2013

# Learning and identification of wireless network internode dynamics using software defined radio

Benjamin Elliott Carroll
*Louisiana State University and Agricultural and Mechanical College,* bcarro1@lsu.edu

# LEARNING AND IDENTIFICATION OF WIRELESS NETWORK INTERNODE DYNAMICS USING SOFTWARE DEFINED RADIO

A Thesis

Submitted to the Faculty of the
Louisiana State University and
Agricultural and Mechanical College
in partial fulfillment of the
requirements for the degree of
Master of Science in Electrical Engineering

in

The Division of Electrical and Computer Engineering

by
Benjamin Elliott Carroll
B.S. in Physics, Louisiana State University, 2008
May 2013

# Acknowledgments

# Table of Contents

# List of Tables

# List of Figures

# Abstract

The recently developed paradigm of cognitive radio wireless devices has been developed with the goal of achieving more customizable and efficient spectrum utilization of commonly used wireless frequency bands. The primary focus of such spectrum utilization approaches has been to discern occupancies and vacancies over portions of the wireless spectrum without necessarily identifying how specific radio frequency (RF) devices contribute to the temporal dynamics of these occupancy patterns within the spectrum. The aim of this thesis is to utilize a hidden semi-Markov model (HSMM) statistical analysis to infer the individual occupancy patterns of specific users from wireless RF observation traces. It is proposed that the HSMM approach for RF device characterization over time may act as a first step towards performing a more complete characterization of the RF spectrum in which the inferred traffic patterns may demonstrate the coexistence of multiple networks, the specific devices comprising each distinct network, and the level of mutual interference between the component networks resultant from such coexistence.

The first main portion of this thesis is the development of a Bayesian learning framework for HSMM characterization of the wireless RF observations, with occupancy periods and each individual RF device being classified as distinct states in the HSMM. The traditional HSMM approach is supplemented with the concept of the hierarchical Dirichlet random process to achieve a minimal number of states needed to effectively capture each distinct device, without the need for strong a priori assumptions regarding the number of devices seen in the RF trace prior to computational analysis. The second portion of the thesis utilizes user-programmed cognitive radios to construct a real-time software-defined RF network environment emulation testbed to assess the accuracy of the HSMM characterization. Finally, the HSMM algorithm is tested on wireless devices operating under an actual implementation of the ubiquitous IEEE 802.11 wireless standard.

# Chapter 1

# Introduction

The wireless communications sector of the telecommunications industry has experienced rapid technological advancement and widespread acceptance over the past few decades. A wide variety of radio frequency (RF) devices have been designed and implemented in the creation of short-range communication networks for residential and occupational usage, in addition to sparking the development of nationwide mobile telephone networks of considerable scale and complexity. Due to the relatively large number of RF communications present over a limited set of available radio frequencies at any given point in time, the Federal Communications Commission (FCC) has been made directly responsible for the regulation and management of the RF spectrum and the wireless devices operating over these frequency bands. This responsibility includes the licensing of certain frequency bands to functionally similar RF devices and the enforcement of stringent operational requirements on these devices in order to mitigate potential interference arising from the coexistence of many devices over the available RF spectrum.

Reports from the FCC have indicated that certain portions of the RF spectrum remain heavily occupied, while large portions of the spectrum tend to be partially occupied or vacant over time [3]. As a result, a large amount of research has recently been devoted towards developing intelligent wireless communication protocols to promote more uniform usage of the available RF bands. The objective of this development seeks to leverage the temporal dynamics of the spectral occupancy patterns to accommodate a large variety of wireless devices with maximum efficiency and minimal performance degradation. This particular problem has been a leading factor in the development of what is known as cognitive radios.

In 2000, Joseph Mitola proposed the concept of wireless devices known as "cognitive radios" capable of learning from their local RF environment and adapting their behavior in real-time based on their inferences [12]. Given the heavily dynamic nature of the RF spectrum, this new paradigm was proposed as a means to promote spectral fairness and optimize communication performance from a multiple network perspective. Implementation of the cognitive radio concept is traditionally performed by software defined radios (SDRs) combining general purpose RF hardware devices, such as antennas and local oscillators, with software-reconfigurable signal processing chains. SDR systems allow for user customization of the RF device's functional behavior and the subsequent adaptability for a single wireless device to communicate using a variety of wireless protocols. With the capability to design RF devices through computer programming instead of laborious stages of hardware construction and testing, SDRs provide the potential for more rapid development into experimentally verifiable solutions to the spectral fairness problem, or indeed any other task envisioned

by the wireless communications engineer. For example, SDRs have been also utilized for adaptively optimizing wireless network throughput, avoiding RF interference from a vast array of different sources, and identifying specific RF devices from detecting certain features embedded within the received signal waveforms.

Recent research has proposed using hidden Markov models (HMMs) for learning the dynamic behavior of portions of the RF spectrum over time [15]. This research attempts to characterize collected RF activity into disjoint time segments corresponding to either spectral vacancies or busy periods comprising observed transmissions, with each different outcome termed as a distinct state in the HMM. The goal of such formulations is to apply the learned outcomes for such HMM fitting to predict the future behavior of the RF spectrum and allow such SDRs to adapt their behavior in order to avoid potential data collisions and promote effective spectral fairness. However, HMMs possess a degree of inflexibility in effectively modeling the time durations for each given state, as well as requiring a fixed number of states being forced upon the model before initiating any characterization of the RF activity [17]. This view of the RF frequency bands as being either occupied (ON) or vacant (OFF) over time cannot reflect the dynamics of how *distinct devices* can contribute to the observed spectral occupancy patterns, especially given the inherent multitude of potential devices transmitting over such RF bands and the resulting dynamism as these devices rapidly occupy and vacate these frequencies over long periods of time. Furthermore, this HMM characterization has no inherent mechanism for determining how potential interference or repeated instances of data packet collisions over the frequency bands of interest can affect future spectral occupancy patterns.

The focus of this thesis is to apply hidden semi-Markov models (HSMMs) with an extension of the Dirichlet random process to overcome the time domain and state space limitations of the HMM, so that multi-user characterization of the RF bands with interference detection capabilities may be achieved. Supplementing the HSMM with the Dirichlet random process concept permits flexibility in the number of states within the Markov modeling of the wireless RF activity, with the added benefit of concentrating upon a smaller model state space to encourage model parsimony. As an added benefit, this HSMM learning algorithm examines the observed wireless activity within the framework of Bayesian statistics, allowing any initial assumptions about the wireless activity to be reflected into prior probability distributions on the HSMM model's parameters. This Bayesian approach to the HSMM formulation simultaneously permits such initial parameter assumptions to be less restrictive than the HMM approach and more dependent on the data being analyzed, resulting in a more direct application towards the dynamic nature of analyzing highly time-variant wireless frequency bands.

In this research, the Bayesian HSMM algorithm is evaluated using wireless traces involving real-time SDR wireless transmissions in conjunction with observations from a widely deployed IEEE 802.11 wireless local area network (WLAN). The RF data observations from the SDR devices are designed to assess the HSMM's accuracy in multi-user detection in various degrees of coordination and interference amongst the operating RF devices. Likewise, the experimental observations of the operational 802.11 networks are conducted to assess

2

the algorithm's capability for RF device identification and network traffic analysis in a more typical wireless network environment.

## 1.1    Organization of Thesis

Chapter 2 of this thesis is meant to describe the fundamental components of a typical SDR unit and to provide a mathematical characterization of the bandpass transmission and reception signal paths for the experimental validation of this research. Chapter 3 will focus on the development of the hidden semi-Markov models within a Bayesian probabilistic framework to learn the model parameters for data in which RF devices correspond to states within the model. A detailed description of the deployed HSMM learning algorithm and the corresponding results of the HSMM learning algorithm for SDR communication scenarios will be provided and evaluated in Chapter 4. In Chapter 5, the algorithm is further extended to collected measurements of a wireless network operating under the commonly used IEEE 802.11 wireless network standard. Further topics of investigation towards improving the HSMM algorithm's performance and identifying higher-level features of wireless network characterization will be suggested in Chapter 6.

# Chapter 2

# Fundamentals of Software Defined Radio

As previously mentioned in the Introduction, the overall objective of software defined radio is to provide a synergistic combination of the basic hardware components of a typical wireless RF device with the processing power of a central processing unit (CPU). The focus of SDR system development is thus to create computer programs defining the system flow through a wireless RF radio and execute the majority of the required signal processing on a CPU throughout the program's execution, while the actual RF hardware essentially acts as the direct interface to the wireless channel of interest. Shifting the processing load of the system onto the CPU allows for high bandwidth communication, along with the flexibility of system design focused on computer programming rather than device fabrication and testing.

This chapter will focus on the general functionalities of both the hardware and software components of a functional SDR, in addition to providing a mathematical system model for how the SDR elements perform transmission and reception in a typical wireless communications scenario. Section 2.3 will additionally discuss the waveform characteristics of several different modulation schemes, since transmissions from SDRs employing such modulations will be used to assess the HSMM learning algorithm's overall performance for wireless device discovery as investigated in Chapter 4.

## 2.1   SDR Hardware

The hardware components of a typical SDR system are primarily responsible for four tasks:

1. Directly transmitting and receiving information signals over the desired RF bandwidth.

2. Modulating the information signal onto the desired RF carrier frequency, or demodulating the RF bandpass signal back to the baseband in the case of signal reception.

3. Converting the intended information signal from the discrete domain to the analog domain in the case of RF transmission, and vice versa for RF reception.

4. Processing and relaying the desired information signals for transmission to/from the host CPU.

Figure 2.1 depicts a typical layout of the separate hardware components used for each these primary tasks. The functional units of the SDR platform can be subdivided into RF daughterboards for signal transmission and reception, converters for switching between analog and digital signal representations, field programmable gate arrays for data rate handling, and a controller for interfacing the SDR with the host CPU.
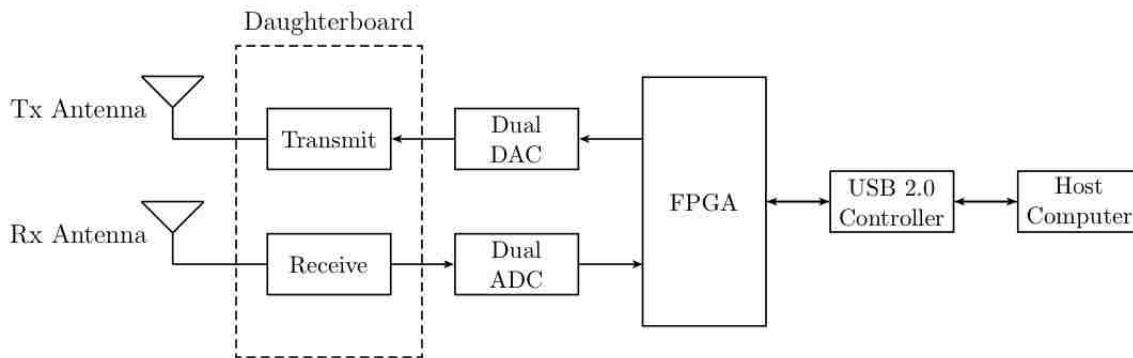


Figure 2.1: SDR Hardware Diagram

While several different SDR-compatible hardware platforms are commercially available, the Universal Software Radio Peripheral (USRP) manufactured by Ettus Research LLC has achieved widespread acceptance for both research and commercial applications. A USRP B100 Bus Series device was utilized in the experiments detailed in Chapter 4, and thus the technical specifications and limitations of this specific radio will be emphasized. Nonetheless, the general limitations of the USRP series are additionally present in SDR platforms provided by other manufacturers, each with their own specific design for balancing high performance radio functionality with the complications of RF design from generic electronic hardware components. Most of the information within this section has been extrapolated from the GNURadio website and the USRP B100 data sheets provided by Ettus Research [2, 8].

## 2.1.1 Daughterboards

Each USRP is capable of housing one or more daughterboards responsible for RF tuning and filtering of noise and spurious frequencies over the desired RF bandwidth. As a result, the daughterboards serve as the closest link in the signal chain to the RF channel. Most of the available RF daughterboards actually possess two separate signal paths, one for transmission and one for reception, and thus are capable of achieving full-duplex communication [1]. In the case of the SBX 400-4400 MHz daughterboard used in the experiments of this thesis, both signal paths have their own 40 MHz bandwidth filters placed directly in front of the attached antennas. Furthermore, each signal path contains local oscillators for carrier modulation and demodulation, as necessitated to achieve full duplex capability. It is worth emphasizing that

the information signals processed through the USRP daughterboards are continuous-time analog RF signals.

## 2.1.2   Analog-to-Digital and Digital-to-Analog Converters

Each USRP contains 2 separate analog-to-digital converters (ADCs) which digitize the analog RF signals received from the daughterboard. Each ADC is capable of real and complex sampling. Complex sampling actually requires the usage of 2 ADCs to decompose the RF signal into its in-phase (I) and quadrature (Q) components, thus limiting the USRP to a single RF input configuration. Since the complex sampling has direct application and widespread usage in wireless communications design, this option is regularly chosen in most USRP experiments. Each ADC is capable of digitizing the analog RF inputs into 12-bit samples at a rate of 64 million samples per second (MS/s). The Shannon-Nyquist sampling theorem suggests the USRP could then perfectly reconstruct received analog signals of 64 MHz bandwidth in the case of complex sampling. However, the front-end filtering in the daughterboard's signal path implies that only signals of less than 40 MHz bandwidth can be reconstructed without the introduction of aliasing.

The digital-to-analog converters (DACs) perform the opposite functionality of the ADC by transforming baseband data into the analog domain. The 2 DACs in the USRP B100 can accept 14-bit samples at a rate of 128 MS/s. Once again, the bandwidth of the transmitted signals must fall below 40 MHz bandwidth to prevent significant waveform distortion from being introduced by the daughterboard's transmitter filter.

## 2.1.3   Field Programmable Gate Arrays

As mentioned in the previous section, the ADC sampling rate alone has imposed a limit of 64 MHz bandwidth maximum for aliasing-free representation of the received RF signals. Since the received baseband signal in the USRP requires transmission to the host CPU for any additional digital processing, it is necessary to additionally consider the *data rate* at which the digitized samples can be reliably transmitted to the CPU. The USRP B100 sends digitized complex IQ samples in 16-bit signed integer format over USB 2.0, which has an average data transfer rate of 32 megabits per second (MB/s). Since there are 4 bytes in one such sample, the maximum ADC sampling rate that can be effectively transported to the host CPU is reduced to 8 MS/s, consequently reducing the maximal RF signal bandwidth to 8 MHz.

A primary role of the field programmable gate array (FPGA) within the USRP is to perform digital down conversion of the sample stream from the ADC to meet the 32 MB/s

data rate constraint imposed by the USB connection.[1] The standard FPGA image for the USRP includes 2 digital down converters (DDCs), each composed of 4 stage cascaded integrator comb (CIC) decimation filters followed by halfband filters, to remove the appropriate amount of samples from the data stream. These filters additionally act to minimize aliasing and maximize out-of-band rejection.[2] For the transmit path, the executed SDR program may require the sample rate from the input baseband samples to be increased before being processed by the DAC. The FPGA image accordingly possesses CIC interpolation filters to perform this functionality, while the digital up converters (DUCs) responsible for completing the sample rate conversion actually reside within the DACs themselves.

The FPGA is also responsible for directly interfacing with the USRP's USB 2.0 controller for data transfer to and from the host computer. When data from the USRP and computer involves multiple channels, the FPGA also multiplexes and demultiplexes the data stream to feed the appropriate transmit and receive channels through the USRP. This direct connection between the FPGA and computer is also what permits the FPGA to be reprogrammed.

## 2.2 SDR Software

The true power of the SDR paradigm resides in the customizability and reconfigurability at the baseband processing level. The USRP is compatible with the extensively developed open-source GNU Radio signal processing development toolkit which provides the most commonly used components of a working radio system. The GNU Radio toolkit is composed as modular C++ programming blocks that are connected together to construct the signal chain of the SDR. Examples of standard blocks includes modulators and demodulators, filters, equalizers, channel coders and decoders, and other data manipulation capabilities. Since C++ is a low-level programming language, the signal processing blocks accommodate the real-time requirements for a high-performance wireless radio platform. Any added functionality not possessed in the GNU Radio library can also be developed through user-created C++ functions.

The complexity of implementing the GNU Radio toolkit is largely minimized by 'wrapping' the C++ processing blocks to equivalent modules within the Python programming language via the Simplified Wrapper Interface Generator (SWIG) [8]. Since Python is a higher-level object-oriented programming language than C++, interconnecting the processing blocks into a functional program is much more easily facilitated by the creation of what are termed flow graphs. Each flow graph consists of the connections of such blocks, depending on whether each block acts as a data source or sink. Each flow graph then possesses a scheduler that handles data transfer between the C++ blocks, with the use of input and

---

[1]Other USRP models allow Ethernet connectivity to the host CPU, allowing for greater observable RF bandwidth than the USRP B100. Nonetheless, hardware-imposed bandwidth constraints will still be present in a similar manner despite the increase in throughput.

[2]Since FPGAs are primarily user-programmable integrated circuits, it is possible to reprogram the FPGA in Verilog to customize such filter design or to enable more DDCs.

output buffers to ensure real-time processing with minimal latency as data flows through the individual processing elements.

## 2.3   Baseband Waveform and Modulation Characteristics

The following sections mathematically describe how the USRP signals are converted back and forth to bandpass signals suitable for transmission and reception over radio frequencies (Section 2.3.1). Also, a short description of several data modulation techniques are discussed in Sections 2.3.3-2.3.6, as these modulation techniques are employed in Chapter 4 for generating USRP transmissions. The mathematical analysis has been adapted and extrapolated from the more detailed expositions found in [16] and [18].

### 2.3.1   Bandpass Signal Representation

By definition, all wireless signals transmitted over radio frequencies are bandpass signals, since the bandwidth-restricted information signals are used to modulate a carrier frequency $f_c$ greater than zero. The desired information signals of interest however are baseband signals, meaning that they possess a finite bandwidth centered around zero frequency. Thus, it is often useful to characterize the bandpass signals into complex baseband representation in order to analyze the information-bearing portion of the wireless signals regardless of the specific carrier frequency being utilized.

Figure 2.2 depicts a complex demodulator used to transform a bandpass signal $s(t)$ into its baseband equivalent $\tilde{s}(t)$. The first step in this process involves the creation of an analytic signal $s_+(t)$ containing only positive frequency components. The analytic signal defined in (2.1) is then a complex-valued waveform composed of the original bandpass signal $s(t)$ and its Hilbert transform $\hat{s}(t) = \mathcal{H}[s(t)] = s(t) * \frac{1}{\pi t}$.

$$s_+(t) = s(t) * \left[ \frac{1}{2}\delta(t) + \frac{j}{2\pi t} \right] \tag{2.1}$$

$$= s(t) * \frac{1}{2}\delta(t) + s(t) * \frac{j}{2\pi t} \tag{2.2}$$

$$= \frac{1}{2}s(t) + \frac{j}{2}\hat{s}(t) \tag{2.3}$$

The baseband equivalent, or complex envelope, $\tilde{s}(t)$ is then generated by scaling $s_+(t)$ and demodulating by the carrier frequency $f_c$.

Since the complex envelope is a complex-valued signal, it possesses both real and imaginary parts denoted as its in-phase (I) and quadrature (Q) signal components respectively.

$$\tilde{s}(t) = 2s_+(t)e^{-j2\pi f_c t} \tag{2.4}$$

$$= \left[ s(t) + j\hat{s}(t) \right] e^{-j2\pi f_c t} \tag{2.5}$$

$$= \left[ s(t)\cos 2\pi f_c t + \hat{s}(t)\sin 2\pi f_c t \right] + j\left[ -s(t)\sin 2\pi f_c t + \hat{s}(t)\cos 2\pi f_c t \right] \tag{2.6}$$

$$= s_I(t) + js_Q(t) \tag{2.7}$$
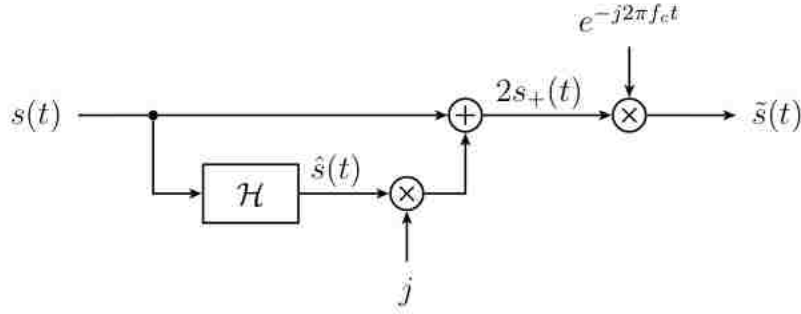


Figure 2.2: Complex Bandpass Demodulation

In a similar manner, Figure 2.3 depicts the modulation of the baseband equivalent signal onto the carrier frequency of interest. Note that the real part of the newly modulated signal is the only relevant information in the signal, since the bandpass signals of interest are real valued.

$$s(t) = \mathrm{Re}\left\{ \tilde{s}(t)e^{j2\pi f_c t} \right\} \tag{2.8}$$

$$= \mathrm{Re}\left\{ \left( s_I(t) + js_Q(t) \right)e^{j2\pi f_c t} \right\} \tag{2.9}$$

It is often more convenient for $s(t)$ to be expressed in polar coordinates, in terms of an instantaneous envelope $a(t) = \sqrt{s_I^2(t) + s_Q^2(t)}$ and instantaneous phase $\phi(t) = \tan^{-1}\frac{s_Q(t)}{s_I(t)}$.

$$s(t) = a(t)\cos\left[ 2\pi f_c t + \phi(t) \right] \tag{2.10}$$

## 2.3.2   Modeling SDR Transmission and Reception Signal Flow

Figures 2.4 and 2.5 display system block diagrams used to model the signal flow for USRP transmitters and receivers respectively. In these depictions, $f_c$ represents the RF carrier frequency used for communicating over the wireless channel. For both signal paths,

9
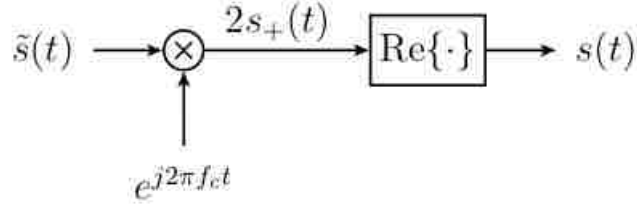
Figure 2.3: Complex Bandpass Modulation

all filtering operations, including front-end filtering at the daughterboard end and halfband filtering in the FPGA, are incorporated into shaping function $b(t, \overrightarrow{x_i})$.
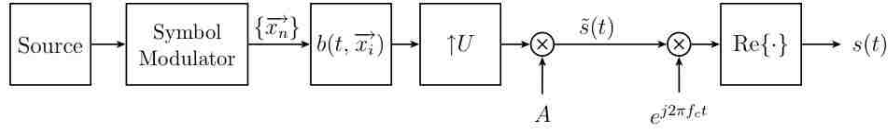


Figure 2.4: SDR Transmitter Diagram
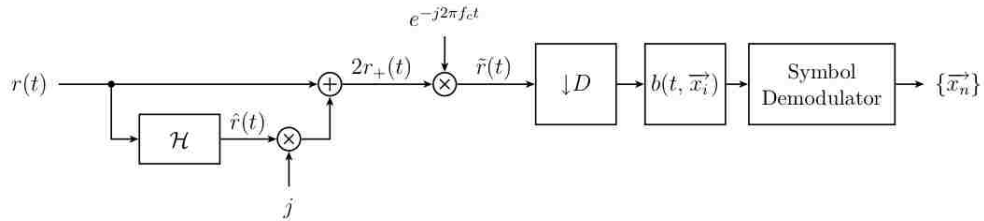


Figure 2.5: SDR Receiver Diagram

Also note that the front-end filters present in the USRP daughterboards are common to both the transmission and reception signal paths. The presence of such filtering introduces some amount of amplitude and phase distortion of the transmitted and received signals even if additional pulse-shaping filters are used to mitigate intersymbol interference. Since the transmitter's front-end filter will not be exactly identical for two USRPs possessing the same daughterboard models, each separate device should produce its own unique spectral characteristics regardless of the baseband waveforms used in any communications over the wireless channel. Since this project of this research deals with identifying specific RF transmitters, it is also worth noting that the front-end filter in the reception path will likewise introduce amplitude and phase distortions of the receiver signal, thus complicating the ability to extract arbitrarily precise timing information about onset RF device activity.

Equation (2.11) mathematically describes how transmitted data is modulated into symbols $\{\overrightarrow{x_n}\}$ and transformed to a complex baseband signal before carrier frequency modulation

10

and transmission over the wireless channel. $A$ represents a separate amplitude corresponding to signal amplification for combating channel noise and aiding symbol detection at the corresponding receiver. The baseband signal $\tilde{s}(t)$ is thus a continuous-time signal produced by a stream of modulated symbols at a symbol rate of $T$ symbols per second.

$$\tilde{s}(t) = A \cdot \sum_{n=-\infty}^{\infty} b(t - nT, \vec{x_n}) \tag{2.11}$$

## 2.3.3  M-ary Phase Shift Keying

One common method for modulating source binary data into symbols is to introduce discrete phase shifts to the baseband equivalent signal in what is known as phase shift keying (PSK). $M$-ary phase shift keying maps source binary data into $\{\vec{x_n}\}$ with $M$ possible symbols $x_n \in \{0, 1, \ldots, M-1\}$. The excess phase $\theta_n$ determining the amount of phase shift per symbol period is given by (2.12).

$$\theta_n = \frac{j2\pi x_n}{M} \tag{2.12}$$

The complex baseband equivalent signal waveform for $M$-ary PSK is given by (2.13). For simplicity, the shaping function of the waveform $h_a(t)$ is modeled as a truncated unit step function of duration $T$ as seen in (2.16).

$$\tilde{s}(t) = A \cdot \sum_{n=-\infty}^{\infty} h_a(t - nT) \, \exp\left\{\frac{j2\pi x_n}{M}\right\} \tag{2.13}$$

$$
\begin{aligned}
h_a(t) &= u_T(t) & \text{(2.14)} \\
&= u(t) - u(t - T) & \text{(2.15)} \\
&= \begin{cases} 1, & 0 \leq t \leq T \\ 0, & \text{else} \end{cases} & \text{(2.16)}
\end{aligned}
$$

Two commonly implemented $M$-ary PSK waveform are binary phase shift keying (BPSK) where $M = 2$ and quadrature phase shift keying (QPSK) where $M = 4$.

## 2.3.4  M-ary Quadrature Amplitude Modulation

While $M$-ary phase shift keying focuses on modulating the instantaneous phase of the baseband signal, $M$-ary quadrature amplitude modulation (QAM) instead directly modulates

the instantaneous envelope of the baseband signal. The real and imaginary parts of each data symbol $x_n = x_{I,n} + jx_{Q,n}$ are separately mapped to one of $M$ possible values. QAM signal space constellations can be constructed in a variety of manners given $M$, although square constellations result when $M$ is an even power of 2 ($x_{I,n}, x_{Q,n} \in \{\pm 1, \pm 3, \dots, \pm(\sqrt{M} - 1)\}$). This signal mapping results in a complex baseband signal waveform can thus be expressed by (2.17) and (2.18).

$$\tilde{s}(t) = A \cdot \sum_{n=-\infty}^{\infty} x_n \, h_a(t - nT) \tag{2.17}$$

$$= A \cdot \sum_{n=-\infty}^{\infty} (x_{I,n} + jx_{Q,n}) \, h_a(t - nT) \tag{2.18}$$

### 2.3.5   Gaussian Minimum Shift Keying

Since coherent $M$-ary PSK introduces abrupt changes in the carrier phase at every data symbol period, the resulting waveforms possess poor spectral compactness due high-frequency components introduced by the phase shifting operations. Continuous pulse modulation (CPM) was developed to improve upon these disadvantages by instead modulating the carrier phase continuously over the course of transmissions. The CPM baseband waveform can be expressed generally as in (2.19), where $\theta_0$ is the initial phase at $t = 0$.

$$\tilde{s}(t) = A \, \exp\left\{ j\left(\phi(t) + \theta_0\right) \right\} \tag{2.19}$$

The data symbols $x_n$ thus introduce modulation with the carrier frequency through the excess phase $\phi(t)$.

$$\phi(t) = 2\pi h \int_0^t \sum_{k=0}^{\infty} x_k \, h_f(\tau - kT) \, d\tau \tag{2.20}$$

The modulation index $h$ in (2.20) is a measure of the instantaneous frequency deviation of the excess phase seen in (2.22).

$$f_{dev}(t) = \frac{1}{2\pi} \frac{d\phi(t)}{dt} \tag{2.21}$$

$$= h \cdot \sum_{k=0}^{\infty} x_k \, h_f(t - kT) \tag{2.22}$$

Here, $h_f(t)$ is a frequency-shaping function with zero value for $t < 0$ and $t > LT$, normalized to have area of $\frac{1}{2}$. An equivalent pulse shaping function $\beta(t)$ may be defined as in the

following equation.

$$\beta(t) = \begin{cases} 0, & t < 0 \\ \int_0^t h_f(\tau)\, d\tau, & 0 \le t \le LT \\ \frac{1}{2}, & t \ge LT \end{cases} \tag{2.23}$$

The resultant full-response CPM baseband waveform may then be expressed by (2.24), where $L = 1$.

$$\tilde{s}(t) = A \cdot \sum_{n=-\infty}^{\infty} \exp\left\{ j\left( \pi h \sum_{k=0}^{n-1} x_k + 2\pi h x_n \beta(t) \right) \right\} u_T(t) \tag{2.24}$$

By choosing binary values of $x_n \in \{-1, +1\}$ and setting the modulation index to $h = \frac{1}{2}$, the baseband waveform is expressed according to (2.25).

$$\tilde{s}(t) = A \cdot \sum_{n=-\infty}^{\infty} \exp\left\{ j\left( \frac{\pi}{2} \sum_{k=0}^{n-1} x_k + \frac{\pi}{2T} x_n \right) \right\} u_T(t) \tag{2.25}$$

As a result, one of two frequencies $f$ for transmission is chosen per baud interval depending on the value of $x_n$, with $f = f_c \pm \frac{1}{4T}$. Since the minimum frequency separation needed to maintain orthogonality between these two frequencies is $\frac{1}{2T}$, this modification to the CPM transmission scheme is known as minimum shift keying (MSK).

Since this version of MSK uses a rectangular frequency shaping function $h_f(t) = \frac{1}{2T} u_T(t)$, the frequency spectrum of the baseband waveform may be further compacted by adding another lowpass filter after the MSK modulating signal. Choosing $H(f)$ as in (2.26) to resemble a Gaussian waveform with zero mean frequency and 3-dB filter bandwidth $B$ results in a Gaussian minimum shift keying (GMSK) modulation scheme.

$$H(f) = \exp\left\{ -\left( \frac{f}{B} \right)^2 \frac{\ln 2}{2} \right\} \tag{2.26}$$

The time-domain representation of this Gaussian filter is thus given by (2.27) through application of the inverse Fourier transform. By lowering the value of the bandwidth-time product $0 < BT < 1$, the baseband waveform will be made more compact in the frequency domain.

$$h(t) = \sqrt{\frac{2\pi}{\ln 2}} B \exp\left\{ -\frac{2\pi^2}{\ln 2} (BT)^2 \right\} \tag{2.27}$$

The equivalent frequency-shaping filter $h_f(t)$ can then be expressed as a cascade of the rectangular shaping function $h_a(t) = \frac{1}{2T} u_T(t)$ and the Gaussian filter $h(t)$.

$$h_f(t) = h_a(t) * h(t) \tag{2.28}$$

$$= \frac{1}{2T} u_T(t + T/2) * \sqrt{\frac{2\pi}{\ln 2}} B \exp\left\{ -\frac{2\pi^2}{\ln 2}(BT)^2 \right\} \tag{2.29}$$

$$= \frac{1}{2T} \sqrt{\frac{2\pi}{\ln 2}}(BT) \int_{t/T-1/2}^{t/T+1/2} \exp\left\{ -\frac{2\pi^2}{\ln 2}(BT)^2 x^2 \right\} dx \tag{2.30}$$

Note that $h_a(t)$ requires a time shift for the frequency-shaping filter to become causal and thus implementable for direct wireless applications. The baseband waveform for a GMSK modulated signal is then expressed by (2.31). Figure 2.6 shows a graphical representation of the transmitter diagram with the addition of the Gaussian shaping filter.

$$\tilde{s}(t) = A \cdot \sum_{n=-\infty}^{\infty} \exp\left\{ j\left( \frac{\pi}{2} \sum_{k=0}^{n-1} x_k + \pi\, x_n\, \beta(t) \right) \right\} \tag{2.31}$$

where $\beta(t) = \int_{-\infty}^{t} h_f(\tau)\, d\tau$.
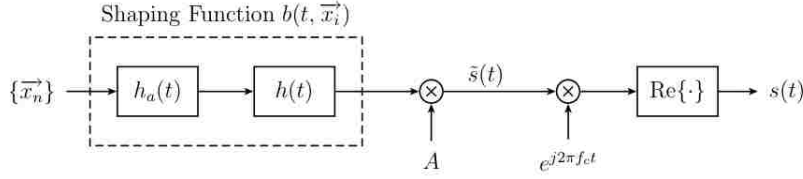


Figure 2.6: GMSK Transmitter Diagram

## 2.3.6 Orthogonal Frequency Division Multiplexing

In contrast to the modulation schemes detailed earlier, orthogonal frequency division multiplexing (OFDM) is a multi-carrier modulation scheme using multiple mutually orthogonal subcarriers to transmit information in parallel streams. As a consequence, OFDM is a block modulation scheme where $N$ data symbols of duration $T_s$ are converted to a block of $N$ parallel data symbols with a new symbol duration of $T = NT_s$. By spacing the subcarriers by multiples of $1/T$ Hz apart, the subcarriers remain orthogonal regardless of the phase differences between all of the subcarriers. The subcarrier orthogonality within a block $\overrightarrow{x_n} = [x_0, x_1, \ldots, x_{N-1}]$ is easily accomplished by taking the $N$-point inverse discrete Fourier transform of the symbols $x_n, k$ from each subcarrier $k$.

$$x_n = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} x_{n,k} e^{j2\pi kt/T}, \quad 0 \le n \le N-1. \tag{2.32}$$

14

OFDM is a popular modulation scheme for reducing intersymbol interference from a non-uniform channel response over the transmission bandwidth, since the available channel bandwidth is subdivided into many subchannels of approximately constant magnitude. The addition of a cyclic prefix also provides a time guard interval to prevent interference from being introduced by successive OFDM blocks. The cyclic prefix consists of appending $N_g$ symbols before the original $N$ data symbols in the original OFDM block, making the new OFDM symbol period $T_g = \left(\frac{N}{N+N_g}\right)T_s$. The resultant baseband waveform is seen in (2.33), with the guard interval defined as $\alpha_g = N_g/N$.

$$\tilde{s}(t) = A \cdot \sum_{n=-\infty}^{\infty} \left[ u_{T_g}(t) \sum_{k=0}^{N-1} x_{n,k}\, e^{j2\pi kt/T_g} + u_{\alpha_g T_g}(t + \alpha_g T_g) \sum_{k=0}^{N-1} x_{n,k}\, e^{j2\pi k(t+T_g)/T_g} \right] \quad (2.33)$$

By judiciously choosing the cyclic prefix's samples as $[x_{n,N-N_g-1}, x_{n,N-N_g}, \ldots, x_{n,N-1}]$, multipath interference for wireless channels of length-$N_g$ impulse responses can be eliminated through removal of the first $N_g$ symbols in the OFDM block at the receiver end. Since no additional information is transmitted through the cyclic prefix, the number of subcarriers $N$ is usually chosen to be much larger than the cyclic prefix length $N_g$ to improve the effective information data rate from transmitter to receiver. Figure 2.7 shows a typical system diagram for an OFDM transmitter.
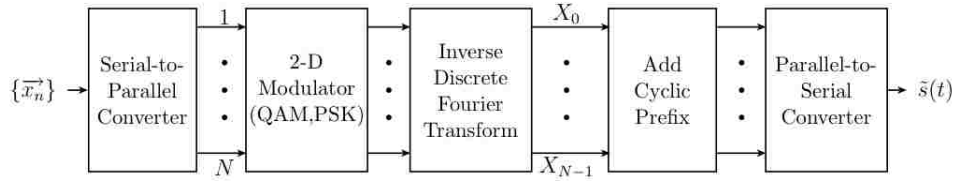


Figure 2.7: OFDM Transmitter Diagram

# Chapter 3

# A Bayesian Approach to Hidden Semi-Markov Modeling

## 3.1 Hidden Markov Models

Many mathematical models in engineering applications involve analyzing the input-output relationships for systems possessing a large number of potential variables, some of which may remain unknown. The intractability of deterministic analysis in such situations often leads to treating the observable system variables as random variables instead, where the variables of interest are treated within a probabilistic framework. Markov chains are a popular statistical tool for modeling the temporal evolution of discrete time systems as a memoryless process undergoing transitions between a finite number of states at each particular point in time. More specifically, a time series with a collection of random variables $X_1, X_2, \ldots, X_t, X_{t+1}, X_{t+2}, \ldots$ at each time unit $t$ can be modeled as a Markov chain where the state of a random variable $X_{n+1}, n \in \{1, 2, \ldots, t-1\}$ is conditionally independent of past and future values given the present state $X_n$.[1]

$$P(X_{n+1} = x | X_1 = x_1, X_2 = x_2, \ldots X_n = x_n) = P(X_{n+1} = x | X_n = x_n) \qquad (3.1)$$

In this equation, the values of each random variable $X_i$ belong to a countable state space, with $X_i \in \{1, 2, \ldots, N\}$ where $N < \infty$.

Many Markov chain applications involve situations where the state of the underlying process can not be determined directly but only through a set of observations $Y_1, Y_2, \ldots, Y_n$ that are a probabilistic function of the system's state. For example, multiple different states may possibly produce the same observable value $Y_i = y$. Since there is no one-to-one mapping between the observations and the underlying state, the process is said to be hidden. A system model possessing this partially observable nature along with the Markovian property in (3.1) is known as a hidden Markov model (HMM).

---

[1]Throughout this chapter, random variables will be denoted with capital letters, while the outcomes associated with such random variables will be given lowercase letters.

A complete HMM $\boldsymbol{\lambda}$ is characterized by five quantities. As mentioned earlier, a HMM consists of a finite state space $\Omega_X = \{1, 2, \ldots, N\}$ of cardinality $N$. An initial state distribution $\boldsymbol{\pi}$ describes the discrete probabilities $0 \leq \pi_i \leq 1$ that each state $S_i$ is the first state occurring in the random process.

$$\pi_i = P(X_1 = i), \quad i = 1, 2, \ldots, . \tag{3.2}$$

The probabilities in which states transition amongst themselves at each time step is described by a state transition matrix $\mathbf{A}$.[2]

$$a_{ij} = P(X_{t+1} = j | X_t = i), \quad 1 \leq i, j \leq N \text{ with } 0 \leq a_{ij} \leq 1 \text{ and } \sum_{i=1}^{N} a_{ij} = 1 \tag{3.3}$$

Associated with each state is an emission probability distribution $b_i$ describing the probability of producing each possible observable value from a finite countable alphabet set $\{y_1, y_2, \ldots, y_M\}$ where $M < \infty$.

$$b_i(m) = P(Y_t = y_m | X_t = i), \quad 1 \leq i \leq N \text{ and } 1 \leq m \leq M \tag{3.4}$$

Thus, the complete HMM can be given as $\boldsymbol{\lambda} = (\boldsymbol{\pi}, \mathbf{A}, \mathbf{B})$, where $\mathbf{B}$ is the $N \times M$ matrix described in (3.4).

Three fundamental tasks are associated with most HMM applications:

1. Given observation sequence $Y = \{Y_1 = y_1, Y_2 = y_2, \ldots, Y_T = y_T\}$ and model $\boldsymbol{\lambda}$, how well does the model match the observation sequences? (Calculate $P(Y|\boldsymbol{\lambda})$)

2. Given $Y$ and $\boldsymbol{\lambda}$, find the optimal possible state sequence $X = \{X_1 = x_1, X_2 = x_2, \ldots, X_T = x_T\}$ for what has been observed. (Maximize $P(X|Y, \boldsymbol{\lambda})$)

3. Given $Y$, what are the optimal model parameters associated with this observation sequence? (Maximize $P(Y|\boldsymbol{\lambda})$)

A conclusive solution to each of these problems would involve enumeration of the probabilities associated with every possible observation sequence, which is inefficient and computationally expensive for large observation sequences. For this reason, it is worthwhile to explore the application of the forward-backward algorithm for the first problem alone, in which these calculations are performed for each time step for partial observation sequences within the entire sequence $Y$.[3]

---

[2]It is worth noting that the state transition matrix $\mathbf{A}$ may be time-inhomogeneous, meaning that $\mathbf{A}$ changes as a function of time $t$. All further analysis regarding Markov models in this project implicitly assumes time-homogeneous Markov processes instead.

[3]The Viterbi and expectation-maximization algorithms are extensively used in a variety of HMM applications for solving the second and third problems respectively. More information into these algorithms can be found in [17]. Only the first problem is analyzed, since this forward-backward procedure is directly used in the context of HSMMs for this thesis.

### 3.1.1 Forward-Backward Algorithm

As mentioned in the previous paragraph, evaluating the performance of how well an HMM $\boldsymbol{\lambda}$ matches an observation sequence $Y = \{Y_1 = y_1, Y_2 = y_2, \ldots, Y_T = y_T\}$ is equivalent to calculating $P(Y|\boldsymbol{\lambda})$. There is also an associated state sequence $X = \{X_1 = x_1, X_2 = x_2, \ldots, X_T = x_T\}$ responsible for producing such observations, although this process is not directly observable. In this analysis, it is assumed that the underlying state sequence $X$ is known.

Assuming that all observations in $Y$ are statistically independent, the probability of producing such an observation sequence conditioned on a realization of the state sequence $X$ is then calculated according to (3.5) and (3.6).

$$P(Y|X, \boldsymbol{\lambda}) = \prod_{t=1}^{T} P(Y_t|X_t, \boldsymbol{\lambda}) \tag{3.5}$$

$$= b_{x_1}(Y_1) \cdot b_{x_2}(Y_2) \cdot b_{x_3}(Y_3) \cdots b_{x_T}(Y_T) \tag{3.6}$$

The probability of the assumed state sequence $X$ occurring in terms of the model $\boldsymbol{\lambda}$ is then computed through repeated usage of the elements within the state transition matrix $\mathbf{A}$ after drawing an initial state $x_1$ according to the initial state distribution $\boldsymbol{\pi}$.

$$P(X|\boldsymbol{\lambda}) = \pi_{x_1} \cdot a_{x_1 x_2} \cdot a_{x_2 x_3} \cdots a_{x_{T-1} x_T} \tag{3.7}$$

Using these two quantities, the probability of the observation sequence is obtained by summing the product of (3.5) and (3.7) over all possible state sequences $\Omega_X^T = \{(X_1, X_2, \ldots, X_T) : X_1 \in \Omega_X, X_2 \in \Omega_X, \cdots, X_T \in \Omega_X\}$.

$$P(Y|\boldsymbol{\lambda}) = \sum_{\Omega_X^T} P(Y|X, \boldsymbol{\lambda}) P(X|\boldsymbol{\lambda}) \tag{3.8}$$

$$= \sum_{\Omega_X^T} \pi_{x_1} b_{x_1}(Y_1) \cdot a_{x_1 x_2} b_{x_2}(Y_2) \cdot a_{x_2 x_3} b_{x_3}(Y_3) \cdots a_{x_{T-1} x_T} b_{x_T}(Y_T) \tag{3.9}$$

Instead of directly computing the sum in (3.9), it is worth noting that there are $N$ possible paths for a state $x_j$ to occur at time $t+1$ after a transition from $X_t$. Thus a forward variable $\alpha_t(i)$ for each state $i$ at time $t$ is defined by (3.10).

$$\alpha_t(i) = P(Y_1, Y_2, \cdots, Y_t, X_t = i|\boldsymbol{\lambda}) \tag{3.10}$$

At the first time instant $t = 1$, the forward variable for all states $i \in \Omega_X$ can easily be calculated according to (3.11).

$$\alpha_1(i) = \pi_i b_i(Y_1), \quad 1 \le i \le N \tag{3.11}$$

At time instant $t = 2$, the forward variable $\alpha_2(j)$ can be computed using the previously calculated value of $\alpha_1(i)$ as in (3.12).

$$\alpha_2(j) = \left[\sum_{i=1}^{N} \alpha_1(i)a_{ij}\right] b_j(Y_2), \quad 1 \le j \le N \tag{3.12}$$

Thus, the forward variable at each time instant $1 \le t \le T - 1$ can be expressed by the following equation through induction as seen in (3.13).

$$\alpha_{t+1}(j) = \left[\sum_{i=1}^{N} \alpha_t(i)a_{ij}\right] b_j(Y_{t+1}), \quad 1 \le j \le N; \; t = 1, 2, \ldots, T - 1 \tag{3.13}$$

Figure 3.1 shows a graphic visualization of how $a_{t+1}(j)$ can be calculated from $a_t(i)$ from the state trajectory in time. The quantity $P(Y|\boldsymbol{\lambda})$ in (3.9) can then be expressed as the sum of the forward variable at time $T$ over all states. Since each of the $N$ states at a time step $t + 1$ can be reached from each of the $N$ possible states at time $t$, the probability of the observation sequence $Y$ given the HMM $\boldsymbol{\lambda}$ can be computed as a sum of the forward variables according to (3.14).

$$P(Y|\boldsymbol{\lambda}) = \sum_{i=1}^{N} \alpha_T(i) \tag{3.14}$$

In summation, the joint probability of the entire observation sequence is broken down into a sum of the probabilities for partial observation sequences at each time instant, since the entries of the state transition matrix allows easy updating for each one-step transition in time. Using this approach, $P(Y|\boldsymbol{\lambda})$ can be calculated in roughly $N^2T$ calculations as opposed to $2T \cdot N^T$ required through direct calculation of (3.9) [17].
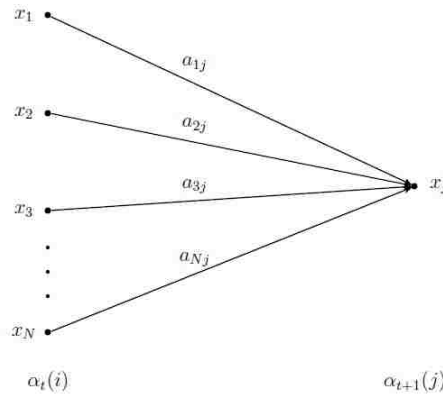


Figure 3.1: Forward Variable Graphic Depiction

Just as the forward variable is used to calculate the probability of a partial observation sequence from time 1 to time $t$, a backward variable $\beta_t(i)$ can likewise be defined to calculate the probability of a partial observation sequence from time $t+1$ to time $T$.[4]

$$\beta_t(i) = P(Y_{t+1}, Y_{t+2}, \ldots, Y_T | X_t = i, \boldsymbol{\lambda}) \tag{3.15}$$

In this case, the backwards variable at time $T$ is initialized to 1 for all states.

$$\beta_T(i) = 1, \quad 1 \leq i \leq N \tag{3.16}$$

For any state $j$ a time $T$, there are $N$ possible paths to the state $i$ at time $T-1$ that could have resulted in a transition to $j$, as seen in Figure 3.2. Solving through induction as in the case of the forward variable, the backwards variable can be expressed as in (3.17).

$$\beta_t(i) = \sum_{j=1}^{N} a_{ij} b_j(Y_{t+1}) \beta_{t+1}(j), \quad 1 \leq i \leq N, \ t = T-1, T-2, \ldots, 1 \tag{3.17}$$

Once again, this involves roughly $N^2 T$ calculations as in the forward case.
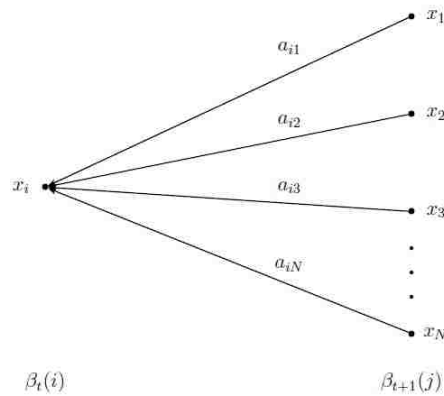


Figure 3.2: Backward Variable Graphic Depiction

## 3.2 Hidden Semi-Markov Models (HSMMs)

Despite the effectiveness of HMMs in modeling a diverse array of physical phenomena, HMMs are of limited utility in modeling the durations in which a state may persist before exiting to a different state. Suppose that a state $i$ in the HMM is entered at time $t$ and remains in that state for a duration of $d$ time units before exiting to another state $j \neq i$.

---

[4]It is this backward variable that will be of extreme utility in the computation of the hidden semi-Markov algorithms discussed later in this thesis.

The duration probability distribution $p_i(d)$ of state $i$ is thus expressed by (3.18) and (3.19).

$$p_i(d) = P(S_t = i, S_{t+1} = i, \ldots, S_{t+d-1} = i, S_{t+d} \neq i | S_{t-1} \neq i) \tag{3.18}$$
$$= (a_{ii})^{d-1}(1 - a_{ii}) \tag{3.19}$$

From examination, the HMM has implicitly imposed a restriction on the potential duration probability distribution to the form of a discrete geometric probability distribution with the expectation value of the duration $d_i$.

$$\bar{d}_i = E[d_i] \tag{3.20}$$
$$= \sum_{d=1}^{\infty} d\, p_i(d) \tag{3.21}$$
$$= \frac{1}{1 - a_{ii}} \tag{3.22}$$

Consequently, it is sometimes useful to extend the HMM to allow the state durations to be a random variable as well, where each state $i$ is further endowed with a state duration probability distribution $D_i(d)$.

$$D_i(d) = P(X_t = 1, X_{t+1} = i, \ldots, X_{t+d-1} = i, X_{t+d} \neq i), \quad 1 \leq i \leq N \tag{3.23}$$

This allows the model to incorporate probabilistic formulations for how long states may persist before switching. This addition to the HMM formulation results in what is known as a hidden semi-Markov model (HSMM). The HSMM has an additional stipulation that no state may transition into itself, so that the state duration distributions fully capture the temporal behavior of the underlying process. These modifications to the original HMM formulation imply that diagonal entries of the state transition matrix are zero by definition ($a_{ii} = 0$ for $1 \leq i \leq N$).

The Markov property must also be slightly altered from (3.1), since the conditional independence must be formulated by the underlying superstate sequence, describing the sequence in which the states have transitioned without considering the durations for each state's visit. If $s$ represents the $s$-th visited state in the superstate sequence, the modified Markov property can be given by (3.24).

$$P(X_{s+1} = x | X_1 = x_1, X_2 = x_2, \ldots X_s = x_s) = P(X_{s+1} = x | X_s = x_s) \tag{3.24}$$

The generative process for a sequence of observations $Y$ from an HSMM can be described by the following outline:

1. Draw an initial state $x_1 = i$ according to the initial state distribution $\boldsymbol{\pi}$.

2. Draw a random duration $d_1$ from the duration probability distribution $D_i(d)$.

3. A sequence of $d_1$ independent, identically distributed (iid) observations $y_{1:d_1} = \{y_1, y_2, \ldots, y_{d_1-1}, y_{d_1}\}$ are emitted from state $i$ according to the emission probability distribution $b_i(y)$.

4. At time $d_1 + 1$, draw the second visited state $x_2 = j$ according to the transition distribution given by the $i$-th row of the state transition matrix $\mathbf{A}$.

5. Draw a random duration $d_2$ from the duration probability distribution $D_j(d)$.

6. A sequence of $d_2$ iid observations $y_{d_1+1:d_1+d_2} = \{y_{d_1+1}, y_{d_1+2}, \ldots, y_{d_1+d_2}\}$ are emitted from state $j$ according to the emission probability distribution $b_j(y)$.

7. Continue steps 4 through 6 throughout the length of the process.

The backward-message passing procedure described in section 3.1.1 requires modification due to the Markovian nature of state-switching over random durations as opposed to each discrete time unit. The backward messages per state $i$ are broken up into two components $B_t(i)$ and $B_t^*(i)$ for convenience, as denoted in [13]. As mentioned earlier, the HSMM now incorporates a set of duration distributions associated with each state, so $\boldsymbol{\lambda} = \left(\boldsymbol{\pi}, \mathbf{A}, \mathbf{B}, \{D_i(\cdot)\}_{i=1}^N\right)$. In this assessment $x_t = i$ means that the process is in the $i$-th state at time $t$. The initialization step for each state essentially remains unchanged from the HMM case.

$$B_T(i) = 1, \quad 1 \leq i \leq N \tag{3.25}$$

The first component $B_t(i)$ of the backward message seen in (3.27) describes the probability of observing a partial observation sequence $y_{t+1:T}$ after a transition from state $i$ to state $j$ at time $t+1$, while the random variable $F_t = 1$ denotes that state $i$ has finished emitting its own observations at time $t$ and is ready to transition to some other state $j \neq i$.

$$B_t(i) = P(y_{t+1:T} | x_t = i, F_t = 1, \boldsymbol{\lambda}), \quad 1 \leq i \leq N \tag{3.26}$$

$$= \sum_{j=1}^N B_t^*(j) \, a_{ij} \tag{3.27}$$

The second component $B_t^*(i)$ of the backwards message is defined by (3.28).

$$B_t^*(i) = P(y_{t+1:T} | x_{t+1} = i, F_t = 1, \boldsymbol{\lambda}), \quad 1 \leq i \leq N \tag{3.28}$$

$$= \sum_{d=1}^{T-t} B_{t+d}(i) \cdot p(D_{t+1} = d | x_{t+1} = i, \boldsymbol{\lambda}) \cdot P(y_{t+1:t+d} | x_{t+1} = i, D_{t+1} = d, \boldsymbol{\lambda})$$

$$+ P(D_{t+1} > T - t | x_{t+1} = i, \boldsymbol{\lambda}) P(y_{t+1:T} | x_{t+1:T} = i, D_{t+1} > T - t, \boldsymbol{\lambda}) \tag{3.29}$$

The final term in (3.29) contains the probability that the sequence of interest has been right-censored in time, in which the final observation at time $T$ does not necessarily denote the last

emitted observation of the last visited state in the data[5] [9]. Since there is little other change to the backward-message passing scheme for HSMMs as opposed to their HMM counterpart, the number of computations remains on the order of $N^2 T$.

## 3.3  HSMMs Using Hierarchical Dirichlet Random Processes

Hidden semi-Markov models possess the same a priori assumption about the number of states $N$ present within the process being examined. Unfortunately, many situations arise in which there is no straightforward interpretation for the number of states present in the data beforehand, especially whenever additive noise arising the data collection process is highly nonlinear or time-variant. In the context of the experiments within Chapter 4, the highly dynamic nature in which wireless devices access the RF frequencies precludes such a possibility of knowing the discoverable state space with exact certainty.

Choosing initial characterizations for the remaining HSMM features can pose equally difficult challenges to finding the best model to fit the observed data. For example, statistical modeling for each emission distribution $b_i$ per state will also require additional parameters as well, which can hardly be known before performing the HSMM state sequence discovery process. The task of choosing the best HSMM matching the data observations would involve creating several HSMMs with different initial parameter sets, finding the optimal updated model parameters with the use of the forward-backward algorithm, and perform cross-validation across the different HSMMs to infer the optimal model matching the data. As an alternate approach, Bayesian statistics provide a more intuitive and natural extension for iteratively learning the HSMM parameters from a collection of observations.

### 3.3.1  Bayesian Foundations of the Algorithm

The more traditional frequentist interpretation of probability focuses on the formulation of a random variable's probability distribution through performing a large number of repeated experiments. Suppose that a random variable $Z$ associated with a set of possible outcomes $\mathcal{A} = \{z_1, z_2, \ldots, z_N\}$. The relative frequency of each event $z \in \mathcal{A}$ is thus constructed by counting the number of times $n_i$ in which $z$ occurs after $n$ repeated experiments. The probability distribution of $Z$ is constructed by taking the limit of the relative frequencies of each possible event $z_i$ as an infinite number of experiments have been conducted.

$$P(Z = z_i) = \lim_{n \to \infty} \frac{n_i}{n}, \quad i = 1, 2, \ldots, N \tag{3.30}$$

---

[5]It is implicitly assumed that the first data point in the sequences of interest do correspond to the first emitted observation of the associated state. However, the HSMM formulations can be modified to left-censored data in which this condition does not hold.

This method possesses obvious computational difficulties due to the impossibility for performing infinitely many experiments, each with infinitely precise control on the physical factors governing the experimental process itself at each performance. Furthermore, computations of joint probabilities involving multiple random variables can become equally cumbersome for these same reasons, especially if the random variables involved are not independent.

Suppose that the desired quantity of interest is to evaluate the joint probability distribution $P(Y, \theta)$ of two random variables $Y$ and $\theta$, where $Y$ is associated with an observation of some experiment and $\theta$ reflects the underlying parameters governing that observation.[6] Provided that $P(\theta) \neq 0$, the conditional distribution of $Y$ given direct knowledge of $\theta$ can be calculated according to Bayes' theorem in (3.31).

$$P(Y|\theta) = \frac{P(Y, \theta)}{P(\theta)} \tag{3.31}$$

Reapplication of Bayes' theorem can likewise be used to calculate the conditional distribution of $\theta$ given direct knowledge of $Y$ under similar non-zero conditions on $P(\theta)$.

$$P(\theta|Y) = \frac{P(Y, \theta)}{P(Y)} \tag{3.32}$$

By equating the joint probability distribution from both expressions, the conditional distribution $P(\theta|Y)$ can be expressed as the product of the conditional distribution $P(Y|\theta)$ and the marginal distribution $P(\theta)$ of the parameter $\theta$.

$$P(\theta|Y)P(Y) = P(Y|\theta) \, P(\theta) \tag{3.33}$$

$$P(\theta|Y) = \frac{1}{P(Y)} P(Y|\theta) \, P(\theta) \tag{3.34}$$

$$\propto P(Y|\theta) \, P(\theta) \tag{3.35}$$

The step from (3.34) and (3.35) is possible since the marginal distribution $P(Y)$ has no factor of $\theta$, and thus may be treated as a constant.[7] Instead of explicitly assuming a particular value for the parameter $\theta$, it is then possible to reflect the uncertainties about $\theta$ through a probability distribution and subsequently evaluate the resulting change in uncertainty about $\theta$ after observing a particular value of $Y$. This data-driven updating of model uncertainties is the fundamental principle in Bayesian probabilistic interpretations. Philosophically, the Bayesian approach is also more intuitive for analyzing observable data that cannot be repeatedly replicated in the sense of the frequentist approach.

The first term in (3.35) is known as the posterior distribution $P(\theta|Y)$, while $P(\theta)$ denotes the prior distribution. The distribution $P(Y|\theta)$ becomes a likelihood function when $Y$ is

---

[6]In the case of $\theta$ being a set of parameters, $P(Y, \theta)$ would obviously be a joint distribution of two or more random variables.

[7]Technically speaking, the expression in (3.32) is an unnormalized probability distribution, since $\int P(\theta|Y) \, dY \neq 1$. Since $\frac{1}{P(Y)}$ is a constant, this omitted factor has no influence for any potential value of $Y$ and thus does not significantly affect any statistical inferences based on this equation.

assumed to be a fixed value or values occuring from a set of observations. For a given likelihood function used for modeling the observed process, it would be beneficial to obtain a computationally efficient way for updating the posterior distribution $P(\theta|Y)$ after collecting values for $Y$. This effort is accomplished by choosing the prior distribution $P(\theta)$ to belong to the same family of distributions as the likelihood function $P(Y|\theta)$. Such distributions are considered to be conjugate distributions to each other, since the posterior distribution as calculated by (3.35) will result in a probability distribution that is functionally similar to the prior distribution $P(\theta)$.[8]

The usefulness of such a conjugacy relationship can be exploited when it is desired to repeatedly sample from the prior distribution $P(\theta)$ after collecting observations of $Y$. Suppose that $z_i$ is a sample value from a prior distribution on a sought parameter $Z \sim f(\lambda)$, where $f$ is the probability distribution over which uncertainty about $Z$ is modeled. After observing values of $Y$, the posterior distribution $P(\theta|Y)$ will thus result in a new probability density $Z' \sim f(\lambda')$, with $\lambda'$ generally being a simple algebraic modification of the original hyperparameter $\lambda$.[9] As each new observation of $Y$ is gathered, the process may be iterated indefinitely, yielding a continually refined model that reflects how each additional gathered data point decreases the initial uncertainty about the process's parameters $Z$. The newly updated posterior distributions may then be sampled as the new prior distribution for $\lambda$ upon observing new data about the underlying process, continually iterating this process to sequentially obtain more accurate estimates for the model parameters. This data-driven approach also reduces algorithmic sensitivity to initial parameter assumptions about the model through sequentially gathering more data.

### 3.3.1.1   Gibbs Sampling

According to this Bayesian approach to modeling HMMs or HSMMs, performing parameter inference would involve sampling from the joint prior distribution $P(\boldsymbol{\theta}) = P(\theta_1, \theta_2, \ldots, \theta_n)$ for the model parameters $\theta_i$ and subsequently updating the posterior distributions $P(\boldsymbol{\theta}|Y)$ for these parameters given the likelihood functions for the observed data. However, sampling from such joint distributions may become difficult if there are probabilistic interdependencies between the model parameters (i.e., the model parameters are not all marginally independent). Gibbs sampling allows the joint distribution to be sampled by treating the model parameters as themselves being a Markov chain and repeatedly sampling from the conditional distributions for the model parameters as a function of time $t$.[10]

---

[8]Explanations of the conjugacy relationships used in the HSMM algorithm of this thesis can be found in Appendix A.

[9]$\lambda$ is termed a hyperparameter of the Bayesian model, since it is some initial value used for modeling the probability distribution describing the primary parameter of interest.

[10]More details on Gibbs sampling may be found in [7, 10].

Let $P(\theta_j | \theta_{-j})$ represent the conditional distribution of $\theta_j$ given the remaining model parameters $\theta_{-j} = \{\theta_1, \theta_2, \ldots, \theta_{j-1}, \theta_{j+1}, \ldots, \theta_n\}$ for every element $j \in \{1, 2, \ldots, n\}$. The Gibbs sampling algorithm is summarized by the following steps :

1. Select initial values for the $n$ random variables $\theta_j$.

2. Repeat for $t = 1, 2, \ldots$ :
   For $k = 1, 2, \ldots, n$ :

$$\theta_k^{(t)} \sim P\left(\theta_k^{(t)} | \theta_1^{(t)}, \ldots, \theta_{k-1}^{(t)}, \theta_{k+1}^{(t-1)}, \ldots, \theta_n^{(t-1)}\right) \tag{3.36}$$

3. Repeat step 2 until the joint distribution $P(\boldsymbol{\theta}) = P(\theta_1, \ldots, \theta_n)$ does not change.

Such repeated sampling upon conditional distributions can only result in producing the true joint distribution if the Markov chain on the set of variables $\boldsymbol{\theta}$ as described in (3.36) is irreducible and aperiodic.[11] According to step 3, this process requires multiple iterations over $t$ of this sampling process before the Markov chain converges to its stationary distribution. The time required for such convergence is known as the burn-in time for the Gibbs sampler. As a result of the stochastic nature for generating the joint probability distributions $P(\boldsymbol{\theta})$, HMM and HSMM parameter inference within a Gibbs sampling framework requires multiple iterations over a given data set before achieving a certain accuracy in estimating the desired parameters.

### 3.3.2   The Dirichlet Distribution

By the fundamental axioms of probability, a discrete probability distribution, or probability mass function (PMF), $\overrightarrow{p} = [p_1, p_2, \ldots, p_N]$ with $N$ components satisfies the following properties listed in (3.37).

$$\begin{cases} 0 \leq p_i \leq 1, & \forall\, i \in \{1, 2, \ldots, N\} \\ \sum_{i=1}^{N} p_i = 1 \end{cases} \tag{3.37}$$

Random PMFs may be constructed by drawing from a Dirichlet distribution with concentration parameter $\overrightarrow{\alpha} = [\alpha_1, \alpha_2, \ldots, \alpha_N]$ defined as $\alpha_i > 0$ for $i \in \{1, 2, \ldots, N\}$. If $\alpha_0 \triangleq \sum_{i=1}^{N} \alpha_i$, the probability of drawing a specific PMF can be expressed with the density function given

---

[11]A Markov chain is considered irreducible if any state may be reached from any other state after some number of transitions. Such a chain is also aperiodic if returns to a state do not regularly occur in multiples of some finite number of transitions.

by (3.38).[12]

$$P(\overrightarrow{p}|\overrightarrow{\alpha}) = \frac{\Gamma(\alpha_0)}{\prod_{i=1}^{N}\Gamma(\alpha_i)} \prod_{i=1}^{N} p_i^{\alpha_i-1} \tag{3.38}$$

$$\propto \prod_{i=1}^{N} p_i^{\alpha_i-1} \tag{3.39}$$

The mean $\mathrm{E}[\overrightarrow{p}]$ and covariance of a draw from a Dirichlet distribution are given by (3.40) and (3.41).

$$\mathrm{E}[\overrightarrow{p}] = \frac{\overrightarrow{\alpha}}{\alpha_0} \tag{3.40}$$

$$\mathrm{Cov}[p_i, p_j] = \begin{cases} \frac{\alpha_i(\alpha_0-\alpha_i)}{\alpha_0^2(\alpha_0+1)}, & i = j \\ \frac{-\alpha_i\alpha_j}{\alpha_0^2(\alpha_0+1)}, & i \neq j \end{cases} \tag{3.41}$$

Note that the covariance $\mathrm{Cov}[p_i, p_j]$ is negative for $i \neq j$, since the $N$ components are not all independent resulting from the constraints in (3.37). If $\alpha_1 = \alpha_2 = \cdots = \alpha_N = c$ where $c > 0$ is a constant, the mean of the resulting PMFs will be a uniform distribution over all $N$ components. Likewise, $|\mathrm{Cov}[p_i, p_j]|$ for $i \neq j$ decreases as $c$ grows increasingly larger. As an example, choosing $\overrightarrow{\alpha} = [1, 1, 1]$ results in a uniform distribution over the $N - 1$-dimensional probability simplex determined by (3.37), with all resulting PMFs tending having equal probability of being drawn. In contrast, there will be a higher probability of drawn a uniformly distributed PMF by choosing $\overrightarrow{\alpha} = [15, 15, 15]$, since the covariance between PMF components will decrease accordingly despite having the same mean $\mathrm{E}[\overrightarrow{p}] = [1/3, 1/3, 1/3]$. As $c \to 0$, almost all weight is given to one component in the PDF, while $\overrightarrow{p}$ tends to the uniform distribution on any draw as $c \to \infty$. When the component weights of $\overrightarrow{\alpha}$ are not all equal, more mass on the drawn PMF will be assigned to the components higher values associated with the corresponding components in $\overrightarrow{\alpha}$.

As explained in Appendix A.4, the Dirichlet distribution serves as a conjugate prior to a multinomial likelihood function. The multinomial distribution described in (3.42) is used to calculate the probability of observing each potential outcome $i \in \{1, 2, \ldots, N\}$ occurring $n_i$ times after $n$ repeated trials of an experiment, where $p_i$ denotes the probability of observing an outcome of $i$ in one trial.

$$p(n_1, n_2, \ldots, n_N|\overrightarrow{p}, n) = \frac{n!}{n_1!n_2!\cdots n_N!} \prod_{i=1}^{N} p_i^{n_i} \tag{3.42}$$

In the context of the HSMM, the state of the model $x_t$ at any given time $t$ is itself an element of the set $\{1, 2, \ldots, N\}$. This fact suggests that the Dirichlet distribution can be used to update beliefs about the transition matrix $\mathbf{A}$ after the state sequence from an observation

---

[12]The gamma function $\Gamma(x)$ is defined as $\Gamma(x) = \int_0^{\infty} t^{x-1} \exp\{-t\}\, dt$ for $x > 0$.

set has been labeled according to the HSMM backward message-passing algorithm described in Section 3.2.

Furthermore, partitioning of the original state space $\{1, 2, \ldots, N\}$ into disjoint sets $\{A_1, A_2, \ldots, A_k\}$ will result in a Dirichlet distribution over the new state space. A more formal description of this property can be expressed as follows.

If $[p_1, p_2, \ldots, p_N] \sim \mathrm{Dir}\big([\alpha_1, \alpha_2, \ldots, \alpha_N]\big)$,

$$\text{then } \Big[ \sum_{i \in A_1} p_i, \sum_{i \in A_2} p_i, \ldots, \sum_{i \in A_k} p_i \Big] \sim \mathrm{Dir}\Big( \Big[ \sum_{i \in A_1} \alpha_i, \sum_{i \in A_2} \alpha_i, \ldots, \sum_{i \in A_k} \alpha_i \Big] \Big). \tag{3.43}$$

This aggregation property suggests that the Dirichlet distribution may be equally beneficial for HSMMs that attempt to flexibly model the state space cardinality within in the Bayesian context.

### 3.3.3 Dirichlet Random Processes

For the Dirichlet distribution considered thus far, a finite number of outcomes has been assumed in the sample space $\mathcal{X} = \{1, 2, \ldots, N\}$ when one trial of an experiment is considered, as implied by the finite number of components $N$ within $\overrightarrow{p}$ and $\overrightarrow{\alpha}$. In the Bayesian framework, constant addition of new observations introduces increasing degrees of freedom to the model under consideration, which can render finite-dimensional models for $\mathcal{X}$ inadequate for accurately modeling the bounded state space as more observations are added. As a result, it is worthwhile to extend the Dirichlet distribution to handle infinite sample spaces through what is known as the Dirichlet random process.[13]

For a measurable space $(\mathcal{X}, \mathcal{B})$ with sample space $\mathcal{X}$ and $\sigma$-algebra $\mathcal{B}$, it is worth considering if it is possible to assign a probability $P(B)$ to some subset $B \in \mathcal{B}$ and its associated sequence of points $\{x_k\}$ in the sample space satisfying (3.44).

$$P(B) = \sum_{k=1}^{\infty} p_k \delta_{x_k}, \quad \text{with } \delta_{x_k} = \begin{cases} 1, x_k \in B \\ 0, x_k \notin B \end{cases} \tag{3.44}$$

If $\alpha$ is a finite non-zero measure on $(\mathcal{X}, \mathcal{B})$, it is possible to create a random probability measure $P$ for any finite measureable partition $\{B_i\}_{i=1}^{N}$ on $\mathcal{X}$ satisying (3.45).

$$\alpha(B_i) = \int_{\mathcal{X}} \mathbb{1}_{B_i}(X) \, d\alpha(x), \quad \text{where } \mathbb{1}_{B_i}(x) = \begin{cases} 1, \ x \in B_i \\ 0, \ x \notin B_i \end{cases} \tag{3.45}$$

---

[13]For a brief description of some relevant elements concerning probability and measure theory relied upon by this section, one can consult Appendix B. More information on Dirichlet random processes may be found in [4], [5], [14], and [19].

Accordingly, the Dirichlet process (DP) with concentration parameter $\alpha$ creates a collection of random variables over the $\sigma$-algebra $\mathcal{B}$ where the probability of each element in the partition is a component drawn from a Dirichlet distribution according to (3.46).

$$\Big[P(B_1), P(B_2), \ldots, P(B_N)\Big] \sim \mathrm{Dir}\Big(\big[\alpha(B_1), \alpha(B_2), \ldots, \alpha(B_N)\big]\Big) \qquad (3.46)$$

From examining equations (3.43) and (3.46), the Dirichlet process can be thought of an an extension of the Dirichlet distribution to incorporate random variables whose index set becomes the $\sigma$-algebra $\mathcal{B}$. Thus, the DP allows a random measure $G$ to be drawn from some base measure $G_0$ given the concentration parameter $\alpha$.

$$G \sim \mathrm{DP}(\alpha, G_0) \qquad (3.47)$$

It follows from the definition of the Dirichlet process that independent draws from a DP will result in probability distributions over different partitions on the sample space $\mathcal{X}$, thus implying that there is zero probability of observing a unique sample from $\mathcal{X}$ common to each probability distribution. In the context of the HSMM, construction of an $N \times N$ state transition matrix $\mathbf{A}$ with each row being a DP realization would not link the states together as they transition, since the underlying partitions for each row's DP would be disjoint. A solution to this difficulty can be found by using one DP realization to create a base measure $\beta$, from which all transition matrix rows are subsequently drawn from independent DPs on $\beta$ subject to a separate concentration parameter $\alpha$. This technique extends the Dirichlet process to a hierarchical Dirichlet process (HDP), in which all rows are tied together on the same common base measure.

$$\beta|\gamma \sim \mathrm{Dir}(\gamma/N_{max}, \ldots, \gamma/N_{max}) \qquad (3.48)$$
$$\overrightarrow{a_i}|\beta, \alpha \sim \mathrm{DP}(\alpha\beta_1, \ldots, \alpha\beta_{N_{max}}) \qquad i = 1, 2, \ldots, N_{max} \qquad (3.49)$$

Since the Dirichlet process is defined to construct measures over an infinite sample space, the $N_{max}$ factor in (3.48) and (3.49) is required to computationally realize the HDP prior distributions in an algorithmic sense. This prior is known as a weak-limit approximation to the HDP, since it converges to a true HDP as $N_{max} \rightarrow \infty$.

# Chapter 4

# Application of the Bayesian HDP-HSMM Algorithm

The focus of this chapter is the implementation of the Bayesian HSMM model described in Chapter 3 towards identifying the transmissions of wireless RF devices. The algorithm's perfomance is tested using collected measurements from a USRP in the local proximity of communicating SDRs as they transmit data between each other. Specific portions of the algorithm discussed in [13] were modified to more reliably model key features of wireless communication scenarios from the perspective of wireless channel-accessing strategies.

## 4.1    Bayesian HDP-HSMM Algorithmic Description

The overall objective of the Bayesian inference algorithm in this thesis is to create an initial HSMM and use the model parameters to create a maximum a posterior (MAP) estimate for the HSMM labeled state sequence given the RF trace data provided to the algorithm. It is desired that each state within the HSMM correspond to distinct observed activities on the wireless channel, such as transmissions originating from a specific RF device, packet collisions between multiple RF devices, or vacant periods over the channel. Performing Bayesian inference in the HDP-HSMM framework is accomplished by combining the message-passing scheme from Section 3.2 with a generative Gibbs sampling approach for block-sampling for the state sequence conditional on the observed data points. Prior to performing the numerical computations, a maximum number of model states $N_{max}$ is used to bound the potential state space to a reasonable size. Likewise, the state labeling procedure is run for a pre-determined number of iterations $N_{iter}$ for the algorithm to converge towards a suitable labeling of the observed semi-Markov chain.

A generative approach for creating the semi-Markov chain described in Sections 3.2 within the HDP framework can be found in the following equations.

$$\beta|\gamma \sim \text{Dir}(\gamma/N_{max}, \ldots, \gamma/N_{max}) \tag{4.1}$$

$$\overrightarrow{a_i}|\beta, \alpha \sim \text{Dir}(\alpha\beta_1, \ldots, \alpha\beta_{N_{max}}) \qquad\qquad i = 1, 2, \ldots, N_{max} \tag{4.2}$$

$$\boldsymbol{\pi}|\rho \sim \text{Dir}(\rho/N_{max}, \ldots, \rho/N_{max}) \tag{4.3}$$

$$\theta_i|H \sim H \qquad\qquad i = 1, 2, \ldots, N_{max} \tag{4.4}$$

$$\omega_i|\Omega \sim \Omega \qquad\qquad i = 1, 2, \ldots, N_{max} \tag{4.5}$$

As the first step in the algorithm, a concentration hyperparameter $\gamma$ is used to create the base measure $\beta$ through a draw from a Dirichlet distribution.[1] In accordance with the concept of the hierarchical Dirichlet process, the transition matrix $\mathbf{A}$ is subsequently initialized through $N_{max}$ independent draws from the base measure created from (4.1). The third step is responsible for initializing the initial state distribution $\boldsymbol{\pi}$ as a draw from a Dirichlet distribution with concentration hyperparameter $\rho$. The duration parameters $\theta_i$ and emission parameters $\omega_i$ for each state are then drawn from the appropriate parameter spaces $H$ and $\Omega$ respectively.

At this point, the HSMM transition matrix $\mathbf{A}$ and initial state distribution $\pi$ have been created. The second fundamental task is to compute the backward messages $B_t(i)$ and $B_t^*(i)$ according to (3.27) and (3.29) for each state over the observation data set length $T$. The drawn duration parameters $\omega_i$, emission parameters $\theta_i$, and data observations $y_{1:T}$ are used to calculate the duration and emission probabilities according to the state-specific distribution functions $D_i(d)$ and $b_i(y_t)$ respectively over the entire data set.

As discussed in Section 4.1.2, the labeled state sequence at each data point is then obtained by block sampling conditional on the initial model parameters $(\mathbf{A}, \mathbf{B}, \boldsymbol{\pi}, \{D_i\}_{i=1}^{N_{max}})$ and the data set $y_{1:T}$. Using the conjugacy relationships, the initial model parameters are updated after labeling the HSMM state sequence. The termination of this step now completes one full iteration of the Bayesian HSMM algorithm. The entire process is repeated for the remaining $N_{iter} - 1$ iterations, with the exception of the initial model creation steps detailed in (4.1) through (4.5). A brief summary of this process is given in the following outline.[2]

1. Create an initialized HDP-HSMM $\boldsymbol{\lambda}^{(0)}$.

2. Set $m = 1$.

3. Calculate the backward messages $B_t()$ and $B_t^*()$ conditioned on the $\boldsymbol{\lambda}^{(m-1)}$ and observed data set $\{y_t\}_{t=1}^T$.

4. Block sample the labeled state sequence $\{x_t\}_{t=1}^T$ conditioned on $\boldsymbol{\lambda}^{(m-1)}$ and $\{y_t\}_{t=1}^T$ as discussed in Section 4.1.2.

5. Update the model hyperparameters based on the labeled state sequence $\{x_t\}_{t=1}^T$ as discussed in Section 4.1.3.

6. Increment $m$.

7. Generate an updated HDP-HSMM $\boldsymbol{\lambda}^{(m)}$.

8. Repeat steps 4-7 until $m > N_{iter}$.

---

[1] A description of how to generate Dirichlet distributions is given in Appendix C.

[2] When discussing the model parameters, the superscript notation $x^{(m)}$ is used to denote the model parameter $x$ at the $m$-th iteration of the algorithm.

### 4.1.1 Modeling of Duration and Emission Distributions

In accordance with the HSMM definition, each state $i$ is endowed with its own duration distribution $D_i(d)$. For the experiments of this thesis, each state's duration distribution is selected to belong to a family of Poisson distributions, each with mean parameter $\lambda_i$, as shown in (4.6). Related work in RF channel modeling has shown that sojourn times for observed wireless nodes and idle periods can be well modeled as a hyper-Erlang distribution for both stationary and non-stationary traffic [6]. Since the hyper-Erlang is closely related to the exponential distribution, the Poisson distribution seen in (4.6) is chosen to function as a discrete time counterpart of the exponential distribution.

$$D_i(d|\lambda_i) = \frac{\lambda_i^d \exp\{-\lambda_i\}}{d!}, \quad d \in 0, 1, 2, \ldots \tag{4.6}$$

The assumption for Poisson likelihood functions for the durations also implies that the gamma distribution can be chosen to draw values for the mean parameter $\lambda_i$ as seen in the conjugacy relationships discussed in Appendix A.1.

The emission distributions $f_i(\overrightarrow{y})$ for each state are similarly modeled as two-dimensional multivariate Gaussian distributions with mean $\overrightarrow{\mu_i}$ and covariance $\boldsymbol{\Sigma}_i$, as described in (4.7).

$$f_i(\overrightarrow{y}|\overrightarrow{\mu_i}, \boldsymbol{\Sigma}_i) = \frac{1}{2\pi|\boldsymbol{\Sigma}_i|} \exp\left\{ -\frac{1}{2}(\overrightarrow{y} - \overrightarrow{\mu_i})^T \boldsymbol{\Sigma}_i^{-1}(\overrightarrow{y} - \overrightarrow{\mu_i}) \right\} \tag{4.7}$$

$$\tag{4.8}$$

The two-dimensional multivariate Gaussian distribution is utilized to model the emissions corresponding to the in-phase and quadrature components received by a USRP observing an RF band of interest at discrete points in time. While a significant advantage for this choice of $f_i(\overrightarrow{y})$ is its conjugacy with a normal inverse Wishart distribution as detailed in Appendix A.3, a further incentive is provided for modeling additive noise over the wireless channel. In the absence of any received RF user's signals, samples for the in-phase and quadrature components of the received signal is often modeled as independent Gaussian random variables.

It should be noted that the received signal waveforms corresponding to the RF users do not generally demonstrate Gaussian time-domain characteristics. Nonetheless, the previous assumptions of Poisson-distributed durations and multivariate Gaussian-distributed emissions per state perform well for the purpose of HSMM state labeling as seen later in Section 4.2.

## 4.1.2 Block Sampling of the State Sequence

In order to perform the posterior inference in the Bayesian setting, it is necessary to construct a joint distribution for the model parameters and state sequence $\{x_t\}_{t=1}^{T}$ according to (4.9).

$$P(\{x_t\}, \{\theta_i\}, \{\overrightarrow{a_i}\}, \{\omega_i\} | \{y_t\}, H, \Omega, \alpha) \tag{4.9}$$

Since this joint distribution is difficult to characterize in terms of the marginal distributions, a Gibbs sampling approach is taken to iteratively sample from the following conditional distributions.

$$P(\{\theta_i\} | \{x_t\}, \{y_t\}, H)$$
$$P(\{\omega_i\} | \{x_t\}, \Omega)$$
$$P(\{\alpha_i\} | \{x_t\}, \alpha)$$
$$P(\{x_t\} | \{\theta_i\}, \{\omega_i\}, \{\alpha_i\}, \{y_t\})$$

While sampling from the first three conditional distributions is easily performed as shown in (4.1)-(4.5), this task is more involved for sampling the hidden state sequence for the HSMM. An efficient means for performing this task couples the backward message calculations according to the drawn model parameters and observation sequence $\{y_t\}$ with random draws from the duration distributions per state. The following procedure gives a pseudo-code representation of this process.

$$t := 1 \tag{4.10}$$

$$next\_state\_unsmoothed = \boldsymbol{\pi}^{(m-1)} \tag{4.11}$$

**while** $t \leq T$ : $\tag{4.12}$

    **for** $j = 1, 2, \ldots, N_{max}$ $\tag{4.13}$

      $log\_domain(j) = B_t^*(j) - \max B_t^*(j) \tag{4.14}$

    **end** $\tag{4.15}$

    **for** $j = 1, 2, \ldots, N_{max}$ $\tag{4.16}$

      $next\_state\_distribution(j) = \exp(log\_domain(j))$

        $* next\_state\_unsmoothed(j) \tag{4.17}$

    **end** $\tag{4.18}$

    $i \sim next\_state\_distribution \tag{4.19}$

    $duration\_prob \sim \mathrm{U}(0,1) \tag{4.20}$

    $d = 0 \tag{4.21}$

    **while** $duration\_prob > 0$ : $\tag{4.22}$

      $p\_d\_marginal = D_i(d+1) \tag{4.23}$

      **if** $p\_d\_marginal == 0$ $\tag{4.24}$

        $d = d + 1 \tag{4.25}$

        **continue** $\tag{4.26}$

      **end** $\tag{4.27}$

      **if** $t + d \leq T$ : $\tag{4.28}$

        $p\_d = exp(sum(LL(i, t : t + d)) + \beta_{t+d}(i) - \beta_t^*(i)) * p\_d\_marginal \tag{4.29}$

        $duration\_prob = duration\_prob - p\_d \tag{4.30}$

        $d = d + 1 \tag{4.31}$

      **end** $\tag{4.32}$

    **end** $\tag{4.33}$

    $x_{t:t+d-1} = i \tag{4.34}$

    $next\_state\_unsmoothed = \overrightarrow{a_i}^{(m-1)} \tag{4.35}$

    $t = t + d \tag{4.36}$

**end** $\tag{4.37}$

The first step of the procedure is to initialize a time counter to $t = 1$ and use the inferred initial state distribution $\boldsymbol{\pi}^{(m-1)}$ for selecting the first state in the model. A vector $log\_domain$ with $N_{max}$ components is created to store the backward message $B_t^*(i)$ calculated using the data observations $y_{t:T}$ for each potential state in the model. The equation (4.14) is formulated so that smaller values per component correspond to the most probable states responsible for producing the observations from time $t$ to $T$. The variable $next\_state\_distribution$ in (4.17)

thus modifies the unsmoothed estimate for the next state distribution to reflect how each state fits the observation sequence $y_{t:T}$ as indicated in the HSMM backward messages. The underlying state $x_t = i$ is then chosen by drawing from the newly modified next state distribution.

Since HSMM states persist for random periods of time, it is necessary to draw from the duration distribution $D_i(d)$ corresponding to state $i$. The steps from (4.20) to (4.34) are used to perform inverse transform sampling for the duration $d$ of the state $x_t$. In these equations, $LL$ is denotes the log-likelihood for the points in consideration using the Gaussian observation distribution for the selected state $i$. Since a state transition must necessarily occur within the HSMM framework, the unsmoothed next state distribution is selected as the transition matrix row $\overrightarrow{a_i}^{(m-1)}$ obtained from the last iteration of the model inference process. The timer is incremented by $d$ samples, and the process is repeated until the timer variable reaches the end of the data set at time $T$.

This block sampling procedure explains the necessity for several iterations of the Bayesian approach to construct an accurately inferred HSMM. Since the Bayesian approach adopts a stochastic means for inferring the model explaining the observation data, there is no globally optimal solution for the model. The sampling approach for creating the model's prior distributions will yield different initial conditions for initiating the inference algorithm, thus resulting in uniquely labeled state sequence and model parameters at the each performance. Achieving accurate results for the state labeling is necessarily dependent on performing enough iterations $M$ of the Gibbs sampling to achieve stationarity, while the conjugacy relationships permit updating the model parameters at each iteration to incrementally refine the model parameters estimates as the Gibbs sampling procedure approaches stationarity.

## 4.1.3   Updating the Model Parameters

The following sections detail how the model parameters are updated between subsequent iterations in the Bayesian HSMM algorithm. The relative ease of parameter updating is accomplished through the judicious choice of the conjugate prior distributions as discussed more thoroughly in Appendix A.

### 4.1.3.1   Updating the Emission Distributions

As indicated in (4.4), the mean vector $\overrightarrow{\mu_i}$ and covariance matrix $\boldsymbol{\Sigma}_i$ for observations produced by each state are initialized through draws from an emission parameter space $H$. Since each observation $\overrightarrow{y}$ is modeled by a multivariate Gaussian distribution, the prior distribution for $\overrightarrow{\mu_i}$ and $\boldsymbol{\Sigma}_i$ is chosen to be a normal inverse Wishart (NIW) distribution as given in (4.38).

$$\overrightarrow{\mu_i}^{(0)}, \boldsymbol{\Sigma}_i^{(0)} \sim \text{NIW}\left(\overrightarrow{\mu_{0,i}}^{(0)}, \kappa_i^{(0)}, \nu_i^{(0)}, \boldsymbol{\Lambda}_i^{(0)}\right), \quad i = 1, \ldots, N_{max} \tag{4.38}$$

The NIW distribution thus introduced four distinct hyperparameters into the model. The hyperparameters $\vec{\mu_0}$ and $\boldsymbol{\Lambda}$ are used to model the expected means per component and expected covariances between components respectively. Likewise, $\kappa_0$ and $\nu_0$ are used to model the variability with which $\vec{\mu_0}$ and $\boldsymbol{\Lambda}$ are respectively drawn from the NIW distribution. Increasing values of $\kappa_0$ lead to decreasing variance from the drawn mean vector $\vec{\mu}$. On the other hand, decreasing values for $\nu_0$ decreases the degrees of freedom for modeling $\boldsymbol{\Sigma}$, resulting in drawn values of $\boldsymbol{\Sigma}$ much closer to $\boldsymbol{\Lambda}$.

After labeling the state sequence $\{x_t\}_{t=1}^{T}$, the observations $\{\vec{y_t}\}_{t=1}^{T}$ are collected into distinct groups per model state $i = 1, \ldots N_{max}$. Supposing that $n_i$ observations belong to each model state $i$ on the $(m-1)$-th iteration, the hyperparameters are updated according to the following four equations of (4.39).[3]

$$\begin{cases} \kappa_i^{(m)} & = \kappa_i^{(m-1)} + n_i \\ \nu_i^{(m)} & = \nu_i^{(m-1)} + n_i \\ \overrightarrow{\mu_{0,i}^{(m)}} & = \left( \kappa_i^{(m-1)} / \kappa_i^{(m)} \right) \overrightarrow{\mu_{0,i}^{(m-1)}} + \left( n_i / \kappa_i^{(m)} \right) \bar{y} \\ \boldsymbol{\Lambda}_i^{(m)} & = \boldsymbol{\Lambda}_i^{(m-1)} + S + \left( \kappa_i^{(m-1)} \, n_i / \kappa_i^{(m)} \right) \left( \bar{y} - \overrightarrow{\mu_{0,i}^{(m-1)}} \right)^T \left( \bar{y} - \overrightarrow{\mu_{0,i}^{(m-1)}} \right) \end{cases} \quad (4.39)$$

New values for the mean vector $\vec{\mu_i}$ and covariance matrix $\boldsymbol{\Sigma}_i$ at iteration $m + 1$ are subsequently drawn for each state using the recalculated hyperparameters from the previous iteration $m$.

$$\overrightarrow{\mu_i}^{(m+1)}, \boldsymbol{\Sigma}_i^{(m+1)} \sim \mathrm{NIW}\left( \overrightarrow{\mu_{0,i}^{(m)}}, \kappa_i^{(m)}, \nu_i^{(m)}, \boldsymbol{\Lambda}_i^{(m)} \right), \quad i = 1, \ldots, N_{max} \qquad (4.40)$$

### 4.1.3.2 Updating the Duration Distributions

Draws from the duration parameter space $\Omega$ for each state are accomplished using the gamma distribution with shape parameter $k$ and scale parameter $\theta$ according to the following equation. Thus, the initialization of the duration distributions is accomplished by drawing distinct Poisson mean parameters $\lambda_i$, with one mean parameter characterizing each state.

$$\lambda_i^{(0)} \sim \mathrm{Gamma}\left( k_i^{(0)}, \theta_i^{(0)} \right), \quad i = 1, \ldots, N_{max} \qquad (4.41)$$

The hyperparameter $k$ models the shape of the gamma distribution, while $\theta$ represents the scale or variability with which $\lambda$ is drawn. As a result, increasing value of $\theta$ will result in more statistical dispersion in the drawn values of $\lambda$. Since it is difficult to visualize how $k$ affects changes in $\lambda$, it is worthwhile to note that the expected value for the drawn Poisson mean parameter is $\mathrm{E}[\lambda_i] = k_i \, \theta_i$, with variance of $\mathrm{Var}[\lambda_i] = k_i \, \theta_i^2$.

---

[3]In the interest of space and maintaining relevancy, consult Appendix A.3 for definitions of the quantities $S$ and $\bar{y}$ respectively.

Suppose that each state $i$ is labeled is visited $n_i$ times in $(m-1)$-st iteration of labeling the superstate sequence, with each visit lasting for $d_j$ time units. The hyperparameters $k$ and $\theta$ may be easily updated at each iteration according to the following equations in (4.42).

$$\begin{cases} k_i^{(m)} &= k_i^{(m-1)} + \sum_{j=1}^{n_i} d_j \\ \theta_i^{(m)} &= \theta_i^{(m-1)}/\left(1 + n_i\,\theta_i^{(m-1)}\right) \end{cases} \tag{4.42}$$

### 4.1.3.3 Updating the State Transition Matrix

The initial creation of the state transition matrix $\mathbf{A}$ in the Bayesian HSMM algorithm is the result of repeatedly drawing from a base measure $\beta^{(0)}$ to construct all $N_{max}$ rows of $\mathbf{A}$ as detailed in (4.43) and (4.44) .

$$\beta^{(0)}|\gamma \sim \mathrm{Dir}(\gamma/N_{max}, \ldots, \gamma/N_{max}) \tag{4.43}$$

$$\overrightarrow{a_i}^{(0)}|\gamma, \beta^{(0)}, \alpha \sim \mathrm{Dir}\left(\alpha\beta_1^{(0)}, \ldots, \alpha\beta_{N_{max}}^{(0)}\right), \quad i = 1, \ldots, N_{max} \tag{4.44}$$

From examining the conjugacy of the Dirichlet distribution with the multinomial likelihood in which states appear within the labeled state sequence $\{x_t\}_{t=1}^T$, one would expect that updating the state transition matrix can be easily performed. There are subtle difficulties introduced by the requirement for zero elements along the main diagonal of $\mathbf{A}$ in a HSMM formulation. As detailed more explicitly in [13], the conjugacy relation can be restored through the introduction of auxiliary random variables; a brief discussion of this process is given below.

After labeling the superstate sequence $\{z_s\}_{s=1}^S$ with $S$ visited states from the $(m-1)$-st iteration of the algorithm, suppose that there are $n$ transitions out of state $i$, with $n_j$ of those to state $j$ for $j = 1, \ldots, N_{max}$ such that $\sum_{j=1}^{N_{max}} n_j = n$. As a means to restore conjugacy, $n$ auxiliary random variables $\{r_{i,k}\}_{k=1}^n$ are independently drawn from a geometric distribution $r_{i,k} \overset{iid}{\sim} \mathrm{Geo}(1 - a_{ii})$ with success parameter $1 - a_{ii}$ as given in (4.45).

$$P(r_{i,k}|a_{ii}) = a_{ii}^{r_{i,k}}(1 - a_{ii}), \quad r_{i,k} \in \{0, 1, \ldots\} \tag{4.45}$$

After using these auxiliary random variables, the base measure $\beta^{(m)}$ is updated, from which each row of the transition matrix $\overrightarrow{a_i}^{(m)}$ can be ascertained through the hierarchical Dirichlet process drawing procedure as described below.

$$\overrightarrow{a_i}^{(m)}|\gamma, \beta^{(m-1)}, \alpha, \{z_s\}_{s=1}^S \sim \mathrm{Dir}\left(\alpha\left(\beta_1^{(m-1)} + n_1\right), \ldots, \alpha\left(\beta_i^{(m-1)} + \sum_{k=1}^n r_{i,k}\right),\right.$$

$$\left. \ldots, \alpha\left(\beta_{N_{max}}^{(m-1)} + n_{N_{max}}\right)\right), \quad i = 1, \ldots, N_{max} \tag{4.46}$$

### 4.1.3.4 Updating the Initial State Distribution

Creation of the initial state distribution $\boldsymbol{\pi}$ is easily accomplished by sampling from a Dirichlet distribution with a concentration parameter having weights of $\rho/N_{max}$ along all $N_{max}$ components as seen in (4.47).

$$\boldsymbol{\pi}^{(0)} \sim \text{Dir}(\rho/N_{max}, \ldots, \rho/N_{max}) \tag{4.47}$$

The conjugacy relationship between the Dirichlet prior on $\boldsymbol{\pi}$ and a multinomial likelihood allows for a simple updating procedure for this model parameter. Suppose $z_1 = i$ is the first visited superstate upon the $(m-1)$-st iteration of the state labeling process. The updated initial state distribution is obtained by sampling $\boldsymbol{\pi}^{(m)} \sim \text{Dir}(\overrightarrow{\alpha})$, with only the component for the $i$-th state being affected as gien in (4.48).

$$\alpha_j = \begin{cases} \rho/N_{max}, & j \neq i \\ (\rho/N_{max}) + 1, & j = i \end{cases} \tag{4.48}$$

## 4.1.4 Modifications to the Algorithm

### 4.1.4.1 Hyperparameter Selection

A naive application of the Bayesian HSMM model may not necessitate endowing each model state with distinct hyperparameters for the observation and duration distributions. Since the application of the HSMM algorithm is used to classify RF users and potential packet collisions from received wireless traces, it would be more beneficial to choose these hyperparameters to reflect some of the basic features with which wireless RF devices typically behave over the course of their transmissions. Although knowledge of each RF device in terms of transmission durations and signal energies cannot be well-known before of the data collection process, it is worthwhile to examine the statistical properties for idle time over the wireless channel medium in which no users are present.

Suppose that the wireless received signal $r[n]$ at discrete instances of time can be represented as the linear superposition of several signal sources $s_i[n]$ in addition to a noise component $n[n]$. Expressed in terms of its in-phase and quadrature components, the received signal can be presented as in (4.51).

$$r[n] = \sum_i s_i[n] + n[n] \tag{4.49}$$

$$= \sum_i \left( s_{i,I}[n] + js_{i,Q}[n] \right) + \left( n_I[n] + jn_Q[n] \right) \tag{4.50}$$

$$= \left( \sum_i s_{i,I}[n] + n_I[n] \right) + j \left( \sum_i s_{i,Q}[n] + n_Q[n] \right) \tag{4.51}$$

In the absence of any RF transmitter, the lone contribution to the received signal at each point is time is solely characterized by the noise added over the wireless channel. A common model for such noise is as additive white Gaussian noise, where both $n_I[n]$ and $n_Q[n]$ are treated as independently drawn from zero-mean univariate Gaussian distributions with variances $\sigma_I^2$ and $\sigma_Q^2$ respectively. As a result, the covariance matrix between the in-phase and quadrature components of the noise can be represented by (4.52).

$$\text{Cov}[n_I, n_Q] = \begin{bmatrix} \sigma_I^2 & 0 \\ 0 & \sigma_Q^2 \end{bmatrix} \tag{4.52}$$

Simulating parameters of such additive white Gaussian noise for the idle state, with the assumption of $\sigma_I^2 = \sigma_Q^2 = 1$, can be accomplished through setting the NIW hyperparameters of (4.38) so that $\overrightarrow{\mu_0} = \overrightarrow{0}$ and $\mathbf{\Lambda}_0 = \mathbf{I}$.[4] By selecting higher values of $\kappa_0$ and lower values of $\nu_0$ for the idle state hyperparameters as opposed to those for the remaining model states, the initialized parameters for the idle state will be more reflective of an AWGN channel than a state corresponding to transmissions from one or more RF devices. Thus if each state in the HSMM is to be tied to a specific RF device or potential packet collisions, varying the degrees of freedom $\nu_0$ with which the covariance matrices $\mathbf{\Sigma}_i$ are drawn will reflect varying degrees of correlation between the in-phase and quadrature signal components.

Likewise, choosing distinct values of the duration hyperparameters $k$ and $\theta$ for the idle state can be used to reflect the relative congestion of the wireless channel over time. Since $\text{E}[d] = k\theta$ is the expected state duration for such hyperparameters, choosing smaller values of $k$ may be used to initialized the idle state's duration to model shorter time periods between observed transmissions. In the context of the experiments within section 4.2, the shape hyperparameter $k$ of the idle state was chosen to be smaller than those used to model the RF device's transmission periods.

### 4.1.4.2 Staging the Inference Process

Repeated application of the discussed Bayesian HSMM algorithm was seen to result in labeled state sequences that frequently involved very short superstate durations for some model states, sometimes comprising durations of 1 to 2 samples for such state occurrences. Since it is highly unlikely that such a small number of samples could correspond to RF transmissions, the iterative process for state labeling was subdivided into two distinct stages to mitigate such fast state-switching within the model. The number of iterations $N_{iter}$ was broken up into one stage of $N_1$ iterations, followed by a subsequent block of $N_2$ iterations using only the model states possessing Poisson mean durations above a small predetermined limit of $\lambda_{max} = 5$ as follows.

---

[4] $\overrightarrow{0}$ here represents a two-dimensional vector with both components equal to zero, while $\mathbf{I}$ represents a $2 \times 2$ identity matrix.

At the end of the $(N_1)$-st iteration, all states with $\lambda_i \leq \lambda_{max}$ were removed from the model, with the transition matrix $\mathbf{A}^{(N_1)}$ and initial state distribution $\boldsymbol{\pi}^{(N_1)}$ renormalized to ensure that the probability spaces for $\mathbf{A}$ and $\boldsymbol{\pi}$ remain intact. Next, the labeled state sequence $\{x_t\}^{(N_1)}$ and superstate sequence $\{z_t\}^{(N_1)}$ were relabeled to include only states within in the newly modified HSMM state space. Properly relabeling the state sequences after such model reduction required reapplication of the backward messages to accurately refit the model to those necessary portions of the state sequence.

Suppose $z_s$ is a superstate having duration $d_s$ is marked for deletion, with previous state $z_{s-1} = i$ of duration $d_{s-1}$ and next state $z_{s+1} = j$ lasting $d_{s+1}$. The procedure for relabeling the state sequences are as follows.

1. Calculate the log-likelihoods for:

   (a) $\overrightarrow{y}_{(t-d_{s-1}-1):(t+d_s-1)}$

   (b) $\overrightarrow{y}_{t:(t+d_s+d_{s+1}-1)}$

2. Calculate the duration probabilities for:

   (a) state $i$ lasting for $d_s + d_{s-1}$

   (b) state $j$ lasting for $d_s + d_{s+1}$

3. Using steps 1 and 2, calculate the backward messages with altered truncation limits for $T$.

   (a) $\beta^*_{t-d_{s-1}-1}(i)$ with $T = t + d_s - 1$

   (b) $\beta^*_t(j)$ with $T = t + d_s + d_{s+1} - 1$

4. $z_s$ is labeled as the state with the larger $\beta^*$ quantity.

5. The state duration $d_s$ and associated labels within $\{x_t\}$ are updated accordingly.

The updated state sequences and durations were then used in the resampling procedure to start the $(N_2)$-st iteration, after which the Bayesian HSMM algorithm continued as previously described. In order to create the model to initiate the second stage, the concentration parameters $\alpha, \gamma$, and $\rho$ are each readjusted to equal the number of states detected at the end of the first stage of the algorithm. Performing enough iterations for the second stage of the algorithm mitigates the inexactitude with which these parameters are chosen, since the algorithm would then undergo enough Gibbs sampling steps to converge to its best estimate of the model parameters.

## 4.2 Experimental Results

### 4.2.1 USRP Transmission Analysis

Analyzing the Bayesian HSMM performance for detecting wireless RF devices was accomplished by programming several USRPs to transmit data according to semi-Markovian behavior as implemented through custom Python programs using GNU Radio.[5] The Python programs enable two USRPs to coordinate their activity through the host PC so that no packet collisions are produced as they transmit data over the program's execution. A third USRP was then utilized to collect wireless RF traces of the generated activity to use as inputs to the Bayesian HSMM algorithm.

The coordinated behavior of the two USRPs is programmed to simulate wireless networks acting in perfect accordance under a medium access protocol to prevent data collisions for users within the network. Further experiments introduce a third transmitting USRP enacting its own semi-Markovian behavior in an autonomous manner, where this USRP is allowed to access and vacate the wireless channel irrespective of the other two coordinated USRPs. Thus, packet collisions between the coordinated USRPs and the autonomous USRP are introduced to the wireless network modeling. It is the goal of the Bayesian HSMM algorithm to identify the number of devices present in each collected RF trace by examining the statistical properties of the received signal over time and to also identify collision instances when two USRPs attempt to transmit data packets simultaneously.

#### 4.2.1.1 Implementing HSMM Transmissions on Multiple Radios

The procedure for implementing two coordinated USRPs under a common semi-Markov chain is outline in (4.53) through (4.82) on the next page. The construction begins with generating an alternating sequence of busy states (denoted as 1's) and idle states (denoted as 0's). During each idle state, program execution is temporarily halted, or "slept", for a random amount of seconds as drawn from an exponential distribution $D_{off} \sim \text{Exp}(lambda)$, where $lambda$ denotes the mean of the exponential distribution.[6] Drawn sleep durations are chosen to be truncated within pre-determined limits of $d\_min$ and $d\_max$ to model a realistic idle time with no transmissions over the wireless channel. Periods between packet transmission can neither occur instantaneously under a well-designed wireless channel access protocol, nor can these periods be excessively long for highly utilized portions of the 2.4 GHz ISM band.

---

[5]The Python programs used to implement the following experiments, along with all other programs used within this research, may by acquired at the web address specified in Appendix D.2.

[6]An exponential distribution with rate parameter $\alpha$ has a distribution of the form $P(d|\alpha) = \alpha \exp\{-\alpha d\}$ for $x, \alpha > 0$. Hence, $\text{E}[D_{off}] = 1/\alpha$. Note that the notation $\text{Exp}(\cdot)$ used in this section denotes the *mean* of this distribution within the parentheses for ease of interpretation in the experimental setting.

For each busy state, a random variable $x$ is drawn from a uniform distribution and compared to a predetermined probability $USRP\_prob$ that determines the relative frequency with which USRP 1 is selected to transmit data for a generated busy state. After the appropriate USRP is chosen, another random variable $z$ is drawn from a uniform distribution. Draws of $z$ are used to select two linked parameters for the chosen USRP: the number of packets sent $num\_pkts$ and the size of each packet $pkt\_size$ in bytes. For example, setting the packet emission probability vector $PE\_prob = [1/2, 1/2]$ with $num\_pkts = [5, 10]$ and $pkt\_size = [50, 75]$ means that a busy state will send 5 packets of 50 bytes with probability $1/2$, and likewise for sending 10 packets of 75 bytes.

Each USRP is allowed to employ either distinct or identical modulation schemes for encoding the binary data within each packet. The true time duration of each USRP's transmissions over the channel will thus be a function of the selected symbol bandwidth or bit rate. The transmission gain with which each USRP sends information over the channel is pre-selected before execution time to model a more realistic distance between each transmitter and the observing USRP within the laboratory setting.

$$x \sim \text{U}(0, 1) \tag{4.53}$$

**if** $x \leq 0.5$ : $\tag{4.54}$

$\quad superstate(1) = 0$ $\tag{4.55}$

**else** : $\tag{4.56}$

$\quad superstate(1) = 1$ $\tag{4.57}$

**for** $i = 2 : num\_states$ : $\tag{4.58}$

$\quad$ **if** $superstate(i-1) == 0$ : $\tag{4.59}$

$\quad\quad superstate(i) = 1$ $\tag{4.60}$

$\quad$ **else** : $\tag{4.61}$

$\quad\quad superstate(i) = 0$ $\tag{4.62}$

**end** $\tag{4.63}$

<br>

**for** $i = 1 : num\_states$ $\tag{4.64}$

$\quad$ **if** $superstate(i) == 0$ : $\tag{4.65}$

$\quad\quad d \sim \text{Exp}(lambda)$ $\tag{4.66}$

$\quad\quad$ **if** $d < d\_min$ : $\tag{4.67}$

$\quad\quad\quad d = d\_min$ $\tag{4.68}$

$\quad\quad$ **elseif** $d > d\_max$ : $\tag{4.69}$

$\quad\quad\quad d = d\_max$ $\tag{4.70}$

$\quad\quad$ sleep for $d$ seconds $\tag{4.71}$

$\quad$ **else** : $\tag{4.72}$

$\quad\quad y \sim \text{U}(0, 1)$ $\tag{4.73}$

$\quad\quad$ **if** $y \leq USRP\_prob$ : $\tag{4.74}$

$\quad\quad\quad$ USRP 1 chosen $\tag{4.75}$

$\quad\quad$ **else** : $\tag{4.76}$

$\quad\quad\quad$ USRP 2 chosen $\tag{4.77}$

$\quad\quad z \sim \text{U}(0, 1)$ $\tag{4.78}$

$$j = \min \left\{ k : \sum_{l=1}^{k} PE\_prob(l) \geq z \right\} \tag{4.79}$$

$\quad\quad$ chosen USRP sends $num\_pkts(j)$ packets of $pkt\_size(j)$ bytes $\tag{4.80}$

$\quad i = i + 1$ $\tag{4.81}$

**end** $\tag{4.82}$

#### 4.2.1.2 Collecting RF Traces Using the USRP

Collection of the wireless traces over the ISM band was performed by running the USRP Python program given in Appendix D.1. The inputs to this program specify the center frequency of interest, the sampling rate with which the received signal appearing at the antenna are digitized by the USRP's ADC, an identifier for the USRP device acting as receiver, and a name for the output file into which the digitized samples are stored in binary format. The binary data within the output file was then read into MATLAB as complex samples, with each the real and imaginary parts of each sample being stored in double format. These in-phase and quadrature components of the received signal serve as the two components of $\overrightarrow{y}$ at each time instant in the Bayesian HSMM algorithm.

Due to the large amount of data stored within the files at the utilized sampling rate of 500 kHz for these experiments, the data was subsequently downsampled to a rate of 1 kHz to allow the Bayesian HSMM algorithm to be conducted in reasonable amounts of time. Although the downsampling operation implies that the information embedded within the RF device transmission would be unrecoverable due to noncompliance of the Shannon-Nyquist sampling theorem, the focus of this thesis is merely to identify the RF devices and potential data collisions over the course of their operation. While decreasing the downsampling factor would also allow the transmission and idle durations to be inferred with greater precision, inherent limitations of the HSMM algorithm's processing time dictate that a compromise between increased timing accuracy and more expedient data analysis should be sought in this application.

#### 4.2.1.3 Two Coordinated OFDM Users

The first considered experiment assesses the Bayesian HSMM algorithm's ability to discern between two coordinated USRPs in a "perfect" transmission environment in which no collisions may occur. Both USRPs were chosen to send OFDM symbols with BPSK as the underlying modulation, as detailed in Table 4.1. As noted, the major difference between the two transmitters is reflected in the transmission power.

Table 4.1: USRP Generation of Experiment 1

|  | USRP 1 | USRP 2 |
|---|---|---|
| Modulation | OFDM-BPSK | OFDM-BPSK |
| Symbol Bandwidth | 250 kHz | 250 kHz |
| Number of Subcarriers | 200 | 200 |
| Packet Size | 508 bytes | 508 bytes |
| Packet Bunches | [5, 10, 15] | [5, 10, 15] |
| Packet Emission Probs. | [1/2, 1/2, 0] | [1/2, 1/2, 0] |
| Transmission Gain | 10.0 | 20.0 |
| Idle State Duration | $D_{off} \sim \text{Exp}(0.2)$ sec, $D_{off} \in [0.3, 0.5]$ sec | |

The initial hyperparameters and other required inputs to the algorithm are also summarized in the following outline.[7]

**Initial Hyperparameters**

- $N_{max} = 8$

- State 1: $\lambda \sim \text{Gamma}(k = 10, \theta = 5)$; $\overrightarrow{\mu}, \Sigma \sim \text{NIW}\Big(\overrightarrow{\mu_0} = \overrightarrow{0}, \Lambda = \mathbf{I}, \kappa = 5,$
  $\nu = 3\Big)$

- States 2-8: $\lambda_i \sim \text{Gamma}(k = 20, \theta = 5)$; $\overrightarrow{\mu_i}, \Sigma_i \sim \text{NIW}\Big(\overrightarrow{\mu_0} = \overrightarrow{0}, \Lambda = \mathbf{I}, \kappa = 8,$
  $\nu = 3\Big)$

- $\alpha = \gamma = \rho = 3$

- $N_1 = 50$ iterations

- $N_2 = 50$ iterations

Figure 4.1 depicts the labeled state sequence after the final iteration of the algorithm, along with the sample magnitudes for each point in the RF trace. A legend for mapping each state to its corresponding can be found in the following bullets. At the completion of the first $N_1$ iterations, the resultant state sequence used only 4 model states and likewise after the final iteration. As seen in Figure 4.1, the algorithm was able to discern the idle state within the RF trace with very good accuracy. Instances of USRP 2's transmissions were well reflected by the state sequence due to the large difference in transmission power with respect to USRP

---

[7]A more detailed list of the inferred experimental results from the HSMM algorithm regarding all of the examples in this chapter may be found in Appendix E.

1. On the other hand, fluctuations in the received power of USRP 1's signal resulted in the creation of two states to model the device's behavior.

- State 1 (Dark Blue): Idle State, having mean parameter $\lambda_1 = 98.059738$

- State 2 (Cyan) : USRP 1, having mean parameter $\lambda_2 = 125.038564$

- State 3 (Green): USRP 2, having mean parameter $\lambda_3 = 141.050548$

- State 5 (Orange): USRP 1, having mean parameter $\lambda_4 = 86.090321$

The inferred state transition matrix $\mathbf{A}$ over the four-dimensional state space reveals that the non-zero elements within the first column are indeed close to 1.[8] In other words, the transition matrix reflects that there is overwhelming probability that the idle state will follow any transmissions by either USRP. The inferred model thus seems to indicate that the observed transmissions over the channel indeed correlate to a well-controlled wireless channel for coordinating users, since there is very little probability that two users' transmissions would immediately succeed each other in time.

$$\mathbf{A} = \begin{bmatrix} & 1 & 2 & 3 & 5 \\ 1 & 0.000000 & 0.610113 & 0.378314 & 0.011573 \\ 2 & 0.901059 & 0.000000 & 0.095198 & 0.003742 \\ 3 & 0.998246 & 0.000217 & 0.000000 & 0.001537 \\ 5 & 0.997800 & 0.001911 & 0.000289 & 0.000000 \end{bmatrix} \tag{4.83}$$

#### 4.2.1.4 Two Coordinated OFDM Users with OFDM Interference

The second experiment was conducted to assess how well the HSMM algorithm is able to discern distinct users and packet collisions for all 3 USRPs employing similar OFDM modulation schemes as detailed in Table 4.2.[9] The initial hyperparameters given to the algorithm are listed in the following outline, with only $N_{max}$ being altered with respect to the first experiment.

---

[8]The first row and columns of the inferred transition matrices considered in the experimental sections are used to label the state numbers for ease of readability.

[9]In the laboratory setup for the remainder of the experiments, the interfering USRP was positioned closer to the observing USRP collecting data measurements. Thus, a value of 0.0 for the transmission gain of USRP 3 does not indicate that the received signal energy from USRP 3 is lower than those for USRPs 1 and 2.
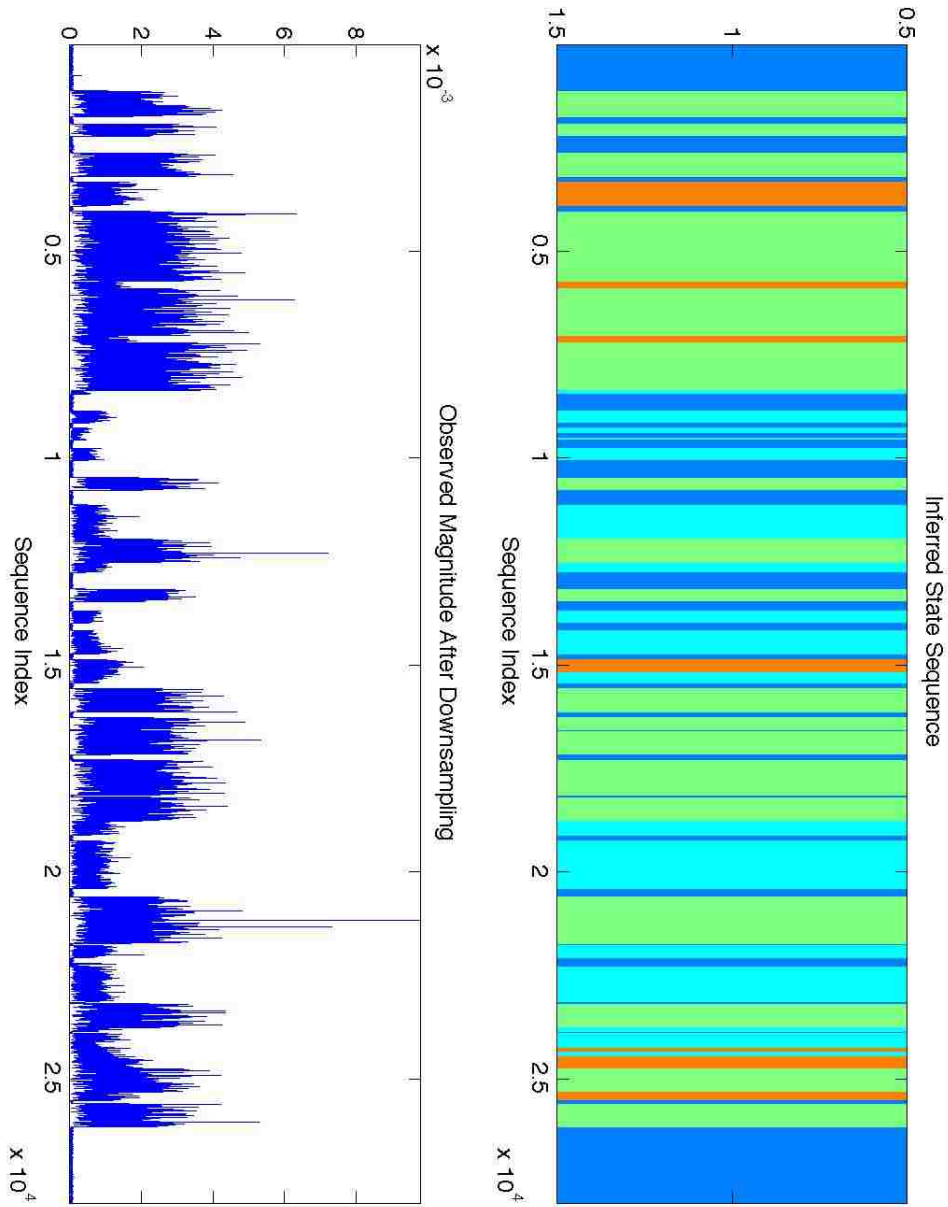
Figure 4.1: Labeled State Sequence for Experiment 1. Dark Blue = Idle State, Orange and Cyan = USRP 1, Green = USRP 2.

Table 4.2: USRP Generation of Experiment 2.

| | Coordinated USRPs | | Interfering USRP |
|---|---|---|---|
| | USRP 1 | USRP 2 | USRP 3 |
| Modulation | OFDM-BPSK | OFDM-BPSK | OFDM-BPSK |
| Symbol Bandwidth | 500 kHz | 250 kHz | 500 kHz |
| Number of Subcarriers | 200 | 200 | 200 |
| Packet Size | 508 bytes | 508 bytes | 508 bytes |
| Packet Bunches | [5, 10, 15] | [5, 10, 15] | [5, 10, 15] |
| Packet Emission Probs. | [1/2, 1/2, 0] | [1/2, 1/2, 0] | [1/2, 1/2, 0] |
| Transmission Gain | 25.0 | 20.0 | 0.0 |
| Idle State Duration | $D_{off} \sim \text{Exp}(0.3)$ sec $D_{off} \in [0.4, 0.9]$ sec | | $D_{off} \sim \text{Exp}(0.2)$ sec, $D_{off} \in [0.3, 0.7]$ sec |

**Initial Hyperparameters**

- $N_{max} = 6$

- State 1: $\lambda \sim \text{Gamma}(k = 10, \theta = 5)$; $\overrightarrow{\mu}, \boldsymbol{\Sigma} \sim \text{NIW}\left(\overrightarrow{\mu_0} = \overrightarrow{0}, \boldsymbol{\Lambda} = \mathbf{I}, \kappa = 5, \nu = 3\right)$

- States 2-6: $\lambda_i \sim \text{Gamma}(k = 20, \theta = 5)$; $\overrightarrow{\mu_i}, \boldsymbol{\Sigma}_i \sim \text{NIW}\left(\overrightarrow{\mu_0} = \overrightarrow{0}, \boldsymbol{\Lambda} = \mathbf{I}, \kappa = 8, \nu = 3\right)$

- $\alpha = \gamma = \rho = 3$

- $N_1 = 50$ iterations

- $N_2 = 50$ iterations

The legend for the labeled state sequence presented in Figure 4.2 in terms of the 4 resultant states is presented as follows.

- State 1 (Dark Blue): USRP 2, with mean duration parameter $\lambda_1 = 69.081473$

- State 2 (Cyan): USRP 1 and USRP 2/USRP 3 Collisions, with mean duration parameter $\lambda_2 = 142.328302$

- State 3 (Green): USRP 3 and USRP1/USRP 3 Collisions, having mean duration parameter $\lambda_3 = 153.371946$

- State 4 (Yellow): Idle State, with mean duration parameter $\lambda_4 = 77.665725$

From examining the labeled state sequence, it is apparent that idle states within the data set once again accurately fit periods of no activity, while collisions between the coordinated users and USRP 3 are also detected well as noted by abrupt state transitions in the middle of packet bursts. However, packet collisions between USRP 2 and USRP 3 are labeled the same as collision-free transmissions from USRP 1. Packet collisions between USRP 1 and 3 are also labeled indistinguishably from collision-free transmissions USRP 3. Thus the signal mixing that occurs during packet collisions is not distinct enough to warrant creations of new states that may indicate the specific pairs of users involved in such packet collisions.

The inferred state transition matrix given in (4.84) also seems to a less apparent degree of channel access coordination within the RF trace compared to that in section 4.2.1.3, as seen by examining the 4th column of $\mathbf{A}$ related to the idle state. On the other hand, the relative busy-idle duty cycles for each semi-Markov chain and the short transmission durations for the interfering USRP seem to be coupled so that many transitions from channel occupancy to channel vacancy occur in the overall RF trace. Thus examining the column of $\mathbf{A}$ corresponding to the labeled idle state is not sufficient for assessing the overall coordination of the channel accessing between users, since many collision instances are nonetheless observed within this RF trace.

$$\mathbf{A} = \begin{bmatrix} & 1 & 2 & 3 & 4 \\ 1 & 0.000000 & 0.377650 & 0.100816 & 0.521534 \\ 2 & 0.036572 & 0.000000 & 0.150152 & 0.813277 \\ 3 & 0.224202 & 0.000028 & 0.000000 & 0.775770 \\ 4 & 0.610809 & 0.043805 & 0.345386 & 0.000000 \end{bmatrix} \tag{4.84}$$

### 4.2.1.5   Two Coordinated OFDM Users with Low-Power OFDM Interference

The third performed experiment generates a semi-Markov chain with the associated USRP parameters given in Table 4.3. By comparison, the experimental setup is very similar to that in the previous section, but with a much lower transmission power and typically longer durations between transmissions for the interfering USRP. The initial hyperparameters given in the subsequent bullets are once again unaltered, with the exception of $N_{max}$ decreased to 5.

**Initial Hyperparameters**

- $N_{max} = 5$

- State 1: $\lambda \sim \text{Gamma}(k = 10, \theta = 5)$; $\overrightarrow{\mu}, \boldsymbol{\Sigma} \sim \text{NIW}\left(\overrightarrow{\mu_0} = \overrightarrow{0}, \boldsymbol{\Lambda} = \mathbf{I}, \kappa = 5,\right.$
  $\left. \nu = 3\right)$
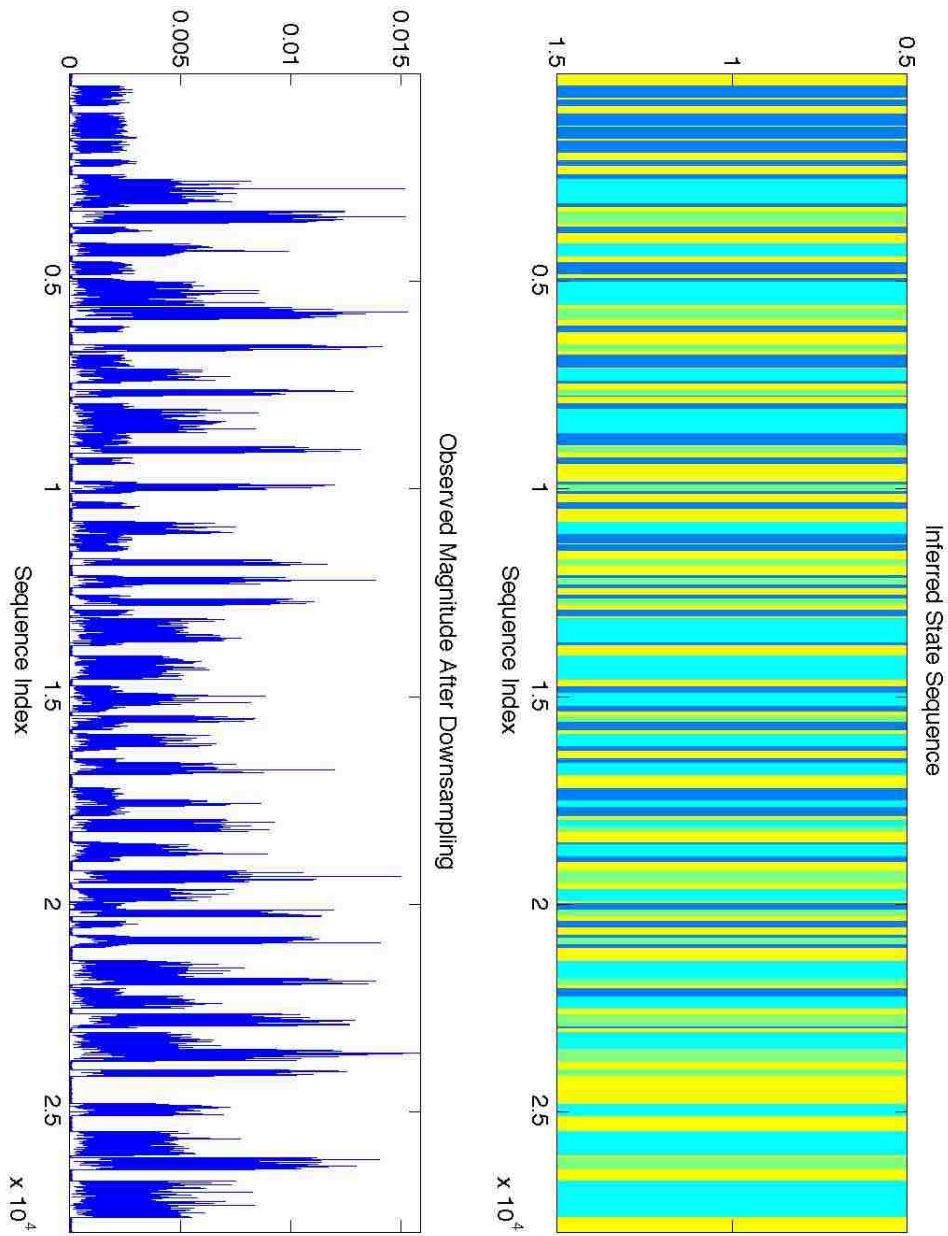
Figure 4.2: Labeled State Sequence for Experiment 2. Yellow = Idle State, Cyan = USRP 1/Collision, Dark Blue = USRP 2, Green = USRP 3/Collisions.

| Coordinated USRPs | | | Interfering USRP |
|---|---|---|---|
| | USRP 1 | USRP 2 | USRP 3 |
| Modulation | OFDM-BPSK | OFDM-BPSK | OFDM-BPSK |
| Symbol Bandwidth | 500 kHz | 250 kHz | 500 kHz |
| Number of Subcarriers | 200 | 200 | 200 |
| Packet Size | 508 bytes | 508 bytes | 508 bytes |
| Packet Bunches | [5, 10, 15] | [5, 10, 15] | [5, 10, 15] |
| Packet Emission Probs. | [1/2, 1/4, 1/4] | [1/2, 1/4, 1/4] | [1/2, 1/4, 1/4] |
| Transmission Gain | 10.0 | 20.0 | -15.0 |
| Idle State Duration | $D_{off} \sim \text{Exp}(0.45)$ sec $D_{off} \in [0.3, 0.7]$ sec | | $D_{off} \sim \text{Exp}(1.5)$ sec, $D_{off} \in [1.0, 3.0]$ sec |

- States 2-5: $\lambda_i \sim \text{Gamma}(k = 20, \theta = 5)$; $\overrightarrow{\mu_i}, \mathbf{\Sigma}_i \sim \text{NIW}\left(\overrightarrow{\mu_0} = \overrightarrow{0}, \mathbf{\Lambda} = \mathbf{I}, \kappa = 8, \nu = 3\right)$

- $\alpha = \gamma = \rho = 3$

- $N_1 = 50$ iterations

- $N_2 = 50$ iterations

The legend for the states appearing the final labeled state sequence depicted in Figure 4.3 is as follows.

- State 1 (Dark Blue): USRP 2, having mean duration parameter $\lambda_1 = 296.183757$

- State 2 (Cyan): Idle State, having mean duration parameter $\lambda_2 = 191.789526$

- State 3 (Yellow): USRP 1 and USRP 3/Collisions, having mean duration parameter $\lambda_3 = 115.911212$

In contrast to the previous experiment, it is much less apparent that there are any packet collisions from the state labeling due to the much reduced interference level introduced by USRP 3. As a consequence, the column of $\mathbf{A}$ corresponding to the idle state is more similar to that of the coordinated network presented in section 4.2.1.3. This fact strengthens the suggestion that the state transition matrix cannot be the sole discriminating evidence for judging the coordination between observed RF devices as associated with the inferred HSMM states.

$$\mathbf{A} = \begin{bmatrix} & 1 & 2 & 3 \\ 1 & 0 & 0.7734 & 0.2266 \\ 2 & 0.7774 & 0 & 0.2226 \\ 3 & 0.1352 & 0.8648 & 0 \end{bmatrix} \tag{4.85}$$

### 4.2.1.6 Two Coordinated OFDM Users with QPSK Interference

The USRP model specifications for the fourth experiment is detailed below in Table 4.4. In this case, the two coordinated OFDM users are given different modulation scheme for subcarriers within their respective OFDM symbols, while the interfering USRP is programmed to use QPSK instead of OFDM as its modulation. The following outline also lists the initial hyperparameters for the HSMM algorithm, with the most noticeable difference being an increase in $N_{max}$ to 8.

Table 4.4: USRP Generation of Experiment 4.

| | Coordinated USRPs | | Interfering USRP |
|---|---|---|---|
| | USRP 1 | USRP 2 | USRP 3 |
| Modulation | OFDM-BPSK | OFDM-QAM64 | QPSK |
| Symbol Bandwidth | 250 kHz | 250 kHz | — |
| Bit Rate | — | — | 250 kHz |
| Number of Subcarriers | 200 | 200 | — |
| Packet Size | 508 bytes | 508 bytes | 508 bytes |
| Packet Bunches | [5, 10, 15] | [5, 10, 15] | [5, 10, 15] |
| Packet Emission Probs. | [1/2, 1/2, 0] | [1/2, 1/2, 0] | [1/2, 1/2, 0] |
| Transmission Gain | 10.0 | 20.0 | 8.0 |
| Idle State Duration | $D_{off} \sim \mathrm{Exp}(0.45)$ sec $D_{off} \in [0.3, 0.7]$ sec | | $D_{off} \sim \mathrm{Exp}(1.2)$ sec, $D_{off} \in [0.5, 1.75]$ sec |

**Initial Hyperparameters**

- $N_{max} = 8$

- State 1: $\lambda \sim \mathrm{Gamma}(k = 10, \theta = 5)$; $\overrightarrow{\mu}, \boldsymbol{\Sigma} \sim \mathrm{NIW}\Big(\overrightarrow{\mu_0} = \overrightarrow{0}, \boldsymbol{\Lambda} = \mathbf{I}, \kappa = 5,$ $\nu = 3\Big)$
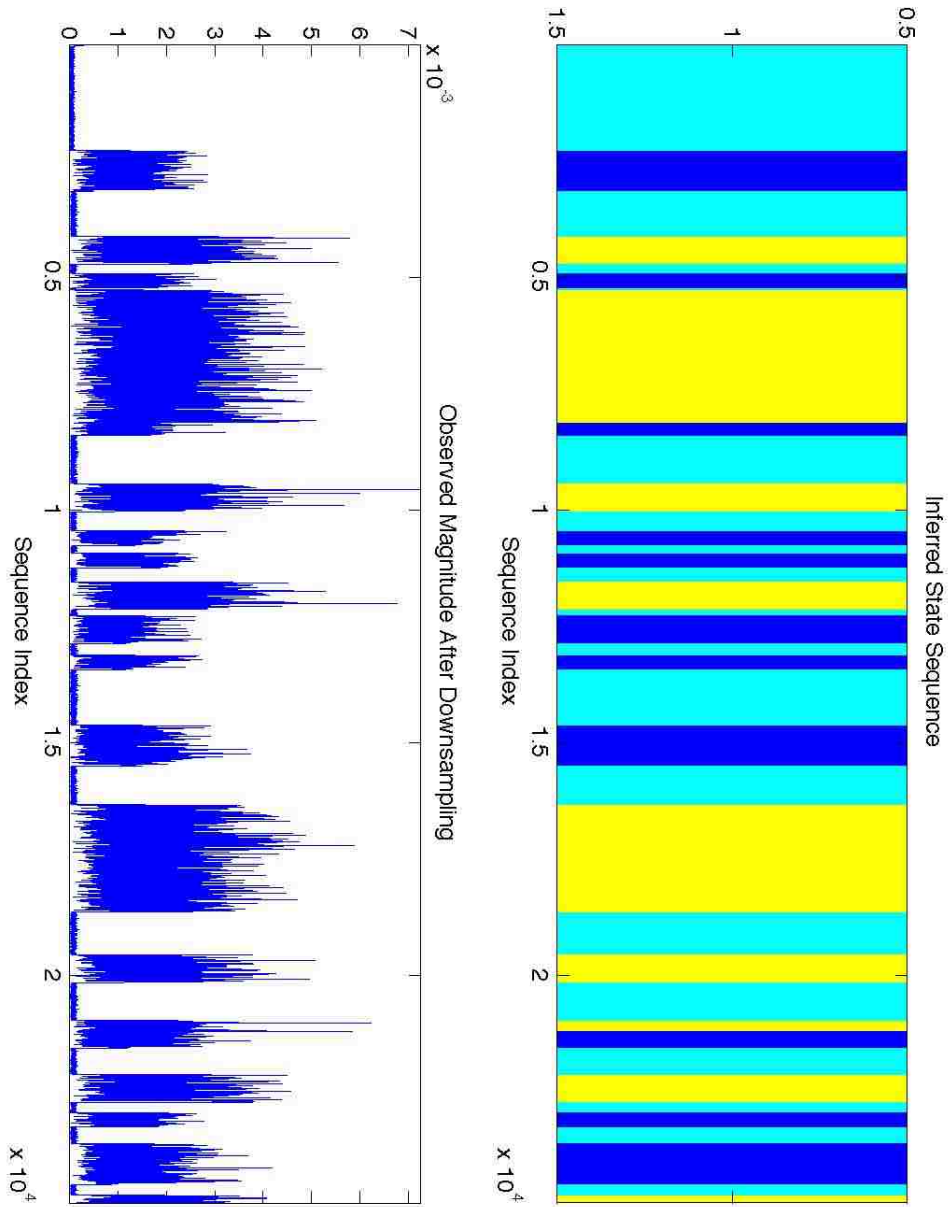
Figure 4.3: Labeled State Sequence for Experiment 3. Cyan = Idle State, Yellow = USRP 1/USRP 3 with Collisions, Dark Blue = USRP 2.

- States 2-8: $\lambda_i \sim \text{Gamma}(k = 20, \theta = 5)$; $\overrightarrow{\mu_i}, \mathbf{\Sigma}_i \sim \text{NIW}\left(\overrightarrow{\mu_0} = \overrightarrow{0}, \mathbf{\Lambda} = \mathbf{I}, \kappa = 8, \nu = 3\right)$

- $\alpha = \gamma = \rho = 3$

- $N_1 = 50$ iterations

- $N_2 = 50$ iterations

The following bullets provide the legend for interpreting the time-domain state switching depicted in Figure 4.4.

- State 1 (Black): Transients, having mean duration parameter $\lambda_1 = 23.346766$

- State 2 (Blue): USRP 3 and USRP 2/USRP 3 Collisions, having mean duration parameter $\lambda_2 = 96.327425$

- State 3 (Cyan): USRP 2, having mean duration parameter $\lambda_3 = 98.187928$

- State 5 (Yellow): Idle State, having mean duration parameter $\lambda_4 = 132.951206$

- State 6 (Orange): USRP 1 and USRP 1/USRP 3 Collisions, having mean duration parameter $\lambda_5 = 132.164126$

- State 7 (Red): USRP 2, having mean duration parameter $\lambda_6 = 95.707505$

As evidenced by the labeled state sequence, the algorithm resulted in a notably larger state space. As in section 4.2.1.4, the algorithm was unable to create distinct states corresponding to which coordinated user's packets collide with interfering USRP 3's transmissions. Furthermore, there seems to be a high degree of coupling between states 3 and 7 over the course of the transmission durations resulting from USRP 2's transmissions, as well as a relatively short lasting transient state 1 occurring infrequently within the state sequence. The strong coupling between states 3 and 7 suggest a systematic error in labeling the state sequence given the inferred model parameters up to the last iteration of the algorithm. This strong degree of frequent switching and larger state space leads to less evident relationship in examining the rows and columns of the state transition matrix $\mathbf{A}$ as given below.

$$\mathbf{A} = \begin{bmatrix} & 1 & 2 & 3 & 5 & 6 & 7 \\ 1 & 0.000000 & 0.020870 & 0.185350 & 0.593991 & 0.199627 & 0.000162 \\ 2 & 0.159109 & 0.000000 & 0.049473 & 0.452194 & 0.166289 & 0.172935 \\ 3 & 0.000064 & 0.023298 & 0.000000 & 0.165509 & 0.023859 & 0.787270 \\ 5 & 0.010114 & 0.505946 & 0.206098 & 0.000000 & 0.078658 & 0.199184 \\ 6 & 0.011801 & 0.011199 & 0.001997 & 0.973571 & 0.000000 & 0.001432 \\ 7 & 0.138084 & 0.001772 & 0.713293 & 0.124696 & 0.022155 & 0.000000 \end{bmatrix} \quad (4.86)$$

Nonetheless, the algorithm does perform well in labeling the idle state and uncollided transmissions for USRPs 1 and 3. Although one may be tempted to say that USRP 2's underlying modulation of QAM-64 may be responsible for introducing larger statistical variations resulting in the more complicated HSMM model, the remaining parameters between the two coordinated OFDM devices are identical, so it is unlikely that the modulation schemes would be solely responsible for introducing significantly different time domain characteristics. This experiment thus demonstrates that an effective implementation of the HSMM algorithm for identifying distinct RF devices should possess some means for evaluating the effects of possible systemic errors at the final iteration before making any conclusive inferences about the observed wireless environment.

### 4.2.1.7   Two Coordinated OFDM Users with GMSK Interference

Table 4.5 presents a summary for the fifth experimental setup, in which the interference introduced by USRP 3 is the result of GMSK transmissions over the wireless channel. Since GMSK is a means of frequency shift keying and is thus distinct from the OFDM transmission scheme and the underlying phase shift keying for USRPs 1 and 2, it is expected that the HSMM algorithm should be able to distinguish GMSK packet transmissions with better accuracy. The following outline also shows that the initial hyperparameters for the HSMM algorithm is unchanged with respect to the initialization within the fifth experiment detailed in section 4.2.1.6.

Table 4.5: USRP Generation of Experiment 5.

| | Coordinated USRPs | | Interfering USRP |
|---|---|---|---|
| | USRP 1 | USRP 2 | USRP 3 |
| Modulation | OFDM-BPSK | OFDM-BPSK | GMSK |
| Symbol Bandwidth | 250 kHz | 250 kHz | — |
| Bit Rate | — | — | 250 kHz |
| Number of Subcarriers | 200 | 200 | — |
| Packet Size | 508 bytes | 508 bytes | 508 bytes |
| Packet Bunches | [5, 10, 15] | [5, 10, 15] | [5, 10, 15] |
| Packet Emission Probs. | [1/2, 1/2, 0] | [1/2, 1/2, 0] | [1/2, 1/2, 0] |
| Transmission Gain | 10.0 | 20.0 | 8.0 |
| Idle State Duration | $D_{off} \sim \text{Exp}(0.2)$ sec $D_{off} \in [0.3, 0.7]$ sec | | $D_{off} \sim \text{Exp}(0.2)$ sec, $D_{off} \in [0.3, 0.7]$ sec |

Figure 4.4: Labeled State Sequence for Experiment 4. Yellow = Idle State, Orange = USRP 1/Collisions, Cyan and Red = USRP 2, Dark Blue = USRP 3/Collisions, Black = Transients.

**Initial Hyperparameters**

- $N_{max} = 8$

- State 1: $\lambda \sim \text{Gamma}(k = 10, \theta = 5)$; $\overrightarrow{\mu}, \mathbf{\Sigma} \sim \text{NIW}\left(\overrightarrow{\mu_0} = \overrightarrow{0}, \mathbf{\Lambda} = \mathbf{I}, \kappa = 5, \nu = 3\right)$

- States 2-8: $\lambda_i \sim \text{Gamma}(k = 20, \theta = 5)$; $\overrightarrow{\mu_i}, \mathbf{\Sigma}_i \sim \text{NIW}\left(\overrightarrow{\mu_0} = \overrightarrow{0}, \mathbf{\Lambda} = \mathbf{I}, \kappa = 8, \nu = 3\right)$

- $\alpha = \gamma = \rho = 3$

- $N_1 = 50$ iterations

- $N_2 = 50$ iterations

The labeled state sequence upon the final iteration of the algorithm is shown in Figure 4.5, with the following outline providing a legend for the 3 identified states within the model.

- State 1 (Blue): USRP 3 and Collisions, having mean duration parameter $\lambda_1 = 131.650711$

- State 2 (Cyan): Idle State, having mean duration parameter $\lambda_2 = 378.450516$

- State 3 (Yellow): USRP 1 and USRP 2, having mean duration parameter $\lambda_3 = 254.233984$

As is apparent from Figure 4.5, the algorithm does indeed perform well in identifying both the idle state and GMSK transmissions over the course of the RF trace. Due to the high power for the introduced interference from GMSK, all collisions occurring between the coordinated users and USRP 3 are considered as belonging to the same state upon the final iteration. Likewise, state 3 also includes transmissions from both coordinated OFDM users as opposed to treating them distinctly. Since several transmissions from USRP 1 are included within state 3 representing vacancies over the wireless channel, it is worthwhile to study if lowering the transmission power of the interfering GMSK user as in the next section can resolve this error in labeling the state sequence.

$$\mathbf{A} = \begin{bmatrix} & 1 & 2 & 3 \\ 1 & 0.000000 & 0.552871 & 0.447129 \\ 2 & 0.883738 & 0.000000 & 0.116262 \\ 3 & 0.734436 & 0.265564 & 0.000000 \end{bmatrix} \tag{4.87}$$
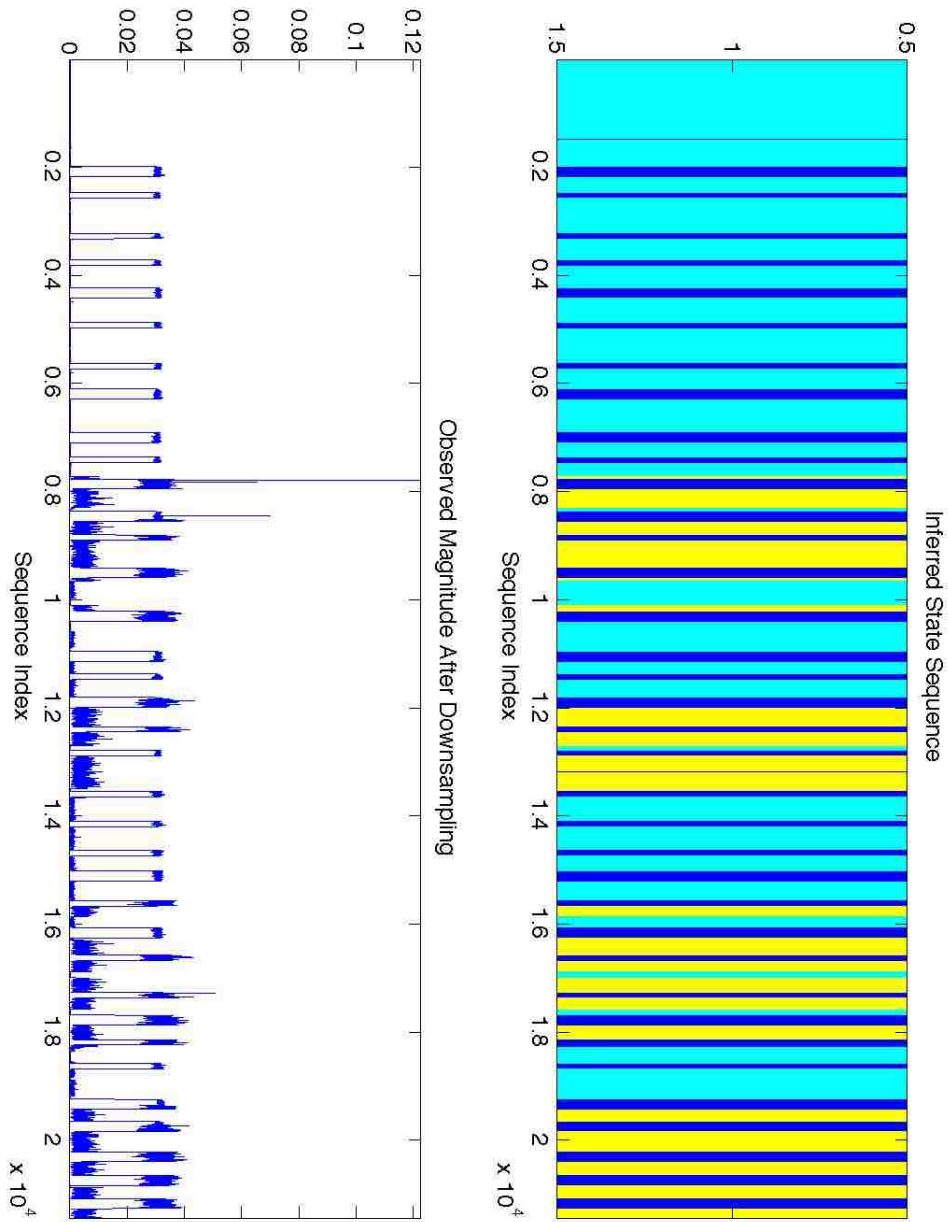
Figure 4.5: Labeled State Sequence for Experiment 5. Cyan = Idle State, Yellow = USRP 1 and USRP 2, Green = USRP 3/Collision.

### 4.2.1.8 Two Coordinated OFDM Users with Low Power GMSK Interference

In order to address some of the difficulties presented in the previous section, the transmission power for the interfering GMSK device was reduced as noted in Table 4.6. The transmitter powers of the coordinated USRPs 1 and 2 were also increased to assess how the relative powers between the devices would affect the final state labeling. The initial hyperparameter setup was left unaltered as well, except for a reduction of $N_{max}$ to 6.

Table 4.6: USRP Generation of Experiment 6.

| | Coordinated USRPs | | Interfering USRP |
|---|---|---|---|
| | USRP 1 | USRP 2 | USRP 3 |
| Modulation | OFDM-BPSK | OFDM-BPSK | GMSK |
| Symbol Bandwidth | 250 kHz | 250 kHz | — |
| Bit Rate | — | — | 250 kHz |
| Number of Subcarriers | 200 | 200 | — |
| Packet Size | 508 bytes | 508 bytes | 508 bytes |
| Packet Bunches | [5, 10, 15] | [5, 10, 15] | [5, 10, 15] |
| Packet Emission Probs. | [1/2, 1/2, 0] | [1/2, 1/2, 0] | [1/2, 1/2, 0] |
| Transmission Gain | 15.0 | 25.0 | 3.0 |
| Idle State Duration | $D_{off} \sim \text{Exp}(0.2)$ sec $D_{off} \in [0.3, 0.7]$ sec | | $D_{off} \sim \text{Exp}(0.2)$ sec, $D_{off} \in [0.3, 0.7]$ sec |

**Initial Hyperparameters**

- $N_{max} = 6$

- State 1: $\lambda \sim \text{Gamma}(k = 10, \theta = 5)$; $\overrightarrow{\mu}, \Sigma \sim \text{NIW}\left(\overrightarrow{\mu_0} = \overrightarrow{0}, \Lambda = \mathbf{I}, \kappa = 5, \nu = 3\right)$

- States 2-6: $\lambda_i \sim \text{Gamma}(k = 20, \theta = 5)$; $\overrightarrow{\mu_i}, \Sigma_i \sim \text{NIW}(\overrightarrow{\mu_0} = \overrightarrow{0}, \Lambda = \mathbf{I}, \kappa = 8, \nu = 3\right)$

- $\alpha = \gamma = \rho = 3$

- $N_1 = 50$ iterations

- $N_2 = 50$ iterations

The legend for interpreting the state labeling as seen in 4.6 is presented below.

- State 1 (Black): USRP 3, having mean duration parameter $\lambda_1 = 65.255393$

- State 2 (Dark Blue): USRP 2, having mean duration parameter $\lambda_2 = 130.422329$

- State 3 (Cyan): Idle State and USRP 1, having mean duration parameter $\lambda_3 = 149.927048$

- State 4 (Green): Collisions and Transients, having mean duration parameter $\lambda_4 = 41.915780$

- State 5 (Yellow): USRP 3 and Collisions, having mean duration parameter $\lambda_5 = 107.176962$

Once again, the final results in the state labeling show missed detection of USRP 1, as it is considered to belong to the same state as vacancies over the channel. Instances of GMSK's presence over the channel are consistently detected and considered as belonging to the same state, regardless of collision instances or undisturbed transmission. States 1 and 4 of the model also represent shorter duration states that seem to result from transient behavior as transmissions initiate from the USRP devices.

$$\mathbf{A} = \begin{bmatrix} & 1 & 2 & 3 & 4 & 5 \\ 1 & 0.000000 & 0.007514 & 0.147488 & 0.778349 & 0.066649 \\ 2 & 0.000001 & 0.000000 & 0.262107 & 0.698158 & 0.039734 \\ 3 & 0.001195 & 0.000002 & 0.000000 & 0.201300 & 0.797503 \\ 4 & 0.004149 & 0.577418 & 0.142121 & 0.000000 & 0.276312 \\ 5 & 0.506790 & 0.064346 & 0.268037 & 0.160827 & 0.000000 \end{bmatrix} \tag{4.88}$$

#### 4.2.1.9 Two Coordinated OFDM Users with GMSK Interference (Second Case)

As a counterpart to the experiment of 4.2.1.7, USRP 3 is presented as interference to a coordinated network of USRPs using OFDM with two different underlying modulation schemes, namely BPSK and QAM-64 as detailed in Table 4.7. It is worth noting that the transmission gains of the three USRPs are identical to those presented in 4.2.1.7, with the another significant differences in experimental setup being the ranges in which the idle state durations may be drawn for each type of network. A list of the initial hyperparameters at the commencement of the algorithm is detailed in the following bulleted list.

**Initial Hyperparameters**

- $N_{max} = 8$

- State 1: $\lambda \sim \text{Gamma}(k = 10, \theta = 5)$; $\overrightarrow{\mu}, \boldsymbol{\Sigma} \sim \text{NIW}\left(\overrightarrow{\mu_0} = \overrightarrow{0}, \boldsymbol{\Lambda} = \mathbf{I}, \kappa = 5, \nu = 3\right)$
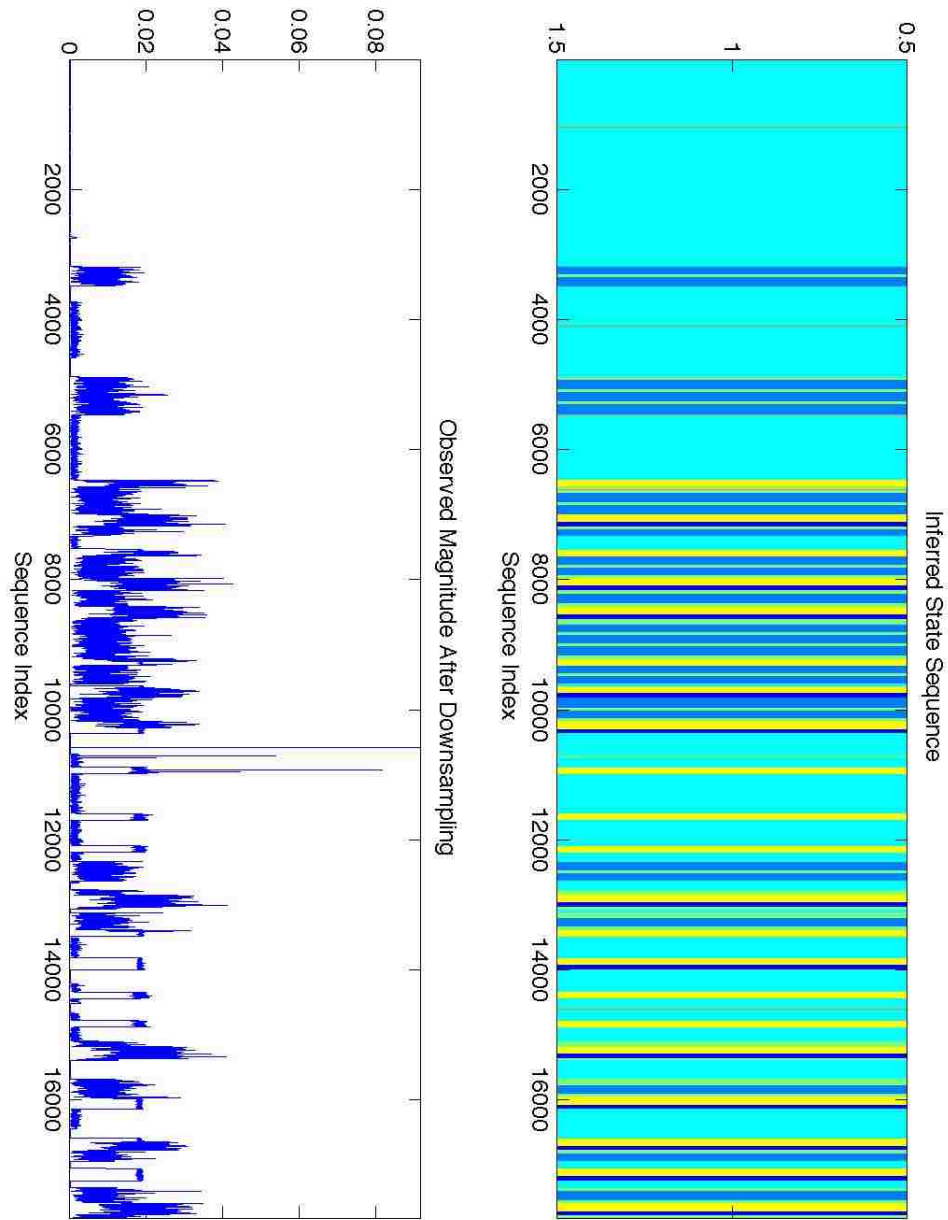
Figure 4.6: Labeled State Sequence for Experiment 6. Cyan = Idle State/USRP 1, Dark Blue = USRP 2, Black and Yellow = USRP 3, Green = Collision/Transients.

Table 4.7: USRP Generation of Experiment 7.

| | Coordinated USRPs | | Interfering USRP |
|---|---|---|---|
| | USRP 1 | USRP 2 | USRP 3 |
| Modulation | OFDM-BPSK | OFDM-QAM64 | GMSK |
| Symbol Bandwidth | 250 kHz | 250 kHz | — |
| Bit Rate | — | — | 250 kHz |
| Number of Subcarriers | 200 | 200 | — |
| Packet Size | 508 bytes | 508 bytes | 508 bytes |
| Packet Bunches | [5, 10, 15] | [5, 10, 15] | [5, 10, 15] |
| Packet Emission Probs. | [1/2, 1/2, 0] | [1/2, 1/2, 0] | [1/2, 1/2, 0] |
| Transmission Gain | 10.0 | 20.0 | 8.0 |
| Idle State Duration | $D_{off} \sim \text{Exp}(0.45)$ sec $D_{off} \in [0.3, 0.7]$ sec | | $D_{off} \sim \text{Exp}(1.2)$ sec, $D_{off} \in [0.5, 1.75]$ sec |

- States 2-8: $\lambda_i \sim \text{Gamma}(k = 20, \theta = 5)$; $\vec{\mu_i}, \Sigma_i \sim \text{NIW}\left(\vec{\mu_0} = \vec{0}, \Lambda = \mathbf{I}, \kappa = 8, \nu = 3\right)$

- $\alpha = \gamma = \rho = 3$

- $N_1 = 50$ iterations

- $N_2 = 50$ iterations

The legend for interpreting the state labeling as seen in 4.7 is presented below. From examining the final HSMM state sequence as depicted in 4.7, the resultant 4 states appearing in the model's labeling seem to clearly identify undisturbed packet transmissions from both USRPs 1 and 2 as well as identifying the idle state accurately. State 2 however identifies both uninterfered transmissions from the GMSK-employed USRP 3 as being identical with any interference that USRP 3 introduces to the coordinated network. This combination can mostly be attributed to the high power with which USRP 3 appears in the RF trace in comparison to the users of the coordinated network. Furthermore, the heavy weighting of first column of the transition matrix $\mathbf{A}$ also tends to reflect the frequent appearance of the GMSK user's transmissions over the course of the RF measurements.

- State 2 (Cyan): USRP 3 and Collisions, having mean duration parameter $\lambda_2 = 138.4176$

- State 3 (Green): Idle State, having mean duration parameter $\lambda_3 = 162.2581$

- State 4 (Yellow): USRP 2, having mean duration parameter $\lambda_4 = 129.6950$

- State 5 (Orange): USRP 1, having mean duration parameter $\lambda_5 = 127.7830$

$$\mathbf{A} = \begin{bmatrix} & 2 & 3 & 4 & 5 \\ 2 & 0.000000 & 0.578705 & 0.002443 & 0.418852 \\ 3 & 0.667084 & 0.000000 & 0.000011 & 0.332905 \\ 4 & 0.531989 & 0.255759 & 0.000000 & 0.212252 \\ 5 & 0.627741 & 0.361974 & 0.010285 & 0.000000 \end{bmatrix} \qquad (4.89)$$
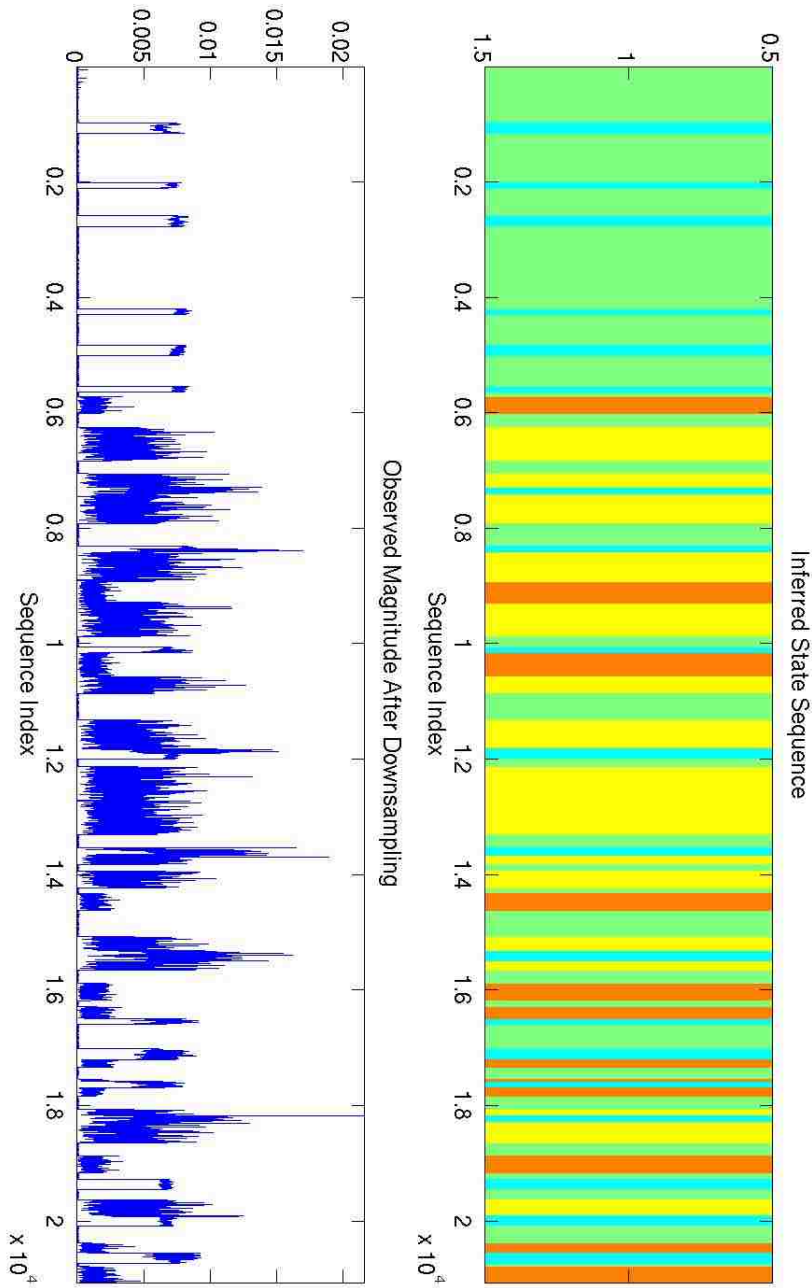
Figure 4.7: Labeled State Sequence for Experiment 7. Green = Idle State, Orange = USRP 1, Yellow = USRP 2, Cyan = USRP 3 and Collisions.

# Chapter 5

# Wireless LAN Analysis Using the HDP-HSMM Algorithm

The focus of this chapter is to assess the HDP-HSMM algorithm's efficacy in identifying distinct transmitters and their associated activity patterns operating under the ubiquitous IEEE 802.11 wireless network standard. The experiments were carried out by establishing a direct connection between a laptop and a nearby access point (AP) functioning to wirelessly broadcast files from a desktop without native wireless capabilities. After establishing the direct link connection between the two computers, bidirectional file transfers were executed with a USRP capturing the wireless trace of the data transmissions over time. These measurements were then fed into the previously explained HDP-HSMM algorithm for further analysis.

## 5.1    Types of IEEE 802.11 Transmissions

Before discussing more details of the experimental setup, it is important to differentiate between the two functionally different types of packets that are sent over the 2.4 GHz wireless band according to the IEEE 802.11 standard. In addition to data packets sent over the channel for traditional node-to-node file transfers, 802.11 devices also transmit and receive a variety of management packets responsible for establishing the networks in which to associate and also for regulating use of the wireless band under a medium access protocol (MAC). As an example, wireless APs periodically transmit short-duration beacon signals to announce its presence to all proximal devices that may seek to join the AP's network. The MAC protocol of the 802.11 standard utilizes a form of carrier sense multiple access with collision avoidance (CSMA/CA) technique to mitigate potential collisions over the wireless channel and to coordinate multiple network peers' transmissions amongst each other. This MAC protocol involves a regular series of management packet transmissions between the network peers to indicate requests to send data as well as clearances to send data.

In the scope of this thesis, it is desired to capture both the data and management packets over the 802.11 network's course of operation, since identification of bidirectional management packet transmissions could provide strong evidence about the potential networks observable within the collected RF trace. As will be shown in the later results, successful capture of the management packets is hindered by the comparatively smaller size of the management packets after the RF trace is downsampled as discussed in Section 4.2.1.2.

## 5.2    802.11 Wireless Trace Collection Procedure

As mentioned at the beginning of this chapter, a wireless broadasting network was established by connecting a desktop to a Cisco Aironet 1200 access point. This connection allowed file transfers from the desktop computer to other laptops within proximity of the AP as well as transfers in the reverse direction. The AP was sent to broadcast on the third channel of the 802.11 subdivision of the 2.4 GHz wireless spectrum; this particular channel was chosen so as to minimize the potential for transmissions associated with other APs within the laboratory building to be observed in the RF traces collected by the USRP.[1] A laptop computer was subsequently allowed to join the AP's broadcast network to commence file sharing between the two computers.

Both the laptop and desktop computers were each given several distinct files of varying size ranging from approximately 100 kilobytes to 1.5 megabytes as detailed in Table 5.1. In each experiment, random permutations in which to transmit the distinct files were generated using MATLAB, and the file transfers were executed between the two computers with the use of a file transfer protocol client.[2] Time intervals between file transfers was randomized in the sense that commencement of the transfer was initiated manually and not through any periodic or computerized methods.

Table 5.1: List of File Sizes Per Device

|   | Laptop | Desktop |
|---|--------|---------|
| 1 | 115 kB | 152 kB |
| 2 | 171 kB | 395 kB |
| 3 | 534 kB | 541 kB |
| 4 | 558 kB | 790 kB |
| 5 | 962 kB | 966 kB |
| 6 | 1,030 kB | 1,193 kB |
| 7 | 1,214 kB | 1,637 kB |

The transmission powers of both the AP and the laptop client were controllable through direct specification of the AP's customizable settings, as mentioned in the next three items.

- In order for the AP to remain 802.11b compliant, data rates below 20 megabits per second (MBps) use the complementary code keying (CCK) modulation scheme detailed

---

[1]For reference, the third channel of the 802.11 standard indicates a center frequency of 2.422 GHz for transmissions. Choosing such a channel helps to reduce potential for interference introduced by the most commonly used channels of 1, 5, 9, and 13 in typical 802.11 deployments. The IEEE 802.11 standard may be consulted for more details about the channel allocation scheme for the 2.4 GHz band. [11]

[2]Note that the files on the laptop set for transmission were sent to the laptop with no possibility for resending, and vice versa. In this way, a specific file only gets sent once over the sequence of file transactions over the wireless channel for a given permutation.

in [11]. The AP's CCK transmitter power was configurable to distinct values within a range of 1 to 100 milliwatts (mW).

- For data rates above 20 Mbps and other intermediate values not mandated by the 802.11b specification, OFDM is used as the modulation scheme for transmissions over the channel. The AP allows the transmitter power to be set at distinct values between 1 and 30 mW.

- The AP also possesses a Limit Client Power option that forces associated clients to the AP to transmit below a maximal power limit. This option was enabled over the course of the experiments, with the client's power limit being varied in a range between 1 to 100 mW. Note that this does not strictly mean that the clients will continuously transmit packets at the level set by the limit set by the AP.

## 5.3   802.11 Wireless Trace Analysis

### 5.3.1   WLAN Experiment 1

The first WLAN experiment is used to assess the algorithm's capability of discerning between the packet transmissions of the laptop and the AP whenever there is a large contrast in transmission power between the two different users. The power specifications for this experiment are detailed in Table 5.2. The initial hyperparameters may also be found in the following outline, with the maximum number of states set to $N_{max} = 3$ over the course of 200 iterations. The sequence in which files are transmitted over this captured RF trace can also be found in Table 5.3 [3].

Table 5.2: Power Specifications for WLAN Experiment 1

| CCK Power | 30 mW |
|---|---|
| OFDM Power | 30 mW |
| Max. Client Power | 1 mW |

**Initial Hyperparameters**

- $N_{max} = 3$

- State 1: $\lambda \sim \text{Gamma}(k = 150, \theta = 2)$; $\overrightarrow{\mu}, \mathbf{\Sigma} \sim \text{NIW}\left(\overrightarrow{\mu_0} = \overrightarrow{0}, \mathbf{\Lambda} = \mathbf{I}, \kappa = 300,\right.$
  $\left. \nu = 102\right)$

---

[3]An 'A' signifies a file originating from the access point, while an 'L' denotes files originating from the laptop computer. The numbers correspond to the specific file sent in accordance with Table 5.1.

- State 2: $\lambda \sim \mathrm{Gamma}(k = 25, \theta = 2)$; $\overrightarrow{\mu}, \mathbf{\Sigma} \sim \mathrm{NIW}\left(\overrightarrow{\mu_0} = \overrightarrow{0}, \mathbf{\Lambda} = \mathbf{I}, \kappa = 3000, \nu = 302\right)$

- State 3: $\lambda \sim \mathrm{Gamma}(k = 30, \theta = 7)$; $\overrightarrow{\mu}, \mathbf{\Sigma} \sim \mathrm{NIW}\left(\overrightarrow{\mu_0} = \overrightarrow{0}, \mathbf{\Lambda} = \mathbf{I}, \kappa = 3000, \nu = 302\right)$

- $N_{iter} = 200$ iterations

Table 5.3: File Transfer Sequence for WLAN Experiment 1

| L3, L6, L5, L1, L4, A7, A6, A1, A3 |

The learned HSMM state sequence is graphically presented in Figure 5.1. As similarly found in the results of the previous chapter, the algorithm is generally successful in differentiating idle periods over the RF band from file transmissions from the two users. On the other hand, the algorithm does not successfully capture every periodic beacon signal sent from the AP. Likewise, there also appears to be several instances of 'stray' packet transmissions not sent from the laptop or AP, as would be found over portions of the RF band possessing coexisting networks.

**Experiment 1 State Legend**

- State 1 (Dark Blue): 802.11 Data Traffic Transmissions, having mean duration parameter $\lambda_1 = 107.4008$, multivariate Gaussian mean vector $\overrightarrow{\mu_1} = [0.0013, -0.0048]$, and multivariate Gaussian covariance matrix $\mathbf{\Sigma}_1$:

$$\mathbf{\Sigma}_1 = \begin{bmatrix} 0.0285 & -8.2705 \times 10^{-4} \\ -8.2705 \times 10^{-4} & 0.0413 \end{bmatrix}$$

- State 2 (Cyan): 802.11 Beacon Signals, having mean duration parameter $\lambda_2 = 1.4065$ multivariate Gaussian mean vector $\overrightarrow{\mu_2} = [-8.6052 \times 10^{-4}, 0.0042]$, and multivariate Gaussian covariance matrix $\mathbf{\Sigma}_2$:

$$\mathbf{\Sigma}_2 = \begin{bmatrix} 0.0477 & 0.0020 \\ 0.0020 & 0.0841 \end{bmatrix}$$

- State 3 (Yellow): Idle State, having mean duration parameter $\lambda_3 = 146.1474$, multivariate Gaussian mean vector $\overrightarrow{\mu_3} = [-6.4365 \times 10^{-5}, 2.5295 \times 10^{-5}]$, and multivariate Gaussian covariance matrix $\mathbf{\Sigma}_3$:

$$\mathbf{\Sigma}_3 = \begin{bmatrix} 3.4359 \times 10^{-5} & -2.4728 \times 10^{-7} \\ -2.4728 \times 10^{-7} & 3.4509 \times 10^{-5} \end{bmatrix}$$

$$\mathbf{A} = \begin{bmatrix} & 1 & 2 & 3 \\ 1 & 0.0000 & 0.9995 & 0.0005 \\ 2 & 0.1112 & 0.0000 & 0.8888 \\ 3 & 0.0000 & 1.0000 & 0.0000 \end{bmatrix} \qquad (5.1)$$

## 5.3.2  WLAN Experiment 2

As discussed in the previous section, the results of the previous experiment did not identify the existence of two distinct users or systematically capture the periodic beacon signal emitted by the AP. The HDP-HSMM algorithm was consequently rerun for the same RF data trace, under identical initial conditions with the exception of an increase in the number of iterations to $N_{iter} = 250$. The purpose of this repeat calculation was to determine a general guidelines for selecting the proper number of iterations and the consequent effect on the final learned HSMM state sequence. As noted, the experimental details as found in Tables 5.4 and 5.5, with the initial hyperparameters in the following outline remain otherwise unaltered.

Table 5.4: Power Specifications for WLAN Experiment 2

| | |
|---|---|
| CCK Power | 30 mW |
| OFDM Power | 30 mW |
| Max. Client Power | 1 mW |

**Initial Hyperparameters**

- $N_{max} = 3$

- State 1: $\lambda \sim \text{Gamma}(k = 150, \theta = 2)$; $\overrightarrow{\mu}, \mathbf{\Sigma} \sim \text{NIW}\left(\overrightarrow{\mu_0} = \overrightarrow{0}, \mathbf{\Lambda} = \mathbf{I}, \kappa = 300, \nu = 102\right)$

- State 2: $\lambda \sim \text{Gamma}(k = 25, \theta = 2)$; $\overrightarrow{\mu}, \mathbf{\Sigma} \sim \text{NIW}\left(\overrightarrow{\mu_0} = \overrightarrow{0}, \mathbf{\Lambda} = \mathbf{I}, \kappa = 3000, \nu = 302\right)$

Figure 5.1: Labeled State Sequence for WLAN Experiment 1. Yellow = Idle State, Dark Blue = 802.11 Data Traffic Packets, Cyan = Beacons.

- State 3: $\lambda \sim \text{Gamma}(k = 30, \theta = 7)$; $\overrightarrow{\mu}, \boldsymbol{\Sigma} \sim \text{NIW}\left(\overrightarrow{\mu_0} = \overrightarrow{0}, \boldsymbol{\Lambda} = \mathbf{I}, \kappa = 3000,\right.$

  $\left.\nu = 302\right)$

- $\alpha = \gamma = \rho = 3$

- $N_{iter} = 250$ iterations

Table 5.5: File Transfer Sequence for WLAN Experiment 2

| L3, L6, L5, L1, L4, A7, A6, A1, A3 |
| --- |

In a similar manner to the first case, the results as visualized in Figure 5.2 indicate that all packet transmissions, originating from either the laptop or the AP, are labeled as belonging to State 1, while only some of the AP's beacon signals are labeled as State 2. In comparing the learned results of the algorithm with those of the previous trial encompassing fewer iterations (summarized below), the learned parameter values for the state's distributions as well as the state transition matrices happen to be quite similar. Thus the increase in number of iterations for the second execution of this RF trace has resulted in quite comparable results as the first execution, thereby indicating that the algorithm's final results have stabilized by around 200 iterations, after which increasing the number of interations would not considerably alter the final learned results.

**Experiment 2 State Legend**

- State 1 (Dark Blue): 802.11 Data Traffic Transmissions, having mean duration parameter $\lambda_1 = 108.7185$, multivariate Gaussian mean vector $\overrightarrow{\mu_1} = [-0.0051, 0.0029]$, and multivariate Gaussian covariance matrix $\boldsymbol{\Sigma}_1$:

$$\boldsymbol{\Sigma}_1 = \begin{bmatrix} 0.0281 & -9.5957 \times 10^{-4} \\ -9.5957 \times 10^{-4} & 0.0414 \end{bmatrix}$$

- State 2 (Cyan): 802.11 Beacon Signals, having mean duration parameter $\lambda_2 = 1.2097$, multivariate Gaussian mean vector $\overrightarrow{\mu_2} = [0.0045, -0.0013]$, and multivariate Gaussian covariance matrix $\boldsymbol{\Sigma}_2$:

$$\boldsymbol{\Sigma}_2 = \begin{bmatrix} 0.0553 & 0.0030 \\ 0.0030 & 0.0884 \end{bmatrix}$$

- State 3 (Yellow): Idle State, having mean duration parameter $\lambda_3 = 146.1633$, multivariate Gaussian mean vector $\overrightarrow{\mu_3} = [-2.1540 \times 10^{-6}, 7.6220 \times 10^{-5}]$, and multivariate Gaussian covariance matrix $\boldsymbol{\Sigma}_3$:

$$\mathbf{\Sigma}_3 = \begin{bmatrix} 3.4284 \times 10^{-5} & 1.2194 \times 10^{-7} \\ 1.2194 \times 10^{-7} & 3.4908 \times 10^{-5} \end{bmatrix}$$

$$\mathbf{A} = \begin{bmatrix} & 1 & 2 & 3 \\ 1 & 0.0000 & 0.8778 & 0.1262 \\ 2 & 0.1314 & 0.0000 & 0.8686 \\ 3 & 0.0280 & 0.9720 & 0.0000 \end{bmatrix} \tag{5.2}$$

### 5.3.3  WLAN Experiment 3

This third presented case investigates the effect of increasing the maximum number of states $N_{max}$ to 4 using the same RF collected trace as the previous two WLAN experiments. The initialized hyperparameter values for this experiment are quite similar to the previous two experiments, with only the addition of an extra possible state with distinct duration hyperparameters from the remaining three model states. A brief summary of the remaining initial conditions for the experiment can be found below.

Table 5.6: Power Specifications for WLAN Experiment 3

| CCK Power | 30 mW |
|---|---|
| OFDM Power | 30 mW |
| Max. Client Power | 1 mW |

**Initial Hyperparameters**

- $N_{max} = 4$

- State 1: $\lambda \sim \text{Gamma}(k = 150, \theta = 2)$; $\overrightarrow{\mu}, \mathbf{\Sigma} \sim \text{NIW}\left(\overrightarrow{\mu_0} = \overrightarrow{0}, \mathbf{\Lambda} = \mathbf{I}, \kappa = 300, \nu = 102\right)$

- State 2: $\lambda \sim \text{Gamma}(k = 25, \theta = 2)$; $\overrightarrow{\mu}, \mathbf{\Sigma} \sim \text{NIW}\left(\overrightarrow{\mu_0} = \overrightarrow{0}, \mathbf{\Lambda} = \mathbf{I}, \kappa = 3000, \nu = 302\right)$
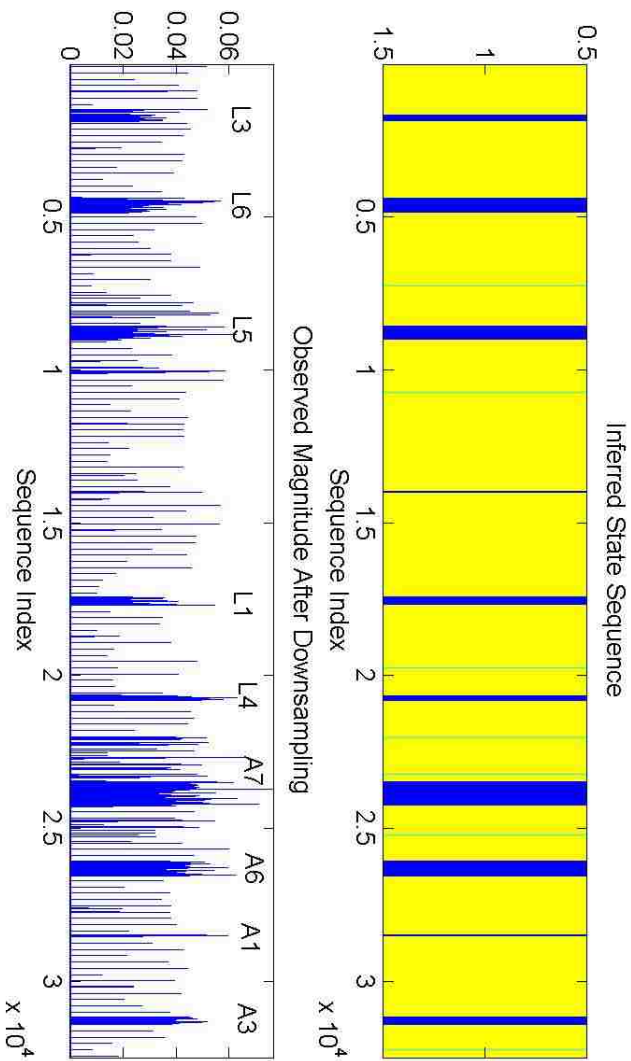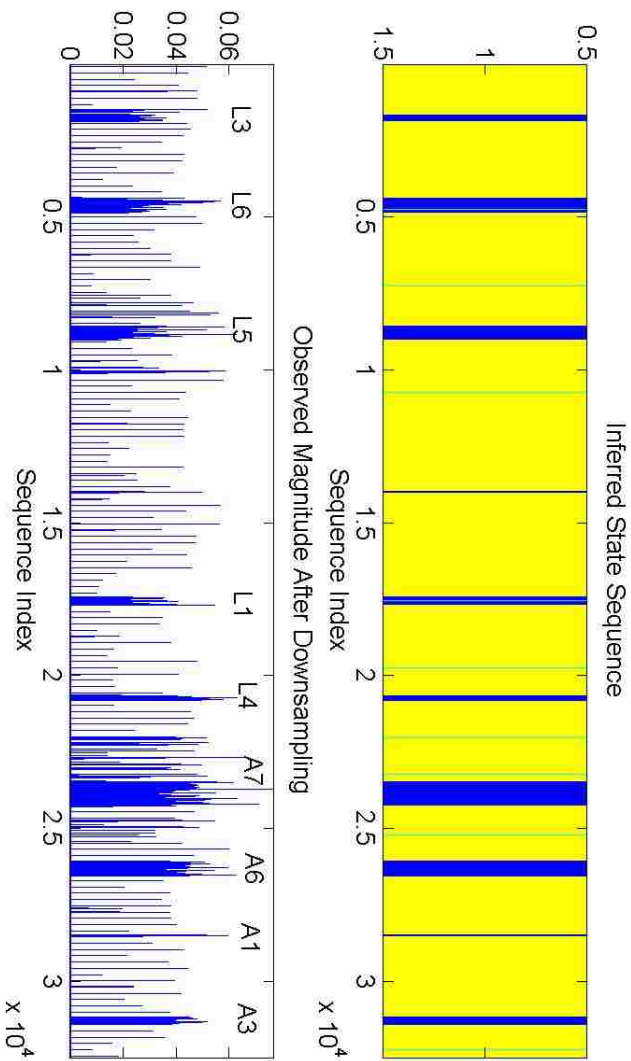
Figure 5.2: Labeled State Sequence for WLAN Experiment 2. Yellow = Idle State, Dark Blue = 802.11 Data Traffic Packets, Cyan = Beacons.

- State 3: $\lambda \sim \text{Gamma}(k = 30, \theta = 7)$; $\overrightarrow{\mu}, \Sigma \sim \text{NIW}\left(\overrightarrow{\mu_0} = \overrightarrow{0}, \Lambda = I, \kappa = 3000, \nu = 302\right)$

- State 4: $\lambda \sim \text{Gamma}(k = 100, \theta = 2)$; $\overrightarrow{\mu}, \Sigma \sim \text{NIW}\left(\overrightarrow{\mu_0} = \overrightarrow{0}, \Lambda = I, \kappa = 300, \nu = 102\right)$

- $\alpha = \gamma = \rho = 3$

- $N_{iter} = 250$ iterations

Table 5.7: File Transfer Sequence for WLAN Experiment 3

| L3, L6, L5, L1, L4, A7, A6, A1, A3 |
| --- |

Despite the increase in the maximum number of allowable states within the model, the learned results indicate that the data trace only possesses three states upon conclusion of the final iteration. Nonetheless, the previously noted difficulties of indistinguishability of different RF devices and sparse beacon signal labeling can be seen in the state sequence depicted in Figure 5.3.

**Experiment 3 State Legend**

- State 1 (Dark Blue): 802.11 Data Traffic Transmissions, having mean duration parameter $\lambda_1 = 203.9010$

- State 2 (Cyan): Idle State, having mean duration parameter $\lambda_2 = 87.4118$

- State 4 (Orange): 802.11 Beacon Signals, having mean duration parameter $\lambda_3 = 0.9792$

$$
\mathbf{A} = \begin{bmatrix}
 & 1 & 2 & 4 \\
1 & 0.0000 & 0.1620 & 0.8380 \\
2 & 0.0176 & 0.0000 & 0.9830 \\
4 & 0.0213 & 0.9871 & 0.0000
\end{bmatrix}
\tag{5.3}
$$

### 5.3.4 WLAN Experiment 4

For this experiment, the transmission power of the laptop was increased to 5 mW as indicated in Table 5.8. The maximum number of states was also increased to $N_{max} = 5$,
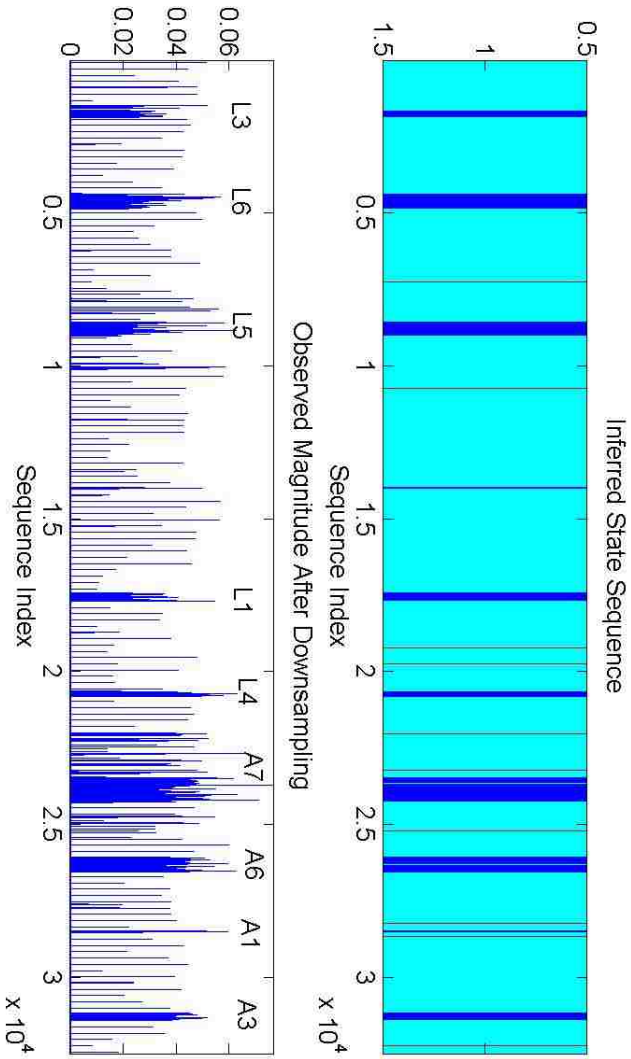
Figure 5.3: Labeled State Sequence for WLAN Experiment 3. Cyan = Idle State, Dark Blue = 802.11 Data Traffic Packets, Orange = Beacons.

with each state having initial hyperparameters as listed in the ensuing outline. Each state was also given a distinct set of initial hyperparameters in order to potentially increase the likelihood of resolving the different RF devices into distinct states.

Table 5.8: Power Specifications for WLAN Experiment 4

| | |
|---|---|
| CCK Power | 30 mW |
| OFDM Power | 30 mW |
| Max. Client Power | 5 mW |

**Initial Hyperparameters**

- $N_{max} = 5$

- State 1: $\lambda \sim \mathrm{Gamma}(k = 150, \theta = 2)$; $\overrightarrow{\mu}, \boldsymbol{\Sigma} \sim \mathrm{NIW}\left(\overrightarrow{\mu_0} = \overrightarrow{0}, \boldsymbol{\Lambda} = \mathbf{I}, \kappa = 300, \nu = 102\right)$

- State 2: $\lambda \sim \mathrm{Gamma}(k = 25, \theta = 2)$; $\overrightarrow{\mu}, \boldsymbol{\Sigma} \sim \mathrm{NIW}\left(\overrightarrow{\mu_0} = \overrightarrow{0}, \boldsymbol{\Lambda} = \mathbf{I}, \kappa = 3000, \nu = 302\right)$

- State 3: $\lambda \sim \mathrm{Gamma}(k = 30, \theta = 7)$; $\overrightarrow{\mu}, \boldsymbol{\Sigma} \sim \mathrm{NIW}\left(\overrightarrow{\mu_0} = \overrightarrow{0}, \boldsymbol{\Lambda} = \mathbf{I}, \kappa = 3000, \nu = 302\right)$

- State 4: $\lambda \sim \mathrm{Gamma}(k = 40, \theta = 8)$; $\overrightarrow{\mu}, \boldsymbol{\Sigma} \sim \mathrm{NIW}\left(\overrightarrow{\mu_0} = \overrightarrow{0}, \boldsymbol{\Lambda} = \mathbf{I}, \kappa = 3000, \nu = 302\right)$

- State 5: $\lambda \sim \mathrm{Gamma}(k = 100, \theta = 2)$; $\overrightarrow{\mu}, \boldsymbol{\Sigma} \sim \mathrm{NIW}\left(\overrightarrow{\mu_0} = \overrightarrow{0}, \boldsymbol{\Lambda} = \mathbf{I}, \kappa = 300, \nu = 102\right)$

- $\alpha = \gamma = \rho = 3$

- $N_{iter} = 250$ iterations

Table 5.9: File Transfer Sequence for WLAN Experiment 4

| |
|---|
| L1, A2, A5, A1, L5, L4, A4, L2, A6, A7 |

From examining the depiction of the learned HSMM superstate sequence in 5.4, an extra state (State 4) is shown to capture the first file transmission from the laptop as well as the transmission immediately succeeding the A5 transmission originating from the access point. This second transmission does not seem to correspond directly to any activity presented within the purposefully created traffic as presented in Table 5.9; correspondingly, it is unknown as to the origin of this transmitted signal or the type of traffic presented by this captured transmission. All known transmissions from the laptop are not consistently labeled as belonging to the same state upon the final iteration of the algorithm, and thus the initialization of each HSMM state with different hyperparameters seems to have little effect on achieving a more consistent labeling of the final HSMM state sequence.

**Experiment 4 State Legend**

- State 2 (Cyan): 802.11 Beacon Signals and Data Traffic Transmissions, having mean duration parameter $\lambda_1 = 1.4991$

- State 3 (Green): Idle State, having mean duration parameter $\lambda_2 = 142.4575$

- State 4 (Yellow): 802.11 Data Traffic Transmissions, having mean duration parameter $\lambda_3 = 114.2218$

- State 5 (Orange): 802.11 Data Traffic Transmissions, having mean duration parameter $\lambda_3 = 120.0889$

$$
\mathbf{A} = 
\begin{bmatrix}
 & 2 & 3 & 4 & 5 \\
2 & 0.0000 & 0.9012 & 0.0048 & 0.0939 \\
3 & 0.9996 & 0.0000 & 0.0002 & 0.0002 \\
4 & 0.9796 & 0.0204 & 0.0000 & 0.0000 \\
5 & 0.9898 & 0.0102 & 0.0000 & 0.0000
\end{bmatrix}
\tag{5.4}
$$
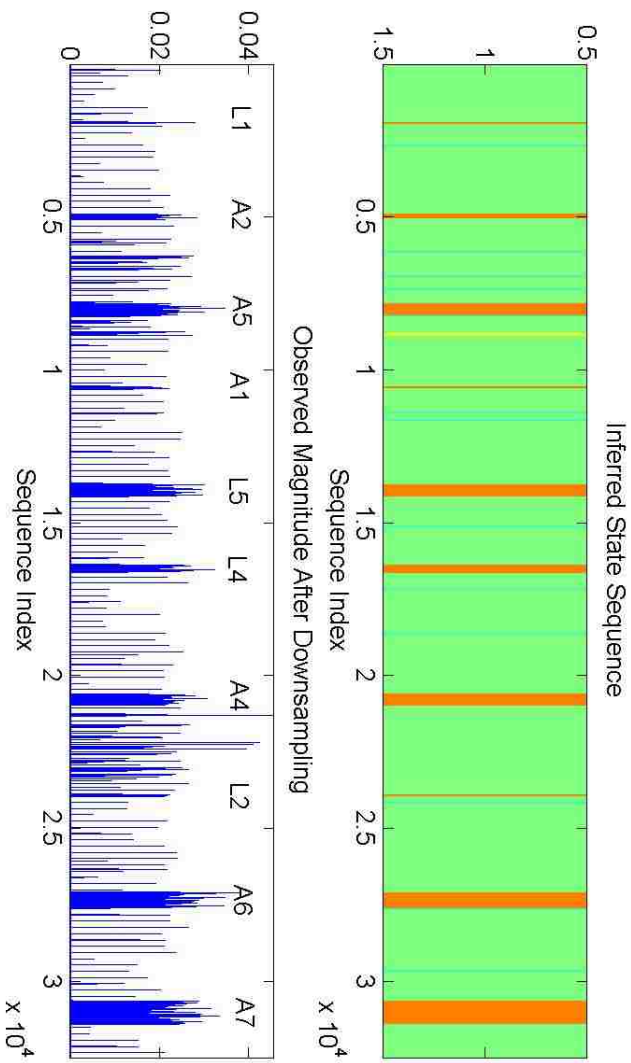
Figure 5.4: Labeled State Sequence for WLAN Experiment 4. Green = Idle State, Orange and Yellow = 802.11 Packets, Yellow and Cyan = Beacons.

# Chapter 6

# Future Work

The experimental results in Chapter 4 indicate that the HSMM algorithm can be utilized to approximate the number of wireless devices within an RF trace after providing coarse model assumptions about device behavior in terms of signal durations and received energy. A prominent feature of the algorithm is its ability to focus on *learning* the statistical features of the separate signal sources without creating sophisticated models for random effects introduced by the wireless channel or the underlying signal features of the received RF signals. This chapter will discuss several ideas for improving the algorithm to increase the accuracy of the inference process to discover more detailed information about the wireless networks responsible for the observed RF traces.

## 6.1 Algorithmic Improvements

As evidenced in Chapter 4 , the choice for modeling the separate state observation distributions by multivariate Gaussian distributions allows for coarse time domain segmentation of the RF traces according to spectral occupancies and vacancies over the wireless channel. Although the observations within the RF traces can hardly be assumed to have Gaussian statistical distributions, the algorithm does an admirable job of segmenting the data into distinct regions of interest without resorting to thorough experimental accumulation of the best statistical distributions for modeling the RF signal sources. A further line of inquiry for developing the HSMM algorithm should pursue whether more appropriate distributions with the desirable conjugacy relationship within the Bayesian framework should be incorporated to model the state emission process.

Characterization of the HSMM state emission process additionally considers the emissions to be independent and identically distributed over the course of each state's duration, thus neglecting *time domain* correlations in the data throughout each state's persistence. Such spectral methods for examining RF signal features has been a more thoroughly investigated and demonstrably successful approach for identifying RF devices in previous research applications. It is worth investigating the potential for adding this approach to modeling the emission distributions so that stronger conclusions can be made with regards to mapping discovered states to unique RF devices or packet collision instances.

Implementation of the algorithm within this thesis was also performed in an offline manner due to the large amount of data within the wireless RF traces. Since the behavior of

wireless traffic over the RF frequencies is highly time variant over long observation windows, the inference process needs modification for reducing the computational complexity so that the inferred model parameters can be quickly utilized by software defined radios for adapting their behavior as wireless traffic patterns change over time. Such a sequential treatment of the inference algorithm would thus ultimately broaden the impact of the algorithm's purpose to implement more efficient methods for practicably achieving better spectral usage efficiencies over the RF band.

The majority of the computational expense in the inference process involves the calculation of the backward message passage scheme described in Sections 3.1.1 and 3.2. A first method for reduce this computational expense would be to minimize the number of data points needed to sufficiently capture the temporal dynamics within the captured RF trace while preserving enough embedded information to statistically distinguish between distinct users. Another possibility would be to investigate how to modify the backward message passing calculations to be performed in parallel algorithms using a generic graphics processing unit (GPU) as opposed to a CPU. The highly parallelized structure of modern GPUs is generally more amenable to performing intensive computations much faster than even multi-core CPUs, and would thus provide a worthwhile candidate for expediting the HSMM learning process for the desired applications.[1]

# 6.2   Algorithm Performance Assessment

In [13] and [14], the performance of their developed HSMM algorithm is calculated in terms of computing the normalized Hamming error between the final learned HSMM state sequence with a corresponding state sequence serving as a ground truth model for the experimentally based data input into the algorithm. Such an error criterion serves as a quantitative means for identifying the fraction of time in which the labeled state sequence correctly identifies the appropriate device within the experimental data provided to the algorithm. The lack of direct synchronization between the coordinated USRP network and the autonomous interfering USRP within the experiments of Chapter 4 presents a unique challenge to creating an overall ground truth model; with further development of the GNU Radio programs used in the experimental generation process, it may be possible to develop such a ground truth model that aggregates the time-based behavior of the distinct acting RF devices into an overall state sequence that can be compared to the final learned HSMM state sequence. Upon completion of such a task, it would then be possible to quantitatively judge how well the algorithm can capture the temporal behavior for each distinct device as they appear over the course of repeated measurements by the observing USRP; then it

---

[1]The experiments within this thesis were performed using a desktop equipped with a Intel Xeon Quad Core W3565 CPU and 5.7 GB of random access memory. For a wireless RF trace consisting of roughly 32,000 complex data points, a typical application of the HSMM algorithm described in Chapter 4 encompassing 250 iterations required approximately 3 hours for completion. Obviously, there is indeed much room for improving the algorithm's computational structuring in terms of CPU execution.

would then be possible to also assess the convergence speed (i.e., number of iterations) at which the algorithm settles on a consistent labeling of the HSMM state sequence.

On the other hand, unidentified RF emissions from unanticipated RF devices with the frequency band of interest may arise within the course of data collection process as seen in Chapter 5. This primary difficulty again arises in temporally characterizing RF devices if there is *no definitive prior information* known about the number or behavior of such devices on the specified RF bandwidth of interest, since there can be no definitive ground truth model for the observed temporal behavior. In the absence of such a ground truth model, the final results of the HSMM algorithm should be assessed in terms of *consistency.* In other words, it is worth comparing the statistical properties amongst individual instances of each similarly labeled superstate in order to seek out potential instances of mislabeled regions within the HSMM labeling. Likewise, such instances can be compared between different states to determine if multiple state can be deemed statistically similar enough to be resolved into a smaller number of states. These sorts of comparison would involve developing some further nonparametric, empirically based probability density estimations.

## 6.3    Further Details about Wireless Network Inference

It is also very desirable to generalize the HSMM algorithm's learned parameters to develop more comprehensive understandings of the wireless networks responsible for driving the observed behavior within each collected RF trace. For example, mapping user-specific occupancy patterns across multiple RF traces may be used to discover how the observed RF devices behave within their operational roles of their specific wireless networks. An identification of transmitter/receiver pairs along with how specific users *react* to data transmissions over time could then motivate a more sophisticated means for improving spectral efficiency amongst distinct networks while minimizing interference between these networks as they compete to channel access within more congested portions of the RF spectrum.

In order to accomplish this task of wireless network discovery, some means must be provided to discern whether subsets of the observed RF devices seem to act in accordance to protocol-based behavior. Since wireless channel access protocols are designed to prevent simultaneous transmissions for multiple network devices, it is worthwhile to assess whether identified packet collisions from the HSMM algorithm can indicate the existence or absence of an underlying protocol governing the behavior of the wireless devices within the RF traces. Thus the inferred timing parameters and observation distributions from the HSMM algorithm should be developed into a general framework for assessing the level of coordination amongst the RF devices and their networks as they contend for access over the RF spectrum.

The requisite goal of wireless network inference using the HSMM algorithm is thus to assess the degree of coordination amongst the identified devices' transmissions within the input wireless RF traces. By examining the timing patterns in which a single user's transmissions occur with respect to the other users' transmissions distinguished within the

trace, we wish to infer whether the timing patterns over which several devices emerge are consistent with bidirectional communications in accordance with some specified protocol governing their behavior over time. Conversely, we may also seek to determine whether the transmission patterns between subsets of users appear in a random fashion, signifying a lack of common protocol regulating their temporal behavior or possibly even indicating the presence of purposefully introduced interference to disrupt the communications between network participants.

## 6.4 Experimental Considerations

As noted in the experimental results of the previous chapter, the inferred state sequences resultant from the HSMM algorithm seems to be dependent on the relative transmission powers of each RF device. Since the observed signal powers for each device is directly correlated to the distances from each RF signal source to the sensors collecting the RF measurements, it is also worthwhile to investigate how RF traces collected from multiple RF sensors may be aggregated to provide more reliable conclusions about the identity of each RF emitter in a more spatially diverse data collection process. Consequently, it is worthwhile to consider how information from multiple sensors can be fused together to provide inputs to the HSMM algorithm, or even perhaps have each sensor independently perform the HSMM inference algorithm in their own operating environment and aggregate their inferred model parameters together in a more cohesive manner. Thus as issue remains of achieving a consistent methods for labeling the observed HSMM states amongst several RF sensing devices.

The developed HSMM model for classifying RF users has also not considered more detailed fading and path loss characterizations introduced by the wireless channels of interest. A more complete implementation of the HSMM algorithm should account for a more realistic modeling of the time-variant features for the wireless channel itself and the corresponding effects that such time variance introduces to the data collection process. Since it is generally more difficult to calibrate RF measurements performed by the USRPs between devices, additional test scenarios should incorporate higher performance spectrum analyzers to collect the RF traces serving as input to the HSMM algorithm.

# Bibliography

[1] Ettus Research. Selecting an RF Daughterboard. `http://www.ettus.com/content/files/kb/Selecting_an_RF_Daughterboard.pdf`, accessed August 2012.

[2] Ettus Research. USRP B100 Bus Series. `https://www.ettus.com/content/files/06983_Ettus_B100_DS_Flyer_HR.pdf`, accessed August 2012.

[3] Spectrum Policy Task Force. "Spectrum policy task force report et docket no. 02-135". Technical report, U.S. Federal Communications Commission, 2002.

[4] E.B. Fox, E.B. Sudderth, M.I. Jordan, and A.S. Willsky. A Sticky HDP-HMM with Application to Speaker Diarization. *Annals of Applied Statistics*, 5(2A):1020–1056, 2011.

[5] Bela A. Frigyik, Amol Kapila, and Maya R. Gupta. "UWEETR-2010-0006: Introduction to the Dirichlet Distribution and Related Processes". Technical report, University of Washington Department of Electrical Engineering, 2010.

[6] S. Geirhofer, Lang Tong, and B.M. Sadler. Dynamic Spectrum Access in the Time Domain: Modeling and Exploiting White Space. *Communications Magazine, IEEE*, 45(5):66 –72, may 2007.

[7] Andrew Gelman, John B. Carlin, Hal S. Stern, and Donald D. Rubin. *Bayesian Data Analysis*. Chapman and Hall, first edition, 1995.

[8] GNU Radio Website. `http://www.gnuradio.org`, accessed August 2012.

[9] Yann Guédon. Estimating Hidden Semi-Markov Chains from Discrete Sequences. *Journal of Computational and Graphical Statistics*, 12(3):604–639, 2003.

[10] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning*. Springer, second edition, 2009.

[11] IEEE. IEEE Standard for Information Technology-Telecommunications and Information Exchange Between Systems-Local and Metropolitan Area Networks-Specific Requirements - Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications, 12 2007.

[12] J. Mitola III. *"Cognitive Radio: An Integrated Agent Architecture for Software Defined Radio"*. PhD thesis, Department of Teleinformatics, Royal Institute of Technology (KTH), Stockholm, Sweden, May 2000.

[13] M. J. Johnson and A. S. Willsky. Bayesian Nonparametric Hidden Semi-Markov Models. *ArXiv e-prints*, March 2012.

[14] Matthew Johnson and Alan S. Willsky. The Hierarchical Dirichlet Process Hidden Semi-Markov Model. In *UAI 2010*, pages 252–259, 2010.

[15] M. Lopez-Benitez and F. Casadevall. Discrete-Time Spectrum Occupancy Model based on Markov Chain and Duty Cycle Models. In *New Frontiers in Dynamic Spectrum Access Networks (DySPAN), 2011 IEEE Symposium on*, pages 90 –99, May 2011.

[16] John G. Proakis and Masoud Salehi. *Digital Communications.* McGraw Hill, fifth edition, 2008.

[17] Lawrence R. Rabiner. A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition. In *Proceedings of the IEEE*, pages 257–286, 1989.

[18] Gordon L. Stüber. *Principles of Mobile Communication.* Springer, third edition, 2011.

[19] Yee Whye Teh, Michael I. Jordan, Matthew J. Beal, and David M. Blei. "Hierarchical Dirichlet Processes". Technical report, University of California, Berkeley Department of Statistics, October 2004.

# Appendix A

# Explanation of Conjugate Pairs in the Bayesian HSMM Algorithm

The intent of this appendix is to provide a rough mathematical explanation of the conjugacy relationship between the prior distributions and likelihood functions for the distributions encountered in Chapter 4 as extrapolated from [7]. As discussed in section 3.3.1, the unnormalized posterior distribution $P(\theta|Y)$ can be computed as the product of the likelihood function $P(Y|\theta)$ and prior distribution $P(\theta)$.

$$P(\theta|Y) \propto P(Y|\theta)P(\theta) \tag{A.1}$$

Suppose that a collection of values $\overrightarrow{y} = [y_1, y_2, \ldots, y_n]^T$ for the random variable $Y$ are yielded as the results of observing some process whose parameters $\theta$ are sought.[1] The likelihood function for a single observation in $\overrightarrow{y}$ is said to belong to the exponential family of distributions if it can be expressed in (A.2), where $f(\cdot), g(\cdot), \phi(\cdot)$, and $u(\cdot)$ represent generic functions.

$$P(y_i|\theta) = f(y_i) \cdot g(\theta) \, \exp\{\phi(\theta)^T u(y_i)\}, \quad \forall\, i \tag{A.2}$$

For independent and identically distributed values of $y_i \in \overrightarrow{y}$, this functional form can be extended to the entire collection $\overrightarrow{y}$.

$$P(\overrightarrow{y}|\theta) = \left[\prod_{i=1}^{n} f(y_i)\right] g(\theta)^n \, \exp\left\{\phi(\theta)^T \cdot \sum_{i=1}^{n} u(y_i)\right\} \tag{A.3}$$

$$\propto g(\theta)^n \, \exp\left\{\phi(\theta)^T \cdot \sum_{i=1}^{n} u(y_i)\right\} \tag{A.4}$$

Since the likelihood for $\theta$ only depends on the values within $\overrightarrow{y}$ through the quantity $t(\overrightarrow{y}) \triangleq \sum_{i=1}^{n} u(y_i)$, it is said that $t(\overrightarrow{y})$ is a sufficient statistic for $\theta$.

Suppose the prior distribution $P(\theta)$ is also chosen to have the form

$$P(\theta) \propto g(\theta)^\eta \, \exp\left\{\phi(\theta)^T \cdot \nu\right\}. \tag{A.5}$$

The posterior distribution as calculated from (A.1) can also be seen to be a member of the

---

[1]$T$ represents the transposition operation as commonly used in linear algebra.

same exponential family expressed in (A.3).

$$P(\theta|\overrightarrow{y}) \propto g(\theta)^{\eta+n} \exp\left\{\phi(\theta)^T \cdot \left[\nu + t(\overrightarrow{y})\right]\right\} \tag{A.6}$$

Thus the chosen prior distribution and the likelihood function are conjugate to each other, since they belong to the same family of distributions. It should be noted that the support set for which the prior and posterior distributions are non-zero should be identical if proper conjugate priors are chosen.

## A.1  Conjugate Prior for a Poisson Likelihood

Suppose that the likelihood function has the form of a Poisson distribution with mean parameter $\lambda$ ($\theta = \lambda$) as shown in (A.7).

$$P(Y = y_i|\lambda) = \frac{\lambda^{y_i}}{y_i!} \exp\{-\lambda\}, \quad y_i \in \{0, 1, 2, \dots\}, \lambda > 0 \tag{A.7}$$

$$= \frac{1}{y_i!} \exp\{-\lambda\} \exp\{\log(\lambda^{y_i})\} \tag{A.8}$$

$$= \frac{1}{y_i!} \exp\{-\lambda\} \exp\{\log(\lambda) \cdot y_i\} \tag{A.9}$$

From inspection of (A.2), the Poisson distribution is a member of the exponential family with $f(y_i) = \frac{1}{x_i!}$, $g(\theta) = \exp\{-\lambda\}$, and sufficient statistic $t(y_i) = y_i$. If the mean parameter $\lambda$ is modeled by a gamma distribution ($\lambda \sim \text{Gamma}(k, \theta)$), the prior distribution will be expressed by (A.11) using a shape parameter $k$ and a scale parameter $\theta$.

$$P(\lambda) \sim \text{Gamma}(k, \theta), \quad k, \theta > 0 \tag{A.10}$$

$$= \frac{1}{\theta^k \, \Gamma(k)} \lambda^{k-1} \exp\left\{-\frac{\lambda}{\theta}\right\}, \quad \Gamma(k) = \int_0^\infty t^{k-1} \exp\{-t\} \, dt \tag{A.11}$$

$$\propto \lambda^{k-1} \exp\left\{-\frac{\lambda}{\theta}\right\}. \tag{A.12}$$

The calculation of the posterior distribution $P(\lambda|\overrightarrow{y})$ for a set of observations $\overrightarrow{y}$ is as follows.[2]

$$P(\lambda|\overrightarrow{y}) \propto \lambda^{\sum y_i} \exp\{-\lambda n\} \cdot \lambda^{k-1} \exp\left\{-\frac{\lambda}{\theta}\right\} \tag{A.13}$$

$$\propto \lambda^{(k+\sum y_i)-1} \exp\left\{-\frac{\lambda(1+n\theta)}{\theta}\right\} \tag{A.14}$$

$$P(\lambda|\overrightarrow{y}) \sim \text{Gamma}\left(k + \sum_{i=1}^{n} y_i \ , \ \frac{\theta}{1+n\theta}\right) \tag{A.15}$$

Comparison of (A.14) with (A.12) proves that the gamma distribution is the appropriate conjugate prior for a Poisson likelihood function. Obtaining a revised estimate of $\lambda$ can be generated by resampling from the initial gamma distribution with the update hyperparameters modified as in (A.16).

$$\begin{cases} k_n = k + \sum_{i=1}^{n} y_i \\ \theta_n = \frac{\theta}{1+n\theta} \end{cases} \tag{A.16}$$

## A.2  Conjugate Prior for a Univariate Gaussian Likelihood

### A.2.1  Unknown Mean, Known Variance

Consider the likelihood of a single scalar observation $y$ as following a Gaussian $\mathcal{N}(\mu, \sigma^2)$ distribution with known variance $\sigma^2$ (hence, $\theta = \mu$).

$$P(y|\theta) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left\{-\frac{1}{2}\frac{(y-\theta)^2}{\sigma^2}\right\} \tag{A.17}$$

The likelihood consequently is an exponential of quadratic form in $\theta$, entailing that the appropriate conjugate prior should also have this same form. In this case, the posterior distribution will then be the product of two exponentials of identical form, yielding another

---

[2]In the context of the presented algorithm, $\overrightarrow{y}$ would represent a collection of observed durations for for a single state after the HSMM state sequence labeling.

exponential form.

$$P(\theta) \propto \exp\{A\theta^2 + B\theta + C\} \tag{A.18}$$

$$\propto \exp\left\{-\frac{1}{2}\frac{(\theta - \mu_0)^2}{\tau_0^2}\right\} \tag{A.19}$$

$$P(\theta) \sim \mathcal{N}\left(\mu_0, \tau_0^2\right) \tag{A.20}$$

The posterior density can then be calculated as follows.

$$P(\theta|y) \propto \exp\left\{-\frac{1}{2}\left[\frac{(y - \theta)^2}{\sigma^2} + \frac{(\theta - \mu_0)^2}{\tau_0^2}\right]\right\} \tag{A.21}$$

$$\propto \exp\left\{-\frac{1}{2}\frac{(\theta - \mu_1)^2}{\tau_1^2}\right\} \tag{A.22}$$

$$P(\theta|y) \sim \mathcal{N}\left(\mu_1, \tau_1^2\right) \tag{A.23}$$

The equations in (A.24) define the new updated hyperparameters for drawing from the posterior distribution.

$$\begin{cases} \frac{1}{\tau_1^2} = \frac{1}{\tau_0^2} + \frac{1}{\sigma^2} \\ \mu_1 = \left(\frac{\tau_1^2}{\tau_0^2}\right)\mu_0 + \left(\frac{\tau_1^2}{\tau_0^2}\right)y \end{cases} \tag{A.24}$$

Extension to a set of observations $\overrightarrow{y}$ is obtained in a similar fashion.

$$P(\theta|\overrightarrow{y}) \propto P(\overrightarrow{y}|\theta)P(\theta) \tag{A.25}$$

$$= \prod_{i=1}^{n} P(y_i|\theta) \cdot P(\theta) \tag{A.26}$$

$$\propto \prod_{i=1}^{n} \exp\left\{-\frac{1}{2}\frac{(y_i - \theta)^2}{\sigma^2}\right\} \cdot \exp\left\{-\frac{1}{2}\frac{(\theta - \mu_0)^2}{\tau_0^2}\right\} \tag{A.27}$$

$$\propto \exp\left\{-\frac{1}{2}\left[\frac{\sum_{i=1}^{n}(y_i - \theta)^2}{\sigma^2} + \frac{(\theta - \mu_0)^2}{\tau_0^2}\right]\right\} \tag{A.28}$$

$$P(\theta|\overrightarrow{y}) \sim \mathcal{N}\left(\mu_n, \tau_n^2\right) \tag{A.29}$$

The equations in (A.30) once again define the updated hyperparameters for drawing from the posterior distribution.

$$\begin{cases} \frac{1}{\tau_n^2} = \frac{1}{\tau_0^2} + \frac{n}{\sigma^2} \\ \mu_n = \left(\frac{\tau_n^2}{\tau_0^2}\right)\mu_0 + \left(\frac{\tau_n^2}{\sigma^2}\right)n\bar{y} \quad \text{with } \bar{y} \triangleq \frac{1}{n}\sum_{i=1}^{n} y_i \end{cases} \tag{A.30}$$

## A.2.2 Known Mean, Unknown Variance

For a a collection of values $\overrightarrow{y} = [y_1, y_2, \ldots, y_n]^T$, the likelihood function is given by (A.31).

$$P(\overrightarrow{y}|\theta) = \left(2\pi\sigma^2\right)^{-n/2} \cdot \exp\left\{-\frac{1}{2\sigma^2}\sum_{i=1}^{n}(y_i - \mu)^2\right\} \tag{A.31}$$

For the prior distribution $P(\sigma^2)$, we choose a Gamma($a$,$b$) prior with $\lambda = \frac{1}{\sigma^2}$, which is more commonly known as an inverse-gamma distribution for the variable $\sigma^2$.

$$P(\sigma^2) = \frac{b^a}{\Gamma(a)}\lambda^{a-1}\left(\sigma^2\right)^{-(a-1)}\exp\left\{-\frac{b}{\sigma^2}\right\} \tag{A.32}$$

$$P(\sigma^2) \sim \text{Inv-Gamma}(a, b) \tag{A.33}$$

Consequently, the posterior distribution is computed as follows.

$$P(\theta|\overrightarrow{y}) \propto \left(\sigma^2\right)^{-n/2}\exp\left\{-\frac{1}{2\sigma^2}\sum_{i=1}^{n}(y_i - \mu)^2\right\} \cdot \left(\sigma^2\right)^{-(a-1)}\exp\left\{-\frac{b}{\sigma^2}\right\} \tag{A.34}$$

$$\propto \left(\sigma^2\right)^{-[(a+n/2)-1]} \cdot \exp\left\{-\frac{\sum(y_i - \mu)^2 + 2b}{2\sigma^2}\right\} \tag{A.35}$$

$$P(\theta|\overrightarrow{y}) \sim \text{Inv-Gamma}\left(a + \frac{n}{2}, b + \frac{1}{2}\sum_{i=1}^{n}(y_i - \mu)^2\right) \tag{A.36}$$

Thus, the inverse gamma distribution is the conjugate prior for a univariate Gaussian likelihood function with known mean and unknown variance, with the updated hyperparameters defined in (A.37).

$$\begin{cases} a_n = a + \frac{n}{2} \\ b_n = b + b + \frac{1}{2}\sum_{i=1}^{n}(y_i - \mu)^2 \end{cases} \tag{A.37}$$

## A.2.3 Unknown Mean and Unknown Variance

The conjugate prior for a collection of univariate Gaussian observations with unknown mean and unknown variance can be obtained by appropriately combining the prior of Sections A.2.1 and A.2.2. As opposed to the previous cases, $P(\theta) = P(\mu, \sigma^2)$ is a joint distribution of two random variables. First, suppose the marginal distribution $P(\sigma^2)$ is modeled by an

inverse gamma distribution according to (A.39).

$$P(\sigma^2) \sim \text{Inv-Gamma}(\nu_0, \sigma_0^2) \tag{A.38}$$

$$= \frac{(\sigma_0^2)^{\nu_0}}{\Gamma(\nu_0)} \lambda^{\nu_0 - 1} \left(\sigma^2\right)^{-(\nu_0 - 1)} \exp\left\{-\frac{\sigma_0^2}{\sigma^2}\right\} \tag{A.39}$$

Then the conditional distribution $P(\mu|\sigma^2)$ can be modeled a Gaussian distribution with a rescaled variance $\sigma^2/\kappa_0$.

$$P(\mu|\sigma^2) \sim \mathcal{N}(\mu_0, \sigma^2/\kappa_0) \tag{A.40}$$

$$= \frac{\kappa_0}{2\pi\sigma_0^2} \exp\left\{-\frac{\kappa_0}{2}\frac{(\mu - \mu_0)^2}{\sigma_0^2}\right\} \tag{A.41}$$

The prior distribution $P(\theta)$ can then be expressed as the product of (A.39) and (A.41) in what is termed a normal-scaled inverse gamma (NIG) distribution.

$$P(\theta) = P(\mu, \sigma_2) \tag{A.42}$$
$$= P(\mu|\sigma^2)\,P(\sigma^2) \tag{A.43}$$
$$\propto \frac{\sqrt{\kappa_0}}{\sqrt{2\pi\sigma_0^2}}(\sigma^2)^{-(\nu_0+1)} \exp\left\{-\frac{2\sigma_0^2 + \kappa_0(\mu - \mu_0)^2}{2\sigma^2}\right\} \tag{A.44}$$

Using (A.31), the posterior distribution $P(\overrightarrow{y}|\theta)$ can be calculated as follows. After algebraically rearranging the terms of (A.45), it is can be seen that the NIG distribution is the conjugate prior for iid observations of a Gaussian distribution of unknown mean and variance.

$$P(\overrightarrow{y}|\theta) \propto \left(2\pi\sigma^2\right)^{-n/2} \exp\left\{-\frac{1}{2\sigma^2}\sum_{i=1}^{n}(y_i - \mu)^2\right\}$$

$$\cdot \frac{\sqrt{\kappa_0}}{\sqrt{2\pi\sigma_0^2}}(\sigma^2)^{-(\nu_0+1)} \exp\left\{-\frac{2\sigma_0^2 + \kappa_0(\mu - \mu_0)^2}{2\sigma^2}\right\} \tag{A.45}$$

$$\sim \text{NIG}\left(\mu_n, \kappa_n, \nu_n, \sigma_n^2\right) \tag{A.46}$$

The updated hyperparameters for drawing from the posterior distribution are as follows in (A.47).

$$\begin{cases} \kappa_n = \kappa_0 + n \\ \mu_n = \left(\kappa_0/\kappa_n\right)\mu_0 + \left(n/\kappa_n\right)\bar{y} \\ \nu_n = \nu_0 + n \\ \sigma_n^2 = \left(\nu_0/\nu_n\right)\sigma_0^2 + \frac{1}{\nu_n}\left[\mu_0^2\kappa_0 - \mu_n^2\kappa_n + \sum_{i=1}^{n} y_i^2\right] \end{cases} \tag{A.47}$$

## A.3 Conjugate Prior for a Multivariate Gaussian Likelihood

Performing posterior inference for a collection of $n$ $k$-dimensional iid multivariate Gaussian observations $\mathcal{Y} = \{\overrightarrow{y_1}, \overrightarrow{y_2}, \ldots, \overrightarrow{y_n}\}$ is very similar to the approach used throughout section A.2 except extrapolated to multiple dimensions. The likelihood for one multivariate Gaussian observation with mean vector $\overrightarrow{\mu}$ and covariance matrix $\mathbf{\Sigma}$ ($\theta = \{\overrightarrow{\mu}, \mathbf{\Sigma}\}$).[3]

$$P(\overrightarrow{y}|\theta) = (2\pi)^{-k/2}\,|\mathbf{\Sigma}|^{-1/2}\,\exp\left\{-\frac{1}{2}(\overrightarrow{y} - \overrightarrow{\mu})^T\mathbf{\Sigma}^{-1}(\overrightarrow{y} - \overrightarrow{\mu})\right\} \tag{A.48}$$

For a collection of $n$ iid vectors $\overrightarrow{y_i}$, the likelihood function becomes the product of the $n$ marginal distributions in (A.48).

$$P(\mathcal{Y}) = (2\pi)^{-kn/2}\,|\mathbf{\Sigma}|^{-n/2}\,\exp\left\{-\frac{1}{2}\sum_{i=1}^{n}(\overrightarrow{y_i} - \overrightarrow{\mu})^T\mathbf{\Sigma}^{-1}(\overrightarrow{y_i} - \overrightarrow{\mu})\right\} \tag{A.49}$$

$$\propto |\mathbf{\Sigma}|^{-n/2}\,\exp\left\{\frac{1}{2}\mathrm{tr}\left(\mathbf{\Sigma}^{-1}S\right)\right\}, \quad S \triangleq \sum_{i=1}^{n}(\overrightarrow{y_i} - \overrightarrow{\mu})(\overrightarrow{y_i} - \overrightarrow{\mu})^T \tag{A.50}$$

$$\tag{A.51}$$

In (A.50), $\mathrm{tr}(\cdot)$ represents the trace operation, while $S$ is called the scatter matrix showing the sum of squares around the mean vector $\overrightarrow{\mu}$.

Supposing that both the mean vector and covariance matrix are unknown, the analysis for the proper conjugate prior is very similar to the procedure detailed earlier in Section A.2.3. The first task is to generate a sample for the covariance matrix $\mathbf{\Sigma}$ from an unnormalized inverse Wishart (IW) distribution, which is a multidimensional extension of the inverse gamma distribution introduced earlier.

$$P(\mathbf{\Sigma}) \propto |\mathbf{\Sigma}|^{-(\nu_0+k+1)/2}\,\exp\left\{-\frac{1}{2}\mathrm{tr}\left(\mathbf{\Lambda}_0\mathbf{\Sigma}_0^{-1}\right)\right\} \tag{A.52}$$

$$\sim \mathrm{IW}(\nu_0, \mathbf{\Lambda}_0^{-1}) \tag{A.53}$$

The conditional distribution $P(\overrightarrow{\mu}|\mathbf{\Sigma})$ is then modeled as a multivariate Gaussian according to (A.48) with a rescaled covariance matrix.

$$P(\overrightarrow{\mu}|\mathbf{\Sigma}) \sim \mathcal{N}\left(\overrightarrow{\mu_0}, \mathbf{\Sigma}/\kappa_0\right) \tag{A.54}$$

$$\propto |\mathbf{\Sigma}|^{-1/2}\,\exp\left\{-\frac{\kappa_0}{2}(\overrightarrow{\mu} - \overrightarrow{\mu_0})^T\mathbf{\Sigma}^{-1}(\overrightarrow{\mu} - \overrightarrow{\mu_0})\right\} \tag{A.55}$$

---

[3]$|\mathbf{\Sigma}|$ denotes the determinant of the matrix $\mathbf{\Sigma}$

The prior distribution $P(\theta)$ is then the product of $P(\boldsymbol{\Sigma})$ and $P(\overrightarrow{\mu}|\boldsymbol{\Sigma})$, resulting in an normal inverse Wishart (NIW) distribution as seen in (A.59).[4]

$$P(\theta) = P(\mu, \boldsymbol{\Sigma}) \tag{A.56}$$

$$= P(\overrightarrow{\mu}|\boldsymbol{\Sigma}) \cdot P(\boldsymbol{\Sigma}) \tag{A.57}$$

$$\propto |\boldsymbol{\Sigma}|^{-(\nu_0+k+1)/2} \exp\left\{ -\frac{1}{2}\mathrm{tr}\big(\boldsymbol{\Lambda}_0\boldsymbol{\Sigma}_0^{-1}\big) \right\}$$

$$\cdot |\boldsymbol{\Sigma}|^{-1/2} \exp\left\{ -\frac{\kappa_0}{2}(\overrightarrow{\mu} - \overrightarrow{\mu_0})^T\boldsymbol{\Sigma}^{-1}(\overrightarrow{\mu} - \overrightarrow{\mu_0}) \right\} \tag{A.58}$$

$$\propto |\boldsymbol{\Sigma}|^{-[(\nu_0+k)/2+1]} \exp\left\{ -\frac{1}{2}\mathrm{tr}\big(\boldsymbol{\Lambda}_0\boldsymbol{\Sigma}_0^{-1}\big) - \frac{\kappa_0}{2}(\overrightarrow{\mu} - \overrightarrow{\mu_0})^T\boldsymbol{\Sigma}^{-1}(\overrightarrow{\mu} - \overrightarrow{\mu_0}) \right\} \tag{A.59}$$

$$P(\theta) \sim \mathrm{NIW}(\overrightarrow{\mu_0}, \kappa_0, \nu_0, \boldsymbol{\Lambda}_0) \tag{A.60}$$

The hyperparameters $\overrightarrow{\mu_0}$ and $\kappa_0$ are used to express the mean value and variance of each component for the generated mean vector $\overrightarrow{\mu}$. Generating a positive definite covariance matrix $\boldsymbol{\Sigma}$ requires two hyperparameters, a degrees-of-freedom parameter $\nu_0 \geq k - 1$ and a positive semidefinite matrix $\boldsymbol{\Lambda}_0$. As $\nu_0 \to \infty$ close to its minimum value of $k - 1$ generates nearly singular matrices, while decreasing $\nu_0$ places more emphasis on the scale matrix $\boldsymbol{\Lambda}_0$.

$$P(\theta) = \mathrm{Pr}(\boldsymbol{\Sigma}) \cdot \mathrm{Pr}(\overrightarrow{\mu}|\boldsymbol{\Sigma}) \tag{A.61}$$

$$\sim \mathrm{IW}_{\nu_0}\big(\Lambda_0^{-1}\big) \cdot \mathcal{N}\big(\overrightarrow{\mu}_0, \boldsymbol{\Sigma}/\kappa_0\big) \tag{A.62}$$

$$\propto |\boldsymbol{\Sigma}|^{-(\nu_0+k+1)/2} \exp\left\{ -\frac{1}{2}\mathrm{tr}\big(\boldsymbol{\Lambda}_0\boldsymbol{\Sigma}_0^{-1}\big) \right\}$$

$$\cdot |\boldsymbol{\Sigma}|^{-1/2} \exp\left\{ -\frac{\kappa_0}{2}(\overrightarrow{\mu} - \overrightarrow{\mu_0})^T\boldsymbol{\Sigma}^{-1}(\overrightarrow{\mu} - \overrightarrow{\mu_0}) \right\} \tag{A.63}$$

$$P(\theta) \propto |\boldsymbol{\Sigma}|^{[(\nu_0+k)/2+1]} \exp\left\{ -\frac{1}{2}\mathrm{tr}\big(\boldsymbol{\Lambda}_0\boldsymbol{\Sigma}_0^{-1}\big) - \frac{\kappa_0}{2}(\overrightarrow{\mu} - \overrightarrow{\mu_0})^T\boldsymbol{\Sigma}^{-1}(\overrightarrow{\mu} - \overrightarrow{\mu_0}) \right\} \tag{A.64}$$

The posterior distribution $P(\theta|\mathcal{Y})$ can then be derived as follows.

$$P(\theta|\mathcal{Y}) \propto P(\mathcal{Y}|\theta) \cdot P(\theta) \tag{A.65}$$

$$\propto |\boldsymbol{\Sigma}|^{-n/2} \exp\left\{ \frac{1}{2}\mathrm{tr}\big(\boldsymbol{\Sigma}^{-1}S\big) \right\}$$

$$\cdot |\boldsymbol{\Sigma}|^{-[(\nu_0+k)/2+1]} \exp\left\{ -\frac{1}{2}\mathrm{tr}\big(\boldsymbol{\Lambda}_0\boldsymbol{\Sigma}_0^{-1}\big) - \frac{\kappa_0}{2}(\overrightarrow{\mu} - \overrightarrow{\mu_0})^T\boldsymbol{\Sigma}^{-1}(\overrightarrow{\mu} - \overrightarrow{\mu_0}) \right\} \tag{A.66}$$

$$P(\theta|\mathcal{Y}) \sim \mathrm{NIW}(\overrightarrow{\mu_n}, \kappa_n, \nu_n, \boldsymbol{\Lambda}_n) \tag{A.67}$$

---

[4]Note that this distribution is unnormalized. As mentioned previously in Chapter 3, the omission of the normalization constant does not affect the inference process.

As a result, the NIW distribution is seen to serve as a conjugate prior for likelihood functions modeled by a multivariate Gaussian distribution. After computing the posterior distribution according to (A.66), the updated values for the hyperparameters can be seen in (A.68).

$$
\begin{cases}
\kappa_n & = \kappa_0 + n \\
\overrightarrow{\mu_n} & = (\kappa_0/\kappa_n)\overrightarrow{\mu_0} + (n/\kappa_n)\bar{y} \\
\nu_n & = \nu_0 + n \\
\Lambda_n & = \Lambda_0 + S + (\kappa_0\,n/\kappa_n)(\bar{y} - \overrightarrow{\mu_0})^T(\bar{y} - \overrightarrow{\mu_0})
\end{cases}
\tag{A.68}
$$

In these formulations, $\bar{x}$ is actually a vector representing the empirical mean along each dimension of these multivariate random variables ($\bar{y}_j = \frac{1}{n}\sum_{i=1}^{n} y_{i,j} \ \forall \ j = 1, 2, \ldots, k$).

# A.4   Conjugate Prior for a Multinomial Likelihood

Suppose that some experiment yields one of $N$ outcomes at each trial, with the probability of each event $i$ occurring given by a probability mass function $\overrightarrow{p} = [p_1, p_2, \ldots, p_N]$. After $n$ trials of such an experiment, each possible event $i \in \{1, 2, \ldots, N\}$ will have occurred $n_i$ times, with $\sum_{i=1}^{N} n_i = n$. The probability that each possible event occurs with the observed frequency can be described by a multinomial distribution given in (A.69).

$$
p(n_1, n_2, \ldots, n_N | \overrightarrow{p}, n) = \frac{n!}{n_1! n_2! \cdots n_N!} \prod_{i=1}^{N} p_i^{n_i}
\tag{A.69}
$$

As mentioned in Section 3.3.2, the Dirichlet distribution serves as a conjugate prior for a multinomial likelihood, as detailed in the following analysis, where $\overrightarrow{n} = [n_1, n_2, \ldots, n_N]$.

$$
P(\theta | \overrightarrow{n}) = P(\overrightarrow{n} | \theta) \cdot P(\theta)
\tag{A.70}
$$

$$
= \frac{n!}{n_1! n_2! \cdots n_N!} \prod_{i=1}^{N} p_i^{n_i} \cdot \frac{\Gamma(\alpha_0)}{\prod_{i=1}^{N} \Gamma(\alpha_i)} \prod_{i=1}^{N} p_i^{\alpha_i - 1}
\tag{A.71}
$$

$$
\propto \prod_{i=1}^{N} p_i^{n_i} \cdot \prod_{i=1}^{N} p_i^{\alpha_i - 1}
\tag{A.72}
$$

$$
\propto \prod_{i=1}^{N} p_i^{\alpha_i + n_i - 1}
\tag{A.73}
$$

$$
\sim \mathrm{Dir}(\overrightarrow{\alpha} + \overrightarrow{n})
\tag{A.74}
$$

# Appendix B

# Relevant Information from Measure Theory

A full understanding for the concepts introduced in Sections 3.3.2 and 3.3.3 require a short introduction to a more axiomatic description of probability and measure theory.[1] In a rough sense, probability is primarily focused on providing a framework for quantifying the expected result of some random experiment, with more likely experimental outcomes possessing a greater value than other less likely outcomes. For such an experiment, each performance will yield one outcome out of the set of all potential outcomes known as the sample space $\mathcal{X}$. Alternatively, one can consider computing the probability for events, or subsets of the sample space. As a quick example, one roll of a regular six-sided die will result in one of six potential outcomes for the number on the top face of the die, resulting in a sample space $\mathcal{X} = \{1, 2, 3, 4, 5, 6\}$ for one experimental trial. In this case, each distinct element in the sample space is considered the event of interest. It may also be desired to define an event as the resultant number on the top face being odd; computing the probability of such an event would then encompass somehow aggregating the probabilities for observing 1, 3, or 5.

A measureable space $(\mathcal{X}, \mathcal{F})$ consists of the experiment's sample space and a number of subsets $\mathcal{F}$ on that sample space known as a $\sigma$-algebra. A $\sigma$-algebra is defined as having the following properties:

1. $\mathcal{F} \neq \emptyset$. ($\mathcal{F}$ is not empty.)

2. If $B \in \mathcal{F}$ , then $B^c \in \mathcal{F}$. In other words, each set in the $\sigma$-algebra must also possess its complement within the $\sigma$-algebra as well.

3. If $\{B_i\}_{i=1}^{\infty} \in \mathcal{F}$, then $\bigcup_{i=1}^{\infty} B_i \in \mathcal{F}$. Thus for every countable collection of sets in the $\sigma$-algebra, the union of every set in that collection must also reside in the $\sigma$-algebra.

Through these three stipulations, it follows that the smallest possible $\sigma$-algebra is $\mathcal{F} = \{\emptyset, \mathcal{X}\}$. On the other hand, the largest possible $\sigma$-algebra contains every potential subset of the sample space $\mathcal{X}$.

It is also useful to define a measure $\mu$ that assigns a non-negative real number to every member within the $\sigma$-algebra $\mathcal{F}$. A measure $\mu : \mathcal{F} \to [0, \infty)$ satisfies the following properties:

---

[1]For a more formal and thorough treatment of these subjects, one can consult *Probability and Measure* by Patrick Billingsley or *Probability: A Graduate Course* by Allan Gut.

1. For all $F \in \mathcal{F}$, $\mu(F) \geq 0$.

2. The measure of the empty set is $\mu(\emptyset) = 0$.

3. For a countable collection of disjoint sets $\{F_i\}_{i \in A} \in \mathcal{F}$ with $F_i \cap F_j = \emptyset$ for $i \neq j$, the measure of the union of those sets is equal to the sum of the measures for each individual set $F_i$.

$$\mu\left(\bigcup_{i \in A} F_i\right) = \sum_{i \in A} \mu(F_i) \tag{B.1}$$

A probability measure $P$ also has the additional property that $P(\mathcal{X}) = 1$. Thus, a probability space $(\mathcal{X}, \mathcal{F}, P)$ is an ordered triple used to calculate the probability for the potential events arising from some experiment.

# Appendix C

# Generating Samples from a Dirichlet Distribution

Many numerical computing software environments, such as MATLAB, include native functions for drawing from commonly used probability distributions including the beta and gamma distribution, yet do not possess such functions for drawing from Dirichlet distributions. This appendix will consider two different techniques for drawing Dirichlet distributions using these more common distributions. The first technique, known as the stick breaking procedure, provides an intuitive way to visualize the creation of Dirichlet distributions through an iterative process. The second technique involving gamma distributions is also included, due to its more computationally efficient structure. A more detailed discussion of these procedures and their derivation can be found in [5].

## C.1  Stick-Breaking Procedure

Suppose that a random variable $X$ is drawn from a beta distribution ($X \sim \text{Beta(a,b)}$). The probability distribution for the beta distribution is given in (C.1), where $a$ and $b$ represent two distinct shape parameters.

$$P(x|a,b) = \frac{\Gamma(a+b)}{\Gamma(a)\Gamma(b)} \, x^{a-1} \, (1-x)^{b-1}, \quad x \in [0,1]; \, a,b > 0 \tag{C.1}$$

As previously discussed in Section 3.3.2, a random probability mass function (PMF) $\overrightarrow{p}$ can be drawn from a Dirichlet distribution such that $\overrightarrow{p} \sim \text{Dir}(\overrightarrow{\alpha})$.

$$P(\overrightarrow{p}|\overrightarrow{\alpha}) = \frac{\Gamma(\alpha_0)}{\prod_{i=1}^{N} \Gamma(\alpha_i)} \prod_{i=1}^{N} p_i^{\alpha_i - 1} \tag{C.2}$$

$$\propto \prod_{i=1}^{N} p_i^{\alpha_i - 1} \tag{C.3}$$

From examining (C.1) and (C.2), it can be seen that a two-dimensional probability mass function $\overrightarrow{p} = [p_1, p_2] \sim \text{Dir}([a,b])$ reduces to a beta distribution $\text{Beta}(a,b)$. Thus, the constraint $\sum_{i=1}^{N} p_i = 1$ for an $N$-component PMF suggests that an iterative procedure can be specified to generate a Dirichlet distribution from drawing from several linked beta distributions.

It is convenient to envision the Dirichlet distribution as breaking a stick of unit length into $N$ components, with each component having a random length. The stick breaking procedure is briefly examined in the following outline.

1. Draw a value $u_1 \sim \text{Beta}(\alpha_1, \sum_{j=2}^{N} \alpha_j)$ and set $p_1 = u_1$. At this point, one component of length $p_1$ has been broken from the stick with $1 - u_1 = 1 - p_1$ of the stick remaining.

2. For $2 \leq i \leq N - 1$, $i - 1$ components, each with random length, have been removed from the stick. As an extension of the first step, $\prod_{j=1}^{i-1}(1 - u_j)$ will remain on the stick after $i$ segmentations. Thus, draw $u_i \sim \text{Beta}(\alpha_i, \sum_{j=i+1}^{N} \alpha_j)$ and set $p_i = u_i \prod_{j=1}^{i-1}(1 - u_j)$.

3. For $i = N$, the remainder of the stick will be assigned to $p_N$, with $p_N = 1 - \sum_{j=1}^{N-1} p_j$.

## C.2    Generating Dirichlet Distributions from Gamma Random Variables

A random variable $X$ drawn from a gamma distribution ($X \sim \text{Gamma}(k, \theta)$) has a probability distribution function given by (C.4) as follows.

$$P(x|k,\theta) = \frac{1}{\theta^k \, \Gamma(k)} x^{a-1} \exp\left\{ -\frac{x}{\theta} \right\}, \quad x > 0; \, k, \theta > 0 \tag{C.4}$$

If $X_i \sim \text{Gamma}(k_i, \theta)$ are marginally independent for $i = 1, \ldots, N$, then the joint probability distribution for $(X_1, \ldots, X_N)$ is a product of the $N$ marginal distribution from (C.4), each with the same scale parameter $\theta$ but distinct shape parameters $k_i$.

$$P(x1, \ldots, x_N | k_1, \ldots, k_n, \theta) = \frac{1}{\theta^{\sum_i k_i} \, \Gamma(\sum_i k_i)} \left( \sum_{i=1}^{N} x_i \right)^{(\sum_i k_i - 1)} \exp\left\{ -\frac{\sum_{i=1}^{N} x_i}{\theta} \right\} \tag{C.5}$$

Examining (C.5), it can be seen that $\sum_{i=1}^{N} X_i \sim \text{Gamma}\left( \sum_{i=1}^{N} k_i, \theta \right)$. By setting $\theta = 1$, it is possible to draw an $N$-component Dirichlet distribution using the gamma distribution through the procedure outlined below.

1. For $i = 1, \ldots, N$, $u_i \sim \text{Gamma}(\alpha_i, 1)$.

2. For $i = 1, \ldots, N$, $p_i = \frac{u_i}{\sum_{i=1}^{N} u_i}$.

The second step is needed to renormalize the drawn components to satisfy the PMF requirement that $\sum_{i=1}^{N} p_i = 1$.

# Appendix D

# Software Programs

## D.1   Acquiring Wireless Traces Using the USRP

```
#!/usr/bin/env python
#
# Copyright 2012 Free Software Foundation, Inc.
#
# This file is part of GNU Radio
#
# GNU Radio is free software; you can redistribute it and/or modify
# it under the terms of the GNU General Public License as published by
# the Free Software Foundation; either version 3, or (at your option)
# any later version.
#
# GNU Radio is distributed in the hope that it will be useful,
# but WITHOUT ANY WARRANTY; without even the implied warranty of
# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
# GNU General Public License for more details.
#
# You should have received a copy of the GNU General Public License
# along with GNU Radio; see the file COPYING.  If not, write to
# the Free Software Foundation, Inc., 51 Franklin Street,
# Boston, MA 02110-1301, USA.
#

"""
Read samples from a UHD device and write to file formatted as binary
outputs single precision complex float values or complex short values
(interleaved 16 bit signed short integers).
"""

################################################################################
Usage: ./uhd_rx_cfile -f center_freq --samp-rate sampling_rate
                                     --args "usrp_identifier" output_file

Example: ./uhd_rx_cile -f 2.422G --samp-rate 500k --args "name=usb_usrp1"
```

wireless_trace.dat

```
################################################################################
from gnuradio import gr, gru, eng_notation
from gnuradio import uhd
from gnuradio.eng_option import eng_option
from optparse import OptionParser
import sys


n2s = eng_notation.num_to_str


class rx_cfile_block(gr.top_block):

    def __init__(self, options, filename):
        gr.top_block.__init__(self)

        # Create a UHD device source
        if options.output_shorts:
            self._u = uhd.usrp_source(device_addr=options.args,
                                      stream_args=uhd.stream_args('sc16',
                                      options.wire_format, args=options.stream_args))
            self._sink = gr.file_sink(gr.sizeof_short*2, filename)
        else:
            self._u = uhd.usrp_source(device_addr=options.args,
                                      stream_args=uhd.stream_args('fc32',
                                      options.wire_format, args=options.stream_args))
            self._sink = gr.file_sink(gr.sizeof_gr_complex, filename)

        # Set the subdevice spec
        if(options.spec):
            self._u.set_subdev_spec(options.spec, 0)

        # Set the antenna
        if(options.antenna):
            self._u.set_antenna(options.antenna, 0)

        # Set receiver sample rate
        self._u.set_samp_rate(options.samp_rate)

        # Set receive daughterboard gain
        if options.gain is None:
            g = self._u.get_gain_range()
            options.gain = float(g.start()+g.stop())/2
    print "Using mid-point gain of", options.gain, "(", g.start(), "-", g.stop(), ")"
        self._u.set_gain(options.gain)
```

```
# Set frequency (tune request takes lo_offset)
if(options.lo_offset is not None):
    treq = uhd.tune_request(options.freq, options.lo_offset)
else:
    treq = uhd.tune_request(options.freq)
tr = self._u.set_center_freq(treq)
if tr == None:
    sys.stderr.write('Failed to set center frequency\n')
    raise SystemExit, 1

# Create head block if needed and wire it up
if options.nsamples is None:
    self.connect(self._u, self._sink)
else:
    if options.output_shorts:
        self._head = gr.head(gr.sizeof_short*2, int(options.nsamples))
    else:
        self._head = gr.head(gr.sizeof_gr_complex, int(options.nsamples))

    self.connect(self._u, self._head, self._sink)

input_rate = self._u.get_samp_rate()

if options.verbose:
    try:
        info = self._u.get_usrp_info()

        mboard_id = info.get("mboard_id").split(" ")[0]
        if info.get("mboard_serial") == "":
            mboard_serial = "no serial"
        else:
            mboard_serial = info.get("mboard_serial")

        rx_id = info.get("rx_id").split(" ")[0]
        if info.get("rx_serial") == "":
            rx_serial = "no serial"
        else:
            rx_serial = info.get("rx_serial")
        rx_antenna = info.get("rx_antenna")
        rx_subdev_spec = info.get("rx_subdev_spec")

        print "Motherboard: %s (%s)" % (mboard_id, mboard_serial) \
        print "Daughterboard: %s (%s, %s, %s)"  \
                          % (rx_id, rx_serial, rx_antenna, rx_subdev_spec)
```

```python
            except:
                print "Args: ", options.args
            print "Rx gain:", options.gain
            print "Rx baseband frequency:", n2s(tr.actual_rf_freq)
            print "Rx DDC frequency:", n2s(tr.actual_dsp_freq)
            print "Rx Sample Rate:", n2s(input_rate)
            if options.nsamples is None:
                print "Receiving samples until Ctrl-C"
            else:
                print "Receving", n2s(options.nsamples), "samples"
            if options.output_shorts:
                print "Writing 16-bit complex shorts"
            else:
                print "Writing 32-bit complex floats"
            print "Output filename:", filename

        # Direct asynchronous notifications to callback function
        if options.show_async_msg:
            self.async_msgq = gr.msg_queue(0)
            self.async_src = uhd.amsg_source("", self.async_msgq)
            self.async_rcv = gru.msgq_runner(self.async_msgq, self.async_callback)

    def async_callback(self, msg):
        md = self.async_src.msg_to_async_metadata_t(msg)
        print "Channel: %i Time: %f Event: %i"  \
                    % (md.channel, md.time_spec.get_real_secs(), md.event_code)


def get_options():
    usage="%prog: [options] output_filename"
    parser = OptionParser(option_class=eng_option, usage=usage)
    parser.add_option("-a", "--args", type="string", default="",
                      help="UHD device address args , [default=%default]")
    parser.add_option("", "--spec", type="string", default=None,
                      help="Subdevice of UHD device where appropriate")
    parser.add_option("-A", "--antenna", type="string", default=None,
                      help="select Rx Antenna where appropriate")
    parser.add_option("", "--samp-rate", type="eng_float", default=1e6,
                      help="set sample rate (bandwidth) [default=%default]")
    parser.add_option("-f", "--freq", type="eng_float", default=None,
                      help="set frequency to FREQ", metavar="FREQ")
    parser.add_option("-g", "--gain", type="eng_float", default=None,
                      help="set gain in dB (default is midpoint)")
    parser.add_option( "-s","--output-shorts", action="store_true", default=False,
                      help="output interleaved shorts instead of complex floats")
```

```
    parser.add_option("-N", "--nsamples", type="eng_float", default=None,
                      help="number of samples to collect [default=+inf]")
    parser.add_option("-v", "--verbose", action="store_true", default=False,
                      help="verbose output")
    parser.add_option("", "--lo-offset", type="eng_float", default=None,
                      help="set daughterboard LO offset to OFFSET [default=hw default]")
    parser.add_option("", "--wire-format", type="string", default="sc16",
                      help="set wire format from USRP [default=%default")
    parser.add_option("", "--stream-args", type="string", default="",
                      help="set stream arguments [default=%default]")
    parser.add_option("", "--show-async-msg", action="store_true", default=False,
                      help="Show asynchronous message notifications from UHD [default=%d

    (options, args) = parser.parse_args ()
    if len(args) != 1:
        parser.print_help()
        raise SystemExit, 1

    if options.freq is None:
        parser.print_help()
        sys.stderr.write('You must specify the frequency with -f FREQ\n');
        raise SystemExit, 1

    return (options, args[0])


if __name__ == '__main__':
    (options, filename) = get_options()
    tb = rx_cfile_block(options, filename)

    try:
        tb.run()
    except KeyboardInterrupt:
        pass
```

# D.2   Other Code

Due to various interdependencies amongst many separate Python programs and MAT-LAB functions, space limitations have precluded their addition within this document itself. The remainder of the software code used for implementing HSMM chains with USRPs and

conducting the Bayesian HSMM inference algorithm is made publicly available at the following website:

http://www.ece.lsu.edu/swei/Documents/publications/thesis/CarrollThesis.html

## D.2.1 HSMM Generation Using USRPs

The following outline shows the folder hierarchy containing the Python programs for implementing the USRP transmission environments used within this thesis. A short functional description accompanies each file or folder name as well.

- **/USRP_Semi-Markov_Implementation/**

    - **/OFDM_uhd/**
        * ofdm_hsmm_tx_pktsize.py: implements an OFDM-transmitting USRP according to a two-state ON-OFF semi-Markov chain (can be used to generate interference to the coordinated scenarios)
        * two_usrp_mixed_hsmm_tx_pkts.py: implements a 3-state semi-Markov chain in which one OFDM transmitter and one non-OFDM transmitter coordinate their activity to prevent simultaneous transmissions
        * two_usrp_ofdm_hsmm_tx_pkts.py: implements a 3-state semi-Markov chain in which two OFDM transmitters coordinate their activity to prevent simultaneous transmissions
        * ofdm_transmit_path.py & ofdm_transmit_path2.py: defines the transmission path needed to create one OFDM transmitter as needed to implement the semi-Markov chains
        * uhd_ofdm_interface.py & uhd_ofdm_interface2.py: creates the Python flow graph object needed to instantiate the OFDM transmitters
        * regular_transmit_path.py: defines the transmission path needed to create one non-OFDM transmitter as needed to implement the semi-Markov chains
        * uhd_reg_interface.py: creates the Python flow graph object needed to instantiate the non-OFDM transmitter

    - **/regular_uhd/**
        * hsmm_tx_pktsize: implements an OFDM transmitting USRP according to a two-state ON-OFF semi-Markov chain (can be used to generate interference to the coordinated scenarios)
        * transmit_path.py: defines the transmission path needed to create one non-OFDM transmitter as needed to implement the semi-Markov chain
        * uhd_interface.py: creates the Python flow graph object needed to instantiate the non-OFDM transmitter

## D.2.2   Bayesian HDP-HSMM Algorithm

The following outline shows the folder hierarchy containing the MATLAB files used to perform the Bayesian HDP-HSMM algorithm as used in the thesis. Many of the main class definition files used for implementing the algorithm can be found in their original form from Matthew Johnson's website as denoted below[1]:

https://github.com/mattjj/matlab-hsmm

- **/commented-hsmm-algorithm/**
    - **/+durations/**
        * poisson.m:  class definition to draw from, resample, & calculate likelihood values for Poisson distributions
    - **/+observations/**
        * gaussian.m:  class definition to draw from, resample, & calculate likelihood values for multivariate Gaussian distributions
    - **/other_m_files/**
        * hsmm_plot.m: plot the HSMM labeled state sequence along with the magnitude of each complex sample within an input USRP trace
        * identify_idle_state.m: identify the idle state
        * new_state_labeling.m: relabel the HSMM state sequence after certain states are marked for removal
        * remove_states: remove short transient states from the HSMM state sequence based on an lower bound on the Poisson mean parameter
        * reshape_trans_matrix.m & reshape_trans_matrix2.m:  reshape the transition matrix to reflect only those states not marked for deletion
        * runlength.m: used to recompute HSMM state durations in the learned state sequence after doing any state sequence modification
        * stationary_dist.m: calculate the stationary distribution for an input transition matrix
        * transition_count.m: count the number of one-step state transitions from an input HSMM state sequence
        * trunc_resample.m:  used to calculate likelihood functions in the short state removal process
    - **/print/**
        * initial_draw_printout.m: print out results of the first draw from the initial prior distributions to an output text file

---

[1]Note that Matthew Johnson, the coauthor of [13] and [14], has released these files into the public domain on the corresponding website.

* posterior_header_printout.m: print out preliminary information about the algorithm's initial parameters (1st stage) to an output text file
* reduced_header_printout.m: print out preliminary information about the algorithm's initial parameters (2nd stage) to an output text file
* reduced_states_printout.m: print out the learned parameters for the states occurring in the labeled HSMM state sequence after the second round of iterations (2nd stage, after short state removal)
* reduced_states_printout2.m: print out the learned parameters for the states occurring in the labeled HSMM state sequence after the second round of iterations and short-state removal
* states_printout.m: print out the learned parameters for the states occurring in the labeled HSMM state sequence

– **/+util/**

* logsumexp.m: used as an aid in speeding up likelihood calculations
* plot_gaussian_2D.m: plots how the scatter of the data points in relation to the learned covariances in each state
* print_dot.m: prints a dot to MATLAB command window at the conclusion of a single iteration
* sample_discrete.m: used to sample from discrete distribution

– hsmm_first_stage_inference.m: script file used to perform the initial stage of iterations for the HDP-HSMM algorithm

– reduced_model_inference.m: script file used to perform the second stage of iterations for the HDP-HSMM algorithm

– hsmm.m: class definition used to create an hsmm object that serves as the main structure that holds all the relevant information concerning the algorithm's results

– states.m: class definition used to perform state labeling and backward message passing

– transitions.m: class definition used to generate and resample the state transition matrices

– initial_state.m: class definition used to generate and resample the initial state distributions

– read_complex_binary.m: imports binary file with USRP trace measurements into a MATLAB compatible format

Figure D.1 provides a graphic description of sequential order in which significant steps of the HSMM algorithm are performed, along with the relevant MATLAB functions, class definition, and methods necessary in each step. The items aside the initialization step state what variables or parameters need to be specified to run the algorithm as needed in the hsmm_first_stage_inference.m and reduced_model_inference.m script files.
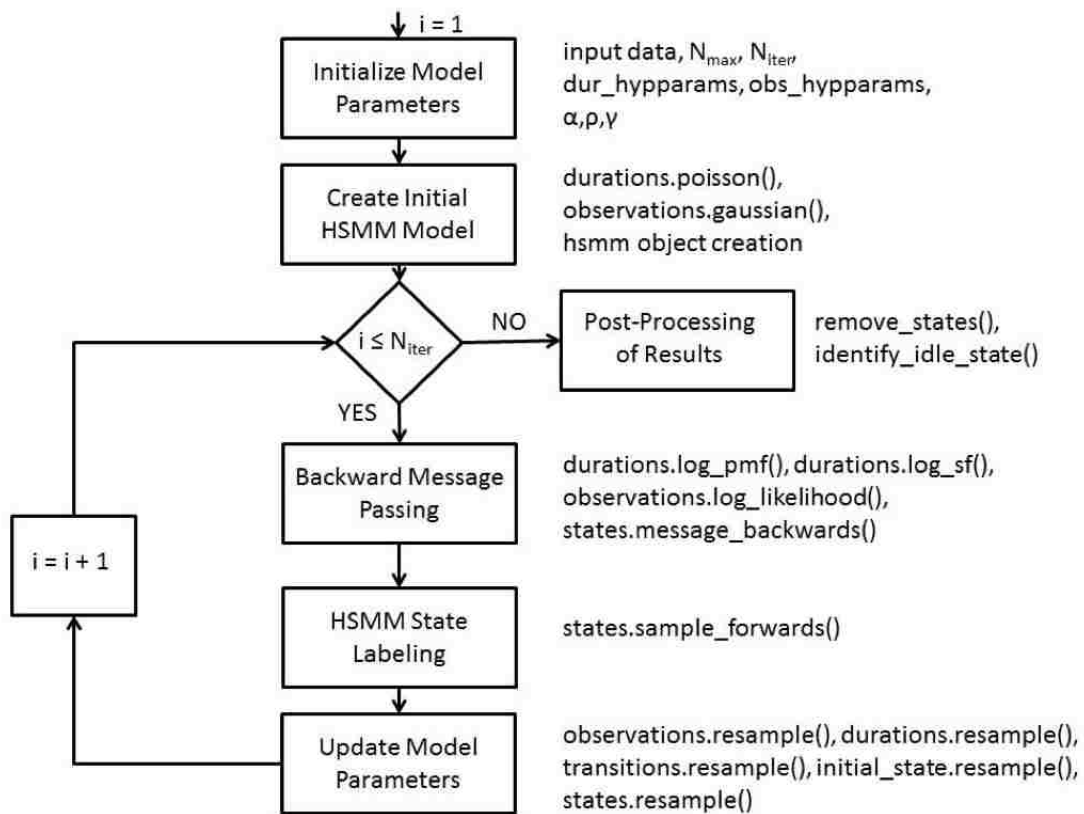
Figure D.1: HSMM Algorithm Flow Diagram

# Appendix E

# Detailed Experimental Results from Chapters 4 and 5

This appendix provides a more in-depth summary of the learned HSMM parameters corresponding to the experiments within Chapters 4 and 5.

## E.1 Chapter 4 Detailed Results

### E.1.1 Two Coordinated OFDM Users

**Initial Hyperparameters**

- $N_{max} = 8$

- State 1: $\lambda \sim \text{Gamma}(k = 10, \theta = 5)$; $\overrightarrow{\mu}, \boldsymbol{\Sigma} \sim \text{NIW}\left(\overrightarrow{\mu_0} = \overrightarrow{0}, \boldsymbol{\Lambda} = \mathbf{I}, \kappa = 5, \nu = 3\right)$

- States 2-8: $\lambda_i \sim \text{Gamma}(k = 20, \theta = 5)$; $\overrightarrow{\mu_i}, \boldsymbol{\Sigma}_i \sim \text{NIW}\left(\overrightarrow{\mu_0} = \overrightarrow{0}, \boldsymbol{\Lambda} = \mathbf{I}, \kappa = 8, \nu = 3\right)$

- $\alpha = \gamma = \rho = 3$

- $N_1 = 50$ iterations

- $N_2 = 50$ iterations

**Resulting Model States**
State 1 (Idle State)
Poisson Rate $\lambda_1 = 98.059738$
Multivariate Gaussian Mean Vector $\overrightarrow{\mu_1} = [-0.000645, 0.000551]$
Multivariate Gaussian Covariance Matrix $\boldsymbol{\Sigma}_1$:

$$\mathbf{\Sigma}_1 = \begin{bmatrix} 0.000172 & -0.000005 \\ -0.000005 & 0.000167 \end{bmatrix}$$

State 2
Poisson Rate $\lambda_2 = 125.038564$
Multivariate Gaussian Mean Vector $\overrightarrow{\mu_2} = [0.000074, 0.000925]$
Multivariate Gaussian Covariance Matrix $\mathbf{\Sigma}_2$:

$$\mathbf{\Sigma}_2 = \begin{bmatrix} 0.005184 & 0.000054 \\ 0.000054 & 0.003902 \end{bmatrix}$$

State 3
Poisson Rate $\lambda_3 = 141.050548$
Multivariate Gaussian Mean Vector $\overrightarrow{\mu_3} = [-0.008029, 0.003744]$
Multivariate Gaussian Covariance Matrix $\mathbf{\Sigma}_3$:

$$\mathbf{\Sigma}_3 = \begin{bmatrix} 0.052985 & -0.000937 \\ -0.000937 & 0.041330 \end{bmatrix}$$

State 5
Poisson Rate $\lambda_5 = 86.090321$
Multivariate Gaussian Mean Vector $\overrightarrow{\mu_5} = [-0.001876, 0.003308]$
Multivariate Gaussian Covariance Matrix $\mathbf{\Sigma}_5$:

$$\mathbf{\Sigma}_5 = \begin{bmatrix} 0.013854 & 0.000074 \\ 0.000074 & 0.009522 \end{bmatrix}$$

Initial State Distribution $\boldsymbol{\pi} = [0.4879, 0.0303, 0.2888, 0.1930]$

State Transition Matrix $\mathbf{A}$:

$$\mathbf{A} = \begin{bmatrix} & 1 & 2 & 3 & 5 \\ 1 & 0.000000 & 0.610113 & 0.378314 & 0.011573 \\ 2 & 0.901059 & 0.000000 & 0.095198 & 0.003742 \\ 3 & 0.998246 & 0.000217 & 0.000000 & 0.001537 \\ 5 & 0.997800 & 0.001911 & 0.000289 & 0.000000 \end{bmatrix}$$

## E.1.2 Two Coordinated OFDM Users with OFDM Interference

**Initial Hyperparameters**

- $N_{max} = 6$

- State 1: $\lambda \sim \text{Gamma}(k = 10, \theta = 5)$; $\overrightarrow{\mu}, \mathbf{\Sigma} \sim \text{NIW}\left(\overrightarrow{\mu_0} = \overrightarrow{0}, \mathbf{\Lambda} = \mathbf{I}, \kappa = 5,\right.$ $\left. \nu = 3\right)$

- States 2-6: $\lambda_i \sim \text{Gamma}(k = 20, \theta = 5)$; $\overrightarrow{\mu_i}, \mathbf{\Sigma}_i \sim \text{NIW}\left(\overrightarrow{\mu_0} = \overrightarrow{0}, \mathbf{\Lambda} = \mathbf{I}, \kappa = 8,\right.$ $\left. \nu = 3\right)$

- $\alpha = \gamma = \rho = 3$

- $N_1 = 50$ iterations

- $N_2 = 50$ iterations

### Resulting Model States

State 1
Poisson Rate $\lambda_1 = 296.183757$
Multivariate Gaussian Mean Vector $\overrightarrow{\mu_1} = [-0.002287, 0.002959]$
Multivariate Gaussian Covariance Matrix $\mathbf{\Sigma}_1$:

$$\mathbf{\Sigma}_1 = \begin{bmatrix} 0.024771 & -0.000058 \\ -0.000058 & 0.018549 \end{bmatrix}$$

State 2 (Idle State)
Poisson Rate $\lambda_2 = 191.789526$
Multivariate Gaussian Mean Vector $\overrightarrow{\mu_2} = [-0.000693, 0.000517]$
Multivariate Gaussian Covariance Matrix $\mathbf{\Sigma}_2$:

$$\mathbf{\Sigma}_2 = \begin{bmatrix} 0.000177 & 0.000000 \\ 0.000000 & 0.000156 \end{bmatrix}$$

State 3
Poisson Rate $\lambda_3 = 115.911212$
Multivariate Gaussian Mean Vector $\overrightarrow{\mu_3} = [0.000638, 0.005621]$
Multivariate Gaussian Covariance Matrix $\mathbf{\Sigma}_3$:

$$\mathbf{\Sigma}_3 = \begin{bmatrix} 0.063751 & 0.001897 \\ 0.001897 & 0.047895 \end{bmatrix}$$

State 4
Poisson Rate $\lambda_4 = 153.371947$
Multivariate Gaussian Mean Vector $\overrightarrow{\mu_4} = [0.005114; -0.001122]$
Multivariate Gaussian Covariance Matrix $\mathbf{\Sigma}_4$:

$$\boldsymbol{\Sigma}_4 = \begin{bmatrix} 0.086058 & -0.002595 \\ -0.002595 & 0.082201 \end{bmatrix}$$

Initial State Distribution $\boldsymbol{\pi} = [0.2803, 0.0038, 0.2660, 0.4498]$

State Transition Matrix $\mathbf{A}$:

$$\mathbf{A} = \begin{bmatrix} & 1 & 2 & 3 & 4 \\ 1 & 0.000000 & 0.377650 & 0.100816 & 0.521534 \\ 2 & 0.036572 & 0.000000 & 0.150152 & 0.813277 \\ 3 & 0.224202 & 0.000028 & 0.000000 & 0.775770 \\ 4 & 0.610809 & 0.043805 & 0.345386 & 0.000000 \end{bmatrix}$$

### E.1.3   Two Coordinated OFDM Users with Low-Power OFDM Interference

**Initial Hyperparameters**

- $N_{max} = 5$

- State 1: $\lambda \sim \text{Gamma}(k = 10, \theta = 5)$; $\overrightarrow{\mu}, \boldsymbol{\Sigma} \sim \text{NIW}\left(\overrightarrow{\mu_0} = \overrightarrow{0}, \boldsymbol{\Lambda} = \mathbf{I}, \kappa = 5, \nu = 3\right)$

- States 2-5: $\lambda_i \sim \text{Gamma}(k = 20, \theta = 5)$; $\overrightarrow{\mu_i}, \boldsymbol{\Sigma}_i \sim \text{NIW}\left(\overrightarrow{\mu_0} = \overrightarrow{0}, \boldsymbol{\Lambda} = \mathbf{I}, \kappa = 8, \nu = 3\right)$

- $\alpha = \gamma = \rho = 3$

- $N_1 = 50$ iterations

- $N_2 = 50$ iterations

**Resulting Model States**

State 1
Poisson Rate $\lambda_1 = 296.183757$
Multivariate Gaussian Mean Vector $\overrightarrow{\mu_1} = [-0.002287, 0.002959]$
Multivariate Gaussian Covariance Matrix $\boldsymbol{\Sigma}_1$:

$$\boldsymbol{\Sigma}_1 = \begin{bmatrix} 0.024771 & -0.000058 \\ -0.000058 & 0.018549 \end{bmatrix}$$

State 2 (Idle State)

Poisson Rate $\lambda_2 = 191.789526$

Multivariate Gaussian Mean Vector $\overrightarrow{\mu_2} = [-0.000693, 0.000517]$

Multivariate Gaussian Covariance Matrix $\boldsymbol{\Sigma}_2$:

$$\boldsymbol{\Sigma}_2 = \begin{bmatrix} 0.000177 & 0.000000 \\ 0.000000 & 0.000156 \end{bmatrix}$$

State 3

Poisson Rate $\lambda_3 = 115.911212$

Multivariate Gaussian Mean Vector $\overrightarrow{\mu_3} = [0.000638, 0.005621]$

Multivariate Gaussian Covariance Matrix $\boldsymbol{\Sigma}_3$:

$$\boldsymbol{\Sigma}_3 = \begin{bmatrix} 0.063751 & 0.001897 \\ 0.001897 & 0.047895 \end{bmatrix}$$

Initial State Distribution $\boldsymbol{\pi} = [0.1158, 0.6661, 0.2181]$

State Transition Matrix $\mathbf{A}$:

$$\mathbf{A} = \begin{bmatrix} & 1 & 2 & 3 \\ 1 & 0 & 0.7734 & 0.2266 \\ 2 & 0.7774 & 0 & 0.2226 \\ 3 & 0.1352 & 0.8648 & 0 \end{bmatrix}$$

## E.1.4  Two Coordinated OFDM Users with QPSK Interference

**Initial Hyperparameters**

- $N_{max} = 8$

- State 1: $\lambda \sim \text{Gamma}(k = 10, \theta = 5)$; $\overrightarrow{\mu}, \boldsymbol{\Sigma} \sim \text{NIW}\left(\overrightarrow{\mu_0} = \overrightarrow{0}, \boldsymbol{\Lambda} = \mathbf{I}, \kappa = 5, \nu = 3\right)$

- States 2-8: $\lambda_i \sim \text{Gamma}(k = 20, \theta = 5)$; $\overrightarrow{\mu_i}, \boldsymbol{\Sigma}_i \sim \text{NIW}\left(\overrightarrow{\mu_0} = \overrightarrow{0}, \boldsymbol{\Lambda} = \mathbf{I}, \kappa = 8, \nu = 3\right)$

- $\alpha = \gamma = \rho = 3$

- $N_1 = 50$ iterations

- $N_2 = 50$ iterations

### Resulting Model States

State 1
Poisson Rate $\lambda_1 = 23.346766$
Multivariate Gaussian Mean Vector $\overrightarrow{\mu_1} = [-0.028191, -0.030406]$
Multivariate Gaussian Covariance Matrix $\boldsymbol{\Sigma}_1$:

$$\boldsymbol{\Sigma}_1 = \begin{bmatrix} 0.014232 & 0.002428 \\ 0.002428 & 0.018758 \end{bmatrix}$$

State 2
Poisson Rate $\lambda_2 = 96.327425$
Multivariate Gaussian Mean Vector $\overrightarrow{\mu_2} = [-0.005928, -0.004105]$
Multivariate Gaussian Covariance Matrix $\boldsymbol{\Sigma}_2$:

$$\boldsymbol{\Sigma}_2 = \begin{bmatrix} 0.066769 & -0.001658 \\ -0.001658 & 0.137198 \end{bmatrix}$$

State 3
Poisson Rate $\lambda_3 = 98.187928$
Multivariate Gaussian Mean Vector $\overrightarrow{\mu_3} = [-0.008969, 0.010114]$
Multivariate Gaussian Covariance Matrix $\boldsymbol{\Sigma}_3$:

$$\boldsymbol{\Sigma}_3 = \begin{bmatrix} 0.037303 & 0.000967 0.000967 & 0.060702 \end{bmatrix}$$

State 5 (Idle State)
Poisson Rate $\lambda_5 = 132.951206$
Multivariate Gaussian Mean Vector $\overrightarrow{\mu_5} = [-0.000152, 0.000604]$
Multivariate Gaussian Covariance Matrix $\boldsymbol{\Sigma}_5$:

$$\boldsymbol{\Sigma}_5 = \begin{bmatrix} 0.000097 & 0.000002 \\ 0.000002 & 0.000103 \end{bmatrix}$$

State 6
Poisson Rate $\lambda_6 = 132.164126$
Multivariate Gaussian Mean Vector $\overrightarrow{\mu_6} = [0.003819, 0.001659]$
Multivariate Gaussian Covariance Matrix $\boldsymbol{\Sigma}_6$:

$$\mathbf{\Sigma}_6 = \begin{bmatrix} 0.004734 & 0.000088 \\ 0.000088 & 0.008005 \end{bmatrix}$$

State 7

Poisson Rate $\lambda_7 = 95.707505$

Multivariate Gaussian Mean Vector $\overrightarrow{\mu_7} = [0.001230, -0.003616]$

Multivariate Gaussian Covariance Matrix $\mathbf{\Sigma}_7$:

$$\mathbf{\Sigma}_7 = \begin{bmatrix} 0.035072 & 0.003015 \\ 0.003015 & 0.070294 \end{bmatrix}$$

Initial State Distribution $\boldsymbol{\pi} = [0.0445, 0.1714, 0.1515, 0.0967, 0.4319, 0.1040]$

State Transition Matrix $\mathbf{A}$:

$$\mathbf{A} = \begin{bmatrix} & 1 & 2 & 3 & 5 & 6 & 7 \\ 1 & 0.000000 & 0.020870 & 0.185350 & 0.593991 & 0.199627 & 0.000162 \\ 2 & 0.159109 & 0.000000 & 0.049473 & 0.452194 & 0.166289 & 0.172935 \\ 3 & 0.000064 & 0.023298 & 0.000000 & 0.165509 & 0.023859 & 0.787270 \\ 4 & 0.010114 & 0.505946 & 0.206098 & 0.000000 & 0.078658 & 0.199184 \\ 5 & 0.011801 & 0.011199 & 0.001997 & 0.973571 & 0.000000 & 0.001432 \\ 6 & 0.138084 & 0.001772 & 0.713293 & 0.124696 & 0.022155 & 0.000000 \end{bmatrix}$$

## E.1.5 Two Coordinated OFDM Users with GMSK Interference

### Initial Hyperparameters

- $N_{max} = 8$

- State 1: $\lambda \sim \text{Gamma}(k = 10, \theta = 5)$; $\overrightarrow{\mu}, \mathbf{\Sigma} \sim \text{NIW}\left(\overrightarrow{\mu_0} = \overrightarrow{0}, \mathbf{\Lambda} = \mathbf{I}, \kappa = 5, \nu = 3\right)$

- States 2-8: $\lambda_i \sim \text{Gamma}(k = 20, \theta = 5)$; $\overrightarrow{\mu_i}, \mathbf{\Sigma}_i \sim \text{NIW}\left(\overrightarrow{\mu_0} = \overrightarrow{0}, \mathbf{\Lambda} = \mathbf{I}, \kappa = 8, \nu = 3\right)$

- $\alpha = \gamma = \rho = 3$

- $N_1 = 50$ iterations

- $N_2 = 50$ iterations

### Resulting Model States

State 1

Poisson Rate $\lambda_1 = 131.650711$
Multivariate Gaussian Mean Vector $\overrightarrow{\mu_1} = [0.017916, 0.003861]$
Multivariate Gaussian Covariance Matrix $\mathbf{\Sigma}_1$:

$$\mathbf{\Sigma}_1 = \begin{bmatrix} 0.282688 & -0.003165 \\ -0.003165 & 0.274904 \end{bmatrix}$$

State 2 (Idle State)
Poisson Rate $\lambda_2 = 378.450516$
Multivariate Gaussian Mean Vector $\overrightarrow{\mu_2} = [0.000154, -0.000019]$
Multivariate Gaussian Covariance Matrix $\mathbf{\Sigma}_2$:

$$\mathbf{\Sigma}_2 = \begin{bmatrix} 0.000158 & 0.000001 \\ 0.000001 & 0.000156 \end{bmatrix}$$

State 3
Poisson Rate $\lambda_3 = 254.233984$
Multivariate Gaussian Mean Vector $\overrightarrow{\mu_3} = [0.000173, 0.000947]$
Multivariate Gaussian Covariance Matrix $\mathbf{\Sigma}_3$:

$$\mathbf{\Sigma}_3 = \begin{bmatrix} 0.006451 & 0.000204 \\ 0.000204 & 0.006147 \end{bmatrix}$$

Initial State Distribution $\boldsymbol{\pi} = [0.2048, 0.6691, 0.1261]$

State Transition Matrix $\mathbf{A}$:

$$\mathbf{A} = \begin{bmatrix} & 1 & 2 & 3 \\ 1 & 0.000000 & 0.552871 & 0.447129 \\ 2 & 0.883738 & 0.000000 & 0.116262 \\ 3 & 0.734436 & 0.265564 & 0.000000 \end{bmatrix}$$

## E.1.6 Two Coordinated OFDM Users with Low-Power GMSK Interference

**Initial Hyperparameters**

- $N_{max} = 6$

- State 1: $\lambda \sim \text{Gamma}(k = 10, \theta = 5)$; $\overrightarrow{\mu}, \mathbf{\Sigma} \sim \text{NIW}\left(\overrightarrow{\mu_0} = \overrightarrow{0}, \mathbf{\Lambda} = \mathbf{I}, \kappa = 5, \nu = 3\right)$

- States 2-6: $\lambda_i \sim \text{Gamma}(k = 20, \theta = 5)$; $\overrightarrow{\mu_i}, \mathbf{\Sigma}_i \sim \text{NIW}(\overrightarrow{\mu_0} = \overrightarrow{0}, \mathbf{\Lambda} = \mathbf{I}, \kappa = 8, \nu = 3\Big)$

- $\alpha = \gamma = \rho = 3$

- $N_1 = 50$ iterations

- $N_2 = 50$ iterations

**Resulting Model States**

State 1
Poisson Rate $\lambda_1 = 65.255393$
Multivariate Gaussian Mean Vector $\overrightarrow{\mu_1} = [0.024193, -0.032945]$
Multivariate Gaussian Covariance Matrix $\mathbf{\Sigma}_1$:

$$\mathbf{\Sigma}_1 = \begin{bmatrix} 0.166973 & 0.015821 \\ 0.015821 & 0.129211 \end{bmatrix}$$

State 2
Poisson Rate $\lambda_2 = 130.422329$
Multivariate Gaussian Mean Vector $\overrightarrow{\mu_2} = [-0.007762, 0.002981]$
Multivariate Gaussian Covariance Matrix $\mathbf{\Sigma}_2$:

$$\mathbf{\Sigma}_2 = \begin{bmatrix} 0.034896 & 0.000554 \\ 0.000554 & 0.022773 \end{bmatrix}$$

State 3 (Idle State)
Poisson Rate $\lambda_3 = 149.927048$
Multivariate Gaussian Mean Vector $\overrightarrow{\mu_3} = [0.000022, -0.000235]$
Multivariate Gaussian Covariance Matrix $\mathbf{\Sigma}_3$:

$$\mathbf{\Sigma}_3 = \begin{bmatrix} 0.000459 & 0.000006 \\ 0.000006 & 0.000346 \end{bmatrix}$$

State 4
Poisson Rate $\lambda_4 = 41.915780$
Multivariate Gaussian Mean Vector $\overrightarrow{\mu_4} = [-0.007581, -0.011208]$
Multivariate Gaussian Covariance Matrix $\mathbf{\Sigma}_4$:

$$\Sigma_4 = \begin{bmatrix} 0.035247 & -0.001818 \\ -0.001818 & 0.022169 \end{bmatrix}$$

State 5
Poisson Rate $\lambda_5 = 107.176962$
Multivariate Gaussian Mean Vector $\overrightarrow{\mu_5} = [0.024286, 0.008171]$
Multivariate Gaussian Covariance Matrix $\Sigma_5$:

$$\Sigma_5 = \begin{bmatrix} 0.166768 & -0.002750 \\ -0.002750 & 0.118607 \end{bmatrix}$$

Initial State Distribution $\boldsymbol{\pi} = [0.4626, 0.0546, 0.2813, 0.0859, 0.1155]$

State Transition Matrix $\mathbf{A}$:

$$\mathbf{A} = \begin{bmatrix}
 & 1 & 2 & 3 & 4 & 5 \\
1 & 0.000000 & 0.007514 & 0.147488 & 0.778349 & 0.066649 \\
2 & 0.000001 & 0.000000 & 0.262107 & 0.698158 & 0.039734 \\
3 & 0.001195 & 0.000002 & 0.000000 & 0.201300 & 0.797503 \\
4 & 0.004149 & 0.577418 & 0.142121 & 0.000000 & 0.276312 \\
5 & 0.506790 & 0.064346 & 0.268037 & 0.160827 & 0.000000
\end{bmatrix}$$

## E.1.7 Two Coordinated OFDM Users with GMSK Interference (2nd Case)

**Initial Hyperparameters**

- $N_{max} = 8$

- State 1: $\lambda \sim \text{Gamma}(k = 10, \theta = 5)$; $\overrightarrow{\mu}, \Sigma \sim \text{NIW}\left(\overrightarrow{\mu_0} = \overrightarrow{0}, \Lambda = \mathbf{I}, \kappa = 5, \nu = 3\right)$

- States 2-8: $\lambda_i \sim \text{Gamma}(k = 20, \theta = 5)$; $\overrightarrow{\mu_i}, \Sigma_i \sim \text{NIW}\left(\overrightarrow{\mu_0} = \overrightarrow{0}, \Lambda = \mathbf{I}, \kappa = 8, \nu = 3\right)$

- $\alpha = \gamma = \rho = 3$

- $N_1 = 50$ iterations

- $N_2 = 50$ iterations

**Resulting Model States**

State 2

Poisson Rate $\lambda_2 = 138.4176$

Multivariate Gaussian Mean Vector $\vec{\mu_2} = [-0.0082, 0.0054]$

Multivariate Gaussian Covariance Matrix $\mathbf{\Sigma}_2$:

$$\mathbf{\Sigma}_2 = \begin{bmatrix} 0.0739 & -0.0020 \\ -0.0020 & 0.0710 \end{bmatrix}$$

State 3

Poisson Rate $\lambda_3 = 162.2581$

Multivariate Gaussian Mean Vector $\vec{\mu_3} = [-2.4264 \times 10^{-4}, 2.4709 \times 10^{-4}]$

Multivariate Gaussian Covariance Matrix $\mathbf{\Sigma}_3$:

$$\mathbf{\Sigma}_3 = \begin{bmatrix} 1.1029 \times 10^{-4} & 1.0869 \times 10^{-7} \\ 1.0869 \times 10^{-7} & 1.0946 \times 10^{-4} \end{bmatrix}$$

State 4

Poisson Rate $\lambda_4 = 129.6950$

Multivariate Gaussian Mean Vector $\vec{\mu_4} = [-0.0013, 0.0037]$

Multivariate Gaussian Covariance Matrix $\mathbf{\Sigma}_4$:

$$\mathbf{\Sigma}_4 = \begin{bmatrix} 0.0236 & -0.0011 \\ -0.0011 & 0.0230 \end{bmatrix}$$

State 5

Poisson Rate $\lambda_5 = 127.7830$

Multivariate Gaussian Mean Vector $\vec{\mu_5} = [-0.0014, 2.7778 \times 10^{-4}]$

Multivariate Gaussian Covariance Matrix $\mathbf{\Sigma}_5$:

$$\mathbf{\Sigma}_5 = \begin{bmatrix} 0.0027 & -2.9807 \times 10^{-5} \\ -2.9807 \times 10^{-5} & 0.0026 \end{bmatrix}$$

Initial State Distribution $\boldsymbol{\pi} = [0.0237, 0.7027, 0.1483, 0.1253]$

State Transition Matrix $\mathbf{A}$:

$$\mathbf{A} = \begin{bmatrix} & 2 & 3 & 4 & 5 \\ 2 & 0.000000 & 0.578705 & 0.002443 & 0.418852 \\ 3 & 0.667084 & 0.000000 & 0.000011 & 0.332905 \\ 4 & 0.531989 & 0.255759 & 0.000000 & 0.212252 \\ 5 & 0.627741 & 0.361974 & 0.010285 & 0.000000 \end{bmatrix}$$

## E.2   Chapter 5 Detailed Results

### E.2.1   WLAN Experiment 1

**Initial Hyperparameters**

- $N_{max} = 3$

- State 1: $\lambda \sim \text{Gamma}(k = 150, \theta = 2)$; $\overrightarrow{\mu}, \mathbf{\Sigma} \sim \text{NIW}\left(\overrightarrow{\mu_0} = \overrightarrow{0}, \mathbf{\Lambda} = \mathbf{I}, \kappa = 300, \nu = 102\right)$

- State 2: $\lambda \sim \text{Gamma}(k = 25, \theta = 2)$; $\overrightarrow{\mu}, \mathbf{\Sigma} \sim \text{NIW}\left(\overrightarrow{\mu_0} = \overrightarrow{0}, \mathbf{\Lambda} = \mathbf{I}, \kappa = 3000, \nu = 302\right)$

- State 3: $\lambda \sim \text{Gamma}(k = 30, \theta = 7)$; $\overrightarrow{\mu}, \mathbf{\Sigma} \sim \text{NIW}\left(\overrightarrow{\mu_0} = \overrightarrow{0}, \mathbf{\Lambda} = \mathbf{I}, \kappa = 3000, \nu = 302\right)$

- $\alpha = \gamma = \rho = 3$

- $N_{iter} = 200$ iterations

**Resulting Model States**
State 1
Poisson Rate $\lambda_1 = 108.7185$
Multivariate Gaussian Mean Vector $\overrightarrow{\mu_1} = [0.0013, -0.0048]$
Multivariate Gaussian Covariance Matrix $\mathbf{\Sigma}_1$:

$$\mathbf{\Sigma}_1 = \begin{bmatrix} 0.0285 & -8.2705 \times 10^{-4} \\ -8.2705 \times 10^{-4} & 0.0413 \end{bmatrix}$$

State 2
Poisson Rate $\lambda_2 = 1.4065$

Multivariate Gaussian Mean Vector $\overrightarrow{\mu_2} = [-8.6052 \times 10^{-4}, 0.0042]$
Multivariate Gaussian Covariance Matrix $\mathbf{\Sigma}_2$:

$$\mathbf{\Sigma}_2 = \begin{bmatrix} 0.0477 & 0.0020 \\ 0.0020 & 0.0841 \end{bmatrix}$$

State 3 (Idle State)
Poisson Rate $\lambda_3 = 146.1474$
Multivariate Gaussian Mean Vector $\overrightarrow{\mu_3} = [-6.4365 \times 10^{-5}, 2.5295 \times 10^{-5}]$
Multivariate Gaussian Covariance Matrix $\mathbf{\Sigma}_3$:

$$\mathbf{\Sigma}_3 = \begin{bmatrix} 3.4359 \times 10^{-5} & -2.4728 \times 10^{-7} \\ -2.4728 \times 10^{-7} & 3.4509 \times 10^{-5} \end{bmatrix}$$

Initial State Distribution $\boldsymbol{\pi} = [0.4357; 0.3754; 0.1888]$

State Transition Matrix $\mathbf{A}$:

$$\mathbf{A} = \begin{bmatrix} & 1 & 2 & 3 \\ 1 & 0.0000 & 0.9995 & 0.0005 \\ 2 & 0.1112 & 0.0000 & 0.8888 \\ 3 & 0.0000 & 1.0000 & 0.0000 \end{bmatrix}$$

## E.2.2  WLAN Experiment 2

### Initial Hyperparameters

- $N_{max} = 3$

- State 1: $\lambda \sim \text{Gamma}(k = 150, \theta = 2)$; $\overrightarrow{\mu}, \mathbf{\Sigma} \sim \text{NIW}\left(\overrightarrow{\mu_0} = \overrightarrow{0}, \mathbf{\Lambda} = \mathbf{I}, \kappa = 300, \nu = 102\right)$

- State 2: $\lambda \sim \text{Gamma}(k = 25, \theta = 2)$; $\overrightarrow{\mu}, \mathbf{\Sigma} \sim \text{NIW}\left(\overrightarrow{\mu_0} = \overrightarrow{0}, \mathbf{\Lambda} = \mathbf{I}, \kappa = 3000, \nu = 302\right)$

- State 3: $\lambda \sim \text{Gamma}(k = 30, \theta = 7)$; $\overrightarrow{\mu}, \mathbf{\Sigma} \sim \text{NIW}\left(\overrightarrow{\mu_0} = \overrightarrow{0}, \mathbf{\Lambda} = \mathbf{I}, \kappa = 3000, \nu = 302\right)$

- $\alpha = \gamma = \rho = 3$

- $N_{iter} = 200$ iterations

**Resulting Model States**

State 1
Poisson Rate $\lambda_1 = 108.7185$
Multivariate Gaussian Mean Vector $\overrightarrow{\mu_1} = [-0.0051, 0.0029]$
Multivariate Gaussian Covariance Matrix $\mathbf{\Sigma}_1$:

$$\mathbf{\Sigma}_1 = \begin{bmatrix} 0.0281 & -9.5957 \times 10^{-4} \\ -9.5957 \times 10^{-4} & 0.0414 \end{bmatrix}$$

State 2
Poisson Rate $\lambda_2 = 1.2097$
Multivariate Gaussian Mean Vector $\overrightarrow{\mu_2} = [0.0045, -0.0013]$
Multivariate Gaussian Covariance Matrix $\mathbf{\Sigma}_2$:

$$\mathbf{\Sigma}_2 = \begin{bmatrix} 0.0553 & 0.0030 \\ 0.0038 & 0.0884 \end{bmatrix}$$

State 3 (Idle State)
Poisson Rate $\lambda_3 = 146.1633$
Multivariate Gaussian Mean Vector $\overrightarrow{\mu_3} = [-2.1540 \times 10^{-6}, 7.6220 \times 10^{-5}]$
Multivariate Gaussian Covariance Matrix $\mathbf{\Sigma}_3$:

$$\mathbf{\Sigma}_3 = \begin{bmatrix} 3.4284 \times 10^{-5} & 1.2194 \times 10^{-7} \\ 1.2194 \times 10^{-7} & 3.4908 \times 10^{-5} \end{bmatrix}$$

Initial State Distribution $\boldsymbol{\pi} = [0.0691; 0.0871; 0.8438]$

State Transition Matrix $\mathbf{A}$:

$$\mathbf{A} = \begin{bmatrix} & 1 & 2 & 3 \\ 1 & 0.0000 & 0.8778 & 0.1262 \\ 2 & 0.1314 & 0.0000 & 0.8686 \\ 3 & 0.0280 & 0.9720 & 0.0000 \end{bmatrix}$$

## E.2.3  WLAN Experiment 3

**Initial Hyperparameters**

- $N_{max} = 4$

- State 1: $\lambda \sim \text{Gamma}(k = 150, \theta = 2)$; $\overrightarrow{\mu}, \boldsymbol{\Sigma} \sim \text{NIW}\left(\overrightarrow{\mu_0} = \overrightarrow{0}, \boldsymbol{\Lambda} = \mathbf{I}, \kappa = 300,\right.$
  $\left. \nu = 102\right)$

- State 2: $\lambda \sim \text{Gamma}(k = 25, \theta = 2)$; $\overrightarrow{\mu}, \boldsymbol{\Sigma} \sim \text{NIW}\left(\overrightarrow{\mu_0} = \overrightarrow{0}, \boldsymbol{\Lambda} = \mathbf{I}, \kappa = 3000,\right.$
  $\left. \nu = 302\right)$

- State 3: $\lambda \sim \text{Gamma}(k = 30, \theta = 7)$; $\overrightarrow{\mu}, \boldsymbol{\Sigma} \sim \text{NIW}\left(\overrightarrow{\mu_0} = \overrightarrow{0}, \boldsymbol{\Lambda} = \mathbf{I}, \kappa = 3000,\right.$
  $\left. \nu = 302\right)$

- State 4: $\lambda \sim \text{Gamma}(k = 100, \theta = 2)$; $\overrightarrow{\mu}, \boldsymbol{\Sigma} \sim \text{NIW}\left(\overrightarrow{\mu_0} = \overrightarrow{0}, \boldsymbol{\Lambda} = \mathbf{I}, \kappa = 300,\right.$
  $\left. \nu = 102\right)$

- $\alpha = \gamma = \rho = 3$

- $N_{iter} = 250$ iterations

**Resulting Model States**

State 1
Poisson Rate $\lambda_1 = 203.9010$
Multivariate Gaussian Mean Vector $\overrightarrow{\mu_1} = [-0.0060, 0.0053]$
Multivariate Gaussian Covariance Matrix $\boldsymbol{\Sigma}_1$:

$$\boldsymbol{\Sigma}_1 = \begin{bmatrix} 0.0311 & -5.3603 \times 10^{-4} \\ -5.3603 \times 10^{-4} & 0.0446 \end{bmatrix}$$

State 2 (Idle State)
Poisson Rate $\lambda_2 = 87.4118$
Multivariate Gaussian Mean Vector $\overrightarrow{\mu_2} = [-5.6561 \times 10^{-5}, 5.6610 \times 10^{-5}]$
Multivariate Gaussian Covariance Matrix $\boldsymbol{\Sigma}_2$:

$$\boldsymbol{\Sigma}_2 = \begin{bmatrix} 3.4921 \times 10^{-5} & -1.8496 \times 10^{-8} \\ -1.8496 \times 10^{-8} & 3.5102 \times 10^{-5} \end{bmatrix}$$

State 4
Poisson Rate $\lambda_4 = 146.9792$
Multivariate Gaussian Mean Vector $\overrightarrow{\mu_4} = [8.1794 \times 10^{-4}, -0.0082]$
Multivariate Gaussian Covariance Matrix $\boldsymbol{\Sigma}_4$:

$$\mathbf{\Sigma}_4 = \begin{bmatrix} 0.0500 & 7.6600 \times 10^{-4} \\ 7.6600 \times 10^{-4} & 0.0792 \end{bmatrix}$$

Initial State Distribution $\boldsymbol{\pi} = [0.0918; 0.7296; 0.1786]$

State Transition Matrix $\mathbf{A}$:

$$\mathbf{A} = \begin{bmatrix} & 1 & 2 & 4 \\ 1 & 0.0000 & 0.1620 & 0.8380 \\ 2 & 0.0176 & 0.0000 & 0.9830 \\ 3 & 0.0213 & 0.9871 & 0.0000 \end{bmatrix}$$

## E.2.4 WLAN Experiment 4

**Initial Hyperparameters**

- $N_{max} = 5$

- State 1: $\lambda \sim \mathrm{Gamma}(k = 150, \theta = 2)$; $\overrightarrow{\mu}, \mathbf{\Sigma} \sim \mathrm{NIW}\Big(\overrightarrow{\mu_0} = \overrightarrow{0}, \mathbf{\Lambda} = \mathbf{I}, \kappa = 300, \nu = 102\Big)$

- State 2: $\lambda \sim \mathrm{Gamma}(k = 25, \theta = 2)$; $\overrightarrow{\mu}, \mathbf{\Sigma} \sim \mathrm{NIW}\Big(\overrightarrow{\mu_0} = \overrightarrow{0}, \mathbf{\Lambda} = \mathbf{I}, \kappa = 3000, \nu = 302\Big)$

- State 3: $\lambda \sim \mathrm{Gamma}(k = 30, \theta = 7)$; $\overrightarrow{\mu}, \mathbf{\Sigma} \sim \mathrm{NIW}\Big(\overrightarrow{\mu_0} = \overrightarrow{0}, \mathbf{\Lambda} = \mathbf{I}, \kappa = 3000, \nu = 302\Big)$

- State 4: $\lambda \sim \mathrm{Gamma}(k = 40, \theta = 8)$; $\overrightarrow{\mu}, \mathbf{\Sigma} \sim \mathrm{NIW}\Big(\overrightarrow{\mu_0} = \overrightarrow{0}, \mathbf{\Lambda} = \mathbf{I}, \kappa = 3000, \nu = 302\Big)$

- State 5: $\lambda \sim \mathrm{Gamma}(k = 100, \theta = 2)$; $\overrightarrow{\mu}, \mathbf{\Sigma} \sim \mathrm{NIW}\Big(\overrightarrow{\mu_0} = \overrightarrow{0}, \mathbf{\Lambda} = \mathbf{I}, \kappa = 300, \nu = 102\Big)$

- $\alpha = \gamma = \rho = 3$

- $N_{iter} = 250$ iterations

**Resulting Model States**

State 2
Poisson Rate $\lambda_2 = 1.4991$
Multivariate Gaussian Mean Vector $\vec{\mu_2} = [0.0036, 0.0061]$
Multivariate Gaussian Covariance Matrix $\boldsymbol{\Sigma}_2$:

$$\boldsymbol{\Sigma}_2 = \begin{bmatrix} 0.0355 & -0.0023 \\ -0.0023 & 0.0441 \end{bmatrix}$$

State 3 (Idle State)
Poisson Rate $\lambda_3 = 142.4575$
Multivariate Gaussian Mean Vector $\vec{\mu_3} = [-9.6582 \times 10^{-5}, 1.1266 \times 10^{-4}]$
Multivariate Gaussian Covariance Matrix $\boldsymbol{\Sigma}_3$:

$$\boldsymbol{\Sigma}_3 = \begin{bmatrix} 3.4805 \times 10^{-5} & -1.9054 \times 10^{-7} \\ -1.9054 \times 10^{-7} & 3.5381 \times 10^{-5} \end{bmatrix}$$

State 4
Poisson Rate $\lambda_4 = 114.2218$
Multivariate Gaussian Mean Vector $\vec{\mu_4} = [7.3428 \times 10^{-4}, 7.6264 \times 10^{-4}]$
Multivariate Gaussian Covariance Matrix $\boldsymbol{\Sigma}_4$:

$$\boldsymbol{\Sigma}_4 = \begin{bmatrix} 0.0033 & -1.1203 \times 10^{-4} \\ -1.1203 \times 10^{-4} & 0.0030 \end{bmatrix}$$

State 5
Poisson Rate $\lambda_5 = 120.0889$
Multivariate Gaussian Mean Vector $\vec{\mu_5} = [-0.0014, -0.0149]$
Multivariate Gaussian Covariance Matrix $\boldsymbol{\Sigma}_5$:

$$\boldsymbol{\Sigma}_5 = \begin{bmatrix} 0.0275 & -8.5148 \times 10^{-5} \\ -8.5148 \times 10^{-5} & 0.0371 \end{bmatrix}$$

Initial State Distribution $\boldsymbol{\pi} = [0.0476; 0.1110; 0.0531; 0.7883]$

State Transition Matrix $\mathbf{A}$:

$$\mathbf{A} = \begin{bmatrix} & 2 & 3 & 4 & 5 \\ 2 & 0.0000 & 0.9012 & 0.0048 & 0.0939 \\ 3 & 0.9996 & 0.0000 & 0.0002 & 0.0002 \\ 4 & 0.9796 & 0.0204 & 0.0000 & 0.0000 \\ 5 & 0.9898 & 0.0102 & 0.0000 & 0.0000 \end{bmatrix}$$

# Vita

Benjamin Elliott Carroll was born on February 8, 1985, in Opelousas, Louisiana. He graduated with a Bachelors in Science degree in Physics in 2008 from Louisiana State University. He is currently pursuing a Masters of Science in Electrical Engineering at Louisiana State University, and is expected to graduate in May 2013. His research interests include wireless communications, digital signal processing, and software defined radio programming. He has worked as a Graduate Research Assistant in the Division of Electrical and Computer Engineering under Dr. Shuangqing Wei from August 2010 to August 2012.