Multi-dimensional Hybrid-space SAGE Algorithm: Joint Element-space and Beamspace Processing

C. M. Tan, M. A. Beach, and A. R. Nix

Centre for Communications Research
University of Bristol, Bristol BS8 1UB, U.K.
Email: Chor.Min.Tan@bristol.ac.uk

ABSTRACT

To date, most of the channel parameters are solely estimated in the element-space domain due to implementation simplicity. Although estimation in the beamspace domain can reduce both the dimensionality of data size and numerical complexity, it is only beneficial if applied to arrays with large number of elements. Here, we propose to implement the SAGE algorithm in joint element-space and beamspace, namely the hybrid-space, such that the underlying features of these domains can be exploited. The proposed algorithm retains the advantages in both domains since user can decide to estimate the parameters corresponding to a particular dimension in a more appropriate manner such that optimum performance can be obtained. We describe the general implementation of the frequency domain SAGE algorithm with hybrid-space processing and present sample of estimated results in both the synthetic and real measurement environments.

I. INTRODUCTION

In high-resolution array signal processing, multi-dimensional channel parameters estimation algorithms have emerged to be an active area of research in response to the need of overcoming the fundamental Rayleigh resolution [1] in Fourier method. Several estimation algorithms based on different implementation philosophies (e.g. subspace-based, maximum-likelihood-based, etc) have been developed. While multi-dimensional algorithms offer a higher resolution capability, a large amount of memory and computational routines are needed due to its large multi-dimensional data input. Although beamspace (BS) processing is able to reduce the data size by forming a number of beams in certain sector within the data, its application is only truly beneficial when the arrays in all dimensions have large number of elements (since the BS algorithm processes the data of all dimensions in their respective BS domains).

Due to the implementation simplicity in the element-space (ES), most researchers still prefer to process their data in ES (i.e. the classical technique) despite the advantages offered by BS processing. In order to combine the underlying advantages offered by both ES and BS processing, we propose a new implementation of the Space-Alternating Generalised Expectation-maximisation (SAGE) [2] algorithm with a combination of ES processing and the newly developed BS processing [3], namely the *hybrid-space* (HS) processing.

One can exploit the benefits of HS processing when at least one of the arrays (in multi-dimensional processing) has large number of elements. We can estimate the parameters in a particular dimension corresponding to the array with small number of elements in normal ES processing, and apply BS processing simultaneously in another dimension with large number of elements. Thus, a reduction in overall computational complexity and effective processing time. In addition, HS processing can offer a higher degree of flexilibity when a certain dimension might not be suitable for BS processing (e.g. an uncalibrated circular array). Furthermore, the automatic pairing procedure of the estimated parameters in all dimensions can still be performed in the standard manner within HS processing.

Here we describe the general implementation of the HS-SAGE algorithm in the frequency domain. We also present some estimated results in both the synthetic and real measurement environments. The results are compared to that of the classical SAGE algorithm in order to emphasise the advantages of using HS processing.

II. CHANNEL MODEL

This section briefly describes the physical channel model used in the proposed algorithm in order to aid the following discussions. The model is developed in the frequency domain specifically for use with the Medav RUSK BRI channel sounder [4]. The reader is referred to [3] and [4] for more detailed treatment of the model.

The R-dimensional (R-D) sounding snapshot data can be represented by a R-D data array in frequency domain, $\mathbf{H}$, whereby its element is given by:

$$H_{k_1,k_2,\cdots,k_R} = \sum_{l=1}^{L}\left(\gamma_l \prod_{r=1}^{R} e^{-jk_r\mu_l^{(r)}}\right) + N_{k_1,k_2,\cdots,k_R} \qquad (1)$$

where $k_r \in [0, K_r - 1]$ denotes the entry of the R-D array, K_r is the number of elements in the r-th dimension array, γ is the path weigth, L is the number of multipath components, $\mu^{(r)}$ represents the r-th dimension harmonics (i.e. parameter to be estimated) [3][5], and N is the complex Additive White Gaussian Noise (AWGN). The narrowband plane wave assumption must also hold for direction-of-arrival / departure (DoA/DoD) estimation. Further, $\Re\{\mu_l^{(r)}\} \in [0, 2\pi]$ must be fulfilled to avoid any estimation ambiguity (Shannon's sampling theorem), where $\Re\{\cdot\}$ is the range space of the argument.

In order to facilitate efficient matrix computation, the *R*-D matrix **H** is rearranged into compact vector form, **x**, through the $vec\{\cdot\}$ operator, by stacking the columns of **H**:

$$\begin{aligned}\mathbf{x} &= vec\left\{ \cdot \right\} \\ &= \left[\mathbf{a}\left(\mu_1^{(R-\mathrm{D})}\right) \vdots\ \mathbf{a}\left(\mu_2^{(R-\mathrm{D})}\right) \vdots\ \cdots \vdots\ \mathbf{a}\left(\mu_L^{(R-\mathrm{D})}\right)\right]\cdot \boldsymbol{\gamma} + \mathbf{n}\end{aligned} \tag{2}$$

where $\mathbf{n} = vec\left\{ \mathbf{N} \right\}$, $\boldsymbol{\gamma} = \left[\gamma_1, \gamma_2, \gamma_3, \cdots\cdots, \gamma_L\right]^T$ and the superscript *T* denotes the vector transposition. The *R*-D steering vector of the *l*-th path, $\mathbf{a}\left(\mu_l^{(R-\mathrm{D})}\right)$, is given by:

$$\mathbf{a}\left(\mu_l^{(R-\mathrm{D})}\right) = \mathbf{a}\left(\mu_1^{(R)}\right) \otimes \cdots \otimes \mathbf{a}\left(\mu_l^{(2)}\right) \otimes \mathbf{a}\left(\mu_l^{(1)}\right) \tag{3}$$

where $\otimes$ denotes the Kronecker product, and the *r*-th dimension steering vector of the *l*-th path is given by:

$$\begin{aligned}\mathbf{a}\left(\mu_l^{(r)}\right) = g\left(\mu_l^{(r)}\right)\cdot e^{j\frac{K_r-1}{2}\mu_l^{(r)}}\Big[\ & 1, e^{-j\mu_l^{(r)}}, \cdots\cdots, \\ & e^{-j(K_r-1)\mu_l^{(r)}}\ \Big]^T\end{aligned} \tag{4}$$

$g\left(\mu_l^{(r)}\right)$ represents the response of the array elements with respect to parameter $\mu_l^{(r)}$ (e.g. beampattern of a linear array). A total of *M* *R*-D channel snapshots can be taken within the quasi-stationary period of the channel and the snapshots can be grouped together for subsequent processing :

$$\mathbf{X} = \left[\ \mathbf{x}_1 \vdots\ \mathbf{x}_2 \vdots\ \mathbf{x}_3 \vdots\ \cdots\cdots \vdots\ \mathbf{x}_M\ \right] \tag{5}$$

Note that the array steering vector in (4) is only applicable for a linear array, e.g. arrays in the frequency and time domains (for Medav RUSK BRI [4]) after proper system calibration. Some modifications are necessary if other array structure is used, e.g. a circular array in the spatial domain (see Section IV and [6] for more details).

The point-source model in (1) can be expanded to represent the channel in a distributed-source environment in order to include the clustering phenomena. Here we define a cluster as a group of multipaths that are closely separated in all dimensions (e.g. due to local scattering). Without loss of generality, the distributed-source channel can be modelled as:

$$H_{k_1,k_2,\cdots,k_R} = \sum_{c=1}^{C}\sum_{l=1}^{L_c}\left(\gamma_{c,l}\prod_{r=1}^{R} e^{-jk_r\mu_{c,l}^{(r)}}\right) + N_{k_1,k_2,\cdots,k_R} \tag{6}$$

$$\mu_{c,l}^{(r)} = \mu_c^{(r)} + \delta\mu_{c,l}^{(r)} \tag{7}$$

where *C* is the number of clusters, L_c is the number of multipaths within a cluster (i.e. distributed sources), $\mu_c^{(r)}$ is the *r*-th dimension nominal parameter (i.e. centroid) of the *c*-th cluster, and $\delta\mu_{c,l}^{(r)}$ is the *r*-th dimension parameter deviation from $\mu_c^{(r)}$ for the *l*-th path within the *c*-th cluster. The distribution of $\mu_{c,l}^{(r)}$ can be modelled according to some standard distribution function, e.g. Gaussian, uniform, exponential, etc. Here we assume that the clusters do not overlap. Similarly, the compact vectorisation representation of the distributed-source channel can be performed in a similar way as (2)-(4) (not illustrated by means of step-by-step procedure here due to space limitation).

III. MULTI-DIMENSIONAL HYBRID-SPACE SAGE ALGORITHM

Suppose that we would like to implement BS processing in the 2nd-dimension of a *R*-D estimation problem, and the rest of the *R*-1 dimensions in the normal ES domain. Firstly, construct the appropriate DFT beamforming matrix, $\mathbf{W}^{(2)}$, that forms B_2 consecutive orthogonal beams encompassing the sector of interest in the 2nd-dimension, where $B_2 < K_2$ (see [3] for detailed construction of $\mathbf{W}^{(2)}$). Secondly, construct the *R*-D HS transformation matrix:

$$\mathbf{F}_{\mathrm{HS}}^{(R-\mathrm{D})} = \mathbf{I}_{K_R} \otimes \cdots\cdots \otimes \mathbf{W}^{(2)} \otimes \mathbf{I}_{K_1} \tag{8}$$

where $\mathbf{I}_{K_r}$ denotes the identity matrix of size K_r. Following that, the ES data, **X**, is transformed into the HS counterpart by performing:

$$\bar{\mathbf{x}}_{\mathrm{HS}} = \mathbf{F}_{\mathrm{HS}}^{(R-\mathrm{D})H}\bar{\mathbf{x}} \tag{9}$$

where $\bar{\mathbf{x}} = E[\mathbf{X}]$ (i.e. averaged over *M* snapshots), and the superscript *H* denotes Hermitian transposition.

The steps outlined above form the pre-processing of the HS-SAGE algorithm, which are also the main differences compared with the classical SAGE algorithm [2]. The following procedures are similar to that of the classical SAGE algorithm, which is based on the maximum-likelihood concept where the estimated results are the most probable values. This is done by maximising a cost function iteratively. Each iteration consists of an *Expectation-step* (E-step) and a *Maximisation-step* (M-step) [2]. Note that most of the computations within one iteration are performed within the M-step and HS-SAGE algorithm aims to reduce the computational burden within this step. Timesaving and reduction in computational complexity can be achieved with HS processing since at least one of the dimensions is processed in the BS domain with reduced data size (recall that the complexity of the SAGE algorithm grows rapidly with increasing number of dimensions and data size).

In the E-step, the *complete data* [2][3] of the *l*-th path, $\hat{\mathbf{z}}_l$, can either be obtained via parallel interference cancellation (PIC) (10) or successive interference cancellation (SIC) (11) technique:

$$\hat{\mathbf{z}}_{l,\langle PIC\rangle} = \bar{\mathbf{x}}_{\mathrm{HS}} - \sum_{\substack{l'=1 \\ l'\neq l}}^{\hat{L}} \hat{\mathbf{s}}_{l'} \tag{10}$$

$$\hat{\mathbf{z}}_{l,\langle SIC\rangle} = \bar{\mathbf{x}}_{\mathrm{HS}} - \sum_{l'=1}^{l-1} \hat{\mathbf{s}}_{l'} \tag{11}$$

where the '^' sign denotes the estimated data, and the signal copy, $\hat{\mathbf{s}}_l$, is given in (17). Note that $\hat{\mathbf{z}}_{l,\langle SIC\rangle}$ is subject to interference from the uncancelled (*l*+1)-th path

as well as the residue components from the previous paths that are not totally removed from $\hat{\mathbf{z}}_{l,\langle SIC\rangle}$. On the other hand, $\hat{\mathbf{z}}_{l,\langle PIC\rangle}$ is only subject to interference from the residue components of other $\hat{L}-1$ paths as long as the model order determination is accurate. From extensive numerical simulations, if the PIC technique is applied, the estimated results might be biased and the algorithm might diverge from steady state if $\hat{L}$ is less than the true number of dominant components in the channel.

In the M-step, the parameters are estimated sequentially by performing the *coordinate-wise-updating* procedure illustrated as follows.

$$\begin{aligned}
\mu_l^{(1)''} &= \arg \max_{\mu_l^{(1)}}\left\{ f\left(\mu_l^{(1)}, \mu_l^{(2)'}, \cdots\cdots, \mu_l^{(R)'} \right)\right\} \\
\mu_l^{(2)''} &= \arg \max_{\mu_l^{(2)}}\left\{ f\left(\mu_l^{(1)''}, \mu_l^{(2)}, \cdots\cdots, \mu_l^{(R)'} \right)\right\} \\
&\vdots \\
\mu_l^{(R)''} &= \arg \max_{\mu_l^{(R)}}\left\{ f\left(\mu_l^{(1)''}, \mu_l^{(2)''}, \cdots\cdots, \mu_l^{(R)} \right)\right\}
\end{aligned} \tag{12}$$

where $\mu_l^{(r)''}$ denotes the newly updated *r*-th dimension argument in the current iteration, $\mu_l^{(r)'}$ is the previously updated argument in the last iteration, and the optimisation cost function is given by:

$$f\left(\mu_l^{(1)}, \cdots, \mu_l^{(R)}\right) = \frac{\left| \mathbf{a}_{\mathrm{HS}}\left(\mu_l^{(R-\mathrm{D})}\right)^H \cdot \hat{\mathbf{z}}_l \right|}{\left\| \mathbf{a}_{\mathrm{HS}}\left(\mu_l^{(R-\mathrm{D})}\right) \right\|_{\mathrm{F}}} \tag{13}$$

where

$$\mathbf{a}_{\mathrm{HS}}\left(\mu_l^{(R-\mathrm{D})}\right) = \mathbf{a}\left(\mu_1^{(R)}\right) \otimes \cdots \otimes \mathbf{b}\left(\mu_l^{(2)}\right) \otimes \mathbf{a}\left(\mu_l^{(1)}\right) \tag{14}$$

$$\mathbf{b}\left(\mu_l^{(2)}\right) = \mathbf{W}^{(2)H}\mathbf{a}\left(\mu_l^{(2)}\right) \tag{15}$$

and $\| \cdot \|_{\mathrm{F}}$ denotes the Frobenius norm operator. The above procedure is repeated in iterations until convergence is achieved, i.e. when the estimated parameters have reached a steady state, or the change in parameters' value in the following few iterations is below a certain threshold level.

Recall that in this example we have applied BS processing in the 2nd-dimension. However, other dimensions can still apply BS processing if required. Note that the BS processing is best applied on linear array whereby its steering vector is column conjugate symmetric, e.g. such as that given by (4). As such, the beamspace steering vector, $\mathbf{b}\left(\mu_l^{(2)}\right)$, is real-valued since the proposed DFT beamforming matrix, $\mathbf{W}^{(2)}$, is also column conjugate symmetric [5]. Computations in real value will, to some extent, increase the processing speed of the algorithm.

Finally, before performing the next (*l*+1)-th E-step, the signal copy of the *l*-th path is reconstructed as follows.

$$\gamma_l = \mathbf{a}_{\mathrm{HS}}\left(\mu_l^{(R-\mathrm{D})''} \right)^{+} \hat{\mathbf{z}}_l \tag{16}$$

$$\hat{\mathbf{s}}_l = \gamma_l \cdot \mathbf{a}_{\mathrm{HS}}\left(\mu_l^{(R-\mathrm{D})''} \right) \tag{17}$$

where $\mathbf{a}_{\mathrm{HS}}(\cdot)^{+}$ is the Moore-Penrose pseudoinverse of $\mathbf{a}_{\mathrm{HS}}(\cdot)$, i.e. a least-squares solution. The initialisation procedure and model order determination (i.e. estimation of $\hat{L}$) of the algorithm can be found in [2] and [3].

IV. SIMULATED RESULTS

In order to validate that the HS-SAGE algorithm achieves a similar performance (in terms of estimation error) as the classical SAGE algorithm, we perform a simulation in a worst-case 3-D synthetical environment using a 2-cluster distributed-source model. Here we estimate the nominal azimuth DoA (ϕ_c), nominal time-delay-of-arrival (TDoA-τ_c), and nominal Doppler shift (ν_c) of the 2 closely spaced clusters. Each cluster is assumed to consist of 10 coherent paths, i.e. L_c = 10, $\forall c$. Their path weights are assumed to be the same (in terms of amplitude), and the complex phase angle is assumed uniformly distributed. Due to the difference in phase angles, fast fading (in terms of resultant power of each cluster) will occur in each cluster.

For simplicity, the 3-D channel parameters within each cluster are assumed to be uncorrelated to each other. The parameters are assumed to be Gaussian distributed with the mean given by the nominal parameter value (within the respective cluster) and the spread is specified in Table 1. The channel is synthesised in the frequency domain (see Section II) using an 8-element uniform circular array (UCA). The signal-to-noise ratio is set to 0 dB. Note that in this case the linear array steering vector such as that given in (4) has to be replaced by that for a UCA [6], given by:

$$\mathbf{a}_{\mathrm{UCA}}(\phi,\theta) = \left[a_1(\phi,\theta), a_2(\phi,\theta), \cdots\cdots, a_{K_\phi}(\phi,\theta) \right]^T \tag{18}$$

where $a_k(\phi,\theta) = g_k(\phi,\theta) \cdot e^{j\frac{2\pi}{\lambda} r \sin\theta \cos(\phi-\beta_{k-1})}$, $g_k(\phi,\theta)$ is the *k*-th element gain response, $\beta_{k-1} = 2\pi\frac{k-1}{K_\phi}$ (see Figure 1), *r* is the UCA radius, θ is the polar angle, and λ is the carrier wavelength. For simplicity, the simulation is implemented with UCA composed of omni-directional elements, and only the azimuth plane is considered, i.e. $\theta_{c,l} = 90°$, $\forall c,l$. Thus, only the nominal azimuth DoA is estimated in the spatial domain. The settings in the synthetical environment are given in Table 1. In addition, we include the Rayleigh resolution [1] of the system in Table 1. Note that the resolution of the system increases as the number of elements (in any dimension) increases.

In this particular simulation, we employ BS processing in the τ and ν domains, while the ϕ domain employs the classical ES processing since the number of spatial elements is small. We reduce the data size in the τ and ν domains by 40% by forming 6 beams (encompassing –25 to +25 Hz) in the ν domain, and 30 beams in the τ domain

(encompassing –200 to +400 ns). For comparison purpose, we apply 3-D HS-SAGE and 3-D classical SAGE algorithms with the same data set to estimate the nominal parameters. The algorithms are executed through 40 iterations using a same computer and their total processing time is recorded.

$\left[\mu_{c,l}^{(\phi)}, \mu_{c,l}^{(\tau)}, \mu_{c,l}^{(\nu)}\right]$	$\left[\phi_{c,l}, 2\pi\Delta f\tau_{c,l}, 2\pi\Delta t\nu_{c,l}\right]$ Δf = frequency array spacing Δt = time array spacing
$\left[r, \Delta f, \Delta t\right]$	[0.57λ, 1 MHz, 10 ms]
$\left[K_\phi, K_\tau, K_\nu\right]$	[8, 50, 10]
$\left[B_\tau, B_\nu\right]$	[30, 6] (for 3-D HS-SAGE)
Nominal parameters of cluster $\left[\phi_c, \tau_c, \nu_c\right]$	$[-10°, 10 \text{ ns}, -1 \text{ Hz}]_{c=1}$ $[10°, 15 \text{ ns}, 1 \text{ Hz}]_{c=2}$
Parameter standard deviation in each cluster $\left[\sigma_c^{(\phi)}, \sigma_c^{(\tau)}, \sigma_c^{(\nu)}\right]$	$[1°, 0.5 \text{ ns}, 0.2 \text{ Hz}]_{\forall c}$
Rayleigh resolution [1] $\left[\phi_{Ray}, \tau_{Ray}, \nu_{Ray}\right]$	[≈38°, 20 ns, 10 Hz]
Nominal parameter distance $\left[\Delta\phi, \Delta\tau, \Delta\nu\right]$	$\left[0.5\phi_{Ray}, 0.25\tau_{Ray}, 0.2\nu_{Ray}\right]$

Table 1: Parameters setting in the synthetical environment

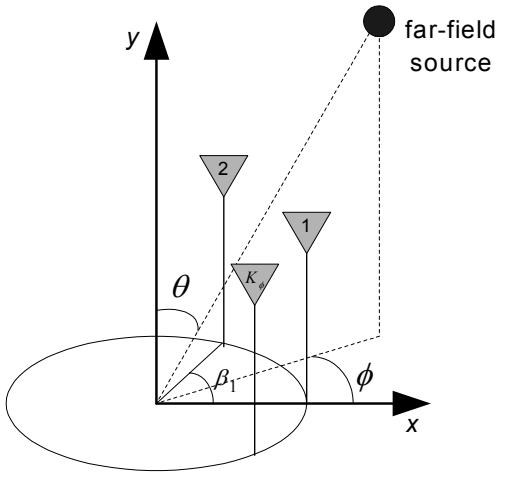

Figure 1: Standard geometry structure of a UCA

The estimation error of the nominal parameter in each iteration, $\xi(\cdot)$, is shown in Figures 2-3. It can be seen that the HS-SAGE algorithm exhibits a similar performance (in terms of convergence characteristic and estimation error) as the classical SAGE algorithm. The ratio of processing time needed by HS-SAGE to that of classical SAGE is 0.58. Significant amount of timesaving is achieved due to the reduction in complexity and array data size with HS-SAGE algorithm. Note that the processing time varies according to the total memory available for the algorithm, speed of the processor, and the efficiency in implementing the algorithm.

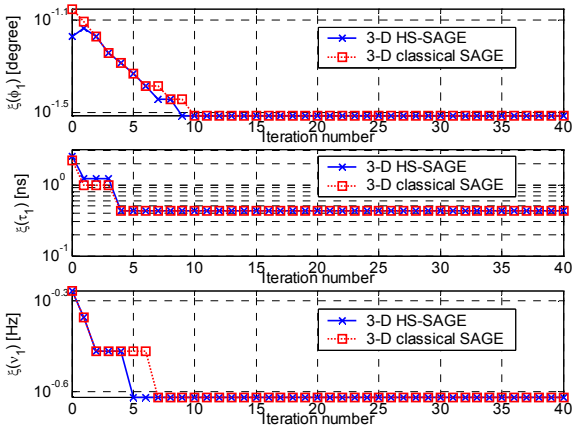

Figure 2: Estimation error of the nominal parameters in each iteration for the 1st cluster

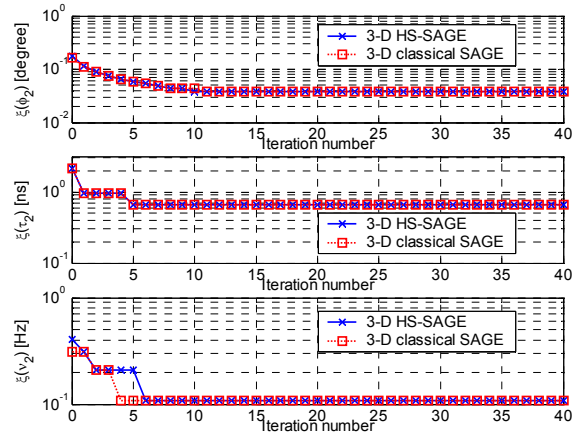

Figure 3: Estimation error of the nominal parameters in each iteration for the 2nd cluster

V. MEASUREMENT RESULTS

In this section, we present some new results from obtained samples of measurement data. The measurements were conducted at 5.2 GHz using a Medav RUSK BRI channel sounder [4] in a corridor-like environment at the University of Bristol (U.K.). The sounding bandwidth was 120 MHz and the transmitted multitone signal period was 0.8 µs. An 8-element UCA with vertically polarised dipole antenna was used at the receiver (Rx). The transmitter (Tx) employed an omni-directional antenna and was pushed at about 1 m/s towards the UCA's boresight direction at 0°. Measurements were recorded every 15 ms. The heights of both Tx and Rx were fixed at 1.55 m and their separation distance (from the starting point of Tx) was 10 m. Since nobody walked through the environment when the measurements were taken, the Doppler shift in the channel is mostly dominated by the movement of the Tx. Figure 4 gives an impression of the measurement environment.

We employ the 3-D HS-SAGE and 3-D classical SAGE algorithms to jointly estimate the DoA, TDoA, and Doppler shift of the multipaths. Each set of the results

corresponds to a 3-D data snapshot with dimension $K_\phi = 8$, $K_\tau = 97$, and $K_\nu = 5$. For 3-D HS-SAGE algorithm, we implement BS processing only in τ domain. The ϕ and ν domains are implemented in ES since they are small. We reduce the data size in the τ domain by approximately 50% by forming 50 beams in the region encompassing –100 to +300 ns since multipaths with delay greater than 300 ns could not be detected within the dynamic range of the sounder. This *a priori* channel information can be obtained by inspecting the power delay profile prior to HS-SAGE implementation.

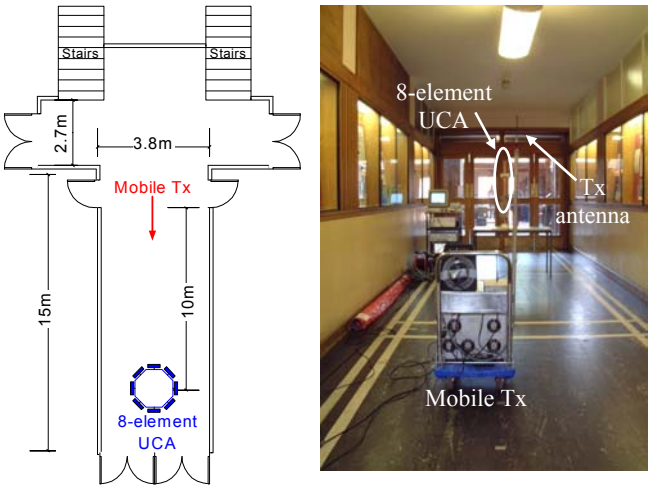

Figure 4: Sketched plan (left) and picture (right) of the measurement environment

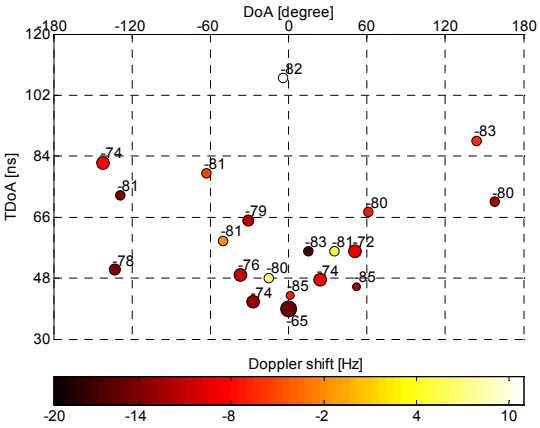

Figure 5: Estimated result of 3-D HS-SAGE algorithm

The sample results from the same set of 3-D data snapshot are displayed in Figures 5-6, whereby the estimated Doppler shift is shading-coded, and the calculated path weight is size-coded with its value (in dB) printed on the graphs. The graphs only display results within a power window of -20 dB relative to the strongest line-of-sight (LOS) component. Observe that the results are very similar to each other and the difference is much less than the intrinsic Rayleigh resolution [1] of the system. The estimated ϕ and ν values also match well with the speed of the mobile Tx (only the Tx was moving in the environment). However, complexity reduction and timesaving is achieved using 3-D HS-SAGE algorithm (0.63 processing time ratio with respect to that of the classical SAGE algorithm).

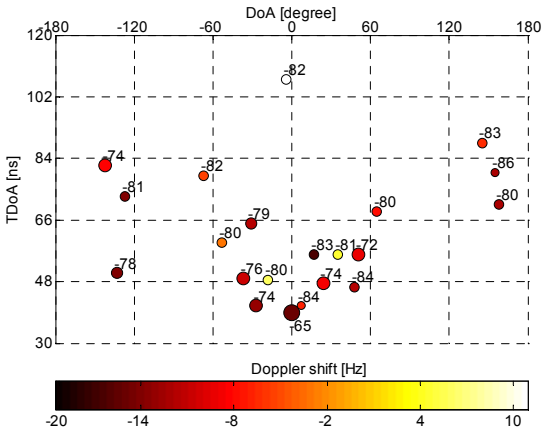

Figure 6: Estimated result of 3-D classical SAGE algorithm

VI. CONCLUSIONS

In this paper we have presented the frequency domain HS-SAGE algorithm that aims to reduce the processing time and complexity of the classical SAGE algorithm. The proposed algorithm achieves a similar performance as the classical SAGE algorithm, but with increased timesaving and reduced numerical complexity. Further work will include enhancing the robustness of the HS-SAGE algorithm to estimate parameters in a distributed-source environment, e.g. the spread of the parameters.

ACKNOWLEDGEMENT

The authors would like to acknowledge the U.K. Mobile VCE for the financial support of C. M. Tan.

REFERENCES

[1] S. Haykin, "*Adaptive Filter Theory*," 2nd edition, Prentice Hall, 1991.

[2] B. H. Fleury, M. Tschudin, R. Heddergott, D. Dahlhaus, K. I. Pedersen, "*Channel parameter estimation in mobile radio environments using the SAGE algorithm*," IEEE JSAC, vol. 17, 3 March 1999, pp. 434-450.

[3] C. M. Tan, M. A. Beach, A. R. Nix, "*Multi-dimensional DFT Beamspace SAGE super-resolution algorithm,*" IEEE SAM 2002 Workshop, Washington, 4-6 Aug. 2002.

[4] R. S. Thomä, D. Hampicke, A. Richter, G. Sommerkorn, A. Schneider, U. Trautwein, W. Wirnitzer, "*Identification of time-variant directional mobile radio channels*," IEEE Trans. Instrum. and Meas., April 2000, pp. 357–364

[5] M. Haardt, "*Efficient one-, two-, and multidimensional high-resolution array signal processing*," PhD. Thesis, ISBN 3-8265-2220-6, 1996.

[6] D. E. N. Davies, "*Circular arrays*," Chap. 12, The Handbook of antenna design, London Peregrinus on behalf of the IEE, 1983.