GENETICALLY OPTIMIZED BP-ANN FOR PARAMETER ESTIMATION OF TIME VARING AUTOREGRESSIVE PROCESS

BY

ATHAUR RAHMAN NAJEEB

A thesis submitted in fulfilment of the requirement for the degree of Doctor of Philosophy in Engineering

Kulliyyah of Engineering
International Islamic University Malaysia

FEBRUARY 2018

ABSTRACT

An optimal intelligent technique to estimate Time Varying Autoregressive (TVAR) model coefficients is proposed in this thesis. Conventionally, three methods may be used to estimate the TVAR coefficients which are Direct Method (DM), Adaptive Methods (AM) and Basis Function Methods (BFM). All of these methods are built on complex mathematics and recursive in nature which increases the computation time. Although the BFM approach is preferred for few reasons such as (1) they are able to track both slow and fast changing dynamics, (2) BFM does not suffer from convergence problem as in AM. However, it is complex to compute their parameters. In addition to that a type of Basis Function (BF) is able to detect Nonstationary Signals (NSS) with similar characteristics with the BF only. Therefore, limiting the use of AM and BFM in broader NSS processing as naturally they have signal dependent characteristics. In this thesis, a hybrid framework of Artificial Neural Network (ANN) and Genetic Algorithm (GA) known as BP-ANN-GA is proposed to estimate TVAR coefficients. Superior performances of ANN in prediction and its ability to learn complex mapping of input to output is combined with optimization ability of GA to perform this task. Two different ANN architectures are proposed, one to represent TVAR and another one for TVAR BF. These ANN architectures consist of three layers. The number of nodes in input layer is determined by model orders with one hidden layer which consists of an artificial neuron. The third layer has a single node which computes the estimation error which is used to update e the synaptic weight using Backpropagation (BP) learning algorithm. Estimated TVAR coefficients are then fed into GA for further optimization by allowing the TVAR coefficient to be changed within certain limits to ensure the stability. Finally, the TVAR coefficients estimated from proposed method are used to reconstruct various NSS and compared with other methods such as AR, TVAR and BF. It is shown that proposed method yields better accuracy than BP-ANN, AR, TVAR and BF methods. It is also found that the GA optimization produces stable TVAR coefficients when the TVAR coefficients are allowed to be optimized in limits of ± 1.0.Interestingly the BP-ANN-GA also exhibits signal independence characteristics such as independence from model orders and BF, therefore allowing its application to be extended to analyze various types of NSS.

ملخص البحث

تقدم هذه الأطروحة تقنية ذكية مثالية لتقدير معاملات نموذج الوقت المتباين للإنحدار الذاتي (TVAR). تقليدياً، يمكن استخدام ثلاث طرق لتقدير معاملات الوقت المتباين للإنحدار الذاتي (TVAR) وهي: الطريقة المباشرة (DM)، أساليب التكيف (AM) وأساليب الوظيفة الأساس (BFM). وكل هذه الأساليب مبنية على رياضيات معقدة ومتكررة في طبيعتها مما يزيد من وقت الحساب. على الرغم من أن نهج BFM يُفضل لعدة أسباب مثل القدرة على تتبع ديناميكيات التغير السريعة والبطيئة، كما أن طريقة BFM لا تعاني من مشكلة التقارب كما هو الحال في AM . ومع ذلك، يوجد صعوبة في حساب معاملات هذه الطريقة، بالإضافة إلى ذلك، فإن نوع من الوظيفة الاساس (BF) قادر على كشف الإشارات غير المستقرة (NSS) ذات الخصائص المماثلة لـBF فقط وهذا يحد من استخدام AM و BFM في معالجة NSS الأوسع لأن لديهم عادةً خصائص تعتمد الإشارة. تقدم هذه الأطروحة هيكل هجين من الشبكة العصبية الاصطناعية (ANN) والخوارزمية الجينية (GA) لتقدير معاملات TVAR والمسمى هيكل BP-ANN-GA. يستفاد من الأداء المتفوق ل ANN في التنبؤ وقدرته على تعلم رسم خرائط معقدة من المدخلات والمخرجات بجانب القدرة الأمثل لـ GA لأداء هذه المهمة. تم اقتراح هيكليتين اثنتين من ANN، واحدة لتمثيل TVAR والأخرى لـTVAR BF وتحتوي كل منها على ثلاث مستويات. ويتحدد عدد النقاط في مستوى المدخلات بأوامر النموذج مع مستوى مخفي واحد يتكون من خلية عصبية اصطناعية واحدة. المستوى الثالث له نقطة واحدة تحسب خطأ التقدير الذي يستخدم لتحديث الوزن المتشابك باستخدام خوارزمية الانتشار الخلفي (BP) للتعلم. ثم يتم تغذية معاملات TVAR المقدرة في GA لمزيد من التحسين عن طريق السماح بتغير معامل TVAR ضمن حدود معينة لضمان الاستقرار. وأخيراً، يتم استخدام معاملات TVAR المقدرة من الطريقة المقترحة لإعادة بناء الإشارات غير المستقرة NSS المختلفة ومقارنتها مع أساليب أخرى مثل AR و TVAR و BF. ويظهر أن الطريقة المقترحة تعطي دقة أفضل من أساليب BP-ANN و AR و TVAR و BF. كما وجد أن تحسين GA ينتج معاملات TVAR مستقرة عندما يسمح لمعاملات TVAR أن تكون الأمثل في حدود ± 1.0 ومن المثير للاهتمام أن BP-ANN-GA يحمل أيضاً خصائص استقلالية للإشارة مثل الاستقلال عن أوامر النموذج و BF، وبالتالي السماح بتطبيقه ليتم توسيعه لتحليل أنواع مختلفة من الإشارات غير المستقرة NSS.

APPROVAL PAGE

The thesis of Athaur Rahman Bin Najeeb has been approved by the following:

Momoh Jimoh E. Salami
Supervisor

Teddy Surya Gunawan
Co-Supervisor

Rini Akmelaiwati
Internal Examiner

Ali Chekima
External Examiner

Aini Binti Hussain
External Examiner

DECLARATION

I hereby declare that this thesis the result of my own investigations, except where otherwise stated. I also declare that it has not been previously or concurrently submitted as a whole for any other degrees at IIUM or other institutions.

Athaur Rahman Bin Najeeb

Signature .... Date ....

INTERNATIONAL ISLAMIC UNIVERSITY MALAYSIA

DECLARATION OF COPYRIGHT AND AFFIRMATION OF FAIR USE OF UNPUBLISHED RESEARCH

GENETICALLY OPTIMIZED BP-ANN FOR PARAMETER ESTIMATION OF TIME VARYING AUTOREGRESSIVE MODEL

I declare that the copyright holder of this dissertation is Athaur Rahman Najeeb

No part of this unpublished research may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording or otherwise without prior written permission of the copyright holder except as provided below

1. Any material contained in or derived from this unpublished research may be used by others in their writing with due acknowledgement.

2. IIUM or its library will have the right to make and transmit copies (print or electronic) for institutional and academic purposes.

3. The IIUM library will have the right to make, store in a retrieved system and supply copies of this unpublished research if requested by other universities and research libraries.

By signing this form, I acknowledged that I have read and understand the IIUM Intellectual Property Right and Commercialization policy.

Affirmed by Athaur Rahman Najeeb

……..…………………….. ………………………..

Signature Date

I dedicate this thesis to the memory of my late father HAJI NAJEEB MUHAMMAD ISHAK for laying the foundation of what I turned out to be in life. May Allah make your resting place as Garden of Jannah.

ACKNOWLEDGMENTS

In the name of Allah, the Most Compassionate, the Most Merciful

Alhamdullilah, all praises, glorification and prayers to Allah, The Creator, The All Knowing, with Whose Grace and Mercy , for making it possible and brought me all this far. I would like to thank my supervisors Prof. Dr. Momoh Jimoh.E. Salami and Assoc. Prof. Dr. Teddy Gunawan who made this thesis possible through their consultations, guidance and fruitful thoughts. Also thanking Prof. Dr. Aisha Abdallah for her constant motivation and checking progress periodically. I am also indebted to Prof. Dr. Abiodun Musa Aibinu, Head of Mechatronics Department at Federal University of Technology Minna, Nigeria for teaching me the art of Intelligent Programming and for providing the impetus for much of my work.

Lastly, immeasurable appreciations go to my family for their endless support, prayers and concern. Finally my appreciation will not complete without sincere gratitude and thanks to staffs of Kulliyyah of Engineering Postgraduate Department for their constant assistance during the research work.

To all, I say Jazakallah.

TABLE OF CONTENT

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

AA	Akaike Algorithm
AIC	Akaike Information Criteria
AIS	Artificial Intelligent System
AM	Adaptive Method
ANN	Artificial Neural Network
AR	Autoregressive
BF	Basis Function
BFM	Basis Function Method
BLMS	Block Least Mean Square
CBF	Chebyshev Basis Function
CC	Correlation Coefficients
CMWT	Continuous Morlet Wavelet Transform
COM	Covariance Method
CORM	Correlation Method
CWD	Choi-Williams Distribution
DF	Delta Function
DCT	Discrete Cosine Transform
DM	Direct Method
DPSS	Discrete Prolate Spheroidal Sequence
EC	Error Correction
ECG	Electrocardiogram
EEG	Electroencephalogram
EMG	Electromyogram
EMD	Empirical Mode Decomposition
EOA	Evolutionary Optimization Algorithms
ES	Expert System
FA	Fuzzy Artmap
FBAR	Forward Backward Autoregressive
FF	Fitness Function
FLS	Fuzzy Logic System
FFT	Fast Fourier Transform
FFN	Feed Forward Network
GA	Genetic Algorithm
GT	Gabor Transform
GE	Gaussian Elimination
GSO	Gram-Schmidt Orthogonalization
HIS	Hybrid Intelligent Systems
HOF	Higher Order Function
HNW	Hann Window

HMW	Hamming Window
IF	Instantaneous Frequency
FPE	Final Prediction Error
KF	Kalman Filtering
LBF	Legendre Basis Function
LMA	Levenberg-Marquardt Algorithm
LMS	Least Mean Square
LNF	Linear Net Function
LOP	Low Order Polynomials
LPC	Linear Predictive Coding
LPE	Linear Prediction Error
LR	Learning Rate
LSET	Least Square Error Technique
LSM	Least Square Method
MDL	Minimum Descriptive Length
MFVSSLMS	Marr Function based Variable Step Size Least Mean Square
MLP	Multilayer Perceptron
MML	Modified Maximum Likehood
MFFN	Multilayer Feedforward Network
MSE	Mean Square Error
MSPE	Mean Square Prediction Error
NLMS	Normalized Least Mean Square
NM	Newton Method
NS	Non-Stationary
NSS	Non-Stationary Signals
OPSA	Optimal Parameter Search Algorithm
PCG	Phonocardiogram
PSD	Power Spectral Density
PRD	Percent Root Mean Square Difference
RW	Rectangle Window
RBF	Radial Basis Function
RN	Recurrent Network
RLS	Recursive Least Square
RMS	Root Mean Square
SGDA	Steepest Gradient Descent Algorithm
SLP	Single Layer Perceptron
SE LMS	Sign Error Least Mean Square
SNR	Signal to Noise Ratio
SSE	Sum Square Error
STFT	Short Time Fourier Transform
ST	S-Transform
S1	Artificial Multicomponent NSS
S2	Multicomponent Sine Wave

S3	High Frequency Sine Wave
S4	Truncated Phonocardiogram
S5	Multicomponent TV NSS
S6	Artificial Multicomponent NSS
S7	Chirp Signal
S8	Multicomponent High Frequency Sine Wave
S9	Frequency Jump Signal
S10	Truncated EEG
S11	Truncated ECG
TBF	Trigonometric Basis Function
TA	Trench Algorithm
TF	Time Frequency
TFR	Time Frequency Representation
TFD	Time Frequency Distribution
TSPE	Total Squared Prediction Error
TV	Time Varying
TIV	Time Invariant
TVAR	Time Varying Autoregressive
TVAR AM	Time Varying Adaptive Method
TVARMA	Time Varying Autoregressive Moving Average
TVAR BF	Time Varying Autoregressive Basis Function
TVMA	Time Varying Moving Average
TIV	Time Invariant
TVS	Time Varying Signals
UP	Uncertainty Principle
YWE	Yule Walker Equation
VFF	Variable Forgetting Factor
VFF RLS	Variable Forgetting Factor Recursive Least Square
WBF	Walsh Basis Function
WTBF	Wavelet Basis Function
WVD	Wigner Ville Distribution
WT	Wavelet Transform

LIST OF SYMBOLS

$a_j[n]$	TVAR coefficients at *n* instance
$\vec{a}_j[n]$	Vector of TVAR coefficients at *n* instance
b	Boundary limits in GA
$y[n]$	Output signal at *n* instances
$\hat{y}[n]$	An estimate for $y[n]$
p	AR / TVAR ‘s model order
$\sigma[n]$	Stationary Gaussian white noise
$\in[n]$	Estimation Error
m	Model order of expansion parameter
a_{ji}	BF’s constant coefficients
$\vec{c}_{j,i}$	Two dimension matrix representing TVAR coefficients in TVAR BF
$c_{i,g}(l,j)$	Correlation function
ξ	Average Mean Error Energy
$\nabla_w \xi$	BP gradient vector
$f_i[n]$	Type of BF
F()	Linear activation function
N	Data size
$x(t)$	One dimensional input signal in time domain
$T_x(t,f)$	Time Frequency Representation of a signal
t	Number of epochs in BP
T	Total number of training pattern in BP
μ	LMS Step Size
$u[n]$	LNF adder function
$\vec{\phi}[n]$	Vector of NSS Samples for LMS input
$e[n]$	Estimation Error
$\lambda_{\max}$	Largest eigenvalue in LMS
λ	RLS forgetting factor
$\overline{R}$	Correlation matrix of LMS input signal
$tr(\vec{R})$	Total Input Power
$w_j[n]$	Synaptic weights for BP-ANN
$\Delta\vec{w}_{n,m}[t]$	Vector of weight update for BP
$\vec{x}[n]$	innovative term represent inputs and BF in TVAR BF

$\nabla_w \varepsilon_n$	Gradient of error function
θ	Threshold of LNF adder function

CHAPTER ONE

INTRODUCTION

1.1 OVERVIEW

NSS are signals with changing statistical characteristics whereby their statistical property such as mean, variance, and covariance are time dependent values. While in a stationary signal, their statistical property remains constant over time. Traditionally, in processing NSS, it is common practice to assume they are stationary. However, this assumption is not always accurate especially for biomedical signals, communication signals, financial time series or acoustic signals due to their strong non-stationarity behaviors. This time dependency characteristics makes it impossible to utilize traditional analysis method such as Fast Fourier Transform (FFT) based estimators as these methods do not account for the non-stationarity characteristics in these NSS.

NSS (also known as time varying (TV) signals) are commonly found in nature, or sometimes it is generated for technological use. In both cases, it is important to understand the characteristics of these NSS and process them with appropriate tools to extract the relevant information. This area of signal processing is founded on a rigorous mathematical exposition and their applications are necessitated by the continuous developments of advanced hardware which produces complex data structure.

Traditionally, many real applications for NSS are developed by assuming the NSS are stationary (Cerutti et al., 2011; David, 2009), however such assumptions maybe valid when the frequency component varies slowly or in the case of weak stationary. However, when the signal has rapid changing frequency components as in Electroencephalogram (EEG), these method gives poor result leading to ambiguous spectral estimation (G.P.Nason, 2006) as it is incapable of tracking frequency changes in transient wave forms. Therefore it brings difficulties in extracting vital information contained in the analyzed signal.

Since 1980s there are significant developments in the analysis of NSS. The time variant (TV) methods has been well studied and proposed to track the changing frequency in a non-stationary signal (V.Oppenheim & Willsky, 1983; Jebu J Rajan & Rayner, 1996; Thonet & Vesin, 1997; Wenxing, 2003; Xu, Zhang, Hua, & Chen, 2003; S.A.Billings, H.l.Wei, & J.Liu, 2008; David, 2009; Hall, Kazlauskas & Pupeikis, 2012; G. R. S. Reddy, Rao, & College, 2014b; R. S. Reddy & Rao, 2015) It was reported in (Witte et al., 2009;Baumgartner, Blinowska, Cichocki, Dickhaus, & Durka, 2013; Wacker & Witte, 2013) that these TV methods are more appropriate for processing non-stationary signals and gradually replacing the TIV methods.

TV methods are often classified into non-parametric and parametric methods. Some of commonly used non-parametric methods include Short Time Fourier Transform (STFT) and Time Frequency (TF) distribution methods such as Wigner Ville Distribution (WVD), Wavelet Transforms (WT), Choi-Williams Distribution (CWD), Empirical Mode Decomposition (EMD) and their enhanced derivations. (F.Hlawatsch & Batels, 1992; Muthuswamy, 2004). In STFT, TV signal is separated into short time intervals fitting a sliding window, whose length is short enough to assume stationarity. Longer segments provide better frequency resolution while shorter segments provide better time resolution. A good resolution in both domains cannot be obtained simultaneously. Other algorithms in this class are limited by uncertainty principle. Further details on these methods are outlined in Chapter 2.

TV parametric methods are based on recursive model with time dependent parameters. Parametric methods are used to overcome the aforementioned deficiencies of non-parametric methods and have been applied to almost all application of signal processing in both stationary and non-stationary cases. More parsimonious representation of signal and higher resolution in time frequency spectra is obtained in the parametric method. It was shown that even though the available signal is very short as in the case of biomedical signals, the parametric method yields very high frequency resolution in spectral estimation.

TV parametric modelling techniques have been applied to almost all fields of endeavor including biomedical signal processing. Among the widely used TV parametric method includes TV Autoregressive (TVAR), TV Moving Average (TVMA) or TV Autoregressive Moving Average (TV ARMA). Among these, TVAR has been investigated by many researchers (Hall, V.Oppenheim, & Willsky, 1983; Eom, 1999; Sodsri, 2003; L. Zhang, Xiong, Liu, Zou, & Guo, 2010;Y. J. Chu, Chan, Zhang, & Tsui, 2012a; R. S. Reddy & Rao, 2015). This evolving paradigm has attained much attention for few reasons, mainly due to computational load to compute TVAR parameters are much less than for TVMA or TVARMA.

TVAR is very much like the conventional Autoregressive (AR), except it has TV coefficients, which changes through time to allow estimation of time dependent characteristics. Furthermore, TVAR inherit all the advantages of AR in terms of high accuracy, good resolution, sharp peak, and less spurious component event for short data record (Y. J. Chu et al., 2012a; W. Wang & Wang, 2003; L. Zhang, Xiong, Liu, Zou, & Guo, 2010). Mathematically, a p-th order AR and TVAR are presented as:

$$\text{TVAR}: \ y[n] = -\sum_{j=1}^{p} a_j[n] y[n-j] + \sigma\ [n] \tag{1.1}$$

$$\text{AR}: y[n] = -\sum_{j=1}^{p} a_j y[n-j] + \sigma\ [n] \tag{1.2}$$

where $a_j[n]\ ; j = 1,2,3 \ldots p$ are time varying AR coefficients

p , is the model order

$\sigma[n]$ is zero mean, stationary Gaussian white noise process also can be represented as $\in [n]$, the estimation error

In AR or TVAR parametric modeling, their coefficients, a_j or $a_j[n]$ contains paramount information about the signal itself. These coefficients are used to reconstruct the signal, plot an instantaneous frequency or plot time varying spectrum of frequency content of signal of interest. Therefore, in AR or TVAR modeling, the main work is to compute their coefficients effectively.

Identifying TVAR coefficients is a statistical problem which involves three stages; model order determination, TV coefficients computation and model verification. Therefore, the selected algorithm to implement TVAR should be computable, robust and reliable to reveal the underlying information.

Two main classes of methods have been proposed in literatures to estimate $a_j[n]$ i.e. (1) an Adaptive Method (AM) and (2) a Basis Function (BF) method. The AM uses recursive algorithm based on adaptive filters to estimate $a_j[n]$. This approach is known as stochastic approach, where the coefficients of the associated models are treated as random processes with stochastic model structure. The most popular methods to deal with this class of models are the Recursive Least Squares (RLS), Least-Mean Squares (LMS) and Kalman Filtering (KF).

While in TVAR Basis Function (TVAR BF) approach, $a_j[n]$ are expended as a finite sequence of predetermined time dependent basis functions. More details, comparison and critical review are further explained in Chapter Two of this thesis.

1.2 PROBLEM STATEMENT AND ITS SIGNIFICANT

TVAR BF approach has received much attention in the literature and has been extensively applied to analyze various biomedical signals such as Electromyogram (EMG) (Al, Ferdjallah, Ph, & D, 2006), Phonocardiogram (PCG) (Bassam, Abbas, & Kasim, 2008), Electrocardiogram (ECG) (Paul, Wan, & Nelson, 2006) , to name few. In TVAR BF method, each TVAR coefficient $a_j[n]$ is assumed as a linear combination of time dependent basis function as expressed in Equation (1.3) :

$$a_j[n] = \sum_{i=0}^{m} a_{ji} f_i[n] \tag{1.3}$$

Where m, a_{ji} and $f_i[n]$ are known as the expansion parameters.
a_{ji} is constant coefficient of expansion parameter.
m is order of the expansion parameter which represents maximum number of basis sequence.
$f_i[n]$ is a set of selected basis function.

Substituting Equation 1.3 into Equation 1.1 yields:

$$y[n] = -\sum_{j=1}^{p} \left(\sum_{i=0}^{m} a_{ji}[n] f_i[n]\right) y[n-j] + \sigma\ [n] \tag{1.4}$$

The accuracy of model parameters in TVAR BF is important to obtain accurate observation, i.e. either to analyze the time-frequency spectrum, signal reconstruction or to observe the instantaneous frequency (IF). While the accuracy itself highly depends on the methods used to determine the model parameters; hence successful implementation of TVAR BF requires appropriate algorithm which will not produce inaccurate model parameters. The use of inaccurate model parameters often leads to poor spatial resolution (Wei, Liu, Billings, & Street, 2010), introduction of artifacts, introduction of noise, erroneous peaks and valleys, system instability and many more.

From Equation 1.4, in order to estimate $a_j[n]$ via BF approach, four different computations are required to be executed simultaneously. The model orders p and m must be chosen, proper basis function is selected from pool of available basis functions and set of a_{ji} is computed for $\{j = 1,2,\cdots, p; i = 1,2,\cdots q; a_0[n] = 1\}$.

Hence, it is obvious that TVAR gives rise to computational complexity. For example, to analyze a NSS with N samples using the conventional AR model order of p will require computation for p coefficients. But the TVAR process requires computation of $a_j[n]$ for each data sample, *n*. There are $N * p$ number of coefficients if TVAR AM is adopted while TVAR BF requires a total number of $(m+1) * p$ parameters to be identified; usually far less than $N * p$. Hence TVAR BF is more economical approach than TVAR AM; nevertheless, the computation load of TVAR BF is still inherently complex and heavy.

In addition to the computational complexity and TVAR BF successful approach in various fields of applications, it has three main drawbacks, the difficulties in determining model orders, complexity in determining the TV coefficients, $a_j[n]$ and no particular rule set to determine the type of basis function.

From Equation 1.4, for each n, $a_j[n]$ has been replaced with an expansion coefficient a_{ji}. Once a_{ji} is obtained, $a_j[n]$ can now be computed using Equation 1.3. In order to compute for a_{ji}, Equation 1.4 is modelled using a Least Square Error Technique (LSET) which leads Equation 1.4 to be represented in a $(m+1)*p$ matrix. Correlation, covariance or least prediction terms (Grenier, 1983; Hall et al., 1983; Pally, 2009; Jebu J Rajan & Rayner, 1996; G. R. S. Reddy, Rao, & College, 2014a) are three available methods to solve the matrix.

While the computation complexity of the correlation and the covariance method is being one issue, their performance matters too. In stationary signal analysis (A.M. Aibinu, Salami, & Shafie, 2012), the correlation technique has shown to produce low resolution spectra due to windowing in data matrix. In addition to that it shows poor performance for short data. While the covariance method may lead to an unstable system by producing false peaks while the major issue in least prediction approaches is appearance of spectral line splitting. These performances are inherited by TV coefficient as well.

The second problem associated with TVAR BF is finding an appropriate method to estimate model order, p and m. For accurate modelling of NSS, it is important to estimate p and m accurately, not to be over modeled or under modeled. At present, many known successful cost function such as Final Prediction Error (FPE), Akaike Information Criteria (AIC), Minimum Descriptive Length (MDL) are subjected to exhaustive search and designed for stationary models. Therefore, their effectiveness in approximating an optimum model order for non-stationary signals is questionable. In addition, the criteria of determining TVAR BF model orders are more complex as TVAR BF has two model orders to be determined. Furthermore the cost function for TVAR BF model orders are rare and has received little attention in literatures (Jebu J Rajan & Rayner, 1996).

As the optimal model order is not known a priori, models orders p and m are evaluated repeatedly in a range of $[1, p]$ for p and $[1, m]$ beginning with a low number, then increasing to a maximum number in an iterative process. Then the best orders are then selected using criteria determined by a TVAR BF model order cost function. This iterative computation resorts to exhaustive search and increases the computation time. One easy solution to avoid this iterative computation search is by fixing the model orders p and m (assumed to be known) and hence decreases the computation time. Unfortunately, fixing the model orders may be an easy way to reduce then computation time, but however, this may have serious effect on performance as the fixed number may not represent the optimal model order. It could be under modeled or over modeled.

The third problem with TVAR BF is the method of selecting the appropriate BF. Since the TVAR BF coefficients are defined on the space spanned by BF, it is paramount importance to select the proper BF. Many different BF have been proposed including Trigonometric (TBF) (Hall et al., 1983) , Wavelet (WTBF) (R. S. Reddy & Rao, 2015; S.A.Billings, H.l.Wei, & J.Liu, 2008), Legendre (LBF), Chebyshev (CBF), Low Order Polynomials (LOP) (Rao, 1970) ,Discrete Prolate Spheroidal Sequence (DPSS) (Zou & Chon, 2004), and Walsh (WBF) to name a few . None of these BFs provide definitive solution, partly because the choice of BF needs some priori knowledge on time variations presence in an input signal, where in most cases it is not available. Furthermore, all of the aforementioned BF is designed for a particular form of dynamics. For example, LBF and DPSS perform well if the frequency changes smoothly with time. On the other hand, WBF behaves well for piece-wise stationary and is applicable on signal with fast changing dynamics (G. R. S. Reddy, Rao, & College, 2014).

Due to their variety of performances, there is no guideline proposed for selecting the best BF which may suit all kind of NSSs. Generally it is known that biomedical signal composes of multiple TV frequencies, therefore choice of one BF is insufficient and inadequate. Such system responds well to multiple set of BF.

Basically, from the previous discussion, problems associated with coefficient determination for TVAR and TVAR BF methods are identified as:

1. TVAR and TVAR BF model order determination methods involve exhaustive and iterative searching methods.
2. TVAR and TVAR BF coefficient computation algorithms are founded on complex mathematics and problem specific with mixed performance.
3. BF Methods are not signal dependent, but problem dependent. BFs have their own unique characteristics in which they able to capture a single dynamic with similar characteristics. Therefore, no specific guideline on selecting a set of BF is available. In the case of multiple transient signals, which composed multiple forms of dynamics; more than one BF may be required.
4. The computational cost is high due to the nature of TVAR and TVAR BF algorithms are iterative and exhaustive.

1.3 RESEARCH PHILOSOPHY

In previous works to estimate TVAR BF model parameters, many algorithms have been identified and improved since it was introduced in 1970. Although these techniques produced satisfactory results for some applications, yet, there are needs to consider more comprehensive mathematic model as the demand for more sophisticated software is on the rise in par with new complex hardware which are being developed rapidly. Therefore a TVAR or TVAR BF model which is applicable on broad range of application is required here.

It is conjectured that an Artificial Neural Network (ANN) based technique for determining appropriate TVAR model parameters would be most appropriate as it is a high performance general non-linear mathematical computing procedure that emulates the operation of biological neural system in the human body. Thus, the iterative and exhaustive searching procedure in TVAR and TVAR BF parameters computations are now mimicked through the learning and weight optimization procedure.

ANN consists of nonlinear layered structures, massive parallelism, connectionist and adaptability which can be utilized in learning and optimizing the TVAR parameters. Thus, the required TV coefficients can be obtained from a well-trained ANN structure and further optimized using Genetic Algorithm (GA). Optimization of TVAR coefficients will bring the estimated signals closer to the targeted signal, and hence signal dependent TVAR coefficients can be obtained. Since GA emulates nature in the search of an optimal solution of a given problem by using biological natural evolution process. It is conjectured, an algorithm based on ANN is the most appropriate tool for computing the TVAR coefficients.

Since hybrid of ANN and GA has been adopted in various fields, with superior performance demonstrated (Z. Abo-Hammour, Alsmadi, Momani, & Abu Arqub, 2013; L. Hu, Qin, Mao, Chen, & Fu, 2016), this splendid combination could be extended to estimate TVAR coefficients as well. However, the challenge here is to identify the ANN parameters, ANN structure, GA parameters and to formulate a relationship between the ANN and GA which work could together to compute TVAR coefficients accurately.

1.4 RESEARCH OBJECTIVES

The main objectives of this research are to develop a signal dependent intelligent algorithm to estimate the TVAR coefficients, while the specific objectives are:

a. To proposed and develop an ANN based algorithm to estimate TVAR coefficients.
b. To develop an intelligent optimization method to optimize the estimated TVAR coefficients using GA.
c. To investigate the performance of the proposed hybrid method (which consists of ANN and GA) to estimate TVAR coefficients of various NSSs to demonstrate its signal dependency characteristics and at same time to reduce model order dependent.

1.5 RESEARCH QUESTIONS

The objective of this thesis is developed based on following research questions:

a. What are the challenges in NSSs and how it has been handled?
b. What are the limitations of the existing TVAR coefficient computing methods?
c. Is it possible to reformulate TVAR model to suit the ANN structure?
d. What role can GA play in optimizing the estimated TVAR model coefficients?
e. Is there a general algorithm exists for analysis of various types of NSSs ?

1.6 RESEARCH APPROACH

To achieve the research objectives, quantitative approach is adopted and this research is conducted on three phases which are:

a. Planning Phase

   Objective of this early phase is to formulate research problems, objectives and scope. Literature review is conducted extensively to identify the research gap on NSS analysis, parametric modeling of NSSs, review of Artificial Intelligent Systems (AIS) including ANN and GA.

b. Implementation

   This phase begin with conceptualizing the proposed design, coding and followed by generation of artificial signals and data collection of biomedical signals from available database. The proposed method is implemented using Matlab®.

   Development of Signal Dependent ANN-GA hybrid framework by rigorous coding, debugging and verification are core work in this phase.

c. Testing and Validation Phase

   The newly developed Signal Dependent Hybrid ANN-GA Framework has been applied on various types of non-stationary signal.

Fig 1.1 depicts the flowchart of research approach.

1.7 RESEARCH SCOPE

This thesis presents a signal dependent ANN- GA framework in order to determine TVAR coefficients based on DM and BF approach for broad varieties of NSS, (namely signal dependent TVAR coefficients) taking advantage of superior performance of ANN combining with optimization accuracy of GA. Such method is known as the hybrid method. This work is limited to determination of TVAR coefficients using ANN and GA only. The TVAR coefficients estimated from the proposed algorithm are then verified by reconstructing various NSS.

1.8 THESIS ORGANIZATION

The thesis is organized into five chapters. The research approach, research overview, problem statement and its significant, research objectives and scope are presented in this chapter. Literature review on processing of NSSs is presented in Chapter Two which covers reviews of both the non-parametric methods and parametric methods such as TVAR, TVAR BF, Adaptive Methods (AM) and BF to estimate the TVAR coefficient.

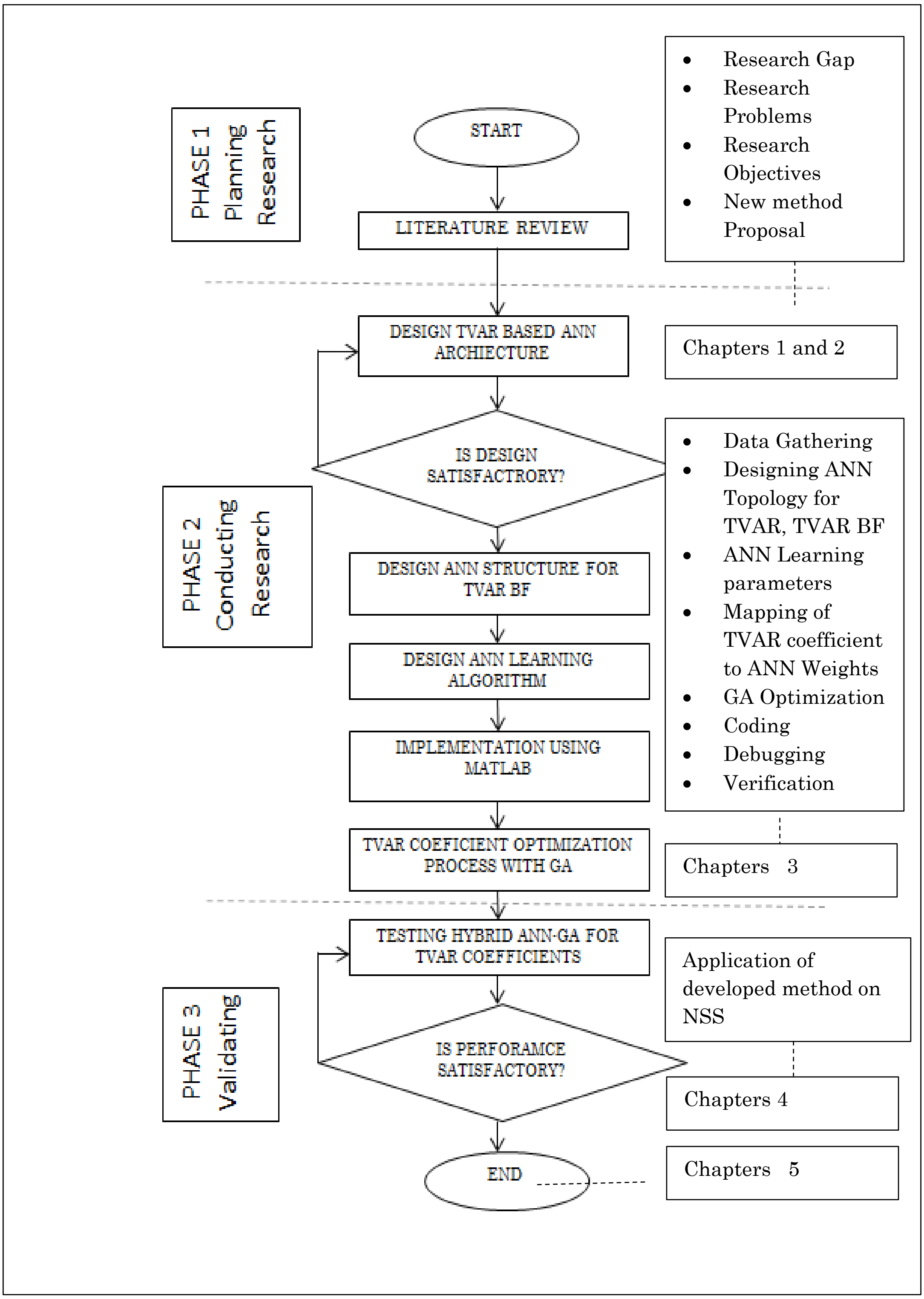


Figure 1.1 Research Approach Flowchart

Findings are presented in a tabular form. Introduction to the Artificial Intelligent System (AIS) which include ANN and GA is presented at the end with a critical analysis which identifies the research gap and thus proposes a way forward to estimate TVAR coefficients.

In Chapter Three, development of the signal dependent BP-ANN-GA framework has been discussed. The proposed hybrid method is based on the mathematical model of TVAR and TVAR BF approach. Two different BP-ANN-GA model s are developed in 3 stages namely (a) establishment of the relationship between ANN and BP Learning algorithm to estimate TVAR, (b) development of ANN structure for TVAR and TVAR BF and finally, (c) optimization using GA. Implementation flowcharts, system structure and simulation on NSSs are presented as well.

Chapter Four present performance analysis of the proposed method. Benchmarks of parameters are decided and TVAR coefficients are estimated using proposed method is conducted. Estimated TVAR coefficients from various methods are then used in reconstruction of various NSSs to show reliability and signal dependency character of proposed method, BP-ANN-GA. The thesis is concluded in Chapter Five where discussion on results and summary of the work is presented. The main contribution is highlighted and recommendation for future work is presented.

CHAPTER TWO

LITERATURE REVIEW

2.1 INTRODUCTION

Naturally, it is very difficult to find an ideal stationary random (stochastic) process in nature. Even a white noise process digitally generated by computers is not ideally stationary because of the limited frequency band. Many non-stationary stochastic signals are approximated as a stationary process or transformed into stationary process for processing. The processing of a stationary signal is based on assumption that the statistical property of those signals such as mean or variance is invariant with respect to time shifting or constant values. In processing stationary signals one assumes only a single realization is available and the algorithms are developed for one single, long sample of process; which is known as an ergodic process.

Practically although many NSSs can be processed using stationary assumption, however, some non-stationary processes such as earthquake, speech signal or biomedical signals as EEG cannot be easily transformed to stationary process. In this non-stationary case, the process is strictly not ergodic; therefore stationary assumption is not applicable, invalid which leads to an inaccurate representation with misleading information. Hence, these signals must be analyzed using a non-stationary method such as Time-Frequency Representation (TFR) methods.

In TFR algorithms, NSSs are presented as a function of both time and frequency $S(t, f)$ known as the spectrogram. A spectrogram is graphical representation of time dependent spectrum which shows energy distribution rephrase over time vs frequency span. With TFR, information on time localization of frequency is visible which was not possible earlier.

Available TFR algorithms can be grouped to non-parametric methods and parametric methods, both with their own advantages and with different stationary assumptions. In non-parametric method such as STFT, the NSS is segmented into small segments, where stationary is assumed within this segment. While in time varying parametric method such as TVAR, model coefficients are allowed to be varying in time, whereby the stationary is assumed by processing each sample, *n* of the NSS. Then a TV spectrum (spectrogram) is computed from TVAR coefficients.

In recent time, parametric method of analyzing NSS becomes a method of choice for researchers due to many advantages over non-parametric method. The parametric modeling of TVS has been successfully applied to solve various problems in human endeavors. Among available TV parametric methods, TVAR is a method of choice. However, the successful application of TVAR heavily depends on efficiency of algorithm used to compute the TVAR coefficients. This algorithm should be sufficient, efficient, robust and computable.

In this chapter, the basic concepts of NSS processing are first reviewed in Section 2.2. The remaining of this chapter is organized as follows: TVAR process modelling is presented in Section 2.3 and their coefficient estimation techniques in Section 2.4, respectively. Two approaches to estimate TVAR coefficients are found in literatures which Adaptive Method (AM) and Basis Function (BF) Method. These methods are further explored in Subsection 2.4.1 and 2.4.2 respectively; while their performances are compared at the end of Section 2.4. A list of related studies is contained in Section 2.5. An introduction to Artificial Intelligent System (AIS) is reviewed in Section 2.6. Finally critical analysis of TVAR parametric method is presented in Section 2.7.

2.2 REVIEW OF NSS ANALYSIS

A NSS is a signal whose frequency changes over time, therefore, its statistical properties such as mean, variance, correlation or covariance also vary in time. A sample of artificial NSS with multiple frequency component is shown in Fig 2.1. Most real world signals including speech signals, sonar signals, communications signals and biomedical signals are NS by nature.

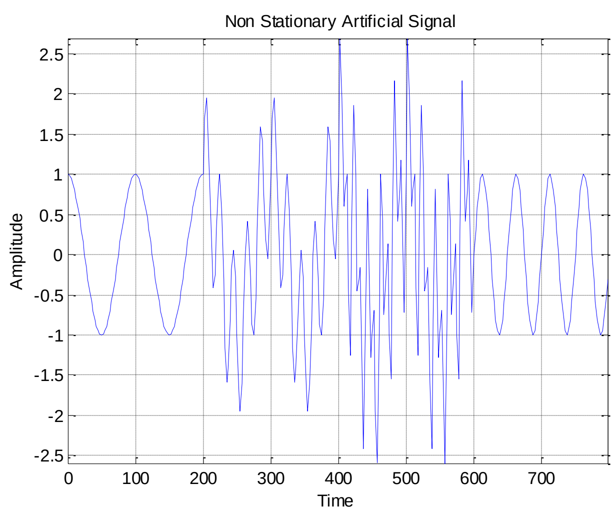


Figure 2.1 Sample of NSS

Analysis of NSS falls in two categories as illustrated in Fig 2.2, which are classical (or also known as traditional methods) and Time Frequency Representation (TFR) methods. Classical methods models the NSS based on assumption that the TV signal is locally stationary or slowly time varying on short data record. One popular algorithm is the Fast Fourier Transform (FFT) algorithm which offers low computation cost, fast and direct in term on implementation. Basically, algorithms under this category are techniques developed for stationary signals, but their applications are extended for NSS without any consideration for TV characteristics.

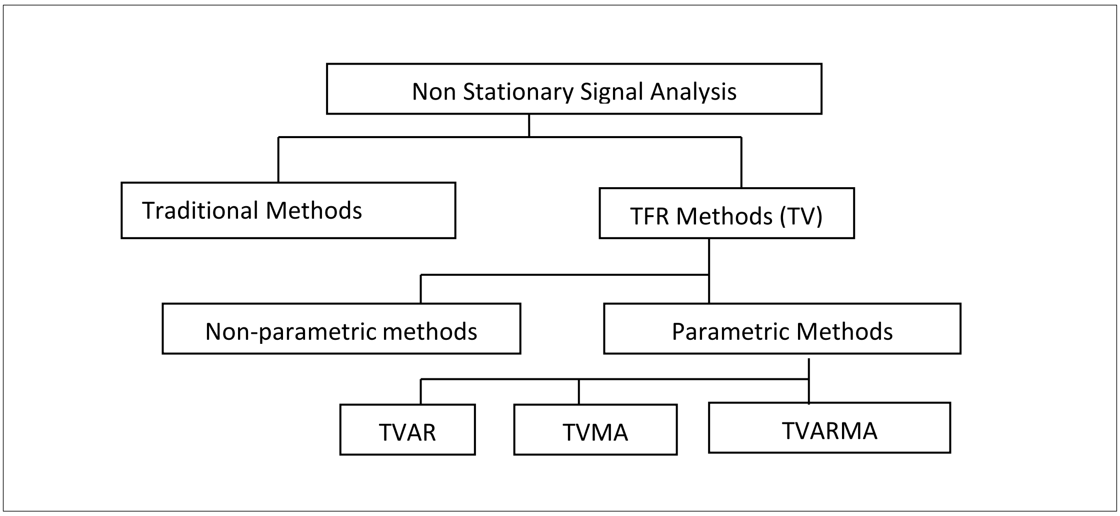


Figure 2.2 Classification of NSS Analysis Methods.

The classical methods may provide good results when the frequency components in a NSS changes slowly with time. But on the other hand, when the input signal has rapidly changing frequency components, even in small data records, classical methods will give a poor estimation or completely misleading spectral representation. This problem can be illustrated by comparing periodograms of a chirp signal with an impulse signal, which is illustrated in Figure 2.3(a) and Figure 2.3(b). Periodograms of both signals are shown in Figure 2.3(c) and Figure 2.3(d) where similar pattern is observed (Sandsten, 2016) . Similar observation was reported in (Moukadem, Djaffar Ould Abdeslam, & Dieterlen, 2014) , where same periodograms pattern was produced for two different chirp signals using FFT. The reason is because FFT integrates the frequency component over time assuming ergodic process, therefore do not contain any information about time localization of the signal (F.Hlawatsch & Batels, 1992).

A more suitable approach for analysis NSS is by using TFR where the NSS are represented as a function of both time and frequency simultaneously. TFR maps a one-dimensional signal from time domain, $x(t)$ into a two-dimensional function of time and frequency, $T_x(t, f)$. The TFR surface over time frequency plane gives more information about the signal including time localization of frequency components. Whilst this is not possible with FFT as FFT establishes one to one relation between time domain and frequency domain by converting time domain representation into frequency domain representation and vice versa.

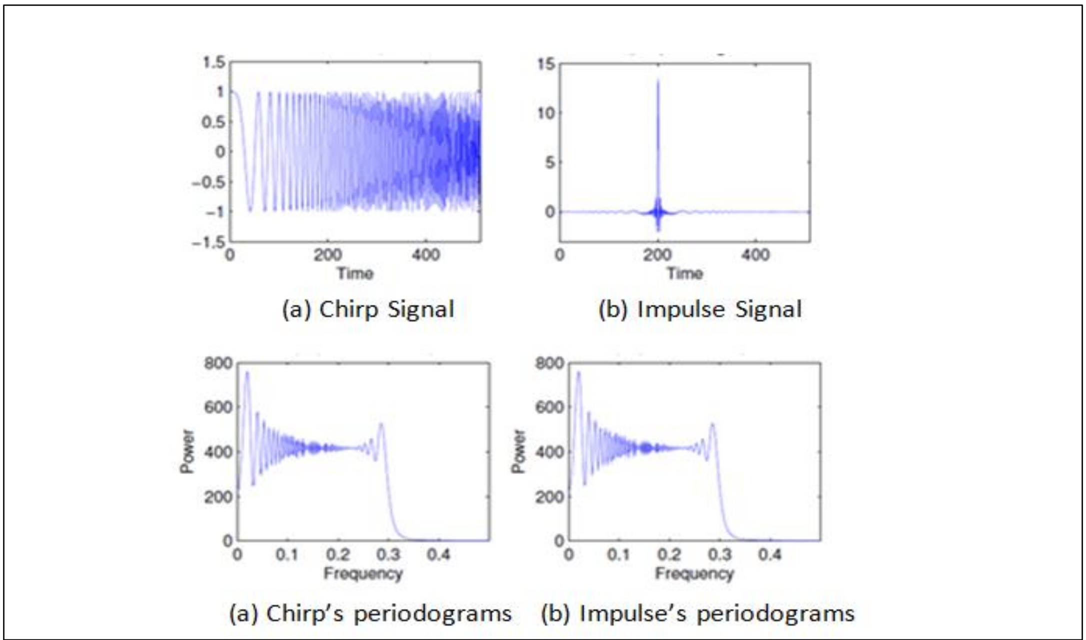


Figure 2.3 Two Different TV Signals With Similar Periodograms (Sandsten, 2016). (a) Shows a chirp signal,(b) an impulse signal (c) Periodograms of chirp signal (d) Periodograms of impulse signal

Available TFR algorithms can be categorized into non-parametric and parametric approaches. In non-parametric TFR, the best known and most frequently applied techniques in biomedical signal processing are STFT, Gabor Transform (GT), CWD, Continuous Morlet Wavelet Transform (CMWT), S-Transform (ST), WVD and their derivate. This list can be further divided into two groups namely linear transforms and quadratic transforms (Wacker & Witte, 2013) . A well detailed mathematical description of these transforms can be found in (F.Hlawatsch & Batels, 1992; White, 1996;Muthuswamy, 2004; Wacker & Witte, 2013; Zhipeng Feng, Ming Liang, 2013) and a brief information is presented here.

STFT is the simplest and most direct transform in this category, where the input signal is separated into small time intervals fitting a sliding window, whose length are short enough to assume stationary. STFT is basically a piece-wise FFT, where by in each segment, the NSS is multiplied with the type of analysis window and subsequently FFT is calculated. The function of this window is to localize the spectrum in time to produce local spectra. This localizing window then moves on time axis to next window one by one to produces spectra for entire signal. The TFR plot using STFT is known as spectrogram. Spectrogram of a chirp signal and an impulse signal is shown in Figure 2.4(a) and Figure 2.4 (b). Comparing this to periodograms in Figure 2.3(a) and Figure 2.3(b), it is obvious that the spectrogram provides more information by emphasizing the time locality. Thus it is more reliable to represent a NSS using a spectrogram compared to periodograms.

Although STFT gives time localization in the spectrum, however , STFT is not always sufficient solution as it suffers from Uncertainty Principle; in which it is unlikely to obtain fine resolution on both time and frequency domain simultaneously.

A short window length yields good time resolution but poor frequency resolution and vice versa. Furthermore, performance of STFT heavily depends on selected window property such as width and type of window such a Rectangular Window (RW) , Hann Window (HNW) or Hamming Window (HMW) to name few. Therefore time-frequency precision is not optimum (Baumgartner, Blinowska, Cichocki, Dickhaus, & Durka, 2013).   More on non-parametric methods' strength and limitation is listed in Table 2.1.

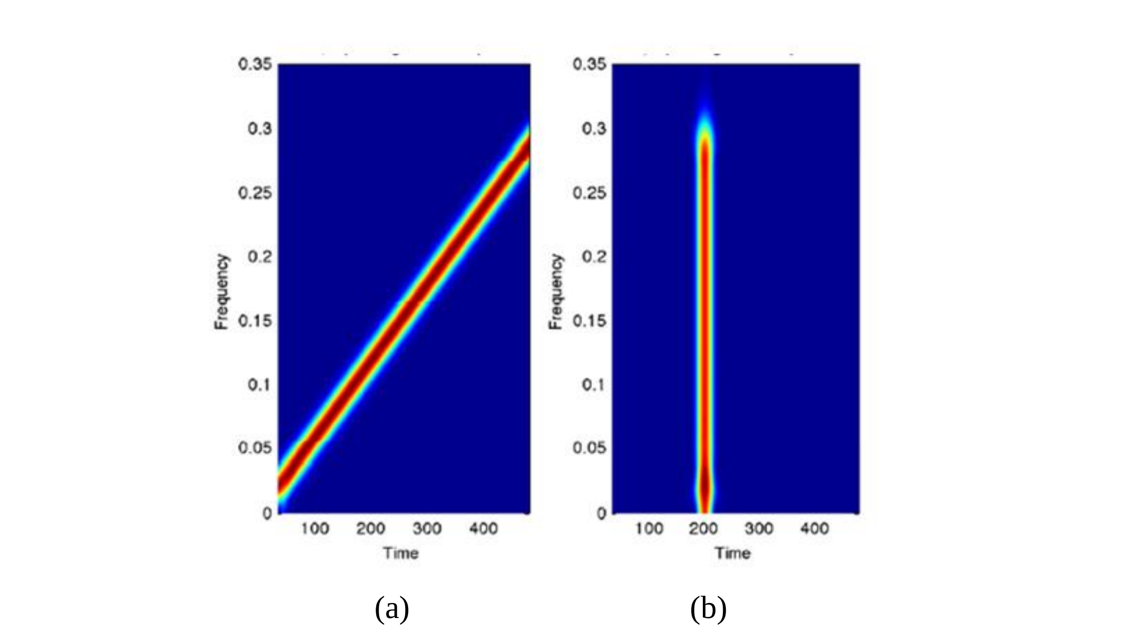


Figure 2.4 Spectrograms of (a) a chirp signal and an (b) impulse signal. Red color indicates high power while blue color indicates low power. (Sandsten, 2016).

On the other hand, parametric methods were proposed to overcome the aforementioned deficiencies of non-parametric methods. By using a parametric modelling technique; one could achieve more parsimonious representation of a signal and a higher resolution spectrum can be obtained even for short data records . Furthermore, they are not subjected to Uncertainty Principle and in addition to that, they are also able to detect slow, fast or slow and fast changing frequency of a multi-component signals.

Table 2.1 Evaluation of Selected Non-Parametric TFR

Method	Strength	Limitations
STFT	Fast Easy implementation Industry acceptable	Time and Frequency resolution tradeoff Heavily depend on window type Width of window as prior information
WVD	High resolution in time and frequency in single component frequency signal	Artifacts in mono-component signal Cross-term in multi-component signal Low resolution in Windowed WVD Complicated mathematics
CWT	Improved resolution of multi-component signal	Subjected to uncertainty principle Complicated mathematics

The underlying mathematics in parametric modeling methods are rational transfer function, where the exact form of this rational transfer function is determined by estimating suitable values for its free parameters. Depending on the location of these parameters, the parametric modeling for non-stationary signal analysis is grouped into Time Varying Autoregressive (TVAR), Time Varying Moving Average (TVMA) and Time Varying Autoregressive Moving Average (TVARMA) models.

Among these three models, TVAR modeling has received the most attention in the literatures in the last few years (A.M. Aibinu, Salami, & Shafie, 2012; Y. J. Chu, Chan, Zhang, & Tsui, 2012; G. R. S. Reddy, Rao, & College, 2014; R. S. Reddy & Rao, 2015;). The reasons for the popularity and preference of TVAR are:

I. TVAR offers computation efficiency and easy implementation (Aibinu et al., 2012).

II. TVMA and TVARMA coefficients require complicated procedure (Palaniappan, 2006)

III. Many natural signals have underlying AR nature (Aboy, Márquez, Mcnames, & Hornero, 2005).

IV. If a signal which is not AR in nature can be modeled as an AR process if a sufficiently large model order is selected (Aboy et al., 2005).

V. Detection of multiple peaks in presence of noise (Eom, 1999).

VI. TVAR inherits all advantages of AR in terms of higher accuracy, sharp peaks, less spurious component (artifacts) (Zhang et al., 2010)

Therefore, for the reasons stated above TVAR is further considered in this research. We now proceed to present detailed review on TVAR and its coefficient determination techniques.

2.3 TVAR MODELING

A TVAR model which is driven by a white noise sequence is formulated as:

$$y[n] = -\sum_{j=1}^{p} a_j[n]y[n-j] + \sigma\ [n] \qquad (2.1)$$

or $$\hat{y}[n] = -\sum_{j=1}^{p} a_j[n]y[n-j] \qquad (2.2)$$

where $a_j[n]$; $j = 1,2,3 \ldots p$ are TVAR coefficients

p is the model order

$\hat{y}[n]$ is the estimate of $y[n]$

$\sigma[n]$ is zero mean, stationary Gaussian white noise process also can be represented as $\in [n]$, the estimation error

Basically, modelling a NSS using TVAR for a single realization of n is to accurately determine model order, p and to estimate the $a_j[n]$.This is realized in three step process for each, n which are model order determination process, the model coefficient estimation and the model validation step. In general, to represent a NSS using TVAR model, the TVAR coefficient determination technique should be sufficient, efficient, robust and computable in order to obtain a successful modeling otherwise results obtained will fail.

Once $a_j[n]$ are computed, the TV Power Spectral Density (PSD) can be calculated using $a_j[n]$ from Eq. (2.3):

$$S_y(n, f) = \frac{\sigma^2}{\left|1 + \sum_j a_j[n] e^{-j2\pi f}\right|^2} \tag{2.3}$$

$$\sigma^2 = \frac{1}{N-p} \sum_{n=p+1}^{N} \varepsilon[n]^2 \tag{2.4}$$

where, $S_y(n, f)$ is TV spectra for sample, n, N is total number of data, $\varepsilon[n]$ is estimation error.

2.4 TVAR COEFFICIENT ESTIMATION METHODS

The estimation of the TVAR coefficients is classified into one of the two categories: namely the AM and the BF methods. The AM methods are based on adaptive filters which process is a stochastic approach where the TVAR coefficients are treated as a random process. While in the BF approach, TVAR coefficients are expanded as a finite sequence of known BF (or set of BF). We now proceed to detail both categories and compare them from literatures.

2.4.1 TVAR model coefficients estimation using AM

In AM, the variation of $a_j[n]$ relies on a dynamic model (Sodsri, 2003;Nosrati et al., 2015) which is defined following the general form of an adaptive filter as :

$$a_j[n] = a_j[n-1] + \Delta a_j[n] \tag{2.5}$$

In this category, the $a_j[n]$ are updated from their previous values, $a_j[n-1]$ while the term $\Delta a_j[n]$ represents an innovation term which is used to optimize $a_j[n-1]$ by adopting various methods which suits the signal. The AMs are named following the type of adaption applied on $\Delta a_j[n]$. Two popular AMs are Least Mean Square (LMS) and Recursive Least Square (RLS) which will be discussed further here in relation to estimation of $a_j[n]$.

The LMS algorithm is based on the steepest descent method. The initial work and simplest form of steepest descent LMS was popularized by Widrow and Hoff (Widrow & Hoff, 1960; Bernard Widrow et al. 1976). This variant of LMS is considered as standard version, where the vector $\Delta\vec{a}_j[n]$ is defined (Rangayyan, 2003) as :

$$\Delta\vec{a}[n] = 2\mu e[n]\vec{\phi}^*[n] \tag{2.6}$$

Where p is filter order, or model order, μ is the step size, or also known as learning rate and $\vec{\phi}[n] = [y[n], y[n-1], \cdots, y[n-p]]^T$ is a vector of sampled NSS. The implementation of Widrow and Hoff steepest descent LMS is direct without much computation as given as:

a. Initialization of LMS parameters, $a_1[n] = 0\,;\,\mu$

b. For each data sample, $n=0, 1, 2, \cdots, N-1$

i. for each value of p

ii. Calculate the estimated signal, $\hat{y}[n] = -\sum_{j=1}^{p} a_j[n]y[n-1]$

iii. Calculate the error signal, $e[n] = y[n] - \hat{y}[n]$

iv. P is selected from minimum of $e[n]$

v. Update the coefficients for $n+1$; $\vec{a}[n] = \vec{a}[n-1] + \mu e[n]\vec{\phi}[n-1]$

The popularity of LMS methods to a large extent, is due to its computational simplicity (Edmonson, 1998; Rangayyan, 2003) and ease of implementation as it does not involves differentiation, squaring, averaging or solving huge matrixes. Furthermore, its behavior is quite simple to understand and appears to be robust as well (Olmos & Laguna, 2000). However, their performance highly depends on the step size, μ, which is a crucial factor that controls the stability and the rate of convergence.

To ensure stability and convergence, (Widrow et al., 1976) have shown that it is necessary to define μ in a range of $0 < \mu < 2/\lambda_{\max}$, where $\lambda_{\max}$ is the largest eigenvalue of a correlation matrix of input signals, $\bar{R}$. However, practically $\lambda_{\max}$ is rarely known and not always easy to apply. Therefore, a simple working rule for $\lambda_{\max}$ is proposed by equating $\lambda_{\max}$ to total input power, $tr(\vec{R})$, which is defined as:

$$tr(R) = \sum_{i=0}^{p-1} E[\left|y(n-i)\right|^2] \qquad (2.7)$$

Such derivation for μ will result in a fixed value for μ. It is reported in various literatures, that the convergence rate of a fixed μ is slow. To increase the convergence rate, one may increase the μ.

Generally, it is known that the convergence time is inversely proportional to μ, hence selecting a large μ will produce a fast converging algorithm and vice versa. However, a large μ tends to increase the gradient noise which is produced naturally by the standard LMS (Widrow, 1966). The gradient noise will get amplified when μ becomes larger.

A low gradient noise and fast convergence are two fundamental requirement for a stable system. In standard LMS algorithm, the gradient noise can be suppressed by using a small μ. However on the other hand, it will not converge fast. But if a large μ is opted for faster convergence, it will also amplify gradient noise. Therefore, both low gradient and fast convergence are not possible to be attained simultaneously using standard LMS algorithm.

For this reasons, the LMS convergence analysis becomes an important aspect to determine stability of a LMS system and receives attention of many researchers. A random search for "convergence analysis of LMS' in IEEE database for the period between 2010 and 2016 returns almost 700 publications.

Various advanced LMS algorithms which are able to converge faster using small μ such as Standard Normalized LMS (NLMS) (Ukte & Kizilkaya, 2016), Marr Function Bases Variable Step Size (MFVSSLMS) (Lu, Feng, & Long, 2013), Variable Step Size (VSS) (Y. Zhang, Chambers, Wang, Kendrick, & Cox, 2007), Sign Error LMS (SELMS) (Sharma, Asery, Sunkaria, & Mittal, 2016) and Function Control Variable Step Size (FCVSS) (Turan, Salman, & Eleyan, 2016), to name few was studied and proposed. Some of these techniques are summarized in Table 2.4.

Although the advanced LMS techniques has successful improved the convergence issue however their usage to determine TVAR coefficient is rare and limited. Most advanced LMS techniques are developed as noise cancellation methods; however, further study is needed on their application to estimate $a_j[n]$.

Table 2.2 Advanced LMS Techniques and Their Innovative Terms.

Advanced LMS Methods	Innovation term
NLMS	$\frac{\mu[n]}{\lVert\vec{y}(n)\rVert^2} e^*[n]\vec{y}[n]$
MFVSS-LMS)	Marr function is defined as: $y = 1-(1-x^2)e^{-x^2/2}$ adopted to construct variable step size as: $\mu[n] = 1-(1-e[n]^2)e^{-e[n]^2/2}$
SFVSS-LMS	Sigmoid function is defined as: $f(x) = 1/1+e^{(-x)}$ And variable size step size is defined as: $\mu[n] = \beta\left(\frac{1}{1+e^{(-\alpha\lvert e[n]\rvert)}} - 0.5\right)$ and a modified version is defined as $\mu[n] = \beta\left(\frac{1}{1+e^{(-\alpha(e(n)-\gamma))}} - \frac{1}{1+e^{(-\alpha(e(n)+\gamma))}}\right)$ where β, α and γ are constants
SE-LMS	Innovation term is defined based on *signum* function as: $\mu[n]\operatorname{sgn}[e^*[n]\vec{y}[n]]$
FCVSS-BLMS	Signal, $x[n]$ is subdivided into L blocks of $\vec{X}[n] = [x[nL], x[nL+1]\cdots x[nL+L-1)]]^T$ Vectors of desired output is defined as $\vec{d}[n] = [d[nL], d[nL+1], \cdots, d[nL+L-1]]^T$ And error, $\vec{e}[n]$ as: $\vec{e}[n] = [e[nL], e[nL+1], \cdots e[nL+K-1]]^T$ computed from $\vec{e}[n] = \vec{d}[n] - \vec{X}[k]\vec{a}[k]$

	coefficient update for BLMS is $\vec{a}_j[n] = \vec{a}_j[n-1] + \mu_B \vec{X}[n]\vec{e}[n]$ FC-VSSBLMS is further defined as $\vec{a}_j[n] = \vec{a}_j[n-1] + \mu_B[n]\vec{X}[n]\vec{e}[n] - \rho_B[n]\,\mathrm{sgn}[\vec{a}[n]e^{-\lambda\lvert a[n]\rvert}$

The second AM, RLS is an alternative to LMS which is proposed to be independent from μ. In RLS the objective is to minimize a error based cost function (weighted linear least square cost function) which is defined as.

$$e[n] = \sum_{i=0}^{n} \lambda^{n-i} \lvert e[i] \rvert^2 \qquad (2.8)$$

And TVAR coefficient update for RLS is given as (Rangayyan, 2003):

$$\vec{a}_j[n] = \vec{a}_j[n-1] + \vec{K}[n]e^*[n] \qquad (2.9)$$

$\vec{K}[n]$ is gain vector matrix defined as:

$$\vec{K}[n] = \frac{\lambda^{-1}\vec{P}[n-1]\vec{\phi}[n]}{1 + \lambda^{-1}\vec{\phi}^T[n]\vec{P}[n-1]\vec{\phi}[n]} \qquad (2.10)$$

where λ is a variable known as forgetting factor; defined between $0 < \lambda < 1$, while $\vec{P}[n]$ is the inverse correlation matric of the input signal. $\vec{P}[n]$ and $\vec{K}[n]$ have dimension of p X p and p X 1 respectively. When compared to standard LMS algorithm, the standard RLS algorithm convergence faster at the cost of increased computation time (Dhiman, Ahmad, & Gulia, 2013) as matrixes $\vec{P}[n]$ and $\vec{K}[n]$ are to be solved in an iterative process. However, the standard RLS algorithm is not suitable for processing TV systems as the rate of convergence degrades with large amount of estimation error when applied to NSS. This inherited limitation of standard RLS is further solved by introducing Variable Forgetting Factor (VFF) (Cho, Kim, & Powers, 1991; Leung & So, 2005; Matsuzaki, 2013). More advanced RLS method such as gradient based VFF (Leung & So, 2005), Logsig Function VFF (J. Wang, 2009), Sign Function VFF (Sukh & Benja, 2014) and Local Polynomial VFF (Z. G. Zhang, Hung, & Chan, 2011) , were also introduced to extend the application of RLS in analyzing NSS.

From the computational aspect, the standard RLS requires $o[N^2]$ computation steps (A. Wu & Liu, 1996), which can be reduced to $o[N]$ by adopting advanced RLS methods such as Split-RLS (A. Wu & Liu, 1996) , Robust Variable Forgetting Factor (VFF) RLS (Paleologu, Benesty, & Ciochiňa, 2008) or a 'QR Decomposition' RLS (Ahmad & Aqil, 2016), to name few. Among these list of improved version of RLS algorithms, two (Cho et al., 1991; Y. J. Chu, Chan, Zhang, & Tsui, 2012) of them has been used to estimate $a_j[n]$ efficiently. The performance analysis of the standard LMS and the standard RLS and their derivatives are summarized in Table 2.3.

Table 2.3 Performance Comparison of LMS and RLS

Adaptive Methods	Strengths	Limitations
LMS	Computation simple. Easy implementation. Track slow varying frequencies unbiased.	Slow convergence. Difficulties in determining step size. Unable to track fast changing frequencies. Stability depending on step size. Noisy.
RLS	Fast convergence. Unsuitable to analyze NSS Track slow varying frequencies. Independent from Eigenvalue factor.	Higher Computation cost. Complicated implementation. Fixed forgetting factor. Performance governed by FF Unable to track fast changing frequencies.
Advanced LMS	Fast convergence. Small step size Easy Implementation.	Application dependent variable step size. Unable to track fast changing

	Track slow varying frequencies. Suppress gradient noise.	frequencies. Eigenvalue factor in innovative term.
Advanced RLS	Fast convergence. Unsuitable to analyze non-stationary signal. Track slow varying frequencies. Independent from Eigenvalue factor.	Medium Computation cost. Complicated implementation. Forgetting factor depends on application. Unable to track fast changing frequencies.

2.4.2 TVAR model coefficients estimation using BF

Instead of using a dynamic adaptive filter model to estimate TVAR coefficient as described in previous section, $a_j[n]$ is now explicitly modeled as summation of weighted BF as:

$$a_j[n] = \sum_{i=0}^{m} a_{ji} f_i[n] \qquad (2.11)$$

where m, a_{ji} and $f_i[n]$ are known as expansion parameters.
a_{ji} is constant coefficient of expansion parameter,
m is the expansion dimension or order of expansion parameter which represents number of basis sequence
$f_i[n]$ is a set of selected BF type.

When compared to AMs, the BF methods are capable of tracking all type of frequencies either it is fast, slow or mixed slow-fast frequencies. Furthermore, BF has reduced the number of parameter need to track $a_j[n]$ (G. R. S. Reddy et al., 2014). In addition to that, BF methods are also independent from VFF and hence, the BF methods are not subjected to Uncertainty Principle.

The idea of using weighted linear combination of known BF was initially introduced by Rao (Rao, 1970) . Since then, TVAR BF approaches have been studied and applied in pure statistics, stochastic process, and many engineering disciplines including modeling biomedical signals (Wacker & Witte, 2013).

Some of early works which emphasize on developing algorithms to estimate TVAR BF coefficient include TV Linear Predictive Coding (LPC) (Hall, V.Oppenheim, & Willsky, 1983) , spectral estimation of TVARMA (Grenier, 1983), Instantaneous Frequency (IF) estimation (Sharnian & Friedlander, 1984). Algorithms developed from above mentioned research work are still been used in TVAR BF applications such as Sodsri (Sodsri, 2003) who introduced TVAR forward and backward algorithm based on LPC. And the latest in the list is work by (G. R. S. Reddy et all, 2014) who re-used Hall's covariance method to estimate $a_j[n]$ taking advantage of processor speed which was not available in early 80s to increase the computing time.

From modeling perspective, the TVAR BF method has been successfully used for modeling, analyzing and synthesizing NSS such as speech signal (Harmat, Juntunen, & Kaipio, 2000), ECG (Kumar et al., 2006), IF (Pally, 2009), EEG (Yang Li, Wei, Billings, & Sarrigiannis, 2011), Phonocardiograms (PCG) (Boukhennoufa, Benmahammed, & Benzid, 2012) to quote few research works.

Although TVAR BF has attracted interest among researcher and applied on various types of NSS, however, most of the available works focus on application of available TVAR BF coefficient estimation algorithms, while developing an algorithm to estimate its coefficient is one rare study with few researches dealt with this issue (Yang Li, Wei, & Billings, 2011; R. S. Reddy & Rao, 2015).

2.4.2.1 TVAR BF Modeling

A p^{th} order TVAR model is formulated as in Equations (2.1) and (2.2). Expansion of $a_j[n]$ into sum of weighted BF is given in Equation (2.11). Substituting Equation (2.11) into Equation (2.1), yields:

$$y[n] = -\sum_{j=1}^{p}(\sum_{i=0}^{m} a_{ji}[n] f_i[n]) y[n-j] + \sigma\ [n] \tag{2.12}$$

While, an estimate for $y[n]$ is given as:

$$\hat{y}[n] = -\sum_{j=1}^{p}(\sum_{i=0}^{m} a_{ji}[n] f_i[n]) y[n-j] \tag{2.13}$$

Using Equation (2.12), a problem which was TV is now reduced to time invariant (TIV) by using the constant expansion coefficients a_{ji} which are obviously independent from time as in Equation (2.13).

By adopting the BF approach, computation of TVAR parameters is now to estimate not the TVAR coefficients $a_j[n]$, but series expansion coefficients. And it involves selecting a proper value for p , m and $f_i[n]$ using an efficient algorithm such. to ensure the system is not over parameterized or under parameterized.

From Equation (2.11) list of $a_j[n]$ can generated in terms of a_{ji} and $f_i[n]$. For example, for arbitrary value of n=4, p=4, m=3, list $a_j[n]$ for $1 \le j \le p$ is given as:

$$
\begin{aligned}
&a_1[4] = a_{1,0} f_0[4] + a_{1,1} f_1[4] + a_{1,2} f_2[4] + a_{1,3} f_3[4] \\
&a_2[4] = a_{2,0} f_0[4] + a_{2,1} f_1[4] + a_{2,2} f_2[4] + a_{2,3} f_3[4] \\
&a_3[4] = a_{3,0} f_0[4] + a_{3,1} f_1[4] + a_{3,2} f_2[4] + a_{3,3} f_3[4] \\
&a_4[4] = a_{4,0} f_0[4] + a_{4,1} f_1[4] + a_{4,2} f_2[4] + a_{4,3} f_3[4] \\
&\vdots \qquad\qquad\qquad \vdots \qquad\qquad\qquad \vdots \\
&a_p[4] = a_{p,0} f_0[4] + a_{p,1} f_1[4] + a_{p,2} f_2[4] + a_{p,m} f_m[4] \qquad (2.14)
\end{aligned}
$$

Hence, the Equation (2.13), which estimates the $\hat{y}[4]$ is now represented as:

$$
\begin{aligned}
\hat{y}[4] = -1 * [\\
&(a_{1,0} f_0[4] + a_{1,1} f_1[4] + a_{1,2} f_2[4] + a_{1,3} f_3[4]) y(4-1) \\
&+ (a_{2,0} f_0[4] + a_{2,1} f_1[4] + a_{2,2} f_2[4] + a_{2,3} f_3[4]) y(4-2) \\
&+ (a_{3,0} f_0[4] + a_{3,1} f_1[4] + a_{3,2} f_2[4] + a_{3,3} f_3[4]) y(4-3) \qquad (2.15) \\
&]
\end{aligned}
$$

From Equations (2.14) and (2.15), we now can represent $a_j[n]$ and $\hat{y}[n]$ in matrix format. Furthermore, by re-defining the BF, TVAR coefficients, input signal into matrix format as in Equation (2.16), the p^{th} order TVAR BF process from Equation (2.12) is now can represented in a compacted matrix as in Equation (2.17)

$$
\begin{aligned}
&\vec{F}[n] = [f_0(n), f_1(n), \cdots, f_m(n)] \\
&\vec{c}_{p,m} = \begin{bmatrix} a_{1,1}, a_{1,2}, \cdots, a_{1,m} \\ \vdots \quad \vdots \quad \vdots \\ a_{p,1}, a_{p,2}, \cdots, a_{p,m} \end{bmatrix} \\
&\vec{x}_j[n] = y[n-j]\vec{F}[n] \\
&\vec{X}_n = [\vec{x}_1(n), \vec{x}_2(n), \cdots, \vec{x}_p(n)] \qquad (2.16)
\end{aligned}
$$

Then Equation (2.12) becomes:

$$
y[n] = \vec{X}_n \vec{c}_{p,m}^{\,T} + \sigma[n] \qquad (2.17)
$$

Equation (2.17) is in the form of standard linear regression model; therefor $\vec{c}_{p,m}$ can be computed by minimizing the total square prediction error using linear least square methods (Hall et al., 1983). Also from the view point of system identification, the terms $\vec{c}_{j,i}$, $\vec{X}_n$ and $\vec{x}_j[n]$ could be regarded as system parameters, inputs and output of a system respectively, thus now allowing the TVAR BF to be modelled by an ANN structure.

Three fundamental issues need to be addressed when modelling a NSS with TVAR BF methods which are:

I. To formulate an effective algorithm to estimate the $a_j[n]$.

II. To select an appropriate model order determination technique.

III. To select a suitable BF to represent the signal.

These issues will be addressed in the next subsections.

2.4.2.2 TVAR BF Coefficient Estimation Methods

To determine a_{ji} for single realization of n, a method based on LPC (Hall et al., 1983) , as implemented by (Pally, 2009; G. R. S. Reddy & Rameswar Rao, 2014) is presented in this section.

In this method, Equation (2.12) and Equation (2.13) is used to derive a Total Squared Prediction Error (TSFE) from TV Linear Prediction Error (LPE) as below:

A TV LPE is defined as:

$$\begin{aligned} v[n] &= y[n] - \hat{y}[n] \\ &= y[n] + \sum_{j=1}^{p}\sum_{i=0}^{m} a_{ji} f_i[n] y[n-j] \end{aligned} \tag{2.18}$$

And the TSPE is specified as :

$$\varepsilon_p[n] = \sum_{\tau} |v_n|^2 = \sum_{\tau} \left| y[n] + \sum_{j=1}^{p} \sum_{i=0}^{m} a_{ji} f_i[n] y[n-j] \right|^2 \qquad (2.19)$$

Then, the criteria proposed by (Hall et al., 1983) is adopted whereby a_{ji} which minimizes $\varepsilon_p[n]$ can be found by setting the gradient of $\varepsilon_p[n]$ with respect to $a^*_{l,g}$ as zero.

$$\frac{\partial \varepsilon_p[n]}{\partial a^*_{l,g}[n]} = \sum_{\tau} \frac{\partial v[n] v^*[n]}{\partial a^*_{l,g}[n]} = \sum_{\tau} v[n] \frac{\partial v^*[n]}{\partial a^*_{l,g}[n]}$$

$$\{l = 1,2,\cdots,p; g = 0,1,2,\cdots,m\} \qquad (2.20)$$

where $v^*[n] = y^*[n] + \sum_{l=1}^{p} \sum_{g=0}^{m} a^*{}_{ji} f_g{}^*[n] y*[n-l]$

And the derivative of $v^*[n]$ with respect to $a^*_{l,g}$ is $f^*_g[n] y^*[n-l]$ consequently. Therefore, Equation (2.20) becomes

$$\sum_{\tau} v[n] f^*_g[n] y^*[n-l] = 0 \qquad (2.21)$$

Substituting for $v[n]$ from Equation (2.18) to (2.21), we arrive as:

$$\sum_{\tau} (y[n] + \sum_{j=1}^{p} \sum_{i=0}^{m} a_{ji} f_i[n] y[n-j]) f^*_g[n] y^*[n-l] = 0$$

$$\sum_{\tau} y[n] f^*_g[n] y^*[n-l] + \sum_{j=1}^{p} \sum_{i=0}^{m} a_{ji} \sum_{\tau} f_i[n] y[n-j] f^*_g[n] y^*[n-l] = 0 \qquad (2.22)$$

Next, a correlation function is defined as,

$$c_{i,g}(l, j) \equiv \sum_{\tau} f_i[n] y[n-j] f^*_g[n] y^*[n-l] \qquad (2.23)$$

Equation (2.23) is substituted back in Equation (2.22) and yields:

$$\sum_{j=1}^{p} \sum_{i=0}^{m} a_{ji} c_{ig}(l, j) = -c_{0,g}(l,0) \qquad (2.24)$$

For the correlation function, $c_{i,g}(l,j)$, the subscripts i and g refers to set of time function, while (l,j) refer to signal samples. Equation (2.24) represents a system of p $(m+1)$ linear equations that must be solved to estimate a_{ji} in each realization of n

The above system of linear equations can be expressed compactly in a matrix form by following vectors :

$$\vec{a}_i = [a_{1i}\ a_{2i}\ a_{3i}\ \cdots a_{pi}]^T \qquad i = 0,1,2,\cdots,m \tag{2.25}$$

and the matrix

$$\vec{C}_{i,g} = \begin{bmatrix} c_{i,g}(1,1) & c_{i,g}(1,2) & \cdots & c_{i,g}(1,p) \\ c_{i,g}(2,1) & c_{i,g}(2,2) & \cdots & c_{i,g}(2,p) \\ \vdots & \vdots & & \vdots \\ c_{i,g}(p,1) & c_{i,g}(p,2) & \cdots & c_{i,g}(p,p) \end{bmatrix} \quad 0 \le i \le m \quad 0 \le g \le m \tag{2.26}$$

and a column vector d_i for Equation 2.26 is described as:

$$\vec{d}_i = [c_{0,i}(1,0)\ c_{0,i}(2,0)\ \cdots c_{0,i}(p,0)]^T \quad 0 \le i \le m \tag{2.27}$$

From Equation (2.23), it is clear that $c_{i,g,} = c_{g,i} = c_{i,g}^T$ so that linear systems as in Equation (2.24) is now rewritten in matrix format as:

$$\begin{bmatrix} \vec{\mathrm{C}}_{0,0} & \vec{C}_{0,1} & \cdots \vec{C}_{0,m} \\ \vec{C}_{1,0} & \vec{C}_{1,1} & \cdots \vec{C}_{1,m} \\ \vdots & \vdots & \vdots \\ \vec{C}_{m,0} & \vec{C}_{q,1} & \cdots \vec{C}_{m,m} \end{bmatrix} \begin{bmatrix} \vec{a}_0 \\ \vec{a}_1 \\ \vdots \\ \vec{a}_m \end{bmatrix} = -\begin{bmatrix} \vec{d}_0 \\ \vec{d}_1 \\ \vdots \\ \vec{d}_d \end{bmatrix} \tag{2.28}$$

or

$$\vec{C}\vec{a} = -\vec{d}$$

Some important notes regarding Equation (2.28) derived above are as follows:

a) Equation (2.28) is computed for each sample, *n*, in order to allow the TVAR coefficient to vary in time.

b) The set of parameters, $a_{j,i}$ are elements of $\vec{a}$, computed by solving Equation (2.28) using either a Correlation Method (CORM), Covariance Method (COM) or a Least Square Method (LSM) as shown in Figure 2.5 .

In Equation (2.21), internal τ is left unspecified, to be specified by the technique adapted. To determine TVAR BF expansion coefficient using the CORM, the input signal, $y[n]$ is assumed to be known over a finite data sequence $[0, N-1]$ and can be extended by zero-padding. In this case, the range in Equation (2.23) is set as $\tau = [0, N-1+p]$ and $c_{i,g}(l, j)$ is referred as the generalized correlation equation. The matrix , $\vec{C}$ in the correlation method is found to have Hermitian-Toeplitz structure which can be easily computed using a Levinson-Durbin algorithm (Abiodun Musa Aibinu, 2010; Pally, 2009) with number of computation $O(p^2(m+1)^2)$.

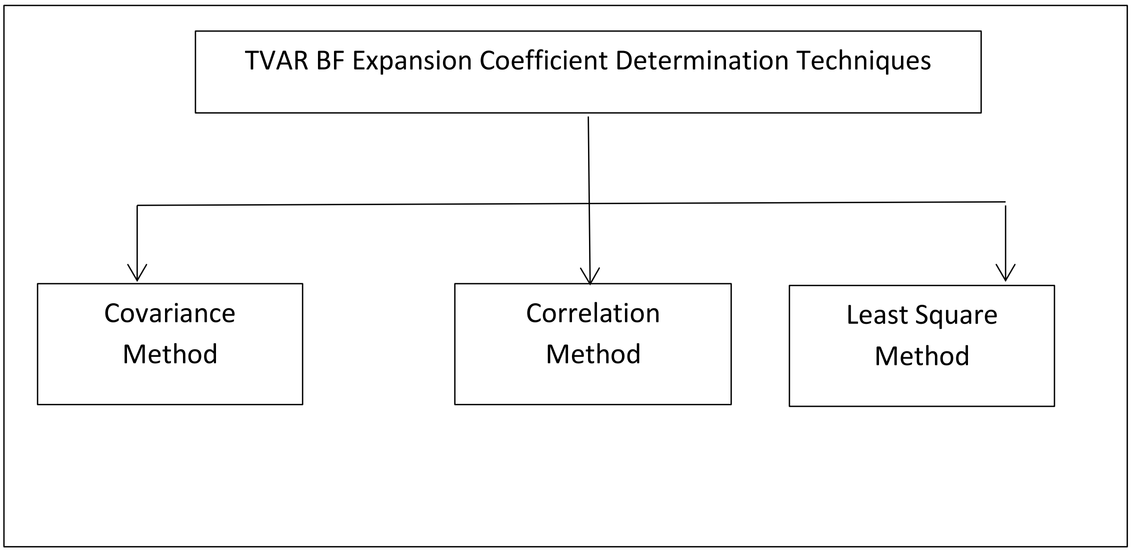


Figure 2.5 TVAR BF Expansion Coefficient Estimation Techniques

On the other hand, a COM does not make any assumption on the data outside $[0, N]$ and thus the range is defined for $\tau = [0, N-1]$. In this case, $c_{i,g}(l, j)$ is now referred as the generalized covariance equation. The matrix $\vec{C}$ in the covariance method has Hermitian Block Hankel structure, whereby application of Levinson-Durbin algorithm is not suitable. Nevertheless a fast covariance algorithm is available in which $\vec{C}$ is inverted using a Trench algorithm (TA), an Akaike Algorithm (AA) or a Wax-Kailath Algorithm (WKA). Complete reference to these methods are discussed in detail by (Pally, 2009; G. R. S. Reddy & Rameswar Rao, 2014).

Alternatively, Equation (2.28) also can be solved by Gaussian Elimination (GE) , which requires $O(p^3(m+1)^3)$. This is not preferred method due to its huge computation and uncertain results which ends in poor estimation of $a_{j,i}$ (G. R. S. Reddy & Rameswar Rao, 2014). Performance analysis on TVAR BF expansion coefficient determination techniques is presented in Table 2.4 and its implementation flowchart is illustrated in Figure 2.6.

Table 2.4 Performance Evaluation of TVAR BF Expansion Coefficient Determination Techniques (Abiodun Musa Aibinu, 2010)

Parameter	Covariance	Correlation
Stability	Not guaranteed	Always stable
Algorithm Procedure	Solving set of linear equation begins with inverse of $\vec{C}$	Minimize forward prediction error.
Computation	Efficient	Inefficient
Implementation	Complex	Direct
Advantages	High Resolution for short data	Higher Resolution for long data
Disadvantages	May produce instable system	Short data may have resolution issues

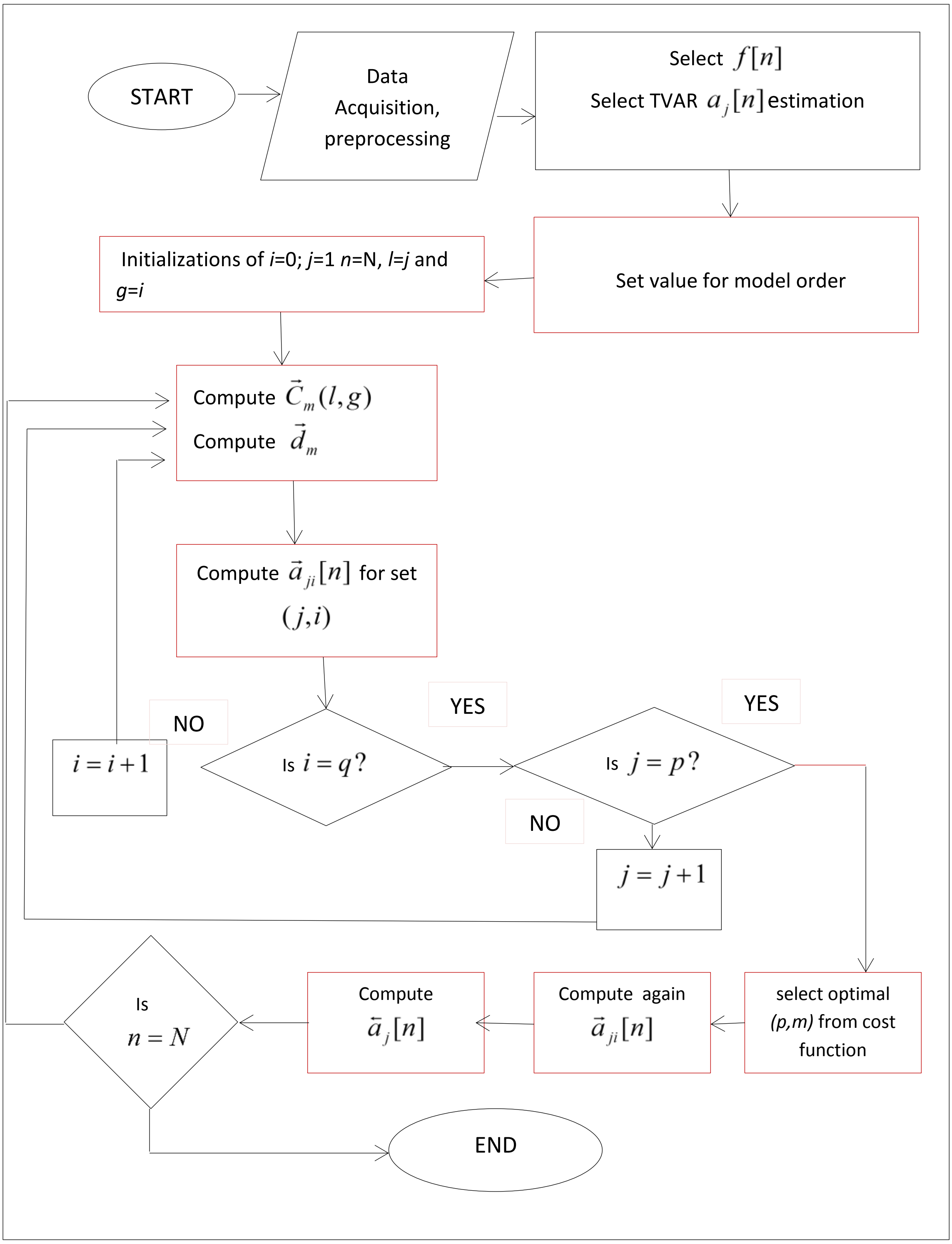


Figure 2.6 TVAR BF Modeling Technique Implementation Flowchart

2.4.2.3 TVAR BF Model Order Determination Techniques

Basically, the model orders (*p* and *m*) determine the number of TVAR coefficient that should be used. The orders can be any number as desired, however it should be as accurate as possible, thus playing a critical role in representing a signal. An over-parametrized model orders may introduce noise, false spectral peaks or artifacts, while on the other hand under-parameterized model orders will not accurately reveals the information in an input signal by producing low resolution spectral plots, hence is not reliable to any extract information from the input signal.

For this reasons, the determination of accurate model orders becomes a significant issue. The model order selection processes are as difficult as computing $a_j[n]$. As a matter of fact, the $a_j[n]$ cannot be estimating without computing the accurate model orders earlier.

The optimal model orders are generally unknown and it is necessary in practice to evaluate many candidates of model orders before the optimal value is selected using certain criteria. Such approach is obviously exhaustive, iterative and consumes huge memory with increased computation time.

There are many criteria for determining the model orders for an TVAR BF process, which can be grouped as follows:

a) Criteria developed for AR

   Methods under this category are exclusively developed for stationary AR whereby the concern is the computation for *p* only. Among commonly used traditional AR model order methods are Final Prediction Error (FPE), Akaike Information Criteria (AIC), Minimum Description Length MDL) and

Criterion Autoregressive Transfer Function (CAT). The best model orders are selected based on criteria defined in those methods through an exhaustive search between $1 < j < p$ where p is the possible largest model order, usually set as *N/4*.

In a study comparing these conventional model order methods, (W. Read, 1990) concluded that since these methods were derived using large sample; hence their performance on short data may vary. Furthermore it also reported that generally these algorithms give low model orders for sinusoidal with noise added , therefore their application on NSS give mixed results. A list of the common traditional methods is presented in Table 2.5.

Table 2.5 AR Model Order Determination Techniques

Criteria	Description
Final Prediction Error (FPE)	$FPE(j) = \sigma_j^2 \dfrac{N+j}{N-j}$ $Optimum(p) = \min(FPE(j))$
Akaike Information Criteria (AIC)	$AIC(j) = N\ln(\sigma_j^2) + 2j$ $Optimum(p) = \min(AIC(j))$
Minimum Description Length(MDL)	$MDL(j) = N\ln(\sigma_j^2) + j\ln(N)$ $Optimum(p) = \min(AIC(j))$
Criterion Autoregressive Transfer Funcion (CAT)	$CAT(j) = (\frac{1}{N}\sum_{k=1}^{j} v_k^{-1}) + v_j^{-1}$; where $v_k = [\frac{N}{(N-k)}]\sigma_k^2$ $optimum(p) = \min(CAT(j))$

b) Criteria developed for TVAR BF

Criteria for TVAR BF model order determination should include computation for both *p* and *m* orders which is more complicated than criteria developed for stationary AR process. It is to be reminded here that the relationship between *p* and *m* in TVAR BF is not the same as for model orders of ARMA. In case of ARMA, *p* is model order for AR process and *q* is model order for MA process. Therefore the criteria or the cost function of TVAR BF and ARMA are not the same.

As both *p* and *m* are unknown a priori, the general approach is to evaluate various sets of *p* and *m* for a range of $(1 < j < p)$ and $(1 < i < m)$ with $m < p$. The maximum of *(p+m)* evaluated is far less than *N* as not to burden the system with over computation. Again, a cost function is applied on the sets of *(p,m)* to select optimal set.

Obviously this process involves an exhaustive search which will increase the computation time. List of cost function for TVAR BF model orders derived from literatures is listed in Table 2.6. TV abbreviation is added in order to distinguish from TIV method of same algorithm .

c) Genetic Algorithm Based Model Order Determination

In last decade, an interesting and promising development has been observed in model order estimation of AR and ARMA parametric technique (Q. Zhang & Wang, 2008; Jian-jun, Dong-Xiao, & Li, 2009;Saini & Prasad, 2010; Abo-Hammour, Alsmadi, Al-Smadi, Zaqout, & Saraireh, 2012; Dhanwani & Wadhe, 2013; Ervural, Beyca, & Zaim, 2016). Genetic Algorithm (GA), an evolutionary optimization technique inspired from biological evolution process has been successfully applied to determine model orders using several approaches.

Table 2.6 List of TVAR BF Model Order's Cost Function

Criteria	Description
TV AIC (Ki H Chon, Zhao, Zou, & Ju, 2005)	$TVAIC(j,i) = N \log \sigma_{j,i}^2 + 2\left[j\sum_{i=1}^{m} i + 1 \right]$ for each n $optimum(p,m) = \min(TVAIC(j,i))$
TV MML (G. R. S. Reddy et al., 2014b)	$TVMML(j,i) = \frac{j(i+1)+1-N}{2}\log(2\pi\beta) - \frac{1}{2}\log\left\|\sum_{n=0}^{N} Z(n)Z^T(n)\right\|$ $optimum(p,m) = \max(TVMML(j,i))$
TV FPE (Y Li, Liu, Tan, & Chan, 2016) for Gaussian Basis Function	$TVFPE(j,i) = \frac{N+j+1}{N-j-1}\sigma_{j,i}^2 + j\sum_{i=1}^{m} k_i$ Where k_i is number of Gaussian Basis Function $optimum(p,m) = \min(TVFPE(j,i))$
TV MDL (Thonet & Vesin, 1997)	$TVMDL(j,i) = N\ln(\sigma_{j,i}^2) + j(i+1)\ln(N)$ $optimum(p,m) = \min(TVMDL(j,i))$
Bayesian Framework (Y. Chu, 2012)	Developed for Bayesian Framework $p(M_i \mid \tilde{x}) = p(\tilde{x} \mid M_i)p(M_i)(\frac{1}{p(\tilde{x})})$ description of this lengthy criteria is given by (Y. Chu, 2012) , page 87

In one method, Ramasawamy, P. (Palaniappan, 2006) combined a technique known as Fuzzy Artmap (FA) with GA to select appropriate model order for EEG signal, whereby the FA has been used as fitness function of GA. Musa (Aibinu, 2010) computed AR model orders in two different approaches, a model order optimization using GA and secondly AR coefficient optimization using fixed model orders. While Zaer (Hammour et al. 2013) adopted a simple Fitness Function (FF) based on MSE to compute model orders of ARMA using GA and simultaneously computed both ARMA model orders and ARMA coefficients.

From literatures, the following advantages offered by GA based model order technique are deduced:

a. It does not use complex mathematical procedure.
b. Independent from exhaustive search.
c. Global search capability either for linear or nonlinear systems of low or high order.
d. Independent from data size.
e. Simultaneously determine model order and coefficient.
f. Higher accuracy compared to exhaustive statistical search methods.

Most of the GA methods to estimate model order found in are designed for stationary process of AR, while methods to estimate model orders for $a_j[n]$ of NSS signals is very rare. One such work is by Zhang (L. Zhang, Xiong, Liu, Zou, & Guo, 2010) who used GA to optimize p and m of TVAR by adopting a stopping criteria based on maximum-likelihood decision (Eom, 1999).

2.4.2.4 TVAR Basis Functions

As the model orders, the choice of BF, $f_i[n]$ is equally critical and significant as it affects the overall performance of TVAR BF by controlling the smoothness and variation speed of TVAR coefficients. The BF has influences on the number of expansion coefficient, *m.* An overdetermined expansion coefficient not only introduces additional noise but will increase the computation time and thus increases the memory utilized to realize the system.

Two general requirements are to be considered while selecting $f_i[n]$, the first one is the $f_i[n]$ must be independent and secondly it should have non-zero value for $n > 1$ and $f_i n] = 1$ for $n = 0$. Furthermore, there is no constraint imposed and there is also no specific guideline available or explicitly mentioned in literatures on selecting proper $f_i[n]$ to fitting a certain NSS nor a $f_i[n]$ that may suits all kind of NSS.

Numerous BFs were proposed in literatures such as Time Polynomial Series (PS), Fourier Sequences (FS), Legendre Polynomials (LP) , Walsh Functions (WP) , Discrete Prolate Spheroidal (DPS) and Wavelet Transforms (WT) , however as mentioned earlier, none of these solutions is perfect fitting for all kind NSS. List of BF with their mathematical description is summarized in Table 2.7.

Table 2.7 List of Basis Functions

Basis Function	Description
Fourier Basis Function / Trigonometric (Hall et al., 1983)	$f_i[n] = \begin{cases} \cos(\omega in) & i \text{ is even} \\ \sin(\omega in) & i \text{ is odd} \end{cases}$ Where $\omega = \pi / N$
Multiscale Radial Basis Function ,MRBF (Y Li et al., 2016)	$\phi_i \lVert x - c_i \rVert = \exp\left[-\frac{\lVert x - c_i \rVert^2}{2\sigma_i^2}\right]$ $c_i = \frac{i \text{ X } N}{m}$ $scale, \sigma_i^2 = \frac{N^2 \text{X } 2^{-S_i}}{\text{m}}$ where $S_i < 10$
Time Polynomial Basis Function, TFBF (G. R. S. Reddy et al., 2014b)	$f_i[n] = \left[\frac{n-p}{N}\right]^i$ $; i = 0,1,2,\cdots,m; p = p, p+1,\cdots N$

Discrete Cosine Transform , DCT (Eom, 1999b)	$f_i(n) = \alpha(i)\cos\left(\frac{\pi i(2n+1)}{2N}\right)$ *where* $\alpha(i) = \sqrt{\frac{1}{N}}, i = 0$ $\alpha(i) = \sqrt{\frac{2}{N}}, i = 1,2,3\ldots m$
Legendre Polynomial Basis Function, LPBF (G. R. S. Reddy et al., 2014a)	$f_0[n] = 1$ $f_1[n] = \frac{2(n-1)}{N-2} = y$ $f_2[n] = \frac{3y^2-1}{2}$ $f_{i+1}[n] = \frac{((2i+1)yf_i[n]) - (if_{i-1}[n])}{(i-1)}$
Chebyshev Basis Function ,CBF (G. R. S. Reddy et al., 2014a)	$f_i[n] = \cos[i\cos^{-1}(k-1)]$ $where\ k = \frac{2(n-1)}{N-1}$

As a matter of fact, each BF has its own unique characteristic which captures dynamics of similar features with the BF. For instance, Polynomial BF and Fourier BF work well for smooth and slow varying coefficients, while Wavelet BF perform well for signals with abrupt changes (Li, et al., 2011). As each BF poses its own characteristics, different BF applied on same input signals results in different parameter estimation.

For NSS with multiple components and with mixed dynamics, use of a single BF is not suitable and insufficient to capture its changing dynamics. Recent works on TVAR BF for biomedical signals suggests employing multiple sets of BF to increase the ability of TVAR BF model to capture NSS with multiple dynamics. Furthermore, having multiple sets of BF alleviate the problem of determining BF as a priori information.

Reddy (Reddy et al., 2014) used a combination of LPBF and WBF on mixed NSS and showed that multiple set of BF produces lower least square error when compared to single basis function. While Ki Chon (Ki H Chon et al., 2005) have demonstrated combination of LP and WP produces better TFR when tested on noisy NSS with overdetermined model order and simultaneously reducing the model order as well . Combination of LP and WP was also adopted by Zou (Rui Zou, HengLiang Wang, & Ki. H. Chon, 2003) to determine the optimal model order using Optimal Parameter Search Algorithm (OPSA). H.L. Wei (Yang Li, Wei, Billings, & Wei, 2009) used multi wavelet BF on EEG signal and compared the performances to RLS.

2.4.2.5 Comparisons between AM and BF methods

A performance analysis between AM and BF methods are summarized in Table 2.8 below:

Table 2.8 Performance Evaluation of AM and BF methods

Category	Strength	Limitations
AM	LMS • Easy Implementation • Low Computation • Detects slow changing frequencies • Stable • Applicable on both TV and TIV signals • Good resolution on both time and frequency domain RLS • Fast convergence • Lower Gradient Error • Detects slow changing frequencies • Stable	LMS • Detect slow varying only • Dependent on step size(SS) • Fast convergence produces high gradient noise • Stability is uncertain for short data • Subjected t uncertainty principle RLS • Detect slow varying only • Complex implementation • Not suitable to TV signals • Dependent on forgetting factor (FF)

	• Stationary signals only	
Modified AM	• Independent from FF and SS • Applicable on both TV and TIV Signal • Fast convergence with low gradient error • High resolution • Efficient on long data	• Detect slow varying only • Complex implementation • High Computation • Slow Convergence Rate in some applications • Application dependent • Iterative and recursive • Large number of coefficients • Unstable in some applications
BF	• Detects variety of signal slow and fast varying • Higher resolution in both time domain and frequency domain • Not subjected to uncertainty principle • Works best for short data or long data • Stable if model parameters are estimated properly • More parsimony signal representation • Improved accuracy in signal representation	• Model order dependent • Accuracy of basis function depend on basis function • High number of coefficients • Lower Computation compared to Adaptive • Recursive and Iterative • Basis function is dependent on signal
Hybrid Techniques	• Inherits advantages of basis function and modified adaptive methods	• High number of coefficients • Recursive and Iterative • High computation • complex Implementation • application dependent

2.5 REVIEW OF RELEVANT STUDIES

The number of TVAR related research papers starting from 1970 that has been studied and their comments are presented in Table 2.9.

Table 2.9 Review of Relevant Studies on TVAR Coefficients Estimation Methods

Year	Article	Method	Comments
1970	The Fitting of Non-stationary Time-Series Models with Time-Dependent Parameters (Rao, 1970)	BF	• Pioneer work on BF • BF 1: Cosine Function • BF 2: Parzen Polynomial • TVAR coefficients using Spectral Approach and Weighted Least Square • Model Order : Not Available
1983	Time-Varying Parametric Modeling of Speech (Hall et al., 1983)	BF	• BF : Fourier Based • Covariance based LPC • Model Order : AIC
1984	Time Varying Autoregressive Modeling of a class of non-stationary signals (Sharnian & Friedlander, 1984)	BF	• BF: Time Basis Function • TVAR coefficient : Correlation solved with modified Yule Walker equation • Model order fixed to 6
1983	Time-Dependent ARMA Modeling of Non-Stationary Signal (Grenier, 1983)	BF	• TVAR : Levinson • TVMA : infinite-order AR process • BF: Discrete Prolate Speroidal Sequence • Model order : Not mentioned
1996	Bayesian approach to parameter estimation and interpolation of time-varying autoregressive processes using Gibbs Sampler (Rajan, Rayner, & Godsill, 1996)	BF	• BF : 5 different sets of Fourier based • Model order : not mentioned • TVAR coefficients : Bayesian approach using Gibbs Sampler
1993	Time –Varying System Identification and model Validation Using Wavelets (Tsatsanis, Giannakis, & Member, 1993)	BF	• Model Order : AIC • BF : Wavelet Basis • Least Square Method
1997	Stationarity assessment with time-varying autoregressive modeling. (Thonet & Vesin, 1997)	BF	• BF : Chebyshev, Hermitte, Legendre • Model order : Modified MDL • TVAR : Covariance LPC
1997	Non-stationary Spectral estimation Based on Robust Time Varying AR Model Excited by t-distribution Process (Sanubari & Keiichi Tokuda, 1997)	Basis Function	• BF : Cosine Function • Model Order : Modified Maximum

			Like hood for p , while for q is fixed . • TVAR coefficient : newton Raphson
1999	Analysis of Acoustic Signatures from Moving Vehicles Using Time-Varying Autoregressive Models (Eom, 1999a)	Basis Function	• BF : Discrete Cosine Function • Covariance LPC • Model Order : Modified log-Likehood function
1998	Adaptive AR Modeling of Non-Stationary Time Series by Means of Kalman Filtering (Arnold, Miltner, Witte, Bauer, & Braun, 1998)	Kalman Filtering	A Time invariant Autoregressive approach by means of Kalman Filtering to determine coefficients of AR process.
2003	Research of Parameter Estimation of Time-Varying AR Model (W. Wang & Wang, 2003)	Adaptive Method, Basis Function and Wigner Distribution	Paper is too simple such proposed method is not elaborated in detail. it was mentioned for TVAR, BF Legendre is used
2005	Multiple Time-Varying Dynamic Analysis Using Multiple Sets of Basis Function (Ki H Chon et al., 2005)	Basis Function	• BF : multiple set orthonormal using Gram-Schmidt • TVAR : TVOPS • Model order
2003	A Robust Time-Varying Identification Algorithm Using Basis function (Rui Zou et al., 2003)	Basis Function	• Model Order : Optimal Parameter Search Algorithm • BF : Legendre and Walsh • TVAR Coefficient : TV-OPS
2008	Identification of Time-Varying Autoregressive Systems Using Maximum a Posteriori Estimation. (Hsiao, 2008)	Monte Carlo Method	Posterior Probability derived from Bayes Theorem
2011	Adaptive time-frequency analysis based on autoregressive modeling (Costa & Hengstler, 2011)	Adaptive Method	• Model Oder : Predictive Least Square (PLS) • Advance RLS • Variable Forget factor
2009	Identification of Time-Varying Systems Using Multiwavelet Basis Function (Y. Li, Wei, & Billings, 2009)	Hybrid Method	Equations are modeled using TVAR Basis function, where BF is multiwavelet. Modified LMS is then applied on the derived multiwavelet t compute for TVAR. Model order is estimated with Modified Generalized Cross-Validation (mGCV)
2012	A new recursive algorithm for time-	Adaptive	• Advance RLS known

	varying autoregressive (TVAR) model estimation and its application to speech analysis (Y. J. Chu et al., 2012a)	Method	as QRRLS • Variable Forgetting Factor • Model Order not mentioned
2013	Uncertainty Modeling and Prediction for Customer Load Demand in Smart Grid (D. Li & Jayaweera, 2013)	Basis Function	• Model Order : Time Varying partial autocorrelation function (TVPAF) • TVAR coefficient : Correlation function, Yule walker • Trigonometric Basis Function
2013	Identification of Time Varying System Using Recursive Estimation Approach and Wavelet Based Recursive Approach (Mate & Patil, 2013)	Adaptive Method Basis Function	• Both Methods was used and compared • Adaptive method : Normalized LMS • Basis Function : multi-wavelet • Model Order : not mentioned
2013	Time Varying Autoregressive Moving Average Models for Covariance Estimation (Wiesel, Bibi, & Globerson, 2013)	Direct	Modified Covariance method applied directly on TVARMA
2014	Instantaneous Frequency Estimation Based on Time-varying Auto Regressive Model And Wax-Kailath Algorithm (G. R. S. Reddy & Rameswar Rao, 2014)	Basis Function	• Model Order : TV Maximum Likehood Estimation (MMLE) • TVAR Coefficient : Covariance method solved with Kax-Kailath Algorithm • BF: Time Polynomial, Legendre ad Chebyshev
2014	Performance Analysis of Basis Function in TVAR Model (G. R. S. Reddy et al., 2014b)	Basis Function	• BF : All type of BF is been tested • TVAR Coefficient : Correlation (YW) • Model order :TVMLE
2014	Non-stationary Signal Prediction Using TVAR Model(G. R. S. Reddy & Rao, 2014)	Basis Function	• BF: Discrete low order cosine function. • TVAR Coefficient : Correlation • Model order : TVMLM
2014	Non-stationary Signal Analysis using TVAR Model(G. R. S. Reddy et al., 2014a)	Basis Function	• BF : LP, Walsh • TVAR Coefficient: Correlation • Model Order TVMLM
2015	Time Varying Autoregressive Model Using Multi-Wavelet Basis Function (R. S. Reddy & Rao, 2015)	Basis Function	• BF : Multi-Wavelet Function TVAR Coefficient : Orthogonal Least

			Square
2015	Efficacy of Adaptive Directed Transfer Function for Neural Connectivity Estimation of EEG Signal During Meditation (Shaw & Routray, 2015)	Adaptive Method	• TVAR Coefficient : Kalman Filtering , solved using Fixed Forgetting Factor RLS • Model order AIC and Scwarz Bayesian Information Criteria
2015	Instantaneous Frequency Estimation of Multi-Component Non-stationary signals using Fourier Bessel Series and Time-Varying Auto Regressive Model (G. R. S. Reddy & Rao, 2015)	Basis Function	• TVAR Coefficient : Fourier Bessel Series TVAR • BF: DCT • Model order :TV MLE •
2015	Adaptive Spectral Estimation of Non-Stationary Biomedical Signals on Autoregressive Modeling and Kalman Filtering (Ouelli, Elhadadi, Aissaoui, & Bouikhalene, 2015)	Adaptive Method (Kalman filtering and RLS)	• TVAR : Kalman Filtering and RLS • Model Order : fixed

2.6 ARTIFICIAL INTELLIGENT SYSTEM (AIS)

The problem of TVAR model coefficient determination possesses a lot of challenges for TVAR implementation into real life data and applications. The reason is the TVAR BF's dependencies on the model orders, the BFs and the expansion coefficient determination techniques. Furthermore, the existing TVAR BF algorithms are problem specific, while the BFs are unknown values. These facts lead to many uncertainties.

One way ahead in this interesting field is to adopt an intelligent method to avoid manual interventions as to assist in decision making process so the TVAR BF system becomes independent, optimized and consistent. An AIS is an effective approach for tackling the above mentioned complexities. AIS such as an ANN, GA, Expert System (ES), Fuzzy Logic System (FLS), Evolutionary Optimization Algorithms (EOA) or a Hybrid Intelligent Systems (HIS) has been long used in signal processing fields as a intelligent decision making tool . From our understanding, an AIS based system can be extended and integrated with TVAR to enhance its capacity and to overcome current challenges in implementing TVAR without opting for TVAR AM or TVAR BF methods.

A performance analysis of AIS components is given in Table 2.10 (Aibinu, 2010). Table 2.10 also suggests that AIS which comprises of ANN and GA could be an appropriate tool for development of hybrid method to determine TVAR coefficients. Musa et al. (Aibinu, Salami, & Shafie, 2012; Aibinu, 2010) developed an hybrid AIS system using ANN and GA to compute AR coefficient , but however on contrary, from our extensive search on literatures, such approach for TVAR is very rare. Since TVAR inherits all the advantages and disadvantages of an AR process, the information provided is applicable for TVAR as well

Table 2.10 Performance Analysis of AIS (Aibinu, 2010)

Criteria	FL	ANN	GA	Tentative Choice
Mathematical Modelling	Slightly Good	Bad	Bad	Already exist
Learning Ability	Bad	Good	Slightly Good	ANN
Knowledge Representation	Good	Bad	Slightly Bad	Already exist
Non-linearity	Good	Good	Good	ANN
Initialization Capability	Bad	Slightly Good	Good	GA
Optimization	Bad	Slightly Good	Good	GA
Fault Tolerance	Good	Good	Good	ANN
Uncertainty Tolerance	Good	Good	Good	ANN

2.6.1 Motivation for ANN

ANN is a general mathematical paradigm which is used as global searching technique by emulating the learning process of human brain (Hu & Hwang, 2002). While, the conventional techniques, either based on parametric methods or non-parametric methods uses algorithmic approach whereby a problem is solved by a set of working rules. Hence, the set of working rules limits the capability of conventional techniques to solve for problems that in known only and not flexible enough to perform well outside its defined domain.

ANN is an efficient computing model with unique characteristics that process problems involving non-linear and complex data especially when the underlying data relationships are not known, imprecise, random or noisy. It has found rigorous implementation in various fields such as target recognition, image compression, pattern classification, clustering, classification, approximation, forecasting, optimization and in control applications (Jain & Mao, 1996).

Basically, a typical ANN is a massive parallel system with largely interconnected simple processing elements known as nodes. Nodes are arranged in layers and joined together by interconnecting synaptic weights, ω, to form a network to facilitate a distributed computing system. An ANN network comprises different types of layers which are input layer, hidden layers, and an output layer. Data are fed into ANN through input nodes, processed in hidden layers and finally released out from the network from output layer. The hidden layers consists one or more processing nodes known as artificial neuron. The most widely used neuron model is McCulloch and Pitts s' artificial neuron as shown in Figure 2.7 (Hu & Hwang, 2002) which consists of two working units, a net function and an activation function.

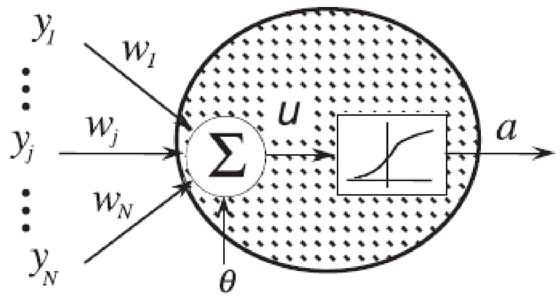


Figure 2.7 McCulloch and Pitts Artificial Neuron Model (Hu & Hwang, 2002)

Weights are 'interconnecting' factor from input layer to hidden layer and play a major role in accuracy of ANN. Optimal weights are obtained by ANN training where the input data is repeatedly processed in an unique computation to produce an output that is closed to a desired value. Performance is improved over time iteratively by updating the synaptic weights until a desired value is attained.

Through this training process, the ANN network is able to learn. The word 'learn' here describes process of updating the network architecture and connection weights so that ANN efficiently performs a specific task. This characteristics of ANN makes them attractive as ANN able to learn underlying rules such as input-output relationship and gives distant advantages to ANN (Jain & Mao, 1996) which is not subjected to a defined working rule. Therefore, the ANN system architecture is divided into two parts, namely the ANN structure, which is the network of layers and the ANN training algorithm, the learning process .

An ANN structure or topology could be a single layer Feed Forward Network (FFN), Multilayer Feedforward Networks (MFFN) or Recurrent Networks (RN). A Single Layer Perceptron (SLP) is a single layer FFN with one input layer and output layer, while a Multilayer Perceptron (MLP) consists of input, hidden and output layer. (Mandry et al., 2002). A MLP has one or more artificial neurons in hidden layer for processing data and shown in Figure 2.8.

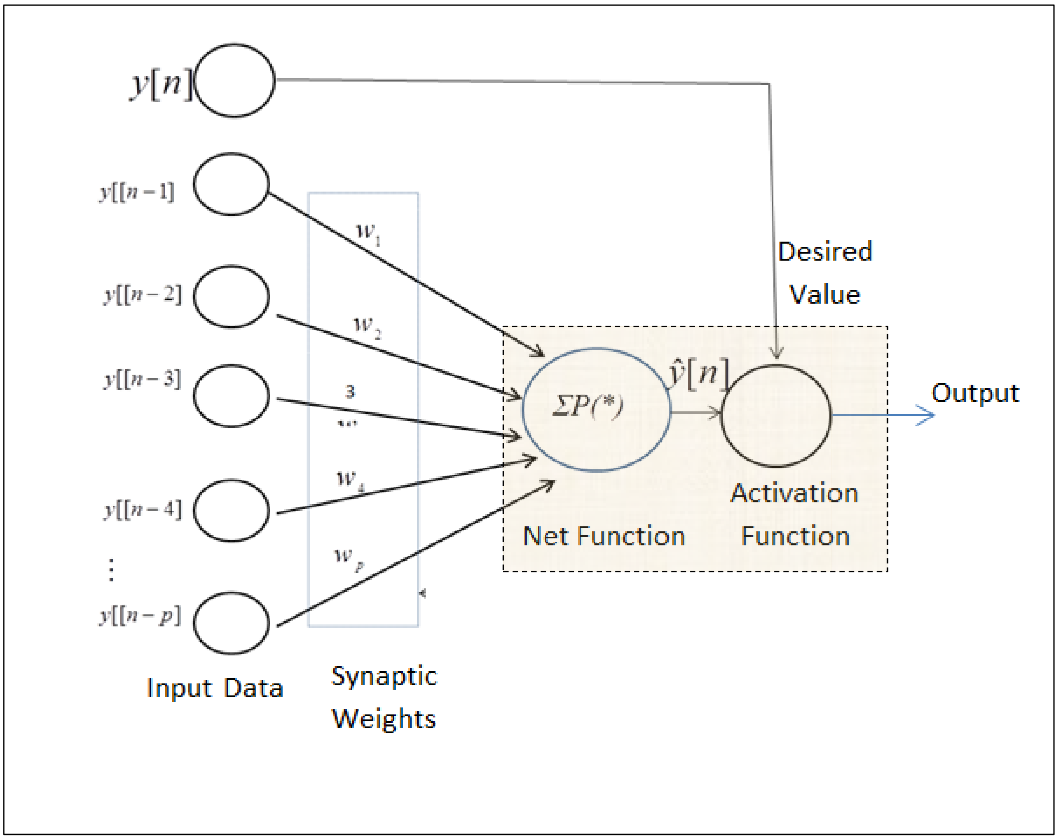


Figure 2.8 A MLP Model

2.6.2 ANN Intelligent Learning: The Training Procedure

The second part of ANN structure is the fundamental trait of intelligent that is ANN learning through ANN training. Although there are various algorithms for ANN training algorithms as shown in Table 2.11, such as Error-Correction (EC), Boltzmann or Hebbian, however the most popular one applicable to FFNN is the EC based Backpropagation Algorithm (BP) which will be presented here. In practice, 80%-90%

of FFNN are based on BP (Hu & Zhao, 2010). The block diagram of FFNN with EC BP is shown in Figure 2.9.

Table 2.11 Summary of ANN Learning Algorithms

Paradigm	Learning Rule	Architecture	Algorithms	Tasks
Supervised	Error Correction	Single, MLP	Perceptron Learning Algorithm (PLA) , Back-propagation, (BP) ,Adaline and Madaline (AM)	Pattern Classification, Approximation, Prediction ,Control
	Boltzman	Recurrent	Boltzman leaning algorithm	Pattern Classification
	Hebbian	MLP FFN	Linear Discriminant Analysis	Data Analysis , Pattern Classification
	Competitive	Competitive	Learning Vector Quantization	Within Class Categorization
		ART network	ARTmap	Pattern Classification,
Unsupervised	Error Correction	MLP FFN	Sammn's Projection	Data Analysis
	Hebbian	FFN, Competitive	Principal Component Analysis (PCA)	Data Analysis, Data Compression
	Competitive	Competitive	Vector Quantization	Classification, Data Compression
		Kohonen's SOM	Kohonen's SOM	Classification, Data Compression
		ART network	ART1, ART2	Categorization
Hybrid	Error Correction and Competitive	RBF network	RBF Learning Algorithm	Pattern Classification, Function approximation, Prediction, Control

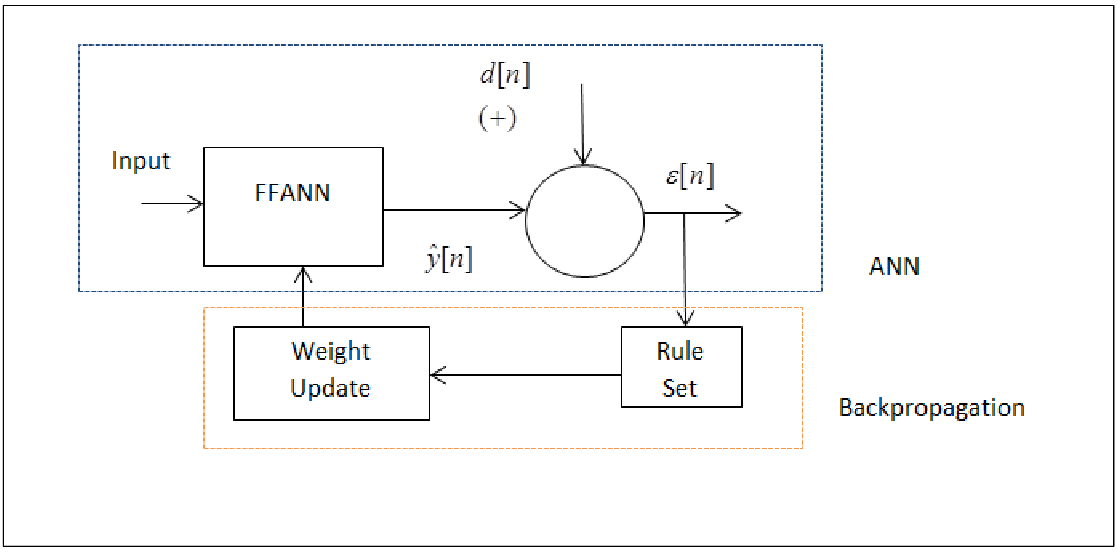


Figure 2.9 Relationship between MLP and BP

The basic principle in ECBP is that the output signal, $\hat{y}[n]$ from output layer is compared to desired signal, $d[n]$ and the error signal $\varepsilon[n] = d[n] - \hat{y}[n]$ is than computed before updating the connection weights $\vec{w}_i[n]$. This is repeated through an iterative process. In each iteration, the objective is to reduce the $\varepsilon[n]$ gradually, making it closer to an accepted value by updating the $\vec{w}_i[n]$ based on ruled defined in the BP learning algorithm. The number of iteration is pre-determined (known as epoch) and a minimum $\varepsilon[n]$ is selected out of the total epochs.. The best fit $\vec{w}_i[n]$ values will have minimum $\varepsilon[n]$.

A number of BP based learning algorithm found in literatures such as on-line neural network learning algorithm for TV inputs (Zhao, 1996), Fast Learning algorithms based on gradient descent of neuron space and Levenberg-Marquardt Algorithm (LMA), Newton Method (Corte & Zou, 2014) to name few. However, the conventional BP learning algorithm is developed based on Steepest Gradient Descent Algorithm (SGDA) (Rumelhart, Hinton, & Williams, 1994) and its algorithm is shown in the following Equations.

The SGDA algorithm adjusts the weights according to steepest descent direction or negative of the gradient, in a direction where the $\varepsilon[n]$ decreases rapidly. The network weight update by SGDA is given as:

$$w[n+1] = w[n] + \Delta w[n] \tag{2.29}$$

with SGDA,

$$\Delta w[n] = -\mu \nabla_w \xi$$
$$= -\mu \frac{\partial \xi}{\partial w_{jn}} \quad (2.30)$$

and

Average Mean Error Energy, $\xi = \frac{1}{N}\sum_{n=1}^{N} \varepsilon[n]^2$ (2.31)

where μ is the step size (or learning rate) and $\nabla_w \xi$ is gradient vector of cost function which sets the observation rule. The derivation of $\nabla_w \xi$ with respect to weight (Aibinu, 2010) is given as :

$$\frac{\partial \xi}{\partial w_{jn}} = \frac{\partial \xi}{\partial y}\frac{\partial y}{\partial u}\frac{\partial u}{\partial w_{jn}} \quad (2.32)$$

And now let the following definition :

$$\partial(n) = -\frac{\partial \xi}{\partial y} \quad (2.33a)$$

$$y[n] = F(u[n]) \quad (2.33b)$$

$$u[n] = \sum_j X_{jn} w_{jn} \quad (2.33c)$$

$$\frac{\partial u}{\partial w_{jn}} = X_{jn} \quad (2.33d)$$

Therefore Equation (2.32) with respect to weight update is derived as:

$$\frac{\partial \xi}{\partial w_{jn}} = \nabla_w \xi = -X_{jn} F^{'}(u[n])\partial[n]$$

and therefore the weight update is given as :

$$w[n+1] = w[n] + \Delta w[n]$$
$$= w[n] - \mu \nabla_w \xi$$
$$= w[n] - \varpi \frac{\partial \xi}{\partial w_{jn}}$$
$$= w[n] + X_{jn} F^{'}(u[n])\partial[n] \quad (2.34)$$

Equation (2.34) shows the synoptic weight update formula derived for a real value MLP using ECBP for single realization of *n*. From implementation perspective, an EC BP algorithm can be described in the following steps:

I. Random weight initialization.
II. Weight computation with ANN.
III. Weight update using ECBP learning algorithm.
IV. The algorithm is stopped when the minimum value of $\varepsilon[n]$ is reached.

2.6.3 BP-ANN Weight Optimization Techniques

The BP for ANN (BP-ANN) technique are well established, well studied and preferred by many researchers (Seman, Bakar, & Bakar, 2010). This is due to its benefits such as simple structure, maneuverability, less computation, strong concurrency and ability to produce arbitrary non-linear input-output relationship (Hu & Zhao, 2010). However, these BP based ANN learning lacks of optimization mechanism and seriously suffers from some inherit limitations.

Apart from suffering from slow convergence , it is also may converge to a wrong solution by converging to a local minima instead of global minima (Kartheeswaran, Dharmaraj, & Durairaj, 2015) . One possible reason for BP-ANN to be trapped in local minima is due to its strong dependence on initial weight guest (Murru & Rossini, 2016). Furthermore, it is obvious when a weight update algorithm has many local minima; devising further effective learning algorithm based on earlier algorithm will be not reliable as well. Therefore although BP-ANN guaranteed a rapid decrement in $\varepsilon[n]$ however, it does not necessarily produce fast convergence or a guaranteed convergence.

Briefly, other known shortcomings of BP ANN (Khan, T.K., & Sudhir Sharma, 2008; Aibinu, 2010; Seman et al., 2010) include:

- Difficulties in determining optimal network structure
- Overfitting parameters
- Require gradient information of the activation function.
- Problem of specific parameter settings
- Dependencies on initial guess for weights.

To overcome these shortcomings of BP-ANN, several methods have been suggested in literatures; which can be categorized into three as;

a) BP-ANN Parameter Optimization methods

Parameter optimization is varying some parameters of the ANN in order to increase the speed of convergence and accuracy of estimation. These include:

i) Rule based weight initialization methods.

As random weight initialization has been heavily affecting performances of BP ANN , many rule based weight initialization methods (Fernándes et. al, 2001; Murru & Rossini, 2016; Yam & Chow, 2000) has been proposed. These proposed methods have fluctuating performances and problem specific.

ii) Introduction of adaptive learning and structure minimization (Filho et al., 2013; Kawafuku & Sasaki, 1999; Madić & Radovanović, 2011)

iii) Dynamic weight updating methods (Alippi & A, 1991; Koščak, Jakša, & Sinčák, 2010)

b) BP-ANN Numerical Optimization

This method proposes an alternative solution to the SGDA by using standard numerical procedures like Quasi Newton's Method (NM), Levenberg-Marquadt (LM) optimization, Online Learning Algorithms (OLA), Conjugated Gradient with Parabolic Interpolation (CGPI) and Fast Learning Algorithms (FLA). Among all the methods, LM algorithm is the most widely used (Khan et al., 2008; Seman et al., 2010;Corte & Zou, 2014; Schiopu, Barbulescu, Kilyeni, Deacu, & Vernica, 2015;) .

c) Genetic Algorithm Based Weight Optimization

Most of the approaches above are problem specific, while others are not well accepted for complexity in application, computational complexity and unstable results produced. One promising approach for BP-ANN weight optimization is by using a GA as it is global in search procedure and efficient to eliminate local minima as it is not subjected gradient based rule. Furthermore, it also solves other issues such as design structure optimization, independent from weight initialization, coefficient initialization or overfitting problem.

2.6.4 Overview of Genetic Algorithm

GA has been widely used as efficient optimization tool in various field of signal processing. GA is Evolutionary Algorithm (EA) based search technique based mimicking biological process of reproduction. They are distinguished from other optimization techniques by the use of concepts of population genetics to guide the search. A complete and detailed information on principles and methodology of GA can be found in Whitley (Whitley, 1994) , K.F. Man et all (Man, Tang, & Kwong, 1996) , K.S. Tang et all (Tang, Man, Kwong, & He, 1996) , just to quote few while huge number of books and articles has been written on GA.

GA optimization procedure is not a rule based optimization tool, as in numerical approach, but based on evolution and 'survival of fitness" concept. Only strong candidates from a pool of population are able to survive from one generation to another until a decision is made by GA algorithm to end the search. This happens in an iterative process, where same process are repeated, past results are exploited and combined with new search till the FF is satisfied.

Even with this extraordinary features, GA does not guarantee a global minima solution, however its solution is definitely not sensitive to local minima, and furthermore able to locate neighborhood of an optimum solution to provide encouraging results (Yeremia, Yuwano, & Raymond, 2013). Furthermore, by considering a search space simultaneously, GA reduces the risk of convergence and trapping in local minima which is a major short coming of BP-ANN algorithm.

Other advantages of GA compared to BP are listed as:

a) *Parameter Coding.* GA works with a coding of parameter set, not the parameter themselves. In this way, they search the whole parameter space.
b) *Population Space.* GA adopts intelligent search process in finding the solution to a problem and the search is from a population of points and not a single point.

c) *Independent of derivative of any parameters.* GA uses fitness function evaluation not the derivatives of equations or knowledge.

d) *Probabilistic.* GA uses probabilistic transition rules in the selection process and not deterministic rule.

2.6.5 Motivation for Hybrid ANN-GA for Parameter Estimation

BP-ANN and GA are two different techniques for optimization and learning, each with its own strength and weakness. Both have been designed in analogy to biological processes that occurs in nature and have been evolved along separate paths from their original form to a complex structure to address current complex problems.

Although both techniques are used mostly separately, when put together they can extend their performance and capability to make use of the best of both algorithms in combination to produce a high performance hybrid of GA-BP ANN systems. Such approach has been studied for the past decade in automation of many complex problems which require a decision making process. A hybrid two stages approach using GA and BP-ANN had been used in the following ways:

a) Using GA as ANN learning algorithm.

   Since GA reduces the possibility of converging to local minima and improves the optimization capability, the GA is used to replace BP algorithms in ANN learning. Various successful application of ANN trained by GA are reported in literatures such as : (Q. Zhang & Wang, 2008;Agarwal, 2015; Fu, Genson, Jan, & Jones, 2011;)

b) Using GA to optimize ANN parameter and structural information

An ANN network structure is normally selected on the basis of researcher or developer's prior knowledge, experience or by a hit and trial approach. . This include deciding the ANN structural parameter such as number of layers, number of neurons and interconnecting the layers, basically this is more than art than of science. By using GA optimization capability, it is possible to improve the ANN architecture for a defined problem prior to the training and learning process of ANN. Some of the successful work have been reported for ANN structure optimization using GA can be found in (Zanchettin, Ludermir, & Almeida, 2011; Islam, Baharudin, Raza, & Nallagownden, 2014; L. Hu, Qin, Mao, Chen, & Fu, 2016;).

c) ANN weight initialization using GA.

As mentioned earlier, one reason for BP-ANN converges to local minima with because it heavily depends on random weight initialization. Furthermore, two different settings of initial weights can result in different convergence rate, even by choosing small initial weights. Random initialization can be avoided by distributing the initial weight in a manner that all weights leading to hidden layer are uniformly distributed and ensure no overlapping occurs (Sodhi, Chandra, & Tanwar, 2014). Some of the weight initialization algorithms are lease-square with standard pseudoinverse matrix, factorization, Nguyen and Widrow method and high performance GA (Wangberg, 2008; Q. Zhang & Wang, 2008). However such approach requires a lot of innovations.

d) ANN weight optimization using GA

A highly innovative method where the GA is used to optimize weight which has been estimated previously using BP-ANN. Some reported works are (Khorani, 2011;Kartheeswaran et al., 2015;)

Among the four method to optimize BP-ANN to avoid converging to local minima , the fourth method is further considered in this thesis. Such proposal should be known as BP-ANN-GA and is presented in the next chapter.

2.6.6 Related work on ANN-GA for Parameter Estimation

Related work on ANN, GA or ANN-GA based algorithms to estimate modal parameters are listed in Table 2.14. Only research works related to AR and ARMA parameter estimation is considered here, whilst physiological modelling on NSS using the artificial intelligence method is not listed.

Table 2.12 AI Based AR Parameter Determination Methods

Year	AI	Application	Reference
1996	Feedback recurrent ANN	AR coefficients	(Tian et al., 1997)
1998	GA	AR model order optimization	(Maust & Cq, 1998)
1991	ANN	ARMA coefficients	(Chon et al., 1999)
2001	GA	ARMA Model Orders	(Minerva & Poli, 2001)
2002	GA	AR Model Orders	(Wu & Chang, 2002)
2004	GA	ARMA Model Orders	(Baragona et al.,004)
2006	GA	AR Model Orders	(Palaniappan, 2006)
2008	BP ANN	AR Coefficients	(Mendez et al., 2008)
2008	FF-BP-ANN	ARMA Coefficients	(Aibinu et. al., 2008)
2010	Hybrid ANN, GA	ARMA Coefficients	(Aibinu, 2010)
2010	Hybrid ANN, GA	TVAR	(Zhang et al., 2010)
2012	Hybrid : FFANN, GA	ARMA coefficients	(Flores, Graff, & Rodriguez, 2012)
2012	ANN (RVNN)	AR Coefficients	(Aibinu et al., 2012)
2012	GA	ARMA parameters (Model orders and coefficient)	(Hammour et al., 2012)

2.7 Critical Analysis and Research Gap

Recent trend in processing a NSS is by adopting parametric technique, as a TVAR or a TVARMA processes. Due to its various goodness obtained when adopting a TVAR, easily it has found way in many fields of human endeavor especially in biomedical signal processing. Modeling a NSS using TVAR process includes determination of accurate model order, p and $a_j[n]$ to be obtained using an appropriate TVAR coefficient computation method. Both factors became a fundamental issue to be tackled for a good approximation of the underlying signal.

For TVAR process, appropriate values for both parameters are obtained using recursive computation. This recursive computation is repeated for each realization, n of the input signal to allow the $a_j[n]$ changes in time.

Basically, from literatures, available TVAR coefficient determination techniques are grouped into an AM (LMS, RLS, and KF) or a BF method, where the latter is more preferred due to its effectiveness in capturing multiple dynamics. BF also independent from various limitation demonstrated by AM such as independent from converging into a local minima and slow convergence. Furthermore ,it does not require initialization of random values for μ and λ.

It has been observed that estimation of the TVAR BF parameters is a very challenging problem. By adopting the TVAR BF approach, one has to decide two model orders (*p*,*m*), select appropriate basis function, $f_i[n]$ and efficient TVAR BF expansion coefficient, a_{ji} estimation algorithms. Therefore, obviously such tasks give rise to number of unknown parameters to be determined and the computation time as well. Other performance related issues of TVAR BF are addressed as:

a) Difficulties in developing an efficient algorithm which simultaneously compute the (*p*,*m*) and TV $a_j[n]$.

b) Problem specific solutions, thus limiting the scope of developed solutions to certain applications.

c) Techniques developed heavily depending on complex mathematical procedures.

d) TVAR coefficient estimations methods are based on exhaustive search.

e) TVAR coefficients are sensitive to (*p*,*m*), thereby limiting the application of TVAR in many cases.

f) Sensitivity of BF in representing various type of NSS. Each BF is good for certain type of NSS which has similar characteristics with the BF.

g) Difficulties in selecting a set of BF is which able to track broad type of NSSs either slow varying, fast varying or with both slow and fast varying frequencies.

Thus, from the listed shortcomings and challenges, we come to an understanding that there is a need to further explore TVAR parameter estimation which should solve at least some of the aforementioned shortcomings, if not all.

One possible way is to investigate AIS based on ANN, GA or hybrid of GA-ANN to determine TVAR model parameters. The new AIS based algorithm should consider a global solution taking advantages of efficient searching capability of ANN and GA. The intelligence of AIS based algorithm is conjectured to make the TVAR less dependent on model orders and type of basis function. Such algorithms will path a new direction in TVAR parameter estimation.

Based on our broad literature review, a two step hybrid method to estimate the TVAR coefficients is proposed in this thesis. The proposed hybrid method takes advantages on superior performance of BP-ANN and GA optimization to combine them together to get the best of both intelligent algorithms to estimate the TVAR BP-coefficients. The BP-ANN will be used to estimate the TVAR coefficients and GA will be used to further optimize the TVAR coefficients. The BP-ANN topology is proposed to be constructed from mathematics of TVAR models. Further discussion will be presented in Chapter Three.

Although BP-ANN and GA has been studied extensively in many area of signal processing either individually or combined together, however, their application to estimate the TVAR coefficient is very rare.

The closest work to the proposed algorithm found on literature was by Zhang ((L. Zhang et al., 2010). However Zhang has developed TVAR estimation method based on TVAR BF and secondly, Zhang has used GA to optimize the set of model orders , p and m based on EOM's Maximum-Likehood cost function (Eom, 1999). Zhang has used DCT BF which is not suitable to capture slow or mixed dynamics contained in a NSS and only applicable on the signal of their interest. By proposing hybrid method of BP-ANN and GA to be developed based in TVAR modelling, we have established and demonstrated the research gap which is required for this thesis.

2.8 Summary

In this chapter, processing of NSS processing techniques has been broadly reviewed at several stages in order to identify the research gap in this field of knowledge. This chapter began with an introduction to NSS analysis methods; where different types of processing techniques were presented. Later, TFR methods were introduced as better processing tool compared to classical methods which makes no assumption on TV characteristics of NSSs. TFR methods are further grouped into non-parametric and parametric method. In Section 2.2, TFR algorithms were discussed and their strength and limitation are summarized in tabular forms. In Table 2.1, popular non-parametric TFR algorithms are listed while their strength and limitation are listed in Table 2.2. This is followed by discussion on parametric method which was introduced to overcome shortcomings of non-parametric TFR methods. Among available parametric TFR, TVAR has been reported as preferred method of many researchers, reasons for this selection are presented at the end of Section 2.3. TVAR parameter estimation techniques are discussed in Section 2.4; whereby two different categories of TVAR coefficient determination techniques are presented which are AM and BF method. AM are presented in subsection 2.4.1, where a list of advanced AM are summarized in Table 2.4 and performances of adaptive algorithms are listed in Table 2.5 respectively. Performance evaluation of TVAR BF coefficient estimation methods are listed in Table 2.6. A flow chart on implementation of TVAR BF methods is shown in Figure 2.5 as well. At the end of Section 2.4, performance analysis of both AM and BF are summarized in Table 2.9. In Section 2.5 a list of related work for TVAR coefficient estimation methods are presented in tabular form. In Section 2.6, AIS is presented which covers the need and motivations for ANN , ANN learning algorithm, and different type of optimization techniques for ANN. List of related work on ANN-GA for AR parameter estimation methods is listed in Table 2.16 at the end od Section 2.6. Finally in Section 2.7, critical analysis and research gap of TVAR parameter estimation methods was presented as a conclusion of this chapter.

CHAPTER THREE

DEVELOPMENT OF BP-ANN-GA ALGORITHM TO ESTIMATE TVAR COEFFICIENTS

3.1 INTRODUCTION

The development of a hybrid BP-ANN-GA algorithm to estimate TVAR coefficients will be presented in this chapter. Generally, BP-MLP based topology is the preferred ANN architecture for most modeling and optimization problems as for its simplicity and direct implementation. Although BP learning rule has been proven to be efficient mathematically, however, it severely suffers from the convergence issues whereby it can be easily trapped in local minimum solution instead of global minima.

It is known that the performance of any ANN architecture depends not only on ANN training algorithm but also on its architectural parameters such as the number of hidden layers, number of nodes in input layer, type of mapping from the input layer to the hidden layer and the type of training. Generally, these parameters are selected based on previous experience or by trial and error approach. For this reasons, this chapter begins with ANN topology selection and designing criteria.

Two different BP-ANN structures are proposed in this chapter, where in the first, the ANN topology is designed for Direct TVAR model. While in the second proposal, the BP-ANN‘s topology is designed from TVAR BF model. In both cases, a mathematical relationship to ANN topology is established.

A two-level hybrid BP-ANN-GA algorithm is proposed in this chapter to estimate TVAR coefficients. In the first level, TVAR coefficients are estimated from synaptic weights of BP-ANN while in the second level, the estimated TVAR coefficients are further optimized using the robust optimization technique, GA algorithm. GA optimization is introduced as a solution for BP-ANN major drawback, the local minima issue. These combinations of two intelligent methods are found to be an innovative approach in estimating TVAR coefficients.

3.2 ANN ARCHITECTURE SELECTING CRITERIA

It has been observed in recent years that there is an increasing research interest to adopt ANN for modeling and optimization problems. From the signal processing perspective, it is imperative to develop a proper understanding of the basis of ANN structure and its impact on the problem being studied prior to proposing a ANN architecture.

Although there are numerous advantages offered by ANN algorithm, however they are achievable only when implementing a suitable architectural parameters and training process. Since there is no definite and explicit method which guides the ANN topology development, determining suitable architectural parameters still remains as a difficult task. These parameters are typically determined in trial and error procedure where few models are proposed and the best which represent the problem is selected. In our case, the modelled ANN should typically represent TVAR mathematics.

In this section, briefly we present the list of decisions to be made in an ANN designing process. Basically, four types of parameters need to be addressed before designing the ANN-GA framework, which are:

a) *ANN topology parameters*

ANN topology represents the way in which neurons and layers are interconnected to form a network framework. Among the structural parameters which to be identified are:

I. Decision on the type of ANN architecture

ANN architecture can be design based on a single layer Feedforward Network (FFN), Multilayer Feedforward Networks (MFFN), Feedback Neural Network (FBNN), Recurrent Networks (RN), Radial Basis Function (RBF) and Kohonen's self-organizing network. Each of this structure is founded on different mathematical background and has its unique advantages. Therefore, for signal processing applications, the best ANN topology is the one which fit the mathematics of the problem being studied.

It was conjectured a FFANN topology should represent a TVAR and TVAR BF modeling.

II. Decision on the number of layers.

Most applications require networks that contain at least three layers which are an input layer, hidden layer and output layer. . Number of hidden layers is decided based the objective of ANN structure and deep of learning that it requires. It is proposed that the proposed BP-ANN should have three layers with one artificial neuron in the hidden layer.

III. Decision on the number of nodes in the input layer.

In BP-ANN, the nodes in input layer should represent the input data and number of nodes should be equal to model order. The input data for BP-ANN for TVAR and input node for BP-ANN for TVAR BF is not same.

IV. Decision on the number of neurons in hidden layer.

Neuron are the main working units in an ANN, selection of neurons depends on problem studied. In BP-ANN, one artificial neuron based on McCulloch and Pitts's model (Hu & Hwang, 2002) which has two parts, the net function and the activation function is proposed in the hidden layer.

V. Decision on the type of the net function.

The input signals connected to neurons in hidden layer with by synaptic weights, while the net function determines how this weighted input are combined inside the neuron. A net function can be selected from a Linear Net Function (LNF), Higher Order Functions (HOF) or a Delta Function (DF) (Raul Rojas, 1996). A linear net function (adder) is proposed for BP-ANN for both TVAR and TVAR BF.

VI. Decision on the type of neuron activation function

An activation function maps the outcome from net function to an output value by performing certain fixed mathematical operation. There are several activation functions available, namely Linear Activation Function (LAF), Sigmoid Activation Function (SAF), and Hyperbolic Tangent (HT), Inverse tangent (IT) or Gaussian Radial

Function (GFR) (Hu & Hwang, 2002). A LAF is proposed to be adopted for BP-ANN.

b) *ANN learning algorithm*

The ultimate objective in proposed BP-ANN modeling is to estimate the synaptic weights. The optimal synaptic weights are obtained from series of iteration where in each iteration the synaptic weight will be updated according to a learning rule until a set of synaptic weight is obtained. BP learning algorithm is popular choice for FFANN.

c) *Selection of GA parameters*

One important parameter for GA setting is definition of a Fitness Function (FF), which determines how 'good' the solution with respect to the given problem. To define a FF, it is necessary to know the detailed problem. It is depends on design variables, should be simple and fast. Since the calculation of FF is done repeatedly in a GA, a very complex FF will make the algorithm exceptionally slow. The GA algorithm is also should be set with the objective is to maximize or minimize the FF.

d) *Establishing a relationship between ANN and GA*

The ANN and GA can be combined to make use the best of both algorithms in four unique ways which are:

- Using GA to optimize ANN architecture.
- Using GA as weight initialization algorithm prior to BP training.
- Using the GA to train ANN instead the BP learning algorithm.
- Using GA to optimize synoptic weights.

It is proposed to use the fourth method that is to optimize synoptic weight of BP-ANN using GA algorithm.

Hence, from the discussion above, we propose an intelligent algorithm of BP-ANN-GA based on TVAR and TVAR BF with features mentioned earlier to estimate the TVAR coefficients. The proposed method will be presented in subsequent sections

3.3 DEVELOPMENT OF BP-ANN FOR TVAR COEFFICIENT ESTIMATION

Two different BP-ANN schemes are proposed, one based on Direct TVAR while the other one based on TVAR BF approach. Both have similar ANN topology, however, the input data and ANN learning objective is different.

3.3.1 Development of ANN architecture for Direct TVAR

Let's consider a Direct TVAR model as in Equation 3.1:

$$y[n] = -\sum_{j=1}^{p} a_j[n]y[n-j] + e[n] \tag{3.1}$$

where $a_j[n]$; $j = 1 \leq j \leq p$ are the TVAR model coefficients,

p is the model order,

$e[n]$ is the error function.

The estimate of $y[n]$ is denoted as $\hat{y}[n]$ is computed from:

$$\hat{y}[n] = a_1[n]y[n-1] + a2[n]y[n-2]\cdots + a_p[n]y[n-p] \tag{3.2}$$

where Eq. (3.2) can be represented in a matrix form as:

$$\hat{y}[n] = \vec{A}_n^T \vec{y}_n \tag{3.3}$$

where

$$\vec{A}^T\ [n] = [a_1[n]\ \ a_2[n]\ a_3[n]\ \cdots\ a_p[n]]^T \tag{3.4}$$

$$\vec{y}[n] = [y(n-1)\,,\, y(n-2)\cdots y(n-p)] \tag{3.5}$$

Hence, from the view point of the system modeling, the terms $\hat{y}[n], \vec{A}^T[n], \vec{y}[n]$ in Equation (3.3) could be regarded as system output, system parameters and input, thus translating these parameters into ANN input and output parameters.

Therefore the three-layer BP-ANN architecture which was proposed to represent Direct TVAR as in Equation. (3.1), is now having following parameters:

a) First layer is input layer, consists of p nodes representing input signal, $\vec{y}$.
b) The hidden layer consists of an artificial processing neuron following the McCulloch and Pitts s' model which consists a LNF (adder) and LAF is proposed as two working units inside this processing neuron.
c) The output layer consists of a single neuron which will compute the error between targeted data and estimated data, and feed the output to BP learning algorithm.

The LNF, as the first working unit inside the single artificial neuron of hidden layer is basically in form of a weighted adder form as in Equation (3.6). It is used to map the input layer to hidden layer by using synaptic weight.

$$\begin{aligned} u[n] &= \delta\Big(\sum_{j=1}^{N} w_j[n]y_j + \theta\Big) \qquad ; N = p \\ &= \delta(w_1 y(n-1) + w(n-2) + \cdots + w_p y(n-p) + \theta) \end{aligned} \tag{3.6}$$

where N is number of nodes in input layer which is set to model order ,*p*.

$w_j[n]$ represents the synaptic weight connecting input signals to neuron at n ,

δ is adder as the net function

θ is threshold

$u[n]$ is output from the LNF adder function.

$u[n]$, output from LNF than goes through the second working unit, LAF which has the following working rule, as in Equation (3.7)

$$F(x) = x \tag{3.7}$$

Consequently, when Equation (3.6) is feed into Equation (3.7), LAF maps into the input, $u[n]$ to its output, $\hat{y}[n]$ as:

$$\begin{aligned}\hat{y}[n] &= F(u[n]) \\ &= F(w_1 y(n-1) + w_2 y(n-2) + \cdots + w_p y(n-p)) \\ &= w_1 y(n-1) + w_2 y(n-2) + \cdots + w_p y(n-p)\end{aligned} \tag{3.8}$$

By comparing Equation (3.8) with Equation (3.2), we found that the $a_j[n]$ is now equivalent to the synoptic weights, $w_j[n]$ of BP-ANN. Hence a relationship between TVAR and BP-ANN has been established.

With these definitions of parameters for BP-ANN architecture for TVAR model, the $a_j[n]$ estimation problem as in Equation (3.1) now becomes a FFANN training problem which topology is shown in Figure 3.1.

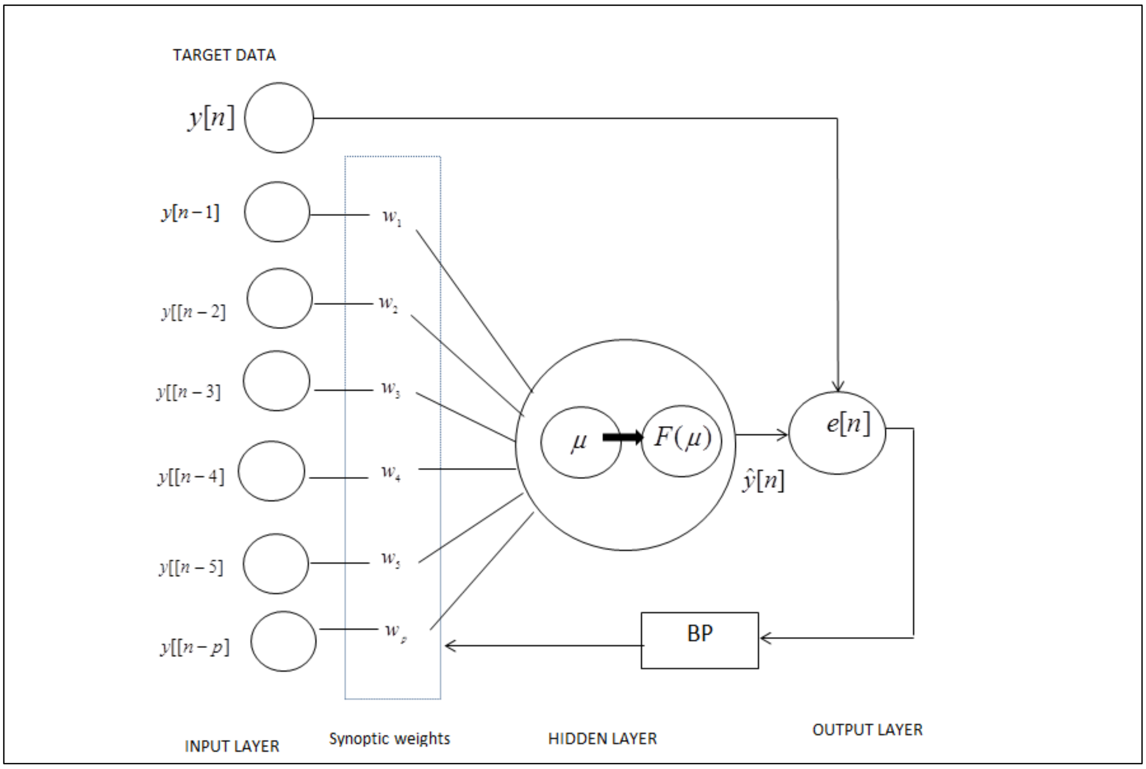


Figure 3.1 Proposed BP-ANN Topology for Direct TVAR Model

3.3.2 Development of ANN architecture for TVAR BF Model

Now we proceed to develop a BP-ANN structure for TVAR BF model as an extension from the BP-ANN developed for Direct TVAR in the previous section. In BF method, the $a_j[n]$ from Equation (3.1) is expanded as now summation of weighted known BF as:

$$a_j[n] = \sum_{i=0}^{m} a_{ji} f_i[n] \tag{3.9}$$

where m, a_{ji} and $f_i[n]$ are known as the expansion parameters,

a_{ji} is constant coefficient of expansion parameter,

m is expansion dimension or order of expansion parameter which represents number of basis sequence

$f_i[n]$ is a set of selected known BFs.

Substituting Equation (3.9) into Equation (3.1), yields

$$y[n] = -\sum_{j=1}^{p}(\sum_{i=0}^{m} a_{ji}[n] f_i[n]) y[n-j] + \sigma\ [n] \tag{3.10}$$

And an estimate of $y[n]$ using TVAR BF model is given as:

$$\hat{y}[n] = -\sum_{j=1}^{p}(\sum_{i=0}^{m} a_{ji}[n] f_i[n]) y[n-j] \tag{3.11}$$

And by redefining the parameters in vector form as in Equation (3.12), Equation (3.11) can be represented in a compact matric form in in Equation (3.13):

$$\begin{aligned}
\vec{F}_i[n] &= [f_0(n), f_1(n), \cdots, f_m(n)] \\
\vec{c}_{j,i} &= \begin{bmatrix} a_{1,1}, a_{1,2}, \cdots, a_{1,m} \\ \vdots \quad \vdots \quad \vdots \\ a_{p,1}, a_{p,2}, \cdots, a_{p,m} \end{bmatrix} \\
\vec{x}_j[n] &= y[n-j]\vec{F}_i[n] \\
\vec{X}_n &= [\vec{x}_1(n), \vec{x}_2(n), \cdots, \vec{x}_p(n)]
\end{aligned} \tag{3.12}$$

where $\vec{F}[n]$ is vector of BF for each realization of *n*,
$\vec{c}_{j,i}$ is two dimensional matrixes for a_{ji} ,
$x_j[n]$ is an innovative term representing multiplication of input data and BF,
$\vec{X}_n$ is vector representing the innovative term $x_j[n]$,

Equation (3.12) is defined for $j = 1,2,3 \cdots p$ and $i = 1,2,3, \cdots, m$ representing model orders of TVAR and BF as in Equation (3.11).

Hence, Equation (3.11) in vector form is:

$$y[n] = \vec{X}_n^T\ \vec{c}_{i,j} \tag{3.13}$$

Again, from the view point of system identification, the terms $\vec{c}_{j,i}$, $\vec{X}_n$ and $\vec{x}_j[n]$ could be regarded as system parameters, inputs and output of a system respectively. Thus allowing TVAR BF to be represented in BP-ANN structure similar to Direct TVAR in previous section.

Input layer of BP-ANN for TVAR BF model $\vec{X}_n$, while the hidden layer consists of an artificial neuron McCulloch and Pitts s' model with LNF and LAF as two working. The third layer consists of a single neuron which will compute the error between targeted data and estimated data, and feed the output to BP learning algorithm.

Output from LNF is given as:

$$\begin{aligned} u[n] &= \delta(\sum_{j=1}^{N} w_j[n]x_j + \theta) \qquad ;N = p \text{ X } m \\ &= (w_1[n][x[n-1] + w_2[n]x[n-2] + \cdots + w_{p(m+1)}[n]x[n-p] + \theta) \end{aligned} \tag{3.14}$$

And feed into LAF as in Equation (3.7)

$$\begin{aligned} \hat{y}[n] &= F(u[n]) \\ &= F(w_1[n][x[n-1] + w_2[n]x[n-2] + \cdots + w_{p(m+1)}[n]x[n-p]) \\ &= w_1[n][x[n-1] + w_2[n]x[n-2] + \cdots + w_{p(m+1)}[n]x[n-p] \end{aligned} \tag{3.15}$$

Where $\vec{x}[n]$ is innovative term as defined in Equation (3.12) representing input data sequence multiplies the BF as in Equation (3.11). Comparing Equation (3.15) to Equation (3.11), now we find that the BP-ANN's synaptic weight from Equation (3.15) is equivalent to expansion coefficients, a_{ji} in Equation (3.11), hence a relationship is established with BP-ANN and TVAR BF. . Once the expansion coefficients a_{ji} are estimated from proposed model for given BF, the TVAR coefficients, $\{a_j[n], j = 1,2,3,\cdots, p\}$ is then estimated from Equation (3.9). BP-ANN (BF) has ANN topology which is similar to Figure 3.1 except of the number and input data in input layer. Parameters of both BP-ANNs are listed in Table 3.1

Table 3.1 Comparison of Parameters for BP-ANN (TVAR) and BP-ANN (TVAR BF)

Parameters	BP-ANN (TVAR)	BP-ANN (TVAR BF)
Number of layers	3	3
Number of hidden layers	1	1
Number of nodes in input layer	*p X m*	*p*
Neurons in hidden layer	1	1
Net function	LNF	LNF
Activation Function	LAF	LAF
Initial weights	1	1
Input Values	$\vec{X}_n$	$\vec{y}[n]$
Output from hidden layer	$\hat{y}[n]$	$\hat{y}[n]$
Synaptic Weight	a_{ji}	a_j

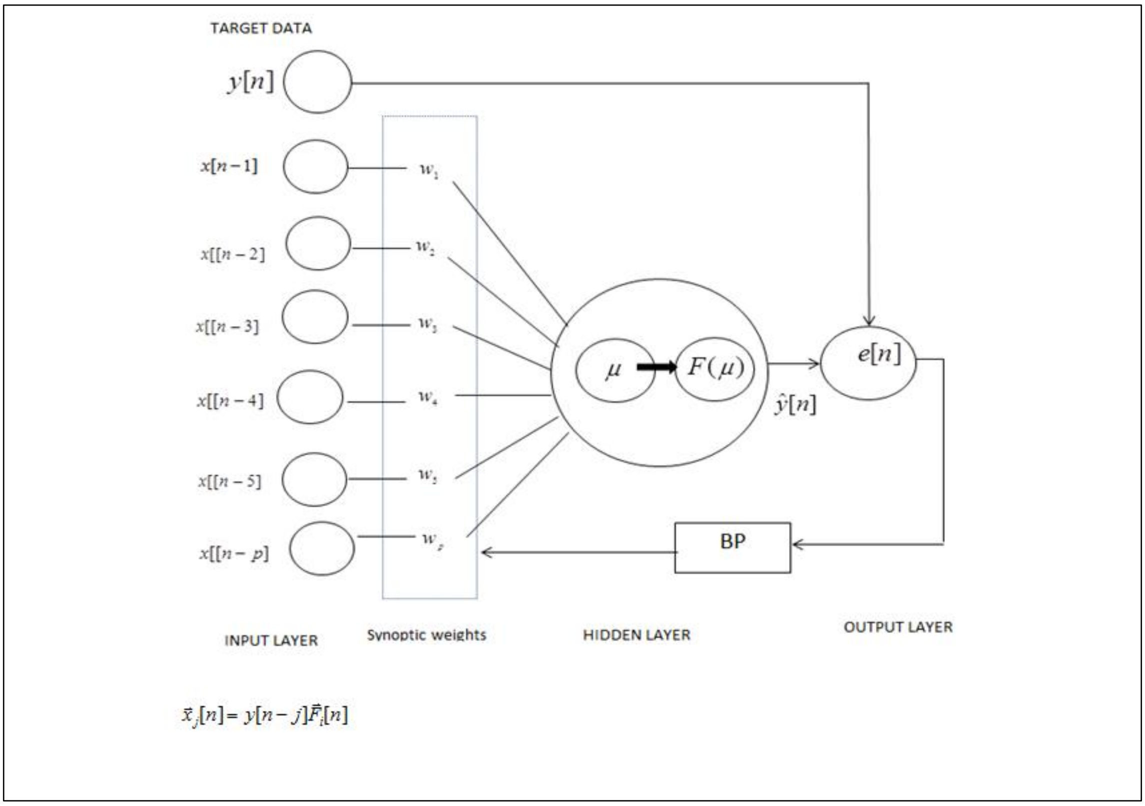


Figure 3.2 Proposed BP-ANN Topology for TVAR BF Model

3.3.3 Relationship between ANN Synaptic Weights and BP

From the previous sections, the problem of TVAR and TVAR BP coefficient determination has been converted to ANN learning and training. And a relationship is established whereby the synoptic weight of the proposed BP-ANN $w_j[n]$ is shown to represents $a_j[n]$ for TVAR model and a_{ji} for TVAR BF model respectively.

Such in both cases, the objectives are to estimate optimal values for $w_j[n]$ to ensure that the output $\hat{y}[n]$ gets sufficiently close to target value, $y[n]$ by using BP algorithm, an ANN training and learning algorithm. The optimal values for $\vec{w}_j[n]$ are attained when the$\varepsilon[n]$, an error based cost function has a global minimum in a series of iteration. In each iteration, the $\vec{w}_j[n]$ will be updated following a specific rule and the $\varepsilon[n]$ will computed.

The $\varepsilon[n]$ is defined to be the difference between the estimated output and a desired output:

$$\varepsilon_n = \sum_{t=1}^{T}(e_n[t])^2 = \sum_{t=1}^{T}(y_n[t] - \hat{y}_n[t])^2 = \sum_{t=1}^{T}(y_n[t] - f(\vec{y}_n, \vec{w}_{n,m}))^2 \tag{3.16}$$

where T the total number of training patterns (epochs),
ε_n is error function that to be minimized,
m is total number of input sequence (nodes) where $m = 1,2\cdots p$

The weight update for each epoch, are governed by the following equation:

$$\vec{w}_{n,m}[t+1] = \vec{w}_{n,m}[t] + \Delta\vec{w}_{n,m}[t] \tag{3.17}$$

where $\Delta\vec{w}_{n,m}[t]$ is a vector of weight update or in another word as corrections made to $\vec{w}_{n,m}[t]$.

Various algorithms are available to compute $\Delta\vec{w}_{n,m}[t]$, a list of those has been listed in Table 2.13 of section 2.6.2 in previous chapter. The derivation for $\Delta\vec{w}_{n,m}[t]$ using gradient method which is basis for BP algorithm begins with the following definition:

$$\Delta w_{n,m} = -\mu\nabla_w\varepsilon_n = -\mu\frac{\partial\varepsilon_n}{\partial w_{n,m}} \tag{3.18}$$

where μ the learning is rate,
$\nabla_w\varepsilon_n$ is the gradient of error function.

And when y_n and $\hat{y}_n(t)$ represented as $d_n(t)$ (desired value), $z_n(t)$ (estimated value), the $\nabla_w \varepsilon_n$ is stated as:

$$\frac{\partial \varepsilon_n}{\partial w_{n,m}} = \sum_{t=1}^{T} \frac{\partial (\varepsilon_n)^2}{\partial w_{n,m}} = \sum_{t=1}^{T} 2[d_n(t) - z_n(t)](-\frac{\partial z(t)}{\partial w_{n,m}}) \tag{3.19}$$

where

$$\frac{\partial z_n(t)}{\partial w_{n,m}} = \frac{\partial F(u_n)}{\partial u_n} \frac{\partial u_n}{\partial w_{n,m}} \tag{3.20}$$

and by following definition on Equation (3.20)

$$\frac{\partial F(u_n)}{\partial u_n} = F^{'}(u_n)\,,\ \frac{\partial u_n}{\partial w_{n,m}} = \frac{\partial}{\partial w_{n,m}} \left(\sum_{j=0}^{p} w_{n,m}[j] x[n-j] \right)$$

which leads to (Hu & Hwang, 2002)

$$\frac{\partial z_n(t)}{\partial w_{n,m}} = F^{'}(u_n) x(n-m) \tag{3.21}$$

Hence $\frac{\partial \varepsilon_n}{\partial w_{n,m}}$ can be written as:

$$\frac{\partial \varepsilon_n}{\partial w_{n,m}} = -2 \sum_{t=1}^{T} [d_n(t) - z_n(t)]\ F^{'}(u_n(t)) x_{n,m} \tag{3.22}$$

and with definition of

$$\delta_n[t] = [d_n(t) - z_n(t)] F^{'}(u_n(t)) \tag{3.23}$$

And therefore, Equation (3.19) is finally deduced as:

$$\frac{\partial \varepsilon_n}{\partial w_{n,m}} = -\mu \sum_{t=1}^{T} \delta_n[t] x_{n,m} \tag{3.24}$$

The term $\delta_n[t]$ represents the error based cost ε_n modulated by derivative of activation function, $F(u_n(t))$ in the hidden layer of BP-ANN and hence Equation (3.23) represents the amount of correction needed for $\vec{w}_{n,m}[t]$ of input data.

Therefore, finally the weight update equation is given as:

$$w_{n,m}[t+1] = w_{n,m}[t] + \mu \sum_{t=1}^{T} \delta_n[t] x_{n,m} \tag{3.25}$$

The weights will be updated in a series of iteration (epoch) using rules defined as in Equation (3.25) until the ε_n reached a certain predefined value which can be accepted as a minimum value. It ε_n does not converge to the predefined value, than the minimum ε_n will be computed from list of ε_n generated for a maximum number of epochs. The optimal weights are obtained for minimum of ε_n .

The type of learning rule algorithm shown above is modified from Levenberg-Marquadt (LM) BP method for each realization, *n*. The parameters based on the mathematical description above are listed in Table 3.2. Proposed ANN architecture for TV ANN with BP to determine TVAR coefficient is shown in Figure 3.2.

Table 3.2 BP Training Parameters

Training Parameters	Values
Learning Rule	Levenberg-Marquardt BP
Learning Rate	Variable determined by Levenberg-Marquardt BP
Mean Square Error	0.01
Maximum number of training (epoch)	Software Default 1000 or MSE is reached

3.3.4 Development of GA Optimization Algorithm

As explained earlier in previous section, there are four ways to optimize an ANN network identified from the literature review, which are optimizing ANN topology parameters using GA, ANN training with GA, ANN synoptic weight GA optimization and ANN weight initialization with GA.

Among these, ANN synaptic weight optimization using GA will suit our research objectives as our proposed BP-ANN architecture was designed to find optimal synaptic weights as these synaptic weights represent TVAR and TVAR BF coefficients. Therefore, GA can be used to further optimize the estimated synaptic weights in order to avoid local minima issues and hence as a solution to the shortcomings of BP learning algorithm also will be achieved too.

Prior to GA implementation, we should decide three input parameters; which are (1) a suitable coding to represent the problem (2) a fitness function(FF) and (3) limit that should be imposed such that the GA will not optimize estimated beyond these values. . The coding is basically a binary string of 1s and 0s which are randomly selected by GA. This initial populations are candidates of solutions which are evaluated using a FF. A FF is a type of objective function that is used to evaluate the goodness of each solution, by providing it a measurement to estimate solution candidates. If solutions are not obtained, a new generation of population will be generated and this process will be repeated iteratively by GA algorithm until the stopping criteria are met as set by FF.

In optimizing the TVAR coefficients, a suitable FF for GA is proposed in this thesis. The proposed FF will decide how close is the designed solution (trained data) to the targeted values by computing the error between these values. Too achieve this objective, the FF will receive three inputs which are the input data sequence, the targeted data sequence and the estimated synaptic weights from BP-ANN. The proposed FF is described as :

$$tt = X^T * \mathrm{A};$$
$$err(n) = |TT - tt| \quad (3.26)$$

where TT is target data,
tt is estimated output computed using newly optimized TVAR coefficients, A
input data sequence X^T as been defined in Equation (3.12).

The GA algorithm will perform a heuristic search through its working steps to produce a set of TVAR coefficients, $\vec{A}_n$ by rule defined in the FF . In each population, it will compute $err[n]$ GA will stop if $err[n]$ reach a predetermined error value or else it will continue for maximum iteration. GA based optimization flowchart is depicted in Figure 3.3 while summary of GA parameters are listed in Table 3.3.

Table 3.3 GA Optimization Parameters

GA parameters	Values
Population Size	Model Order
Initial population	TVAR Coefficient estimated from BP ANN network
Population Coding	Double
Fitness Function	As defined in Equation 3.26
Fitness Function Objective	Minimization
Selection Criteria	Default: Roulette wheel selection
Stopping Criteria	Default : 100 generations or error less than 0.01
Number of variables in the problem	p or pXm
Lower bound and Upper bound	$a_j[n] \pm b$;

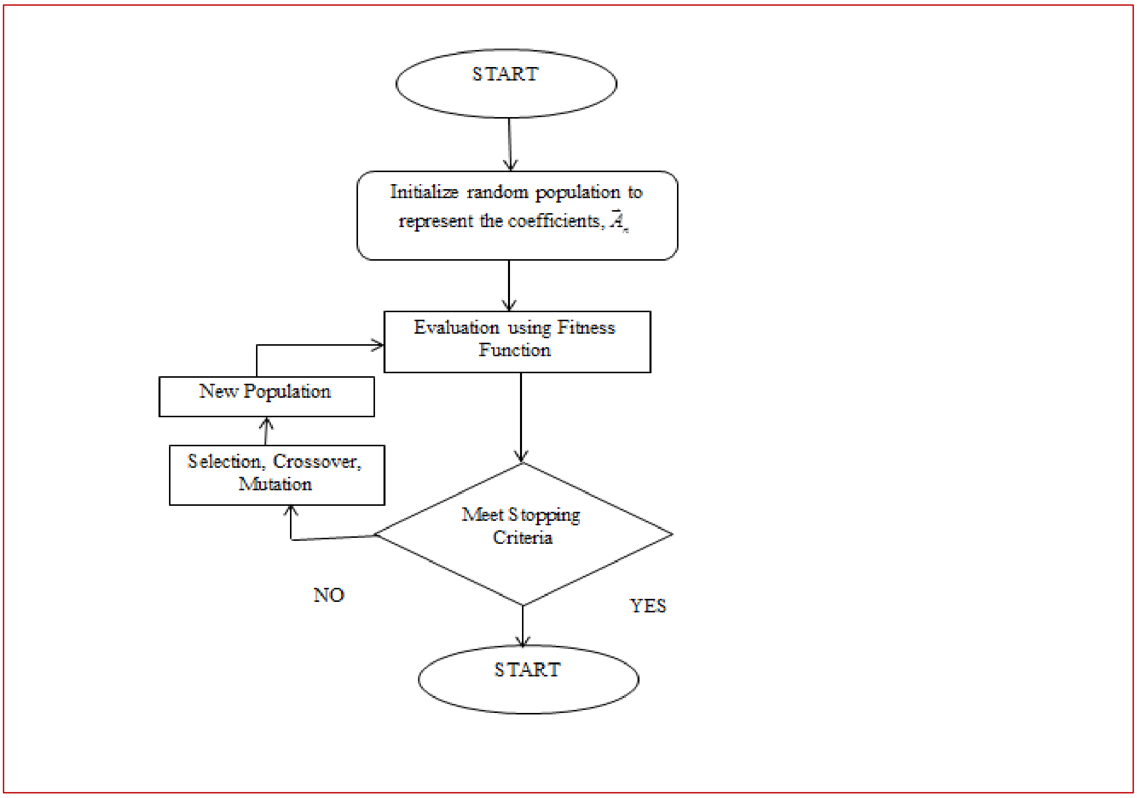


Figure 3.3 GA Optimization Flowchart

3.4 IMPLEMENTATION OF THE PROPOSED HYBRID BP-ANN-GA

The hybrid of BP-ANN-GA is implemented as a two level algorithm where during the first level, the TVAR coefficients are estimated from BP-ANN. Two different ANN scheme were developed in Section 3.3 which are based on Direct TVAR and TVAR BF. In the second level, the estimated TVAR coefficients are feed into the GA algorithm for optimizations. Such, the relationship between BP-ANN and GA is established in here as shown in Figure 3.4.

In BP-ANN-GA hybrid algorithm, the limit , b is used to set lower limits and upper limits on estimated $w_j[n]$ such the GA optimized coefficients will be $w_j[n] \pm b$. This is to control the GA algorithm to produce stable $w_j[n]$, in which the poles of the TVAR system will be inside unit circle, else the GA optimization may produce optimized $w_j[n]$ far from the estimated $w_j[n]$.

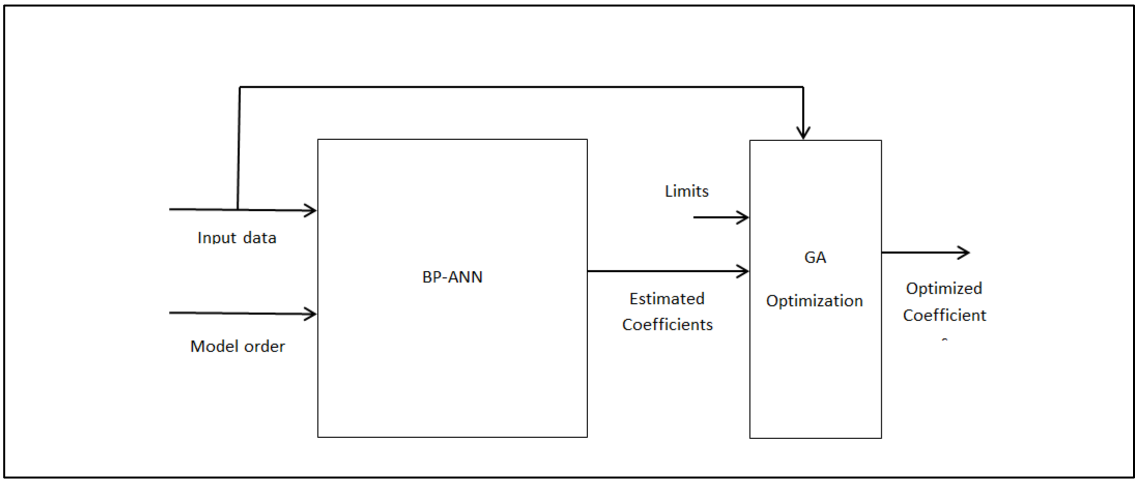


Figure 3.4 BP-ANN-GA Schematic Diagram

An implementation flowchart of BP-ANN-GA is shown in Figure 3.5. It is implemented four steps namely, data normalization, data framing, BP-ANN training and GA optimization as shown in Figure 3.5. The data normalization and framing are part of preprocessing steps while training and optimization are the processing steps of BP-ANN-GA.

I. Data Normalization

The first stage in the proposed hybrid BP-ANN-GA algorithm is data normalization, which is fundamental preprocessing for mining and learning from data. It is a process to reduce unwanted variations in the signal by limiting the data with a certain range. It also functions as scaling or averaging a signal and shown to greatly affect the accuracy of the system.

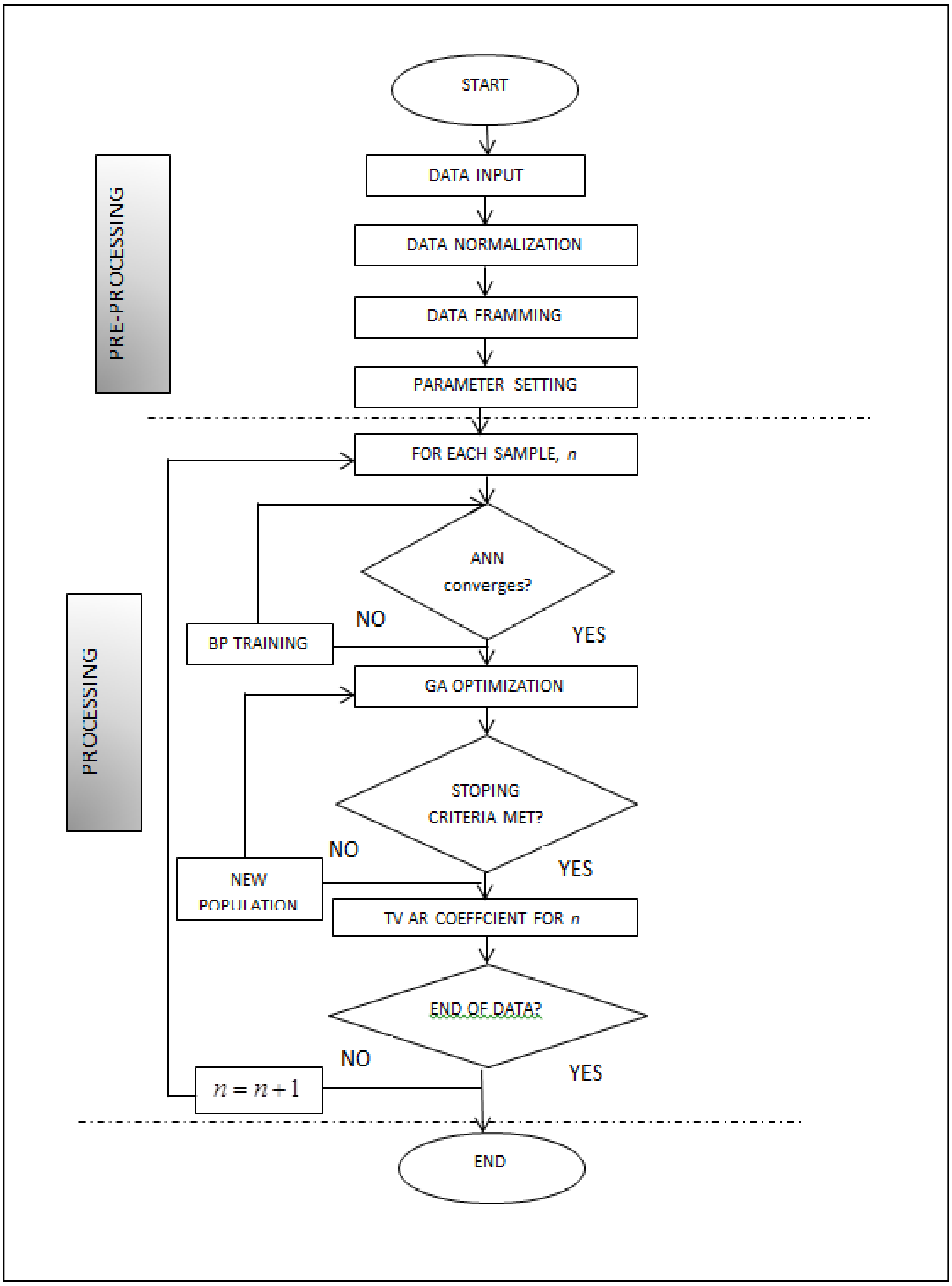


Figure 3.5 Hybrid BP-ANN-GA Implementation Flowchart

Basically normalization removes biasness which may contain in an input signal. The most common normalization methods are listed below:

Z score normalization

$$X_{new} = \frac{x - \bar{x}}{\sigma_x^2} \quad (3.27a)$$

MinMax Normalization

$$X_{new} = \frac{x - x_{\min}}{x_{\max} - x_{\min}} \quad (3.27b)$$

Unitary Normalization

$$X_{new} = \frac{x}{x_{\max}} \quad (3.27c)$$

where $x_{\min}$ and $x_{\max}$ are the minimum and maximum values of the data respectively. $\bar{x}$ is the mean value of the data and σ_x^2 is the variance of the dataset.

MinMax and Unitary data normalization of a multicomponent artificial NSS as in Equation (3.28) is shown in Figure 3.6. Both normalizations retain the signal structure but their values have been changed. MSE of 0.3461 and 0.8166 is computed for MinMax and Unitary.

$$\begin{cases} x(t) = \cos(2\pi f_1 t) & 0 \le t \le 100 \text{ ms} \\ x(t) = \cos(2\pi f_1 t) + \cos(2\pi f_2 t) & 100 \le t \le 200 \text{ ms} \\ x(t) = \cos(2\pi f_1 t) + \cos(2\pi f_2 t) + \cos(2\pi f_3 t) & 200 \le t \le 300 \text{ ms} \\ x(t) = \cos(2\pi f_4 t) & 300 \le t \le 400 \text{ ms} \end{cases} \quad (3.29)$$

II. Data Framing

While data normalization is to improve the signal content , data framing is concerned with structural representation of the input signal whereby the lengthy input data is truncated into blocks for easier and faster processing.

The long data is framed by size of model orders into a matrix form. Figure 3.7 shows data framing of a random data with size of 15 and model order 4.

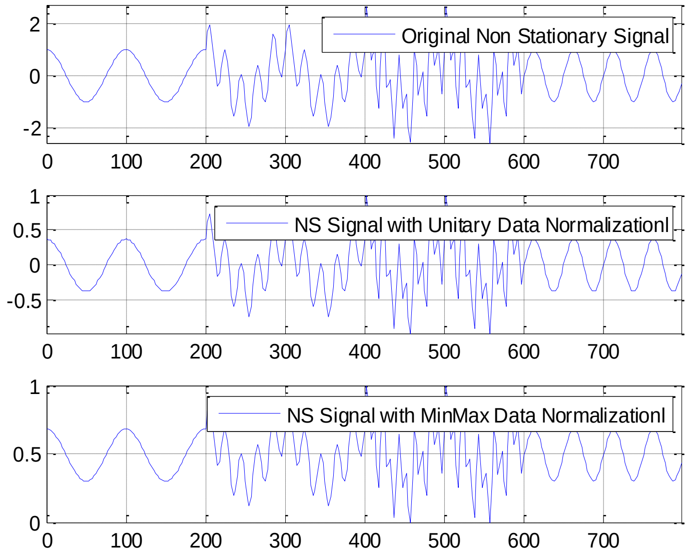
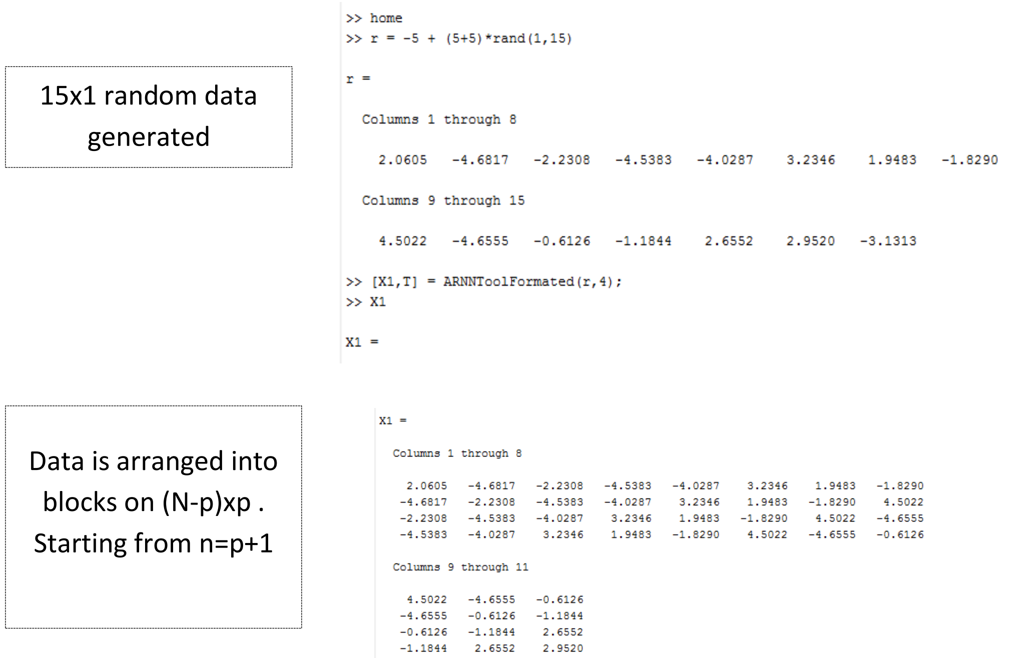


Figure 3.6 Comparing Data Normalization of NSS




```
>> home
>> r = -5 + (5+5)*rand(1,15)

r =

  Columns 1 through 8

    2.0605   -4.6817   -2.2308   -4.5383   -4.0287    3.2346    1.9483   -1.8290

  Columns 9 through 15

    4.5022   -4.6555   -0.6126   -1.1844    2.6552    2.9520   -3.1313

>> [X1,T] = ARNNToolFormated(r,4);
>> X1

X1 =
```

```
X1 =

  Columns 1 through 8

    2.0605   -4.6817   -2.2308   -4.5383   -4.0287    3.2346    1.9483   -1.8290
   -4.6817   -2.2308   -4.5383   -4.0287    3.2346    1.9483   -1.8290    4.5022
   -2.2308   -4.5383   -4.0287    3.2346    1.9483   -1.8290    4.5022   -4.6555
   -4.5383   -4.0287    3.2346    1.9483   -1.8290    4.5022   -4.6555   -0.6126

  Columns 9 through 11

    4.5022   -4.6555   -0.6126
   -4.6555   -0.6126   -1.1844
   -0.6126   -1.1844    2.6552
   -1.1844    2.6552    2.9520
```

Figure 3.7 Sample of Data Framing

III. BP-ANN training

After data normalization and framing, sequence of input data $y(n-1), y(n-2), \cdots, y(n-p)$ will be fed into BP-ANN as input nodes to estimate $a_j[n]$ as been presented in Subsection 3.3.2 and 3.3.3.

IV. TVAR coefficient optimization using GA

The estimated $a_j[n]$ are further optimized using GA as been described in Section 3.3.

3.5 IMPLEMENTATION RESOURCES

The proposed algorithm is implemented using Matlab® Version 8.10.0.604 (R2013a) using a HP EliteBook 8440p Dual Core I5 M540 @2.53 GHz with 6 GB memory. The source codes are attached in Appendix.

Two types of data were utilized to validate the proposed methods which are:

I. Artificial NSSs

Examples such as chirp signal, frequency jump signals, and frequency changing multicomponent signals with unknown TVAR coefficients.

II. Biomedical Signal Data

A set of Phonocardiogram (PCG) raw data recorded on clinical environment was contributed by Associate Prof. Dr. Samuel Schmidt from Department of Health Science and Technology, Aalborg University, Aalborg, Denmark. Detailed description of these data downloading and clinical setup is detailed in (Castro, Moukadem, Schmidt, Dieterlen, & Coimbra, 2015)

Other biomedical data set such as EEG which was contributed by Prof John Guttag, Dr. Zaeeshan and colleagues (Liu et al., 2016) also downloaded from The Massachusetts Institute of Technology heart sounds database (MITHSDB) contributed by.

3.6 VALIDATION OF PROPOSED METHOD

A multicomponent sinusoidal NSS as in Equation (3.29) which contains 4 frequencies 10Hz, 50Hz, 100Hz and 20Hz has been used to verify the proposed method. This input signal is shown in Figure 3.8.

TVAR coefficients for the NSS are estimated from both BP-ANN and BP-ANN-GA to compare GA and without GA. It was tested for the model orders { p =5,10,25 and 35 } to stimulate low model order and high model order. While the expansion model order, q was set to data length. In the GA optimization, boundaries were set to 1.0.

Screenshots of the execution process is shown in Figure 3.9 and Figure 3.10. Figure 3.9(a) shows screenshot from Matlab® command prompt after simulations where results are displayed. While Figure 3.9(b) shows screenshot of GA optimization during execution process.

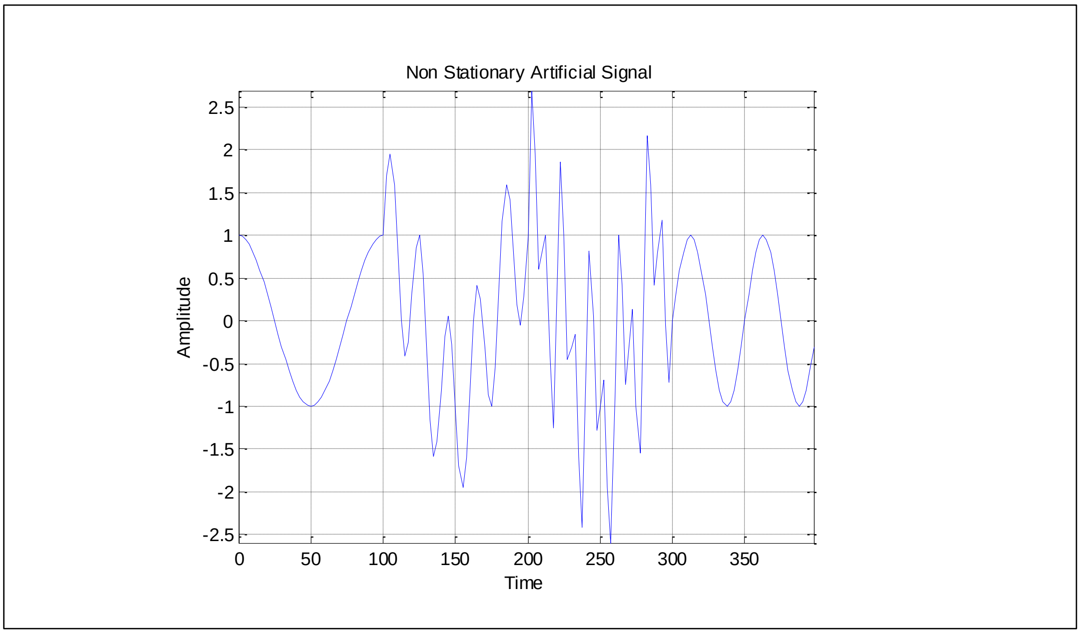


Figure 3.8 Multicomponent Artificial NSS

The first ten values of estimated TVAR coefficients at data sequence *n=50* for various model orders are shown in Table 3.4. The MSE between original signal and estimated signal for various model orders of both BP-ANN and BP-ANN-GA are shown in Table 3.5. Signal reconstructed from TVAR coefficients estimated from BP-ANN with BP-ANN-GA for various model orders are plotted in Figures 3.11, 3.12, 3.13 and 3.14. The Plot of MSE trajectories are shown in Figure 3.15.

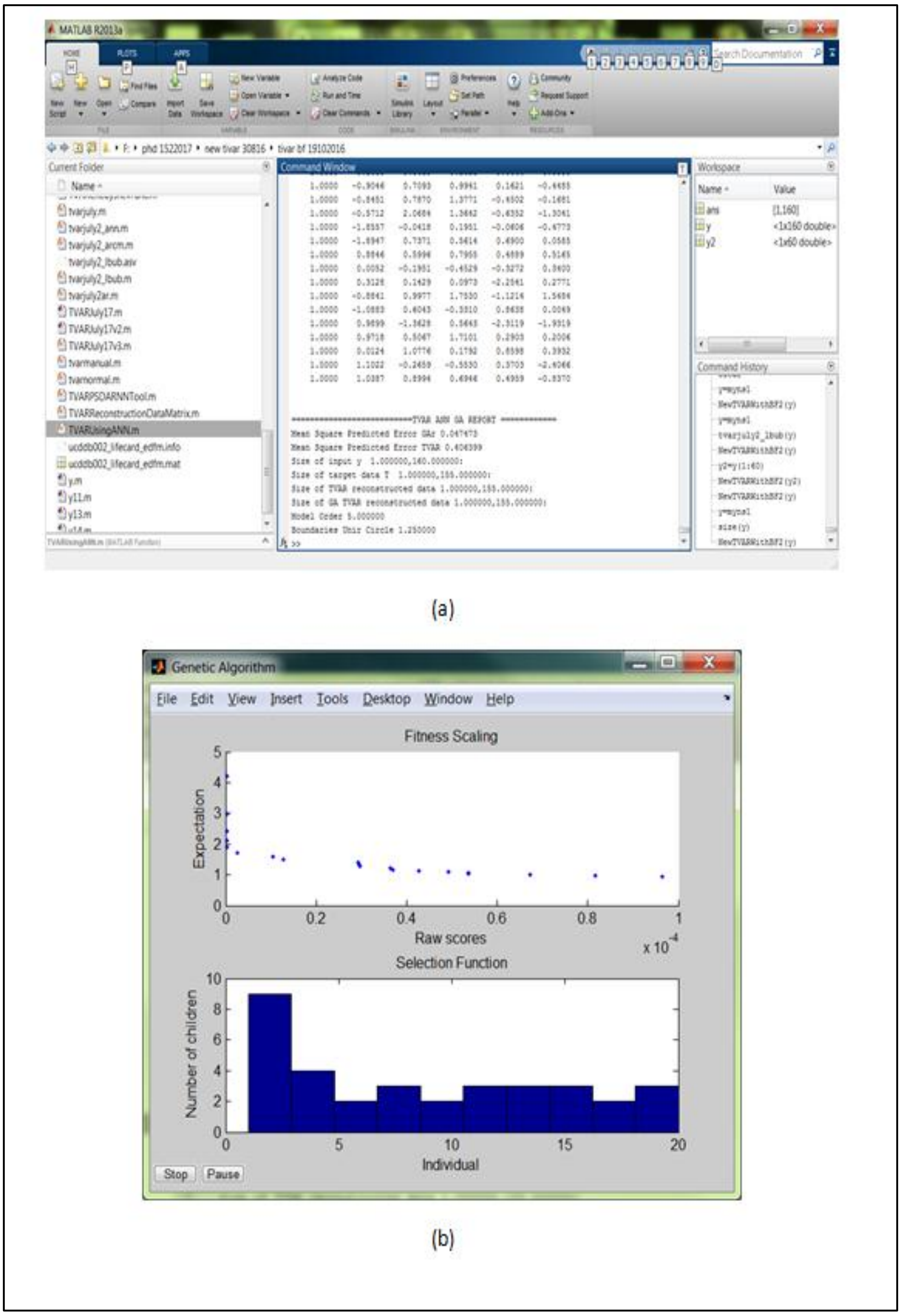


Figure 3.9 Execution Screenshots; (a) Matlab command prompt (b) GA Progress

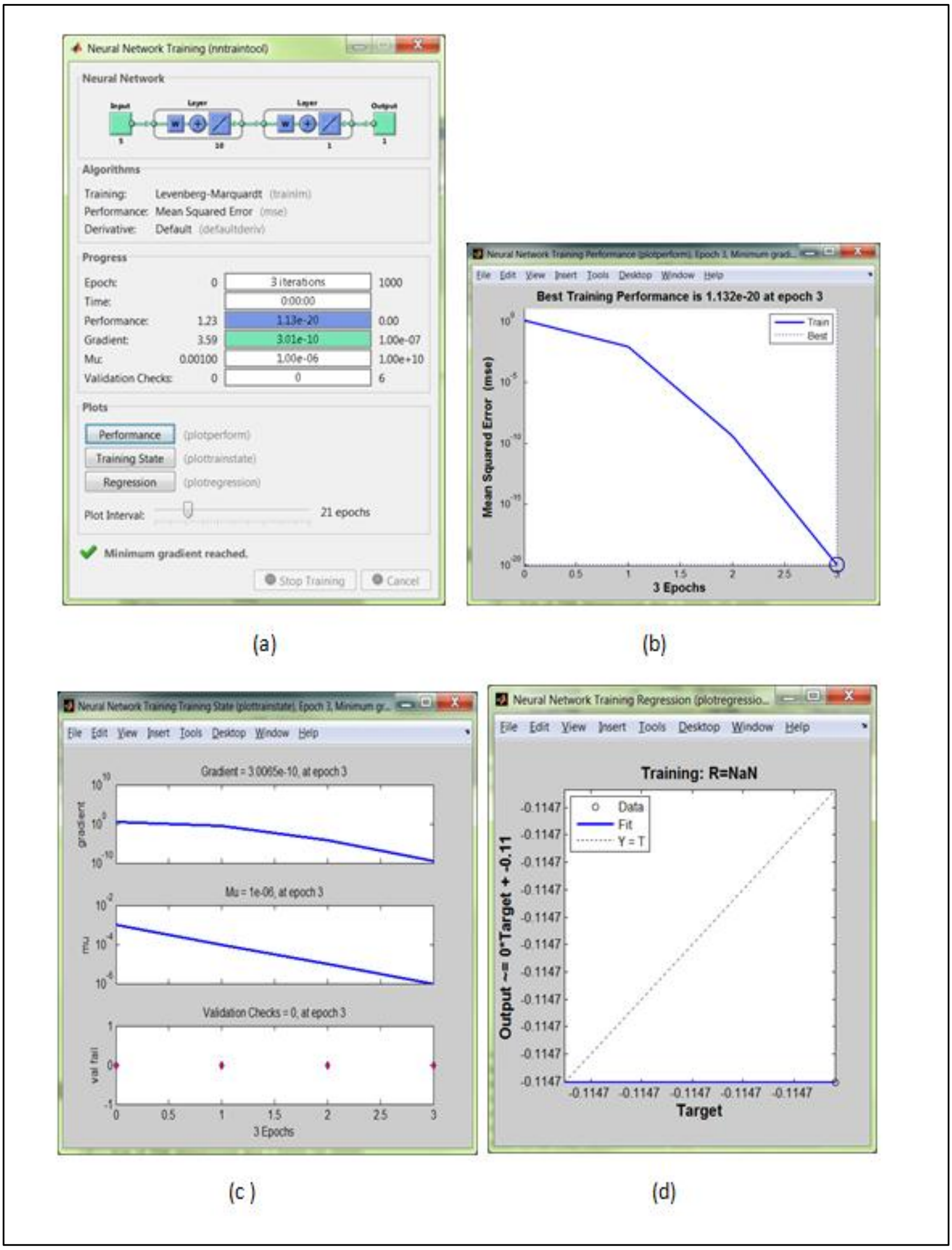


Figure 3.10 BP Learning Process (a) Summary of result (b), (c), (d) Performance plots

Table 3.4 Time Varying Coefficients for n=50 estimated using proposed method

METHOD		11 first values TVAR Coefficients for n=50										
Model Order	Method	$a_1[50]$	$a_2[50]$	$a_3[50]$	$a_4[50]$	$a_5[50]$	$a_6[50]$	$a_7[50]$	$a_8[50]$	$a_9[50]$	$a_{10}[50]$	$a_{11}[50]$
5	BP-ANN	0.8927	0.6010	0.3485	-0.5188	-0.1775	-	-	-	-	-	-
	BP-ANN-GA	0.8411	0.9071	0.1751	0.6973	-0.6521	-	-	-	-	-	-
10	BP-ANN	-1.7955	2.2194	-2.1760	-1.5886	0.1986	2.2399	0.6365	-1.4865	1.4460	-0.0906	-
	BP-ANN-GA	-3.0242	0.9922	-1.3214	-1.8428	1.3044	0.9933	0.3887	-0.3420	0.3587	0.6534	-
25	BP-ANN	-0.2580	0.1498	2.3948	0.8317	-1.1812	1.3044	-0.4155	1.2479	1.7392	1.6325	-0.7800
	BP-ANN-GA	0.7556	0.9203	3.2043	1.9536	0.0284	1.6034	0.0108	0.5411	1.1899	0.8350	-0.0988
35	BP-ANN	0.8003	-0.5374	-0.2586	-0.6572	-0.8365	1.0564	1.1575	0.2311	-0.7665	1.2201	-0.8648
	BP-ANN-GA	0.6745	0.1335	0.7029	0.3628	0.3832	0.9565	0.8146	0.1327	0.4353	0.3192	0.1897

Table 3.5 MSE of BP-ANN and BP-ANN-GA

Model Order ,p	BP ANN	BP ANN GA
5	0.406399	0.047473
10	0.552687	0.002364
25	1.28055	0.000165
35	1.828706	0.00012

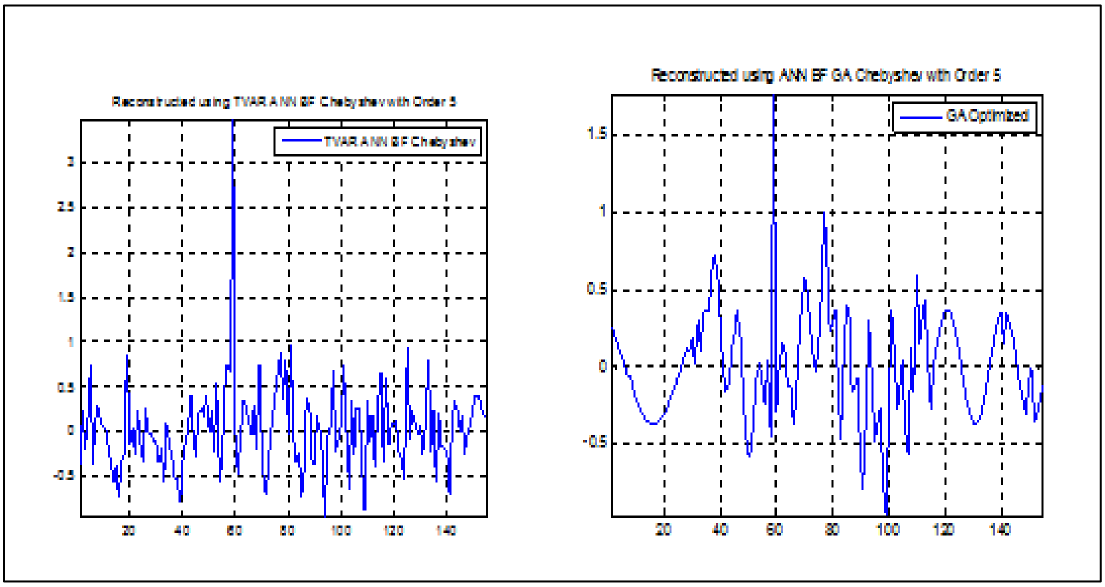


Figure 3.11 Reconstructed NSS for Order 5 (a) BP-ANN (b) BP-ANN -GA

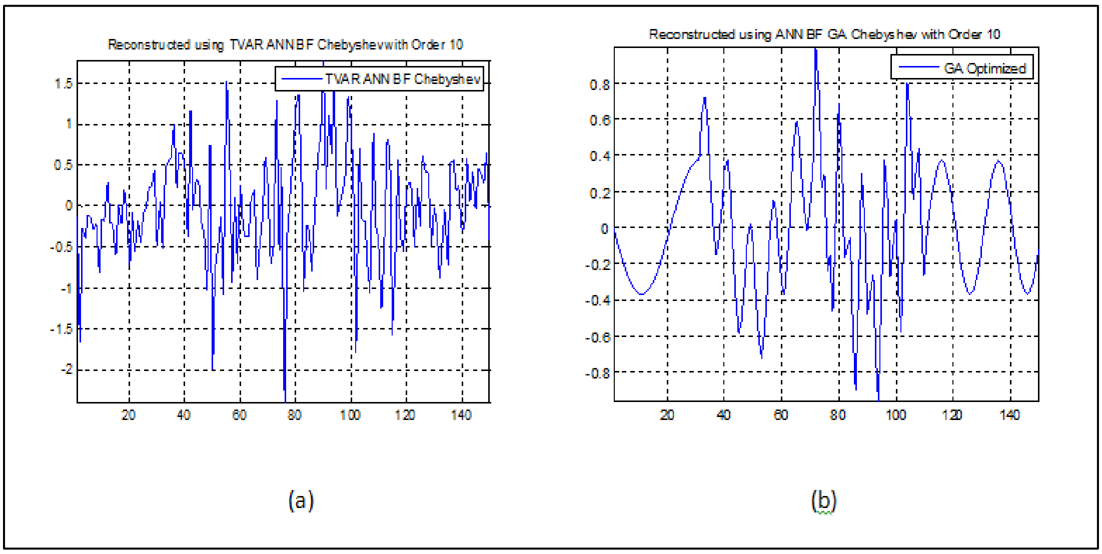


Figure 3.12 Reconstructed NSS for Order 10 (a) BP-ANN (b) BP-ANN-GA

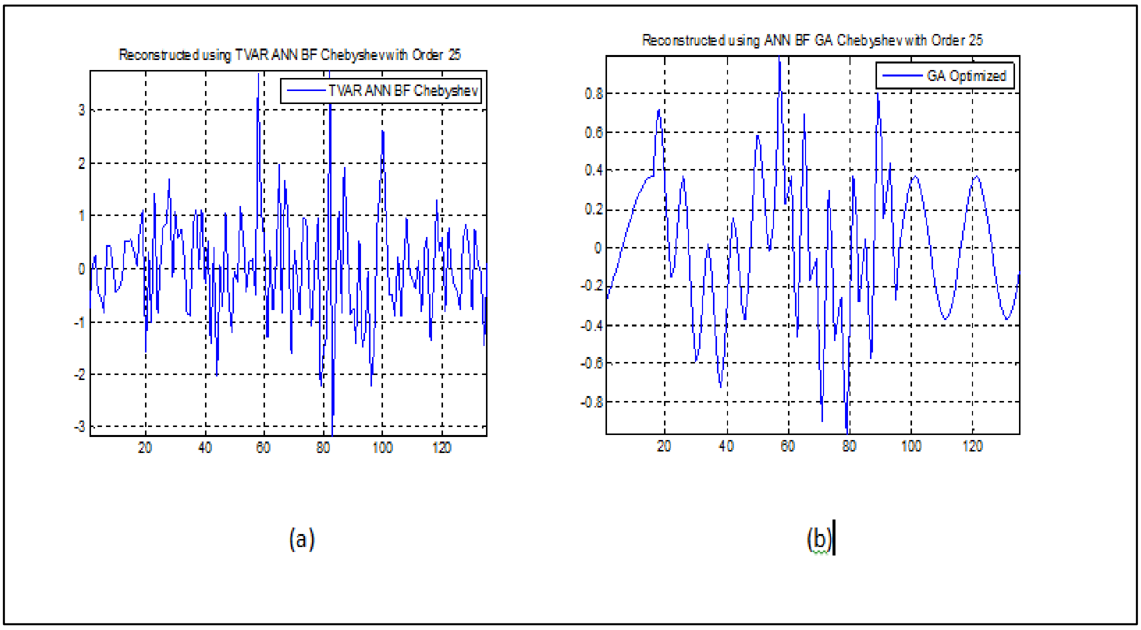


Figure 3.13 Reconstructed NSS for Order 25 (a) BP-ANN (b) BP-ANN-GA

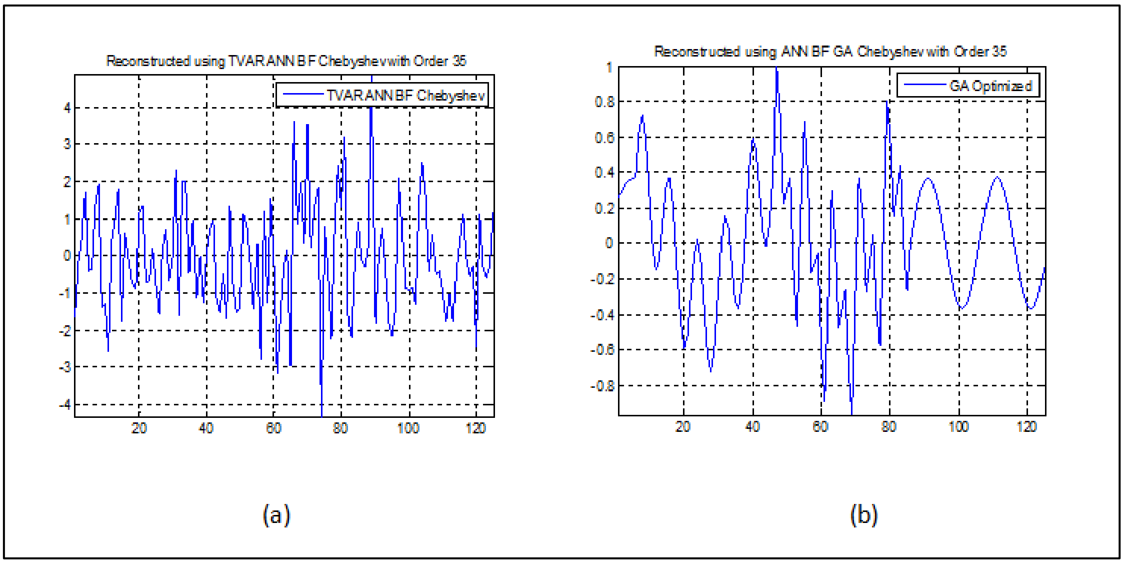


Figure 3.14 Reconstruction NSS for Order 35 (a) BP-ANN (b) BP-ANN-GA

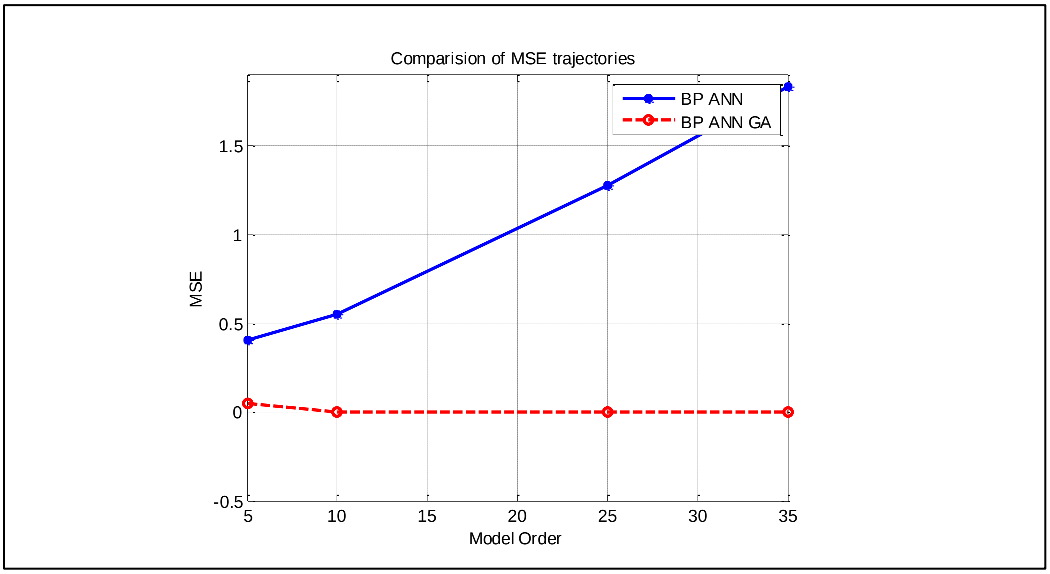


Figure 3.15 MSE Trajectories of BP-ANN and BP-ANN-GA

The results from Tables 3.4, Table 3.5 and Figures 3.11 – 3.15 above confirm and validate the accuracy of the proposed method BP-ANN-GA when compared to BP-ANN in estimating the TVAR coefficients. This is due to:

I. MSE trajectory of BP-ANN is not stable where it increases with model order and hence demonstrated model orders dependence characteristic as seen in Figure 3.15. While, for all model orders, the MSE trajectory of BP-ANN-GA is stable and independent from model orders. Therefore the ability of GA to optimize estimated TVAR coefficients are successfully demonstrated here.

II. From Figures 3.11 until 3.14, it is noted that the predicted signal using BP-ANN-GA has similar wave form and amplitude with original signal. While the signal predicted from BP-ANN is distorted as it is corrupted with noise due to inaccurate TVAR coefficients produced by inaccurate model orders.

3.7 SUMMARY

An innovative method to estimate TVAR coefficients using ANN and GA has been presented in this chapter. One significant advantage of the proposed technique, BP-ANN-GA is it does not involve deriving a *p X m* matrix to represent TVAR coefficient as in Correlation or Covariance methods. This has eliminated the need to develop complex mathematical procedure to estimate the TVAR coefficients through recursive rule based algorithms. As the matter of fact, the TVAR coefficients are now estimated by using an internal mechanism which links the TVAR coefficients to synaptic weight of BP-ANN topology. Thus, the rule base algorithm in conventional method is now converted to ANN optimization problem. Two three layer BP-ANN network architecture is proposed in Section 3.3 to represent Direct TVAR and TVAR BF approach. Relationship between synaptic weights of BP-ANN and TVAR coefficient is established and shown in section 3.3.1 and 3.3.2. A LM based, rule of weight updating for BP training for proposed BP-ANN is discussed in section 3.3.3. While a scheme for GA optimization to further optimize synoptic weight is proposed in section 3.3.4 as well. The implementation of the newly proposed BP-ANN-GA has been explained with a signal flow diagram and implementation flowchart in Section 3.4. In Section 3.5, we have demonstrated that the proposed BP-ANN-GA algorithm outperforms the BP-ANN in estimating TVAR coefficients. The proposed method is validated by showing its ability in reconstructing the original signal from the estimated TVAR coefficients.

CHAPTER FOUR

RESULTS AND DISCUSSIONS

4.1 INTRODUCTION

The results of implementation of the proposed method, BP-ANN-GA which was developed in Chapter Three are presented in this chapter. The proposed algorithm is investigated for its 'signal independence' characteristics such as independence from influence of model orders and BFs. This is done by estimating the TVAR coefficients and using it to reconstruct the original signal from the estimated TVAR coefficients. The performance of the proposed method is then compared with various methods. The NS characteristics of input signal are represented by both computer stimulated NSS and biomedical signals.

The chapter starts by examining the effects of varying the model orders and the effect of imposing various limits on GA design variables. The results obtained here are then used as a 'standard' parameter for further analysis of the proposed method in Section 4.4. In Section 4.4 various NSSs are reconstructed using the TVAR coefficients estimated from various methods. Performances are measured using various performance metrics as presented in Section 4.4.1. These metrics measure how close is the predicted signal to the original signal, hence reflecting the accuracy of the estimated TVAR coefficients. Signal reconstructed from the optimum TVAR coefficients will estimate the more accurate and as closer to the original signal.

4.2 EFFECTS OF VARYING THE LIMITS ON GA DESIGN VARIABLE

GA its input parameters which are chosen prior the optimization process begin. These parameters are listed in Table 3.4, among which are variables listed as *ub* (upper bound) and *lb* (lower bound). Basically *ub* and *lb* represent lower and upper limits that should be imposed on the design variables; x to produce an estimated solution in the range between $lb \leq x \leq ub$. Although *ub* and *lb* could take any real numbers, however, it is impractical to allow the GA optimized TVAR coefficients to go beyond any arbitrary number as this will lead to an unstable system and also unnecessary prolonged iteration in GA. Therefore, the objective in this section is to investigate the effects of various limits in order to identify an optimal boundary limit for which GA will work efficiently and effectively. Testing for boundaries was simultaneously carried with various model order and the results are presented in next section.

4.3 EFFECT OF MODEL ORDERS ON BP-ANN-GA

It is commonly known that the accuracy of the estimated coefficients of any parametric modeling heavily depends on model orders. In addition, they also determine the number of coefficients to be estimated. An optimal model order will produce coefficients which contain the characteristics of the input signal. Hence, model orders became important criteria that should not be neglected in any TVAR modeling.

Inappropriate coefficients with noise or insufficient coefficients will be obtained if the model orders are too low (under parameterized) or too high (over parameterized). In the conventional method, optimum model orders are obtained from an exhaustive iterative search which is time consuming and involves heavy computations.

Various model orders which represent insufficient model order and over-parameterized model orders are tested on BP-ANN-GA. It is conjectured that the proposed BP-ANN-GA is independent from the model orders while the optimum boundary value are to be investigated.

4.3.1 Computer Simulated NSSs

Four NSS of various dynamics are selected to analyze the accuracy of the estimated TVAR coefficients from the proposed method using various GA limits and various model orders. The first three signals are computer generated artificial NSS with known frequencies, while the last signal is truncated from an eight second PCG of a healthy subject. Details of the input signal are briefed below.

Signal 1(S1) is an artificial multicomponent NSS which as expressed by Equation (4.1) as:

$$\begin{cases} x_1(t) = \cos(2\pi f_1 t) & 0 \le t \le 100 \text{ ms} \\ x_2(t) = \cos(2\pi f_1 t) + \cos(2\pi f_2 t) & 100 \le t \le 200 \text{ ms} \\ x_3(t) = \cos(2\pi f_1 t) + \cos(2\pi f_2 t) + \cos(2\pi f_3 t) & 200 \le t \le 300 \text{ ms} \\ x_4(t) = \cos(2\pi f_4 t) & 300 \le t \le 400 \text{ ms} \end{cases}$$

$$x(t) = x_1(t) + x_2(\text{t}) + x_3(\text{t}) + x_4(\text{t}) \quad (4.1)$$

Signal 2 (S2) is multicomponent sine wave with equation as expressed by Equation (4.2)

$$x(t) = A\sin(2\pi f_1 t) + B\sin(2\pi f_2 t) + C\sin(2\pi f_3 t)$$
$$where \;\; f_1 = 5Hz, \;\; f_2 = 20Hz, \;\; f_3 = 10 \;\;\text{ with } fs = 500Hz \text{ and } A\,and\;B = 1 \tag{4.2}$$

Signal 3 (S3) is fast changing multiple sine waves with added noise expressed by Equation 4.3:

$$x(t) = \sin(2\pi(f_1 + at)t) + \sin(2\pi(f_{21} + at)t) + \sin(2\pi t)\sin(2\pi(f_3 + at)t)$$

$$with\; t = 0:0.1:0.511; \;\; a = 100; \;\; f_1 = 200Hz, \; f_2 = 220Hz; \; f_3 = 100Hz \tag{4.3}$$

Signal 4 (S4) is a biomedical signal truncated from a health PCG signal which was recorded for eight seconds. The original PCG which and has total of 32000 samples is truncated to 200 samples with approximate of five milliseconds recordings. The frequency content in S4 is insignificant as the objective of these simulations is to investigate the response of the proposed algorithm on model orders and boundary limits as mentioned earlier. A temporal representation of those input NSS S1, S2, S3 and S4 are shown in Figure 4.1.

4.3.2 Performance Analysis of Various Limits and Model Orders

The performance of the BP-ANN algorithm is evaluated for 5 different model orders, which are 3, 7, 10, 15 and 20 to represent a low model order till a high model order. The BP-ANN-GA algorithm is further evaluated for the same model orders and also for a range of boundary values from 0.25 to 2.25 with an interval 0.25 for each model order. In another word, the BP-ANN-GA algorithm is tested for changing boundaries on design variables where by $lb = a_j[n] - b$ and $ub = a_j[n] + b$; where $b = [0.25, 0.5, \cdots 2.25]$.

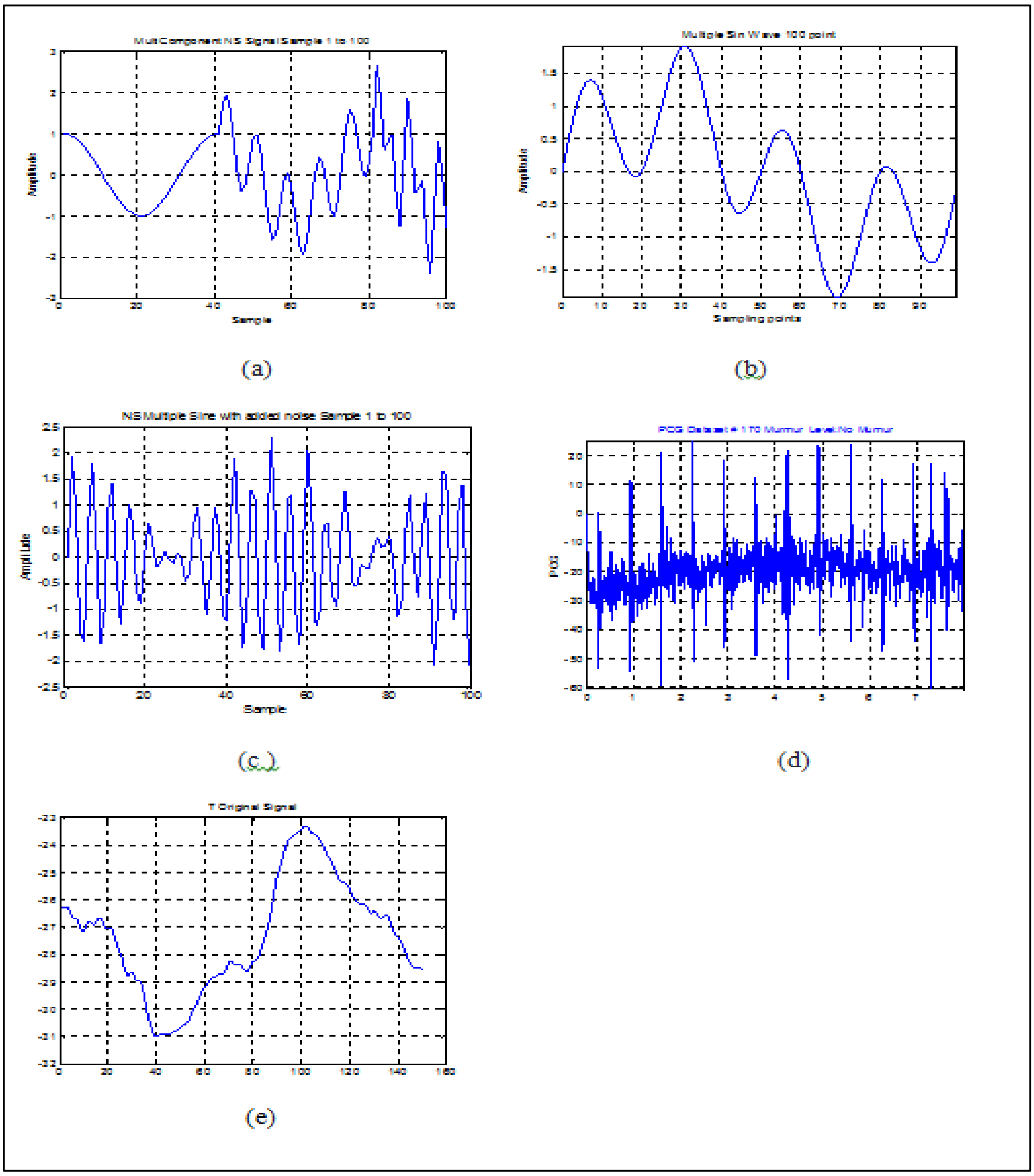


Figure 4.1 Artificial NSS (a) S1 (b) S2 (c) S3 (d) PCG Signal and (e) S4

Table 4.1 S1 MSE comparison for BP-ANN and BP-ANN-GA

Model Orders	3		7		10		15		20	
Range (-lb,ub)	BPANN	BPANN GA	BPANN	BPANN GA	BPANN	BPANN GA	BPANN	BPANN GA	BPANN	BPANN GA
0.25	0.883	0.479	1.152	0.618	1.562	0.425	2.613	0.678	3.061	0.743
0.50	0.690	0.174	1.491	0.199	1.854	0.185	2.496	0.034	3.830	0.254
0.75	0.790	0.126	1.178	0.022	1.861	0.023	2.580	0.017	2.683	0.023
1.0	0.826	0.068	1.378	0.019	1.912	0.021	2.820	0.008	2.989	0.001
1.25	0.636	0.017	1.428	0.011	1.625	0.001	2.310	0.001	3.021	0.002
1.5	0.822	0.006	1.152	0.006	1.670	0.001	1.710	0.002	3.269	0.001
1.75	0.724	0.001	1.491	0.051	1.867	0.001	2.680	0.001	2.821	0.003
2.0	0.810	0.007	1.390	0.007	1.750	0.001	2.020	0.003	2.761	0.005
2.25	0.763	0.004	1.400	0.018	1.890	0.001	2.361	0.001	3.491	0.001
Average	0.772	N.A.	1.34	N.A.	1.777	N.A.	2.399	N.A.	3.102	N.A

Table 4.2 S2 MSE comparison for BP-ANN and BP-ANN-GA

Model Orders	3		7		10		15		20	
Range (-lb,ub)	BPANN	BPANN GA	BPANN	BPANN GA	BPANN	BPANN GA	BPANN	BPANN GA	BPANN	BPANN GA
0.25	1.064	0.472	1.175	0.308	1.749	0.616	2.277	0.556	1.822	0.803
0.5	1.085	0.146	1.099	0.097	1.615	0.172	2.765	0.253	0.999	0.074
0.75	1.047	0.032	1.129	0.006	1.556	0.017	2.317	0.015	1.144	0.003
1.0	1.055	0.015	1.269	0.008	1.463	0.002	3.168	0.001	1.539	0.001
1.25	1.159	0.003	1.111	0.001	1.463	0.001	1.637	0.001	1.967	0.001
1.5	1.019	0.001	1.445	0.000	1.670	0.001	2.790	0.002	1.135	0.001
1.75	1.065	0.002	1.192	0.000	1.266	0.000	1.989	0.001	1.809	0.001
2.0	0.969	0.001	1.292	0.000	1.477	0.001	2.013	0.001	1.081	0.001
2.25	0.976	0.001	1.257	0.000	1.349	0.00	1.989	0.001	1.738	0.001
Average	1.049	N.A.	1.219	N.A.	1.512	N.A.	2.327	N.A.	1.471	N.A.

Table 4.3 S3 MSE comparison for BP-ANN and BP-ANN-GA

Model Orders	3		7		10		15		20	
Range (-lb,ub)	BPANN	BPANN GA	BPANN	BPANN GA	BPANN	BPANN GA	BPANN	BPANN GA	BPANN	BPANN GA
0.25	0.730	0.434	1.241	0.538	1.201	0.320	2.243	0.619	2.700	0.785
0.5	0.722	0.219	1.490	0.348	1.220	0.081	1.694	0.039	3.060	0.097
0.75	0.789	0.119	1.276	0.095	0.958	0.001	1.622	0.021	2.001	0.003
1.0	0.626	0.031	0.956	0.035	1.129	0.001	1.648	0.015	2.037	0.003
1.25	0.873	0.041	0.967	0.003	1.008	0.001	1.876	0.022	2.475	0.001
1.5	0.683	0.001	0.940	0.001	1.011	.000	2.161	0.001	2.385	0.001
1.75	0.627	0.001	1.074	0.002	1.689	0.001	1.941	0.003	2.240	0.001
2.0	0.598	0.001	0.872	0.002	1.086	0.001	1.213	0.001	2.085	0.002
2.25	0.670	0.000	0.954	0.001	1.183	0.001	1.350	0.002	2.713	0.001
Average	0.702	N.A.	1.081	N.A.	1.165	N.A.	1.750	N.A.	2.410	N.A.

Table 4.4 S4 MSE comparison for BP-ANN and BP-ANN-GA

Model Orders	3		7		10		15		20	
Range (-lb,ub)	BPANN	BPANN GA	BPANN	BPANN GA	BPANN	BPANN GA	BPANN	BPANN GA	BPANN	BPANN GA
0.25	5.435	2.130	5.290	0.104	5.239	0.014	5.051	0.674	4.839	0.075
0.5	5.440	0.348	5.299	0.007	5.208	0.04	4.960	0.249	4.737	0.072
0.75	5.430	0.016	5.270	0.003	5.194	0.07	5.040	0.070	4.856	0.021
1.0	5.432	0.001	5.300	0.004	5.170	0.03	5.005	0.04	4.788	0.006
1.25	5.441	0.002	5.291	0.001	5.239	0.05	4.971	0.067	4.908	0.005
1.5	5.434	0.001	5.281	0.001	5.149	0.01	5.009	0.02	4.891	0.006
1.75	5.440	0.003	5.316	0.001	5.176	0.01	4.976	0.02	4.916	0.007
2.0	5.430	0.001	5.311	0.003	5.183	0.01	5.041	0.01	4.756	0.001
2.25	5.440	0.001	5.303	0.001	5.209	0.02	5.013	0.01	4.965	0.074
Average	5.436	N.A.	5.296	N.A.	5.196	N.A.	5.000	N.A.	4.851	N.A.

The TVAR coefficients which are estimated from both methods are later used to reconstruct the input signal. The MSE between the input signal and the estimated signal are computed and shown in Table 4.1, Table 4.2, Table 4.3 and Table 4.4 for signals S1, S2, S3 and S4 respectively.

Since the BP-ANN is not subjected to *ub* and *lb*, they are unintentionally been tested repeatedly nine times for each input signal together while searching for optimum limits for GA algorithm. The average MSE for each model order is computed and shown in Figure 4.2.

From Figure 4.2, we could notice that the BP-ANN algorithm is dependent on the model orders. From Table 4.2, for model orders 3, 7, 10, 15 and 20, S1 shows increasing values of average MSE starts from 1.049, 1.219, 1.512, 2.327, and 1.471 respectively. Similar pattern was observed for S2 and S3 where average MSE increases with model order. While the S4's average MSE shows decrement from 5.436, 5.296, 5.196, 5.0 and 4.851 with the same model orders.

Basically, for all input signals, MSE from BP-ANN algorithm are not converging to a solution; which is an indication for on-going exhausted searching process. Hence, the inaccurate model orders of the BP-ANN algorithm are not estimating the coefficients accurately. Furthermore, the BP based learning rule has tendency to converge to local minima stopping criteria, which falsely produces an inaccurate TVAR coefficients.

The MSE obtained for various boundaries and various model orders using BP-ANN-GA algorithm is also shown in Table 4.1, 4.2, 4.3 and 4.4 for S1, S2, S3 and S4 respectively. Figure 4.3, Figure 4.4, Figure 4.5 and Figure 4.6 shows MSE trajectories for model orders 3,7,10,15 and 20 for boundary values, $b = 0.25, 0.5, 0.75, 1.0, 1.25, 1.5, 1.75, 2.0, 2.25, 2.5$.

It is observed in Figure 4.3 for all model orders the MSE decreases with increasing of *b* before becomes stable with approximate MSE 0.001 after limits of 1.0. Similar observation was observed for S2, S3 and S4 as noticed in Figure 4.4, Figure 4.5 and Figure 4.6.

The result of adopting GA for optimization is impressive as the observation of the MSE plot from Figures 4.3, 4.4, 4.5 and 4.6 shows that for any model order the BP-ANN-GA algorithm produces a MSE value of 0.001 for boundaries above 1.0. Therefore, the BP-ANN-GA algorithm demonstrated a unique characteristics which is independence from model orders when GA‘s solutions are allowed to vary in ± 1.0 or above for any model orders.

Hence, for this reason, the TVAR coefficients estimated from BP-ANN algorithm should be allowed to vary in this range, $a_j[n]-1 \le a_j[n] \le a_j[n]+1$ in GA optimization step. Therefore, the model order is set to 10 and boundary value is set to 1.0 as chosen value for these parameters for further analysis in this chapter. The comparison of input signal, S1 with the reconstructed signal using model order 10 and boundary 1.0 is shown in Figure 4.7. The other reconstructed signals; S2, S3 and S4 are not shown in here.

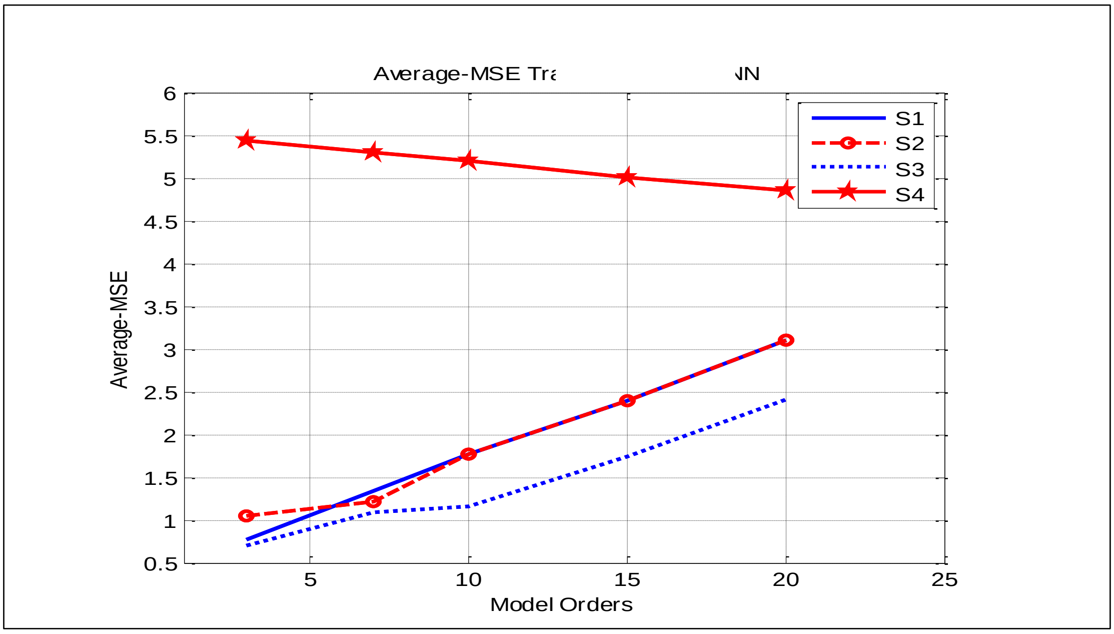


Figure 4. 2 Comparing MSE for BP-ANN for Various Model Orders

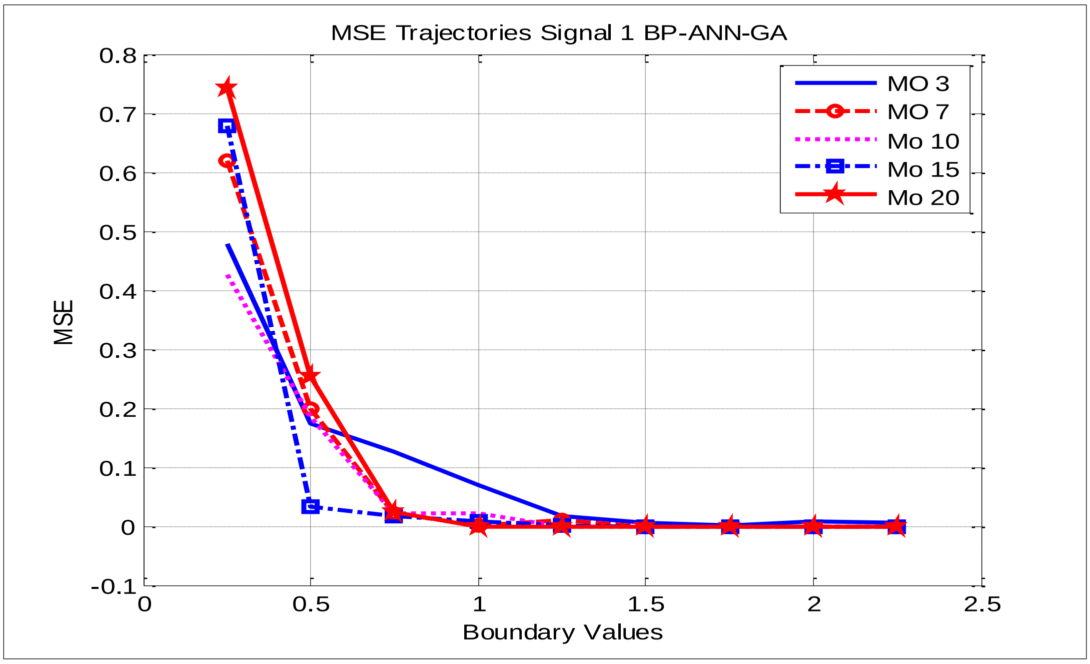


Figure 4.3 MSE Trajectories for S1 using BP-ANN-GA algorithm

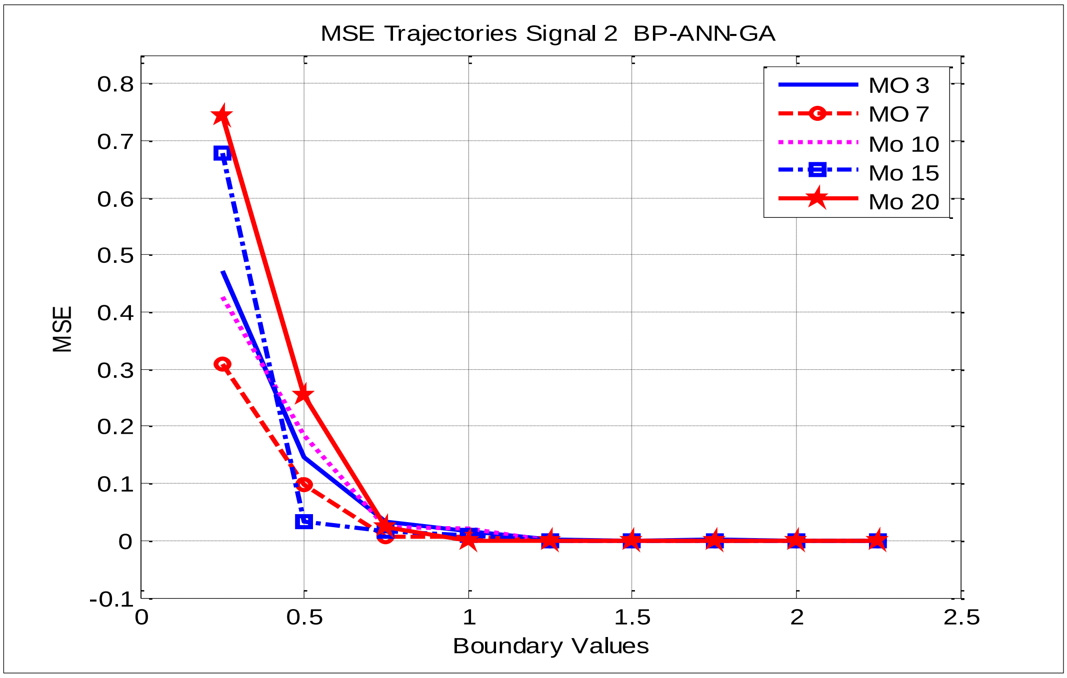


Figure 4.4 MSE Trajectories for S2 using BP-ANN-GA algorithm

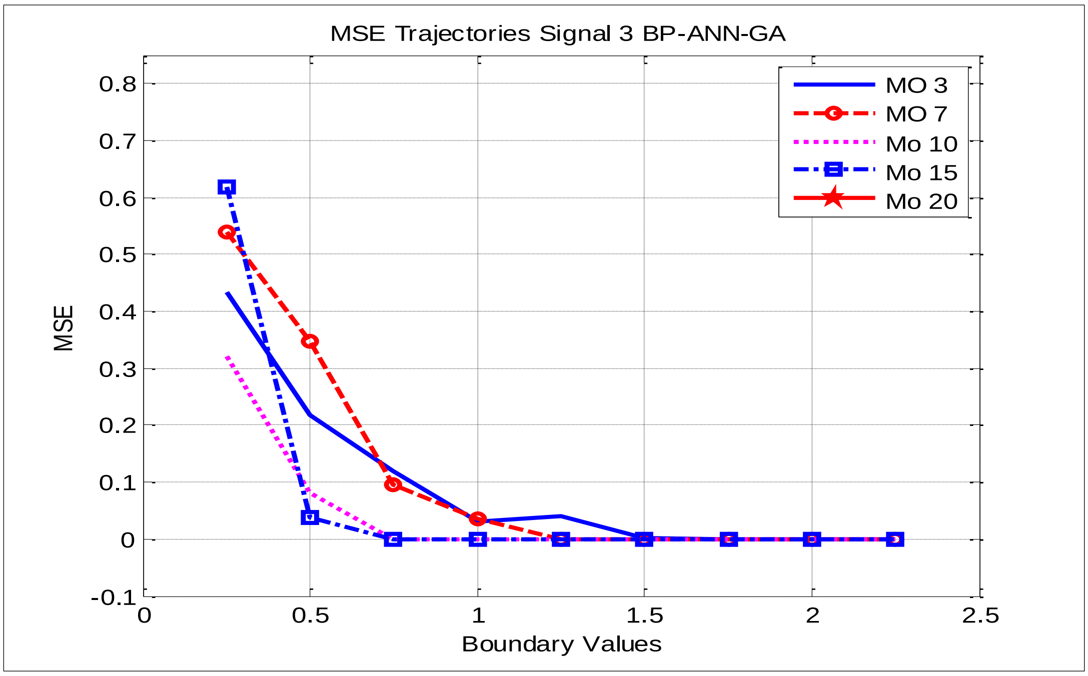


Figure 4.5 MSE Trajectories for S3 using BP-ANN-GA algorithm

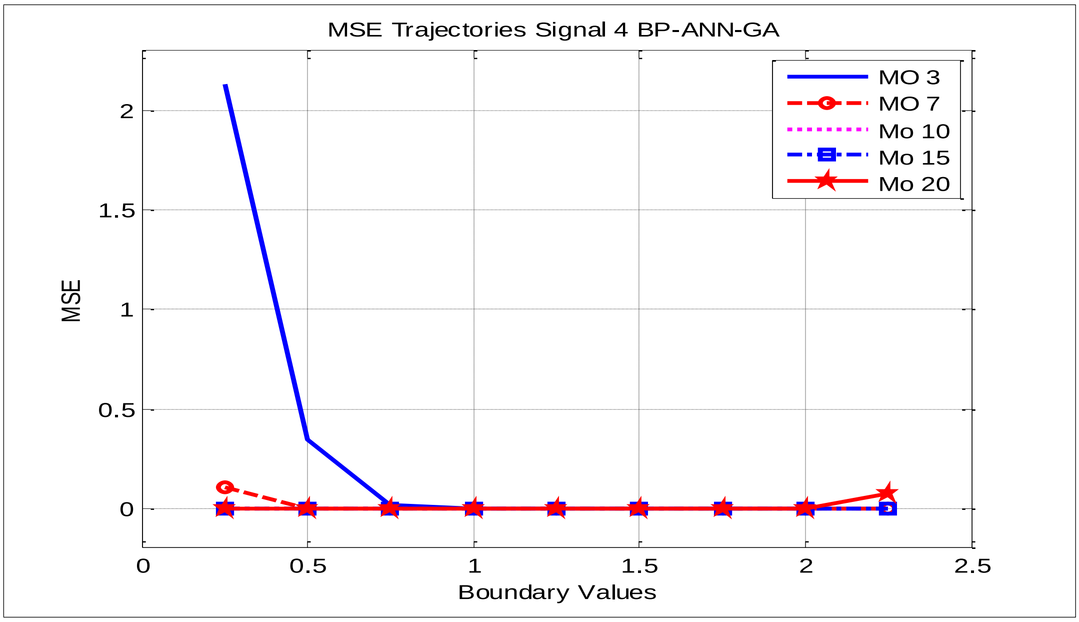


Figure 4.6 MSE Trajectories for S4 using BP-ANN-GA algorithm

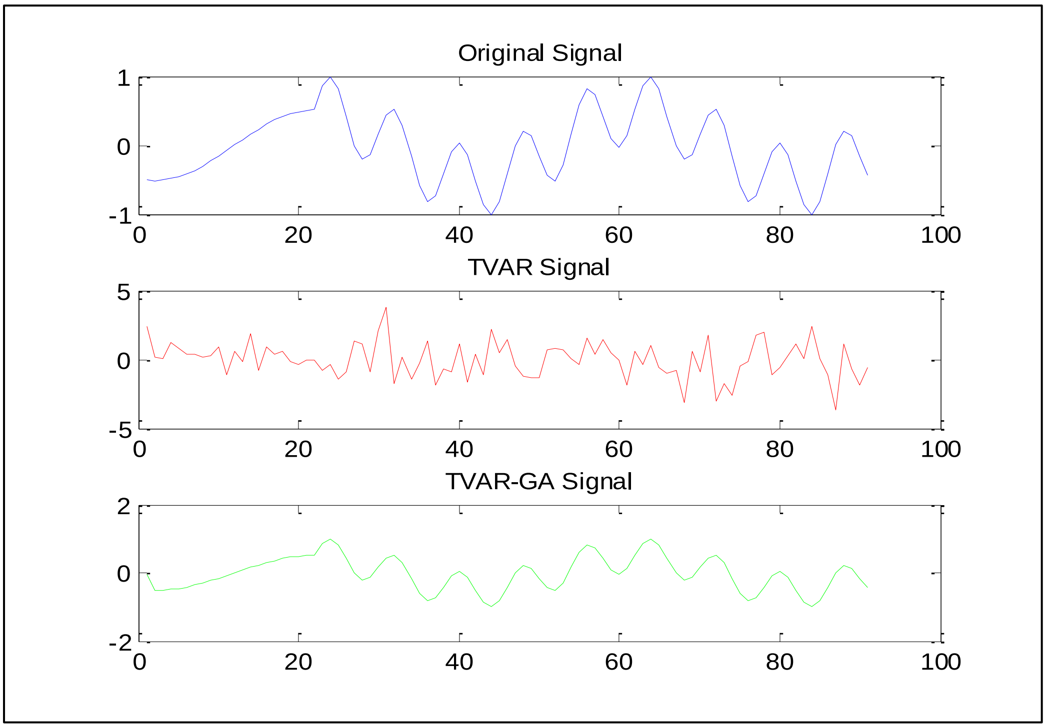


Figure 4.7 Comparison of Original and Reconstructed S1 Using BP-ANN-GA

Hence, it can be a conclusion that the BP-ANN-GA algorithm has improved the performances where a lower and stable MSE is observed when compared to BP-ANN alone. This observation can be explained by the fact that TVAR coefficients estimated using BP-ANN is inaccurate as BP algorithm often converges to local minima. These TVAR coefficients are then optimized using GA optimization process to produce more accurate TVAR coefficients.

4.4 NSS SIGNAL RECONSTRUCTION

Signal reconstruction or commonly known as signal recovery is one key performance index to measure the accuracy of estimated TVAR coefficients; that is to measure how well the estimated TVAR coefficients could reproduce the original signal. In general reconstruction or signal recovery is the process of assembling back the components into the original signal with least loss of information. In this section, we examine the accuracy of the estimated TVAR coefficients from various methods including proposed method in recovering the original signal. The original input signals are grouped into various types of computer stimulated (artificial) NSS and biomedical signals .Quality of reconstructed signals is assessed using metrics described in Section 4.4.1.

4.4.1 Performance Metrics

Five performance measurement indexes are employed to evaluate the signal reconstruction qualities which are the Mean Square Error (MSE) or also known as Mean Square Prediction Error (MSPE), the Correlation Coefficient (CC), the Root

Mean Square (RMS), the Percentage Root Mean Square Difference (PRD) and the Signal to Notice Ratio (SNR).

MSE a standard criterion for signal quality and fidelity is one of the most common error metrics in various problems due to its mathematical advantages. It is the natural way to define the energy of the error signal. Furthermore they are memoryless and computed at each sample. By denoting $x_i(n)$ as the input signal and $y_o(n)$ as the estimated signal of length, *N*, the MSE is given by:

$$\Psi = \frac{1}{N}\sum_{n=1}^{N}\left|(x_i(n) - y_i(n))\right|^2 \tag{4.4}$$

The best fit of Ψ occurs when the value is approximately equal to zero.

The CC is the strength of the linear relationship between $x_i(n)$ and $y_o(n)$ where it quantifies the closeness between both signals. A Pearson Product Moment Correlation Coefficient (PPMCC) which is used as a standard deviation is given by:

$$CC_{xy} = \frac{\sum_{n=1}^{N}(x_i - \hat{x})(y_o - \hat{y})}{(N-1)s_x s_y} \tag{4.5}$$

where, $\hat{x}, \hat{y}$ are the median of input signal and estimated signal , s_x and s_y are standard deviation of x_i and y_o.

The value of CC is always between -1 to +1, where the value +1 indicates highly correlated and a -1 implies highly not similar. Generally correlation greater than 0.8 is described as strong while less than 0.5 is described as weak.

The RMSE is used to measure the reconstructed signal distortion from input signal. Basically it is a measurement of the imperfection which measures how far an average error is from 0. The RMSE equation is given by:

$$RMSE_{xy} = \sqrt{\frac{\sum_{n=1}^{N}(x_i - y_o)^2}{N}} \tag{4.6}$$

A PRD is used to measure the distortion in a reconstructed signal with respect to the original signal in a signal reconstruction. The PRD equation is given by:

$$PRD_{xy} = 100\sqrt{\frac{\sum_{n=1}^{N}(x_i - y_o)^2}{\sum_{n=1}^{N}(x_i - \hat{x})^2}} \tag{4.7}$$

A SNR is used to analyze the noise factor in a signal. Higher the SNR lesser will be the noise. It is measured in decibel (dB) and defined by:

$$SNR_{xy} = 10\log_{10}\left(\frac{\sum_{n=1}^{N}[x_i(n)]^2}{\sum_{n=1}^{N}(x_i(n) - y_o(n))^2}\right) \tag{4.8}$$

4.4.2 Algorithms and Parameters

The TVAR Coefficient estimation algorithms and its parameters used are listed in Table 4.5. Six methods were used which are AR, TVAR Correlation, BP-ANN, BP-ANN-GA, BP-ANN Chebyshev and BP-ANN-GA Chebyshev.

Table 4.5 Parameters for Signal Reconstruction Analysis

Parameters	Values
Objective	To evaluate estimated TVAR coefficients from various methods including proposed method.
Input Signals	(1) Artificial NSS with various frequency contain and speed (2) Processed high resolution biomedical signals
Methods	(1) AR (10) representing stationary approach (2) TVAR(10) representing time varying correlation approach (3) BP-ANN (10) representing proposed method (4) BP-ANN-GA (10) with limits 1.0 to optimize TVAR coefficient estimated from(3) (5) BP-ANN for Chebyshev Basis function with expansion order 5 (6) BP-ANN-GA for Chebyshev Basis function with expansion order 5
Model Order	Arbitrary value , $p=10$ and $m=5$
Boundaries for GA	1.0
Performance Measurement Indexes	MSE, RMSE, CC, PRD and SNR
Results presentation	comparison tables, bar charts, temporal representation of reconstructed

4.4.3 Reconstruction of Artificial NSS from TVAR Coefficients

Six artificial NSS representing various characteristics are selected to analyze the efficiency of the proposed method in estimating TVAR coefficients. The parameters used in selected methods are shown in Table 4.5. The estimated TVAR coefficients are then used to reconstruct the original signal for their accuracy. Selected input signals are further explained in sub-section 4.4.3.1.

4.4.3.1 Input Signals

Signal 5, (S5,) as in Equation (4.1) is a multi-component TV sine wave.

Signal 6 ,(S6), as in Equation (4.9) below represents a rapid sine wave frequency.

$$y(n) = 3e^{(-4n)} + \sin(2\pi 20n) \qquad (4.9)$$

Signal 7 (S7), is described in Equation (4.10):

$$y(n) = \cos(2\pi(0.01n)) + 0.5\cos(4\pi(0.01n) + 2\pi(0.53)) + 0.5\cos(6\pi(0.01n)^2) \qquad (4.10)$$

Signal 8, (S8), is a chirp signal with frequency defined over interval 0.1s to 0.4s

$$y(n) = \cos(2\pi(0.1n) + (A\pi n^2) \;\; ; A = 0.009375 \qquad (4.11)$$

Signal 9, (S9) is multicomponent rapid changing sinusoids as expressed in Equation (4.12)

$$y(n) = \sin(2\pi(f_1 An)n) + \sin(2\pi(f_2 An)n) + \sin(2\pi n).\sin(2\pi(f_3 An)n \qquad (4.12)$$

$$A = 100,\ f_1 = 200,\ f_2 = 220,\ f_3 = 100$$

Signal 10 (S10) is a frequency jump signal. Time representations of these signals ARE shown in Figure 4.8.

4.4.3.2 Results and Analysis

The computed performance metrics from the application of various methods in reconstructing the input signals from the estimated TVAR coefficients are shown in Table 4.6. The best algorithm for metrics MSE, RMSE and PRD are selected based on minimum value while CC best algorithm is selected based on its closeness to 1 and the best algorithm for SNR is selected from their maximum value. A summary of the 'best performance' algorithms are shown in Table 4.7.

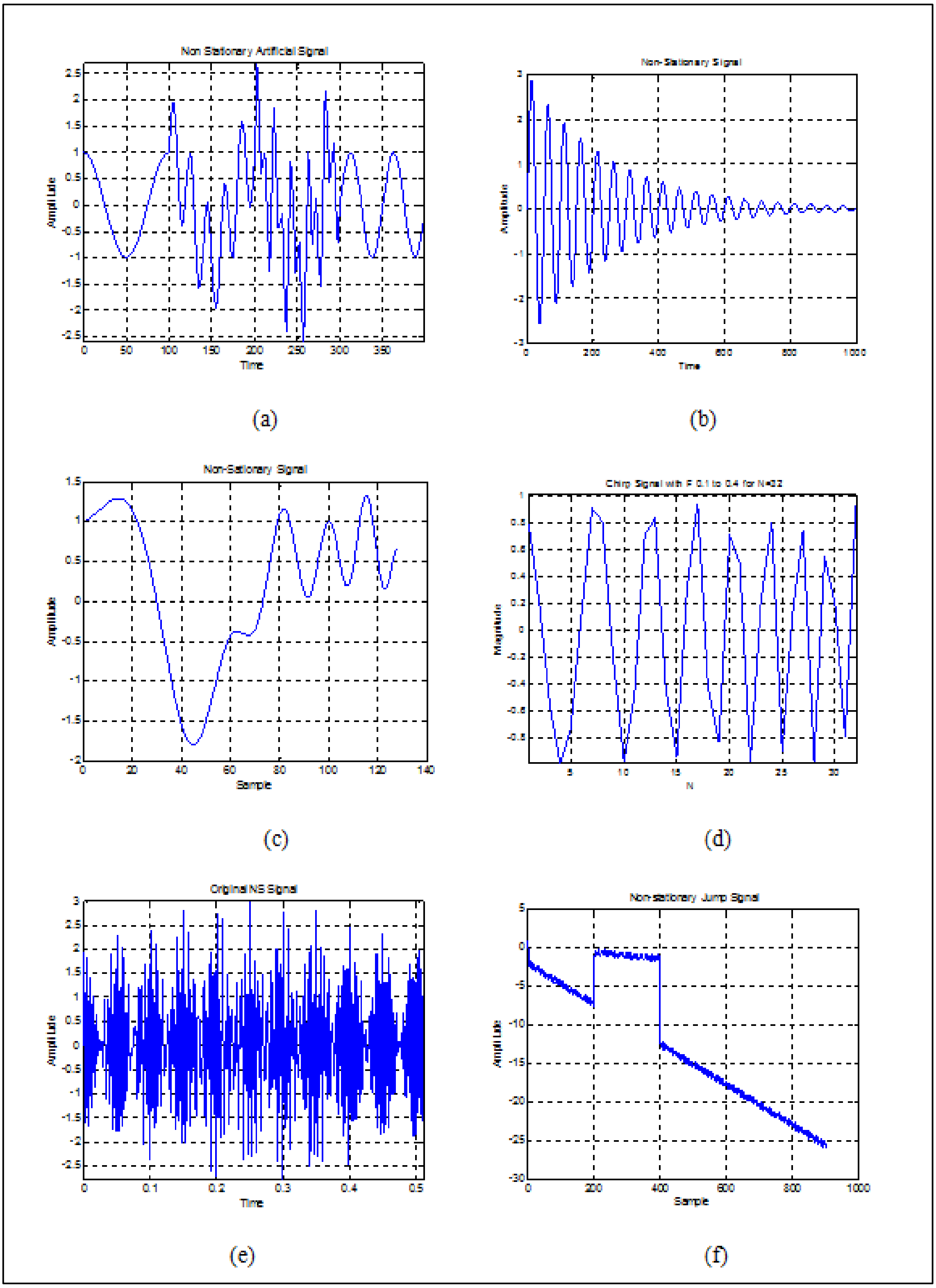


Figure 4.8 Artificial NSS (a) S5 (b) S6 (c) S7 (d) S8 (e) S9 (f) S10

Table 4.6 Performance Analysis of Various TVAR Estimation Methods

Signal	Method	MSE	CC	RMSE	PRD	SNR
S5	AR	0.369	0.921	0.607	64.963	3.74
	TVAR	0.040	0.851	0.200	57.775	4.765
	BP-ANN	0.510	-0.102	0.714	205.882	-6.272
	BP-ANN-GA	**0.009**	**0.958**	**0.099**	**28.592**	**10.870**
	BP-ANN (Cheb)	2.455	-0.228	1.566	451.2	-13.091
	BP-ANN-GA (Cheb)	0.057	0.811	0.239	69.9	3.235
S6	AR	0.227	0.935	0.476	63.61	3.928
	TVAR	0.004	0.984	0.060	12.769	12.769
	BP-ANN	0.274	-0.742	0.5232	199.354	-5.993
	BP-ANN-GA	**0.0002**	**0.997**	**0.0139**	**5.296**	**25.522**
	BP-ANN (Cheb)	2.144	-0.093	1.464	557.94	-14.93
	BP-ANN-GA (Cheb)	0.0004	0.996	0.0218	8.31	21.630
S7	AR	0.053	0.993	0.230	24.705	12.144
	TVAR	0.002	0.981	0.170	24.066	0.981
	BP-ANN	1.923	-0.676	1.386	196.099	-5.849
	BP-ANN-GA	**0.004**	**0.996**	**0.0659**	**9.307**	**20.624**
	BP-ANN (Cheb)	2.775	-0.285	1.665	235.600	-7.443
	BP-ANN-GA (Cheb)	0.005	0.9949	0.070	9.959	20.035
S8	AR	0.073	0.926	0.270	38.229	8.352
	TVAR	0.094	0.912	0.307	43.465	7.23
	BP-ANN	2.324	0.044	1.526	215.481	-6.668
	BP-ANN-GA	0.0108	0.989	0.104	14.725	16.638
	BP-ANN (Cheb)	2.155	0.002	1.468	207.273	-6.330
	BP-ANN-GA (Cheb)	**0.0001**	**1.000**	**0.0000**	**0.0004**	**148.112**
S9	AR	0.582	0.936	0.763	68.089	3.338
	TVAR	0.041	0.912	0.204	54.608	5.25
	BP-ANN	0.524	0.129	0.724	193.803	-5.747
	BP-ANN-GA	0.0044	**0.983**	0.0670	17.946	**14.920**
	BP-ANN (Cheb)	2.3190	-0.190	1.522	407.488	-12.202
	BP-ANN-GA (Cheb)	**0.0015**	0.9944	**0.0395**	**10.571**	19.5169
S10	AR	**0.346**	0.997	**0.588**	**4.00**	**27.950**
	TVAR	8.261	0.994	2.874	19.546	14.178
	BP-ANN	866.043	-0.9812	29.428	200.097	-6.024
	BP-ANN-GA	0.423	**0.997**	0.650	4.425	27.081
	BP-ANN (Cheb)	596.470	-0.775	24.422	165.974	-4.401
	BP-ANN-GA (Cheb)	433.635	-0.704	20.823	141.517	-3.0162

Table 4.7 Best Algorithm for NSS reconstruction

	Best MSE	Best CC	Best RMSE	Best PRD	Best SNR
S5	BP-ANN-GA	BP-ANN-GA	BP-ANN-GA	BP-ANN-GA	BP-ANN-GA
S6	BP-ANN-GA	BP-ANN-GA	BP-ANN-GA	BP-ANN-GA	BP-ANN-GA
S7	BP-ANN-GA	BP-ANN-GA	BP-ANN-GA	BP-ANN-GA	BP-ANN-GA
S8	BP-ANN-GA (Cheb)	BP-ANN-GA (Cheb)	BP-ANN-GA (Cheb)	BP-ANN-GA (Cheb)	BP-ANN-GA (Cheb)
S9	BP-ANN-GA (Cheb)	BP-ANN-GA	BP-ANN-GA (Cheb)	BP-ANN-GA (Cheb)	BP-ANN-GA
S10	AR	BP-ANN-GA	AR	AR	*AR*

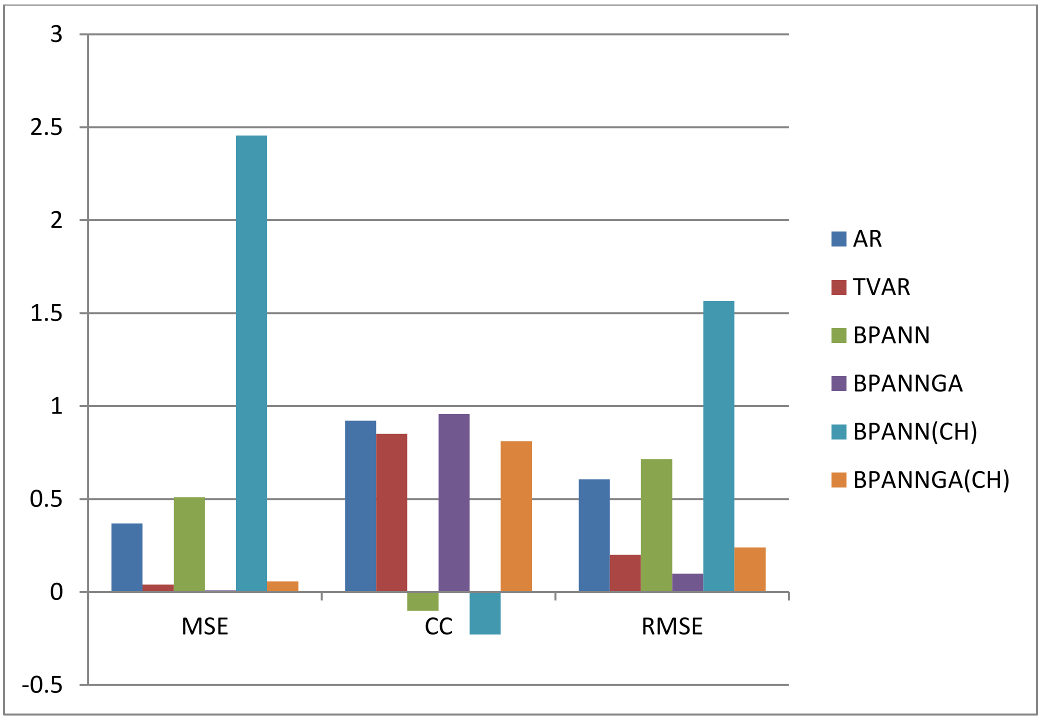


Figure 4.9 Performance Metrics of Various TVAR Estimation Methods for S5

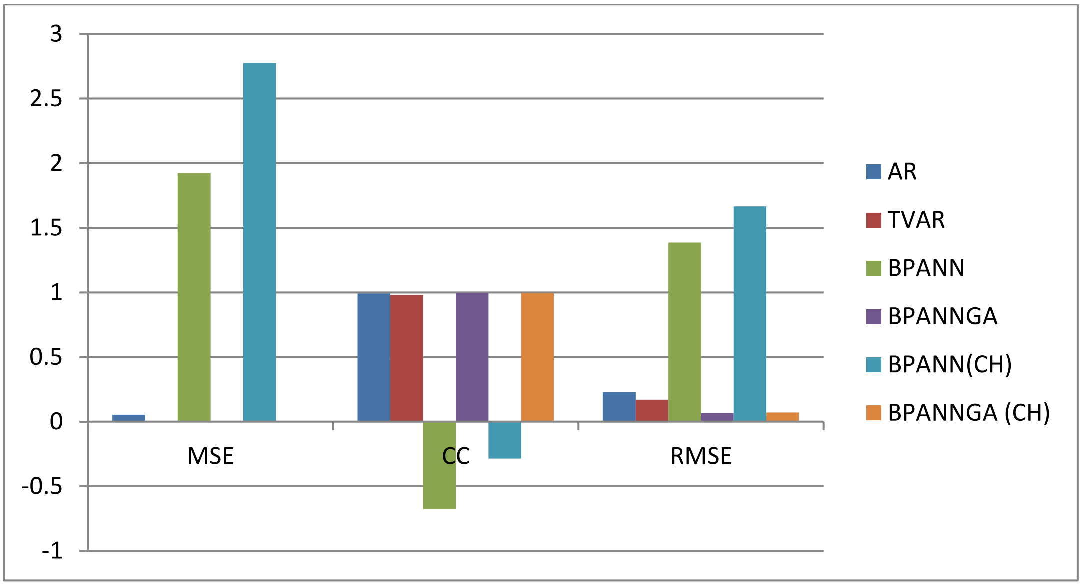


Figure 4.10 Performance Metrics of Various TVAR Estimation Methods for S7

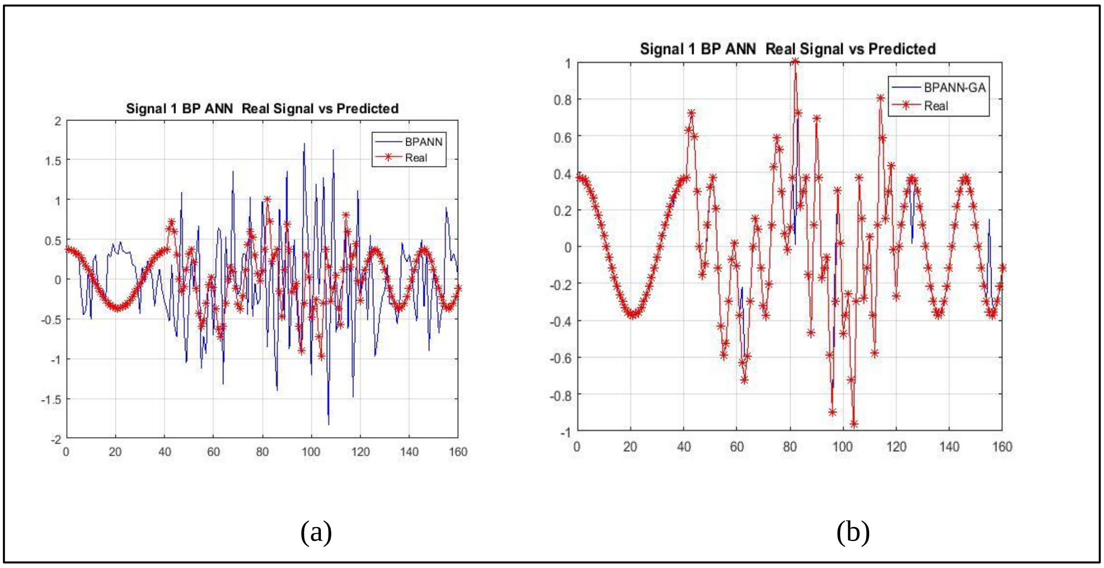
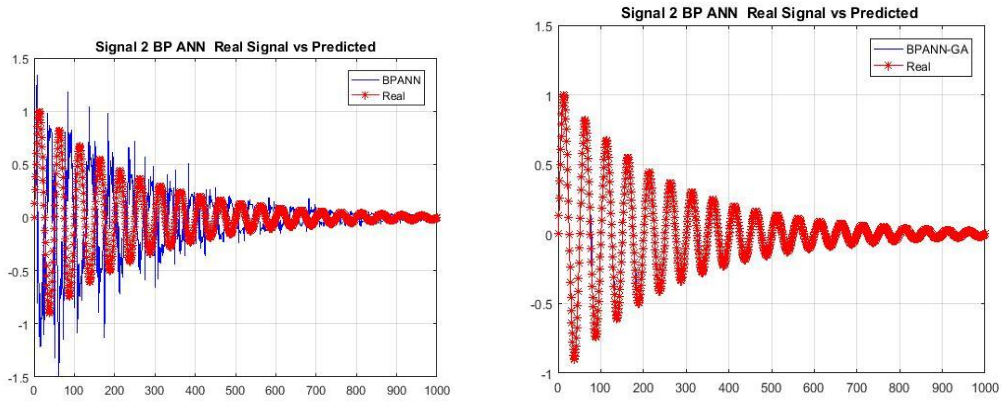
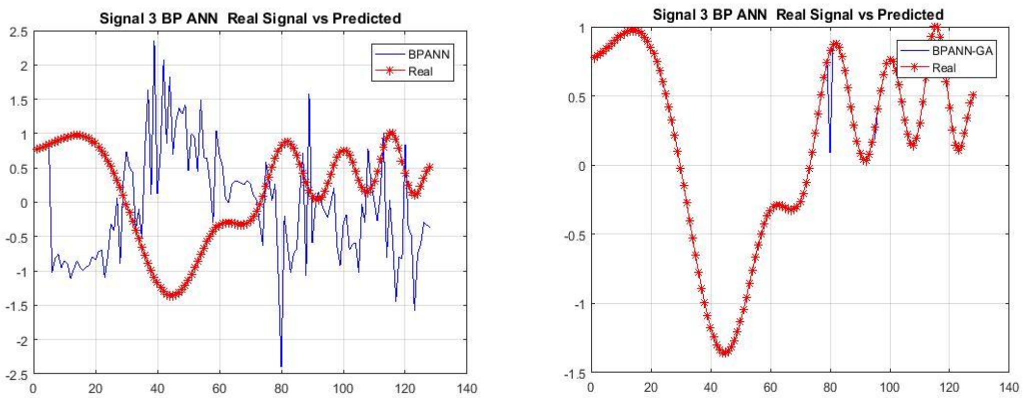


Figure 4.11 Comparison of Original and Reconstructed S5 (a)BP-ANN (b) BP-ANN-GA



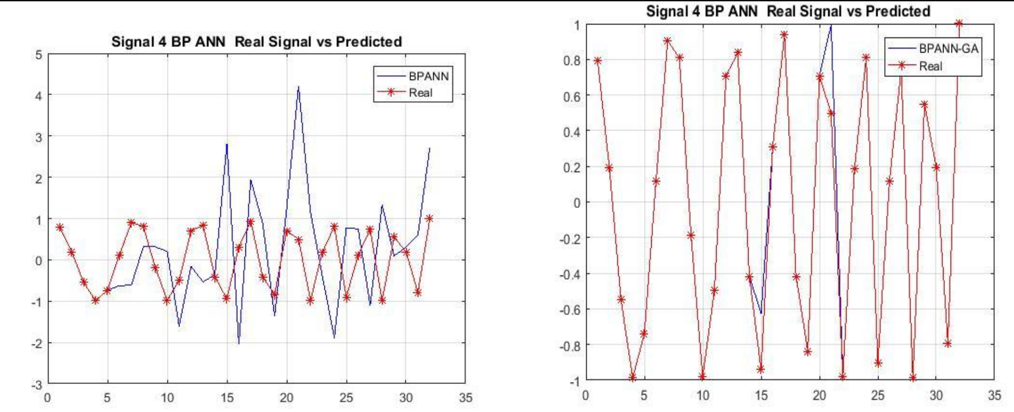
(a) (b)

Figure 4.12 Comparison of Original and Reconstructed S6 a) BP-ANN (b) BP-ANN-GA



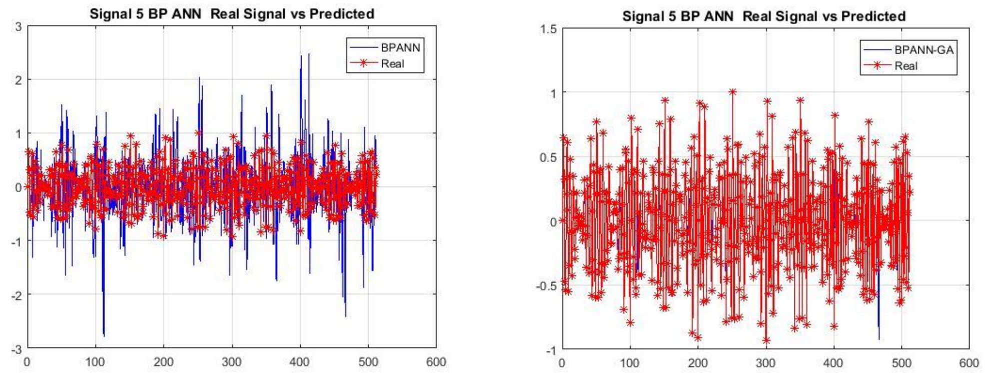
(a) (b)

Figure 4.13 Comparison of Original and Reconstructed S7
(a) BP-ANN (b) BP-ANN-GA



(a) (b)

Figure 4.14 Comparison of Original and Reconstructed S8
(a) BP-ANN (b) BP-ANN-GA



(a) (b)

Figure 4.15 Comparison of Original and Reconstructed S9
(a) BP-ANN (b) BP-ANN-GA

Bar charts comparing MSE, RMSE and CC for S5 and S7 are shown in Figure 4.9 and in Figure 4.10 respectively. Bar charts for other signals are not shown as the data is easily readable from Table 4.6 and Table 4.7. The reconstructed signals for S5, S6, S7, S8 and S9 using BP-ANN and BP-ANN-GA is shown in Figures 4.11, 4.12, 4.13, 4.14 and 4.15.

In general, it is noted that the methods with GA optimization appears to out-perform the methods without GA optimization. Table 4.7 shows that the best algorithms are the one which optimize the estimated TVAR using GA optimization, such as BP-ANN-GA and BP-ANN-GA (Chebyshev). Metrics from Table 4.6 shows these GA based methods produce minimum MSE, most close to 0, lowest PRD percentage and highest SNR value for all signals; which is also reflected from bar charts in Figures 4.9 and 4.10.

The result using BP-ANN-GA is also impressive because it is able to track all input signals which have different waveforms. The comparison between the recovered signals and the original signals for BP-ANN and BP-ANN-GA are plotted in Figures 4.11, 4.12, 4.13, 4.14 and 4.15. The solid blue line represents the reconstructed signal (also known as predicted signal) and red dotted line with star represents the original signals. Figures 4.11(a), 4.12(a), 4.13(a), 4.14(a) and 4.15(a) shows reconstructed signals using BP-ANN compared to original signals. While Figures 4.11(b), 4.12(b), 4.13(b), 4.14(b) and 4.15(b) shows reconstructed signals using BP-ANN-GA compared to the original signals. As it can be seen from all figures, BP-ANN-GA is able to track and attain smooth estimates while BP-AN is not. Although BP-ANN is able to follow the real signal, but smooth estimation and accurate waveform with relevant amplitude is not attained. The reason is BP-ANN is subjected to accuracy of input nodes, which is equal to the number of model order while BP-ANN-GA is not. Hence BP-ANN is estimating the TVAR coefficients inaccurately which does not represent the characteristics of input signals.

As summary, BP-ANN based training with GA optimization shows more accurate results compared to BP-ANN without GA optimization for all types of signals which is studied in this section.

4.5 BIOMEDICAL SIGNALS's TVAR COEFFICIENTS ESTIMATION

In this section, the TVAR coefficients are estimated from more methods including various BF as listed in Table 4.8 and compared to BP-ANN-GA. Description of these methods are detailed in Table 4.9. The estimated TVAR coefficients are then used to reconstruct the original biomedical signals, whereby their results and analysis are further analyzed.

The model orders are set as (10, 5) for (*p*,*m*) for BF methods, while estimated TVAR coefficients are optimized with limits $a_j[n] \pm 1.0$ by GA optimization process. This benchmark has been decided from simulations in Section 4.3.

Table 4.8 Description of The Input Biomedical Signals

Name	Filename	Description	Source
S10	S001R01_edfm.mat	Processes and high resolution EEG Motor Movement Signals recorded for 10s	Physionet
S11	aami3am.mat	Processed ECG data recorded for 10s with Fs=720 Hz	Physionet

Table 4.9 Methods Description

Method	TVAR Estimation methods descriptions	Parameters
AR	Autoregressive with model order 10	*p*=10
TVAR	Time Varying Autoregressive with fixed order of 10. TVAR Coefficients are computed using Correlation method	*p*=10
BP-ANN	Proposed method back propagation Artificial Neural Network without proposed GA optimization process	*p*=10, *m*=5
BP-ANN-GA	Proposed method back propagation Artificial Neural Network with proposed GA optimization process	*p*=10, *m*=5, limits ±1.0
BF Legendre	Basis function generated from Legendre polynomials	*p*=10, *m*=5
Bf-Legendre-GA	Basis function generated from Legendre polynomials with proposed GA optimization process	*p*=10, *m*=5, limits ±1.0
BF Fourier	Basis function generated from Fourier polynomials	*p*=10, *m*=5
BF Fourier -GA	Basis function generated from Fourier polynomial s with proposed GA optimization process	*p*=10, *m*=5, limits ±1.0
BF Cheb	Basis function generated from Chebyshev polynomials	*p*=10, *m*=5
BF Cheb-GA	Basis function generated from Chebyshev polynomial s with proposed GA optimization process	*p*=10, *m*=5, limits ±1.0
BF DCT	Basis function generated from Discrete Cosine Transform polynomials	*p*=10, *m*=5
BF DCT-GA	Basis function generated from Discrete Cosine Transform polynomial s with proposed GA optimization process	*p*=10, *m*=5, limits ±1.0
BF Walsh	Basis function generated from Walsh polynomials	*p*=10, *m*=5
BF Walsh-GA	Basis function generated from Walsh polynomial s with proposed GA optimization process	*p*=10, *m*=5, limits ±1.0
BF GS (L+W)	Basis function generated from Gram Schmidt Orthogonalization process using Legendre and Walsh polynomials	*p*=10, *m*=5
BF GS(L+W) -GA	Basis function generated from Gram Schmidt Orthogonalization with propose GA optimization	*p*=10, *m*=5, limits ±1.0

4.5.1 TVAR Coefficients of an EEG Motor Signal (S10)

S10, an EEG of motoric movement (Schalk et. al., 2004) are used in this section which is downloaded from Open Access Biomedical Signal, Physionet. Details of these EEG signal is available in (Liu et al., 2016). The original S10 which has 1600 samples with 10s recording is truncated to 50 samples with 0.5s length. The objective of truncating the original data to half is to reduce computation time without any alteration to original data sequence. The original S10 and truncated S10 is shown in Figure 4.16.

Estimated 11th data sequence TVAR coefficients ($a_1(11)\cdots a_{10}(11)$) from various BF with and without GA optimizations for the S10 is shown in Table 4.11.

These TVAR coefficients are then used to reconstruct first 50 samples of truncated S10 which performance analysis is presented in Table 4.10. 50 first data are reconstructed to clearly track the reconstructed signal.

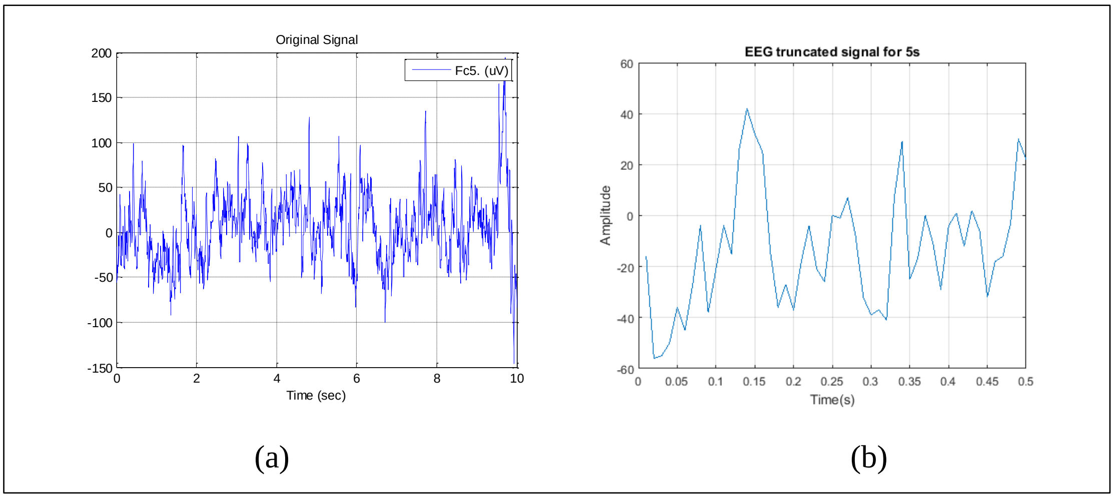


Figure 4.16 Original EEG Motoric Signal (a) Full and (b) Truncated

Table 4.10 Performance Analysis of Reconstructed S10 Using Various Methods

Signal	Method	N	Elapse time (s)	MSE	CC	RMS	PRD	SNR
S1	AR	800	0.015	1211.798	0.186	34.81	98.626	0.120
	TVAR	800	1.118	0.029	0.826	0.170	61.660	4.199
	BP-ANN	800	1963.713	0.513	-0.124	0.716	259.751	-8.291
	BP-ANN-GA	800	1963.713	0.002	0.982	0.051	18.6274	14.596
	BF Legendre	800	1782.745	0.124	-0.303	0.352	127.793	-2.130
	Bf-Legendre-GA	800	1782.745	0.000	1.000	0.000	0.000015	136.319
	BF Fourier	800	1800.522	0.133	-0.253	0.365	132.659	-2.455
	BF Fourier -GA	800	1800.522	0.000	1.000	0.000	0.000012	138.574
	BF Cheb	800	1769.795	0.145	-0.389	0.382	138.496	-2.829
	BF Cheb-GA	800	1769.795	0.000	1.000	0.000	0.000014	137.301
	BF DCT	800	1775.539	0.161	0.027	0.402	145.951	-3.284
	BF DCT-GA	800	1775.539	0.000	1.000	0.000	0.000012	138.194
	BF Walsh	800	1775.33	0.157	0.233	0.396	143.732	-3.151
	BF Walsh-GA	800	1775.33	0.000	1.000	0.000	0.000013	137.906
	BF GS (L+W)	800	1778.345	0.124	-0.302	0.352	127.993	-2.140
	BF GS(L+W) -GA	800	1778.345	0.000	1.000	0.000	0.00013	137.634

Table 4.11 Estimation of $a_j[11]$ for S10 Using Various Methods

Method	a1	a2	a3	a4	a5	a6	a7	a8	a9	a10
TVAR Corr	-0.962	0.269	-0.404	0.332	-0.127	0.237	-0.426	0.291	-0.010	-0.005
BPANN	-1.094	0.476	1.901	-0.769	0.538	-0.886	-0.455	-1.183	1.617	1.189
BPANN-GA	0.019	1.162	-0.292	-0.984	-1.815	2.125	-0.406	-2.111	-0.840	1.188
BF Fourier	-0.053	-0.396	-0.505	-0.056	0.151	0.337	-0.153	-0.067	0.059	0.015
BF Fourier-GA	-0.146	-0.214	-0.324	-0.564	0.389	1.011	0.417	0.167	0.115	-0.543
BF DCT	-1.370	1.495	-1.437	1.065	-0.843	0.671	-0.678	0.597	-0.341	0.142
BF DCT-GA	-0.540	1.029	-1.925	1.934	-0.351	0.857	-1.008	0.071	-1.114	0.362
BF Che	-1.025	0.266	-0.406	0.328	-0.136	0.228	-0.429	0.289	-0.014	-0.067
BF Che-GA	-1.426	0.349	-0.844	0.292	0.474	0.668	-0.333	-0.519	0.076	0.009
BF Legen	-0.887	0.255	-0.470	0.354	-0.144	0.264	-0.426	0.275	-0.015	0.012
BF Legen-GA	-0.751	-0.475	0.291	0.658	0.594	-0.124	-1.261	0.295	0.211	0.133
BF Walsh	0.950	0.190	0.042	-0.026	0.157	0.248	0.038	-0.093	0.175	0.092
BF Walsh-GA	0.320	-0.185	-0.210	-0.988	-0.094	0.905	0.383	-0.129	0.413	0.347
BF GS(W+L)	0.040	-0.474	-0.392	-0.033	0.170	0.239	-0.200	-0.115	0.149	0.043
BF GS(W+L) GA	-0.665	0.012	0.054	-0.473	0.940	-0.153	-0.381	-0.709	1.050	-0.909

From Table 4.10, the highest MSE is produced by AR process, while the lowest are produced by methods with GA optimization. This indicates that the AR process is not suitable to model rapid changing NSS such as biomedical signals, while methods without GA showing mixed results, suggest that the results depends on type of BF and model order (p and m) used. Interestingly, MSE lower than 0.0001, which is almost 0 is obtained when estimated TVAR is optimized with GA. In addition, GA optimized methods also shows high SNR and low PRD value, a measurement of distortion between original data and reconstructed data.

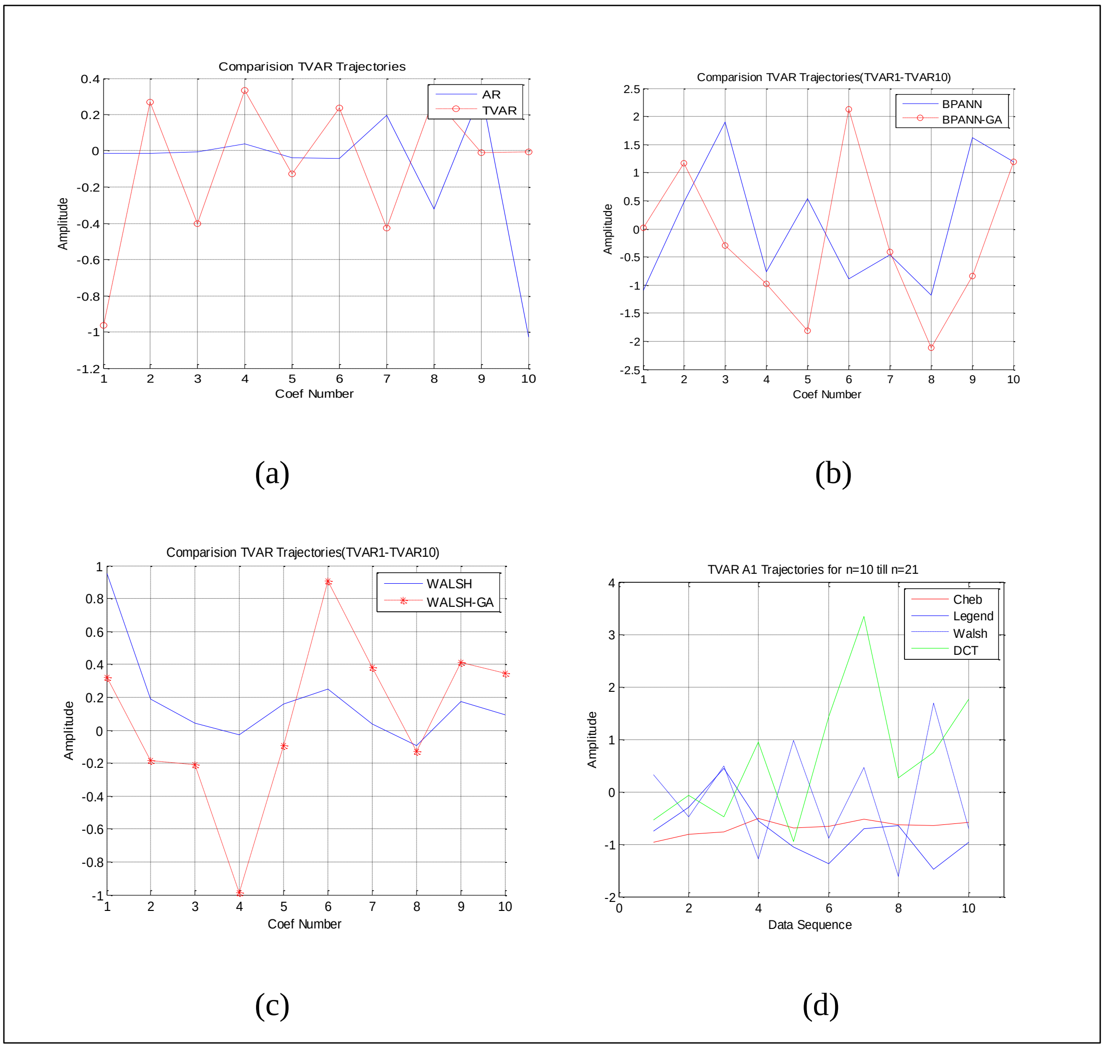


Figure 4.17 TVAR Trajectories for S10 (a) AR and TVAR (b) BP-ANN and BP-ANN-GA (c) Walsh and Walsh-GA and (d) Chebyshev, Legendre, Walsh and DCT

TVAR trajectories from Table 4.11 are shown in Figure 4.17. Figure 4.17(a) shows $a_1(11)\cdots a_{10}(11)$ of AR and TVAR Correlation, Figure 4.17(b) shows $a_1(11)\cdots a_{10}(11)$ of BP-ANN and BP-ANN-GA while Figure 4.17(c) shows $a_1(11)\cdots a_{10}(11)$ of BF Walsh and Walsh-GA. Figure 4.17(d) shows $a_1[11]\cdots a_1[21]$ of basis functions Chebyshev, Legendre, Walsh and Gram-Schmidt Orthogonalization (GSO) . No significant result observed by plotting the TVAR coefficients, except as an evident of numerical values from reconstruction simulations. Figure 4.17(d) shows all BF produces similar trajectories of $a_1[11]\cdots a_1[21]$.

First 50 data sequences of original signal and reconstructed signal is shown in Figure 4.18. Figure 4.18(a) shows reconstructed signal from AR process, which is clearly not close to original signal. This is coherent with matrixes produced by AR processing Table 4.9. Figure 4.18(b) shows signal reconstructed using TVAR coefficient estimated from TVAR Correlation methods. MSE computed to be 0.029. Although MSE is 0.029, however other matrixes shows low value and reconstructed signal is now following original signal efficiently.

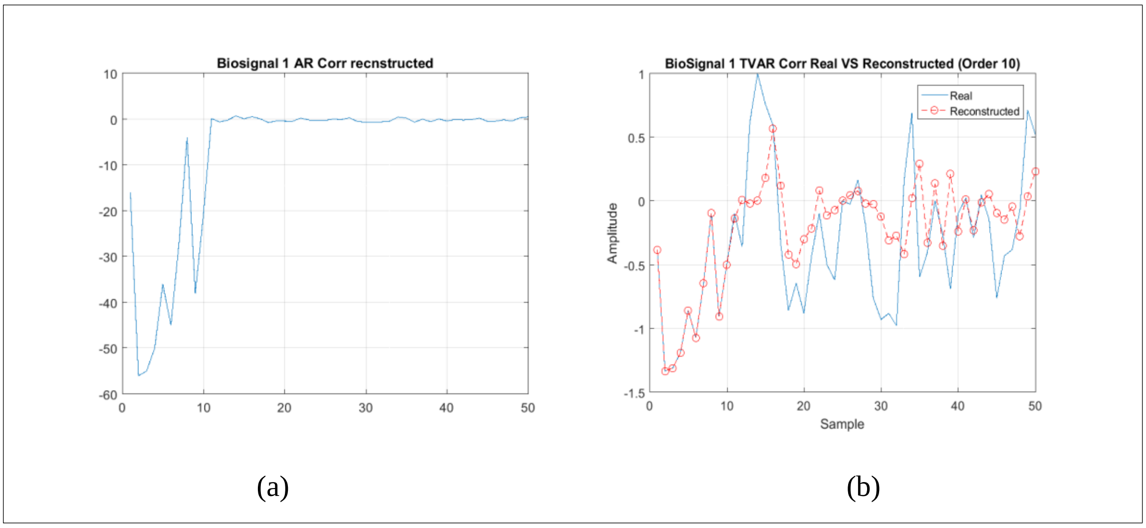


Figure 4.18 Comparison of original and reconstructed S10 from various methods (a) AR Correlation (b) TVAR Correlation

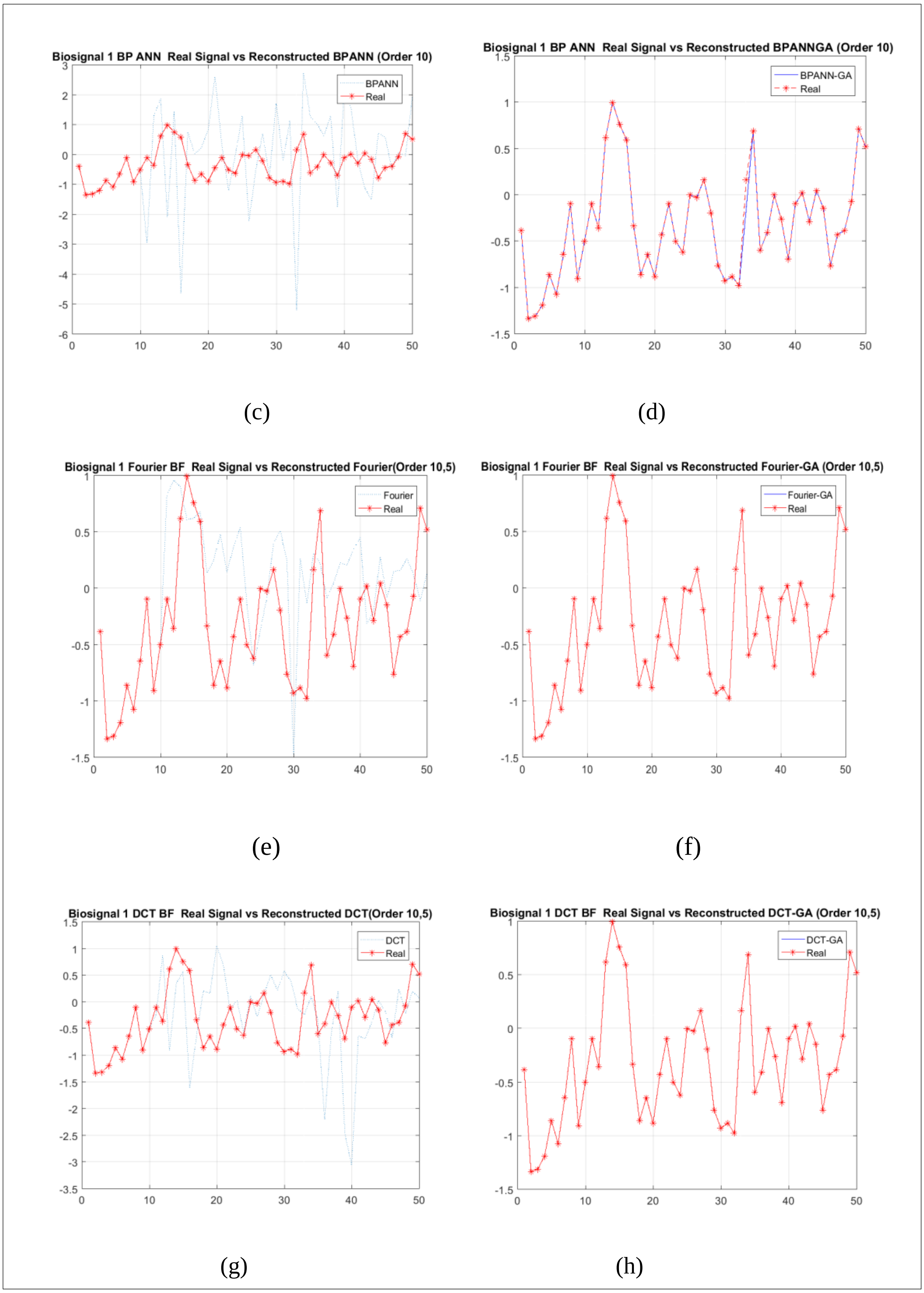


Figure 4.18 (Continued) (c) BP-ANN (d) BP-ANN-GA (e) Fourier (f) Fourier-GA (g) DCT and (h) DCT-GA

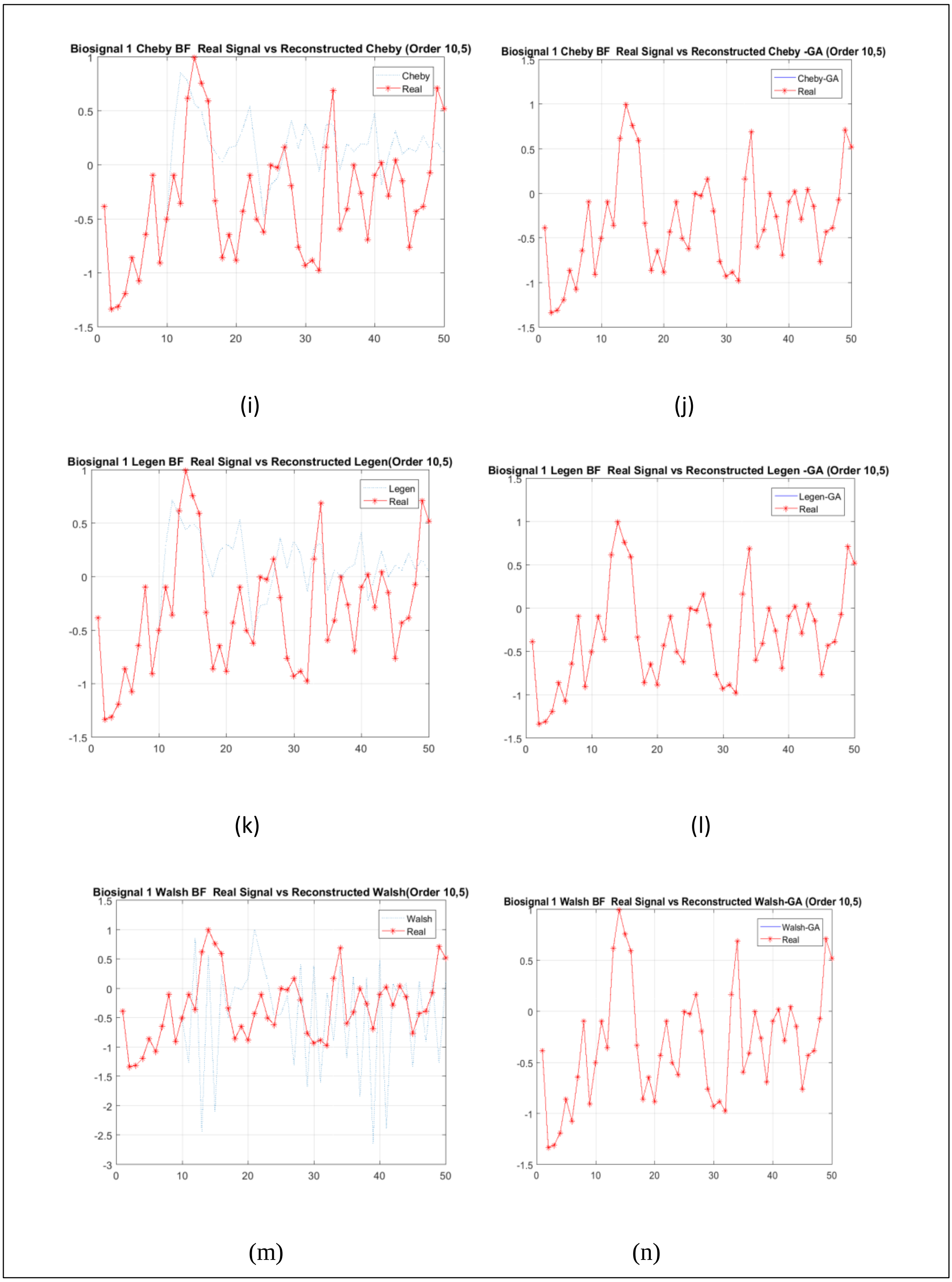
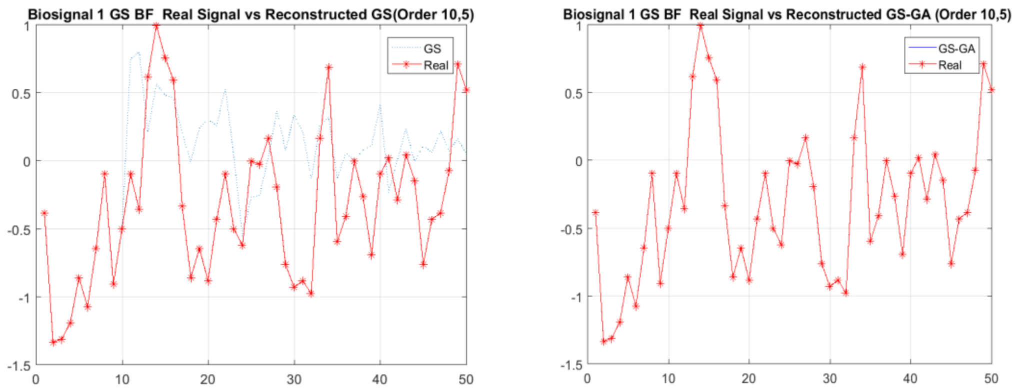


Figure 4.18 (Continued) (i) Chebyshev (j) Chebyshev-GA (k) Legendre (l) Legendre-GA (m) Walsh and (n) Walsh-GA



(o) (p)

Figure 4.18 (Continued) (o) GSO (d) GSO-GA

Figure 4.18 (e) ,(g), (i), (k), (m), (o) shows comparison of the signal reconstructed from TVAR coefficients estimated from various BF while Figure (f), (h), (j), (l), (n), (p) shows comparison signal reconstructed using the TVAR coefficients estimated from BF methods which has been optimized using GA . Generally, the TVAR coefficients without GA optimization are not able to represent the original data. Although MSE computed are significantly low, but the reconstructed signals are not accurate. On the other hand, GA optimized TVAR coefficients from all BF and BP-ANN-GA show impressive observations in which they reconstructed the original signal almost perfectly.

4.5.2 TVAR Coefficients of an ECG Signal (S11)

In this section we estimate the TVAR coefficients for S11, an ECG data sequence using various estimation methods as listed in Table 4.9. Data are downloaded from Open Access Biomedical Signal, Physionet which complete details are further explained in (Liu et al., 2016). The original S11 data was recorded for 10s with 7200 number of data and with sampling frequency of 720Hz. For analysis purpose, the original S11 is truncated to 800 samples, with an estimated length of 1.11s. Figure 4.19(a) and Figure 4.19(b) show the S11's original and its truncated signal respectively.

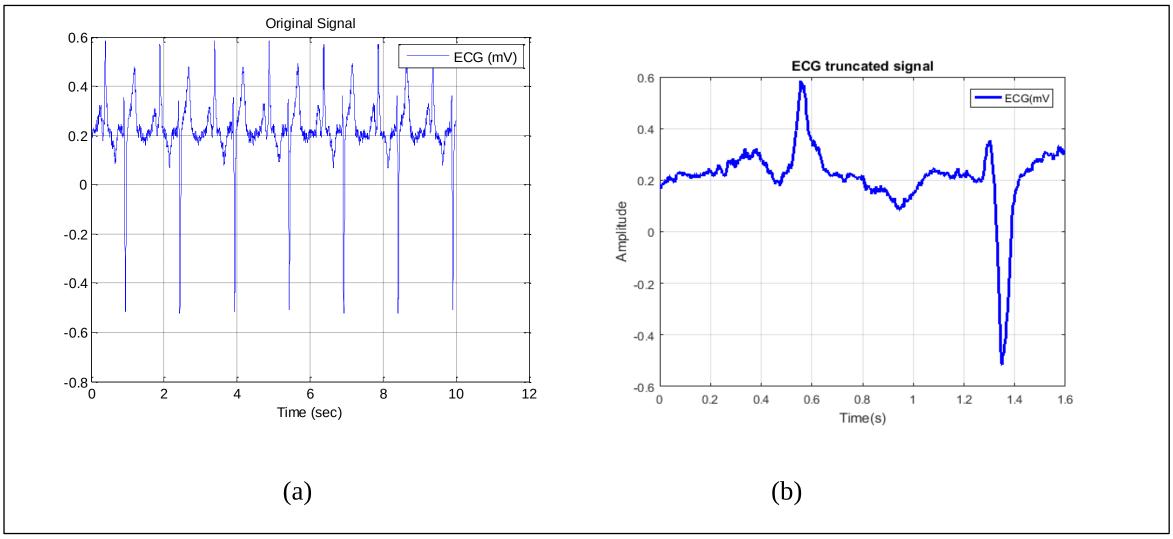


Figure 4.19 S11's (a) Original and its (b) Truncated Waveform

The truncated ECG's TVAR coefficients estimated from various methods including proposed method, BP-ANN-GA for sample, n=11 are listed in Table 4.12. These TVAR coefficients were then used to reconstruct the original truncated S11 and their performance analysis using various metrics are listed in Table 4.13.

Table 4.12 Estimation of $a_j[11]$ for S11 Using Various Methods

Methods	a1	a2	a3	a4	a5	a6	a7	a8	a9	a10
TVAR Corr	-0.8625	-0.0072	-0.0639	0.0279	0.0033	-0.0007	0.0379	-0.0728	0.0014	0.1313
BPANN	-1.5433	0.2284	1.4110	-1.0777	0.5413	-1.0412	-0.4399	-1.4497	1.4252	0.8704
BPANN-GA	-1.8371	0.8640	2.1417	-1.3530	0.6735	-1.0337	0.3356	-1.2847	0.9878	1.6318
BF Fourier	0.0310	-0.5878	-0.3888	-0.1342	0.0846	0.1549	0.1498	-0.0142	-0.0915	0.0871
BF Fourier-GA	-0.7611	-0.6675	0.3078	-0.3009	-0.8607	0.8674	1.0321	0.9578	-0.4274	0.8875
BF DCT	-1.3105	1.2431	-1.0134	0.4971	-0.3610	0.1418	-0.0108	-0.0977	0.1540	-0.0015
BF DCT-GA	-0.6938	1.1536	-1.2262	1.0418	-0.8046	0.9999	0.6213	0.4935	-0.4120	-0.1776
BF CHE	-0.9387	-0.0077	-0.0700	0.0307	0.0035	-0.0004	0.0407	-0.0789	0.0010	0.0551
BF CHE-GA	-0.2282	-0.4850	-0.2729	-0.1740	-0.5148	0.5549	0.9275	0.4911	0.3994	0.3347
BF LEGEN	-0.7904	-0.0117	-0.1492	0.0634	-0.0130	0.0081	0.0334	-0.0648	0.0057	0.1294
BF LEGEN-GA	-0.0808	0.3619	0.5902	-0.5650	0.1043	-0.0505	-0.4340	-0.3644	0.9763	0.5488
BF WALSH	1.0262	0.0797	0.0021	-0.0375	-0.0031	-0.0001	0.0403	0.0019	-0.0759	-0.0200
BF WALSH-GA	1.2948	-0.6985	0.4128	0.4030	-0.9393	0.3061	-0.5450	0.8097	-0.7098	0.7238
BF GS(W+L)	0.1160	-0.6510	-0.3007	-0.0742	0.0678	0.0643	0.0622	-0.0286	-0.0259	0.1364
BF GS(W+L)-GA	0.9256	-0.2664	0.1632	0.3058	0.7066	-0.2065	0.2595	-0.2973	-0.2425	-0.2318

Table 4.13 Performance Analysis of Reconstructed S11

Method	**N**	**Elapse time (s)**	**MSE**	**CC**	**RMS**	**PRD**	**SNR**
AR	800	0.075	0.031	0.995	0.1765	70.422	3.0458
TVAR	800	1.231	0.006	0.992	0.0826	19.266	14.304
BP-ANN	800	1989.220	0.745	-0.553	0.8635	201.462	-6.0839
BP-ANN-GA	800	1989.220	***0.000***	***1.000***	***0.0000***	***0.000015***	***136.650***
BF Fourier	800	1832.624	0.618	-0.652	0.7860	183.398	-5.2679
BF Fourier GA	800	1832.624	***0.000***	*1.000*	*0.0000*	*0.000014*	*136.783*
BF-Cheb	800	1840.266	0.677	-0.753	0.8226	191.927	-5.663
BF-Cheb GA	800	1840.266	***0.000***	*1.000*	*0.0000*	*0.000015*	*136.243*
BF-Legend	800	1845.284	0.581	-0.721	0.7618	177.738	-4.996
BF-Legend GA	800	1845.284	***0.000***	*1.000*	*0.0000*	*0.000015*	*136.382*
BF DCT	800	1810.000	0.379	-0.106	0.6160	143.732	-3.151
BF DCT GA	800	1810.000	***0.000***	***1.000***	***0.0000***	***0.000016***	***136.138***
BF Walsh	800	1860.122	0.349	0.045	0.5911	137.895	-2.790
BF Walsh GA	800	1860.122	***0.000***	**1.000**	**0.0000**	**0.000017**	**135.293**
BF-ANN (GS)	800	1831.655	0.580	-0.721	0.7618	177.738	-4.996
BF-ANN-GA (GS)	800	1831.655	***0.000***	***1.000***	***0.0000***	***0.000016***	***135.998***

It is obvious from Table 4.13 that the improved performances are obtained from algorithms which have adopted GA optimization on estimated TVAR coefficients. The methods which have adopted GA optimization shows minimum MSE, CC value closer to 1.0, minimum RMS, minimum distortion and highest SNR values. These values are made bold and italic in Table 4.13.

The reason for the superior performance of GA is due to its FF. In each population of solutions candidates, FF computes the error between the targeted value and estimated value for that population. GA then selects the population which gives minimum error as optimized TVAR coefficients.

The comparison of the original data and reconstructed data value for $y[11], y[12], \cdots y[20]$ is listed in Table 4.14. Square Sum Error (SSE) between the original value and estimated value are computed and shown as performance measurement for these values. High SSE values are observed for methods without GA optimization. This is due to inaccurate of reconstructed values as the BF methods heavily depend on model order and type of signal used. In contrary, the methods which uses GA optimizations are showing excellent performance whereby their SSE are almost equal to 0.00 and their estimated signals values are almost equal to original signal value when stated in three decimal places.

The results which are shown in Table 4.14 are coherent with results shown in Table 4.13 where both show that the method with GA optimization outperforms the BF method without optimization. Similar result was obtained for BP-ANN and BP-ANN-GA where by estimated TVAR coefficients becomes more accurate when optimized using GA.

Table 4.14 Comparison of S11 Original and Estimated Values

Method	y11	y12	y13	y14	y15	y16	y17	y18	y19	y20	SSE
Original	0.342	0.328	0.342	0.329	0.328	0.355	0.368	0.355	0.368	0.368	-
BF WALSH	0.319	-0.303	0.297	-0.303	0.318	-0.302	0.320	-0.304	0.323	-0.327	2.158
BF WALSH-GA	0.342	0.328	0.342	0.329	0.328	0.355	0.368	0.355	0.368	0.368	0.000
BF LEGEND	-0.241	-0.240	-0.217	-0.263	-0.304	-0.285	-0.275	-0.272	-0.261	-0.288	3.7792
BF LEGEND-GAA	0.342	0.328	0.342	0.329	0.328	0.355	0.368	0.355	0.368	0.368	0.000
BF GS	-0.188	-0.302	-0.167	-0.296	-0.294	-0.289	-0.276	-0.273	-0.262	-0.289	3.776
BF GS-GA	0.342	0.328	0.342	0.329	0.328	0.355	0.368	0.355	0.368	0.368	0.00
BF FOURIER	-0.211	-0.289	-0.269	-0.260	-0.287	-0.300	-0.268	-0.285	-0.276	-0.306	3.906
BF FOURIER-GA	0.342	0.328	0.342	0.328	0.328	0.355	0.368	0.355	0.368	0.368	0.000
BF DCT	-0.213	-0.284	0.003	-0.158	-0.254	0.440	0.068	-0.229	0.147	0.163	1.905
BF DCT-GA	0.342	0.328	0.342	0.328	0.328	0.355	0.368	0.355	0.368	0.368	0.000
BF CHEB	-0.301	-0.303	-0.276	-0.303	-0.304	-0.302	-0.301	-0.304	-0.300	-0.327	4.2457
BF CHEB-GA	0.342	0.328	0.342	0.328	0.328	0.355	0.368	0.355	0.368	0.368	0.000

The waveform comparison between reconstructed ECG signals with its original signal is shown for various methods Figure 4.20. The original signals are shown with solid blue line, while the reconstructed signals are shown red lines with circles.

As seen in Figure 4.20(a) and 4.20(b), the AR method and the TVAR Correlation's estimated signal has similar waveform, but different amplitude. The MSE computed from AR method is 0.031; while from the TVAR Correlation is 0.006. Although these values are significantly low, however, their SNR is low with 3.04 and 14.304 respectively for AR and TVR Correlation.

The BF based reconstructed of truncated ECG signal images are shown in Figures 4.20(c),(e),(g),(i),(k) while with GA optimization are shown in Figures 4.20(d),(f),(h),(j) and (l). The waveforms from methods without GA optimization are inaccurate with high distortion and low SNR. For instance, signal reconstructed from the TVAR coefficients estimated using BP-ANN only produce 201.462 PRD and -6.0839 SNR. While, using BP-ANN-GA produces 0.00015 PRD and 136.650 SNR values with similar waveform with the original signal. Similar observation was obtained for BF methods without GA optimization which produces distorted values, while GA optimization brings the estimated signals close to the original signal.

Observation in this section clearly shows the efficiency of the GA based methods such as BP-ANN-GA or BF-GA is more accurate in estimating TVAR coefficients more accurately than traditional methods or parametric methods without optimizations.

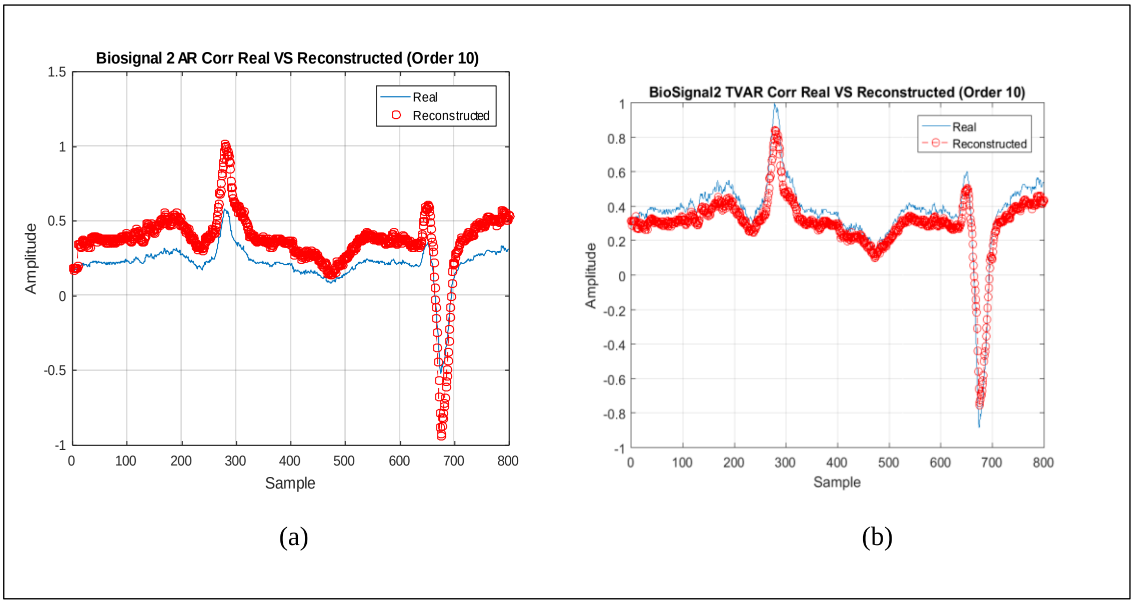


Figure 4.20 Comparison of original and reconstructed S11 (a) AR (b) TVAR Correlation

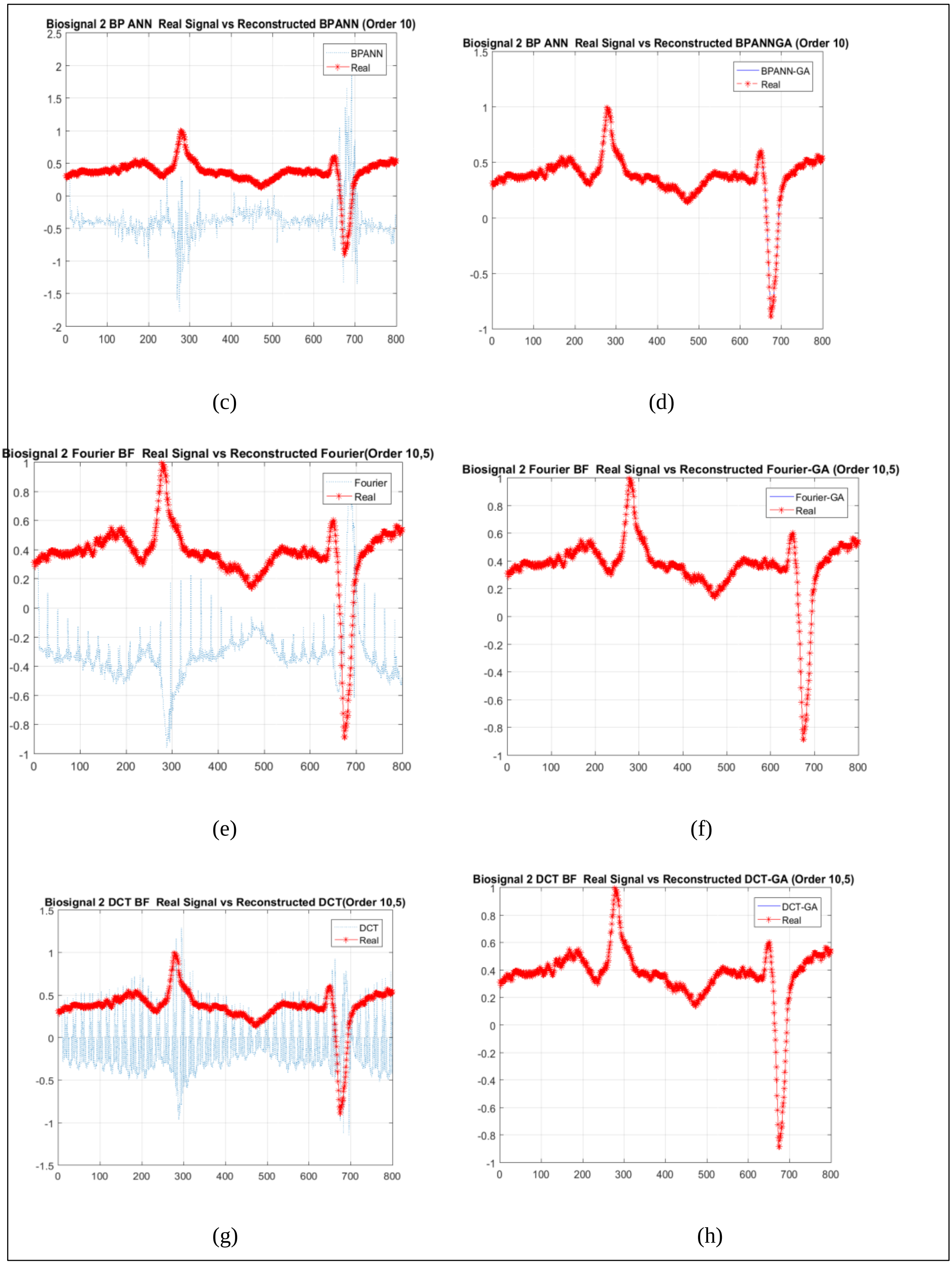


Figure 4.20 (Continued (c) BP-ANN (d) BP-ANN-GA (e) Fourier BF (f) Fourier BF-GA (g) DCT BF (h) DCT BF-GA

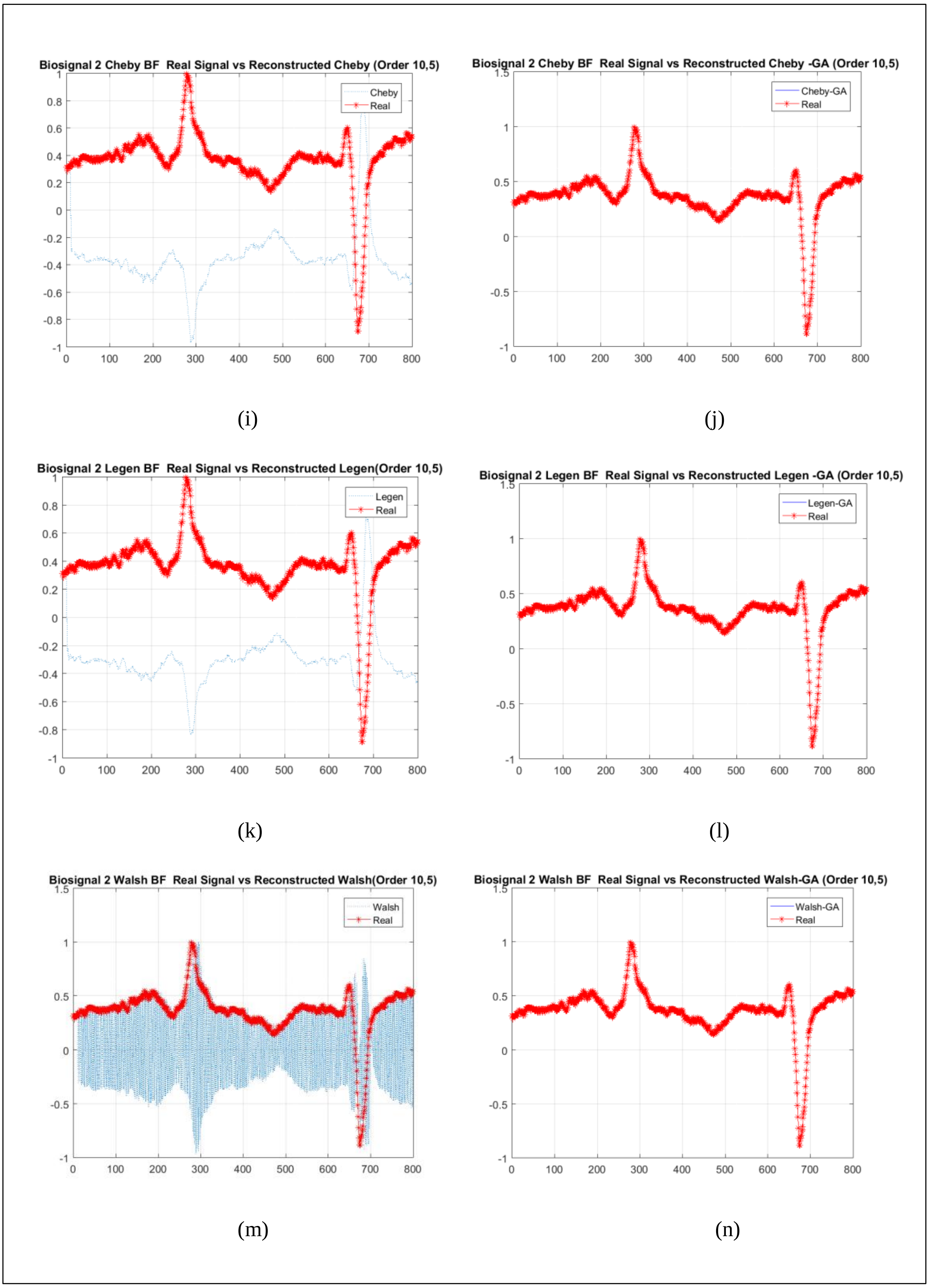
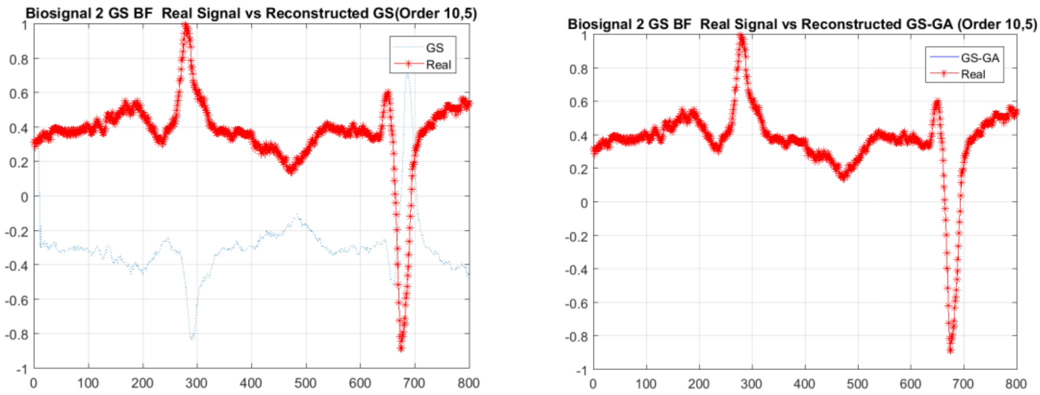


Figure 4.20 (Continued) (i) Chebyshev BF (j) Chebyshev BF –GA (k) Legendre BF (l) Legendre BF-GA (m) Walsh BF (h) Walsh BF-GA



(o) (p)

Figure 4.20 (Continued) (o) GSO BF (p) GSO – GA

4.6 SUMMARY

In this chapter, the performance analysis of the proposed method (BP-ANN-GA) to estimate the TVAR coefficients have been evaluated and reported. TVAR coefficients are estimated using BP-ANN-GA and compared to using various methods such as AR, TVAR Correlation, BP-ANN, and various BF without and with GA. This chapter starts with investigation on effects of various model orders on proposed BP-ANN and BP-ANN-GA methods using artificial ANN. Results from Section 4.2 shows BP-ANN‘s performance is influenced by model orders while BP-ANN-GA is not influenced by mode orders. BP-ANN-GA was able to optimize estimated TVAR coefficients to produce optimal TVAR coefficients for any model orders.

Effect of varying GA optimization limits was investigated in Section 4.3. This is done by allowing the estimated TVAR coefficients to be optimized within various limits starting from 0 till 2.0 with interval of 0.25. When the limits set to 0, MSE between original and reconstructed signal produced by BP-ANN-GA algorithm has no difference from BP-ANN, however it gradually decreases with limit and stable condition is attained when the limits reached ≥ 1.0. This observation is reflected in Figures 4.3 till Figures 4.6 on different stimulated data.

NS computer stimulated data and biomedical signals such as EEG motoric movement and ECG signals were reconstructed using estimated TVAR coefficients and optimized TVAR coefficients. In Section 4.4, six computed stimulation signals of various degrees, which contain slow varying frequencies and rapid varying frequencies were used to evaluate the proposed BP-ANN-GA. The ability of BP-ANN-GA method was shown to reconstruct the NSS with minimum MSE accurately was demonstrated here.

While in Section 4.5, EEG and ECG signal were reconstructed using AR, TVAR, BP-ANN, BP-ANN-GA, BF methods without and with GA optimization. It has shown that GA optimization on any method has outperformed methods without GA optimization. Generally, in all simulation, the proposed method BP-ANN-GA either using BF approach or not, has demonstrated improved performance over methods without intelligent approach. The newly proposed BP-ANN-GA method also does not suffer from effect of local-minima, independent from the effects of model orders when the estimated TVAR coefficient is optimized in limit of ± 1.0. Similar observation was observed when GA optimization is applied on various BF methods. Basically, the ability of BP-ANN-GA and GA optimized BF method has been demonstrated to accurately estimate the TVAR coefficients regardless the model order and type of BF.

CHAPTER FIVE

CONCLUSION AND RECOMMENDATIONS

5.1 CONCLUSION

The research objective of this thesis has been met in the sense an intelligent algorithm to estimate TVAR coefficients has been proposed and developed. A two level intelligent algorithm based on BP-ANN and GA has been developed whereby the objective of BP-ANN is to estimate the TVAR coefficients while the GA is formulated to optimize the estimated coefficients.

Two different approaches for BP-ANN were developed, which is based on Direct TVAR (Correlation) approach and TVAR BF approach. Both have similar structure, but differ in the number of nodes in input layer and synaptic weight representation. The proposed BP-ANN was designed based on three layer MLP architecture. In the BP-ANN based on Direct TVAR approach, the number of nodes in input layer is made equal to model order. It consists of input data nodes, which are connected to the second layer by connection weights ($\vec{w}[n]$). The second layer has an artificial neuron which has an LAF and LAF to produce an estimated output $\hat{y}[n]$. It has been shown mathematically, that from this ANN topology, the $\vec{w}[n]$ are equal to TVAR coefficients and hence, now, the objective is to estimate the appropriate $\vec{w}[n]$ for each sample, *n*. The third layer consists of single node which computes $\varepsilon[n]$ and feeds to BP learning rule which updates the $\vec{w}[n]$ recursively until $\varepsilon[n]$ is minimized.

While the input layer BP-ANN for TVAR BF consists of $p\ X\ m$ input nodes. Whereby in this case, the $\vec{w}[n]$ is shown to represent the expansion coefficients of BF, respectively. These expansion coefficients are then used to compute TVAR coefficients.

Estimated TVAR coefficients are fed into GA algorithm through two GA variables known as *lower bound (LB)* and *upper bound (UB)*. Basically, *LB* and *UB* are limits set on GA design variable in which range the TVAR coefficients should be optimized. In each iterations, the 'fitness' of optimized TVAR coefficient are evaluated using GA's FF. A FF was written to compute $\hat{y}[n]$ from optimized TVAR coefficient and an estimation error was computed by comparing with $y[n]$. TVAR coefficients with minimum estimation error will be selected by GA as optimized TVAR coefficients.

The proposed technique, BP-ANN-GA, was validated by estimating TVAR coefficients of various type of artificial NSS, truncated EEG and a truncated ECG signals. Here the results of analysis was found to be satisfactory as the proposed method produce more accurate TVAR coefficients compared to other methods as BP-ANN,AR, TVAR Correlation and various BF methods.

The evaluation of BP-ANN with various model orders clearly shows BP-ANN is heavily associated to model orders and local minima. Therefore, it produces inaccurate TVAR coefficient. While on the other hand, the BP-ANN-GA produces more accurate TVAR coefficients for all model orders, hence shows that BP-ANN-GA is an algorithm which is independent from model orders. This is proven by comparing performance of reconstructed of various NSS signals for both methods.

By setting model orders (*p*, *m)* to (10, 5) and limits of *lb* and *ub* to $a_j[n] \pm 1.0$ as benchmark, the developed algorithm are further investigated for various BF approach including GSO BF which is orthogonalization of Legendre and Walsh polynomials. The GSO BF represents multiple set of BF to analyze a NSSs. A list of BFs was tested on various NSS and results were measured using six performance metrics. It is observed that TVAR coefficients without the GA optimization is less accurate and do not represent the characteristics of original signal. In contrary, when the estimated TVAR coefficients are optimized using GA; it is able to represent and reconstruct the original signal accurately regardless the type of BF used

And therefore, the proposed algorithm, BP-ANN-GA has reduced the BF dependencies as well which makes the proposed algorithm as signal independent rather than problem dependent solution.

5.2 CONTRIBUTION TO KNOWLEDGE

The major contributions of this research are:

a) In this thesis, an intelligent method based on ANN, (BP-ANN method) was developed to estimate TVAR coefficients without involving any complex mathematical derivation. The developed BP-ANN also does not involve finding solution of complex matrices which is associated with rule based algorithm such as TVAR Correlation or TVAR BF. This conventional rule based algorithm where finding a solution is extremely iterative and involves heavy computing. BP-ANN offers lighter solution for linear, nonlinear and multicomponent NSS.

b) The GA optimization method developed in this work has been used to overcome local minima issues which was the main disadvantage of BP-ANN that limits BP-ANN implementation in many applications.

c) A new two level BP-ANN-GA has developed and shown to be independent from model order and BF. It is also a signal independent algorithm rather than problem dependent algorithm; hence it is applicable on board range of NSS.

5.3 RECOMMENDATIONS FOR FUTURE WORK

a) The developed algorithm, BP-ANN-GA have outperformed some know techniques such as TVAR Correlation, BP-ANN, TVAR BF; whereby the superiority of the developed algorithm is its signal independent characteristics, however there it still need to be improved from the speed of computation. The large time of completion is a drawback for the developed method.

b) Since the developed method was able to estimate TVAR coefficients accurately, its application for classification and automated diagnostic system should be further investigated.

c) The modification of the developed algorithm to extend its application on Forward and Backward Autoregressive Model (FBAR) and Time Varying Autoregressive Moving Average (TVARMA) are recommended for future work.

REFERENCES

Abo-Hammour, Z., Alsmadi, O., Momani, S., & Abu Arqub, O. (2013). A genetic algorithm approach for prediction of linear dynamical systems. *Mathematical Problems in Engineering, 2013*. http://doi.org/10.1155/2013/831657

Abo-Hammour, Z. S., Alsmadi, O. M. K., Al-Smadi, A. M., Zaqout, M. I., & Saraireh, M. S. (2012). ARMA model order and parameter estimation using genetic algorithms. *Mathematical and Computer Modelling of Dynamical Systems*, *18*(2), 201–221. http://doi.org/10.1080/13873954.2011.614068

Aboy, M., Márquez, O. W., Mcnames, J., & Hornero, R. (2005). Adaptive Modeling and Spectral Estimation of Nonstationary Biomedical Signals Based on Kalman Filtering. *IEEE Transactions on Bio-Medical Engineering*, *52*(8), 1485–1489. http://doi.org/10.1109/TBME.2005.851465

Abramovich, Y. I., Spencer, N. K., & Turley, M. D. E. (2007). Order Estimation and Discrimination Between Stationary and Time-Varying (TVAR) Autoregressive Models. *IEEE Transactions on Signal Processing*, *55*(6), 2861–2876. Retrieved from http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=4203034

Agarwal, M. (2015). Text recognition from image using Artificial Neural Network and Genetic Algorithm. In *International Conference on Green Computing and Internet of Things (ICGioT 2015)* (pp. 1610–1617). Noida, IND: IEEE.

Ahmad, M., & Aqil, M. (2016). QR Decomposition Based Recursive Least Square Adaptation of Autoregressive EEG Features. In *international Conference on Intelligent Systems Engineering (ICISE) 2016*. IEEE. http://doi.org/10.1109/INTELSE.2016.7475176

Aibinu, a. M., Salami, M. J. E., & Shafie, a. a. (2012). Artificial neural network based autoregressive modeling technique with application in voice activity detection. *Engineering Applications of Artificial Intelligence*, *25*(6), 1265–1276. http://doi.org/10.1016/j.engappai.2012.05.012

Aibinu, A. M. (2010). *Development of Artifiical Neural Network based Parameter Models for Biomedical Signals and Image Analysis*. IIUM, Kuala Lumpur.

Aibinu, A. M., Salami, M. J. E., & Shafie, A. a. (2012). Artificial neural network based autoregressive modeling technique with application in voice activity detection. *Engineering Applications of Artificial Intelligence*, *25*(6), 1265–1276. http://doi.org/10.1016/j.engappai.2012.05.012

Aibinu, A. M., Salami, M. J. E., Shafie, A. A., & Najeeb, A. R. (2008). Increasing The Speed of Convergence of an Artificial Neural Network based ARMA

Coefficients Determination Technique. *World Academy of Science, Engineering and Technology*, (42), 166–172.

Akay, Y. M., Akay, M., Welkowitz, W., Semmlow, J. L., & Kostis, J. B. (1993). Noninvasive acoustical detection of coronary artery disease: a comparative study of signal processing methods. *IEEE Transactions on Bio-Medical Engineering*, *40*(6), 571–8. http://doi.org/10.1109/10.237677

Al, A., Ferdjallah, M., Ph, D., & D, A. K. P. (2006). Muscle Fatigue Analysis For Healthy Adults Using TVAR Model With Instantaneous Frequency Estimation. In *Proceedings of the Thirty Eighth Sounteeastern Symposium on System Theory 2006* (pp. 44–47). IEEE. http://doi.org/10.1109/SSST.2006.1619081

Alippi, C., & A. (1991). Weight Update in Back-Propagation Neural Network The Role of Activation Functions. In *International Joint Conference on Neural Network* (pp. 560–565). Singapore: IEEE. http://doi.org/10.1109/IJCNN.1991.170459

Arnold, M., Miltner, W. H., Witte, H., Bauer, R., & Braun, C. (1998). Adaptive AR modeling of nonstationary time series by means of Kalman filtering. *IEEE Transactions on Bio-Medical Engineering*, *45*(5), 553–62. http://doi.org/10.1109/10.668741

Baragona, R., Battaglia, F., & Cucina, D. (2004). THRESHOLD AUTOREGRESSIVE MOVING AVERAGE MODELS : SOME SIMULATION RESULTS ∗ . In *Proceedings Mathematical and Statistical Methods for the Analysis of Insurance and Financial Data*. Elselvier. http://doi.org/9788890135507

Bassam, R., Abbas, A. K., & Kasim, R. M. (2008). Spectral Characteristics Estimation of PCG Signals based on ARMAX Nonlinear Modeling. In *4th Kuala Lumpur International Conference on Biomedicall ENgineering* (pp. 242–246). Kuala Lumpur: IFMBE Proceedings. http://doi.org/https://doi.org/10.1007/978-3-540-69139-6_63

Baumgartner, C., Blinowska, K. J., Cichocki, A., Dickhaus, H., & Durka, P. J. (2013). Discussion of " Time-frequency Techniques in Biomedical Signal Analysis : A Tutorial Review of Similarities and Differences " Dr . M . H . Moradi ; Biomedical Engineering Faculty ; Dr . M . H . Moradi ; Biomedical Engineering Faculty ; *Methods in Information Medicine*, (4), 297–307.

Boukhennoufa, N., Benmahammed, K., & Benzid, R. (2012). Effective PCG Signals Compression Technique Using an Enhanced, *48*, 89–102.

Castro, A., Moukadem, A., Schmidt, S., Dieterlen, A., & Coimbra, M. (2015). Analysis of the electromechanical activity of the heart from synchronized ECG and PCG signals of subjects under stress. In *BIOSIGNALS 2015 - 8th International Conference on Bio-Inspired Systems and Signal Processing, Proceedings; Part of 8th International Joint Conference on Biomedical*

Engineering Systems and Technologies, BIOSTEC 2015 (pp. 49–56). Lisbon, Portugal: SCITEPRESS Digital Librar. http://doi.org/10.5220/0005202400490056

Cerutti, B. S., Baselli, G., Bianchi, A. M., Caiani, E., Contini, D., Cubeddu, R., … Signorini, M. G. (2011). Biomedical Signal and Image Processing :A Need for Close Integration. *Ieee Pulse, 2*(June 2011), 41–54. http://doi.org/10.1109/MPUL.2011.941522

Cho, Y. S., Kim, S. B., & Powers, E. J. (1991). Time-varying spectral estimation using AR models with variable forgetting factors. *IEEE Transactions on Signal Processing*, *39*(6), 1422–1426. http://doi.org/10.1109/78.136549

Chon, K. H., Hoyer, D., Armoundous, A., & Marsh, D. (1999). Robust Nonlinear Autoregressive Moving Average Model Parameters Estimation Using Stochastic Recurrent Artificial Neural Network. *Annals of Biomedical Engineering, 4,* 538–547. Retrieved from https://www.ncbi.nlm.nih.gov/pubmed/10468238

Chon, K. H., Zhao, H., Zou, R., & Ju, K. (2005). Multiple time-varying dynamic analysis using multiple sets of basis functions. *IEEE Transactions on Bio-Medical Engineering*, *52*(5), 956–60. http://doi.org/10.1109/TBME.2005.845362

Chu, Y. (2012). *Recursive local estimation : algorithm, performance and applications*. Pokfulam, Hong Kong. Retrieved from http://hub.hku.hk/handle/10722/181861

Chu, Y. J., Chan, S. C., Zhang, Z. G., & Tsui, K. M. (2012a). A new recursive algorithm for time-varying autoregressive (TVAR) model estimation and its application to speech analysis. In *2012 IEEE International Symposium on Circuits and Systems* (pp. 1026–1029). Seoul, South Korea: IEEE. http://doi.org/10.1109/ISCAS.2012.6271402

Chu, Y. J., Chan, S. C., Zhang, Z. G., & Tsui, K. M. (2012b). A new regularized TVAR-based algorithm for recursive detection of nonstationarity and its application to speech signals. In *2012 IEEE Statistical Signal Processing Workshop, SSP 2012* (pp. 361–364). IEEE. http://doi.org/10.1109/SSP.2012.6319704

Corte, D. T. La, & Zou, Y. M. (2014). Newton's method backpropagation for complex-valued holomorphic multilayer perceptrons. In *Proceedings of the International Joint Conference on Neural Networks* (pp. 2854–2861). http://doi.org/10.1109/IJCNN.2014.6889384

Costa, A. H., & Hengstler, S. (2011). Adaptive time–frequency analysis based on autoregressive modeling. *Signal Processing*, *91*(4), 740–749. http://doi.org/10.1016/j.sigpro.2010.07.020

David, L. (2009). *Parametric Time – Frequency Analysis for Discrimination of Non – Stationary Signals*. National University of Coluimbia.

Dhanwani, D., & Wadhe, P. A. (2013). Study Of Hybrid Genetic Algorithm Using Artificial Neural Network In Data Mining For The Diagnosis Of Stroke Disease. *International Journal of Computational Engineering Research*, *3*(May 2011), 95–100. http://doi.org/10.1.1.300.9868

Dhiman, J., Ahmad, S., & Gulia, K. (2013). Comparison between Adaptive filter Algorithms (LMS, NLMS and RLS). In *International Journal of Science, Engineering and Technology Research* (Vol. 2, pp. 2278–7798). India: IJSETR.

Dorantes-Mendez, G., Charleston-Villalobos, S., Gonzalez-Camarena, R., Chi-Lem, G., & Aljama-Corrales, T. (2008). Imaging of simulated crackle sounds distribution on the chest. In *Conference proceedings : ... Annual International Conference of the IEEE Engineering in Medicine and Biology Society. IEEE Engineering in Medicine and Biology Society. Conference* (Vol. 2008, pp. 4801–4804). British Columbia, Canada: IEEE. http://doi.org/10.1109/IEMBS.2008.4650287

Edmonson, W. W. (1998). A Simpli ed Global Least Mean Square Algorithm for Adaptive IIR Filtering, *45*(3), 379–384.

Eom, K. B. (1999). Analysis of acoustic signatures from moving vehicles using time-varying autoregressive models. *Multidimensional Systems and Signal Processing*, *10*(4), 357–378. http://doi.org/10.1023/A:1008475713345

Ervural, B. C., Beyca, O. F., & Zaim, S. (2016). Model Estimation of ARMA Using Genetic Algorithms: A Case Study of Forecasting Natural Gas Consumption. *Procedia - Social and Behavioral Sciences*, *235*(October), 537–545. http://doi.org/10.1016/j.sbspro.2016.11.066

F.Hlawatsch, & Batels, G. F. B. (1992). Linear and quadtratic time frequency signal representations. *Signal Processing Magazine,IEEE*, *9*(2), 21–67. http://doi.org/10.1109/79.127284

Fernández-Redondo, M., & Hernández-Espinosa, C. (2001). Weight initialization methods for multilayer feedforward. In *Proceedings of the European Symposium on Artificial Neural Networks* (pp. 119–124). Bruges, Belgium: D-Facto Publication. http://doi.org/10.1109/IJCNN.2001.939011

Filho, F., Maia, H. Z., Mateus, T. H. a, Ozpineci, B., Tolbert, L. M., & Pinto, J. O. P. (2013). Adaptive selective harmonic minimization based on ANNs for cascade multilevel inverters with varying DC sources. *IEEE Transactions on Industrial Electronics*, *60*(5), 1955–1962. http://doi.org/10.1109/TIE.2012.2224072

Flores, J. J., Graff, M., & Rodriguez, H. (2012). Evolutive design of ARMA and ANN models for time series forecasting. *Renewable Energy*, *44*, 225–230. http://doi.org/10.1016/j.renene.2012.01.084

Fu, J., Genson, J., Jan, Y. K., & Jones, M. (2011). Using artificial neural network to determine favorable wheelchair tilt and recline usage in people with spinal cord injury: Training ANN with genetic algorithm to improve generalization. In *Proceedings - International Conference on Tools with Artificial Intelligence, ICTAI* (pp. 25–32). Boca Ratopn, FL: IEEE. http://doi.org/10.1109/ICTAI.2011.13

G.P.Nason. (2006). Stationary and Non Stationary Time Series. In L. J. C. H.M. Mader, S.G. Coles, C.B. Connor (Ed.), *Statistics in Volcanology* (1st ed., p. 29). London: The Geological Society.

Grenier, Y. (1983). Time-Dependent ARMA Modeling of Non stationary Signal. *IEEE Transactions on Acoustics,Speech and Signal Processing*, *ASSP-31*(4), 12. http://doi.org/10.1109/TASSP.1983.1164152

Hall, M. G., V.Oppenheim, A., & Willsky, A. S. (1983). TIME-VARYING PARAMETRIC MODELING OF SPEECH. *Signal Processing*, *5*(049), 267–285.

Harmat, A., Juntunen, M., & Kaipio, J. P. (2000). Time-varying Autoregressive Modeling of Audio andSpeech Signals. In *10th European Signal Processing Conference* (p. 4). Tampere, Finland: IEEE.

Hendrantoro, G., Mauludiyanto, A., & Handayani, P. (n.d.). An autoregressive model for simulation of time-varying rain rate, (4), 0–3.

Hsiao, T. (2008). Identification of Time-Varying Autoregressive Systems Using Maximum a Posteriori Estimation. *IEEE Transactions on Signal Processing*, *56*(8), 3497–3509. http://doi.org/10.1109/TSP.2008.919393

Hu, C. H. C., & Zhao, F. Z. F. (2010). Improved Methods of BP Neural Network Algorithm and its Limitation. In *Information Technology and Applications (IFITA), 2010 International Forum on* (Vol. 1, pp. 11–14). Kunming: IEEE. http://doi.org/10.1109/IFITA.2010.324

Hu, L., Qin, L., Mao, K., Chen, W., & Fu, X. (2016). Optimization of Neural Network by Genetic Algorithm for Flowrate Determination in Multipath Ultrasonic Gas Flowmeter. *IEEE Sensors Journal*, *16*(5), 1158–1167. http://doi.org/10.1109/JSEN.2015.2501427

Hu, Y. H., & Hwang, J. (2002a). Signal Processing Using the Multilayer Perceptron. In Y. Hen Hu & J. N. Hwang (Eds.), *Handbook of Neural Netowrk Signal Processing*. London: CRC PRESS.

Hu, Y. H., & Hwang, J. N. (Eds.). (2002b). *Handbook of Neural Network Signal Processing*. Florida: CRC PRESS.

Islam, B. U., Baharudin, Z., Raza, M. Q., & Nallagownden, P. (2014). Optimization of neural network architecture using genetic algorithm for load forecasting. *2014 5th International Conference on Intelligent and Advanced Systems (ICIAS)*, 1–6. http://doi.org/10.1109/ICIAS.2014.6869528

Jain, A. K., & Mao, J. (1996). Artificial Neural Network: A Tutorial. *IEEE Computer Society*, *29*, 31–44. http://doi.org/10.1109/2.485891

Jian-jun, W., Dong-Xiao, N., & Li, L. (2009). An ARMA Cooperate with Artificial Neural Network Approach in Short-Term Load Forecasting. *2009 Fifth International Conference on Natural Computation*, 60–64. http://doi.org/10.1109/ICNC.2009.253

Kartheeswaran, S., Dharmaraj, D., & Durairaj, C. (2015). A Hybrid Genetic Algorithm and Back-Propagation Artificial Neural Network Based Simulation System for Medical Image Reconstruction in Noise-Added Magnetic Resonance Imaging Data. In *International Conference on Green Engineering and Technologies, IC_GET 2015* (pp. 1–6). Coimbatore, India: IEEE. http://doi.org/10.1109/GET.2015.7453863

Kawafuku, M., & Sasaki, M. (1999). Adaptive learning method of neural network controller using an immune feedback law. In *6th International Conference on Neural Information Processing ICONIP'99* (pp. 502–507). Perth, WA: IEEE. http://doi.org/doi: 10.1109/ICONIP.1999.845645

Khan, A. U., T.K., B., & Sudhir Sharma. (2008). Genetic Algorithm Based Backpropagation Neural Network Performs better than Backpropagation Neural Network in Stock Rates Prediction. In *IJCSNS International Journal of Computer Science and Network Security* (Vol. 8, pp. 162–166). Seoul, Korea: IJCSNS. Retrieved from http://paper.ijcsns.org/07_book/200807/20080724.pdf

Khorani, V. (2011). Artificial Neural Network Weights Optimization Using ICA , GA , ICA-GA and R-ICA-GA : Comparing Performances.

Koščak, J., Jakša, R., & Sinčák, P. (2010). Stochastic weight update in the backpropagation algorithm on feed-forward neural networks. In *Proceedings of the International Joint Conference on Neural Networks* (pp. 1–4). Barcelona: IEEE. http://doi.org/10.1109/IJCNN.2010.5596870

Kumar, D., Carvalho, P., Antunes, M., Henriques, J., Eugenio, L., Schmidt, R., & Habetha, J. (2006). Detection of S1 and S2 Heart Sounds by High Frequency Signatures. In *28th Annual International Conference of the IEEE Engineering in Medicine and Biology Society.* (Vol. 1, pp. 1410–1416). New York, USA: IEEE. http://doi.org/10.1109/IEMBS.2006.260735

Leung, S. H., & So, C. F. (2005). Gradient-based variable forgetting factor RLS algorithm in time-varying environments. *IEEE Transactions on Signal Processing*, *53*(8), 3141–3150. http://doi.org/10.1109/TSP.2005.851110

Li, D., & Jayaweera, S. K. (2013). Uncertainty modeling and prediction for customer load demand in smart grid. In *2013 IEEE Energytech, Energytech 2013*. Cleveland, Ohio,USA: IEEE. http://doi.org/10.1109/EnergyTech.2013.6645360

Li, Y., Liu, Q., Tan, S., & Chan, R. (2016). High-resolution time-frequency analysis of EEG signals using multiscale radial basis functions. *Neurocomputing*, *195*(C), 96–103. http://doi.org/10.1016/j.neucom.2015.04.128

Li, Y., Wei, H., & Billings, S. a. (2011). Identification of Time-Varying Systems Using Multi-Wavelet Basis Functions. *IEEE Transactions on Control Systems Technology*, *19*(3), 656–663. http://doi.org/10.1109/TCST.2010.2052257

Li, Y., Wei, H. L., & Billings, S. a. (2009). Time-varying signal processing using multi-wavelet basis functions and a modified block least mean square algorithm. Retrieved from http://eprints.whiterose.ac.uk/74650/

Li, Y., Wei, H. L., Billings, S. a., & Sarrigiannis, P. G. (2011). Time-varying model identification for time-frequency feature extraction from EEG data. *Journal of Neuroscience Methods*, *196*(1), 151–158. http://doi.org/10.1016/j.jneumeth.2010.11.027

Li, Y., Wei, H. L., Billings, S. A., & Wei, H. (2009). Time-Varying Signal Processing Using Multi-wavelet Basis Functions and A Modified Block Least Mean Square Algorithm Time-Varying Signal Processing Using Multi-Wavelet Basis Functions and A Modified Block Least Mean Square Algorithm, (998), 1–17.

Liu, C., Li, Q., Moody, B., Juan, R. A., Chorro, F. J., Castells, F., … Samieinasab, M. R. (2016). An open access database for the evaluation of heart sound algorithms. *Physiological Measurement*, *37*(12), 2181–2213. http://doi.org/10.1088/0967-3334/37/12/2181

Lu, B., Feng, C., & Long, G. (2013). A New Variable Step-Size LMS Adaptive Algorithm Based on Marr Function. *2013 International Conference on Information Technology and Applications*, 214–217. http://doi.org/10.1109/ITA.2013.56

Madić, M., & Radovanović, M. (2011). Optimal selection of ANN training and architectural parameters using Taguchi method: A case study. *FME Transactions*, *39*, 79–86. Retrieved from http://scindeks.ceon.rs/article.aspx?artid=1451-20921102079M

Man, K. F., Tang, K. S., & Kwong, S. (1996). Genetic algorithms: Concepts and applications. *IEEE Transactions on Industrial Electronics*, *43*(5), 519–534. http://doi.org/10.1109/41.538609

Mate, S. N., & Patil, A. B. (2013). Identification of Time Varying System Using Recursive Estimation Approach and Wavelet Based Recursive Estimation Approach. *International Journal of Engineering Science and Innovative Technology (IJESIT)*, *2*(3), 526–532.

Matsuzaki, H. (2013). A Simple Restart Scheme for Improving Convergence Rates of RLS Filters in Tracking Time-varying Delay Systems. In *SICE Annual*

Conference 2013. Nagoya, Japan: IEEE. Retrieved from http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=6736147

Maust, R. S., & Cq, H. (1998). Reduced Order Modeling Using a Genetic Algorithm. In *Proceedings of the 13th Southeastern Symposium on System Theory* (pp. 67–71). Morgantown, WV: IEEE. http://doi.org/10.1109/SSST.1998.660021

Minerva, T., & Poli, I. (2001). Building ARMA Models with Genetic Algorithm. In *Applications of Evolutionary Computing* (pp. 335–342). Italy: Springer Berlin Heidelberg. http://doi.org/10.1007/3-540-45365-2_35

Morettin, P. A. (n.d.). Wavelet Based Time-varying Vector Autoregressive Modelling, (October 2006).

Moukadem, A., Djaffar Ould Abdeslam, & Dieterlen, A. (2014). *Time-Frequency Domain for Segmentation and Classification of Non-Statinary Signals* (1st ed.). New Jersey, USA: ISTE LTD, John Wiley and sons.

Murru, N., & Rossini, R. (2016). A Bayesian approach for initialization of weights in backpropagation neural net with application to character recognition. *Neurocomputing*, *193*, 92–105. http://doi.org/10.1016/j.neucom.2016.01.063

Muthuswamy, J. (2004). Biomedical Signal Analysis. In *Standard Handbook of Biomedical Engineering and Design* (13th ed., pp. 1–30). Tempe, Arizona: McGraw-Hill Companies.

Najeeb, A. R., Salami, M. J. E., Gunawan, T., & Aibinu, a M. (2016). Review of Parameter Estimation Techniques for Time-Varying Autoregressive Models of Biomedical Signals. *International Journal of Signal Processing Systems*, *4*(3), 220–225. http://doi.org/10.18178/ijsps.4.3.220-225

Nosrati, H., Shamsi, M., Taheri, S. M., & Sedaaghi, M. H. (2015). Adaptive Networks Under Non-Stationary Conditions: Formulation, Performance Analysis, and Application. *IEEE Transactions on Signal Processing*, *63*(16), 4300–4314. http://doi.org/10.1109/TSP.2015.2436363

Olmos, S., & Laguna, P. (2000). Steady-state msb convergence of lms adaptive filters with deterministic reference inputs with applications to biomedical signals. *IEEE Transactions on Signal Processing*, *48*(8), 2229–2241. http://doi.org/10.1109/78.852004

Ouelli, A., Elhadadi, B., Aissaoui, H., & Bouikhalene, B. (2015). Adaptive Spectral Estimation of Non-Stationary Bimedical Signals Based on Autoregressive Modeling and Kalman Filtering. *American Journal of Electrical and Electronic Engineering*, *2(4)*, 59–67.

Palaniappan, R. (2006). Towards Optimal Model Order Selection for Autoregressive Spectral Analysis of Mental Tasks Using Genetic Algorithm. *International*

Journal of Computer Science and Network Security, *6*(1), 153–162. Retrieved from http://paper.ijcsns.org/07_book/200601/200601A22.pdf

Paleologu, C., Benesty, J., & Ciochiňa, S. (2008). A robust variable forgetting factor recursive least-squares algorithm for system identification. *IEEE Signal Processing Letters*, *15*(3), 597–600. http://doi.org/10.1109/LSP.2008.2001559

Pally, R. K. (2009). *Implementation of Instantaneous Frequency Estimation based on Time-Varying AR Modeling*. Blacksburg, Virgina.

Paul, A. S., Wan, E. a, & Nelson, A. T. (2006). Noise reduction for heart sounds using a modified minimum-mean squared error estimator with ECG gating. *Conference Proceedings : ... Annual International Conference of the IEEE Engineering in Medicine and Biology Society. IEEE Engineering in Medicine and Biology Society. Conference*, *1*, 3385–90. http://doi.org/10.1109/IEMBS.2006.259809

Rajan, J. J., Rayner, P., & Godsill, S. J. (1996). A Bayesian Approach to Parameter Estimation and Interpolation of Time-Varying Autoregressive Processes using the Gibbs Sampler. *IEE Proceedings Visin, Image and Signal Processing*, *144*(4), 249–256. http://doi.org/10.1049/ip-vis:19971305

Rajan, J. J., & Rayner, P. J. W. (1996). Generalize Feature Extraction for Time Varying Autoregressive Models. *IEEE Transactions on Signal Processing*, *44*(October). http://doi.org/10.1109/78.539034

Rangayyan, R. M. (2003). *Biomedical Signal Analysis, A case-study Approach*. (Stamatios V. Kartalopoulos, Ed.) (1st ed.). New Jersey, USA: Wiley Interschience.

Rao, T. S. (1970). The Fitting of Non-stationary Time-Series Models with Time-Dependent Parameters. *Journal of the Royal Statistical Society. Series B (Statistical Methodology)*, *32*(2), 312–322. Retrieved from http://www.jstor.org/stable/2984537

Raul Rojas. (1996). *Neural Networks - A Systematic Introduction* (1st ed.). New York: Springer-Verlag, Berling.

Reddy, G. R. S., & Rameswar Rao. (2014). Instantaneous Frequency Estimation Based On Time-Varying Auto Regressive Model And WAX-Kailath Algorithm. *An International Journal of Signal Processing (SPIJ)*, *8*(4), 43–66.

Reddy, G. R. S., & Rao, R. (2014). Non-Stationary Signal Prediction Using TVAR Model. In *International Conference on Communications and Signal Processing (ICCSP), 2014* (pp. 1692–1697). Melmaruvathur,Hyderbad, India: IEEE. http://doi.org/10.1109/ICCSP.2014.6950136

Reddy, G. R. S., & Rao, R. (2015). Instantaneous Frequency Estimation of Multi-Component Non- Stationary Signals using Fourier Bessel series and Time-Varying Auto Regressive Model. In *International Journal of Electronics and*

Telecommunications (Vol. 61, pp. 365–376). http://doi.org/10.2478/eletel-2015-0048

Reddy, G. R. S., Rao, R., & College, C. V. R. (2014a). Non-stationary Signal Analysis using TVAR Model. *International Journal of Signal Processing, Image Processing and Pattern Recognition*, *7*(2), 411–430.

Reddy, G. R. S., Rao, R., & College, C. V. R. (2014b). Performance Analysis of Basis Functions in TVAR Model. *International Journal of Signal Processing, Image Processing and Pattern Recognition*, *7*(3), 317–338. http://doi.org/http://dx.doi.org/10.14257

Reddy, R. S., & Rao, R. (2015). Time-Varying Autoregressive Model Using Multi-Wavelet Basis Functions. *Computer Science and Telecommunication 2015*, *2*(2), 47–60.

Rui Zou, HengLiang Wang, & Ki. H. Chon. (2003). A Robust Time-Varying Identification Algorithm Using Basis Function. *Annals of Biomedical Engineering*, *31*, 840–853.

Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1994). Learning Internal Representation by Error Propogation. In *Parallel Distributed Processing, Vol. ! Foundations* (pp. 318–362). M.I.T. Press.

S.A.Billings, H.l.Wei, & J.Liu. (2008). *Time-Varying Parametric Modelling and Time-Dependent Spectral Characterisation with Applications To EEG Signals Using Multi-Wavelets*. Sheffield , United Kingdom.

Saini, D. K., & Prasad, R. (2010). Order Reduction of Linear Interval Systems Using Genetic Algorithm, *2*(5), 316–319.

Sandsten, M. (2016). *Time-Frequency Analysis of Time-Varying Signals and Non-Stationary Processes*. Lund, Sweeden. Retrieved from http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.278.3775&rep=rep1&type=pdf

Sanubari, J., & Keiichi Tokuda. (1997). Non Stationary Spectral Estimation Based on Robust Time Varying A R Model Excited by t-Distribution process. In *IEEE TENCON Speech and Image Technologies for Computing and Telecommunication* (pp. 51–54). Japan: IEEE. http://doi.org/10.1109/ICASSP.1992.226568

Schalk, G., McFarland, D. J., Hinterberger, T., Birbaumer, N., & Wolpaw, J. R. (2004). BCI2000: A general-purpose brain-computer interface (BCI) system. *IEEE Transactions on Biomedical Engineering*, *51*(6), 1034–1043. http://doi.org/10.1109/TBME.2004.827072

Schiopu, R., Barbulescu, C., Kilyeni, S., Deacu, A., & Vernica, A. (2015). ANN backpropagation power consumption forecasting. In *International Conference on*

Computer as a Tool - EUROCON 2015. Salamanca, Spain: IEEE. http://doi.org/10.1109/EUROCON.2015.7313774

Seman, N., Bakar, Z. A., & Bakar, N. A. (2010). The Optimization of Artificial Neural Networks Connection Weights using Genetic Algorithms for Isolated Spoken Malay Parliamentary Speeches. In *Proceedings of ICCIA 2010 - 2010 International Conference on Computer and Information Application* (pp. 162–166). IEEE. http://doi.org/10.1109/ICCIA.2010.6141561

Sharma, L. D., Asery, R., Sunkaria, R. K., & Mittal, D. (2016). Comparative study of fetal ECG elicitation using adaptive filtering techniques. In *Proceeding of IEEE - 2nd International Conference on Advances in Electrical, Electronics, Information, Communication and Bio-Informatics, IEEE - AEEICB 2016* (Vol. 3). http://doi.org/10.1109/AEEICB.2016.7538399

Sharnian, K. C., & Friedlander, B. (1984). Time-Varying Autoregressice modeling of a class of nonstationary signals. In *IEEE International Conference on Acoustics, Speech adn Signal Processing , ICASSP 84* (pp. 2–5). IEEE. http://doi.org/10.1109/ICASSP.1984.1172536

Shaw, L., & Routray, A. (2015). Efficacy of adaptive directed transfer function for neural connectivity estimation of EEG signal during meditation. In *2nd International Conference on Signal Processing and Integrated Networks, SPIN 2015* (pp. 198–202). IEEE. http://doi.org/10.1109/SPIN.2015.7095413

Sodhi, S. S., Chandra, P., & Tanwar, S. (2014). A New Weight Initialization Method for Sigmoidal Feedforward Artificial Neural Networks. In *International Joint Conference on Neural Networks (IJCNN 2014)* (Vol. 110078, pp. 291–298). Beijing, China: IEEE. Retrieved from http://ieeexplore.ieee.org/document/6889373/

Sodsri, C. (2003). *Time Varying Autoregressive Modelling for Non-Stationary Signal and Its Frequency Analysis*. The Pennsylavania State Univeristy. Retrieved from https://etda.libraries.psu.edu/files/final_submissions/1261

Sukh, S., & Benja, C. (2014). Variable Forgetting Factor RLS Algorithm doe Adaptive Echo Cancellation. In *International Conference on Control, Automation and Systems (ICCAS 2014)* (pp. 971–974). Korea: IEEE. Retrieved from http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=6987926

Tang, K. S., Man, K. F., Kwong, S., & He, Q. (1996). Genetic algorithms and their applications. *IEEE Signal Processing Magazine*, *13*(November), 22–37. http://doi.org/10.1109/79.543973

Thonet, G., & Vesin, J.-M. (1997). Stationarity assessment with time-varying autoregressive modeling. *1997 IEEE International Conference on Acoustics, Speech, and Signal Processing*, *5*, 3721–3724. http://doi.org/10.1109/ICASSP.1997.604677

Tian, J., Juhola, M., & Grönfors, T. (1997). AR parameter estimation by a feedback neural network. *Computational Statistics & Data Analysis*, *25*(1), 17–24. http://doi.org/10.1016/S0167-9473(96)00084-9

Tsatsanis, M. K., Giannakis, G. B., & Member, S. (1993). Time-Varying System Identification and Model Validation Using Wavelets. *IEEE Transactions on Signal Processing*, *41*(12), 3512–3523. http://doi.org/10.1109/78.258089

Turan, C., Salman, M. S., & Eleyan, A. (2016). A new variable step-size block LMS algorithm for a non-stationary sparse systems. In *Proceedings of the 2015 12th International Conference on Electronics Computer and Computation, ICECCO 2015* (pp. 1–4). Almaty: IEEE. http://doi.org/10.1109/ICECCO.2015.7416869

Ukte, A., & Kizilkaya, A. (2016). Comparing the Performances of Least Mean Squares Based Multirate Adaptive Filters, (6), 0–4.

W. Read. (1990). *Model Order Determination Methods For Autregressive Radio Direction Finding Techniques*. Ottawa, Canada. Retrieved from http://pubs.drdc-rddc.gc.ca/BASIS/pcandid/www/engpub/DDW?W%3DSYSNUM=64096

Wacker, M., & Witte, H. (2013). Time-frequency techniques in biomedical signal analysis. a tutorial review of similarities and differences. *Methods of Information in Medicine*, *52*(4), 279–96. http://doi.org/10.3414/ME12-01-0083

Wang, J. (2009). A Variable Forgetting Factor RLS Adaptive Filtering Algorithm. In *Proceedings - 2009 3rd IEEE International Symposium on Microwave, Antenna, Propagation and EMC Technologies for Wireless Communications, MAPE 2009* (Vol. 2, pp. 1127–1130). http://doi.org/10.1109/MAPE.2009.5355946

Wang, W., & Wang, W. (2003). Research on parameter estimation of time-varying AR model. *IEEE International Symposium on Personal, Indoor and Mobile Radio Communications, PIMRC*, *3*, 2378–2382. http://doi.org/10.1109/PIMRC.2003.1259144

Wangberg, S. C. (2008). An Internet-based diabetes self-care intervention tailored to self-efficacy. *Health Education Research*, *23*(1), 170–9. http://doi.org/10.1093/her/cym014

Wei, H. L., Liu, J., Billings, S. A., & Street, M. (2010). Time-Varying Parametric Modelling and Time-Dependent Spectral Characterisation with Applications To EEG Signals Using Multi-Wavelets. *International Journal of Modelling, Identification and Control*, *9*(3), 215–224. Retrieved from http://eprints.whiterose.ac.uk/74634/1/977.pdf

White, J. K. H. P. R. (1996). the Analysis of Non-Stationary Signals Using Time-Frequency Methods. *Journal of Sound and Vibration*, *190*(3), 419–447. http://doi.org/10.1006/jsvi.1996.0072

Whitley, D. (1994). *A genetic algorithm tutorial*. *Statistics and Computing* (Vol. 4). http://doi.org/10.1007/BF00175354

Widrow, B. (1966). *Adaptive Filters I : Fundamentals*. Stanford University, California. Retrieved from http://www-isl.stanford.edu/~widrow/papers/t1966adaptivefilters.pdf

Widrow, B., & Hoff, M. (1960). Adaptive switching circuits. *1960 IRE WESCON Convention Record*.

Widrow, B., McCool, J. M., Larimore, M. G., & Johnson, C. R. (1976). Stationary and Nonstationary Learning Characteristics of the Lms Adaptive Filter. In *Proceedings of the IEEE* (Vol. 64, pp. 1151–1162). http://doi.org/10.1109/PROC.1976.10286

Wiesel, A., Bibi, O., & Globerson, A. (2013). Time Varying Autoregressive Moving Average Models for Covariance Estimation. *IEEE Transactions on Signal Processing*, *61*(11), 2791–2801. http://doi.org/10.1109/TSP.2013.2256900

Witte, H., Ungureanu, M., Ligges, C., Hemmelmann, D., Wüstenberg, T., Reichenbach, J., … Leistritz, L. (2009). Signal Informatics as an Advanced Integrative Concept in the Framework of Medical Informatics. *Methods of Information in Medicine*, 18–28. http://doi.org/10.3414/ME9133

Wu, A., & Liu, K. J. R. (1996). Split Recursive Least- Squares : Algorithms , Architectures , and Applications. *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, *43*(9), 645–658. http://doi.org/10.1109/82.536761

Wu, B., & Chang, C.-L. (2002). Using genetic algorithms to parameters (d; r) estimation for threshold autoregressive models. *Computational Statistics & Data Analysis*, *38*, 315–330. http://doi.org/10.1016/S0167-9473(01)00030-5

Yam, J. Y. F., & Chow, T. W. S. (2000). A weight initialization method for improving training speed in feedforward neural network. *Neurocomputing*, *30*(1-4), 219–232. http://doi.org/10.1016/S0925-2312(99)00127-7

Yeremia, H., Yuwano, N. A., & Raymond, P. (2013). Genetic Algorithm and Neural Network for Optical Character Recognition. *Journal of Computer Science*, *9*(11), 1435–1442. http://doi.org/10.3844/jcssp.2013.1435.1442

Yu, X., Efe, M. O., & Kaynak, O. (2002). A General Backpropagation Algorithm for Feedforward Neural Networks Learning. *IEEE Transactions on Neural Networks / a Publication of the IEEE Neural Networks Council*, *13*(1), 251–254. http://doi.org/10.1109/72.977323

Zanchettin, C., Ludermir, T. B., & Almeida, L. M. (2011). Hybrid training method for MLP: Optimization of architecture and training. *IEEE Transactions on Systems,* *Man, and Cybernetics, Part B: Cybernetics*, *41*(4), 1097–1109. http://doi.org/10.1109/TSMCB.2011.2107035

Zhang, L., Xiong, G., Liu, H., Zou, H., & Guo, W. (2010a). Time- representation based on time-varying autoregressive model with applications to non-stationary rotor vibration analysis. *Sadhana - Academy Proceedings in Engineering Sciences*, *35*(2), 215–232. http://doi.org/10.1007/s12046-010-0016-y

Zhang, L., Xiong, G., Liu, H., Zou, H., & Guo, W. (2010b). Time-Frequency Representation Based on Time Varying Autoregressive Model With Applications To Non-Stationary Rotor Vibration Analysis. *Indian Academy of Science*, *35*(April), 215–232. http://doi.org/doi.org/10.1007/s12046-010-0016-y

Zhang, Q., & Wang, C. (2008). Using Genetic Algorithm to Optimize Artificial Neural Network: A Case Study on Earthquake Prediction. In *2008 Second International Conference on Genetic and Evolutionary Computing* (pp. 128–131). Hubei, CN: IEEE. http://doi.org/10.1109/WGEC.2008.96

Zhang, Y., Chambers, J. a., Wang, W., Kendrick, P., & Cox, T. J. (2007). A New Variable Step-Size LMS Algorithm with Robustness to Nonstationary Noise. *2007 IEEE International Conference on Acoustics, Speech and Signal Processing - ICASSP '07*, (1), III–1349–III–1352. http://doi.org/10.1109/ICASSP.2007.367095

Zhang, Z. G., Hung, Y. S., & Chan, S. C. (2011). Local polynomial modeling of time-varying autoregressive models with application to time-frequency analysis of event-related EEG. *IEEE Transactions on Bio-Medical Engineering*, *58*(3), 557–66. http://doi.org/10.1109/TBME.2010.2089686

Zhao, Y. (1996). On-line neural network learning algorithm with exponential convergence rate. *Electronics Letters*, *32*(15), 1381–1382. http://doi.org/10.1049/el:19960895

Zou, R., & Chon, K. H. (2004). Robust algorithm for estimation of time-varying transfer functions. *IEEE Transactions on Bio-Medical Engineering*, *51*(2), 219–28. http://doi.org/10.1109/TBME.2003.820381

APPENDIX A

PUBLICATIONS AND AWARDS

Journals

(1) Najeeb, A. R., Salami, M. J. E., Gunawan, T., & Aibinu, A.M. (2016). Review of Parameter Estimation Techniques for Time-Varying Autoregressive Models of Biomedical Signals. *International Journal of Signal Processing Systems*, *4*(3), 220–225. http://doi.org/10.18178/ijsps.4.3.220-225

(2) A. R. Najeeb, A. M. Aibinu, M. N. Nwohu, M. J. E. Salami and A.H. B. Salau, "Performance Analysis of Clustering Based Genetic Algorithm," *2016 International Conference on Computer and Communication Engineering (ICCCE)*, Kuala Lumpur, 2016, pp. 327-331. doi: 10.1109/ICCCE.2016.76

Conferences

(1) Najeeb, A. ., Salami, M. J. E., & Aibinu, A. M. (2015). Review of Parameter Estimation Techniques for Time-Varying Autoregressive Models of Biomedical Signals. In *International Conference on Signal and Information Processing (ICSIP 2015)*. Bangkok.

(2) Najeeb, Athaur Rahman and Gunawan, Teddy Surya and Aibinu, Abiodun Musa (2017) *Nonstationary signal reconstruction from TVAR coefficients.* In: 2017 IEEE International Conference on Smart Instrumentation, Measurement and Applications (ICSIMA 2017), 27th-29th Nov. 2017, The Everly Putrajaya Hotel, Kuala Lumpur

Award

(1) GOLD MEDAL AWARD at IIUM’s Kulliyyah of Engineering Research and Innovation Exhibitin (KERIE) 2013

APPENDIX B

SOURCE CODES

MAIN File

```
function TVARBPANN(y)
tic
global LearningRate XX TT
LearningRate=0.065;
NFFT=512;
InputNodes=5; % InutNodes=ModelOrder
HiddenLayer=10;
OutputLayer =1;
F=[ 200];  % Frequency in the signal
SNR=80;      % Signal to Noise Ratiothe model
% =======
% Generating the signal and cutting out neces
%[y]=EffectOfNoise(F,SNR);

%PRE-PROCESSING
lN =length(y);
y=y(1:lN);
[X,T] = ARNNToolFormated(y./max(y),InputNodes);

gap=1.0;
InputS=y./max(y);

for t=1:InputNodes
    yest(t)=InputS(t);
    r2(t)=InputS(t);
    GAr2(t)=InputS(t);

end

signal='Signal test BP ANN ';
SignalWrite=('Signal test2 BP ANN ');
SignalWriteReorder='Signal tst  BP ANN Coef Reorder';

fileID = fopen('Signaltst_BPANNGA.txt','w');

M=[0,0;0,0];

dlmwrite(SignalWrite,M);
 datanumber=InputNodes;

% START PROCESSING
for q=1:length(X)
    XX=X(:,q);   % Input Data Preprocessing
```

```
   TT=T(:,q);  % Target Data Preprocessing
   datanumber=datanumber+q;

   [NewTVARCoef]= TVARUsingANN(XX,TT,HiddenLayer);    % ANN to compute
TVAR
   TVARCoefficients(q,1:InputNodes+1)=NewTVARCoef    % Storing TVAR
coefficients
   lb= NewTVARCoef(2:end)- gap;
   ub= NewTVARCoef(2:end)+ gap;
   [x,fval,exitflag,output,population,score] = InteractiveGA(InputNodes,lb,ub);
   GATVARCoefficients(q,1:InputNodes+1)=[1,x];
  % tracking TVAR coefficients first 5 values
  a1(1)=datanumber;
  a1(2)=GATVARCoefficients(1,2);
  a1(3)=GATVARCoefficients(1,3);
  a1(4)=GATVARCoefficients(1,4);
  a1(5)=GATVARCoefficients(1,5);
  a1(6)=GATVARCoefficients(1,6);

  dlmwrite(SignalWrite,a1,'-append');
end
%
%
save('TestTVAR7b','X','T','y','TVARCoefficients','GATVARCoefficients','XXD','XX
XD')


% read recorded TVAR coefficients and reorder
M=dlmread(SignalWrite);
M2 =M.';
dlmwrite(SignalWriteReorder,M2);

%%%% reconstruction.

[RowX,ColX]=size(X);
for k=1:ColX
r(k)=TVARCoefficients(k,2:end)*X(:,k);
r2(InputNodes+k)=TVARCoefficients(k,2:end)*X(:,k);
end
for k=1:ColX
GAr(k)=GATVARCoefficients(k,2:end)*X(:,k);
GAr2(InputNodes+k)=GATVARCoefficients(k,2:end)*X(:,k);
end


figure(1000);
plot(T,'rs')
hold on
plot(r,'g')
```

```
hold on
plot(GAr,'b')
legend(['Original Signal'], ['TVAR signal '], ['GA Optimized TVAR'])
title('Comparision with 3 methods');
grid on;
axis tight;

 err22=abs(InputS-GAr2).^2;
 MSE_Gar2 = sum(err22(:))/length(InputS);

 err=abs(InputS-r2).^2;
 MSE_tvar2 = sum(err(:))/length(InputS);

% =======================
% Performance Measurement

 %MSE ( Error Signal )
 errBPANN=abs(InputS-r2).^2;
 MSE_BPANN = sum(errBPANN(:))/length(y);

 errBPANNGA=abs(InputS-GAr2).^2;
 MSE_BPANNGA = sum(errBPANNGA(:))/length(y);

 % Cross correlation
 CC_BPANN2= corrcoef(InputS,r2);
 CC_BPANN=CC_BPANN2(1,2);


 CC_BPANNGA2= corrcoef(InputS,GAr2);
 CC_BPANNGA=CC_BPANNGA2(1,2);


 % RMSE
RMSE_BPANN=sqrt(MSE_BPANN);
RMSE_BPANNGA=sqrt(MSE_BPANNGA);

%PRD
SumSquareInput=sum(InputS.^2);

PrBPANN=sum(errBPANN(:))/SumSquareInput;
PRD_BPANN = 100* sqrt(PrBPANN);

PrBPANNGA=sum(errBPANNGA(:))/SumSquareInput;
PRD_BPANNGA = 100* sqrt(PrBPANNGA);

%SNR
SNR_BPANN=10*log10(SumSquareInput/sum(errBPANN(:)));
SNR_BPANNGA=10*log10(SumSquareInput/sum(errBPANNGA(:)));
```

```
%partial
err2=abs(T-GAr).^2;
MSE_Gar = sum(err2(:))/length(y);

err=abs(T-r).^2;
MSE_tvar = sum(err(:))/length(y);

fprintf('\n==========================TVAR BP ANN GA REPORT
============\n');
fprintf('%s',signal);
fprintf('Model Order %f\n', InputNodes);
fprintf('Boundaries Unir Circle %f\n', gap);
fprintf('Mean Square Predicted Error BP ANN  partial %f\n', MSE_tvar);
fprintf('Mean Square Predicted Error BP ANN GA partial  %f\n', MSE_Gar);
fprintf('\n======== BP ANN  (Chebyshev)========= \n');
fprintf('MSE BP ANN  full %f\n', MSE_BPANN);
fprintf('CC BP ANN  full %f\n', CC_BPANN);
fprintf('RMSE BP ANN  full %f\n', RMSE_BPANN);
fprintf('PRD BP ANN  full %f\n', PRD_BPANN);
fprintf('SNR BP ANN  full %f\n', SNR_BPANN);


fprintf('\n======== BP ANN GA (Chebyshev)========= \n');
fprintf('MSE BP ANN GA full %f\n', MSE_BPANNGA);
fprintf('CC BP ANN GA full %f\n', CC_BPANNGA);
fprintf('RMSE BP ANN GA full %f\n', RMSE_BPANNGA);
fprintf('PRD BP ANN GA full %f\n', PRD_BPANNGA);
fprintf('SNRBP ANN GA full %f\n', SNR_BPANNGA);

 fprintf('\n======== END ========= \n');

 %open file to record data
fprintf(fileID,'\r\n==========================TVAR BP ANN GA REPORT
============\r\n');
fprintf(fileID,' %s\r\n ',signal);
fprintf(fileID,'Model Order %f \r\n', InputNodes);
fprintf(fileID,'Boundaries Unir Circle %f \r\n', gap);
fprintf(fileID,'%s',signal);
fprintf(fileID, 'Mean Square Predicted Error BP ANN  partial %f \r\n', MSE_tvar);
fprintf(fileID,'Mean Square Predicted Error BP ANN GA partial  %f \r\n', MSE_Gar);
fprintf(fileID,'\r\nn======== BP ANN  (Chebyshev)========= \r\n');
fprintf(fileID,'MSE BP ANN  full %f \r\n', MSE_BPANN);
fprintf(fileID,'CC BP ANN  full %f \r\n', CC_BPANN);
fprintf(fileID,'RMSE BP ANN  full %f \r\n', RMSE_BPANN);
fprintf(fileID,'PRD BP ANN  full %f \r\n', PRD_BPANN);
fprintf(fileID,'SNR BP ANN  full %f \r\n', SNR_BPANN);
fprintf(fileID,'\r\nn======== BP ANN GA (Chebyshev)========= \r\n');
fprintf(fileID,'MSE BP ANN GA full %f \r\n', MSE_BPANNGA);
fprintf(fileID,'CC BP ANN GA full %f \r\n', CC_BPANNGA);
```

```
fprintf(fileID,'RMSE BP ANN GA full %f \r\n', RMSE_BPANNGA);
fprintf(fileID,'PRD BP ANN GA full %f \r\n', PRD_BPANNGA);
 fprintf(fileID,'SNRBP ANN GA full %f \r\n', SNR_BPANNGA);

fclose(fileID);

figure(2000);
plot(GAr2,'b')
hold on
plot(InputS,'-r*')
strtitle=sprintf('%s Real Signal vs Predicted ',signal);
%legend('GA Optimized TVAR ANN BF');
legend(['BPANN-GA'],['Real'])
title(strtitle);

grid on;
print('\BPANNFig\Signal1_BPANN','-r 300','-dpng');
saveas(figure(2000),[pwd '\BPANNFig\Signal1_BPANN.fig']);


figure(2003);
plot(r2,'b');
hold on
plot(InputS,'-r*')
strtitle2=sprintf('%s Real Signal vs Predicted',signal);
legend(['BPANN'],['Real'])
title(strtitle2);

grid on
print('\BPANNFig\Signal1_BPANN_GA','-r 300','-dpng');
saveas(figure(2003),[pwd '\BPANNFig\Signal1_BPANN_GA.fig']);
end
```

Data Formatting

```
function [X,T] = ARNNToolFormated(y,AROrder)
y=y(:);
XFormatted= fbaw_over(y,AROrder,(((AROrder-1)*100/AROrder)));
P=y(AROrder+1:end).'; % CREATE THE TARGET DATA
A=length(XFormatted);
B=length(P);
if A > B
   X=XFormatted(:,1:B);
   T=P(1:B);
elseif A<B
   X=XFormatted(:,1:A);
   T=P(1:A);
```

```
elseif A==B
   X=XFormatted;
   T=P;
End
```

Artificial Neural Network

```
function [NewTVARCoef]= TVARUsingANN(X,T,HiddenLayer)
[TVAROrder,Column]=size(X);
minmaxA= minmax(X);
net = newff(minmaxA,[HiddenLayer 1],{'purelin','purelin'});
net.trainParam.show=NaN;
net.biasConnect = [ 0 ; 0];
net = train(net,X,T);
% output = sim(net,X);
Weight1=net.IW{1,1};
Weight2=net.LW{2,1};
Bias1=net.B{1}; Bias2=net.B{2};
AWeightHidden=Weight1';
AWeightOutput=Weight2';
for Output= 1: TVAROrder;
   InitialTVARCoefX=0;
   for Reference=1:HiddenLayer
      ZZ = AWeightOutput(Reference,1)*AWeightHidden(Output,Reference);
      InitialTVARCoefX= InitialTVARCoefX + ZZ;
   end
   TVARCoef(Output)=InitialTVARCoefX;
end
NewTVARCoef=[1 -fliplr(TVARCoef)];
```

Genetic Algorithm

```
function [x,fval,exitflag,output,population,score] = InteractiveGA(nvars,lb,ub)
%% This is an auto generated MATLAB file from Optimization Tool.

%% Start with the default options
options = gaoptimset;
%% Modify options setting
options = gaoptimset(options,'Display', 'off');
options = gaoptimset(options,'PlotFcns', {  @gaplotexpectation @gaplotselection });
[x,fval,exitflag,output,population,score] = ...
ga(@MyFunct,nvars,[],[],[],[],lb,ub,[]
```

GA Fitness Function

```
function [ y ] = MyFunct( x )
%UNTITLED Summary of this function goes here
%   Detailed explanation goes here
global XX TT

A= x(:);
tt=XX'*A;
y=abs(TT-tt);

% y = 3*x(1) + 2*x(2).^2 - 36;
End
```

Autoregressive Method

```
function tvarmanual(y)
 NFFT=512;
InputNodes=10 % InutNodes=ModelOrder
HiddenLayer=10;
OutputLayer =1;
F=[ 200];  % Frequency in the signal
SNR=80;      % Signal to Noise Ratiothe model
% =======
% Generating the signal and cutting out neces
%[y]=EffectOfNoise(F,SNR);
lN =length(y)
y=y(1:lN);
%[X,T] = ARNNToolFormated(y./max(y),InputNodes);

order=10;
y=y./max(y); %format
%predicte Y initialization
for t=1:order
    yest(t)=y(t);
     y2(t)=y(t);
end

%prediction starts from P+1 data
for q=2:length(y)

    %intermediate y ; generating y(n-1), y(n-2), y(n-3) ..y(n-p_
   y2(q)=y(q);

   if(q-order >0 ) order2=order; end
   if(q-order <0)  order2=q+1; end

   if(q-order<0)
```

```
      fprintf('order for %i is %i\n',q,order2);
   end
   for pp=1:order2

    [A_yw E(pp)]=aryule(y2,pp);
     AIC(pp)=log(E(pp)) + 2*pp/(length(y2));
  end
    % get the minimim order
    [minAIC, orderAIC] = min(AIC);
    fprintf('order for %i is %i',q,minAIC');
    neworder=orderAIC
    [A E(q)]=aryule(y2,neworder);
    %reconstruct the y
    %y2(q) is the ori
   % yest(q) is the estimated
   sum_vari=0;
    for i=1:neworder
        vari=0;
        if((q-i) > 0)
         vari=A(i+1)*y2(q-i);
        end
        sum_vari=sum_vari+vari;
    end
    yest(q)=(-1*sum_vari);


FormatedData=TVARReconstructionDataMatrix(y2,orderAIC);
NewARCoeff= fliplr(A(2:end));
MyOutput= [1, -NewARCoeff*FormatedData];
end

figure(1);plot(y);title('Original Signal');grid on;
figure(2);plot(y2);title('Original Signal 2');grid on;
figure(3);plot(yest);title('Predicted Signal ');grid on;
figure(4);plot(MyOutput);title('Predicted Musa reconstr Signal ');grid on;

end
```

TVAR Correlation

```
function TVARCorr2(y)
lN =length(y);

ModelOrder=10;
order=ModelOrder;
signal='BioSignal2 TVAR Corr';
SignalWrite='BioSignal2 TVAR Corr';
SignalWriteReorder='BioSignal2 TVAR Corr Coef Reorder';
```

```
fileID = fopen('Biosignal_2_TVAR.txt','w');
% Data Nrmalization and Formatting

[X,T] = ARNNToolFormated(y./max(y),order);
y=y./max(y); %format
M=[0,0;0,0];
dlmwrite(SignalWrite,M);

% initialization of signals
% estimation starts at y>order

for t=1:order
    yest(t)=y(t);
    yn(t)=y(t);
    MyOutput2(t)=y(t);

end

%prediction starts from P+1 data
%datanumber for recording in file purpose
datanumber=order;
length(X);
tic;
for q=1:length(X)
    XX=X(:,q);
    y2=XX;
    datanumber=datanumber+1;

    % correlation, determination of a(n) ; starting from n=model orderder
     [A_yw E]=aryule(y2,order);
     AIC=log(E) + 2*order/(length(y2));
     NewARCoeff= fliplr(A_yw(2:end));
     MyOutput(q)= -NewARCoeff*y2;
     MyOutput2(order+q)=MyOutput(q);

% tracking TVAR coefficients first 5 values
   a1(1)=datanumber;
   a1(2)=A_yw(2);
   a1(3)=A_yw(3);
   a1(4)=A_yw(4);
   a1(5)=A_yw(5);
   a1(6)=A_yw(6);
   a1(7)=A_yw(7);
   a1(8)=A_yw(8);
   a1(9)=A_yw(9);
   a1(10)=A_yw(10);
   a1(11)=A_yw(11);
   dlmwrite(SignalWrite,a1,'-append');
```

```
end
ElapseTime=toc;
% read recorded TVAR coefficients and reorder
M=dlmread(SignalWrite);
M2 =M.';
dlmwrite(SignalWriteReorder,M2);

strtitle2=sprintf('Reconstructed using TVAR ANN Correlation  %s with Order
%i',signal,ModelOrder);

strtitle1=sprintf('BioSignal 1 ');
%strtitle=sprintf('%s Real VS Reconstructed',signal);
strtitle=sprintf('%s Real VS Reconstructed (Order %i)',signal,ModelOrder);

figure(1000);
plot(y);
hold on;
plot(MyOutput2,'r--o');
legend(['Real '],['Reconstructed']);
title(strtitle);
grid on;
xlabel('Sample');
ylabel('Amplitude');
saveas(figure(1000),[pwd '\Biosignal2\Signal_2_CMP_real_TVARCORR.fig']);
saveas(figure(1000),[pwd '\Biosignal2\Signal_2_CMP_real_TVARCORR.png']);

figure(1);
plot(y);
title(strtitle1);
grid on;axis on; xlabel('Samples'), ylabel('Amplitude');

%figure(2);plot(yest);title(strtitle2);grid on;axis on; xlabel('Samples'),
ylabel('Amplitude');
%figure(3);plot(MyOutput);title('Predicted matric mul ');grid on;
%figure(4);plot(MyOutput2);title(strtitle2);grid on;;axis on; xlabel('Samples'),
ylabel('Amplitude');
figure(100);
plot(MyOutput2);
strtitle100=sprintf('%s reconstructed',signal);
title(strtitle100);
grid on
saveas(figure(100),[pwd '\Biosignal2\BioSignal_2_TVARCORR.fig']);
saveas(figure(100),[pwd '\Biosignal2\BioSignal_2_TVARCORR.png']);

%figure(5);plot(T);title('Original Signal T ');grid on;

% Performance Measurement

 %MSE ( Error Signal )
```

```
 err3=abs(y-MyOutput2).^2;
 MSE_myoutput = sum(err3(:))/length(y)
%std deviaton.. not used
StdInput=std(y);
StdMyoutput=std(MyOutput2);
MeanInput=mean(y);
MeanMyouput=mean(MyOutput2);


ccc= corrcoef(y,MyOutput2)
cc=ccc(1,2)
size(ccc)

%RMSE
MyRMSE1=sqrt(MSE_myoutput)


SumSquareInput=sum(y.^2)

%PRD
%cr=sum(y)/sum(MyOutput2)
display('prd');
SquareDiffInputOutput=(y-MyOutput2).^2;
SumSquareDiffInput=sum(SquareDiffInputOutput)
sum(err3(:))

Pr=sum(err3(:))/SumSquareInput
PRD = 100* sqrt(Pr)

%err3
sum(err3(:))

bb=SumSquareInput/sum(err3(:))
v=log10((SumSquareInput/sum(err3(:))))
10*log10((SumSquareInput/sum(err3(:))))

SNR=10*log10(SumSquareInput/sum(err3(:)))
%snr(y,MyOutput2)

 fprintf('\n========================TVAR ANN  REPORT
===========\n');
 fprintf('MSE %f\n',  MSE_myoutput);
 fprintf('Cross Correlation  %f\n',  cc);
fprintf('RMSE  %f\n', MyRMSE1);
 fprintf('PRD  %f\n',  PRD);
 fprintf('SNR  %f\n',  SNR);
 fprintf('Model Order %i\n', ModelOrder);
 fprintf('Signal  %s\n',signal);
  fprintf('Elapse Time %f\n',ElapseTime);
```

```
   fprintf('\n=========================TVAR ANN  REPORT
===========\n');
 fprintf(fileID,'MSE %f\n',  MSE_myoutput);
 fprintf(fileID,'Cross Correlation  %f\n',  cc);
fprintf(fileID,'RMSE  %f\n', MyRMSE1);
 fprintf(fileID,'PRD  %f\n',  PRD);
 fprintf(fileID,'SNR  %f\n',  SNR);
 fprintf(fileID,'Model Order %i\n', ModelOrder);
 fprintf(fileID,'Signal  %s\n',signal);
   fprintf('Elapse Time %f\n',ElapseTime);
end
```

Basis Function Chebyshev

```
function [BF]=BFChebyshev(m,N,n)

%[q,N]=size(Xt);
%m=expansion  model order
%N=data length ( denoted as n)
%n is sample number ( basically q from main main file)
 k=(2*(n-1))/(N-1);
 for mm=1:m+1
 BF(mm)=cos(m*acos(k-1));
End
```

Basis Function DCT

```
function [DCTBF]=BFCosine(m,n,q)

%[q,N]=size(Xt);
%m=expansion  model order
%n=data length
%q is sample number ( q is ARN_BFTVAR)

  alpha=0;
 for mm=1:m+1
        %mm
        if(mm==1)
            alpha(mm)=sqrt(1/n);
        elseif(mm>1)
            alpha(mm)=sqrt(2/n);
        end
        DCTBF(mm)=alpha(mm)*cos((pi*mm*(2*q+1))/(2*n));

   End
```

Basis Function Fourier

```
function [DCTBF]=BFFourier(m,n,q)

%[q,N]=size(Xt);
%m=expansion  model order
%n=data length
%q is sample number ( q is ARN_BFTVAR)
 for mm=1:m+1
       %mm
       if mod(mm,2) == 0   %number is even
       DCTBF(mm)=cos((1/n)*mm-1*q);
       else
        %number is odd
        DCTBF(mm)=sin((1/n)*mm-1*q);
       end

   end
```

Basis Function Legendre

```
function [BF]=BFLegendre(m,N,n)

%[q,N]=size(Xt);
%m=expansion  model order
%N=data length ( denoted as n)
%n is sample number ( basically q from main main file)

 for mm=1:m

     if(mm ==1)
        BF(mm)=1;
     end
     if(mm==2)
        BF(mm)=(2*(n-1))/(N-2);
       % y=BF(2);
     end

     if(mm==3)
        BF(mm)=((3*BF(2)^2)-1)/2;
     end
       if(mm>2)
           BF(mm+1)=( ((2*mm+1)*BF(2)*BF(mm))-(mm*BF(mm-1)))/(mm+1);
       end
   end
```

Basis Function Walsh

```
function [BF]=BFWalshOri(m,N,n)

%[q,N]=size(Xt);
%m=expansion  model order
%N=data length ( denoted as n)
%n is sample number ( basically q from main main file)
% XX is the original data sequence

for mm=1:m+1
     bin=de2bi(mm);
     s=1;
    for dd=1:length(bin)
        s=s*sign(cos(bin(dd)*2^(dd-1)*pi*n));
    end
    BF(mm)=s;
 end
```

Basis Function Gram Schmidt Orthogonalization

```
function [BF]=BFGramSchmidt(m,N,n)

%[q,N]=size(Xt);
%m=expansion  model order
%N=data length ( denoted as n)
%n is sample number ( basically q from main main file)
%Compose of Walsh and Legendre Basis Funcion


L=BFLegendre(m,N,n)
W=BFWalshOri(m,N,n)

A= [ L ; W ];
 [Q R] = qr(A);
 [S T]= size(R);
 BF=R(1,:)
```

TVAR Coefficient Estimation using BF and GA

```
function ARN_BFTVAR_CHEB(y)
%Written By Athaur Rahman Najeeb
%input NS Signal
% Process, BF, computation of expansion coefficient, TVAR coeff from BP ANN
% and link to GA Optimization

tic
```

```
global LearningRate XX TT
LearningRate=0.065;
NFFT=512;
InputNodes=10; % InutNodes=ModelOrder
HiddenLayer=10;
ModelOrder=InputNodes;
OutputLayer =1;
F=[ 200];  % Frequency in the signal
SNR=80;      % Signal to Noise Ratiothe model
gap=1.0;

InputS=y./max(y); %Data Normalization
orderP=InputNodes;
%Record File Initialization

signal='Biosignal 2 Cheby BF ';
SignalWrite=('Biosignal 2 Cheby  BF ');
SignalWriteReorder=('Biosignal 2 Cheby BF Reorder');
SignalWriteGA='Biosignal 2 BF Cheby  BF GA';
SignalWriteReorderGA=('Biosignal 2 BF Cheby GA Reorder');

fileID = fopen('Biosignal_2_BF_Cheby.txt','w');
fileID2=fopen('Biosignal_2_Data_BF_Cheby.txt','w');

M=[0,0;0,0];
dlmwrite(SignalWrite,M);

% Generating the signal and cutting out neces
%[y]=EffectOfNoise(F,SNR);
lN =length(y);

[X,T] = ARNNToolFormated(y./max(y),InputNodes);

%Initialization for Signal Predicton
for t=1:InputNodes
   yest(t)=InputS(t);
   r2(t)=InputS(t);
   GAr2(t)=InputS(t);

end

datanumber=InputNodes; % Writing to File

%set m =5, basis expansion
m=5;
for q=1:length(X)
   XX=X(:,q);   % Input Data Preprocessing
   TT=T(:,q);  % Target Data Preprocessing
```

```
    datanumber=InputNodes+q;


    %DCT BASIS FUNCTION
    %DCTBF=BFFourier(m,length(XX),q);

    %  %Discrete Cosine Basis Function
    %CBF=BFCosine(m,length(XX),q);

    % Chebychev Basis Function
    CheBF=BFChebyshev(m,length(XX),q);

    BF=CheBF;

    % new inputdata BF*data
    NewInputNodes=length(XX)*m;
    hhh=0;
    v=length(XX)+1;
    for hj=1:length(XX)

        for mmm=1:m+1
            hhh=hhh+1;
            NewDataInput(hhh)=XX(v-hj,1)*BF(mmm);
        end
    end

%compute TVAR expansion coefficient
 [AE AERR]=aryule(NewDataInput,length(NewDataInput));
 Coeff(1:length(NewDataInput)+1)=AE;
 NewCoeff=Coeff(2:end);
 NewCoeffMat=vec2mat(NewCoeff,m+1);

 %Compute TVAR Coefficient from Expansion Coefficient
 for(u=1:orderP)
      TVARBFCoeff(q,u)=sum(NewCoeffMat(u,:));
 end
  if(q==2)
      size(Coeff)
       size(NewCoeff)
       Coeff
        NewCoeff
         TVARBFCoeff

         size(T)
         size(NewDataInput)
  end
 %compute coefficient
%GA
    lb= TVARBFCoeff(q,:)-gap;
```

```
   ub= TVARBFCoeff(q,:)+gap;

   [x,fval,exitflag,output,population,score] = InteractiveGA(InputNodes,lb,ub);
   GATVARCoefficients(q,1:InputNodes+1)=[1,x];
 %AA=reshape(NewCoeff(m,

  % recording
   a1(1)=datanumber

   a1(2)=GATVARCoefficients(q,2);
   a1(3)=GATVARCoefficients(q,3);
    a1(4)=GATVARCoefficients(q,4);
     a1(5)=GATVARCoefficients(q,5);
      a1(6)=GATVARCoefficients(q,6);
      a1(7)=GATVARCoefficients(q,7);
      a1(8)=GATVARCoefficients(q,8);
      a1(9)=GATVARCoefficients(q,9);
      a1(10)=GATVARCoefficients(q,10);
      a1(11)=GATVARCoefficients(q,11);

      if(q==2)
          a1
      end

      %bf coefficients
   tv(1)=datanumber;
   tv(2)=TVARBFCoeff(q,1);
   tv(3)=TVARBFCoeff(q,2);
   tv(4)=TVARBFCoeff(q,3);
   tv(5)=TVARBFCoeff(q,4);
   tv(6)=TVARBFCoeff(q,5);
    tv(7)=TVARBFCoeff(q,6);
     tv(8)=TVARBFCoeff(q,7);
      tv(9)=TVARBFCoeff(q,8);
       tv(10)=TVARBFCoeff(q,9);
        tv(11)=TVARBFCoeff(q,10);

         dlmwrite(SignalWriteGA,a1,'-append');
   dlmwrite(SignalWrite,tv,'-append');
end
ElapseTime=toc;

M=dlmread(SignalWrite);
M2 =M.';
dlmwrite(SignalWriteReorder,M2);

MM=dlmread(SignalWriteGA);
MM2 =MM.';
dlmwrite(SignalWriteReorderGA,MM2);
```

```
%reconstruction

size(TVARBFCoeff)
size(T)

[RowX,ColX]=size(X);
for k=1:ColX
r(k)= TVARBFCoeff(k,:)*X(:,k);
r2(orderP+k)=TVARBFCoeff(k,:)*X(:,k);
end

for k=1:ColX
GAr(k)=GATVARCoefficients(k,2:end)*X(:,k);
GAr2(InputNodes+k)=GATVARCoefficients(k,2:end)*X(:,k);
end

%write data
fprintf(fileID2,'Original,');
for gh=InputNodes+1:InputNodes+10
   fprintf(fileID2,'%f,',InputS(gh));
end
fprintf(fileID2,'\nBF,');
for gh2=InputNodes+1:InputNodes+10
   fprintf(fileID2,'%f,',r2(gh2));
end
fprintf(fileID2,'\nBF-GA,');
for gh3=InputNodes+1:InputNodes+10
   fprintf(fileID2,'%f,',GAr2(gh3));
end

figure(1000);
plot(InputS,'-.')
hold on
plot(r2,'-m')
hold on
plot(GAr2,'-ro')
legend(['Original Signal'], ['BF Cheby '],['GA'])
title('Biosignal 2 Comp Real, Cheby and GA');
grid on;
axis tight;
saveas(figure(1000),[pwd '\Biosignal2\Biosignal_2_Cheby_CMP_3.fig']);
saveas(figure(1000),[pwd '\Biosignal2\Biosignal_2_Cheby_CMP_3.png']);



 err22=abs(InputS-GAr2).^2;
   MSE_Gar2 = sum(err22(:))/length(InputS);
```

```
   err=abs(InputS-r2).^2;
    MSE_tvar2 = sum(err(:))/length(InputS);

     SumSquareErrorBF=sum((r2-InputS).^2);
    SumSquareErrorBFGA=sum((GAr2-InputS).^2);


% =======================
% Performance Measurement

 %MSE ( Error Signal )
 errBF=abs(InputS-r2).^2;
 MSE_BF= sum(errBF(:))/length(y);

 errBFGA=abs(InputS-GAr2).^2;
 MSE_BFGA = sum(errBFGA(:))/length(y);

 % Cross correlation
 CC_BF2= corrcoef(InputS,r2);
 CC_BF=CC_BF2(1,2);


 CC_BFGA2= corrcoef(InputS,GAr2);
 CC_BFGA=CC_BFGA2(1,2);


 % RMSE
RMSE_BF=sqrt(MSE_BF);
RMSE_BFGA=sqrt(MSE_BFGA);

%PRD
SumSquareInput=sum(InputS.^2);

PrBF=sum(errBF(:))/SumSquareInput;
PRD_BF = 100* sqrt(PrBF);

PrBFGA=sum(errBFGA(:))/SumSquareInput;
PRD_BFGA = 100* sqrt(PrBFGA);

%SNR
SNR_BF=10*log10(SumSquareInput/sum(errBF(:)));
SNR_BFGA=10*log10(SumSquareInput/sum(errBFGA(:)));

%partial
   err2=abs(T-GAr).^2;
    MSE_Gar = sum(err2(:))/length(y);

   err=abs(T-r).^2;
```

```
MSE_tvar = sum(err(:))/length(y);



fprintf('\n=========================REPORT============\n');
fprintf('%s',signal);
 fprintf('Model Order %f\n', InputNodes);
  fprintf('Expansion Model Order %f \r\n',m);
 fprintf('Boundaries Unir Circle %f\n', gap);
% fprintf('Mean Square Predicted Error BP ANN  partial %f\n', MSE_tvar);
%  fprintf('Mean Square Predicted Error BP ANN GA partial  %f\n', MSE_Gar);
 fprintf('\n======== BF (Chebyshev ========= \n');
fprintf('MSE BF full %f\n', MSE_BF);
fprintf('CC BF full %f\n', CC_BF);
  fprintf('RMSE BF  full %f\n', RMSE_BF);
  fprintf('PRD BF  full %f\n', PRD_BF);
   fprintf('SNR BF full %f\n', SNR_BF);
    fprintf('SUM Square Error %f\n',SumSquareErrorBF)

   fprintf('\n======== GA Optimization========= \n');
   fprintf('MSE BF GA full %f\n', MSE_BFGA);
  fprintf('CC BF GA full %f\n', CC_BFGA);
  fprintf('RMSE BF GA full %f\n', RMSE_BFGA);
  fprintf('PRD BF GA full %f\n', PRD_BFGA);
   fprintf('SNR BF GA full %f\n', SNR_BFGA);
   fprintf('elapseTime %f\n',ElapseTime);
 fprintf('SUM Square Error GA %f\n',SumSquareErrorBFGA)
  fprintf('\n======== END ========= \n');

 %open file to record data
 fprintf(fileID,'\r\n=========================TVAR BP ANN GA
REPORT ============\r\n');
  fprintf(fileID,' %s\r\n ',signal);
 fprintf(fileID,'Model Order %f \r\n', InputNodes);
  fprintf(fileID,'Expansion Model Order %f \r\n',m);
 fprintf(fileID,'Boundaries Unir Circle %f \r\n', gap);
 fprintf(fileID,'%s',signal);
 %  fprintf(fileID, 'Mean Square Predicted Error BP ANN  partial %f \r\n',
MSE_tvar);
 %  fprintf(fileID,'Mean Square Predicted Error BP ANN GA partial  %f \r\n',
MSE_Gar);
  fprintf(fileID,'\r\nn======== BF GA (Chebyshev)========= \r\n');
fprintf(fileID,'MSE BF full %f \r\n', MSE_BF);
  fprintf(fileID,'CC BF full %f \r\n', CC_BF);
     fprintf(fileID,'RMSE BF  full %f \r\n', RMSE_BF);
       fprintf(fileID,'PRD BF full %f \r\n', PRD_BF);
  fprintf(fileID,'SNR BF full %f \r\n', SNR_BF);
   fprintf(fileID,'SUM Square Error %f\n',SumSquareErrorBF);
      fprintf(fileID,'\r\nn======== BP ANN GA (DCT)========= \r\n');
```

```
   fprintf(fileID,'MSE BF GA full %f \r\n', MSE_BFGA);
   fprintf(fileID,'CC BF GA full %f \r\n', CC_BFGA);
 fprintf(fileID,'RMSE BF GA full %f \r\n', RMSE_BFGA);
fprintf(fileID,'PRD BF GA full %f \r\n', PRD_BFGA);
 fprintf(fileID,'SNR BP ANN GA full %f \r\n', SNR_BFGA);
 fprintf(fileID,'elapseTime %f\n',ElapseTime);
  fprintf(fileID,'SUM Square Error GA %f\n',SumSquareErrorBFGA);


    fclose(fileID);

figure(2000);
plot(GAr2,'b')
hold on
plot(InputS,'-r*')
strtitle=sprintf('%s Real Signal vs Reconstructed Cheby -GA (Order %i,%i)
',signal,ModelOrder,m);
%legend('GA Optimized TVAR ANN BF');
legend(['Cheby-GA'],['Real'])
title(strtitle);
grid on;
%print('\BPANNFig\Signal_1_BPANN','-r 300','-dpng');
saveas(figure(2000),[pwd '\Biosignal2\Biosignal_2_CMP_Cheby_GA.fig']);
saveas(figure(2000),[pwd '\Biosignal2\Biosignal_2_CMP_Cheby_GA.png']);



figure(2003);
plot(r2,':');
hold on
plot(InputS,'-r*')
strtitle2=sprintf('%s Real Signal vs Reconstructed Cheby (Order
%i,%i)',signal,ModelOrder,m);
legend(['Cheby'],['Real'])
title(strtitle2);
grid on
%print('\BPANNFig\Signal_1_BPANN_GA','-r300','-dpng');
saveas(figure(2003),[pwd '\Biosignal2\Biosignal_2_CMP_Cheby.fig']);
saveas(figure(2003),[pwd '\Biosignal2\Biosignal_2_CMP_Cheby.png']);



figure(2004);
plot(r2);
strtitle2004=sprintf('%s Reconstructed Cheby  (Order %i,%i)',signal,ModelOrder,m);
legend(['Cheby'])
title(strtitle2004);
grid on
%print('\BPANNFig\Signal_1_BPANN_GA','-r300','-dpng');
```

```
saveas(figure(2004),[pwd '\Biosignal2\Biosignal_2_Cheby.fig']);
saveas(figure(2004),[pwd '\Biosignal2\Biosignal_2_Cheby.png']);

figure(2005);
plot(GAr2);
strtitle2005=sprintf('%s Reconstructed Cheby  GA (Order
%i,%i)',signal,ModelOrder),m;
legend(['Cheby-GA'])
title(strtitle2005);
grid on
%print('\BPANNFig\Signal_1_BPANN_GA','-r300','-dpng');
saveas(figure(2005),[pwd '\Biosignal2\Biosignal_2_Cheby_GA.fig']);
saveas(figure(2005),[pwd '\Biosignal2\Biosignal_2_Cheby_GA.png']);

%{
%instaneous frequency
z = hilbert(T); Fs=400; % for myns1
instfreq = Fs/(2*pi)*diff(unwrap(angle(z)));
suminst1=sum(instfreq);
figure(1000);
plot(instfreq);
xlabel('Time');
ylabel('IF Hz');
grid on;
title('Original Signal Instantaneous Frequency');

z2 = hilbert(r); Fs=400;
instfreq2 = Fs/(2*pi)*diff(unwrap(angle(z2)));
suminst2=sum(instfreq2);
figure(1002);
plot(instfreq2);
xlabel('Time');
ylabel('IF Hz');
grid on;
iftitle1=sprintf('IF Plot TVAR ANN BF %s GA ',cbf);
title(iftitle1);

z3 = hilbert(GAr); Fs=400;
instfreq3 = Fs/(2*pi)*diff(unwrap(angle(z3)));
suminst3=sum(instfreq3);
figure(1001);
plot(instfreq3);
xlabel('Time');
ylabel('IF Hz');
grid on;
iftitle2=sprintf('IF Plot TVAR ANN BF %s GA %.1f',cbf,gap);
title(iftitle2);

figure(2000);
```

```
subplot(2,2,1);
plot(instfreq);
title('Original Signal IF ');
subplot(2,2,2);
plot(instfreq2,'r');
title('TVAR ANN BF Signal IF');
subplot (2,2,3);
plot(instfreq3,'g');
title('TVAR ANN BF-GA Signal IF');

size(instfreq);
size(instfreq2);
size(instfreq3);

errIF_TVARANNBF=suminst1-suminst2;
errIF_TVARANNBFGA=suminst1-suminst3;

 mseerror=abs(instfreq-instfreq2).^2;
    MSE_IF_TVAR = sum(mseerror(:))/length(instfreq);

     errif_mse2=abs(instfreq-instfreq3).^2;
     MSE_IF_TVAR_GA = sum(errif_mse2(:))/length(instfreq);


   err=abs(T-r).^2;
    MSE_tvar = sum(err(:))/length(y);

fprintf('IF TVAR ANN BF Error %f\n',errIF_TVARANNBF);
fprintf('IF TVAR ANN BF GA Error %f\n',errIF_TVARANNBFGA);

fprintf('IF TVAR ANN BF MSE Error %f\n',MSE_IF_TVAR);
fprintf('IF TVAR ANN BF GA MSE  %f\n',MSE_IF_TVAR_GA);
%}
    %}
    end
```