ABSTRACT
Many underwater acoustic channels exhibit correlated (semi-deterministic) multipath arrivals. Such channels are often time-varying with extensive multipath delay and yet have a limited number of degrees of freedom due to inter-path correlation. Traditional channel estimation algorithms do not exploit this correlation structure. To exploit cross-tap correlation, the channel impulse response is projected into a lower-dimensional signal subspace determined by the eigenvectors (bases) of the channel covariance matrix associated with the signal, represented by a set of uncorrelated channel components. Assuming constant bases, a model-based channel tracking algorithm is proposed, in which a priori knowledge of the characteristics of the channel, in the form of an autoregressive (AR) model for the uncorrelated channel components, is incorporated directly into the structure of the algorithm. The AR model determines the state transition matrix for a Kalman filter, which is used to further improve the tracking performance by compensating for the time variation intrinsically within the algorithm recursion. The proposed algorithms track only the small signal subspace and produces significant saving in computations. Performance is demonstrated with real data.

Categories and Subject Descriptors
I.5.4 [Applications: Signal processing]

General Terms
Algorithms

Keywords
Acoustic communication, correlated channel, model-based approach, Kalman filter

1. INTRODUCTION
Phase coherent demodulation of communication signals relies, explicitly or implicitly, on the accurate estimation of the impulse responses of the communication channels. For channel estimation based equalizers, the channel impulse responses (CIRs) can be estimated by minimizing the signal prediction error, the error between the received and predicted signals, using training or decision symbols. Traditional approaches treated all multipath arrivals as independent and uncorrected, in what is known as the uncorrelated scattering (US) model. Multipath arrivals in underwater acoustic channels are often found to be cross-correlated on a short-term scale (less than the channel temporal coherence time) for the purpose of channel tracking. For correlated channels, the channel covariance matrix can be decomposed into a group of eigenvectors (bases) associated with the signal (with large eigenvalues) and the rest eigenvectors associated with the noise. The channel components obtained by projecting the channel impulse response onto the lower-dimensional signal subspace are then uncorrelated. Assume constant (time-invariant) bases, the channel components can be estimated/tracked using the recursive least square (RLS) methods. This approach assumes that all channel components evolve with time with the same rate, which is not supported by the data. To capture the channel dynamics, an autoregressive (AR) model for the uncorrelated channel components are formulated in this paper, whose coefficients are estimated from the time-varying channel covariance matrix. The priori knowledge of the characteristics of the channel is incorporated directly into the structure of the algorithm for channel estimation and symbol equalization. Assuming a properly defined model structure and reliably estimated model parameters, one may obtain more accurate channel estimates and therefore improved demodulation performance. This approach can be generalized to other channels that vary more rapidly with time, wherein the signal bases may also be time varying and need to be tracked simultaneously. This latter case will be addressed elsewhere.

In slowly varying channels, one can exploit the time variation of the channel to improve channel tracking. Fuxjaeger and Ilitis [1] adopted a path-centric model where both the amplitude and delay of the channel multipath components were modeled as AR processes and estimated via Extended Kalman filter. Kalman filter takes advantage of the channel's time evolution model but estimating its parameters can be a challenging problem since the time-varying parameters are not directly observed except for some special cases (e.g., Jakes model for radio based channel [2]). Tsatsanis et al. [3] proposed a method to estimate the model parameters from the covariance matrix of the input and output data for channels and demonstrate the method's effectiveness in a simulated channel with very short delay spread. This approach is adopted here but is applied to the covariance matrix of the CIRs instead. A Kalman filter is then used to adaptively track the channel components. The tracking performance is a function of the model parameters, which are determined by a trade-off analysis; the results are demonstrated with real data.

Since the signal subspace has a small number of degrees of freedom \( L, L \ll N \), where \( N \) is the number of taps used to represent the original CIR, the reduced rank implies a significant saving in numerical computations for the recursive algorithm. For example, the conventional RLS algorithm involves \( N \times N \) calculations. In comparison, the reduced degrees of freedom only require \( L \times L \) calculations, when one tracks only the signal subspace.

This paper is organized as follows. The system model and a model-based tracking algorithm together with performance assessment are proposed in Sec. II. The proposed model-based tracking algorithm is applied to the AUVFest07 data in Section III.
to demonstrate its tracking performance in terms of the signal prediction error. The results are compared with that obtained using the traditional LMS and RLS adaptive algorithms. Also shown is the tracking result applying RLS only to the signal subspace. A summary of the paper is given in Section IV.

2. SYSTEM MODEL

2.1 Signal Model

For a broadband signal with a bandwidth much less than the carrier frequency $f_c$, the transmitted signals can be written as

$$x(t) = \Re \{\hat{x}(t)e^{j2\pi f_c t}\},$$

where $\hat{x}(t)$ represents the signal envelope. The received signal envelope, in terms of the expression $y(t) = \Re \{\hat{y}(t)e^{j2\pi f_c t}\}$, can be written as:

$$\hat{y}(t) = \int h(t;\tau)\hat{x}(t-\tau)d\tau,$$

where $h(t;\tau)$ denotes the base-band channel impulse response at (geo)time $t$ and $\tau$ denotes the multipath delay time. The channel is said to be wide-sense stationary (WSS) if the expected value satisfies [4]

$$E[h^*(t;\tau)h(t';\tau')] = \Gamma(t-t', \tau-\tau'),$$

where $E[x]$ denotes the ensemble average of $x$. A WSS channel is said to exhibit delay uncorrelated scattering (US) if

$$\Gamma(t-t', \tau-\tau') = \delta(t-t')\delta(\tau-\tau).$$

Note that $\Gamma(\tau, \tau) = \Gamma(\tau, \tau; 0)$ measures the cross-correlation between paths arriving at delay times $\tau$ and $\tau'$ and will be referred to as cross-path correlation. Uncorrelated scattering means that the multipath arrivals at different delay times are uncorrelated. Using the Ergodic theorem, one replaces the ensemble average by the time average

$$\Gamma(t, \tau) = E[h^*(t;\tau)h(t';\tau')] = \frac{1}{T} \int_T^T h^*(t;\tau)h(t';\tau')dt,$$ (1)

where $T$ is the time duration of the data samples; $\Gamma$ is also known as the covariance matrix of the multipaths. One also defines a cross-path coherence given by

$$\rho(\tau, \tau') = \frac{E[h^*(t;\tau)h(t';\tau')]}{\sqrt{E[h^*(t;\tau)^2] E[h^*(t';\tau')]^2}}.$$ (2)

Assuming that the signals is WSS and band limited, expressing the CIR in discrete samples, $h(k, t) = h(\tau = kT_0; t)$, $k = 1, \cdots, N$ where $T_0$ is the inverse of the sampling rate, one can express the coherence matrix $\Gamma$ and the coherence matrix $\rho$ as an $N \times N$ matrix, whose $i,j$-th element is given by Eq. (1) and Eq. (2) respectively. These two matrices will be a diagonal matrix if the multipaths are uncorrelated (the US model). On the other hand, if either matrix has substantial non-diagonal elements, it suggests that some of the path arrivals are correlated. This paper is concerned with the latter case, referred to as a correlated channel (CS) model, since many underwater acoustic channels exhibit non-zero cross-path correlations (see Sec. III below). In this case, one can perform eigenvalue decomposition (EVD) of the (square) channel covariance matrix, $\Gamma = Q\Lambda Q^H$, where $Q$ is a matrix of orthonormal eigenvectors, $\Lambda = \text{diag}(\lambda_1, \lambda_2, \cdots, \lambda_N)$, $\Lambda = \text{diag}(\chi_1, \chi_2, \cdots, \chi_N)$ is a diagonal matrix of eigenvalues, in descending order; the superscript $H$ denotes Hermitian conjugate.

It should be noted that this eigenvector decomposition is conducted in the delay-time domain, not in the spatial domain. Given that the eigenvectors of $Q$ form a complete set, one can expand the channel taps at geotime $n$, $h(n) = [h_1(n), h_2(n), \cdots, h_N(n)]^T$, in terms of the orthonormal eigenvectors, $\mathbf{h}(n) = \sum_{i=1}^{N} \alpha_i(n) \mathbf{q}_i + \cdots + \alpha_N(n) \mathbf{q}_N = \mathbf{Q} \mathbf{z}(n)$, where $\mathbf{z}(n) = [z_1(n), z_2(n), \cdots, z_N(n)]^T$; the superscript $T$ denotes transpose of a vector. Note that $\alpha_i(n)$ is the projection of the CIR onto the $\mathbf{q}_i$ vector, to be referred to as the channel components in the eigenvector space. One also finds $E[\mathbf{h}(n)\mathbf{h}^H(n)] = \mathbf{Q} E[\mathbf{z}(n)\mathbf{z}^H(n)] = \mathbf{Q}\Lambda\mathbf{Q}^H$ or $E[\mathbf{z}(n)\mathbf{z}^H(n)] = \Lambda$. In other words, the channel components are uncorrelated, since $\Lambda$ is a diagonal matrix. In the $z$ space, the channel follows a US model.

In practice, to improve the sensitivity to channel variation and reduce the dynamic range of the $\Lambda$ matrix, one estimates and removes the “deterministic” components of $\mathbf{A}$. Defining $\tilde{\mathbf{h}}(t) = \mathbf{h}(t) - \mathbf{H}$, where $\mathbf{H}$ is the channel mean (to be estimated from the probe signal or training data), one obtains the covariance matrix $E[\tilde{\mathbf{h}}(t)\tilde{\mathbf{h}}^H(t)] = \mathbf{Q} E[\mathbf{z}(n)\mathbf{z}^H(n)] = \mathbf{Q}\Lambda\mathbf{Q}^H$, where $\Lambda$ and $\mathbf{Q}$ are the corresponding eigenvalues and eigenvectors. We shall for the rest of the paper deal with the reduced channel, $\tilde{\mathbf{h}}(t) = \mathbf{h}(t) - \mathbf{H}$. For notational simplicity, we shall henceforth drop the tilde above the symbol $\tilde{\mathbf{A}}$ and $\tilde{\mathbf{Q}}$ when doing so will not create any confusion.

2.2 Auto-regressive Analysis

Assume that the channel components $\mathbf{z}(n)$ follow a $p$-order Markov process for the discrete-time random process

$$\mathbf{z}(n) = \sum_{l=1}^{p} \Phi(n, l)\mathbf{z}(n-l) + \mathbf{\eta}(n),$$

where $\Phi(n, l)$ represents the state transition matrix, $\mathbf{\eta}(n)$ is the process noise vector where $\mathbf{\eta}(n) = [\eta_1(n), \eta_2(n), \cdots, \eta_N(n)]^T$, and $n$ refers to the (measurement) time. The state transition matrix can be determined from the covariance of the channel components as follows [1, 3, 5]. Multiplying Eq. (3) (from the right) by $\mathbf{z}^H(n-m)$ (where $m$ is a number between 0 and $p$) and taking an ensemble average, one has

$$E[\mathbf{z}(n)\mathbf{z}^H(n-m)] = \sum_{l=1}^{p} \Phi(n, l)E[\mathbf{z}(n-l)\mathbf{z}^H(n-m)] + E[\mathbf{\eta}(n)\mathbf{z}^H(n-m)].$$

(4)

Given that the channel components are uncorrelated with each other, $E[\mathbf{z}(n)\mathbf{z}^H(m)] = R_z(n-m)\delta(n-m)$, and uncorrelated with the noise, $E[\mathbf{\eta}(n)\mathbf{z}^H(m)] = 0$ for $m \neq 0$, one finds

$$E[\mathbf{\eta}(n)\mathbf{z}^H(n-m)] = \sum_{l=1}^{p} \Phi(n, l)E[\mathbf{\eta}(n)\mathbf{z}^H(n-m-l)] + E[\mathbf{\eta}(n)\mathbf{\eta}^H(n-m)] = R_\eta\delta(m),$$

(5)
Given the measurements of the covariance matrix $R_l$, one can solve Eq. (7) to determine the $p$ transition coefficients $[\Phi_1, \Phi_2, \ldots, \Phi_p]$ for each ($i$-th) tap. For $m = 0$, we have

$$R_i(0) = \sum_{m=0}^{p} \Phi_{ij} R_i(-l) + R_{ij},$$

from which, one can determine the variance of its process noise having determined the transition coefficients $\Phi_{ij}$ using Eq. (7). Equations (7) and (8) are known as Yule-Walker Equation.

### 2.3 Model Based Tracking Algorithm

Channel tracking will be conducted using Kalman filter by (recursively) minimizing the signal prediction error. The output residual, after subtracting the “predominant components” $\hat{R}^H d(n)$ from the received data $x(n) = h^H (n) d(n) + v(n)$, is given by

$$y(n) = x(n) - \hat{h}^H d(n) = \hat{h}^H (n) d(n) + v(n),$$

where $d(n) = [d(n), d(n-1), \ldots, d(n-N)]^T$ is the transmitted data vector, assumed to be an i.i.d sequence for a given modulation scheme, $\hat{h}(n)$ is the residue CIR vector defined above, and $v(n)$ is the observation noise, assumed to be additive white Gaussian with a variance given by $\sigma^2 = E[v(n)v'(n)]$. Expanding $\hat{h}(n)$ in terms of the eigenvectors (defined above),

$$\hat{h}(n) = Q z(n),$$

one finds

$$y(n) = z^H (n) d_q(n) + v(n)$$

where $d_q(n) = Q^H d(n)$ is the transmitted data vector in the $z$ space. One may re-organize Eq. (3) as

$$\begin{bmatrix}
  z(n) \\
  z(n-1) \\
  \vdots \\
  z(n-p+1)
\end{bmatrix} = \begin{bmatrix}
  \Phi(1) & \Phi(2) & \ldots & \Phi(p) \\
  I_n & 0_n & \ldots & 0_n \\
  \vdots & \ddots & \ddots & \vdots \\
  0_n & \ldots & I_n & 0_n
\end{bmatrix} \begin{bmatrix}
  z(n-1) \\
  z(n-2) \\
  \vdots \\
  z(n-p)
\end{bmatrix} + \begin{bmatrix}
  \eta(n) \\
  \eta(n) \\
  \vdots \\
  \eta(n)
\end{bmatrix}$$

Defining an extended data vector $D(n) = [d_q(n), \theta_{p \times (p-1)}^T]$, a channel state vector $Z(n) = \begin{bmatrix} \tilde{z}(n), \ldots, \tilde{z}(n-p+1) \end{bmatrix}^T$, and a model/process noise vector $N(n) = \begin{bmatrix} \eta(1), \theta_{p \times (p-1)}^T \end{bmatrix}$, Eq. (3A) and Eq. (10) can be expressed in the state space form as

$$Z(n) = \Phi Z(n-1) + N(n)$$

$$y(n) = D(n) Z(n) + v(n),$$

where the state transition matrix $\Phi_k$ is given by the matrix in Eq. (3A) and the covariance matrix of process noise is denoted as $R_z = E[N(n)N'(n)]$. In comparison, the exponentially weighted RLS algorithm can be phrased as follows [6],

$$Z(n) = \lambda^{-1/2} Z(n-1)$$

$$y(n) = D(n) Z(n) + v(n)$$

Compared with Eq. (11), one notes that the RLS algorithm assumes: 1) an order one Markov process, $p = 1$, 2) no process noise for the state transition, and 3) that the taps have the same transition coefficients $\lambda^{-1/2}$, where $\lambda$ is the forgetting factor in RLS algorithm. The last assumption implies that the covariance matrix $E[z(n)z'(n)]$ is proportional to the unit matrix. For correlated channels, one expects the covariance matrix $E[z(n)z'(n)] = \Delta$ to have significantly different eigenvalues. This implies that the state transition coefficients are different for different taps. In that case, the use of the RLS algorithm is not justified. For correlated channels, one should incorporate the channel knowledge in terms of the AR equation as given in Eq. (11).

The Kalman filter tracking algorithm works as follows. Given the current state vector $Z(n)$, one calculates the predicted signal $\hat{y}(n)$ and the signal prediction error $\alpha(n) = y(n) - \hat{y}(n)$. Based on $Z(n)$ and $\alpha(n)$, one estimates the next state vector $Z(n+1)$ as $Z(n+1) = \Phi Z(n) + G(n)\alpha(n)$, where $\Phi Z(n)$ is the prediction based on the AR model, Eq. (11), and $G(n)\alpha(n)$ is the update to the prediction, where the matrix $G(n)$ is known as the Kalman gain. The above process repeats itself, each time incrementing $n$ by 1. The schematic of this processing is shown in Figure 1. At each step, the Kalman filter attempts to balance the prediction with the update-to-the-prediction to produce a statistically optimal estimate of the underlying system state. At each step, the algorithm outputs the residue CIR, $\hat{h}(n) = Q z(n)$, and the squared signal prediction error is denoted as
where usually use the observable mean.

2.4 Reduced Rank Approximation

In the above discussions, we have assumed that the state vector $\mathbf{z}(n)$ has dimension $N$ as the original CIR. If the channel covariance matrix has rank $r$, where $r < N$, one needs only to track the channel (the AR equation) in the dimension $r$ signal space; $r$ is referred to as the signal rank. Given that the observable mean vector quantity $\mathbf{r}$, and $y(n), y(n)$ are not directly measurable, the complexity of the estimator will be low while showing a good performance. The algorithm has been termed as RAE-RLS (Reduced-rank Amplitude Estimation using RLS) for brevity.

2.4.1 Reduced-rank Amplitude Estimation using RLS (RAE-RLS)

The channel component vector $\mathbf{z}_n(n)$ can be estimated by minimizing the MSE

$$
\mathbf{z}_n(n) = \arg \min \mathbb{E} \left[ \left( \mathbf{h}_n(n) - \mathbf{Q} \mathbf{z}_n(n) \right)^2 \right].
$$

Since the mean-squared channel estimation error is unobservable, one usually use the observable mean-squared signal prediction error as a surrogate as given below

$$
\mathbf{z}_n(n) = \arg \min \mathbb{E} \left[ \left( y(n) - \mathbf{z}_n(n) \mathbf{d}_n(n) \right)^2 \right],
$$

where $\mathbf{d}_n(n) = \mathbf{Q} \mathbf{d}(n)$ is the projection of the data vector onto the rank-$r$ signal subspace. The MMSE formulation, Eq. (14), serves as the basis for a recursive least-square algorithm

$$
\mathbf{z}_n(n) = \arg \min \sum_{n=0}^{N} \beta(n,k) \left[ y(n) - \mathbf{z}_n(k) \mathbf{d}_n(n) \right]^2,
$$

where $\beta(n,k)$ is the weighting function. If $\beta(n,k) = \lambda^{-k}$, where $\lambda$ is known as the forgetting factor, then Eq. (15) takes on an EWRLS form. The CIR rank-$r$ estimation, $\hat{\mathbf{h}}(n)$, can be reconstructed by $\hat{\mathbf{h}}(n) = \mathbf{Q} \hat{\mathbf{z}}(n)$. The tracking of correlated CIR $\mathbf{h}(n)$ of dimension $N$ is now reduced to tracking $\mathbf{z}_n(n)$ of dimension $r$, where $r \ll N$. If the subspace dimension $r$ is small and can describe the channel well, the complexity of the estimator will be low while showing a good performance. The algorithm has been termed as RAE-RLS (Reduced-rank Amplitude Estimation using RLS) for brevity.

2.4.2 Reduced-rank Model-based Amplitude Estimation (RMAE)

The Kalman filter algorithm (shown in Figure 1) using the reduced rank approximation is given in Table I. This algorithm will be referred as RMAE (Reduced-rank Model-based Amplitude Estimation) algorithm in this paper. The number of operations for each step is also given in Table I. The algorithm has a complexity on the order of $\max(N,2rN^2)$.

The signal prediction error using the (reduced) rank $r$ approximation is given by

$$
\xi_n(n) = (\mathbf{h}_n(n) - \hat{\mathbf{h}}_n(n))^T \mathbf{d}(n) + \nu(n),
$$

where $\hat{\mathbf{h}}_n(n)$ is the CIR estimated using only the $r$ leading eigenvectors. The mean-squared channel estimation error $\epsilon_r = tr \mathbb{E} \left[ \left( \hat{\mathbf{h}}_n(n) - \hat{\mathbf{h}}_n(n) \right)^2 \right]$ can be expressed as [7]

$$
\epsilon_r = \sum_{n=1}^{N} \chi_n + \sigma^2 \epsilon_r,
$$

where $\chi_n$ is the $n$th eigenvalue of the channel covariance matrix and the quantity $\sigma^2_\beta$ is the variance of the estimation noise in the signal subspace. Larger variance $\sigma^2_\beta$ means more stochastic deviation from the mean signal subspace. The tail summation of eigenvalue $\sum_{n=r+1}^{N} \chi_n$ is known as the model bias and $\sigma^2_\epsilon$ is the estimation noise. If $r = N$, the model bias is zero and the mean-squared error is the noise variance $\epsilon_r = \sigma^2_\beta N$. The rank $r$ projection improves on the full-rank projection whenever $\epsilon_r = \sigma^2_\beta N$. The choice of $r$, which minimizes $\epsilon_r$ is given by

$$
\hat{r}_n = \arg \min \epsilon_r.
$$

It’s worthy to note that $\hat{r}_n$ is not necessary the true rank of the signal covariance matrix; it is a tradeoff between model bias and estimation noise. Since in reality, estimation error vector $\mathbf{h}(n) - \hat{\mathbf{h}}_n(n)$ is not directly measurable, we express the mean-squared signal prediction error by combining (16) and (17)

$$
\mathbb{E} \left[ \xi_n(n)^2 \right] = \sigma^2_\beta \left( \sum_{n=r+1}^{N} \chi_n + \sigma^2_\epsilon \right) + \sigma^2_{\epsilon r}.
$$

This signal prediction error will be evaluated using real data latter in section III.
Table 1. Proposed Kalman-Filter Based Channel Tracking Algorithm, RMAE

| Estimate $\hat{h}(n), \bar{h}$ and $\bar{h}(n)$, for $n=1,\ldots,M$ from training data. |
| Calculate eigenvector $\Gamma = \sum_{m=1}^{M} \hat{h}(n) \hat{h}^H(n) = QAQ^H$ |
| Collect $Q_r$ from $r$ leading eigenvector. |
| Calculate $z_r(n) = Q^r \hat{h}(n)$ . |
| Estimate $\Phi_r$, $R_N$, and set $\sigma_r^2$ (Use ambient noise as an initial guess) |
| Set $Z(0) = \theta_{p=1}$ , |
| For each symbol $n=1,2,3,\ldots,K$ |
| Num. of operations |
| $D(n) = \begin{bmatrix} d^T(n)Q_z^v & 0_{(N-1)r} \end{bmatrix}$ |
| $G(n) = \Phi_r \Phi(n,n-1)D^H(n)$ |
| $\left[D(n)K(n,n-1)D^H(n) + \sigma_r^2 \right]^{-1}$ |
| $3(rp)^2 + rp$ |
| $\alpha(n) = y(n) - D(n)Z(n)$ |
| $rp$ |
| $Z(n+1) = \Phi_r Z(n) + G(n) \alpha(n)$ |
| $(rp)^2 + rp$ |
| $K(n) = K(n,n-1) - \Phi_r G(n)D(n)K(n,n-1)$ |
| $3(rp)^2$ |
| $K(n+1,n) = \Phi_r K(n)\Phi_r^H + R_N$ |
| $2(rp)^3$ |
| Estimator output: $\hat{h}_{\text{BASE}}(n) = Q_r z_r(n)$ |
| $Nr$ |
| End |

3. MODEL-BASED CHANNEL TRACKING: EXPERIMENTAL DATA

This section applies the channel-tracking algorithm presented in Sec. II to the data described below. (See Table I for the details of the RMAE algorithm.) Recall that the RMAE algorithm applies an AR model based Kalman filter in the signal subspace. It contains two parameters, the order parameter $p$ of the AR model, and the rank parameter $r$ of the assumed signal subspace dimension. Without prior information, one resorts to signal processing to determine what values to use based on evaluating the performance accuracy and processing complexities. The trade-off analysis is given in Sec. 3.2 to determine the order parameter and Sec. 3.3 to estimate the channel rank parameter. With the educated determination of these two parameters, the RMAE algorithm is applied to the data described below to determine the mean squared signal prediction error in Sec. 3.4. The model-based channel tracking results are compared with that determined using the non-modeled based algorithms.

In the communication setup, data starts with training symbols. From the training data one estimates $\hat{h}(n)$ (using, for example, the LS method) where $n=1,2,\ldots,M$ for $M$ training symbols, from which one obtains $\hat{h}$, $\bar{h}(n)$, and the eigenvalue and eigenvector matrix $\Gamma = \sum_{m=1}^{M} \hat{h}(n) \hat{h}^H(n) = QAQ^H$ as discussed above. From the eigenvalue spectrum shown in Figure 2, one obtains an order of magnitude initial estimate of the dimension of the signal subspace $L$. Projecting the CIR onto the eigenvector space $z(n) = Q^H \hat{h}(n)$, one determines the AR coefficients from the covariance matrix $E[z(n)z^H(l)] = R_{(n-l)} \delta_l$ as discussed earlier in Sec. 2.2. It is expected that the eigenvectors and AR coefficients determined from the training data should be a good approximation of that determined from the packet of data (when transmitted symbols are known). So a low dimension projection comes with lower cost and can simplify the problem quite significantly.

Equipped with the eigenvectors and the AR coefficients, one can proceed with channel tracking for the rest of data using decision symbols, assuming that they are correctly decoded. The performance is measured in terms of the squared signal prediction error $\xi^2(n)$ defined in preceding texts.

![Figure 2. Eigenvalue $\chi$, for AUVFest07 data as a function of eigenvalue index $n = 1,\ldots,40$ (in dB scale).](image)

3.1 Acoustic Communication Data

To verify the tracking performance of the algorithm proposed in Section II, we revisit the 2007 Autonomous Underwater Vehicle Festival (AUVFest07) acoustic communication data previously reported in [8]. The experiment was conducted in 20m of calm water under relatively calm sea conditions, corresponding to sea state of 0. The sound speed is nearly a constant for this environment. The source and receivers were deployed closed to the bottom 19 and 17 m mounted on a rigid body so that signal fluctuation due to source or receiver motion is not an issue. The source-receiver range was 5 km for this data. Binary phase-shift keying (BPSK) signals coded with m-sequence of 511 chips were transmitted during this experiment. The received signal with a carrier frequency of 17 kHz was sampled at 80k samples/sec, brought to baseband, lowpass filtered, and downsampled to a rate of 8kHz corresponding to two baseband sample per transmitted data symbol interval. A 100 taps FIR filter model was used to represent the time-varying CIR in baseband. The CIR magnitude $|\hat{h}(n)|$ estimated from one packet of data is shown in Figure 3(a).
Based on the estimated CIR, one determines the mean CIR $\bar{h}$ and the residual CIR $\hat{h}(n)=\bar{h}(n)-\bar{h}$, from which one calculates the channel covariance and cross-path coherence matrix using the definition in preceding texts. Figure 3(b) shows the channel covariance matrix, where a tap is half symbol long. It shows that the dominant arrivals have non-negligible cross-correlation. Figure 3(c) shows the cross-path coherence matrix, which indicates that many paths are coherent with each other (with coherence > 0.8) even for some arrivals with small amplitudes. The high coherence between the arrivals is a positive indication that these paths are highly correlated, violating the US assumption.

Using Eq. (9) and noting that $E[z(n)z^*(n)]=\chi$, as a first order estimation of the channel residual, one can approximate $\bar{h}(n)=q_nz(n)$ since the first eigenvalue is larger than the rest eigenvalues. To display the CIR time variation, one plots $|q_nz(n)|$ as a function of time $n$ as shown in Figure 3(d).

![Figure 3](image)

**Figure 3.** (a) CIR measured in AUVFest07 experiment. (b) Multipath covariance matrix as a function of tap number. (c) Cross-path coherence as a function of tap number. (d) time variation of $|q_nz(n)|$ in dB scale, the fluctuation of the CIR projected onto the most significant eigenvector.

### 3.2 Performance vs. AR order

The AR order $p$ is one of the parameters that may affect the signal prediction error. As discussed earlier, the choice of the AR order $p$ is generally a tradeoff between the computational complexity and accuracy. Higher order AR processes may be more accurate in generating channel prediction but are computationally more intensive. Lower order AR processes involve fewer computations and are generally more preferred, although they may be less accurate. The question is: what is the lowest order $p$ that yields an acceptable tracking error? The answer requires a trade-off analysis.

To address the above question, we study the normalized signal prediction MSE, defined by $E[|\xi_p(n)|^2]/E[|x(n)|^2]$, as a function of the order parameter $p$, where $x(n)$ is the received signal, $\xi_p(n)$ is the signal prediction error using order $p$ AR model, and $E[x]$ denotes average over the entire data samples (after the training). The results are displayed in Figure 4 using four values of rank $r$: $r = 10, 20, 30$ and 47. For each case, we vary $p$ from 1 to 3. One finds that the normalized signal prediction MSE reaches its minimum value at $p=2$, which corresponds to $\sim$3dB MSE drop at rank $r=47$. On the other hand, the computational cost increases significantly (see Table I) from $p=1$ to 2. If 3dB MSE loss is not critical, then $p = 1$ will be preferred.

![Figure 4](image)

**Figure 4.** Normalized signal prediction MSE $E[|\xi_p(n)|^2]/E[|x(n)|^2]$ as a function of AR order $p$.

### 3.3 Performance of rank $r$ approximation

The parameter $r$ can be estimated by a trade-off analysis between the model bias, and noise variance taking into consideration also the algorithm complexity. We may evaluate the signal prediction error as function of $r$, i.e., (when) only $r$ variables of $z(n)$ are involved in the tracking algorithm. Denoting $\xi(n)$ as the signal prediction error using signal subspace of dimension $r$, the normalized signal prediction MSE, defined by: $E[|\xi(n)|^2]/E[|x(n)|^2]$, is shown in Figure 5 for the RAE-RLS algorithm and the RMAE algorithm (with AR order $p=1$ and $p=2$). It is shown that the normalized MSE for the RMAE algorithm drops to a minimum value at around $r=50$ and slightly goes up with an increasing $r$ value. On the other hand, the normalized MSE using the RAE-RLS algorithm drops to its minimum at around $r=48$ and goes up rapidly with increasing $r$ value. As shown by Eq. (18), the normalized MSE involves a sum of the model bias and estimation error variance. The model bias term decreases with increasing $r$, since a larger signal subspace is used. On the other hand, the estimation error variance increases with $r$ due to increasing channel estimation noise in the (less significant) signal subspace. One notes that when the RMAE algorithm estimates the model parameters correctly for each decoupled $z$ component (state transition coefficient and process noise covariance), tracking more variables than $r$ does not significantly improve the tracking performance. One finds that the minimum normalized MSE for RAE-RLS is $-19.1$dB and the minimum normalized MSE for the RMAE algorithm is $-23.8$dB ($r=50$ and $p=2$). The MSE using the RMAE algorithm is much
lower than that using RAE-RLS algorithm because the latter uses the 
forgetting factor for all channel components and ignored the 
process noise. There are also the LMS and EWRLS algorithms, 
which tracks all ($N=100$) channel taps. One finds that the 
normalized signal prediction MSE for these two algorithms are 
-13.8dB and -13.1dB, respectively (as shown by the marks on right 
edge of Figure 5). The results show that: (1) the RLS algorithm 
tracking the correlated signal subspace (RAE-RLS) outperforms 
the conventional RLS algorithm tracking the full channel space 
(ignoring the channel correlation), and (2) the model-based 
approach (RMAE algorithm), when tracking only few tap 
coefficients ($r \geq 5$), outperforms the LMS and EWRLS algorithm 
tracking all ($N=100$) taps. Note that $\hat{r}_m$ in practice may not be 
the same as the dimension of the signal subspace based on the 
eigenvalue spectrum.

4. SUMMARY AND CONCLUSION

Multipath arrivals are often cross-correlated with each other in 
many underwater communication channels (as shown by the data 
presented above). Traditional channel estimation/tracking 
algorithms do not exploit this correlation structure. For these 
correlated channels, the channel covariance matrix can be 
decomposed into a group of eigenvectors (bases) associated with 
the signal (with large eigenvalues) and the rest eigenvectors 
associated with the noise. The channel components obtained by 
projecting the channel impulse response onto the lower-
dimensional signal subspace are then uncorrelated. To capture the 
channel dynamics, an AR model for the uncorrelated channel 
components are formulated in this paper. The AR model 
determines the state transition matrix and process-noise 
covariance for a Kalman filter, which is used to further improve 
the tracking performance by compensating for the time variation 
intrinsically within the algorithm recursion. Note that the RLS 
algorithm ignores the process noise and assumes that the state 
transition coefficients are the same for all taps, which is not 
supported by the data shown above. It is no surprise that one 
obtains superior tracking performance when applying the model-
based algorithm to real data compared with that using the standard 
RLS and LMS algorithms. The proposed model-based approach 
using the Kalman filter provides an explicit mechanism for 
exploiting the channel statistics, such as decorrelation rate, the 
channel mean and noise covariance, yielding a “goodness of fit” 
of the model in addition to the commonly used signal prediction 
error. Significant saving in computations can be obtained by 
tracking the channel in the (small) signal subspace instead of 
tracking all tap coefficients as done in traditional tracking 
algorithms.

5. ACKNOWLEDGMENTS

The work is supported by the Taiwan National Science Council.
6. REFERENCES


[2] "See for example, W. C. Jakes, *Microwave Mobile Communications*. New York, Wiley, 1974. In this case, the AR parameters can be obtained by fitting the autocorrelation function to the PSD and then solve the resulting Yule-Walker equations."


