A Case Study towards Evaluation of Redundant Multi-Sensor Data Fusion

Sebastian Blank, Tobias Föhst, and Karsten Berns

University of Kaiserslautern, Department of Computer Sciences,
Robotics Research Laboratory
{blank, foehst, berns} @ cs.uni-kl.de

Abstract. The focus of this case study is the analysis and evaluation of signal level fusion approaches that are common to the domain of commercial vehicles. For this task a Simulink[1] environment for simulating both sensors and fusion algorithms was implemented and several experiments were conducted. Application oriented scenarios were used to deduce guidelines for the optimal usage (and limitations) of these algorithms.

1 Introduction and Motivation

Data Fusion of redundantly measured data has become a research topic of high interest in the field of commercial vehicles. This is because of the steadily increasing number of sensors that are needed for e. g. system self-monitoring or perception of the environment. Aligning measurements in terms of fusion approaches entails significant benefits like e. g. higher accuracy of the aligned sensor readings and increased system reliability due to redundancy as characterize in [7]. Thus, it gives a more expressive and accurate image of the status (both internal and external) which allows for enhanced usability and reliability. Besides that, it might be even more important that data fusion allows for data consistency throughout the entire machine. This is of special importance when tasks common to commercial vehicles like documentation or automation are pursued. Various approaches are available to deal with the challenge of general data fusion. These approaches can be classified according to the kind of the data used in the fusion process as presented in [2]. The most common fusion level in the field of embedded systems is signal (low) level fusion that contains several dozens approaches. However, not all of them are suited for applications in this specific field due to certain constraints that have to be met. These constraints could be e. g. computational complexity, the availability of a figure of merit or the ability to deal with dynamically changing system configurations. In this paper the following approaches will be discussed and evaluated: weighted average fusion, Kalman filter, median and threshold voters (as discussed in [8]) as well as a modified fuzzy voting algorithm.

[1] Matlab and Simulink are registered trademarks of The MathWorks, Inc.

2 Low Level Fusion Algorithms

2.1 Weighted Average

The most simple and intuitive approach of the afore mentioned is the weighted average approach. Here all available sensor values s_i are summed up with an emphasis factor w_i representing their assumed reliability and quality. In mathematical terms this can be expressed as $out = \sum_{i=0}^{n} w_i \cdot s_i / \sum_{i=0}^{n} w_i$. The main concern when using this kind of fusion approach is to select an optimal set of weighting factors. In case of static weights either a equal emphasis (i. e. $w_i = w_j \; \forall i, j \in [1, n]$) or knowledge of the available sensor hardware is employed to ensure proper weights for each input. In more advanced applications it might be of interest to adapt the weights to the performance of the respective sensor w. r. t. the others determined by a metric. Furthermore, it can be stated that weighted average fusion can be performed under real time conditions (see [5]) due to its simple nature.

2.2 Median Voter

The core idea behind this algorithm is that the element in the middle of a given set of (ordered) sensor readings is (in the average case) the most reliable one. Thus, first of all a sorted list of all sensor readings is generated: $s_i \leq s_j \forall i, j \in [1, n], n \in \mathbb{N}$. The algorithm then starts to discard elements from either side until only one (or two for an even number of sensors) element remains. The output can therefore be determined as:

$$\text{out} = \begin{cases} s_{n/2} & \text{if } n \text{ is odd} \\ (s_{n/2} + s_{n/2+1})/2 & \text{if } n \text{ is even} \end{cases}$$

The upside of such an approach is that if a sufficient degree of redundancy is available the error w.r.t the ground truth can be assumed to be low. On the other hand this also implies the major drawback: the hypothesis raised by this approach is only based on one or two values and thus is very dependent on the diversity of the measurement principles utilized.

2.3 Threshold Voter

As already indicated by the name the route taken here is to determine the most appropriate sensor readings considering their respective distances to each other. The above statement can be reformulated: those sensor pairs that have minimal distance to each other resemble the most reliable source of information. Hence the first step is to compute the pair wise distance between all available sensor readings $d_{ij} = s_i - s_j \; \forall i, j : i \neq j$. If this distance is below a defined threshold $d_{ij} \leq t$ both sensors shall be considered for the fusion process. Therefore, a weighting variable (w_i) is introduced for each sensor value. It contains the respective number of times the above statement is valid. Therefore, the fused output can be computed as $out = \sum_{i=0}^{n} w_i \cdot s_i / \sum_{i=0}^{n} w_i$. The advantage of this idea is that the resulting output is very robust to distortions of single outliers. The downside, on the other hand, is the static threshold concept that is only suited for dense sensor clusters with similar properties.

2.4 Kalman Filter

A Kalman Filter[2] is an optimal linear estimator (in a statistical sense) that is widely used in the field of sensor fusion due to its outstanding properties. It is composed of a two-cycle process: an update and a prediction phase. The prediction process uses a system model to derive a future system state based on the previous state and the inputs that the system is exposed to. This can be denoted as $\hat{x}_k^- = A \cdot \hat{x}_{k-1} + b \cdot u_{k-1}$ with the symbols given in table below. The adjacent correction phase makes use of a measurement model and the latest sensor reading to fuse the projected system state with the measured one. This can be formulated as $\hat{x}_k = \hat{x}_k^- + K_k \left(z_k - H \cdot \hat{x}_k^-\right)$. As one can already conclude exhaustive system and senor modeling is required. Besides this, two major conditions have to be fulfilled in order to ensure convergence of this approach. The first one is the linearity condition. It states that the predicted state is linear dependent on current state and Gaussian measurement noise. If this cannot be assumed a so called Extended Kalman Filter (*EKF*, see [9]) has to be employed. The second condition demands the independence of system and measurement noise between sampling times (i. e. no temporal correlation) and that the noise satisfies Gaussian distribution with zero mean.

A – System model	B – Input matrix (optional)
K – Kalman gain matrix	H – Measurement model
u – input (optional)	z – measurement

2.5 Fuzzy Voter

The first step of this approach is similar to the already discussed threshold voter: The distance (i. e. difference) of all pair wise permutations of available sensor data is computed. The interpretation this time however, is that they correspond with the degree of agreement (expressed by the fuzzy sets) of two sensors. The respective degrees of membership in each set are then inferred and defuzzyfied. This way a score is computed for each sensor that resembles the algorithm's confidence in the correctness of the respective sensor reading at a given time. Furthermore, it is used as a weighting factor for computing the fused output and the overall confidence in fusion accuracy.

An approach like this can be found in [4] wherein the usage in a brake by wire system is described. The algorithm presented there however, performed rather poorly in the simulation environment used for the experiments of this paper. This is for multiple reasons: An exponential growth of the number of fuzzy

[2] An excellent and more detailed introduction to Kalman filtering can be found in [6],[3] and [1]

rules (that have to be manually implemented) prohibits application in systems with high degree of redundancy. The number of rules can be denoted as an exponential function of the number of sensors n and the number of fuzzy sets m used to determine distance of sensor readings: $m^{n(n-1)/2}$. As can be seen even for a moderate degree of redundancy this would result in massive implementation overhead and poor computational performance. Therefore, Hoseinnezhad et al. suggested selecting a subset of rules instead of using all of them. In our case this resulted in a membership vector that was too sparse in order to allow for proper functionality of the algorithms. Thus, a new method of performing the inference was needed. It turned out that a closed form solution seems to be best suited for the applications considered in this paper.

The reason for this is that a closed form approach reduces manual labor in the implementation process to a minimum. Furthermore it also reduced the number of operations performed in the inference step. Another inconvenience that the authors experienced with Hoseinnezhad's approach is that for any given set of sensor readings some will be discarded because they have the greatest distance in the set. This is because they represent the worst source of information by definition of the algorithm. In case of highly accurate sensors, however, it does not make sense to discard a sensor with a measurement error of e. g. below 0.1 % if this is good enough for the application. Thus in the opinion of the authors the relative condition should be amended by a second, absolute one. This enables the user to set a tolerance for what can be considered a sufficient accuracy level in the respective application. Hence, a so called sensitivity threshold was implemented that can be altered during runtime if desired. In the experiment performed in this case study, however, it was set to be constant.

Hence, the modified algorithm can be described as follows. At first the distances for each pair of sensor readings at a given time is determined. In the next step the computed agreement is normalized based on the tolerance level defined by the user. By moving the lower bound of the low agreement fuzzy set the sensitivity of the agreement distribution can be defined. This allows for adjusting of the "dead band" width. A sensor reading difference within this band will result in a positive score and thus increases the influence of the respective sensor in the fused output. Once a distance is mapped to the available fuzzy sets, a membership vector for this distance can be determined. The next step is the so called defuzzyfication process. In order to project the found fuzzy values back into the "crisp" world a "score" is computed for each sensor that indicates its assumed reliability. This is done in the following way. The membership vector that represents the mapped distance for a pair of sensors is used to compute the score for both sensors that provided the measurements. A membership in high and medium agreement will result in a positive score contribution while membership in the low agreement set will decrease the score. The output is now computed as the (normalized) weighted sum of all sensor readings multiplied with their respective scores. The scores, however, can be used once more to extract the degree of certainty for the fused output. For this a metric was enforced that considers the value of all sensor scores. This approach seems intuitive since overall good sensor measurements will result in a high confidence and vice versa. Furthermore, the scores can also be used to determine a confidence level for each individual sensor since they express the distance of the sensor from the most likely correct value at a given time. In case of diverging sensors a fall-back strategy (weighted average with equal weights) was introduced. The necessity for such a mechanism, however, decreases with increasing degree of redundancy of measurement available in the system.

3 Experiments

In order to perform a systematic, in-depth evaluation of sensor fusion approaches a simulation environment is required. That is because this way one is capable of generating controlled and repeatable experiments as opposed to tests performed on real machines. Thus, a universal simulation environment was implemented using Matlab Simulink. The top level structure is presented in figure 1.

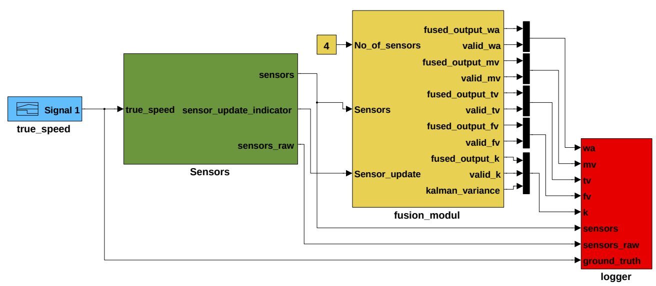


Fig. 1: Simulation environment implemented in Simulink

As can be seen it is composed of two main blocks containing the core functionality and several others used for auxiliary tasks like e. g. signal generation and post processing. On the very left of the image (blue block) the ground truth is fed into the system that represents the undistorted property to be monitored by the sensors. This signal is then passed on to the next block (dark green) that represents the signal processing (i. e. measurement) performed by the sensors. In the presented simulation framework the properties of a given sensor are described using 13 parameters representing e. g. bias, noise, failure modes, sample times and information encoding. In a second step the peripheral information preprocessing is emulated. This is done based on another 11 parameters for each simulated sensor that represent the DSP (Digital Signal Processor) properties (e. g. sample intervals, filtering and quantization effects). Thus, the ground truth is transformed into n different (erroneous) signals that reflect the characteristics of each sensor.

The generated signals are handed on to the central component of this framework: the block containing the fusion algorithms. Here the respective sensor signals are processed using five different signal level fusion methods (weighted average, median voter, threshold voter, fuzzy voter and a Kalman filter). The results (containing the fused output and a validity/confidence flag if available) are then stored in work space variables that can be analyzed later on. Before the actual experiments can be performed several tasks have to be performed: the sensors that shall be used have to be selected and modeled. Furthermore, a ground truth signal has to be defined that covers a variety of scenarios and is therefore representative for a typical application (see figure 2).

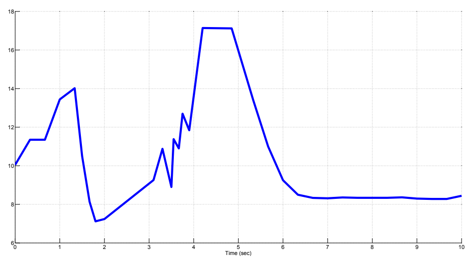


Fig. 2: Ground truth signal

Let us assume that physical property that has to be measured and fused is the vehicle's ground speed. This choice is motivated by the fact that there exists a variety of sensors with different measurement principles for this task. Other than that almost arbitrary properties would deliver the same results so this selection does not limit the degree of expressiveness of the results observed here.

As can be seen the signal can be roughly separated in three sections. The first section ($0 \leq t < 3sec$) represents a signal with modest dynamic ($a \leq 1\frac{m}{s^2}$) in it. The second section ($3 \leq t < 6sec$) illustrates a highly dynamic vehicle movement with about $\pm 2\frac{m}{s^2}$ and some oscillations. The dynamical character of the rest of the signal can be assumed to be low. Although a signal like this would be highly unlikely in such a short time span in reality it is never-the-less optimal for the simulated test since it covers a large amount of everyday situations in a short period of time.

As pointed out above the second decision prior to the experiments is the set of sensors to be used. For this purpose several devices were investigated under lab conditions to extract their characteristics with respect to the parameters that are needed to model them in the simulation environment. Inspired by the maximum degree of redundancy that one would be able to find in a common real application the number of sensors was limited to four. A diverse set of ground speed sensors was selected that is comprised of a high accuracy wheel encoder (low latency, low noise), the machine's tachometer signal computed from the gear speed (lower resolution, moderate noise), a ground speed radar (high noise, medium update rates) and a GPS based speed signal (high latency). A sample of all four simulated sensors for the ground speed signal described above is presented in figure 3.

Once the above modeling is completed one can start to pursue the actual experiments. In order to be able to draw meaningful conclusions with respect to

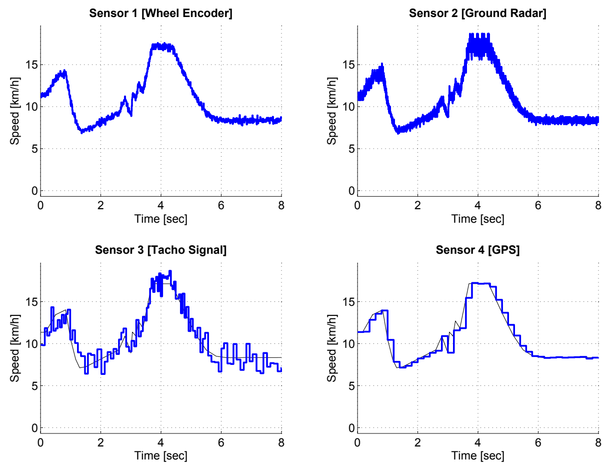


Fig. 3: Simulated sensor data

the application domain of commercial vehicles, selecting the proper test scenarios is of utmost importance. Typical challenges that arise in everyday operation are e. g. sensor drift, severe transient noise and sensor failure. It can be assumed that these classes cover a decent percentage of problems that engineers experience when dealing with redundant sensor fusion. Thus, they can serve as a performance measure in multiple respects. Besides these scenarios the performance of the respective algorithms under regular operation (i. e. the absence of the issues mentioned above) is presented and discussed in the following sections. For the sake of analyzing the results one has to concentrate the vast amounts of data produced during the experiments into a small set of numbers that capture the full essence of the results. In this paper we selected the absolute mean error (i. e. $|ground\_truth - fusion\_result|$), absolute maximum error as well as the statistical error distribution parameters $\mathcal{N}(\mu, \sigma)$ since they give the reader a rather good idea what the results look like without the need to provide the respective figure.

First of all let us take a look at the performance of the algorithms during regular operation. The results are illustrated in figure 4. Here the maximum deviation for each approach is marked in red while the mean deviation is colored in blue. Since the error distribution parameters are almost equal for all approaches ($0.2 < \sigma < 0.4$ and $\mu \approx 0$) they are not denoted in the figure.

As can be seen the Kalman filter delivers the best results in both cases followed by the fuzzy and median voter. The biggest maximum error was encountered with the threshold voter while the weighted average approach ($w = (0.7, 0.5, 0.4, 0.9) = const$) produced the largest mean deviation. However, one has to keep in mind that this scenario is optimally suited for the Kalman approach since all sensor errors are modeled as normally distributed (with perfect knowledge of their distribution parameters) and the algorithms parameters were tuned to this scenario using a variance estimator. Thus, the Kalman Filter should rather be considered as a reference implementation of an optimal estimator in

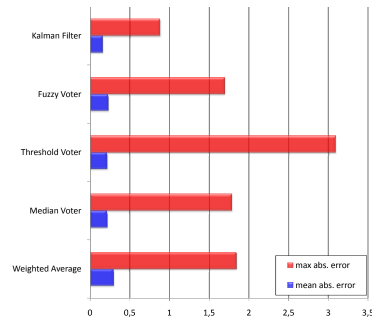


Fig. 4: Fusion result under normal conditions

this case. In general it can be seen that mean error of all voting based approaches is approximately equal. The descending ranking for this experiment for the remaining approaches would be fuzzy-, median- and threshold voter with the weighted average being close fifth due to its slightly higher mean deviation.

The rest of the experiments section is devoted to the scenarios that test the behavior of the investigated approaches under distortion in form of sensor signal degrading or even failure. For this purpose seven scenarios were created that belong to the challenge classes discussed above. Sensor drift was dealt with in scenarios 1,2 and 5. In scenario 1 a miscalibrated[3] ground radar sensor was simulated. Experiment 5 deals with a constant additive drift of 15 km/h in the wheel encoder sensor while in scenario 2 the drift is linear increasing from 0 to 20 km/h over the simulation time. Scenarios 3 and 4 focus on signal loss due to e. g. malfunction. In the first one the GPS speed signal (low update rate) is lost while in the second one the same happens to a sensor with high update rate. The last range of issues dealt with here is noise. In scenario 6 noise in from of slip is simultaneously added to both wheel bound sensor while in the final experiment excessive noise of the ground speed radar is simulated. The results are presented in table 1 as well as figure 5.

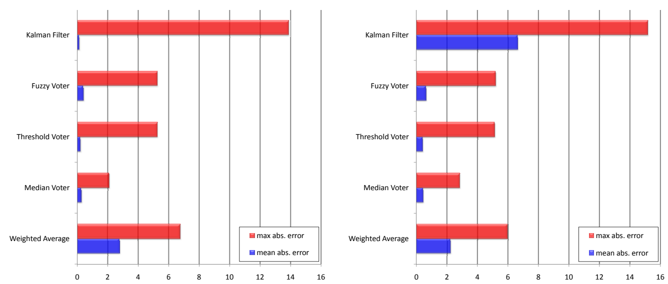
Table 1: Result of test scenarios

Scenario	1	2	3	4	5	6	7
Weighted Average	×	×	×	×	(×)	✓	(×)
Median Voter	(✓)	✓	✓	✓	✓	✓	✓
Threshold Voter	(✓)	✓	✓	✓	✓	✓	✓
Fuzzy Voter	(✓)	✓	✓	✓	✓	✓	✓
Kalman Filter	✓	(✓)	✓	×	×	×	✓

In the table each algorithm is assigned an evaluation for each experiment. A check mark indicates that it performs well in the scenario while a cross implies failure. Braces are used to express tendencies and should be interpreted as ”rather passed” or ”rather failed”. The results for the two scenarios with failing sensors are depicted separately in figure 5. This it because it is rather surprising

[3] sensor delivers mph speed rather than the km/h assumed by the algorithms

to see the degree of influence of the sensor's update rate. While the Kalman filter shows the by far smallest mean error in case of a sensor failure of the GPS (low update rate) it delivers the worst results in case of a sensor failure with high update rates. This can be explained with the static models that are used in the Kalman approach implemented here. The remaining algorithms seem to be rather not influenced significantly by the properties of the failing sensor.



(a) GPS fails (b) wheel encoder fails

Fig. 5: Deviation of fusion output in case of sensor failure

3.1 Evaluation

The experiments have shown that the most intuitive approach to fuse redundant sensor data (i. e. weighted average) is in most cases too simple to deliver the expected performance. Its main issue is that it lacks robustness to distortion and signal degrading.

Kalman will only work properly if sensor system properties (e. g. variance etc.) are previously known and certain requirements (see section 2.4) are met. Thus, this approach is not suited to integrate the sensing capabilities available on collaborating machines or other external sensor units like e. g. implements. Furthermore, it fails in certain scenarios as shown in the experiments section. Thus, it can be stated that a Kalman filter will deliver superior performance for most standard applications but will fail in case of certain distortions like e. g. sensor failure. Therefore, it should only be used in systems where the sensor configuration is well investigated and does not change over time. Besides that the sensor functionality has to be assured by means of constant monitoring.

Voting algorithms have proven to be a good compromise between dependable fusion performance and robustness. The median voter approach delivers good results in scenarios with limited signal degrading and failure since the faulty information will not be incorporated into the fusion result. This selectivity is also its biggest weakness since this means that the fusion hypothesis is only based on one or two sensors and thus volatile to severe degrading. The remaining candidates are the (crisp) threshold voter and the fuzzy approach. Both base on the same principle but use different representations. Their performance is almost equally good. Due to its additional confidence information for both sensors and output and adjustability, the fuzzy voter seems to be the best candidate for versatile and robust fusion systems for commercial vehicles. This is especially true in case of dynamically changing systems.

4 Conclusion and Outlook

In this paper commonly used signal level sensor fusion approaches were introduced and their respective performance was evaluated in several application scenarios that form a challenge in the everyday usage of such algorithms. The experiments have underlined the advantages and shortcoming of each approach that have to be considered when designing a fusion system for a commercial vehicle. It became obvious that a tradeoff between fusion accuracy and robustness exists. The fuzzy voter has proven to be an interesting candidate for further investigation since it represents a good compromise of both worlds. Thus, its properties should be examined more closely and its performance in a real application shall be tested. Furthermore the high degree of adaptability offers room for improvement beyond the already experienced behavior.

Acknowledgments

This work was partly funded by the German Federal Ministry of Education and Research (BMBF) in the context of the project iGreen (No.: 01 IA08005 P).

References

1. R.G. Brown and P.Y.C. Hwang. *Introduction to Random Signals and Applied Kalman Filtering*. John Wiley and Sons, Inc., 1997.
2. B.V. Dasarathy. Sensor fusion potential exploitation-innovative architectures and illustrative applications. In *Proceedings of the IEEE*, volume 85, pages pp. 24–38, Huntsville (AL), USA, January 1997.
3. M.S. Grewal and A.P. Andrews. *Kalman Filtering - Theory and Practice Using MATLAB*. Wiley and Sons, Inc., New York, third edition, 2008.
4. R. Hoseinnezhad and A. Bab-Hadiashar. Fusion of redundant information in brake-by-wire systems using a fuzzy voter. *Journal of Advances in Information Fusion*, 1(1):pp. 52–62, July 2006.
5. R.C. Luo, Y.C. Chou, and O. Chen. Multisensor fusion and intergation: Algorithms, applications, and future research directions. In *Proceeding to the International Conference on Mechatronics and Automation*, pages pp. 1986–1991, Habin, China, August 2007.
6. P. Maybeck. *Stochastic Models, Estimation and Control*. Academic Press, 1979.
7. H.B. Mitchell. *Multi-Sensor Data Fusion: An Introduction*. Springer, first edition, September 2007.
8. B. Parhami. Voting algorithms. *IEEE Transactions on Reliability*, 43(4):pp. 617–629, December 1994.
9. G. Welch and G. Bishop. *An Introduction to the Kalman Filter*. University of North Carolina Chapel Hill, July 2006.