Infinite Factorial Finite State Machine for Blind Multiuser Channel Estimation

Francisco J. R. Ruiz⬤ , Isabel Valera, Lennart Svensson⬤ , and Fernando Perez-Cruz⬤ , *Senior Member, IEEE*

Abstract—New communication standards need to deal with machine-to-machine communications, in which users may start or stop transmitting at any time in an asynchronous manner. Thus, the number of users is an unknown and time-varying parameter that needs to be accurately estimated in order to properly recover the symbols transmitted by all users in the system. In this paper, we address the problem of joint channel parameter and data estimation in a multiuser communication channel in which the number of transmitters is not known. For that purpose, we develop the infinite factorial finite state machine model, a Bayesian nonparametric model based on the Markov Indian buffet that allows for an unbounded number of transmitters with arbitrary channel length. We propose an inference algorithm that makes use of slice sampling and particle Gibbs with ancestor sampling. Our approach is fully blind as it does not require a prior channel estimation step, prior knowledge of the number of transmitters, or any signaling information. Our experimental results, loosely based on the LTE random access channel, show that the proposed approach can effectively recover the data-generating process for a wide range of scenarios, with varying number of transmitters, number of receivers, constellation order, channel length, and signal-to-noise ratio.

Index Terms—Bayesian nonparametrics, stochastic finite state machine, multiuser communications, machine-to-machine.

I. INTRODUCTION

ONE OF the trends in wireless communication networks (WCNs) is the increase of heterogeneity [1]. Nowadays, users of WCNs are no longer only humans talking, and the number of services and users are booming. Machine-to-machine (M2M) communications and the Internet of Things (IoT) will shape the traffic in WCN in the years to come [2]–[5]. While there are millions of M2M cellular devices already using second, third and fourth generation cellular networks, the industry expectation is that the number of devices will increase ten-fold in the coming years [6].

M2M traffic, which also includes communication between a sensor/actuator and a corresponding application server in the network, is distinct from consumer traffic, which has been the main driver for the design of long term evolution (LTE) systems. First, while current consumer traffic is characterized by small number of long lived sessions, M2M traffic involves a large number of short-lived sessions, with typical transactions of a few hundred bytes. The short payloads involved in M2M communications make it highly inefficient to establish dedicated bearers for data transmission. Therefore, in some cases it is better to transmit small payloads in the random access request itself [7]. Second, a significant number of battery powered devices are expected to be deployed at adverse locations such as basements and tunnels (e.g., underground water monitors and traffic sensors) that demand superior link budgets. Motivated by this need for increasing the link budget for M2M devices, transmission techniques that minimize the transmit power for short burst communications are needed [8]. Third, the increasing number of M2M devices requires new techniques on massive access management [9], [10]. Due to these differences, there are strong arguments for the need to optimize WCNs specifically for M2M communications [6].

The nature of M2M traffic leads to multiuser communication systems in which a large number of users may aim at entering or leaving the system (i.e., start or stop transmitting) at any given time. In this context, we need a method that allows the users to access the system in a way that the signaling overhead is reduced. The method should determine the number of users transmitting in a communication system, and jointly perform channel estimation and detect the transmitted data, with minimum pilot signaling. This problem appears in several specific applications. For instance, in the context of wireless sensor networks, where the communication nodes can often switch on and off asynchronously during operation. It also appears in massive multiple-input multiple-output (MIMO) multiuser communication systems [11], [12], in which the base station has a very large number of antennas and the mobile devices use a single antenna to communicate within the network. In a code division multiple access (CDMA) context, a set of terminals randomly access the channel to communicate with a common access point, which receives the superposition

of signals from the active terminals only [13]. In these applications, the number of users is an unknown and time-varying parameter that we need to infer.

In this paper, we aim at solving the channel estimation and symbol detection problems in a fully unsupervised way, without the need of signaling data. Our approach is thus suitable for applicability on the random access channel, in which more than one terminal may decide to transmit data at the same time. To this end, we advocate for the use of Bayesian nonparametric (BNP) tools, which can adapt to heterogeneous structures by considering an unbounded number of users in their prior. We develop a BNP model that has the required flexibility to account for any number of transmitters without the need of additional previous knowledge or bounds. We are interested in showing that BNP models can solve the problem blindly and that versatility can improve detection in heterogeneous WCNs. Thus, or goal is to show that we can recover the messages and infer the number of transmitters using as little information as possible. The transmitted messages are therefore assumed to lack additional pilots or other structure. This makes our approach applicable on any interference-limited multiuser communication system in which the transmission of pilots or the need for synchronization may severely limit the spectral efficiency of the WCN.

The main difference between our model and the existing approaches in the literature is that our model is fully blind—with respect to the channel coefficients, the number of transmitters, and the transmitted symbols—and can recover the number of users without the need of synchronization. Indeed, the problem of multi-user detection has been widely studied in the literature, using standard linear methods (least squares or minimum mean squared error) or non-linear approaches (e.g., successive interference cancellation or parallel interference cancellation) [14], [15]. Most of these approaches require synchronization techniques (see [13], [16], [17]). In order to perform the user identification step, the common approach is to pre-specify signature sequences that are specific for each user [18]–[20], which imposes a constraint on the total number of users in the system.

Our model also differs from existing approaches in other ways. For example, in [21] users may only enter or leave the system at certain pre-defined instants. In [22] the user identification step is performed first without symbol detection. The approach in [23] requires knowledge of the channel, and the method in [13] is restricted to flat-fading channels. The methods in [24] and [25] come at the cost of both important approximations and prohibitive computational cost (experiments are presented with short frames of only ten symbols). In contrast, our model is fully blind, it does not need a previous channel estimation step, and it is not restricted to flat-fading channels or very short frames.

All these methods assume that the number of users in the system is known, which makes sense in a DS-CDMA system but may represent a limitation in other scenarios. They also require synchronization mechanisms. We build on previous BNP approaches that do not have these constraints [26]–[28]. In particular, we extend the method in [28] to channels with memory, and we address the exponential complexity

of [26] and [27], which is only applicable for BPSK systems and for a channel length of at most 2. Instead, the complexity of our algorithm scales with the square of the channel length.

In summary, our model does not need any of the following requirements: a constraint on the number of users, that the users are synchronized with the network, that the users transmit a preamble, that the channel is known, or that the channel length equals one. Our model allows for an unbounded number of transmitters due to its nonparametric nature, and it is not restricted to memoryless channels because transmitters are modeled as finite state machines. Our goal is to show that we can recover the number of transmitters and their payloads with very little information.

Our model follows the three steps of cognitive communications. It gathers the incoming signal, which is a mixture of all transmitters that are active at a given time instant (perception); it learns the number of users, the channel they face, and it detects their payload in a blind way (learning); and it assigns the symbols from each user and passes that information to the network to act upon (decision-making). These features can mitigate the effect of collisions and reduce the number of retransmissions over the random access channel, which is of mayor concern in LTE-A systems when M2M communications start being a driving force in WCNs [29], [30].

A. Technical Contributions

We model the multiuser communication system as an infinite factorial finite state machine (IFFSM) in which a potentially infinite number of finite state machines (FSMs), each representing a single transmitter, contribute to generate the observations. The symbols transmitted by each user correspond to the inputs of each FSM, and its memory accounts for the multipath propagation between each transmitter-receiver pair. The output of the IFFSM corresponds to the received signal, which depends on the inputs and the current states of the active FSMs (i.e., the active transmitters), under noise. Our IFFSM considers that transmitters may start or stop transmitting at any time, and it ensures that only a finite subset of the users become active during any finite observation period, while the remaining (infinite) transmitters remain in an idle state (i.e., they do not transmit).

As for most BNP models, one of the main challenges of our IFFSM is posterior inference. We develop a suitable inference algorithm by building a Markov chain Monte Carlo (MCMC) kernel using particle Gibbs with ancestor sampling (PGAS) [31], a recently proposed algorithm that belongs to the broader family of particle Markov chain Monte Carlo (PMCMC) methods [32]. This algorithm presents quadratic complexity with respect to the memory length, avoiding the exponential complexity of previous approaches, such as forward-filtering backward-sampling schemes [26], [27], [33], [34]. Our experimental results, based on the LTE random access channel, show that the proposed approach efficiently solves user activity detection, channel estimation and data detection in a jointly and fully blind way.

B. Organization of the Paper

The rest of the paper is organized as follows. In Section II, we review the basic building blocks that we use to develop our model, namely, the stochastic finite-memory FSM and the Markov Indian buffet process (mIBP). Section III details our proposed IFFSM, whereas we describe the inference algorithm in Section IV. Sections V and VI are devoted to the experiments and conclusions, respectively.

II. BACKGROUND

Here we describe the two main building blocks that we use to develop our IFFSM: the FSM [35] and the mIBP [34].

A. Stochastic Finite State Machines

FSMs have been applied to a huge variety of problems in many areas, including biology (e.g., neurological systems), signal processing (e.g., speech modeling), control, communications and electronics [36], [37]. In its general form [35], an FSM is defined as an abstract machine consisting of:

- A set of states, which might include one or several initial and final states.
- A set of discrete or continuous inputs.
- A set of discrete or continuous outputs.
- A state transition function, which takes the current input and state and returns the next state.
- An emission function, which takes the current state and input and returns an output.

The transition and emission functions are, in general, stochastic (e.g., hidden Markov models (HMMs) [38], [39]), which implies that the input events and the states are not directly observable through the output events (instead, each input and state produces one of the possible output events with a certain probability). We focus on stochastic finite-memory FSMs, in which each state can be represented as a finite-length sequence containing the last L inputs. This FSM relies on a finite memory L and a finite alphabet $\mathcal{X}$. Each input $x_t \in \mathcal{X}$ produces a deterministic change in the state of the FSM, and a stochastic observable output $\mathbf{y}_t$ at time t. The next state and the output depend on the current state and the input. The state can be expressed as the vector containing the last L inputs, i.e., $[x_t, x_{t-1}, \ldots, x_{t-L+1}]$, therefore yielding $|\mathcal{X}|^L$ different states, where $|\mathcal{X}|$ denotes the cardinality of the set $\mathcal{X}$. Each state can only transition to $|\mathcal{X}|$ different states, depending on the next input event. A state diagram of an HMM and an FSM is depicted in Figure 1.

For any finite-memory FSM, the probability distribution over the inputs x_t and the observations (outputs) $\mathbf{y}_t$ can be written as

$$p\left(\{x_t, \mathbf{y}_t\}_{t=1}^T\right) = \prod_{t=1}^{T} p(x_t) \prod_{t=1}^{T} p(\mathbf{y}_t | x_t, \ldots, x_{t-L+1}), \quad (1)$$

i.e., the likelihood of each observation $\mathbf{y}_t$ depends not only on x_t, but also on the previous $L - 1$ inputs. The model also requires the specification of the initial state, which is defined by the inputs $x_0, \ldots, x_{1-L}$. We show the graphical model for an FSM with memory length $L = 2$ in Figure 2.
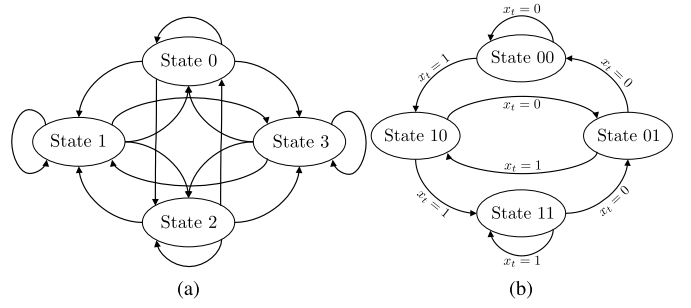

Fig. 1. (a) State diagram of an HMM with 4 states, in which all transitions among states are allowed. (b) State diagram of a stochastic finite-memory FSM with memory length $L = 2$ and $\mathcal{X} = \{0, 1\}$. Each state can be represented by the vector containing the last L input events. Each state can only transition to other $|\mathcal{X}| = 2$ states, depending on the input event x_t. Hence, not all transitions among states are allowed, but it is possible to reach any other state in the graph after $L = 2$ transitions.
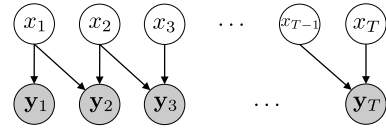

Fig. 2. Graphical representation of an FSM model with memory length $L = 2$. Here, x_t denotes the transmitted symbol at time t, and $\mathbf{y}_t$ is the received (observed) signal. The arrows indicate that the t-th received symbol depends on both the t-th and the $(t - 1)$-th transmitted symbols.

Note that this model can be equivalently represented as a standard HMM with a sparse transition probability matrix of size $|\mathcal{X}|^L \times |\mathcal{X}|^L$. However, the HMM representation of the FSM leads to a computationally intractable inference algorithm due to the exponential dependency on the memory length L of the state space cardinality.

B. Markov Indian Buffet Process

The central idea behind BNPs is the replacement of classical finite-dimensional prior distributions with general stochastic processes, allowing for an open-ended number of degrees of freedom in a model [40]. They constitute an approach to model selection and adaptation in which the model complexity is allowed to grow with data size [41].

Within the family of BNP models, the Markov Indian buffet process (mIBP) is a variant of the Indian buffet process (IBP) [42] and constitutes the main building block of the infinite factorial hidden Markov model (IFHMM), which considers an infinite number of first-order Markov chains with binary-valued states that evolve independently [34].

The mIBP places a prior distribution over a binary matrix $\mathbf{S}$ with a finite number of rows and an infinite number of columns. The t-th row represents time step t, whereas the m-th column contains the (binary) states of the m-th Markov chain. Each element $s_{tm} = (\mathbf{S})_{tm} \in \{0, 1\}$ indicates whether the m-th Markov chain is active at time instant t, with $t = 1, \ldots, T$, $m = 1, \ldots, M$, and $M \to \infty$. The states s_{tm} evolve according to the transition matrix

$$\mathbf{A}^m = \begin{pmatrix} 1 - a^m & a^m \\ 1 - b^m & b^m \end{pmatrix}, \quad (2)$$

i.e., $a^m = p(s_{tm} = 1|s_{(t-1)m} = 0)$ is the transition probability from inactive to active, and $b^m = p(s_{tm} = 1|s_{(t-1)m} = 1)$ is the self-transition probability of the active state. We assume an inactive initial state for all chains at time $t = 0$.

In order to specify the prior distribution over the transition probabilities a^m and b^m, we focus here on the stick-breaking construction of the IBP in [43], because it allows for simple and efficient inference algorithms. Under this construction, we first introduce the notation $a^{(m)}$ to denote the sorted values of a^m, such that $a^{(1)} > a^{(2)} > a^{(3)} > \ldots$, with

$$a^{(1)} \sim \text{Beta}(\alpha, 1), \tag{3}$$

and

$$p\left(a^{(m)}|a^{(m-1)}\right) \propto \left(a^{(m)}\right)^{\alpha-1} \mathbb{I}\left(0 \le a^{(m)} \le a^{(m-1)}\right), \tag{4}$$

being $\mathbb{I}(\cdot)$ the indicator function, which takes value one if its argument is true and zero otherwise. Here, α is the concentration hyperparameter of the model, which controls the expected number of latent Markov chains that become active *a priori*. The prior over variables b^m is given by

$$b^m \sim \text{Beta}(\beta_0, \beta_1), \tag{5}$$

which does not depend on the index m. In (3) and (5), α, β_0 and β_1 are hyperparameters of the model. This prior distribution ensures that, for any finite value of T, only a finite number of columns M_+ become active, while the rest of them remain in the all-zero state.

III. MODEL DESCRIPTION

We use the stochastic finite-memory FSM and the mIBP detailed in Section II as building blocks for our IFFSM model, which we use for modeling a multiuser communication system in which the number of transmitters is potentially infinite.

The key idea of our approach is to model the multiuser communication system as a factorial FSM in which each FSM represents a user, and the inputs to the FSM are the symbols sent by each transmitter. The memory of the FSM accounts for the multipath propagation between each transmitter-receiver pair, and the output of the factorial FSM model corresponds to the received signal, which depends on the inputs and the current states of all the active FSMs (transmitters).

In order to account for an unbounded number of transmitters (parallel FSMs), we need to consider an inactive state, such that the observations cannot depend on those transmitters that are inactive. Furthermore, we need to ensure that only a finite number of them become active for any finite-length observed sequence. As detailed in Section II-B, the mIBP in [34] meets these requirements by placing a prior distribution over binary matrices with an infinite number of columns.

A. Infinite Factorial Finite State Machine

We place a mIBP prior over an auxiliary binary matrix $\mathbf{S}$, where each element s_{tm} indicates whether the m-th FSM is active at time instant t. While active, the input symbol to the m-th FSM at time instant t, denoted by x_{tm}, is assumed to belong to the set $\mathcal{A}$, with finite cardinality $|\mathcal{A}|$. Here, $\mathcal{A}$ represents
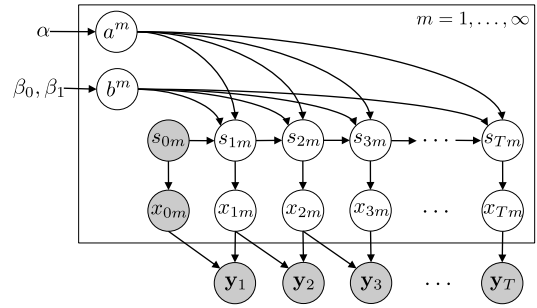

Fig. 3. Graphical model of the IFFSM with memory length $L = 2$. Here, x_{tm} denotes the transmitted symbol by device m at time instant t, and $\mathbf{y}_t$ are the observations (a combination of the signals transmitted by all the active devices). The variable s_{tm} indicates whether device m is active or inactive at time instant t. The variables a^m and b^m govern the transition probabilities from inactive to active and vice-versa. The hyperparameters of the model are α, β_0, and β_1.

the constellation, and x_{tm} stands for the transmitted symbol of the m-th transmitter at time instant t. We assume that the transmitted symbols are independent and uniformly distributed on the set $\mathcal{A}$. While inactive, we can assume that $x_{tm} = 0$ and, therefore, each input $x_{tm} \in \mathcal{X}$, with $\mathcal{X} = \mathcal{A} \bigcup \{0\}$. This can be expressed as

$$x_{tm}|s_{tm} \sim \begin{cases} \delta_0(x_{tm}) & \text{if } s_{tm} = 0, \\ \mathcal{U}(\mathcal{A}) & \text{if } s_{tm} = 1, \end{cases} \tag{6}$$

where $\delta_0(\cdot)$ denotes a point mass located at 0, and $\mathcal{U}(\mathcal{A})$ denotes the uniform distribution over the set $\mathcal{A}$.

In the IFFSM model, each input symbol x_{tm} does not only influence the observation $\mathbf{y}_t$ at time instant t, but also the future $L-1$ observations, $\mathbf{y}_{t+1}, \ldots, \mathbf{y}_{t+L-1}$. Therefore, the likelihood function for $\mathbf{y}_t$ depends on the last L input symbols of all the FSMs, yielding

$$p(\mathbf{y}_t|\mathbf{X}) = p\big(\mathbf{y}_t|\{x_{tm}, x_{(t-1)m}, \ldots, x_{(t-L+1)m}\}_{m=1}^{\infty}\big), \tag{7}$$

with $\mathbf{X}$ being the $T \times M$ matrix that contains all symbols x_{tm}. We assume innactive states $s_{tm} = 0$ for $t \le 0$ (note that $s_{tm} = 0$ implies $x_{tm} = 0$ and viceversa).

The resulting IFFSM model, particularized for $L = 2$, is shown in Figure 3. Note that this model can be equivalently represented as a non-binary version of the IFHMM in [34], using the extended states given by the vector $[x_{tm}, x_{(t-1)m}, \ldots, x_{(t-L+1)m}]$. However, we maintain the representation in Figure 3 because it allows deriving an efficient inference algorithm.

B. Observation Model

The model described in the previous section is general and can be applied to any sequence that can be explained by a potentially unbounded number of parallel FSMs. In this section, we particularize this model for its applicability on wireless communications, in which the parallel chains correspond to different transmitters. The IFFSM requires two general conditions for the likelihood model to be valid as the number of FSMs M tends to infinity: i) the likelihood must be invariant to permutations of the chains, and ii) the distribution on $\mathbf{y}_t$ cannot depend on any parameter of the m-th FSM
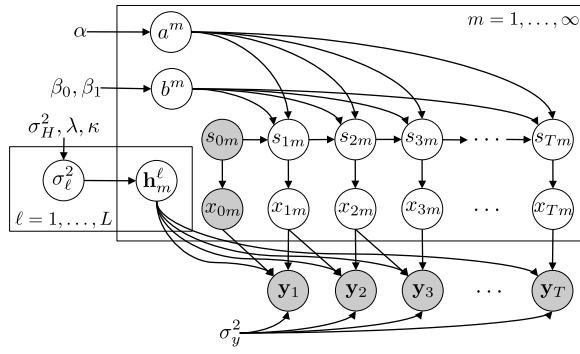
Fig. 4. Graphical Gaussian observation model for an IFFSM with memory length $L = 2$. It combines the building block from Figure 3 with the channel model coefficients $\mathbf{h}_m^\ell$, the prior variance of those channel coefficients σ_ℓ, and the noise variance σ_y^2.

if $s_{\tau m} = 0$ for $\tau = t - L + 1, \ldots, t$. These conditions are met by simultaneous transmissions in WCNs. The first one says that the likelihood does not depend on how we label the different transmitters, and the second one indicates that inactive transmitters should not affect the observations. Among others, the standard discrete-time interference channel model with additive white Gaussian noise fulfils these restrictions:

$$\mathbf{y}_t = \sum_{m=1}^{M_+} \sum_{\ell=1}^{L} \mathbf{h}_m^\ell x_{(t-\ell+1)m} + \mathbf{n}_t. \tag{8}$$

In (8), x_{tm} represents the complex input (i.e., a symbol from a given constellation) at time instant t for the m-th FSM (user), $\mathbf{h}_m^\ell$ are the channel coefficients (emission parameters, in the BNP literature), and $\mathbf{n}_t$ is the additive white noise, which is circularly symmetric complex Gaussian distributed,[1] i.e.,

$$\mathbf{n}_t \sim \mathcal{CN}\left(\mathbf{0}, \sigma_y^2 \mathbf{I}_D, \mathbf{0}\right). \tag{9}$$

The hyperparameter σ_y^2 is the noise variance, $\mathbf{I}_D$ is the identity matrix of size D, and D is the number of receiving antennas, and hence the dimensionality of $\mathbf{y}_t$, $\mathbf{h}_m^\ell$ and $\mathbf{n}_t$. Consequently, the probability distribution over the received complex vector at time t is described by

$$\mathbf{y}_t \big| \left\{ \mathbf{h}_m^\ell \right\}, \mathbf{X} \sim \mathcal{CN}\left(\sum_{m=1}^{M_+} \sum_{\ell=1}^{L} \mathbf{h}_m^\ell x_{(t-\ell+1)m}, \sigma_y^2 \mathbf{I}_D, \mathbf{0} \right). \tag{10}$$

We finally place a circularly symmetric complex Gaussian prior over the channel coefficients, i.e.,

$$\mathbf{h}_m^\ell | \sigma_\ell^2 \sim \mathcal{CN}\left(\mathbf{0}, \sigma_\ell^2 \mathbf{I}_D, \mathbf{0}\right), \tag{11}$$

and an inverse-gamma hyper-prior over the variances σ_ℓ^2,

$$p\left(\sigma_\ell^2\right) = \frac{(v_\ell)^\tau}{\Gamma(\tau)} \left(\frac{1}{\sigma_\ell^2}\right)^{\tau+1} \exp\left\{-\frac{v_\ell}{\sigma_\ell^2}\right\}, \tag{12}$$

[1] The complex Gaussian distribution $\mathcal{CN}(\boldsymbol{\mu}, \boldsymbol{\Gamma}, \mathbf{C})$ over a vector $\mathbf{x}$ of length D is given by $p(\mathbf{x}) = \frac{1}{\pi^D \sqrt{\det(\boldsymbol{\Gamma}) \det(\mathbf{P})}}$
$\times \exp\left\{-\frac{1}{2}\left[(\mathbf{x}-\boldsymbol{\mu})^\mathsf{H}, (\mathbf{x}-\boldsymbol{\mu})^\top\right] \begin{bmatrix} \boldsymbol{\Gamma} & \mathbf{C} \\ \mathbf{C}^\mathsf{H} & \boldsymbol{\Gamma}^\star \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{x}-\boldsymbol{\mu} \\ (\mathbf{x}-\boldsymbol{\mu})^\star \end{bmatrix}\right\}$, where $\mathbf{P} = \boldsymbol{\Gamma}^\star - \mathbf{C}^\mathsf{H} \boldsymbol{\Gamma}^{-1} \mathbf{C}$, $(\cdot)^\star$ denotes the complex conjugate, and $(\cdot)^\mathsf{H}$ denotes the conjugate transpose. A circularly symmetric complex Gaussian distribution has $\boldsymbol{\mu} = \mathbf{0}$ and $\mathbf{C} = \mathbf{0}$.

where $v_\ell = (\tau - 1)\sigma_H^2 e^{-\lambda(\ell-1)}$ and $\tau = 2 + \kappa^{-2}$, being σ_H^2, λ and κ hyperparameters of the model. The mean and standard deviation of the variances σ_ℓ^2 are respectively given by $\mathbb{E}[\sigma_\ell^2] = \sigma_H^2 e^{-\lambda(\ell-1)}$ and $\mathrm{Std}[\sigma_\ell^2] = \kappa \mathbb{E}[\sigma_\ell^2]$. The choice of this particular prior is based on the assumption that the channel coefficients $\mathbf{h}_m^\ell$ are *a priori* expected to decay with the index ℓ, since they model the multipath effect. However, if the data contains enough evidence against this assumption, the posterior distribution will assign higher probability mass to larger values of σ_ℓ^2.

We depict the corresponding graphical model in Figure 4, with a value of $L = 2$.

IV. INFERENCE

One of the main challenges in Bayesian probabilistic models is posterior inference, which involves the computation of the posterior distribution over the hidden variables in the model given the data. In most models of interest, including BNP models, the posterior distribution cannot be obtained in closed form, and an approximate inference algorithm is used instead. In many BNP time series models, inference is carried out via MCMC methods [44]. In particular, for the IFHMM [34], typical approaches rely on a blocked Gibbs sampling algorithm that alternates between sampling the number of parallel chains and the global variables (emission parameters and transition probabilities) conditioned on the current value of matrices $\mathbf{S}$ and $\mathbf{X}$, and sampling matrices $\mathbf{S}$ and $\mathbf{X}$ conditioned on the current value of the global variables. We follow a similar algorithm, which proceeds iteratively as follows:

- **Step 1:** Add M_{new} new inactive chains[2] (FSMs) using an auxiliary slice variable and a slice sampling method. In this step, the number of considered parallel chains is increased from its initial value M_+ to $M^\ddagger = M_+ + M_{\text{new}}$ (we do not update M_+ because the new chains are in the all-zero state).
- **Step 2:** Sample the states s_{tm} and the input symbols x_{tm} of all the considered chains. Compact the representation by removing those FSMs that remain inactive in the entire observation period, consequently updating M_+.
- **Step 3:** Sample the global variables, i.e., the transition probabilities a^m and b^m and the observation parameters $\mathbf{h}_m^\ell$ for each active chain ($m = 1, \ldots, M_+$), as well as the variances σ_ℓ^2.

In **Step 1**, we follow the slice sampling scheme for inference in BNP models based on the IBP [34], [43], which effectively transforms the model into a finite factorial model with $M^\ddagger = M_+ + M_{\text{new}}$ parallel chains. We first sample an auxiliary slice variable ϑ, which is distributed as

$$\vartheta \,|\, \mathbf{S}, \left\{a^{(m)}\right\}_{m=1}^{M_+} \sim \mathrm{Uniform}(0, a_{\min}), \tag{13}$$

where $a_{\min} = \min_{m: \exists t, s_{tm} \neq 0} a^{(m)}$, and we can replace the uniform distribution with a more flexible scaled beta distribution. Then, starting from $m = M_+ + 1$, new variables $a^{(m)}$ are

[2] An inactive chain is a chain in which all elements $s_{tm} = 0$.

iteratively sampled from

$$
p\left(a^{(m)}|a^{(m-1)}\right) \propto \exp\left(\alpha \sum_{t=1}^{T} \frac{1}{t}\left(1 - a^{(m)}\right)^t\right)
$$
$$
\times \left(a^{(m)}\right)^{\alpha-1}\left(1 - a^{(m)}\right)^T \mathbb{I}\left(0 \le a^{(m)} \le a^{(m-1)}\right), \quad (14)
$$

with $a^{(M_+)} = a_{\min}$, until the resulting value is less than the slice variable, i.e., until $a^{(m)} < \vartheta$. Since Eq. (14) is log-concave in $\log a^{(m)}$ [43], we can apply adaptive rejection sampling (ARS) [45]. Let M_{new} be the number of new variables $a^{(m)}$ that are greater than the slice variable. If $M_{\text{new}} > 0$, then we expand the representation of matrices $\mathbf{S}$ and $\mathbf{X}$ by adding M_{new} zero columns, and we sample the corresponding per-chain global variables (i.e., $\mathbf{h}_m^\ell$ and b^m) from the prior, given in Eqs. (11) and (5), respectively.

Step 2 consists in sampling the elements of the matrices $\mathbf{S}$ and $\mathbf{X}$ given the current value of $M^{\ddagger}$ and the global variables. In this step, several approaches can be taken. A naïve Gibbs sampling algorithm that sequentially samples each element x_{tm} (jointly with s_{tm}) is simple and computationally efficient, but it presents poor mixing properties due to the strong couplings between successive time steps [34], [46]. An alternative to Gibbs sampling typically applied in factorial HMMs is blocked sampling, which sequentially samples each parallel chain, conditioned on the current value of the remaining ones. This approach requires a forward-filtering backward-sampling (FFBS) sweep in each of the chains, which yields runtime complexity of $\mathcal{O}(TM^{\ddagger}|\mathcal{X}|^{L+1})$ for our IFFSM model. The exponential dependency on L makes this method computationally intractable. In order to address this limitation of the FFBS approach, we propose to jointly sample matrices $\mathbf{S}$ and $\mathbf{X}$ using PGAS, an algorithm recently developed for inference in state-space models and non-Markovian latent variable models [31]. The runtime complexity of this algorithm scales as $\mathcal{O}(PTM^{\ddagger}L^2)$, where P is the number of particles used for the PGAS kernel. Details on the PGAS approach are given in Section IV-A.

After running PGAS, we remove those chains that remain inactive in the whole observation period. This implies removing some columns of $\mathbf{S}$ and $\mathbf{X}$ as well as the corresponding variables $\mathbf{h}_m^\ell$, a^m and b^m, and updating M_+.

In **Step 3**, we sample the global variables in the model from their complete conditional distributions.[3] The complete conditional distribution over the transition probabilities a^m under the semi-ordered stick-breaking construction [43] is

$$
a^m|\mathbf{S} \sim \text{Beta}\left(n_{01}^m, 1 + n_{00}^m\right), \quad (15)
$$

being n_{ij}^m the number of transitions from state i to state j in the m-th column of $\mathbf{S}$. For the self-transition probabilities of the active state b^m, we have

$$
b^m|\mathbf{S} \sim \text{Beta}\left(\beta_0 + n_{11}^m, \beta_1 + n_{10}^m\right). \quad (16)
$$

The complete conditional distributions over the emission parameters $\mathbf{h}_m^\ell$ for all chains $m = 1, \ldots, M_+$ and for all taps

---

[3]The complete conditional is the conditional distribution of a hidden variable, given the observations and the rest of hidden variables.

$\ell = 1, \ldots, L$ are given by complex Gaussians of the form

$$
\mathbf{h}^{(d)}|\mathbf{Y}, \mathbf{X}, \left\{\sigma_\ell^2\right\} \sim \mathcal{CN}\left(\boldsymbol{\mu}_{\text{POST}}^{(d)}, \boldsymbol{\Gamma}_{\text{POST}}, \mathbf{0}\right), \quad (17)
$$

for $d = 1, \ldots, D$, where $\mathbf{Y}$ is the $T \times D$ matrix containing all vectors $\mathbf{y}_t$. Here, we have defined for notational simplicity $\mathbf{h}^{(d)}$ as the vector that contains the d-th component of vectors $\mathbf{h}_m^\ell$ for all m and ℓ, as given by

$$
\mathbf{h}^{(d)} = \left[\left(\mathbf{h}_1^1\right)_d, \ldots, \left(\mathbf{h}_1^L\right)_d, \ldots, \left(\mathbf{h}_{M_+}^1\right)_d, \ldots, \left(\mathbf{h}_{M_+}^L\right)_d\right]^\top. \quad (18)
$$

We can write the parameters $\boldsymbol{\mu}_{\text{POST}}^{(d)}$ and $\boldsymbol{\Gamma}_{\text{POST}}$ as follows. We first define the extended matrix $\mathbf{X}^{\text{ext}} = [\mathbf{X}^{(1)}, \ldots, \mathbf{X}^{(M_+)}]$ of size $T \times LM_+$, with

$$
\mathbf{X}^{(m)} = \begin{bmatrix} x_{1m} & 0 & 0 & \cdots & 0 \\ x_{2m} & x_{1m} & 0 & \cdots & 0 \\ x_{3m} & x_{2m} & x_{1m} & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ x_{Tm} & x_{(T-1)m} & x_{(T-2)m} & \cdots & x_{(T-L+1)m} \end{bmatrix}, \quad (19)
$$

$\boldsymbol{\Sigma}$ as the $L \times L$ diagonal matrix containing all variables σ_ℓ^2, and $\mathbf{y}^{(d)}$ as the T-dimensional vector containing the d-th element of each observation $\mathbf{y}_t$. Then, the posterior parameters in Eq. (17) are given by

$$
\boldsymbol{\Gamma}_{\text{POST}} = \left(\left(\mathbf{I}_{M_+} \otimes \boldsymbol{\Sigma}\right)^{-1} + \frac{1}{\sigma_y^2}\left(\mathbf{X}^{\text{ext}}\right)^{\mathsf{H}}\mathbf{X}^{\text{ext}}\right)^{-1} \quad (20)
$$

and

$$
\boldsymbol{\mu}_{\text{POST}}^{(d)} = \frac{1}{\sigma_y^2}\boldsymbol{\Gamma}_{\text{POST}}\left(\mathbf{X}^{\text{ext}}\right)^{\mathsf{H}}\mathbf{y}^{(d)}, \quad (21)
$$

being $(\cdot)^{\mathsf{H}}$ the conjugate transpose, '$\otimes$' the Kronecker product, and $\mathbf{I}_{M_+}$ the identity matrix of size M_+.

Regarding the complete conditionals of the variances σ_ℓ^2, they are given by inverse-gamma distributions of the form

$$
p\left(\sigma_\ell^2 | \left\{\mathbf{h}_m^\ell\right\}_{m=1}^{M_+}\right)
$$
$$
\propto \left(\frac{1}{\sigma_\ell^2}\right)^{1+\tau+DM_+} \exp\left\{-\frac{\nu_\ell + \sum_{m=1}^{M_+}||\mathbf{h}_m^\ell||_2^2}{\sigma_\ell^2}\right\}, \quad (22)
$$

being $||\mathbf{h}_m^\ell||_2^2$ the squared L^2-norm of the vector $\mathbf{h}_m^\ell$.

A. Particle Gibbs With Ancestor Sampling

We rely on PGAS [31] for Step 2 of our inference algorithm, in order to obtain a sample of the matrices $\mathbf{S}$ and $\mathbf{X}$ (which at this stage of the inference algorithm are truncated to $M^{\ddagger}$ columns). PGAS is a method within the framework of PMCMC [32], which is a systematic way of combining sequential Monte Carlo (SMC) and MCMC to take advantage of the strengths of both techniques.

In PGAS, a Markov kernel is constructed by running an SMC sampler in which one particle is set deterministically to a reference input particle. This reference particle corresponds to the output of the previous PGAS iteration (extended to account for the M_{new} new FSMs). After a complete run of the SMC
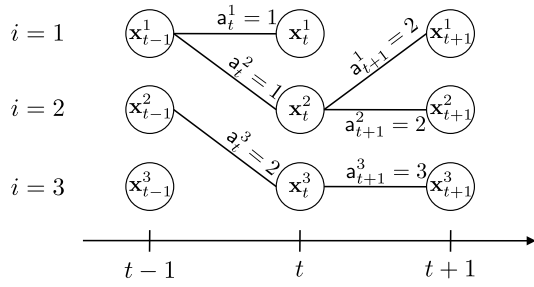
Fig. 5. Example of the connection of particles in PGAS. We represent $P = 3$ particles $\mathbf{x}_\tau^i$ for $\tau = \{t-1, t, t+1\}$. The index a_t^i denotes the ancestor particle of $\mathbf{x}_\tau^i$. It can be seen that, e.g., the trajectories $\mathbf{x}_{1:t+1}^1$ and $\mathbf{x}_{1:t+1}^2$ only differ at time instant $t+1$.

algorithm, a new reference trajectory is obtained by selecting one of the particle trajectories with probabilities given by their importance weights. In this way, the resulting Markov kernel leaves its target distribution invariant, regardless of the number of particles used. In contrast to other particle Gibbs with backward simulation methods [47], [48], PGAS can also be applied to non-Markovian latent variable models, i.e., models that are not expressed on a state-space form [31], [49]. In this section, we briefly describe the PGAS algorithm and provide the necessary equations for its implementation.

In PGAS, we assume a set of P particles for each time instant t, each representing the hidden states $\{x_{tm}\}_{m=1}^{M^\ddagger}$ (hence, they also represent $\{s_{tm}\}_{m=1}^{M^\ddagger}$). We denote the state of the i-th particle at time t by the vector $\mathbf{x}_t^i$ of length $M^\ddagger$. Similarly, the input reference particle for each time instant is denoted as $\mathbf{x}_t'$. We also introduce the ancestor indexes $\mathsf{a}_t^i \in \{1, \ldots, P\}$ in order to denote the particle that precedes the i-th particle at time t. That is, a_t^i corresponds to the index of the ancestor particle of $\mathbf{x}_t^i$. Let also $\mathbf{x}_{1:t}^i$ be the ancestral path of particle $\mathbf{x}_t^i$, i.e., the particle trajectory that is recursively defined as

$$\mathbf{x}_{1:t}^i = \left(\mathbf{x}_{1:t-1}^{\mathsf{a}_t^i}, \mathbf{x}_t^i \right). \tag{23}$$

Figure 5 shows an example to clarify the notation.

The machinery inside the PGAS algorithm resembles an ordinary particle filter, with two main differences: one of the particles is deterministically set to the reference input sample, and the ancestor of each particle is randomly chosen and stored during the algorithm execution. Algorithm 1 summarizes the procedure. For each time instant t, we first generate the ancestor indexes for the first $P - 1$ particles according to the importance weights w_{t-1}^i. Given these ancestors, the particles are then propagated across time according to a distribution $r_t(\mathbf{x}_t | \mathbf{x}_{1:t-1}^{\mathsf{a}_t})$. For simplicity, and dropping the global variables from the notation for conciseness, we assume that

$$r_t\left(\mathbf{x}_t | \mathbf{x}_{1:t-1}^{\mathsf{a}_t}\right) = p\left(\mathbf{x}_t | \mathbf{x}_{t-1}^{\mathsf{a}_t}\right)$$
$$= \prod_{m=1}^{M^\ddagger} p(x_{tm}|s_{tm}) p\left(s_{tm}|s_{(t-1)m}^{\mathsf{a}_t}\right), \tag{24}$$

i.e., particles are propagated as in Figure 3 using a simple bootstrap proposal kernel, $p(x_{tm}, s_{tm}|s_{(t-1)m}, x_{(t-1)m})$. The P-th particle is instead deterministically set to the reference particle, $\mathbf{x}_t^P = \mathbf{x}_t'$, whereas the ancestor indexes a_t^P are sampled

---

Algorithm 1 Particle Gibbs With Ancestor Sampling

Input: Reference particle $\mathbf{x}_t'$ for $t = 1, \ldots, T$, and global variables
Output: Sample $\mathbf{x}_{1:T}^{\text{out}}$ from the PGAS Markov kernel
1: Draw $\mathbf{x}_1^i \sim r_1(\mathbf{x}_1)$ for $i = 1, \ldots, P - 1$ (Eq. (24))
2: Set $\mathbf{x}_1^P = \mathbf{x}_1'$
3: Compute the weights w_1^i for $i = 1, \ldots, P$ (Eq. (25))
4: **for** $t = 2, \ldots, T$ **do**
5:   Draw $\mathsf{a}_t^i \sim$ Categorical$(w_{t-1}^1, \ldots, w_{t-1}^P)$ for $i = 1, \ldots, P - 1$
6:   Compute $\widetilde{w}_{t-1|T}^i$ for $i = 1, \ldots, P$ (Eq. (26))
7:   Draw $\mathsf{a}_t^P \sim$ Categorical$(\widetilde{w}_{t-1|T}^1, \ldots, \widetilde{w}_{t-1|T}^P)$
8:   Draw $\mathbf{x}_t^i \sim r_t(\mathbf{x}_t | \mathbf{x}_{1:t-1}^{\mathsf{a}_t^i})$ for $i = 1, \ldots, P - 1$ (Eq. (24))
9:   Set $\mathbf{x}_t^P = \mathbf{x}_t'$
10:   Set $\mathbf{x}_{1:t}^i = (\mathbf{x}_{1:t-1}^{\mathsf{a}_t^i}, \mathbf{x}_t^i)$ for $i = 1, \ldots, P$ (Eq. (23))
11:   Compute the weights w_t^i for $i = 1, \ldots, P$ (Eq. (25))
12: **end for**
13: Draw $k \sim$ Categorical$(w_T^1, \ldots, w_T^P)$
14: **return** $\mathbf{x}_{1:T}^{\text{out}} = \mathbf{x}_{1:T}^k$

---

according to some weights $\widetilde{w}_{t-1|T}^i$. Indeed, this is a crucial step that vastly improves the mixing properties of the MCMC kernel.

We now focus on the computation of the importance weights w_t^i and the ancestor weights $\widetilde{w}_{t-1|T}^i$. For the former, the particles are weighted according to

$$w_t^i = \frac{p(\mathbf{x}_{1:t}|\mathbf{y}_{1:t})}{p(\mathbf{x}_{1:t-1}|\mathbf{y}_{1:t-1}) r_t(\mathbf{x}_t|\mathbf{x}_{1:t-1})}$$
$$\propto \frac{p(\mathbf{y}_{1:t}|\mathbf{x}_{1:t}) p(\mathbf{x}_{1:t})}{p(\mathbf{y}_{1:t-1}|\mathbf{x}_{1:t-1}) p(\mathbf{x}_{1:t-1}) p(\mathbf{x}_t|\mathbf{x}_{t-1})}$$
$$= p(\mathbf{y}_t|\mathbf{x}_{t-L+1:t}), \tag{25}$$

being $\mathbf{y}_{\tau_1:\tau_2}$ the set of observations $\{\mathbf{y}_t\}_{t=\tau_1}^{\tau_2}$. We have applied (24) to derive this expression. Eq. (25) implies that, in order to obtain the importance weights, it suffices to evaluate the likelihood at time t.

The weights $\widetilde{w}_{t-1|T}^i$ used to draw a random ancestor for the reference particle are given by

$$\widetilde{w}_{t-1|T}^i = w_{t-1}^i \frac{p(\mathbf{x}_{1:t-1}^i, \mathbf{x}_{t:T}'|\mathbf{y}_{1:T})}{p(\mathbf{x}_{1:t-1}^i|\mathbf{y}_{1:t-1})}$$
$$\propto w_{t-1}^i \frac{p(\mathbf{y}_{1:T}|\mathbf{x}_{1:t-1}^i, \mathbf{x}_{t:T}') p(\mathbf{x}_{1:t-1}^i, \mathbf{x}_{t:T}')}{p(\mathbf{y}_{1:t-1}|\mathbf{x}_{1:t-1}^i) p(\mathbf{x}_{1:t-1}^i)}$$
$$\propto w_{t-1}^i p(\mathbf{x}_t'|\mathbf{x}_{t-1}^i) \prod_{\tau=t}^{t+L-2} p(\mathbf{y}_\tau|\mathbf{x}_{1:t-1}^i, \mathbf{x}_{t:T}'). \tag{26}$$

In order to obtain this expression, we have made use of the Markov property of the model, and we have also ignored factors that do not depend on the particle index i. Note that the transition probability $p(\mathbf{x}_t|\mathbf{x}_{t-1})$ factorizes across the parallel chains of the factorial model, as given in (24). We also note that, for memoryless models (i.e., $L = 1$), Eq. (26) can be simplified, since the product in the last term is not present and, therefore, $\widetilde{w}_{t-1|T}^i \propto w_{t-1}^i p(\mathbf{x}_t'|\mathbf{x}_{t-1}^i)$.

(a) Inferred M_+.

(b) Recovered transmitters.

(c) ADER.

(d) SER.

(e) MSE.

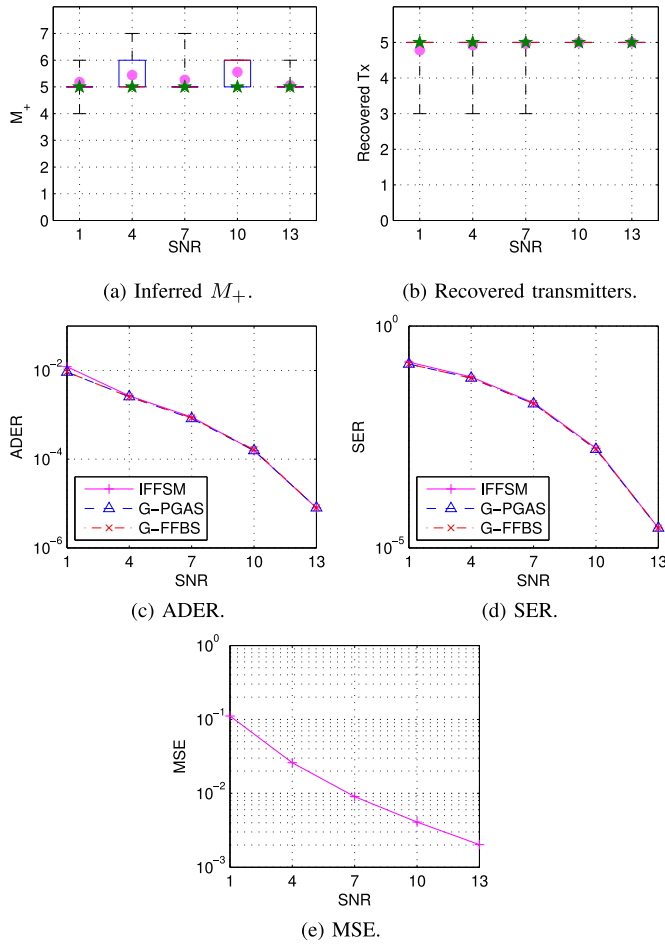Fig. 6.   Results for different SNRs ($L = 1$).

B. Computational Complexity

For any channel length L, the resulting complexity of each iteration of the algorithm scales as $\mathcal{O}(PTM^{\ddagger}L^2)$. This is because the most expensive step is the computation of the weights $\widetilde{w}_{t-1|T}^i$ in (26) for $i = 1, \ldots, P$, which has complexity scaling as $\mathcal{O}(PM^{\ddagger}L^2)$. This computation needs to be performed for each time instant $t = 1, \ldots, T$, and hence the resulting overall complexity.

V. EXPERIMENTS

In this section, we apply the IFFSM model in Section III to the problem of user activity detection and blind channel estimation. We use two different channel models, namely, a standard multipath Rayleigh fading channel model in Section V-A and a ray-tracing model in Section V-B.

A. Multipath Rayleigh Fading

In LTE, when the user equipment (UE) wants to transmit information, it needs to ask for resources in the random access channel (RACH) [50]. This is the first of four messages between the enhanced Node B (eNB), i.e., the network access point, and the UE before the actual transmission of information starts. The UE selects one of 64 random sequences with

839 symbols and uses 72 subcarriers (1.08 MHz) to transmit the sequence in 0.8 ms in a 1-ms sub-frame (0.2 ms are used for cyclic prefix and time guard band). In the typical configuration, there is a RACH subframe every 10-ms frame. If two UEs transmit the same preamble sequence at the same physical RACH, one of them (or both) would not get a response from the network and it (they) would need to resend the RACH preamble, although in most cases this channel goes unused.

In our simulations, we assume that instead of a RACH preamble asking for resources, our IoT devices send a 500-symbol sequence with the payload that they want to transmit for 0.5 ms, and the remaining 0.5 ms are reserved for time guard band. The timing of the IoT might not be as accurate as typical cellphones, and hence we allow for much larger guard band. We assume that the devices can start their transmission at any time in the first half of the physical RACH, and we also assume 5 active users in each physical RACH channel interval. We consider a Rayleigh AWGN channel, i.e., the channel coefficients and the noise are circularly-symmetric complex Gaussian distributed with zero mean, where the covariances matrices are $\sigma_{\ell}^2 \mathbf{I}$ and $\sigma_y^2 \mathbf{I}$, respectively. We simulate a base scenario with $D = 20$ receiving antennas, quadrature phase-shift keying (QPSK) modulation (the constellation is normalized to yield unit energy), and $\sigma_y^2 = 2$. Using this base configuration, we vary one of the parameters while holding the rest fixed.

We set the hyperparameters of the IFFSM as $\sigma_H^2 = 1$, $\lambda = 0.5$, $\kappa = 1$, $\alpha = 1$, $\beta_0 = 2$ and $\beta_1 = 0.1$. The choice of β_0 and β_1 is based on the fact that we expect the active Markov chains to remain active and, therefore, the transition probabilities from active to active b^m, which are Beta(β_0, β_1) distributed, are *a priori* expected to be of the order of one over a few hundred bits. In order to avoid getting trapped in local modes of the posterior, we use a tempering procedure. We first add artificial noise so that the resulting $\sigma_y^2 = 10^{1.2}$ and after each iteration we reduce this noise by a factor of 0.9995, until there is no artificial noise left. After that, we run additional iterations to favor exploitation.

Evaluation: For the recovered transmitters,[4] we evaluate the performance in terms of the activity detection error rate (ADER), the symbol error rate (SER) and the mean squared error (MSE) of the channel coefficient estimates. The ADER is the probability of detecting activity (inactivity) in a transmitter while that transmitter is actually inactive (active). When computing the SER, an error is computed at time t whenever the estimated symbol for a transmitter differs from the actual transmitted symbol, considering that the transmitted symbol while inactive is $x_{tm} = 0$. The MSE for each transmitter is
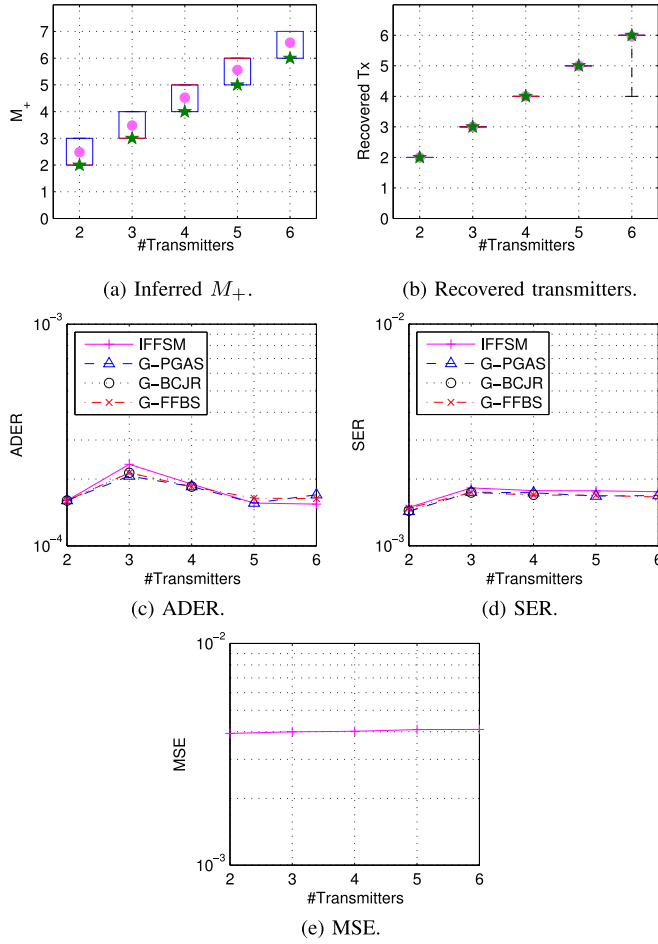
$$\text{MSE}_m = \frac{1}{LD} \sum_{d, \ell} \left\| \left(\mathbf{h}_m^{\ell} \right)_d - \left(\hat{\mathbf{h}}_m^{\ell} \right)_d \right\|^2, \qquad (27)$$

being $\hat{\mathbf{h}}_m^{\ell}$ the inferred channel coefficients.

We compare our approach (denoted by IFFSM in the plots) with three genie-aided methods which have perfect knowledge

---

[4]Our procedure is fully blind, so it might return spurious transmitters or it might mix two or more transmitters in a single chain. We discard those chains in our evaluations.
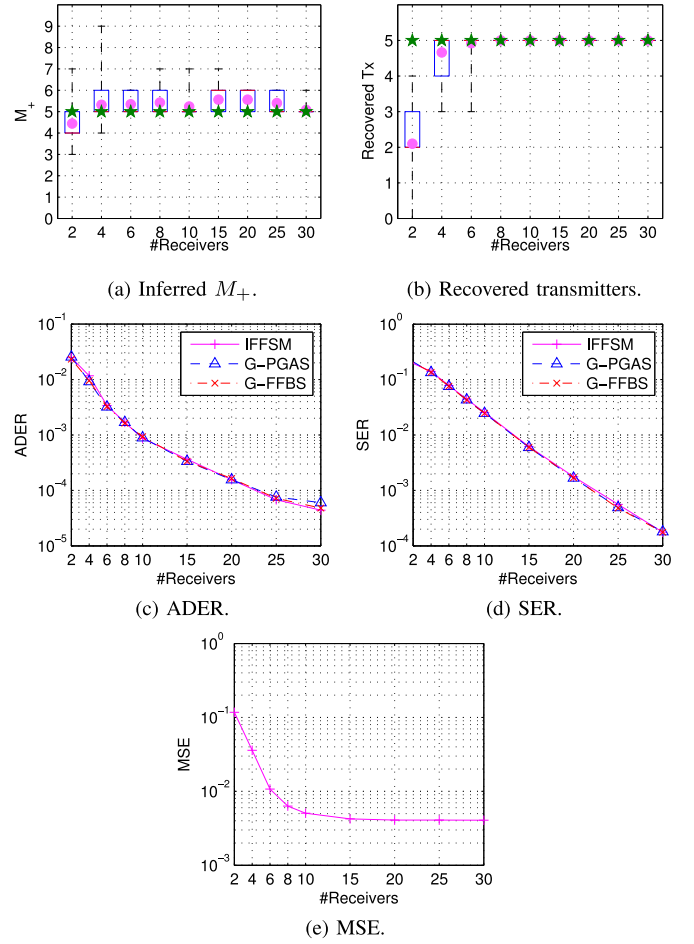
(a) Inferred M_+.

(b) Recovered transmitters.

(c) ADER.

(d) SER.

(e) MSE.

Fig. 7. Results for different number of transmitters ($L = 1$).

(a) Inferred M_+.

(b) Recovered transmitters.

(c) ADER.

(d) SER.

(e) MSE.

Fig. 8. Results for different number of receiving antennas ($L = 1$).

of the true number of transmitters and channel coefficients.[5] In particular, we run: (i) The PGAS algorithm that we use in Step 2 of our inference algorithm (G-PGAS), (ii) the FFBS algorithm over the equivalent factorial HMM with state space cardinality $|\mathcal{A} \bigcup \{0\}|^L$ (G-FFBS), and (iii) the optimum BCJR algorithm [51], over an equivalent single HMM with a number of states equal to $|\mathcal{A} \bigcup \{0\}|^{LN_t}$, being N_t the true number of transmitters (G-BCJR). Due to their complexity, we only run the BCJR algorithm in scenarios with $|\mathcal{A} \bigcup \{0\}|^{2LN_t} \leq 10^6$, and the FFBS in scenarios with $|\mathcal{A} \bigcup \{0\}|^{2L} \leq 10^6$.

For each considered scenario, we run 50 independent simulations, each with different simulated data. We run 20000 iterations of our inference algorithm, and obtain the inferred symbols $\hat{x}_{tm}$ as the component-wise *maximum a posteriori* (MAP) solution over the last 2000 samples. The estimates of the channel coefficients $\hat{\mathbf{h}}_m^\ell$ are then obtained as the MAP solution, conditioned on the data and the inferred symbols $\hat{x}_{tm}$. For the BCJR algorithm, we obtain the symbol estimates according to the component-wise MAP solution for each transmitter m and each instant t. For the genie-aided PGAS and FFBS methods, we follow a similar approach by running the algorithms for 10000 iterations and considering the last 2000 samples to
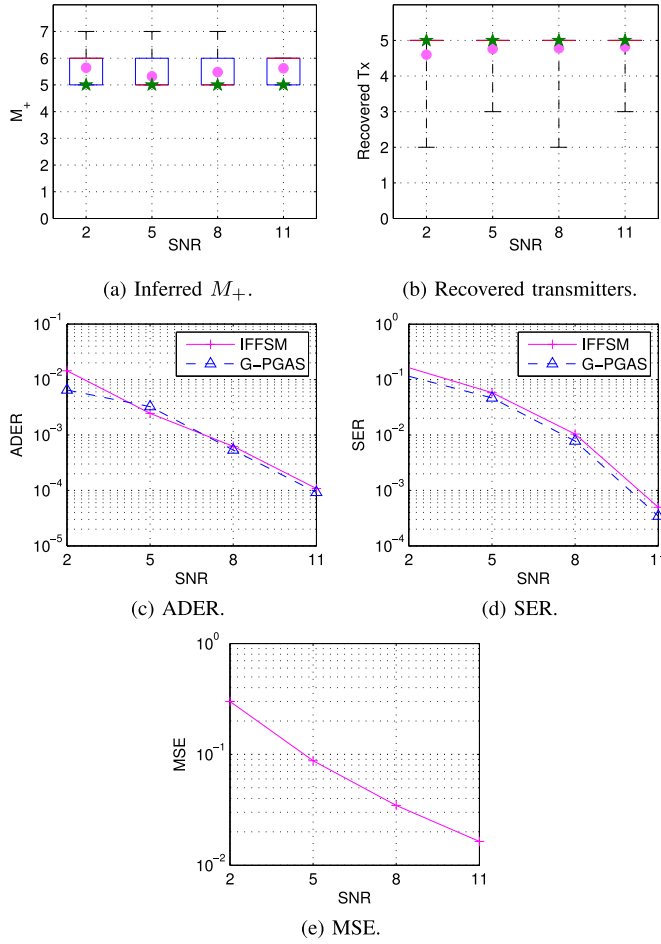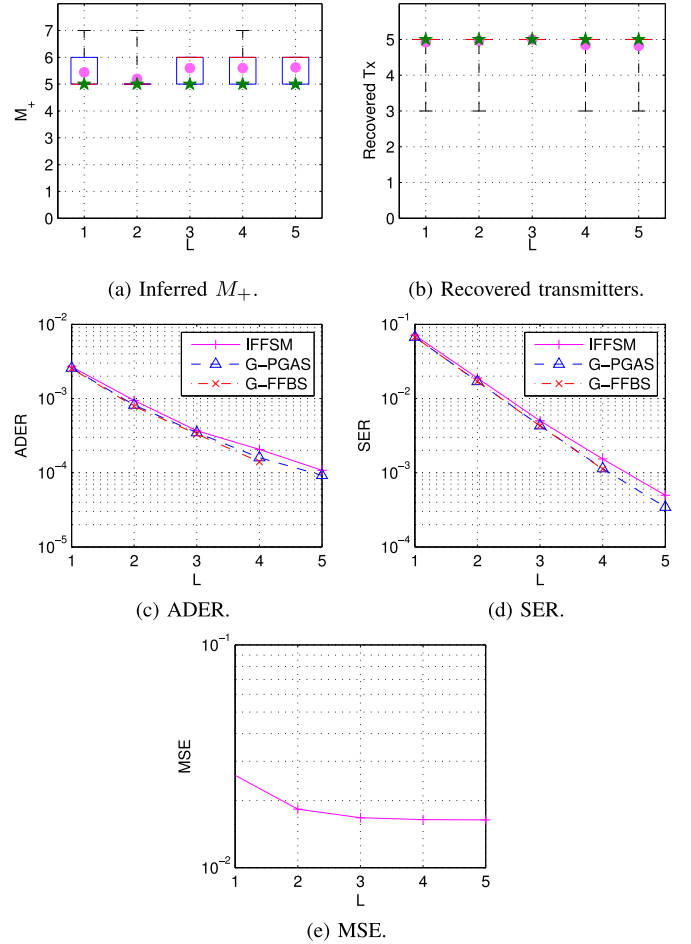
obtain the symbol estimates. Unless otherwise specified, we use $P = 3000$ particles for PGAS.

Results with perfect knowledge of the channel memory: We first evaluate the performance of our model and inference procedure assuming that the memory of the channel is accurately known. We initially consider memoryless channels, i.e., with memory length $L = 1$. Figure 6 shows the results when the noise variance varies from $\sigma_y^2 = 10^{1.2}$ to $\sigma_y^2 = 1$ (SNR from 1dB to 13dB).[6] Specifically, we first show a box-plot representation[7] of the inferred number of transmitters M_+, and also a box-plot representation of the number of recovered transmitters (i.e., how many of the true transmitters we are able to recover). For the recovered transmitters, we additionally show the ADER, the SER, and the MSE. As expected, the performance improves with the SNR. For low values of the SNR, transmitters are more likely to be masked by the noise and, therefore, on average we recover a slightly lower number of transmitters. We also observe that the performance (in terms of ADER and SER) of the proposed IFHMM is very similar

---

[5]For the genie-aided methods, we use $a^m = 0.998$ and $b^m = 0.002$.

[6]We obtain the per-user SNR (in dB) as $10 \log_{10}\left(\frac{DL}{\sigma_y^2}\right)$.

[7]We depict the 25-th, 50-th and 75-th percentiles in the standard format, as well as the most extreme values. Moreover, the mean value is represented with a pink circle, and the true number of transmitters is represented with a green star.

(a) Inferred M_+.

(b) Recovered transmitters.

(c) ADER.

(d) SER.

(e) MSE.

Fig. 9.   Results for different SNRs ($L = 5$).

(a) Inferred M_+.

(b) Recovered transmitters.

(c) ADER.

(d) SER.

(e) MSE.

Fig. 10.   Results for different values of L.

to the genie-aided methods, which have perfect knowledge of the number of transmitters and channel coefficients.

Figure 7 shows the results when the true number of transmitters ranges from 2 to 6. Although the number of parameters to be estimated grows with the number of transmitters, we observe that the performance is approximately constant. The IFHMM recovers all the transmitters in nearly all the simulations, with performance similar to the genie-aided methods.

Figure 8 shows the results when the number of receiving antennas varies from 2 to 30. In this figure, we observe that we need at least 6 receivers in order to properly recover the messages of all the transmitters. We see no further improvement in detectability with more than 8 receivers. As expected, the performance in terms of ADER and SER improves when the number of receiving antennas increases, as the diversity in the observations helps to recover the transmitted symbols and channel coefficients. This behavior is similar to the obtained by the genie-aided PGAS and FFBS, as shown in this figure. Note that the MSE curve flattens after approximately 15 receivers, as it reaches the threshold imposed by the noise level.

We now evaluate the performance of our model and inference procedure for frequency-selective channels, i.e.,

considering $L > 1$. Figure 9 shows the results when the noise variance varies from $\sigma_y^2 = 10^{1.8}$ to $\sigma_y^2 = 10^{0.9}$ (SNR from 2dB to 11dB), considering $L = 5$ to generate the data. We use the true value of the channel length L for inference. In the figure, we show the ADER, the SER, the MSE, a box-plot representation of the inferred number of transmitters M_+, and also a box-plot representation of the number of recovered transmitters. As in the memoryless case, the performance improves with the SNR. However, the values of the resulting SER and ADER are much lower than in the memoryless case for a fixed value of the noise variance. For instance, the SER in Figure 6 for $\sigma_y^2 = 10^{1.2}$ (SNR=1dB) is one order of magnitude above the reported SER in Figure 9 for the same value of the noise variance (SNR=8dB). This is a sensible result, because the channel memory adds more redundancy in the observed sequence, and therefore the resulting SNR is higher for a fixed value of the noise variance. Our inference algorithm is able to exploit such redundancy to better estimate the transmitted symbols, despite the fact that more channel coefficients need to be estimated. Note that the performance in terms of ADER and SER is similar to the genie-aided PGAS-based method.

In Figure 10, we show the obtained results for different values of the parameter L, ranging from 1 to 5. We use the true value of the channel length L for inference, and we
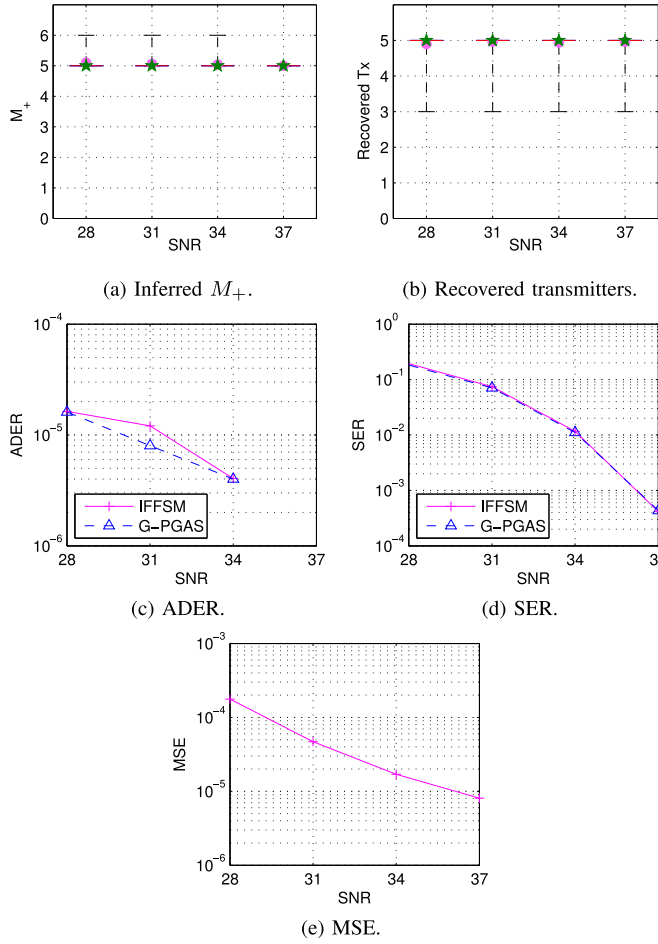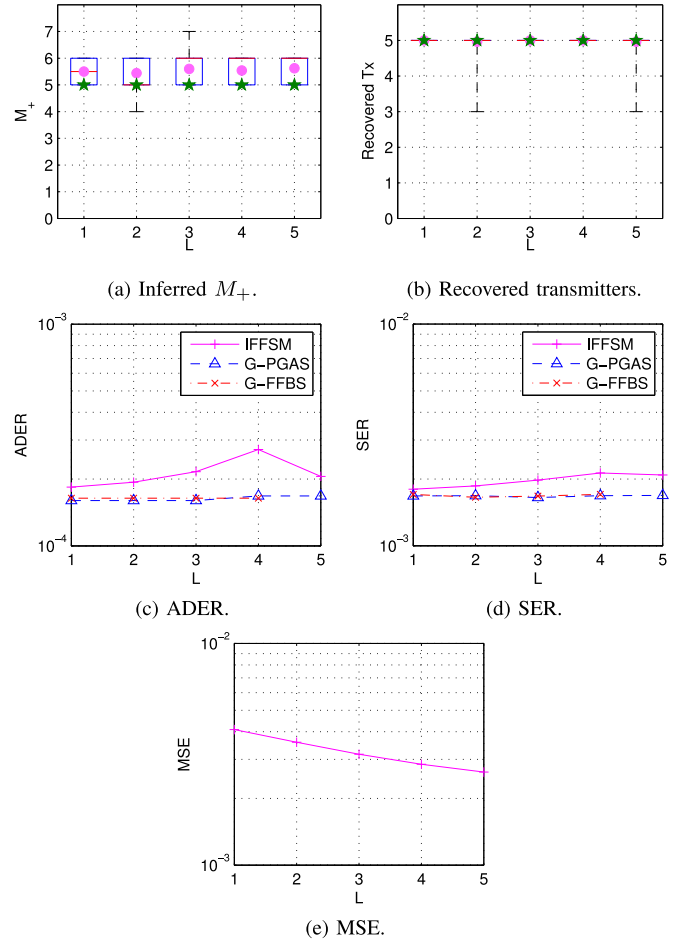
(a) Inferred M_+.

(b) Recovered transmitters.

(c) ADER.

(d) SER.

(e) MSE.

Fig. 11.   Results for different SNRs using a 1024-QAM constellation.

(a) Inferred M_+.

(b) Recovered transmitters.

(c) ADER.

(d) SER.

(e) MSE.

Fig. 12.   Results for different channel lengths ($L_{\text{true}} = 1$).

consider $\sigma_y^2 = 10^{0.9}$ in these experiments. The figure shows the ADER, the SER, the MSE, and box-plot representations of the inferred number of transmitters M_+ and the number of recovered transmitters. Here, it becomes clear that our model can exploit the redundancy introduced by the channel memory, as the performance in terms of SER and ADER improves as L increases. The MSE also improves with L, although it reaches a constant value for $L > 3$, similarly to the experiments in which we increase the number of receivers (although differently, in both cases we add redundancy to the observations). We can also observe that the performance is similar to the genie-aided methods (we do not run the FFBS algorithm for $L = 5$ due to its computational complexity).

Finally, we run an experiment with a different constellation order. Specifically, we use a 1024-QAM constellation, normalized to yield unit energy, and we vary the noise variance from $\sigma_y^2 = 10^{1.5}$ to $\sigma_y^2 = 10^{2.4}$ (SNR from 28dB to 37dB). In order to improve the mixing of the sampling procedure, and due to the high cardinality of $\mathcal{A}$, after running 5000 iterations of the inference algorithm, we run 20000 additional iterations in which we sequentially sample each hidden chain conditioned on the current value of the remaining ones, similarly to the standard FFBS algorithm for factorial models. We proceed similarly for the genie-aided PGAS method. We do not run the genie-aided FFBS or BCJR methods due to their high computational complexity in this scenario. We plot in Figure 11 the ADER, SER, and MSE, as well as a box-plot representation of the inferred number of transmitters M_+ and the number of recovered transmitters. We observe that, for the considered values of the SNR, we can recover the five transmitters in nearly all the cases. Furthermore, the SER curve of our approach closely follows the curve of the genie-aided method with perfect knowledge of the channel coefficients and the number of transmitters.

Results for mismatched channel length: In the experiments above, we have assumed that the true value of the channel length is known at the receiver side. As this scenario might not be realistic, we now run an experiment to show that we can properly estimate the transmitted symbols and the channel coefficients as long as our inference algorithm considers a sufficiently large value of L. For this purpose, we use our base experimental setup and generate data using $L = 1$ (i.e., memoryless channel). However, we use different values for the channel length L for inference.

In Figure 12, we show the obtained results for L ranging from 1 to 5. The obtained ADER and SER do not significantly degrade with increasing values of L, and we are able to recover the five transmitters in nearly all the cases. Interestingly, the
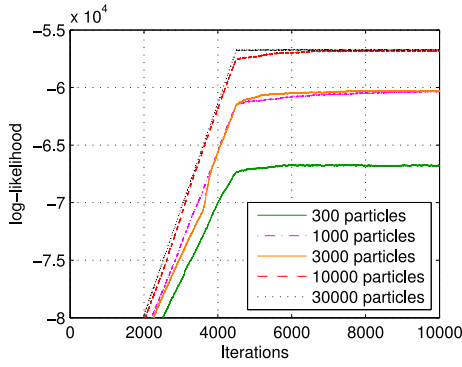
Fig. 13. Log-likelihood for varying number of particles ($L = 1$). The initial slope is due to the tempering procedure, in which we linearly increase the SNR at each iteration.

MSE improves as L increases. This is a consequence of the way we measure it when L is larger than the ground truth, as we compare our channel estimates with zero. The fact that the MSE becomes lower indicates that we obtain better estimates for the zero coefficients than for the non-zero ones, which in turn implies that our inference algorithm can properly decrease the channel variances σ_ℓ^2 when needed.

Sensitivity to the number of particles: Finally, we evaluate the impact of the number of particles in the PGAS kernel on the performance of the proposed algorithm. Note that, as the effective dimensionality of the hidden space increases, we should expect a larger number of particles to be required in order to properly estimate the transmitted symbols. To see this, we design an experiment with 10 transmitters and $\sigma_y^2 = 10^{0.3}$. Figure 13 shows the log-likelihood trace plot for 10000 iterations of the inference algorithm, with a number of particles ranging from 300 to 30000. This experiment is based on a single run of the algorithm, as it is enough to understand its behavior. It can be seen that the best performance is achieved with the largest number of considered particles. Additionally, this plot suggests that $P = 10000$ particles are enough for this scenario.

We also show in Figure 14 the number of inferred transmitters M_+, as well as the number of recovered transmitters, for each value of P. In this figure, we represent with a green star the true number of transmitters (again, these results are obtained after a single run of the algorithm). Although we infer $M_+ = 10$ transmitters with only $P = 3000$ particles, Figure 14b shows that we only recover 8 of them. The other two transmitters have been mixed in the last two inferred chains. In agreement with Figure 13, increasing the number of particles from $P = 10000$ to 30000 does not seem to improve performance: in both cases our algorithm is able to recover all the transmitters. Furthermore, even the genie-aided PGAS algorithm, which has perfect knowledge of the channel coefficients, needs a large value of P (above 3000) in order to recover all the transmitters.

We can conclude from these plots that we should adjust the number of particles based on the number of transmitters. However, the number of transmitters is an unknown quantity that we need to infer. There are two ways to overcome this apparent limitation. A sensible solution is to adaptively select
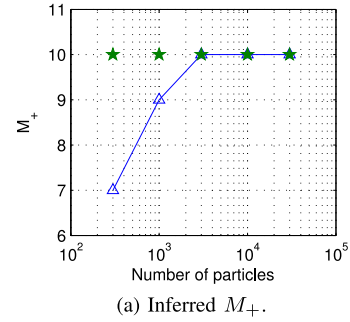
(a) Inferred M_+.
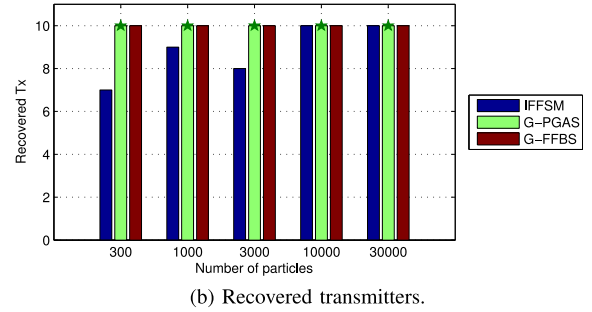
(b) Recovered transmitters.

Fig. 14. Number of inferred and recovered transmitters for varying number of particles ($L = 1$). The green star indicates the ground truth.
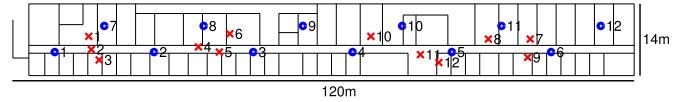

Fig. 15. Plane of the considered office building. Circles represent receivers, and crosses represent transmitters. All transmitters and receivers are placed at a height of 2 metres.

the number of particles P as a function of the current number of active transmitters, M_+. In other words, as we gather evidence for the presence of more transmitters, we consequently increase P. A second approach, which is computationally less demanding but may present poorer mixing properties, consists in running the PGAS inference algorithm sequentially over each chain, conditioned on the current value of the remaining transmitters, similarly to the standard FFBS procedure for the IFHMM [34]. Alternatively, we can apply the PGAS algorithm over fixed-sized blocks of randomly chosen transmitters.

B. Ray-Tracing Model

With the aim of considering a more realistic communication scenario, we use WISE software [52] to design an indoor wireless system. This software tool, developed at Bell Laboratories, includes a 3D ray-tracing propagation model, as well as algorithms for computational geometry and optimization, to calculate measures of radio-signal performance in user-specified regions. Its predictions have been validated with physical measurements.

Using WISE software and the map of an office located at Bell Labs Crawford Hill, we place 12 receivers and 12 transmitters across the office, intentionally placing the transmitters together in order to ensure that interferences occur in the nearby receivers. Figure 15 shows the considered map.

TABLE I
RESULTS FOR THE WI-FI EXPERIMENT

Model	L				
	1	2	3	4	5
IFFSM	12/13	12/12	12/12	12/12	12/12
IFHMM	6/31	6/20	6/21	0/57	–

(a) # Recovered transmitters / Inferred M_+.

Model	L				
	1	2	3	4	5
IFFSM	8.13	8.09	1.39	0.37	0.25
IFHMM	8.20	4.11	0.84	–	–

(b) MSE of the channel coefficients ($\times 10^{-5}$).

We consider a Wi-Fi transmission system with a bandwidth of 20 MHz or, equivalently, 50 ns per channel tap. We simulate the transmission of 1000-symbol bursts over this communication system, using a QPSK constellation normalized to yield unit energy. We scale the channel coefficients by a factor of 100, and we consequently scale the noise variance by 10^4, yielding $\sigma_y^2 \approx 7.96 \times 10^{-9}$. We set the transmission power to 0 dBm. Each transmitter becomes active at a random point, uniformly sampled in the interval $[1, T/2]$. We set $T = 2000$ time instants, so the burst duration of 1000 symbols ensures overlapping among all the transmitted signals.

Wi-Fi systems are not limited by the noise level, which is typically small enough, but by the users' interferences, which can be avoided by using a particular frequency channel for each user. Our goal is to show that cooperation of receivers in a Wi-Fi communication system can help recover the symbols transmitted by several users even when they simultaneously transmit over the same frequency channel, therefore allowing for a larger number of users in the system.

In our experiments, we vary L from 1 to 5. Five channel taps correspond to the radio signal travelling a distance of 750 m, which should be enough given the dimensions of this office space (the signal suffers attenuation when it reflects off the walls, so we should expect it to be negligible in comparison to the line-of-sight ray after a 750-m travelling distance). Following the tempering procedure described above, we initialize the algorithm with $\sigma_y^2 \approx 15.85$ and we linearly increase the SNR for around 26600 iterations, running 3400 additional iterations afterwards. We compare our IFFSM with a nonbinary IFHMM model with state space cardinality $|\mathcal{A} \bigcup \{0\}|^L$ using FFBS sweeps for inference (we do not run the FFBS algorithm for $L = 5$ due to its computational complexity). In this case, we set the hyperparameters as $\sigma_H^2 = 0.01$, $\lambda = 0.5$, $\kappa = 1$, $\alpha = 1$, $\beta_0 = 2$ and $\beta_1 = 0.1$.

We show in Table Ia the number of transmitters recovered after running the two inference algorithms, together with the inferred value of M_+, averaged for the last 2000 iterations. We see that the IFFSM recovers the 12 transmitters in all cases, and it does not tend to create spurious chains. In contrast, the IFHMM significantly overestimates the number of transmitters, which deteriorates the overall symbol estimates and, as a consequence, not all the transmitted symbols are recovered. In the best case, the IFHMM only recovers 6 out of the 12 transmitters. In the extreme case of $L = 4$, the inference algorithm for the IFHMM estimates 57 chains, but none of them

corresponds to a true transmitter; rather, each transmitter is explained by a combination of several chains.

For completeness, we additionally report in Table Ib the MSE of the channel coefficients, averaged for the last 2000 iterations. We sort the transmitters so that the MSE is minimized, and ignore the extra inferred transmitters. As expected, the MSE decreases as we consider a larger value of L, since the model better fits the true radio propagation model.

VI. CONCLUSION AND DISCUSSION

In this paper, we have proposed a fully blind approach for joint channel estimation and detection of the transmitted data when the number of transmitters is unknown. Our approach is based on a BNP model, which we refer to as IFFSM, and considers a potentially unbounded number of stochastic finite-memory FSMs that evolve independently over time. Our model and the PGAS-based inference algorithm allow us to readily account for frequency-selective channels, avoiding the exponential complexity with respect to the channel length of previous approaches. We have evaluated the performance of our approach through a comprehensive experimental design using both synthetic and real data.

These results are promising for the suitability of BNPs applied to signal processing for communications. As a result, this paper opens several further research lines on this area. Some examples of such possible future research lines are:

- Improving the scalability of the inference algorithm. This would allow us to account for both a larger number of transmitters and larger observation sequences at a sublinear computational cost. We believe that further investigations on faster approximate inference algorithms would be of great interest and importance for the development of practical receivers that operate in real-time.

- An online inference algorithm. In many cases, the receiver does not have access to a fixed window of observations, but data arrives instead as a never-ending stream. Thus, an online inference algorithm instead of a batch one is also of potential interest.

- Modelling of time-varying channels. Our current approach is restricted to static channels, i.e., the channel coefficients do not vary over time. A potentially useful research line may consist in taking into account the temporal evolution of the channel coefficients.

- An extension of the model that accounts for coding schemes. A channel coding scheme can be used in order to add redundancy to the user's data, effectively decreasing the resulting bit error probability. This redundancy can potentially be included into the model and exploited by the inference algorithm.

- An extension for sparse code multiple access (SCMA). Our approach can also be adapted to the particularities of the uplink SCMA techniques [53], [54], which use non-orthogonal transmission signals in the random access channel to reduce latency. The signal structure and the sparsity of the code/pilot scheme may simplify the inference procedure of the Bayesian model, at the cost of an upper bound on the number of potential users in the

network. This would result in a model that does not need to be nonparametric, but our joint approach may still reduce the errors of the current several-stage detectors.

An extension of our IFFSM, less related to the aforementioned research lines, consists in a semi-Markov approach. In this way, transmitters are assumed to send only a burst of symbols during the observation period, and a changepoint detection algorithm may detect the (de)activation instants.
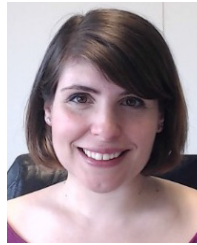
REFERENCES

[1] "Small cell market status," Informa Telecoms and Media, London, U.K., Rep., Dec. 2012.

[2] "M2M white paper: The growth of device connectivity," FocalPoint Group, Ellenton, FL, USA, Rep., 2003. [Online]. Available: goo.gl/S0j5O

[3] "Global machine to machine communication," Vodafone, London, U.K., Rep., 2010. [Online]. Available: goo.gl/4V8az

[4] "Device connectivity unlocks value," Ericsson, Stockholm, Sweden, Rep., Jan. 2011. [Online]. Available: goo.gl/alou6

[5] S.-Y. Lien, K.-C. Chen, and Y. Lin, "Toward ubiquitous massive accesses in 3GPP machine-to-machine communications," *IEEE Commun. Mag.*, vol. 49, no. 4, pp. 66–74, Apr. 2011.

[6] H. S. Dhillon, H. Huang, H. Viswanathan, and R. A. Valenzuela, "Fundamentals of throughput maximization with random arrivals for M2M communications," *IEEE Trans. Commun.*, vol. 62, no. 11, pp. 4094–4109, Nov. 2014.

[7] Y. Chen and W. Wang, "Machine-to-machine communication in LTE-A," in *Proc. IEEE Veh. Technol. Conf.*, Ottawa, ON, Canada, Sep. 2010, pp. 1–4.

[8] H. S. Dhillon, H. C. Huang, H. Viswanathan, and R. A. Valenzuela, "Power-efficient system design for cellular-based machine-to-machine communications," *IEEE Trans. Wireless Commun.*, vol. 12, no. 11, pp. 5740–5753, Nov. 2013.

[9] C.-Y. Tu, C.-Y. Ho, and C.-Y. Huang, "Energy-efficient algorithms and evaluations for massive access management in cellular based machine to machine communications," in *Proc. IEEE Veh. Technol. Conf.*, Sep. 2011, pp. 1–5.

[10] C.-Y. Ho and C.-Y. Huang, "Energy-saving massive access control and resource allocation schemes for M2M communications in OFDMA cellular networks," *IEEE Wireless Commun. Lett.*, vol. 1, no. 3, pp. 209–212, Jun. 2012.

[11] J. Hoydis, S. T. Brink, and M. Debbah, "Massive MIMO in the UL/DL of cellular networks: How many antennas do we need?" *IEEE J. Sel. Areas Commun.*, vol. 31, no. 2, pp. 160–171, Feb. 2013.

[12] L. Lu, G. Y. Li, A. L. Swindlehurst, A. Ashikhmin, and R. Zhang, "An overview of massive MIMO: Benefits and challenges," *IEEE J. Sel. Topics Signal Process.*, vol. 8, no. 5, pp. 742–758, Oct. 2014.

[13] M. A. Vázquez and J. Míguez, "User activity tracking in DS-CDMA systems," *IEEE Trans. Veh. Technol.*, vol. 62, no. 7, pp. 3188–3203, Sep. 2013.

[14] S. Verdu, *Multiuser Detection*. Cambridge, U.K.: Cambridge Univ. Press, 1998.

[15] M. Jiang and L. Hanzo, "Multiuser MIMO-OFDM for next-generation wireless systems," *Proc. IEEE*, vol. 95, no. 7, pp. 1430–1469, Jul. 2007.

[16] W.-C. Wu and K.-C. Chen, "Identification of active users in synchronous CDMA multiuser detection," *IEEE J. Sel. Areas Commun.*, vol. 16, no. 9, pp. 1723–1735, Dec. 1998.

[17] P. Pad, A. Mousavi, A. Goli, and F. Marvasti, "Simplified MAP-MUD for active user CDMA," *IEEE Commun. Lett.*, vol. 15, no. 6, pp. 599–601, Jun. 2011.

[18] Z. Pi and U. Mitra, "On blind timing acquisition and channel estimation for wideband multiuser DS-CDMA systems," *J. VLSI Signal Process. Syst. Signal Image Video Technol.*, vol. 30, nos. 1–3, pp. 127–142, 2002.

[19] S. Buzzi, L. Venturino, A. Zappone, and A. De Maio, "Blind user detection in doubly dispersive DS/CDMA fading channels," *IEEE Trans. Signal Process.*, vol. 58, no. 3, pp. 1446–1451, Mar. 2010.

[20] A. Mousavi, P. Pad, P. Delgosha, and F. Marvasti, "A new decoding scheme for errorless codes for overloaded CDMA with active user detection," in *Proc. 18th Int. Conf. Telecommun.*, May 2011, pp. 201–205.

[21] K. W. Halford and M. Brandt-Pearce, "New-user identification in a CDMA system," *IEEE Trans. Commun.*, vol. 46, no. 1, pp. 144–155, Jan. 1998.

[22] A. Angelosante, E. Biglieri, and M. Lops, "Multiuser detection in a dynamic environment—Part II: Joint user identification and parameter estimation," *IEEE Trans. Inf. Theory*, vol. 55, no. 5, pp. 2365–2374, May 2009.

[23] H. Zhu and G. B. Giannakis, "Exploiting sparse user activity in multiuser detection," *IEEE Trans. Commun.*, vol. 59, no. 2, pp. 454–465, Feb. 2011.

[24] D. Angelosante, E. Grossi, G. B. Giannakis, and M. Lops, "Sparsity-aware estimation of CDMA system parameters," in *Proc. IEEE 10th Workshop Signal Process. Adv. Wireless Commun.*, Perugia, Italy, Jun. 2009, pp. 697–701.

[25] D. Angelosante, E. Biglieri, and M. Lops, "Low-complexity receivers for multiuser detection with an unknown number of active users," *Signal Process.*, vol. 90, no. 5, pp. 1486–1495, May 2010.

[26] I. Valera, F. J. R. Ruiz, and F. Perez-Cruz, "Infinite factorial unbounded hidden Markov model for blind multiuser channel estimation," in *Proc. 4th Int. Workshop Cogn. Inf. Process.*, Copenhagen, Denmark, May 2014, pp. 1–6.

[27] I. Valera, F. J. R. Ruiz, and F. Perez-Cruz, "Infinite factorial unbounded-state hidden Markov model," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 9, pp. 1816–1828, Sep. 2015.

[28] I. Valera, F. J. R. Ruiz, L. Svensson, and F. Perez-Cruz, "A Bayesian nonparametric approach for blind multiuser channel estimation," in *Proc. Eur. Signal Process. Conf.*, Nice, France, 2015, pp. 2766–2770.

[29] M. S. Ali, E. Hossain, and D. I. Kim, "LTE/LTE-A random access for massive machine-type communications in smart cities," *IEEE Commun. Mag.*, vol. 55, no. 1, pp. 76–83, Jan. 2017.

[30] M. Polese, M. Centenaro, A. Zanella, and M. Zorzi, "M2M massive access in LTE: RACH performance evaluation in a smart city scenario," in *Proc. Int. Conf. Commun.*, Kuala Lumpur, Malaysia, May 2016, pp. 1–6.

[31] F. Lindsten, M. I. Jordan, and T. B. Schön, "Particle Gibbs with ancestor sampling," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 2145–2184, 2014.

[32] C. Andrieu, A. Doucet, and R. Holenstein, "Particle Markov chain Monte Carlo methods," *J. Roy. Stat. Soc. B Stat. Methodol.*, vol. 72, no. 3, pp. 269–342, 2010.

[33] Z. Ghahramani and M. I. Jordan, "Factorial hidden Markov models," *Mach. Learn.*, vol. 29, nos. 2–3, pp. 245–273, 1997.

[34] J. Van Gael, Y. W. Teh, and Z. Ghahramani, "The infinite factorial hidden Markov model," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 21. Vancouver, BC, Canada, 2009, pp. 1697–1704.

[35] J. E. Hopcroft, R. Motwani, and J. D. Ullman, *Introduction to Automata Theory, Languages, and Computation*, 3rd ed. Boston, MA, USA: Addison-Wesley, 2006.

[36] J. A. Anderson and T. J. Head, *Automata Theory With Modern Applications*. Cambridge, U.K.: Cambridge Univ. Press, 2006.

[37] J. Wang, *Handbook of Finite State Based Models and Applications*, 1st ed. Boca Raton, FL, USA: CRC Press, 2012.

[38] L. E. Baum and T. Petrie, "Statistical inference for probabilistic functions of finite state Markov chains," *Ann. Math. Stat.*, vol. 37, no. 6, pp. 1554–1563, 1966.

[39] L. Rabiner and B. Juang, "An introduction to hidden Markov models," *IEEE ASSP Mag.*, vol. 3, no. 1, pp. 4–16, Jan. 1986.

[40] M. I. Jordan, *Hierarchical Models, Nested Models and Completely Random Measures*, M.-H. Chen, D. Dey, P. Mueller, D. Sun, and K. Ye, Eds. New York, NY, USA: Springer, 2010.

[41] P. Orbanz and Y. W. Teh, "Bayesian nonparametric models," in *Encyclopedia of Machine Learning*. Boston, MA, USA: Springer, 2010.

[42] T. L. Griffiths and Z. Ghahramani, "The Indian buffet process: An introduction and review," *J. Mach. Learn. Res.*, vol. 12, pp. 1185–1224, Apr. 2011.

[43] Y. W. Teh, D. Görür, and Z. Ghahramani, "Stick-breaking construction for the Indian buffet process," in *Proc. Int. Conf. Artif. Intell. Stat.*, vol. 11, 2007, pp. 556–563.

[44] C. P. Robert and G. Casella, *Monte Carlo Statistical Methods* (Springer Texts in Statistics). New York, NY, USA: Springer-Verlag, 2005.

[45] W. R. Gilks and P. Wild, "Adaptive rejection sampling for Gibbs sampling," *J. Roy. Stat. Soc. C (Appl. Stat.)*, vol. 41, no. 2, pp. 337–348, 1992.

[46] S. L. Scott, "Bayesian methods for hidden Markov models: Recursive computing in the 21st century," *J. Amer. Stat. Assoc.*, vol. 97, no. 457, pp. 337–351, 2002.

[47] N. Whiteley, C. Andrieu, and A. Doucet, "Efficient Bayesian inference for switching state-space models using particle Markov chain Monte Carlo methods," Dept. Math., Univ. Bristol, Bristol, U.K., Rep., 2010. [Online]. Available: https://arxiv.org/abs/1011.2437

[48] F. Lindsten and T. B. Schön, "Backward simulation methods for Monte Carlo statistical inference," *Found. Trends Mach. Learn.*, vol. 6, no. 1, pp. 1–143, 2013.

[49] I. Valera, F. J. R. Ruiz, L. Svensson, and F. Perez-Cruz, "Infinite factorial dynamical model," in *Proc. Adv. Neural Inf. Process. Syst.*, Montreal, QC, Canada, 2015, pp. 1666–1674.

[50] S. Sesia, I. Toufik, and M. Baker, *LTE—The UMTS Long Term Evolution: From Theory to Practice*, 2nd ed. Chichester, U.K.: Wiley, 2011.

[51] L. R. Bahl, J. Cocke, F. Jelinek, and J. Raviv, "Optimal decoding of linear codes for minimizing symbol error rate," *IEEE Trans. Inf. Theory*, vol. 20, no. 2, pp. 284–287, Mar. 1974.

[52] S. J. Fortune *et al.*, "WISE design of indoor wireless systems: Practical computation and optimization," *IEEE Comput. Sci. Eng.*, vol. 2, no. 1, pp. 58–68, Mar. 1995.

[53] K. Au *et al.*, "Uplink contention based SCMA for 5G radio access," in *Proc. Globecom Workshops*, Austin, TX, USA, 2014, pp. 900–905.

[54] A. Bayesteh, E. Yi, H. Nikopour, and H. Baligh, "Blind detection of SCMA for uplink grant-free multiple-access," in *Proc. 11th Int. Symp. Wireless Commun. Syst.*, Barcelona, Spain, 2014, pp. 853–857.

Isabel Valera received the master's and Ph.D. degrees from University Carlos III, Madrid, in 2012 and 2014, respectively. She was a Post-Doctoral Researcher with the University of Cambridge and a Humboldt Post-Doctoral Fellowship Holder with Max Planck Institute for Software Systems. She is currently leading the Probabilistic Learning Group with the Department of Empirical Inference, Max Planck Institute for Intelligent Systems, Tübingen.

Lennart Svensson was born in Älvängen, Sweden, in 1976. He received the M.S. degree in electrical engineering and the Ph.D. degree from the Chalmers University of Technology, Gothenburg, Sweden, in 1999 and 2004, respectively, where he is currently a Professor of signal processing. His main research interests include machine learning and Bayesian inference in general, and nonlinear filtering, deep learning, and multitarget tracking in particular.

Fernando Perez-Cruz (SM'06) was born in Sevilla, Spain, in 1973. He received the B.Sc./M.Sc. degree in electrical engineering from the University of Sevilla in 1996 and the Ph.D. degree in electrical engineering from the Technical University of Madrid in 2000. He is the Chief Data Scientist with Swiss Data Science Center (ETH Zurich and EPFL). He has been a Technical Staff Member with Bell Labs and an Associate Professor with the Department of Signal Theory and Communication, University Carlos III, Madrid, and the Department of Computer Science, Stevens Institute of Technology. He has been a Visiting Professor with Princeton University under a Marie Curie Fellowship and a Research Scientist with Amazon. He has also held positions with Gatsby Unit, London, Max Planck Institute for Biological Cybernetics, Tübingen, BioWulf Technologies, New York, the Technical University of Madrid, Madrid, and Alcala University, Madrid. He has organized several machine learning, signal processing, and information theory conferences. He has supervised eight Ph.D. students and numerous M.Sc. students, as well as one junior and one senior Marie Curie Fellow. His current research interest lies in machine learning and information theory and its application to signal processing and communications.

Francisco J. R. Ruiz received the B.Sc. degree in electrical engineering from the University of Seville, Spain, in 2010, and the M.Sc. degree in multimedia and communications and the Ph.D. degree from University Carlos III, Madrid, Spain, in 2012 and 2015, respectively. He is currently a Post-Doctoral Research Scientist with the Department of Computer Science, Columbia University and the Engineering Department, University of Cambridge. He holds a Marie Skłodowska-Curie fellowship in the context of the E.U. Horizon 2020 program. His research is focused on statistical machine learning, specially Bayesian modeling and inference.