9-28-2009

Tracking Rhythmicity in Biomedical Signals using Sequential Monte Carlo methods

Sungan Kim
Portland State University

DISSERTATION APPROVAL

The abstract and dissertation of Sunghan Kim for the Doctor of Philosophy in Electrical and Computer Engineering were presented September 28, 2009, and accepted by the dissertation committee and the doctoral program.

COMMITTEE APPROVALS: [redacted]

James McNames

[redacted]

Dan Hammerstrom

[redacted]

Martin Siderius

[redacted]

Lisa Zurk

[redacted]

Gerardo Lafferriere
Representative of the Office of Graduate Studies

DOCTORAL PROGRAM APPROVAL: [redacted]

Malgorzata Chrzanowska-Jeske, Director
Electrical and Computer Engineering
Ph.D. Program

ABSTRACT

An abstract of the dissertation of Sunghan Kim for the Doctor of Philosophy in Electrical and Computer Engineering presented September 28, 2009.

Title: Tracking Rhythmicity in Biomedical Signals Using Sequential Monte Carlo Methods

Cyclical patterns are common in signals that originate from natural systems such as the human body and man-made machinery. Often these cyclical patterns are not perfectly periodic. In that case, the signals are called *pseudo-periodic* or *quasi-periodic* and can be modeled as a sum of time-varying sinusoids, whose frequencies, phases, and amplitudes change slowly over time. Each time-varying sinusoid represents an individual rhythmical component, called a *partial*, that can be characterized by three parameters: frequency, phase, and amplitude. Quasi-periodic signals often contain multiple *partials* that are harmonically related. In that case, the frequencies of other partials become exact integer multiples of that of the slowest partial. These signals are referred to as *multi-harmonic signals*. Examples of such signals are electrocardiogram (ECG), arterial blood pressure (ABP), and human voice.

A *Markov process* is a mathematical model for a random system whose future and past states are independent conditional on the present state. Multi-harmonic

signals can be modeled as a stochastic process with the Markov property. The Markovian representation of multi-harmonic signals enables us to use state-space tracking methods to continuously estimate the frequencies, phases, and amplitudes of the partials.

Several research groups have proposed various signal analysis methods such as hidden Markov Models (HMM), short time Fourier transform (STFT), and Wigner-Ville distribution to solve this problem. Recently, a few groups of researchers have proposed Monte Carlo methods which estimate the posterior distribution of the fundamental frequency in multi-harmonic signals sequentially. However, multi-harmonic tracking is more challenging than single-frequency tracking, though the reason for this has not been well understood. The main objectives of this dissertation are to elucidate the fundamental obstacles to multi-harmonic tracking and to develop a reliable multi-harmonic tracker that can track cyclical patterns in multi-harmonic signals.

TRACKING RHYTHMICITY IN BIOMEDICAL SIGNALS USING SEQUENTIAL MONTE CARLO METHODS

by

SUNGHAN KIM

A dissertation submitted in partial fulfillment of
the requirements for the degree of

DOCTOR OF PHILOSOPHY
in
ELECTRICAL AND COMPUTER ENGINEERING

Portland State University
2009

DEDICATION

This dissertation is dedicated to my lovely wife, Hyunjung (Rosemary) Mok.
I was able to complete it only with her faithful prayer and support.

An excellent wife is the crown of her husband - Proverbs 12:4

ACKNOWLEDGEMENTS

I appreciate positive feedback and encouragement from my dissertation committee members, who are Drs. Dan Hammerstrom, Martin Siderius, Lisa Zurk, Gerardo Lafferriere, and James McNames.

Especially, I am indebted to Dr. McNames who has been my excellent advisor. He has had so much influence on my life, which I will never forget.

Contents

List of Tables

List of Figures

List of Algorithms

Chapter 1

Introduction

1.1 Background

Signals generated by natural systems and man-made machinery exhibit cyclical patterns. Estimation of these cyclical patterns in signals is essential to analyze the characteristics of the systems that generate the signals. Signals are called *periodic* when they satisfy the condition:

$$x(t) = x(t + T) \tag{1.1}$$

where T is a fundamental period of the periodic signal. The periodic signal $x(t)$ repeats itself every T. Therefore, $\{x(t)|\tau < t < \tau + T\}$ is a complete description of the signal $x(t)$ where τ can have any value. The top plot in Fig. 1.1 shows an example of a periodic signal whose period T is 1 s in the time domain. The inverse of the fundamental period T is the fundamental frequency f of the signal.

Any periodic signal can be decomposed into a sum of sines and cosines whose frequencies and amplitudes are constant. This decomposition is called the *Fourier series representation.* The Fourier series for a periodic signal $x(t)$ can be written in three different forms as follows,

$$x(t) = \sum_{k=-\infty}^{\infty} c_k e^{jk\theta(t)} \tag{1.2}$$

$$= \frac{a_0}{2} + \sum_{k=1}^{\infty} a_k \cos\left[k\theta(t) + \phi_k\right] \tag{1.3}$$

$$= \frac{a_0}{2} + \sum_{k=1}^{\infty} a_{1,k} \cos\left[k\theta(t)\right] + a_{2,k} \sin\left[k\theta(t)\right] \tag{1.4}$$

where $\theta(t)$ is equal to $2\pi t/T$. (1.2) is called a complex form, (1.3) is called an angular form, and (1.4) is called a rectangular form. These three forms are interchangeable. For instance, knowing the phases ϕ_k and amplitudes a_k in (1.3) is equivalent to knowing the coefficients $a_{1,k}$ and $a_{2,k}$ in (1.4). In the rectangular form, the Fourier coefficients can be computed as,

$$a_{1,k} = \frac{1}{\pi} \int_t^{t+T} x(t) \cos\left[k\theta(t)\right] \, \mathrm{d}t \tag{1.5}$$

$$a_{2,k} = \frac{1}{\pi} \int_t^{t+T} x(t) \sin\left[k\theta(t)\right] \, \mathrm{d}t \tag{1.6}$$

However, real signals with cyclical patterns rarely meet the condition (1.1) exactly. Rather, most signals meet a condition:

$$x(t) \approx x\left[t + T(t)\right] \tag{1.7}$$

where $\approx$ represents *approximate equality* and $T(t)$ is a time-varying fundamental period. Such signals that are not perfectly periodic yet *approximately periodic* are called *pseudo-periodic* or *quasi-periodic* signals. Quasi-periodic signals can be modeled as a linear combination of time-varying sinusoids. Each time-varying sinusoid represents an individual rhythmical component, *partial*, which can be completely characterized by its frequency, phase, and amplitude. In contrast to Fourier series these parameters are not necessarily constant. The sum-of-sinusoids model can be written as,

$$x(t) = \frac{a_0(t)}{2} + \sum_{k=1}^{\infty} a_{1,k}(t) \cos\left[\theta_k(t)\right] + a_{2,k}(t) \sin\left[\theta_k(t)\right] \tag{1.8}$$

where $a_0(t)$, $a_{1,k}(t)$, $a_{2,k}(t)$, and $T_k(t)$ are time-varying parameters. The instantaneous phase $\theta_k(t)$ can be written as,

$$\begin{aligned} \theta_k(t) &= \int_0^t 2\pi/T_k(\tau)\, \mathrm{d}\tau && (1.9) \\ &= \int_0^t 2\pi f_k(\tau)\, \mathrm{d}\tau && (1.10) \end{aligned}$$

Since observed signals are commonly contaminated by noise, a more complete model can be expressed as follows,

$$\begin{aligned} \boldsymbol{x}(t) &= \boldsymbol{s}(t) + \boldsymbol{n}(t) && (1.11) \\ &= \frac{\boldsymbol{a}_0(t)}{2} + \sum_{k=1}^{\infty} \boldsymbol{a}_{1,k}(t) \cos\left[\boldsymbol{\theta}_k(t)\right] + \boldsymbol{a}_{2,k}(t) \sin\left[\boldsymbol{\theta}_k(t)\right] + \boldsymbol{n}(t) && (1.12) \end{aligned}$$

where $\boldsymbol{n}(t)$ is an additive noise.

The term *periodic* is not appropriate to refer to these cyclical patterns in quasi-periodic signals because the signals are not exactly periodic. Instead, *rhythmicity* refers to the cyclical pattern that can be represented as a linear combination of time-varying sinusoids, i.e. partials. *Rhythmicity tracking*, then, is continuous estimation of the frequencies, phases, and amplitudes of the partials.

Rhythmical signals can be categorized in various ways. One way to do so is to study how the partials in the signals are related. In some cases, there could be only one partial. In other cases, there could be multiple partials that are related to one another through some nonlinearity. A special case arises when a quasi-periodic signal contains multiple partials and they are harmonically related. This type of signal is called a *multi-harmonic signal*, which can be expressed as follows,

$$\begin{aligned}\boldsymbol{x}(t) &= \boldsymbol{s}(t) + \boldsymbol{n}(t) && (1.13)\\ &= \frac{\boldsymbol{a}_0(t)}{2} + \sum_{k=1}^{\infty} \boldsymbol{a}_{1,k}(t)\cos\left[k\boldsymbol{\theta}(t)\right] + \boldsymbol{a}_{2,k}(t)\sin\left[k\boldsymbol{\theta}(t)\right] + \boldsymbol{n}(t) && (1.14)\end{aligned}$$

where $\boldsymbol{\theta}(t)$ can be written as,

$$\begin{aligned}\theta(t) &= \int_0^t 2\pi/\boldsymbol{T}(\tau)\,\mathrm{d}\tau && (1.15)\\ &= \int_0^t 2\pi\boldsymbol{f}(\tau)\,\mathrm{d}\tau && (1.16)\end{aligned}$$

The frequencies of other partials in the multi-harmonic signal are exact integer multiples of the slowest partial's frequency. This slowest partial is called the *fundamental partial* and its frequency called the *fundamental frequency.* The other partials in the signal are referred to as *harmonic partials* and their frequencies called *harmonic frequencies.* A *harmonic set* refers to the fundamental and harmonic partials as a whole and the number of harmonic sets in the signal is not limited to one. When there is more than one harmonic set, the signal is called a *multiple harmonic set signal.* The harmonic sets in the multiple harmonic set signal can be independent of each other or related with each other through nonlinearity such as amplitude or frequency modulation. A two harmonic set signal, whose harmonic sets are independent of each other, can be expressed as,

$$\begin{aligned} \boldsymbol{x}(t) =& \boldsymbol{s}(t) + \boldsymbol{n}(t) && (1.17)\\ =& \frac{\boldsymbol{a}_0(t)}{2} + \sum_{k=1}^{\infty} \boldsymbol{a}_{1,k}(t)\cos\left[k\boldsymbol{\theta}_1(t)\right] + \boldsymbol{a}_{2,k}(t)\sin\left[k\boldsymbol{\theta}_1(t)\right] + \\ & \sum_{k=1}^{\infty} \boldsymbol{a}_{3,k}(t)\cos\left[k\boldsymbol{\theta}_2(t)\right] + \boldsymbol{a}_{4,k}(t)\sin\left[k\boldsymbol{\theta}_2(t)\right] + \boldsymbol{n}(t) && (1.18) \end{aligned}$$

where $\boldsymbol{\theta}_1(t)$ and $\boldsymbol{\theta}_2(t)$ can be written as,

$$\begin{aligned} \boldsymbol{\theta}_1(t) &= \int_0^t 2\pi/\boldsymbol{T}_1(t)\,\mathrm{d}\tau \\ &= \int_0^t 2\pi\boldsymbol{f}_1(t)\,\mathrm{d}\tau && (1.19)\\ \boldsymbol{\theta}_2(t) &= \int_0^t 2\pi/\boldsymbol{T}_2(t)\,\mathrm{d}\tau \\ &= \int_0^t 2\pi\boldsymbol{f}_2(t)\,\mathrm{d}\tau && (1.20) \end{aligned}$$

where $\boldsymbol{f}_1(t)$ and $\boldsymbol{f}_2(t)$ are the fundamental frequencies of the first and second harmonic sets, respectively.

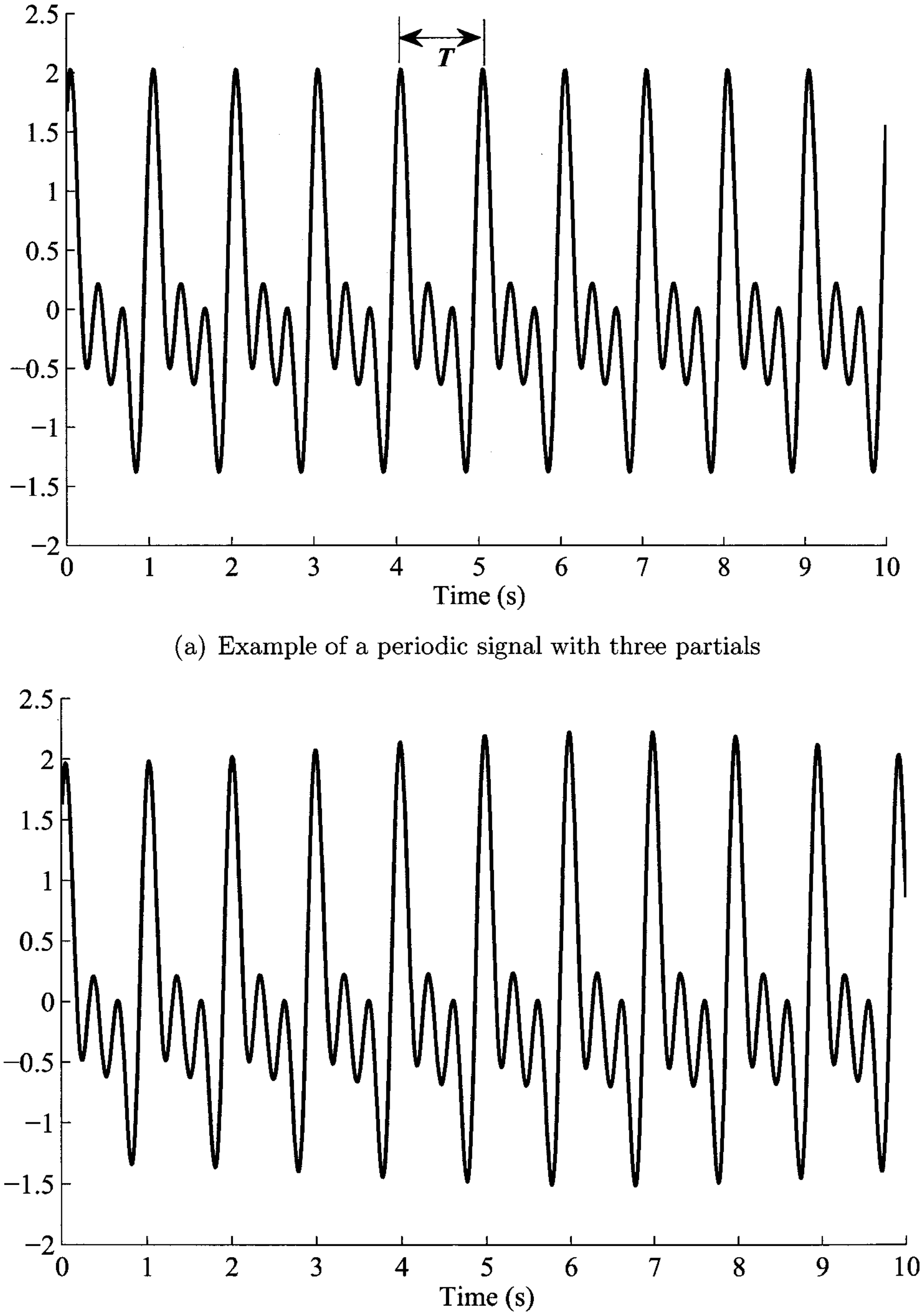
So far, signals have been assumed to be continuous-time. However, these days continuous-time signals are converted to discrete-time signals to be stored in a digital format and signal processing is performed digitally. The conversion process of continuous-time signals into discrete-time signals is called *sampling.* While continuous-time signals are a function of a continuous time argument t, discrete-time signals are a function of discrete integers n. Discrete-time signals are a sequence of quantities or values and each value is called a *sample.* Each sample of discrete-time signals is evenly spaced in the time domain and this time space between samples is called a *sampling period* T_s. The reciprocal of the sampling period is called a *sampling frequency* f_s. Discrete-time signals alone do not provide any information on the sampling frequency, but it can be calculated when both continuous-time and discrete-time signals are available. When the continuous signal $\boldsymbol{x}(t)$ in (1.14) is sampled at f_s, the sampling result $\boldsymbol{x}_n$ can be expressed as,

$$\begin{aligned}\boldsymbol{x}_n &= \boldsymbol{s}_n + \boldsymbol{n}_n && (1.21)\\ &= \frac{\boldsymbol{a}_{0,n}}{2} + \sum_{k=1}^{\infty} \boldsymbol{a}_{1,k,n} \cos\left(k\boldsymbol{\theta}_n\right) + \boldsymbol{a}_{2,k,n} \sin\left(k\boldsymbol{\theta}_n\right) + \boldsymbol{n}_n && (1.22)\end{aligned}$$

where $\boldsymbol{\theta}_n$ can be written as,

$$\begin{aligned}\boldsymbol{\theta}_n &= \sum_{l=0}^{n} \frac{2\pi}{f_\mathrm{s}\boldsymbol{T}_l} && (1.23)\\ &= \sum_{l=0}^{n} \frac{2\pi \boldsymbol{f}_l}{f_\mathrm{s}} && (1.24)\end{aligned}$$

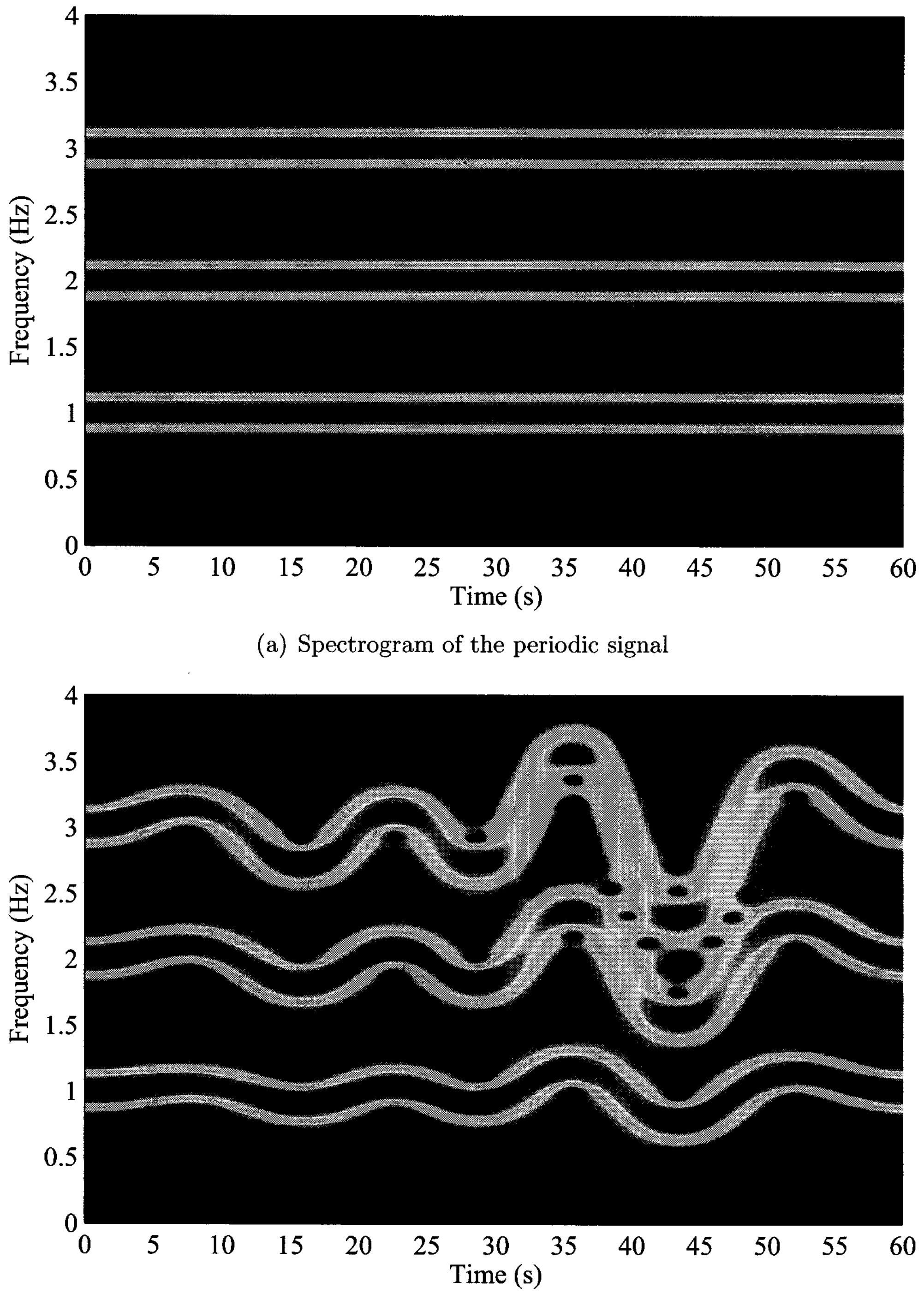
Two plots in Fig. 1.1 compare periodic and quasi-periodic signals in the time domain. The signals are synthetically generated based on the signal models in (1.4) and (1.14) with three harmonic partials. In the top plot, the period T of the signal is 1 s. Therefore, the signal repeats itself exactly every 1 s. In the bottom plot the period T of the signal is approximately 1 s and the magnitude of the signal changes slowly. A spectrogram is an image that shows how the spectral density of a signal changes over time. They are also known as sonograms or voicegrams. Two plots in Fig. 1.2 compare the spectrograms of the periodic and quasi-periodic signals. They illustrate the constant and slowly changing fundamental frequencies of the periodic and quasi-period signals, respectively. The signal shown in Fig. 1.1 is the first 10 s of the entire 60 s signal whose spectrogram is shown in Fig. 1.2.



(a) Example of a periodic signal with three partials

(b) Example of a quasi-periodic signal with three partials

Figure 1.1: Periodic versus quasi-periodic signals in the time domain.



(a) Spectrogram of the periodic signal

(b) Spectrogram of the quasi-periodic signal

Figure 1.2: Spectrograms of the periodic and quasi-periodic signals, whose first 10 s period is shown in Fig. 1.1.

1.2 Significance

1.2.1 Conventional versus Proposed Methods

Many techniques have been proposed for frequency tracking in quasi-periodic signals. One may be tempted to apply one of the frequency tracking techniques to multi-harmonic tracking by having each partial tracked separately and combining the results. It is also possible to estimate the fundamental partial alone first then other harmonic partials since their frequencies are integer multiples of the fundamental frequency. However, it is sub-optimal to extend those frequency tracking techniques to multi-harmonic tracking because frequency tracking techniques are not capable of taking the harmonic relationship between the partials into account. Multi-harmonic tracking methods track the fundamental partial along with its harmonic partials simultaneously considering their harmonic relationship. A major advantage of multi-harmonic tracking over individual frequency tracking is that the fundamental frequency estimate becomes more accurate and has less variation since multi-harmonic tracking techniques take advantage of the variation in the harmonic frequencies.

There are a few techniques being used for the purpose of multi-harmonic tracking. They include time-frequency distributions (TFDs) [1,2], adaptive frequency estimation methods [3], pitch detection algorithm (PDAs) [4,5], and state-space model based algorithms [6,7]. However, some of them (PDAs) are only capable of tracking the fundamental frequency and others (TFDs) do not take advantage of the harmonic relationship between the partials. Many of those techniques rely on the local stationarity assumption of the signals, which means that they estimate the frequencies and amplitudes of the partials on a segment-by-segment basis not a sample-by-sample basis. In contrast, the state-space model based multi-harmonic tracking algorithms can track the frequencies and amplitudes of the partials on a sample-by-sample basis and take into account the harmonic relationship between the partials. A newly proposed multi-harmonic tracking method is a state-space model based technique. It is not only capable of what other state-space model based methods such as the extended Kalman filter based tracker can do, but also superior to them because it properly addresses major problems in multi-harmonic tracking due to the harmonic relationship between the partials. The new method is more robust to noise and provides more complete and accurate estimates.

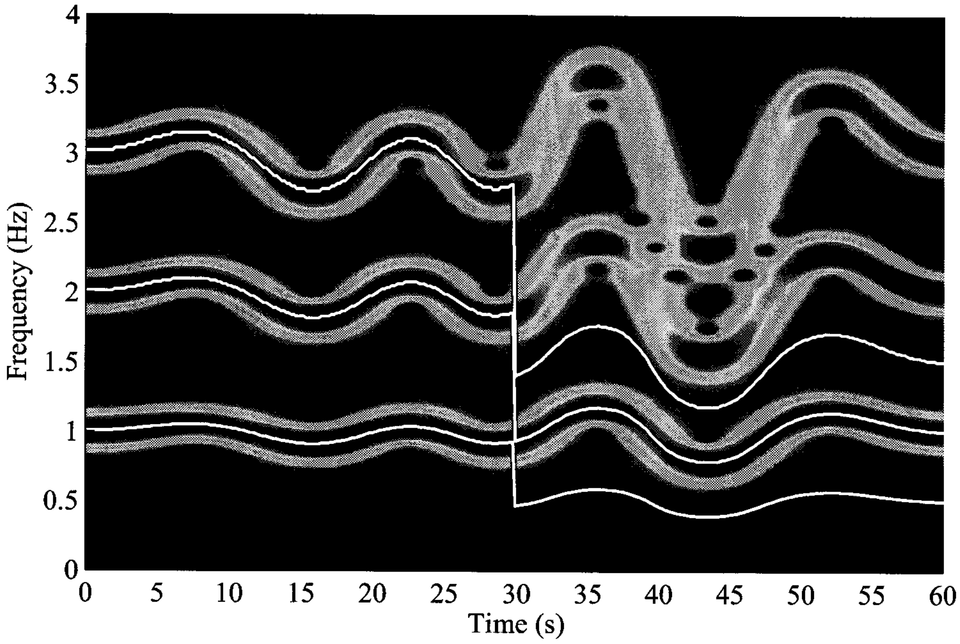
1.2.2 Applications

There are many multi-harmonic applications. One example is heart rate tracking in electrocardiograms (ECGs). This is a good example of a quasi-periodic signal with many harmonic partials. The ECG is a recording of the electrical activity of the heart over time. The fundamental frequency of ECG is the heart rate. Accurate estimation of the heart rate is critical to diagnose abnormal rhythms of the heart caused by conductive tissue damage or abnormal dissolved salt level. The heart rate in ECG signals is typically estimated utilizing time-domain beat detection algorithms where the inverse of beat-to-beat intervals is the estimate of the heart rate. However, the accuracy of beat detection is greatly affected by noise level. Alternatively, the heart rate in ECG signals can be done in the frequency domain. It can be a challenging task because it is not uncommon that the harmonic partials have more power than the fundamental partials. Multi-harmonic tracking methods based on the sum-of-sinusoids model are an appropriate tool for continuous estimation of the heart rate in ECG signals. Two plots in Fig.

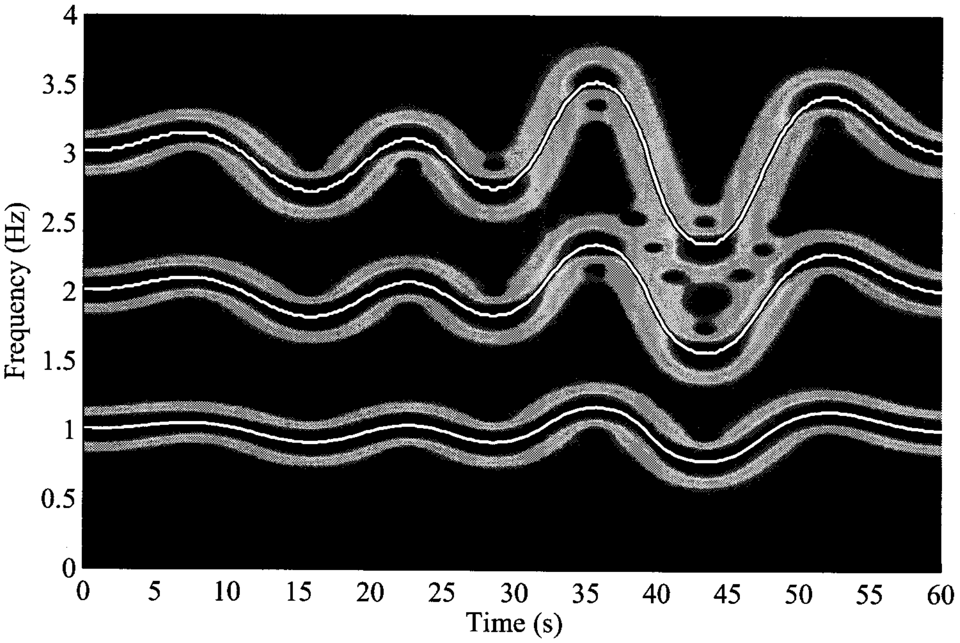
1.3 show a synthetic example of the extended Kalman filter (EKF) based multi-harmonic tracker (top) that fails to track the partials in an electrocardiogram (ECG) signal and the newly proposed multi-harmonic tracker (bottom) that tracks them accurately. In the top plot, the estimated frequency of the second partial matches the true frequency of the fundamental partial. This estimation error is due to the fact that the EKF-based tracker cannot handle the harmonic relationship between the partials properly. In the bottom plot, the newly proposed tracker tracks the true fundamental frequency and its harmonics accurately throughout the entire signal duration.

1.3 Objectives

This dissertation will meet five major objectives. *The first one* is to develop a new sequential Monte Carlo (SMC) algorithm for robust and accurate multi-harmonic tracking. Although the unique challenges in multi-harmonic tracking motivated development of the new SMC algorithm, it is a much more general tool that can be applied to other *a posteriori* distribution estimation problems. *The second* is to develop a single rhythmical component tracking algorithm. Single rhythmical component tracking is a special case of multi-harmonic tracking where the number of partials is only one. The single rhythmical component tracking algorithm is applied to tremor tracking in binary spike trains. *The third* is to develop a single harmonic set tracking algorithm. Its application is tracking the heart rate in ECGs. Robust heart rate tracking in ECGs is challenging because real ECG signals are often contaminated by severe noise due to medical interventions and/or mechanical system interferences. The proposed multi-harmonic tracking algorithm can track



(a) EKF-based multi-harmonic tracker



(b) Proposed multi-harmonic tracker

Figure 1.3: Estimated frequencies using (a) the EKF-based multi-harmonic tracker and (b) the new multi-harmonic tracker on top of the spectrogram of the synthetic ECG signal.

the heart rate robustly even when such severe noise is present in the signals. *The fourth* is to develop an amplitude modulated harmonic set tracking algorithm. Here, the signal of interest is modeled to have two multi-harmonic sets that interact with each other through amplitude modulation. One assumption is made that the fundamental frequency of the first harmonic set is constant and known. The main goal, then, is to estimate the degree of the amplitude modulation of the second harmonic set due to the first one. Its application is to track the heart rate and pulse pressure variation (PPV) index of arterial blood pressure (ABP) signals recorded from subjects on full respiratory support. The PPV index is a critical parameter to monitor fluid responsiveness in the operating room and critical care settings. *The final objective* is to develop a multiple harmonic set tracking algorithm. Multiple harmonic set tracking is the most challenging multi-harmonic tracking case, where the number of multi-harmonic sets is more than one and the fundamental frequency of each harmonic set is unknown and time-varying. Its application is to track the heart rate, respiratory rate, and PPV index of ABP signals recorded from subjects with spontaneous breathing. Here the number of multi-harmonic sets is two.

1.4 Overview

The rest of the dissertation consists of nine chapters: literature review, problem definition, contributions, new particle filtering method, single rhythmical component tracking, single harmonic set tracking, amplitude modulated harmonic set tracking, multiple harmonic set tracking, and summary and conclusions. Chapter 2 (literature review) reviews traditional and latest frequency/multi-harmonic tracking algorithms. They are categorized into six large groups mainly based on technologies.

Chapter 3 (problem definition) defines the rhythmicity tracking problem as continuous posterior distribution estimation. Within the posterior distribution estimation framework it will be shown that in multi-harmonic tracking the posterior distribution of the fundamental frequency is multi-modal, which is the reason that traditional frequency tracking techniques often fail.

Chapter 4 (contributions) lists specific goals and contributions of the dissertation.

Chapter 5 (new particle filtering method) describes details of two versions of the new particle filtering technique and other conventional variants of particle filters.

Chapter 6 (single rhythmical component tracking) discusses a practical application of the single rhythmical component tracking case. The application is tracking tremorous activity exhibited in neuronal signals of subjects with movement disorders such as Parkinson's disease (PD) and essential tremor (ET).

Chapter 7 (single harmonic set tracking) discusses a practical application of the single harmonic set tracking case, which is tracking the heart rate in electrocardiograms (ECG).

Chapter 8 (amplitude modulated harmonic set tracking) discusses a practical application of the amplitude modulated harmonic set tracking case, which is tracking the pulse pressure variation (PPV) index of arterial blood pressure (ABP) signals under mechanical ventilation.

Chapter 9 (multiple harmonic set tracking) discusses a practical application of the multiple harmonic set tracking case. The application is tracking the pulse pressure variation (PPV) index of arterial blood pressure (ABP) signals of spontaneously breathing subjects.

Chapter 10 (summary and conclusions) summarizes and concludes the entire dissertation.

Appendix A provides the notational conventions used throughout this dissertation.

Chapter 2

Literature Review

Accurate detection of rhythmical components (single-frequency or multi-harmonics) in observed signals has been an important task in various applications such as speech analysis, communication, seismic analysis, and target tracking/identification. There are various technologies available in the literature. Some of them are developed for specific types of signals while others are more generally applicable. Some of them are quite heuristic while others are principled or based on models with clear assumptions. The scope of frequency/multi-harmonic detection/tracking is too wide to cover completely. However, the frequency/multi-harmonic detection/tracking problem can be divided into five groups as follows,

1. Fixed single-frequency with fixed amplitude
2. Fixed single-frequency with time-varying amplitude
3. Time-varying single rhythmical component with fixed amplitude
4. Time-varying single rhythmical component with time-varying amplitude
5. Time-varying multi-harmonic components with time-varying frequency and amplitude

Depending on whether quantities of interest are fixed or time-varying these can be treated as an estimation or tracking problem. Although the categorization above is helpful to grasp the scope of frequency detection and tracking problems, it is not easy to match each frequency detection/tracking technique with one of those categories. For this reason, this chapter is organized based on the tracking methodology. One exception is that several distinct methods are grouped under the category of pitch detection algorithms (PDAs). Although each of those methods utilizes distinct technologies, they are specifically developed in the context of pitch detection. They share some common assumptions such as a local stationarity of signals. Although the review does not cover all techniques available in the literature, it includes most common frequency/multi-harmonic detection/tracking methods at the time of writing.

2.1 Zero-crossing method

A zero-crossing method is the simplest way to detect the frequency of rhythmical components in narrow-band signals. This method is applicable when the amplitude of rhythmical components in the signal is assumed to be known or not of interest. Let's assume that a signal of interest is narrow-banded and locally stationary. Since it is a narrow-band signal, a simple sinusoidal model can describe it well as follows,

$$\begin{aligned} \boldsymbol{y}_n &= \sin(2\pi \boldsymbol{f} n T_\mathrm{s}) + \boldsymbol{v}_n, \\ \boldsymbol{f} &= \frac{1}{2\boldsymbol{T}} \end{aligned} \tag{2.1}$$

where n is a discrete time index, $\boldsymbol{y}_n$ a signal sample (observation or measurement), N the number of samples within a short time period, T_s a sample interval, $\boldsymbol{f}$ the estimate frequency, and $\boldsymbol{T}$ the mean interval between zero-crossings within the time period, $0 \leq n \leq N$.

A good example of the zero-crossing method application is seismic trace analysis. Since seismic signals are narrow-band, one can estimate the vibration-frequency of seismic activities, which is proportional to the inverse of zero-crossing intervals, using the zero-crossing method. Allen introduced an automatic earthquake recognition and timing program on a single seismic trace based on the zero-crossing algorithm [8]. The generic zero-crossing method, however, is sensitive to noise and cannot measure the intensity (amplitude) of rhythmical activities recorded in the signal.

Several authors have proposed more advanced zero-crossing methods by combining the generic zero-crossing method and other signal processing techniques. For example, Lim *et al.* proposed an automatic method for real-time electroencephalogram (EEG) sleep state analysis [9]. Their method is a combination of zero-crossing and peak detection algorithms. Nguyen *et al.* introduced a frequency computation technique for real-time monitoring, control, and protection of power systems [10]. They generalized the zero-crossing detection to a level crossing detection, which is insensitive to amplitude changes and able to track small frequency deviations as a fraction of a cycle. Miller utilized the zero-crossing method in combination with energy measurements to determine the fundamental frequency of speech by segmenting the signal into pitch periods [11]. There are other examples of applying the variants of the generic zero-crossing method to frequency detection problems [12–14]. For example, Rabiner and Schafer proposed averaging the number of zero-crossings over a short period of time to reduce the variance of the zero-crossing estimate [15].

The advantage of the zero-crossing method over other frequency detection and tracking algorithms is computational simplicity. Implementing the zero-crossing based frequency detector on hardware only requires comparator circuits and counters. However, the zero-crossing method itself cannot measure the amplitude of rhythmical components unless other energy measurement methods are combined with it. More importantly, harmonic structures buried in the signal do not benefit it or improve its accuracy since harmonic structures do not meet the narrow-banded signal assumption that the zero-crossing method is based on.

2.2 Angle Difference Estimator

The basic notion of the angle difference estimator is, simply, that the derivative of the continuous angle $\phi(t)$ is the continuous instantaneous frequency $f(t)$:

$$f(t) = \frac{1}{2\pi}\frac{\mathrm{d}\phi(t)}{\mathrm{d}t}. \tag{2.2}$$

An important issue to address before discussing how to implement the angle difference estimator is how to define the differentiation operation in discrete time, since all signals throughout this dissertation are assumed to be discrete-time signals sampled at a proper frequency f_{s}. One solution is to convolve the angle with a finite impulse response (FIR) differentiator d_n as defined in [16]:

$$f_n = \frac{1}{2\pi}\phi_n * d_n \tag{2.3}$$

where $*$ denotes the time-domain convolution operation. However, it is common to approximate this differentiation operation with a simple differencing operation since the differencing operation (FIR filtering) is the simplest case of performing the time-domain convolution. There are several ways to define the discrete-time difference operation as follows,

$$\hat{f}_{n,1} = \frac{1}{2\pi}\left(\phi_n - \phi_{n-1}\right) \tag{2.4}$$

$$\hat{f}_{n,2} = \frac{1}{2\pi}\left(\phi_{n+1} - \phi_n\right) \tag{2.5}$$

$$\hat{f}_{n,3} = \frac{1}{4\pi}\left(\phi_{n+1} - \phi_{n-1}\right) \tag{2.6}$$

Third definition $\hat{f}_{n,3}$ is preferable to the others because it yields zero group delay [17,18].

The first step to develop the angle difference estimator is to obtain the *analytic signal* associated with the observed signal y_n using the discrete-time *Hilbert transform*. The analytic signal z_n is a complex-valued version of a real-valued signal y_n where the negative frequency components of the Fourier transform of y_n are discarded. It can be expressed as,

$$z_n = y_n + jH[y_n] \tag{2.7}$$

where $H[\cdot]$ denotes the discrete-time Hilbert transform. Assuming that $a_n \cos(\phi_n)$ is a good model for the observed signal y_n, the phase between the real and imaginary parts of z_n is ϕ_n since $z_n = a_n e^{j\phi_n}$ [17]. The second step is to model the instantaneous angle as a discrete-time polynomial of arbitrary order as given,

$$\phi_n = \sum_{p=0}^{M} \beta_p n^p. \tag{2.8}$$

Then, by definition the estimate of the discrete-time frequency is,

$$\hat{\boldsymbol{f}}_n = \frac{1}{2\pi} \sum_{p=1}^{M} p\beta_p n^{p-1}. \tag{2.9}$$

Now, there are two formulae (2.6) and (2.9) for the discrete-time frequency. The third step is to set an equation as follows and obtain its solution,

$$\hat{\boldsymbol{f}}_n = \frac{1}{2\pi} \sum_{p=-\frac{N}{2}}^{\frac{N}{2}} \alpha_p \phi_{n+p} = \frac{1}{2\pi} \sum_{p=1}^{M} p\beta_p n^{p-1} \tag{2.10}$$

where the values of coefficients α_p are pre-determined based on the order N of the discrete-time differencing operation [18]. There are more advanced techniques to reduce the variance of the angle difference estimator by proper filtering and smoothing [19].

No matter how accurate the angle difference estimate can be, it is not applicable to the multi-harmonic tracking problem for several reasons. First, it cannot take advantage of harmonic structures to improve its performance. Second, the amplitudes of rhythmical components have to be estimated separately using other techniques. Third, the estimate is on the frame-by-frame basis not sample-by-sample basis. In addition to all of these issues there is a more fundamental problem using the angle difference estimation technique. In order to apply the Hilbert transform to the signal, $\boldsymbol{y}_n$, it should meet the necessary conditions discussed in [17]. Otherwise the instantaneous angle ϕ_n is susceptible to error. I demonstrated that the Hilbert transform itself introduces spurious noise in the angle when the signal does not meet the necessary conditions [20].

2.3 Time-Frequency Distributions (TFDs)

Over the last 60 years many researchers have tried to devise useful tools to describe the changes of a nonstationary signal's power density in time- and frequency-domains simultaneously. The spectrogram or short-time Fourier transform (STFT) is currently the most common tool to study time-varying spectral contents of signals. However, it has an inevitable weakness that is the trade-off between the time- and frequency-domain resolutions. In order to achieve a better frequency-domain resolution, the length of frames should be large, which results in a poor time-domain resolution. The time-domain resolution can be improved by using short-length frames. But this results in a poor frequency-domain resolution. This trade-off could be a minor issue when the spectral contents of the nonstationary signal change slowly. However, when the spectral contents of the nonstationary signal change rapidly, the spectrogram cannot capture the rapid change of the spectral contents appropriately. For example, the speech signal is highly nonlinear and its spectral component changes rapidly. The spectrogram or STFT is not a suitable tool to analyze the speech signal.

These unique characteristics of speech signals motivated the early study of time-frequency distributions (TFDs) [21]. TFDs estimate the frequency distribution of the signal at a given time and calculate the fraction of the energy in a certain frequency or time-range. In the 1930's, Wigner studied the statistics of quantum mechanics [22]. Based on his work, Gabor, Ville, and Page separately developed the early version of TFDs in the 1940's and 50's [23–25]. The prevalent form of the early version of TFDs is called the *Wigner-Ville distribution.* It can be written as follows,

$$\boldsymbol{w}(t,\omega) = \frac{1}{2\pi}\int \boldsymbol{s}^*\left(t-\frac{1}{2}\tau\right)e^{-j\tau\omega}\boldsymbol{s}\left(t+\frac{1}{2}\tau\right)\mathrm{d}\tau \tag{2.11}$$

where * represents a complex conjugate operation. The Wigner-Ville distribution can be interpreted as the short time Fourier transform of the signal's autocorrelation. Although the Wigner-Ville distribution has been a common tool for the time-frequency domain analysis for many years, the Wigner-Ville distribution introduces the spurious peaks, which are not related with real spectrum contents of the signal [26]. Cohen introduced a more general form of the Wigner-Ville distribution, which can be written as follows,

$$\boldsymbol{p}(t,\omega) = \frac{1}{4\pi^2}\int\int\int \boldsymbol{s}^*\left(u-\frac{1}{2}\tau\right)e^{-j\theta t}e^{-j\tau\omega}e^{j\theta u}\phi(\theta,\tau)\boldsymbol{s}\left(u+\frac{1}{2}\tau\right)\mathrm{d}u\mathrm{d}\tau\mathrm{d}\theta \tag{2.12}$$

where $\phi(\theta,\tau)$ is a user-defined function or kernel [27]. According to Cohen's TFD formula the Wigner-Ville distribution is a special case of this general form, which can be obtained by substituting $\phi(\theta,\tau)$ with 1. Cohen has summarized what kind of a kernel $\phi(\theta,\tau)$ is required to obtain different types of TFDs [28].

Many research groups have put their effort into utilizing the TFDs to analyze nonstationary signals. In the early 1980's, for example, Claasen *et al.* discussed the properties of the original Wigner distribution as a tool for continuous and discrete time-frequency signal analysis [29,30]. Especially in [31] they studied the relationship between the Wigner distribution and other time-frequency signal transforms such as spectrograms. They pointed out that spectrograms are a weighted version in time and frequency of the Wigner distribution. They also demonstrated that one can interpret the spectral analysis method in terms of the Wigner distribution. Later, Cohen categorized the applications of the TFDs into three groups [28]. The first subject is to use the TFDs to reveal meaningful information that other methods cannot. For example, the TFDs can capture more rapid transients in speech signals than the spectrogram. The second subject is to capture a certain property of the signal's time-frequency content that can be well represented by a given TFD, although the TFD may not be a good choice to study other properties of the signal. The last subject is to use the TFDs as "a carrier of the information of a signal". In this case, it doesn't matter whether the TFDs are the good representation of the time-frequency energy density. For example, Boashash utilized the Wigner-Ville distribution as a pattern recognition tool [32,33]. The early work of Boashash was probably the first attempt to use the Wigner-Ville distribution for real applications [34]. He studied the instantaneous frequency of a signal reflected from the ground using the Wigner-Ville distribution to calculate the attenuation coefficients of the ground. Janse and Kaizer used the Wigner-Ville distribution for the design of loudspeakers, where the non-stationarity of signals must be handled properly [35]. Chester *et al.* pointed out that a usual discrete Fourier transform is not a good tool to represent some frequency domain attributes of human speech [36]. They presented an important discrete Wigner distribution to uncover these frequency-domain attributes of human speech. Chester and Wilbur discussed theoretical and implementation aspects of a band-selectable discrete Wigner distribution for signal analysis [37]. In the same year, Martin and Flandrin proposed a general class of spectral estimators of the Wigner-Ville spectrum of nonstationary processes [38]. By laying the generalized framework for spectral estimation they showed the versatility of the "new pseudo-Wigner estimators." More recently, Pola *et al.* performed a comparative evaluation among classical (FFT-based) and time-frequency distribution (TFD) spectral estimators [39]. As expected, the TFD-based spectral estimator was superior to classical (FFT-based) methods when a high time resolution is needed. In order to monitor the heart rate variability (HRV) of subjects they applied the estimators to real interbeat interval time series, which is an ununiformly sampled ECG. The results demonstrated that the TFD-based spectral estimator can represent the rapid transient of the time series correctly. Chan *et al.* applied five time-frequency methods to synthetic and real heart rate signals to study HRV characteristics [40]. They are the short-time Fourier transform, the Choi-Williams distribution, the smoothed pseudo Wigner-Ville distribution, the filtering smoothed pseudo Wigner-Ville distribution compensation, and the discrete wavelet transform. According to their assessment results the filtering smoothed pseudo Wigner-Ville distribution and the discrete wavelet transform are better than the rest in terms of minimizing spectrum interference from the transient component.

There are only a few references in the literature, that utilize TFDs for analysis of multi-harmonic signals [1,2]. For example, Jones *et al.* proposed an adaptive time-frequency distribution for the purpose of improving the time-frequency resolution of multi-component signals [41,42]. Zhang *et al.* proposed a novel adaptive harmonic Fourier transform for voiced speech signal analysis [2]. Their method has a better resolution than the classical short-time Fourier transform and minimizes the cross interferences produced by the Wigner-Ville distribution.

Despite considerable effort of many investigators to make the TFDs a useful tool for time-frequency domain analysis of nonstationary signals, there is a main drawback of using the TFDs for the multi-harmonic component analysis: it is not trivial to obtain the numerical values of the fundamental frequency and energy density given the TFDs. Although the TFDs are a good tool to visualize the transients and transitions of nonstationary signals, extracting the numerical values of the frequency and energy density correctly out of the TFDs is difficult. Another issue with the TFDs is that some TFDs, such as the Page distribution, may not represent actual energy density since the longer a particular frequency is observed, the larger the energy density becomes at that frequency [28].

2.4 Adaptive Frequency Estimation Methods

There are various adaptive frequency tracking methods in the literature. Some of them are popular and general while others applicable only for specific applications. It is not feasible to discuss all of them here. However, it is a reasonable attempt to categorize those popular adaptive frequency tracking methods into three groups: adaptive line enhancer (ALE), adaptive notch filtering (ANF), and adaptive comb filtering (ACF). Although they are devised for different purposes, these adaptive techniques are based on the least-mean-square (LMS) algorithm introduced by Widrow and Hoff in 1959 [43]. The LMS algorithm is an adaptive algorithm based on a gradient-based method of steepest descent, which finds the nearest local minimum of a given function [44]. In the 1970's Widrow *et al.* described the principles and applications of adaptive noise cancelling, which is a method of estimating signals corrupted by additive noise or interference [45]. One of the major contributions of their work is to identify the self-tuning filter and notch filter as the special cases of the adaptive noise canceler. The self-tuning filter, which can retrieve the periodic and stochastic components of a signal, serves as an adaptive line enhancer. The notch filter, which suppresses an additive undesired sinusoidal interference, can be realized by an adaptive noise canceler, whose bandwidth is easy to control. Treichler studied the transient and convergent behavior of the ALE using an eigenvalue-eigenvector analysis of the ALE impulse response [46]. Around that time Zeidler *et al.* introduced the ALE for multiple sinusoid retrieval in uncorrelated white noise [47]. The difference between the ALE and ANF is enhancing or suppressing the periodic components of given signals. This difference, however, is not significant. Rao *et al.* demonstrated that the ANF can be used as the ALE with proper configuration of delay taps [48]. They also proposed an effective way to build an IIR filter that requires only half the number of parameters in comparison to the ARMA model introduced by Friedlander [49]. Although the ALE and ANF have been applied for various noise cancelling applications successfully [45, 50], they are not suitable for harmonic tracking for several reasons. The first reason is that they are not capable of tracking harmonically related components in the signal. There are some papers that propose the ALEs for multiple sinusoid component retrieval [47, 51, 52]. However, multiple sinusoid component retrieval is not the same as multi-harmonic tracking. The second reason, which is more critical, is that the ALE/ANF may converge to the nearest local minimum of the performance error surface. In the case of multi-harmonic tracking, the performance error surface has many local minima. Therefore, the ALE/ANF would not be able to converge to the global minimum unless they are initialized carefully to be close to the global minimum.

Adaptive comb filtering is an appropriate method to handle harmonically related components in the signal. Adaptive comb filters differ from the ALE/ANF due to their ability to enhance or attenuate multi-harmonic components in the signal *jointly*. Comb filters are typically in two general forms: Feed-forward and feedback forms. The comb filter in the feed-forward form filters out harmonic components in the signal $\boldsymbol{x}_n$ by delaying $\boldsymbol{x}_n$ by k samples. The difference function and the Z-domain transfer function of are as follows,

$$\begin{aligned} \boldsymbol{y}_n &= \boldsymbol{x}_n + \alpha\ \boldsymbol{x}_{n-k} \\ H(z) &= \frac{Y(z)}{X(z)} = \frac{z^k + \alpha}{z^k}. \end{aligned}$$

On the other hand, the comb filter in the feedback form retrieves harmonic components in the signals $\boldsymbol{x}_n$ by delaying $\boldsymbol{y}_n$ by k samples. The difference function and the Z-domain transfer function are as follows,

$$\begin{aligned} \boldsymbol{y}_n &= \boldsymbol{x}_n + \alpha\ \boldsymbol{y}_{n-k} \\ H(z) &= \frac{Y(z)}{X(z)} = \frac{1}{z^k - \alpha}. \end{aligned}$$

The difference between them can be seen as whether there are either an all-zero system (feed-forward) or an all-pole system (feedback).

Elliott *et al.* introduced a simple adaptive comb filter using the least-mean-square (LMS) algorithm [53]. Soon after his publication, Nehorai proposed an adaptive comb filter for harmonic signal enhancement with two cascaded parts: one is for fundamental frequency estimation and the other for harmonic amplitudes and phases estimation [54]. The fundamental frequency estimation uses the recursive maximum likelihood (RML) method while the harmonic amplitudes and phases estimation the recursive least square (RLS) methods [55]. In the paper he argued that simulation results indicate that the variances of the estimates are of the same order of magnitudes as the Cramer-Rao bound (CRB) for sufficiently large data sets. Veeneman *et al.* applied a fully adaptive comb filter to enhance block-coded speech signals corrupted with framing noise caused by discontinuities at the block boundaries [56]. Their comb filter is advanced to adapt *both pitch period and filter coefficients.* There have been several attempts to build more optimal and efficient comb filters for time-varying harmonic extraction [3,57,58]. More recently several research groups have implemented the comb filter as an adaptive filter-bank [59,60]. Especially Sun *et al.* performed a statistical analysis of a modified adaptive filter-bank for the estimation of multi-harmonic signals [59].

Jang *et al.* proposed an adaptive IIR comb filter for harmonic signal cancellation [3]. They argue that the proposed comb filtering is more robust than Nehorai and Poraz's comb filtering technique [54]. However, the main contribution of their paper is to recognize local minima problems of notch and comb filters and demonstrate mathematically that the error surface of the comb filter has multiple minima. All notch and comb filtering techniques rely on some kind of optimization methods to search for the best set of fundamental frequencies and the amplitudes of harmonic components. The local minima problem stems from the fact that the optimization methods, e.g. least-mean-square (LMS) or recursive least-square (RLS), cannot handle the multi-harmonic tracking error surface, which has multiple minima. Although Jang *et al.* argues that their comb filter is more robust to this local minima issue than conventional comb filters, they implicitly admit that the convergence of their comb filter model does rely on the carefully chosen initial conditions [3]. Overall, adaptive notch and comb filtering techniques in the literature require cautious initialization to guarantee the filter's convergence.

2.5 Pitch Detection Algorithms (PDAs)

One of common multi-harmonic signals is human voice. It is sound made by a human using his vocal folds, tongue, and lips. The frequency of the fundamental partial in a sound is called, *pitch*. The goal of pitch detection algorithms (PDAs) is to determine the pitch of a quasiperiodic signal such as human voice. Pitch detection is an essential part of various speech processing systems. It is useful for speaker recognition, speech instruction to the hearing impaired, and vocoder (speech analysis-synthesis systems) [61]. In the course of some 40 years, numerous PDAs have been proposed for more accurate and robust pitch detection [62]. But, they are sensitive to noise and interference and limited to particular domains (speech or song) and their performance becomes unacceptable at low SNRs. The following summary attempts to cover the most prevalent PDAs rather than the complete list of them. Additional details are available in [63] or [62].

PDAs are generally categorized into two main classes: time-domain PDAs and short-term analysis PDAs [64]. The time-domain PDAs include zero-crossing rate (ZCR) [11], slope event rate, autocorrelation method [4], and the YIN estimator [5] that is based on the difference function. They are simple to understand and computationally efficient in comparison to short-term analysis PDAs [62]. Those methods, however, do not take an advantage of the harmonic relationship between the fundamental and harmonic partials, which provides useful information to reduce the variance of the frequency estimation. They are only useful for only narrow-band signals and susceptible to noise. The short-term analysis PDAs are based on some type of short-term transformation. In these methods the speech signal is split up into a series of frames. Each frame is simply a small number of consecutive samples of the signal, which may overlap with other adjacent frames. A single frame should be at least as long as two or three complete periods. Although numerous types of short-term analysis PDAs have been proposed, they belong to either correlation techniques or frequency-domain analysis [64]. The correlation techniques include the autocorrelation method and the average magnitude difference function (AMDF). These techniques measure how similar the speech signal is to itself at a given time lag (delay). If the signal is periodic or quasi-periodic, the time lag (delay) on which the measure of the similarity increases will represent the period of the signal.

The typical frequency-domain analysis is a cepstrum. A cepstrum is the result of taking the Fourier transform of the logarithmic spectrum of the signal. The cepstrum is plotted against the quefrency, which is a measure of time (similar to the time lag). The main idea of the cepstrum is that when the signal contains harmonic components, the spectrum of the signal itself comes to have several peaks at the frequencies corresponding to these harmonics and appears to be periodic. The Fourier transform of the logarithmic spectrum is going to have a large value at a quefrency which corresponds to the distance between the spectrum peaks. Overall, the short-term analysis PDAs are computationally more intensive than the time-domain PDAs. However, they are in general more accurate and more robust to noise.

Two main sources of errors in PDAs are voicing decision errors and false pitch estimates due to noise and signal distortion within voiced segments. Under bad signal conditions, such as low signal-to-noise ratio (SNR), the performance of PDAs deteriorates severely. One common solution to noisy pitch estimates is *pitch tracking*. Instead of estimating pitches independently based on individual frames, pitch tracking methods adopt more global searching mechanisms using dynamic programming (DP) and/or hidden Markov model [63]. Wang *et al.* proposed a PDA that searches to find an optimum pitch track using a DP [65]. A few years earlier Droppo *et al.* suggested a time-domain MAP pitch tracking method utilizing two-state hidden Markov model (HMM) [66]. Recently, Tabrikian *et al.* proposed a MAP pitch tracking method that significantly outperforms the previous methods in noisy environments [63].

There are two main drawbacks of the PDAs. First, they are not as accurate as state-space based methods since they are frame-based and have only a discrete representation of the frequency possibilities. Second, they can only estimate or track the pitch of the quasi-periodic signal without providing any information on the amplitudes of the fundamental and harmonic partials. In other words, unless the PDAs are combined with other signal processing techniques, they are not an appropriate solution to track the fundamental frequency and the amplitudes of its harmonics.

Speech signal representation is a more general topic of speech signal analysis than pitch detection. The goal of speech signal representation is to analyze a speech signal to obtain its complete characterization, which provides enough information based on which one can re-synthesize a quasiperiodic portion of the speech signal as close as possible. The first step of speech signal representation is to model a glottal excitation waveform. Traditionally researchers used multiple impulses or random sequences superimposed on periodic pitch pulses to model a glottal excitation waveform [67]. In the early 80's, however, several analysis/synthesis approaches were proposed using the sine-wave model [68, 69]. In the mid 80's, McAulay proposed the sinusoidal representation of speech signal as a new approach to this problem. McAulay *et al.* assumed that the excitation is composed of sinusoidal components of arbitrary amplitudes, frequencies, and phases [7]. After modeling the glottal excitation as a sum of sine waves, which results in the sinusoidal representation for speech signal, speech signal representation becomes a problem of parameter extraction for amplitudes, frequencies, and phases. The first step of the parameter extraction is to break down the speech signal into contiguous sequential frames. Then, for each frame, a set of parameters (amplitudes, frequencies, and phases) is selected that minimizes the mean-squared error between the framed $\boldsymbol{y}_n$ and synthetic $\boldsymbol{s}_n$ speech signals. This approach is equivalent to a Fourier series representation of a periodic signal. In other words, the set of parameters of the synthetic speech model can be estimated from the short-time Fourier transform (STFT) of the speech signal. This approach, however, is non-recursive imposing a heavy computational load and is based on the local stationarity assumption within each frame. One frame should also contain several fundamental periods for accurate estimation of the pitch. One hazard of this approach is that the assumption of stationarity does not hold within speech regions where both the location and the number of peaks change rapidly. McAulay introduced the concept of *birth* and *death* of sinusoidal components to address this issue. However, his method is heuristic and requires additional user-specified parameters that are application-specific. McAulay's modeling of the glottal excitation as a sum of sine waves was ingenious, but his parameter extraction method left a lot of room for improvement in terms of the optimality, efficiency, and robustness. The following sections will discuss alternative methods to extract parameters of the sinusoidal model of quasi-periodic signals efficiently and accurately.

2.6 Continuous State-Space Model Based Algorithms

A *Markov process* is a mathematical model for a random system whose future and past states are *independent* conditional on the present state. A Markov process is memoryless since the future and past states of the system are independent conditional on the present state. A Markov process is a useful signal processing tool because many signal processing applications can be modeled as Markov processes. After modeling, the main goal is to estimate the state of the system given observable measurements that are related to that state through a known model,

$$\boldsymbol{x}_{n+1} = f(\boldsymbol{x}_n, \boldsymbol{u}_n) \qquad\qquad \boldsymbol{y}_n = h(\boldsymbol{x}_n, \boldsymbol{v}_n) \tag{2.13}$$

where $\boldsymbol{x}_n$ is the state vector, $\boldsymbol{y}_n$ is the observation vector, $\boldsymbol{u}_n$ is process noise, $\boldsymbol{v}_n$ is the measurement noise, $f(\boldsymbol{x}_n, \boldsymbol{u}_n)$ is the process model, and $h(\boldsymbol{x}_n, \boldsymbol{v}_n)$ is the measurement model. The initial state $\boldsymbol{x}_0$ is assumed to be distributed according to a density function $\pi_0(\boldsymbol{x}_0)$.

About 50 years ago, Kalman derived a set of equations for efficient recursive filtering to optimally estimate the state of *linear* systems [70]. The system is *linear* when both process and measurement models are linear. A linear system can be generally expressed as,

$$\boldsymbol{x}_{n+1} = A_n\boldsymbol{x}_n + B_n\boldsymbol{u}_n \qquad\qquad \boldsymbol{y}_n = C_n\boldsymbol{x}_n + D_n\boldsymbol{v}_n \tag{2.14}$$

where A_n is the state-transition matrix, B_n the input matrix, C_n the output matrix, and D_n the measurement noise matrix. However, the state estimation problem becomes difficult if either of the process and measurement models are nonlinear. The multi-harmonic tracking application is an example of a nonlinear Markov process where the process model is nonlinear due to the nonlinear relationship between the angles $\boldsymbol{\theta}_n$ and observed signal $\boldsymbol{y}_n$. The subsequential sections will discuss three popular suboptimal approaches: extended and unscented Kalman filtering, hidden-Markov model, and sequential Monte Carlo methods.

2.6.1 Extended and Unscented Kalman Filtering

The extended Kalman filter (EKF) and unscented Kalman filter (UKF) rely on local linear approximations of nonlinear systems. When the goal is to estimate only the frequency of a signal with a single partial, the EKF is equivalent to the digital phase-locked loop (PLL) [71, 72]. Schmidt suggested the linearization of a nonlinear system about the current state estimate, which has become the most widely used nonlinear state-space estimator, i.e. the extended Kalman filter [73]. In 1990 Parker *et al.* applied this extended Kalman filtering technique to estimate the amplitudes, frequencies, and phases of the rhythmical signals containing N_{h} harmonic components [74]. In many ways the EKF is more powerful and efficient than the STFT-based methods that McAulay proposed in [7]. First, it is no longer necessary to break down the quasi-periodic signal into contiguous sequential frames. Using the EKF one can estimate the state of the system on a sample-by-sample basis. Secondly, the EKF does not require the local stationarity of the signal. Thirdly, estimation, i.e. parameter extraction, is carried out recursively from time $n-1$ to time n. The state-space model that Parker proposed can be written as follows,

$$\boldsymbol{y}_n = \boldsymbol{a}_{1,n} \cos(\boldsymbol{\theta}_n) + \sum_{k=2}^{N_\mathrm{h}} \boldsymbol{a}_{1,k,n} \cos(k\boldsymbol{\theta}_n) + \boldsymbol{a}_{2,k,n} \sin(k\boldsymbol{\theta}_n) + \boldsymbol{v}_n \tag{2.15}$$

where N_h is the number of harmonics (assumed known), $\boldsymbol{\theta}_n$ is the fundamental instantaneous angle, $\boldsymbol{a}_{1,k,n}$ and $\boldsymbol{a}_{2,k,n}$ are the sinusoidal coefficients, and $\boldsymbol{v}_n$ is white Gaussian noise with variance R_v. The recursions for the EKF are available in many references [75].

Although Parker *et al.* demonstrated the outstanding performance of the EKF estimating the frequency and amplitude of a periodic signal based on simulation study, they missed a very important aspect of the multi-harmonic tracking problem. As Jang *et al.* mathematically proved, the error surface contains several local minima in multi-harmonic tracking applications [3]. In other words, the more harmonic partials the multi-harmonic signal contains, the more severely the posterior distribution $p(\boldsymbol{x}_{0:n}|\boldsymbol{y}_{0:n})$ becomes multi-modal, where $\boldsymbol{y}_{0:n}$ is a sequence of observations $\boldsymbol{y}_0, \ldots, \boldsymbol{y}_n$. This is problematic since since the EKF relies on local linear approximations of the system. As a result, the convergence of the EKF heavily depends on the initial conditions of the EKF recursions. The EKF is also sensitive to abrupt changes in frequency, signal drops, and artifact, which are common in real signals. Therefore, the EKF is not a suitable method for multi-harmonic tracking applications.

2.6.2 Hidden Markov Models (HMMs)

The basic theory of Markov chains (MCs) and hidden Markov models (HMMs) has been known to mathematicians for over 100 years. However, it was only a

Table 2.1: List of HMM Elements.

Name	Symbol
Length of the observation sequence	T
No. of hidden states	N
No. of observation symbols	M
Hidden states	Q
Discrete set of possible symbol observations	V
Hidden state transition probability distribution	A
Observation symbol probability distribution in hidden state j	B
Initial state distribution	π

few decades ago when people started applying it explicitly to signal processing problems. The definition of an HMM is "a doubly stochastic process with an underlying stochastic process that is not observable (it is hidden), but can only be observed through another set of stochastic processes that produce the sequence of observed symbols" [76]. There are eight elements of an HMM which are listed in Table 2.1.

It is common to represent an HMM using a simple notation $\lambda = (A, B, \pi)$ where the three distributions, A, B, and π, are the most important elements of an HMM. The state transition probability distribution, $a_{ij} = p(q_j \text{ at } n+1 | q_i \text{ at } n)$, relates the current hidden state q_i at n to the next hidden state q_j at $n+1$, where n is a discrete time index. The observation symbol probability distribution, $b_i(k) = p(v_k \text{ at } n | q_i \text{ at } n)$, relates the current hidden state q_i at n to a symbol v_k at n. There are three types of problems that can be solved for a given HMM: the first is calculating the probability of the observation sequence, the second choosing a hidden state sequence, which is "optimal" given the observation sequence, and the third tuning the model parameters, $\{A, B, \pi\}$, to maximize the probability of the observation sequence. The second problem is closely related with the frequency tracking application uncovering the optimal hidden state sequence of the model. The hidden state sequence in the frequency tracking application is the frequency track.

Streit *et al.* demonstrated how the problem of frequency line tracking can be formulated in terms of HMMs [77]. In their HMM frequency line tracking algorithm the A matrix represents the likely extent of the frequency fluctuations and the B matrix characterizes the statistical relation between the hidden state (frequency) at time n and the measurement at time n. A user-specified range of frequencies is divided into a finite number of *frequency cells*, which are hidden states of the model. It is important to notice that the measurement takes a form of a detection and relies on the fast Fourier transform (FFT) of the signal. If the spectral power in a particular frequency cell at time n is larger than that in all other frequency cells and exceeds a user-specified threshold D, the cell is claimed to be *detected* and its corresponding index is used as a measurement. Given the measurement sequence Z, which is a sequence of cell indices, the goal is to obtain the optimal state sequence I_{opt} that maximizes $p(Z|I)$. This optimal state sequence is referred to as the *Viterbi track* since the optimization is done using the Viterbi dynamic programming algorithm [77]. The B matrix is a function of the threshold D and SNR. These parameters play a critical role in optimizing the HMM frequency line tracker. Although the authors demonstrated that the threshold D can be optimized by solving a nonlinear equation, the equation includes other critical parameters to be optimized separately. The HMM frequency line tracker requires some user experience for proper optimization. The main weakness of the HMM frequency line tracker is that it can only track the frequency of an observed sine wave. A few years later, Barrett *et al.* extended the HMM frequency line tracker so that a newer version of the tracker can track not only the frequency but also the amplitude and angle of a noisy sine wave [78]. The main difference between the HMM frequency line tracker and frequency/amplitude/angle tracker is that the later uses complex FFT data instead of the absolute value of FFT. According to their simulation results the HMM frequency/amplitude/angle tracker outperformed the HMM frequency line tracker at the expense of increased numerical complexity and computational load. Although there is more recent work on the HMM-based frequency tracking [79,80], all HMM-based frequency tracking algorithms are limited to individual frequency tracking applications and not capable of tracking harmonically related partials in the signals such as speech, music, or various pressure signals.

A few research groups have proposed HMM-based harmonic/pitch tracking methods over the last decade. They modeled the signals of their interest such as speech and music signals as a sum of harmonically related sinusoids and used HMMs to estimate the smooth fundamental frequency track and the amplitude of each harmonic partial [6,63,81]. Deisher *et al.* modeled the speech signal as a sum of weighted sinusoids as follows,

$$\boldsymbol{y}_n = \sum_{k=1}^{N_\mathrm{h}} \boldsymbol{a}_{k,n} \cos\left(k\boldsymbol{\omega}_0 n + \boldsymbol{\phi}_{k,n}\right) \tag{2.16}$$

where N_h is the number of sinusoids used in the representation and $\boldsymbol{a}_{k,n}$ and $\boldsymbol{\phi}_{k,n}$ are the amplitude and angle associated with the k^{th} angular frequency $k\boldsymbol{\omega}_0$. They applied the HMM-based MMSE estimator to speech corrupted by additive noise, i.e. $\boldsymbol{z}_n = \boldsymbol{y}_n + \boldsymbol{v}_n$, to find the harmonic sinusoidal model parameters $\{\boldsymbol{a}_{k,n}, \boldsymbol{\omega}_0, \boldsymbol{\phi}_{k,n}\}$ of each block of clean speech. The first step of the estimation algorithm is to take the discrete Fourier transform (DFT) of a noisy speech block, which is denoted as $\boldsymbol{Z}$. Then, the DFT of a clean speech block, i.e. $\hat{\boldsymbol{Y}}$, is estimated using ergodic and Gaussian mixture HMMs. $\hat{\boldsymbol{Y}}$ is an optimal estimator of $\boldsymbol{Y}$ in a sense of MMSE. Given $\hat{\boldsymbol{Y}}$ the maximum likelihood frequency trajectory is calculated using the efficient Viterbi algorithm. Deisher *et al.* demonstrated reasonable performance of the proposed algorithm. Obtaining $\hat{\boldsymbol{Y}}$, however, requires training of two autoregressive HMMs using clean speech collected from a group of people representing potential users and noise from the expected operating environment. This training process is cumbersome and specific to certain applications. More recently, Tabrikian *et al.* proposed a maximum *a posteriori* (MAP) pitch tracking algorithm using harmonic models based on HMMs. This is more general and computationally more efficient than Deisher's algorithm. Their model for the measurement of a given voiced frame is slightly different from the previous one and can be written as,

$$\boldsymbol{y}_n = \sum_{k=1}^{N_\mathrm{h}} \boldsymbol{a}_{1,k,n} \cos\left(k\boldsymbol{\omega}_0 n\right) + \boldsymbol{a}_{2,k,n} \sin\left(k\boldsymbol{\omega}_0 n\right) + \boldsymbol{v}_n \tag{2.17}$$

where $\boldsymbol{\omega}_0$ stands for the fundamental angular frequency of the signal and the coefficients $\boldsymbol{a}_{1,k,n}$ and $\boldsymbol{a}_{2,k,n}$ carry the information on the intensity and angle of the k^{th} harmonic of the signal. (2.17), then, can be written in matrix notation as follows,

$$\boldsymbol{y} = \boldsymbol{A}\left(\boldsymbol{\omega}_0\right) \boldsymbol{c} + \boldsymbol{v} \tag{2.18}$$

where $\boldsymbol{c} \triangleq \left[\boldsymbol{a}_1, \ldots, \boldsymbol{a}_{N_\mathrm{h}}, \boldsymbol{b}_1, \ldots, \boldsymbol{b}_{N_\mathrm{h}}\right]^T$ and the matrix $\boldsymbol{A}(\boldsymbol{\omega}_0)$ can be partitioned as

$\boldsymbol{A}(\boldsymbol{\omega}_0) = [\boldsymbol{A}_c(\boldsymbol{\omega}_0)\, \boldsymbol{A}_s(\boldsymbol{\omega}_0)]$. The elements of $\boldsymbol{A}(\boldsymbol{\omega}_0)$ are given by,

$$\boldsymbol{A}_{c,k,n}(\boldsymbol{\omega}_0) = \cos(k\boldsymbol{\omega}_0 n) \tag{2.19}$$

$$\boldsymbol{A}_{s,k,n}(\boldsymbol{\omega}_0) = \sin(k\boldsymbol{\omega}_0 n). \tag{2.20}$$

The maximum likelihood function can be written as,

$$p\left(\boldsymbol{y}|\boldsymbol{\omega}_0, \boldsymbol{c}, \boldsymbol{\sigma}_{\boldsymbol{v}}^2\right) = \frac{1}{\left(2\pi\boldsymbol{\sigma}_{\boldsymbol{v}}^2\right)^{1/2}} \mathrm{e}^{-\|\boldsymbol{y}-\boldsymbol{A}(\omega_0)\boldsymbol{c}\|/(2\sigma_{\boldsymbol{v}}^2)}. \tag{2.21}$$

Then, the maximum likelihood (ML) estimator can be obtained by maximizing the likelihood function in (2.21) with respect to the unknown parameters $\{\hat{\boldsymbol{\omega}}_0, \hat{\boldsymbol{c}}, \hat{\boldsymbol{\sigma}}_{\boldsymbol{v}}^2\}$. One thing to notice is that this ML estimator provides only an independent parameter estimation for each individual frame. Incorporating *a priori* knowledge on the smooth behavior of speech parameters (especially frequency) over time they proposed the MAP estimator of the fundamental frequency track, that uses measurements collected over several frames. The MAP estimator can be implemented by using a dynamic programming procedure.

Most HMMs-based frequency tracking algorithms have been developed extensively for speech analysis. Although their performance has been improved over the last few decades, there are several weaknesses for the algorithms to be applied to broader applications. The first weakness is that most HMMs-based frequency tracking algorithms require the frequency domain representation of each frame such as the DFT as the input of the algorithms. This means that the signal has to be segmented by non-overlapping or overlapping frames assuming the local stationarity within each frame. The duration of individual frames is a limiting factor, which controls the trade-off between the frequency- and time-domain resolutions. For various applications, where the local stationarity assumption doesn't hold, the HMMs-based frequency tracking algorithms may not be appropriate. The second is that the HMMs-based frequency tracking algorithms require the *locally* estimated fundamental frequency estimates prior to *globally* optimal solution. The locally estimated fundamental frequency estimates rely on individual frames ignoring the prior statistical transition probability density function $\boldsymbol{A}$. Therefore, this so-called *measurement track* of the locally estimated fundamental frequency estimates is not a globally optimal estimate of the true track. The globally optimal estimate of the true track, so-called *Viterbi track*, can be obtained only after the algorithms see a large number of frames. In their simulation results Barrett *et al.* showed the difference between the measurement track and Viterbi track clearly [78].

2.6.3 Sequential Monte Carlo Methods

The extended Kalman filter relies on the mean and covariance statistics to represent the posterior marginal distribution $p(\boldsymbol{x}_n|\boldsymbol{y}_{0:n})$ of the state given a sequence of observations $\boldsymbol{y}_{0:n} = \{\boldsymbol{y}_0, \ldots, \boldsymbol{y}_n\}$. This, however, is a useful summary of the distribution only when the distribution is unimodal. For some nonlinear processes the posterior distribution is multi-modal and in these cases the distribution cannot be sufficiently described by the mean and covariance statistics. The multi-modal posterior distribution requires a more complete set of statistics to represent all of the important features. Monte Carlo methods (MCMs) are a possible alternative [82]. MCMs estimate an unknown distribution up to a normalizing constant. Sequential Monte Carlo methods (SMCM) carry out this estimation process sequentially (or recursively) to reduce the computation load of the algorithm.

In theory the posterior distribution is estimated up to a normalizing constant by drawing a sequence of sufficient random samples from the distribution. In practice, however, it is not possible to draw independent samples from the posterior distribution, $p(\boldsymbol{x}_{0:n}|\boldsymbol{y}_{0:n})$, that is unknown. The concept of an *importance sampling* needs to be introduced to address this issue. Importance sampling is a technique to represent the unknown posterior distribution as a weighted combination of point masses, where the independent samples are drawn from a *importance distribution*, $q(\boldsymbol{x}_{0:n}|\boldsymbol{y}_{0:n})$, which a user can choose. Only one constraint on the choice of the importance distribution is that its support must be larger than that of $p(\boldsymbol{x}_{0:n}|\boldsymbol{y}_{0:n})$. The expected value of any statistic of the state can then be estimated as follows,

$$\mathrm{E}[g(\boldsymbol{x}_{0:n})] \approx \frac{1}{N_\mathrm{p}} \sum_{i=1}^{N_\mathrm{p}} w_n^{(i)} g(\boldsymbol{x}_{0:n}^{(i)}) \qquad (2.22)$$

that converges as $\mathcal{O}(N_\mathrm{p}^{-1})$ by the law of large numbers. Often, the SMCM are referred to as *particle filters* (PF). The term, "particle", is used to represent the random state trajectories $\boldsymbol{x}_{0:n}$. A single particle, i, corresponds to the i^{th} random state trajectory $\boldsymbol{x}_{0:n}^{(i)}$.

Doucet praised the remarkable flexibility and usefulness of Monte Carlo algorithms that have been revolutionizing applied statistics and related fields such as bioinformatics and econometrics [83]. Although he tried to provide a complete review of MCM in a signal processing context in [83], he left out an important issue that needs to be addressed in applying the traditional MCM to many signal processing applications such as multi-harmonic tracking. That is the fact that particle filters (PFs) are criticized by the computational requirements proportional to the number of particles N_p needed for reasonable approximations of the posterior distributions. This is problematic because the number of particles N_{p} needed scales with both the state dimension ℓ and the duration of the observation sequence n at the time when the state estimate is needed. In the multi-harmonic tracking case the state dimension ℓ increases proportionally to the number of harmonics N_{h}. In order to overcome this curse of dimensionality it is necessary to improve upon canonical particle filters (PF).

Chapter 3

Problem Definition

3.1 Stochastic Process

A stochastic process is the counterpart to a deterministic process. For a deterministic process, future evolution is completely known and unique given an initial condition of the process. In contrast, future evolution of a stochastic process is not determined or unique due to the indeterminacy of the process. Many paths of evolutions are possible from the known initial point. A stochastic process is also referred to as a *random process.* The randomness of the stochastic process is described by its *probability distribution.* The probability distribution has information on which one among multiple possible paths is more or less probable. For example, a sequence of random variables is a realization of a stochastic process in the case of discrete time. Many different sequences of random variables are possible as realizations of the discrete time stochastic process. Knowing the probability distribution of the stochastic process one can only tell how probable it is for a particular sequence to occur. Many time series can be modeled as stochastic processes such as the human voice, stock market fluctuations, seismic signals, acoustic measurements of machinery vibration, and biomedical data. There are many kinds of stochastic processes such as the Bernoulli process, Poisson process, point process, Gaussian process, and Markov process. They can be used to model coin flipping (Bernoulli process), weather prediction (Markov process), and radioactive decay (Poisson process). In order to choose an appropriate stochastic process prior knowledge on the characteristics of a phenomenon of interest should be used.

3.2 State-Space Methods

A physical system can be mathematically represented as a set of input, output, and states related via a set of models, which can be nonlinear and stochastic. States are unknown variables that represent a system's condition at a given time. For example, in making reinforced glass the temperature between the glass plates cannot be measured directly although it is a critical variable, i.e. state, that represents the condition of the manufacture process. The state-space is an imaginary space consisting of the minimum set of states of a stochastic process that describes a physical system. The output is a sequence of observed measurements of the system. The input is a source that drives the system. In the example of the reinforced glass manufacture process, the output can be a temperature measured around the plates and the input is a white Gaussian noise that models the fluctuation in the temperature. Typically the state-space representation of a system is defined by two equations: a process and a measurement equation. A general state-space representation can be written as,

$$\boldsymbol{x}_{n+1} = f_n\left(\boldsymbol{x}_n, \boldsymbol{u}_n\right), \qquad \boldsymbol{y}_n = h_n\left(\boldsymbol{x}_n, \boldsymbol{v}_n\right) \tag{3.1}$$

where $\boldsymbol{u}_n$ and $\boldsymbol{v}_n$ are inputs, $\boldsymbol{y}_n$ an output, $\boldsymbol{x}_n$ the state of the system at time n, and the functions $f_n(\cdot)$ and $h_n(\cdot)$ are the process and measurement models, respectively. Two functions $f_n(\cdot)$ and $h_n(\cdot)$ are time-indexed, which means that the process and measurement models themselves can be time varying.

The goal of state-space tracking is to estimate the unknown state $\boldsymbol{x}_n$ given a sequence of measurements $\boldsymbol{y}_{0:n}$, where $\boldsymbol{y}_{0:n}$ represents $\{\boldsymbol{y}_0, \ldots, \boldsymbol{y}_n\}$. The state-space representation originated from modern control theory, which utilizes the time-domain mathematical model of a dynamic system. The state-space representation is flexible and versatile since the equations can be nonlinear and stochastic. It has been adopted in signal processing and now is a widely used signal processing paradigm. However, a set of process and measurement equations is not something that is available or given to a user readily. A user has to devise them based on some prior domain knowledge of the system. In other words, a modeling step precedes an actual application step of state-space tracking.

3.3 Markov Process

A *Markov process* is a special case of stochastic processes, which represents the random evolution of a *memoryless system.* A system is called *memoryless* when its future behavior depends only on the current state of the system. The future of a Markov process is independent of the history of the process conditional upon the current state. A stochastic process is a Markov process if it satisfies the condition below,

$$p\left(\boldsymbol{x}_{n+h}|\boldsymbol{x}_s, \forall s \leq n\right) = p\left(\boldsymbol{x}_{n+h}|\boldsymbol{x}_n\right), \forall h > 0 \tag{3.2}$$

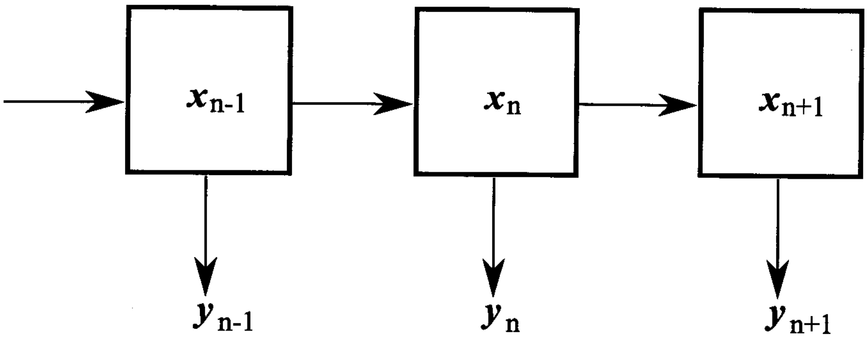


Figure 3.1: Conceptual diagram of Markov processes.

The index space (time scale) and state-space are two main elements that characterize Markov processes. There are four types of Markov processes depending on the continuity of these two elements: discrete time Markov chains, continuous time Markov chains, discrete time Markov processes, and continuous time Markov processes.

Fig. 3.1 illustrates a conceptual diagram of a discrete time Markov process where $\boldsymbol{x}_n$ is the current state and $\boldsymbol{y}_n$ the current measurement. An example of a Markovian process is the Monopoly game whose next state of the board is determined completely by the current state and the next roll of the dice. In contrast, card games are not Markovian processes since each card represents a memory of the past moves. However, Markovian representations are not strictly limited to Markovian processes. In some cases, non-Markovian processes can have Markovian representations. Assuming that $\boldsymbol{x}$ is a non-Markovian process, let us define a new process $\boldsymbol{z}$ that can be expressed as,

$$\boldsymbol{z}(t) = \{\boldsymbol{x}(s) : s \in [\alpha(t), \beta(t)]\}. \tag{3.3}$$

If the new process $\boldsymbol{z}$ has the Markov property, it becomes a Markovian representation of the original process $\boldsymbol{x}$. This is possible because the concept of the present and future can be extended so that they represent intervals in the index space (time scale) instead of particular points in the space. For example, a moving average (MA) time series is a non-Markovian process with a Markovian representation because the future state $\boldsymbol{x}_{n+1}$ depends only on some of current and past states $\boldsymbol{x}_n, \ldots, \boldsymbol{x}_{n-M+1}$, where M is the order of the MA time series. The Markovian representation of this non-Markovian process can be obtained by including the states in the intervals $[n, n-M+1]$ into the "current" state.

3.4 Posterior Distribution and Bayesian Estimation

A posterior distribution is an important term in rhythmicity tracking within the framework of the sum-of-sinusoids model. A posterior distribution, which is denoted as $p(\boldsymbol{x}|\boldsymbol{y})$, is a probability density function that represents a probability distribution of what is known about a random variable $\boldsymbol{x}$ taking the observation $\boldsymbol{y}$ into account. Simply, the posterior distribution $p(\boldsymbol{x}|\boldsymbol{y})$ is a conditional probability distribution of $\boldsymbol{x}$ given $\boldsymbol{y}$. In contrast, a prior distribution $p(\boldsymbol{x})$ is a description of a variable $\boldsymbol{x}$ in the absence of any data. The likelihood function $p(\boldsymbol{y}|\boldsymbol{x})$ is a probability density function that describes a conditional probability of $\boldsymbol{y}$ given $\boldsymbol{x}$. Let us assume that $\boldsymbol{x}$ is a binary character $\{0, 1\}$ to send via a channel and $\boldsymbol{y}$ a received signal. A prior distribution $p(\boldsymbol{x})$ describes which character 0 or 1 is more or less frequently sent. The frequency of characters to send is a predetermined quantity before receiving the actual signal. When a binary character $\{0, 1\}$ is sent via a channel, a receiver receives a signal $\boldsymbol{y}$ that appears to be 0 or 1. The likelihood distribution $p(\boldsymbol{y}|\boldsymbol{x})$ describes what a received signal $\boldsymbol{y}$ is likely to be when the actually sent character is 0 or 1. When the channel is perfect, i.e. noise-free, $\boldsymbol{y}$ is the same as $\boldsymbol{x}$, which means that the receiver does not make any mistake with determining which character was originally sent. However, real channels are usually noisy, and original characters are altered by channel noise. The likelihood distribution $p(\boldsymbol{y}|\boldsymbol{x})$ represents how severe this alteration of original characters is. The posterior distribution $p(\boldsymbol{x}|\boldsymbol{y})$ describes how probable an original character is to be 0 or 1 given a received signal $\boldsymbol{y}$. For example, when $\boldsymbol{y}$ is -0.85, a original character $\boldsymbol{x}$ is probably -1 since $p(\boldsymbol{x} = -1|\boldsymbol{y} = -0.85)$ is greater than $p(\boldsymbol{x} = 1|\boldsymbol{y} = -0.85)$. When $\boldsymbol{y}$ is 1.01, $\boldsymbol{x}$ is probably 1 since $p(\boldsymbol{x} = 1|\boldsymbol{y} = 1.1)$ is greater than $p(\boldsymbol{x} = -1|\boldsymbol{y} = 1.1)$. The posterior distribution $p(\boldsymbol{x}|\boldsymbol{y})$ is used to reason about $\boldsymbol{x}$ given $\boldsymbol{y}$.

The posterior distribution $p(\boldsymbol{x}|\boldsymbol{y})$ can be calculated by multiplying a prior distribution $p(\boldsymbol{x})$ and the likelihood distribution $p(\boldsymbol{y}|\boldsymbol{x})$ via Bayes' theorem as follows,

$$p(\boldsymbol{x}|\boldsymbol{y}) = \frac{p(\boldsymbol{x})p(\boldsymbol{y}|\boldsymbol{x})}{\int_{-\infty}^{\infty} p(\boldsymbol{x})p(\boldsymbol{y}|\boldsymbol{x})\,\mathrm{d}x} \tag{3.4}$$

where the denominator is a normalizing constant. When the random variable $\boldsymbol{x}$ and the observation $\boldsymbol{y}$ are independent, $p(\boldsymbol{y}|\boldsymbol{x})$ becomes $p(\boldsymbol{y})$. Then, the posterior distribution $p(\boldsymbol{x}|\boldsymbol{y})$ is equal to a prior $p(\boldsymbol{x})$. In this case, knowing $\boldsymbol{y}$ is not helpful to estimate $\boldsymbol{x}$. However, when there is some type of relationship between $\boldsymbol{x}$ and $\boldsymbol{y}$, the posterior distribution $p(\boldsymbol{x}|\boldsymbol{y})$ becomes more useful than a prior distribution $p(\boldsymbol{x})$ in reasoning about $\boldsymbol{x}$. These three distributions, i.e. a prior, posterior, and likelihood, are important concepts for understanding estimation methods. There are three popular methods of estimation: method of moments, maximum likelihood (ML) estimation, and Bayesian estimation. The method of moments is usually easy to implement and yields consistent estimators. However, the estimators are often not the best and can be misleading. Generally the maximum likelihood estimation and Bayesian estimation are preferred and more accurate than the method of moments. There are many similarities between those two estimation methods. The most important one is that they utilize the likelihood, prior, and posterior distributions. The main difference between them is that the maximum likelihood estimation relies only on the likelihood distribution $p(\boldsymbol{y}|\boldsymbol{x})$ while Bayesian estimation uses the prior distribution $p(\boldsymbol{x})$ and the likelihood distribution $p(\boldsymbol{y}|\boldsymbol{x})$. The maximum likelihood estimation computes the likelihood and finds a parameter value that maximizes it. It can be written as,

$$\hat{\boldsymbol{x}}_{\mathrm{ML}} = \mathrm{argmax}_{\boldsymbol{x}}\, p(\boldsymbol{y}|\boldsymbol{x}) \tag{3.5}$$

where ML stands for the maximum likelihood and argmax stands for the argument of the maximum, that is the value of the given argument $\boldsymbol{x}$ for which the following function attains its maximum value. There are two common Bayesian estimation methods depending on the choice of estimation quality measures. The first Bayesian estimation is called the minimum mean square error (MMSE) estimation. It minimizes the mean square error (MSE), which can be expressed as,

$$\mathrm{MSE} = \mathrm{E}\left[(\hat{\boldsymbol{x}} - \boldsymbol{x})^2\right] \tag{3.6}$$

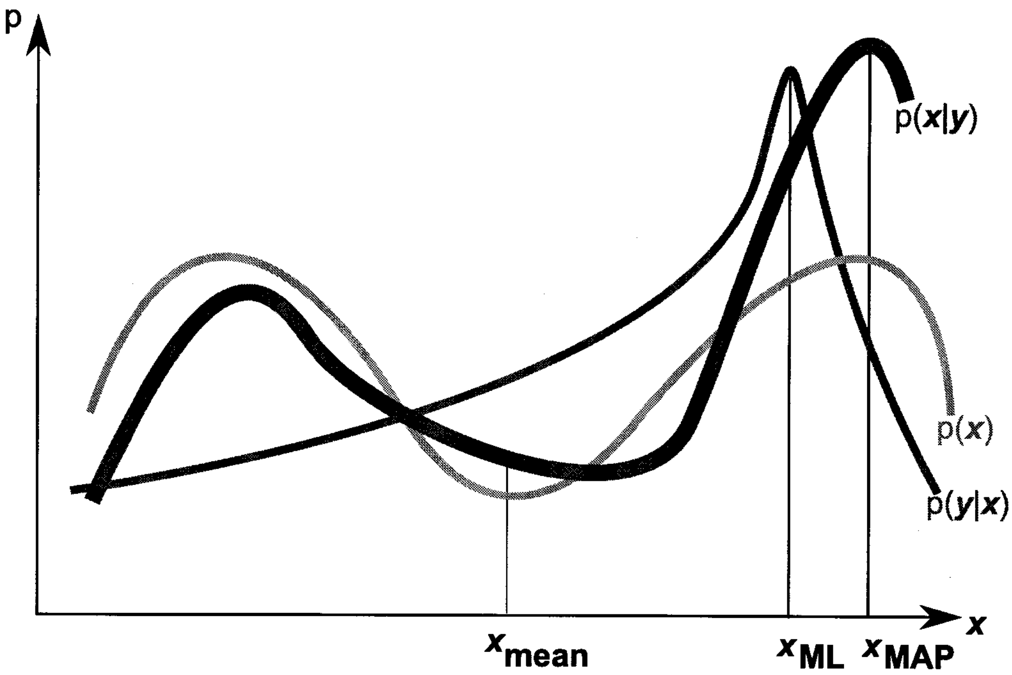


Figure 3.2: Example of the prior, likelihood, and posterior distributions with the mean estimate $\boldsymbol{x}_{\text{mean}}$ of $\boldsymbol{x}$, MAP estimate $\boldsymbol{x}_{\text{MAP}}$, and maximum likelihood estimate $\boldsymbol{x}_{\text{ML}}$.

The MMSE estimator can be expressed as,

$$\begin{aligned}\hat{\boldsymbol{x}}_{\text{MMSE}} &= \mathrm{E}\left[\boldsymbol{x}|\boldsymbol{y}\right] \\ &= \int_{-\infty}^{\infty} \boldsymbol{x}p(\boldsymbol{x}|\boldsymbol{y})\,\mathrm{d}x \end{aligned} \tag{3.7}$$

The second Bayesian estimation is called the maximum *a posteriori* (MAP) estimation. It minimizes the most probable error by searching for $\boldsymbol{x}$ that maximizes the posterior distribution $p(\boldsymbol{x}|\boldsymbol{y})$. It can be expressed as,

$$\begin{aligned}\hat{\boldsymbol{x}}_{\text{MAP}} &= \text{argmax}_{\boldsymbol{x}}\, p(\boldsymbol{x}|\boldsymbol{y}) \\ &= \text{argmax}_{\boldsymbol{x}} \frac{p(\boldsymbol{x})p(\boldsymbol{y}|\boldsymbol{x})}{\int_{-\infty}^{\infty} p(\boldsymbol{x})p(\boldsymbol{y}|\boldsymbol{x})\,\mathrm{d}x} \end{aligned} \tag{3.8}$$

When the posterior distribution can be approximated as Gaussian, the MMSE estimation yields good results. However, if the posterior distribution becomes multimodal, $\hat{\boldsymbol{x}}_{\mathrm{MMSE}}$ can be very misleading. Since both the ML and MAP estimation methods utilize the likelihood distribution $p(\boldsymbol{y}|\boldsymbol{x})$, there is some similarity between those methods. For example, if $\boldsymbol{x}$ is evenly distributed, its prior distribution $p(\boldsymbol{x})$ becomes constant. Then,

$$\begin{aligned}\hat{\boldsymbol{x}}_{\mathrm{MAP}} &= \mathrm{argmax}_{\boldsymbol{x}}\, p(\boldsymbol{x}|\boldsymbol{y}) \\ &= \mathrm{argmax}_{\boldsymbol{x}} \frac{p(\boldsymbol{x})p(\boldsymbol{y}|\boldsymbol{x})}{\int_{-\infty}^{\infty} p(\boldsymbol{x})p(\boldsymbol{y}|\boldsymbol{x})\,\mathrm{d}x} \\ &= \mathrm{argmax}_{\boldsymbol{x}} \frac{c \cdot p(\boldsymbol{y}|\boldsymbol{x})}{\int_{-\infty}^{\infty} c \cdot p(\boldsymbol{y}|\boldsymbol{x})\,\mathrm{d}x} \\ &= \mathrm{argmax}_{\boldsymbol{x}} \frac{p(\boldsymbol{y}|\boldsymbol{x})}{\int_{-\infty}^{\infty} p(\boldsymbol{y}|\boldsymbol{x})\,\mathrm{d}x} \\ &= \hat{\boldsymbol{x}}_{\mathrm{ML}}, \end{aligned} \tag{3.9}$$

that is, the MAP estimation becomes identical to the ML estimation. Fig. 3.2 illustrates an example posterior distribution and the MAP estimation of $\boldsymbol{x}$. The MAP estimation is the most probable value of $\boldsymbol{x}$ given $\boldsymbol{y}$. Therefore, the MAP estimation minimizes the most probable error rather than the expected value of the estimation error.

3.5 Recursive Bayesian Estimation

The posterior distribution of the entire history of the state given all measurements up to current time n can be expressed as $p(\boldsymbol{x}_{0:n}|\boldsymbol{y}_{0:n})$, where $\boldsymbol{x}_{0:n}$ is equivalent to $\{\boldsymbol{x}_0, \ldots, \boldsymbol{x}_n\}$. The marginal posterior distribution $p(\boldsymbol{x}_n|\boldsymbol{y}_{0:n})$ describes the posterior distribution of only the current state given all measurements up to current time n. No matter which distribution needs to be estimated, it would be very inefficient to compute $p(\boldsymbol{x}_{0:n}|\boldsymbol{y}_{0:n})$ without utilizing previously computed distributions such as $p(\boldsymbol{x}_{0:n-1}|\boldsymbol{y}_{0:n-1})$. By adopting the state-space model framework, the posterior distribution $p(\boldsymbol{x}_{0:n}|\boldsymbol{y}_{0:n})$ can be computed by updating $p(\boldsymbol{x}_{0:n-1}|\boldsymbol{y}_{0:n-1})$. Given $p(\boldsymbol{x}_{0:n-1}|\boldsymbol{y}_{0:n-1})$ and the process model in (3.1), the predicted posterior distribution $p(\boldsymbol{x}_{0:n}|\boldsymbol{y}_{0:n-1})$ can be computed since the process model describes how $\boldsymbol{x}_n$ propagates from $\boldsymbol{x}_{n-1}$. As measurement $\boldsymbol{y}_n$ becomes available, $p(\boldsymbol{x}_{0:n}|\boldsymbol{y}_{0:n})$ can be updated from $p(\boldsymbol{x}_{0:n}|\boldsymbol{y}_{0:n-1})$ since the measurement model in (3.1) describes how $\boldsymbol{y}_n$ is related to $\boldsymbol{x}_n$. By alternating these prediction and measurement updates one can compute the posterior distribution recursively.

Recursive Bayesian estimation is a Bayesian approach to estimating an unknown posterior distribution recursively over time using incoming measurements and a state-space model. Let us consider a filtering problem where the marginal posterior distribution $p(\boldsymbol{x}_n|\boldsymbol{y}_{0:n})$ is of interest. As explained above, this marginal posterior distribution can be computed efficiently by updating $p(\boldsymbol{x}_{n-1}|\boldsymbol{y}_{0:n-1})$. The update recursions can be written as follows,

$$\begin{aligned} p(\boldsymbol{x}_n|\boldsymbol{y}_{0:n}) &= \frac{p(\boldsymbol{y}_n|\boldsymbol{x}_n, \boldsymbol{y}_{0:n-1})p(\boldsymbol{x}_n|\boldsymbol{y}_{0:n-1})}{p(\boldsymbol{y}_n|\boldsymbol{y}_{0:n-1})} \\ &= \frac{p(\boldsymbol{y}_n|\boldsymbol{x}_n)p(\boldsymbol{x}_n|\boldsymbol{y}_{0:n-1})}{p(\boldsymbol{y}_n|\boldsymbol{y}_{0:n-1})} \end{aligned} \tag{3.10}$$

$$\begin{aligned} p(\boldsymbol{x}_n|\boldsymbol{y}_{0:n-1}) &= \int p(\boldsymbol{x}_n, \boldsymbol{x}_{n-1}|\boldsymbol{y}_{0:n-1})\,\mathrm{d}\boldsymbol{x}_{n-1} \\ &= \int p(\boldsymbol{x}_n|\boldsymbol{x}_{n-1}, \boldsymbol{y}_{0:n-1})p(\boldsymbol{x}_{n-1}|\boldsymbol{y}_{0:n-1})\,\mathrm{d}\boldsymbol{x}_{n-1} \\ &= \int p(\boldsymbol{x}_n|\boldsymbol{x}_{n-1})p(\boldsymbol{x}_{n-1}|\boldsymbol{y}_{0:n-1})\,\mathrm{d}\boldsymbol{x}_{n-1} \end{aligned} \tag{3.11}$$

where (3.10) is called a *measurement update* and (3.11) a *prediction update*. Mathematically the recursions are an elegant solution. But, in most cases, the prediction step (integral) is impossible or very difficult to evaluate analytically. The following section discusses an alternative way.

3.6 Monte Carlo Methods

Monte Carlo methods (MCMs) refer to a class of computational algorithms that rely on repeated random sampling to compute integrals that are impossible or difficult to evaluate analytically. Let us consider a distribution $g(\boldsymbol{x})$, whose integral is difficult to evaluate analytically. Its integral can be written as,

$$I = \int g(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x} \tag{3.12}$$

When $\boldsymbol{x}$ lies in a high dimension, evaluating the integral becomes more challenging. MCMs first factorize $g(\boldsymbol{x})$ as follows,

$$g(\boldsymbol{x}) = f(\boldsymbol{x})\pi(\boldsymbol{x}) \tag{3.13}$$

where $\pi(\boldsymbol{x})$ is a probability density function (PDF), that is,

$$\begin{aligned} \pi(\boldsymbol{x}) &\geq 0 \\ \int \pi(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x} &= 1. \end{aligned} \tag{3.14}$$

If $N \gg 1$ independent samples are drawn from $\pi(\boldsymbol{x})$, the integral I can be estimated as a sum of those samples. It can be expressed as,

$$\begin{aligned} I &= \int g(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x} \\ &= \int f(\boldsymbol{x})\pi(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x} \\ &= \mathrm{E}\left[f(\boldsymbol{x})\right] \end{aligned} \tag{3.15}$$

$$\hat{I}_N = \frac{1}{N}\sum_{i=1}^{N} f(\boldsymbol{x}^{(i)}), \tag{3.16}$$

where $\hat{I}_N$ is an unbiased estimate of I, which almost surely converges to I as N increases. The error variance does not increase as the dimension of $\boldsymbol{x}$ increases. However, drawing samples from the PDF $\pi(\boldsymbol{x})$ may not be always easy because a closed-form mathematical expression for $\pi(\boldsymbol{x})$ may not exist and $\pi(\boldsymbol{x})$ can be evaluated only numerically for specific values of $\boldsymbol{x}$. Therefore, it is necessary to define another probability density function (PDF) $q(\boldsymbol{x})$, from which random samples can be drawn. Any arbitrary probability density function (PDF) can be the importance density $q(\boldsymbol{x})$ as long as its support is larger than the PDF $\pi(\boldsymbol{x})$'s and independent random samples can be drawn from it. That is,

$$\begin{aligned} I &= \int f(\boldsymbol{x})\pi(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x} \\ &= \int f(\boldsymbol{x})\frac{\pi(\boldsymbol{x})}{q(\boldsymbol{x})}q(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x} \end{aligned} \tag{3.17}$$

$$\begin{aligned} \hat{I}_N &= \frac{1}{N}\sum_{i=1}^{N} f(\boldsymbol{x}^{(i)})\frac{\pi(\boldsymbol{x}^{(i)})}{q(\boldsymbol{x}^{(i)})} \\ &= \frac{1}{N}\sum_{i=1}^{N} f(\boldsymbol{x}^{(i)})\hat{w}(\boldsymbol{x}^{(i)}) \end{aligned} \tag{3.18}$$

where $\hat{w}(\boldsymbol{x}^{(i)})$ is an unnormalized weight of the i^{th} sample. Since the sum of the unnormalized weights is N,

$$\sum_{i=1}^{N} \hat{w}(\boldsymbol{x}^{(i)}) = N \tag{3.19}$$

the estimate $\hat{I}_N$ can be rewritten as follows,

$$\begin{aligned}\hat{I}_N &= \frac{1}{N}\sum_{i=1}^{N} f(\boldsymbol{x}^{(i)})\hat{w}(\boldsymbol{x}^{(i)}) \\ &= \sum_{i=1}^{N} f(\boldsymbol{x}^{(i)})\frac{\hat{w}(\boldsymbol{x}^{(i)})}{N} \\ &= \sum_{i=1}^{N} f(\boldsymbol{x}^{(i)})\frac{\hat{w}(\boldsymbol{x}^{(i)})}{\sum_{i=1}^{N}\hat{w}(\boldsymbol{x}^{(i)})} \\ &= \sum_{i=1}^{N} f(\boldsymbol{x}^{(i)})w(\boldsymbol{x}^{(i)}) \end{aligned} \tag{3.20}$$

where $w(\boldsymbol{x}^{(i)})$ is a normalized weight of the i^{th} sample. The PDF $q(\boldsymbol{x})$ is called an *importance density* and the sampling scheme described above is referred to as *importance sampling.*

3.7 Sequential Monte Carlo Methods

Sequential Monte Carlo methods (SMCMs) are better known as *particle filters,* which are a technique to estimate a sequence of unknown variables *recursively* over time. SMCMs are an alternative to the extended or unscented Kalman filter (EKF/UKF) with the advantage of approaching the optimal Bayesian estimate when the number of samples is large enough. Also, SMCMs have considerable merit over the EKF/UKF if the posterior distribution of interest has several modes of comparable amplitude [84]. Many research groups have demonstrated the suitability and versatility of SMCMs for various applications such as positioning and navigation [85], tracking [86–88], blind equalization [83], fault detection [89], computer vision [90], and speech recognition [91].

SMCMs assume that the state $\boldsymbol{x}_n$ and measurement $\boldsymbol{y}_n$ can be modeled as the first-order Markov process as shown in (3.1). Therefore, $p(\boldsymbol{y}_n|\boldsymbol{x}_n)$ and $p(\boldsymbol{x}_{n+1}|\boldsymbol{x}_n)$ can be computed as measurement $\boldsymbol{y}_n$ becomes available. However, SMCMs cannot be directly applied to this state-space model because there is no means to draw independent random samples directly from the posterior distribution $p\left(\boldsymbol{x}_{0:n}|\boldsymbol{y}_{0:n}\right)$. The importance sampling scheme introduced in Section 3.6 can be adopted to enable random sampling. By weighting the random samples drawn from the importance density $q(\boldsymbol{x}_{0:n}|\boldsymbol{y}_{0:n})$ appropriately, the posterior distribution $p\left(\boldsymbol{x}_{0:n}|\boldsymbol{y}_{0:n}\right)$ can be represented as a weighted combination of point masses, i.e. $f(\boldsymbol{x}^{(i)})$ in (3.18). Then, the expected value of any statistic of the state variable $\boldsymbol{x}_{0:n}$ can be estimated as follows,

$$\begin{aligned}
\mathrm{E}\left[g(\boldsymbol{x}_{0:n})\right] &= \int g\left(\boldsymbol{x}_{0:n}\right) p\left(\boldsymbol{x}_{0:n}|\boldsymbol{y}_{0:n}\right) \mathrm{d}\boldsymbol{x}_{0:n} \\
&\approx \sum_{i=1}^{N_\mathrm{p}} \frac{\tilde{\boldsymbol{w}}_n^{(i)}}{\sum_{j=1}^{N_\mathrm{p}} \tilde{\boldsymbol{w}}_n^{(j)}} g\left(\tilde{\boldsymbol{x}}_{0:n}^{(i)}\right) \\
&= \sum_{i=1}^{N_\mathrm{p}} \boldsymbol{w}_n^{(i)} g\left(\tilde{\boldsymbol{x}}_{0:n}^{(i)}\right) \qquad (3.21)
\end{aligned}$$

where E is the expectation operator, N_p the number of random samples, and $\boldsymbol{x}_{0:n}^{(i)}$ the i^th random state trajectory, and $\boldsymbol{w}_n^{(i)}$ the i^th *importance weight* of the corresponding state trajectory. The random state trajectories are often referred to as *particles*. Here, a *state* refers to a set of random samples drawn from the importance density, corresponding state variables, and an estimate of output $\hat{\boldsymbol{y}}_n$. Then, a *state trajectory*, i.e. *particle*, represents a history of state evolution through time. In other words, one particle $\boldsymbol{x}_{0:n}^{(i)}$ represents the entire record of how the state has evolved from the beginning $n = 0$.

SMCMs can be seen as an implementation of importance sampling by updating the weights $w(\boldsymbol{x}_n^{(i)})$ in (3.21) to compute the estimate of the posterior distribution $\hat{p}(\boldsymbol{x}_{0:n}|\boldsymbol{y}_{0:n})$ upon the arrival of measurement $\boldsymbol{y}_n$ given $\hat{p}(\boldsymbol{x}_{0:n-1}|\boldsymbol{y}_{0:n-1})$. As mentioned earlier, any arbitrary probability density function (PDF) can be the importance density $q\left(\boldsymbol{x}_{0:n}|\boldsymbol{y}_{0:n}\right)$ as long as its support is larger than the posterior distribution $p\left(\boldsymbol{x}_{0:n}|\boldsymbol{y}_{0:n}\right)$ and independent random samples can be drawn from it. However, in SMCMs, the importance density $q\left(\boldsymbol{x}_{0:n}|\boldsymbol{y}_{0:n}\right)$ needs to satisfy one more condition as follows,

$$\begin{aligned} q(x_{0:n}|y_{0:n}) &= q(\boldsymbol{x}_n|\boldsymbol{x}_{0:n-1}, \boldsymbol{y}_{0:n})q(\boldsymbol{x}_{0:n-1}|\boldsymbol{y}_{0:n-1}) \\ &= q(\boldsymbol{x}_0|\boldsymbol{y}_0)\prod_{k=1}^{n} q(\boldsymbol{x}_k|\boldsymbol{x}_{0:k-1}, \boldsymbol{y}_{0:k}) \end{aligned} \tag{3.22}$$

Then, the weight $w(\boldsymbol{x}_n^{(i)})$ can be updated *sequentially* as follows,

$$\begin{aligned} w(\boldsymbol{x}_n^{(i)}) &\propto \frac{p(\boldsymbol{x}_{0:n}|\boldsymbol{y}_{0:n})}{q(\boldsymbol{x}_{0:n}|\boldsymbol{y}_{0:n})} \\ &= \frac{p(\boldsymbol{y}_n|\boldsymbol{x}_n^{(i)})p(\boldsymbol{x}_n^{(i)}|\boldsymbol{x}_{n-1}^{(i)})p(\boldsymbol{x}_{0:n-1}^{(i)}|\boldsymbol{y}_{0:n-1})}{q(\boldsymbol{x}_n^{(i)}|\boldsymbol{x}_{0:n-1}^{(i)}, \boldsymbol{y}_{0:n})q(\boldsymbol{x}_{0:n-1}^{(i)}|\boldsymbol{y}_{0:n-1})} \\ &= \frac{p(\boldsymbol{y}_n|\boldsymbol{x}_n^{(i)})p(\boldsymbol{x}_n^{(i)}|\boldsymbol{x}_{n-1}^{(i)})}{q(\boldsymbol{x}_n^{(i)}|\boldsymbol{x}_{0:n-1}^{(i)}, \boldsymbol{y}_{0:n})} w(\boldsymbol{x}_{n-1}^{(i)}). \end{aligned} \tag{3.23}$$

Since $p(\boldsymbol{y}_n|\boldsymbol{x}_n^{(i)})$ and $p(\boldsymbol{x}_n^{(i)}|\boldsymbol{x}_{n-1}^{(i)})$ can be known upon the arrival of measurement $\boldsymbol{y}_n$ using the state-space model in (3.1), the current weight $w(\boldsymbol{x}_n^{(i)})$ can be updated readily from the previous weight $w(\boldsymbol{x}_{n-1}^{(i)})$ after computing the importance density $q(\boldsymbol{x}_n|\boldsymbol{x}_{0:n-1}, \boldsymbol{y}_{0:n})$. The ideal importance density is the posterior distribution itself, i.e. $p(\boldsymbol{x}_n|\boldsymbol{x}_{0:n-1}, \boldsymbol{y}_{0:n})$. The posterior distribution is *ideal* since using it as the importance density yields the minimum variance estimate of $\mathrm{E}\left[g(\boldsymbol{x}_{0:n})\right]$. However, the analytical evaluation of the ideal importance density is not possible since it is the posterior distribution itself that needs to be estimated. Although the optimal importance density can be obtained for a special case, usually it can be only approximated [92]. The simplest approximation is to use the prior distribution $p(\boldsymbol{x}_n|\boldsymbol{x}_{n-1})$ as the importance density $q(\boldsymbol{x}_n|\boldsymbol{x}_{0:n-1}, \boldsymbol{y}_{0:n})$. Then, the weight recursion is simplified as,

$$w(\boldsymbol{x}_n^{(i)}) \propto p(\boldsymbol{y}_n|\boldsymbol{x}_n^{(i)})w(\boldsymbol{x}_{n-1}^{(i)}). \tag{3.24}$$

Particle filters that are implemented hereafter are based on this approximation of the importance density.

3.8 Rhythmicity Tracking as Posterior Distribution Estimation

Quasi-periodic signals can be modeled as a sum of time-varying sinusoids, which represent rhythmical components, i.e. partials, of the signals. The goal of rhythmicity tracking in quasi-periodic signals, then, is to estimate the frequencies, phases, and amplitudes (parameters in (1.12)) of those partials sequentially. This sum-of-sinusoids model of a quasi-periodic signal can be integrated into a *state-space representation* of a physical system that generates the signal. In rhythmicity tracking of quasi-periodic signals the sum-of-sinusoids model is the measurement equation where $\boldsymbol{y}_n$ is a contaminated signal sample at time n, $\boldsymbol{x}_n$ a vector of frequencies and sinusoid coefficients of the rhythmical components, and $\boldsymbol{v}_n$ additive measurement noise. The main goal in the state-space model approach to rhythmicity tracking is to estimate the model parameters as accurately as possible, where the measure of "accuracy" may vary with application. However, accurate estimation of the model parameters does not point out the fundamental problem that we attempt to solve in rhythmicity tracking. It is critical to recognize this fundamental problem to properly address important issues in rhythmicity tracking. Rhythmicity tracking is essentially continuous estimation of the posterior distribution of the frequencies and amplitudes of the rhythmical components, i.e. partials, in the signal. Traditional frequency tracking techniques assume that the posterior distribution is Gaussian and choose the mean value of the distribution as the estimate of frequencies, phases, and amplitudes. Depending on the number of rhythmical components and their relationship, however, the Gaussian assumption may not hold and the posterior distribution may be *multi-modal.* In that case, the mean is not a good representative of the distribution, and more complete descriptions are necessary to represent its important features. For example, the mean of the posterior distribution of the fundamental frequency given a sequence of signal samples, that is,

$$\begin{aligned}\hat{\boldsymbol{f}}_n &= \mathrm{E}\left[\boldsymbol{f}_n\right] \\ &= \int \boldsymbol{f}_n \, p(\boldsymbol{f}_n|\boldsymbol{y}_{0:n}) \, \mathrm{d}\boldsymbol{f} \qquad (3.25)\end{aligned}$$

may be located between modes that correspond to *probable* frequency values. Therefore, a reasonable estimate of the fundamental frequency is the frequency value that corresponds to the tallest mode of the distribution, which minimizes the *most probable error*. This estimate is the maximum a *posteriori* (MAP) estimate, which

$$\hat{\boldsymbol{f}}_n = \text{argmax}_{\boldsymbol{f}}\, p(\boldsymbol{f}_n|\boldsymbol{y}_{0:n}) \tag{3.26}$$

Given the frequency value $\boldsymbol{f}_n$ the coefficients $(\boldsymbol{a}_{1,1,n}, \ldots, \boldsymbol{a}_{1,\mathrm{N}_h,n}, \boldsymbol{a}_{2,1,n}, \ldots, \boldsymbol{a}_{2,\mathrm{N}_h,n})$ have a simple linear relationship with the signal. Therefore, the posterior distribution of partials' phases and amplitudes can be modeled as a Gaussian distribution and the *optimal* solution can be obtained using the Kalman filter technique. However, the frequency has a nonlinear relationship with the signal through the cosine and sine functions. Especially when there are multiple harmonically-related partials, the posterior distribution of the fundamental frequency $p(\boldsymbol{f}_n|\boldsymbol{y}_{0:n})$ becomes non-Gaussian and multi-modal. In that case, approximating the posterior distribution as a Gaussian is not adequate [85]. By adopting sequential Monte Carlo methods (SMCM), i.e. particle filters, the multi-modality of the posterior distribution of the fundamental frequency can be handled properly.

The subsequent sections demonstrate the multi-modality of the marginal posterior distribution of the fundamental frequency in the multi-harmonic signal, discuss weaknesses of the canonical particle filtering technique, and describe the statistical signal models for applications included in the dissertation.

3.9 Multi-modal Posterior Example

For the sake of illustrating the multiple modes in the posterior distribution, let us consider a much simpler case in this section that permits us to calculate the posterior distribution. In general it is difficult to solve for the posterior distribution exactly even if a state space model of the process is known. However, if we use the simplifying assumptions that (1) the coefficients and fundamental frequency have uniform prior distributions and do not change over time and (2) the measurement noise is Gaussian, then we can solve for the posterior distribution of the fundamental frequency with a least squares approach. In this case the observation model is,

$$\boldsymbol{y}_n = \bar{\boldsymbol{y}} + \sum_{k=1}^{N_\mathrm{h}} \boldsymbol{a}_{1,k} \cos\left(k\boldsymbol{\omega} n\right) + \boldsymbol{a}_{2,k} \sin\left(k\boldsymbol{\omega} n\right) + \boldsymbol{v}_n \tag{3.27}$$

where N_h is the number of harmonics (assumed known), $\boldsymbol{\omega}$ is the angular fundamental frequency, $\bar{\boldsymbol{y}}$ is the slowly changing signal mean, $\boldsymbol{a}_{1,k}$ and $\boldsymbol{a}_{2,k}$ are the fixed valued sinusoidal coefficients, and $\boldsymbol{v}_n$ is white Gaussian noise with variance R_v. If the fundamental frequency and measurement noise are known, this model is a linear function of the signal mean and sinusoidal coefficients.

We can collect the unknown parameters into a vector, which can be written as,

$$\boldsymbol{x} = \begin{bmatrix} \bar{\boldsymbol{y}} & \boldsymbol{a}_1 & \ldots \boldsymbol{a}_k & \boldsymbol{b}_1 & \ldots \boldsymbol{b}_k & \boldsymbol{\omega} \end{bmatrix}. \tag{3.28}$$

Then, the posterior distribution is given by

$$\begin{aligned} p(\boldsymbol{x}|\boldsymbol{y}_{0:n}; r) &= \frac{p(\boldsymbol{y}_{0:n}|\boldsymbol{x}; r)p(\boldsymbol{x})}{p(\boldsymbol{y}_{0:n}; r)} \\ &\propto p(\boldsymbol{y}_{0:n}|\boldsymbol{x}; r) \end{aligned}$$

where $\boldsymbol{y}_{0:n} \triangleq (\boldsymbol{y}_0, \ldots, \boldsymbol{y}_n)$. Since we have assume a uniform prior, $p(\boldsymbol{x}) = c$ for some constant c, the posterior distribution is the same as the likelihood function.

For a specified value of the fundamental frequency, the remaining parameters can be estimated by linear least squares, or, equivalently, maximum likelihood. An unbiased estimate of the measurement noise variance R_v is then given by

$$\hat{r} = \frac{1}{N_{\mathrm{T}} - 1 - 2N_{\mathrm{h}}} \sum_{n=1}^{N_{\mathrm{T}}} (\boldsymbol{y}_n - \hat{\boldsymbol{y}}_n)^2 \tag{3.29}$$

where N_{T} is the number of samples. Finally, the posterior distribution of the angular frequency $\boldsymbol{\omega}$ can be obtained as follows

$$p(\boldsymbol{\omega}|\boldsymbol{y}_{0:n}) \approx \max_{\bar{\boldsymbol{y}},\boldsymbol{a}_k,\boldsymbol{b}_k} p(\boldsymbol{y}_{0:n}|\boldsymbol{x};\hat{r}). \tag{3.30}$$

Fig. 3.3 shows the estimated posterior distribution evaluated as a function of frequency for a signal with $N_{\mathrm{T}} = 500$ samples, $N_{\mathrm{h}} = 5$ partials (4 harmonic partials), the true fundamental frequency 2 Hz, the amplitudes $\boldsymbol{a}_{1,2k+1}$ and $\boldsymbol{a}_{2,2k+1}$ equal 1, $\boldsymbol{a}_{1,2k}$ and $\boldsymbol{a}_{2,2k} = 10$ equal 10, and a signal-to-noise ratio (SNR) of 3 dB. The amplitudes of the even-numbered partials $\boldsymbol{a}_{1,2k}$ and $\boldsymbol{a}_{2,2k}$ were chosen to be 10 times greater than those of the odd-numbered partials $\boldsymbol{a}_{1,2k+1}$ and $\boldsymbol{a}_{2,2k+1}$ to enhance the modes that correspond to the sub-harmonics of the fundamental frequency. Fig. 3.3 demonstrates that the marginal posterior distribution of the fundamental frequency of the multi-harmonic signal is truly multi-modal.

3.10 Weaknesses of Canonical Particle Filtering

The approximation (3.21) converges to the true value almost surely as the number of particles becomes large enough. There are, however, three major drawbacks in using the canonical PFs: two curses of dimensionality and using the mean value

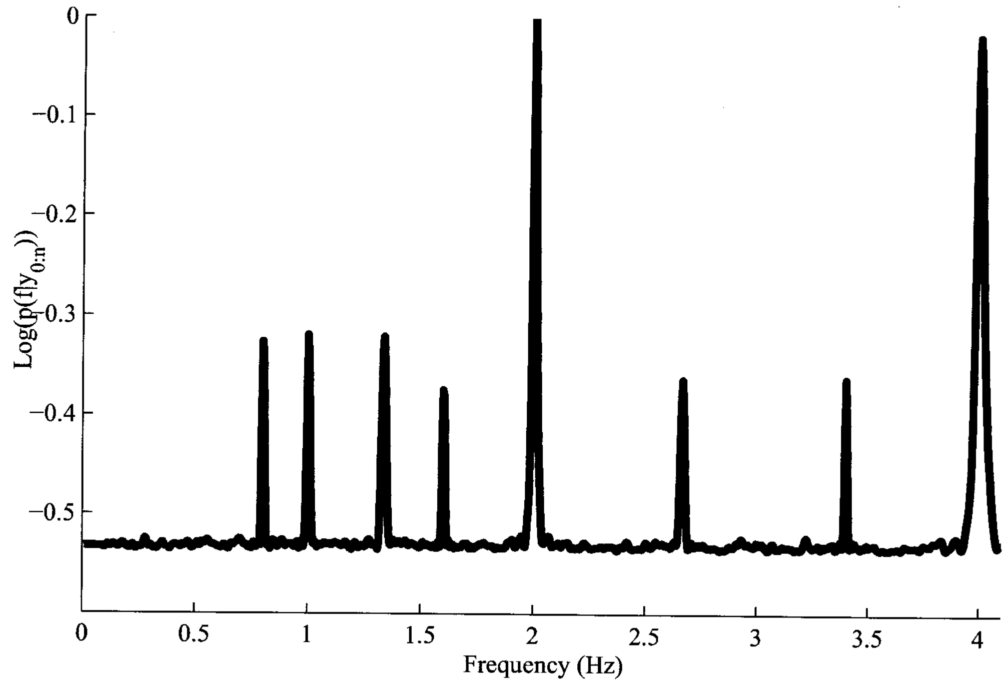


Figure 3.3: Logarithmic posterior PDF of the fundamental frequency of a synthetic multi-harmonic signal with 5 harmonic partials whose true fundamental frequency is 2 Hz.

to represent multi-modal posterior distributions. The two curses of dimensionality are due to the fact that the number of particles N_{p}, for reasonable approximations of posterior distribution, scales with both the state dimension ℓ and duration of the observation sequence n at the time the state estimate is needed. The first curse of dimensionality related with the state dimension ℓ can be managed by adopting the Rao-Blackwellization of canonical PFs when the state-space model meets certain conditions. The Rao-Blackwell theorem shows how to improve upon any given estimator by calculating a conditional expected value integrating out an ancillary statistic [93]. In particle filtering, the conditional expected value is the nonlinear portion of the state-space while the ancillary statistic is the linear portion of it. Casella *et al.* proposed an improvement for sampling schemes such as Accept-

Reject and Metropolis algorithms by using the entire set of simulated random variables, which is the result of applying the Rao-Blackwellization Theorem [94]. A few years later, Doucet *et al.* developed a Rao-Blackwellization based particle filtering method that takes advantage of the analytic structure present in a specific class of state-space models [92].

Suppose that we can partition the state vector, $\boldsymbol{x}_n$, into the linear $\boldsymbol{x}_n^{\mathrm{L}}$ and nonlinear $\boldsymbol{x}_n^{\mathrm{N}}$ portions, where the system is linear when the nonlinear portion of the state, $\boldsymbol{x}_n^{\mathrm{N}}$ is known. Then, the state-space models that belong to this class can be expressed as follows,

$$\boldsymbol{x}_{n+1}^{\mathrm{L}} = F_n(\boldsymbol{x}_n^{\mathrm{N}})\boldsymbol{x}_n^{\mathrm{L}} + \boldsymbol{u}_n^{\mathrm{L}} \tag{3.31}$$

$$\boldsymbol{x}_{n+1}^{\mathrm{N}} = f_n(\boldsymbol{x}_n^{\mathrm{N}}, \boldsymbol{u}_n^{\mathrm{N}}) \tag{3.32}$$

$$\boldsymbol{y}_n = H_n(\boldsymbol{x}_n^{\mathrm{N}})\boldsymbol{x}^{\mathrm{L}} + \boldsymbol{v}_n \tag{3.33}$$

where $F_n(\boldsymbol{x}_n^{\mathrm{N}})$ is a time-varying state-transition matrix that is a function of $\boldsymbol{x}_n^{\mathrm{N}}$ and $H_n(\boldsymbol{x}_n^{\mathrm{N}})$ is an observation matrix that is also a function of $\boldsymbol{x}_n^{\mathrm{N}}$. In this case the linear Kalman filter recursions can be used to *optimally* estimate the linear portion of the state for each particle, $\boldsymbol{x}_n^{\mathrm{L},(i)}$. This can dramatically reduce the number of particles needed to represent the remaining nonlinear portion of the posterior state distribution, $p(\boldsymbol{x}_{0:n}^{\mathrm{N}}|\boldsymbol{y}_{0:n})$, which is in a smaller dimensional space. If the linear process noise $\boldsymbol{u}_n^{\mathrm{L}}$ and the measurement noise $\boldsymbol{v}_n$ are Gaussian, then the conditional linear posterior distribution $p(\boldsymbol{x}_{0:n}^{\mathrm{L}}|\boldsymbol{y}_{0:n}, \boldsymbol{x}_{0:n}^{\mathrm{N}})$ is also Gaussian and the total posterior distribution is given by

$$p(\boldsymbol{x}_{0:n}|\boldsymbol{y}_{0:n}) = p(\boldsymbol{x}_{0:n}^{\mathrm{L}}|\boldsymbol{y}_{0:n}, \boldsymbol{x}_{0:n}^{\mathrm{N}})p(\boldsymbol{x}_{0:n}^{\mathrm{N}}|\boldsymbol{y}_{0:n}) \tag{3.34}$$

This type of particle filtering is called the *Rao-Blackwellized Particle Filter* or sometimes the *marginalized particle filter* (MPF) since the nonlinear portion of the state space is estimated from the marginal distribution. The MPF leverages the traditional Kalman filter to optimally estimate the linear portion of the state and reduces the memory and computational requirements by reducing the dimension of the state vector that is estimated with a particle filter. Gustafsson *et al.* have surveyed the state of the art in theory and practice of the MPF [95].

This, however, does not solve the second curse of dimensionality: the number of particles required for accurate estimation of an expected value of the posterior distribution still increases exponentially with the duration of the observations n at the time the estimate is needed. The relationship between the duration of the observation sequence n and the number of particles needed was not clearly identified until [96]. Gordon *et al.* pointed out that all of the importance weights $w_n^{(i)}$ become very small after a few iterations. This results in a large population of particles that contribute very little to the estimate in (2.22). This phenomenon is known as, *sample degeneracy.* Resampling schemes have been proposed to address this problem which essentially use a *bootstrap filter* to sample the posterior distribution with replacement. This approach is acknowledged as the first successful employment of SMCM for nonlinear filtering [97]. However, this approach generates many duplicate particles of the most probable state trajectories and results in less *sample diversity*, or coverage of the set of all possible state trajectories. This is called *sample impoverishment.* When the true posterior distribution changes abruptly or develops a new prominent mode, PFs may take long time or fail to lock on to the new prominent mode.

The third problem is that estimates of expected values (2.22) do not always provide acceptable measures of multi-modal posterior distributions. For example, when the posterior distribution is multi-modal, the mean of the distribution may be at an improbable state trajectory that is located between modes. In these cases an alternative, such as the median or most probable state trajectories may be preferred estimates.

The last two problems can be solved with a variation of canonical particle filters developed for maximum *a posteriori* (MAP) estimation. These produce an estimate of the state that approximately maximizes the posterior distribution

$$\hat{\boldsymbol{x}}_{0:n} \approx \mathrm{argmax}_{\boldsymbol{x}_{0:n}}\, p(\boldsymbol{x}_{0:n}|\boldsymbol{y}_{0:n}). \tag{3.35}$$

This is suitable for multi-modal distributions because it essentially selects the state estimate as the most prominent mode, which minimizes the most probable error. It is, however, not trivial to obtain a MAP estimate from typical particle filters that represent the posterior as weighted point masses, i.e. $f(\boldsymbol{x}^{(i)})$ in (3.18). The estimate is not equivalent to choosing the state trajectory with the largest importance weight, i.e. $\max w_n^{(i)}$, because the weights are dependent on both the target distribution $p(\boldsymbol{x}_{0:n}|\boldsymbol{y}_{0:n})$ and the importance distribution $q(\boldsymbol{x}_{0:n}|\boldsymbol{y}_{0:n})$ that the particles are drawn from [97]. However, MAP estimates can be obtained by simply selecting the state trajectory with the largest posterior probability

$$\hat{\boldsymbol{x}}_{0:n} = \mathrm{argmax}_i\, p(\boldsymbol{x}_{0:n}^{(i)}|\boldsymbol{y}_{0:n}) \tag{3.36}$$

The second curse of dimensionality can be addressed by combining this approach with the Viterbi algorithm [98]. Godsill *et al.* introduced maximum a posteriori sequence estimation using particle filters for the first time [97, 99]. Although a computational cost increases up to N_{p}^2 per time step as compared to N_{p} for the canonical particle filter, this approach increases the total number of candidate state trajectories exponentially with time n. Since the MAP state trajectory may sequentially choose states from any particle in the population, even particles that are in unlikely regions of state space may, in an instant, become relevant and contribute to the MAP sequence. This solves the sample degeneracy problem by alleviating the need for particle resampling and the sample impoverishment problem. Moreover, MAP estimation is more appropriate to represent the most probable mode of multi-modal distributions than mean estimation that is typically used with the canonical PF.

MAP estimation combined with the marginalized particle filter can address all major drawbacks of the canonical particle filter appropriately. Although Hutter *et al.* suggested this approach [100], the recursions have never been derived or implemented. My dissertation will describe a method for combining the advantages of marginalized particle filters and MAP particle filters based on the Viterbi algorithm and to demonstrate its versatility in various applications that the traditional extended Kalman filter and canonical particle filter are not suitable for.

The dissertation includes tracking problems in four types of quasi-periodic signals: single rhythmical component signal, single harmonic set signal, amplitude-modulated harmonic set signal, and multiple harmonic set signal. Since we take the state-space model approach to these tracking problems, building the statistical signal models for them is the first step. The next chapter describes four state-space models for the tracking problems and explains how each model can be used in the framework of particle filtering to solve practical tracking problems.

Chapter 4

Contributions

This chapter summarizes the main contributions of the dissertation with specific applications. There are five main contributions: new particle filtering technique development, single rhythmical component tracking, single harmonic set tracking, amplitude modulated harmonic set tracking, and multiple harmonic set tracking. The first contribution is to develop a new particle filtering algorithm that combines the advantages of the marginalized particle filter (MPF) and maximum *a posteriori* particle filter (MAP-PF) based on the Viterbi algorithm. The second contribution is to track the instantaneous frequency of a single rhythmical component signal using the proposed particle filter. Its application is tracking tremor frequency exhibited in binary spike trains. The third contribution is to track the fundamental frequency of a single harmonic set signal. Electrocardiogram (ECG) is a typical example of a single harmonic set signal. The new particle filter will be used to track the heart rate (fundamental frequency) in ECGs. The fourth contribution is to track the degree of amplitude modulation in an amplitude modulated harmonic set signal, where one harmonic set is amplitude-modulated by the other. Tracking the pulse pressure variation (PPV) index in arterial blood pressure (ABP) recorded from subjects under full respiratory support is an excellent application of amplitude modulated harmonic set tracking. The last contribution is to track the fundamental frequencies and the degree of amplitude modulation in a multiple harmonic set signal. Its application is tracking the respiratory rate, heart rate, and PPV index in ABP signals recorded from spontaneously breathing subjects.

4.1 New Particle Filtering Technique

A newly proposed particle filter is called the maximum *a posteriori* adaptive marginalized particle filter (MAM-PF). This is a hybrid particle filtering method that leverages the advantages of the maximum *a posteriori* particle filter (MAP-PF) and marginalized particle filter (MPF) algorithms. As mentioned previously, the canonical PFs suffer from two curses of dimensionality due to the fact that the number of particles N_p for reasonable approximations of *a posteriori* distribution scales with both the state dimension ℓ and duration of the observation sequence n. The MPF portion of the new algorithm handles the state dimension ℓ efficiently by partitioning the state into the linear and nonlinear portions as shown in (3.31)-(3.33). The sequential Monte Carlo method (SMCM) is applied to only the nonlinear portion of the state while the linear portion is *optimally* estimated by adopting the linear Kalman filter recursions. The MAP-PF portion of the new algorithm addresses the curse of dimensionality related to the duration of the observation sequence n. The MAP-PF portion of the new algorithm permits the particles to densely cover the nonlinear portion of the state space. This prevents the sample impoverishment problem that would normally be caused by resampling. The MAP-PF portion of the new algorithm also addresses the issue of how to represent of the multi-modal posterior distribution appropriately. Typically, the canonical PFs estimates the state as the mean of the posterior distribution. However, it is inappropriate when the distribution becomes multi-modal. The MAP-PF portion of the new algorithm calculates MAP estimation that corresponds to the most probable (tallest) mode of the multi-modal posterior distribution based on the Viterbi algorithm.

One of the critical issues in combining the Viterbi algorithm for MAP estimation and the MPF algorithm is accounting for the model error caused by evaluating the model at improbable state trajectories. Sample impoverishment refers to a phenomenon where all particles collapse to only highly probable state trajectories due to resampling. Under that situation, the model is evaluated at only highly probable state trajectories and the model error is negligible. However, this is not the case when applying the Viterbi algorithm to obtain MAP estimation since the particles cover the nonlinear portion of the state-space *densely* without the sample impoverishment issue. In the case where the likelihood function $p\left(\boldsymbol{y}_n|\boldsymbol{y}_{0:n-1}, \boldsymbol{x}_n^{\mathrm{N},(i)}\right)$ must be evaluated for particles $\boldsymbol{x}_n^{\mathrm{N},(i)}$ whose values may be far away from probable values. Since there is only one true state trajectory and the particles cover the state-space *densely*, most particles become associated with improbable state trajectories. Therefore, for most particles $\boldsymbol{x}_n^{\mathrm{N},(i)}$ the variance of measurement prediction residual, i.e. $\boldsymbol{y}_n - \hat{\boldsymbol{y}}^i_{n|0:n-1}$, becomes much larger than the measurement noise variance $R_{\boldsymbol{v},n}$. This underestimation of the prediction error causes the likelihood function to have a distribution that is too narrow, which in turn distorts the posterior distribution and ultimately leads to suboptimal particle selection. One elegant solution to this problem is to continuously estimate the prediction error covariance from the residuals for each particle. The adaptive covariance estimation method proposed by Myers is adopted and modified [101].

This is why the new particle filter is called the maximum *a posteriori* "adaptive" marginalized particle filter (MAM-PF).

Another critical issue in combining the Viterbi algorithm for MAP estimation and the MPF algorithm is handling the likelihood function $p\left(\boldsymbol{y}_n|\boldsymbol{y}_{0:n-1}, \boldsymbol{x}_n^{\mathrm{N},(i)}\right)$ appropriately during the Viterbi process. This likelihood function represents the distribution of the prediction error based on all the past measurements $\boldsymbol{y}_{0:n-1}$ and the current nonlinear state $\boldsymbol{x}_n^{\mathrm{N},(i)}$. This likelihood function can be obtained from the linear Kalman filter recursions as,

$$p\left(\boldsymbol{y}_n|\boldsymbol{y}_{0:n-1}, \boldsymbol{x}_n^{\mathrm{N},(i)}\right) = p\left(\boldsymbol{y}_n|\hat{\boldsymbol{x}}_{n|0:n-1}^{\mathrm{L},(k)}, \boldsymbol{x}_n^{\mathrm{N},(i)}\right). \tag{4.1}$$

It is important to recognize that when the Viterbi algorithm maximizes the likelihood function, it searches for the best (most probable) trajectory over the nonlinear $p\left(\boldsymbol{x}_n^{\mathrm{N},(i)}|\boldsymbol{x}_{n-1}^{\mathrm{N},(k)}\right)$ and linear $p\left(\boldsymbol{y}_n|\boldsymbol{x}_n^{\mathrm{N},(i)}, \hat{\boldsymbol{x}}_{n|0:n-1}^{\mathrm{L},(k)}\right)$ portions of the state space where $k = 1, \ldots, N_{\mathrm{p}}$. This means that the MAM-PF algorithm requires that the linear Kalman filter recursions be applied N_{p} times for each particle. This is essentially so that the maximization over all possible previous trajectories correctly accounts for the effect of the linear state estimates on the likelihood function. Applying the linear Kalman filter recursions N_{p} times for each particle requires too much computational power and makes the MAM-PF impractical. It is necessary to reduce the computational burden of the MAM-PF significantly without compromising its performance.

To summarize, the first contribution is to propose a new particle filter algorithm (MAM-PF) that leverages the advantages of the MAP-PF and MPF algorithms and devise a way to reduce its computational burden greatly with minimal performance loss.

4.2 Single Rhythmical Component Tracking

The single rhythmical component signal model (measurement model) can be written as,

$$\boldsymbol{y}_n = \boldsymbol{a}_{0,n} + \boldsymbol{a}_{1,n} \cos(\boldsymbol{\theta}_n) + \boldsymbol{a}_{2,n} \sin(\boldsymbol{\theta}_n) + \boldsymbol{v}_n \tag{4.2}$$

where $\boldsymbol{\theta}_n$ is the instantaneous angle of the fundamental frequency $\boldsymbol{f}_n$, $\boldsymbol{a}_{0,n}$ the signal trend, $\boldsymbol{a}_{1,n}$ and $\boldsymbol{a}_{2,n}$ the sinusoidal coefficients, and $\boldsymbol{v}_n$ a white Gaussian noise with variance $r_{\boldsymbol{v}}$. In this model, the signal $\boldsymbol{y}_n$ is only a nonlinear function of the instantaneous angle $\boldsymbol{\theta}_n$. Given the instantaneous angle $\boldsymbol{\theta}_n$, the state-space model is a linear function of the other parameters such as the coefficients $\boldsymbol{a}_{1,n}$ and $\boldsymbol{a}_{2,n}$ and the signal trend $\bar{\boldsymbol{y}}_n$. These linear parameters can be estimated *optimally* using the Kalman filtering technique. Since the parameters are not known in practical applications, it is common to use a random walk model [74]. That is,

$$\boldsymbol{a}_{0,n+1} = \boldsymbol{a}_{0,n} + \boldsymbol{u}_{a,n} \tag{4.3}$$

$$\boldsymbol{a}_{1,n+1} = \boldsymbol{a}_{1,n} + \boldsymbol{u}_{a,n} \tag{4.4}$$

$$\boldsymbol{a}_{2,n+1} = \boldsymbol{a}_{2,n} + \boldsymbol{u}_{a,n} \tag{4.5}$$

where $\boldsymbol{u}_.$ represents a white Gaussian noise with variance $q_{\boldsymbol{u}_.}$. The instantaneous angle $\boldsymbol{\theta}_n$, however, can be modeled based on some domain knowledge. For the applications included in the dissertation it is known that the instantaneous frequency $\boldsymbol{f}_n$ changes slowly within a certain range. Therefore, the instantaneous angle $\boldsymbol{\theta}_n$

can be modeled as,

$$\bar{\boldsymbol{f}}_{n+1} = g\left[\bar{\boldsymbol{f}}_n + \boldsymbol{u}_{\bar{f},n}\right] \tag{4.6}$$

$$\boldsymbol{f}_{n+1} = \bar{\boldsymbol{f}}_n + \alpha\left(\boldsymbol{f}_n - \bar{\boldsymbol{f}}_n\right) + \boldsymbol{u}_{f,n} \tag{4.7}$$

$$\boldsymbol{\theta}_{n+1} = \boldsymbol{\theta}_n + 2\pi T_{\mathrm{s}}\boldsymbol{f}_n \tag{4.8}$$

where $\bar{\boldsymbol{f}}_n$ is the mean fundamental frequency, T_{s} is the sampling interval, α is an autoregressive (AR) coefficient, and $g[\cdot]$ represents a nonlinear reflecting function to account for the limited frequency range. A value of $\alpha = 1$ results in a random walk model while $\alpha = 0$ results in a white noise model. The nonlinear reflecting function can be expressed as,

$$g[f] = \begin{cases} f_{\text{max}} - (f - f_{\text{max}}) & f_{\text{max}} < f \\ f & f_{\text{min}} < f \leq f_{\text{max}} \\ f_{\text{min}} + (f_{\text{min}} - f) & f \leq f_{\text{min}} \end{cases} \tag{4.9}$$

where f_{min} and f_{max} represent the extreme frequency values.

Single rhythmical component tracking is a special case of the multi-harmonic tracking where the number of rhythmical components is one. In this case the posterior distribution of the frequency is unimodal. The application of the single rhythmical component tracking technique is tracking the frequency and magnitude of tremor activity in binary spike trains. Binary spike trains are a sequence of 0's and 1's where 1's correspond to a neuronal cell's firing activity. Binary spike trains are constructed from microelectrode recordings (MERs), which represent electrical activities of neuronal cells in the brain. MERs recorded during neurosurgery for subjects with movement disorders such as Parkinson's disease and essential tremor often contain pathophysiological rhythmicity. This rhythmicity is exhibited in MERs through pulse frequency modulation that causes the fluctuations in the mean firing rate of neuronal cells. Physiologically, the rhythmicity is due to local synchronization of neuronal cells. It is often called "tremor" because its frequency is similar to that of physical tremor of the limbs. It is intermittent and usually does not have any harmonic structure. Analyzing the frequency and magnitude of tremor in binary spike trains is a critical step for further study to characterize the relation of two or more tremor signals [102, 103]. This relationship is often referred to as phase coupling, which is an important clue for the origin of tremor in movement disorders [104, 105].

Hurtado *et al.* have conducted the most thorough study of tremor signals to date [103]. They relied on the Hilbert transform to estimate Gabor's analytical signal and to track the tremor frequency of binary spike trains. However, binary spike trains rarely meet the condition necessary for this estimation to be accurate. More importantly, their method can track only tremor frequency. I developed a tremor tracker based on the extended Kalman filter (EKF) [20], which can track both the frequency and intensity of tremor in binary spike trains. Since the posterior distribution of the tremor frequency is unimodal in tremor tracking, a new tremor tracker based on the proposed particle filtering method may not outperform the EKF-based tremor tracker substantially. However, it is still worthwhile to compare the performance of two tremor trackers based on synthetic and real binary spike trains. Specially when the tremor suddenly starts and stops, the new tremor tracker may be able to lock on to the tremor more quickly than the EKF-based one.

4.3 Single Harmonic Set Tracking

The single harmonic set signal model (measurement model) can be written as,

$$\boldsymbol{y}_n = \boldsymbol{a}_{0,n} + \left[\sum_{k=1}^{N_\text{h}} \boldsymbol{a}_{1,k,n} \cos\left(k\boldsymbol{\theta}_n\right) + \boldsymbol{a}_{2,k,n} \sin\left(k\boldsymbol{\theta}_n\right)\right] + \boldsymbol{v}_n \qquad (4.10)$$

where N_h is the number of harmonics or partials (assumed to be known), $\boldsymbol{\theta}_n$ the instantaneous angle of the fundamental frequency $\boldsymbol{f}_n$, $\boldsymbol{a}_{0,n}$ the signal trend, $\boldsymbol{a}_{1,k,n}$ and $\boldsymbol{a}_{2,k,n}$ the sinusoidal coefficients of the k^{th} partial, and $\boldsymbol{v}_n$ is a white Gaussian noise with variance $r_{\boldsymbol{v}}$. The process models for the sinusoidal coefficients $\boldsymbol{a}_{1,k,n}$ and $\boldsymbol{a}_{2,k,n}$, the signal trend $\boldsymbol{a}_{0,n}$, and the angle $\boldsymbol{\theta}_n$ are the same as in the single rhythmical component signal model in (4.5) and (4.8).

The single harmonic set signal model in (4.10) is a linear function of the coefficients $\boldsymbol{a}_{1,k,n}$ and $\boldsymbol{a}_{2,k,n}$ and the signal trend $\boldsymbol{a}_{0,n}$ given the instantaneous angle $\boldsymbol{\theta}_n$. Therefore, these quantities can be marginalized and calculated *optimally* using a linear Kalman filter. However, the fundamental frequency $\boldsymbol{f}_n$ has a nonlinear relationship with the signal $\boldsymbol{y}_n$ and its posterior distribution is multi-modal. The particle filtering technique can handle this multi-modality of the distribution properly. When the posterior distribution is multi-modal, the MAP estimate is generally a more reasonable estimate than the mean estimate since the mean of the distribution may be improbable. Therefore, the proposed maximum *a posteriori* adaptive marginalized particle filter (MAM-PF) is an appropriate method for multi-harmonic tracking.

Its application is tracking the heart rate in ECGs. Most heart rate estimation techniques involve beat detection algorithms. The heart rate, then, is defined as the inverse of intervals between detected beats. However, this approach is susceptible to noise and does not provide a continuous estimate of the heart rate. More critically, it requires a very reliable beat detection algorithm, which is difficult to design and implement. In contrast, the MAM-PF multi-harmonic tracker can estimate the heart rate in ECGs accurately even when they are contaminated by severe noise due to medical interventions and/or the mechanical system. It also provides a continuous measure of the heart rate.

Traditional multi-harmonic tracking techniques such as PDAs often pay attention to only the fundamental component in the signal. However, it is not uncommon that the fundamental component in ECGs has less power than higher harmonic components. It is, therefore, challenging to estimate the heart rate in ECGs using traditional techniques. The MAM-PF multi-harmonic tracker estimates the fundamental frequency, i.e. heart rate, by taking into account the harmonic relationship between the fundamental and harmonic partials. Therefore, it can estimate the heart rate even when the fundamental partial is not present in the signal. The MAM-PF multi-harmonic tracker will be compared to the EKF-based multi-harmonic tracker using synthetic and real ECG signals. The main purpose of this comparison is to demonstrate that the MAM-PF multi-harmonic tracker can handle the multi-modality of the posterior distribution of the fundamental frequency $\boldsymbol{f}$ than the EKF-based one.

4.4 Amplitude Modulated Harmonic Set Tracking

The amplitude-modulated harmonic set signal model is for quasi-periodic signals containing two harmonic components where one component is amplitude-modulated by the other. The amplitude-modulated harmonic set signal model (measurement model) can be expressed as,

$$\begin{aligned}
\boldsymbol{y}_n =& \boldsymbol{a}_{0,n} + \boldsymbol{s}_{1,n} + \sum_{k=1}^{N_{\mathrm{h}}^{\mathrm{c}}} \boldsymbol{m}_{k,n} \boldsymbol{s}_{2,k,n} + \boldsymbol{v}_n && (4.11)\\
\boldsymbol{s}_{1,n} =& \sum_{k=1}^{N_{1,\mathrm{h}}} \boldsymbol{a}_{1,k,n} \cos\left(k\boldsymbol{\theta}_{1,n}\right) + \boldsymbol{a}_{2,k,n} \sin\left(k\boldsymbol{\theta}_{1,n}\right) && (4.12)\\
\boldsymbol{m}_{k,n} =& 1 + \sum_{j=1}^{N_{1,\mathrm{h}}} \boldsymbol{a}_{3,k,j,n} \cos\left(j\boldsymbol{\theta}_{1,n}\right) + \boldsymbol{a}_{4,k,j,n} \sin\left(j\boldsymbol{\theta}_{1,n}\right) && (4.13)\\
\boldsymbol{s}_{2,k,n} =& \sum_{k=1}^{N_{2,\mathrm{h}}} \boldsymbol{a}_{5,k,n} \cos\left(k\boldsymbol{\theta}_{2,n}\right) + \boldsymbol{a}_{6,k,n} \sin\left(k\boldsymbol{\theta}_{2,n}\right) && (4.14)
\end{aligned}$$

where $\boldsymbol{s}_{1,n}$ and $\boldsymbol{s}_{2,k,n}$ are the first and second harmonic set, $\boldsymbol{m}_{k,n}$ the amplitude modulation factor, $N_{1,\mathrm{h}}$ a known number of harmonics of the first harmonic set, $N_{2,\mathrm{h}}$ that of the second harmonic component, $\boldsymbol{\theta}_{1,n}$ the instantaneous angle of the first harmonic component, and $\boldsymbol{\theta}_{2,n}$ the instantaneous angle of the second harmonic component. The process models for the sinusoidal coefficients $\{\boldsymbol{a}_{1,k,n}, \ldots, \boldsymbol{a}_{6,k,n}\}$, the signal trend $\boldsymbol{a}_{0,n}$, and the instantaneous angle $\boldsymbol{\theta}_{2,n}$ are the same as in the single rhythmical component signal model in (4.5) and (4.8). The instantaneous angle $\boldsymbol{\theta}_{1,n}$ is modeled as in (4.15).

Multi-harmonic tracking is for signals which contain a single set of multi-harmonics. However, some signals can contain multiple sets of multi-harmonics. Multiple sets of multi-harmonics often have a nonlinear interaction with each other. One possible nonlinear interaction is amplitude modulation. Here, it is assumed that the signal has two multi-harmonic sets and the first harmonic set modulates the amplitude of the second. The main goal is to measure the degree of amplitude modulation. A secondary goal is to estimate the fundamental frequency of the second harmonic set.

This problem arises in tracking the pulse pressure variation index (ΔPP) in ABP signals recorded under a full mechanical ventilation. The respiratory rate is known and constant since patients are under full mechanical ventilation. Since the first harmonic set $\boldsymbol{s}_{1,n}$ models the respiratory component, the fundamental frequency of the first harmonic set is modeled to be constant. Therefore, the instantaneous angle of the first harmonic component $\boldsymbol{\theta}_{1,n}$ can be written as,

$$\boldsymbol{\theta}_{1,n} = 2\pi n T_{\mathrm{s}} \boldsymbol{f}_1 \tag{4.15}$$

where $\boldsymbol{f}_1$ is equal to the mechanical ventilation rate.

The pulse pressure variation index in ABP signals simply denotes the degree of amplitude modulation of the cardiac component by the respiratory component. Numerous studies have shown that the pulse pressure variation ΔPP is one of most specific and sensitive predictors of fluid responsiveness in mechanically ventilated patients [106]. The current state-of-art in estimating the pulse pressure variation index relies on a beat detection algorithm, which is susceptible to noise [107]. In this dissertation a novel automatic algorithm is proposed to track the pulse pressure variation index continuously without utilizing any beat detection method.

4.5 Multiple Harmonic Set Tracking

The multiple harmonic set signal model (measurement model) can be expressed as follows,

$$\boldsymbol{y}_n = \boldsymbol{a}_{0,n} + \left[\sum_{j=1}^{N_\mathrm{c}} \sum_{k=1}^{N_{j,\mathrm{h}}} \boldsymbol{a}_{1,j,k,n} \cos\left(k\boldsymbol{\theta}_{j,n}\right) + \boldsymbol{a}_{2,j,k,n} \sin\left(k\boldsymbol{\theta}_{j,n}\right) \right] + \boldsymbol{v}_n \tag{4.16}$$

When N_c is the number of harmonic components, $N_{j,\mathrm{h}}$ the number of harmonics or partials of the j^th harmonic set, and $\boldsymbol{\theta}_{j,n}$ the fundamental angle of the j^th harmonic set. The process models for $\boldsymbol{a}_{1,j,k,n}$, $\boldsymbol{a}_{0,n}$, and $\boldsymbol{\theta}_{j,n}$ are the same as in the single rhythmical component signal model in (4.5) and (4.8).

The number of harmonics N_h may be given or need to be estimated. A simple estimation method is using a conservative upper bound on N_h where the excessive partials' coefficients will be estimated as nearly zero at a cost of extra variance. If some pilot data is available, N_h can be estimated more accurately based on spectral analysis of the pilot data.

Multiple harmonic set tracking is the most general case of multi-harmonic tracking. In this case the number of harmonic sets is more than one and the fundamental frequency of each harmonic set is unknown and time-varying. The type of nonlinear interactions between the multi-harmonics sets is also not limited to one. For example, two multi-harmonic sets can interact with each other through amplitude modulation and frequency modulation. Here it is assumed that there are two multi-harmonic sets and the first set modulates the amplitude of the second. The dimension of the particle filtering space is two since there are two fundamental frequencies to track. Applying the MAM-PF to multiple harmonic set tracking is difficult for a couple of reasons. The most critical reason is that the computational burden of particle filtering increases exponentially as the dimension of particle filtering increases. For instance, if N_p particles are needed to track only the fundamental frequency of the first harmonic set, the number of particles needed to track both the fundamental frequencies of two multiharmonic sets is N_p^2. One solution is to partition the multi-dimensional (2-D) space of particle filtering into two 1-D spaces. Having particles cover two 1-D spaces densely is more efficient than having them cover one 2-D space. However, this partition is possible only when two fundamental frequencies, $\boldsymbol{f}_{1,n}$ and $\boldsymbol{f}_{2,n}$, can be modeled to be independent of each other.

One application is tracking the respiratory component, the cardiac component, and the interaction between them in ABP signals recorded from subjects with spontaneous breathing. Currently, there are a very limited number of references on simultaneous monitoring of those three components in any pressure signals [108]. Accurate and simultaneous monitoring of the respiratory rate, the heart rate, and their interaction would provide rich information for physicians to diagnose potential respiratory dysfunction and cardiovascular diseases. The multiple harmonic set signal tracker based on the MAM-PF will be a useful tool to achieve this goal.

The subsequent chapters describe each of the five main contributions: new particle filtering technique (Chapter 5), single rhythmical component tracking (Chapter 6), single harmonic set tracking (Chapter 7), amplitude modulated harmonic set tracking (Chapter 8), and multiple harmonic set tracking (Chapter 9). The focus of Chapter 5 is to explain the theoretical and technical part of developing the new particle filtering method while the rest chapters are focused on explaining the clinical importance of the applications and describing methodology for them.

Chapter 5

New Particle Filtering Method

This chapter describes details of two versions of the new particle filtering technique and other conventional variants of particle filters. The simulation results based on synthetic signals in this chapter also demonstrate the superior performance of the new particle filtering technique in comparison to a conventional particle filter, which is applicable to the state-space model in (3.31)–(3.33).

5.1 Algorithm Development

Several research groups have used the term "marginalized MAP estimation" to describe algorithms for various applications such as a fault diagnosis for autonomously operating systems [100], state estimation of jump Markov linear systems [109], tempo tracking and rhythm quantization in music [110], and detection for Orthogonal Frequency Division Modulation (OFDM) systems [111]. Doucet *et al.* described an algorithm that obtains the marginal MAP estimate of the state of a Jump Markov Linear System based on Markov chain Monte Carlo (MCMC) methods [109]. Cemgil *et al.* discussed the possibility of computing the MAP trajectory after integrating out (Rao-Blackwellizing) the hidden variables based on the SMC methods [110]. However, they were aware that Rao-Blackwellization causes coupling between all possible particle trajectories and that the Viterbi algorithm does not find the actual MAP trajectory in this case. In order to find the true MAP trajectory, they suggested to apply an iterative improvement technique or simulated annealing to the best trajectory obtained by running the Viterbi algorithm, though this also only produces an estimate of the optimal MAP trajectory. Yee *et al.* used Rao-Blackwellized SMC methods to develop a MAP detector for realistic orthogonal frequency division multiplexing (OFDM) systems [111]. They divided the estimation problem into two stages: marginalization and MAP estimation. They first utilized the marginalized particle filter (MPF) to obtain the minimum mean-square-error (MMSE) estimate of encoded symbols and searched for the MAP estimate of original symbols based on the MMSE estimate of encoded symbols. None of these algorithms can be applied to general stochastic processes that can be expressed in the form of (3.31)–(3.33) and asymptotically converges to the optimal solution.

5.1.1 Standard Resampling Particle Filter

The standard resampling particle filter algorithm incorporates the stratified resampling scheme to overcome the sample degeneracy issue [112]. In the stratified resampling scheme, particles are resampled when the estimated number of effective particles $\hat{N}_\mathrm{e}$ becomes smaller than a user-specified threshold, N_t. The number of effective particles is given by

$$\hat{N}_\mathrm{e} = \frac{1}{\sum_{i=1}^{N_\mathrm{P}} \left(\boldsymbol{w}_n^{(i)}\right)^2} \tag{5.1}$$

where $\boldsymbol{w}_n^{(i)}$ is a normalized importance weight of the i^{th} particle and N_{p} is the number of particles. The normalized importance weights $\boldsymbol{w}_n$ are calculated at each time index from the unnormalized importance weights $\tilde{\boldsymbol{w}}_n$, which are derived from the normalized importance weights from the previous time index $\boldsymbol{w}_{n-1}$. Here it is assumed that the importance density has been chosen such that it can be factored as,

$$q(\boldsymbol{x}_{0:n}|\boldsymbol{y}_{0:n}) = q_n(\boldsymbol{x}_n|\boldsymbol{x}_{0:n-1}, \boldsymbol{y}_{0:n})q(\boldsymbol{x}_{0:n-1}|\boldsymbol{y}_{0:n-1}) \tag{5.2}$$

so that the importance weights can be calculated *recursively* as follows, Then, the weight update recursion can be written as,

$$\tilde{\boldsymbol{w}}_n^{(i)} = \boldsymbol{w}_{n-1}^{(i)} \frac{p(\boldsymbol{y}_n|\boldsymbol{x}_n^{(i)})p(\boldsymbol{x}_n^{(i)}|\boldsymbol{x}_{n-1}^{(i)})}{q_n(\boldsymbol{x}_n^{(i)}|\boldsymbol{x}_{n-1}^{(i)}, \boldsymbol{y}_{0:n})} \tag{5.3}$$

where $\tilde{\boldsymbol{w}}_n^{(i)}$ represents an unnormalized importance weight at time n and $\boldsymbol{w}_{n-1}^{(i)}$ a normalized importance weight at time $n-1$. This recursion is mathematically sound but cannot be implemented since the marginal importance density $q_n(\boldsymbol{x}_n|\boldsymbol{x}_{0:n-1}, \boldsymbol{y}_{0:n})$ is a quantity that we are trying to obtain. There are several ways to estimate the marginal importance density. The most common choice is to substitute it with the prior probability given by the process model $p(\boldsymbol{x}_n|\boldsymbol{x}_{n-1})$. The weight update recursion in (5.3), then, is simplified as follows,

$$\begin{aligned}\tilde{\boldsymbol{w}}_n^{(i)} &= \boldsymbol{w}_{n-1}^{(i)} \frac{p(\boldsymbol{y}_n|\boldsymbol{x}_n^{(i)})p(\boldsymbol{x}_n^{(i)}|\boldsymbol{x}_{n-1}^{(i)})}{q_n(\boldsymbol{x}_n^{(i)}|\boldsymbol{x}_{n-1}^{(i)}, \boldsymbol{y}_{0:n})} \\ &= \boldsymbol{w}_{n-1}^{(i)} p(\boldsymbol{y}_n|\boldsymbol{x}_n^{(i)}) \end{aligned} \tag{5.4}$$

where the current weight is the previous weight multiplied by the likelihood function, $p(\boldsymbol{y}_n|\boldsymbol{x}_n^{(i)})$. The current weight $\tilde{\boldsymbol{w}}_n^{(i)}$ is normalized as follows,

$$\boldsymbol{w}_n^{(i)} = \frac{\tilde{\boldsymbol{w}}_n^{(i)}}{\sum_j^{N_\text{P}} \tilde{\boldsymbol{w}}_n^{(j)}}. \tag{5.5}$$

Algorithm 1 explains the details of the standard resampling PF.

Algorithm 1 Standard Resampling Particle Filter (PF).

```
Initialization
  for $i = 1, ..., N_\mathrm{p}$ do
    Sample $\tilde{\boldsymbol{x}}_0^{(i)} \sim \pi_0(\tilde{\boldsymbol{x}}_0|\boldsymbol{y}_0)$
    Calculate initial importance weights:
      $\tilde{\boldsymbol{w}}_0^{(i)} = \pi_0(\boldsymbol{x}_0)p(\boldsymbol{y}_0|\tilde{\boldsymbol{x}}_0^{(i)})$
  end for
  for $i = 1, \ldots, N_\mathrm{p}$ do
    $\boldsymbol{w}_0^{(i)} = \tilde{\boldsymbol{w}}_0^{(i)} / \sum_j^{N_\mathrm{p}} \tilde{\boldsymbol{w}}_0^{(j)}$
  end for
Sequential Importance Sampling
  for $n = 1, \ldots, N_\mathrm{T}$ do
    if $\boldsymbol{N}_\mathrm{e} \geq N_\mathrm{t}$ then
      for $i = 1, \ldots, N_\mathrm{p}$ do
        $\boldsymbol{x}_{n-1}^{(i)} = \tilde{\boldsymbol{x}}_{n-1}^{(i)}$
      end for
    else
      Set $k = 1$ and $\tau = 1/N_\mathrm{p}$
      for $i = 1, \ldots, N_\mathrm{p}$ do
        while $\boldsymbol{w}_{n-1}^{(k)} \leq \tau$ do
          $k = k + 1$
        end while
        $\boldsymbol{x}_{n-1}^{(i)} = \tilde{\boldsymbol{x}}_{n-1}^{(k)}$
        $\boldsymbol{w}_{n-1}^{(i)} = 1/N$
        $\tau = \tau + 1/N$
      end for
    end if
    for $i = 1, \ldots, N_\mathrm{p}$ do
      $\tilde{\boldsymbol{x}}_n^{(i)} \sim q_n(\tilde{\boldsymbol{x}}_n^{(i)}|\boldsymbol{x}_{n-1}^{(i)}, \boldsymbol{y}_{0:n})$
      $\tilde{\boldsymbol{w}}_n^{(i)} = \boldsymbol{w}_{n-1}^{(i)} \frac{p(\boldsymbol{y}_n|\tilde{\boldsymbol{x}}_n^{(i)})p(\tilde{\boldsymbol{x}}_n^{(i)}|\boldsymbol{x}_{n-1}^{(i)})}{q_n(\tilde{\boldsymbol{x}}_n^{(i)}|\boldsymbol{x}_{n-1}^{(i)}, \boldsymbol{y}_{0:n})}$
    end for
    for $i = 1, \ldots, N_\mathrm{p}$ do
      $\boldsymbol{w}_n^{(i)} = \frac{\tilde{\boldsymbol{w}}_n^{(i)}}{\sum_j^{N_\mathrm{p}} \tilde{\boldsymbol{w}}_n^{(j)}}$
    end for
  end for
```

5.1.2 Maximum *A Posteriori* Particle Filter (MAP-PF)

The MAP-PF utilizes the Viterbi algorithm to provide the MAP sequence estimate of the state. By utilizing the Viterbi algorithm the MAP-PF can avoid the sample degeneracy and sample impoverishment problems of the standard PF method [97–99]. Therefore, the MAP-PF does not require any resampling scheme, which is devised to overcome the sample degeneracy problem but causes the sample impoverish problem.

The MAP-PF algorithm is memory-efficient because at each sample time the Viterbi algorithm discards $N_{\mathrm{p}}^2 - N_{\mathrm{p}}$ possible trajectories and only retains the N_{p} *most probable* ones. Here is how it is done. Each particle is propagated from the previous time index $n-1$ to the current time index n according to the marginal importance density $q_n(\boldsymbol{x}_n|\boldsymbol{x}_{n-1}, \boldsymbol{y}_{0:n})$. Then, the MAP-PF searches all possible trajectories of each particle for the *most probable* trajectory, which can be expressed as,

$$k^* = \underset{k}{\operatorname{argmax}} \ \boldsymbol{\alpha}_{n-1}^{(k)} p\left(\boldsymbol{x}_n^{(i)}|\boldsymbol{x}_{n-1}^{(k)}\right) \tag{5.6}$$

where k^* represents the index of the most probable trajectory. Although the current particle $\boldsymbol{x}_n^{(i)}$ is propagated from $\boldsymbol{x}_{n-1}^{(i)}$, the MAP-PF treats $\boldsymbol{x}_n^{(i)}$ as if it is propagated from $\boldsymbol{x}_{n-1}^{(k^*)}$. Among all possible trajectories N_{p} only the most probable path survives and the rest $N_{\mathrm{p}} - 1$ are discarded. Since this searching process is repeated for all particles N_{p}, the number of trajectories that are discarded is $(N_{\mathrm{p}} - 1)\, N_{\mathrm{p}} = N_{\mathrm{p}}^2 - N_{\mathrm{p}}$.

It is important to understand that the MAP-PF does not calculate or track importance weights $\boldsymbol{w}_n^{(i)}$ for each particle as in the standard resampling PF. The reason is that the MAP-PF does not estimate the mean or other moments of the state posterior distribution. Instead the MAP-PF simply tracks the posterior probability of each state trajectory, which is represented by the coefficient $\boldsymbol{\alpha}_n^{(i)}$. Algorithm 2 explains the details of the MAP-PF.

Algorithm 2 MAP Particle Filter (MAP-PF).

```
Initialization
  for $i = 1, ..., N_\mathrm{p}$ do
    Sample $\boldsymbol{x}_0^{(i)} \sim \pi_0(\boldsymbol{x}_0)$
    $\boldsymbol{\alpha}_0^{(i)} = \pi_0(\boldsymbol{x}_0^{(i)}) p(\boldsymbol{y}_0 | \boldsymbol{x}_0^{(i)})$
    $\boldsymbol{z}_0^{(i)} = \boldsymbol{x}_0^{(i)}$
  end for
  $i^* = \underset{i}{\operatorname{argmax}}\ \boldsymbol{\alpha}_0^{(i)}$
  $\hat{\boldsymbol{x}}_0 = \boldsymbol{x}_0^{(i^*)}$
Sequential MAP Estimation
  for $n = 1, \ldots, N_\mathrm{T}$ do
    for $i = 1, ..., N_\mathrm{p}$ do
      $\boldsymbol{x}_n^{(i)} \sim q_n(\boldsymbol{x}_n^{(i)} | \boldsymbol{x}_{n-1}^{(i)}, \boldsymbol{y}_{0:n})$
    end for
    for $i = 1, ..., N_\mathrm{p}$ do
      $k^* = \underset{k}{\operatorname{argmax}}\ \boldsymbol{\alpha}_{n-1}^{(k)} p(\boldsymbol{x}_n^{(i)} | \boldsymbol{x}_{n-1}^{(k)})$
      $\boldsymbol{z}_{0:n}^{(i)} = \left\{ \boldsymbol{z}_{0:n-1}^{(k^*)},\ \boldsymbol{x}_n^{(i)} \right\}$
      $\boldsymbol{\alpha}_n^{(i)} = \boldsymbol{\alpha}_{n-1}^{(k^*)} p(\boldsymbol{x}_n^{(i)} | \boldsymbol{x}_{n-1}^{(k^*)}) p(\boldsymbol{y}_n | \boldsymbol{x}_n^{(i)})$
    end for
    $i^* = \underset{i}{\operatorname{argmax}}\ \boldsymbol{\alpha}_n^{(i)}$
    $\hat{\boldsymbol{x}}_{0:n} = \boldsymbol{z}_{0:n}^{(i^*)}$
  end for
```

5.1.3 Marginalized Particle Filter (MPF)

The marginalized particle filter (MPF) can be applied to special state space models in which a portion of the state space is nonlinear and the other portion can be modeled as a linear process conditional upon the nonlinear portion of the state vector. When this partition can be performed, the linear portion of the state can be sequentially estimated using the conventional linear Kalman filter and particle filtering can be used to estimate the nonlinear portion of the state vector. There are several advantages of using the MPF algorithm when the partitioning of the state vector is possible. First, partitioning the state vector reduces the dimensionality of the state space where particle filtering needs to be used. Since particle filtering is computationally expensive, it is advantageous to limit the use of particle filtering to a portion of the entire state space. Secondly, the MPF reduces the variance of the posterior distribution estimation by providing an *optimal* estimate for the linear portion of the state space.

The main difference between the standard PF algorithm and the MPF algorithm is that the latter has the linear Kalman filter recursions between the particle propagation and the weight update steps. On the particle propagation step only the nonlinear portion of the state $\boldsymbol{x}_n^{\mathrm{N},(i)}$ is obtained according to the marginal importance density $q_n(\boldsymbol{x}_n^{\mathrm{N},(i)}|\boldsymbol{x}_{n-1}^{\mathrm{N},(i)}, \boldsymbol{y}_{0:n})$. After incorporating this nonlinear state into the state-space model in (3.31)–(3.33) the state-space model becomes completely linear with respect to the linear portion of the state $\boldsymbol{x}_n^{\mathrm{L},(i)}$. Then, its filtered quantity $\boldsymbol{x}_{n|0:n}^{\mathrm{L},(i)}$ and predicted quantity $\boldsymbol{x}_{n+1|0:n}^{\mathrm{L},(i)}$ are computed sequentially via the Kalman filter recursions. In order to complete the weight update recursion the probability $p(\boldsymbol{y}_n^{(i)}|\boldsymbol{y}_{0:n-1}, \boldsymbol{x}_n^{\mathrm{N},(i)})$ needs to be computed. The probability

$p(\boldsymbol{y}_n^{(i)}|\boldsymbol{y}_{0:n-1}, \boldsymbol{x}_n^{\mathrm{N},(i)})$ can be written as,

$$p(\boldsymbol{y}_n^{(i)}|\boldsymbol{y}_{0:n-1}, \boldsymbol{x}_n^{\mathrm{N},(i)}) = p(\boldsymbol{y}_n|\hat{\boldsymbol{x}}_{n|0:n-1}^{\mathrm{L},(i)}, \boldsymbol{x}_n^{\mathrm{N},(i)}) \tag{5.7}$$

$$\sim \mathcal{N}(\hat{\boldsymbol{y}}_{n|0:n-1}^{(i)}, \boldsymbol{R}_{\mathrm{e},n}^{(i)}) \tag{5.8}$$

where

$$\hat{\boldsymbol{y}}_{n|0:n-1}^{(i)} = H_n(\boldsymbol{x}_n^{\mathrm{N},(i)})\hat{\boldsymbol{x}}_{n|0:n-1}^{\mathrm{L},(i)} \tag{5.9}$$

$$\boldsymbol{R}_{\mathrm{e},n}^{(i)} = H_n\left(\boldsymbol{x}_n^{\mathrm{N},(i)}\right) \boldsymbol{C}_{n|0:n-1}^{(i)} H_n\left(\boldsymbol{x}_n^{\mathrm{N},(i)}\right)^T + R_v. \tag{5.10}$$

Algorithm 3 explains the details of the MPF.

Algorithm 3 Marginalized Particle Filter (MPF).

```
Initialization
  for $i = 1, ..., N_\mathrm{p}$ do
    Sample $\boldsymbol{x}_0^{\mathrm{N},(i)} \sim \pi_0(\boldsymbol{x}_0^\mathrm{N})$
    $\hat{\boldsymbol{x}}_{0:-1}^{\mathrm{L},(i)} = \mathrm{E}\left[\boldsymbol{x}_0^{\mathrm{L},(i)} | \boldsymbol{x}_0^{\mathrm{N},(i)}\right]$
    Initial Weight Calculation:
      $\tilde{\boldsymbol{w}}_0^{(i)} = p(\boldsymbol{y}_0 | \boldsymbol{x}_0^{\mathrm{N},(i)}, \boldsymbol{x}_0^\mathrm{L})$
  end for
Marginalized Sequential Estimation
  for $n = 0, ..., N_\mathrm{T}$ do
    if $N_\mathrm{e} \leq N_\mathrm{t}$ then
      Resample
    end if
    for $i = 1, ..., N_\mathrm{p}$ do
      Particle Propagation
        $\boldsymbol{x}_n^{\mathrm{N},(i)} \sim q_n(\boldsymbol{x}_n^{\mathrm{N},(i)} | \boldsymbol{x}_{n-1}^{\mathrm{N},(i)}, \boldsymbol{y}_{0:n})$
      Kalman Filtering
        Measurement Update:
        $\boldsymbol{R}_{\mathrm{e},n}^{(i)} = H_n\left(\boldsymbol{x}_n^{\mathrm{N},(i)}\right) \boldsymbol{C}_{n|0:n-1}^{(i)} H_n\left(\boldsymbol{x}_n^{\mathrm{N},(i)}\right)^T + R_v$
        $\boldsymbol{K}_n = \boldsymbol{C}_{n|0:n-1} H_n\left(\boldsymbol{x}_n^{\mathrm{N},(i)}\right)^T \left(\boldsymbol{R}_{\mathrm{e},n}^{(i)}\right)^{-1}$
        $\hat{\boldsymbol{y}}_{n|0:n-1}^{(i)} = H_n(\boldsymbol{x}_n^{\mathrm{N},(i)}) \hat{\boldsymbol{x}}_{n|0:n-1}^{\mathrm{L},(i)}$
        $\hat{\boldsymbol{x}}_{n|0:n}^{\mathrm{L},(i)} = \hat{\boldsymbol{x}}_{n|0:n-1}^{\mathrm{L},(i)} + \boldsymbol{K}_n \left[\boldsymbol{y}_n - \hat{\boldsymbol{y}}_{n|0:n-1}^{(i)}\right]$
        $\boldsymbol{C}_{n|0:n} = \left[I - \boldsymbol{K}_n H_n\left(\boldsymbol{x}_n^{\mathrm{N},(i)}\right)\right] \boldsymbol{C}_{n|0:n-1}^{(i)}$
        Time Update:
        $\hat{\boldsymbol{x}}_{n+1|0:n}^{\mathrm{L},(i)} = F_n(\boldsymbol{x}_n^{\mathrm{N},(i)}) \hat{\boldsymbol{x}}_{n|0:n}^{\mathrm{L},(i)}$
        $\boldsymbol{C}_{n+1|0:n}^{(i)} = F_n(\boldsymbol{x}_n^{\mathrm{N},(i)}) \boldsymbol{C}_{n|0:n} F_n(\boldsymbol{x}_n^{\mathrm{N},(i)})^T + \boldsymbol{Q}_u^\mathrm{L}$
      Weight Update
        $p(\boldsymbol{y}_n | \boldsymbol{y}_{0:n-1}, \boldsymbol{x}_{0:n}^{\mathrm{N},(i)}) \sim \mathcal{N}(\hat{\boldsymbol{y}}_{n|0:n-1}^{(i)}, \boldsymbol{R}_{\mathrm{e},n}^{(i)})$
        $\tilde{\boldsymbol{w}}_n^{(i)} = \boldsymbol{w}_{n-1}^{(i)} \frac{p(\boldsymbol{y}_n^{(i)} | \boldsymbol{y}_{0:n-1}, \boldsymbol{x}_n^{\mathrm{N},(i)}) p(\boldsymbol{x}_n^{\mathrm{N},(i)} | \boldsymbol{x}_{n-1}^{\mathrm{N},(i)})}{q_n(\boldsymbol{x}_n^{\mathrm{N},(i)} | \boldsymbol{x}_{0:n-1}^{\mathrm{N},(i)}, \boldsymbol{y}_{0:n})}$
    end for
    for $i = 1, \ldots, N_\mathrm{p}$ do
      $\boldsymbol{w}_n^{(i)} = \frac{\tilde{\boldsymbol{w}}_n^{(i)}}{\sum_j^{N_\mathrm{p}} \tilde{\boldsymbol{w}}_n^{(j)}}$
    end for
  end for
```

5.1.4 Optimal MAP Adaptive Marginalized PF (MAMPF)

One can apply the Viterbi algorithm to obtain the MAP state trajectory within the canonical particle filter framework [97, 99]. However, the Viterbi algorithm does not guarantee the true MAP state trajectory when the state vector is marginalized [110]. It has never been described how to obtain the MAP state trajectory with the marginalized state vector within the particle filter framework.

The MAM-PF is a hybrid particle filtering method which leverages the advantages of the MAP-PF and MPF algorithms. The MAP-PF portion of the algorithm permits the particles to densely cover the nonlinear portion of the state space. While this prevents the sample impoverishment problem that would normally be caused by resampling, it also means that the likelihood function $p\left(\boldsymbol{y}_n|\boldsymbol{y}_{0:n-1}, \boldsymbol{x}_n^{\mathrm{N},(i)}\right)$ must be evaluated for particles $\boldsymbol{x}_n^{\mathrm{N},(i)}$ whose values may be far away from probable values. In this case some of the variation in $\boldsymbol{y}_n$ caused by the true underlying state would not be accounted for and the residual variance $\boldsymbol{y}_n - \hat{\boldsymbol{y}}_{n|0:n-1}$ will be much larger than the measurement noise $R_{\boldsymbol{v},n}$ or the prediction error $R_{\boldsymbol{e},n}$ provided by the Kalman filter recursions. This underestimation of the prediction error causes the likelihood function to have a distribution that is too narrow, which in turn distorts the posterior distribution and ultimately leads to suboptimal particle selection. This is a critical problem that has not been addressed previously and only occurs when attempting to use both marginalization and MAP estimation with the Viterbi algorithm.

One elegant solution to this problem is to continuously estimate the prediction error covariance from the residuals for each particle. We adopted and modified the adaptive covariance estimation method proposed in [101]. To ensure that the estimated covariance matrix is positive semi-definite, we propose using an eigenvalue decomposition of the covariance matrix and eliminating all non-positive eigenvalues. We denote this operation as $[\boldsymbol{R}]_+$.

A second critical issue that occurs in merging marginalization and MAP estimation is that the likelihood function $p\left(\boldsymbol{y}_n|\boldsymbol{y}_{0:n-1}, \boldsymbol{x}_n^{\mathrm{N},(i)}\right)$ must be handled carefully because it is only conditioned on the nonlinear portion of the state vector. This distribution can be obtained from the Kalman filter recursions, as was done for the MPF

$$p\big(\boldsymbol{y}_n|\boldsymbol{y}_{0:n-1}, \boldsymbol{x}_n^{\mathrm{N},(i)}\big) = p\Big(\boldsymbol{y}_n|\hat{\boldsymbol{x}}_{n|0:n-1}^{\mathrm{L},(i)}, \boldsymbol{x}_n^{\mathrm{N},(i)}\Big) \tag{5.11}$$

$$\sim \mathcal{N}(\hat{\boldsymbol{y}}_{n|0:n-1}^{(i)}, \boldsymbol{R}_{e,n}^{(i)}) \tag{5.12}$$

However, during the maximization over all past trajectories it is crucial to recognize that this includes the linear portion of the state space, $\hat{\boldsymbol{x}}_{n|0:n-1}^{\mathrm{L},(i)}$, unlike the MAP-PF. Algorithm 4 explains a complete account of the Optimal MAM-PF recursions. The covariance coefficient β is a user-specified parameter that controls the memory of the recursion for first order recursive estimation of the adaptive signal prediction error covariance, $\boldsymbol{R}_{e,n}$.

Algorithm 4 Optimal MAM-PF.

```
Initialization
  for $i = 1, ..., N_\mathrm{p}$ do
    Sample $\boldsymbol{x}_0^{\mathrm{N},(i)} \sim \pi_0(\boldsymbol{x}_0^{\mathrm{N}})$ & $\hat{\boldsymbol{x}}_{0:-1}^{\mathrm{L},(i)} = \mathrm{E}\left[\boldsymbol{x}_0^{\mathrm{L},(i)} | \boldsymbol{x}_0^{\mathrm{N},(i)}\right]$
    $\boldsymbol{\alpha}_0^{(i)} = \pi_0\left(\boldsymbol{x}_0^{\mathrm{N},(i)}\right) p\left(\boldsymbol{y}_0 | \boldsymbol{x}_0^{\mathrm{N},(i)}, \boldsymbol{x}_0^{\mathrm{L},(i)}\right)$ & $\boldsymbol{z}_0^{(i)} = \boldsymbol{x}_0^{(i)}$
  end for
  $i^* = \underset{i}{\mathrm{argmax}}\ \alpha_0^{(i)}$ & $\hat{\boldsymbol{x}}_0 = \boldsymbol{x}_0^{i^*}$
  for $n = 1, ..., N_\mathrm{T}$ do
    for $i = 1, ..., N_\mathrm{p}$ do
      Particle Propagation:  $\boldsymbol{x}_n^{\mathrm{N},(i)} \sim q_n\left(\boldsymbol{x}_n^{\mathrm{N},(i)} | \boldsymbol{x}_{n-1}^{\mathrm{N},(i)}, \boldsymbol{y}_n\right)$
      Marginalized Sequential Estimation
      for $k = 1, ..., N_\mathrm{p}$ do
        $\hat{\boldsymbol{y}}_{n|0:n-1} = H_n\left(\boldsymbol{x}_n^{\mathrm{N},(i)}\right) \hat{\boldsymbol{x}}_{n|0:n-1}^{\mathrm{L},(k)}$ & $\boldsymbol{e}_n = \boldsymbol{y}_n - \hat{\boldsymbol{y}}_{n|0:n-1}$
        $\boldsymbol{R}_{\boldsymbol{v},n} = \left[\boldsymbol{e}_n \boldsymbol{e}_n^\mathrm{T} - H_n\left(\boldsymbol{x}_n^{\mathrm{N},(i)}\right) \boldsymbol{C}_{n|0:n-1}^{(k)} H_n\left(\boldsymbol{x}_n^{\mathrm{N},(i)}\right)^T\right]_+$
        $\hat{\boldsymbol{R}}_{v,n}^{(i,k)} = \beta \hat{\boldsymbol{R}}_{v,n-1}^{(k)} + (1-\beta) \boldsymbol{R}_{\boldsymbol{v},n}$
        $\boldsymbol{R}_{\boldsymbol{e},n}^{(k)} = H_n\left(\boldsymbol{x}_n^{\mathrm{N},(i)}\right) \boldsymbol{C}_{n|0:n-1}^{(k)} H_n\left(\boldsymbol{x}_n^{\mathrm{N},(i)}\right)^T + \hat{\boldsymbol{R}}_{v,n}^{(k)}$
        $\boldsymbol{K}_n^{(k)} = \boldsymbol{C}_{n|0:n-1}^{(k)} H_n\left(\boldsymbol{x}_n^{\mathrm{N},(i)}\right)^T \left(\boldsymbol{R}_{\boldsymbol{e},n}^{(k)}\right)^{-1}$
        $\hat{\boldsymbol{x}}_{n|0:n}^{\mathrm{L},(k)} = \hat{\boldsymbol{x}}_{n|0:n-1}^{\mathrm{L},(k)} + \boldsymbol{K}_n^{(k)} \left[\boldsymbol{y}_n - \hat{\boldsymbol{y}}_{n|0:n-1}^{(k)}\right]$
        $\boldsymbol{C}_{n|0:n}^{(k)} = \left[I - \boldsymbol{K}_n^{(k)} H_n\left(\boldsymbol{x}_n^{\mathrm{N},(i)}\right)\right] \boldsymbol{C}_{n|0:n-1}^{(k)}$
        $\hat{\boldsymbol{x}}_{n+1|0:n}^{\mathrm{L},(i,k)} = F_n\left(\boldsymbol{x}_n^{\mathrm{N},(i)}\right) \hat{\boldsymbol{x}}_{n|0:n}^{\mathrm{L},(k)}$ & $\boldsymbol{C}_{n+1|0:n}^{(i,k)} = F_n\left(\boldsymbol{x}_n^{\mathrm{N},(i)}\right) \boldsymbol{C}_{n|0:n}^{(k)} F_n\left(\boldsymbol{x}_n^{\mathrm{N},(i)}\right)^T + \boldsymbol{Q}_u^\mathrm{L}$
      end for
      MAP Estimation
      $k^* = \underset{k}{\mathrm{argmax}}\ \boldsymbol{\alpha}_{n-1}^{(k)} p\left(\boldsymbol{y}_n | \boldsymbol{x}_n^{\mathrm{N},(i)}, \hat{\boldsymbol{x}}_{n|0:n-1}^{\mathrm{L},(k)}\right) p\left(\boldsymbol{x}_n^{\mathrm{N},(i)} | \boldsymbol{x}_{n-1}^{\mathrm{N},(k)}\right)$
      $\boldsymbol{\alpha}_n^{(i)} = \boldsymbol{\alpha}_{n-1}^{(k^*)} p\left(\boldsymbol{y}_n | \boldsymbol{x}_n^{\mathrm{N},(i)}, \hat{\boldsymbol{x}}_{n|0:n-1}^{\mathrm{L},(k^*)}\right) p\left(\boldsymbol{x}_n^{\mathrm{N},(i)} | \boldsymbol{x}_{n-1}^{\mathrm{N},(k^*)}\right)$
      $\boldsymbol{C}_{n+1|0:n}^{(i)} = \boldsymbol{C}_{n+1|0:n}^{(i,k^*)}$ & $\hat{\boldsymbol{R}}_{v,n}^{(i)} = \hat{\boldsymbol{R}}_{v,n}^{(i,k^*)}$
      $\hat{\boldsymbol{x}}_{n+1|0:n}^{\mathrm{L},(i)} = \hat{\boldsymbol{x}}_{n+1|0:n}^{\mathrm{L},(i,k^*)}$ & $\hat{\boldsymbol{x}}_n^{(i)} = \left[\hat{\boldsymbol{x}}_{n|0:n}^{\mathrm{L},(k^*)},\ \boldsymbol{x}_n^{\mathrm{N},(i)}\right]^\mathrm{T}$ & $\boldsymbol{z}_{0:n}^{(i)} = \left[\boldsymbol{z}_{0:n-1}^{(k^*)}, \hat{\boldsymbol{x}}_n^{(i)}\right]$
    end for
    Update MAP State Estimate
    $i^* = \underset{i}{\mathrm{argmax}}\ \boldsymbol{\alpha}_n^{(i)}$ & $\hat{\boldsymbol{x}}_{0:n} = \boldsymbol{z}_{0:n}^{i^*}$
  end for
```

5.1.5 Fast MAP Adaptive Marginalized PF (MAMPF)

A key computational disadvantage of the Optimal MAM-PF algorithm is that the linear Kalman filter recursions must be applied N_p times for each particle, which results in N_p^2 Kalman filter recursions for each time update of the state estimate. This is necessary to ensure that the maximization over all possible previous trajectories correctly accounts for the effect of the linear state estimates on the likelihood function, which is given by $p\left(\boldsymbol{x}_n|\boldsymbol{x}_n^{\mathrm{N},(i)}, \hat{\boldsymbol{x}}_{n|0:n-1}^{\mathrm{L},(k)}\right)$ for the i^{th} particle. However, in most cases the likelihood function does not strongly affect the selection of the previous trajectory and this term can be eliminated from the MAP estimation step. Therefore, the searching step for the most probable trajectory can be simplified as follows,

$$k^* = \underset{k}{\operatorname{argmax}}\ \boldsymbol{\alpha}_{n-1}^{(k)} p\Big(\boldsymbol{y}_n|\boldsymbol{x}_n^{\mathrm{N},(i)}, \hat{\boldsymbol{x}}_{n|0:n-1}^{\mathrm{L},(k)}\Big)\, p\Big(\boldsymbol{x}_n^{\mathrm{N},(i)}|\boldsymbol{x}_{n-1}^{\mathrm{N},(k)}\Big) \tag{5.13}$$

$$\approx \underset{k}{\operatorname{argmax}}\ \boldsymbol{\alpha}_{n-1}^{(k)} p\Big(\boldsymbol{y}_n|\boldsymbol{x}_n^{\mathrm{N},(i)}, \hat{\boldsymbol{x}}_{n|0:n-1}^{\mathrm{L},(i)}\Big)\, p\Big(\boldsymbol{x}_n^{\mathrm{N},(i)}|\boldsymbol{x}_{n-1}^{\mathrm{N},(k)}\Big) \tag{5.14}$$

$$= \underset{k}{\operatorname{argmax}}\ \boldsymbol{\alpha}_{n-1}^{(k)} p\Big(\boldsymbol{x}_n^{\mathrm{N},(i)}|\boldsymbol{x}_{n-1}^{\mathrm{N},(k)}\Big)\,. \tag{5.15}$$

This approximation sacrifices the asymptotic optimality of the Optimal MAM-PF, but substantially reduces the computational burden since the selection of the most probable trajectory for a single particle no longer requires N_p times of the Kalman filter recursions for all possible past trajectories. Rather, the most probable trajectory can be determined before the Kalman filter recursions since (5.15) does not require any linear state estimates $\hat{\boldsymbol{x}}_{n|0:n-1}^{\mathrm{L},(k)}$ or $\hat{\boldsymbol{x}}_{n|0:n-1}^{\mathrm{L},(i)}$. In other words, the selection of the most probable trajectory is not affected by new information buried in the current measurement $\boldsymbol{y}_n$. Algorithm 5 explains the details of the Fast MAM-PF.

Algorithm 5 Fast MAM-PF.

```
Initialization
  for $i = 1, ..., N_\mathrm{p}$ do
    Sample $\boldsymbol{x}_0^{\mathrm{N},(i)} \sim \pi_0(\boldsymbol{x}_0^{\mathrm{N}})$ & $\hat{\boldsymbol{x}}_{0:-1}^{\mathrm{L},(i)} = \mathrm{E}\left[\boldsymbol{x}_0^{\mathrm{L},(i)} | \boldsymbol{x}_0^{\mathrm{N},(i)}\right]$
    $\boldsymbol{\alpha}_0^{(i)} = \pi_0\left(\boldsymbol{x}_0^{\mathrm{N},(i)}\right) p\left(\boldsymbol{y}_0 | \boldsymbol{x}_0^{\mathrm{N},(i)}, \boldsymbol{x}_0^{\mathrm{L},(i)}\right)$ & $\boldsymbol{z}_0^{(i)} = \boldsymbol{x}_0^{(i)}$
  end for
  $i^* = \underset{i}{\mathrm{argmax}}\ \alpha_0^{(i)}$ & $\hat{\boldsymbol{x}}_0 = \boldsymbol{x}_0^{i^*}$
  for $n = 1, ..., N_\mathrm{T}$ do
    for $i = 1, ..., N_\mathrm{p}$ do
      MAP Estimation
      Particle Propagation
        $\boldsymbol{x}_n^{\mathrm{N},(i)} \sim q_n\left(\boldsymbol{x}_n^{\mathrm{N},(i)} | \boldsymbol{x}_{n-1}^{\mathrm{N},(i)}, \boldsymbol{y}_n\right)$
        $k^* = \underset{k}{\mathrm{argmax}}\ \boldsymbol{\alpha}_{n-1}^{(k)} p\left(\boldsymbol{x}_n^{\mathrm{N},(i)} | \boldsymbol{x}_{n-1}^{\mathrm{N},(k)}\right)$
      Marginalized Sequential Estimation
      Measurement Update
        $\hat{\boldsymbol{y}}_{n|0:n-1} = H_n\left(\boldsymbol{x}_n^{\mathrm{N},(i)}\right) \hat{\boldsymbol{x}}_{n|0:n-1}^{\mathrm{L},(k^*)}$ & $\boldsymbol{e}_n = \boldsymbol{y}_n - \hat{\boldsymbol{y}}_{n|0:n-1}$
        $\boldsymbol{R}_{\boldsymbol{v},n} = \left[\boldsymbol{e}_n \boldsymbol{e}_n^{\mathrm{T}} - H_n\left(\boldsymbol{x}_n^{\mathrm{N},(i)}\right) \boldsymbol{C}_{n|0:n-1}^{(k^*)} H_n\left(\boldsymbol{x}_n^{\mathrm{N},(i)}\right)^T\right]_+$
        $\hat{\boldsymbol{R}}_{v,n} = \beta \hat{\boldsymbol{R}}_{v,n-1}^{(k^*)} + (1 - \beta) \boldsymbol{R}_{\boldsymbol{v},n}$
        $\boldsymbol{R}_{\boldsymbol{e},n} = H_n\left(\boldsymbol{x}_n^{\mathrm{N},(i)}\right) \boldsymbol{C}_{n|0:n-1}^{(k^*)} H_n\left(\boldsymbol{x}_n^{\mathrm{N},(i)}\right)^T + \hat{\boldsymbol{R}}_{v,n}$
        $\boldsymbol{K}_n = \boldsymbol{C}_{n|0:n-1}^{(k^*)} H_n\left(\boldsymbol{x}_n^{\mathrm{N},(i)}\right)^T (\boldsymbol{R}_{\boldsymbol{e},n})^{-1}$ & $\hat{\boldsymbol{x}}_{n|0:n}^{\mathrm{L}} = \hat{\boldsymbol{x}}_{n|0:n-1}^{\mathrm{L},(k^*)} + \boldsymbol{K}_n \boldsymbol{e}_n$
        $\boldsymbol{C}_{n|0:n} = \left[I - \boldsymbol{K}_n H_n\left(\boldsymbol{x}_n^{\mathrm{N},(i)}\right)\right] \boldsymbol{C}_{n|0:n-1}^{(k^*)}$
      Time Update
        $\hat{\boldsymbol{R}}_{v,n}^{(i)} = \hat{\boldsymbol{R}}_{v,n}$ & $\hat{\boldsymbol{x}}_{n+1|0:n}^{\mathrm{L},(i)} = F_n\left(\boldsymbol{x}_n^{\mathrm{N},(i)}\right) \hat{\boldsymbol{x}}_{n|0:n}^{\mathrm{L}}$
        $\boldsymbol{C}_{n+1|0:n}^{(i)} = F_n\left(\boldsymbol{x}_n^{\mathrm{N},(i)}\right) \boldsymbol{C}_{n|0:n} F_n\left(\boldsymbol{x}_n^{\mathrm{N},(i)}\right)^T + \boldsymbol{Q}_u^{\mathrm{L}}$
      $\boldsymbol{\alpha}_n^{(i)} = \boldsymbol{\alpha}_{n-1}^{(k^*)} p\left(\boldsymbol{y}_n | \boldsymbol{x}_n^{\mathrm{N},(i)}, \hat{\boldsymbol{x}}_{n|0:n-1}^{\mathrm{L},(k^*)}\right) p\left(\boldsymbol{x}_n^{\mathrm{N},(i)} | \boldsymbol{x}_{n-1}^{\mathrm{N},(k^*)}\right)$
      $\hat{\boldsymbol{x}}_n^{(i)} = \left[\hat{\boldsymbol{x}}_{n|0:n}^{\mathrm{L}}, \boldsymbol{x}_n^{\mathrm{N},(i)}\right]^{\mathrm{T}}$ & $\boldsymbol{z}_{0:n}^{(i)} = \left[\boldsymbol{z}_{0:n-1}^{(k^*)}, \hat{\boldsymbol{x}}_n^{(i)}\right]$
    end for
    Update MAP State Estimate
    $i^* = \underset{i}{\mathrm{argmax}}\ \boldsymbol{\alpha}_n^{(i)}$ & $\hat{\boldsymbol{x}}_{0:n} = \boldsymbol{z}_{0:n}^{i^*}$
  end for
```

5.2 Particle Filtering Algorithm Comparison

The previous sections described five different particle filter algorithms: standard resampling PF, MAP-PF, MPF, Optimal MAM-PF, and Fast MAM-PF. Among them only three algorithms are compared in terms of performance and computational load. Three PF algorithms include the MPF, Optimal MAM-PF, and Fast MAM-PF. The standard PF and MAP-PF algorithms are not considered for the comparison since they do not use marginalization to reduce the dimension of the nonlinear state where particle filters are used. These offer no advantages over the remaining three algorithms that utilize marginalization.

Multi-harmonic signal tracking is an excellent example where the state can be partitioned into the nonlinear and linear portions as in (3.31)–(3.32). The signal (measurement) model of the single harmonic set signal (simplest form of a multi-harmonic signal) can be written as,

$$\boldsymbol{y}_n = \bar{\boldsymbol{y}}_n + \left[\sum_{k=1}^{N_{\mathrm{h}}} \boldsymbol{a}_{k,n} \cos\left(k\boldsymbol{\theta}_n\right) + \boldsymbol{b}_{k,n} \sin\left(k\boldsymbol{\theta}_n\right) \right] + \boldsymbol{v}_n \tag{5.16}$$

where N_{h} is the number of harmonics or partials (assumed to be known), $\boldsymbol{\theta}_n$ the instantaneous angle of the fundamental frequency $\boldsymbol{f}_n$, $\bar{\boldsymbol{y}}_n$ the signal trend, $\boldsymbol{a}_{k,n}$ and $\boldsymbol{b}_{k,n}$ the sinusoidal coefficients of the k^{th} partial, and $\boldsymbol{v}_n$ is a white Gaussian noise with variance $r_{\boldsymbol{v}}$.

The nonlinear portion of the state $\boldsymbol{x}_n^{\mathrm{N}}$ includes the fundamental frequency $\boldsymbol{f}_n$

and the instantaneous angle $\boldsymbol{\theta}_n$ whose process model can be expressed as,

$$\begin{aligned}\bar{\boldsymbol{f}}_{n+1} &= g\left[\bar{\boldsymbol{f}}_n + \boldsymbol{u}_{\bar{f},n}\right] \\ \boldsymbol{f}_{n+1} &= \bar{\boldsymbol{f}}_n + \alpha\left(\boldsymbol{f}_n - \bar{\boldsymbol{f}}_n\right) + \boldsymbol{u}_{f,n} \\ \boldsymbol{\theta}_{n+1} &= \boldsymbol{\theta}_n + 2\pi T_{\mathrm{s}}\boldsymbol{f}_n \end{aligned} \tag{5.17}$$

where $\bar{\boldsymbol{f}}_n$ is the mean fundamental frequency, T_{s} is the sampling interval, α is an autoregressive (AR) coefficient, and $g[\cdot]$ represents a nonlinear reflecting function to account for the limited frequency range. The linear portion of the state $\boldsymbol{x}_n^{\mathrm{L}}$ includes the sinusoidal coefficients and the signal trend whose process model can be written as,

$$\boldsymbol{a}_{k,n+1} = \boldsymbol{a}_{k,n} + \boldsymbol{u}_{\boldsymbol{a},n} \tag{5.18}$$

$$\boldsymbol{b}_{k,n+1} = \boldsymbol{b}_{k,n} + \boldsymbol{u}_{\boldsymbol{b},n} \tag{5.19}$$

$$\bar{\boldsymbol{y}}_{n+1} = \bar{\boldsymbol{y}}_n + \boldsymbol{u}_{\bar{\boldsymbol{y}},n}. \tag{5.20}$$

In order to compare three particle filter algorithms, I implemented three multi-harmonic trackers utilizing those algorithms. All three multi-harmonic trackers share the same state-space model described above. Then, those trackers were applied to multi-harmonic signals that are synthetically generated based on the state-space model. Fig. 5.1 depicts the spectrogram of a synthetically generated 5 min multi-harmonic signal whose number of harmonics N_{h} is 10. Table 5.1 summarizes the parameters used to generate the synthetic signals.

In order to quantify the performance of the multi-harmonic trackers, two performance measures are computed: normalized mean-square-error (NMSE) of the

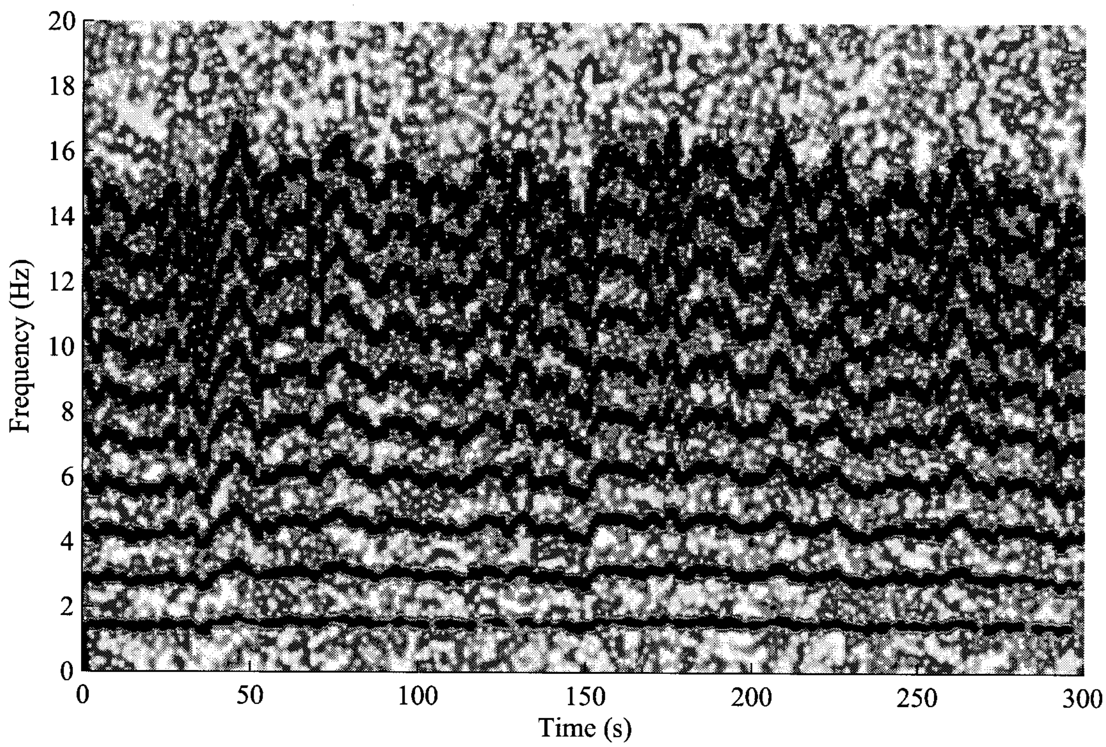


Figure 5.1: Spectrogram of a synthetic multi-harmonic signal with 10 harmonic partials.

Table 5.1: List of user-specified parameters to generate synthetic multi-harmonic signals.

Name	Symbol	Value
Number of harmonics	N_h	10
Sampling frequency	f_s	40 Hz
Signal duration	l	5 min
Number of samples	N_T	12,000
Signal-to-noise ratio	SNR	10 dB
Fundamental mean frequency	$\bar{\omega}$	3π
Frequency coefficient	α	0.99

predicted signal and mean-square-error of the frequency estimate (FMSE). NMSE measures how accurately the state estimates describe the original signal while FMSE represents the accuracy of the fundamental frequency estimation alone. For real signals FMSE typically cannot be computed since it requires knowledge of the true state which is unknown. NMSE of signal estimation ranges from 0 to inf. When its value is below 1, the tracker does a better job than a simple signal mean estimator. If its value is greater than 1, the tracker performs worse than estimating the signal to be equal to the signal mean. NMSE can be computed as follows,

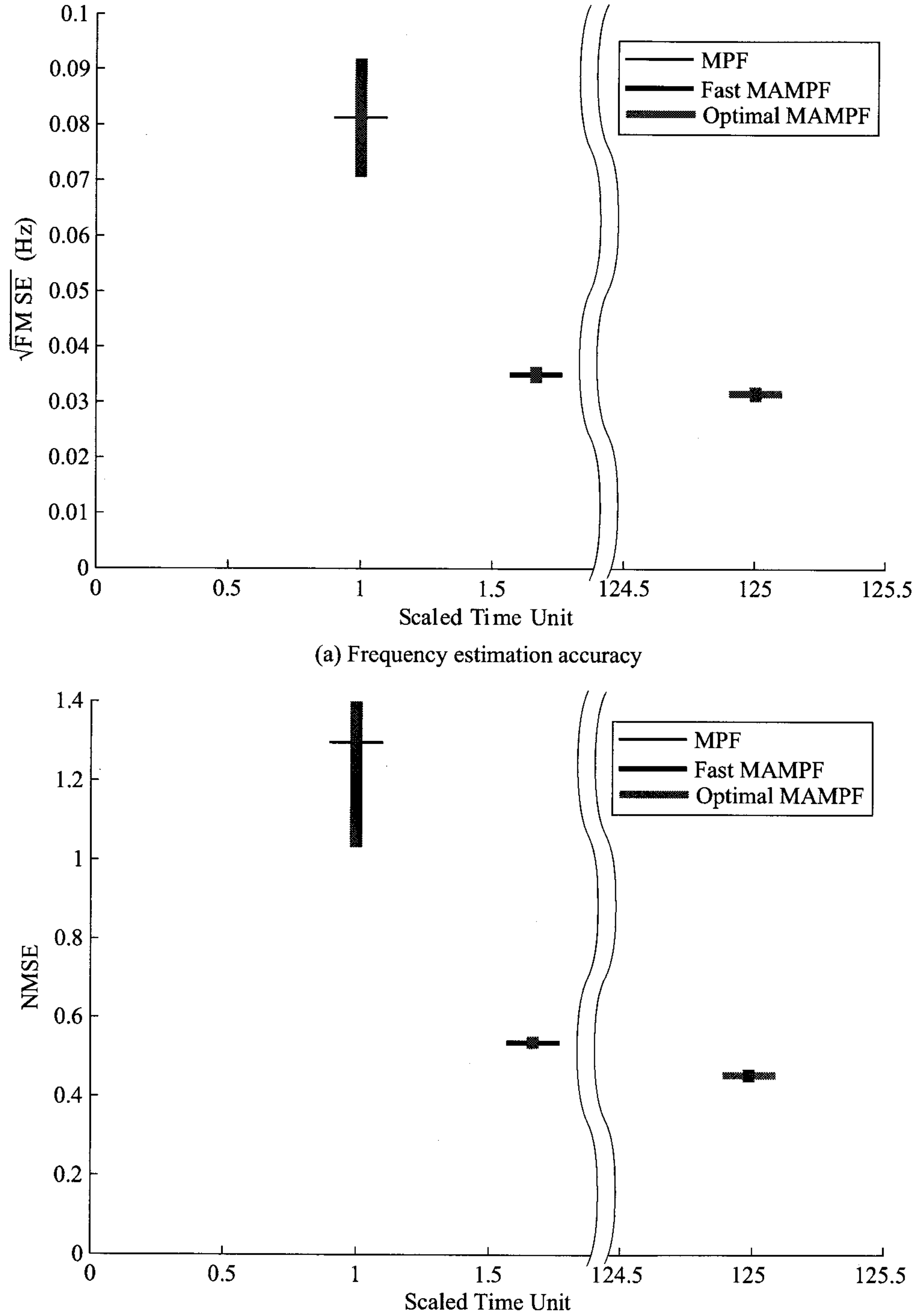
$$\mathrm{NMSE} = \frac{\sum_{n=1}^{N_\mathrm{T}} \left(\boldsymbol{y}_n - \hat{\boldsymbol{y}}_n\right)^2}{\sum_{n=1}^{N_\mathrm{T}} \left(\boldsymbol{y}_n - \bar{\boldsymbol{y}}_n\right)^2} \tag{5.21}$$

where $\bar{\boldsymbol{y}}_n$ represents the signal mean. FMSE can be written as follows,

$$\mathrm{FMSE} = \frac{\sum_{n=1}^{N_\mathrm{T}} \left(\boldsymbol{f}_n - \hat{\boldsymbol{f}}_n\right)^2}{N_\mathrm{T}} \tag{5.22}$$

whose unit is Hz^2. This will be reported as $\sqrt{\mathrm{FMSE}}$ since its unit is Hz.

Two plots in Fig. 5.2 depict $\sqrt{\mathrm{FMSE}}$ (top) and NMSE (bottom) of the three multi-harmonic trackers versus relative simulation times. The trackers used the same number of particles $N_\mathrm{p} = 100$. The Optimal MAM-PF multi-harmonic tracker substantially outperformed the MPF multi-harmonic tracker in terms of both performance measures. The main reason that the MPF tracker has a substantially larger $\sqrt{\mathrm{FMSE}}$ than the Optimal MAM-PF tracker is that the MPF tracker loses track of the true fundamental frequency occasionally and tracks one of the subharmonics, which correspond to the local maxima in the posterior distribution



(a) Frequency estimation accuracy

(b) Signal estimation accuracy

Figure 5.2: (a) $\sqrt{\text{FMSE}}$ versus simulation time. (b) NMSE versus simulation time. The horizontal lines represent the mean values while the vertical bars the one-standard-deviation ranges around the means.

shown in 3.3. The Optimal MAM-PF tracker is less likely to track the subharmonics erroneously than the MPF tracker since the particles of the Optimal MAM-PF are distributed through the entire fundamental frequency range and do not suffer from sample impoverishment. The erroneous fundamental frequency tracking of the MPF tracker also causes its NMSE to be slightly greater than 1. The simulation results demonstrate that the Optimal MAM-PF tracker performs substantially better than the MPF tracker. However, the issue is that the computational burden of the Optimal MAM-PF is much greater than that of the MPF. Fig. 5.2 shows that the Optimal MAM-PF tracker requires approximately 125 times more simulation time than the MPF tracker does. This may be unbearable in many cases considering that the MPF algorithm itself is already computationally expensive. In contrast, the Fast MAM-PF tracker requires only 1.6 times more simulation time than the MPF tracker does while its performance is comparable to the Optimal MAM-PF in terms of both FMSE and NMSE. This result is in line with the fact that the computational burdens of the MPF and MAM-PF algorithms are proportional to $\mathcal{O}(N_\mathrm{p})$ and $\mathcal{O}(N_\mathrm{p}^2)$, respectively with $N_\mathrm{p} = 100$.

5.3 Summary

This chapter provides details of the new particle filtering technique, called the *MAP Adaptive Marginalized PF*. The simulation results based on synthetic signals clearly demonstrate the superior performance of the new particle filtering technique in comparison to a conventional particle filter such as the MPF algorithm. The computational burden of the proposed particle filtering technique can be eased substantially by approximating the MAP state estimation step without sacrificing its performance.

Chapter 6

Single Rhythmical Component Tracking

This chapter discusses a practical application of the single rhythmical component tracking case. The application is tracking tremorous activity exhibited in neuronal signals of subjects with movement disorders such as Parkinson's disease (PD) and essential tremor (ET).

6.1 Introduction to the Clinical Problem

Tremor is one of the most disabling symptoms of many movement disorders such as Parkinson's disease (PD) and essential tremor (ET). Each movement disorder that causes tremor has a typical range of tremor frequencies that are observed in practice. For example, the tremor frequency of essential tremor ranges from 4–12 Hz [113].

Tremor activity can be measured with many types of instrumentation and sensors including electroencephalograms (EEG), magnetoencephalograms (MEG), electromyograms (EMG), accelerometers, gyroscopes, and microelectrode recordings (MER). Most tremor signals are quasi-periodic and nearly sinusoidal. The frequency of quasi-periodic tremorous activities in the signals is called the instantaneous tremor frequency (ITF) and the intensity of them called the instantaneous tremor amplitude (ITA).

A number of recent studies have focused on characterizing the relationship of two or more tremor signals. In many cases these signals are obtained from different types of instrumentation (e.g., MER and EMG). One of the surprising findings of these studies is that even when two signals contain significant tremor at the same frequency, these signals are not always coherent or phase-coupled [102, 103]. This suggests that the tremor either originates from multiple sources or that the tremor is modulated by uncoupled sources of unknown origin. A few studies have also found that the phase-coupling between pairs of tremor signals varies over time [103–105, 114].

Phase-coupling refers to a phenomenon of synchronization between tremorous activities in two signals. In the phase-coupling study, the main goal is to measure the degree of this synchronization between two signals' tremorous activities. However, one of the difficulties with studying phase-coupling is that this signal behavior cannot be characterized with traditional signal processing and time series analysis techniques that assume that the signals are generated by a linear stochastic process. These methods are essentially blind to subtle nonlinear effects, such as intermittent phase-coupling. This presents an opportunity for new signal processing methods that can estimate how the degree of phase-coupling between pairs of tremor signals varies over time. In order to measure the degree of the synchronization between tremorous activities, one first needs to segment the signals into *tremor-on* and *tremor-off* periods based on the tremor strength measurement, which is estimated instantaneous tremor amplitude (ITA). This step is called *tremor detection.* The next step is to track the instantaneous tremor frequency (ITF) of the tremor-on periods. It is common to perform the detection and tracking steps separately.

Neuronal recordings such as microelectrode recordings are an important type of signal to study since they are direct recordings of neuronal activities. Neuronal recordings are widely modeled as point processes consisting of a series of action potentials, or spikes, that are treated as all-or-none events. Most researches believe that all of the useful information is conveyed in the timing of these events. So, it is common practice to detect spikes in neuronal recordings during the early stages of analysis and focus all subsequent analysis on binary spike trains that consist of a 1 at the time of each spike occurrence and 0 elsewhere. Tremor in binary spike trains is exhibited through pulse frequency modulation that causes fluctuations in the mean firing rate [115]. The magnitude of fluctuations in the mean firing rate is the instantaneous tremor amplitude (ITA) and the frequency of the fluctuations is the instantaneous tremor frequency (ITF).

6.2 Summary of Prior Work

Hurtado *et al.* conducted the most thorough study of intermittent coupling of tremor signals to date [103,105]. They studied the synchronization between tremor-related activities in single-unit spike trains and EMG, where spike trains were recorded from globus pallidus internus (GPi) and EMG from the abductor pollicis (APB) in parkinsonian subjects. Prior to the synchronization study, they first *detected tremor-on periods* of the signals relying on traditional time-frequency analysis. Their tremor detection algorithm involves setting a threshold for instantaneous tremor amplitude (ITA). They selected a threshold value based on visual inspection of the signal's spectral components. Then, they applied an ITF tracking method based on the Hilbert transform to tremor-on periods of the signals. The Hilbert transform produces an estimate of Gabor's analytic signal. However, spike trains rarely meet the conditions necessary for this estimate to be accurate [17]. In particular, the representation of spikes as impulses results in a broad signal bandwidth that makes it difficult to track a single frequency. My previous work demonstrated that the Hilbert transform based ITF tracker does not produce an accurate estimate of ITF [20]. In the same work a Kalman filter based ITF tracker was proposed and simulation results demonstrated that it outperforms both the Hilbert transform based ITF tracker and the spectrogram based ITF tracker. Since the ITF has a nonlinear relationship with the signal, the extended Kalman filter (EKF) was utilized to implement the proposed ITF tracker, which uses a first-order Taylor series approximation around estimates of the current state. More recently, I proposed an extended Kalman smoother based tremor tracker, which can track the instantaneous tremor amplitude (ITA) and frequency (ITF) simultaneously [116].

This chapter describes a new tremor tracker which utilizes the fast maximum *a posteriori* adaptive marginalized particle filter (MAM-PF). While the EKF based tremor tracker relies on the local linearization of the current state to handle the nonlinear relationship between the signal and the ITF, the MAM-PF based tremor tracker applies the particle filtering technique to estimate the ITF and the conventional linear Kalman filter to estimate the ITA. The subsequent sections describe the extended Kalman filter based tremor tracker and the MAM-PF based tremor tracker and compare the performance of two tremor trackers based on synthetic and real spike trains.

6.3 Methodology

6.3.1 Measurement Model

The fluctuating firing intensity of binary spike trains can be modeled as follows,

$$\boldsymbol{y}_n = \boldsymbol{a}_{1,n} \cos(\boldsymbol{\theta}_n) + \boldsymbol{a}_{2,n} \sin(\boldsymbol{\theta}_n) + \boldsymbol{v}_n \tag{6.1}$$

where $\boldsymbol{\theta}_n$ is the instantaneous angle of the instantaneous tremor frequency (ITF), $\boldsymbol{a}_{1,n}$ and $\boldsymbol{a}_{2,n}$ the sinusoidal coefficients, and $\boldsymbol{v}_n$ a white Gaussian noise with variance $r_{\boldsymbol{v}}$. This signal model is the same as the single rhythmical component signal model shown in (4.2). The instantaneous tremor amplitude (ITA) can be expressed in terms of the coefficients, $\boldsymbol{a}_{1,n}$ and $\boldsymbol{a}_{2,n}$, as follows,

$$\boldsymbol{p}_n = \sqrt{\boldsymbol{a}_{1,n}^2 + \boldsymbol{a}_{2,n}^2} \tag{6.2}$$

where $\boldsymbol{p}_n$ represents the ITA. As mentioned previously, the single rhythmical component signal model can be seen as a special case of the single harmonic set signal model where the number of harmonics or partials is only 1.

6.3.2 Process Model

Given the instantaneous angle $\boldsymbol{\theta}_n$, the state-space model is a linear function of the other parameters such as the coefficients $\boldsymbol{a}_{1,n}$ and $\boldsymbol{a}_{2,n}$. These linear parameters can be estimated *optimally* using the Kalman filtering technique. Since these parameters are not known in practical applications, it is common to use a random walk model [74]. That is,

$$\boldsymbol{a}_{1,n+1} = \boldsymbol{a}_{1,n} + \boldsymbol{u}_{a,n} \tag{6.3}$$

$$\boldsymbol{a}_{2,n+1} = \boldsymbol{a}_{2,n} + \boldsymbol{u}_{a,n} \tag{6.4}$$

where $\boldsymbol{u}_{a,n}$ represents a white Gaussian noise with variance $q_{\boldsymbol{a}}$. The instantaneous tremor angle $\boldsymbol{\theta}_n$, however, can be modeled based on some domain knowledge. It is well known that the instantaneous tremor frequency (ITF) $\boldsymbol{f}_n$ changes slowly within a certain range [113]. Therefore, the instantaneous tremor angle $\boldsymbol{\theta}_n$ can be modeled as,

$$\bar{\boldsymbol{f}}_{n+1} = g\left[\bar{\boldsymbol{f}}_n + \boldsymbol{u}_{\bar{f},n}\right] \tag{6.5}$$

$$\boldsymbol{f}_{n+1} = \bar{\boldsymbol{f}}_n + \alpha\left(\boldsymbol{f}_n - \bar{\boldsymbol{f}}_n\right) + \boldsymbol{u}_{f,n} \tag{6.6}$$

$$\boldsymbol{\theta}_{n+1} = \boldsymbol{\theta}_n + 2\pi T_{\mathrm{s}}\boldsymbol{f}_n \tag{6.7}$$

where $\bar{\boldsymbol{f}}_n$ is the mean tremor frequency, T_{s} is the sampling interval, α is an autoregressive (AR) coefficient, and $g[\cdot]$ represents a nonlinear reflecting function to account for the limited frequency range. A value of $\alpha = 1$ results in a random walk model while $\alpha = 0$ results in a white noise model. The nonlinear reflecting function can be expressed as,

$$g[f] = \begin{cases} f_{\max} - (f - f_{\max}) & f_{\max} < f \\ f & f_{\min} < f \leq f_{\max} \\ f_{\min} + (f_{\min} - f) & f \leq f_{\min} \end{cases} \tag{6.8}$$

where $f_{\min}$ and $f_{\max}$ represent the possible extreme tremor frequency values. Now, The state vector $\boldsymbol{x}_n$ can be written as,

$$\begin{aligned} \boldsymbol{x}_n &= \begin{bmatrix} \boldsymbol{x}_{1,n} & \boldsymbol{x}_{2,n} & \boldsymbol{x}_{3,n} & \boldsymbol{x}_{4,n} \end{bmatrix}^{\mathrm{T}} \\ &= \begin{bmatrix} \boldsymbol{\theta}_n & \boldsymbol{f}_n & \boldsymbol{a}_{1,n} & \boldsymbol{a}_{2,n} \end{bmatrix}^{\mathrm{T}} \end{aligned} \tag{6.9}$$

6.3.3 Extended Kalman Filter Recursions

The Kalman filter recursions consist of the measurement update and time update. The measurement update produces the filtered estimates $\hat{\boldsymbol{x}}_{n|n}$ based on the previous predicted estimates $\hat{\boldsymbol{x}}_{n|n-1}$ and the current measurement $\boldsymbol{y}_n$. From the filtered estimates $\hat{\boldsymbol{x}}_{n|n}$ the time update produces the predicted estimates $\hat{\boldsymbol{x}}_{n+1|n}$ according to the state-space model.

The extended Kalman filter linearizes the nonlinear state-space model, $f(\boldsymbol{x}_n)$ and $h(\boldsymbol{x}_n)$ at different estimates of the state. The Jacobian of $f(\boldsymbol{x}_n)$ is evaluated at the filtered estimate $\hat{\boldsymbol{x}}_{n|n}$ and the Jacobian of $h(\boldsymbol{x}_n)$ is evaluated at the predicted estimate $\hat{\boldsymbol{x}}_{n|n-1}$. They can expressed as follows,

$$F_n = \left.\frac{\partial f_n(\boldsymbol{x})}{\partial \boldsymbol{x}}\right|_{\boldsymbol{x}=\hat{\boldsymbol{x}}_{n|n}} \tag{6.10}$$

$$H_n = \left.\frac{\partial h_n(\boldsymbol{x})}{\partial \boldsymbol{x}}\right|_{\boldsymbol{x}=\hat{\boldsymbol{x}}_{n|n-1}} \tag{6.11}$$

where $\hat{\boldsymbol{x}}_{n|n-\ell}$ denotes the estimate of the state vector based on the measurements $y_{0:n-\ell} = \{y_0, \ldots, y_{n-\ell}\}$. The subsequential sections explain the recursions for the EKF in detail.

Initialization

The EKF requires the user to provide an a priori estimate of the state mean and state error covariance matrices. These affect how quickly the tracker initially "locks in", but do not affect the steady-state performance. For the a priori state mean $\hat{\boldsymbol{x}}_{0|-1} = [0 \ \bar{f} \ 0 \ 0]^{\mathrm{T}}$ was chosen, where $\bar{f}$ is the mean tremor frequency determined by domain knowledge. I chose a diagonal initial state error covariance matrix, $P_{0|-1}$. Section 6.3.5 describes how to estimate the elements of the matrix.

EKF Recursions

The filtered and predicted state estimates can be computed directly from the well-known EKF recursions,

$$H_n = \begin{bmatrix} -\hat{\boldsymbol{x}}_{3,n|n-1}\sin\left(\hat{\boldsymbol{x}}_{1,n|n-1}\right) + \hat{\boldsymbol{x}}_{4,n|n-1}\cos\left(\hat{\boldsymbol{x}}_{1,n|n-1}\right) \\ 0 \\ \cos\left(\hat{\boldsymbol{x}}_{1,n|n-1}\right) \\ \sin\left(\hat{\boldsymbol{x}}_{1,n|n-1}\right) \end{bmatrix}^{\mathrm{T}} \tag{6.12}$$

$$r_{\mathrm{e},n} = r + H_n P_{n|n-1} H_n^{\mathrm{T}} \tag{6.13}$$

$$K_n = P_{n|n-1} H_n^{\mathrm{T}} r_{\mathrm{e},n}^{-1} \tag{6.14}$$

$$\hat{\boldsymbol{y}}_{n|n-1} = h(\hat{\boldsymbol{x}}_{n|n-1}) \tag{6.15}$$

$$\hat{\boldsymbol{x}}_{n|n} = \hat{\boldsymbol{x}}_{n|n-1} + K_n\left(\boldsymbol{y}_n - \hat{\boldsymbol{y}}_{n|n-1}\right) \tag{6.16}$$

$$P_{n|n} = P_{n|n-1} - K_n r_{\mathrm{e},n} K_n^{\mathrm{T}} \tag{6.17}$$

$$F_n = \begin{bmatrix} 1 & 2\pi T_s \hat{\boldsymbol{x}}_{2,n|n} & 0 & 0 \\ 0 & \alpha \hat{\boldsymbol{x}}_{2,n|n} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{6.18}$$

$$P_{n+1|n} = F_n P_{n|n} F_n^{\mathrm{T}} + Q \tag{6.19}$$

$$\hat{\boldsymbol{x}}_{n+1|n} = f(\hat{\boldsymbol{x}}_{n|n}) \tag{6.20}$$

These recursions produce both the filtered estimates $\hat{\boldsymbol{x}}_{n|n}$ and the predicted estimates $\hat{\boldsymbol{x}}_{n|n-1}$ of all the state variables.

6.3.4 Fast MAP Adaptive Marginalized Particle Filter Recursions

The Fast MAM-PF recursions consist of the particle propagation, the conventional linear Kalman filter updates, and the MAP state estimation. The state vector $\boldsymbol{x}_n$ is partitioned into the nonlinear $\boldsymbol{x}_n^{\mathrm{N}}$ and linear $\boldsymbol{x}_n^{\mathrm{L}}$ portions. In this application, only the instantaneous tremor angle $\boldsymbol{\theta}_n$ belongs to $\boldsymbol{x}_n^{\mathrm{N}}$ and the remaining state variables belong to $\boldsymbol{x}_n^{\mathrm{L}}$, which can be expressed as,

$$\boldsymbol{x}_n^{\mathrm{N}} = \begin{bmatrix} \boldsymbol{\theta}_n \end{bmatrix} \tag{6.21}$$

$$\boldsymbol{x}_n^{\mathrm{L}} = \begin{bmatrix} \boldsymbol{a}_{1,n} & \boldsymbol{a}_{2,n} \end{bmatrix}^{\mathrm{T}} . \tag{6.22}$$

Initialization

The MAM-PF also requires the user to provide an a priori estimate of the state mean and state error covariance matrices. For the a priori state mean $\hat{\boldsymbol{x}}_{0|-1} = [0 \ \bar{f} \ 0 \ 0]^{\mathrm{T}}$ was chosen, where $\bar{f}$ is the mean tremor frequency determined by domain knowledge. I chose a diagonal initial state error covariance matrix, $P_{0|-1}$. Section 6.3.5 describes how to estimate the elements of the matrix.

Fast MAM-PF Recursions

The details of the Fast MAM-PF recursions are listed in Algorithm 5 in Chapter 5.

6.3.5 Parameter Selection

There are several parameters in the measurement (6.1) and process models (6.3)–(6.7), which need to be specified or estimated. However, most of them can be estimated based on domain knowledge. Tables 6.1 and 6.2 summary the user-specified parameters for the EKF and MAM-PF tremor trackers, respectively. Since tracking the instantaneous tremor frequency (ITF) is the main goal of the current application, the frequency process noise variance q_f and the mean frequency process noise variance $q_{\bar{f}}$ are the most critical parameters that affect the performance of tremor tracking most. Except for those two parameters, the two tremor trackers share the same parameter values.

The user-specified parameters listed in Tables 6.1 and 6.2 have been optimized and used in my previous work [20, 116–119]. Those parameters do not need to be optimized for individual spike trains. The set of optimized parameters can be used for any spike trains.

6.3.6 Synthetic Spike Trains with Tremor

The true ITF of real spike trains can be only estimated. However, it is necessary to have spike trains with a known (or true) ITF to accurately compare the performance of two tremor trackers in Monte Carlo simulations. In [20] McNames *et al.* described a process for generating synthetic spike trains with a known ITF in detail. A spike train is created based on an integrate-and-fire model of a renewal process introduced in [120]. The distribution of a parameter τ determines the regularity of the firing rate, which is explained in [115].

Two types of synthetic spike trains were created: Continuous-tremor spike

Table 6.1: Summary of user-specified parameters for the EKF tremor tracker.

Name	Symbol	Value
Frequency coefficient	α	0.999
Frequency process noise variance	q_f	10 T_s
Mean Frequency process noise variance	$q_{\bar{f}}$	0.01 T_s
Amplitudes process noise variance	q_a	var($\boldsymbol{y}$)/10
Measurement noise variance	r	var($\boldsymbol{y}$)/10
Minimum possible frequency	f_{min}	3 Hz
Maximum possible frequency	f_{max}	10 Hz
Mean tremor frequency	$\bar{f}$	5 Hz
Phase initial	$\boldsymbol{\theta}_0$	0
Frequency initial	$\boldsymbol{f}_0$	5 Hz
Frequency mean initial	$\bar{\boldsymbol{f}}_0$	5 Hz
Amplitudes initial	$\boldsymbol{a}_0$	0.01
Initial phase variance	$q_{\theta,0\|-1}$	3.28
Initial frequency variance	$q_{f,0\|-1}$	4.08
Initial amplitude variance	$q_{a,0\|-1}$	0.1

Table 6.2: Summary of user-specified parameters for the Fast MAM-PF tremor tracker.

Name	**Symbol**	**Value**
Number of particles	N_{p}	1e3
Frequency coefficient	α	0.999
Covariance coefficient	β	0.98
Frequency process noise variance	q_f	1e-4 T_s
Mean Frequency process noise variance	$q_{\bar{f}}$	1e-6 T_s
Amplitudes process noise variance	q_a	var($\boldsymbol{y}$)/10
Measurement noise variance	r	var($\boldsymbol{y}$)/10
Min. mean frequency range	f_{min}	3 Hz
Max. mean frequency range	f_{max}	10 Hz
Mean tremor frequency	$\bar{f}$	5 Hz
Phase initial	$\boldsymbol{\theta}_0$	0
Amplitudes initial	$\boldsymbol{a}_0$	0.01
Initial phase variance	$q_{\theta,0\|-1}$	3.28
Initial frequency variance	$q_{f,0\|-1}$	4.08
Initial amplitude variance	$q_{s,0\|-1}$	0.1

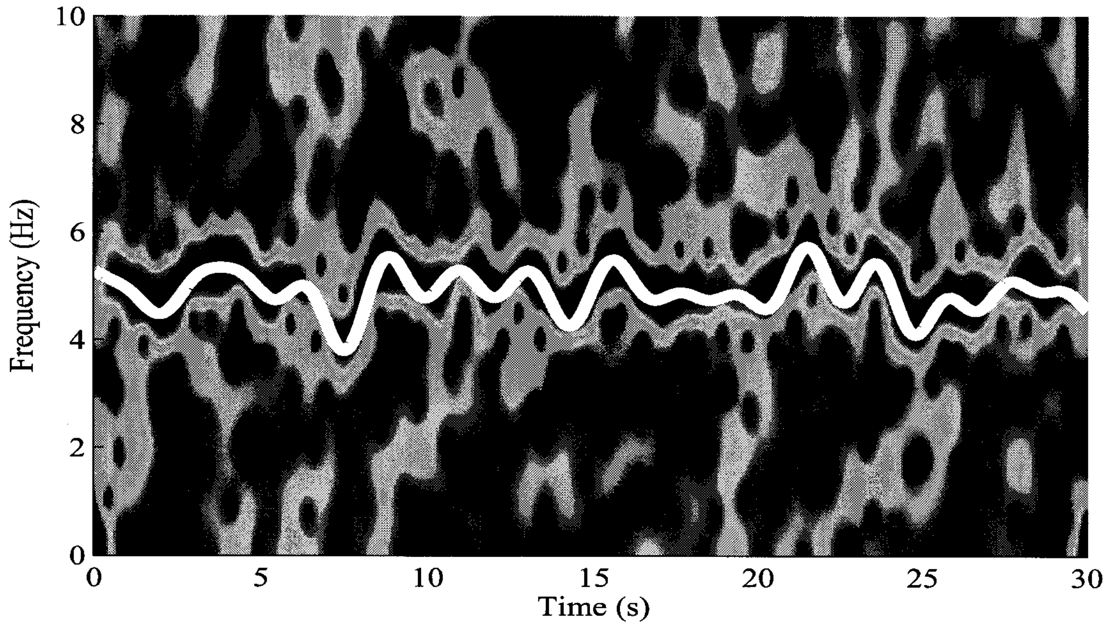


Figure 6.1: Spectrogram of an example binary spike train whose tremor is continuous where the thick white line represents the true ITF.

trains and intermittent-tremor spike trains. Each spike train lasts for 30 s. For continuous-tremor spike trains, tremorous activity lasts throughout the signal duration. Fig. 6.1 shows the spectrogram of an example continuous-tremor spike train. For intermittent-tremor spike trains, tremorous activity lasts only during the first and last 10 s periods. From 10 s to 20 s the firing rate was kept constant, which resulted in no tremor. Fig. 6.2 illustrates the spectrogram of an example intermittent-tremor spike train.

6.3.7 Real Spike Trains with Tremor

Both the EKF based and MAM-PF based tremor trackers were applied to a binary spike train constructed from a microelectrode recording (MER) recorded during stereotactic neurosurgery for a subject with Parkinson's disease (PD). Spikes in

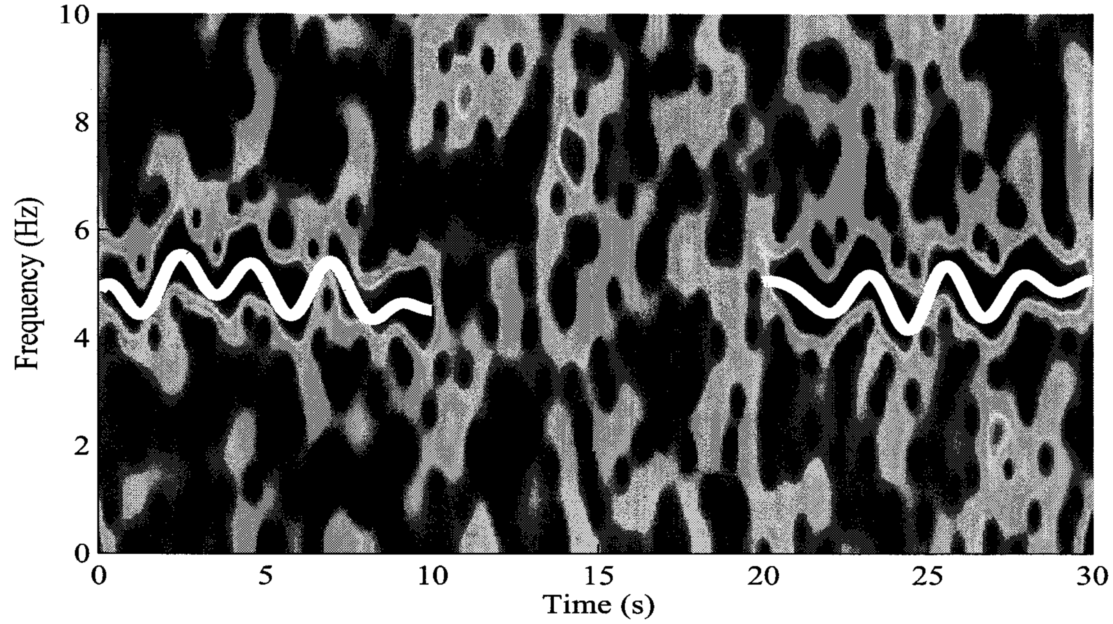


Figure 6.2: Spectrogram of an example binary spike train whose tremor is intermittent where the thick white line represents the true ITF.

the MER were detected using an automatic spike detection algorithm described in [121]. The duration of this real spike train was also 30 s and the original sampling frequency was 22 kHz.

6.4 Results and Discussion

The performance of two tremor trackers was compared by plotting the absolute difference between the true tremor frequency and its estimate versus time, where the absolute difference can be expressed as,

$$\mathrm{AFE}_n = \left| \boldsymbol{f}_n - \hat{\boldsymbol{f}}_n \right| \tag{6.23}$$

where AFE stands for the absolute frequency error. Fig. 6.3 illustrates the simulation results based on 100 continuous-tremor spike trains. The thick grey line represents the averaged AFE_n of the EKF based tremor tracker over the 100 simulations while the thin black line represents that of the MAM-PF based tremor tracker. The shaded areas around the averaged AFE_n represent the one standard deviation range of the EKF (very light grey) and MAM-PF (light grey) trackers. Overall, the averaged AFE_n of the MAM-PF based tremor tracker tends to be slightly greater than that of the EKF based tremor tracker. The mean values of the averaged AFE_n over 30 s were 0.36 Hz and 0.34 Hz, for the MAM-PF and EKF based tremor trackers respectively. As long as tremor is continuous in spike trains and the tremor trackers "lock in" to the true tremor frequency, the EKF based tremor tracker tracks the ITF slightly better than the MAM-PF based tremor tracker. The reason for better performance is that the EKF can track a single mode better than the MAM-PF when the EKF has locked on to the tremor signal. However, the performance of the MAM-PF will improve as the number of particles used increases.

Fig. 6.4 shows the simulation results based on 100 intermittent-tremor spike

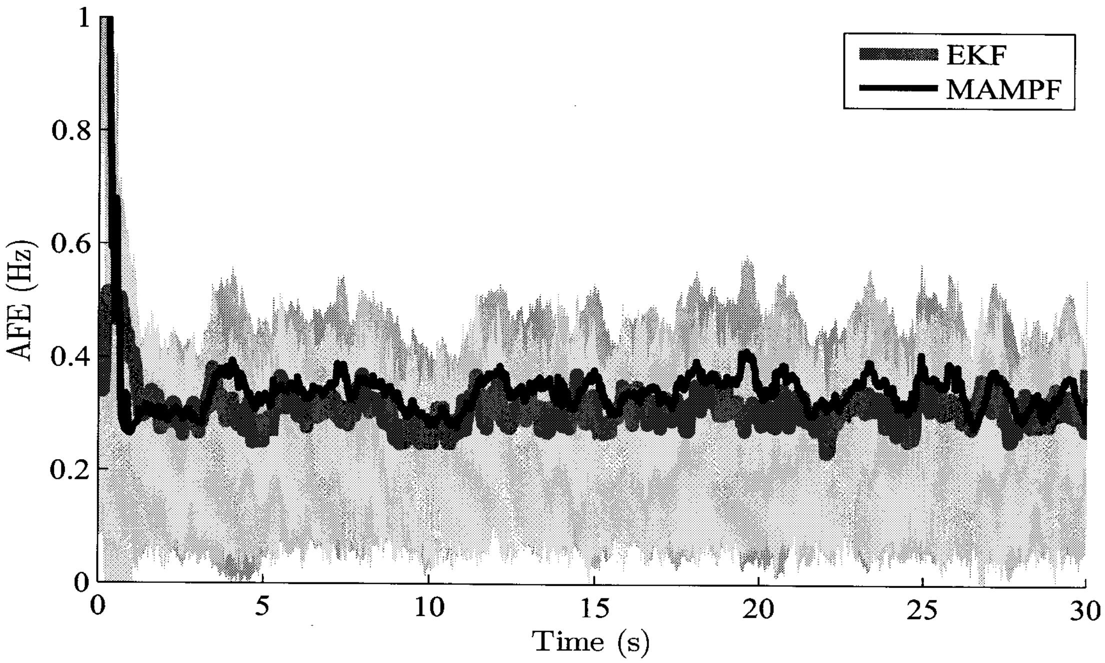


Figure 6.3: AFE versus time for synthetic binary spike trains whose tremor is continuous where the line represents the averaged AFE value and the light grey area represents one standard deviation range of AFE values over 100 simulations.

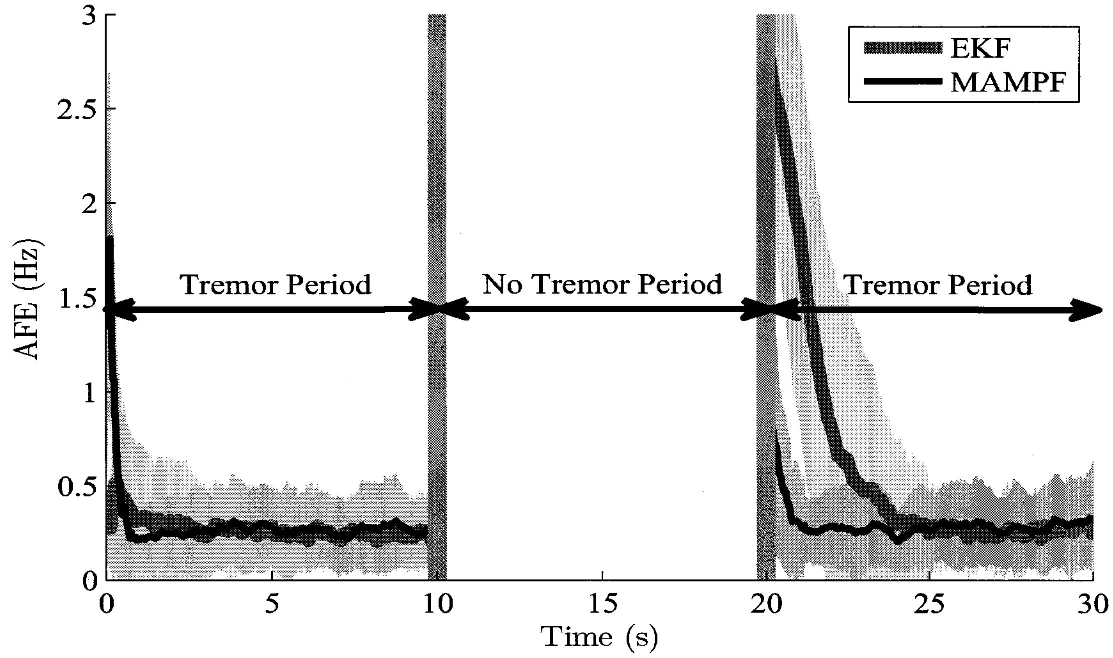


Figure 6.4: AFE versus time for synthetic binary spike trains whose tremor is intermittent where the line represents the averaged AFE value and the light grey area represents one standard deviation range of AFE values over 100 simulations.

trains, where tremor lasts only during the first and last 10 s periods. After the absence of tremorous activities between 10 s and 20 s, the MAM-PF based tremor tracker locks on to the true tremor frequency within 1 s on average, which is shown as the rapid decrease of the averaged AFE$_n$ after 20 s. However, the EKF based tremor tracker takes about 4 s to lock on to the true tremor frequency after 20 s. This simulation result demonstrates that the MAM-PF based tremor tracker can start tracking tremor quicklier than the EKF based tremor tracker, which is an important feature of a tremor tracker since tremorous activities are often intermittent. Fig. 6.5 depicts the averaged ITA, $\boldsymbol{p}_n$ in (6.2), of the 100 intermittent-tremor spike trains. During the first 10 s period, the averaged ITA of the MAM-PF based tremor tracker tends to be greater than that of the EKF based tremor tracker.

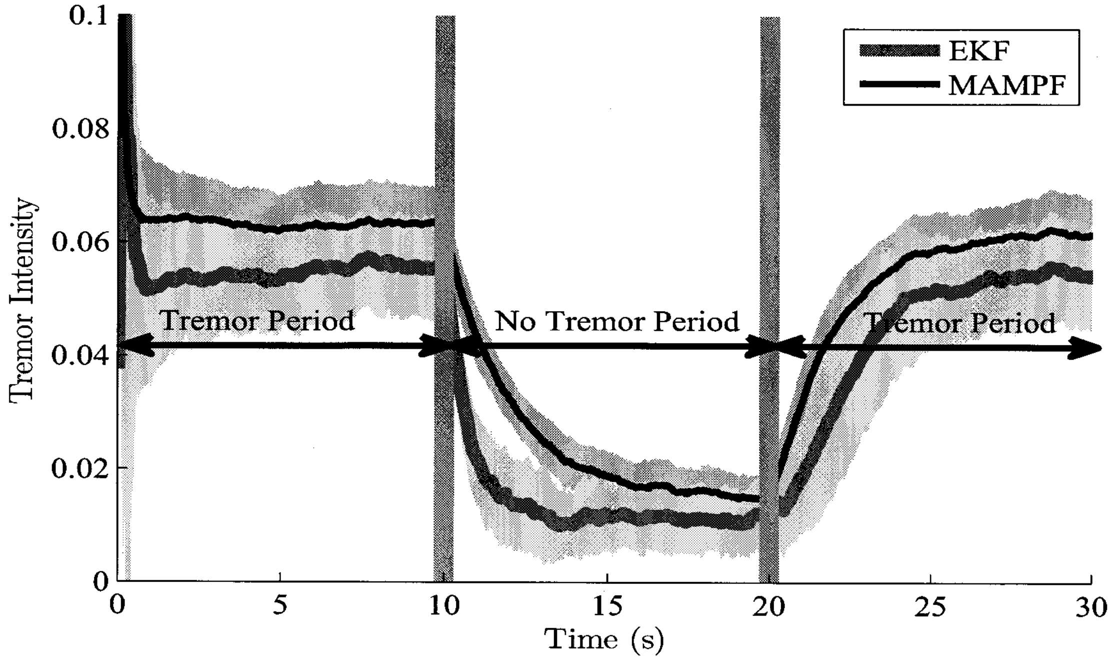


Figure 6.5: Tremor intensity versus time for synthetic binary spike trains whose tremor is intermittent.

When tremor is absent during the second 10 s period, the averaged ITAs of both tremor trackers rapidly decrease. After the absence of tremor, the averaged ITA of the MAM-PF based tremor tracker tends to increase more rapidly than that of the EKF based tremor tracker. This simulation result supports the fact that the MAM-PF based tremor tracker can *detect* the presence of tremorous activities in spike trains better than the EKF based tremor tracker.

Fig. 6.6 illustrates the estimated ITFs of the real binary spike train on top of its spectrogram. The MAM-PF based tremor tracker takes about 1 s to lock on to the tremor frequency while the EKF based tremor tracker takes about 7 s to do so. Figs. 6.7 and 6.8 depict the spectrograms of the prediction error e_n, which can

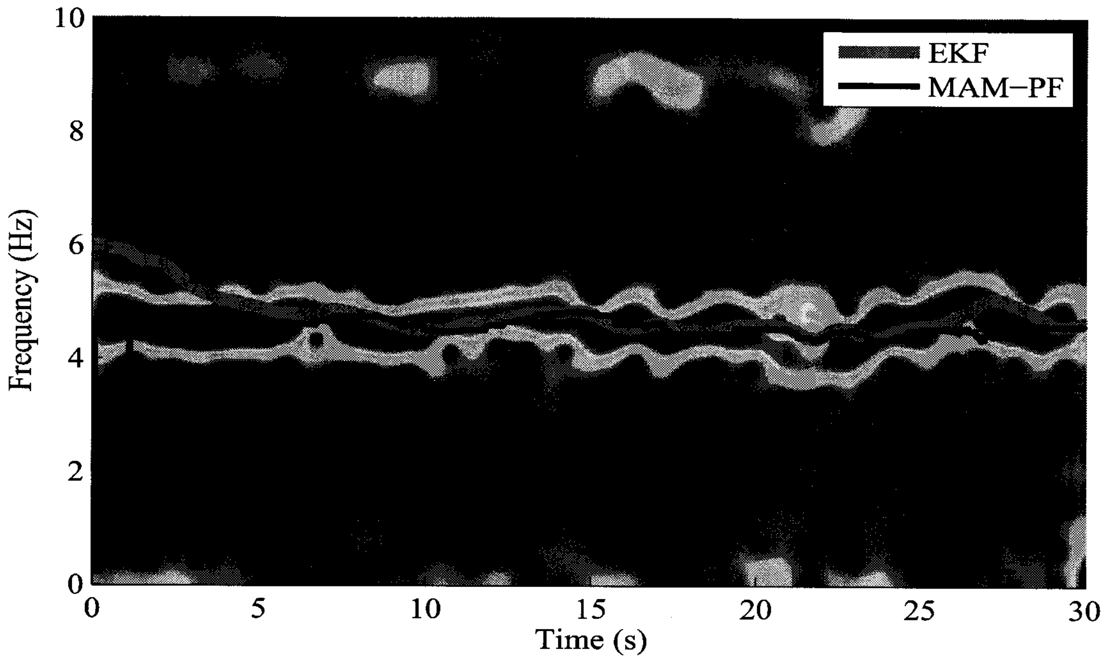


Figure 6.6: Estimated ITFs on top of the spectrogram of the real spike train from a subject with Parkinson's disease (PD).

be written as,

$$e_n = y_n - \hat{y}_{n|0:n-1} \tag{6.24}$$

where $\hat{y}_{n|0:n-1}$ represents the prediction estimate of the spike train y_n. Fig. 6.7 shows the 4.7 Hz residual tremor activity between 0 s and 6 s. On the other hand, Fig. 6.8 shows very little residual tremor activities.

6.5 Conclusion

The simulation results based on synthetic and real binary spike trains demonstrate that the MAM-PF based tremor tracker can attain lock faster than the EKF and that the performance is nearly as good. The ability to lock on to tremor quickly

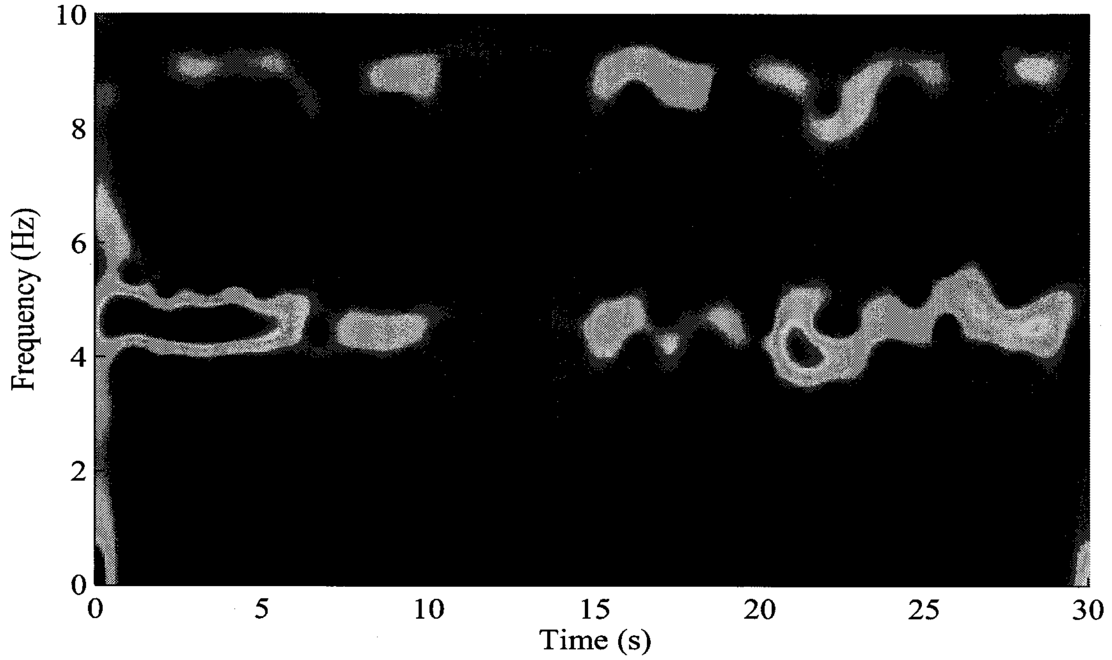


Figure 6.7: Spectrogram of a residual signal with the EKF based tremor tracker.

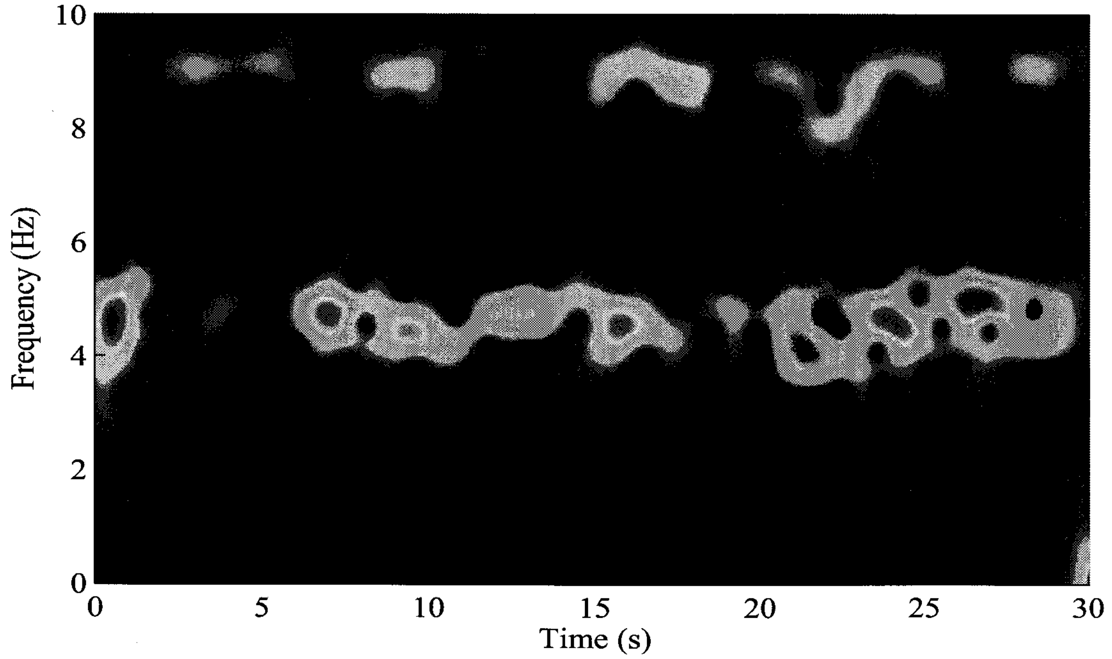


Figure 6.8: Spectrogram of a residual signal with the MAM-PF based tremor tracker.

is a very important feature of a good tremor tracker since tremorous activities in binary spike trains are often intermittent.

Chapter 7

Single Harmonic Set Tracking

This chapter discusses a practical application of the single harmonic set tracking case. The application is tracking heart rate in electrocardiograms (ECGs). The heart rate tracking method described in this chapter estimates the heart rate in ECG signals continuously on a sample-by-sample basis, which is different from other conventional heart rate estimation methods that yield one heart rate estimate per each cardiac cycle.

7.1 Introduction to the Clinical Problem

Heart rate is traditionally defined as the number of heart beats per minute. Although the heart rate can be measured wherever an artery's pulsation is transmitted to the body surface, a special method such as electrocardiography (ECG or EKG) is often used to ensure a more precise measure of the heart rate. ECG is a recording of the voltage between selectively placed electrodes on the skin, which reflects the activity of the heart muscle. Fig. 7.1 depicts a schematic representation of a typical ECG signal of one cardiac cycle. A typical ECG recording of a healthy cardiac cycle consists of a P wave, a QRS complex, and a T wave. The P wave corresponds to the depolarization of the right atrium. Atrial fibrillation results in the absence of the P wave. The QRS complex corresponds to the depolarization of the right and left ventricles. It has a much larger amplitude than the P wave since the mass of ventricle muscles is greater than that of atrium muscles. The careful analysis of the QRS complex can be used to diagnose several heart diseases such as myocardial infarction, cardiac arrhythmias, and ventricular hypertrophy. The T wave corresponds to the repolarization of the ventricles, which means the recovery of the muscles. The size and sign of the T wave can be used to diagnose coronary ischemia and hyperkalemia. As shown in Fig. 7.1, the R wave is typically the most noticeable wave, which is easy to detect. Therefore, the heart rate can be expressed as the number of R waves per minute in an ECG signal. However, the heart rate can be computed in relation of R-R intervals (or inter-beat intervals). The R-R interval refers to the time elapsing between two adjacent R waves in an ECG signal. The reciprocal of this R-R interval is often used as the heart rate, whose unit is in Hz.

The heart rate as the reciprocal of the R-R intervals does not remain constant, but rather fluctuates rhythmically for resting healthy subjects. Heart rate variability (HRV) refers to these rhythmic variations (or fluctuations) in the heart rate. For medical and diagnostic purposes, monitoring HRV accurately and reliably is an important but challenging task to perform. For resting healthy subjects, the heart rate fluctuates due to cardio-acceleration during inspiration and cardio-deceleration during expiration. Reduced HRV is used as a marker of reduced vagal activity. In brain electrophysiology, HRV has been recognized as a marker of normal functional states such as wakefulness or sleep, or of abnormal states such as epilepsy [122]. In the cardiovascular control system study, the analysis of HRV

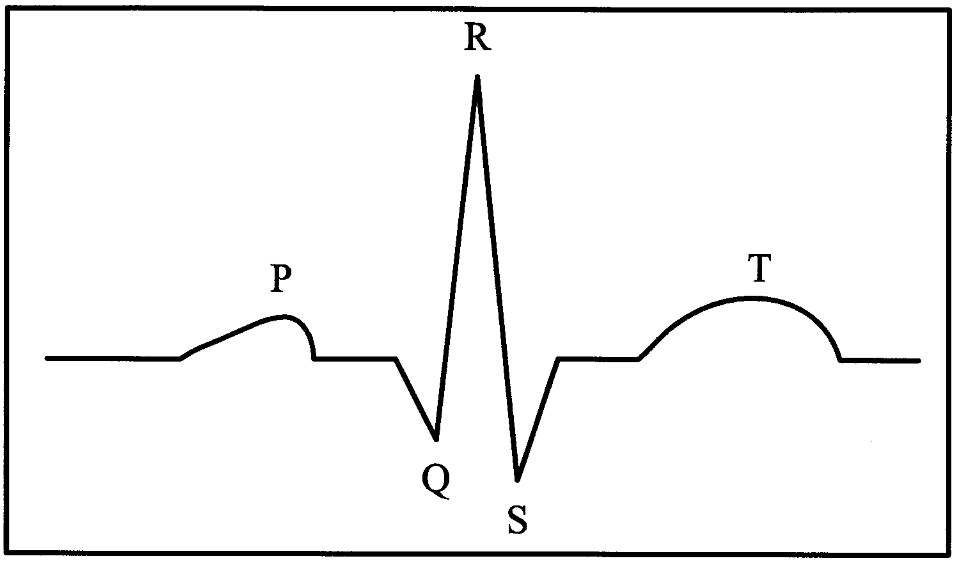


Figure 7.1: Schematic representation of normal ECG of one cardiac cycle.

provides a quantitative means of assessing the functioning of the cardiovascular control systems [123].

Accurate heart rate estimation is a prior step toward HRV analysis. Typically, the first step of heart rate estimation is detecting the R waves in the ECG signal. Then, instead of the ECG signal itself, a series of R-R interval values becomes the focus of analysis. The short-time power spectral analysis of this R-R interval series reveals a consistent high-frequency component ($\sim 0.25\,\mathrm{Hz}$), which is interpreted as a quantitative assessment of respiratory arrhythmia [124]. Other low-frequency components about 0.1 Hz and 0.03 Hz have also been noticed, which are known as Mayer waves [125].

The heart rate as the reciprocal of R-R intervals has a better time resolution than the heart rate as the number of heart beats per minute. However, its time resolution is limited by the cardiac cycle periods since only one heart rate value can be computed per each cardiac cycle. By adopting the state-space model framework, the heart rate can be computed on a sample-by-sample basis instead of a beat-by-beat basis. This high resolution heart rate is called an *instantaneous heart rate* and is expressed in units of Hz. The power spectral analysis of the instantaneous heart rate can reveal more precise and reliable frequency contents of the heart rate. A few research groups have proposed novel methods to estimate the instantaneous heart rate in ECG signals within state-space methods [126,127]. They utilized the Kalman filtering technique to estimate the instantaneous heart rate. This chapter proposes an accurate and robust instantaneous heart rate estimation method utilizing the Fast MAM-PF technique.

7.2 Summary of Prior Work

The heart rate, which is the reciprocal of R-R intervals, will be referred to as the traditional heart rate in order to distinguish it from the instantaneous heart rate. Computation of the traditional heart rate requires detection of the R waves in ECG signals. An algorithm to detect the R waves is called a *beat detection* algorithm. There are various beat detection algorithms available in the literature and commercially [107,124,128,129]. Pan *et al.* developed a real-time QRS detection algorithm of ECG signals, which is based upon digital analyses of slope, amplitude, and width [129]. Afonso *et al.* designed a digital signal processing algorithm to detect heartbeats of ECG signals [128]. Their algorithm incorporates a filter bank which decomposes the ECG signal into several sub-bands with uniform frequency bandwidths. Aboy *et al.* proposed a more general beat detection algorithm, which can be applied to pressure signals [107]. Their algorithm utilizes a filter bank with variable cutoff frequencies and spectral estimates of the heart rate. Overall beat detection algorithms are susceptive to noise and prone to any sudden changes in QRS morphology. More importantly, the R-R intervals (or inter-beat intervals) based on the beat detection result yield the traditional heart rate whose time resolution can be improved.

Ebrahim *et al.* presented a robust sensor fusion method to estimate the heart rate based on the Kalman filter technique in [127]. The main objective of their work was to combine heart rate measurements from multiple sensors to obtain an estimate of heart rate and a confidence value associated with every heart rate estimate. In detail, they evaluated three sensor measurements and the predicted estimate of the heart rate, which yield 16 possible hypotheses for the current state of the available data. A Kalman filter was applied to the most likely hypothesis to derived the *fused* estimate. McNames *et al.* proposed a novel state-space model of cardiovascular signals such as ECG and arterial blood pressure (ABP) and described the extended Kalman filter being applied to it for estimation of cardiovascular parameters such as the heart rate in [126]. Their results demonstrated that the extended Kalman filter with the proposed state-space model can be applied to several important applications such as tracking cardiovascular parameters of clinical interest from ABP and pulse oximetry (POX) signals, characterizing the intracranial pressure (ICP) pulsatile morphology, and estimating pulse pressure variation (PPV) from ABP signals obtained from mechanically ventilated swine. In my recent work [117], it was demonstrated that the particle filter is a more suitable method than the EKF for the purpose of heart rate tracking in ECG signals. Chapter 3 also shows that tracking the fundamental frequency (heart rate) of a multiharmonic signal (an ECG signal) can be framed as continuously estimating the posterior distribution of the fundamental frequency $p(\boldsymbol{f}_n|\boldsymbol{y}_{0:n})$, which is multi-modal and non-Gaussian. Since the Kalman filter assumes the unimodality of the posterior distribution, it is prone to estimation errors when applied to the multi-harmonic tracking application.

This chapter describes how to implement the ECG heart rate tracker utilizing the Fast MAM-PF and demonstrates the superior performance of the proposed heart rate tracker to the conventional MPF based heart rate tracker via Monte Carlo simulations.

7.3 Methodology

This section describes the state-space model of ECG signals and implementation of the Fast MAM-PF and MPF heart rate trackers.

7.3.1 Measurement Model

The ECG signal can be modeled as follows,

$$\boldsymbol{y}_n = \boldsymbol{a}_{0,n} + \left[\sum_{k=1}^{N_\mathrm{h}} \boldsymbol{a}_{1,k,n} \cos\left(k\boldsymbol{\theta}_n\right) + \boldsymbol{a}_{2,k,n} \sin\left(k\boldsymbol{\theta}_n\right) \right] + \boldsymbol{v}_n \tag{7.1}$$

where N_h is the number of harmonics or partials due to cardiac activities, $\boldsymbol{a}_{0,n}$ the slow signal trend, $\boldsymbol{a}_{1,k,n}$ and $\boldsymbol{a}_{2,k,n}$ the sinusoidal coefficients of the k^{th} cardiac partial, $\boldsymbol{\theta}_n$ the instantaneous angle of the heart rate $\boldsymbol{f}_n$, and $\boldsymbol{v}_n$ is white Gaussian noise with variance $r_{\boldsymbol{v}}$. This signal model is the same as the single harmonic set signal model shown in (4.10). The number of cardiac partials N_h may be known or need to be estimated. A simple method is using a conservative upper bound on N_h where the excessive partials' sinusoidal coefficients will be estimated as nearly zero at a cost of extra variance. If some pilot ECG signals are available, N_h can be estimated more accurately based on spectral analysis of the pilot signals.

7.3.2 Process Model

Given the instantaneous angle $\boldsymbol{\theta}_n$, the state-space model is a linear function of the signal trend $\boldsymbol{a}_{0,n}$ and the coefficients $\boldsymbol{a}_{1,k,n}$ and $\boldsymbol{a}_{2,k,n}$. These linear parameters can be estimated *optimally* using the Kalman filtering technique. These parameters can be modeled as a random walk model, which can be expressed as,

$$\boldsymbol{a}_{0,n+1} = \boldsymbol{a}_{0,n} + \boldsymbol{u}_{a,n} \tag{7.2}$$

$$\boldsymbol{a}_{1,k,n+1} = \boldsymbol{a}_{1,k,n} + \boldsymbol{u}_{a,n} \tag{7.3}$$

$$\boldsymbol{a}_{2,k,n+1} = \boldsymbol{a}_{2,k,n} + \boldsymbol{u}_{a,n} \tag{7.4}$$

where $\boldsymbol{u}_{a,n}$ represents a white Gaussian noise with variance $q_{\boldsymbol{a}}$.

The heart rate $\boldsymbol{f}_n$ changes slowly within a certain range. Therefore, the instantaneous angle $\boldsymbol{\theta}_n$ of the heart rate $\boldsymbol{f}_n$ can be modeled as follows,

$$\bar{\boldsymbol{f}}_{n+1} = g\left[\bar{\boldsymbol{f}}_n + \boldsymbol{u}_{\bar{f},n}\right] \tag{7.5}$$

$$\boldsymbol{f}_{n+1} = \bar{\boldsymbol{f}}_n + \alpha\left(\boldsymbol{f}_n - \bar{\boldsymbol{f}}_n\right) + \boldsymbol{u}_{f,n} \tag{7.6}$$

$$\boldsymbol{\theta}_{n+1} = \boldsymbol{\theta}_n + 2\pi T_\mathrm{s}\boldsymbol{f}_n \tag{7.7}$$

where $\bar{\boldsymbol{f}}_n$ is the mean heart rate, T_s is the sampling interval, α is an autoregressive (AR) coefficient, and $g[\cdot]$ represents a nonlinear reflecting function to account for the limited heart rate range. The nonlinear reflecting function can be expressed as,

$$g[f] = \begin{cases} f_{\max} - (f - f_{\max}) & f_{\max} < f \\ f & f_{\min} < f \leq f_{\max} \\ f_{\min} + (f_{\min} - f) & f \leq f_{\min} \end{cases} \tag{7.8}$$

where $f_{\min}$ and $f_{\max}$ represent the possible extreme heart rate values. Now, The state vector $\boldsymbol{x}_n$ can be written as,

$$\begin{aligned} \boldsymbol{x}_n &= \begin{bmatrix} \boldsymbol{x}_{1,n} & \boldsymbol{x}_{2,n} & \boldsymbol{x}_{3,n} & \boldsymbol{x}_{4,n} & \cdots & \boldsymbol{x}_{2N_\mathrm{h}+3,n} \end{bmatrix}^\mathrm{T} \\ &= \begin{bmatrix} \boldsymbol{\theta}_n & \boldsymbol{f}_n & \boldsymbol{a}_{0,n} & \boldsymbol{a}_{1,1,n} & \cdots & \boldsymbol{a}_{2,N_\mathrm{h},n} \end{bmatrix}^\mathrm{T} \end{aligned} \tag{7.9}$$

which can be partitioned into the nonlinear and linear portions as follows,

$$\boldsymbol{x}_n^\mathrm{N} = \begin{bmatrix} \boldsymbol{\theta}_n & \boldsymbol{f}_n \end{bmatrix}^T \tag{7.10}$$

$$\boldsymbol{x}_n^\mathrm{L} = \begin{bmatrix} \boldsymbol{a}_{0,n} & \boldsymbol{a}_{1,1,n} & \cdots & \boldsymbol{a}_{2,N_\mathrm{h},n} \end{bmatrix}^T . \tag{7.11}$$

7.3.3 Fast MAM-PF Heart Rate Tracker

The Fast MAM-PF heart rate tracker utilizes the Fast MAM-PF algorithm to continuously estimate the heart rate and morphology of an ECG signal. Since the state vector $\boldsymbol{x}_n$ of the ECG signal state-space model is partitioned into the nonlinear and linear portions, the Fast MAM-PF algorithm can be directly applied to the state-space model to implement the Fast MAM-PF heart rate tracker. The details of the Fast MAM-PF recursions are listed in Algorithm 5 in Chapter 5. Table 7.1 summarizes user-specified parameter values for the Fast MAM-PF heart

Table 7.1: Summary of user-specified parameters for the filters (MAM-PF/MPF).

Name	Symbol	Value
Number of particles	N_p	150/180
Resampling threshold	N_t	$N_\mathrm{p}/100$
Number of harmonics	N_h	10
Frequency coefficient	α	0.999
Covariance coefficient	β	0.999
Max. mean heart rate	$\bar{f}_\mathrm{max}$	2 Hz
Min. mean heart rate	$\bar{f}_\mathrm{min}$	1 Hz
Heart rate variance	$\boldsymbol{q}_f$	1e-4
Amplitude variance	$\boldsymbol{q}_a$	var($\boldsymbol{y}$)/100
Measurement variance	r	var($\boldsymbol{y}$)/10
Initial phase	$\boldsymbol{\theta}_0$	0
Initial mean heart rate	$\bar{\boldsymbol{f}}_0$	1 Hz
Initial heart rate	$\boldsymbol{f}_0$	1 Hz
Initial amplitudes	$\boldsymbol{a}_0$	0.01

rate tracker.

7.3.4 MPF Heart Rate Tracker

The MPF heart rate tracker utilizes the MPF algorithm to track the heart rate and morphology of an ECG signal. Like the Fast MAM-PF algorithm, the MPF algorithm can also be applied to the ECG state-space model without any modification. The details of the MPF recursions are listed in Algorithm 3 in Chapter 5. Table 7.1 summarizes user-specified parameter values for the MPF heart rate tracker.

7.4 Performance Comparison

7.4.1 Equalizing Computational Load

The Fast MAM-PF requires more computation than the MPF due to the Viterbi algorithm to search for the most probable past path of each particle. In order to make a fair comparison between these two filters, I first ran the Fast MAM-PF with 150 particles and measured the simulation duration. Then, I ran the MPF with various numbers of particles to find the number of particles with which the MPF takes the same simulation duration. With 180 particles the simulation time of the MPF was equal to that of the Fast MAM-PF. This process approximately equalizes the processing time required for each filter for the ECG heart rate tracking application discussed in this chapter.

7.4.2 Synthetic ECG Signals

The true heart rate of ECG signals can only be estimated. However, it is necessary to have ECG signals with a known true heart rate to accurately compare the performance of two heart rate trackers in Monte Carlo simulations. For the purpose of performance comparison of two trackers, two types of synthetic ECG signals were generated. The first type of synthetic ECG signals mimics realistic ECG signals whose heart rate and morphology change slowly over time. They were generated based on the measurement model shown in (7.1) and process models shown in (7.2)–(7.7). Table 7.2 summarizes the further details of the synthetic ECG signal generator.

The second type of synthetic ECG signals mimics a hypothetic ECG signal

Table 7.2: List of user-specified parameters to generate synthetic ECG signals.

Name	Symbol	Value
Number of harmonics	N_h	10
Sampling frequency	f_s	50 Hz
Signal duration	l	5 min
Number of samples	N_T	1.5e4
Signal-to-noise ratio	SNR	10 dB
Heart rate coefficient	α	0.999
Amplitude variance	$q_{\boldsymbol{a}}$	0.1
Heart rate variance	$q_{\boldsymbol{f}}$	1e-4
Initial mean heart rate	$\bar{\boldsymbol{f}}_0$	1.5 Hz
Initial heart rate	$\boldsymbol{f}_0$	1.5 Hz
Initial amplitude	$\boldsymbol{a}_0$	1

whose heart rate abruptly changes from 2 Hz to 4 Hz and whose cardiac partials' amplitudes remain constant. The purpose of generating this unrealistic ECG signal was to contrast the ability of two heart rate trackers to adapt to rapid changes in the heart rate. Two trackers were applied to the hypothetical ECG signal 100 times and the heart rate estimates were averaged over the simulations.

7.4.3 Real ECG Signal

Both heart rate trackers were applied to a real electrocardiogram (ECG) signal sampled at 500 Hz containing a high level of noise. The goal of this experiment was to compare the ability of the trackers to retain track on a noisy signal amid realistic signal artifacts such as signal dropouts and medical intervention. The ECG signal chosen here is one of the noisiest signals in the MIMIC database [130] on PhysioNet [131]. The duration of the ECG signal is 30 min.

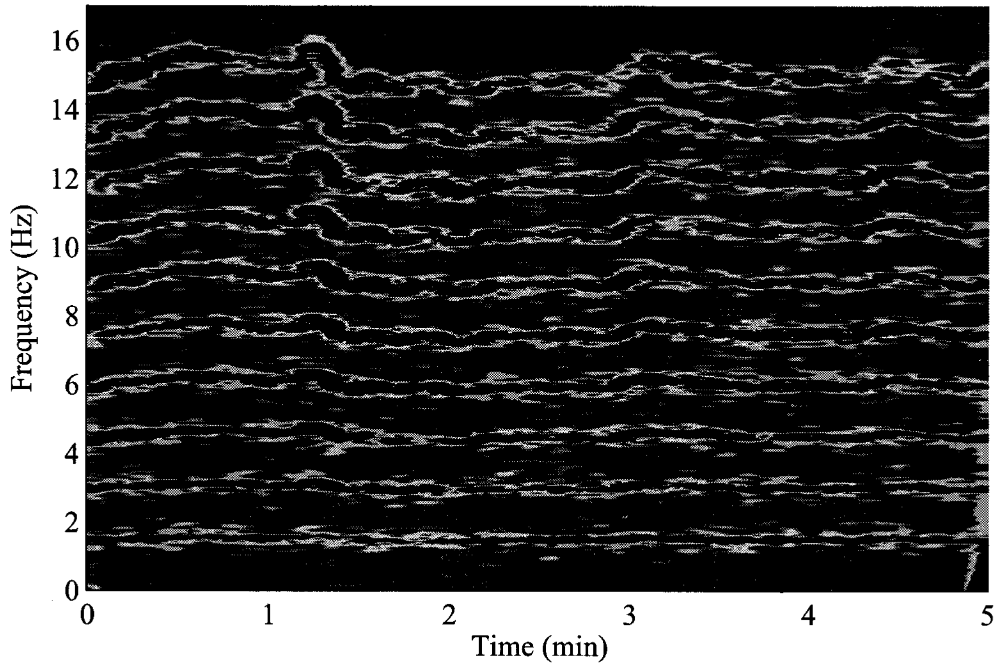
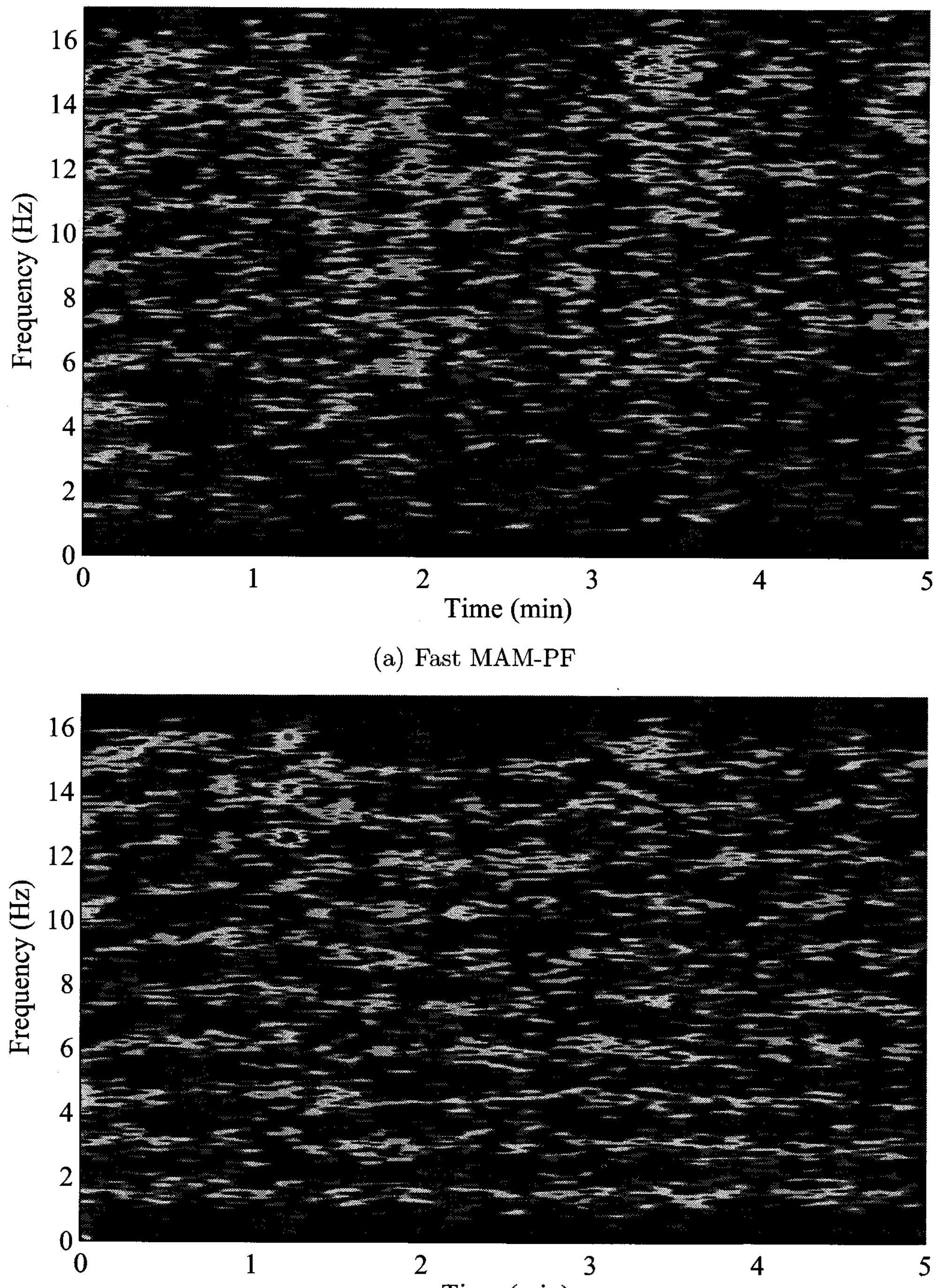


Figure 7.2: Spectrogram of an example synthetic ECG signal with 10 harmonic partials.

7.5 Results and Discussion

7.5.1 Synthetic Signal

Fig. 7.2 shows the spectrogram of an exemplary synthetic ECG signal. The total number of partials was 10 and the mean value of the heart rate was 1.5 Hz. Two plots in Fig. 7.3 illustrates the spectrograms of signal estimation residuals using the MAM-PF (top) and MPF (bottom) trackers, respectively. In the residual spectrograms there were not any significant multi-harmonic structures due to cardiac activities, which indicates that both trackers were able to track the heart rate and its harmonic partials in the synthetic ECG signal almost perfectly. Since the signal-to-noise ratio (SNR) of the synthetic ECG signal was 10 dB and noise due to signal drops or medical intervention was not simulated, the MPF tracker



(a) Fast MAM-PF

(b) MPF

Figure 7.3: Spectrograms of synthetic ECG signal estimation residuals using (a) the Fast MAM-PF and (b) the MPF.

performed as well as the MAM-PF tracker. Fig. 7.4 illustrates two trackers' averaged absolute frequency error (AFE_n) over 100 simulations. The thick light grey line represents the averaged AFE_n of the MPF tracker while the thin black line represents that of the MAM-PF tracker. The light grey area around the averaged AFE_n shows the standard deviation of the AFE_n over 100 simulations. Fig. 7.4 demonstrates that there was not any significant performance difference between the MAM-PF and MPF trackers since the synthetic ECG signals simulated almost noise free ECG signals.

Fig. 7.5 illustrates the superior performance of the MAM-PF heart rate tracker over the MPF heart rate tracker when the heart rate abruptly changes. The abrupt change in the heart rate from 2 Hz to 4 Hz results in the sudden change in the posterior distribution of the heart rate, that is $p(\boldsymbol{f}_n|\boldsymbol{y}_{0:n})$. The MPF cannot handle this situation since the MPF's particles are densely populated only in a small portion of the state space after resampling. In other words, the MPF suffers from the sample impoverishment issue. The top plot in Fig. 7.5 shows the spectrogram of a synthetic ECG signal whose heart rate abruptly changes from 2 Hz to 4 Hz after 60 s. The bottom plot in Fig. 7.5 depicts the averaged heart rate estimates over 100 simulations of the MAM-PF (dark grey) and MPF (light grey) heart rate trackers. Since the Fast MAM-PF does not rely on the resampling scheme and is therefore not prone to the effects of sample impoverishment, the Fast MAM-PF heart rate tracker was able to regain its track of the true heart rate 4 Hz while the MPF failed completely to regain track. The MPF tracker kept tracking the subharmonic 2 Hz of the true heart rate 4 Hz after 60 s, when the heart rate changed from 2 Hz to 4 Hz.

Four plots in Fig. 7.6 depict the distributions of the particles of the Fast

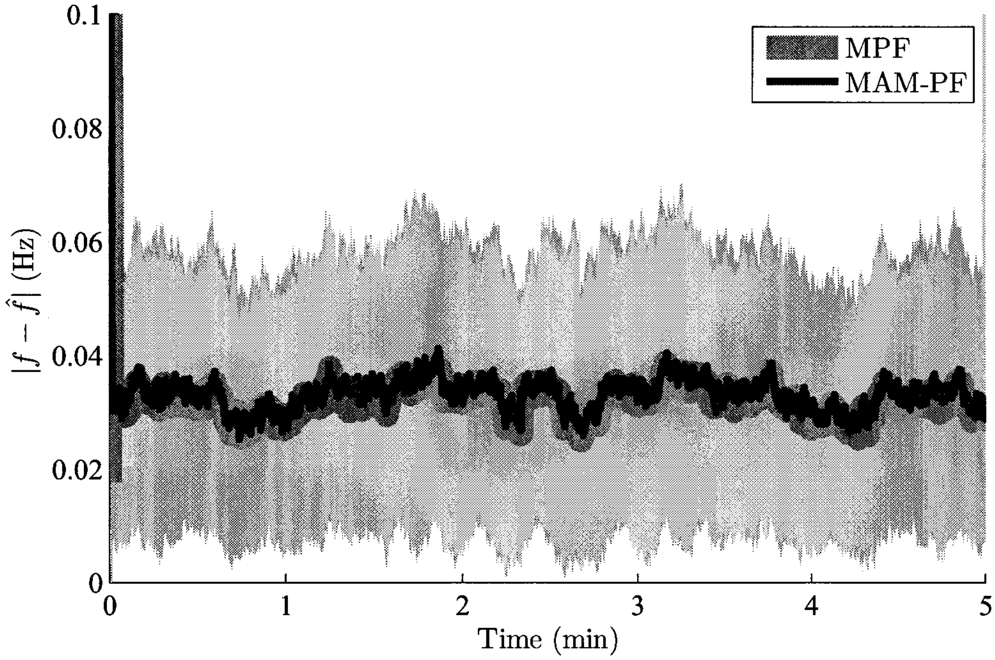


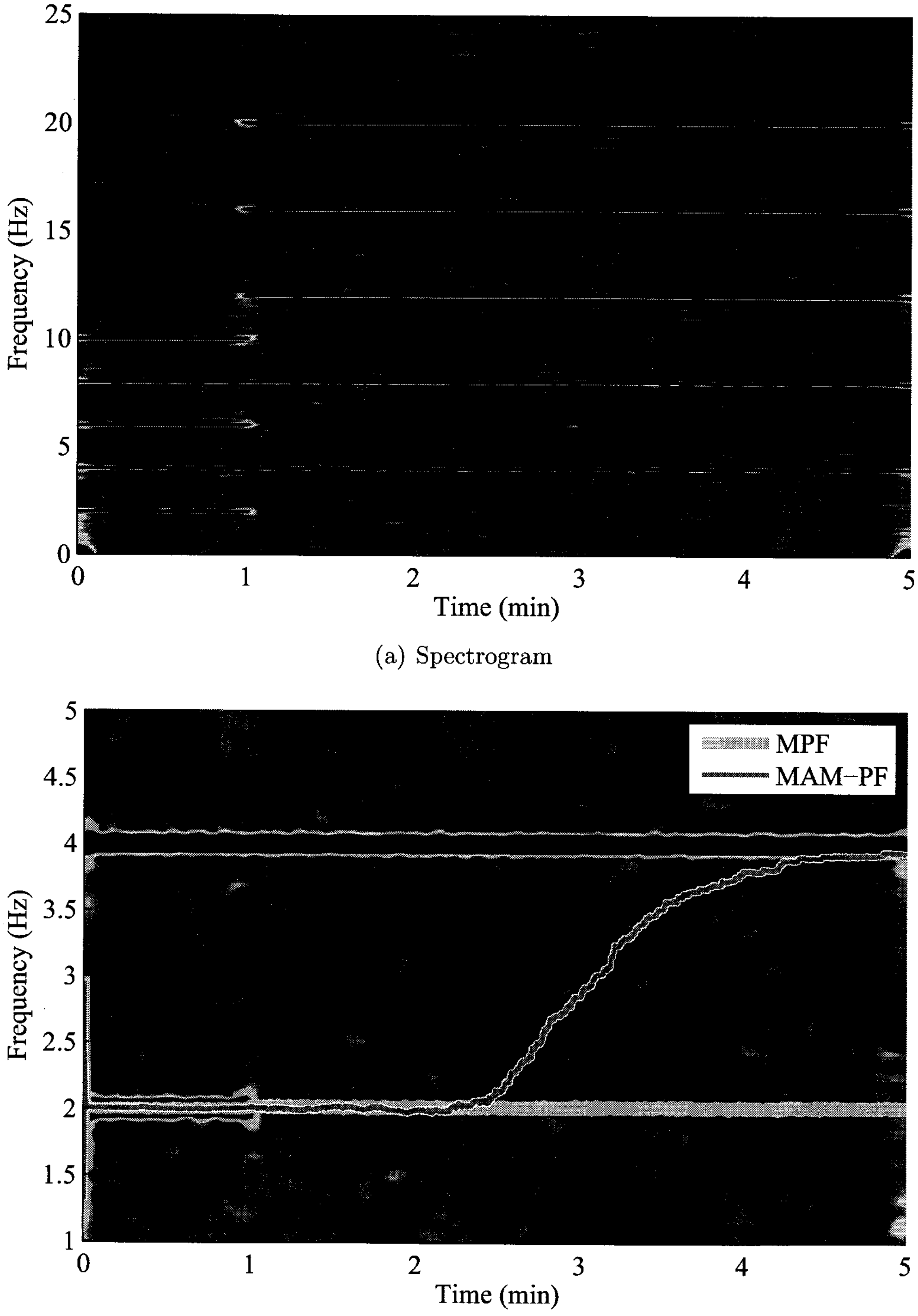
Figure 7.4: Averaged absolute frequency error (AFE_n) using the MAM-PF and MPF where the light grey area represents the standard deviation of AFE_n over 100 simulations.

MAM-PF and MPF trackers at time 60 s (left column) and 300 s (right column). The green dots in two top plots represent the Fast MAM-PF's particles α while the red dots in two bottom plots represent the MPF's particles $\boldsymbol{w}$. Due to particle resampling the MPF's particles $\boldsymbol{w}$ were densely distributed around the true heart rate 2 Hz at 60 s. Although the true heart rate became 4 Hz after 60 s, all the particles $\boldsymbol{w}$ were still distributed around 2 Hz and did not move to the true heart rate 4 Hz until the end of simulation time 300 s. In contrast, the Fast MAM-PF's particles α were almost evenly distributed over the entire heart rate range, which was 1 Hz–5 Hz in this simulation study. At 60 s only a few particles around the true heart rate 2 Hz had large values. At 300 s other particles around the true heart rate 4 Hz had large values. Therefore, the Fast MAM-PF tracker was able to track the true heart rate in spite of its abrupt change at 60 s.

7.5.2 Real Signal

The top plot in Fig. 7.7 illustrates the spectrogram of the real ECG signal. It shows that the ECG signal was corrupted by noise such as signal drops and medical intervention. The bottom plot in Fig. 7.7 illustrates the averaged heart rate estimates (MPF: white, MAM-PF: grey) over 100 simulations on top of the spectrogram of the real ECG signal. Although the ECG signal is quite noisy between 9 min and 17 min, the averaged heart rate estimate of the MAM-PF heart rate tracker remained on top of the true heart rate (thick red band at around 1.5 Hz) during the entire duration 30 min of the ECG signal. This simulation result indicates that the MAM-PF heart rate tracker never failed to track the true heart rate of the ECG signal over the 100 simulations. In contrast, the averaged heart rate estimate of the MPF heart rate tracker deviated from the true heart rate after about 12 min. It is important to understand that the mismatch between the true heart rate and the averaged heart rate estimate does not indicate that the MPF heart rate tracker lost its track of the true heart rate all the time over the 100 simulations. It rather indicates that the MPF tracker occasionally failed to track the true heart rate over the simulations.

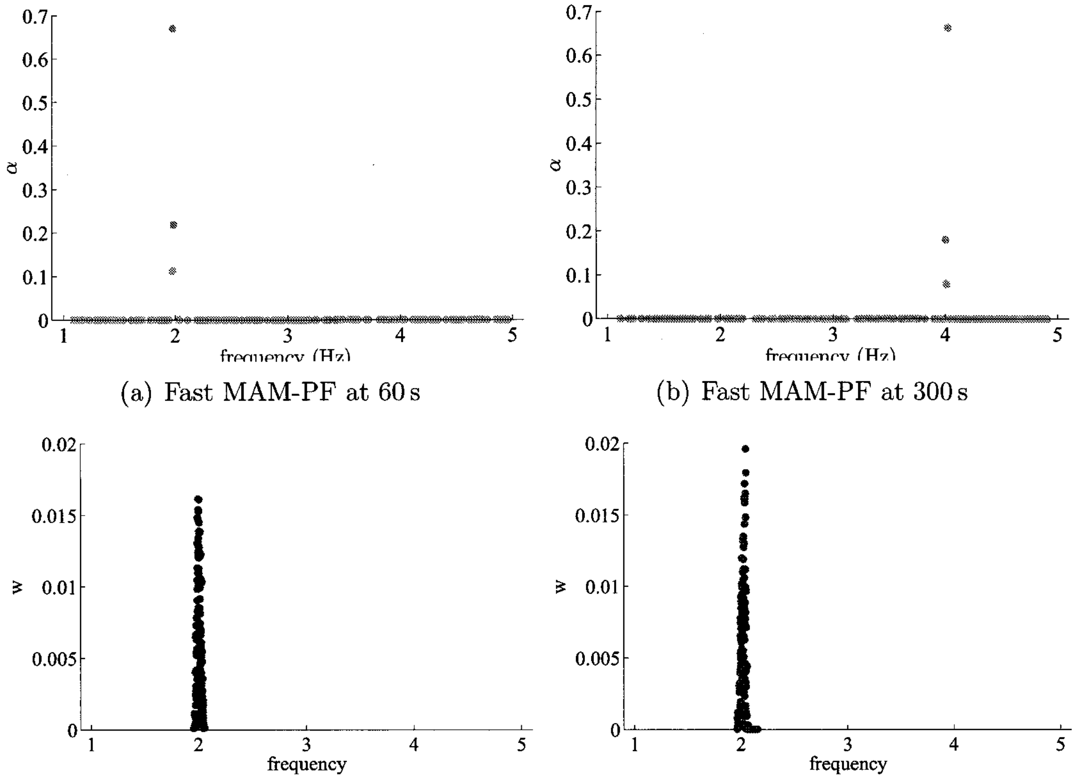
Two plots in Fig. 7.8 depict exemplary cases among the 100 simulations. The top plot in Fig. 7.8 shows the spectrogram of the real ECG signal estimation residual using the MPF heart rate tracker while the bottom plot illustrates that using the MAM-PF heart rate tracker. The estimation residual of the MAM-PF heart rate tracker did not show any harmonic structures due to the cardiac activity,



(a) Spectrogram

(b) Averaged heart rate estimates

Figure 7.5: (a) Spectrogram of the synthetic ECG signal with an abrupt heart rate change (b) Averaged heart rate estimates (MPF: light grey, MAM-PF: dark grey).



(a) Fast MAM-PF at 60 s

(b) Fast MAM-PF at 300 s

(c) MPF at 60 s

(d) MPF at 300 s

Figure 7.6: Distributions of particles (α) of the Fast MAM-PF and those ($\boldsymbol{w}$) of the MPF (a) α versus frequency at 60 s. (b) α versus frequency at 300 s. (c) $\boldsymbol{w}$ versus frequency at 60 s. (d) $\boldsymbol{w}$ versus frequency at 300 s.

which means that the MAM-PF tracker could estimate the ECG signal and the heart rate very well. Between 12 min and 14 min the MAM-PF tracker could not estimate the true heart rate correctly due to the high noise level. After 14 min, however, the tracker regained its track as the noise level decreased. Contrarily, the spectrogram of the estimation residual of the MPF heart rate tracker showed the first cardiac partial after 12 min until the end of the signal duration. This was because the MPF heart rate tracker lost its track of the true heart rate due to the high noise level and never regains its track. After 22 min the second cardiac partial was eliminated although the first cardiac partial was not. This situation occurred because the MPF heart rate tracker tracked the true second cardiac partial at around 3 Hz as if it were the third cardiac partial. In other words, the heart rate estimate of the MPF tracker was about 1 Hz, which is a sub-harmonic of the true heart rate 1.5 Hz.

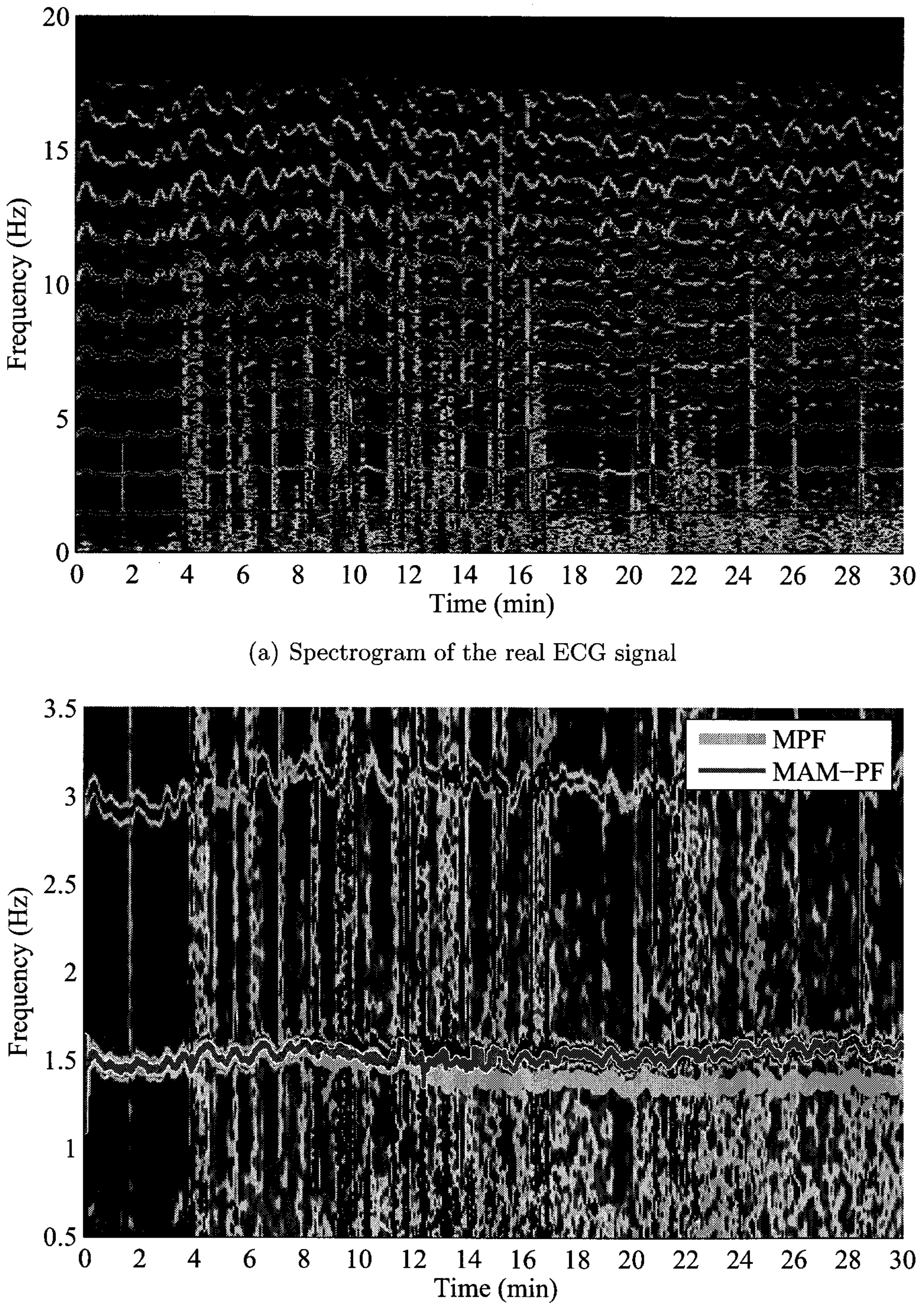
7.6 Conclusion

Two heart rate trackers have been implemented based on the conventional MPF and proposed Fast MAM-PF. The conventional MPF suffers from sample impoverishment resulting from the particle resampling scheme. However, the Fast MAM-PF is not prone to the effects of sample impoverishment since it relies on the Viterbi algorithm to obtain the MAP state estimate.

The simulation results based on synthetic and real ECG signals demonstrate that the Fast MAM-PF is a more suitable method than the MPF for the heart rate tracking in ECG signals. Especially when the heart rate changes abruptly or the background noise level increases due to signal drops and medical intervention, the Fast MAM-PF heart rate tracker outperforms the MPF heart rate tracker. Since the computational load of two methods has been equalized as explained in 7.4.1, the superior performance of the Fast MAM-PF heart rate tracker over the MPF heart rate tracker is solely attributable to the Fast MAM-PF algorithm's excellency in handling rapid changes in the posterior distribution $p(\boldsymbol{f}_n|\boldsymbol{y}_{0:n})$.

Chapter 6 compares the MAM-PF against the EKF because the main goal was to demonstrate that the MAM-PF can outperform the EKF even when the posterior distribution of interest is uni-modal. However, Chapter 7 compares the MAM-PF against the conventional MPF instead of the EKF first because the posterior distribution of interest is multi-modal, which causes the EKF to perform very poorly, and secondly because the main goal was to demonstrate the superior performance of the MAM-PF over the MPF which is one of the conventional PFs applicable to the nonlinear state-space model of our interest.

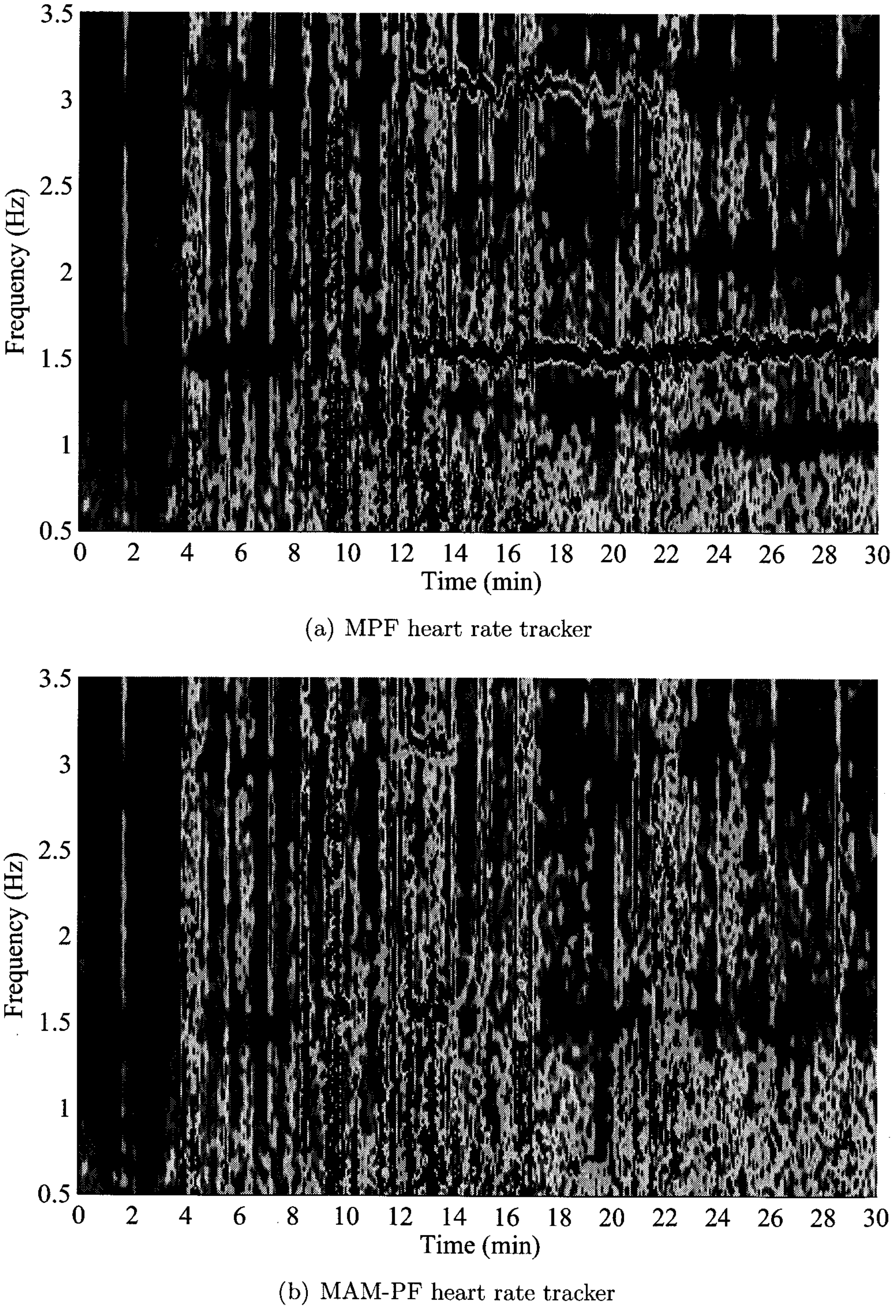
Tracking nonlinear state-space model parameters is not a trivial task. Specially when the posterior distribution of nonlinear parameters is multi-modal, it becomes more challenging to obtain accurate estimates of those parameters. Although the EKF requires the least computation load, its performance may not be satisfactory in estimating those nonlinear parameters because it approximates the non-Gaussian or multi-modal posterior distribution of the parameters as a simple Gaussian distribution. Using the MPF one can obtain the mean estimate of the posterior distribution that converges to the true mean value as the number of particles used increases. However, when the posterior distribution of interest is multi-modal, the mean value may not be an informative quantity to summarize the posterior distribution no matter how accurate its estimate is. The MAM-PF is one of many variants of the conventional PF technique just as the MPF is. However, it computes the MAP estimates of nonlinear parameters, which minimize the *most probable error*. The MAP estimate of a nonlinear parameter such as the fundamental frequency of a multi-harmonic signal can be very different from the mean estimate due to the multi-modality of the posterior distribution. Fig. 7.5 clearly illustrates a case where the mean estimate of a nonlinear parameter can be completely misleading. However, obtaining the MAP estimates of nonlinear parameters requires heavy computational load. For example, the Optimal MAM-PF can require a hundred times of the computational burden than the MPF as shown in Fig. 5.2. The Fast MAM-PF technique was devised to ease the heavy computational burden so that one can obtain the MAP estimates of nonlinear parameters at a reasonable cost.



(a) Spectrogram of the real ECG signal

(b) Averaged heart rate estimates (MPF: white, MAM-PF: grey) on top of the spectrogram of the real ECG signal

Figure 7.7: (a) Spectrogram of the real ECG signal whose duration is 30 min (b) Averaged heart rate estimates (MPF: white, MAM-PF: grey) over 100 simulations.



(a) MPF heart rate tracker

(b) MAM-PF heart rate tracker

Figure 7.8: (a) Spectrogram of ECG signal estimation residuals using the MPF tracker (b) Spectrogram of ECG signal estimation residuals using the Fast MAM-PF tracker.

Chapter 8

Amplitude Modulated Harmonic Set Tracking

This chapter discusses a practical application of the amplitude modulated harmonic set tracking case. The application is tracking the pulse pressure variation (PPV) index of arterial blood pressure (ABP) signals under mechanical ventilation.

8.1 Introduction to the Clinical Problem

In many critical care settings clinicians must decide whether patients should be given intravenous fluid boluses and other therapies to improve perfusion. Excessive fluid can be damaging by impairing lung function when it decreases oxygen delivery to tissues and contributes to organ failure. Insufficient fluid can result in insufficient tissue perfusion which can also contribute to organ failure. Determining the best course of fluid therapy for a patient is difficult and clinicians have few clinical signs to guide them.

The pulse pressure variation (PPV) index is a measure of the respiratory effect on the variation of systemic arterial blood pressure in patients receiving full mechanical ventilation. It is a dynamic predictor of increases in cardiac output due to an infusion of fluid. Numerous studies have demonstrated that PPV is one of the most sensitive and specific predictors of fluid responsiveness. Specifically, PPV has been shown to be useful as a dynamic indicator to guide fluid therapy in different patient populations receiving mechanical ventilation [106]. For instance, PPV was found to exhibit better performance as a predictor of fluid responsiveness in patients before off-pump coronary artery bypass grafting than standard static pre-load indices [132]. PPV has also been shown to be useful for predicting and assessing the hemodynamic effects of volume expansion and a reliable predictor of fluid responsiveness in mechanically ventilated patients with acute circulatory failure related to sepsis [133, 134]. Another study concluded that PPV can be used to predict whether or not volume expansion will increase cardiac output in postoperative patients who have undergone coronary artery bypass grafting [135]. A critical review of studies investigating predictive factors of fluid responsiveness in intensive care unit patients concluded that PPV and other dynamic parameters should be used preferentially to static parameters to predict fluid responsiveness [136].

The standard method for calculating PPV often requires simultaneous recording of arterial and airway pressure. Pulse pressure (PP) is manually calculated on a beat–to–beat basis as the difference between systolic and diastolic arterial pressure. Maximal PP ($\mathrm{PP}_{\mathrm{max}}$) and minimal PP ($\mathrm{PP}_{\mathrm{min}}$) are calculated over a single respiratory cycle, which is determined from the airway pressure signal. Pulse pressure variations PPV are calculated in terms of $\mathrm{PP}_{\mathrm{max}}$ and $\mathrm{PP}_{\mathrm{min}}$ and expressed as a percentage,

$$\Delta\mathrm{PP}(\%) = 100 \times \frac{\mathrm{PP}_{\mathrm{max}} - \mathrm{PP}_{\mathrm{min}}}{(\mathrm{PP}_{\mathrm{max}} + \mathrm{PP}_{\mathrm{min}})/2} \tag{8.1}$$

8.2 Summary of Prior Work

There are few publicly available algorithms to automatically estimate PPV accurately and reliably. Aboy *et al.* have previously described a beat detection-based PPV algorithm [137]. This algorithm was made publicly available by the authors and it has been adopted by Philips Medical Systems. Currently, their PPV index is displayed in real-time on the Philips Intellivue MP70 monitors (Intellivue MP70, Philips Medical Systems). Its accuracy against the current standard obtained by manual annotations were assessed by Cannesson [138]. A limitation of their algorithm adopted by Philips in Intellivue MP70 monitors is that it may not work adequately in regions of abrupt hemodynamic changes and this may limit its applicability in operating room environments. An enhanced automatic algorithm for PPV estimation during regions of abrupt hemodynamic changes was recently proposed to address this potential limitation [139].

The subsequent sections provide a detailed description of the proposed PPV index tracking algorithm utilizing the Fast MAM-PF and demonstrates its performance on real ABP signals from the Massachusetts General Hospital (MGH) Waveform Database [130] on PhysioNet [131]. Contrary to two empirical algorithms described in [137, 139], the proposed algorithm is the first PPV method based on a statistical state-space model for arterial blood pressure signals and optimal estimation methods. This powerful framework eliminates the need for the use of heuristics and empirical parameter determination.

8.3 Algorithm Description

This section describes the state-space model of ABP signals under full mechanical ventilation and provides the details of the automated PPV index tracking algorithm.

8.3.1 Measurement Model

The measurement model of the ABP signal under mechanical ventilation is the same as those in (4.11)–(4.14). It can be reexpressed as follows,

$$\boldsymbol{y}_n = \boldsymbol{\gamma}_n + \boldsymbol{\mu}_n + \boldsymbol{v}_n = \boldsymbol{\gamma}_n + \sum_{k=1}^{N_{\mathrm{h}}^{\mathrm{c}}} \boldsymbol{\rho}_{k,n} \boldsymbol{\kappa}_{k,n} + \boldsymbol{v}_n \tag{8.2}$$

$$\boldsymbol{\gamma}_n = \sum_{k=1}^{N_{\mathrm{h}}^{\mathrm{r}}} \boldsymbol{r}_{1,k,n} \cos(k\boldsymbol{\theta}_n^{\mathrm{r}}) + \boldsymbol{r}_{2,k,n} \sin(k\boldsymbol{\theta}_n^{\mathrm{r}}) \tag{8.3}$$

$$\boldsymbol{\kappa}_{k,n} = \boldsymbol{c}_{1,k,n} \cos(k\boldsymbol{\theta}_n^{\mathrm{c}}) + \boldsymbol{c}_{2,k,n} \sin(k\boldsymbol{\theta}_n^{\mathrm{c}}) \tag{8.4}$$

$$\boldsymbol{\rho}_{k,n} = 1 + \sum_{j=1}^{N_{\mathrm{h}}^{\mathrm{r}}} \boldsymbol{m}_{1,k,j,n} \cos(j\boldsymbol{\theta}_n^{\mathrm{r}}) + \boldsymbol{m}_{2,k,j,n} \sin(j\boldsymbol{\theta}_n^{\mathrm{r}}) \tag{8.5}$$

where $\boldsymbol{\gamma}_n$ is the respiratory signal, $\boldsymbol{\mu}_n$ the amplitude-modulated cardiac signal, $\boldsymbol{\rho}_{k,n}$ the amplitude modulation factor of the k^{th} cardiac harmonic partial, $\boldsymbol{\kappa}_{k,n}$ the k^{th} cardiac harmonic signal, $\boldsymbol{\theta}_n^{\mathrm{r}}$ the instantaneous respiratory angle, $\boldsymbol{\theta}_n^{\mathrm{c}}$ the instantaneous cardiac angle, $\boldsymbol{f}_n^{\mathrm{r}}$ the instantaneous respiratory rate, $N_{\mathrm{h}}^{\mathrm{r}}$ the number of respiratory partials, $N_{\mathrm{h}}^{\mathrm{c}}$ the number of cardiac partials, and $\boldsymbol{v}_n$ the white Gaussian measurement noise with variance r. The rationale for this measurement model is explained in [140]. The simulation results in the same work demonstrate the suitability of the model over the traditional ABP model.

8.3.2 Process Model

The state vector $\boldsymbol{x}_n$ includes the instantaneous respiratory and cardiac angles $\boldsymbol{\theta}^{\mathrm{r}}_n$ and $\boldsymbol{\theta}^{\mathrm{c}}_n$, the instantaneous mean cardiac frequency $\bar{\boldsymbol{f}}^{\mathrm{c}}_n$, the instantaneous cardiac frequency $\boldsymbol{f}^{\mathrm{c}}_n$, and the sinusoidal coefficients $\{\boldsymbol{r}_{\cdot,k,n}, \boldsymbol{c}_{\cdot,k,n}, \boldsymbol{m}_{\cdot,k,j,n}\}$, that represent the morphology of the ABP signal. The process model can be expressed as,

$$\boldsymbol{\theta}^{\mathrm{r}}_{n+1} = 2\pi(n+1)T_{\mathrm{s}}\boldsymbol{f}^{\mathrm{r}} \tag{8.6}$$

$$\boldsymbol{\theta}^{\mathrm{c}}_{n+1} = \boldsymbol{\theta}^{\mathrm{c}}_n + 2\pi T_{\mathrm{s}}\boldsymbol{f}^{\mathrm{c}}_n \tag{8.7}$$

$$\bar{\boldsymbol{f}}^{\mathrm{c}}_{n+1} = g\left[\bar{\boldsymbol{f}}^{\mathrm{c}}_n + \boldsymbol{u}_{\bar{\boldsymbol{f}}^{\mathrm{c}},n}\right] \tag{8.8}$$

$$\boldsymbol{f}^{\mathrm{c}}_{n+1} = \bar{\boldsymbol{f}}^{\mathrm{c}}_n + \alpha\left(\boldsymbol{f}^{\mathrm{c}}_n - \bar{\boldsymbol{f}}^{\mathrm{c}}_n\right) + \boldsymbol{u}_{\boldsymbol{f}^{\mathrm{c}},n} \tag{8.9}$$

$$\boldsymbol{r}_{\cdot,k,n+1} = \boldsymbol{r}_{\cdot,k,n} + \boldsymbol{u}_{\boldsymbol{r},n} \tag{8.10}$$

$$\boldsymbol{c}_{\cdot,k,n+1} = \boldsymbol{c}_{\cdot,k,n} + \boldsymbol{u}_{\boldsymbol{c},n} \tag{8.11}$$

$$\boldsymbol{m}_{\cdot,k,j,n+1} = \boldsymbol{m}_{\cdot,k,j,n} + \boldsymbol{u}_{\boldsymbol{m},n} \tag{8.12}$$

where $\boldsymbol{f}^{\mathrm{r}}$ is the constant respiratory frequency, $\boldsymbol{f}^{\mathrm{c}}_n$ the instantaneous cardiac frequency, T_{s} the sampling period, $\bar{\boldsymbol{f}}^{\mathrm{c}}_n$ the instantaneous mean cardiac frequency, α the autoregressive coefficient, and $\boldsymbol{u}_{\boldsymbol{r},n}$, $\boldsymbol{u}_{\boldsymbol{c},n}$, and $\boldsymbol{u}_{\boldsymbol{m},n}$ the process noises with variances $q_{\boldsymbol{r}}$, $q_{\boldsymbol{c}}$, and $q_{\boldsymbol{m}}$. Since this process model is for ABP signals of patients with full mechanical support, the respiratory frequency $\boldsymbol{f}^{\mathrm{r}}$ has a known constant value.

8.3.3 ABP Signal Signal Tracking

The first step toward PPV index tracking is to continuously estimate ABP signal parameters of the measurement model shown in (8.2)–(8.5). By applying the Fast

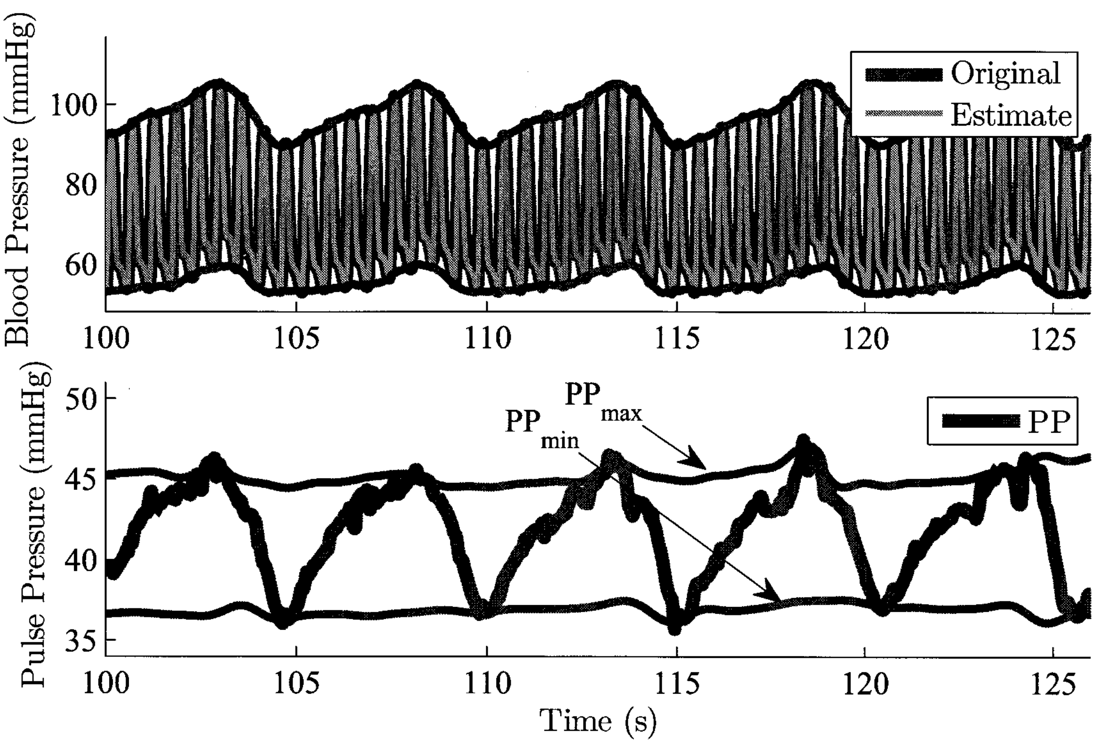


Figure 8.1: Top: Original ABP signal (red) and its estimate (green) with automatically computed envelopes (blue). Bottom: Automatically computed PP signal (red) and its envelopes (blue).

MAM-PF directly to the ABP signal state-space model, one can compute the MAP estimates of all ABP signal parameters. The details of the Fast MAM-PF recursions are listed in Algorithm 5 in Chapter 5. Table 8.1 summarizes user-specified parameter values for the Fast MAM-PF ABP signal tracker.

8.3.4 ABP Signal Envelope Estimation

Given the estimated signal parameters in (8.6)–(8.12), it is possible to estimate the upper envelope, $e_{\mu,n}$, and lower envelope, $e_{\ell,n}$, of ABP signals by following steps below,

$$\begin{aligned}
\boldsymbol{\theta}^{\mathrm{c}}_{\max,n} &= \arg\max_{\theta} \sum_{k=1}^{N^{\mathrm{c}}_{\mathrm{h}}} \boldsymbol{\rho}_{k,n} \left[\boldsymbol{c}_{1,k,n} \cos(k\theta) + \boldsymbol{c}_{2,k,n} \sin(k\theta)\right] && (8.13)\\
\boldsymbol{\theta}^{\mathrm{c}}_{\min,n} &= \arg\min_{\theta} \sum_{k=1}^{N^{\mathrm{c}}_{\mathrm{h}}} \boldsymbol{\rho}_{k,n} \left[\boldsymbol{c}_{1,k,n} \cos(k\theta) + \boldsymbol{c}_{2,k,n} \sin(k\theta)\right] && (8.14)\\
\boldsymbol{\kappa}_{\max,k,n} &= \boldsymbol{c}_{1,k,n} \cos\left(k\boldsymbol{\theta}^{\mathrm{c}}_{\max,n}\right) + \boldsymbol{c}_{2,k,n} \sin\left(k\boldsymbol{\theta}^{\mathrm{c}}_{\max,n}\right) && (8.15)\\
\boldsymbol{\kappa}_{\min,k,n} &= \boldsymbol{c}_{1,k,n} \cos\left(k\boldsymbol{\theta}^{\mathrm{c}}_{\min,n}\right) + \boldsymbol{c}_{2,k,n} \sin\left(k\boldsymbol{\theta}^{\mathrm{c}}_{\min,n}\right) && (8.16)\\
\boldsymbol{e}_{\mu,n} &= \boldsymbol{\gamma}_n + \sum_{k=1}^{N^{\mathrm{c}}_{\mathrm{h}}} \boldsymbol{\rho}_{k,n} \boldsymbol{\kappa}_{\max,k,n} && (8.17)\\
\boldsymbol{e}_{\ell,n} &= \boldsymbol{\gamma}_n + \sum_{k=1}^{N^{\mathrm{c}}_{\mathrm{h}}} \boldsymbol{\rho}_{k,n} \boldsymbol{\kappa}_{\min,k,n} && (8.18)
\end{aligned}$$

where $\arg\max_x f(x)$ and $\arg\min_x f(x)$ are operators to obtain the value of x for which $f(x)$ attains its maximum and minimum values, respectively. The top plot in Fig. 8.1 shows a five respiratory cycle period of an ABP signal $\boldsymbol{y}_n$ (thick red), its estimate $\hat{\boldsymbol{y}}_n$ (thin green), and its estimated envelopes $\boldsymbol{e}_{\mu,n}$ and $\boldsymbol{e}_{\ell,n}$ (blue).

8.3.5 Pulse Pressure Signal Envelope Estimation

The pulse pressure (PP) signal is the difference between the upper envelope, $\boldsymbol{e}_{\mu,n}$, and lower envelope, $\boldsymbol{e}_{\ell,n}$, of the ABP signal. This PP signal oscillates roughly at the respiratory rate as shown in the bottom plot in Fig. 8.1. This oscillation is due to the respiratory effect on the variation of systemic ABP under full ventilation support [133]. Within each respiratory cycle PP reaches its maximum ($\mathrm{PP}_{\max}$) and minimum ($\mathrm{PP}_{\min}$) values, which are two critical parameters to compute the PPV index. Traditionally, the $\mathrm{PP}_{\max}$ and $\mathrm{PP}_{\min}$ values have been computed only once per each respiratory cycle. Given the estimated signal parameters in (8.6)–(8.12), however, one can compute the continuous equivalents of $\mathrm{PP}_{\max}$ and $\mathrm{PP}_{\min}$. They are the upper, $\varepsilon_{\mu,n}$, and lower, $\varepsilon_{\ell,n}$, envelopes of the PP signal. The upper envelope, $\varepsilon_{\mu,n}$, is the continuous estimate of $\mathrm{PP}_{\max}$ and the lower envelope, $\varepsilon_{\ell,n}$, that of $\mathrm{PP}_{\min}$. The $\varepsilon_{\mu,n}$ and $\varepsilon_{\ell,n}$ estimation steps can be written as follows,

$$\varrho_{k,n} = \sum_{j=1}^{N_\mathrm{h}^\mathrm{r}} \boldsymbol{m}_{1,k,j,n} \cos(j\theta) + \boldsymbol{m}_{2,k,j,n} \sin(j\theta) \tag{8.19}$$

$$\boldsymbol{\theta}^\mathrm{r}_{\max,n} = \arg\max_{\theta} \sum_{k=1}^{N_\mathrm{h}^\mathrm{c}} (1 + \varrho_{k,n}) (\boldsymbol{\kappa}_{\max,k,n} - \boldsymbol{\kappa}_{\min,k,n}) \tag{8.20}$$

$$\boldsymbol{\theta}^\mathrm{r}_{\min,n} = \arg\min_{\theta} \sum_{k=1}^{N_\mathrm{h}^\mathrm{c}} (1 + \varrho_{k,n}) (\boldsymbol{\kappa}_{\max,k,n} - \boldsymbol{\kappa}_{\min,k,n}) \tag{8.21}$$

$$\varrho_{\max,k,n} = \sum_{j=1}^{N_\mathrm{h}^\mathrm{r}} \boldsymbol{m}_{1,k,j,n} \cos\left(j\boldsymbol{\theta}^\mathrm{r}_{\max,n}\right) + \boldsymbol{m}_{2,k,j,n} \sin\left(j\boldsymbol{\theta}^\mathrm{r}_{\max,n}\right) \tag{8.22}$$

$$\varrho_{\min,k,n} = \sum_{j=1}^{N_\mathrm{h}^\mathrm{r}} \boldsymbol{m}_{1,k,j,n} \cos\left(j\boldsymbol{\theta}^\mathrm{r}_{\min,n}\right) + \boldsymbol{m}_{2,k,j,n} \sin\left(j\boldsymbol{\theta}^\mathrm{r}_{\min,n}\right) \tag{8.23}$$

$$\varepsilon_{\mu,n} = \sum_{k=1}^{N_\mathrm{h}^\mathrm{c}} (1 + \varrho_{\max,k,n}) (\boldsymbol{\kappa}_{\max,k,n} - \boldsymbol{\kappa}_{\min,k,n}) \tag{8.24}$$

$$\varepsilon_{\ell,n} = \sum_{k=1}^{N_\mathrm{h}^\mathrm{c}} (1 + \varrho_{\min,k,n}) (\boldsymbol{\kappa}_{\max,k,n} - \boldsymbol{\kappa}_{\min,k,n}) \tag{8.25}$$

where $1 + \varrho_{k,n}$ is equal to $\boldsymbol{\rho}_{k,n}$ and $\varepsilon_{\mu,n}$ and $\varepsilon_{\ell,n}$ are the continuous estimates of the $\mathrm{PP}_{\max}$ and $\mathrm{PP}_{\min}$, respectively. The blue lines in the bottom plot in Fig. 8.1 represent the upper, $\varepsilon_{\mu,n}$, and lower, $\varepsilon_{\ell,n}$, envelopes of the PP signal obtained by following the steps described above.

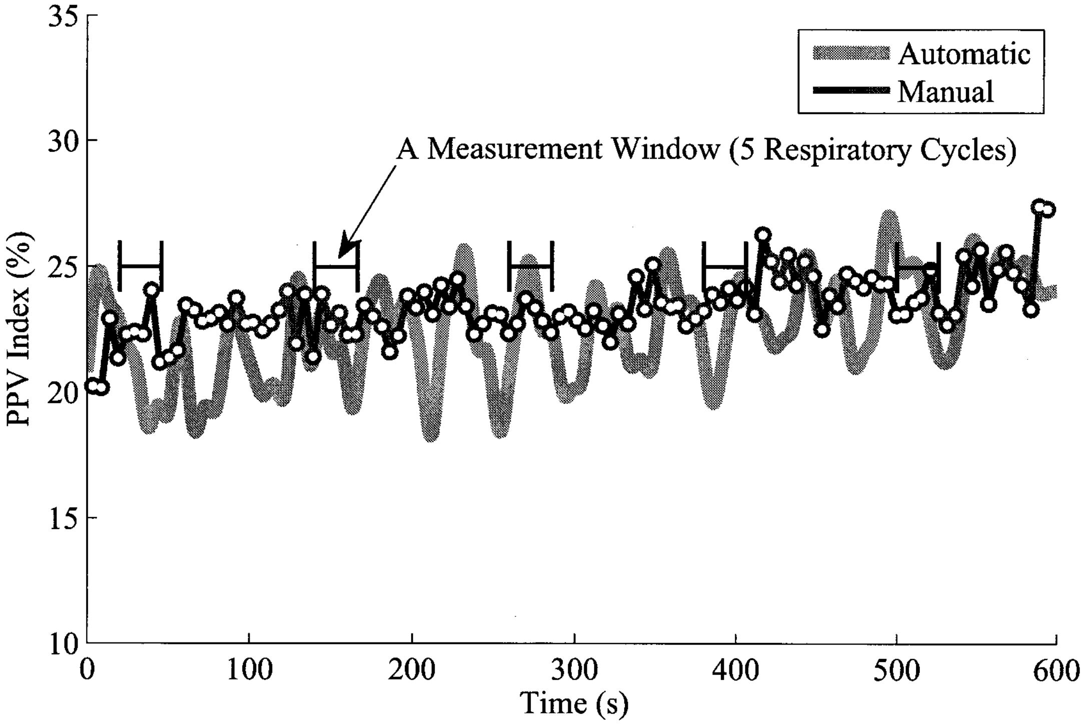


Figure 8.2: Automatic PPV index (green) and manual PPV index (red) over the entire ABP signal duration (10 minutes). One PPV index measurement is computed from each measurement window, which is a time period of 5 respiratory cycles.

8.3.6 Pulse Pressure Variation Calculation

Given the $\varepsilon_{\mu,n}$ and $\varepsilon_{\ell,n}$ values, it is straightforward to calculate the PPV index. It can be computed as follows,

$$\text{PPV}(\%) = 100 \times \frac{\varepsilon_{\max} - \varepsilon_{min}}{(\varepsilon_{\max} + \varepsilon_{\min})/2} \tag{8.26}$$

This new PPV index is different from the traditional PPV index described in (8.1) because the new one is continuous in time while the traditional one can be obtained only once per each respiratory cycle.

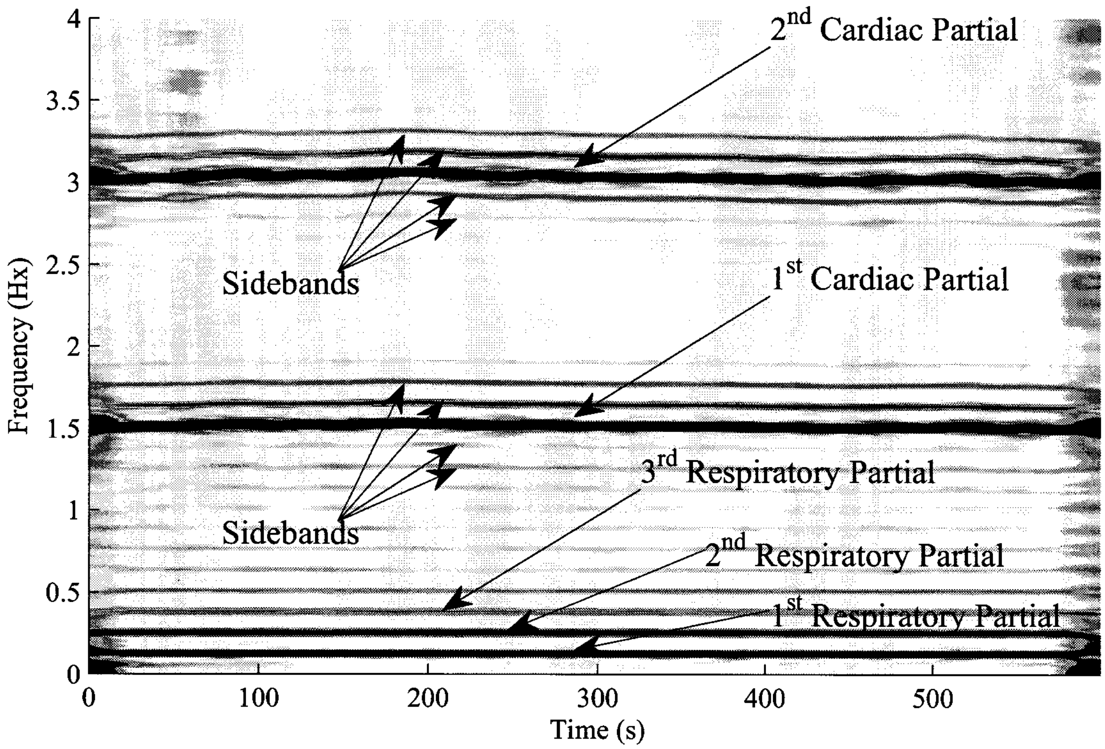


Figure 8.3: Spectrogram of one of the 26 ABP signals from subjects under full ventilation support.

Fig. 8.2 illustrates an example of the automatically computed continuous PPV index (thick green) and the manually obtained discrete PPV index (thin red) of a real 10 min ABP signal. Each hollow white dot represents a "discrete" PPV index, which can be obtained once per each respiratory cycle.

8.4 Algorithm Assessment

8.4.1 Assessment Data

The Massachusetts General Hospital (MGH) waveform database on PhysioNet is a comprehensive collection of electronic recordings of hemodynamic and electrocardiographic waveforms patients in critical care units [130, 131]. It consists of recordings from 250 patients representing a broad spectrum of physiologic and pathophysiologic states. The typical recording includes arterial blood pressure (ABP) signals in addition to seven other types of waveforms. By visually inspecting the spectrogram of ABP signals we identified 26 patients whose respiratory rate remained constant at least for 10 consecutive minutes. The constant respiratory rate shown in spectrogram was used as an indicator of full respiratory support. Fig. 8.3 shows the spectrogram of one of the 26 ABP signals. The total duration of the 26 ABP signals was a little over 4 hours (260 min). The original sample rate f_s of the signals was 360 Hz, but they were downsampled by a factor of 9, so that the final sample rate f_s was 40 Hz.

Typically the constant respiratory rate $\boldsymbol{f}^\mathrm{r}$ of subjects under full ventilation support is easily accessible medical information since it is precisely controlled by the ventilator. However, the MGH database does not provide this information. Therefore, the respiratory rate $\boldsymbol{f}^\mathrm{r}$ of each of the 26 ABP signals was estimated by following three steps. First, a given ABP signal was lowpass-filtered at 1 Hz to remove all cardiac components. Second, multiple synthetic cosine signals $\cos(2\pi n T_\mathrm{s} f)$ were generated by sweeping the frequency f from 0.01 Hz to 1 Hz. Finally, the cross-correlation between the lowpass-filtered ABP signal and synthetic cosine signals was calculated to find the best match and choose the frequency of the best matched synthetic cosine signal as the estimate of the constant respiratory rate. In this study, it was necessary to perform this offline analysis. However, it should be noticed that normally this type of offline analysis would not be necessary because $\boldsymbol{f}^\mathrm{r}$ is known.

The number of cardiac components N^c was 10 and that of respiratory com-

Table 8.1: Summary of user-specified parameters for the PPV index tracker.

Name	Symbol	Value
No. particles	N^{p}	250
No. cardiac components	N^{c}	10
No. respiratory components	N^{r}	3
Minimum heart rate	$f^{\mathrm{c}}_{\mathrm{min}}$	50/60 Hz
Maximum heart rate	$f^{\mathrm{c}}_{\mathrm{max}}$	150/60 Hz
Measurement noise variance	r	var(y)/1e3
Cardiac frequency variance	$q_{f^{\mathrm{c}}}$	7e-6 T_s
Respiratory amplitude variance	q_a, q_b	1e-5 T_s
Modulation factor amplitude variance	q_c, q_d	1e-9 T_s
Cardiac amplitude variance	q_e, q_f	1e-6 T_s
Initial respiratory amplitude	u_a, u_b	std(y)/1e2
Initial modulation factor amplitude	u_c, u_d	std(y)/1e2
Initial cardiac amplitude	u_e, u_f	std(y)

ponents N^{r} was 3. The number of particles was 250. Table 8.1 lists the parameter values used for the PPV index estimator.

8.4.2 Manual PPV Annotations (Current Standard)

The peaks and troughs of the ABP signals were *manually* detected and the PPV indices (current standard) were calculated as defined in (8.1). They are referred to as manual PPV indices $\mathrm{PPV}_{\mathrm{manu}}$. $\mathrm{PPV}_{\mathrm{auto}}$ represents PPV indices obtained using the proposed PPV index tracking algorithm.

3 ABP signals out of the 26 ABP signals were excluded from the study because it was not possible to annotate their PP signals due to a high noise level or abnormal cardiac activity.

8.4.3 Statistical Analysis

5 PPV index measurements were taken per subject separately by 2 min. Each PPV index measurement is an averaged value over 5 respiratory cycles. Fig. 8.2 shows the measurement windows. The proposed PPV index tracking algorithm was assessed by calculating the agreement (mean ± standard deviation) between PPV_{auto} and PPV_{manu} measurements and using Bland-Altman analysis.

A Bland-Altman plot is a statistical visualization method that is often used in the assessment of PPV estimation algorithms in order to determine the agreement between two different PPV estimates. It has the difference ΔPPV between PPV_{auto} and PPV_{manu} measurements on the y-axis and the PPV_{manu} measurements on the x-axis. It shows the overall accuracy of estimation and any estimation bias or trend versus PPV_{manu} measurements. It is used to compare the standard using manual annotations with the proposed automatic estimation algorithm.

8.5 Results

Fig. 8.4 depicts the Bland-Altman plot of the 23 subjects. There are 5 PPV measurements available per each subject. Most (98%) of PPV_{auto} measurements were in agreement with PPV_{manu} measurements within ±5% accuracy, and 100% within ± 7% accuracy.

Table 8.2 lists the mean ± standard deviation of 5 PPV_{manu} and PPV_{auto} measurements for each subject. The second column is for PPV_{manu} and the third column for PPV_{auto}.

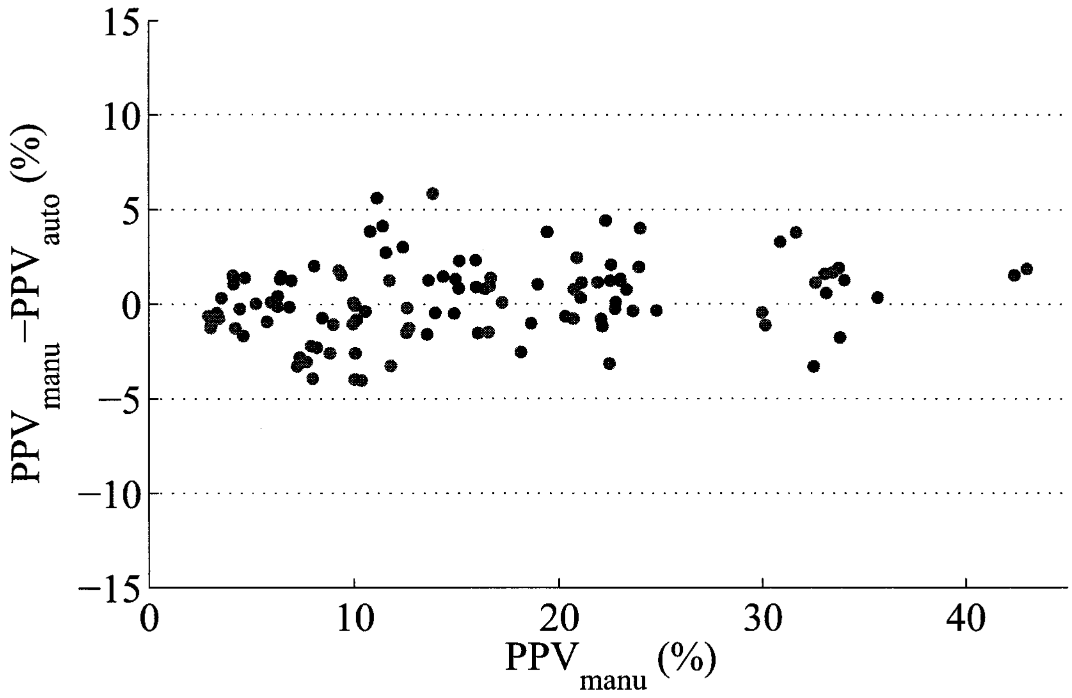


Figure 8.4: Bland-Altman plot of the 23 subjects.

8.6 Discussion

8.6.1 Significance

Despite the availability of a commercial device for PPV monitoring, the PICCO PPV system (PICCO Pulsion Medical Systems, Munich, Germany), the need for additional independent PPV estimation algorithms is significant for several reasons. Some studies have suggested that the PICCO PPV algorithm may not work well in certain situations [141]. The validation results in other studies indicate that the PICCO system may not work well during regions of abrupt hemodynamic changes [139]. Secondly, the proposed PPV algorithm is based on a sound statistical model for ABP signals and it can be implemented and used to accurately

Table 8.2: Summary of the mean and standard deviation of the $\mathrm{PPV}_{\mathrm{manu}}$ and $\mathrm{PPV}_{\mathrm{auto}}$ measurements.

Subject	$\mathrm{PPV}_{\mathrm{manu}}$ (%)	$\mathrm{PPV}_{\mathrm{auto}}$ (%)
1	23.2±1.0	23.6±1.0
2	23.0±0.3	22.8±1.6
3	13.7±0.8	14.3±1.2
4	12.3±1.1	9.1±3.1
5	36.3±5.9	33.9±6.7
6	8.7±1.8	9.5±2.0
7	18.2±2.6	18.8±1.7
8	6.3±1.3	6.2±0.8
9	9.6±1.7	7.1±0.8
10	15.9±0.8	14.8±0.8
11	5.3±1.0	5.5±0.7
12	3.1±0.2	4.0±0.3
13	31.8±1.6	31.5±0.8
14	15.3±2.8	13.1±1.8
15	33.9±1.1	34.3±1.8
16	4.6±1.1	3.4±1.0
17	20.4±2.3	20.1±2.7
18	9.8±0.8	11.9±2.2
19	9.6±2.1	11.2±1.4
20	11.9±0.5	10.3±1.2
21	8.7±1.8	11.9±1.9
22	21.7±1.9	18.9±1.2
23	20.9±2.1	20.0±0.7

estimate PPV in data already collected and archived. Finally, a detailed description has been provided to ensure that other researchers and medical manufacturers can implement it, use it for research purposes, and independently validate the results obtained using commercial PPV monitoring systems.

It is important to note that the proposed algorithm is a completely new design from the previously developed algorithm in [137]. The algorithm in [137] was made publicly available by the authors and due to its performance has been adopted by Philips Medical Systems. Currently, the PPV algorithm is displayed in real-time on the Philips Intelliveu MP70 monitors (Intellivue MP70, Philips Medical Systems) and has been used in numerous studies related to PPV and fluid responsiveness. Its ability to monitor fluid responsiveness in the operating room and its accuracy against the current standard obtained by manual annotations were assessed by Cannesson [138]. PPV is considered the best predictor of fluid responsiveness in this setting. However, previously it was not possible to conveniently monitor this index in the operating room or in the intensive care unit because it had to be manually calculated. Thus, the automatic PPV estimation technique has potential clinical application for fluid management optimization in the operating room [138].

Aboy *et al.* pointed out that the PPV algorithm adopted by Philips in their Intelliveu MP70 monitors may not work adequately in regions of abrupt hemodynamic changes or significant artifact [139].

8.6.2 Algorithm's Advantages

This chapter describes and assesses a novel automatic algorithm designed to estimate the pulse pressure variation (PPV) index from arterial blood pressure (ABP) signals in regions of abrupt hemodynamic changes. The proposed algorithm is based on a statistical state-space model for arterial blood pressure signals and sequential Monte Carlo estimation methods.

The major algorithm design difference of the proposed algorithm with respect to other published algorithms [137, 139] is the fact that the proposed method is based on a statistical state-space model and estimation of the cardiovascular pressure signal. The state-space modeling stage results in an algorithm that is more robust to hemodynamic changes and artifacts. The statistical state-space

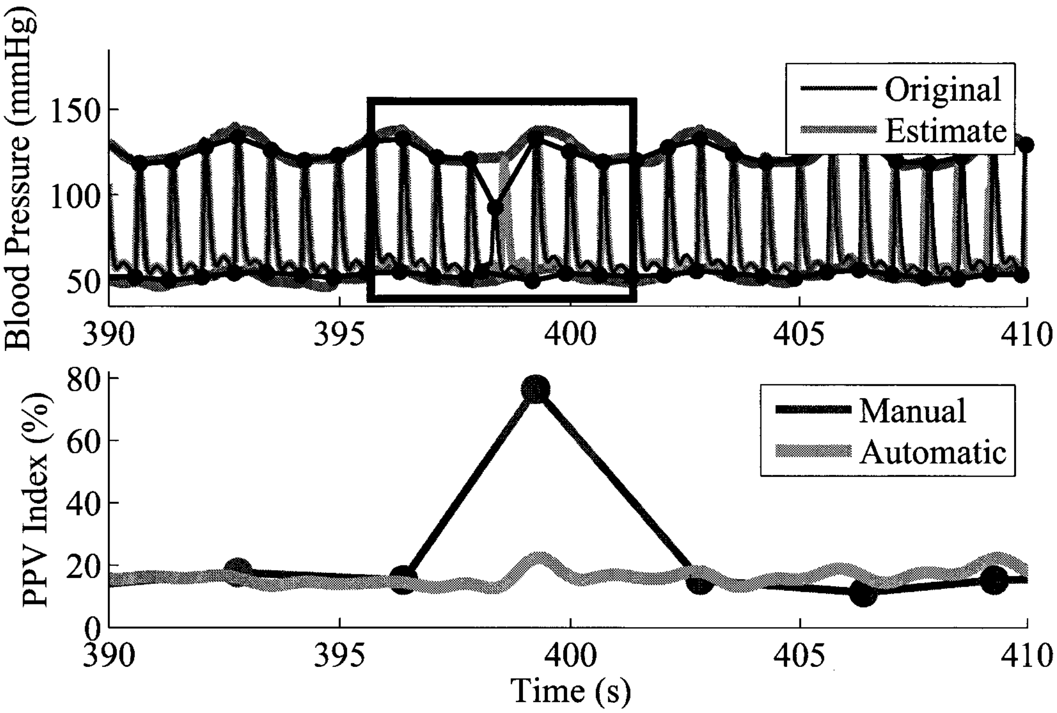


Figure 8.5: Top: Original ABP signal (red) with its manually annotated envelopes (black) and signal estimate (green) with its automatically computed envelopes (purple). Bottom: Manual PPV indices (red) and automatic PPV indices (green) where one of the manual PPV indices has an abnormally high value.

signal model and associated model parameter estimation algorithm automatically filter out noise and artifact that cannot be captured with the model. Since the statistical signal model is based on cardiovascular physiology and pathophysiology, signal features that are not physiological in nature are automatically filtered out.

Figs. 8.5–8.6 exemplify a case where signal features that are not physiological in nature are automatically filtered out resulting in more accurate PPV index estimation than manual annotation. The top plot in Fig. 8.5 illustrates 6 respiratory cycles of the ABP signal (red) and its estimate (green). It also shows the manually annotated signal envelopes (black) and the automated computed signal envelopes (light blue). The bottom plot in Fig. 8.5 depicts the PPV_{manu} and PPV_{auto} over the

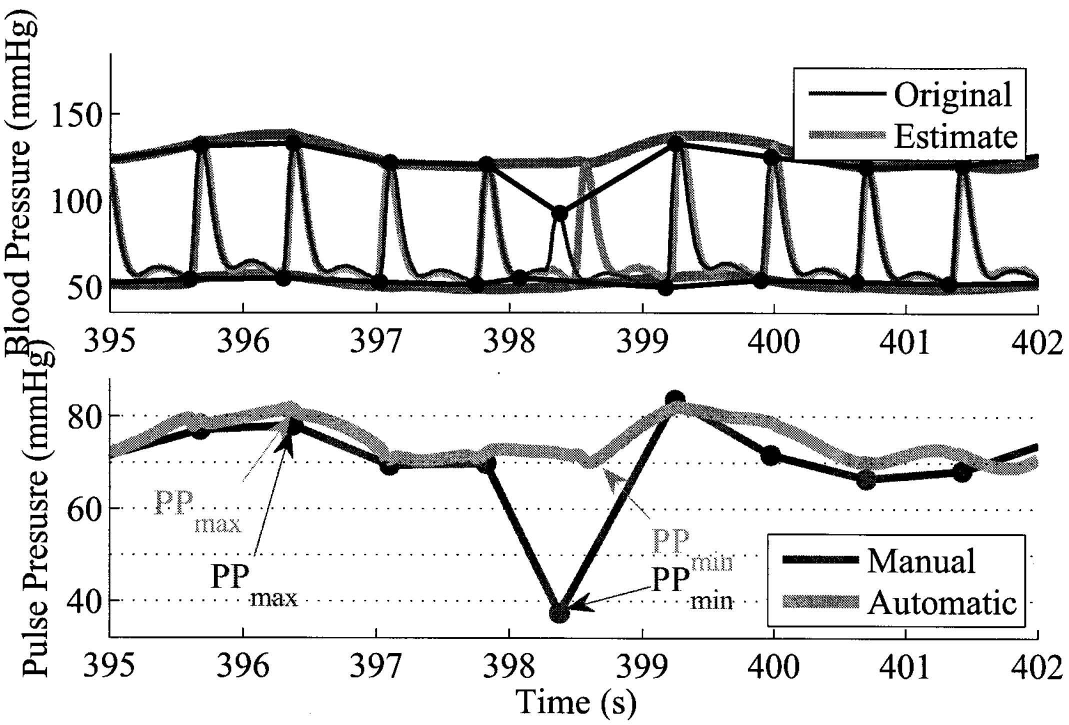


Figure 8.6: Top: Original ABP signal (red) with its manually annotated envelopes (black) and signal estimate (green) with its automatically computed envelopes (purple). Bottom: Manual PP signal (red) and automatic PP signal (green) where the manual PP signal decreases momentarily due to an irregular and abnormal heart beat in the ABP signal.

same period. Around 399 s, the PPV_{manu} value (red) abruptly increases up to 78% while the PPV_{auto} value (green) remains at 20%. Around 403 s, the PPV_{manu} value returns to 20%. Fig. 8.6 focuses on the time period marked with the black rectangular box in Fig. 8.5. The top plot in Fig. 8.6 shows that the heart beat between 398 s and 399 s is irregular and abnormal. As a result, the corresponding PP_{manu} shown in the bottom plot reaches a very small minimum value ($PP_{min,manu}$: 38%) between 398 s and 399 s. However, the automatically computed minimum PP value ($PP_{min,auto}$) over the same period is as high as 70%. This discrepancy between the manual annotation and the proposed automatic method results from the capability of the MAM-PF algorithm, which estimates the ABP signal based on the state-space model. While the original heart beat 398 s and 399 s in Fig. 8.6 is abnormal in a physiological sense, the estimated heart beat over the same time period shows the physiologically expected morphology and location of the heart beat.

8.6.3 Study Limitations

The algorithm's assessment was based on only 23 subjects with pre-recorded ABP data. Additionally, for each subject five PPV estimates were used in the assessment study. This assessment was designed to be an engineering algorithm validation against current standard manual annotations, and not a clinical validation study. Consequently, a clinical validation study assessing the ability of the proposed algorithm to monitor fluid responsiveness in the operating room in situations involving abrupt hemodynamic changes still needs to be conducted. This would require the proposed algorithm to be first adopted as part of a commercial system as was the case with our previous automatic PPV algorithm [137].

8.7 Summary

This chapter describes a novel automatic PPV estimation algorithm based on a statistical state-space model inspired in the underlying cardiovascular and respiratory physiology. This algorithm uses the MAM-PF for optimal parameter estimation. The assessment results indicate good agreement against the current standard PPV (98% within ± 5% of the current standard). The algorithm was designed to work during regions of abrupt hemodynamic changes and signal artifact.

Chapter 9

Multiple Harmonic Set Tracking

This chapter discusses a practical application of the multiple harmonic set tracking case, where the number of harmonic sets is 2. The application is tracking the pulse pressure variation (PPV) index from arterial blood pressure (ABP) signals under spontaneously breathing conditions.

9.1 Introduction to the Clinical Problem

Chapter 8 describes a novel automatic algorithm designed to estimate the PPV index in subjects receiving full mechanical ventilation. As mentioned previously, the PPV index is a dynamic predictor of increases in cardiac output due to an infusion of fluid, which is one of the most sensitive and specific predictors of fluid responsiveness. Michard has shown that the PPV index is a useful dynamic indicator to guide fluid therapy in different patient populations receiving mechanical ventilation [106].

Accurately predicting the hemodynamic response to volume expansion is important not only in mechanically ventilated subjects, but it may also have potential in situations involving spontaneous breathing subjects [142, 143]. Although numerous studies have demonstrated the usefulness of the PPV index as a reliable predictor of fluid responsiveness in mechanically ventilated subjects, its usage in spontaneously breathing subjects in need of fluid therapy has been discussed only in a few studies. Recently, the research community has been paying attention to accurate and reliable indices to predict fluid responsiveness in spontaneously breathing subjects [144–146]. In their recent study, Dahl *et al.* have shown that PPV can be used as a predictor of fluid responsiveness during spontaneous breathing in animal models [142].

Fluid responsiveness prediction in spontaneously breathing subjects is difficult. Since the swings in intrathoracic pressure are minor during spontaneous breathing, dynamic parameters like pulse PPV and systolic pressure variation (SPV) are usually small and more difficult to calculate [142]. Additionally, while there are currently only a few publicly available algorithms to automatically estimate PPV accurately and reliably in the case of mechanically ventilated subjects [137, 139], at the moment there are no automatic algorithms for estimating PPV in spontaneously breathing subjects. This has, in part, prevented and made difficult research involving the use of PPV during spontaneous breathing.

This chapter describes a novel automatic algorithm that can be used to obtain the pulse pressure variation (PPV) index from arterial blood pressure (ABP) signals during spontaneous breathing. The proposed algorithm is based on a statistical state-space model for arterial blood pressure signals and the modified version of the Fast MAM-PF, called the *Dual MAM-PF.* The Dual MAM-PF is devised to

The proposed automatic algorithm's performance is assessed against the current standard (manual annotations) using real ABP signals from the Massachusetts

General Hospital (MGH) Waveform Database [130] on PhysioNet [131].

9.2 Methodology

This section describes the state-space model of ABP signals of spontaneously breathing subjects and provides the details of the automatic PPV index tracking algorithm.

9.2.1 Measurement Model

The measurement model of the ABP signal with spontaneous breathing activity is shown in (9.1)–(9.4), where $\boldsymbol{\gamma}_n$ is the respiratory signal, $\boldsymbol{\mu}_n$ the amplitude-modulated cardiac signal, $\boldsymbol{\rho}_{k,n}$ the amplitude modulation factor of the k^{th} cardiac harmonic partial, $\boldsymbol{\kappa}_{k,n}$ the k^{th} cardiac harmonic partial, $\boldsymbol{\theta}_n^{\text{r}}$ the instantaneous respiratory angle, $\boldsymbol{\theta}_n^{\text{c}}$ the instantaneous cardiac angle, $\boldsymbol{f}_n^{\text{r}}$ the instantaneous respiratory rate, N_{h}^{r} the number of respiratory partials, N_{h}^{c} the number of cardiac partials, and $\boldsymbol{v}_n$ the white Gaussian measurement noise with variance r. This measurement model was also used in Chapter 8.

$$\boldsymbol{y}_n = \boldsymbol{\gamma}_n + \boldsymbol{\mu}_n + \boldsymbol{v}_n = \boldsymbol{\gamma}_n + \sum_{k=1}^{N_{\text{h}}^{\text{c}}} \boldsymbol{\rho}_{k,n} \boldsymbol{\kappa}_{k,n} + \boldsymbol{v}_n \tag{9.1}$$

$$\boldsymbol{\gamma}_n = \sum_{k=1}^{N_{\text{h}}^{\text{r}}} \boldsymbol{r}_{1,k,n} \cos\left(k\boldsymbol{\theta}_n^{\text{r}}\right) + \boldsymbol{r}_{2,k,n} \sin\left(k\boldsymbol{\theta}_n^{\text{r}}\right) \tag{9.2}$$

$$\boldsymbol{\rho}_{k,n} = 1 + \sum_{j=1}^{N_{\text{h}}^{\text{r}}} \boldsymbol{m}_{1,k,j,n} \cos\left(j\boldsymbol{\theta}_n^{\text{r}}\right) + \boldsymbol{m}_{2,k,j,n} \sin\left(j\boldsymbol{\theta}_n^{\text{r}}\right) \tag{9.3}$$

$$\boldsymbol{\kappa}_{k,n} = \sum_{k=1}^{N_{\text{h}}^{\text{c}}} \boldsymbol{c}_{1,k,n} \cos\left(k\boldsymbol{\theta}_n^{\text{c}}\right) + \boldsymbol{c}_{2,k,n} \sin\left(k\boldsymbol{\theta}_n^{\text{c}}\right) \tag{9.4}$$

9.2.2 Process Model

The state vector $\boldsymbol{x}_n$ includes the instantaneous respiratory $\boldsymbol{\theta}^{\mathrm{r}}_n$ and cardiac $\boldsymbol{\theta}^{\mathrm{c}}_n$ angles, the instantaneous mean respiratory $\bar{\boldsymbol{f}}^{\mathrm{r}}_n$ and cardiac $\bar{\boldsymbol{f}}^{\mathrm{c}}_n$ frequencies, the instantaneous respiratory $\boldsymbol{f}^{\mathrm{r}}_n$ and cardiac $\boldsymbol{f}^{\mathrm{c}}_n$ frequencies, and the sinusoidal coefficients $\{\boldsymbol{r}_{1,k,n}, \boldsymbol{r}_{2,k,n}, \boldsymbol{c}_{1,k,n}, \boldsymbol{c}_{2,k,n}, \boldsymbol{m}_{1,k,n}, \boldsymbol{m}_{2,k,n}\}$, that represent the morphology of the ABP signal. The process model can be expressed as,

$$\boldsymbol{\theta}^{\mathrm{r}}_{n+1} = \boldsymbol{\theta}^{\mathrm{r}}_n + 2\pi T_{\mathrm{s}} \boldsymbol{f}^{\mathrm{r}}_n \tag{9.5}$$

$$\boldsymbol{\theta}^{\mathrm{c}}_{n+1} = \boldsymbol{\theta}^{\mathrm{c}}_n + 2\pi T_{\mathrm{s}} \boldsymbol{f}^{\mathrm{c}}_n \tag{9.6}$$

$$\bar{\boldsymbol{f}}^{\mathrm{r}}_{n+1} = g\left[\bar{\boldsymbol{f}}^{\mathrm{r}}_n + \boldsymbol{u}_{\bar{\boldsymbol{f}}^{\mathrm{r}},n}\right] \tag{9.7}$$

$$\bar{\boldsymbol{f}}^{\mathrm{c}}_{n+1} = g\left[\bar{\boldsymbol{f}}^{\mathrm{c}}_n + \boldsymbol{u}_{\bar{\boldsymbol{f}}^{\mathrm{c}},n}\right] \tag{9.8}$$

$$\boldsymbol{f}^{\mathrm{r}}_{n+1} = \bar{\boldsymbol{f}}^{\mathrm{r}}_n + \alpha\left(\boldsymbol{f}^{\mathrm{r}}_n - \bar{\boldsymbol{f}}^{\mathrm{r}}_n\right) + \boldsymbol{u}_{\boldsymbol{f}^{\mathrm{r}},n} \tag{9.9}$$

$$\boldsymbol{f}^{\mathrm{c}}_{n+1} = \bar{\boldsymbol{f}}^{\mathrm{c}}_n + \alpha\left(\boldsymbol{f}^{\mathrm{c}}_n - \bar{\boldsymbol{f}}^{\mathrm{c}}_n\right) + \boldsymbol{u}_{\boldsymbol{f}^{\mathrm{c}},n} \tag{9.10}$$

$$\boldsymbol{r}_{\cdot,k,n+1} = \boldsymbol{r}_{\cdot,k,n} + \boldsymbol{u}_{\boldsymbol{r},n} \tag{9.11}$$

$$\boldsymbol{c}_{\cdot,k,n+1} = \boldsymbol{c}_{\cdot,k,n} + \boldsymbol{u}_{\boldsymbol{c},n} \tag{9.12}$$

$$\boldsymbol{m}_{\cdot,k,n+1} = \boldsymbol{m}_{\cdot,k,n} + \boldsymbol{u}_{\boldsymbol{m},n} \tag{9.13}$$

where $\boldsymbol{f}^{\mathrm{r}}_n$ is the instantaneous respiratory frequency, $\boldsymbol{f}^{\mathrm{c}}_n$ the instantaneous cardiac frequency, T_{s} the sampling period, $\bar{\boldsymbol{f}}^{\mathrm{r}}_n$ the instantaneous mean respiratory frequency, $\bar{\boldsymbol{f}}^{\mathrm{c}}_n$ the instantaneous mean cardiac frequency, α the autoregressive coefficient of frequency, and $\boldsymbol{u}_{\boldsymbol{r},n}$, $\boldsymbol{u}_{\boldsymbol{c},n}$, and $\boldsymbol{u}_{\boldsymbol{m},n}$ the process noises with variances $q_{\boldsymbol{r}}$, $q_{\boldsymbol{c}}$, and $q_{\boldsymbol{m}}$. In contrast to the case in Chapter 8, the instantaneous respiratory frequency $\boldsymbol{f}^{\mathrm{r}}_n$ is unknown and variable since the current study involves subjects with spontaneous breathing activity. The clipping function $g[\cdot]$ limits the range of instantaneous mean frequencies, which can be written as,

$$g[f] = \begin{cases} f_{\max} - (f - f_{\max}) & f_{\max} < f \\ f & f_{\min} < f \leq f_{\max} \\ f_{\min} + (f_{\min} - f) & f \leq f_{\min}. \end{cases} \tag{9.14}$$

The range of instantaneous mean frequencies is assumed to be known as domain knowledge.

9.2.3 Dual Maximum *A-Posterior* Marginalized PF

The proposed automated PPV index estimation method requires accurate estimates of the instantaneous respiratory frequency $\boldsymbol{f}_n^{\mathrm{r}}$, the instantaneous cardiac frequency $\boldsymbol{f}_n^{\mathrm{c}}$, and the morphology of the ABP signal. Although the Fast MAM-PF based ABP signal tracker is capable of tracking multiple frequencies, there are two major issues in using the Fast MAM-PF algorithm as the ABP signal tracker for ABP signals of spontaneously breathing subjects. The first issue is that the morphology of the signal, which is represented by the sinusoidal coefficients in (9.3)–(9.4), does not belong to the linear state any more. Since the modulating signal $\boldsymbol{\rho}_{k,n}$ is multiplied by the cardiac signal $\boldsymbol{\kappa}_{k,n}$, their sinusoidal coefficients $\boldsymbol{c}_{\cdot,k,n}$ and $\boldsymbol{m}_{\cdot,k,j,n}$ are nonlinear parameters of the measurement model in (9.1). The Fast MAM-PF is applicable only to state-space models whose state vector can be partitioned into the linear and nonlinear portions [147]. The second issue is that as the dimension of the state, where particle filters are used, increases the number of necessary particles to cover the state increases exponentially. As a result, the computational burden of the Fast MAM-PF increases exponentially. The portion

Algorithm 6 Dual MAM-PF.

```
Initialization
for $j = 1, 2$ do
  for $i = 1, \dots, N_\mathrm{p}$ do
    Sample $\boldsymbol{x}_0^{\mathrm{P}_j,(i)} \sim \pi_0(\boldsymbol{x}_0^{\mathrm{P}_j})$ & $\hat{\boldsymbol{x}}_{0:-1}^{\mathrm{K},(i)} = \mathrm{E}\left[\boldsymbol{x}_0^{\mathrm{K},(i)} | \boldsymbol{x}_0^{\mathrm{P},(i)}\right]$
    $\boldsymbol{\alpha}_0^{(i)} = \pi_0\left(\boldsymbol{x}_0^{\mathrm{P},(i)}\right) p\left(\boldsymbol{y}_0 | \boldsymbol{x}_0^{\mathrm{P},(i)}, \boldsymbol{x}_0^{\mathrm{K},(i)}\right)$ & $\boldsymbol{z}_0^{(i)} = \boldsymbol{x}_0^{(i)}$
  end for
  $i_{j,0}^* = \operatorname*{argmax}_{i_j} \boldsymbol{\alpha}_{j,0}^{(i_j)}$
end for
$i_{\mathrm{MAP},0}^* = \operatorname*{argmax}_{i_{j,0}^*} \boldsymbol{\alpha}_{j,0}^{(i_{j,0}^*)}$ & $\hat{\boldsymbol{x}}_0 = \{\boldsymbol{x}_0^{\mathrm{P}_1,(i_{1,0}^*)}, \boldsymbol{x}_0^{\mathrm{P}_2,(i_{2,0}^*)}, \boldsymbol{x}_0^{\mathrm{K},(i_{\mathrm{MAP},0}^*)}\}$
Time Loop
for $n = 1, \dots, N_\mathrm{T}$ do
  for $j = 1, 2$ do
    for $i_j = 1, \dots, N_\mathrm{p}$ do
      $\boldsymbol{x}_n^{\mathrm{P}_j,(i_j)} \sim q_n\left(\boldsymbol{x}_n^{\mathrm{P}_j,(i_j)} | \boldsymbol{x}_{n-1}^{\mathrm{P}_j,(i_j)}, \boldsymbol{y}_n\right)$ & $k_j^* = \operatorname*{argmax}_{k_j} \boldsymbol{\alpha}_{j,n-1}^{(k_j)} p\left(\boldsymbol{x}_n^{\mathrm{P}_j,(i_j)} | \boldsymbol{x}_{n-1}^{\mathrm{P}_j,(k_j)}\right)$
      $\boldsymbol{x}_n^{\mathrm{P},(i_j)} = \{\boldsymbol{x}_n^{\mathrm{P}_1,(i_{1,n-1}^*)}, \dots, \boldsymbol{x}_n^{\mathrm{P}_j,(i_j)}, \dots, \boldsymbol{x}_n^{\mathrm{P}_{N_j},(i_{N_j,n-1}^*)}\}$
      $\hat{\boldsymbol{y}}_{n|0:n-1} = H_n\left(\boldsymbol{x}_n^{\mathrm{P},(i_j)}\right) \hat{\boldsymbol{x}}_{n|0:n-1}^{\mathrm{K},(k_j^*)}$ & $\boldsymbol{e}_n = \boldsymbol{y}_n - \hat{\boldsymbol{y}}_{n|0:n-1}$
      $\boldsymbol{R}_{\boldsymbol{v},n} = \left[\boldsymbol{e}_n \boldsymbol{e}_n^\mathrm{T} - H_n\left(\boldsymbol{x}_n^{\mathrm{P},(i_j)}\right) \boldsymbol{C}_{n|0:n-1}^{(k_j^*)} H_n\left(\boldsymbol{x}_n^{\mathrm{P},(i_j)}\right)^T\right]_+$
      $\hat{\boldsymbol{R}}_{v,n} = \beta \hat{\boldsymbol{R}}_{v,n-1}^{(k_j^*)} + (1-\beta) \boldsymbol{R}_{\boldsymbol{v},n}$
      $\boldsymbol{R}_{\boldsymbol{e},n} = H_n\left(\boldsymbol{x}_n^{\mathrm{P},(i_j)}\right) \boldsymbol{C}_{n|0:n-1}^{(k_j^*)} H_n\left(\boldsymbol{x}_n^{\mathrm{P},(i_j)}\right)^T + \hat{\boldsymbol{R}}_{v,n}$
      $\boldsymbol{K}_n = \boldsymbol{C}_{n|0:n-1}^{(k_j^*)} H_n\left(\boldsymbol{x}_n^{\mathrm{P},(i_j)}\right)^T (\boldsymbol{R}_{\boldsymbol{e},n})^{-1}$ & $\hat{\boldsymbol{x}}_{n|0:n}^{\mathrm{K},i_j} = \hat{\boldsymbol{x}}_{n|0:n-1}^{\mathrm{K},(k_j^*)} + \boldsymbol{K}_n \boldsymbol{e}_n$
      $\boldsymbol{C}_{n|0:n} = \left[I - \boldsymbol{K}_n H_n\left(\boldsymbol{x}_n^{\mathrm{P},(i_j)}\right)\right] \boldsymbol{C}_{n|0:n-1}^{(k_j^*)}$
      $\hat{\boldsymbol{x}}_{n+1|0:n}^{\mathrm{K},(i_j)} = F_n\left(\boldsymbol{x}_n^{\mathrm{P},(i_j)}\right) \hat{\boldsymbol{x}}_{n|0:n}^{\mathrm{K},i_j}$ & $\hat{\boldsymbol{R}}_{v,n}^{(i_j)} = \hat{\boldsymbol{R}}_{v,n}$
      $\boldsymbol{C}_{n+1|0:n}^{(i_j)} = F_n\left(\boldsymbol{x}_n^{\mathrm{P},(i_j)}\right) \boldsymbol{C}_{n|0:n} F_n\left(\boldsymbol{x}_n^{\mathrm{P},(i_j)}\right)^T + \boldsymbol{Q}_u^\mathrm{K}$
      $\boldsymbol{\alpha}_{j,n}^{(i_j)} = \boldsymbol{\alpha}_{j,n-1}^{(k_j^*)} p\left(\boldsymbol{y}_n | \boldsymbol{x}_n^{\mathrm{P},(i_j)}, \hat{\boldsymbol{x}}_{n|0:n-1}^{\mathrm{K},(k_j^*)}\right) p\left(\boldsymbol{x}_n^{\mathrm{P}_j,(i_j)} | \boldsymbol{x}_{n-1}^{\mathrm{P}_j,(k_j^*)}\right)$
    end for
    $i_{j,n}^* = \operatorname*{argmax}_{i_j} \boldsymbol{\alpha}_{j,n}^{(i_j)}$
  end for
  $i_{\mathrm{MAP},n}^* = \operatorname*{argmax}_{i_{j,n}^*} \boldsymbol{\alpha}_{j,n}^{(i_{j,n}^*)}$ & $\hat{\boldsymbol{x}}_n = \{\hat{\boldsymbol{x}}_n^{\mathrm{P}_1,(i_{1,n}^*)}, \hat{\boldsymbol{x}}_n^{\mathrm{P}_2,(i_{2,n}^*)}, \hat{\boldsymbol{x}}_n^{\mathrm{K},(i_{\mathrm{MAP},n}^*)}\}$
end for
```

of the state space where particle filters are used is called the *particle space.* Since the new ABP signal tracker has to estimate both the instantaneous respiratory frequency $\boldsymbol{f}_n^{\mathrm{r}}$ and the instantaneous cardiac frequency $\boldsymbol{f}_n^{\mathrm{c}}$, the dimension of the particle state becomes 2, which results in a quadruple increase of computational burden if the Fast MAM-PF has to be used for the current application. In order to address these two major issues a new ABP signal tracker is proposed, which utilizes a modified version of the Fast MAM-PF. It is called, the *Dual MAM-PF.* The term "Dual" is borrowed from Dual Kalman filters, in which the state is divided into two portions and each portion is estimated separately assuming that the other portion is known and equal to the currently estimate value. While the Fast MAM-PF treats a two-dimensional particle space as a whole, the Dual MAM-PF partitions the two-dimensional particle space into two one-dimensional particle spaces assuming independence between two particle space variables, which are the instantaneous respiratory frequency $\boldsymbol{f}_n^{\mathrm{r}}$ and the instantaneous cardiac frequency $\boldsymbol{f}_n^{\mathrm{c}}$.

All ABP signals included in this study were recorded from spontaneously breathing subjects. Therefore, the ABP signal tracker has to track both the instantaneous respiratory frequency $\boldsymbol{f}_n^{\mathrm{r}}$ and the instantaneous cardiac frequency $\boldsymbol{f}_n^{\mathrm{c}}$ along with the signal morphology.

Suppose that the state vector $\boldsymbol{x}$ can be partitioned as follows,

$$\boldsymbol{x}_n = \begin{bmatrix} \boldsymbol{x}_n^{\mathrm{P}} \\ \boldsymbol{x}_n^{\mathrm{K}} \end{bmatrix} \tag{9.15}$$

where $\boldsymbol{x}_n^{\mathrm{P}}$ represents the particle state and $\boldsymbol{x}_n^{\mathrm{K}}$ the Kalman state. The particle state is the portion of the state where particle filters are used as defined earlier while the Kalman state is the portion of the state where extended Kalman filters are used. The state variables whose posterior distributions are known to be multi-modal belong to the particle state while those whose posterior distributions are known to be Gaussian or uni-modal belong to the Kalman state. In [147] it was demonstrated that the posterior distribution of the instantaneous frequency of a multi-harmonic signal is truly multi-modal. Given the state-space model in (9.1)–(9.4), instantaneous respiratory frequency $\boldsymbol{f}_n^{\mathrm{r}}$ and the instantaneous cardiac frequency $\boldsymbol{f}_n^{\mathrm{c}}$ are the particle state variables and the sinusoidal coefficients such as $\boldsymbol{r}_{\cdot,k,n}$, $\boldsymbol{c}_{\cdot,k,n}$, and $\boldsymbol{m}_{\cdot,k,j,n}$ are the Kalman state variables. Assuming that the particle state variables are independent of each other the particle state $\boldsymbol{x}_n^{\mathrm{P}}$ can be partitioned further as,

$$\boldsymbol{x}_n^{\mathrm{P}} = \begin{bmatrix} \boldsymbol{x}_n^{\mathrm{P}_1} \\ \boldsymbol{x}_n^{\mathrm{P}_2} \end{bmatrix} \tag{9.16}$$

$$\boldsymbol{x}_{n+1}^{\mathrm{P}_1} = f_{1,n}\left(\boldsymbol{x}_n^{\mathrm{P}_1}, \boldsymbol{u}_n^{\mathrm{P}_1}\right) \tag{9.17}$$

$$\boldsymbol{x}_{n+1}^{\mathrm{P}_2} = f_{2,n}\left(\boldsymbol{x}_n^{\mathrm{P}_2}, \boldsymbol{u}_n^{\mathrm{P}_2}\right) \tag{9.18}$$

where $\boldsymbol{x}_n^{\mathrm{P}_1}$ and $\boldsymbol{x}_n^{\mathrm{P}_2}$ represent the first and second particle state variables, respectively. This partitioning breaks down a two-dimensional particle space $\boldsymbol{x}_n^{\mathrm{P}}$ into two one-dimensional particle spaces. The total posterior distribution is given by,

$$p(\boldsymbol{x}_{0:n}|\boldsymbol{y}_{0:n}) = p(\boldsymbol{x}_{0:n}^{\mathrm{K}}|\boldsymbol{y}_{0:n}, \boldsymbol{x}_{0:n}^{\mathrm{P}})p(\boldsymbol{x}_{0:n}^{\mathrm{P}}|\boldsymbol{y}_{0:n}) \tag{9.19}$$

$$=p(\boldsymbol{x}_{0:n}^{\mathrm{K}}|\boldsymbol{y}_{0:n}, \boldsymbol{x}_{0:n}^{\mathrm{P}})p(\boldsymbol{x}_{0:n}^{\mathrm{P}_1}|\boldsymbol{y}_{0:n})p(\boldsymbol{x}_{0:n}^{\mathrm{P}_2}|\boldsymbol{y}_{0:n}). \tag{9.20}$$

Algorithm 1 provides a complete description of the Dual MAM-PF algorithm, where N_{T} represents the total number of signal samples, N_{p} the number of particles for each one-dimensional particle space, j the particle state variable index, and i_j the particle index of the j^{th} particle state variable. The total number of particles used in the Dual MAM-PF algorithm is $2N_{\mathrm{p}}$ instead of N_{p}^2. At each time step n the Dual MAM-PF searches for the best trajectory of each particle i_j from the previous trajectory. This searching step can be written as,

$$\begin{aligned} k_j^* = \underset{k_j}{\operatorname{argmax}} \ & \boldsymbol{\alpha}_{j,n-1}^{(k_j)} p\Big(\boldsymbol{y}_n | \boldsymbol{x}_n^{\mathrm{P}_j,(i_j)}, \hat{\boldsymbol{x}}_{n|0:n-1}^{\mathrm{K},(k_j)}\Big) \cdots \\ & p\Big(\boldsymbol{x}_n^{\mathrm{P}_j,(i_j)} | \boldsymbol{x}_{n-1}^{\mathrm{P}_j,(k_j)}\Big) \qquad (9.21) \\ \approx \underset{k_j}{\operatorname{argmax}} \ & \boldsymbol{\alpha}_{j,n-1}^{(k_j)} p\Big(\boldsymbol{y}_n | \boldsymbol{x}_n^{\mathrm{P}_j,(i_j)}, \hat{\boldsymbol{x}}_{n|0:n-1}^{\mathrm{K},(i_j)}\Big) \cdots \\ & p\Big(\boldsymbol{x}_n^{\mathrm{P}_j,(i_j)} | \boldsymbol{x}_{n-1}^{\mathrm{P}_j,(k_j)}\Big) \qquad (9.22) \\ = \underset{k_j}{\operatorname{argmax}} \ & \boldsymbol{\alpha}_{j,n-1}^{(k_j)} p\Big(\boldsymbol{x}_n^{\mathrm{P}_j,(i_j)} | \boldsymbol{x}_{n-1}^{\mathrm{P}_j,(k_j)}\Big). \qquad (9.23) \end{aligned}$$

Given the best trajectory for each particle i_j, corresponding Kalman state variables $\boldsymbol{x}^{\mathrm{P}_j,(i_j)}$, i.e. sinusoidal coefficients, are updated utilizing the extended Kalman filter. Then, the MAP estimate of $\boldsymbol{x}_n^{\mathrm{P}_i}$ is obtained based on the value of the coefficient $\boldsymbol{\alpha}_{j,n}^{(i_j)}$ as follows,

$$\begin{aligned} i_{j,n}^* &= \underset{i_j}{\operatorname{argmax}} \ \boldsymbol{\alpha}_{j,n}^{(i_j)} \qquad (9.24) \\ \hat{\boldsymbol{x}}_n^{\mathrm{P}_i} &= \boldsymbol{x}_n^{\mathrm{P}_i,(i_{j,n}^*)} \qquad (9.25) \end{aligned}$$

Since there are two groups of particles i_1 and i_2, we need to select the best estimate of the Kalman state vector $\boldsymbol{x}_n^{\mathrm{K}}$ among two potential estimates: $\boldsymbol{x}_n^{\mathrm{K},(i_{1,n}^*)}$ and $\boldsymbol{x}_n^{\mathrm{K},(i_{2,n}^*)}$.

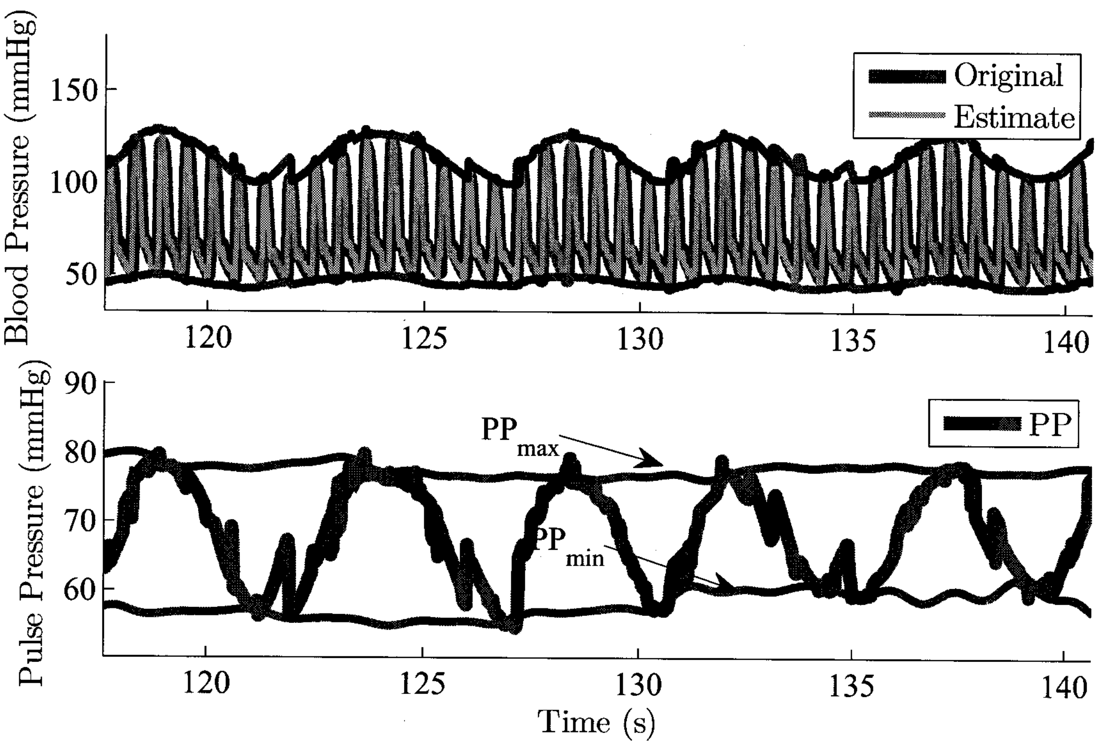


Figure 9.1: Top: Original ABP signal (red) and its estimate (green) with automatically computed envelopes (blue). Bottom: Automatically computed PP signal (red) and its envelopes (blue).

The actual estimate of the Kalman state vector $\boldsymbol{x}_n^{\mathrm{K}}$ can be selected as follows,

$$i^*_{\mathrm{MAP},n} = \begin{cases} i^*_{1,n} & \boldsymbol{\alpha}_{1,n}^{(i^*_{1,n})} \geq \boldsymbol{\alpha}_{2,n}^{(i^*_{2,n})} \\ i^*_{2,n} & \boldsymbol{\alpha}_{1,n}^{(i^*_{1,n})} < \boldsymbol{\alpha}_{2,n}^{(i^*_{2,n})}. \end{cases} \tag{9.26}$$

$$\hat{\boldsymbol{x}}_n^{\mathrm{K}} = \boldsymbol{x}_n^{\mathrm{K},(i^*_{\mathrm{MAP},n})} \tag{9.27}$$

Then, the estimate of the entire state $\boldsymbol{x}_n$ can be expressed as,

$$\hat{\boldsymbol{x}}_n = \{\hat{\boldsymbol{x}}_n^{\mathrm{P}_1,(i^*_{1,n})}, \hat{\boldsymbol{x}}_n^{\mathrm{P}_2,(i^*_{2,n})}, \hat{\boldsymbol{x}}_n^{\mathrm{K},(i^*_{\mathrm{MAP},n})}\} \tag{9.28}$$

9.2.4 ABP Signal Envelope Estimation

Given the estimated signal parameters in (9.5)–(9.13), it is possible to estimate the upper envelope, $\boldsymbol{e}_{\mu,n}$, and lower envelope, $\boldsymbol{e}_{\ell,n}$, of ABP signals by following steps below,

$$\boldsymbol{\theta}^{\mathrm{c}}_{\max,n} = \arg\max_{\theta} \sum_{k=1}^{N^{\mathrm{c}}_{\mathrm{h}}} \boldsymbol{\rho}_{k,n} \left[\boldsymbol{c}_{1,k,n} \cos\left(k\theta\right) + \boldsymbol{c}_{2,k,n} \sin\left(k\theta\right)\right] \tag{9.29}$$

$$\boldsymbol{\theta}^{\mathrm{c}}_{\min,n} = \arg\min_{\theta} \sum_{k=1}^{N^{\mathrm{c}}_{\mathrm{h}}} \boldsymbol{\rho}_{k,n} \left[\boldsymbol{c}_{1,k,n} \cos\left(k\theta\right) + \boldsymbol{c}_{2,k,n} \sin\left(k\theta\right)\right] \tag{9.30}$$

$$\boldsymbol{\kappa}_{\max,k,n} = \boldsymbol{c}_{1,k,n} \cos\left(k\boldsymbol{\theta}^{\mathrm{c}}_{\max,n}\right) + \boldsymbol{c}_{2,k,n} \sin\left(k\boldsymbol{\theta}^{\mathrm{c}}_{\max,n}\right) \tag{9.31}$$

$$\boldsymbol{\kappa}_{\min,k,n} = \boldsymbol{c}_{1,k,n} \cos\left(k\boldsymbol{\theta}^{\mathrm{c}}_{\min,n}\right) + \boldsymbol{c}_{2,k,n} \sin\left(k\boldsymbol{\theta}^{\mathrm{c}}_{\min,n}\right) \tag{9.32}$$

$$\boldsymbol{e}_{\mu,n} = \boldsymbol{\gamma}_n + \sum_{k=1}^{N^{\mathrm{c}}_{\mathrm{h}}} \boldsymbol{\rho}_{k,n} \boldsymbol{\kappa}_{\max,k,n} \tag{9.33}$$

$$\boldsymbol{e}_{\ell,n} = \boldsymbol{\gamma}_n + \sum_{k=1}^{N^{\mathrm{c}}_{\mathrm{h}}} \boldsymbol{\rho}_{k,n} \boldsymbol{\kappa}_{\min,k,n} \tag{9.34}$$

where $\arg\max_x f(x)$ and $\arg\min_x f(x)$ are operators to obtain the value of x for which $f(x)$ attains its maximum and minimum values, respectively. The top plot in Fig. 9.1 shows a five respiratory cycle period of an ABP signal $\boldsymbol{y}_n$ (thick red), its estimate $\hat{\boldsymbol{y}}_n$ (thin green), and its estimated envelopes $\boldsymbol{e}_{\mu,n}$ and $\boldsymbol{e}_{\ell,n}$ (blue), which are described in (9.33) and (9.34).

9.2.5 Pulse Pressure Signal Envelope Estimation

The pulse pressure (PP) signal is the difference between the upper envelope $\boldsymbol{e}_{\mu,n}$ and lower envelope $\boldsymbol{e}_{\ell,n}$ of the ABP signal. The bottom plot in Fig. 9.1 shows an exemplary PP signal oscillating roughly at the respiratory rate. Within each respiratory cycle PP reaches its maximum ($\mathrm{PP}_{\max}$) and minimum ($\mathrm{PP}_{\min}$) values, which are two critical parameters to compute the PPV index. Given the estimated signal parameters in (9.5)–(9.13), one can compute the continuous equivalents of $\mathrm{PP}_{\max}$ and $\mathrm{PP}_{\min}$. They are the upper, $\boldsymbol{\varepsilon}_{\mu,n}$, and lower, $\boldsymbol{\varepsilon}_{\ell,n}$, envelopes of the PP signal. The upper envelope $\boldsymbol{\varepsilon}_{\mu,n}$ is the continuous estimate of $\mathrm{PP}_{\max}$ and the lower envelope $\boldsymbol{\varepsilon}_{\ell,n}$ that of $\mathrm{PP}_{\min}$. The $\boldsymbol{\varepsilon}_{\mu,n}$ and $\boldsymbol{\varepsilon}_{\ell,n}$ values can be estimated as described below,

$$\boldsymbol{\varrho}_{k,n} = \sum_{j=1}^{N_\mathrm{h}^\mathrm{r}} \boldsymbol{m}_{1,k,j,n} \cos(j\theta) + \boldsymbol{m}_{2,k,j,n} \sin(j\theta) \tag{9.35}$$

$$\boldsymbol{\theta}^\mathrm{r}_{\max,n} = \arg\max_{\theta} \sum_{k=1}^{N_\mathrm{h}^\mathrm{c}} (1 + \boldsymbol{\varrho}_{k,n}) (\boldsymbol{\kappa}_{\max,k,n} - \boldsymbol{\kappa}_{\min,k,n}) \tag{9.36}$$

$$\boldsymbol{\theta}^\mathrm{r}_{\min,n} = \arg\min_{\theta} \sum_{k=1}^{N_\mathrm{h}^\mathrm{c}} (1 + \boldsymbol{\varrho}_{k,n}) (\boldsymbol{\kappa}_{\max,k,n} - \boldsymbol{\kappa}_{\min,k,n}) \tag{9.37}$$

$$\boldsymbol{\varrho}_{\max,k,n} = \sum_{j=1}^{N_\mathrm{h}^\mathrm{r}} \boldsymbol{m}_{1,k,j,n} \cos\left(j\boldsymbol{\theta}^\mathrm{r}_{\max,n}\right) + \boldsymbol{m}_{2,k,j,n} \sin\left(j\boldsymbol{\theta}^\mathrm{r}_{\max,n}\right) \tag{9.38}$$

$$\boldsymbol{\varrho}_{\min,k,n} = \sum_{j=1}^{N_\mathrm{h}^\mathrm{r}} \boldsymbol{m}_{1,k,j,n} \cos\left(j\boldsymbol{\theta}^\mathrm{r}_{\min,n}\right) + \boldsymbol{m}_{2,k,j,n} \sin\left(j\boldsymbol{\theta}^\mathrm{r}_{\min,n}\right) \tag{9.39}$$

$$\boldsymbol{\varepsilon}_{\mu,n} = \sum_{k=1}^{N_\mathrm{h}^\mathrm{c}} (1 + \boldsymbol{\varrho}_{\max,k,n}) (\boldsymbol{\kappa}_{\max,k,n} - \boldsymbol{\kappa}_{\min,k,n}) \tag{9.40}$$

$$\boldsymbol{\varepsilon}_{\ell,n} = \sum_{k=1}^{N_\mathrm{h}^\mathrm{c}} (1 + \boldsymbol{\varrho}_{\min,k,n}) (\boldsymbol{\kappa}_{\max,k,n} - \boldsymbol{\kappa}_{\min,k,n}) \tag{9.41}$$

where $1 + \boldsymbol{\varrho}_{k,n}$ is equal to $\boldsymbol{\rho}_{k,n}$ and $\boldsymbol{\varepsilon}_{\mu,n}$ and $\boldsymbol{\varepsilon}_{\ell,n}$ are the continuous estimates of the $\mathrm{PP}_{\max}$ and $\mathrm{PP}_{\min}$, respectively. The blue lines in the bottom plot in Fig. 9.1 represent the upper $\boldsymbol{\varepsilon}_{\mu,n}$ and lower $\boldsymbol{\varepsilon}_{\ell,n}$ envelopes of the PP signal, which are obtained by following the method described above.

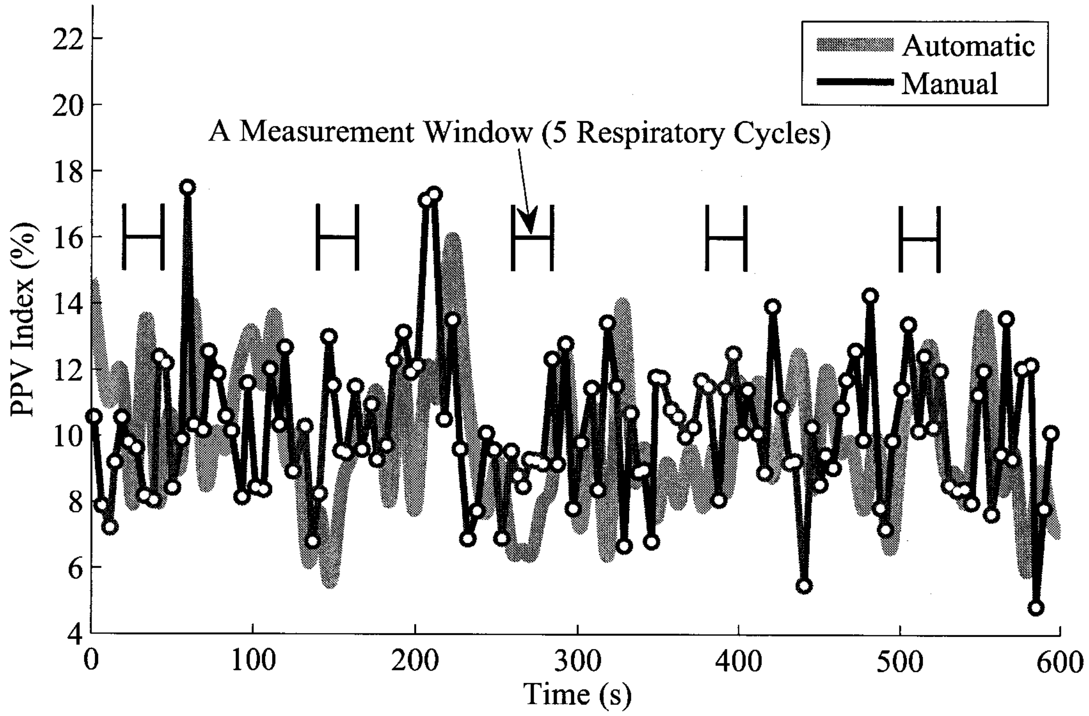


Figure 9.2: Automatic PPV index (green) and manual PPV index (red) over the entire ABP signal duration (10 minutes). One PPV index measurement is computed from each measurement window, which is a time period of 5 respiratory cycles.

9.2.6 Pulse Pressure Variation Calculation

Given the $\varepsilon_{\mu,n}$ and lower $\varepsilon_{\ell,n}$ values, it is straightforward to calculate the PPV index. It can be computed as follows,

$$\text{PPV}(\%) = 100 \times \frac{\varepsilon_{\max} - \varepsilon_{min}}{(\varepsilon_{\max} + \varepsilon_{\min})/2} \tag{9.42}$$

This new PPV index is different from the traditional PPV index described in (8.1) because the new one is continuous in time while the traditional one can be obtained only once per each respiratory cycle.

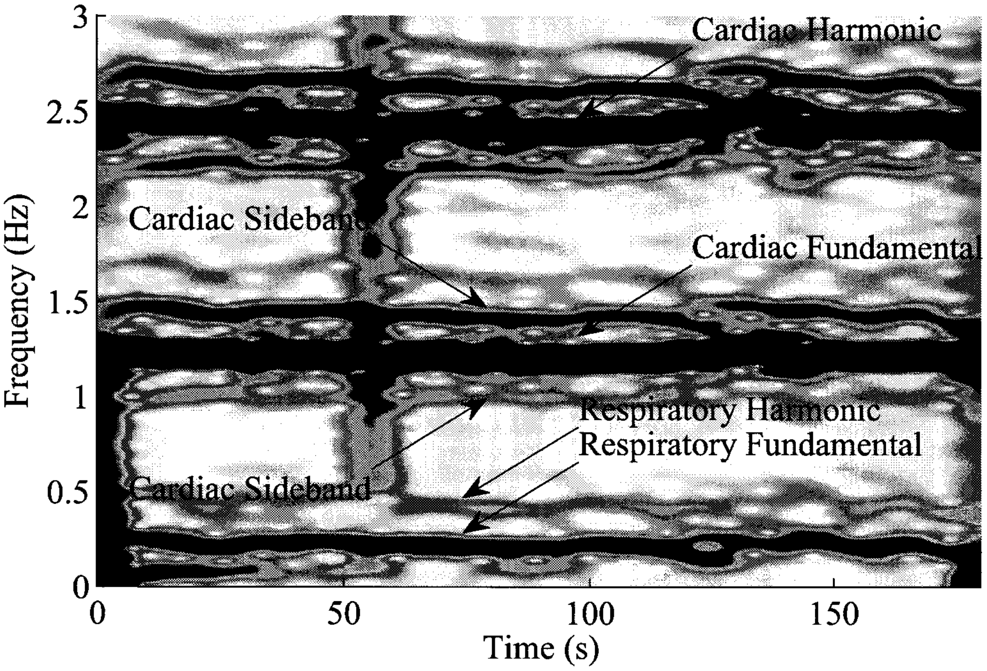


Figure 9.3: Spectrogram of one of the 12 ABP signals recorded from spontaneously breathing subjects.

Fig. 9.2 illustrates an example of the automatically computed continuous PPV index (thick green) and the manually obtained discrete PPV index (thin red) of a real 10 min ABP signal. Each hollow white dot represents a "discrete" PPV index, which can be obtained once per each respiratory cycle.

9.3 Algorithm Assessment

9.3.1 Assessment Data

The Massachusetts General Hospital (MGH) waveform database on PhysioNet is a comprehensive collection of electronic recordings of hemodynamic and electrocardiographic waveforms subjects in critical care units [130,131]. Among 250 subjects 12 subjects who breathed spontaneously were identified by visually inspecting the

Table 9.1: Summary of user-specified parameters for the PPV index tracker.

Name	Symbol	Value
No. particles	$2N^{\mathrm{p}}$	500
No. cardiac components	N^{c}	10
No. respiratory components	N^{r}	3
Minimum respiratory rate	$f^{\mathrm{r}}_{\mathrm{min}}$	6/60 Hz
Maximum respiratory rate	$f^{\mathrm{r}}_{\mathrm{max}}$	30/60 Hz
Minimum heart rate	$f^{\mathrm{c}}_{\mathrm{min}}$	50/60 Hz
Maximum heart rate	$f^{\mathrm{c}}_{\mathrm{max}}$	140/60 Hz
Measurement noise variance	r	var(y)/1e3
Respiratory frequency variance	$q_{f^{\mathrm{r}}}$	1e-6 T_s
Cardiac frequency variance	$q_{f^{\mathrm{c}}}$	1e-6 T_s
Respiratory amplitude variance	q_a, q_b	var(y)1e-6T_s
Modulation factor amplitude variance	q_c, q_d	var(y)1e-8T_s
Cardiac amplitude variance	q_e, q_f	var(y)1e-6T_s
Initial respiratory amplitude	u_a, u_b	std(y)/1e1
Initial modulation factor amplitude	u_c, u_d	std(y)/1e3
Initial cardiac amplitude	u_e, u_f	std(y)/1e1

spectrogram and time-series of ABP signals. Fig. 9.3 shows the spectrogram of one of the 12 ABP signals. Each ABP signal is 10 min long and the total duration of the 12 ABP signals was 2 hours. The original sample rate f_{s} of the signals was 360 Hz, but they were downsampled by a factor of 9, so that the final sample rate f_{s} was 40 Hz.

The number of cardiac components N^{c} was 5 and that of respiratory components N^{r} 2. The total number of particles $2N^{\mathrm{p}}$ was 500. Table 9.1 lists the parameter values used for the PPV index estimator.

9.3.2 Manual PPV Annotations (Current Standard)

The peaks and troughs of the ABP signals were manually annotated and the PPV indices (current standard) were calculated as defined in (8.1). They are referred to as manual PPV indices $\mathrm{PPV}_{\mathrm{manu}}$. $\mathrm{PPV}_{\mathrm{auto}}$ represents PPV indices obtained using the proposed PPV index tracking algorithm.

9.3.3 Statistical Analysis

The statistical analysis used 5 PPV index measurements for each subject, and each measurement was separated by 2 minutes. Each PPV index measurement is an averaged value over 5 respiratory cycles. Fig. 9.2 shows the 2 min apart measurement periods. The proposed PPV index tracking algorithm was assessed by calculating the agreement (mean ± standard deviation) between $\mathrm{PPV}_{\mathrm{auto}}$ and $\mathrm{PPV}_{\mathrm{manu}}$ measurements and using Bland-Altman analysis.

A Bland-Altman plot is a statistical visualization method that is often used in the assessment of PPV estimation algorithms in order to determine the agreement between two different PPV estimates. It has the difference $\Delta\mathrm{PPV}$ between $\mathrm{PPV}_{\mathrm{auto}}$ and $\mathrm{PPV}_{\mathrm{manu}}$ measurements on the y-axis and the $\mathrm{PPV}_{\mathrm{manu}}$ measurements on the x-axis. It shows the overall accuracy of estimation and any estimation bias or trend versus $\mathrm{PPV}_{\mathrm{manu}}$ measurements. The automatic estimation algorithm was compared against the current standard (manual annotations) using this Bland-Altman plot.

9.4 Results and Discussion

Fig. 9.4 depicts the Bland-Altman plot of the 12 subjects. There are 5 PPV measurements available per each subject. All of $\mathrm{PPV}_{\mathrm{auto}}$ measurements were in agreement with $\mathrm{PPV}_{\mathrm{manu}}$ measurements within $\pm 3.5\%$ accuracy.

Table 9.2 lists the mean ± standard deviation of 5 $\mathrm{PPV}_{\mathrm{manu}}$ and $\mathrm{PPV}_{\mathrm{auto}}$

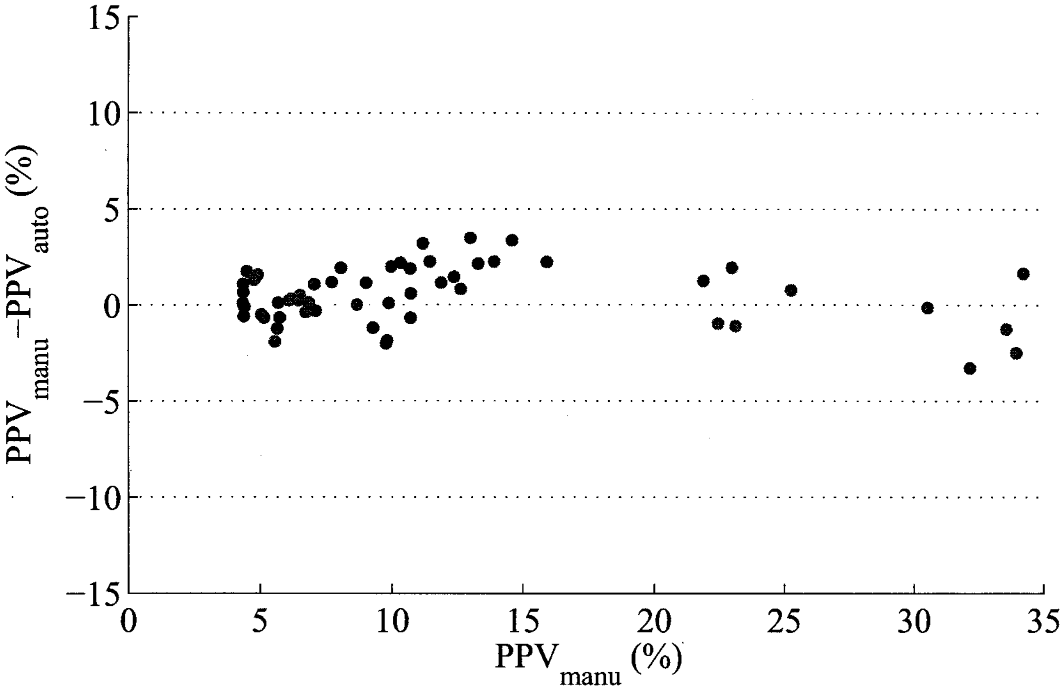


Figure 9.4: Bland-Altman plot of the 12 subjects.

measurements for each subject. The second column is for PPV_{manu} and the third column for PPV_{auto}.

9.5 Discussion

9.5.1 Algorithm's Advantages

The proposed algorithm is the first automatic method described in the literature especially designed to estimate and track the PPV index in situations involving spontaneous breathing. It is important to note that the proposed algorithm is completely different from the algorithm described in [137], which only worked for mechanically ventilated subjects. As mentioned in the previous chapter, the major algorithm design difference of the proposed algorithm with respect to previously

Table 9.2: Summary of the mean and standard deviation of the PPV_{manu} and PPV_{auto} measurements.

Subject	PPV_{manu} (%)	PPV_{auto} (%)
1	9.8±1.0	8.9±1.0
2	32.9±1.5	33.9±2.3
3	10.3±1.0	10.6±0.8
4	4.6±0.3	3.3±0.3
5	10.3±1.1	9.2±1.3
6	12.9±2.7	11.3±2.0
7	5.9±0.5	6.1±0.5
8	12.5±0.8	10.3±1.5
9	7.6±1.0	6.7±0.9
10	4.8±0.5	5.3±1.3
11	6.5±0.8	6.5±0.6
12	23.1±1.3	22.8±1.8

published algorithms [137, 139] is the fact that the proposed method is based on a statistical state-space model and estimation of the cardiovascular pressure signal. The state-space modeling stage results in an algorithm that is more robust to hemodynamic changes and artifacts. The statistical state-space signal model and associated model parameter estimation algorithm automatically filter out noise and artifact that cannot be captured with the model.

Figs. 9.5–9.6 exemplify a case where signal features that are not physiological in nature are automatically filtered out and as a result PPV_{auto} is more reasonable than PPV_{manu}. The top plot in Fig. 9.5 illustrates 4 respiratory cycles of the ABP signal (red) and its estimate (green). It also shows the manually annotated signal envelopes (black) and the automated computed signal envelopes (light blue). The bottom plot in Fig. 9.5 depicts the PPV_{manu} and PPV_{auto} over the same period. Around at 535 s, the PPV_{manu} value (red) abruptly increases up to 35% while the PPV_{auto} value (green) remains at 8%. Around at 540 s, the PPV_{manu} value returns to 8%. Fig. 9.6 focuses on the time period marked with the black rectangular box in Fig. 9.5. The top plot in Fig. 9.6 shows that the heart beat between 535 s and 535.5 s is contaminated by noise and has an abnormal morphology. As a result, the corresponding $\mathrm{PP}_{\mathrm{manu}}$ shown in the bottom plot reaches a large maximum value ($\mathrm{PP}_{\mathrm{min,manu}}$: 105 mmHg) around at 535 s. However, the automatically computed maximum PP value ($\mathrm{PP}_{\mathrm{min,auto}}$) at the same time is as low as 83 mmHg. This discrepancy between the manual annotation and the proposed automatic method results from the capability of the MAM-PF algorithm, which estimates the ABP signal based on the state-space model. While the original heart beat between 535 s and 535.5 s in Fig. 9.6 is abnormal in a physiological sense, the estimated heart beat over the same time period shows the physiologically expected morphology and location of the heart beat.

9.5.2 Study Limitations

The algorithm's assessment was based on only 12 subjects with pre-recorded ABP data. Additionally, for each subject five PPV estimates were used in the assessment study. This assessment was designed to be an engineering algorithm validation against current standard manual annotations, and not a clinical validation study. Consequently, a clinical validation study assessing the ability of the proposed algorithm to monitor fluid responsiveness in the operating room in situations involving spontaneously breathing subjects still needs to be conducted. This may require the proposed algorithm to be first adopted as part of a commercial system as was the case with the previous automatic PPV algorithm in [137].

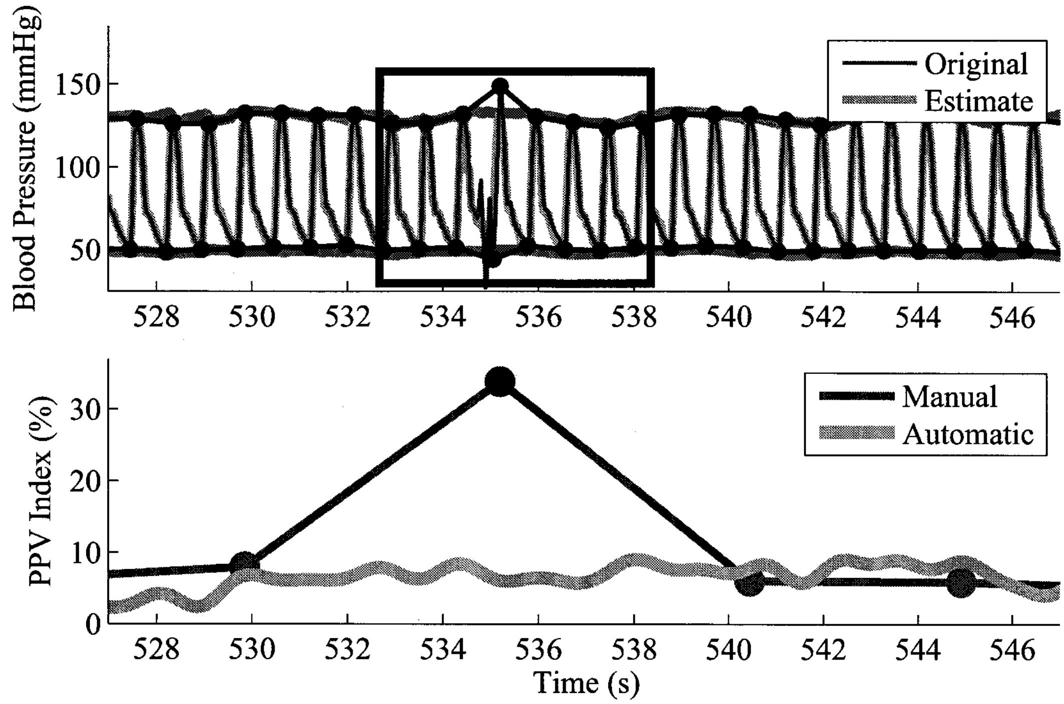


Figure 9.5: Top: Original ABP signal (red) with its manually annotated envelopes (black) and signal estimate (green) with its automatically computed envelopes (purple). Bottom: Manual PPV indices (red) and automatic PPV indices (green) where one of the manual PPV indices has an abnormally high value.

9.6 Summary

This chapter describes the first automatic PPV tracking algorithm for spontaneously breathing subjects. This novel algorithms is based on a statistical state-space model inspired in the underlying cardiovascular and respiratory physiology. This algorithm uses the modified version of the MAM-PF technique for optimal parameter estimation. The assessment results indicate good agreement against the current standard PPV. All of PPV_{auto} measurements were in agreement with PPV_{manu} measurements within ±3.5% accuracy.

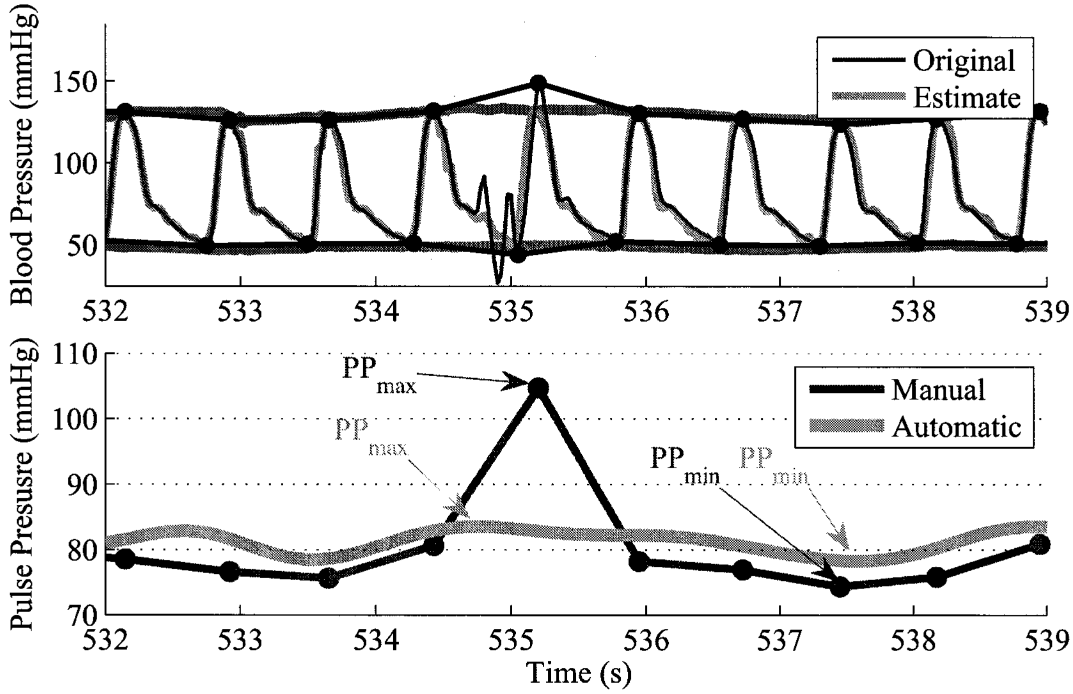


Figure 9.6: Top: Original ABP signal (red) with its manually annotated envelopes (black) and signal estimate (green) with its automatically computed envelopes (purple). Bottom: Manual PP signal (red) and automatic PP signal (green) where the manual PP signal increases momentarily due to a contaminated heart beat in the ABP signal.

Chapter 10

Summary and Conclusions

This dissertation explains the background and significance of tracking rhythmicity in quasi-periodic signals and summarizes various techniques to tackle this problem, that are available in the literature. Chapter 5 summarizes several conventional particle filtering algorithms and provides the details of the proposed particle filtering algorithm, MAM-PF, which overcomes critical issues related with the conventional particle filtering algorithms. Chapters 6–9 discuss how the proposed MAM-PF algorithm can be applied to four practical biomedical applications. Those applications are closely related with various clinical problems ranging from monitoring tremor for people with Parkinson's disease to guiding physicians as to the best fluid therapy for a patient in operating room or critical care environments.

10.1 Significance of Rhythmicity Tracking

Rhythmicity refers to the cyclical pattern that can be represented as a linear combination of time-varying sinusoids, i.e. partials. Rhythmicity tracking is continuous estimation of the frequencies, phases, and amplitudes of the partials. When a quasi-periodic signal contains harmonically related multiple partials, it is called a multi-harmonic signal. Multi-harmonic signals are very common. Examples include the human voice and the electrocardiogram (ECG). The frequencies and amplitudes of the multi-harmonic signal's partials are important features of the signal and it is essential to accurately estimate them for further analysis. There are a few techniques in current use for the purpose of multi-harmonic signal tracking. However, some of those techniques are only capable of tracking the frequency and others do not take advantage of the harmonic relationship between the partials. More importantly, most of them rely on the local stationarity assumption of the signals. This presents an opportunity for new signal processing methods that can accurately estimate the frequency and amplitude of the multi-harmonic signal on a sample-by-sample basis and take into account the harmonic relationship between the partials.

10.2 Advantages of the Proposed MAM-PF

The state-space method is a tool to estimate a valuable but unknown quantity, i.e. state, given a sequence of measurements by representing a physical system as a mathematical model. When this mathematical model is linear, the conventional Kalman filter can be used to *optimally* estimate the state. However, many systems cannot be represented as simple linear state-space models and nonlinearities must be addressed to account for the complexity of the systems. Although variants of the linear Kalman filter such as the extended or unscented Kalman filter (EKF/UKF) have been devised to estimate the state of a nonlinear state-space model, they often yield unreliable and inaccurate estimates. The particle filtering technique is an alternative to the EKF/UKF and many research groups have demonstrated its suitability and versatility for various applications. During the last two decades several variants of the canonical particle filtering technique have been proposed, including the marginalized particle filter (MPF) and the maximum *a posteriori* particle filter (MAP-PF). However, they suffer from two critical issues that have not been well understood in the literature. The proposed MAM-PF is a hybrid particle filtering method that leverages the advantages of the MAP-PF and MPF. It is devised to overcome two well known problems of the particle filtering method: sample degeneracy and sample impoverishment. A drawback of the MAM-PF is its heavy computational load. However, the computational load can be dramatically reduced with an approximation that does not substantially affect performance, so that the computation burden of the MAM-PF is comparable to that of the MPF.

10.3 Applications

The MAM-PF is so versatile that it can be applied to numerous applications. This dissertation discusses four practical biomedical applications. Each application involves a unique type of biomedical signal and corresponding state-space model. With a carefully set state-space model the MAM-PF algorithm can be utilized to estimate and track the signal's valuable parameters such as tremor frequency (Chapter 6), heart rate (Chapter 7), and pulse pressure variation (Chapters 8–9). For each application, the performance of the MAM-PF algorithm was thoroughly assessed against the best current practice using synthetic and real signals. The simulation results demonstrate the outstanding performance of the MAM-PF algorithm. The contributions of the dissertation are not merely technical but clinical too. Tremor tracking in Chapter 6 is a first step toward studying phase-coupling between two tremorous activities, which is critical in the pathogenesis study of movement disorders. Heart rate tracking in Chapter 7 is an important task because rhythmical heart rate variability (HRV) is a useful quantity for medical and diagnostic purposes. Pulse pressure variation (PPV) is a sensitive and specific predictor of fluid responsiveness for patients in need of fluid therapy. Determining the best of fluid therapy for a patient is difficult and clinicians have few clinical signs to guide them. There are few publicly available algorithms that automatically estimate the PPV index accurately and reliably. Especially, the PPV tracking algorithm proposed in Chapter 9 is the first automatic method described in the literature that is specially designed to estimate and track the PPV index in situations involving spontaneous breathing.

10.4 Potential Future Work

The proposed MAM-PF algorithm can be applied to any state-space model whose state can be partitioned into the nonlinear and linear portions. It is advantageous especially when the posterior distribution of any of state parameters is known to be multi-modal. However, for accurate and reliable estimation, the MAM-PF algorithm requires a sufficient number of particles just like other particle filtering techniques and as a result the computational burden can be quite demanding, which makes it unpractical for real-time applications. One possible solution is *parallel computing*, which is a form of computation that carries out many calculations simultaneously. Currently, the MAM-PF algorithm is implemented in a way that at each time index the particles are processed one by one in series. However, using parallel computing, the MAM-PF algorithm can be implemented to process several particles in parallel. For example, if 10 particles are processed simultaneously, the MAM-PF algorithm becomes 10 times faster. With multi-core processors and smart programming the parallel computing based MAM-PF can be implemented and may be useful for various real-time applications where the conventional methods such as the EKF/UKF perform poor or fail.

References Cited

[1] P. Wright, "Short-time fourier transforms and wigner-ville distributions applied to the calibration of power frequency harmonic analyzers," *IEEE Transactions on Instrumentation and Measurement*, vol. 48, no. 2, pp. 475 – 478, Apr. 1999.

[2] F. Z. Y. Q. C. G. Bi, "Adaptive harmonic fractional fourier transform," in *Circuits and Systems, 2000. Proceedings. ISCAS 2000 Geneva. The 2000 IEEE International Symposium on*, vol. 5, Geneva, May 2000, pp. 45–48.

[3] Y. K. Jang and J. F. Chicharo, "Adaptive iir comb filter for harmonic signal cancellation," *Inter. J. Electronics*, vol. 75, no. 2, pp. 241–250, Aug. 1993.

[4] D. Charalampidis, "Fast-varying pitch tracking: A new approach to speech modeling," in *Proceedings of SPIE - The International Society for Optical Engineering*, vol. 6560. Orlando, FL: SPIE, Apr 2007, p. 65600P, speech modeling;Pitch detection;Pitch tracking;Warping;. [Online]. Available: http://dx.doi.org/10.1117/12.719070

[5] A. de Cheveigné and H. Kawahara, "Yin, a fundamental frequency estimator for speech and music," *The Journal of Acoustical Society of America*, vol. 111, no. 4, pp. 1917–1930, Apr. 2002.

[6] B. Doval and X. Rodet, "Fundamental frequency estimation and tracking using maximumlikelihood harmonic matching and hmms," in *IEEE International Conference on Acoustics, Speech, and Signal Processing, 1993. ICASSP-93.*, vol. 1, Minneapolis, MN, USA, Apr 1993, pp. 221–224.

[7] R. J. McAulary and T. F. Quatieri, "Speech analysis/synthesis based on a sinusoidal representation," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. ASSP-34, no. 4, pp. 744–754, Aug 1986.

[8] R. V. Allen, "Automatic earthquake recognition and timing from single traces," *Bulletin of the Seismological Society of America*, vol. 68, no. 5, pp. 1521–1532, Oct. 1978.

[9] A. J. Lim and W. D. Winters, "Practical method for automatic real-time eeg sleep state analysis," *IEEE Transactions on Biomedical Engineering*, vol. BME-27, no. 4, pp. 212–220, Apr. 1980.

[10] C. Nguyen and K. Srinivasan, "New technique for rapid tracking of frequency deviations based on level crossings," in *IEEE Power Engineering Society 1984 Winter Meeting.* Dallas, TX: IEEE, New York, NY, 1984, pp. 84WM 103–108.

[11] N. J. Miller, "Pitch detection by data reduction," in *IEEE Symp on Speech Recognition, Pap and Abstr.* Pittsburgh, PA: IEEE, New York, NY, Apr. 1974, pp. 122–130.

[12] B. Srivastava, "Some results on the detection of sinusoidal signals by zero crossing method in the presence of band limited gaussian noise," *Int. J. Electronics*, vol. 27, no. 3, pp. 283–288, Sep. 1969.

[13] M. Ito and R. Donaldson, "Zero- crossing measurements for analysis and recognition of speech sounds," *IEEE Trans Audio Electroacoust*, vol. AU-19, no. 3, pp. 235–242, Sep. 1971.

[14] M. Kobayashi, H. Osawa, N. Morinaga, and T. Namekawa, "On the mean frequency measurement system using correlation detection," *IEEE Transactions on Aerospace and Electronic Systems*, vol. AES-10, no. 3, pp. 364–372, May 1974.

[15] L. R. Rabiner and R. W. Schafer, *Digital Procesing of Speech Signals.* Englewood Cliffs, NJ: Prentice Hall, 1978.

[16] A. V. Oppenheim and R. W. Schafer, *Discrete-Time Signal Processing*, 2nd ed. Upper Saddle River, NJ: Prentice Hall, 1999.

[17] B. Boashash, "Estimating and interpreting the instantaneous frequency of a signal—Part 1: Fundamentals," *Proceedings of the IEEE*, vol. 80, no. 4, pp. 520–538, Apr 1992.

[18] B. Boashash, P. O'shea, and M. Arnold, "Algorithms for instantaneous frequency estimation: a comparative study," in *Proc. SPIE Conf. Advanced Algorithm and Architectures for Signal Processing V*, vol. 1348, San Diego, CA, July 1990.

[19] S. Kay, "Statistically and computationally efficient frequency estimation," in *ICASSP '88*, New York, 1988, pp. 2292–2295.

[20] S. Kim and J. McNames, "Tracking tremor frequency in spike trains using the extended Kalman smoother," *IEEE Transactions on Biomedical Engineering*, vol. 53, no. 8, pp. 1569–1577, Aug 2006.

[21] R. K. POTTER, G. A. KOPP, and H. C. GREEN, *Visible Speech*, 1st ed. New York: D Van Nostrand, 1947.

[22] E. P. Wigner, "On the quantum correction for thermodynamic equilibrium," *Phys. Rev.*, vol. 40, pp. 749–759, 1932.

[23] D. Gabor, "Theory of communication," *Journal of IEEE (London)*, vol. 93, pp. 429–457, 1946.

[24] J. Ville, "Théorie et applications de la notion de signal analytique," *Cables et Transmission*, vol. 2A, pp. 61–74, 1948.

[25] C. Page, "Instantaneous power spectra," *Journal of Appl. Phys.*, vol. 23, no. 1, pp. 103–106, Jan. 1952.

[26] W. D. Mark, "Spectral analysis of the convolution and filtering of non-stationary stochastic processes," *J. Sound Vib.*, vol. 11, no. 1, pp. 19–63, Jan. 1970.

[27] L. Cohen, "Generalized phase-space distribution functions," *J. Math. Phys.*, vol. 7, pp. 781–786, May 1966.

[28] L. Cohen, "Time-frequency distributions - a review," in *Proceedings of the IEEE*, vol. 77, no. 7, Jul. 1989, pp. 941–981.

[29] T. Claasen and W. Mecklenbrauker, "The wigner distribution-a tool for time-frequency signal analysis; part i: continous-time signals," *Philips J. Res.*, vol. 35, pp. 217–250, 1980.

[30] T. Claasen and W. Mecklenbrauker, "The wigner distribution-a tool for time-frequency signal analysis; part ii: dicrete-time signals," *Philips J. Res.*, vol. 35, pp. 276–300, 1980.

[31] T. Claasen and W. Mecklenbrauker, "The wigner distribution-a tool for time-frequency signal analysis; part iii: relations with other time-frequency signal transformations," *Philips J. Res.*, vol. 35, pp. 372–389, 1980.

[32] B. Boashash and F. Rodriguez, "Recognition of time-varying signals in the time-frequency domain by means of the wigner distribution," in *Proc. IEEE ICASSP 84*, 1984, pp. 22.5/1–4.

[33] B. Boashash and E. de Bazelaire, "Pattern recognition in the time-frequency domain by means of the wigner-ville distribution," in *Proc. GRETSI 9^{th} Colloq.*, 1983, pp. 879–884.

[34] B. Boashash, "Représentation temps-fréquence," in *Soc. Nat. ELF Aquitaine.* Pau, France: Publ. Recherches, 1978, pp. 373–378.

[35] C. Janse and J. Kaizer, "Time-frequency distributions of loudspeakers: the application of the wigner distribution," *J. Audio Eng. Soc.*, vol. 31, pp. 198–223, 1983.

[36] D. Chester, F. Tayloer, and M. Doyle, "The wigner distribution in speech processing applications," *J. Franklin Inst.*, vol. 318, no. 6, pp. 415–430, Dec. 1984.

[37] D. Chester and J. Wilbur, "Time and spatial varying cam and ai signal analysis using the wigner distribution," in *Proceedings - ICASSP, IEEE In-*

ternational Conference on Acoustics, Speech and Signal Processing. Tampa, FL: IEEE, New York, NY, 1985, pp. 1045–1048.

[38] W. Martin and P. Flandrin, "Wigner-ville spectral analysis of nonstationary processes," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. ASSP-33, no. 6, pp. 1461–1470, Dec. 1985.

[39] S. Pola, A. Macerata, M. Emdin, and C. Marchesi, "Estimation of the power spectral density in nonstationary cardiovascular time series: Assessing the role of the time-frequency representations (tfr)," *IEEE Transactions on Biomedical Engineering*, vol. 43, no. 1, pp. 46–59, 1996.

[40] H.-L. Chan, H.-H. Huang, and J.-L. Lin, "Time-frequency analysis of heart rate variability during transient segments," *Annals of Biomedical Engineering*, vol. 29, no. 11, pp. 983–996, Nov. 2001.

[41] G. Jones and B. Boashash, "Time-frequency analysis of multicomponent signals," in *Signal Proceeding V: Theories and Applications*, L. Torres, E. Masgrau, and M. Lagunas, Eds. Elsevier Science, 1990, pp. 141–144.

[42] G. Jones and B. Boashash, "Instantaneous frequency, instantaneous bandwith and the analysis of multicomponent signals," in *IEEE International Conference on Acoustics, Speech, and Signal Processing, 1990.*, vol. 5, Albuquerque, NM, USA, Apr. 1990, pp. 2467–2470.

[43] B. Widrow and M. J. Hoff, "Adaptive switching circuits," in *Proc. IRE Wescon Conf. Rec.* IEEE, New York, NY, 1960, pp. 96–104, part 4.

[44] E. Wiener, N. Hopf, "On a class of singular integral equations," in *Proc. Prussian Acad. Math-Phys. Ser.*, 1931, pp. 696–.

[45] B. Widrow, J. John R. Glover, J. M. McCool, J. Kaunitz, C. S. Williams, R. H. Hearn, J. R. Zeidler, J. Eugene Dong, and R. C. Goodlin, "Adaptive noise cancellation: Principles and applications," in *Proceedings of the IEEE*, vol. 63, no. 12, Dec. 1975, pp. 1692–1717.

[46] J. R. Treichler, "Transient and convergent behavior of the adaptive line enhancer," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. ASSP-27, no. 1, pp. 53–62, Feb. 1979.

[47] J. R. Zeidler, E. H. Satorius, D. M. Chabries, and H. T. Wexler, "Adaptive enhancement of multiple sinusoids in uncorrelated noise," *IEEE Transactions on Acoustics, Speech, And Signal Processing*, vol. ASSP-26, no. 3, pp. 240–254, Jun. 1978.

[48] D. Rao and S.-Y. Kung, "Adaptive notch filtering for the retrieval of sinusoids in noise," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 32, no. 4, pp. 791–802, Aug. 1984.

[49] B. Friedlander, "A recursive maximum likelihood algorithm for arma line enhancement," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 30, no. 4, pp. 651–657, Aug. 1982.

[50] M. Mojiri, M. Karimi-Ghartemani, and A. Bakhshai, "Time-domain signal analysis using adaptive notch filter," *IEEE Transactions on Signal Processing*, vol. 55, no. 1, pp. 85–93, Jan. 2007.

[51] P. Tichavský and A. Nehorai, "Comparative study of four adaptive frequency trackers," *IEEE Transactions on Signal Processing*, vol. 45, no. 6, pp. 1473–1484, Jun 1997.

[52] P. Tichavský and P. Handel, "Two algorithms for adaptive retrieval of slowly time-varying multiple cisoids in noise," *IEEE Transactions on Signal Processing*, vol. 43, no. 5, pp. 1116–1127, May 1995.

[53] S. J. Elliott and P. Darlington, "Adaptive cancellation of periodic, synchronously sampled interference," *IEE Trans. Acoustic Speech Signal Processing*, vol. ASSP-33, no. 3, pp. 715–717, Jun. 1985.

[54] A. Nehorai and B. Porat, "Adaptive comb filtering for harmonic signal enhancement," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 34, no. 5, pp. 1124–1138, Oct. 1986.

[55] A. Nehorai and B. Porat, "Adaptive comb filtering for harmonic signal enhancement," in *IEEE International Conference on Acoustics, Speech, and Signal Processing '85*, vol. 10. Tampa, FL, USA: IEEE, New York, NY, Apr. 1985, pp. 335–338.

[56] D. Veeneman and B. Mazor, "A fully adaptive comb filter for enhancing block-coded speech," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 37, no. 6, pp. 955–957, Jun. 1989.

[57] N. Chernoguz, "A single-parameter adaptive comb filter," in *2001 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '01)*, vol. 6. IEEE, New York, NY, May 2001, pp. 3749–3752.

[58] K. Nishi and S. Ando, "Optimal comb filter for time-varying harmonics extraction," *IEICE Transactions on Fundamentals of Electronics, Communications, and Computer Sciences*, vol. E81-A, no. 8, pp. 1622–1627, Aug. 1998.

[59] H. Sun and L. Shue, "Analysis of an adaptive filter-bank for harmonic measurement and estimation," in *Proceedings - IEEE International Symposium on Circuits and Systems (ISCAS 2007)*, New Orleans, LA, U.S.A., May 2007, pp. 2427–2430.

[60] D. Cyrill, J. McNames, and M. Aboy, "Adaptive comb filters for quasiperiodic physiologic signals," in *Proceedings of the 25th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, vol. 3. IEEE, New York, NY, Sep. 2003, pp. 2439–2442.

[61] L. R. Rabiner, M. J. Cheng, A. E. Rosenberg, and C. A. McGonegal, "A comparative performance study of several pitch detection algorithms," *IEEE Transactions on Acoustics, Speech, and Signal*, vol. ASSP-24, no. 5, pp. 399–418, October 1976.

[62] D. Gerhard, "Pitch extraction and fundamental frequency: History and current techniques," Dept. of Computer Science, University of Regina, Department of Computer Science, University of Regina, Regina, Saskatchewan, Canada, Tech. Rep. TR-CS 2003-06, November 2003.

[63] J. Tabrikian, S. Dubnov, and Y. Dickalov, "Maximum a-posteriori probability pitch tracking in noisy environments using harmonic model," *IEEE TRANSACTIONS ON SPEECH AND AUDIO PROCESSING*, vol. 12, no. 1, pp. 76–87, Jan 2004.

[64] W. J. Hess, *Advances in Speech Signal Processing.* Dekker, 1992, ch. Pitch Voicing Determination, pp. 3–48.

[65] C. Wang and S. Seneff, "Robust pitch tracking for prosodic modeling in telephone speech," in *IEEE International Conference on Acoustics, Speech, and Signal Processing, 2000*, vol. 3. Istanbul, Turkey: IEEE, 2000, pp. 1343–1346.

[66] J. Droppo and A. Acero, "Maximum a posteriori pitch tracking," in *International Conference on Spoken Language Processing - 1998*, 1998, pp. 943–946.

[67] B. Atal and J. R. Remde, "A new model of lpc excitation for producting natural-sounding speech at low bit rate," in *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing*, Paris, 1982, p. 614.

[68] L. B. Almeida and F. M. Silva, "Variable-frequency synthesis: An improved harmonic coding scheme," in *Proc. Int. Conf. Acoust., Speech, and Signal Processing*, San Diego, CA, 1984, p. 27.5.1.

[69] P. Hedelin, "A tone-oriented voice-excited vocoder," in *Proc. Int. Conf. Acoust. Speech, Signal Processing*, Atlanta, GA, 1981, p. 205.

[70] R. E. Kalman, "A new approach to linear filtering and prediction problems," *Transactions of the ASME - Journal of Basic Engineering*, vol. 82, pp. 35–45, 1960.

[71] D. R. Polk and S. C. Gupta, "Quasi-optimum digital phase-locked loops," *IEEE Transactions on Communications*, vol. 21, no. 1, pp. 75–82, Jan 1973.

[72] C. N. Kelly and S. C. Gupta, "The digital phase-locked loop as a near-optimum FM demodulator," *IEEE Transactions on Communications*, vol. 20, no. 3, pp. 406–411, Jun 1972.

[73] S. F. Schmidt, "Computational techniques in Kalman filtering," NATO Advisory Group for Aerospace Research and Development, AGARDograph 139, Feb 1970, in Theory and Applications of Kalman Filtering.

[74] P. J. Parker and B. D. Anderson, "Frequency tracking of nonsinusoidal periodic signals in noise," *Signal Processing*, vol. 20, no. 2, pp. 127–152, Jun 1990.

[75] T. Kailath, A. H. Sayed, and B. Hassibi, *Linear Estimation.* Prentice Hall, 2000.

[76] L. R. Rabiner and B. H. Juang, "An introduction to hidden markov models," *IEEE ASSP Magazine*, vol. 3, pp. 4–16, Jan. 1986.

[77] R. L. Streit and R. F. Barrett, "Frequency line tracking using hidden markov models," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 38, pp. 709–729, Apr. 1990.

[78] R. F. Barrett and D. A. Holdsworth, "Frequency tracking using hidden markov models with amplitude and phase information," *IEEE Transactions on Signal Processing*, vol. 41, no. 10, pp. 2965–2976, Oct. 1993.

[79] S. Paris and C. Jauffret, "Frequency line tracking using hmm-based schemes [passive sonar]," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 39, no. 2, pp. 439–449, Apr. 2003.

[80] X. Xie and R. Evans, "Multiple target tracking and multiple frequency line tracking usinghidden markov models," *IEEE Transactions on Signal Processing*, vol. 39, no. 12, pp. 2659–2676, Dec. 1991.

[81] M. Deisher and A. Spanias, “Speech enhancement using state-based estimation and sinusoidal modeling,” *Journal of the Acoustical Society of America*, vol. 102, no. 2, pp. 1141–1148, Aug. 1997.

[82] N. Metropolis, A. Rosenbluth, and M. Rosenbluth, “Equation of state calculations by fast computing machines,” *Journal of Chemical Physics*, vol. 21, p. 1087, Jun. 1953.

[83] A. Doucet and X. Wang, “Monte carlo methods for signal process [a review in the statistical signal processing context],” *IEEE Signal Processing Magazine*, vol. 22, no. 6, pp. 152–170, Nov. 2005.

[84] D. S. Sivia, *Data Analysis: A Bayesian Tutorial.* Oxford University Press, USA, 1996.

[85] F. Gustafsson, F. Gunnarsson, N. Bergman, U. Forssell, J. Jansson, R. Karlsson, and P.-J. Nordlund, “Particle filters for positioning, navigation, and tracking,” *IEEE Trans Signal Process*, vol. 50, no. 2, pp. 425–437, Feb 2002.

[86] C. Dubois and M. Davy, “Joint detection and tracking of time-varying harmonic components: A flexible bayesian approach,” *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 15, no. 4, pp. 1283–1295, May 2007.

[87] M. Klaas, M. Briers, N. de Freitas, A. Doucet, S. Maskell, and D. Lang, “Fast particle smoothing: If i had a million particles,” in *The 23rd international conference on machine learning*, ser. ACM International Conference Proceeding Series, vol. 148. ACM: New York, NY, 2006, pp. 481–488.

[88] Z. Khan, T. Batch, and F. Dellaert, "A rao-blackwellized particle filter for eigentracking," in *Proc. IEEE Comput. Soc. Conf. Comput. Vision Pattern Recognit.*, vol. 2, IEEE Computer Society. Washington, DC, United States: IEEE, Jun 27 - Jul 2 2004, pp. II980–II986.

[89] N. de Freitas, "Rao-blackwellised particle filtering for fault diagnosis," in *Aerospace Conference Proceedings, 2002. IEEE*, vol. 4, 2002, pp. 4–1767–4–1772.

[90] M. Isard and A. Blake, "Condensationconditional density propagation for visual tracking," *International Journal of Computer Vision*, vol. 29, no. 1, pp. 5–28, Aug. 1998.

[91] J. Vermaak, C. Andrieu, A. Doucet, and S. Godsill, "Particle methods for bayesian modeling and enhancement of speechsignals," *IEEE Transactions on Speech and Audio Processing*, vol. 10, no. 3, pp. 173–185, Mar. 2002.

[92] A. Doucet, S. Godsill, and C. Andrieu, "On sequential Monte Carlo sampling methods for Bayesian filtering," *Statistics and Computing*, vol. 10, no. 3, pp. 197–208, Jul 2000.

[93] E. Lehmann, *Theory of Point Estimation.* New York, U.S.A.: Wiley, 1983.

[94] G. Casella and C. P. Robert, "Rao-blackwellisation of sampling schemes," *Biometrika*, vol. 83, no. 1, pp. 81–94, Mar. 1996.

[95] F. Gustafsson, T. B. Schon, R. Karlsson, and P.-J. Nordlund, "State-of-the-art for the marginalized particle filter," in *Nonlinear Statistical Signal Processing Workshop, 2006 IEEE*, 13-15 Sep. 2006, pp. 172–174.

[96] N. Gordon, D. Salmond, and A. Smith, "Novel approach to nonlinear/non-gaussian bayesian state estimation," in *IEE Proceedings-F*, vol. 140, no. 2, 1993, pp. 107–113.

[97] O. Cappé, S. Godsill, and E. Moulines, "An overview of existing methods and recent advances in sequential Monte Carlo," in *Proceedings of the IEEE*, vol. 95, no. 5. IEEE-INST ELECTRICAL ELECTRONICS ENGINEERS INC, May 2007, pp. 899–924.

[98] A. Viterbi, "Error bounds for convolutional codes and an asymptotically optimum decoding algorithm," *IEE Transactions on Information Theory*, vol. 13, no. 2, pp. 260–269, Apr. 1967.

[99] S. Godsill, A. Doucet, and M. West, "Maximum A Posteriori sequence estimation using Monte Carlo particle filters," *Ann. Inst. Statist. Math.*, vol. 53, no. 1, pp. 82–96, 2001.

[100] F. Hutter and R. Dearden, "The gaussian particle filter for diagnosis of non-linear systems," in *Proceedings of the* 14^{th} *International Conference on Principles of Diagnosis (DX'03)*, 2003, pp. 65–70.

[101] K. A. Myers and B. D. Tapley, "Adaptive sequential estimation with unknown noise statistics," *IEEE Transactions on Automatic Control*, vol. 21, no. 4, pp. 520–523, Aug. 1976.

[102] J. Hurtado, J. Lachaux, D. J. Beckley, and C. M. Gray, "Inter- and Intralimb Oscillator Coupling in Parkinsonian Tremor," *Movement Disorders*, vol. 15, no. 4, pp. 683–691, 2000.

Hurtado, J.M. and Lachaux, J. and Beckley, D.J. and Gray, C.M.

[103] J. Hurtado, L. Rubchinsky, K. Sigvardt, and V. Wheelock, "Temporal evolution of oscillations and synchrony in GPi/Muscle pairs in Parkinson's disease," *Journal of Neurophysiology*, vol. 93, pp. 1569–1584, 2005.

[104] B. Hellwig, B. Schelter, B. Guschlbauer, J. Timmer, and C. Lücking, "Dynamic synchronisation of central oscillators in essential tremor," *Clinical Neurophysiology*, vol. 114, pp. 1462–1467, 2003.

[105] J. Hurtado, L. Rubchinsky, and K. Sigvardt, "Statistical method for detection of phase-locking episodes in neural oscillations," *Journal of Neurophysiology*, vol. 91, pp. 1883–1898, 2004.

[106] F. Michard, "Changes in arterial pressure during mechanical ventilation," *Anesthesiology*, vol. 103, no. 2, pp. 419–428, Aug. 2005.

[107] M. Aboy, J. McNames, T. Thong, D. Tsunami, M. S. Ellenby, and B. Goldstein, "An automatic beat detection algorithm for pressure signals." *IEEE Trans Biomed Eng*, vol. 52, no. 10, pp. 1662–70, Oct 2005.

[108] R. Berger, J. Saul, P. Albrecht, S. Stein, and R. Cohen, "Respiratory effects on arterial pressure: a novel signal analysis approach," in *Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society, 1988.*, vol. 2, Nov. 1998, pp. 533–534.

[109] A. Doucet and C. Andrieu, "Iterative algorithms for state estimation of jump markov linear systems," *IEEE Transactions on Signal Processing*, vol. 49, no. 6, pp. 1216–1227, Jun. 2001.

[110] A. T. Cemgil and B. Kappen, “Monte Carlo methods for tempo tracking and rhythm quantization,” *Journal of Artificial Intelligence Research (JAIR)*, vol. 18, pp. 45–81, 2003.

[111] D. Yee, J. Reilly, and T. Kirubarajan, “A blind sequential monte carlo detector for ofdm systems in the presence of phase noise, multipath fading, and channel order uncertainty,” *IEEE Transactions on Signal Processing*, vol. 55, no. 9, pp. 4581–4598, Sep 2007.

[112] G. Kitagawa, “Monte-carlo filter and smoother for non-gaussian nonlinear state sapace models,” *J. Comput. and Graph. Statist.*, vol. 1, pp. 1–25, 1996.

[113] R. J. Elble, “Diagnostic criteria for essential tremor and differential diagnosis.” *Neurology*, vol. 54, no. 11 Suppl 4, pp. S2–S6, 2000.

[114] J. Hurtado, C. Gray, L. Tamas, and K. Sigvardt, “Dynamics of tremor-related oscillations in the human globus pallidus: A single case study,” in *Proceedings, National Academy of Science*, vol. 96, Feb. 1999, pp. 1674–16–79.

[115] J. McNames, “Optimal rate filters for biomedical point processes,” in *Annual International Conference of the IEEE Engineering in Medicine and Biology Proceedings*, 2005, pp. 145–148, this is a test.

[116] S. Kim and J. McNames, “Detecting and tracking tremor in spike trains using the rectangular model based extended kalman smoother,” Accepted to be published in Journal of Neuroscience Methods.

[117] S. Kim, L. Holmstrom, and J. McNames, “Multiharmonic Tracking Using Marginalized Particle Filters,” in *30th Annual International Conference of*

the IEEE Engineering in Medicine and Biology Society - Proceedings, Aug. 2008, pp. 29–33.

[118] S. Kim and J. McNames, "Tracking intermittent tremor frequency with a particle filter," in *IEEE/SP 14th Workshop on Statistical Signal Processing*, 26-29 Aug. 2007, pp. 171–175.

[119] S. Kim and J. McNames, "Tracking tremor frequency in spike trains using the extended Kalman filter," in *Annual International Conference of the Engineering in Medicine and Biology Society*, 2005, pp. 7576–7579.

[120] D. H. Johnson, "Point process models of single-neuron discharges," *Journal of Computational Neuroscience*, vol. 3, pp. 275–299, 1996.

[121] S. Kim, J. McNames, and K. Burchiel, "Detecting tremors in microelectrode recordings without using a spike detector." *Annual International Conference of the Engineering in Medicine and Biology Society*, vol. 1, pp. 357–360, 2004. [Online]. Available: http://dx.doi.org/10.1109/IEMBS.2004.1403166

[122] M. Pagani, F. Lombardi, S. Guzzetti, O. Rimoldi, R. Furlan, P. Pizzinelli, G. Sandrone, G. Malfatto, S. Dell'Orto, E. Piccaluga, M. Turiel, G. Baselli, S. Cerutti, and A. Malliani, "Power Spectral Analysis of Heart Rate and Arterial Pressure Variability as a Marker of Sympatho-Vagal Interactoin in Man and Conscious Dog," *Journal of the American Heart Association*, vol. 59, pp. 178–193, 1986.

[123] S. Akselrod, D. Gordon, F. Ubel, D. Shannon, A. Berger, and R. Cohen, "Power spectrum analysis of heart rate fluctuatino: a quantitative probe of

beat-to-beat cardiovascular control," *American Association for the Advancement of Science*, vol. 213, no. 4504, pp. 220–222, Jul. 1981.

[124] B. Pomeranz, R. Macaulay, M. Caudill, I. Kutz, D. Adam, D. Gordon, K. Kilborn, A. Barger, D. Shannon, R. Cohen, and H. Benson, "Assessment of autonomic function in humans by heart rate spectral analysis," *American Journal of Physiology*, vol. 248, pp. H151–H153, 1985.

[125] J. Penàz, "Mayer waves: History and methodology," *Automedica*, vol. 2, pp. 135–141, 1978.

[126] J. McNames and M. Aboy, "Statistical modeling of cardiovascular signals and parameter estimation based on the extended kalman filter," *IEEE Transactions on Biomedical Engineering*, vol. 55, no. 1, pp. 119–129, Jan. 2008.

[127] M. Ebrahim, J. M. Feldman, and I. Bar-Kana, "A robust sensor fusion method for heart rate estimation," *Journal of Clinical Monitoring and Computing*, vol. 13, no. 6, pp. 385–393, Nov. 1997.

[128] V. Afonso, W. Tompkins, T. Nquyen, and L. Shen, "ECG beat detection using filter banks," *IEEE Transactions on Biomedical Engineering*, vol. 46, no. 2, pp. 192–202, Feb. 1999.

[129] J. Pan and W. J. Tompkins, "A Real-Time QRS Detection Algorithm," *IEEE Transactions on Biomedical Engineering*, vol. BME-32, no. 3, pp. 230–236, Mar. 1985.

[130] G. B. Moody and R. G. Mark, "A database to support development and evaluation of intelligent intensive care monitoring," *Computers in Cardiology*, vol. 23, pp. 657–660, 1996.

[131] A. L. Goldberger, L. A. N. Amaral, L. Glass, J. M. Hausdorff, P. C. Ivanov, R. G. Mark, J. E. Mietus, G. B. Moody, C.-K. Peng, and H. E. Stanley, "PhysioBank, PhysioToolkit, and PhysioNet: Components of a new research resource for complex physiologic signals," *Circulation*, vol. 101, no. 23, pp. e215–e220, 2000 (June 13), circulation Electronic Pages: http://circ.ahajournals.org/cgi/content/full/101/23/e215.

[132] C. K. Hofer, L. Furrer, S. Matter-Ensner, M. Maloigne, R. Klaghofer, M. Genoni, and A. Zollinger, "Volumetric preload measurement by thermodilution: a comparison with transoesophageal echocardiography." *Br J Anaesth*, vol. 94, no. 6, pp. 748–55, Jun 2005. [Online]. Available: http://dx.doi.org/10.1093/bja/aei123

[133] F. Michard, S. Boussat, D. Chemla, N. Anguel, A. Mercat, Y. Lecarpentier, C. Richard, M. R. Pinsky, and J. L. Teboul, "Relation between respiratory changes in arterial pulse pressure and fluid responsiveness in septic patients with acute circulatory failure." *American Journal of Respiratory and Critical Care Medicine*, vol. 162, no. 1, pp. 134–8, Jul 2000.

[134] F. Michard, L. Ruscio, and J. L. Teboul, "Clinical prediction of fluid responsiveness in acute circulatory failure related to sepsis." *Intensive Care Medicine*, vol. 27, no. 7, p. 1238, Jul 2001.

[135] A. Kramer, D. Zygun, H. Hawes, P. Easton, and A. Ferland, "Pulse pressure variation predicts fluid responsiveness following coronary artery bypass surgery." *Chest*, vol. 126, no. 5, pp. 1563–8, Nov 2004. [Online]. Available: http://dx.doi.org/10.1378/chest.126.5.1563

[136] F. Michard and J.-L. Teboul, "Predicting fluid responsiveness in ICU patients: a critical analysis of the evidence." *Chest*, vol. 121, no. 6, pp. 2000–8, Jun 2002.

[137] M. Aboy, J. McNames, T. Thong, C. Phillips, M. Ellenby, and B. Goldstein, "A novel algorithm to estimate the pulse pressure variation index," *IEEE Transactions on Biomedical Engineering*, vol. 51, pp. 2198–2203, 2004.

[138] M. Cannesson, J. Slieker, O. Desebbe, C. Bauer, P. Chiari, R. Hnaine, and J.-J. Lehot, "The ability of a novel algorithm for automatic estimation of the respiratory variations in arterial pulse pressure to monitor fluid responsiveness in the operating room." *Anesth Analg*, vol. 106, no. 4, pp. 1195–200, table of contents, Apr 2008. [Online]. Available: http://dx.doi.org/10.1213/01.ane.0000297291.01615.5c

[139] M. Aboy, C. Crespo, and D. Austin, "An enhanced automatic algorithm for estimation of respiratory variations in arterial pulse presssure during periods of abrupt hemodynamic changes." *IEEE Trans Biomed Eng*, p. In Print, 2009.

[140] S. Kim, M. Aboy, and J. McNames, "New signal model for arterial blood pressure signals," in *Biosignal Interpretation Conference, 2009*, New Haven, CT, Jun. 2009, p. In press.

[141] C. Wiesenack, C. Fiegl, A. Keyser, C. Prasser, and C. Keyl, "Assessment of fluid responsiveness in mechanically ventilated cardiac surgical patients." *Eur J Anaesthesiol*, vol. 22, no. 9, pp. 658–65, Sep 2005.

[142] M. K. Dahl, S. T. Vistisen, J. Koefoed-Nielsen, and A. Larsson, "Using an expiratory resistor, arterial pulse pressure variations predict fluid responsiveness during spontaneous breathing: an experimental porcine study," *Critical Care*, vol. 13, no. 2, Mar. 2009.

[143] A. Coudray, J.-A. Romand, M. Treggiari, and K. Bendjelid, "Fluid responsiveness in spontaneously breathing patients: A review of indexes used in intensive care," *Critical Care Medicine*, vol. 33, no. 12, pp. 2757–2762, 2005.

[144] D. D. Backer and M. R. Pinsky, "Can one predict fluid responsiveness in spontaneously breathing patients?" *Intensive Care Medicine*, vol. 33, no. 7, pp. 1111–11 113, Jul. 2007.

[145] B. Lamia, A. Ochagavia, X. Monnet, D. Chemla, C. Richard, and J.-L. Teboul, "Echocardiographic prediction of volume responsiveness in critically ill patients with spontaneously breathing activity," *Intensive Care Medicine*, vol. 33, no. 7, pp. 1125–1132, Jul. 2007.

[146] S. Heenen, D. D. Backer, and J.-L. Vincent, "How can the response to volume expansion in patients with spontaneous respiratory movements be predicted?" *Critical Care*, vol. 10, no. 4, Jul. 2006.

[147] S. Kim, L. Holmstrom, and J. McNames, "The maximum a posteriori adaptive marginalized particle filter," submitted to Advances on Signal Processing.

Appendix A

Notational Conventions

I have adopted the notation used in [97] with minor modifications. Boldface is used to denote random processes, normal face for deterministic parameters and functions, upper case letters for matrices, lower case letters for vectors and scalars, superscripts in parenthesis for particle indices, upper-case superscripts for nonlinear/linear indication, and subscripts for time indices. The list below summarizes the notational conventions.

- $\boldsymbol{r}$: random processes
- d: deterministic parameters and functions
- M: matrices
- v: vectors and scalar
- $\cdot^{(i)}$: particle indices
- $\cdot^{N}$: nonlinear/linear indication
- $\cdot_{n}$: time indices
- $\tilde{\boldsymbol{w}}$: unnormalized/normalized particle weight

- $\tilde{\boldsymbol{x}}$: state before/after resampling

A.1 Examples

- $\boldsymbol{x}_n^{\mathrm{N},(i)}$: the nonlinear portion of the state vector for the i^{th} state trajectory (i.e., particle) at discrete time n
- $\tilde{\boldsymbol{w}}^{(i)}$: the unnormalized particle weight of the i^{th} particle
- $\boldsymbol{w}^{(i)}$: the normalized particle weight of the i^{th} particle
- $\tilde{\boldsymbol{x}}_n^{(i)}$: the i^{th} particle before resampling
- $\boldsymbol{x}_n^{(i)}$: the i^{th} particle after resampling
- $f_n(\boldsymbol{x})$: a time-varying function f that takes a random vector $\boldsymbol{x}$ as its input
- QM^{-1}: multiplication of a matric Q and the inverse of a matric M