8-2015

A Switched System Identification Approach to Spindle Modeling

Mohsin Farooq

A Switched System Identification Approach to Spindle Modeling

by

Mohsin Farooq

A Thesis Submitted in Partial Fulfillment of the Requirements for the Degree of Master of Science in Computer Engineering

Supervised by

Dr. Juan Cockburn
Department of Computer Engineering
Kate Gleason College of Engineering
Rochester Institute of Technology
Rochester, New York
August 2015

Approved by:

Dr. Juan Cockburn, Associate Professor
Thesis Advisor, Department of Computer Engineering

Dr. Raymond Ptucha, Assistant Professor
Committee Member, Department of Computer Engineering

Dr. Andres Kwasinski, Associate Professor
Committee Member, Department of Computer Engineering

Dedication

I would like to dedicate this to the loved ones in my life. My parents, siblings and friends.

Acknowledgments

A huge thank you to those whose academic guidance and mentorship helped me develop my passions into a thesis that excited and motivated me. Thank you to my advisor, Dr. Juan Cockburn, who worked tirelessly with me. I am very honored to have gotten the opportunity to work with you and hope to do so in the future as well. Thank you very much to my committee members, Dr. Raymond Ptucha and Dr. Andres Kwasinski, whom I also bothered with obscure questions related to problems in their respective fields. I am very grateful for the opportunity I had at RIT. Thank you very much to my friends Kassaundra and Jeffrey who spent hours helping me revise my thesis.

Contents

List of Figures

Abstract

A Switched System Identification Approach to Spindle Modeling

Mohsin Farooq

Due to new advances in convex optimization, in particular, semidefinite programming, previously infeasible problems are now in the realm of possibility. Mainly, there have been new breakthroughs in the modeling of signals as the output of switched dynamical systems where the switching indicates underlying events of interest. This method is known as hybrid system identification. These problems can be formulated as polynomial optimization problems by which, through algebraic reformulations, convex optimization approaches now exist.

In this work, we explore the application of these new approaches, which lay at the intersection of systems and control with machine learning, for the detection of events in electroencephalogram (EEG) signals.

Our particular focus on EEG signals is twofold. First, these signals are routinely used to monitor the quality of sleep, which is critical to both physical and mental health. Second, the onset of the internet-of-things has driven industry to develop affordable, in home, EEG sleep monitors.

Most of these devices will take advantage of cloud services where vast amounts of sleep data will be processed.

There have been various attempts to develop automatic staging systems using mostly machine learning approaches such as Support Vector Machines and Neural Networks. However, there is very limited research that explores the use of switched dynamical systems to model sleep waveforms.

This thesis work is the first step towards this direction. It focuses on modeling spindles, found in stage two of sleep, as switched Autoregressive (AR) models where the switching events are used to determine if a spindle occurred. Various aspects of the problem are considered, such as those related to error introduced by noise and the effect of model order.

The results presented in this work reveal potential new approaches to unsupervised classification of spindles and event based feature detection in complex signals.

Chapter 1

Introduction

With the increase in computational power as well as advances in techniques related to machine learning, controls, and optimization, many new methods are being developed which lay at the intersection of the three disciplines. Through these methods researchers hope to achieve better learning algorithms for various applications. One area being researched extensively for applications related to these disciplines is that of hybrid system identification (HID) via polynomial optimization. Hybrid system identification is a problem in which multiple Linear Time Invariant (LTI) models are used to fit dynamical observed data.

1.1 Contribution of the Thesis

This thesis uses methods from hybrid system identification via polynomial optimization to model "spindles" found in EEG signal data. Hybrid

system identification is the idea of finding separate linear dynamical systems (known as autoregressive models) that can be used to describe some underlying time dependent data. In recent years there have been new advances in the theory of polynomial optimization. This is due to the discovery of convex relaxations which allow better bounds for problems that are generally NP-hard to solve. The relaxations are based on the idea of moments which arise when exploring the dual problem to the Sum of Squares (SOS) problem. These relaxations have made it possible for new techniques to be developed in many areas such as system identification, optimal control, and many other domains.

1.2 Thesis Layout

The layout of the thesis is as follows: In chapter 2 we will discuss background information related to polynomial optimization. Firstly, we will begin by introducing the notation necessary to discuss problems that are polynomial in nature. We will then discuss general polynomial optimization and convex relaxations which allow us to solve them efficiently and globally. These relaxations are based on the idea of moments as well as tools to solve them such as GloptiPoly and SparsePOP. In chapter 3, we will begin by introducing the underlying structure and notation necessary to discuss autoregressive models. We will then move onto the idea of set membership identification and the algebraic reformulations

associated with solving the identification problem through moments. In particular we will discuss the hybrid decoupling constraint [1] which is a key component in the formulation of hybrid system identification. Throughout both chapters we will give examples where necessary so that this thesis can standalone. In chapter 4 we will briefly discuss the problem formulation and the data set used so that we may apply the method of hybrid system identification to spindle detection. This is followed by the results and analysis chapter in which descriptions of the results are presented. Lastly, we conclude with a brief description of the work completed and propose future work that can be done.

Chapter 2

Polynomial Optimization

In this chapter we will discuss the area of polynomial optimization and some methods of solving them via Linear Matrix Inequality (LMI) relaxations based on moments. An LMI is a linear inequality constraint defined on a matrix. Relaxations are methods that allow us to place lower or upper bounds on problems, solving them within a certificate of the optimal value. The field of polynomial optimization has been recently getting more attention and its methods have been used in various fields such as Operations Research, Computer Vision and Control System to solve challenging real world problems.

2.1 Basic Definitions

In this section we will introduce some definition and notation used to describe polynomials.

Given a vector $\boldsymbol{x} \in \mathbb{R}^n$ and an integer vector $\boldsymbol{\alpha} \in \mathbb{N}^n$, a monomial (in n

variables $x_1, \ldots, x_n$) is defined as

$$\boldsymbol{x}^{\boldsymbol{\alpha}} := \prod_{k=1}^{n} {x_k}^{\alpha_k}$$

The degree of a monomial with exponent $\boldsymbol{\alpha} \in \mathbb{N}^n$ is

$$\deg(\boldsymbol{x}^{\boldsymbol{\alpha}}) = |\boldsymbol{\alpha}| := \sum_{k=1}^{n} \alpha_k$$

A polynomial $\boldsymbol{p} \in \mathbb{R}[\boldsymbol{x}]$ of degree $d \in \mathbb{N}$ is a linear combination of monomials:

$$p(\boldsymbol{x}) := \sum_{|\alpha| \leq d} p_{\boldsymbol{\alpha}} \boldsymbol{x}^{\boldsymbol{\alpha}}$$

and $\boldsymbol{p} := (p_{\boldsymbol{\alpha}}), |\boldsymbol{\alpha}| \leq d$ is the vector of its coefficients in a given monomial basis.

2.2 Polynomial Optimization

We will begin by defining the problem of polynomial optimization.

> Given a multivariate polynomial $p(x_1, x_2, \ldots, x_n)$, find a point $\boldsymbol{x}^* \in \mathbb{R}^n$ where $p^* = p(\boldsymbol{x}^*)$ attains a minimum.

Using mathematical notation the problem can be written as

$$\mathbb{P} \mapsto p* \ := \ \min_{\boldsymbol{x} \in \mathbb{R}^n} p(\boldsymbol{x}) \tag{2.1}$$

where $p(\boldsymbol{x}) : \mathbb{R}^n \to \mathbb{R}$ and $\mathbb{P}$ denotes the set of n-variate polynomials.

In addition, if $\boldsymbol{x}$ is restricted to take values in a compact set $K \subset \mathbb{R}^d$, the problem becomes

$$\mathbb{P}_K \mapsto p_K^* := \min_{x \in K} p(\boldsymbol{x}) \tag{2.2}$$

The constraint sets K to be considered here are sets generated by the points satisfying a set of r polynomial inequalities

$$K = \{\boldsymbol{x} \in \mathbb{R}^n : g_i(\boldsymbol{x}) \geq 0, g_i(\boldsymbol{x}) \in \mathbb{R}[\boldsymbol{x}], i = 1, \ldots, r\} \tag{2.3}$$

Such compact sets are called *semi-algebraic*.

Polynomial inequalities as above generate compact sets that are not necessarily convex. Therefore, it is difficult to solve these optimization problems, let alone find a global solution. Due to their non-convex nature both in the cost and the constraints, these problems are known to be NP hard, that is, there is no polynomial–time algorithm which is able to solve them.

A strategy that proved to be successful to solve problem (2.1) is to approximate it with a hierarchy of convex semidefinite relaxations. Such relaxations can be constructed using sums of squares of polynomials and the dual theory of moments [2]. Lasserre showed that problems (2.1) and (2.2) can be transformed, respectively, into the following equivalent problems:

$$\mathcal{P} \mapsto p^* := \min_{\mu \in P(\mathbb{R}^n)} \int p(\boldsymbol{x}) \mu(d\boldsymbol{x}) \tag{2.4}$$

$$\mathcal{P}_K \mapsto p^* := \min_{\mu \in \mathcal{P}(K)} \int p(\boldsymbol{x})\mu(d\boldsymbol{x}) \tag{2.5}$$

where $\mathcal{P}(K)$ is the space of finite Borel probability measures on K (a probability measure μ on X is a positive measure such that $\mu(X) = 1$.) This result is summarized in the following theorem:

Theorem 2.2.1 (Lasserre 2001)

1. $\mathbb{P} = \mathcal{P}$.
2. *if* $\boldsymbol{x}^*$ *is the global minimizer of* $\mathbb{P}$ *then the probability measure* $\mu := \delta_{\boldsymbol{x}^*}$ *is admissible for* $\mathcal{P}$.
3. *For every optimal solution* μ^* *of* $\mathcal{P}_K$, $p(\boldsymbol{x}) = \min\ \mathbb{P}_K, \mu^*$*- almost everywhere.*

(For a Proof, see Proposition 2.1 in [2]).

Problems (2.4) and (2.5) are now convex but infinite dimensional. However an approximation can be made that relaxes this problem to some given order n.

This conversion enables us to develop a convergent sequence of Linear Matrix inequality (LMI) relaxations to problem (2.2) in which we optimize over the moments $m_\alpha \doteq \mathrm{E}_\mu[x^\alpha]$

2.3 Polynomial Optimization via Moments

In this section, we will discuss the idea of moments and their importance to solving polynomial optimization problems. This section was taken primarily from Henrion *et al.* in [3]. We will answer the following questions:

- What are Moments?
- What are the convex relaxations?
- What is the equivalent problem?
- How do we formulate it?
- What do the different constraints and objective function correspond to?
- What are the general tools used to solve this problem?

A moment is some quantitative measure which can be used to describe the structure of a distribution of values. This can be found in probability theory where the first moment μ_1 is known as the mean and the second moment μ_2 is known as the variance. The moment problem is to assert when a given sequence of numbers represent the successive moments $\int \boldsymbol{x}^{\alpha} d\mu(\boldsymbol{x}), \alpha = 0, 1, \ldots,$ of a nonnegative measure μ. Here positivity suggests convexity, which can be used to define a scalar product for

polynomials. This is important as it allows for orthogonal polynomials, decompositions and expansions [4].

If the sequence of scalars $\{m_k\}_{k=1}^n$ are moments, and the idea is to find a probability measure supported in $\mathbb{R}$ s.t. $\{m_k\}$ represents the first n moments. They can be represented as the following:

$$m_\alpha = \mathrm{E}_\mu(\boldsymbol{x}^{\boldsymbol{\alpha}}) = \int_K \boldsymbol{x}^{\boldsymbol{\alpha}} \mu(d\boldsymbol{x}) \tag{2.6}$$

We can then use these moments to formulate the problem of polynomial optimization as follows:

$$\begin{aligned} p_N^* = \min_m \ & \sum_\alpha p_\alpha m_\alpha \\ & s.t.\ M_N(m) \succeq 0, \\ & L_N(g_i m) \succeq 0, k = 1, \ldots, d \end{aligned} \tag{2.7}$$

Here $M_N(m)$ is known as the *moment* matrix and $L_N(g_i m)$ is known as the *localizing* matrix. The moment matrix arises through the multiplication of two polynomials given in a basis b which is defined by:

$$\begin{aligned} b(\boldsymbol{x}) = [1,\ & x_1,\ x_2,\ \cdots,\ x_n, \cdots, x_1^2,\ x_1x_2, \cdots,\ x_1x_n,\ x_2x_3,\ \cdots,\ x_n^2, \\ & x_1^3, x_1^2x_2, x_1^2x_3,\ \cdots, x_n^3, \cdots, x_1^d,\ \cdots, x_n^d]^T \end{aligned} \tag{2.8}$$

We can then represent any polynomial of degree d by a vector.

Example: Given a polynomial $p(\boldsymbol{x}) = 2 + x_1^2 + 2x_1x_2 + 3x_2^2$ of degree 2 and the standard basis

$$b(\boldsymbol{x}) \;=\; [1, x_1, x_2, x_1^2, x_1x_2, x_2^2]^T$$

note that,

$$2 = (p_{\boldsymbol{\alpha}})\, x^{\boldsymbol{\alpha}} \qquad (\boldsymbol{\alpha} = 00) \tag{2.9}$$

$$x_1^2 = (p_{\boldsymbol{\alpha}})\, x^{\boldsymbol{\alpha}} \qquad (\boldsymbol{\alpha} = 20) \tag{2.10}$$

$$2x_1x_2 = (p_{\boldsymbol{\alpha}})\, x^{\boldsymbol{\alpha}} \qquad (\boldsymbol{\alpha} = 11) \tag{2.11}$$

$$3x_2^2 = (p_{\boldsymbol{\alpha}})\, x^{\boldsymbol{\alpha}} \qquad (\boldsymbol{\alpha} = 02) \tag{2.12}$$

where (p_α) denotes the coefficient corresponding to the monomial x_α. This polynomial has a vector representation $\boldsymbol{p} \in \mathbb{R}^6$ given by

$$\boldsymbol{p} = \begin{bmatrix} 2 & 0 & 0 & 1 & 2 & 3 \end{bmatrix}^T$$

By the Riesz representation theorem we can use the standard inner product in $\mathbb{R}^n$, $\langle \boldsymbol{v}, \boldsymbol{w} \rangle = \boldsymbol{v}^T \boldsymbol{w}$, to represent the linear functionals $L_y : \mathcal{P} \mapsto \mathbb{R}$

$$p \mapsto L_y(p) = \langle \boldsymbol{p}, y \rangle = \sum_{\alpha \in \mathbb{N}^n} p_{\boldsymbol{\alpha}} y_{\boldsymbol{\alpha}} \tag{2.13}$$

Here we have done a change of variables from x to y where we go to a higher dimension which is no longer nonlinear as we represent all monomials as individual variables. This allows us to define a bilinear

mapping $\langle \cdot, \cdot \rangle_y : \mathcal{P} \times \mathcal{P} \to \mathbb{R}$ between two polynomials being multiplied

$$\langle p, q \rangle_y = L_y(p\,q) = \langle \boldsymbol{p}, M(\boldsymbol{y})\boldsymbol{q} \rangle \tag{2.14}$$

$M(\boldsymbol{y})$ represents a matrix whose rows and columns are indexed by the basis found in (2.8). Taking the index vector $\boldsymbol{\alpha}$ of the first polynomial and index vector $\boldsymbol{\beta}$ from the second polynomial, the entry $(\boldsymbol{\alpha}, \boldsymbol{\beta})$ of $M(\boldsymbol{y})$ can be written as:

$$[M(y)]_{\boldsymbol{\alpha\beta}} = L_y([b(\boldsymbol{x})b(\boldsymbol{x})^T]_{\boldsymbol{\alpha\beta}}) = y_{\boldsymbol{\alpha}+\boldsymbol{\beta}} \tag{2.15}$$

Since the basis is infinite dimensional (for polynomials of arbitrary degree), the matrix $M(y)$ is also infinite but for a polynomial of a given degree it will have finite dimensions. The dimension of the moment matrix $M(y)$ depends on the relaxation order of the problem, for a relaxation of order d the moment matrix $M_d(y)$ will have coefficients of polynomials order up to $2d$. For example, the moment matrices $M_1(y)$ and $M_2(y)$ are

$$M_1(y) = \begin{bmatrix} 1 & y_{10} & y_{01} \\ y_{10} & y_{20} & y_{11} \\ y_{01} & y_{11} & y_{02} \end{bmatrix} \tag{2.16}$$

$$M_2(y) = \begin{bmatrix} 1 & y_{10} & y_{01} & y_{20} & y_{11} & y_{02} \\ y_{10} & y_{20} & y_{11} & y_{30} & y_{21} & y_{12} \\ y_{01} & y_{11} & y_{02} & y_{21} & y_{12} & y_{03} \\ y_{20} & y_{30} & y_{21} & y_{40} & y_{31} & y_{22} \\ y_{11} & y_{21} & y_{12} & y_{31} & y_{22} & y_{13} \\ y_{02} & y_{12} & y_{03} & y_{22} & y_{13} & y_{04} \end{bmatrix} \tag{2.17}$$

The localization matrices have similar representation except that they represent constraints in the original problem. Given some polynomial inequality constraint $g_i \geq 0$ this can be converted to a localization matrix by defining some bilinear mapping $\langle \cdot, \cdot \rangle_{gy} : \mathcal{P} \times \mathcal{P} \mapsto \mathbb{R}$ as follows:

$$(p, q) \to \langle p, q \rangle_{gy} = L_y(gpq) = \langle \boldsymbol{p}, M(gy)\boldsymbol{q} \rangle \tag{2.18}$$

Similarly, the number of entries in the localization matrix also depend on the relaxation order defined. The original problem can be modeled exactly, as the relaxation order approaches ∞. However, in many cases we only need a relaxation order between 2-8 to produce near optimal results. There also exist methods to detect whether the global optimum is reached numerically at a given relaxation order k. This can be found at [5]. Let's look at an example where a polynomial optimization is formulated as (2.7), this example is also taken from course material taught be Henrion which can be found here [6].

Example: Take the following polynomial optimization problem:

$$\begin{aligned} \text{maximize} \quad & x_2 \\ \text{subject to} \quad & 3 - 2x_2 - x_2^1 - x_2^2 \geq 0, \\ & -x_1 - x_2 - x_1 x_2 \geq 0, \\ & 1 + x_1 x_2 \geq 0. \end{aligned}$$

This defines a non-convex region as shown in Figure 2.3. Since this

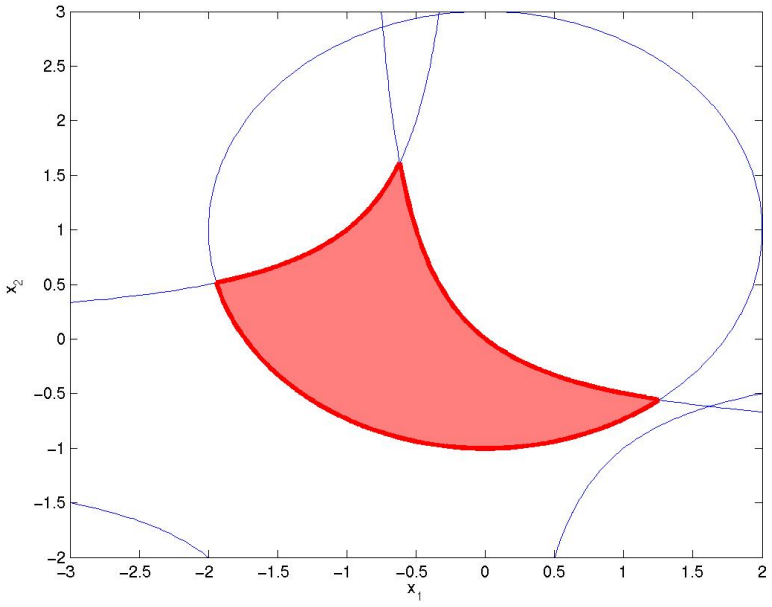

Figure 2.1: Plot of Polynomial Inequalities Described by the Constraints [3].

region is non-convex, we first try a relaxation of order 1 to solve the

problem. The problem above therefore gets converted to:

$$
\begin{aligned}
\text{maximize} \quad & y_{01} \\
\text{subject to} \quad & \begin{bmatrix} 1 & y_{10} & y_{01} \\ y_{10} & y_{20} & y_{11} \\ y_{01} & y_{11} & y_{02} \end{bmatrix} \succeq 0, \\
& 3 - 2y_{01} - y_{20} - y_{02} \geq 0, \\
& -y_{10} - y_{01} - y_{11} \geq 0, \\
& 1 + y_{11} \geq 0.
\end{aligned}
$$

This creates a convex bound described by the projection of y_{10}, and y_{01} as shown in Figure 2.3. The solution this relaxed problem pro-

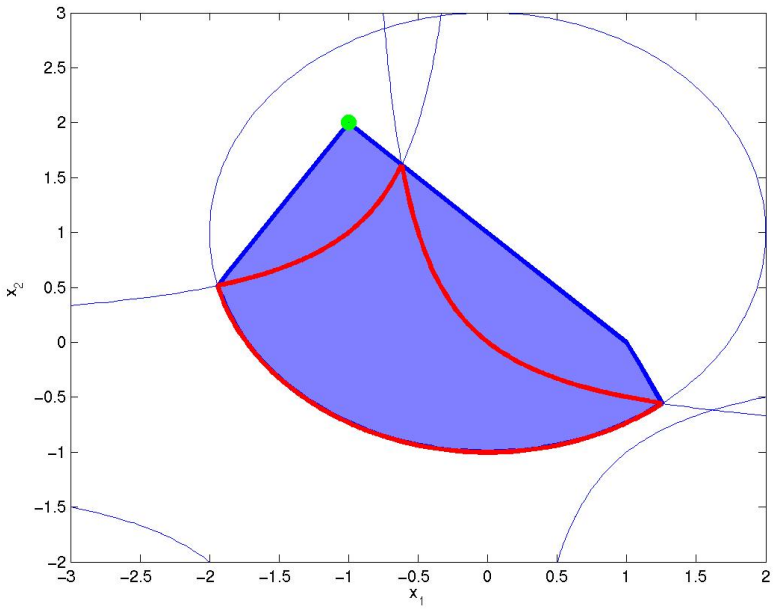

Figure 2.2: Plot of Polynomial Inequalities with Projected LMI Feasible Set of Relaxation Order 1 as Described Above [3].

vides an upper bound of 2 on $y_{01} = x_2$ on the optimal value. We can

try a second order relaxation which will also contain expressions of degree up to 4. The following is the equivalent formulation with relaxation of order 2:

$$\max \quad y_{01}$$

subject to

$$\begin{bmatrix} 1 & y_{10} & y_{01} & y_{20} & y_{11} & y_{02} \\ y_{10} & y_{20} & y_{11} & y_{30} & y_{21} & y_{12} \\ y_{01} & y_{11} & y_{02} & y_{21} & y_{12} & y_{03} \\ y_{20} & y_{30} & y_{21} & y_{40} & y_{31} & y_{22} \\ y_{11} & y_{21} & y_{12} & y_{31} & y_{22} & y_{13} \\ y_{02} & y_{12} & y_{03} & y_{22} & y_{13} & y_{04} \end{bmatrix} \succeq 0,$$

$$\begin{bmatrix} 3-2y_{01}-y_{20}-y_{02} & , 3y_{10}-2y_{11}-y_{30}-y_{12} & , 3y_{01}-2y_{02}-y_{21}-y_{03} \\ 3y_{10}-2y_{11}-y_{30}-y_{12} & , 3y_{20}-2y_{21}-y_{40}-y_{22} & , 3y_{11}-2y_{12}-y_{31}-y_{13} \\ 3y_{01}-2y_{02}-y_{21}-y_{03} & , 3y_{11}-2y_{12}-y_{31}-y_{13} & , 3y_{02}-2y_{03}-y_{22}-y_{04} \end{bmatrix} \succeq 0,$$

$$\begin{bmatrix} -y_{10}-y_{01}-y_{11} & , -y_{20}-y_{11}-y_{21} & , -y_{11}-y_{02}-y_{12} \\ -y_{20}-y_{11}-y_{21} & , -y_{30}-y_{21}-y_{31} & , -y_{21}-y_{12}-y_{22} \\ -y_{11}-y_{02}-y_{12} & , -y_{21}-y_{12}-y_{22} & , -y_{12}-y_{03}-y_{13} \end{bmatrix} \succeq 0,$$

$$\begin{bmatrix} 1+y_{11} & , y_{10}+y_{21} & , y_{01}+y_{12} \\ y_{10}+y_{21} & , y_{20}+y_{31} & , y_{11}+y_{22} \\ y_{01}+y_{12} & , y_{11}+y_{22} & , y_{02}+y_{13} \end{bmatrix} \succeq 0.$$

It can be seen that as the relaxation order increases the number of

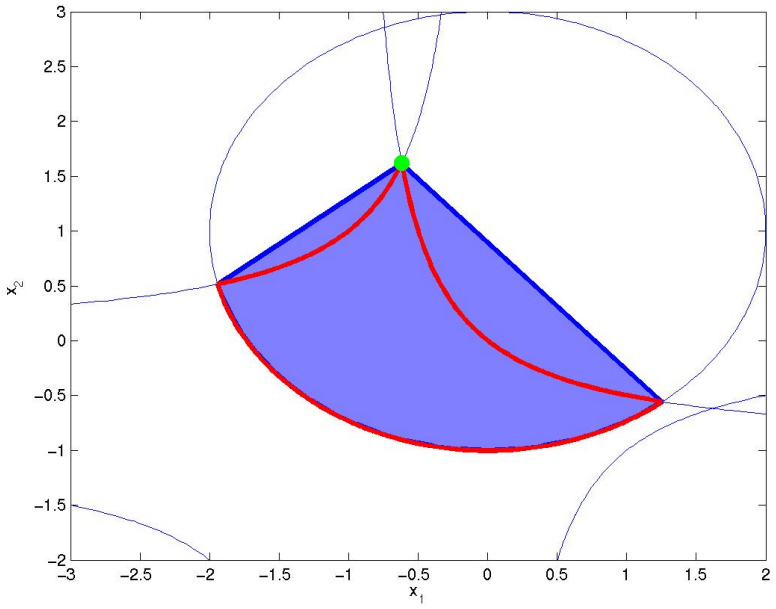

Figure 2.3: Plot of Polynomial Inequalities in Example with Projected LMI Feasible Set of Relaxation Order 2 [3].

optimization variables increase significantly. However, there is also an advantage as can be seen in Figure 2.3; the convex bound gets tighter and, in this case, the upper bound coincides with the global optimal solution at $y_{01}^* = x_2^* = 1.6180$. So with a relaxation order of 2 we were able to find the global optimal solution of a system with extrema subject to non-convex constraints.

2.4 Software Tools

As these methods become more prevalent the number of free tools available to solve these problems has numerically increased. Matlab tools

such as CVX [7] exist to solve general convex optimization problems, including SDP problems. However, when using CVX, all the moment constraints must be formulated explicitly, making it very tedious and error prone. There are other tools for directly solving polynomial optimization problems such as GloptiPoly [8] and SparsePOP [9] that transcribe the moments problem to a corresponding semidefinite program that is then submitted to an SDP solver of choice. These are the programs used in this work.

Chapter 3

Hybrid System Identification

In order to discuss hybrid system identification we must first lay a foundation for system identification. The fundamental principle of system identification is as follows: given a sequence of input/output data, find a dynamical system that can describe that data. This is significant in various fields – from controls to machine learning – for a variety of reasons. In controls it allows for the creation of mathematical models that can then be used to design controllers to modify the overall behavior of the system such as, make an unstable system stable, etc. In machine learning it allows for the representation of high quantities of data in lower order dynamical models, reducing dimension. However, many times this data contains noise, making the process difficult.

Typically the structure of the system to be identified must be properly parameterized and fixed. For instance, we can decide to model a time series as

$$y_t = \sum_{k=1}^{n} a_k y_{t-k} + u_t \tag{3.1}$$

This model assumes that the current observed values can be expressed as a linear combination of previous data values. These class of systems are called (deterministic) autoregressive (AR) models. This is significant as it allows us to both reduce the number of dimensions needed to explain a given data sequence, as well as give a model that can be compared to others for similarity.

If the parameters of the model are constant, an AR model can also be described as a transfer function that has no zeros and only has poles that are the roots of the denominator polynomial in z.

$$H(z) = \frac{Y(z)}{U(z)} = \frac{1}{1 - \sum_{k=1}^{n} a_k z^{-k}} = \frac{1}{\sum_{k=1}^{n} (1 - q_k z^{-k})} \tag{3.2}$$

In (3.2) the numbers $q_k \in \mathbb{C}$ denote the poles of the system.

The main advantage of the system identification approach [10] is that mathematical models are obtained from experimental input-output data measurements. In general, to make system identification practical, the structure of the model and of the noise is often chosen *a priory*.

A Hybrid system is a system that involves the interaction between discrete–time dynamics , continuous–time dynamics and discrete–events such as sudden impulses and switching. They provide an effective framework to model complex biological systems with large number of operating modes.

As an example, a hybrid system that will be used in this work is given by the following equation:

$$y_t = \sum_{i=1}^{n_a} a_i(\lambda_{t-1}) y_{t-i} + \sum_{i=1}^{n_c} c_i(\lambda_{t-1}) u_{t-i} \tag{3.3}$$

This model is known as an Autoregressive Exogenous (ARX) system where exogenous implies the model is also dependent on input along with previous states. The coefficients $c_i(\cdot)$ correspond to the input data and the coefficients $a_i(\cdot)$ correspond to the previous output data. The parameter λ_t is a switching variable that indexes the active model at a given time t. There are various ways in which switching transitions can be described. In this work we will use a *Jump Linear System (JLS)* model where λ is an unknown deterministic and finite-valued switching variable [11]. This class of models, when parameterized as above, are affine in their parameters.

Hybrid system identification provides the foundation for the identification of models described by (3.3) from input/output data. The identification approach minimizes the number of affine models needed, as well as the error associated with using the model to describe the data. This problem has been previously solved in [12] using a method called General Principal Component Analysis (GPCA) for the case when no measurement error (or noise) is present. Recently this approach has been extended to the case of noisy measurements using a greedy algorithm [13]

as well as a randomized hit-and-run algorithm [14] which in general do not find a global optimal solution but produce good results. There has also been a deterministic algorithm by Ozay *et al.* based on convex relaxations related to rank minimization [15]. The two algorithms primarily used in this thesis are the hit-and-run algorithm and the drop rank algorithm.

In the next section we will discuss the problem of hybrid system identification in more detail and elaborate on the algebraic reformulations that will allow us to solve this problem using the methods listed before. Lastly, we will present information regarding the two algorithms.

3.1 Problem Statement

Given noisy input/output measurements, find a hybrid system of the form

$$y_t = -\sum_{i=1}^{n_a} a_i(\lambda_{t-1})(y_{t-i} + e_{t-i}) + \sum_{i=1}^{n_b} b_i(\lambda_{t-1})u_{t-i} - e_t \tag{3.4}$$

that is consistent with the data.

In this formulation the error affects separately each measurement and the objective is to find the coefficients in the presence of data corrupted by noise. This formulation is ill posed, without additional constraints

there may be infinitely many solutions depending on the amount of error assumed. Therefore, a bound on the error needs to be placed. In addition, two further assumptions will be made: (i) The order of the model n_a is known and (ii) The number of models s is also known.

Note also that (3.4) is affine in the parameters.

In this work we will not be using any input in the model so (3.4) becomes

$$y_t = -\sum_{i=1}^{n_a} a_i(\lambda_{t-1})(y_{t-i} + e_{t-i}) - e_t \tag{3.5}$$

3.2 Hybrid Decoupling Constraint

During the identification process we do not know a priori which system is active at any point in time. The *hybrid decoupling constraint* introduced in [11] is a clever way to address this issue. First it forms the product of all the switching models (3.5)

$$p_k(\boldsymbol{e}, \boldsymbol{a}) \doteq \prod_{j=1}^{s} \left((y_t + e_t) + \sum_{i=1}^{n} a_i(j)(y_{t-i} + e_{t-i}) \right) = 0 \tag{3.6}$$

where $p_k(\boldsymbol{e}, \boldsymbol{a})$ is a polynomial in the error and coefficient variables. Introduce the following vectors

$$\boldsymbol{b}_i = \begin{bmatrix} 1 & a_1(i) & \dots & a_n(i) \end{bmatrix}^T,$$
$$\boldsymbol{r}_k = \begin{bmatrix} y_k - e_k & y_{k-1} - e_{k-1} & \dots & y_{k-n} - e_{k-n} \end{bmatrix}^T,$$

where $\boldsymbol{b}_i \in \mathbb{R}^n$ is the vector of parameters of the i^{th} model and $\boldsymbol{r}_k \in \mathbb{R}^n$ is the vector of the last n noisy measurements with respect to time k. This allows us to write the hybrid decoupling condition (3.6) as

$$p_k(\boldsymbol{e}, \boldsymbol{a}) = \prod_{i=1}^{s} (\boldsymbol{b}_i^T \boldsymbol{r}_k) = \prod_{i=1}^{s} (\boldsymbol{r}_k^T \boldsymbol{b}_i) = \nu_s(\boldsymbol{r}_k)^T \boldsymbol{c}_s = 0 \tag{3.7}$$

where $\boldsymbol{c}_s$ is a vector of coupled parameters and $\nu_s(\boldsymbol{r}_k)$ is a vector encoding information about the last n noisy measurements up to time k.

The collection of L measurements gives rise to the matrix equation

$$\boldsymbol{V}_s(\boldsymbol{r}, \boldsymbol{e})\boldsymbol{c} \doteq \begin{pmatrix} \nu_s(\boldsymbol{r}_1, \boldsymbol{e})^T \\ \vdots \\ \nu_s(\boldsymbol{r}_L, \boldsymbol{e})^T \end{pmatrix} \boldsymbol{c} = 0 \tag{3.8}$$

where $\boldsymbol{V}_s(\boldsymbol{r}, \boldsymbol{e})$ is called the Veronese matrix. The hybrid identification problem becomes one of finding a noise sequence e such that $\boldsymbol{V}_s$ has a non-trivial null space. The vector $\boldsymbol{c}$ that spans the kernel of the Veronese mapping provides the coupled coefficients from which the parameters of each switched system $\boldsymbol{b}_i$ can be extracted.

We will later discuss how the parameters of each switched system can be obtained from the solution to (3.8). Let us look at an example to further understand the Veronese matrix.

Example: Consider a hybrid system consisting of two second order AR system, *e.g.*, $s = 2$ and $n = 2$. The AR model equations are:

$$(y_t + a_1(y_{t-1} + e_{t-1}) + a_2(y_{t-2} + e_{t-2}) - e_t), \quad s = 1 \tag{3.9}$$

$$(y_t + b_1(y_{t-1} + e_{t-1}) + b_2(y_{t-2} + e_{t-2}) - e_t), \quad s = 2 \tag{3.10}$$

We now use the *hybrid decoupling constraint* and multiply the two.

$$\begin{aligned}(y_t + a_1(y_{t-1} + e_{t-1}) + a_2(y_{t-2} + e_{t-2}) - e_t)\cdot \\ (y_t + b_1(y_{t-1} + e_{t-1}) + b_2(y_{t-2} + e_{t-2}) - e_t) = 0\end{aligned} \tag{3.11}$$

The goal is to create the corresponding Veronese matrix and find the null space of it, in order to do this we first multiply (3.11) out to get (3.12).

$$\begin{aligned}&(y_t - e_t)^2 + (-a_1 - b_1)(y_t - e_t)(y_{t-1} - e_{t-1}) \\ &+ (-a_2 - b_2)(y_t - e_t)(y_{t-2} - e_{t-2}) + (a_1 b_1)(y_{t-1} - e_{t-1})^2 \\ &+ (a_1 b_2 + a_2 b_1)(y_{t-1} - e_{t-1})(y_{t-2} - e_{t-2}) \\ &+ (a_2 b_2)(y_{t-2} - e_{t-2})^2\end{aligned} \tag{3.12}$$

Now the coupling of the coefficients of (3.7) can factored from the measurements as show below

$$\boldsymbol{c}_s = [1, (-a_1 - b_1), (-a_2 - b_2), (a_1 b_1), (a_1 b_2 + a_2 b_1), (a_2 b_2)]^T \tag{3.13}$$

$$\begin{aligned} v_s(\boldsymbol{r}_k, e) = [&(y_t - e_t)^2, (y_t - e_t)(y_{t-1} - e_{t-1}), \\ &(y_t - e_t)(y_{t-2} - e_{t-2}), (y_{t-1} - e_{t-1})^2, \\ &(y_{t-1} - e_{t-1})(y_{t-2} - e_{t-2}), (y_{t-2} - e_{t-2})^2]^T \end{aligned} \tag{3.14}$$

Additionally we can substitute values for y and create the Veronese matrix we would like to find the null space of. Let's assume that the noise free measurements are $y = [1, 2, 3, 4, 5]$. Furthermore consider a window of length 5, *e.g.* $L = 5$. Then the Veronese matrix of the dimension 5×6 or $L \times q$ where $q = \binom{n+s}{n}$) is given by

$$\boldsymbol{V}_s = \begin{bmatrix} (e_1 - 1.0)^2 & 0 & 0 & 0 & 0 & 0 \\ (e_2 - 2.0)^2 & (e_1 - 1.0)(e_2 - 2.0) & 0 & (e_1 - 1.0)^2 & 0 & 0 \\ (e_3 - 3.0)^2 & (e_3 - 3.0)(e_2 - 2.0) & (e_3 - 3.0)(e_1 - 1.0) & (e_2 - 2.0)^2 & (e_1 - 1.0)(e_2 - 2.0) & (e_1 - 1.0)^2 \\ (e_4 - 4.0)^2 & (e_3 - 3.0)(e_4 - 4.0) & (e_2 - 2.0)(e_4 - 4.0) & (e_3 - 3.0)^2 & (e_3 - 3.0)(e_2 - 2.0) & (e_2 - 2.0)^2 \\ (e_5 - 5.0)^2 & (e_5 - 5.0)(e_4 - 4.0) & (e_3 - 3.0)(e_5 - 5.0) & (e_4 - 4.0)^2 & (e_3 - 3.0)(e_4 - 4.0) & (e_3 - 3.0)^2 \end{bmatrix}$$

Note that in the presence of noise each row of the Veronese matrix is a polynomial in the noise samples.

If the noise sequence e is known the coupled parameters vector $\boldsymbol{c}$ can be found solving the linear algebra problem $\boldsymbol{V}_s(\boldsymbol{e})\boldsymbol{c} = 0$. The particular case

when the error e is 0 corresponds to the GPCA formulation proposed in [12]. In practice even if the measurements were noise free, it is possible that the Veronese matrix is not deficient for a given numerical accuracy. Therefore, the solution is found find the singular vector of the Veronese matrix corresponding to the smallest singular value (which in the ideal case is zero).

Since in the noisy measurements case the error sequence e is not known, the hybrid decoupling constraints imposes a system of polynomial equations. Since now it is necessary to find both $\boldsymbol{e}$ and $\boldsymbol{x}$ the problem becomes significantly more difficult. An approach often used is to alternatively fix $\boldsymbol{e}$ and $\boldsymbol{c}$ and solve for the variables not fixed. This is essentially the approach that the hit-and-run algorithm take. It fixes $\boldsymbol{c}$ and solves a polynomial problem to find $\boldsymbol{e}$ and then fixes $\boldsymbol{e}$ and solves an SVD problem to obtain $\boldsymbol{c}$ in a randomized fashion. Note that this approach does not guarantee convergence but in practice leads to acceptable results, on average.

3.3 Extraction of Model Parameters

While solving for the coupled parameters is a significant part of the problem, the extraction of the individual model parameters is also of major importance. Given the data points, we treat the problem as a cluster

which is made up of a union of different hyperplanes. These hyperplanes correspond to the different models, and since we specify the number of sub models, we know the number of hyperplanes that exist, and the problem is formulated as finding s hyperplanes where the individual model parameters $\{\boldsymbol{b}_i\}_{i=1}^{n}$ are the normal vectors. This was introduced by Vidal in his dissertation which can be found in [16]. This is done with the polynomial differentiation algorithm in which each hyperplane is represented by the polynomial at a given data point which belongs to the hyperplane. Then by taking the derivative at a given point, the model parameters for the given data points can be extracted. This can be represented as follows:

$$\boldsymbol{b}_i = \frac{Dp_i(\boldsymbol{y}_i)}{||Dp_i(\boldsymbol{y}_i)||} \tag{3.15}$$

However, without knowing *a priori* which data point belongs to which hyperplane, it is not possible to solve for the parameters. Therefore first some data points are chosen such that they lie inside a given hyperplane and not at any intersection of two. This is done by selecting the data points with the smallest geometric distance between the point and a hyperplane. (3.16) shows that this avoids the problem of finding data points at intersections due to dividing by the derivative which would be approximately zero at an intersection.

$$\boldsymbol{y}_n = \arg \min_{\boldsymbol{x} \in \boldsymbol{X}: Dp_n(\boldsymbol{x}) \neq 0} \frac{|p_n(\boldsymbol{x})|}{||Dp_n(\boldsymbol{x})||} \tag{3.16}$$

By switching between solving for (3.16) and (3.15) all of the model parameters can be extracted. The algorithm is as follows: It is important to

Algorithm 1 Polynomial Differentiation Algorithm (PDA)

solve $L_n \boldsymbol{c}_n = 0$
set $p_n(\boldsymbol{x}) = \boldsymbol{c}_n^T v_n(\boldsymbol{x})$
for $i = n : 1$,
  $\boldsymbol{y}_i = \arg\min_{\boldsymbol{x} \in \boldsymbol{X} : Dp_i(\boldsymbol{x}) \neq 0} \dfrac{|p_i(\boldsymbol{x})|}{||Dp_i(\boldsymbol{x})||}$
  $\boldsymbol{b}_i = \dfrac{Dp_i(\boldsymbol{y}_i)}{||Dp_i(\boldsymbol{y}_i)||}$
  $p_{i-1}(\boldsymbol{x}) = \dfrac{p_i(\boldsymbol{x})}{\boldsymbol{b}_i^T \boldsymbol{x}}$,
end;

note that starting with the initial solution of the coupled coefficients c_n, coefficients for c_{n-i} also need to be found. This can be done by exploiting the structure of the coupled coefficients as the number of models change.

3.4 Numerical Algorithms

We will now discuss the two main numerical algorithms that were implemented in this thesis. The main algorithm used is a randomized algorithm that does not guarantee convergence but can be used to obtain good results. This algorithm is taken from Feng *et al.* in [14]. The second algorithm is a deterministic algorithm based on reducing the rank of the Veronese matrix through convex relaxations developed by Fazel *et al.* This formulation can be found in [17].

3.4.1 Hit-and-Run Algorithm

The purpose of the algorithm is to find an admissible noise sequence e such that the smallest singular value of the Veronese matrix, evaluated at that noise sequence, is approximately zero and then to extract the vector corresponding to that singular value.

First we choose the window length L, *e.g.* the number of data samples to form the Veronese matrix. The we choose an initial random guess for the coupled parameters vector $\boldsymbol{x}_0$ such that $\|\boldsymbol{x}_o\| = 1$

Then the algorithm starts by fixing the vector $\boldsymbol{x}$ and solving for the noise vector $\boldsymbol{e}$ (at initialization $\boldsymbol{x} = \boldsymbol{x}_0$ as above)

$$\begin{aligned} \boldsymbol{e}^* &= \arg\min_{e} ||\boldsymbol{V}_s(\boldsymbol{r},\boldsymbol{e})\boldsymbol{x}||_2^2 \\ s.t.\ &||\boldsymbol{e}||_\infty \leq \bar{e} \end{aligned} \tag{3.17}$$

Once the admissible error vector e (of dimension L) is obtained, it is fixed and we proceed to find a new coupled parameter vector $\boldsymbol{x}^*$ such that

$$\begin{aligned} \boldsymbol{x}^* &= \arg\min_{x} ||\boldsymbol{V}_s(\boldsymbol{r},\boldsymbol{e})\boldsymbol{x}||_2^2 \\ s.t.\ &||\boldsymbol{x}||_2^2 = 1 \end{aligned} \tag{3.18}$$

These two steps are repeated alternatively until the objective value of (3.17) and (3.18) converge adequately close to each other. Note that this "convergence" criterion may lead us to a local minimum.

The fourth step is to use the values of $\boldsymbol{x}^*$ and $\boldsymbol{e}^*$ resulting from the above iterative process and find the 2-norm squared of the image of $\boldsymbol{x}^*$ under the Veronese mapping:

$$\gamma = ||\boldsymbol{V}_s(\boldsymbol{r}, \boldsymbol{e}^*)\boldsymbol{x}^*||_2^2 \tag{3.19}$$

The value γ found in (3.19) is then used in the final hit-and-run section, where a random direction $\boldsymbol{d}$ is taken if it improves the overall objective. The final step involves solving for two variables, a step size β and the error sequence $\boldsymbol{e}$. The procedure for picking a direction which improves the objective function and its justification can be found in [14]. 3.20 shows the final step required to finish the algorithm.

$$\begin{aligned}(\boldsymbol{e}, \beta) = \arg\min\ & ||\boldsymbol{V}_s(\boldsymbol{r}, \boldsymbol{e})(\boldsymbol{x} + \beta\boldsymbol{d})||_2^2 - \gamma||\boldsymbol{x} + \beta\boldsymbol{d}||_2^2 \\ s.t.\ & ||\boldsymbol{e}||_\infty \leq \bar{e}\end{aligned} \tag{3.20}$$

These steps are all put together and the final algorithm can be seen in Algorithm 2.

3.4.2 Drop Rank

The second algorithm is the Drop Rank algorithm developed by (Ozay and Feng [18]). It is based on the idea that once some admissible noise e is found such that $\boldsymbol{V}_s(\boldsymbol{r}, e)$ is rank deficient, the coupled parameters can then be extracted from the null space. Although rank minimization is an NP-hard problem, recent relaxations have been found to be good

Algorithm 2 Hit-and-Run Algorithm

step 1: Generate random vector $\boldsymbol{x}_0$ with $||\boldsymbol{x}_0|| = 1$
repeat
 repeat
  Step 2: $\boldsymbol{x} \leftarrow \boldsymbol{x}^*$ and solve:
  $\zeta:\ \boldsymbol{e}^* \leftarrow \arg\min_{\boldsymbol{e}, ||\boldsymbol{e}||_\infty \leq \bar{e}} ||\boldsymbol{V}_s(\boldsymbol{r}, \boldsymbol{e})\boldsymbol{x}||_2^2$
  Step 3: $\boldsymbol{e} \leftarrow \boldsymbol{e}^*$ and solve:
  $\eta:\ \boldsymbol{x}^* \leftarrow \arg\min_{\boldsymbol{x}, ||\boldsymbol{x}||_2^2 = 1}\ ||\boldsymbol{V}_s(\boldsymbol{r}, \boldsymbol{e})\boldsymbol{x}||_2^2$
 until
  $\zeta \approx \eta$

 Step 4: $\gamma \leftarrow ||\boldsymbol{V}_s(\boldsymbol{r}, \boldsymbol{e}^*)\boldsymbol{x}^*||_2^2$
 repeat
  Step 5: Pick a random direction $\boldsymbol{d}$ and solve:
  $(\boldsymbol{e}, \beta) \leftarrow \arg\min_{||\boldsymbol{e}||_\infty \leq \bar{e}}\ ||\boldsymbol{V}_s(\boldsymbol{r}, \boldsymbol{e})(\boldsymbol{x} + \beta\boldsymbol{d})||_2^2 - \gamma||\boldsymbol{x} + \beta\boldsymbol{d}||_2^2$

 until an improving direction is found within a number of iterations
until a number of iterations
return $\boldsymbol{x}^*$

surrogates [19]. The trace of an embedding of the positive semidefinite Veronese matrix is minimized. The algorithm is summarized below. Where $\boldsymbol{X}$ corresponds to the noisy Veronese matrix. After the Veronese

Algorithm 3 Drop Rank Algorithm

$\boldsymbol{X} \in \mathbb{R}^{m \times n}$
$\boldsymbol{W}_y = I_{m \times m}$
$\boldsymbol{W}_z = I_{n \times n}$
Solve

$$\min_{\boldsymbol{X}, \boldsymbol{Y}, \boldsymbol{Z}} \ \text{trace} \begin{bmatrix} \boldsymbol{W}_y \boldsymbol{Y} & 0 \\ 0 & \boldsymbol{W}_z \boldsymbol{Z} \end{bmatrix}$$
$$\text{subject to} \begin{bmatrix} \boldsymbol{Y} & \boldsymbol{X} \\ \boldsymbol{X}^T & \boldsymbol{Z} \end{bmatrix} \succeq 0$$

Decompose $\boldsymbol{X} = \boldsymbol{U}\boldsymbol{D}\boldsymbol{V}^T$ via SVD.
return $\boldsymbol{X}$

matrix is found, the vector from the right null space corresponding to the minimum singular value of $\boldsymbol{X}$ will represent the coupled parameters.

Chapter 4

Problem Formulation

In this chapter we will discuss the problem of sleep staging and how hybrid system identification may be applied to get meaningful results. We will first discuss the use of Electroencephalogram (EEG) signals in sleep staging along with the different stages of sleep. We will then discuss the aspect of sleep staging to which we will apply hybrid system identification. Finally we will discuss the dataset that will be used along with the preprocessing that will be done.

4.1 Sleep Staging

Sleep staging is the classification of sleep into 5 different stages. These stages are 1, 2, 3, 4 and REM. Sleep staging has been getting increasing visibility due to advances in machine learning techniques that no longer require sleep technicians to be awake with a patient all night to score their EEG signals. EEG signals are used for sleep staging as they contain

distinct information regarding different stages of sleep. One of these descriptors, which we will focus on for the remainder of the thesis, are spindles found in stage two of sleep. Additional information regarding EEG signal processing and sleep staging can be found in [20].

4.2 Spindle Dataset

Spindles are bursts of brain activity visible on EEG signals during stage two of sleep [21]. They can occur due to various reasons, but are generally characteristic of tranquil sleep. In order for an interval of EEG signal to be characterized as a spindle it not only must exhibit oscillatory behavior but this behavior must also last longer than 0.5 seconds as described in [21].

The sleep spindles dataset used in this thesis is taken from [22]. From this dataset, Excerpt 1 was used which is sampled at a frequency of 100 Hz. The data is of a woman 51 years of age taken from the C3-A1 EEG channel which is located on the central left side of the brain. The Excerpt was annotated by two experts who found 52 and 115 spindles respectively and the data collected from the patient was done so over a 30 minute time period. The results presented in the next section are found using the annotations of expert one. However, the programming methods written are general and can be used to extract all possible spindles

from the dataset.

Before the data could be analyzed using hybrid system identification, it was cleaned and extracted from eight separate EDF files. EDF stands for the European Data Format and is used to compress data into a standardized format. A function for parsing EDF files and creating a Matlab structure was used which can be found in [23]. This method was used to write an algorithm to systematically go through each Excerpt.edf file and place the data into a larger structured form which can be used later for learning. Preprocessing was done on the EEG signal via the use of a bandpass filter and had the range of 0.3 Hz to 35Hz. These requirements are described in [21]. The code for this filtering was taken from [24].

We are now ready to apply the methods in this thesis for spindle detection. We do not present the code related to extracting of the data as it is not necessary for the discussion of the problem. However, any code written in this thesis is available upon request.

Chapter 5

Results and Analysis

In this chapter we will describe the tests that were run and the results that were obtained. We will begin by showing a synthetic test that was generated to demonstrate the approach. We will then show some results obtained from applying the methods outlined in this thesis. Various model orders will be shown along with varying error tolerances. We will then conclude with some test results obtained from testing on raw data along with exploring the different parameters possible.

5.1 Synthetic Test

We first created a test in order to show what the idea of our approach to spindle detection is. We took a spindle found from the dataset as ground truth and implanted it into randomly generated noise with half the amplitude of the spindle. This can be seen in Figure 5.1 where it is clear that the spindle can be seen from approximately $t = 100$ to

$t = 250$.

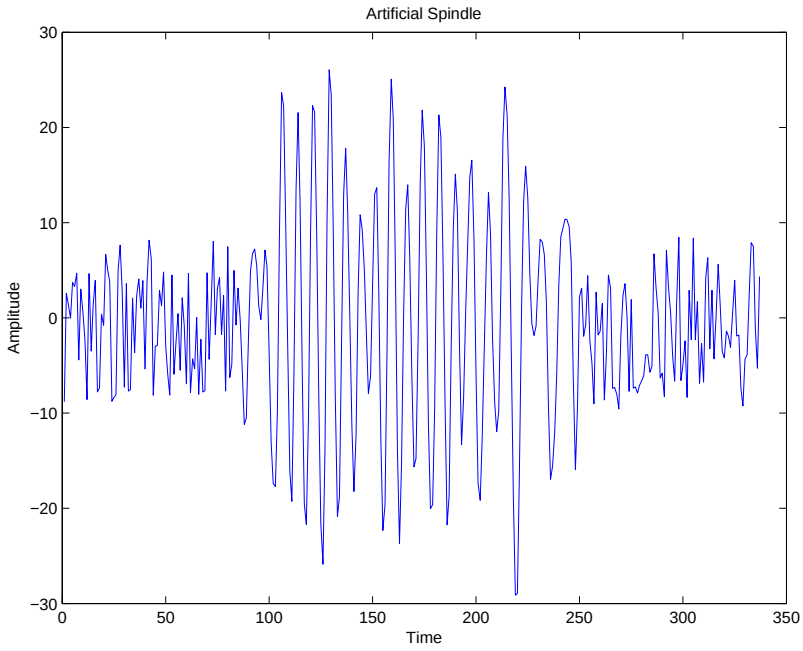

Figure 5.1: Synthetic Spindle Example Used to Show Our Method of Detection.

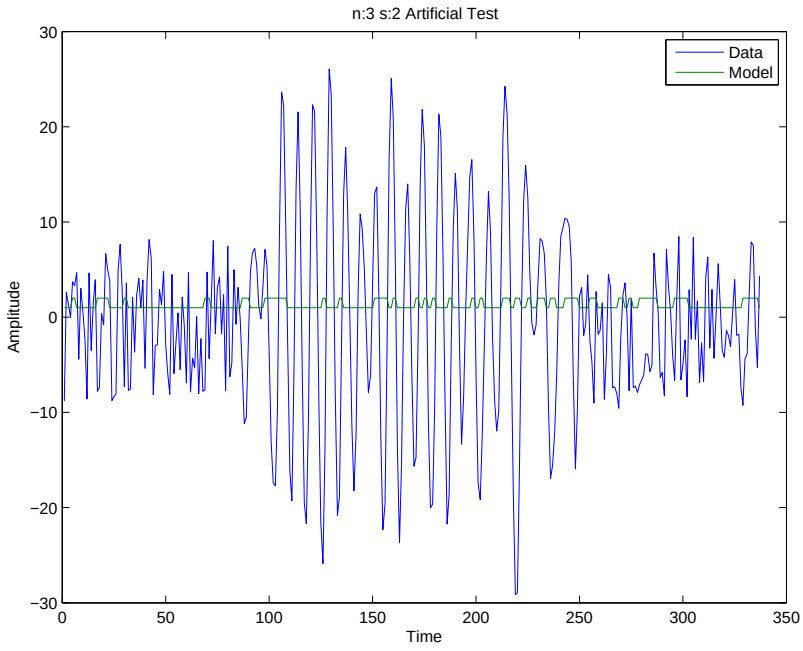

Figure 5.2: Synthetic Spindle Example Using GPCA with Two Switched Models of Order 3 Overlaid on the Data.

We then used GPCA to find two switched models and increased the model order from 3 up to 6 in order to see its effect on classification. Figure 5.2 on the previous page shows the result for a model order of 3. It can be seen that due to additional noise in the spindle there is a significant amount of spikes (which we label as sparse classification) where the opposite model happens to be a better fit for a brief moment of time. Therefore with a model order of 3 the GPCA cannot capture all of the data in the spindle. So the model order has to be increased. Figure 5.3 shows the case where the model order is set to 4. It can be

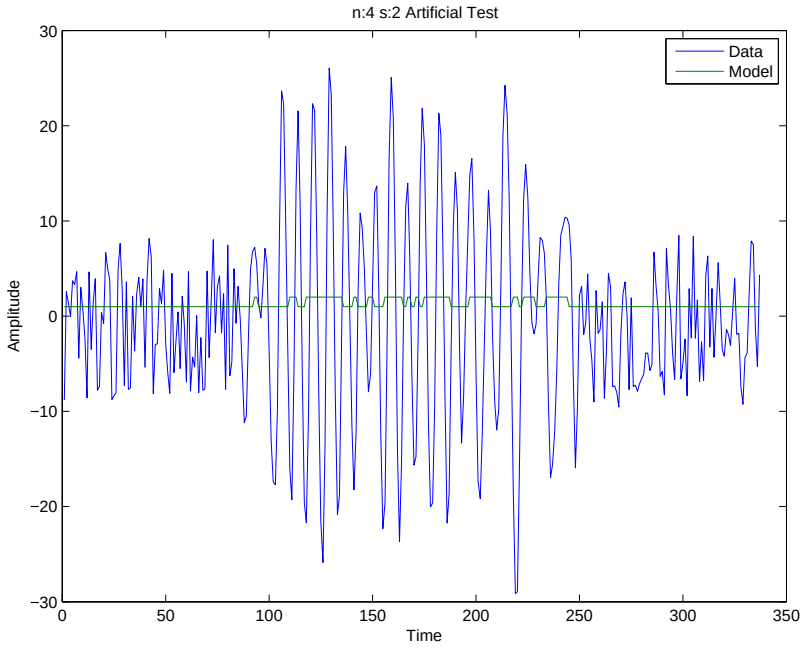

Figure 5.3: Synthetic Spindle Example Using GPCA with Two Switched Models of Order 4 Overlaid on the Data.

seen that there is much better separation of spindle and noise and far fewer jumps. This seems like a good candidate for the Hit-and-Run (HR)

algorithm.

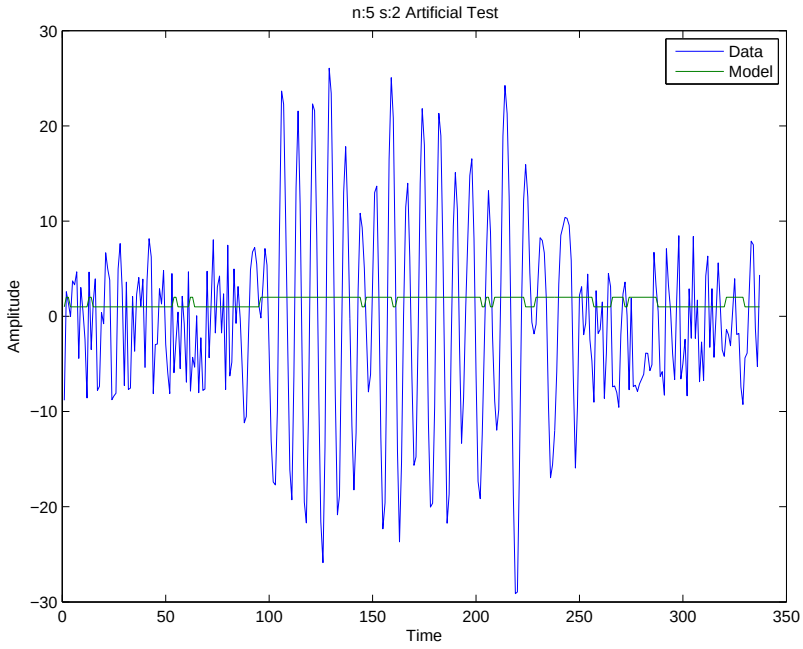

Figure 5.4: Synthetic Spindle Example Using GPCA with Two Switched Models of Order 5 Overlaid on the Data.

Figures 5.4 and 5.5 have model order of 5 and 6 respectively. Model order of 5 is a good candidate for the HR algorithm as it generates less jumps but also begins to learn the noise pattern. This can be seen further in Figure 5.5 where all of the data is classified as one model with few sparse switches. This means that the model order is too high to separate the data accurately. Let's take a look at how the hit and run algorithm performs on the same data with model orders of 4 and 6.

Figures 5.6 and 5.7 show the results from the HR algorithm.

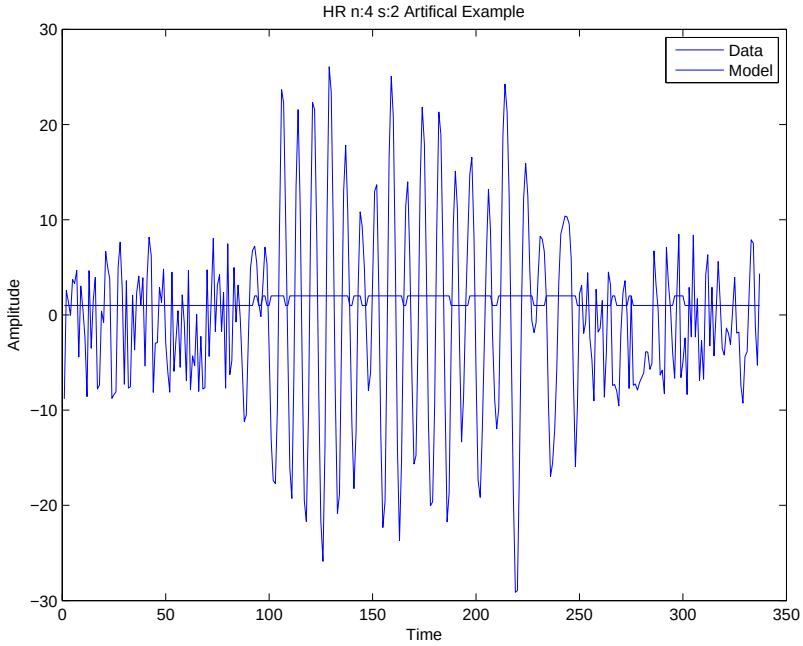

Figure 5.6: Synthetic Spindle Example Using HR with Two Switched Models of Order 4 Overlaid on the Data.

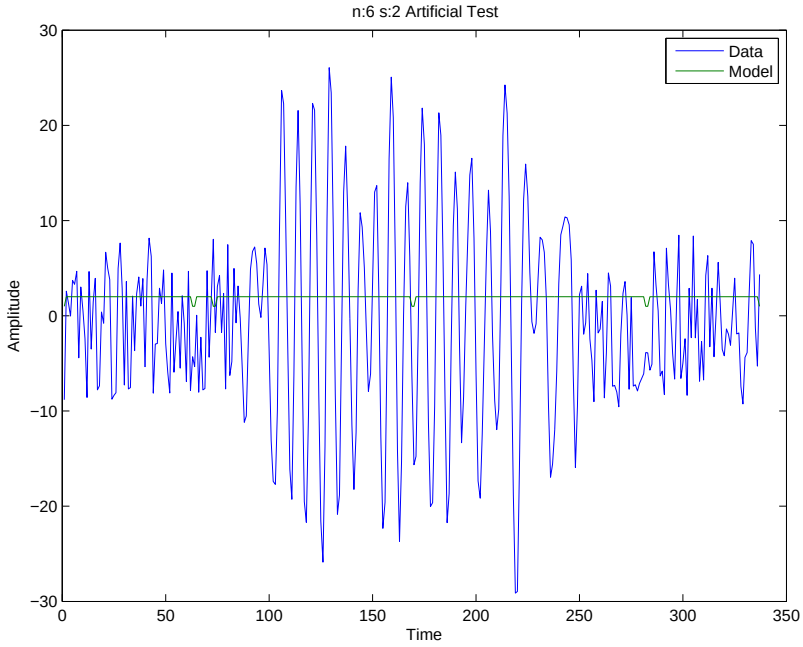

Figure 5.5: Synthetic Spindle Example Using GPCA with Two Switched Models of Order 6 Overlaid on the Data.

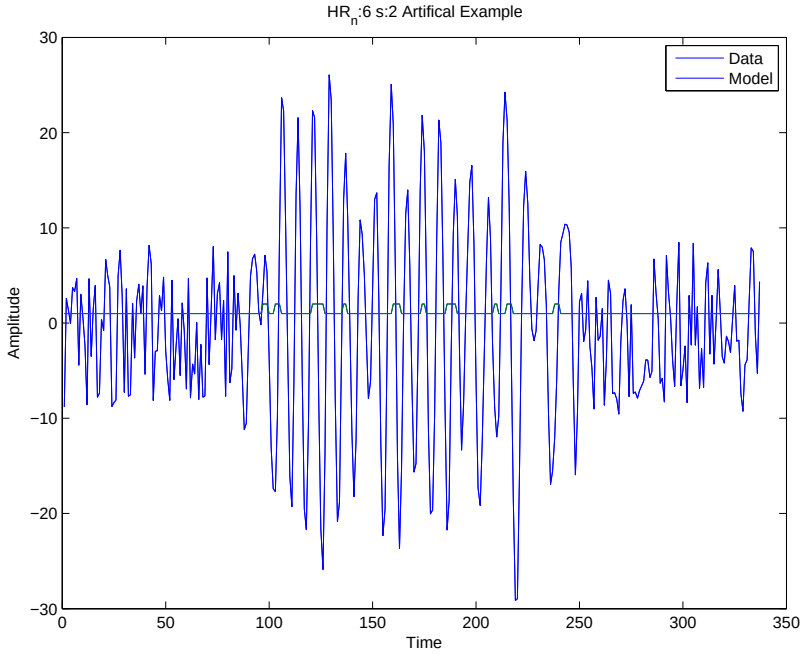

Figure 5.7: Synthetic Spindle Example Using HR with Two Switched Models of Order 6 Overlaid on the Data.

It can be seen that with an error tolerance of 5% the HR algorithm with model order 4 performed similarly to the GPCA algorithm with model order 5. This shows that within some error tolerances the HR algorithm can dismiss noise that would otherwise cause switching in the GPCA algorithm. Before we begin tests on the real data, it is important to mention that a relaxation order of 4 was chosen when applying the HR algorithm. Although this relaxation order may not obtain globally optimal results, it provided an objective gap between step 1 and step 2 of the HR algorithm that was sufficiently close in a reasonable amount of time.

5.2 Real Data with Model Order

In this section we show the application of hybrid system identification to find spindles in EEG signals that also contain real non-spindle data. This is raw EEG signal data that will be windowed around spindle information so that there is enough variance in the data to obtain meaningful separation. We show two separate cases where in the first, a window is taken where the spindle is at the center, and in the second the spindle information is located in the beginning of the window. We show how model order as well as window size affect model separation.

5.2.1 Spindle Amidst Error

We start by applying GPCA on the annotated data in order to find good candidates for the HR algorithm. Nine candidates are presented below, where the model order for each candidate ranges from 4 to 6. We begin by comparing the GPCA algorithm with the HR algorithm on a system with model order 4. Figure 5.8 shows the GPCA separation on the data while Figure 5.9 shows the HR algorithm applied on the same data. Firstly, it is important to note that the actual spindle lies between $t = 100$ to $t = 200$ and is surrounded by the rest of the EEG signal on both sides for the same duration as the spindle occurs.

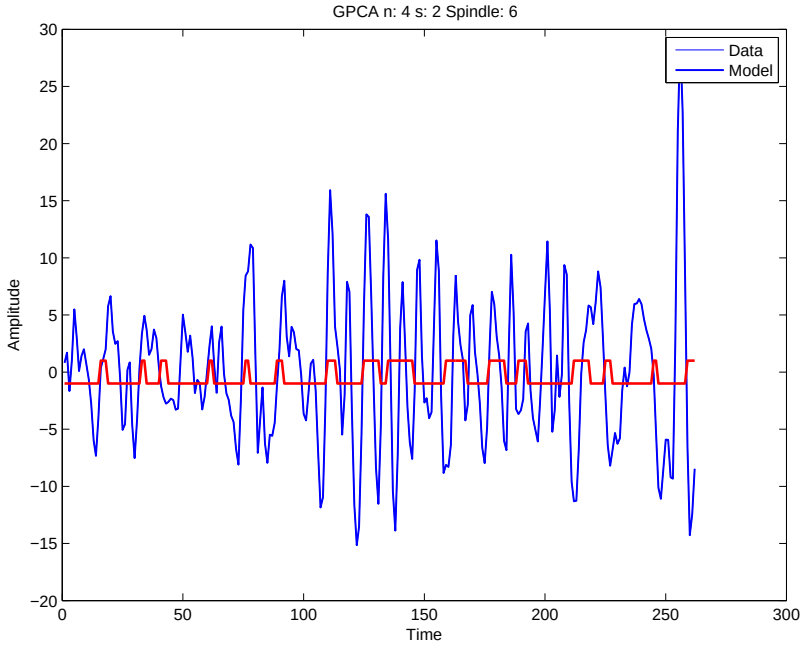

Figure 5.8: Spindle 6 Taken from an Annotated EEG Signal with Two Switched Models of Order 4 Overlaid on the Data.

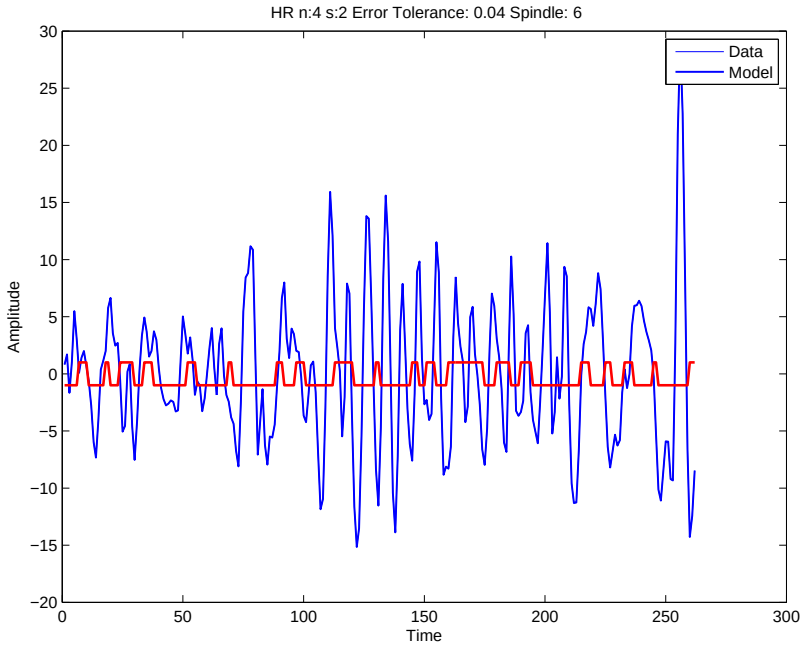

Figure 5.9: Spindle 6 Taken from an Annotated EEG Signal with Two Switched Models of Order 4 Found Through HR Overlaid on the Data.

We observe in Figure 5.8 that one model characterizes mostly the spindle information while the other characterizes the remainder of the information. However, there does exist some switching over the spindle (e.g. $t = 150$) which might be attributed to the glitch found in the spindle. It is also important to note that in this window there does not seem to be many distinguishing features which separate the spindle from other information, making differentiation between the two difficult. Figure 5.9 shows the HR algorithm applied on the same data with an error tolerance of 4%. Slightly worse results can be observed in the HR algorithm which leads us to believe that the error tolerance specified might have been too great, or in the cases where there was contention, the opposite (non-spindle) model represented the data better.

Figures 5.10 and 5.11 below show a similar test except on a different spindle. Unlike the last case, here there seems to be more differentiation. The HR algorithm was able to obtain good separation with an error tolerance of 4%. It can be noticed that as the error tolerance increased, the number of sparse spikes also increased.

Figures 5.13 and 5.15 show that an error tolerance which is too high acts equivalently as trying to over-fit a higher order AR model on the data. This is due to a higher order model being able to contain and describe more variety and makes switching obsolete.

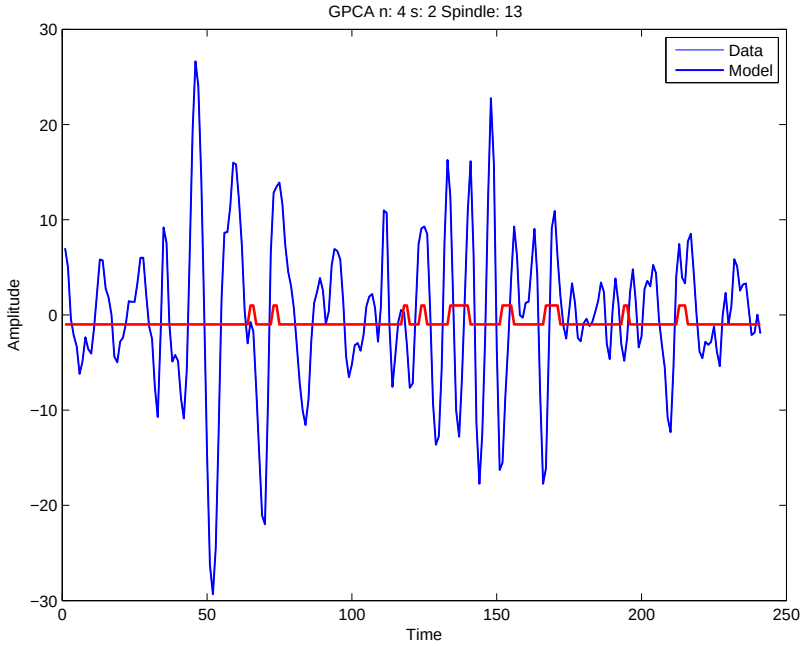

Figure 5.10: Spindle 13 Taken from an Annotated EEG Signal with Two Switched Models of Order 4 Overlaid on the Data.

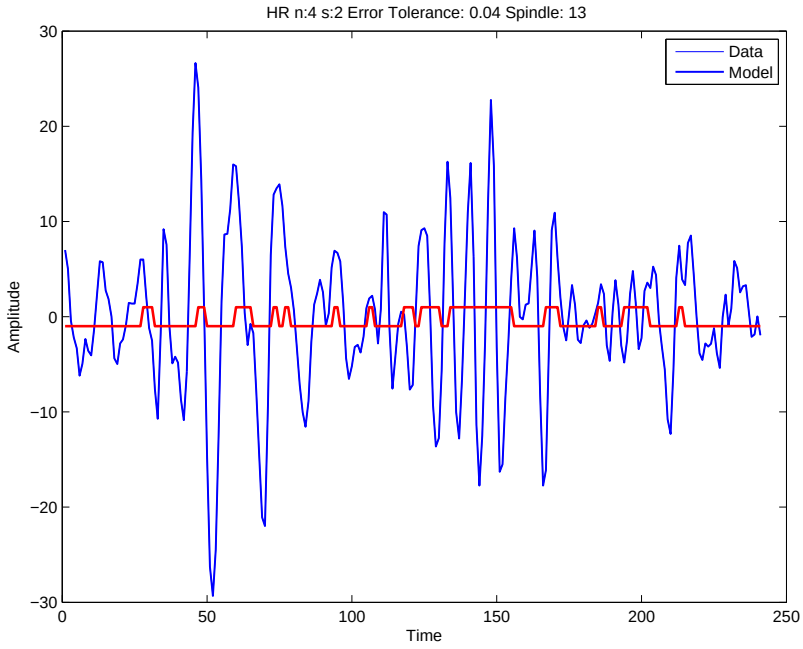

Figure 5.11: Spindle 13 Taken from an Annotated EEG Signal with Two Switched Models of Order 4 Found Through HR Overlaid on the Data.

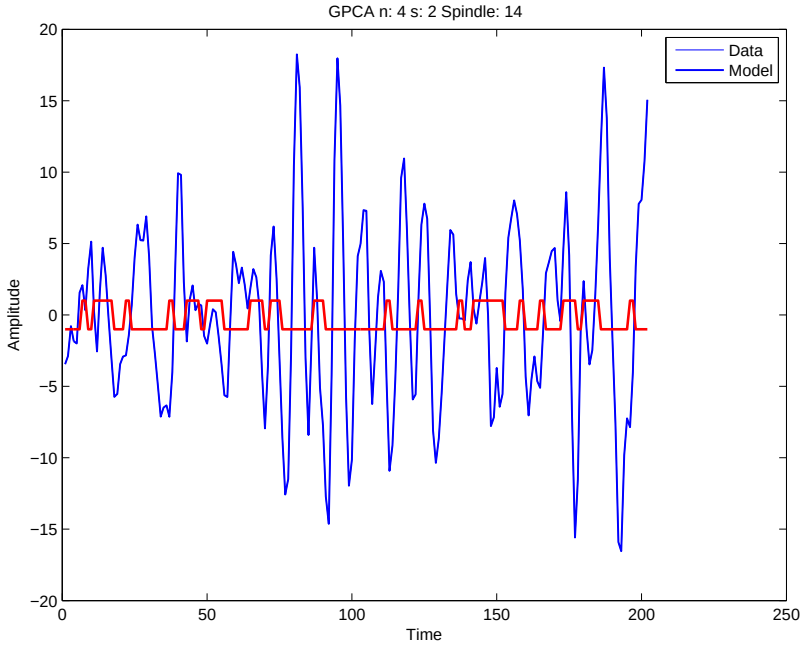

Figure 5.12: Spindle 14 Taken from an Annotated EEG Signal with Two Switched Models of Order 4 Overlaid on the Data.

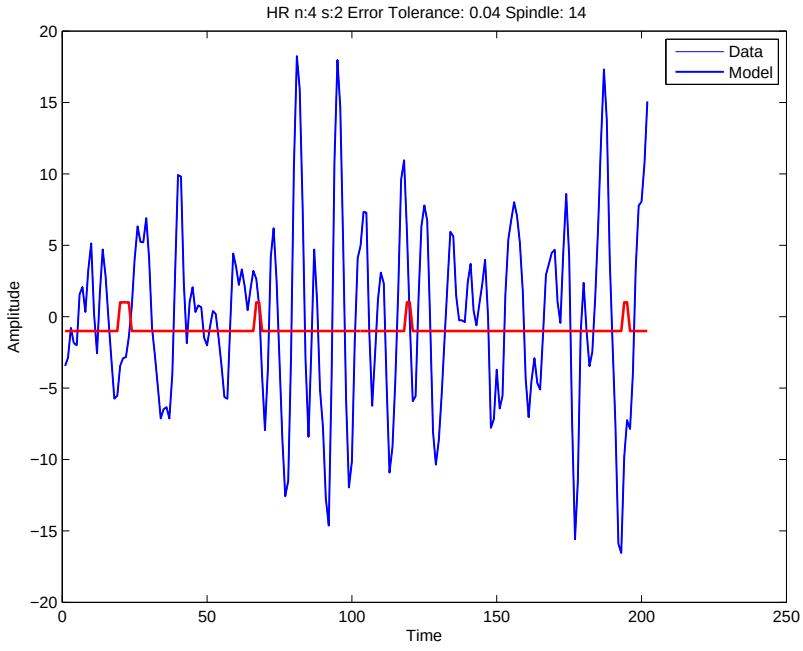

Figure 5.13: Spindle 14 Taken from an Annotated EEG Signal with Two Switched Models of Order 4 Found Through HR Overlaid on the Data.

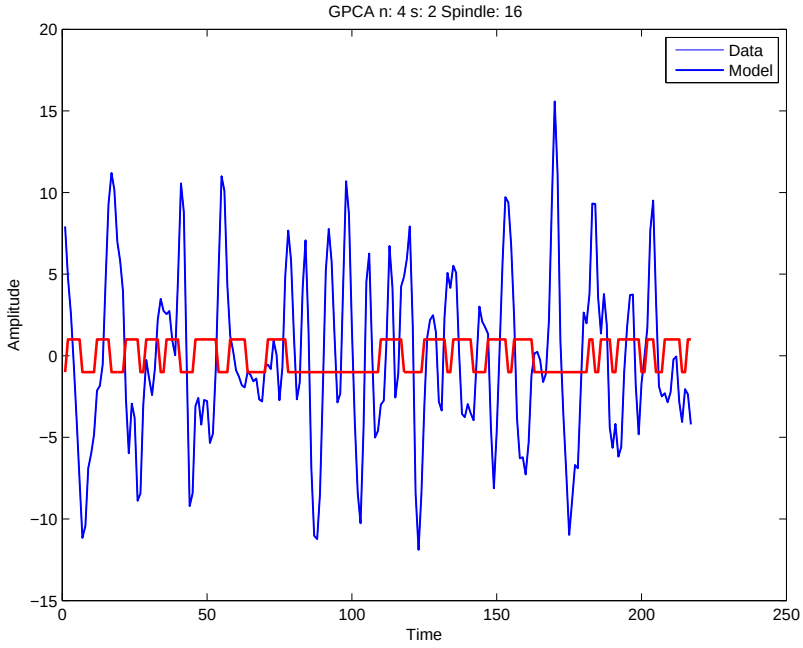

Figure 5.14: Spindle 16 Taken from an Annotated EEG Signal with Two Switched Models of Order 4 Overlaid on the Data.

In Figures 5.14 and 5.16 in the GPCA model we are able to observe separation between the spindle and other data while using HR, as shown in Figures 5.15 and 5.17, there seems to be information lost. This again shows a limitation in manually picking the model order and the error tolerance. This can be assumed as this is not the case with the GPCA algorithm.

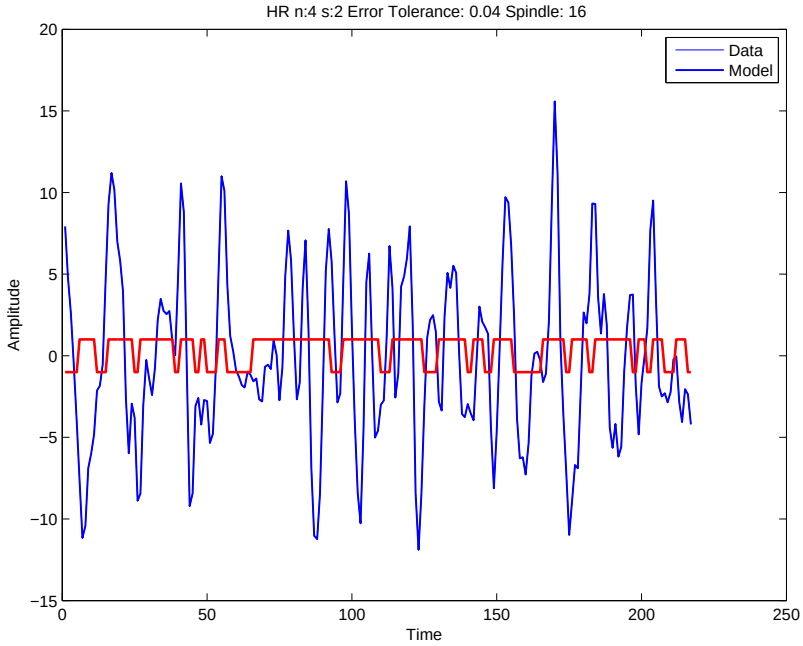

Figure 5.15: Spindle 16 Taken from an Annotated EEG Signal with Two Switched Models of Order 4 Found Through HR Overlaid on the Data.

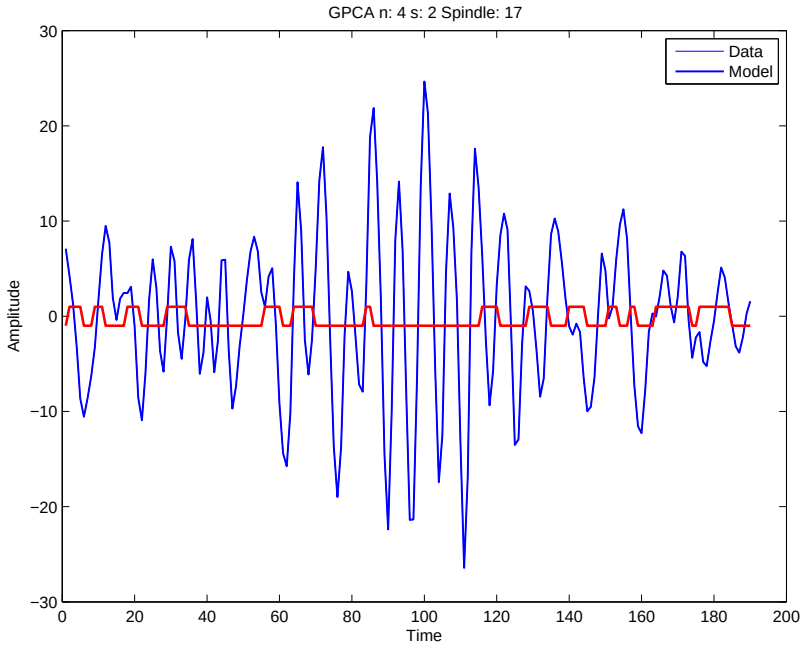

Figure 5.16: Spindle 17 Taken from an Annotated EEG Signal with Two Switched Models of Order 4 Overlaid on the Data.

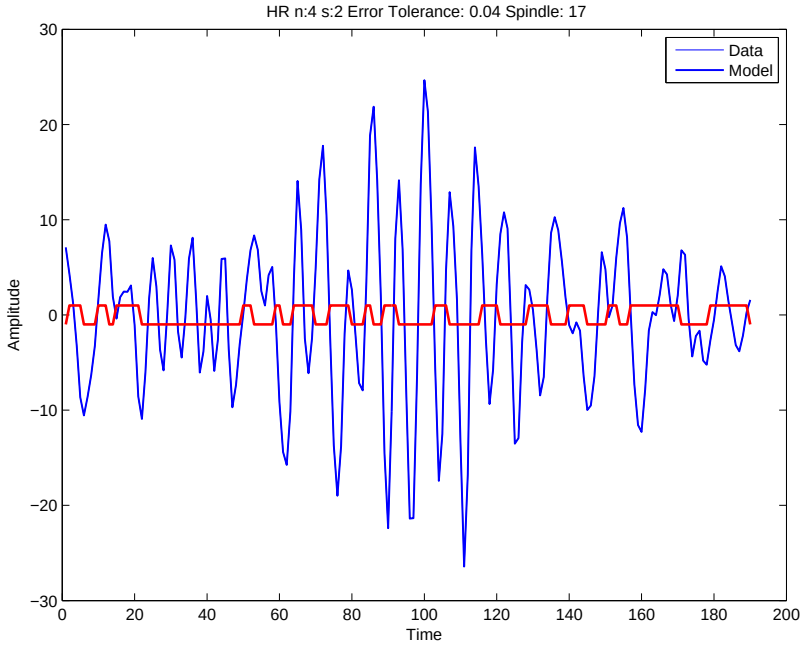

Figure 5.17: Spindle 17 Taken from an Annotated EEG Signal with Two Switched Models of Order 4 Found Through HR Overlaid on the Data.

Figures 5.18 and 5.19 show results similar to the synthetic tests that were run in the previous section. It can be seen that the spindle is well differentiated. Moreover, the HR algorithm seems to produce better results than the GPCA. The test done in these figures used a model order of 5, one higher than the tests shown in Figures 5.12 and 5.13, and obtained better results. This implies that a model order of 5 is more descriptive of the underlying data and can be used to describe additional features. Figures 5.20 and 5.21 show the same spindle from Figures 5.14 and 5.15 except with models of order 5 applied to the data. Slightly better results can be seen through the GPCA algorithm.

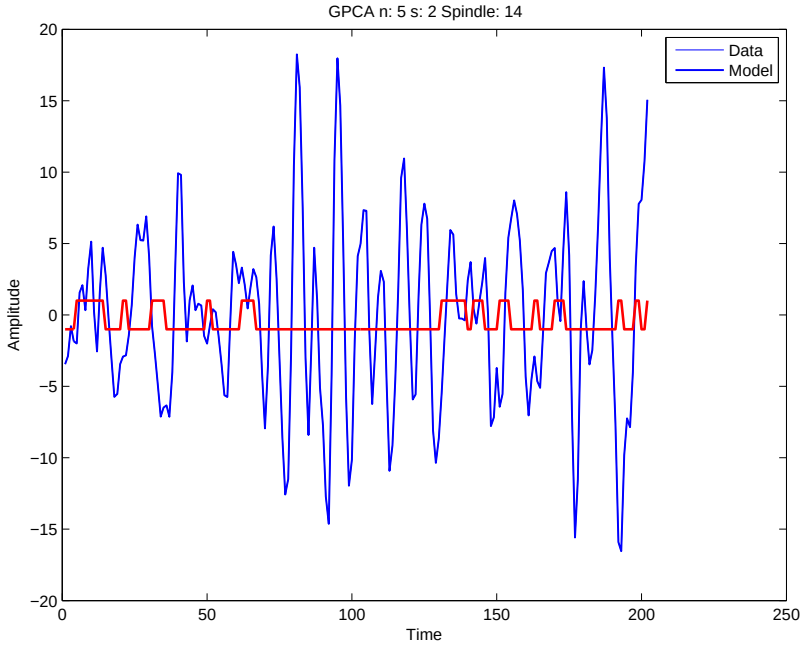

Figure 5.18: Spindle 14 Taken from an Annotated EEG Signal with Two Switched Models of Order 5 Overlaid on the Data.

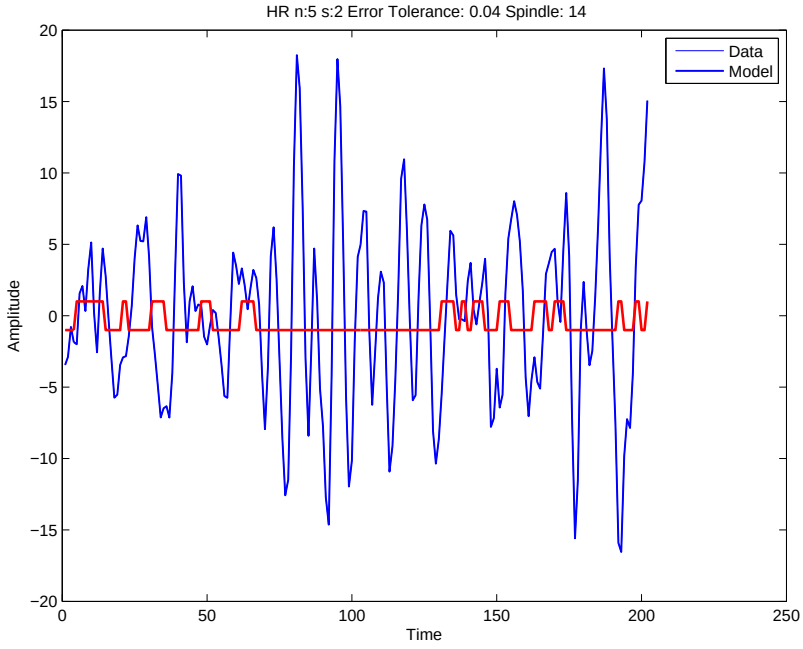

Figure 5.19: Spindle 14 Taken from an Annotated EEG Signal with Two Switched Models of Order 5 Found Through HR Overlaid on the Data.

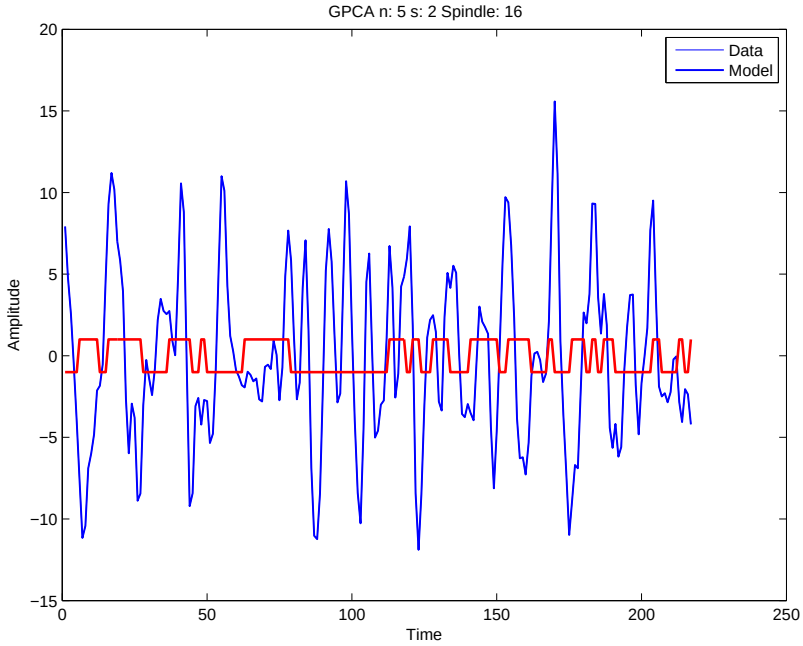

Figure 5.20: Spindle 16 Taken from an Annotated EEG Signal with Two Switched Models of Order 5 Overlaid on the Data.

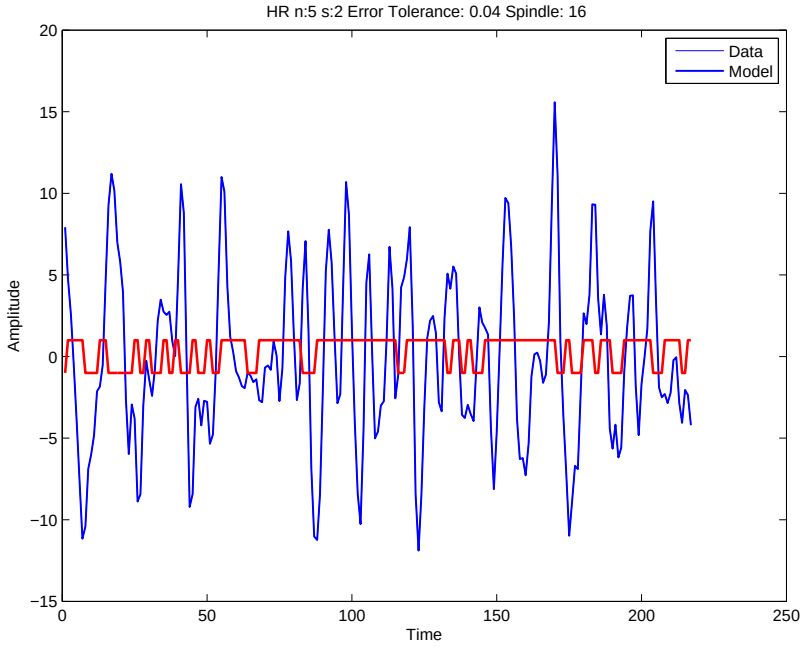

Figure 5.21: Spindle 16 Taken from an Annotated EEG Signal with Two Switched Models of Order 5 Found Through HR Overlaid on the Data.

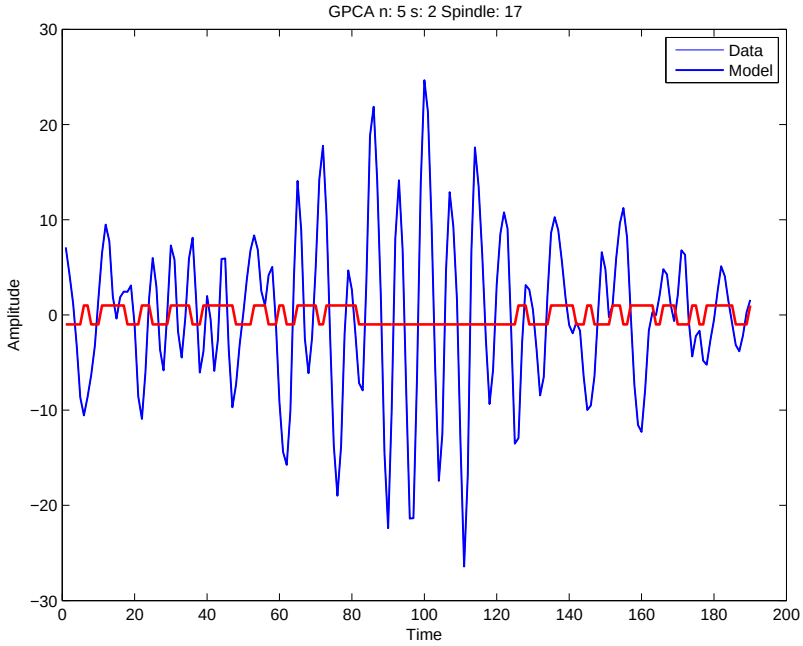

Figure 5.22: Spindle 17 Taken from an Annotated EEG Signal with Two Switched Models of Order 5 Overlaid on the Data.

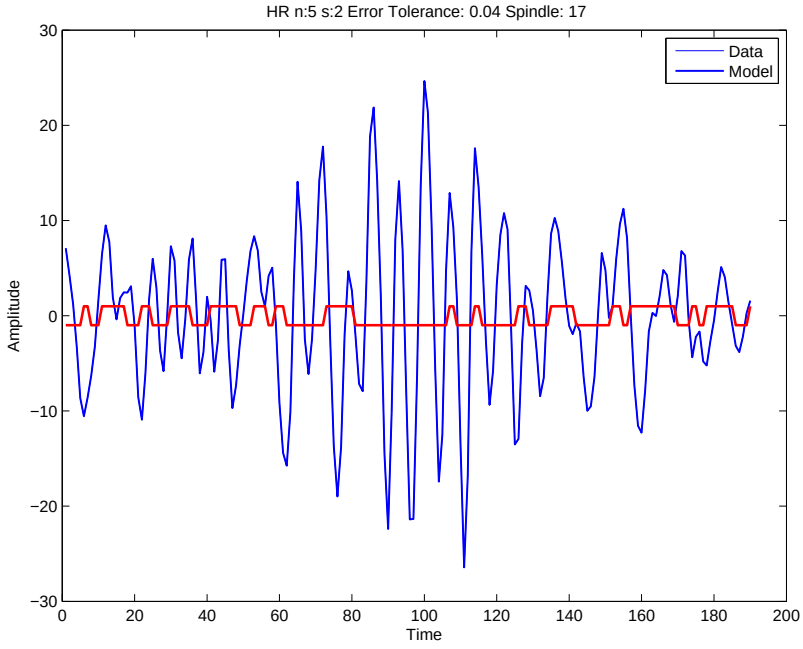

Figure 5.23: Spindle 17 Taken from an Annotated EEG Signal with Two Switched Models of Order 5 Found Through HR Overlaid on the Data.

Figures 5.22 and 5.23 show GPCA and HR run on spindle 17 with the model order set to 5. Compared to previous results shown in Figures 5.16 and 5.17, a model order of 5 was more descriptive and had better separation between the data. Both GPCA and HR showed clear separation between spindle and non-spindle data. Lastly, Figure 5.24 shows the GPCA algorithm run with a model order of 6 on spindle 6. We can see there is far better separation with a model order of 6 than 4. No HR model was generated for this due to a lack of computational resources.

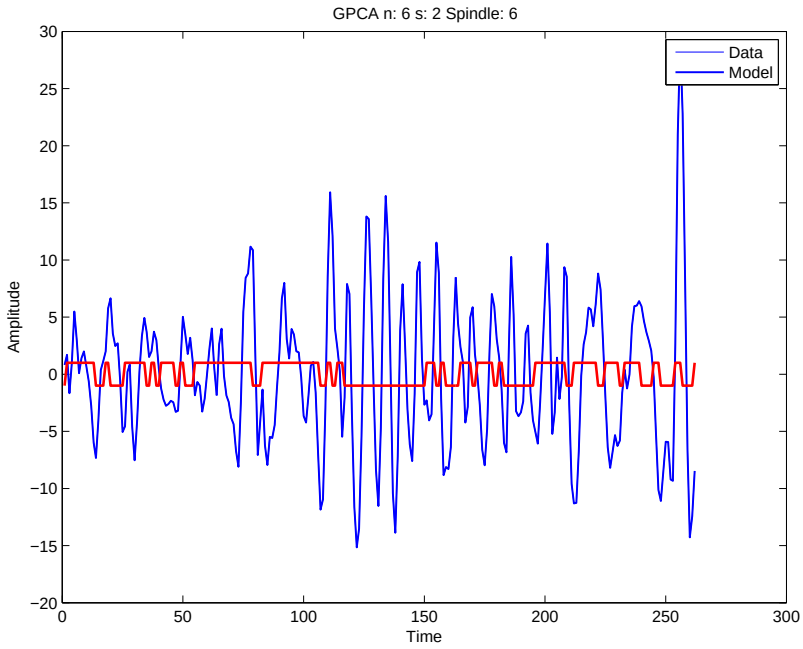

Figure 5.24: Spindle 6 Taken from an Annotated EEG Signal with Two Switched Models of Order 6 Overlaid on the Data.

5.2.2 Spindle with Tail Error

Below, the same candidates shown in the previous section are used to illustrate the affect of window size on separation. Results are shown in which the spindle is present in the first half of the window. We hypothesize that there will be less separation due to there being less variance in the data provided. Additionally, we show results from GPCA and HR run only on model order of 4 to save space and give a condensed summary of the affect of window size.

Figures 5.25 and 5.26 show the results obtained from applying GPCA and HR respectively. Both obtained relatively poor results due to the spindle being poorly differentiated. Perhaps in this case, due to the lack of varying information, the model order could have been decreased in order to force representation of the data through less information.

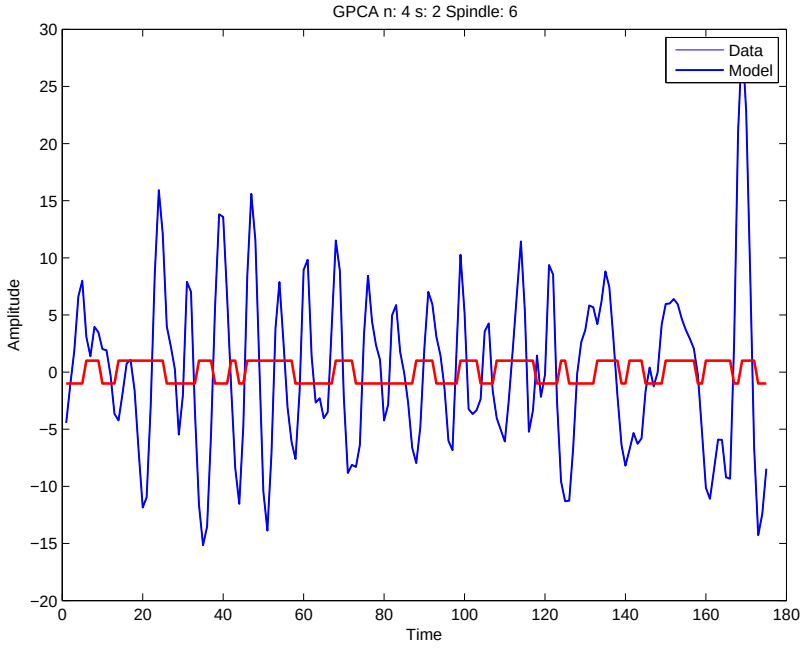

Figure 5.25: Spindle 6 Taken from an Annotated EEG Signal with Two Switched Models of Order 4 Overlaid on the Data.

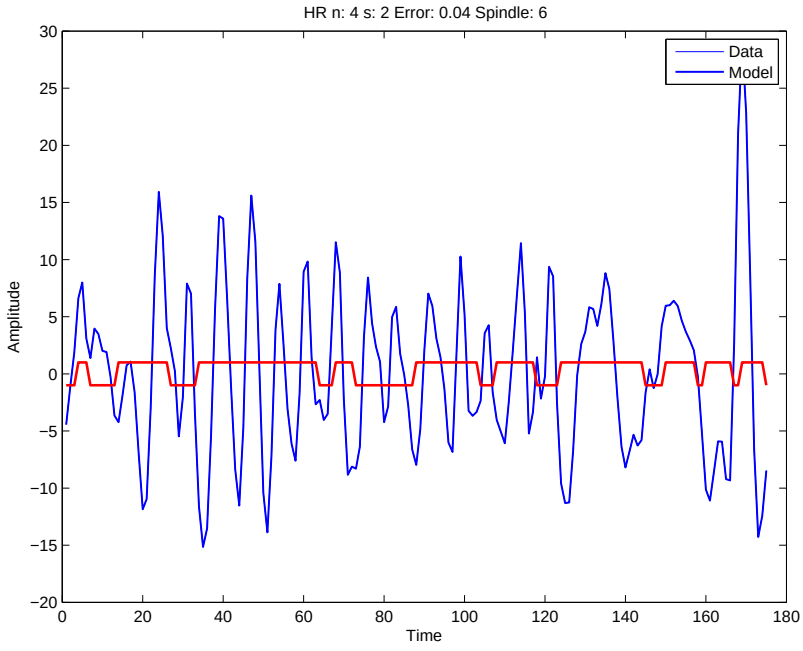

Figure 5.26: Spindle 6 Taken from an Annotated EEG Signal with Two Switched Models of Order 4 Found Through HR Overlaid on the Data.

As can be seen below in Figures 5.27 and 5.28, both the GPCA and HR algorithm obtained very similar results when applied to spindle 13. Although, there were certain areas it classified correctly, the majority of the information was not separated accurately. It is difficult to say what this may be attributed to.

The models in Figures 5.29 and 5.30 appear to be near identical and slightly better at separating the two sets of information in spindle 14. However, it seems like the spindle information is almost equally described by both underlying models, with a slight emphasis towards one over the other.

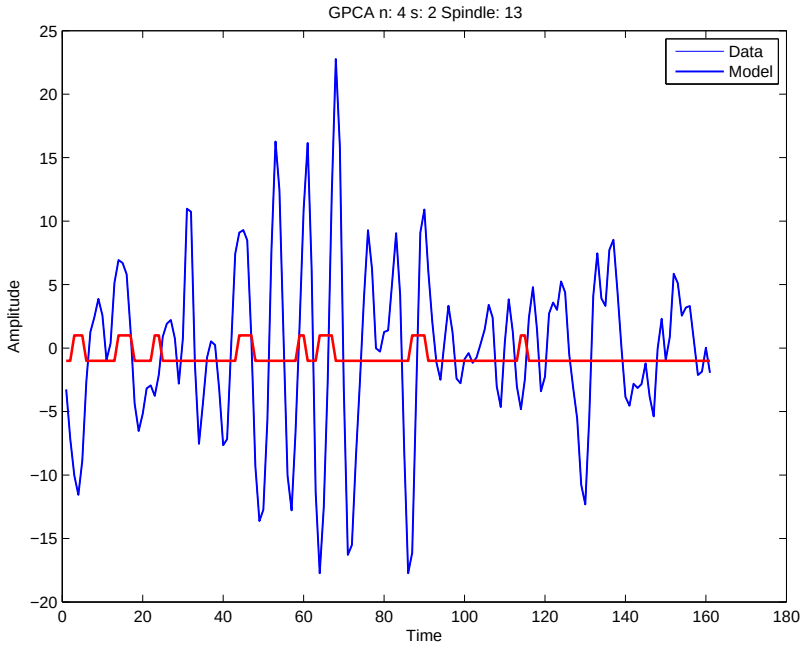

Figure 5.27: Spindle 13 Taken from an Annotated EEG Signal with Two Switched Models of Order 4 Overlaid on the Data.

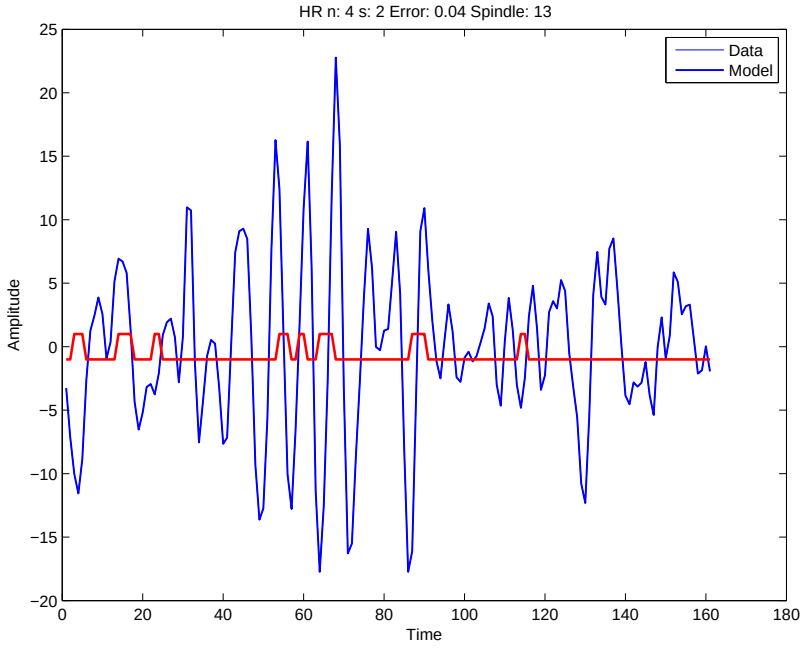

Figure 5.28: Spindle 13 Taken from an Annotated EEG Signal with Two Switched Models of Order 4 Found Through HR Overlaid on the Data.

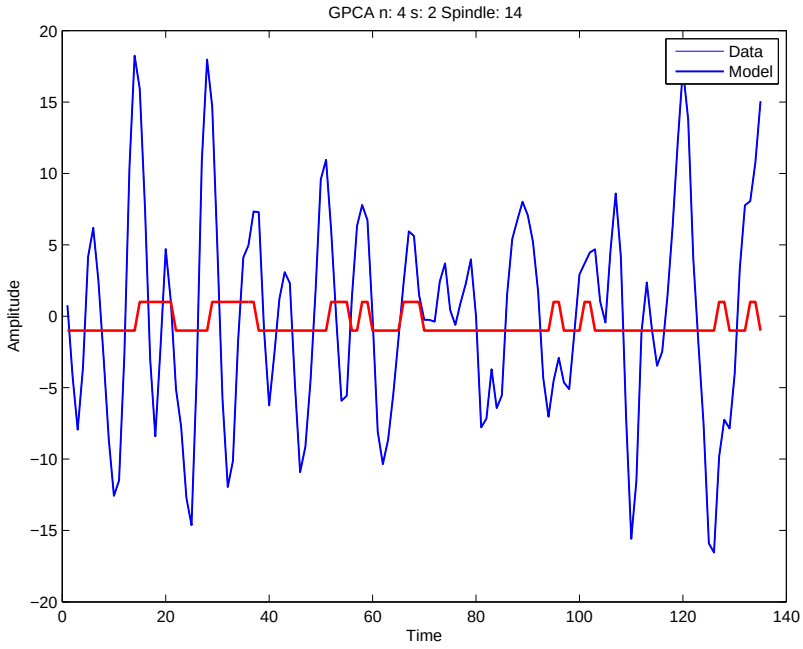

Figure 5.29: Spindle 14 Taken from an Annotated EEG Signal with Two Switched Models of Order 4 Overlaid on the Data.

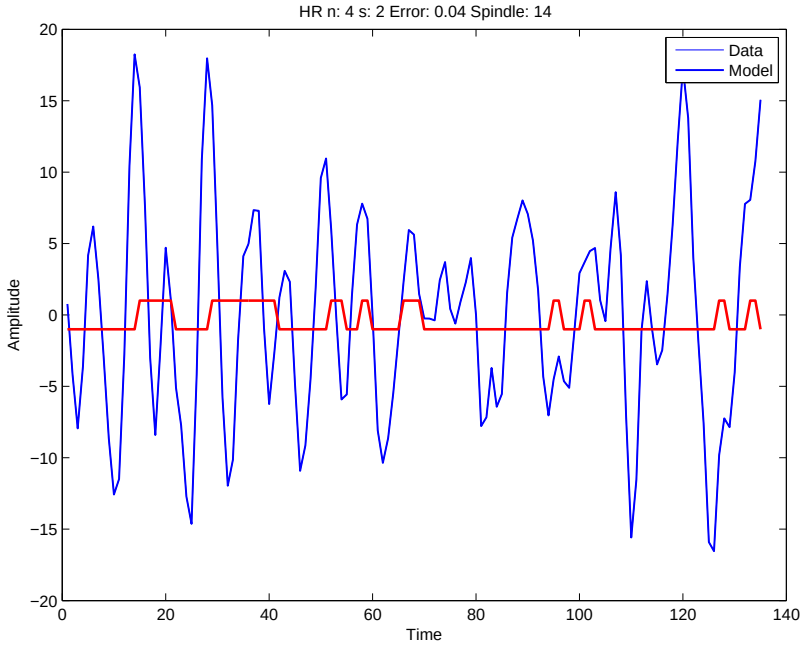

Figure 5.30: Spindle 14 Taken from an Annotated EEG Signal with Two Switched Models of Order 4 Overlaid on the Data.

The next four figures from Figure 5.31 to 5.34 show a similar pattern where the HR algorithm seems to do a poor job at separating. However, in the case of test 17, the GPCA model is able to differentiate the spindle information decently, while that is not the case for test 16. This may be due to there being an amplitude difference between the spindle and non-spindle information that can be observed in test 17. As shown in our synthetic test, an amplitude difference seems to correlate with good separation, even when measurement error is ignored.

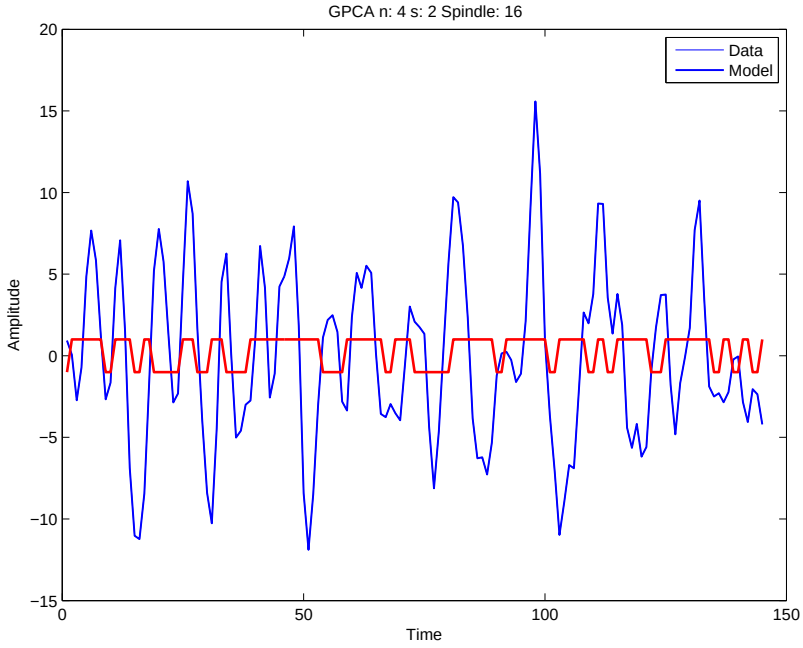

Figure 5.31: Spindle 16 Taken from an Annotated EEG Signal with Two Switched Models of Order 4 Overlaid on the Data.

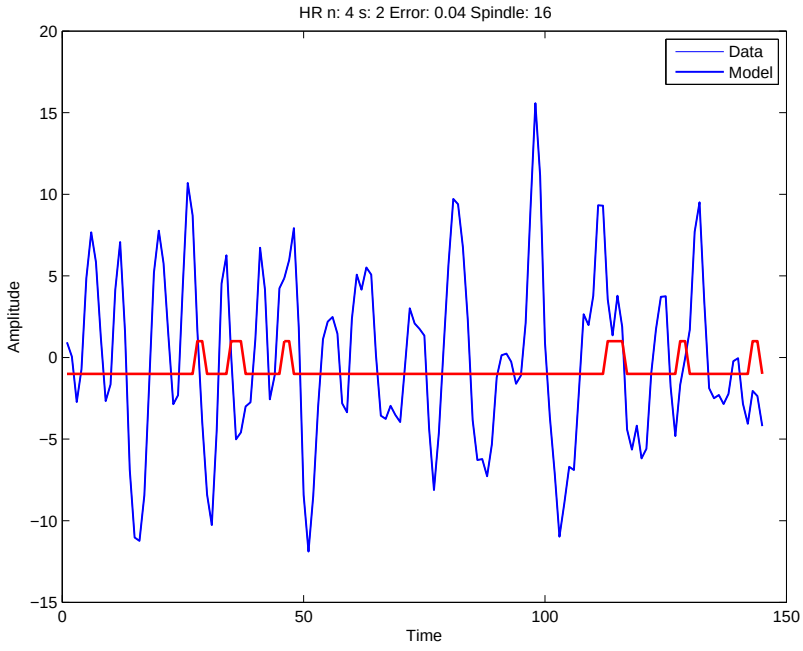

Figure 5.32: Spindle 16 Taken from an Annotated EEG Signal with Two Switched Models of Order 4 Found Through HR Overlaid on the Data.

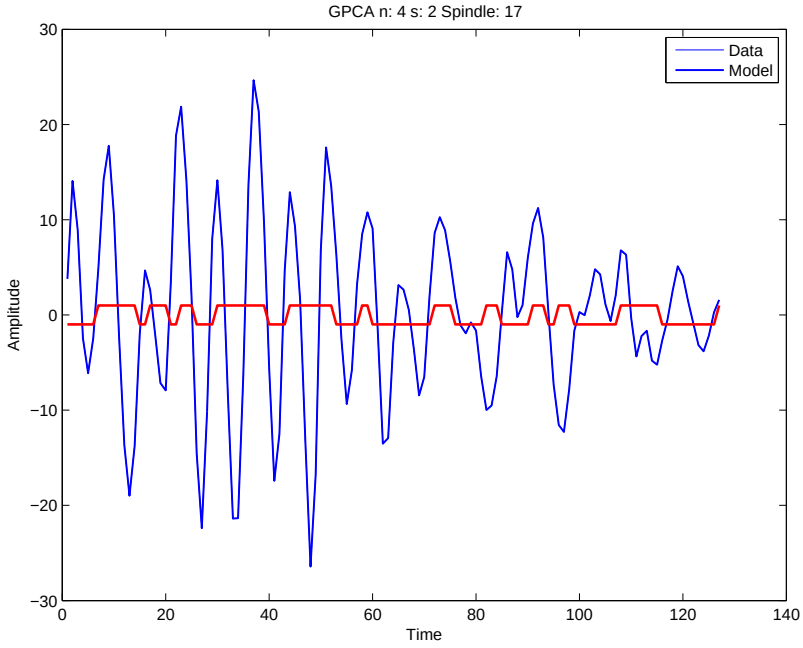

Figure 5.33: Spindle 17 Taken from an Annotated EEG Signal with Two Switched Models of Order 4 Overlaid on the Data.

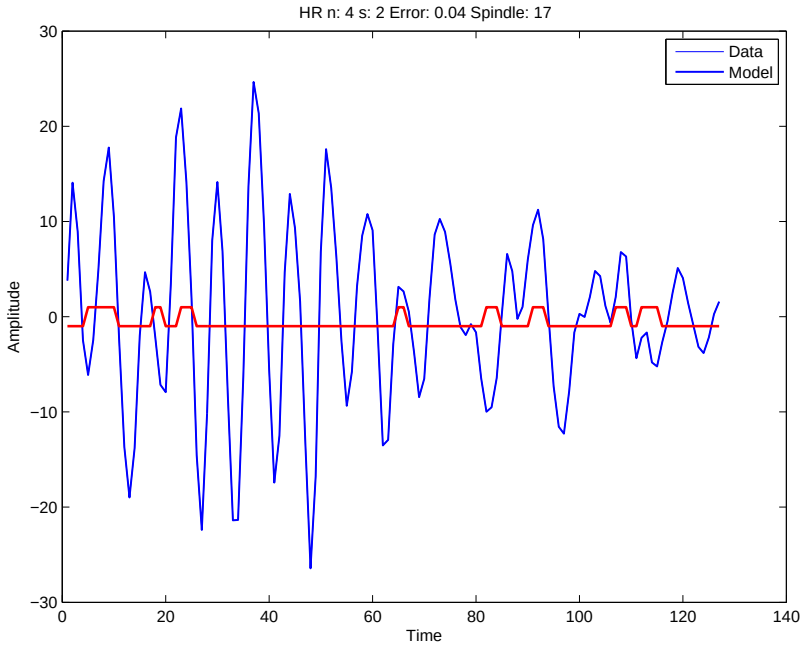

Figure 5.34: Spindle 17 Taken from an Annotated EEG Signal with Two Switched Models of Order 4 Found Through HR Overlaid on the Data.

This concludes the tests done on model order. It can be summarized that model order has a crucial affect on the separation of data. If the model order is too high, all information can be fit into one model. Where as if the model order is too low, there seems to be significant switching between the two models. Therefore an approach where the choice of model order is implicit in the solution of the problem such that it maximizes separability between the two models would be ideal.

5.3 Error Tolerance on HR

In this section we analyze the affect of error tolerance on the respective switched model system obtained using HR. We use a window size three times the size of the spindle, with the spindle (information) located at the center. Additionally, we apply an error tolerance from 2% to 10% in increments of two. The GPCA algorithm will be used as a benchmark, shown in Figure 5.35. Although there are jumps between the two systems located at non-spindle locations, it is easy to see that the majority of the spindle is represented by model two where as the non-spindle information is represented by model one.

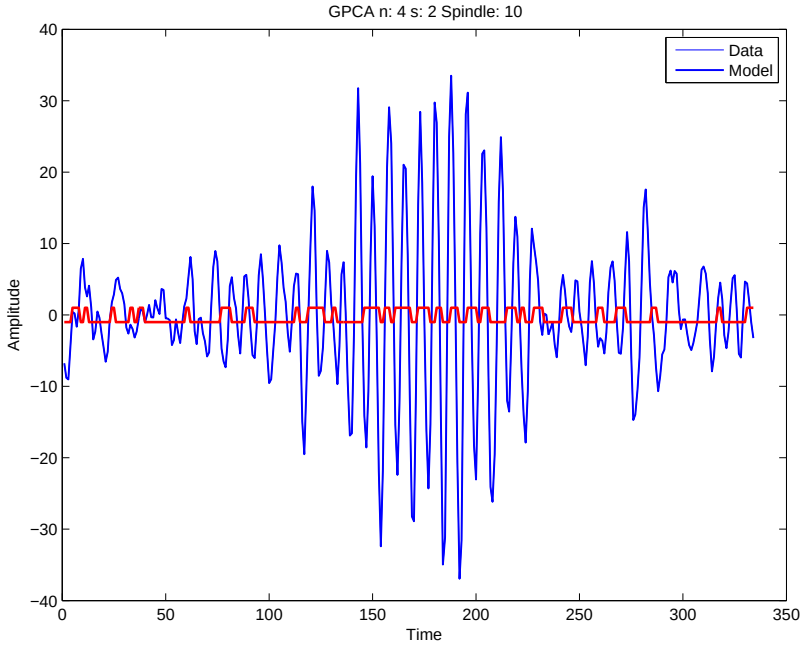

Figure 5.35: Spindle 10 Taken from an Annotated EEG Signal with Two Switched Models of Order 4 Overlaid on the Data using GPCA.

We begin by applying the HR algorithm with a 2% error tolerance. The result obtained can be viewed in Figure 5.36. It can be observed visually that there does not seem to be any change. This implies the error tolerance may be too low. Figure 5.37 shows an error tolerance of 4% applied to the data. Here, there are visible changes that begin to arise such as longer duration of continuous modeling and less occurring of spikes. However, as we increase the error percentage to 6%, there seems to be a complete disconnect and the models do a very poor job separating the information. This may be due to a local minimum solution which was picked due to the objective gap between step 1 and step 2 of the HR algorithm being close enough. However, as better tools are developed

for solving polynomial optimization problems via moments, a relaxation order high enough can be chosen which would guarantee optimality.

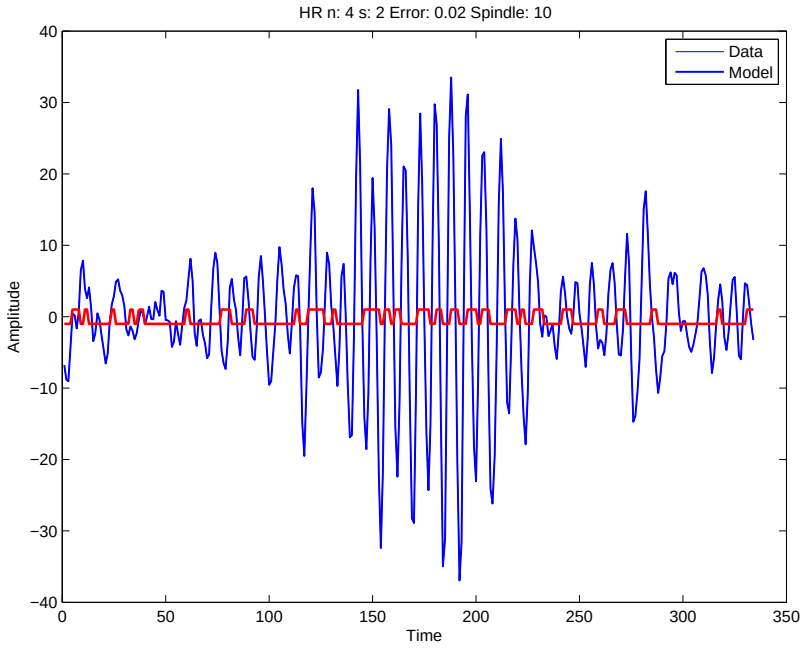

Figure 5.36: Spindle 10 Taken from an Annotated EEG Signal with Two Switched Models of Order 4 Overlaid on the Data using HR with Error Tolerance of 4%.

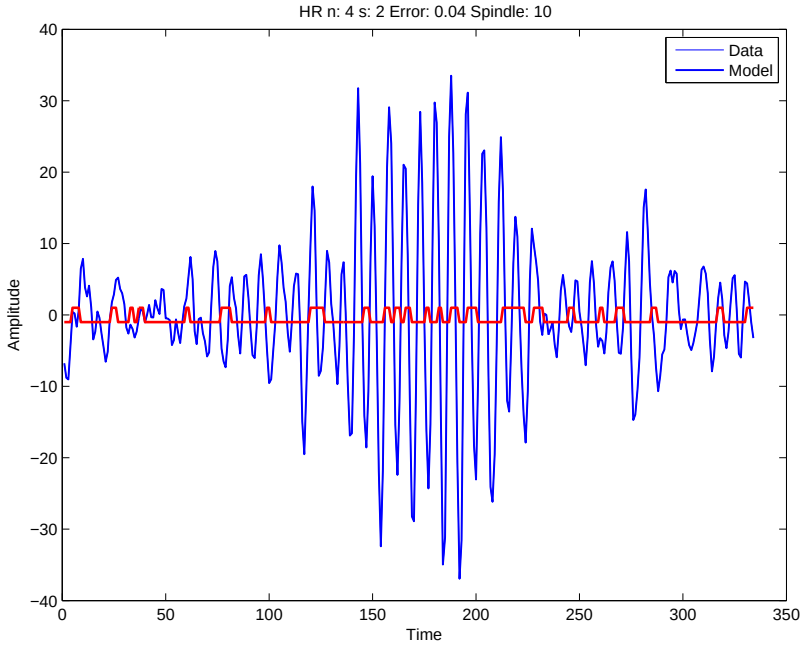

Figure 5.37: Spindle 17 Taken from an Annotated EEG Signal with Two Switched Models of Order 5 Overlaid on the Data.

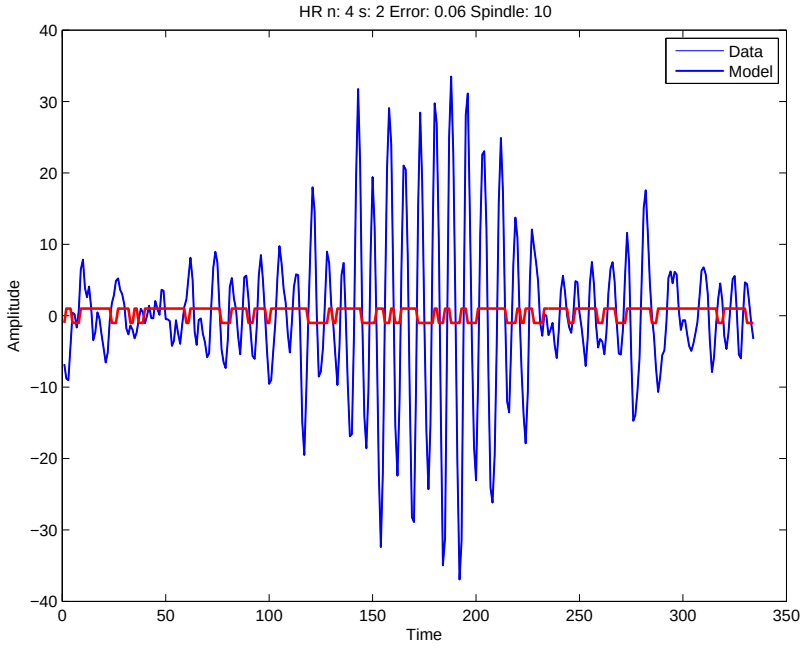

Figure 5.38: Spindle 17 Taken from an Annotated EEG Signal with Two Switched Models of Order 5 Overlaid on the Data.

Lastly, Figures 5.39 and 5.40 show results with error tolerance of 8% and 10% respectively. An error tolerance of 8% seems to produce similar results to the one found in 4%. Figure 5.40 has the model identifying the spindle inverted from the previous examples and shows good results in separating the spindle from non-spindle information.

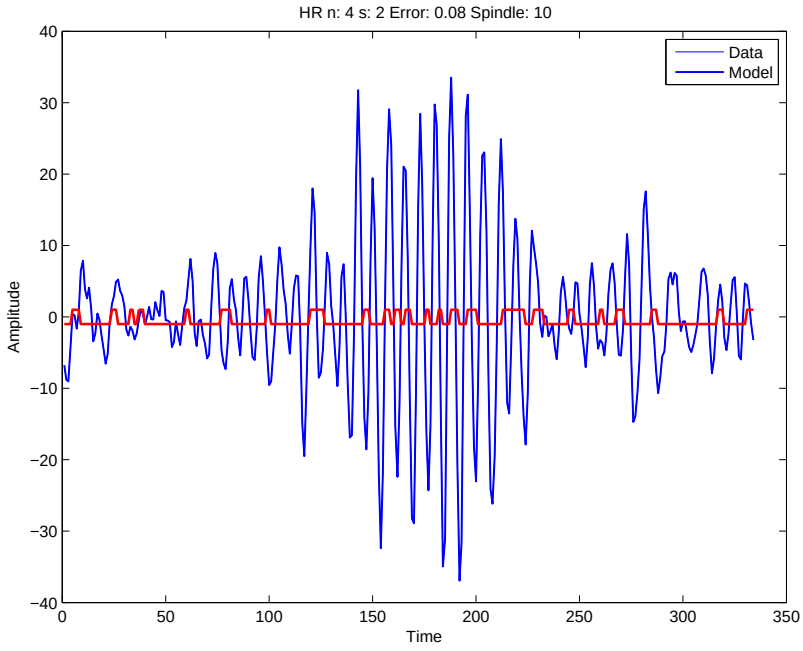

Figure 5.39: Spindle 17 Taken from an Annotated EEG Signal with Two Switched Models of Order 5 Overlaid on the Data.

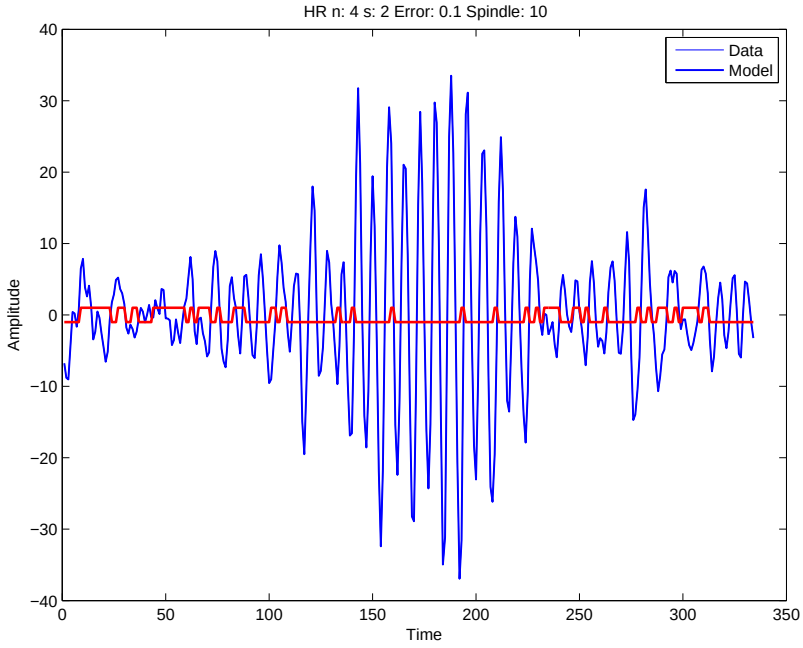

Figure 5.40: Spindle 17 Taken from an Annotated EEG Signal with Two Switched Models of Order 5 Overlaid on the Data.

In conclusion, it is difficult to obtain a valid comparison as the solutions are local optima and therefore as you increase the error tolerance, previously found local optima can still be chosen. It is evident there exists a correlation between increasing error tolerance and model representation. As the error tolerance increases, one of the two models starts to dominate the classification of the data.

5.4 Gloptipoly Vs. SparsePOP

We show a comparison between two tools used to solve polynomial optimization problems. We used both Gloptipoly3 and sparsePOP300 to

solve step one of the HR algorithm. A comparison between the problem formulation time and solving time is shown below. The test was done by creating data corresponding to AR models of different order and size. Table 5.4 below shows the number of models, model order, and relaxation order that were used for both problems.

Table 5.1: Test Parameters used to Compare Speed.

s	n	L	Relaxation Order
2	2	11	2
2	2	21	2
2	3	11	2
2	3	16	2
2	3	21	2
2	3	46	2
2	4	11	2
2	4	14	2
2	4	44	2

The test results seen in Table 5.4 clearly show the significant difference between GloptiPoly3 and SparsePOP. As the number of data samples increases even slightly, there is a drastic affect on the size of the moments matrices generated. Therefore, these tests were run on a windows cluster with 128GB of RAM and an AMD Opteron Processor 6376. Through observations of the solving time it can be seen that GloptiPoly3 would not be feasible to use for the size of problems that were solved in this thesis. However, SparsePOP was able to solve the same problems in a fraction of the time taken by GloptiPoly3 as can be seen in the table below.

Table 5.2: Test Results From GloptiPoly3 and SparsePOP.

s	n	L	Create (s) Glopt	Solve (s) Glopt	Create (s) SPOP	Solve (s) SPOP
2	2	11	0.3552	22.7147	0.4462	7.9244
2	3	11	0.3523	24.6316	0.6573	12.1938
2	4	11	0.2520	22.5853	0.9103	20.9707
2	4	14	0.6290	293.3383	1.7405	29.5204
2	3	16	0.2154	617.2756	0.3127	6.8934
2	2	21	2.5091	1.9561e+04	0.8421	17.4825
2	3	21	2.7724	1.8794e+04	1.3923	28.5570
2	3	46	N/A	N/A	1.0684	24.6015
2	4	44	N/A	N/A	1.6010	41.3001

Conclusions

5.5 Concluding Remarks

In this thesis, the difficult problem of hybrid system identification was applied to a real world ill-defined problem of spindle detection in EEG signals. We attempted to capture spindle information through switching of a linear hybrid dynamical systems identified from EEG noisy recordings.

The Hybrid identification problem of affine linear models from noisy measurements was formulated as a polynomial optimization problem via the hybrid decoupling constraint. Then an overview of polynomial optimization was presented to show how these problems can be solved as a sequence of convex problems, through convex LMI relaxations. These problem belong to the class of semidefinite programs for which efficient numerical solvers exist.

From the solution to the semidefinite program the individual model parameters can be extracted through the minimum singular value of a so-called Veronese matrix.

In the results section we observed that for well formed spindle data the model switching can clearly distinguish between spindle and non-spindle data. However, when real data was taken, there was not much variance between spindle and non-spindle data. Therefore, picking the right model order to illustrate accurate separation was difficult and had to be known *a priori*. A similar statement can be made for the error tolerance. Based on the error associated with collecting the data, the error tolerance would have to be known in order to obtain meaningful results in an unsupervised manner. As this was an initial approach, there are many changes that can be made to both the algorithms used as well as the method of detection to obtain better results. It is important to note that the area of hybrid system identification is new and as these techniques get better, they will allow us to obtain much better results on various applications.

5.6 Future Work

While this thesis demonstrates a new approach to doing spindle modeling there are many future directions that can be explored. Here we give a list of opportunities for future research.

- A method for finding switched dynamical systems with an implicit model order that maximizes the separation between both models

would be a significant breakthrough not only for this problem but for improved hybrid system identification.

- Additionally due to experiencing many difficulties in model parameter separation, a method of finding switched dynamical systems where the model parameters are not coupled would be another major breakthrough, as no information would be lost recovering the model parameters and the switched models would be known exactly.

- As observed in this thesis, there were many cases where despite a single model representing the majority of a spindle, there was still significant amount of switching that occurred. An additional constraint can be included to penalize switches and reward staying on one model for a longer duration.

- If the problem of spindle detection is pursued with these methods, an additional constraint should be placed on the duration a model is held. This is important in sleep staging because rapid oscillatory movement in not considered a spindle unless it lasts longer than 0.5 seconds.

Bibliography

[1] G. Chesi, "LMI techniques for optimization over polynomials in control: a survey," *Automatic Control, IEEE Transactions on*, vol. 55, no. 11, pp. 2500–2510, 2010.

[2] J. B. Lasserre, "Global optimization with polynomials and the problem of moments," *SIAM Journal on Optimization*, vol. 11, no. 3, pp. 796–817, 2001.

[3] D. Henrion and J.-B. Lasserre, "Convergent relaxations of polynomial matrix inequalities and static output feedback," *Automatic Control, IEEE Transactions on*, vol. 51, no. 2, pp. 192–202, 2006.

[4] H. Landau and A. M. Society, *Moments in Mathematics*. AMS short course lecture notes, American Mathematical Society, 1987.

[5] R. E. Curto and L. A. Fialkow, *Solution of the truncated complex moment problem for flat data*, vol. 568. American Mathematical Soc., 1996.

[6] D. Henrion, "Optimization on linear matrix inequalities for polynomial systems control," *arXiv preprint arXiv:1309.3112*, 2013.

[7] M. Grant and S. Boyd, "CVX: Matlab software for disciplined convex programming, version 2.1." http://cvxr.com/cvx, Mar. 2014.

[8] D. Henrion, J. B. Lasserre, and J. Loefberg, "Gloptipoly 3: moments, optimization and semidefinite programming," *Optimization Methods and Software*, vol. 24, no. 4–5, pp. 761–779, 2009.

[9] H. Waki, S. Kim, M. Kojima, M. Muramatsu, and H. Sugimoto, "Algorithm 883: Sparsepop—a sparse semidefinite programming relaxation of polynomial optimization problems," *ACM Trans. Math. Softw.*, vol. 35, pp. 15:1–15:13, July 2008.

[10] L. Ljung, *System identification*. Springer, 1998.

[11] R. Vidal, S. Soatto, Y. Ma, and S. Sastry, "An algebraic geometric approach to the identification of a class of linear hybrid systems," in *Decision and Control, 2003. Proceedings. 42nd IEEE Conference on*, vol. 1, pp. 167–172, IEEE, 2003.

[12] Y. Ma and R. Vidal, "Identification of deterministic switched ARX systems via identification of algebraic varieties," in *HSCC 2005* (M. Morari and L. Thiele, eds.), vol. 3414 of *LNCS*, pp. 449–465, Springer, 2005.

[13] N. Ozay, M. Sznaier, C. Lagoa, and O. Camps, "A sparsification

approach to set membership identification of switched affine systems," *Automatic Control, IEEE Transactions on*, vol. 57, pp. 634–648, March 2012.

[14] C. Feng, C. M. Lagoa, and M. Sznaier, "Hybrid system identification via sparse polynomial optimization," in *American Control Conference (ACC), 2010*, pp. 160–165, IEEE, 2010.

[15] N. Ozay, C. Lagoa, and M. Sznaier, "Robust identification of switched affine systems via moments-based convex optimization," in *Decision and Control, 2009 held jointly with the 2009 28th Chinese Control Conference. CDC/CCC 2009. Proceedings of the 48th IEEE Conference on*, pp. 4686–4691, Dec 2009.

[16] R. Vidal, *Generalized Principal Component Analysis (GPCA): an Algebraic Geometric Approach to Subspace Clustering and Motion Segmentation*. PhD thesis, University of California at Berkley, 2003.

[17] N. Ozay, C. Lagoa, and M. Sznaier, "Set membership identification of switched linear systems with known number of subsystems," *Automatica*, vol. 51, pp. 180–191, 2015.

[18] C. Feng, C. M. Lagoa, N. Ozay, and M. Sznaier, "Hybrid system identification: An SDP approach," in *Decision and Control (CDC), 2010 49th IEEE Conference on*, pp. 1546–1552, IEEE, 2010.

[19] M. Fazel, "Matrix rank minimization with applications," *Elec. Eng. Dept., Stanford University*, vol. 54, pp. 1–130, 2002.

[20] S. Sanei and J. A. Chambers, *EEG signal processing*. John Wiley & Sons, 2013.

[21] A. Rechtschaffen and A. Kales, "A manual of standardized terminology, techniques and scoring system for sleep stages of human subjects," *US Government Printing Office, US Public Health Service*, 1968.

[22] S. Devuyst, T. Dutoit, and M. Kerkofs, "The dreams project." TCTS Laboratory and UniversitÃl' Libre de Bruxelles - CHU de Charleroi Sleep Laboratory, 2003-2008. Conducted by University of MONS.

[23] B. Shoelson, "edfread." Matlab Central, June 2013. http://www.mathworks.com/matlabcentral/fileexchange/31900-edfread.

[24] E. Fehrmann, "Automated sleep classification using the new sleep stage standards," Master's thesis, Rochester Institute of Technology, 2013.