Published in IET Signal Processing
Received on 2nd December 2010
Revised on 10th May 2011
doi: 10.1049/iet-spr.2010.0347

ISSN 1751-9675

Underwater acoustic channel estimation based on sparse recovery algorithms

C. Qi[1,2] *X. Wang*[3] *L. Wu*[1]

[1]*School of Information Science and Engineering, Southeast University, Nanjing 210096, People's Republic of China*
[2]*Key Laboratory of Underwater Acoustic Signal Processing of Ministry of Education, Nanjing, People's Republic of China*
[3]*Department of Electrical Engineering, Columbia University, New York, NY 10027, USA*
E-mail: wangx@ee.columbia.edu

Abstract: The authors consider underwater acoustic (UWA) channel estimation based on sparse recovery using the recently developed homotopy algorithm. The UWA communication system under consideration employs orthogonal frequency-division multiplexing (OFDM) and receiver preprocessing to compensate for the Doppler effect before channel estimation. The authors first extend the original homotopy algorithm which is for real-valued signals to the complex field. The authors then propose two enhancements to the sparse recovery-based UWA channel estimator by exploiting the UWA channel temporal correlations, including the use of a first-order Gauss–Markov model and the recursive least-squares algorithm for channel tracking. Moreover the authors propose a scheme to optimise the pilot placement over the OFDM subcarriers based on the discrete stochastic approximation. Simulation results show that the homotopy algorithm offers faster and more accurate UWA channel estimation performance than other sparse recovery methods, and the proposed enhancements and pilot placement optimisation offer further performance improvement.

1 Introduction

Underwater acoustic (UWA) channel has been regarded as one of the most difficult communication medium because of the large delay spread and the frequency-dependent Doppler spread [1]. Since the bandwidth is usually comparable with the carrier frequency, the system is essentially wideband. On the other hand, the channel attenuation that becomes more severe for higher frequency limits the acoustic propagation and the available bandwidth. Earlier systems employ the frequency shift keying (FSK) to modulate signals onto discrete frequency tones with guard time and guard bands [2]. Later coherent demodulation techniques are adopted to increase the data rate and spectral efficiency by adaptive phase tracking and equalisation [3]. More recently, the orthogonal frequency-division multiplexing (OFDM) technique, which has prevailed in wireless communication systems, has also been applied to UWA communications [4]. OFDM transforms the frequency-selective channel into parallel flat-fading narrowband subchannels, where each subband only needs a single-tap equaliser. Therefore the high complexity associated with the long decision-feedback equaliser (DFE) to combat inter-symbol interference (ISI) that may spread up to several hundreds of symbols in single-carrier systems is substantially mitigated.

One of the main challenges of UWA communications is channel estimation. The UWA channel is a time-varying multipath channel with large delay spread, that is, a doubly selective channel. Moreover, the channel exhibits sparsity that can be exploited by the channel estimator. For example, a basis expansion model (BEM) is used to explore the delay-Doppler sparsity in [5]. More practical approaches concentrate on exploiting UWA channel sparsity after some preprocessing, that is, resampling [4, 6]. Since the UWA channel impulse response (CIR) is usually dominated by only a small number of significant paths, most channel coefficients are either zero or nearly zero [1]. Many recent works focus on estimation of sparse channels by making use of the recently emerged compressed sensing (CS) techniques for efficient reconstruction of sparse signals from a few number of linear measurements [7]. Among the many CS algorithms, matching pursuit (MP), orthogonal matching pursuit (OMP) and basis pursuit (BP) have already been applied to channel estimation for UWA OFDM systems [6, 8]. Specifically, in [6], BP is shown to outperform OMP, especially for severe Doppler spread conditions. In [8], three BP algorithms including ℓ_1-LS, YALL1 and SpaRSA are compared. However, the complexity of BP algorithms is much higher than that of the OMP. For UWA communications where the channel estimates have to be frequently updated, applying BP for real-time channel estimation is computationally very expensive. Hence it is important to consider alternative high-performance CS algorithms with lower complexity. Moreover, other avenues should be explored to further improve the channel estimation performance, such as exploiting the channel temporal correlation, and optimising the pilot symbol placement.

In this paper, we consider the application of the homotopy algorithm to sparse UWA channel estimation. We first extend

the original homotopy algorithm that is for real-valued signals to the complex field. We then propose two enhancements to the CS-based UWA channel estimators that exploit the temporal correlation of the channel response. The first one is based on a first-order Gauss–Markov model and uses the previous channel estimate to assist the current one. The other enhancement is to use the recursive least-squares (RLS) algorithm together with the sparse recovery to track the time-varying UWA channel. Another contribution of this work is the pilot placement optimisation where we employ the discrete stochastic approximation technique to perform off-line design of the pilot symbol placement over the OFDM subcarriers to minimise the mean-square channel estimation error.

The notations used in this paper are according to the convention. Symbols for matrices (upper case) and vectors (lower case) are in bold italic face. $(\cdot)^{\mathrm{T}}$, $(\cdot)^{\mathrm{H}}$, $|\cdot|$, $\|\cdot\|_1$ $\|\cdot\|_2$, $\mathbb{C}$, $\mathbb{R}$, diag$\{\cdot\}$, $\boldsymbol{I}_L$, $\boldsymbol{0}_{M\times N}$ and $\mathcal{CN}$ denote the transpose, the conjugate transpose (Hermitian), the absolute value, the ℓ_1-norm, the ℓ_2-norm, the set of complex numbers, the set of real numbers, the diagonal matrix, the $\boldsymbol{L}\times\boldsymbol{L}$ identity matrix, the $\boldsymbol{M}\times\boldsymbol{N}$ zero matrix and the complex Gaussian distribution, respectively. $O(\cdot)$ denotes the order of complexity. $\hat{\phi}$ denotes the estimate of the parameter of interest ϕ.

The remainder of the paper is organised as follows. Section 2 describes the UWA channel model, the pre-processing steps and the sparse channel estimation problem. Section 3 develops the complex homotopy algorithm and two enhancements that exploit channel temporal correlation for the CS-based channel estimator. Section 4 discusses the design of optimised pilot placement. Simulation results are given in Section 5. Finally, Section 6 concludes the paper.

2 System descriptions

We consider the UWA channel that has a time-varying multipath CIR

$$h(\tau, t)=\sum_{i=1}^{S}\alpha_i(t)\delta(\tau-\tau_i(t)) \tag{1}$$

where S, $\alpha_i(t)$ and $\tau_i(t)$ denote the number of total paths, the ith path attenuation and the path delay, respectively [4]. Supposing the dominant Doppler shift is caused by the relative movement between the transmitter and the receiver, all paths have the similar Doppler scaling factor $\varepsilon(t)$ such that

$$\tau_i(t)\simeq\tau_i-\varepsilon(t)\cdot t \tag{2}$$

Let $s(t)=\mathrm{Re}\{x(t)\mathrm{e}^{\mathrm{j}2\pi f_c t}\}$ denote the transmitted signal in passband, where f_c is the carrier frequency and $x(t)$ is the baseband OFDM signal. Then the received passband signal is

$$\begin{aligned} r(t)&=\int_{-\infty}^{\infty}h(\tau, t)s(t-\tau)\mathrm{d}\tau+n(t)\\ &=\mathrm{Re}\left\{\sum_{i=1}^{S}\alpha_i(t)x((1+\varepsilon(t))t-\tau_i(t))\right.\\ &\quad\left.\times\,\mathrm{e}^{\mathrm{j}2\pi f_c((1+\varepsilon(t))t-\tau_i(t))}\right\}+n(t) \end{aligned} \tag{3}$$

where $n(t)$ is the additive Gaussian noise. Within the duration of each OFDM packet, the parameters are treated as constants since the UWA channel coherence time is usually on the order of seconds, whereas each OFDM packet is no more than hundreds of milliseconds [9]. Then (3) can be written as

$$\tilde{r}(t)=\mathrm{Re}\left\{\sum_{i=1}^{S}\alpha_i x((1+\varepsilon)t-\tau_i)\mathrm{e}^{\mathrm{j}2\pi f_c((1+\varepsilon)t-\tau_i)}\right\}+n(t) \tag{4}$$

We notice that it is scaled in time by $1/(1+\varepsilon)$ and the Doppler shift $\mathrm{e}^{\mathrm{j}2\pi f_c\varepsilon t}$ is frequency-dependent. Since the bandwidth is comparable with the carrier frequency in UWA communications, the Doppler shift cannot be regarded as the same for the whole band. Then several approaches have been proposed to mitigate the Doppler effect [10]. Here we adopt a two-step approach [4]. The first step is resampling and the second is the carrier frequency offset (CFO) compensation. Each OFDM packet contains one preamble and one postamble, that is, linear frequency-modulated (LFM) waveforms. By cross-correlating the received signal with the known preamble and postamble, the receiver estimates the length of each OFDM packet and then obtains an estimated Doppler scaling factor $\hat{\varepsilon}$. As the frequency range used for UWA communications is usually tens of thousands of hertz, we can directly sample the received passband signal without down conversion. We resample the received signal in (4) as

$$\begin{aligned} \tilde{y}(t)&=\tilde{r}\left(\frac{t}{1+\hat{\varepsilon}}\right)\\ &=\mathrm{Re}\left\{\sum_{i=1}^{S}\alpha_i x\left(\frac{1+\varepsilon}{1+\hat{\varepsilon}}t-\tau_i\right)\mathrm{e}^{\mathrm{j}2\pi f_c((1+\varepsilon)/(1+\hat{\varepsilon})t-\tau_i)}\right\}+\tilde{n}(t) \end{aligned} \tag{5}$$

where $\tilde{n}(t)$ is the resampled version of $n(t)$. The equivalent received baseband signal is

$$y(t)=\sum_{i=1}^{S}\alpha_i x\left(\frac{1+\varepsilon}{1+\hat{\varepsilon}}t-\tau_i\right)\mathrm{e}^{\mathrm{j}2\pi f_c((\varepsilon-\hat{\varepsilon})/(1+\hat{\varepsilon})t-\tau_i)}+n_B(t) \tag{6}$$

where $n_B(t)$ is the equivalent baseband noise of $\tilde{n}(t)$. We define the residual CFO as

$$f_o=\left(\frac{\varepsilon-\hat{\varepsilon}}{1+\hat{\varepsilon}}\right)f_c \tag{7}$$

which is assumed to be uniformly distributed over the whole bandwidth [4]. Therefore a wideband system is converted into a narrow band system with frequency-independent CFOs. We can then apply the well-developed null-subcarrier-based scheme for CFO compensation [11]. More discussions on the residue CFO are found in [6] where the inter-channel interference (ICI) is also considered.

Owing to the large delay spread of UWA channel, we prefer ZP-OFDM to CP-OFDM [12]. ZP-OFDM saves the transmission energy by zero-padding rather than filling cyclic prefix. Considering a ZP-OFDM system, there are totally N_d subcarriers, among which $N_{\mathrm{p}}(N_{\mathrm{p}}\le N_d)$ are selected as pilots, with positions $k_1<k_2<\cdots<k_{N_p}$ and N_u null subcarriers used for CFO compensation. We denote the transmit pilots and the receive pilots as $X(k_1)$, $X(k_2),\ldots,X(k_{N_{\mathrm{p}}})$ and $Y(k_1)$, $Y(k_2),\ldots,Y(k_{N_{\mathrm{p}}})$, respectively. Considering a UWA channel whose CIR length

doi: 10.1049/iet-spr.2010.0347

after sampling is L, we have

$$\boldsymbol{y} = \boldsymbol{A}\boldsymbol{h} + \boldsymbol{\eta} \tag{8}$$

where $\boldsymbol{A} = \boldsymbol{X}\boldsymbol{F}_{N_p \times L}$, $\boldsymbol{X} = \text{diag}\{X(k_1), X(k_2), \ldots, X(k_{N_p})\}$, $\boldsymbol{F}_{N_p \times L}$ is a submatrix selected by the row indices $[k_1, k_2, \ldots, k_{N_p}]$ and column indices $[0, 1, \ldots, L-1]$ from the standard $\boldsymbol{N}_d \times \boldsymbol{N}_d$ Fourier matrix, $\boldsymbol{y} = [Y(k_1), Y(k_2), \ldots, Y(k_{N_p})]^{\mathrm{T}}$, $\boldsymbol{h} = [h(0), h(1), \ldots, h(L-1)]^{\mathrm{T}}$, $\boldsymbol{\eta} = [\eta(0), \eta(1), \ldots, \eta(N_p - 1)]^{\mathrm{T}} \sim \mathcal{CN}(\boldsymbol{0}, \sigma_\eta^2 \boldsymbol{I}_{N_p})$.

Since the system sampling interval is much smaller compared to the channel delay spread, most channel coefficients are either zero or nearly zero, which means that $\boldsymbol{h}$ is a sparse vector. If $\boldsymbol{A}$ has more rows than columns, that is, $N_p > L$, then (8) is a standard least-squares (LS) problem. However, we are more interested in the sparse case with $N_p < L$, that is, the number of pilots is less than the number of channel coefficients.

3 Channel estimation based on sparse recovery

Suppose that the channel vector $\boldsymbol{h} \in \mathbb{R}^L$ has S non-zero components, with $S \ll L$. Then we can recover $\boldsymbol{h}$ from (8) by solving the following ℓ_0-norm minimisation problem

$$\min_{\boldsymbol{h}} ||\boldsymbol{h}||_0 \text{ s.t. } ||\boldsymbol{y} - \boldsymbol{A}\boldsymbol{h}||_2 \le \sigma_\eta \tag{9}$$

where $\|\boldsymbol{h}\|_0$ counts the number of non-zero elements of $\boldsymbol{h}$. This is an NP-hard combinatorial problem. However, it can be replaced by the following ℓ_1-norm optimisation problem [13]

$$\min_{\boldsymbol{h}} ||\boldsymbol{h}||_1 \text{ s.t. } ||\boldsymbol{y} - \boldsymbol{A}\boldsymbol{h}||_2 \le \sigma_\eta \tag{10}$$

Currently, methods for solving (10) can be roughly divided into two classes, including convex optimisation algorithms and greedy algorithms. The convex optimisation algorithms include BP algorithms such as ℓ_1-LS [14], YALL1 [8], SpaRSA [15] and other optimisation solvers. However, these methods usually have high computational complexities. The greedy algorithms sequentially search for the locally optimal solutions and they include methods such as MP, OMP [16], CoSaMP [17], subspace pursuit (SP) [18] and homotopy [19]. Since in UWA communications frequent channel estimation is required, low-complexity greedy algorithms are usually preferred. Among the greedy algorithms, CoSaMP and SP require explicit knowledge of the sparsity. MP and OMP have been extensively studied with their variants [20, 21]; and it is known that they may not always achieve satisfactory performance [22]. In [19], it is shown that homotopy has the same order of complexity as OMP, while its sparse recovery performance is as good as that obtained from convex optimisation. So in the following, we will apply the homotopy algorithm to UWA channel estimation.

3.1 Complex homotopy

Real-valued homotopy is proposed in [19] as a least angle regression (LARS) algorithm with the least absolute shrinkage and selection operator (LASSO) modifications. It constructs a sparse solution by iteratively selecting the matrix columns and eventually forming a linear combination of them closest to the signal. From the geometric perspective, homotopy always takes a step along an equiangular direction that has equal angles with all vectors selected so far. Since the UWA channel is a sparse complex-valued vector, we will first extend the homotopy algorithm from the real field to the complex field. Note that unlike the MP and OMP algorithms, for which the extension to the complex field is straightforward by simply replacing the transpose operator by the Hermitian operator, the extension of homotopy to the complex field is more involved.

Consider the following unconstrained optimisation problem

$$\min_{\boldsymbol{h} \in \mathbb{C}^L} ||\boldsymbol{y} - \boldsymbol{A}\boldsymbol{h}||_2^2/2 + \lambda ||\boldsymbol{h}||_1 \tag{11}$$

Homotopy starts with a large $\lambda \in \mathbb{R}$ and $\boldsymbol{h} = 0$, and terminates when $\lambda \to 0$ with $\boldsymbol{h}$ converging to the solution to the 'noiseless' sparse recovery problem

$$\min_{\boldsymbol{h}} ||\boldsymbol{h}||_1 \text{ s.t. } \boldsymbol{y} = \boldsymbol{A}\boldsymbol{h} \tag{12}$$

For the 'noisy' sparse recovery problem considered in this paper, the stopping condition $\lambda \to 0$ should be changed to

$$||\boldsymbol{y} - \boldsymbol{A}\boldsymbol{h}||_2 \le \sigma_\eta \tag{13}$$

Denote

$$f_\lambda(\boldsymbol{h}) = ||\boldsymbol{y} - \boldsymbol{A}\boldsymbol{h}||_2^2/2 + \lambda ||\boldsymbol{h}||_1 \tag{14}$$

A necessary condition for $\boldsymbol{h}$ to be a minimiser of $f_\lambda(\boldsymbol{h})$ is that the subdifferential of $f_\lambda(\boldsymbol{h})$ to be zero, that is

$$\partial f_\lambda(\boldsymbol{h}) = -\boldsymbol{A}^H(\boldsymbol{y} - \boldsymbol{A}\boldsymbol{h}) + \lambda \partial ||\boldsymbol{h}||_1 = \boldsymbol{0} \tag{15}$$

where the subdifferential $\partial\|\boldsymbol{h}\|_1$ of $\|\boldsymbol{h}\|_1$ is given by

$$\partial ||\boldsymbol{h}||_1 = \left\{ \boldsymbol{w} \in \mathbb{C}^L \left| \begin{array}{ll} w(i) = \dfrac{h(i)}{|h(i)|}, & h(i) \neq 0 \\ w(i) = \{v \in \mathbb{C}, |v| \le 1\}, & h(i) = 0 \end{array} \right. \right\} \tag{16}$$

where $h(i)$ and $w(i)$ denote the ith component of $\boldsymbol{h}$ and $\boldsymbol{w}$, respectively. It is observed that (16) is different from that of the real-valued homotopy algorithm in [19]. Let $T = \{i: h(i) \neq 0\}$ denote the support of $\boldsymbol{h}$ and $\boldsymbol{c} = \boldsymbol{A}^{\mathrm{H}}(\boldsymbol{y} - \boldsymbol{A}\boldsymbol{h})$ denote the correlations between the dictionary matrix and the residue. Then the condition $\partial f_\lambda(\boldsymbol{h}) = \boldsymbol{0}$ can be written as equivalent to the following two conditions

$$\left\{ \begin{array}{l} \boldsymbol{c}(T) = \lambda \cdot \dfrac{\boldsymbol{h}(T)}{|\boldsymbol{h}(T)|} \\ |\boldsymbol{c}(T^{\mathrm{c}})| \le \lambda \end{array} \right. \tag{17}$$

where T^{c}, $\boldsymbol{c}(T)$ and $\boldsymbol{h}(T)$ denote the complement of T, the correlations on the support T and the channel parameters on T, respectively. Homotopy traces a solution path by maintaining these two conditions. The complex-valued homotopy algorithm is summarised in Fig. 1. The main difference from the real-valued homotopy algorithm lies in Step 2. The step size $\gamma_l \in \mathbb{C}$ and the moving direction $\boldsymbol{d}_l \in \mathbb{C}^L$ have to be decided so that a new solution at the $(l+1)$th step is obtained as

$$\boldsymbol{h}_{l+1} = \boldsymbol{h}_l + \gamma_l \boldsymbol{d}_l \tag{18}$$

Algorithm 1

Step 0: Initialisation

$\boldsymbol{c}_0 = \boldsymbol{A}^H\boldsymbol{y}$, $\lambda_0 = \max_{i\in\{1,\ldots,L\}}|c_0(i)|$, $I_0 = \arg\max_{i\in\{1,\ldots,L\}}|c_0(i)|$;
set both $\boldsymbol{h}_l \in \mathbb{C}^L$ and $\boldsymbol{d}_l \in \mathbb{C}^L$ to be zero vectors.

FOR $l = 0, 1, \ldots,$

Step 1: Find the direction
solve $\boldsymbol{A}^H(I_l)\boldsymbol{A}(I_l)\boldsymbol{d}_l(I_l) = \boldsymbol{c}_l(I_l)/\lambda_l$ and get $\boldsymbol{d}_l$;
set the components of $\boldsymbol{d}_l$ outside I_l to be zero.

Step 2: Decide the step size
define $\boldsymbol{c}_l(I_l^c) = \boldsymbol{A}^H(I_l^c)(\boldsymbol{y} - \boldsymbol{A}\boldsymbol{h}_l)$, $d\boldsymbol{\mu}(I_l^c) = \boldsymbol{A}^H(I_l^c)\boldsymbol{A}\boldsymbol{d}_l$;
FOR $i \in I_l^c$
define $c_R = \mathrm{Re}\{c_l(i)\}$, $c_I = \mathrm{Im}\{c_l(i)\}$, $\mu_R = \mathrm{Re}\{d\mu(i)\}$ and $\mu_I = \mathrm{Im}\{d\mu(i)\}$;
solve $(\mu_R^2 + \mu_I^2 - 1)\gamma_l^2 + 2(\lambda_l - c_R\mu_R - c_I\mu_I)\gamma_l + c_R^2 + c_I^2 - \lambda_l^2 = 0$ and get two roots for γ_l, from which we select the smaller one, denote as $\gamma(i)$;
EndFOR (i)
define $\gamma_l^+ = \min\{\gamma(i),\ i \in I_l^c\}$ and $I_s = \{i \in I_l \mid \mathrm{Re}\{h_l(i)\}/\mathrm{Re}\{d_l(i)\} = \mathrm{Im}\{h_l(i)\}/\mathrm{Im}\{d_l(i)\}\}$;
if $I_s \neq \varnothing$
$\gamma_l^- = \min\{-\mathrm{Re}\{h_l(i)\}/\mathrm{Re}\{d_l(i)\},\ i \in I_s\}$, $\gamma_l = \min\{\gamma_l^+, \gamma_l^-\}$
else
$\gamma_l = \gamma_l^+$
end if
set $\boldsymbol{h}_{l+1} = \boldsymbol{h}_l + \gamma_l\boldsymbol{d}_l$ and $\lambda_{l+1} = \lambda_l - \gamma_l$;
set $\boldsymbol{c}_{l+1}(I_l) = (\lambda_l - \gamma_l)\boldsymbol{h}_l(I_l)/|\boldsymbol{h}_l(I_l)|$ and $\boldsymbol{c}_{l+1}(I_l^c) = \boldsymbol{c}_l(I_l^c) - \gamma_l d\boldsymbol{\mu}(I_l^c)$.

Step 3: Update the selection
if $\gamma_l = \gamma_l^+$
add a new column indicated by γ_l^+ into I_l and get I_{l+1}
else
delete the column indicated by γ_l^- from I_l and get I_{l+1}
end if

Step 4: Check the stop condition
if $\|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{h}_{l+1}\|_2 \leq \sigma_\eta$, quit the iteration;
otherwise continue ...

EndFOR (l)

Fig. 1 *Complex homotopy*

where $\boldsymbol{h}_l \in \mathbb{C}^L$ is initialised to be a zero vector and converges to the solution of (11). Since the current selected columns always follow (17), we have

$$\boldsymbol{c}_{l+1}(I_l) = (\lambda_l - \gamma_l)\frac{\boldsymbol{h}_l(I_l)}{|\boldsymbol{h}_l(I_l)|} \tag{19}$$

On the other hand, we want to find a new column from I_l^c at the $(l+1)$th step. Since

$$\boldsymbol{c}_{l+1} = \boldsymbol{A}^H(\boldsymbol{y} - \boldsymbol{A}\boldsymbol{h}_{l+1}) \tag{20}$$

plugging (18) into (20), we obtain

$$\boldsymbol{c}_{l+1} = \boldsymbol{A}^H(\boldsymbol{y} - \boldsymbol{A}\boldsymbol{h}_l) - \gamma_l\boldsymbol{A}^H\boldsymbol{A}\boldsymbol{d}_l \tag{21}$$

Then we define

$$\boldsymbol{c}_l(I_l^c) = \boldsymbol{A}^H(I_l^c)(\boldsymbol{y} - \boldsymbol{A}\boldsymbol{h}_l)$$

$$d\boldsymbol{\mu}(I_l^c) = \boldsymbol{A}^H(I_l^c)\boldsymbol{A}\boldsymbol{d}_l$$

so that

$$\boldsymbol{c}_{l+1}(I_l^c) = \boldsymbol{c}_l(I_l^c) - \gamma_l d\boldsymbol{\mu}(I_l^c) \tag{22}$$

Once a column outside I_l appears to have the same projections with the current residue as the columns inside I_l, which in fact occurs when (19) happens to equal (22) in amplitude as

$$|\boldsymbol{c}_l(i) - \gamma_l d\boldsymbol{\mu}(i)| = \lambda_l - \gamma_l, \quad i \in I_l^c \tag{23}$$

we consider it to be one possible candidate for the next selection because the residue can decline equally in this direction. It is the reason for solving the quadratic equation in Step 2.

Moreover, the selection of the breakpoint in the solution path for the complex homotopy should also be extended. The breakpoint occurs when one component of $\boldsymbol{h}_l(I_l)$ crosses zero, which means both the real and imaginary parts equal to zeros. If

$$I_s = \left\{ i \in I_l \left| \frac{\mathrm{Re}\{h_l(i)\}}{\mathrm{Re}\{d_l(i)\}} = \frac{\mathrm{Im}\{h_l(i)\}}{\mathrm{Im}\{d_l(i)\}} \right. \right\} \neq \varnothing$$

doi: 10.1049/iet-spr.2010.0347

then

$$\gamma_l^- = \min_{i \in I_s} \left\{ -\frac{\mathrm{Re}\{h_l(i)\}}{\mathrm{Re}\{d_l(i)\}} \right\} \tag{24}$$

Finally, the step size γ_l is determined to be the minimum between γ_l^+ and γ_l^-, if γ_l^- exists.

The above extension to the complex field can be verified by forcing $c_I = 0$ and $\mu_I = 0$ in the quadratic equation of Step 2 of Fig. 1, which then reduces to the real-field form of the homotopy algorithm [19]. Unlike the OMP method that removes the entire projection at each step, homotopy only removes part of it, which can be thought of as a moderate greedy algorithm. Moreover, homotopy allows the columns to enter as well as to leave the current selection, which makes it more powerful than OMP. In terms of the computational complexity, homotopy is roughly on the same order as OMP with $\mathcal{O}(N_p^2 L)$ while the complexity of the convex optimisation algorithms is $\mathcal{O}(L^3)$.

3.2 Incorporating channel temporal correlation

The basic idea is to use the channel estimate $\boldsymbol{h}^{(m-1)}$ from the previous time slot $(m-1)$ to assist the current channel estimation $\boldsymbol{h}^{(m)}$. Since the UWA channel is temporally correlated, we model it using a simple first-order Gauss–Markov process, given by

$$\boldsymbol{h}^{(m)} = \kappa \boldsymbol{h}^{(m-1)} + \boldsymbol{v}^{(m)} \tag{25}$$

$$\text{with } \kappa = J_0(2\pi f_d T_s) \tag{26}$$

where J_0, f_d and T_s denote the zeroth order Bessel function of the first kind, the maximum Doppler shift and the symbol period, respectively. $\boldsymbol{v}^{(m)} \sim \mathcal{CN}(\boldsymbol{0}, \sigma_v^2 \boldsymbol{I}_L)$ is a Gaussian noise term independent of $\boldsymbol{h}^{(m-1)}$. We rewrite (8) as

$$\boldsymbol{y}^{(m)} = \boldsymbol{A}\boldsymbol{h}^{(m)} + \boldsymbol{\eta}^{(m)} \tag{27}$$

Combining (25) and (27) using matrix notation, we have

$$\begin{bmatrix} \boldsymbol{y}^{(m)} \\ \kappa \boldsymbol{h}^{(m-1)} \end{bmatrix} = \begin{bmatrix} \boldsymbol{A} \\ \boldsymbol{I}_L \end{bmatrix} \boldsymbol{h}^{(m)} + \begin{bmatrix} \boldsymbol{\eta}^{(m)} \\ -\boldsymbol{v}^{(m)} \end{bmatrix} \tag{28}$$

It is obvious that we can still apply the sparse recovery algorithms to solve for $\boldsymbol{h}^{(m)}$ with the following two stopping conditions

$$||\boldsymbol{y}^{(m)} - \boldsymbol{A}\boldsymbol{h}^{(m)}||_2 \le \sigma_\eta \tag{29}$$

and

$$||\boldsymbol{h}^{(m)} - \kappa \boldsymbol{h}^{(m-1)}||_2 \le \sigma_v \tag{30}$$

At low signal-to-noise ratio (SNR) where σ_η is comparable to σ_v, (28) can be regarded as having additional observations compared to (27), where improved channel estimation performance is expected.

OFDM transforms the frequency-selective UWA channel into parallel flat-fading subchannels. Each subchannel may experience a different Doppler shift because of the wideband nature of the UWA channel. We therefore rewrite the constant κ in (26) as a diagonal matrix

$$\boldsymbol{\kappa} = \boldsymbol{F}_L^{-1} \cdot \mathrm{diag}\{J_0(2\pi f_{d_1} T_s), J_0(2\pi f_{d_2} T_s), \ldots, J_0(2\pi f_{d_L} T_s)\} \cdot \boldsymbol{F}_L \tag{31}$$

where $f_{d_1}, f_{d_2}, \ldots, f_{d_L}$ and $\boldsymbol{F}_L$ are the Doppler shift for each subchannel and an $\boldsymbol{L} \times \boldsymbol{L}$ Fourier matrix, respectively. Similarly as before, we can apply sparse recovery algorithms to solve for $\boldsymbol{h}^{(m)}$.

3.3 Channel tracking via RLS

The RLS algorithm is widely used for tracking time-varying processes. Here we consider further improving the UWA channel estimation performance by integrating the RLS algorithm with the sparse recovery algorithm. In particular, we modify the objective function (11) as

$$\min_{\boldsymbol{h}^{(m)} \in \mathbb{C}^L} \sum_{i=i_0}^{m} \beta^{m-i} ||\boldsymbol{y}^{(i)} - \boldsymbol{A}\boldsymbol{h}^{(m)}||_2^2/2 + \lambda^{(m)} ||\boldsymbol{h}^{(m)}||_1 \tag{32}$$

where i_0 and m are the starting and ending points of the weighted window, respectively. m is also the current time where the channel estimate is to be obtained. $\beta \in [0, 1)$ is the forgetting factor chosen to tradeoff between the convergence speed and the tracking performance [23]. Correspondingly, (14) and (15) are modified as

$$f_{\lambda^{(m)}}(\boldsymbol{h}^{(m)}) = \sum_{i=i_0}^{m} \beta^{m-i} ||\boldsymbol{y}^{(i)} - \boldsymbol{A}\boldsymbol{h}^{(m)}||_2^2/2 + \lambda^{(m)} ||\boldsymbol{h}^{(m)}||_1 \tag{33}$$

and

$$\partial f_{\lambda^{(m)}}(\boldsymbol{h}^{(m)}) = -\boldsymbol{A}^H \left(\sum_{i=i_0}^{m} \beta^{m-i} \boldsymbol{y}^{(i)} - \boldsymbol{A} \sum_{i=i_0}^{m} \beta^{m-i} \boldsymbol{h}^{(m)} \right) + \lambda^{(m)} \partial ||\boldsymbol{h}^{(m)}||_1 \tag{34}$$

We define

$$\boldsymbol{c}^{(m)} = \boldsymbol{A}^H \left(\frac{1-\beta}{1-\beta^m} \sum_{i=i_0}^{m} \beta^{m-i} \boldsymbol{y}^{(i)} - \boldsymbol{A}\boldsymbol{h}^{(m)} \right) \tag{35}$$

Then (17) is modified as

$$\begin{cases} \boldsymbol{c}^{(m)}(T) = \dfrac{(1-\beta)\lambda^{(m)}}{1-\beta^m} \cdot \dfrac{\boldsymbol{h}^{(m)}(T)}{|\boldsymbol{h}^{(m)}(T)|} \\ |\boldsymbol{c}^{(m)}(T^c)| \le \dfrac{(1-\beta)\lambda^{(m)}}{1-\beta^m} \end{cases} \tag{36}$$

We can then apply the complex homotopy algorithm by substituting $\boldsymbol{y}$ and λ into (11) with

$$\boldsymbol{z}^{(m)} = \frac{1-\beta}{1-\beta^m} \sum_{i=i_0}^{m} \beta^{m-i} \boldsymbol{y}^{(i)} \tag{37}$$

and

$$\xi = \frac{(1-\beta)\lambda^{(m)}}{1-\beta^m} \tag{38}$$

respectively. The iterative procedure for updating (37) becomes

$$\boldsymbol{z}^{(m)} = \frac{\beta-\beta^m}{1-\beta^m}\boldsymbol{z}^{(m-1)} + \frac{1-\beta}{1-\beta^m}\boldsymbol{y}^{(m)} \tag{39}$$

Similarly, we can integrate the RLS algorithm with other sparse recovery algorithms such as OMP and YALL1. The performance improvement by the RLS weighted sparse recovery is determined by β and i_0. For rapid time-varying UWA channels, we set β small and control the weighted window length by setting i_0 close to m; while for relatively slowly time-varying UWA channels, we choose β close to 1 and increase $m - i_0$.

4 Optimised pilot placement

Although the best pilot placement for the LS channel estimation in OFDM systems is equipowered, equispaced and phase shift orthogonal [24], there is no general theory on the optimised pilot placement for channel estimation using sparse recovery algorithms. It is known that under the noiseless condition, $\boldsymbol{h}$ in (8) can be recovered from $\boldsymbol{y}$ with high accuracy when $\boldsymbol{A}$ satisfies the so-called RIP property [25]. However, there is no known method to test in polynomial time whether a given matrix satisfies RIP. Here we propose to obtain the optimised pilot placement over N_{p} subcarriers through off-line training using either the statistical channel model or the actual UWA channel measurement data. Considering selecting N_{p} pilot positions from a total of N_d subcarriers, there are $\binom{N_d}{N_p}$ possible pilot placements. For example, if $N_p = 12$ and $N_d = 256$, we have $\binom{256}{12} = 1.27 \times 10^{21}$ possible pilot placements. Therefore it is computationally prohibitive to exhaustively search for the optimised placement. We propose to use the discrete stochastic approximation method to sequentially search for a near-optimal pilot placement, which typically exhibits a fast convergence [26].

We define the normalised mean-squared error (MSE) corresponding to a pilot placement $\boldsymbol{p}$ and channel realisation $\boldsymbol{h}$ for a specific channel estimation algorithm as

$$f(\boldsymbol{p}, \boldsymbol{h}, \hat{\boldsymbol{h}}) = \frac{E\{\|\boldsymbol{h}-\hat{\boldsymbol{h}}\|_2^2\}}{||\boldsymbol{h}||_2^2} \tag{40}$$

where the expectation is with respect to the channel noise, $\boldsymbol{p} = \{k_1, k_2, \ldots, k_{N_p}\}$, $\boldsymbol{h}$ and $\hat{\boldsymbol{h}}$ denote the pilot placement, the channel realisation and the estimated channel, respectively. Thus the average MSE over all possible channel realisations is

$$g(\boldsymbol{p}) = E_{\boldsymbol{h}}\{f(\boldsymbol{p}, \boldsymbol{h}, \hat{\boldsymbol{h}})\} \tag{41}$$

Our objective is then to find the optimised pilot placement with the minimum average MSE for a specific sparse recovery algorithm, that is

$$\min_{\boldsymbol{p}} g(\boldsymbol{p}) \tag{42}$$

The pilot placement optimisation procedure based on the discrete stochastic approximation algorithm is summarised in Fig. 2. Starting from a random pilot placement $\boldsymbol{p}$, we run the homotopy algorithm for different channel realisations to obtain an averaged MSE. We then obtain another pilot placement by randomly changing one pilot position of the current $\boldsymbol{p}$. We run the homotopy algorithm on the new pilot placement and compare the average MSE with the previous one, and make a choice for the next move. The above steps are repeated for a number of iterations and at each iteration, the frequency of the picked pilot placement is updated. The final choice of the pilot placement is the one with the highest occurring frequency. We perform the same procedure to optimise the pilot placements for other sparse recovery algorithms, such as OMP and YALL1.

Although in theory Fig. 2 requires a memory size of $\boldsymbol{p}$ to store the occupancy probabilities of all possible pilot placements, in the actual implementation, the memory requirement is much smaller. Noticing that most pilot placement will not occur at all during the M iterations of the algorithm, we can use a memory size of $\boldsymbol{N}_x = MN_p$. In particular, an $\boldsymbol{N}_x \times 1$ vector $\pi[m] = [\pi[m, 1], \ldots, \pi[m, N_x]]^{\mathrm{T}}$ with elements $\pi[m, i] \in [0, 1]$ and $\Sigma_i \pi[m, i] = 1$ is initialised. We use a $\boldsymbol{N}_x \times N_p$ matrix $\boldsymbol{P}$ to

Algorithm 2

Step 0: Initialisation
$n \Leftarrow 0, k \Leftarrow 0$;
randomly generate one pilot placement $\boldsymbol{p}_0$ and set $\tilde{\boldsymbol{p}}_0 = \boldsymbol{p}_0$;
set $\boldsymbol{P} = \boldsymbol{0}_{N_x \times N_p}$, $\boldsymbol{P}[1] = \boldsymbol{p}_0$ and $\hat{\boldsymbol{p}}_0 = \boldsymbol{p}_0$;
set $\pi[0, \boldsymbol{p}_0] = 1$, $\pi[0, \boldsymbol{p}] = 0$ for all $\boldsymbol{p} \neq \boldsymbol{p}_0$.

FOR $n = 0, 1, \ldots, M-1$
FOR $k = 0, 1, \ldots, N_p - 1$

Step 1: Sampling and evaluation
$m = n * N_p + k$;
given $\boldsymbol{p}_m$ and $g(\boldsymbol{p}_m)$ from the last step, generate another $\tilde{\boldsymbol{p}}_m \backslash \boldsymbol{p}_m$ uniformly,
where $\tilde{\boldsymbol{p}}_m = \boldsymbol{p}_m$ except for the k-th pilot position;
calculate $g(\tilde{\boldsymbol{p}}_m)$.

Step 2: Acceptance
if $g(\tilde{\boldsymbol{p}}_m) < g(\boldsymbol{p}_m)$
$\boldsymbol{p}_{m+1} = \tilde{\boldsymbol{p}}_m$
else
$\boldsymbol{p}_{m+1} = \boldsymbol{p}_m$
end if

Step 3: Updating state occupation probabilities
searching $\boldsymbol{p}_{m+1}$ in $\boldsymbol{P}$;
if found
$q_m =$ the found row index in $\boldsymbol{P}$
else
$q_m = m + 1$, store $\boldsymbol{p}_{m+1}$ into $\boldsymbol{P}$
end if
$\pi[m+1] = \pi[m] + (\boldsymbol{r}[q_m] - \pi[m])/(m+1)$;

Step 4: Selection
if $\pi[m+1, \boldsymbol{p}_{m+1}] > \pi[m+1, \hat{\boldsymbol{p}}_m]$
$\hat{\boldsymbol{p}}_{m+1} = \boldsymbol{p}_{m+1}$
else
$\hat{\boldsymbol{p}}_{m+1} = \hat{\boldsymbol{p}}_m$
end if

EndFOR (k)
EndFOR (n)

Fig. 2 *Pilot placement optimisation*

doi: 10.1049/iet-spr.2010.0347

Table 1 Parameters of the simulation

number of total subcarriers	$N_d = 512$
number of null subcarriers	$N_u = 110$
number of pilot subcarriers	$N_p = 20$
length of zero padding	$N_G = 64$
number of multipaths	$S = 5$
length of CIR	$L = 50$
carrier frequency	$f_c = 24$ kHz
signal bandwidth	$B = 4$ kHz
Doppler spread at f_c	2.7 Hz

store each selected pilot placement. Once entering Step 3, we first search $\boldsymbol{P}$ to check whether $\boldsymbol{p}_{m+1}$ exists. If so, we store the found row index of $\boldsymbol{P}$ into q_m. Otherwise, we store $\boldsymbol{p}_{m+1}$ into $\boldsymbol{P}[m]$ and set $q_m = m + 1$. Then we update $\pi[m]$ where $\boldsymbol{r}[q_m]$ is defined as a $N_x \times 1$ zero vector except with 1 for its q_m-th element. In this way we guarantee the component of $\boldsymbol{P}[m]$ for the same pilot placement is unique.

5 Simulation results

Three BP algorithms including ℓ_1-LS, SpaRSA and YALL1 are compared in [8] where YALL1 is concluded to be the best. So in this section, we compare homotopy with YALL1 and OMP. We use the following sparse multipath channel model

$$c(\tau) = \sum_{i=1}^{S} a_i \delta(\tau - \tau_i)$$

to obtain channel realisations. $\{a_i\} \sim \mathcal{CN}(\mathbf{0}, \mathrm{e}^{-b\tau_i}\boldsymbol{I}_S)$. $b = 1/16$ is the exponential power delay profile and τ_i is the delay spread for the ith path [27]. In practice, we may replace the simulated channel realisations with the real UWA channel measurement data, if they are available. It is verified in [6] that the simulations usually give the same performance trend as the real UWA experiments.

The parameters used in our simulations are listed in Table 1. The bandwidth of 4 kHz is centred around $f_c = 24$ kHz and divided into 512 OFDM subcarriers, among which 20 and 110 are used for pilots and null subcarriers, respectively. The OFDM symbol length is 128.2 ms, which is the reciprocal of the subcarrier spacing. The length of the guard interval is 16 ms or equivalently $N_G = 64$ after sampling. A five-path channel with the maximum channel delay spread $L = 50$ is considered. According to [6], mild Doppler spread with the velocity standard deviation $\sigma_v = 0.1$ is assumed, which results in the Doppler spread at f_c to be about 2.7 Hz.

5.1 Random pilot placement

We first use random pilot placement and compare homotopy, OMP and YALL1 with RLS- and GM-enhanced counterparts as shown in Fig. 3. The MSE of homotopy is better than that of OMP, especially for SNR >25 dB. Homotopy and YALL1 are not much different. We combine the advantages of the first-order GM model and the RLS-weighted sparse recovery, which is denoted as RLS CS with GM. We set the forgetting factor $\beta = 0.992$ and weighted window length $m - i_0 = 12$. It is observed from Fig. 3 that RLS homotopy with GM performs better than RLS OMP with GM. RLS YALL1 is inferior to the above two since the

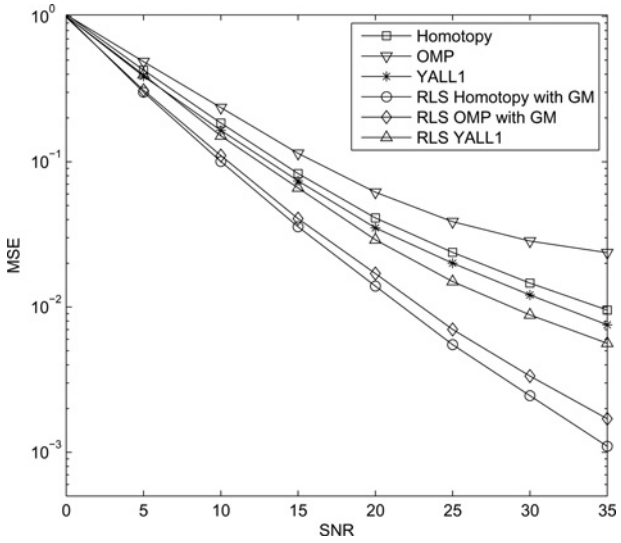

Fig. 3 *Performance of different CS algorithms with GM and RLS*

Table 2 Running times of different CS algorithms

Algorithm type	CPU time (in s)	
	Random pilot	Optimised pilot
homotopy	0.563	0.561
OMP	0.526	0.453
YALL1	2.313	2.311
RLS homotopy with GM	0.584	0.583
RLS OMP with GM	0.475	0.474
RLS YALL1	2.328	2.328

GM model cannot be incorporated by the YALL1 solver. The complexities of these algorithms in terms of the CPU running time are compared in Table 2 where SNR is fixed to be 30 dB. The experiments are performed using MATLAB v7.9 (R2009b) running on a Lenovo laptop with an Intel Core 2 Duo CPU at 2.5 GHz and 2GB of memory. Homotopy uses about 0.563 s, which is similar to 0.526 s of OMP and much lower than 2.313 s of YALL1. RLS homotopy with GM is about 0.02 s higher than homotopy. Since OMP cannot always guarantee successful sparse recovery and sometimes it has to run out of all matrix columns, the time cost of OMP is even 0.05 s higher than that of RLS OMP with GM.

5.2 Optimised pilot placement

We use the simulated UWA channel data to off-line train the pilot placement for homotopy, OMP and YALL1, each running 120 iterations with Fig. 2. We also compare it with the exhaustive search of correspondingly 2400 iterations. As shown in Fig. 4, the algorithm converges to a pilot placement with low MSE very fast. For all three CS algorithms, the algorithm picks a pilot placement that outperforms the best one out of 2400 exhaustive searches in no more than 100 iterations. After 2400 iterations, we obtain an optimised pilot placement for each of homotopy, OMP and YALL1. From them, we select the CS algorithm and corresponding optimised pilot placement with the smallest MSE. According to Fig. 4, we select homotopy and corresponding optimised pilot placement to be (56, 66, 78, 99, 123, 163, 174, 184, 195, 219, 243,

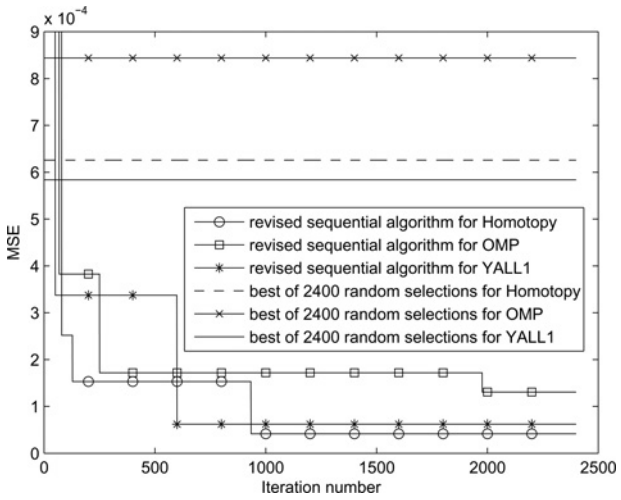

Fig. 4 *Convergence of pilot placement optimisation algorithm*

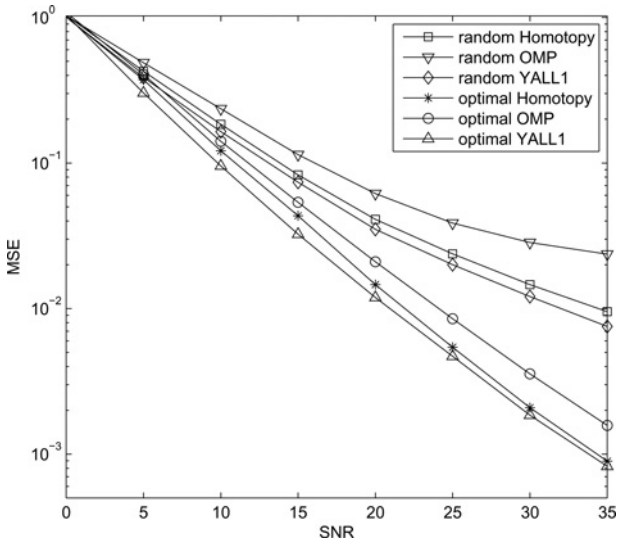

Fig. 5 *Performance improvement with optimised pilot placement*

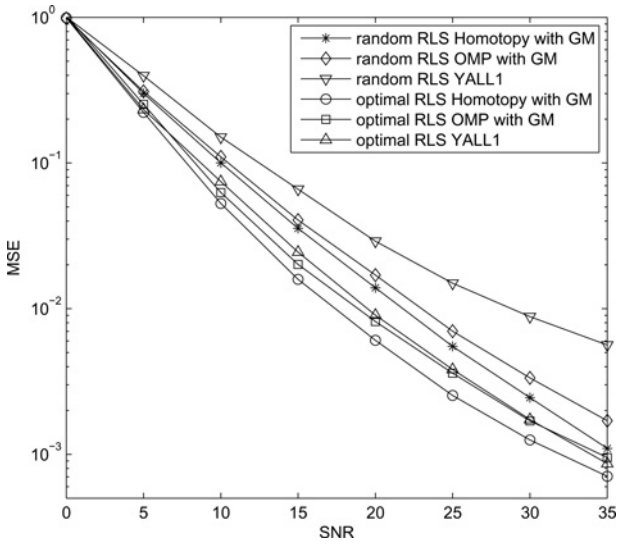

Fig. 6 *Performance improvement with the RLS, GM and optimised pilot placement*

272, 290, 302, 312, 333, 344, 370, 417, 452). In Fig. 5, we compare the random pilot and optimised pilot for different CS algorithms. Since the MSE is averaged over 1000 channel realisations, we cannot guarantee the 1000 sparse recovery to be always successful. The optimised pilot placement shows its ability to kick out the case that the CS algorithms fail or obtain large MSE, which leads to about 6 dB improvement at SNR = 30 dB. In Fig. 6, we further compare RLS- and GM-modified CS algorithms with the random pilot and optimised pilot, respectively. Homotopy still proves to be the best. On the other hand, we also simulate the traditional cubic-spline-interpolation-based LS channel estimation. The same number of equal-spaced pilots is employed. However, we always obtain the MSE larger than 10^3, which is out of the scale of Fig. 6.

6 Conclusions

We have treated the UWA OFDM channel estimation based on sparse recovery using the complex version of the homotopy algorithm. In addition, we have proposed two enhancements to exploit the temporal correlation of the UWA channel, including a first-order Gauss–Markov model and the use of the RLS algorithm for tracking the time-varying UWA channel. Moreover, we have proposed a pilot placement optimisation scheme to further improve the channel estimation performance. In practice, if actually UWA channel measurement data is available, then we can use them to obtain the optimised pilot placement over the OFDM subcarriers. Simulation results show that the proposed homotopy algorithm together with the Gauss–Markov modelling and RLS tracking, as well as pilot placement optimisation, offers substantial performance improvement compared with the state-of-the-art UWA channel estimators.

7 References

1 Stojanovic, M., Preisig, J.: 'Underwater acoustic communication channels: propagation models and statistical characterization', *IEEE Commun. Mag.*, 2009, **47**, (1), pp. 84–89
2 Kilfoyle, D.B., Baggeroer, A.B.: 'The state of the art in underwater acoustic telemetry', *IEEE J. Ocean. Eng.*, 2000, **25**, (1), pp. 4–27
3 Stojanovic, M., Catipovic, J., Proakis, J.: 'Phase coherent digital communications for underwater acoustic channels', *IEEE J. Ocean. Eng.*, 1994, **19**, (1), pp. 100–111
4 Li, B., Zhou, S., Stojanovic, M., Freitag, L., Willett, P.: 'Multicarrier communication over underwater acoustic channels with nonuniform Doppler shifts', *IEEE J. Ocean. Eng.*, 2008, **33**, (2), pp. 198–209
5 Taubock, G., Hlawatsch, F., Eiwen, D., Rauhut, H.: 'Compressive estimation of doubly selective channels in multicarrier systems: leakage effects and sparsity-enhancing processing', *IEEE J. Sel. Topics Signal Proc.*, 2010, **4**, (2), pp. 255–271
6 Berger, C.R., Zhou, S., Preisig, J.C., Willett, P.: 'Sparse channel estimation for multicarrier underwater acoustic communication: from subspace methods to compressed sensing', *IEEE Trans. Signal. Proc.*, 2010, **58**, (3), pp. 1708–1721
7 Bajwa, W.U., Haupt, J., Sayeed, A.M., Nowak, R.: 'Compressed channel sensing: a new approach to estimating sparse multipath channels', *Proc. IEEE*, 2010, **98**, (6), pp. 1058–1076
8 Huang, J.Z., Berger, C.R., Zhou, S., Huang, J.: 'Comparison of basis pursuit algorithms for sparse channel estimation in underwater acoustic OFDM'. Proc. MTS/IEEE OCEANS, Sydney, Australia, May 2010, pp. 1–6
9 Berger, C.R., Wang, Z., Huang, J., Zhou, S.: 'Application of compressive sensing to sparse channel estimation', *IEEE Commun. Mag.*, 2010, **48**, (11), pp. 164–174
10 Yerramalli, S., Mitra, U.: 'Blind resampling parameter estimation for doubly selective underwater acoustic channels'. Proc. ISCAS, Paris, France, May 2010, pp. 809–812
11 Liu, H., Tureli, U.: 'A high-efficiency carrier estimator for OFDM communications', *IEEE Commun. Lett.*, 1998, **2**, (4), pp. 104–106

doi: 10.1049/iet-spr.2010.0347

12 Li, B., Zhou, S., Stojanovic, M., Freitag, L.: 'Pilot-tone based ZP-OFDM demodulation for an underwater acoustic channel'. Proc. MTS/IEEE OCEANS, Boston, MA, USA, September 2006, pp. 1–5

13 Tropp, J.A.: 'Just relax: Convex programming methods for identifying sparse signals in noise', *IEEE Trans. Inf. Theory*, 2006, **52**, (3), pp. 1030–1051

14 Kim, S.-J., Koh, K., Lustig, M., Boyd, S., Gorinevsky, D.: 'An interiorpoint method for large-scale ℓ_1-regularized least squares', *IEEE Trans. Signal Proc.*, 2007, **1**, (4), pp. 606–617

15 Wright, S.J., Nowak, R.D., Figueiredo, M.A.T.: 'Sparse reconstruction by separable approximation', *IEEE Trans. Signal Proc.*, 2009, **57**, (7), pp. 2479–2493

16 Tropp, J.A., Gilbert, A.C.: 'Signal recovery from random measurements via orthogonal matching pursuit', *IEEE Trans. Inf. Theory*, 2007, **53**, (12), pp. 4655–4666

17 Needell, D., Tropp, J.A.: 'CoSaMP: Iterative signal recovery from incomplete and inaccurate samples', *Appl. Comput. Harm. Anal.*, 2008, **26**, (3), pp. 301–321

18 Dai, W., Milenkovic, O.: 'Subspace pursuit for compressive sensing signal reconstruction', *IEEE Trans. Inf. Theory*, 2009, **55**, (5), pp. 2230–2249

19 Donoho, D.L., Tsaig, Y.: 'Fast solution of ℓ_1-norm minimization problems when the solution may be sparse', *IEEE Trans. Inf. Theory*, 2008, **54**, (11), pp. 4798–4812

20 Neira, L.R., Lowe, D.: 'Optimized orthogonal matching pursuit approach', *IEEE Signal Proc. Lett.*, 2002, **9**, (4), pp. 137–140

21 Andrle, M., Neira, L.R., Sagianos, E.: 'Backward-optimized orthogonal matching pursuit approach', *IEEE Signal Proc. Lett.*, 2004, **11**, (9), pp. 705–708

22 Jin, Y., Rao, B.D.: 'Performance limits of matching pursuit algorithms'. Proc. ISIT, Toronto, Canada, July 2008, pp. 2444–2448

23 Akino, T.K.: 'Optimal-weighted RLS channel estimation for rapid fading MIMO channels', *IEEE Trans. Wirel. Commun.*, 2008, **7**, (11), pp. 4248–4260

24 Barhumi, I., Leus, G., Moonen, M.: 'Optimal training design for MIMO OFDM systems in mobile wireless channels', *IEEE Trans. Signal Proc.*, 2003, **51**, (6), pp. 1615–1624

25 Candes, E.J., Tao, T.: 'Decoding by linear programming', *IEEE Trans. Inf. Theory*, 2005, **51**, (12), pp. 4203–4215

26 Jang, I.S., Wang, X.: 'Discrete stochastic approximation algorithms for design of optimal sensor fusion rules', *Int. J. Sens. Netw.*, 2007, **2**, (3/4), pp. 211–217

27 Raghavendra, M.R., Giridhar, K.: 'Improving channel estimation in 1 systems for sparse multipath channels', *IEEE Proc. Lett.*, 2005, **12**, (1), pp. 52–55