

DSP Based OFDM Receiver for Time-varying Underwater Acoustic Channels

Bo Peng, Pierluigi Salvo Rossi, Hefeng Dong, and Kimmo Kansanen

Department of Electronics and Telecommunications

Norwegian University of Science and Technology,

Email: bo.peng@iet.ntnu.no, salvorossi@iet.ntnu.no, hefeng.dong@iet.ntnu.no, kimmo.kansanen@iet.ntnu.no

Abstract—This paper presents a digital signal processor (DSP) based iterative orthogonal frequency division multiplexing (OFDM) receiver for time-varying underwater acoustic channels. The time-varying channel is modelled by basis expansion model (BEM). To explore the inherent sparsity of underwater acoustic channel, orthogonal matching pursuit (OMP) algorithm is adopted. A fast Fourier transform (FFT) based low-complexity implementation of OMP algorithm is employed to reduce computational complexity and save memory space. The receiving system is implemented on a multi-core DSP evaluation module TMDX-EVM6678L. The performance of proposed receiving system is validated through experimental data collected from real world. With two cores running at 1 GHz, the real-time processing is achieved and the system performance is satisfying.

Index Terms—Underwater acoustic communication, DSP, Iterative receiver, orthogonal frequency division multiplexing (OFDM)

I. INTRODUCTION

Underwater acoustic network has wide applications in oceanographic data collection, underwater structure monitoring and scientific exploration. Designing reliable and high data rate physical communication is key for the success of underwater acoustic network. Underwater acoustic channel which often exhibits doubly-selective (both time selective and frequency selective) fading provides significant challenge for high performance communication in the physical layer. Various techniques have been proposed to combat the challenge caused by the channel. For single carrier system, the advanced adaptive turbo equalization technique has been adopted to deal with the severe time-varying inter symbol interference (ISI) generated by long channel delay spread and Doppler spread [1]. Orthogonal frequency division multiplexing (OFDM), which enables to use low complexity equalizer in ISI limited channel, has also been studied for underwater acoustic channel [2], [3]. To combat the inter carrier interference (ICI) caused by Doppler effect, receivers with ICI equalizer are investigated in [4]–[7].

Real-time implementation of underwater communication system has also been studied in [8]–[12]. For single carrier system, the adaptive multi-channel equalizer is applied in [11]. Direct spread spectrum technique is used in [10] and turbo equalizer is incorporated in [8], [12]. Underwater modem with OFDM technique is proposed in [9]. However, ICI is ignored in [9], which can affect its performance in time-varying environment.

In this work, we present our DSP implementation of an iterative OFDM receiver. Unlike work in [9], ICI is considered in our implementation. To the best of our knowledge, it is the first real-time implementation of iterative OFDM underwater acoustic communication receiver that takes ICI into account. The basis expansion model (BEM) [13]–[16] is used to model the time-varying channel. To explore the sparsity inherited in sparse channel, orthogonal matching pursuit (OMP) based channel estimation [4], [17] method is applied in this work. A low-complexity OMP implementation based on FFT is presented to reduce computational complexity and save memory space. The band limited ICI equalizer is implemented to combat ICI. The receiver operates in an iterative manner (turbo receiver) which can further improve the performance. The receiving system is implemented on a multi-core DSP TMS320C6678 running at 1 GHz. The performance is tested through recorded sea trial data. The computational complexity is affordable when two cores are used and the bit error rate (BER) performance is satisfying.

The rest of paper is organized as follows. Section II introduces the signal model. Section III describes the channel estimator. Section IV presents the DSP implementation and section V shows the performance evaluation.

Notation: Upper-case (resp. lower-case) bold letters refer to matrices (resp. column vectors) with $[A]_{n,m}$ (resp. a_n) being the (n, m) th (resp. n th) entry of \mathbf{A} (resp. \mathbf{a}) and \mathbf{a}_i represents the i th column of matrix \mathbf{A} , unless stated otherwise. $(\cdot)^T$, $(\cdot)^H$, $(\cdot)^*$ and $E(\cdot)$ stand for transpose, Hermitian transpose, conjugate operation and expectation operation respectively. \otimes and \odot refer to Kronecker product and Hadamard product. \mathbf{F} represents normalized Fourier matrix with $\tilde{\mathbf{a}} = \mathbf{F}\mathbf{a}$ being the FFT of \mathbf{a} . $\text{diag}(\mathbf{a})$ refers to a diagonal matrix with \mathbf{a} on the main diagonal; $(\mathbf{a})_{\mathcal{P}}$ stands for a column vector with entries of \mathbf{a} corresponding to the indices from the set \mathcal{P} ; \mathbf{I}_G and $\mathbf{1}_G$ denote a $G \times G$ identity matrix. $\hat{\mathbf{a}}$ is the estimation of \mathbf{a} and $j = \sqrt{-1}$.

II. SIGNAL MODEL

A block of L_b information bits $\{a_1, \dots, a_{L_b}\}$ is encoded by a half rate convolutional channel encoder with generator polynomial $[1, \frac{171}{133}]_8$. The encoded bits are interleaved by a random interleaver and mapped to a vector of complex symbols denoted as $\mathbf{s}[n] = [s_0[n], \dots, s_{N_c-1}[n]]$ for the n th block. The symbol vector $\mathbf{s}[n]$ is modulated on N_c

subcarriers to generate N_c time-domain symbols as $v_k[n] = \frac{1}{\sqrt{N_c}} \sum_{i=0}^{N_c-1} s_i[n] e^{j\frac{2\pi ik}{N_c}}$ which can be efficiently implemented through IFFT. To avoid inter-block interferences, a cyclic prefix (CP) with length of N_p is inserted before each data block. After that, the time-domain symbols are passed through pulse-shaping filter and upconverted on the carrier frequency.

At receiver, the received signal is downconverted and sampled with sample rate T . After removing CP, compensating carrier frequency offset (CFO), we can express the received based band signal within one block as [18]

$$\mathbf{y} = \sum_{q=1}^Q \text{diag}(\mathbf{v}_q) \mathbf{B}[q] \mathbf{x} + \mathbf{w} \quad (1)$$

with BEM which models the time-varying channel impulse response as $h[n, l] = \sum_{q=1}^Q b_{q,l} v_q[n]$. The q th base function is denoted as $v_q[n]$ and \mathbf{v}_q in vector form. The q th BEM coefficient with delay tap l is written as $b_{q,l}$ and $\mathbf{B}[q]$ is a circulant matrix with first column being $[b_{q,0}, \dots, b_{q,L_\tau-1}, 0, \dots, 0]^T$. The noise vector is denoted as \mathbf{w} . Applying FFT, we obtain

$$\tilde{\mathbf{y}} = \sum_{q=1}^Q \Delta[q] \mathbf{F} \mathbf{B}[q] \mathbf{s} + \tilde{\mathbf{w}} \quad (2)$$

where $\Delta[q] = \mathbf{F} \text{diag}(\mathbf{v}_q) \mathbf{F}^H$ and $\mathbf{s} = \mathbf{1}_G \otimes \mathbf{s}$.

III. ITERATIVE CHANNEL ESTIMATION

The channel estimation is based on linear model $\tilde{\mathbf{y}} = \mathbf{U} \boldsymbol{\xi}$, where $\boldsymbol{\xi}$ is a column vector containing BEM coefficients for estimation. The vector $\tilde{\mathbf{y}} = (\tilde{y})_{\mathcal{P}}$ refers to the frequency-domain received signals at subcarrier index set \mathcal{P} . The subcarrier index set \mathcal{P} includes only pilot subcarrier indexes in the first iteration and contains all the subcarrier indexes in the remaining iterations. The matrix \mathbf{U} is written as

$$\mathbf{U} = [(\Delta[1] \text{diag}(\mathbf{f}_1) \mathbf{s}_{\text{soft}})_{\mathcal{P}}, \dots, (\Delta[Q] \text{diag}(\mathbf{f}_Q) \mathbf{s}_{\text{soft}})_{\mathcal{P}}], \quad (3)$$

where \mathbf{s}_{soft} represents soft symbol estimates in one OFDM block.

To explore inherent channel sparsity, the OMP algorithm is used for channel estimation. Moreover, the OMP algorithm does not require any channel statistical information which is usually not available for underwater acoustic channel. The OMP algorithm [19] at s th iteration is summarized as follows

- Solve the optimization problem as

$$t = \arg \max_t \frac{|\mathbf{u}_t^H \tilde{\mathbf{y}}|^2}{\|\mathbf{u}_t\|^2}, t \notin \mathcal{I}_{s-1}, \quad (4)$$

update matrix $\tilde{\mathbf{U}}[s] = [\tilde{\mathbf{U}}[s-1] \quad \mathbf{u}_t]$ and estimate $\boldsymbol{\xi}$ as

$$\hat{\boldsymbol{\xi}} = (\tilde{\mathbf{U}}^H[s] \tilde{\mathbf{U}}[s])^{-1} \tilde{\mathbf{U}}^H[s] \tilde{\mathbf{y}}. \quad (5)$$

- Update $\mathcal{I}_s = \mathcal{I}_{s-1} \cup \{t\}$ and residual vector

$$\tilde{\mathbf{y}} = \tilde{\mathbf{y}} - \tilde{\mathbf{U}}[s] \hat{\boldsymbol{\xi}}. \quad (6)$$



Fig. 1. DSP EVM TMDXEVM6678L

IV. DSP IMPLEMENTATION

The receiving system is implemented on a multi-core DSP EVM TMDXEVM6678L which includes a DSP chip composed of 8 cores. Each core supports both fixed point operation and floating point operation with maximum raw computational performance up to 44.8 GMACS and 22.4 GFLPS (@1.4 GHz operating frequency) [20]. Each core also integrates 32 KB of L1 program and data cache, 512 KB of L2 memory which can be configured as either mapped RAM or cache. In addition, the chip integrates 4 MB of multi-core shared memory (MCSM). MCSM is cached by default. However the cache coherence of MCSM is needed to be maintained by programmer, while the cache coherence of L2 memory is maintained by DSP hardware. The EVM also contains 512 MB external DDR3 memory connecting with DSP device through external memory interface.

In our application, we use two DSP cores (DSP core 0 and DSP core 1) running at 1 GHz and all L2 memory is configured as RAM. DSP core 0 is used as main processor while DSP core 1 is adopted as co-processor. The coordination between two DSP cores is realized by inter-processor communication module. When co-processor is needed, the main processor would send a request through IPC and then execute its own tasks until the results from co-processor are needed. After receiving the request from main processor, the co-processor would execute the task according to the request, send the result to main processor again and get ready for new tasks. After receiving the feedback from co-processor, main processor would forward to following tasks. We use C language to implement our algorithm on DSP. To further improve the processing speed, the compiler is set up with optimization level 3. Besides, we use several programming strategies such as loop unrolling, applying intrinsic operations and using TI libraries to speed up processing time. The 32-bit single precision floating point is used to represent data for most of our implementations except SISO decoder which uses signed Q16.15 data type. The conversion between floating point and signed Q16.15 can be efficiently executed as 2 points per CPU cycle per core. The detailed implementation strategies for each processing unit are shown below.

A. Implementation for CFO compensation

We use the method in [2] for CFO estimation and compensation. The pre-defined possible CFOs are divided into two groups evenly. Each DSP core computes cost function related to one group of CFOs and find out the CFO that minimize the cost function. Finally, DSP core 0 determines the CFO estimation from the two possible CFOs corresponding to each group.

B. Implementation for channel estimation

As shown in (4), $\mathbf{U}^H \tilde{\mathbf{y}}$ must be computed at each iteration of OMP algorithm, which can induce heavy computational burden if computed directly. Moreover, \mathbf{U} would be dependent on soft symbol estimation which means additional process for computing \mathbf{U} would be needed for an iterative receiver. Such process would cause heavy computational burden. However, we actually can adopt FFT to reduce both computational complexity and memory consumption as below.

Denoting $\check{\mathbf{U}}[p]$ to be a matrix containing columns in \mathbf{U} related to p th basis function, we have

$$\begin{aligned} \check{\mathbf{U}}^H[p] \tilde{\mathbf{y}} &= \begin{bmatrix} \mathbf{s}_{\text{soft}}^H \text{diag}(\mathbf{f}_1)^H \Delta^H[p] \tilde{\mathbf{y}} \\ \vdots \\ \mathbf{s}_{\text{soft}}^H \text{diag}(\mathbf{f}_{L_\tau})^H \Delta^H[p] \tilde{\mathbf{y}} \end{bmatrix} \\ &= \mathbf{F}_{1:L_\tau}^H \text{diag}(\mathbf{s}_{\text{soft}})^H \Delta^H[p] \tilde{\mathbf{y}}, \end{aligned} \quad (7)$$

where $\mathbf{s}_{\text{soft}} = \mathbf{1}_G \otimes \mathbf{s}_{\text{soft}}$. Since $\Delta[p]$ is a circulant matrix, $\check{\mathbf{U}}^H[p] \tilde{\mathbf{y}}$ could be efficiently computed through FFT.

In addition, we need to calculate the norm of column vectors in matrix \mathbf{U} at each iteration of the iterative receiver. We would show that FFT can be used to reduce the computational complexity and save memory. We denote

$$\psi[p] = \left[\left[\check{\mathbf{U}}^H[p] \tilde{\mathbf{U}}[p] \right]_{1,1}, \dots, \left[\check{\mathbf{U}}^H[p] \tilde{\mathbf{U}}[p] \right]_{L_\tau, L_\tau} \right]. \quad (8)$$

For $\psi_m[p]$, we can express it as

$$\begin{aligned} \psi_m[p] &= \mathbf{e}_m^H \mathbf{F}^H \text{diag}(\mathbf{s}_{\text{soft}})^H \Delta^H[p] \Delta[p] \text{diag}(\mathbf{s}_{\text{soft}}) \mathbf{F} \mathbf{e}_m \\ &= \mathbf{e}_m^H \mathbf{F}^H \text{diag}(\mathbf{s}_{\text{soft}})^H \mathbf{F} \text{diag}(\tilde{\mathbf{v}}[p]) \mathbf{F}^H \text{diag}(\mathbf{s}_{\text{soft}}) \mathbf{F} \mathbf{e}_m \\ &= \phi_m^H \tilde{\mathbf{v}}[p] \end{aligned} \quad (9)$$

where $\tilde{\mathbf{v}}[p] = \mathbf{v}[p] \odot \mathbf{v}^*[p]$ and $\phi_m = (\mathbf{F}^H \text{diag}(\mathbf{s}_{\text{soft}}) \mathbf{F} \mathbf{e}_m)^* \odot (\mathbf{F}^H \text{diag}(\mathbf{s}_{\text{soft}}) \mathbf{F} \mathbf{e}_m)$. Thus we obtain $\psi[p] = \Phi \tilde{\mathbf{v}}[p]$ with $\phi_m = \Phi \mathbf{e}_m$. Φ would be a circulant matrix so that $\psi[p]$ can be efficiently computed through FFT.

Since we use BEM with order two as channel model, each DSP core would calculate cost function in (4) related to each base function respectively. After determining $\check{\mathbf{U}}[s]$, it is important to note that

$$\check{\mathbf{U}}^H[s] \tilde{\mathbf{y}} = \left[\left(\check{\mathbf{U}}^H[s-1] \tilde{\mathbf{y}} \right)^T \check{\mathbf{u}}^H[s] \tilde{\mathbf{y}} \right]^T \quad (10)$$

can be computed recursively so that we only need to calculate the inner product of two column vectors at each iteration. $\check{\mathbf{U}}^H[s] \tilde{\mathbf{U}}[s]$ can be computed as

$$\check{\mathbf{U}}^H[s] \tilde{\mathbf{U}}[s] = \begin{bmatrix} \check{\mathbf{U}}^H[s-1] \tilde{\mathbf{U}}[s-1] & \check{\mathbf{U}}^H[s-1] \tilde{\mathbf{u}}[s] \\ \check{\mathbf{u}}^H[s] \tilde{\mathbf{U}}[s-1] & \check{\mathbf{u}}^H[s] \tilde{\mathbf{u}}[s] \end{bmatrix}, \quad (11)$$

which means only $\check{\mathbf{U}}^H[s-1] \tilde{\mathbf{u}}[s]$ is required to compute at each OMP iteration. $\check{\mathbf{U}}^H[s-1] \tilde{\mathbf{u}}[s]$ can be computed directly by using two DSP cores when iteration counter s is less than a pre-determined threshold. When the counter s exceeds that threshold, we can use the same method as computing $\mathbf{U}^H \tilde{\mathbf{y}}$ to calculate $\check{\mathbf{U}}^H[s-1] \tilde{\mathbf{u}}[s]$. To compute $\hat{\xi}$, we follow the iterative block Cholesky decomposition method [21]. Denoting the Cholesky decomposition of $\check{\mathbf{U}}^H[s] \tilde{\mathbf{U}}[s]$ is $\check{\mathbf{U}}^H[s] \tilde{\mathbf{U}}[s] = \mathbf{G}^H[s] \mathbf{G}[s]$ and $\mathbf{q}[s] = \check{\mathbf{U}}^H[s] \tilde{\mathbf{y}}$, it is worth noting that the result of solving $\mathbf{G}^H[s] \beta[s] = \mathbf{q}[s]$ can be obtained recursively as

$$\beta[s] = \left[\beta^T[s-1] \frac{q_s[s] - \mathbf{g}_{s,1:s-1}^H \mathbf{q}[s-1]}{[\mathbf{G}[s]]_{s,s}} \right]^T. \quad (12)$$

To compute $\check{\mathbf{U}}[s] \hat{\xi}$, FFT might be also used to reduce computational complexity. When iteration counter s is less than the threshold, we can coordinate two DSP cores to compute it directly. If iteration counter s exceeds the threshold, we could express $\check{\mathbf{U}}[p] \hat{\xi}[p]$ where $\check{\mathbf{U}}[p]$ is a matrix composed of columns in $\check{\mathbf{U}}[s]$ related to the p th base function and $\hat{\xi}[p]$ contains elements in $\hat{\xi}$ corresponding to the p th base function as

$$\check{\mathbf{U}}[p] \hat{\xi}[p] = \Delta[p] \text{diag}(\mathbf{F}_{\mathcal{I}_s[p]} \hat{\xi}[p]) \mathbf{s}_{\text{soft}} \quad (13)$$

where $\mathcal{I}_s[p]$ includes elements in \mathcal{I}_s related to the p th basis function. After OMP algorithm terminates, we can compute the channel estimation as

$$\hat{\mathbf{H}} = \sum_{p=1}^2 \tilde{\Delta}[p] \text{diag}(\mathbf{F}_{\mathcal{I}[p]} \hat{\xi}[p]) \quad (14)$$

where $\tilde{\Delta}[p]$ is a banded approximation version of $\Delta[p]$ and is pre-computed and stored in external DDR3 RAM. \mathcal{I} contains the selected indexes from OMP algorithm and $\mathcal{I}[p]$ includes elements in \mathcal{I} related to the p th base function.

C. Implementation for ICI equalizer

We apply banded approximation for channel matrix so that subcarrier by subcarrier MMSE ICI equalizer as [22] which can naturally be parallel implemented by multi-core processor. In our implementation, all subcarriers that carry information symbols are divided into two groups. Each DSP core is responsible for equalizing and demapping one group of information symbols. We only need to compute half of the covariance matrix since it is a Hermitian matrix. The soft demapper is implemented as same as normal turbo equalizer and can refer to [23].

V. PERFORMANCE EVALUATION

The system performance is evaluated through experimental data from sea trial at the Trondheim fjord, Norway. The system setup is presented in Table I. As shown in Table I, the block duration (including CP) is up to 562 ms. If the data processing can be finished within this period, real-time processing can be achieved. The time-domain oversampling technique has been shown that it is capable of improving the performance of

TABLE I
SYSTEM CONFIGURATION

Carrier frequency	12 KHz
CP duration	50 ms
Number of subcarriers	1024
Symbol rate	2 ks/s
Modulation scheme	8-PSK

TABLE II
CPU PROCESSING TIME

Function	Processing time
CFO compensation	0.76 ms
Vector norm calculation	0.101 ms
OMP channel estimation	87.7 ms
ICI equalization & Soft demapper	5.25 ms
SISO decoder	6.77 ms
Soft mapper	0.347 ms
Total time per iteration	101 ms

OFDM systems [18] so that an oversampling factor of two is employed in our implementation. All the subcarriers are divided into groups every eight subcarriers. The pilot structure for each group is shown in (15) where P, 0 and D refer to pilot position, null position and data position respectively. The pilots and null are used for channel estimation in the 1st iteration, while pilots, null and soft data estimation are employed for channel estimation in the later iterations. We adopt discrete prolate spheroidal sequences as base functions [14] in order to avoid Gibbs effect caused by complex exponential base functions. The order of BEM is chosen to be two and the one-side band size for equalizer is selected to be two. The OMP algorithm terminates after 180 iterations and the CFO compensation coefficient is selected from a group of possible values with group length of 32.

$$[0 \ P \ P \ 0 \ D \ D \ D \ D] \quad (15)$$

The CPU processing time for each function is presented in Table II. As presented in Table II, it consumes 101 ms to finish one iteration. If the maximum iteration number for the receiver is three, the receiver requires 303 ms to finish one data block processing, which is less than the duration of each data block so that real-time processing is able to be obtained. It is worth noting that the conventional LS would require to compute $\mathbf{U}^H \mathbf{U}$ and its inversion where \mathbf{U} is a 2048×400 matrix. Such operation would need large computational resources. Besides, the iteration number for OMP algorithm could be reduced when channel exhibits much stronger sparsity. It is also notable that the performance can be further improved by increasing CPU clock rate (the DSP processor can run upto 1.4 GHz) or using more DSP cores.

During the experiments, an 8-element receiving hydrophone array with vertical spatial distance between two adjacent hydrophones at 1 m was close to shore deployed, ranging from 1.5 m to 8.5 m. The transmitting transducer was deployed

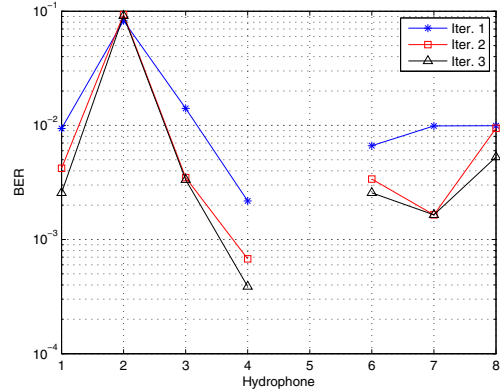


Fig. 2. BER for location 1

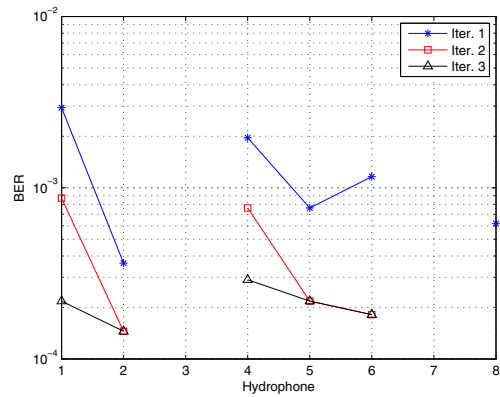


Fig. 3. BER for location 2

from NTNU's research vessel Gunnerus with a depth of 15 m. The vessel was located at two different locations with different horizontal distances from the receiving array. At location 1, approximately 1 km away from the receiver, three burst transmissions with 11 blocks (in total 20736 bits) were conducted, while four burst transmissions (in total 27648 bits) were carried out at location 2 which is approximately 1.3 km away from the receiver array. The achieved average bit error rate performances for each hydrophone at location 1 and 2 are shown in Fig. 2 and Fig. 3 respectively. The achieved BER for location 1 after 3 iteration is on the order of 10^{-3} except hydrophone 2 while the BER for location 2 is even lower since the sea state is much calm at location 2.

VI. CONCLUSION

This paper investigates the DSP implementation of an iterative OFDM receiver for time-varying underwater acoustic channel. Based on BEM, the OMP channel estimation is efficiently implemented through FFT which is able to reduce executing time and save memory space. With careful optimization on the program, the real-time processing is achieved. The DSP based OFDM receiver is also validated through recorded data from sea experiments and the BER performance

is satisfying. The proposed implementation can be integrated into future advanced underwater acoustic modem.

REFERENCES

- [1] J. W. Choi, T. Riedl, K. Kim, A. Singer, and J. Preisig, "Adaptive linear turbo equalization over doubly selective channels," *IEEE J. Ocean. Eng.*, vol. 36, no. 4, pp. 473–489, Oct 2011.
- [2] B. Li, S. Zhou, M. Stojanovic, L. Freitag, and P. Willett, "Multicarrier communication over underwater acoustic channels with nonuniform doppler shifts," *IEEE J. Ocean. Eng.*, vol. 33, no. 2, pp. 198–209, April 2008.
- [3] M. Stojanovic, "Low complexity OFDM detector for underwater acoustic channels," in *OCEANS 2006*, Sept 2006, pp. 1–6.
- [4] C. Berger, S. Zhou, J. Preisig, and P. Willett, "Sparse channel estimation for multicarrier underwater acoustic communication: From subspace methods to compressed sensing," *IEEE Trans. Signal Process.*, vol. 58, no. 3, pp. 1708–1721, March 2010.
- [5] J. Huang, S. Zhou, J. Huang, C. Berger, and P. Willett, "Progressive inter-carrier interference equalization for ofdm transmission over time-varying underwater acoustic channels," *IEEE J. Sel. Topics Signal Process.*, vol. 5, no. 8, pp. 1524–1536, Dec 2011.
- [6] K. Tu, D. Fertoni, T. Duman, M. Stojanovic, J. Proakis, and P. Hursky, "Mitigation of intercarrier interference for ofdm over time-varying underwater acoustic channels," *IEEE J. Ocean. Eng.*, vol. 36, no. 2, pp. 156–171, April 2011.
- [7] Z. Wang, S. Zhou, G. Giannakis, C. Berger, and J. Huang, "Frequency-domain oversampling for zero-padded OFDM in underwater acoustic communications," *IEEE J. Ocean. Eng.*, vol. 37, no. 1, pp. 14–24, Jan 2012.
- [8] T. Husoy, M. Pettersen, B. Nilsson, T. Oberg, N. Warakagoda, and A. Lie, "Implementation of an underwater acoustic modem with network capability," in *OCEANS, 2011 IEEE - Spain*, June 2011, pp. 1–10.
- [9] H. Yan, L. Wan, S. Zhou, Z. Shi, J.-H. Cui, J. Huang, and H. Zhou, "DSP based receiver implementation for OFDM acoustic modems," *Physical Communication*, vol. 5, no. 1, pp. 22 – 32, 2012. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S1874490711000474>
- [10] Y. Zheng, Z. Yang, M. Yue, B. Han, Z. Chen, and J. Wang, "Dsp implementation of direct-sequence spread spectrum underwater acoustic modems with networking capability," in *Oceans - St. John's, 2014*, Sept 2014, pp. 1–5.
- [11] L. Freitag, M. Grund, S. Singh, J. Partan, P. Koski, and K. Ball, "The whoi micro-modem: an acoustic communications and navigation system for multiple platforms," in *OCEANS, 2005. Proceedings of MTS/IEEE*, Sept 2005, pp. 1086–1092 Vol. 2.
- [12] B. Peng and H. Dong, "DSP based real-time single carrier underwater acoustic communications using frequency domain turbo equalization," *Physical Communication*, vol. 18, Part 1, pp. 40 – 48, 2016. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S1874490715000634>
- [13] Z. Tang, R. Cannizzaro, G. Leus, and P. Banelli, "Pilot-assisted time-varying channel estimation for OFDM systems," *IEEE Trans. Signal Process.*, vol. 55, no. 5, pp. 2226–2238, May 2007.
- [14] T. Zemen and C. Mecklenbrauker, "Time-variant channel estimation using discrete prolate spheroidal sequences," *IEEE Trans. Signal Process.*, vol. 53, no. 9, pp. 3597–3607, Sept 2005.
- [15] P. Salvo Rossi and R. Muller, "Slepian-based two-dimensional estimation of time-frequency variant MIMO-OFDM channels," *IEEE Signal Process. Lett.*, vol. 15, pp. 21–24, 2008.
- [16] F. Pena-Campos, R. Carrasco-Alvarez, O. Longoria-Gandara, and R. Parra-Michel, "Estimation of fast time-varying channels in OFDM systems using two-dimensional prolate," *IEEE Trans. Wireless Commun.*, vol. 12, no. 2, pp. 898–907, February 2013.
- [17] W. Li and J. Preisig, "Estimation of rapidly time-varying sparse channels," *IEEE J. Ocean. Eng.*, vol. 32, no. 4, pp. 927–939, Oct 2007.
- [18] B. Peng, P. Salvo Rossi, H. Dong, and K. Kansanen, "Time-domain oversampled OFDM communication in doubly-selective underwater acoustic channels," *IEEE Commun. Lett.*, vol. 19, no. 6, pp. 1081–1084, June 2015.
- [19] T. Cai and L. Wang, "Orthogonal matching pursuit for sparse signal recovery with noise," *IEEE Trans. Inf. Theory*, vol. 57, no. 7, pp. 4680–4688, July 2011.
- [20] "TMS320C6678 multicore fixed and floating-point digital signal processor (rev. e)."
- [21] G. Golub and C. Van Loan, *Matrix Computations*, ser. Johns Hopkins Studies in the Mathematical Sciences. Johns Hopkins University Press, 2012.
- [22] P. Schniter, "Low-complexity equalization of OFDM in doubly selective channels," *IEEE Trans. Signal Process.*, vol. 52, no. 4, pp. 1002–1011, April 2004.
- [23] A. Dejonghe and L. Vandendorpe, "Turbo-equalization for multilevel modulation: an efficient low-complexity scheme," in *Communications, 2002. ICC 2002. IEEE International Conference on*, vol. 3, 2002, pp. 1863–1867 vol.3.