Region of Feasibility of Interference Alignment in Underwater Sensor Networks

Parul Pandey, *Student Member, IEEE*, Mohammad Hajimirsadeghi, and Dario Pompili, *Associate Member, IEEE*

***Abstract*—To enable underwater applications such as coastal and tactical surveillance, undersea explorations, and real-time picture/video acquisition, there is a need to achieve high data-rate and reliable communications underwater, which translates into attaining high acoustic channel spectral efficiencies. Interference alignment (IA), which has been recently proposed for radio-frequency multiple-input–multiple-output (MIMO) terrestrial communication systems, aims at improving the spectral efficiency by enabling nodes to transmit data simultaneously at a rate equal to half of the interference-free channel capacity. The core of IA in the space domain lies in designing transmit precoding matrices for each transmitter such that all the interfering signals align at the receiver along a direction different from that of the desired signal. While promising, however, there are still challenges to solve for the practical use of IA underwater, i.e., imperfect acoustic channel knowledge, high computational complexity, and high communication delay. In this paper, a feasibility study on the employment of IA underwater is presented. A novel distributed computing framework for sharing processing resources in the network so to parallelize and speed IA algorithms up is proposed; also, such framework enables "ensemble learning" of various precoding matrices computed using different (competing) IA algorithms so to achieve efficient alignment of the interference at the receiver. The robustness of the IA technique against imperfect acoustic channel knowledge is also quantified by estimating precoding matrices based on predicted channel coefficients. Finally, the performance of an algorithm to predict the underwater acoustic channel impulse response is presented using real data sets.**

***Index Terms*—Interference cancellation, interference suppression, underwater acoustics, wireless sensor networks.**

I. Introduction

UNDERWATER acoustic sensor networks (UW–ASNs) [1] consist of static and mobile sensor nodes deployed to perform collaborative monitoring tasks over a body of water. These networks enable oceanographic applications such as environmental monitoring, offshore exploration, and video-assisted navigation. Due to propagation limitations of radio frequency (RF) and optical waves underwater, i.e., high medium absorption and scattering, respectively, acoustic communication technology is employed to transfer information wirelessly between underwater nodes that are more than 100 m apart. However, because of the small bandwidth, limited to a few tens of kilohertz using current acoustic communication technology, there is a need to maximize the underwater spectral efficiency. This is essential to enable high data-rate reliable multimedia applications such as video/audio stream transfer, transfer of metadata associated with these streams, and time-critical monitoring processes.

To support high data-rate wireless terrestrial applications, multiple-input–multiple-output (MIMO) systems have been recently proposed. Such systems, which are composed of nodes endowed with multiple antennas, are able to exploit the scattering and multipath fading (due to tens of paths) in such a way as to provide higher spectral efficiencies using the same transmission output power. The term antennas used here actually refers to electroacoustic transducers; such term is used only for establishing a better analogy with MIMO systems in the terrestrial radio environment. Similarly, the MIMO technology can take advantage of the rich scattering and heavy multipath of the underwater acoustic environment (characterized by up to hundreds of multiple paths) so to increase data transmission rates and improve link reliability in UW–ASNs [2]. While still not mature, the promise of this new technology has also been recognized by the underwater acoustic communication community in recent years.

To increase the spectral efficiency of multiuser wireless terrestrial networks, a technique called interference alignment (IA) has been proposed [3], [4]. This technique enables the transmitter–receiver pairs ("users") to transmit data simultaneously at a rate equal to half their interference-free channel capacity. Each receiver receives signal (desired signal) from the transmitter with which it is communicating and also receives signal (undesired/interfering signal) from nearby transmitters. The goal of IA is to design transmit signals for all users (transmitter–receiver pairs) in such a way that the undesired signals at each receiver fall in the same subspace and that subspace of undesired signal are linearly independent of the subspace of the desired signal after decoding (and, therefore, are easily suppressible). The receiver then applies an interference-suppression filter to project the desired signal onto the interference-free dimension of the network.

While IA is a very promising technique to increase the network capacity, there are a few research challenges to solve for its practical use underwater, namely, the assumption of "perfect" channel knowledge, the high computational complexity of IA algorithms, the extremely large propagation delay of acoustic

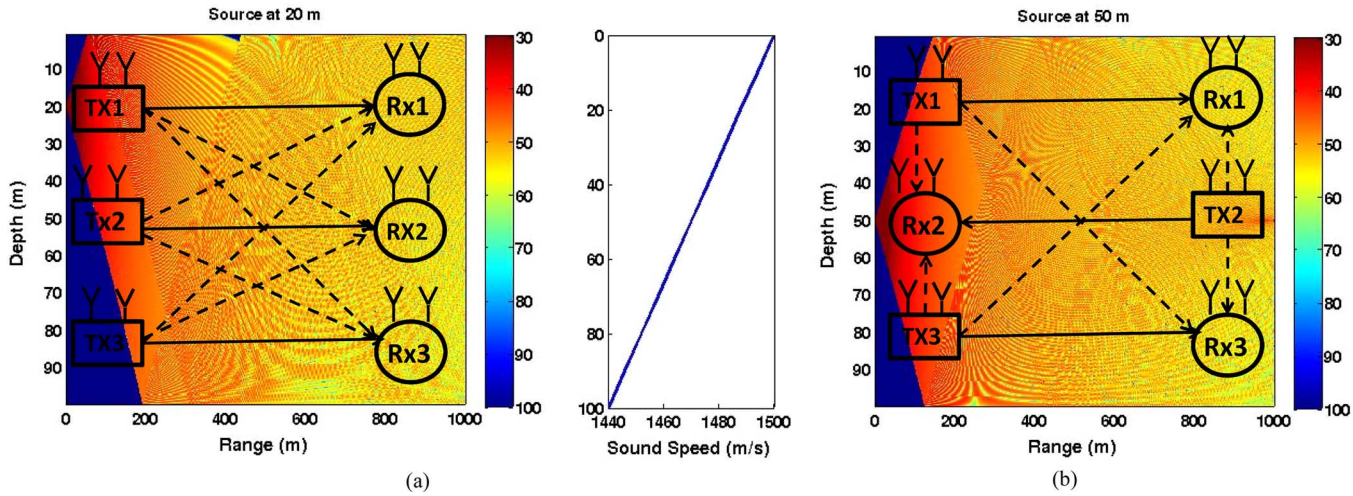
Manuscript received December 27, 2012; revised July 22, 2013 and October 18, 2013; accepted November 27, 2013. Date of publication January 02, 2014; date of current version January 09, 2014. This work was supported by the National Science Foundation (NSF) CAREER Award OCI-1054234.

Associate Editor: J. Potter.

The authors are with the Department of Electrical and Computer Engineering, Rutgers—The State University of New Jersey, Piscataway, NJ 08854 USA (e-mail: parul@cac.rutgers.edu; hajimirsadeghi.sm@cac.rutgers.edu; pompili@cac.rutgers.edu.

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/JOE.2013.2293932



(a) (b)

Fig. 1. (a) Topology 1: An UW–ASN with $K = 3$ users (i.e., 3 Tx–Rx pairs) with $N_T = N_R = 2$ (i.e., two antennas on each node). For a clear visualization, we have shown the Bellhop channel profile in shallow water (maximum depth of 100 m, range up to 1 km) associated *only* with Tx3. The transmission loss of the channel is given in decibels. Here, we used the linearly decreasing sound-speed profile (right subfigure) with the depth being the same as in the left subfigure. (b) Topology 2: Variation of spectral efficiency. Here, we have shown the Bellhop channel profile associated *only* with Tx1 for the linearly decreasing sound-speed profile.

waves underwater (five orders of magnitude higher than for terrestrial RF waves), and the tight synchronization requirements. Existing IA algorithms assume "perfect" knowledge of the channel coefficients at the communicating nodes; achieving an accurate estimation of the time- and space-varying underwater acoustic channel, however, is a challenging open-research task. Also, the propagation speed of acoustic waves (five orders of magnitude lower than RF waves in air) varies with water temperature, salinity, and pressure (i.e., depth), which cause wave paths to bend toward regions of lower sound speed (Fig. 1). Acoustic waves are also reflected from the surface and bottom. Such uneven wave propagation results in convergence (or shadow) zones, which are characterized by a lower (or higher) transmission loss due to constructive (or destructive) multipath (yellow or blue in the figure, respectively).

In this work, we study the effect of an inaccurate estimation of the underwater acoustic channel on IA and discuss the tradeoffs of multiplexing and link reliability, power, and number of concurrent users associated with IA. Also, to overcome the challenge of computational complexity and communication delay, we introduce a novel distributed computing framework for sharing processing resources in the network. We provide the computing infrastructure to support the distributed capabilities of existing IA algorithms by forming an elastic resource pool; using the collective computational capability of this pool, which is composed of neighboring network nodes, computational tasks can be executed in parallel, i.e., in a distributed manner. We present a study on the increase in the region of feasibility of computationally intensive IA algorithms in UW environments through our framework. This involves studying the tradeoffs between computational gain (in terms of speedup over standalone computation) versus communication overhead (and delay) incurred as well as the effect of IA technique on the performance in terms of spectral efficiency. These tradeoffs help us define the scenarios in which the distributed realization of IA is feasible in UW environments. Note that we define the region of feasibility as a set of parameters, namely, the number of users, the number of antennas, and the maximal distance between nodes for which IA is feasible, given the coherence time of the acoustic channel (i.e., the time duration over which the channel impulse response is considered static). By "feasibility" we mean that the overall time required for the execution of an IA algorithm (including the overall computation and communication time involved) is less than the coherence time of the channel. This allows reusability of the computed IA matrices over the coherence time of the channel. To be conservative, in our study, we assume the reusability period to be 50% of the estimated coherence time of the channel.

In [5], Chitre *et al.* implemented IA in the time domain for single-input–single-output (SISO) underwater systems and designed scheduling algorithms that generate high-throughput schedules for the transmission of packets for a given network geometry. These scheduling algorithms leverage the large propagation delay in underwater networks. To the best of our knowledge, ours is the first attempt to improve underwater acoustic communication performance by studying the suitability of IA in the space domain, which can potentially lead to much better performance than its time-domain counterpart as it fully exploits the MIMO capabilities of nodes with multiple antennae. Some work has been done in the area of terrestrial wireless IA in the space domain, although most works restricted themselves to specific scenarios and settings, and none focused on underwater networks: in [6], El Ayach *et al.* study IA in measured MIMO orthogonal frequency-division multiplexing (MIMO–OFDM) interference channels in a terrestrial environment. The measurement campaign includes a variety of indoor and outdoor measurement scenarios. The paper shows that IA achieves the claimed scaling factors, or degrees of freedom, in several measured channel settings for a three-user, two-antenna per node setup; in [7], IA has been applied to terrestrial cellular networks; in [8], Gollakota *et al.* proposed an IA-based medium-access control (MAC) protocol for MIMO

local area networks (LANs) and proposed an algorithm, which is tailored for a specific LAN setting where the access points are connected by a wired infrastructure (e.g., Ethernet), that decides which nodes should be served concurrently.

In this paper, we present a study on the increase in the region of feasibility of computationally intensive IA algorithms in UW environments made possible by our distributed computing framework. We provide robustness to IA so to overcome its intrinsic sensitivity to the quality of the channel knowledge. We introduce "ensemble learning" [9] through our framework, which allows us to choose the best precoding/decoding matrices from different (competing) IA algorithms available in the literature. This leverages on the fact that, depending on the scenario (e.g., the number of nodes, topology, underwater channel conditions, etc.), different IA algorithms exhibit different behaviors/performance. Robustness is also achieved by leveraging on the time correlation in underwater acoustic channel: we use a simple, yet effective, autoregression model to predict the channel; we then estimate the precoding/decoding matrices based on the predicted channel coefficients. Furthermore, we present a study of the performance gain achieved by transmitting all precoding/decoding matrices (which guarantees better IA performance at the price of a higher communication overhead) versus transmitting the pair precoding/decoding matrix that best represents all the estimated pairs (which reduces the communication overhead by lowering IA performance).

To summarize, the main contributions of this work are listed as follows.

- We study whether it is feasible and practical to use IA underwater under different channel and network conditions; we also investigate the tradeoffs between multiplexing and link reliability, output power, and number of concurrent users in the network.
- We propose a distributed computing framework to parallelize iterative IA algorithms and show the gains in terms of achievable spectral efficiency; we discuss the region of feasibility for practical realization of these algorithms and determine to what extent we can parallelize these algorithms, given different underwater acoustic channel coherence times; we implement "ensemble learning" through our framework to select on the fly the precoding/decoding matrices from various IA algorithms so to maximize the spectral efficiency.
- We introduce robustness in IA by leveraging the time correlation between underwater acoustic channel coefficients; we provide a comparative study between spectral efficiency and the communication overhead when unique precoding/decoding matrices are used at the transmitter side for each of the predicted channel coefficients versus when a common precoding/decoding matrix is used.

The rest of this paper is organized as follows. In Section II, we provide the necessary background on IA and study the various associated tradeoffs. In Section III, we discuss the region of feasibility for the IA implementation underwater. In Section IV, we propose a distributed computing framework and explain how we exploit it to support IA underwater. In Section V, we introduce robustness to IA by estimating the underwater acoustic channel coefficients. In Section VI, we use real-data values of underwater acoustic channels to study the performance of a channel prediction algorithm to be used to support IA. Finally, in Section VII, we draw the main conclusions of this work and provide a brief note on future work.

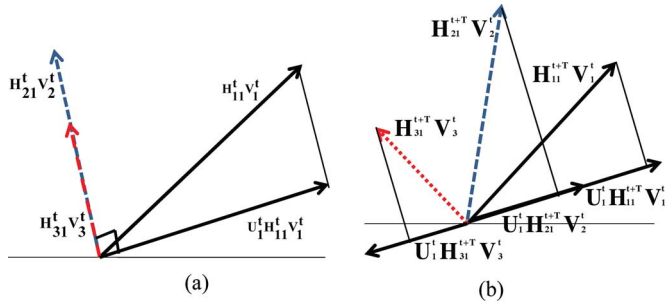


Fig. 2. (a) Under perfect channel knowledge, interfering signals $\mathbf{H_{21}^t V_2^t}$ and $\mathbf{H_{31}^t V_3^t}$ align perfectly. (b) Under imperfect channel knowledge, interfering signals $\mathbf{H_{21}^{t+T} V_2^t}$ and $\mathbf{H_{31}^{t+T} V_3^t}$ do not align, thus resulting in residual interference (leakage).

II. Background and Preliminaries

In this section, first, we present a brief background on IA and formulate how IA can be applied to the wideband underwater acoustic channel. Then, we study the effect of imperfect channel conditions on the performance of IA. Finally, we discuss the tradeoff between multiplexing and link reliability, the need for power control, and the computational complexity of IA algorithms.

A. Interference Alignment and Cancellation

We consider a generic system model, a K-user interference channel system, as shown in Fig. 1. Each transmitter Tx is equipped with N_T antennas and each receiver Rx is equipped with N_R antennas. Each Tx_i is communicating with Rx_i, $\forall i = 1 \ldots K$ [10], [11]. When $K = 2$ and $N_T = N_R = 2$, the channel between transmitter i and receiver j is given as $\mathbf{H_{ij}}$, which can be decomposed as

$$\mathbf{H_{ij}} = \begin{bmatrix} h_{11}^{ij} & h_{12}^{ij} \\ h_{21}^{ij} & h_{22}^{ij} \end{bmatrix} \tag{1}$$

where each entry h_{kl}^{ij} is a complex number whose *magnitude* represents the signal attenuation from transmitter antenna k to receiver antenna l in a time slot and whose *phase* represents the propagation delay (in Fig. 2, $k, l \in \{1, 2\}$). At transmitter i, $\mathbf{x}_i$ is a $d_i \times 1$ symbol vector, where d_i is the number of independent information streams or the degree of freedom for the ith transmitter. The goal of IA is to design transmit precoding matrices $\mathbf{V_i}$ of dimensions $N_T \times d_i$ for each transmitter. The transmitted signal is then given as $\mathbf{s_i} = \mathbf{V_i x_i}$ of dimension $N_T \times 1$.

These matrices are chosen such that, by encoding with them, all the interfering signals lie in a subspace that is linearly independent of the subspace of the desired signal. The heart of IA in the spatial domain lies in constructing these transmit precoding matrices. To decode, the receiver projects the received signal onto a vector that is orthogonal to the vector of the interfering signal. The received signal vector at receiver j is given as

$$\mathbf{r_i} = \mathbf{H_{ii} V_i x_i} + \sum_{j=1, i \neq j}^{K} \mathbf{H_{ji} V_j x_j} + \mathbf{n_i} \tag{2}$$

where the first term is the desired signal at Rx_i and the second term is the interference from all other transmitters. Here, $\mathbf{n_i}$ is the $N_R \times 1$ additive white Gaussian noise (AWGN) or thermal noise vector.

To decode, the receiver projects the received signal onto a decoding vector $\mathbf{U_i}$ that is orthogonal to the data vector of interfering signals. Such a decoding vector can be found by imposing $\mathbf{U_i} = \emptyset(\mathbf{H_{ji}V_i}) = \emptyset([\mathbf{H_{mi}V_m}]^\dagger)$, where $\dagger$ represents the Hermitian or conjugate transpose and $\emptyset(\mathbf{A})$ represents the null space of a generic matrix $\mathbf{A}$, i.e., the set of all matrices $\mathbf{z}$ for which $\mathbf{Az} = \mathbf{0}$. The interference suppression filter $\mathbf{U_i}$ at Rx_i is the null space of all interference matrices $\mathbf{H_{ji}V_i}$ and $[\mathbf{H_{mi}V_m}]^\dagger$ at Rx_i. After applying the interference suppression filter, i.e., $\mathbf{U_i}^\dagger$, the received signal at receiver i is given as

$$\mathbf{y_i} = \mathbf{U_i}^\dagger \mathbf{H_{ii}V_i x_i} + \sum_{j=1, j\neq i}^{K} \mathbf{U_i}^\dagger \mathbf{H_{ji}V_j x_j} + \mathbf{U_i}^\dagger \mathbf{n_i}. \quad (3)$$

The sum of the moduli of the second term in (3) is the total interference at receiver i, which is called interference leakage; e.g., the leakage at Rx_i is defined as $\sum_{j=1,j\neq i}^{K} \|\mathbf{U_i H_{ji} V_j x_j}\|$. In the case of ideal IA, i.e., when the channel knowledge is perfect, the interference signals lie in the same subspace and the interference suppression filter eliminates the interference completely.

B. Interference Alignment for a Wideband Underwater Acoustic Channel

Modeling the channel between a pair of antennas as a single complex number is accurate only for narrowband or flat channels, but becomes less so as the width of the system bandwidth increases. The underwater channel is wideband in nature; hence, to be able to apply IA underwater, we employ OFDM. The idea behind OFDM is to divide the whole band into sub-bands, such that each sub-band undergoes flat fading (assuming that the bandwidth of each sub-band is smaller than the coherence bandwidth of the channel); then, IA can be applied in each sub-band. In this paper and in all our simulations, we assume the channel bandwidth to be 20 kHz.

In [12], Qarabaqi and Stojanovic propose a model to estimate the small-scale variations in the underwater acoustic channel, which can be thought of as those variations that occur over a communication transaction (a packet or a frame of packets). Small-scale modeling of the underwater channel aims at statistically characterizing random effects such as scattering, which leads to the formation of micromultipaths along with the macropaths. Note that the model in [12] is valid only for small-scale phenomena and cannot be used to model the large-scale variations of the acoustic loss. We can model these micropaths along with the macropaths for a path p in the underwater acoustic channel as $h_{p,i} = h_p \gamma_{p,i}$, where h_p is the macropath channel and $\gamma_{p,i}$ represents the scattering component in the underwater channel. This expression implies that the effect of scattering in a narrowband system is that of a multiplicative distortion only. Each scattering component $\gamma_{p,i}$ has multiple micropaths with gain $|\gamma_{p,i}|$ and delay $\delta\tau_{p,i}$. In the absence of any dominant path, the amplitude of each micromultipath (γ) can be modeled as a Rayleigh random variable (r.v.) and its delay is modeled as a Gaussian r.v.; the gain and the amplitude of the nominal parameter h_p are taken from the Bellhop model [13], given the system geometry and the sound-speed profile.

C. Channel Information and Interference Leakage

We now discuss the effect of perfect and imperfect channel knowledge on IA at the communicating nodes (pair).

1) Perfect Channel Knowledge: Let us consider a $K = 3$ user system with $N_T = N_R = 2$ and $d_i = 1$, where d_i can also be interpreted as the number of independent streams transmitted by Tx_i. The channel from Tx_i to Rx_i at time t is given as $\mathbf{H_{ii}^t}$. According to IA, in the case of perfect channel knowledge, the interfering signals lie in the same subspace, and such subspace is independent of the subspace of the desired signal. Fig. 2(a) shows the signals received at Rx_1; here, $\mathbf{H_{21}^t V_2^t}$ and $\mathbf{H_{31}^t V_3^t}$ are the interference signals, whereas $\mathbf{H_{11}^t V_1^t}$ is the desired signal. The dimension of the received signals is $2d_i \times 1$ (i.e., $N_T \times 1$ in this example). We see that both interference signals align, i.e., overlap, perfectly. The interference suppression filter is then applied to these interference signals, which are completely canceled. In the case of ideal IA, the decoding vector $\mathbf{U_1} = \emptyset(\mathbf{H_{21}V_2}) = \emptyset([\mathbf{H_{31}V_3}]^\dagger)$, i.e., it lies in the null space of the interference signals. Hence, in this ideal case, we have $\mathbf{U_i^t H_{ji}^t V_j^t} = \mathbf{0}, \forall j \neq i$, i.e., the interference leakage is completely removed by interference suppression filters and $rk(\mathbf{U_i^t H_{ii}^t V_i^t}) = d_i$, where $rk()$ represents the rank of the desired signal, i.e., the number of parallel streams (degree of freedom).

2) Imperfect Channel Knowledge: Now we consider the alignment at Rx_1 after T [s] from the last channel probing at instant t. We assume that the channel has changed with time (i.e., $T > T_c$, where T_c is the coherence time of the channel) and that the new channel matrix is $\mathbf{H}^{t+T}$. We also assume that the nodes do not have updated channel information. As a result, they continue to use the precoding and decoding matrices $\mathbf{V_i^t}$ and $\mathbf{U_i^t}$, respectively, estimated at time t (based on $\mathbf{H}^t$). Fig. 2(b) shows the interference alignment at Rx_1. Differently from the ideal case depicted in Fig. 2(a), now the interference signals $\mathbf{H_{21}^{t+T} V_2^t}$ and $\mathbf{H_{31}^{t+T} V_3^t}$ no longer align perfectly; this is because the channel has changed and the precoding/decoding matrices used were estimated based on the old channel knowledge $\mathbf{H}^t$. Hence, the interfering signals no longer lie in the same subspace. For this reason, new interference suppression filters would need to be estimated to cancel the interference. As a result, after applying the (old) interference suppression filter $\mathbf{U}_1^t$, the interference is not completely canceled out. The interference leakage in this case is $\mathbf{U_i^t H_{ji}^{t+T} V_j^t} \neq \mathbf{0}, \forall j \neq i$.

As a result, the bit error rate (BER) increases, which leads to the corresponding decrease in the net bit rate, defined as $D_{\mathrm{net}} = d_i \times C \times (1 - \mathrm{BER})$ [b/s], where the capacity $C = B \times \eta$ [b/s] is given as a product of the used bandwidth B [Hz] and the spectral efficiency of the modulation η [b/s/Hz]. The capacity of each individual data stream is multiplied by the number of parallel streams of the network to obtain the capacity of the MIMO system. The higher d_i (degree of multiplexing), the higher is the spectral efficiency of the MIMO system.

3) Tradeoffs Associated With IA for Imperfect Channel Knowledge: Here we study the tradeoffs associated with IA

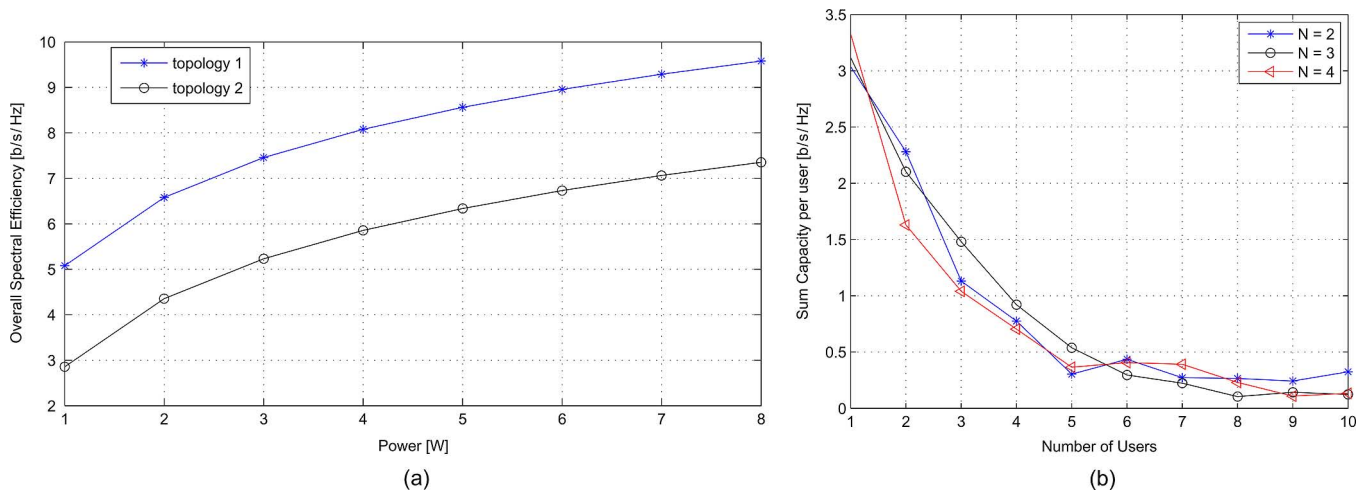


Fig. 3. (a) Variation of spectral efficiency with topology 1 and topology 2 [as given in Figs 1(a) and (b), respectively]. (b) Variation of network capacity with the number of users for different number of antennas ($N = 2, 3, 4$).

when perfect channel knowledge is not available to the nodes. Specifically, we discuss the tradeoff between multiplexing and link reliability as well as what happens when we increase the number of users in case of imperfect channel knowledge. We motivate the need for power control as well as for a distributed computing framework to handle the complexity of the distributed IA algorithms. These tradeoffs have been introduced in [14].

a) Multiplexing and Link Reliability: In MIMO transmissions, to increase the spectral efficiency, multiple data streams are sent out in parallel. If d_i is the number of independent streams sent out, then the multiplexing gain is d_i itself. At Rx_i, the rank of the subspace of the desired signal is d_i. For perfect alignment, the subspace of interfering signals $\mathbf{V_j x_j}$ should have a rank equal to $N - d_i$, where N is the number of transmit antennas and should span the same subspace, linearly independent to the desired-signal subspace. In the case of imperfect channel estimation, however, the interference signals may occupy a subspace of dimension higher than $N - d_i$. To overcome this effect, the number of independent streams d_i at the transmitter can be reduced, which gives a higher link reliability; this is achieved, however, at the cost of a lower multiplexing gain.

b) Power Control: We see that due to the proximity of Tx_2 to Rx_1 in Fig. 1(b), the signal received from Tx_2 is stronger than that received from Tx_1. This is an example of the so-called near–far effect, which is a crucial capacity-limiting effect in code-division multiple-access (CDMA) systems. Hence, the signal from Tx_2 will cause interference at Rx_1, where the residual interference will be $\sum_{j=2}^{3} P^j \|\mathbf{U_1 H_{j1} V_j}\|$. Hence, the higher the power of TX_2, the higher will the interference leakage be.

Fig. 3(a) compares the spectral efficiency of the system associated with the two "extreme" topologies, and shows the performance drop of about 2 b/s/Hz in topology 2 due to this effect.

c) Number of Active Users: In the case of imperfect channel estimation, the interference leakage $\sum_{j=1, j\neq i}^{K} \|\mathbf{U_i H_{ji} V_j}\|$ cannot be neglected. Specifically, as the number of users K increases, the interference leakage also increases, leading to higher BER (i.e., lower link reliability). In Fig. 3(b), we see that, as K increases, the capacity per user of the system decreases; on the other hand, if K is too low, the gain from IA is not exploited properly.

d) Computational Complexity and Communication Delay: Generally speaking, IA algorithms used for estimating the precoding/decoding matrices, e.g., [15] and [16], are computationally intensive as they involve multiple eigenvector calculations and matrix multiplications. Also, these algorithms require the exchange of information between the transmitter and the receiver, which cannot happen instantaneously due to the large underwater communication delay. Furthermore, these calculations need to be performed in a fraction of the channel coherence time T_c so that the estimated precoding/decoding matrices is reusable for the remaining time of the coherence time. To overcome these issues, we propose a novel distributed computing framework that reduces the computation time for estimating the precoding matrices by executing in parallel the tasks composing the IA algorithm(s).

III. Region of Feasibility of Interference Alignment Underwater

Existing IA algorithms (e.g., [15] and [16]) to compute precoding/decoding matrices are computation intensive and require exchange of information between the transmitter and the receiver, which cannot happen "instantaneously" (as ideally desirable) due to the large underwater propagation delay. If the coherence time of the channel is smaller than the time taken to compute the precoding/decoding matrices, then these matrices will not be useful as the channel conditions will have changed before IA can be effectively employed. Moreover, existing IA algorithms rely on initial conditions to converge toward (hopefully) optimal precoding/decoding matrices. By using multiple initial conditions we can decrease the possibility of the IA algorithms to converge to only suboptimal precoding/decoding matrices, which happens because these IA algorithms are iterative in nature and may get "stuck" in local minima. We study the time taken by IA algorithms to converge to their solutions as the independent parameters, namely, the number of users

(K), the number of transmit antennas (N_T), the number of initial conditions (n_{IC}), and the pair distance (dis) are varied, given a certain channel coherence time (T_c). We discuss for what values of these parameters will IA be feasible for a given coherence time. Given a set of parameters, we put a conservative constraint that the time taken to estimate the precoding/decoding matrices be less than 50% of the coherence time of the channel. This allows us to use the precoding/decoding matrices within the coherence time of the channel. The set of parameters meeting this constraint forms the region of feasibility of IA in UW–ASNs.

We consider that the execution of an IA algorithm to estimate the precoding/decoding matrices takes place at one of the nodes. We call this node *master* and the other communicating nodes *slaves*. We consider the network topology to be similar to that in Fig. 1(a). We profiled the execution time of the iterative algorithms in [15] and [16] by assuming homogeneous computational capability of all sensor nodes. The time for one iteration of the IA algorithm is approximately 300 ms (given $K = 3$, $N_T = N_R = 3$, and $\mathrm{dis} = 0.2$ km). We consider the number of iterations (n_{IT}) to be 10 and the number of unique initial conditions (n_{IC}) at each node to be 5. The master transmits the chosen initial conditions to each communicating node within a guard interval (T_g) of 200 ms. The propagation delay (T_p) for underwater environment is taken to be 0.67 s/km [1], and the transmission time (T_t) is given as the ratio of the packet size (i.e., the data to be transmitted) and the data rate, where the packet format proposed in [17] is considered to calculate the packet size. Under these realistic assumptions, the total time taken to transmit from the master to the farthest slave node is $T_p + T_t + (n_{\mathrm{SN}} - 1)T_g$, where n_{SN} is the number of slave nodes. Hence, the total coordination time to transmit and receive precoding/decoding matrices is given as

$$T_{\mathrm{coord}} = 2[T_p + T_t + (n_{\mathrm{SN}} - 1)T_g]. \tag{4}$$

Note that in (4) the factor 2 takes into consideration the two sets of data exchanges between the master and the nodes: in the first exchange of data, the master sends the initial conditions to each node; then, in the second exchange, the master sends the final precoding/decoding matrices to the communicating nodes. On the other hand, the computation time is given as

$$T_{\mathrm{comp}} = 0.3 n_{\mathrm{IT}} n_{\mathrm{IC}} \tag{5}$$

where the constant value 0.3 s gives the time taken for a transmitter to compute IA matrices for one initial condition and iteration. The total time T_{tot} taken for estimation of precoding/decoding matrices is the sum of the coordination (4) and computation time (5), i.e., $T_{\mathrm{tot}} = T_{\mathrm{coord}} + T_{\mathrm{comp}}$.

To study the feasibility of IA underwater, we consider that one of the nodes in the network (serving as the master) estimates the precoding and decoding vector for all the communicating nodes. Each node sends its channel coefficients to the master, which then estimates the precoding/decoding matrices for each slave node and sends them to each communicating slave. The amount of data exchanged (in bytes) between the master and a slave node based on the packet format in [17] is $5 + (N_T d_k n_{\mathrm{IC}} 8) + (N_R N_T K 8)$ bytes, where the first term is the data size of the header, the second term represents the data size of the channel coefficients of all nodes involved in communication (each channel coefficient is a complex floating point number and takes 8 B), and the third term represents the data size of the initial precoding vectors given to each slave node. Similarly, each slave node uses the initial precoding matrices and returns to the master a final precoding and decoding matrices computed from IA algorithm. The amount of data exchanged in this step is $5 + (N_T d_k 8) + (N_R d_k 8)$ bytes, where the second and third terms represent the size of the precoding and decoding matrices, respectively. Finally, from the precoding/decoding matrices received from all the slave nodes, the master selects the matrix that gives the maximum spectral efficiency and sends it to all the slave nodes. The data exchanged for this step is $5 + (N_T d_k 8) + (N_R d_k 8)$ bytes.

The time taken includes the computation time for these matrices and the communication overhead between the nodes and the master. In Fig. 4(a), we depict the performance of such scenarios as the number of users increases. We also plot the results for different numbers of initial conditions. The horizontal solid red, dashed blue, and dotted green lines indicate the different measures of fraction of coherence time, which ranges in [1,10] s. As mentioned earlier, we target to estimate the precoding/decoding matrices in a fraction (50%) of the coherence time. From Fig. 4(a), we see that for a number of users K smaller than 3 and a number of initial conditions n_{IC} smaller than 10, while keeping N_T, N_R, and dis fixed, IA can be used in environments with a channel coherence time greater than 1 s. As the number of users increases (with a fixed number of initial conditions), we see that higher coherence times are required for IA matrices to be usable. For a number of users greater than 9 and the number of initial conditions greater than 20, IA is feasible only in environments with coherence times of the order of tens of seconds. From Fig. 4(b), we see that the total time taken increases as the maximum distance between the master and the nodes in the network increases. The rate at which the time taken increases with the distance is, however, lower in comparison to the rate of increase with the number of users.

In Fig. 5(a), we see that as the number of transmit antennas increases for a fixed number of receiver antennas as well as a fixed number of users, the total time taken increases negligibly. This indicates that increasing the number of transmit antennas does not significantly increase the total time taken to apply IA. In Fig. 5(b), we investigate the IA performance underwater from an energy point of view. The requirements refer to the energy needed for transmission of data packets. As expected, we see that the energy consumed increases by increasing the maximum distance between the master and the nodes.

In [18], Yang studies the underwater acoustic channel coherence time using various real-data sets consisting of measured channel impulse responses. Yang observes that for some channels the coherence time is as high as 30 s, while for others it is around 0.1 s. This difference is attributed to different environmental conditions. Hence, in certain environments, e.g., calm ocean and lakes with high coherence times (e.g., in winter time), IA can be used.

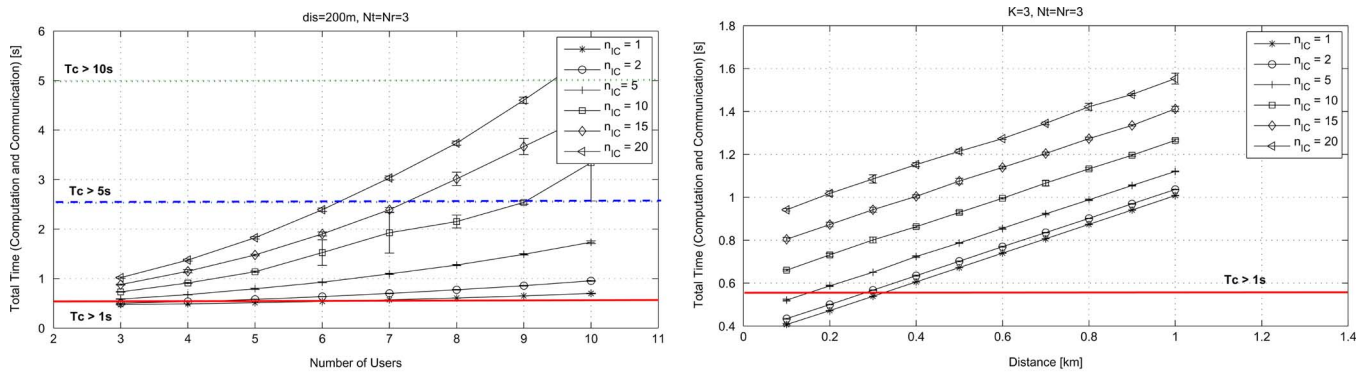


Fig. 4. Total time taken versus (a) the number of users and (b) the maximum node distance for a different number of initial conditions.

IV. Extension of Region of Feasibility for IA via Distributed Computing

We propose a distributed computing framework that reduces the computation time for estimating the precoding matrices by executing in parallel the tasks composing the IA algorithm(s). Such framework, however, introduces some overhead, which needs to be "absorbed" by the gain it brings (i.e., reducing the overall execution time of the IA distributed algorithm): this is another tradeoff involving computational complexity and communication delay. This framework, based on our previous work in [19], has the potential to extend further the region of feasibility of IA in the underwater environment. To distribute the computation load of nodes, we utilize the computing and storage capabilities of the nodes in the vicinity to form an elastic resource pool that can process massive amounts of locally generated data in parallel. We propose a ubiquitous computing solution that is aimed at organizing UW sensor nodes in the vicinity into a wirelessly connected local computing grid. The collective computational capability of this computing grid can be exploited to perform distributed computation.

A. Distributed IA Algorithm

We now explain a powerful iterative IA algorithm [15] for which our framework provides the computing infrastructure to support its distributed capabilities. This algorithm is computation intensive as it involves multiple matrix and eigenvector calculations. The total residual interference $\mathbf{I_j}$ at the receiver of user j due to interference from all undesired transmitters $k \neq j$ is

$$\mathbf{I_j} = \mathrm{Tr}[\mathbf{U_j}^{\dagger}\mathbf{Q_j}\mathbf{U_j}]$$
$$\mathbf{Q_j} = \sum_{k=1,k\neq j}^{K} \frac{\mathbf{P_k}}{d_k}\mathbf{H_{kj}}\mathbf{V_k}\mathbf{V_k}^{\dagger}\mathbf{H_{kj}}^{\dagger} \qquad (6)$$

where $\mathbf{P_k}$ is the transmit power at transmitter k. Each of the d_j columns of $\mathbf{U_j}$ is given by $\mathbf{U_{j[n]}} = \nu_n[\mathbf{Q_j}], n = 1, \ldots, d_j$, where $\nu_n[\mathbf{Q_j}]$ is the eigenvector corresponding to the nth smallest eigenvalue of $\mathbf{Q_j}$. In the beginning of the iterative algorithm, the transmit precoding matrices are initialized with some random values and the interference suppression filters of the original network are calculated using (6). The receiver of user j chooses its interference suppression filter $(\mathbf{U_j})$ to minimize the leakage interference $(\mathbf{I_j})$ due to all undesired transmitters. The d_j-dimensional received signal subspace that contains the least interference is the space spanned by the eigenvectors corresponding to the d_j smallest eigenvalues of the interference covariance matrix $\mathbf{Q_j}$ (or $\nu_n[\mathbf{Q_j}]$).

After determining $\mathbf{U_j}$, the transmitter and the receiver switch their roles. This network is called a "reciprocal" network. The estimated interference suppression filters $(\mathbf{U_j})$ of the original network now become the precoding matrices $(\overleftarrow{\mathbf{V_j}})$ for the reciprocal network, where the arrow at the top indicates that this vector belongs to the reciprocal network. Similar to the original network, in the reciprocal network (with transmitters and receivers switched) the total interference leakage at receiver j due to interference from all undesired transmitters $k \neq j$ is given by $\overleftarrow{\mathbf{I_j}} = \mathrm{Tr}[\overleftarrow{\mathbf{U_j}}^{\dagger}\overleftarrow{\mathbf{Q_j}}\overleftarrow{\mathbf{U_j}}]$. Note that the interference suppression filter $\overleftarrow{\mathbf{U_j}}$ for the receivers of the reciprocal network are calculated only to be used as the transmit precoding matrices of the original network in the next iteration. The iterative algorithm alternates between the original and reciprocal networks with only the receivers updating their interference suppression filters (at every iteration) to minimize their total leakage interference. The algorithm in [16] has the same communication overhead as the one in [15], but requires higher computation time.

B. Leveraging Our Distributed Framework in UW–ASNs

In our framework, one of the nodes serves as the "master," i.e., it is responsible for parallelizing the tasks among transmitter/receiver pair *and* the nodes in the neighborhood, called service providers (SPs). We assume that only the communicating nodes are chosen to serve as SPs. Our scenario is shown in Fig. 6.

Self-organization in the distributed framework involves the selection of master and SPs. We assume that, for any communication transaction, one of the communicating nodes establishes itself as the master and all other nodes serve as SPs. Before the beginning of any communication session between the nodes, these selections are made. The master is made aware of the availability of SPs through voluntary service advertisements from the SPs themselves; such advertisements include information about the current position, amount of computing resources in terms of normalized CPU cycles, memory [in bytes], and communication [in bits per second], the start and end times of

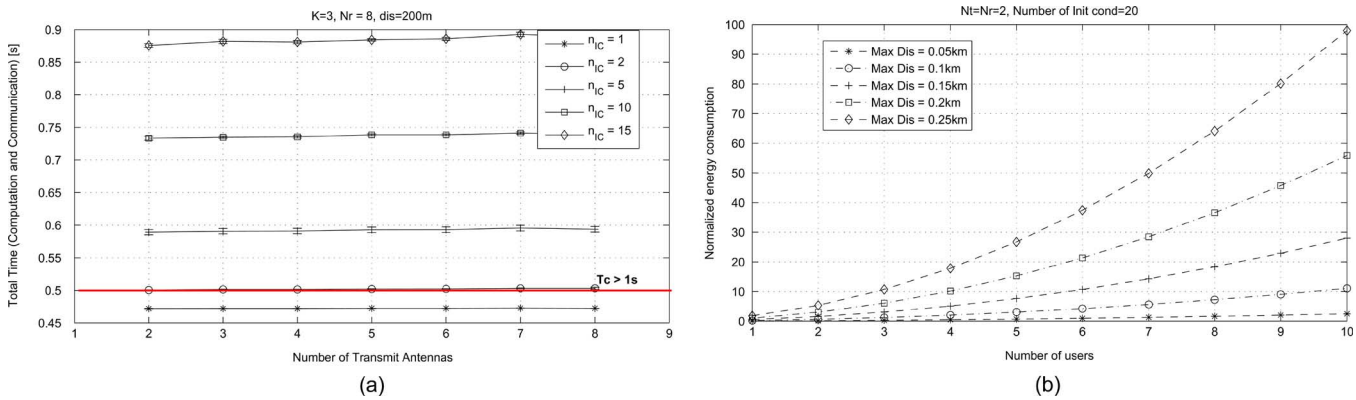


Fig. 5. (a) Total time taken for estimation of precoding/decoding matrices versus the number of transmit antennas by varying the number of initial conditions. (b) Energy expenditure for estimation of IA precoding matrices versus the number of users by varying the maximum distance between communicating nodes.

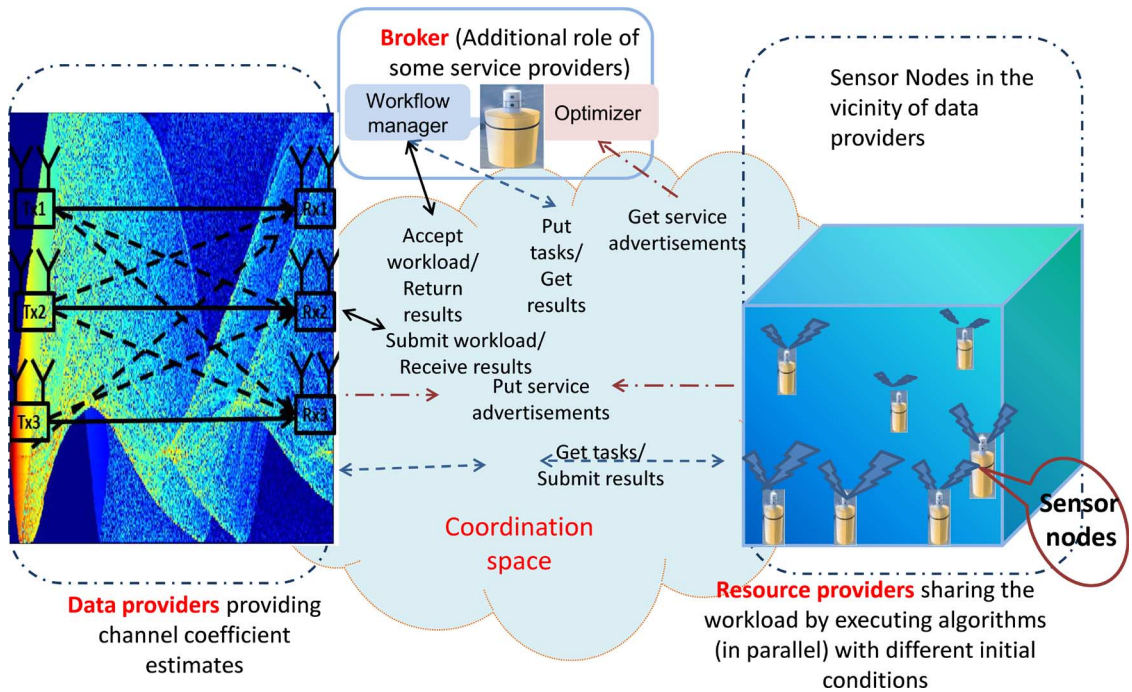


Fig. 6. Overview of the envisioned distributed computing framework for optimizing computation-intensive IA algorithms.

the availability of those resources, and the amount of residual energy at each SP. Using these values, the master will select a set of SPs to parallelize the task efficiently. In our solution, we assume that all the nodes that are communicating simultaneously serve as SPs. Once one of the nodes has established itself as the master, it broadcasts this information to all the nodes. The master does not change for a communication session, which may last several coherence-time intervals and may involve the exchange of several data packets.

Each communicating node estimates the channel from itself to all other communicating nodes and broadcasts this information. The master chooses equally spaced initial conditions for the iterative IA algorithm so as: 1) to ensure good coverage of the n-dimensional search space; and 2) to avoid choosing final precoding/decoding matrices "stuck" in local minima, hence, suboptimal. The master transmits a unique initial condition to each SP and the number of iterations required to execute the iterative algorithm. Once the SPs have computed the precoding matrices, they send their results to the master, which selects the vector pair that maximizes the network spectral efficiency. In this computing-grid framework, each node executes the iterative algorithm locally: this way, by incurring only small communication overhead caused by message exchanged among the nodes of the mobile grid, we are able to overcome the challenge of high communication delay. We have characterized the wireless computing grid as mobile because our computing devices, which form the grid, may consist of autonomous underwater vehicles (AUVs) along with static sensor nodes.

When using our distributed computing framework, the total coordination time to transmit and receive the precoding/decoding matrices changes from (4) to

$$T_{\text{coord}}^{\text{framework}} = 3 \cdot [T_p + T_t + (n_{\text{SP}} - 1)T_g] \tag{7}$$

where n_{SP} is the number of service providers. Note that, in our calculations, we assume that there are no packet errors or retransmission protocol in place. The factor of 3 takes into consideration the three sets of data exchange between the master and the nodes. In the first exchange of data, the master sends initial conditions to each node; in the second exchange of data,

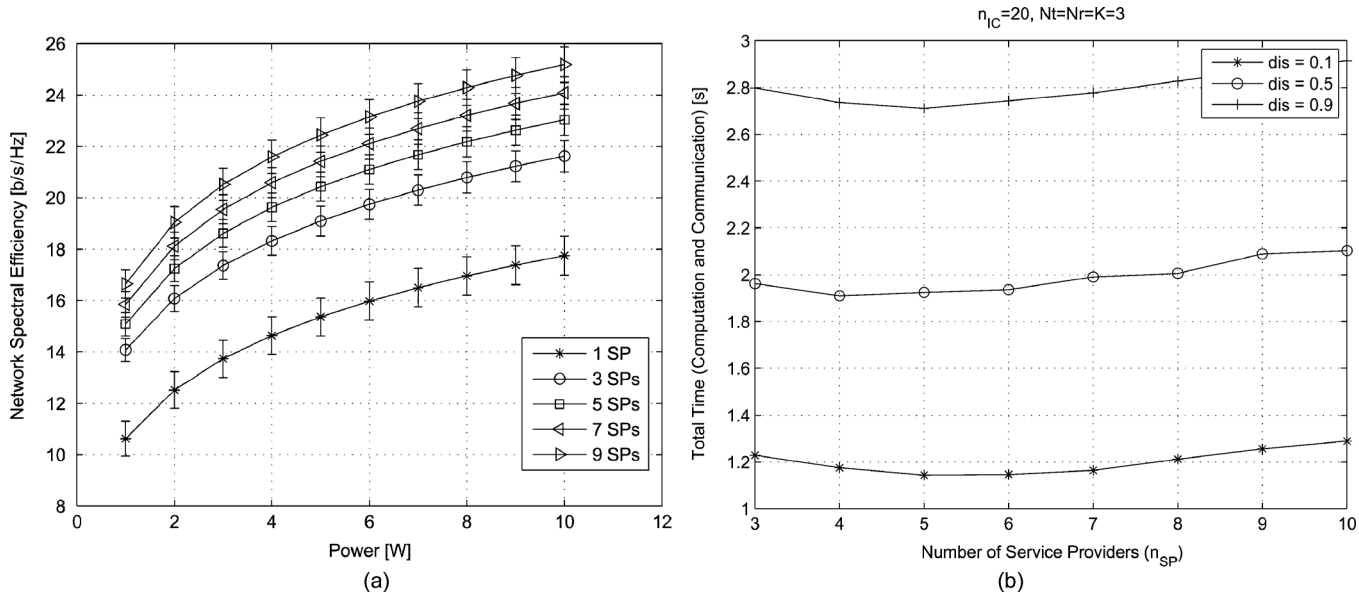


Fig. 7. (a) Variation of network spectral efficiency with power for different numbers of SPs for Algorithm 1. (b) Time taken by Algorithm 1 to compute precoding/decoding matrices by the distributed framework.

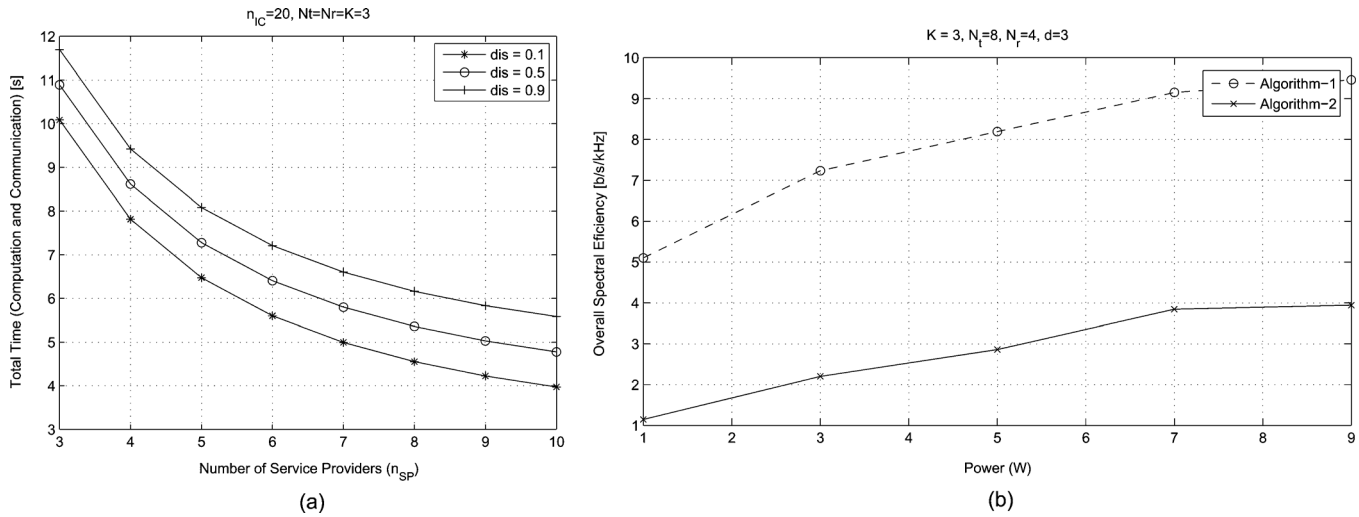


Fig. 8. (a) Time taken by Algorithm 2 to compute precoding/decoding matrices by the distributed framework. (b) Spectral efficiency gain achieved through ensemble learning.

the master receives the best set of precoding/decoding matrices from the communicating nodes; in the third and final stage of communication, the master sends the final precoding/decoding matrices selected from the set of nodes sent by the SPs. As the number of SPs increases from 1 to 3, the spectral efficiency increases by 5 b/s/Hz: so, for example, in a 20-kHz band system, the network capacity increases by 100 kb/s. The number of iterations for all service providers is assumed to be 10.

In Fig. 7(a), we see that, at a particular transmit power, as the number of SPs increases, the network spectral efficiency increases. Each SP estimates precoding/decoding vector using a unique initial condition and the master selects that pair of precoding/decoding matrices that gives the maximum spectral efficiency. With only one initial condition, the algorithm may get stuck in the local minima and give suboptimal results. By trying different initial conditions, we increase the probability of not getting stuck in a local minimum. Hence, in general, the higher the number of SPs, the higher is the network spectral efficiency.

In Fig. 7(b), we depict the total time taken for computation and communication of precoding/decoding matrices versus the number of SPs. We show this variation for different distances between communicating nodes. We see that in the beginning the time taken decreases as the number of SPs increases: e.g., for a distance equal to 0.5 km, the time taken decreases from 2.0 to 1.7 s as the number of SPs is increased from 3 to 5; however, if the number of SPs is increased beyond 5, the time taken increases. This indicates that, although the computation time has decreased by parallelizing the task of estimating IA matrices, the time for communication between nodes is high and the reduction in computation time does not compensate for the increase in communication cost.

V. Providing Robustness to IA

Several algorithms to estimate precoding/decoding matrices for IA have been proposed. Unfortunately, these matrices are NP-hard to obtain, and closed-form solutions have been found

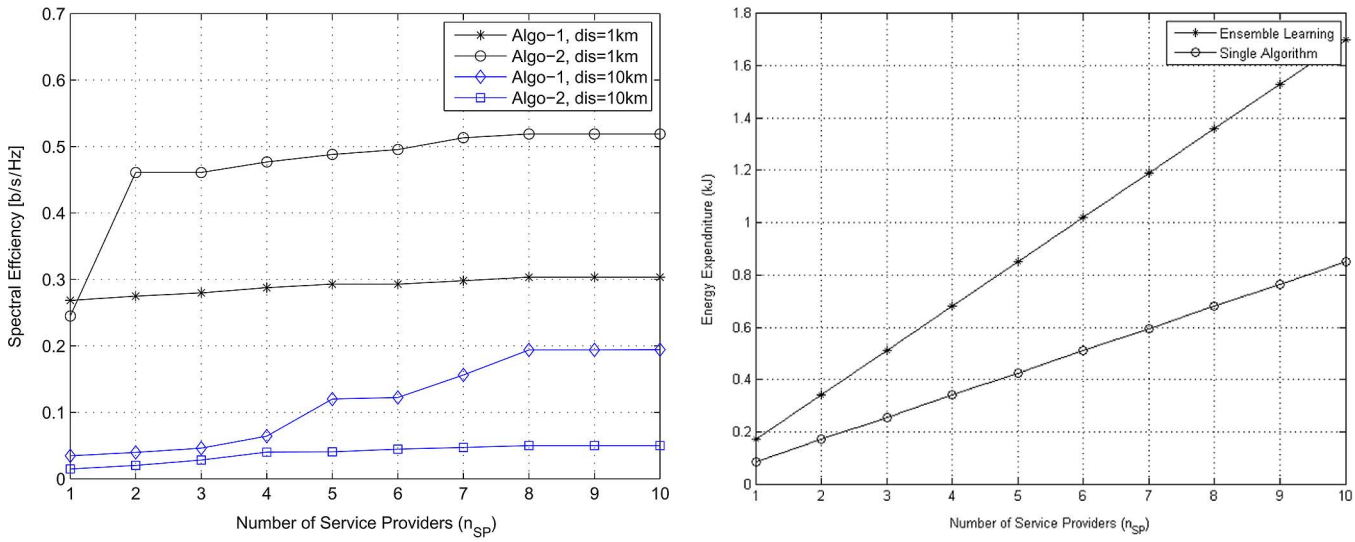


Fig. 9. (a) Performance gain in terms of spectral efficiency for different algorithms (Algorithm 1 [15] and Algorithm 2 [16]) via our distributed framework. (b) Energy expenditure of different algorithms executed via our distributed framework.

only for few special cases, such as in [3]. Even characterizing the feasibility of IA when perfect channel knowledge is assumed is a nontrivial task (see [20]). As an alternative to closed-form designs, heuristics have been proposed in the literature. In one of these algorithms [15] (which we refer to as Algorithm 1), Gomadam *et al.* aim at minimizing the interference leakage at each receiver so that, in the best case, interference alignment is perfectly attained. On the other hand, in [16] (which we refer to as Algorithm 2), the precoding/decoding matrices are estimated by minimizing the interference at each receiver

$$\min_{\{\mathbf{V}_l\}_{l=1}^{K}, l\neq k, \mathbf{U}_k} \quad rk(\mathbf{J}_k), \qquad \text{subject to: } rk(\mathbf{S}_k) = d$$

where $\mathbf{J}_k$ and $\mathbf{S}_k$ are the subspaces spanned by the desired and interference signals, respectively, at receiver k. Specifically, when $d = (N_T + N_R)/(K + 1)$, Algorithm 1 outperforms Algorithm 2; whereas, when $d > (N_T + N_R)/(K + 1)$, Algorithm 2 performs better. Both of these algorithms are *iterative* in nature and may be stuck in local minima while selecting the precoding/decoding matrices, which would lead to suboptimal results. Hence, we utilize "ensemble learning," i.e., we select the best precoding/decoding vector resulting from these different algorithms based on a given criterion, which, in our case, is the maximization of the spectral efficiency. Fig. 8(a) shows the time taken by Algorithm 2 to compute precoding/decoding matrices using our framework, while Fig. 8(b) gives the performance of the two algorithms in terms of spectral efficiency.

We have previously discussed that IA requires perfect channel knowledge at the nodes to estimate accurately the precoding/decoding vector so to achieve perfect alignment. We also pointed out that estimating precoding/decoding matrices requires substantial computation time due to eigenvector and matrix calculations. Hence, triggering IA every time the channel estimation is performed may be very inefficient if not prohibitive; also, it may render IA matrices unusable if the channel coherence time is small. For this reason, we make use of channel prediction for the estimation of precoding/decoding matrices, and present here a comparative study on how the performance gains vary with respect to the communication overhead for two competing strategies: 1) when we send the precoding matrices for each multiple of coherence time versus; and 2) when we send common precoding/decoding vector for all estimated channels.

A. Ensemble Learning

Ensemble learning is the process by which multiple models are strategically generated and combined to solve a particular computational intelligence problem [9]. Ensemble learning is primarily used to improve the (classification, prediction, function approximation, etc.) performance of a model, or to reduce the likelihood of an unfortunate selection of a poor one. We introduce ensemble learning in our distributed framework where different nodes work on different (competing) IA algorithms *simultaneously*. Both Algorithms 1 and 2 have the same communication overhead, which is the transmission of initial precoding/decoding matrices to the service providers. Our goal is to observe whether, by estimating precoding/decoding matrices through two different algorithms that incur the same communication overhead, we are able to gain in performance by incurring only a slight overhead increase for energy expenditure (through computation overhead as these different algorithms work on the same data). To achieve this goal, the framework tries to leverage the different performance benefits that these algorithms bring depending on different scenarios. Energy expenditure here is the energy spent at the service providers to compute the precoding/decoding matrices.

In Fig. 9(a), we see that each of the two algorithms performs differently in different domains, e.g., Algorithm 1 performs better than Algorithm 2 in terms of spectral efficiency when the distance between nodes is 10 km. Conversely, Algorithm 2 performs better than Algorithm 1 for internode distances equal to 1 km. The spectral efficiency gain using Algorithm 2 is around 60%; however, its energy consumption is higher by

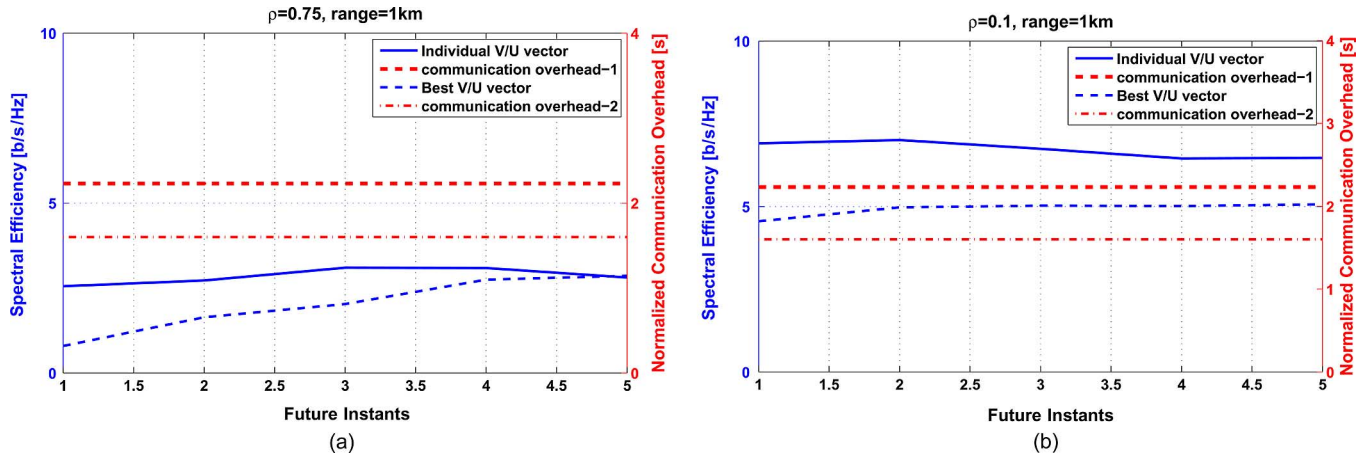


Fig. 10. Performance in terms of spectral efficiency (left y-axis in blue) and normalized communication overhead (right y-axis in red) with (a) correlation coefficient and range equal to 0.75 and 1 km, respectively, and maximum depth of 100 m; and (b) correlation coefficient and range equal to 0.1 and 1 km, respectively.

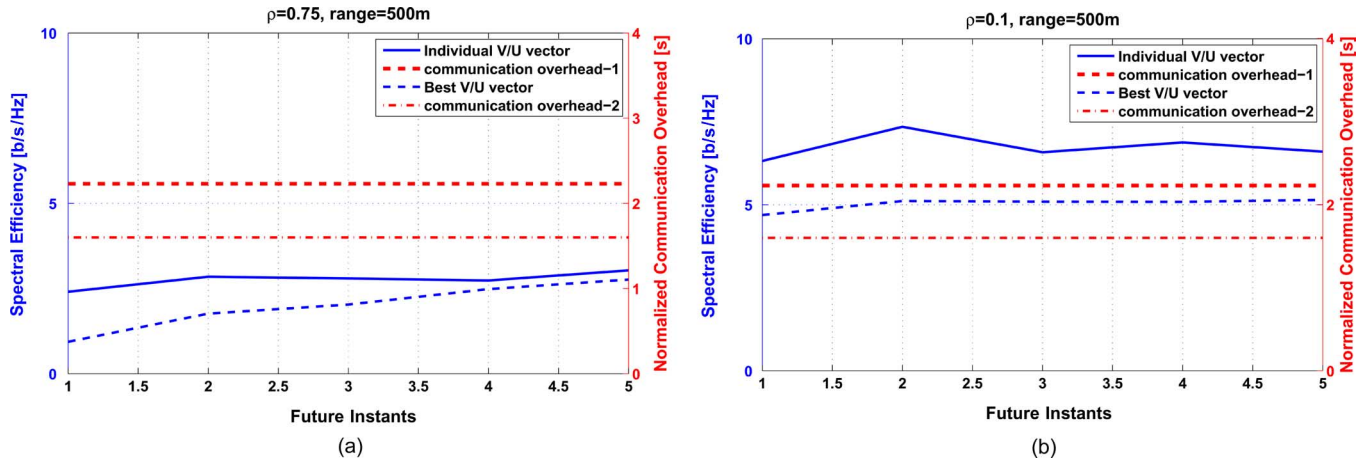


Fig. 11. Performance in terms of spectral efficiency (left y-axis in blue) and normalized communication overhead (right y-axis in red) with (a) correlation coefficient and range equal to 0.75 and 500 m, respectively, and maximum depth of 100 m; and (b) correlation coefficient and range equal to 0.1 and 500 m, respectively.

almost 50%. We observe that the performance of the two algorithms varies as the distance between nodes varies. In Fig. 9(b), we show the energy expenditure of the two algorithms when they are executed using our distributed computational framework.

B. Robustness Through Channel Prediction

We provide robustness to the IA technique against imperfect channel knowledge by leveraging on the time correlation between underwater acoustic channel coefficients. We use a simple autoregressive model to simulate the underwater channel, given by $\mathbf{X}_t = c + \sum_{i=1}^{p} \phi_i \mathbf{X}_{t-i} + \boldsymbol{\epsilon}_t$, where c is a constant, $\mathbf{X}_{t-i}$ represents the old channel values, p is the number of channel instants considered for prediction, ϵ_t is the white noise with normal distribution, zero mean, and finite variance, and the parameter ϕ represents the model coefficients estimated from previous p channel coefficients [21]. In our simulation, we predict channel coefficients for five consecutive coherence time intervals from the current instant. These channel coefficients are generated for topology 1 from the Bellhop channel model. Among the estimated precoding/decoding matrices, the pair that gives the highest spectral efficiency across all channel is selected. We provide a comparative study between spectral efficiency and communication overhead when unique precoding/decoding matrices are used at the transmitter side for each multiple of the coherence time versus when a common precoding/decoding matrix is selected. The comparison for both these cases is in terms of performance and communication overhead.

Fig. 10(a) shows the performance when correlation between channel coefficients of two consecutive coherence time intervals is 0.75 and the internode distance is 1 km. The x-axis of the figure shows the time difference between the current and future instants for which the estimation of IA vectors is performed. The y-axis on the left-hand side (in blue) shows the spectral efficiency for two scenarios: 1) when a unique precoding/decoding matrices for each channel instant is used; and 2) when a common precoding/decoding matrix is used. The y-axis on the right-hand side (in red) shows the normalized communication cost for these two cases. In Fig. 10(b), we see the performance of IA through predicted channel coefficients when the correlation between channel coefficients at consecutive coherence time is 0.1. Fig. 11 depicts the same analysis when the distance between communicating nodes is 500 m.

The inference from Figs. 10 and 11 is that the performance (in terms of spectral efficiency) of common precoding/decoding matrix is poorer than unique matrices at all instants. In addition

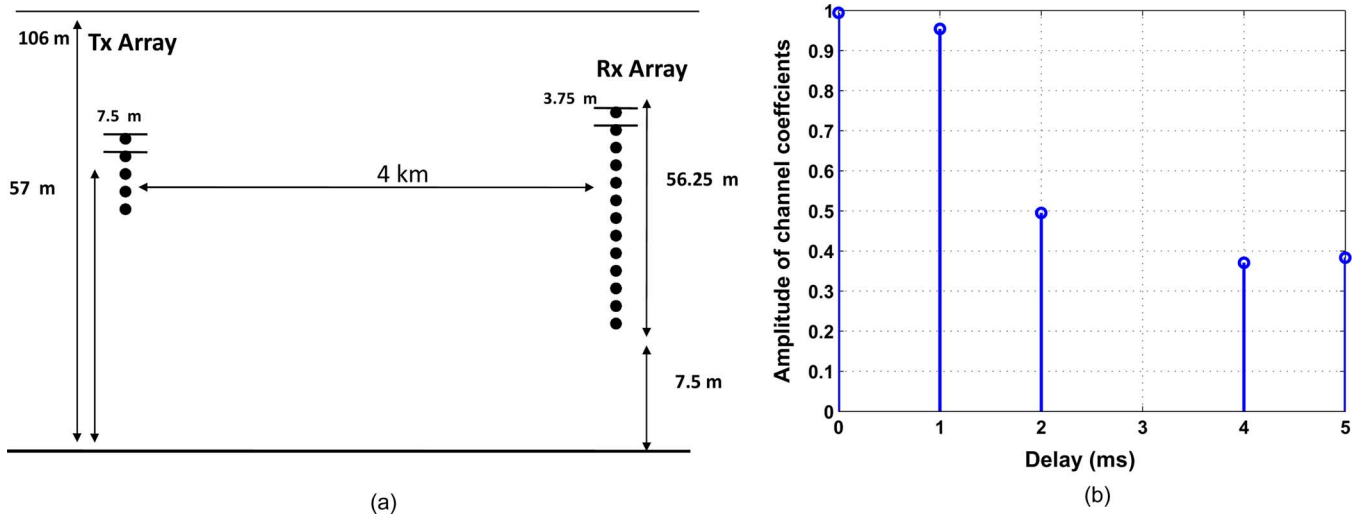


Fig. 12. (a) Channel geometry for the KAM08 experiments. (b) CIR of a receiver at a fixed instant.

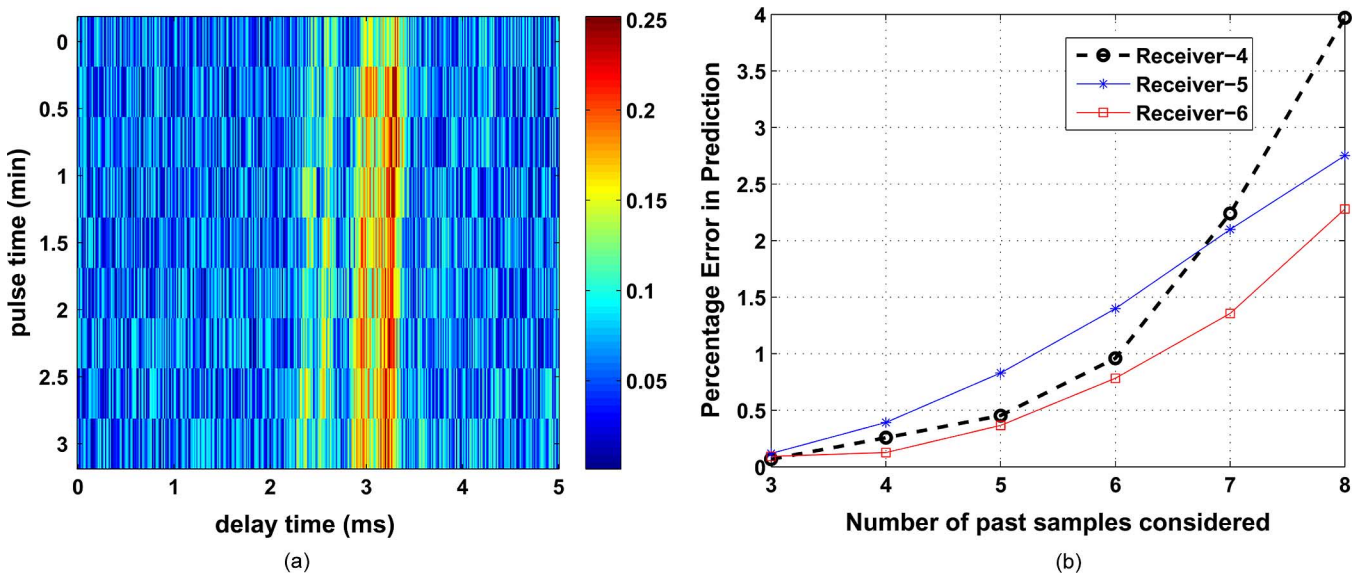


Fig. 13. (a) Temporal evolution of the channel response at 80-m depth with the source at 57-m depth. (b) Performance of one-step prediction of the channel as the number of past samples is varied.

to that, the loss in performance of common matrices with respect to unique matrices is higher when the correlation coefficient is lower, i.e., the loss in performance of common matrices with respect to unique matrices is higher when ρ is 0.1 in comparison to when it is 0.75. The communication overhead for both common and unique matrices is roughly the same.

VI. Evaluation With Experimental Data

The focus of this section is to use real-data acoustic channel values to study the performance of prediction algorithm in [22] to predict the underwater channel impulse response (CIR). Given the complexity of the underwater environment, this analysis will help us understand the feasibility of an algorithm to predict CIRs, which can be used for estimation of future precoding/decoding vectors.

In Section II, we studied that when the communicating nodes do not have updated channel information, the interfering signals no longer align perfectly. As a result, the existing interference suppression filters are no longer able to cancel interference, which leads to leakage interference. Hence, knowledge of the channel plays an important role in interference alignment. The better the prediction of channel coefficients, the lower is the leakage interference. We observed that the algorithm introduced in [22] performed better than the autoregressive model to predict future CIRs for data sets from the Kauai Acoustic communications MURI 2008 experiment (KAM08) carried out in shallow water from June 16 to July 2, 2008, west of Kauai, HI, USA. The prediction using autoregressive model was poor because, for this model, we assumed a constant correlation coefficient between the channel coefficients. This, however, might not be the case with the real data. Hence, the prediction algorithm in [22] is used to find one-step prediction values of the channel for real data. Although this algorithm was initially developed for terrestrial channels, its generality makes it applicable to the underwater environment. As seen earlier, this allows us to choose between unique or common precoding/decoding vectors and to reduce the communication overhead at the cost of marginal reduction in spectral efficiency. The discrete underwater channel

is formed by sampling the continuous channel with sampling period T, i.e., $h_d(n) = h(nT)$ and $P(t) = \sum_{i=0}^{M-1} c_i t^i$, where $P(t)$ is the polynomial fitted to previous CIR samples (M). This polynomial is then extrapolated to predict the future channel state. Parameter c_i is estimated from past channel values. This analysis is done for both real and imaginary parts of the channel. This algorithm considers the local characteristics of the channel; hence, to bound the estimation error, the past values considered cannot span a long range. In Section IV-B, we leveraged our distributed framework to estimate the precoding/decoding vectors for a group of concurrently communicating nodes. Similarly, the framework can be leveraged to estimate CIRs for a number of future channel instants.

We show experimental results from the data collected during the KAM08 experiment [23]. The schematic of the KAM08 experiment is shown in Fig. 12(a). The receiver array is moored in about 106-m water at 4 km from an 8-element source array. The middle element at 57-m depth was chosen as a single transmitter during the communications experiment for the data used in this paper. The vertical arrays consisted of 16 elements spanning a 56.25-m aperture with 3.75-m element spacing. The coherence time calculated for this data set is approximately 100 ms. Hence, this is a fast varying channel. In Fig. 12(b), we show the CIR for a receiver. The vertical axis indicates amplitude of channel coefficients, and the horizontal axis shows time delay. In Fig. 13(a), we plot the temporal evolution of the channel for one of the receivers.

In Fig. 13(b), we show the performance of the prediction algorithm for one-step prediction of the underwater channel. As explained earlier, the performance of the algorithm depends on the number of past values considered, and we see that as the number of past values increases, the estimation error also increases. Given the same number of past values, as expected, the estimation error is higher when the channel is changing faster.

VII. Conclusion

We discussed various parameters such as the number of users, the number of antennas, and the distance between nodes to decide the feasibility of the IA technique (in terms of the time taken) underwater, given the coherence time of the wireless acoustic channel. We also studied the energy expenditure incurred for transmission of IA matrices between communicating nodes. We proposed a distributed computing framework to overcome the challenge of computational complexity and communication delay faced by IA algorithms. We observed that parallelism is not only possible (notwithstanding the large acoustic propagation delays underwater), but, in fact, increases the region of feasibility. We also presented solutions to provide robustness to IA through ensemble learning, i.e., the execution of multiple competing algorithms in parallel. We introduced robustness by predicting the channel to estimate the precoding/decoding matrices. We also provided a comparative study between spectral efficiency and communication overhead when unique precoding/decoding matrices are used at the transmitter side for each of the predicted channel coefficients versus when a common precoding/decoding matrix is used. We studied the underwater channel using data sets from underwater experiments and tested the performance of a prediction algorithm to predict the channel impulse response is some underwater environments. As a future work, we plan to implement our proposed solution on an emulator using real acoustic modems, initially with the underwater acoustic channel simulated using the Bellhop model.

References

[1] I. F. Akyildiz, D. Pompili, and T. Melodia, "Underwater acoustic sensor networks: Research challenges," *Ad Hoc Netw.*, vol. 3, no. 3, pp. 257–279, May 2005.

[2] S. Roy, T. M. Duman, V. McDonald, and J. G. Proakis, "High-rate communication for underwater acoustic channels using multiple transmitters and space-time coding: Receiver structures and experimental results," *IEEE J. Ocean. Eng.*, vol. 32, no. 3, pp. 663–688, Jul. 2007.

[3] V. Cadambe and S. Jafar, "Interference alignment and degrees of freedom of the K-user interference channel," *IEEE Trans. Inf. Theory*, vol. 54, no. 8, pp. 3425–3441, Aug. 2008.

[4] A. Ghasemi, A. Motahari, and A. Khandani, "Interference alignment for the K-user MIMO interference channel," in *Proc. IEEE Int. Symp. Inf. Theory*, Austin, TX, USA, Jun. 2010, pp. 360–364.

[5] M. Chitre, M. Motani, and S. Shahabudeen, "Throughput of networks with large propagation delays," *IEEE J. Ocean. Eng.*, vol. 37, no. 4, pp. 645–658, Oct. 2012.

[6] O. El Ayach, S. Peters, and R. Heath, "The feasibility of interference alignment over measured MIMO-OFDM channels," *IEEE Trans. Veh. Technol.*, vol. 59, no. 9, pp. 4309–4321, Nov. 2010.

[7] C. Suh and D. Tse, "Interference alignment for cellular networks," in *Proc. Annu. Allerton Conf. Commun. Control Comput.*, Urbana-Champaign, IL, USA, Nov. 2008, pp. 1037–1044.

[8] S. Gollakota, S. Perli, and D. Katabi, "Interference alignment and cancellation," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 39, no. 4, pp. 159–170, Aug. 2009.

[9] G. Brown, J. Wyatt, R. Harris, and X. Yao, "Diversity creation methods: A survey and categorisation," *Inf. Fusion*, vol. 6, no. 1, pp. 5–20, Apr. 2005.

[10] S. Jafar and M. Fakhereddin, "Degrees of freedom for the MIMO interference channel," *IEEE Trans. Inf. Theory*, vol. 53, no. 7, pp. 2637–2642, Jul. 2007.

[11] S. Jafar and S. Shamai, "Degrees of freedom region of the MIMO X channel," *IEEE Trans. Inf. Theory*, vol. 54, no. 1, pp. 151–170, Jan. 2008.

[12] P. Qarabaqi and M. Stojanovic, "Small scale characterization of underwater acoustic channels," in *Proc. ACM Int. Workshop UnderWater Netw.*, Los Angeles, CA, USA, Nov. 2012, DOI: 10.1145/2398936.2398946.

[13] M. Porter, "Bellhop code," [Online]. Available: http://oalib.hlsresearch.com/Rays/HLS-2010-1.pdf

[14] P. Pandey and D. Pompili, "On the region of feasibility of interference alignment in underwater sensor networks," in *Proc. IEEE Underwater Commun., Channel Model. Validation*, La Spezia, Italy, Sep. 2012.

[15] K. Gomadam, V. Cadambe, and S. Jafar, "A distributed numerical approach to interference alignment and applications to wireless interference networks," *IEEE Trans. Inf. Theory*, vol. 57, no. 6, pp. 3309–3322, Jun. 2011.

[16] D. Papailiopoulos and A. Dimakis, "Interference alignment as a rank constrained rank minimization," in *Proc. IEEE Global Telecommun. Conf.*, Miami, FL, USA, Dec. 2010, DOI: 10.1109/GLOCOM.2010.5684037.

[17] A. Valera, P. W. Q. Lee, H.-P. Tan, H. Liang, and W. K. G. Seah, "Implementation and evaluation of multihop ARQ for reliable communications in underwater acoustic networks," in *Proc. OCEANS Conf.*, Bremen, Germany, May 2009, DOI: 10.1109/OCEANSE.2009.5278117.

[18] T. C. Yang, "Toward continuous underwater acoustic communications," in *Proc. OCEANS Conf.*, Quebec City, QC, Canada, Sep. 2008, DOI: 10.1109/OCEANS.2008.5151827.

[19] H. Viswanathan, E. K. Lee, and D. Pompili, "An autonomic resource provisioning framework for mobile computing grids," in *Proc. Int. Conf. Autonom. Comput.*, San Jose, CA, USA, Sep. 2012, pp. 79–84.

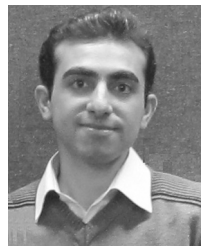
[20] C. Yetis, T. Gou, S. Jafar, and A. Kayran, "Feasibility conditions for interference alignment," in *Proc. IEEE Global Telecommun. Conf.*, Honolulu, HI, USA, Nov. 2009, DOI: 10.1109/GLOCOM.2009.5425326.

[21] H. Akaike, "Fitting autoregressive models for prediction," *Ann. Inst. Stat. Math.*, vol. 21, no. 1, pp. 243–247, Dec. 1969.

[22] Z. Shen, J. G. Andrews, and B. L. Evans, "Short range wireless channel prediction using local information," in *Conf. Record 37th Asilomar Conf. Signals Syst. Comput.*, Monterey, CA, USA, Nov. 2003, DOI: 10.1109/ACSSC.2003.1292104.

[23] H. C. Song, W. S. Hodgkiss, and P. A. van Walree, "Phase-coherent communications without explicit phase tracking," *J. Acoust. Soc. Amer.*, vol. 128, no. 3, pp. 969–972, Sep. 2010.

Parul Pandey (S'07) received the B.S. degree in electronics and communication engineering from Indira Gandhi Institute of Technology, Delhi, India, in 2008 and the M.S. degree in electrical and computer engineering, mentored by Dr. A. Bolton, from the University of Utah, Salt Lake City, UT, USA, in 2011. Currently, she is working toward the Ph.D. degree under the guidance of Dr. D. Pompili at the NSF Center for Cloud and Autonomic Computing, Department of Electrical and Computer Engineering (ECE), Rutgers—The State University of New Jersey, Piscataway, NJ.



Mohammad Hajimirsdeghi received the B.S. degree in electrical engineering (with honors, first rank) from Shahed University, Tehran, Iran, in 2008 and the M.S. degree in telecommunications from Sharif University of Technology (SUT), Tehran, Iran, in 2011. Currently, he is working toward the Ph.D. degree in electrical and computer engineering at Rutgers—The State University of New Jersey, Piscataway, NJ.

His research interests include wireless communication and networking. Before coming to the United States, he was a Communication Engineer at Iran Telecommunication Research Center (ITRC), Tehran, Iran.

Dario Pompili (S'04–A'07) received the "Laurea" (integrated B.S. and M.S.) and Ph.D. degrees in telecommunications and system engineering from the University of Rome "La Sapienza," Rome, Italy, in 2001 and 2004, respectively, and the Ph.D. degree in electrical and computer engineering from Georgia Institute of Technology, Atlanta, GA, USA, in June 2007.

He is an Associate Professor with the Department of Electrical and Computer Engineering (ECE), Rutgers—The State University of New Jersey, Piscataway, NJ. He is the Director of the Cyber Physical Systems Laboratory (CPS-Lab), which focuses on research problems in mobile computing, wireless communications and networking, sensor networks, and datacenter management.

Dr. Pompili is a recipient of the prestigious 2011 National Science Foundation (NSF) CAREER Award, the 2012 Office of Naval Research (ONR) Young Investigator Program Award, and the 2012 Defense Advanced Research Projects Agency (DARPA) Young Faculty Award.