Distributed and Decentralized State Estimation of Fractional Order Systems

Martin Kupper*†, Iñigo Sesar Gil* and Sören Hohmann*
*Institute of Control Systems, Karlsruhe Institute of Technology, Karlsruhe, Germany
Email: martin.kupper@kit.edu, inigo.sesar-gil@kit.edu, soeren.hohmann@kit.edu
†ITK Engineering AG, Rülzheim, Germany

***Abstract*—This paper is about the decentralization and distribution of a Kalman filter for fractional order systems. A fractional order discrete state space for a global system is introduced and divided into different submodules. The distribution of the model and of the state estimation algorithm into submodules leads to small and scalable units, which do not need a central processing node. Each submodule performs its computation locally. All information required by other nodes is communicated between the nodes directly. Finally, an example is given to compare the fractional Kalman filter (FKF) for the overall system with the distributed and decentralized fractional Kalman filter (DDFKF).**

I. Introduction

A. Fractional Systems

In the last years, several applications of fractional calculus have been developed. For example fractional calculus has introduced improvements in the fields of electromagnetism, the diffusion and wave propagation, heat transmission, biology, traffic systems and economics [1].

The main benefits of fractional calculus are the possibility to obtain mathematical models, which describe the results closer to the experimental measurements, the ability to predict more accurately the dynamical behavior of physical systems and the possibility to obtain simplified models with just a few physically motivated parameters [2]–[4].

A fractional order dynamic model is also useful for modelling electrodynamic processes. A remarkable advantage is that it allows to introduce the non-linear effects like friction and slipping in an easier way than any other dynamic model of integer order [5], [6]. The fractional model forms a basis for the state-feedback control, but if the states are not measurable, an estimation tool suitable for fractional systems is required. The state identification problem in fractional systems is more complex than in integer order systems. For example in [1] a fractional state variable filter and a fractional Kalman filter is developed.

Another application of fractional calculus are approaches for the estimation of Li-ion battery parameters which avoid integer order approximations. It is important to estimate several parameters, such as the state of charge (SOC), state of health (SOH), voltages and temperatures with high accuracy. This is essential in order to increase the power, reliability and expected life of the battery. An example of a fractional order battery model is described in [3]. That model results from a simplification of an electrochemical model that describes the battery behaviour using partial differential equations. Instead of using integer order partial differential equations, a simpler fractional order model is used in order to approximate the experimental measurements accurately. In [4] a fractional battery model is introduced using impedance measurements. A late-lumping parameter identification method for a fractional battery model is presented in [7].

B. Distribution and Decentralization

Until the 1960s, all calculations in a computer system were performed by a central processor. The sensors of a system transmitted the signals to the processor where all the information was processed. However, this centralization presents disadvantages for large scale systems. On the one hand, all measurements must be available in a central processing unit in order to proceed with the algorithm which can lead to communication bottlenecks. On the other hand, the complete state-vector has to be estimated, which is computationally intensive. These properties can cause some problems in practical applications [8]. A solution is a functional decentralization and distribution of the system. So it is necessary to develop a decentralized and distributed state estimation method for large scale fractional order systems which is the subject of this paper. An example for a large scale fractional order system are battery packs where many cells are linked together.

II. Fractional Calculus

The Fractional Kalman Filter (FKF) [9] is defined for discrete-time fractional order systems and therefore we introduce the Grünwald-Letnikov approximation of a fractional derivative:

$$ {}_0D^q_{t_k} x(t_k) = \Delta^q x_k = \frac{1}{h^q} \sum_{j=0}^{k} (-1)^j \binom{q}{j} x_{k-j} \qquad (1)$$

where $q \in \mathbb{R}^+$ is the order of the fractional derivative, k is the current sample of the sequence for which the derivative is calculated and h is the sampling interval. The symbol D is used to represent the fractional derivative of a function with the lower bound 0 and the upper bound t_k. This section deals with difference equations that use discrete sequences, and not with differential equations based on continuous functions. Therefore, according to the formula of the q-th order difference in equation (1), and also for simplicity reasons the value of h is set to $h = 1$, similar to [9].

$$\Delta^q x_k = x_k + \sum_{j=1}^{k} (-1)^j \binom{q}{j} x_{k-j} \tag{2}$$

Isolating the term x_k in the last equation, equation (3) is obtained.

$$x_k = \Delta^q x_k - \sum_{j=1}^{k} (-1)^j \binom{q}{j} x_{k-j} \tag{3}$$

Based on equation (3), the linear stochastic discrete fractional order state space system can be obtained, which is defined by equations (4) to (8) [9]:

$$\begin{align}
\Delta^\gamma \boldsymbol{x}_k &= \boldsymbol{A}\boldsymbol{x}_{k-1} + \boldsymbol{B}\boldsymbol{u}_{k-1} + \boldsymbol{\omega}_{k-1} \tag{4}\\
\boldsymbol{x}_k &= \Delta^\gamma \boldsymbol{x}_k - \sum_{j=1}^{k} (-1)^j \boldsymbol{\Gamma}_j \boldsymbol{x}_{k-j} \tag{5}\\
\boldsymbol{y}_k &= \boldsymbol{C}\boldsymbol{x}_k + \boldsymbol{\nu}_k \tag{6}\\
\boldsymbol{\Gamma}_j &= \text{diag}\left[\binom{q_1}{j}, \cdots, \binom{q_N}{j}\right] \tag{7}\\
\Delta^\gamma \boldsymbol{x}_k &= \begin{bmatrix} \Delta^{q_1} \boldsymbol{x}_{1,k} \\ \vdots \\ \Delta^{q_N} \boldsymbol{x}_{N,k} \end{bmatrix} \tag{8}
\end{align}$$

where $\boldsymbol{x}_k$ is the state vector, $\boldsymbol{u}_k$ is the system input, $\boldsymbol{y}_k$ is the system output, $\boldsymbol{\omega}_k$ is the system noise, $\boldsymbol{\nu}_k$ is the output noise and $q_1, \ldots, q_N$ are the orders of the particular system equations.

As described in [10] and [11], the initial values for the states have to be given from $t = -\infty$ to the current time $t = t_k = kh$ in order to calculate the fractional derivative. Therefore the sums in equations (1), (2), (3) and (5) have to consider the values from $j = 1$ to $j = \infty$. On the one hand in the practical implementation of such a filter the past values are usually unknown, so that we can deal only with the data given from $t = 0$ on. We also suppose that the estimation algorithm starts while the system is in a position of rest. On the other hand, the memory length B_L in a particular processor is limited. Due to the weighting of the past values with the binomial coefficients, we can neglect values that lie farther in the past which is known as *short memory principle*. In [9] it was shown that depending on the sampling time and system time constants a value for B_L can be found, which shows enough accuracy so that a further increase gives no additional advantage. For the following sections, it can be assumed that the buffer length is chosen sufficiently large so that all relevant values are considered. It also should be noted, that according to [11] the representation of the fractional system given by equations (4) to (8) builds not a classical state space description. It is rather a pseudo state space description, because it depends not only on the current state $\boldsymbol{x}_k$, but also on the values of $\boldsymbol{x}$ from the past. However, for simplicity reasons, in the following sections the system described above is named ”state space system”.

III. Fractional Kalman Filter

The recursive algorithm of the FKF for a linear stochastic discrete fractional order state space is given by equations (9) to (16) [9].

$$\begin{align}
\Delta^\gamma \hat{\boldsymbol{x}}_k^- &= \boldsymbol{A}\hat{\boldsymbol{x}}_{k-1}^+ + \boldsymbol{B}\boldsymbol{u}_{k-1} \tag{9}\\
\hat{\boldsymbol{x}}_k^- &= \Delta^\gamma \hat{\boldsymbol{x}}_k^- - \sum_{j=1}^{k} (-1)^j \boldsymbol{\Gamma}_j \hat{\boldsymbol{x}}_{k-j}^+ \tag{10}\\
\boldsymbol{P}_k^- &= (\boldsymbol{A} + \boldsymbol{\Gamma}_1)\boldsymbol{P}_{k-1}^+(\boldsymbol{A} + \boldsymbol{\Gamma}_1)^T \nonumber\\
&\quad + \boldsymbol{Q}_{k-1} + \sum_{j=2}^{k} \boldsymbol{\Gamma}_j \boldsymbol{P}_{k-j}^+ \boldsymbol{\Gamma}_j^T \tag{11}\\
\boldsymbol{K}_k &= \boldsymbol{P}_k^- \boldsymbol{C}_k^T (\boldsymbol{C}_k \boldsymbol{P}_k^- \boldsymbol{C}_k^T + \boldsymbol{R}_k)^{-1} \tag{12}\\
\hat{\boldsymbol{x}}_k^+ &= \hat{\boldsymbol{x}}_k^- + \boldsymbol{K}_k(\boldsymbol{y}_k - \boldsymbol{C}\hat{\boldsymbol{x}}_k^-) \tag{13}\\
\boldsymbol{P}_k^+ &= (\boldsymbol{I} - \boldsymbol{K}_k\boldsymbol{C})\boldsymbol{P}_k^-(\boldsymbol{I} - \boldsymbol{K}_k\boldsymbol{C})^T + \boldsymbol{K}_k\boldsymbol{R}_k\boldsymbol{K}_k^T \nonumber\\
&= (\boldsymbol{I} - \boldsymbol{K}_k\boldsymbol{C})\boldsymbol{P}_k^- \tag{14}
\end{align}$$

which is usually initialized with

$$\begin{align}
\hat{\boldsymbol{x}}_0^+ &= E\{\boldsymbol{x}_0\} \approx \boldsymbol{0}, \tag{15}\\
\boldsymbol{P}_0^+ &= E\{(\boldsymbol{x}_0 - \hat{\boldsymbol{x}}_0^+)(\boldsymbol{x}_0 - \hat{\boldsymbol{x}}_0^+)^T\} \approx (100 \ldots 1000)\boldsymbol{I} \tag{16}
\end{align}$$

when the initial values are unknown. $(*)^-$ presents the a priori estimation and $(*)^+$ the a posteriori estimation, $\boldsymbol{P}, \boldsymbol{Q}, \boldsymbol{R}$ are the covariance matrices of the estimated values, the system noise and the measurement noise, respectively. It is assumed that $\boldsymbol{P}$, $\boldsymbol{R}$ and $\boldsymbol{Q}$ are symmetric, $\boldsymbol{\nu}_k$ and $\boldsymbol{\omega}_k$ are uncorrelated and with zero expected value and $E\{(\boldsymbol{x_j x_k})\} \approx 0$ for $j \neq k$. The proof of the FKF can be found in [9]. It is an optimal state vector estimator using the knowledge of the system model, input and output signals.

IV. Distribution and Decentralization

In the case of fractional order systems, there is the need for higher computation power and memory, because the estimation of the states and covariance matrices has to consider vectors of past values and is therefore computationally more intensive compared to a classical (integer order) Kalman filter.

Therefore, the reliability, efficiency, and cost of the computation for a large scale system can be improved if a decentralized and distributed approach is used. The components of a distributed system communicate and coordinate their actions by passing messages, in order to achieve a common goal.

Decentralization means that different measuring or sensing devices are grouped into some nodes or subsystems, so that each node performs a filtering based on locally available measurements. *Distribution* means, that a different subspace of the global state space is formed in each subsystem [8], [12]. Since different nodes have to exchange local estimates in a fusion step, the use of transformation matrices is required. This structure makes possible that - in general - not every node is necessarily connected with each of the other nodes. In [13] and [12] a Distributed and Decentralized Kalman filter (DDKF) for linear and nonlinear integer order systems is presented. In [8] it was shown that the DDKF is equivalent to the linear Kalman filter, if there is no decentralization of measurements and distribution of the model applied. The main advantages of the DDKF are the scalability of the filter network and the reduced order of the local systems compared to the global one [8].

A. Distribution of the System Equations

In complex systems, the order of the local subsystems is substantial smaller than the order of the global system which allows to perform their calculations comfortably. Assuming that not all equations of the global system are coupled. [8]

To contemplate a distributed and decentralized system, we have to project the global system into the different subsystems by a division of the vectors and the use of convenient transformations, respectively. First, we define different transformation matrices for the states, the input and the output as follows to indicate and differ the local values of the subsystem i from the global ones:

$$\begin{aligned} \boldsymbol{x}_{i,k} &= \boldsymbol{T}_i \boldsymbol{x}_k && (17) \\ \boldsymbol{u}_{i,k} &= \boldsymbol{S}_i \boldsymbol{u}_k && (18) \\ \boldsymbol{y}_{i,k} &= \boldsymbol{M}_i \boldsymbol{y}_k && (19) \end{aligned}$$

If the dynamic system has multiple inputs, a distribution of the input-vector is possible using the transformation matrix $\boldsymbol{S}_i$, so that each subsystem will have a local input signal or input-vector. Similarly, the state and output vectors can be partitioned using the matrices $\boldsymbol{T}_i$ and $\boldsymbol{M}_i$. Therefore the dimension of the local vectors form a subspace of the global state space.

Furthermore, all elements of the transformation matrices will be either 0 or 1, because they represent a selection of the elements of the global state-vector. The partition of the global model is often based on the different components of the system or different physical domains as described in [8]. The output vector is then divided as follows:

$$\boldsymbol{y}_k = \begin{pmatrix} \boldsymbol{y}_{1,k}^T & \boldsymbol{y}_{2,k}^T & \cdots & \boldsymbol{y}_{l,k}^T \end{pmatrix}^T . \qquad (20)$$

The output noise is divided analogously and it is assumed that the different components of the output noise vectors are uncorrelated

$$\boldsymbol{\nu}_k = \begin{pmatrix} \boldsymbol{\nu}_{1,k}^T & \boldsymbol{\nu}_{2,k}^T & \cdots & \boldsymbol{\nu}_{l,k}^T \end{pmatrix}^T . \qquad (21)$$

Then the local measurement equation can be written as follows [14]:

$$\boldsymbol{y}_{i,k} = \boldsymbol{M}_i \boldsymbol{C} \boldsymbol{x}_k + \boldsymbol{\nu}_{i,k} \qquad (22)$$

B. Selection of the Transformation Matrices

The selection of the local state-vector depends on the transformation matrix. If the global system model is known in advance, the transformation matrices will define each subsystem. In practical applications, two situations can appear. On the one hand, it is possible that the global system is identified and modelled first, and then local subsystems are derived from it. On the other hand, it is also possible that the different local subsystems are modelled independently, and then, based on their connection, the global system is obtained. The result in both cases will be a set of global system matrices and a group of transformation matrices $\boldsymbol{T}_i, \boldsymbol{S}_i, \boldsymbol{M}_i$ that will determine the local system matrices.

However, it is also possible that, due to the equations of the global model, all subsystems are connected with each other, so that the distribution might have no extent.

In order to avoid this, single states with a small influence on the local states can be neglected. This can be done systematically, if all entries of the system matrix, which are below a minimum value, are set to zero. This procedure can be implemented based on an analysis of the global state matrix [8].

As every submodule uses only a subspace of the system, it is necessary to add a fusion step in the Kalman filter where the estimates of different submodules are transferred to each other and merged. More details about the distribution of the system can be found in [13], [15] and [8].

V. Distributed and Decentralized Fractional Kalman Filter

A. Initialization

In the first step of the Distributed and Decentralized Fractional Kalman Filter (DDFKF), we make use of the transformation matrices from equations (17) to (19) to calculate the system matrices of the local subsystems:

$$\begin{aligned} \boldsymbol{A}_i &= \boldsymbol{T}_i \boldsymbol{A} \boldsymbol{T}_i^\diamond && (23) \\ \boldsymbol{B}_i &= \boldsymbol{T}_i \boldsymbol{B} \boldsymbol{S}_i^\diamond && (24) \\ \boldsymbol{C}_i &= \boldsymbol{M}_i \boldsymbol{C} \boldsymbol{T}_i^\diamond && (25) \\ \boldsymbol{\Gamma}_{i,j} &= \boldsymbol{T}_i \boldsymbol{\Gamma}_j \boldsymbol{T}_i^\diamond && (26) \\ \boldsymbol{Q}_i &= \boldsymbol{T}_i \boldsymbol{Q} \boldsymbol{T}_i^\diamond && (27) \\ \boldsymbol{R}_i &= \boldsymbol{M}_i \boldsymbol{R} \boldsymbol{M}_i^\diamond && (28) \end{aligned}$$

$\boldsymbol{Z}^\diamond$ means the Moore-Penrose-Pseudoinverse of a matrix $\boldsymbol{Z}$:

$$\boldsymbol{Z}^\diamond = \left(\boldsymbol{Z}^T \boldsymbol{Z} \right)^{-1} \boldsymbol{Z}^T \qquad (29)$$

which is used, because the transformation matrices are not square in general, so that it is not always possible to calculate the inverse. The local system matrices (eq. (23) to (28)) can be used to define the local system equations for each subsystem with index i. After all transformation matrices have been selected, the initialization of the DDFKF algorithm is done in the same form as in the DDKF:

$$\hat{\boldsymbol{x}}^+_{i,0} = \boldsymbol{0} \tag{30}$$

$$\boldsymbol{P}^+_{i,0} = (100\dots1000)\boldsymbol{I} \tag{31}$$

B. Prediction

With the previous transformation matrices, the prediction step is defined in a form similar to the FKF, by equations (32) to (34).

$$\Delta^\gamma \hat{\boldsymbol{x}}^-_{i,k} = \boldsymbol{T}_i \boldsymbol{A} \boldsymbol{T}_i^\diamond \hat{\boldsymbol{x}}^+_{i,k-1} + \boldsymbol{T}_i \boldsymbol{B} \boldsymbol{S}_i^\diamond \boldsymbol{u}_{i,k-1} \tag{32}$$

$$\hat{\boldsymbol{x}}^-_{i,k} = \Delta^\gamma \hat{\boldsymbol{x}}^-_{i,k} - \sum_{j=1}^{z} (-1)^j \boldsymbol{T}_i \boldsymbol{\Gamma}_j \boldsymbol{T}_i^\diamond \hat{\boldsymbol{x}}^+_{i,k-j} \tag{33}$$

$$\begin{aligned}\boldsymbol{P}^-_{i,k} &= \boldsymbol{T}_i(\boldsymbol{A}+\boldsymbol{\Gamma}_1)\boldsymbol{T}_i^\diamond \boldsymbol{P}^+_{i,k-1} \left(\boldsymbol{T}_i^\diamond\right)^T (\boldsymbol{A}+\boldsymbol{\Gamma}_1)^T \boldsymbol{T}_i^T + \\ &+ \boldsymbol{T}_i \boldsymbol{Q}_{k-1} \boldsymbol{T}_i^T + \sum_{j=2}^{z} \boldsymbol{T}_i \boldsymbol{\Gamma}_j \boldsymbol{T}_i^\diamond \boldsymbol{P}^+_{i,k-j} \left(\boldsymbol{T}_i^\diamond\right)^T \boldsymbol{\Gamma}_j^T \boldsymbol{T}_i^T\end{aligned} \tag{34}$$

$z = \min[k, B_L]$ stands for the upper limit of the sum, with B_L being the buffer or memory length. As it can be seen, all matrices defined in the global system have to be transformed into the local subsystems using the corresponding transformation matrices. Equation (33) provides the local a priori state vector estimate and equation (34) gives the corresponding local a priori covariance matrix.

C. Correction

In the next step, the estimates of the state-vector and the covariance matrix are corrected based on local measurements, as shown by equations (35) to (37). However, these are not the a posteriori estimation values. These will be obtained later using the information of all relevant measurements. This is why we use a tilde $(\tilde{*})$ in the equations for this values. For calculating the covariance matrix after correction in equation (35), an equivalent form is used, that is positive definite, if the corresponding a priori matrix is also positive definite. Equation (36) provides the Kalman gain, which scales the influence of the measurements in the correction step.

$$\tilde{\boldsymbol{P}}^+_{i,k} = \left((\boldsymbol{P}^-_{i,k})^{-1} + \boldsymbol{C}^T_{i,k} \boldsymbol{R}^{-1}_{i,k} \boldsymbol{C}_{i,k}\right)^{-1} \tag{35}$$

$$\boldsymbol{K}_{i,k} = \tilde{\boldsymbol{P}}^+_{i,k} \boldsymbol{C}^T_{i,k} \boldsymbol{R}^{-1}_{i,k} \tag{36}$$

$$\tilde{\boldsymbol{x}}^+_{i,k} = \hat{\boldsymbol{x}}^-_{i,k} + \boldsymbol{K}_{i,k}(\boldsymbol{y}_{i,k} - \boldsymbol{C}_{i,k} \hat{\boldsymbol{x}}^-_{i,k}) \tag{37}$$

D. Fusion

Finally, the local estimates calculated by each node are corrected in the fusion step, taking the estimates obtained by other nodes into account. This last step is identical to the fusion step of the DDKF [8], [13]–[15]:

$$\boldsymbol{E}_{ji,k} = \left(\boldsymbol{V}^T_{ji}\right)^\diamond \left(\left(\tilde{\boldsymbol{P}}^+_{j,k}\right)^{-1} - \left(\boldsymbol{P}^-_{j,k}\right)^{-1}\right) \boldsymbol{V}^\diamond_{ji} \tag{38}$$

$$\boldsymbol{e}_{ji,k} = \left(\boldsymbol{V}^T_{ji}\right)^\diamond \left(\left(\tilde{\boldsymbol{P}}^+_{j,k}\right)^{-1} \tilde{\boldsymbol{x}}^+_{j,k} - \left(\boldsymbol{P}^-_{j,k}\right)^{-1} \hat{\boldsymbol{x}}^-_{j,k}\right) \tag{39}$$

$$\left(\boldsymbol{P}^+_{i,k}\right)^{-1} = \left(\boldsymbol{P}^-_{i,k}\right)^{-1} + \sum_{j=1}^{l} \boldsymbol{E}_{ji,k} \tag{40}$$

$$\hat{\boldsymbol{x}}^+_{i,k} = \boldsymbol{P}^+_{i,k} \left(\left(\boldsymbol{P}^-_{i,k}\right)^{-1} \hat{\boldsymbol{x}}^-_{i,k} + \sum_{j=1}^{l} \boldsymbol{e}_{ji,k}\right) \tag{41}$$

l is the number of the local subsystems and $\boldsymbol{V}_{ji} = \boldsymbol{T}_i \boldsymbol{T}_j^\diamond$ is used to transform an estimation from subsystem j to subsystem i. Equation (38) indicates how much the information, that is, the inverse of the state estimation covariance matrix, in the local subsystem j has increased due to the local measurements and is then transformed into subsystem i. Equation (39) gives analogously the raise of the information of the estimated states in subsystem j which is transformed into subsystem i. The last two equations provide the a posteriori values after the fusion of the local estimates. In order to obtain these equations, we make all the assumptions explained in this article.

E. Proof

The proof of the presented algorithm is similar to the proofs of the FKF and DDKF. The main difference with respect to the FKF is that local matrices and vectors are used in each subsystem instead of using the global model and vectors. The main difference with respect to the DDKF is that some deductions are more complicated due to the fractional operator.

Based on the algorithm of the FKF, substituting equation (9) in (10), equation (42) is obtained:

$$\hat{\boldsymbol{x}}^-_{k} = \boldsymbol{A} \hat{\boldsymbol{x}}^+_{k-1} + \boldsymbol{B} \boldsymbol{u}_{k-1} - \sum_{j=1}^{k} (-1)^j \boldsymbol{\Gamma}_j \hat{\boldsymbol{x}}^+_{k-j} \tag{42}$$

On the other hand equation (23) yields to:

$$\boldsymbol{A} = \boldsymbol{T}_i^\diamond \boldsymbol{A}_i \boldsymbol{T}_i \tag{43}$$

Similarly, substituting $\boldsymbol{x}_k$ from (17) and $\boldsymbol{B}$ and $\boldsymbol{\Gamma}_j$ from (24) and (26) in equation (42), equation (44) is obtained:

$$\begin{aligned}\boldsymbol{T}_i^\diamond \hat{\boldsymbol{x}}^-_{i,k} &= \boldsymbol{T}_i^\diamond \boldsymbol{A}_i \boldsymbol{T}_i \boldsymbol{T}_i^\diamond \hat{\boldsymbol{x}}^+_{i,k-1} + \boldsymbol{T}_i^\diamond \boldsymbol{B}_i \boldsymbol{S}_i \boldsymbol{S}_i^\diamond \boldsymbol{u}_{i,k-1} \\ &- \sum_{j=1}^{k} (-1)^j \boldsymbol{T}_i^\diamond \boldsymbol{\Gamma}_{i,j} \boldsymbol{T}_i \boldsymbol{T}_i^\diamond \hat{\boldsymbol{x}}^+_{i,k-j}\end{aligned} \tag{44}$$

Then, premultiplying equation (44) by $\boldsymbol{T}_i$ and taking into account that $\boldsymbol{T}_i \boldsymbol{T}_i^\diamond = \boldsymbol{S}_i \boldsymbol{S}_i^\diamond = \boldsymbol{I}$, equation (45) can be achieved:

$$\hat{\boldsymbol{x}}^-_{i,k} = \boldsymbol{A}_i \hat{\boldsymbol{x}}^+_{i,k-1} + \boldsymbol{B}_i \boldsymbol{u}_{i,k-1} - \sum_{j=1}^{k} (-1)^j \boldsymbol{\Gamma}_{i,j} \hat{\boldsymbol{x}}^+_{i,k-j} \tag{45}$$

Finally, substituting equations (23) to (26) and replacing the upper limit of the sum by $z = \min[k, B_L]$, equations (32) and (33) of the DDFKF algorithm are obtained.

Equation (34) is proved analogously with the assumption that past state vectors are uncorrelated. This assumption can be relaxed by using the infinite dimensional form of the state-space representation, but this will lead to a higher computational cost.

$$\begin{aligned}
\boldsymbol{P}^-_{i,k} &= E[(\boldsymbol{x}_{i,k} - \hat{\boldsymbol{x}}^-_{i,k})(\boldsymbol{x}_{i,k} - \hat{\boldsymbol{x}}^-_{i,k})^T] \\
&= (\boldsymbol{A}_i + \boldsymbol{\Gamma}_{i,1}) \cdot \\
&\cdot E[(\boldsymbol{x}_{i,k-1} - \hat{\boldsymbol{x}}^+_{i,k-1})(\boldsymbol{x}_{i,k-1} - \hat{\boldsymbol{x}}^+_{i,k-1})^T] \cdot \\
&\cdot (\boldsymbol{A}_i + \boldsymbol{\Gamma}_{i,1})^T + E[\boldsymbol{\omega}_{i,k-1}\boldsymbol{\omega}^T_{i,k-1}] + \\
&+ \sum_{j=2}^{k} \boldsymbol{\Gamma}_{i,j} E[(\boldsymbol{x}_{i,k-j} - \hat{\boldsymbol{x}}^+_{i,k-j})(\boldsymbol{x}_{i,k-j} - \hat{\boldsymbol{x}}^+_{i,k-j})^T] \boldsymbol{\Gamma}^T_{i,j} \\
&= (\boldsymbol{A}_i + \boldsymbol{\Gamma}_{i,1}) \boldsymbol{P}^+_{i,k-1} (\boldsymbol{A}_i + \boldsymbol{\Gamma}_{i,1})^T + \boldsymbol{Q}_{i,k-1} + \\
&+ \sum_{j=2}^{k} \boldsymbol{\Gamma}_{i,j} \boldsymbol{P}^+_{i,k-j} \boldsymbol{\Gamma}^T_{i,j}.
\end{aligned}$$

Substituting the local matrices using equations (23) to (28), equation (34) is obtained.

The proof of equations (35) to (37) is exactly identical to the corresponding proof of the DDKF or the FKF, respectively. The proof can be found in [8], [13] or [9].

The proof of the fusion step is also similar to the proof of the fusion step of the DDKF which is shown in detail in [8], [13]. The main difference is how the transformation of the covariance matrices from a subsystem j into subsystem i is derived which is shown in the following paragraph. First, the a priori state covariance matrix of subsystem j transformed into the global system description g can be expressed as follows:

$$\begin{aligned}
\boldsymbol{P}^-_{jg,k} &= E\left\{(\boldsymbol{x}_{jg,k} - \hat{\boldsymbol{x}}^-_{jg,k})(\boldsymbol{x}_{jg,k} - \hat{\boldsymbol{x}}^-_{jg,k})^T\right\} \\
&= E\left\{\boldsymbol{T}^\diamond_j(\boldsymbol{x}_{j,k} - \hat{\boldsymbol{x}}^-_{j,k})(\boldsymbol{x}_{j,k} - \hat{\boldsymbol{x}}^-_{j,k})^T \left(\boldsymbol{T}^\diamond_j\right)^T\right\} \\
&= \boldsymbol{T}^\diamond_j \boldsymbol{P}^-_{j,k} \left(\boldsymbol{T}^\diamond_j\right)^T. \qquad (46)
\end{aligned}$$

Transforming the obtained result into another subsystem i, equation (47) can be established:

$$\begin{aligned}
\boldsymbol{P}^-_{ji,k} &= E\left\{\boldsymbol{T}_i(\boldsymbol{x}_{jg,k} - \hat{\boldsymbol{x}}^-_{jg,k})(\boldsymbol{x}_{jg,k} - \hat{\boldsymbol{x}}^-_{jg,k})^T \boldsymbol{T}^T_i\right\} \\
&= \boldsymbol{T}_i \boldsymbol{P}^-_{jg,k} \boldsymbol{T}^T_i = \boldsymbol{T}_i \boldsymbol{T}^\diamond_j \boldsymbol{P}^-_{j,k} \left(\boldsymbol{T}^\diamond_j\right)^T \boldsymbol{T}^T_i. \qquad (47)
\end{aligned}$$

which describes a transformation of the covariance matrix from subsystem j to subsystem i. Using equation (34), replacing i by j and excluding $\boldsymbol{T}_j$ on the left side and $\boldsymbol{T}^T_j$ on the right side, we obtain equation (48)

$$\begin{aligned}
\boldsymbol{P}^-_{j,k} &= \boldsymbol{T}_j(\boldsymbol{A} + \boldsymbol{\Gamma}_1)\boldsymbol{T}^\diamond_j \boldsymbol{P}^+_{j,k-1} \left(\boldsymbol{T}^\diamond_j\right)^T (\boldsymbol{A} + \boldsymbol{\Gamma}_1)^T \boldsymbol{T}^T_j + \\
&+ \boldsymbol{T}_j \boldsymbol{Q}_{k-1} \boldsymbol{T}^T_j + \sum_{r=2}^{q} \boldsymbol{T}_j \boldsymbol{\Gamma}_r \boldsymbol{T}^\diamond_j \boldsymbol{P}^+_{j,k-r} \left(\boldsymbol{T}^\diamond_j\right)^T \boldsymbol{\Gamma}^T_r \boldsymbol{T}^T_j \\
&= \boldsymbol{T}_j \boldsymbol{G}_{j,k} \boldsymbol{T}^T_j \qquad (48)
\end{aligned}$$

where the middle terms are summarized in $\boldsymbol{G}_{j,k}$. Replacing $\boldsymbol{P}^-_{j,k}$ in equation (47) by equation (48), the following equation holds:

$$\boldsymbol{P}^-_{ji,k} = \boldsymbol{T}_i \boldsymbol{T}^\diamond_j \boldsymbol{T}_j \boldsymbol{G}_{j,k} \boldsymbol{T}^T_j \left(\boldsymbol{T}^\diamond_j\right)^T \boldsymbol{T}^T_i \qquad (49)$$

Following the same structure as the DDKF, we reduce equation (49) as follows:

$$\boldsymbol{P}^-_{ji,k} \;=\; \boldsymbol{V}_{ji} \boldsymbol{T}_j \boldsymbol{G}_{j,k} \boldsymbol{T}^T_j \boldsymbol{V}^T_{ji} = \boldsymbol{V}_{ji} \boldsymbol{P}^-_{j,k} \boldsymbol{V}^T_{ji} \qquad (50)$$

The rest of the proof of the fusion step is identical to the proof of the DDKF with the usage of equations (46) to (50).

VI. Simulation Results

Example 1: In this section, we provide a comparison between the estimated states of the FKF, the DDFKF and the state values from the simulation of a fractional order system. We consider a linear stochastic discrete fractional order state space system defined by the following matrices and vectors:

$$\begin{aligned}
\boldsymbol{A} &= \begin{bmatrix} -1.2 & 0 & 0 \\ 0 & -1.4 & -0.6 \\ 0 & 0.4 & -0.3 \end{bmatrix}, \quad \boldsymbol{B} = \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix} \\
\boldsymbol{C} &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}, \quad \gamma = \begin{bmatrix} 0.7 \\ 1.2 \\ 0.8 \end{bmatrix}, \quad \boldsymbol{x}_0 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \\
\boldsymbol{R} &= E[\boldsymbol{\nu}_k \boldsymbol{\nu}^T_k] = \begin{bmatrix} 0.01 & 0 \\ 0 & 0.01 \end{bmatrix} \\
\boldsymbol{Q} &= E[\boldsymbol{\omega}_k \boldsymbol{\omega}^T_k] = \begin{bmatrix} 0.003 & 0 & 0 \\ 0 & 0.003 & 0 \\ 0 & 0 & 0.003 \end{bmatrix} \\
\boldsymbol{P}_0 &= \begin{bmatrix} 100 & 0 & 0 \\ 0 & 100 & 0 \\ 0 & 0 & 100 \end{bmatrix}
\end{aligned}$$

with the sampling period $T = 1$ s, the simulation time is 300 s and the buffer length is 300. This system can be divided into two independent subsystems. The dimensions of the state matrix in the first node will be 1x1 and in the second subsystem 2x2. The transformation matrices are chosen as follows:

$$\begin{aligned}
\boldsymbol{T}_1 &= \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}, \quad \boldsymbol{T}_2 = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \\
\boldsymbol{S}_1 &= \boldsymbol{S}_2 = 1 \\
\boldsymbol{M}_1 &= \begin{bmatrix} 1 & 0 \end{bmatrix}, \quad \boldsymbol{M}_2 = \begin{bmatrix} 0 & 1 \end{bmatrix}
\end{aligned}$$

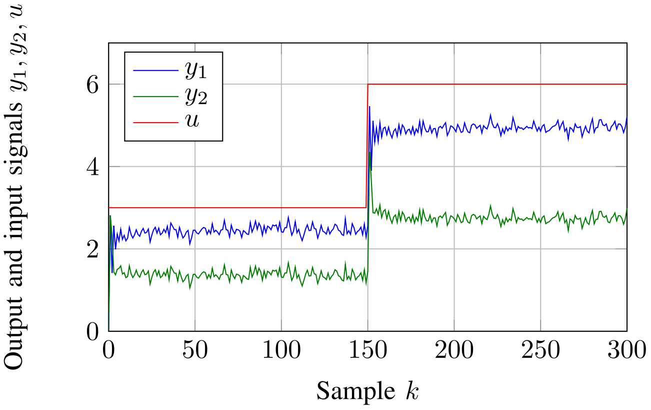

Fig. 1. Signal sequence of the input and outputs

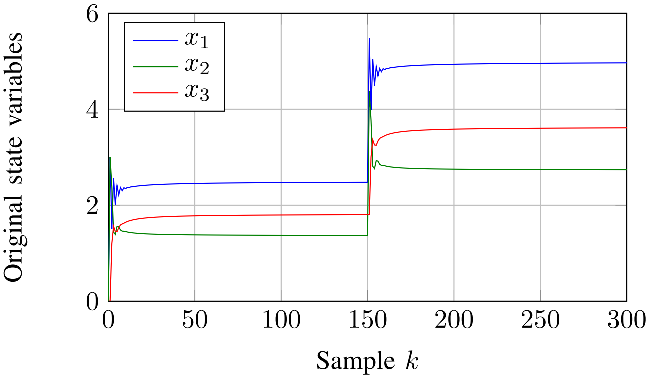

Fig. 2. Original state variables given from the simulation

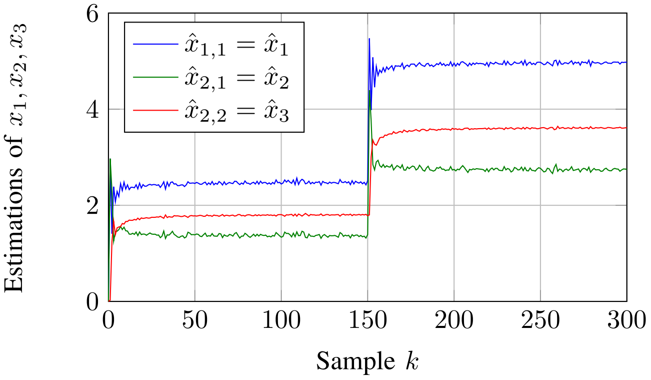

Fig. 3. Estimation of the states. $\hat{x}_1$ is given from the first subsystem, $\hat{x}_2$ and $\hat{x}_3$ are given from the second subsystem

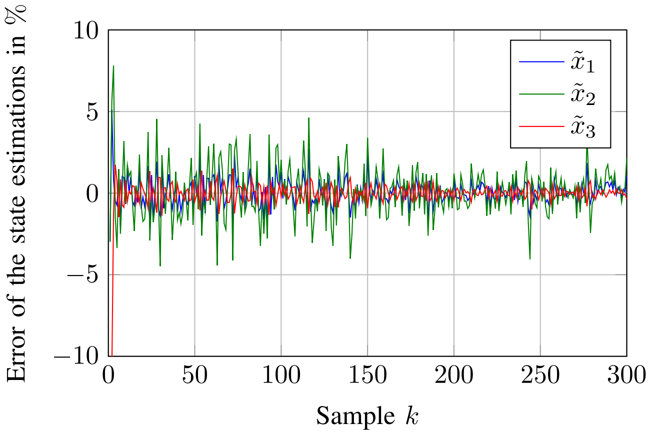

Fig. 4. Estimation errors of the state variables in percent for example 1 using the central FKF

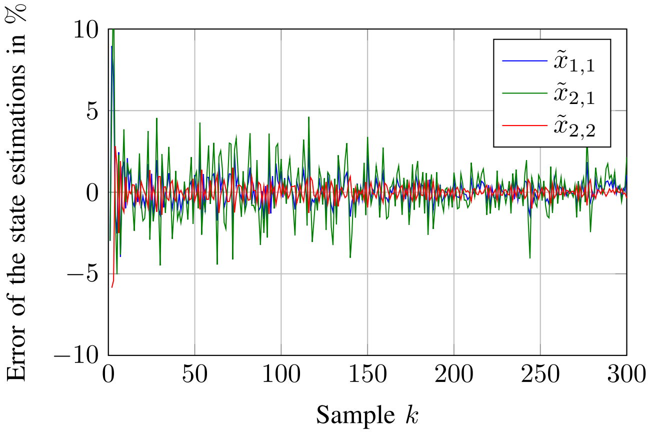

Fig. 5. Estimation errors of the state variables in percent for example 1 using the DDFKF

The input and output signals are shown in Fig. 1 and the original state variables from the simulation are given in Fig. 2. The signal to noise ratios of the output signals are calculated to $\text{SNR}_1 = 31$ dB and $\text{SNR}_2 = 26$ dB for y_1 and y_2, respectively. Comparing the results of the DDFKF in Fig. 3 with the original variables in Fig. 2, it follows that the estimation algorithm works accurately. In Fig. 4 the estimation error in percent is given for the central FKF. Comparing the results of the FKF with the DDFKF given in Fig. 5 it can be seen that both algorithms work adequate. In both cases the error is unbiased as it tends to zero for $t \to \infty$.

Example 2: In this section we consider a slightly modified system compared to example 1. As the system in example 1 can easily be divided into separate submodules, we are now dealing with a system where the submodules are coupled. Therefore, the system matrix A is modified as follows:

$$\boldsymbol{A} = \begin{bmatrix} -1.2 & 0.05 & 0 \\ 0 & -1.4 & -0.6 \\ 0 & 0.4 & -0.3 \end{bmatrix}$$

The rest of the system of example 1 remains the same. We see that now the state x_2 influences subsystem 1. Therefore, it has to be considered in the local system equations. Since x_2 also depends on x_3 and we want to achieve an order reduction in the submodules, we assume x_2 as time invariant in subsystem 1. Additionally, since x_2 has only a small influence

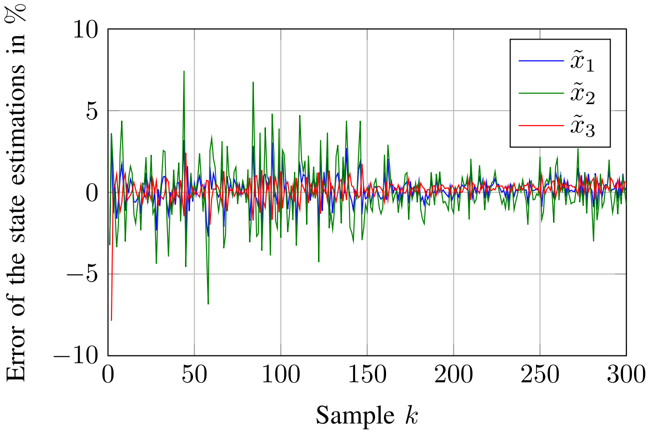

Fig. 6. Estimation errors of the state variables in percent for example 2 using the central FKF

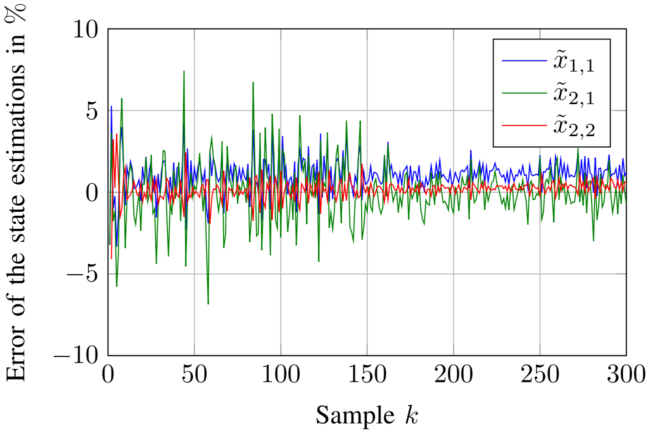

Fig. 7. Estimation errors of the state variables in percent for example 2 using the DDFKF

on x_1 and also the estimation of x_2 from subsystem 2 is transferred to subsystem 1, we expect still a good estimation of all the states. Summarizing this procedure, the following system matrices are chosen in the subsystems:

$$\boldsymbol{A}_1 = \begin{bmatrix} -1.2 & 0.05 \\ 0 & 1 \end{bmatrix}, \quad \boldsymbol{A}_2 = \begin{bmatrix} -1.4 & -0.6 \\ 0.4 & -0.3 \end{bmatrix}$$

Along with this adaption, we additionally reduce the buffer length to $B_L = 100$. In Fig. 6 the estimation error in percent of the FKF for example 2 is given. Since all information is available in the central algorithm, it shows similar results compared to example 1 in Fig. 4. It shows only a small impairment, because the buffer length is reduced. In Fig. 7 the estimation error of the DDFKF is given. It shows that even if the buffer length is reduced and the model is simplified, the DDFKF algorithm leads to accurate estimations. However, a bias in the estimation is observable, but from a practical view it can be neglected as long it is in the range of about 1%.

VII. Conclusion

This article presents a distributed and decentralized Kalman filter algorithm for a fractional order system. This algorithm combines the advantages of the fractional Kalman filter with the distribution and decentralization scheme. Regarding the distribution and decentralization, it was shown how the system is divided into submodules using transformation matrices for the input vector, output vector and the state vector. These matrices split the respective vectors into different components, which correspond to each subsystem. The distribution and decentralization of the system results in small and scalable nodes with reduced computational effort and less required memory. Especially in the case of fractional order systems, where the complexity is higher compared to an integer order system this leads to a large benefit. It is also shown that the DDFKF features as high accuracy for the estimated states as the FKF even though not all measurements are available in the respective nodes. Another advantage is that only relevant information is transmitted between the nodes, which reduces the communication effort and leads to a higher flexibility and scalability.

References

[1] A. Dzielinski and D. Sierociuk, “Ultracapacitor modelling and control using discrete fractional order state-space model,” *Acta Montanistica, Slovaca*, vol. 13, no. 1, pp. 136 – 145, 2008.

[2] X. Li, C. Essex, and M. Davidson, “A local fractional derivative,” in *ASME Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, Chicago, Illinois, USA, September 2003.

[3] J. Sabatier, M. Merveillaut, J. M. Francisco, F. Guillemard, and D. Porcelatto, “Fractional models for lithium-ion batteries.” in *European Control Conference (ECC)*, Zurich, Switzerland, 2013.

[4] J. Illig, T. Chrobak, M. Ender, J. P. Schmidt, D. Klotz, and E. Ivers-Tiffee, “Studies on lifepo4 as cathode material in li-ion batteries,” *ECS Transactions*, vol. 28, no. 30, pp. 3–17, october 2010.

[5] I. Podlubny, *Fractional differential equations*, ser. Mathematics in Science and Engineering. Academic Press, 1999.

[6] C. A. Monje, Y. Chen, B. M. Vinagre, D. Xue, and V. Feliu, *Fractional-order Systems and Controls*. Springer, 2010.

[7] M. Eckert, M. Kupper, and S. Hohmann, “Functional fractional calculus for system identification of battery cells.” *Automatisierungstechnik*, vol. 62, no. 62, pp. 272–281, March 2014.

[8] P. Hilgers, “Verteilte zustandsschätzung nichtlinearer systeme.” Ph.D. dissertation, University of Ilmenau, Universitatsverlag Ilmenau, 2012.

[9] D. Sierociuk and A. Dzielinski, “Fractional kalman filter algorithm for the states, parameters and order of fractional system estimation.” *International Journal of Applied Mathematics and Computer Science*, vol. 16, no. 1, pp. 129 – 140, 2006.

[10] C. F. Lorenzo and T. T. Hartley, *Initialized fractional calculus*. National Aeronautics and Space Administration, Glenn Research Center, 2000.

[11] J. Sabatier, C. Farges, and J.-C. Trigeassou, “Fractional systems state space description: some wrong ideas and proposed solutions,” *Journal of Vibration and Control*, vol. 20, no. 7, pp. 1076–1084, 2014.

[12] P. Hilgers and C. Ament, “Distributed and decentralized estimation of non-linear systems.” in *Proceedings of the IEEE International Conference on Control Applications*, Yokohama, Japan, 2010, pp. 328 – 333.

[13] A. G. O. Mutambara and H. F. Durrant-Whyte, “A formally verified modular decentralized robot control system.” in *Proceedings of the 1993 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Yokohama, Japan, July 1993, pp. 2023 – 2030.

[14] ——, “Modular scalable robot control,” in *Multisensor Fusion and Integration for Intelligent Systems, 1994. IEEE International Conference on MFI ’94*, Oct 1994, pp. 121–127.

[15] ——, “Distributed decentralised robot control.” in *Proceedings of the American Control Conference*, Baltimore, Maryland, June 1994, pp. 2266 – 2267.