TIME DELAY BASED FAILURE-ROBUST DIRECTION OF ARRIVAL ESTIMATION

T. W. Pirinen *

International Computer
Science Institute (ICSI)
1947 Center St., Berkeley, CA 94704, USA

J. Yli-Hietanen

Tampere University of Technology (TUT)
Institute of Signal Processing
Korkeakoulunkatu 1, 33720 Tampere, Finland

ABSTRACT

There is an increasing interest in autonomous and unattended sensory systems, including applications in smart meeting rooms and surveillance of large areas. In these systems, direction of arrival (DOA) estimation has an important role. Although several time delay based algorithms have been proposed, little attention has been given to hardware failures and malfunctions. These are, however, an important consideration when smart arrays and intelligent sensor systems are developed. In this paper, we address this problem by modifying a previously proposed single-failure detector to suit a multiple failure case. The performance of the modified method is tested both with simulations and speech data recorded using a microphone array in a meeting room. Results indicate that the proposed method can detect failures effectively. The detector is also combined with DOA estimators. Results show that this combination is robust against failures and the detector provides an efficient tool for array self-diagnosis.

1. INTRODUCTION

Recent developments in sensory systems and hardware have created new applications for small passive arrays. As new applications have appeared, also new problems and challenges require consideration. One of these problems is the possibility of errors and failures that corrupt the array processing. This problem has been acknowledged also in acoustic processing [1], and attempts have been made to to measure the reliability of estimation [2], validate the used sensors [3], and detect failed hardware [4]. The risk of failures becomes even larger if systems are automatically deployed, unattended, or operate for long periods of time in varying conditions. Also, the price for any such system should be as low as possible, enabling the use of these systems in more applications. However, low-cost hardware may have lower quality and larger probability of failures.

One key application for acoustic arrays is direction of arrival (DOA) estimation and source localization [5]. Time delay based methods have been suggested as the most robust approach for DOA estimation [6], and several algorithms have been proposed [6, 7, 8]. Failure detection for time delay based methods was proposed in [4]. This approach assumed a small sensor array and approximately planar wavefronts. The method searched for a sensor producing low-confidence time delays and a detection buffer was used to decide if the sensor had failed. However, the detection stage in [4] is not suitable for detecting multiple failures. In this paper, we modify the detection stage to suit the multiple failure case. The performance of the modified detector is tested with simulations and the detector is also combined with several time delay based direction of arrival estimation methods to demonstrate its usefulness. This combination also results in a failure-robust time delay based direction of arrival estimator. A speech data example from a meeting room recording is also given.

*Author is visiting ICSI from TUT until August 2004. Partial funding from the Nokia Foundation is gratefully acknowledged.

Results indicate that the proposed method can effectively detect failed channels. When these channels are removed from the processing, estimation can continue without the failure effects. Also, the detector provides an efficient tool for array self-diagnosis and can provide users with status of the hardware and e.g. alerts for maintenance. In addition, the proposed method is not limited to DOA estimation only, but can be used for other array applications, such as beamforming.

The paper is organized a follows. Section 2 gives briefly the prerequisites for the problem at hand. The failure detection method is explained and modifications proposed in Section 3 and results from simulations and real data are given in Section 4. Finally, Sections 5 and 6 provide some discussion, ideas for further work and concluding remarks.

Because the experiments provided a host of results, an online page was set up to provide all the results for the interested reader: These complete results can be found at:
`http://www.cs.tut.fi/sgn/arg/spatial/sam04/`

2. PROPAGATION VECTOR AND TIME DELAYS

Propagation of a planar wave can be described with the *propagation vector*, denoted with $\boldsymbol{k}$. This vector has the direction of wavefront propagation and its magnitude is the inverse of the propagation speed of the wave [9]. When a planar wave passes two sensors of a *sensor pair* (i, j), there is a time delay $\tau_{(i,j)}$ between the sensor signals. This time delay can conveniently be characterized using the propagation vector [10]

$$\tau_{(i,j)} = \boldsymbol{x}_{(i,j)} \cdot \boldsymbol{k} \tag{1}$$

where $\boldsymbol{x}_{(i,j)}$ is the *sensor vector* from sensor i to sensor j. Using all sensor pairs in the array and Eqn. (1), all time delays in the array are determined by a single matrix equation

$$\boldsymbol{\tau} = \boldsymbol{X}\boldsymbol{k} \tag{2}$$

where $\boldsymbol{X}$ is a matrix having the $\binom{N}{3}$ sensor vectors as its rows. (N is the number of sensors.) Therefore, the time delays are a linear function of the propagation vector. Given $\boldsymbol{\tau}$, DOA estimate is then simply the solution of Eqn. (2) for $\boldsymbol{k}$ from estimated time delays [7]. Typically the time delay estimation is based on cross-correlation between channels [11].

There are several ways to solve Equation (2):

1. The Moore-Penrose least squares pseudoinverse (LS)

$$\hat{\boldsymbol{k}} = \left(\boldsymbol{X}^T\boldsymbol{X}\right)^{-1}\boldsymbol{X}^T\hat{\boldsymbol{\tau}} \tag{3}$$

2. Averaging within subarrays (AVG)

$$\hat{\boldsymbol{k}} = \frac{1}{\#\mathcal{V}}\sum_{v\in\mathcal{V}}\boldsymbol{X}_v^{-1}\hat{\boldsymbol{\tau}}_v \tag{4}$$

   where $\mathcal{V}$ is the set of subarrays and $\boldsymbol{X}_v$ and $\hat{\boldsymbol{\tau}}_v$ are the sensor vectors and time delays in subarray v, respectively. Typically the set $\mathcal{V}$ is the set of all triplets of linearly independent sensor vectors. The nonlinear version of this method using trigonometric functions and angles of arrival has been given in [12].

3. Time delay selection (TDS) for least squares solution [13]. In this approach only the most confident time delays are used for solution. The definition for confidence will be explained further in Section 3.

4. Search for a solution $\boldsymbol{k}_0$ such that the difference between estimated time delays $\hat{\boldsymbol{\tau}}$ and time delays $\boldsymbol{\tau}_0 = \boldsymbol{X}\boldsymbol{k}_0$ is minimized. A computationally efficient method based on the *count-distance* (CNT) difference metric was proposed in [14].

The first three approaches are closed form solutions, whereas the methods in the last category are exhaustive searches in the solution space and thus computationally much more expensive. To be precise, the TDS solution requires also some searching to find the best time delays but the computations required for this operation are only a fraction of those required in the actual search methods. Also, the AVG method can become computationally expensive if the number of sensors, and thus subarrays, is large.

Note that this kind of estimation gives an unambiguous 3-D DOA estimate, provided that the array is also 3-D, and does not require knowledge on propagation speed. In fact, the propagation vector estimate gives an estimate also for the propagation speed, although in practical situations this estimate is very rough.

3. FAILURE DETECTION

If there is a malfunction in a sensor or some other part of the signal channel, the received signal is corrupted. As a result, time delay estimates related to this sensor may become erroneous. Because this work focuses on time delay based DOA estimation, signal errors not corrupting the time delay estimates are not an issue and therefore do not require detection.

Consider a closed path formed using sensor vectors, i.e. a set $\mathcal{P}$ of sensor pairs selected such that

$$\sum_{(i,j)\in\mathcal{P}}\boldsymbol{x}_{(i,j)} = 0 \tag{5}$$

Having $\boldsymbol{k} \neq 0$ it follows from Eqn. (1) that

$$\sum_{(i,j)\in\mathcal{P}}\tau_{(i,j)} = 0 \tag{6}$$

In other words, whenever a planar wave creates time delays in an array, the sum of time delays should be zero on any closed sensor vector path in the array if estimation has been successful. Errors in time delay estimates can be written as

$$\hat{\tau}_{(i,j)} = \tau_{(i,j)} + e_{\tau(i,j)} \tag{7}$$

and these errors propagate to the closed-path sum of estimates (6)

$$\sum_{\mathcal{P}}\hat{\tau}_{(i,j)} = \underbrace{\sum_{\mathcal{P}}\tau_{(i,j)}}_{=0} + \sum_{\mathcal{P}}e_{\tau(i,j)} = \sum_{\mathcal{P}}e_{\tau(i,j)} \tag{8}$$

Therefore, error testing can be done by computing closed-path sums from the array and this is the starting point for the failure detection algorithm presented in [4]. After time delays from all sensor pairs have been estimated, the algorithm tests all sensor pair *triangles* for the zero sum condition

$$\Delta_{(i,j,l)} = \begin{cases} 1 \ , & \left|\hat{\tau}_{(i,j)} + \hat{\tau}_{(j,l)} - \hat{\tau}_{(i,l)}\right| < D_{(i,j,l)} \\ 0 \ , & \text{otherwise} \end{cases} \tag{9}$$

Here, $D_{(i,j,l)}$ is the testing threshold that can be obtained by first setting the norm of maximum tolerable propagation vector error and then computing the threshold using the geometry of the sensor vector triangle [13]. The results of testing form the *confidence factors* [13] for each time delay:

$$\Upsilon(i,j) = \sum_{l\in\mathcal{X}-\{i,j\}}\Delta_{(i,j,l)} \tag{10}$$

If all confidence factors of some sensor are zeros, a failure indicator is given to this sensor. If a sensor produces failure indicators at a rate exceeding the given thresholds, it will be detected as failed.

The detection method in [4] searched only for one sensor producing failure indicators and was therefore able to detect only one failure at a time. The method could be used recursively to detect several failures, but convergence is not guaranteed if two or more sensors fail simultaneously or within the detection buffer.

To overcome this problem, each channel should have its own failure indicator sequence. Instead of one, all sequences are then monitored to decide if the failure threshold has been exceeded in some channel. Therefore we define the following failure indication buffer

$$\boldsymbol{W}(i,n) = \begin{cases} 1 \ , & \forall s\in\mathcal{S}_i : s = 0 \\ 0 \ , & \text{otherwise} \end{cases} \tag{11}$$

Here, i is the sensor index, n indexes the time frame being processed and $\mathcal{S}_i$ is the set of confidence factors for sensor i at time instant n. In other words, if all confidence factors of sensor i are zeros at time n, $\boldsymbol{W}(i,n)$ is set to one, otherwise zero. For the reasons explained in [4], failure detection can not be based on information from one time frame, but consecutive time frames must be used. The detector then decides sensor i to have failed if D_{fail} or more of previous B values on row i of $\boldsymbol{W}$ are ones.

4. EXPERIMENTS AND RESULTS

4.1. Simulations

The proposed detection method was tested with simulations and real data. In the simulations, time delays were generated and sensor failure was simulated using the procedure described in [4]. Briefly, in this method array operation begins without failures. If a failure occurs, a three-state model is used to generate time delays from three different distributions (error states). To avoid excessive

©2004 IEEE. Original version to appear in *Proceedings of the Third IEEE Sensor Array and Multichannel Signal Processing Workshop (SAM 2004)*. Personal use of this material is permitted. However, permission to reprint/ republish this material for advertising or promotional purposes or for creating new collective works for resale or redistribution to servers or lists, or to reuse any copyrighted component of this work in other works must be obtained from the IEEE.

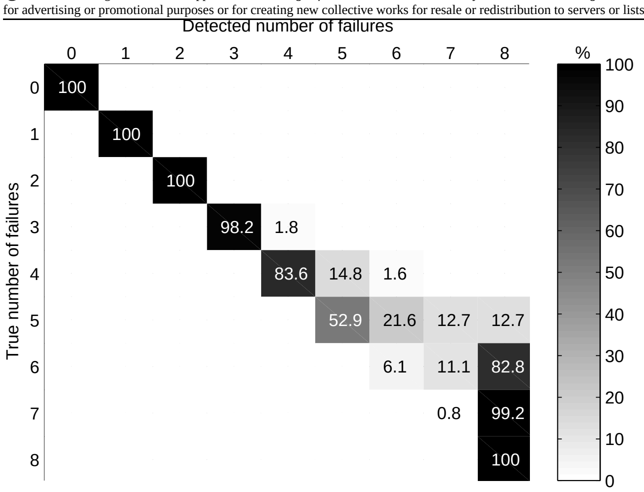

Fig. 1. Confusion matrix for detection of number of failed sensors. Rows are numbered by the true number of failures, and columns by detection. When there were less than three failures, detection was perfect. Accuracy remains up to four failures, which is satisfactory because with five or more failures 3-D DOA estimation is not possible with the used eight sensor array. Zero entries are not labeled.

idealization, some errors do occur even when there are no failures, and in the failure mode time delays may be partially correct.

The simulated array had eight sensors at the apices of a 0.31 m cube. The base of the cube corresponds to a maximum time delay of approximately nine samples at 10 kHz sampling rate and propagation speed $c = 343\ \frac{\text{m}}{\text{s}}$.

Simulations consisted of 40000 DOA estimation runs, with 1000 time frames in each run. Number of failing sensors was selected randomly from 0 to 8. Each case had equal probability of $1/9$. Thresholds for triangular tests, $D_{(i,j,l)}$ were computed using value $1/(20c)$ for the norm of maximum tolerable error, as in [4]. Failure detection buffer length was $B = 10$ frames with detection threshold $D_{fail} = 5$ frames.

Detection performance was analyzed in several stages. Firstly, the ability to detect the number of failures was tested. The confusion matrix for this task is given in Figure 1. With less than three failures, there are no errors and over 80% accuracy remains up to four failures. After five failures, performance collapses, but this is not a problem because with five or more failures, 3-D DOA estimation is not possible with the used eight sensor array.

Note that the confusion matrix is not symmetric, because rows and columns have different interpretations. Rows represent the true number of failures, whereas columns give the detection result. The detection error tends to be biased towards too large values. When the number of failures is large, errors in time delays become interconnected and these cases (e.g. seven and eight failures) are difficult to separate, which explains the large errors. As explained above, this is not a significant problem, since 3-D DOA estimation is not possible in these situations.

Because wrong sensors may be detected as failed even if the number of failures was correctly identified, correct identification of the *set* of failed sensors was the second stage of performance analysis. Table 1 shows the results for this analysis for four cases. Two first columns represent the failures detected in correct order.

	correct order % of all	correct order % of correct #	correct set % of all	correct set % of correct #
1	100.0	100.0	100.0	100.0
2	100.0	100.0	100.0	100.0
3	98.2	100.0	98.2	100.0
4	82.8	99.1	83.6	100.0
5	51.0	96.3	52.9	100.0
6	6.1	100.0	6.1	100.0
7	0.8	100.0	0.8	100.0
8	34.1	34.1	100.0	100.0

Table 1. Detection accuracy for failed sensors. In the first two columns only detection in correct order was allowed. Two latter columns allow also the possibility of wrong order of detections and detection of a failure before a it had occurred, as long as the correct set was detected in the end. Detection performances are given relative to all cases (colums 1 and 3) and only those cases in which the number of failures was correctly detected (columns 2 and 4). Small differences between correct order and correct set detections demonstrate that up to five failures the detection happens in correct order.

Values are given relative to all cases and cases in which the number was correctly identified, respectively. Two latter columns give the cases in which the order of detections may have been wrong (and even noncausal), but in the end the correct set of failed sensors was detected. These results are also given relative to all cases and only those cases where number of failures was correctly detected, respectively. Up to five failures, the differences between detection in correct order and detection of correct set do not exceed one percentage point. This indicates that whenever the failed set is detected correctly, the detection also happens in correct order and causally.

The third stage of the analysis was the measurement of detection delay, i.e. the time elapsed from failure to detection. The average delays in samples for detection of first, second, third, etc. failure were 16.1, 14.7, 10.6, 7.8, 4.3, 4.2, 6.0 and 27.4, respectively.

Actual DOA estimation performance was analyzed by computing estimates from simulated time delays. Used estimation methods were LS, AVG, TDS and CNT (see Section 2). Failure detection and removal of failed channels was combined with each method (detection stage adds letter 'd' at the end of each acronym). Error metric was the angular error, i.e. angle between true and estimated vectors in 3-D space

$$\theta = \text{acos}\left(\frac{\left|\hat{\boldsymbol{k}}^T\boldsymbol{k}\right|}{\|\hat{\boldsymbol{k}}\|\|\boldsymbol{k}\|}\right) \qquad (12)$$

This metric was chosen, because it measures the DOA estimation accuracy directly and also corresponds to great circle distances between true and estimated DOA. Moreover, it should be emphasized that Euclidean distance is not suitable for this error analysis because the DOA estimation space is spherical, although the propagation vector space is Euclidean. To keep the metric consistent, only DOA estimates after the failure detection were used in error analysis.

Average angular errors for each failure case are given in Figure 2 and as an example of the error distributions, cumulative distribu-

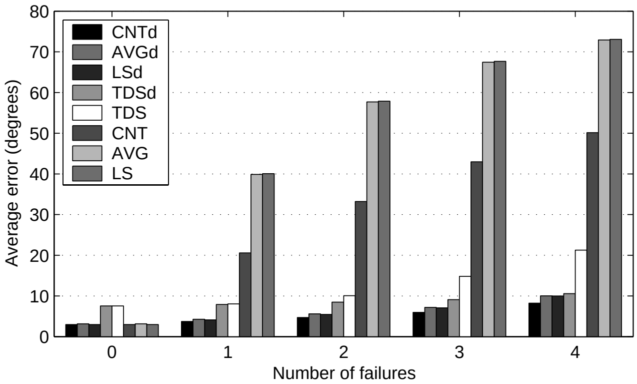

Fig. 2. Average errors in degrees (angle between vectors) for DOA estimators in simulations as a function of number of failures. Proposed detection method (acronyms ending with 'd') improves estimation performance in all failure cases, and methods using detection outperform the others in all cases. In each failure case, methods are given in the order they are in the legend.

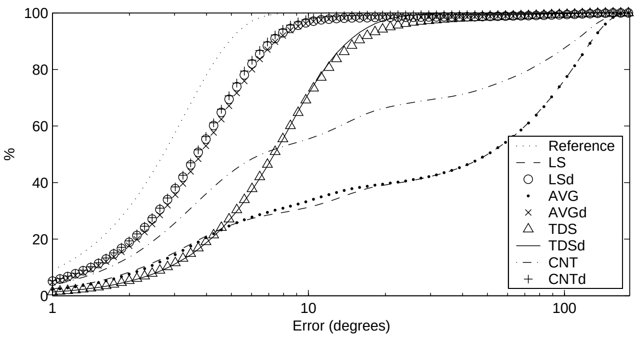

Fig. 3. Cumulative distributions of error (angle between vectors) for DOA estimates when **one sensor** has failed. Failure degrades the performance of all methods, except TDS. Detection (acronyms ending with 'd') restores the estimation performance and gives better performance than TDS only. Angle axis is logarithmic for visualization.

tion of error for all methods in case of one and three failures are given in Figures 3 and 4, respectively. Results clearly indicate that the proposed detection method improves DOA estimation performance in the presence of failures, even for the robust TDS method. This, combined with the fact that detection accuracy is good for small numbers of failures, indicates that the proposed method is robust to failures.

It should be noted that the TDS method is indeed robust and performs quite well even without the detection stage. The tradeoff for this robustness, however, is the large error (more than twice of that of the other methods) when no failures have occurred.

4.2. Speech data experiment

The proposed method was also tested with speech data from the IDIAP Smart Meeting Room [15]. The signals have been recorded with an eight sensor circular array at 16 kHz sample rate. Because the IDIAP database does not provide locations for speakers, there

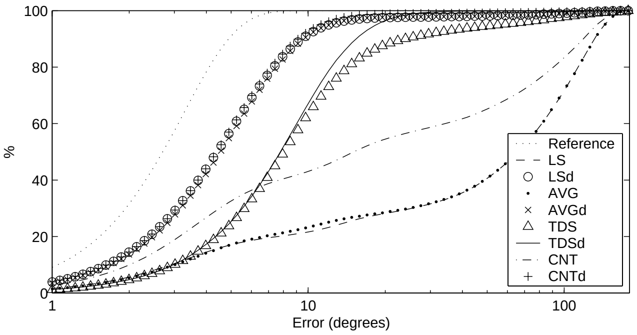

Fig. 4. Cumulative distributions of error (angle between vectors) for DOA estimates when **three sensors** have failed. Now the failure degradation for TDS is also visible. Failure detection provides still major improvements, although original performance can not be achieved. Angle axis is logarithmic for visualization.

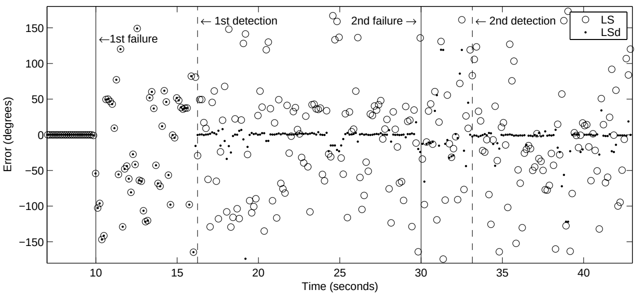

Fig. 5. DOA estimation errors for a speech signal with LS and LSd methods. Least squares estimate without failures was used as the reference. Failures occur at $t = 10.0$ s and $t = 30.0$ s. Detections are at $t = 16.3$ s and $t = 33.2$ s. Failure detection (in LSd) restores the DOA estimation capability.

was no absolute reference for DOA estimates. However, for the failure detection task it is sufficient to use the DOA estimates from original signals not including failure effects. In the given example, failures were inserted at $t = 10.0$ s in channel 2 and at $t = 30.0$ s in channel 4. In the first failure, output of channel 2 was replaced with Gaussian white noise. The second failure modeled a channel synchronization problem, inserting 0.5 s of silence at $t = 30.0$ and thus making the channel lag all other channels.

DOA estimation results for the example are given in Fig. 5. Failures were detected at $t = 16.3$ and $t = 33.2$, respectively. After detections, the LSd method using detection produces correct results while LS method not utilizing detection remains largely erroneous. The occasional deviations from apparent speaker direction are caused by pauses in speech. Time delays were estimated with cross correlation, using windows of 4096 samples with 50% overlap.

5. DISCUSSION

How often and what kind of failures will occur in practical situations? This is a difficult question to answer, because exact models of different types of failures and their probabilities would require very specific knowledge on the target application and hardware,

©2004 IEEE. Original version to appear in *Proceedings of the Third IEEE Sensor Array and Multichannel Signal Processing Workshop (SAM 2004)*. Personal use of this material is permitted. However, permission to reprint/ republish this material for advertising or promotional purposes or for creating new collective works for resale or redistribution to servers or lists, or to reuse any copyrighted component of this work in other works must be obtained from the IEEE.

which may be difficult to obtain or completely unavailable.

In addition, exact modelling of failures would be unwise because the objective of the research was to come up with a general and widely applicable method. It is unlikely that this kind of model would include all necessary failure types. In fact, this is the main reason for developing algorithm-level techniques for failure detection, because failure databases can not contain information of all different types of failures that may occur. Of course, if any additional information is available, its use should obviously be considered to improve detection.

However, even with the somewhat simple experiments, the performance and usability of the proposed method has been verified. Although the simulations do not model failures exactly, they show that whenever there is erroneous behavior in time delays the proposed method can be used for failure detection. The speech data example supports these observations.

The increase in computational complexity introduced by the failure detection is low. Mainly, the computational load consists of computing the $\binom{N}{3}$ triangular tests (9) and summing the results into confidence factors (10). In addition there is a minor memory requirement for the failure indication buffer and an indexing table for the triangles. The detection thresholds and the indexing table can be computed beforehand offline. If TDS estimation is used, the triangular tests are required already for the actual estimation and thus the increase in computations is almost negligible. Grid-based DOA estimation is computationally much more demanding than failure detection, and therefore the increase in computational demands should be acceptable. If the computational increases are tolerable, this method can also be used for many other array applications not using time-delay based DOA estimation, such as beamforming.

The proposed method has some critical points and downsides. Several thresholds are required in detection stage and care has to be taken when selecting them. Topics for further research include reducing the number of required thresholds by combining error information more flexibly from different stages. Thresholds should also be parameterized more to make them easy to determine, perhaps automatically. A more fundamental improvement would be to replace simple thresholding with better and more intelligent detection algorithms.

Because planar wavefronts were assumed, performance in near-field situations is a problem. Any significant nonplanarities may result into false alarms. These may also be caused by discontinuous sound activity, especially if there are long periods of silence or uncorrelated input. To access these factors, and also for proper performance analysis, comprehensive real data experiments are needed. However, the speech example is encouraging because the speaker was relatively close to the array and pauses in speech provided some discontinuity, yet the algorithm performed well.

6. CONCLUSIONS

In this paper, a new algorithm-level method for detecting multiple channel failures in a small sensor array was proposed. The performance of the detection method was demonstrated with simulations and real data. Results indicate that the proposed method can effectively detect and remove failed channels from array processing, thus providing failure-robust DOA estimators and a tool for array self-diagnosis. This is an important feature when intelligent sensor systems are considered.

7. REFERENCES

[1] W. A. Rosen and A. D. George, "Fault tolerance in autonomous acoustic arrays," *Journal of the Franklin Institute*, vol. 336, no. 1, pp. 19–32, 1999.

[2] T. Pirinen, P. Pertilä, and A. Visa, "Toward intelligent sensors - reliability for time delay based direction of arrival estimates," in *Proceedings of the 2003 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'03)*, 2003, vol. V, pp. 197–200.

[3] P. Aarabi, "Localization-based sensor validation using the Kullback-Leibler divergence," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 34, no. 2, pp. 1007–1016, 2004.

[4] T. Pirinen, J. Yli-Hietanen, P. Pertilä, and A. Visa, "Detection and compensation of sensor malfunction in time delay based direction of arrival estimation," in *(to appear) Proceedings of the 2004 IEEE International Symposium on Circuits and Systems (ISCAS 2004)*, 2004.

[5] J. C. Chen, Y. Kung, and R.E. Hudson, "Source localization and beamforming," *IEEE Signal Processing Magazine*, vol. 19, no. 2, pp. 30–39, 2002.

[6] M. Brandstein and D. Ward, Eds., *Microphone Arrays*, Springer-Verlag, 2001.

[7] J. Yli-Hietanen, K. Kalliojärvi, and J. Astola, "Low-complexity angle of arrival estimation of wideband signals using small arrays," in *Proceedings of the 8th IEEE Signal Processing Workshop on Statistical Signal and Array Signal Processing*, 1996, pp. 109–112.

[8] K. Varma, T. Ikuma, and A. A. Beex, "Robust TDE-based DOA-estimation for compact audio arrays," in *Proceedings of the Second IEEE Sensor Array and Multichannel Signal Processing Workshop (SAM 2002)*, 2002, pp. 214–218.

[9] D. E. Dudgeon and R. M. Merserau, *Multidimensional digital signal processing*, Prentice-Hall, 1984.

[10] S. Haykin and J. Justice, *Array signal processing*, Academic Press, 1985.

[11] C. Knapp and G. C. Carter, "The generalized correlation method for estimation of time delay," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 24, no. 4, pp. 320–327, 1976.

[12] R. Blumrich and J. Altmann, "Medium range localisation of aircraft via triangulation," *Applied Acoustics*, vol. 61, no. 1, pp. 65–82, 2000.

[13] J. Yli-Hietanen, K. Koppinen, and J. Astola, "Time-delay selection for robust angle of arrival estimation," in *Proceedings of the IASTED International Conference on Signal and Image Processing*, 1999, pp. 81–83.

[14] J. Yli-Hietanen, K. Kalliojärvi, and J. Astola, "Robust time-delay based angle of arrival estimation," in *Proceedings of the 1996 IEEE Nordic Signal Processing Symposium (NORSIG 96)*, 1996, pp. 219–222.

[15] D. Moore, "The IDIAP smart meeting room," Tech. Rep. 02-07, Dalle Molle Institute of Perceptual Artificial Intelligence (IDIAP), 2002.

©2004 IEEE. Original version to appear in *Proceedings of the Third IEEE Sensor Array and Multichannel Signal Processing Workshop (SAM 2004)*. Personal use of this material is permitted. However, permission to reprint/ republish this material for advertising or promotional purposes or for creating new collective works for resale or redistribution to servers or lists, or to reuse any copyrighted component of this work in other works must be obtained from the IEEE.