Abstract—Telehealth and wearable equipment can deliver personal healthcare and necessary treatment remotely. One major challenge is transmitting large amount of biosignals through wireless networks. The limited battery life calls for low-power data compressors. Compressive Sensing (CS) has proved to be a low-power compressor. In this study, we apply CS on the compression of multichannel biosignals. We firstly develop an efficient CS algorithm from the Block Sparse Bayesian Learning (BSBL) framework. It is based on a combination of the block sparse model and multiple measurement vector model. Experiments on real-life Fetal ECGs showed that the proposed algorithm has high fidelity and efficiency. Implemented in hardware, the proposed algorithm was compared to a Discrete Wavelet Transform (DWT) based algorithm, verifying the proposed one has low power consumption and occupies less computational resources.

Keywords—Compressive Sensing (CS), Block Sparse Bayesian Learning, ECG, Wireless Telemonitoring

I. INTRODUCTION

Personal healthcare benefits from the development of wireless telemonitoring. Using on-body sensors of these telemonitoring systems, biosignals reflecting health status can be easily collected and transmitted, and diagnosis is thus performed remotely at a data central. This client–central model calls for low-power telemonitoring techniques [1]. One major challenge is huge amount of data collected versus limited battery life of portable devices and limited bandwidth of wireless networks. Signals need to be compressed before transmitting. However, most traditional compression methods consist of sophisticated matrix-vector multiplication and encoding which subsequently drains the battery.

Compressive sensing (CS) can be used as an efficient, lossy compression method. In this framework, the complexity is shifting from battery driven devices to data centrals, where a CS algorithm is used to recover original recordings. In practice, telemonitoring systems allow users to move freely or engage in activities. Artifacts caused by body movements deteriorate the quality of signals [2], [3] and may result in varied sparsity. These pose challenges for CS algorithms to fidelity recover the biosignals. Recently, Zhang et al. [4] proposed the Block Sparse Bayesian Learning (BSBL) framework, which showed excellent performance when recovering less-sparse signals with high fidelity. However, to recovery multichannel signals, existing BSBL algorithms [4], [5] have to recover them channel-by-channel.

Recently a spatio-temporal sparse Bayesian learning model was proposed for compressive sensing of multichannel biosignals [2], which is a combination of the block sparse model [4] and the multiple measurement vector model [6]. In this work we extend our fast BSBL algorithm [5] for this model, such that it is suitable for compressed sensing of multi-channel biosignals with fast speed. It has high fidelity and efficiency in recovering real-life fetal ECGs. We also provide the evaluation of CS-based and DWT-based compressors in the platform of Field Programmable Gate Array (FPGA). Results show that the CS-based framework has fewer compression latency, consumes less resource and dynamic power.

Throughout the paper, bold letters are reserved for vectors and matrices. The concatenation of scalars and vectors are denoted by \( y = \{y_1, \ldots, y_S\} \) and \( X = \{x_1, \ldots, x_N\} \). \( \otimes \) denotes the Kronecker product and \( \text{Tr}(A) \) denotes the trace of \( A \). \( \text{vec}(A) \) represents the vectorization of \( A \). \( I_d \) denotes an identity matrix with size \( d \).

II. THE PROPOSED ALGORITHM

The Multiple-Measurement Vector (MMV) model [6] can be applied to multi-channel physiological signals \( X \), where \( X \triangleq \{x^1, \ldots, x^P\} \) and \( x^i \in \mathbb{R}^{N \times 1} \) is a column vector represents the data samples of the \( i \)-th channel. \( X \in \mathbb{R}^{N \times P} \) is also called a packet. The compressed measurements \( Y \) is obtained as

\[
Y = \Phi X,
\]

where \( \Phi \) is the sensing matrix. To minimize the power consumption, Bernoulli sensing matrix \( \Phi \in \{0,1\}^{M \times N} \) whose entries consist of 0s and 1s is preferred [5], [7].

As in [2], the block sparse structure [4] is incorporated into the above MMV model. In Fig 1 the \( i \)-th (where \( i \in \{1, \ldots, g\} \)) block is denoted by \( X_i \), which has the size \( d_i \times P \). A packet \( X \) is therefore divided into \( g \) blocks. We then
leads to the following cost function,

\[ C = \beta^{-1}I + \sum_{m \neq i} \Phi_m \gamma_i \Phi_i^T + \Phi_i \gamma_i \Phi_i^T \]

(11)

where \( C_{-i} \equiv \beta^{-1}I + \sum_{m \neq i} \Phi_m \gamma_m \Phi_m^T \). Using the Woodbury Identity, (10) can be rewritten as:

\[
\begin{align*}
\mathcal{L} &= N \log |C_{-i}| + \text{Tr} \left[ Y^T C_{-i}^{-1} Y \right] \\
&\quad + N \log |I_d_i + \gamma_i s_i| - \text{Tr} \left[ q_i^T (\gamma_i^{-1} I_d_i + s_i)^{-1} q_i \right] \\
&= \mathcal{L}(-i) + \mathcal{L}(i)
\end{align*}
\]

where \( s_i \equiv \Phi_i^T C_{-i}^{-1} \Phi_i, q_i \equiv \Phi_i^T C_{-i}^{-1} Y \) and

\[
\gamma_i \text{ can be efficiently updated by optimizing over } \mathcal{L}(i),
\]

\[
\gamma_i = \frac{1}{d_i} \text{Tr} \left[ s_i^{-1} (q_i q_i^T - s_i) s_i^{-1} \right].
\]

(13)

We optimize the cost function (10) with Fast Marginalized Likelihood Maximization (FMLM) method [5]. Let \( \Phi_i \in \mathbb{R}^{M \times d_i} \) be the \( i \)th column block in \( \Phi \), \( C \) in (3) can be rewritten as:

\[
C = \beta^{-1}I + \sum_{m \neq i} \Phi_m \gamma_m \Phi_m^T
\]

(12)

The measurements \( Y \) can be assumed to be i.i.d Gaussian noise with the precision parameter given by \( \beta \).

Given (4) and (5), we derive the posterior \( p(X|Y; \gamma_i, \beta) \) as follows:

\[
p(X|Y; \gamma_i, \beta) = \mathcal{M}N(X; \mu, \Sigma, I_P)
\]

(6)

where

\[
\mu = \beta \Sigma \Phi^T Y
\]

(7)

\[
\Sigma = (\Gamma^{-1} + \beta \Phi^T \Phi)^{-1}.
\]

(8)

Similar as in [5], the Type II Maximum Likelihood method is used to estimate the parameters \( \{\gamma_i\}, \beta \), which leads to the following cost function,

\[
\mathcal{L}(\gamma_i, \beta) \equiv -2 \log p(Y; \gamma_i, \beta) = N \log |C| - \text{Tr} \left[ Y^T C^{-1} Y \right]
\]

(10)

where \( C = \beta^{-1}I_M + \Phi \Gamma \Phi^T \).

We optimize the cost function (10) with Fast Marginalized Likelihood Maximization (FMLM) method [5]. Let \( \Phi_i \in \mathbb{R}^{M \times d_i} \) be the \( i \)th column block in \( \Phi \), \( C \) in (3) can be rewritten as:

\[
C = \beta^{-1}I + \sum_{m \neq i} \Phi_m \gamma_m \Phi_m^T
\]

(11)

where \( C_{-i} \equiv \beta^{-1}I + \sum_{m \neq i} \Phi_m \gamma_m \Phi_m^T \). Using the Woodbury Identity, (10) can be rewritten as:

\[
\begin{align*}
\mathcal{L} &= N \log |C_{-i}| + \text{Tr} \left[ Y^T C_{-i}^{-1} Y \right] \\
&\quad + N \log |I_d_i + \gamma_i s_i| - \text{Tr} \left[ q_i^T (\gamma_i^{-1} I_d_i + s_i)^{-1} q_i \right] \\
&= \mathcal{L}(-i) + \mathcal{L}(i)
\end{align*}
\]

where \( s_i \equiv \Phi_i^T C_{-i}^{-1} \Phi_i, q_i \equiv \Phi_i^T C_{-i}^{-1} Y \) and

\[
\gamma_i \text{ can be efficiently updated by optimizing over } \mathcal{L}(i),
\]

\[
\gamma_i = \frac{1}{d_i} \text{Tr} \left[ s_i^{-1} (q_i q_i^T - s_i) s_i^{-1} \right].
\]

(13)

The proposed algorithm (denoted as MBSBL-FM) is given in Fig. 2. Within each iteration, it only updates

1: procedure MBSBL-FM(Y, \Phi, \eta)
2: Outputs: \( X, \Sigma, \gamma \)
3: Initialize \( \beta^{-1} = 0.01 \|Y\|_F^2 \)
4: Calculate \( \{s_i\}, \{q_i\} \)
5: while not converged do
6: Calculate \( \gamma_i = \frac{1}{d_i} \text{Tr} \left[ s_i^{-1} (q_i q_i^T - s_i) s_i^{-1} \right], \forall i \)
7: Calculate \( \Delta \mathcal{L}(i) = \mathcal{L}(\gamma_i) - \mathcal{L}(\gamma_i), \forall i \)
8: Select the \( i \)th block s.t. \( \Delta \mathcal{L}(i) = \min \{ \Delta \mathcal{L}(i) \} \)
9: Re-calculate \( \mu, \Sigma, \{s_i\}, \{q_i\} \)
10: Re-calculate the convergence criterion
11: end while
12: end procedure

Figure 2. The MBSBL-FM Algorithm.
the time domain [2, 3], therefore people often resort to a transformed domain in firstly seeking the sparse coefficients $A$, which can be expressed as

$$Y = (\Phi D)A,$$

(14)

where $D$ may be a Discrete Cosine Transform (DCT) Matrix and $A \triangleq \{\alpha_1, \ldots, \alpha_P\}$ were the transformed coefficients, i.e., $x_t = D \alpha_t$. Then we obtain the reconstructed signal $X$ by $X = DA$.

III. TELEMONITORING OF MULTI-CHANNEL FETAL ECGS VIA COMPRESSIVE SENSING

We conducted experiments using real-life Fetal ECGs [7]. The recordings contained abdominal 8-channel signals sampled at 256Hz. This dataset is hard for most existing CS algorithms to decompress. Only BSBL algorithms [5], [7] showed high fidelity. In this experiment, we compared 11 state-of-the-art CS solvers as shown in Table I. The recovery was performed in the DCT transformed domain using (14).

Table I

| CS Algorithm | Objective Function | Types* |
|--------------|--------------------|--------|
| BP [8]       | $\ell_1$ minimization |        |
| SL0 [9]      | smooth $\ell_0$ minimization |        |
| Group BP [8] | group $\ell_1$ minimization | 1      |
| BSBL-EM [4]  | Block Sparse Bayesian Learning | 1\(\alpha\) |
| BSBL-BO [4]  | Block Sparse Bayesian Learning | 1\(\beta\) |
| BSBL-FM [5]  | Block Sparse Bayesian Learning | 1      |
| SPG-MMV [8]  | MMV $\ell_1$ minimization | 2      |
| ISL0 [10]    | MMV smooth $\ell_0$ minimization | 2\(\alpha\) |
| T-MSBL [8]   | MMV Sparse Bayesian Learning | 2\(\beta\) |
| MBSBL-FM     | MMV Block Sparse Bayesian Learning | 1\(\gamma\) |

*1: Block Sparse Model. 
*2: MMV Model. 
*3: Intra-block Correlation information. 
*4: Inter-Vector Correlation information.

The dataset was divided into packets. Each packet $X \in \mathbb{R}^{N \times P}$ was constructed with 1s ECG recordings where $N = 256$ and $P = 8$. We ran the experiment for 10 trials. In each trial, a new Bernoulli sensing matrix $\Phi \in \{0,1\}^{M \times N}$ with exactly two 1s of each column was generated. The compression ratio $CR = (N - M)/N$ was varied from 0.4 to 0.8. With each parameter setting, the Normalized Mean Square Error (NMSE $\triangleq \|\hat{X} - X\|^2_2/\|X\|^2_2$) and CPU time were recorded. The computer used in the experiments had a 2.9GHz CPU and 16GB RAM.

From Fig 3, we clearly see that only BSBL algorithms showed the best performance with CR ranged from 0.4 to 0.7. The proposed algorithm, MBSBL-FM, was almost 24 times faster than BSBL-BO and 17 times faster than BSBL-EM, while still yielded similar NMSE value. It achieved a good balance between speed and performance.

IV. HARDWARE EVALUATION

In this section the energy consumption of DWT-based and CS-based compressors was evaluated on Field-Programmable Gate Array (FPGA). Unlike DSP or embedded platforms [11], FPGA favors parallel implementation and fix-point arithmetic. It implements only the logics related to the compressor while the rest are holding reset. Therefore, it is more suitable for low-energy applications.

The FPGA uses Registers (or Flip-Flops, denoted as FF) to store the bit-information and Look-Up-Tables (denoted as LUT) to implement the combinational logics. Data are stored in on-chip Memories (denoted as RAM). The dynamic power consumption (denoted as $P_d$) of a compressor reflects the design activity and switching events in the chip. In order to achieve low power, the chip should minimize circuitry activities and avoid using multipliers (denoted by MUL).

A. Implementations

The DWT-based compressor implemented in [5] was used for comparison. It adopted lift-based filtering and used the multiplierless LeGall 5/3 wavelet filter. The transformation stages of the DWT was set to 4.

For each single-channel biosignal, the CS-based compressor consists only one matrix-vector multiplication, i.e., $y = \Phi x$. In FPGA, the compression can be fully-parallelized with the on-the-fly scheme: Let $\phi_i$ denotes the $i$th column of the sensing matrix $\Phi$, $y(k)$ denotes the compressed data vector after have received $k$ samples $\{x_1, \ldots, x_k\}$. Starts
with \( y^{(0)} = 0 \), the compressed data vector is iteratively updated with each new sample \( x_i \),

\[
y^{(i)} = \Phi_i x_i + y^{(i-1)},
\]

(15)

The compression is done once the last sample \( x_N \) of a packet has been acquired. When \( \Phi \) is the Gaussian sensing matrix, each \( \phi_i \) is real valued. (15) must be calculated in \( M \) clock cycles with one multiplier unit. However, when \( \Phi \) is the Bernoulli matrix with two 1s of each column, (15) can be implemented in one clock-cycle without multiplier. This is the reason why using the Bernoulli matrix can reduce energy consumption and save other hardware resources.

Note that the above procedure is performed on all biosignals from different channels in a parallel way.

B. Evaluation

Table II shows the power consumption and other used hardware resources by the DWT-based compressor and the CS-based compressor when compressing a single-channel biosignal\(^1\). The CS-based compressor adopted two kinds of sensing matrices. One was the random Gaussian sensing matrix (denoted by CS-Gaussian). The second was the Bernoulli sensing matrix as described above (denoted by CS-Bernoulli). In addition, we calculated the compression latency (denoted by \( t_L \)) which is the number of the clock cycles used for compressing after a packet has been acquired.

The CS-Gaussian compressor required one multiplier and multiple clock-cycles to generate the compressed measurements. It also needed to store the whole quantized sensing matrix. Thus this implementation consumed more RAMs and energy. On the other hand, both the DWT-based compressor and the CS-Bernoulli compressor were multiplierless. The CS-Bernoulli had minimal latency, consumed only \( \sim 1/3 \) Registers and LUTs than the DWT-based one. In order to reduce the RAM usage, it stored the locations of 1s in the sensing matrix. Totally, the CS-Bernoulli compressor saved 31.25% dynamic power against the DWT implementation.

For clarity, we only present the statistics on a single-channel biosignal. For multichannel biosignals, the advantages are more significant.

V. Conclusion

In this paper, we proposed a fast compressive sensing algorithm that is based on a hybrid model of the block sparse model and the MMV model. It is suitable for compressive sensing of multichannel biosignals. Experiments on real-life Fetal ECGs showed that the proposed algorithm has both high fidelity and efficiency. We also compared the consumed hardware resources by the proposed algorithm and a DWT-based compression algorithm during data compression. The results showed that the proposed algorithm (adopting the Bernoulli sensing matrix) required shorter compression latency, less computational resources, and less power consumption.

REFERENCES

[1] M. Duarte, G. Shen, A. Ortega, R. Baraniuk, M. Duarte, G. Shen, A. Ortega, and R. Baraniuk, “Signal compression in wireless sensor networks,” Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences, vol. 370, no. 1958, pp. 118–135, 2012.
[2] Z. Zhang, B. D. Rao, and T.-P. Jung, “Compressed sensing for energy-efficient wireless telemonitoring: Challenges and opportunities,” in Asilomar Conference on Signals, Systems and Computers (Asilomar 2013), 2013.
[3] T. Martin, E. Jovanov, and D. Raskovic, “Issues in wearable computing for medical monitoring applications: a case study of a wearable ECG monitoring device,” in The Fourth International Symposium on Wearable Computers. IEEE, 2000, pp. 43–49.
[4] Z. Zhang and B. Rao, “Extension of SBL algorithms for the recovery of block sparse signals with intra-block correlation,” Signal Processing, IEEE Transactions on, vol. 61, no. 8, pp. 2009–2015, 2013.
[5] B. Liu, Z. Zhang, G. Xu, H. Fan, and Q. Fu, “Energy efficient telemonitoring of physiological signals via compressive sensing: A fast algorithm and power consumption evaluation,” arXiv preprint arXiv:1309.7843, 2013.
[6] Z. Zhang and B. D. Rao, “Sparse signal recovery with temporally correlated source vectors using sparse bayesian learning,” IEEE Journal of Selected Topics in Signal Processing, vol. 5, no. 5, pp. 912–926, 2011.
[7] Z. Zhang, T.-P. Jung, S. Makeig, and B. Rao, “Compressed sensing for energy-efficient wireless telemonitoring of noninvasive fetal ECG via block sparse bayesian learning,” Biomedical Engineering, IEEE Transactions on, vol. 60, no. 2, pp. 300–309, 2013.
[8] E. V. D. Berg and M. P. Friedlander, “Probing the pareto frontier for basis pursuit solutions,” SIAM Journal on Scientific Computing, vol. 31(2), pp. 890–912, 2008.
[9] G. H. Mohimani, M. Babaie-Zadeh, and C. Jutten, “A fast approach for overcomplete sparse decomposition based on smoothed \( l^0 \) norm,” IEEE Transactions on Signal Processing, vol. 57(1), pp. 1–13, 2008.
[10] M. M. Hyder and K. Mahata, “Direction-of-arrival estimation using a mixed \( l_0 , l_2 \) norm approximation,” IEEE Transactions on Signal Processing, vol. 58, pp. 4646–4655, 2010.
[11] H. Mamaghanian, N. Khaled, D. Atienza, and P. Vanderheyden, “Compressed sensing for real-time energy-efficient ECG compression on wireless body sensor nodes,” Biomedical Engineering, IEEE Transactions on, vol. 58, no. 9, pp. 2456–2466, 2011.