Kalman Filtering in Wireless Sensor Networks

REDUCING COMMUNICATION COST IN STATE-ESTIMATION PROBLEMS

ALEJANDRO RIBEIRO,
IOANNIS D. SCHIZAS,
STERGIOS I. ROUMELIOTIS,
and GEORGIOS B. GIANNAKIS

FABIO BUCCIARELLI

A wireless sensor network (WSN) is a collection of physically distributed sensing devices that can communicate through a shared wireless channel. Sensors can be deployed, for example, to detect the presence of a contaminant in a water reservoir, to estimate the temperature in an orange grove, or to track the position of a moving target.

The promise of WSNs stems from the benefits of distributed sensing and control. For example, in the target-tracking setup depicted in Figure 1, where sensors measure their distance to a target whose trajectory is to be estimated, the benefit of distributed sensing is the availability of observations with high signal-to-noise ratio (SNR). Whether collected by a passive radar, which estimates distances by the strength of an electromagnetic signature emitted by the target, or by an active radar, which gauges the reflection of a probing signal, measured signal strength decreases with increasing distance. Observation noise, however, remains unchanged because it is determined by circuit design and the operational environment. Consequently, in passive and active radar, the SNR of distance observations is inversely related to the distance being measured. In a conventional radar system a few expensive stations are deployed to cover a substantial area. Most of the time, the distance between the target and the sensors is large, and the observation SNR is low. Since the WSN comprises a large number of sensors, at each point in time a few sensors are close to the target, and thus measured distances are smaller. Although the circuitry of the sensors in the WSN is of lower quality than that of stations in a conventional radar system of comparable cost, the decrease in SNR due to the larger circuit noise power is more than offset by the smaller distances measured. Therefore, a WSN offers the potential to reduce localization error.

WSNs offer several advantages beyond those inherent to their distributed nature. Because sensors are independent hardware units, the likelihood of a large number of them

Digital Object Identifier 10.1109/MCS.2009.935569

1066-033X/10/$26.00©2010IEEE

failing simultaneously is small. Thus, WSNs have built-in redundancy, which can improve robustness relative to centralized processing. Redundancy also simplifies network deployment because optimizing sensor placements is not critical. Considering also the fact that it is not necessary to wire the sensors together, network deployment can be as simple as scattering the sensors over the area of interest. See [2]–[4] for discussions of additional advantages, issues, and applications of WSNs.

Although WSNs present attractive features, challenges associated with the scarcity of bandwidth and power in wireless communications have to be addressed. For the state-estimation problems discussed here, observations about a common state are collected by physically distributed terminals. To perform state estimation, sensors may share these observations with each other or communicate them to a fusion center for centralized processing. In either scenario, the communication cost in terms of bandwidth and power required to convey observations is large enough to merit attention. To explore this point, consider a vector state $\mathbf{x}(n) \in \mathbb{R}^p$ at time n and let the kth sensor collect observations $\mathbf{y}_k(n) \in \mathbb{R}^q$. The linear state and observation models are

$$\mathbf{x}(n) = \mathbf{A}(n)\mathbf{x}(n-1) + \mathbf{u}(n), \tag{1}$$
$$\mathbf{y}_k(n) = \mathbf{H}_k(n)\mathbf{x}(n) + \mathbf{v}_k(n), \tag{2}$$

where the driving noise vector $\mathbf{u}(n)$ is normal and uncorrelated across time with covariance matrix $\mathbf{C}_u(n)$, while the normal observation noise $\mathbf{v}_k(n)$ has covariance matrix $\mathbf{C}_v(n)$ and is uncorrelated across time and sensors.

With K vector observations $\{\mathbf{y}_k(n)\}_{k=1}^K$ available, the optimal mean squared error (MSE) estimation of the state $\mathbf{x}(n)$ for the linear model (1), (2) is accomplished by a Kalman filter. Brute force collection of these observations, however, incurs a communication cost commensurate with the product of the number K of sensors in the network, the number of scalar observations in the $\mathbf{y}_k(n)$ vectors, and the number of bits used to quantize each component of $\mathbf{y}_k(n)$.

The communication cost incurred by brute force collection of observations is not only large but unnecessary. It is possible to reduce the impact of the three factors mentioned above by exploiting information redundancy across observations $\mathbf{y}_{k_1}(n)$ and $\mathbf{y}_{k_2}(n)$ collected by different sensors, between different scalar observations composing the vector $\mathbf{y}_k(n)$ at a given sensor, and within each individual scalar observation. Indeed, because all sensors are observing the same state $\mathbf{x}(n)$, the measurements $\mathbf{y}_{k_1}(n)$ and $\mathbf{y}_{k_2}(n)$ are correlated. As a consequence of this correlation, it is possible for sensor k_1 to estimate the observation of sensor k_2 and use this estimate to reduce the cost of communicating its own observation to k_2. The correlation between individual components of the vector observation $\mathbf{y}_k(n)$ can be exploited to group scalar observations in a vector of reduced dimensionality. Finally, it is not necessary to finely quantize components of $\mathbf{y}_k(n)$ but only to the extent that further precision in the quantization of $\mathbf{y}_k(n)$ contributes to reducing the error in the estimation of the state $\mathbf{x}(n)$.

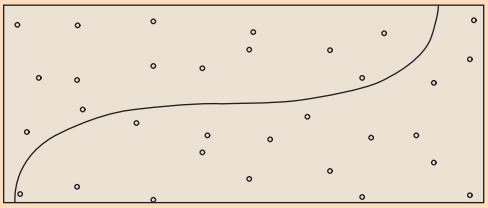

FIGURE 1 Target tracking with a wireless sensor network. Wireless sensor networks offer an inherent advantage in estimation problems due to distributed data collection. For a target-tracking application it is likely that some sensors, not necessarily the same over time, are always close to the target. Due to proximity, these sensors provide observations with a larger signal-to-noise ratio than observations that would be acquired by a single centralized sensor.

To reduce the cost of communicating the components of $\mathbf{y}_k(n)$, we discuss filters that estimate the state $\mathbf{x}(n)$ based on quantized representations of the original observations $\mathbf{y}_k(n)$ using a small number of bits, typically between one and three. Finely quantized versions of $\mathbf{y}_k(n)$ can be used in lieu of the nonquantized observations $\mathbf{y}_k$ in standard Kalman filters. This substitution is not possible with coarsely quantized versions, motivating the design of state estimators that incorporate the nonlinear quantization operation into the observation model. The challenge in this estimation problem is that the quantization operator is discontinuous. In principle, it is therefore necessary to resort to nonlinear state-estimation tools, such as the unscented Kalman filter [5] or the particle filter [6], resulting in prohibitive computational cost for WSN deployment. However, it turns out that despite the discontinuous observation model it is possible to build state-estimation algorithms whose structure and computational cost is similar to a standard Kalman filter. These algorithms are presented in the section "Quantized Kalman Filters."

We begin by considering quantization to a single bit by resorting to the transmission of the sign of the innovations sequence. Quantization to multiple bits is addressed through an iterative quantizer. Whereas coarse quantization to a few bits per observation increases the MSE of estimates relative to a Kalman filter using finely quantized observations, performance analysis of quantized Kalman filters shows that the increase in MSE is small. As we detail in the section "Quantized Kalman Filters," quantization to a single bit per observation increases the MSE by a factor of $\pi/2 \approx 1.57$ with respect to a standard Kalman filter, while quantization to 2 bits and 3 bits results in relative penalties of 1.15 and 1.05; see also [7] and [8]. Applications of quantized Kalman filters using 1 bit and 3 bits per observation are presented for a simulated target-tracking problem and an experimental multiple robot localization problem.

To exploit the correlation between observation data acquired at different sensors, recursive data-aggregation protocols are discussed. Brute force collection of $\mathbf{y}_k(n)$ observations incurs a large communication cost because, in addition to transmitting their own observations, sensors transmit observations received from other sensors in previous communications. Instead of forwarding separate observations, sensors forward linear combinations of their local information with messages received from neighboring sensors, which are also linear combinations formed at earlier times. State estimation in this context calls for the design of MSE optimal estimators for $\mathbf{x}(n)$ based on recursive linear combinations of data. To this end, the section "Consensus-Based Distributed Kalman Filtering and Smoothing" describes how to design the messages exchanged among sensors and the information-combining rules.

Finally, the correlation between components of $\mathbf{y}_k(n)$ allows sensors to reduce the dimensionality of their observation data $\mathbf{y}_k(n)$. The compression procedure is designed to trade off transmission cost as dictated by the reduced dimension and estimation accuracy as quantified by the MSE. Given a limited power budget available at each sensor and the fact that communication takes place over nonideal channel links, the goal is to design linear dimensionality-reducing operators that minimize the state-estimate MSE when operating over noisy channels. Two scenarios are considered, differentiated by whether state estimation takes place at a fusion center or at predetermined sensors in an ad hoc topology.

The reader interested on technical details can find a brief literature guide in "Further Reading."

QUANTIZED KALMAN FILTERS

To study quantized Kalman filters we initially focus on scalar observations $\{y_k(n)\}_{k=1}^K$, resulting in an observation model of the form $y_k(n) = \mathbf{h}_k^T(n)\mathbf{x}(n) + v_k(n)$ with noise variance $\sigma_v(n)$. We also assume that a scheduling algorithm is in place to decide which sensor is to transmit at time n. Therefore, with $k(n)$ denoting the scheduled sensor at time n, an efficient means of quantizing and transmitting $y(n) := y_{k(n)}(n)$ is sought. In more precise terms we study state-estimation problems when nonquantized amplitude observations $y(n)$ are mapped to messages $m(n)$ containing a small number of bits.

Quantization results in a subtle change in the state-estimation problem. Instead of seeking the minimum mean squared error (MMSE) estimator $\hat{\mathbf{x}}(n\,|\,\mathbf{y}_{0:n})$ based on past observations $\mathbf{y}_{0:n} := [y(0),\ldots,y(n)]^T$, the problem transmutes into that of finding the MMSE estimator $\hat{\mathbf{x}}(n\,|\,\mathbf{m}_{0:n})$ based on past messages $\mathbf{m}_{0:n} := [m(0),\ldots,m(n)]^T$. Although both estimators are given by the respective conditional means, the use of past observations yields a canonical linear-state estimation, whereas the use of past messages is a challenging nonlinear estimation. Since the computational cost of most nonlinear state-estimation

Further Reading

A review of research challenges associated with WSNs can be found in [3]. Comprehensive references dealing with various applications and research problems are included in [2] and [S1]. The sign of innovations Kalman filter is presented in [7]. From a more general point of view, the intermingling of quantization and estimation has a long history; early references include [S2] and [S3]. In the context of wireless sensor networks, the problem is revisited in [S4]–[S6]. An introduction to this topic can be found in [S7]. The iterative sign of innovations Kalman filter is developed in [S8]. More general quantization rules for Kalman filtering problems can be found in [8]. The distributed Kalman smoother state estimators can be found in [24], whereas alternative distributed implementations are available in [15]–[17] and [19]. Detailed treatment of distributed computation and estimation are given in [22], [24], and [29], and the references therein. The intertwining of dimensionality reduction with estimation and tracking is further developed in [12], [31], [S7], and [S8].

REFERENCES

[S1] S. Kumar, F. Zao, and D. Shepherd, Eds., *IEEE Signal Processing Mag. (Special Issue on Collaborative Information Processing)*, vol. 19, no. 2, Mar. 2002.

[S2] R. Curry, W. Vandervelde, and J. Potter, "Nonlinear estimation with quantized measurements—PCM, predictive quantization, and data compression," *IEEE Trans. Inform. Theory*, vol. 16, no. 2, pp. 152–161, Mar. 1970.

[S3] D. Williamson, "Finite wordlength design of digital Kalman filters for state estimation," *IEEE Trans. Automat. Contr.*, vol. 30, no. 10, pp. 930–939, Oct. 1985.

[S4] H. Papadopoulos, G. Wornell, and A. Oppenheim, "Sequential signal encoding from noisy measurements using quantizers with dynamic bias control," *IEEE Trans. Inform. Theory*, vol. 47, no. 3, pp. 978–1002, Mar. 2001.

[S5] A. Ribeiro and G. B. Giannakis, "Bandwidth-constrained distributed estimation for wireless sensor networks, Part I: Gaussian case," *IEEE Trans. Signal Processing*, vol. 54, no. 3, pp. 1131–1143, Mar. 2006.

[S6] A. Ribeiro and G. B. Giannakis, "Bandwidth-constrained distributed estimation for wireless sensor networks, Part II: Unknown pdf," *IEEE Trans. Signal Processing*, vol. 54, no. 7, pp. 2784–2796, July 2006.

[S7] J.-J. Xiao, A. Ribeiro, Z.-Q. Luo, and G. B. Giannakis, "Distributed compression-estimation using wireless sensor networks," *IEEE Signal Processing Mag.*, vol. 23, no. 4, pp. 27–41, July 2006.

[S8] I. D. Schizas, G. B. Giannakis, and Z. Q. Luo, "Distributed estimation using reduced-dimensionality sensor observations," *IEEE Trans. Signal Processing*, vol. 55, no. 8, pp. 4284–4299, Aug. 2007.

algorithms is excessive for WSN deployments, the goal is to find filters that can deal with quantization discontinuities while retaining the small computational requirements and memory footprints of conventional Kalman filters. These properties are present in the sign of innovations Kalman filter (SOI-KF) and its variants discussed in this section; see also [7].

Sign of Innovations Kalman Filter

In state-estimation problems, the innovations sequence is defined as the difference between the current observation and its prediction based on past observations. The intuition supporting this definition is that this difference contains the information that the current observation $y(n)$ has about the state $\mathbf{x}(n)$ that is not conveyed by previous observations $\mathbf{y}_{0:n-1}$. It is thus natural to define the predicted estimates as $\hat{y}(n\,|\,\mathbf{m}_{0:n-1}) := \mathbb{E}[y(n)\,|\,\mathbf{m}_{0:n-1}]$, the corresponding innovations sequence as $\tilde{y}(n\,|\,\mathbf{m}_{0:n-1}) := y(n) - \hat{y}(n\,|\,\mathbf{m}_{0:n-1})$, and the message $m(n)$ as a quantized version of $\tilde{y}(n\,|\,\mathbf{m}_{0:n-1})$. As a first approach, consider quantization to a single bit per observation and let messages exchanged consist of the sign of the innovations sequence, that is, $m(n) := \text{sign}[\tilde{y}(n|\mathbf{m}_{0:n-1})] = \text{sign}[y(n) - \hat{y}(n\,|\,\mathbf{m}_{0:n-1})]$. The sequence $m(n)$ indicates whether the observation $y(n)$ is larger or smaller than the prediction $\hat{y}(n\,|\,\mathbf{m}_{0:n-1})$ based on past messages $\mathbf{m}_{0:n-1}$.

The estimation task at hand is then to find the MMSE estimate $\hat{\mathbf{x}}(n\,|\,\mathbf{m}_{0:n})$ of the state $\mathbf{x}(n)$ given the current and past messages $\mathbf{m}_{0:n}$. The MMSE estimate is given by the conditional expectation $\mathbb{E}[\mathbf{x}(n)\,|\,\mathbf{m}_{0:n}]$, which in principle can be determined by computing the corresponding multidimensional integral of the state $\mathbf{x}(n)$ weighted by the conditional distribution $p[\mathbf{x}(n)\,|\,\mathbf{m}_{0:n}]$ of the state, given messages $\mathbf{m}_{0:n}$. Evaluating this integral, in turn, requires knowing the probability density function (pdf) $p[\mathbf{x}(n)\,|\,\mathbf{m}_{0:n}]$, which can be found using the prediction-correction algorithm described below.

The *prediction step* involves obtaining the prediction pdf $p[\mathbf{x}(n)\,|\,\mathbf{m}_{0:n-1}]$ from the correction pdf $p[\mathbf{x}(n-1)|\,\mathbf{m}_{0:n-1}]$. The state $\mathbf{x}(n)$ at time n is the sum of $\mathbf{A}(n)\mathbf{x}(n-1)$ and the independent input noise $\mathbf{u}(n)$. Therefore, to obtain the prediction pdf $p[\mathbf{x}(n)\,|\,\mathbf{m}_{0:n-1}]$, it suffices to propagate the correction pdf $p[\mathbf{x}(n-1)\,|\,\mathbf{m}_{0:n-1}]$ through the linear transformation $\mathbf{A}(n)$ and then convolve the result with the normal pdf $\mathcal{N}[\mathbf{u}(n);\,\mathbf{0},\,\mathbf{C}_u(n)]$ of the driving noise.

The *correction step* starts from the prediction pdf and computes the correction pdf $p[\mathbf{x}(n)\,|\,\mathbf{m}_{0:n}]$. This computation can be done by applying Bayes's rule to the random variables $\mathbf{x}(n)$ and $m(n)$ to obtain

$$p[\mathbf{x}(n)\,|\,\mathbf{m}_{0:n}] = p[\mathbf{x}(n)|\mathbf{m}_{0:n-1}]\frac{\Pr\{m(n)\,|\,\mathbf{x}(n),\,\mathbf{m}_{0:n-1}\}}{\Pr\{m(n)|\,\mathbf{m}_{0:n-1}\}}. \tag{3}$$

In spirit, these prediction-correction steps are not different from the corresponding ones in the Kalman filter. With linear state propagation, linear observations, normal driving inputs, and normal observation noise, the prediction pdfs $p[\mathbf{x}(n)\,|\,\mathbf{y}_{0:n-1}]$ and the correction pdfs $p[\mathbf{x}(n)\,|\,\mathbf{y}_{0:n-1}]$ are normal. As such, prediction and correction pdfs are completely characterized by their means and covariances, which are the quantities that the Kalman filter tracks. Thus, the prediction step in the Kalman filter can be interpreted as propagating the correction pdf $p[\mathbf{x}(n-1)\,|\,\mathbf{y}_{0:n-1}]$ of the previous step to the prediction pdf $p[\mathbf{x}(n)\,|\,\mathbf{y}_{0:n-1}]$ through convolution. Likewise, the correction step uses Bayes's rule to obtain the correction pdf $p[\mathbf{x}(n)\,|\,\mathbf{y}_{0:n}]$ from the prediction pdf $p[\mathbf{x}(n)\,|\,\mathbf{y}_{0:n-1}]$.

Because quantization is a nonlinear operation, the probability distributions $p[\mathbf{x}(n)\,|\,\mathbf{m}_{0:n-1}]$ and $p[\mathbf{x}(n)\,|\,\mathbf{m}_{0:n}]$ necessary to find $\hat{\mathbf{x}}(n\,|\,\mathbf{m}_{0:n})$ are not normal. Consequently, it is not sufficient to track their first two moments, and the prediction-correction becomes computationally costly. An alternative approximation in nonlinear filtering (see [9]) is to model the prediction pdf $p[\mathbf{x}(n)\,|\,\mathbf{m}_{0:n-1}]$ as normal so that, at least for the prediction step, only the mean and covariance must be propagated as performed by (see also Figure 2)

$$\hat{\mathbf{x}}(n|\mathbf{m}_{0:n-1}) = \mathbf{A}(n)\hat{\mathbf{x}}(n-1\,|\,\mathbf{m}_{0:n-1}), \tag{4}$$

$$\mathbf{M}(n|\mathbf{m}_{0:n-1}) = \mathbf{A}(n)\mathbf{M}(n-1|\mathbf{m}_{0:n-1})\mathbf{A}^T(n) + \mathbf{C}_u(n). \tag{5}$$

Even with this simplifying approximation, $p[\mathbf{x}(n)\,|\,\mathbf{m}_{0:n}]$ is not normal. Indeed, the probability $\Pr\{m(n)\,|\,\mathbf{x}(n),\,\mathbf{m}_{0:n-1}\}$ of observing $m(n)$ given the state $\mathbf{x}(n)$ and past observations $\mathbf{m}_{0:n-1}$ can be rewritten as $\Pr\{m(n)\,|\,\mathbf{x}(n)\}$ because conditioning on past messages given the present state is redundant. Furthermore, $m(n) = 1$ is equivalent to $y(n) - \hat{y}(n\,|\,\mathbf{m}_{0:n-1}) \geq 0$, which, using the observation model in (2), yields $\mathbf{h}^T(n)\mathbf{x}(n) - \hat{y}(n\,|\,\mathbf{m}_{0:n-1}) \geq v(n)$. Similarly, $m(n) = -1$ is equivalent to $y(n) - \hat{y}(n\,|\,\mathbf{m}_{0:n-1}) < 0$ and, from the observation model, to $\mathbf{h}^T(n)\mathbf{x}(n) - \hat{y}(n\,|\,\mathbf{m}_{0:n-1}) < v(n)$. Given that the observation noise $v(n)$ is normal, the probability of $v(n)$ being larger or smaller than $\mathbf{h}^T(n)\mathbf{x}(n) - \hat{y}(n\,|\,\mathbf{m}_{0:n-1})$ can be expressed in terms of the normal cumulative distribution function. Comparing these comments with Bayes's rule (3), we deduce that $p[\mathbf{x}(n)\,|\,\mathbf{m}_{0:n-1}]$ is the product of the normal pdf $p[\mathbf{x}(n)\,|\,\mathbf{m}_{0:n-1}]$ and the normal cumulative distribution $\Pr\{m(n)\,|\,\mathbf{x}(n),\,\mathbf{m}_{0:n-1}\}$. The remaining term $\Pr\{m(n)|\,\mathbf{m}_{0:n-1}\}$ is a normalizing constant.

While the correction pdf in (3) is not normal, the MMSE estimate is nonetheless obtained as the solution of the expected value integral, which could be evaluated numerically. It is noteworthy, however, that a closed-form expression for this integral exists and leads to the correction step [7]

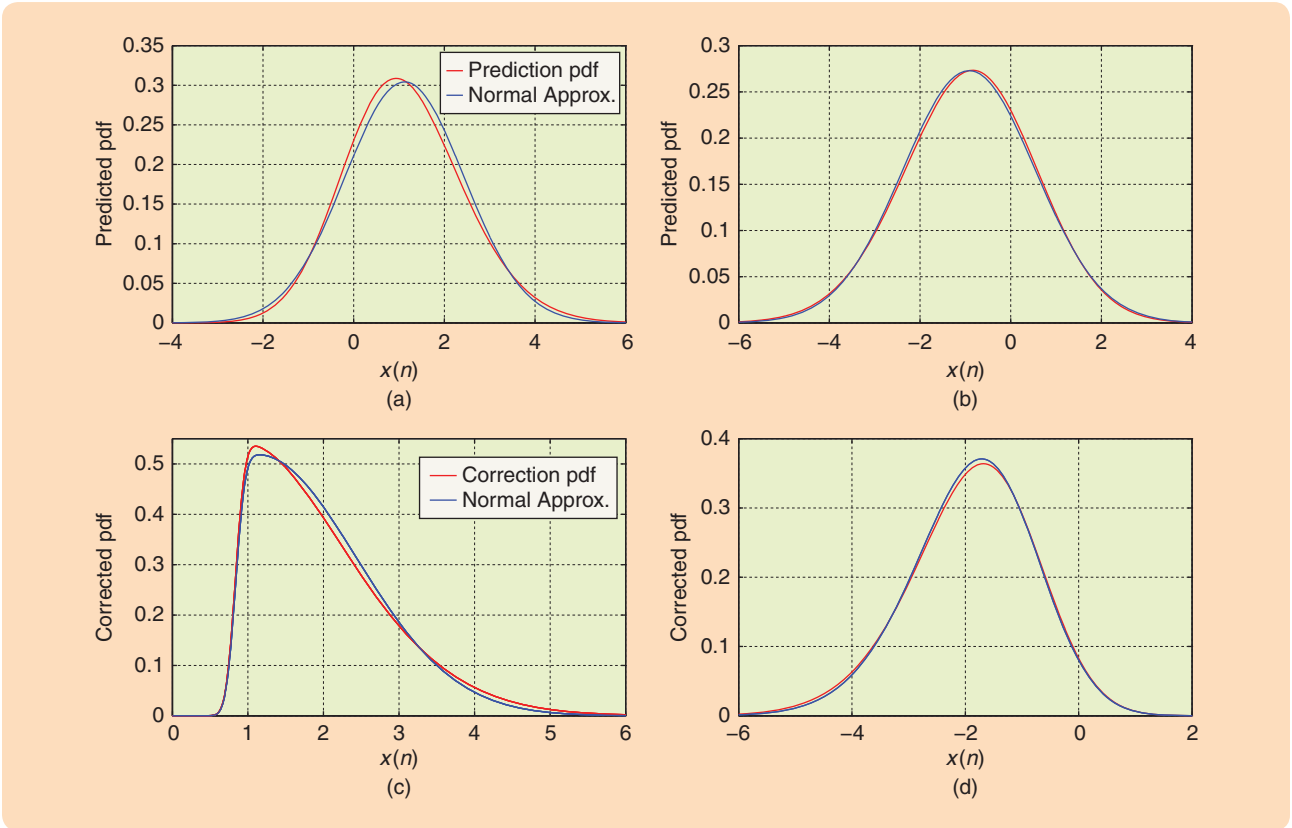

FIGURE 2 Normal approximation in the sign of innovations Kalman filter (SOI-KF). The prediction probability distribution function (pdf) $p[\mathbf{x}(n)|\mathbf{m}_{0:n-1}]$ for a scalar state model $\mathbf{x}(n) = x(n)$ and the normal approximation $\mathcal{N}[\mathbf{x}(n); \hat{\mathbf{x}}(n|\mathbf{m}_{0:n-1}), \mathbf{M}(n|\mathbf{m}_{0:n-1})]$ used to derive the SOI-KF are compared in (a) and (b). The signal to noise ratio $\text{SNR} := h^2(n)\mathbb{E}[x^2(n)]/\sigma_v^2(n)$ of the state-observation model in (a) is SNR = 10 dB, whereas in (b) SNR = 0 dB. The approximation works best for the small SNR = 0 dB in (b) but it is accurate even for the high SNR = 10 dB in (a). The comparison in (c) and (d) is between the actual correction pdf $p[\mathbf{x}(n)|\mathbf{m}_{0:n}] \propto p[\mathbf{x}(n)|\mathbf{m}_{0:n-1}]$ $Pr\{m(n)|\mathbf{x}(n), \mathbf{m}_{0:n-1}\}$ and the approximation $\tilde{p}[\mathbf{x}(n)|\mathbf{m}_{0:n}] \propto \mathcal{N}[\mathbf{x}(n); \hat{\mathbf{x}}(n|\mathbf{m}_{0:n-1}), \mathbf{M}(n|\mathbf{m}_{0:n-1})]Pr\{m(n)|\mathbf{x}(n), \mathbf{m}_{0:n-1}\}$. The comparison in (c) is for SNR = 10 dB, whereas in (d) is for SNR = 0 dB. Inspection of (c) and (d) reveals that the first moment of the approximated pdf $\tilde{p}[\mathbf{x}(n)|\mathbf{m}_{0:n}]$ is similar to the first moment of $p[\mathbf{x}(n)|\mathbf{m}_{0:n}]$. Approximating $\hat{\mathbf{x}}(n|\mathbf{m}_{0:n})$ as the first moment of $\tilde{p}[\mathbf{x}(n)|\mathbf{m}_{0:n}]$ is thus justified.

$$\hat{\mathbf{x}}(n|\mathbf{m}_{0:n}) = \hat{\mathbf{x}}(n|\mathbf{m}_{0:n-1}) + \frac{(\sqrt{2/\pi})\mathbf{M}(n|\mathbf{m}_{0:n-1})\mathbf{h}(n)}{\sqrt{\mathbf{h}^T(n)\mathbf{M}(n|\mathbf{m}_{0:n-1})\mathbf{h}(n) + \sigma_v^2(n)}} m(n), \tag{6}$$

$$\mathbf{M}(n|\mathbf{m}_{0:n}) = \mathbf{M}(n|\mathbf{m}_{0:n-1}) - \frac{(2/\pi)\mathbf{M}(n|\mathbf{m}_{0:n-1})\mathbf{h}(n)\mathbf{h}^T(n)\mathbf{M}(n|\mathbf{m}_{0:n-1})}{\mathbf{h}^T(n)\mathbf{M}(n|\mathbf{m}_{0:n-1})\mathbf{h}(n) + \sigma_v^2(n)}. \tag{7}$$

The SOI-KF, which amounts to a recursive application of (4), (5) and (6), (7), is similar to the Kalman filter in that it requires only a few algebraic operations per iteration. Moreover, comparison of the SOI-KF covariance correction equation with the corresponding covariance correction for the standard Kalman filter based on the innovations reveals that they are identical except for the factor $2/\pi$.

The similarity between the covariance updates of the Kalman filter and the SOI-KF allows for a simple performance comparison. The variance of the state estimates increases with each prediction step and decreases with each correction step. Starting with the same covariance matrix $\mathbf{M}(n-1|\mathbf{m}_{0:n-1}) = \mathbf{M}(n-1|\mathbf{y}_{0:n-1})$ at time $n-1$, a Kalman filter and an SOI-KF have identical predicted covariance matrices, that is, $\mathbf{M}(n|\mathbf{m}_{0:n-1}) = \mathbf{M}(n|\mathbf{y}_{0:n-1})$, at time n. To compare the corrected variances of the Kalman filter and the SOI-KF, it is informative to examine the per-step covariance reductions. For the Kalman filter, the per-step covariance reduction is defined as $\Delta\mathbf{M}^{\text{KF}}(n) := \mathbf{M}(n-1|\ \mathbf{y}_{0:n-1}) - \mathbf{M}(n|\mathbf{y}_{0:n})$, while, for the SOI-KF, it is defined as $\Delta\mathbf{M}(n) := \mathbf{M}(n-1|\mathbf{m}_{0:n-1}) - \mathbf{M}(n|\mathbf{m}_{0:n})$. It is not difficult to recognize that these reductions are related by the factor $2/\pi$, that is, $\Delta\mathbf{M}(n) = (2/\pi)\Delta\mathbf{M}^{\text{KF}}(n)$. Using the sign of innovations $m(n)$ thus entails a penalty of $1 - 2/\pi = 36\%$ relative to the variance reduction afforded by the actual innovations $y(n)$. This penalty is

modest for the use of a coarse quantization rule of one bit per scalar observation.

On the other hand, while $2/\pi$ relates the per-step covariance reductions, these reductions accumulate over time and eventually could cause considerable loss in SOI-KF performance relative to the Kalman filter. Therefore, while the relative penalty of using the sign of the innovations in lieu of the actual innovation is small, the absolute penalty could be considerable. These limitations motivate consideration of finer multibit quantization rules, which are discussed below.

Algorithmic Implementation

Algorithms shown in tables 1 and 2 delineate implementation of the SOI-KF in a WSN. The observation-transmission algorithm in Table 1 is run by only one sensor at a time, as dictated by the scheduling algorithm. The goal of this algorithm is to compute and transmit the sign of innovations $m(n)$. The scheduled sensor $S_{k(n)}$ uses its observation $y(n) = y_{k(n)}(n)$ to form the predicted estimates $\hat{\mathbf{x}}(n\,|\,\mathbf{m}_{0:n-1})$ for the state and $\hat{y}(n\,|\,\mathbf{m}_{0:n-1})$ for the observation. The sign of innovations sequence $m(n)$ is computed as the sign of the difference between the observation and its predicted estimate and then broadcast to all other sensors. The objective of the reception-estimation algorithm in Table 2, which is run continually by all sensors, is to estimate the state $\mathbf{x}(n)$ using all received messages $\mathbf{m}_{0:n}$. To this end, sensors use prediction-correction equations similar to the expressions used by the Kalman filter. Therefore, at each time slot the state prediction $\hat{\mathbf{x}}(n\,|\,\mathbf{m}_{0:n-1})$ and associated covariance matrix $\mathbf{M}(n\,|\,\mathbf{m}_{0:n-1})$ are computed. After a sensor receives the sign of innovations message $m(n)$, the corrected estimate $\hat{\mathbf{x}}(n\,|\,\mathbf{m}_{0:n})$ and corresponding covariance matrix $\mathbf{M}(n\,|\,\mathbf{m}_{0:n})$ are obtained. Except for minor differences in the correction equations, this algorithm is identical to the Kalman filter.

Computational and memory requirements of the algorithms in tables 1 and 2 are affordable for low-cost sensors. Storing state estimates $\hat{\mathbf{x}}(n\,|\,\mathbf{m}_{0:n-1})$ and $\hat{\mathbf{x}}(n\,|\,\mathbf{m}_{0:n})$ and their respective covariance matrices $\mathbf{M}(n\,|\,\mathbf{m}_{0:n-1})$ and $\mathbf{M}(n\,|\,\mathbf{m}_{0:n})$ requires $p^2 + p$ memory elements, where p is the number of elements in the state vector $\mathbf{x}(t)$. Memory is also required to store the system model, that is, $\mathbf{A}(n)$, $\mathbf{C}_u(n)$, $\mathbf{h}(n)$, and $\sigma_{v(n)}$, which requires on the order of p^2

TABLE 1 Sign of innovations Kalman filter (SOI-KF) observation-transmission algorithm. The SOI-KF observation-transmission algorithm is run by the scheduled sensor $S_{k(n)}$ to collect the observation $y(n) = y_{k(n)}(n)$ and compute and broadcast the message $m(n)$. The observation prediction $\hat{y}(n|\mathbf{m}_{0:n-1})$ is computed using linear transformations of the previous state estimate $\hat{\mathbf{x}}(n-1|\mathbf{m}_{0:n-1})$. The message $m(n)$ is the sign of the innovation $\tilde{y}(n|\mathbf{m}_{0:n-1}) := y(n) - \hat{y}(n|\mathbf{m}_{0:n-1})$.

Algorithm 1-A SOI-KF – Observation and transmission

Require: $\hat{\mathbf{x}}(n-1|\mathbf{m}_{0:n-1})$
Ensure: $m(n)$
1: Collect observation $y(n) = y_{k(n)}(n)$
2: Compute state prediction $\hat{\mathbf{x}}(n|\mathbf{m}_{0:n-1}) = \mathbf{A}(n)\hat{\mathbf{x}}(n-1|\mathbf{m}_{0:n-1})$
3: Compute observation prediction $\hat{y}(n|\mathbf{m}_{0:n-1}) = \mathbf{h}^T(n)\hat{\mathbf{x}}(n|\mathbf{m}_{0:n-1})$
4: Construct binary observation $m(n) = \text{sign}[y(n) - \hat{y}(n|\mathbf{m}_{0:n-1})]$
5: Transmit $m(n)$

TABLE 2 Sign of innovations Kalman filter (SOI-KF) reception-estimation algorithm. The SOI-KF reception-estimation algorithm is run continually by all sensors to compute state estimates $\hat{\mathbf{x}}(n|\mathbf{m}_{0:n})$. In the prediction step (Step 2), linear transformations of the previous state estimate $\hat{\mathbf{x}}(n-1|\mathbf{m}_{0:n-1})$ and the covariance matrices $\mathbf{M}(n-1|\mathbf{m}_{0:n-1})$ yield the predicted estimate $\hat{\mathbf{x}}(n|\mathbf{m}_{0:n-1})$ and its corresponding covariance matrix $\mathbf{M}(n|\mathbf{m}_{0:n-1})$. The information contained in the sign of innovations message $m(n)$ is incorporated in the correction step (Step 4). The correction step is similar to the conventional Kalman filter. The covariance matrix update, in particular, is identical except for the factor $2/\pi$.

Algorithm 1-B SOI-KF – Reception and estimation

Require: prior estimate $\hat{\mathbf{x}}(-1|-1)$ and covariance matrix $\mathbf{M}(-1|-1)$
1: **for** $n = 0$ to ∞ **do** {repeat for the life of the network}
2: Compute predicted estimate $\hat{\mathbf{x}}(n|\mathbf{m}_{0:n-1})$ and covariance matrix $\mathbf{M}(n|\mathbf{m}_{0:n-1})$

$$\hat{\mathbf{x}}(n|\mathbf{m}_{0:n-1}) = \mathbf{A}(n)\hat{\mathbf{x}}(n-1|\mathbf{m}_{0:n-1})$$
$$\mathbf{M}(n|\mathbf{m}_{0:n-1}) = \mathbf{A}(n)\mathbf{M}(n-1|\mathbf{m}_{0:n-1})\mathbf{A}^T(n) + \mathbf{C}_u(n)$$

3: Receive binary observation $m(n)$
4: Compute corrected estimate $\hat{\mathbf{x}}(n|\mathbf{m}_{0:n})$ and covariance matrix $\mathbf{M}(n|\mathbf{m}_{0:n})$

$$\hat{\mathbf{x}}(n|\mathbf{m}_{0:n}) = \hat{\mathbf{x}}(n|\mathbf{m}_{0:n-1}) + \frac{(\sqrt{2/\pi})\mathbf{M}(n|\mathbf{m}_{0:n-1})\mathbf{h}(n)}{\sqrt{\mathbf{h}^T(n)\mathbf{M}(n|\mathbf{m}_{0:n-1})\mathbf{h}(n) + \sigma_v^2(n)}}m(n)$$

$$\mathbf{M}(n|\mathbf{m}_{0:n}) = \mathbf{M}(n|\mathbf{m}_{0:n-1}) - \frac{(2/\pi)\mathbf{M}(n|\mathbf{m}_{0:n-1})\mathbf{h}(n)\mathbf{h}^T(n)\mathbf{M}(n|\mathbf{m}_{0:n-1})}{\mathbf{h}^T(n)\mathbf{M}(n|\mathbf{m}_{0:n-1})\mathbf{h}(n) + \sigma_v^2(n)}$$

5: **end for**

memory elements. Prediction equations (4) and (5) require p^2 and p^3 flops, respectively. Correction equations (6) and (7) necessitate on the order of p^2 flops and, in the case of (6), the computation of a scalar square root. The latter operation is the only one that is not shared with a Kalman filter based on the actual innovations $\tilde{y}(n\,|\,\mathbf{m}_{0:n-1})$.

Because the scheduled sensor $S_{k(n)}$ also runs the reception-estimation algorithm in Table 2, it does not take full advantage of the information that $y_{k(n)}(n)$ contains about the state $\mathbf{x}(n)$. In principle, sensors not scheduled at time n could improve performance of their estimates by combining their own observations $y_k(n)$. Local information is left out from the algorithms in tables 1 and 2 because the goal of the SOI-KF is to obtain a synchronized estimate $\hat{\mathbf{x}}(n\,|\,\mathbf{m}_{0:n})$ across all sensors.

Filter Implementation

While the MSE updates of the Kalman filter and its quantized version in (7) are similar, the update of the state estimates has a different form. As it turns out, it is possible to express the state update in (6) in a form that exemplifies its link with the Kalman filter update. By replacing the innovation $\tilde{y}(n\,|\,\mathbf{m}_{0:n-1})$ with its sign $m(n)$, the units of the observations are lost. To recover these units, let $\sigma_{\tilde{y}}(n\,|\,\mathbf{m}_{0:n-1}) \coloneqq \sqrt{\mathbb{E}[\tilde{y}^2(n\,|\,\mathbf{m}_{0:n-1})]}$ denote the standard deviation of the innovations sequence and define the SOI-KF innovation as $\tilde{m}(n\,|\,\mathbf{m}_{0:n-1}) \coloneqq \sigma_{\tilde{y}}(n\,|\,\mathbf{m}_{0:n-1})m(n)$. The innovation sequence has zero-mean and its variance is given by the denominator of the MSE update in (7). According to this definition, the units of the SOI-KF innovation $\tilde{m}(n\,|\,\mathbf{m}_{0:n-1})$ are those of $\tilde{y}(n\,|\,\mathbf{m}_{0:n-1})$, and their average energies are the same, that is, $\mathbb{E}[\tilde{m}^2(n\,|\,\mathbf{m}_{0:n-1})] = \mathbb{E}[\tilde{y}^2(n\,|\,\mathbf{m}_{0:n-1})]$.

Recalling the definition of the Kalman gain and replacing $m(n)$ with $\tilde{m}(n\,|\,\mathbf{m}_{0:n-1})$ in (6), the SOI-KF takes a form more reminiscent of the Kalman filter

$$\mathbf{k}(n) \coloneqq \frac{\mathbf{M}(n\,|\,\mathbf{m}_{0:n-1})\mathbf{h}(n)}{\mathbf{h}^T(n)\mathbf{M}(n\,|\,\mathbf{m}_{0:n-1})\mathbf{h}(n) + \sigma_v^2(n)}, \tag{8}$$

$$\hat{\mathbf{x}}(n\,|\,\mathbf{m}_{0:n}) = \hat{\mathbf{x}}(n\,|\,\mathbf{m}_{0:n-1}) + (\sqrt{2/\pi})\mathbf{k}(n)\tilde{m}(n\,|\,\mathbf{m}_{0:n-1}), \tag{9}$$

$$\mathbf{M}(n\,|\,\mathbf{m}_{0:n}) = [\mathbf{I} - (2/\pi)\mathbf{k}(n)\mathbf{h}^T(n)]\,\mathbf{M}(n\,|\,\mathbf{m}_{0:n-1}). \tag{10}$$

The gain $\mathbf{k}(n)$ in (8) has the same functional expression as the gain used in the Kalman filter, while the MSE updates are identical except for the factor $2/\pi$. The state updates differ only in the factor $\sqrt{2/\pi}$ and in the replacement of

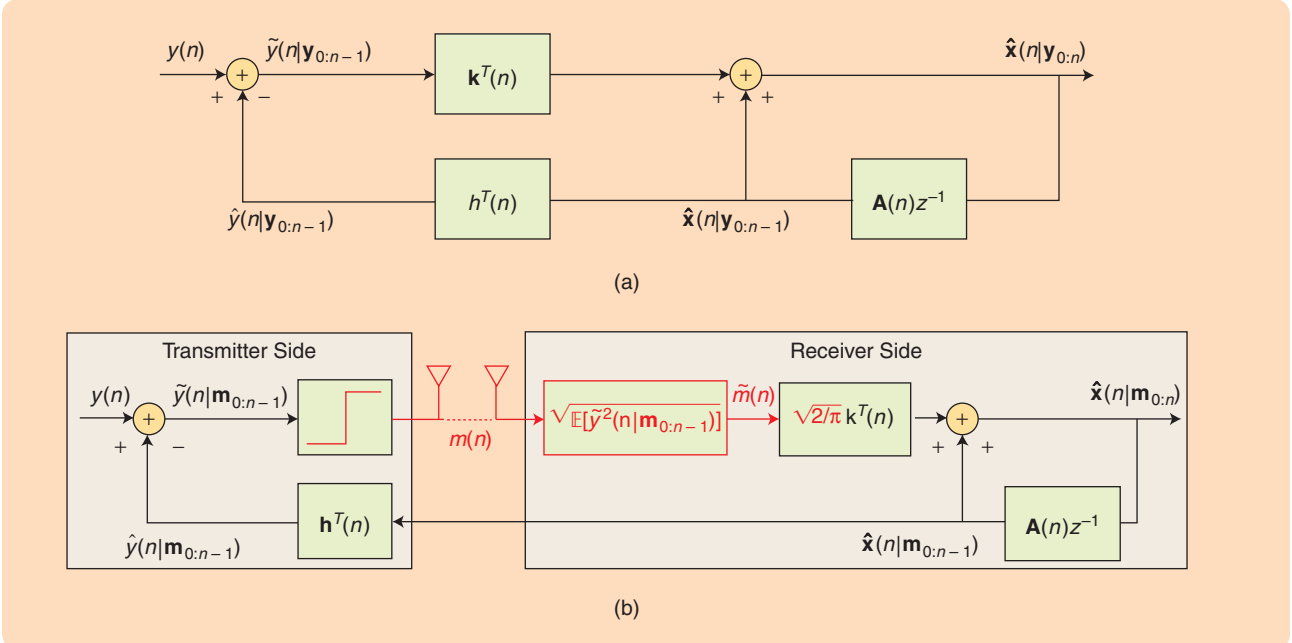

FIGURE 3 Block diagram of the sign of innovations Kalman filter (SOI-KF) compared with the standard Kalman filter. The Kalman filter (a) contains a feedback loop to compute state and observation predictions as linear transformations of the state estimate for the previous time slot. The observation prediction is subtracted from the observation to form the innovation. The innovation is then multiplied by the Kalman gain and added to the state prediction to form the corrected estimate. Likewise, the SOI-KF (b) has a feedback loop that starts with a delayed copy of the corrected estimate $\hat{\mathbf{x}}(n-1\,|\,\mathbf{m}_{0:n-1})$ to compute the state $\hat{\mathbf{x}}(n|\mathbf{m}_{0:n-1})$ and observation $\hat{y}(n|\mathbf{m}_{0:n-1})$ predictions, as well as the innovation $\tilde{y}(n\,|\,\mathbf{m}_{0:n-1})$. The highlighted differences with the Kalman filter include the hard limiter used to obtain the sign message *m(n);* the transmission-reception stage; the computation of the SOI-KF innovation $\tilde{m}(n) \coloneqq \tilde{m}(n\,|\,\mathbf{m}_{0:n-1})$; and the use of a scaled Kalman gain before addition to the predicted estimate. The scheduled sensor also utilizes $m(n)$ to compute the corrected estimate as signified by the dotted line in the transmission-reception stage.

the innovation $\tilde{y}(n\,|\,\mathbf{m}_{0:n-1})$ by the SOI-KF innovation $\tilde{m}(n\,|\,\mathbf{m}_{0:n-1})$. As the first- and second-order moments of $\tilde{y}(n\,|\,\mathbf{m}_{0:n-1})$ and $\tilde{m}(n\,|\,\mathbf{m}_{0:n-1})$ are identical, the factor $\sqrt{2/\pi}$ appearing in the state update explains the factor $2/\pi$ in the MSE update. The difference between the SOI-KF correction and the Kalman filter correction is that in the SOI-KF the magnitude of the correction at each step is determined by the magnitude of $\sigma_{\tilde{y}}(n\,|\,\mathbf{m}_{0:n-1})$, which is the same regardless of how large or small the actual innovation $\tilde{y}(n\,|\,\mathbf{m}_{0:n-1})$ is.

Expressing the correction step as in (8), (10) simplifies the comparison between the block diagrams of the Kalman filter and the SOI-KF. The block diagram for the Kalman filter in Figure 3 includes the feedback loop on the right that starts with a delayed copy of the corrected estimate $\hat{\mathbf{x}}(n-1\,|\,\mathbf{y}_{0:n-1})$ and computes the predicted estimate $\hat{\mathbf{x}}(n\,|\,\mathbf{y}_{0:n-1})$ along with the observation prediction $\hat{y}(n\,|\,\mathbf{y}_{0:n-1})$. The observation prediction is then subtracted from the observation $y(n)$ to compute the innovation $\tilde{y}(n\,|\,\mathbf{y}_{0:n-1})$. The innovation is then multiplied by the Kalman gain $\mathbf{k}(n)$ and added to the predicted estimate to yield the corrected estimate $\hat{\mathbf{x}}(n\,|\,\mathbf{y}_{0:n})$. The block diagram for the SOI-KF in Figure 3 contains the same feedback loop that starts with a delayed copy of the corrected estimate $\hat{\mathbf{x}}(n-1\,|\,\mathbf{m}_{0:n-1})$ to compute state $\hat{\mathbf{x}}(n\,|\,\mathbf{m}_{0:n-1})$ and observation $\hat{y}(n\,|\,\mathbf{m}_{0:n-1})$ predictions as well as the innovation $\tilde{y}(n\,|\,\mathbf{m}_{0:n-1})$. The SOI-KF passes the innovation through a hard limiter to obtain the sign message $m(n)$, which is then broadcast to other sensors. Upon reception, the message $m(n)$ is multiplied by the innovation's variance to yield the SOI-KF innovation $\tilde{m}(n\,|\,\mathbf{m}_{0:n-1})$. The innovation is then multiplied by a scaled Kalman gain and added to the predicted estimate to yield the corrected estimate $\hat{\mathbf{x}}(n\,|\,\mathbf{m}_{0:n})$. The scheduled sensor also utilizes $m(n)$ to compute the corrected estimate as signified by the dotted line in the transmission-reception stage.

Target Tracking with Sign of Innovations Kalman Filter

Target tracking based on distance-only measurements is a typical problem in bandwidth-constrained distributed estimation using WSNs [6], [10] for which an extended SOI-KF to nonlinear models appears to be attractive. Consider K sensors randomly and uniformly deployed in a square region of $L \times L$ m^2 and suppose that the sensor positions $\{\mathbf{p}^k\}_{k=1}^K$ are known.

The WSN is deployed to track the position $\mathbf{p}(n) := [p_1(n), p_2(n)]^T$ of a target, whose state model accounts for position $\mathbf{p}(n)$ and velocity $\mathbf{s}(n) := [s_1(n), s_2(n)]^T$ but not for acceleration, which is captured by the system noise. Under these assumptions, the state equation for tracking is [11]

$$\mathbf{p}(n) = \mathbf{p}(n-1) + T_s\mathbf{s}(n-1) + (T_s^2/2)\mathbf{u}(n), \quad (11)$$
$$\mathbf{s}(n) = \mathbf{s}(n-1) + T_s\mathbf{u}(n), \quad (12)$$

where T_s is the sampling period and the random vector $\mathbf{u}(n)$ is zero-mean white normal; that is, $p(\mathbf{u}(n)) = \mathcal{N}(\mathbf{u}(n); \mathbf{0}, \sigma_u^2\mathbf{I})$. Sensors gather information about their distance to the target by measuring the received power of a pilot signal following the path-loss model $y_k(n) = \alpha\log\|\mathbf{p}(n) - \mathbf{p}^k\| + v(n)$, with constant $\alpha \geq 2$, $\|\mathbf{p}(n) - \mathbf{p}^k\|$ denoting the distance between the target and the kth sensor, and $v(n)$ the observation noise with pdf

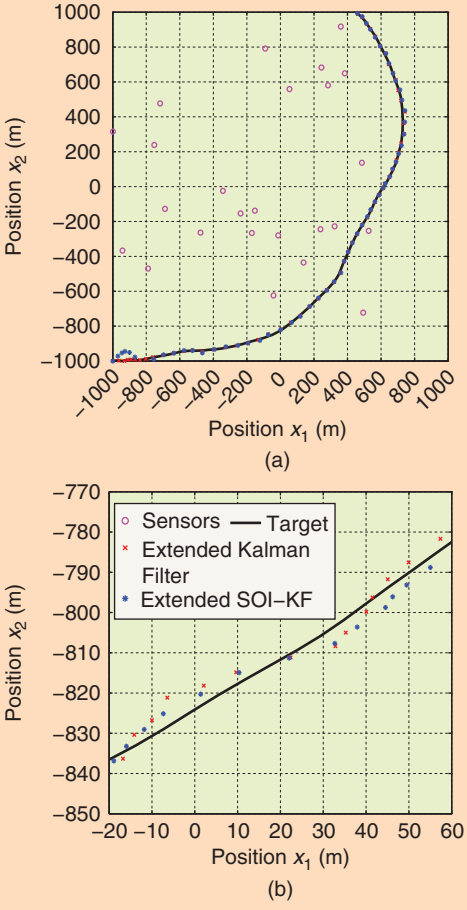

FIGURE 4 Target-tracking with the extended sign of innovations Kalman filter (SOI-KF). The entire trajectory is shown in (a), while a detail is shown in (b). The target trajectory is displayed along with extended Kalman filter estimates computed using the nonquantized amplitude observations $y(n)$ and extended SOI-KF estimates computed using the sign of innovations messages $m(n)$. The extended Kalman filter and extended SOI-KF estimates are indistinguishable. The target moves according to the zero-acceleration model in (11), (12). Randomly deployed sensors provide distance-only observations of the form $y_k(n) = \alpha\log\|\mathbf{p}(n) - \mathbf{p}^k\| + v(n)$, where $\mathbf{p}(n)$ is the target's position, $\mathbf{p}^k$ is the position of the kth sensor, and $v(n)$ is white Gaussian noise with variance σ_v. The scheduling algorithm works in cycles. At the beginning of each cycle, the sensor closest to the predicted estimate of the target's position $\hat{\mathbf{p}}(n\,|\,\mathbf{m}_{0:n-1})$ is scheduled followed by the second closest and so on, until the Tth closest sensor is scheduled, completing the cycle. For this example, the parameters are $T = 4$ slots, $T_s = 1$ s, and $K = 100$ sensors deployed in a 2-km-by-2-km square with $\alpha = 3.4$, $\sigma_u = 0.2$ m/s^2, and $\sigma_v = 1$.

$p(v(n)) = \mathcal{N}(v(n); 0; \sigma_v^2)$. This model may arise when sensors measure the power of a radar signal that impinges on the target's surface and bounces back to the sensors [12].

Mimicking an extended Kalman filter approach, this observation model can be linearized around a neighborhood of $\hat{\mathbf{p}}(n \mid n-1)$ to obtain an approximate observation model, which, along with the state evolution in (11), (12), is of the form (1), (2). It is now possible to use the SOI-KF to track the target's position $\mathbf{p}(n)$, which offers an extended SOI-KF variant, that reduces the communication cost of the extended Kalman filter. Because of the linearization of the observation model, the resulting tracker computes an approximate linearized MMSE estimate of the target's position. To study the properties of the resulting estimates we resort to simulations whose results are depicted in figures 4–6. It can be seen that the extended SOI-KF succeeds in tracking the target with position errors smaller or in the order of 15 m. While this accuracy is a result of the specific experiment, the point here is that the extended Kalman filter based on the observations $y_k(n)$ and the extended SOI-KF yield almost identical performance even when the former relies on nonquantized amplitude observations, while the extended SOI-KF is based only on the transmission of a single bit per sensor.

The effect of packet losses in the MSE performance of the SOI-KF is illustrated in Figure 7 for the extended version presented for target tracking. To implement the SOI-KF in a distributed WSN it is assumed that estimates $\hat{\mathbf{x}}(n \mid \mathbf{m}_{0:n})$ are equal at all sensors; see the algorithms in tables 1 and 2. This assumption is needed so that the predicted observations $\hat{y}(n \mid \mathbf{m}_{0:n-1})$ coincide, resulting in the consistency of the sign of innovations $m(n)$ computed at the scheduled sensor with its interpretation at the receiving sensors. In reality estimates $\hat{\mathbf{x}}(n \mid \mathbf{m}_{0:n})$ may be different at different sensors due to erroneously decoded packets. Since this lack of synchronized estimates propagates in time, it is fair to ask whether the accumulation of packet errors ends up garbling the filter's

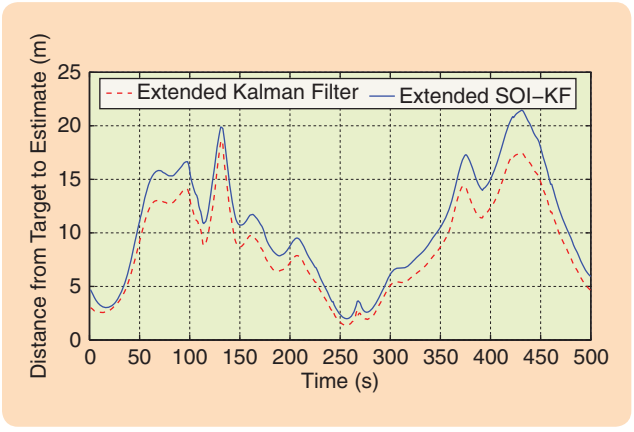

FIGURE 5 Mean-squared error (MSE) of target position estimates. The MSE of the sign of innovations Kalman filter (SOI-KF) is theoretically predicted to be equivalent to the MSE of a Kalman filter where the observation noise covariance matrix is multiplied by $\pi/2$ [7]. For the same estimation problem, the MSE of the extended SOI-KF is thus expected to be close to the MSE of the extended Kalman filter, as illustrated by the simulations for the wireless sensor network target-tracking problem shown in Figure 4.

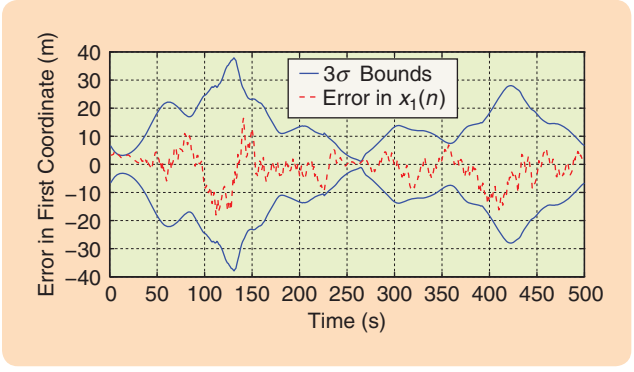

FIGURE 6 Consistency test for extended sign of innovations Kalman filter (SOI-KF). The SOI-KF is derived based on a normal approximation of the prediction pdf $p[\mathbf{x}(n) \mid \mathbf{m}_{0:n-1}]$. For the extended SOI-KF target-tracking problem of Figure 4, further model mismatch is introduced by the linearization of the observation model. Notwithstanding, model consistency is observed as demonstrated by the comparison between the observed location error $\mathbf{p}(n) - \hat{\mathbf{p}}(t)$ and the standard deviation of location estimates $\hat{\mathbf{p}}(n \mid \mathbf{m}_{0:n})$ as given by the square root of the diagonal entries of the covariance matrix $\mathbf{M}(n \mid \mathbf{m}_{0:n})$. The plot shows the first component of the error $\mathbf{p}(n) - \hat{\mathbf{p}}(n)$ and the square root of the first diagonal entry of $\mathbf{M}(n \mid \mathbf{m}_{0:n})$. For consistent models, estimation errors must remain within three times the square root of the mean-squared error, that is, within the 3σ bounds, with 99.87% probability. Consistency is indeed observed for the extended SOI-KF target-tracking problem.

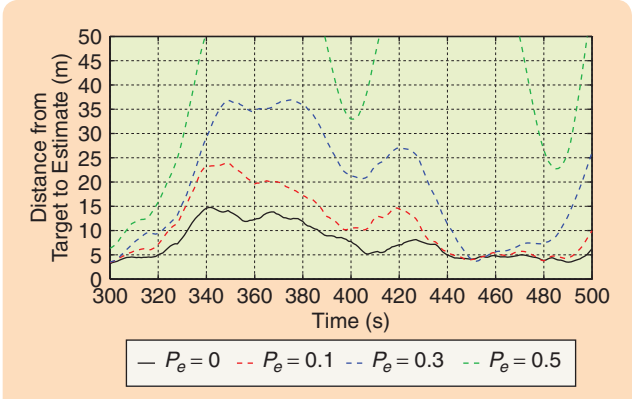

FIGURE 7 Effect of packet errors on the extended sign of innovations Kalman filter (SOI-KF). The root-mean-squared error (RMSE) of the extended SOI-KF for the target-tracking problem of Figure 4 is shown when packets are lost with probabilities $P_e = 0$, $P_e = 0.1$, $P_e = 0.3$, and $P_e = 0.5$. Errors in packet decoding cause the predicted observations $\hat{y}(n \mid \mathbf{m}_{0:n-1})$ to drift across different sensors, which results in an inconsistency between the sign of innovations $m(n)$ computed at the sensor scheduled for transmission and its interpretation by receivers. This effect is not catastrophic if the drift of the predicted observations $\hat{y}(n \mid \mathbf{m}_{0:n-1})$ is small compared with the observation noise variance $\sigma_v(n)$. For the simulation parameters of Figure 4, the filter's RMSE performance degrades smoothly for $P_e = 0.1$ and $P_e = 0.3$. For $P_e = 0.5$ the RMSE error increases beyond acceptable levels.

output. Figure 7 shows that lost packets have a mild effect on the filter's performance by showing the MSE for the extended SOI-KF when packet error probabilities P_e vary between 0 and 0.3. Small and moderate error probabilities $P_e = 0.1$ and $P_e = 0.3$ have a noticeable but not catastrophic effect on the MSE of the filter. When the packet error probability is $P_e = 0.5$, the MSE error increases beyond acceptable levels. The resiliency of the filter is maintained as long as the drift of predicted observations $\hat{y}(n \,|\, \mathbf{m}_{0:n-1})$ is small compared with the observation noise variance $\sigma_v(n)$.

Iterative Sign of Innovations Kalman Filter

Consider now a general quantization scenario, where the scheduled sensor at time n broadcasts an L-bit message $\mathbf{m}(n)$ with entry $m^{(l)}(n)$ denoting the lth bit. As with the SOI-KF, it is natural to consider quantization of the innovations sequence $\tilde{y}(n \,|\, \mathbf{m}_{0:n-1})$, where having l bits broadens the set of possible quantizers. A simple idea is to transmit the L most significant bits of the innovation, but this approach is suboptimal since it amounts to uniform quantization of $\tilde{y}(n \,|\, \mathbf{m}_{0:n-1})$, whose distribution is close to normal. An optimal quantizer for $\tilde{y}(n \,|\, \mathbf{m}_{0:n-1})$ is not difficult to design using Lloyd's algorithm, an idea developed to construct quantized Kalman filters in [8].

Alternatively, we can build on the simplicity of the SOI-KF recursions by designing an iterative means for selecting the individual bits as signs of an extended innovation sequence [8]. More precisely, let $\mathbf{m}^{(l)}(n) := [m^{(1)}(n), \ldots, m^{(l)}(n)]^T$ denote the first l bits of the message $\mathbf{m}(n)$ and define $\hat{y}^{(l)}(n \,|\, \mathbf{m}_{0:n-1}) := \mathbb{E}[y(n) \,|\, \mathbf{m}^{(l)}(n), \mathbf{m}_{0:n-1}]$, which is the observation estimate given past messages and the first l bits of the current message. By convention, $\hat{y}^{(0)}(n \,|\, \mathbf{m}_{0:n-1}) := \mathbb{E}[y(n) \,|\, \mathbf{m}_{0:n-1}]$ is the observation estimate given past messages only. An extended innovations sequence can then be defined, where each observation $y(n)$ generates L terms. When the observation $y(n)$ becomes available, the innovation $\tilde{y}^{(1)}(n \,|\, \mathbf{m}_{0:n-1}) := y(n) - \hat{y}^{(0)}(n \,|\, \mathbf{m}_{0:n-1})$ is first defined followed by $\tilde{y}^{(2)}(n \,|\, \mathbf{m}_{0:n-1}) := y(n) - \hat{y}^{(1)}(n \,|\, \mathbf{m}_{0:n-1})$. This process is repeated L times, and at the lth step the innovation $\tilde{y}^{(l)}(n \,|\, \mathbf{m}_{0:n-1}) := y(n) - \hat{y}^{(l-1)}(n \,|\, \mathbf{m}_{0:n-1})$ is added to the sequence. Messages are then obtained as the signs of the elements of this sequence $m^{(l)}(n) := \text{sign}[\tilde{y}^{(l)}(n \,|\, \mathbf{m}_{0:n-1})]$.

This iterative process is illustrated in Figure 8. An observation estimate $\hat{y}^{(0)}(n \,|\, \mathbf{m}_{0:n-1})$ is computed using past messages and, depending on whether the observation $y(n)$ falls to the right or left of this estimate, the most significant bit $m^{(1)}(n)$ is set to +1 or −1. The observation estimate is updated using this bit to obtain an improved estimate $\hat{y}^{(1)}(n \,|\, \mathbf{m}_{0:n-1})$. The second most significant bit is subsequently set to +1 or −1 depending on whether $y(n)$ is to the right or left of this updated estimate. This process is repeated sequentially until the Lth bit of the message $\mathbf{m}(n)$ is computed.

A subtlety in this iterative scheme is that the observation estimates are not linear transformations of the state estimates, that is, $\hat{y}^{(l)}(n \,|\, \mathbf{m}_{0:n-1}) \neq \mathbf{h}^T(n)\, \mathbb{E}[\mathbf{x}(n) \,|\, \mathbf{m}^{(l)}(n), \mathbf{m}_{0:n-1}] := \mathbf{h}^T(n)\hat{\mathbf{x}}^{(l)}(n \,|\, \mathbf{m}_{0:n-1})$. In fact, it follows from the observation model that the estimate of the observations is $\hat{y}^{(l)}(n \,|\, \mathbf{m}_{0:n-1}) = \mathbf{h}^T(n)\hat{\mathbf{x}}^{(l)}(n \,|\, \mathbf{m}_{0:n-1}) + \mathbb{E}[v(n) \,|\, \mathbf{m}^{(l)}(n)]$. The second term in this sum is nonzero in general because the message bits $\mathbf{m}^{(l)}(n)$ contain information about the noise $v(n)$.

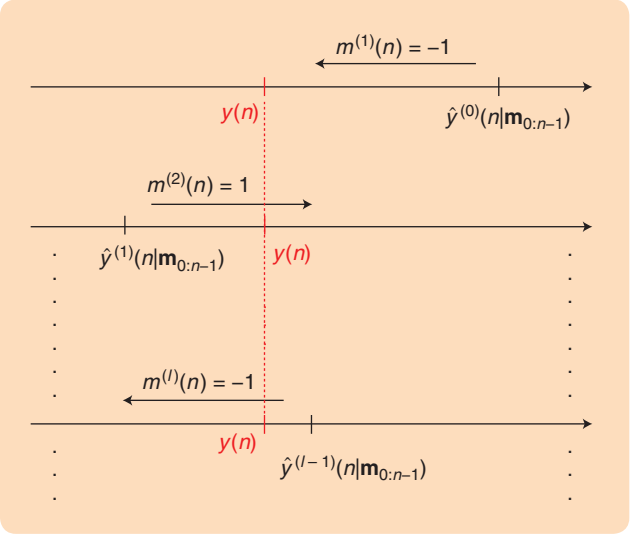

FIGURE 8 Message generation in iterative sign of innovations Kalman filter (SOI-KF). The iterative SOI-KF is an adaptation of the SOI-KF that allows transmission of L-bit messages $\mathbf{m}(n)$. The first bit $m^{(1)}(n)$ is the sign of the innovation. To compute subsequent bits, estimates $\hat{y}^{(l)}(n|\mathbf{m}_{0:n-1})$ incorporating information from past messages and the first $l < L$ bits of the current message are used. Bits are iteratively defined as the sign of this extended innovation sequence. This scheme is illustrated here as a series of threshold comparisons. An observation estimate $\hat{y}^{(0)}(n|\mathbf{m}_{0:n-1})$ is computed using past messages. Depending on whether the observation $y(n)$ falls to the right or left of this estimate, the most significant bit $m^{(1)}(n)$ is set to +1 or −1. The observation estimate is updated using this bit to obtain an improved estimate $\hat{y}^{(1)}(n|\mathbf{m}_{0:n-1})$. The second most significant bit is set to +1 or −1 depending on whether $y(n)$ is to the right or left of this updated estimate. This process is repeated sequentially until the Lth bit of the message $\mathbf{m}(n)$ is obtained. In the lth step, the observation $y(n)$ is compared with $\hat{y}^{(l-1)}(n|\mathbf{m}_{0:n-1})$ to determine the lth bit $m^{(l)}(n)$.

The situation is similar to a state-estimation problem with correlated observation noise. A simple solution in related cases consists of augmenting the state $\mathbf{x}(n)$ to include the observation noise $v(n)$. The augmented state estimates include the noise estimate needed to compute $\hat{y}^{(l)}(n \,|\, \mathbf{m}_{0:n-1})$. Specifically, define the augmented state $\mathbf{x}_{\text{A}}(n) := [\mathbf{x}^T(n), v(n)]^T$ by appending the noise $v(n)$ to the state $\mathbf{x}(n)$ and the augmented driving noise $\mathbf{u}_{\text{A}}(n) := [\mathbf{u}^T(n), v(n)]^T$. Using these definitions along with $\mathbf{A}_{\text{A}}(n) = \begin{pmatrix} \mathbf{A}(n) & 0 \\ 0 & 0 \end{pmatrix}$ and $\mathbf{h}_{\text{A}}(n) := [\mathbf{h}^T(n), 1]^T$, the model in (1), (2) can be rewritten as

$$\mathbf{x}_{\text{A}}(n) = \mathbf{A}_{\text{A}}(n)\mathbf{x}_{\text{A}}(n-1) + \mathbf{u}_{\text{A}}(n), \tag{13}$$

$$y(n) = \mathbf{h}_{\text{A}}^T(n)\mathbf{x}_{\text{A}}(n) + v_{\text{A}}(n), \tag{14}$$

where, by construction, the new observation noise $v_{\rm A}(n)$ is identically zero and thus can be thought of as normal noise with variance $\sigma^2_{v_{\rm A}}(n)=0$. The correlation matrix of the augmented driving noise is a block-diagonal matrix $\mathbf{C}_{u_{\rm A}}(n)$, with $\mathbf{C}_u(n)$ in the upper left corner and $\sigma^2_v(n)$ in the lower right one.

The augmented state formulation (13), (14) entails a state with increased dimension but is otherwise equivalent to (1), (2). However, the formulation has the appealing property that MMSE estimates of the augmented state $\mathbf{x}_{\rm A}(n)$ contain MMSE estimates of the original state $\mathbf{x}(n)$ and the observation noise $v(n)$. As a result, $\hat{y}^{(l)}(n\,|\,\mathbf{m}_{0:n-1})$ can be obtained as a linear transformation of the augmented estimates $\hat{\mathbf{x}}^{(l)}_{\rm A}(n\,|\,\mathbf{m}_{0:n-1})$. Indeed, with $v_{\rm A}(n)=0$ it follows that $\hat{y}^{(l)}(n\,|\,\mathbf{m}_{0:n-1})=\mathbf{h}^{\rm T}_{\rm A}(n)\hat{\mathbf{x}}^{(l)}_{\rm A}(n\,|\,\mathbf{m}_{0:n-1})$.

The SOI-KF recursions can now be applied iteratively as outlined in the algorithms in tables 3 and 4. The observation-transmission algorithm requires as inputs the previous state estimate $\hat{\mathbf{x}}_{\rm A}(n-1\,|\,\mathbf{m}_{0:n-1})$ and covariance matrix $\mathbf{M}_{\rm A}(n-1\,|\,\mathbf{m}_{0:n-1})$. Using $y(n)$, the predicted estimate and its covariance matrix $\hat{\mathbf{x}}^{(0)}_{\rm A}(n\,|\,\mathbf{m}_{0:n-1})$ and $\mathbf{M}^{(0)}_{\rm A}(n\,|\,\mathbf{m}_{0:n-1})$ are obtained in steps 1 and 2. These steps, which are identical to those performed by the SOI-KF, are employed to initialize a loop for obtaining the L bits of the message $\mathbf{m}(n)$ as summarized in steps 3–7. Each iteration of this loop yields an observation prediction $\hat{y}^{(l-1)}(n\,|\,\mathbf{m}_{0:n-1})$ found by linearly transforming the state prediction $\hat{\mathbf{x}}^{(l-1)}_{\rm A}(n\,|\,\mathbf{m}_{0:n-1})$, as shown in step 4. These predictions are based on past messages $\mathbf{m}_{0:n-1}$ and the $l-1$ bits computed in previous iterations of the loop. The sign of innovations is then computed by comparing $y(n)$ with its prediction $\hat{y}^{(l-1)}(n\,|\,\mathbf{m}_{0:n-1})$ to yield the lth bit $m^{(l)}(n)$ of the current message as outlined in step 5. Finally, the correction step in (6), (7) is run to compute estimates $\hat{\mathbf{x}}^{(l)}_{\rm A}(n\,|\,\mathbf{m}_{0:n-1})$ that incorporate the information contained in the first l bits as summarized in step 6. In these equations, both the augmented state variables and model parameters, replace the nonaugmented variables and parameters in (6), (7). Also recall that the observation noise covariance for the augmented system is zero, which explains why it does not appear in the denominator of the correction equations. Upon completing the Lth iteration, all bits of the message $\mathbf{m}(n)$ are available and subsequently broadcast to all receiving sensors in range.

The iterative SOI-KF reception-estimation algorithm in Table 4 includes the first steps of the observation-transmission algorithm in Table 3 to compute the predicted

TABLE 3 Iterative sign of innovations Kalman filter (SOI-KF) observation-transmission algorithm. Steps 1 and 2 initialize the iterative SOI-KF by acquiring the observation $y(n)=y_{k(n)}(n)$ and computing state estimates based on past messages $\mathbf{m}_{0:n-1}$. The iterative process used to compute the L-bit message $\mathbf{m}(n)$ is carried out by the loop in steps 3–7. A linear transformation of the state estimate $\hat{\mathbf{x}}^{(l-1)}_{\rm A}(n\,|\,\mathbf{m}_{0:n-1})$ in step 4 yields the observation estimate $\hat{y}^{(l-1)}(n\,|\,\mathbf{m}_{0:n-1})$ based on past messages and the first $l-1$ bits of the current message. The lth bit $m^{(l)}(n)$ is then defined as the sign of the difference between the observation $y(n)$ and this prediction in step 5. Step 6 is carried out after the computation of each message bit $m^{(l)}(n)$ to yield state estimates $\hat{\mathbf{x}}^{(l)}_{\rm A}(n\,|\,\mathbf{m}_{0:n-1})$ based on past messages $\mathbf{m}_{0:n-1}$ and the first l bits $m^{(l)}(n)$ of the current message. Note the use of state augmentation.

Algorithm 2-A Iterative SOI-KF – Observation and transmission

Require: $\hat{\mathbf{x}}(n-1|n-1)$ and $\mathbf{M}(n-1|n-1)$ Ensure: $\mathbf{m}(n)$

1: Collect observation $y(n)=y_{k(n)}(n)$

2: Compute predicted estimate $\hat{\mathbf{x}}^{(0)}_{\rm A}(n|\mathbf{m}_{0:n-1})$ and covariance matrix $\mathbf{M}^{(0)}_{\rm A}(n|\mathbf{m}_{0:n-1})$

$$\hat{\mathbf{x}}^{(0)}_{\rm A}(n|\mathbf{m}_{0:n-1})=\mathbf{A}_{\rm A}(n)\hat{\mathbf{x}}_{\rm A}(n-1|\mathbf{m}_{0:n-1})$$
$$\mathbf{M}^{(0)}_{\rm A}(n|\mathbf{m}_{0:n-1})=\mathbf{A}_{\rm A}(n)\mathbf{M}_{\rm A}(n-1|\mathbf{m}_{0:n-1})\mathbf{A}^{\rm T}_{\rm A}(n)+\mathbf{C}_{\mathbf{u}_{\rm A}}(n)$$

3: **for** $l=1$ to L **do** {repeat for length of message}

4: Observation prediction given past messages and previous bits (if any)

$$\hat{y}^{(l-1)}(n|\mathbf{m}_{0:n-1})=\mathbf{h}^{\rm T}_{\rm A}(n)\hat{\mathbf{x}}^{(l-1)}_{\rm A}(n|\mathbf{m}_{0:n-1})$$

5: The lth bit of the $\mathbf{m}(n)$ message is

$$m^{(l)}(n)=\text{sign}[y(n)-\hat{y}^{(l-1)}(n|\mathbf{m}_{0:n-1})]$$

6: Compute corrected estimate $\hat{\mathbf{x}}^{(l)}_{\rm A}(n|\mathbf{m}_{0:n-1})$ and covariance matrix $\mathbf{M}^{(l)}_{\rm A}(n|\mathbf{m}_{0:n-1})$

$$\hat{\mathbf{x}}^{(l)}_{\rm A}(n|\mathbf{m}_{0:n-1})=\hat{\mathbf{x}}^{(l-1)}_{\rm A}(n|\mathbf{m}_{0:n-1})+\frac{(\sqrt{2/\pi})\mathbf{M}^{(l-1)}_{\rm A}(n|\mathbf{m}_{0:n-1})\mathbf{h}_{\rm A}(n)}{\sqrt{\mathbf{h}^{\rm T}_{\rm A}(n)\mathbf{M}^{(l-1)}_{\rm A}(n|\mathbf{m}_{0:n-1})\mathbf{h}_{\rm A}(n)}}m^{(l)}(n)$$

$$\mathbf{M}^{(l)}_{\rm A}(n|\mathbf{m}_{0:n-1})=\mathbf{M}^{(l-1)}_{\rm A}(n|\mathbf{m}_{0:n-1})-\frac{(2/\pi)\mathbf{M}^{(l-1)}_{\rm A}(n|\mathbf{m}_{0:n-1})\mathbf{h}_{\rm A}(n)\mathbf{h}^{\rm T}_{\rm A}(n)\mathbf{M}^{(l-1)}_{\rm A}(n|\mathbf{m}_{0:n-1})}{\mathbf{h}^{\rm T}_{\rm A}(n)\mathbf{M}^{(l-1)}_{\rm A}(n|\mathbf{m}_{0:n-1})\mathbf{h}_{\rm A}(n)}$$

7: **end for**

8: Transmit $\mathbf{m}(n)$

estimate $\hat{\mathbf{x}}_{\mathrm{A}}^{(0)}(n\,|\,\mathbf{m}_{0:n-1})$ and the covariance matrix $\mathbf{M}_{\mathrm{A}}^{(0)}(n\,|\,\mathbf{m}_{0:n-1})$, as shown in step 2. Step 3 denotes reception of the message $\mathbf{m}(n)$, and steps 4–6 represent the loop needed to compute the corrected estimate. The core of this loop is the recursive computation of $\hat{\mathbf{x}}_{\mathrm{A}}^{(l)}(n\,|\,\mathbf{m}_{0:n-1})$ and $\mathbf{M}_{\mathrm{A}}^{(l)}(n\,|\,\mathbf{m}_{0:n-1})$ carried on in step 5. The corrected estimate and corresponding covariance matrix are obtained after the L iterations are completed.

Vector Observations

The state estimation discussed above relies on a message $\mathbf{m}(n)$ of length L formed after quantizing the scalar observation $y(n)$. Below, the general case is addressed where each sensor records a vector observation $\mathbf{y}(n) = \mathbf{H}(n)\mathbf{x}(n) + \mathbf{v}(n)$, where $\mathbf{y}(n) \in \mathbb{R}^q$, $\mathbf{H}(n) \in \mathbb{R}^{q\times p}$, and the noise vector $\mathbf{v}(n) \in \mathbb{R}^q$ has pdf $p[\mathbf{v}(n)] = \mathcal{N}(\mathbf{v}(n); \mathbf{0}, \mathbf{C}_{\mathbf{v}(n)})$. The method pursued here exploits both the correlation between components of $\mathbf{y}(n)$ and the fact that each of its component contains limited information about the state $\mathbf{x}(n)$. Note that if the observation noise is correlated, that is, $\mathbf{C}_{\mathbf{v}(n)} \neq \mathbf{I}_q$, prewhitening can be applied to obtain

$$\begin{aligned}\overline{\mathbf{y}}(n) &= \mathbf{C}_{\mathbf{v}(n)}^{-1/2}\mathbf{y}(n) = \mathbf{C}_{\mathbf{v}(n)}^{-1/2}\mathbf{H}(n)\mathbf{x}(n) + \mathbf{C}_{\mathbf{v}(n)}^{-1/2}\mathbf{v}(n)\\ &= \overline{\mathbf{H}}(n)\mathbf{x}(n) + \overline{\mathbf{v}}(n),\end{aligned} \tag{15}$$

where the noise vector $\overline{\mathbf{v}}(n)$ is now white with covariance matrix $\mathbf{C}_{\overline{\mathbf{v}}(n)} = \mathbf{I}_q$.

When the length of the observation vector exceeds that of the state vector, that is, $q > p$, optimal dimensionality reduction can be performed by employing the QR factorization of the observation matrix $\mathbf{H}(n) = \mathbf{Q}_1(n)\mathbf{R}_1(n)$, where $\mathbf{Q}_1(n) \in \mathbb{R}^{q\times p}$ has p orthonormal columns and $\mathbf{R}_1(n) \in \mathbb{R}^{p\times p}$ is upper triangular. By projecting the whitened observations onto the space spanned by the rows of $\mathbf{Q}_1(n)$, the measurement equation (15) takes the form

$$\check{\mathbf{y}}(n) = \mathbf{Q}_1^T\overline{\mathbf{y}}(n) = \mathbf{R}_1(n)\mathbf{x}(n) + \check{\mathbf{v}}(n), \tag{16}$$

where the new noise vector $\check{\mathbf{v}}(n) := \mathbf{Q}_1^T\overline{\mathbf{v}}(n)$ has covariance matrix $\mathbf{C}_{\check{\mathbf{v}}(n)} = \mathbf{I}_p$. The observation model in (16) is equivalent to the one in (15), but the dimensionality of $\check{\mathbf{y}}(n)$ is p. Without loss of generality we can thus restrict attention to models with $q \leq p$, that is, with observation dimensionality not larger than state dimensionality. When $p > q$ we compute $\check{\mathbf{y}}(n)$ and work with the observation model in (16).

With $q \leq p$ and given L bits for quantizing the q scalar components of the vector $\mathbf{y}(n)$, or the p components of $\check{\mathbf{y}}(n)$ when $q > p$, an optimal bit-allocation strategy requires testing q possibilities per bit, leading to the exponential number q^L of possible quantizers. Instead, an iterative scalar quantization approach can be devised, whereby each bit is selected so as to maximize the expected reduction on the trace of the state estimate's covariance matrix. Specifically, let $\mathbf{M}_{\mathrm{A}}^{(l-1)}(n) := \mathbf{M}_{\mathrm{A}}(n\,|\,\mathbf{m}^{(l-1)}(n), \mathbf{m}_{0:n-1})$ be the augmented state estimate's covariance after $l-1$ bits of

TABLE 4 Iterative sign of innovations Kalman filter (SOI-KF) reception-estimation algorithm. As for the SOI-KF, the iterative SOI-KF reception-estimation algorithm is run continually by all sensors to compute state estimates $\hat{\mathbf{x}}_{\mathrm{A}}(n|\mathbf{m}_{0:n})$. The core of the algorithm is the loop in steps 4–6. In the lth iteration of the loop, the estimate $\hat{\mathbf{x}}_{\mathrm{A}}^{(l-1)}(n|\mathbf{m}_{0:n-1})$ is updated by incorporating the information contained in the lth bit $m^{(l)}(n)$ (step 5). The estimate $\hat{\mathbf{x}}_{\mathrm{A}}(n|\mathbf{m}_{0:n})$ and its corresponding covariance matrix are obtained after L iterations of this loop (steps 7 and 8).

Algorithm 2-B Iterative SOI-KF – Reception and estimation

Require: prior estimate $\hat{\mathbf{x}}(-1|-1)$ and covariance matrix $\mathbf{M}(-1|-1)$

1: **for** $n = 0$ to ∞ **do** {repeat for the life of the network}

2: Compute predicted estimate $\hat{\mathbf{x}}_{\mathrm{A}}^{(0)}(n|\mathbf{m}_{0:n-1})$ and covariance matrix $\mathbf{M}_{\mathrm{A}}^{(0)}(n|\mathbf{m}_{0:n-1})$

$$\begin{aligned}\hat{\mathbf{x}}_{\mathrm{A}}^{(0)}(n|\mathbf{m}_{0:n-1}) &= \mathbf{A}_{\mathrm{A}}(n)\hat{\mathbf{x}}_{\mathrm{A}}(n-1|\mathbf{m}_{0:n-1})\\ \mathbf{M}_{\mathrm{A}}^{(0)}(n|\mathbf{m}_{0:n-1}) &= \mathbf{A}_{\mathrm{A}}(n)\mathbf{M}_{\mathrm{A}}(n-1|\mathbf{m}_{0:n-1})\mathbf{A}_{\mathrm{A}}^T(n) + \mathbf{C}_{\mathbf{u}_{\mathrm{A}}}(n)\end{aligned}$$

3: Receive message $\mathbf{m}(n)$

4: **for** $l = 1$ to L **do** {repeat for length of message}

5: Compute corrected estimate $\hat{\mathbf{x}}_{\mathrm{A}}^{(l)}(n|\mathbf{m}_{0:n-1})$ and covariance matrix $\mathbf{M}_{\mathrm{A}}^{(l)}(n|\mathbf{m}_{0:n-1})$

$$\hat{\mathbf{x}}_{\mathrm{A}}^{(l)}(n|\mathbf{m}_{0:n-1}) = \hat{\mathbf{x}}_{\mathrm{A}}^{(l-1)}(n|\mathbf{m}_{0:n-1}) + \frac{(\sqrt{2/\pi})\mathbf{M}_{\mathrm{A}}^{(l-1)}(n|\mathbf{m}_{0:n-1})\mathbf{h}_{\mathrm{A}}(n)}{\sqrt{\mathbf{h}_{\mathrm{A}}^T(n)\mathbf{M}_{\mathrm{A}}^{(l-1)}(n|\mathbf{m}_{0:n-1})\mathbf{h}_{\mathrm{A}}(n)}}m^{(l)}(n)$$

$$\mathbf{M}_{\mathrm{A}}^{(l)}(n|\mathbf{m}_{0:n-1}) = \mathbf{M}_{\mathrm{A}}^{(l-1)}(n|\mathbf{m}_{0:n-1}) - \frac{(2/\pi)\mathbf{M}_{\mathrm{A}}^{(l-1)}(n|\mathbf{m}_{0:n-1})\mathbf{h}_{\mathrm{A}}(n)\mathbf{h}_{\mathrm{A}}^T(n)\mathbf{M}_{\mathrm{A}}^{(l-1)}(n|\mathbf{m}_{0:n-1})}{\mathbf{h}_{\mathrm{A}}^T(n)\mathbf{M}_{\mathrm{A}}^{(l-1)}(n|\mathbf{m}_{0:n-1})\mathbf{h}_{\mathrm{A}}(n)}$$

6: **end for**

7: Corrected estimate $\hat{\mathbf{x}}_{\mathrm{A}}(n|\mathbf{m}_{0:n}) = \hat{\mathbf{x}}_{\mathrm{A}}^{(L)}(n|\mathbf{m}_{0:n-1})$

8: Corrected covariance matrix $\mathbf{M}_{\mathrm{A}}(n|\mathbf{m}_{0:n}) = \mathbf{M}_{\mathrm{A}}^{(L)}(n|\mathbf{m}_{0:n-1})$

9: **end for**

the message $\mathbf{m}(n)$ are utilized in the iterative SOI-KF corrections and consider the augmented measurement matrix $\mathbf{H}_{\mathrm{A}}(n) \coloneqq [\overline{\mathbf{H}}(n)\ \mathbf{I}_q]$ of the whitened observation model in (15); when $q > p$, consider the reduced-dimensionality model in (16) and redefine $\mathbf{H}_{\mathrm{A}}(n) \coloneqq [\mathbf{R}_1(n)\ \mathbf{I}_q]$. The lth bit is allocated to quantize the ξth entry $y(n, \xi)$ of the observation $\mathbf{y}(n)$ if the corresponding row vector $\mathbf{h}_{\mathrm{A}}^T(n, \xi)$ of the augmented measurement matrix $\mathbf{H}_{\mathrm{A}}(n)$ maximizes the quantity

$$\mathrm{tr}\left(\frac{2}{\pi}\frac{[\mathbf{M}_{\mathrm{A}}^{(l-1)}(n)\mathbf{h}_{\mathrm{A}}(n,\xi)\mathbf{h}_{\mathrm{A}}^T(n,\xi)\mathbf{M}_{\mathrm{A}}^{(l-1)}(n)]_{1:p,\,1:p}}{\mathbf{h}_{\mathrm{A}}^T(n,\xi)\mathbf{M}_{\mathrm{A}}^{(l-1)}(n)\mathbf{h}_{\mathrm{A}}(n,\xi)}\right) = \frac{2}{\pi}\frac{\|[\mathbf{M}_{\mathrm{A}}^{(l-1)}(n)\mathbf{h}_{\mathrm{A}}(n,\xi)]_{1:p}\|^2}{\mathbf{h}_{\mathrm{A}}^T(n,\xi)\mathbf{M}_{\mathrm{A}}^{(l-1)}(n)\mathbf{h}_{\mathrm{A}}(n,\xi)}, \quad (17)$$

among all $\xi = 1, \ldots, q$. This process requires evaluating Lq terms of the form (17) to determine all of the bits of the message $\mathbf{m}(n)$.

Experimental Results

The iterative extended SOI-KF algorithm is tested experimentally and compared to the extended Kalman filter based on nonquantized amplitude observations. The application involves autonomous underwater vehicles performing cooperative multirobot localization [13], [14]. The state vector to be estimated is $\mathbf{x} = [x_1, y_1, \nu_1, \phi_1, x_2, y_2, \nu_2, \phi_2]^T$, where x_i, y_i denote the position coordinates of robot i, ν_i is the horizontal velocity, and ϕ_i is the heading direction (yaw), for $i = 1, 2$. A zero-acceleration model is used to describe the nonholonomic motion of each vehicle with $x_i(n+1) = x_i(n) + \nu_i(n)T_s\cos\phi_i(n)$, $y_i(n+1) = y_i(n) + \nu_i(n)T_s\sin\phi_i(n)$, $\nu_i(n+1) = \nu_i(n) + u_{\dot{\nu}_i}(n)T_s$, $\phi_i(n+1) = \phi_i(n) + u_{\dot{\phi}_i}(n)T_s$, where $T_s = 0.05$ s is the sampling period, $u_{\dot{\nu}_i}(n) \sim \mathcal{N}(u_{\dot{\nu}_i}(n); 0, \sigma_{\dot{\nu}_i}^2)$, and $u_{\dot{\phi}_i}(n) \sim \mathcal{N}(u_{\dot{\phi}_i}(n); 0, \sigma_{\dot{\phi}_i}^2)$, with $\sigma_{\dot{\nu}_1} = \sigma_{\dot{\nu}_2} = 0.28$ m/s^2, $\sigma_{\dot{\phi}_1} = \sigma_{\dot{\phi}_2} = 8.3$ deg/s.

The yaw ϕ_i, pitch ψ_i, and roll ζ_i, of robot i are measured by an onboard MicroStrain 3DM-GX1 AHRS inertial sensor. The noise in these measurements is modeled as zero-mean normal with standard deviations $\sigma_{\phi_i} = \sigma_{\psi_i} = \sigma_{\zeta_i} = 2°$, $i = 1, 2$. The yaw measurement $y_{\phi_i}(n) = \phi_i(n) + v_{\phi_i}(n)$, $v_{\phi_i}(n) \sim \mathcal{N}(v_{\phi_i}(n); 0, \sigma_{\phi_i}^2)$ is processed by the filter to provide corrections for the estimates of the heading direction of the vehicle. The pitch measurement $\psi_{mi}(n) = \psi_i(n) + v_{\psi_i}(n)$, $v_{\psi_i}(n) \sim \mathcal{N}(v_{\psi_i}(n); 0, \sigma_{v_{\psi_i}}^2)$ is combined with the measurement $\nu'_{mi} = \nu'_i(n) + v_{\nu'_i}, v_{\nu'_i}(n) \sim \mathcal{N}(v_{\nu'_i}(n); 0, \sigma_{v_{\nu'_i}}^2)$, $\sigma_{v_{\nu'_i}} = 0.25$ m/s, of the vehicles' longitudinal velocity for constructing a measurement of its horizontal velocity $y_{\nu_i}(n) = \nu'_{mi}(n)\cos\psi_{mi}(n) \simeq \nu_i(n) + v_{\nu_i}(n)$, $i = 1, 2$. To obtain this approximation for $y_{\nu_i}(n)$, a first-order Taylor series approximation is applied, and the noise $v_{\nu_i}(n) \simeq \nu'_{mi}(n)\sin\psi_{mi}(n)v_{\psi_i}(n) - \cos\psi_{mi}(n)v_{\nu'_i}(n)$ is approximated as zero-mean normal with variance $\sigma_{v_{\nu_i}}^2(n) = (\nu'_{mi}(n))^2\sin^2\psi_{mi}(n)\sigma_{\psi_i}^2 + \cos^2\psi_{mi}(n)\sigma_{\nu'_i}^2$. The longitudinal velocity measurement ν'_{mi} is inferred from measurements of the propeller's rate of rotation using a linear approximation [14].

The autonomous underwater vehicles are equipped with acoustic modems for underwater communication and for calculating the range to the transmitting robot using the time of flight and speed of sound in water (1500 m/s). The distance measurement between the two vehicles is $y_{d_{ij}}(n) = \sqrt{(x_i(n) - x_j(n))^2 + (y_i(n) - y_j(n))^2} + v_{d_{ij}}(n)$, where $v_{d_{ij}}(n) \sim \mathcal{N}(v_{d_{ij}}(n); 0, \sigma_{d_{ij}}^2)$ and $\sigma_{d_{ij}} = 10$ m.

The duration of the experiment is about 15 min, during which time the two autonomous underwater vehicles cover distances of 1.16 km and 1.18 km, respectively. Two scenarios are examined. In the first one, the robots communicate to each other their nonquantized amplitude measurements of velocity and heading along with the 24 distance measurements recorded during this experiment; see Figure 9. These measurements are processed locally on each vehicle by an extended Kalman filter to jointly estimate both trajectories. In the second scenario, only 3 bits per scalar measurement are communicated between the two autonomous vehicles, and the iterative extended SOI-KF is used for processing them. As shown in figures 10 and 11, the iterative extended SOI-KF is able to approximate the robots' trajectories estimated by the extended Kalman filter, while using only 3 bits per observation. The root mean squared differences between the extended Kalman filter and iterative extended SOI-KF estimates for the x and y coordinates of the two robots are (2.78 m, 2.95 m) and (1.55 m, 5.35 m), respectively, all of which are less than 0.5% of the distance traveled by each robot.

CONSENSUS-BASED DISTRIBUTED KALMAN FILTERING AND SMOOTHING

We now assume that sensors can exchange information only with neighboring sensors located within a predefined transmission radius. The challenge is to obtain the quantities needed to perform MMSE estimation with manageable communication cost. Toward this end, sensors can exchange messages with their neighbors to estimate quantities that are essential for implementing the centralized Kalman filter recursions. To compute these quantities, sensors use recursive data aggregation schemes to combine their local information with messages they receive from their neighbors. Since the information gathered at different sensors is correlated, these recursive schemes reduce the communication cost of the distributed state-estimation algorithms [15]–[19].

Instrumental for implementing distributed Kalman filter approaches is the notion of consensus averaging, which performs distributed computation of sample averages across sensors. In consensus averaging, each sensor maintains a local state variable that provides an estimate of the desired sample average and performs two steps to update these local state variables. In the first step, each

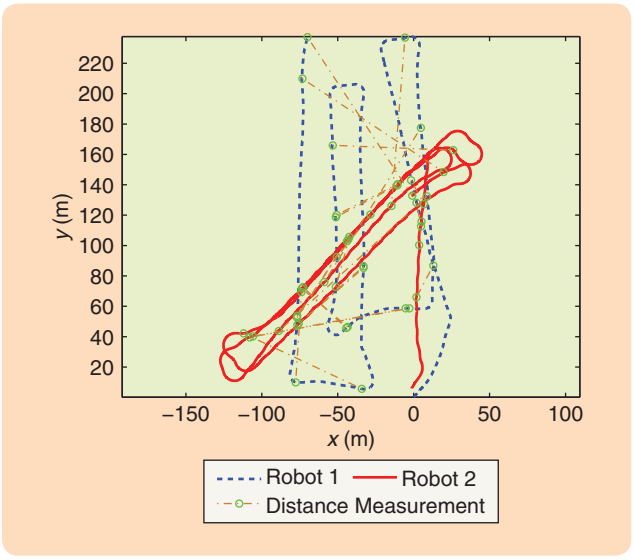

FIGURE 9 Cooperative localization. The estimated trajectories of Robot 1 (dashed line), Robot 2 (solid line), and the locations where robot-to-robot distance measurements $y_{d_{ij}}$ are recorded (small circles connected with dash-dotted lines). Each robot also uses measurements of its heading direction y_{ϕ_i} and horizontal velocity y_{ν_i} for $i = 1, 2$. The time step is $T_s = 0.05$ s. The standard deviations of the zero-mean white normal noise corrupting the distance, heading direction, and horizontal velocity measurements are $\sigma_{\phi_i} = 2°$, $\sigma_{d_{ij}} = 10$ m, and $\sigma_{\nu_i}(n) = \sqrt{(\nu'_{mi}(n))^2 \sin^2\psi_{mi}(n)\sigma^2_{\psi_i} + \cos^2\psi_{mi}(n)\sigma^2_{\nu'_i}}$, where $\sigma_{\psi_i} = 2°$, $\sigma_{\nu'_i} = 0.25$ m/s, and $\psi_{mi}(n)$ and $\nu'_{mi}(n)$, respectively, denote measurements of the pitch angle and longitudinal velocity at time step n. The motion of each robot is described by a constant-velocity model.

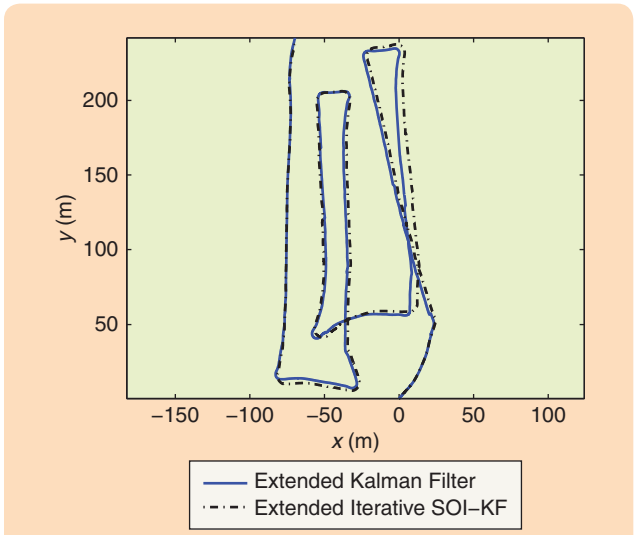

FIGURE 10 Comparison of the extended Kalman filter and iterative sign of innovations Kalman filter (SOI-KF) with 3 bits. The solid line shows the estimated trajectory of Robot 1 when nonquantized measurements of its heading direction ϕ_1, horizontal velocity ν_1, and distance d_{12} to Robot 2 are processed by an extended Kalman filter. The dash-dotted line shows the estimated trajectory of Robot 1 when the same measurements are quantized using 3 bits and processed by the iterative extended SOI-KF. The motion of Robot 1 is described by a constant velocity model. The time step is $T_s = 0.05$ s. The standard deviations of the zero-mean white normal noise corrupting the distance, heading direction, and horizontal velocity measurements are $\sigma_{\phi_1} = 2°$, $\sigma_{d_{12}} = 10$ m, and $\sigma_{\nu_1}(n) = \sqrt{(\nu'_{m1}(n))^2 \sin^2\psi_{m1}(n)\sigma^2_{\psi_1} + \cos^2\psi_{m1}(n)\sigma^2_{\nu'_1}}$, where $\psi_{m1}(n)$ and $\nu'_{m1}(n)$, respectively, denote measurements of the pitch angle and longitudinal velocity at time step n, with $\sigma_{\psi_1} = 2°$ and $\sigma_{\nu'_1} = 0.25$ m/s.

sensor transmits its estimate to neighboring sensors and receives, in return, the neighboring sensors' local estimates. In the second step, each sensor updates its state by forming a weighted sum of its own and neighboring estimates. Upon selecting suitable weights to form these updates and assuming ideal inter-sensor communications, the sensor estimates converge to the desired sample average as the number of updates grows large [20]–[22].

Consensus-averaging schemes perform poorly in the presence of noise in the intersensor links, to the extent that they diverge [23]. However, algorithms based on consensus averaging that are robust against noise perturbations for distributed estimation are available [24]. Consensus-averaging schemes can also be utilized to distribute the centralized Kalman filter recursions. The prediction recursions are identical to (4)–(5). Using the *information form* of the Kalman filter [25, p. 139], however, the corrected state and corresponding corrected error covariance matrix are given by

$$\mathbf{M}(n\,|\,n) := \mathbf{M}(n\,|\,\mathbf{y}_{0:n}) = [\mathcal{I}(n) + \mathbf{M}^{-1}(n\,|\,n-1)]^{-1}, \tag{18}$$

$$\hat{\mathbf{x}}(n\,|\,n) := \hat{\mathbf{x}}(n\,|\,\mathbf{y}_{0:n}) = \hat{\mathbf{x}}(n\,|\,n-1) + \mathbf{M}(n\,|\,n)[\psi(n) - \mathcal{I}(n)\hat{\mathbf{x}}(n\,|\,n-1)], \tag{19}$$

where

$$\mathcal{I}(n) := \sum_{k=1}^{K} \sigma^{-2}_{v_k}(n)\mathbf{h}_k(n)\mathbf{h}^T_k(n),$$
$$\psi(n) := \sum_{k=1}^{K} \sigma^{-2}_{v_k}(n)\mathbf{h}_k(n)y_k(n). \tag{20}$$

If sensors have available local estimates $\hat{\mathbf{x}}_k(n-1\,|\,n-1)$ and the corresponding $\mathbf{M}_k(n-1\,|\,n-1)$, then each sensor could run the prediction recursions in (4)–(5) in a distributed fashion provided that the model parameters $\mathbf{A}(n)$ and $\mathbf{C}_u(n)$ are known at each sensor. However, (18) and (19) require $\mathcal{I}(n)$ and $\psi(n)$ to be acquired at each sensor. This acquisition can be carried out using consensus averaging because $\mathcal{I}(n)$ and $\psi(n)$ are averages with the kth summand located at sensor k.

Existing distributed Kalman filter approaches rely on consensus-based algorithms to form estimates $\hat{\mathcal{I}}_k(n; n:n+N)$ and $\hat{\psi}_k(n; n:n+N)$ through $N+1$ recursions starting at n and finishing at $n+N$ [15], [17]. When these estimates are plugged into (18) and (19), local estimates $\hat{\mathbf{x}}_k(n\,|\,n; n:n+N)$ for the state $\mathbf{x}(n)$ using observations up to time n necessitate consensus iterations performed between times n and $n+N$ as signified by the

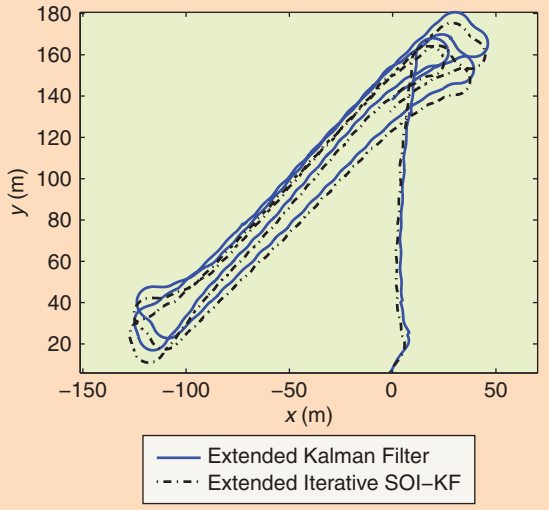

FIGURE 11 Comparison of extended Kalman filter and iterative sign of innovations Kalman filter (SOI-KF) with 3 bits. The solid line shows the estimated trajectory of Robot 2 when nonquantized measurements of its heading direction ϕ_2, horizontal velocity v_2, and distance d_{21} to Robot 1 are processed by the extended Kalman filter. The dash-dotted line shows the estimated trajectory of Robot 2 when the same measurements are quantized using 3 bits and processed by the iterative extended SOI-KF. The motion of Robot 2 is described by a constant velocity model. The time step is $T_s = 0.05$ s. The standard deviations of the zero-mean white normal noise corrupting the distance, heading direction, and horizontal velocity measurements are $\sigma_{\phi_2} = 2°$, $\sigma_{d_{21}} = 10$ m, and $\sigma_{v_2}(n) = \sqrt{(v'_{m2}(n))^2 \sin^2\psi_{m2}(n)\sigma^2_{\psi_2} + \cos^2\psi_{m2}(n)\sigma^2_{v'_2}}$, where $\psi_{m2}(n)$ and $v'_{m2}(n)$, respectively, denote the measurements of the pitch angle and longitudinal velocity at time step n, with $\sigma_{\psi_2} = 2°$ and $\sigma_{v'_2} = 0.25$ m/s.

notation $n{:}n+N$ within $\hat{\mathbf{x}}_k(n\,|\,n; n{:}n+N)$. As a consequence, there is a delay of N steps in obtaining estimates $\hat{\mathbf{x}}_k(n\,|\,n; n{:}n+N)$. This delay confines the operation of the associated distributed Kalman filters to applications with slowly varying $\mathbf{x}(n)$ or fast communications needed to complete $N \gg 1$ consensus iterations within the time interval separating the acquisition of $y_k(n)$ and $y_k(n+1)$. For finite N these distributed Kalman filtering schemes yield estimates $\hat{\mathbf{x}}_k(n\,|\,n; n{:}n+N)$ that are not MSE optimal given the available information in $\hat{\psi}(n; n{:}n+N)$. This suboptimality renders the distributed Kalman filter estimates in [15]–[17] inconsistent with the underlying data model, which in turn is known to yield estimation errors violating the 3σ consistency bounds. This inconsistency may lead to unstable Kalman filters, as illustrated by the numerical examples presented below [28, Sect. 5.4-5.6]. Additionally, these distributed schemes inherit the noise sensitivity present in the consensus-averaging schemes they rely on.

Distributed Kalman Smoothing

As noted above, standard filtering schemes incur a time delay N due to the consensus iterations required to determine $\hat{\mathbf{x}}_k(n\,|\,n; n{:}n+N)$. This delay suggests the development of fixed-lag distributed Kalman smoothers [24]. Fixed-lag smoothers allow sensors, at time instant $n+N$, to form local MMSE optimal smoothed estimates $\hat{\mathbf{x}}_k(n\,|\,n+\nu; n+\nu{:}n+N)$ for $\nu = 0, 1, \ldots, N$, which take advantage of all available data within the interval $[n, n+N]$. This approach leads to estimates with smaller MSE than the corrected estimates $\hat{\mathbf{x}}_k(n\,|\,n; n{:}n+N)$. Sensors can also form zero delay ($N = 0$) corrected estimates $\hat{\mathbf{x}}_k(n\,|\,n; n{:}n)$ as well as anytime linear MSE optimal estimates $\{\hat{\mathbf{x}}_k(n+l-\nu\,|\,n+l; n+l{:}n+N)\}_{l,\nu=0}^{N}$.

Centralized recursions can be derived by state augmentation and modification of the standard Kalman filter recursions [25, p. 177]. To obtain a distributed Kalman smoother variant, sensors must be able to estimate $\mathcal{I}(n)$ and $\psi(n)$ in a distributed fashion. Furthermore, to overcome the limitations present in existing consensus-averaging Kalman filtering schemes, noise-robust alternatives can be used to provide estimates of $\mathcal{I}(n)$ and $\psi(n)$ [24]. Local consensus averaging recursions can be obtained after rewriting $\psi(n)$ as the solution of the separable minimization problem

$$\{\psi_1(n), \ldots, \psi_K(n)\} := \arg \min_{\psi_1=\cdots=\psi_K} \sum_{k=1}^{K} \|\psi_k - K\mathbf{h}_k(n)\sigma_{v_k}^{-2}(n)y_k(n)\|^2. \quad (21)$$

Problem (21) can be solved using the alternating direction method of multipliers algorithm to form local recursions yielding estimates $\hat{\mathcal{I}}_k(n; n{:}n+N)$ and $\hat{\psi}_k(n; n{:}n+N)$ [24], [29]. These local recursions entail second-order updates of these estimates, per sensor k, which involve linear combinations of the local estimates $\{\hat{\psi}_{k'}(n; n{:}n+\nu-1), \hat{\psi}_{k'}(n; n{:}n+\nu-2)\}_{k'\in\mathcal{N}_k}$ and $\{\hat{\mathcal{I}}_{k'}(n; n{:}n+\nu-1), \hat{\mathcal{I}}_{k'}(n; n{:}n+\nu-2)\}_{k'\in\mathcal{N}_{k'}}$ respectively, which are received from its neighbors in $\mathcal{N}_k$. The estimates $\hat{\mathcal{I}}_k(n; n{:}n+N)$ and $\hat{\psi}_k(n; n{:}n+N)$ converge to $\mathcal{I}(n)$ and $\psi(n)$ and reach consensus as $N \to \infty$ under ideal links. In the presence of communication noise, these estimates converge in the mean sense, while their noise-induced variance remains bounded. This noise resiliency allows sensors to exchange quantized data further lowering communication cost.

The key behind forming anytime linear MSE optimal estimates is to view $\hat{\psi}_k(n; n{:}n+\nu)$ as a consensus-enriched observation vector per sensor k. Besides $y_k(n)$, variables $\hat{\psi}_k(n; n{:}n+\nu)$ include data received from neighboring sensors. Because $\hat{\psi}_k(n; n{:}n+\nu)$ contains more information about the state $\mathbf{x}(t)$ than the information contained in $y_k(n)$, state estimates based on $\hat{\psi}_k(n; n{:}n+\nu)$ exhibit improved MSE performance. Furthermore, note that variables $\hat{\psi}_k(n; n{:}n+\nu)$ become more informative as ν increases. Instrumental in deriving the distributed smoother is the linear relationship

$$\hat{\psi}_k(n; n{:}n+\nu) = \hat{\mathcal{I}}_k(n; n{:}n+\nu)\mathbf{x}(n) + \hat{\mathbf{v}}_k(n; n{:}n+\nu), \quad (22)$$

between $\hat{\psi}_k(n;n:n+\nu)$ and $\hat{\mathcal{I}}_k(n;n:n+\nu)$, which is valid for ideal intersensor links [24]. In (22) $\hat{\mathbf{v}}_k(n;n:n+\nu)$ is zero-mean normal with a covariance matrix that can be evaluated at sensor k using recursions similar to those involved in computing $\hat{\mathcal{I}}_k(n;n:n+\nu)$.

Each sensor utilizes its local consensus-enriched data $\hat{\psi}_k(n;n:n+\nu)$ to implement local smoothing recursions corresponding to the observation model (22) and state model (1). These local Kalman smoother recursions as well as the distributed scheme for forming estimates $\hat{\psi}_k(n;n:n+\nu)$ and $\hat{\mathcal{I}}_k(n;n:n+\nu)$ constitute the distributed Kalman smoother. After all sensors initialize their state and covariance matrices based on a priori information, sensor k performs the following two steps at time $n+N$:

- **Step 1.** Compute estimates $\hat{\psi}_k(n+\nu;n+\nu:n+N)$ and $\hat{\mathcal{I}}_k(n+\nu;n+\nu:n+N)$, for $\nu=0,\ldots,N$, using the consensus-averaging recursions in [24].
- **Step 2.** Determine anytime state estimates $\hat{\mathbf{x}}_k(n+\nu-l\,|\,n+\nu;n+\nu:n+N)$ for $l,\nu=0,\ldots,N$ using the smoothing recursions corresponding to the observation model (22) and the state model (1).

The local anytime linear estimates $\hat{\mathbf{x}}_k(n+\nu-l\,|\,n+\nu;n+\nu:n+N)$ obtained at sensor k using the distributed smoother are MSE optimal in the sense that for $l=0,\ldots,N$

$$\hat{\mathbf{x}}_k(n+\nu-l\,|\,n+\nu;n+\nu:n+N) := \overline{\mathbb{E}}[\mathbf{x}(n+\nu-l)\,|\,\{\hat{\psi}_k(n';n':n'+N)\}_{n'=0}^{n},\ \{\hat{\psi}_k(n';n':n+N)\}_{n'=n+1}^{n+\nu}],$$

where $\overline{\mathbb{E}}[\mathbf{x}\,|\,\mathbf{y}] := \mathbf{C}_{xy}\mathbf{C}_{yy}^{-1}\mathbf{y}$ denotes the (linear) MMSE estimator of $\mathbf{x}$ given $\mathbf{y}$, while $\mathbf{C}_{xy}$ is the cross-covariance matrix between $\mathbf{x}$ and $\mathbf{y}$. As the number of consensus iterates goes to infinity, the local state estimates converge to their centralized counterparts, that is, for $k=1,\ldots,K$, and $n=0,1,\ldots,$

$$\lim_{N\to\infty}\hat{\mathbf{x}}_k(n-l\,|\,n;n:n+N)=\hat{\mathbf{x}}(n-l\,|\,n),$$

$$\lim_{N\to\infty}\mathbf{M}_k(n-l\,|\,n;n:n+N)=\mathbf{M}(n-l\,|\,n),\ l=0,\ldots,N.$$

Over the interval $[n,n+N]$, the distributed smoother produces a sequence of local MSE optimal state estimates $\hat{\mathbf{x}}_k(n-l\,|\,n;n:n+\nu)$, for $l,\nu=0,\ldots,N$. The MSE associated with $\{\hat{\mathbf{x}}_k(n-l\,|\,n;n:n+\nu)\}_{l=0}^{N}$, decreases as ν increases since $\hat{\psi}_k(n;n:n+\nu)$ converges to $\hat{\psi}(n)$. Moreover, the distributed smoother exhibits robustness against noise perturbations and trades off delay for MSE reduction.

Simulated Test Case

The MSE performance of the distributed smoother is compared here with alternative distributed Kalman filter approaches, namely, those in [15] and [16]. Figure 12 depicts the MSE of the aforementioned approaches at a given sensor, in the presence of quantization noise. The quantization noise at sensor k is uniformly distributed over $[-Q_k/2^{m_k}, Q_k/2^{m_k}]^p$, where Q_k specifies the dynamic range of the quantizer and m_k denotes the number of bits used to quantize $\hat{\psi}_k$ and $\hat{\mathcal{I}}_k$. It can be seen that the distributed smoother estimates the state process through the local corrected estimates $\hat{x}_k(n\,|\,n;n:n)$, and the MSE reaches steady state. In Figure 12, the MSE of the local estimators discussed in [15] and [16] diverges. This behavior is expected since the corresponding distributed filtering schemes are inconsistent with the true observation model, causing errors to accumulate for the fast varying $x(n)$. Figure 13 depicts the estimation error and corresponding 3σ bounds at a given sensor, when estimating at time slot n state $x(n-l)$ using $\hat{x}_k(n-l\,|\,n;n:n)$ for $l=0,1,5$. Note that larger delays lead to lower MSEs.

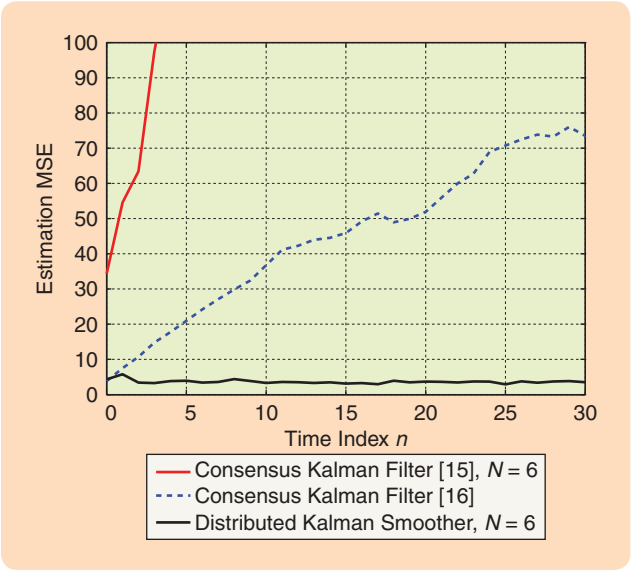

FIGURE 12 Empirically estimated mean-squared error (MSE) versus time for various distributed Kalman filtering/smoothing approaches. A wireless sensor network comprising $K=60$ sensors is used. The state process is scalar with $\mathbf{A}(n)=1$, $C_u=4$, and initial conditions $\mathbb{E}[x(-1)]=0$ and $\sigma_x^2(-1)=1$. At time n, sensor k acquires the scalar $y_k(n)$ for which $h_j(n)=h_j$ is normally distributed, while $\sigma_{vk}^2=1.5$. The number of consensus iterations used to estimate $\mathcal{I}(\mathcal{N})$ and $\boldsymbol{\psi}(n)$ is fixed at $N=6$.

REDUCED-DIMENSIONALITY KALMAN FILTER

Dimensionality reduction of vector observations $\mathbf{y}_k(n)$ can exploit the correlation present among the entries of sensor vector observations $\mathbf{y}_k(n)$ further reducing transmission cost. Besides allowing sensors to comply with power and bandwidth budgets, dimensionality reduction is further motivated by state-estimation problems with rapidly changing states, for example when tracking high-speed targets. In these cases observation sampling is faster than communication rate, and dimensionality-reducing operators are necessary for matching the sampling period of a signal, dictated by a desirable estimation accuracy, with the

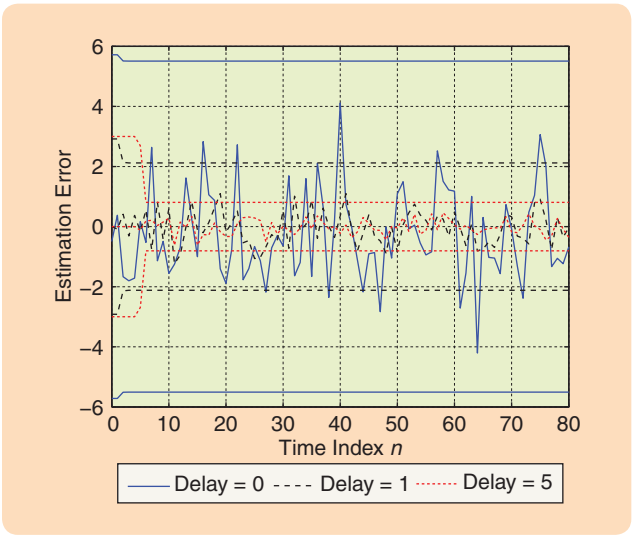

FIGURE 13 Estimation delay and mean-squared error tradeoff for the distributed Kalman smoother. The estimation error associated with $\hat{x}_k(n-l|n;n:n)$ decreases as the delay l increases. The 3σ bounds illustrate the consistency of the distributed Kalman smoother.

data rate that is supported by the communication links. The following subsections outline two distributed Kalman filtering approaches based on reduced-dimensionality data. The first approach is developed for WSN topologies that include a fusion center, whereas the second approach is developed for infrastructureless ad hoc WSNs.

Fusion Center Based Wireless Sensor Networks

Consider first a WSN topology involving K sensors linked with a fusion center; see Figure 14. Each sensor acquires an $M_k \times 1$ observation vector $\mathbf{y}_k(n)$, obeying the linear model

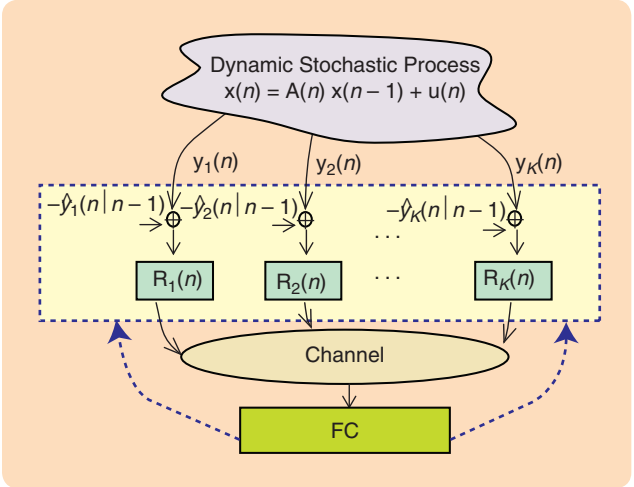

FIGURE 14 Distributed setup for estimating $\mathbf{x}(n)$ using fusion-center-based wireless sensor networks. Sensors receive the state prediction $\hat{\mathbf{x}}(n|n-1)$ from the fusion center. Then, the sensors form the observation innovation signal $\tilde{y}_k(n) = y_k(n) - \hat{y}_k(n|n-1)$ and, prior to transmission to the fusion center, reduce its dimensionality using a wide matrix $\mathbf{R}_k(n)$.

in (1). The entries of $\mathbf{y}_k(n)$ are obtained, for example, by collecting M_k samples per time slot n. Using an $l_k \times M_k$ matrix $\mathbf{R}_k(n)$ with $l_k \leq M_k$, sensor k forms the reduced-dimensionality vector $\mathbf{R}_k(n)[\mathbf{y}_k(n) - \hat{\mathbf{y}}_k(n|n-1)]$, where $\hat{\mathbf{y}}_k(n|n-1)$ is subtracted from $\mathbf{y}_k(n)$ to save power during transmission. In practice, sensors transmit their data to the fusion center over nonideal links that are corrupted by multiplicative fading and additive noise. Using time division multiple access or an alternative scheme supporting orthogonal communication [30, Chap. 15], each sensor can transmit its information without interfering with other sensors. In this case, the signal received at the fusion center from sensor k is

$$\mathbf{r}_k(n) = \mathbf{D}_k\mathbf{R}_k(n)[\mathbf{y}_k(n) - \hat{\mathbf{y}}_k(n|n-1)] + \mathbf{w}_k(n), \quad (23)$$

where $\mathbf{D}_k$ accounts for the multiplicative fading corrupting the link from sensor k to the fusion center, and $\mathbf{w}_k(n)$ denotes the reception noise at the fusion center. The fusion center concatenates the received vectors $\mathbf{r}_k(n)$ in a super vector $\mathbf{r}(n)$ and aims to track the state $\mathbf{x}(n)$ using the received data $\{\mathbf{r}(i)\}_{i=0}^{n}$. Specifically, the fusion center seeks the (linear) MMSE state estimates $\hat{\mathbf{x}}(n) = \overline{\mathbb{E}}[\mathbf{x}(n)\,|\,\mathbf{r}(0),\ldots,\mathbf{r}(n)]$ recursively based on the reduced-dimensionality data in (23). Having available $\hat{\mathbf{x}}(n-1|n-1)$, the fusion center relies on the innovation process of the received signal $\mathbf{r}(n)$, namely $\tilde{\mathbf{r}}(n|n-1) := \mathbf{r}(n) - \overline{\mathbb{E}}[\mathbf{r}(n)\,|\,\mathbf{r}(0),\ldots,\mathbf{r}(n-1)] = \mathbf{r}(n)$, and adds to the predictor $\hat{\mathbf{x}}(n|n-1) = \overline{\mathbb{E}}[\mathbf{x}(n)\,|\,\mathbf{r}(0),\ldots,\mathbf{r}(n-1)]$ the correction term $\overline{\mathbb{E}}[\mathbf{x}(n)\,|\,\tilde{\mathbf{r}}(n|n-1)]$ to obtain the corrected estimate $\hat{\mathbf{x}}(n|n)$.

The dimensionality-reducing matrices $\{\mathbf{R}_k(n)\}_{k=1}^{K}$ can be obtained to minimize the estimation MSE $E[\|\mathbf{x}(n) - \hat{\mathbf{x}}(n|n)\|^2]$ at the fusion center. Recall that $\hat{\mathbf{x}}(n|n)$ is a function of $\{\mathbf{R}_k(n)\}$. Since each sensor has a prespecified transmission budget P_k, a power constraint is required to bound the covariance of the transmitted signal $\mathbf{R}_k(n)[\mathbf{y}_k(n) - \hat{\mathbf{y}}_k(n|n-1)]$. Then, the dimensionality-reducing operators can be obtained as the minimizer of [12]

$$\min_{\{\mathbf{R}_k(n)\}_{k=1}^{K}} \mathbb{E}[\|\mathbf{x}(n) - \hat{\mathbf{x}}(n|n)\|^2], \text{ subject to}$$
$$\text{subject to } \operatorname{tr}(\mathbf{R}_k(n)\mathbf{C}_{\tilde{y}_k\tilde{y}_k}(n|n-1)\mathbf{R}_k^T(n)) \leq P_k$$
$$\text{for all } k = 1,\ldots,K, \quad (24)$$

where $\mathbf{C}_{\tilde{y}_k\tilde{y}_k}(n|n-1)$ is the covariance matrix of $\tilde{\mathbf{y}}_k(n|n-1) := \mathbf{y}_k(n) - \hat{\mathbf{y}}_k(n|n-1)$. Upon solving (24), the fusion center feeds $\{\mathbf{R}_k(n)\}_{k=1}^{K}$ back to the sensors for them to perform dimensionality reduction. Since the fusion center has sufficient resources, it can mitigate the fusion center to sensor channel effects and ensure reliable transmission of the dimensionality-reducing matrices to the sensors.

Power awareness is effected through the constraints present in (24), so that the dynamic range of the transmitted quantities is regulated in the mean sense. Further,

selecting $\hat{\mathbf{y}}_k(n\,|\,n-1) \coloneqq \overline{\mathbb{E}}[\mathbf{y}_k(n)\,|\,\mathbf{r}(0),\ldots,\mathbf{r}(n-1)] = \mathbf{H}_k(n)\hat{\mathbf{x}}(n\,|\,n-1)$ as in [12] and [32] enables sensor k to reduce the required power because it transmits the nonredundant local innovation $\mathbf{y}_k(n) - \hat{\mathbf{y}}_k(n\,|\,n-1)$, which has a smaller variance and thus dynamic range than its raw data $\mathbf{y}_k(n)$. This scheme is possible as long as the fusion center feeds back to each sensor the state predictor $\hat{\mathbf{x}}(n\,|\,n-1)$. It turns out that the state predictor, as well as the corresponding covariance matrix in the reduced-dimensionality Kalman filter, can be determined along the lines of the full-dimensionality Kalman filter. However, the Kalman gain per time step n, as well as the correction recursions for the corrected estimate and the corresponding error covariance, are obtained in a different manner. These recursions turn out to depend on the dimensionality-reducing matrices $\{\mathbf{R}_k(n)\}_{k=1}^K$, the channels $\mathbf{D}_k$, and the noise covariance $\mathbf{C}_{w_k}$ [12].

The dimensionality-reducing matrices in (24) can be derived in closed form only for $K = 1$. In the multisensor scenario, however, solving (24) incurs complexity that increases exponentially with K [33]. For $K = 1$, it turns out that the optimal compression matrices $\mathbf{R}_1^o(n)$ can be written as a function of the state and observation model parameters $\mathbf{A}(n), \mathbf{C}_u(n), \mathbf{H}_1(n)$, and $\mathbf{C}_{v_1}(n)$ as well as the channel matrix $\mathbf{D}_1$, the noise covariance $\mathbf{C}_{w_1}$, and the power budget P_1. The channel matrices can be obtained at the fusion center using well-established training techniques [34, p. 383], while the model parameters $\mathbf{A}(n), \mathbf{C}_u(n), \mathbf{H}_1(n)$ can be determined based on the physics of the problem, for example, the target kinematics [28, Chap. 6]. Intuitively, the optimal matrix $\mathbf{R}_1^o(n)$ selects the entries of $\tilde{\mathbf{y}}_1(n\,|\,n-1)$ in which $\tilde{\mathbf{x}}(n\,|\,n-1)$ is strongest and transmits them over the highest quality channels. In the multisensor case, the results obtained for the single sensor scenario can be utilized to develop a block coordinate descent algorithm, where the cost in (24) is minimized with respect to a matrix $\mathbf{R}_k(n)$, for $k = 1,\ldots,K$, while treating the rest as fixed [12]. This algorithm is ensured to converge at least to a stationary point of the cost in (24).

In Figure 15, the MSE performance of the channel-aware approach in [12] is compared with the channel-unaware approach in [31]. Both the MSE of [12] and [31] obtained from the trace of the corrected error covariance matrix (theoretical), as well as through Monte Carlo simulations (empirical), are plotted. In the channel-aware approach the MSE reaches steady state, while in the channel unaware approach the MSE diverges. This divergence is expected since the channel-unaware approach does not account for channel imperfections when designing the dimensionality-reducing operators.

Ad Hoc Wireless Sensor Networks

Consider next an ad hoc WSN deployed to estimate the state vector in a distributed fashion. The motivation behind using ad hoc instead of fusion center topologies is to improve resilience to isolated points of failure. During time slot n, a single transmitting (broadcasting) sensor m is

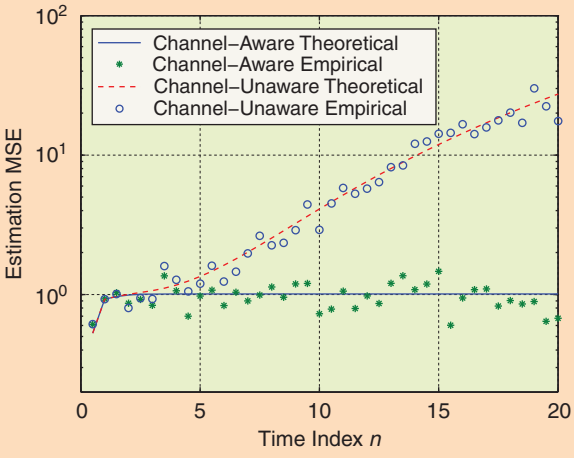

FIGURE 15 Estimated mean-squared error at the fusion center versus time n. The setting involves a wireless sensor network with $K = 50$ sensors and each sensor acquiring $M_k = 10$ temperature measurements. The temperature evolves according to a zero-acceleration propagation model. Each sensor reduces the dimensionality of $\tilde{\mathbf{y}}_k(n|n-1)$ to $l_k = 1$, and transmits these scalars over noisy links to the fusion center.

operating. Sensor m uses a $l \times M_m$ matrix $\mathbf{R}_m(n)$ with $l \leq M_m$ to form and broadcast the reduced-dimensionality vector $\mathbf{R}_m(n)\check{\mathbf{y}}_m(n)$ to all neighboring sensors. The vector $\check{\mathbf{y}}_m(n)$ is the innovation signal of the observation $\mathbf{y}_m(n)$ for ideal channels, and $\check{\mathbf{y}}_m(n) = \mathbf{y}_m(n)$ for nonideal channels; see [12] for further details. The mapping between time slot n and the transmitting sensor m can be determined using distributed sensor scheduling techniques [35]. One possible way is to allow neighboring sensors with high-quality observations to broadcast their data, letting all sensors, even those with less informative data, to form accurate state estimates.

The reduced-dimensionality data $\mathbf{R}_m(n)\check{\mathbf{y}}_m(n)$ are transmitted from sensor m to other sensors in range over links corrupted by multiplicative fading and additive noise. Based on the received data $\mathbf{r}_k^m(n)$ from sensor m, all sensors in range wish to form linear MMSE estimates of $\mathbf{x}(n)$. The dimensionality-reducing operator $\mathbf{R}_m(n)$ at the transmitting sensor m is selected to minimize the estimation MSE at the receiving sensor with the worst reception link from the broadcasting sensor m. The worst reception link is the one with the smallest reception SNR. In summary, the reduced-dimensionality Kalman filter state estimator entails two phases. In the first phase, the broadcasting sensor forms $\check{\mathbf{y}}_m(n)$ from $\mathbf{y}_m(n)$, performs dimensionality reduction of $\check{\mathbf{y}}_m(n)$ and broadcasts $\mathbf{R}_m(n)\check{\mathbf{y}}_m(n)$ to all other sensors in range. The second phase involves reception of $\mathbf{r}_k^m(n)$ at each sensor k, through nonideal links, and further implementation of corresponding Kalman filter recursions.

The reduced-dimensionality schemes for state estimation developed for ad hoc topologies and WSNs with a fusion center have complementary strengths. Dimensionality

reduction for WSNs with a fusion center offers better MSE performance because all available data at each time instant are gathered and processed at the fusion center. Their ad hoc counterparts are flexible to trade off estimation accuracy for robustness. In the ad hoc setup, only one sensor transmits per time slot, which further improves power efficiency [12].

Target Tracking Test Case

Distributed Kalman filtering using reduced-dimensionality data is applicable to target-tracking based on distance-only measurements. Sensors are deployed to track the position $\mathbf{x}(n) = [x_1(n), x_2(n)]^T$ of a target. The associated state and observation models are given in (11)–(12). This setting is also considered in the SOI-KF context, where sensors quantize their scalar observations prior to transmission. As mentioned earlier, dimensionality reduction is an essential prerequisite to quantization. The scalar observations acquired by the SOI-KF can be obtained after compressing a vector of multiple observations in the interest of complying with power and bandwidth constraints. To isolate the impact of dimensionality reduction on the MSE of tracking estimates, no quantization effects are considered here. The local tracking recursions corresponding to the distributed tracking scheme described earlier can be obtained after applying similar steps as in the standard extended Kalman filter, where the nonlinear observation model is linearized in the neighborhood of $\hat{\mathbf{x}}(n|n-1)$.

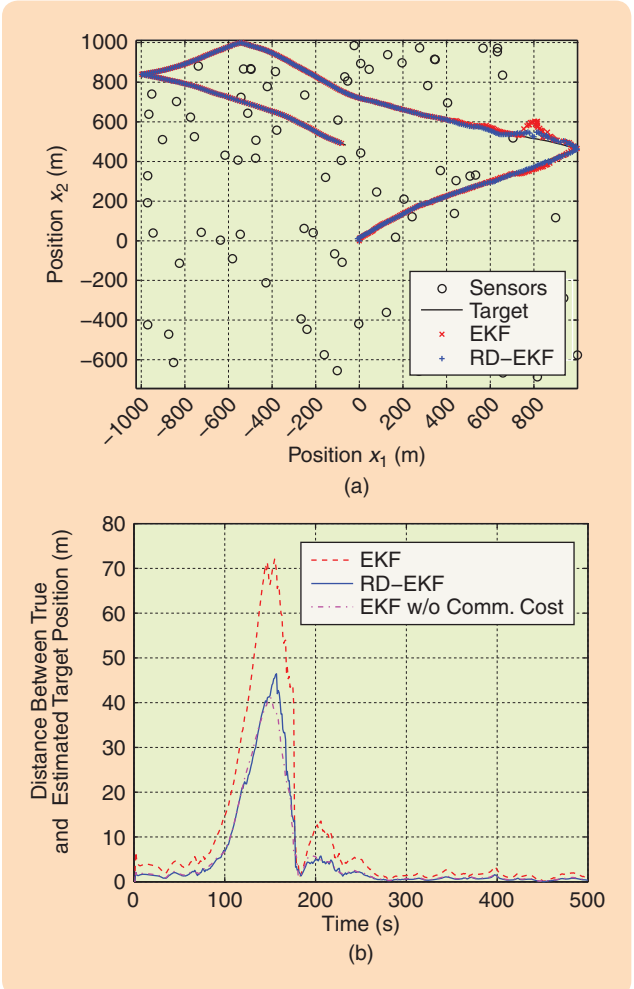

FIGURE 16 Target-tracking with the extended Kalman filter (EKF) and the reduced-dimensionality extended Kalman filter (RD-EKF) under nonideal links. The wireless sensor network has the same characteristics as the network utilized in Figure 4. Part (a) depicts the true and estimated target trajectories obtained by the RD-EKF and the standard EKF. In the RD-EKF, each sensor acquires observation data having dimensionality 10×1 and compresses the data to a scalar. The standard deviation of the estimation error associated with the RD-EKF and the EKF is shown in (b).

Figure 16(a) depicts the true and estimated target trajectories obtained by the reduced-dimensionality extended Kalman filter and the standard extended Kalman filter. Further, Figure 16(b) displays the standard deviation of the estimation error associated with the standard extended Kalman filter and its reduced-dimensionality version. Reduced extended Kalman filter is applied so that the dimensionality of the observation vectors is reduced down to one (scalar). When the extended Kalman filter is implemented under the same communication cost, meaning that one scalar is transmitted per sensor per time slot, then the reduced extended Kalman filter outperforms the standard extended Kalman filter. This behavior is expected because the data broadcast in the reduced extended Kalman filter are more informative about the target position than in the extended Kalman filter, since they are constructed by judicious dimensionality reduction of multiple observations. The estimation error's standard deviation of the extended Kalman filter with no communication rate constraint is also plotted in Figure 16(b). In this case, each sensor transmits ten scalars per time step n, instead of one scalar as in reduced dimensionality extended Kalman filter. Tracking performance is almost identical for both the reduced and standard extended Kalman filters.

CONCLUSIONS

Several strategies to reduce the communication cost of state-estimation problems in WSNs were presented. To reduce the cost of disseminating the information collected by distributed sensors we discussed strategies to exploit the redundancy in information provided by individual observations collected at different sensors, different observations collected at different sensors, and different observations acquired at the same sensor.

The first strategy leads to the development of quantized Kalman filters based on the sign of the innovations sequence. We discussed how the resulting SOI-KF has MSE performance and complexity that come close to those of the equivalent Kalman filter based on the nonquantized observations. Specifically, the computational cost is almost identical except that the SOI-KF necessitates computation of a

scalar square root for each transmitted bit. The MSE performances of the SOI-KF and the Kalman filter can be compared by looking at the MSE reductions at each iteration. The MSE reduction of the SOI-KF is $2/\pi$ times smaller than the MSE reduction of a Kalman filter based on the nonquantized observations, entailing a relative performance penalty of $1 - 2/\pi \approx 0.36$ for quantization to a single bit per observation. An iterative version of the SOI-KF was discussed to deal with multibit quantization. The promise of the approach was illustrated with simulated as well as experimental results.

The information redundancy resulting from the correlated observations acquired among different sensors is exploited through the construction of an iterative data aggregation and state-estimation algorithm. A MMSE optimal distributed Kalman smoother was developed, which offers anytime optimal state estimates for stationary and nonstationary random signals. As corroborated by simulations, the distributed Kalman smoother is flexible to trade off estimation delay for MSE reduction, while exhibiting noise resilience.

Finally, the redundancy present in different observations collected at the same sensor motivated the development of channel-aware algorithms for estimating state processes based on reduced-dimensionality data collected at power-limited sensors. Both fusion center-based WSNs as well as ad hoc topologies were considered. Linear dimensionality-reducing matrices were derived at the fusion center to account for the sensors' limited power and noisy links as well as to minimize the estimator's MSE. Further, distributed Kalman filtering schemes for ad hoc topologies were developed to achieve resilience to fusion center failures and save transmission power. State-estimation schemes for fusion center-based and ad hoc topologies possess complementary strengths. Fusion center-based filters achieve higher estimation accuracy, while their ad hoc counterparts gain in robustness and power savings.

ACKNOWLEDGMENTS

This research was supported in part by NSF Award 0931239, USDoD ARO Grant W911NF-05-1-0283, and also through collaborative participation in the C&N Consortium sponsored by the U.S. ARL under the CTA Program, Cooperative Agreement DAAD19-01-2-0011. The authors would like to thank Daniel Stilwell and Darren Maczka at Virginia Tech for providing the autonomous underwater vehicle experimental data; Esha Nerurkar and Nikolas Trawny at the University of Minnesota for their help with the iterative extended SOI-KF experimental validation; and Hao Zhu for providing the scripts to generate Figure 16.

AUTHOR INFORMATION

Alejandro Ribeiro (aribeiro@seas.upenn.edu) has been an assistant professor at the Department of Electrical and Systems Engineering at the University of Pennsylvania, Philadelphia, since 2008. He received the B.Sc. in electrical engineering from the Universidad de la Republica Oriental del Uruguay, Montevideo, in 1998. From 2003 to 2008 he was at the Department of Electrical and Computer Engineering, the University of Minnesota, Minneapolis, where he received the M.Sc. and Ph.D. in electrical engineering. From 1998 to 2003 he was a member of the technical staff at Bellsouth Montevideo. His research interests lie in the areas of communication, signal processing, and networking. His current research focuses on wireless communications and networking, distributed signal processing, and wireless sensor networks. He is a Fulbright scholar and received best student paper awards at ICASSP 2005 and ICASSP 2006. He can be contacted at Electrical and Systems Engineering, University of Pennsylvania, Room 203 Moore Building, 200 South 33rd Street, Philadelphia, PA 19104 USA.

Ioannis D. Schizas received the diploma in computer engineering and informatics (with honors) from the University of Patras, Greece, in 2004 and the M.Sc. in electrical and computer engineering from the University of Minnesota, Minneapolis, in 2007. Since August 2004, he has been working toward the Ph.D. in the Department of Electrical and Computer Engineering, the University of Minnesota, Minneapolis. His research interests lie in the areas of communication theory, signal processing, and networking. His current research focuses on distributed signal processing with wireless sensor networks as well as distributed compression and source coding.

Stergios I. Roumeliotis received the diploma in electrical engineering from the National Technical University of Athens, Greece, in 1995 and the M.S. and Ph.D. in electrical engineering from the University of Southern California in 1999 and 2000, respectively. From 2000 to 2002 he was a postdoctoral fellow at the California Institute of Technology. Since 2002 he has been with the Department of Computer Science and Engineering at the University of Minnesota, where he is currently an associate professor. He is the recipient of the NSF PECASE award, the McKnight Land-Grant Professorship award, and a NASA Tech Briefs award. His research interests include inertial navigation of aerial and ground autonomous vehicles, fault detection and identification, and sensor networks. Recently, his research has focused on distributed estimation under communication and processing constraints as well as active sensing for reconfigurable networks of mobile sensors.

Georgios B. Giannakis received the diploma in electrical engineering from the National Technical University of Athens, Greece, in 1981. From 1982 to 1986 he was with the University of Southern California, where he received the M.Sc. in electrical engineering, the M.Sc. in mathematics, and the Ph.D. in electrical engineering in 1983, 1986, and 1986, respectively. Since 1999 he has been a professor at the University of Minnesota, where he holds an ADC Chair in Wireless Telecommunications in the Department of Electrical and Computer Engineering and serves as the director of

the Digital Technology Center. His interests span communications, networking, and statistical signal processing. He has published more than 275 journal papers, 450 conference papers, two edited books, and two research monographs. His current research focuses on complex-field and network coding, cooperative wireless communications, cognitive radios, cross-layer designs, mobile ad hoc networks, and wireless sensor networks. He is the corecipient of six paper awards from the IEEE Signal Processing and Communications Societies, including the G. Marconi Prize Paper Award in Wireless Communications. He received Technical Achievement Awards from the Signal Processing Society (2000), from EURASIP (2005), a Young Faculty Teaching Award, and the G.W. Taylor Award for Distinguished Research from the University of Minnesota.

REFERENCES

[1] R. E. Kalman, "A new approach to linear filtering and prediction problems," *Trans. ASME, J. Basic Eng. (Series D)*, vol. 82, pp. 35–45, 1960.

[2] Z. Q. Luo, M. Gastpar, J. Liu, and A. Swami, Eds., *IEEE Signal Processing Mag. (Special Issue on Distributed Signal Processing in Sensor Networks)*, vol. 23, no. 4, July 2006.

[3] K. Romer and F. Mattern, "The design space of wireless sensor networks," *IEEE Trans. Wireless Commun.*, vol. 11, no. 6, pp. 54–61, Dec. 2004.

[4] A. Mainwaring, D. Culler, J. Polastre, R. Szewczyk, and J. Anderson, "Wireless sensor networks for habitat monitoring," in *Proc. 1st ACM Int. Workshop Wireless Sensor Networks and Applications*, Atlanta, GA, Sept. 2002, vol. 3, pp. 88–97.

[5] S. Julier and J. Uhlmann, "Unscented filtering and nonlinear estimation," *Proc. IEEE*, vol. 92, no. 3, pp. 401–422, Mar. 2004.

[6] P. Djuric, J. Kotecha, J. Zhang, Y. Huang, T. Ghirmai, M. Bugallo, and J. Miguez, "Particle filtering," *IEEE Signal Processing Mag.*, vol. 20, no. 5, pp. 19–38, Sept. 2003.

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

[8] E. J. Msechu, S. I. Roumeliotis, A. Ribeiro, and G. B. Giannakis, "Decentralized quantized Kalman filtering with scalable communication cost," *IEEE Trans. Signal Processing*, vol. 56, no. 8, pp. 3727–3741, Aug. 2008.

[9] J. Kotecha and P. Djuric, "Gaussian particle filtering," *IEEE Trans. Signal Processing*, vol. 51, no. 10, pp. 2602–2612, Oct. 2003.

[10] P. Djuric, M. Vemula, and M. Bugallo, "Tracking with particle filtering in tertiary wireless sensor networks," in *Proc. Int. Conf. Acoustics, Speech, Signal Processing*, Philadelphia, PA, Mar. 2005, vol. 4, pp. 757–760.

[11] F. Gustafsson, F. Gunnarsson, N. Bergman, U. Forssell, J. Jansson, R. Karlsson, and P.-J. Nordlund, "Particle filters for positioning, navigation, and tracking," *IEEE Trans. Signal Processing*, vol. 50, no. 2, pp. 425–437, Feb. 2002.

[12] H. Zhu, I. D. Schizas, and G. B. Giannakis, "Power efficient dimensionality reduction for distributed channel-aware Kalman tracking using WSNs," *IEEE Trans. Signal Processing*, vol. 57, no. 8, pp. 3193–3207, Aug. 2009.

[13] S. I. Roumeliotis and G. A. Bekey, "Distributed multi-robot localization," *IEEE Trans. Robot. Automat.*, vol. 18, no. 5, pp. 781–795, Oct. 2002.

[14] D. K. Maczka and D. J. Stilwell, "Experiments in distributed navigation for AUV platoons," in *Proc. IEEE/MTS OCEANS*, Vancouver, BC, Sept. 2007, pp. 141–146.

[15] D. P. Spanos, R. Olfati-Saber, and R. J. Murray, "Approximate distributed Kalman filtering in sensor networks with quantifiable performance," in *Proc. 4th Int. Symp. Information Processing in Sensor Networks*, Los Angeles, CA, Apr. 2005, pp. 133–139.

[16] R. Olfati-Saber, "Distributed Kalman filter with embedded consensus filters," in *Proc. 44th IEEE Conf. Decision and Control*, Seville, Spain, Dec. 2005, pp. 8179–8184.

[17] M. Alanyali and V. Saligrama, "Distributed tracking in multi-hop networks with communication delays and packet losses," in *Proc. 13th IEEE Workshop Statistical Signal Processing*, Bordeaux, France, July 2005, pp. 1190–1195.

[18] U. A. Khan and J. M. F. Moura, "Distributed Kalman filters in sensor networks: Bipartite fusion graphs," in *Proc. 15th IEEE Workshop Statistical Signal Processing*, Madison, WI, Aug. 2007, pp. 700–704.

[19] P. Alricksson and A. Rantzer, "Distributed Kalman filtering using weighted averaging," in *Proc. 17th Int. Symp. Mathematical Theory of Networks and Systems*, Kyoto, Japan, July 2006, pp. 134–140.

[20] A. Jadbabaie, J. Lin, and S. Morse, "Coordination of groups of mobile autonomous agents using nearest neighbor rules," *IEEE Trans. Automat. Contr.*, vol. 48, no. 6, pp. 988–1001, June 2003.

[21] R. Olfati-Saber and R. M. Murray, "Consensus problems in networks of agents with switching topology and time-delays," *IEEE Trans. Automat. Contr.*, vol. 49, no. 9, pp. 1520–1533, Sept. 2004.

[22] L. Xiao and S. Boyd, "Fast linear iterations for distributed averaging," *Syst. Control Lett.*, vol. 53, no. 9, pp. 65–78, Sept. 2004.

[23] L. Xiao, S. Boyd, and S.-J. Kim, "Distributed average consensus with least mean-square deviation," *J. Parallel Distrib. Comput.*, vol. 67, no. 1, pp. 33–46, Jan. 2007.

[24] I. D. Schizas, G. B. Giannakis, S. I. Roumeliotis, and A. Ribeiro, "Consensus in ad hoc WSNs with noisy links—Part II: Distributed estimation and smoothing of random signals," *IEEE Trans. Signal Processing*, vol. 56, no. 4, pp. 1650–1666, Apr. 2008.

[25] B. D. Anderson and J. B. Moore, *Optimal Filtering*. Englewood Cliffs, NJ: Prentice Hall, 1979.

[26] D. Spanos, R. Olfati-Saber, and R. J. Murray, "Distributed sensor fusion using dynamic consensus," in *Proc. 16th IFAC*, Prague, Czech Republic, 2005, vol. 1, pp. 423–428.

[27] R. Olfati-Saber and J. S. Shamma, "Consensus filters for sensor networks and distributed sensor fusion," in *Proc. 44th IEEE Conf. Decision and Control*, Seville, Spain, Dec. 2005, pp. 6698–6703.

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

[29] D. Bertsekas and J. Tsitsiklis, *Parallel and Distributed Computation: Numerical Methods*, 2nd ed. Nashua, NH: Athena-Scientific, 1999.

[30] J. G. Proakis and M. Salehi, *Digital Communications*, 5th ed. New York, NY: McGraw-Hill, 2008.

[31] H. Chen, K. Zhang, and X. R. Li, "Optimal data compression for multisensor target tracking with communication constraints," in *Proc. Conf. Decision and Control*, Bahamas, Dec. 2004, vol. 3, pp. 2650–2655.

[32] M. Kisialiou and Z. Q. Luo, "Reducing power consumption in a sensor network by information feedback," in *Proc. European Signal Processing Conf.*, Florence, Italy, Sept. 2006, vol. 1, pp. 221–225.

[33] Z.-Q. Luo, G. B. Giannakis, and S. Zhang, "Optimal linear decentralized estimation in a bandwidth constrained sensor network," in *Proc. Int. Symp. Information Theory*, Adelaide, S.A., Sept. 2005, pp. 1441–1445.

[34] S. Benedetto and E. Biglieri, *Principles of Digital Transmission with Wireless Application*. Norwell, MA: Kluwer, 1999.

[35] F. Zhao, J. Shin, and J. Reich, "Information-driven dynamic sensor collaboration," *IEEE Signal Processing Mag.*, vol. 19, no. 2, pp. 61–72, Mar. 2002.

[36] A. I. Mourikis and S. I. Roumeliotis, "Optimal sensing strategies for mobile robot formations: Resource constrained localization," in *Proc. Robotics: Science and Systems Conf.*, Cambridge, MA, June 2005, pp. 281–288.

[37] V. Gupta, T. Chung, B. Hassibi, and R. M. Murray, "On a stochastic sensor selection algorithm with applications in sensor scheduling and dynamic sensor coverage," *Automatica*, vol. 42, no. 2, pp. 251–260, Feb. 2006.

[38] T. H. Chung, V. Gupta, B. Hassibi, J. Burdick, and R. M. Murray, "Scheduling for distributed sensor networks with single sensor measurement per time step," in *Proc. Int. Conf. Robotics and Automation*, New Orleans, LA, 2004, vol. 1, pp. 187–192.

[39] J. Aslam, Z. Butler, F. Constantin, V. Crespi, G. Cybenko, and D. Rus, "Tracking a moving object with a binary sensor network," in *Proc. 1st Int. Conf. Embedded Networked Sensor Systems*, Los Angeles, CA, 2003, pp. 150–161.