EUROPEAN COOPERATION
IN THE FIELD OF SCIENTIFIC
AND TECHNICAL RESEARCH

EURO-COST

COST 273 TD(05)105
Lisbon, Portugal
Nov 9-11, 2005

SOURCE: [1]Technische Universität Wien, Institut für Nachrichtentechnik und Hochfrequenztechnik, Wien, Österreich
[2]Forschungszentrum Telekommunikation Wien (ftw.), Wien, Österreich
[3]Technische Universität Ilmenau, Communications Research Laboratory, Ilmenau, Deutschland

Validating a Novel Automatic Cluster Tracking Algorithm on Synthetic IlmProp Time-Variant MIMO Channels

Nicolai Czink[1,2], Giovanni Del Galdo[3]

[1]Gusshausstraße 25/389 1040 Wien AUSTRIA
Phone: +43 (1) 58801 38979
Fax: +43 (1) 58801 38999
Email: nicolai.czink@tuwien.ac.at

Validating a novel automatic cluster tracking algorithm on synthetic time-variant MIMO channels

Nicolai Czink[1,2], Giovanni Del Galdo[3]
[1]Institut für Nachrichtentechnik und Hochfrequenztechnik, Technische Universität Wien, Austria
[2]Forschungszentrum Telekommunikation Wien (ftw.), Wien, Austria
[3]Technische Universität Ilmenau, Communications Research Laboratory, Ilmenau, Deutschland
E-Mail: nicolai.czink@tuwien.ac.at, giovanni.delgaldo@tu-ilmenau.de

Abstract

On the way to answer the controversial question "What is a cluster?", we introduce a novel cluster tracking mechanism which is based on the multi-path component distance (MCD). Starting from cluster estimates obtained by a recently introduced framework which automatically clusters parametric MIMO channel data, we are *tracking cluster centroids* in the multidimensional parameter domain.

To validate our algorithm, we are using synthetic MIMO channels generated by the IlmProp simulation tool. Furthermore we conducted a theoretical investigation on the influence of efficient parameter estimators and found that the estimation variances are negligible when using a sufficiently good measurement equipment.

1 Introduction

The problem of identifying clusters is currently the perhaps most controversial topic in channel characterisation. Discussions start right at the definition of a cluster itself. The only common agreement seems that a cluster should show some kind of similar parameters, which evolve smoothly over time.

In many papers visual inspection of measurement data from single snapshots was used [1, 2], which becomes impractical for large amounts of measurement data. Recently, an automatic algorithm was introduced in [3], which is based on clustering windowed parametric estimates and tracking the cluster centroids. The window-based clustering algorithm was subsequently improved by using the multi-path component distance (MCD) as the distance function in [4].

This algorithm was even further improved by including power into the cluster identification and extended to a whole framework. The framework is clustering the environment, decides on the correct number of clusters and prunes outliers. This is done for every single data window.

The last step is to subsequently track the obtained clusters. By cluster tracking we are able to extract the time-evolution of the parameters. The algorithm is validated by tracking MIMO channel parameters of a synthetic time-variant environment generated by the IlmProp channel model.

Thus we provide a new algorithm on the way to answer the question: "What is a cluster?"

2 Problem

The starting point is a large number of time-varying multi-dimensional parametric MIMO channel data. It has been investigated in several studies that these parameters tend to appear in clusters, i.e. in groups of multi-path components (MPCs) with similar parameters, such as delay, angles-of-arrival (AoA) and angles-of-departure (AoD). The problem is to find an automatic procedure to identify and track these clusters.

For cluster identification, we use a novel framework introduced recently [5], which clusters the environment, decides on the correct number of clusters, and prunes outliers. The input data to the clustering algorithm is an $K \times L$ array, where K is the number of estimated multipaths, and L is the number of channel parameters. Typically, the dimensions of L are power (P), delay (τ), azimuth and elevation AoA (φ_{AoA}, θ_{AoA}) and azimuth and elevation AoD (φ_{AoD}, θ_{AoD}). The output of the clustering algorithm is a set of cluster centroids in parameter space together with the allocated MPCs for each clustered scenario. In this paper we neglect the elevation domains, as clusters are insufficiently separated there.

To identify the time-variant behaviour of clusters, we need to track the cluster centroids over several time windows. The tracking algorithm must be able to decide on cluster birth, death, or movement.

3 Tracking algorithm

The cluster tracking mechanism is able to capture the movement of clusters, but yet with very low complexity. The idea is based on the distance between the clusters' centroids. As the centroids are given in the multi-dimensional parameter space, i.e. angles and delay, we chose the MCD [6] as suitable distance metric to cope with angular periodicity as well as data scaling.

Considering two subsequent sets of a number of N_{old} old and N_{new} new cluster centroids $\mathbf{c}_i^{(\text{old})}$ and $\mathbf{c}_j^{(\text{new})}$, where $i = 1 \ldots N_{\text{old}}$ and $j = 1 \ldots N_{\text{new}}$, the algorithm reads as

---

Tracking algorithm:

1. Calculate the distance between any old and any new centroid using the MCD.
2. For each new centroid:
   a. Calculate the distance and index of the closest old centroid.
   b. IF smallest distance $>$ threshold, treat centroid as new cluster
3. For each old centroid:
   a. Check number of close new centroids within distance threshold
   b. IF number = 1, old cluster moved.
   c. IF number $>$ 1, cluster split:
      – closest new cluster is treated as old cluster moved
      – other close ones are treated as new clusters

Ad 1) The distances between the centroids are arranged in the distance matrix $\mathbf{D}$ with dimension $N_{\text{old}} \times N_{\text{new}}$, where each element is calcualted as

$$[\mathbf{D}]_{ij} = \text{MCD}(\mathbf{c}_i^{(\text{old})}, \mathbf{c}_j^{(\text{new})}),$$

i.e. the distance between the ith old and jth new centroid. All further evaluations can now easily be done by searching in the distance matrix.

Ad 2a) For each column of $\mathbf{D}$, search for the smallest entry in the distance matrix. The indices i^*j^* of this value identifies the closest old cluster.

Ad 2b) If the distance $[\mathbf{D}]_{i^*j^*}$ between a new cluster and the closest old cluster exceeds a specified threshold ε, the cluster is treated as a new cluster.

Ad 3) We now check for each old cluster, if it has moved.

Ad 3a) For each row in $\mathbf{D}$ count the number of elements smaller than ε.

Ad 3b) If only one new cluster is in the vicinity of the old cluster, the old cluster has moved.

Ad 3c) If many new clusters are in the vicinity of the old cluster, the old cluster moved towards the closest new one. The other close ones are treated as new.

To every *new* cluster a unique cluster-ID is assigned. If a movement is identified, the moved cluster inherits the cluster-ID from its predecessor.

4 Validation Scenario

To validate the clustering framework and the tracking algorithm, we use a synthetic time-variant scenario obtained by the IlmProp channel model [7]. The IlmProp is a flexible geometry-based Multi-User MIMO channel modelling tool, capable of dealing with time variant frequency selective scenarios. Figure 1 illustrates the capabilities of the IlmProp. Three mobiles (M1, M2, and M3) move around the Base Station (BS). Their curvilinear trajectories are shown. The BS and mobile terminals can employ any number of antennas arranged in an array with an arbitrary geometry. The channel is computed as a sum of the Line Of Sight (LOS) and of a number of rays which represent the multi-path components. The latter are obtained by point-like scatterers, which can be placed at will. The model supports both single- and multiple-reflections. In Figure 1 the LOS for M1, a single bounce ray and a double bounce ray are shown. The information about where the scatterers are, and how the paths are linked to them can be set arbitrarily, or it can be derived by parameter estimations from channel measurements. The information about the scenario is stored in form of Cartesian coordinates and their evolution in time. The scenario may include obstacles (such as buildings), which can obstruct the propagation paths. More information on the model, as well as the source code and some exemplary scenarios can be found at *http://tu-ilmenau.de/ilmprop*.

Figure 2 shows the scenario we use to validate the automatic cluster tracking algorithm. A single mobile (MS) is moving on a linear trajectory (along the x-axis) towards the Base Station (BS). The LOS path was artificially suppressed. We consider the uplink phase, so that the BS acts as the receiver.

The scenario exhibits 8 separate clusters. The cluster numbering corresponds to the automatic cluster identification (see Section 6). All propagations paths are characterized by single bounce reflections with the exception of the ones named $6a$ and $6b$, which realize double bounce paths. Solid black lines indicate the interconnections between MS, clusters and the BS. The path towards the cluster number 10 is obstructed by a building during most of the mobile's trajectory. For this reason it is marked by a dashed line.

The clustering algorithm is meant to identify clusters from the geometrical parameters extracted from channel measurements by a parameter estimation technique. The idea is to extract these parameters, i.e. delay time, direction of arrival and departure, and path-strength, directly from the IlmProp, and feed them to the cluster identification framework. Subsequently, the clusters are tracked. In the next section we consider the issue of the variance introduced by the parameter estimator, showing that for the aims of our investigation it can be neglected.

5 Influence of the parameter estimator

The investigation we carry out from the IlmProp parameters inherently assumes that the IlmProp correctly models the propagation phenomenon and additionally that the parameter estimation technique delivers perfect estimates of the parameters considered for the clustering algorithm. The parameters we consider are delay time τ, Direction of Arrival (DoA) φ_{R}, Direction of Departure (DoD) φ_{T}, and complex path-strength $(\gamma_{\mathrm{Re},i} + j\gamma_{\mathrm{Im},i})$, for each of the MPCs. In this section we investigate the issue of the variances introduced by a parameter estimation technique.

One possibility to assess this would be to actually carry out the parameter estimation processing on the synthetic channel matrices computed by the IlmProp, as one would do on real channel measurements.

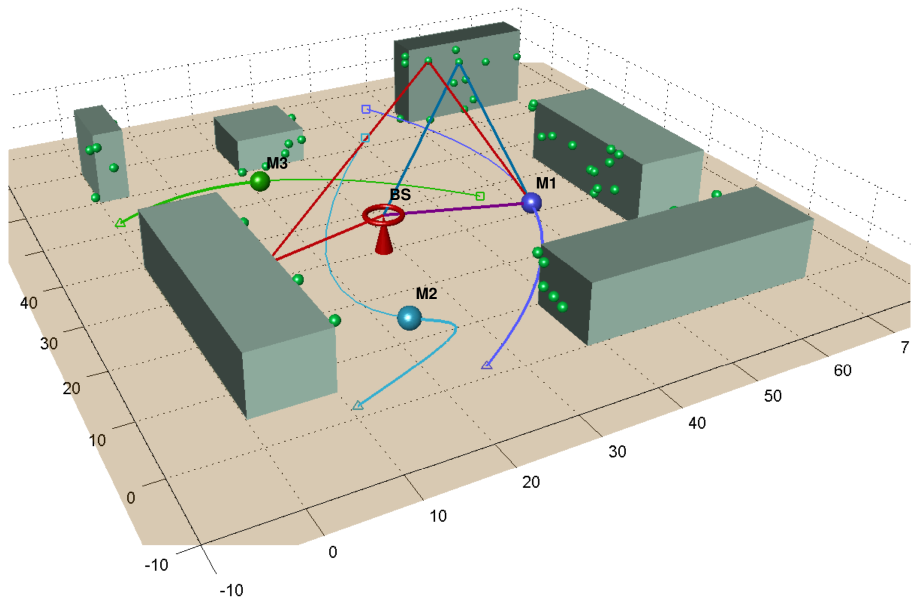

Figure 1: Sample scenario generated with IlmProp to illustrates the capabilities of the channel model

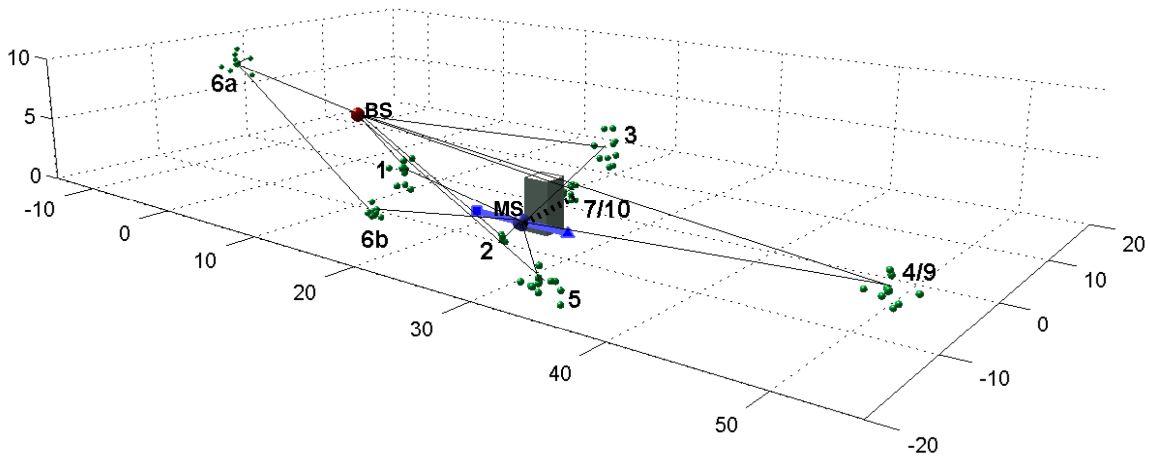

Figure 2: Scenario used to validate the automatic cluster tracking algorithm; cluster numbers correspond to the output of the tracking algorithm

Although this strategy is something which we will pursue in the near future, in this contribution we want to follow a more theoretical approach.

Assuming that the parameter estimator is an unbiased efficient estimator, we can compute the variance of the white Gaussian noise that the estimator adds to the exact parameters, by means of the Cramer-Rao Lower Bound (CRLB).

Let the vector $\boldsymbol{\theta} \in \mathbb{C}^{L\times 1}$ contain the L exact parameters, and $\hat{\boldsymbol{\theta}} \in \mathbb{C}^{L\times 1}$ its noisy estimates. By taking an unbiased efficient estimator we set

$$\mathrm{E}\left\{ \left(\boldsymbol{\theta} - \hat{\boldsymbol{\theta}}\right) \cdot \left(\boldsymbol{\theta} - \hat{\boldsymbol{\theta}}\right)^{\mathrm{T}} \right\} \equiv \boldsymbol{C}, \tag{1}$$

where the matrix $\boldsymbol{C} \in \mathbb{C}^{L\times L}$ is the CRLB.

At time snapshot t, the parameter estimator receives from the channel sounder a noisy observation of the channel in the form of a three-dimensional array $\boldsymbol{H} \in \mathbb{C}^{M_{\mathrm{R}}\times M_{\mathrm{T}}\times F}$, where M_{R}, and M_{T} are the number of receive and transmit antennas, respectively, and F is the number of frequency bins. The channel realization $\boldsymbol{H}$ is a result of a known matrix valued function $\boldsymbol{S}(\cdot) : \mathbb{R}^{L\times 1} \rightarrow \mathbb{C}^{M_{\mathrm{R}}\times M_{\mathrm{T}}\times F}$ which maps the L real-valued parameters contained in $\boldsymbol{\theta}$ to the $M_{\mathrm{R}} \cdot M_{\mathrm{T}} \cdot F$ values contained in the observation $\boldsymbol{H}$, plus a noise term, as in

$$\boldsymbol{H} = \boldsymbol{S}(\boldsymbol{\theta}) + \boldsymbol{N}, \tag{2}$$

where the matrix $\boldsymbol{N}$ contains i.i.d. complex white Gaussian noise.

Let $\boldsymbol{\theta}$ contain the parameters of K paths. The ith path is described by its delay time τ_i, DoA $\varphi_{\mathrm{R},i}$, DoD $\varphi_{\mathrm{T},i}$, as well as by a complex path-strength, expressed by its real and imaginary part $\gamma_{\mathrm{Re},i}$ and $\gamma_{\mathrm{Im},i}$, respectively. The parameters are arranged as follows

$$\boldsymbol{\theta} = \left[\tau_1 \ldots \tau_K \;,\; \varphi_{\mathrm{R},1} \ldots \varphi_{\mathrm{R},K} \;,\; \varphi_{\mathrm{T},1} \ldots \varphi_{\mathrm{T},K} \;,\; \gamma_{\mathrm{Re},1} \ldots \gamma_{\mathrm{Re},K} \;,\; \gamma_{\mathrm{Im},1} \ldots \gamma_{\mathrm{Im},K}\right]^{\mathrm{T}}, \tag{3}$$

so that the vector $\boldsymbol{\theta}$ has length $L = 5 \cdot K$.

The function $\boldsymbol{S}(\boldsymbol{\theta})$ for a specific frequency f returns a $M_{\mathrm{R}} \times M_{\mathrm{T}}$ matrix which can be written as

$$\boldsymbol{A}_{\mathrm{R}}(\varphi_{\mathrm{R},1\ldots K}) \cdot \boldsymbol{\Lambda}(\tau_{1\ldots K}, \gamma_{\mathrm{Re},1\ldots K}, \gamma_{\mathrm{Im},1\ldots K}) \cdot \boldsymbol{A}_{\mathrm{T}}^{\mathrm{H}}(\varphi_{\mathrm{T},1\ldots K}) \in \mathbb{C}^{M_{\mathrm{R}}\times M_{\mathrm{T}}}, \tag{4}$$

where

$$\boldsymbol{A}_{\mathrm{R}}(\varphi_{\mathrm{R},1\ldots K}) = \left[\boldsymbol{a}_{\mathrm{R}}\left(\varphi_{\mathrm{R},1}\right) \;\ldots\; \boldsymbol{a}_{\mathrm{R}}\left(\varphi_{\mathrm{R},K}\right)\right] \in \mathbb{C}^{M_{\mathrm{R}}\times K} \tag{5}$$

$$\boldsymbol{\Lambda}(\tau_{1\ldots K}, \gamma_{\mathrm{Re},1\ldots K}, \gamma_{\mathrm{Im},1\ldots K}) = \begin{bmatrix} \lambda_1 & 0 & \ldots & 0 \\ 0 & \lambda_2 & & 0 \\ 0 & & \ddots & \vdots \\ 0 & 0 & \ldots & \lambda_K \end{bmatrix} \in \mathbb{C}^{K\times K} \tag{6}$$

$$\lambda_i = \left(\gamma_{\mathrm{Re},i} + j\gamma_{\mathrm{Im},i}\right) \mathrm{e}^{-j2\pi f \tau_i} \tag{7}$$

$$\boldsymbol{A}_{\mathrm{T}}(\varphi_{\mathrm{T},1\ldots K}) = \left[\boldsymbol{a}_{\mathrm{T}}\left(\varphi_{\mathrm{T},1}\right) \;\ldots\; \boldsymbol{a}_{\mathrm{T}}\left(\varphi_{\mathrm{T},K}\right)\right] \in \mathbb{C}^{M_{\mathrm{T}}\times K}. \tag{8}$$

The terms $\boldsymbol{a}_{\mathrm{R}}(\cdot)$ and $\boldsymbol{a}_{\mathrm{T}}(\cdot)$ are the array steering vectors for the receive and transmit side, respectively. The function $\boldsymbol{s}(\cdot) : \mathbb{R}^{L\times 1} \rightarrow \mathbb{C}^{M_{\mathrm{R}}\cdot M_{\mathrm{T}}\cdot F\times 1}$ computes F channel matrices as seen in equation (4), stacks them in a three-dimensional array of size $M_{\mathrm{R}} \times M_{\mathrm{T}} \times F$ and finally applies the $\mathrm{vec}[\cdot]$ operator to obtain a column vector of length $M_{\mathrm{R}} \cdot M_{\mathrm{T}} \cdot F$.

We now rewrite equation (2) in vector form, by applying the $\mathrm{vec}\{\cdot\}$ operator, so that

$$\boldsymbol{h} = \boldsymbol{s}(\boldsymbol{\theta}) + \boldsymbol{n}, \tag{9}$$

where

$$\boldsymbol{h} \doteq \operatorname{vec}\{\boldsymbol{H}\} \in \mathbb{C}^{M_{\mathrm{R}} \cdot M_{\mathrm{T}} \cdot F \times 1} \tag{10}$$

$$\boldsymbol{n} \doteq \operatorname{vec}\{\boldsymbol{N}\} \in \mathbb{C}^{M_{\mathrm{R}} \cdot M_{\mathrm{T}} \cdot F \times 1}, \tag{11}$$

and $\boldsymbol{s}(\cdot)$ is the vector valued function mapping $\mathbb{R}^{L\times 1} \rightarrow \mathbb{C}^{M_{\mathrm{R}} \cdot M_{\mathrm{T}} \cdot F \times 1}$.

In order to determine the CRLB we need first to compute the Fisher Information Matrix (FIM) $\boldsymbol{F}(\boldsymbol{\theta})$ by means of the Jacobian $\boldsymbol{D}(\boldsymbol{\theta})$. Note that a more complete and rigorous derivations of the CRLB can be found in [8].

The Jacobian $\boldsymbol{D}(\boldsymbol{\theta})$ is the matrix valued function containing the first order partial derivatives of the data model $\boldsymbol{s}(\boldsymbol{\theta})$ with respect to $\boldsymbol{\theta}$ as

$$\boldsymbol{D}(\boldsymbol{\theta}) = \frac{\partial}{\partial \boldsymbol{\theta}^{\mathrm{T}}} \boldsymbol{s}(\boldsymbol{\theta}) \in \mathbb{C}^{M_{\mathrm{R}} \cdot M_{\mathrm{T}} \cdot F \times L}. \tag{12}$$

The Fisher Information Matrix $\boldsymbol{F}(\boldsymbol{\theta})$ can be computed as

$$\boldsymbol{F}(\boldsymbol{\theta}) = \frac{2}{\sigma_n^2} \operatorname{Re}\left\{\boldsymbol{D}^{\mathrm{H}}(\boldsymbol{\theta}) \cdot \boldsymbol{D}(\boldsymbol{\theta})\right\} \in \mathbb{C}^{L\times L}, \tag{13}$$

where we assume that noise correlation matrix $\boldsymbol{R}_n \in \mathbb{C}^{M_{\mathrm{R}} \cdot M_{\mathrm{T}} \cdot F \times M_{\mathrm{R}} \cdot M_{\mathrm{T}} \cdot F}$ of the noise term $\boldsymbol{n}$ is $\sigma_n^2 \boldsymbol{I}$ and $\operatorname{Re}\{\cdot\}$ extract the real part of its argument.

The L values on the diagonal of the FIM tell us how "informative" the channel observation is with respect to each of the L different parameters. The off-diagonal elements, on the other hand, account for the mutual information between the parameters.

The CRLB $\boldsymbol{C}(\boldsymbol{\theta})$ can be easily computed from the FIM $\boldsymbol{F}(\boldsymbol{\theta})$ as

$$\boldsymbol{C}(\boldsymbol{\theta}) = \boldsymbol{F}^{-1}(\boldsymbol{\theta}), \tag{14}$$

so that the ith element of the diagonal of $\boldsymbol{C}(\boldsymbol{\theta})$ is the variance of the noisy estimate of the ith element of $\boldsymbol{\theta}$.

In order to compute *numerical* values for the CRLB we first need to derive its symbolic expression for a specific data model $\boldsymbol{s}(\boldsymbol{\theta})$. Subsequently we substitute the numerical values of all variables and evaluate the expression.

For our simulations we assume a 16×16 MIMO system, where both base station and mobile employ a 16-sensor Uniform Linear Array (ULA). All antennas are assumed omnidirectional. The parameter estimator receives for every time snapshot $F = 301$ frequency bins spanning a bandwidth of 120 MHz.

Deriving the CRLB first symbolically and then numerically involves a huge effort in computation time. Therefore we considered two clusters only, the clusters marked with 4 and 5, reduced to having a number of scatters of 4 and 6 respectively. Cluster 4 shows much larger delay, i.e. it has worse path SNR. Hence we expect lower variances for cluster 5.

Figure 3 (a)-(e) show the results from this evaluation for every dimension of the parameter space for the two considered clusters. Plots (a)-(b) show that the standard deviation of the azimuth is usually lower than 2° with one outlier at 4°. The standard deviation of the delay is smaller than 15 ns as shown in plot (c). Plots (d) and (e) demonstrate the high accuracy of the estimation of the path weight, which is below 2×10^{-3} corresponding to a maximum deviation of $0.1\,\%$ of the true value. As clearly seen, the the variances for the parameters of cluster 5 are indeed smaller.

The results show that the variances are so low, that we can neglect them and simply feed the exact parameters to the clustering algorithm. The next section shows the results of our simulation.

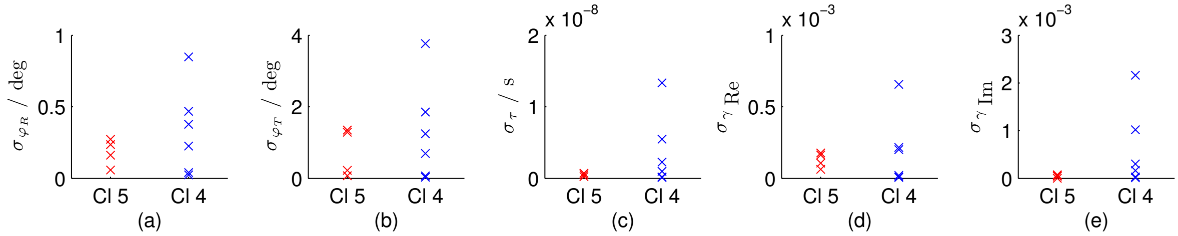

Figure 3: Standard deviations which are achieved by an efficient estimator for the simulated environment: (a) azimuth at the Tx, (b) azimuth at the Rx, (c) delay, (d) real part of path weight, (e) imaginary part of path weight.

6 Results

Figure 2 shows the simulated scenario, details are provided in Section 4. The channel model provided 80 subsequent equidistant snapshots in time of the smoothly time-varying channel, while the MS moved towards the BS.

For clustering the synthetic scenario, we collected 3 subsequent time snapshots in a sliding window. The clustering algorithm identified between 6 and 7 clusters in each time window. Subsequently, the tracking algorithm was applied.

The results are shown in Figure 4, where the evolution of the clusters over time are shown in ascending order (a)-(f). Clusters are indicated by coloured spheres, paths are shown as crosses in the spheres. The number next to the sphere indicates the unique cluster-ID.

Figure 4a shows the first time window, where seven clusters were identified. Clusters 1 and 2 are quite close to each other, but can still be clearly distinguished by the algorithm. Cluster 7 describes the propagation paths, which will be shadowed in the following windows. In the next time windows (b)-(d) cluster 4 is split into 2 clusters, because some of the components in cluster 4 are quite separated. The missing cluster 8 was also a short-living appearance of cluster 9. Moreover, all clusters move in the parameter space, most prominently, clusters 1 and 2 start to separate. In the last time windows (e)-(f) the previously shadowed cluster appears again and gets a new cluster-ID 10. Both, the identification and tracking were achieved fully automatic without any user interaction.

7 Conclusions

We demonstrated the performance of a novel cluster identification and tracking algorithm by clustering a synthetic environment obtained from the IlmProp channel model. The algorithm was able to correctly identify clusters and track their movement in the parameter space.

As we used the parametric channel data, we investigated the influence of a parameter estimator that achieves the Cramer-Rao lower bound. Surprisingly, for a nowadays standard channel measurement system, the variances of the estimates are very low. Hence, we can neglect estimation errors.

Future work is to let the clustering and tracking framework undergo the ultimate test of processing real-world measurement data. As the results from synthetic channels look quite convincing, we are quite positive for doing the next step.

Acknowledgements

This work was conducted within the EC funded network of excellence NEWCOM.

References

[1] K. Yu, Q. Li, D. Cheung, and C. Prettie, "On the tap and cluster angular spreads of indoor WLAN channels," in *Proceedings of IEEE Vehicular Technology Conference Spring 2004*, Milano, Italy,

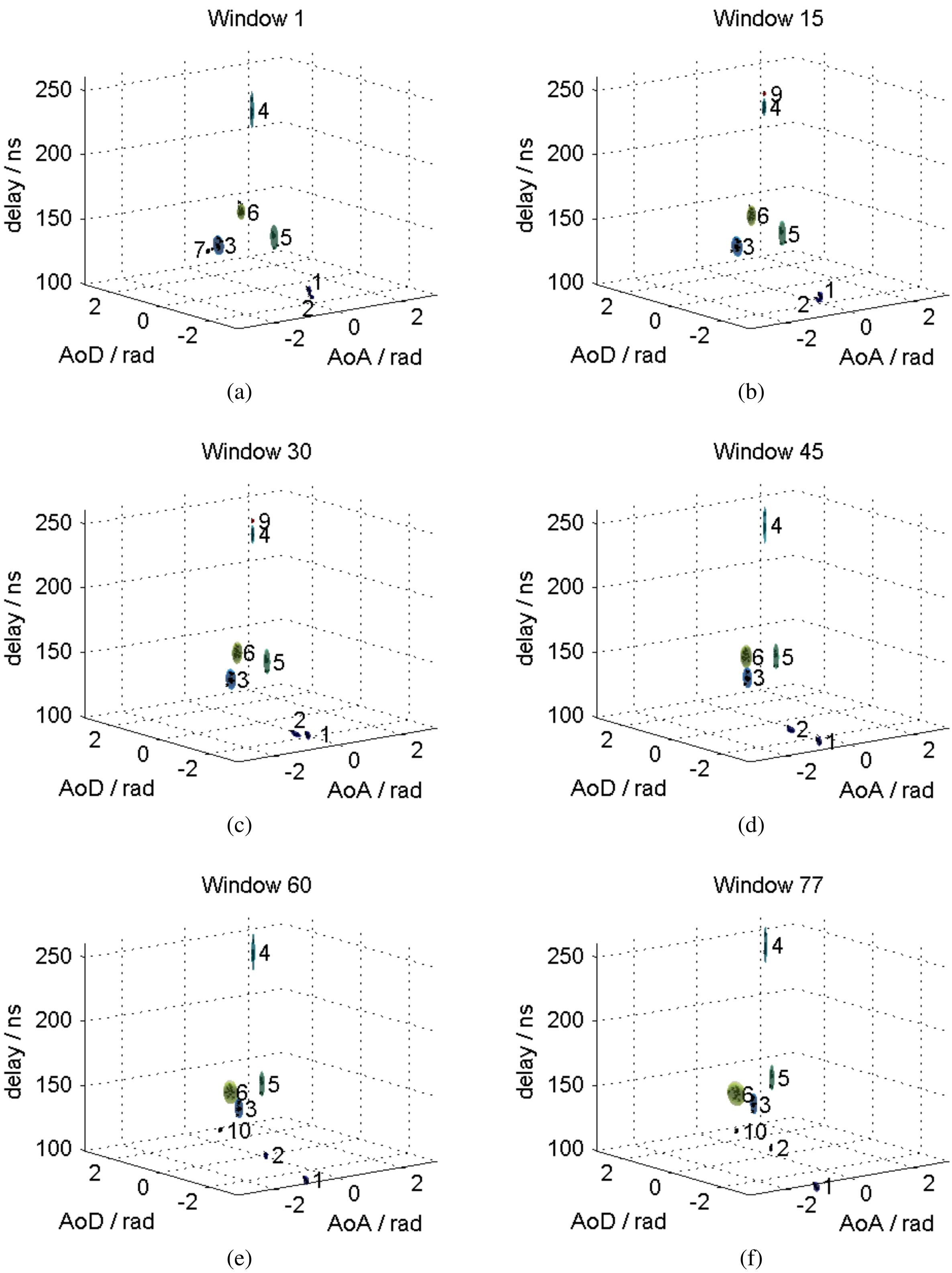

Figure 4: Results from cluster tracking: (a)-(f) represent different time windows in ascending order

May 17–19, 2004.

[2] C.-C. Chong, C.-M. Tan, D. Laurenson, S. McLaughlin, M. Beach, and A. Nix, "A new statistical wideband spatio-temporal channel model for 5-GHz band WLAN systems," *IEEE Journal on Selected Areas in Communications*, vol. 21, no. 2, pp. 139 – 150, Feb. 2003.

[3] J. Salo, J. Salmi, N. Czink, and P. Vainikainen, "Automatic clustering of nonstationary MIMO channel parameter estimates," *ICT'05*, May 2005, Cape Town, South Africa.

[4] N. Czink, P. Cera, J. Salo, E. Bonek, J.-P. Nuutinen, and J. Ylitalo, "Automatic clustering of MIMO channel parameters using the multi-path component distance measure," in *WPMC'05*, Aalborg, Denmark, Sept. 2005.

[5] N. Czink and P. Cera, "A novel framework for clustering parametric MIMO channel data including mpc powers," in *COST 273 Post-Project Meeting*, Lisbon, Portugal, Nov. 2005.

[6] M. Steinbauer, H. Özcelik, H. Hofstetter, C. Mecklenbräuker, and E. Bonek, "How to quantify multipath separation," *IEICE Trans. Electron.*, vol. E85, no. 3, pp. 552–557, March 2002.

[7] G. D. Galdo, M. Haardt, and C. Schneider, "Geometry-based channel modelling of MIMO channels in comparison with channel sounder measurements," *Advances in Radio Science - Kleinheubacher Berichte*, 2003.

[8] A. Richter, "Estimation of radio channel parameters: Models and algorithms," Ph.D. dissertation, Ilmenau University of Technology, 2005.