Fusion of Heterogeneous Sensors Data

Wilfried Elmenreich
Mobile Systems Group
Institute of Networked and Embedded Systems
University of Klagenfurt
Austria
wilfried.elmenreich@uni-klu.ac.at

Robert Leidenfrost
Vienna University of Technology
Austria
robert.leidenfrost@gmail.com

Abstract

A configuration with heterogeneous sensors using different measurement approaches most likely overcome the problem of correlated measurement errors as they occur when employing a set of homogeneous sensors that suffer from the same problems.

A heterogeneous approach requires sensor fusion algorithms that take the different uncertainties of the sensors into account. In this paper we elaborate two sensor fusion methods for this task. The first algorithm uses the estimated variance of each sensor measurement in order to find the optimal averaging weights. The second algorithm considers the covariances and thus provides a more sophisticated model at the cost of higher complexity in implementation and computation.

1 Introduction

Critical components are often replicated in systems in order to increase the dependability of the overall system. From the view of the systems engineer, using replicated components of the same type are preferred in order to ease comparison of results, faulty sensor diagnosis, and maintenance.

However, when considering sensor sources, the combined results from a set of homogeneous sensors often suffers from the same problems as a single sensor. For example, an infrared distance sensor that is susceptible to ambient light will eventually reproduce this error behavior in all sensors of the same type.

To overcome this problem of correlated sensor failures, it is advantageous to employ heterogeneous sensors with mutual independent error behavior. A heterogenous approach, however, requires sensor fusion algorithms that take the different uncertainties of the sensors into account. Uncertainty may not only vary by sensor type, but also with measurement value (e. g., the Sharp GP2D12 distance sensor has a problem measuring objects near the value of its minimum detection range that is around 10 cm).

It is the objective of this paper to present a sensor fusion approach that supports the fusion of data from heterogeneous sensors with uncertainty functions that change dynamically with the measurement range. The method is quite generic but in order to show an example of its use, we discuss it in a time-triggered system architecture. Therefore, we present an architecture for distributed sensing and data communication and a sensor fusion

algorithm that combines measurements from heterogeneous sensors with respect to their current measurement uncertainty.

The sections of the paper are organized as follows: Section 2 briefly describes the proposed sensor fusion architecture. In Section 3, we present two fusion algorithms based on statistical sensor models. Section 4 evaluates both algorithms using data from infrared and ultrasonic distance sensors. Finally, Section 5 discusses related approaches. The paper is concluded in Section 6.

2 Time-Triggered Sensor Fusion Model

2.1 Architectural Aspects

The Time-Triggered Sensor Fusion Model [2] proposes the implementation of a sensor fusion application on top of the Time-Triggered Architecture [10].

The Time-Triggered Architecture proposes a strictly synchronous design, where each task and communication activity is planned *a priori* in a static schedule. All distributed nodes are synchronized to a global time base, which enables the nodes to perform coordinated actions like measurement or actuator settings. Furthermore, the design supports an easy verification of the timing constraints.

The Time-Triggered Sensor Fusion Model describes a set of jobs that represent all necessary activities like measurement, data processing, decision, and actuation. The jobs are represented as vertexes in a distributed graph, whereas each communication activity is represented by an edge between the service providing linking interface (SPLIF) of the job that provides the data and the service requesting linking interface (SRLIF) of the job that

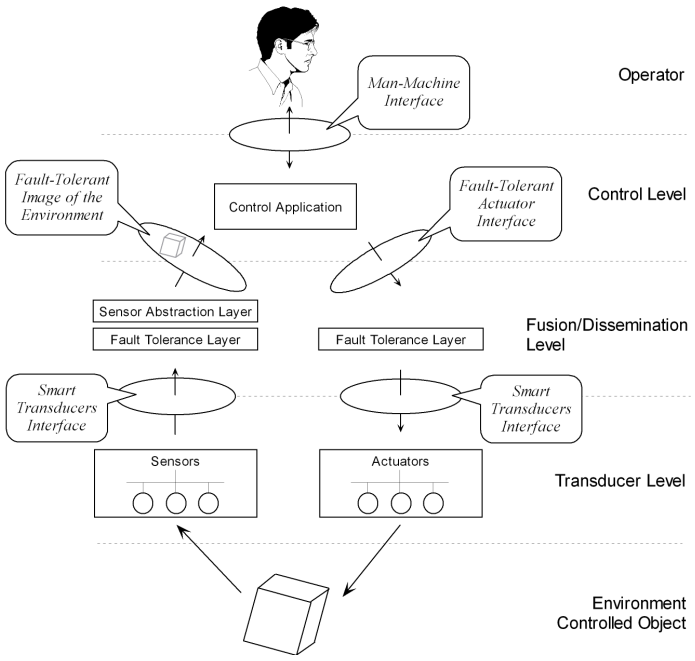

Figure 1. Data flow in the time-triggered sensor fusion model

receives the data. A physical node may host one or several jobs, thus two logically different tasks may be split up into two jobs but still be executed on the same microcontroller subsequently.

The job graph is furthermore structured hierarchically into three levels in order to distinguish between transducers (direct interfaces to the environment), fusion and dissemination activities, and decision activities.

Figure 1 depicts a control loop modeled by the Time-Triggered Sensor Fusion Model. Interfaces are illustrated by a disc with arrows indicating the possible data flow directions across the interface. Physical sensors and actuators are located on the borderline to the process environment and are represented by circles. All other components of the system are outlined as boxes. The model distinguishes three levels of data processing with well-defined interfaces between them. The *transducer level* contains the sensors and actuators that interact directly with the controlled object. A *smart transducer interface* provides a consistent borderline to the above *fusion/dissemination level*. This level contains fault tolerance and sensor fusion tasks. The *control level* is the highest level of data processing within the control loop. The control level is fed by a dedicated view of the environment (established by transducer and fusion/dissemination level) and outputs control decisions to a *fault-tolerant actuator interface*. User commands from an operator interact with the control application via the *man-machine interface*.

Prerequisites for implementing an application in the Time-Triggered Sensor Fusion Model are

1. A deterministic time-triggered communication system that supports coordinated task execution.
2. A known upper bound for the computation time of each job in the real-time control loop. Thus, the Worst-Case-Execution Time (WCET) has to be determined for each job.

The first constraint can be achieved by choosing an appropriate communication system such as TTP/A [11], TTP/C [16], Flexray [4], TTCAN [6], and Time-Triggered Ethernet [9].

The experimental results presented in Section 4 have been acquired using a TTP/A communication network.

The second constraint is non-trivial for complex algorithms and/or complex hardware due to effects of pipelines/caches. In practice, engineers measure the execution time by several runs of an algorithm in order to estimate its WCET. However, for most algorithms, measuring the execution time for an arbitrary input is not sufficient to find the WCET. Exact algorithms for WCET determination can be found in [13] and [18].

For this reason, we prefer sensor fusion algorithms with low complexity in order to be able to determine a safe bound for its execution time. Examples for such algorithms are Kalman Filtering [8], which periodically requires a fixed number of matrix operations, Marzullo's abstract reliable sensor approach [12], and, as well, the Confidence-Weighted Averaging (CWA) [1] being further described in Section 3.

3 Fusion Algorithms

3.1 Problem Statement

Given is a set of sensors that measure the same real-time entity in the process environment. We assume the sensors to be calibrated, so the expected average measurement error within the measurement range is 0.

The sensors can be of different nature and therefore are expected to provide measurements with a different measurement uncertainty. Moreover, the measurement uncertainty of a sensor is not assumed to be constant, but varies within the measurement range.

According to the *Guide to the Expression of Uncertainty in Measurement* [7], the statistical variance should be used as a measure for uncertainty.

A smart sensor, as proposed in the architecture from Section 2, will thus provide two aspects of a measurement. The first aspect is the measurement of the observed entity, whereas the second aspect is a value indicating the variance of the first value.

3.2 Confidence-Weighted Averaging

This algorithm takes several sensor measurements and performs an averaging algorithm. To combine the measurements, a simple mean of all the values does not usually perform well enough, since we assume the inputs to have different uncertainty. Thus, we use adjusted weights to fuse N measurements:

$$x_{FUSED} = \sum_{i=1}^{N} x_i w_i \quad \text{with} \quad \sum_{i=1}^{N} w_i = 1 \tag{1}$$

The weights w_i are determined by the variance of the measurement. In [3] it is shown that the following weight function is optimal, given that the measurement errors of the particular sensors are statistically independent:

$$w_i = \frac{1}{\mathbb{V}(S_i) \sum_{j=1}^{n} \frac{1}{\mathbb{V}(S_j)}} \tag{2}$$

The variance of the result can also be estimated from the variances of the particular measurements:

$$\mathbb{V}(S_{FUSED}) = \frac{1}{\sum_{i=1}^{n} \frac{1}{\mathbb{V}(S_i)}}. \tag{3}$$

When fusing two or more sensors, the resulting expected variance is always lower than the best selected sensor.

Thus, the algorithm is easy to implement and provides robustness against sensor errors, i. e., sensor errors affect the calculated output in an attenuated way, making the systems performance degrading slowly with increased number of sensor faults and inaccuracies.

3.3 Extended-Confidence Weighted Averaging

The assumption of independency of sensor errors, as taken in the CWA algorithm, cannot be made in the general case. Ignoring correlations between fusion inputs results in a suboptimal assignment of weights to each observation, and a distorted estimate of the variance of the result.

In order to take correlations between sensor errors into account, the CWA method has been extended as described in [15].

As with CWA, Extended Confidence-Weighted Averaging (ECWA) is based on a weighted averaging approach as described by Equation 1. However, the optimal weight vector $\mathbf{w}$ is calculated as a function of the error covariance matrix of all sensors:

$$\mathbf{w}^* = (\boldsymbol{\sigma_{11}}\mathbf{1^T} - \mathbf{c_1}\mathbf{1}^T - \mathbf{1}\mathbf{c_1}^T + \mathbf{C}^*)^{-1}(\boldsymbol{\sigma_{11}} - \boldsymbol{c_1}) \tag{4}$$

$$\mathbf{w_1} = 1 - \mathbf{1}^T\mathbf{w}^* \tag{5}$$

$$\mathbf{w} = \begin{bmatrix} \mathbf{w_1} \\ \mathbf{w}^* \end{bmatrix} \tag{6}$$

with σ_{11} is the first element in the first row of the covariance matrix, $\mathbf{c_1}$ represents the first column of the covariance matrix except for its first element, $\mathbf{C}^*$ is the covariance matrix without its first column and first row and $\mathbf{w}^*$ is the weight vector $\mathbf{w}$ without its first element. $\mathbf{1}$ is a vector where all elements are 1.

The expected variance of the fused result of a weighted average is now calculated by the more general formula as a function of the weights and the correlation matrix:

$$\mathbb{V}(S_{FUSED}) = \sum_{i=1}^{n}\sum_{j=1}^{n} w_i w_j \sigma_{ij}. \tag{7}$$

For sensor measurement with rather uncorrelated errors, the ECWA behaves the same as the CWA algorithm. While the ECWA algorithm is a little bit more demanding in its computation, it is expected to overcome problems with data sources with different correlations, as for example a configuration with two sensors of type A and one sensor of type B. Assuming that the error functions are more correlated between the two sensors of the same type, the ECWA algorithm will put a greater weight on the sensor of type B since it is expected to be more independent.

4 Experimental Evaluation

We have fused data from three Sharp GP2D02 infrared sensors and the Panasonic ultrasonic sensors using the CWA and ECWA algorithm.

4.1 Measurement Architecture

The sensors are mounted on a small four-wheeled robot and instrumented by a TTP/A [11] fieldbus network that periodically triggers sensor measurements. The measurement results are sent via a gateway to a host pc that gathers and analyzes the data.

The measurement set-up is depicted in Figure 2.

An obstacle is placed in front of the robot in several defined distances. The accuracy of the placement of the obstacle is about 0.5 cm, while the sensor measurement errors are around several centimeters, allowing us to neglect the placement error.

We have collected a set of several thousand measurements for distances between 10 cm to 100 cm.

In this set-up the robot did not move. For future experiments we are planning to automatically move the robot towards a wall while measuring the covered distance.

4.2 Results and Discussion

The error distribution from the measurements by an infrared and an ultrasonic sensor are depicted in Figure 3. The infrared sensors showed a rather high inaccuracy with a bizarre likelihood distribution. The ultrasonic sensors turned out to be more accurate.

Figures 4 and 5 depict the efficiency of the CWA and ECWA fusion algorithm for several possible homogeneous and heterogeneous combinations. The performance regarding the accuracy of the fused result is similar for both algorithms. However, when correlation in error functions are present, the CWA overestimates the confidence of the fusion result. Using the ECWA algorithm, the quality of the result is estimated correctly. However, while the ECWA algorithm showed to perform better, it requires more information about the

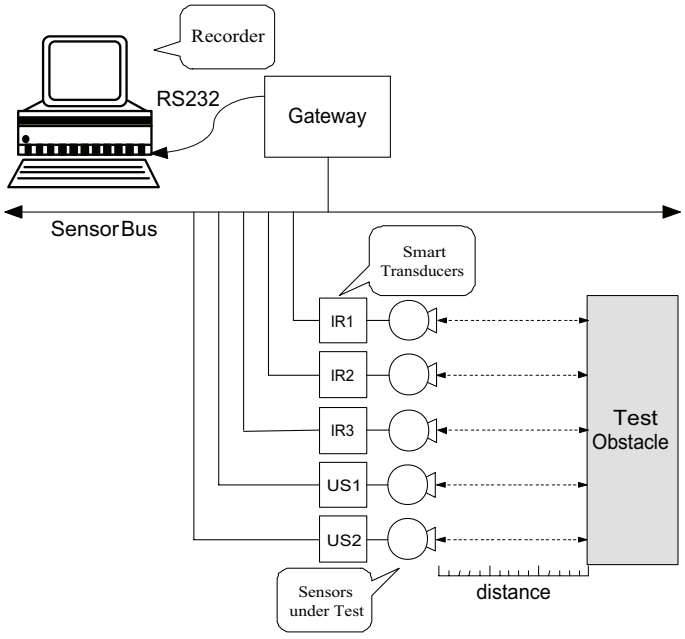

Figure 2. Measurement set-up for gathering the sensor data

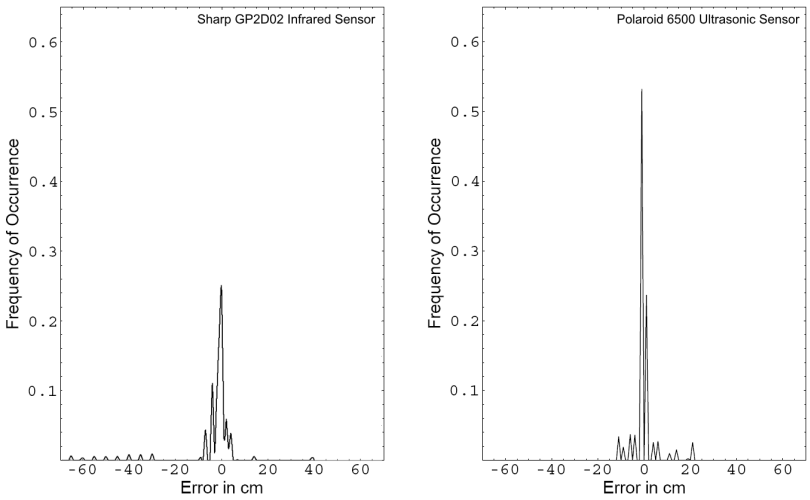

Figure 3. Error histograms for two different distance sensors

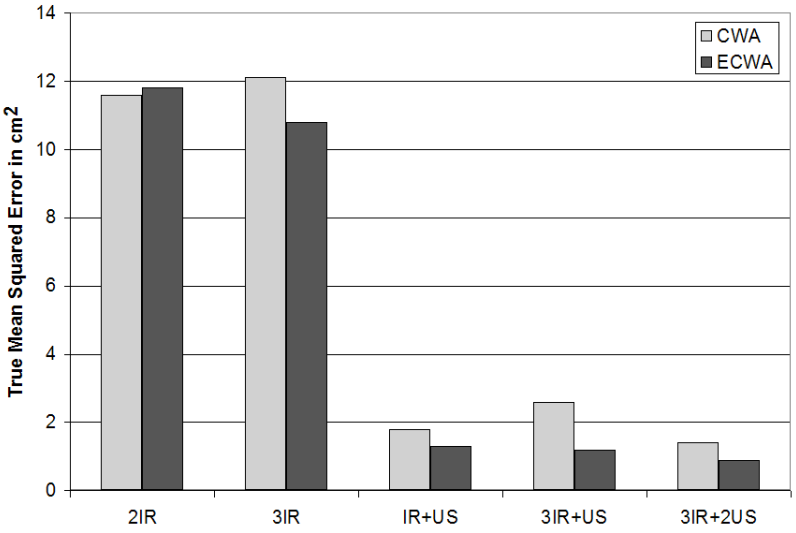

Figure 4. Result from the fusion of several sensor configurations

overall system than CWA. Since the CWA requires only the estimated variance for each sensor, each sensor can be calibrated and analyzed separately and then the system can be set together out of these sensors without requiring any cross-analysis of sensor behavior. For ECWA this composability principle is not given, since for each new sensor added to the system, one needs to perform an analysis of the error correlation between the new sensor and all other sensors.

Moreover, the CWA algorithm requires only one division, one multiplication, and two add operations for each sensor measurement. The resulting complexity of the algorithm is $\Omega(n)$ where n is the number of sensors. Thus, a CWA can be easily applied for small embedded devices. On the other hand, ECWA requires several matrix operations and a matrix inversion in order to determine the fusion weights. Since we assume the variance of subsequent measurements to change during system operation (the correlation matrix is

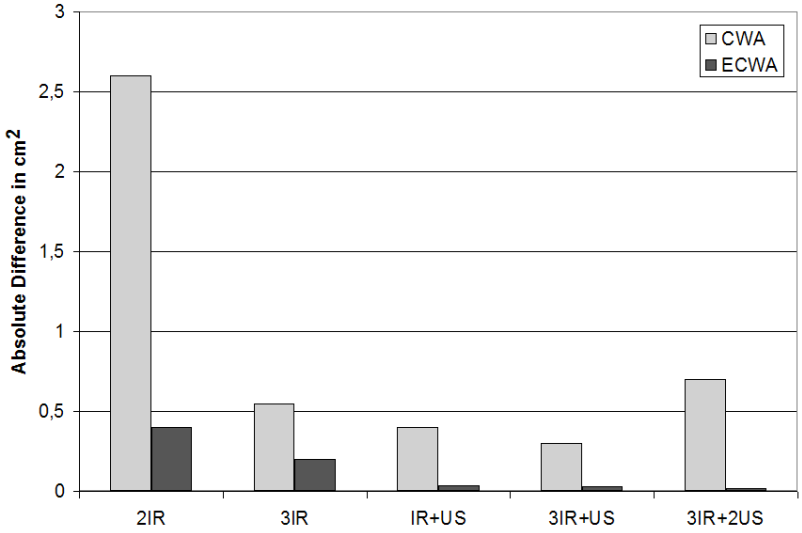

Figure 5. Difference in estimation of the resulting variance of the fusion result

assumed to be *a priori* known, thus the covariances can be derived from the variance), these matrix operations have to be performed in real-time for each fusion operation. Taking into account the complexity of the matrix inversion [17], the complexity of ECWA is $\Omega(n^2 log(n))$. Even for small n the number of operations is considerably higher than for CWA.

On the other hand, when the number of used sensors is large, the correlation among the measurement errors becomes more important. For the number of sensors going to infinity, the CWA model predicts a variance of 0 for the fused result, while in the ECWA model this value correctly approaches the sensors' covariance.

5 Related Approaches

5.1 Sensor Selection

One possible approach would be a selection of the measurement with the best, that is lowest, variance among all the measurements of one instant. While sensor selection [5] is rather easy to implement, this approach suffers from the following drawbacks:

- The algorithm is not very robust against faulty measurements. For example if a sensor provides a faulty measurement but pretends to deliver a very accurate value, the faulty value is selected in spite of the other available correct measurements which have been annotated with higher variance estimation than the faulty sensor.
- With subsequent correct measurement samples over time, the algorithm might select a different sensor from one instant to the other, if the annotated variance of the sensors change. This can cause a disturbance of the fused value with the sampling frequency, if calibration of the sensors is imperfect (which usually is the case in real systems).
- The result from the sensor selection algorithm will only be of the same quality as the

best sensor. In the analysis in section 3 of this paper we have shown that it is possible to improve the quality of the fused result over that limit of the best single sensor.

5.2 Abstract Reliable Sensors

Marzullo [12] proposed a method for fusion of continuous-valued sensor measurements by fault-tolerant interval intersection. This approach and improved algorithms based on this approach [14] have been used extensively in clock synchronization algorithms but have been also applied to sensor fusion.

In Marzullo's algorithm, each sensor measurement is modeled by an interval that should contain the real sensor measurement. If a sensor delivers a measurement with the real value outside this interval, the sensor is considered to be faulty. It is required to have a fault hypothesis about the maximum expected number of faulty sensors at the same time.

Using such a fault-tolerant approach will provide a rather robust behavior. However, with respect to the behavior of the sensors used in this paper, defining an adequate interval will be difficult. The distinction between a correct working sensor with a high measurement deviation and a sensor considered faulty introduces discontinuities at the interval boundaries. Such discontinuities are not meaningful for real sensors, which provide an accuracy that is typical known only in the order of magnitude. Large intervals lead to worse average performance while modeling a sensor's correctness only within a small requires resilience against a high number of faulty sensors. This requires a tradeoff between delicate interval selection and tolerable faults.

However, the fault-tolerant concept of abstract reliable sensors could be integrated with confidence-weighted averaging in order to achieve resilience against extreme sensor faults.

6 Conclusion

The proposed approach supports real-time sensor fusion and control applications by proposing a rigid time-triggered measurement, communication, and computation scheme.

The CWA algorithm supports the fusion of data from several different heterogeneous sensors. When it can be assumed that the error functions of the sensors are statistically independent, the fusion nodes can be designed and implemented independently from each other using the CWA algorithm. However, experimental data has shown that at least for distance sensors there is always a small amount of correlated sensor errors. The error correlations have been expected for homogeneous sensor configurations, but also exist for sensors from different kind.

Especially in configurations with groups of several heterogeneous sensors, the error correlation between any two sensors will likely be rather different. Without taking these information into account, the fusion result will be suboptimal and overestimated.

Using the ECWA algorithm, these correlations can be taken into account, however at the cost of a more complex algorithm in terms of required information and computational complexity. In our example this did not so much influence the quality of the fused result, but greatly increased the estimation of the result's quality.

In future work we are planning to simplify the modeling of sensor error correlation in order to reduce the complexity and required knowledge for the system set-up.

Acknowledgment

This work was supported by the Austrian FWF project TTCAR under contract No. P18060-N04. We would like to thank the anonymous reviewers for their constructive comments on an earlier version of this paper.

References

[1] W. Elmenreich and P. Peti. Achieving dependability in a time-triggered network by sensor fusion. In *Proceedings of the 6th IEEE International Conference on Intelligent Engineering Systems (INES)*, pages 167–172, Opatija, Croatia, May 2002.

[2] W. Elmenreich and S. Pitzek. The time-triggered sensor fusion model. In *Proceedings of the 5th IEEE International Conference on Intelligent Engineering Systems*, pages 297–300, Helsinki–Stockholm–Helsinki, Finland, September 2001.

[3] W. Elmenreich and A. Schörgendorfer. Fusion of continuous-valued sensor measurements using statistical analysis. In *International Symposium on Mathematical Methods in Engineering (MME-06)*, Ankara, Turkey, April 2006.

[4] Flexray Consortium. *FlexRay Communications System Protocol Specification Version 2.1*, 2005. Available at http://www.flexray.com.

[5] C. Giraud and B. Jouvencel. Sensor selection: A geometrical approach. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, volume 2, pages 555–560, Pittsburgh, PA, USA, August 1995.

[6] F. Hartwich, B. Müller, T. Führer, and R. Hugel. Time triggered communication on CAN. In *Proceedings 7th International CAN Conference*, Amsterdam, The Nederlands, 2000.

[7] International Organization for Standardization (ISO), Genève, Switzerland. *Guide to the Expression of Uncertainty in Measurement*, 1st edition, 1993.

[8] R. E. Kalman. A new approach to linear filtering and prediction problems. *Transaction of the ASME, Series D, Journal of Basic Engineering*, 82:35–45, March 1960.

[9] H. Kopetz, A. Ademaj, P. Grillinger, and K. Steinhammer. The Time-Triggered Ethernet (TTE) design. In *Proceedings of the 8th International Symposium on Object-Oriented Real-Time Distributed Computing (ISORC)*, pages 22–33, Seattle, WA, USA, May 2005.

[10] H. Kopetz and G. Bauer. The Time-Triggered Architecture. *Proceedings of the IEEE*, 91(1):112–126, January 2003.

[11] H. Kopetz et al. Specification of the TTP/A protocol. Research Report 61/2002, Technische Universität Wien, Institut für Technische Informatik, Vienna, Austria, September 2002. Version 2.00.

[12] K. Marzullo. Tolerating failures of continuous-valued sensors. *ACM Transactions on Computer Systems*, 8(4):284–304, November 1990.

[13] P. Puschner and A. Burns. A review of worst-case execution-time analysis. *Journal of Real-Time Systems*, 18(2/3):115–128, May 2000.

[14] U. Schmid and K. Schossmaier. How to reconcile fault-tolerant interval intersection with the Lipschitz condition. *Distributed Computing*, 14(2):101–111, April 2001.

[15] A. Schörgendorfer and W. Elmenreich. Extended confidence-weighted averaging in sensor fusion. In *Proceedings of the Junior Scientist Conference JSC'06*, pages 67–68, Vienna, Austria, April 2006.

[16] TTAGroup. *Specification of the TTP/C Protocol V1.1*, 2003. Available at http://www.ttagroup.org.

[17] A. Tveit. On the complexity of matrix inversion. Technical report, Department of Computer and Information Science, Norwegian University of Science and Technology, Trondheim, Norway, 2003.

[18] I. Wenzel, B. Rieder, R. Kirner, and P. Puschner. Measurement-based worst-case execution time analysis. In *Proc. 3rd IEEE Workshop on Software Technologies for Future Embedded and Ubiquitous Systems (SEUS'05)*, pages 7–10, Seattle, WA, USA, May 2005.