Ad Hoc Networks 34 (2015) 184–195

Contents lists available at ScienceDirect

Ad Hoc Networks

journal homepage: www.elsevier.com/locate/adhoc

Modeling position uncertainty of networked autonomous underwater vehicles

Baozhi Chen, Dario Pompili *

Department of Electrical and Computer Engineering, Rutgers University, New Brunswick, NJ, United States

ARTICLE INFO

Article history:
Received 1 March 2014
Received in revised form 13 August 2014
Accepted 2 September 2014
Available online 10 September 2014

Keywords:
Underwater sensor networks
Position uncertainty
Autonomous underwater vehicles
Unscented Kalman Filter

ABSTRACT

Recently there has been increasing engineering activity in the deployment of Autonomous Underwater Vehicles (AUVs). Different types of AUVs are being used for applications ranging from ocean exploration to coastal tactical surveillance. These AUVs generally follow a predictable trajectory specified by the mission requirements. Inaccuracies in models for deriving position estimates and the drift caused by ocean currents, however, lead to uncertainty when estimating an AUV's position. In this article, two forms of position uncertainty – *internal* and *external* – are studied, which are the position uncertainty associated with a particular AUV *as seen by itself* and that *as seen by others*, respectively. Then, a statistical model to estimate the internal uncertainty for a general AUV is proposed. Based on this model, a novel mathematical framework using Unscented Kalman Filtering is developed to estimate the external uncertainty. Finally, the benefits of this framework for several location-sensitive applications are shown via emulations.

1. Introduction

Recently UnderWater Acoustic Sensor Networks (UW-ASNs) [1] have been deployed to carry out collaborative monitoring missions such as oceanographic data collection, disaster prevention, and navigation. To ensure coverage of the vast undersampled 3D aquatic environment, Autonomous Underwater Vehicles (AUVs) endowed with sensing and wireless communication capabilities become essential. These AUVs – which can be divided into two classes, *propeller-less/buoyancy-driven* (e.g., gliders) and *Propeller-Driven Vehicles* (PDVs) – rely on local intelligence with minimal onshore operator dependence. Due to propagation limitations of radio frequency and optical waves, i.e., high medium absorption and scattering, respectively, acoustic communication technology is employed to transfer vital information (data and control) multi-hopping between AUVs underwater and, ultimately, to a surface or onshore station where this information is collected and analyzed.

In terrestrial sensor networks, the position of a node can be characterized by a single point because localization error can be made small by using the Global Positioning System (GPS), which, however, does not work underwater. In contrast, underwater inaccuracies in localization models and the drift caused by ocean currents will significantly increase the position uncertainty of AUVs. Hence, using a deterministic point is not sufficient to pinpoint the position of underwater vehicles. Furthermore, in the water, such a deterministic approach may cause problems such as errors in inter-vehicle communications, vehicle collisions, loss of synchronization – all possibly leading to mission failures [2].

To address the problems caused by position uncertainty, we introduce a probability region to characterize stochastically a node's position. Depending on the view

* Corresponding author.
E-mail addresses: baozhi_chen@cac.rutgers.edu (B. Chen), pompili@ece.rutgers.edu (D. Pompili).

http://dx.doi.org/10.1016/j.adhoc.2014.09.003
1570-8705/

of the different nodes, we define two forms of position uncertainty, i.e., *internal uncertainty*, the position uncertainty associated with a particular entity/node (such as an AUV) *as seen by itself*; and *external uncertainty*, the uncertainty associated with a particular entity/node *as seen by others*. These two notions introduce a shift in AUV localization – *from a deterministic to a probabilistic view* – which can be leveraged to improve the performance of solutions for a variety of problems. For example, in *UW-ASNs*, by using the external-uncertainty notion, error failures in geographical routing protocols can be decreased, and the output power at the transmitting node can be optimized with the constraint to guarantee a certain Signal-to-Noise Ratio (SNR) at the receiver (taking into account not only channel impairments but also position uncertainty). This notion can also be used in *underwater robotics* to minimize the risk of vehicle collisions, in *underwater localization* to increase the position accuracy by selecting a subset of nodes characterized by a low external uncertainty to be used as "anchors" (i.e., reference nodes employed in "trilateration"), and in *task allocation*, i.e., the problem of selecting a subset of AUVs to accomplish a mission, to geocast the mission details to the AUVs within a certain region. Finally, this notion plays a major role in *data processing/visualization* to improve the quality of 3D data reconstruction as the AUV deviation from the original mission path can be estimated and factored in.

To enable these applications, each node needs to estimate the external uncertainty of other nodes. To do this, the nodes need to first estimate their internal uncertainty and then broadcast it to the neighbors. Due to the large *network latency* (including communication transmission and propagation delay) and *information loss*, this received uncertainty information is a delayed version of a node's internal uncertainty and is used as the base for the neighbors to estimate the sender's uncertainty (i.e., external uncertainty). As a result, these two forms of uncertainty are generally different. To estimate the external uncertainty, we first propose a statistical approach to model the internal uncertainty of AUVs following predictable trajectories. Based on this internal uncertainty, we then propose a solution using the Unscented Kalman Filter (UKF) algorithm to predict the external uncertainty associated with any localization technique and leverage this information for performance improvement. Note that in [3] we introduced and used the notion of external uncertainty, whose region for simplicity was considered to be equal to that characterizing the internal uncertainty (its lower bound). Here we remove that simplifying assumption and rigorously model the external uncertainty by incorporating network latency and packet loss, which brings great benefits to a variety of problems.

The remainder of this article is organized as follows: in Section 2, we present the notions of internal and external uncertainty and discuss the benefits of using these two notions. We then propose our solution for modeling these uncertainties in Sections 3 and 4, followed by a thorough performance evaluation in Section 5; conclusions are finally drawn in Section 6.

2. The external uncertainty and its benefits in UW-ASNs

We define here the two types of position uncertainty, discuss the relationship between them, and comment on the benefits of using the external uncertainty in a variety of research areas and problems.

2.1. Internal uncertainty

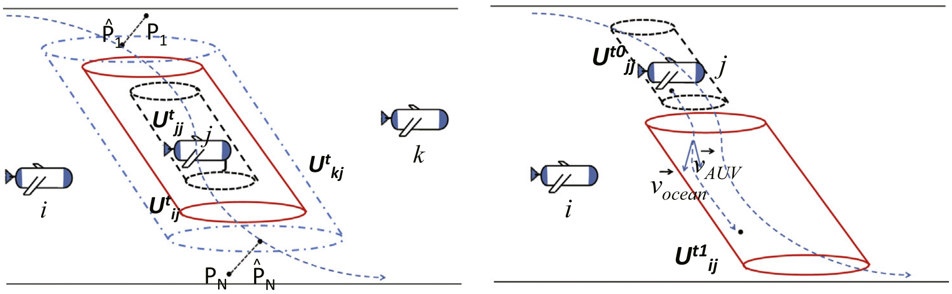
This represents the position uncertainty associated with a particular entity/node (such as an AUV) *as seen by itself*. Many approaches such as those using Kalman Filter (KF) [4,5] have been proposed to estimate this uncertainty assuming that the variables to be estimated have linear relationships among them, and that the noise is additive and Gaussian. While simple and robust, KF is not optimal when the linearity assumption among the variables does not hold. On the other hand, approaches using nonlinear filters such as the extended or unscented KF attempt to minimize the mean squared errors in estimates by jointly considering the navigation location and the sensed states such as underwater terrain features, which is non-trivial, especially in the unstructured underwater environment.

2.2. External uncertainty

This represents the position uncertainty associated with a particular entity/node *as seen by others*. Let us denote the internal uncertainty, a 3D region associated with any node $j \in \mathcal{N}$, the set of network nodes, as $\mathcal{U}_{jj}$; and the external uncertainties, 3D regions associated with j as seen by $i, k \in \mathcal{N}$, as $\mathcal{U}_{ij}$ and $\mathcal{U}_{kj}$, respectively ($i \neq j \neq k$). In general, $\mathcal{U}_{jj}, \mathcal{U}_{ij}$, and $\mathcal{U}_{kj}$ are different from each other; also, due to asymmetry, $\mathcal{U}_{ij}$ is in general different from $\mathcal{U}_{ji}$. External uncertainties may be derived from the broadcast/propagated internal-uncertainty estimates (e.g., using *one-hop or multi-hop neighbor discovery mechanisms*) and, hence, will be affected by the end-to-end (e2e) *network latency* and *information loss*. The estimation of the external-uncertainty region $\mathcal{U}_{ij}$ of a generic node j at node i (with $i \neq j$) involves the participation of both i and j. Fig. 1(a) illustrates the internal- and external-uncertainty regions and their difference; j's uncertainty regions seen by j itself ($\mathcal{U}_{jj}$, i.e., the internal uncertainty), by i (i.e., $\mathcal{U}_{ij}$) and by k (i.e., $\mathcal{U}_{kj}$) are all depicted to be different (general case). Note that, as shown in Fig. 1(b), in general, the longer an AUV remains underwater, the larger its external and internal uncertainties. Estimating $\mathcal{U}_{ij}$ involves estimating the change of $\mathcal{U}_{jj}$ with time; hence, in this work we propose a solution to predict $\mathcal{U}_{ij}$ based on the statistical estimation of $\mathcal{U}_{jj}$.

2.3. Benefits to underwater applications

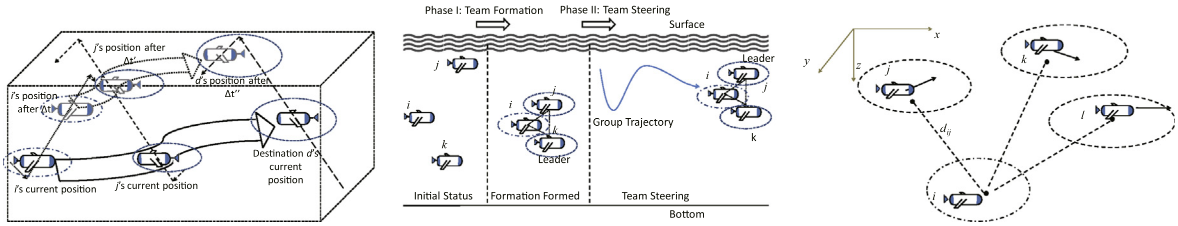
We present here applications and research areas where the proposed notion of external uncertainty can be applied to improve performance.



(a) Internal ($\mathcal{U}_{jj}$) and external uncertainty regions ($\mathcal{U}_{ij}$, $\mathcal{U}_{kj}$)

(b) Variation over time of the external uncertainty of j as seen by i

Fig. 1. Illustration of the external-uncertainty concept (note that superscript *t*, used in this work only when needed, indicates time instant *t*).



(a) Energy minimization for DTNs

(b) Collision avoidance for team formation and steering

(c) Uncertainty minimization in localization

Fig. 2. Research areas and problems that benefit from the notion of external uncertainty (broken-line circles denote external uncertainty while broken-dotted-line circles denote internal uncertainty; note that, for the sake of visualization simplicity, we use circles instead of 3D shapes).

2.3.1 Communication protocols for UW-ASNs

In UW-ASNs, the external uncertainty can be used to improve the performance of networking solutions. For example, as shown in [3], a solution that considers external uncertainty can be used for Delay-Tolerant Networks (DTNs). As shown in Fig. 2(a), by leveraging the predictability of AUVs' trajectories, delaying packet transmissions in such a way as to wait for the optimal network topology (thus trading e2e delay for throughput and/or energy consumption) can minimize communication energy consumption for delay-tolerant traffic. Also, by optimizing statistically the transmission output power, routing errors/failures can be reduced, which decreases the overall energy/bandwidth utilization.

2.3.2. Underwater robotics

In underwater robotics, a team of AUVs can collaborate to explore a 3D region and take measurements in space and time. To derive the spatio-temporal correlation of the measurements, these AUVs need to keep a geometric formation and steer through the region (Fig. 2(b)). They also need to keep a distance between each other in order to avoid vehicle collisions. In [6], a solution is proposed to minimize the time to form the geometric formation while avoiding collisions. However, that solution assumes the gliders to have correct location information, which is a strong requirement in the underwater environment. The solution can be made more robust against ocean currents and acoustic channel impairments by exploiting the concept of external uncertainty, e.g., a control algorithm can be designed to minimize the probability that two AUVs are within the collision region. This concept can also be used to adapt the sampling strategy based on the variation of the measurements.

2.3.3. Underwater localization

To perform self-localization, AUVs may need to rely on other anchor nodes (e.g., AUVs) whose positions may not be accurate, as in Fig. 2(c). Localization errors, however, may increase if an AUV relies on anchors with large position uncertainty. The external-uncertainty notion can therefore be used to decrease errors and computation complexity, e.g., by selecting the optimal subset of anchors (with small external uncertainty) so to minimize the new internal uncertainty.

2.3.4. Task allocation

Our notion of uncertainty can also be applied in task allocation, whose objective is to choose a subset of vehicles to accomplish reliably a mission with specific requirements while trying to maximize the remaining energy after the mission or to minimize the time to complete the mission itself [7]. By using the external-uncertainty notion, a team of AUVs that are "closer" to the target can be selected, which may lead probabilistically to less time and/or energy to complete the mission.

2.3.5. Data processing and visualization

Once the measurements are received by the onshore station, oceanographers need to visualize and analyze sensor data for a multitude of ocean science studies. The external-uncertainty notion can improve the quality of 3D data reconstruction because it provides vehicles' deviation from their mission path.

3. Modeling the internal uncertainty

To obtain the external uncertainty $\mathcal{U}_{ij}$, j needs to estimate its internal-uncertainty region $\mathcal{U}_{jj}$ first; then, $\mathcal{U}_{jj}$ is broadcast to its neighbors. Upon receiving this information, i will derive $\mathcal{U}_{ij}$ based on $\mathcal{U}_{jj}$. Therefore, it is necessary to estimate $\mathcal{U}_{jj}$ before $\mathcal{U}_{ij}$ can be derived. For this reason, here we first provide a statistical model for internal-uncertainty estimation; then, we propose an UKF-based solution to predict the external uncertainty.

Our internal-uncertainty model works for *any* localization algorithm as it relies on a statistical approach to estimate the confidence region using the position estimates of an AUV. We assume that each AUV follows a predictable trajectory, which is reasonable as AUVs generally need to follow the pre-planned path(s) to take measurements. The advance in mechanical, electrical, and computer engineering technologies has made it possible for AUVs to steer autonomously close to (or on) the pre-planned path(s) under disturbance such as ocean currents. Moreover, AUVs are designed to follow some regular movement pattern (e.g., saw-tooth trajectories for glider). Assume without loss of generality that the trajectory of an AUV can be described by a function $(x, y, z) = \mathbf{f}(t, \boldsymbol{\theta})$, where $\boldsymbol{\theta}$ is the list of p parameters that needs to be specified by the AUV's trajectory (note that a point on the ocean surface, e.g., the initial position of the vehicle before going underwater, is taken as the coordinate origin). Also, assume $\mathbf{f}$ is differentiable except at a countable set of discrete points. This assumption generally holds since AUVs rely on forces such as mechanical propulsion and/or buoyancy for acceleration, which results in a differentiable trajectory except at points where abrupt change is made. For example, an underwater glider follows a saw-tooth pattern, whose piecewise trajectory can be described as a line (differentiable) segment $(x, y, z) = (at + x_0, bt + y_0, ct + z_0)$. In this case, $\boldsymbol{\theta} = (a, b, c, x_0, y_0, z_0)$. Assume AUV j estimates its own coordinates, $P_n = (x_n, y_n, z_n)$, at sampling times t_n ($n = 1 \ldots N$), and its trajectory segment is of the form $P(t) = \mathbf{f}(t, \boldsymbol{\theta})$: we need to estimate $\boldsymbol{\theta}$ so that the trajectory can be determined. Note that P_n can be estimated using existing localization techniques such as dead reckoning, particle filtering, or long baseline navigation [8]. Consequently, based on the derived trajectory, j's internal uncertainty (i.e., confidence region) can be estimated, as detailed below.

From *nonlinear regression theory* in statistics, we estimate a vehicle's trajectory using the Gauss–Newton Algorithm [9], which relies on the linear approximation using the Taylor expansion $\mathbf{f}(t, \boldsymbol{\theta}) \approx \mathbf{f}(t, \boldsymbol{\theta}^{(0)}) + \mathbf{Z}^0(\boldsymbol{\theta} - \boldsymbol{\theta}^{(0)})$, where $\mathbf{Z}^0 = \mathbf{Z}_{|\boldsymbol{\theta}=\boldsymbol{\theta}^{(0)}} = \frac{\partial \mathbf{f}}{\partial \boldsymbol{\theta}}_{|\boldsymbol{\theta}=\boldsymbol{\theta}^{(0)}}$ and $\boldsymbol{\theta}^{(0)}$ is the vector that provides the initial values of these parameters in order to start the algorithm. The objective of our estimation is to find the best $\hat{\boldsymbol{\theta}}$ such that $S(\boldsymbol{\theta}) = \sum_{n=1}^{N} \|P_n - \mathbf{f}(t_n, \boldsymbol{\theta})\|^2$ is minimized. The idea of the Gauss–Newton Algorithm is to find $\hat{\boldsymbol{\theta}}$ through iteration, i.e., to update $\boldsymbol{\theta}^{(q)}$ iteratively until convergence. From nonlinear regression theory, it has been shown that $\hat{\boldsymbol{\theta}} - \boldsymbol{\theta}^{(0)} = (\mathbf{Z}^T\mathbf{Z})^{-1}\mathbf{Z}^T\boldsymbol{\epsilon}^{(0)}$, where $\boldsymbol{\epsilon}^{(0)} = P(t) - \mathbf{f}(t, \boldsymbol{\theta}^{(0)})$. Starting from the initial position, we have the following iterative formula to estimate $\boldsymbol{\theta}^{(q)}$, i.e., $\boldsymbol{\theta}^{(q+1)} = \boldsymbol{\theta}^{(q)} + \boldsymbol{\delta}^{(q)}$, where $\boldsymbol{\delta}^{(q)} = [(\mathbf{Z}^q)^T\mathbf{Z}^q]^{-1}\mathbf{Z}^q\boldsymbol{\epsilon}^{(q)}$ and $\boldsymbol{\epsilon}^{(q)} = P(t) - \mathbf{f}(t, \boldsymbol{\theta}^{(q)})$.

From statistical inference theory, we obtain asymptotically $\hat{\boldsymbol{\theta}} \sim \mathcal{N}(\boldsymbol{\theta}, \sigma^2\mathbf{C}^{-1})$, where $\mathcal{N}()$ is the normal distribution, $\mathbf{C} = \mathbf{Z}^T\mathbf{Z}$, and σ^2 is the variance; this leads to $\frac{\mathbf{a}^T\hat{\boldsymbol{\theta}} - \mathbf{a}^T\boldsymbol{\theta}}{\hat{\sigma}(\mathbf{a}^T\mathbf{C}^{-1}\mathbf{a})^{1/2}} \sim t_{N-p}$, where t_{N-p} is the t-distribution with $N - p$ degrees of freedom [9], $\mathbf{a}^T = [0, 0, \ldots, 0, 1, 0, \ldots, 0]$ is a vector with p elements, $\hat{\sigma} = S(\hat{\boldsymbol{\theta}})/(N - p)$. Note that this actually gives the confidence interval for each element in $\boldsymbol{\theta}$. It is also proven in [9] that asymptotically

$$\frac{\mathbf{f}(t_N, \boldsymbol{\theta}) - \mathbf{f}(t_N, \hat{\boldsymbol{\theta}})}{\hat{\sigma}\left[1 + (\mathbf{Z}_N)^T\mathbf{Z}^T\mathbf{Z}^{-1}\mathbf{Z}_N\right]^{1/2}} \sim t_{N-p}. \tag{1}$$

Finally, the internal-uncertainty region is given by the approximate $100(1-\alpha)\%$ confidence interval at t_N as,

$$\mathbf{f}(t_N, \hat{\boldsymbol{\theta}}) \pm \hat{t}^{\alpha/2}_{N-p}\hat{\sigma}\left[1 + (\mathbf{Z}_N)^T\mathbf{Z}^T\mathbf{Z}^{-1}\mathbf{Z}_N\right]^{1/2}, \tag{2}$$

where $\hat{t}^{\alpha/2}_{N-p}$ is the $100(1-\alpha/2)\%$ of the t-distribution with $N - p$ degrees of freedom and $\mathbf{Z}_N = \frac{\partial \mathbf{f}(t_N, \boldsymbol{\theta})}{\partial \boldsymbol{\theta}}$ is the Jacobian matrix of $\mathbf{f}()$ at t_N.

3.1. Note

Our solution is based on the first-order Taylor expansion. This method is known as Delta Method in statistics. This basic method can be extended to second-order Taylor expansion. However, to use second-order Delta method, it is generally required that the first-order derivative $\frac{\partial \mathbf{f}}{\partial \boldsymbol{\theta}}_{|\boldsymbol{\theta}=\boldsymbol{\theta}^{(0)}} = 0$ and the second-order derivative be non-zero [10]. Under such conditions, $\mathbf{f}(t_N, \boldsymbol{\theta}) - \mathbf{f}(t, \boldsymbol{\theta})$ will converge in distribution to $\chi^2_{|\boldsymbol{\theta}|}$. If the first-order derivative is non zero, estimating the distribution of the second-order Taylor expansion is more complicated as we need to consider the first and second power of the random variables together. Extending to higher-order Taylor expansion will be far more complicated. In general, adding higher-order expansion achieves more prediction accuracy; however, it also makes the prediction far more complicated and the extra accuracy achieved is often not worth the increased complexity.

4. Modeling the external uncertainty

To present our external-uncertainty model, we start from the estimation of the one-hop external uncertainty; then, we show how this estimate can be used to adjust dynamically the update interval; last, but not least, we extend the one-hop estimate to the case in which a node is multiple hops away. Note that in this work we focus on algorithms to model the external uncertainty. We assume that the vehicles are synchronized in time initially through GPS (when on the surface) and time synchronization algorithms are run to maintain a *coarse* synchronization when underwater, i.e., the nodes synchronize the time with a selected reference node (which may have time drifting).

4.1. One-hop external uncertainty estimation

After receiving j's trajectory and the internal-uncertainty region parameters, i.e., $\hat{\theta}, \mathbf{f}(t_N, \hat{\theta})$, and $\hat{t}^{\alpha/2}_{N-p}\hat{\sigma}[1+ (\mathbf{Z}_N)^T\mathbf{Z}^\mathbf{T}\mathbf{Z}^{-1}\mathbf{Z}_N]^{1/2}$, AUV i can update the estimate of j's external-uncertainty region. Due to packet delays and losses in the network, j's external-uncertainty regions as seen by single- and multi-hop neighbors are *delayed versions* of j's own internal uncertainty. Hence, when using *multi-hop neighbor discovery schemes*, the internal uncertainty $\mathcal{U}_{jj}$ provides a *lower bound* for all the external uncertainties associated with that node, $\mathcal{U}_{ij}, \forall i \in \mathcal{N}$. Consequently, we derive $\mathcal{U}_{ij}$ based on the received $\mathcal{U}_{jj}$.

We use UKF to predict how the internal uncertainty 'propagates' (and in general 'deteriorates') through the network. This is done in two steps, as detailed below: *(1) Region prediction*, to predict the current position of an AUV assuming that its previous location is at a point in the internal-uncertainty region; then, the external-uncertainty region is obtained by taking the set containing these predicted positions and *(2) Distribution estimation*, to calculate the probability density function (pdf) of the current position by integrating the internal-uncertainty pdf over points with the same predicted position.

4.1.1. Region prediction

AUV i first predicts j's position assuming j is located at a point in $\mathcal{U}_{jj}$ and then considers the union of all these predicted points. The movement model of j can be described using the following nonlinear dynamical system (note that the equivalent discrete-time dynamic equation can be derived as in [11] by means of the state-space method using iterations); AUV i estimates the state from step $q = 1$ whenever $\mathcal{U}_{jj}$ is received, where q is incremented until a new $\mathcal{U}_{jj}$ is received (in which case q is reset to 1). Hence,

$$\mathbf{s_j^q} = \mathbf{F_j s_j^{q-1}} + \mathbf{o}(\mathbf{s_j^{q-1}}) + \mathbf{G u_j^{q-1}} + \mathbf{B w_j^{q-1}} \qquad (3)$$

represents the state-transition equation for the system describing the motion of AUV j between steps $q-1$ and q (spaced T [s] apart). Here, $\mathbf{s_j^q} = [x_j^q, y_j^q, z_j^q, \dot{x}_j^q, \dot{y}_j^q, \dot{z}_j^q, v_{j,x}^{oc}, v_{j,y}^{oc}, v_{j,z}^{oc}]^T$ represents 3D position, velocity, and ocean-current velocity of AUV j at step q, $\mathbf{o}(\mathbf{s_j^{q-1}})$ is the ocean-current prediction function (which is generally nonlinear), $\mathbf{u_j^{q-1}} = [u_j^{q-1,x}, u_j^{q-1,y}, u_j^{q-1,z}]^T$ is the control input for $t \in [(q-1)T, qT)$ (which is used not to offset the drifting, but to steer the AUV to move along its trajectory), and $\mathbf{w_j^{q-1}} = [w_j^{q-1,x}, w_j^{q-1,y}, w_j^{q-1,z}, w_{oc,j}^{q-1,x}, w_{oc,j}^{q-1,y}, w_{oc,j}^{q-1,z}]^T$ represents the discrete random acceleration caused by non-ideal noise in the control input and/or the variation in ocean current speed. Throughout this article, when used as superscript, T indicates matrix transpose; otherwise, it represents the time interval. Note that $\mathbf{o}(\mathbf{s_j^{q-1}})$ can be predicted using ocean-current models or data from real-time onshore ocean observing systems; also, note that, as AUVs are spaced a few kilometers apart, the currents affecting the AUVs are generally different. Ocean current is affected by many complicated factors (which are global information) including (but not limited to) surface wind, Coriolis effect, temperature and salinity differences, gravitational pull of the Moon and the Sun, depth contours, shoreline configurations and interaction with other currents. It is unrealistic for the resource-limited AUV to get all these global data and run complex models that considers all these factors to predict the current velocity, even though simplified model may probably be used (in this case it will incur more prediction errors). On the other hand, the onshore systems can use powerful super-computers or data centers to finish such estimation and send back the prediction to the AUV through the acoustic communication networks.

In (3), $\mathbf{F_j}, \mathbf{G}$, and $\mathbf{B}$ are matrices to adjust the state $\mathbf{s_j^q}$ according to the previous state, control input, and random acceleration noise, respectively, and are defined as,

$$\mathbf{F_j} = \begin{bmatrix} \mathbf{I}_3 & T_j'\mathbf{I}_3 & T_j'\mathbf{I}_3 \\ 0 & \mathbf{I}_3 & \mathbf{I}_3 \\ 0 & 0 & \mathbf{I}_3 \end{bmatrix}, \quad \mathbf{G} = \begin{bmatrix} 0 \\ \mathbf{I}_3 \\ 0 \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} 0 & 0 \\ \mathbf{I}_3 & 0 \\ 0 & \mathbf{I}_3 \end{bmatrix},$$

where $\mathbf{I}_3$ is the 3×3 identity matrix, T_j' [s] is the difference between the current time and the last time when $\mathcal{U}_{jj}$ was estimated or the last update time that UKF was run, i.e., $T_j' = t_{now} - t_{\mathcal{U}_{jj}}$ if i receives j's updated internal uncertainty after the last UKF update, whereas $T_j' = T$ if i does not receive j's update message, where $t_{\mathcal{U}_{jj}}$ is the time when $\mathcal{U}_{jj}$ is estimated by j and T is the UKF update interval.

The variable $\mathbf{w_j^{q-1}}$ represents 3D samples of discrete-time white Gaussian noise; hence, $\mathbf{w_j^{q-1}} \sim \mathcal{N}(0, \mathbf{Q})$, where $\mathbf{Q} \geqslant 0$ is the covariance matrix of the process. The random acceleration is also assumed to be independent on the three axes. Here we assume that an AUV can measure the ocean-current velocity using sensors such as Acoustic Doppler Current Profiler (ADCP), which are, however, expensive. For AUVs without ADCP, we can force the state for ocean current to be zero; in this case, the model would reduce to a linear KF and the effect of ocean current should be treated as noise, which is accounted by $\mathbf{w_j^{q-1}}$.

It is worth noting that (3) includes delays due to transmission, propagation, reception, and packet loss. As the ocean-current velocity is generally nonlinear, (3) expresses a nonlinear relationship between $\mathbf{s_j^q}$ and $\mathbf{s_j^{q-1}}$. Therefore, a nonlinear Kalman filter should be used. In this work we use UKF because this filter can provide more accurate prediction than the extended KF (another type for nonlinear prediction) while having the same computation complexity of $\mathcal{O}(L^3)$ for state estimation, where L (9 in our case) is the dimension of the state variable, as proven in [12].

The position observed by the AUV at step q is related to the state by the *measurement equation*, $\mathbf{P_j^q} = \mathbf{H s}_j^\mathbf{q} + T_j'\widetilde{\mathbf{C}}\mathbf{v_j^{q-1}}$, where $\mathbf{P_j^q} = [P_j^{q,x}, P_j^{q,y}, P_j^{q,z}]$ represents the *observed position* of the AUV at step q; here, $\mathbf{H} = [\mathbf{I}_3 \quad 0 \quad 0]$ is the matrix that extracts the position, whereas $\widetilde{\mathbf{C}} = [0 \quad \mathbf{I}_3 \quad 0]^T$ adds the noise. The variable $\mathbf{v_j^{q-1}} = [v_j^{q-1,x}, v_j^{q-1,y}, v_j^{q-1,z}]^T$ represents the *measurement noise* in *velocity*, expressed as 3D samples of discrete-time white Gaussian noise. Hence, $\mathbf{v_j^q} \sim \mathcal{N}(0, \mathbf{R})$, where $\mathbf{R} \geqslant 0$ is the covariance matrix of the process. The observed position of the AUV $\mathbf{P_j^q}$ is therefore the actual position of the AUV affected by a measurement noise, which we represent as a Gaussian variable.

The UKF algorithm provides a computationally efficient set of recursive equations to estimate the state of such process, and can be proven to be the optimal filter in the minimum square sense [13]. To implement the UKF algorithm, we need to extend the state vector to the augmented vector $\mathbf{s_j^{q,+}} = [(\mathbf{s_j^q})^T, (\mathbf{w_j^{q-1}})^T, (\mathbf{v_j^{q-1}})^T]^T$ and use the corresponding covariance matrix $\mathbf{V_j^{q,+}} = \mathbb{E}[\mathbf{s_j^{q,+}}(\mathbf{s_j^{q,+}})^T]$. The use of UKF at AUV i reduces the number of necessary location updates. In fact, the filter is used to *estimate the position* at the AUV based on measurements, which is a common practice in robotics, and to *predict* the position of the AUVs thus limiting message exchange (i.e., reducing the need for frequent position updates). The position of j can be estimated and predicted at i based on past estimates $\mathbf{P_j^q}$. To update the state vector, i needs to calculate a so-called $L \times (2L+1)$ sigma point matrix $\boldsymbol{\chi}_{j,q-1}$ with the following column vectors $\boldsymbol{\chi}_{j,q-1}^m$ $(m = 0, \ldots, 2L)$, i.e., for $m = 0$, $\boldsymbol{\chi}_{j,q-1}^0 = \mathbf{s_j^{q,+}}$; for $m = 1, \ldots, L$, $\boldsymbol{\chi}_{j,q-1}^m = \mathbf{s_j^{q,+}} + \left[(L+\lambda)\mathbf{V_j^{q,+}}\right]_m^{1/2}$; for $m = L+1, \ldots, 2L$, $\boldsymbol{\chi}_{j,q-1}^m = \mathbf{s_j^{q,+}} - \left[(L+\lambda)\mathbf{V_j^{q,+}}\right]_m^{1/2}$.

Here $[(L+\lambda)\mathbf{V_j^{q,+}}]_m^{1/2}$ is the m-th column of the matrix square root of $(L+\lambda)\mathbf{V_j^{q,+}}$, where $\lambda = \varsigma^2(L+\kappa) - L$ is a scaling factor depending on ς and κ that controls the spread of the sigma points. These sigma vectors $\boldsymbol{\chi}_{j,q-1}^m$ $(m = 0, \ldots, 2L)$ are propagated through the nonlinear state estimate, i.e., (3), denoted by $\mathcal{T}$ here, $\mathcal{Y}_{j,q-1}^m = \mathcal{T}(\boldsymbol{\chi}_{j,q-1}^m)$. The state and covariance are predicted by recombining these weighted sigma points, i.e.,

$$\hat{\mathbf{s}}_{\mathbf{j}}^{\mathbf{q}-} = \sum_{m=0}^{2L} W_s^m \boldsymbol{\chi}_{j,q-1}^m, \tag{4}$$

$$\hat{\mathcal{Y}}_j^{q-1} = \sum_{m=0}^{2L} W_s^m \mathcal{Y}_{j,q-1}^m, \tag{5}$$

$$\mathbf{V_j^{q-}} = \sum_{m=0}^{2L} W_c^m [\mathcal{Y}_{j,q-1}^m - \hat{\mathbf{s}}_{\mathbf{j}}^{\mathbf{q}}][\mathcal{Y}_{j,q-1}^m - \hat{\mathbf{s}}_{\mathbf{j}}^{\mathbf{q}}]^T, \tag{6}$$

where $W_s^0 = \lambda/(L+\lambda), W_c^0 = \lambda/(L+\lambda) - (1 - \varsigma^2 + \beta)$, and $W_s^m = W_c^m = 1/[2(L+\lambda)]$ for $m = 1, \ldots, 2L; \beta$ is related to the distribution of $\mathbf{s}$. Normal values are $\varsigma = 10^{-3}$, $\kappa = 1$, and $\beta = 2$. If the distribution of $\mathbf{s}$ is Gaussian, $\beta = 2$ is optimal. Here the superscript q-means that the state $\hat{\mathbf{s}}_{\mathbf{j}}^{\mathbf{q}-}$ or $\mathbf{V_j^{q-}}$ covariance estimate is *a priori* estimate. Eqs. (5) and (6) describe how i predicts the state of AUV j before receiving the measurement (*a priori* estimate). Then, i projects the covariance matrix ahead. Once received the measurement $\mathbf{P_j^q}, i$ updates the Kalman gain $\mathbf{K_j^q}$, and corrects the state estimate and covariance matrix according to the measurement, i.e.,

$$\mathbf{V}_{j,\tilde{y}_q\tilde{y}_q} = \sum_{m=0}^{2L} W_s^m [\mathcal{Y}_{j,q}^m - \hat{\mathcal{Y}}_{j,q-1}^m][\mathcal{Y}_{j,q}^m - \hat{\mathcal{Y}}_{j,q-1}^m]^T,$$

$$\mathbf{V}_{j,\tilde{\mathbf{s}}_q\tilde{y}_q} = \sum_{m=0}^{2L} W_s^m [\mathbf{s}_{\mathbf{j}}^{q,m} - \hat{\mathbf{s}}_{\mathbf{j}}^{q-1,m}][\mathcal{Y}_{j,q}^m - \hat{\mathcal{Y}}_{j,q-1}^m]^T,$$

$$\mathbf{K_j^q} = \mathbf{V}_{j,\tilde{\mathbf{s}}_q\tilde{y}_q}(\mathbf{V}_{j,\tilde{y}_q\tilde{y}_q})^{-1}, \tag{7}$$

$$\hat{\mathbf{s}}_{\mathbf{j}}^{\mathbf{q}} = \hat{\mathbf{s}}_{\mathbf{j}}^{\mathbf{q}-} + \mathbf{K_j^q}(\mathcal{Y}_j^q - \mathcal{Y}_j^{q-1}), \tag{8}$$

$$\mathbf{V_j^q} = \mathbf{V_j^{q-}} - \mathbf{K_j^q}\mathbf{V}_{j,\tilde{y}_q\tilde{y}_q}(\mathbf{K_j^q})^T, \tag{9}$$

where (7) updates the Kalman gain, (8) calculates the new state (*a posteriori estimate*), and (9) updates the covariance matrix. Note that the complexity of the above computations is the same as in the extended KF [12] and that the processing cost at i is much lower than the communication cost (the output power used by acoustic transmitters underwater is in the order of tens of Watts).

Finally, if we denote the UKF filtering at time t_q from position $\mathbf{p}$ at t_{q-1} as $h_{UKF}(t_q, \mathbf{p}, t_{q-1})$, the predicted external-uncertainty region at step q is $\mathcal{U}_{ij}^q = \{h_{UKF}(qT, \mathbf{p}, (q-1)T) | \mathbf{p} \in \mathcal{U}_{ij}^{q-1}\}$, which, for simplicity, we further simplify in $\mathcal{U}_{ij}^q = h_{UKF}(qT, \mathcal{U}_{ij}^{q-1}, (q-1)T)$.

4.1.2. Distribution estimation

Assume $\mathbf{p} \in \mathcal{U}_{ij}^0$ is predicted from point $\mathbf{p}'$ at step $q-1$, i.e., $\mathbf{p} = h_{UKF}(qT, \mathbf{p}', (q-1)T)$, $\mathbf{p}' \in \mathcal{U}_{ij}^{q-1}$. The pdf $g_{ij}^q(\mathbf{p})$ of the external uncertainty $\mathcal{U}_{ij}^q$ at step q can be derived from the pdf $g_{ij}^{q-1}(\mathbf{p})$ of $\mathcal{U}_{ij}^{q-1}$ as

$$g_{ij}^q(\mathbf{p}) = \int_{\mathbf{p}=h_{UKF}(qT,\mathbf{p}',(q-1)T),\mathbf{p}'\in\mathcal{U}_{ij}^{q-1}} g_{ij}^{q-1}(\mathbf{p}')d\mathbf{p}'.$$

With the help of UKF and the probability theory, we can derive the external uncertainty and its pdf. Note that the initial pdf $g_{ij}^0(\mathbf{p})$ is the t-distribution on $\mathcal{U}_{jj}$ (i.e., $\mathcal{U}_{jj}^0$) received from j. To reduce the complexity, we convert an uncertainty region (internal or external) into its discrete counterparts, i.e., we divide an uncertainty region into a finite number of equal-size small regions. When the number of small regions is sufficiently large, the UKF filtering on each small region can be approximated by the UKF filtering on a point – e.g., the *centroid* – in this small region. Hence, the predicted external-uncertainty region can be approximated as the region contained in the hull of these predicted points. The pdf functions are also approximated by the probability mass functions on discrete points, which simplifies the pdf estimation after UKF filtering.

4.2. Adjustment of the UKF update interval

So far, we have assumed the update interval T for the UKF algorithm to be *fixed*. A small T determines frequent updates, i.e., the estimation error is corrected in a timely manner; however, frequent external-uncertainty estimations lead to waste of computation resources and energy, causing large network overhead. On the other hand, a large T would save such resources; yet, it may lead to large estimation errors (due to slow update and correction) and thus worse overall performance. To capture this tradeoff, we propose an algorithm to maximize T (i.e., to minimize the update overhead) while keeping the estimation error within an acceptable range: AUV j selects the optimal value T^* such that the prediction errors of all its neighbors (denoted by $\mathcal{N}_j$) are below a specified threshold $e_{\max}$. To do this, j needs to estimate the prediction errors of its neighbors. Say, j estimates the prediction error for i. At each step q, each AUV j emulates the prediction procedure performed at i, calculates its actual new position by filtering the new measurement. Then, j checks to see if the probability of i's prediction error being greater than a maximum error $e_{\max}$ is within a probability threshold, i.e., if $\Pr\{\|\mathbf{P_j^q} - \mathbf{H}\hat{\mathbf{s}}_\mathbf{j}^\mathbf{q}\| > e_{\max}\} < \gamma$. To make the formulation clear, we denote $\mathbf{P_j^q}, \hat{\mathbf{s}}_\mathbf{j}^\mathbf{q}$, and $\mathbf{v_j^q}$ by $\mathbf{P_{ij}^q}, \hat{\mathbf{s}}_\mathbf{ij}^\mathbf{q}$, and $\mathbf{v_{ij}^q}$, respectively. Letting $\mathbf{\Xi_{ij}^q} = \mathbf{P_j^q} - \mathbf{H}\hat{\mathbf{s}}_\mathbf{j}^\mathbf{q}$, this condition transforms into the compact expression: $\Pr\{\mathbf{\Xi_{ij}^q}(\mathbf{\Xi_{ij}^q})^T > e^2_{\max}\} < \gamma$. From the measurement equation $\mathbf{P_{ij}^q} = \mathbf{H}\mathbf{s_{ij}^q} + T\widetilde{\mathbf{C}}\mathbf{v_{ij}^{q-1}}$, assuming $\mathbf{v_{ij}^{q-1}} \sim \mathcal{N}(\mathbf{0}, \xi_{ij}^2\mathbf{I}_3)$, we can see that $\mathbf{\Xi_{ij}^q}(\mathbf{\Xi_{ij}^q})^T/(\xi_{ij}^2 T^2)$ has the χ^2-distribution χ_3^2 (note that $\mathbf{\Xi_{ij}^q}$ has 3 elements). From the probability constraint, for each i, the maximal time update interval is therefore $T = \frac{e_{\max}}{\xi_{ij}\sqrt{\hat{\chi}_{\gamma,3}}}$, where $\hat{\chi}_{\gamma,3}$ is the $(1-\gamma)\%$ of χ_3^2-distribution; therefore, $T^* = \min_{i \in \mathcal{N}_j} \frac{e_{\max}}{\xi_{ij}\sqrt{\hat{\chi}_{\gamma,3}}}$.

To sum up, the steps to adjust update interval are: first, AUV j sends out its own internal-uncertainty at the pre-set period; and then it computes $\mathbf{\Xi_{ij}^q}$ in order to calculate ξ_{ij}, which is in turn to get the optimal update interval T^*. When abrupt change occurs, it resets T back to the preset value.

4.3. External-uncertainty estimation across multiple links

For a multi-hop neighbor AUV j, depending on the selection of the path to j, the estimated uncertainty region may be different. This is because j's uncertainty region estimated by intermediate vehicles is generally different for different paths. This depends on factors such as availability of ocean-current information, packet loss, communication delays (which introduces asynchronous updates of the external uncertainty at different nodes). Our objective for the multi-hop estimation is to select the estimate that gives minimum uncertainty. To compare the degree of uncertainty, we use *information entropy* as the metric, i.e.,

$$\mathcal{H}_{\mathcal{U}_{ij}} = -\int_{\mathbf{p} \in \mathcal{U}_{ij}} g_{ij}(\mathbf{p}) \log(g_{ij}(\mathbf{p})) d\mathbf{p}; \tag{10}$$

here, the bigger $\mathcal{H}_{\mathcal{U}_{ij}}$, the more uncertain $\mathcal{U}_{ij}$. The reason to use this metric instead of simply using the size of the uncertainty region is that the entropy characterizes better the uncertainty. To estimate the j's uncertainty region as it propagates along a path r_{ij}, i estimates the uncertainty region broadcast by $k = prev(i, r_{ij})$, which is i's previous hop along r_{ij}. The estimated uncertainty region $\mathcal{U}_{ij,k \in r_{ij}}$ at time t_{now} is denoted as,

$$\mathcal{U}_{ij,r_{ij}} = h_{UKF}(t_{now}, \mathcal{U}_{kj,prev(k,r_{ij})}, t_{kj,prev(k,r_{ij})}), \tag{11}$$

where $\mathcal{U}_{kj,prev(k,r_{ij})}$ is the most recently received estimate of $\mathcal{U}_{kj}$ by k that is sent at time $t_{kj,prev(k,r_{ij})}$. If we denote the set of all paths from i to j as $\mathcal{P}_{ij}$, then the external-uncertainty region of j estimated by i is $\mathcal{U}_{ij} = \arg\min_{\mathcal{U}_{ij,r_{ij}}, r_{ij} \in \mathcal{P}_{ij}} \mathcal{H}_{\mathcal{U}_{ij,r_{ij}}}$. Note that this multi-hop estimation will incur low overhead as, from (11), we see that this estimation is performed recursively, i.e., i can use its neighbor's external-uncertainty estimation for multi-hop node j to estimate $\mathcal{U}_{ij}$.

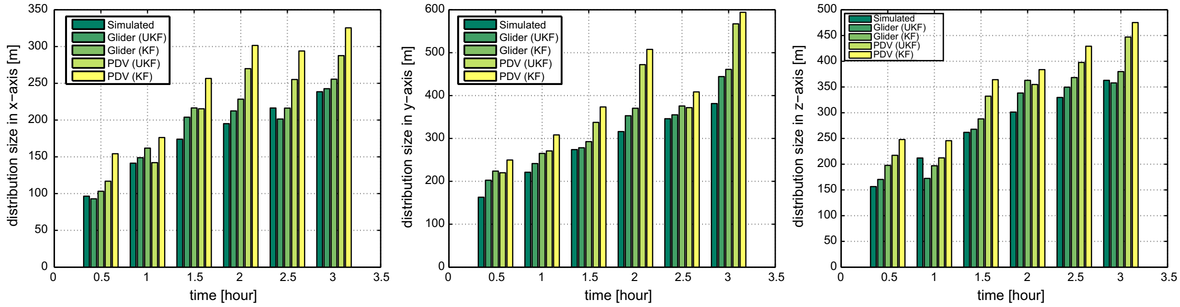
5. Performance evaluation

We present here the assumptions and setup that our simulations are based on. In our simulations, the acoustic channel is modeled as in [3]. We assume that an AUV estimates its own position underwater by Dead Reckoning, i.e., using previous position and estimated vehicle velocity (any other localization method can be used too). We assume that an AUV's drift (i.e., the relative displacement from its trajectory) is a 3D Markov process whose drifting in any of the x, y, z direction is Gaussian and whose magnitude along each direction in any time interval φ is $\sqrt{\varphi}\phi$, where ϕ is a scaling factor [14]. The simulation parameters are listed in Table 1; the AUVs are initially randomly deployed in the 3D region. We use typical velocities for PDVs varying from 2 to 10 K m/h [7]. The PDV velocity is dependent on various nonlinear factors like drag force and motor friction. For gliders, we assume the trajectory segment is described by a linear form, whereas for PDVs by the quadratic form $(x, y, z) = 0.5\zeta t^2 + \boldsymbol{\eta} t + \mathbf{P}_0$, where $\zeta, \boldsymbol{\eta}$, and $\mathbf{P}_0$ denote the acceleration, velocity, and initial position of a PDV, respectively, as in the kinematic model in [15]. A glider initially starts from its position with randomly distributed in the 3D region, then it glides randomly up/down to the bottom at an angle that is uniformly distributed in the pitch angle range given in Table 1; when it reaches the boundary given, it changes its direction and glides to the other direction at another random angle, and so on. For PDVs, we model their typical kinematic trajectory by choosing the trajectory to be piecewise quadratic curves (described as above) in a vertical plane, where for each piece $|\zeta|$ is uniformly distributed at $[0.1, 0.4]$ m/s^2 (typical PDV acceleration speed) and $|\boldsymbol{\eta}|$ being uniformly distributed at $[0.5, 2.8]$ m/s (typical PDV speed).

Note that our sampling/update interval is taken to be 30 s, which is also the interval between samples used for estimating the AUV trajectory in Section 3. For each trajectory segment we use the samples collected during this segment. Hence, the time window used to estimate the

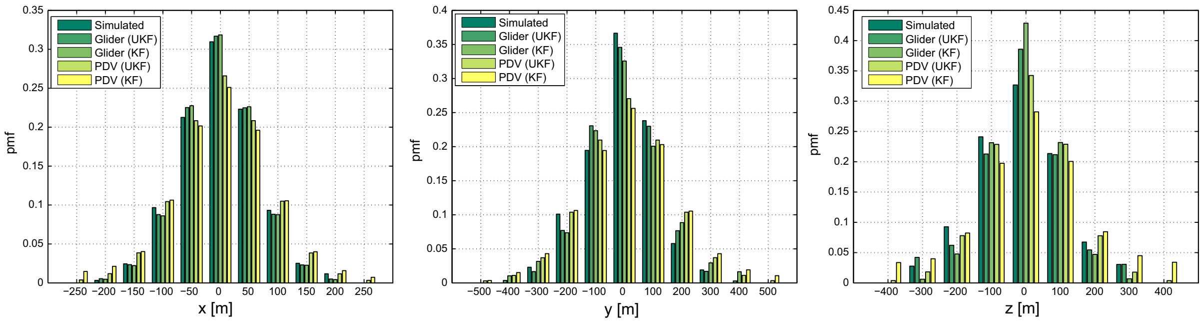
Table 1
Emulation parameter values.

Parameter	Value
Initial deployment region	2.5(L) × 2.5(W) × 1(H) K m^3
Sampling/update interval	30 s
Transmission power	[1, 10] W
Glider horizontal speed	0.3 m/s
Gliding depth range	[0, 500] m
Pitch angle range	[10°, 35°]



(a) Estimated sizes along the x coordinate (b) Estimated sizes along the y coordinate (c) Estimated sizes along the z coordinate

Fig. 3. External-uncertainty prediction accuracy: estimated 3D region sizes.



(a) pmf along the x coordinate (b) pmf along the y coordinate (c) pmf along the z coordinate

Fig. 4. External-uncertainty prediction accuracy: estimated 3D region probability mass functions (pmfs).

AUV location starts from the time when the AUV changes the current trajectory. Further investigation can be done using a fixed number of the last samples to estimate the location and its associated uncertainty, and then derive the optimal value for this number.

5.1. External-uncertainty prediction accuracy

We are interested in comparing the external-uncertainty prediction accuracy of our proposed UKF algorithm with that predicted using a simple KF. We compare the 3D sizes and probability mass functions (pmfs) between those obtained in simulations and those predicted by our model. To obtain statistical relevance in the results, simulations of 100 rounds were performed for predictions of gliders and PDVs, and the average results are plotted in Figs. 3 and 4. Note that by 'Glider (UKF)' and 'Glider (KF)' we denote the uncertainty for a glider predicted using the UKF and KF, respectively; a similar notation is also used for PDVs. From these figures, we can see that our external-uncertainty model using UKF gives more accurate predictions than that using KF on the region sizes and distribution functions for both types of vehicles. In any of these axis, the vehicle may be randomly located in a range $[\tau - \rho/2, \tau + \rho/2]$, where τ is the expected location of this vehicle in this direction. We call ρ the size of the uncertainty region as it determines the range the AUV may be distributed in. Fig. 3 plots these sizes at different times, where the horizontal axis is the time duration that an AUV remains underwater. We assume that at time 0 there is no position uncertainty (e.g., AUVs are on the ocean surface where GPS is accessible) and that estimations of the external uncertainty are run at the same time. To compare the distribution functions, in Fig. 4 we also align the pmfs (i.e., move the expected positions of the vehicle in these three cases to 0). Each pmf value at a discrete position, say x_0, is calculated by checking if the vehicle lies in $[x_0 - \psi/2, x_0 + \psi/2]$, where ψ is the interval size. Note that the prediction accuracy for gliders is generally better than that for PDVs: this is because gliders follow saw-tooth trajectories (piecewise line segments), which are easier to predict than the nonlinear trajectories followed by PDVs. This is because parametric uncertainties like ocean current uncertainty make it more difficult to estimate nonlinear function (i.e., trajectory of PDVs) than linear function (i.e., trajectory of gliders) [16]. Also, note that the longer an AUV stays underwater, the less accurate the prediction is: provided an accuracy threshold, our model can be used for AUVs to decide when to surface for position correction (e.g., to get a GPS fix).

5.2. Impact on UW-ASNs

To appreciate the impact of using the external uncertainty on underwater communications and networking, we implemented the external-uncertainty model in QUO VADIS [3], a Quality of Service (QoS)-aware communication optimization framework we developed for UW-ASNs. Using our position-prediction model, QUO VADIS minimizes energy consumption for communication by delaying

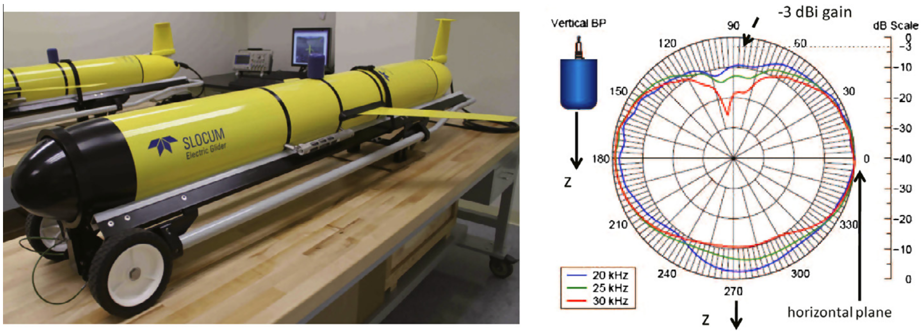
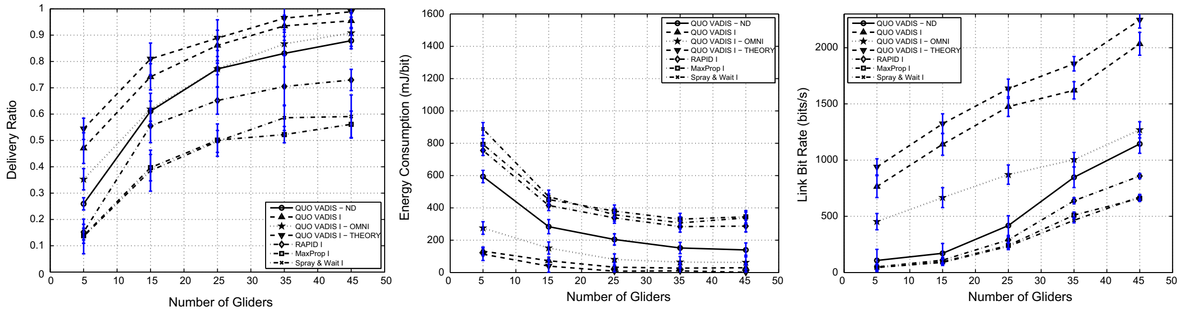


Fig. 5. SLOCUM glider with a BT-25UF transducer on top of the payload (left); horizontal plane radiation pattern of the BT-25UF (right).

packet transmissions in order to wait for a favorable network topology (thus trading e2e delay for energy and/or throughput). QUO VADIS leverages the predictability of AUV trajectories to estimate the best future positions of other AUVs and forwards the packet at the optimal time so to minimize the e2e communication energy. We compare QUO VADIS with well-known DTN solutions, i.e., RAPID [17], MaxProp [18], and Spray and Wait [19], which do not consider the position uncertainty of AUVs. QUO VADIS is proposed to serve two classes of traffic: *Class I*, i.e., *delay-tolerant, loss-tolerant traffic*, and *Class II*, i.e., *delay-tolerant, loss-sensitive traffic*. We denote QUO VADIS for Class I traffic using the BT-25UF acoustic transducer (a device to convert acoustic into electrical energy and viceversa – see Fig. 5), for Class I traffic using the ideal omni-directional transducer, for Class II traffic using the BT-25UF transducer, for Class I traffic using the ideal omni-directional transducer, and the solution with no delaying of the transmission as 'QUO VADIS I', 'QUO VADIS I–OMNI', 'QUO VADIS II', 'QUO VADIS II–OMNI', and 'QUO VADIS–ND', respectively. Also, we denote the RAPID solution with Class I constraints as 'RAPID I' and the solution with Class II constraints as 'RAPID II'. Last, we are interested to see the theoretical performance of our QUO VADIS (denoted by 'QUO VADIS–THEORY'), i.e., AUVs stay well on the planned trajectory and there is no localization uncertainty.

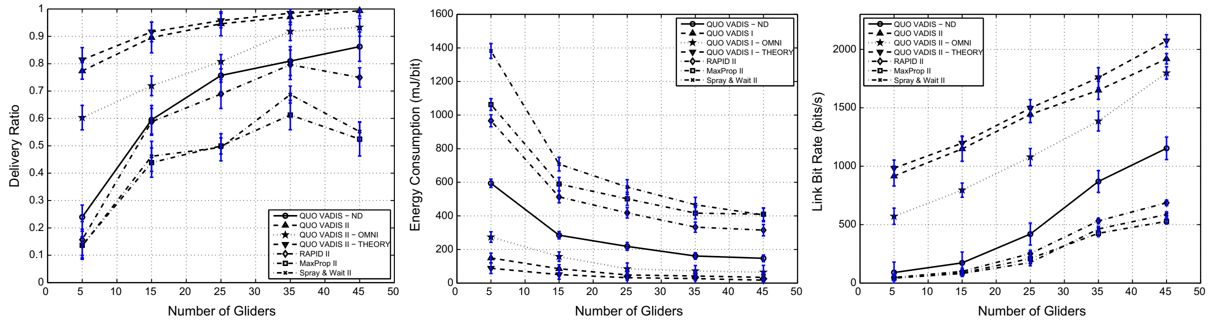
These existing DTN solutions have been proposed for communications within extreme and performance-challenged environments where continuous e2e connectivity does not hold most of the time [20,21]. Many approaches such as Resource Allocation Protocol for Intentional DTN (RAPID) routing [17], Spray and Wait [19], and MaxProp [18], are solutions mainly for intermittently connected terrestrial networks. RAPID [17] translates the e2e routing metric requirement such as minimizing average delay, minimizing worst-case delay, and maximizing the number of packets delivered before a deadline into per-packet utilities. At a transfer opportunity, it replicates a packet that locally results in the highest increase in utility. To estimate the minimum average delay, worst-case delay or number of packets delivered, the nodes needs to estimate delay distribution among nodes based on the sequence of packet forwarding and number of buffered packets in each node, which may lead to exponential computation complexity (to reduce complexity, exponential distribution is used to approximate the calculation in [17]). Spray and Wait [19] "sprays" a number of copies per packet into the network, and then "waits" until one of these nodes meets the destination. In this way it balances the tradeoff between the energy consumption incurred by flooding-based routing schemes and the delay incurred by spraying only one copy per packet in one transmission. The computation complexity of Spray and Wait is $\mathcal{O}(1)$ as it does randomly decide whether to forward packet without using any network information. MaxProp [18] prioritizes both the schedule of packets transmissions and the schedule of packets to be dropped, based on the path likelihoods to peers estimated from historical data and complementary mechanisms including acknowledgments, a head-start for new packets, and lists of previous intermediaries. The computation complexity of MaxProp is $\mathcal{O}(N_{node}^2)$ per node per contact, where N_{node} is the number of nodes. It is shown that MaxProp performs better than protocols that know the meeting schedule between peers. These terrestrial DTN solutions may not achieve the optimal performance underwater as the characteristics of underwater communications are not considered. Hence, in the rest of this section, we focus on related solutions for UW-ASNs.

As shown in Figs. 6 and 7, QUO VADIS outperforms RAPID, MaxProp, and Spray and Wait as these solutions transfer packets once the neighbors are in the transmitter's range. They perform well for a scenario where the connectivity is intermittent. However, the performance may not be optimal as in such scenario the link performance may be low. In contrast, by using the notion of external uncertainty, QUO VADIS predicts and waits for the best network configuration, where nodes move closer for the best communications; consequently, both e2e delivery ratio and link bit rate of QUO VADIS are the highest while its energy consumption is minimal. If the prediction is perfect, the performance (QUO VADIS–THEORY) has better performance than non-perfect scenarios. Note that among these 3 DTN solutions, RAPID performs the best: this is because RAPID prioritizes old packets so they will not be dropped. MaxProp gives priority to new packets; older, undelivered packets are dropped in the middle. Spray and Wait works in a similar way, i.e., it does not give priority to older packets. On the other hand, Spray and Wait is slightly better than MaxProp: this is because in our scenario the network connectivity is generally not disrupted.



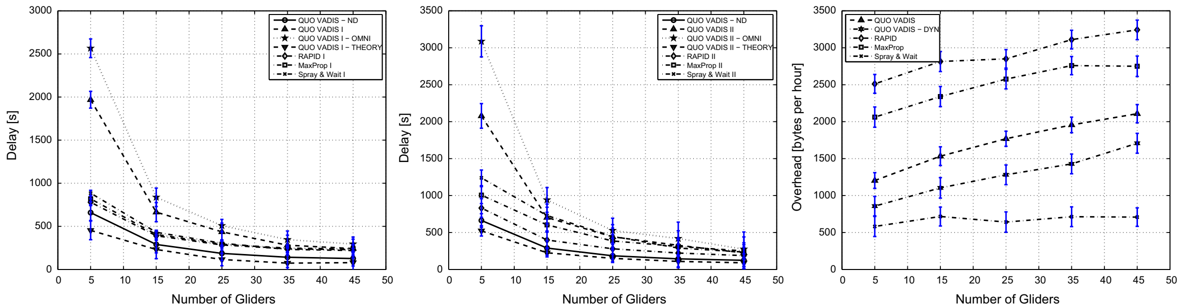
(a) Delivery ratio (b) Energy consumption (c) Link bit rate

Fig. 6. Impact on UW-ASNs: performance comparison for Class I traffic with DTN protocols.



(a) Delivery ratio (b) Energy consumption (c) Link bit rate

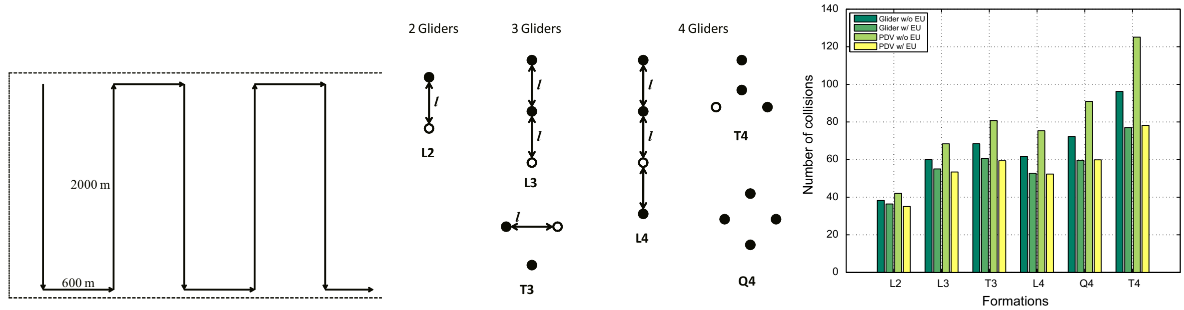
Fig. 7. Impact on UW-ASNs: performance comparison for Class II traffic with DTN protocols.



(a) Class I traffic: e2e delay (b) Class II traffic: e2e delay (c) Overhead per node

Fig. 8. Impact on UW-ASNs: comparison of e2e delay and overhead.

To see whether QUO VADIS can meet the delay requirement of the delay-tolerant traffic, we plot the e2e delays of these solutions. As shown in Fig. 8(a) and (b), QUO VADIS–ND gives the lowest e2e delay of the non-perfect cases. Compared to QUO VADIS and QUO VADIS–OMNI, QUO VADIS–ND does not wait for the vehicles to move to the optimal configuration, which results in more retransmissions. Still, as the vehicle speed is much lower than the underwater acoustic speed, QUO VADIS–ND needs much less time than QUO VADIS and QUO VADIS–OMNI even though more retransmissions are needed (thus resulting in higher communication delay). Similarly, the huge difference between vehicle and acoustic speed leads to the result that QUO VADIS and QUO VADIS–OMNI need more time than the DTN protocols, especially when the number of vehicles is small (i.e., where average inter-vehicle distance is large). On the other hand, by taking the position uncertainty into account, communications using QUO VADIS–ND are more reliable than those using RAPID, MaxProp or Spray and Wait, so lower e2e delay is incurred. QUO VADIS has lower delay than QUO VADIS–OMNI due to the improvement in communications by exploiting the directional transducer gain. Also, Class II traffic generally suffers from higher e2e delay than Class I due to the need for retransmissions. Finally, note that as the number of gliders increases the delays of QUO VADIS and QUO VADIS–OMNI drop quickly: this is because average inter-vehicle distances become smaller and the number of close neighbors



(a) Lawn-mower style to sample a region (b) Formation of AUVs (c) Number of AUV collisions

Fig. 9. Impact on underwater robotics: simulation scenario and results for different vehicle geometry formations.

increases, which reduces the need for a glider to wait a long time until a neighbor moves close.

To quantify the networking cost of all these solutions, in Fig. 8(c) we compare their overhead; even though QUO VADIS achieves the best network performance, its overhead is not the highest. The protocols with the higher overhead are in fact RAPID and MaxProp. In order to work, RAPID needs much control information – average size of past transfer opportunities, expected meeting times with nodes, list of packets delivered since last exchange, updated delivery delay estimate based on current buffer state, and information about other packets if modified since last exchange with the peer – which takes a large number of bytes. MaxProp needs to exchange a list of values including probabilities of meeting every other node on each contact, which basically corresponds to gathering global information achievable only through high neighbor discovery overhead. Compared to RAPID and MaxProp, QUO VADIS only needs to exchange the external-uncertainty information of itself and that of the destination node. The Spray and Wait protocol reduces transmission overhead by spreading only a few data packets to the neighbors. The source node then stops forwarding and lets each node carry a copy and perform direct transmissions. In our simulations, we selected the number of copies (sent to neighbors) to be one so to make the comparison fair and keep the overhead low. Last, but not least, we also implemented the dynamic adjustment of update intervals, which is denoted by 'QUO VADIS–DYN'. Compared with QUO VADIS, by using the external-uncertainty notion the proposed dynamic adjustment algorithm saves a significant amount of overhead. Note that here it is not necessary to differentiate the two classes of traffic as the overhead difference is small.

5.3. Impact on underwater robotics

Our approach brings appreciable benefits also to underwater robotics, where AUVs need to form a team in a specific formation, steer through the 3D region of interest, and take application-dependent measurements such as temperature and salinity. In [6], team formation and steering algorithms relying on underwater acoustic communications are proposed to enable glider swarming that is robust against ocean currents and acoustic channel impairments (e.g., high propagation and transmission delay, and low communication reliability). Given the number of AUVs to form the team and the formation geometry (which depends on the monitoring application), the gliders need to reach their positions in the specified formation; then, once the formation has been formed, they need to move through the region along a predefined trajectory while maintaining the formation. Algorithms for the AUVs to minimize the time to form the specified geometry formation and to steer the team through the 3D underwater area while keeping the team formation are proposed in [6] (Fig. 2(b)); without considering position uncertainty, vehicle collision is simply defined deterministically to occur when the inter-vehicle distance is below a predefined threshold. Conversely, using our external-uncertainty notion, we can improve the collision avoidance algorithms by adopting a probabilistic approach, i.e., we require the probability of two vehicles having a distance below a threshold be less than a given probability. This can make collision avoidance algorithms more robust in the underwater environment where position information is not accurate.

We implement the team formation and steering algorithms proposed in [6] for a simulated 10-h long mission where the team moves in a lawn-mower style to scan an area, as shown in Fig. 9(a). During the mission, the AUVs are required to form horizontal formations, as shown in Fig. 9(b), where 'L', 'T', and 'Q' denote 'Linear', 'Triangular', and 'Quadrangular' formations, respectively, and the number following these letters represents the number of AUVs in use. The closest inter-vehicle distance is set as 200 m. Two AUVs are defined to collide if the probability of their distance being within 10 m is greater than 0.1. We further assume that the displacement of the vehicles is incurred by random horizontal currents whose speeds follow the 2D Gaussian distribution $\mathcal{N}(\mathbf{0}, \sigma_0^2\mathbf{I_2})$, where $\sigma_0 = 0.05$ m/s and $\mathbf{I_2}$ is the 2×2 identity matrix. The number of collisions are counted and plotted in Fig. 9(c). We can see that the external-uncertainty-aware algorithms can effectively reduce the number of vehicle collisions, and the higher the number of vehicles used, the more collisions the external-uncertainty-aware algorithms can prevent. Here, 'Glider w/o EU', 'Glider w/ EU', 'PDV w/o EU', and 'PDV w/o EU' denote algorithms used for a team of gliders using the above external-uncertainty-aware constraints, a team of gliders not using the above external-uncertainty-aware constraints, a team of PDVs using the above external-uncertainty-aware constraints, and a team of PDVs not using the above external-uncertainty-aware constraints, respectively.

6. Conclusion and future work

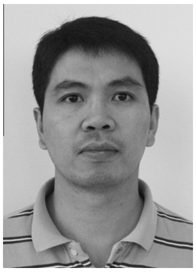
We defined two forms of position uncertainty for autonomous underwater vehicles – *internal* and *external* (depending on the view of the vehicles). We introduced a statistical model for internal-uncertainty estimation that works with any underwater localization scheme; based on this model, we designed an Unscented Kalman Filter to estimate the external uncertainty, and showed its effectiveness on several location-sensitive applications ranging from underwater acoustic communication and localization to distributed robotics and data processing/visualization. As future work, we plan to implement these solutions on underwater acoustic modems and field test them on a small fleet of autonomous vehicles performing adaptive sampling.

Acknowledgement

This work was supported by the NSF CAREER Award No. OCI-1054234.

References

[1] I.F. Akyildiz, D. Pompili, T. Melodia, Underwater acoustic sensor networks: research challenges, Ad Hoc Networks, vol. 3, Elsevier, 2005, pp. 257–279. no. 3.
[2] X. Xiang, Z. Zhou, X. Wang, Self-adaptive on demand geographic routing protocols for mobile ad hoc networks, in: Proc. of IEEE International Conference on Computer Communications (INFOCOM), Anchorage, AL, May 2007.
[3] B. Chen, D. Pompili, QUO VADIS: QoS-aware underwater optimization framework for inter-vehicle communication using acoustic directional transducers, in: Proc. of IEEE Communications Society Conference on Sensor, Mesh, and Ad Hoc Communications and Networks (SECON), Salt Lake City, UT, June 2011.
[4] M. Blain, S. Lemieux, R. Houde, Implementation of a ROV navigation system using acoustic/Doppler sensors and Kalman filtering, in: Proc. of IEEE International Conference on Engineering in the Ocean Environment (OCEANS), San Diego, CA, September 2003.
[5] S. Williams, I. Mahon, Simultaneous localisation and mapping on the Great Barrier Reef, in: Proc. of IEEE International Conference on Robotics and Automation (ICRA), New Orleans, LA, April 2004.
[6] B. Chen, D. Pompili, Team formation and steering algorithms for underwater gliders using acoustic communication, Computer Communications, vol. 35, Elsevier, 2012, pp. 1017–1028. no. 9.
[7] I.S. Kulkarni, D. Pompili, Task allocation for networked underwater vehicles in critical missions, IEEE J. Sel. Areas Commun. 28 (5) (2010) 716–727.
[8] J.C. Kinsey, R.M. Eustice, L.L. Whitcomb, A survey of underwater vehicle navigation: recent advances and new challenges, in: Proc. of IFAC Conference of Manoeuvering and Control of Marine Craft, Lisbon, Portugal, September 2006.
[9] G.A.F. Seber, C.J. Wild, Nonlinear Regression, John Wiley & Sons, 1989.
[10] G. Casella, R.L. Berger, Statistical Inference, second ed., Duxbury Press, 2001.
[11] P.S. Maybeck, Stochastic Models, Estimation, and Control, vol. 1, Academic Press, New York, 1979.
[12] R.V.D. Merwe, E.A. Wan, The square-root unscented Kalman filter for state and parameter-estimation, in: Proc. of International Conference on Acoustics, Speech, and Signal Processing (ICASSP), Salt Lake City, UT, May 2001.
[13] R.G. Brown, P.Y.C. Hwang, Introduction to Random Signals and Applied Kalman Filtering, third ed., John Wiley & Sons, Inc., Hoboken, NJ, 1996.
[14] S.M. Ross, Introduction to Probability Models, eighth ed., Academic Press, 2003.
[15] K.A. Morgansen, B.I. Triplett, D.J. Klein, Geometric methods for modeling and control of free-swimming fin-actuated underwater vehicles, IEEE Trans. Robot. 23 (6) (2007) 1184–1199.
[16] R.F. Stengel, Optimal Control and Estimation, Dover Publications, New York, 1994.
[17] A. Balasubramanian, B. Levine, A. Venkataramani, DTN routing as a resource allocation problem, in: Proc. of ACM Special Interest Group on Data Communication (SIGCOMM), August 2007.
[18] J. Burgess, B. Gallagher, D. Jensen, B.N. Levine, MaxProp: routing for vehicle-based disruption-tolerant networks, in: Proc. of Conference on Computer Communications (INFOCOM), Barcelona, Spain, April 2006.
[19] T. Spyropoulos, K. Psounis, C.S. Raghavendra, Spray and wait: an efficient routing scheme for intermittently connected mobile networks, in: Proc. of the ACM SIGCOMM Workshop on Delay-Tolerant Networking (WDTN), Philadelphia, PA, August 2005.
[20] K. Fall, A delay-tolerant network architecture for challenged internets, in: Proc. of ACM Special Interest Group on Data Communication (SIGCOMM), Karlsruhe, Germany, August 2003.
[21] S. Burleigh, A. Hooke, L. Torgerson, K. Fall, V. Cerf, B. Durst, K. Scott, H. Weiss, Delay-tolerant networking: an approach to interplanetary internet, IEEE Commun. Mag. 41 (6) (2003) 128C136.



Baozhi Chen received his B.S. degree from Beijing University of Posts and Telecommunications, Beijing, China, in 2000. He received his M.S. degree from Columbia University in New York City in 2003. In 2012, he received his Ph.D. degree in Electrical and Computer Engineering at Rutgers, the State University of New Jersey – New Brunswick, under the supervision of Prof. D. Pompili. His research interests lie in underwater communications and networking, communication and coordination of autonomous underwater vehicles, and wireless communications and networking, wireless sensor networks, and wireless body-area networks. He was a co-chair of work-in-progress poster session in ACM WUWNet'12. He also served as technical program committee member in BodyNets'10, GreenNets'll, and GreenNets'12, and as reviewer for ACM and IEEE journals and conferences.

Dario Pompili is an Assoc. Prof, in the Dept. of Electrical and Computer Engineering (ECE) at Rutgers University, where he is the director of the Cyber-Physical Systems Laboratory (CPS Lab) and the site co-director of the NSF Center for Cloud and Autonomic Computing (CAC). He received a Ph.D. in ECE from the Georgia Institute of Technology (GaTech) in 2007. In 2005, he was awarded GaTech BWN-Lab Researcher of the Year. He had previously received his 'Laurea' (integrated B.S./M.S.) and Doctorate degrees in Telecommunications and Systems Engineering from the University of Rome "La Sapienza", Italy, in 2001 and 2004, respectively. In 2011, Dr. Pompili received the NSF CAREER award to design efficient communication solutions for underwater multimedia applications and the Rutgers/ECE Outstanding Young Researcher award. In 2012, he received the ONR Young Investigator Program (YIP) award to develop an uncertainty-aware autonomic mobile computing grid framework as well as the DARPA Young Faculty Award (YFA) to enable complex real-time information processing based on compute-intensive models for operational neuroscience. He is a Senior Member of both the IEEE Communications Society and the ACM.