Decentralized Extended Information Filter for Single-Beacon Cooperative Acoustic Navigation: Theory and Experiments

Sarah E. Webster, *Member, IEEE*, Jeffrey M. Walls, *Student Member, IEEE*, Louis L. Whitcomb, *Fellow, IEEE*, and Ryan M. Eustice, *Senior Member, IEEE*

***Abstract*—We report a decentralized extended information filter (DEIF) algorithm designed for single-beacon cooperative acoustic navigation of one or more client underwater vehicles. In single-beacon cooperative acoustic navigation, ranges and state information from a single reference source (the server) are used to improve localization and navigation of an underwater vehicle (the client). The ranges and state information are obtained using underwater acoustic modems and a synchronous-clock time-of-flight paradigm. Apart from the server's acoustic data broadcasts, the client has no access to the server's position or sensor measurements. We show that at the instance of each range measurement update, the DEIF algorithm yields identical results for the current vehicle state estimate as the corresponding centralized extended information filter (CEIF), which fully tracks the joint probability distribution between the server and client. We compare the state estimation results of the DEIF algorithm with that of a CEIF and three other filters reported in the literature. The evaluation is performed using both simulated data and an experimental dataset comprised of one surface craft and two autonomous underwater vehicles.**

***Index Terms*—Decentralized estimation, distributed robot systems, information filters, marine robotics, networked robots.**

I. Introduction

THIS paper reports the derivation, simulation, and experimental evaluation of a decentralized extended information filter (DEIF) algorithm for single-beacon cooperative acoustic navigation. The DEIF is designed for use within an acoustic underwater navigation paradigm using synchronous clocks and one-way-travel-time (OWTT) measurements [1], [2], although the DEIF's formulation may have applicability in other low-throughput, distributed estimation applications outside of this reported scenario.

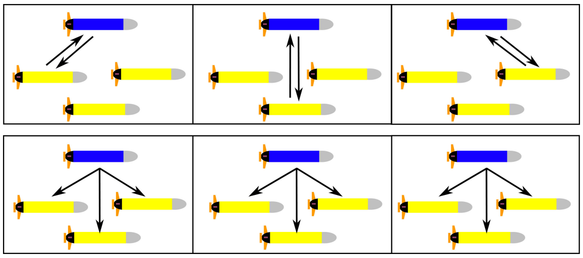


Fig. 1. The minimum time between updates for each client using two-way travel-time (TWTT) ranging (upper panel) is the cumulative sum of the TWTT for each client. In contrast, the minimum time between updates for each client using OWTT ranging (lower panel) remains constant, regardless of the number of clients.

In the subsea domain, our goal is to enable high-precision absolute navigation of underwater vehicles without expensive sensor suites, and for missions with length scales on the order of 100 km. As the cost of underwater vehicles has decreased, it has become tractable to deploy multiple vehicles to collect scientific data in the world's oceans. Deployments of multiple low-cost vehicles enable the collection of datasets over length scales too large to be covered by a single vehicle or too dangerous to risk an expensively instrumented vehicle. However, the requirement to estimate the position of multiple vehicles simultaneously, especially vehicles without expensive, state-of-the-art navigation sensors, provides a challenge for existing navigation systems.

Most underwater acoustic navigation systems are narrowband and are based upon measuring two-way travel-time (TWTT) time-of-flight ranges [3], [4]. In these systems, each vehicle must interrogate the acoustic network in order to obtain a time-of-flight measurement between it and all replying nodes. As a result, as illustrated in Fig. 1, the rate at which multiple vehicles can receive navigation updates decreases linearly as the number of navigated vehicles in the water increases. In comparison, OWTT ranging can be determined by knowing precisely the transmit and receive times of an underwater acoustic telemetry packet. The result is a direct one-way time-of-flight measurement from source to receiver. Thus, when a server node broadcasts to the network, all receiving client nodes can passively measure their one-way time-of-flight to the server node. The advantage over TWTT ranging is that OWTT ranging readily scales to a multi-vehicle environment within a server/client architecture, because the overall update rate for each client

Manuscript received November 30, 2012; accepted March 7, 2013. This paper was recommended for publication by Associate Editor J. A. Castellanos and Editor W. K. Chung upon evaluation of the reviewers' comments. This work was supported in part by the National Science Foundation under Award IIS-0746455, Award ANT-1039951, Award ATM-0427220, and Award IIS-0812138.

S. E. Webster is with the Applied Physics Laboratory, University of Washington, Seattle, WA 98105 USA (e-mail: swebster@apl.washington.edu).

J. M. Walls is with the Department of Mechanical Engineering, University of Michigan, Ann Arbor, MI 48109 USA (e-mail: jmwalls@umich.edu).

L. L. Whitcomb is with the Department of Mechanical Engineering, The Johns Hopkins University, Baltimore, MD 21218 USA (e-mail: llw@jhu.edu).

R. M. Eustice is with the Department of Naval Architecture and Marine Engineering, University of Michigan, Ann Arbor, MI 48109 USA (e-mail: eustice@umich.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/TRO.2013.2252857

1552-3098/$31.00 © 2013 IEEE

remains constant. The disadvantage is increased complexity in the hardware design because all nodes must carry synchronized stable clock hardware.

In our work, we employ the Woods Hole Oceanographic Institution (WHOI) Micro-Modem, i.e., an underwater acoustic modem capable of synchronous-clock transmission [5], [6]. This synchronous-clock feature allows the Micro-Modem to directly and accurately measure time of arrival (TOA) between a source and receiver when using a user-supplied external reference clock. This common time base allows for a synchronous modem communication/navigation system, whereby acoustic telemetry broadcasts can encode time of origin information as well as local state information. Acoustic telemetry packets can be broadcast to the vehicle network, allowing all receiving nodes to passively measure their OWTT to the source node. The OWTT-derived range knowledge, when used in conjunction with the decoded acoustic telemetry data and other onboard vehicle navigation data, provides a mechanism to enable cooperative acoustic navigation.

We have previously reported centralized algorithms for synchronous-clock, OWTT, cooperative, acoustic navigation and evaluated their performance using experimental data from both shallow-water [2], [7] and deep-water field trials [8], [9]. In this paper, we present a novel decentralized algorithm, evaluate its performance with experimental data, and compare the results with other commonly used estimation frameworks. Within the context of single-beacon navigation, the decentralized approach provides a flexible, scalable solution for vehicle navigation. Navigation algorithms that rely on a centralized observer suffer from the severely limited bandwidth and high latency associated with underwater acoustic communication in comparison with typical land-based radio frequency communication networks [10]. Given the speed of sound in water ($\sim$1500 m/s), transmitting acoustic data over ranges on the order of kilometers results in latency on the order of seconds. Although the useful channel capacity of acoustic modem technology has increased dramatically in recent years, achieving throughput of up to 5000 bps [5], operationally the average throughput is on the order of 10–50 bps due to the low duty cycle with which these messages are typically transmitted during deep-water at-sea operation [11]. The proposed DEIF is well suited to single-beacon navigation over limited-capacity and high-latency underwater acoustic telemetry because this distributed approach only requires a small, fixed quantity of information to be transmitted from the server to the client.

The contributions of this study build primarily on our previous work [12], [13] and include the following:

1) a detailed theoretical derivation and analysis of the DEIF algorithm;
2) a comprehensive analysis of the DEIF algorithm based on simulation and real-world experimental data;
3) a thorough comparison of the DEIF algorithm with other OWTT navigation frameworks found in the literature and used throughout the community.

The DEIF algorithm reported herein consists of two parts. The first runs on the server, which is assumed to be a moving vehicle (a ship or another underwater vehicle, for example), but the server could also be a fixed beacon. The second part is designed to run locally on a submerged vehicle, which is referred to as the client, with real-time access to the client's own onboard navigation sensors and infrequent, asynchronous reception of acoustic broadcasts from the server vehicle. Notably, the client does *not* have access to real-time measurements from the server, but only information that is contained in the server's acoustic broadcasts. Note that we characterize this filter as "decentralized," as opposed to "distributed" because we do not employ a fusion center [14].

The rest of this paper is organized as follows: Section II describes previous work in several relevant areas. Section III presents the derivation of the DEIF and shows that it produces identical results to that of a centralized extended information filter (CEIF) (the dual equivalent of a centralized extended Kalman filter) immediately following each range measurement update. Sections IV and V present implementation details and results evaluating the performance of the DEIF and other reported approaches, using both simulated data and real-world experimental data. This comparison is performed across several operational scenarios, including when significant mutual correlation exists between the server and the client. Section VI discusses practical considerations of the DEIF algorithm, and Section VII concludes this paper.

II. Previous Work

The results in decentralized single-beacon cooperative navigation reported in this paper are informed by several areas of research, discussed below: single-beacon underwater navigation, decentralized multi-robot cooperative localization, decentralized estimation in the context of multiple underwater vehicles, and the use of the information filter for navigation in the general field of mobile robotics.

A. Single-Beacon Underwater Navigation

Single-beacon navigation relies on range measurements from a single beacon, which is referred to as the server, to provide a position reference to one or multiple client vehicles. The server's position estimate is typically more accurate than that of the clients' either through access to continuous or intermittent georeferenced position information, e.g., from a global positioning system (GPS), or because the server has high fidelity sensors that allow for more accurate dead-reckoning than the clients, e.g., a high-grade inertial measurement unit. The observability requirements of such server–client networks are explored in [15]–[17]. Any form of navigation that relies solely on range observations for localization is fundamentally limited by the geometry of the source and receiver. However, the relative observability of the two vehicles can inform an intelligent server control strategy, as in [18], to drive down uncertainty in the client.

Navigation with a single, fixed beacon, whose position is known *a priori*, has been reported using several different estimation techniques—a least-squares approach by Scherbatyuk [19] and Baccou and Jouvencel [20], and a vehicle-based extended Kalman filter (EKF) by Larsen [21] and Gadre and Stilwell [16]. Navigation with respect to a moving reference beacon, whose position is not known *a priori*, is reported by McPhail and Pebody [22] using a nonlinear least-mean-squares method; by Eustice *et al.* [2], [7] using a maximum likelihood estimation method; and by Webster *et al.* [8], [9] using a centralized extended Kalman filter (CEKF). As reported, each of these methods is only structurally tractable for post-processing, although the authors of [2], [7], and [22] suggest improvements that would allow for real-time implementations. Morice and Veres [23] report geometric bounding techniques that are applicable in real time, and simulation results for range-based underwater navigation. See [8] for an extensive review of single-beacon navigation.

B. Cooperative Multi-robot Localization

As fielding teams of robots with complementary sensor characteristics has become more practical, research in decentralized, multi-robot cooperative localization has intensified. Roumeliotis and Bekey [24] present a distributed EKF solution that tracks the global state composed of all robot positions. Other distributed approaches incorporating relative robot observations include particle filters by Howard [25], and graph-based methods by Kim *et al.* [26] and Indelman *et al.* [27]. These methods require unconstrained bandwidth, a nonlossy communication channel, or both. These can be achieved in post-processing, but are impractical in real time for an underwater acoustic network given the restrictions of the underwater acoustic channel [28].

Ribeiro *et al.* [29] and Nerurkar *et al.* [30] perform distributed cooperative localization only requiring the transmission of a single bit per measurement within their sign-of-innovation Kalman filter. This quantized method is able to construct an approximation of the centralized filter onboard each platform, although is not tolerant of a lossy channel.

A distributed information filter-based approach for multi-vehicle cooperative localization is proposed by Bailey *et al.* [31]. Each vehicle in the network maintains a local pose-graph (a graph where edges are constraints and nodes are poses) that fuses local measurements with observations of the relative-pose to other vehicles. A central fusion center then processes local pose-graphs and relative-pose observations to estimate the full joint distribution over all vehicle poses. To efficiently transmit local pose-graphs to the central server, the authors partition the local pose-graph into segments according to a “product-rule decomposition.” When the platform transmits new nodes within its pose-graph to a fusion center, this interplatform message composition is equivalent to the delta information packets used herein and first presented by Webster *et al.* [12]. The DEIF algorithm presented herein is similar to the algorithm in [31] applied to the two-vehicle unidirectional communication topology. Our client vehicle, however, runs a single filter, whereas [31] maintains a separate client filter and fusion center.

Cooperative localization has also been addressed within the larger framework of decentralized data fusion—see [32], for example, and [14]. In the latter, the authors present a system that employs an information filter that uses a distributed Cholesky modification for delayed states. This enables individual clients to share their local estimates within the network without the use of a centralized fusion center. Another research area that is related to multi-robot localization is the use of consensus algorithms, for tasks such as cooperative multi-robot mapping [33].

C. Decentralized Underwater Multi-vehicle Navigation

Decentralized estimation in the context of underwater communication and navigation faces unique constraints in terms of low data rates and high latency, which renders many of the decentralized estimation solutions from terrestrial applications unsuitable for use with underwater vehicles.

An approach to decentralized underwater navigation, which is reported by Maczka *et al.* [34], treats each range measurement as originating from an independent source (to minimize acoustic data telemetry) and fuses them in a Kalman filter ignoring any correlation between the sender and receiver. The authors refer to this method as the egocentric extended Kalman filter (EEKF), because it does not consider possible correlation between the navigation estimates of each vehicle. This method has modest data telemetry requirements, is robust to packet loss, and easily scales to large networks; however, ignoring correlation among the vehicles can cause overconfidence and divergence, as is noted [34], which states: “The complete system behavior when the cross covariance terms P_{ij} are neglected is of great interest. These terms represent the information that is common between vehicles i and j. Neglecting them results in larger Kalman gains for each vehicle, overly optimistic covariance calculations, and could result in the divergence of the Kalman filter. . ..”

Bahr *et al.* [35] address cooperative localization of multiple underwater and surface vehicles using a bank of decentralized vehicle-based filters. This study reports the use of multiple surface vehicles with access to GPS to provide range measurements to the underwater vehicles. An acoustic broadcast from a surface vehicle encodes the mean and covariance estimate of that vehicle, and multiple receiving vehicles use the acoustic broadcast to perform a range measurement update. In order to avoid overconfidence, the authors present a bookkeeping approach, which is referred to as the interleaved update (IU) algorithm, in which each vehicle runs multiple Bayes estimators that track the source of each range measurement.

In a separate work, Bahr *et al.* [36] present a multi-vehicle cooperative localization method that, as an expansion of the moving long baseline concept proposed by Vaganay *et al.* [37], encompasses multiple range sources and real-time operations. Fallon *et al.* [38] generalize [36] to consider navigation in the context of a single range source. Similar to our work, Fallon *et al.* [38] rely on a single, moving, georeferenced server to support the localization of multiple vehicles through asynchronous acoustic broadcasts. The authors present a client-based EKF and a minimization strategy that uses current and historic server positions, ranges between client and server positions (both current and historic), and the distance traveled by the client between range measurements. Like in the EEKF algorithm [34], the server and clients are assumed to be uncorrelated, and range measurement updates are performed using the absolute position and covariance of the server. One of the benefits of the formulation in [38] is that the algorithm is trivially robust to packet loss, although ignoring the correlation between clients and server may lead to an overconfident estimate of the client position, as noted in [34].

In a separate work, Fallon *et al.* [39] present a measurement distribution framework that allows multiple platforms to share navigation information (including intervehicle range measurements) to enable a fully consistent distributed solution. The framework is based upon each node locally recreating the state of the entire system using range measurements between nodes and knowledge of each node's dead-reckoning between successive range measurements. The method provides a bookkeeping mechanism that relies on acknowledgments between vehicles to eventually provide all of the requisite information to each node. Once a node has collected information for the entire system up to a certain time (range measurements and dead-reckoning information for all nodes), a fully consistent cooperative navigation solution can be calculated. The convergence rate of this algorithm is sensitive to packet loss, as it requires the retransmission of data packets that are not acknowledged.

D. *Extended Information Filter for Navigation and SLAM*

The Kalman filter in the inverse covariance form, which is known as the information filter, has several properties that make it a good candidate for navigation in multi-robot applications. Derived in detail by Mutambara [40], the extended information filter (EIF) has been employed in distributed form for terrestrial robot applications such as vehicle navigation by Bozorg *et al.* [41], simultaneous localization and mapping (SLAM) by Thrun and Liu [42] and Reece and Roberts [43], and multi-robot localization by Bailey *et al.* [31] (as described earlier). The EIF has also been used successfully within decentralized estimation for linear systems with known network topologies by Grime *et al.* [44]. In the context of underwater vehicles, there are examples of the EIF being employed in SLAM algorithms by Eustice *et al.* [45] and in a distributed fashion by Diosdado and Ruiz [46].

Nonlinear smoothing algorithms have become popular in the SLAM community due to advances in sparse linear-algebra techniques [47], [48]. A delayed-state filtering SLAM framework, such as [45] and the work reported herein, is also a smoothing framework over the delayed states. The key difference being that nonlinear smoothing algorithms periodically relinearize constraints, whereas a filter linearizes only once. Fallon *et al.* [49] present an underwater navigation algorithm fusing independent OWTT observations from a surface vehicle with side-scan in a nonlinear incremental smoothing and mapping framework (iSAM). In practice, the use of nonlinear filtering frameworks that do not relinearize is widespread and known to provide accurate estimates [50].

III. Decentralized Extended Information Filter

We consider the problem of estimating the pose of a client vehicle given acoustic range observations from the server to the client, as illustrated in Fig. 2. The CEKF [8], [9], which has concurrent access to sensor data from all vehicles, serves as the "gold-standard" and is well suited to modeling this cooperative localization problem for its ability to track correlation between multiple vehicles. Although communicating the sensor measurements from each underwater vehicle in real time to the CEKF is not (practically) feasible, many filtering operations in the information form of the CEKF, i.e., the CEIF, have properties that lend themselves to distributed computation. The DEIF algorithm provides a way to distribute the computation of a CEIF for a server and client vehicle under a low bandwidth requirement.

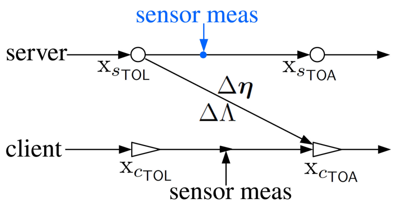


Fig. 2. Server poses (circles) and client poses (triangles) are linked by the range measurement. As described later, at the time-of-arrival (TOA), the centralized filter has access to the server's measurements between the time-of-launch (TOL) and the TOA, whereas the DEIF does not.

The implementation of the DEIF requires two separate filters, each of which processes sensor data causally and asynchronously. The server-side filter has real-time access to its local sensor data only; it is independent of, and has no knowledge of, the client state, client sensor measurements, or range measurements. At the time-of-launch (TOL) of each acoustic broadcast, the server-side filter computes a compact fixed-size representation of its state and uncertainty to transmit acoustically to the client. As described by Webster *et al.* [51], the server-to-client range is measured using the OWTT of the acoustic broadcast assuming a known sound velocity profile. The client-side filter has real-time access to the client's sensor data and the asynchronous acoustic broadcasts from the server, but does not have access to the server's raw sensor measurements, except for the processed information contained in the server-to-client acoustic broadcasts. The DEIF algorithm allows the client filter to exactly reconstruct the CEIF/CEKF at the TOA of server acoustic broadcasts.

In the following, we present a review of the salient features of the CEIF that lead to its distributed operation within the DEIF. We then report the DEIF algorithm.

A. *Extended Information Filter*

The EIF is characterized by the information matrix $\boldsymbol{\Lambda}$ and the information vector $\boldsymbol{\eta}$, which can be defined in terms of the mean $\boldsymbol{\mu}$ and covariance $\boldsymbol{\Sigma}$ of the state vector $\boldsymbol{x}$ as

$$\boldsymbol{\Lambda} = \boldsymbol{\Sigma}^{-1}, \quad \boldsymbol{\eta} = \boldsymbol{\Lambda}\boldsymbol{\mu} \tag{1}$$

where

$$\boldsymbol{\Sigma} = E\left[(\boldsymbol{x}-\boldsymbol{\mu})(\boldsymbol{x}-\boldsymbol{\mu})^{\top}\right], \quad \boldsymbol{\mu} = E[\boldsymbol{x}] \tag{2}$$

and $E[\cdot]$ is the expectation operator [50].

1) Process Prediction: For the general process prediction equations, we consider a state vector with two terms in it: the current state $\boldsymbol{x}_k$ of a vehicle and a collection of delayed states (i.e., previous states) $\boldsymbol{x}_p$. The stacked vector $\mathbf{x}_k$ represents the combined state at time k given all data up through time k

$$\mathsf{x}_k = \left[\boldsymbol{x}_k^\top, \boldsymbol{x}_p^\top\right]^\top \tag{3}$$

and has an associated information matrix and vector given by

$$\boldsymbol{\Lambda}_k = \begin{bmatrix}\Lambda_{kk} & \Lambda_{kp}\\ \Lambda_{pk} & \Lambda_{pp}\end{bmatrix}, \quad \boldsymbol{\eta}_k = \begin{bmatrix}\eta_k\\ \eta_p\end{bmatrix}. \tag{4}$$

Here, Λ_{ii} and η_i represent the blocks of the stacked information matrix $\boldsymbol{\Lambda}_k$ and vector $\boldsymbol{\eta}_k$ at time k, respectively, corresponding to state component $\boldsymbol{x}_{i\in\{k,p\}}$.

The process model predicting the state one time step ahead is given by

$$\bar{\mathsf{x}}_{k+1} = \boldsymbol{f}(\mathsf{x}_k, \boldsymbol{u}_{k+1}) + \boldsymbol{v}_k \tag{5}$$

with distribution $p(\boldsymbol{x}_{k+1}, \boldsymbol{x}_p | \mathbf{Z}^{1:k}, \mathrm{U}^{1:k+1})$, where

$$\bar{\boldsymbol{\Lambda}}_{k+1} = \begin{bmatrix}\Psi_k & \boldsymbol{Q}_k^{-1}\boldsymbol{F}_k\boldsymbol{\Omega}_k^{-1}\Lambda_{kp}\\ \Lambda_{pk}\boldsymbol{\Omega}_k^{-1}\boldsymbol{F}_k^\top\boldsymbol{Q}_k^{-1} & \Lambda_{pp}-\Lambda_{pk}\boldsymbol{\Omega}_k^{-1}\Lambda_{kp}\end{bmatrix} \tag{6}$$

$$\bar{\boldsymbol{\eta}}_{k+1} = \begin{bmatrix}\boldsymbol{Q}_k^{-1}\boldsymbol{F}_k\boldsymbol{\Omega}_k^{-1}\eta_k + \Psi_k\big(\boldsymbol{f}(\boldsymbol{\mu}_k, \boldsymbol{u}_{k+1}) - \boldsymbol{F}_k\boldsymbol{\mu}_k\big)\\ \eta_p - \Lambda_{pk}\boldsymbol{\Omega}_k^{-1}\eta_k^\star\end{bmatrix}. \tag{7}$$

Here, $\mathbf{Z}^{1:k}$ is the set of sensor measurements up through time k, $\mathrm{U}^{1:k+1}$ is the set of control inputs up through time $k+1$, $\boldsymbol{f}(\cdot)$ is the nonlinear process model with Jacobian $\boldsymbol{F}_k$ (evaluated about $\boldsymbol{\mu}_k$), $\boldsymbol{Q}_k$ is the covariance of the zero-mean Gaussian process noise $\boldsymbol{v}_k$, and, for notational convenience,

$$\Psi_k = \left(\boldsymbol{Q}_k + \boldsymbol{F}_k\Lambda_{kk}^{-1}\boldsymbol{F}_k^\top\right)^{-1} \tag{8}$$

$$\boldsymbol{\Omega}_k = \Lambda_{kk} + \boldsymbol{F}_k^\top\boldsymbol{Q}_k^{-1}\boldsymbol{F}_k \tag{9}$$

$$\eta_k^\star = \eta_k - \boldsymbol{F}_k^\top\boldsymbol{Q}_k^{-1}\big(\boldsymbol{f}(\boldsymbol{\mu}_k, \boldsymbol{u}_{k+1}) - \boldsymbol{F}_k\boldsymbol{\mu}_k\big) \tag{10}$$

as derived by Eustice *et al.* [45].

The blocks of the information matrix and vector corresponding to past states, Λ_{pp} and η_p, respectively, are only changed by an additive term when they are propagated forward in time. For a linear process model, the additive term can be calculated without linearization or need for the current mean. Furthermore, all other blocks of the information matrix and vector are changed in a way that does not depend on the information of the past states.

2) Process Prediction With Augmentation: In (6) and (7), the current state at time k is propagated to time $k+1$ so that

$$\bar{\mathsf{x}}_{k+1} = \left[\boldsymbol{x}_{k+1}^\top, \boldsymbol{x}_p^\top\right]^\top. \tag{11}$$

If we augment the stacked state vector to include the state at time $k+1$ in addition to the original states

$$\bar{\mathsf{x}}_{k+1} = \left[\boldsymbol{x}_{k+1}^\top, \boldsymbol{x}_k^\top, \boldsymbol{x}_p^\top\right]^\top \tag{12}$$

then the process prediction equations, which represent the distribution $p(\boldsymbol{x}_{k+1}, \boldsymbol{x}_k, \boldsymbol{x}_p | \mathbf{Z}^{1:k}, \mathrm{U}^{1:k+1})$, have a different structure described by

$$\bar{\boldsymbol{\Lambda}}_{k+1} = \begin{bmatrix}\boldsymbol{Q}_k^{-1} & -\boldsymbol{Q}_k^{-1}\boldsymbol{F}_k & \mathbf{0}\\ -\boldsymbol{F}_k^\top\boldsymbol{Q}_k^{-1} & \boldsymbol{\Omega}_k & \Lambda_{kp}\\ \mathbf{0} & \Lambda_{pk} & \Lambda_{pp}\end{bmatrix} \tag{13}$$

$$\bar{\boldsymbol{\eta}}_{k+1} = \begin{bmatrix}\boldsymbol{Q}_k^{-1}\big(\boldsymbol{f}(\boldsymbol{\mu}_k, \boldsymbol{u}_{k+1}) - \boldsymbol{F}_k\boldsymbol{\mu}_k\big)\\ \eta_k^\star\\ \eta_p\end{bmatrix} \tag{14}$$

where the above equations use the same variable definitions as in (6) and (7). Equations (13) and (14) can each be written as the sum of two terms:

$$\bar{\boldsymbol{\Lambda}}_{k+1} = \begin{bmatrix}\boldsymbol{Q}_k^{-1} & -\boldsymbol{Q}_k^{-1}\boldsymbol{F}_k & \mathbf{0}\\ -\boldsymbol{F}_k^\top\boldsymbol{Q}_k^{-1} & \boldsymbol{F}_k^\top\boldsymbol{Q}_k^{-1}\boldsymbol{F}_k & \mathbf{0}\\ \mathbf{0} & \mathbf{0} & \mathbf{0}\end{bmatrix} + \begin{bmatrix}\mathbf{0} & \mathbf{0} & \mathbf{0}\\ \mathbf{0} & \Lambda_{kk} & \Lambda_{kp}\\ \mathbf{0} & \Lambda_{pk} & \Lambda_{pp}\end{bmatrix} \tag{15}$$

$$\bar{\boldsymbol{\eta}}_{k+1} = \begin{bmatrix}\boldsymbol{Q}_k^{-1}\big(\boldsymbol{f}(\boldsymbol{\mu}_k, \boldsymbol{u}_{k+1}) - \boldsymbol{F}_k\boldsymbol{\mu}_k\big)\\ -\boldsymbol{F}_k^\top\boldsymbol{Q}_k^{-1}\big(\boldsymbol{f}(\boldsymbol{\mu}_k, \boldsymbol{u}_{k+1}) - \boldsymbol{F}_k\boldsymbol{\mu}_k\big)\\ \mathbf{0}\end{bmatrix} + \begin{bmatrix}\mathbf{0}\\ \eta_k\\ \eta_p\end{bmatrix}. \tag{16}$$

The first term contains the process prediction information and the second term contains the prior information (4) at time k, but padded with zeros corresponding to the state at time $k+1$. These zeros reflect that at time k, prior to prediction, there is no information about the state at time $k+1$. This operation is purely additive and, in fact, only modifies the Λ_{kk} and η_k blocks of the original information matrix and vector.

As observed by Eustice *et al.* [45], prediction with augmentation results in $\bar{\boldsymbol{\Lambda}}_{k+1}$ having a sparse, block-tridiagonal structure. The sparsity of $\bar{\boldsymbol{\Lambda}}_{k+1}$ is important in the context of acoustic navigation because it bounds the amount of information that must be acoustically transmitted from the server to the client in order for the client to fully reconstruct the server's estimated state (see Section III-C).

3) Measurement Update: The observation model is

$$\boldsymbol{z}_k = \boldsymbol{h}(\bar{\mathsf{x}}_k) + \boldsymbol{w}_k \tag{17}$$

which leads to an updated posterior, i.e., $p(\boldsymbol{x}_k, \boldsymbol{x}_p | \mathbf{Z}^{1:k}, \mathrm{U}^{1:k})$, that is parametrized by

$$\boldsymbol{\Lambda}_k = \bar{\boldsymbol{\Lambda}}_k + \boldsymbol{H}_k^\top\boldsymbol{R}_k^{-1}\boldsymbol{H}_k \tag{18}$$

$$\boldsymbol{\eta}_k = \bar{\boldsymbol{\eta}}_k + \boldsymbol{H}_k^\top\boldsymbol{R}_k^{-1}\big(\boldsymbol{z}_k - \boldsymbol{h}(\bar{\boldsymbol{\mu}}_k) + \boldsymbol{H}_k\bar{\boldsymbol{\mu}}_k\big) \tag{19}$$

where $\boldsymbol{z}_k$ is the measurement vector, $\boldsymbol{R}_k$ is the covariance matrix of the zero-mean Gaussian measurement noise $\boldsymbol{w}_k$, and $\boldsymbol{h}(\cdot)$ is the nonlinear measurement model with Jacobian $\boldsymbol{H}_k$ (evaluated about $\bar{\boldsymbol{\mu}}_k$) [45].

The information matrix is additively updated by the matrix outer product $\boldsymbol{H}_k^\top\boldsymbol{R}_k^{-1}\boldsymbol{H}_k$. In general, the measurement Jacobian $\boldsymbol{H}_k$ is sparse resulting in an update step that only affects a few subblocks of the information matrix and vector. Onboard navigation sensor observations at the server or client only modify, respectively, the server state or client state. Range measure-

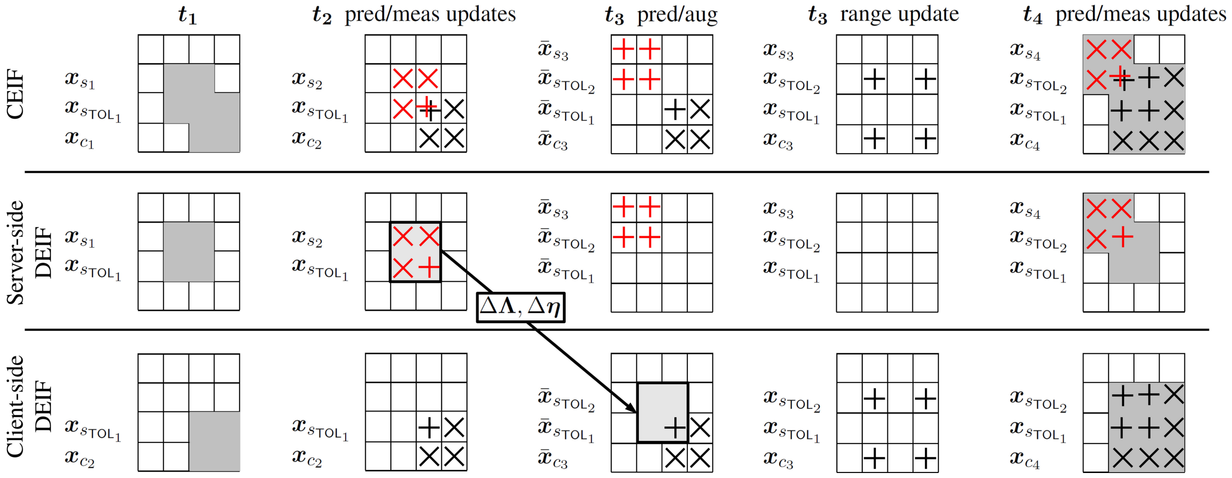


Fig. 3. Graphical comparison of the CEIF and the DEIF's server and client filters. In each sequence, the "$+$" and "$\times$" indicate additive and nonadditive operations, respectively. Red and black operators designate operations performed by the server and client vehicles, respectively. The gray areas in the first and last panels depict the nonzero sparsity pattern of the information matrix. Both the server and client DEIF filters track a portion of the centralized state vector, with the client-side DEIF achieving equality with the CEIF (for the states they share in common) immediately following the range measurement update.

ments between the server and the client change both the current client state and the delayed server state.

B. Centralized Extended Information Filter Operation

The CEIF is simply the information form dual [40] of the CEKF [8], [9]. It processes sensor data from all vehicles causally and asynchronously in real time.

1) Centralized Extended Information Filter State Vector: Since range measurements are generally made between server TOL states and client TOA states, the CEIF tracks a stacked state vector containing the current server and client states, as well as copies of the TOL server states. At each acoustic TOL, a copy of the server state is appended onto the state vector via the prediction with augmentation operation (14). After n acoustic broadcasts, the state vector consists of

$$\mathbf{x}_k = \left[\boldsymbol{x}_{s_k}^\top, \boldsymbol{x}_{s\text{TOL}_n}^\top, \ldots, \boldsymbol{x}_{s\text{TOL}_2}^\top, \boldsymbol{x}_{s\text{TOL}_1}^\top, \boldsymbol{x}_{c_k}^\top\right]^\top \tag{20}$$

where we use the following notation: $\mathbf{x}_k$ denotes the entire stacked state at time k; $\boldsymbol{x}_{s_k}$ and $\boldsymbol{x}_{c_k}$ are the current server and client states, respectively; and $\boldsymbol{x}_{s\text{TOL}_i}$ is the server state at the ith TOL. We note that this delayed-state framework is a smoothing algorithm, as TOL delayed states benefit from new information.

Before the first server-to-client broadcast, the information matrix is block diagonal between the server and client, as the vehicles have shared no information. Since initially each vehicle state evolves independently of the other vehicle, the CEIF operation mirrors the operation of two *independent* information filters, each tracking one vehicle.

2) Server-to-Client Broadcasts: At the TOA of server broadcast, the client measures the OWTT-derived range to the server. The observation model for the range measurement is

$$z_k = \left\| \boldsymbol{x}_{c\text{TOA}_{xyz}} - \boldsymbol{x}_{s\text{TOL}_{xyz}} \right\| + w_k \tag{21}$$

where "*xyz*" indicates the position components of the client and server state vectors, and $w_k \sim \mathcal{N}(0, R_k)$ is the measurement noise. Noise in the range measurement is a result of several nonlinear error sources including multipath and timing imprecision. The error distribution, while strictly non-Gaussian, is commonly modeled as Gaussian [18], [52], [53] following an outlier rejection step. As mentioned previously, the range measurement Jacobian is sparse, as it only involves the client TOA position and the server TOL delayed state. Therefore, only blocks of the information matrix and vector corresponding to these states are modified during the measurement update.

Following the first range measurement update, the information matrix contains nonzero off-diagonal blocks between the server TOL state (i.e., a delayed state) and the current client state. However, because of the structure of prediction in the information form, the current server state information evolves in a way that does *not* depend on its delayed state given a linear server process model [see (6) and (7)], and the block of the information matrix corresponding to the delayed state only changes additively. Moreover, measurement updates for the current server state do *not* depend on its delayed state given linear observation models. This implies that the server accumulates information (through prediction and measurement updates with linear models) independent of the range measurement update. The CEIF operation is illustrated in Fig. 3.

The above understanding motivates the design of the DEIF algorithm, which distributes the CEIF server prediction and measurement updates to a filter running onboard the server, while a separate filter running onboard the client processes client prediction and measurement updates, including range measurement updates.

C. Server-Side DEIF Operation

The server-side filter has real-time access to sensor data only from server-based sensors; it is independent of the client state, client sensor measurements, and range measurements. The server-side implementation is described in Algorithm 1.

1) Server-Side Decentralized Extended Information Filter State Vector: The standalone server filter maintains an estimate

Algorithm 1 Server-side DEIF

```
Require: $\boldsymbol{\Lambda}_{s_0}, \boldsymbol{\eta}_{s_0}$ {initial server belief}
$\boldsymbol{\Lambda}_{s_{\mathsf{TOL}}}, \boldsymbol{\eta}_{s_{\mathsf{TOL}}} \leftarrow 0$ {last TOL belief}
loop
    if $k$ is $\mathsf{TOL}_n$ then
        $\Delta\boldsymbol{\Lambda}_s, \Delta\boldsymbol{\eta}_s \leftarrow$ calcDelta($\boldsymbol{\Lambda}_{s_k}, \boldsymbol{\eta}_{s_k}, \boldsymbol{\Lambda}_{s_{\mathsf{TOL}}}, \boldsymbol{\eta}_{s_{\mathsf{TOL}}}$)
        broadcastPacket($\Delta\boldsymbol{\Lambda}_s$, $\Delta\boldsymbol{\eta}_s$)
        $\boldsymbol{\Lambda}_{s_{\mathsf{TOL}}}, \boldsymbol{\eta}_{s_{\mathsf{TOL}}} \leftarrow \boldsymbol{\Lambda}_{s_k}, \boldsymbol{\eta}_{s_k}$ {store for next $\Delta$ calc}
        $\boldsymbol{\Lambda}_{s_k}, \boldsymbol{\eta}_{s_k} \leftarrow$ predictAugment($\boldsymbol{\Lambda}_{s_k}, \boldsymbol{\eta}_{s_k}$)
    else
        $\boldsymbol{\Lambda}_{s_k}, \boldsymbol{\eta}_{s_k} \leftarrow$ predict($\boldsymbol{\Lambda}_{s_k}, \boldsymbol{\eta}_{s_k}$)
    end if
    $\boldsymbol{\Lambda}_{s_k}, \boldsymbol{\eta}_{s_k} \leftarrow$ localMeasUpdate($\boldsymbol{\Lambda}_{s_k}, \boldsymbol{\eta}_{s_k}, \boldsymbol{z}_{s_k}$)
    $k \leftarrow k + 1$
end loop
```

of the current server state as well as copies of TOL server states. Each time a new acoustic range packet is broadcast (i.e., at the TOL), a copy of the current server state is appended onto the state vector. This results in a state vector, after n acoustic packets have been transmitted, of the form

$$\mathbf{x}_{s_k} = \left[\boldsymbol{x}_{s_k}^\top, \boldsymbol{x}_{s_{\mathsf{TOL}_n}}^\top, \ldots, \boldsymbol{x}_{s_{\mathsf{TOL}_2}}^\top, \boldsymbol{x}_{s_{\mathsf{TOL}_1}}^\top\right]^\top \tag{22}$$

where we have adopted the following notation convention: $\mathbf{x}_{s_k}$ denotes the entire stacked server state at time k; $\boldsymbol{x}_{s_k}$ is the current server state; and $\boldsymbol{x}_{s_{\mathsf{TOL}_i}}$ is the server state when the ith range packet was broadcast. The corresponding information matrix has a sparse, block tridiagonal structure. As in the CEIF, only the blocks of the information matrix corresponding to the current server state and the most recent TOL delayed state are affected by predictions and measurement updates—changes in the information matrix and vector are constrained between successive TOLs.

In addition to the information matrix $\boldsymbol{\Lambda}_{s_k}$ and vector $\boldsymbol{\eta}_{s_k}$ that characterize the current server state $\mathbf{x}_{s_k}$, a copy of the information matrix and vector at the last TOL are also stored (i.e., $\boldsymbol{\Lambda}_{s_{\mathsf{TOL}_{n-1}}}$ and $\boldsymbol{\eta}_{s_{\mathsf{TOL}_{n-1}}}$). This allows the server to compute the difference in the information matrix and vector between TOLs, which is termed the "delta information."

2) Delta Server Information for Acoustic Transmission: To initiate a range measurement, the server broadcasts a range packet containing delta information about the server state. This delta information encapsulates all information gained via predictions and measurement updates about the server state since the last TOL

$$\Delta\boldsymbol{\Lambda}_{s_{\mathsf{TOL}_{n-1:n}}} = \boldsymbol{\Lambda}_{s_{\mathsf{TOL}_n}} - \boldsymbol{\Lambda}'_{s_{\mathsf{TOL}_{n-1}}} \tag{23}$$

$$\Delta\boldsymbol{\eta}_{s_{\mathsf{TOL}_{n-1:n}}} = \boldsymbol{\eta}_{s_{\mathsf{TOL}_n}} - \boldsymbol{\eta}'_{s_{\mathsf{TOL}_{n-1}}} \tag{24}$$

where $\boldsymbol{\Lambda}'_{s_{\mathsf{TOL}_{n-1}}}$ and $\boldsymbol{\eta}'_{s_{\mathsf{TOL}_{n-1}}}$ represent the information at the last TOL but padded with zeros to match the dimension of $\boldsymbol{\Lambda}_{s_{\mathsf{TOL}_n}}$ and $\boldsymbol{\eta}_{s_{\mathsf{TOL}_n}}$. The zero padding represents the fact that at the last TOL, the server had no information about its state at the next TOL. These range packets are processed onboard the client-side DEIF to reconstruct the CEIF, as described in Section III-D.

The calculation and transmission of delta server information, as described here, allows us to delegate the task of processing server sensor data in the CEIF to an independent server-side DEIF. The server-side DEIF operation is depicted in Fig. 3.

Algorithm 2 Client-side DEIF

```
Require: $\boldsymbol{\Lambda}_{c_0}, \boldsymbol{\eta}_{c_0}$ {initial client belief}
loop
    $\boldsymbol{\Lambda}_{c_k}, \boldsymbol{\eta}_{c_k} \leftarrow$ predict($\boldsymbol{\Lambda}_{c_k}, \boldsymbol{\eta}_{c_k}$)
    if $\Delta\boldsymbol{\Lambda}_s, \Delta\boldsymbol{\eta}_s \leftarrow$ receivedPacket() then
        $\boldsymbol{\Lambda}_{c_k}, \boldsymbol{\eta}_{c_k} \leftarrow$ addDelta($\boldsymbol{\Lambda}_{c_k}, \boldsymbol{\eta}_{c_k}, \Delta\boldsymbol{\Lambda}_s, \Delta\boldsymbol{\eta}_s$)
        $\boldsymbol{\Lambda}_{c_k}, \boldsymbol{\eta}_{c_k} \leftarrow$ rangeMeasUpdate($\boldsymbol{\Lambda}_{c_k}, \boldsymbol{\eta}_{c_k}, z_{r_k}$)
    end if
    $\boldsymbol{\Lambda}_{c_k}, \boldsymbol{\eta}_{c_k} \leftarrow$ localMeasUpdate($\boldsymbol{\Lambda}_{c_k}, \boldsymbol{\eta}_{c_k}, \boldsymbol{z}_{c_k}$)
    $k \leftarrow k + 1$
end loop
```

D. Client-Side DEIF Operation

The client-side DEIF has real-time access to the client's onboard sensor data and the asynchronous range packets from the server but does *not* have access to the server's raw sensor measurements; the only server state information available to the client is that which is encoded in the server-to-client acoustic broadcasts. The client-side DEIF reconstructs a local copy of the server-state by sequentially incorporating information encapsulated in the acoustic broadcasts. The client-side implementation is described in Algorithm 2.

1) Client-Side State Vector: In addition to the current client state, the DEIF maintains a copy of historic server states reconstructed from the delta information broadcasts. As a result, the DEIF state vector consists of two parts: the current client state and the TOL server delayed states

$$\mathbf{x}_{c_k} = \left[\boldsymbol{x}_{s_{\mathsf{TOL}_{n-1}}}^\top, \ldots, \boldsymbol{x}_{s_{\mathsf{TOL}_2}}^\top, \boldsymbol{x}_{s_{\mathsf{TOL}_1}}^\top, \boldsymbol{x}_{c_k}^\top\right]^\top \tag{25}$$

where we adopt the convention that $\mathbf{x}_{c_k}$ denotes the entire stacked client state vector at time k; $\boldsymbol{x}_{c_k}$ is the current client state; and $\boldsymbol{x}_{s_{\mathsf{TOL}_i}}$ is the server TOL state when the ith acoustic packet was broadcast. The client-side DEIF maintains the same state vector as the CEIF with the exception that the client-side DEIF does not track the current server state $\boldsymbol{x}_{s_k}$. We also note that, unlike the server, the client is able to use nonlinear process and observations models, as in [12]. The client vehicle is also free to add delayed-state client poses in order to estimate its smooth trajectory.

2) Incorporating Delta Server Information: At the TOA of a range packet onboard the client, the delta information included in the packet is incorporated followed by the range measurement update. The delta information is incorporated into the client-side DEIF by simple addition, in the analogous operation to (23) and (24):

$$\boldsymbol{\Lambda}_{c_{\mathsf{TOA}_n}} = \boldsymbol{\Lambda'}_{c_{\mathsf{TOA}_n}} + \Delta\boldsymbol{\Lambda}_{s_{\mathsf{TOL}_{n-1:n}}} \tag{26}$$

$$\boldsymbol{\eta}_{c_{\mathsf{TOA}_n}} = \boldsymbol{\eta'}_{c_{\mathsf{TOA}_n}} + \Delta\boldsymbol{\eta}_{s_{\mathsf{TOL}_{n-1:n}}} \tag{27}$$

where $\boldsymbol{\Lambda'}_{c_{\mathsf{TOA}_n}}$ is the client information matrix at the current TOA *before* the delta information is incorporated, and $\boldsymbol{\Lambda}_{c_{\mathsf{TOA}_n}}$ is the client information matrix at the current TOA *after* the delta information is incorporated. The client-side information is padded with zeros to reflect that, prior to incorporating the delta information packet, the client has no information regarding the newest TOL server state.

3) Range Measurement Updates: At the TOA of the range packet, after the delta information is incorporated, the range measurement update is performed. This operation is equivalent to the corresponding operation in the CEIF. As noted in Section III-C2, the delta information encapsulates all of the information that the filter has gained about the server state since the last acoustic packet was broadcast. The simplicity of this computation is one of the advantages of the information filter. The client-side DEIF operation is depicted in Fig. 3.

E. Equivalency of DEIF and Centralized Filter

Immediately after performing a range update, the client-side DEIF is identical to the equivalent centralized filter (CEIF or CEKF) estimate for the states that they have in common. There are several subtleties in this observation that we address here for clarity.

1) Filter Distributions: A range measurement is an observation between the server position at the TOL and the client position at the (later occurring) TOA, i.e., $\boldsymbol{x}_{s_{\mathsf{TOL}}}$ and $\boldsymbol{x}_{c_{\mathsf{TOA}}}$, respectively. Comparing the probability distributions of the centralized filter and DEIF immediately after the range measurement update, we find that they are *not* strictly identical:

$$\mathsf{DEIF}:$$
$$p\Big(\boldsymbol{x}_{c_{\mathsf{TOA}}}, \boldsymbol{x}_{s_{\mathsf{TOL}}} \big| z_{r_{\mathsf{TOA}}}, \mathbf{Z}_c^{1:\mathsf{TOA}}, \mathbf{U}_c^{1:\mathsf{TOA}}, \boxed{\mathbf{Z}_s^{1:\mathsf{TOL}}}, \boxed{\mathbf{U}_s^{1:\mathsf{TOL}}}\Big) \tag{28}$$

$$\mathsf{Centralized}:$$
$$p\Big(\boldsymbol{x}_{c_{\mathsf{TOA}}}, \boldsymbol{x}_{s_{\mathsf{TOL}}} \big| z_{r_{\mathsf{TOA}}}, \mathbf{Z}_c^{1:\mathsf{TOA}}, \mathbf{U}_c^{1:\mathsf{TOA}}, \boxed{\mathbf{Z}_s^{1:\mathsf{TOA}}}, \boxed{\mathbf{U}_s^{1:\mathsf{TOA}}}\Big) \tag{29}$$

where $z_{r_{\mathsf{TOA}}}$ is the most recent OWTT range measurement, $\mathbf{Z}_c^{1:\mathsf{TOA}}$ is the set of client sensor measurements up to the TOA, $\mathbf{U}_c^{1:\mathsf{TOA}}$ is the set of client control inputs up to the TOA, $\mathbf{Z}_s^{1:\mathsf{TOL}}$ is the set of server sensor measurements up to the TOL, and $\mathbf{U}_s^{1:\mathsf{TOL}}$ is the set of server control inputs up to the TOL. Note that the centralized filter has access to the server's sensor measurements and control inputs between the TOL and the TOA of which the DEIF has no knowledge (see Fig. 2).

The ramifications of this are that the DEIF performs a range measurement between the current client state and the best estimate of the server's state at the TOL given server sensor measurements and control inputs *only up to the TOL*. In contrast, the centralized filter performs a range measurement between the current client state and the best estimate of the server's state at the TOL given server sensor measurement and control inputs *up to the TOA*. Thus, the centralized filter is performing a smoothing operation on the server's state at the TOL, because it has access to additional information from the server's sensors after the data packet was broadcast.

2) Two-Step Delayed Update: To address this apparent discrepancy and to provide a fair comparison of the DEIF and the CEIF in our experiments, we use a two-step delayed update in the centralized filter. First, we perform an update for the range measurement with only server measurements and control inputs up through the TOL:

$$p\Big(\boldsymbol{x}_{c_{\mathsf{TOA}}}, \boldsymbol{x}_{s_{\mathsf{TOL}}} \big| z_{r_{\mathsf{TOA}}}, \mathbf{Z}_c^{1:\mathsf{TOA}}, \mathbf{U}_c^{1:\mathsf{TOA}}, \mathbf{Z}_s^{1:\mathsf{TOL}}, \mathbf{U}_s^{1:\mathsf{TOL}}\Big). \tag{30}$$

Second, we perform another update for the server using measurements and control inputs occurring between the TOL and the TOA (see Fig. 2):

$$p\Big(\boldsymbol{x}_{c_{\mathsf{TOA}}}, \boldsymbol{x}_{s_{\mathsf{TOL}}} \big| z_{r_{\mathsf{TOA}}}, \mathbf{Z}_c^{1:\mathsf{TOA}}, \mathbf{U}_c^{1:\mathsf{TOA}}, \mathbf{Z}_s^{1:\mathsf{TOL}}, \mathbf{U}_s^{1:\mathsf{TOL}}, \boxed{\mathbf{Z}_s^{\mathsf{TOL:TOA}}}, \boxed{\mathbf{U}_s^{\mathsf{TOL:TOA}}}\Big). \tag{31}$$

This provides a fair comparison so that the DEIF distribution in (28) is identical to the centralized filter distribution in (30), without compromising the centralized filter's final distribution, i.e., (31) is identical to (29).[1]

Between range measurements, the centralized filter and DEIF estimates of the client's state will not be identical. Correlation that develops between the client and server states causes new server measurements to *smooth* the client state in the centralized case. Since these server measurements are not immediately available to the client-side DEIF, no smoothing occurs until *after* a delta information packet is received. However, at the instant immediately after *each* range measurement update, the filter estimates will be identical (when using the two-step delayed update) because the information contained in the delta information represents all previous server measurements.

F. Ramifications of Filter Design

A final note on this derivation, which is mentioned briefly in Section III-B, is that the server's process and observation models must be linear in order for the delta information calculated onboard the server to match the delta information that would be calculated in the equivalent CEIF. Linear plant and observation models guarantee that the process prediction and measurement updates are independent of the server's current state. Thus, the calculated delta information is independent of the actual value of the state. This is essential because the standalone server filter will have a different estimate of the server's current state than the CEIF, a result of the CEIF's server estimate being conditioned on previous range measurements. In contrast, there is no linearity requirement for the client models, as shown in [12], which employs a six-degree-of-freedom (DOF) nonlinear process model for the client.

It is also worth noting that, in practice, it is undesirable, and unnecessary, for the state vector dimension to grow without bound, as per (22) and (25). Instead, we marginalize out the oldest historic server states on both the server and the client in order to maintain a fixed-length state vector dimension [45].

[1] From an implementation perspective, we augment the centralized state vector with copies of the server state at every measurement event between the TOL and the TOA to achieve a premarginalized equivalent of (29) and then retroactively apply the TOL to TOA server measurements, followed by marginalizing out the extra server states to arrive at (31).

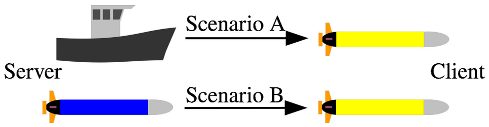


Fig. 4. Algorithm performance is tested using two different two-node topologies: *Scenario A*, where the server (shown as a ship) has continuous access to GPS, and *Scenario B*, where the server (shown as an AUV) has only intermittent access to GPS.

Minimally, the most recent TOL delayed state must be kept in the stacked state vector in order for the delta information computation to be carried out correctly.

IV. Implementation Details

The results described herein illustrate two points. First, we demonstrate the numerical equivalence of the DEIF and the CEIF, which was established analytically in Section III. Second, we compare the performance of the DEIF with several alternate filtering approaches reported in the literature. The comparison, which is similar to that presented by Walls and Eustice [13], is performed in post-processing between a pure dead-reckoned (DR) algorithm, a CEIF/CEKF, an EEKF, the IU algorithm, and a simple EKF that broadcasts the GPS position of the server in each acoustic packet instead of filtering the server state. The state description and process model, as well as the implementation of these filters, are described below.

The different filter frameworks are compared using both simulated and real-world experimental data for two operational scenarios illustrated in Fig. 4: a server that has continuous access to GPS (e.g., a support ship or surface buoy), and a server that does not have continuous access to GPS (e.g., one autonomous underwater vehicle (AUV) surfacing intermittently to receive GPS updates, while supporting a continuously submerged AUV).

A. Vehicle State Description

Since attitude and depth are easily instrumented with bounded error, we consider only x, y horizontal position estimation during our experiments. Range measurements can easily be projected to the local-level plane with knowledge of relative vehicle depth. Furthermore, it is advantageous to represent the vehicle state with the smallest dimension possible to minimize bandwidth requirements for transmitting state information over the acoustic channel. Results using a full 6-degree-of-freedom (DOF) nonlinear process model with the DEIF have been previously reported by Webster *et al.* [12]. Further discussions on the justification and ramifications of using this simplified state model for the server and client are covered in Section VI.

We employ a 2-DOF process model for both the server and the client vehicles

$$\boldsymbol{x} = [x,\, y,\, \dot{x},\, \dot{y}]^\top \tag{32}$$

where the platform position in the local-level plane is denoted by the x, y pair, and the corresponding world-frame velocities are $\dot{x}$ and $\dot{y}$.

Each filter is implemented using a constant-velocity linear process model resulting in a one-step prediction of the form

$$\bar{\boldsymbol{x}}_{k+1} = \boldsymbol{F}_k \boldsymbol{x}_k + \boldsymbol{v}_k. \tag{33}$$

Here, $\boldsymbol{F}_k$ is the discrete-time linear state transition matrix:

$$\boldsymbol{F}_k = \begin{bmatrix} \boldsymbol{I} & \boldsymbol{I}\Delta T \\ \boldsymbol{0} & \boldsymbol{I} \end{bmatrix} \tag{34}$$

where ΔT is the period between time steps.

B. Filter Implementations

We have implemented four filters and a DR filter for comparison with the DEIF algorithm. Each is briefly described below. We employ the convention of using "s" and "c" subscripts to designate the server and client states, respectively.

1) Centralized Extended Information Filter: The CEIF serves as our benchmark gold-standard solution. The CEIF is a post-processing formulation that has access to all sensor measurements from all vehicles and tracks the stacked state of the server and client platforms:

$$\mathbf{x} = [\boldsymbol{x}_s^\top, \boldsymbol{x}_c^\top]^\top. \tag{35}$$

Since the CEIF is the dual equivalent of the CEKF (each is the inverse of the other), for ease of comparison, we implement the CEKF algorithm, as reported in [8] and [9]. For cohesiveness with the previous discussion, throughout our results, we refer to this CEKF generated result as the CEIF. The estimated mean and covariance of the equivalent CEIF are

$$\boldsymbol{\mu} = \begin{bmatrix} \boldsymbol{\mu}_s \\ \boldsymbol{\mu}_c \end{bmatrix}, \quad \boldsymbol{\Sigma} = \begin{bmatrix} \boldsymbol{\Sigma}_{ss} & \boldsymbol{\Sigma}_{sc} \\ \boldsymbol{\Sigma}_{cs} & \boldsymbol{\Sigma}_{cc} \end{bmatrix}. \tag{36}$$

The mean and covariance follow the standard Kalman prediction equations with a combined state transition matrix $\boldsymbol{F}_k$ and noise covariance matrix $\boldsymbol{Q}_k$ given by

$$\boldsymbol{F}_k = \mathrm{blkdiag}(\boldsymbol{F}_{s_k}, \boldsymbol{F}_{c_k}) \tag{37}$$

$$\boldsymbol{Q}_k = \mathrm{blkdiag}(\boldsymbol{Q}_{s_k}, \boldsymbol{Q}_{c_k}). \tag{38}$$

To correctly model range measurement updates, the CEIF augments the global state to include the server state at TOL

$$\mathbf{x}' = [\boldsymbol{x}_s^\top, \boldsymbol{x}_c^\top, \boldsymbol{x}_{s_{\mathrm{TOL}}}^\top]^\top. \tag{39}$$

This allows the filter to perform a standard nonlinear Kalman update with the OWTT observation, as in (21). Once the measurement update has been completed, the augmented TOL state can be marginalized out in order to maintain a bounded state vector dimension.

Initially, the navigation estimates of each vehicle are uncorrelated so that the global covariance matrix is block diagonal. However, sharing intervehicle range measurements builds correlation between vehicle navigation estimates [13]. The CEIF tracks this correlation because it has access to all measurements from all platforms.

2) Egocentric Extended Kalman Filter: The egocentric distributed approach assumes that each range measurement originates at an independent source. This is essentially equivalent to a

CEKF with all of the off block-diagonal elements of the covariance matrix actively held to zero. Distributing this filter simply requires that the server transmit its local mean and covariance $\boldsymbol{\mu}_s$ and $\boldsymbol{\Sigma}_s$, respectively. Acoustic data packets are constant size, as only local information is transmitted. Therefore, the EEKF can trivially scale up to arbitrarily large networks and is robust to packet loss. However, because internode correlation is not tracked, this method can result in an overconfident estimate of uncertainty [34].

In order to perform a range measurement update, the client constructs a combined state vector and covariance matrix by appending the transmitted statistics of the server

$$\boldsymbol{\mu}_c' = \begin{bmatrix} \boldsymbol{\mu}_c \\ \boldsymbol{\mu}_s \end{bmatrix} \quad \boldsymbol{\Sigma}_c' = \begin{bmatrix} \boldsymbol{\Sigma}_{cc} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\Sigma}_{ss} \end{bmatrix}. \tag{40}$$

The measurement update then proceeds with the standard Kalman update with the measurement model given in (21). Following the update, the state elements corresponding to the server are marginalized out. Note that this filter does not track correlation; the next range update will follow the same update procedure assuming no correlation, resulting in a double counting of information because the client's state has already been informed by the server's state.

3) Interleaved Update Algorithm: Bahr *et al.* [35] propose the IU algorithm as a solution to the problem of inconsistency seen in the EEKF. To avoid overconfidence, the IU algorithm maintains a set of different vehicle navigation estimates and only performs range measurement updates between estimates that are known to be uncorrelated. The IU algorithm is a bookkeeping approach that can be employed with any filtering modality (e.g., EKF, particle filter, unscented Kalman filter). For comparison with the other acoustic navigation frameworks considered in this paper, we present the IU as applied to an EKF.

Under the IU framework, each platform maintains a bank of EKFs. At time k, the set of state vectors and covariance matrices tracked by the ith vehicle, which are denoted $\mathcal{U}_i(k)$ and $\Sigma_i(k)$, respectively, are defined as

$$\mathcal{U}_i(k) = \left\{\boldsymbol{\mu}_i^1(k), \ldots, \boldsymbol{\mu}_i^{2^{v-1}}(k)\right\} \tag{41}$$

$$\Sigma_i(k) = \left\{\boldsymbol{\Sigma}_i^1(k), \ldots, \boldsymbol{\Sigma}_i^{2^{v-1}}(k)\right\} \tag{42}$$

where v is the total number of vehicles in the network. Each platform also stores the origin of every acoustic broadcast in a transmission matrix $\boldsymbol{T}$, where each row represents a filter within its local set and each column corresponds to a vehicle in the network. Entries $\boldsymbol{T}_{ij}$ represent the last time that the ith filter used the jth vehicle to update its navigation estimate. During an acoustic broadcast, each source node transmits its transmission matrix, as well as its entire bank of filters. A receiving node updates each of its filters by searching for a filter in the transmitted set that does not contain an update that could be correlated. This ensures that double counting of information will not occur where correlation could exist. The full mechanics of this update step are described in [35].

In a two-vehicle network with unidirectional communication, range measurement updates are performed using the server's transmitted state estimate and a client state estimate that has not previously been informed by a server estimate (i.e., DR), thus ensuring that correlated information is not naively fused. As a result of this methodology, the IU produces an estimate of the client's state that has unbounded growth in uncertainty over time, although it improves over DR.

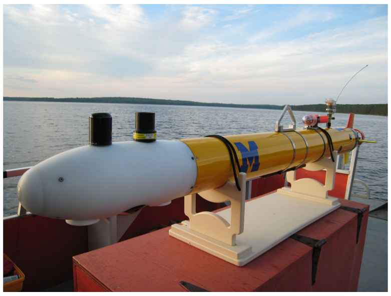


Fig. 5. Iver31—one of the two custom Iver AUVs used in field experiments.

4) Raw Global Positioning System: The raw GPS filtering scheme mirrors the implementation of the EEKF, except each server broadcast encodes an independent observation of its position from a raw GPS measurement. In this scenario, each measurement is assumed to be uncorrelated and results in a consistent, bounded-error, position estimate. Note that this method is only applicable for a server that has an independent position observation source available when the acoustic packet is sent. This is in contrast with all other methods described, which utilize all range measurements between the server and client regardless of access to GPS. The raw GPS algorithm offers robustness to packet loss and the ability to fuse information from multiple independent servers (see [2] and [36]).

V. Results

This section reports results using both simulated data and experimental data for a three-vehicle deployment consisting of one surface platform (a ship) and two autonomous underwater vehicles (AUVs). The simulated dataset mimics the experimental dataset, using identical sensor characteristics and approximately the same geometry. We explore the performance of each filtering strategy within the two communication topologies shown in Fig. 4.

A. Vehicle and Sensor Descriptions

The physical experiment was performed using a topside surface craft and two modified Ocean-Server, Inc., Iver2 AUVs, Iver28 and Iver31, which are depicted in Fig. 5. Each AUV includes a typical advanced DR sensor suite, as tabulated in Table I and reported in [54]. The AUVs measure body-frame velocities with a 600 kHz RDI Doppler velocity log (DVL), attitude with a Microstrain 3DM-GX1-AHRS, and depth with a Desert Star Systems SSP-1 digital pressure sensor. We project the body-frame velocity measurements into the world-frame via the Microstrain and treat these as linear observations of the $\dot{x}, \dot{y}$ elements of our state with a first-order covariance estimate of the combined DVL and AHRS covariance. To calibrate the compass, we fit a nonlinear bias function to the magnetic compass

TABLE I
NAVIGATION SENSORS: SAMPLING FREQUENCY AND NOISE CHARACTERISTICS

Client Sensors	**State Variable**	**Frequency**	**Noise (1 std)**
Microstrain	ϕ, θ, ψ	25 Hz	2°
DVL	$\dot{x}, \dot{y}$	3 Hz	5 cm/s
pressure depth	z	2 Hz	10 cm
acoustic modem	slant range	every ∼15 sec	1 m

Server Sensors	**State Variable**	**Frequency**	**Noise (1 std)**
GPS	x, y	1 Hz	3 m

reading during a separate calibration mission. The topside platform only observes world-frame position as measured by GPS. The noise values reported in Table I were used for all other sensor models. Additionally, the OWTT measurement model included a Mahalanobis distance check as a final outlier rejection test. The sensor characteristics of the experimental system (see Table I) were also used to create the simulated dataset.

The schedule and source of the acoustic broadcasts were defined by a fixed, 145-s time-division multiple access (TDMA) schedule, consisting of six server broadcasts and four subsea broadcasts from each AUV. This TDMA cycle corresponds to one acoustic broadcast roughly every 15 s. The acoustic broadcasts were transmitted with a carrier frequency of 25 kHz and frequency-shift keying encoding (rate 0 on the WHOI Micro-Modem). Each acoustic broadcast lasted approximately 2–3 s and transmitted 32 bytes of data.

B. Survey Description

During the experimental survey, the AUVs were programmed to follow rectangular grid survey patterns, oriented approximately 90° to each other, with tracklines of roughly 500 m length spaced 50 m apart, as shown in Fig. 6. The vehicles surfaced at the end of each trackline to acquire GPS, but these GPS measurements are not used during the filter comparisons below, except where noted. The survey lasted 1.4 h, during which the AUVs traveled 5.1 and 5.3 km, respectively. During the survey, we positioned the ship upwind (North) and then drifted downwind (South) sending and receiving acoustic data packets. This trajectory was repeated several times. The simulated survey is designed to mimic the experimental survey, except that the ship's track is simplified to a diamond shape. In the survey simulation, the vehicle track length is 5.3 km, and the simulated survey lasts 1.5 h.

We compare the performance of the filters in post-processing. The datasets are bidirectional (all nodes broadcast and receive data packets), time-synchronized, and recorded to disk. As a result, we are able to selectively ignore certain measurements to artificially create the different experimental conditions, such as limiting the server's access to GPS measurements. Scenario A, as depicted in Fig. 4, is performed by the ship supporting Iver28, where Iver28 uses none of its GPS measurements. Scenario B, as also depicted in Fig. 4, is performed by Iver28 supporting Iver31. In this scenario, Iver28 only incorporates GPS measurements during the portions of its survey shown in green in Fig. 6.

TABLE II
AVERAGE NORM OF THE DIFFERENCE BETWEEN CEIF AND DEIF

Trial	**Avg. Norm [m]**	**Avg. Norm at TOAs [m]**
Sim: Scenario A	0.129	1.0E-6
Exp: Scenario A	0.056	2.0E-6
Sim: Scenario B	0.013	1.7E-4
Exp: Scenario B	0.056	3.0E-5

C. Equivalency of DEIF and Centralized Extended Information Filter

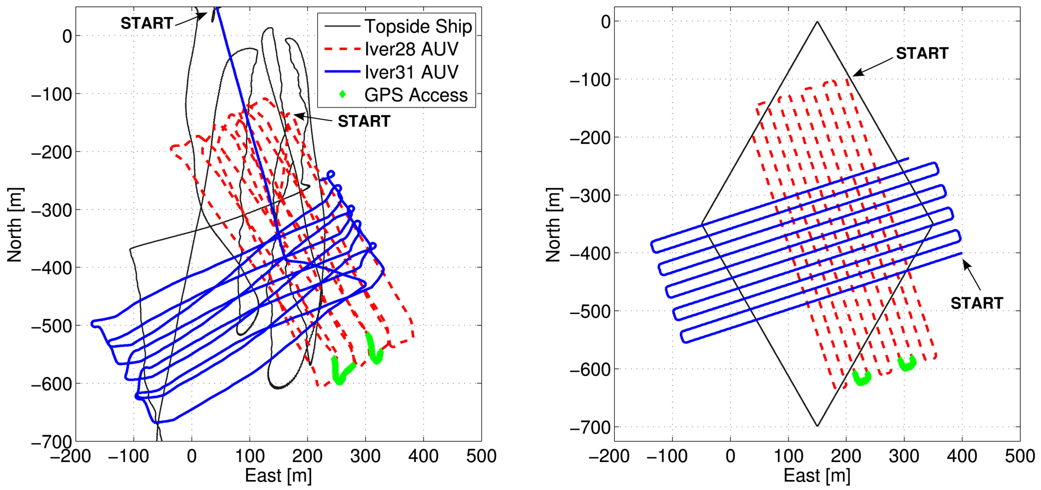
Fig. 7 shows the norm of the difference of the state vector, i.e., $\| \boldsymbol{x}_{c_{\mathrm{DEIF}}} - \boldsymbol{x}_{c_{\mathrm{CEIF}}} \|_2$, comparing the estimated mean of the client state vector in the DEIF versus the CEIF over the course of the survey. The four subplots show results for simulations in Fig. 7(a) and experiments in Fig. 7(b), where Scenario A is on the left and Scenario B is on the right. The lower plot of each of the four subplots highlights the norm of the difference immediately after the range measurements (marked by asterisks). The same data are shown, but the ordinates on the lower plots have been scaled by several orders of magnitude to show the precision with which the DEIF is able to reproduce the results of the CEIF at the TOA.

As discussed in Section III-E, the DEIF is specifically designed to produce state estimates that are comparable with the CEIF *immediately after* each range update; *between* range updates the estimates differ primarily due to the smoothing effect of subsequent server sensor measurements. The results from both the simulation and the experiment (see Fig. 7) show these effects, where asterisks (*) mark the times immediately after range measurement updates. Table II shows the average norm of the difference between the CEIF and the DEIF over the course of the survey for the different trials. The small variations between the filters immediately after range updates are on the order of fractions of a millimeter and are accounted for by round-off errors associated with numerical precision.

The difference in the filter estimates between acoustic broadcasts is larger in Scenario A (left plots in Fig. 7) because, in the CEIF, absolute position measurements from GPS continuously drive down the uncertainty in both the client and the server. In contrast, in the DEIF, there is no additional absolute position information between acoustic broadcasts, creating a difference between the filters *between* acoustic broadcasts. In Scenario B (right plots in Fig. 7), neither filter has access to absolute position measurements, except during two discrete intervals. Thus, the filters' uncertainty estimates are more closely matched for all time steps.

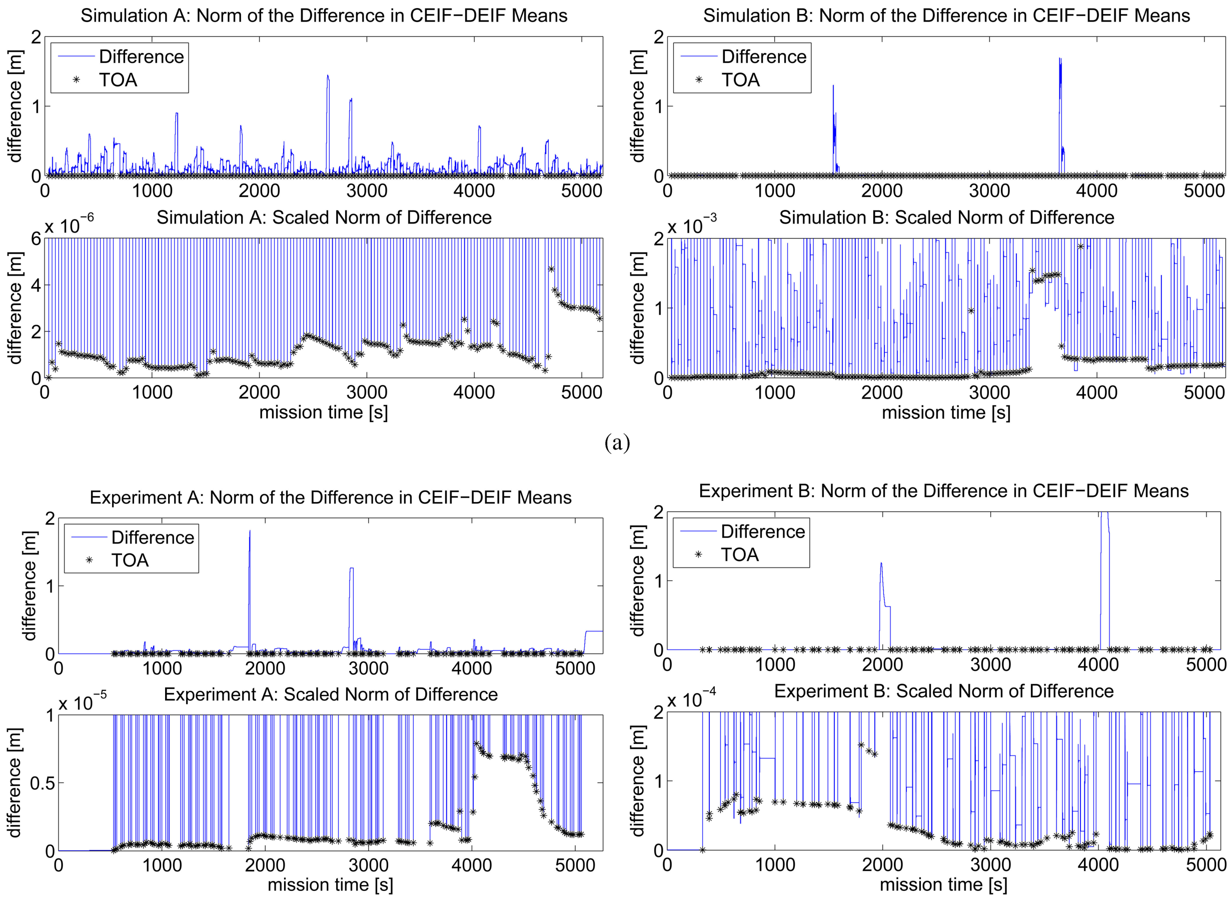
D. Filter Performance Comparison

We use two different metrics to compare the performance of the filters: 3-σ uncertainty of the client position estimate and normalized estimation error squared (NEES) averaged over ten runs. The 3-σ bounds demonstrate each filter's ability to localize the client vehicle and provide a notion of consistency when compared with the CEIF. The CEIF is our gold-standard because it has access to all sensor measurements of all vehicles



(a) (b)

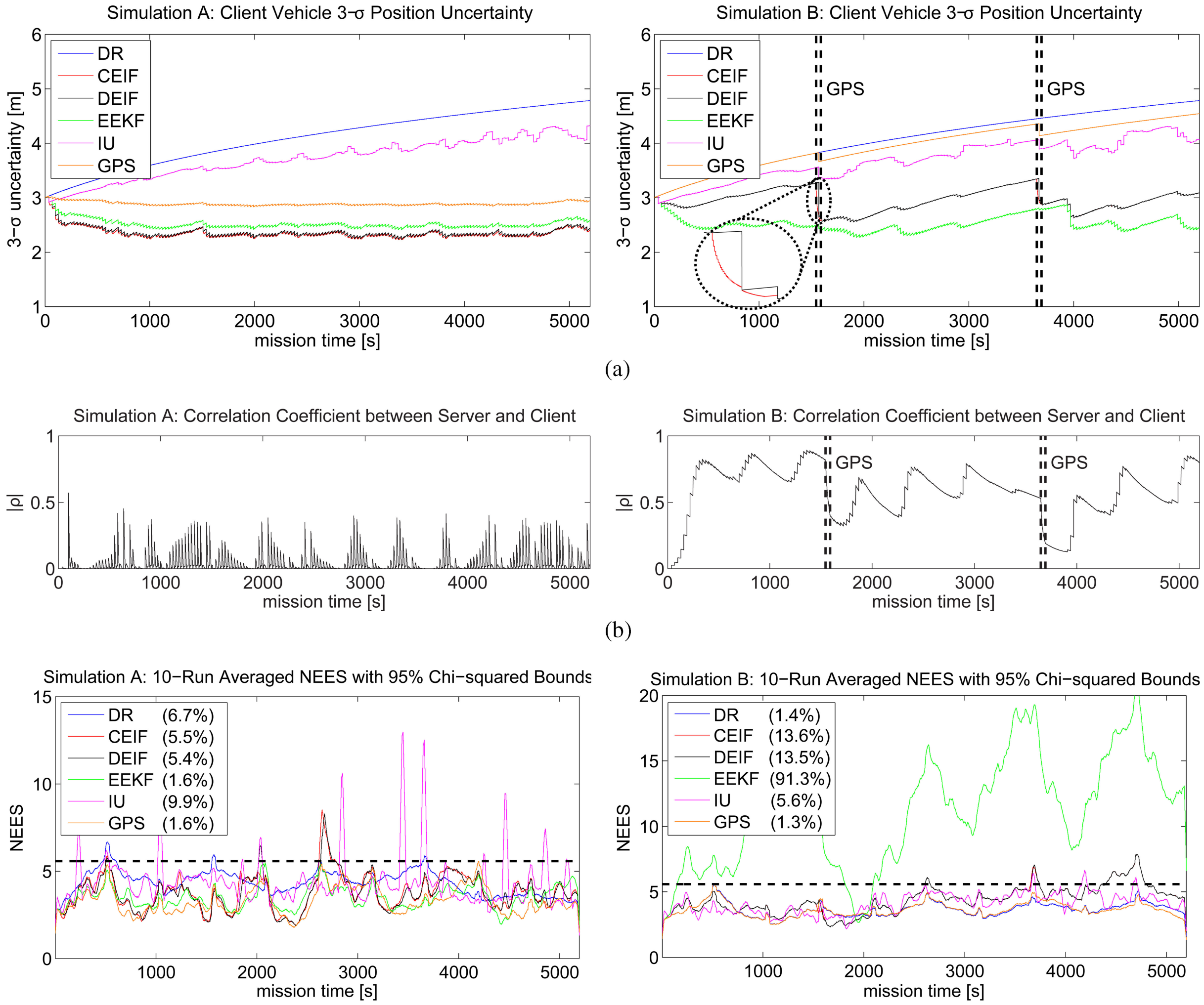
Fig. 6. Trajectories of the topside platform (a ship) and two AUVs during our field experiment (a) and in the simulated dataset (b). Sections of Iver28's trajectory shown in green indicate where its GPS measurements were incorporated into the filters during operational Scenario B (Iver28 supporting Iver31).



(a)

(b)

Fig. 7. Norm of the difference in client state between DEIF and CEIF for simulations (a) and experiments (b). The left column in each subfigure represents Scenario A (ship with continuous access to GPS supporting Iver28); the right column represents Scenario B (Iver28 with intermittent access to GPS supporting Iver31). The lower plots in all subfigures provide a scaled view of the ordinate axis for visual clarity. Note that in all cases, the CEIF and DEIF are numerically equivalent at the TOA.



(a)

(b)

(c)

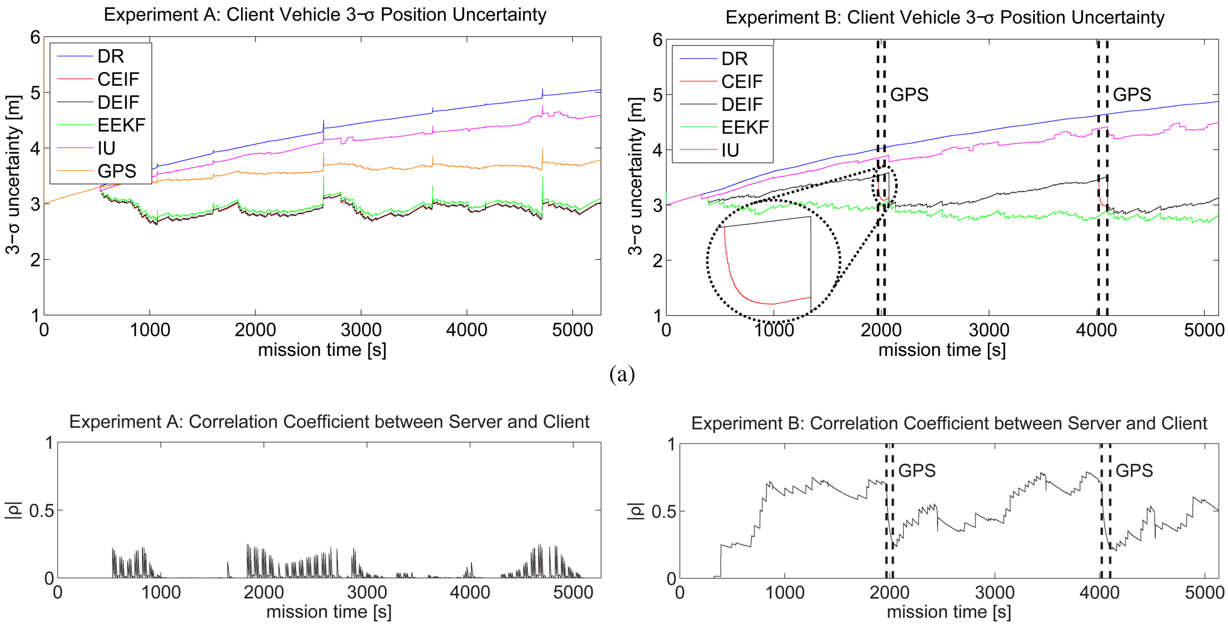
Fig. 8. Simulation results for Scenario A (ship supporting Iver28) and Scenario B (Iver28 supporting Iver31) on the left and right, respectively. (a) Filter estimated uncertainty. The inset in Scenario B shows where the DEIF uncertainty estimate differs briefly from the CEIF during the time when GPS was available. (b) Correlation coefficient (absolute value) between the server and the client, as calculated by the CEIF. The development of significant correlation between the server and the client is evident in Scenario B. (c) Ten-run average NEES. The horizontal black line indicates the one-sided χ^2 95% confidence bound. Note that the displayed NEES are smoothed with a 1-min moving average filter for ease of visibility and comparison. The percentages in the legend represent the amount of time the (nonsmoothed) NEES for each filter exceeds the 95% confidence bound.

and, therefore, should have the lowest valid uncertainty estimate. The NEES provides a true measure of filter consistency based on whether the estimation errors of the filter are zero mean and have magnitudes commensurate with the covariance of the filter [50]. Because the NEES can only be calculated when the true state of the system is known, we calculate it for the simulated data only. The performance of the filters is summarized here and discussed in detail below for each filter.

Figs. 8 and 9 illustrate the results for all of the filters evaluated, using simulated and experimental data, respectively. Both cases show similar trends. Subfigures (a) in both Figs. 8 and 9 show the 3-σ uncertainty estimate for each of the filters evaluated. The DEIF has the lowest consistent uncertainty estimate and most closely matches the CEIF in both scenarios. The EEKF shows the next smallest uncertainty estimate in Scenario A but produces an overly optimistic uncertainty estimate in Scenario B that is not consistent with the CEIF. The raw GPS method produces an uncertainty estimate that is larger than the EEKF in Scenario A, and only a marginal improvement over DR in Scenario B, because range measurements are only incorporated when the server AUV has access to GPS measurements at the surface. The IU algorithm produces an uncertainty estimate that is the largest of the filters evaluated for Scenario A. The IU's uncertainty estimate shows an improvement over DR but is unbounded in time.

Subfigures (b) in both Figs. 8 and 9 show the absolute correlation coefficient between the server and the client in the CEIF over the course of the survey (Scenario A). The development of significant correlation between the server and client is evident in Scenario B. The EEKF's inconsistent estimate results from ignoring this correlation (discussed below).

Fig. 8(c) compares the NEES for all filters against the 95% confidence bound. Note that the NEES for each filter has been smoothed over a 1-min window for readability. None of the



(a)

(b)

Fig. 9. Experimental results for Scenario A (ship supporting Iver28) and Scenario B (Iver28 supporting Iver31) on the left and right, respectively. (a) Filter estimated uncertainty. The inset in Scenario B shows where the DEIF uncertainty estimate differs briefly from the CEIF during the time when GPS was available. (b) Correlation coefficient (absolute value) between the server and the client, as calculated by the CEIF. The development of significant correlation between the server and the client is evident in Scenario B.

filters deviate far from the 95% chi-squared bounds, except for the EEKF, which clearly produces an inconsistent estimate in Scenario B when significant correlation between the server and the client exists. Spikes in the NEES are clearly visible in the IU, CEIF, and DEIF during vehicle turns at the end of the tracklines. These are caused by the constant-velocity, linear process model not accurately modeling turn dynamics. This, coupled with the receipt of OWTTs measurements during turns (which drives uncertainty down during the uncertain turn event), leads the filter to temporarily be more confident in its position than is accurate.

1) Decentralized Extended Information Filter Results: The DEIF navigation estimate onboard the client vehicle matches the CEIF estimate within numerical precision at the TOA, as seen in Fig. 7, for both simulated and experimental scenarios. In Scenario B, Iver31's uncertainty estimate differs briefly from the CEIF during the time when GPS was available, as shown in the insets of Figs. 8 and 9. Once surfaced, Iver28 receives several GPS measurements, which immediately drive the uncertainty down for both vehicles in the CEIF; however, it is not until the next acoustic broadcast from the server vehicle (Iver28) that the client platform's DEIF (Iver31) is able to incorporate the delta information with GPS and fully match the CEIF. The CEIF and DEIF are identical immediately after each range measurements but differ between broadcasts for as long as Iver28 is receiving GPS measurements. Note that in the simulated survey [see Fig. 8(a)], the difference between the DEIF and the CEIF occurs across multiple acoustic broadcasts; this is because the simulation does not prevent acoustic broadcasts during the receipt of GPS measurements.

2) Egocentric Extended Kalman Filter Results: The EEKF uncertainty estimate closely matches the CEIF in Scenario A when the correlation between the server and the client is small. However, in Scenario B, both the simulation and experiment clearly show that the EEKF fails to produce an estimate consistent with the CEIF. This is a result of the large correlation that persists between each vehicle's position estimate causing the client vehicle's EEKF to double-count successive range measurements, as described in Section IV-B. Relative range measurements create large correlation between vehicles. Conversely, absolute position observations tend to destroy correlation between previously correlated vehicles, which is why the EEKF uncertainty estimate more closely matches the CEIF in Scenario A. (See [13, App.] for a simplified example illustrating this effect due to GPS.) The NEES computed in simulation from Scenario B clearly demonstrates the inconsistency of this filter in situations where vehicle estimates are highly correlated.

3) Raw Global Positioning System Results: The raw GPS method in Scenario A results in a bounded-error position estimate, with 3-σ uncertainty closer to the CEIF than the IU algorithm, in both the simulation and the experiment. In Scenario B, this method is an improvement over DR, but not significantly because the server vehicle only sends range packets during two short intervals when GPS is available. Note that results from this method are not shown for the experimental trial of Scenario B in Fig. 4 because the transducer on the AUV is above the waterline when the vehicle is at the surface receiving GPS, and is, therefore, unable to transmit range packets. The drawbacks to this method are that it cannot achieve the lowest uncertainty estimate without filtering topside measurements (as with the CEIF) and that the server must have had recent access to a GPS fix in order to execute a range measurement. However, this algorithm is trivially robust to

TABLE III
OPERATIONAL PACKET LOSS STATISTICS

Communication Channel A⇌B			**% Packet Loss** A→B	A←B
Ship	⇌	Iver31	15.3%	27.9%
Ship	⇌	Iver28	**19.0%***	32.4%
Iver28	⇌	Iver31	**34.4%***	41.4%

*These communication paths are the ones used to run the DEIF.

packet loss or failure of the server node, and the client is able to incorporate measurements from several independent servers without modification and arrive at a consistent estimate.

4) Interleaved Update Algorithm Results: The IU algorithm shows improvement over dead-reckoning with an uncertainty estimate that is guaranteed to be consistent, as shown by its NEES. However, the filter's estimate exhibits unbounded growth in uncertainty over time. As mentioned in Section IV-B3, this is expected for a two-vehicle topology with unidirectional communication.

VI. DISCUSSION

In this section, we discuss several topics related to the operational implementation of the DEIF: packet loss, filter telemetry requirements, and when linear process models are required. We also discuss possible (nonoptimal) extensions of the DEIF approach described herein to multi-vehicle topologies.

A. Packet Loss

The DEIF approach described herein relies on the sequential broadcast and receipt of delta information packets. In practice, the nonlossy communication assumption cannot be met, as packets are routinely lost to an often faulty acoustic communication channel, making packet loss an operational concern for real-time implementation. In the experimental dataset used in this study, only successful acoustic broadcasts were used in post-processing to run the filter. Table III tabulates the packet loss statistics for each of the different communication paths. The two entries in bold represent the communication paths used to run the DEIF experiments reported here.

A number of possible solutions exist to address the issue of packet loss when implementing the DEIF in real time for use in the field. We describe several proposed methods below: 1) sending redundant information, 2) utilizing acknowledgments from the client platform, and 3) an alternative formulation of the delta information packet. In practice, we expect these and other solutions to be developed and refined over time as this algorithm matures and is used in the field.

1) Redundant Information Packets: Our first safeguard against dropped acoustic packets is to broadcast redundant information. A delta information message describes a transition of the server state between consecutive TOLs, in particular, the delta information at the nth TOL relates the server state from time TOL_{n-1} to time TOL_n, abbreviated $\Delta_{s_{\mathsf{TOL}_{n-1:n}}}$. Since we cannot rely on the client receiving every packet, we require the server to broadcast delta packets corresponding to transitions from the last k TOLs to the current TOL: $\Delta_{s_{\mathsf{TOL}_{n-1:n}}}, \ldots, \Delta_{s_{\mathsf{TOL}_{n-k:n}}}$. In this case, the server easily computes the delta packets, as in (23) and (24), for each delta relation by marginalizing out intermediate states. For example, the server vehicle tracks state

$$\mathbf{x}_{s_k} = \left[\boldsymbol{x}_{s_k}^\top, \boldsymbol{x}_{s_{\mathsf{TOL}_n}}^\top, \ldots, \boldsymbol{x}_{s_{\mathsf{TOL}_{n-k}}}^\top\right]^\top.$$

To compute the delta packet corresponding to the transition $\Delta_{s_{\mathsf{TOL}_{n-k:n}}}$, we simply marginalize out states corresponding to $\mathsf{TOL}_{n-1}, \ldots, \mathsf{TOL}_{n-k+1}$ and calculate the delta information as per usual.

2) Client Acknowledgment: In practice, subsea vehicles typically send some minimal acoustic state data to report general mission health. We propose to encode the last server TOL state received, which is denoted TOL_ℓ, into each client state packet sent to the server. Under this scheme, in addition to the standard one-step delta information packet, i.e., $\Delta_{s_{\mathsf{TOL}_{n-1:n}}}$, the server would also broadcast the delta information relative to the last known good TOL, i.e., $\Delta_{s_{\mathsf{TOL}_{\ell:n}}}$. This approach allows for the client vehicle to resume normal usage should it ever miss a regular one-step packet.

3) Alternate Packet Composition: Recently, Walls and Eustice [55] proposed an alternate packet formulation, dubbed the origin-state method, which allows the client vehicle to reconstruct the server information matrix in a way that is robust to packet loss. While this approach solves a different problem than the DEIF, the authors of [55] have applied their algorithm to a modified DEIF implementation and have shown it to be robust to a lossy acoustic channel, subject to certain operational restrictions.

B. Filter Telemetry Requirements

Because of the severe constraints imposed on data packet size by the limited capacity of the acoustic channel, we include a brief discussion of the telemetry requirements for the different algorithms compared in this paper. The amount of data that can be transmitted in an acoustic data packet depend on the carrier frequency, bandwidth, and encoding method of the signal, as well as the characteristics of the local sound channel [11]. Currently, the WHOI Micro-Modem supports data rates ranging from a single 32-byte frame per packet (rate 0, encoded with frequency shift keying) to eight 256-byte frames per packet (rate 5, encoded with phase-shift keying) [5]. Assuming one broadcast every 15 s, the resulting maximum throughput varies from 128 bytes/min to 8 kb/min. The experiments, as described in Section V-A, used rate 0.

The relative data packet size demanded by each filter is summarized in Table IV for a server–client network topology. We place a small overhead on each broadcast by including depth information because we project range measurements into the local-level plane. Both the EEKF and IU only require that local state and covariance corresponding to x, y position be encoded, since each measurement is considered independent, and the client filter only needs access to elements of the state involved in the measurement update. Therefore, taking advantage of symmetry, the EEKF and IU require transmitting two floats for mean and three for covariance in the acoustic broadcast. Note that the telemetry requirements for IU implemented in

TABLE IV
TELEMETRY PAYLOAD REQUIREMENTS FOR A SERVER–CLIENT TOPOLOGY

Estimation Method	**Floats**†	**Bytes**‡
DEIF	45	180
DEIF*	15	60
IU	6	24
EEKF	6	24
Raw GPS	3	12

† Includes one float for depth.
‡Assumes single precision (4 byte) floats (this will vary depending on how many bits are used to encode the float).
*Alternate DEIF implementation discussed in the text.

an n-vehicle network with bidirectional communication grows with $\mathcal{O}(n^2)$. If the uncertainty of the GPS measurement is known by the vehicle beforehand (a reasonable assumption), only two floats for the x, y GPS position must be transmitted by the acoustic broadcast in the raw GPS method. Delta state in the DEIF is computed between the last TOL augmented state and the current state; therefore, the DEIF requires eight floats (2×4 for state dimension four) for the delta information vector and 36 for the delta information matrix in each data packet.

The DEIF* in the table refers to a DEIF in which the server state contains only world-frame x, y position, further reducing the state model. In this case, the process model follows an odometry-driven control input (e.g., integrated velocity measurements) and white-noise. This implementation is the subject of current research, and preliminary results, which are currently under review, show that it achieves a far lower telemetry payload at the cost of a less confident estimate.

C. Linear Process Models

As noted in Section III-F, the client process model is *not* required to be linear, but the server process model and observation models must be linear for the DEIF to identically reproduce the results of the CEIF. For the implementation of the DEIF described herein, we employ a linear process model for the client as well as the server for several reasons. For DOFs that are well instrumented, compared with the actual system dynamics, a common simplification of the process model is to exclude those well-instrumented states from the set of estimated states. For the AUVs used in this experiment, and many AUVs on the market, attitude and depth are well instrumented compared with the dynamics of the vehicle [56]. In addition, Webster *et al.* [12] verified that the majority of the uncertainty resides in x, y position elements of the client state vector. Therefore, the use of linear models for the client is a reasonable assumption.

A simplified process model for the server, in the case where the server is a ship, is justified by the server's continuous access to GPS measurements. In addition, the simplified server process model reduces the amount of data required to broadcast state information from the server to the client, an important operational concern given the limited capacity of the underwater acoustic channel [28]. Extending the DEIF framework to accommodate a nonlinear server, while continuing to ensure identical results between the CEIF and DEIF, would require that the client transmit a new linearization point to the server following a range measurement update.

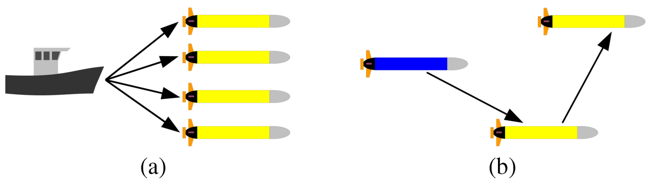


Fig. 10. Two multi-vehicle topologies that are simple (suboptimal) extensions of the DEIF: (a) the multi-client topology where a single server supports multiple clients and (b) the cascaded network topology where clients also perform as servers.

D. Multi-vehicle Topologies

Two suboptimal multi-vehicle estimation topologies that are simple extensions of the DEIF are shown in Fig. 10. The first is a multi-client topology, in which a single server supports multiple clients. The second is a cascaded topology, in which the server–client model described in this paper is extended such that the client supports one or more additional subsea clients.

Extending the two-vehicle DEIF presented in this paper to either multi-vehicle topology is trivial. In the multi-client topology shown in Fig. 10(a), the server performs a single acoustic broadcast, and each client runs its own independent DEIF with no knowledge of the other clients. This is scalable to as many vehicles as can operate within acoustic range of the server. Note, however, that in this case, the DEIF does not reproduce the distribution produced by a single global CEIF that tracks correlation between *all* vehicles. Instead, each DEIF is equivalent to a corresponding two-vehicle CEIF that tracks only the server and an individual client.

In the serially connected network topology shown in Fig. 10(b), unidirectional communication of position information is a simple, cascaded implementation of two DEIFs. Note that clients at the end of a serially connected network will have access to all range information in the system and will be able to exactly reproduce the results of the CEIF. However, intermediate clients will not, and, therefore, will not be able to exactly reproduce the results of the CEIF.

The performance of the DEIF compared with other algorithms presented in the literature has been investigated by Walls and Eustice [13] for these topologies using experimental data from a three-node network. Packet loss, as noted in Section VI-A, remains an operational concern for both of these extensions.

VII. CONCLUSION AND FUTURE WORK

This paper has presented a detailed derivation of the DEIF algorithm for the synchronous-clock acoustic navigation of a single subsea client vehicle. We showed analytically that the DEIF exactly reproduces the estimate of the corresponding server–client CEIF at the TOA. Simulation and experimental trials validated the effectiveness of the DEIF at consistently localizing the client vehicle, as well as reproducing the CEIF. A comparative analysis demonstrated the performance of the DEIF and several previously reported approaches to cooperative acoustic navigation. Additionally, we showed that the DEIF performs favorably when compared with previously reported decentralized acoustic cooperative localization algorithms.

While the DEIF achieves excellent theoretical results, a real-time implementation requires additional overhead to address packet loss, as described in Section VI. Current on-going work continues to investigate more robust information descriptions that can handle dropped acoustic transmissions, such as the preliminary work reported in [55]. Furthermore, we seek an algorithm that can scale to larger networks and share information bidirectionally to accommodate many AUVs operating over very large operational areas.

REFERENCES

[1] R. M. Eustice, L. L. Whitcomb, H. Singh, and M. Grund, "Recent advances in synchronous-clock one-way-travel-time acoustic navigation," in *Proc. IEEE/MTS OCEANS Conf. Exhib.*, Boston, MA, USA, Sep. 2006, pp. 1–6.

[2] R. M. Eustice, H. Singh, and L. L. Whitcomb, "Synchronous-clock one-way-travel-time acoustic navigation for underwater vehicles," *J. Field Robot.*, vol. 28, no. 1, pp. 121–136, 2011.

[3] M. Hunt, W. Marquet, D. Moller, K. Peal, W. Smith, and R. Spindel, "An acoustic navigation system," Woods Hole Ocean. Inst., Woods Hole, MA, USA, Tech. Rep. WHOI-74-6, Dec. 1974.

[4] P. Milne, *Underwater Acoustic Positioning Systems*. Houston, TX, USA: Gulf, 1983.

[5] L. Freitag, M. Grund, S. Singh, J. Partan, P. Koski, and K. Ball, "The WHOI micro-modem: An acoustic communications and navigation system for multiple platforms," in *Proc. IEEE/MTS OCEANS Conf. Exhib.*, Washington, DC, USA, Sep. 2005, pp. 1086–1092.

[6] L. Freitag, M. Grund, J. Partan, K. Ball, S. Singh, and P. Koski, "Multi-band acoustic modem for the communications and navigation aid AUV," in *Proc. IEEE/MTS OCEANS Conf. Exhib.*, Washington, DC, USA, Sep. 2005, pp. 1080–1085.

[7] R. M. Eustice, L. L. Whitcomb, H. Singh, and M. Grund, "Experimental results in synchronous-clock one-way-travel-time acoustic navigation for autonomous underwater vehicles," in *Proc. IEEE Int. Conf. Robot. Autom.*, Rome, Italy, Apr. 2007, pp. 4257–4264.

[8] S. E. Webster, R. M. Eustice, H. Singh, and L. L. Whitcomb, "Advances in single-beacon one-way-travel-time acoustic navigation for underwater vehicles," *Int. J. Robot. Res.*, no. 8, pp. 935–950, Jul. 2012.

[9] S. E. Webster, R. M. Eustice, H. Singh, and L. L. Whitcomb, "Preliminary deep water results in single-beacon one-way-travel-time acoustic navigation for underwater vehicles," in *Proc. IEEE/RSJ Int. Conf. Intell. Robot. Syst.*, St. Louis, MO, USA, Oct. 2009, pp. 2053–2060.

[10] D. Kilfoyle and A. Baggeroer, "The state of the art in underwater acoustic telemetry," *IEEE J. Ocean. Eng.*, vol. 25, no. 1, pp. 4–27, Jan. 2000.

[11] S. Singh, S. E. Webster, L. Freitag, L. L. Whitcomb, K. Ball, J. Bailey, and C. Taylor, "Acoustic communication performance of the WHOI Micro-Modem in sea trials of the *Nereus* vehicle to 11,000 m depth," in *Proc. IEEE/MTS OCEANS Conf. Exhib.*, Biloxi, MS, USA, Oct. 2009, pp. 1–6.

[12] S. E. Webster, L. L. Whitcomb, and R. M. Eustice, "Preliminary results in decentralized estimation for single-beacon acoustic underwater navigation," in *Proc. Robot. Sci. Syst. Conf.*, Zaragoza, Spain, Jun. 2010.

[13] J. M. Walls and R. M. Eustice, "Experimental comparison of synchronous-clock cooperative acoustic navigation algorithms," in *Proc. IEEE/MTS OCEANS Conf. Exhib.*, Kona, HI, USA, Sep. 2011, pp. 1–7.

[14] H. Mu, T. Bailey, P. Thompson, and H. Durrant-Whyte, "Decentralised solutions to the cooperative multi-platform navigation problem," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 47, no. 2, pp. 1433–1449, Apr. 2011.

[15] T. Song, "Observability of target tracking with range-only measurements," *IEEE J. Ocean. Eng.*, vol. 24, no. 24, pp. 383–387, Jul. 1999.

[16] A. Gadre and D. Stilwell, "A complete solution to underwater navigation in the presence of unknown currents based on range measurements from a single location," in *Proc. IEEE/RSJ Int. Conf. Intell. Robot. Syst.*, Edmonton, AB, Canada, Aug. 2005, pp. 1420–1425.

[17] X. Zhou and S. Roumeliotis, "Robot-to-robot relative pose estimation from range measurements," *IEEE Trans. Robot.*, vol. 24, no. 6, pp. 1379–1393, Dec. 2008.

[18] M. F. Fallon, G. Papadopoulos, J. J. Leonard, and N. M. Patrikalakis, "Cooperative AUV navigation using a single maneuvering surface craft," *Int. J. Robot. Res.*, vol. 29, no. 12, pp. 1461–1474, Oct. 2010.

[19] A. Scherbatyuk, "The AUV positioning using ranges from one transponder LBL," in *Proc. IEEE/MTS OCEANS Conf. Exhib.*, San Diego, CA, USA, Oct. 1995, vol. 4, pp. 1620–1623.

[20] P. Baccou and B. Jouvencel, "Homing and navigation using one transponder for AUV, postprocessing comparisons results with long base-line navigation," in *Proc. IEEE Int. Conf. Robot. Autom.*, Washington, DC, USA, May 2002, vol. 4, pp. 4004–4009.

[21] M. B. Larsen, "Autonomous navigation of underwater vehicles," Ph.D. dissertation, Tech. Univ. Denmark, Kongens Lyngby, Denmark, Feb. 2001.

[22] S. McPhail and M. Pebody, "Range-only positioning of a deep-diving autonomous underwater vehicle from a surface ship," *IEEE J. Ocean. Eng.*, vol. 34, no. 4, pp. 669–677, Oct. 2009.

[23] C. P. Morice and S. M. Veres, "Geometric bounding techniques for underwater localization using range-only sensors," *Proc. Inst. Mech. Eng. Part I: J. Syst. Contr. Eng.*, vol. 225, no. 1, pp. 74–84, 2011.

[24] S. Roumeliotis and G. Bekey, "Distributed multirobot localization," *IEEE Trans. Robot. Autom.*, vol. 18, no. 5, pp. 781–795, Oct. 2002.

[25] A. Howard, "Multi-robot simultaneous localization and mapping using particle filters," *Int. J. Robot. Res.*, vol. 25, no. 12, pp. 1243–1256, 2006.

[26] B. Kim, M. Kaess, L. Fletcher, J. Leonard, A. Bachrach, N. Roy, and S. Teller, "Multiple relative pose graphs for robust cooperative mapping," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2010, pp. 3185–3192.

[27] V. Indelman, P. Gurfil, E. Rivlin, and H. Rotstein, "Graph-based distributed cooperative navigation," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2011, pp. 4786–4791.

[28] J. Partan, J. Kurose, and B. N. Levine, "A survey of practical issues in underwater networks," *SIGMOBILE Mob. Comput. Commun. Rev.*, vol. 11, pp. 23–33, Oct. 2007.

[29] A. Ribeiro, G. B. Giannakis, and S. I. Roumeliotis, "SOI-KF: Distributed Kalman filtering with low-cost communications using the sign of innovations," *IEEE Trans. Signal Process.*, vol. 54, no. 12, pp. 4782–4795, Dec. 2006.

[30] E. D. Nerurkar, K. X. Zhou, and S. I. Roumeliotis, "Hybrid estimation framework for multi-robot cooperative localization using quantized measurements," Res. Note, Dept. Comput. Sci. Eng., Univ. Minnesota, Minneapolis, MN, USA, 2011.

[31] T. Bailey, M. Bryson, H. Mu, J. Vial, L. McCalman, and H. Durrant-Whyte, "Decentralised cooperative localisation for heterogeneous teams of mobile robots," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2011, pp. 2859–2865.

[32] T. Bailey and H. Durrant-Whyte, "Decentralised data fusion with delayed states for consistent inference in mobile ad hoc networks," Australian Centre for Field Robotics, Univ. Sydney, Sydney, N.S.W., Australia, Tech. Rep., 2007. (see http://www-personal.acfr.usyd.edu.au/tbailey/techreports/index.html)

[33] R. Aragues, J. Cortes, and C. Sagues, "Distributed consensus on robot networks for dynamically merging feature-based maps," *IEEE Trans. Robot.*, vol. 28, no. 4, pp. 840–854, Aug. 2012.

[34] D. Maczka, A. Gadre, and D. Stilwell, "Implementation of a cooperative navigation algorithm on a platoon of autonomous underwater vehicles," in *Proc. IEEE/MTS OCEANS Conf. Exhib.*, Vancouver, BC, Canada, Oct. 2007, pp. 1–6.

[35] A. Bahr, M. Walter, and J. Leonard, "Consistent cooperative localization," in *Proc. IEEE Int. Conf. Robot. Autom.*, Kobe, Japan, May 2009, pp. 3415–3422.

[36] A. Bahr, J. J. Leonard, and M. F. Fallon, "Cooperative localization for autonomous underwater vehicles," *Int. J. Robot. Res.*, vol. 28, no. 6, pp. 714–728, Jun. 2009.

[37] J. Vaganay, J. Leonard, J. Curcio, and J. Willcox, "Experimental validation of the moving long base-line navigation concept," in *Proc. IEEE/OES Autonom. Underwater Veh.*, Jun. 2004, pp. 59–65.

[38] M. F. Fallon, G. Papadopoulos, and J. J. Leonard, "Cooperative AUV navigation using a single surface craft," in *Proc. Field Service Robot.*, Cambridge, MA, USA, Jul. 2009, pp. 1–10.

[39] M. Fallon, G. Papadopoulos, and J. Leonard, "A measurement distribution framework for cooperative navigation using multiple AUVs," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2010, pp. 4256–4263.

[40] A. G. O. Mutambara, *Decentralized Estimation and Control for Multisensor Systems*. Boca Raton, FL, USA: CRC, 1998.

[41] M. Bozorg, E. Nebot, and H. Durrant-Whyte, "A decentralised navigation architecture," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 1998, vol. 4, pp. 3413–3418.

[42] S. Thrun and Y. Liu, "Multi-robot SLAM with sparse extended information filers," in *Robotics Research*, (ser. Springer Tracts in Advanced

Robotics), P. Dario and R. Chatila, Eds. Berlin, Heidelberg, Germany: Springer-Verlag, 2005, vol. 15, pp. 254–266.

[43] S. Reece and S. Roberts, "Robust, low-bandwidth, multi-vehicle mapping," in *Proc. Int. Conf. Inf. Fusion*, Jul. 2005, vol. 2, pp. 1319–1326.

[44] S. Grime, H. F. Durrant-Whyte, and P. Ho, "Communication in decentralized data-fusion systems," in *Proc. Amer. Control Conf.*, Jun. 1992, pp. 3299–3303.

[45] R. M. Eustice, H. Singh, and J. J. Leonard, "Exactly sparse delayed-state filters for view-based SLAM," *IEEE Trans. Robot.*, vol. 22, no. 6, pp. 1100–1114, Dec. 2006.

[46] J. Diosdado and I. Ruiz, "Decentralised simultaneous localisation and mapping for AUVs," in *Proc. IEEE OCEANS Eur. Conf. Exhib.*, Jun. 2007, pp. 1–6.

[47] F. Dellaert and M. Kaess, "Square root SAM: Simultaneous localization and mapping via square root information smoothing," *Int. J. Robot. Res.*, vol. 25, no. 12, pp. 1181–1204, Dec. 2006.

[48] M. Kaess, A. Ranganathan, and F. Dellaert, "iSAM: Incremental smoothing and mapping," *IEEE Trans. Robot.*, vol. 24, no. 6, pp. 1365–1378, Dec. 2008.

[49] M. Fallon, M. Kaess, H. Johannsson, and J. Leonard, "Efficient AUV navigation fusing acoustic ranging and side-scan sonar," in *Proc. IEEE Int. Conf. Robot. Autom.*, Shanghai, China, May 2011, pp. 2398–2405.

[50] Y. Bar-Shalom, X. R. Li, and T. Kirubarajan, *Estimation With Applications to Tracking and Navigation*. New York, NY, USA: Wiley, 2001.

[51] S. E. Webster, R. M. Eustice, C. Murphy, H. Singh, and L. L. Whitcomb, "Toward a platform-independent acoustic communications and navigation system for underwater vehicles," in *Proc. IEEE/MTS OCEANS Conf. Exhib.*, Biloxi, MS, USA, Oct. 2009, pp. 1–7.

[52] D. A. Smallwood and L. L. Whitcomb, "Model-based dynamic positioning of underwater robotic vehicles: Theory and experiment," *IEEE J. Ocean. Eng.*, vol. 29, no. 1, pp. 169–186, Jan. 2004.

[53] L. L. Whitcomb, D. R. Yoerger, H. Singh, and J. Howland, "Advances in underwater robot vehicles for deep ocean exploration: navigation, control, and survey operations," in *Robotics Research: The Ninth International Symposium*, J. Hollerbach and D. Koditschek, Eds. London, U.K.: Springer-Verlag, 1999, ch. 13, pp. 439–448.

[54] H. Brown, A. Kim, and R. Eustice, "Development of a multi-AUV SLAM testbed at the University of Michigan," in *Proc. IEEE/MTS OCEANS Conf. Exhib.*, Quebec City, QC, Canada, Sep. 2008, pp. 1–6.

[55] J. M. Walls and R. M. Eustice, "An origin state method for lossy synchronous-clock acoustic navigation," in *Proc. IFAC Workshop Navigat. Guidance Control Underwater Veh.*, Porto, Portugal, Apr. 2012.

[56] J. C. Kinsey, R. M. Eustice, and L. L. Whitcomb, "A survey of underwater vehicle navigation: Recent advances and new challenges," in *Proc. IFAC Conf. Manoeuvr. Control Marine Craft*, Lisbon, Portugal, Sep. 2006.

Sarah E. Webster (S'08–M'12) received the B.S. degree in mechanical engineering from the Massachusetts Institute of Technology, Cambridge, MA, USA, in 2000 and the Ph.D. degree in mechanical engineering from the Johns Hopkins University, Baltimore, MD, USA, in 2010.

She is currently a postdoctoral scholar with the Ocean Physics Department, Applied Physics Laboratory, University of Washington, Seattle, WA, USA. Her research interests include navigation for autonomous underwater vehicles, decentralized estimation, and long-term autonomy.

Jeffrey M. Walls (S'11) received the B.S. degree in mechanical engineering from the University of Virginia, Charlottesville, VA, USA, in 2009 and the M.S. degree in mechanical engineering from the University of Michigan, Ann Arbor, MI, USA, in 2011. Currently, he is pursuing the Ph.D. degree with the Department of Mechanical Engineering, University of Michigan.

His research interest are in the areas of cooperative navigation and coordinated planning for autonomous underwater vehicles.

Louis L. Whitcomb (S'86–M'95–SM'02–F'11) received the Ph.D. degree in electrical engineering from Yale University, New Haven, CT, USA, in 1992.

He is currently a Professor with the Department of Mechanical Engineering, with secondary appointment in the Department of Computer Science, the Johns Hopkins University, Baltimore, MD, USA, where he is the Director of the Laboratory for Computational Sensing and Robotics. His research interests include the design, dynamics, and control of robotic systems.

Dr. Whitcomb is the Louis R. Sardella Faculty Scholar with the G.W.C. Whiting School of Engineering, The Johns Hopkins University.

Ryan M. Eustice (S'00–M'05–SM'10) received the B.S. degree in mechanical engineering from Michigan State University, East Lansing, MI, USA, in 1998 and the Ph.D. degree in ocean engineering from the Massachusetts Institute of Technology/Woods Hole Oceanographic Institution Joint Program, Woods Hole, MA, USA, in 2005.

Currently, he is an Assistant Professor with the Department of Naval Architecture and Marine Engineering, University of Michigan, Ann Arbor, MI, USA, with joint appointments in the Department of Electrical Engineering and Computer Science and in the Department of Mechanical Engineering. His research interests include autonomous navigation and mapping, computer vision and image processing, mobile robotics, and autonomous underwater vehicles.