Hierarchical Decentralized State Estimation With Unknown Correlation for Multiple and Partially Overlapping State Vectors

Jonatan Ralf Axel Klemets[1] and Morten Hovd[2]

***Abstract*— A fusion method for hierarchical decentralized state estimation is investigated with unknown cross-correlation. The process is divided into several local estimators for different subsystems that can operate independently from each other. Each of the local estimators estimates only part of the entire state vector, where some of the states overlap with each other. Using the proposed fusion method, the local state estimates are fused together, to reconstruct the global state vector and to get an improved state estimate. A comparison with other existing fusion methods was made through simulations and showed a reduction in estimation error when using the proposed method.**

I. Introduction

Increasing demands for efficient operation and profitability in the industry require a better understanding of the dynamics of the process. This has led to an increasing interest in more advanced control systems, e.g., Model Predictive Controllers (MPC) and Economic MPCs. To successfully implement such techniques, more frequent information of the states and disturbances is required [1]. Measuring all the important states and disturbances is often too expensive and often not possible. Therefore, they need to be estimated, using appropriate state estimators as, e.g., in [2], and [3]. For chemical processes and other large-scale systems, using centralized state estimators are in general not favorable due to the high computational complexity. Instead, it is preferable to decompose the problem into several subsystems with local estimators, that uses the locally available measurements [4].

In [5] the concepts of distributed and parallel state estimation was introduced, where it was concluded that distributed state estimation is a viable option when estimating the states for complex and large-scale systems. Information can be shared between all the subsystems or only with the neighboring subsystems. From the shared information, it should be possible to obtain a global estimate. Two main approaches exist for dealing with the distributed/decentralized state estimation problem:

One approach is to have every subsystem estimate the entire state vector using a global model for each local estimator. A global estimate is then received by fusing all the local estimates in a centralized fusion layer as in [6]. The main drawback of this approach is that it is a full order problem and each subsystem must have access to a full dynamical model of the system. Therefore, this approach is poorly suited for most chemical processes and other large-scale systems.

In the second approach, the different subsystems compute only a partial estimate using a local model and then transmits the information to its neighbors. This approach is also referred as partition-based state estimation [7]. It is in general more preferable for large-scale systems as in [4] since it results in low order estimation problems that rely only on local dynamic and measurement models. What information that is shared between the neighboring subsystems vary depending on the distributed estimation scheme, but can be the measurements, state estimates, and corresponding covariances. The main drawback with this method is that it requires good performance for the communication between the local estimators, where corrupted information, information delays, and transmission frequency limitations may cause problems.

Consensus algorithms for distributed state estimation based on Kalman filters have recently received much attention [8], [9], [10]. In general, they belong to the second approach, where each node (local estimator) computes an estimate and shares it with its neighbors. The state estimates by the different nodes converge to the same values using a consensus algorithm. The Kalman based consensus filters have mostly been applied to target tracking [11], where each local estimator tries to estimate the same state vector. However, a consensus based method for merging partially overlapping state estimates was also proposed in [12].

One aspect most of the distributed state estimation methods have in common is that they are typically designed using a top-down approach, i.e., they start with a global model and then decompose it into smaller subsystems. The local estimators for the decomposed subsystems then often employ the same state estimation algorithm.

A major issue in process control is that most chemical plants together with their different units are unique. Therefore, most models have to be obtained using empirical modeling, using different system identification techniques that requires excitation of the process. However, these experiments are often time-consuming and expensive since it interferes with the process operation. As a consequence, the accuracy of the models varies greatly, where some parts of the process can have very accurate nonlinear models, while other parts have basic linear models. Thus, using the same estimator algorithm for every subsystem would be suboptimal. Instead, it would be more practical to use a bottom-up approach, where each local estimator is independently designed using a local model and the local measurements. The local estimates

[1] Jonatan Ralf Axel Klemets is with the Department of Engineering Cybernetics, Norwegian University of Science and Technology, NTNU, NO-7491 Trondheim, Norway `jonatan.klemets@itk.ntnu.no`

[2] Morten Hovd is with the Department of Engineering Cybernetics, Norwegian University of Science and Technology, NTNU, NO-7491 Trondheim, Norway `Morten.Hovd@itk.ntnu.no`

can be sent to a fusion center, from which a more accurate global state estimate can be recovered.

In this work, a fusion method for hierarchical decentralized state estimation is proposed, where the local estimators work independently of each other and send their estimates to a fusion center. The overlapping local estimates have been scaled using weighting matrices resulting in a reduction in estimation error once a global estimate has been obtained.

II. Preliminaries

A. Notation

Let $A \in \mathbb{R}^{n\times m}$ denote a real matrix with n rows and m columns. For a matrix A, its transpose is denoted A^T, and A^{-1} denotes its inverse. The identity and the null matrix of appropriate dimension are given by I and $\mathbf{0}$. The notation $A \succeq 0$, $A \preceq 0$ means the matrix is semi-positive and semi-negative definite respectively. The estimate of a stochastic state $x \in \mathbb{R}^n$, can be represented with a Gaussian distribution $x \sim N(\hat{x}, P)$, where $\hat{x} \in \mathbb{R}^n$ is the estimated mean, and $P \in \mathbb{R}^{n\times n}$ is the covariance matrix. A Gaussian distribution can be represented by an ellipsoid $\varepsilon(\hat{x}, P) = \{x|(x-\hat{x})^T P^{-1}(x-\hat{x}) \leq 1\}$.

B. Problem Formulation

Consider a discrete-time system described by a nonlinear dynamic model $f(\cdot)$ and its measurement model $h(\cdot)$:

$$x(k+1) = f(x(k), u(k)) + w(k) \tag{1}$$

$$z(k) = h(x(k)) + v(k) \tag{2}$$

where $x(k) \in \mathbb{R}^{nx}$, $u(k) \in \mathbb{R}^{nu}$, and $z(k) \in \mathbb{R}^{nz}$ are the states, inputs, and measurements respectively. The process noise $w(k)$ and the measurement noise $v(k)$ are uncorrelated zero mean white Gaussian noise with the respective covariance matrices $Q(k)$, and $R(k)$. Decomposing the system into n subsystems of a lower order where index i identifies the i-th subsystem represented as:

$$x_i(k+1) = f_i(x_i(k), u_i(k), X_i(k)) + w_i(k) \tag{3}$$

$$z_i(k) = h_i(x_i(k), X_i(k)) + v_i(k) \tag{4}$$

Here the vector $X_i(k)$ represents the states which are shared among the neighboring subsystems, but for which the model equations are unknown to subsystem i.

Each subsystem uses a decentralized state estimator, that can operate independently, i.e., it does not require any information from any of the other subsystems. For the local estimator to operate without the need to communicate with its neighbors, the local model in (3) is augmented with integrators that represent the interacting states $X_i(k)$. Thus, the augmented model for the local estimators becomes:

$$x_i^a(k+1) = \begin{bmatrix} x_i(k+1) \\ X_i(k+1) \end{bmatrix} = \begin{bmatrix} f_i(x_i(k), u_i(k), X_i(k)) \\ X_i(k) \end{bmatrix} \tag{5}$$

It is assumed that the augmented state vector $x_i^a(k)$ is observable using the locally available measurements $z_i(k)$. The local estimators can be of different type, but they all produce a local estimate together with a covariance matrix that can be represented by $N(\hat{x}_i^a, P_i^a)$. Each of the local estimates $\hat{x}_i^a$ represents a smaller part of the global state vector in (1) where some parts of the local estimates overlap with each other as in Fig. 2. The objective is to collect and fuse the local estimates together in a fusion center, using a hierarchical decentralized structure (see Fig. 1). However, the role of the fusion center is not only to assemble the global state vector, but also to get a more accurate state estimate.

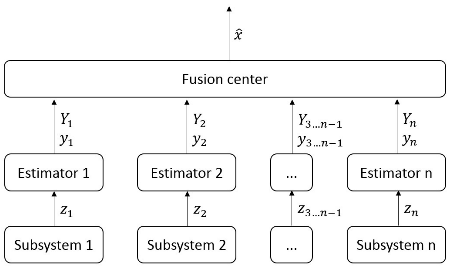

Fig. 1: The hierarchical decentralized state estimator.

III. Hierarchical Decentralized State Estimation

A state vector and its covariance matrix can be represented as an information vector and an information matrix also known as their canonical form [13]. The canonical form is commonly used in distributed and decentralized state estimation, since fusing estimates then becomes equivalent to adding the information matrices and vectors together. The information matrix and information vector are defined as:

$$Y(k|k) = P(k|k)^{-1} \tag{6}$$

$$y(k|k) = P(k|k)^{-1}\hat{x}(k|k) \tag{7}$$

where $\hat{x}(k|k)$ is the state estimate, and $P(k|k)$ is its covariance matrix at time k.

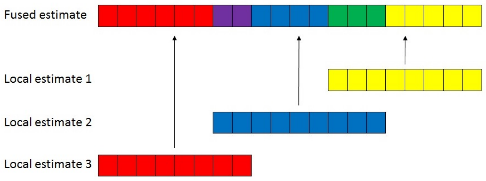

Fig. 2: Fusion of partially overlapping local estimates

Hierarchical state estimation can be done by computing the global information matrix, and vector from n different subsystems:

$$Y(k|k) = Y(k|k-1) + \sum_{i=1}^{n}\Big(Y_i(k|k) - Y_i(k|k-1)\Big) \tag{8}$$

$$y(k|k) = y(k|k-1) + \sum_{i=1}^{n}\Big(y_i(k|k) - y_i(k|k-1)\Big) \tag{9}$$

Here, each local estimator is assumed to have the global model available and tries to estimate the whole state vector using the locally available measurements. This can essentially be interpreted as having a centralized estimator that computes a global *a priori* estimate $y(k|k-1)$, $Y(k|k-1)$, and then receives the additional information containing the measurements from each local estimator. This approach has been shown by [14] to be equivalent to using a centralized state estimator, but the requirement of a global model makes it impractical for large-scale systems. Therefore, instead it would be preferable to fuse all the *a posteriori* information matrices and vectors directly from the local models:

$$Y(k|k) = \sum_{i=1}^{n} H_i Y_i(k|k) H_i^T \tag{10}$$

$$y(k|k) = \sum_{i=1}^{n} H_i y_i(k|k) \tag{11}$$

Where H_i is a transformation matrix, that maps the local state vectors $\hat{x}_i(k|k)$ to the global state vector $\hat{x}(k|k)$, i.e. $\hat{x}_i(k|k) = H_i^T \hat{x}(k|k)$.

The global estimate in (10) and (11) is optimal if there is no correlation between the local estimates. This assumption rarely holds and has therefore been referred to as the "naive approach" [15]. If the cross-correlations are ignored, it will result in overconfident and inconsistent state estimates. For the proposed hierarchical decentralized state estimator, consistency will not be a major issue, since the fused estimates have no influence on the local estimates. However, the cross-correlations should still be considered in order to improve the accuracy of the fused estimate.

IV. Fusion strategies

This section briefly covers some of the existing methods for fusing estimates. From here on the index $(k|k)$, and $(k|k-1)$ will be dropped for ease of notation.

A. State fusion under known correlation

The Bar-Shalom Campo (BC) formula [16] is a well-known method for incorporating the known cross-correlation:

$$\begin{aligned} P^{BC} &= P_1 \qquad\qquad (12)\\ &\quad - (P_1 - P_{12})(P_1 + P_2 - P_{12} - P_{21})^{-1}(P_1 - P_{21}) \\ \hat{x}^{BC} &= (P_2 - P_{21})(P_1 + P_2 - P_{12} - P_{21})^{-1}\hat{x}_1 \qquad (13)\\ &\quad + (P_1 - P_{12})(P_1 + P_2 - P_{12} - P_{21})^{-1}\hat{x}_2 \end{aligned}$$

where P_{12} and P_{21} constitute the cross-correlations. From a Maximum Likelihood sense, this formula results in consistent fusion. However, keeping track on and maintaining these cross-correlations is expensive, especially for large-scale systems. Instead, different suboptimal strategies are often used, that gives a fused solution without the need to maintain the cross-correlations.

B. State fusion under unknown correlation

Several methods exist for fusing estimates with unknown correlations. One of the most popular methods used is known as Covariance Intersection (CI) [17]. It determines the fused estimate by multiplying the information matrices and vectors with a scalar weight, $\omega \in \mathbb{R}_{[0,1]}$:

$$P^{CI} = (\omega \cdot Y_1 + (1-\omega) \cdot Y_2)^{-1} \tag{14}$$

$$\hat{x}^{CI} = P^{CI}(\omega \cdot y_1 + (1-\omega) \cdot y_2) \tag{15}$$

Numerous approaches for determining the weight ω exists, e.g., [18] and [19] but in general, they attempt to minimize the trace or the determinant of P^{CI}. The CI method guarantees that the fused estimate is consistent as long as the local estimates are consistent. It does so by overestimating the covariances in all directions, thus ensuring the fused covariance is larger than for the worst case cross-correlation scenario. This is also one of its drawbacks, as it does not necessarily reduce the estimation error.

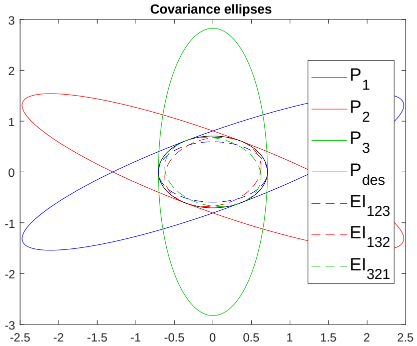

Fig. 3: Covariance ellipses P_1, P_2, and P_3, the desired ellipse P_{des} and the fused covariances EI_{123}, EI_{132}, and EI_{321} when fusing using Ellipsoidal Intersection in different orders.

Contrary to CI, which attempts to obtain a minimum overestimation of the intersection region between covariances, the Ellipsoidal Intersection (EI) [20], and the Largest Ellipsoid Algorithm (LEA) [21] methods aim to find the maximum ellipsoid inside the region of the intersection.

EI ensures that the uncertainty decreases for the fused estimate, and gives a reduction in the estimation error compared to using CI. It does so by calculating the mutual covariance Γ, and the mutual mean γ, yielding:

$$P^{EI} = (P_1^{-1} + P_2^{-1} - \Gamma^{-1})^{-1} \tag{16}$$

$$\hat{x}^{EI} = P^{EI}(P_1^{-1}\hat{x}_1 + P_2^{-1}\hat{x}_2 - \Gamma^{-1}\gamma) \tag{17}$$

Both the EI and LEA methods are similar since they both aim to obtain the maximum ellipsoid inside the region of the intersection. Where they use eigenvalue decomposition, to find a transformation matrix T such that,

$$\tilde{P}_1 = T P_1 T^T \tag{18}$$

$$\tilde{P}_2 = T P_2 T^T \tag{19}$$

becomes diagonal matrices with $\tilde{P}_1$ being equal to the identity matrix. The intersection ellipsoid can then be computed,

$$P^{LEA} = T^{-1} \tilde{P}^{LEA} T^{-T} \tag{20}$$

where $[\tilde{P}^{LEA}]_{j,j} = \min([\tilde{P}_1]_{j,j}, [\tilde{P}_2]_{j,j})$. The similarity between EI and LEA was also addressed in [22]. However, the fused mean in LEA is not adapted to the fused covariance, which is a significant difference compared to EI.

C. State fusion for more than two local estimates

The framework of the three fusion methods (CI, EI, and LEA) have been devised for fusing two estimates only. When extended to multiple estimates the general recommendation is to sequentially apply these fusion methods [23], [24]. However, as demonstrated in [25] when using CI for three or more estimates it tends to overestimate the covariance, and therefore, the minimum overestimate no longer holds. On the other hand, using a sequential approach for EI, and LEA, may lead to an underestimate of the uncertainty, that is, result in a covariance that is smaller than the maximum ellipsoid inside the intersection region as seen in Fig 3. Furthermore, the order of the sequence of which the fusion is done effects the final estimates as also illustrated in Fig 3, where the three covariances P_1, P_2, and P_3 end up with a different fused covariance depending on the sequential order.

D. State fusion for partially overlapping state estimates

Compared to fusing state estimates that correspond to the same state vector, research regarding fusing partial overlapping state vectors is fairly limited. A consensus based method for fusing the overlapping parts have been proposed in [12]. In [26] an empirical method was investigated, where the EI method was extended to deal with partially overlapping state vectors. However, both these methods are better suited to fully distributed state estimation, where the local estimates are shared between its neighbors without any fusion center.

A more suitable technique for the desired hierarchical fusion structure was proposed in [27]. There, fusing unequal state vectors is treated as a Weighted Least Squares (WLS) problem, where the fused state estimate becomes:

$$\hat{x}^{WLS} = K \begin{bmatrix} \hat{x}_1^T & \cdots & \hat{x}_n^T \end{bmatrix}^T \tag{21}$$

The gain K for unknown cross-correlations can be computed:

$$K = (H^T Y^{WLS} H)^{-1} H^T (Y^{WLS})^{-1}. \tag{22}$$

where $H := \begin{bmatrix} H_1^T \\ \vdots \\ H_n^T \end{bmatrix}$, and $Y^{WLS} := \begin{bmatrix} \omega_1 Y_1 & & \\ & \ddots & \\ & & \omega_n Y_n \end{bmatrix}$.

The scalar weights ω_i can be obtained using, e.g., CI such that $\omega_i \geq 0$, and $\sum_{i=1}^{n} \omega_i = 1$. However, no more information was given on how to compute ω_i. Furthermore, CI is not particularly well suited for dealing with partially overlapping state estimates. For instance, if the goal is to minimize the trace of the fused covariance, then using larger weights for the local estimates of lower dimensions would in general, be beneficial, even if these estimates are inaccurate.

V. Proposed fusion method

Motivated by some of the shortcomings when fusing multiple and partially overlapping state estimates, a method is proposed that tries to address some of these issues. The proposed method is inspired by EI [20] and LEA [21], as it tries to obtain a fused covariance that corresponds to the maximum ellipsoid inside the region of intersection. Therefore, similar to EI, it does not guarantee consistency [28] unless the local estimates are weakly correlated [24]. However, instead of using a sequential approach, the aim is to compute static weight matrices W_i such that the global state estimate becomes,

$$\hat{x} = Y^{-1} y \tag{23}$$

where,

$$Y = \sum_{i=1}^{n} H_i W_i Y_i W_i^T H_i^T \tag{24}$$

$$y = \sum_{i=1}^{n} H_i W_i Y_i W_i^T \hat{x}_i. \tag{25}$$

The weights W_i in (24), and (25) will be obtained by solving two semidefinite programming (SDP) problems.

Proposition 5.1: [29] An inner ellipsoidal approximation of the intersection for the ellipsoids $H_1 Y_1 H_1^T, ..., H_n Y_n H_n^T$, can be obtained by solving the following Linear Matrix Inequality (LMI):

$$\max_{Z, \lambda_i} \ (\det(Z))^{1/nx} \tag{26}$$

$$\text{subject to:} \quad \lambda_i \geq 0, \qquad \forall i \tag{27}$$

$$\begin{pmatrix} I & \mathbf{0} & H_i Y_i^{\frac{1}{2}} H_i^T Z \\ \mathbf{0} & 1 - \lambda_i & \mathbf{0} \\ Z H_i Y_i^{\frac{1}{2}} H_i^T & \mathbf{0} & \lambda_i I \end{pmatrix} \succeq 0, \ \forall i \tag{28}$$

The approximation of the intersection, Y_0 is calculated from:

$$Y_0 = Z^{-2} \tag{29}$$

The obtained Y_0 from (29) represents the desired covariance (e.g., $P_{\text{des}} = Y_0^{-1}$ in Fig. 3), but it doesn't provide any information on how to choose the weights W_i. E.g., assuming $H_i = I, \ \forall i$, then selecting W_i such that $W_1 Y_1 W_1^T = Y_0$ and setting $W_i Y_i W_i^T = 0$ for $i = 2...n$, would result in (24) becoming equivalent to Y_0, but it wouldn't necessarily improve the global estimate. Instead, W_i should be chosen such that (24) becomes equal (or close) to Y_0, but more importantly, so that it reduces the estimation error by emphasizing the more accurate local estimates. Therefore, Y_0 will be used for finding transformation matrices T_i, that diagonalizes and manipulates the orientation of Y_i. If the matrices in (24) are diagonalized and aligned with Y_0, then W_i can be selected such that it scales the variances in the desired directions.

The singular value composition of Y_0 is given by

$$[U_0, D_0] = \text{svd}(Y_0), \tag{30}$$

from which, a transformation matrix can be defined,

$$T_0 = D_0^{-\frac{1}{2}} U_0^T \tag{31}$$

that makes $T_0 Y_0 T_0^T = I$. The following transformation matrices can be computed for $i = 1, ..., n$:

$$[U_i, D_i] = \text{svd}(T_0 H_i Y_i H_i^T T_0^T) \tag{32}$$

$$T_i = U_i^T D_0^{-\frac{1}{2}} U_0^T \tag{33}$$

The transformation matrices, T_i can be used to bring Y_0 and Y_i within a space where they have compatible orientations:

$$\tilde{Y_0} = T_i Y_0 T_i^T = I \tag{34}$$
$$\tilde{Y_i} = T_i H_i Y_i H_i^T T_i^T \tag{35}$$

where $\tilde{Y_0}$ and $\tilde{Y_i}$ are diagonal matrices. Next, the goal is to find some diagonal scaling weights $\tilde{W}_i$, such that the corresponding ellipsoid when transformed back to original space becomes:

$$Y_0 = \sum_{i=1}^{n} T_i^{-1} \tilde{W}_i \tilde{Y}_i T_i^{-T} \tag{36}$$
$$= \sum_{i=1}^{n} T_i^{-1} \tilde{W}_i T_i H_i Y_i H_i^T T_i^T T_i^{-T} \tag{37}$$
$$= \sum_{i=1}^{n} T_i^{-1} \tilde{W}_i T_i H_i Y_i H_i^T \tag{38}$$

Theorem 5.1: The weights $\tilde{W}_i$ can be obtained by solving:

$$\max_{\tilde{W}_i} \left(\det\left(\sum_{i=1}^{n} T_i^{-1} \tilde{W}_i T_i H_i Y_i H_i^T\right)\right)^{1/nx} \tag{39}$$
$$\text{subject to:} \quad \tilde{W}_i \geq 0, \qquad \forall i \tag{40}$$
$$\left\| \sum_{i=1}^{n} T_i^{-1} \tilde{W}_i T_i \right\|_2 \leq 1 \tag{41}$$

Proof: From (38), $(T_i^{-1} \tilde{W}_i T_i)$ can be considered as a weight matrix, that when multiplied with $H_i Y_i H_i^T$ results in:

$$(T_i^{-1} \tilde{W}_i T_i) H_i Y_i H_i^T = H_i Y_i H_i^T (T_i^{-1} \tilde{W}_i T_i)^T \succeq 0 \tag{42}$$

The symmetry is due to $\tilde{W}_i$, and $\tilde{Y}_i$ in (36) being diagonal matrices and the constraint in (40) guarantees semi-positive definiteness. The upper bound placed on $\tilde{W}_i$ through (41) ensure that $\sum_{i=1}^{n} T_i^{-1} \tilde{W}_i T_i H_i Y_i H_i^T \preceq Y_0$. Therefore, the weights $\tilde{W}_i$ are optimal if $\sum_{i=1}^{n} T_i^{-1} \tilde{W}_i T_i = I$, which results in $\sum_{i=1}^{n} T_i^{-1} \tilde{W}_i T_i H_i Y_i H_i^T = Y_0$ ■

Remark 5.1: Ideally, the constraint in (41) could be replaced with $\sum_{i=1}^{n} T_i^{-1} \tilde{W}_i T_i = I$. However, due to possible round-off error and other numerical issues when computing T_i, it can give poor results. Therefore, the relaxed constraint in (41) is preferred and should result in $\sum_{i=1}^{n} T_i^{-1} \tilde{W}_i T_i \approx I$.

Because $\tilde{W}_i$ are diagonal matrices, (36) can be written as

$$Y_0 = \sum_{i=1}^{n} T_i^{-1} \tilde{W}_i^{\frac{1}{2}} \tilde{Y}_i \tilde{W}_i^{\frac{T}{2}} T_i^{-T} \tag{43}$$
$$= \sum_{i=1}^{n} T_i^{-1} \tilde{W}_i^{\frac{1}{2}} T_i H_i Y_i H_i^T T_i^T \tilde{W}_i^{\frac{T}{2}} T_i^{-T} \tag{44}$$

The weights in the original space then become:

$$W_i = H_i^T T_i^{-1} \tilde{W}_i^{\frac{1}{2}} T_i H_i, \quad \forall i. \tag{45}$$

An illustration of the proposed fusion method can be seen in Fig. 4. The individual covariances get scaled using the weights in (45), so the uncertainty mainly increases in the inaccurate directions. This forces the fused estimate to put more emphasis on the accurate estimates.

The algorithm for computing W_i can be summarized as:

Algorithm

1) Compute the maximum ellipsoid inside the region of intersection Y_0 for $H_1 Y_1 H_1^T, \ldots, H_n Y_n H_n^T$ by solving the LMI in (26), subject to (27), and (28).
2) Obtain the transformation matrices T_i that aligns Y_0 with Y_i for all i, from (30), (31) (32), and (33).
3) Compute the diagonal weight matrices $\tilde{W}_i$ by solving the LMI in (39) subject to (40), and (41).
4) Calculate the weights W_i in original space from (45).

The global state estimate can then be obtained from (23), (24), and (25). Alternatively, (21), and (22) can be used, where $\omega_i Y_i$ in Y^{WLS} can be replaced with $W_i Y_i W_i^T$.

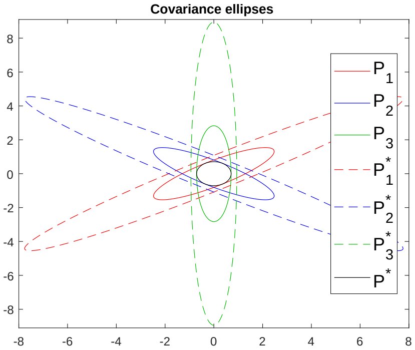

Fig. 4: Covariance ellipses P_1, P_2, and P_3, the weighted covariances $P_1^* = (W_1 Y_1 W_1^T)^{-1}$, $P_2^* = (W_2 Y_2 W_2^T)^{-1}$, $P_3^* = (W_3 Y_3 W_3^T)^{-1}$ and their fused covariance, $P^* = (W_1 Y_1 W_1^T + W_2 Y_2 W_2^T + W_3 Y_3 W_3^T)^{-1}$.

The main drawback with the proposed algorithm is that it requires solving two SDPs, which can be computational demanding, especially for large-scale systems. However, unless there are significant changes in the covariance matrices from the local estimates (due to, e.g., high non-linearities), updating the weights at every iteration should not be necessary. Instead, the weights can be kept constant until the operating conditions in the process changes.

VI. Simulations

A. Lorenz Attractor

Let's consider three estimators, that tries to estimate the states of the Lorenz attractor,

$$\dot{x}_1 = -\sigma(x_1 - x_2) \tag{46}$$
$$\dot{x}_2 = x_1(\rho - x_3) - x_2 \tag{47}$$
$$\dot{x}_3 = x_1 x_2 - \beta x_3 \tag{48}$$
$$z_i = h_i \begin{bmatrix} x_1 & x_2 & x_3 \end{bmatrix}^T \tag{49}$$

where $\sigma = 10$, $\rho = 28$, and $\beta = 8/3$. Each estimator uses an Unscented Kalman filter (UKF) [30] with the measurements: $h_1 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$, $h_2 = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$, $h_3 = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix}$. The covariance matrices for the process and measurement noise are assumed to be known and are for $i = 1, 2, 3$:

$$Q_i = \text{diag}(\begin{bmatrix} 0.5^2 & 0.5^2 & 0.5^2 \end{bmatrix}), \quad R_i = \text{diag}(\begin{bmatrix} 0.01^2 & 0.1^2 \end{bmatrix})$$

TABLE I: RMSE for the Lorenz attractor

	RMSE x_1	RMSE x_2	RMSE x_3
Local Est. 1	**0.0101**	0.0969	1.7172
Local Est. 2	1.1101	**0.0101**	0.0992
Local Est. 3	0.0968	1.4670	**0.0103**
Fast CI	0.1315	0.1036	0.0905
EI_{123}	0.0108	**0.0101**	0.0109
EI_{132}	**0.0101**	0.0106	0.0107
EI_{321}	0.0110	0.0112	**0.0103**
Proposed (updated W_i)	**0.0101**	**0.0101**	**0.0103**
Proposed (static W_i)	0.0101	0.0101	0.0104

TABLE II: Parameters for the CSTR and Separator

$x_{A1} = 0.923$ wt(%)	$T_1 = 315.1$ K	$F_1 = 74.5$ kg/s
$x_{B1} = 0.074$ wt(%)	$T_2 = 315.2$ K	$F_{10} = 8.3$ kg/s
$x_{A2} = 0.919$ wt(%)	$T_3 = 314.9$ K	$F_2 = 75.0$ kg/s
$x_{B2} = 0.081$ wt(%)	$Q_1 = 10.0$ kJ/s	$F_{20} = 0.5$ kg/s
$x_{A3} = 0.806$ wt(%)	$Q_2 = 10.0$ kJ/s	$F_3 = 8.0$ kg/s
$x_{B3} = 0.184$ wt(%)	$Q_3 = 10.0$ kJ/s	$F_r = 66.2$ kg/s
$x_{A10} = 0.90$ wt(%)	$V_1 = 89.4$ m^3	$F_p = 0.8$ kg/s
$x_{A20} = 0.80$ wt(%)	$V_2 = 90.0$ m^3	$c_p = 25$ kJ/kg K
$E_1/R = -100$ K	$V_3 = 3.27$ m^3	$\alpha_A = 3.5$
$E_2/R = -150$ K	$k_1 = 0.020$ 1/s	$\alpha_B = 1.1$
$\Delta H_1 = -40$ kJ/kg	$k_2 = 0.018$ 1/s	$\alpha_C = 0.5$
$\Delta H_2 = -50$ kJ/kg	$\rho = 0.15$ kg/m^2	$T_{10}, T_{20} = 315$ K

The Lorenz attractor is simulated for 1000 samples with a step size of 0.01 using a 4th order Runge-Kutta numerical integration. To improve the result, the estimated states from the local estimators are fused, using a hierarchical structure as in Fig. 2, but with all states overlapping. The cross-correlations between the estimates are assumed to be unknown, and thus the estimation error for the fast CI [19], the EI, and the proposed method are compared. The root mean squared error (RMSE) for the respective estimator and the three fusion methods can be seen in Table I, where the subscript of 'EI' denotes the sequential order the estimates have been fused.

As seen in Table I, all three fusion methods give an overall reduction in the estimation error compared to the local estimates. However, the proposed method gives a better result compared to the two other methods, where there is little improvement when updating the weights at every iteration compared to using the same (static) weights. It can also be seen that the estimation error for the EI method depends on the order which the local estimates have been fused in.

B. CSTR and Flash Separator

A chemical process consisting of two continuous stirred tank reactors (CSTRs) and a flash separator connected in series is considered as seen in Fig 5. A mixture of A and B is fed into the two CSTRs, in which two reactions occur, i.e., $A \to B$ and $B \to C$. The outlet of the second CSTR is fed into the flash separator at a flow rate F_2. The overhead of the separator is condensed and passed to a downstream unit at flow rate F_r and the bottom product stream is removed at flow rate F_3. The dynamic model obtained via mass and energy balances are the same as in [31], but the flows and thus the volumes are assumed to be constant resulting in:

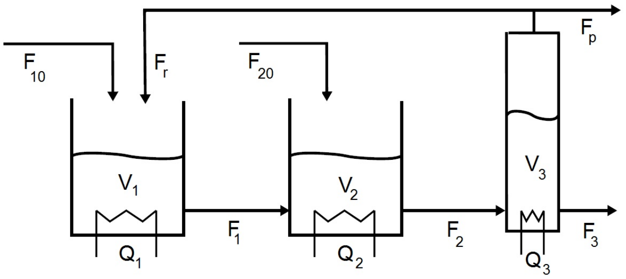

Fig. 5: Diagram of 2 CSTR and Flash Separator.

CSTR 1 (Estimator 1)

$$\frac{dx_{A1}}{dt} = \frac{F_{10}x_{A10} + F_r x_{Ar} - F_1 x_{A1}}{\rho V_1} - k_1 e^{\frac{-E_1}{RT_1}} x_{A1} \tag{50}$$

$$\frac{dx_{B1}}{dt} = \frac{F_{10}(1 - x_{A10}) + F_r x_{Br} - F_1 x_{B1}}{\rho V_1} + k_1 e^{\frac{-E_1}{RT_1}} x_{A1} - k_2 e^{\frac{-E_2}{RT_1}} x_{B1} \tag{51}$$

$$\frac{dT_1}{dt} = \frac{F_{10}T_{10} + F_r T_3 - F_1 T_1}{\rho V_1} - \frac{\Delta H_1}{c_p} k_1 e^{\frac{-E_1}{RT_1}} x_{A1} - \frac{\Delta H_2}{c_p} k_2 e^{\frac{-E_2}{RT_1}} x_{B1} + \frac{Q_1}{\rho c_p V_1} \tag{52}$$

CSTR 2 (Estimator 2)

$$\frac{dx_{A2}}{dt} = \frac{F_{20}x_{A20} + F_1 x_{A1} - F_2 x_{A2}}{\rho V_2} - k_1 e^{\frac{-E_1}{RT_2}} x_{A2} \tag{53}$$

$$\frac{dx_{B2}}{dt} = \frac{F_{20}(1 - x_{A20}) + F_1 x_{B1} - F_2 x_{B2}}{\rho V_2} + k_1 e^{\frac{-E_1}{RT_2}} x_{A2} - k_2 e^{\frac{-E_2}{RT_2}} x_{B2} \tag{54}$$

$$\frac{dT_2}{dt} = \frac{F_{20}T_{20} + F_1 T_1 - F_2 T_2}{\rho V_2} - \frac{\Delta H_1}{c_p} k_1 e^{\frac{-E_1}{RT_2}} x_{A2} - \frac{\Delta H_2}{c_p} k_2 e^{\frac{-E_2}{RT_2}} x_{B2} + \frac{Q_2}{\rho c_p V_2} \tag{55}$$

Separator (Estimator 3)

$$\frac{dx_{A3}}{dt} = \frac{F_2 x_{A2} - (F_p + F_r)x_{Ar} - F_3 x_{A3}}{\rho V_1} \tag{56}$$

$$\frac{dx_{B3}}{dt} = \frac{F_2 x_{B2} - (F_p + F_r)x_{Br} - F_3 x_{B3}}{\rho V_1} \tag{57}$$

$$\frac{dT_3}{dt} = \frac{F_2 T_2 - (F_p + F_r + F_3)T_3}{\rho V_1} + \frac{Q_3}{\rho c_p V_3} \tag{58}$$

with the algebraic equations:

$$x_{Ar} = \frac{\alpha_A x_{A3}}{\bar{x}_3}, \quad x_{Br} = \frac{\alpha_B x_{B3}}{\bar{x}_3}, \tag{59}$$

$$\bar{x}_3 = \alpha_A x_{A3} + \alpha_B x_{B3} + \alpha_C(1 - x_{A3} - x_{B3}). \tag{60}$$

The process is split into three local estimators; one for each of the reactors, and one for the separator. The available measurements z_i, and inputs u_i for the estimators are:

$$z_1 = \begin{bmatrix} x_{A3} \\ x_{B1} \\ T_1 \end{bmatrix}, \; z_2 = \begin{bmatrix} x_{A1} \\ x_{B2} \\ T_2 \end{bmatrix}, \; z_3 = \begin{bmatrix} x_{A2} \\ x_{B3} \\ T_3 \end{bmatrix}, \; \begin{matrix} u_1 = x_{A10} \\ u_2 = x_{A20} \end{matrix}.$$

The process and measurement noise are given by:

$$Q_i = \operatorname{diag}(\begin{bmatrix}0.005^2 & 0.005^2 & 0.1^2\end{bmatrix})$$
$$R_i = \operatorname{diag}(\begin{bmatrix}0.0002^2 & 0.0001^2 & 0.1^2\end{bmatrix})$$

for $i = 1, 2, 3$. All three models are augmented with three states as in (5) with x_{Ai}, x_{Bi}, and T_i from the previous estimator. The parameters and the steady-state values are shown in Table II. A UKF was used for all the local estimators, and the process is simulated for 100s with a sampling time of 0.5s, where the inputs u_1 and u_2 are varying with two different sinusoidal signals.

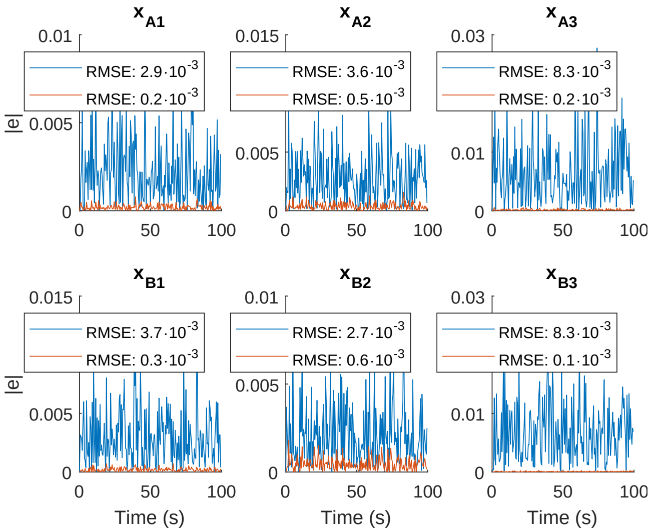

Fig. 6: RMSE for the 2 CSTR and reactor process. The blue line is the method in [27] with $\omega_1 = \omega_2 = \omega_3 = 1/3$, and the red line is using the proposed method.

An EI method for fusing unequal state vectors was presented in [26], where the overlapping parts get marginalized out and shared with the neighboring nodes. This gives an improvement in the local estimates, but it doesn't provide a global state vector and requires a different structure than the hierarchical decentralized state estimator. Instead, the proposed fusion method is compared to using (21), and (22) with the scalar weights, $\omega_1 = \omega_2 = \omega_3 = 1/3$, where the resulting RMSE for the compositions can be seen in Fig. 6.

VII. Conclusion

A method for fusing multiple and partially overlapping state estimates with unknown cross-correlation has been proposed. The method calculates weight matrices that scales the local estimates such when fused together gives a more accurate estimate. Simulations showed a reduction in the estimation error compared to other existing methods.

References

[1] X. Zhu and S. Wang, "State estimation for nonlinear systems by using dynamic optimization," *IEEE Networking Sensing and Control Proceedings*, pp. 918–922, 2005.

[2] M. Mansouri, H. Nounou, and M. Nounou, "State estimation of a chemical reactor process model a comparative study," *10th Int. Multi-Conference on Systems, Signals & Devices*, pp. 1–6, 2013.

[3] C. Venkateswarlu, "Advances in monitoring and state estimation of bioreactors," *Journal of Sci. and Ind. Res.*, vol. 63, pp. 491–498, 2004.

[4] R. Vadigepalli and F. J. Doyle, "A distributed state estimation and control algorithm for plantwide processes," *IEEE Transactions on Control Systems Technology*, vol. 11(1), pp. 119–127, 2003.

[5] D. M. Falcao, F. F. Wu, and L. Murphy, "Parallel and distributed state estimation," *IEEE Trans. on Power Sys.*, vol. 10(2), pp. 724–730, 1995.

[6] Y. Zhu *et al.*, "The optimality for the distributed Kalman filtering fusion with feedback," *Automatica*, vol. 37(9), pp. 1489–1493, 2001.

[7] M. Farina, G. Ferrari-Trecate, and R. Scattolini, "Moving-horizon partition-based state estimation of large-scale systems," *Automatica*, vol. 46(5), pp. 910–918, 2010.

[8] R. Olfati-Saber, "Distributed Kalman filtering for sensor networks," *Proc. 46th IEEE Conf. on Decision and Control*, vol. 54(2), 2007.

[9] W. Li *et al.*, "Weighted average consensus-based unscented Kalman filtering," *IEEE Trans. on Cybernetics*, vol. 46(2), pp. 558–567, 2016.

[10] ——, "A survey on multisensor fusion and consensus filtering for sensor networks," *Discrete Dyn. Nat. Soc.*, vol. 2015, pp. 1–12, 2015.

[11] A. T. Kamal *et al.*, "Information weighted consensus filters and their application in distributed camera networks," *IEEE Transactions on Automatic Control*, vol. 58(12), pp. 3112–3125, 2013.

[12] S. S. Stankovic, M. S. Stankovic, and D. M. Stipanovic, "Consensus based overlapping decentralized estimator," *IEEE Transactions on Automatic Control*, vol. 54(2), pp. 410–415, 2009.

[13] Y. Bar-Shalom *et al.*, Eds., *Estimation with Applications to Tracking and Navigation: Theory, Algorithms, and Software*. NY: Wiley, 2001.

[14] H. R. Hashemipour, S. Roy, and A. J. Laub, "Decentralized structures for parallel Kalman filtering," *IEEE Transactions on Automatic Control*, vol. 33(1), pp. 88–94, 1988.

[15] K. C. Chang, C. Chong, and S. Mori, "Analytical and computational evaluation of scalable distributed fusion algorithms," *IEEE Trans. on Aerospace and Electronic Systems*, vol. 46(4), pp. 2022–2034, 2010.

[16] Y. Bar-Shalom and L. Campo, "The effect of the common process noise on the two-sensor fused-track covariance," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 22(6), 1986.

[17] S. J. Julier and J. K. Uhlmann, "A non-divergent estimation algorithm in the presence of unknown correlations," *Proceedings of the American Control Conference*, vol. 4, pp. 2369–2373, 1997.

[18] L. Chen *et al.*, "Fusion under unknown correlation - covariance intersection as a special case," *Proceedings of the 5th International Conference on Information Fusion*, vol. 2, pp. 905–912, 2002.

[19] D. Franken and A. Hupper, "Improved fast covariance intersection for distributed data fusion," *Proceedings of the 7th International Conference on Information Fusion*, 2005.

[20] J. Sijs and M. Lazar, "State fusion with unknown correlation: Ellipsoidal intersection," *Automatica*, vol. 48(8), pp. 1874–1878, 2012.

[21] A. R. Benaskeur, "Consistent fusion of correlated data sources," *Proceedings of the 28th Annual Conference of the IEEE Industrial Electronics Society*, vol. 4, pp. 2652–2656, 2002.

[22] B. Noack, J. Sijs, and U. D. Hanebeck, "Algebraic analysis of data fusion with ellipsoidal intersection," *in Proc. of the 2016 IEEE Int. Conf. on Multisensor Fusion and Integration for Intelligent Sys.*, 2016.

[23] Z. Deng and other, "Sequential covariance intersection fusion Kalman filter," *Information Sciences*, vol. 189(15), pp. 293–309, 2012.

[24] B. Liu, X. Zhan, and Z. H. Zhu, "Multisensor parallel largest ellipsoid distributed data fusion with unknown cross-covariances," *Sensors*, vol. 17(7), 2017.

[25] M. A. Bakr and S. Lee, "Distributed multisensor data fusion under unknown correlation and data inconsistency," *Sensors*, vol. 17(11), 2017.

[26] J. Sijs, U. Hanebeck, and B. Noack, "An empirical method to fuse partially overlapping state vectors for distributed state estimation," *Proceedings of the 2013 European Control Conference*, 2013.

[27] B. Noack, J. Sijs, and U. D. Hanebeck, "Fusion strategies for unequal state vectors in distributed Kalman filtering," *Proceedings of 19th IFAC World Congress*, vol. 47(19), pp. 3262–3267, 2014.

[28] B. Noacka *et al.*, "Decentralized data fusion with inverse covariance intersection," *Automatica*, vol. 79, pp. 35–41, 2017.

[29] A. Ben-Tal and A. Nemirovski, Eds., *Lectures on Modern Convex Optimization: Analysis, Algorithms, and Engineering Applications*. Philadelphia: MPS-SIAM Series on Optimization, 2001.

[30] S. J. Julier and J. Uhlmann, "Unscented filtering and nonlinear estimation," *Proceedings of the IEEE*, vol. 92(3), pp. 401–422, 2004.

[31] B. T. Stewart *et al.*, "Cooperative distributed model predictive control," *Systems and Control Letters*, vol. 59(8), pp. 460–469, 2010.