

Kalman Filters for Time Delay of Arrival-Based Source Localization

Ulrich Klee, Tobias Gehrig, and John McDonough

Institut für Theoretische Informatik, Universität Karlsruhe, Am Fasanengarten 5, 76131 Karlsruhe, Germany

Received 9 February 2005; Revised 13 October 2005; Accepted 17 October 2005

In this work, we propose an algorithm for acoustic source localization based on time delay of arrival (TDOA) estimation. In earlier work by other authors, an initial closed-form approximation was first used to estimate the true position of the speaker followed by a Kalman filtering stage to smooth the time series of estimates. In the proposed algorithm, this closed-form approximation is eliminated by employing a Kalman filter to directly update the speaker's position estimate based on the observed TDOAs. In particular, the TDOAs comprise the observation associated with an extended Kalman filter whose state corresponds to the speaker's position. We tested our algorithm on a data set consisting of seminars held by actual speakers. Our experiments revealed that the proposed algorithm provides source localization accuracy superior to the standard spherical and linear intersection techniques. Moreover, the proposed algorithm, although relying on an iterative optimization scheme, proved efficient enough for real-time operation.

Copyright © 2006 Hindawi Publishing Corporation. All rights reserved.

1. INTRODUCTION

Most practical acoustic source localization schemes are based on *time delay of arrival estimation* (TDOA) for the following reasons: such systems are conceptually simple. They are reasonably effective in moderately reverberant environments. Moreover, their low computational complexity makes them well-suited to real-time implementation with several sensors.

Time delay of arrival-based source localization is based on a two-step procedure.

- (1) The TDOA between all pairs of microphones is estimated, typically by finding the peak in a cross-correlation or *generalized cross-correlation* function [1].
- (2) For a given source location, the squared error is calculated between the estimated TDOAs and those determined from the source location. The estimated source location then corresponds to that position which minimizes this squared error.

If the TDOA estimates are assumed to have a Gaussian-distributed error term, it can be shown that the least-squares metric used in Step (2) provides the maximum likelihood (ML) estimate of the speaker location [2]. Unfortunately, this least-squares criterion results in a nonlinear optimization problem that can have several local minima. Several authors have proposed solving this optimization problem with standard gradient-based iterative techniques. While such

techniques typically yield accurate location estimates, they are typically computationally intensive and thus ill-suited for real-time implementation [3, 4].

For any pair of microphones, the surface on which the TDOA is constant is a hyperboloid of two sheets. A second class of algorithms seeks to exploit this fact by grouping all microphones into pairs, estimating the TDOA of each pair, then finding the point where all associated hyperboloids most nearly intersect. Several closed-form position estimates based on this approach have appeared in the literature; see Chan and Ho [5] and the literature review found there. Unfortunately, the point of intersection of two hyperboloids can change significantly based on a slight change in the eccentricity of one of the hyperboloids. Hence, a third class of algorithms was developed wherein the position estimate is obtained from the intersection of several spheres. The first algorithm in this class was proposed by Schau and Robinson [6], and later came to be known as *spherical intersection*. Perhaps the best-known algorithm from this class is the *spherical interpolation* method of Smith and Abel [7]. Both methods provide closed-form estimates suitable for real-time implementation.

Brandstein et al. [4] proposed yet another closed-form approximation known as *linear intersection*. Their algorithm proceeds by first calculating a bearing line to the source for each pair of sensors. Thereafter, the point of nearest approach is calculated for each pair of bearing lines, yielding a potential

source location. The final position estimate is obtained from a weighted average of these potential source locations.

In the algorithm proposed here, the closed-form approximation used in prior approaches is eliminated by employing an extended Kalman filter to directly update the speaker's position estimate based on the observed TDOAs. In particular, the TDOAs comprise the observation associated with an extended Kalman filter whose state corresponds to the speaker's position. Hence, the new position estimate comes directly from the update formulae of the Kalman filter. It is worth noting that similar approaches have been proposed by Dvorkind and Gannot [8] for an acoustic source localizer, as well as by Duraiswami et al. [9] for a combined audio-video source localization algorithm based on a particle filter.

We are indebted to a reviewer who called our attention to other recent works in which particle filters were applied to the acoustic source localization problem [10, 11]. As explained in the tutorial by Arulampalam et al. [12], particle filters represent a generalization of Kalman filters that can handle nonlinear and non-Gaussian state estimation problems. This is certainly a desirable characteristic, and makes particle filters of interest for future study. It remains to be seen, however, whether particle filters will prove better-suited for acoustic source localization than the extended Kalman filters considered here. To wit, Arulampalam et al. [12] discuss several problems that can arise with the use of particle filters, namely, *degeneracy* and *sample impoverishment*. While solutions for circumventing these problems have appeared in the literature, the application of a particle filter to a tracking problem clearly requires a certain amount of engineering to obtain a working system, much as with our approach based on the Kalman filter. Moreover, it is not clear that the assumptions inherent in the Kalman filter, namely, linearity and Gaussianity, make it unsuitable for the speaker tracking problem: Hahn and Tretter [13] show that the observation noise encountered in time delay of arrival estimation is in fact Gaussian, as required by a Kalman filter. Moreover, as shown here, the nonlinearity seen in the acoustic source localization problem is relatively mild and can be adequately handled by performing several local iterations for each time step as explained in [14]. Such theoretical considerations, notwithstanding, the question of whether Kalman or particle filters are better suited for speaker tracking, will only be answered by empirical studies. We believe that such studies should be conducted on real, rather than simulated, data such as we have used for the experiments discussed in Section 5, as only results obtained on real data will be truly compelling. We hope to make such empirical comparisons the topic of a future publication.

The balance of this work is organized as follows. In Section 2, we review the process of source localization based on time delay of arrival estimation. In particular, we formulate source localization as a problem in nonlinear least-squares estimation, then develop an appropriate linearized model. Section 3 summarizes the standard and extended Kalman filters. It also presents a less well-known variant known as the iterated extended Kalman filter. Section 4 first discusses two possible models for speaker motion, then

discusses how the preceding development can be combined to develop an acoustic localization algorithm capable of tracking a moving speaker. Section 5 presents the results of our initial experiments comparing the proposed algorithm to the standard techniques. Section 6 presents our conclusions and plans for future work. The appendix presents a numerically stable implementation of the Kalman filtering algorithms discussed in this work that is based on the Cholesky decomposition.

2. SOURCE LOCALIZATION

Consider a sensor array consisting of several pairs of microphones. Let \mathbf{m}_{i1} and \mathbf{m}_{i2} , respectively, denote the positions of the first and second microphones in the i th pair, and let $\mathbf{x} \in \mathbf{R}^3$ denote the position of the speaker. Then the *time delay of arrival* (TDOA) between the microphones can be expressed as

$$T(\mathbf{m}_{i1}, \mathbf{m}_{i2}, \mathbf{x}) = \frac{\|\mathbf{x} - \mathbf{m}_{i1}\| - \|\mathbf{x} - \mathbf{m}_{i2}\|}{s}, \quad (1)$$

where s is the speed of sound. Denoting

$$\mathbf{x} = \begin{bmatrix} x \\ y \\ z \end{bmatrix}, \quad \mathbf{m}_{ij} = \begin{bmatrix} m_{ij,x} \\ m_{ij,y} \\ m_{ij,z} \end{bmatrix} \quad (2)$$

allows (1) to be rewritten as

$$T_i(\mathbf{x}) = T(\mathbf{m}_{i1}, \mathbf{m}_{i2}, \mathbf{x}) = \frac{1}{s}(d_{i1} - d_{i2}), \quad (3)$$

where

$$\begin{aligned} d_{ij} &= \sqrt{(x - m_{ij,x})^2 + (y - m_{ij,y})^2 + (z - m_{ij,z})^2} \\ &= \|\mathbf{x} - \mathbf{m}_{ij}\| \end{aligned} \quad (4)$$

is the distance from the source to microphone \mathbf{m}_{ij} . Source localization based on a maximum likelihood (ML) criterion [2] proceeds by minimizing the error function

$$\epsilon(\mathbf{x}) = \sum_{i=0}^{N-1} \frac{1}{\sigma_i^2} [\hat{\tau}_i - T_i(\mathbf{x})]^2, \quad (5)$$

where $\hat{\tau}_i$ is the observed TDOA for the i th microphone pair and σ_i^2 is the error covariance associated with this observation. The TDOAs can be estimated with a variety of well-known techniques [1, 15]. Perhaps the most popular method involves the *phase transform* (PHAT), a variant of the generalized cross-correlation (GCC), which can be expressed as

$$R_{12}(\tau) = \frac{1}{2\pi} \int_{-\pi}^{\pi} \frac{X_1(e^{j\omega\tau})X_2^*(e^{j\omega\tau})}{|X_1(e^{j\omega\tau})X_2^*(e^{j\omega\tau})|} e^{j\omega\tau} d\omega. \quad (6)$$

For reasons of computational efficiency, $R_{12}(\tau)$ is typically calculated with an inverse FFT. Thereafter, an interpolation is performed to overcome the granularity in the estimate corresponding to the sampling interval [1].

Solving for that \mathbf{x} minimizing (5) would be eminently straightforward were it not for the fact that (3) is nonlinear in $\mathbf{x} = (x, y, z)$. In the coming development, we will find it useful to have a linear approximation. Hence, we take a partial derivative with respect to x on both sides of (3) and write

$$\frac{\partial T_i(\mathbf{x})}{\partial x} = \frac{1}{s} \cdot \left[\frac{x - m_{i1,x}}{d_{i1}} - \frac{x - m_{i2,x}}{d_{i2}} \right]. \quad (7)$$

Taking partial derivatives with respect to y and z similarly, we find

$$\nabla_{\mathbf{x}} T_i(\mathbf{x}) = \frac{1}{s} \cdot \left[\frac{\mathbf{x} - \mathbf{m}_{i1}}{d_{i1}} - \frac{\mathbf{x} - \mathbf{m}_{i2}}{d_{i2}} \right]. \quad (8)$$

Although (5) implies we should find that \mathbf{x} which minimizes the instantaneous error criterion, we would be better advised to attempt to minimize such an error criterion over a *series* of time instants. In so doing, we exploit the fact that the speaker's position cannot change instantaneously, thus, both the present $\hat{\tau}_i(t)$ and past TDOA estimates $\{\hat{\tau}_i(n)\}_{n=0}^{t-1}$ are potentially useful in estimating a speaker's current position $\mathbf{x}(t)$. Hence, let us approximate $T_i(\mathbf{x})$ with a first-order Taylor series expansion about the last position estimate $\hat{\mathbf{x}}(t-1)$ by writing

$$T_i(\mathbf{x}) \approx T_i(\hat{\mathbf{x}}(t-1)) + \mathbf{c}_i^T(t)[\mathbf{x} - \hat{\mathbf{x}}(t-1)], \quad (9)$$

where we have defined the row vector

$$\begin{aligned} \mathbf{c}_i^T(t) &= [\nabla_{\mathbf{x}} T_i(\mathbf{x})]_{\mathbf{x}=\hat{\mathbf{x}}(t-1)}^T \\ &= \frac{1}{s} \cdot \left[\frac{\mathbf{x} - \mathbf{m}_{i1}}{d_{i1}} - \frac{\mathbf{x} - \mathbf{m}_{i2}}{d_{i2}} \right]_{\mathbf{x}=\hat{\mathbf{x}}(t-1)}^T. \end{aligned} \quad (10)$$

Substituting the linearization (9) into (5) provides

$$\begin{aligned} \epsilon(\mathbf{x}; t) &\approx \sum_{i=0}^{N-1} \frac{1}{\sigma_i^2} \{ \hat{\tau}_i(t) - T_i(\hat{\mathbf{x}}(t-1)) - \mathbf{c}_i^T(t)[\mathbf{x} - \hat{\mathbf{x}}(t-1)] \}^2 \\ &= \sum_{i=0}^{N-1} \frac{1}{\sigma_i^2} [\bar{\tau}_i(t) - \mathbf{c}_i^T(t)\mathbf{x}]^2, \end{aligned} \quad (11)$$

where

$$\bar{\tau}_i(t) = \hat{\tau}_i(t) - T_i(\hat{\mathbf{x}}(t-1)) + \mathbf{c}_i^T(t)\hat{\mathbf{x}}(t-1), \quad (12)$$

for $i = 0, \dots, N-1$. Let us define

$$\begin{aligned} \bar{\boldsymbol{\tau}}(t) &= \begin{bmatrix} \bar{\tau}_0(t) \\ \bar{\tau}_1(t) \\ \vdots \\ \bar{\tau}_{N-1}(t) \end{bmatrix}, & \hat{\boldsymbol{\tau}}(t) &= \begin{bmatrix} \hat{\tau}_0(t) \\ \hat{\tau}_1(t) \\ \vdots \\ \hat{\tau}_{N-1}(t) \end{bmatrix}, \\ \mathbf{T}(\hat{\mathbf{x}}(t)) &= \begin{bmatrix} T_0(\hat{\mathbf{x}}(t)) \\ T_1(\hat{\mathbf{x}}(t)) \\ \vdots \\ T_{N-1}(\hat{\mathbf{x}}(t)) \end{bmatrix}, & \mathbf{C}(t) &= \begin{bmatrix} \mathbf{c}_0^T(t) \\ \mathbf{c}_1^T(t) \\ \vdots \\ \mathbf{c}_{N-1}^T(t) \end{bmatrix} \end{aligned} \quad (13)$$

so that (12) can be expressed in matrix form as

$$\bar{\boldsymbol{\tau}}(t) = \hat{\boldsymbol{\tau}}(t) - [\mathbf{T}(\hat{\mathbf{x}}(t-1)) - \mathbf{C}(t)\hat{\mathbf{x}}(t-1)]. \quad (14)$$

Similarly, defining

$$\boldsymbol{\Sigma} = \begin{bmatrix} \sigma_0^2 & & & \\ & \sigma_1^2 & & \\ & & \ddots & \\ & & & \sigma_{N-1}^2 \end{bmatrix} \quad (15)$$

enables (11) to be expressed as

$$\epsilon(\mathbf{x}; t) = [\bar{\boldsymbol{\tau}}(t) - \mathbf{C}(t)\mathbf{x}]^T \boldsymbol{\Sigma}^{-1} [\bar{\boldsymbol{\tau}}(t) - \mathbf{C}(t)\mathbf{x}]. \quad (16)$$

In past work, the criterion in (5) was minimized for each time instant t , typically with a closed-form approximation [4–7, 16]. Thereafter, some authors have proposed using a Kalman filter to smooth the position estimates over time [17]. In this work, we propose to incorporate the smoothing stage directly into the estimation. This is accomplished as follows: first we note that (16) represents a *nonlinear* least-squares estimation problem that has been appropriately linearized; we can associate $\hat{\boldsymbol{\tau}}(t)$ with the *observation* vector appearing in a Kalman filter such as we will encounter in Section 3. Moreover, we can define a model for the motion of the speaker, in the form typically seen in the *process equation* of a Kalman filter. Thereafter, we can apply the standard Kalman filter update formulae directly to the given recursive estimation problem without ever having recourse to a closed-form approximation for the speaker's position. It is worth noting that similar approaches have been proposed by Dvorkind and Gannot [8] for an acoustic source localizer, as well as by Duraiswami et al. [9] for a combined audio-video source localization algorithm based on a particle filter.

To see more clearly how this approach can be implemented, we review the Kalman filter and several variations thereof in Section 3.

3. KALMAN FILTERING

To set the stage for the development to follow, this section summarizes the Kalman filter based on the Riccati equation, as well as the extended Kalman filter.

3.1. Riccati-based Kalman filter

Our purpose here is to present, without proof, the principal quantities and equations required to implement a Kalman filter based on the Riccati equation. Let $\mathbf{x}(t)$ denote the current state of a Kalman filter and $\mathbf{y}(t)$ the current observation. Normally, $\mathbf{x}(t)$ cannot be observed directly and thus must be inferred from the times series $\{\mathbf{y}(t)\}_t$; this is the primary function of the Kalman filter. The operation of the Kalman filter is governed by a *state space model* consisting of a *process* and an *observation* equation, respectively,

$$\mathbf{x}(t+1) = \mathbf{F}(t+1, t)\mathbf{x}(t) + \nu_1(t), \quad (17)$$

$$\mathbf{y}(t) = \mathbf{C}(t)\mathbf{x}(t) + \nu_2(t), \quad (18)$$

where $\mathbf{F}(t+1, t)$ and $\mathbf{C}(t)$ are the known *transition* and *observation* matrices. By definition, the transition matrix satisfies

$$\mathbf{F}(t+1, t)\mathbf{F}(t, t+1) = \mathbf{F}(t, t+1)\mathbf{F}(t+1, t) = \mathbf{I}. \quad (19)$$

In (17)–(18), the *process* and *observation noise* terms are denoted by $\nu_1(t)$ and $\nu_2(t)$, respectively. These noise terms are by assumption zero mean, white Gaussian random vector processes with covariance matrices defined by

$$\mathcal{E}\{\nu_i(t)\nu_i^T(k)\} = \begin{cases} \mathbf{Q}_i(t) & \text{for } t = k, \\ \mathbf{0} & \text{otherwise,} \end{cases} \quad (20)$$

for $i = 1, 2$. Moreover, $\nu_1(t)$ and $\nu_2(k)$ are statistically independent such that $\mathcal{E}\{\nu_1(t)\nu_2^T(k)\} = \mathbf{0}$ for all t and k .

In the sequel, it will prove useful to define two estimates of the current state $\mathbf{x}(t)$: let $\hat{\mathbf{x}}(t | \mathcal{Y}_{t-1})$ denote the *predicted state estimate* of $\mathbf{x}(t)$ obtained from all observations $\mathcal{Y}_{t-1} = \{\mathbf{y}(i)\}_{i=0}^{t-1}$ up to time $t-1$. The *filtered state estimate* $\hat{\mathbf{x}}(t | \mathcal{Y}_t)$, on the other hand, is based on all observations $\mathcal{Y}_t = \{\mathbf{y}(i)\}_{i=0}^t$ up to time t . The *predicted observation* is then given by

$$\hat{\mathbf{y}}(t | \mathcal{Y}_{t-1}) = \mathbf{C}(t)\hat{\mathbf{x}}(t | \mathcal{Y}_{t-1}) \quad (21)$$

as can be readily derived from (17). By definition, the *innovation* is the difference

$$\boldsymbol{\alpha}(t) = \mathbf{y}(t) - \mathbf{C}(t)\hat{\mathbf{x}}(t | \mathcal{Y}_{t-1}) \quad (22)$$

between actual and predicted observations. Exploiting the statistical independence of $\nu_1(t)$ and $\nu_2(t)$, the correlation matrix of the innovations sequence can be expressed as

$$\begin{aligned} \mathbf{R}(t) &= \mathcal{E}\{\boldsymbol{\alpha}(t)\boldsymbol{\alpha}^T(t)\} \\ &= \mathbf{C}(t)\mathbf{K}(t, t-1)\mathbf{C}^T(t) + \mathbf{Q}_2(t), \end{aligned} \quad (23)$$

where

$$\mathbf{K}(t, t-1) = \mathcal{E}\{\boldsymbol{\epsilon}(t, t-1)\boldsymbol{\epsilon}^T(t, t-1)\} \quad (24)$$

is the correlation matrix of the *predicted state error*,

$$\boldsymbol{\epsilon}(t, t-1) = \mathbf{x}(t) - \hat{\mathbf{x}}(t | \mathcal{Y}_{t-1}). \quad (25)$$

The *Kalman gain* is defined as

$$\mathbf{G}(t) = \mathcal{E}\{\mathbf{x}(t+1)\boldsymbol{\alpha}^T(t)\}\mathbf{R}^{-1}(t). \quad (26)$$

This definition can be readily shown to be equivalent to

$$\mathbf{G}(t) = \mathbf{F}(t+1, t)\mathbf{K}(t, t-1)\mathbf{C}^T(t)\mathbf{R}^{-1}(t). \quad (27)$$

To calculate $\mathbf{G}(t)$, we must know $\mathbf{K}(t, t-1)$ in advance. The latter is available from the *Riccati equation*, which can be stated as

$$\mathbf{K}(t+1, t) = \mathbf{F}(t+1, t)\mathbf{K}(t)\mathbf{F}^T(t+1, t) + \mathbf{Q}_1(t), \quad (28)$$

$$\mathbf{K}(t) = [\mathbf{I} - \mathbf{F}(t, t+1)\mathbf{G}(t)\mathbf{C}(t)]\mathbf{K}(t, t-1), \quad (29)$$

where

$$\mathbf{K}(t) = \mathcal{E}\{\boldsymbol{\epsilon}(t)\boldsymbol{\epsilon}^T(t)\} \quad (30)$$

TABLE 1: Calculations for Kalman filter based on the Riccati equation.

<p><i>Input vector process:</i> $\mathbf{y}(1), \mathbf{y}(2), \dots, \mathbf{y}(t)$.</p> <p><i>Known parameters:</i></p> <ul style="list-style-type: none"> (i) state transition matrix: $\mathbf{F}(t+1, t)$, (ii) measurement matrix: $\mathbf{C}(t)$, (iii) covariance matrix of process noise: $\mathbf{Q}_1(t)$, (iv) covariance matrix of measurement noise: $\mathbf{Q}_2(t)$, (v) initial diagonal loading: σ_D^2. <p><i>Initial conditions:</i></p> $\hat{\mathbf{x}}(1 \mathcal{Y}_0) = \mathbf{x}_0,$ $\mathbf{K}(1, 0) = \frac{1}{\sigma_D^2}\mathbf{I}. \quad (36)$ <p><i>Computation:</i> $t = 1, 2, 3, \dots$</p> $\begin{aligned} \mathbf{R}(t) &= \mathbf{C}(t)\mathbf{K}(t, t-1)\mathbf{C}^T(t) + \mathbf{Q}_2(t), \\ \mathbf{G}(t) &= \mathbf{F}(t+1, t)\mathbf{K}(t, t-1)\mathbf{C}^T(t)\mathbf{R}^{-1}(t), \\ \boldsymbol{\alpha}(t) &= \mathbf{y}(t) - \mathbf{C}(t)\hat{\mathbf{x}}(t \mathcal{Y}_{t-1}), \\ \hat{\mathbf{x}}(t+1 \mathcal{Y}_t) &= \mathbf{F}(t+1, t)\hat{\mathbf{x}}(t \mathcal{Y}_{t-1}) + \mathbf{G}(t)\boldsymbol{\alpha}(t), \\ \mathbf{K}(t) &= [\mathbf{I} - \mathbf{F}(t, t+1)\mathbf{G}(t)\mathbf{C}(t)]\mathbf{K}(t, t-1), \\ \mathbf{K}(t+1, t) &= \mathbf{F}(t+1, t)\mathbf{K}(t)\mathbf{F}^T(t+1, t) + \mathbf{Q}_1(t). \end{aligned} \quad (37)$
--

is the correlation matrix of the *filtered state error*,

$$\boldsymbol{\epsilon}(t) = \mathbf{x}(t) - \hat{\mathbf{x}}(t | \mathcal{Y}_t). \quad (31)$$

Finally, the filtered state estimate can be updated based on the Kalman gain $\mathbf{G}(t)$ and innovation $\boldsymbol{\alpha}(t)$ according to

$$\hat{\mathbf{x}}(t+1 | \mathcal{Y}_t) = \mathbf{F}(t+1, t)\hat{\mathbf{x}}(t | \mathcal{Y}_{t-1}) + \mathbf{G}(t)\boldsymbol{\alpha}(t). \quad (32)$$

These relations are summarized in Table 1.

We have now formulated the *one-step prediction* form of the Kalman filter, which returns the predicted state estimate $\hat{\mathbf{x}}(t+1 | \mathcal{Y}_t)$. In Section 3.2, we will require the filtered state estimate $\hat{\mathbf{x}}(t | \mathcal{Y}_t)$, which can be obtained as follows. Taking an expectation conditioned on \mathcal{Y}_t on both sides of (17), we can write

$$\hat{\mathbf{x}}(t+1 | \mathcal{Y}_t) = \mathbf{F}(t+1, t)\hat{\mathbf{x}}(t | \mathcal{Y}_t) + \hat{\nu}_1(t | \mathcal{Y}_t). \quad (33)$$

As the process noise is zero mean, we have $\hat{\nu}_1(t | \mathcal{Y}_t) = \mathbf{0}$, so that (33) becomes

$$\hat{\mathbf{x}}(t+1 | \mathcal{Y}_t) = \mathbf{F}(t+1, t)\hat{\mathbf{x}}(t | \mathcal{Y}_t). \quad (34)$$

To obtain the desired filtered estimate, we multiply both sides of (34) by $\mathbf{F}(t | t+1)$ and invoke (19) and write

$$\hat{\mathbf{x}}(t | \mathcal{Y}_t) = \mathbf{F}(t, t+1)\hat{\mathbf{x}}(t+1 | \mathcal{Y}_t). \quad (35)$$

3.2. Extended Kalman filter (EKF)

For the sake of completeness, we provide here a brief derivation of the general extended Kalman filter (EKF). This development is based on that in Haykin [18, Section 10.10].

To begin, let us split the filtered state estimate update formula (35) into two steps. Firstly, we make a one-step prediction to update $\hat{\mathbf{x}}(t-1 | \mathcal{Y}_{t-1})$ to $\hat{\mathbf{x}}(t | \mathcal{Y}_{t-1})$, which is achieved by (34). Secondly, we update $\hat{\mathbf{x}}(t | \mathcal{Y}_{t-1})$ to $\hat{\mathbf{x}}(t | \mathcal{Y}_t)$, which is achieved by substituting (32) into (35) and using (19) to simplify

$$\hat{\mathbf{x}}(t | \mathcal{Y}_t) = \hat{\mathbf{x}}(t | \mathcal{Y}_{t-1}) + \mathbf{G}_F(t)\boldsymbol{\alpha}(t), \quad (38)$$

where the *filtered Kalman gain* $\mathbf{G}_F(t)$ is defined as

$$\mathbf{G}_F(t) = \mathbf{F}(t, t+1)\mathbf{G}(t). \quad (39)$$

The complete filtering algorithm is then

$$\mathbf{R}(t) = \mathbf{C}(t)\mathbf{K}(t, t-1)\mathbf{C}^T(t) + \mathbf{Q}_2(t), \quad (40)$$

$$\mathbf{G}_F(t) = \mathbf{K}(t, t-1)\mathbf{C}^T(t)\mathbf{R}^{-1}(t), \quad (41)$$

$$\boldsymbol{\alpha}(t) = \mathbf{y}(t) - \mathbf{C}(t)\hat{\mathbf{x}}(t | \mathcal{Y}_{t-1}), \quad (42)$$

$$\hat{\mathbf{x}}(t | \mathcal{Y}_t) = \hat{\mathbf{x}}(t | \mathcal{Y}_{t-1}) + \mathbf{G}_F(t)\boldsymbol{\alpha}(t), \quad (43)$$

$$\mathbf{K}(t) = [\mathbf{I} - \mathbf{G}_F(t)\mathbf{C}(t)]\mathbf{K}(t, t-1), \quad (44)$$

$$\mathbf{K}(t+1, t) = \mathbf{F}(t+1, t)\mathbf{K}(t)\mathbf{F}^T(t+1, t) + \mathbf{Q}_1(t), \quad (45)$$

$$\hat{\mathbf{x}}(t+1 | \mathcal{Y}_t) = \mathbf{F}(t+1, t)\hat{\mathbf{x}}(t | \mathcal{Y}_t). \quad (46)$$

To formulate the extended Kalman filter, we first posit a less restrictive state-space model, namely,

$$\begin{aligned} \mathbf{x}(t+1) &= \mathbf{F}(t+1, t)\mathbf{x}(t) + \nu_1(t), \\ \mathbf{y}(t) &= \mathbf{C}(t, \mathbf{x}(t)) + \nu_2(t), \end{aligned} \quad (47)$$

where the observation *functional*¹ $\mathbf{C}(t, \mathbf{x}(t))$ is, in general, nonlinear and time varying. The main idea behind the EKF is then to linearize this functional about the most recent state estimate $\hat{\mathbf{x}}(t | \mathcal{Y}_{t-1})$. The corresponding linearization can be written as

$$\mathbf{C}(t) = \left. \frac{\partial \mathbf{C}(t, \mathbf{x})}{\partial \mathbf{x}} \right|_{\mathbf{x}=\hat{\mathbf{x}}(t | \mathcal{Y}_{t-1})}. \quad (48)$$

In the above, entry (i, j) of $\mathbf{C}(t, \mathbf{x})$ is the partial derivative of the i th component of $\mathbf{C}(t, \mathbf{x})$ with respect to the j th component of \mathbf{x} .

Based on (48), we can express the first-order Taylor series of $\mathbf{C}(t, \mathbf{x}(t))$ as

$$\mathbf{C}(t, \mathbf{x}(t)) \approx \mathbf{C}(t, \hat{\mathbf{x}}(t | \mathcal{Y}_{t-1})) + \mathbf{C}(t)[\mathbf{x}(t) - \hat{\mathbf{x}}(t | \mathcal{Y}_{t-1})]. \quad (49)$$

¹ Most authors formulate the extended Kalman filter with a nonlinear process functional $\mathbf{F}(t, \mathbf{x}(t))$ in addition to the observation functional $\mathbf{C}(t, \mathbf{x}(t))$; see, for example, Haykin [18, Section 10.10]. This more general formulation is not required here.

TABLE 2: Calculations for extended Kalman filter.

Input vector process: $\mathbf{y}(1), \mathbf{y}(2), \dots, \mathbf{y}(t)$.

Known parameters:

- (i) state transition matrix: $\mathbf{F}(t+1, t)$,
- (ii) nonlinear measurement functional: $\mathbf{C}(t, \mathbf{x}(t))$,
- (iii) covariance matrix of process noise: $\mathbf{Q}_1(t)$,
- (iv) covariance matrix of measurement noise: $\mathbf{Q}_2(t)$,
- (v) initial diagonal loading: σ_D^2 .

Initial conditions:

$$\begin{aligned} \hat{\mathbf{x}}(1 | \mathcal{Y}_0) &= \mathbf{x}_0, \\ \mathbf{K}(1, 0) &= \frac{1}{\sigma_D^2} \mathbf{I}. \end{aligned} \quad (53)$$

Computation: $t = 1, 2, 3, \dots$,

$$\mathbf{R}(t) = \mathbf{C}(t)\mathbf{K}(t, t-1)\mathbf{C}^T(t) + \mathbf{Q}_2(t), \quad (54)$$

$$\mathbf{G}_F(t) = \mathbf{K}(t, t-1)\mathbf{C}^T(t)\mathbf{R}^{-1}(t), \quad (55)$$

$$\boldsymbol{\alpha}(t) = \mathbf{y}(t) - \mathbf{C}(t, \hat{\mathbf{x}}(t | \mathcal{Y}_{t-1})), \quad (56)$$

$$\hat{\mathbf{x}}(t | \mathcal{Y}_t) = \hat{\mathbf{x}}(t | \mathcal{Y}_{t-1}) + \mathbf{G}_F(t)\boldsymbol{\alpha}(t), \quad (57)$$

$$\mathbf{K}(t) = [\mathbf{I} - \mathbf{G}_F(t)\mathbf{C}(t)]\mathbf{K}(t, t-1), \quad (58)$$

$$\mathbf{K}(t+1, t) = \mathbf{F}(t+1, t)\mathbf{K}(t)\mathbf{F}^T(t+1, t) + \mathbf{Q}_1(t), \quad (59)$$

$$\hat{\mathbf{x}}(t+1 | \mathcal{Y}_t) = \mathbf{F}(t+1, t)\hat{\mathbf{x}}(t | \mathcal{Y}_t). \quad (60)$$

Note: the linearized matrix $\mathbf{C}(t)$ is computed from the nonlinear functional $\mathbf{C}(t, \mathbf{x}(t))$ as in (48).

Using this linearization, the nonlinear state-space equations (47) can be written as

$$\begin{aligned} \mathbf{x}(t+1) &= \mathbf{F}(t+1, t)\mathbf{x}(t) + \nu_1(t), \\ \bar{\mathbf{y}}(t) &\approx \mathbf{C}(t)\mathbf{x}(t) + \nu_2(t), \end{aligned} \quad (50)$$

where we have defined

$$\bar{\mathbf{y}}(t) = \mathbf{y}(t) - [\mathbf{C}(t, \hat{\mathbf{x}}(t | \mathcal{Y}_{t-1})) - \mathbf{C}(t)\hat{\mathbf{x}}(t | \mathcal{Y}_{t-1})]. \quad (51)$$

As everything on the right-hand side of (51) is known at time t , $\bar{\mathbf{y}}(t)$ can be regarded as an observation.

The extended Kalman filter is obtained by applying the computations in (40)–(46) to the linearized model in (50), whereupon we find

$$\begin{aligned} \hat{\mathbf{x}}(t+1 | \mathcal{Y}_t) &= \mathbf{F}(t+1, t)\hat{\mathbf{x}}(t | \mathcal{Y}_t), \\ \hat{\mathbf{x}}(t | \mathcal{Y}_t) &= \hat{\mathbf{x}}(t | \mathcal{Y}_{t-1}) + \mathbf{G}_F(t)\boldsymbol{\alpha}(t), \\ \boldsymbol{\alpha}(t) &= \bar{\mathbf{y}}(t) - \mathbf{C}(t)\hat{\mathbf{x}}(t | \mathcal{Y}_{t-1}) \\ &= \mathbf{y}(t) - \mathbf{C}(t, \hat{\mathbf{x}}(t | \mathcal{Y}_{t-1})). \end{aligned} \quad (52)$$

These computations are summarized in Table 2.

3.3. Iterated extended Kalman filter (IEKF)

We now consider a further refinement of the extended Kalman filter. Repeating (54)–(57) of Table 2, we can write

$$\mathbf{R}(t, \hat{\mathbf{x}}(t | \mathcal{Y}_{t-1})) = \mathbf{C}(t)\mathbf{K}(t, t-1)\mathbf{C}^T(t) + \mathbf{Q}_2(t), \quad (61)$$

$$\mathbf{G}_F(t, \hat{\mathbf{x}}(t | \mathcal{Y}_{t-1})) = \mathbf{K}(t, t-1)\mathbf{C}^T(t, \hat{\mathbf{x}}(t | \mathcal{Y}_{t-1})) \times \mathbf{R}^{-1}(t, \hat{\mathbf{x}}(t | \mathcal{Y}_{t-1})), \quad (62)$$

$$\boldsymbol{\alpha}(t, \hat{\mathbf{x}}(t | \mathcal{Y}_{t-1})) = \mathbf{y}(t) - \mathbf{C}(t, \hat{\mathbf{x}}(t | \mathcal{Y}_{t-1})), \quad (63)$$

$$\hat{\mathbf{x}}(t | \mathcal{Y}_t) = \hat{\mathbf{x}}(t | \mathcal{Y}_{t-1}) + \mathbf{G}_F(t, \hat{\mathbf{x}}(t | \mathcal{Y}_{t-1})) \times \boldsymbol{\alpha}(t, \hat{\mathbf{x}}(t | \mathcal{Y}_{t-1})), \quad (64)$$

where we have explicitly indicated the dependence of the relevant quantities on $\hat{\mathbf{x}}(t | \mathcal{Y}_{t-1})$. Jazwinski [14, Section 8.3] describes an *iterated extended Kalman filter* (IEKF), in which (61)–(64) are replaced with the *local iteration*,

$$\begin{aligned} \mathbf{R}(t, \boldsymbol{\eta}_i) &= \mathbf{C}(\boldsymbol{\eta}_i)\mathbf{K}(t, t-1)\mathbf{C}^T(\boldsymbol{\eta}_i) + \mathbf{Q}_2(t), \\ \mathbf{G}_F(t, \boldsymbol{\eta}_i) &= \mathbf{K}(t, t-1)\mathbf{C}^T(\boldsymbol{\eta}_i)\mathbf{R}^{-1}(t, \boldsymbol{\eta}_i), \\ \boldsymbol{\alpha}(t, \boldsymbol{\eta}_i) &= \mathbf{y}(t) - \mathbf{C}(t, \boldsymbol{\eta}_i), \end{aligned} \quad (65)$$

$$\begin{aligned} \boldsymbol{\zeta}(t, \boldsymbol{\eta}_i) &= \boldsymbol{\alpha}(t, \boldsymbol{\eta}_i) - \mathbf{C}(\boldsymbol{\eta}_i)[\hat{\mathbf{x}}(t | \mathcal{Y}_{t-1}) - \boldsymbol{\eta}_i], \\ \boldsymbol{\eta}_{i+1} &= \hat{\mathbf{x}}(t | \mathcal{Y}_{t-1}) + \mathbf{G}_F(t, \boldsymbol{\eta}_i)\boldsymbol{\zeta}(t, \boldsymbol{\eta}_i), \end{aligned}$$

where $\mathbf{C}(\boldsymbol{\eta}_i)$ is the linearization of $\mathbf{C}(t, \boldsymbol{\eta}_i)$ about $\boldsymbol{\eta}_i$. The local iteration is initialized by setting

$$\boldsymbol{\eta}_1 = \hat{\mathbf{x}}(t | \mathcal{Y}_{t-1}). \quad (66)$$

Note that $\boldsymbol{\eta}_2 = \hat{\mathbf{x}}(t | \mathcal{Y}_t)$ as defined in (64). Hence, if the local iteration is run only once, the IEKF reduces to the EKF. Normally (65) are repeated, however, until there are no substantial changes between $\boldsymbol{\eta}_i$ and $\boldsymbol{\eta}_{i+1}$. Both $\mathbf{G}_F(t, \boldsymbol{\eta}_i)$ and $\mathbf{C}(\boldsymbol{\eta}_i)$ are updated for each local iteration. After the last iteration, we set

$$\hat{\mathbf{x}}(t | \mathcal{Y}_t) = \boldsymbol{\eta}_f \quad (67)$$

and this value is used to update $\mathbf{K}(t)$ and $\mathbf{K}(t+1, t)$. Jazwinski [14, Section 8.3] reports that the IEKF provides faster convergence in the presence of significant nonlinearities in the observation equation, especially when the initial state estimate $\boldsymbol{\eta}_1 = \hat{\mathbf{x}}(t | \mathcal{Y}_{t-1})$ is far from the optimal value. The calculations for the iterated extended Kalman filter are summarized in Table 3.

3.4. Numerical stability

All variants of the Kalman filter discussed in Sections 3.1–3.3 are based on the Riccati equation (28)–(29). Unfortunately, the Riccati equation possesses poor numerical stability properties [18, Section 11] as can be seen from the following: substituting (29) into (28) and making use of (19) provide

$$\begin{aligned} \mathbf{K}(t+1, t) &= \mathbf{F}(t+1, t)\mathbf{K}(t, t-1)\mathbf{F}^T(t+1, t) \\ &\quad - \mathbf{G}(t)\mathbf{C}(t)\mathbf{K}(t, t-1)\mathbf{F}^T(t+1, t) + \mathbf{Q}_1(t). \end{aligned} \quad (79)$$

TABLE 3: Calculations for the iterated extended Kalman filter.

Input vector process: $\mathbf{y}(1), \mathbf{y}(2), \dots, \mathbf{y}(t)$.

Known parameters:

- (i) state transition matrix: $\mathbf{F}(t+1, t)$,
- (ii) nonlinear measurement functional: $\mathbf{C}(t, \mathbf{x}(t))$,
- (iii) covariance matrix of process noise: $\mathbf{Q}_1(t)$,
- (iv) covariance matrix of measurement noise: $\mathbf{Q}_2(t)$,
- (v) initial diagonal loading: σ_D^2 .

Initial conditions:

$$\begin{aligned} \hat{\mathbf{x}}(1 | \mathcal{Y}_0) &= \mathbf{x}_0, \\ \mathbf{K}(1, 0) &= \frac{1}{\sigma_D^2} \mathbf{I}. \end{aligned} \quad (68)$$

Computation: $t = 1, 2, 3, \dots$,

$$\boldsymbol{\eta}_1 = \hat{\mathbf{x}}(t | \mathcal{Y}_{t-1}). \quad (69)$$

Iterate: $i = 1, 2, 3, \dots, f-1$,

$$\mathbf{R}(t, \boldsymbol{\eta}_i) = \mathbf{C}(\boldsymbol{\eta}_i)\mathbf{K}(t, t-1)\mathbf{C}^T(\boldsymbol{\eta}_i) + \mathbf{Q}_2(t), \quad (70)$$

$$\mathbf{G}_F(t, \boldsymbol{\eta}_i) = \mathbf{K}(t, t-1)\mathbf{C}^T(\boldsymbol{\eta}_i)\mathbf{R}^{-1}(t, \boldsymbol{\eta}_i), \quad (71)$$

$$\boldsymbol{\alpha}(t, \boldsymbol{\eta}_i) = \mathbf{y}(t) - \mathbf{C}(t, \boldsymbol{\eta}_i), \quad (72)$$

$$\boldsymbol{\zeta}(t, \boldsymbol{\eta}_i) = \boldsymbol{\alpha}(t, \boldsymbol{\eta}_i) - \mathbf{C}(\boldsymbol{\eta}_i)[\hat{\mathbf{x}}(t | \mathcal{Y}_{t-1}) - \boldsymbol{\eta}_i], \quad (73)$$

$$\boldsymbol{\eta}_{i+1} = \hat{\mathbf{x}}(t | \mathcal{Y}_{t-1}) + \mathbf{G}_F(t, \boldsymbol{\eta}_i)\boldsymbol{\zeta}(t, \boldsymbol{\eta}_i). \quad (74)$$

Calculate:

$$\hat{\mathbf{x}}(t | \mathcal{Y}_t) = \boldsymbol{\eta}_f, \quad (75)$$

$$\mathbf{K}(t) = [\mathbf{I} - \mathbf{G}_F(t, \hat{\mathbf{x}}(t | \mathcal{Y}_t))\mathbf{C}(\mathbf{x}(t | \mathcal{Y}_t))]\mathbf{K}(t, t-1), \quad (76)$$

$$\mathbf{K}(t+1, t) = \mathbf{F}(t+1, t)\mathbf{K}(t, t-1)\mathbf{F}^T(t+1, t) + \mathbf{Q}_1(t), \quad (77)$$

$$\hat{\mathbf{x}}(t+1 | \mathcal{Y}_t) = \mathbf{F}(t+1, t)\hat{\mathbf{x}}(t | \mathcal{Y}_t). \quad (78)$$

Note: the local iteration over i continues until there is no significant difference between $\boldsymbol{\eta}_i$ and $\boldsymbol{\eta}_{i+1}$.

Manipulating (27), we can write

$$\mathbf{R}(t)\mathbf{G}^T(t) = \mathbf{C}(t)\mathbf{K}(t, t-1)\mathbf{F}^T(t+1, t). \quad (80)$$

Then, upon substituting (80) for the matrix product $\mathbf{C}(t)\mathbf{K}(t, t-1)\mathbf{F}^T(t+1, t)$ appearing in (79), we find

$$\begin{aligned} \mathbf{K}(t+1, t) &= \mathbf{F}(t+1, t)\mathbf{K}(t, t-1)\mathbf{F}^T(t+1, t) \\ &\quad - \mathbf{G}(t)\mathbf{R}(t)\mathbf{G}^T(t) + \mathbf{Q}_1(t). \end{aligned} \quad (81)$$

Equation (81) illustrates the problem inherent in the Riccati equation: as $\mathbf{K}(t+1, t)$ is the covariance matrix of the predicted state error $\boldsymbol{\epsilon}(t+1, t)$, it must be positive definite. Similarly, $\mathbf{R}(t)$ is the covariance matrix of the innovation $\boldsymbol{\alpha}(t)$ and must also be positive definite. Moreover, if $\mathbf{F}(t+1, t)$ and $\mathbf{G}(t)$ are full rank, then the terms $\mathbf{F}(t+1, t)\mathbf{K}(t, t-1)\mathbf{F}^T(t+1, t)$ and $\mathbf{G}(t)\mathbf{R}(t)\mathbf{G}^T(t)$ are positive definite as well. Therefore, (81) implies that a positive definite matrix $\mathbf{K}(t+1, t)$ must be calculated as the *difference* of the positive definite matrix $\mathbf{F}(t+1, t)\mathbf{K}(t, t-1)\mathbf{F}^T(t+1, t) + \mathbf{Q}_1(t)$ and positive definite

matrix $\mathbf{G}(t)\mathbf{R}(t)\mathbf{G}^T(t)$. Due to finite precision errors, the resulting matrix $\mathbf{K}(t+1, t)$ can become indefinite after a sufficient number of iterations, at which point the Kalman filter exhibits a behavior known as *explosive divergence* [18, Section 11].

As discussed in the appendix, a more stable implementation of the Kalman filter can be developed based on the *Cholesky decomposition* or *square root* of $\mathbf{K}(t+1, t)$, which is by definition that unique lower triangular matrix $\mathbf{K}^{1/2}(t+1, t)$ achieving

$$\mathbf{K}(t+1, t) \triangleq \mathbf{K}^{1/2}(t+1, t)\mathbf{K}^{T/2}(t+1, t). \quad (82)$$

The Cholesky decomposition of a matrix exists if and *only* if the matrix is symmetric and positive definite [19, Section 4.2.3]. The basic idea behind the square-root implementation of the Kalman filter is to update $\mathbf{K}^{1/2}(t+1, t)$ instead of $\mathbf{K}(t+1, t)$ directly. By updating or *propagating* $\mathbf{K}^{1/2}(t+1, t)$ forward in time, it can be assured that $\mathbf{K}(t+1, t)$ remains positive definite. Thereby, a numerically stable algorithm is obtained *regardless* of the precision of the machine on which it runs. The appendix presents a procedure whereby $\mathbf{K}^{1/2}(t+1, t)$ can be efficiently propagated in time using a series of *Givens rotations* [19, Section 5.1.8].

In the acoustic source localization experiments conducted thus far, the numerical stability has proven adequate even using the Kalman filter based directly on the Riccati equation. Instabilities can arise, however, when the audio features are supplemented with video information as in Gehrig et al. [20]. Hence, we have included the appendix for the sake of completeness.

4. SPEAKER TRACKING WITH THE KALMAN FILTER

In this section, we discuss the specifics of how the linearized least-squares position estimation criterion (16) can be recursively minimized with the iterated extended Kalman filter presented in Section 3.3. We begin by associating the “linearized” TDOA estimate $\bar{\mathbf{r}}(t)$ in (14) with the modified observation $\bar{\mathbf{y}}(t)$ appearing in (51). Moreover, we recognize that the linearized observation functional $\mathbf{C}(t)$ in (48) required for the Kalman filter is given by (10) and (13) for our acoustic localization problem. Furthermore, we can equate the TDOA error covariance matrix Σ in (15) with the observation noise covariance $\mathbf{Q}_2(t)$. Hence, we have all relations needed on the observation side of the Kalman filter. We need only supplement these with an appropriate model of the speaker dynamics to develop an algorithm capable of tracking a moving speaker, as opposed to merely finding his position at a single time instant. This is our next task.

4.1. Dynamic model

Consider the simplest model of speaker dynamics, wherein the speaker is “stationary” inasmuch as he moves only under the influence of the process noise $\mathbf{v}_1(t)$. Assuming the process noise components in the three directions are statistically

independent, we can write

$$\mathbf{Q}_1(t) = \sigma_p^2 T^2 \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad (83)$$

where T is the time elapsed since the last update of the state estimate, and σ_p^2 is the process noise power. Typically σ_p^2 is set based on a set of empirical trials to achieve the best localization results.

4.2. Position updates

Before performing an update, it is first necessary to determine the time T that has elapsed since an observation was last received. This factor appears as a parameter of the process noise covariance matrix $\mathbf{Q}_1(t)$ in (83). Although we assume the audio sampling is synchronous for all sensors, it cannot be assumed that the speaker constantly speaks, nor that all microphones receive the direct signal from the speaker’s mouth, that is, the speaker sometimes turns so that he is no longer facing the microphone array. As only the direct signal is useful for localization [21], the TDOA estimates returned by those sensors receiving only the indirect signal reflected from the walls should not be used for position updates. This is most easily assured by setting a threshold on the PHAT (6), and using for source localization only those microphone pairs returning a peak in the PHAT above the threshold [21]. This implies that no update at all is made if the speaker is silent.

Given the dynamic model in Section 4.1, we now have everything required for an acoustic speaker tracking system. The position update equations are given in Table 3. The nonlinear functional $\mathbf{C}(t, \mathbf{x}(t))$ appearing there corresponds to the TDOA model

$$\mathbf{T}(t, \mathbf{x}(t)) = \begin{bmatrix} T_0(\mathbf{x}(t)) \\ T_1(\mathbf{x}(t)) \\ \vdots \\ T_{N-1}(\mathbf{x}(t)) \end{bmatrix}, \quad (84)$$

where the individual components $T_i(\mathbf{x}(t))$ are given by (3)–(4). The linearized functional $\mathbf{C}(t) = \mathbf{C}(\mathbf{x}(t))$ is given by (10) and (13). To gain an appreciation for the severity of the nonlinearity in this particular Kalman filtering application, we plotted the actual value of $T_i(\mathbf{x}(t))$ against the linearized version. These plots are shown in Figures 1 and 2, respectively, for deviations in the x - and y -directions from the point about which $T_i(\mathbf{x}(t))$ was linearized. For these plots, the D -array in Figure 5 was used and the operating point was taken as $(x, y, z) = (2.950, 4.080, 1.700)$ m in room coordinates, which is approximately in the middle of the room. As is clear from the plots, for deviations of ± 1 m from the nominal, the linearized TDOA is within 2.33% of the true value for movement in the x -direction, and within 1.38% for movement in the y -direction.

Although the IEKF with the local iteration (72)–(74) was used for the experiments reported in Section 5, the localization system ran in less than real time on a Pentium Xeon

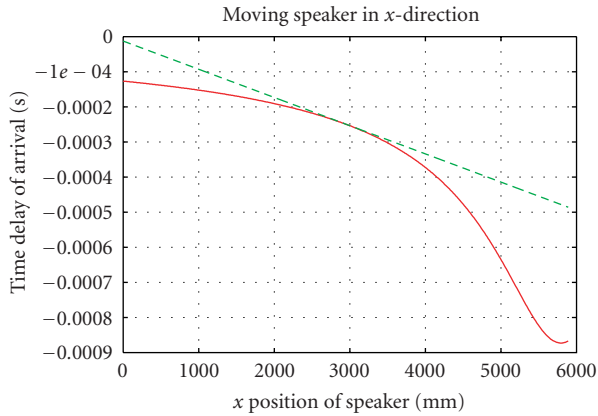


FIGURE 1: Actual versus linearized $T_i(\mathbf{x}(t))$ for movement in the x -direction.

processor with a clock speed of 3.0 GHz. This is so because during normal operation very few local iterations are required before the estimate converges, typically five or fewer. The local iteration compensates for the difference between the original nonlinear least-squares estimation criterion (5) and the linearized criterion (11). The difference between the two is only significant during startup and when a significant amount of time has passed since the last update, as in such cases the initial position estimate is far from the true speaker location. Once the speaker's position has been acquired to a reasonable accuracy, the linearized model (11) matches the original (5) quite well. The use of such a linearized model can be equated with the *Gauss-Newton method*, wherein higher order terms in the series expansion (9) are neglected. The connection between the Kalman filter and the Gauss-Newton method is well-known, as is the fact that the convergence rate of the latter is superlinear if the error $\hat{\tau}_i - T_i(\mathbf{x})$ is small near the optimal solution $\mathbf{x} = \mathbf{x}^*$. Further details can be found in Bertsekas [22, Section 1.5].

5. EXPERIMENTS

The test set used to evaluate the algorithms proposed here contains approximately three hours of audio and video data recorded during a series of seminars by students and faculty at the Universität Karlsruhe in Karlsruhe, Germany. The seminar room is approximately 6×7 m with a calibrated camera in each corner. An initial set of seven seminars was recorded in the fall of 2003. At that time, the seminar room was equipped with a single linear 16-channel microphone array with an intersensor spacing of 4.1 cm, as shown in Figure 3. The linear array was used for both beamforming and source localization experiments. In 2004, the audio sensors in the seminar room were enhanced to the configuration shown in Figure 5. The 16-channel linear array was replaced with a 64-channel Mark III microphone array developed at the US National Institute of Standards (NIST). This large array is intended primarily for beamforming, and was not used for the source localization experiments reported here.

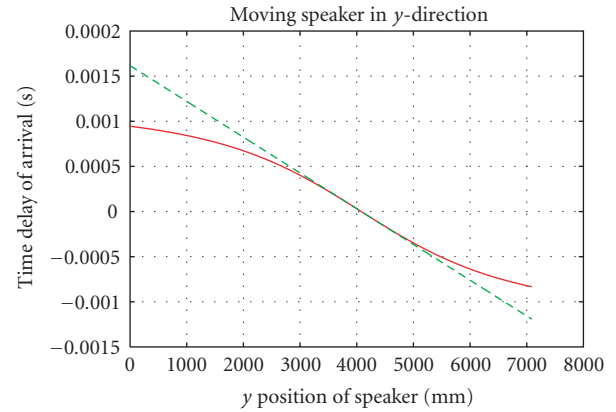


FIGURE 2: Actual versus linearized $T_i(\mathbf{x}(t))$ for movement in the y -direction.

Four T -shaped arrays were mounted on the walls of seminar room. The intersensor spacing of the T -arrays was chosen as either 20 or 30 cm in order to permit more accurate source localization. The T -shape was chosen to permit three-dimensional localization, which was impossible with the initial linear configuration. In the balance of this section, we report experimental results on both sensor configurations.

Prior to the start of the seminars, the four video cameras in the corners of the room had been calibrated with the technique of Zhang [23]. The location of the centroid of the speaker's head in the images from the four calibrated video cameras was manually marked every 0.7 second. Using these hand-marked labels, the true position of the speaker's head in three dimensions was calculated using the technique described in [24]. These "ground truth" speaker's positions are accurate to within 10 cm.

As the seminars took place in an open lab area used both by seminar participants as well as students and staff engaged in other activities, the recordings are optimally suited for evaluating acoustic source localization and other technologies in a realistic, natural setting. In addition to speech from the seminar speaker, the far field recordings contain noise from fans, computers, and doors, in addition to cross-talk from other people present in the room.

5.1. Experiments with linear microphone array

The simple dynamic model presented in Section 4.1 was used for all source localization experiments based on the IEKF reported in this section and in Section 5.2. Table 4 presents the results of a set of experiments comparing the new IEKF algorithm proposed in this work to the spherical intersection (SX) method of Schau and Robinson [6], the spherical interpolation (SI) method of [25] as well as the linear intersection (LI) technique of Brandstein et al. [4].

The SX method used three microphones of the array, numbers 0, 2, and 4, to make an initial estimate of the speaker's position. Only three microphones can be used, as the array is not two-dimensional, unlike the array in the original work [6]. To improve estimation results, the SI method

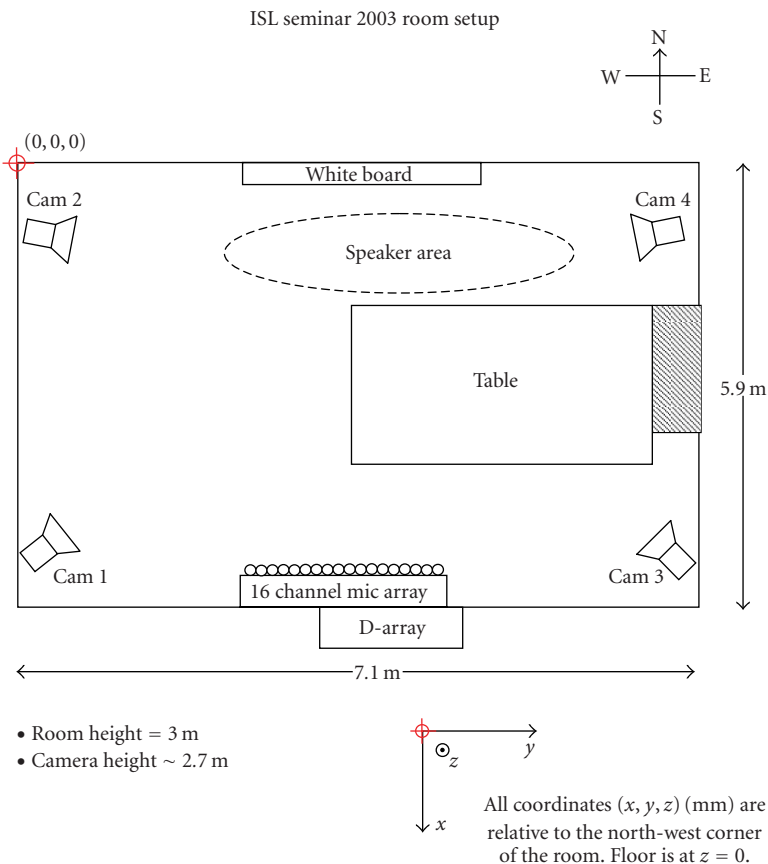


FIGURE 3: The room setup for the seminar recordings 2003 at Universität Karlsruhe.

TABLE 4: Experimental results of source localization and tracking algorithms.

Algorithm	RMS error			
	x (cm)	y (cm)	Azimuth (deg)	Depth (cm)
SX	144.6	166.6	24.6	148
SX + Kalman filter	138.1	153.5	20.4	145
SI	121.4	132.0	16.5	133
SI + Kalman filter	121.2	131.7	16.4	132
LI	225.0	130.5	17.6	234
LI + Kalman filter	196.1	111.5	13.3	207
IEKF	91.7	119.7	12.9	100

extends the ideas of the SX and enables the use of more than four microphones to take advantage of the redundancy. The array was divided into two subarrays and all microphones of these subarrays were used to estimate two positions. The final position estimate was the average of the two initial estimates. The LI and IEKF techniques, on the other hand, made use of the same set of 12 microphone pairs. These pairs were formed out of the microphone array by dividing the array into two eight-channel subarrays and taking each possible pair of microphones with an interelement distance of 8.14 cm. In all cases studied here, the TDOAs were estimated using the PHAT (6). Figure 4 shows the configuration of the array and definition of the microphone pairs in detail.

The results shown in Table 4 summarize the position estimation error over the 14 segments of the seminar data. The root mean square (RMS) errors were obtained by comparing the true speaker's positions obtained from the video labels with the position estimates produced by the several acoustic source localization algorithms. The position estimates from the Kalman filter were initially produced in Cartesian coordinates then converted to azimuth and depth. Table 4 reports results in both the original Cartesian coordinates, as well as azimuth and depth, as the latter are more meaningful for the linear array considered here. Position estimates from the SX, SI, and LI methods lying outside the physical borders of the room were omitted.

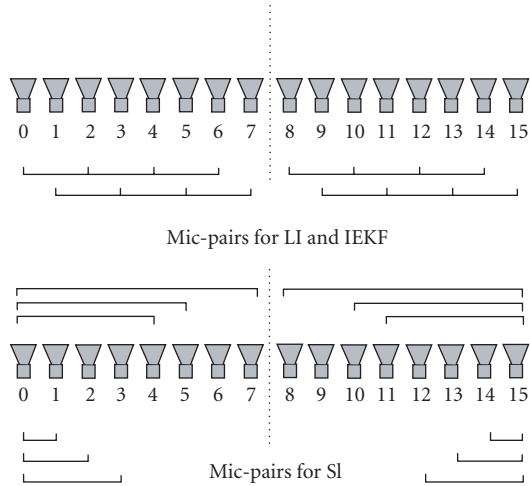


FIGURE 4: Use of microphone pairs for the different methods.

Without any smoothing, the source localization estimates returned by both the LI and SX methods are very inaccurate. The LI method provides particularly poor estimates in depth. Kalman filtering improved the position estimates provided by both the SX and LI methods, yet the average RMS distance from the true source location remained large. The SI method shows significantly better performance than the SX and LI methods both in total precision as well as in stability of the position estimations, as the results did not show the big variance of the first two methods. On the other hand, this improved stability reduces the improvement given by Kalman filtering, hence the filtered results do not show the big improvements noticeable for the SX and LI methods. The new IEKF approach outperformed all methods for both azimuth and depth. We attribute this superior performance largely to the elimination of the initial closed-form estimate associated with the LI, SI, and SX methods, and its inherent inaccuracy.

The performance of the IEKF could be further improved by implementing an adaptive threshold on the PHAT as proposed in [21]. The total gain is about 46% relative in terms of azimuth and about 47% in depth as shown in Table 5.

5.2. Experiments with *T*-shaped microphone arrays

Here we report the results of a set of experiments conducted with the *T*-shaped arrays whose location is indicated in Figure 5. For the new sensor configuration, we report results only in Cartesian coordinates, as azimuth and depth are no longer meaningful. Shown in Table 6 are source localization results obtained with the IEKF on two data sets: the *development set* consisting of three 15-minute segments, on which the parameters of the Kalman filter were tuned, and the *evaluation set* consisting of ten 15-minute segments chosen from five seminars. These results were obtained with a fixed PHAT threshold. Only TDOAs for pairs of microphones from the *same T*-array were estimated, as the distances between the *T*-arrays were too great to allow for reliable cross-correlation. The TDOAs from all microphone pairs for which the PHAT was above the fixed threshold were concatenated into a *single*

observation vector, which was then used to update the position estimate. As can be seen upon comparing the results in Table 6 with those in Table 4, the *T*-shaped arrays provide for significantly more accurate speaker localization. Moreover, the *T*-shape enables three-dimensional estimation. In the column of Table 6 labeled “3D,” we report the total RMS error for all dimensions; in the column labeled “2D,” the height or *z*-dimension is ignored.

Given that the position estimate can only be updated when the speaker is actually speaking, we also investigated the application of speech activity detection (SAD) to the source localization problem. We trained a neural net-based speech activity detector on the data from the close-talking microphone worn by the speaker, and only updated for time segments declared to be speech by the detector. We still retained the threshold criterion on the PHAT of each microphone pair, to ensure the update was based the signal received directly from the speaker’s mouth. As shown in Table 6, the use of an explicit SAD module provided a marginal improvement in the performance of the source localizer. We suspect that this is so because the threshold on the PHAT already provides an effective means of speech activity detection.

We also tested the LI method on the *T*-shaped arrays; the SI and SX methods require the calculation of all time delays with respect to a reference microphone and, as noted previously, the distances between the *T*-arrays are too great to estimate TDOAs reliably. In this case, we calculated a single bearing line for each microphone array, then calculated the point of nearest intersection for each unique pair of bearing lines. The final position estimate was given by the average of the points of nearest intersection, as specified in Brandstein et al. [4]. The LI results are shown in Table 7 for the evaluation set.

Comparing the results of Tables 6 and 7, we see that the IEKF still clearly outperforms LI. The accuracy of the LI method improves in the *x*-direction when the four *T*-arrays are used instead of the single linear array. The accuracy in the *y*-direction, however, degrades due to the fact that fewer microphone pairs are available to resolve the location of the source along this dimension.

In a final set of studies, we investigated the effect of speaker movement on the number of local iterations required by the IEKF. In Figure 6, we plotted the number of local iterations versus the time since the last update. Figure 7 shows the number of local iterations plotted against the distance the speaker has moved since the last update. Finally, Figure 8 displays local iterations versus speaker velocity. For each of the three cases, we calculated a regression line, which is also shown in the plot. As is clear from the figures, the average number of local iterations increases in proportion to the time since the last update, distance moved since the last update, and speaker velocity. These results correspond to our expectations, in that significant speaker movement implies that the linearized error criterion (11) does not initially match the true criterion (5). Hence, several local iterations are required for the position estimate to converge. Five or fewer local iterations were required for convergence in all cases, however, so that the system is still suitable for real-time speaker tracking.

TABLE 5: IEKF with and without adaptive threshold.

Algorithm	RMS error			
	x (cm)	y (cm)	Azimuth (deg)	Depth (cm)
IEKF	91.7	119.7	12.9	100
IEKF with adaptive threshold	52.2	61.9	6.97	52.9

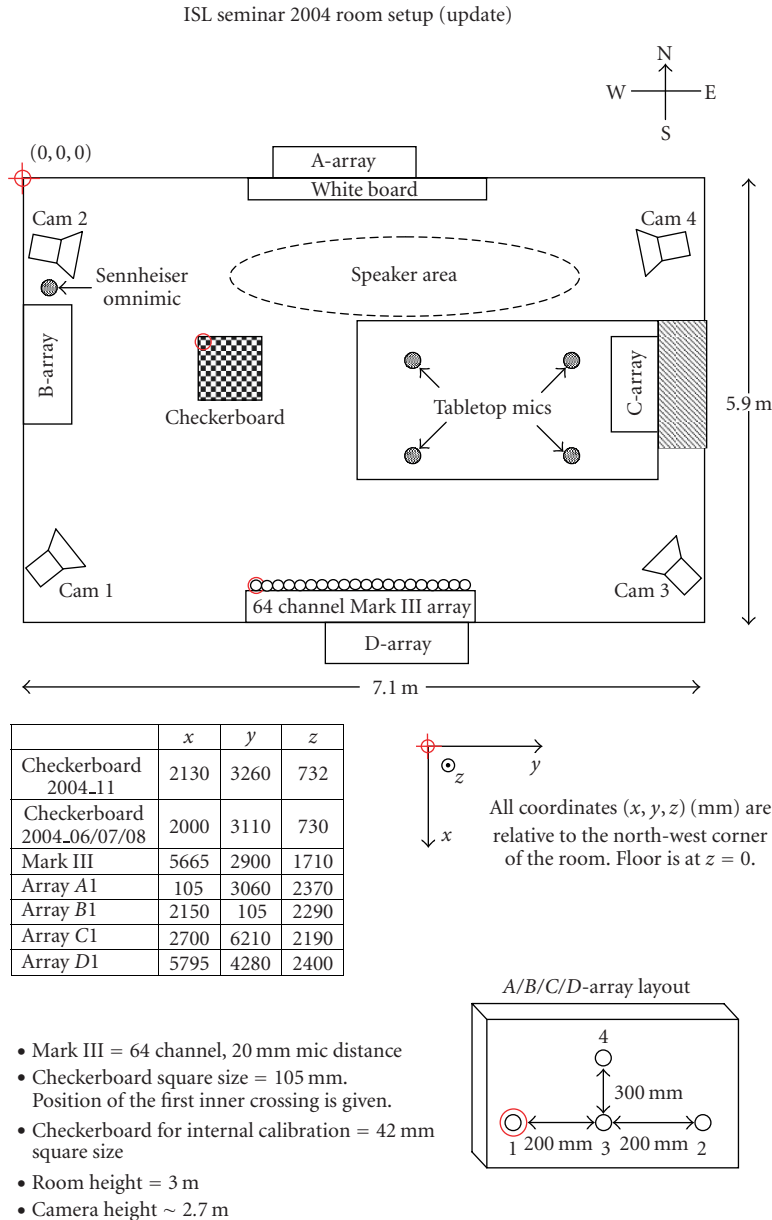


FIGURE 5: Current sensor configuration in the seminar room at Universität Karlsruhe.

Shown in Figure 9 are several images from a seminar recording. The light square marks the true speaker's position obtained from the hand-labeled images. The dark square is the three-dimensional position estimate after back projection to the two-dimensional image plane.

6. CONCLUSIONS

In this work, we have proposed an algorithm for acoustic source localization based on time delay of arrival estimation. In earlier work by other authors, an initial closed-form

TABLE 6: IEKF source localization results with the T -shaped arrays, both without and with explicit speech activity detection (SAD) on the close-talking microphone (CTM).

Experiment	RMS error (cm)				
	x	y	z	2D	3D
dev. set	39.0	40.4	9.9	56.3	57.1
eval. set	35.3	34.9	10.3	51.8	53.0
dev. set with SAD on CTM	36.0	35.6	9.4	50.7	51.6
eval. set with SAD on CTM	34.5	32.8	9.9	49.2	50.3

TABLE 7: Source localization results for the T -shaped arrays based on linear intersection.

Experiment	RMS error (cm)				
	x	y	z	2D	3D
LI	114.2	177.2	45.7	211.5	216.6
LI + Kalman filter	109.9	175.4	45.4	207.6	212.8

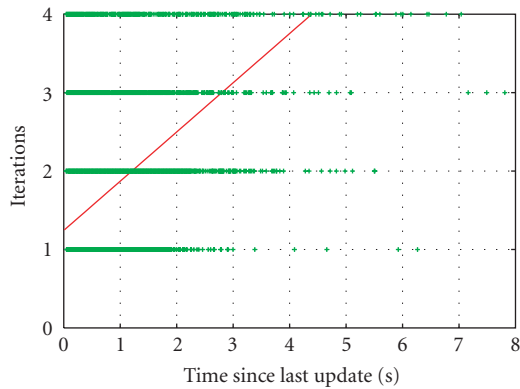


FIGURE 6: Number of local iterations versus time since last update.

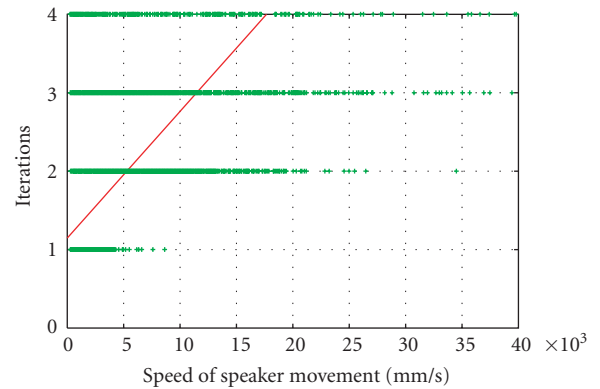


FIGURE 8: Number of local iterations versus speaker velocity.

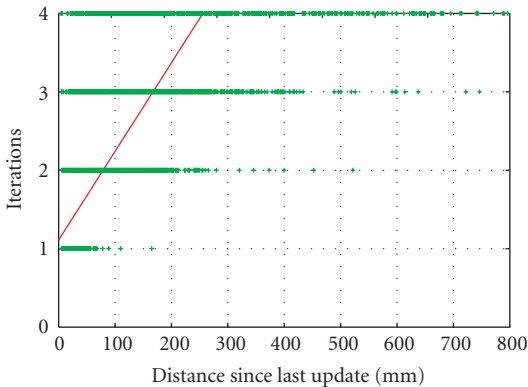


FIGURE 7: Number of local iterations versus distance moved since last update.

approximation was first used to estimate the true position of the speaker followed by a Kalman filtering stage to smooth the time series of position estimates. In our proposed algorithm, the closed-form approximation is eliminated by employing a Kalman filter to directly update the speaker's position estimate based on the observed TDOAs. To be more

precise, the TDOAs comprise the observation associated with an extended Kalman filter whose state corresponds to the speaker's position. We tested our algorithm on a data set consisting of seminars held by actual speakers, and recorded simultaneously with one or more microphone arrays, as well as four calibrated video cameras. From the video images, the true position of the speaker was extracted and used as "ground truth" for the automatic localization experiments. Our experiments revealed that the proposed algorithm provided source localization superior to the standard spherical and linear intersection techniques. We found that further improvements in localization accuracy could be obtained by adaptively setting a threshold on the PHAT function. Moreover, our algorithm ran in less than real time on an Intel Xeon processor with a 3.0 GHz clock speed.

In an other recent work [20], we have extended our technique to include features obtained from calibrated video cameras, as in the work by Strobel et al. [17]. This can be accomplished in a straightforward manner through the use of a video-based detector to find the speaker's face in a video image. Thereafter, the difference between the location of the detected face in the image and its predicted position obtained by back projecting from three-dimensional room



FIGURE 9: Images from four calibrated video cameras taken during a seminar recording at Universität Karlsruhe.

coordinates to two-dimensional image coordinates serves as the innovation vector. Hence, it is never necessary to triangulate the speaker's position from several video images as was done in [17]. We need only follow the approach adopted in this work and perform the standard update for a Kalman filter. Such an *incremental* update procedure was investigated by Welch and Bishop [26] for a different set of sensors. Welch [27] also analyzed this incremental scheme in terms of the observability criterion typically encountered in state-space control theory. He found that although the state of the system (i.e., the speaker's position) is *not* observable based on the information from any single sensor, the state becomes observable when the information from all sensors is combined. To achieve a stable update, it is only necessary to ensure that all sensors are sampled frequently enough.

APPENDIX

IMPLEMENTATION BASED ON THE CHOLESKY DECOMPOSITION

Using manipulations similar to those leading to (81), it is possible to combine (76)–(77) as

$$\begin{aligned} \mathbf{K}(t+1, t) &= \mathbf{F}\mathbf{K}(t, t-1)\mathbf{F}^T \\ &\quad - \mathbf{F}\mathbf{K}(t, t-1)\mathbf{C}^T(\boldsymbol{\eta}_i)\mathbf{R}^{-1}(t, \boldsymbol{\eta}_i) \\ &\quad \times \mathbf{C}(\boldsymbol{\eta}_i)\mathbf{K}(t, t-1)\mathbf{F}^T + \mathbf{Q}_1(t), \end{aligned} \quad (\text{A.1})$$

where $\mathbf{F} = \mathbf{F}(t+1, t)$ is the constant transition matrix and $\boldsymbol{\eta}_i$ is the current local iterate. Equation (A.1) is a statement of the *Riccati equation* for this particular formulation of the Kalman filter. The Riccati equation specifies how the state error correlation matrix $\mathbf{K}(t+1, t)$ can be recursively updated or *propagated* in time. As is well documented in the literature, however, this equation has poor numerical stability properties [18, Section 13.3]. In this section, we present a remedy, the *square-root implementation* [28], for the explosive

divergence phenomenon seen in Kalman filters based on the Riccati equation. Our discussion follows that in Haykin [18, Section 11].

The *Cholesky decomposition* [19, Section 4.2] or *square-root* $\mathbf{K}^{1/2}(t+1, t)$ of the state estimation error covariance matrix $\mathbf{K}(t+1, t)$ is the unique lower triangular matrix achieving

$$\mathbf{K}^{1/2}(t+1, t)\mathbf{K}^{T/2}(t+1, t) \triangleq \mathbf{K}(t+1, t). \quad (\text{A.2})$$

Square-root implementations of RLS estimators and Kalman filters propagate $\mathbf{K}^{1/2}(t+1, t)$. The advantage of this approach is that the Cholesky decomposition exists only for positive definite matrices [19]. Thus, in propagating $\mathbf{K}^{1/2}(t+1, t)$ instead of $\mathbf{K}(t+1, t)$, we ensure that the latter remains positive definite. The square-root implementation is derived from the following well-known lemma [28].

Lemma 1 (matrix factorization). *Given any two $N \times M$ matrices \mathbf{A} and \mathbf{B} with dimensions $N \leq M$,*

$$\mathbf{A}\mathbf{A}^T = \mathbf{B}\mathbf{B}^T \quad (\text{A.3})$$

if, and only if, there exists a unitary matrix $\boldsymbol{\theta}$ such that

$$\mathbf{A}\boldsymbol{\theta} = \mathbf{B}. \quad (\text{A.4})$$

To develop an update strategy based on the Cholesky decomposition, we set $\mathbf{Q}_2(t) = \boldsymbol{\Sigma}$, $\mathbf{Q}_1(t) = \mathbf{Q}(t)$, and write

$$\mathbf{A} = \begin{bmatrix} \boldsymbol{\Sigma}^{1/2} & \vdots & \mathbf{C}(\boldsymbol{\eta}_i)\mathbf{K}^{1/2}(t, t-1) & \vdots & \mathbf{0} \\ \dots & \dots & \dots & \dots & \dots \\ \mathbf{0} & \vdots & \mathbf{F}\mathbf{K}^{1/2}(t, t-1) & \vdots & \mathbf{Q}^{1/2}(t) \end{bmatrix}. \quad (\text{A.5})$$

We seek a unitary transform $\boldsymbol{\theta}$ that achieves $\mathbf{A}\boldsymbol{\theta} = \mathbf{B}$, such that

$$\begin{aligned} \mathbf{A}\mathbf{A}^T &= \begin{bmatrix} \boldsymbol{\Sigma}^{1/2} & \vdots & \mathbf{C}(\boldsymbol{\eta}_i)\mathbf{K}^{1/2}(t, t-1) & \vdots & \mathbf{0} \\ \dots & \dots & \dots & \dots & \dots \\ \mathbf{0} & \vdots & \mathbf{F}\mathbf{K}^{1/2}(t, t-1) & \vdots & \mathbf{Q}^{1/2}(t) \end{bmatrix} \\ &\quad \cdot \begin{bmatrix} \boldsymbol{\Sigma}^{T/2} & \vdots & \mathbf{0} \\ \dots & \dots & \dots \\ \mathbf{K}^{T/2}(t, t-1)\mathbf{C}^T(\boldsymbol{\eta}_i) & \vdots & \mathbf{K}^{T/2}(t, t-1)\mathbf{F}^T \\ \dots & \dots & \dots \\ \mathbf{0} & \vdots & \mathbf{Q}^{T/2}(t) \end{bmatrix} \\ &= \begin{bmatrix} \mathbf{B}_{11} & \vdots & \mathbf{0} & \vdots & \mathbf{0} \\ \dots & \dots & \dots & \dots & \dots \\ \mathbf{B}_{21} & \vdots & \mathbf{B}_{22} & \vdots & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{B}_{11}^T & \vdots & \mathbf{B}_{21}^T \\ \dots & \dots & \dots \\ \mathbf{0} & \vdots & \mathbf{B}_{22}^T \\ \dots & \dots & \dots \\ \mathbf{0} & \vdots & \mathbf{0} \end{bmatrix} = \mathbf{B}\mathbf{B}^T, \end{aligned} \quad (\text{A.6})$$

where both \mathbf{B}_{11} and \mathbf{B}_{22} are lower triangular. Performing the required multiplications on the block components, we find

$$\mathbf{B}_{11}\mathbf{B}_{11}^T = \boldsymbol{\Sigma} + \mathbf{C}(\boldsymbol{\eta}_i)\mathbf{K}(t, t-1)\mathbf{C}^T(\boldsymbol{\eta}_i), \quad (\text{A.7})$$

$$\mathbf{B}_{21}\mathbf{B}_{11}^T = \mathbf{F}\mathbf{K}(t, t-1)\mathbf{C}^T(\boldsymbol{\eta}_i), \quad (\text{A.8})$$

$$\mathbf{B}_{21}\mathbf{B}_{21}^T + \mathbf{B}_{22}\mathbf{B}_{22}^T = \mathbf{F}\mathbf{K}(t, t-1)\mathbf{F}^T + \mathbf{Q}(t). \quad (\text{A.9})$$

From (A.7), clearly,

$$\mathbf{B}_{11}\mathbf{B}_{11}^T = \mathbf{R}(t, \boldsymbol{\eta}_i). \quad (\text{A.10})$$

Hence, as \mathbf{B}_{11} is lower triangular,

$$\mathbf{B}_{11} = \mathbf{R}^{1/2}(t, \boldsymbol{\eta}_i). \quad (\text{A.11})$$

Based on (71),

$$\mathbf{G}_F(t, \boldsymbol{\eta}_i)\mathbf{R}(t, \boldsymbol{\eta}_i) = \mathbf{K}(t, t-1)\mathbf{C}^T(\boldsymbol{\eta}_i). \quad (\text{A.12})$$

Substituting for $\mathbf{K}(t, t-1)\mathbf{C}^T(\boldsymbol{\eta}_i)$ in (A.8), we find

$$\mathbf{B}_{21}\mathbf{B}_{11}^T = \mathbf{F}\mathbf{G}_F(t, \boldsymbol{\eta}_i)\mathbf{R}(t, \boldsymbol{\eta}_i) = \mathbf{F}\mathbf{G}_F(t, \boldsymbol{\eta}_i)\mathbf{B}_{11}\mathbf{B}_{11}^T, \quad (\text{A.13})$$

where the last equality follows from (A.10). Hence,

$$\mathbf{B}_{21} = \mathbf{F}\mathbf{G}_F(t, \boldsymbol{\eta}_i)\mathbf{R}^{1/2}(t, \boldsymbol{\eta}_i). \quad (\text{A.14})$$

Finally, substituting (55) into (A.14), we can rewrite (A.9) as

$$\begin{aligned} \mathbf{B}_{22}\mathbf{B}_{22}^T &= \mathbf{F}\mathbf{K}(t, t-1)\mathbf{F}^T - \mathbf{B}_{21}\mathbf{B}_{21}^T + \mathbf{Q}(t) \\ &= \mathbf{F}\mathbf{K}(t, t-1)\mathbf{F}^T - \mathbf{F}\mathbf{K}(t, t-1)\mathbf{C}^T(\boldsymbol{\eta}_i)\mathbf{R}^{-1}(t, \boldsymbol{\eta}_i) \\ &\quad \times \mathbf{C}(\boldsymbol{\eta}_i)\mathbf{K}^T(t, t-1)\mathbf{F}^T + \mathbf{Q}(t). \end{aligned} \quad (\text{A.15})$$

The last equation, together with (A.1), implies

$$\mathbf{B}_{22}\mathbf{B}_{22}^T = \mathbf{K}(t+1, t) \quad (\text{A.16})$$

or, as \mathbf{B}_{22} is lower triangular,

$$\mathbf{B}_{22} = \mathbf{K}^{1/2}(t+1, t). \quad (\text{A.17})$$

In light of (A.11)–(A.17), we have

$$\begin{aligned} \mathbf{A}\boldsymbol{\theta} &= \begin{bmatrix} \boldsymbol{\Sigma}^{1/2} & \vdots & \mathbf{C}(\boldsymbol{\eta}_i)\mathbf{K}^{1/2}(t, t-1) & \vdots & \mathbf{0} \\ \dots & \dots & \dots & \dots & \dots \\ \mathbf{0} & \vdots & \mathbf{F}\mathbf{K}^{1/2}(t, t-1) & \vdots & \mathbf{Q}^{1/2}(t) \end{bmatrix} \boldsymbol{\theta} \\ &= \begin{bmatrix} \mathbf{R}^{1/2}(t, \boldsymbol{\eta}_i) & \vdots & \mathbf{0} & \vdots & \mathbf{0} \\ \dots & \dots & \dots & \dots & \dots \\ \mathbf{F}\mathbf{G}_F(t, \boldsymbol{\eta}_i)\mathbf{R}^{1/2}(t, \boldsymbol{\eta}_i) & \vdots & \mathbf{K}^{1/2}(t+1, t) & \vdots & \mathbf{0} \end{bmatrix} = \mathbf{B}. \end{aligned} \quad (\text{A.18})$$

Although a new estimate of the predicted state error covariance matrix is generated with each local iteration, only the final result $\mathbf{K}^{1/2}(t+1, t)$ is saved for use on the succeeding time step.

The final position update is achieved as follows: through forward substitution we can find that $\boldsymbol{\zeta}'(t, \boldsymbol{\eta}_i)$ achieving

$$\boldsymbol{\zeta}(t, \boldsymbol{\eta}_i) = \mathbf{R}^{1/2}(t, \boldsymbol{\eta}_i)\boldsymbol{\zeta}'(t, \boldsymbol{\eta}_i), \quad (\text{A.19})$$

where $\boldsymbol{\zeta}(t, \boldsymbol{\eta}_i)$ is defined in (73). The development in Section 4.1 shows that \mathbf{F} is upper triangular for stationary source model of interest here. Hence, we can find that $\boldsymbol{\zeta}''(t, \boldsymbol{\eta}_i)$ achieving

$$\mathbf{F}\boldsymbol{\zeta}''(t, \boldsymbol{\eta}_i) = \mathbf{B}_{21}\boldsymbol{\zeta}'(t, \boldsymbol{\eta}_i) \quad (\text{A.20})$$

through back substitution on \mathbf{F} . Finally, as in (74), we update $\boldsymbol{\eta}_i$ according to

$$\boldsymbol{\eta}_{i+1} = \hat{\mathbf{x}}(t | \mathcal{Y}_{t-1}) + \boldsymbol{\zeta}''(t, \boldsymbol{\eta}_i), \quad (\text{A.21})$$

where $\boldsymbol{\eta}_1 = \hat{\mathbf{x}}(t | \mathcal{Y}_{t-1})$. The new state estimate $\hat{\mathbf{x}}(t+1 | \mathcal{Y}_t)$ is taken as the final iterate $\boldsymbol{\eta}_f$.

A unitary transform $\boldsymbol{\theta}$ that imposes the desired zeros on \mathbf{A} can be readily constructed from a set of Givens rotations; details are given in [29].

ACKNOWLEDGMENT

This work was sponsored by the European Union under the integrated project CHIL, *Computers in the Human Interaction Loop*, Contract No. 506909.

REFERENCES

- [1] M. Omologo and P. Svaizer, "Acoustic event localization using a crosspower-spectrum phase based technique," in *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '94)*, vol. 2, pp. 273–276, Adelaide, SA, Australia, April 1994.
- [2] S. M. Kay, *Fundamentals of Statistical Signal Processing: Estimation Theory*, Prentice-Hall, Englewood Cliffs, NJ, USA, 1993.
- [3] M. S. Brandstein, *A framework for speech source localization using sensor arrays*, Ph.D. thesis, Brown University, Providence, RI, USA, 1995.
- [4] M. S. Brandstein, J. E. Adcock, and H. F. Silverman, "A closed-form location estimator for use with room environment microphone arrays," *IEEE Transactions on Speech and Audio Processing*, vol. 5, no. 1, pp. 45–50, 1997.
- [5] Y. T. Chan and K. C. Ho, "A simple and efficient estimator for hyperbolic location," *IEEE Transactions on Signal Processing*, vol. 42, no. 8, pp. 1905–1915, 1994.
- [6] H. C. Schau and A. Z. Robinson, "Passive source localization employing intersecting spherical surfaces from time-of-arrival differences," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 35, no. 8, pp. 1223–1225, 1987.
- [7] J. O. Smith and J. S. Abel, "Closed-form least-squares source location estimation from range-difference measurements," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 35, no. 12, pp. 1661–1669, 1987.
- [8] T. G. Dvorkind and S. Gannot, "Speaker localization exploiting spatial-temporal information," in *Proceedings of the IEEE International Workshop on Acoustic Echo and Noise Control (IWAENC '03)*, pp. 295–298, Kyoto, Japan, September 2003.
- [9] R. Duraiswami, D. Zotkin, and L. S. Davis, "Multimodal 3-D tracking and event detection via the particle filter," in *Proceedings of IEEE Workshop on Detection and Recognition of Events in Video in Association with IEEE International Conference on Computer Vision*, pp. 20–27, Vancouver, BC, Canada, July 2001.
- [10] D. B. Ward, E. A. Lehmann, and R. C. Williamson, "Particle filtering algorithms for tracking an acoustic source in a reverberant environment," *IEEE Transactions on Speech and Audio Processing*, vol. 11, no. 6, pp. 826–836, 2003.
- [11] E. A. Lehmann, D. B. Ward, and R. C. Williamson, "Experimental comparison of particle filtering algorithms for acoustic source localization in a reverberant room," in *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal*

- Processing (ICASSP '03)*, vol. 5, pp. 177–180, Hong Kong, China, April 2003.
- [12] M. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, “A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking,” *IEEE Transactions on Signal Processing*, vol. 50, no. 2, pp. 174–188, 2002.
- [13] W. R. Hahn and S. A. Tretter, “Optimum processing for delay-vector estimation in passive signal arrays,” *IEEE Transactions on Information Theory*, vol. 19, no. 5, pp. 608–614, 1973.
- [14] A. H. Jazwinski, *Stochastic Processes and Filtering Theory*, Academic Press, New York, NY, USA, 1970.
- [15] J. Chen, J. Benesty, and Y. A. Huang, “Robust time delay estimation exploiting redundancy among multiple microphones,” *IEEE Transactions on Speech and Audio Processing*, vol. 11, no. 6, pp. 549–557, 2003.
- [16] Y. A. Huang, J. Benesty, G. W. Elko, and R. M. Mersereati, “Real-time passive source localization: a practical linear-correction least-squares approach,” *IEEE Transactions on Speech and Audio Processing*, vol. 9, no. 8, pp. 943–956, 2001.
- [17] N. Strobel, S. Spors, and R. Rabenstein, “Joint audio-video signal processing for object localization and tracking,” in *Microphone Arrays*, M. Brandstein and D. Ward, Eds., chapter 10, Springer, Heidelberg, Germany, 2001.
- [18] S. Haykin, *Adaptive Filter Theory*, Prentice-Hall, Englewood Cliffs, NJ, USA, 4th edition, 2002.
- [19] G. H. Golub and C. F. Van Loan, *Matrix Computations*, The Johns Hopkins University Press, Baltimore, Md, USA, 3rd edition, 1996.
- [20] T. Gehrig, K. Nickel, H. K. Ekenel, U. Klee, and J. McDonough, “Kalman filters for audio-video source localization,” in *Proceedings of IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA '05)*, pp. 118–121, New Paltz, NY, USA, October 2005.
- [21] L. Armani, M. Matassoni, M. Omologo, and P. Svaizer, “Use of a CSP-based voice activity detector for distant-talking ASR,” in *Proceedings of 8th European Conference on Speech Communication and Technology (EUROSPEECH '03)*, vol. 2, pp. 501–504, Geneva, Switzerland, September 2003.
- [22] D. P. Bertsekas, *Nonlinear Programming*, Athenea Scientific, Belmont, Mass, USA, 1995.
- [23] Z. Zhang, “A flexible new technique for camera calibration,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 11, pp. 1330–1334, 2000.
- [24] D. Focken and R. Stiefelwagen, “Towards vision-based 3-D people tracking in a smart room,” in *Proceedings of 4th IEEE International Conference on Multimodal Interfaces (ICMI '02)*, pp. 400–405, Pittsburgh, Pa, USA, October 2002.
- [25] J. S. Abel and J. O. Smith, “The spherical interpolation method for closed-form passive source localization using range difference measurements,” in *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '87)*, vol. 12, pp. 471–474, Dallas, Tex, USA, April 1987.
- [26] G. F. Welch and G. Bishop, “SCAAT: incremental tracking with incomplete information,” in *Proceedings of 24th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '97)*, pp. 333–344, Los Angeles, Calif, USA, August 1997.
- [27] G. F. Welch, *SCAAT: incremental tracking with incomplete information*, Ph.D. thesis, University of North Carolina, Chapel Hill, NC, USA, 1996.
- [28] A. H. Sayed and T. Kailath, “A state-space approach to adaptive RLS filtering,” *IEEE Signal Processing Magazine*, vol. 11, no. 3, pp. 18–60, 1994.
- [29] J. McDonough, U. Klee, and T. Gehrig, “Kalman filtering for time delay of arrival-based source localization,” Tech. Rep. 104, Interactive Systems Laboratories, Universität Karlsruhe, Karlsruhe, Germany, December 2004.

Ulrich Klee started his studies at Universität Karlsruhe (TH), Karlsruhe, Germany, in summer 2000 and will finish his education with a Diploma degree in computer science in spring 2006. Parts of this work were done during a one-semester scholarship at Carnegie Mellon University, Pittsburgh, USA.



Tobias Gehrig was born in Karlsruhe, Germany, in 1980. He finished high school in 1999 and received an Award for extraordinary achievements in natural science from the Alumni Association. Since October 2000, he has studied computer science at the University of Karlsruhe. He began working at the Institut für Theoretische Informatik in 2004 as a Student Assistant involved in data collection and source localization. His student thesis (*Studienarbeit*) was about audio-video source localization using Kalman filters.



John McDonough received his Bachelor of Science in 1989 and Master of Science in 1992 from Rensselaer Polytechnic Institute. From January 1993 until August 1997, he worked at the Bolt, Beranek, and Newman Corporation in Cambridge, Massachusetts primarily on large vocabulary speech recognition systems. In September 1997, he began doctoral studies at the Johns Hopkins University in Baltimore, Maryland, which he completed in April 2000. Since January of 2000, he has been employed at the Interactive Systems Laboratories at Universität Karlsruhe (TH), Germany, as a Researcher and Lecturer.

