A framework for automatic clustering of parametric MIMO channel data including path powers

Nicolai Czink[†], Pierluigi Cera[†○], Jari Salo[‡], Ernst Bonek[†], Jukka-Pekka Nuutinen[*], Juha Ylitalo[**]
[†]Institut für Nachrichtentechnik und Hochfrequenztechnik, Technische Universität Wien, Austria
[○]ERASMUS Student from University of Bologna, Italy
[‡]Radio Laboratory/SMARAD, Helsinki University of Technology, Finland
[*]Elektrobit Testing Ltd., Oulu, Finland
[**]Centre for Wireless Communications, University of Oulu, Finland
nicolai.czink@nt.tuwien.ac.at

***Abstract*— We present a solution to the problem of identifying clusters from MIMO measurement data in a data window, with a minimum of user interaction. Conventionally, visual inspection has been used for the cluster identification. However this approach is impractical for a large amount of measurement data. Moreover, visual methods lack an accurate definition of a "cluster" itself.**

We introduce a framework that is able to cluster multi-path components (MPCs), decide on the number of clusters, and discard outliers. For clustering we use the K-means algorithm, which iteratively moves a number of cluster centroids through the data space to minimize the total difference between MPCs and their closest centroid. We significantly improve this algorithm by following changes: (i) as the distance metric we use the multi-path component distance (MCD), (ii) the distances are weighted by the powers of the MPCs. The implications of these changes result in a definition of a "cluster" itself that appeals to intuition.

We assess the performance of the new algorithm by clustering real-world measurement data from an indoor big hall environment.

***Keywords*—MIMO channel; MPC clustering; geometry-based stochastic channel models**

I. Introduction

Many advanced radio channel models base on the concept of multi-path clusters consisting of many multi-path components (MPCs) showing similar parameters such as azimuth and elevation of arrival and departure, and delay [1], [2]. The major problem of these models is the accurate parametrisation of clusters, where the parameters have to be extracted from measurement data. Currently there is no fully automatic clustering algorithm available to identify clusters from multi-dimensional parametric MIMO channel estimates. In many papers visual inspection of measurement data was used [3], [4], which becomes impractical for large amounts of measurement data. Recently, a semi-automatic algorithm was introduced in [5], which bases 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 [6], [7].

In this paper we propose to use a new framework consisting of three algorithms to significantly improve the clustering performance: (i) a new clustering algorithm, (ii) a cluster validation metric and (iii) an improved cluster shape pruning. This framework leads to an intrinsic and intuitive definition of a "cluster" itself. In the following we will describe the framework, its single components and their interaction. Finally we assess the performance of the new framework by applying it to measurement data.

II. Problem description

The starting point is a large number of multi-dimensional parametric channel estimation data, obtained from MIMO measurements. The measurements provide numerous snapshots of the impulse response of the – typically time-varying – radio channel. These measurements are fed to a high-resolution algorithm, e.g. SAGE [8], to estimate the channel parameters for each snapshot individually. It has been found in several MIMO studies that these parameters tend to appear in clusters, i.e. in groups of multi-path components (MPCs) with similar parameters, e.g. [3], [4]. The problem is to find an automatic procedure to identify and track these clusters.

We consider one data window with a number of L MPCs, where every single MPC is represented by its power P_l, $l = 1 \ldots L$, and a parameter vector $\mathbf{x}_l$ containing the delay (τ), azimuth and elevation AoA (φ_{AoA}, θ_{AoA}) and azimuth and elevation AoD (φ_{AoD}, θ_{AoD}). The data for all paths are collected in the vector $\mathbf{P} = [P_1 \ldots P_L]^T$ and the matrix $\mathbf{X} = [\mathbf{x}_1 \ldots \mathbf{x}_L]^T$.

III. Framework

For automatic clustering, a range for the expected number of clusters has to be specified first, then the algorithm assigns each MPC to a cluster and estimates the correct number of clusters ("cluster validation") (see Algorithm 1). Initially, a range $[K_{\text{min}}, K_{\text{max}}]$ for the possible number of clusters has to be specified. The number of clusters, K, and the data from all MPCs, $\mathbf{P}$ and $\mathbf{X}$, are external parameters for the clustering algorithm. For each possible K, the clustering algorithm KPowerMeans is performed, the results are collected in the data sets $\mathcal{R}_K$. Subsequently, each result is validated by CombinedValidate which provides the validation index v_K.

The optimum number of clusters K_{opt} is finally determined by the largest validation index v_K with corresponding cluster set $\mathcal{R}_{\text{opt}}$. This optimum set is then pruned by ShapePrune for improved visualisation.

```
Framework algorithm:
  1. Do For all number of clusters $K = K_{\text{min}}$ To $K_{\text{max}}$
       a. Cluster window with $K$ clusters:
          $\mathcal{R}_K = \text{KPowerMeans}(\mathbf{P}, \mathbf{X}, K)$
       b. Validate $K$ clusters:
          $v_K = \text{CombinedValidate}(\mathcal{R}_K)$
       c. Next $K$
  2. Find optimum number of clusters:
     $K_{\text{opt}} = \arg\max_K v_K$, $\mathcal{R}_{\text{opt}} = \mathcal{R}_{K_{\text{opt}}}$
  3. Prune optimum cluster set:
     $\mathcal{R}^p = \text{ShapePrune}(\mathcal{R}_{\text{opt}})$
```

Alg. 1: Framework Algorithm

```
KPowerMeans clustering algorithm:
  1. Randomly choose $K$ initial centroid positions
     $\mathbf{c}_1^{(0)}, \ldots, \mathbf{c}_K^{(0)}$
  2. For $i = 1$ To MaxIterations
       a. Assign MPCs to cluster centroids and store indices:
          $\mathcal{I}_l^{(i)} = \arg\min_k \{P_l \cdot \text{MCD}(\mathbf{x}_l, \mathbf{c}_k^{(i-1)})\}$,   (1)
          $\mathcal{I}^{(i)} = [\mathcal{I}_1^{(i)} \ldots \mathcal{I}_L^{(i)}]$, $\mathcal{C}_k^{(i)} = \text{Indices}_l(\mathcal{I}_l^{(i)}{=}k)$
       b. Recalculate cluster centroids $\mathbf{c}_k^{(i)}$ from the allocated MPCs
          to coincide with the clusters' centres of gravity:
          $\mathbf{c}_k^{(i)} = \frac{\sum_{j \in \mathcal{C}_k^{(i)}} (P_j \cdot \mathbf{x}_j)}{\sum_{j \in \mathcal{C}_k^{(i)}} P_j}$   (2)
       c. If $\mathbf{c}_k^{(i)} = \mathbf{c}_k^{(i-1)}$ for all $k = 1 \ldots K$, then GoTo 3.
          Else Next $i$
  3. Return $\mathcal{R}_K = [\mathcal{I}^{(i)}, \mathbf{c}_k^{(i)}]$
```

Alg. 2: KPowerMeans algorithm

A. Clustering algorithm — KPowerMeans

The task of the clustering algorithm is to assign a cluster index to each of the L MPCs. The concept of the K-means algorithm [9] is well suited for this challenge if one uses the appropriate distance function. Algorithm 2 describes the proposed KPowerMeans algorithm, which introduces the novelty of regarding powers of the MPCs.

This algorithm iteratively minimizes the total sum of power-weighted distances of each path to its associated cluster centroid. In the following the single steps of the algorithm are described in more detail.

Ad 1) The centroid starting positions are chosen randomly from the data $\mathbf{X}$.

Ad 2a) Every MPC is associated with a cluster centroid such that the function of the total sum of differences

$$D = \sum_{l=1}^{L} P_l \cdot \text{MCD}(\mathbf{x}_l, \mathbf{c}_{\mathcal{I}_l^{(i)}}) \tag{3}$$

is minimized. We use the MCD as the basic distance function [6], [10] but also include the *power of the paths*, which has not been considered in previous works. It can be shown that the global distance (3) can be minimized by the introduced algorithm, when using (1). The index $\mathcal{I}_l^{(i)}$ is the cluster number for the lth multi-path in the ith iteration step. Vice-versa, the set $\mathcal{C}_k^{(i)}$ contains the MPC indices belonging to the kth cluster in the ith iteration step.

By including power into the distance function, cluster centroids are pulled to points with strong powers. This is intuitive and yields massive performance improvements, to be demonstrated in Section IV. Considering receiver design one usually adresses the most dominant clusters, which are characterised by *power*. So, in development of MIMO transceiver algorithms, the weighting by power is quite natural. Furthermore, the global distance function (3) is an inherent definition for a cluster:

> *For a given number of clusters, clusters are chosen such that they minimize the total distance from their centroids.*

This implies that, for a given K, clusters are selected such that *the cluster angular and cluster delay spreads are minimized*, which is again intuitive.

Ad 2b) In the second step of the iteration, the centroids move to the centres of gravity of the groups of MPCs allocated in the previous step. Note that moving centroids can result in a new group of MPCs that will be associated with the centroid in the next iteration step.

Ad 2c) If the centroids do not move any more the algorithm has converged to a stable solution. Should this procedure take too much time, it stops after a maximum number of iterations.

Ad 3) The output of the algorithm is the index set $\mathcal{I}^{(i)}$ and the associated cluster centroids $\mathbf{c}^{(i)}$, which were obtained by the last iteration.

As usual, when using algorithms with random initial values, we perform KPowerMeans multiple times. The best result is determined by the smallest value of (3).

B. Cluster validation — CombinedValidate

For cluster validation we used a combination of two methods well-known in literature [11], the Caliñski-Harabasz index and the Davies-Bouldin criterion. Both indices and their proposed combination are described in the next paragraphs.

1) Caliñski-Harabasz index: When clustering L MPCs in K cluster, the Caliñski-Harabasz index (CH) is given as

$$\text{CH}(K) = \frac{\text{tr}(\mathbf{B})/(K-1)}{\text{tr}(\mathbf{W})/(L-K)} ,$$

which corresponds to the ratio between the traces of the *between-cluster scatter matrix* $\mathbf{B}$ and the *within-cluster scatter matrix* $\mathbf{W}$ [11]. Using the MCD as distance function, $\text{tr}(\mathbf{B})$ and $\text{tr}(\mathbf{W})$ are respectively given as

$$\text{tr}(\mathbf{B}) = \sum_{k=1}^{K} L_k \cdot \text{MCD}(\mathbf{c}_k, \overline{\mathbf{c}})^2 ,$$

$$\text{tr}(\mathbf{W}) = \sum_{k=1}^{K} \sum_{j \in \mathcal{C}_k} \text{MCD}(\mathbf{x}_j, \mathbf{c}_k)^2 ,$$

where L_k denotes the number of MPCs related to the kth cluster and

$$\bar{\mathbf{c}} = \frac{\sum_{l=1}^{L}(P_l \cdot \mathbf{x}_l)}{\sum_{l=1}^{L} P_l}$$

denotes the global centroid of the entire data set.

If we calculate the CH index for different values of K, e.g. in the range $[K_{\text{min}},\ K_{\text{max}}]$, the number of cluster K_{CH} corresponding to the best partition is achieved as

$$K_{\text{CH}} = \arg\max_{K}\{\text{CH}(K)\}\ , \tag{4}$$

corresponding to the partition with the most compact and separate cluster.

2) Davies-Bouldin index: The Davies-Bouldin index (DB) is a function of *intra-cluster compactness* and *inter-cluster separation* [11]. Using the MCD, the *compactness* S_k of the kth cluster is given as

$$S_k = \frac{1}{L_k}\sum_{l\in\mathcal{C}_k}\text{MCD}(\mathbf{x}_l, \mathbf{c}_k),$$

and the *separation*, i.e. the distance, between two centroids i and j, is defined as

$$d_{ij} = \text{MCD}(\mathbf{c}_i, \mathbf{c}_j)\,.$$

Finally the considered DB index is given as

$$\text{DB}(K) = \frac{1}{K}\sum_{i=1}^{K} R_i\ ,$$

where

$$R_i = \max_{\substack{j=1,\dots,K\\ j\neq i}}\left\{\frac{S_i + S_j}{d_{ij}}\right\}\ .$$

When calculating the DB index for different values of K, the optimum number of cluster K_{DB}, corresponding to the best partition, is achieved as

$$K_{\text{DB}} = \arg\min_{K}\{\text{DB}(K)\}\ .$$

As for the CH index, also the DB index bases on seeking for the partition with the most compact but separated clusters.

3) Combined Validation: A combination of the two introduced validation criteria yield significant performance improvements (see Section IV. The basic idea of the CombinedValidate (CV) index is to restrict valid choices of the optimum number of clusters by a threshold set in the DB index. Subsequently the CH index is used to decide on the optimum number out of the restricted set of possibilities.

We consider the set of feasible choices $\mathbf{F} = \{K_1, \dots, K_N\} \subseteq [K_{\text{min}},\ K_{\text{max}}]$ containing only the values K_i for which the following condition is satisfied,

$$\text{DB}(K_i) \leq t \cdot \min_{K}\{\text{DB}(K)\}\,,$$

where we chose $t = 2$. The optimum number of clusters K_{opt} is then obtained as

$$K_{\text{opt}} = \arg\max_{K\in\mathbf{F}}\{\text{CH}(K)\}\ .$$

In the unrestricted case $\mathbf{F} \equiv [K_{\text{min}},\ K_{\text{max}}]$ we obtain K_{opt} using (4).

ShapePrune algorithm:

```
1. Initialize pruned result set with optimum set: $\mathcal{R}^{(p)} = \mathcal{R}_{\text{opt}}$
2. For $k = 1$ To $K_{\text{opt}}$
     a. Save the original power and spread of the $k$th cluster:
        $P_k^{(0)} = \sum_{j\in\mathcal{C}_k} P_j$,
        $\mathbf{S}_k^{(0)} = [\sigma_\tau,\ \sigma_{\varphi_{\text{AoA}}},\ \sigma_{\varphi_{\text{AoD}}},\ \sigma_{\theta_{\text{AoA}}},\ \sigma_{\theta_{\text{AoD}}}]^T$
     b. While $P_k^{(\text{cur})} > p \cdot P_k^{(0)}$ And $\mathbf{S}_k^{(\text{cur})} > s \cdot \mathbf{S}_k^{(0)}$, remove the MPC with largest distance
          – Find MPC with largest distance to current centroid $\mathbf{c}_k$
          – Remove MPC from $\mathcal{R}^{(p)}$
          – Recalculate $P_k^{(\text{cur})}$ and $\mathbf{S}_k^{(\text{cur})}$.
     c. Restore the last deleted MPC
     d. Next $k$
3. Return $\mathcal{R}^{(p)}$
```

Alg. 3: ShapePrune Algorithm

C. Cluster pruning — ShapePrune

After successfully finding the optimum number of clusters, we use the ShapePrune cluster pruning algorithm for discarding outliers. This is achieved by removing data points that have largest distance from their own cluster centroid. As a constraint, cluster power and cluster spreads must not change significantly. This last condition allows to preserve the clusters' original power and shape, which is fundamental to achieve consistent results. The resulting algorithm is summarized in Algorithm 3.

For each cluster, the algorithm discards the MPCs with the largest distance to the cluster centroid, until one of the constraints is not fulfilled. The single steps of the algorithm are described in the following.

Ad 2a) Before starting to prune the kth cluster, the algorithm stores the original values of its cumulative power to $P_k^{(0)}$, and its (vector-valued) cluster spread to $\mathbf{S}_k^{(0)}$, where $\sigma_\tau,\ \sigma_{\varphi_{\text{AoA}}},\ \sigma_{\varphi_{\text{AoD}}},\ \sigma_{\theta_{\text{AoA}}},\ \sigma_{\theta_{\text{AoD}}}$ denote the rms cluster spreads of delay, and azimuth and elevation angles of departure and arrival of cluster k, respectively.

Ad 2b) Until the power of the cluster and its cluster spreads are below the two respective specified thresholds, the algorithm removes the MPC with largest distance to the centroid from the cluster, where we use the MCD as distance function. We define the power and spread thresholds as a fraction p of the original power and factor s of the original cluster spreads, respectively. Since we have to cope with a vector-valued spread, we define the condition $\mathbf{S}_k^{(\text{cur})} > s \cdot \mathbf{S}_k^{(0)}$ to be satisfied, when it holds true for all dimensions separately.

Ad 2c) Since we want the cluster power and spread to be larger than the specified thresholds, we have to restore the last pruned MPC. This implementation simply allows to speed up computation time.

Ad 3) The output of the algorithm is the pruned set of MPCs $\mathcal{R}^{(p)}$.

IV. Results

A. *Using MCD as distance function*

The advantage of using the MCD as distance function for clustering algorithms is discussed extensively in [7], where the performance is compared to different distance measures. It was shown that using the MCD significantly improves clustering performance.

B. *CombinedValidate*

We tested the performance of the cluster validation scheme at different angular cluster spreads. For this we used synthetic MIMO channel data obtained from the 3GPP spatial channel model (SCM) [12], implemented by [13], but we extended the model to cope with varying angular spreads. For the following evaluation, we used 200 different samples of MIMO channels with 6 clusters, where each cluster consisted of 8 MPCs.

Fig. 1 demonstrates the performance of the different cluster validation indices, i.e. the novel CombinedValidate, the Caliñski-Harabasz, and the Davies-Bouldin index. The Figure shows the fraction of the correctly estimated number of clusters versus the cluster angular spreads. The CH index has troubles with finding the correct number of clusters with low cluster spreads. On the other hand the DB index decreases with larger cluster spreads. The CombinedValidate index always outperforms the CH index and outperforms the DB index for cluster angular spreads larger than 2.5°. We demonstrated in [7] that the clustering framework almost always finds the true (simulated) clusters as long as the correct number of clusters is detected.

C. *KPowerMeans + CombinedValidate*

To test our clustering algorithm we used real-world MIMO measurements conducted with the wideband radio channel sounder PropSound CS in a big hall with 8 Tx and 16 Rx antennas. A description of the measurements is provided in [6]. In a post-processing step parametric channel estimates were obtained using the SAGE algorithm [8]. We consider a sample data set of a line-of-sight (LOS) and of a non-LOS (NLOS) measurement scenario.

Fig. 2 shows the considered LOS snapshot of the MIMO channel, MPCs are colour-coded with their power. Visual inspection gives the impression of nicely separated clusters in space. Applying our clustering framework without user interaction (yet without pruning) to this data, we obtain the result depicted in Fig. 3. The resulting partition into seven clusters realizes the optimum trade-off between cluster compactness and separation. Since the two small groups of MPCs, denoted by purple and light blue colour, represent an insignificant contribution to the total power, they are combined with the two larger clusters, represented by the corresponding colours. Surprisingly, the large group of MPCs (around 27

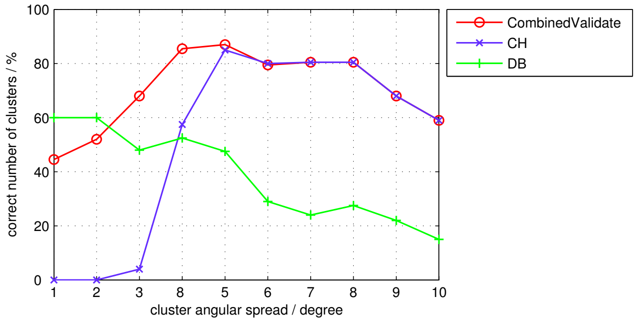

Fig. 1. Comparing performance of validity indices

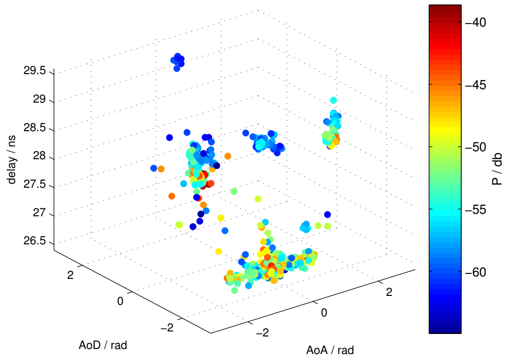

Fig. 2. Unclustered MIMO measurement data in LOS scenario; power of MPCs is colour-coded

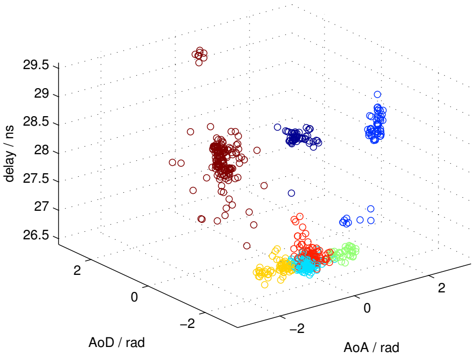

Fig. 3. Automatically clustered environment without pruning, 7 clusters were identified

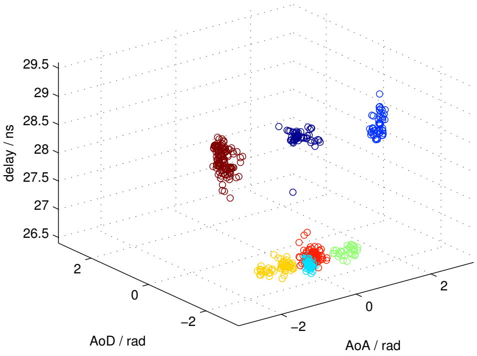

Fig. 4. Results of clustering: weak components were removed

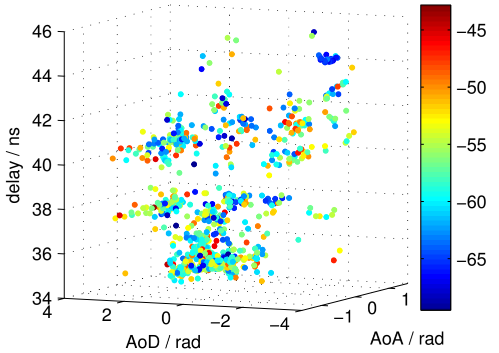

Fig. 5. Unclustered MIMO measurement data in NLOS scenario; power of MPCs

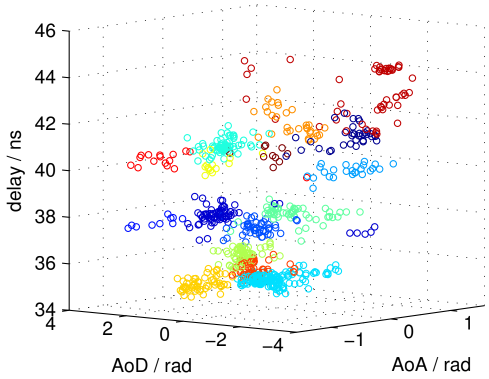

Fig. 6. Results of clustering after pruning: weak components were removed

ns), holding most of the total power, is split into four separate clusters. Cluster centres are attracted by strong powers. As MPCs powers in this group are strongly varying, it is most sensible to split up this group into several clusters. Here, the algorithm splits up clusters that could by visual inspection, perhaps, be considered one cluster.

Results of clustering this environment disregarding MPCs power is shown in [6].

D. KPowerMeans + CombinedValidate + ShapePrune

Using the pruning algorithm with $s = p = 0.9$, outlier paths are removed. Fig. 4 shows results of applying the whole framework algorithm including pruning. The clustering algorithm results in well-defined separable clusters. The pruning algorithm improves the visibility without changing cluster parameters. Clusters can be well identified, they are indicated as MPCs showing the same colour.

Obviously, the weak-powered cluster at large delay was pruned. This makes sense as its power did not add much to the channel. Also the large light blue cluster, around 28 ns now looks smaller, but still has similar properties to the original one.

As a second example we demonstrate the capabilities of the clustering algorithm for NLOS data which is far more challenging. Fig. 5 shows the considered snapshot of the channel, the clustering and pruning results are shown in Fig. 6. In this scenario 16 clusters were identified. Still, the resulting cluster set looks convincing.

V. Conclusions

One of the main problems in evaluating channel measurements is the identification of multi-path clusters.

We presented a scalable framework to automatically identify multi-path clusters from MIMO channel measurement data that is novel in four respects: (i) The framework algorithm enables to cluster MIMO channel parameters automatically with a minimum of user input; (ii) by including power and the MCD into the K-means concept, we make it applicable to clustering in propagation research; (iii) the cluster validation provides a trustworthy estimate of the correct number of clusters; (iv) the implemented cluster pruning algorithm does not change the cluster behaviour significantly, but improves visibility and future cluster tracking performance. Furthermore, the clustering algorithm introduces a convincing, inherent definition of a cluster itself.

We evaluated the performance of our clustering algorithm with both, synthetic and real-world MIMO channel data. We could demonstrate that the algorithm even outperforms visual inspection.

Acknowledgements

Part of this work was generously supported by Elektrobit Testing Ltd., as well as by the European Union funded Network-of-Excellence NEWCOM.

References

[1] A. Molisch, “Modeling the MIMO propagation channel,” *Belgian Journal of Electronics and Communications*, no. 4, pp. 5–14, 2003.

[2] L. Correia, Ed., *Mobile Broadband Multimedia Networks*. To be published by Elsevier, 2006.

[3] 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, May 17–19, 2004.

[4] 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.

[5] J. Salo, J. Salmi, N. Czink, and P. Vainikainen, “Automatic clustering of nonstationary MIMO channel parameter estimates,” May 2005, Cape Town, South Africa.

[6] 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.

[7] ——, “Improving clustering performance by using the multi-path component distance,” *IEE Electronics Letters*, vol. 42, no. 1, pp. 44–45, Jan. 2006.

[8] B. H. Fleury, M. Tschudin, R. Heddergott, D. Dahlhaus, and K. I. Pedersen, “Channel parameter estimation in mobile radio environments using the SAGE algorithm,” *IEEE JSAC*, no. 3, pp. 434–450, 17 1999.

[9] J. Han and M. Kamber, *Data Mining, Concepts, and Techniques*. Morgan Kaufmann Publishers, 2001.

[10] 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.

[11] U. Maulik and S. Bandyopadhyay, “Performance evaluation of some clustering algorithms and validity indices,” *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 24, no. 12, pp. 1650–1654, Dec. 2002.

[12] “Spatial channel model for Multiple Input Multiple Output (MIMO) simulations (3GPP TR 25.996), v6.1.0,” Sep. 2003. [Online]. Available: www.3gpp.org

[13] J. Salo, G. Del Galdo, J. Salmi, P. Kyösti, M. Milojevic, D. Laselva, and C. Schneider, “MATLAB implementation of the 3GPP spatial channel model (3GPP TR 25.996),” Online, jan 2005, available: http://www.tkk.fi/Units/Radio/scm/.