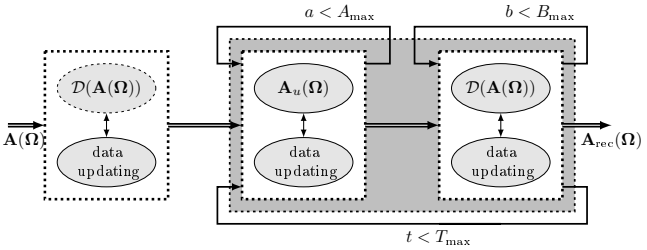



Fig. 3. Schematical FSM of one path of the parallel inpainting-based recovery computation including a global data updating in each iteration.

corresponding LNS output. As a result, the nontrivial functions of the inpainting recovery algorithm can be implemented into HDL as follows

$$u_1\sqrt{\frac{u_2}{u_3}} \xrightarrow{\log_2} \log_2(u_1) + \frac{\log_2(u_2)}{2} - \frac{\log_2(u_3)}{2} \tag{12}$$

for the update term $\beta(i,j) \cdot |\nabla \mathbf{A}(i,j)|$ in the eq. (2) and (3) as well as the two different variants of the ansitropic diffusion $\mathcal{D}$ operations in eq. (4) and (5)

$$\mathcal{D}_e: \; u_1 e^{-u_2^2} \xrightarrow{\log_2} \log_2(u_1) - u_2^2 \log_2(e), \tag{13}$$

$$\mathcal{D}_f: \; \frac{u_1}{1+u_2^2} \xrightarrow{\log_2} \log_2(u_1) - \log_2(1+u_2^2), \tag{14}$$

whereby only one diffusion is necessary [8], [13]. Exploiting the LNS, the nontrivial functions in the hardware architecture exhibit only simple structures as MUX, multiplier, adders and LUTs for constant factors.

### B. Hardware architecture

The hardware architecture for the inpainting-based recovery is realized by a FSM using normalized neural input data $\mathbf{A} \in [0,1]^{M \times N}$ in FPNS format. The FSM is separable into three superordinate parts as shown in fig. 3. The depicted data path is one schematically branch of a parallel implementation with global data updating unit, which allocate the computed data to the next calculation cycle (just wiring). A single reconstruction branch requires 13 input sample inside and around the mask $\mathbf{\Omega}$ in order to determine e.g. the gradient $\mathcal{G}$ and Laplacian $\mathcal{L}$.

After an initial anisotropic diffusion $\mathcal{D}$ and data updating, the detection and completion of the isophotes in $\mathbf{A}(\mathbf{\Omega})$ are computed by the proposed LNS-based realization of the normalized gradient (2) and the slope limiter (3). For $\mathcal{D}$, the argument $p_{u,v}/K$ is calculated in LNS, before the exponential function in eq. (4) is computed in FPNS as a multiplication of the negative argument and $\log_2(e)$, while the fractional function in eq. (5) needs to be realized in LNS by inverting the sign after adding a one in FPNS. All counter are used to ensure that the described alternating procedure of the inpainting-based recovery is executed $T_{\max}(A_{\max} + B_{\max})$ times.

## V. Results

The hardware architecture evaluation of the here proposed inpainting-based compression and reconstruction approach for correlated neural signals are shown in the following. All simulation results include neural raw data. Each array $\mathbf{A} \in \mathbb{R}^{M \times N}$ used in this work consists $M = 48$ different electrode and $N = 128$ time samples. Comparable to fig. 1, a logical mask $\mathbf{\Omega} \in \{0,1\}^{M \times N}$ in form of a regular grid of several $6 \times 6$ patches is applied. In order to achieve a compression ratio of $\eta \approx 62\%$ the mask comprises a vertical and horizontal spacing of only 3 samples between the patches. The reconstruction results presented in this work are based on a large amount of neural signals and the inpainting parameters shown in tab. II.

TABLE II
Used evaluation parameters to recover neural signals.

| $\eta$ in % | $T_{\max}$ | $A_{\max}$ | $B_{\max}$ | $\Delta u$ | $K$ |
|---|---|---|---|---|---|
| 62 | 50 | 15 | 5 | 0.1 | 0.25 |

In order to assess the capability of the inpainting algorithm, the *normalized mean squared error* (NMSE)

$$\text{NMSE} = \frac{\|\mathbf{A}(\mathbf{\Omega}) - \hat{\mathbf{A}}(\mathbf{\Omega})\|_{\text{F}}}{\|\mathbf{A}(\mathbf{\Omega})\|_{\text{F}}} \tag{15}$$

is computed, where $\mathbf{A}(\mathbf{\Omega})$ stands for the original array and $\hat{\mathbf{A}}(\mathbf{\Omega})$ marks the recovery result. The letter F in expression $\|\cdot\|_{\text{F}}$ in eq. (15) denotes the *Frobenius* norm.

Several function approximations of the non-linear functions in the LOG and ALOG converter modules with a varying error constraint are generated (see tab. III). Note, that the size of the datapath is always two higher than the specified error (e.g. for $\epsilon_{\max} = 2^{-12}$, an datapath width of 14 is considered).

### A. Algorithmic performance

Fig. 4 shows the mean NMSE and convergence behavior of the inpainting-based reconstruction depending on the iteration $t$, which includes the alternating procedures of computation of the update $\mathbf{A}_u(\mathbf{\Omega})$ and the anisotropic diffusion $\mathcal{D}$. The comparison of the different implementations of the anisotropic diffusion in MATLAB, both double and fixed-point format is illustrated in fig. 4(a) and 4(b), respectively. As described above, the exponential $\mathcal{D}_e$ and fractional function $\mathcal{D}_f$ are analyzed regarding error and convergence for the inpainting smoothing operation. While the double format realization requires a large amount of iterations $t$ in order to adequately reconstruct the correlated neural signals, the fixed-point implementation achieves a suitable and stable error value after only a few iterations $t$. This gainful phenomena can be explained by quantization effects of the conversions into the LNS and back to FPNS, which acts as a inherent thresholding operation. Therefore, this implementation forces values less and more important greater than a defined range to zero. This demonstrates that the proposed inpainting algorithm still has room with respect to the reconstruction procedure in terms of quality and speed. In this evaluation, the anisotropic diffusion performed by the fractional function $\mathcal{D}_f$ reaches both a faster convergence and a higher remaining NMSE during the inpainting recovery.

Fig. 4(c) illustrates the reconstruction quality enhancement depending on the selected quantization level. The increment of the word size within the proposed hardware architecture based on LNS leads to an improvement in terms of NMSE. Depending on the application and type of medical diagnostics,

TABLE III
OVERVIEW OF THE INVESTIGATED FUNCTION APPROXIMATIONS FOR THE LOG AND ALOG HARDWARE MODULES WITH VARYING $\epsilon_{\max}$.

| $\epsilon_{\max}$ | $2^{-12}$ | $2^{-14}$ | $2^{-16}$ | $2^{-18}$ | $2^{-20}$ |
|---|---|---|---|---|---|
| $k_{\text{LOG}}$ | 23 | 43 | 86 | 174 | 341 |
| $k_{\text{ALOG}}$ | 25 | 51 | 102 | 205 | 415 |

TABLE IV
ESTIMATED SYNTHESIS RESULTS REGARDING VIRTEX-5 XC5VFX200T.

| Module | Used | Available | Utilization |
|---|---|---|---|
| Number of Slices Regs | 10094 | 122880 | 8 % |
| Number of Slice LUTs | 23440 | 122880 | 19 % |
| Number of LUT-FF pairs | 5652 | 27882 | 20 % |
| Number of DSPs | 171 | 384 | 44 % |

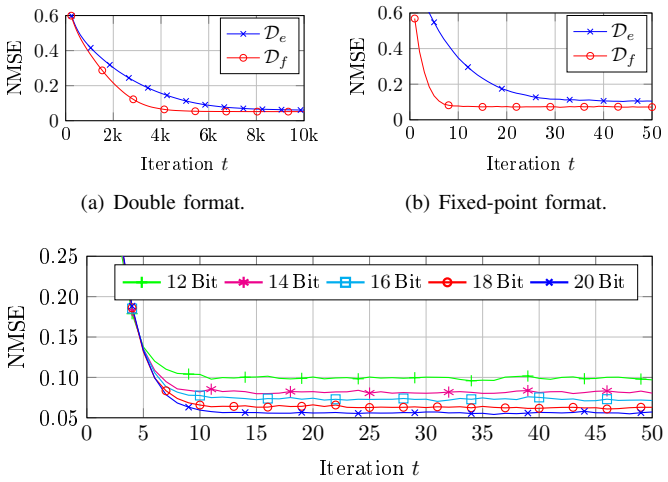
a resolution of equal or greater than 16 bits is recommended in order to design an efficient architecture with regard to accuracy, speed and hardware issues (area, energy, etc.). This resolution correspondences to a NMSE less than 0.8, which is consistent with the results of tab. I.

### B. FPGA-specific evaluation

The synthesis results given in this paper were applied to a Xilinx Virtex-5 xc5vfx200t FPGA, shown in tab. IV. The non-optimized hardware implementation reaches a target clock frequency of 79.107 MHz. The proposed architecture achieves a throughput of 32 961 parallel inpainting recoveries ($3 \times 3$ patches in $\mathbf{\Omega}$ and $T_{\max} = 15$) per second based on 16 Bits. With this advantage, the large amount of recorded neural data can be analyzed by medical scientists in time.

## VI. CONCLUSION

The first hardware reconstruction architecture for neurological multichannel signals compressed by inpainting-based mask is proposed. A high throughput implementation of the nontrivial computations required for the recovery is achieved by utilizing the logarithmic number system with non-uniform piecewise linear approximation. This increases the recovered signal quality and computation speed with only negligible error. Compared to a MATLAB double implementation the proposed architecture generates results with at least similar accuracy while the iterative-based computations are dramatically reduced because of an inherent zero forcing by LNS conversion and error correction behavior of the structure-based inpainting algorithm.




(a) Double format. (b) Fixed-point format.

(c) Fixed-point format. Quantization effect regarding inpainting by $\mathcal{D}_f$.

Fig. 4. Mean error and convergence behavior of inpainting recovery for correlated neural signals as a function of the iteration $t$. Comparison of exponential $\mathcal{D}_e$ and fractional anisotropic diffusion $\mathcal{D}_f$ using MATLAB double (a) and fixed-point format (b), respectively. A significant reduction of required iteration $t$ for similar recovery quality can be observed. Furthermore, in (c) an enhanced signal quality can be obtained by increasing the resolution.

## REFERENCES


[1] R. Meier, H. Dittrich, A. Schulze-Bonhage, and A. Aertsen, "Detecting epileptic seizures in long-term human eeg: A new approach to automatic online and real-time detection and classification of polymorphic seizure patterns," *Journal of Clinical Neurophysiology*, vol. 25, no. 3, 2008.

[2] P. Shenoy, K. Miller, J. Ojemann, and R. Rao, "Generalized features for electrocorticographic bcis," *Biomedical Engineering, IEEE Transactions on*, vol. 55, Jan 2008.

[3] J. Pistor, J. Hoeffmann, D. Rotermund, and E. T. et al., "Development of a fully implantable recording system for ECoG signals," in *Design, Automation & Test in Europe Conference & Exhibition (DATE)*, 2013.

[4] K. Wise, D. Anderson, J. Hetke, D. Kipke, and K. Najafi, "Wireless implantable microsystems: high-density electronic interfaces to the nervous system," *Proceedings of the IEEE*, jan 2004.

[5] G. K. Wallace, "The JPEG still picture compression standard," *Commun. ACM*, vol. 34, no. 4, pp. 30–44, apr 1991.

[6] E. Candes, J. Romberg, and T. Tao, "Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information," *Information Theory, IEEE Transactions on*, 2006.

[7] S. Schmale, B. Knoop, D. Peters-Drolshagen, and S. Paul, "Structure reconstruction of correlated neural signals based on inpainting for brain monitoring," in *Biomedical Circuits and Systems Conference (BioCAS), 2015 IEEE*, Oct 2015, pp. 1–4.

[8] M. Bertalmio, G. Sapiro, V. Caselles, and C. Ballester, "Image inpainting," in *Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques*, ser. SIGGRAPH '00, 2000.

[9] S. Schmale, H. Lange, B. Knoop, D. Peters-Drolshagen, and S. Paul, "Compression and reconstruction methodology for neural signals based on patch ordering inpainting for brain monitoring," *41st IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2016)*, mar 2016, (accepted).

[10] S. Schmale, J. Hoeffmann, B. Knoop, G. Kreiselmeyer, H. Hamer, D. Peters-Drolshagen, and S. Paul, "Exploiting correlation in neural signals for data compression," *22nd European Signal Processing Conference (EUSIPCO 2014)*, sep 2014.

[11] J. N. Mitchell, "Computer Multiplication and Division Using Binary Logarithms," *IRE Transactions on Electronic Computers*, Aug. 1962.

[12] Epilepsy Centre Erlangen (EZE), Germany, "Data of the Epilepsy Centre," 2013, http://www.epilepsiezentrum.uk-erlangen.de/.

[13] P. Perona and J. Malik, "Scale-space and edge detection using anisotropic diffusion," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, Jul 1990.

[14] J.-M. Muller, *Elementary functions : Algorithms and Implementation*, 2nd ed. Birkhaeuser Boston, 2006.

[15] J. Rust, F. Ludwig, and S. Paul, "Low Complexity QR-Decomposition Architecture Using the Logarithmic Number System," in *Design, Automation and Test in Europe Conference and Exhibition (DATE)*, 2013.

[16] T. Parr, "Enforcing Strict Model-view Separation in Template Engines," in *Proceedings of the 13th International Conference on World Wide Web*, New York, NY, USA, 2004.

[17] J. Detrey and F. de Dinechin, "A VHDL library of LNS operators," in *Conference Record of the Thirty-Seventh Asilomar Conference on Signals, Systems and Computers*, nov. 2003.