CHAPTER 4

TIME-VARYING SURROGATE DATA TO ASSESS NONLINEARITY

4.1 INTRODUCTION

The method of surrogate data belongs to the family of statistical tests known as hypothesis testing, which nowadays is the most popular test used in nonlinear time series analysis to investigate the existence of nonlinear dynamics underlying experimental data. The method of surrogate data applied on a single time series is based on: 1) establishing a statistical hypothesis (the null hypothesis) against which observations are tested; 2) generating a set of time series (the surrogate series) that, in accordance with the null hypothesis, shares given properties of the original series but lacks the property that is under investigation; 3) calculating a certain index (the discriminating statistic) from the original series and from each surrogate series; and 4) performing a statistical test that, if the discriminating statistic computed for the original series results significantly different from the values obtained for the surrogate set, allows to reject the null hypothesis with some predetermined confidence level.

Two main approaches have been identified to generate surrogate series according to this null hypothesis (Theiler and Prichard 1996): one is to fit a linear model to the original series and then feed the estimated model by independent noise realizations, generating “typical realizations” of the linear stochastic process; the second is to take a Fourier Transform (FT) of the data,
randomize the phases, and then invert the FT to generate ‘constrained realizations’ of the process (Theiler and Prichard 1996). Both these approaches generate surrogate series that have the same linear correction function of the original series, but are otherwise random as specified by the null hypothesis.

### 4.1.1 Time Invariant Autoregressive (TIV-AR) Surrogates

In its classical formulation, the method of surrogate data investigates the presence of nonlinear dynamics assuming that the observed time series is stationary (Theiler et al. 1992). Due to this assumption, the surrogate series generated by the method is time invariant (TIV), i.e., it has statistical properties that are invariant to translation of the time origin. With regard to the method of surrogate data, ambiguities between nonstationarity and nonlinearity might arise when the null hypothesis of having a TIV linear process is rejected by using a nonlinear discriminating statistic.

Given the time series $x(n), n=0,1,\ldots,N$, TIV – AR surrogates were generated by performing AR prediction of the series

$$x(n) = a(0) + \sum_{i=1}^{P} a(i)x(n - i) + e(n) \quad (4.1)$$

where $e(n)$ is the prediction error and the coefficients $a(0)$ and $a(1), \ldots, a(P)$ describe, respectively, constant and linear contributions to the dynamics of $x$. The AR model of (4.1) was identified by the least squares method, yielding estimates of the model coefficients, $\hat{a}(0), \hat{a}(1), \ldots, \hat{a}(P)$ and of the residuals $\hat{e}(0), \hat{e}(1), \ldots, \hat{e}(N)$ as well. A surrogate of the series $x$ was then generated according to the typical residual-based bootstrap approach (Politis 2003), i.e., by resampling the residuals and feeding into the estimated AR model.
\[ x_{sTV}(n) = \hat{a}(0) + \sum_{i=1}^p \hat{a}(i) + x_{sTV}(n - i) + \epsilon(rnd(N)) \quad (4.2) \]

where \( rnd(N) \) is a random number drawn from the discrete uniform distribution with maximum equal to \( N \).

### 4.1.2 Nonparametric Surrogates

Contrary to AR surrogates, nonparametric surrogates do not assume a linear model to reproduce the linear autocorrelation present in the original series, but operate in the frequency domain according to a constrained realization approach. In this study, we considered the two types of constrained surrogates used in nonlinear analysis studies (Faes et al. 2009): 1) the Fourier transform (FT) surrogates (Theiler et al. 1992), generated by computing the FT of the series \( x \), substituting the Fourier phases with random numbers uniformly distributed between 0 and \( 2\pi \), and finally, performing the inverse FT; 2) the iteratively refined amplitude – adjusted FT (IAAFT) surrogates (Schreiber and Schmitz 1996), generated by using an iterative procedure that alternatively constrains the surrogate series to have the same power spectrum (by replacing the squared Fourier amplitudes of the candidate surrogate series with those of the original series) and to have the same amplitude distribution (by a rank ordering procedure) of the original series.

### 4.2 Ziggurat Method for Random Number Generation

#### 4.2.1 Normal distribution

Simulations requiring Gaussian random numbers are critical in fields including communications, financial modeling, and many others. A wide range of Gaussian random number generators (GRNGs) have been described in the literature (Thomas et al. 2007). They all utilize well-
understood basic mathematical principles, usually involving transformations of uniform random numbers. The requirement to generate extremely large numbers of Gaussian random numbers elevates the importance of the quality of the GRNG. For example, while Gaussian random numbers with absolute values greater than $6\sigma$ or $7\sigma$ rarely occur; it is precisely those extreme events that could contribute disproportionately to certain rare but important system behaviors that the simulation aims to explore. Samples from an ideal GRNG with absolute value exceeding $9\sigma$ occur with probability $2.26 \times 10^{-19}$. For $10\sigma$, the corresponding probability is $1.52 \times 10^{-23}$.

GRNGs classify into four basic categories: cumulative density function (CDF) inversion, transformation, rejection and recursive methods. The CDF inversion method simply inverts the CDF to produce a random number from a desired distribution. Transformation methods involve the direct transformation of uniform random numbers to a Gaussian distribution. The third category, rejection, again starts with uniform random numbers and a transformation, but has the additional step of conditionally rejecting some of the transformed values. Recursion, the final category, utilizes linear combinations of previously generated Gaussian numbers to produce new outputs.

There are seven different algorithms are used to generate Gaussian random numbers in the category of rejection method.

1. Polar algorithm
2. Marsaglia – Bray algorithm
3. Ratio of uniforms algorithm
4. Leva’s Ratio-of-uniform algorithm
5. Ahrens – Dieter table free algorithm
6. Gaussian random distribution (GRAND) algorithm
7. Ziggurat algorithm
This section describes the implementation of a high performance Gaussian Random number generation using ziggurat method from a given decreasing density. The ziggurat algorithm which divides the area under the probability density functions into three regions: rectangular, wedge and tail. The rejection method is then used and this amounts to determining whether a random point falls into one of the three regions. It uses two tables’ integer $k_i$ and real $w_i$. Some 99% of the time, the required $x$ is produced by: Generate a random 32bit integer $j$ and let $i$ be the index formed from the rightmost 8bits of $j$. If $j < k_i$ return $x = j \times w_i$. We illustrate with Matlab code version 7 that provides separate function for generation of Gaussian variables with a short procedure for setting up the necessary tables.

4.2.2 The Ziggurat Method

The Ziggurat method uses the rejection method to generate a random variate from an arbitrary decreasing probability density function. Our description of the ziggurat method in this section follows the notation of Marsaglia (Marsaglia and Tsang 2000).

The rejection method for generating a random variate can be described as follows. Let $C$ be the set of points $(x, y)$ under a plot of the curve $y = f(x)$ with finite area. Let $Z$ be a set of points containing $C$: $Z \ni C$. The basic idea of the rejection method is: Choose random points $(x, y)$ uniformly from $Z$ until you get one that falls in $C$, then return to $x$ as the required variate. The density of such an $x$ will be $c \times f(x)$, with $c$ as the normalizing constant that makes $c \times f(x)$ a proper density. Three of the main criteria for choosing the covering set $Z$ are as follows: 1) Make it easy and fast to select a random point $(x, y)$ from $Z$, 2) Make it easy and fast to decide whether the random point $(x, y)$ from $Z$ also falls in $C$, and 3) Make $\frac{\text{area}(C)}{\text{area}(Z)}$, the efficiency of the rejection procedure is close to 1.
The probability density function or pdf of the normal distribution is the bell shaped curve,

\[ f(x) = ce^{x^2/2} \]  \hspace{1cm} (4.3)

where \( c = 1/\sqrt{2\pi} \) is a normalizing constant that we can ignore. If we generate random points \((x, y)\) uniformly distributed in the plane, and reject any of them that do not fall under this curve, the remaining \(x's\) form our desired normal distribution. The ziggurat algorithm covers the area under the pdf by a slightly larger area with \(n\) sections. Figure 4.1 has \(n=8\); actual code might use \(n=128\). The top \(n-1\) sections are rectangles. The bottom section is a rectangle together with an infinite tail under the graph of \(f(x)\). The rectangles and bottom strip are chosen so that they are all of equal area \(v\), and their right-hand edge is denoted \(x_i\). The leftmost rectangle \(R_0\) is assumed to be empty and \(x_0 = 0\).

Figure 4.1 The ziggurat method with 7 rectangles and a bottom, base strip
The following pseudo code describes the complete ziggurat algorithm (which shows the rectangle, wedge and tail regions):

1. INITIALIZATION
2. n is the size of the w and k tables
3. \( i=0 \ldots n-1 \) and \( r = x_{n-1} \)
4. for \( i=0 \) \( w_0 = 0.5^{32} \ v/f(r) \), \( k_0 = [2^{32} \ (rf(r)/v)] \)
5. for, \( i=1,2,3 \ldots n-1 \) \( w_i = 0.5^{32} \ x_i \), \( k_i = [2^{32} \ (x_{i-1}/x_i)] \)
6. \( \{f_i\} \), where \( f_i = \exp(-r^2/2) \)
7. U – uniform generation from (0,1)
8. REPEAT
9. Generate a unsigned random 32bit integer j
10. Set index \( i = \text{bitand}(j,127) \) and \( x = j \times w_i \)
11. IF \( \text{abs}(j) < k_i \) THEN return\((x) /\ast \text{Rectangle} /\ast /\)
12. IF \( i=0 \) THEN /* Tail */
13. DO
14. \( x = \log(U) \times (1/r); \ y = \log(U); \)
15. \( \text{While} \ ((y+y) > (x^*x)) \)
16. \( \text{end} \)
17. \( \text{if} \( x > 0 \) \)
18. \( \text{rexp} = r+x; \)
19. \( \text{else} \)
20. \( \text{rexp} = -(r+x); \)
21. \( \text{end} \)
22. Return \((\text{rexp}) \)
23. IF \( (f_i+(f_{i-1} - f_i)U < \exp(-x^2/2)) /\ast \text{Wedge} /\ast /\)
24. Return\((x) \)
25. Until false.
For our fastest application, we form the ziggurat with layers of 127 rectangles with common area \( v \), and a bottom strip of area \( v \). The rectangles end at \( x_1 < x_2 < x_3 < x_4 \ldots < x_{127} = r \). Assume those \( x \)'s have been determined, as described in the next section. Since, as in most applications, we provide uniform (0,1) variates \( U \) by floating a random integer (32bit unsigned long), we may save time by incorporating the float operation into the step that forms the \( x \)'s that are to be returned: For each value of the index \( i \) in \( 1 \leq i \leq 255 \), form the 32 bit integer \( k_i = \lfloor 2^{32} \left( x_{i-1} / x_i \right) \rfloor \) and set \( w_i = 0.5^{32} x_i \). For the special index \( i=0 \), set \( k_0 = \lfloor 2^{32} \left( r f(r) / v \right) \rfloor \) and \( w_0 = 0.5^{32} \ v / f(r) \).

### 4.2.3 Setting up the Ziggurat

Given the (scaled) target density \( f(x) \), \( x \geq 0 \), we want to find 127 equal – area rectangles, such as pictured in Figure 4.2, so that the covering set \( Z \) is very close to \( C \).

![Ziggurat Diagram](image.png)

**Figure 4.2** The ziggurat, showing every 10\(^{th}\) after the first 20 of 127 rectangles
A picture of a ziggurat with some of its 127 rectangles is in Figure 4.2, for the normal density. The rectangles are so closely layered that images blur in the lower part if all 127 are shown. So, after the first 20, only rectangles $R_{30}$, $R_{40}$, $R_{68}$, …… and the final $R_{127}$ are shown. Those few will give an idea of how closely Z covers C, and the closeness of $x_{i-1}/x_i$ to unity for almost all of the rectangles.

The right edges of the rectangles are at $0 = x_1 < x_2 < x_3 < x_4, \ldots < x_{127}$. From the figure, it is clear that we must have, using r to designate the rightmost end point $r = x_{127}$. Even if $v$ is given, it is not easy to find the necessary $x$’s. Instead, a feasible procedure is: define a function of $r$, the rightmost end point, say $z(r)$, by a sequence of Maple-like commands:

$$z(r) = x_{127} = r; v = rf(r) + \int_r^{x} f(x)dx$$

(4.4)

Zhang et al (2005) Proposed the random point is only accepted if it falls under the pdf curve, otherwise it is rejected. The probability of accepting a point $P_{\text{accept}}$ is given by (Leoang et al. 2003):

$$P_{\text{accept}} = \frac{\text{area}(c)}{\text{area}(z)} + \frac{\int e^{-x^2/2}dx}{2\sqrt{n}}$$

(4.5)

Finally, the probability that a point is not drawn from a rectangular region, $P_{\text{nrect}}$ (i.e it is a wedge, tail or rejected) can be calculated as follows:

$$P_{\text{nrect}} = 1 - \frac{\text{area}(\text{rect})}{\text{area}(Z)}$$

(4.6)

$$= \sum_{i=1}^{n-2} x_i \left( f(x_i) - f(x_{i+1}) \right) + x_{n-1} f(x_{n-1})$$

(4.7)
4.2.4 Algorithm Implementation

It turns out that storing a third table, \( f_i = f(x_i) \) is no more costly in terms of the final size of the complied program than is a more complicated version that computes the \( f(x_i) \)'s. Assuming that table, the essential part of the generating procedure in Matlab looks like this, assuming the variables have been declared, static tables \( k(128) \), \( w(128) \), and \( f(128) \) are setup, and sub functions are generated like shr3 and UNI. The result is a remarkably simple generating procedure, using a 32 bit integer generator such as shr3, described below:

For \( i = 0: \infty \)

\[
j = \text{shr3}; \ i = \text{bitand}(j, 127);
\]

\[
x = j \times w(i);
\]

if \( j < k(i) \) return \( x \);

if \( i == 0 \) return \( x \) from the tail

if ( \( \text{UNI} \times (f(i-1)-f(i)) < f(x) - f(i) \) ) return \( x \);

end

The infinite for loop is executed until one of the three returns provides the required \( x \) (better than 99% of the time from the first return). The method works for any decreasing density, conveniently scaled as \( f(x) \), for which a tail method can be provided. For symmetric densities, the right half is used and a random attached. The procedure differ only in that different \( f \)'s are used in third return; three tables \( k(128) \), \( w(128) \), and \( f(128) \) need to be constructed from that function and different tail methods need to be provided.

function \( [\text{jsrr,shr,uni,va}] = \text{shr3}(@sr) \);

\( jz = \text{uint32}(@sr); \)
jsr1=bitshift(jz,13); %Left shift
jsr2=jsr1/2^17; %right shift
jsr3=bitshift(jsr2,5);
jsrr =double(jsr3);
shr  = jz+jsr3;
v = double(shr);
uni  = 0.5+v*0.2328306e-9;
va  = bitand(shr,127);

shr3 uses an inline 3 (two left shift and one right shift) shift register, and UNI floats it to (0,1). It is very fast, has period $2^{32}-1$ and does very well in tests of randomness – in particular for sequences made from the last 8 bits.

The existence of nonlinear dynamics in the considered time series was investigated in accordance with the well-known method of surrogate data. This approach is based on a null hypothesis to be rejected, a surrogate dataset constructed in accordance with the null hypothesis, a discriminating statistic that has to be calculated on original and surrogate series, and a statistical test allowing it to reject (if it is the case) the null hypothesis.

The general null hypothesis assumed in nonlinearity tests is that the investigated time series is a realization of a linear stochastic process. However, a more specific null hypothesis is often required to be set, depending on the ability of the surrogates to preserve the statistical properties of the original series. Being conceived to test for nonlinearity, all the four algorithms considered here preserve, though with a different degree of approximation, the linear autocorrelation of the original series.
4.3 SURROGATE DATA METHOD

Generally, a surrogate data testing method involves three ingredients: 1) a null hypothesis; 2) a method to generate surrogate data; and 3) testing statistics for significant evaluation. The null hypothesis in the present study is that investigated data from a linear Gaussian process. If the null hypothesis is rejected, then we conclude that the data are either non-Gaussian or come from nonlinear process. The surrogate data are generated in such a way that it is Gaussian distributed but has the same second order spectral properties (in the bivariate, auto spectra) as the original data. The testing statistics shows the amplitude of the periodogram and the steps for generation of the surrogate data and the theoretical rationale behind the steps as follows (Xue Wang et al. 2007).

Consider two zero mean stationary random processes $x(t)$ and $y(t)$. These processes may or may not be linear Gaussian processes. Their one-sided auto spectra $S_{xx}$ and $S_{yy}$ can be estimated (Wu 2005);

$$S_{xx}(f) = 2 \lim_{T \to \infty} \frac{1}{T} E[|X_k(f,T)|^2]$$

$$S_{yy}(f) = 2 \lim_{T \to \infty} \frac{1}{T} E[|Y_k(f,T)|^2] \quad (4.8)$$

Where $T$ is the duration of the data, the expectation is taken over multiple realizations and $X_k$ and $Y_k$ are the Fourier Transform of $x(t)$ and $y(t)$. Now we consider how to generate two zero – mean linear Gaussian processes $x'(t)$ and $y'(t)$ which have the same second order statistical properties as $x(t)$ and $y(t)$. Let $X'_k$ and $Y'_k$ expressed in real and imaginary parts, $X'_R, X'_I, Y'_R$ and $Y'_I$ these Fourier Transforms are

$$X'_k(f) = X'_R(f) - jX'_I(f)$$
Since \( x'(t) \) and \( y'(t) \) are normally distributed with zero mean values, \( X'_R, X'_I, Y'_R \) and \( Y'_I \) are also normally distributed with zero means (Wu 2005). The covariance matrix \( (\Sigma) \) of \( X'_R, X'_I, Y'_R \) and \( Y'_I \) should be selected such that the auto-spectra and cross spectra of \( x'(t) \) and \( y'(t) \) are the same as those of the original data, i.e., \( S_{xx}' = S_{xx} \) and \( S_{yy}' = S_{yy} \).

In practice, \( S_{xx} \) and \( S_{yy} \) are estimated from the data. The question is how to draw Gaussian variables for \( X'_R, X'_I, Y'_R \) and \( Y'_I \) such that \( S_{xx}' = S_{xx} \) and \( S_{yy}' = S_{yy} \). It can be shown that the relation between \( X'_R, X'_I, Y'_R \) and \( Y'_I \) and \( S_{xx} \) and \( S_{yy} \) are as follows (Bendat and Persol 1986).

\[
E[X'_R X'_I] = E[Y'_R Y'_I] = 0
\]

\[
E[X'_R X'_R] = E[X'_I X'_I] = \left( \frac{T}{4} \right) S_{xx}'
\]

\[
E[Y'_R Y'_R] = E[Y'_I Y'_I] = \left( \frac{T}{4} \right) S_{yy}'
\]

(4.10)

According to these relations the covariance matrix \( (\Sigma) \) for the real and imaginary parts of \( X'_R \) and \( Y'_R \) for each frequency is

\[
\Sigma_x = E \left[ \begin{pmatrix} X'_R \\ X'_I \end{pmatrix} \begin{pmatrix} X'_R & X'_I \end{pmatrix} \right] = \frac{T}{4} \begin{pmatrix} S_{xx} & 0 \\ 0 & S_{xx} \end{pmatrix}
\]

\[
\Sigma_y = E \left[ \begin{pmatrix} Y'_R \\ Y'_I \end{pmatrix} \begin{pmatrix} Y'_R & Y'_I \end{pmatrix} \right] = \frac{T}{4} \begin{pmatrix} S_{yy} & 0 \\ 0 & S_{yy} \end{pmatrix}
\]

(4.11)

Choosing \( S_{xx}' = S_{xx} \) and \( S_{yy}' = S_{yy} \), and \( X'_R, X'_I, Y'_R \) and \( Y'_I \) are then generated by taking sample from a Gaussian distribution with zero mean and covariance matrix \( \Sigma \). Once \( X'_R, X'_I, Y'_R \) and \( Y'_I \) are generated for each
frequency, the surrogate data $x'(t)$ and $y'(t)$ are the inverse Fourier Transform of $X_k'(f) = X_R(f) - jX_I(f)$ and $Y_k'(f) = Y_R(f) - jY_I(f)$.

For easy implementation, we summarize the earlier analysis as follows.

**Step 1)** Estimate $S_{xx}$ and $S_{yy}$ for original data according to (4.8)

**Step 2)** Calculate Covariance matrix $\Sigma$ according to (4.11)

**Step 3)** Draw the values (realizations) of $X_R', X_I', Y_R'$ and $Y_I'$ from the Gaussian processes with zero mean and covariance matrix ($\Sigma$) for each frequency. This can be done with Matlab function based on the ziggurat method (Marsaglia, and Tsang, 2000).

**Step 4)** Take the inverse Fourier Transform of $X_k'$ and $Y_k'$ to obtain the surrogate data $x'(t)$ and $y'(t)$. To ensure where surrogate data are real valued, the negative frequency parts of $X_k'$ and $Y_k'$ are taken as the complex conjugate of the positive frequency parts.

**Step 5)** Repeat steps 3) and 4) to generate multiple realizations of the surrogate data.

$X_R'$ and $X_I'$ could be drawn from Gaussian distribution with zero mean and covariance matrix $\Sigma_x$. This simplified method is similar to the method proposed by Timmer (1995). The probability density function for the extreme value distribution (type I) with location parameter $\mu$ and scale parameter $\sigma$ is (Evans 1993).

$$y = \left(\frac{1}{\sigma}\right) \exp \left(-\exp \left(-\frac{x-\mu}{\sigma}\right) - \frac{x-\mu}{\sigma}\right)$$

(4.12)
The exact PDF is determined once \( \mu \) and \( \sigma \) are known. From this distribution, one can determine the threshold for any desired significance level (i.e. - p value).

### 4.4 RESULTS AND DISCUSSION

A seven minutes recording of 9 channels of EEG (C3, A2, O1, O2, C4, P3, P4, F3, and F4) was used in our test (16,207 trials) and shown in Figure 4.3. Precise numbers of sleep stages and standard characteristics derived from hypnograms are in Table 4.1 Number of sleep stages (17s records) are in the first column; data in the second column are the average values over 5 subjects (C3A2, C4P4, F3C3, P3O1, P4O2) related to total sleep time, the beginning of sleep is set as the first appearance of the sleep stage 2. Sleep efficiencies is the ratio of time spent in stages 1-4 or REM sleep to the whole sleep time, where also movement time and some awakenings during the night are included; in sleep medicine this parameter discriminates some sleep disorders.

Considering that the surrogate data contour plots are computed based on a randomly selected data set among \( P=1500 \) available, these maximum value comparisons suggest the known fact that there are no nonlinear or non-Gaussian components in the original data. The histogram and Gaussian fit of input data \( x(t) \) and \( y(t) \) and surrogate data \( x'(t) \) and \( y'(t) \) are shown in Figure 4.4.
Figure 4.3  Human Sleep EEG signals (9 channels C3, A2, O1, O2, C4, P3, P4, F3, and F4)
Table 4.1  Number of sleep stages (first column) and sleep efficiency and average percentage of sleep stages during the sleep (mean ± standard deviation)

<table>
<thead>
<tr>
<th>Total trails 16,207</th>
<th>Sleep Efficiency[%]: 92.6 ± 5.3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Waking</td>
<td>761</td>
</tr>
<tr>
<td>Stage1</td>
<td>809</td>
</tr>
<tr>
<td>Stage2</td>
<td>8024</td>
</tr>
<tr>
<td>Stage3</td>
<td>1536</td>
</tr>
<tr>
<td>Stage4</td>
<td>1753</td>
</tr>
<tr>
<td>REM sleep</td>
<td>3271</td>
</tr>
<tr>
<td>Movement time</td>
<td>107</td>
</tr>
</tbody>
</table>

Upper panel figures represent histogram and Gaussian fit of original data x(t) and y(t) and bottom panel figures represent histogram and Gaussian fit surrogate data x(t) and y(t). The PDFs are obtained from these parameters according to equation (4.12) and are plotted along with the histogram in blue color in Figure 4.5. Notice that the PDFs here are multiplied by the total number of surrogate data sets (1500) to match the histogram. From the PDFs, the threshold for a significance level of p < 0.05. By comparing the original data’s maximum power spectra values with the thresholds, it can be seen from the null hypothesis that the original data coming from linear Gaussian process cannot be rejected, a theoretically expected result.
Figure 4.4  Histogram and Gaussian fit of the a) original (top row) and b) surrogate data (bottom row)

Figure 4.5  PDF of (a) original data power spectra and (b) the surrogate data power spectra with Gaussian Fit
We have also performed a study to examine whether fitting an extreme value distribution to the empirical histogram is a viable approach. The fitted model of the location parameter ($\mu$) and scale parameter ($\sigma$) versus the number of surrogate data sets used are plotted in Figures 4.6 (a) and (b). It can be seen after examined around 1500 surrogate data sets, the estimation becomes linear.

![Location parameter ($\mu$)](image1)

(a) Location parameter ($\mu$)

![Scale parameter ($\sigma$)](image2)

(b) Scale parameter ($\sigma$)

Figure 4.6 $\mu$ and $\sigma$ versus the number of surrogate data sets
Table 4.2 Different frequency band detection and the corresponding true value determined from one sided Power spectra

<table>
<thead>
<tr>
<th>Wave</th>
<th>Bandwidth (Hz)</th>
<th>Time range (sec)</th>
<th>Amplitude (µV)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Delta</td>
<td>0, 1-4</td>
<td>0-5</td>
<td>10-30</td>
</tr>
<tr>
<td>Theta</td>
<td>4-8</td>
<td>5-9</td>
<td>30-70</td>
</tr>
<tr>
<td>Alpha</td>
<td>8-12</td>
<td>9-13</td>
<td>&gt; 70</td>
</tr>
<tr>
<td>Beta</td>
<td>&gt; 12</td>
<td>13 – 16</td>
<td>&lt; 100</td>
</tr>
<tr>
<td>Sleep Spindles (sigma)</td>
<td>11.5 – 16</td>
<td>16-20</td>
<td>&lt; 50</td>
</tr>
</tbody>
</table>

Table 4.2 provides the two most important characteristics of EEG elements: frequency and amplitude. Frequency is the inverse value of duration of EEG segment. The frequency range is divided into four bands: beta (12Hz and higher), alpha (8-12Hz), theta (4-8Hz) and delta (0,1-4Hz). Amplitude of EEG is taken as the peak to peak value. After the magnitude of amplitude EEG signal is divided into low-voltage, middle and high – voltage EEG. For delta, theta and alpha bands these values are as follows: 10-30µV for low voltage, 30-70µV for middle voltage and above 70µV for high voltage EEG. For beta band, the values are lower: below 10µV low voltage, 10-25µV middle voltage, and above 25µV high voltage EEG.

Beta activity is typical during wakefulness. Alpha occurs in relaxed state with eyes closed. Theta waves are dominant in normal wake state in children; in adult people theta activity appears only in small amount especially in drowsiness. Delta waves are present in deep sleep; in healthy people they do not exist in wake state. Sleep spindles are rhythmic activity in 12-14Hz frequency range and amplitude below 50 µV. The power spectra for the original (solid line) and the surrogate (dotted line) data are shown in Figure 4.7.
Figure 4.7 One sided power spectra estimated from the recording P3O1 electrode in 1sec period. The solid line is for the original data and dotted line is for the surrogate data

Three peaks around 1-14Hz can be seen, indicating the presence of synchronized activities at these frequencies. According to the traditional classification the first peak belongs to the delta band (1-4Hz) and the second peak belongs to the alpha band (8-12Hz) and third peak belongs to the beta band (>12Hz). Past research has examined whether such oscillatory neural activities self-couple and couple each other in a nonlinear fashion (Schanze and Eckhorn 1997). We study this issue with new surrogate test method.

Table 4.3 provides comparison between Olbrich and Achermann (2008) and our study. However, a few spindles are not detected by the algorithm with the chosen parameters, i.e. the maximum of \( r_k \) remains smaller than the threshold \( r_b \). Therefore, the method proposed in our analysis uses a
surrogate data approach that needs to be further optimized for more sensitive spindle detection.

Table 4.3 Comparison between AR model and AR model with Surrogate data approach

<table>
<thead>
<tr>
<th>Name of the Electrode</th>
<th>AR – Model (Olbrich et al. 2008)</th>
<th>This study</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Bandwidth (Hz)</td>
<td>Amplitude (µV)</td>
</tr>
<tr>
<td>C3A2</td>
<td>9 - 15</td>
<td>≈32</td>
</tr>
<tr>
<td>C4P4</td>
<td>11-15</td>
<td>25-42</td>
</tr>
<tr>
<td>F3C3</td>
<td>Relaxatory pole in the AR model (zero frequency)</td>
<td>11-16</td>
</tr>
<tr>
<td>P3O1</td>
<td>Relaxatory pole in the AR model (zero frequency)</td>
<td>10.5-15</td>
</tr>
<tr>
<td>P4O2</td>
<td>10 - 16</td>
<td>&lt; 15</td>
</tr>
</tbody>
</table>

In this study sleep spindles were detected in 1s overlapped segments, wide 0.5 – 16Hz band containing delta and theta was used for better eye movement separation. Alpha activity was excluded as it would increase auto covariance. With consensus scoring we expect that the agreement would improve (Olbrich and Achermann 2008). The cost benefit of this laborious task would have been low.

In this algorithm there are two different thresholds but as it can be seen from Table III, the amplitude criterion was most important (i.e. mostly amplitude in REM sleep is less than 50 µV and NREM sleep is greater than 100 µV). Schwilden et al. (2002) reported that 90% of the EEG did not show
any nonlinearity. They suggested that only under some pathological conditions, such as epilepsy, the brain signals manifest nonlinear effects. Other studies (Schack et al. 2002), (Shils et al. 1996), have shown that nonlinear characteristics exist between theta and gamma EEG signals during short term memory processing. To settle these debates, one needs carefully constructed statistical tests. The method proposed here represents our effort in this direction. We will contrast our method with other often applied techniques in this area.

Multiple ways are currently in use to generate surrogate data sets to test the significance of the sleep spindle detection. In one method, the Fourier Transform (FT) is estimated for each segment and the phase of the Fourier Transform is randomized without changing the Fourier amplitude. The result is then inverse Fourier Transformed to generate a surrogate time series (Schwilden and Jeleazcov 2002), (Schanze and Eckhorn 1997). While the phase randomization destroyed nonlinearity and leads to linear Gaussian process (Venema 2006), the Fourier amplitudes are random variables for a stationary process as well. By keeping them constant, one loses an important degree of freedom (Timmer 1995). In contrast, our method, in which both amplitudes and phases of the Fourier Transform are randomly generated, produces surrogate data sets that are explicitly Gaussian, linear and share the same second order statistics with the original data. Another method, called amplitude adjusted Fourier Transform (AAFT), has been used in recent studies (Chon, 2005). AAFT was designed to test the null hypothesis that the observed time series is a monotonic nonlinear transformation of a linear Gaussian process. This method has the same problems as the previous phase randomization method. In addition, the generated surrogate data sets usually do not have the same power spectrum as the original data, leading to false rejections when the discriminating statistics are sensitive to second-order statistical properties (Schreiber and Schmitz 1997).
4.5 SUMMARY

The approach proposed in this chapter is able to provide, thanks to the use of an efficient procedure to identify time invariant linear models (Zou et al. 2003), (Chon et al. 2005), time invariant surrogate data that mimic possible changes through time in the properties of a given time series. This approach permits application of the surrogate-based test for nonlinearity to a wider class of null hypothesis, including nonstationarity behaviors. While the improved parametric approach for the generation of typical surrogate realizations, recent studies (Keylock 2006), (Keylock 2007) proposed wavelet-based methods to generate constrained surrogate series preserving the average local properties of the original data, and are thus also suitable to track time dependent signal features. A comparative analysis of these two approaches regarding nonlinearity tests in time series with time invariant features could constitute an interesting extension in this study. In summary, detection of sleep spindles using the surrogate method proposed in this chapter is shown to give accurate results when applied to test the significance of power spectral amplitudes. It is based on solid statistical principles and overcomes some weaknesses in previous methods for the same purpose. It is expected to become a useful addition to the repertoire of nonlinear analysis methods for neuroscience and other biomedical signal processing applications.