A MAP-Based Order Estimation Procedure for Sparse Channel Estimation

Sajad Daei[1(✉)], Massoud Babaie-Zadeh[1], and Christian Jutten[2]

[1] Department of Electrical Engineering, Sharif University of Technology, Tehran, Iran
{sajad2008se,mbzadeh}@yahoo.com

[2] GIPSA-Lab, Institut Universitaire de France, Grenoble, France
christian.jutten@gipsa-lab.grenoble-inp.fr

Abstract. Recently, there has been a growing interest in estimation of sparse channels as they are observed in underwater acoustic and ultra-wideband channels. In this paper we present a new Bayesian sparse channel estimation (SCE) algorithm that, unlike traditional SCE methods, exploits noise statistical information to improve the estimates. The proposed method uses approximate maximum a posteriori probability (MAP) to detect the non-zero channel tap locations while least square estimation is used to determine the values of the channel taps. Computer simulations shows that the proposed algorithm outperforms the existing algorithms in terms of normalized mean squared error (NMSE) and approaches Cramér-Rao lower bound of the estimation. In addition, it has low computational cost when compared to the other algorithms.

Keywords: Bayesian · Sparse channel estimation · Cramér-Rao lower bound

1 Introduction

Fast and accurate channel estimation at the receiver is often of much importance due to the need for optimal demodulation and decoding in limited time. Sparse channels, those whose time domain impulse response has much less non-zero taps than their length, have been observed in many practical scenarios such as acoustic underwater [1], ultrawideband propagation [2] and seismic exploration [3]. Since traditional channel estimation methods, such as the least squares method, fail to exploit sparsity of these channels, in the last decade, several sparse channel estimation (SCE) methods have been proposed to improve the estimates [4–10].

In [4,10], two iterative approaches called ITD-SE and MIDE are reported which utilize thresholds to detect the channel support[1] followed by a structured

C. Jutten—This work has been partly funded by ERC project 2012-ERC-AdG-320684 CHESS.

[1] Non-zero channel tap locations.

E. Vincent et al. (Eds.): LVA/ICA 2015, LNCS 9237, pp. 344–351, 2015.
DOI: 10.1007/978-3-319-22482-4_40

least square (LS) estimate to determine the values of the channel taps. Simple structure, low complexity and no dependency on the channel order along with acceptable accuracy are the advantages of these threshold-based support detection methods. Moreover, in [5], an iterative MAP-based approach is introduced to jointly estimate the location and the values of the taps. For this purpose, three algorithms have been examined: L2MAP with Threshold, LASSO-MAP with Threshold and Backward-Detection MAP. These methods have very low complexity and near-optimal performance at high SNRs; nevertheless they presume the channel sparsity level is known a priori while it's rarely known in practice. Furthermore, they have limitations on the sparsity rate of the channel.

The algorithms mentioned above, neglect noise statistical information and posterior information of the channel support, which can explain their limited performance. In this paper, to overcome the problems mentioned, as in [4], we present a two stage Bayesian procedure, based on support detection and then channel estimation. For the former part, following [11], we propose a MAP-based tap detection approach which not only considers sparsity of the channel but also exploits noise statistics and posterior information of the channel support to improve the estimates; and for the latter, a structured least square estimation is applied. Unlike Bayesian approaches in SCE algorithms that usually assume Gaussian distribution for the channel, regarding the procedure in [11], the channel distribution in our algorithm is arbitrary. Experimental results demonstrate that our algorithm approaches the Cramér-Rao lower bound of the estimation based on knowing the true channel support (called CRB-S in [4]) at high SNRs. Besides, it has a low computational cost. Note that as our main contribution to SCE algorithms is a support detection approach using approximate MAP, we call it Support Detection using Maximum A posteriori Probability (SDMAP) in this paper.

The paper is organized as follows: System Model and MAP Setup are given in Sect. 2. In Sect. 3, SDMAP algorithm is proposed. Experimental results are investigated in Sect. 4 in order to compare the performance of SDMAP with existing algorithms in term of normalized mean square error (NMSE) and computational complexity. Finally, we conclude the paper in Sect. 5.

Notation: Throughout the paper, we denote scalars with lowercase letters (e.g., x), vectors with lowercase boldfaced letters (e.g., $\mathbf{x}$) and matrices with uppercase boldfaced letters (e.g., $\mathbf{X}$). $\mathbf{x}_i$ stands for the ith column of the matrix $\mathbf{X}$. Sets are designated by uppercase calligraphic letters; the cardinality of the set $\mathcal{S}$ is $|\mathcal{S}|$. We use $\mathbf{x}_{\mathcal{S}}$ to denote the $|\mathcal{S}|$- dimensional vector of the entries in the vector $\mathbf{x}$ indexed by S. Also, for any $m \times n$ matrix $\mathbf{X}$, we use $\mathbf{X}_{\mathcal{S}}$ to denote the $m \times |\mathcal{S}|$ matrix corresponding to the columns of $\mathbf{X}$ indexed by $\mathcal{S}$ and $\mathbf{X}(\mathcal{S}_1, \mathcal{S}_2)$ to denote the $|\mathcal{S}_1| \times |\mathcal{S}_2|$ matrix corresponding to the rows and columns of $\mathbf{X}$ indexed by $\mathcal{S}_1$ and $\mathcal{S}_2$ respectively. $\mathbf{I}_m$ is denoted for the $m \times m$ identity matrix. $\|\mathbf{x}\|$ means the 2-norm of the vector $\mathbf{x}$. Finally, $|\mathbf{x}|$ stands for a vector whose elements are the absolute values of the corresponding elements of $\mathbf{x}$.

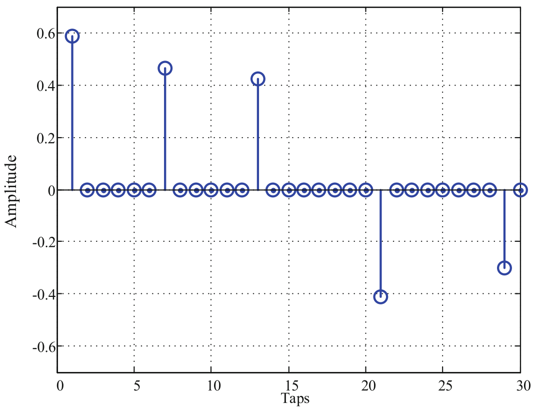

Fig. 1. Time domain discrete impulse response of a sparse channel

2 System Model and MAP Setup

2.1 System Model

Typically, channel estimation is accomplished by sending a training sequence and processing the channel output. Mathematically, let $\{u_n\}_{i=1}^{L}, L \in \mathbb{N}$ denote a training sequence and $\mathbf{h} \in \mathbb{R}^N, N \in \mathbb{N}$ be the finite discrete impulse response of the channel (See Fig. 1). The resulting observations $\mathbf{y} \in \mathbb{R}^M, M = L+N-1$ are the convolution of the training signal $\mathbf{u} = [u_1, u_2, u_3, ..., u_L]^T$ and the impulse response $\mathbf{h} = [h_1, h_2, h_3, ..., h_N]^T$ corrupted by an additive noise vector $\mathbf{n}$. In matrix form, we have

$$\mathbf{y} = \mathbf{U}\mathbf{h} + \mathbf{n} = \mathbf{U}_{\mathcal{S}}\mathbf{h}_{\mathcal{S}} + \mathbf{n}, \tag{1}$$

where, $\mathbf{n} \sim \mathcal{N}(\underline{\mathbf{0}}, \sigma^2\mathbf{I}_M)$ is an $M \times 1$ Gaussian noise vector, $\mathcal{S}$ is the true support set of $\mathbf{h}$ and $\mathbf{U}$ is the $M \times N$ training Toeplitz matrix with the first column $[u_1, u_2, u_3, ..., u_L, 0, ..., 0]^T$ as in [4].

We assume that the sparse channel vector $\mathbf{h}$ is modeled as $\mathbf{h} = \mathbf{h}_B \odot \mathbf{h}_G$, in which $\odot$ is element-wise Hadamard multiplication, $\mathbf{h}_B$ is an $N \times 1$ vector, whose elements are independent and identically distributed (i.i.d) Bernoulli random variables with success probability $P_a = \frac{|\mathcal{S}|}{N}$ and the elements of $\mathbf{h}_G$ are drawn from an arbitrary distribution. Clearly, $\mathbf{h}_B$ models the support of $\mathbf{h}$, with a sparsity level equal to P_a.

2.2 MAP Setup

The goal is to estimate $\mathbf{h}$ from knowledge of the observation vector $\mathbf{y}$ and the training signal $\mathbf{u}$. To achieve this goal, first, we obtain an estimate of the channel support, $\mathcal{S}$, via MAP detection procedure, which is given by,

$$\hat{\mathcal{S}}_{\text{MAP}} = \underset{\mathcal{S}}{\operatorname{argmax}}\ \mathbb{P}\{\mathbf{y}|\mathcal{S}\}\mathbb{P}\{\mathcal{S}\}, \tag{2}$$

in which, $\mathbb{P}$ denotes the probability distribution. Since each element of $\mathbf{h}$ is active according to a Bernoulli distribution with success probability P_a, $\mathbb{P}(\mathcal{S})$ is given by,

$$\mathbb{P}\{\mathcal{S}\} = {P_a}^{|\mathcal{S}|}(1 - P_a)^{N-|\mathcal{S}|}. \tag{3}$$

Rather than obtaining the probability of $\mathbf{y}$ conditioned on the support, $\mathbb{P}\{\mathbf{y}|\mathcal{S}\}$, directly, we serve the approach in [11] to make our algorithm independent of the channel distribution. For this purpose, we project $\mathbf{y}$ onto the orthogonal complement of $\mathbf{U}_{\mathcal{S}}$ via multiplying (1) by $\mathbf{\Pi}_{\mathcal{S}}^{\perp} = \mathbf{I}_M - \mathbf{U}_{\mathcal{S}}({\mathbf{U}_{\mathcal{S}}}^T\mathbf{U}_{\mathcal{S}})^{-1}{\mathbf{U}_{\mathcal{S}}}^T$ which leads to $\mathbf{\Pi}_{\mathcal{S}}^{\perp}\mathbf{y} = \mathbf{\Pi}_{\mathcal{S}}^{\perp}\mathbf{n} \sim \mathcal{N}(\underline{\mathbf{0}}, \sigma^2\mathbf{\Pi}_{\mathcal{S}}^{\perp})$. Ignoring constant multiplicative factors, we have,

$$\begin{aligned}\mathbb{P}\{\mathbf{y}|\mathcal{S}\} &\mathrel{\underset{\sim}{\propto}} \mathbb{P}\{\mathbf{\Pi}_{\mathcal{S}}^{\perp}\mathbf{y}|\mathcal{S}\} \\ &\mathrel{\underset{\sim}{\propto}} \exp\Big(-\frac{1}{2}(\mathbf{\Pi}_{\mathcal{S}}^{\perp}\mathbf{y})^T(\sigma^2\mathbf{\Pi}_{\mathcal{S}}^{\perp})^{-1}(\mathbf{\Pi}_{\mathcal{S}}^{\perp}\mathbf{y})\Big),\end{aligned} \tag{4}$$

in which, $\underset{\sim}{\propto}$ denotes approximate proportion[2]. Since evaluation of the support in (2) leads to prohibitive computational task, alternatively, we propose a support detection procedure in the next section that requires a fitness function which is defined by,

$$\begin{aligned}\mu(\mathcal{S}) &\triangleq \ln\Big(\mathbb{P}\{\mathbf{y}|\mathcal{S}\}\mathbb{P}\{\mathcal{S}\}\Big) \\ &= \ln\Big(\exp(-\frac{1}{2\sigma^2}(\mathbf{y}^T\mathbf{\Pi}_{\mathcal{S}}^{\perp}\mathbf{y}))\Big) + \ln\Big({P_a}^{|\mathcal{S}|}(1-P_a)^{N-|\mathcal{S}|}\Big) \\ &\propto \frac{1}{\sigma^2}\Big(\mathbf{y}^T\mathbf{U}_{\mathcal{S}}(\mathbf{U}_{\mathcal{S}}^T\mathbf{U}_{\mathcal{S}})^{-1}\mathbf{U}_{\mathcal{S}}^T\mathbf{y}\Big) + 2|\mathcal{S}|\ln\Big(\frac{|\mathcal{S}|}{N-|\mathcal{S}|}\Big).\end{aligned} \tag{5}$$

After finding dominant channel support using the SDMAP scheme of the next section, it only remains to determine the values of the channel taps at the obtained support. To accomplish this, structured least square estimation is applied as follows,

$$\hat{\mathbf{h}} = (\mathbf{U}_{\mathcal{S}}^T\mathbf{U}_{\mathcal{S}})^{-1}\mathbf{U}_{\mathcal{S}}^T\mathbf{y}. \tag{6}$$

3 Proposed Algorithm for Support Detection (SDMAP)

In this section, we introduce our algorithm to detect the channel support. This algorithm is presented in two steps, first support candidates selection and then estimation of the channel order.

3.1 Support Candidates Selection

To obtain support candidates, first, we compute unstructured least square estimate, $\hat{\mathbf{h}} = (\mathbf{U}^T\mathbf{U})^{-1}\mathbf{U}^T\mathbf{y}$, and sort the absolute value of the elements in $\hat{\mathbf{h}}$, $|\hat{\mathbf{h}}|$, in descending order and keep the respective indices $\mathcal{S}$.

[2] A justification of (4) for Gaussian channels is given in [11].

3.2 Order Estimation Procedure

To obtain the channel order, first, we initialize the channel order by,

$$P = \#\{|\hat{\mathbf{h}}_i| > \frac{\max(|\hat{\mathbf{h}}|)}{2}, 1 \leqslant i \leqslant N, i \in \mathbb{N}\}, \tag{7}$$

in which, $\max(|\mathbf{h}|)$ stands for the largest element in the vector $|\mathbf{h}|$. Regarding the initial order P, we determine the direction toward which the current support, $\mathcal{S}_P = \{\mathcal{S}(i), i = 1, ..., P\}$, is inclined. In this regard, to determine the move direction we use some criteria which will be discussed further. After finding the direction toward which the current support tends to move, the support order changes until some stopping criteria are satisfied or the number of maximum move steps is exceeded.

Forming the direction and stopping criteria suitably, requires the knowledge of the noisy part of the fitness function (5). To choose suitable stopping rules, we can exploit the pure noisy part of the fitness function which is given by,

$$\mu_{\mathbf{n}}(\mathcal{S}) = \frac{1}{\sigma^2}\Big(\mathbf{n}^T \underbrace{\mathbf{U}_{\mathcal{S}}(\mathbf{U}_{\mathcal{S}}^T\mathbf{U}_{\mathcal{S}})^{-1}\mathbf{U}_{\mathcal{S}}^T}_{\mathbf{H}}\mathbf{n}\Big). \tag{8}$$

Since $\mathbf{H}$ in (8) is a symmetric, idempotent matrix with $\text{rank}(\mathbf{H}) = |\mathcal{S}|$, $\mu_{\mathbf{n}}(\mathcal{S})$ is chi-squared distributed with $\nu = |\mathcal{S}|$ degrees of freedom, i.e. $\chi^2(\nu)$, in which ν is the parameter of the chi-squared distribution [12, Theorem A.87]. As the mean and variance of this chi-squared distributed random variable are ν and 2ν, respectively, we can obtain tolerance limits[3] of $\mu_{\mathbf{n}}(\mathcal{S})$ as follow,

$$\begin{aligned} \text{Lower-bound:} \quad & lb(\nu) = \nu - \alpha\sqrt{2\nu}; \\ \text{Upper-bound:} \quad & ub(\nu) = \nu + \beta\sqrt{2\nu}, \end{aligned} \tag{9}$$

in which, α and β are chosen such that about 10 % of the distribution occurs outside the bounds in (9). In our simulations, we used $\alpha = 1.1$ and $\beta = 1.3$. Considering the effect of noise on the fitness function (5) and trying to reduce it, we define the direction criterion as follows,

$$\begin{cases} \text{Move backward (i.e. keep } P-1\text{) if,} \\ \mu(\mathcal{S}_{P-1}) - \mu(\mathcal{S}_P) > lb(P-1) - ub(P) \quad \& \\ \mu(\mathcal{S}_{P+1}) - \mu(\mathcal{S}_P) < ub(P+1) - lb(P); & \text{(10a)} \\ \text{Move forward (i.e. keep } P+1\text{) if,} \\ \mu(\mathcal{S}_{P-1}) - \mu(\mathcal{S}_P) < lb(P-1) - ub(P) \quad \& \\ \mu(\mathcal{S}_{P+1}) - \mu(\mathcal{S}_P) > ub(P+1) - lb(P); & \text{(10b)} \\ \text{Don't move if,} \\ \text{neither of the two above is satisfied.} & \text{(10c)} \end{cases}$$

[3] Tolerance limits are the bounds that the probability of random variable occurrence outside them is a certain value.

Qualitatively spoken, moving backward simultaneously requires that the tendency of the current support to change to the previous support (i.e. $\mu(\mathcal{S}_{P-1}) - \mu(\mathcal{S}_P)$) be greater than the lower limit of the noisy part and the tendency to change to the next support (i.e. $\mu(\mathcal{S}_{P+1}) - \mu(\mathcal{S}_P)$) be less than the upper limit of the noisy part. Likewise, moving forward simultaneously requires that the tendency to change to the previous support be less than the lower limit of the noisy part and the tendency to change to the next support be greater than the upper limit of the noisy part.

After finding the direction using (10), the algorithm moves in the obtained direction until approximately no change is observed in the fitness function. This is accomplished by using a stopping rule which is given by:

$$\mu(\mathcal{S}_{P_{\text{new}}}) - \mu(\mathcal{S}_{P_{\text{pre}}}) < ub(P_{\text{new}}) - lb(P_{\text{pre}}), \tag{11}$$

Algorithm 1. SDMAP Algorithm

procedure SDMAP($\mathbf{U}$, $\mathbf{y}$, σ^2)
  $\mathbf{corr} = \mathbf{y}^T\mathbf{U}$, $\mathbf{A} = \mathbf{U}^T\mathbf{U}$
  $\hat{\mathbf{h}} = \mathbf{A}^{-1}\mathbf{corr}$
  Sort the elements in $|\hat{\mathbf{h}}|$ in descending order and save the respective indexes in $\mathcal{S}_c$
  $P_{\text{init}} \leftarrow \#\{|\hat{\mathbf{h}}_i| > \frac{\max(|\hat{\mathbf{h}}|)}{2}, 1 \leqslant i \leqslant N, i \in \mathbb{N}\}$
  if (10a) is satisfied **then**
    $P \leftarrow P_{init}$
    $\mu_{\text{new}} \leftarrow \mu(\mathcal{S}_{P-1})$
    repeat
      $P \leftarrow P - 1$
      $\mu_{\text{old}} \leftarrow \mu_{\text{new}}$
      $\mu_{\text{new}} \leftarrow \mu(\mathcal{S}_{P-1})$
    until $\mu_{\text{new}} - \mu_{\text{old}} < ub(P-1) - lb(P) \;\vee\; P = 1$
  else if (10b) is satisfied **then**
    $P \leftarrow P_{init}$
    $\mu_{\text{new}} \leftarrow \mu(\mathcal{S}_{P+1})$
    repeat
      $P \leftarrow P + 1$
      $\mu_{\text{old}} \leftarrow \mu_{\text{new}}$
      $\mu_{\text{new}} \leftarrow \mu(\mathcal{S}_{P+1})$
    until $\mu_{\text{new}} - \mu_{\text{old}} < ub(P+1) - lb(P) \;\vee\; P = N$
  end if
  $\mathcal{S}_f = \mathcal{S}_c(1 : P)$
  $\hat{\mathbf{h}}_f = \mathbf{A}(\mathcal{S}_f, \mathcal{S}_f)^{-1}\mathbf{corr}(\mathcal{S}_f)$
  function $\mu(\mathcal{S})$
    $F = \frac{1}{\sigma^2}(\mathbf{corr}(\mathcal{S})\mathbf{A}(\mathcal{S},\mathcal{S})^{-1}\mathbf{corr}(\mathcal{S})^T) + 2|\mathcal{S}|\ln(\frac{|\mathcal{S}|}{N-|\mathcal{S}|})$
    return F
  end function
  Output $\hat{\mathbf{h}}_f$
end procedure

in which, P_{pre} and P_{new} are the previous and new orders, respectively. (11) qualitatively expresses that the algorithm stops when the tendency of the previous support $\mathcal{S}_{P_{\text{pre}}}$ to change to the new support $\mathcal{S}_{P_{\text{new}}}$ (i.e. $\mu(\mathcal{S}_{P_{\text{new}}}) - \mu(\mathcal{S}_{P_{\text{pre}}})$) is less than the upper bound of the noisy part. The final pseudocode form of our algorithm is given in Algorithm 1.

4 Computer Simulations

In this section, we investigate the performance of our algorithm (SDMAP) in comparison with five algorithms: L2MAP [5], Backward-MAP [5], LASSO-MAP [5], MIDE [10] and ITD-SE [4]. For this purpose, we consider a sparse channel with length $N = 30$ and support size $|\mathcal{S}| = 5$ (see Fig. 1) and draw the elements of the training matrix, $\mathbf{U}_{M\times N}, M = 50$, from a zero-mean i.i.d. Gaussian distribution ($\mathcal{N}(0, \frac{1}{N})$). The estimation efficiency is evaluated using normalized mean squared error (NMSE) which is defined as,

$$\text{NMSE} = \frac{1}{N_{\text{MC}}} \sum_{n=1}^{N_{\text{MC}}} \frac{\|\mathbf{h} - \hat{\mathbf{h}}_n\|^2}{\|\mathbf{h}\|^2}, \tag{12}$$

where, N_{MC} is the number of Monte Carlo iterations, $\hat{\mathbf{h}}_n$ is the channel estimator in the n^{th} experiment and $\mathbf{h}$ is the true channel. To compare computational complexity of the proposed algorithm with other methods, we use CPU time as a simple metric. Our simulation is implemented using MATLAB 2012 on a laptop computer with 2.4 GHz Intel i5 processor and 4 GB memory running the Windows 7 64 bit operating system.

From Fig. 2(a), we observe that SDMAP algorithm outperforms all the other compared algorithms in the sense of NMSE and approaches the theoretic lower bound CRB-S at high SNRs. Figure 2(b) demonstrates the computational efficacy of our algorithm over the other methods.

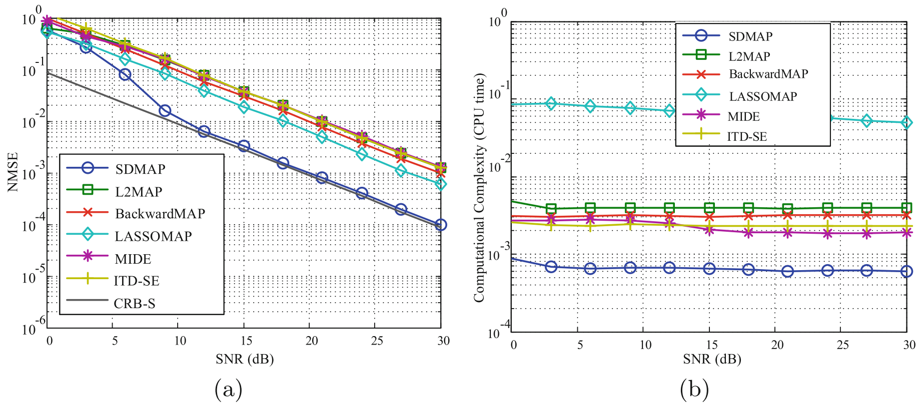

Fig. 2. Performance comparison. (a) NMSE versus SNR. (b) Computational complexity versus SNR

5 Conclusion

In this paper, we proposed a new Bayesian strategy for channel estimation called SDMAP. As it can be seen from simulation results, SDMAP has strengths in terms of NMSE and low computational cost. The reason is that our algorithm utilizes a priori information of noise in the support detection stage. The use of noise statistics provides the posterior information of the channel support, and finally leads to reducing the misdetection.

References

1. Kilfoyle, D.B., Baggeroer, A.B.: The state of the art in underwater acoustic telemetry. IEEE J. Oceanic Eng. **25**(1), 4–27 (2000)
2. Molisch, A.F.: Ultrawideband propagation channels-theory, measurement, and modeling. IEEE Trans. Veh. Technol. **54**(5), 1528–1545 (2005)
3. Claerbout, J.F., Muir, F.: Robust modeling with erratic data. Geophysics **38**(5), 826–844 (1973)
4. Carbonelli, C., Vedantam, S., Mitra, U.: Sparse channel estimation with zero tap detection. IEEE Trans. Wirel. Commun. **6**(5), 1743–1763 (2007)
5. Niazadeh, R., Babaie-Zadeh, M., Jutten, C.: An alternating minimization method for sparse channel estimation. In: Vigneron, V., Zarzoso, V., Moreau, E., Gribonval, R., Vincent, E. (eds.) LVA/ICA 2010. LNCS, vol. 6365, pp. 319–327. Springer, Heidelberg (2010)
6. Subramanian, S.: Compressed sensing for sparse underwater channel estimation: Some practical considerations (2010). arXiv preprint arXiv:1002.2677
7. Sharp, M., Scaglione, A.: Estimation of sparse multipath channels. In: Military Communications Conference, MILCOM, pp. 1–7. IEEE (2008)
8. Cotter, S.F., Rao, B.D.: Sparse channel estimation via matching pursuit with application to equalization. IEEE Trans. Commun. **50**(3), 374–377 (2002)
9. Karabulut, G.Z., Yongacoglu, A.: Sparse channel estimation using orthogonal matching pursuit algorithm. In: Vehicular Technology Conference, VTC, vol. 6, pp. 3880–3884. IEEE (2004)
10. Wan, F., Mitra, U., Molisch, A.: The modified iterative detector/estimator algorithm for sparse channel estimation. In: OCEANS, pp. 1–6. IEEE (2010)
11. Masood, M., Al-Naffouri, T.Y.: Sparse reconstruction using distribution agnostic Bayesian matching pursuit. IEEE Trans. Sig. Process. **61**(21), 5298–5309 (2013)
12. Rao, C.R., Toutenburg, H., Shalabh, H.C., Schomaker, M.: Linear Models and Generalizations: Least Squares and Alternatives, 3rd edn. Springer, Heidelberg (2008)