A Fuzzy Approach to Low Level Sensor Fusion with Limited System Knowledge

Sebastian Blank
Robotics Research Lab
Dep. of Computer Sciences
University of Kaiserslautern
Kaiserslautern, Germany
blank@cs.uni-kl.de

Tobias Föhst
Robotics Research Lab
Dep. of Computer Sciences
University of Kaiserslautern
Kaiserslautern, Germany
foehst@cs.uni-kl.de

Karsten Berns
Robotics Research Lab
Dep. of Computer Sciences
University of Kaiserslautern
Kaiserslautern, Germany
berns@cs.uni-kl.de

Abstract – *When facing the need to perform low-level sensor fusion with only very limited knowledge available one has to come up with an alternative to the well known and proven Kalman filter. A very interesting candidate for such applications is the so called fuzzy (or soft) voter. This algorithm makes use of fuzzy logic principles to fuse signals in an efficient way and provides a figure of merit as well as sensor monitoring capabilities with very moderate demand for computation performance and memory. In this paper a computational efficient alternative implementation of soft voting for embedded applications is described. Furthermore, its performance is examined using scenarios typical to harsh operations environments using simulation.*

Keywords: robust sensor fusion, fuzzy logic, voting, limited system knowledge, computational efficient

1 Introduction

The most common approach chosen to master low level signal fusion is the Kalman filter[1]. It is very popular and commonly used for a very broad range of purposes like e. g. localization [10], street traffic modeling [5], sensor calibration [2], or for economic processes [9].

This is because Kalman filtering allows for superior fusion performance and has the ability to estimate (i. e. overcome periods of insufficient measurement by means of forward projection of the last valid state). The price for such outstanding properties, however, is the need for extensive knowledge of the system's properties and characteristics. Among the parameters that have to be acquired prior to being able to use the filter is the system and measurement model that expresses the relation among the system inherent states to the measured quantities and the respective noise. Furthermore, a few assumptions concerning the correlation of the process noise over time and its statistical distribution have to be assured in order to be able to derive a functional system.

In many cases information of such level of detail is too costly to obtain for the intended application or cannot be acquired at all. This is especially true when sensor fusion in dynamical sensor networks (like e. g. Zig Bee, Bluetooth, or IEEE 802.11) or information exchange among collaborating machines of varying manufacturers are concerned. Here, not necessarily all devices that are able to provide additional information are previously known (see [3]). Therefore, they cannot be analyzed and modeled in advance as would be required for the Kalman filter.

Thus, an alternative approach that requires very little or even no knowledge about the system components and their collaboration has to be found. Besides the ability to function with very limited system knowledge there are several other demands when dealing with low level multi-sensor fusion. Among these are high robustness, computational efficiency, a high degree of accuracy and ease of use just to name a few.

Commonly used methods in this field are the weighted average method, various voting algorithms based on crisp numbers (like e. g. threshold and median voter, see [8]) and the already mentioned Kalman filter as well as other more elaborate estimation and inference methods. Another quite popular method in this field is applied in fuzzy techniques, that are utilized in different applications[2] including sensor fusion. They share the common ability to express vagueness and thus are not limited to an "either or" decision but allow for gradual decision making. A well known candidate for fuzzy low-level sensor fusion is the so called soft voter.

Since we mainly focus on embedded applications computational efficiency is a major concern. Therefore, techniques that have excessive demands with respect to computational performance and memory are not suited for this scenario since one has to be able to perform data fusion in real time with very limited hardware resources. As demonstrated in previous work by the authors, the fuzzy voter is a promising candidate that meets all of the above requirements. Based on a soft voting approach developed by Hoseinnezhad et al. a method optimized for use in embedded applications like e.g. vehicles is derived in section 2 of this paper. In section 3 the performance of the developed algorithm is investigated in various experiments concerning the fusion accuracy

[1] see [7] and [4] for a detailed introduction to Kalman filtering

[2] like e. g. classification, feedback control, and clustering

in the presence of sensor signal degrading and malfunction. The last section gives a resume concerning the experienced characteristics of the approach and provides and outlook on future research related to this paper.

2 Fuzzy Voting Approach

A fuzzy voting approach intended for low level signal fusion was proposed by Hoseinnezhad et al. The algorithm and its application performance in a break-by-wire system is presented in [6]. The main steps of the method are outlined in the next paragraph.

At first the distance for each pair of sensor readings (s_i, s_j) at a given time is computed as $d_{ij} = |s_i - s_j|$. These distance values are subsequently mapped to fuzzy sets that represent the sensor agreement as a function of absolute distance. Based on the resulting membership vector the results of the fuzzy rules used for the inference step are computed. They consist of a contribution to the output and sensor faultiness values that are mapped to fuzzy sets once more to receive a confidence measure for each sensor. This time a centroid norm is employed to perform inference. The fused output can then be determined as the weighted sum of all rule outputs with the degree of rule fulfillment as weighting factor.

Even though the basic idea sounds rather interesting the authors experienced several drawbacks in the process of using this approach for fusion in highly redundant embedded applications. The first concern is the characteristics of the approach with an increasing number of sensors. One aspect of this is the need for a second instance of fuzzy for the sensor confidence and the exponential growth of number of rules. One way of dealing with this is to select only a subset as suggested by Hoseinnezhad et al. In case of our application environment, this resulted in a insufficiently small number of rules that applied. Thus, the algorithm was no longer able to deliver the expected performance. The complexity of the approach designed for the break-by-wire system is $\mathcal{O}(n^2 \cdot m + m^2 + l)$ for n sensor, m sensor agreement fuzzy sets, and l sensor faultiness sets. In case of the mentioned application these numbers are given as $n = m = l = 3$. Since the number of sensors providing redundant measurements is quite limited for the predominant number of real applications[3] none of these factors can be neglected.

As the reader can see in the next section, the complexity of the approach suggested in this paper can be summarized as $\mathcal{O}(n^2 \cdot m)$ and is thus significantly smaller for all practical cases.

2.1 A Revised Approach to Soft Voting

As outlined in the preceding paragraph the initial approach does not seem to be suited for the kind of applications that are considered in this paper. Thus, a new way of performing fuzzy voting has to be found with better scaling characteristics concerning the number of sensors and sets used. The provided complexity notion, however, reveals to be only part of the problem since not just the complexity class but also the factors matter for typical numbers of sensors employed. One key element to increase the performance seems to be the avoidance of the classical fuzzy rules used for inference. Therefore, we suggest to compute so called *sensor scores* that can be utilized as weighting factors directly instead of computing several rule outputs and combining them later on since this signifies a severe overhead. Furthermore, the process of generating a sensor confidence can be simplified by making use of the scores instead of a second group of fuzzy sets that have to be inferred once more. Besides that, an approach like this allows to add an overall fusion confidence at no additional costs. A scheme that illustrated the proposed approach is given in figure 1.

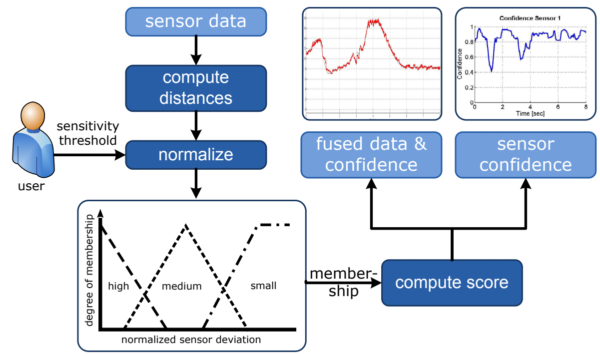


Figure 1: Proposed fuzzy voting approach

As before the first step in the process of fuzzy voting is computation of sensor measurement distance. Afterwards the deviation is normalized prior to being mapped to the fuzzy sets representing the sensor agreement. This step is introduced to allow the user to express a proper definition of sufficient accuracy for the given application in form of a single parameter. Once this is done the membership vector $\mu_x(d_{ij})$ can be computed for the agreement sets. Latter express the degree of agreement in the sensor readings in a closed linear form. Therefore, it is possible to compute the membership values $\mu_{set}(d_{ij})$ very efficiently. The specific shapes used in the experiments following this section are presented in figure 2.

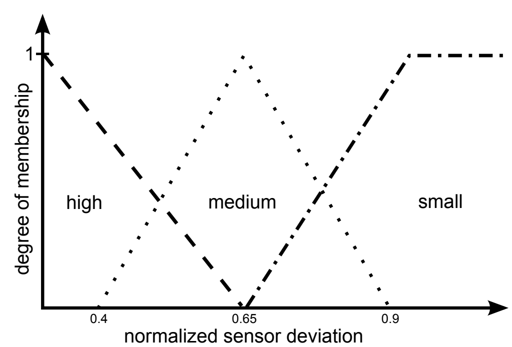


Figure 2: Fuzzy sets

The lower border of the high agreement set is defined to

[3] A good estimate of the upper limit would be five or six.

a value of 0.65 on a normalized scale. As can be seen the medium set (0.4 to 0.9) is symmetric with a peak at 0.65. The small agreement set starts at 0.65 and reaches a membership factor of 1 at 0.9.

Instead of making use of a fuzzy rule set for the inference step sensor scores are computed. The contribution to the score of the sensor i and j that supplied the measurements the distance d_{ij} is based on, can be denoted as $\mu_{high}(d_{ij}) + \mu_{med}(d_{ij}) - \mu_{low}(d_{ij})$. As one can see a membership in both the high and medium set will result in a positive contribution while a membership in the small agreement set will decrease the sensor score. In case a sensor achieves a negative score it is normalized to zero and thus represents an insufficient sensor performance. Once all distances are processed in this way the resulting scores can be utilized for several purposes. At first they represent the authority of a given sensor w. r. t. the fused output as denoted in (1).

$$out = \begin{cases} \sum_{i=0}^{n} s_i \cdot \text{score}_i / \sum_{i=0}^{n} \text{score}_i & \textit{if } \sum_{i=0}^{n} \text{score}_i > 0 \\ \sum_{i=0}^{n} s_i / ||s_i|| & \textit{else} \end{cases} \tag{1}$$

In case no sensor receives a score greater than zero (i. e. fulfills the previously set accuracy demand) a fall-back strategy is introduced in form of an equal weight fusion (with $||s_i||$ being the number of sensors) to ensure valid output even in case of sensor malfunction or too small degree of redundancy (≤ 3). Even though the achievable degree of accuracy is significantly lower than the one of other approaches weighted average has no limitations or conditions that restrict its application and can thus deliver valid fused outputs that are somewhat close to the ground truth even in the worst of circumstances. The necessity for such a mechanism however, decreases with increasing degree of redundancy of measurement available in the system.

A second use for the scores is the determination of a sensor confidence indicator. Since a normalized score is computed for each sensor they can be directly employed for this purpose without further notification. Combining those scores furthermore allows for generating a source of information for the user concerning the algorithm status. It can be employed as a kind of system health indicator that reveals internal changes like e. g. the need to switch to the fall-back strategy and thus allows for the necessary actions to be taken. The definition of the algorithm confidence is given in (2).

$$\textit{confidence} = \begin{cases} \sum_{i=0}^{n} \text{score} / n & \textit{if } \sum_{i=0}^{n} \text{score}_i > 0 \\ 0 & \textit{else} \end{cases} \tag{2}$$

As can be seen the algorithm will emit a positive value in between 0 and 1 that indicates the current overall confidence level in the sensors and thus gives an idea of the assumed fusion quality that can be currently provided.

2.2 Self-Adapting Sensitivity Threshold

As we will see in the experiment section finding an optimal setting for the sensitivity value is a rather difficult task. This is because a single number is unable to accommodate for the dynamic changes during the fusion process. No matter which value is selected it will lead to significantly increased deviations during some periods of the process due to a suboptimal setup.

Therefore, the introduction of an adaption scheme seems reasonable. The key idea is to be flexible enough to rapidly adapt to changing sensing quality in an automated way. For this purpose one needs a parameter to monitor the fusion process. This, however, is already available in the presented fuzzy voter as discussed earlier in form of the fusion confidence.

Therefore, a rather simple adaptation feedback control-like strategy is proposed here:

if average_confidence ≤ 0.1 **then**
  sensitivity_scaling += 0.05
end if
if average_confidence > 0.75 **then**
  sensitivity_scaling *= 0.95
end if

Besides minimizing the error (as shown in section 3.2) this strategy will also lead to keeping the sensor and fusion confidence value in the middle of its spectrum of [0,1]. As an added benefit this allows for maximization of the expressiveness of the respective values since saturation effects[4] can be avoided in this way.

In order to minimize the influence of outliers (i. e. changing the sensitivity threshold due to only a few consecutive poor measurements) averaging of the fusion confidence by means of a moving average filter seems to be indicated. Once more we face a trade-off: minimizing the number of false-positives vs. delay in adjustment. In the experiments conducted in this paper a window size of 50 measurements resulted in rather good results.

2.3 Efficient encoding

Since computational performance and memory are usually quite limited in embedded applications an efficient way of representing and accessing the information used during the algorithm presented in the sections above is needed.

Thus, let us take a look at the first step of the algorithm: the sensor distance determination. The most convenient form of storing the incoming data is an upper triangular distance matrix D as presented in 3.

$$D = \begin{pmatrix} d_{12} & \dots & d_{1n} \\ \vdots & \iddots & \\ d_{(n-1)n} & & \mathbf{0} \end{pmatrix} \tag{3}$$

[4]Signals close to the extreme have very limited dynamic range due to the boundedness of the signal in between 0 and 1.

Since the order does not matter for the absolute distance $d_{12} = d_{21}$ the full matrix can be compressed into a $(n-1) \times (n-1)$ upper triangular matrix for n sensors to be processed. Once all sensor data is received for one algorithm step they can be normalized and stored back into the matrix as shown before. The next step in the algorithm is the assignment of the distances to their respective membership w. r. t. the fuzzy sets. Once more the upper triangular matrix can be utilized. This time, however, $m \times 1$ vectors (m is the number of fuzzy sets used) are stored instead of scalars and it is now called M. In this paper three sets were used. Thus, the membership vectors can be denoted as

$$\vec{m_{ij}} = \begin{pmatrix} \mu_{high}(d_{ij}) \\ \mu_{med}(d_{ij}) \\ \mu_{high}(d_{ij}) \end{pmatrix} \quad (4)$$

As opposed the original approach we do not rely on fuzzy rules for the inference step but rather apply scalars to represent the expressiveness of a sensor reading. Therefore, the membership vectors can be simplified at this point. In order to relate the summed up membership vectors to the respective sensor it is important to find a unique way to determine the proper assignment based on the indices in the matrix to allow for efficient computation. The value at row k and in column l can be traced back to a pair of sensors (i, j) according to 5.

$$(k, l) \rightarrow i = k, j = l + k \quad (5)$$

Thus, the earlier introduced score can be efficiently computed by iterating over the matrix:

for $k = 1 : (n-1)$ **do**
  $score_i + = m_{kl}$
  for $l = 1 : (n-1)$ **do**
    $score_j + = m_{kl}$
  end for
end for

Now the remaining algorithm steps can be performed and the fused output as well as the sensor and fusion confidence are computed.

3 Experiments

In order to investigate the performance of the proposed fuzzy voting approach in a controlled environment a simulation environment like Simulink[5] seems to be a good choice.

Thus, a simulation environment was implemented that allows the modeling of sensor behavior and fusion algorithm performance. The idea behind is to feed a signal that represents the ground truth into the system. This signal is handed on to a number of Simulink blocks that modify it in order to emulate the process of a sensor measurement in the real world. Those processes are performed in parallel for each simulated sensor. These signals are then passed through the fusion algorithm. The resulting output as well as the other simulation parameters are recorded and the analyzed off-line after the simulation run is completed[6].

For the experiments in this next section a typical fusion scenario in the field of commercial vehicles was selected. The fusion task here focuses on a vehicle's ground speed. Thus, a ground truth signal was chosen that represents typical scenarios that one could find in the daily operation. The signal includes parts with low, medium ($\pm 1 m/s^2$), and high ($\pm 2.5 m/s^2$) dynamical changes as well as static periods and oscillation patterns. Four physical sensor were selected to be modeled into the environment. This resembles a degree of redundancy that can be assumed to be available in the real application. In order to capture as much of the diversity that one would be able to find in physical systems as possible a diverse set of sensors was selected for the simulation. It consists of a wheel encoder, a tacho sensor, a microwave ground speed radar, and a high accuracy GPS receiver with a coupled inertial measurement unit. To capture as much of the characteristics as possible the actual sensor was investigated under lab conditions and the discovered properties were captured in the model parameters of each sensor block.

Now that the environment is presented the performance of the fuzzy voting approach shall be investigated. The key parameter of this approach concerning the fusion accuracy is the already introduced sensitivity threshold. As will be underlined with the following experiments a too optimistic (deviation is underestimated, i. e. threshold too small) setup will result in a high percentage of usage of the fall-back strategy and thus reduced accuracy. Too pessimistic estimation on the other hand will cause the algorithm to include too many below average accuracy measurements in the fused output even though more selectivity could be applied that would result in significantly better results.

Hence an initial experiment was conducted to investigate the optimal setup for the sensitivity parameter. The results are presented in figure 3.

3.1 Fixed Sensitivity Threshold

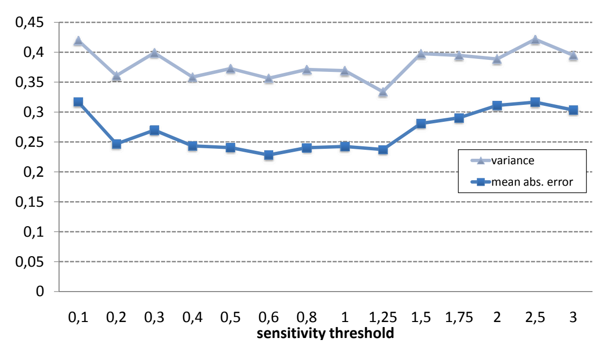


Figure 3: Mean absolute error and error variance for varying sensitivity threshold settings under normal conditions

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

[6]In order to keep this section brief only a very rough overview is provided here. For more details about the simulation environment and the sensor modeling see [1]

For the given scenario of vehicle ground speed fusion in a range of 0 to approximately 25 km/h both the means absolute deviation with respect to the ground truth and its variance for various sensitivity thresholds settings are denoted in the diagram. Each measurement resembles a dataset of over 40,000 samples. The data underlines the already discussed effect of over and under estimation. As can be seen an optimal setup can be found for threshold in between 0.4 and 1.25 for normal operating conditions (i. e. the absence of sensor degrading or malfunction).

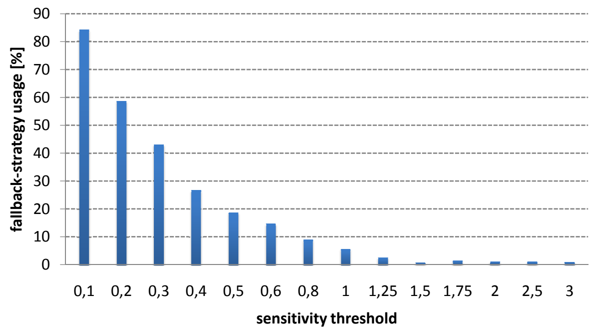


Figure 4: Percentage of usage of the fall-back strategy for multiple sensitivity threshold setups

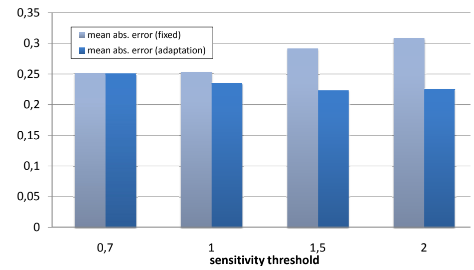
As can be seen in figure 4 the utilization percentage of the fall-back strategy can be assumed to be exponential. Thus, setting smaller than 0.4 will cause the algorithm to make use of the fall-back strategy too often. As shown above, this will result in result a 50 % larger deviation than what could be achieved with an optimal setup. A value greater than 1.25 will on the one side reduce fall-back percentage to less than 1% but at the same time this gain in accuracy is entirely lost to the poor measurements that are included due to the overestimation of the sensor distance.

The experiments have shown that finding the optimal parameters is tedious and only valid as long as the assumptions concerning the sensor signal quality are not violated. Thus, it is obvious that a dynamic way of adapting to the currently available signals is needed. The strategy proposed in section 2.2 was applied to the previously used experimental setup.

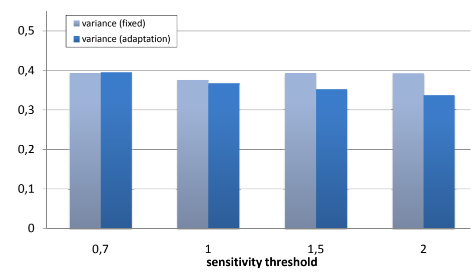
3.2 Self-Adapting Sensitivity Threshold

In this section the effectiveness of a self-adapting sensitivity threshold shall be examined. Before the experiments can be conducted the adaptation range needs to be determined. Based on the results of the previous section (see figure 3) the lower limit for the sensitivity threshold was selected to be 0.5 since this signifies the lower limit of the optimal fixed parameter setting. The upper limit, however, was not so easy to come up with. In a series of preliminary tests a threshold of 2.0 has proven to be sufficient even in case of signal degrading or sensor malfunction (as we will see later on in this section). The initial threshold provided to the algorithm is 1.5 for all simulations performed here. This is because this setting allows for fast adaption in either way.

In order to evaluate the effect of self-adaptation several experiments were conducted. At first the mean absolute deviation and its variance were analyzed for normal sensor signals. The results are depicted in figure 5.



(a) abs. mean error



(b) error variance

Figure 5: Evaluation of fusion deviation with fixed and self-adapting sensitivity thresholds

As can be seen, the results express the already assumed behavior. Figure 5a reveals an obvious trend: While the error increases with higher sensitivity settings in case of fixed threshold it decreases at the same time in case of a self-adapting threshold. The same trend can be experienced when taking a look at the distribution parameter of the deviation. While the variance remains more or less constant for fixed thresholds it decreases for adapting thresholds.

Now that the impact of adaptation on scenarios with regular sensor data is discussed let us turn our attention to situations where sensor biasing or failure occurs. For this purpose four scenarios for biased/degrading data and two covering sensor failure are investigated in the following paragraph.

At first, let us take a look at the scenarios dealing with degrading sensor signals that result in biased data. In table 1 the percental improvement of the fusion deviation $\overline{\Delta}$, as well as its statistical parameters (μ, σ) are provided. Furthermore, the deviation of the average employed threshold setting $(\overline{\theta})$ of the adaptive approach relative to the fusion algorithm with fixed settings can be seen.

Even though not all possible scenarios can be emulated here some typical candidates were selected to investigate the fusion performance while degrading sensor signals occur. Constant and variable offsets (scenario 1 and 2), as well as excessive signal noise and simulated wheel slip (scenarios 3 and 4) were simulated. As can be seen the improvement due to self-adjusting sensitivity is significant. Furthermore, it can be seen that the threshold is adjusted in both ways. In

Table 1: Increase in accuracy for corrupted sensor data

	01	02	03	04
$\overline{\Delta}$	-12.77 %	-7.71 %	-7.62 %	-19.11 %
σ	-30.84 %	-13.54 %	-3.62 %	-19.59 %
μ	-10.05 %	-6.95 %	-3.03 %	-13.70 %
$\overline{\theta}$	+ 2.56 %	-3.52 %	-54.48 %	+7.92 %

case of constant offset and slip the average threshold is increased to keep the rejection rate low enough to maintain a sufficient degree of redundancy. In case of noise the threshold is kept close to the minimal allowed threshold since the algorithm can be more selective about rejecting of measurements. A similar increase in performance can be observed in case of sensor malfunction. The results for GPS and wheel encoder failure are presented in table 2.

Table 2: Deviation reduction due to adaptation for simulated sensor failure

	01	02
$\overline{\Delta}$	-11.79 %	-14.59 %
σ	-34.34 %	-45.68 %
μ	-11.52 %	-12.63 %
$\overline{\theta}$	- 13.4 %	-14.13 %

Again, the accuracy can be increased by means of adapting the sensitivity to the assumed currently available sensing quality. The dynamic character of the adaption scheme can be best seen in a plot of the effective threshold over the simulation time. Such a plot is provided in figure 6.

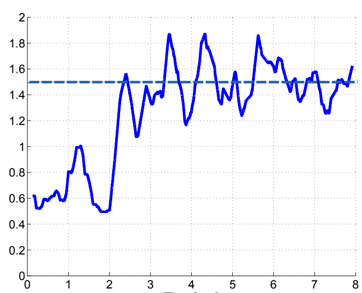


Figure 6: Averaged sensitivity threshold (window size 0,01s) for sensor failure at $t = 2s$ and a mean sensitivity setting of 1.5

As can be seen the threshold level is lowered close to the minimal allowed setting during the period with fully functional sensors ($t \leq 2s$). As soon as the measurements of one senor are lost the sensitivity is increased to maintain in sufficient level of redundancy for the fusion algorithm. Still, it is kept below the initial setting of 1.5 and thus keeps the accuracy at the best possible level.

The end of this section lets us assess the overall performance of the approach presented in this paper. As shown in our previous work (see [1]) it delivers equal or even better fusion results (with respect to the deviation of the fused output from the ground truth) than other popular low level fusion approaches like e. g. median voting, weighted average approaches, or Kalman filters and is thus well suited for fusion tasks in embedded systems. Therefore, the scope of the final experiment is to relate the performance w. r. t. the original fuzzy voting approach perused by Hoseinnezhad et al. For this purpose a series of experiments with various ground truth signals was performed and the respective deviation in from of the mean squared error extracted.

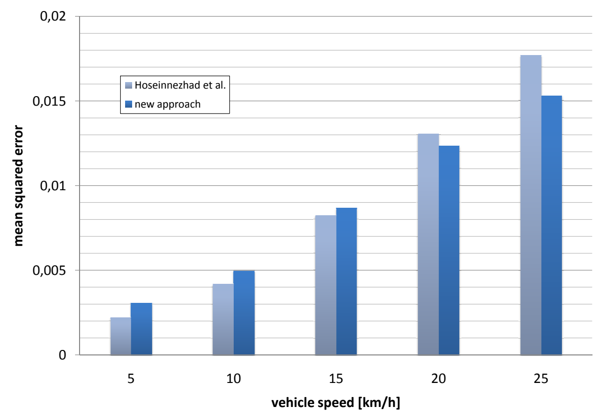


Figure 7: Divination comparison between fuzzy voting approach perused by Hoseinnezhad et al. and the approach presented in this paper

As can be seen in figure 7 it can be stated that the overall performance of both approaches is quite similar even though the new approach does not utilize classic fuzzy inference and is thus more computational efficient. The scenario selected here are five constant speeds (between 5 and 25 km/h) that are measured with four sensors. It can be seen that the new approach outperforms the old one for inputs greater than approximately $12km/h$. Equal results could be observed when other ground truth signals were employed. In case of a ramp function (amplitude $\in [10, 20]$ km/h) the new approach performed about 7 % better. The same improvement ($\approx 8.2\%$) could be seen in an exponential function input in the range of 0 to 20 km/h. A sinus pattern (amplitude between 13 and 17 km/h) yield an even greater improvement of $\approx 25\%$. Thus, it can be stated that the new approach can be assumed to be superior to the original one while requiring less computational effort and memory. Therefore, it seems well suited for the intended use in embedded systems.

4 Conclusion and Outlook

In summary it can be stated that the modified fuzzy voting approach proposed in this paper is well suited for low level data fusion. An accuracy of the fused output w. r. t. the ground truth of $\leq 1\%$ can be achieved during normal operation. Even a failure in one out of the four available sensors will only reduce the accuracy to a level of $\leq 2.5\%$. Furthermore, it could be observed that two (out of four) severely corrupted sensor signals still ensure a deviation smaller than

5 %. In conclusion the results of the experiments underline that fuzzy voting offers both good accuracy and a high degree of robustness. The introduced sensitivity adaptation scheme has proven to be able to improve the degree of accuracy in standard environment and especially in case of sensor signal degrading and failure. Other than that, the algorithm provides a measure of fusion confidence that allows decisions to be not only based on the fused output but also on the assumed validity of these values. Besides that, a confidence value for each sensor is provided that can be utilized for sensor monitoring. In conclusion it can be stated that such an approach is suited for applications in automation that feature dynamical changes in the system configuration caused by e. g. failure, addition, or exchange of sensors at runtime. Thus, it can be stated that the new method delivers slightly better results than the original approach taken by Hoseinnezhad et al. while significantly reducing the computational overhead and flexibility.

The next step is to test the algorithm's performance in a field test on an actual vehicle. Since a ground truth signal (that serves as a reference for the sensor data) is no longer available an experimental setup has to be found that allows for deduction of dependable and very accurate results. Another challenging aspect is the need for synchronization between different sensing sources. Besides the sheer fusion performance other aspects like minimum sustainable loop time, memory requirements, and robustness shall be investigated.

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] S. Blank, T. Föhst, and K. Berns. A case study towards evaluation of redundant multi-sensor data fusion. In K. Berns, C. Schindler, K. Dreßler, B. Jörg, R. Kalmar, and J. Hirth, editors, *Commercial Vehicle Technology 2010 Proceedings of the 1st Commercial Vehicle Technology Symposium (CVT 2010)*, Kaiserslautern, Germany, March 2010. Shaker Verlag.

[2] V. Cevher and J. H. McClellan. Sensor array calibration via tracking with the extended kalman filter. In *ICASSP 01 Proceedings of the Acoustics, Speech, and Signal Processing*, pages pp. 2817–2820, Washington, DC, USA, 2001. IEEE Computer Society.

[3] D. Estrin, R. Govindan, J.Heidemann, and S. Kumar. Next century challenges: Scalable coordination in sensor networks. In *Proceedings of the ACM/IEEE International Conference on Mobile Computing and Networking*, pages pp. 263–270, Seattle, WA, USA, August 1999. ACM.

[4] M.S. Grewal and A.P. Andrews. *Kalman Filtering - Theory and Practice Using MATLAB*. Wiley and Sons, Inc., New York, third edition, 2008.

[5] G.J. Grindey, S.M. Amin, E.Y. Rodin, and A. Garcia-Ortiz. Kalman filter approach to traffic modeling and prediction. In *Intelligent Transportation Systems*, volume 3207, pages pp. 234–240, Pittsburgh, PA, USA, October 1997. SPIE.

[6] 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.

[7] P. Maybeck. *Stochastic Models, Estimation and Control*. Academic Press, 1979.

[8] B. Parhami. Voting algorithms. *IEEE Transactions on Reliability*, 43(4):pp. 617–629, December 1994.

[9] G.K. Pasricha. Kalman filter and its economic applications. Technical report, University of California, Santa Cruz, CA, USA, October 2006.

[10] N. Schmitz, J. Koch, M. Proetzsch, and K. Berns. Fault-tolerant 3D localization for outdoor vehicles. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 941–946, Beijing, China, October 9-15 2006.