ELEC 5840:
Distributed Kalman Filtering

Jason Kootsher
MSEE University of Colorado Denver
Independent Study, Dr. M. Radenkovic

I. Introduction

The study and application of Kalman filtering is a well understood problem in control theory, with many applications in aerospace, aeronautics, ballistics, and general tracking problems. The purpose of this report and analysis is to study the types of problems where an array of sensors is responsible for tracking the state of a body, where each sensor contributes a small amount of information on the body. Only the full sensor array is capable of giving the state estimate, as opposed to an individual sensor or subset of sensors.

II. Distributed Kalman Filtering and Sensor Fusion in Sensor Networks

A. Report

The summary of this paper covers the topic of distributed Kalman filtering using sensor information. State data is estimated two ways; the centralized Kalman filter (CKF) and micro Kalman filter (DKF). In order to assess these two forms however, a third definition is required for the Kalman filter so as to frame the sensor information; the information Kalman filter (IKF).

Beginning with the IKF, a state and sensor model is defined as such:

$$\begin{aligned} x_{k+1} &= A_k x_k + B_k w_k \\ z_k &= H_k x_k + v_k \end{aligned}$$

Equations (A_1 & A_2)

The quantity **z** is a vector quantity occupying a real, p-dimensional subspace and is one of n such **z**'s. The number n represents the total number of such p-dimensional sensors. Additionally, the quantities **w** and **v** are white Gaussian noise (WGN) with the following variances:

$$\begin{aligned} \sigma_w^2 &= Q_k \\ \sigma_v^2 &= R_k \end{aligned}$$

Equations (A_3 & A_4)

The matrices Q and R here are clearly diagonal, due to them representing variances for each of the n **w**'s and **v**'s respectively, but for generalized cases, these matrices need not be diagonal and would be representative of covariances. An additional assumption is made for the equation $\mathbf{A_1}$, where the initial condition $\mathbf{x_0}$ is picked from a normal distribution. With this initial condition, it is possible to generate the sets for both **x** and **z**.

From the sensor model given in equation $\mathbf{A_2}$ it is now possible to generate a state estimate with Markov properties:

$$\begin{aligned} \hat{x}_k &= E(x_k|Z_k) \\ \bar{x}_k &= E(x_k|Z_{k-1}) \\ Z(k) &= \{z_0, z_1, \ldots, z_k\} \end{aligned}$$

Equations (A_5 & A_6 & A_7)

With all generating quantities now defined, the CKF can be introduced given the following algorithm:

$$\begin{aligned} M_k^{-1} &= P_k^{-1} + H_k^T R_k^{-1} H_k \\ K_k &= M_k H_k^T R_k^{-1} \\ \hat{x}_k &= \bar{x}_k + K_k(z_k - H_k \bar{x}_k) \\ P_{k+1} &= A_k M_k A_k^T + B_k Q_k B_k^T \\ \bar{x}_{k+1} &= A_k \hat{x}_k \end{aligned}$$

Algorithm (1)

The quantities P and M are state covariance matrices and their inverses are state information matrices. The matrix quantity K is the Markov process encompassing all statistical state information up to time k and acts as a projection operator. The quantity is projects on is the time varying state **z**, wherein the resulting projection gives the solution space for the estimate of **x**.

With a complete description of the CKF, it is now possible to define the DKF. To summarize up to this point, the CKF description above gives an estimate of state using a single Kalman filter and relating the statistics of each node to create a centralized sink node wherein the tracking is to be calculated. The approach of the DKF will rely on defining a Kalman filter at each node and summing the distributed outcomes in order to obtain an estimate of state.

The DKF sensing model for a given node i is as follows:

$$z_i(k) = H_i x(k) + v_i(k)$$

Equation (A_8)

The notation of this sensing model differs from that in $\mathbf{A_2}$ because the goal of the CKF, to reiterate once again,was to analyze a sink node and this sensing model is specific to each individual node. The relationship to the quantities in the above sensing model can be related to the CKF sensing model with the following:

$$z_c = col(z_1, z_2, ..., z_n)$$
$$v_c = col(v_1, v_2, ..., v_n)$$
$$H_c = [H_1; H_2; ...; H_n]$$

Equations (A_9 & A_{10} & A_{11})

Taking the relationship further and applying it to the covariance of $\mathbf{v_c}$, defined by equation $\mathbf{A_4}$, it can be shown:

$$R_c = diag(R_1, R_2, ..., R_n)$$
$$M = (P + H_c^T R_c^{-1} H_c)^{-1}$$
$$K_c = M H_c^T R_c^{-1}$$

Equations (A_{12} & A_{13} & A_{14})

This set of equations shows the individual covariance of each node i and how the projection operation changes through the refined covariance expression M. The individual nodes can be further explored with the following relationships:

$$\hat{x} = \bar{x} + M(H_c^T R_c^{-1} z_c - H_c^T R_c^{-1} H_c \bar{x})$$
$$S = \frac{1}{n} H_c^T R_c^{-1} H_c = \frac{1}{n} \sum_{i=1}^{n} H_i^T R_i^{-1} H_i$$
$$y = \frac{1}{n} \sum_{i=1}^{n} H_i^T R_i^{-1} z_i$$

Equations (A_{15} & A_{16} & A_{17})

with $\mathbf{A_{15}}$ being the state propagation, S being the average information matrix, and **y** being the average sensor measurement.

The outstanding issue with each of these node measurements and the relationship of the DKF to the CKF is in the gain of the DKF. Since each Kalman filter is particular to each node, it stands to reason that for n nodes of the same type of filter, the gain of each micro Kalman filter must be multiplied by n so as to achieve the gain produced by the CKF. Taking this into account and redefining Algorithm (1) in terms of micro Kalman filter expressions, the following algorithm is obtained:

$$M_\mu(k) = (P_\mu^{-1}(k) + S)^{-1}$$
$$\hat{x}(k) = \bar{x}(k) + M_\mu(y - S\bar{x}(k))$$
$$P_\mu(k+1) = A(k)M_\mu(k)A^T(k) + B(k)Q_\mu(k)B^T(k)$$
$$\bar{x}(k+1) = A(k)\hat{x}(k)$$

Algorithm (2)

The largest advantage to Algorithm (2), and one of the major reasons is it advantageous to use in distributed systems, is the number of flops required for calculation. Algorithm (1) is on the order of O(m^2n), whereas Algorithm (2) is on the order of O(m^2). Clearly, for larger data sets it is computationally advantageous to use the latter of these algorithms.

B. Simulation & Analysis

For both the CKF and DKF simulations, the following state equation was used to generate "true" state for each time k:

$$x(k+1) = A_0(k)x(k) + B_0(k)w(k)$$
$$A_0(k) = A_0 = I + \epsilon A_0 + \frac{1}{2}(\epsilon A_0)^2 + \frac{1}{6}(\epsilon A_0)^3$$
$$B_0(k) = B_0 = \epsilon B_0$$

"True" State Generator

The first simulation utilized the computationally expensive procedure detailed by Algorithm (1). The following assumptions and initial conditions were used:

$$n = 200, t = 12s, \Delta t = .02s, I = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$
$$Q = 25I, R = \frac{1}{n}I, H = P = I$$
$$A_0 = \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}, B_0 = I, x_0 = \begin{pmatrix} 15 & -10 \end{pmatrix}^T$$

CKF Initial Conditions

Upon implementing Algorithm (1) as defined in the previous section (and whose code can be found in Appendix A), the following plots were obtained:

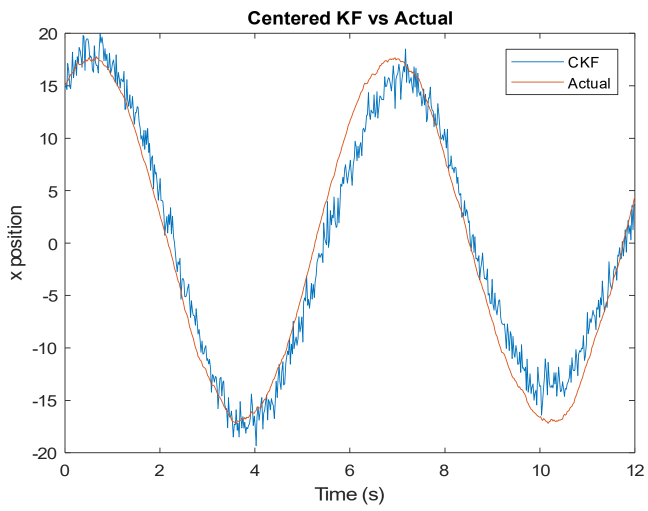

Figure 1

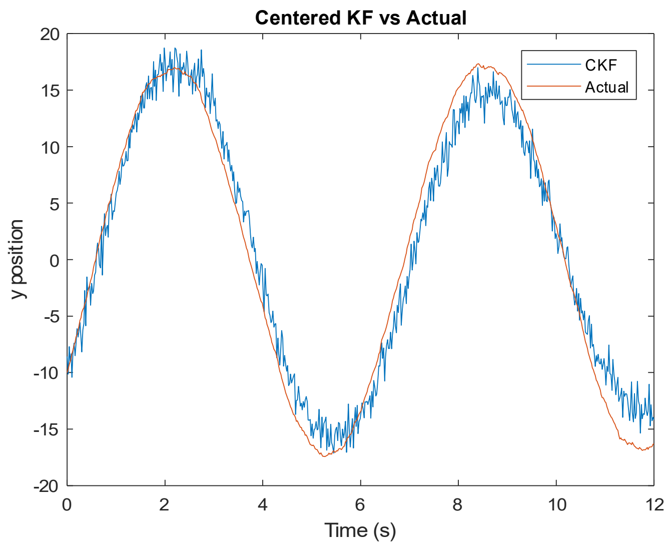

Figure 2

It can easily be seen that the CKF when applied to the sensor data is a near perfect trace of the “true” data. Further analysis of the unfiltered error gives the following results:

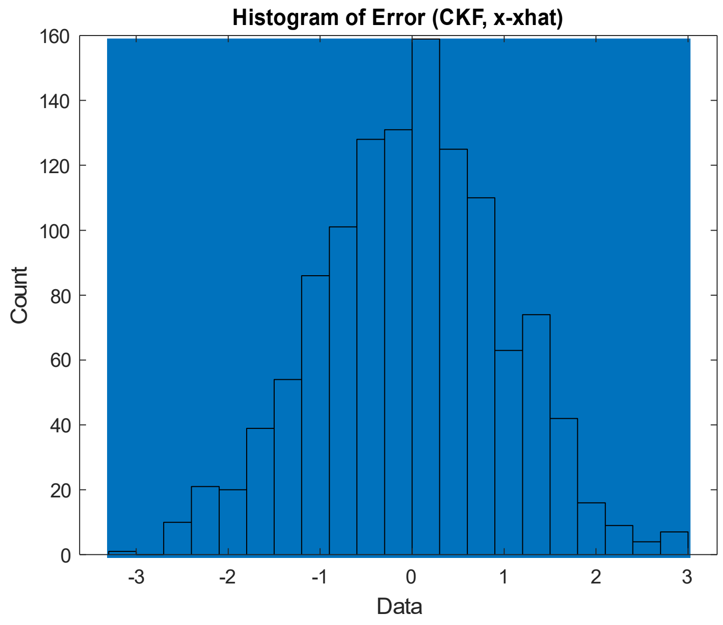

Figure 3

From here, it is obvious that the unfiltered error follows a normal distribution, which will only happen when the filtered error is (or very near) zero. This is due to the fact that our sensor model is related to our state model through a linear transformation, wherein the only difference between the two models is WGN. Since WGN follows a normal distribution, it is expected that as our measured state converges toward the actual state, the only remaining terms would be the WGN contribution. Therefore, the unfiltered error, being comprised of only these terms, should also follow a normal distribution.

When plotting this unfiltered error against a normal distribution on a probability plot, the above is reaffirmed:

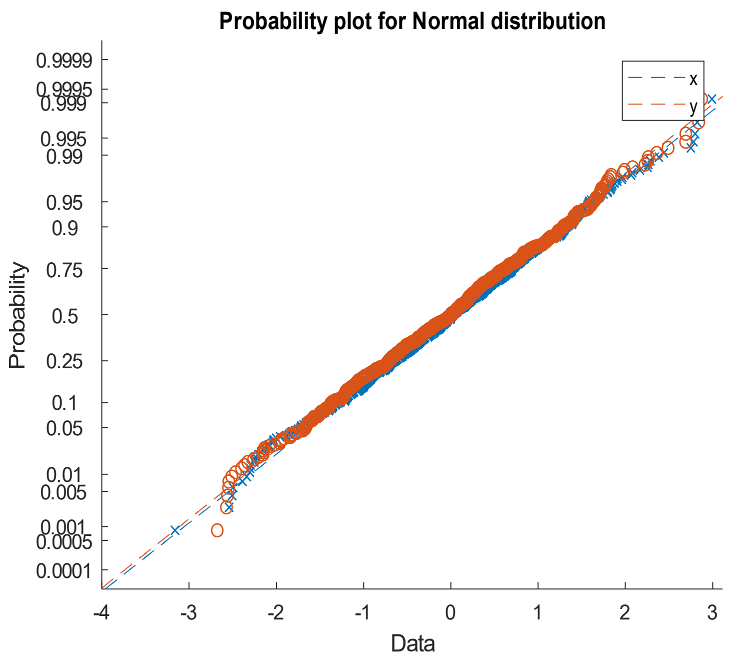

Figure 4: CKF Unfiltered Error

Moving along to the second simulation, Algorithm (2) is utilized for its computationally efficient (relative to Algorithm 1) procedure. The following assumptions and initial conditions were used:

$$n = 200, t = 12s, \Delta t = .02s, I = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$
$$Q_i = Q = 25I, R_i = 100(i)^{\frac{1}{2}} I, H = P = I$$
$$A_0 = \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}, B_0 = I, x_0 = \begin{pmatrix} 15 & -10 \end{pmatrix}^T$$

DKF Initial Conditions

The implementation of Algorithm (2) per its definition (code can also be found in Appendix A) gave way to the following results:

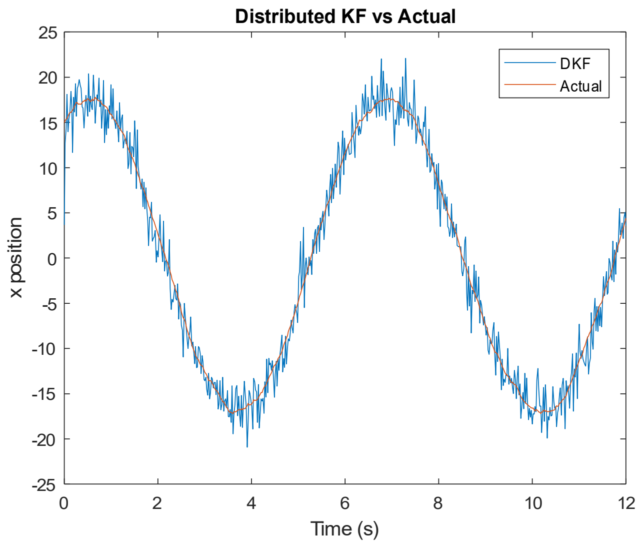

Figure 5

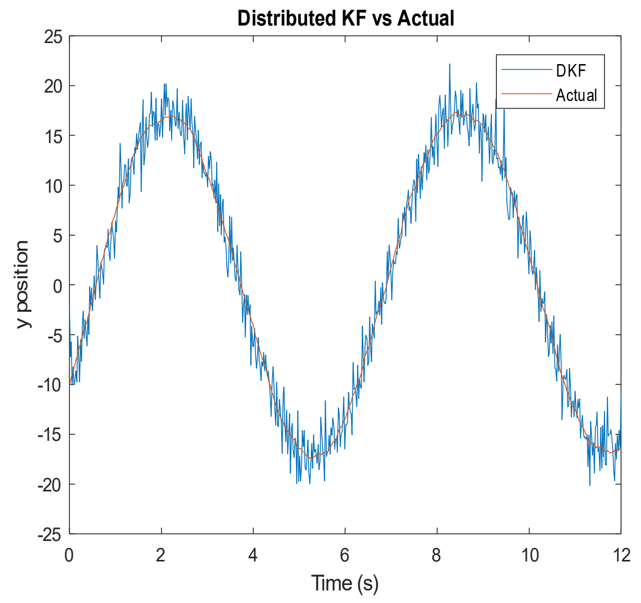

Figure 6

Using the same error analysis as was done for Algorithm (1):

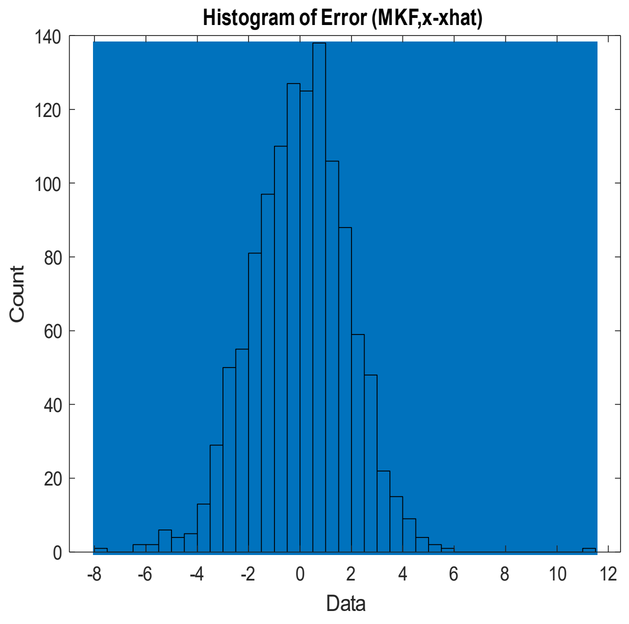

Figure 7

With the same logic applied to the DKF unfiltered error:

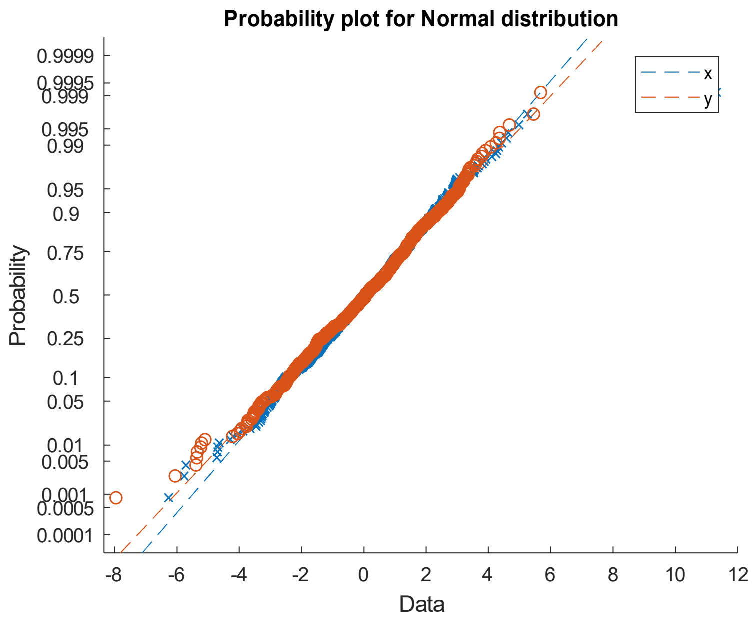

Figure 8: DKF Unfiltered Error

Additional analysis was performed in order to compare the two computationally different approaches and to verify that the implementation of micro Kalman Filters does indeed result in the same (similar due to WGN) curve tracing. These comparisons are as follows:

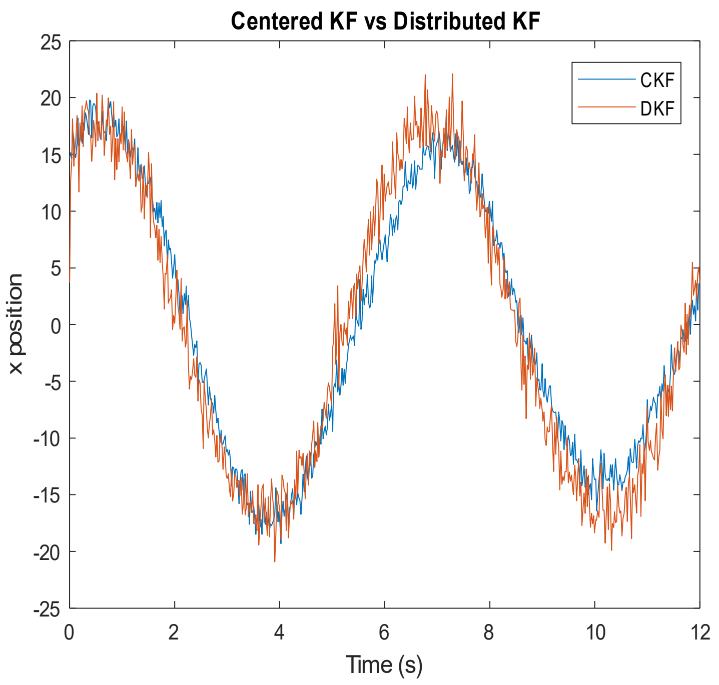

Figure 9

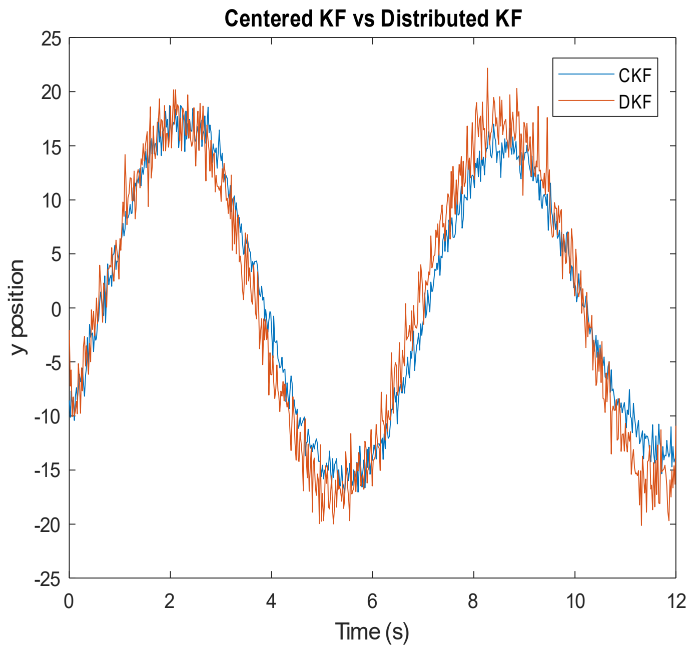

Figure 10

C. Conclusion

In summary, a mathematical approach for a distributed sensor array and its ability to perform the function of a traditional Kalman filter has been shown and verified through two separate formulations. The CKF and DKF (micro Kalman approach) both track state accurately, and as expected. The major benefit of this distributed approach is in its ability to discern and predict state information given only small amounts of information about the problem from each sensor, as opposed to requiring total state knowledge at every input. The additional benefit upon comparison of these two distributed approaches comes in the lower computational cost of further implementing n Kalman filters and analyzing state as a filter consensus, as opposed to creating a single Kalman filter and applying it to a state consensus.

Appendix A: Code

Github: https://github.com/jkootsher/Kalman

Appendix B: Bibliography

1 R. Olfati-Saber. Distributed Kalman Filter with Embedded Consensus Filters. In CDC-ECC'05, pages 8179--8184, 2005.

2 R. Olfati-Saber, "Distributed Kalman filtering for sensor networks", *Proc. IEEE Conf. Decision Control*, pp. 5492-5498, 2007-Dec.