

Waleed A. Abdulhafiz* and Alaa Khamis

Bayesian Approach with Pre- and Post-Filtering to Handle Data Uncertainty and Inconsistency in Mobile Robot Local Positioning

Abstract: One of the important issues in mobile robots is finding the position of robots in space. This is normally achieved by using a sensor to locate the position of the robot. However, relying on more than one sensor and then using multisensor data fusion algorithms tends to be more reliable than just using a reading from a single sensor. If these sensors provide inconsistent data, catastrophic fusion may occur, and thus the estimated position of the robot obtained will be less accurate than if an individual sensor is used. This article uses an approach that relies on combining modified Bayesian fusion algorithm with Kalman filtering to estimate the position of a mobile robot. Two case studies are presented to prove the efficiency of the proposed approach in estimating the position of a mobile robot. Both scenarios show that combining fusion with filtering provides an accurate estimate of the location of the robot by handling the problem of uncertainty and inconsistency of the data provided by the sensors.

Keywords: Multisensor data fusion, Bayesian approach, Kalman filtering, mobile robot positioning, ICNSC2013.

*Corresponding author: Waleed A. Abdulhafiz, Low and Medium Voltage Division, SIEMENS, Cairo, Egypt, e-mail: waleed.ashraf@siemens.com

Alaa Khamis: Engineering Science Department, Suez University, Suez, Egypt

1 Introduction

Multisensor data fusion is a multidisciplinary research area borrowing ideas from many diverse fields such as signal processing, information theory, statistical estimation and inference, and artificial intelligence. This is indeed reflected in the variety of the techniques reported in the literature [8].

Many definitions for data fusion exist in the literature. The Joint Directors of Laboratories [13] defines data fusion as a “multilevel, multifaceted process handling the automatic detection, association, correlation, estimation, and combination of data and information from several sources.” Klein [10] generalizes this definition, stating that data can be provided either by a single source or by multiple sources. Both definitions are general and can be applied in different fields including remote sensing. In [3], the authors present a review and discussion of many data fusion definitions. On the basis of the identified strengths and weaknesses of previous work, a principled definition of data fusion is proposed as the study of efficient methods for automatically or semiautomatically transforming data from different sources and different points in time into a representation that provides effective support for human or automated decision making.

Data fusion finds wide application in many areas of autonomous systems. Autonomous systems must be able to perceive the physical world and physically interact with it through computer-controlled mechanical devices. A critical problem of autonomous systems is the imperfection aspects of the data that the system is processing for situation awareness. These imperfection aspects [8] include uncertainty, imprecision, incompleteness, inconsistency, and ambiguity of the data that may result in wrong beliefs about the system state and/or environment state. These wrong beliefs can lead, consequently, to wrong decisions. To handle this problem, multisensor data fusion techniques are used for the dynamic integration of the multithread flow of data provided by a homogenous or heterogeneous network of sensors into a coherent picture of the situation.

This article discusses how the approach proposed in [1] could be used to handle the imperfection aspects of the data used to estimate the position of a mobile robot. The approach relied on combining a modified Bayesian (MB) fusion algorithm with Kalman filtering [14]. The approaches were pre-filtering, post-filtering, and pre-post-filtering. These approaches have been applied experimentally to handle the problem of data uncertainty and inconsistency in a mobile robot.

The remainder of this article is organized as follows: Section 2 reviews the most commonly used techniques for mobile robot positioning followed by describing Bayesian and geometric approaches in Section 3. The approaches used are presented in Section 4. Two case studies of position estimation of a mobile robot are discussed in Section 5 to show the efficacy of the proposed approaches. Finally, the conclusion and future work are summarized in Section 6.

2 Mobile Robot Positioning

Mobile robot positioning provides an answer for the question: “Where is the robot?” Positioning technique solutions can be roughly categorized into relative position measurements (dead reckoning) and absolute position measurements. In the former, the robot position is estimated by applying to a previously determined position the course and distance traveled since. In the latter, the absolute position of the robot is computed from measuring the direction of incidence of three or more actively transmitted beacons, using artificial or natural landmarks, or using model matching to estimate the absolute location of the robot [9]. This section introduces the different techniques used to find the position of a robot [2, 4, 12]. However, there will always be an error in the readings provided by these techniques, and therefore, the concept of multisensor data fusion is commonly used to handle different imperfection aspects of the data.

2.1 Odometry

Odometry is a relative positioning method that uses encoders to measure wheel rotation and/or steering orientation of the robot. An optical encoder is an example for odometry that converts rotary motion to a series of pulses to know the distance moved by a robot’s wheel. Using odometry to estimate the position of a mobile robot has the advantage of being self-contained and always capable of providing an estimate of the position. The disadvantage is that errors grow without bound unless an independent reference is used periodically to reduce the error.

2.2 Inertial Navigation

Another relative positioning technique is inertial navigation. For inertial navigation, gyroscopes and accelerometers are used to measure the rate of rotation and acceleration, respectively. This positioning technique is self-contained and the measurements are integrated once or twice to yield the position. However, the inertial sensor data drifts with time due to the integration constant, and thus, any small error increases without bound after integration. Therefore, it is unsuitable for accurate positioning over an extended period of time. Another disadvantage of inertial navigation is the high cost of the sensors used.

2.3 Active Beacons

Active beacons compute the absolute position of the robot from measuring the direction of incidence of three or more actively transmitted beacons. Examples of active beacons are ground-based RF systems and global positioning systems.

2.4 Artificial Landmark

To allow for position estimation, distinctive landmarks are placed at known locations in the environment; three or more landmarks must be in view. The landmarks can be designed for optimal detectability even under adverse environmental conditions. Position errors are bounded but detection of external landmarks and real time position fixing may not always be possible.

2.5 Natural Landmark

The landmarks are distinctive features in the environment so there is no need for preparation of the environment. However, the reliability is not as high as that of artificial landmarks.

2.6 Model Matching

In model matching, the information acquired from the robot's sensors is compared to a map or a model of the environment. If features match, then the robot's absolute location can be estimated.

3 Data Uncertainty and Inconsistency

Combining data from several sources using multisensor data fusion algorithms exploits the data redundancy to reduce the uncertainty. However, if these several sources provide inconsistent data, catastrophic fusion may occur where the performance of multisensor data fusion is significantly lower than the performance of each of the individual sensors. This section discusses four different approaches with different levels of complexity and ability to handle uncertainty and inconsistency.

3.1 Geometric Redundant Fusion (GRF)

The GRF method is another method used for fusing uncertain data coming from multiple sensors. The fusing of m uncertain n -dimensional data points is based on a weighted linear sum of the measurements as follows:

$$x_f = \sum_{i=1}^m W_i z_i, \quad (1)$$

where x_f is the fused value and W_i is the weighting matrix for measurement i , z_i . Applying expected values to (1) and assuming no measurement bias yields the condition

$$\sum_{i=1}^m W_i = I. \quad (2)$$

For a given set of data, the weighting matrix, W , and the covariance matrix, Q , are formed as follows:

$$W = [W_1 \ W_2 \ \dots \ W_m], \quad (3)$$

$$Q = \begin{pmatrix} \sigma_1^2 & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & \sigma_m^2 \end{pmatrix}, \quad (4)$$

where σ_i^2 is the uncertainty ellipsoid matrix for measurement i . Using Lagrange multipliers, the following results for the weighting matrices and fused variance are obtained:

$$W_i = \left(\sum_{i=1}^m \sigma_i^{-2} \right)^{-1} \sigma_i^{-2}, \quad (5)$$

$$\sigma_f^2 = \left(\sum_{i=1}^m \sigma_i^{-2} \right)^{-1}. \quad (6)$$

In the one-dimensional (1D) case with two measurements the fused result becomes

$$x_f = \frac{\sigma_1^2 z_2 + \sigma_2^2 z_1}{\sigma_1^2 + \sigma_2^2}, \quad (7)$$

$$\sigma_f^2 = \frac{1}{\sigma_1^{-2} + \sigma_2^{-2}}. \quad (8)$$

Although the GRF method handles the fusion of m measurements in n -dimensional space in an efficient manner, it does not include information about the spacing of the means of the m measurements in the calculation of the fused variance. That is, the magnitude of the spatial separation of the means is not used in the calculation of the uncertainty of the fused result. Figures 1 and 2 show fusing of two 1D measurements using GRF. It can be observed that the uncertainty of the GRF remained the same regardless of the separation or the inconsistency of the measurements.

Therefore, the GRF method provides a fused result with identical uncertainty independent of whether the measurements have identical or highly spatially separated means. To overcome this problem, a heuristic method was developed to consider the level and direction of measurement uncertainty as reflected in the level and direction of disparity between the original measurements. Thus, the output is no longer purely statistically based but can still provide a reasonable measure of the increase or decrease of uncertainty of the fused data.

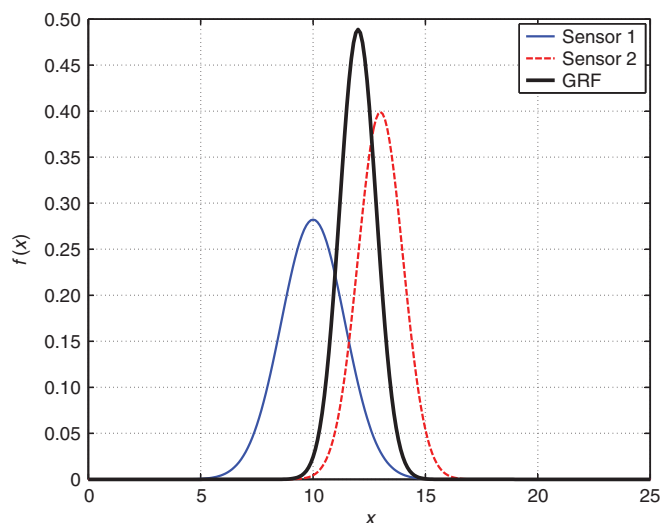


Figure 1. 1D Fusion of Two Measurements in Agreement.

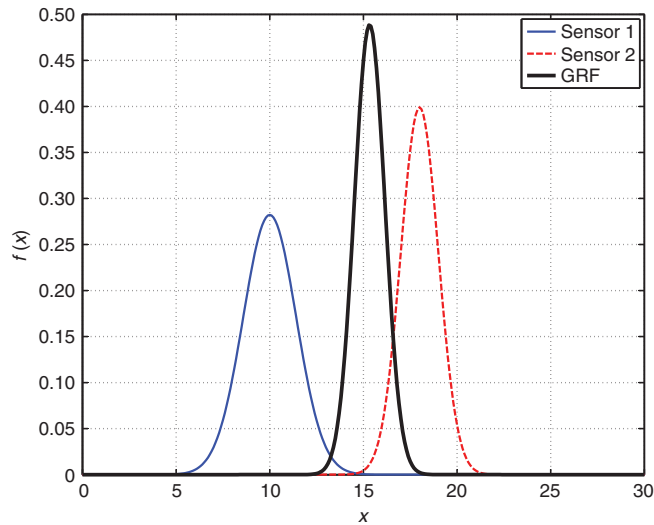


Figure 2. 1D Fusion of Two Measurements in Disagreement.

3.2 Heuristic Geometric Redundant Fusion (HGRF)

The desired heuristic would modify the GRF result so that reliable results are produced for all ranges of measurement disparity or inconsistency. It will contain information about the separation of the mean of the measurements when finding the uncertainty of the fused result [5]. The following cases show how the HGRF result changes as the separation between the measurements increase. The uncertainty of each measurement and of the fused result is shown in the form of an ellipsoid.

Figure 3 shows two measurements with no separation, or consistent. For this case, the HGRF uncertainty region is equivalent to the uncertainty region generated by the GRF method.

Figure 4 shows two measurements that somehow partially agree. The HGRF uncertainty region increased relative to the GRF, which is the same as in Figure 3; however, the HGRF uncertainty decreased relative to the two sensors' uncertainty.

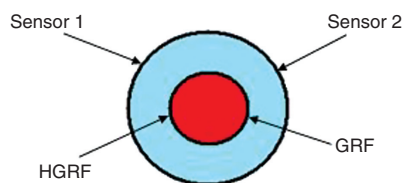


Figure 3. No Separation Between Measurements.

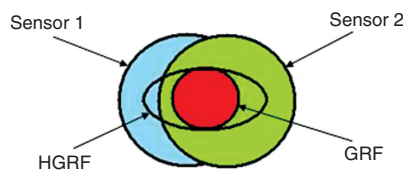


Figure 4. Measurements Agree.

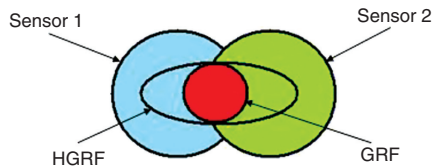


Figure 5. Measurements Disagree.

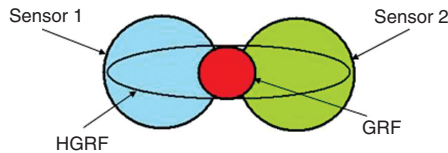


Figure 6. Measurement Error.

Figure 5 shows two measurements in disagreement, or inconsistent. The HGRF uncertainty region is larger than the uncertainty of each sensor, and thus the uncertainty increases. It can be observed that the GRF uncertainty remained the same as in the previous cases.

Figure 6 shows two measurements completely separated or inconsistent, which indicates measurement error. The resulting HGRF uncertainty ellipsoid covers the entire range of the measurement ellipsoids along the dimensions of measurement error. In this case, the increase in the uncertainty indicates the occurrence of measurement error.

The derivation of the fusion heuristic begins with the simple case of fusing two 1D measurements. Then the method is extended to the fusion of m measurements that are n -dimensional. The first step is to calculate the fused mean, x_f , and the fused variance, σ_f^2 , using the GRF method given in equations (7) and (8). This yields the fused result with no adjustments for measurement bias or error. The next step is to calculate the measurement spacing vectors, h_i , that quantify the vector distance between the measurements and x_f . The measurement spacing vectors provide an indication of the relative disparity of each measurement. From h_i , the separation vector r is calculated, which quantifies the overall separation of the measurements with respect to the x_f . The components of r are determined as follows:

1. The first axis of the separation vector, r_1 , is set to the magnitude of the largest h_i . The following equation provides this result:

$$r_1 = \max(|h_i|), \quad i=1, \dots, m. \quad (9)$$

2. The second axis of the separation vector, r_2 , is set to the largest projection of an h_i on a plane normal to the largest h_i . This is done by the following equation:

$$r_2 = \max \left(\left| h_i - \frac{h_i \bullet r_1}{r_1 \bullet r_1} r_1 \right| \right), \quad i=1, \dots, m \quad i \neq l. \quad (10)$$

3. The remaining axes of the separation vector, r_i , are determined similarly according to the following equation:

$$r_i = \max \left(\left| h_i - \frac{h_i \bullet r_1}{r_1 \bullet r_1} r_1 - \dots - \frac{h_i \bullet r_{i-1}}{r_{i-1} \bullet r_{i-1}} r_{i-1} \right| \right), \quad (11)$$

where h_i has not been used previously to determine an axis of r .

The objective of the previous step is to quantify the level of disparity between all the measurements. Then, apply an uncertainty scaling function $F_c(r)$. The uncertainty scaling function determines how σ_f^2 should be

adjusted. This adjustment depends on the size of the separation vector, r . Finally, apply $F_c(r)$ to σ_f^2 in an appropriate frame of reference to get the HGRF fused variance, σ_N^2 [5]. Algorithm 1 summarizes those steps.

Algorithm 1: Heuristic Geometric Redundant Fusion

Algorithm (HGRF)

Input: σ_1, σ_2, Z
Output: x_f, σ_N^2
1 begin
2 Calculate x_f and σ_f^2 using (7) and (8), respectively
3 $h_i \leftarrow \sigma_i^{-1}(x_f - z_i)$
4 Calculate r using (9), (10), and (11)
5 Calculate $F_c(r)$ using (12) and (13)
6 $\sigma_N^2 \leftarrow F_c(r)\sigma_f^2$

To derive a relationship between the separation r and the uncertainty scaling function, the following conditions must be satisfied:

- $F_c(r=0)=1$
- $F_c(r=0.5)=\sqrt{m}$
- $F_c(r=1)=2\sqrt{m}$

and subject to the following constraints:

- $\frac{dF_c}{dr}$ at $r=0$, should equal 0
- $\frac{dF_c}{dr}$ at $r=1$, should equal m

where m is the number of measurements being fused. Fitting a fourth-order polynomial to the above constraints yields the following curve for r in the range $0 \leq r \leq 1$:

$$F_c(r) = 1 + (7\sqrt{m} - 11)r^2 - (7\sqrt{m} - 18)r^3 + (2\sqrt{m} - 8)r^4, \quad (12)$$

and in the region of $r > 1$, the curve must be

$$F_c(r) = r\sqrt{m} + \sqrt{m}. \quad (13)$$

3.3 Simplified Bayesian (SB) Approach

Bayesian inference is a statistical data fusion algorithm based on Bayes' theorem of conditional or a posteriori probability to estimate an n -dimensional state vector X after the observation or measurement Z has been made. Assuming a state-space representation, the Bayes estimator provides a method for computing the posterior (conditional) probability distribution/density of the hypothetical state x_k at time k given the set of measurements $Z^k = \{z_1, \dots, z_k\}$ (up to time k) and the prior distribution as follows:

$$p(x_k | Z^k) = \frac{p(z_k | x_k) p(x_k | Z^{k-1})}{p(Z^k | Z^{k-1})}, \quad (14)$$

where

- $p(z_k|x_k)$ is called the likelihood function and is based on the given sensor measurement model.
- $p(x_k|Z^{k-1})$ is called the prior distribution and incorporates the given transition model of the system.
- The denominator is merely a normalizing term to ensure that the probability density function integrates to one.

The probabilistic information contained in Z about X is described by the probability density function $p(Z|X)$, which is a sensor-dependent objective function based on observation. The likelihood function relates the extent to which the a posteriori probability is subject to change and is evaluated either via offline experiments or by utilizing the available information about the problem. If the information about the state X is made available independently before any observation is made, then likelihood function can be improved to provide more accurate results. Such a priori information about X can be encapsulated as the prior probability and is regarded as subjective because it is not based on observed data. The information supplied by a sensor is usually modeled as a mean about a true value, with uncertainty due to noise represented by a variance that depends on both the measured quantities themselves and the operational parameters of the sensor. A probabilistic sensor model is particularly useful because it facilitates a determination of the statistical characteristics of the data obtained. This probabilistic model captures the probability distribution of measurement by the sensor z when the state of the measured quantity x is known. This distribution is extremely sensor specific and can be experimentally determined. Gaussian distribution is one of the most commonly used distributions to represent the sensor uncertainties and is given by the following equation:

$$p(Z=z_j | X=x) = \frac{1}{\sigma_j \sqrt{2\pi}} \exp \left\{ -\frac{(x-z_j)^2}{2\sigma_j^2} \right\}, \quad (15)$$

where j represents the sensors. Thus, if there are two sensors that are modeled using (15), then from Bayes' theorem the fused mean of the two sensors is given by the maximum a posteriori (MAP) estimate,

$$x_f = \frac{\sigma_2^2}{\sigma_1^2 + \sigma_2^2} z_1 + \frac{\sigma_1^2}{\sigma_1^2 + \sigma_2^2} z_2, \quad (16)$$

where σ_1 is the standard deviation of sensor 1 and σ_2 is the standard deviation of sensor 2. The fused variance is given by

$$\sigma_f^2 = \frac{1}{\sigma_1^{-2} + \sigma_2^{-2}}. \quad (17)$$

Algorithm 2 summarizes the steps of SB fusion.

Algorithm 2: Simplified Bayesian (SB)

Input: $\sigma_1, \sigma_2, z_1, z_2$
Output: x_f, σ_f^2
1 begin
2 Calculate x_f **as given in (16)**
3 Calculate σ_f^2 **as given in (17)**

3.4 Modified Bayesian (MB) Approach

Sensors often provide data that are spurious due to sensor failure or due to some ambiguity in the environment. The SB approach described previously does not handle the spurious data efficiently. The approach yields the same weighted mean value whether data from one sensor is bad or not, and the posterior distribution always

has a smaller variance than either of individual distributions being multiplied. This can be seen in (16). The SB does not have a mechanism to identify if data from a certain sensor is spurious, and thus it might lead to inaccurate estimation. In [6], an MB approach has been proposed, which considers measurement inconsistency.

$$p(X=x|Z=z_1, z_2) \propto \frac{1}{\sigma_1 \sqrt{2\pi}} \exp\left\{-\frac{(x-z_1)^2}{2\sigma_1^2 f}\right\} \times \frac{1}{\sigma_2 \sqrt{2\pi}} \exp\left\{-\frac{(x-z_2)^2}{2\sigma_2^2 f}\right\} \quad (18)$$

As shown in [6], modification observed in (18) causes an increase in the variance of the individual distribution by a factor given by

$$f = \frac{M^2}{M^2 - (z_1 - z_2)^2}. \quad (19)$$

The parameter M is the maximum expected difference between the sensor readings. Larger difference in the sensor measurements causes the variance to increase by a bigger factor. The MAP estimate of state x remains unchanged but the variance of the fused posterior distribution changes. Thus, depending on the squared difference in measurements from the two sensors, the variance of the posterior distribution may increase or decrease as compared to the individual Gaussian distributions that represent the sensor models. Algorithm 3 shows the steps of MB fusion.

Algorithm 3: Modified Bayesian (MB)

Input: $\sigma_1, \sigma_2, x_1, x_2$
Output: x_f, σ_f^2
1 begin
2 $\xi \leftarrow \sigma_1 / \sigma_2$
3 Calculate x_f as given in (16)
4 Calculate the factor f as in (19)
5 Find the fused variance by multiplying each σ_i^2 in (17) by f

The difference between the SB and the MB can be seen in Figures 7 and 8. In this example, there are two sensors where sensor 1 has a standard deviation of 2 and sensor 2 has a standard deviation of 4. In the first

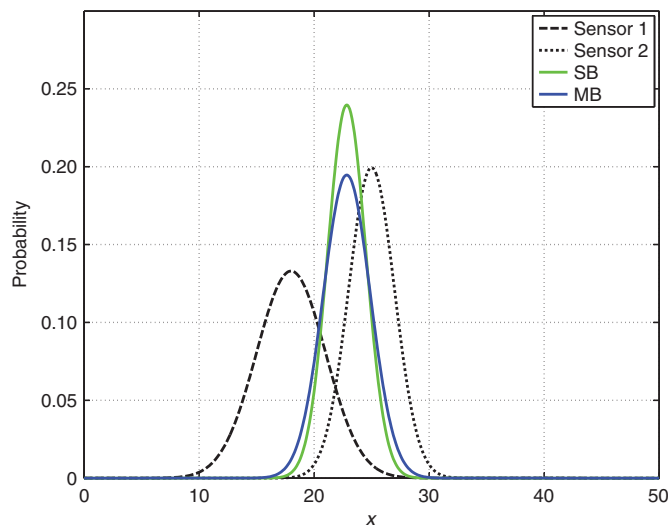


Figure 7. Two Sensors in Agreement.

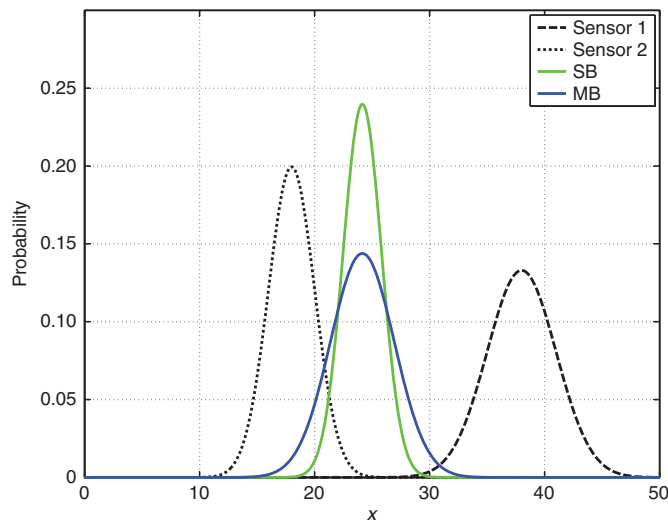


Figure 8. Two Sensors in Disagreement.

case shown in Figure 7, the two sensors are in agreement. It can be seen that fused posterior distribution obtained from the proposed strategy has a lower value of variance than that of each of the distributions being multiplied, indicating that fusion leads to a decrease in posterior uncertainty.

In the second case in Figure 8, the two sensors are in disagreement. The fused posterior distribution obtained from the MB has a larger variance as compared to the variance of both sensors. However, the fused variance due to the SB was the same as the fused variance in Figure 7. This concluded as in [6] that the MB was highly effective in identifying inconsistency in sensor data and thus reflecting the true state of the measurements.

3.5 Comparative Study

To evaluate the difference between the four approaches mentioned previously, a comparative study was carried out. This study considers a mobile robot moving in a straight line with a constant velocity of 7.8 cm/s. The position of the robot is tracked using two measurements coming from two sensors: optical encoder and Hall effect sensor. To detect the position of the robot, the readings coming from the sensors are being fused, using SB and MB as well as GRF and HGRF methods, during 20 s of travel. The standard deviations of the optical encoder and Hall effect sensor are 2.378 and 2.260 cm, respectively. Therefore, the measurements coming from the optical encoder have a higher uncertainty.

Figure 9 shows the uncertainty curves of the sensors as well as the fused results at $t=10$ s. It can be observed that the variance of the GRF and the SB were the same, and thus the uncertainty curves were completely overlapping. However, the HGRF result followed the heuristic specification and covered the uncertainty curves of the measurements. Compared to the SB and the GRF, the MB showed a response to the separation or inconsistency of the measurements, however, not as much as the HGRF. The fused mean of the four approaches was exactly the same value of 78.4 cm, whereas the mean values of the optical encoder and Hall effect sensor were 75.4 and 79.1 cm, respectively. The fused mean is more biased towards the accurate reading of the Hall effect sensor.

To compare the different approaches in terms of the calculation time, Figure 10 shows how the running time of each algorithm changes over the 20 s. It can be seen that the HGRF method is always taking a longer time compared to the other methods. The other approaches have approximately the same running time.

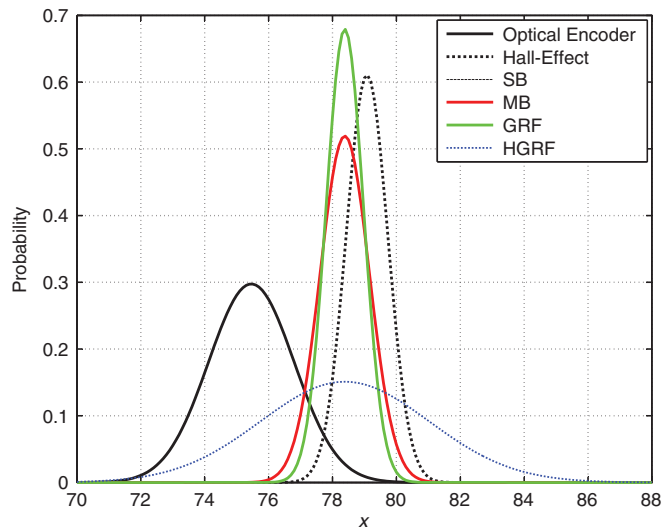


Figure 9. Uncertainty Curves.

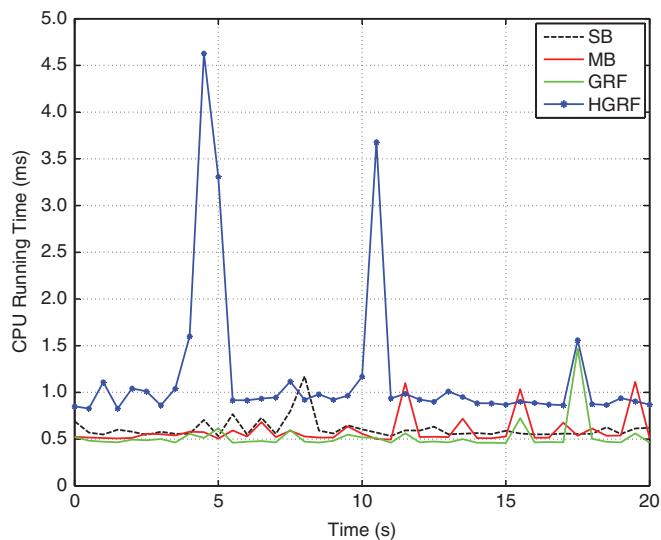


Figure 10. Time Taken to Perform Each Algorithm Throughout Simulation Time.

It can be observed from Figure 11 that the HGRF shows a very big change in the fused variance due to its high sensitivity to measurement inconsistency. The MB also showed a response to the separation inconsistency between the measurements; however, it was not as great as that of the HGRF. The fused variance using both the SB and the GRF was exactly the same as seen by the perfectly overlapping curves, and the variance value did not change throughout the simulation time.

The error curves in Figure 12 show that the fusion result is generally more accurate with fewer errors compared to the error caused by the uncertainty of each measurement. The optical encoder shows a higher range of error due to its higher uncertainty than the Hall effect sensor.

Table 1 summarizes the main differences between each of the four multisensor data fusion approaches. In general, the MB outperforms all the other approaches in terms of accuracy, time, and variance change.

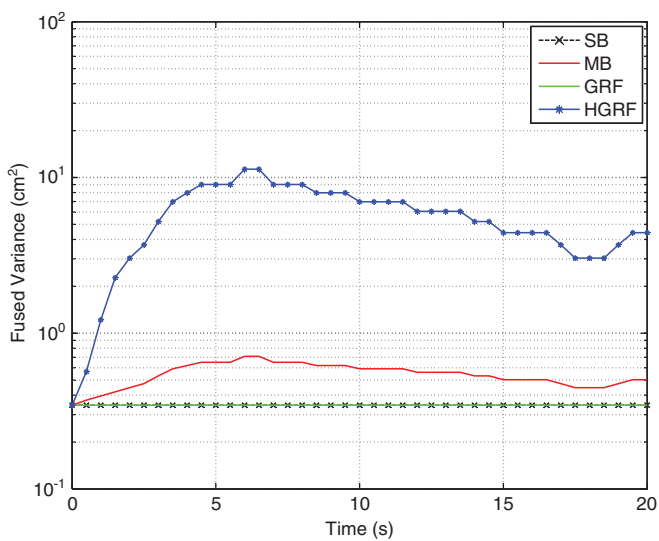


Figure 11. Trend of the Fused Variance Throughout Simulation Time.

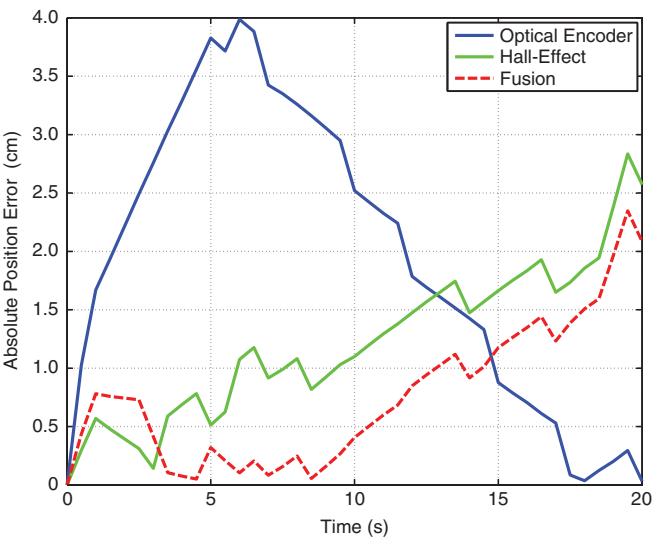


Figure 12. Measurement and Fusion Errors.

Table 1. Comparison Between Different Fusion Techniques.

	Fused Variance	Running Time	Trend of Variance	Fused Mean
SB	Not affected by inconsistency	Medium	No change	Same for all methods
MB	Deals with data inconsistency	Short	Smooth change	
GRF	Not affected by inconsistency	Shortest	No change	
HGRF	Deals with data inconsistency	Longest	Big change	

4 Proposed Approach

The three proposed approaches that were presented in [1] rely on combining an MB fusion algorithm with Kalman filtering. The three techniques are pre-filtering, post-filtering, and pre-post-filtering. The following subsections describe how filtering is applied to the sensor data, the fused data, or both.

4.1 Modified Bayesian Fusion with Pre-Filtering (F-MB)

The first proposed technique is the pre-filtering (F-MB), which involves adding Kalman filters before the MB fusion node. A Kalman filter is added to every sensor to filter out noise from sensor measurements, as illustrated in Figure 13. The filtered measurements are then fused together using MB to get a single result that represents the state at a particular instant of time. Algorithm 4 shows the steps of the F-MB.

Algorithm 4: The Pre-Filtering Algorithm (F-MB)

Input: $\sigma_1, \sigma_2, z_1(k), z_2(k), x_1(k-1), x_2(k-1),$
 $P_1(k-1), P_2(k-1)$
Output: $x_f(k), \sigma_f^2(k)$

```

1 begin
2    $\xi \leftarrow \sigma_1 / \sigma_2$ ;
3   for  $j \leftarrow 1$  to 2 do
4      $(x_j(k), P_j(k)) \leftarrow$  Call Kalman Filter Algorithm
5    $x_f(k) \leftarrow x_1(k) / (1 + \xi^2) + x_2(k) / (1 + \xi^{-2})$ ;
6   Calculate  $f$  as in (19);
7    $\sigma_f^2(k) \leftarrow (\sigma_1^{-2} f^{-1} + \sigma_2^{-2} f^{-1})^{-1}$ ;

```

4.2 Modified Bayesian Fusion with Post-Filtering (MB-F)

The second proposed technique is the post-filtering (MB-F), which involves adding a Kalman filter after the fusion node in order to filter out the noise from the fused estimate, as shown in Algorithm 16 and illustrated in Figure 14. The second proposed technique is to add a Kalman filter after the fusion node, which fuses the measurements using MB to produce x_{int} . Kalman filtering is then applied to the fused state x_{int} in order to filter out the noise, as shown in Algorithm 5. The output of the Kalman filter represents the state x_f at a particular instant of time as well as the variance of the estimated fused state P_f .

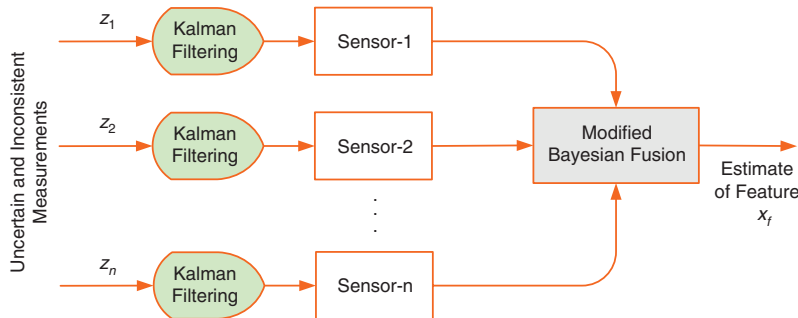


Figure 13. Modified Bayesian Fusion with Pre-Filtering.

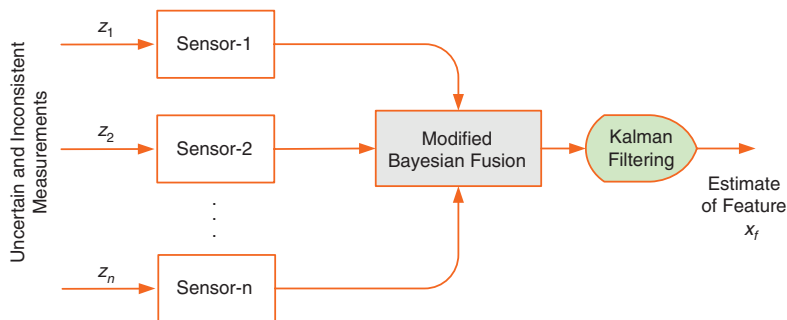


Figure 14. Modified Bayesian Fusion with Post-Filtering.

Algorithm 5: The Post-Filtering Algorithm (MB-F)

Input: $\sigma_1, \sigma_2, z_1(k), z_2(k), x_f(k-1), P_f(k-1)$
Output: $x_f(k), P_f(k)$

1 begin
2 $\xi \leftarrow \sigma_1 / \sigma_2$
3 $x_{\text{int}}(k) \leftarrow z_1(k) / (1 + \xi^2) + z_2(k) / (1 + \xi^{-2})$
4 Calculate f as in (19)
5 $\sigma_{\text{int}}^2(k) \leftarrow (\sigma_1^{-2} f^{-1} + \sigma_2^{-2} f^{-1})^{-1}$;
6 $(x_f(k), P_f(k)) \leftarrow \text{Call Kalman Filter Algorithm}$;

4.3 Modified Bayesian Fusion with Pre- and Post-Filtering (F-MB-F)

In this technique, Kalman filter is applied before and after the fusion node as illustrated in Figure 15. The algorithm of this technique is the integration of Algorithms 14 and 16, as shown in Algorithm 6.

Algorithm 6: The Pre- and Post-Filtering Algorithm (F-MB-F)

Input: $\sigma_1, \sigma_2, z_1(k), z_2(k), x_1(k-1), x_2(k-1), x_f(k-1), P_f(k-1),$
 $P_2(k-1), P_f(k-1)$
Output: $x_f(k), P_f(k)$

1 begin
2 $\xi \leftarrow \sigma_1 / \sigma_2$;
3 for $j \leftarrow 1$ **to** 2 **do**
4 $x_j(k) \leftarrow \text{Call Kalman Filter Algorithm}$;
5 $x_{\text{int}}(k) \leftarrow x_1(k) / (1 + \xi^2) + x_2(k) / (1 + \xi^{-2})$;
6 Calculate f as in (19);
7 $\sigma_{\text{int}}^2(k) \leftarrow (\sigma_1^{-2} f^{-1} + \sigma_2^{-2} f^{-1})^{-1}$;
8 $(x_f(k), P_f(k)) \leftarrow \text{Call Kalman Filter Algorithm}$;

5 Experimental Results

In this section, the proposed techniques are applied in a mobile robot position estimation problem. These techniques are applied in two mobile robot positioning case studies. The first case study estimates the position of the robot in a 1D problem and the second case study is a 2D problem.

5.1 Evaluation Metrics

The performance of the algorithms was evaluated on the basis of five criteria:

- CPU running time: This represents the total processing time of the algorithm to estimate the position of the robot throughout the traveling time. It is desired to minimize this running time.

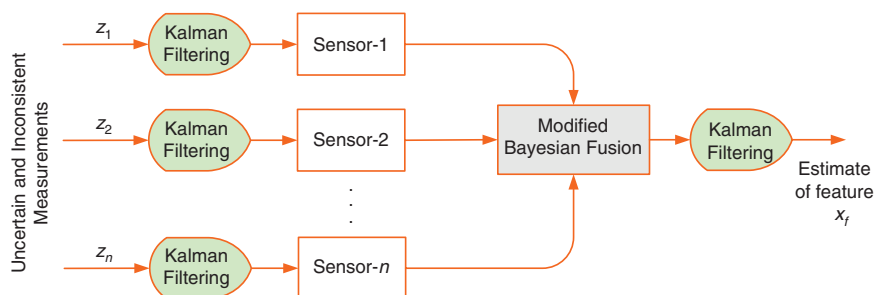


Figure 15. Modified Bayesian Fusion with Pre- and Post-Filtering.

- Residual sum of squares (RSS): This represents the summation of the squared difference between the theoretical position of the robot and the estimated state at each time sample. The smaller the RSS, the more accurate the algorithm will be because this means that the estimated position of the robot is getting closer to the theoretical position. This is given by:

$$\text{RSS} = \sum_{i=1}^n (x_{\text{theoretical},i} - x_{\text{estimated},i})^2. \quad (20)$$

- Variance (P): This represents the variance of the estimated position of the robot. The variance will reflect the performance of the filters in each algorithm.
- Coefficient of correlation: This is a measure of association that shows how the state estimate of each technique is related to the theoretical state. The coefficient of correlation will always lie between -1 and $+1$. For example, a correlation close to $+1$ indicates that the two data are very strongly positively correlated [7].
- Criterion function (CF): A computational decision-making method was used to calculate a CF that is a numerical estimate of the utility associated with each of the three proposed techniques. A weighting function w (from 0 to 1) will be defined for each criterion (time, RSS, variance), depending on its importance. The three weights should sum up to 1. The cost value c (calculated from the experiments) of each technique is obtained, and finally, CF is calculated as the weighted sum of the utility for each technique as follows:

$$\text{CF} = w_1 \times \frac{c_1}{c_{1\max}} + w_2 \times \frac{c_2}{c_{2\max}} + w_3 \times \frac{c_3}{c_{3\max}}, \quad (21)$$

where w_1 , w_2 , and w_3 are the weights of the time, RSS, and variance respectively. These weights are adjusted according to the application and desire of the user. In this case study, it was assumed that $w_1=0.3$, $w_2=0.3$, and $w_3=0.4$. The values c_1 , c_2 , and c_3 are the values obtained from the experiments for the time, RSS, and variance, respectively. The values $c_{1\max}$, $c_{2\max}$, and $c_{3\max}$ represent the maximum value achieved in each of the criteria: time, RSS, and variance, respectively. The objective is to minimize this function such that the algorithm will produce accurate estimates in a short time with a minimum variance.

5.2 Case Study 1: 1D Scenarios

For the first case study, a differential drive mobile robot is used in the experiments. The right wheel's motor is equipped with a Hall effect sensor, while the left wheel's motor is equipped with an optical encoder. The microcontroller aligns the data and adjusts the resolutions so that both sensors can have the same resolution and the data could be compared with each other. The distance moved by each wheel is calculated given the sampling time of the data and the constant velocity of the robot. The microcontroller then sends the data to Matlab to perform the proposed algorithms in order to estimate the position of the robot given the uncertain and inconsistent data collected from the sensors.

Several experiments were carried out in order to model the uncertainty of each sensor, which was represented in the form of a white Gaussian noise with a standard deviation of 2.378 and 2.260 cm for the optical encoder and Hall effect sensor, respectively. In addition, it was specified to have the robot moving with a constant linear velocity of 7.8 cm/s and a disturbance of approximately 0.493 cm/s. The sampling time is 0.5 s and the robot should move in a straight line for 20 s.

5.2.1 Measurements from Two Sensors

The raw data collected from the sensors were real time and were used to carry out 5000 iterations in order to simulate the results of the proposed algorithms using Matlab. Figure 16 shows the errors that are produced

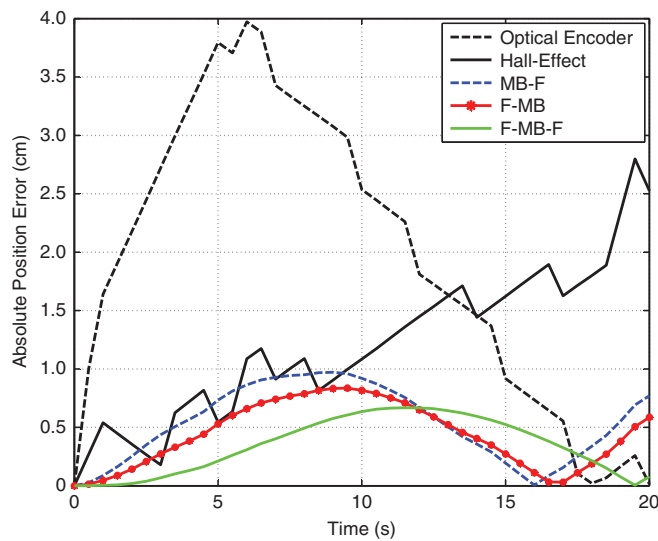


Figure 16. Measurements and Estimated Errors of a 1D Scenario.

due to the noisy uncertain measurements obtained from the sensors. In addition, the figure shows the errors that are produced due to estimating the position of the robot using the proposed three algorithms. The error represents the difference between the theoretical state and the output state of each algorithm at a particular sample time. The measurement errors are high compared to the estimation errors. The first interpretation of this result is that the proposed techniques provide better estimates than relying on the measurements directly.

Table 2 summarizes the average results of 5000 runs for the three evaluation metrics described previously. The minimal value in each criterion has been bolded. Although MB takes the minimal execution time, yet it has the maximum RSS value compared to other techniques and this is clear in the CF shown in Table 2.

Figure 17 shows the estimated variance of each technique. It is clear that the variance of the MB-F and F-MB-F converged earlier to lower values than MB and F-MB. This proves the efficiency of the Kalman filters in MB-F and F-MB-F. Table 2 shows that MB-F has a smaller variance than F-MB-F. Moreover, the proposed techniques' estimates were all positively and strongly correlated to the theoretical values with a correlation coefficient of +0.99.

It can be seen from Figure 18 that MB-F has outperformed the other techniques, followed by F-MB-F. The least performance was observed in MB, and thus, this proves that combining fusion with Kalman filtering improves the estimation of the states.

5.2.2 Measurements from Three Sensors

The same experiments were repeated but using three sensors. Readings from the sensors were fused using centralized and decentralized architectures. In centralized fusion, the data from all sensors are fused simulta-

Table 2. Computational Decision-Making Chart for the Fusion Techniques Using Two Sensors of a 1D Scenario.

Fusion Techniques	Criteria c_1 , c_2 , c_3 , respectively			Criterion Function
	Time (s)	RSS (cm ²)	P (cm ²)	
MB	0.001	36.729	3.068	0.724
F-MB	0.011	10.885	3.079	0.692
MB-F	0.006	15.713	0.399	0.292
F-MB-F	0.017	7.154	0.405	0.464
c_{\max}	0.017	36.729	3.079	

Bold type represents the minimal value in each criterion.

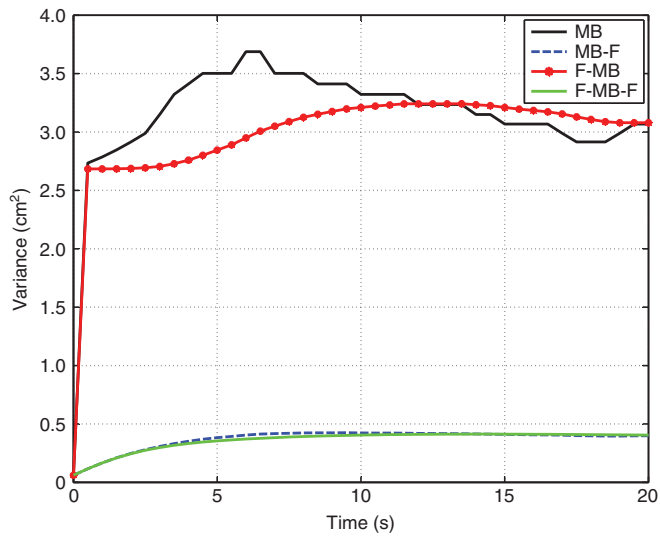


Figure 17. Estimated Variance of 1D Scenario.

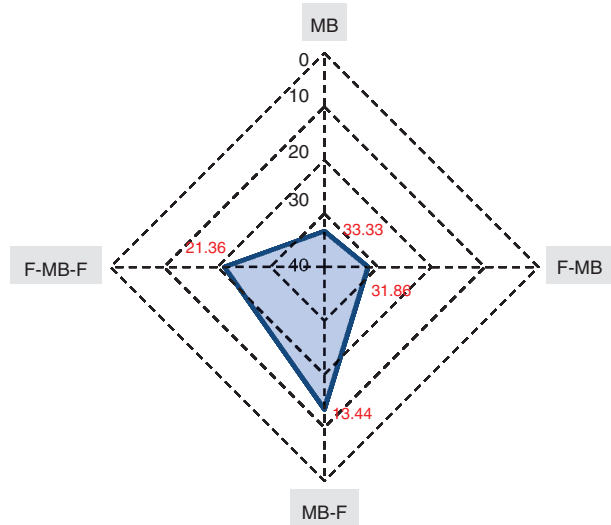


Figure 18. Radar Chart of CF of Two Sensors.

neously, as shown in Figure 19. Alternatively, the data can be fused sequentially, and this is called decentralized fusion, as shown in Figure 20. The decentralized fusion is more robust in terms of individual component failure and is more efficient in using communication resources as compared to conventional schemes [11].

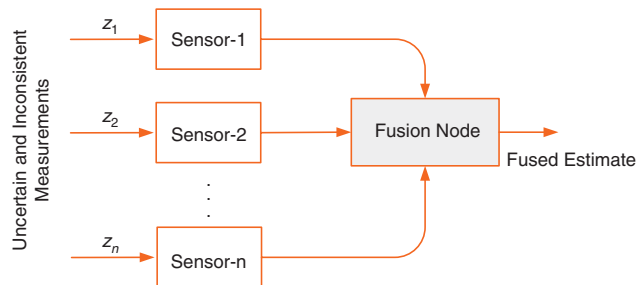


Figure 19. Centralized Fusion Scheme.

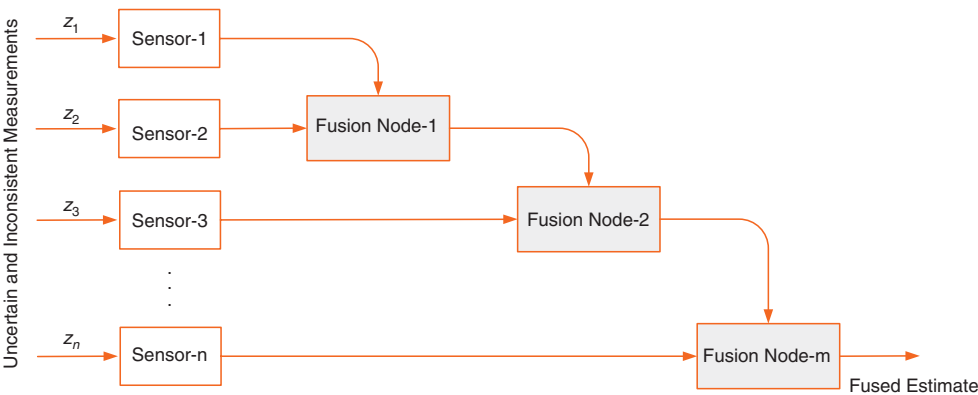


Figure 20. Decentralized Fusion Scheme.

Table 3. Computational Decision-Making Chart for the Fusion Techniques Using Three Sensors of 1D Scenario.

Fusion Techniques	Time (s)		RSS (cm ²)		P (cm ²)		Criterion Function	
	Centralized	Decentralized	Centralized	Decentralized	Centralized	Decentralized	Centralized	Decentralized
MB	0.056	0.029	76.076	70.288	2.607	2.038	0.917	0.701
F-MB	0.072	0.046	31.338	28.764	2.555	2.034	0.795	0.603
MB-F	0.063	0.034	50.838	49.956	0.363	0.322	0.499	0.378
F-MB-F	0.077	0.051	25.108	23.666	0.364	0.322	0.455	0.339
c_{\max}	0.077		76.076		2.607			

As shown in Table 3, the decentralized fusion generally outperforms the centralized in terms of running time, RSS, and estimated variance. However, the shortest running time was the MB of the decentralized system, whereas the longest time was the centralized F-MB-F. This was the same case when fusing using two sensors; the direct fusion was faster than F-MB-F. Regarding RSS, the F-MB-F was the most accurate with minimal error in both cases of the centralized and decentralized, whereas the maximum error occurred in the MB. In Table 3, the underlined value presents the best performance shown by the decentralized system of the F-MB-F. This is a reasonable result because the presence of the Kalman filters produces estimates with less noise than the measurements and close to the accuracy of the theoretical states.

Moreover, the variances of the estimated state in the centralized and decentralized cases were the least in MB-F and F-MB-F, as shown in Figures 21 and 22. There was a noticeable difference between the variance of these techniques and the variance of F-MB and MB. Therefore, this proves that the filtering process is working efficiently in the cases of post-filtering as well as the pre-post-filtering.

Overall, the F-MB-F had the least CF and then following it is the MB-F with a very small difference, as shown in the radar chart in Figure 23. It can be concluded that both techniques will produce reliable results; however, it would be recommended to use MB-F in applications where time is an important factor and to use the F-MB-F if high accuracy is required.

5.3 Case Study 2: 2D Scenario

In this case study, a simulation of 5000 iterations was carried out using Matlab to locate the position of the robot by finding its x - and y -coordinates. It was assumed that two sensors were used to find how much the robot has moved in the x and another two sensors to find how much the robot has moved in the y . The sensors used in the simulation were assumed to have uncertainty modeled as a white Gaussian noise. The standard deviations of the sensors used to find the x -coordinates of the robot are 4.3 and 6.8 cm. However, the standard deviations of the sensors used to find the y -coordinates are 4.5 and 6.6 cm. In addition, it was assumed that

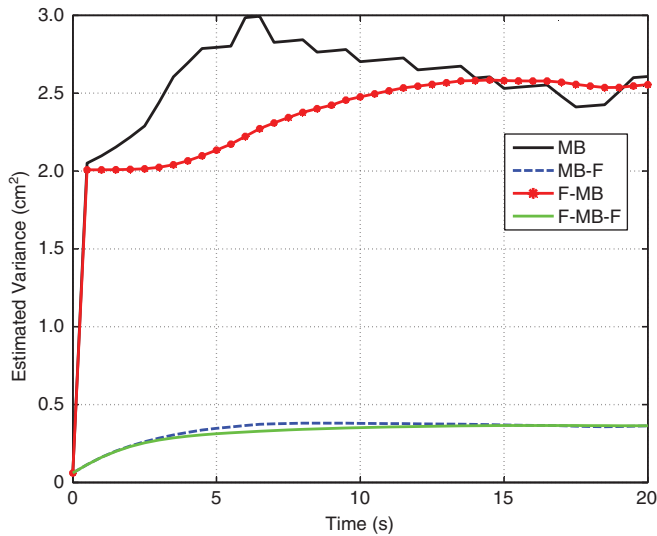


Figure 21. Estimated Variance Using Centralized Fusion.

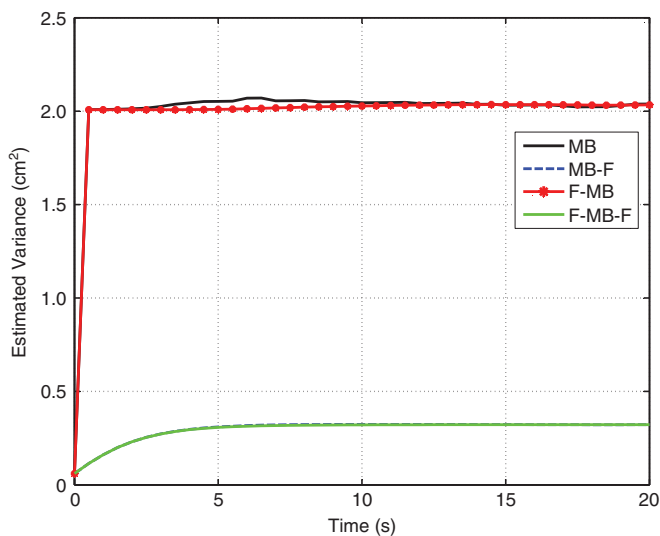


Figure 22. Estimated Variance Using Decentralized Fusion.

the robot is moving with a speed of 7.8 cm/s in the x and 15.6 cm/s in the y . The sampling time is 0.5 s and the robot was simulated to move for 20 s.

Figure 24 shows the errors that are produced due to the noisy uncertain measurements obtained from the sensors. In addition, the figure shows the errors that are produced due to estimating the x -coordinates of the robot using the proposed three algorithms. The error represents the difference between the theoretical state and the output state of each algorithm at a particular sample time. The measurement errors are high compared to the estimation errors.

Figure 25 shows the errors that are produced due to the noisy uncertain measurements obtained from the sensors as well as the errors of the proposed three algorithms to get the y -coordinates of the robot. Similar to the results shown in Figure 24, the measurement errors are high compared to the estimation errors. Thus, the proposed techniques provide better estimates than relying on the measurements directly.

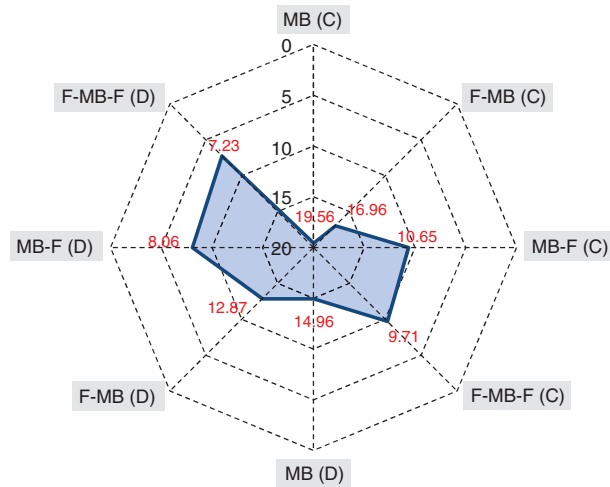


Figure 23. Radar Chart of CF of Three Sensors in 1D Scenario.

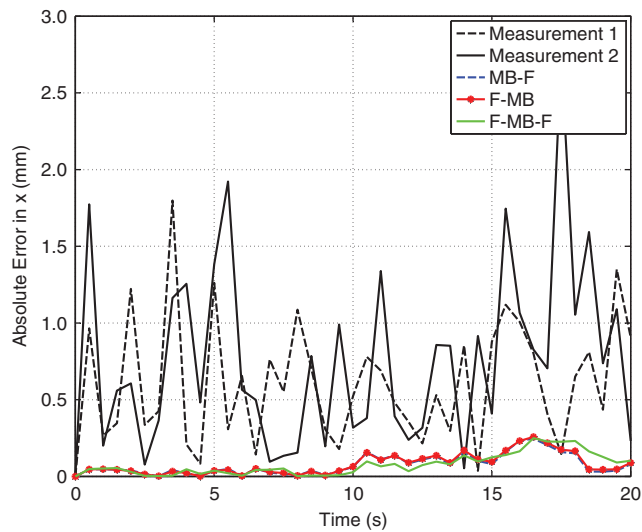


Figure 24. Measurements and Estimated Errors to Find the x-Coordinates of the Moving Robot.

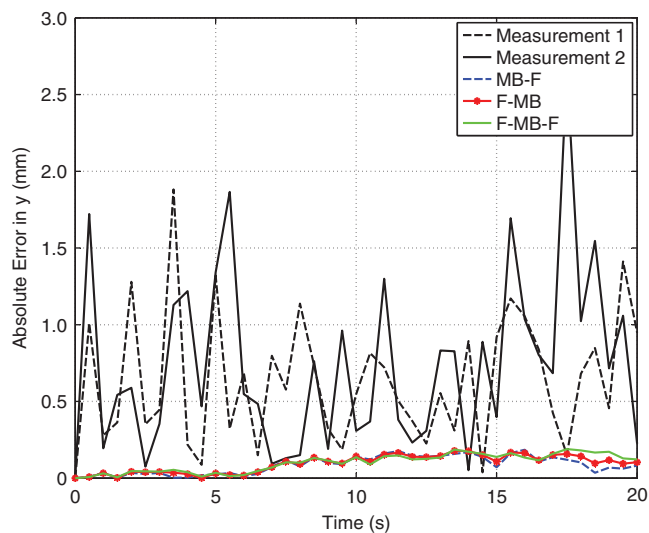


Figure 25. Measurements and Estimated Errors to Find the y-Coordinates of the Moving Robot.

Table 4. Computational Decision-Making Chart for the Fusion Techniques Using Two Sensors of 2D Scenario.

Fusion Techniques	Criteria c_1, c_2, c_3 , respectively			Criterion Function
	Time (s)	RSS (cm ²)	P (cm ²)	
MB	0.001	0.481	15.939	0.703
F-MB	0.004	0.006	11.674	0.461
MB-F	0.003	0.005	0.920	0.173
F-MB-F	0.005	0.006	0.796	0.354
c_{\max}	0.005	0.481	15.939	

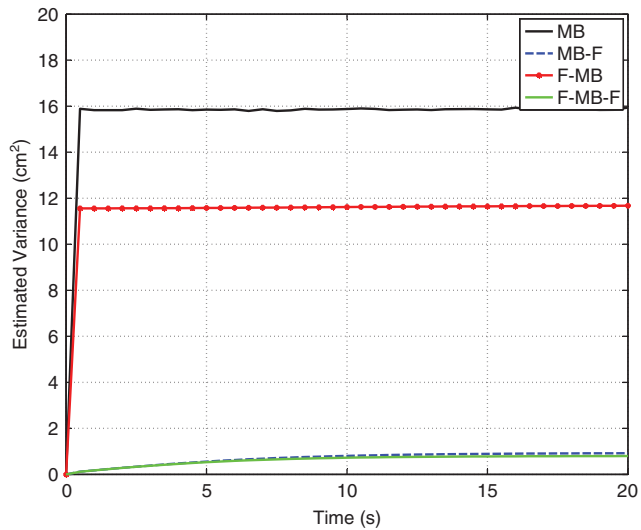
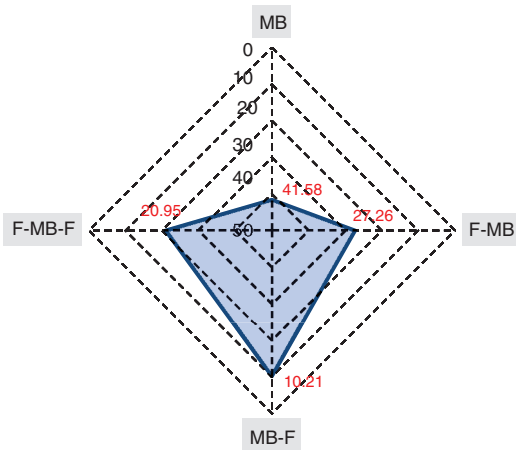
**Figure 26.** Estimated Variance of 2D Scenario.**Figure 27.** Radar Chart of CF of 2D scenario.

Table 4 shows the average results of 5000 runs for the three evaluation metrics described previously. Similar to the results obtained in the 1D scenario, MB takes the minimal execution time, yet it has the maximum RSS value compared to other techniques.

Figure 26 shows that the variance of the MB-F and F-MB-F converged earlier to lower values than those of MB and F-MB. This proves the efficiency of the Kalman filters in MB-F and F-MB-F. Table 4 shows that F-MB-F has a smaller variance than MB-F.

Similar to the result observed in the 1D, MB-F has outperformed the other techniques, followed by F-MB-F, as seen in Figure 27. The least performance was observed in MB, and thus this proves that combining fusion with Kalman filtering improves the estimation of the states.

6 Conclusion

In this article, three techniques for multisensor data fusion have been considered to solve the mobile robot positioning problem. These techniques combine an MB data fusion algorithm with pre- and post-filtering in order to handle the uncertainty and inconsistency problems of sensory data. The experimental results proved that combining fusion with filtering improves the RSS and variance of the fused result. The article also proved that MB-F will still be efficient for any n number of sensors and particularly in the decentralized fusion architecture. Future research is possible in numerous directions. More powerful filtering techniques such as particle filters can be used instead of Kalman filtering to allow for fusion of data with nonlinear dynamics. In addition, the 2D scenario could be applied experimentally on a real robot.

Received September 29, 2013; previously published online January 11, 2014.

Bibliography

- [1] W. A. Abdulhafiz and A. Khamis, Bayesian approach to multisensor data fusion with pre- and post-filtering, in: *IEEE 2013 International Conference on Networking Sensing and Control*, Paris, France, 2013.
- [2] J. Borenstein, H. R. Everett and L. Feng, *Navigating mobile robots: sensors and techniques*, A. K. Peters, Ltd., Wellesley, MA, 1999.
- [3] H. Boström, S. Andler, M. Brohede, R. Johansson, A. Karlsson, J. V. Laere, L. Niklasson, M. Nilsson, A. Persson and T. Ziemke, On the definition of information fusion as a field of research, Technical Report, Informatics Research Centre, University of Skövde, 2007.
- [4] G. Cook, *Mobile robots: navigation, control and remote sensing*, Wiley-IEEE Press, 2011.
- [5] J. D. Elliott, *Multisensor fusion within an encapsulated logical device architecture*, Master's thesis, University of Waterloo, Waterloo, 1999.
- [6] D. Garg, M. Kumar and R. Zachery, A generalized approach for inconsistency detection in data fusion from multiple sensors, in: *American Control Conference*, pp. 2078–2083, June 2006.
- [7] G. Keller and B. Warrack, *Statistics for management and economics*, 4th ed., Duxbury Press, Belmont, California, 1997.
- [8] B. Khaleghi, A. Khamis, F. O. Karray and S. N. Razavi, Multisensor data fusion: a review of the state-of-the-art, in: *Information Fusion*, Elsevier B.V., 2011.
- [9] A. Khamis, Lecture 3 – Mobile robot positioning, MCTR1002: autonomous systems, Mechatronics Engineering Department, German University in Cairo, 2011–2012.
- [10] L. A. Klein, Sensor and data fusion concepts and applications, in: *Society of Photo-Optical Instrumentation Engineers (SPIE)*, Bellingham, WA, USA, 1993.
- [11] M. Kumar, D. P. Garg and R. A. Zachery, A method for judicious fusion of inconsistent multiple sensor data, *IEEE Sens. J.* **7** (2007), 723–733.
- [12] R. Siegwart and I. Nourbakhsh, *Introduction to autonomous mobile robots*, MIT Press, Cambridge, Massachusetts, 2004.
- [13] U.S. Department of Defense, Data Fusion Subpanel of the Joint Directors of Laboratories, Technical Panel for C3, *Data Fusion Lexicon*, 1991.
- [14] G. Welch and G. Bishop, *An introduction to the Kalman filter*, Department of Computer Science, University of North Carolina at Chapel Hill, July 2006.