CS4035 - Cyber Data Analytics | Assignment 5: Information Fusion

By : Hendra Hadhil Choiri (4468457)
Prahesa Kusuma Setia (4501543)

1. Distributed MLE

a. Suppose there are local estimates $xhat_1$ and $xhat_2$ with respective covariance matrices P_1 and P_2. By using the Bayes Rule, the optimal fused estimate is shown in below formula:

$$P^{-1} = P_1^{-1} + P_2^{-1}$$

$$xhat = P(P_1^{-1} * xhat_1 + P_2^{-1} * xhat_2)$$

b. Functions MLE and fuse_estimates have been implemented and tested by using single_sample.mat. The local, fused, and correct x positions and the P values are shown in Figure 1 of Appendix.We can observe that after those 2 locals are fused, the resulted xhat is getting closer to the correct x.

c. We have error covariance matrix of local estimates P_1 and P_2 (from first and second sensors in single_sample.mat) by fusing them together into fused estimate we have error covariance matrix P. After that, we calculate whether the fused estimate has higher accuracy than the local estimate or not by calculating the log det of their P inverses. We can see that the fused P^{-1} has bigger metric (log det(P_1^{-1})=-0.7645; log det(P_2^{-1})=1.6087; log det(P^{-1})=2.7617), therefore the fused estimate has higher accuracy than the local ones.

d. By computing covariance matrix P of fused estimates from sensor 1 until 6 and comparing it with P obtained from centralized estimator using log det of those matrices, we found out that they have the same value (10.5171). We can also see from Figure 2 in Appendix that we have improved the accuracy (denoted by increase in the log det metric after each fusion). Thus, the size difference for each fusion is always positive (as shown in Figure 2b).

e. The estimation error plots of multiple_sample.mat are shown in Figure 3 of Appendix. The change of accuracy is still equivalent to point 1d (i.e. the same plot as in Figure 2, the accuracy is improved after each fusion).

2. Distributed Kalman Filter

a. Function to compute Kalman filter has been implemented in kalman_filter.m. We followed definition provided in the assignment notes when we implemented it.

b. We tested our kalman_filter implementation using test_kalman.m file provided. The resulting plot can be seen in Figure 4 (a) of Appendix with red color. We can observe that the centralized estimate is more accurate, mostly closer to 0 compared to local estimates.

c. Using the function fuse_estimates as already implemented in previous section, we can fuse Kalman filter result of each sensor using Naive Bayesian approach. We can see in Figure 4 that by adding more sensor in the fusion, we improve the accuracy of our estimate (from wide amplitude range of black graph into yellow narrower amplitude). However, the Optimal bayesian approach still has better performance after all of the sensors are fused.

d. To check the improvement, the covariance of each time series error per fusion is computed. The plot is shown in Figure 5 in Appendix. Initially the error is very high because it is initialized with zeros. Then, after fused with next sensors, the error is decreasing. However, after the fusion with fifth sensor, the error is a little higher. So, the fusion does not always guarantee the improvement of error in time series.

Appendix

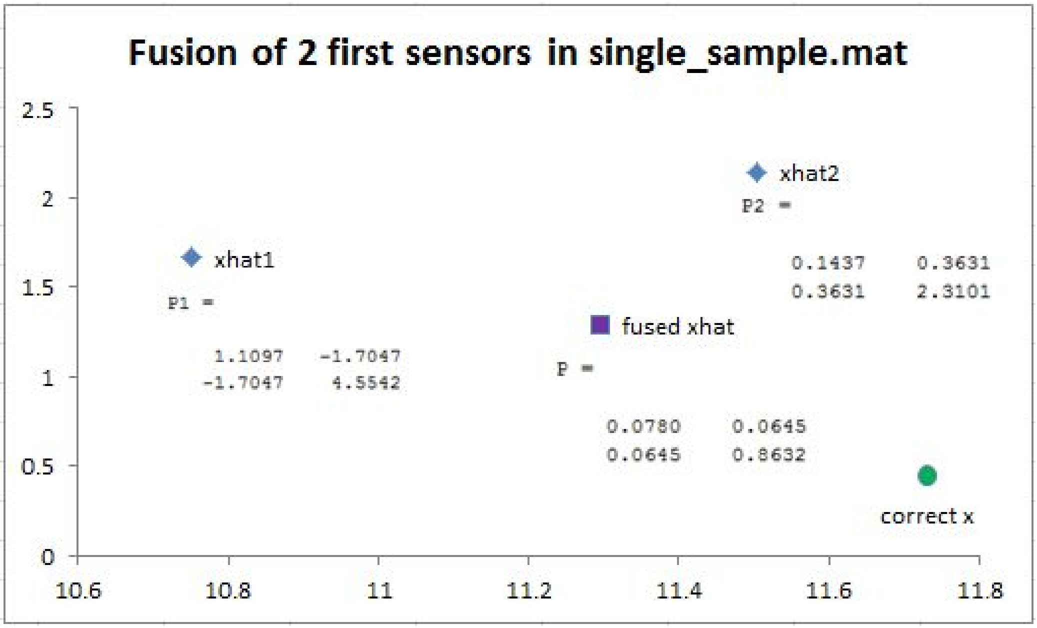
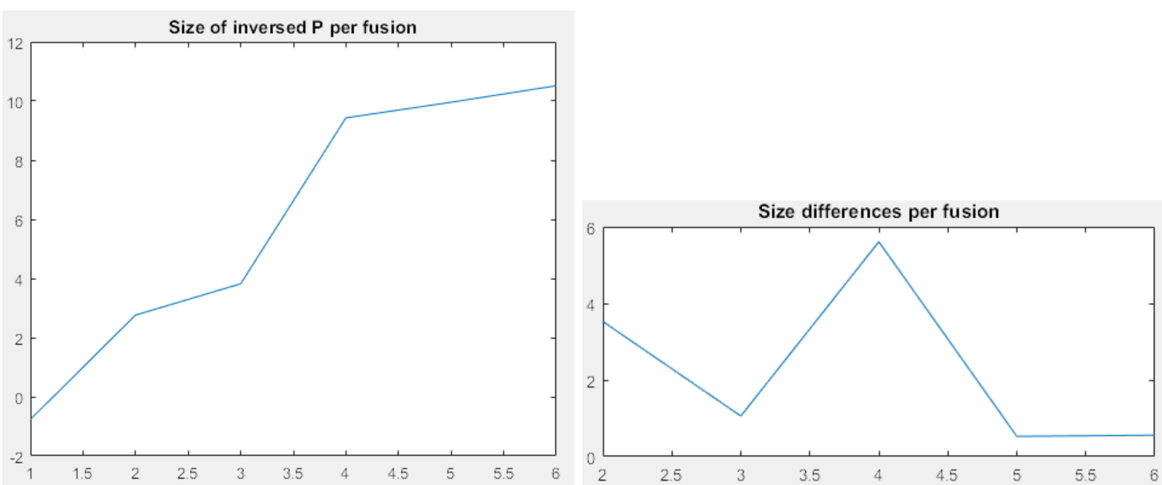
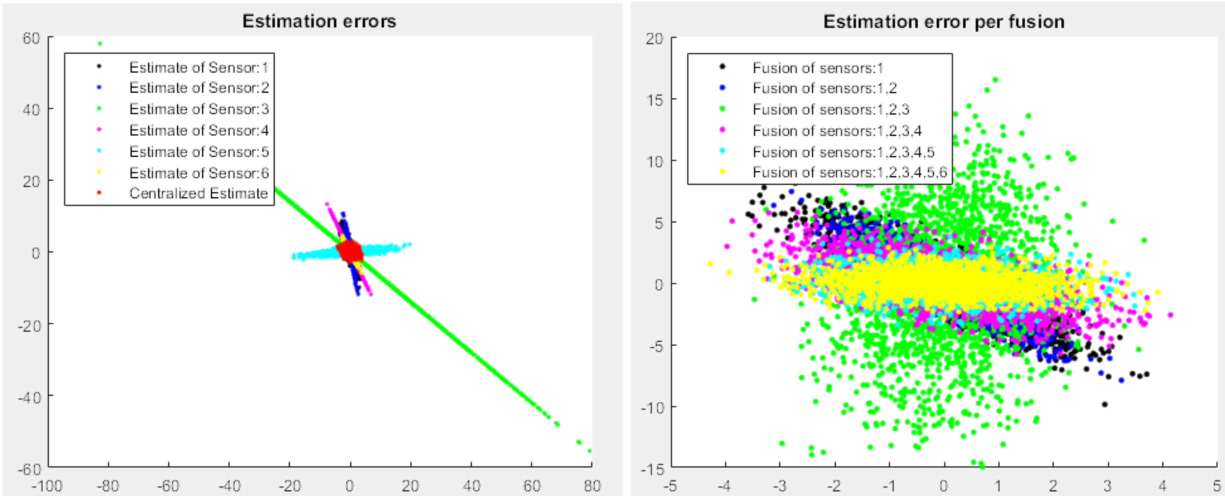


Figure 1. The local estimates from sensor 1 and 2 of single_sample.mat, and their fusion. The fused xhat becomes closer to correct x.



(a) (b)

Figure 2. Inversed P size estimated by log(det(P^{-1})): (a) the sizes after each fusion (b) the differences



(a) (b)

Figure 3. MLE estimation error of multiple_sample dataset, (a) per sensor (b) after each fusion

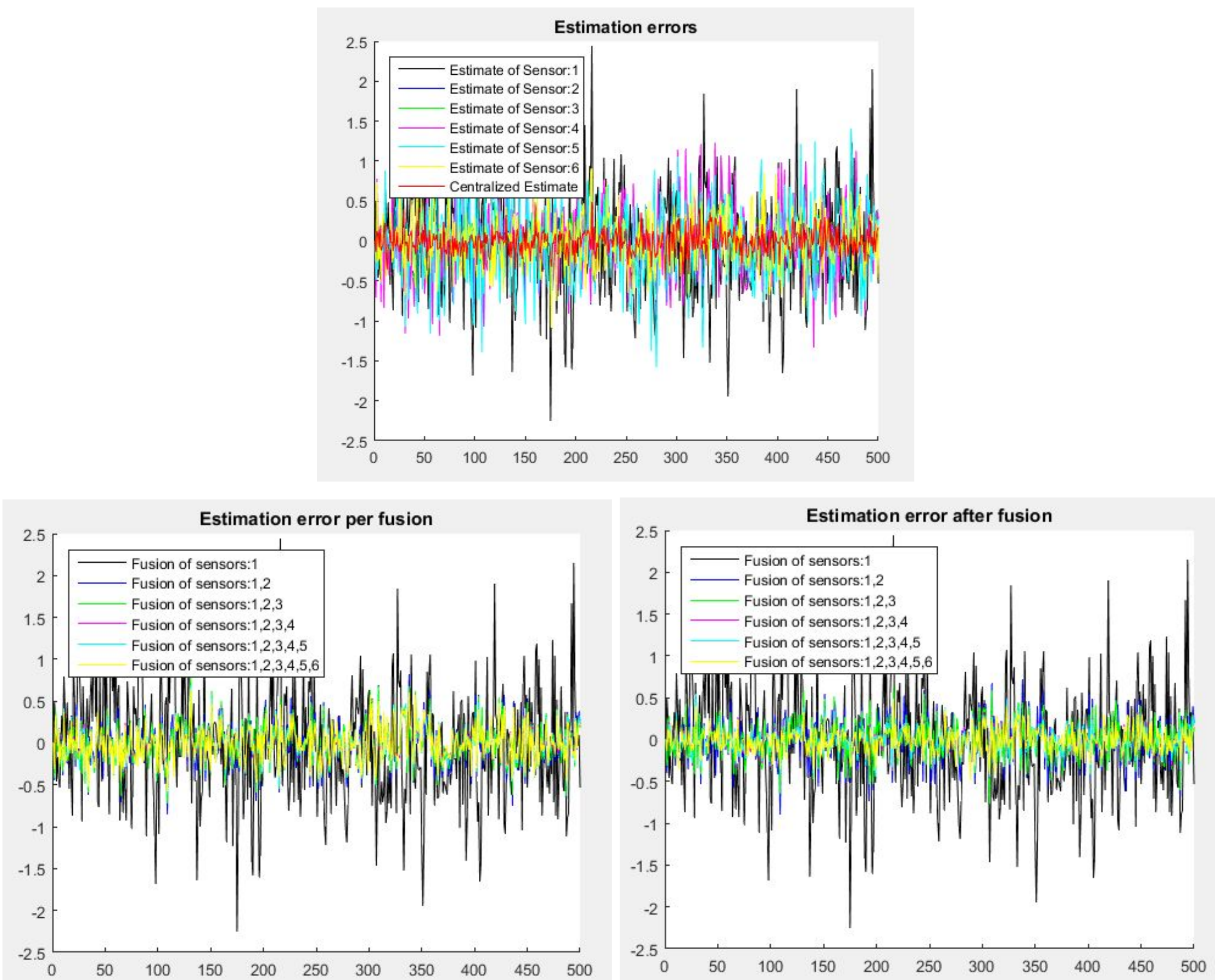


Figure 4. Kalman filter plot of (a) estimation errors, (b) estimation error per fusion, (c) estimation error after fusion (by using Optimal bayesian approach).

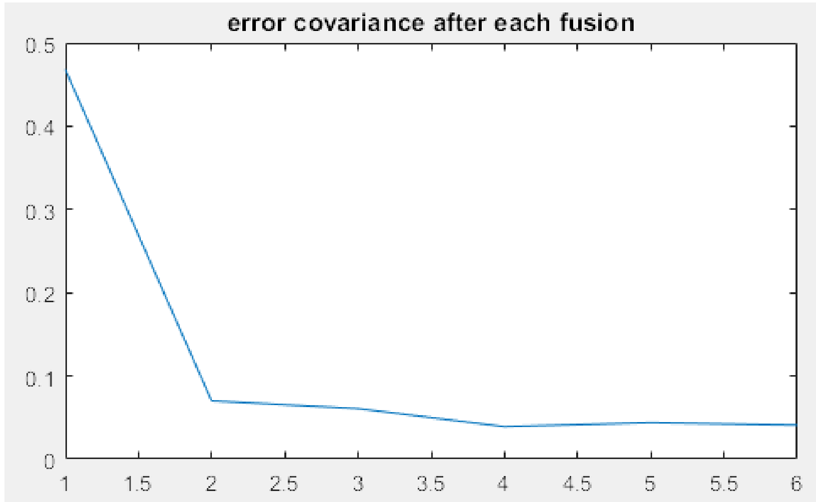


Figure 5. Covariance error per fusion of dynamic_state.mat estimation by using Kalman filter