Application of the Evidence Procedure to the Estimation of Wireless Channels

Dmitriy Shutin*, Gernot Kubin*, Bernard H. Fleury**

***Abstract*— In this paper we address the application of the Bayesian Evidence Procedure to the estimation of wireless channels. The proposed scheme is based on Relevance Vector Machines (RVM) originally proposed by M. Tipping. RVMs allow to estimate channel parameters as well as to assess the number of multipath components constituting the channel within the Bayesian framework by locally maximizing the evidence integral. We show that in the case of channel sounding using pulse-compression techniques, it is possible to cast the channel model as a general linear model, thus allowing RVM methods to be applied. We extend the original RVM algorithm to the multiple-observation/multiple-sensor scenario by proposing a new graphical model to represent multipath components. Through the analysis of the Evidence Procedure we develop a thresholding algorithm that is used in estimating the number of components. We also discuss the relationship of the Evidence Procedure to the standard Minimum Description Length (MDL) criterion. We show that the maximum of the evidence corresponds to the minimum of the MDL criterion. The applicability of the proposed scheme is demonstrated with synthetic as well as real-world channel measurements, and a performance increase over the conventional MDL criterion applied to maximum-likelihood estimates of the channel parameters is observed.**

I. Introduction

DEEP understanding of wireless channels is an essential prerequisite to satisfy the ever-growing demand for fast information access over wireless systems. A wireless channel contains explicitly or implicitly all the information about the propagation environment. To ensure reliable communication, the transceiver should be constantly aware of the channel state. In order to make this task feasible, accurate channel models, which reproduce in a realistic manner the channel behavior, are required. However, efficient joint estimation of the channel parameters, e.g., number of the multipath components (model order), their relative delays, Doppler frequencies, directions of the impinging wavefronts, and polarizations is a particularly difficult task. It often leads to analytically intractable and computationally very expensive optimization procedures. The problem is often relaxed by assuming that the number of multipath components is fixed, which simplifies optimization in many cases [1], [2]. However, both underspecifying and overspecifying the model order leads to significant performance degradation: residual intersymbol interference impairs the performance of the decoder in the former case, while additive noise is injected in the channel equalizer in the latter: the excessive components amount only to the random fluctuations of the background noise. To amend this situation, empirical methods like cross-validation can be employed (see, for example [3]). Cross-validation selects the optimal model by measuring its performance over a validation data set and selecting the one that performs the best. In case of practical multipath channels, such data sets are often unavailable due to the time-variability of the channel impulse responses. Alternatively, one can employ model selection schemes in the spirit of Ockham's razor principle: simple models (in terms of the number of parameters involved) are preferred over more complex ones. Examples are the Akaike Information Criterion (AIC) and Minimum Description Length (MDL) [4], [5]. In this paper we show how the Ockham's principle can be effectively used to perform estimation of the channel parameters coupled with estimating the model order, i.e., the number of wavefronts.

Consider a certain class of parametric models (hypotheses) $\mathcal{H}_i$ defined as the collection of prior distributions $p(\boldsymbol{w}_i|\mathcal{H}_i)$ for the model parameters $\boldsymbol{w}_i$. Given the measurement data $\mathcal{Z}$ and a family of conditional distributions $p(\mathcal{Z}|\boldsymbol{w}_i, \mathcal{H}_i)$, our goal is to infer the hypothesis $\hat{\mathcal{H}}$ and the corresponding parameters $\hat{\boldsymbol{w}}$ that maximize the posterior

$$\{\hat{\boldsymbol{w}}, \hat{\mathcal{H}}\} = \underset{\boldsymbol{w}_i, \mathcal{H}_i}{\operatorname{argmax}} \Big\{ p(\boldsymbol{w}_i, \mathcal{H}_i|\mathcal{Z}) \Big\}. \tag{1}$$

The key to solving (1) lies in inferring the corresponding parameters $\boldsymbol{w}_i$ and $\mathcal{H}_i$ from the data $\mathcal{Z}$, which is often a nontrivial task. As far as the Bayesian methodology is concerned, there are two ways this inference problem can be solved [6, sec. 5]. In the *joint estimation method*, $p(\boldsymbol{w}_i, \mathcal{H}_i|\mathcal{Z})$ is maximized directly with respect to the quantities of interest $\boldsymbol{w}_i$ and $\mathcal{H}_i$. This often leads to computationally-intractable optimization algorithms. Alternatively, one can rewrite the posterior $p(\boldsymbol{w}_i, \mathcal{H}_i|\mathcal{Z})$ as

$$p(\boldsymbol{w}_i, \mathcal{H}_i|\mathcal{Z}) = p(\boldsymbol{w}_i|\mathcal{Z}, \mathcal{H}_i) p(\mathcal{H}_i|\mathcal{Z}) \tag{2}$$

and maximize each term on the right-hand side sequentially from right to left. This approach is known as the *marginal estimation method*. Marginal estimation methods (MEM) are well exemplified by Expectation-Maximization (EM) algorithms and used in many different signal processing applications (see [2], [3], [7]). MEMs are usually easier to compute, however they are prone to land in a local rather than global optimum. We recognize the first factor on the right-hand side of (2) as a parameter posterior, while the other one is a posterior

∗ Dmitriy Shutin and Gernot Kubin are with the Signal Processing and Speech Communication Laboratory, Graz University of Technology, Graz, Austria. Email: dshutin@tugraz.at, gernot.kubin@tugraz.at

∗∗ Bernard H. Fleury is with the Institute of Electronic Systems, Aalborg University, Fredrik Bajers Vej 7A, DK-9220 Aalborg, Denmark, and with Forschungszentrum Telekommunikation Wien (ftw.), Donau City Strasse 1, A-1220 Wien, Austria. Email:bfl@kom.aau.dk

for different model hypotheses. It is the maximization of $p(\mathcal{H}_i|\mathcal{Z})$ that guides our model selection decision. Then, the data analysis consists of two steps [8, ch. 28], [9]:

1) Inferring the parameters under the hypothesis $\mathcal{H}_i$

$$p(\boldsymbol{w}_i|\mathcal{Z},\mathcal{H}_i) = \frac{p(\mathcal{Z}|\boldsymbol{w}_i,\mathcal{H}_i)p(\boldsymbol{w}_i|\mathcal{H}_i)}{p(\mathcal{Z}|\mathcal{H}_i)} \equiv \frac{\text{Likelihood} \times \text{Prior}}{\text{Evidence}}. \tag{3}$$

2) Comparing different model hypotheses using the model posterior

$$p(\mathcal{H}_i|\mathcal{Z}) \propto p(\mathcal{Z}|\mathcal{H}_i)p(\mathcal{H}_i) \equiv \text{Evidence} \times \text{Hypothesis Prior}. \tag{4}$$

In the second stage, $p(\mathcal{H}_i)$ measures our subjective prior over different hypotheses before the data is observed. In many cases it is reasonable to assign equal probabilities to different hypotheses, thus reducing the hypothesis selection to selecting the model with the highest evidence $p(\mathcal{Z}|\mathcal{H}_i)$[1]. The evidence can be expressed as the following integral:

$$p(\mathcal{Z}|\mathcal{H}_i) = \int p(\mathcal{Z}|\boldsymbol{w}_i,\mathcal{H}_i)p(\boldsymbol{w}_i|\mathcal{H}_i)d\boldsymbol{w}_i. \tag{5}$$

The evidence integral (5) plays a crucial role in the development of Schwarz's approach to model order estimation [10] (Bayesian Information Criterion), as well as in a Bayesian interpretation of Rissanen's MDL principle and its variations [5], [11], [12]. Maximizing (5) with respect to the unknown model $\mathcal{H}_i$ is known as the evidence maximization procedure, or Evidence Procedure (EP) [13], [14].

Equations (3), (4) and (5) form the theoretical framework for our joint model and parameter estimation. The estimation algorithm is based on Relevance Vector Machines. Relevance Vector Machines (RVM), originally proposed by M. Tipping [15], are an example of the marginal estimation method that, for a set of hypotheses $\mathcal{H}_i$, iteratively approximates (1) by alternating between the model selection, i.e., maximizing (5) with respect to $\mathcal{H}_i$, and inferring the corresponding model parameters from maximization of (3). RVMs have been initially proposed to find sparse solutions to general linear problems. However, they can be quite effectively adapted to the estimation of the impulse response of wireless channels, thus resulting in an effective channel parameter estimation and model selection scheme within the Bayesian framework.

The material presented in the paper is organized as follows: Section II introduces the signal model of the wireless channel and the used notation, Section III explains the framework of the EP in the context of wireless channels. In Section IV we explain how model selection is implemented within the presented framework and discuss the relationship between the EP and the MDL criterion for model selection. Finally, Section V presents some application results illustrating the performance of the RVM-based estimator in synthetic as well as in actual wireless environments.

[1]In the Bayesian literature, the evidence is also known as the likelihood for the hypothesis $\mathcal{H}_i$.

II. Channel estimation using pulse-compression technique

Channel estimation usually consists of two steps: 1) sending a specific sounding sequence $s(t)$ through the channel and observing the response $y(t)$ at the other end, and 2) estimating the channel parameters from the matched-filtered received signal $z(t)$ (Fig. 1). It is common to represent the multipath channel response as the sum of delayed and weighted Dirac impulses, with each impulse representing one individual multipath component (see, for example, [16, sec. 5]. Such special structure of the channel impulse response implies that the filtered signal $z(t)$ should have a sparse structure. Unfortunately, this sparse structure is often obscured by additive noise and temporal dispersion due to the finite bandwidth of the transmitter and receiver hardware. This motivates the application of algorithms capable of recovering this sparse structure from the measurement data.

Let us consider an equivalent baseband channel sounding scheme shown in Fig. 1. The sounding signal $s(t)$ (Fig. 2) consists of periodically repeated burst waveforms $u(t)$, i.e., $s(t) = \sum_{i=-\infty}^{\infty} u(t - iT_f)$, where $u(t)$ has duration $T_u \leq T_f$ and is formed as $u(t) = \sum_{m=0}^{M-1} b_m p(t - mT_p)$. The sequence $b_0 \dots b_{M-1}$ is the known sounding sequence consisting of M chips, and $p(t)$ is the shaping pulse of duration T_p, $MT_p = T_u$. Furthermore, we assume that the receiver (Rx) is equipped

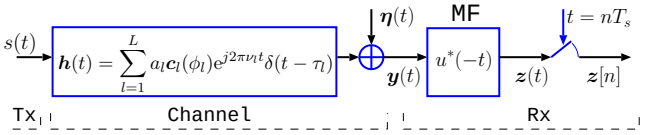


Fig. 1: An equivalent baseband model of the radio channel with receiver matched filter (MF) front-end.

with a planar antenna array consisting of P sensors located at positions $\boldsymbol{s}_1, \dots, \boldsymbol{s}_P \in \mathbb{R}^2$ with respect to an arbitrary reference point. Let us now assume that the maximum absolute

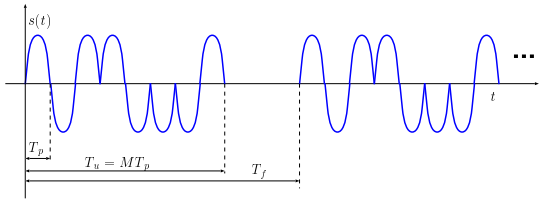


Fig. 2: Sounding sequence $s(t)$.

Doppler frequency of the impinging waves is much smaller than the inverse of a single burst duration $1/T_u$. This low Doppler frequency assumption is equivalent to assuming that, within a single observation window equivalent to the period of the sounding sequence, we can safely neglect the influence of the Doppler shifts.

The received signal vector $\boldsymbol{y}(t) \in \mathbb{C}^{P\times 1}$ for a single burst waveform is given as [2]:

$$\boldsymbol{y}(t) = \sum_{l=1}^{L} a_l \boldsymbol{c}(\phi_l) \mathrm{e}^{j2\pi\nu_l t} u(t - \tau_l) + \boldsymbol{\eta}(t).$$

Here, a_l, τ_l and ν_l are respectively the complex gain, the delay, and the Doppler shift of the lth multipath component. The P-dimensional complex vector $\boldsymbol{c}(\phi_l) = [c_1(\phi_l), \ldots, c_P(\phi_l)]^T$ is the steering vector of the array. Provided the coupling between the elements can be neglected, its components are given as $c_p(\phi_l) = f_p(\phi_l)\exp(j2\pi\lambda^{-1}\langle \boldsymbol{e}(\phi_l), \boldsymbol{s}_p\rangle)$ with λ, $\boldsymbol{e}(\phi_l)$ and $f_p(\phi_l)$ denoting the wavelength, the unit vector in $\mathbb{R}^2$ pointing in the direction of the incoming wavefront determined by the azimuth ϕ_l, and the complex electric field pattern of the pth sensor, respectively. The additive term $\boldsymbol{\eta}(t) \in \mathbb{C}^{P\times 1}$ is a vector-valued complex white Gaussian noise process, i.e., the components of $\boldsymbol{\eta}(t)$ are independent complex Gaussian processes with double-sided spectral density N_0.

The receiver front-end consists of a matched filter (MF) matched to the transmitted sequence $u(t)$. Under the low Doppler frequency assumption the term $\mathrm{e}^{j2\pi\nu_l t}$ stays time-invariant within a single burst duration, i.e., equal to a complex constant that can be incorporated in the complex gain a_l. The signal $\boldsymbol{z}(t)$ at the output of the MF is then given as

$$\boldsymbol{z}(t) = \sum_{l=1}^{L} a_l \boldsymbol{c}(\phi_l) R_{uu}(t-\tau_l) + \boldsymbol{\xi}(t), \tag{6}$$

where $R_{uu}(t) = \int u(t')u^*(t+t')dt'$ is the autocorrelation function of the burst waveform $u(t)$ and $\boldsymbol{\xi}(t) = \int \boldsymbol{\eta}(t')u^*(t+t')dt'$ is a spatially white P-dimensional vector with each element being a zero-mean wide-sense stationary (WSS) Gaussian noise with autocorrelation function

$$\begin{aligned} R_{\xi\xi}(t) =& E\{\xi_p(t')\xi_p^*(t+t')\} = N_0 R_{uu}(t), \quad \text{and} \\ & E\{\xi_p(t')\xi_p(t+t')\} = 0. \end{aligned} \tag{7}$$

Here $E\{\cdot\}$ denotes the expectation operator. Equation (6) states that the MF output is a linear combination of L scaled and delayed kernel functions $R_{uu}(t-\tau_l)$, weighted across sensors as given by the components of $\boldsymbol{c}(\phi_l)$ and observed in the presence of the colored noise $\boldsymbol{\xi}(t)$.

In practice, however, the output of the MF is sampled with the sampling period $T_s \leq T_p$, resulting in P N-tuples of the MF output, where N is the number of MF output samples. By collecting the output of each sensor into a vector, we can rewrite (6) in a vector form:

$$\boldsymbol{z}_p = \boldsymbol{K}\boldsymbol{w}_p + \boldsymbol{\xi}_p, \quad p = 1\ldots P, \tag{8}$$

where we have defined

$$\begin{aligned} \boldsymbol{z}_p =& [z_p[0], z_p[1], \ldots, z_p[N-1]]^T, \\ \boldsymbol{w}_p =& [a_1 c_p(\phi_1), \ldots, a_L c_p(\phi_L)]^T, \\ \boldsymbol{\xi}_p =& [\xi_p[0], \xi_p[1], \ldots, \xi_p[N-1]]^T. \end{aligned}$$

The additive noise vectors $\boldsymbol{\xi}_p$, $p = 1\ldots P$, possess the following properties that will be exploited later:

$$E\{\boldsymbol{\xi}_p\} = \boldsymbol{0}, E\{\boldsymbol{\xi}_m\boldsymbol{\xi}_k^H\} = \boldsymbol{0}, \text{for } m \neq k, \text{ and} \tag{9}$$

$$E\{\boldsymbol{\xi}_p\boldsymbol{\xi}_p^H\} = \boldsymbol{\Sigma} = N_0\boldsymbol{\Lambda}, \text{where } \Lambda_{i,j} = R_{uu}((i-j)T_s). \tag{10}$$

Note that (10) follows directly from (7). The matrix $\boldsymbol{K}$, also called the design matrix, accumulates the shifted and sampled versions of the kernel function $R_{uu}(t)$. It is constructed as $\boldsymbol{K} = [\boldsymbol{r}_1, \ldots, \boldsymbol{r}_L]$, with $\boldsymbol{r}_l = [R_{uu}(-\tau_l), R_{uu}(T_s - \tau_l), \ldots, R_{uu}((N-1)T_s - \tau_l)]^T$.

In general, the channel estimation problem is posed as follows: given the measured sampled signals $\boldsymbol{z}_p, p = 1\ldots P$, determine the order L of the model and estimate optimally (with respect to some quality criterion) all multipath parameters a_l, τ_l, and ϕ_l, for $l = 1\ldots L$. In this contribution we restrict ourselves to the estimation of the model order L along with the vector $\boldsymbol{w}_p$, rather than of the constituting parameters τ_l, ϕ_l, and a_l. We will also quantize, although arbitrarily fine[2], the search space for the multipath delays τ_l. Thus, we do not try to estimate the path delays with infinite resolution, but rather fix the delay values to be located on a grid with a given mesh determining the quantization error. The size of the delay search space L_0 and the resulting quantized delays $\mathcal{T} = \{T_1, \ldots, T_{L_0}\}$ form the initial model hypothesis $\mathcal{H}_0$, which would manifest itself in the L_0 columns of the design matrix $\boldsymbol{K}$. This allows to formulate the channel estimation problem as a standard linear problem to which the RVM algorithm can be applied.

As it can be seen, our idea lies in finding the closest approximation of the continuous-time model (6) with the discrete-time equivalent (8). By incorporating the model selection in the analysis, we also strive to find the most compact representation (in terms of the number of components), while preserving good approximation quality. Thus, our goal is to estimate the channel parameters $\boldsymbol{w}_p$ as well as to determine how many multipath components $L \leq L_0$ are present in the measured impulse response. The application of the RVM framework to solve this problem follows in the next section.

III. Evidence maximization, Relevance Vector Machines and wireless channels

We begin our analysis following the steps outlined in Section I. In order to ease the algorithm description we first assume that $P = 1$, i.e., only a single sensor is used. Extensions to the case $P > 1$ is carried out later in Section III-B. To simplify the notations we also drop the subscript index p in our further notations.

From (8) it follows that the observation vector $\boldsymbol{z}$ is a linear combination of the vectors from the column-space of $\boldsymbol{K}$, weighted according to the parameters $\boldsymbol{w}$ and embedded in the correlated noise $\boldsymbol{\xi}$. In order to correctly assess the order of the model, it is imperative to take the noise process into account. It follows from (10) that the covariance matrix of the noise is proportional to the unknown spectral height N_0, which should therefore be estimated from the data. Thus, the model hypotheses $\mathcal{H}_i$ should include the term N_0. In the following analysis we assume that $\beta = N_0^{-1}$ is Gamma-distributed [15], with the corresponding probability density function (pdf) given as

$$p(\beta|\kappa, \upsilon) = \frac{\kappa^\upsilon}{\Gamma(\upsilon)}\beta^{\upsilon-1}\exp(-\kappa\beta), \tag{11}$$

with parameters κ and υ predefined so that (11) accurately reflects our *a priori* information about N_0. In the absence of

[2]There is actually a limit beyond which it makes no sense to make the search grid finer, since it will not decrease the variance of the estimates, which is lower-bounded by the Crammer-Rao bound [2].

any *a priori* knowledge one can make use of a non-informative (i.e., flat in the logarithmic domain) prior by fixing the parameters to small values $\kappa = \upsilon = 10^{-4}$ [15]. Furthermore, to steer the model selection mechanism, we introduce an extra parameter (hyperparameter) $\alpha_l,\ l = 1 \dots L_0$, for each column in $\boldsymbol{K}$. This parameter measures the contribution or relevance of the corresponding weight w_l in explaining the data $\boldsymbol{z}$ from the likelihood $p(\boldsymbol{z}|\boldsymbol{w}_i, \mathcal{H}_i)$. This is achieved by specifying the prior $p(\boldsymbol{w}|\boldsymbol{\alpha})$ for the model weights:

$$p(\boldsymbol{w}|\boldsymbol{\alpha}) = \prod_{l=1}^{L_0} \frac{\alpha_l}{\pi} \exp(-|w_l|^2 \alpha_l). \tag{12}$$

High values of α_l will render the contribution of the corresponding column in the matrix $\boldsymbol{K}$ 'irrelevant', since the weight w_l is likely to have a very small value (hence they are termed *relevance hyperparameters*). This will enable us to prune the model by setting the corresponding weight w_l to zero, thus effectively removing the corresponding column from the matrix and the corresponding delay T_l from the delay search space $\mathcal{T}$. We also see that α_l^{-1} is nothing else as the prior variance of the model weight w_l. Also note that the prior (12) implicitly assumes statistical independence of the multipath contributions.

To complete the Bayesian framework, we also specify the prior over the hyperparameters. Similarly to the noise contribution, we assume the hyperparameters α_l to be Gamma-distributed with the corresponding pdf

$$p(\boldsymbol{\alpha}|\zeta, \epsilon) = \prod_{l=1}^{L} \frac{\zeta^{\epsilon}}{\Gamma(\epsilon)} \alpha_l^{\epsilon-1} \exp(-\zeta \alpha_l), \tag{13}$$

where ζ and ϵ are fixed at some values that ensure an appropriate form of the prior. Again, we can make this prior non-informative by fixing ζ and ϵ to small values, e.g., $\epsilon = \zeta = 10^{-4}$.

Now, let us define the hypothesis $\mathcal{H}_i$ more formally. Let $\mathcal{P}(\mathcal{S})$ be a power set consisting of all possible subsets of basis vector indices $\mathcal{S} = \{1 \dots L_0\}$, and $i \mapsto \mathcal{P}(i)$ be the indexing of $\mathcal{P}(\mathcal{S})$ such that $\mathcal{P}(0) = \mathcal{S}$. Then for each index value i the hypothesis $\mathcal{H}_i$ is the set $\mathcal{H}_i = \{\beta; \alpha_j, j \in \mathcal{P}(i)\}$. Clearly, the initial hypothesis $\mathcal{H}_0 = \{\beta; \alpha_j, j \in \mathcal{S}\}$ includes all possible potential basis functions.

Now we are ready to outline the learning algorithm that estimates the model parameters $\boldsymbol{w}$, β, and hyperparameters $\boldsymbol{\alpha}$ from the measurement data $\boldsymbol{z}$.

A. Learning algorithm

Basically, learning consists of inferring the values of $\boldsymbol{w}_i$ and the hypothesis $\mathcal{H}_i$ that maximize the posterior (2): $p(\boldsymbol{w}_i, \mathcal{H}_i|\mathcal{Z}) \equiv p(\boldsymbol{w}_i, \boldsymbol{\alpha}_i, \beta|\boldsymbol{z})$. Here $\boldsymbol{\alpha}_i$ denotes the vector of all evidence hyperparameters associated with the ith hypothesis. The latter expression can also be rewritten as

$$p(\boldsymbol{w}, \boldsymbol{\alpha}, \beta|\boldsymbol{z}) = p(\boldsymbol{w}|\boldsymbol{z}, \boldsymbol{\alpha}, \beta) p(\boldsymbol{\alpha}, \beta|\boldsymbol{z}). \tag{14}$$

The explicit dependence on the hypothesis index i has been dropped to simplify the notation. We recognize that the first term $p(\boldsymbol{w}|\boldsymbol{z}, \boldsymbol{\alpha}, \beta)$ in (14) is the weight posterior and the other one $p(\boldsymbol{\alpha}, \beta|\boldsymbol{z})$ is the hypothesis posterior. From this point we can start with the Bayesian two-step analysis as has been indicated before.

Assuming the parameters $\boldsymbol{\alpha}$ and β are known, estimation of model parameters consists of finding values $\boldsymbol{w}$ that maximize $p(\boldsymbol{w}|\boldsymbol{z}, \boldsymbol{\alpha}, \beta)$. Using Bayes' rule we can rewrite this posterior as

$$p(\boldsymbol{w}|\boldsymbol{z}, \boldsymbol{\alpha}, \beta) \propto p(\boldsymbol{z}|\boldsymbol{w}, \boldsymbol{\alpha}, \beta) p(\boldsymbol{w}|\boldsymbol{\alpha}, \beta). \tag{15}$$

Consider the Bayesian graphical model [17] in Fig. 3. This graph captures the relationship between different variables involved in (14). It is a useful tool to represent the dependencies among the variables involved in the analysis in order to factor the joint density function into contributing marginals.

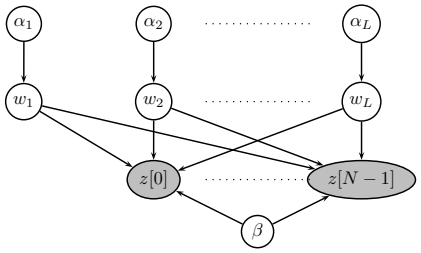


Fig. 3: Graph representing the discrete-time model of the wireless channel.

It immediately follows from the structure of the graph in Fig. 3 that $p(\boldsymbol{z}|\boldsymbol{w}, \boldsymbol{\alpha}, \beta) = p(\boldsymbol{z}|\boldsymbol{w}, \beta)$ and $p(\boldsymbol{w}|\boldsymbol{\alpha}, \beta) = p(\boldsymbol{w}|\boldsymbol{\alpha})$, i.e., $\boldsymbol{z}$ and $\boldsymbol{\alpha}$ are conditionally independent given $\boldsymbol{w}$ and β, and $\boldsymbol{w}$ and β are conditionally independent given $\boldsymbol{\alpha}$. Thus, (15) is equivalent to

$$p(\boldsymbol{w}|\boldsymbol{z}, \boldsymbol{\alpha}, \beta) \propto p(\boldsymbol{z}|\boldsymbol{w}, \beta) p(\boldsymbol{w}|\boldsymbol{\alpha}), \tag{16}$$

where the second factor on the right-hand side is given in (12). The first term is the likelihood of $\boldsymbol{w}$ and β given the data. From (8) it follows that

$$p(\boldsymbol{z}|\boldsymbol{w}, \beta) = \frac{\exp\{-(\boldsymbol{z} - \boldsymbol{K}\boldsymbol{w})^H \beta \boldsymbol{\Lambda}^{-1} (\boldsymbol{z} - \boldsymbol{K}\boldsymbol{w})\}}{\pi^N |\beta^{-1} \boldsymbol{\Lambda}|}.$$

Since both right-hand factors in (16) are Gaussian densities, $p(\boldsymbol{w}|\boldsymbol{z}, \boldsymbol{\alpha}, \beta)$ is also a Gaussian density with the covariance matrix $\boldsymbol{\Phi}$ and mean $\boldsymbol{\mu}$ given as

$$\boldsymbol{\Phi} = (\boldsymbol{A} + \beta \boldsymbol{K}^H \boldsymbol{\Lambda}^{-1} \boldsymbol{K})^{-1}. \tag{17}$$

$$\boldsymbol{\mu} = \beta \boldsymbol{\Phi} \boldsymbol{K}^H \boldsymbol{\Lambda}^{-1} \boldsymbol{z}, \tag{18}$$

The matrix $\boldsymbol{A} = \mathrm{diag}(\boldsymbol{\alpha})$ is a diagonal matrix that contains the evidence parameters α_l on its main diagonal. Clearly, $\boldsymbol{\mu}$ is a maximum *a-posteriori* (MAP) estimate of the parameter vector $\boldsymbol{w}$ under the hypothesis $\mathcal{H}_i$, with $\boldsymbol{\Phi}$ being the covariance matrix of the resulting estimates. This completes the model fitting step.

Our next step is to find parameters $\boldsymbol{\alpha}$ and β that maximize the hypothesis posterior $p(\boldsymbol{\alpha}, \beta|\boldsymbol{z})$ in (14). This density function can be represented as $p(\boldsymbol{\alpha}, \beta|\boldsymbol{z}) \propto p(\boldsymbol{z}|\boldsymbol{\alpha}, \beta) p(\boldsymbol{\alpha}, \beta)$, where $p(\boldsymbol{z}|\boldsymbol{\alpha}, \beta)$ is the evidence term and $p(\boldsymbol{\alpha}, \beta) = p(\boldsymbol{\alpha}) p(\beta)$ is the hypothesis prior. As it was mentioned earlier, it is quite

reasonable to choose non-informative priors since we would like to give all possible hypotheses $\mathcal{H}_i$ an equal chance of being valid. This can be achieved by setting ζ, ϵ, κ, and υ to very small values. In fact, it can be easily concluded (see derivations in the Appendix) that maximum of the evidence $p(\boldsymbol{z}|\boldsymbol{\alpha},\beta)$ coincides with the maximum of $p(\boldsymbol{z}|\boldsymbol{\alpha},\beta)p(\boldsymbol{\alpha},\beta)$ when $\zeta=\epsilon=\kappa=\upsilon=0$, which effectively results in the noninformative hyperpriors for $\boldsymbol{\alpha}$ and β.

This formulation of prior distributions is related to automatic relevance determination (ARD) [14], [18]. As a consequence of this assumption, the maximization of the model posterior is equivalent to the maximization of the evidence, which is known as the Evidence Procedure [13].

The evidence term $p(\boldsymbol{z}|\boldsymbol{\alpha},\beta)$ can be expressed as

$$\begin{aligned} p(\boldsymbol{z}|\boldsymbol{\alpha},\beta) &= \int p(\boldsymbol{z}|\boldsymbol{w},\beta)p(\boldsymbol{w}|\boldsymbol{\alpha})d\boldsymbol{w} \\ &= \frac{\exp\Big(-\boldsymbol{z}^H(\beta^{-1}\boldsymbol{\Lambda}+\boldsymbol{K}\boldsymbol{A}^{-1}\boldsymbol{K}^H)^{-1}\boldsymbol{z}\Big)}{\pi^N|\beta^{-1}\boldsymbol{\Lambda}+\boldsymbol{K}\boldsymbol{A}^{-1}\boldsymbol{K}^H|}, \end{aligned} \tag{19}$$

which is equivalent to (5), where conditional independencies between variables have been used to simplify the integrands. In the Bayesian literature this quantity is known as *marginal likelihood* and its maximization with respect to the unknown hyperparameters $\boldsymbol{\alpha}$ and β is a *type-II maximum likelihood* method [19]. To ease the optimization, several terms in (19) can be expressed as a function of the weight posterior parameters $\boldsymbol{\mu}$ and $\boldsymbol{\Phi}$ as given by (17) and (18). Then, by taking the derivatives of the logarithm of (19) with respect to $\boldsymbol{\alpha}$ and β and by setting them to zero, we obtain its maximizing values as (see also Appendix)

$$\alpha_l = \frac{1}{\Phi_{ll}+|\mu_l|^2}, \tag{20}$$

$$\beta^{-1} = \frac{\mathrm{tr}[\boldsymbol{\Phi}\boldsymbol{K}^H\boldsymbol{\Lambda}^{-1}\boldsymbol{K}]+(\boldsymbol{z}-\boldsymbol{K}\boldsymbol{\mu})^H\boldsymbol{\Lambda}^{-1}(\boldsymbol{z}-\boldsymbol{K}\boldsymbol{\mu})}{N}. \tag{21}$$

In (20) μ_l and Φ_{ll} denote the lth element of, respectively, the vector $\boldsymbol{\mu}$, and the main diagonal of the matrix $\boldsymbol{\Phi}$. Unlike the maximizing values obtained in the original RVM paper [15, eq.(18)], (21) is derived for the extended, more general case of colored additive noise $\boldsymbol{\xi}$ with the corresponding covariance matrix $\beta^{-1}\boldsymbol{\Lambda}$ arising due to the MF processing at the receiver. Clearly, if the noise is assumed to be white, expressions (20) and (21) coincide with those derived in [15]. Also note that $\boldsymbol{\alpha}$ and β are dependent as it can be seen from (20) and (21).

Thus, for a particular hypothesis $\mathcal{H}_i$ the learning algorithm proceeds by repeated application of (17) and (18), alternated with the update of the corresponding evidence parameters $\boldsymbol{\alpha}_i$ and β from (20) and (21), as depicted in Fig. 4, until some suitable convergence criterion has been satisfied. Provided a good initialization $\boldsymbol{\alpha}_i^{[0]}$ and $\beta^{[0]}$ is chosen[3], the scheme in Fig. 4 converges after j iterations to the stationary point of the system of coupled equations (17), (18), (20), and (21). Then, the maximization (1) is performed by selecting the hypothesis that results in the highest posterior (2).

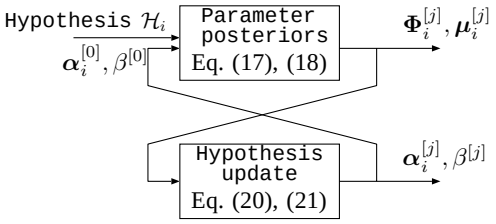


Fig. 4: Iterative learning of the parameters; The superscript $[j]$ denotes the iteration index.

In practice, however, we will observe that during the re-estimation some of the hyperparameters α_l diverge, or, in fact, become numerically indistinguishable from infinity given the computer accuracy[4]. The divergence of some of the hyperparameters enables us to approximate (1) by performing an on-line model selection: starting from the initial hypothesis $\mathcal{H}_0$, we prune the hyperparameters that become larger than a certain threshold as the iterations proceed by setting them to infinity. In turn, this sets the corresponding coefficient w_l to zero, thus "switching off" the lth column in the kernel matrix $\boldsymbol{K}$ and removing the delay T_l from the search space $\mathcal{T}$. This effectively implements the model selection by creating smaller hypotheses $\mathcal{H}_i < \mathcal{H}_0$ (with fewer basis functions) without performing an exhaustive search over all the possibilities. The choice of the threshold will be discussed in Section IV.

B. Extensions to multiple channel observations

In this subsection we extend the above analysis to multiple channel observations or multiple antenna systems. When detecting multipath components any additional channel measurement (either in time, by observing several periods of the sounding sequence $u(t)$, or in space, by using multiple sensor antenna) can be used to increase detection quality. Of course, it is important to make sure that the multipath components are time-invariant within the observation interval. The basic idea how to incorporate several channel observations is quite simple: in the original formulation each hyperparameter α_l was used to control a single weight w_l and thus the single component. Having several channel observations, a single hyperparameter α_l now controls weights representing contribution of *the same* physical multipath component, but present in the different channel observations.

Usage of a single parameter in this case expresses the channel coherence property in the Bayesian framework. The corresponding graphical model that illustrates this idea for a single hyperparameter α_l is depicted in Fig. 5. It is interesting to note that similar ideas, though in a totally different context, were adapted to train neural networks by allowing a single hyperparameter to control a group of weights [18]. Note that it is also possible to introduce an individual hyperparameter $\alpha_{p,l}$ for each weight $w_{p,l}$, but this eventually decouples the problem into P separate one-dimensional problems and as the

[3]Later in Section V we consider several rules for initializing the hyperparameters.

[4]In the finite sample size case, however, this will only happen in the high SNR regime. Otherwise, α_l will take large but still finite values. In Section IV-A we elaborate more on the conditions that lead to convergence/divergence of this learning scheme.

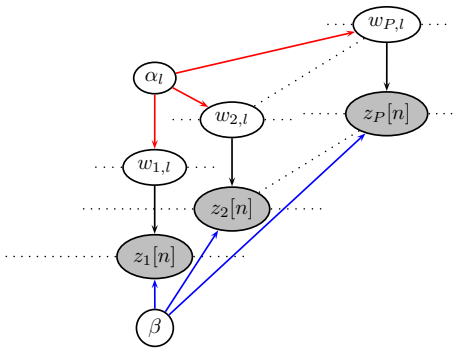


Fig. 5: Usage of α_l in a multiple-observation discrete-time wireless channel model to represent P coherent channel measurements.

result any dependency between the consecutive channels is ignored.

Now, let us return to (8). It can be seen that the weights $\boldsymbol{w}_p$ capture the structure induced by multiple antennas. However, for the moment we ignore this structure and treat the components of $\boldsymbol{w}_p$ as a wide-sense stationary (WSS) process over the individual channels, $p = 1 \dots P$. We will also allow each sensor to have a different MF. This might not necessarily be the case for wireless channel sounding, but thus a more general situation can be considered. Different matched filters result in different design matrices $\boldsymbol{K}_p$, and thus different noise covariance matrices $\boldsymbol{\Sigma_p}$, $p = 1 \dots P$. We will however require that the variance of the input noise remains the same and equals $N_0 = \beta^{-1}$ for all channels, so that $\boldsymbol{\Sigma_p} = N_0 \boldsymbol{\Lambda}_p$, and the noise components are statistically independent among the channels. Then, by defining

$$\tilde{\boldsymbol{\Sigma}} = \beta^{-1} \begin{bmatrix} \boldsymbol{\Lambda}_1 & & \boldsymbol{0} \\ & \ddots & \\ \boldsymbol{0} & & \boldsymbol{\Lambda}_P \end{bmatrix}, \quad \tilde{\boldsymbol{A}} = \underbrace{\begin{bmatrix} \boldsymbol{A} & & \boldsymbol{0} \\ & \ddots & \\ \boldsymbol{0} & & \boldsymbol{A} \end{bmatrix}}_{P \times P \text{ block matrix}},$$

$$\tilde{\boldsymbol{K}} = \begin{bmatrix} \boldsymbol{K}_1 & & \boldsymbol{0} \\ & \ddots & \\ \boldsymbol{0} & & \boldsymbol{K}_P \end{bmatrix}, \quad \tilde{\boldsymbol{z}} = \begin{bmatrix} \boldsymbol{z}_1 \\ \vdots \\ \boldsymbol{z}_P \end{bmatrix}, \quad \tilde{\boldsymbol{w}} = \begin{bmatrix} \boldsymbol{w}_1 \\ \vdots \\ \boldsymbol{w}_P \end{bmatrix}, \tag{22}$$

we rewrite equation (8) as

$$\tilde{\boldsymbol{z}} = \tilde{\boldsymbol{K}} \tilde{\boldsymbol{w}} + \tilde{\boldsymbol{\xi}}. \tag{23}$$

A crucial point of this system representation is that the hyperparameters α_l are shared by P channels as it can be seen in the structure of the matrix $\tilde{\boldsymbol{A}}$. This will have a corresponding effect on the hyperparameter re-estimation algorithm.

From the structural equivalence of (8) and (23) we can easily infer that equations (17) and (18) are modified as follows:

$$\boldsymbol{\Phi}_p = (\boldsymbol{A} + \beta \boldsymbol{K}_p^H \boldsymbol{\Lambda}_p^{-1} \boldsymbol{K}_p)^{-1}, \tag{24}$$

$$\boldsymbol{\mu}_p = \beta \boldsymbol{\Phi}_p \boldsymbol{K}_p^H \boldsymbol{\Lambda}_p^{-1} \boldsymbol{z}_p, \quad p = 1 \dots P. \tag{25}$$

The expressions for the hyperparameter updates become a bit more complicated but are still straight-forward to compute. It is shown in the Appendix that:

$$\alpha_l = \frac{P}{\sum_{p=1}^{P} \left(\Phi_{p,ll} + |\mu_{p,l}|^2 \right)}, \tag{26}$$

$$N_0 = \beta^{-1} = \frac{1}{NP} \left(\sum_{p=1}^{P} \operatorname{tr}[\boldsymbol{\Phi}_p \boldsymbol{K}_p^H \boldsymbol{\Lambda}_p^{-1} \boldsymbol{K}_p] + \right. \\ \left. + \sum_{p=1}^{P} (\boldsymbol{z}_p - \boldsymbol{K}_p \boldsymbol{\mu}_p)^H \boldsymbol{\Lambda}_p^{-1} (\boldsymbol{z}_p - \boldsymbol{K}_p \boldsymbol{\mu}_p) \right) \tag{27}$$

where $\mu_{p,l}$ is the lth element of the MAP estimate of the parameter vector $\boldsymbol{w}_p$ given by (25), and $\Phi_{p,ll}$ is the lth element on the main diagonal of $\boldsymbol{\Phi}_p$ from (24). Comparing the latter expressions with those developed for the single channel case, we observe that (26) and (27) use multiple channels to improve the estimates of the noise spectral height and channel weight hyperparameters. They also offer more insight into the physical meaning of the hyperparameters $\boldsymbol{\alpha}$. On the one hand, the hyperparameters are used to regularize the matrix inversion (24), needed to obtain the MAP estimates of the parameters $w_{p,l}$ and their corresponding variances. On the other hand, they act as the inverse of the second noncentral moments of the coefficients $w_{p,l}$, as can be seen from (26).

IV. Model Selection and Basis Pruning

The ability to select the best model to represent the measured data is an important feature of the proposed scheme, and thus it is paramount to consider in more detail how the model selection is effectively achieved. In Section III-A we have briefly mentioned that during the learning phase many of the hyperparameters α_l's tend to large values, meaning that the corresponding weights w_l's will cluster around zero according to the prior (12). This will allow us to set these coefficients to zero, thus effectively pruning the corresponding basis function from the design matrix. However the question how large a hyperparameter has to grow in order to prune its corresponding basis function has not yet been discussed. In the original RVM paper [15], the author suggests using a threshold α_{th} to prune the model. The empirical evidence collected by the author suggests setting the threshold to "a sufficiently large number" (e.g., $\alpha_{\text{th}} = 10^{12}$). However, our theoretical analysis presented in the following section will show that such high thresholds are only meaningful in very high SNR regimes, or if the number of channel observations P is sufficiently large. In more general, and often more realistic, scenarios such high thresholds are absolutely impractical. Thus, there is a need to study the model selection problem in the context of the presented approach more rigorously.

Below, we present two methods for implementing model selection within the proposed algorithm. The first method relies on the statistical properties of the hyperparameters α_l, when the update equations (24), (25), (26), and (27) converge to a stationary point. The second method exploits the relationship that we will establish between the proposed scheme and the Minimum Description Length principle [4], [8], [20], [21], thus linking the EP to this classical model selection approach.

A. Statistical analysis of the hyperparameters in the stationary point

The decision to keep or to prune a basis function from the design matrix is based purely on the value of the corresponding hyperparameter α_l. In the following we analyze the convergence properties of the iterative learning scheme depicted in Fig. 4 using expressions (24), (25), (26), and (27), and the resulting distribution of the hyperparameters once convergence is achieved.

We start our analysis of the evidence parameters α_l by making some simplifications to make the derivations tractable:

- P channels are assumed.
- The same MF is used to process each of the P sensor output signals, i.e., $\boldsymbol{K}_p = \boldsymbol{K}$ and $\boldsymbol{\Sigma}_p = \boldsymbol{\Sigma} = \beta^{-1}\boldsymbol{\Lambda}$, $p = 1 \dots P$.
- The noise covariance matrix $\boldsymbol{\Sigma}$ is known, and $\boldsymbol{B} = \boldsymbol{\Sigma}^{-1}$.
- We assume the presence of a single multipath component, i.e., $L = 1$, with known delay τ. Thus, the design matrix is given as $\boldsymbol{K} = [\boldsymbol{r}(\tau)]$, where $\boldsymbol{r}(\tau) = [R_{uu}(-\tau), R_{uu}(T_s - \tau), \dots, R_{uu}((N-1)T_s - \tau)]^T$ is the associated basis function.
- The hyperparameter associated with this component is denoted as α.

Our goal is to consider the steady-state solution α_∞ for hyperparameter α in this simplified scenario. In this case (24) and (25) simplify to

$$\phi = (\alpha + \boldsymbol{r}(\tau)^H \boldsymbol{B} \boldsymbol{r}(\tau))^{-1},$$
$$\mu_p = \phi \boldsymbol{K}^H \boldsymbol{B} \boldsymbol{z}_p = \frac{\boldsymbol{r}(\tau)^H \boldsymbol{B} \boldsymbol{z}_p}{\alpha + \boldsymbol{r}(\tau)^H \boldsymbol{B} \boldsymbol{r}(\tau)}, \quad p = 1 \dots P.$$

Inserting these two expressions into (26) yields

$$\alpha^{-1} = \frac{1}{\alpha + \boldsymbol{r}(\tau)^H \boldsymbol{B} \boldsymbol{r}(\tau)} + \frac{\sum_p \left| \frac{\boldsymbol{r}(\tau)^H \boldsymbol{B} \boldsymbol{z}_p}{\alpha + \boldsymbol{r}(\tau)^H \boldsymbol{B} \boldsymbol{r}(\tau)} \right|^2}{P}. \quad (28)$$

From (28) the solution α_∞ is easily found to be

$$\alpha_\infty = \frac{(\boldsymbol{r}(\tau)^H \boldsymbol{B} \boldsymbol{r}(\tau))^2}{\frac{1}{P}\sum_p |\boldsymbol{r}(\tau)^H \boldsymbol{B} \boldsymbol{z}_p|^2 - \boldsymbol{r}(\tau)^H \boldsymbol{B} \boldsymbol{r}(\tau)}. \quad (29)$$

A closer look at (29) reveals that the right-hand side expression might not always be positive since the denominator can be negative for some values of $\boldsymbol{z}_p$. This contradicts the assumption that the hyperparameter α is positive[5]. A further analysis of (28) reveals, that (26) converges to (29) if, and only if, the denominator of (29) is positive:

$$\frac{1}{P}\sum_p |\boldsymbol{r}(\tau)^H \boldsymbol{B} \boldsymbol{z}_p|^2 > \boldsymbol{r}(\tau)^H \boldsymbol{B} \boldsymbol{r}(\tau). \quad (30)$$

Otherwise, the iterative learning scheme depicted in Fig. 4 diverges, i.e., $\alpha_\infty = \infty$. This can be inferred by interpreting (26) as a nonlinear dynamic system that, at iteration j, maps $\alpha^{[j-1]}$ into the updated value $\alpha^{[j]}$. The nonlinear mapping is given by the right-hand side of (26), where the quantities $\boldsymbol{\Phi}_p$ and $\boldsymbol{\mu}_p$ depend on the values of the hyperparameters at iteration $j-1$. In Fig. 6 we show several iterations of this mapping that illustrate how the solution trajectories evolve. If condition (30) is satisfied, the sequence of solutions $\{\alpha^{[j]}\}$ converges to a stationary point (Fig. 6(a)) given by (29). Otherwise, $\{\alpha^{[j]}\}$ diverges (Fig. 6(b)). Thus, (28) is a stationary point only provided the condition (30) is satisfied:

$$\alpha_\infty = \begin{cases} \frac{(\boldsymbol{r}(\tau)^H \boldsymbol{B} \boldsymbol{r}(\tau))^2}{\frac{\sum_p |\boldsymbol{r}(\tau)^H \boldsymbol{B} \boldsymbol{z}_p|^2}{P} - \boldsymbol{r}(\tau)^H \boldsymbol{B} \boldsymbol{r}(\tau)}; & \text{cond. (30) is satisfied} \\ \infty; & \text{otherwise.} \end{cases} \quad (31)$$

Practically, this means that for a given measurement $\boldsymbol{z}_p$, and known noise matrix $\boldsymbol{B}$, we can immediately decide whether a given basis function $\boldsymbol{r}(\tau)$ should be included in the basis by simply checking if (30) is satisfied or not.

A similar analysis is performed in [22], where the behavior of the likelihood function with respect to a single parameter is studied. The obtained convergence results coincide with ours when $P = 1$. Expression (30) is, however, more general and accounts for multiple channel observations and colored noise. In [22] the authors also suggest that testing (30) for a given basis function $\boldsymbol{r}(\tau)$ is sufficient to find a sparse representation and no further pruning is necessary. In other words, each basis function in the design matrix $\boldsymbol{K}$ is subject to the test (30) and, if the test fails, i.e., (30) does not hold for the basis function under test, the basis function is pruned.

In case of wireless channels, however, we have experimentally observed that even in simulated high-SNR scenarios such pruning results in a significantly overestimated number of multipath components. Moreover, it can be inferred from (30) that, as the SNR increases, the number of functions pruned with this approach decreases, resulting in less and less sparse representations. This motivates us to perform a more detailed analysis of (31).

Let us slightly modify the assumptions we made earlier. We now assume that the multipath delay τ is unknown. The design matrix is constructed similarly but this time $\boldsymbol{K} = [\boldsymbol{r}_l]$, where

$$\boldsymbol{r}_l = [R_{uu}(-T_l), \dots, R_{uu}((N-1)T_s - T_l)]^T$$

is the basis function associated with the delay $T_l \in \mathcal{T}$ used in our discrete-time model. Under these assumptions the input signal $\boldsymbol{z}_p$ is nothing else but the basis function $\boldsymbol{r}(\tau)$ scaled and embedded in the additive complex zero-mean Gaussian noise with covariance matrix $\boldsymbol{\Sigma}$, i.e.,

$$\boldsymbol{z}_p = w_p \boldsymbol{r}(\tau) + \boldsymbol{\xi}_p. \quad (32)$$

Let us further assume that $w_p \in \mathbb{C}$, $p = 1 \dots P$ are unknown but fixed complex scaling factors. In further derivations we assume, unless explicitly stated otherwise, that the condition (30) is satisfied for the basis $\boldsymbol{r}_l$. By plugging (32) into (29) and rearranging the result with respect to α_∞^{-1} we arrive at:

$$\begin{aligned} \alpha_\infty^{-1} = & \frac{|\boldsymbol{r}_l^H \boldsymbol{B} \boldsymbol{r}(\tau)|^2 \sum_p |w_p|^2}{P |\boldsymbol{r}_l^H \boldsymbol{B} \boldsymbol{r}_l|^2} + \\ & \frac{2 \sum_p \mathrm{Re}\{w_p \boldsymbol{r}_l^H \boldsymbol{B} \boldsymbol{r}(\tau) \boldsymbol{\xi}_p^H \boldsymbol{B} \boldsymbol{r}_l\}}{P |\boldsymbol{r}_l^H \boldsymbol{B} \boldsymbol{r}_l|^2} + \\ & \frac{\boldsymbol{r}_l^H \boldsymbol{B} \left(\sum_p \boldsymbol{\xi}_p \boldsymbol{\xi}_p^H \right) \boldsymbol{B} \boldsymbol{r}_l}{P |\boldsymbol{r}_l^H \boldsymbol{B} \boldsymbol{r}_l|^2} - \frac{1}{\boldsymbol{r}_l^H \boldsymbol{B} \boldsymbol{r}_l}. \end{aligned} \quad (33)$$

[5]Recall that α^{-1} is the prior variance of the corresponding parameter w. This constrains α to be nonnegative.

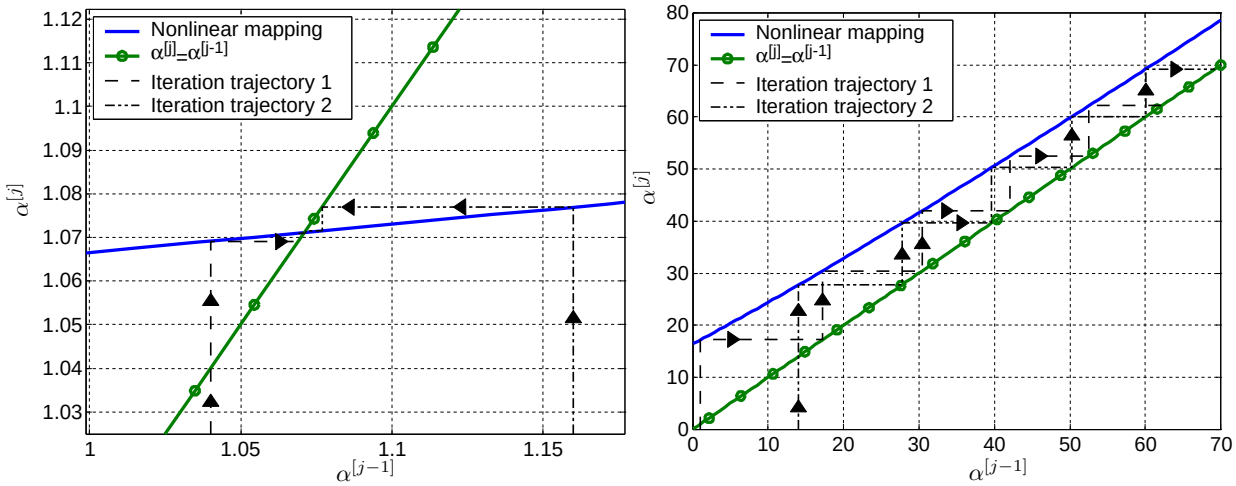


Fig. 6: Evolution of the two representative solution trajectories for two cases: (a) $\{\alpha^{[j]}\}$ converges, (b) $\{\alpha^{[j]}\}$ diverges.

Now, we consider two scenarios. In the first scenario $\tau = T_l \in \mathcal{T}$, i.e., the discrete-time model matches the observed signal. Although unrealistic, this allows to study the properties of α_∞^{-1} more closely. In the second scenario, we study what happens if the discrete-time model does not match perfectly the measured signal. This case helps us to define how the model selection rules have to be adjusted to consider possible misalignment of the path component delays in the model.

1) Model match: $\tau = T_l$: In this situation, $\boldsymbol{r}_l = \boldsymbol{r}(\tau)$, and thus (33) can be further simplified according to

$$\alpha_\infty^{-1} = \frac{\sum_p |w_p|^2}{P} + \frac{2\sum_p \mathrm{Re}\{w_p \boldsymbol{\xi}_p \boldsymbol{B}\boldsymbol{r}_l\}}{P(\boldsymbol{r}_l^H \boldsymbol{B}\boldsymbol{r}_l)} + \\ + \frac{\boldsymbol{r}_l^H \boldsymbol{B}\left(\sum_p \boldsymbol{\xi}_p \boldsymbol{\xi}_p^H\right)\boldsymbol{B}\boldsymbol{r}_l}{P(\boldsymbol{r}_l^H \boldsymbol{B}\boldsymbol{r}_l)^2} - \frac{1}{\boldsymbol{r}_l^H \boldsymbol{B}\boldsymbol{r}_l}, \tag{34}$$

where the only random quantity is the additive noise term $\boldsymbol{\xi}_p$. This allows us to study the statistical properties of the finite stationary point in (31).

Equation (34) shows how the noise and multipath component contribute to α_∞^{-1}. If all w_p are set to be zero, i.e., there is no multipath component, then $\alpha_\infty^{-1} = \alpha_\mathrm{n}^{-1}$ reflects only the noise contribution:

$$\alpha_\mathrm{n}^{-1} = \frac{\boldsymbol{r}_l^H \boldsymbol{B}\left(\sum_p \boldsymbol{\xi}_p \boldsymbol{\xi}_p^H\right)\boldsymbol{B}\boldsymbol{r}_l}{P(\boldsymbol{r}_l^H \boldsymbol{B}\boldsymbol{r}_l)^2} - \frac{1}{\boldsymbol{r}_l^H \boldsymbol{B}\boldsymbol{r}_l}. \tag{35}$$

On the other hand, in the absence of noise, i.e., in the infinite SNR case, the corresponding hyperparameter α_∞^{-1} includes the contribution of the multipath component[6] α_s^{-1}:

$$\alpha_\mathrm{s}^{-1} = \frac{\sum_p |w_p|^2}{P} + \frac{2\sum_p \mathrm{Re}\{w_p \boldsymbol{\xi}_p^H \boldsymbol{B}\boldsymbol{r}_l\}}{P(\boldsymbol{r}_l^H \boldsymbol{B}\boldsymbol{r}_l)}. \tag{36}$$

In a realistic case, both noise and multipath component are present, and α_∞^{-1} consists of the sum of two contributions $\alpha_\infty^{-1} = \alpha_\mathrm{s}^{-1} + \alpha_\mathrm{n}^{-1}$. Both quantities α_s^{-1} and α_n^{-1} are random variables with pdf's depending on the number of channel observations P, the basis function $\boldsymbol{r}_l$, and the noise covariance matrix $\boldsymbol{\Sigma}$. In the sequel we analyze their statistical properties.

We first consider α_s^{-1}. The first term on the right-hand side of (36) is a deterministic quantity that equals the average power of the multipath component. The second one, on the other hand, is random. The product $\mathrm{Re}\{w_p \boldsymbol{\xi}_p^H \boldsymbol{B}\boldsymbol{r}_l\}$ in (36) is recognized as the cross-correlation between the additive noise term and the basis function $\boldsymbol{r}_l$. It is Gaussian distributed with expectation and variance given as

$$E\left\{\frac{2\sum_p \mathrm{Re}\{w_p \boldsymbol{\xi}_p^H \boldsymbol{B}\boldsymbol{r}_l\}}{P(\boldsymbol{r}_l^H \boldsymbol{B}\boldsymbol{r}_l)}\right\} = 0, \quad \text{and} \\ E\left\{\left(\frac{2\sum_p \mathrm{Re}\{w_p \boldsymbol{\xi}_p^H \boldsymbol{B}\boldsymbol{r}_l\}}{P(\boldsymbol{r}_l^H \boldsymbol{B}\boldsymbol{r}_l)}\right)^2\right\} = \frac{2\sum_p |w_p|^2}{P^2(\boldsymbol{r}_l^H \boldsymbol{B}\boldsymbol{r}_l)}, \tag{37}$$

respectively, where $E\{\cdot\}$ denotes the expectation operator. Thus, α_s^{-1} is distributed as

$$\alpha_\mathrm{s}^{-1} \sim \mathcal{N}\Big(\frac{\sum_p |w_p|^2}{P}, \frac{2\sum_p |w_p|^2}{P^2(\boldsymbol{r}_l^H \boldsymbol{B}\boldsymbol{r}_l)}\Big), \tag{38}$$

which is a normal distribution with the mean given by the average power of the multipath component and variance proportional to this power.

Now, let us consider the term α_n^{-1}. In (35) the only random element is $\sum_{p=1}^P \boldsymbol{\xi}_p \boldsymbol{\xi}_p^H$. This random matrix is known to have a complex Wishart distribution [23], [24] with the scale matrix $\boldsymbol{\Sigma}$ and P degrees of freedom. Let us denote

$$\boldsymbol{c} = \frac{\boldsymbol{B}\boldsymbol{r}_l}{\sqrt{P}\boldsymbol{r}_l^H \boldsymbol{B}\boldsymbol{r}_l} \quad \text{and} \quad x = \boldsymbol{c}^H \sum_{p=1}^P \boldsymbol{\xi}_p \boldsymbol{\xi}_p^H \boldsymbol{c}. \tag{39}$$

It can be shown that x is Gamma-distributed, i.e., $x \sim \mathcal{G}(P, \sigma_c^2)$, with the shape parameter P and the scale parameter σ_c^2 given as

$$\sigma_c^2 = \boldsymbol{c}^H \boldsymbol{\Sigma} \boldsymbol{c} = \frac{1}{P(\boldsymbol{r}_l^H \boldsymbol{B}\boldsymbol{r}_l)}.$$

[6] Actually, the second term in the resulting expression vanishes in a perfectly noise-free case, and then $\alpha_\mathrm{s}^{-1} = \sum_p |w_p|^2 / P$.

The pdf of x reads

$$p(x|P,\sigma_c^2) = \frac{x^{P-1}}{\Gamma(P)(\sigma_c^2)^P}\mathrm{e}^{-x/\sigma_c^2}. \quad (40)$$

The mean and the variance of x are easily computed to be

$$\begin{aligned} \mathrm{E}\{x\} &= P\sigma_c^2 = \frac{1}{\boldsymbol{r}_l^H\boldsymbol{B}\boldsymbol{r}_l}, \\ \mathrm{Var}\{x\} &= P(\sigma_c^2)^2 = \frac{1}{P(\boldsymbol{r}_l^H\boldsymbol{B}\boldsymbol{r}_l)^2}. \end{aligned} \quad (41)$$

Taking the term $-1/(\boldsymbol{r}_l^H\boldsymbol{B}\boldsymbol{r}_l)$ in (35) into account, we introduce a variable $\tilde{\alpha}_{\mathrm{n}}^{-1}$: a zero mean random variable with the pdf

$$p_{\tilde{\alpha}_{\mathrm{n}}^{-1}}(x|P,\sigma_c^2) = \frac{(x-\mathrm{E}\{x\})^{P-1}}{\Gamma(P)(\sigma_c^2)^P}\mathrm{e}^{-(x-\mathrm{E}\{x\})/\sigma_c^2}, \quad (42)$$

which is equivalent to (40), but shifted so as to correspond to a zero-mean distribution. However, it is known that only positive values of α_{n}^{-1} occur in practice. The probability mass of the negative part of (42) equals the probability that the condition (30) is not satisfied and the resulting α_∞ eventually diverges to infinity and is pruned. Taking this into account the pdf of α_{n}^{-1} reads

$$p_{\alpha_{\mathrm{n}}^{-1}}(x) = P_{\mathrm{n}}\delta(x) + (1-P_{\mathrm{n}})\mathcal{I}^+(x)\tilde{p}_{\alpha_{\mathrm{n}}^{-1}}(x|P,\sigma_c^2), \quad (43)$$

where $\delta(\cdot)$ denotes a Dirac delta function, P_{n} is defined as

$$P_{\mathrm{n}} = \int_{-1/(\boldsymbol{r}_l^H\boldsymbol{B}\boldsymbol{r}_l)}^{0} \tilde{p}_{\alpha_{\mathrm{n}}^{-1}}(x|P,\sigma_c^2)dx,$$

and $\mathcal{I}^+(\cdot)$ is the indicator function of the set of positive real numbers:

$$\mathcal{I}^+(x) = \begin{cases} 0 & x \le 0 \\ 1 & x > 0. \end{cases}$$

A closer look at (43) shows that as P increases the variance of the Gamma distribution decreases, with α_{n}^{-1} concentrating at zero. In the limiting case as $P \to \infty$, (43) converges to a Dirac delta function localized at zero, i.e., $\alpha_{\mathrm{n}} = \infty$. This allows natural pruning of the corresponding basis function. This situation is equivalent to averaging out the noise, as the number of channel observations grows. Practically, however, P stays always finite, which means that (38) and (43) have a certain finite variance.

The pruning problem can now be approached from the perspective of classical detection theory. To prune a basis function, we have to decide if the corresponding value of α^{-1} has been generated by the noise distribution (43), i.e., the *null hypothesis*, or by the pdf of $\alpha_{\mathrm{s}}^{-1} + \alpha_{\mathrm{n}}^{-1}$, i.e., the *alternative hypothesis*. Computing the latter is difficult. The problem might be somewhat relaxed by taking the assumption that α_{s}^{-1} and α_{n}^{-1} are statistically independent. However proving the plausibility of this assumption is difficult. Even if we were successful in finding the analytical expression for the pdf of the alternative hypothesis, such model selection approach is hampered by our inability to evaluate (38) since the gains w_p's are not known *a priori*. However, we can still use (43) to select a threshold.

Recall that the presented algorithm allows to learn (estimate) the noise spectral height $N_0 = \beta^{-1}$ from the measurements. Assuming that we know β, and, as a consequence, the whole matrix $\boldsymbol{B}$ then, for any basis function $\boldsymbol{r}_l$ in the design matrix $\boldsymbol{K}$ and the corresponding hyperparameter α_l, we can decide with an *a priori* specified probability ρ that α_l is generated by the distribution (43). Indeed, let $\alpha_{\mathrm{th}}^{-1}$ be a ρ-quantile of (43) such that the probability $P(\alpha^{-1} \le \alpha_{\mathrm{th}}^{-1}) = \rho$. Since (43) is known exactly, we can easily compute $\alpha_{\mathrm{th}}^{-1}$ and prune all the basis functions for which $\alpha_l^{-1} \le \alpha_{\mathrm{th}}^{-1}$.

2) Model mismatch: $\tau \neq T_l$: The analysis performed above relies on the knowledge that the true multipath delay τ belongs to $\mathcal{T}$. Unfortunately, this is often unrealistic and the model mismatch $\tau \notin \mathcal{T}$ must be considered. To be able to study how the model mismatch influences the value of the hyperparameters we have to make a few more assumptions. Let us for simplicity select the model delay T_l to be a multiple of the chip period T_p. We will also need to assume a certain shape of the correlation function $R_{uu}(t)$ to make the whole analysis tractable. It may be convenient to assume that the main lobe of $R_{uu}(t)$ can be approximated by a raised cosine function with period $2T_p$. This approximation makes sense if the sounding pulse $p(t)$ defined in Sec. II is a square root raised cosine pulse. Clearly, this approximation can also be applied for other shapes of the main lobe, but the analysis of quality of such approximation remains outside the scope of this paper.

Just as in the previous case, we can split the expression (33) into the the multipath component contribution α_s^{-1}

$$\alpha_s^{-1} = \frac{|\gamma(\tau)|^2\sum_p |w_p|^2}{P} + \frac{2\sum_p \mathrm{Re}\{w_p\gamma(\tau)\boldsymbol{\xi}_p^H\boldsymbol{B}\boldsymbol{r}_l\}}{P|\boldsymbol{r}_l^H\boldsymbol{B}\boldsymbol{r}_l|}, \quad (44)$$

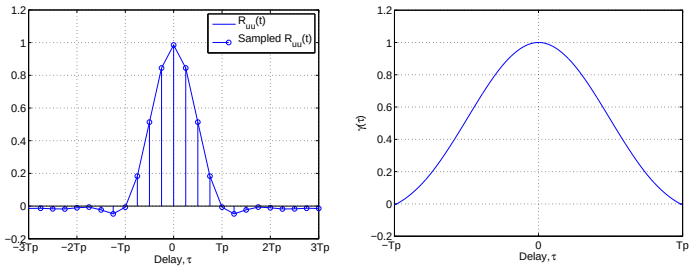
where

$$\gamma(\tau) = \frac{\boldsymbol{r}_l^H\boldsymbol{B}\boldsymbol{r}(\tau)}{\boldsymbol{r}_l^H\boldsymbol{B}\boldsymbol{r}_l}, \quad (45)$$

and the same noise contribution α_n^{-1} defined in (35). It can be seen that the $\gamma(\tau)$ makes (44) differ from (36), and as such it is the key to the analysis of the model mismatch. Note that this function is bounded as $|\gamma(\tau)| \le 1$, with equality following only if $\tau = T_l$. Note also that in our case for $|\tau - T_l| < T_p$ the correlation $\gamma(\tau)$ is strictly positive.

Due to the properties of the sounding sequence $u(t)$, the magnitude of $R_{uu}(t)$ for $|t| > T_p$ is sufficiently small and in our analysis of model mismatch can be safely assumed to be zero. Furthermore, if $\boldsymbol{r}_l$ is chosen to coincide with the multiple of the sampling period $T_l = lT_s$, then it follows from (10) that the product $\boldsymbol{r}_l^H\boldsymbol{B} = \boldsymbol{r}_l^H\boldsymbol{\Sigma}^{-1} = \beta\boldsymbol{e}_l^H$ is a vector with all elements being zero except the lth element, which is equal to β. Thus, the product $\boldsymbol{r}_l^H\boldsymbol{B}\boldsymbol{r}(\tau)$ for $|\tau - T_l| < T_p$ must have a form identical to that of the correlation function $R_{uu}(t)$ for $|t| < T_p$. It follows that when $|\tau - T_l| \ge T_p$ the correlation $\gamma(\tau)$ can be assumed to be zero, and it makes sense to analyze (44) only when $|\tau - T_l| < T_p$. In Fig. 7 we plot the correlation functions $R_{uu}(t)$ and $\gamma(\tau)$ for this case.

Since the true value of τ is unknown, we assume this parameter to be random, uniformly distributed in the interval $[T_l - T_p, T_l + T_p]$. This in turn induces a corresponding



(a) (b)

Fig. 7: Evaluated correlation functions a) $R_{uu}(t)$ and b) $\gamma(\tau)$.

distributions for the random variables $\gamma(\tau)$ and $\gamma(\tau)^2$, which enter, respectively, the second and first terms on the right-hand side of (44).

It can be shown that in this case $\gamma(\tau) \sim \mathcal{B}(0.5, 0.5)$, where $\mathcal{B}(0.5, 0.5)$ is a Beta distribution [25] with both distribution parameters equal to $1/2$. The corresponding pdf $p_\gamma(x)$ is given in this case as

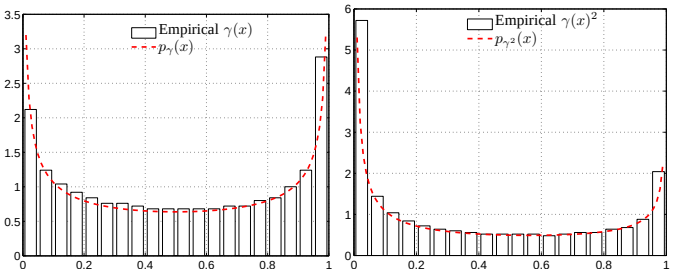
$$p_\gamma(x) = \frac{1}{B(0.5, 0.5)} x^{-\frac{1}{2}} (1-x)^{-\frac{1}{2}}, \tag{46}$$

where $B(\cdot, \cdot)$ is a Beta-function [26] with $B(0.5, 0.5) = \pi$.

It is also straight-forward to compute the pdf of the term $\gamma(\tau)^2$:

$$p_{\gamma^2}(x) = \frac{1}{\pi} x^{-\frac{3}{4}} (1-\sqrt{x})^{-\frac{1}{2}}. \tag{47}$$

The corresponding empirical and theoretical pdf's of $\gamma(\tau)$ and $\gamma(\tau)^2$ are shown in Fig. 8.



(a) (b)

Fig. 8: Comparison between the empirical and theoretical pdf's of a) $\gamma(\tau)$ and b) $\gamma(\tau)^2$ for the cosine approximation case. To compute the histogram $N = 5000$ samples were used.

Now we have to find out how this information can be utilized to design an appropriate threshold. In the case of a perfectly matched model the threshold is selected based on the noise distribution (43). In the case of a model mismatch, the term (44) measures the amount of the interference resulting from the model imperfection.

Indeed, if $|\tau - T_l| \geq T_p$, then the resulting $\gamma(\tau) = 0$, and thus $\alpha_s^{-1} = 0$. The corresponding evidence parameter α_∞^{-1} is then equal to the noise contribution α_n^{-1} only and will be pruned using the method we described for the matched model case. If however, $|\tau - T_l| < T_p$, then a certain fraction of α_s^{-1} will be added to the noise contribution α_n^{-1}, thus causing the interference. In order to be able to take this interference into account and adjust the threshold accordingly, we propose the following approach.

The amount of interference added is measured by the magnitude of α_s^{-1} in (44). It consists of two terms: the first one is the multipath power, scaled by the factor $\gamma(\tau)^2$:

$$\gamma(\tau)^2 \frac{\sum_p |w_p|^2}{P}. \tag{48}$$

The second term is a cross product between the multipath component and the additive noise, scaled by $\gamma(\tau)$:

$$\gamma(\tau) \frac{2 \sum_p \mathrm{Re}\{w_p \boldsymbol{\xi}_p^H \boldsymbol{B} \boldsymbol{r}_l\}}{P(\boldsymbol{r}_l^H \boldsymbol{B} \boldsymbol{r}_l)}. \tag{49}$$

Both terms have the same physical interpretation as in (36), but with scaling factors $\gamma(\tau)$ depending on the true value of τ.

We see that in (44) there are quite a few unknowns: we do not know the true multipath delay τ, the multipath gains w_p, as well as the instantaneous noise value $\boldsymbol{\xi}$. To be able to circumvent this uncertainties, we consider the large sample size case, i.e, $P \to \infty$ and invoke the law of large numbers to approximate (48) and (49) by their expectations.

First of all, using (37) it is easy to see that

$$E\left\{ \gamma(\tau) \frac{2 \sum_p \mathrm{Re}\{w_p \boldsymbol{\xi}_p^H \boldsymbol{B} \boldsymbol{r}_l\}}{P(\boldsymbol{r}_l^H \boldsymbol{B} \boldsymbol{r}_l)} \right\} = 0.$$

The other term (48) converges to $\gamma(\tau)^2 E\{|w_p|^2\}$ as P grows. So, even in the high SNR regime and infinite number of channel observations P the term (48) does not go to zero. In order to assess how large it is, we approximate the gains of the multipath component w_p by the corresponding MAP estimate μ_p obtained with (25).

The correlation function $\gamma(\tau)$ can also be taken into account. Since we know the distributions of both $\gamma(\tau)$ and $\gamma(\tau)^2$, we can summarize these by the corresponding mean values. In fact, we will need the mean only for $\gamma(\tau)^2$ since it enters the irreducible part of α_s^{-1}.

In our case it is computed as:

$$E\{\gamma(\tau)^2\} = \int_0^1 \frac{x}{\pi} x^{-\frac{3}{4}} (1-\sqrt{x})^{-\frac{1}{2}} dx = \frac{3}{8} \tag{50}$$

Having obtained the mean, we can approximate the interference $\hat{\alpha}_s^{-1}$ due to the model mismatch as

$$\hat{\alpha}_s^{-1} = 3/8 \times \frac{\sum_{p=0}^{P-1} |\mu_p|^2}{P}, \tag{51}$$

The final threshold that accounts for the model mismatch is then obtained as

$$\hat{\alpha}_{\mathrm{th}}^{-1} = \hat{\alpha}_s^{-1} + \alpha_{\mathrm{th}}^{-1}, \tag{52}$$

where $\alpha_{\mathrm{th}}^{-1}$ is the threshold developed earlier for the matched model case.

B. Improving the learning algorithm to cope with the model selection

In the light of the model selection strategy considered here we anticipate two major problems arising with the learning

algorithm discussed in Section III. The first one is the estimation of the channel parameters that requires computation of the posterior (24). Even for the modest sizes of the hypothesis $\mathcal{H}_i$ (from 100 to 200 basis functions), the matrix inversion is computationally very intensive. This issue becomes even more critical if we consider a hardware implementation of the estimation algorithm. The second problem arises due to the non-vanishing correlation between the basis vectors $\boldsymbol{r}_l$ constituting the design matrix $\boldsymbol{K}$. A very undesirable consequence of this correlation is that the evidence parameters α_l associated with these vectors become also correlated, and thus no longer represent the contribution of a single basis function. As a consequence the developed model selection rules are no longer applicable.

It is, however, possible to circumvent these two difficulties by modifying the learning algorithm as discussed below. The basic idea consists of estimating the channel parameters for each basis independently. In other words, instead of solving (24), (25), (26), and (27) jointly for all L basis functions, we find a solution for each basis vector separately. First, the new data vector $\boldsymbol{x}_{p,l}$ for the lth basis is computed as

$$\boldsymbol{x}_{p,l} = \boldsymbol{z}_p - \sum_{k=1,k\neq l}^{L} \boldsymbol{r}_k \mu_{p,l}. \tag{53}$$

This new data vector $\boldsymbol{x}_{p,l}$ now contains the information relevant to the basis $\boldsymbol{r}_l$ only. It is then used to update the corresponding posterior statistics as well as evidence parameters exclusively for the lth basis as follows:

$$\Phi_l = (\alpha_l + \beta \boldsymbol{r}_l^H \boldsymbol{\Lambda}^{-1} \boldsymbol{r}_l)^{-1}, \tag{54}$$

$$\mu_{p,l} = \beta \Phi_l \boldsymbol{r}_l^H \boldsymbol{\Lambda}^{-1} \boldsymbol{x}_{p,l}, \quad p = 1 \dots P. \tag{55}$$

Note that expressions (54) and (55) are now scalars, unlike their matrix counterparts (24) and (25). Similarly, we update the evidence parameters as

$$\alpha_l = \frac{P}{\sum_{p=1}^{P} \left(\Phi_l + |\mu_{p,l}|^2 \right)}. \tag{56}$$

Updates (54), (55), and (56) are performed for all L components sequentially. Once all components are updated, we update the noise hyperparameter N_0:

$$N_0 = (\beta^{-1}) = \frac{1}{NP} \left(\sum_{p=1}^{P} \mathrm{tr}[\boldsymbol{\Phi}(\boldsymbol{K})^H \boldsymbol{\Lambda}^{-1} \boldsymbol{K}] + \right. \\ \left. + \sum_{p=1}^{P} (\boldsymbol{z}_p - \boldsymbol{K}\boldsymbol{\mu}_p)^H \boldsymbol{\Lambda}^{-1} (\boldsymbol{z}_p - \boldsymbol{K}\boldsymbol{\mu}_p) \right). \tag{57}$$

The above updating procedures constitute a single iteration of the modified learning algorithm. This iteration is repeated until some suitable convergence criterion is satisfied. Note that the procedure described here is an instance of the SAGE algorithm. This opens a potential to unite both SAGE and Evidence Procedure, allowing to implement simultaneous parameter and model order estimation within the SAGE framework.

This iterative method, also known as successive interference cancellation, allows solving both anticipated problems. First of all, there is no need to compute matrix inversion at each iteration. Second, the obtained values of α now reflect the contribution of a single basis function only, since they were estimated while the contribution of other bases was canceled in (53).

Now, at the end of each iteration, once the new value of the noise is obtained using (57), we can decide to prune some of the components, as described in Section IV-A.

C. MDL principle and Evidence Procedure

The goal of this section is to establish a relationship between the classical information-theoretic criteria for model selection, such as Minimum Description Length (MDL) [4], [5], [8], [20], and the Evidence Procedure discussed here. For simplicity we will only consider a single channel observation case, i.e., $P = 1$. Extension to the case $P > 1$ is straightforward.

The MDL criterion was originally formulated from the perspective of coding theory as a solution to the problem of balancing the code length and the resulting length of the data encode with this code. This concept however can naturally be transferred to general model selection problems.

In terms of parameter estimation theory, we can interpret the length of the encoded data as the parameter likelihood evaluated at its maximum. The length of the code is equivalent to what is known in the literature as the *stochastic complexity* [11], [20], [21]. The Bayesian interpretation of the stochastic complexity term obtained for likelihood functions from an exponential family (see [20] for more details) is of particular interest for our problem at hand. The Description Length in this case is given as

$$\mathrm{DL}(\mathcal{H}_i) = \underbrace{-\log(p(\boldsymbol{z}|\boldsymbol{w}_{MAP}, \mathcal{H}_i)}_{\text{model performance}} + \\ \underbrace{\frac{L}{2}\log\frac{N}{2\pi} - \log(p(\boldsymbol{w}_{MAP}|\mathcal{H}_i)) + \log(\sqrt{|\boldsymbol{I}_1(\boldsymbol{w}_{MAP})|})}_{\text{stochastic complexity}}. \tag{58}$$

Here $\boldsymbol{I}_1(\boldsymbol{w}_{MAP})$ is the Fisher information matrix of a single sample evaluated at the MAP estimate of the model parameter vector, and $p(\boldsymbol{w}_{MAP}|\mathcal{H}_i)$ is the corresponding prior for this vector.

Thus, joint model and parameter estimation schemes should aim at minimizing the DL so as to find the compromise between the model fit (likelihood) and the number of the parameters involved. The latter is directly proportional to the stochastic complexity term.

We will now show, that the EP employed in our model selection scheme results in a very similar expression.

Let us once again come back to the evidence term (19). To exemplify the main message that we want to convey here, we will compute the integral in (19) differently. For each model hypothesis defined as in Section III, let us define $\Delta(\boldsymbol{w}_i) = -\log(p(\boldsymbol{z}|\boldsymbol{w}_i, \beta_i)) - \log(p(\boldsymbol{w}_i|\boldsymbol{\alpha}_i))$. Then equation (19) can be expressed as

$$p(\boldsymbol{z}|\boldsymbol{\alpha}_i, \beta_i) = \int \exp(-\Delta(\boldsymbol{w}_i)) d\boldsymbol{w}_i. \tag{59}$$

Now we proceed by computing the integral (59) using a Laplace method [8, ch. 27], also known as a saddle-point approximation. The essence of the method consists of computing the second order Taylor series around the argument that maximizes the integrand in (59), which is the MAP estimate of the model parameters $\boldsymbol{\mu}_i$ given in (18). In our case $\Delta(\boldsymbol{w}_i)$ is known to be quadratic, since both $p(\boldsymbol{z}|\boldsymbol{w}_i, \beta_i)$ and $p(\boldsymbol{w}_i|\boldsymbol{\alpha}_i)$ are Gaussian, so the approximation is exact.

It is then easily verified that for the hypothesis $\mathcal{H}_i$ with $|\mathcal{P}(i)| = L$ basis functions

$$p(\boldsymbol{z}|\boldsymbol{\alpha}_i,\beta_i) = \int \exp(-(\boldsymbol{w}_i - \boldsymbol{\mu}_i)^H \boldsymbol{\Phi}_i^{-1}(\boldsymbol{w}_i - \boldsymbol{\mu}_i))d\boldsymbol{w}_i \times \exp(-\Delta(\boldsymbol{\mu}_i)) = \exp(-\Delta(\boldsymbol{\mu}_i))\pi^L|\boldsymbol{\Phi}_i|, \tag{60}$$

By taking the logarithm of (60) and changing the sign of the resulting expression we arrive at the final expression for the negative log-evidence

$$-\log(p(\boldsymbol{z}|\boldsymbol{\alpha}_i, \beta_i)) = -\log(p(\boldsymbol{z}|\boldsymbol{\mu}_i, \beta_i)) - \log(p(\boldsymbol{\mu}_i|\boldsymbol{\alpha}_i)) - L\log(\pi) - \log(|\boldsymbol{\Phi}_i|). \tag{61}$$

Noting that $\boldsymbol{\Phi}_i$ has been computed using N data samples, and that in this case $\log(|\boldsymbol{\Phi}_i/N|) = \log(|\boldsymbol{I}_1^{-1}(\boldsymbol{\mu}_i)|)$, we rewrite (61) as

$$\mathrm{DL}(\mathcal{H}_i) = \underbrace{-\log(p(\boldsymbol{z}|\boldsymbol{\mu}_i, \beta_i))}_{\text{model performance}} + \underbrace{L\log(\frac{N}{\pi}) - \log(p(\boldsymbol{\mu}_i|\boldsymbol{\alpha}_i)) + \log(|\boldsymbol{I}_1(\boldsymbol{\mu}_i)|)}_{\text{model complexity}}, \tag{62}$$

We note that (58) and (62) are essentially similar, with the distinction that the latter accounts for complex data. Thus we conclude that maximizing evidence (or minimizing the negative log-evidence) is equivalent to minimizing the DL.

Let us now consider how this can be exploited in our case. In general, the MDL concept assumes presence of multiple *estimated* models. The model that minimizes the DL functional is then picked as the optimal one. In our case, evaluation of the DL functional for all possible hypotheses $\mathcal{H}_i$ is way too complex. In order to make this procedure more efficient, we can exploit the estimated evidence information.

Consider the graph shown in Fig. 9. Each node on the graph

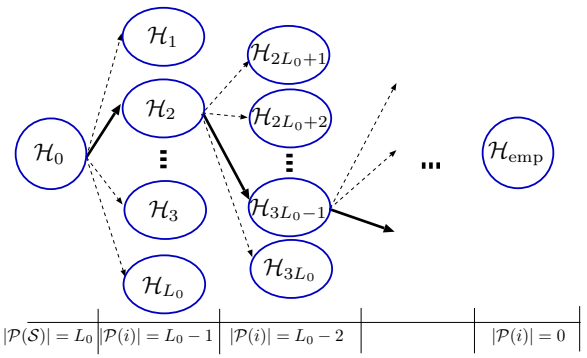


Fig. 9: Model selection by evidence evaluation.

corresponds to a certain hypothesis $\mathcal{H}_i$ consisting of $|\mathcal{P}_i|$ basis functions. An edge emanating from a node is associated with a certain basis function from the hypothesis $\mathcal{H}_i$. Should the path through the graph include this edge, the corresponding basis function would be pruned, leading to a new smaller hypothesis. Clearly, the optimal path through the graph should be the one that minimizes the DL criterion. Now, let us propose a strategy to find the optimal model without evaluating all possible paths through the graph.

At the initial stage, we start in the leftmost node, which corresponds to the full hypothesis $\mathcal{H}_0$. We then proceed with the learning using the iterative scheme depicted in Fig. 4 to obtain the estimates of the evidence parameters α_l, $l \in \mathcal{P}(0)$, for each basis function in $\mathcal{H}_0$. Once convergence is achieved, we evaluate the correspondind description length DL_0 for this hypothesis using (62). Since the optimal path should decrease the DL, the hypothesis at the next stage $\mathcal{H}_i$ is selected by moving along the edge that corresponds to the basis function with the largest value of α (i.e., the basis function with the smallest evidence). For the newly selected hypothesis $\mathcal{H}_i$ we again estimate the evidence parameters $\boldsymbol{\alpha}_i$ and the corresponding description length DL_i. If $\mathrm{DL}_0 < \mathrm{DL}_i$, then the hypothesis $\mathcal{H}_0$ achieves the minimum of the description length and it is then selected it as a solution. Otherwise, i.e., if $\mathrm{DL}_0 > \mathrm{DL}_i$, we continue along the graph, each time pruning a basis function with the smallest evidence and comparing the description length at each stage. We proceed so until the DL does not decrease any more, or until we stop at the last node that has no basis functions at all. Such an empty hypothesis corresponds to the case when there is no structure in the observed data. In other words it corresponds to the case when the algorithm failed to find any multipath components. This technique requires searching between L_0 to a maximum of $L_0(L_0+1)/2$ possible hypotheses, while a total search requires testing a total of 2^{L_0} different models.

V. Application of the RVM to wireless channels

The application of the proposed channel estimation scheme coupled with the considered model selection approach requires two major components: 1) it needs a proper construction of the kernel design matrix that is dense enough to ensure good delay resolution, and 2) the iterative nature of the algorithm requires a good initialization.

The construction of the design matrix $\boldsymbol{K}$ can be done with various approaches, depending on how much *a priori* information we have about the possible positions of the multipath components. The columns of the matrix $\boldsymbol{K}$ contain the shifted versions of the kernel $R_{uu}(nT_s - T_l)$, $l = 1 \ldots L_0$, where T_l are the possible positions of the multipath components that form the search space $\mathcal{T}$. The delays T_l can be selected uniformly to cover the whole delay span or might be chosen so as to sample some areas of the impulse response more densely, where multipath components are likely to appear. Note that the delays T_l are not constrained to fall on a regular grid. The power-delay profile (PDP) may be a good indicator of how to place the multipath components. Initialization of the model hyperparameters can also be done quite effectively. In the sequel we propose two different initialization techniques.

The simplest one consists of evaluating the condition (30) for all the basis functions in the already created design matrix $\boldsymbol{K}$. For those basis functions that satisfy condition (30), the corresponding evidence parameter is initialized using (29). Other basis functions are removed from the design matrix $\boldsymbol{K}$. Such initialization assumes that there is no interference between the neighboring basis functions. It makes sense to employ it when the minimal spacing between the elements in $\mathcal{T}$ is at most half the duration of the sounding pulse T_p.

In the case when the spacing is denser, it is better to use independent evidence initialization. This type of initialization is in fact coupled with the construction of the design matrix $\boldsymbol{K}$ and relies on the successive interference cancellation scheme discussed in the Section IV-B. To make the procedure work, we need to set the initial channel coefficients to zero, i.e., $\boldsymbol{\mu}_p \equiv 0$. The basis vectors $\boldsymbol{r}_l$ are computed as usual according to the delay search space $\mathcal{T}$. The initialization iterations start by computing (53). The basis $\boldsymbol{r}_l$ that is best aligned with the residual $\boldsymbol{x}_{p,l}$ is then selected. If the selected $\boldsymbol{r}_l$ satisfies condition (30), it is included in the design matrix $\boldsymbol{K}$, and the corresponding parameters Φ_l, $\mu_{p,l}$, and α_l are computed according to (54), (55), and (56), respectively. These steps are continued until all bases with delays from the search space $\mathcal{T}$ are initialized, or until the basis vector that does not satisfy the condition (30) is encountered.

Of course, in order to be able to use this initialization scheme, it is crucial to get a good initial noise estimate. The initial noise parameter $N_0^{[0]}$ can in most cases be estimated from the tails of the channel impulse response, where multipath components are unlikely to be present or too weak to be detected. Generally, we have observed that the algorithm is less sensitive to the initial values of the hyperparameters $\boldsymbol{\alpha}$, but proper initialization of the noise spectral height is crucial.

Now we can describe the simulation setup used to assess the performance of the proposed algorithm.

A. Simulation setup

The generation of the synthetic channel is done following the block-diagram shown in Fig. 1: a single period $u(t)$ of the sounding sequence $s(t)$ is filtered by the channel with the impulse response $\boldsymbol{h}(t)$, and complex white Gaussian noise is added to the channel outputs to produce the received signal $\boldsymbol{y}(t)$. The received signal is then run through the MF. The continuous-time signals at the output of the MF are represented with cubic splines. The resulting spline representation is then used to obtain the sampled output $z_p[n]$, $p = 1 \dots P$, with $n = 0 \dots N-1$. Output signals $z_p[n]$ are then used as the input to the estimation algorithm.

For all P channel observations we use the same MF, and thus $\boldsymbol{\Phi} = \boldsymbol{\Phi}_p$, $\boldsymbol{K} = \boldsymbol{K}_p$, and $\boldsymbol{\Sigma} = \boldsymbol{\Sigma}_p$, $p = 1 \dots P$. Without loss of generality, we assume a shaping pulse of the duration $T_p = 10$nsec. The sampling period is assumed to be $T_s = T_p/N_s$, where N_s is the number of samples per chip used in the simulations. The sounding waveform $u(t)$ consists of $M = 255$ chips. We also assume the maximum delay spread in all simulations to be $\tau_{spread} = 1.27\mu$sec. With these parameters, a one-sample/chip resolution results in $N = 128$ samples. The autocorrelation function $R_{uu}(t)$ is also represented with cubic splines, allowing a proper construction of the design matrix $\boldsymbol{K}$ according to the predefined delays in $\mathcal{T}$. Realizations of the channel parameters $w_{l,p}$ are randomly generated according to (12).

The performance of the algorithm is also evaluated under different SNR's at the output of the MF, defined as

$$\text{SNR} = 10 \log_{10} \Big(\frac{1/\alpha}{N_0} \Big). \tag{63}$$

For simplicity, we assumed that in the case $L > 1$ all simulated multipath components have the same expected power α^{-1}. Although this is not always a realistic assumption, it ensures that all simulated multipath components present in the measurement will be "treated" equally.

B. Numerical simulations

Let us now demonstrate the performance of the model selection schemes discussed in Section IV on synthetic, as well as on measured channels.

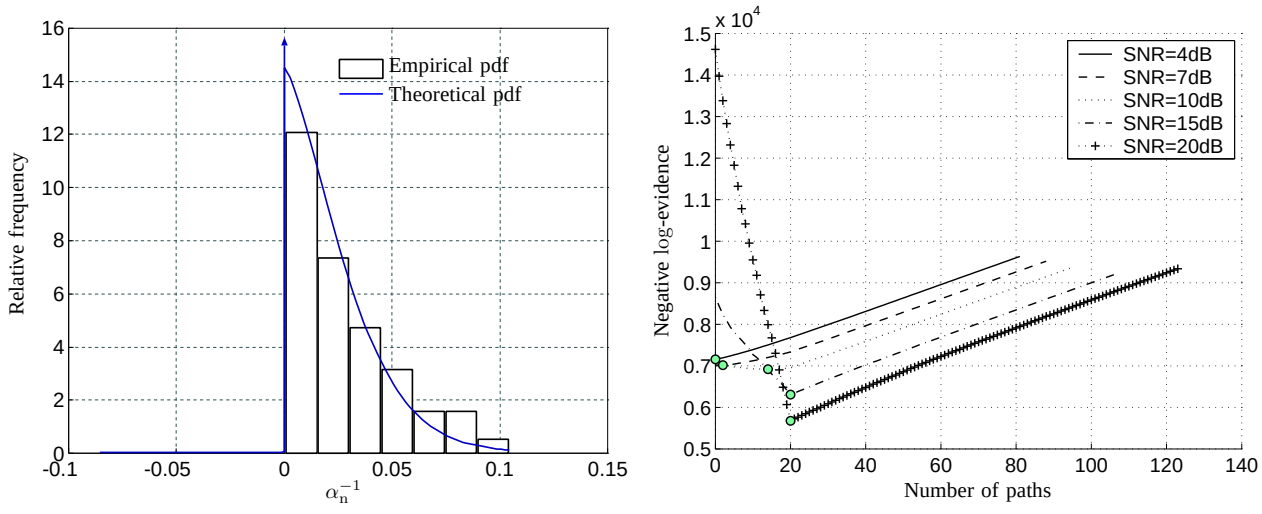
1) Multipath detection with the perfect model match: First we consider the distribution of the hyperparameters once the stationary point has been reached. In order to do that, we apply the learning algorithm to the full hypothesis $\mathcal{H}_0$. The delays in $\mathcal{H}_0$ are evenly positioned over the length of the impulse response: $\mathcal{T} = \{lT_s; l = 0 \dots N-1\}$, i.e., $L_0 = N$. Here, we simulate the channel with a single multipath component, i.e., $L = 1$, having the delay τ' equal to a multiple of the sampling period T_s. Thus, in the design matrix $\boldsymbol{K}$ corresponding to the full hypothesis $\mathcal{H}_0$ there will be a basis function that coincides with the contribution of the true multipath component. Once the parameters have been learned, we partition all the hyperparameters α into those attributed to the noise, i.e., α_{n}, and one parameter that corresponds to the multipath component α_{s}, i.e., the one associated with the delay $T_l = \tau'$.

In a next step, we compare the obtained histogram of α_{n}^{-1} with the theoretical pdf $p_{\alpha_{\text{n}}^{-1}}(x)$ given in (43). The corresponding results are shown in Fig. 10(a). A very good match between the empirical and theoretical pdf's can be observed.

Similarly, we investigate the behavior of the negative $\log$-evidence versus the size of the hypothesis. We consider a similar simulation setup as above, however with more than just one multipath component to make the results more realistic. Figure 10(b) depicts the evaluated negative $\log$-evidence (61) as a function of the model order, evaluated for a single realization, when the true number of components is $L = 20$, and the number of channel observations is $P = 5$.

Note that, as the SNR increases, there are fewer components subject to the initial pruning, i.e., those that do not satisfy condition (30). We also observe that the minimum of the negative $\log$-evidence (i.e., maximum of the evidence) becomes more pronounced as the SNR increases, which has an effect of decreasing the variance of the model order estimates.

In order to find the best possible performance of the algorithm, we first perform some simulations assuming that



(a) (b)

Fig. 10: Evidence-based model selection criteria. a) Empirical (bar plot) and theoretical (solid line) pdf's of hyperparameters $\alpha_{\rm n}^{-1}$ (SNR $=$ 10dB, and $P = 10$). To compute the histogram $N = 500$ samples was used.; b) Negative $\log$-evidence as a function of the model order (number of paths) for different SNR values ($P = 5$, and $L = 20$).

the discrete-time model (8) perfectly matches the continuous-time model (6), i.e., $\tau_l \in \mathcal{T},\ l = 1, \ldots, L$. This is realized by drawing uniformly L out of N possible delay values in the interval $[0, T_s(N-1)]$. Again, $\mathcal{T} = \{lT_s; l = 0 \ldots N-1\}$. The number of multipath components in the simulated channels is set to $L = 5$ and the channel is sampled with $N_s = 2$ samples per chip.

In this simulation we evaluate the detection performance by counting the errors made by the algorithms. Two types of errors can occur: (a) an *insertion error*– an erroneous detection of a non-existing component; (b) a *deletion error*– a loss of an existing component. The case when an estimated delay $\hat{T}_l$ matches one of the true simulated delays is called a *hit*. We further define the *multipath detection rate* as the ratio between the number of hits to the true number of components L plus the number of insertion errors. It follows that the detection rate is equal to 1 only if the number of hits equals the true number of components. If, however, the algorithm makes any deletion or insertion errors, the detection rate is then strongly smaller than 1. We study the detection rates for both model selection schemes versus different SNR's. The presented results are averaged over 300 independent channel realizations.

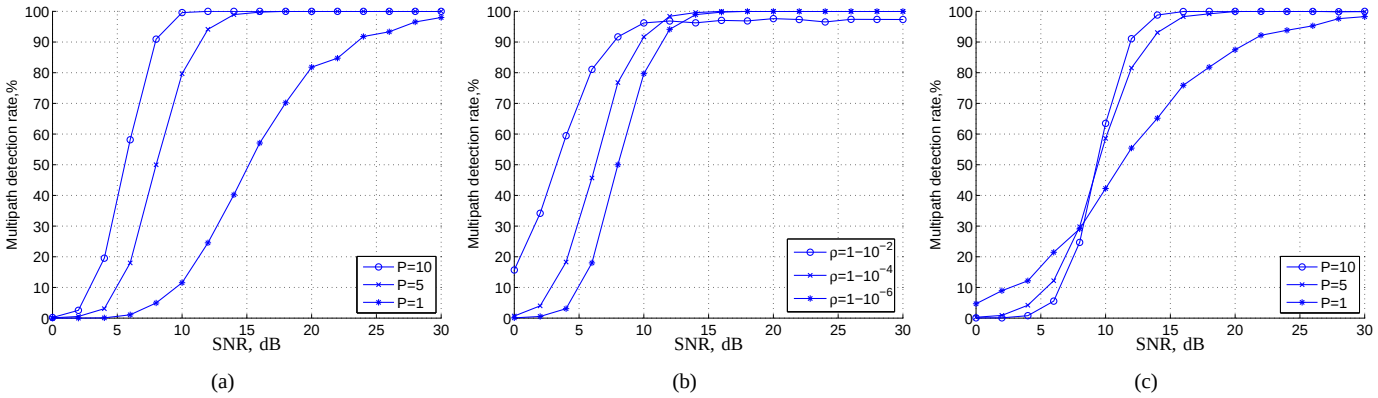
We start with the model selection approach based on the threshold selection using the ρ-quantile of the noise distribution - quantile-based model selection. The results shown in Fig. 11(a) are obtained for $\rho = 1-10^{-6}$ and different numbers of channel observations P. It can be seen that, as P increases, the detection rate significantly improves. To obtain the results shown in Fig. 11(b) we fix the number of channel observations at $P = 5$ and vary the value of the quantile ρ. It can be seen that as ρ approaches unity, the threshold is placed higher, meaning that fewer noise components can be mistakenly detected as multipath components, thus slightly improving the detection rate. However higher thresholds require a higher SNR to achieve the same detection rate, as compared for the thresholds obtained with lower ρ.

The next plot in Fig.11(c) shows the multipath detection rate when the model is selected based on the evaluation of the negative $\log$-evidence under different model hypotheses (negative $\log$-evidence model selection). It is interesting to note that in this case the reported curves behave quite differently from those shown in Fig. 11(a). First, we see that for the case $P = 1$ the behavior of this method is slightly better, compared to the threshold-based method in Fig. 11(a). But as P grows, the performance of the multipath detection does not increase proportionally, but rather exhibits a threshold-like behavior. In other words, multipath detection based on the negative $\log$-evidence and alike MDL-based model selection requires the SNR above a certain threshold in order to operate reliably. Furthermore, this threshold is independent of the number of channel observations P.

Thus from Fig. 11(a) and Fig. 11(c) we can conclude that the quantile-based method performs better in a sense that it can always be improved by increasing the number of channel observations. Further, model selection using the thresholding approach can be performed on-line, concurrent with parameters estimation, while in the other case multiple models have to be learned.

Now, let us consider how the EP performs when the multipath component delays are on the real line, rather than on a discrete grid. Clearly, this case corresponds more to the real-life situation.

2) Multipath detection with the model mismatch: In the real world the delays of the multipath components do not necessarily coincide with the elements in $\mathcal{T}$ used to approximate the continuous-time model (6). By using the discrete-time models to approximate the continuous-time counterparts, we would necessarily expect some performance degradation in terms of an increased number of components. This problem is similar to the problem that occurs in fractional delay filters (FDF) [27].



(a) (b) (c)

Fig. 11: Multipath detection rates based on the EP. (a) Quantile-based model selection versus P: $\rho = 1 - 10^{-6}$, $L = 5$; (b) Quantile-based model selection versus ρ: $P = 5$, $L = 5$; (c) Negative $\log$-evidence-based detection versus P.

An FDF aims at approximating a delay that is not a multiple of the sampling period. As shown in [27], such filters have infinite impulse response. Though FIR approximations exist, they require several samples to represent a single delay.

Since there is an inevitable mismatch between the continuous-time and discrete-time models, it is worth asking how densely we should quantize the delay line to form the design matrix in order to achieve the best performance. It is convenient to select the delays in $\mathcal{T}$ of the discrete-time model as a multiple of the sampling period T_s. As the sampling rate increases the true delay values get closer to some elements in $\mathcal{T}$, thus approaching the continuous-time model (6).

We simulate a channel with a single multipath component that has a random delay, uniformly distributed in the interval $[0, \tau_{spread}]$.

The criterion used here to assess the performance of the algorithm is the probability of correct path extraction. This probability is defined to be the conditional probability that, given any path is detected, the algorithm finds exactly one component with the absolute difference between the estimated and the true delay less than the chip pulse duration T_p. Notice that the probability of correct path extraction is conditioned on the path detection, i.e., it is evaluated for the cases when the estimation algorithm is able to find at least one component.

It is also interesting to compare the performance of the EP with other parameter estimation techniques. Here we consider the SAGE algorithm [2] that has become a popular multipath parameter estimation technique. The SAGE algorithm, however, does not provide any information about the number of multipath components. To make the comparison fair, we augment it with the standard MDL criterion [4], [5] to perform model selection.

Thus, we are going to compare three different model selection algorithms: the quantile-based (or threshold-based) scheme with a pre-selected quantile $\rho = 1 - 10^{-6}$, the SAGE+MDL method, and negative $\log$-evidence method. We are also going to use the threshold-based method to demonstrate the difference between two EP initialization schemes: the joint initialization, and the independent initialization, discussed in Section V. In all simulations the negative $\log$-evidence method was initialized using independent initialization.

We start with channels sampled with $N_s = 1$ sample/chip resolution and $P = 5$ channel observations. We see that the shown methods have different probabilities of path detection (Fig.12(a)), i.e., they require different SNR to achieve the same path detection probability. The threshold-based methods can be, however, adjusted by selecting the quantile ρ appropriately. As we see, with $\rho = 1 - 10^{-6}$, the threshold-based and SAGE+MDL methods achieve the same probabilities of path detection. The resulting probabilities of correct path extraction are shown in Fig. 12(b). Note that for low SNR comparison of the methods is meaningless, since too few paths are detected. However, above SNR $\approx$ 15dB, with all methods we can achieve similar high path detection probability, which allows direct comparison of the correct path extraction probabilities. We can hence infer that, in this regime, model selection with negative $\log$-evidence is superior to other methods, since it has higher probabilities of path extraction. In other words this means that at higher SNR this method will introduce fewer artifacts.

What is also important is that as the SNR increases, the correct path extraction rate drops. This happens simply because our model has a fixed resolution in the delay. As the result, at the higher SNR several components from the our model are used to approximate a single one with a delay between the sampling instances. This leads to the degradation of the correct path extraction rate since the number of components is overestimated.

Now, let us increase the sampling rate and study the case $N_s = 2$ (Fig. 12(c), and Fig. 12(d)). We see that the probabilities of path extraction are now higher for all methods. A slight difference between the two EP initialization schemes can also be observed. Note however that the performance increase is higher for the SAGE+MDL and negative $\log$-evidence algorithms, which both rely on the same model selection concept.

Finally, the last case with $N_s = 4$ is shown in Fig. 12(e) and Fig. 12(f). Again SAGE+MDL and negative $\log$-evidence schemes achieve higher correct path extraction probabilities as compared to the threshold-based method. The performance of the latter also increases with the sampling rate, but un-

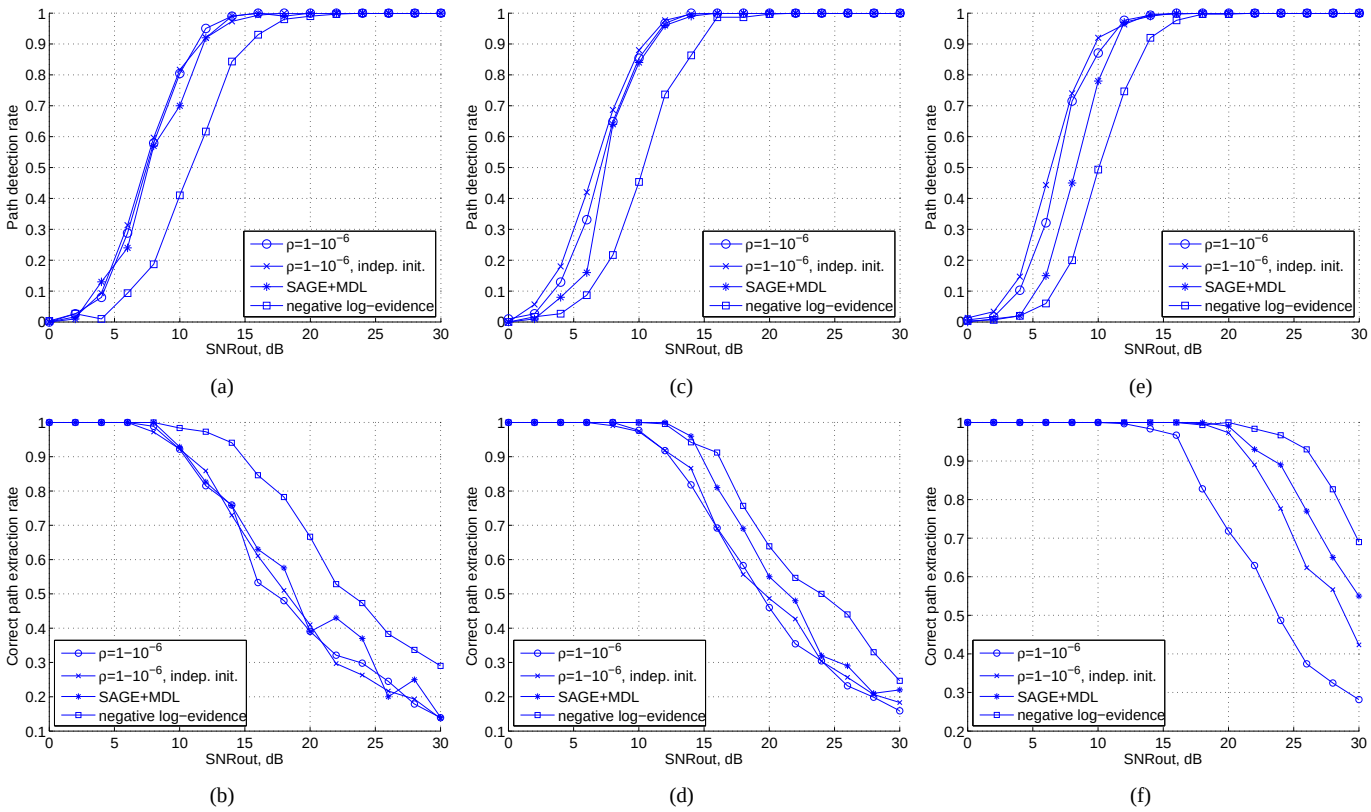


Fig. 12: Comparison of the model selection schemes in a single path scenario. (a,c,e) path detection probability, and (b,d,f) probability of correct path extraction for $P = 5$, and (a,b) $N_s = 1$; (c,d) $N_s = 2$; and (e,f) $N_s = 4$.

fortunately not as fast as that of the Description Length-based model selection. Theses plots also demonstrate the difference between the two proposed initializations of the EP. In Fig. 12(e) we see that in this case the independent initialization outperforms the joint one. As already mentioned, this distinction becomes noticeable, once the basis functions in $\boldsymbol{K}$ exhibit significant correlation, what is the case for $N_s \gtrsim 2$.

C. Results for measured channels

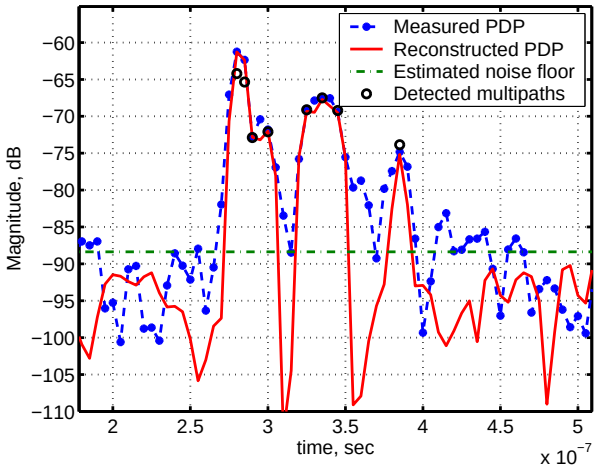
We also apply the proposed algorithm to the measured data collected in in-door environments. Channel measurements were done with the MIMO channel sounder PropSound manufactured by Elektrobit Oy. The basic setup for channel sounding is equivalent to the block-diagram shown in Fig. 1. In the conducted experiment the sounder operated at the carrier frequency 5.2GHz with a chip period of $T_p = 10$nsec. The output of the matched filter was sampled with the period $T_s = T_p/2$, thus resulting in a resolution of 2 samples per chip. The sounding sequence consisted of $M = 255$ chips, resulting in the burst waveform duration of $T_u = MT_p = 0.255\mu$sec.

Based on visual inspection of the PDP of the measured channels, the delays T_l in the search space $\mathcal{T}$ are positioned uniformly in the interval between 250nsec and 1000nsec, with spacing between adjacent delays equal to T_s. This corresponds to the delay search space $\mathcal{T}$ consisting of 151 elements. The initial estimate of the noise floor is obtained from the tail of the measured PDP. The algorithm stops once the relative change of the evidence parameters between two successive iterations is smaller than 0.0001%. The corresponding detection results for different number of channel observations are shown in Fig. 13.

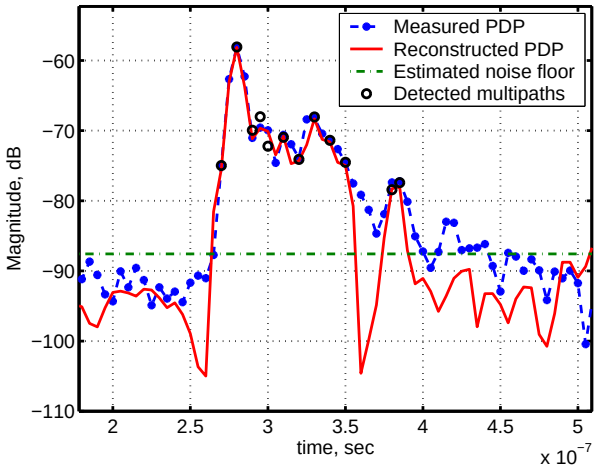
When $P = 1$ (see Fig. 13(a)), the independent initialization results in only 9 basis functions constituting the initial hypothesis $\mathcal{H}_0$. The final estimated number of components is found to be $L = 8$. As expected, increasing the number of channel observations P makes it possible to detect and estimate components with smaller SNR. For the case of $P = 5$ we detect already $L = 12$ components (Fig. 13(b)), and for $P = 32$, $L = 15$ components (Fig. 13(c)). This shows that increasing the number of observations not necessarily brings a proportional increase of the detected components, thus suggesting that there might be a limit given by the true number of multipath components.

VI. Conclusion

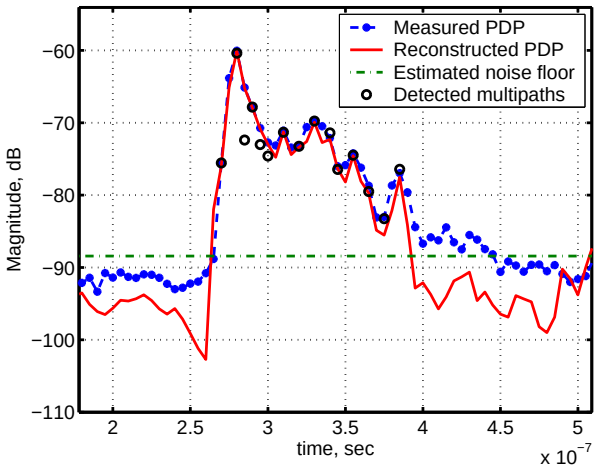
This paper demonstrates the application of the Evidence Procedure to the analysis of wireless channels. The original formulation of this method, known as Relevance Vector Machines, was reformulated to cope with the estimation of wireless channels. We extended the method to the complex domain and colored additive noise. We further extended the RVM to multiple channels by proposing a new graphical Bayesian model, where a single evidence parameter controls each multipath component observed with multiple channels. To our knowledge this is a new concept that can be useful not only for estimation, but also for simulating wireless channels.



(a) $P = 1$; Estimated number of multipath components $L = 8$.



(b) $P = 5$; Estimated number of multipath components $L = 12$.



(c) $P = 32$; Estimated number of multipath components $L = 15$.

Fig. 13: Multipath detection results for quantile-based method with $\rho = 1 - 10^{-6}$.

Evidence parameters were originally introduced to control the sparsity of the model. Assuming a single path scenario we were able to find the statistical laws that govern the values of the evidence parameters once the estimation algorithm has converged to the stationary point. It was shown that in low SNR scenarios the evidence parameters do not attain infinite values, as has been assumed in the Tipping's original RVM formulation, but stay finite with values depending on the particular SNR level. This knowledge enabled us to develop model selection rules based on the discovered statistical laws behind the evidence parameters.

In order to be able to apply these rules in practice, we also proposed a modified learning algorithm that exploits the principle of successive interference cancellation. This modification not only allows to avoid computationally intensive matrix inversions, but also removes the interference between the neighboring basis functions in the design matrix.

Model mismatch case was also considered in our analysis. We were able to assess the possible influence of the finite algorithm resolution and, to some extent, take it into account by adjusting the corresponding model selection rules.

We also showed the relationship between the EP and the classical model selection based on the MDL criterion. It was found that the maximum of the evidence corresponds to the minimum of the corresponding description length criterion. Thus, EP can be used as the classical MDL-like model selection scheme, but also allows faster and more efficient threshold-based implementation.

The EP framework was also compared with the multipath estimation using the SAGE algorithm augmented with the MDL criterion.

According to the simulation results, the Description Length-based methods, i.e., negative $\log$-evidence and SAGE+MDL method, give better results in terms of the achieved probabilities of correct path extraction. They also improve faster as the sampling rate grows. However, these model selection strategies require learning multiple models in parallel, which, of course, imposes additional computational load. The threshold-based method, on the other hand, allow to perform model selection on-line, thus being more efficient, but its performance increase with the growing sampling rate is more modest. The performance of the threshold-based method also depends on the value of the quantile ρ. In our simulations we set $\rho = 1-10^{-6}$, which results in the same probability of the path detection as in the SAGE+MDL algorithm. However, other values of ρ can be used, thus giving a way to further optimize the performance of the threshold-based method.

The comparison between the SAGE and EP schemes clearly shows that estimating evidence parameters really pays off. Introducing them in the computation of the model complexity, as it is done in the negative $\log$-evidence approach, results in the best performance, compared to the other two methods. Although the negative $\log$-evidence methods needs a slightly higher SNR to reliably detect channels, it however results in the highest probability of the path extraction.

To summarize, we think that the EP is a very promising method that can be superior to the standard model selection algorithms like MDL, both in accuracy and in computational

efficiency. It also offers a number of possibilities: the evidence parameters can also be estimated within the SAGE framework, thus extending the list of multipath parameters and enabling on-line model selection within the SAGE algorithm. As the consequence, this would allow to adapt the design matrix by estimating the delays τ_l from the data. The threshold-based method also opens perspectives for on-line remodeling, i.e., adding or removing components during the estimation of the model parameters, which might result in much better and sparser models. Since the evidence paremeters reflect the contribution of the multipath components, they might also be useful in applications, where it is necessary to define some measure of confidence for a multipath component.

APPENDIX

To derive the update expressions for the evidence parameters in the multiple channels case, we first rewrite (19) using the definitions (22). Since both terms under the integral are Gaussian densities, the result can be easily evaluated as

$$p(\tilde{\boldsymbol{z}}|\boldsymbol{\alpha},\beta) = \int p(\tilde{\boldsymbol{z}}|\tilde{\boldsymbol{w}},\beta)p(\tilde{\boldsymbol{w}}|\boldsymbol{\alpha})d\tilde{\boldsymbol{w}} = \frac{\exp\Big(-\tilde{\boldsymbol{z}}^H(\beta^{-1}\tilde{\boldsymbol{\Lambda}}+\tilde{\boldsymbol{K}}\tilde{\boldsymbol{A}}^{-1}\tilde{\boldsymbol{K}}^H)^{-1}\tilde{\boldsymbol{z}}\Big)}{\pi^{PN}|\beta^{-1}\tilde{\boldsymbol{\Lambda}}+\tilde{\boldsymbol{K}}\tilde{\boldsymbol{A}}^{-1}\tilde{\boldsymbol{K}}^H|}. \quad (64)$$

For the sake of completeness we also consider hypermodel priors $p(\boldsymbol{\alpha},\beta)$ in the derivation of the hyperparameter update expressions. Thus, our goal is to find the values of $\boldsymbol{\alpha}$, and β that maximize $\mathcal{L}(\boldsymbol{\alpha},\beta|\tilde{\boldsymbol{z}}) = \log(p(\tilde{\boldsymbol{z}}|\boldsymbol{\alpha},\beta)p(\boldsymbol{\alpha},\beta))$. This is achieved by taking the partial derivatives of $\mathcal{L}(\boldsymbol{\alpha},\beta|\tilde{\boldsymbol{z}})$ with respect to $\boldsymbol{\alpha}$ and β, and equating them to zero [19]. It is convenient to maximize $\mathcal{L}(\boldsymbol{\alpha},\beta|\tilde{\boldsymbol{z}})$ with respect to $\log(\alpha_l)$ and $\log(\beta)$ since the derivatives of the prior terms in the logarithmic domain are simpler.

First we prove the following matrix identity that we will exploit later

$$|\boldsymbol{B}^{-1}||\boldsymbol{A}^{-1}||\boldsymbol{A}+\boldsymbol{K}^H\boldsymbol{B}\boldsymbol{K}| = |\boldsymbol{B}^{-1}+\boldsymbol{K}\boldsymbol{A}^{-1}\boldsymbol{K}^H|. \quad (65)$$

Proof:

$$|\boldsymbol{B}^{-1}||\boldsymbol{A}^{-1}||\boldsymbol{A}+\boldsymbol{K}^H\boldsymbol{B}\boldsymbol{K}| =$$
$$|\boldsymbol{B}^{-1}||\boldsymbol{A}^{-1}||\boldsymbol{K}^H[(\boldsymbol{K}\boldsymbol{A}^{-1}\boldsymbol{K}^H)^{-1}+\boldsymbol{B}]\boldsymbol{K}| =$$
$$|\boldsymbol{B}^{-1}||\boldsymbol{A}^{-1}||\boldsymbol{K}||(\boldsymbol{K}\boldsymbol{A}^{-1}\boldsymbol{K}^H)^{-1}+\boldsymbol{B}||\boldsymbol{K}^H| =$$
$$|\boldsymbol{K}||\boldsymbol{A}^{-1}||\boldsymbol{K}^H||[(\boldsymbol{K}\boldsymbol{A}^{-1}\boldsymbol{K}^H)^{-1}+\boldsymbol{B}]\boldsymbol{B}^{-1}| =$$
$$|\boldsymbol{K}\boldsymbol{A}^{-1}\boldsymbol{K}^H[(\boldsymbol{K}\boldsymbol{A}^{-1}\boldsymbol{K}^H)^{-1}\boldsymbol{B}^{-1}+\boldsymbol{I}]| =$$
$$|\boldsymbol{B}^{-1}+\boldsymbol{K}\boldsymbol{A}^{-1}\boldsymbol{K}^H|$$

■

Now, we can begin with the derivation of the update of the hyperparameters α_l. Let us define $\tilde{\boldsymbol{B}}^{-1} = \beta^{-1}\tilde{\boldsymbol{\Lambda}}$. According to (65) we see that

$$|\tilde{\boldsymbol{B}}^{-1}+\tilde{\boldsymbol{K}}\tilde{\boldsymbol{A}}^{-1}\tilde{\boldsymbol{K}}^H| =$$
$$|\tilde{\boldsymbol{B}}^{-1}||\tilde{\boldsymbol{A}}^{-1}||\tilde{\boldsymbol{A}}+\tilde{\boldsymbol{K}}^H\tilde{\boldsymbol{B}}\tilde{\boldsymbol{K}}| = |\tilde{\boldsymbol{B}}^{-1}||\tilde{\boldsymbol{A}}^{-1}||\tilde{\boldsymbol{\Phi}}^{-1}|.$$

Making use of this result, we can write

$$\frac{\partial\mathcal{L}(\boldsymbol{\alpha},\beta|\tilde{\boldsymbol{z}})}{\partial\log(\alpha_l)} = \frac{\partial}{\partial\log\alpha_l}\Bigg\{-\log|\tilde{\boldsymbol{B}}^{-1}||\tilde{\boldsymbol{A}}^{-1}||\tilde{\boldsymbol{\Phi}}^{-1}|-$$
$$\tilde{\boldsymbol{z}}^H(\tilde{\boldsymbol{B}}^{-1}+\tilde{\boldsymbol{K}}\tilde{\boldsymbol{A}}^{-1}\tilde{\boldsymbol{K}}^H)^{-1}\tilde{\boldsymbol{z}}+\sum_{l=1}^{L}(\epsilon\log\alpha_l-\zeta\alpha_l)\Bigg\} =$$
$$\frac{\partial\log|\boldsymbol{A}|^P}{\partial\log\alpha_l}+\sum_{p=1}^{P}\frac{\partial\log|\boldsymbol{\Phi}_p|}{\partial\log\alpha_l}+(\epsilon-\zeta\alpha_l)$$
$$-\tilde{\boldsymbol{z}}^H\frac{\partial(\tilde{\boldsymbol{B}}-\tilde{\boldsymbol{B}}\tilde{\boldsymbol{K}}(\tilde{\boldsymbol{A}}+\tilde{\boldsymbol{K}}^H\tilde{\boldsymbol{B}}\tilde{\boldsymbol{K}})^{-1}\tilde{\boldsymbol{K}}^H\tilde{\boldsymbol{B}})}{\partial\log\alpha_l}\tilde{\boldsymbol{z}},$$

where in the latter expression the Woodbury inversion identity [28] was used to expand the term $(\tilde{\boldsymbol{B}}^{-1}+\tilde{\boldsymbol{K}}\tilde{\boldsymbol{A}}^{-1}\tilde{\boldsymbol{K}}^H)^{-1}$. After taking the derivative we arrive at

$$\frac{\partial\mathcal{L}(\boldsymbol{\alpha},\beta|\tilde{\boldsymbol{z}})}{\partial\log(\alpha_l)} = P\,\mathrm{tr}\Big[\boldsymbol{A}^{-1}\frac{\partial\boldsymbol{A}}{\partial\log\alpha_l}\Big]+\sum_{p=1}^{P}\mathrm{tr}\Big[\boldsymbol{\Phi}_p^{-1}\frac{\partial\boldsymbol{\Phi}_p}{\partial\log\alpha_l}\Big]$$
$$+(\epsilon-\zeta\alpha_l)-\tilde{\boldsymbol{z}}^H\tilde{\boldsymbol{B}}\tilde{\boldsymbol{K}}\tilde{\boldsymbol{\Phi}}\frac{\partial(\tilde{\boldsymbol{A}}+\tilde{\boldsymbol{K}}^H\tilde{\boldsymbol{B}}\tilde{\boldsymbol{K}})}{\partial\log\alpha_l}\tilde{\boldsymbol{\Phi}}\tilde{\boldsymbol{K}}^H\tilde{\boldsymbol{B}}\tilde{\boldsymbol{z}} =$$
$$P-\sum_{p=1}^{P}\mathrm{tr}\Big[\alpha_l\boldsymbol{E}_{ll}\boldsymbol{\Phi}_p\Big]+(\epsilon-\zeta\alpha_l)-$$
$$\tilde{\boldsymbol{z}}^H\tilde{\boldsymbol{B}}\tilde{\boldsymbol{K}}\tilde{\boldsymbol{\Phi}}\alpha_l\tilde{\boldsymbol{E}}_{ll}\tilde{\boldsymbol{\Phi}}\tilde{\boldsymbol{K}}^H\tilde{\boldsymbol{B}}\tilde{\boldsymbol{z}}.$$

Here $\boldsymbol{E}_{ll}$ is a matrix with the lth element on the main diagonal equal to 1, and all other elements being zero. Similarly, $\tilde{\boldsymbol{E}}_{ll}$ is the P-times repetition of $\boldsymbol{E}_{ll}$ on its main diagonal. By noting that $\tilde{\boldsymbol{\mu}} = \tilde{\boldsymbol{\Phi}}\tilde{\boldsymbol{K}}^H\tilde{\boldsymbol{B}}\tilde{\boldsymbol{z}}$, we arrive at

$$\frac{\partial\mathcal{L}(\boldsymbol{\alpha},\beta|\tilde{\boldsymbol{z}})}{\partial\log(\alpha_l)} = P-\sum_{p=1}^{P}\mathrm{tr}\Big[\alpha_l\boldsymbol{E}_{ll}\boldsymbol{\Phi}_p\Big]+$$
$$(\epsilon-\zeta\alpha_l)-\tilde{\boldsymbol{\mu}}^H\alpha_l\tilde{\boldsymbol{E}}_{ll}\tilde{\boldsymbol{\mu}} = 0.$$

Solving for α_l, we obtain the final expression for the hyperparameter update

$$\alpha_l = \frac{P+\epsilon}{\sum_{p=1}^{P}\Big(\Phi_{p,ll}+|\mu_{p,l}|^2\Big)+\zeta}.$$

Note that by setting $\zeta = \epsilon = 0$ we effectively remove the influence of the prior $p(\boldsymbol{\alpha}|\zeta,\epsilon)$.

We proceed similarly to calculate the update of β

$$\frac{\partial\mathcal{L}(\boldsymbol{\alpha},\beta|\tilde{\boldsymbol{z}})}{\partial\log(\beta)} = \sum_{p=1}^{P}\frac{\partial\log|\boldsymbol{B}_p|}{\partial\log\beta}+\sum_{p=1}^{P}\frac{\partial\log|\boldsymbol{\Phi}_p|}{\partial\log\beta}+(\upsilon-\kappa\beta)$$
$$-\tilde{\boldsymbol{z}}^H\frac{\partial(\tilde{\boldsymbol{B}}-\tilde{\boldsymbol{B}}\tilde{\boldsymbol{K}}(\tilde{\boldsymbol{A}}+\tilde{\boldsymbol{K}}^H\tilde{\boldsymbol{B}}\tilde{\boldsymbol{K}})^{-1}\tilde{\boldsymbol{K}}^H\tilde{\boldsymbol{B}})}{\partial\log\beta}\tilde{\boldsymbol{z}} =$$
$$\sum_{p=1}^{P}\frac{\partial\log\beta^N|\boldsymbol{\Lambda}_p^{-1}|}{\partial\log\beta}+\sum_{p=1}^{P}\mathrm{tr}\Big[\boldsymbol{\Phi}_p^{-1}\frac{\partial\boldsymbol{\Phi}_p}{\partial\log\beta}\Big]+$$
$$(\upsilon-\kappa\beta)-\tilde{\boldsymbol{z}}^H\frac{\partial\beta\tilde{\boldsymbol{\Lambda}}^{-1}}{\partial\log\beta}\tilde{\boldsymbol{z}}+$$
$$\tilde{\boldsymbol{z}}^H\frac{\partial(\beta\tilde{\boldsymbol{\Lambda}}^{-1}\tilde{\boldsymbol{K}}(\tilde{\boldsymbol{A}}+\tilde{\boldsymbol{K}}^H\beta\tilde{\boldsymbol{\Lambda}}^{-1}\tilde{\boldsymbol{K}})^{-1}\tilde{\boldsymbol{K}}^H\beta\tilde{\boldsymbol{\Lambda}}^{-1})}{\partial\log\beta}\tilde{\boldsymbol{z}} =$$

$$PN - \sum_{p=1}^{P} \mathrm{tr}\left[\boldsymbol{\Phi}_p^{-1}\boldsymbol{\Phi}_p \frac{\partial(\boldsymbol{A} + \boldsymbol{K}_p^H \beta \boldsymbol{\Lambda}_p^{-1}\boldsymbol{K}_p)}{\partial \log\beta}\boldsymbol{\Phi}_p\right] +$$
$$(\upsilon - \kappa\beta) - \tilde{\boldsymbol{z}}^H\beta\tilde{\boldsymbol{\Lambda}}^{-1}\tilde{\boldsymbol{z}} + \tilde{\boldsymbol{z}}^H\beta\tilde{\boldsymbol{\Lambda}}^{-1}\tilde{\boldsymbol{K}}\tilde{\boldsymbol{\Phi}}\tilde{\boldsymbol{K}}^H\beta\tilde{\boldsymbol{\Lambda}}^{-1}\tilde{\boldsymbol{z}} +$$
$$\tilde{\boldsymbol{z}}^H\beta\tilde{\boldsymbol{\Lambda}}^{-1}\tilde{\boldsymbol{K}}\frac{\partial(\tilde{\boldsymbol{A}} + \tilde{\boldsymbol{K}}^H\beta\tilde{\boldsymbol{\Lambda}}^{-1}\tilde{\boldsymbol{K}})}{\partial\log\beta}\tilde{\boldsymbol{K}}^H\beta\tilde{\boldsymbol{\Lambda}}^{-1}\tilde{\boldsymbol{z}} +$$
$$\tilde{\boldsymbol{z}}^H\beta\tilde{\boldsymbol{\Lambda}}^{-1}\tilde{\boldsymbol{K}}\tilde{\boldsymbol{\Phi}}\tilde{\boldsymbol{K}}^H\beta\tilde{\boldsymbol{\Lambda}}^{-1}\tilde{\boldsymbol{z}} =$$
$$PN - \sum_{p=1}^{P} \mathrm{tr}\left[\boldsymbol{K}_p^H\beta\boldsymbol{\Lambda}_p^{-1}\boldsymbol{K}_p\boldsymbol{\Phi}_p\right] +$$
$$(\upsilon - \kappa\beta) - \tilde{\boldsymbol{z}}^H\beta\tilde{\boldsymbol{\Lambda}}^{-1}\tilde{\boldsymbol{z}} + \tilde{\boldsymbol{z}}^H\beta\tilde{\boldsymbol{\Lambda}}^{-1}\tilde{\boldsymbol{K}}\tilde{\boldsymbol{\mu}}$$
$$+\tilde{\boldsymbol{\mu}}^H\tilde{\boldsymbol{K}}^H\beta\tilde{\boldsymbol{\Lambda}}^{-1}\tilde{\boldsymbol{K}}\tilde{\boldsymbol{\mu}} + \tilde{\boldsymbol{\mu}}^H\tilde{\boldsymbol{K}}^H\beta\tilde{\boldsymbol{\Lambda}}^{-1}\tilde{\boldsymbol{z}}.$$

Thus we arrive at the final expression:

$$\frac{\partial\mathcal{L}(\boldsymbol{\alpha},\beta|\tilde{\boldsymbol{z}})}{\partial\log(\beta)} = PN - \sum_{p=1}^{P}\mathrm{tr}\left[\boldsymbol{K}_p^H\beta\boldsymbol{\Lambda}_p^{-1}\boldsymbol{K}_p\boldsymbol{\Phi}_p\right] +$$
$$(\upsilon - \kappa\beta) - \sum_{p=1}^{P}(\boldsymbol{z}_p - \boldsymbol{K}_p\boldsymbol{\mu}_p)^H\beta\boldsymbol{\Lambda}_p^{-1}(\boldsymbol{z}_p - \boldsymbol{K}_p\boldsymbol{\mu}_p) = 0.$$

Solving for β we finally obtain

$$\beta = (PN + \upsilon)\left(\sum_{p=1}^{P}\mathrm{tr}\left[\boldsymbol{K}_p^H\boldsymbol{\Lambda}_p^{-1}\boldsymbol{K}_p\boldsymbol{\Phi}_p\right] + \sum_{p=1}^{P}(\boldsymbol{z}_p - \boldsymbol{K}_p\boldsymbol{\mu}_p)^H\boldsymbol{\Lambda}_p^{-1}(\boldsymbol{z}_p - \boldsymbol{K}_p\boldsymbol{\mu}_p) + \kappa\right)^{-1}.$$

Here again the choice $\kappa = \upsilon = 0$ removes the influence of the prior $p(\beta|\kappa,\upsilon)$ on the evidence maximization.

REFERENCES

[1] H. Krim and M. Viberg, "Two decades of array signal processing research: the parametric approach," *IEEE Signal Processing Mag.*, pp. 67–94, July 1996.

[2] B. Fleury, M. Tschudin, R. Heddergott, D. Dahlhaus, and K. Ingeman Pedersen, "Channel parameter estimation in mobile radio environments using the SAGE algorithm," *IEEE Journal on Selected Areas in Communications*, vol. 17, no. 3, pp. 434–450, March 1999.

[3] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*, 2nd ed. John Wiley & Sons, Inc., 2000.

[4] T. Wax, M. Kailath, "Detection of signals by information theoretic criteria," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 33, no. 2, pp. 387– 392, April 1985.

[5] J. Rissanen, "Modeling by the shortest data description." *Automatica 14*, pp. 465–471, 1978.

[6] S. Haykin, Ed., *Kalman Filtering and Neural Networks*. John Wiley & Sons, Inc., 2001.

[7] M. Feder and E. Weinstein, "Parameter estimation of superimposed signals using the EM algorithm," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 36, no. 4, pp. 477–489, April 1988.

[8] D. J. MacKay, *Information Theory, Inference, and Learning Algorithms*. Cambridge University Press, 2003.

[9] W. Fitzgerald, "The Bayesian approach to signal modelling." in *IEE Colloquium on Non-Linear Signal and Image Processing (Ref. No. 1998/284)*, May 1998, pp. 9/1–9/5.

[10] G. Schwarz, "Estimating the dimension of a model." *Annals of Statistics*, vol. 6, no. 2, p. 461464, 1978.

[11] J. J. Rissanen, "Fisher information and stochastic complexity," *IEEE Transactions on Information Theory*, vol. 42, no. 1, pp. 40–47, January 1996.

[12] A. Lanterman, "Schwarz, Wallace, and Rissanen: Intertwining themes in theories of model order estimation," *International Statistical Review*, vol. 69, no. 2, pp. 182–215, January 2001.

[13] D. J. C. MacKay, "Bayesian interpolation," *Neural Computation*, vol. 4, no. 3, pp. 415–447, 1992.

[14] ——, "Bayesian methods for backpropagation networks," in *Models of Neural Networks III*, E. Domany, J. L. van Hemmen, and K. Schulten, Eds. New York: Springer-Verlag, 1994, ch. 6, pp. 211–254.

[15] M. Tipping, "Sparse Bayesian learning and the Relevance Vector Machine," *Journal of Machine Learning Research*, vol. 1, pp. 211–244, June 2001.

[16] T. S. Rappaport, *Wireless communications. Principles and practice.* Prentice Hall PTR, 2002.

[17] D. Heckerman, "A tutorial on learning with bayesian networks," Microsoft Research, Advanced Technology Division, One Microsoft Way Redmond, WA 98052, Tech. Rep. MSR-TR-95-06, March 1995.

[18] R. Neal, *Bayesian Learning for Neural Networks*, ser. Lecture Notes in Statistics. New York: Springer-Verlag, 1996, vol. 118.

[19] O. Berger, *Statistical decision theory and Bayesian analysis.*, 2nd ed. Springer, 1985.

[20] P. Grünwald, "A tutorial introduction to the minimum description length principle." in *Advances in Minimum Description Length: Theory and Applications*, P. Grünwald, I. Myung, and M. Pitt, Eds. MIT Press, 2005.

[21] A. Barron, J. Rissanen, and B. Yu, "The minimum description length principle in coding and modeling," *IEEE Transactions on Information Theory*, vol. 44, no. 6, pp. 2743–2760, October 1998.

[22] A. C. Faul and M. E. Tipping, "Analysis of sparse Bayesian learning," in *Advances in Neural Information Processing Systems*, T. G. Dietterich, S. Becker, and Z. Ghahramani, Eds., vol. 14, no. 383-389. MIT Press, 2002.

[23] K. Conradsen, A. Nielsen, J. Schou, and H. Skriver, "A test statistic in the complex Wishart distribution and its application to change detection in polarimetric SAR data," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 41, no. 1, pp. 4–19, 2003.

[24] N. Goodman, "Statistical analysis based on a certain multivariate complex Gaussian distribution (An introduction)," *Ann. Math. Stat.*, vol. 34, pp. 152–177, 1963.

[25] M. Evans, N. Hastings, B., and Peacock, *Statistical Distributions*, 3rd ed. New York: Wiley, 2000.

[26] M. Abramowitz and I. A. Stegun, *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables*. Dover, 1972.

[27] T. Laakso, V. Välimäki, M.Karjalainen, and U. Laine, "Splitting the unit delay [FIR/all pass filters design]," *IEEE Signal Processing Magazine*, vol. 13, no. 1, pp. 30–60, January 1996.

[28] G. H. Golub and C. F. V. Loan, *Matrix Computations*. The Johns Hopkins University Press, 1996.