Consensus-Based Distributed Filtering for GNSS

Amir Khodabandeh, Peter J.G. Teunissen and
Safoora Zaminpardaz

Additional information is available at the end of the chapter

Abstract

Kalman filtering in its distributed information form is reviewed and applied to a network of receivers tracking Global Navigation Satellite Systems (GNSS). We show, by employing consensus-based data-fusion rules between GNSS receivers, how the consensus-based Kalman filter (CKF) of individual receivers can deliver GNSS parameter solutions that have a comparable precision performance as their network-derived, fusion center dependent counterparts. This is relevant as in the near future the proliferation of low-cost receivers will give rise to a significant increase in the number of GNSS users. With the CKF or other distributed filtering techniques, GNSS users can therefore achieve high-precision solutions without the need of relying on a centralized computing center.

Keywords: distributed filtering, consensus-based Kalman filter (CKF), global navigation satellite systems (GNSS), GNSS networks, GNSS ionospheric observables

1. Introduction

Kalman filtering in its decentralized and distributed forms has received increasing attention in the sensor network community and has been extensively studied in recent years, see e.g. [1–8]. While in the traditional centralized Kalman filter setup all sensor nodes have to send their measurements to a computing (fusion) center to obtain the state estimate, in the distributed filtering schemes the nodes only share limited information with their neighboring nodes (i.e. a subset of all other nodes) and yet obtain state estimates that are comparable to that of the centralized filter in a minimum-mean-squared-error sense. This particular feature of the distributed filters would potentially make the data communication between the nodes cost-effective and develop the nodes' capacity to perform *parallel* computations.

Next to sensor networks, distributed filtering can therefore benefit several other applications such as formation flying of aerial vehicles [9], cooperative robotics [10] and disciplines that concern the Global Navigation Satellite Systems (GNSS). The latter is the topic of this present

contribution. The GNSS have been proven to be an efficient tool for determination of time varying parameters that are of importance for Earth science disciplines like positioning, deformation, timing and atmosphere [11, 12]. Parameter estimation in GNSS often relies on the data processing of a *network* of receivers that collect measurements from visible GNSS satellites. In the context of sensor networks, GNSS network receivers therefore serve as sensor nodes, providing their data to a computing center, thereby computing network-based parameter solutions in a (near) real-time manner. In this contribution we intend to demonstrate how consensus algorithms [13] and the corresponding consensus-based Kalman filter (CKF), as a popular means for distributed filtering, can take an important role in GNSS applications for which a network of receivers are to be processed. Although each single receiver can run its own local filter to deliver GNSS-derived solutions, the precision of such single-receiver solutions is generally much lower than its network-derived counterparts, see e.g. [14, 15]. It will be shown, through a CKF setup, that single-receiver parameter solutions can achieve precision performances similar to that of their network-based versions, provided that a sufficient number of iterative communications between the neighboring receivers are established. The importance of such consensus-based single-receiver solutions is well appreciated in the light of the recent development of new GNSS constellations as well as the proliferation of low-cost mass-market receivers [16–18]. With the increase in the number and types of GNSS receivers, many more GNSS users can establish their own measurement setup to determine parameters that suit their needs. By taking recourse to the CKF or other distributed filtering techniques, GNSS users can therefore potentially deliver high-precision parameter solutions without the need of having a computing center.

The structure of this contribution is as follows. We first briefly review the principles of the standard Kalman filter and its information form in Section 2. The additivity property of the information filter that makes this filter particularly useful for distributed processing is also highlighted. In Section 3 we discuss average consensus rules on which the sensor nodes agree to fuse each other information. Different consensus protocols are discussed and a 'probabilistic' measure for the evaluation of their convergence rates is proposed. Section 4 is devoted to the CKF algorithmic steps. Its two time-scale nature is remarked and a three-step recursion for evaluating the consensus-based error variance matrix is developed. In Section 5 we apply the CKF theory to a small-scale network of GNSS receivers collecting ionospheric observables over time. Conducting a precision analysis, we compare the precision of the network-based ionospheric solutions with those of their single-receiver and consensus-based counterparts. It is shown how the CKF of each receiver responses to an increase in the number of iterative communications between the neighboring nodes. Concluding remarks and future outlook are provided in Section 6.

2. Kalman filtering

Consider a time series of observable random vectors $y_1, ..., y_t$. The goal is to *predict* the unobservable random state-vectors $x_1, ..., x_t$. By the term 'prediction', we mean that the observables $y_1, ..., y_t$ are used to estimate *realizations* of the random vectors $x_1, ..., x_t$. Accordingly, the

means of the state-vectors $x_1, \ldots, x_t$ can be known, while their unknown realizations still need to be guessed (predicted) through observed realizations of $y_1, \ldots, y_t$. In the following, to show on which set of observables prediction is based, we use the notation $\widehat{x}_{t|\tau}$ as the predictor of x_t when based on $y_{[\tau]} = [y_1^T, \ldots, y_\tau^T]^T$. The expectation, covariance and dispersion operators are denoted by E(.), C(.,.) and D(.), respectively. The capital Q is reserved for (co)variance matrices. Thus $C(x_t, y_\tau) = Q_{x_t y_\tau}$.

2.1. The Kalman filter standard assumptions

To predict the state-vectors in an optimal sense, one often uses the minimum mean squared error (MMSE) principle as the optimality criterion, see e.g., [19–25]. In case no restrictions are placed on the class of predictors, the MMSE predictor $\widehat{x}_{t|\tau}$ is given by the conditional mean $E\left(x_t|y_{[\tau]}\right)$, known as the Best Predictor (BP). The BP is unbiased, but generally nonlinear, with exemptions, for instance in the Gaussian case. In case x_t and $y_{[\tau]}$ are jointly Gaussian, the BP becomes linear and identical to its linear counterpart, i.e. the Best Linear Predictor (BLP)

$$\widehat{x}_{t|\tau} = E(x_t) + Q_{x_t y_{[\tau]}} Q_{y_{[\tau]} y_{[\tau]}}^{-1} \left\{ y_{[\tau]} - E\left(y_{[\tau]}\right) \right\} \tag{1}$$

Eq. (1) implies that (1) the BLP is unbiased, i.e. $E(\widehat{x}_{t|\tau}) = E(x_t)$, and that (2) the prediction error of a BLP is always uncorrelated with observables on which the BLP is based, i.e. $C(x_t - \widehat{x}_{t|\tau}, y_{[\tau]}) = 0$. These two basic properties can be alternatively used to uniquely specify a BLP [26].

The Kalman filter is a *recursive* BP (Gaussian case) or a recursive BLP. A recursive predictor, say $\widehat{x}_{t|t}$, can be obtained from the previous predictor $\widehat{x}_{t|t-1}$ and the newly collected observable vector y_t. Recursive prediction is thus very suitable for applications that require *real-time* determination of temporally varying parameters. We now state the standard assumptions that make the Kalman filter recursion feasible.

The dynamic model: The linear dynamic model, describing the time-evolution of the state-vectors x_t, is given as

$$x_t = \Phi_{t, t-1} x_{t-1} + d_t, \quad t = 1, 2, \ldots \tag{2}$$

with

$$E(x_0) = x_{0|0}, \quad D(x_0) = Q_{x_0 x_0} \tag{3}$$

and

$$E(d_t) = 0, \quad C(d_t, d_s) = S_t \delta_{t, s}, \quad C(d_t, x_0) = 0 \tag{4}$$

for the time instances $t, s = 1, 2, \ldots$, with $\delta_{t, s}$ being the Kronecker delta. The nonsingular matrix $\Phi_{t, t-1}$ denotes the transition matrix and the random vector d_t is the system noise. The system noise d_t is thus assumed to have a zero mean, to be uncorrelated in time and to be uncorrelated

with the initial state-vector x_0. The transition matrix from epoch s to t is denoted as $\Phi_{t,s}$. Thus $\Phi_{t,s}^{-1} = \Phi_{s,t}$ and $\Phi_{t,t} = I$ (the identity matrix).

The measurement model: The link between the observables y_t and the state-vectors x_t is assumed given as

$$y_t = A_t x_t + \varepsilon_t, \quad t = 1, 2, \ldots \tag{5}$$

with

$$\mathrm{E}(\varepsilon_t) = 0, \quad \mathrm{C}(\varepsilon_t, \varepsilon_s) = R_t \delta_{t,s}, \quad \mathrm{C}(\varepsilon_t, x_0) = 0, \quad \mathrm{C}(\varepsilon_t, d_s) = 0 \tag{6}$$

for $t, s = 1, 2, \ldots$, with A_t being the known design matrix. Thus the zero-mean measurement noise ε_t is assumed to be uncorrelated in time and to be uncorrelated with the initial state-vector x_0 and the system noise d_t.

2.2. The three-step recursion

Initialization: As the mean of x_0 is known, the best predictor of x_0 in the absence of data is the mean $\mathrm{E}(x_0) = x_{0|0}$. Hence, the initialization is simply given by

$$\widehat{x}_{0|0} = x_{0|0}, \quad P_{0|0} = Q_{x_0 x_0} \tag{7}$$

That the initial error variance matrix $P_{0|0} = \mathrm{D}(x_0 - \widehat{x}_{0|0})$ is identical to the variance matrix $Q_{x_0 x_0}$ follows from the equality $\mathrm{D}(x_0 - x_{0|0}) = \mathrm{D}(x_0)$.

Time update: Let us choose $\Phi_{t,t-1}\widehat{x}_{t-1|t-1}$ as a candidate for the BLP $\widehat{x}_{t|t-1}$. According to Eq. (1), the candidate would be the BLP if it fulfills two conditions: (1) it must be unbiased and (2) it must have a prediction error uncorrelated with the previous data $y_{[t-1]}$. The first condition, i.e. $\mathrm{E}(\Phi_{t,t-1}\widehat{x}_{t-1|t-1}) = \mathrm{E}(x_t)$, follows from Eq. (2) and the equalities $\mathrm{E}(\widehat{x}_{t-1|t-1}) = \mathrm{E}(x_{t-1})$ and $\mathrm{E}(d_t) = 0$. The second condition, i.e. $\mathrm{C}(x_t - \Phi_{t,t-1}\widehat{x}_{t-1|t-1}, y_{[t-1]}) = 0$, follows from the fact that the prediction error $x_t - \Phi_{t,t-1}\widehat{x}_{t-1|t-1}$ is a function of the previous BLP prediction error $x_{t-1} - \widehat{x}_{t-1|t-1}$ and the system noise d_t, i.e. (cf. 2)

$$x_t - \Phi_{t,t-1}\widehat{x}_{t-1|t-1} = \Phi_{t,t-1}(x_{t-1} - \widehat{x}_{t-1|t-1}) + d_t, \tag{8}$$

that are both uncorrelated with the previous data $y_{[t-1]}$. Hence, the time update is given by

$$\widehat{x}_{t|t-1} = \Phi_{t,t-1}\widehat{x}_{t-1|t-1}, \quad P_{t|t-1} = \Phi_{t,t-1}P_{t-1|t-1}\Phi_{t,t-1}^T + S_t \tag{9}$$

The error variance matrix $P_{t|t-1} = \mathrm{D}(x_t - \widehat{x}_{t|t-1})$ follows by applying the covariance propagation law to (8), together with $\mathrm{C}(x_{t-1} - \widehat{x}_{t-1|t-1}, d_t) = 0$.

Measurement update: In the presence of new data y_t, one may yet offer $\widehat{x}_{t|t-1}$ as a candidate for the BLP $\widehat{x}_{t|t}$. Such a candidate fulfills the unbiasedness condition $\mathrm{E}(\widehat{x}_{t|t-1}) = \mathrm{E}(x_t)$, but *not* necessarily the zero-correlation condition, that is, $\mathrm{C}(x_t - \widehat{x}_{t|t-1}, y_{[t]}) \neq 0$. Note, however, that $\mathrm{C}(x_t - \widehat{x}_{t|t-1}, y_{[t-1]}) = 0$. Thus the zero-correlation condition $\mathrm{C}(x_t - \widehat{x}_{t|t-1}, y_{[t]}) = 0$ would have been met if the most recent data y_t of $y_{[t]} = \left[y_{[t-1]}^T, y_t^T \right]^T$ would be a function of the previous data $y_{[t-1]}$, thereby *fully* predicted by $y_{[t-1]}$. Since an observable is its own best predictor, this implies that $y_t = A_t \widehat{x}_{t|t-1}$, where $A_t \widehat{x}_{t|t-1}$ is the BLP of y_t. But this would require the zero-mean quantity $v_t = y_t - A_t \widehat{x}_{t|t-1}$ to be identically zero which is generally not the case. It is therefore the presence of v_t that violates the zero-correlation condition. Note that v_t is a function of the prediction error $x_t - \widehat{x}_{t|t-1}$ and the measurement noise ε_t, i.e. (cf. 5)

$$v_t = A_t(x_t - \widehat{x}_{t|t-1}) + \varepsilon_t, \tag{10}$$

that are both uncorrelated with $y_{[t-1]}$. Therefore, v_t cannot be predicted by the previous data $y_{[t-1]}$, showing that v_t contains truly *new* information. That is why v_t is sometimes referred to as the *innovation* of y_t, see e.g. [27–29]. We now amend our earlier candidate $\widehat{x}_{t|t-1}$ by adding a linear function of v_t to it. It reads $\widehat{x}_{t|t} = \widehat{x}_{t|t-1} + K_t v_t$, with K_t being an unknown matrix to be chosen such that the zero-correlation condition is met. Such a matrix, known as the Kalman gain matrix, is uniquely specified by

$$K_t = P_{t|t-1} A_t^T Q_{v_t v_t}^{-1} \quad \Leftrightarrow \quad \mathrm{C}(x_t - \widehat{x}_{t|t-1} - K_t v_t, y_t) = 0 \tag{11}$$

since $\mathrm{C}(x_t - \widehat{x}_{t|t-1}, y_t) = P_{t|t-1} A_t^T$ and $\mathrm{C}(v_t, y_t) = Q_{v_t v_t}$. The measurement update reads then

$$\widehat{x}_{t|t} = \widehat{x}_{t|t-1} + K_t v_t, \quad \text{with} \quad P_{t|t} = P_{t|t-1} - K_t Q_{v_t v_t} K_t^T \tag{12}$$

The error variance matrix $P_{t|t} = \mathrm{D}(x_t - \widehat{x}_{t|t})$ follows by an application of the covariance propagation law, together with $\mathrm{C}(x_t - \widehat{x}_{t|t-1}, v_t) = P_{t|t-1} A_t^T$. Application of the covariance propagation law to (10) gives the variance matrix of v_t as follows

$$Q_{v_t v_t} = A_t P_{t|t-1} A_t^T + R_t \tag{13}$$

since $\mathrm{C}(x_t - \widehat{x}_{t|t-1}, \varepsilon_t) = 0$.

2.3. A remark on the filter initialization

In the derivation of the Kalman filter one assumes the mean of the random initial state-vector x_0, in Eq. (3), to be known, see e.g. [30–37]. This is because of the BLP structure (1) that needs knowledge of the means $\mathrm{E}(x_t)$ and $\mathrm{E}(y_{[\tau]})$. Since in many, if not most, applications the means of the state-vectors $x_1, \ldots, x_t$ are unknown, such derivation is therefore not appropriate. As shown in Ref. [38], one can do away with this need to have both the initial mean $x_{0|0}$ and variance

matrix $Q_{x_0 x_0}$, given in Eq. (3), known. The corresponding three-step recursion would then follow the Best Linear Unbiased Prediction (BLUP) principle and not that of the BLP. The BLUP is also a MMSE predictor, but within a more restrictive class of predictors. It replaces the means $E(x_t)$ and $E(y_{[\tau]})$ by their corresponding Best Linear Unbiased Estimators (BLUEs). Within such BLUP recursion, the initialization Eq. (7) is revised and takes place at time instance $t = 1$ in the presence of the data y_1. Provided that matrix A_1 is of full column rank, the predictor $\widehat{x}_{1|1}$ follows from solving the normal equations

$$N_1 \widehat{x}_{1|1} = r_1, \quad \text{with} \quad N_1 = A_1^T R_1^{-1} A_1, \quad r_1 = A_1^T R_1^{-1} y_1 \tag{14}$$

Thus

$$\widehat{x}_{1|1} = N_1^{-1} r_1, \quad \text{and} \quad P_{1|1} = N_1^{-1} \tag{15}$$

The above error variance matrix $P_{1|1}$ is thus *not* dependent on the variance matrix of x_1, i.e. $Q_{x_1 x_1} = \Phi_{1,0} Q_{x_0 x_0} \Phi_{1,0}^T + S_1$. This is, however, not the case with the variance matrix of the predictor $\widehat{x}_{1|1}$ itself, i.e. $Q_{\widehat{x}_{1|1} \widehat{x}_{1|1}} = D(\widehat{x}_{1|1})$. This variance matrix is given by [38]

$$Q_{\widehat{x}_{1|1} \widehat{x}_{1|1}} = Q_{x_1 x_1} + P_{1|1} \tag{16}$$

showing that $P_{1|1} \neq Q_{\widehat{x}_{1|1} \widehat{x}_{1|1}}$. Matrices $P_{t|t}$ and $Q_{\widehat{x}_{t|t} \widehat{x}_{t|t}}$ $(t = 1, 2, \ldots)$ are used for two *different purposes*. The error variance matrix $P_{t|t} = D(x_t - \widehat{x}_{t|t})$ is a measure of 'closeness' of $\widehat{x}_{t|t}$ to its target random vector x_t, thereby meant to describe the 'quality' of prediction, i.e. precision of the prediction error $(x_t - \widehat{x}_{t|t})$. The variance matrix $Q_{\widehat{x}_{t|t} \widehat{x}_{t|t}} = D(\widehat{x}_{t|t})$ however, is a measure of closeness of $\widehat{x}_{t|t}$ to the nonrandom vector $E(x_t)$, as $D(\widehat{x}_{t|t}) = D(E(x_t) - \widehat{x}_{t|t})$. Thus $Q_{\widehat{x}_{t|t} \widehat{x}_{t|t}}$ does *not* describe the quality of prediction, but instead the precision of the predictor $\widehat{x}_{t|t}$.

The MMSE of the BLUP recursion is never smaller than that of the Kalman filter, as the Kalman filter makes use of additional information, namely, the known mean $x_{0|0}$ and variance matrix $Q_{x_0 x_0}$. When the stated information is available, the BLUP recursion is shown to encompass the Kalman filter as a special case [39]. In the following we therefore assume that the means of the state-vectors $x_1, \ldots, x_t$ are unknown, a situation that often applies to GNSS applications.

2.4. Filtering in information form

The three-step recursion presented in Eqs. (7), (9) and (12) concerns the time-evolution of the predictor $\widehat{x}_{t|t}$ and the error variance matrix $P_{t|t}$. As shown in Eq. (15), both $P_{1|1}$ and $\widehat{x}_{1|1}$ can be determined by the normal matrix $N_1 = P_{1|1}^{-1}$ and the right-hand-side vector $r_1 = P_{1|1}^{-1} \widehat{x}_{1|1}$. One can therefore alternatively develop recursion concerning the time-evolution of $P_{t|t}^{-1}$ and $P_{t|t}^{-1} \widehat{x}_{t|t}$. From a computational point of view, such recursion is found to be very suitable when the inverse-variance or *information* matrices S_t^{-1} and R_t^{-1} serve as input rather than the variance matrices S_t and R_t. To that end, one may define [34]

$$\text{information vector} \quad i_{t|\tau} := P_{t|\tau}^{-1}\widehat{x}_{t|\tau} \quad \text{and} \quad \text{information matrix} \ \mathcal{I}_{t|\tau} := P_{t|\tau}^{-1} \qquad (17)$$

Given the definition above, the information filter recursion concerning the time-evolution of $i_{t|t}$ and $\mathcal{I}_{t|t}$ would then follow from the recursion Eqs. (15), (9) and (12), along with the following matrix-inversion equalities

$$
\begin{aligned}
\text{Time-update:} & \quad (\Phi_{t,t-1}P_{t-1|t-1}\Phi_{t,t-1}^T + S_t)^{-1} & = & \ M_t - M_t(M_t + S_t^{-1})^{-1}M_t \\
\text{Measurement-update:} & \quad (P_{t|t-1} - P_{t|t-1}A_t^T Q_{v_tv_t}A_t P_{t|t-1})^{-1} & = & \ P_{t|t-1}^{-1} + A_t^T R_t^{-1} A_t
\end{aligned}
\qquad (18)
$$

where $M_t = \Phi_{t-1,t}^T P_{t-1|t-1}^{-1}\Phi_{t-1,t}$.

The algorithmic steps of the information filter are presented in **Figure 1**. In the absence of data, the filter is initialized by the zero information $i_{1|0} = 0$ and $\mathcal{I}_{1|0} = 0$. In the presence of the data y_t, the corresponding normal matrix N_t and right-hand-side vector r_t are added to the time update information $i_{t|t-1}$ and $\mathcal{I}_{t|t-1}$ to obtain the measurement update information $i_{t|t}$ and $\mathcal{I}_{t|t}$.
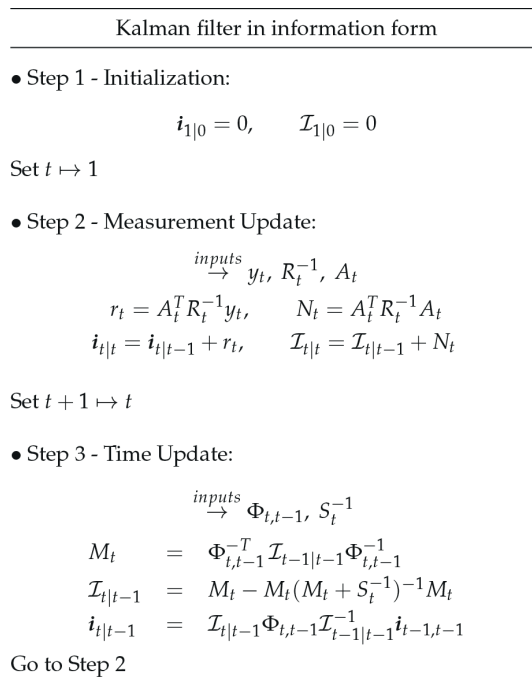

Kalman filter in information form

• Step 1 - Initialization:

$$i_{1|0} = 0, \qquad \mathcal{I}_{1|0} = 0$$

Set $t \mapsto 1$

• Step 2 - Measurement Update:

$$\overset{inputs}{\rightarrow} y_t, R_t^{-1}, A_t$$

$$r_t = A_t^T R_t^{-1} y_t, \qquad N_t = A_t^T R_t^{-1} A_t$$

$$i_{t|t} = i_{t|t-1} + r_t, \qquad \mathcal{I}_{t|t} = \mathcal{I}_{t|t-1} + N_t$$

Set $t + 1 \mapsto t$

• Step 3 - Time Update:

$$\overset{inputs}{\rightarrow} \Phi_{t,t-1}, S_t^{-1}$$

$$
\begin{aligned}
M_t & = \Phi_{t,t-1}^{-T}\mathcal{I}_{t-1|t-1}\Phi_{t,t-1}^{-1} \\
\mathcal{I}_{t|t-1} & = M_t - M_t(M_t + S_t^{-1})^{-1}M_t \\
i_{t|t-1} & = \mathcal{I}_{t|t-1}\Phi_{t,t-1}\mathcal{I}_{t-1|t-1}^{-1}i_{t-1,t-1}
\end{aligned}
$$

Go to Step 2

Figure 1. Algorithmic steps of the information filter recursion concerning the time-evolution of the information vector $i_{t|t}$ and matrix $\mathcal{I}_{t|t}$.

The transition matrix $\Phi_{t,t-1}$ and inverse-variance matrix S_t^{-1} would then be used to time update the previous information $i_{t-1|t-1}$ and $\mathcal{I}_{t-1|t-1}$.

Singular matrix S_t: In the first expression of Eq. (18) one assumes the variance matrix S_t to be nonsingular and invertible. There are, however, situations where some of the elements of the state-vector x_t are nonrandom, i.e., the corresponding system noise is identically zero. As a consequence, the variance matrix S_t becomes singular and the inverse-matrix S_t^{-1} does not exist. An example of such concerns the presence of the GNSS carrier-phase ambiguities in the filter state-vector which are treated constant in time. In such cases the information time update in **Figure 1** must be generalized so as to accommodate singular variance matrices S_t. Let $\tilde{S}_t$ be an invertible sub-matrix of S_t that has the *same* rank as that of S_t. Then there exists a full-column rank matrix H_t such that

$$S_t = H_t \tilde{S}_t H_t^T \tag{19}$$

Matrix H_t can be, for instance, structured by the columns of the identity matrix I corresponding to the columns of S_t on which the sub-matrix $\tilde{S}_t$ is positioned. The special case

$$S_t = \begin{bmatrix} \tilde{S}_t & 0 \\ 0 & 0 \end{bmatrix} = \begin{bmatrix} I \\ 0 \end{bmatrix} \tilde{S}_t \begin{bmatrix} I \\ 0 \end{bmatrix}^T \quad \Rightarrow \quad H_t = \begin{bmatrix} I \\ 0 \end{bmatrix}, \tag{20}$$

shows an example of the representation (19). With Eq. (19), a generalization of the time update (**Figure 1**) can be shown to be given by

$$\mathcal{I}_{t|t-1} = M_t - M_t H_t \left[H_t^T M_t H_t + \tilde{S}_t^{-1} \right]^{-1} H_t^T M_t \tag{21}$$

Thus instead of S_t^{-1}, the inverse-matrix $\tilde{S}_t^{-1}$ and H_t are assumed available.

2.5. Additivity property of the information measurement update

As stated previously, the information filter delivers outcomes equivalent to those of the Kalman filter recursion. Thus any particular preference for the information filter must be attributed to the *computational* effort required for obtaining the outcomes. For instance, if handling matrix inversion requires low computational complexity when working with the input inverse-matrices S_t^{-1} and R_t^{-1}, the information filter appears to be more suitable. In this subsection we will highlight yet another property of the information filter that makes this recursion particularly useful for *distributed* processing.

As shown in **Figure 1**, the information measurement update is *additive* in the sense that the measurement information N_t and r_t is added to the information states $\mathcal{I}_{t|t-1}$ and $i_{t|t-1}$. We now make a start to show how such additivity property lends itself to distributed filtering. Let the measurement model Eq. (5) be partitioned as

$$y_t = A_t x_t + \varepsilon_t \quad \Rightarrow \quad \begin{bmatrix} y_{1,t} \\ \vdots \\ y_{i,t} \\ \vdots \\ y_{n,t} \end{bmatrix} = \begin{bmatrix} A_{1,t} \\ \vdots \\ A_{i,t} \\ \vdots \\ A_{n,t} \end{bmatrix} x_t + \begin{bmatrix} \varepsilon_{1,t} \\ \vdots \\ \varepsilon_{i,t} \\ \vdots \\ \varepsilon_{n,t} \end{bmatrix}, \quad t = 1, 2, \ldots \tag{22}$$

Accordingly, the observable vector y_t is partitioned into n sub-vectors $y_{i,t}$ $(i = 1, \ldots, n)$, each having its own design matrix $A_{i,t}$ and measurement noise vector $\varepsilon_{i,t}$. One can think of a network of n sensor nodes where each collects its own observable vector $y_{i,t}$, but aiming to determine a common state-vector x_t. Let us further assume that the nodes collect observables independently from one another. This yields

$$C(\varepsilon_{i,t}, \varepsilon_{j,t}) = R_{i,t} \delta_{i,j}, \quad \text{for} \quad i, j = 1, \ldots, n, \quad \text{and} \quad t = 1, 2, \ldots \tag{23}$$

Thus the measurement noise vectors $\varepsilon_{i,t}$ $(i = 1, \ldots, n)$ are assumed to be *mutually* uncorrelated. With the extra assumption Eq. (23), the normal matrix $N_t = A_t^T R_t^{-1} A_t$ and right-hand-side vector $r_t = A_t^T R_t^{-1} y_t$ can then be, respectively, expressed as

$$N_t = \sum_{i=1}^{n} N_{i,t}, \quad \text{and} \quad r_t = \sum_{i=1}^{n} r_{i,t} \tag{24}$$

where

$$N_{i,t} = A_{i,t}^T R_{i,t}^{-1} A_{i,t}, \quad \text{and} \quad r_{i,t} = A_{i,t}^T R_{i,t}^{-1} y_{i,t} \tag{25}$$

According to Eq. (24), the measurement information of each node, say $N_{i,t}$ and $r_{i,t}$, is individually added to the information states $\mathcal{I}_{t|t-1}$ and $i_{t|t-1}$, that is

$$\mathcal{I}_{t|t} = \mathcal{I}_{t|t-1} + \sum_{i=1}^{n} N_{i,t}, \qquad i_{t|t} = i_{t|t-1} + \sum_{i=1}^{n} r_{i,t} \tag{26}$$

Now consider the situation where each node runs its own *local* information filter, thus having its own information states $\mathcal{I}_{i,t|t}$ and $i_{i,t|t}$ $(i = 1, \ldots, n)$. The task is to recursively update the local states $\mathcal{I}_{i,t|t}$ and $i_{i,t|t}$ in a way that they remain equal to their *central* counterparts $\mathcal{I}_{t|t}$ and $i_{t|t}$ given in Eq. (26). Suppose that such equalities hold at the time update, i.e. $\mathcal{I}_{i,t|t-1} = \mathcal{I}_{t|t-1}$ and $i_{i,t|t-1} = i_{t|t-1}$. Given the number of contributing nodes n, each node just needs to be provided with the *average* quantities

$$\overline{N}_t = \frac{1}{n} \sum_{i=1}^{n} N_{i,t}, \quad \text{and} \quad \overline{r}_t = \frac{1}{n} \sum_{i=1}^{n} r_{i,t} \tag{27}$$

The local states $\mathcal{I}_{i,t|t-1}$ and $i_{i,t|t-1}$ would then be measurement updated as (cf. 26)

$$\mathcal{I}_{i,t|t} = \mathcal{I}_{i,t|t-1} + n\overline{N}_t, \qquad i_{i,t|t} = i_{i,t|t-1} + n\overline{r}_t \qquad (28)$$

that are equal to the central states $\mathcal{I}_{t|t}$ and $i_{t|t}$, respectively. In this way one has multiple distributed local filters $i = 1, \ldots, n$, where each recursively delivers results identical to those of a central filter.

To compute the average quantities $\overline{N}_t$ and $\overline{r}_t$, node i may need to receive all other information $N_{j,t}$ and $r_{j,t}$ ($j \neq i$). In other words, node i would require direct connections to *all* other nodes $j \neq i$, a situation that makes data communication and processing power very expensive (particularly for a large number of nodes). In the following cheaper ways of evaluating the averages $\overline{N}_t$ and $\overline{r}_t$ are discussed.

3. Average consensus rules

In the previous section, we briefly discussed the potential applicability of the information filter as a tool for handling the measurement model Eq. (22) in a distributed manner. With the representation Eq. (28) however, one may be inclined to conclude that such applicability is limited to the case where the nodes $i = 1, \ldots, n$, have 'direct' communication connections to one another in order to receive/send their measurement information $N_{i,t}$ and $r_{i,t}$ ($i = 1, \ldots, n$).

Instead of having direct connections, the idea is now to relax such a stringent requirement by assuming that the nodes are linked to each other *at least* through a 'path' so that information can flow from each node to all other nodes. It is therefore assumed that each node along the path plays the role of an *agent* transferring information to other nodes. To reach the averages $\overline{N}_t$ and $\overline{r}_t$, the nodes would then agree on specific 'fusion rules' or *consensus protocols*, see e.g. [6, 8, 40]. Note that each node exchanges information with neighboring nodes (i.e. those to which the node has direct connections) and *not* the entire nodes. Therefore, a *repeated* application of the consensus protocols is required to be carried out. The notion is made precise below.

3.1. Communication graphs

The way the nodes interact with each other to transfer information is referred to as the interaction topology between the nodes. The interaction topology is often described by a directed graph whose vertices and edges, respectively, represent the nodes and communication links [4]. The interaction topology may also undergo a *finite* number of changes over sessions $k = 1, \ldots, k_o$. In case of one-way links, the directions of the edges face toward the receiving nodes (vertices). Here we assume that the communication links between the nodes are two-way, thus having *undirected* (or bidirectional) graphs. Examples of such representing a network of 20 nodes with their interaction links are shown in **Figure 2**. Let an undirected graph at session k be denoted by $\mathcal{G}_k = (\mathcal{V}, \mathcal{E}_k)$ where $\mathcal{V} = \{1, \ldots, n\}$ is the vertex set and $\mathcal{E}_k \subset \{(i,j) \mid i, j \in \mathcal{V}\}$ is the edge set. We assume that the nodes remain unchanged over time, that is why the subscript k is omitted for $\mathcal{V}$. This is generally not the case with their interaction
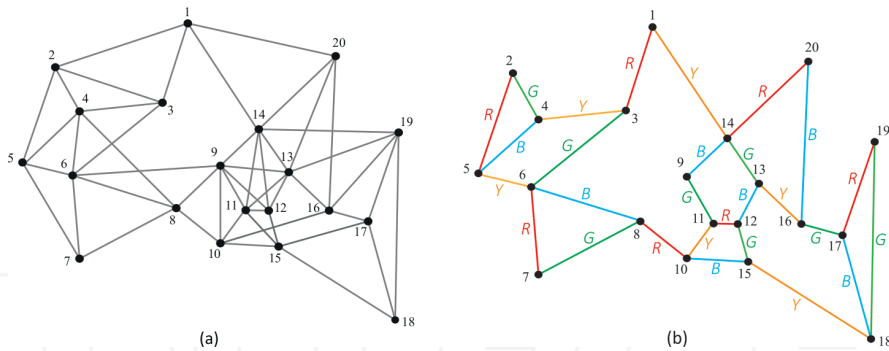
Figure 2. Communications graphs of 20 sensor nodes. The edges represent two-way communication links between the nodes. (a) Network with 49 links. (b) Network with different numbers of links over four sessions: 7 links in session 1 (R), 6 links in session 2 (Y), 8 links in session 3 (G) and 7 links in session 4 (B).

links though, i.e. the edge set $\mathcal{E}_k$ depends on k. As in **Figure 2 (b)**, the number of links between the nodes can be different for different sessions $k = 1, \ldots, k_o$. Each session represents a graph that may not be *connected*. In a 'connected' graph, every vertex is linked to all other vertices at least through one path. In order for information to flow from each node to all other nodes, the *union* of the graphs $\mathcal{G}_k$ ($k = 1, \ldots, k_o$), i.e.

$$\mathcal{G} = (\mathcal{V}, \mathcal{E}) \quad \text{with} \quad \mathcal{E} = \bigcup_{k=1}^{k_o} \mathcal{E}_k \quad (k_o : \text{a finite number}) \tag{29}$$

is therefore assumed to be connected. We define the *neighbors* of node i as those to which the node i has direct links. For every session k, they are collected in the set $\mathcal{N}_{i,k} = \{j \mid (j, i) \in \mathcal{E}_k\}$. For instance for network (a) of **Figure 2**, we have only one session, i.e. $k_o = 1$, in which $\mathcal{N}_{2,1} = \{1, 3, 4, 5\}$ represents the neighbors of node 2. In case of network (b) however, we have different links over four sessions, i.e. $k_o = 4$. In this case, the neighbors of node 2 are given by four sets: $\mathcal{N}_{2,1} = \{5\}$ in session 1 (red), $\mathcal{N}_{2,2} = \{\}$ in session 2 (yellow), $\mathcal{N}_{2,3} = \{4\}$ in session 3 (green) and $\mathcal{N}_{2,4} = \{\}$ in session 4 (blue).

3.2. Consensus protocols

Given the right-hand-vector $r_{i,t}$, suppose that node i aims to obtain the average $\bar{r}_t$ for which all other vectors $r_{j,t}$ ($\forall j \neq i$) are required to be available (cf. 27). But the node i only has access to those of its neighbors, i.e. the vectors $r_{j,t}$ ($j \in \mathcal{N}_{i,k}$). For the first session $k = 1$, it would then seem to be reasonable to compute a *weighted-average* of the available vectors, i.e.

$$r_{i,t}(1) = \sum_{j \in \{i, \mathcal{N}_{i,1}\}} w_{ij}(1) r_{j,t} \tag{30}$$

as an *approximation* of $\bar{r}_t$, where the scalars $w_{ij}(1)$ ($j \in \{i, \mathcal{N}_{i,1}\}$) denote the corresponding weights at session $k = 1$. Now assume that all other nodes $j \neq i$ agree to apply the fusion rule

Eq. (30) to those of their own neighbors. Thus the neighboring nodes $j \in \mathcal{N}_{i,k}$ also have their own weighted-averages $r_{j,t}(1)$. But they may have access to those to which the node i has *no* direct links. In other words, the weighted-averages $r_{j,t}(1)$ ($j \in \mathcal{N}_{i,1}$) contain information on the nodes to which the node i has no access. For the next session $k = 2$, it is therefore reasonable for the node i to *repeat* the fusion rule Eq. (30), but now over the new vectors $r_{j,t}(1)$ ($j \in \{i, \mathcal{N}_{i,2}\}$), aiming to improve on the earlier approximation $r_{i,t}(1)$. This yields the following iterative computations

$$r_{i,t}(k) = \sum_{j \in \{i, \mathcal{N}_{i,k}\}} w_{ij}(k) r_{j,t}(k-1), \quad k = 1, 2, \ldots \tag{31}$$

with $r_{j,t}(0) := r_{j,t}$. Choosing a set of weights $w_{ij}(k)$, the nodes $i = 1, \ldots, n$ agree on the consensus protocol (31) to iteratively fuse their information vectors $r_{i,t}(k)$. Here and in the following, we use the letter 'k' for the '*session number*' $k = 1, \ldots, k_o$ (cf. 29) and for the '*number of iterative communications*' $k = 1, \ldots, k_n$ (cf. 34). The maximum iteration k_n is assumed to be not smaller than the maximum session number k_o, i.e. $k_n \geq k_o$.

The question that now comes to the fore is how to choose the weights $w_{ij}(k)$ such that the approximation $r_{i,t}(k)$ gets close to $\bar{r}_t$ through the iteration Eq. (31). More precisely, the stated iteration becomes favorable if $r_{i,t}(k) \to \bar{r}_t$ when $k \to \infty$ for all nodes $i = 1, \ldots, n$. To address this question, we use a multivariate formulation. Let p be the size of the vectors $r_{i,t}$ ($i = 1, \ldots, n$). We define the higher-dimensional vector $r = \left[r_{1,t}^T, \ldots, r_{n,t}^T \right]^T$. The multivariate version of Eq. (31) reads then

$$\begin{bmatrix} r_{1,t}(k) \\ \vdots \\ r_{i,t}(k) \\ \vdots \\ r_{n,t}(k) \end{bmatrix} = \begin{bmatrix} w_{11}(k)I_p & \cdots & w_{1i}(k)I_p & \cdots & w_{1n}(k)I_p \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ w_{i1}(k)I_p & \cdots & w_{ii}(k)I_p & \cdots & w_{in}(k)I_p \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ w_{n1}(k)I_p & \cdots & w_{ni}(k)I_p & \cdots & w_{nn}(k)I_p \end{bmatrix} \begin{bmatrix} r_{1,t}(k-1) \\ \vdots \\ r_{i,t}(k-1) \\ \vdots \\ r_{n,t}(k-1) \end{bmatrix}, \quad k = 1, 2, \ldots \tag{32}$$

or

$$r(k) = \left[W(k) \otimes I_p \right] r(k-1), \quad k = 1, 2, \ldots \tag{33}$$

The $n \times n$ weight matrix $W(k)$ is structured by $w_{ij}(k)$ ($j \in \{i, \mathcal{N}_{i,k}\}$) and $w_{ij}(k) = 0$.

($j \notin \{i, \mathcal{N}_{i,k}\}$). The symbol $\otimes$ is the Kronecker matrix product [41]. According to Eq. (33), after k_n iterations the most recent iterated vector $r(k_n)$ is linked to the initial vector $r(0)$ by $\left[\prod_{k=1}^{k_n} W(k) \otimes I_p \right] r(0)$. Thus the vectors $r_{i,t}(k)$ ($i = 1, \ldots, n$) converge to $\bar{r}_t$ when

$$L(k_n) := \prod_{k=1}^{k_n} W(k) \to \frac{1}{n} e_n e_n^T, \quad \text{as} \quad k_n \to \infty \tag{34}$$

where the n-vector e_n contains ones. If the condition Eq. (34) is met, the set of nodes $\{1, \ldots, n\}$ can *asymptotically* reach average consensus [4]. It can be shown that (34) holds if the weight

matrices $W(k)$ $(k = 1, ..., k_o)$ have bounded nonnegative entries with positive diagonals, i.e. $w_{ij}(k) \geq 0$ and $w_{ii}(k) > 0$, having row- and column-sums equal to one, i.e. $\sum_{j=1}^{n} w_{ij}(k) = 1$ and $\sum_{i=1}^{n} w_{ij}(k) = 1$ $(i, j = 1, ..., n)$, see e.g. [3, 5, 40, 42, 43].

Examples of such consensus protocols are given in **Table 1**. As shown, the weights form a symmetric weight matrix $W(k)$, i.e. $w_{ji}(k) = w_{ij}(k)$. In all protocols presented, self-weights $w_{ii}(k)$ are chosen so that the condition $\sum_{j=1}^{n} w_{ij}(k) = 1$ is satisfied. The weights of Protocols 1 and 2 belong to the class of 'maximum-degree' weights, while those of Protocol 3 are referred to as 'Metropolis' weights [8]. The weights of Protocols 1 and 3 are driven by the degrees (number of neighbors) of nodes $i = 1, ..., n$, denoted by $dg_i(k) = \mathcal{N}_{i,k}$. For instance, in network (a) of **Figure 2** we have $dg_1(1) = 4$ as node 1 has 4 neighbors, while $dg_{14}(1) = 7$ as node 14 has 7 neighbors. Protocol 4 is only applicable to networks like (b) in **Figure 2**, i.e. when each node has *at most* one neighbor at a session [4]. In this case, each node exchanges its information to just one neighbor at a session. Thus for two neighboring nodes i and j we have $w_{ii}(k) = w_{jj}(k) = w_{ij}(k) = 0.5$, each averaging $r_{i,t}(k-1)$ and $r_{j,t}(k-1)$ to obtain $r_{i,t}(k) = r_{j,t}(k)$.

To provide insight into the applicability of the protocols given in **Table 1**, we apply them to the networks of **Figure 2**. Twenty values (scalars), say r_i $(i = 1, ..., 20)$, are generated whose average is equal to 5, i.e. $\bar{r} = 5$. Each value is assigned to its corresponding node. For network (a), Protocols 1, 2 and 3 are separately applied, whereas Protocol 4 is only applied to network (b). The corresponding results, up to 30 iterations, are presented in **Figure 3**. As shown, the iterated values $r_i(k)$ $(i = 1, ..., 20)$ get closer to their average (i.e. $\bar{r} = 5$), the more the number of iterative communications.

3.3. On convergence of consensus states

Figure 3 shows that the states $r_{i,t}(k)$ $(i = 1, ..., n)$ converge to their average $\bar{r}_t$, but with *different* rates. The convergence rate depends on the initial states $r_{i,t}(0) = r_{i,t}$ and on the consensus protocol employed. From the figure it seems that the convergence rates of Protocols 1 and 3 are about the same, higher than those of Protocols 2 and 4. Note that the stated results are

Protocols	$w_{ij}(k)$ $(\{i,j\} \in \mathcal{E}_k)$	$w_{ii}(k)$
Protocol 1	$\dfrac{1}{\max\limits_{u\ \in\ \{1,...,n\}} \{dg_u(k)\}}$	$1 - \sum_{u \neq i} w_{iu}(k)$
Protocol 2	$\frac{1}{n}$	$1 - \sum_{u \neq i} w_{iu}(k)$
Protocol 3	$\frac{1}{1+\max\{dg_i(k), dg_j(k)\}}$	$1 - \sum_{u \neq i} w_{iu}(k)$
Protocol 4	$\frac{1}{2}$	$1 - \sum_{u \neq i} w_{iu}(k)$
otherwise $w_{ij}(k) = 0$		

The degree (number of neighbors) of node i is denoted by $dg_i(k) = \#\mathcal{N}_{i,k}$. Protocol 4 is only applicable when each node has *at most* one neighbor at a session.

Table 1. Examples of average-consensus protocols forming the weights $w_{ij}(k)$ in Eq. (31).
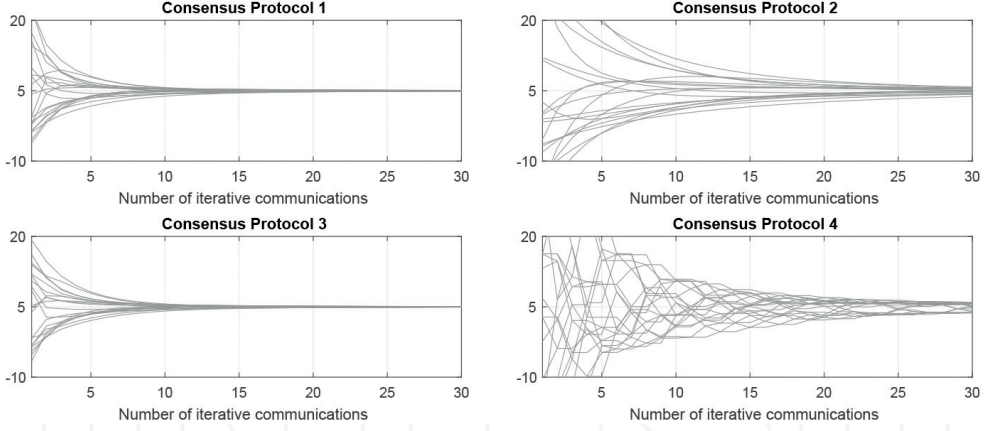
Figure 3. Performances of protocols 1, 2 and 3 (network (a) of **Figure 2**) and protocols 4 (network (b)) in delivering the average of 20 values (scalars). The iterated values get closer to their average (i.e. $\bar{r} = 5$), the more the number of iterative communications.

obtained on the basis of specific 'realizations' of $r_{i,t}$ ($i = 1, ..., n$). Consider the states $r_{i,t}$ to be *random* vectors. In that case, can the results be still representative for judging the convergence performances of the protocols? To answer this question, let us define the difference vectors $dr_{i,t}(k) = r_{i,t}(k) - \bar{r}_t$ that are collected in the higher-dimensioned vector $dr(k) = \left[dr_{1,t}^T(k), ..., dr_{n,t}^T(k) \right]^T$. The more the number of iterations, the smaller the norm of $dr(k)$ becomes. According to Eq. (33), after k_n iterations the difference vector $d_r(k_n)$ is linked to $r = \left[r_{1,t}^T, ..., r_{n,t}^T \right]^T$ through

$$dr(k_n) = \left[\left(L(k_n) - \frac{1}{n} e_n e_n^T \right) \otimes I_p \right] r \tag{35}$$

Now let the initial states $r_{i,t}$ have the same mean and the same variance matrix $D(r_{i,t}) = Q$ ($i = 1, ..., n$), but mutually uncorrelated. An application of the covariance propagation law to (35), together with $L(k_n)e_n = e_n$, gives

$$D(dr(k_n)) = \left(L^2(k_n) - \frac{1}{n} e_n e_n^T \right) \otimes Q \tag{36}$$

Thus the closer the squared matrix $L^2(k_n)$ to $(1/n)e_n e_n^T$, the smaller the variance matrix Eq. (36) becomes. In the limit when $k_n \rightarrow \infty$, the stated variance matrix tends to zero. This is what one would expect, since $dr(k_n) \rightarrow 0$. Under the conditions stated in Eq. (34), matrices $W(k)$ have $\lambda_n = 1$ as the largest absolute value of their eigenvalues [42]. A symmetric weight matrix W can then be expressed in its *spectral* form as

$$W = \sum_{i=1}^{n-1} \lambda_i u_i u_i^T + \frac{1}{n} e_n e_n^T \tag{37}$$

with the eigenvalues $\lambda_1 \leq \ldots \leq \lambda_{n-1} < \lambda_n = 1$, and the corresponding orthogonal unit eigenvectors $u_1, \ldots, u_{n-1}, u_n = (1/\sqrt{n})e_n$. By a repeated application of the protocol W, we get $L(k_n) = W^{k_n}$. Substitution into Eq. (36), together with Eq. (37), gives finally

$$D(dr(k_n)) = \left(\sum_{i=1}^{n-1} \lambda_i^{2k_n} u_i u_i^T \right) \otimes Q \leq \lambda_{n-1}^{2k_n} [I_n \otimes Q] \tag{38}$$

The above equation shows that the entries of the variance matrix (36) are largely driven by the second largest eigenvalue of W, i.e. λ_{n-1}. The smaller the scalar $|\lambda_{n-1}|$, the faster the quantity $\lambda_{n-1}^{2k_n}$ tends to zero, as $k_n \to \infty$. The scalar $|\lambda_{n-1}|$ is thus often used as a measure to judge the convergence performances of the protocols [7]. For the networks of **Figure 2**, $|\lambda_{n-1}|$ of Protocols 1, 2 and 3 are about 0.92, 0.97, 0.91, respectively. As Protocol 3 has the smallest $|\lambda_{n-1}|$, it is therefore expected to have the best performance. Note, in Protocol 4, that the weight matrix $W(k)$ varies in every session, the performance of which cannot be judged by a single eigenvalue λ_{n-1}. One can therefore think of another means of measuring the convergence performance. Due to the randomness of the information vectors $r_{i,t}$ $(i = 1, \ldots, n)$, one may propose 'probabilistic' measures such as

$$\text{Prob} \left(\max_i \| dr_{i,t}(k_n) \|_Q \leq q \right), \quad (q > 0) \tag{39}$$

to evaluate the convergence rates of the protocols, where $\| dr_{i,t} \|_Q^2 := dr_{i,t}^T Q^{-1} dr_{i,t}$. Eq. (39) refers to the probability that the maximum-norm of the difference vectors $dr_{i,t}(k_n) = r_{i,t}(k_n) - \bar{r}_t$ $(i = 1, \ldots, n)$ is not larger than a given positive scalar q for a fixed number of iterations k_n. The higher the probability Eq. (39), the better the performance of a protocol. For the scalar case $Q = \sigma^2$, Eq. (39) is reduced to

$$\text{Prob} \left(\max_i | dr_{i,t}(k_n) | \leq q\sigma \right) \tag{40}$$

which is the probability that the absolute differences $|dr_{i,t}(k_n)|$ $(i = 1, \ldots, n)$ are not larger than q times the standard-deviation σ. For the networks of **Figure 2**, 100,000 normally-distributed vectors as samples of $r = [r_1, \ldots, r_{20}]^T$ are simulated to evaluate the probability (40). The results for Protocols 1, 2, 3 and 4 are presented in **Figure 4**. The stated probability is plotted as a function of q for three numbers of iterative communications $k_n = 10, 20$ and 30. As shown, Protocol 3 gives rise to highest probabilities, while Protocol 2 delivers lowest probabilities. After 10 iterations, the probability of having absolute differences smaller than one-fifth of the standard-deviation σ (i.e. $q = 0.2$) is about 80% for Protocol 1, whereas it is less than 5% for Protocol 2. After 30 iterations, the stated probability increases to 80% for Protocol 2, but close to 100% for Protocols 1 and 3.

Figure 4 demonstrates that the convergence performance of Protocol 4 is clearly better than that of Protocol 2, as it delivers higher probabilities (for the networks of **Figure 2**). Such a conclusion however, cannot be made on the basis of the results of **Figure 3**. This shows that
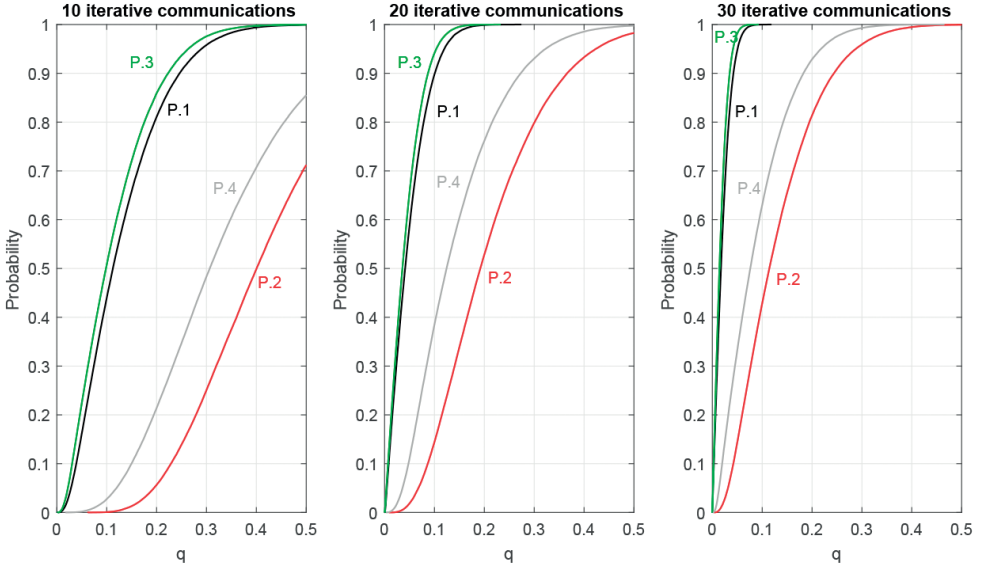
Figure 4. Probability Eq. (40) as a function of q for three numbers of iterative communications $k_n = 10, 20$ and 30 (protocols 1 (P.1), 2 (P.2), 3 (P.3) and 4 (P.4)). It refers to the probability that the absolute differences $|r_{i,t}(k_n) - \bar{r}_t|$ ($i = 1, ..., 20$) are not larger than q times the standard-deviation σ (cf. **Figure 2**).

results obtained on the basis of specific 'realizations' of $r_{i,t}$ ($i = 1, ..., n$) are not necessarily representative.

4. Consensus-based Kalman filters

4.1. Two time-scale approach

In Section 2.5 we discussed how the additivity property of the measurement update Eq. (26) offers possibilities for developing multiple distributed local filters $i = 1, ..., n$, each delivering local states $\mathcal{I}_{i,t|t}$ and $i_{i,t|t}$ equal to their central counterparts $\mathcal{I}_{t|t}$ and $i_{t|t}$. In doing so, each node has to evaluate the averages $\overline{N}_t$ and $\bar{r}_t$ at *every* time instance t. Since in practice the nodes do not necessarily have direct connections to each other, options such as the consensus-based fusion rules (cf. Section 3) can alternatively be employed to 'approximate' $\overline{N}_t$ and $\bar{r}_t$. As illustrated in **Figures 3** and **4**, such consensus-based approximation requires a number of iterative communications between the nodes in order to reach the averages $\overline{N}_t$ and $\bar{r}_t$. The stated iterative communications clearly require some time to be carried out and must take place during every time interval $[t, t + 1]$ (see **Figure 5**). We distinguish between the *sampling* rate Δ and the *sending* rate δ. The sampling rate refers to the frequency with which the node i collects its observables $y_{i,t}$ ($t = 1, 2, ...$), while the sending rate refers to the frequency with which the node i sends/receives information $N_{j,t}(k)$ and $r_{j,t}(k)$ ($k = 1, ..., k_n$) to/from its neighboring nodes. As shown in **Figure 5**, the sending rate δ should therefore be reasonably smaller
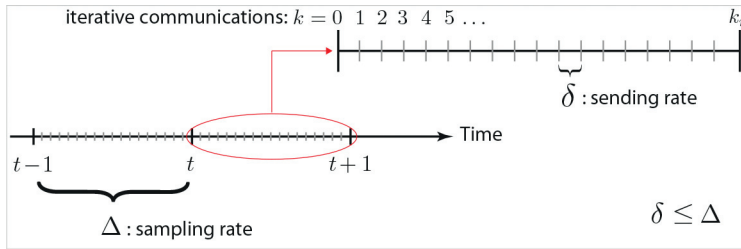
Figure 5. The two time-scale nature of a consensus-based Kalman filter (CKF): The data sampling time-scale $t = 1, 2, ...,$ versus the data sending time-scale $k = 1, ..., k_n$. The sending rate δ must be reasonably smaller than the sampling rate Δ so as to be able to incorporate consensus protocols into the CKF setup.

than the sampling rate Δ so as to be able to incorporate consensus protocols into the information filter setup. Such a consensus-based Kalman filter (CKF) would thus generally be of a two time-scale nature [2], the data sampling time-scale $t = 1, 2, ...,$ versus the data sending time-scale $k = 1, ..., k_n$. The CKF is a suitable tool for handling real-time data processing in a distributed manner for the applications in which the state-vectors x_t ($t = 1, 2, ...$) change rather slowly over time (i.e. Δ can take large values) and/or for the cases where the sensor nodes transfer their data rather quickly (i.e. δ can take small values).

Under the assumption $\delta \leq \Delta$, the CKF recursion follows from the Kalman filter recursion by considering an extra step, namely, the 'consensus update'. The algorithmic steps of the CKF in information form are presented in **Figure 6**. Compare the recursion with that of the information filter given in **Figure 1**. Similar to the information filter, the CKF at node i is initialized by the zero information $\mathcal{I}_{i, 1|0} = 0$ and $i_{i, 1|0} = 0$. In the presence of the data $y_{i, t}$, node i computes its local normal matrix $N_{i, t}$ and right-hand-side vector $r_{i, t}$ to send them to its neighboring nodes $j \in \mathcal{N}_{i,k}$ ($k = 1, ..., k_n$). In the consensus update, iterative communications between the neighboring nodes $\{i, \mathcal{N}_{i,k}\}$ are carried out to approximate the averages $\overline{N}_t$ and $\overline{r}_t$ by $N_{i, t}(k_n)$ and $r_{i, t}(k_n)$, respectively. After a finite number of communications k_n, the consensus states $N_{i, t}(k_n)$ and $r_{i, t}(k_n)$ are, respectively, added to the time update information $\mathcal{I}_{i, t|t-1}$ and $i_{i, t|t-1}$ to obtain their measurement update version $\mathcal{I}_{i, t|t}$ and $i_{i, t|t}$ at node i (cf. 28). The time update goes along the same lines as that of the information filter.

4.2. Time evolution of the CKF error covariances

With the consensus-based information filter, presented in **Figure 6**, it is therefore feasible to develop multiple distributed filters, all running in parallel over time. By taking recourse to an average-consensus protocol, not all the nodes are needed to be directly linked, thereby allowing non-neighboring nodes to also benefit from information states of each other. The price one has to pay for such an attractive feature of the CKF is that the local predictors

$$\widehat{x}_{i, t|t} = \mathcal{I}_{i, t|t}^{-1} \ i_{i, t|t}, \quad i = 1, ..., n, \tag{41}$$

will have a *poorer* precision performance than that of their central counterpart $\widehat{x}_{t|t}$. This is due to the fact that the consensus states $N_{i, t}(k_n)$ and $r_{i, t}(k_n)$ ($i = 1, ..., n$) are just *approximations* of
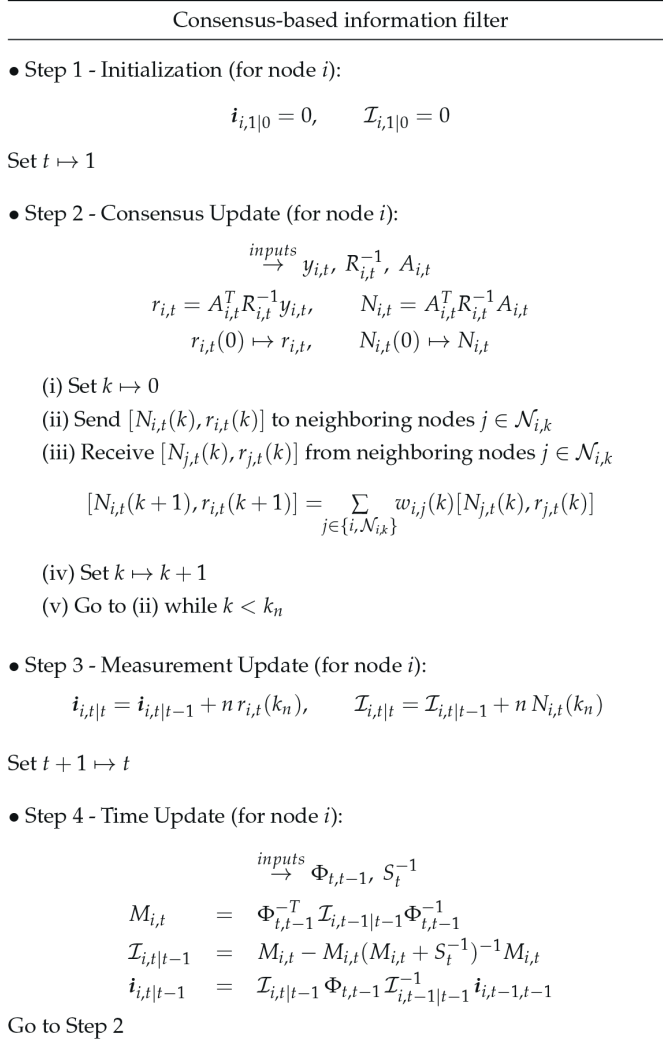

Consensus-based information filter

- Step 1 - Initialization (for node i):

$$i_{i,1|0} = 0, \qquad \mathcal{I}_{i,1|0} = 0$$

Set $t \mapsto 1$

- Step 2 - Consensus Update (for node i):

$$\overset{inputs}{\to} y_{i,t}, \ R_{i,t}^{-1}, \ A_{i,t}$$

$$r_{i,t} = A_{i,t}^{T} R_{i,t}^{-1} y_{i,t}, \qquad N_{i,t} = A_{i,t}^{T} R_{i,t}^{-1} A_{i,t}$$

$$r_{i,t}(0) \mapsto r_{i,t}, \qquad N_{i,t}(0) \mapsto N_{i,t}$$

(i) Set $k \mapsto 0$

(ii) Send $[N_{i,t}(k), r_{i,t}(k)]$ to neighboring nodes $j \in \mathcal{N}_{i,k}$

(iii) Receive $[N_{j,t}(k), r_{j,t}(k)]$ from neighboring nodes $j \in \mathcal{N}_{i,k}$

$$[N_{i,t}(k+1), r_{i,t}(k+1)] = \sum_{j \in \{i, \mathcal{N}_{i,k}\}} w_{i,j}(k)[N_{j,t}(k), r_{j,t}(k)]$$

(iv) Set $k \mapsto k+1$

(v) Go to (ii) while $k < k_n$

- Step 3 - Measurement Update (for node i):

$$i_{i,t|t} = i_{i,t|t-1} + n\, r_{i,t}(k_n), \qquad \mathcal{I}_{i,t|t} = \mathcal{I}_{i,t|t-1} + n\, N_{i,t}(k_n)$$

Set $t + 1 \mapsto t$

- Step 4 - Time Update (for node i):

$$\overset{inputs}{\to} \Phi_{t,t-1}, \ S_t^{-1}$$

$$
\begin{aligned}
M_{i,t} &= \Phi_{t,t-1}^{-T} \mathcal{I}_{i,t-1|t-1} \Phi_{t,t-1}^{-1} \\
\mathcal{I}_{i,t|t-1} &= M_{i,t} - M_{i,t}(M_{i,t} + S_t^{-1})^{-1} M_{i,t} \\
i_{i,t|t-1} &= \mathcal{I}_{i,t|t-1} \Phi_{t,t-1} \mathcal{I}_{i,t-1|t-1}^{-1} i_{i,t-1,t-1}
\end{aligned}
$$

Go to Step 2

Figure 6. Algorithmic steps of the CKF in information form concerning the time-evolution of the local information vector $i_{i,t|t}$ and matrix $\mathcal{I}_{i,t|t}$ of node i.

the averages $\overline{N}_t$ and $\overline{r}_t$. Although they reach the stated averages as $k_n \to \infty$, one of course always comes up with a *finite* number of communications k_n. As a consequence, while the inverse-matrix $\mathcal{I}_{t|t}^{-1}$ represents the error variance matrix $P_{t|t} = D(x_t - \hat{x}_{t|t})$ (cf. 17), the inverse-matrices $\mathcal{I}_{i,t|t}^{-1}$ $(i = 1, ..., n)$ do *not* represent the error variance matrices $P_{i,t|t} = D(x_t - \hat{x}_{i,t|t})$. To see this, consider the local prediction errors $(x_t - \hat{x}_{i,t|t})$ which can be expressed as (**Figure 6**)

$$x_t - \widehat{x}_{i,t|t} = \mathcal{I}_{i,t|t}^{-1} \left\{ \mathcal{I}_{i,t|t-1} \left(x_t - \widehat{x}_{i,t|t-1} \right) - n \left[r_{i,t}(k_n) - N_{i,t}(k_n) x_t \right] \right\} \tag{42}$$

Note that the terms $\left(x_t - \widehat{x}_{i,t|t-1} \right)$ and $\left[r_{i,t}(k_n) - N_{i,t}(k_n) x_t \right]$ are uncorrelated. With l_{ij} as the entries of the product matrix $L(k_n)$ in Eq. (34), one obtains4

$$\left[r_{i,t}(k_n) - N_{i,t}(k_n) x_t \right] = \sum_{j=1}^{n} l_{ij} \left(r_{j,t} - N_{j,t} x_t \right), \quad \Rightarrow \quad D\left(\left[r_{i,t}(k_n) - N_{i,t}(k_n) x_t \right] \right) = \sum_{j=1}^{n} l_{ij}^2 N_{j,t} \tag{43}$$

since $D\left(r_{j,t} - N_{j,t} x_t \right) = N_{j,t}$. With this in mind, an application of the covariance propagation law to (42) results in the error variance matrix

$$P_{i,t|t} = \mathcal{I}_{i,t|t}^{-1} \left\{ \mathcal{I}_{i,t|t-1} P_{i,t|t-1} \mathcal{I}_{i,t|t-1} + n^2 \sum_{j=1}^{n} l_{ij}^2 N_{j,t} \right\} \mathcal{I}_{i,t|t}^{-1} \tag{44}$$

that is not necessarily equal to $\mathcal{I}_{i,t|t}^{-1}$ (see the following discussion on Eqs. (47) and (48)).

In **Figure 7** we present the three-step recursion of the error variance matrix $P_{i,t|t}$ (for node i). As shown, the node i would need an *extra* input, i.e., the term $\sum_{j=1}^{n} l_{ij}^2 N_{j,t}$ in order to be able to compute $P_{i,t|t}$. In practice however, such additional information is absent in the CKF setup. This means that the node i does not have enough information to evaluate the error variance matrix $P_{i,t|t}$. Despite such restriction, it will be shown in Section 5 how the recursion of $P_{i,t|t}$ conveys useful information about the performance of the local filters $i = 1, \ldots, n$, thereby allowing one to *a-priori* design and analyze sensor networks with different numbers of iterative communications.

To better appreciate the recursion given in **Figure 7**, let us consider a special case where a *stationary* state-vector x_t is to be predicted over time. Thus $\Phi_{t,t-1} = I$ and $S_t = 0$ ($t = 1, 2, \ldots$). Moreover, we assume that all nodes deliver the same normal matrices $N_{i,t} = N$ ($i = 1, \ldots, n$).

The central error variance matrix $P_{t|t}$ would then simply follow by inverting the sum of all normal matrices over n nodes and t time instances. Collecting observables up to and including time instance t, the stated variance matrix reads $P_{t|t} = (1/tn) N^{-1}$. We now compare $P_{t|t}$ with its consensus-based local counterpart at node i, i.e. $P_{i,t|t}$. The aforementioned assumptions, together with $\sum_{j=1}^{n} l_{ij} = 1$, give

$$N_{i,t}(k_n) = \sum_{j=1}^{n} l_{ij} N_{j,t} = N, \quad \text{and} \quad n^2 \sum_{j=1}^{n} l_{ij}^2 N_{j,t} = \alpha n N \tag{45}$$

in which the scalar α is given by

$$\alpha := n \sum_{j=1}^{n} l_{ij}^2 \tag{46}$$

Substitution into the stated recursion provides us with the time-evolution of the error variance matrix $P_{i,t|t}$ as follows (**Figure 7**)
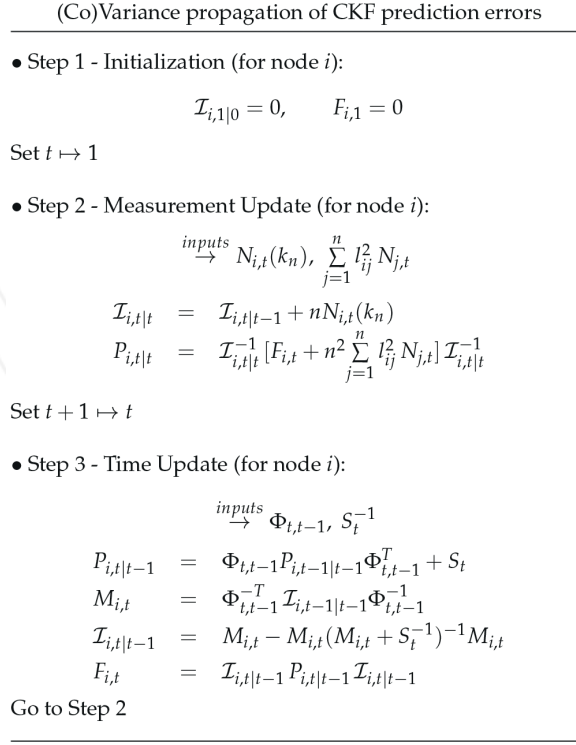

(Co)Variance propagation of CKF prediction errors

- Step 1 - Initialization (for node i):

$$\mathcal{I}_{i,1|0} = 0, \qquad F_{i,1} = 0$$

Set $t \mapsto 1$

- Step 2 - Measurement Update (for node i):

$$\overset{inputs}{\rightarrow} N_{i,t}(k_n), \ \sum_{j=1}^{n} l_{ij}^2 N_{j,t}$$

$$\mathcal{I}_{i,t|t} = \mathcal{I}_{i,t|t-1} + n N_{i,t}(k_n)$$
$$P_{i,t|t} = \mathcal{I}_{i,t|t}^{-1}[F_{i,t} + n^2 \sum_{j=1}^{n} l_{ij}^2 N_{j,t}]\,\mathcal{I}_{i,t|t}^{-1}$$

Set $t+1 \mapsto t$

- Step 3 - Time Update (for node i):

$$\overset{inputs}{\rightarrow} \Phi_{t,t-1}, \ S_t^{-1}$$

$$P_{i,t|t-1} = \Phi_{t,t-1}P_{i,t-1|t-1}\Phi_{t,t-1}^T + S_t$$
$$M_{i,t} = \Phi_{t,t-1}^{-T}\mathcal{I}_{i,t-1|t-1}\Phi_{t,t-1}^{-1}$$
$$\mathcal{I}_{i,t|t-1} = M_{i,t} - M_{i,t}(M_{i,t} + S_t^{-1})^{-1}M_{i,t}$$
$$F_{i,t} = \mathcal{I}_{i,t|t-1}P_{i,t|t-1}\mathcal{I}_{i,t|t-1}$$

Go to Step 2

Figure 7. The three-step recursion of the error variance matrix $P_{i,t|t} = \mathrm{D}(x_t - \widehat{x}_{i,t|t})$ for node i. The *extra* term $\sum_{j=1}^{n} l_{ij}^2 N_{j,t}$ would be required to compute $P_{i,t|t}$. The entries of the product matrix $L(k_n)$ in Eq. (34) are denoted by l_{ij}.

$$\mathcal{I}_{i,1|0} = 0, \quad F_{i,1} = 0$$
$$\downarrow$$
$$\mathcal{I}_{i,1|1} = nN, \qquad\qquad \Rightarrow \quad P_{i,1|1} = \alpha\frac{1}{n}N^{-1}$$
$$\downarrow$$
$$\mathcal{I}_{i,2|1} = \mathcal{I}_{i,1|1}, \quad F_{i,2} = \alpha nN \qquad \Rightarrow \quad P_{i,2|1} = P_{i,1|1} \tag{47}$$
$$\vdots \qquad\qquad\qquad \vdots \qquad\qquad \vdots$$
$$\mathcal{I}_{i,t|t-1} = \mathcal{I}_{i,t-1|t-1}, \quad F_{i,t} = \alpha(t-1)nN \ \Rightarrow \ P_{i,t|t-1} = P_{i,t-1|t-1}$$
$$\downarrow$$
$$\mathcal{I}_{i,t|t} = tnN, \qquad\qquad \Rightarrow \quad P_{i,t|t} = \alpha\frac{1}{tn}N^{-1}$$

This shows that the consensus-based error variance matrix $P_{i,t|t}$ is α times its central counterpart $P_{t|t} = (1/tn)N^{-1}$. With the vector $l := [l_{i1}, \ldots, l_{in}]^T$, application of the Cauchy-Schwarz inequality gives the lower-bound

$$\alpha = \left(e_n^T e_n\right)\left(l^T l\right) \geq \left(l^T e_n\right)^2 = 1 \tag{48}$$

as $l^T e_n = 1$. Thus scalar α is never smaller than 1, i.e. $P_{i,t|t} \geq P_{t|t}$, showing that the performance of the consensus-based predictor $\hat{x}_{i,t|t}$ is never better than that of its central version $\hat{x}_{t|t}$. The lower-bound Eq. (48) is reached when $l = (1/n)e_n$, i.e. when $l_{ij} = 1/n$ $(j = 1, \dots, n)$. According to Eq. (34), this can be realized if $L(k_n) \to (1/n)e_n e_n^T$, for which the number of iterations k_n might be required to be reasonably large. The conclusion reads therefore that the local filters at nodes $i = 1, \dots, n$, generate information matrices $\mathcal{I}_{i,t|t}$, the inverse of which are different from the actual error variance matrices of the predictors $\hat{x}_{i,t|t}$, i.e. $\mathcal{I}_{i,t|t}^{-1} \neq P_{i,t|t}$.

5. Applications to GNSS

The purpose of this section is to demonstrate how the CKF theory, discussed in Section 4, can play a pivotal role in applications for which the GNSS measurements of a *network* of receivers are to be processed in a real-time manner. In a GNSS network setup, each receiver serves as a sensor node for receiving observables from visible GNSS satellites to determine a range of different parameters such as positions and velocities in an Earth-centered Earth-fixed coordinate system, atmospheric delays, timing and instrumental biases, see e.g. [11, 12]. As the observation equations of the receivers have satellite specific parameters in common, the receivers' observables are often integrated through a computing (fusion) center to provide network-derived parameter solutions that are more precise than their single-receiver versions. Now the idea is to deliver GNSS parameter solutions without the need of having a computing center, such that their precision performance is still comparable to that of network-derived solutions.

As previously discussed, consensus-based algorithms and in particular the CKF can be employed to process network data in a distributed filtering scheme, i.e. no computing center is required. In order to illustrate such applicability, we simulate a network of 13 GNSS receivers located in Perth, Western Australia (**Figure 8**). As shown in the figure, each node (white circle) represents a receiver having data links (red lines) to its neighbors with inter-station distances up to 4 km. We therefore assume that the receivers receive each other data within the ranges not longer than 4 km. For instance, receiver 1 is directly connected to receivers 2 and 6, but not to receiver 3 (the inter-station distance between receivers 1 and 3 is about 8 km).

5.1. GNSS ionospheric observables: Dynamic and measurement models

Although the GNSS observables contain information on various positioning and non-positioning parameters, here we restrict ourselves to *ionospheric observables* of the GPS *pseudo-range* measurements only [44]. One should however bear in mind that such restriction is made just for the sake of presentation and illustration of the theory discussed in Sections 3 and 4.
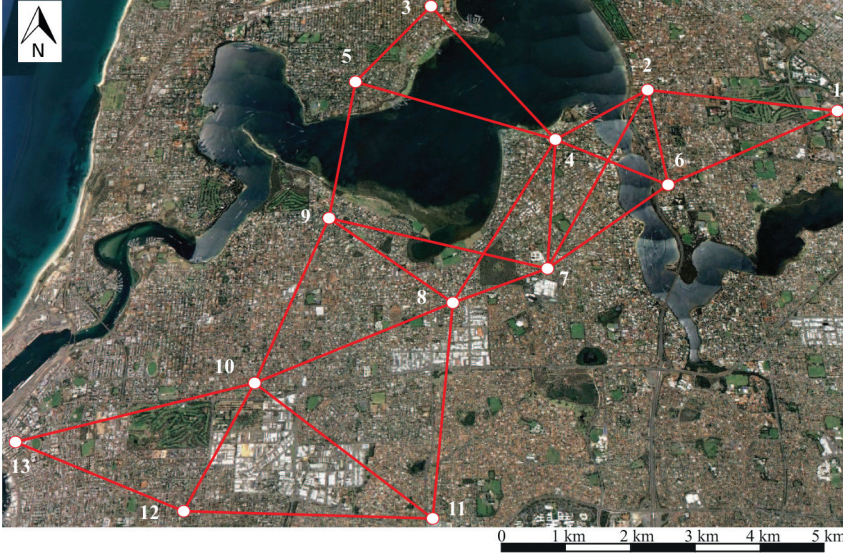
Figure 8. A network of 13 GNSS receivers simulated over Perth, Western Australia. Each node (white circle) represents a receiver tracking GNSS satellites. The receivers have data links to their neighbors with inter-station distances up to 4 km. The data links are shown by red lines.

Would one, for instance, make use of the very precise carrier-phase measurements and/or formulate a multi-GNSS measurement setup, solutions of higher precision levels are therefore expected.

Let the scalar $y_{i,t}^s$ denote the pseudo-range ionospheric observable that the receiver i collects from satellite s at time instance t. The corresponding measurement model, formed by the between-satellite differences $y_{i,t}^{ps} := y_{i,t}^s - y_{i,t}^p$ ($s \neq p$), reads (cf. 5)

$$y_{i,t}^{ps} = \left\{ a_{i,t;o}^{ps} v_{o,t} + a_{i,t;\phi}^{ps} v_{\phi,t} + a_{i,t;\psi}^{ps} v_{\psi,t} \right\} - b_t^{ps} + \varepsilon_{i,t}^{ps} \tag{49}$$

where the term within $\{.\}$ refers to the first-order slant ionospheric delays, and b_t^{ps} denotes the between-satellite differential code biases (DCBs). We use a regional single-layer model [45, 46] to represent the slant ionospheric delays in terms of 1) $v_{o,t}$ as the vertical total electron content (TEC), 2) $v_{\phi,t}$ and 3) $v_{\psi,t}$ as the south-to-north and west-to-east spatial gradient of $v_{o,t}$, respectively. The corresponding known coefficients follow from Ref. [47]

$$a_{i,t;o}^s = \frac{1}{\cos\left(z_{i,t}^s\right)}, \quad a_{i,t;\phi}^s = \frac{1}{\cos\left(z_{i,t}^s\right)} \left(\phi_{i,t}^s - \phi_{o,t}\right), \quad a_{i,t;\psi}^s = \frac{1}{\cos\left(z_{i,t}^s\right)} \cos\left(\phi_{i,t}^s\right) \left(\psi_{i,t}^s - \psi_{o,t}\right)$$

$$\tag{50}$$

with $(.)^{ps} := (.)^s - (.)^p$. The angles $\psi_{i,t}^s$ and $\phi_{i,t}^s$, respectively, denote the longitude and latitude of the ionospheric piercing points (IPPs) corresponding to the receiver-to-satellite line-of-sight

$i-s$ (see **Figure 9**). They are computed with respect to those of the reference IPP at time instance t, i.e. $\psi_{o,t}$ and $\phi_{o,t}$. The angle $z_{i,t}^s$ denotes the zenith angle of the IPPs. These angles are computed based on the mean Earth's radius 6378.137 km and height of layer 450 km. The measurement noises $\varepsilon_{i,t}^s$ are assumed to be mutually uncorrelated with the dispersion (cf. 6)

$$D\left(\varepsilon_{i,t}^s\right) = \frac{1.02^2}{\left(0.02 + \sin\left(\theta_{i,t}^s\right)\right)^2} \sigma^2 \tag{51}$$

forming the variance matrices R_t in Eq. (6), where $\theta_{i,t}^s$ is the satellite elevation angle. The scalar σ is set to $\sigma \approx 65.6$ cm as the zenith-referenced standard-deviation of the GPS 'geometry-free' pseudo-range measurements [48].

Suppose that m number of satellites $s = 1, \ldots, m$, are tracked by the network receivers $i = 1, \ldots, n = 13$, during the observational campaign. The state-vector sought is structured as

$$x_t = \left[v_{o,t}, v_{\phi,t}, v_{\psi,t}, b_t^{p1}, b_t^{p2}, \ldots, b_t^{pm}\right]^T \tag{52}$$

Thus the state-vector x_t contains three TEC parameters $v_{o,t}$, $v_{\phi,t}$, $v_{\psi,t}$ and $(m-1)$ between-satellite DCBs b_t^{ps} ($s \neq p$). The dynamic model is assumed to be given by (cf. 2, 4 and 21)

$$\begin{bmatrix} v_{o,t} \\ v_{\phi,t} \\ v_{\psi,t} \end{bmatrix} = \begin{bmatrix} v_{o,t-1} \\ v_{\phi,t-1} \\ v_{\psi,t-1} \end{bmatrix} + \begin{bmatrix} d_o \\ d_\phi \\ d_\psi \end{bmatrix}, \quad \text{and} \quad b_t^{ps} = b_{t-1}^{ps} \quad s \neq p \tag{53}$$

Thus the DCBs b_t^{ps} are assumed constant in time, while the temporal behavior of the TEC parameters $v_{o,t}$, $v_{\phi,t}$, $v_{\psi,t}$ is captured by a random-walk process. The corresponding zero-mean
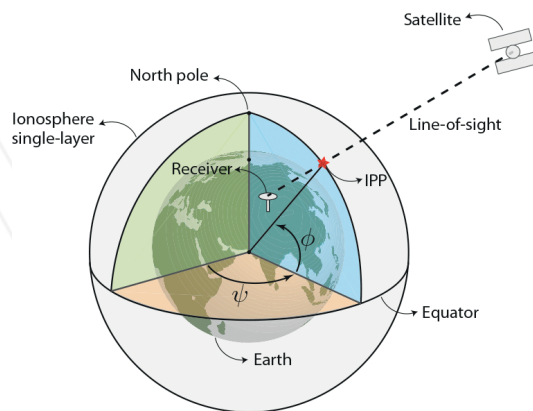

Figure 9. Longitude (ψ) and latitude (ϕ) of an ionospheric piercing point (IPP) corresponding to a receiver-to-satellite line-of-sight. The distance scales are exaggerated.

process noises are assumed to be mutually uncorrelated, having the standard-deviations $\sigma_{d_o} = 1$ mm/$\sqrt{sec}$ and $\sigma_{d_\phi} = \sigma_{d_{psi}} = 5$ mm/rad/$\sqrt{sec}$ [49].

5.2. Observational campaign

The network receivers $i = 1, \ldots, n$ ($n = 13$), shown in **Figure 8**, are assumed to track GPS satellites over 16 hours from 8:00 to 24:00 Perth local time, on 02-06-2016. The observation sampling rate is set to $\Delta = 1$ minute. Thus the number of observational epochs (time instances) is 960. As to the data sending rate δ (cf. 5), we assume three different sending rates $\delta = 5, 10$ and 15 seconds. Thus the number of iterative communications between the neighboring receivers takes the values $k_n = 4, 6$ and 12. The consensus protocol 3 (**Table 1**) is applied to the CKF of each receiver.

As the satellites revolve around the Earth, not all of which are simultaneously visible to the 'small-scale' network of **Figure 8**. Their visibility over time is shown in **Figure 10** (left panel) in which the satellites with elevation angles smaller than 10 degrees are excluded. There are 31 GPS satellites (i.e. $m = 31$), with PRN 4 absent (PRN refers to the satellite identifier). PRN 22 has the maximum duration of visibility, while PRN 21 has the minimum duration of visibility. Note also that PRNs 2, 6, 16, 17, 19, 26 and 32 disappear (set) and reappear (re-rise). That is why their visibility is shown via two separate time intervals. **Figure 10** (right panel) shows the trajectories of the ionospheric pierce points on the ionospheric single layer that are made by receiver-to-satellite line-of-sight paths. It is the spatial distribution of these points that drives the coefficients $a_{i,t;o}^{ps}$, $a_{i,t;\phi}^{ps}$, $a_{i,t;\psi}^{ps}$ in Eq. (49).

In the following we present precision analyses on the *measurement update* solutions of x_t in Eq. (52), given the network and satellite configurations shown in **Figures 8** and **10**, respectively. Throughout the text, PRN 10 is chosen as the pivot satellite p (cf. (49)). By the term 'standard-deviation', we mean the square-root of prediction errors' variance.
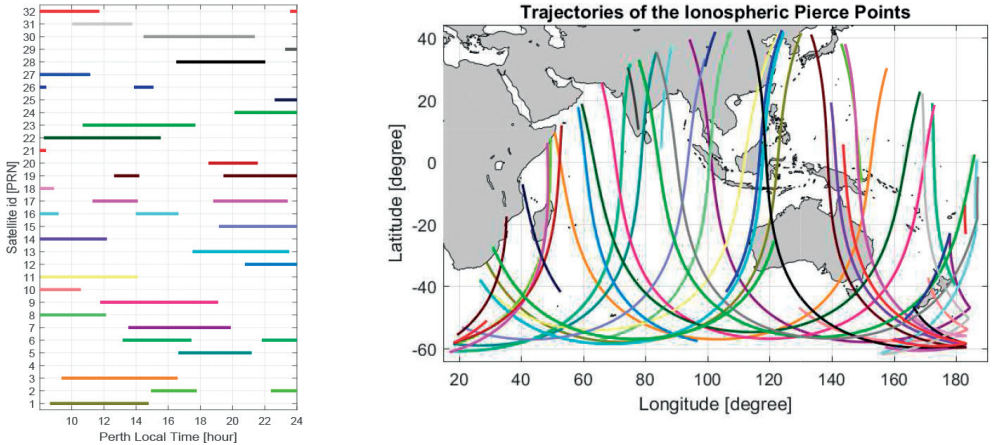

Figure 10. *Left*: GPS satellites visibility over time, viewed from Perth, Western Australia. *Right*: Trajectories of the corresponding IPPs made by receiver-to-satellite line-of-sight paths. The satellites are indicated by different colors.

5.3. Central (network-based) versus local (single-receiver) solutions

Before discussing the precision performance of the CKF solutions, we first compare the network-based (central) TEC solutions with the solutions that are obtained by the data of one single-receiver only (referred to as the local solutions). At the filter initialization, the standard-deviations of the local TEC solutions are $\sqrt{13} \approx 3.6$ times larger than those of the central TEC solutions (i.e. square-root of the number of nodes). This is because of the fact that each of the 13 network receivers independently provides equally precise solutions. In that case, the central solution follows then by averaging all the 13 local solutions. Due to the common dynamic model Eq. (53) however, the local solutions become correlated over time. After the filter initialization, the central solution would therefore *not* follow the average of its local versions. The standard-deviation results, after one hour of the filter initialization, are presented in **Figure 11**. Only the results of the receiver 1 are shown as local solutions (in red). As shown, the standard-deviations get stable over time as the filters reach their steady-state. On the right panel of the figure, the local-to-central standard-deviation ratios are also presented. In case of the vertical TECs $v_{o,t}$, the ratios vary from 1.5 to 3. For the horizontal gradients $v_{\phi,t}$ and $v_{\psi,t}$, the ratios are about 2 and 2.5, respectively.

5.4. Role of CKF in improving local solutions

With the results of **Figure 11**, we observed that the central TEC solutions considerably outperform their local counterparts in the sense of delivering more precise outcomes, i.e. the local-to-central standard-deviation ratios are considerably larger than 1. We now employ the
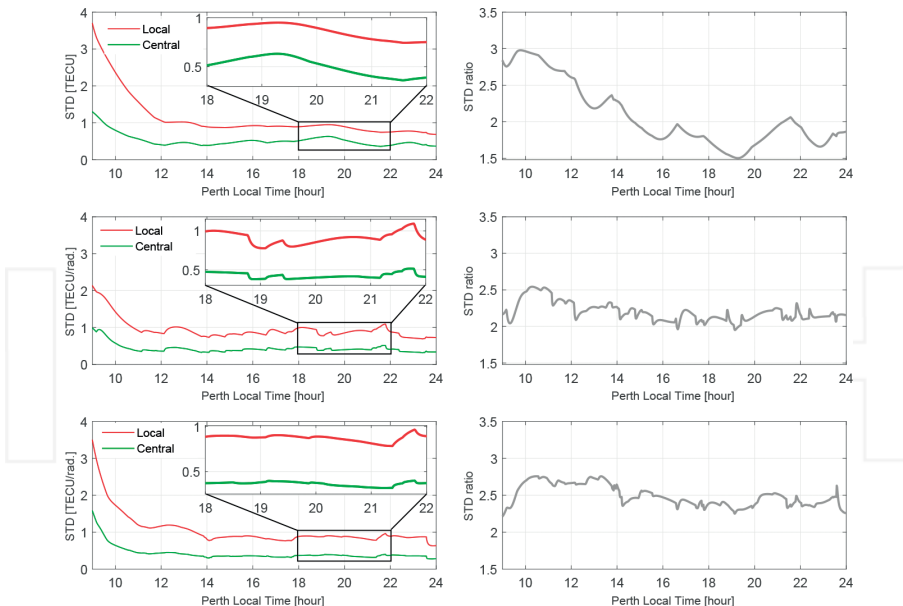

Figure 11. *Left*: Standard-deviation of the central (green) and local (red) solutions of the TEC parameters $v_{o,t}$ (top), $v_{\phi,t}$ (middle) and $v_{\psi,t}$ (bottom) as functions of time. *Right*: The corresponding local-to-central standard-deviation ratios.

CKF for each node (receiver) $i = 1, \ldots, 13$, to *improve* the local solutions' precision performance via consensus-based iterative communications between the receivers. In doing so, we make use of the three-step recursion given in **Figure 7** to evaluate the error variance matrices $P_{i,t|t}$ ($i = 1, \ldots, 13$), thereby computing the CKF-to-central standard-deviation ratios. The stated ratios are presented in **Figure 12** for two different data sending rates $\delta = 15$ seconds (left panel) and $\delta = 5$ seconds (right panel). In both cases, the CKF-to-central standard-deviation ratios are smaller than their local-to-central versions shown in **Figure 11** (right panel), illustrating that employing the CKF does indeed improve the local solutions' precision. Since more iterative communications take place for $\delta = 5$, the corresponding ratios are very close to 1. In that case, the CKF of each receiver is expected to have a similar precision performance to that of the central (network-based) filter. For the case $\delta = 15$ however, the CKF performance of each receiver does very much depend on the number of the receiver's neighbors. This is because of the fact that only 4 iterative communications between the receivers take place (i.e. $k_n = 4$). The receivers with the minimum number of neighbors, i.e. receivers 1, 3 and 13 (**Figure 8**), have the worst precision performance as the corresponding ratios take largest values. On the other hand, the receivers with the maximum number of neighbors, i.e. receivers 4, 7, 9 and 8, have the best performance as the corresponding ratios are close to 1.

Next to the solutions of the TEC parameters $\nu_{o,t}$, $\nu_{\phi,t}$ and $\nu_{\psi,t}$, we also analyze CKF solutions of the between-satellite DCBs b_t^{ps} ($s \neq p$) in Eq. (52). Because of the difference in the satellites visibility over time (cf. **Figure 10**), the DCBs' standard-deviations are quite distinct and very much depend on the duration of the satellites visibility. The more a pair of satellites $p - s$ are
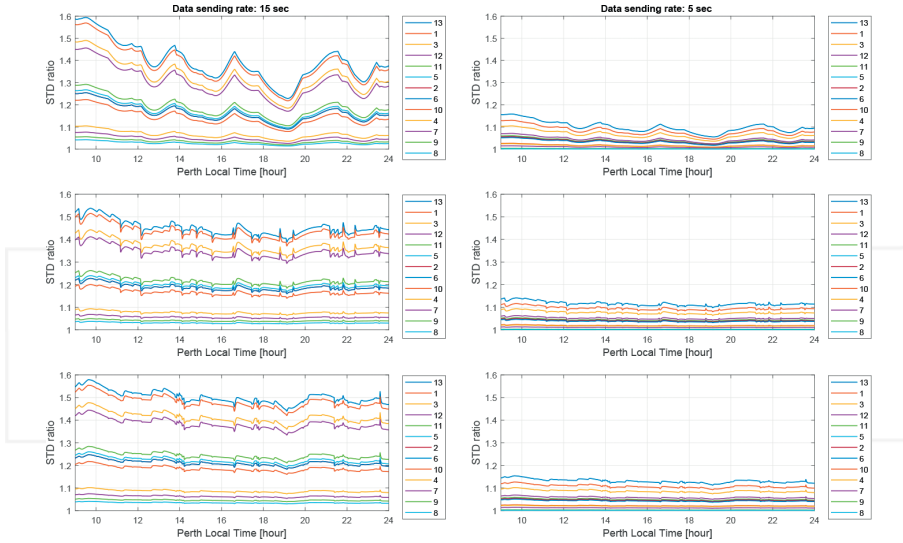

Figure 12. Time-series of the CKF-to-central standard-deviation ratios corresponding to the TEC parameters $\nu_{o,t}$ (top), $\nu_{\phi,t}$ (middle) and $\nu_{\psi,t}$ (bottom). *Left*: The data sending rate is set to $\delta = 15$ seconds (i.E. 4 iterative communications). *Right*: The data sending rate is set to $\delta = 5$ seconds (i.E. 12 iterative communications). The results of the nodes (receivers) are indicated by different colors.

visible, the smaller the standard-deviation is expected. We now consider the required time to have between-satellite DCBs solutions with standard-deviation smaller than 0.5 nanoseconds. Because of the stated difference in the standard-deviations, each between-satellite DCB corresponds to a different required time. For the central filter, the minimum value of such required time is 7 minutes, with the 25th percentile as 12, median as 38, 75th percentile as 63 and the maximum as 84 minutes. Thus after 84 minutes of the filter initialization, all central DCB solutions have standard-deviations smaller than 0.5 nanoseconds. Such percentiles can be represented by a 'boxplot'. We compute the stated percentiles for all the CKF solutions and compare their boxplots with the central one in **Figure 13**. The results are presented for three
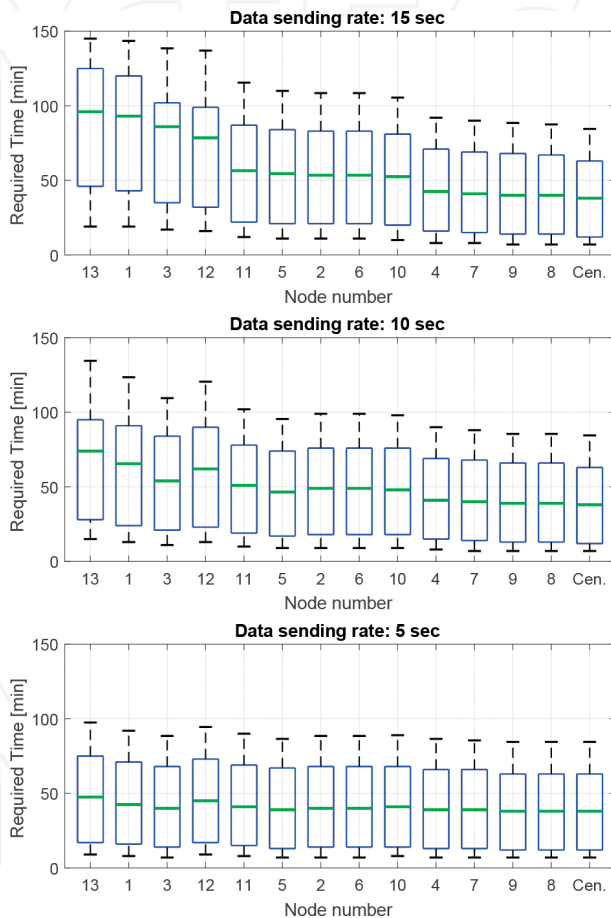

Figure 13. Boxplots of the required time (minutes) to have between-satellite DCBs solutions with standard-deviation smaller than 0.5 nanoseconds. The performance of the CKF of each node (receiver) is compared to that of the central filter (Cen.). In each boxplot, the horizontal lines from bottom to top show the minimum (black), 25th percentile (blue), median (green), 75th percentile (blue) and maximum (black) of the stated required time. The data sending rate is set to *Top*: $\delta = 15$ seconds (i.E. 4 iterative communications), *Middle*: $\delta = 10$ seconds (i.E. 6 iterative communications), and *Bottom*: $\delta = 5$ seconds (i.E. 12 iterative communications).

different data sending rates $\delta = 15$ seconds (top), $\delta = 10$ seconds (middle) and $\delta = 5$ seconds (bottom). As shown, the more the number of iterative communications, the more similar the boxplots becomes, i.e. the nodes (receivers) are reaching consensus. Similar to the TEC solutions, the DCB precision performance of the CKF corresponding to the receivers 4, 7, 9 and 8 is almost similar to that of the central one, irrespective of the number of iterative communications. This follows from the fact that the stated receivers have the maximum number of neighbors (**Figure 8**), thus efficiently approximating the averages $\overline{N}_t$ and $\overline{r}_t$ in Eq. (28) after a few iterations. On the other hand, the receivers with the minimum number of neighbors require more number of iterative communications in order for their CKF precision performance to get similar to that of the central filter.

6. Concluding remarks and future outlook

In this contribution we reviewed Kalman filtering in its information form and showed how the additive measurement update (28) can be realized by employing average-consensus rules, even when *not* all nodes are directly connected, thus allowing the sensor nodes to develop their own distributed filters. The nodes are assumed linked to each other *at least* through a 'path' so that information can flow from each node to all other nodes. Under this assumption, average-consensus protocols can deliver consensus states $[N_{i,t}(k_n), r_{i,t}(k_n)]$ as an approximation of the averages $[\overline{N}_t, \overline{r}_t]$ in Eq. (28) at every time instance $t = 1, 2, \ldots$, thus allowing one to establish a CKF recursion at every node $i = 1, \ldots, n$. To improve the stated approximation, the neighboring nodes have to establish a number of iterative data communications to transfer and receive their consensus states. This makes the CKF implementation applicable only for the applications in which the state-vectors change rather slowly over time (i.e. the sampling rate Δ can take large values) and/or for the cases where the sensor nodes transfer their data rather quickly (i.e. the sending rate δ can take small values).

We developed a three-step recursion of the CKF error variance matrix (**Figure 7**). This recursion conveys useful information about the precision performance of the local filters $i = 1, \ldots, n$, thereby enabling one to a-priori design and analyze sensor networks with different numbers of iterative communications. As an illustrative example, we applied the stated recursion to a small-scale network of GNSS receivers and showed the role taken by the CKF in improving the precision of the solutions at each single receiver. In near future the proliferation of low-cost receivers will give rise to an increase in the number of GNSS users. Employing the CKF or other distributed filtering techniques, GNSS users can therefore potentially deliver high-precision parameter solutions without the need of having a computing center.

Acknowledgements

Author details

Amir Khodabandeh[1]*, Peter J.G. Teunissen[1,2] and Safoora Zaminpardaz[1]

*Address all correspondence to: amir.khodabandeh@curtin.edu.au

1 Curtin University of Technology, Australia

2 Delft University of Technology, The Netherlands

References

[1] Cattivelli FS, Sayed AH. Diffusion strategies for distributed Kalman filtering and smoothing. IEEE Transactions on Automatic Control. 2010;**55**(9):2069-2084

[2] Das S, Moura JMF. Consensus+innovations distributed Kalman filter with optimized gains. IEEE Transactions on Signal Processing. 2017;**65**(2):467-481

[3] Jadbabaie A, Lin J, Stephen Morse A. Coordination of groups of mobile autonomous agents using nearest neighbor rules. IEEE Transactions on Automatic Control. 2003;**48**(6):988-1001

[4] Kingston DB, Beard RW. Discrete-time average-consensus under switching network topologies. In: American Control Conference, 2006. IEEE; 2006. pp. 3551-3556

[5] Moreau L. Stability of multiagent systems with time-dependent communication links. IEEE Transactions on Automatic Control. 2005;**50**(2):169-182

[6] Olfati-Saber R, Murray RM. Consensus problems in networks of agents with switching topology and time-delays. IEEE Transactions on Automatic Control. 2004;**49**(9):1520-1533

[7] Scherber DS, Papadopoulos HC. Locally constructed algorithms for distributed computations in ad-hoc networks. In: Proceedings of the 3rd International Symposium on Information Processing in Sensor Networks. ACM; 2004. pp. 11-19

[8] Xiao L, Boyd S, Lall S. A scheme for robust distributed sensor fusion based on average consensus. In: Proceedings of the 4th International Symposium on Information Processing in Sensor Networks. IEEE Press; 2005. pp. 63-70

[9] Rigatos GG. Distributed filtering over sensor networks for autonomous navigation of UAVs. Intelligent Service Robotics. 2012;**5**(3):179-198

[10] Sugar TG, Kumar V. Control of cooperating mobile manipulators. IEEE Transactions on Robotics and Automation. 2002;**18**(1):94-103

[11] Hofmann-Wellenhof B, Lichtenegger H, Wasle E. GNSS: Global Navigation Satellite Systems: GPS, Glonass, Galileo, and More. New York: Springer; 2008

[12] Teunissen PJG, Montenbruck O, editors. Springer Handbook of Global Navigation Satellite Systems. Switzerland: Springer; 2017

[13] Casbeer DW, Beard R. Distributed information filtering using consensus filters. In: American Control Conference, 2009. ACC'09. IEEE; 2009. pp. 1882-1887

[14] Khodabandeh A, Teunissen PJG. An analytical study of PPP-RTK corrections: Precision, correlation and user-impact. Journal of Geodesy. 2015;**89**(11):1109-1132

[15] Li W, Nadarajah N, Teunissen PJG, Khodabandeh A, Chai Y. Array-aided single-frequency state-space RTK with combined GPS, Galileo, IRNSS, and QZSS L5/E5a observations. Journal of Surveying Engineering. 2017;**143**(4):04017006

[16] Li X, Ge M, Dai X, Ren X, Fritsche M, Wickert J, Schuh H. Accuracy and reliability of multi-GNSS real-time precise positioning: GPS, GLONASS, BeiDou, and Galileo. Journal of Geodesy. 2015;**89**(6):607-635

[17] Odolinski R, Teunissen PJG. Single-frequency, dual-GNSS versus dual-frequency, single-GNSS: A low-cost and high-grade receivers GPS-BDS RTK analysis. Journal of Geodesy. 2016;**90**(11):1255-1278

[18] Zaminpardaz S, Teunissen PJG, Nadarajah N. GLONASS CDMA L3 Ambiguity Resolution and Positioning. Berlin, Heidelberg: GPS Solutions; 2016. pp. 1-15

[19] Brammer K, Siffling G. Kalman-Bucy Filters. Berlin: Artech House; 1989

[20] Candy JV. Signal Processing: Model Based Approach. New Jersey: McGraw-Hill, Inc.; 1986

[21] Gelb A. Applied Optimal Estimation. London: MIT press; 1974

[22] Gibbs BP. Advanced Kalman Filtering, Least-Squares and Modeling: A Practical Handbook. New Jersey: Wiley; 2011

[23] Jazwinski AH. Stochastic Processes and Filtering Theory. Maryland: Dover Publications; 1991

[24] Kailath T. Lectures on Wiener and Kalman Filtering. Number 140. Vienna: Springer; 1981

[25] Kalman RE. A new approach to linear filtering and prediction problems. Journal of Basic Engineering. 1960;**82**(1):35-45

[26] Teunissen PJG. Best prediction in linear models with mixed integer/real unknowns: Theory and application. Journal of Geodesy. 2007;**81**(12):759-780

[27] Bode HW, Shannon CE. A simplified derivation of linear least square smoothing and prediction theory. Proceedings of the IRE. 1950;**38**(4):417-425

[28] Kailath T. An innovations approach to least-squares estimation–part I: Linear filtering in additive white noise. Automatic Control, IEEE Transactions on. 1968;**13**(6):646-655

[29] Zadeh LA, Ragazzini JR. An extension of Wiener's theory of prediction. Journal of Applied Physics. 1950;**21**(7):645-655

[30] Anderson BDO, Moore JB. Optimal Filtering. Vol. 11. New Jersey: Prentice-hall Englewood Cliffs; 1979

[31] Bar-Shalom Y, Li XR. Estimation and Tracking- Principles, Techniques, and Software. Vol. 1993. Norwood, MA: Artech House, Inc; 1993

[32] Grewal MS, Andrews AP. Kalman Filtering; Theory and Practice Using MATLAB. 3rd ed. New Jersey: John Wiley and Sons; 2008

[33] Kailath T, Sayed AH, Hassibi B. Linear Estimation. New Jersey: Prentice-Hall; 2000

[34] Maybeck PS. Stochastic Models, Estimation, and Control. Vol. 1. Academic Press; 1979. Republished 1994

[35] Simon D. Optimal State Estimation: Kalman, H [Infinity] and Nonlinear Approaches. New Jersey: John Wiley and Sons; 2006

[36] Sorenson HW. Kalman filtering techniques. In: Leondes CT editor. Advances in Control Systems Theory and Applications, Vol. 3; 1966. pp. 219-292

[37] Stark H, Woods JW. Probability, Random Processes, and Estimation Theory for Engineers. Englewood Cliffs, New Jersey: Prentice-Hall; 1986

[38] Teunissen PJG, Khodabandeh A. BLUE, BLUP and the Kalman filter: Some new results. Journal of Geodesy. 2013;**87**(5):461-473

[39] Khodabandeh A, Teunissen PJG. A recursive linear MMSE filter for dynamic systems with unknown state vector means. GEM - International Journal on Geomathematics. 2014; **5**(1):17-31

[40] Ren W, Beard RW. Consensus of information under dynamically changing interaction topologies. In: American Control Conference, 2004. Proceedings of the 2004, Vol. 6. IEEE; 2004. pp. 4939-4944

[41] Henderson HV, Pukelsheim F, Searle SR. On the history of the Kronecker product. Linear and Multilinear Algebra. 1983;**14**(2):113-120

[42] Horn RA, Johnson CR. Matrix Analysis. New York: Cambridge UP; 1985

[43] Wolfowitz J. Products of indecomposable, aperiodic, stochastic matrices. Proceedings of the American Mathematical Society. 1963;**14**(5):733-737

[44] Blewitt G. An automatic editing algorithm for GPS data. Geophysical Research Letters. 1990;**17**(3):199-202

[45] Mannucci AJ, Wilson BD, Yuan DN, Ho CH, Lindqwister UJ, Runge TF. A global mapping technique for GPS-derived ionospheric total electron content measurements. Radio Science. 1998;**33**(3):565-582

[46] Schaer S. Mapping and predicting the Earth's ionosphere using the global positioning system. PhD thesis. Bern, Switzerland: University of Bern; 1999

[47] Brunini C, Azpilicueta FJ. Accuracy assessment of the GPS-based slant total electron content. Journal of Geodesy. 2009;**83**(8):773-785

[48] Khodabandeh A, Teunissen PJG. Array-aided multifrequency GNSS Ionospheric sensing: Estimability and precision analysis. IEEE Transactions on Geoscience and Remote Sensing. 2016;**54**(10):5895-5913

[49] Julien O, Macabiau C, Issler JL. Ionospheric delay estimation strategies using Galileo E5 signals only. In: GNSS 2009, 22nd International Technical Meeting of The Satellite Division of the Institute of Navigation, Savannah; 2009. pp. 3128-3141