An Origin State Method for Lossy Synchronous-clock Acoustic Navigation [⋆]

Jeffrey M. Walls [*] and **Ryan M. Eustice** [**]

[*] *Department of Mechanical Engineering*
[**] *Department of Naval Architecture and Marine Engineering*
University of Michigan,
Ann Arbor, MI 48109,
Email: jmwalls@umich.edu, eustice@umich.edu

Abstract: This paper examines the problem of localizing a network of underwater vehicles using inter-vehicle range observations derived from measuring the one-way-travel-time (OWTT) of acoustic broadcasts. We report the derivation of the novel origin state method; an algorithm for distributing a local pose-graph as a sequence of minimal bandwidth information packets that is robust to packet loss. We demonstrate the effectiveness of this algorithm as an extension of the decentralized extended information filter (DEIF) for synchronous-clock acoustic navigation through post-process implementation using a real-world data set.

Keywords: Robot navigation, Kalman filters, distributed models, covariance, state estimation

1. INTRODUCTION

Underwater vehicles typically fuse Doppler body-frame velocity, attitude, and pressure depth observations to produce dead-reckoned (DR) navigation solutions. Without absolute position measurements, (e.g., global positioning system (GPS)), DR position errors grows unbounded in time. Higher quality DR sensors are only capable of reducing the rate of uncertainty growth, therefore, alternative methods for constraining navigation errors are required.

Underwater acoustic navigation systems attain bounded-error navigation through range-only observations to beacons with known position. Range observations are obtained from measuring the time-of-flight (TOF) of acoustic signals and assuming a known sound speed profile. The long-baseline (LBL) navigation framework, for example, employs a network of fixed reference beacons to which vehicles can measure range (Hunt et al., 1974). LBL, however, limits the area of operations to the coverage footprint of the reference beacons. Furthermore, narrowband LBL lacks the ability to scale up to large groups of vehicles because only a single vehicle can interrogate the beacon network at any one time.

Synchronous-clock acoustic navigation (Eustice et al., 2011) is an application of cooperative localization, which involves a group of vehicles augmenting their position estimates with inter-platform range observations. Much like other acoustic navigation methods, receiving platforms measure one-way-travel-time (OWTT) range to a transmitting platform. The OWTT derived distance is a function of the transmitter position at the time-of-launch (TOL) and the receiver position at the time-of-arrival (TOA). Synchronous-clock acoustic navigation can be thought of as a moving long-baseline (MLBL) approach (Vaganay et al., 2004). Advantageously, synchronous-clock acoustic networks can scale to arbitrarily many vehicles because all vehicles within acoustic range of the transmitting platform passively receive a range measurement leading to constant time update rates. Several methods have been proposed to incorporate OWTT-derived relative-range observations into a navigation framework; each presenting a trade off between consistency, bandwidth, and robustness to lossy communication.

The rest of this paper is organized as follows. Section 2 reviews existing approaches to cooperative localization and synchronous-clock acoustic navigation. Next, Section 3 highlights the ability of the decentralized extended information filter (DEIF) algorithm to transmit a local pose-graph as a sum of 'delta information' packets. Section 4 then presents the derivation of the origin state method, a novel reinterpretation of the DEIF that is robust to packet loss. Section 5 demonstrates the application of the modified decentralized extended information filter (MDEIF) in a real-world experiment consisting of a topside ship and an autonomous underwater vehicle (AUV). Finally, Section 6 summarizes and presents recommendations for future work.

2. COOPERATIVE LOCALIZATION

The goal of general cooperative localization is simply to estimate the position of vehicles within the network given local information, gained via onboard sensors, and external information, gained via inter-vehicle measurements (such as OWTT range measurements). Several methods for synchronous-clock acoustic navigation have been proposed in the literature. Many of these methods can be conceptualized using a pose-graph, which is a graphical model in which past vehicle poses, or delayed states, are represented by graph nodes and their interdependence is encoded by graph edges (Fig. 1).

[⋆] This work was supported by a grant from the National Science Foundation under awards IIS-0746455 and ANT-1039951.

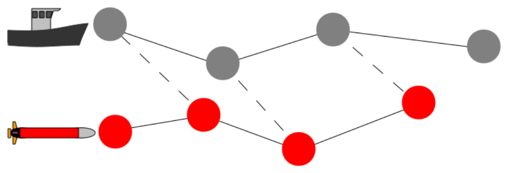


Fig. 1. Global pose-graph for a two-vehicle network. Circle nodes represent vehicle poses, solid edges depict locally observable state transitions, while dashed edges represent relative-range observations.

2.1 Prior Work

Eustice et al. (2006) initially proposed a maximum likelihood estimate (MLE) solution for synchronously navigating subsea nodes via ranging to surface ships. Other algorithms based upon the Kalman family of filters (Webster et al., 2009; Fallon et al., 2010; Bahr et al., 2009), and factor-graphs (Cunningham et al., 2010) also exist. These algorithms are summarized below.

The centralized extended Kalman filter (CEKF) (Webster et al., 2009) is a post-process solution that assumes access to all sensor measurements from all platforms. The CEKF is a global multi-platform pose-graph method that maintains all current vehicle poses and all necessary TOL poses in its state vector (Fig. 1). Each new TOL transmitter state is added as a node on the global pose-graph. Edges are then added representing OWTT range constraints between transmitter TOL nodes and TOA receiver states. The CEKF recursively estimates the pose-graph within a Kalman filter framework, tracking the global state mean and covariance. Sharing inter-platform range measurements builds correlation between vehicle navigation estimates. The utility of the CEKF is that it tracks the full covariance matrix of the network and therefore serves as a baseline for comparison.

A simple approach to distributing the underwater cooperative localization problem is to have each platform only estimate its current state as opposed to the full global pose-graph. OWTT measurement updates are made by including the transmitting platform's local mean and covariance estimate in each acoustic broadcast. The naively distributed extended Kalman filter (NEKF) method is essentially equivalent to the CEKF but with inter-vehicle correlation actively held zero. This method does not track the full pose-graph and correlation that develops as a result of acoustic broadcasts—resulting in an overconfident estimate (Walls and Eustice, 2011). If, however, the correlation remains small between platforms (e.g., when one vehicle has constant access to GPS) the NEKF performs relatively well.

Bahr et al. (2009) proposed the interleaved update (IU) algorithm as a consistent distributed cooperative localization solution. The IU maintains a bank of NEKFs and a table that tracks the origins of each inter-vehicle observation. This entire bank of filters is transmitted within each acoustic broadcast. By careful bookkeeping, relative-range measurement updates are only performed using information that is guaranteed to be uncorrelated. The result is a conservative, but consistent, navigation solution.

The decentralized extended information filter (DEIF) (Webster et al., 2010) is another distributed navigation algorithm. The DEIF exactly matches the CEKF result under a constrained network topology of a single client vehicle receiving acoustic broadcasts from a server vehicle. The authors insightfully observed that information accrued between server broadcasts can be transmitted to the client platform in a small additive packet, termed 'delta information'. The client filter simply adds these delta information packets to reassemble the global pose-graph. This algorithm is described in greater detail in Section 3 and serves as the motivation for our origin state variant.

Cunningham et al. (2010) cooperatively builds landmark-based maps of an environment. The authors present a graphical approach for sharing local map information across a network of vehicles. Their method, termed decentralized data fusion-smoothing and mapping (DDF-SAM), condenses a local landmark-based map and communicates this graph among the network. A global estimation module is then able to consistently combine local graphs and arrive at a global solution. While DDF-SAM does provide a low bandwidth means for distributing local pose-graphs, it does not explicitly decompose the graph to achieve the minimal bandwidth required for the acoustic channel.

2.2 Proposed Method

We consider pose-graph navigation frameworks for their ability to easily encode correlation between separate platforms that develops as a consequence of relative-range observations. Ignoring correlation between platforms can lead to inconsistent results by essentially double-counting information.

In a pose-graph cooperative localization framework, each platform estimates the global graph of the network. Fig. 1 illustrates the problem of estimating the position of two vehicles given local observations and relative-range constraints. Note that each local-graph must minimally contain all nodes that are involved in relative-range observations. We claim that each platform must accomplish two tasks:

(1) Construct and distribute the local pose-graphs of all platforms in the network. In Fig. 1 this step corresponds to building a separate graph for each vehicle using only the solid edges.
(2) Compute a global estimate (i.e., global pose-graph). This action is illustrated as adding the dashed lines in Fig. 1.

The novel contribution of this work is a method for decomposing, distributing, and later reconstructing a local pose-graph using the concept of an origin state. Our algorithm allows relative-pose constraints to be added within a cooperative localization framework with bandwidth and robustness in mind. This method can be directly applied to the DEIF algorithm to make it robust to a lossy communication channel. Moreover, this result also provides possible application to work such as DDF-SAM by providing a more bandwidth constrained method for distributing local pose-graphs.

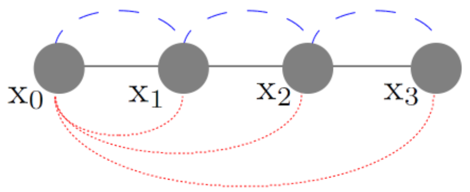


Fig. 2. Local graph decomposition. Coarsely dashed arcs represent state transitions transmitted by the DEIF algorithm. Finely dashed arcs represent state transitions transmitted by the origin state method where $\mathbf{x}_0$ is the agreed upon origin node.

3. DECENTRALIZED EXTENDED INFORMATION FILTER

The DEIF algorithm allows a subsea vehicle to simultaneously reconstruct the topside ship's (or other support vehicle's) local pose-graph and incorporate OWTT derived relative-range constraints. The interested reader is referred to (Webster et al., 2010) for an in depth description of the algorithm, though a brief outline is given below. In the following discussion, the subsea vehicle and support vehicle are referred to as the client and server, respectively. The extended information filter (EIF) tracks the distribution parametrized by the information matrix, Λ, and information vector, $\boldsymbol{\eta}$, which are defined in terms of the covariance, Σ, and mean, $\boldsymbol{\mu}$, of the state vector, $\mathbf{x}$, as

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

The DEIF algorithm accomplishes the previously outlined two steps of pose-graph cooperative localization (local pose-graph distribution and global estimation) within a filtering framework. The DEIF assumes that communication, and consequently relative-range observations, occur in a single direction (i.e, server to client). Therefore, only the server (transmitter) is concerned with communicating its local pose-graph and only the client (receiver) performs a global estimation procedure as it is the sole platform with access to relative-range observations.

The real insight of the DEIF algorithm lies in its ability to distribute the server's local-pose graph as a series of delta information packets. The simple sum of this series exactly reproduces the server pose-graph. Conceptually, each delta information packet describes a one-step transition adding a new node connected to the previous node. These delta information transitions are illustrated in Fig. 2.

3.1 DEIF Operation

The DEIF consists of two information filters: one maintained by the server and only incorporating local measurements, the second running on the client vehicle incorporating its local DR observations as well as relative-range constraints from the server.

Server Side Implementation The server filter augments its state vector at each TOL with its current state, so that after N acoustic broadcasts, its state vector at time k is

$$\mathbf{x}_s(k) = [\mathbf{x}_{s_k}^\top, \mathbf{x}_{s_{t_{N-1}}}^\top, \ldots, \mathbf{x}_{s_{t_0}}^\top]^\top,$$

where the 's' subscript denotes server states and t_i represents the i^{th} TOL. This state vector describes the local pose-graph up to time k with each node representing a TOL state. At the next TOL, the server filter computes a delta information packet to be encoded in the next transmission. The delta information, expressed as,

$$\begin{aligned} \Delta\Lambda_{s_N} &= \Lambda_{s_N} - \Lambda_{s_{N-1}} \\ \Delta\boldsymbol{\eta}_{s_N} &= \boldsymbol{\eta}_{s_N} - \boldsymbol{\eta}_{s_{N-1}} \end{aligned} \tag{1}$$

encompasses all local measurements and predictions that have occurred since the previous TOL, effectively describing a one step transition. The full derivation of this process is described in detail in (Webster et al., 2010). Briefly, the delta information computation is possible because the information matrix exhibits a sparse block tri-diagonal form and measurement updates and predictions only additively modify a small fixed sized portion of the information matrix and vector. Therefore, delta information represents the change in this small area of the information matrix and vector corresponding to the new TOL state and the previous TOL state.

For example, if we consider the server pose-graph at the second TOL, t_1, its state vector would also include the TOL state at time t_0,

$$\mathbf{x}_s(t_1) = [\mathbf{x}_{s_{t_1}}^\top, \mathbf{x}_{s_{t_0}}^\top]^\top.$$

The corresponding information matrix and vector are

$$\Lambda_{s_1} = \begin{bmatrix} \Lambda_{s_{t_1}s_{t_1}} & \Lambda_{s_{t_1}s_{t_0}} \\ \Lambda_{s_{t_0}s_{t_1}} & \Lambda_{s_{t_0}s_{t_0}} \end{bmatrix}, \ \boldsymbol{\eta}_{s_2} = \begin{bmatrix} \boldsymbol{\eta}_{s_{t_1}} \\ \boldsymbol{\eta}_{s_{t_0}} \end{bmatrix}.$$

At the next TOL, t_2, the server state vector becomes

$$\mathbf{x}_s(t_2) = [\mathbf{x}_{s_{t_2}}^\top, \mathbf{x}_{s_{t_1}}^\top, \mathbf{x}_{s_{t_0}}^\top]^\top.$$

The information matrix and vector now take the form

$$\Lambda_{s_2} = \begin{bmatrix} \Lambda_{s_{t_2}s_{t_2}} & \Lambda_{s_{t_2}s_{t_1}} & 0 \\ \Lambda_{s_{t_1}s_{t_2}} & \Lambda'_{s_{t_1}s_{t_1}} & \Lambda_{s_{t_1}s_{t_0}} \\ 0 & \Lambda_{s_{t_0}s_{t_1}} & \Lambda_{s_{t_0}s_{t_0}} \end{bmatrix}, \ \boldsymbol{\eta}_{s_2} = \begin{bmatrix} \boldsymbol{\eta}_{s_{t_2}} \\ \boldsymbol{\eta}'_{s_{t_1}} \\ \boldsymbol{\eta}_{s_{t_0}} \end{bmatrix},$$

where the sub elements $\Lambda'_{s_{t_1}s_{t_1}} \neq \Lambda_{s_{t_1}s_{t_1}}$ and $\boldsymbol{\eta}'_{s_{t_1}} \neq \boldsymbol{\eta}_{s_{t_1}}$. The delta information is computed as

$$\Delta\Lambda_{s_2} = \Lambda_{s_2} - \Lambda_{s_1} = \begin{bmatrix} \Lambda_{s_{t_2}s_{t_2}} & \Lambda_{s_{t_2}s_{t_1}} & 0 \\ \Lambda_{s_{t_1}s_{t_2}} & (\Lambda'_{s_{t_1}s_{t_1}} - \Lambda_{s_{t_1}s_{t_1}}) & 0 \\ 0 & 0 & 0 \end{bmatrix},$$

$$\Delta\boldsymbol{\eta}_{s_2} = \boldsymbol{\eta}_{s_2} - \boldsymbol{\eta}_{s_1} = \begin{bmatrix} \boldsymbol{\eta}_{s_{t_2}} \\ \boldsymbol{\eta}'_{s_{t_1}} - \boldsymbol{\eta}_{s_{t_1}} \\ 0 \end{bmatrix},$$

where Λ_{s_1} and $\boldsymbol{\eta}_{s_1}$ are padded with zeros for conformability. The extraneous zero rows and columns are dropped from the delta information matrix and vector for transmission. Each delta information packet displays this same nonzero pattern leading to a fixed bandwidth packet. This compact packet encodes the changes in the server pose-graph from t_1 to t_2, which is broadcast to the client vehicle at the TOL.

Note that the server is only required to keep the last two TOL states in its state vector for delta information computation. Therefore, in a practical implementation older states can be marginalized out without consequence. Advantageously, the memory requirement of the information filter grows linearly with the number of past states, $\mathcal{O}(N)$, as opposed to the covariance form of the Kalman filter which grows quadratically, $\mathcal{O}(N^2)$.

Client Side Implementation The client filter tracks the global pose-graph consisting of nodes representing its current state as well as previous server TOL states. Server TOL states are accumulated through the sequence of delta information packets received at the TOA via the reverse operation described in (1). This additive step reconstructs the server local pose-graph up to its TOL.

After receiving N acoustic broadcasts at time k, the client state vector includes

$$\mathbf{x}_c(k) = [\mathbf{x}_{c_k}^\top, \mathbf{x}_{s_{t_{N-1}}}^\top, \ldots, \mathbf{x}_{s_{t_0}}^\top]^\top,$$

where the 'c' subscript denotes client states. After incorporating the newest delta information packet, the filter proceeds with the new range measurement update following the standard Kalman procedure, completing the global estimation step. Note that following this relative-range measurement update, the client DEIF estimate is exactly equivalent to the estimate maintained by the CEKF.

The DEIF's sequential nature of the delta information computation imposes a non-lossy communication requirement. By missing a single delta information packet, the receiving platform can no longer reconstruct the transmitter's local pose-graph. A practical workaround is to use a transmission redundancy scheme; however, this vitiates the algorithm's low-bandwidth motivation.

4. ORIGIN STATE METHOD

The origin state method is motivated by the non-lossy communication requirement of the DEIF. While DEIF delta information packets represent a one-step transition between consecutive nodes, there is no physical interpretation behind the meaning of each packet. Instead, we propose an alternative called the origin state method, which decomposes the pose-graph into meaningful components that allow for the pose-graph to be reconstructed even in the presence of missed packets.

The basic idea behind the origin state method is to represent nodes in the pose-graph relative to another node, the origin, as opposed to a transition from the previous node as in the DEIF. This difference is illustrated in Fig. 2. Instead of representing the relationship between nodes as an additive delta information packet, we explicitly solve for the joint marginal distribution of the new node and the origin node. The process of computing an origin state packet and reconstructing the pose-graph given a new origin state packet is demonstrated below.

4.1 Computing Origin State Packets

Assume that the transmitter's state vector, after three TOL at times t_0, t_1, and t_2, is

$$\mathbf{x}_2 = [\mathbf{x}_{t_2}^\top, \mathbf{x}_{t_1}^\top, \mathbf{x}_{t_0}^\top]^\top.$$

The corresponding information matrix and vector are:

$$\Lambda_2 = \begin{bmatrix} \Lambda_{t_2,t_2} & \Lambda_{t_2,t_1} & 0 \\ \Lambda_{t_1,t_2} & \Lambda_{t_1,t_1} & \Lambda_{t_1,t_1} \\ 0 & \Lambda_{t_0,t_1} & \Lambda_{t_0,t_0} \end{bmatrix}, \ \boldsymbol{\eta}_2 = \begin{bmatrix} \boldsymbol{\eta}_{t_2} \\ \boldsymbol{\eta}_{t_1} \\ \boldsymbol{\eta}_{t_0} \end{bmatrix}, \tag{2}$$

where the off-diagonal zero is from the Markov property. At the next TOL state, t_3, the state vector is augmented with the current platform pose. The information matrix and vector now take the form

$$\Lambda_3 = \left[\begin{array}{|cc|cc} \hline \Lambda_{t_3,t_3} & \Lambda_{t_3,t_2} & 0 & 0 \\ \Lambda_{t_2,t_3} & \Lambda'_{t_2,t_2} & \Lambda_{t_2,t_1} & 0 \\ \hline 0 & \Lambda_{t_1,t_2} & \Lambda_{t_1,t_1} & \Lambda_{t_1,t_0} \\ 0 & 0 & \Lambda_{t_0,t_1} & \Lambda_{t_0,t_0} \end{array}\right], \ \boldsymbol{\eta}_3 = \left[\begin{array}{|c|} \hline \boldsymbol{\eta}_{t_3} \\ \boldsymbol{\eta}'_{t_2} \\ \hline \boldsymbol{\eta}_{t_1} \\ \boldsymbol{\eta}_{t_0} \end{array}\right], \tag{3}$$

with boxed elements indicating new (from augmentation) or changed values. Note that $\Lambda'_{t_2,t_2} \neq \Lambda_{t_2,t_2}$ and likewise $\boldsymbol{\eta}'_{t_2} \neq \boldsymbol{\eta}_{t_2}$. The transition from (2) to (3) contains all local information that has been gained by the transmitter from local measurements.

The origin state packet is computed as the joint marginal of the transmitter distribution over the new TOL node, $\mathbf{x}_{t_3}$, and the origin state, $\mathbf{x}_{t_0}$. The origin state is simply a node in the pose-graph that the transmitter and the receiver both agree upon. In this example, let $\mathbf{x}_{t_0}$ be the origin state. In the information form, marginalization is defined by the Schur complement resulting in

$$\Lambda_3^s = \begin{bmatrix} \Lambda_{t_3,t_3}^s & \Lambda_{t_3,t_0}^s \\ \Lambda_{t_0,t_3}^s & \Lambda_{t_0,t_0}^s \end{bmatrix}, \ \boldsymbol{\eta}_3^s = \begin{bmatrix} \boldsymbol{\eta}_{t_3}^s \\ \boldsymbol{\eta}_{t_0}^s \end{bmatrix}, \tag{4}$$

where the 's' superscript indicates that this is the joint marginal distribution computed on the server. An origin state packet is computed at each TOL and communicated to the receiver where the series of all origin state packets is used to reconstruct the transmitter's pose-graph. The dimension of the origin state packet is equal to that of the delta information required by the DEIF.

4.2 Reconstructing Pose-Graphs from Origin State Packets

The goal of the receiving platform is to reconstruct the transmitter local pose-graph given the sequence of origin state packets. In this example, assume that the receiver has acquired the pose-graph up to the previous TOL, (i.e., up to t_2 in (2)), and then receives the new origin state packet, expressed as (4). The received origin state packet encodes the joint marginal of the transmitter state at the TOL and the origin. For notational convenience, we also define the joint marginal of the previous delayed state, $\mathbf{x}_{t_2}$, and the origin state, $\mathbf{x}_{t_0}$, over the previous distribution, (2),

$$\Lambda_2^c = \begin{bmatrix} \Lambda_{t_2,t_2}^c & \Lambda_{t_2,t_0}^c \\ \Lambda_{t_0,t_2}^c & \Lambda_{t_0,t_0}^c \end{bmatrix}, \ \boldsymbol{\eta}_2^c = \begin{bmatrix} \boldsymbol{\eta}_{t_2}^c \\ \boldsymbol{\eta}_{t_0}^c \end{bmatrix}, \tag{5}$$

where this time the 'c' superscript indicates that this joint marginal is computed on the client platform.

The client can now reconstruct (3) using just (2), (4), and (5). By studying how the new origin state packet, (4), was computed via the marginalization operation of (3) (refer to Appendix A), we equate terms and explicitly solve for the unknown values of the information matrix and vector. The full pose-graph up to time t_3 is then defined in the information form by computing the following system,

$$\begin{aligned} \Omega^{-1} &= \Lambda_{t_0,t_2}^{c\ -1}(\Lambda_{t_0,t_0}^c - \Lambda_{t_0,t_0}^s)\Lambda_{t_2,t_0}^{c\ -1} \\ \beta &= -\Lambda_{t_3,t_0}^s(\Omega^{-1}\Lambda_{t_2,t_0})^{-1} \\ \boldsymbol{\nu} &= \left[\Lambda_{t_0,t_2}^c\Omega^{-1}\right]^{-1}(\boldsymbol{\eta}_{t_0}^c - \boldsymbol{\eta}_{t_0}^s) \\ \Lambda_{t_3,t_3} &= \Lambda_{t_3,t_3}^s + \beta\Omega^{-1}\beta^\top \\ \Lambda_{t_3,t_2} &= \beta \\ \Lambda'_{t_2,t_2} &= \Omega + \Lambda_{t_2,t_1}\Lambda_{t_1,t_1}^{-1}\Lambda_{t_1,t_2}^{-1} \\ \boldsymbol{\eta}_{t_3} &= \boldsymbol{\eta}_{t_3}^s + \beta\Omega^{-1}\boldsymbol{\nu} \\ \boldsymbol{\eta}'_{t_2} &= \boldsymbol{\nu} + \Lambda_{t_2,t_1}\Lambda_{t_1,t_1}^{-1}\boldsymbol{\eta}_{t_1}. \end{aligned} \tag{6}$$

The equations in (6) solve for the full joint distribution given the joint marginals over each TOL state and the origin state. Intuitively, this method can be thought of as inferring the correlation between a new TOL state and all other past states by observing how its correlation with a known origin state is affected. We can see this in the terms $(\Lambda^c_{t_0,t_0} - \Lambda^s_{t_0,t_0})$ and $(\boldsymbol{\eta}^c_{t_0} - \boldsymbol{\eta}^s_{t_0})$, which express the difference in information known about the origin state by the client and server vehicles.

If a packet is lost in transmission, the receiving platform still reconstructs an equivalent pose-graph with subsequent packets as if the lost TOL state had been marginalized out. The robustness of the origin state method to dropped transmissions is an important feature especially when using an unstable communication channel, such as the acoustic channel.

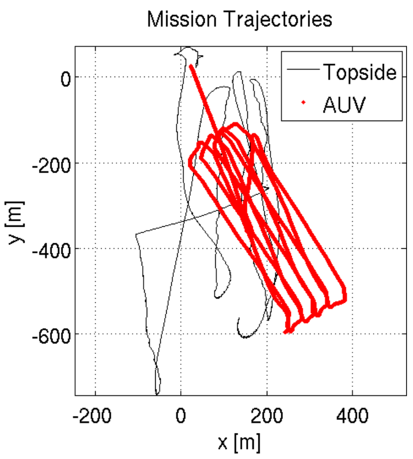
4.3 Numerical Stability

The origin state method defines a transmission scheme robust to packet loss using a small fixed amount of bandwidth; however, there is a catch. Over time the correlation between each new TOL state and the origin state decreases due to process noise. As mentioned previously, the origin state method computes the full joint distribution over the current state and all past states by observing how information gained between transmissions affects the estimate of an origin state. As the correlation between the newest state and the origin state decreases, the ability to numerically compute the difference in information known about the origin state diminishes. Eventually, the origin state method breaks as a result of numerical instability.

A simple solution to this issue is to ensure that the origin state is continually moved forward in time, so as to remain as close as possible to the new TOL state. Origin shifting can easily be accomplished in the underwater scenario by allowing communication to occur bi-directionally among platforms so that each platform can transmit an index relating to the most recently received TOL pose from all other platforms in addition to its usual origin state packet. Each platform can then track the most recent state received by all other vehicles and shift its origin state forward accordingly. The drawback is that client communication is no longer passive.

4.4 Modified Decentralized Extended Information Filter

The MDEIF operation mirrors the standard DEIF except for the acoustic packet composition. In the MDEIF, the server vehicle transmits the origin state joint marginal of each TOL state. The client vehicle now reconstructs the local information from the server as previously outlined, and computes the current delta information *exactly* as if the server vehicle had done so. This delta information is then handed over to the existing DEIF machinery. The resulting MDEIF algorithm produces the same navigation solution as the DEIF while lifting the non-lossy communications constraint. For practical implementation, an origin shifting scheme similar to the one outlined above is necessary.



(a) Vehicle trajectories

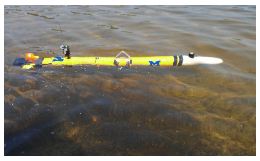

(b) Iver2 AUV

Fig. 3. Experimental Setup.

Table 1. AUV Navigation Sensors: Sampling Frequency and Noise Characteristics.

Client Sensors	**Variable**	**Frequency**	**Noise**
Microstrain AHRS	ϕ, θ, ψ	25.0 Hz	2.0°
Doppler velocity log	$\dot{x}, \dot{y}, \dot{z}$	3.0 Hz	5.0 cm/s
pressure depth	z	2.0 Hz	10 cm
acoustic modem	slant range	every ∼15 sec	1 m
Server Sensors	**Variable**	**Frequency**	**Noise**
GPS	x, y	1.0 Hz	3.0 m

5. EXPERIMENTS

5.1 Experimental Setup

A two-node AUV trial was carried out using one custom modified Ocean-Server Iver2 AUV and a topside surface craft. The AUV followed a lawn-mower pattern with roughly 500 m tracklines spaced 50 m apart as depicted in Fig. 3. The topside ship drifted to various positions around the survey area during the mission.

The AUV was equipped with a typical advanced dead-reckoned sensor suite (as detailed in (Brown et al., 2008) and summarized in Table 1) comprised of a 600 kHz RDI Doppler velocity log (DVL), a Microstrain 3DM-GX1 attitude and heading reference system (AHRS), and a Desert Star Systems SSP-1 digital pressure sensor. We define the state of each vehicle at time k as

$$\mathbf{x}_k = [x_k, y_k, \dot{x}_k, \dot{y}_k]^\top ,$$

where the vehicle position in the local-level plane is described by the xy pair and the corresponding world-frame velocities are $\dot{x}\dot{y}$. Since we consider attitude to be instrumented with bounded error, we project the DVL body-frame velocity measurements into the world-frame and treat these as linear observations of the AUV velocity. The topside vehicle only observes world-frame position as measured by a GPS receiver.

The source of each acoustic transmission was defined by a fixed time division multiple access (TDMA) schedule during which each vehicle was assigned a time-slot to send a data packet. The network maintained a 145 second TDMA cycle, which consisted of 6 topside broadcasts and 4 subsea broadcasts. During the experiment, the subsea

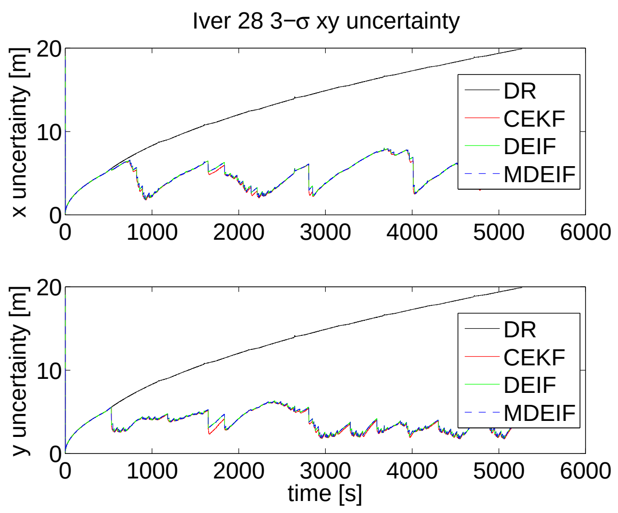


Fig. 4. Estimated AUV position uncertainty. Note that the estimates produced by the CEKF, DEIF, and MDEIF are nearly identical.

vehicle received roughly 81% of acoustic transmissions from the ship, while the topside ship received about 67% of the subsea transmissions.

We compared the performance of the DEIF and the MDEIF through post-process implementation. In order to run the DEIF, we made the assumption that successful acoustic broadcasts were known ahead of time in order to correctly compute each delta information; an assumption that would not hold true in a real-time implementation. The MDEIF uses acoustic broadcasts sent from the AUV to the topside ship to shift the origin state forward in time as proposed earlier. No assumptions regarding successful transmission were made by the MDEIF so that its performance in post-process accurately reflects its real-time performance.

5.2 *Results*

The xy position uncertainty estimates are shown in Fig. 4 for the DEIF and MDEIF, as well as for the CEKF for comparison. The position uncertainty is greatly reduced by incorporating OWTT relative-range measurements between the two vehicles.

Fig. 5 presents the difference in xy position estimates for the DEIF and MDEIF. The difference remains under several millimeters throughout the experiment and, with the exception of a brief period toward the end of the mission (corresponding to a period of lost communication), the difference is on the order of numerical precision. During the period of lost communication, the origin is not shifted forward in time so we expect the origin state method to have difficulty with numerical precision. The difference remains small, however, and the algorithm is able to recover.

We see that the MDEIF performs nearly identical to the DEIF without requiring a non-lossy communication channel or redundant transmission schemes.

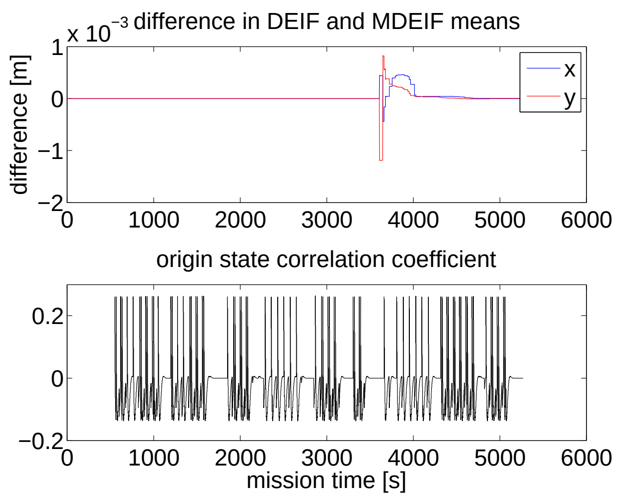


Fig. 5. The top plot shows the difference between the DEIF and MDEIF position estimates, which remain negligible throughout the experiment. The lower plot shows the correlation coefficient between the origin state and the current topside state. The correlation coefficient quickly goes to zero unless the origin is shifted forward in time. The difference between the DEIF and MDEIF is largest during a period when the origin is not shifted forward (t=3400 to t=3700), although this difference quickly decreases once regular communication resumes (t=4000).

6. CONCLUSION

In this paper, we have shown that cooperative localization can be segmented into two component tasks: local graph synthesis and global graph estimation. The proposed origin state method is a low bandwidth solution to the first component. Under the origin state framework, receiving platforms can reconstruct the transmitter's local pose-graph as a combination of many joint marginal origin state packets.

We also presented an extension to the DEIF estimation framework, the MDEIF, which lifts the non-lossy communication requirement of the DEIF. We showed the performance of the MDEIF on an experimental data set including a single AUV and a topside ship, and found it to be equal to the DEIF and CEKF in terms of accuracy.

Future work will address the numerical instability that occurs when the current state of the transmitting vehicle is no longer correlated with the origin state. We will also examine other estimation frameworks that, in concert with the origin state method, could incorporate relative-range observations across a large network of vehicles.

ACKNOWLEDGEMENTS

We are grateful to the University of Michigan Biological Station for their logistical support during field experiments.

REFERENCES

Bahr, A., Walter, M.R., and Leonard, J.J. (2009). Consistent cooperative localization. In *Proc. IEEE Int. Conf. Robot. and Automation*, 4295–4302.

Brown, H., Kim, A., and Eustice, R. (2008). Development of a multi-AUV SLAM testbed at the University of Michigan. In *Proc. IEEE/MTS OCEANS Conf. Exhib.*, 1–6. Quebec, Canada.

Cunningham, A., Paluri, B., and Dellaert, F. (2010). DDF-SAM: Fully distributed SLAM using constrained factor graphs. In *Proc. IEEE/RSJ Int. Conf. Intell. Robots and Syst.*, 3025–3030.

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

Eustice, R.M., Singh, H., and Whitcomb, L.L. (2011). Synchronous-clock one-way-travel-time acoustic navigation for underwater vehicles. *J. Field Robot.*, 28(1), 121–136.

Fallon, M.F., Papadopoulos, G., Leonard, J.J., and Patrikalakis, N.M. (2010). Cooperative AUV navigation using a single maneuvering surface craft. *Int. J. Robot. Res.*, 29(12), 1461–1474.

Hunt, M., Marquet, W., Moller, D., Peal, K., Smith, W., and Spindel, R. (1974). An acoustic navigation system. Technical Report WHOI-74-6, Woods Hole Ocean. Inst.

Vaganay, J., Leonard, J., Curcio, J., and Willcox, J. (2004). Experimental validation of the moving long base-line navigation concept. In *Proc. IEEE/OES Autonomous Underwater Vehicles Conf.*, 59–65.

Walls, J.M. and Eustice, R.M. (2011). Experimental comparison of synchronous-clock cooperative acoustic navigation algorithms. In *Proc. IEEE/MTS OCEANS Conf. Exhib.*, 1–7. Kona, HI.

Webster, S.E., Eustice, R.M., Singh, H., and Whitcomb, L.L. (2009). Preliminary deep water results in single-beacon one-way-travel-time acoustic navigation for underwater vehicles. In *Proc. IEEE/RSJ Intl. Conf. Intell. Robots and Syst.*, 2053–2060. St. Louis, MO.

Webster, S.E., Whitcomb, L.L., and Eustice, R.M. (2010). Preliminary results in decentralized estimation for single-beacon acoustic underwater navigation. In *Proc. Robot.: Sci. & Syst. Conf.* Zaragoza, Spain.

Appendix A. ORIGIN STATE METHOD DERIVATION

This section details the steps taken to obtain (6). This procedure allows the client to solve for the joint distribution over all server TOL states,

$$\Lambda_3 = \begin{bmatrix} \boxed{\begin{matrix}\Lambda_{t_3,t_3} & \Lambda_{t_3,t_2} \\ \Lambda_{t_2,t_3} & \Lambda'_{t_2,t_2}\end{matrix}} & \begin{matrix} 0 & 0 \\ \Lambda_{t_2,t_1} & 0 \end{matrix} \\ \begin{matrix} 0 & \Lambda_{t_1,t_2} \\ 0 & 0 \end{matrix} & \begin{matrix} \Lambda_{t_1,t_1} & \Lambda_{t_1,t_0} \\ \Lambda_{t_0,t_1} & \Lambda_{t_0,t_0} \end{matrix} \end{bmatrix}, \quad \boldsymbol{\eta}_3 = \begin{bmatrix} \boxed{\begin{matrix}\boldsymbol{\eta}_{t_3} \\ \boldsymbol{\eta}'_{t_2}\end{matrix}} \\ \boldsymbol{\eta}_{t_1} \\ \boldsymbol{\eta}_{t_0} \end{bmatrix}, \tag{A.1}$$

given information that the client has access to including the server joint distribution after the previous origin state packet,

$$\Lambda_2 = \begin{bmatrix} \Lambda_{t_2,t_2} & \Lambda_{t_2,t_1} & 0 \\ \Lambda_{t_1,t_2} & \Lambda_{t_1,t_1} & \Lambda_{t_1,t_1} \\ 0 & \Lambda_{t_0,t_1} & \Lambda_{t_0,t_0} \end{bmatrix}, \quad \boldsymbol{\eta}_2 = \begin{bmatrix} \boldsymbol{\eta}_{t_2} \\ \boldsymbol{\eta}_{t_1} \\ \boldsymbol{\eta}_{t_0} \end{bmatrix}, \tag{A.2}$$

and the new origin state packet,

$$\Lambda_3^s = \begin{bmatrix} \Lambda^s_{t_3,t_3} & \Lambda^s_{t_3,t_0} \\ \Lambda^s_{t_0,t_3} & \Lambda^s_{t_0,t_0} \end{bmatrix}, \quad \boldsymbol{\eta}_3^s = \begin{bmatrix} \boldsymbol{\eta}^s_{t_3} \\ \boldsymbol{\eta}^s_{t_0} \end{bmatrix}, \tag{A.3}$$

which is simply the joint marginal of (A.1) over the newest TOL state at t_3, and the origin state at t_0 computed by the server. Note that the client will solve for the boxed elements in (A.1).

To make the derivation clear, we define two more marginal distributions each computed by the client. First, we define the joint marginal of (A.1) over the transmitter states at times t_3 and t_2 as well as the origin state at time t_0,

$$\Lambda_3^c = \begin{bmatrix} \Lambda_{t_3,t_3} & \Lambda_{t_3,t_2} & 0 \\ \Lambda_{t_2,t_3} & \Lambda^{c\ \prime}_{t_2,t_2} & \Lambda^c_{t_2,t_0} \\ 0 & \Lambda^c_{t_0,t_2} & \Lambda^c_{t_0,t_0} \end{bmatrix}, \quad \boldsymbol{\eta}_3^n = \begin{bmatrix} \boldsymbol{\eta}_{t_3} \\ \boldsymbol{\eta}^{c\ \prime}_{t_2} \\ \boldsymbol{\eta}^c_{t_0} \end{bmatrix}. \tag{A.4}$$

The receiver does not have outright access to this distribution, but solving for it will serve as an intermediate step. Second, we express the joint marginal of the distribution up to time t_2, (A.2), over the previous delayed state at time t_2, and the origin state at time t_0,

$$\Lambda_2^c = \begin{bmatrix} \Lambda^c_{t_2,t_2} & \Lambda^c_{t_2,t_0} \\ \Lambda^c_{t_0,t_2} & \Lambda^c_{t_0,t_0} \end{bmatrix}, \quad \boldsymbol{\eta}_2^c = \begin{bmatrix} \boldsymbol{\eta}^c_{t_2} \\ \boldsymbol{\eta}^c_{t_0} \end{bmatrix}. \tag{A.5}$$

The strategy to solve for (A.1) first involves explicitly writing (A.3) by applying the Schur complement to (A.4) and then equating known terms. We can write the new origin state packet in terms of a marginal of (A.4) as

$$\begin{bmatrix} \Lambda^s_{t_3,t_3} & \Lambda^s_{t_3,t_0} \\ \Lambda^s_{t_0,t_3} & \Lambda^s_{t_0,t_0} \end{bmatrix} = \begin{bmatrix} \Lambda_{t_3,t_3} - \Lambda_{t_3,t_2}\Lambda^{c\ \prime -1}_{t_2,t_2}\Lambda_{t_2,t_3} & -\Lambda_{t_3,t_2}\Lambda^{c\ \prime -1}_{t_2,t_2}\Lambda^c_{t_2,t_0} \\ -\Lambda^c_{t_0,t_2}\Lambda^{c\ \prime -1}_{t_2,t_2}\Lambda_{t_2,t_3} & \Lambda^c_{t_0,t_0} - \Lambda^c_{t_0,t_2}\Lambda^{c\ \prime -1}_{t_2,t_2}\Lambda^c_{t_2,t_0} \end{bmatrix},$$

$$\begin{bmatrix} \boldsymbol{\eta}^s_{t_3} \\ \boldsymbol{\eta}^s_{t_0} \end{bmatrix} = \begin{bmatrix} \boldsymbol{\eta}_{t_3} - \Lambda_{t_3,t_2}\Lambda^{c\ \prime -1}_{t_2,t_2}\boldsymbol{\eta}^{c\ \prime}_{t_2} \\ \boldsymbol{\eta}^c_{t_0} - \Lambda^c_{t_0,t_2}\Lambda^{c\ \prime -1}_{t_2,t_2}\boldsymbol{\eta}^{c\ \prime}_{t_2} \end{bmatrix}. \tag{A.6}$$

Solving the above leads to

$$\begin{aligned} \Omega^{-1} &= \Lambda^{c\ \prime -1}_{t_2,t_2} = \Lambda^{c\ -1}_{t_0,t_2}(\Lambda^c_{t_0,t_0} - \Lambda^s_{t_0,t_0})\Lambda^{c\ -1}_{t_2,t_0} \\ \beta &= -\Lambda^s_{t_3,t_0}(\Omega^{-1}\Lambda_{t_2,t_0})^{-1} \\ \boldsymbol{\nu} &= \boldsymbol{\eta}^{c\ \prime}_{t_2} = \left[\Lambda^c_{t_0,t_2}\Omega^{-1}\right]^{-1}(\boldsymbol{\eta}^c_{t_0} - \boldsymbol{\eta}^s_{t_0}) \\ \Lambda_{t_3,t_3} &= \Lambda^s_{t_3,t_3} + \beta\Omega^{-1}\beta^\top \\ \Lambda_{t_3,t_2} &= \beta \\ \Lambda_{t_2,t_3} &= \beta^\top \\ \boldsymbol{\eta}_{t_3} &= \boldsymbol{\eta}^s_{t_3} + \beta\Omega^{-1}\boldsymbol{\nu}. \end{aligned} \tag{A.7}$$

The only remaining unknown elements from (A.1) are Λ'_{t_2,t_2} and $\boldsymbol{\eta}'_{t_2}$. We take a similar approach to solve this by writing (A.4) after applying the Schur complement to (A.1) and then equating terms.

$$\begin{bmatrix} \Lambda_{t_3,t_3} & \Lambda_{t_3,t_2} & 0 \\ \Lambda_{t_2,t_3} & \Lambda^{c\ \prime}_{t_2,t_2} & \Lambda^c_{t_2,t_0} \\ 0 & \Lambda^c_{t_0,t_2} & \Lambda^c_{t_0,t_0} \end{bmatrix} = \begin{bmatrix} \Lambda_{t_3,t_3} & \Lambda_{t_3,t_2} & 0 \\ \Lambda_{t_2,t_3} & \Lambda'_{t_2,t_2} - \Lambda_{t_2,t_1}\Lambda^{-1}_{t_1,t_1}\Lambda_{t_1,t_2} & -\Lambda_{t_2,t_1}\Lambda^{-1}_{t_1,t_1}\Lambda_{t_1,t_0} \\ 0 & -\Lambda_{t_0,t_1}\Lambda^{-1}_{t_1,t_1}\Lambda_{t_1,t_2} & \Lambda_{t_0,t_0} - \Lambda_{t_0,t_1}\Lambda^{-1}_{t_1,t_1}\Lambda_{t_1,t_0} \end{bmatrix},$$

$$\begin{bmatrix} \boldsymbol{\eta}_{t_3} \\ \boldsymbol{\eta}^{c\ \prime}_{t_2} \\ \boldsymbol{\eta}^c_{t_0} \end{bmatrix} = \begin{bmatrix} \boldsymbol{\eta}_{t_3} \\ \boldsymbol{\eta}_{t_2} - \Lambda_{t_2,t_1}\Lambda^{-1}_{t_1,t_1}\boldsymbol{\eta}_{t_1} \\ \boldsymbol{\eta}_{t_0} - \Lambda_{t_0,t_1}\Lambda^{-1}_{t_1,t_1}\boldsymbol{\eta}_{t_1} \end{bmatrix}. \tag{A.8}$$

The final unknown terms can then be expressed as

$$\begin{aligned} \Lambda'_{t_2,t_2} &= \Lambda^{c\ \prime}_{t_2,t_2} + \Lambda_{t_2,t_1}\Lambda^{-1}_{t_1,t_1}\Lambda_{t_1,t_2} \\ \boldsymbol{\eta}'_{t_2} &= \boldsymbol{\eta}^{c\ \prime}_{t_2} + \Lambda_{t_2,t_1}\Lambda^{-1}_{t_1,t_1}\boldsymbol{\eta}_{t_1}. \end{aligned} \tag{A.9}$$

Together (A.7) and (A.9) present a method to solve for the unknown elements of (A.1).