Adaptive Notch Filtering for Tracking Multiple Complex Sinusoid Signals

by

Paul Thomas Wheeler

A Doctoral Thesis submitted in partial fulfilment of the requirements for the award of the degree of Doctor of Philosophy (PhD), at Loughborough University.

May 2015

Advanced Signal Processing Group,
School of Electronic and Electrical Engineering,
Loughborough University, Loughborough
Leicestershire, UK, LE11 3TU.

CERTIFICATE OF ORIGINALITY

This is to certify that I am responsible for the work submitted in this thesis, that the original work is my own except as specified in acknowledgements or in footnotes, and that neither the thesis nor the original work contained therein has been submitted to this or any other institution for a degree.

..... (Signed)

..... (candidate)

I dedicate this thesis to my late Mother; and hope that it may inspire my daughters: Shannon, Lara and Megan.

Abstract

This thesis is related to the field of digital signal processing; where the aim of this research is to develop features of an infinite impulse response adaptive notch filter capable of tracking multiple complex sinusoid signals.

Adaptive notch filters are commonly used in: Radar, Sonar, and Communication systems, and have the ability to track the frequencies of real or complex sinusoid signals; thus removing noise from an estimate, and enhancing the performance of a system.

This research programme began by implementing four currently proposed adaptive notch structures. These structures were simulated and compared: for tracking between two and four signals; however, in their current form they are only capable of tracking real sinusoid signals.

Next, one of these structures is developed further, to facilitate the ability to track complex sinusoid signals. This original structure gives superior performance over Regalia's comparable structure under certain conditions, which has been proven by simulations and results.

Complex adaptive notch filter structures generally contain two parameters: the first tracks a target frequency, then the second controls the adaptive notch filter's bandwidth. This thesis develops the notch filter, so that the bandwidth parameter can be adapted via a method of steepest ascent; and also investigates tracking complex-valued chirp signals.

Lastly, stochastic search methods are considered; and particle swarm optimisation has been applied to reinitialise an adaptive notch filter, when tracking two signals; thus more quickly locating an unknown frequency, after the frequency of the complex sinusoid signal jumps.

Contents

Statement of Originality

The contributions of this thesis are focussed upon the development of an adaptive notch filter, for the tracking of multiple complex sinusoid signals. The novelty of the contributions is supported by the following international journals and conference papers.

In Chapter 3 four ANFs are compared, and this comparison has been published along with the CANF development from Chapter 4 in:

1. P. T. Wheeler, and J. A. Chambers, "Real Adaptive Notch Filter Analysis and a New Structure Developed for Tracking Complex Sinusoidal Signals", in the 9th IMA International Conference on Mathematics in Signal Processing, Dec. 2012.

Chapter 4 contains the original CANF development, which has been published as an Electronics Letter in:

2. P. T. Wheeler, and J. A. Chambers, "Complex adaptive notch filter structure for tracking multiple complex sinusoidal signals", Electronics Letters, vol.49, no.3, pp. 179–181, Jan. 2013.

The novel adaptation of the notch bandwidth parameter is considered in Chapter 5, which is published in:

3. P. T. Wheeler, and J. A. Chambers, "Simultaneous Adaptation of Bandwidth and Frequency Parameters within a Complex Notch Filter", Intelligent Signal Processing Conference 2013 (ISP 2013), IET, London, pp. 1–6, Dec. 2013.

Chapter 6 then considers tracking complex-valued chirp signals, and adapting the notch bandwidth parameter for this type of signal, which has been published in:

4. P. T. Wheeler, and J. A. Chambers, "Tracking complex-valued multicomponent chirp signals using a complex notch filter with adaptive

bandwidth and frequency parameters", IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 6414–6418, May 2014.

A collection of the research from Chapters 3 to 6 is due to be published in:

5. P. T. Wheeler, J. A. Chambers and P. A. Regalia, "A Survey of Digital All-pass Filter based Real and Complex Adaptive Notch Filters", Chapter 4, Festschrift for Professor A. Constantinides, 2015.

In Chapter 7 stochastic search methods are considered, and a publication from this chapter is currently in preparation as:

6. P. T. Wheeler, and J. A. Chambers, "Exploiting Particle Swarm Optimisation to Reinitialise a Complex Adaptive Notch Filter whilst Tracking Multiple Frequencies".

Acknowledgements

I am extremely grateful to my supervisor Professor Jonathon Chambers for his kind interest, generous support and constant advice throughout the past four years. I have benefited tremendously from his rare insight, and exceptional knowledge. This thesis would never have been written without his patient mentoring. It is my very great privilege to have been one of his research students.

I would like to express my sincere thanks to Doctor Paul Lepper the internal examiner, and Doctor Robert Edwards for acting as second supervisor, particularly for providing guidance and advice; and to Doctor Syed Mohsen Naqvi for providing additional support to resources when required. I would also like to thank Professor Philip Regalia for providing further useful insight into the topic of my thesis.

I'd also like to express my gratitude to some of the people who have influenced me in industry. Firstly, Professor Paul Davies for advice, ideas and for his recommendation to the advanced signal processing group at Loughborough. Also, Bob Gregory from Rolls-Royce for his support and encouragement and for sharing some of his vast knowledge in electronic product development. Lastly, I'd like to thank my current employer General Electric Aviation Systems for paying the final year of my tuition fees.

P. T. Wheeler

May, 2015

List of Acronyms

ANF	Adaptive Notch Filter
CANF	Complex Adaptive Notch Filter
CORDIC	COordinate Rotation DIgital Computer
CSS	Complex Sinusoid Signal
CVCS	Complex-Valued Chirp Signal
DCS	Direct Coefficient Scaling
DSP	Digital Signal Processing
ECG	ElectroCardioGraphy
FIR	Finite Impulse Response
FPGA	Field Programmable Gate Array
FTF	Fast recursive-least-squares Transversal Filter
GNSS	Global Navigation Satellite System
IIR	Infinite Impulse Response
I&Q	In-phase and Quadrature
LMS	Least Mean Square
NFA	Notch Filter A

NFB	Notch Filter B
NLMS	Normalised LMS
PID	Proportional Integral Derivative
PLL	Phase Locked Loop
PSO	Particle Swarm Optimisation
QPSK	Quadrature Phase Shift Keying
RLS	Recursive Least Squares
RMS	Root Mean Square
RSS	Real Sinusoid Signal
SNR	Signal-to-Noise Ratio
SNR_IR	SNR Improvement Ratio
SQP	Sequential Quadratic Programming

List of Symbols

Bold and upper-case letters represent vectors and matrices. Capitals are used to denote Z-transforms. Some frequently used notations are as follows:

$\mathbf{a}_{opt}$	Optimum Wiener filter
$b(n)$	Bandpass output
$E\{.\}$	Mathematical expectation operator
$e^{j\omega}$	Exponential form of a complex number
$e(n)$	Notch filter output
$d(n)$	Desired signal
$\hat{\boldsymbol{d}}$	Filter output
g	Global best position in PSO: 'gbest'
$grad(n)$	Gradient term: for the frequency being tracked
$grad\_\alpha(n)$	Gradient term: for the notch bandwidth parameter
$H(z)$	Transfer function
$H_{ap}(z)$	All-pass transfer function
$H_{bp}(z)$	Band-pass transfer function
$H_{notch}(z)$	Notch transfer function

(i)	Index of a particle in PSO
Im	Imaginary part of a complex signal
j	$\sqrt{-1}$
J	Cost function
K	Step ahead forward prediction error filter
m	Mean value
N	Tap length: of a filter
(n)	Discrete time index
P	Power spectral density
p_i	Particle's best position in PSO: 'pbest'
Re	Real part of a complex signal
R_{xx}	Auto-correlation matrix
$\mathbf{r}_{xd}$	Cross-correlation vector
$u(n)$	Input signal
v_i	Velocity of a particle in PSO
$x(n)$	Regressor term in an equation-error approach
x_i	Particle coordinates in PSO
$y(n)$	Output signal
z	Z-transform
α	Alpha: the parameter that controls the notches bandwidth
β	Beta: the parameter that adapts to the tracked frequency

γ	Gamma: the forgetting factor, a value from 0 to 1
θ	Theta: the phase angle, which relates to β
$\hat{\theta}(n)$	Estimate at one instance
ι	Iota: the inertia weight in PSO
μ	Mu: the step size or adaptation gain
σ^2	Sigma: the variance of an estimate
ψ	Psi: the recursive calculation of gradient energy
χ	Chi: the constriction factor in PSO
Σ	Sigma notation: Sum of lower to upper limits
$\int$	Integral
$\oint_c$	Contour integral
$_0$ *or* (0)	Initial value of a parameter
$\omega(n)$	Estimated output frequency
$\bigtriangledown$	Difference in a signal
$\angle$	Phase angle of a CSS
∂	Differentiated value
$\Delta\omega$	Sample bias of an estimate
$(.)^*$	Complex conjugate
$\|.\|$	Modulus of a complex number
†	A value smaller than 0.00005

List of Figures

List of Tables

Chapter 1

INTRODUCTION INCLUDING AIMS AND OBJECTIVES

1.1 Introduction to Adaptive Notch Filters

The specific aim of this PhD programme is to develop aspects of a Complex Adaptive Notch Filter (CANF) capable of tracking multiple Complex Sinusoid Signal (CSSs); with primary applications in Radar, Sonar and Communication Systems. This research has focussed primarily on four areas, which are:

1. Creating a new type of structure,
2. Adapting the notch bandwidth parameter in the structure,
3. Considering the tracking of a complex-valued chirp signal, and
4. Enhancing the performance of the system further by implementing stochastic search methods to reinitialise the notch filter.

This introductory chapter is split into three sections: where the first section introduces the topic of Adaptive Notch Filters (ANFs), next some of the applications for ANFs are listed in the second section; then in the third section, the aims, objectives, and the structure of this thesis are discussed. Thus to begin, ANFs are introduced, where firstly their application in Digital Signal Processing (DSP) is highlighted.

Generically, the aim in signal processing is to manipulate data via advanced mathematical techniques; and in adaptive filtering unknown signals or data is located and separated from a noisy environment, and adaptation is necessary if the signal is unknown or time-varying. Specifically, ANFs locate, extract, and track, sinusoid signals from a sampled data source; and this process is shown in Figure 1.1.

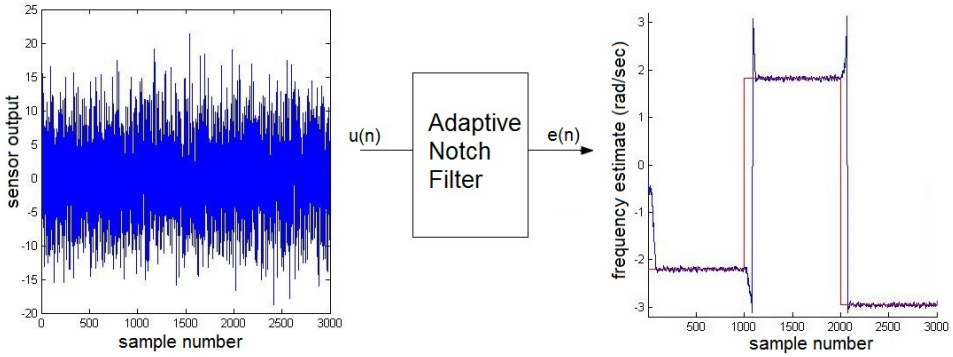


Figure 1.1. A conventional adaptive notch filter simulation, showing the signal immersed in noise, the filter, then the frequency of the signal which corresponds to the notch frequency.

Currently, ANFs are generally synthesised by utilising all-pass functions such as [1] - [3], although previously Direct Coefficient Scaling (DCS) implementations such as [4] have also been used; an all-pass implementation is shown below in Figure 1.2.

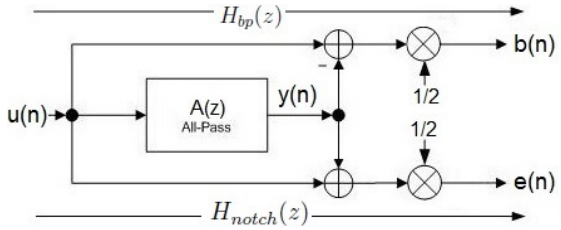


Figure 1.2. The implementation of a notch filter, which is built from an all-pass function $A(z)$; herein the bandpass output is denoted as $b(n)$, and the notch output is denoted as $e(n)$.

Therefore, considering Figure 1.2, the notch and bandpass equations are

$$H_{notch}(z) = \frac{1}{2}[1 + A(z)] \qquad H_{bp}(z) = \frac{1}{2}[1 - A(z)]; \tag{1.1.1}$$

wherein, in the form primarily applied in this thesis

$$A(z) = \frac{z^{-1}\beta - \alpha}{1 - \alpha z^{-1}\beta}. \tag{1.1.2}$$

Next, the stages of an ANF can be explained, by splitting them into the following four areas, which are labelled a) to d):

a) Time domain

Initially, all signals are sampled in the time domain, where each sample will have a numerical value: and in this thesis the data is sampled in discrete intervals of (n); however, in DSP more advanced techniques are possible in the frequency domain. Therefore in this application, a z-transform converts from a discrete time form into a complex frequency domain representation: where $z = e^{j\theta}$.

b) Frequency domain

The frequency domain shows all the individual frequency components that are present at one instant in time; and once in this domain, as the aim in this application is to locate a target frequency an inverse transform is not required. The frequency domain may also contain phase information, which is required to recombine the frequency components when converting back to the time domain; phase information is also important when considering complex frequencies.

c) Notch

A simple fixed notch filter response is shown below in Figure 1.3, this shows that two parameters can be altered; where β controls the notch frequency, and α controls the notch bandwidth.

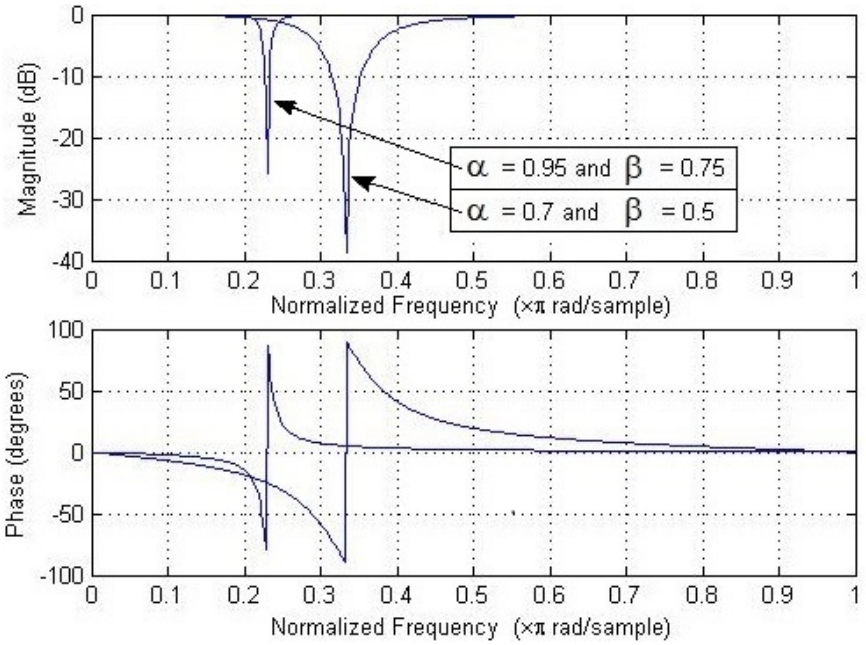


Figure 1.3. The effect of changing the notch frequency and bandwidth parameters in a fixed notch filter B structure from [1].

d) Locating unknown frequencies

In the frequency domain, by minimising a cost function, it is possible to locate an unknown target frequency with an ANF; an example of which is shown in Figure 1.4.

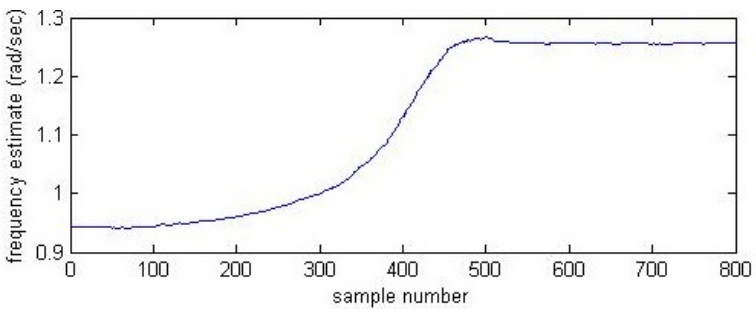


Figure 1.4. Locating a target signal with an adaptive notch filter, where the target signal frequency is 1.2566.

Adaptive notch filtering has been around for several decades, and Finite Impulse Response (FIR) structures and their realisation is indeed mature; however, the use of Infinite Impulse Response (IIR) structures is not complete or fully understood. An IIR structure applies feedback from its output, thus requires a shorter length of filter, which produces a more effective response per computation; however, this feedback creates instability if not properly constrained, whilst FIR filters are unconditionally stable. Therefore, developing a new type of IIR structure will produce a beneficial piece of research in the area of DSP.

Many structures have been proposed for ANF that facilitate the ability to track a single time-varying Real Sinusoid Signal (RSS), or multiple RSSs. Tracking these signals and eliminating noise is important practically; particularly with technology advancing to smaller geometries, increasing the vulnerability of a system to noise related errors. As we live in a changing world, noise sources will continually shift; increasing and decreasing; therefore, every system is susceptible to noise, particularly moving vehicles. Thus, being able to mitigate the effects of this changing noise; by filtering it out, is clearly key to any powerful robust system.

1.1.1 The Strengths of Adaptive Notch Filters

Why would ANFs be used? This is a question that can be asked; and to answer this question the strengths of this technology are:

1. The design [1] adapts simply and is computationally efficient, thus it is canonic in the number of multipliers and delay elements,

2. Structures are easily extendible to facilitate the tracking of multiple signals,

3. ANFs have the ability to mitigate broadband noise, and can provide Signal-to-Noise Ratio (SNR) performance enhancements; thus enhancing the performance of a system,

4. This technology provides numerically robust solutions, and

5. As this design is based on a structurally lossless prototype [1], the all-pass solution provides no gain enhancement; therefore, the signal cannot be detrimentally changed, even if there is an error in the algorithm.

This thesis focuses on developing aspects of a CANF, which is an ANF that is capable of tracking a CSS; where a CSS contains a real (Re) and an imaginary (Im) component. Although there are many ANFs, far fewer CANFs have been published to date. Tracking multiple sinusoid signals is also possible with ANFs: in real or complex form; and this is achieved by cascading the structures.

1.2 Published Applications of Adaptive Notch Filters

There have been many applications for ANFs, and the applications noted in this research programme are described now in this section; where these applications were noted from 1991-2014.

Traditionally radar and sonar are the primary applications for ANFs, and a publication in 2008 [5] shows that ANFs are still used in this type of system.

A further military application found in this review has been noted as, 'Inserting an ANF into the missile acceleration control loop sensor feedback to reject the bending mode which causes the resonance or instability' [6].

A common recent application for ANFs is communication systems, such as the global navigation satellite system [7], and a ultra-wideband based wireless multimedia system [8].

Notch filters are also commonly used in safety critical control systems: as stated in [9]; where complexity is an issue, due to design size limits in high reliability technology.

Another area where ANFs have been used is medical measurements, where notable applications are removing power-line noise from Biomedical signals: [10] - [11], and Electrocardiography (ECG) [12].

ANFs have also been implemented in various power applications, such as: tracking phases in AC systems: [13] - [14], active power filters [15], and removing voltage flicker [16].

Assisting with the mechanical resonant mode compensation problem for hard disk drives is another application, where publications are noted as [17] and [18].

In 2008 an audio application was noted as effective bass enhancement in [19]; and there has also been an ANF implementation within an earthquake simulator, which is referenced as [20].

A further notable utilisation for this technology is removing harmonics and inter-harmonics [21]; then quasi harmonic signals in [22], which was published in 2012, and provides another recent application.

One point to note is that from 2006 to 2008 ten of these referenced applications were published, highlighting that ANFs are still a fast developing

area of interest.

1.3 Aims, Objectives and Thesis Structure

The specific aim of this PhD programme is to develop aspects of a CANF; and this research has focussed primarily on four areas, which are:

1. Creating a new type of structure: Chapter 4,
2. Adapting the notch bandwidth parameter in the structure: Chapter 5,
3. Considering the tracking of a complex-valued chirp signal: Chapter 6, and
4. Enhancing the performance of the system further by implementing stochastic search methods to reinitialise the notch filter: Chapter 7.

This research began by implementing four existing adaptive notch structures, due to: Chambers & Constantinides [1], Regalia [2], Cho, Choi & Lee [3], and Kwan & Martin [4], which were simulated and compared; which is included as Chapter 3 of this thesis. However, in this form these existing structures are capable of tracking only RSS.

Next in Chapter 4, one of these structures which was originally developed for RSSs is extended to track CSSs. This provides a comparison to the recent research of Regalia detailed in [23], where the lattice structure from [2], has been developed to a form capable of tracking CSSs.

Chapter 5 of this thesis considers adapting the notch bandwidth parameter, as this should improve the performance of the CANF further. In this chapter three methods are evaluated for adapting the notch bandwidth parameter, and it is adapted in different parts of the structure: when multiple CSSs are tracked.

In Chapter 6 tracking complex-valued chirp signals is considered, and the notch bandwidth parameter is also adapted in this scenario; this chapter also considers the tracking of a complex-valued chirp signal and a hopping signal simultaneously; and includes the computational complexity of the structures proposed in Chapters 4 - 6 of this thesis: for a floating point Matlab implementation.

The final research topic considered within Chapter 7 of this thesis; applies stochastic search methods to reinitialise a notch filter, thus quickly locating a new frequency. This chapter also specifically considers applying a stochastic search method in the scenario where two signals are being tracked and one signal hops to an unknown frequency.

Lastly, this thesis is concluded in Chapter 8, where opportunities for further research are also highlighted. There is also an Appendix in Chapter 9, which contains additional information that is relevant to this thesis.

Chapter 2

REVIEW OF IIR ADAPTIVE NOTCH FILTERS

In this chapter, literature relevant to developing aspects of a Complex Adaptive Notch Filter (CANF) is reviewed. The first section contains a historical background on the field of adaptive filtering; the second section then introduces ANF structures, next section three shows the synthesis of the four structures that are evaluated in this thesis: [1] - [4], which track real sinusoid signals. Then other CANFs are discussed in the fourth section, highlighting their relevance to the later chapters of this thesis. Lastly, in the fifth section the evaluation criteria for ANFs are described which have been applied in this research.

2.1 Historical Background

The development of linear least-squares estimation theory, a subject with close links to adaptive filter theory; and adaptive filter theory has been historically documented by Kailath as [24] in 1974, and in 1986 by Haykin [25]. The studies indicate from earliest times that man has been interested in the interpretation of observations to make estimations and predictions. For instance the Babylonians employed a rudimentary form of Fourier series to this end. However, the onset of a theory of estimation is attributed to Galileo Galilei, who in 1632, attempted to minimise some functions of errors.

The next milestone was the inception point of linear least-squares estimation theory, which was completed by Gauss in 1795; when Gauss first used the method of least-squares to study the motion of heavenly bodies. Although later, argument arose as whether Gauss or Legendre initiated this theory, now it has been accepted that Gauss was indeed the pioneer.

Least-squares was applied to stochastic or non-deterministic processes by Kolmogorov, Krein and Wiener in the late 1930s and early 1940s. Although there were different aims, as they worked independently, there was an overlap in the results that they created. Next, Kolmogorov who was inspired by Wold's earlier work on the decomposition of stationary processes, then comprehensively developed the prediction of discrete time processes. Masani found that the importance of Kolomogorov's work, in the prediction of a scalar value, was so useful that he wrote "So thorough had been Kolmogorov's treatment of the univariable in the discrete case that there is little left to do", as mentioned in [24].

Relationships were recognised by Krein, between Kolmogorov's work and some earlier results by Szego on orthogonal polynomials. Which led to the application of bilinear transformations, and then these results were extended to continuous-time. The true father of linear-least squares estimation in engineering could however be seen to be Wiener, and Shannon acknowledged that "Credit should be given to Professor N. Wiener, whose elegant solution of the problems of filtering and prediction of stationary ensembles has considerably influenced the writer's thinking in the field". Solving anti-aircraft fire-control problems and providing a solution to the continuous time linear prediction problem were indeed aspects of Wiener's work. Wiener also developed an explicit formula for the optimum predictor, and considered the filtering problem of estimating a representation of an additive noise process. Wiener created an explicit formula in terms of an integral equation known as the Wiener-Hopf equation. This equation arose first in 1894 in astrophysics,

and therefore has been widely studied. Then in 1947 Levinson, formulated the classical Wiener filtering problem in discrete-time. In terms of discrete-time signals, the Wiener-Hopf integral equation is in a matrix form, and is known as the normal equation

$$R_{xx}\mathbf{a}_{opt} = \mathbf{r}_{xd}, \tag{2.1.1}$$

wherein, R_{xx} is the autocorrelation matrix of the FIR filter input vector $\mathbf{x}(n-K)$, i.e. $R_{xx} = E\{\mathbf{x}(n-K)\mathbf{x}(n-K)^T\}$ with $\mathbf{x}(n-K)^T = [x(n-K), x(n-K-1), ..., x(n-K-N+1)]$. Herein, N is the tap length of the transversal FIR filter, $E\{.\}$ is the mathematical expectation operator, n is the discrete time index, K is the step ahead parameter of the forward prediction error filter, $(.)^T$ is a vector transpose; and $\mathbf{r}_{xd}$ is the cross-correlation vector between the FIR filter input vector $\mathbf{x}(n-K)$ and the desired response $d(n)$, i.e. $\mathbf{r}_{xd} = E\{\mathbf{x}(n-K)d(n)\}$; lastly, the parameter vector of the optimum Wiener filter in transversal form is $\mathbf{a}_{opt}$.

The solution of (2.1.1) is termed as the normal equation, due to the fundamental characteristic of optimal filters in the least-squares sense which was published in 1984 by Papoulis as [26]. In a geometric sense, the estimation error $e(n)$ is normal to the vector which represents the filter output $\hat{d}(n)$ as shown in Figure 2.1. In addition the filter input data $\mathbf{x}(n-K)$, used to form $\hat{d}(n)$, is orthogonal to the estimation error; this is known as the "principal of orthogonality". A further point to add is that capitals are used in this thesis to denote matrices, whilst bold text is used to denote vectors.

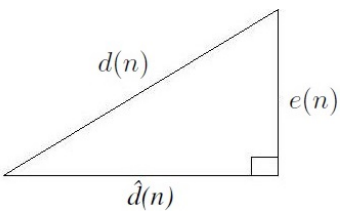


Figure 2.1. The geometric representation of the normal equation.

If the input data's statistics are stationary, then the correlation matrix R_{xx} is Toeplitz, named after Toeplitz who was a mathematician [27]. Toeplitz creates a square matrix if all the elements on its diagonal are equal, and the elements on any other diagonal parallel to the main diagonal are equal; Levinson exploited this property to find a recursive procedure for the solution of the normal equation. Levinson's motivation to formulate this algorithm was to see the effect on the filter output of increased filter order. Interestingly, he did not want to resolve the equation for each new order, and instead used the results for filter order N-1, to calculate the results for filter order N: forming a recursive algorithm. Levinson published his work in 1947 as [28] the Wiener Root Mean Square (RMS) error criterion in filter design and prediction. Durbin rediscovered Levinson's recursive algorithm in 1960, when he fitted scalar time-series data to autoregressive models. In 1963 Whittle deduced that there is close affinity between the Levinson-Durbin recursion and that of Szego's polynomials. These polynomials showed an unusual property unlike the classical orthogonal polynomials defined in a straight line, Szego's polynomials are defined on a circle. Thus, to obtain a recursion for their generation two auxiliary polynomials must be introduced.

The next major development in terms of least-squares estimation was delayed until the late 1960s. The work that followed Wiener and Kolomogorov's developments, was inappropriate for application to overcome the shortfalls evident in the precedent theory: assumption of stationarity, requirement of an infinite amount of available data, and the restriction of scalar quantities. Consequently, in the late 1950s for the space age applications; it was necessary to determine satellite orbits, from vector observations composed of position and velocity measurements. These measurements were accumulated with each pass of a satellite; therefore, the requirement for a new algorithm and approach was clear. Swerling was first to propose a new algorithm; however, it was Kalman who developed a more restrictive

algorithm, thus providing the solution to this particular space age problem. The first solution by Kalman was for a discrete-time problem; but later by collaborating with Bucy a continuous-time solution was developed, known as the Kalman-Bucy filter. Then, it was Kailath who derived a solution to the linear filter problem, creating the Kalman filter via the innovatius approach. This process was truly innovative as it allowed the whiteness to be tested, which was documented in a book by Candy in 1986 entitled 'Signal processing the model-based approach' [29].

Work on adaptive filters truly began in the 1950s and many researchers worked on various applications of this technology. Widrow and Hoff in [30] considered the application of an adaptive filter for a pattern recognition problem, and from this scheme named the 'Adaptive linear threshold logic element' came the Least Mean Square (LMS) algorithm. Theoretically, the LMS algorithm was developed by Robbins and Monroe in 1951 [31], to solve certain statistical sequential parameter estimation problems. The LMS algorithm may also sometimes be referred to as a stochastic gradient method, because on average, it moves down the error performance surface, i.e. calculating the mean-squared-error as a function of the filter parameters, in the direction of the true negative gradient. The essential difference between the stochastic approximation method and the LMS algorithm is, the former uses a reducing step size parameter. The LMS algorithm does not follow this method completely as if the step size decays to zero, the adaptive capacity would be lost, which is clearly not the case. In 1967 Sakrison [32] noted the use of a Newton search direction would improve the convergence properties of stochastic approximation methods.

Kalman filter theory was applied by Godard in 1974 to a transversal FIR filter, for the development of an adaptive filter algorithm. He was not the first researcher to implement this approach, but his approach is often regarded as the best [33]. The Recursive Least Squares (RLS) algorithm, which

is computationally complex, was obtained from a method of least-squares, this method was derived by many researchers, but Plackett is recognised as the first in [34]. As the RLS algorithm offers much faster convergence many researchers have attempted to simplify its complexity, initially by Morf in 1974 solving the deterministic dual, which parallels the stochastic problem that Levinson then went on to solve. Well known forms of these fast algorithms are the Fast Kalman algorithm [35] and the Fast RLS Transversal Filter (FTF) algorithm [36].

The concept of the design of an adaptive IIR filter which minimises a mean-square-error first appeared in signal processing in the mid 1970s, which is documented in [37] and [38]. Whereas, the derivation of algorithms for the stability theory approach for the adaptation of IIR filters, was initially considered in the late 1970s and the early 1980s by Larimore, Treichler and Johnson in [39]. Since then, many new improved adaptive algorithms have been developed; one example of this is the affine projection algorithm, which is detailed in [40].

An ANF generally minimises a mean-square-error, to adjust the frequency of its notch; thus estimating the frequency (θ) of a target signal: which is being tracked. ANFs are produced from structures, which may be cascaded to track multiple target signals. This type of approach avoids stability problems in learning in more general IIR structures.

The first references to filtering CSSs as considered in this research were [41] - [43]; which are discussed in detail in the introduction to tracking CSSs: Chapter 4 of this thesis. It should be duly noted that Chambers' thesis [44] provided an excellent source of background information for this thesis and research.

The key parts to an ANF are the 'structure' and the 'adaptive algorithm'; although, other aspects of an ANF such as: adapting the notch bandwidth parameter, evaluating the algorithm's convergence, or considering a specific

application, may provide suitable topics for research; thus considering this point, structures are discussed first.

2.2 Introduction to Adaptive Notch Filter Structures

To introduce ANF structures, firstly the two methods used to implement them are discussed; these methods are defined as Direct Coefficient Scaling (DCS) and all-pass structures; which Regalia highlights in his 2010 publication [23], and 'Adaptive IIR Filtering in Signal Processing and Control' [45], which is a research monograph that he published in 1995.

DCS structures were the original method exploited to implement ANFs, and examples of these are Kwan & Martin [4] and [46]; where Nehorai's structure [46] has been widely cited in this field. However, all-pass structures such as: [1], [2] and [3] have been published more recently. Currently, all-pass structures are considered to provide superior solutions as: 1) DCS solutions alter the magnitude response of a filter, whilst an all-pass solution passes all frequencies unaltered, and 2) all-pass structures generally require less computations in synthesis, thus providing more simplistic solutions.

When tracking multiple sinusoid signals equation-error and output-error solutions are required, and these solutions are produced by cascading DCS or all-pass structures. Hence on the next page, Figure 2.2 and Figure 2.3 demonstrate the difference between equation-error and output-error solutions.

An output-error approach minimises some function of the 'overall' notch filter output $e(n)$: which is typically the squared error; the overall error is calculated for the complete structure, and creates a gradient term $grad(n)$ for each arm of the structure. Therefore, the output-error approach requires an arm for each frequency being tracked, where these arms produce a gradient; and in Figure 2.2, the gradient terms are shown as $grad(n)\_1$ and $grad(n)\_2$.

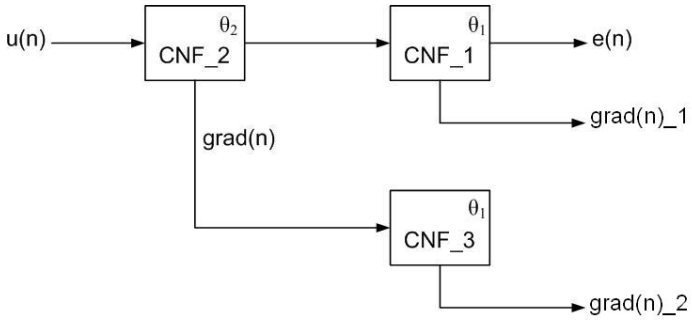


Figure 2.2. The output-error approach for tracking two sinusoid signals.

Whereas, an equation-error approach produces a local notch filter output $e(n)$, and filter regressor signal term $x(n)$ for each notch filter in the cascade. Therefore, in the equation-error approach each notch filter is tracking an individual frequency; which is being fed from the output of the previous notch filter, where the previous notch filter is assumed to have removed the frequency it has tracked from the input to the next notch filter in the chain. In Figure 2.3 below, the local errors are shown as $e(n)\_1$ and $e(n)\_2$; then the remaining regressor terms, are defined as $x(n)\_1$ and $x(n)\_2$.

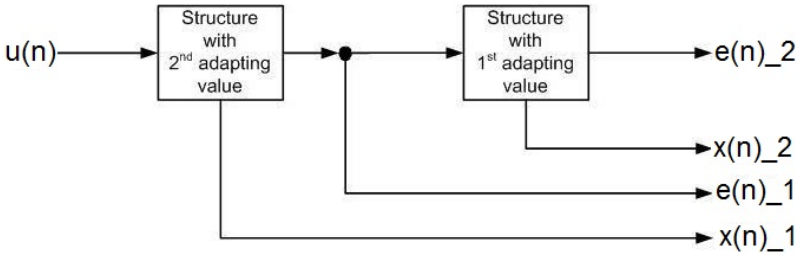


Figure 2.3. The equation-error approach for tracking two sinusoid signals.

From Figure 2.3, it can be observed that equation-error approaches simplify the overall computational complexity of a structure utilised to track multiple sinusoid signals; however, at the cost of robustness to tracking close proximity sinusoids. Thus, under certain conditions an equation-error ap-

proach is much more likely to fail than an output-error method, which is demonstrated in Chapter 3 of this thesis.

In this research four particular structures are evaluated, which were highlighted by Regalia in his 2010 publication [23]. These four structures are due to: Chambers & Constantinides [1], Regalia [2], Cho, Choi & Lee [3], and Kwan & Martin [4]. Although, Chambers & Constantinides' structure has been abbreviated to Chambers [1] in this thesis.

The four approaches selected from [23] provide a good comparison as they contain a mixture of DCS, all-pass, equation-error and output-error approaches. The justification for this statement is that Kwan & Martin's design [4] provides a DCS solution in an output-error form; the remaining three designs: [1], [2] and [3], are all-pass structures, where [2] and [3] are equation-error approaches; whilst [1] is an output-error solution. These structures are evaluated in Chapter 3 of this thesis; where Chapter 3: An Evaluation of Real Notch Filters, highlights the strengths and weaknesses of each approach; which concludes this introduction to ANF structures.

2.3 Synthesising Adaptive Notch Filter Structures

A sensible way to compare these four structures is to investigate how they are synthesised. Therefore, in this section the structures are introduced in terms of their second order notch transfer function, update equation and diagram: which shows how each structure can be implemented.

Hence from this information, all four structures can be synthesized in Matlab, in a form capable of tracking a single RSS; where the structures required for tracking multiple RSSs are included in Chapter 3.

Next, the synthesis of Chambers' structure is shown first; followed by Regalia's approach, then Cho, Choi & Lee's method, and the section is completed with the synthesis of Kwan & Martin's structure.

2.3.1 Chambers' All-pass Structure

The notch transfer function for Chambers' Notch Filter B (NFB) approach [1] is

$$H_{notch}(z) = \frac{(1+\alpha)}{2}\frac{1-2\beta z^{-1}+z^{-2}}{1-(1+\alpha)\beta z^{-1}+\alpha z^{-2}}. \tag{2.3.1}$$

In this structure α is the pole radius squared, and $\beta = \cos\omega_0$. Then, β is updated by applying the NLMS equation as follows

$$\beta(n) = \beta(n-1) - \mu\frac{e(n)grad(n)}{\psi(n)}; \tag{2.3.2}$$

wherein, $\psi(n) = \psi(n-1)\gamma + (1-\gamma)grad(n)^2$.

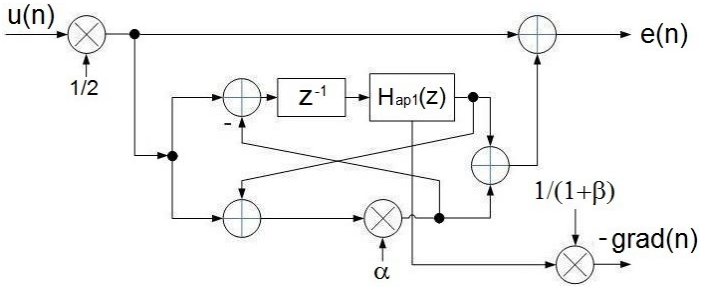


Figure 2.4. Chambers' output-error all-pass notch filter B structure [1].

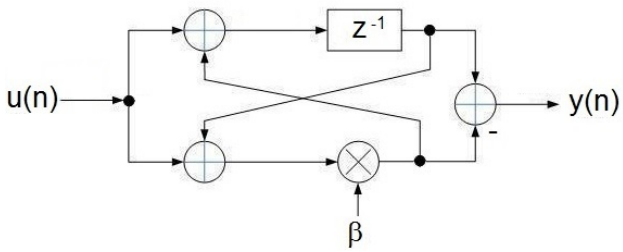


Figure 2.5. The all-pass filter $H_{ap1}(z)$ shown in Figure 2.4 from Chambers' paper [1].

2.3.2 Regalia's All-pass Solution

The second structure: Regalia's all-pass method from [2], is implemented as shown in Figure 2.6 below; wherein $x(n)$ is the regressor term.

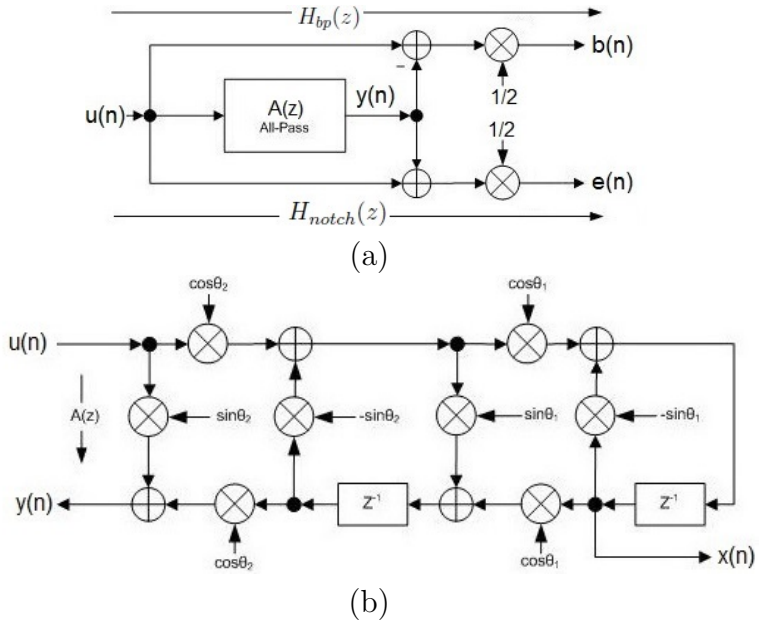


Figure 2.6. A block diagram of Regalia's original structure from 1991 [2], which is used in an equation-error approach; where (a) shows how the all-pass structure can be implemented to produce a band-pass filter and a notch filter; herein the bandpass output is denoted as $b(n)$, and the notch output is denoted as $e(n)$. Then the second part (b) demonstrates the implementation of the all-pass filter A(z).

This structure's notch transfer function is

$$H_{notch}(z) = \frac{2(k_2 + k_1(1 + k_2)z^{-1} + z^{-2})}{1 + k_1(1 + k_2)z^{-1} + k_2 z^{-2}}; \tag{2.3.3}$$

wherein, $k_2 = r^2$ and $k_1 = -\cos\omega_0$; then in synthesis $k_1 = \sin\theta_1$ and $k_2 = \sin\theta_2$.

The parameter k_1 is updated by applying the NLMS equation as follows

$$k_1(n) = k_1(n-1) - \mu\frac{e(n)x(n)}{\psi(n)}; \tag{2.3.4}$$

herein, $\psi(n) = \psi(n-1)\gamma + (1-\gamma)(x(n))^2$.

2.3.3 Cho, Choi & Lee's All-pass Method

The notch transfer function for the third structure by Cho, Choi & Lee [3] is

$$H_{notch}(z) = \frac{1 + k_0(1 + k_1)z^{-1} + k_1 z^{-2}}{1 + a_0(1 + a_1)z^{-1} + a_1 z^{-2}}; \tag{2.3.5}$$

herein $a_1 = \alpha k_1$, $k_1 = 1$ and $k_0 = a_0 = -\cos\omega_0$.

Then to implement the update the following parameters are calculated:

$A(n) = 2.x(n-1)$, $B(n) = x(n-2) + x(n-1)$,

$C(n) = \gamma C(n-1) + A(n).B(n)$, and $D(n) = \gamma D(n-1) + A(n)^2$;

wherein $x(n)$ is the regressor term. Please note that as shown above, an additional delay is required to generate the $B(n)$ parameter; and also observe that, no adaptation gain is used in this implementation; although the authors recommend setting γ to 0.5. Also, upper case letters are used to follow the convention in [3], as they correspond to calculating parameters which form the reflection coefficients.

These parameters then allow k_0 to be updated via the update $k_0 = -\frac{C(n)}{D(n)}$.

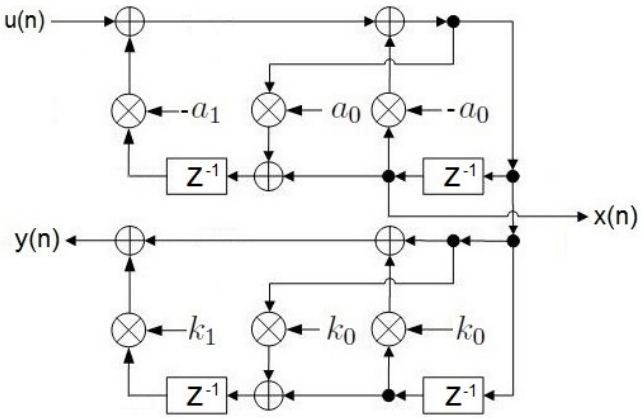


Figure 2.7. Cho, Choi & Lee's all-pass structure [3].

2.3.4 Kwan & Martin's Direct Coefficient Scaling Solution

The fourth structure considered is Kwan & Martin's structure [4], and its notch transfer function is

$$H_{notch}(z) = \frac{2-k_2}{2} \frac{1 - \frac{2(2-k_2-k_1^2)}{2-k_2} z^{-1} + z^{-2}}{1-(2-k_2-k_1^2)z^{-1}+(1-k_2)z^{-2}}; \tag{2.3.6}$$

wherein, r is the pole radius which equates to $r = \sqrt{1-k_2}$, and $k_1 = \sqrt{1-r^2-2r\cos\omega_0}$, herein ω_0 is the notch frequency.

The sensitivity output $s'(n)$: which equates to the gradient with the adjustment $grad(n) = \frac{2s'(n)}{k_2}$; updates the tracking parameter k_1 via the Normalised Least Mean Square (NLMS) update as follows

$$k_1(n) = k_1(n-1) - \mu \frac{e(n)grad(n)}{\psi(n)}; \tag{2.3.7}$$

herein, μ is the adaptation gain; and $\psi(n) = \psi(n-1)\gamma + (1-\gamma)grad(n)^2$; where, $0 \ll \gamma < 1$: which is the forgetting factor.

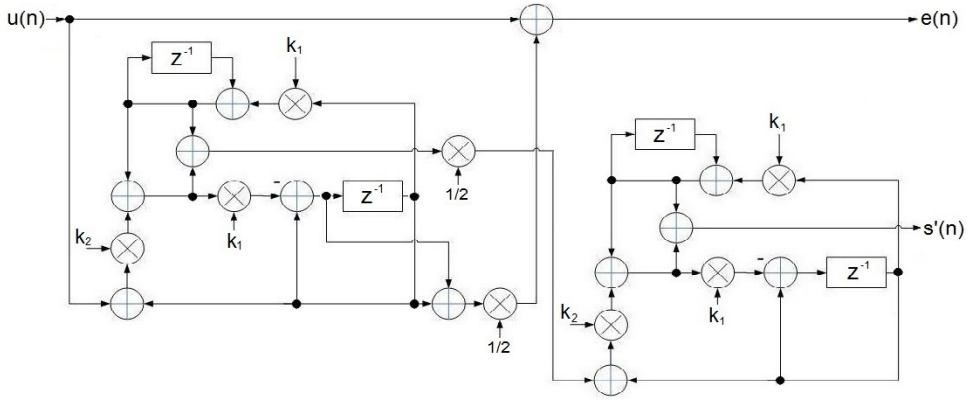


Figure 2.8. Kwan & Martin's direct coefficient scaling structure implemented to track a single sinusoid signal, which is created from two bi-quads; although the second bi-quad may be simplified slightly [4].

From these notch transfer functions, it's clear that the DCS form is more complex to synthesise than the all-pass forms; particularly when cascaded to track multiple sinusoid signals in its output-error form: which is shown in Chapter 3.

2.4 Relevant Complex Adaptive Notch Filter Publications

This section focusses on CANFs, primarily considering the research of Nishimura and Regalia's 2010 publication [23], where Regalia develops his structure into a form capable of tracking CSSs, which is considered first.

2.4.1 Regalia's Complex Adaptive Notch Filter Structure

The paper which is most relevant to this research is Regalia's modified structure [23]: which he has developed to enable his design to track CSSs. This publication provides a strong foundation, and good reference point for comparison in this research.

The structure which Regalia published in 2010 has been included as Figure 2.9 in this thesis. Although, interestingly this structure is included in his 1999 publication [47] as Fig. 6; and further details of variations to this structure are also included within this comprehensive paper.

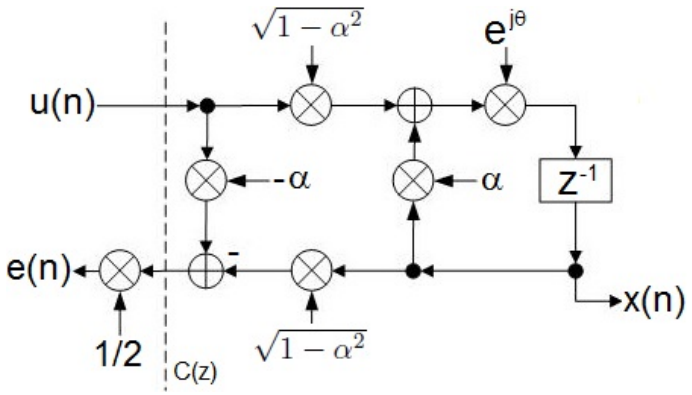


Figure 2.9. Regalia's design developed into a form capable of tracking complex sinusoid signals [23].

As in his earlier publications, Regalia also applies the equation-error approach in [23]; although, he has not included a simulation to demonstrate that his method is capable of tracking multiple CSSs. However, he does include Fig. 5, which claims a gradient descent algorithm fails to track a CSS; although this research disputes this result, as it proves that it is possible to track a CSS with a gradient descent method, which this research refers to as an output-error approach. Chapter 4 of this thesis demonstrates that actually an output-error approach is more robust than an equation-error method; and shows a scenario where Regalia's equation-error approach fails to track two CSSs.

Within [23] Regalia demonstrates that his structure is capable of tracking Complex-Valued Chirp Signal (CVCS) in Fig. 3 of his paper; which is investigated in Chapter 6 of this thesis, wherein the performance of both approaches is compared for tracking a CVCS. However, it should be noted that tracking multiple CVCS is a significant challenge.

2.4.2 Nishimura's Complex Adaptive Notch Filter Structure

Nishimura has published several papers on CANFs; where the first was published in 1996 as [48]. In Nishimura's research he has developed Kwan & Martin's design [4] into a CANF, and his earlier papers show a few variations of the structure he developed, such as [48] and [49]: where the structure from [49] is shown in Figure 2.10; however, his later papers, such as [50]: which was published in 2013, do not include a structure.

Interestingly, Nishimura applies an output-error approach for the tracking of multiple signals, which is common throughout his publications. His 2013 publication [50] also includes a detailed tracking analysis, as do many of his publications.

In [50] Nishimura et al. have referenced Regalia's CANF; therefore, it is surprising that they have not compared the performance of their structure

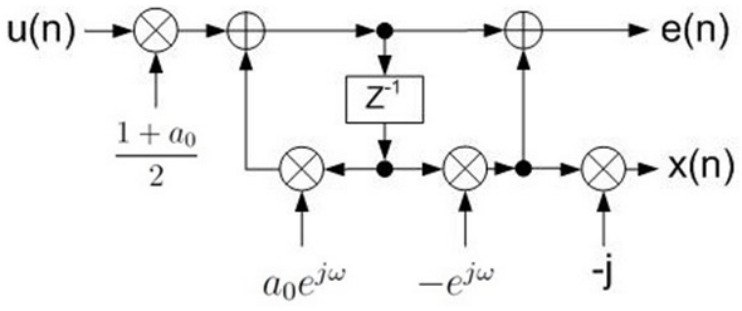


Figure 2.10. Nishimura's complex adaptive notch filter structure from [49].

to his. In his research Nishimura et al. also adapts the notch bandwidth parameter, which is discussed in Chapter 5 of this thesis. Nishimura also considers the tracking of linear chirps in [51], and publishes some analysis of this; which highlights the relevance of Chapter 6 of this thesis; then in [13]: which is his most recent paper that was published in 2014, he exploits CANFs to track the phases of AC components. Therefore in summary, Nishimura's research provides another useful CANF structure, which employs an output-error method for tracking multiple signals; however, it would be interesting to see how its performance compares to Regalia's or the structure proposed in this research: which is a suggestion for future research.

Also observe that Chapter 5 of this thesis investigates adapting the notch bandwidth parameter: which is a topic that Regalia has not actively researched, and the literature relevant on adapting the notch bandwidth parameter is included in the introduction to Chapter 5. Which is also the case for Chapter 7 of this thesis, where this chapter implements a genetic algorithm.

In the final section of this chapter, which completes this review of relevant literature; the evaluation of adaptive schemes used for enhancing sinusoids in noise is considered.

2.5 Adaptive Notch Filter Evaluation Criteria

Primarily, three methods have been used as evaluation criteria for ANFs in this thesis, which are described in the next three parts of this section; these three methods are: bias, variance and Signal-to-Noise-Ratio Improvement Ratio (SNR_IR). However, the SNR has also been calculated in this thesis, by applying the formula

$$SNR(dB) = 10.\log_{10}\left[\frac{S^2}{\sigma^2}\right]; \tag{2.5.1}$$

herein, S is the amplitude of the signal, and σ^2 is the variance of the estimate.

2.5.1 Bias

The first of these bias, can be described as the average distance away from the target value for an estimator. Consider the actual frequency is θ, the estimation of a frequency at one instance is $\hat{\theta}(n) = f(\mathbf{x})$, where $\mathbf{x}$ denotes the observation sequence, then the mean of the estimator $E\{\hat{\theta}(n)\} = m_\theta$; thus the bias of the estimator is given by

$$\Delta\omega = \theta - m_\theta. \tag{2.5.2}$$

In practice, to obtain the mean a sample average is used, which is

$$m_\theta = \frac{1}{N}\sum_{n=1}^{N}(\hat{\theta}(n));$$

therefore, the practical estimate for bias can be shown to be

$$\Delta\omega \approx \theta - \left[\frac{1}{N}\sum_{n=1}^{N}(\hat{\theta}(n))\right]. \tag{2.5.3}$$

In this research, this is calculated from a sufficient number of samples to attain statistical stability; which in practice is achieved by averaging 200

steady-state yielded reliable results.

2.5.2 Variance

The second method variance, represents the average squared deviation of the estimator away from the mean, which may be expressed as

$$\sigma_{\hat{\theta}}^2 = E\{|\hat{\theta} - m_\theta|^2\}; \tag{2.5.4}$$

again this has been estimated for the last 200 values; and in practice the variance is estimated via the following formula

$$\sigma_{\hat{\theta}}^2 \approx \frac{1}{N}\sum_{n=1}^{N}(\hat{\theta} - m_\theta). \tag{2.5.5}$$

Evaluating Complex-Valued Chirp Signals

When tracking CVCSs: which is considered in Chapter 6 of this thesis; CVCSs should be evaluated differently as their frequency is constantly changing; therefore, a CANF may favour certain parts of a CVCS. Thus when evaluating a CVCS, its full range should be considered; and only the first hundred or so values should be omitted i.e. whilst the CANF is locking onto the target signal. Hence by applying this method, a result cannot select the favoured part of a CVCS for one approach, which ensures a fair comparison.

However, if as a consequence of using the complete CVCS, the number of samples was an issue in a particular scenario, a sensible approach would be to select various samples over the CVCSs full range, possibly in steps of a thousand; thus reducing the complexity of this performance estimate. Please also note, that bias cannot be used to evaluate a CVCS, as its frequency is constantly changing.

2.5.3 Signal-to-Noise Ratio Improvement Ratio

The last measurement considered in this thesis is the SNR_IR, which is a purely a theoretical performance estimation; thus cannot be applied to the data. Please note that, this performance estimation has only been applied to the real structures.

To begin with Kwan and Martin's paper [4] provides a description of the SNR_IR, which can be defined as:

$$SNR\_IR = \frac{SNRout}{SNRin};$$

herein, $SNRin$ is the SNR at the input to the filter, and $SNRout$ is the SNR at the output of the filter.

Kwan and Martin's paper contains a table of the SNR_IR for Hush's, Nehorai's and their structure, which is included in this thesis as Table 2.1; wherein r is a given pole radius.

Table 2.1. Signal-to-noise-ratio improvement ratios for real adaptive notch filter structures published in [4].

Paper	Kwan & Martin [4]	Nehorai [46]	Hush [52]
Calculation	$\frac{2}{1-r^2}$	$\frac{1}{1-r}$	$\frac{1+r^2}{1-r^2}$

To complete this section of the review, the SNR_IR is calculated for the four structures primarily considered in this research, and interestingly a published value has not been noted for Regalia's or Cho Choi & Lee's structure. The calculations for Chambers' structure have been included, as this is the design developed in this research; although it has been acknowledged that the SNR_IR for Chambers' structure has been published in [1].

Whilst analysing these results it was noted that the research by Hush et al. [52] showed a similar calculation to Chambers' structure [1].

Now moving on, the SNR_IR is calculated with the following method. As Kwan & Martin state in [4] "The SNR improvement ratio is defined as the SNR at the output of the bandpass filter divided by the SNR of the input signal". Now as Regalia shows in [2]: which Figure 2.6 a) also illustrates, the bandpass and notch transfer functions are in fact very similar; as from the all-pass function $A(z)$, the bandpass transfer function $H_{bp}(z)$ is

$$H_{bp}(z) = \frac{1}{2}[1 - A(z)]; \tag{2.5.6}$$

then for the notch transfer function $H_{notch}(z)$, is almost the same which is

$$H_{notch}(z) = \frac{1}{2}[1 + A(z)]. \tag{2.5.7}$$

Therefore, the bandpass function may be used to derive the SNR_IR for a notch filter starting with the $SNRin$; which assuming that the signal power is equal to one, is

$$SNRin = \frac{1}{\sigma_n^2}; \tag{2.5.8}$$

wherein, σ_n^2 is the variance of the noise. Then the $SNRout$ is defined as

$$SNRout = \frac{|H_{bp}(e^{j2\pi\omega_0})|^2}{\sigma_n^2 \frac{1}{2\pi j} \oint_c |H_{bp}(z)|^2 \frac{dz}{z}}. \tag{2.5.9}$$

Therefore, the SNR_IR for a notch filter can be derived to be

$$SNR\_IR = \frac{SNRout}{SNRin} = \frac{\frac{|H_{bp}(e^{j2\pi\omega_0})|^2}{\sigma_n^2 \frac{1}{2\pi j} \oint_c |H_{bp}(z)|^2 \frac{dz}{z}}}{\frac{1}{\sigma_n^2}}; \tag{2.5.10}$$

which simplifies to

$$SNR\_IR = \frac{|H_{bp}(e^{j2\pi\omega_0})|^2}{\frac{1}{2\pi j}\oint_c |H_{bp}(z)|^2 \frac{dz}{z}}. \tag{2.5.11}$$

Then as Kwan & Martin state in [4], assuming the input consists of a single sinusoid in white noise with power σ_n^2, the ratio can be computed as follows

$$SNR\_IR = \left[\frac{1}{2\pi}\int_{-\pi}^{\pi} H_{bp}(\omega)H_{bp}^*(\omega)d\omega\right]^{-1}. \tag{2.5.12}$$

This simplification occurs because $|H_{bp}(e^{j2\pi\omega_0})|^2 = 1$; which is exploited for Chambers' NFB structure next; then for the other structures compared in this Chapter, and Chapter 3 of this thesis.

The SNR_IR for Chambers' NFB structure

The $|H_{bp}(e^{j2\pi\omega_0})|^2$ for the NFB structure [1] is calculated as follows

$$H_{bp}(e^{j\omega}) = \frac{(1-\alpha)}{2}\frac{(1-e^{j\omega})(1+e^{j\omega})}{1-(1+\alpha)\beta e^{j\omega}+\alpha e^{j2\omega}}, \tag{2.5.13}$$

where $\beta = \cos 2\pi\omega_0 = \cos\omega_0$. Therefore, $|H_{bp}(e^{j\omega})|^2$ is calculated as

$$|H_{bp}(e^{j\omega})|^2 = \frac{(1-\alpha)^2}{4}\frac{(1-e^{j\omega})(1+e^{j\omega})}{(1-(1+\alpha)\beta e^{-j\omega}+\alpha e^{-2j\omega})}.\frac{(1-e^{-j\omega})(1+e^{-j\omega})}{(1-(1+\alpha)\beta e^{j\omega}+\alpha e^{2j\omega})}.$$

Then by: multiplying this equation out, replacing β with $\cos\omega_0$, and applying trigonometric identities: including the relation $\cos\theta \equiv \frac{1}{2}(e^{j\theta}+e^{-j\theta})$, produces the expression $|H_{bp}(e^{j\omega_0})|^2 =$

$$\frac{4(1-2\alpha+\alpha^2)(1-\cos^2\omega_0)}{4(1+\alpha^2+(1+\alpha)^2\cos^2\omega_0-(\alpha+\alpha^2)\cos^2\omega_0+2\alpha\cos 2\omega_0-(1+\alpha)\cos^2\omega_0)};$$

which simplifies to

$$|H_{bp}(e^{j\omega_0})|^2 = \frac{1-2\alpha+\alpha^2-\cos^2\omega_0+2\alpha\cos^2\omega_0-\alpha^2\cos^2\omega_0}{1-2\alpha+\alpha^2-\cos^2\omega_0+2\alpha\cos^2\omega_0-\alpha^2\cos^2\omega_0} = 1.$$

Next, as it has been proven that $|H_{bp}(e^{j\omega})|^2 = 1$. Now the remainder of the SNR_IR is evaluated as follows, starting with

$$|H_{bp}(z)|^2 = \frac{(1-\alpha)^2}{4}\frac{(1+z^{-1})(1-z^{-1})}{1-(1+\alpha)\beta z^{-1}+\alpha z^{-2}}\frac{(1+z)(1-z)}{1-(1+\alpha)\beta z+\alpha z^2}.$$

Therefore, the SNR_IR can be expressed as

$$SNR\_IR = \left[\frac{1}{2\pi j}\oint_c \frac{(1-\alpha)^2}{4}\frac{(z^2-1)(1-z^2)dz}{(z^2-(1+\alpha)\beta z+\alpha)(1-(1+\alpha)\beta z+\alpha z^2)z}\right]^{-1}.$$

To simplify this expression, assuming that $\beta = 0$: as [1] and [4] did, thus $\omega_0 = \frac{\pi}{2}$; which leaves

$$SNR\_IR = \left[\frac{(1-\alpha)^2}{8\pi j}\oint_c \frac{(z^2-1)(1-z^2)dz}{(z^2+\alpha)(1+\alpha z^2)z}\right]^{-1};$$

where the poles are $z = 0$, $j\sqrt{\alpha}$, $-j\sqrt{\alpha}$, $j\frac{1}{\sqrt{\alpha}}$ and $-j\frac{1}{\sqrt{\alpha}}$. However, the last two poles are outside the unit circle, and thus do not contribute to the integral. Then completing the square, and using the 'cover up rule' to extract the residuals gives us

$$SNR\_IR = \left[\frac{(1-\alpha)^2}{8\pi j}2\pi j\Sigma\{Res\}\right]^{-1}$$

Wherein, Res = residuals inside $|z| = 1$, of

$$\frac{(z^2-1)(1-z^2)}{(z-j\sqrt{\alpha})(z+j\sqrt{\alpha})(1-j\sqrt{\alpha}z)(1+j\sqrt{\alpha}z)z}, \text{ which equates to}$$

$$SNR\_IR = \left[\frac{(1-\alpha)^2}{4}\frac{(1+\alpha)+(1+\alpha)}{2\alpha(1-\alpha)}+\frac{-1}{\alpha}\right]^{-1} = \left[\frac{(1-\alpha)^2}{4}\frac{2\alpha}{\alpha(1-\alpha)}\right]^{-1};$$

yielding the final result for the NFB's SNR_IR to be

$$SNR\_IR = \frac{2}{1-\alpha}. \tag{2.5.14}$$

The SNR_IR for Chambers' NFA structure

Now for the Notch Filter A (NFA) structure [1], the bandpass transfer function is

$$H_{bp}(z) = \frac{(1-\alpha)z^{-1}(\beta - z^{-1})}{1-(1+\alpha)\beta z^{-1} + \alpha z^{-2}}. \tag{2.5.15}$$

Then calculating $|H_{bp}(z)|^2$ produces the expression

$$|H_{bp}(z)|^2 = \frac{(1-\alpha)^2(z\beta - 1)(\beta - z)z}{(z^2 - (1+\alpha)\beta z + \alpha)(1-(1+\alpha)\beta z + \alpha z^2)},$$

which allows the SNR_IR to be calculated as

$$SNR\_IR = \left[\frac{(1-\alpha)^2}{2\pi j} \underbrace{\oint_c \frac{(z\beta-1)(\beta-z)dz}{(z^2-(1+\alpha)\beta z+\alpha)(1-(1+\alpha)\beta z+\alpha z^2)}}_{=2\pi j \sum K}\right]^{-1};$$

where K = Residuals of poles inside $|z| = 1$. Now to make this easier, consider $\beta = 0$. Where the poles are $z = j\sqrt{\alpha}$, $-j\sqrt{\alpha}$, $\frac{1}{j\sqrt{\alpha}}$ and $-\frac{1}{j\sqrt{\alpha}}$; as before the last two poles should be omitted; producing the expression

$$= [(1-\alpha)^2 \sum Residuals\_of \frac{z}{(z-j\sqrt{\alpha})(z+j\sqrt{\alpha})(1-j\sqrt{\alpha}z)(1+j\sqrt{\alpha}z)}]^{-1},$$

which equates to

$$SNR\_IR = \left[(1-\alpha)^2\left[\frac{1}{2}\frac{1}{(1+\alpha)(1-\alpha)} + \frac{1}{2}\frac{1}{(1+\alpha)(1-\alpha)}\right]\right]^{-1}.$$

This expression then simplifies to

$$SNR\_IR = \left[\frac{(1-\alpha)^2}{(1+\alpha)(1-\alpha)}\right]^{-1};$$

which rearranges to the final solution for the NFA structure, which is

$$SNR\_IR = \frac{(1+\alpha)}{(1-\alpha)}. \tag{2.5.16}$$

The SNR_IR for Regalia's structure

Firstly, the all-pass function was located from Regalia's 1988 publication [47] as (4.3), which is

$$A(z) = \frac{k_2 + k_1(1+k_2)z^{-1} + z^{-2}}{1 + k_1(1+k_2)z^{-1} + k_2 z^{-2}}. \tag{2.5.17}$$

Then the band-pass function $|H_{bp}(z)|^2 = \frac{1}{2}[1 - A(z)]$ is derived to be

$$|H_{bp}(z)|^2 = \frac{(1 + k_1(1+k_2)z^{-1} + k_2 z^{-2}) - (k_2 + k_1(1+k_1)z^{-1} + z^{-2})}{1 + k_1(1+k_2)z^{-1} + k_2 z^{-2}};$$

which simplifies to

$$|H_{bp}(z)|^2 = \frac{1 - k_2 + k_2 z^{-2} - z^{-2}}{1 + k_1(1+k_2)z^{-1} + k_2 z^{-2}} = \frac{(1-k_2)(1-z^{-2})}{1 + k_1(1+k_2)z^{-1} + k_2 z^{-2}}.$$

Interestingly, this is the same result as was derived for the NFB all-pass function; except $k_2 = r^2$, therefore, the SNR_IR for Regalia's structure is

$$SNR\_IR = \frac{2}{1-r^2}. \tag{2.5.18}$$

The SNR_IR for Cho, Choi & Lee's structure

For this structure, the bandpass transfer function needs to be derived from the notch transfer function; where the notch transfer function is

$F(z) = \dfrac{1 + k_0(1 + k_1)z^{-1} + k_1 z^{-2}}{1 + a_0(1 + a_1)z^{-1} + a_1 z^{-2}}$. Herein, $a_1 = \alpha k_1$ and $k_1 = 1$; therefore,

$$H_{bp}(z) = 1 - F(z) = \frac{(a_0(1 + a_1) - k_0(1 + k_1))z^{-1} + (a_1 - k_1)z^{-2}}{1 + a_0(1 + a_1)z^{-1} + a_1 z^{-2}}. \tag{2.5.19}$$

Next, calculating $|H_{bp}(z)|^2$ produces the equation

$$\begin{aligned} |H_{bp}(z)|^2 = & \frac{(a_0(1 + a_1) - k_0(1 + k_1))z + (a_1 - k_1)}{z^2 + a_0(1 + a_1)z + a_1} \cdot \\ & \frac{(a_0(1 + a_1) - k_0(1 + k_1))z + (a_1 - k_1)z^2}{1 + a_0(1 + a_1)z + a_1 z^2}. \end{aligned} \tag{2.5.20}$$

Then assuming $k_0 = a_0 = 0$, provides a SNR_IR equation of

$$\begin{aligned} SNR\_IR &= \left[\frac{1}{2\pi j}\oint_c \frac{(a_1 - k_1)(a_1 - k_1)z^2 dz}{(z^2 + a_1)(1 + a_1 z^2)z}\right]^{-1} \\ &= \left[\frac{1}{2\pi j}\oint_c \frac{(a_1^2 z - 2a_1 k_1 z + k_1^2 z)dz}{(z^2 + a_1)(1 + a_1 z^2)}\right]^{-1}; \end{aligned} \tag{2.5.21}$$

where the poles are $z = j\sqrt{\alpha}$, $-j\sqrt{\alpha}$, $\frac{1}{j\sqrt{\alpha}}$ and $-\frac{1}{j\sqrt{\alpha}}$. Again omitting the last two poles, then calculating the sum of the residuals of the poles, produces the expression

$$SNR\_IR = \left[\frac{(a_1 - k_1)^2}{2(1 + a_1)(1 - a_1)} + \frac{(a_1 - k_1)^2}{2(1 + a_1)(1 - a_1)}\right]^{-1};$$

which simplifies to the final solution for Cho, Choi & Lee's structure, which is

$$SNR\_IR = \frac{1 - a_1^2}{(a_1 - 1)^2} = \frac{(1 - a_1)(1 + a_1)}{(1 - a_1)(1 - a_1)} = \frac{1 + a_1}{1 - a_1}. \tag{2.5.22}$$

Now to complete this section, Table 2.2 contains the SNR_IR for the structures which have just been evaluated; and Table 2.2 also includes calculated values for the SNR_IR, using $\alpha = 0.9025$: where α is equivalent to r^2, which is the pole radius.

Table 2.2. Signal-to-noise-ratio improvement ratios for the structures evaluated in this research.

Paper	Chambers NFA [1]	Chambers NFB [1]	Regalia [2]	Cho, Choi & Lee [3]	Kwan & Martin [4]
Calculation	$\frac{1+\alpha}{1-\alpha}$	$\frac{2}{1-\alpha}$	$\frac{2}{1-r^2}$	$\frac{1+\alpha}{1-\alpha}$	$\frac{2}{1-r^2}$
SNR_IR	19.51	20.51	10.78	19.51	10.78

It is no surprise that these results are similar, as three of these designs are based around the all-pass transfer function.

2.6 Summary

This chapter may be summarised as follows: The first section provided a historical background on the subject of adaptive filtering; then the second section introduced ANF structures, explaining the difference between DCS and all-pass structures, before moving onto equation-error and output-error approaches: which are required to track multiple sinusoid frequencies. Then four second order structures were introduced, which were due to: Chambers [1], Regalia [2], Cho, Choi & Lee [3], and Kwan & Martin [4].

In the third section the synthesis of these four structures was demonstrated; and when considering the synthesis Chambers' structure appears the simplest; however, it was clear that the direct-coefficient-scaling approach is much more complex than the all-pass solutions.

Then the fourth section discussed CANFs, where Regalia's 2010 publication is reviewed, highlighting its relevance to this research; then Nishimura's

research is summarised. The last section of this chapter, introduced the three evaluation criteria applied in this thesis, which are: bias, variance and SNR improvement ratio. The SNR improvement ratio was calculated for the four structures, where Chambers' NFB structure provided the best result. Please take note that some interesting points are highlighted for papers published between 1991-2013 in Appendix 9.1. of this thesis.

This concludes the relevant literature review; next, Chapter 3 compares the four structures due to: Chambers [1], Regalia [2], Cho, Choi & Lee [3], and Kwan & Martin [4], for tracking up to four RSSs; providing further information on the structures, simulations, and tabular results, thus highlighting the strengths and weaknesses of each structure.

Chapter 3

TRACKING REAL SINUSOID SIGNALS WITH ADAPTIVE NOTCH FILTERS

This chapter considers adaptive notch filtering for tracking up to four Real Sinusoid Signals (RSSs). The chapter is split into two sections, where the first section introduces the IIR Least Mean Square (LMS) algorithm, shows the effect of changing the notch bandwidth parameter, and provides further information on the structures considered in the second section; then, the second section of the chapter compares the structures due to: Chambers [1], Regalia [2], Cho, Choi & Lee [3], and Kwan & Martin [4], for the tracking of up to four RSSs; where these structures were formally introduced in the previous chapter.

3.1 Introduction

The first section of this chapter begins by, introducing the LMS algorithm; then shows different notch bandwidths, and lastly defines a RSS.

3.1.1 The Infinite Impulse Response

Least Mean Square Algorithm

The IIR LMS algorithm is at the centre of the research papers [1] - [4]. Thus, it should be duly noted that, an update equation, such as the LMS algorithm, is certainly the heart to any adaptive design. Therefore, a brief mathematical explanation follows of the IIR LMS algorithm; please also note that [53]: a book titled 'Statistical Digital Signal Processing and Modelling', has provided a useful source of theoretical information on this topic.

To begin with examine Figure 3.1, where the transfer function $H(z)$ for this figure can be expressed in the form

$$H(z) = \frac{b_0 + b_1 z^{-1} + ... + b_q z^{-q}}{1 + a_1 z^{-1} + ... + a_p z^{-p}}; \tag{3.1.1}$$

wherein Figure 3.1, $u(n)$ is the input signal and $y(n)$ is the output signal. Then respectfully for equation (3.1.1), p and q represent the orders of the numerator and denominator respectively; herein, the values p and q are connected to stages c) and d) in the computation that follows.

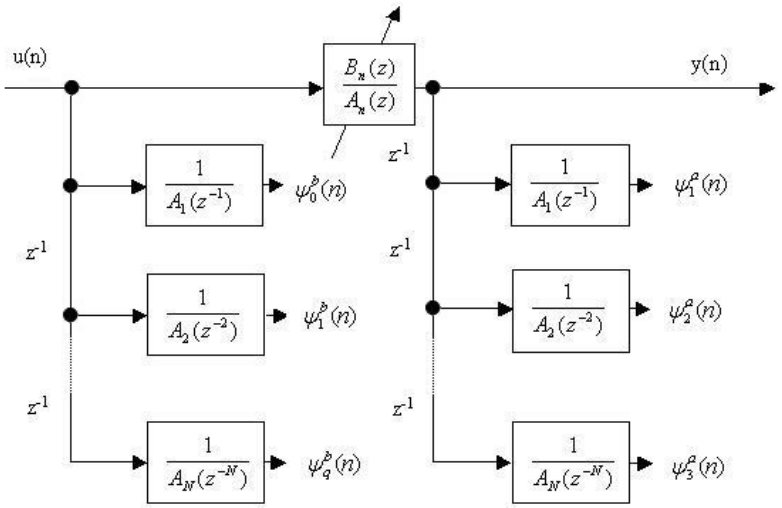


Figure 3.1. A possible structure for the infinite impulse response adaptive least mean square algorithm [53] p539.

Figure 3.1 is updated via the following computation, where the value $e(n)$ equates to the error calculated in the algorithm, and $d(n)$ represents the desired signal. This simplified LMS IIR adaptive filter assumes that the step size or adaptation gain μ, is small enough to form an estimate of the gradient energy recursively; and these estimates $\psi_k^a(n)$ and $\psi_k^b(n)$ are applied to update the adaptive filter coefficients. It should be noted that this structure is initialised with $\mathbf{a}_0 = \mathbf{b}_0 = 0$: as this is the usual practice in a practical implementation; however, other values such as random numbers could be applied.

Computation: For n = 0,1... calculate

a) $y(n) = \mathbf{a}_n^T \mathbf{y}(n-1) + \mathbf{b}_n^T \mathbf{u}(n)$

b) $e(n) = d(n) - y(n)$

c) For k=1,2,...,p

$$\psi_k^a(n) = y(n-k) + \Sigma_{l=1}^{p} a_n(l)\psi_k^a(n-l)$$

$$a_{n+1}(k) = a_n(k) + \mu e(n)\psi_k^a(n)$$

d) For k=0,1,...,q

$$\psi_k^b(n) = u(n-k) + \Sigma_{l=1}^{p} a_n(l)\psi_k^b(n-l)$$

$$b_{n+1}(k) = b_n(k) + \mu e(n)\psi_k^b(n)$$

This computation shows that the IIR LMS algorithm is updating two sets of coefficient values, one for the numerator and the second for the denominator. The two separate updates are shown by the input and output loops in Figure 3.1, and by steps c) and d); however, step c) is connected to step d) by the inclusion of a_n in the calculation $\psi_k^b(n)$.

Please note that this description is for an output-error formulation; and also that the although the low complexity of the LMS is attractive; the weakness of this method is that setting the adaptation gain is key to its success; as if the gain is too high the estimate will oscillate, whilst if the gain is too low the estimate may not converge quickly enough.

Comment on Recursive Adaptive Filtering

Adapting recursive filters are more challenging than non-recursive filters due to the gradients involved. The main challenge in an adaptive filter is to keep the instantaneous poles inside the unit circle, thus avoiding instabilities. This may be simpler for a notch filter as the poles can be kept constant by fixing one parameter. For instance with Chambers' approach, the poles are kept constant by fixing the notch bandwidth parameter α. Once α has been fixed, the second parameter β adapts to track a frequency, which is a positive aspect of working with notch filters i.e. being able to fix one parameter.

3.1.2 The Effect of Changing the Notch Bandwidth Parameter

The effect of applying different values to the notch bandwidth parameter is demonstrated next. Thus, Figure 3.2 shows the effect of applying three different values to the notch bandwidth parameter α, whilst fixing β at 0.5: which are the two parameters in Chambers' structure.

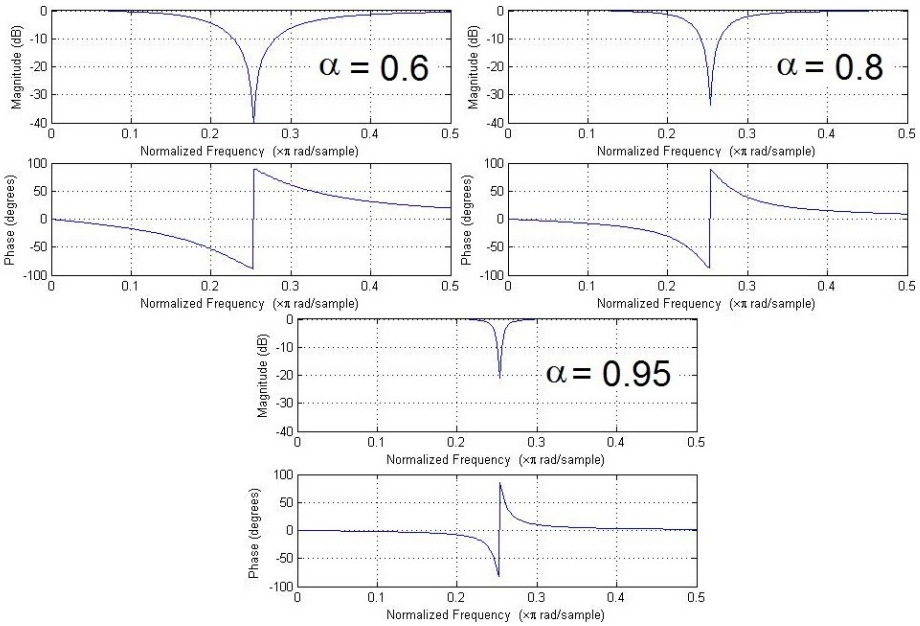


Figure 3.2. Notch filter plots for the notch filter B structure [1], demonstrating different notch filter widths created by changing the notch bandwidth parameter α.

It should be duly noted that, when notches overlap problems occur when cascading structures via the equation-error approach. This notch overlap may be caused by notches being very close together, or by using a lower value of α i.e. 0.6 as shown in Figure 3.2; therefore, the equation-error approach will have the inability to resolve sinusoids that are close together.

3.1.3 Further Analysis Related to the Four Structures

As the previous chapter primarily discussed the synthesis of the structures from the four publications [1] - [4], and calculated the SNR improvement ratio; in this section the additional content of each paper is reviewed in more detail.

Frequency Tracking using Constrained Adaptive Notch Filters Synthesised from All-Pass Sections by Chambers and Constantinides

Chambers' publication [1] contains two notch filter structures which can be created using the all-pass blocks shown in the previous chapter; these structures are referred to as Notch Filter A (NFA) and Notch Filter B (NFB): where the NFB structure has been used for comparison in this chapter, and its synthesis was included in Chapter 2. The NFA structure, shown in Figure 3.3, is slightly more computationally complex than the NFB structure; however, the NFA structure may offer a solutions to potential applications, due to its differing structure: which may produce an advantage in a particular application.

A simulation has been included to demonstrate the different tracking results produced from the NFA and NFB structures, which may be observed as Figure 3.4.

It is clear from observing Figure 3.4, that these two structures create quite different results; as the NFB structure overshoots far more than the NFA structure. This demonstrates that the performance of this all-pass

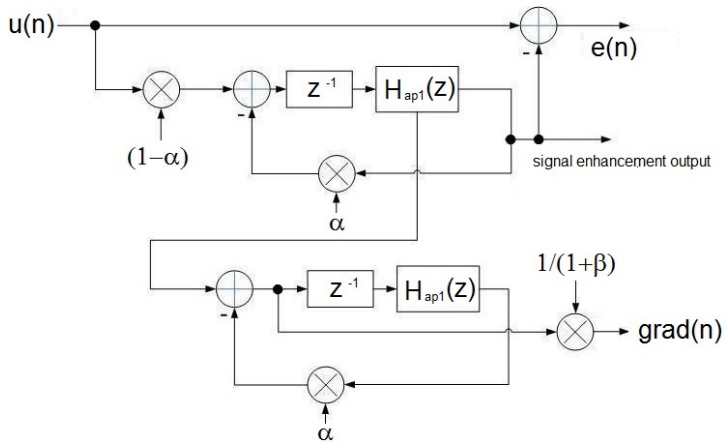


Figure 3.3. Chambers' notch filter A all-pass adaptive notch filter structure from [1].

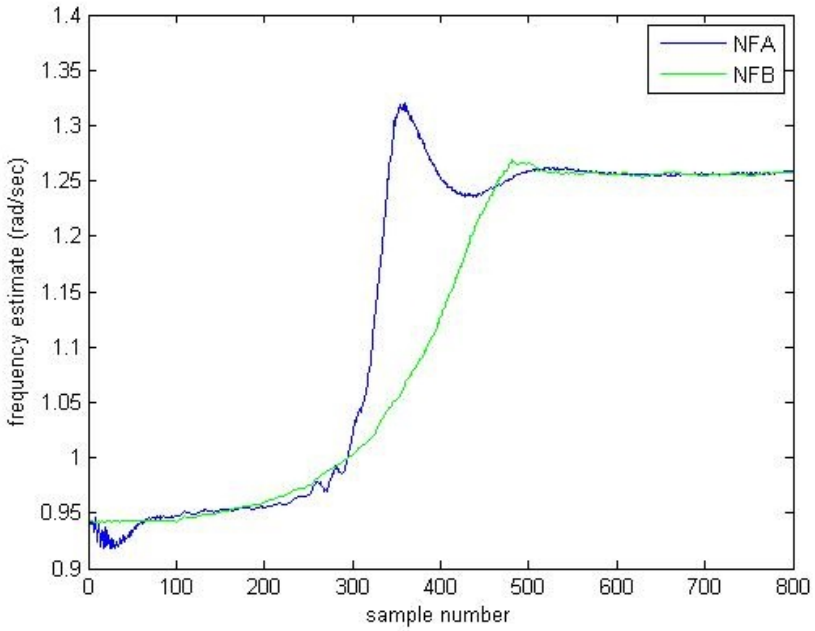


Figure 3.4. Simulation results for tracking one Real Sinusoid Signal with Chambers' notch filter A and notch filter B structures [1].

notch filter structure can be changed significantly, by constructing the notch filter in a different manner; and there are many ways to implement and different types of all-pass structure.

Chambers' paper provides a comprehensive description of the innovative NFA and NFB structures, which includes their SNR improvement ratio. This

publication also comprises of a simulation that compares the NFA and NFB structures to Kwan & Martin's structure [4] for the tracking of a single RSS, then another simulation where the NFB structure is compared to [4] for the tracking of two RSSs; however, the results are not tabulated for either of the simulations: although all the parameters to recreate these simulations are included.

An Improved Lattice-Based Adaptive IIR Notch Filter by Regalia

Regalia's structure from [2] is created from an all-pass filter, this method utilises a COordinate Rotation DIgital Computer (CORDIC) approach and applies a Schur recursion, which is clearly described in [45]: Regalia's research monograph; which provides an excellent foundation to the subject of IIR Adaptive Filtering, as it contains detailed analysis, questions, and examples on many structures and approaches, then suggests areas for further research.

In this publication [2], Regalia applies a stable associated differential equation; which when cascaded to track multiple signals produces an equation-error approach. Regalia's research demonstrates that equation-error methods outperform the output-error variance minimization methods, since the basin of attraction is wider for the instrumental variable methods: for a fixed notch filter bandwidth; however, this wider basin of attraction can be a disadvantage when multiple frequencies are present. Regalia compares his approach to: Cho, Choi & Lee [3], and Kwan & Martin [4], providing a range of simulations; although only includes one table, which is for this particular result; and unfortunately does not consider multiple RSSs in this publication.

Adaptive Line Enhancement by using an IIR Lattice Notch Filter

by Cho, Choi & Lee

Cho, Choi & Lee have published an all-pass design in [3], which is an IIR filter followed by an FIR filter; as if this had been an IIR followed by an FIR, computational complexity would have been an issue. They propose two adaptation methods, where the first method adapts two coefficients for a sinusoid, whilst the second adapts only one coefficient: which is the method evaluated in this chapter. They provide four tables comparing there method to Rao and Kung [54]: as did Kwan & Martin, and Nehorai's approach [46]; however, only show and plot the mean-square-error.

As described in this paper, the effect of changing the λ parameter is shown in Figure 3.5. Herein, observe that by using a value of 0.5 for the λ parameter[1], you achieve a more stable result with faster convergence; this demonstrates that in this paper λ does not directly equate to the adaptation gain as it did in [1] - [2] and [4]; and that in [3] they refer to the forgetting factor as λ not as γ, which has been changed for consistency in this thesis.

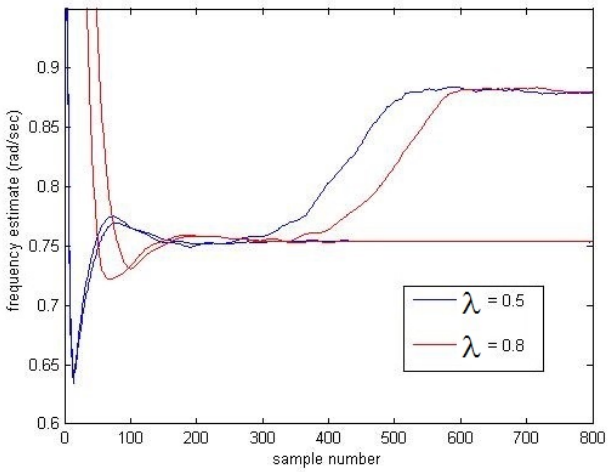


Figure 3.5. The effect of changing the lambda (λ) parameter in Cho, Choi & Lee's structure [3].

[1]Please note, that the preferred value of λ: i.e. 0.5, has been used for the simulations in this chapter.

Overall this is a interesting piece of research, which Regalia regularly cites; and it provides a good point for comparison.

Cho and Lee also published a more comprehensive paper in 1993 [55]; where they primarily evaluated equation-error and output-error responses; although they refer to an output-error approach as a triangular cascade; and the equation-error approach as a linear cascade structure. They also state that the triangular structure performs better than the linear cascade structure, but requires more computational complexity. They declare that second order ANFs provide better results for the detection of multiple sinusoids than the higher-order ANFs, which is useful to know; and that it's the transfer function and algorithm that matter, not the structure in the case of cascading low-order ANFs; and this in an interesting point to be considered in the analysis that follows in the next section. This paper only considers tracking four sinusoid signals which are equally spaced apart, and does not tabularise the results, which would make them much clearer; however, it is an interesting publication which contains a lot of good analysis including a SNR improvement ratio calculation. The structures from [55] are included in Appendix 9.1, as they are not the focus of this publication; although one structure is based on Kwan & Martin's approach.

Adaptive Detection and Enhancement of Multiple Sinusoids Using a Cascade IIR filter by Kwan & Martin

In Kwan & Martin's publication [4], a structure known as a bi-quad is applied to facilitate the tracking of RSSs; this method has been described as Direct Coefficient Scaling (DCS) by Regalia in [23]: which differs from the all-pass structures applied in this thesis. Two bi-quad structures are considered in this paper, the first for constant bandwidth which is the version evaluated in this chapter, and the second for signals with a constant quality factor. This paper also contains comprehensive information on tracking mul-

tiple sinusoids applying the output-error approach, and includes a range of simulations with results embodied into two tables: where one compares the approach to a method by Rao and Kung [54]. Kwan & Martin's paper is well cited, and also provides a thorough description of the SNR improvement ratio. Lastly, as all-pass structures are currently generally considered superior to DCS approaches, this paper provides a good reference for comparison.

3.1.4 The Definition of a Discrete Real Sinusoid Signal

In this fourth and final part of this section, the form of the RSSs tracked in this chapter is defined; therefore, a RSS takes the form

$$u(n) = S\sin(\omega(n)) + W(n); \tag{3.1.2}$$

wherein, the signal fed into the filter input is $u(n)$, S is the amplitude of the signal, $\omega(n)$ is the frequency being tracked, lastly $W(n)$ is the random white noise added to the signal. Please note that ω may be normalised, by multiplying with a factor of 2π.

This definition completes this introduction, which comprised of further information relevant to the tracking of RSSs; leading onto the next section of this chapter, which compares the four structures: firstly for the tracking of up to four RSSs, then secondly by the number of computations required by each structure.

3.2 A Comparison of Four Designs for Tracking up to Four Real Sinusoid Signals

Over the last three decades various structures have been proposed to perform digital adaptive notch filtering; however, a critical comparison of the key important approaches has not been performed for tracking multiple RSSs.

Hence, as a structure may excel under certain conditions, it may however be weak in other scenarios.

Historically, all-pass variations have been developed more recently than DCS; and generally all-pass designs are computationally simpler and usually deliver superior tracking performance, when compared to structures based on DCS: so naturally have been the preferred choice.

An analysis of four adaptive notch filters is undertaken to assess their abilities to track between two and four RSSs. These four structures due to: Chambers [1], Regalia [2], Cho, Choi & Lee [3], and Kwan & Martin [4], all utilise the Normalised Least Mean Square (NLMS) type of learning algorithm, thereby permitting a fair comparative study. This analysis investigates the differences in performance between output-error and equation-error approaches, and also compares a DCS method to three all-pass decompositions [1] - [3].

In this section the results obtained from tracking multiple RSSs are presented. These results are shown in Figure 3.7 - Figure 3.10, and Table 3.1 - Table 3.4, where the first two sets of results are for tracking two sinusoids, the third for tracking three sinusoids, and the fourth for tracking four sinusoids. Please note that all the results in this chapter have been obtained using an ensemble average of ten, which has been found sufficient to obtain statistically stable results. The frequencies tracked were randomly selected and a further analysis of these frequencies is included in Section 3.2.6, which is in the later part of this chapter.

3.2.1 Tracking a Single Real Sinusoid Signal

The four structures have been implemented initially to track a single RSS, producing similar results as shown in Figure 3.6. The tracking performance of each structure is recorded in the following colours: [1] is presented in blue, [2] is shown as red, [3] is displayed as black, and [4] is shown in green.

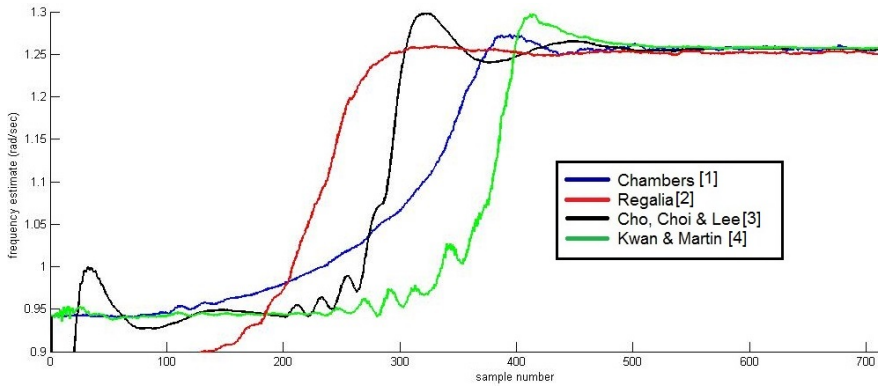


Figure 3.6. Tracking a single real sinusoid signal with approaches [1] - [4].

All the structures were initialised with the notch frequency at 0.9425 radians, which is equivalent to 0.15 in the range -0.5 to 0.5, and the conversion being multiplication by 2π. The frequency being tracked was also set to 0.9425 radians for the first 100 samples; which then hopped to 1.2566 radians at sample 101, thus the tracking of this change can be observed over the remaining samples. The SNR applied to the signals in Figure 3.6 was 10 dB. The NLMS adaptation parameters set for each algorithm were: μ [1] = 0.001, γ [1] = 0.99, μ [2] = 0.005, λ [2] = 0.97, λ [3] = 0.05, γ [3] = 0.9, and μ [4] = 0.01, γ [4] = 0.97.

After these preliminary results, which show similar performance; structures [1] - [4] were implemented and compared against each other for tracking multiple RSSs, and up to four RSSs were tracked.

It should be noted that, from [1] the NFB structure was selected for this evaluation; and from [4], the constant bandwidth version has been used throughout these results. The reason for this decision is that the NFA structure is more complex in terms of computations, therefore utilising this approach would be less attractive, particularly for an output-error approach; as inherently an output-error approach is more complex than an equation-error solution.

To facilitate consistency throughout the results, the value 0.9025 has

been used for the term α: which is the pole radius parameter. Also note, that the performance of each algorithm was optimised empirically through the adaptation gains (μ), whilst ensuring convergence to a satisfactory solution.

3.2.2 Tracking Two Real Sinusoid Signals

The results in Table 3.1 correspond to the tracking of two RSSs, which are initialised at 0.6283 and 0.6912 radians in the ANFs, with target frequencies of 0.8796 - ω_1 and 10.7540 - ω_2. These signals have amplitudes of 1 and 0.01, and the variance of the noise added to the signals was 0.01; which corresponds to SNR of 17 and -3 dB. The NLMS adaptation parameters set for each algorithm were: μ [1] = 0.000125, γ [1] = 0.99, μ [2] = 0.001, λ [2] = 0.97, λ [3] = 0.05, γ [3] = 0.98, and μ [4] = 0.005, γ [4] = 0.9.

Table 3.1. Results for tracking two real sinusoid signals a); wherein $\Delta\omega_1$ & $\Delta\omega_2$ are sample biases and σ_1^2 & σ_2^2 are the sample variances; and a † labels a value smaller than 0.00005.

	Chambers [1]	Regalia [2]	Cho, Choi & Lee [3]	Kwan & Martin [4]
$\Delta\omega_1$	0.0011	0.0018	-0.0005	0.0002
$\Delta\omega_2$	0.000†	0.0002	0.0001	0.000†
σ_1^2	3.556×10^{-4}	2.743×10^{-3}	1.034×10^{-3}	3.426×10^{-4}
σ_2^2	2.491×10^{-5}	1.5532×10^{-4}	3.224×10^{-4}	2.6989×10^{-5}

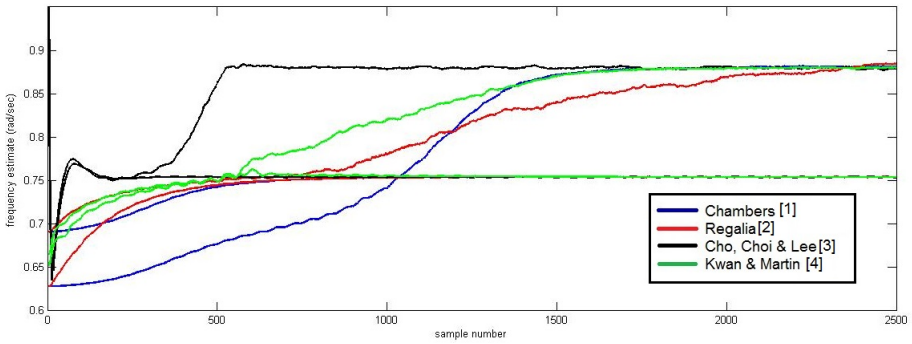


Figure 3.7. Tracking results for two real sinusoid signals corresponding to the results in Table 3.1.

In Figure 3.7 observe that Regalia's method [2] does not fully converge

to the solution, and if the sample range is extended the value converges to approximately 0.9; however, this value does not fully stabilise.

Table 3.2 provides results for the the tracking of two RSSs, which are initialised at 1.2566 and 1.3195 in the ANFs, with target frequencies of 2.1991 - ω_1 and 1.1310 - ω_2. These signals have amplitudes of 1.02 and 1.24, and the amount of noise added was 0.01; which corresponds to SNR of 17.2 and 18.9 dB. The NLMS adaptation parameters set for each algorithm were: μ [1] = 0.000225, γ [1] = 0.9, μ [2] = 0.001, λ [2] = 0.97, λ [3] = 0.05, γ [3] = 0.98, and μ [4] = 0.006, γ [4] = 0.9.

Table 3.2. Results for tracking two real sinusoid signals b); wherein $\Delta\omega_1$ & $\Delta\omega_2$ are sample biases and σ_1^2 & σ_2^2 are the sample variances; and a † labels a value smaller than 0.00005.

	Chambers [1]	Regalia [2]	Cho, Choi & Lee [3]	Kwan & Martin [4]
$\Delta\omega_1$	0.000†	0.0015	0.0004	0.000†
$\Delta\omega_2$	0.000†	-0.0035	0.000†	0.0003
σ_1^2	6.648×10^{-5}	1.451×10^{-4}	1.798×10^{-4}	3.857×10^{-5}
σ_2^2	4.778×10^{-5}	9.839×10^{-4}	9.359×10^{-5}	1.006×10^{-4}

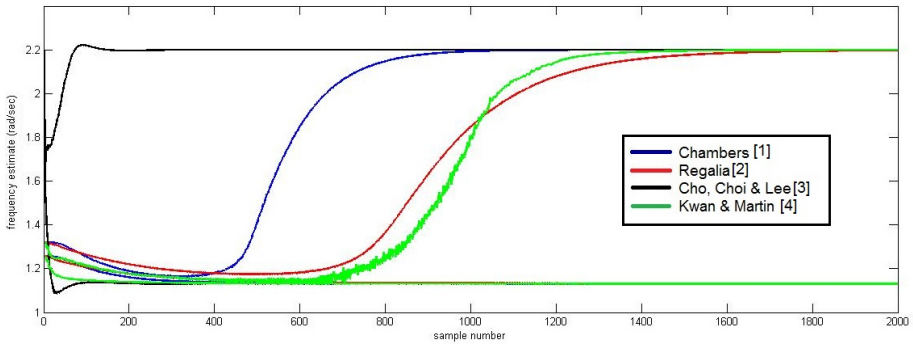


Figure 3.8. Tracking results for two real sinusoid signals corresponding to the results in Table 3.2.

When working in the range -0.5 to 0.5, the values 0.6283 and 0.6912 radians, are equivalent to 0.1 and 0.11.

Therefore, these results show similar performance for all structures with these parameters. Analysing the results further shows that Cho, Choi &

Lee's structure converges fastest: at around 200 samples, with Chambers' solution providing the second quickest convergence: at around 1100 samples; however, Chambers' structure also achieves the least variance in the final solution.

3.2.3 Tracking Three Real Sinusoid Signals

The results collated in Table 3.3, are for tracking three RSSs that were initialised to: 1.131, 1.382 - ω_2 and 1.257 - ω_3, which correspond to the target frequencies, where the frequency at 1.131 instantaneously hops to 1.508 - ω_1 at sample number 1000. These signals amplitudes are: 1.24, 1.00 and 1.02, then noise was added to yield SNRs of 4.1, 2.2 and 2.4 dB. The

Table 3.3. Results for tracking three real sinusoid signals; wherein $\Delta\omega_1$–$\Delta\omega_3$ are the sample biases, and σ_1^2–σ_3^2 are the sample variances; and a † labels a value smaller than 0.00005.

	Chambers [1]	Regalia [2]	Cho, Choi & Lee [3]	Kwan & Martin [4]
$\Delta\omega_1$	0.000†	0.020	-0.001	0.000†
$\Delta\omega_2$	0.000†	0.009	0.008	0.000†
$\Delta\omega_3$	-0.001	0.005	0.011	1.2564
σ_1^2	1.72×10^{-4}	2.43×10^{-3}	9.66×10^{-4}	8.93×10^{-5}
σ_2^2	2.01×10^{-4}	1.43×10^{-3}	3.82×10^{-3}	1.29×10^{-4}
σ_3^2	1.28×10^{-4}	1.84×10^{-3}	3.21×10^{-3}	1.48×10^{-4}

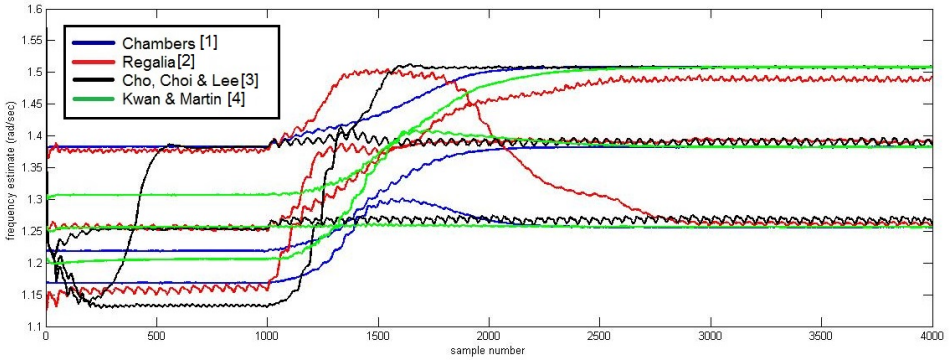


Figure 3.9. Tracking results for three real sinusoid signals corresponding to the results in Table 3.3.

NLMS adaptation parameters set for each algorithm were: μ [1] = 0.00015,

γ [1] = 0.99, μ [2] = 0.003, λ [2] = 0.97, λ [3] = 0.05, γ [3] = 0.98, and μ [4] = 0.004, γ [4] = 0.9.

The results from tracking three RSSs show that, Kwan & Martin's structure converges to the most accurate solution, with Chambers' solution being a close second; however, Chambers' solution converges more quickly than Kwan & Martin's structure. Again, Cho, Choi & Lee's solution converges very quickly; although shows the most variance in its solution. Figure 3.9 also shows that oscillation in the tracking is appearing, this is due to the fact that the adaptation gains have not been reduced enough for this scenario, and this is discussed further in Section 3.2.5.

3.2.4 Tracking Four Real Sinusoid Signals

The results in Table 3.4 correspond to the tracking of four RSSs; which are initialised at: 2.8274 - ω_1, 2.5761 - ω_2, 1.2566 and 0.7540 - ω_3; however, one target frequency 1.2566 instantaneously changes to 0.6283 - ω_4 at 1000 iterations. These signals amplitudes are: 1.24, 1.02, 1.00 and 0.5, where the noise added was 0.3; which corresponds to SNR of 4.1, 2.4, 2.2 and -3.8 dB. The NLMS adaptation parameters set for each algorithm were: μ [1] = 0.000025, γ [1] = 0.99, μ [2] = 0.001, λ [2] = 0.97, λ [3] = 0.05, γ [3] = 0.99, and μ [4] = 0.005, γ [4] = 0.9.

With the implementation applied in this research, when tracking four RSSs, all the algorithms except Chambers' showed some instability i.e. they did not converge to the solution. With Kwan & Martin's structure being the most unstable: requiring twenty-one simulations to obtain ten results; whilst for Cho, Choi & Lee's structure, it was necessary to simulate twelve times.

Table 3.4. Results for tracking four real sinusoid signals; wherein $\Delta\omega_1$–$\Delta\omega_4$ are the sample biases, and σ_1^2–σ_4^2 are the sample variances; and a † labels a value smaller than 0.00005.

	Chambers [1]	Regalia [2]	Cho, Choi & Lee [3]	Kwan & Martin [4]
$\Delta\omega_1$	0.0002	-0.0023	-0.0001	0.000†
$\Delta\omega_2$	-0.0025	-0.0052	0.000†	0.0003
$\Delta\omega_3$	-0.0001	-0.0102	-0.0004	0.0002
$\Delta\omega_4$	0.002	0.0125	0.0036	-0.0001
σ_1^2	2.0478×10^{-5}	3.4400×10^{-4}	0.0012	1.5172×10^{-4}
σ_2^2	3.9586×10^{-5}	8.8078×10^{-5}	0.0010	4.5953×10^{-4}
σ_3^2	5.7369×10^{-5}	7.8236×10^{-4}	9.7064×10^{-4}	1.6630×10^{-4}
σ_4^2	3.9586×10^{-4}	7.8752×10^{-4}	0.0031	1.2033×10^{-4}

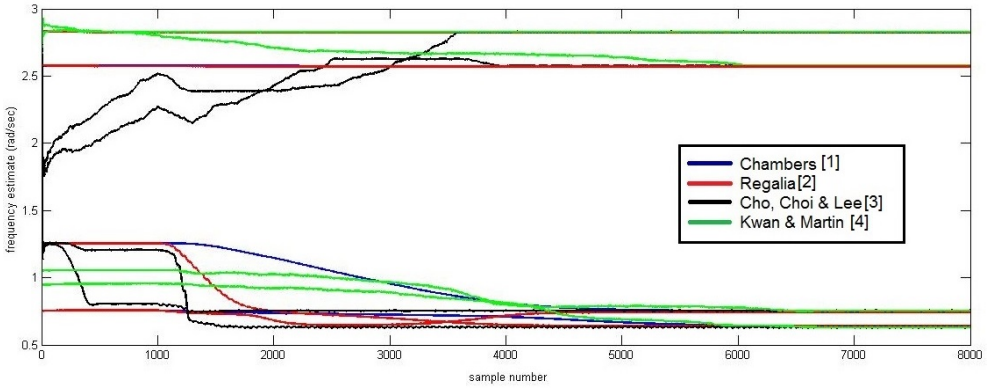


Figure 3.10. Tracking results for four real sinusoid signals corresponding to the results in Table 3.4.

3.2.5 The Normalised Least Mean Square Parameters Applied in this Chapter

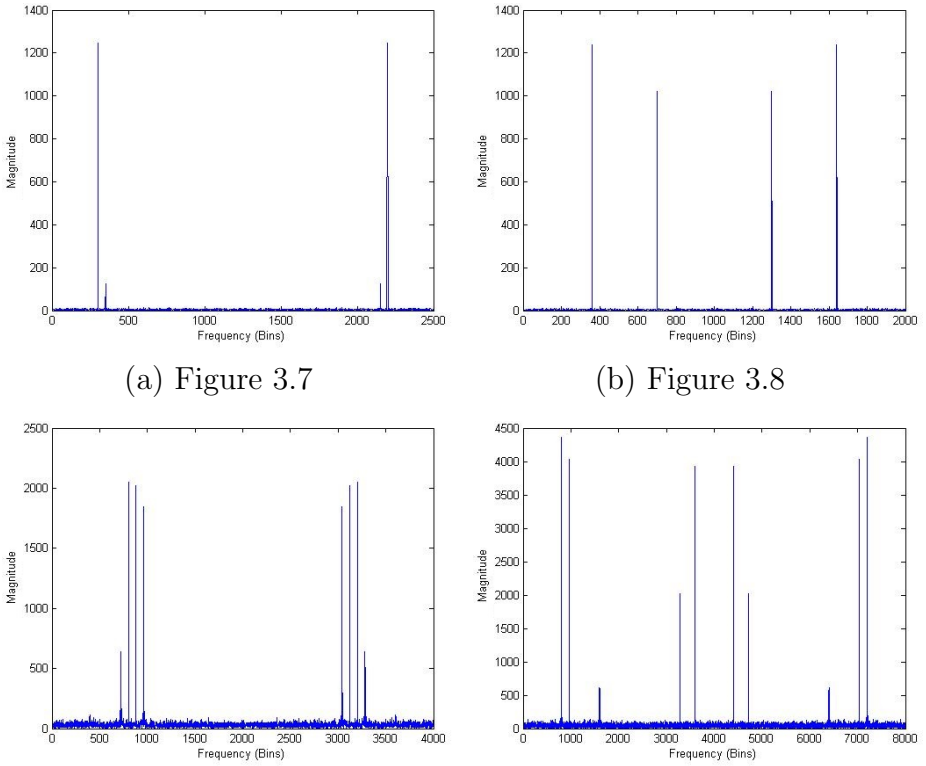
In this analysis, it became clear that as the number of RSS tracked increased, the value of the adaptation gain needed to be reduced; otherwise, oscillations appear as can be observed in Figure 3.9. To demonstrate this point a table of the adaptation gains used in this chapter is included as Table 3.5.

Table 3.5. The Normalised Least Mean Square parameters applied, for the results in this chapter. Wherein, the values used in 1 RSS is Figure 3.6, 2 RSSs a) is Figure 3.7, 2 RSSs b) is Figure 3.8, 3 RSSs is Figure 3.9, and 4 RSSs is Figure 3.10.

	Chambers [1]		Regalia [2]		Cho, Choi & Lee [3]		Kwan Martin [4]	
	μ	γ	μ	γ	λ	γ	μ	γ
1 RSS	0.001	0.99	0.005	0.97	0.5	0.9	0.01	0.97
2 RSSs a)	0.000125	0.99	0.001	0.97	0.5	0.98	0.005	0.9
2 RSSs b)	0.000225	0.9	0.001	0.97	0.5	0.98	0.006	0.9
3 RSSs	0.00015	0.99	0.003	0.97	0.5	0.98	0.004	0.9
4 RSSs	0.000025	0.99	0.001	0.97	0.5	0.99	0.005	0.9

3.2.6 Analysis of the Frequencies selected in this Chapter

The frequencies tracked in the simulations in this chapter were randomly selected, and a further analysis of these frequencies is shown below.



(a) Figure 3.7 (b) Figure 3.8

(c) Figure 3.9 (d) Figure 3.10

Figure 3.11. Frequency analysis of the signals tracked in this chapter.

These results are included for completeness, to show any relationship in

the selected frequencies: which may be useful in any further analysis.

3.2.7 The Computational Complexity of the Four Structures

In this section, the computational complexity of the four structures is considered for a direct Matlab implementation, which is embodied in Table 3.6; providing another suitable method of comparison.

Table 3.6. The complete complexity of the four real structures required at one time sample, whilst tracking one and two real sinusoid signals.

Complexity of the four structures whilst tracking a single real sinusoid signal			
Method	$\div s$	$\times s$	+s
Chambers [1]	2	9	9
Regalia [2]	1	15	10
Cho, Choi & Lee [3]	1	15	17
Kwan & Martin [4]	3	14	12
Complexity of the four structures whilst tracking two real sinusoid signals			
Method	$\div s$	$\times s$	+s
Chambers [1]	4	20	24
Regalia [2]	2	27	19
Cho, Choi & Lee [3]	2	30	32
Kwan & Martin [4]	6	36	27

Table 3.6 clarifies that Kwan & Martin's solution is indeed the most complex, and also confirms that the equation-error approaches simplify the computational complexity, when tracking multiple sinusoids. It should be noted that particularly for Field Programmable Gate Array (FPGA) implementations, a divide is more complex to implement than a multiplication; and that these designs have been synthesised in Matlab using floating point implementation.

3.3 Discussions on the Tracking of Multiple Real Sinusoid Signals

The remarks provided on the overall performance of each structure are as follows. Chambers' structure [1] provides a high performance, robust solution, that converges well in most simulations; although, as this is an output-error approach; it is more computationally complex than the equation-error solutions.

Regalia's structure [2] provides a strong solution; although, this structure struggles to track multiple RSSs in certain conditions: such as RSSs being in close proximity. Cho, Choi & Lee's structure [3] provides a very powerful solution; however, initially this is quite unstable during convergence, and also shows the most variance in the final solution; therefore, this solution would not suit certain scenarios.

Kwan & Martin's structure [4] is generally outperformed by the all-pass solutions, the solution is also quite computationally complex, particularly as this is an output-error approach.

These simulations also show that each all-pass structure has some superior properties to the others; therefore, the most appropriate structure should be carefully selected for any application. The computational complexity for these structures in tracking one and two RSSs has been included, which confirms that Kwan & Martin's structure is more computationally complex than the all-pass structures, and that equation-error approaches do certainly reduce the overall complexity of a system that tracks multiple RSSs.

3.4 Summary

This chapters' first section introduced the IIR LMS algorithm, next it showed different notch bandwidths, then it reviewed the four papers due to: Chambers [1], Regalia [2], Cho, Choi & Lee [3], and Kwan & Martin [4], and lastly

defined a RSS.

Then the chapters' second section comprised of simulations and analysis relevant to these four papers: which were also introduced in the literature review. These simulations and analysis provide a comparison of these four structures for tracking up to four RSSs, thus highlighting the strengths and weaknesses of each approach; and lastly this chapter included the computational complexity of each approach, and frequency plots of the signals tracked in this chapter.

Therefore, after providing a thorough evaluation of these structures capable of tracking RSSs, in the next chapter CSSs are considered. The first part of the chapter describes a CSS, the next section then provides a summary of Regalia's recent publication [23]. The final section develops the new structure proposed from this research, and concludes by comparing its performance to Regalia's scheme.

Chapter 4

COMPLEX ADAPTIVE NOTCH FILTER DEVELOPMENT

4.1 Introduction

This chapter considers tracking complex sinusoids, whilst the previous chapter considered real sinusoids. The chapter begins by defining a Complex Sinusoid Signal (CSS), then introduces Complex Adaptive Notch Filters (CANFs); in the next section Regalia's 2010 publication [23] is discussed. Then the third section develops Chambers' structure [1] into an equivalent complex form, and considers tracking two CSSs. Next, the new structure is compared to Regalia's approach; and lastly the results are discussed, then the chapter is summarised.

A CSS has a real and imaginary component, as is generally the case in complex signals. The complex LMS algorithm was initially considered around 1975, when the first publication on this topic was written by Widrow as [41]; this is a short two page paper, which considers an adaptive linear combiner, showing a structure and the algorithm.

Papers in this field appear to have been steadily published over time, with a further publications in 1986 [42] by Shynk: where he generalizes the Gauss-

Newton algorithm for adaptive IIR filters to include complex coefficients, in another short publication, again showing a structure and the algorithm.

In 1994 Pei and Tseng published a more comprehensive paper as [43], which is clearly an IIR CANF that is built from first-order structures; however, this requires two structures to track a single CSS: which they refer to as a chirp signal. Pei and Tseng highlight that their approach can be applied to suppress narrowband interference in Quadrature Phase Shift Keying (QPSK) spread-spectrum communication systems; and show several simulations: which the earlier publications have not included, one of which demonstrates that changing the notch bandwidth parameter clearly alters the magnitude response in this approach. Pei and Tseng's paper [43] clearly shows the concept of a CANF, although this idea has now been refined more, most significantly by removing the second CANF structure, and also by utilising all-pass components.

The paper which is most relevant to this chapter was published in 2010 by Regalia [23], which provides an excellent benchmark for comparison; please note that this paper has been reviewed in Chapter 2 of this thesis, although the simulations are included in the next section.

4.1.1 The Definition of a Discrete Complex Sinusoid Signal

Thus to begin, a CSS is defined; in [23] Regalia describes that tracking a single frequency ω_0, for a CSS of the form

$$u(n) = Se^{j(\omega_0 n+\phi)} + W(n), \tag{4.1.1}$$

can be achieved: which is the form considered in this chapter. Within this equation S is a scale factor that defines the amplitude of the signal, and ϕ is a random phase uniformly distributed over $(0, 2\pi)$, the final term $W(n)$ is zero mean unity variance complex white noise.

4.2 A Complex Adaptive Notch Filter by Regalia

Now, Regalia's recent 2010 publication [23] is described; and in this paper, he modified the lattice structure from his earlier research [2], so it has the ability to track CSS; this modified structure is shown in Figure 4.1.

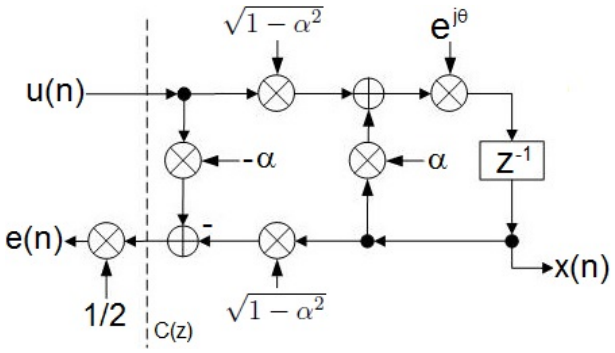


Figure 4.1. Regalia's modified structure capable of tracking complex sinusoid signals [23].

Now the mathematics of Regalia's paper is described, and the update is achieved as follows

$$x(n) = Im[e(n)grad(n)]; \tag{4.2.1}$$

wherein, $x(n)$ is an intermediate term, and Im is the imaginary part of a CSS. This intermediate term $x(n)$ is then used to update ψ as

$$\psi(n) = \gamma\psi(n-1) + (1-\gamma)Re[x(n)^*x(n)]; \tag{4.2.2}$$

herein, Re is the real part of a complex signal and $(.)^*$ is the complex conjugate of the term $x(n)$. The terms $x(n)$ and ψ then update theta as follows

$$\theta(n) = \theta(n-1) + \mu Im\left[\frac{x(n)}{\psi(n)}\right]. \tag{4.2.3}$$

When implementing this approach, it was also discovered that a phase shift of 2π is necessary, when θ becomes greater or less than $\pm\pi$. This can be observed in Figure 4.2, when the frequency hops at 1000 and 2000 iterations, by the estimated signal jumping over the scale. It should be noted that $\theta(n)$, is the parameter which is plotted to produce the simulation results.

Next, the results that Regalia published demonstrating his structures ability to track a frequency hopping CSS is shown in Figure 4.2.

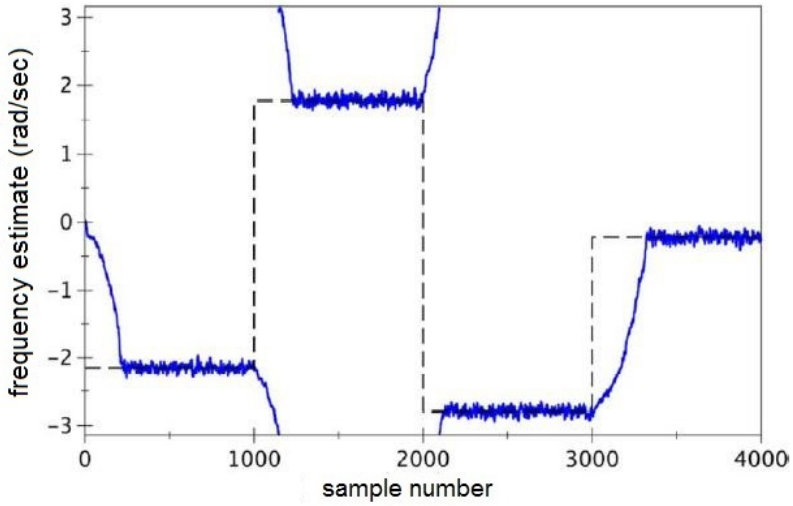


Figure 4.2. Regalia's published result for tracking a hopping single complex sinusoid signal [23].

Interestingly, Regalia claims that a gradient descent algorithm fails to track a CSS where his structure succeeds, and this claim is shown in Figure 4.3.

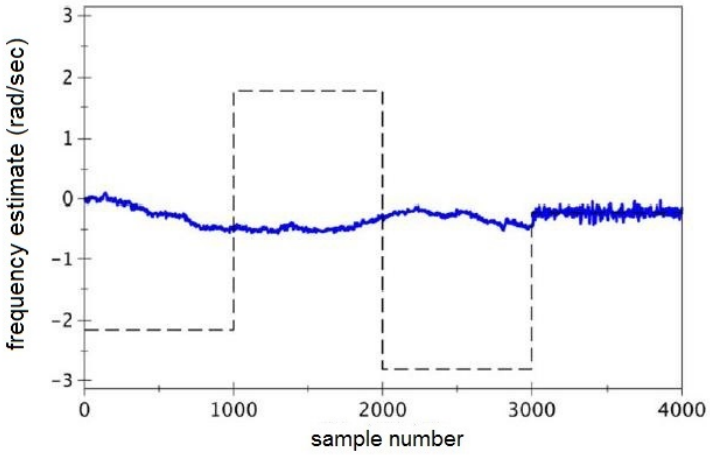


Figure 4.3. Regalia's frequency hop experiment using a gradient descent algorithm, instead of his proposed scheme [23].

Finally in this paper, it should be noted that Regalia compares his scheme to a gradient descent approach, this result is shown in Figure 4.4.

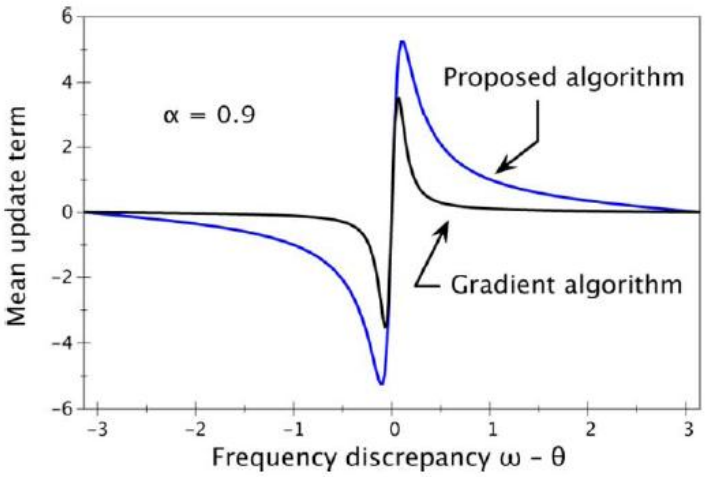


Figure 4.4. Regalia's comparison of his scheme against a gradient descent approach [23]; which compares the mean driving terms of his proposed method with a gradient descent algorithm, when normalised for the same local convergence properties.

Therefore, as Regalia has created an equation-error approach; an original output-error structure is developed, this is described in the next section: which is a gradient algorithm.

4.3 Original Complex Adaptive Notch Filter Development

In this section, Chambers' structure from [1] is developed into a form capable of tracking CSSs, as few of these structures have been modified to enable the tracking of CSSs. Many systems are now commonly functioning with CSSs particularly in communications e.g. In-phase and Quadrature (I&Q) sampling in sensors or in modulators / demodulators.

This design has the advantage that, being developed from an all-pass design based on a structurally lossless prototype [1], which is canonic in the number of multipliers and delay elements; it can also be easily expanded to facilitate the tracking of multiple CSSs. The output-error learning algorithm has also been developed, which facilitates tracking of CSSs that have similar frequencies. Tracking similar frequencies is not possible in the recent work of Regalia shown in [23], as demonstrated in this chapter.

It should be noted that in this research, as in [23], that the results for tracking CSS have been achieved with a relatively low SNR of 0 dB.

4.3.1 Filter Realisation

Initially, a standard first order real all-pass filter is considered, whose z-domain transfer function is

$$H_{ap}(z) = \frac{z^{-1} - \alpha}{1 - \alpha z^{-1}}, \tag{4.3.1}$$

where $0 \ll \alpha < 1$ and is a real coefficient. The all-pass structure is modified to the following form, which can be derived as

$$H_{ap}(z) = \frac{z^{-1}\beta - \alpha}{1 - \alpha z^{-1}\beta}; \tag{4.3.2}$$

wherein, it should observed that an additional phase parameter β has been introduced. This value equates to $\beta = e^{j\theta}$, where θ is the complex phase shift

angle which is equivalent to the frequency being tracked. Then Figure 4.5 shows the notch structure created from utilising this complex all-pass filter. The derivation of the z-domain transfer function for the proposed complex notch filter shown in Figure 4.5 is

$$H_{notch}(z) = \frac{E(z)}{U(z)} = \frac{1}{2}\{1 + A(z)\} = \frac{1}{2}\frac{(1+\alpha)(1-z^{-1}\beta)}{1-\alpha z^{-1}\beta}. \tag{4.3.3}$$

The parameter α controls the notch width of the filter in this equation, and α is fixed during the learning process for stability reasons. Whilst on the other hand, the parameter β controls the notch frequency, and is adapted during learning.

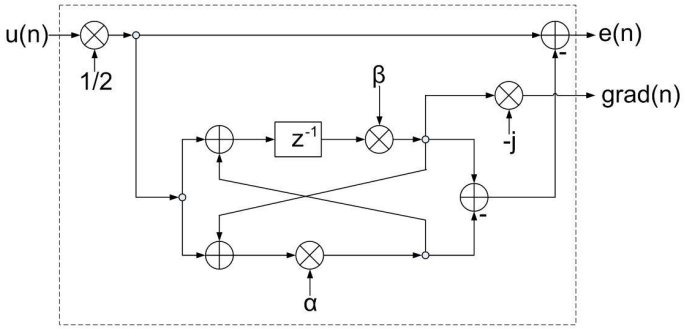


Figure 4.5. The proposed complex adaptive notch filter structure, where the structure within the box defined by the dashed line is denoted CNF_ in Figure 4.6.

The input to the gradient output of the z-domain transfer function, that is required later in developing the learning algorithm, is given as

$$\frac{GRAD(z)}{U(z)} = \frac{1}{2}\frac{(1+\alpha)(-jz^{-1}\beta)}{1-\alpha z^{-1}\beta}. \tag{4.3.4}$$

In the next section of this chapter, the learning algorithm to control β: or equivalently θ is developed.

4.3.2 Learning Algorithm Development

An output-error based learning algorithm is derived, where the cost function (J) is defined as

$$J = |e(n)|^2 = e(n)e^*(n), \tag{4.3.5}$$

herein, $|.|$ and $(.)^*$ denote the modulus and conjugate of a complex number. An update for the unknown parameter θ can be derived from J via the LMS type of algorithm. The update equation derivation begins as

$$\theta(n) = \theta(n-1) - \frac{\mu}{2} \bigtriangledown J|_{\theta=\theta_{n-1}}, \tag{4.3.6}$$

wherein the adaptation gain is μ. By applying differentiation by parts to (4.3.5) gives

$$\bigtriangledown J = e(n) \bigtriangledown e^*(n) + \bigtriangledown e(n)e^*(n), \tag{4.3.7}$$

thus (4.3.6) becomes

$$\theta(n) = \theta(n-1) - \mu Re(e^*(n) \bigtriangledown \ e(n)), \tag{4.3.8}$$

herein the signal $\bigtriangledown e(n) = grad(n)$ in Figure 4.6. The non-quadratic nature of the cost function J in (4.3.5) with respect to θ in $\beta = e^{j\theta}$, requires the adoption of a normalised LMS type update for θ.

The NLMS algorithm includes a recursive calculation of the gradient energy ψ. This calculation is

$$\psi(n) = \psi(n-1)\gamma + (1-\gamma)(grad(n).grad^*(n)), \tag{4.3.9}$$

wherein the γ term is the forgetting factor: which is a value between zero and one. The parameter γ is chosen to be approximately ≥ 0.9 for most

results, and the initialising value for ψ_0 is 1.0. Updating (4.3.8) with (4.3.9) then yields the final update equation for the updated θ to be

$$\theta(n) = \theta(n-1) - \mu Re\left[e^*(n)\frac{grad(n)}{\psi(n)}\right]. \tag{4.3.10}$$

From the conversion mentioned after (4.3.8): $\beta(n) = e^{j\theta(n)}$, the angle of $\beta(n)$, may now be plotted: which is a result in radians.

4.3.3 Tracking Two Complex Sinusoid Signals

Now, the structure required to track multiple CSSs is considered. To track two CSSs with this new structure an output-error approach is applied; thus, calculating gradients as shown in Figure 4.6. Herein, an additional complex notch filter structure CNF_3 is required to generate the gradient value for CNF_2. This requires only two additional significant multipliers as in Figure 4.5, i.e. for the α and β multipliers.

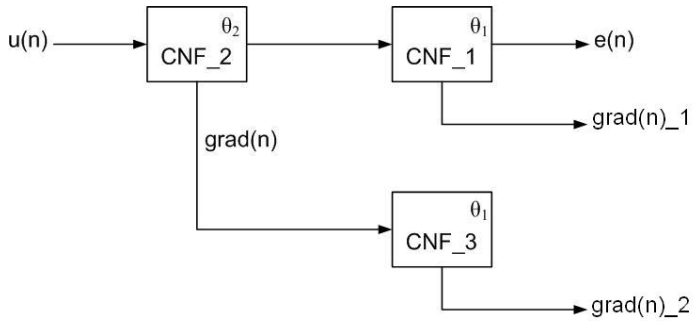


Figure 4.6. The proposed structure required for tracking two complex sinusoid signals.

This structure is easily expanded to track more than two frequencies; however, in this chapter just two CSSs are considered. To preserve the stability of the notch structures, α is fixed to 0.8 in all these simulations. The adaptation gain μ is empirically selected to avoid instability in the learning algorithm in these experiments. Please note that formal stability analysis has not been completed in this thesis, however background information to

facilitate this may be found in [34].

4.4 Simulation Results and Comparison

This section of the chapter, provides a comparison of this new structure to Regalia's approach [23], for tracking frequency hopping CSSs. The simulations for tracking a single CSS are shown in Figure 4.7; next Figure 4.8 shows the result for tracking two CSSs. The results within this chapter, as in [23], have been achieved with a SNR of 0 dB.

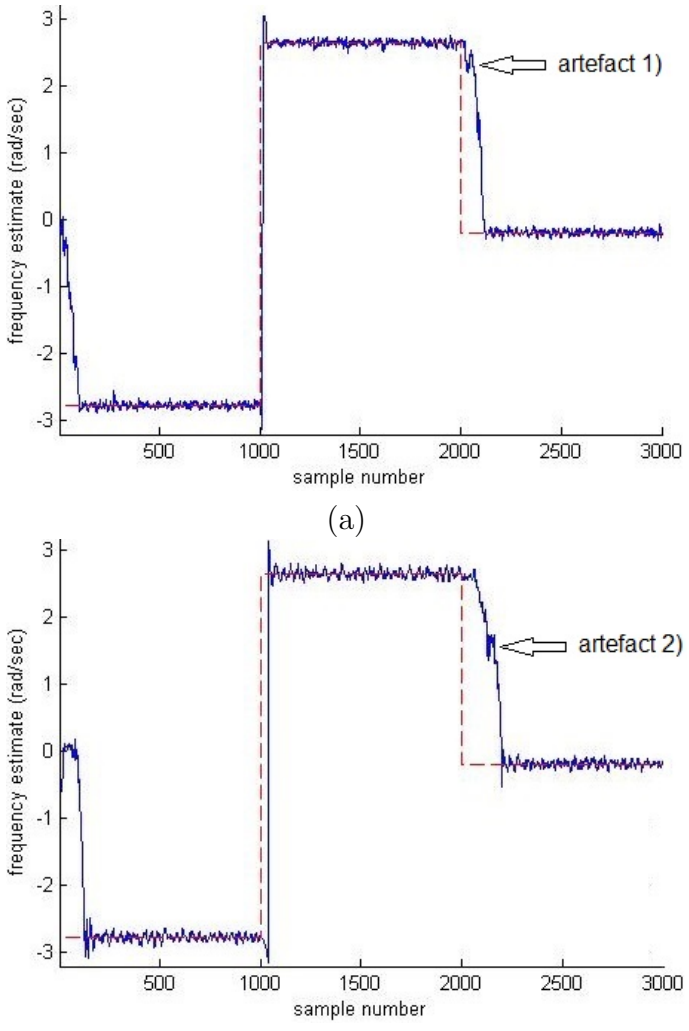


Figure 4.7. A comparison for tracking one complex sinusoid signal with: (a) The proposed structure, and (b) Regalia's structure [23].

Both Figure 4.7 and Figure 4.8 include a target signal which is frequency hopping. This target signal (shown in a dotted red line) is first initialised, then instantaneously changes or hops every 1000 samples in Figure 4.7.

The proposed structure in Figure 4.7 (a) achieved its result with the following tailoring parameters set to: $\mu = 0.15$ and $\gamma = 0.97$. In both simulations θ_0 was initialised as -1.26. Similar tracking performance for Regalia's structure [5], with the result displayed in Figure 4.7 (b), was achieved with the values $\mu = 0.05$ and $\gamma = 0.8$.

Therefore, Figure 4.7 demonstrates that the proposed structure performs as well as, or better than the previous approach [23], as there are visible convergence improvements in these results. One example of the improved performance is the faster convergence to the final frequency, which occurs following the final hop at sample number 2000.

It should be noted that adding the term ψ to Regalia's scheme presented in [23] improves the performance of his structure further, thus adopting a NLMS approach; which facilitates a fair comparison, as if ψ was omitted from [23], the structure proposed in this chapter significantly outperforms Regalia's.

Both Figure 4.7 (a) and (b) contain an artefact, where the notch output is oscillating: which is labelled as 'artefact 1)' in (a), and 'artefact 2)' in (b); and this artefact occurs in simulations when the magnitude of the gradient approaches zero.

Now the ability of both structures is compared for tracking the frequencies of two CSSs, and this result is shown in Figure 4.8[1], on the next page; and within Figure 4.8 the exact normalised frequencies that are tracked finish the simulation at 1.5708 and 1.1938. The parameters that produced the result in Figure 4.8 (a) are: $\mu = 0.08$ and $\gamma = 0.99$, whereas for Figure 4.8

[1] An analysis of the CSSs frequencies tracked in Figure 4.8 is included in section 6.4.3 of this thesis.

(b): $\mu = 0.02$ and $\gamma = 0.8$, these values have been found empirically to yield optimum results for convergence with equivalent tracking performance.

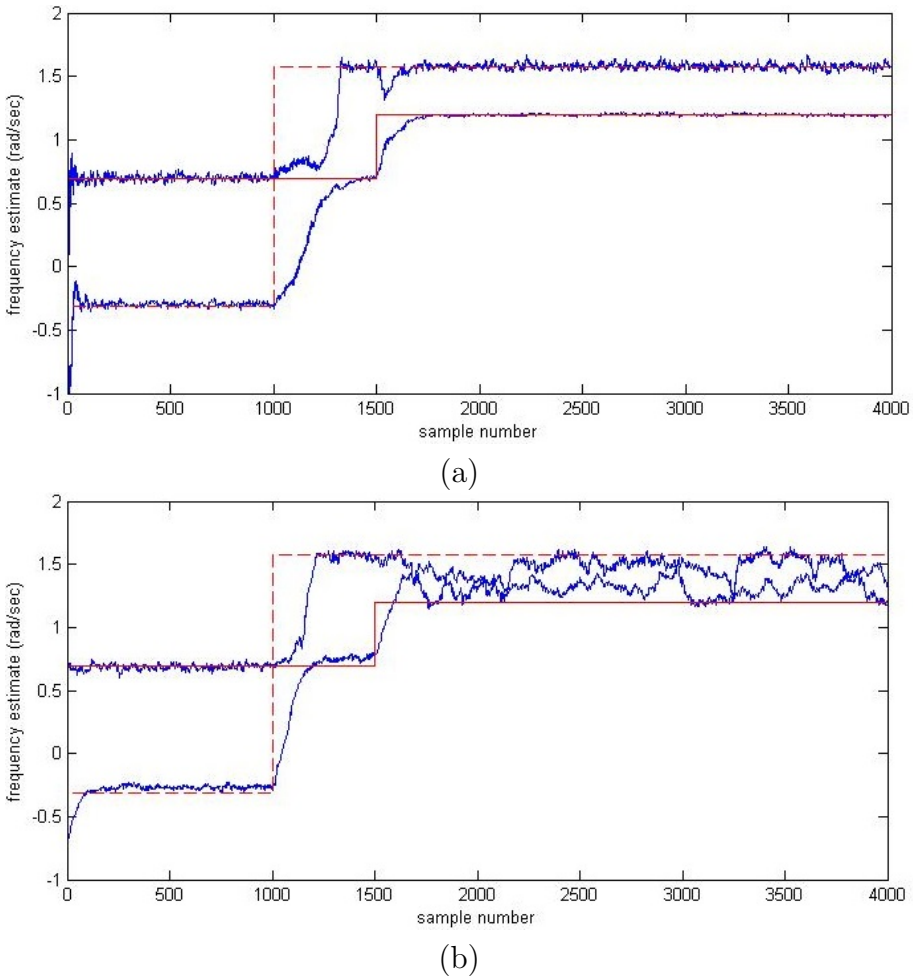


Figure 4.8. A comparison for tracking two complex sinusoid signals with: (a) The proposed structure, and (b) Regalia's structure [23].

From Figure 4.8 (b), please observe when the target frequencies are closer together: after sample number 1500, Regalia's structure fails to converge, this effect worsens if you decrease the value of α further below 0.8. This failing is a consequence of using an equation-error based learning algorithm in [23], which reduces the complexity of the gradient generated in the learning algorithm. The frequencies within Figure 4.8 were selected at the point Regalia's design fails; as it cannot track frequencies that are within 0.377: when α is set to 0.8.

The average estimated frequency errors and the variances in the frequency estimates are shown in Table 4.1, which are calculated from the last 200 samples in Figure 4.8 (a) and (b).

The advantage of this proposed structure is confirmed by the reduction in error and variance presented in Table 4.1.

Table 4.1. Results for tracking two complex sinusoid signals, comparing Regalia's method to the proposed structure; wherein $\Delta\omega_1$ & $\Delta\omega_2$ are sample biases and σ_1^2 & σ_2^2 are the sample variances.

	Regalia [23]	The proposed structure
$\Delta\omega_1$	0.0235	-0.0014
$\Delta\omega_2$	-0.0176	0.0001
σ_1^2	0.0075	0.0030
σ_2^2	0.0145	0.000775

Notes on the new structure

It should be observed that, both the structure in [23], and this new structure, require only one delay element when utilised to track a CSS: which is equivalent to a single pole and zero or one notch. The original structures, which are only capable of tracking RSS, require two delay elements, thus have two poles and two zeros: which creates two symmetrical notches.

4.5 Discussions on Complex Adaptive Notch Filter Development

An original CANF structure has also been developed from [1], and this new structure is based upon an output-error learning algorithm. Its superior tracking performance over Regalia's CANF [23], has been demonstrated; particularly for tracking two CSS in close proximity.

Generally, an output-error approach will facilitate more reliable tracking than an equation-error approach for real and complex signals; however, output-error approaches are slightly more computationally complex. For practical applications such as safety critical control systems [9]: where notch

filters are often used, the low complexity and robustness of this proposed structure are indeed very attractive.

4.6 Summary

This chapter firstly introduced tracking CSSs, then evaluated Regalia's 2010 publication [23]. Next, it developed Chambers' structure [1] into an equivalent form, and compared its performance to Regalia's scheme [23] for the tracking of a single CSS, then two CSSs; this provided a comparison of the equation-error and output-error approaches within CANFs, thus highlighting the advantages of both methods.

The next chapter of this thesis considers adapting the notch bandwidth parameter, with the aim of improving the performance of the CANF developed in this chapter further.

Chapter 5

ADAPTING THE NOTCH BANDWIDTH AND FREQUENCY PARAMETERS: α AND β SIMULTANEOUSLY

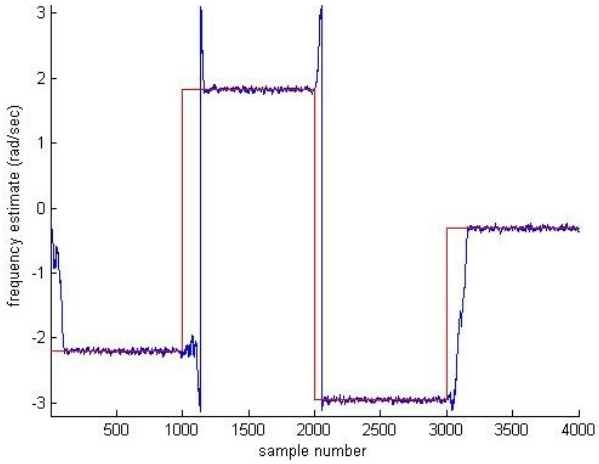
5.1 Introduction

In this chapter, the new CANF structure is extended further; as previously in Chapter 4 the notch bandwidth parameter α was fixed. Now, the design is developed so that both parameters α and β can be adapted simultaneously, to facilitate further improvements to the performance of this CANF.

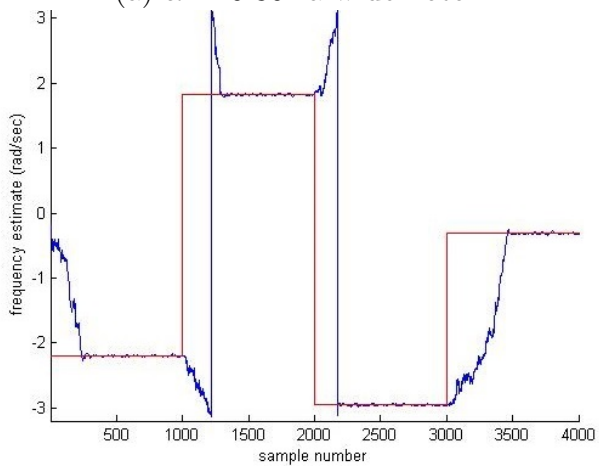
Within this CANF β tracks the frequency of the CSS, whilst α controls the bandwidth of the notch; where α is a real value in the range: $0 \ll \alpha < 1$. However, α must not exceed unity, as if this was the case, a pole would move outside the unit circle; thus, creating instability in the filter. Previously, little research has been published with respect to updating the notch bandwidth parameter, although a literature review on this topic has been included in the next section of this chapter.

Please note that, all results within this chapter have been produced with a SNR of 0 dB, unless otherwise stated.

The effect of using different values of α is demonstrated in Figure 5.1. Consider when observing Figure 5.1[1], that part (a) utilises a low value of α, to implement a wide notch, whilst (b) applies a value of α close to unity, to produce a narrow notch.



(a) $\alpha = 0.80$: a wide notch.



(b) $\alpha = 0.95$: a narrow notch.

Figure 5.1. The effect of using different values of α when tracking a complex sinusoid signal, the red curves represent the target frequencies whilst the blue curves are the frequency estimates - this notation is used in the ensuing figures.

Observe within Figure 5.1, that in part (a) the 'wide notch' quickly locates the target frequency, however, the estimate contains a significant amount of noise: once the target signal has been located; whilst in (b),

[1]To observe different notch widths, please refer to Figure 3.2 from Chapter 3 of this thesis.

the 'narrow notch' is much slower to locate the target frequency; although once the target frequency has been located, the noise on the estimate is significantly less.

Table 5.1 clearly shows the number of samples required to locate the target frequency in Figure 5.1 at: initialisation 0 samples, hop one - 1000 samples, hop two - 2000 samples, then at hop three - 3000 samples. Thus highlighting that a wide notch locates a target frequency much more quickly than a narrow notch.

Table 5.1. The number of samples required to locate a target signal with a wide and a narrow notch.

	Number of samples required to locate the target			
	Initialisation	Hop One	Hop Two	Hop Three
(a) wide notch	100	100	50	100
(b) narrow notch	200	200	150	450

Therefore assuming a hopping CSS is being tracked, stronger performance can be achieved by selecting a wide notch each time the CANF searches for a frequency; then adapting to a narrow notch once the target frequency has been located. This approach will improve a system's SNR improvement ratio[2], thus ultimately the CANF's performance.

To further demonstrate the improvement achieved in the final values, the mean and the variance for the last 200 samples from Figure 5.1 have been included in Table 5.2.

Table 5.2. The difference from the final value and the variance, when tracking a complex sinusoid signal with a narrow and a wide notch; wherein $\Delta\omega$ is the sample bias and σ^2 is the sample variance.

	(a) wide notch $\alpha = 0.8$	(b) narrow notch $\alpha = 0.95$
$\Delta\omega$	0.0001	-0.0002
σ^2	1.3563×10^{-5}	2.2431×10^{-6}

[2]For more information on the SNR improvement ratio, please refer to Section 2.5.3, from Chapter 2 of this thesis.

Table 5.2 shows that using a narrow notch can reduce the variance by a factor of 10, which can again be reduced to 10^{-7} by increasing α to 0.98; however, sometimes in a simulation where α is set to 0.98 the target frequency is not located with such a narrow notch.

Please note, that this assumes the target frequency is fixed or hopping, and tracking constantly changing signals such as Complex-Valued Chirp Signal (CVCS), will be investigated in Chapter 6 of this thesis.

5.1.1 Literature Review on Adapting the Notch Bandwidth Parameter

There are very few papers relevant to updating the notch bandwidth parameter in ANFs; which are reviewed as follows.

In 1993, Knill developed the ANF which this research is based on [1], to adapt the notch bandwidth parameter. Within Knill's publication [56], she derived the update for α to be

$$\alpha(n) = \alpha(n-1) - \mu_\alpha \frac{\bigtriangledown_{\alpha i}(n)}{\psi(n)}, \tag{5.1.1}$$

although, Knill refers to ψ as σ^2 in [56]. Next, she defines the gradient estimate of the adaptation surface $\bigtriangledown_{\alpha i}(n)$, the mean square estimate as

$$\bigtriangledown_{\alpha i}(n) = \frac{\partial e^2(n)}{\partial \alpha_i(n)} = -2e(n)\frac{\partial y(n)}{\partial \alpha(n)}. \tag{5.1.2}$$

This is interesting, however Knill does not include any results or performance improvements for the update of α in this publication.

From 2001 to 2003 Mvuma, Nishimura, and Hinamoto published several papers that investigated improving the performance of ANFs further; two of these papers [57] and [58] directly update the notch bandwidth parameter.

When reviewing these papers, it was noted that these researchers utilise

the same all-pass notch transfer function that this CANF was developed from i.e.

$$H(z) = \frac{1 - 2\beta z^{-1} + z^{-2}}{1 - \beta(1+\alpha)z^{-1} + \alpha z^{-2}}. \tag{5.1.3}$$

In [57] the β and α parameters are updated with the following equations, herein β is adapted with the following update

$$\beta(n) = \beta(n-1) + \mu_\beta \frac{y(n)x(n)}{\psi_\beta(n)}; \tag{5.1.4}$$

wherein, ψ_β is the gradient energy in the NLMS algorithm as with this original ANF structure. Then α is updated with the following equation

$$\alpha(n) = \alpha(n-1) - \mu_\alpha y(n)\psi(n); \tag{5.1.5}$$

wherein, they state that it can be shown that $\psi(n)$ is

$$\psi(n) = \beta(n)(1+\alpha(n))\psi(n-1) - \alpha(n))\psi(n-2) + \beta(n)e(n-1) - e(n-2); \tag{5.1.6}$$

Interestingly, the signs reverse in their second paper [58], where the update equation for β is declared as

$$\beta(n) = \beta(n-1) - \mu_\beta y(n)x(n), \tag{5.1.7}$$

then α is updated as follows

$$\alpha(n) = \alpha(n-1) + \mu_\alpha y_\alpha(n)\psi_\alpha(n). \tag{5.1.8}$$

A significant point to note herein, is that the updates utilised for α and β are both using the opposite signs, as in [57] β is reduced and α is increased,

however in [58] β is increased whilst α is reduced. This result is highlighted later in this chapter, particularly since it has not been explained in these papers.

A further point to note, is that the α parameter in [57] does not converge to unity, in [57] it converges to values between 0.77 and 0.88 for the four results shown in Figure 3 of [57], which is similar to Figure 4 in [58].

To quote Mvuma "For optimization of the performance of the filter, α is adaptively adjusted to search for its optimum value, α_{opt}, which corresponds to the maximum value of the performance index using an LMS adaptation algorithm." Therefore, he does not explain why the sign has changed; which is a significant point that is considered in this chapter.

They also describe how adapting α improves the SNR improvement ratio or factor as they refer to it in both papers; and this improvement is shown as Figure 2 within [57] and [58].

These papers certainly provide an interesting reference point for adapting a notches bandwidth; and Nishimura has continued to work on ANFs, publishing papers on CANFs such as [50] and [51]: which are discussed in section 2.4.2 of Chapter 2.

Also in 2003, Punchalard et al. published a paper [59], which again is based on the same all-pass notch transfer function which this CANF was developed from

$$H_{notch}(z) = \frac{1 + 2\beta z^{-1} + z^{-2}}{1 + \beta(1+\alpha)z^{-1} + \alpha z^{-2}}; \tag{5.1.9}$$

therefore, it may be deduced that this structure adapts the notch bandwidth parameter quite effectively. However, two of the signs in 5.1.9 are inverted, which appears in error. Again, Punchalard derives effectively the same updates as Knill (5.1.1), thus updating both α and β with a negative sign.

Punchalard does not show the value to which α converges, although they do include well presented three-dimensional plots showing the effect of adapting the notch bandwidth parameter, which is shown as Figure 1 in [59]. Punchalard also limited the range of α from 0.7 to 0.96, and highlighted the same advantages included in the introduction to this chapter, with respect to the effects of adapting α. A last point is that in the conclusion, he also suggests further work on the theoretical analysis of the algorithm.

In 2008 Levin and Ioannou showed a specific application for adapting the notch bandwidth parameter in [18], which is selecting a notch bandwidth parameter for individual disk drives. Although, Levin does not directly adapt the notch bandwidth parameter in his design, section C of this paper describes how the notch bandwidth parameter is changed off-line, then replaced with an on-line configuration; which is clearly a different approach to a fully adaptive structure. Levin, who originally worked on neural networks has continued to research ANFs and published another paper in 2011; however, he did not actively adapt the notch bandwidth parameter in this publication.

In Chapter 10 of Regalia's research monograph [45], he shows that adapting the notch bandwidth parameter in a direct co-efficient scaling structure such as Kwan and Martin's structure alters the magnitude response; therefore, bandwidth adaptation should only be attempted in all-pass structures. In Chapter 10, Regalia also adapts the notch bandwidth parameter by reducing it over time, which obviously wouldn't work for a hopping CSS; although this clearly shows that he recognises the benefits of applying this update.

5.2 Methods Evaluated for Adapting the Notch Bandwidth Parameter α

Now, as a literature review on updating α the notch bandwidth parameter has been completed, which is known as α in this structure and research; α is updated for the CANF that has been developed in Chapter 4. Four methods are considered for updating α simultaneously whilst tracking a CSS with β; and these four methods have been described in Sections 5.2.1 to 5.2.5.

5.2.1 The Partial Gradient Term for Adapting α

In this first method, a partial gradient term is derived and evaluated for updating α; which is the method that was applied for updating β. To begin this process, observe Figure 5.2, which shows the structure required to generate the partial gradient update; '$grad\_\alpha(n)_{part}$' the gradient for the parameter α is derived as follows.

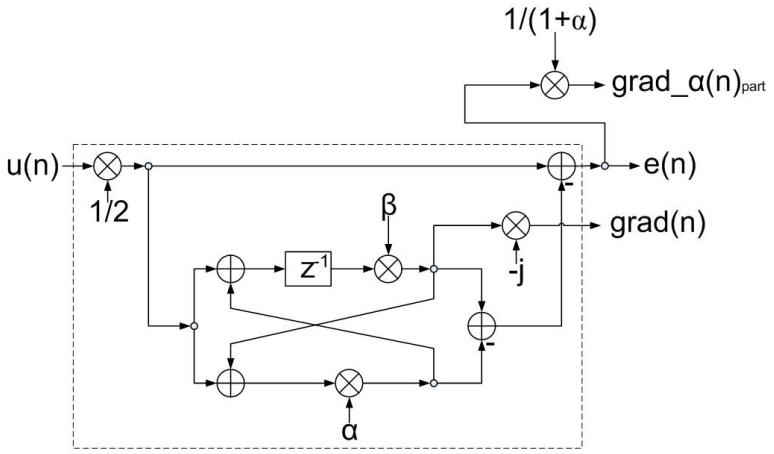


Figure 5.2. The structure required to implement the partial gradient update for α.

Firstly, recall from Section 3.3 of this thesis, that the z-domain transfer

function for this CANF is

$$H_{notch}(z) = \frac{E(z)}{U(z)} = \frac{1}{2}\{1 + A(z)\} = \frac{1}{2}\frac{(1+\alpha)(1-z^{-1}\beta)}{(1-\alpha z^{-1}\beta)}. \tag{5.2.1}$$

Then, if it is assumed that α is fixed in the denominator: as was the case for β; since α is real, thus the gradient for α can be calculated to be

$$\frac{\partial H_{notch}(z)}{\partial \alpha} = \frac{1}{2}\frac{(1-z^{-1}\beta)}{(1-\alpha z^{-1}\beta)} = \frac{GRAD\_\alpha(z)_{part}}{U(z)}. \tag{5.2.2}$$

Interestingly, the expression (5.2.2) is very similar to the notch filter output $e(n)$ from the notch structure (5.2.1); with the exception of a multiplication of $(1+\alpha)$, which can simply be cancelled by multiplying by its inverse. Next as before, the instantaneous cost function J for this structure, is defined as

$$J = |e(n)|^2 = e^*(n)e(n). \tag{5.2.3}$$

Again, $|.|$ and $(.)^*$ denote respectively the modulus and conjugate of a complex number. As before, a LMS type update for the unknown parameter α can be derived from J. This derivation begins with the update equation

$$\alpha(n) = \alpha(n-1) + \frac{\mu_\alpha}{2} \bigtriangledown J|_{\alpha(n)=\alpha(n-1)}, \tag{5.2.4}$$

where μ is the adaptation gain. Hence, by applying the differentiation of a product rule to (5.2.3), creates the expression

$$\bigtriangledown J = e(n) \bigtriangledown e^*(n) + \bigtriangledown e(n) e^*(n), \tag{5.2.5}$$

herein, as $\bigtriangledown e(n) = e(n)\dfrac{1}{(1+\alpha(n-1))}$ thus (5.2.4) becomes

$$\alpha(n) = \alpha(n-1) + \mu_\alpha Re\left(e^*(n)e(n)\frac{1}{(1+\alpha(n-1))}\right). \tag{5.2.6}$$

The NLMS algorithm includes a recursive calculation of the gradient energy ψ; and this calculation is

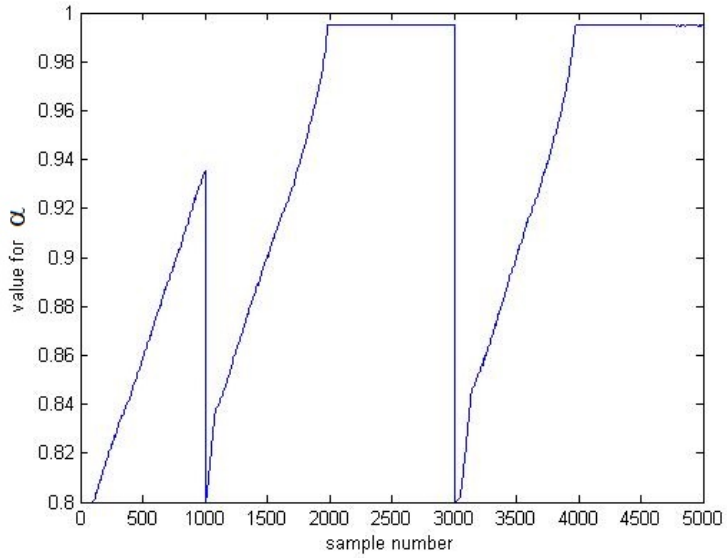
$$\psi_\alpha(n) = \psi_\alpha(n-1)\gamma + (1-\gamma)(grad\_\alpha(n).grad\_\alpha^*(n)), \tag{5.2.7}$$

where the term γ is the forgetting factor: this is a value between zero and one, which is typically $\gg 0$. Thus, updating (5.2.6) with (5.2.7) yields the final update equation for α to be

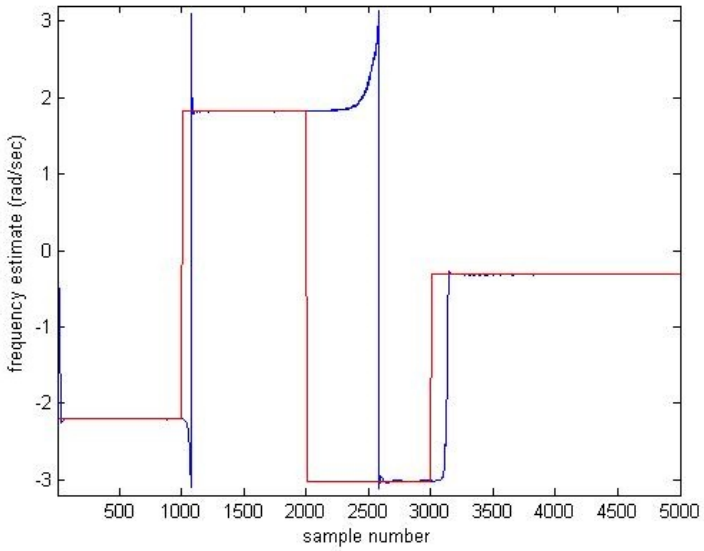
$$\alpha(n) = \alpha(n-1) + \mu Re(e^*(n).grad\_\alpha(n)/\psi_\alpha(n)). \tag{5.2.8}$$

When this is simulated, as shown in Figure 5.3 (a), this method does not converge as expected: due to a simplification that is explained later in this section; and the approach still fails to converge, even with a significant reduction in noise, which was implemented by increasing the SNR from 0 to 14 dB. From Figure 5.3 (a) observe that at 2000, and 4000 samples, the gradient attempts to overshoot unity: which would create instability; therefore α must be constrained, thus demonstrating its non-convergence with this approach. Also note that, β does not track the target frequency correctly, which is shown in Figure 5.3 (b), as between 2000 and 2600 samples β fails to lock onto the new target frequency, as shown in Figure 5.8 (b).

To create these results, different adaptation gains are required for the update of α and β; wherein μ_β was defined as 0.1; whilst μ_α was set to 0.0004. The forgetting factor: γ, used for the adaptation of both α and β was 0.9. The parameter α was initialised at 0.8, and if the notch filter output: $e(n)$ exceeded three, α was reset to 0.8, in an attempt to facilitate more stability in the overall convergence of both values.



(a) The adaptation of α in this CANF



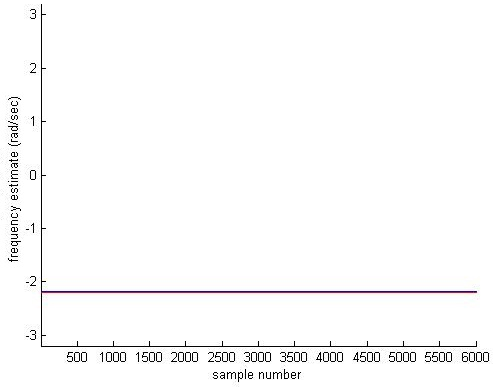
(b) The tracking performance of the CANF

Figure 5.3. The partial gradient approach for adapting α, whilst following a hopping complex sinusoid signal which is tracked by β; with a signal-to-noise-ratio of 14 dB.

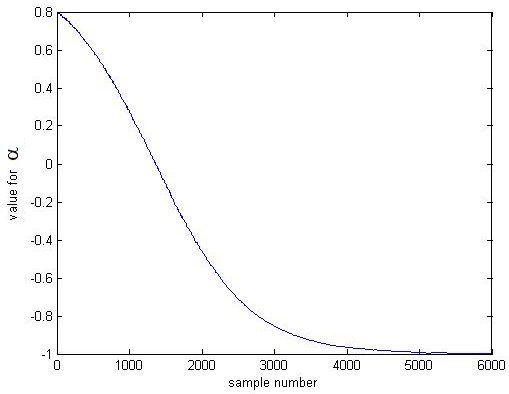
5.2.2 The Cost Function for the Notch Bandwidth Parameter

A key point to be noted is that in this derivation the cost function applies a direction of ascent as α updates; previously this has been implemented by other researchers although not fully explained, which is the case in [57] and [58]. The expected result was that the cost function for α would be implemented with a direction of descent.

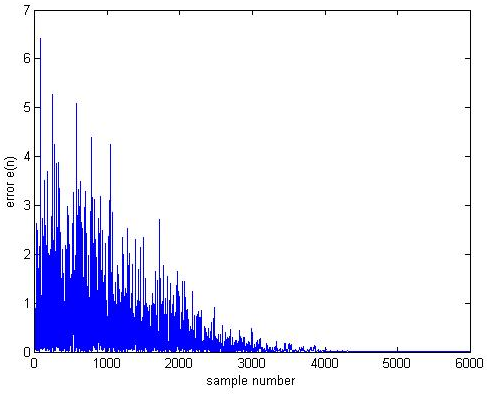
This can be explained, by producing a simulation where the notch frequency is perfect, whilst the notch bandwidth is defined, then the noise variance does indeed decrease; this simulation is included as Figure 5.4; wherein, part (a) shows that the notch frequency is fixed at the target frequency, whilst part (b) illustrates the adaptation of α, which adapts from 0.8 converging to minus one, lastly part (c) demonstrates that the noise variance decreases as α converges to minus one. Therefore, Figure 5.4 shows that a cost function applying a direction of descent reduces the noise output, however, if β also adapts the performance of the adaptive filter reduces significantly; as α should not converge to minus one; which is the case, as shown in part (b).



(a) Shows that β has been fixed to the target frequency



(b) The unrestricted adaptation of α applying a method of steepest descent



(c) The notch filter output e(n) output from the CANF

Figure 5.4. Adapting α, whilst β is fixed at the target frequency; with the full gradient term and a decreasing cost function.

Following on from Figure 5.4, the reduced performance is demonstrated in Figure 5.5, where the transfer function for this notch filter is plotted whilst β is fixed, with values of α set to -0.8 then -0.99.

Thus, if a single CSS with noise is assumed, whilst β is set at the exact correct frequency of this CSS; then the only output of the filter will be the noise. Therefore, to minimise the noise the notch filter will become as wide as possible, as it attempts to reduce the noise variance. This effect is shown in Figure 5.5, which illustrates as α approaches minus one, the notch widens over the complete frequency spectrum, thus producing essentially an all-stop response.

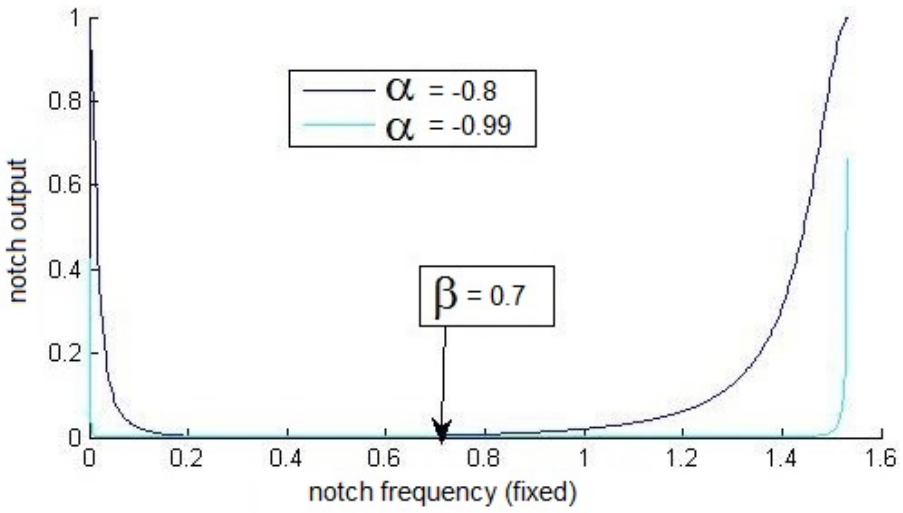


Figure 5.5. The frequency response of the notch filter, wherein α is approaching minus one, whilst β is fixed at the target frequency.

Thus considering Figures 5.4 and 5.5, the effect of α increasing is demonstrated in Figure 5.6; wherein the integral: which is defined in (5.2.9) and (5.2.10), also increases. Therefore, this implies that a method of steepest ascent should be used to adapt α, to enable the transfer function to approach the perfect notch case. The perfect notch case is when the CSS has been removed, so all that is left in the notch output is the white noise.

Describing this mathematically, the noise variance (σ^2) of the notch out-

put signal, which is found from the power spectral density, can be defined as

$$\sigma^2 = \frac{1}{2\pi}\int_{-\pi}^{\pi} P(\omega)d\omega; \tag{5.2.9}$$

where $P(\omega)$ is the power spectral density of the notch output signal. This equation may also be re-arranged to the form

$$\sigma^2 = \frac{\sigma_N^2}{2\pi}\int_{-\pi}^{\pi} |C(\omega)|^2 d\omega, \tag{5.2.10}$$

where $C(\omega) = C(z)|_{z=e^{j\omega}}$, and the remaining term σ_N^2 is the input noise variance. Thus a steepest ascent algorithm should be utilised in the adaptation of α. Therefore, the cost function should apply a direction of ascent to produce a solution to this integral.

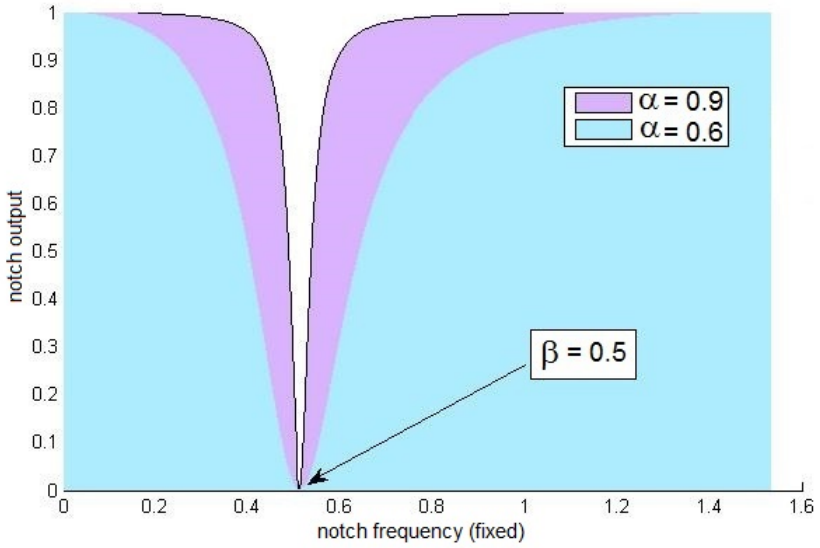


Figure 5.6. The increase in the integral or mean-square-error as α approaches unity.

Thus, Figure 5.6 demonstrates that for the adaptation of α, the update equation should apply a direction of ascent; therefore the update equation

for α should be

$$\alpha(n) = \alpha(n-1) + \mu_\alpha Re(e^*(n).grad\_\alpha(n)/\psi_\alpha(n)). \tag{5.2.11}$$

Now that the correct form of the learning algorithm has been identified, as the partial gradient approach significantly reduced the design's ability to converge; this method is developed further in the next section. Therefore, a full gradient term is developed next, by removing the assumption that α is fixed in the denominator; which implies that the partial gradient approach is too simplistic for the update of α, and therefore is not considered further in this form within a steepest ascent algorithm.

5.2.3 The Full Gradient Term for the Update of α

As the initial results showed that the partial gradient result achieved poor performance; by removing the assumption that α is fixed in the denominator, a full gradient term can be derived as follows. As with the partial gradient approach, this process begins with the notch filter equation

$$H_{notch}(z) = \frac{E(z)}{U(z)} = \frac{1}{2}\frac{(1+\alpha)(1-z^{-1}\beta)}{(1-\alpha z^{-1}\beta)} \tag{5.2.12}$$

$$= \frac{1}{2}(1+\alpha)(1-z^{-1}\beta)(1-\alpha z^{-1}\beta)^{-1}. \tag{5.2.13}$$

Now, as the term $(1-z^{-1}\beta)$ does not contain α, there is no need to differentiate it with respect to α. Thus, differentiating the remaining two products containing α from (5.2.13), with respect to α; via the differentiation of products rule creates the expression

$$\frac{\partial}{\partial\alpha} = \frac{1}{2}(1-z^{-1}\beta)[z^{-1}\beta(1+\alpha)(1-\alpha z^{-1}\beta)^{-2} + (1-\alpha z^{-1}\beta)^{-1}]. \tag{5.2.14}$$

Equation (5.2.14) can now be rearranged to the form

$$\frac{\partial}{\partial\alpha} = \frac{1}{2}(1 - z^{-1}\beta)\left[\frac{z^{-1}\beta(1+\alpha)}{(1-\alpha z^{-1}\beta)^2} + \frac{1}{(1-\alpha z^{-1}\beta)}\right]. \tag{5.2.15}$$

Next, implementing a common denominator for (5.2.15), creates the expression

$$\frac{\partial}{\partial\alpha} = \frac{1}{2}(1 - z^{-1}\beta)\left[\frac{z^{-1}\beta(1+\alpha) + (1-\alpha z^{-1}\beta)}{(1-\alpha z^{-1}\beta)^2}\right]. \tag{5.2.16}$$

Now by multiplying out (5.2.16), and simplifying this term leaves

$$\frac{\partial}{\partial\alpha} = \frac{1}{2}(1 - z^{-1}\beta)\frac{(1+z^{-1}\beta)}{(1-\alpha z^{-1}\beta)^2}. \tag{5.2.17}$$

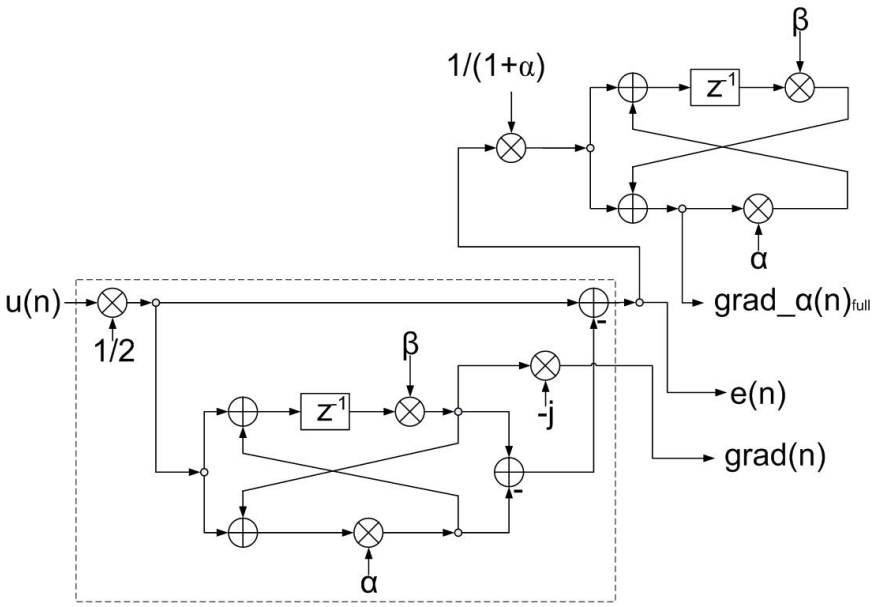
Lastly, splitting the denominator, creates the expression

$$\frac{\partial}{\partial\alpha} = \frac{1}{2}\frac{(1+z^{-1}\beta)}{(1-\alpha z^{-1}\beta)} \times \frac{(1-z^{-1}\beta)}{(1-\alpha z^{-1}\beta)}. \tag{5.2.18}$$

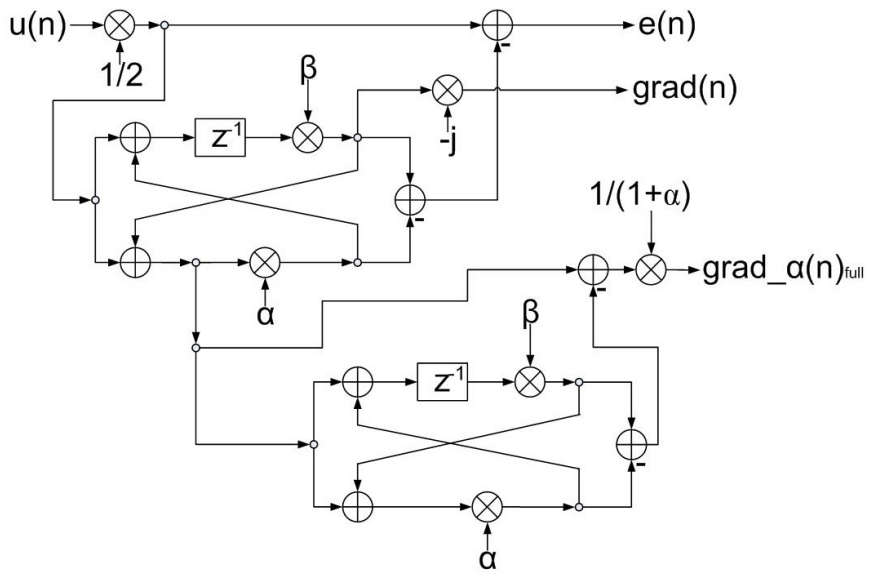
To implement this expression a second filter is required, as to multiply two z-domain transfer functions, it is necessary to cascade two filters in the time domain. Therefore, the full gradient term has been derived as

$$\frac{GRAD\_\alpha(z)_{full}}{U(z)} = \frac{1}{2}\left[\underbrace{\frac{(1+z^{-1}\beta)}{(1-\alpha z^{-1}\beta)}}_{A} \times \underbrace{\frac{(1-z^{-1}\beta)}{(1-\alpha z^{-1}\beta)}}_{B}\right]. \tag{5.2.19}$$

The structure required to generate the full gradient term for α, which is created from two cascaded filters, is shown in Figure 5.7, this can be built in two configurations shown as (a) and (b), which produce similar results; although (b) requires two more additions in its implementation. The different methods for creating the full gradient term occur as A and B from (5.2.19) can be implemented in two ways.



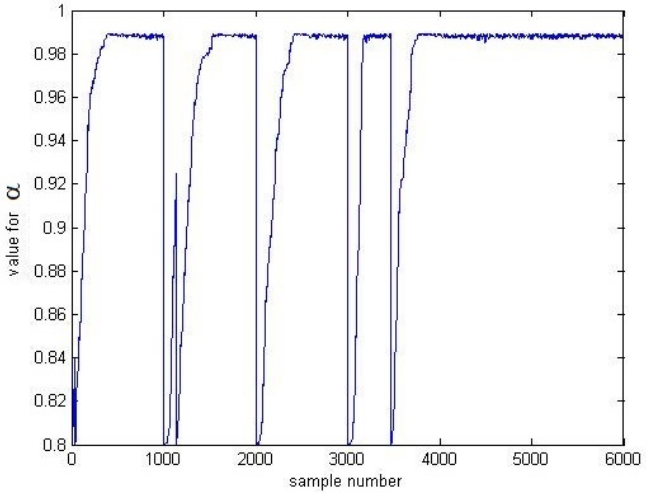
(a) Implementation one of the full gradient term for updating α



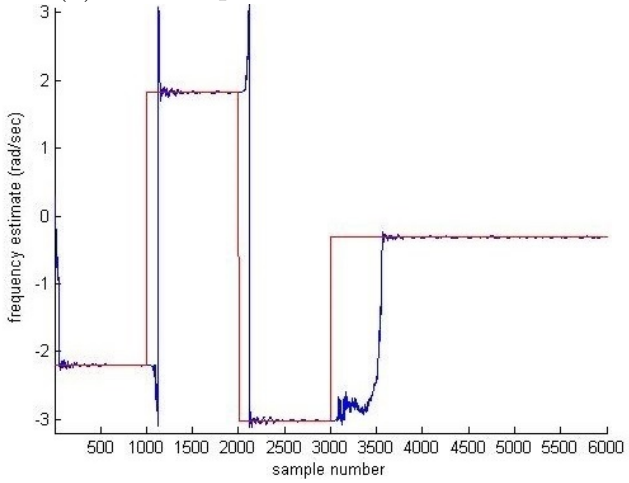
(b) Implementation two of the full gradient term for updating α

Figure 5.7. Two structures capable of implementing the full gradient term for updating α.

In Figure 5.8 the full gradient structure is simulated; and this result demonstrates that this approach provides promising results. Figure 5.8 (a) shows that the result is now clearly stable, as α is not attempting to exceed unity. Also, α is stabilising to the expected value i.e. just below unity, whilst β is tracking correctly, which can be observed in Figure 5.8 (b). Please also observe, that the noise on the estimate is reducing further as α increases, as there is less noise around 4500 samples, when compared to 3250 samples. Therefore, altogether this result provides a significant improvement to the partial gradient approach.



(a) The adaptation of α in this CANF



(b) The tracking performance of the CANF

Figure 5.8. The full gradient approach for adapting α, whilst following a hopping complex sinusoid signal which is tracked by β.

The simulation Figure 5.8 was produced with μ_β set to 0.15; and μ_α being 0.0008. The forgetting factor γ was fixed at 0.9 for the adaptation of α and β.

Again, α was initialised at 0.8, then only adapted if the gradient for β is greater than five, where the gradient for β is shown as grad(n) in Figure 5.7. Then the adaptation of α was constrained by:

if $\alpha(n) > 0.989$

$\alpha(n) = \alpha(n-1)$;

end;

if $\alpha(n) < 0.8$

$\alpha(n) = \alpha(n-1)$;

end;

if $e_{conj}(n) > 6.\psi_{error}$

$\alpha = 0.8$;

end;.

Please note, that these constraints have been found empirically to produce the optimum result in terms of convergence speed and robustness, and are discussed further later in section 5.2.6.

When simulating this structure it can be observed that, occasionally α is adapting before β has located the target frequency, and this effect is shown at 3100 samples in Figure 5.8; as α has adapted before β locks onto the target frequency and requires to be reset to 0.8, which occurs at 3470 samples. This issue can be corrected, and further constraints are discussed later in this chapter.

In the next section, as a significant improvement has been achieved, the full gradient term is simplified via heuristic methods.

5.2.4 Heuristic Simplifications of the Full Gradient Term

Approach 1

Now applying a heuristic approach, which reduces complexity; observing that previously the gradient for β as in Chapter 4 Section 4.3.2 is derived following the β multiplier, it was noted that the point close to the α multiplication: shown in Figure 5.9 as x, may assist in deriving the gradient for α.

Also, it was observed that the partial gradient component: $\frac{1-z^{-1}\beta}{1-\alpha z^{-1}\beta}$ from (5.2.19); did not successfully converge to a solution, therefore the other component: $\frac{1+z^{-1}\beta}{1-\alpha z^{-1}\beta}$ was trialled; and as the transfer function at the bottom of the all-pass filter can be derived to be the second component of this equation. Thus, this part of the full gradient term may be found to provide a suitable approximation for this value. Interestingly, this term contains a high-pass zero, thus provides more smoothing of the gradient.

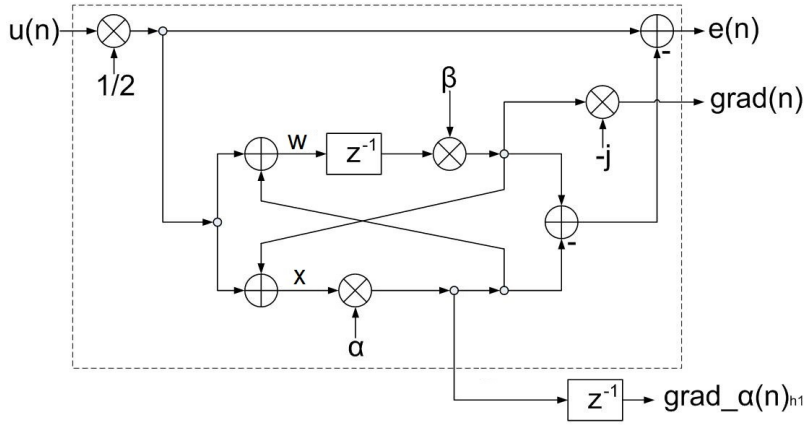


Figure 5.9. The structure which implements the first heuristic simplification for updating α.

Calculating this mathematically, the point shown as x in Figure 5.9 can be derived as $\mathrm{X}(z) = \frac{1}{2}\frac{(1+z^{-1}\beta)}{(1-\alpha z^{-1}\beta)}U(z)$; which is the second component of (5.2.19). The expression for $\mathrm{X}(z)$ can be proven, as before, by using the point w: shown in Figure 5.9, which can be derived to be $\mathrm{W}(z) =$

$\frac{1}{2}\frac{(1+\alpha)}{(1-\alpha z^{-1}\beta)}U(z)$. Then by adding a delay, which replaces the extra filter required. The gradient term applied in this first heuristic approach, can be defined to be

$$\frac{GRAD\_\alpha(z)_{h1}}{U(z)} = \frac{1}{2}\frac{(1+z^{-1}\beta)}{(1-\alpha z^{-1}\beta)}z^{-1}. \tag{5.2.20}$$

Interestingly, this first heuristic approach showed some convergence, which is demonstrated in Figure 5.10, where part (a) of this simulation shows that α is converging to approximately 0.96 each time the target frequency hops; and part (b) demonstrates that the target frequency is correctly tracked, as β locks onto the target frequency each time it hops. Simulations have has been extended to 7000 samples, to prove that α is converging.

This result was created with μ_β set to 0.1; and μ_α equal to 0.03. As before γ was set to 0.9, for the adaptation of both α and β. As with the previous result, α was initialised at 0.8, then α's adaptation was constrained as follows:

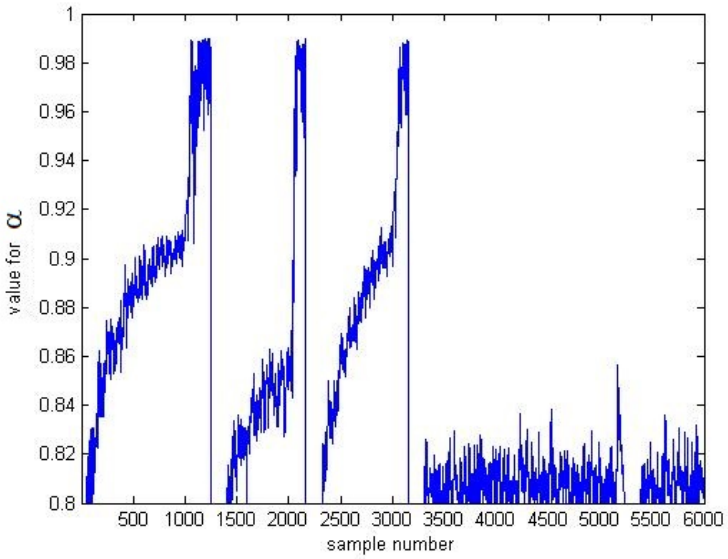
if $\alpha(n) > 0.99$
  $\alpha(n) = \alpha(n-1)$;
end;
if $\alpha(n) < 0.8$
  $\alpha(n) = \alpha(n-1)$;
end;
if $e_{conj}(n) > 10$
  $\alpha = 0.8$;
end;.

However, by observing Figure 5.10 (a), it is clear that α does not appear to approach either of these limits.

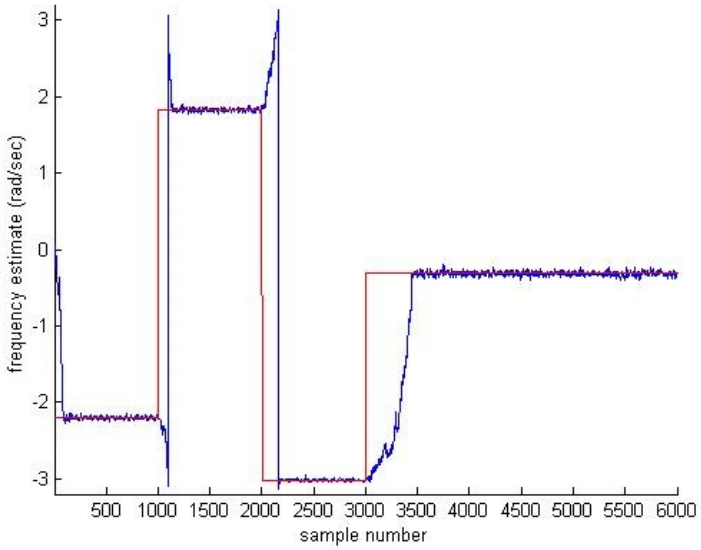
In conclusion, this simulation shows that the full gradient term for α can be approximated, but as a consequence of this simplification some perfor-

mance is lost; as this solution performs poorly around zero, which is shown from 3000 to 6000 samples.

Therefore, a second slightly more complex simplification is trialled in the next part of this section.



(a) The adaptation of α in this CANF



(b) The tracking performance of the CANF

Figure 5.10. Results for following a hopping complex sinusoid signal which is tracked by β, with the first heuristic simplification of the full gradient term for adapting α.

Approach 2

As a significant amount of performance was lost with the first simplification, a second simplification is now investigated which utilises the full gradient term, although without implementing a second filter. This includes an additional multiplication that was not applied in the first simplification, along with the delay which again replaces the second filter. This structure required to implement this term is shown in Figure 5.11.

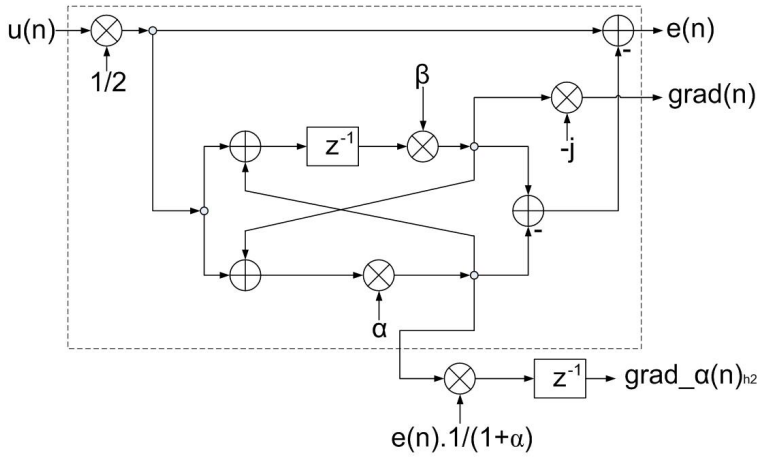


Figure 5.11. The structure required to apply the second heuristic approach for updating α.

Then, expressing the second heuristic simplification mathematically, produces the gradient term

$$\frac{GRAD\_\alpha(z)_{h2}}{U(z)} = \frac{1}{2}\left[\frac{(1+z^{-1}\beta)(1-z^{-1}\beta)}{(1-\alpha z^{-1}\beta)^2}z^{-1}\right]. \tag{5.2.21}$$

Therefore, this approach utilised both terms A & B from (5.2.19) without cascading the structure. This second simplification initially converged, however then drifted, thus was not pursued further.

Next, a method of interconnecting the adaptation of α and β is investigated; i.e. the adaptation of α will be influenced by the adaptation of β.

5.2.5 The Interconnected Parameter Approach

At the beginning of this chapter, it was noted that one possible approach to implement the adaptation of α, is for α to adapt, once β has stabilised. Therefore, a successful method for adapting α could be to interconnect it with the adaptation of β.

Now, as the NLMS algorithm contains the term ψ from the adaptation of β, which has been called ψ_β; the term ψ_β can be used in the update of α as shown in (5.2.22) below; which defines the update equation for α, in this interconnected approach as

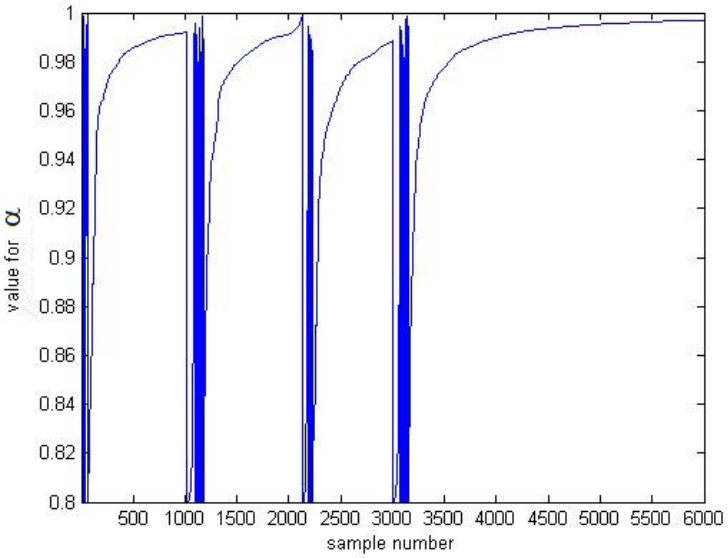
$$\alpha(n) = \alpha(n-1) + \mu_\alpha Re\left(e^*(n)e(n)\frac{1}{(1+\alpha)}/\psi_\beta(n) \right). \tag{5.2.22}$$

The term ψ_β is the recursive calculation of gradient energy, thus implies that as β stabilises, ψ_β increases, which has the effect of increasing α in a stable manor to a maximum, which as previously stated is a value just below unity.

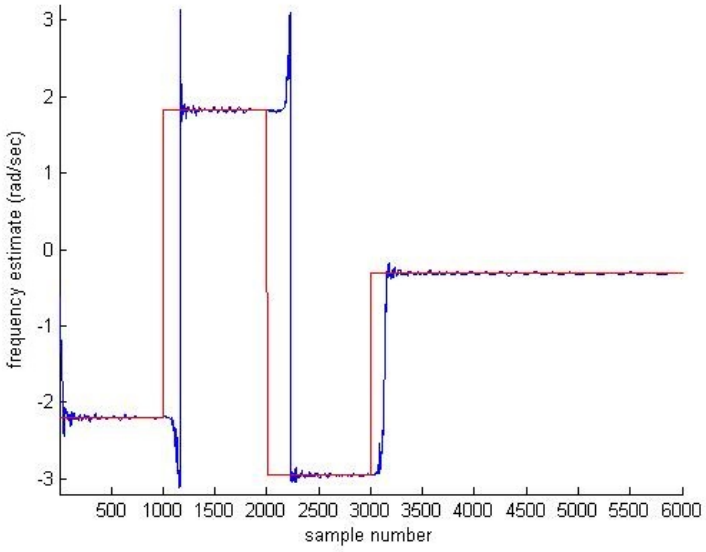
The structure required to implement this result is shown in Figure 5.2, where the only difference from the partial gradient approach is that ψ_β has been used in the NLMS update instead of ψ_α.

This approach shows quite a promising result, when implemented for the partial gradient term, and the result is shown in Figure 5.12 (a). Then Figure 5.12 (b) shows that β is still correctly tracking the CSS's frequency, whilst the noise on the estimate is significantly reducing as α increases.

This simulation was achieved with μ_β fixed at 0.2; whilst μ_α was set to 0.1; which are large values, when compared to the other simulations. As before γ was set to 0.9, for the adaptation of both α and β.



(a) The adaptation of α in this CANF



(b) The tracking performance of the CANF

Figure 5.12. The interconnected parameter approach for adapting α, whilst following a hopping complex sinusoid signal which is tracked by β.

Again α was initialised at 0.8, and with this approach, it was constrained via the following method:

if $\alpha(n) > 1$

$\alpha(n) = \alpha(n-1)$;

end;

if $e_{conj}(n) > 10$

$\alpha = 0.8$;

end;.

The gradient energy ψ_β in the adaptation of β

For this interconnected approach, the effect of utilising the parameter ψ_β has been plotted in Figure 5.13.

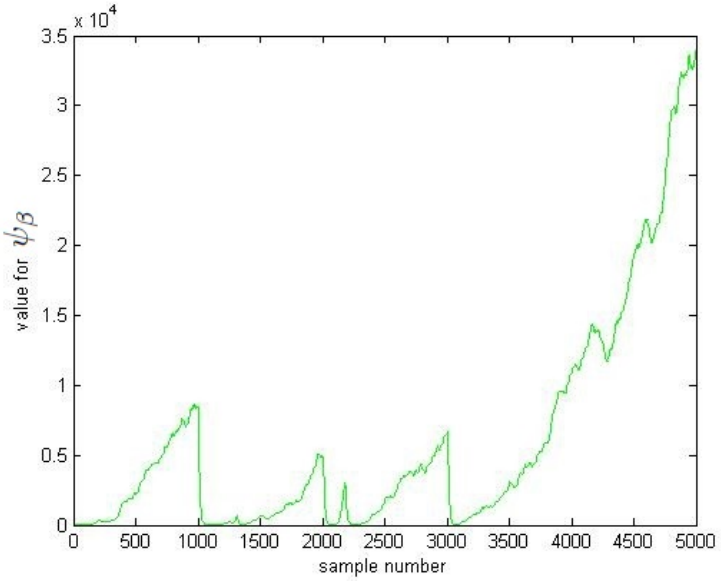


Figure 5.13. The value of ψ_β when tracking a complex sinusoid signal, wherein ψ_β is utilised in the adaptation of α.

Figure 5.13 shows that as β locks onto the target frequency, as is the case at: 250, 1250, 2250 and 3250 samples; the recursive calculation of gradient energy for β increases: which is the term ψ_β. As ψ_β grows, the notch filter output $e^*(n)$ directly controls the value of alpha, as $grad\_\alpha(n)/\psi_\beta$ influences

the update less. It was shown earlier: in 5.2.1, that the notch filter output $e(n)$ is the partial gradient of α; therefore this explains why α successfully converges to a value just below unity.

Simplifying this explanation further, once β successfully locks onto a target frequency; the gradient energy increases; which has the effect of enabling the notch filter output $e(n)$ to directly control the value of α. As $e(n)$ is the gradient of α, this provides a satisfactory result.

This concludes four approaches for adapting the parameter α, and the results can be summarised as follows: a_1) the partial gradient term does not converge correctly, thus produces poor convergence in the results; a) the full gradient term, converges quickly and provides significant improvement on the variance achieved; however, this requires a second filter in its implementation, which will add complexity to the design; b) the heuristic simplification of the full gradient approach provides some improvement, although looses a lot of performance; c) the interconnected parameter approach provides a very stable adaptation for α, however does not converge as quickly as the full gradient term.

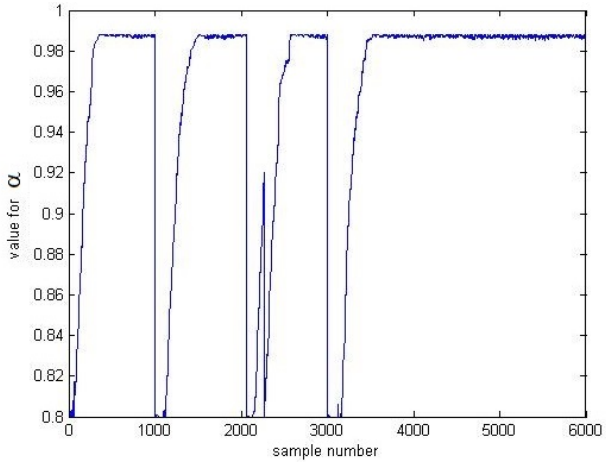
For clarity, the final values that these approaches converge to along with their variances, have been included in Tables 5.3 and 5.4, which are located in the discussions section at the end of this chapter.

5.2.6 Further Constraints Necessary for the Update of α

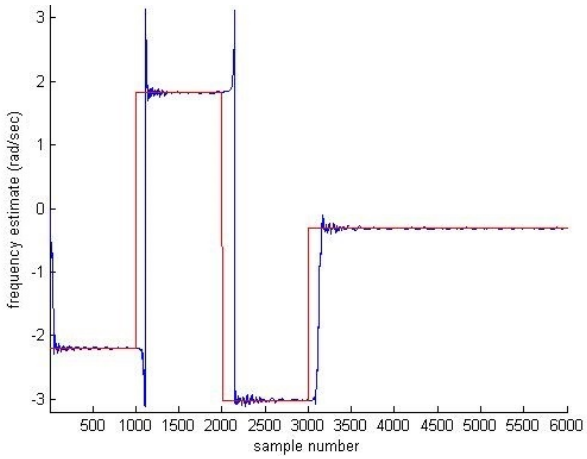
Now, further constraints are implemented to improve the overall performance of the CANF further for the full gradient term. As it was observed that although the variance has been improved, some of the tracking speed when locking onto a different frequency has been lost; thus if α adapts before β locks onto the target frequency the CANF's performance is reduced. Therefore, the adaptation of α has been constrained further in an attempt to speed up the tracking speed.

This research found that by only adapting α if the gradient of β is less than a value, i.e. when β has locked onto a target frequency, results could be improved further; this result has been shown in Figure 5.14. The constraint that was applied herein, was to only adapt α if the gradient for β is greater than five, and all other parameters are the same as in Figure 5.8.

Another approach to implement a further constraint is to fix α, when the notch filter output exceeded a limit for a fixed duration: n samples; and this method did achieve some success, however sometimes triggers incorrectly; therefore, stopping the adaptation of α unnecessarily.



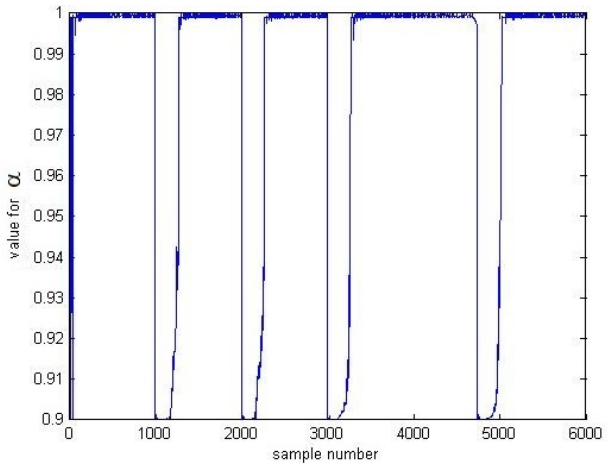
(a) The adaptation of α in this CANF



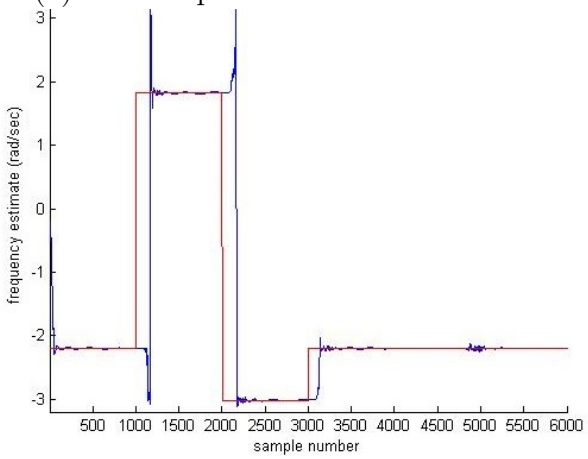
(b) The tracking performance of the CANF

Figure 5.14. Results for tracking a hopping complex sinusoid signal, with a further constrained full gradient term updating α.

A further point to note, which was discovered in this research is, if α approaches unity for a length of time, instability occurs; as α is approaching the structure's stability limit i.e. a pole on the unit circle. Therefore, when α is at its limit the structure shows some instability, and this appears as a blip around 4800-5000 samples which is shown in Figure 5.15. However, this instability can be removed by restricting the adaptation of α to a limit, and this has been implemented in the results; although, this would seem the sensible place to include this information. Please also note when observing Figure 5.15, that noise has almost entirely been removed; unfortunately, some of this performance has to be lost to improve stability.



(a) The adaptation of α in this CANF



(b) The tracking performance of the CANF

Figure 5.15. The instability that occurs, when α approaches unity for a period of time.

Some further observations are: setting the constraints is key to the success of adapting both parameters; therefore, care should be applied when selecting the constraints for certain environments, and the next development step could be to apply adaptation to the selection of constraints.

If the amount of noise changes, the value at which the notch filter output e(n) resets α needs to be adjusted; therefore, for lower noise levels it must be reduced, and for higher noise levels it must be increased.

A method that proved successful for resetting the adaptation of α, was to create a ψ_{error} parameter, which resets the value of α to 0.8; this can be implemented as follows

$$\psi_{error}(n) = \psi_{error}(n-1)\gamma + (1-\gamma).e_{conj}(n);$$
$$\text{if } e_{conj}(n) > 6.\psi_{error}$$
$$\quad \alpha = 0.8;$$
$$\text{end;}$$

wherein, $e_{conj}(n) = e(n)^* e(n)$ thus the notch filter output is converted to be an absolute value.

5.3 Tracking Two Complex Sinusoid Signals, whilst Adapting a Single Value for α

Within the first section of this chapter, it was demonstrated that α can be adapted simultaneously with β to track a single CSS; now, the effect of adapting α whilst tracking two CSSs frequencies is investigated.

For these investigations, the three working solutions are considered, which are: a) the full gradient approach, b) the heuristic simplification, and c) the interconnected parameter approach.

In the output-error approach shown in Figure 5.16, if additional notch filters were required for updating α, this will produce a more complex structure. Therefore, initially only the value of α in the final notch structure has

been adapted, which is labelled as CNF_1 in Figure 5.16.

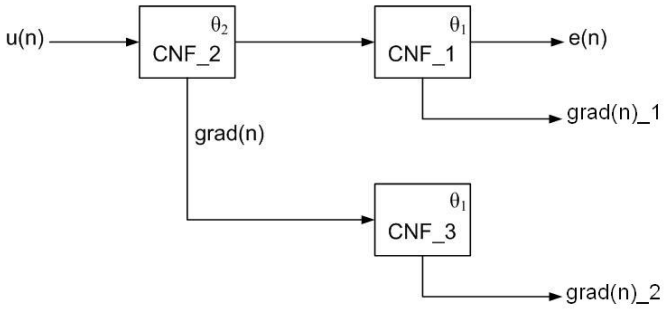


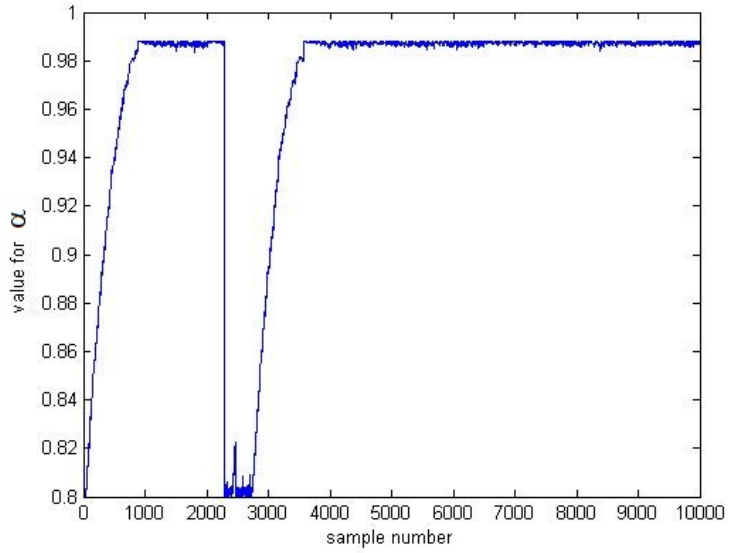
Figure 5.16. Potential structure for adapting a single value for α whilst tracking two complex sinusoid signals.

Again, the results from this section have been embodied into Table 5.5; which is located in the discussions section at the end of this chapter.

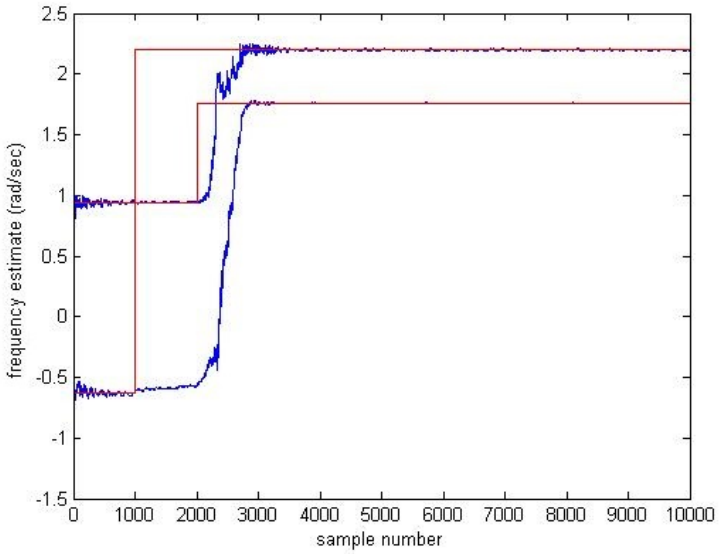
5.3.1 The Full Gradient Approach for Tracking Two Complex Sinusoid Signals, whilst Adapting a Single α Value

In this section, two CSSs are tracked; whilst a single value for α is adapted in the part of the structure shown as CNF_1 in Figure 5.16. The parameter α, is updated via the full gradient approach in CNF_1, and this updated value of α is utilised throughout the entire structure.

This approach shows convergence and an improvement on previous results, which can be observed in Figure 5.17.



(a) The adaptation of α in this CANF



(b) The tracking performance of the CANF

Figure 5.17. Tracking two frequencies whilst adapting a single value for α with the full gradient approach.

From this result, it is clear that α remains at 0.8, until β locks onto both frequencies, once both values of β are locked, as is the case around 2250 samples; α increases to a maximum, thus facilitating a further reduction in noise on the tracked signals. This reduction in noise is clearly visible, when comparing the noise at 2500 samples, to the noise at 10000 samples.

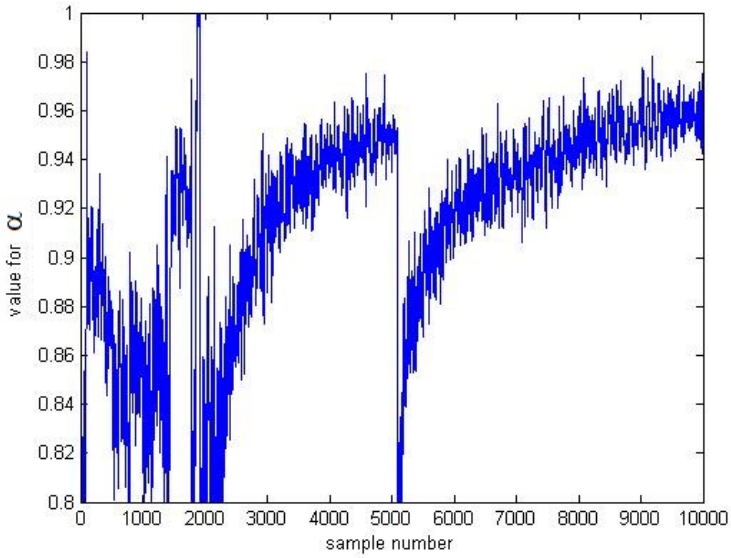
This simulation was produced with $\mu_\beta = 0.1$; and $\mu_\alpha = 0.01$, and γ was fixed at 0.8, for the adaptation of both α and β. For this result, α was initialised at 0.8; and the constraints applied to this result, were the same as for the full gradient term when tracking a single CSS; which is shown in section 5.2.3.

5.3.2 The Heuristic Simplification Approach for Tracking Two Complex Sinusoid Signals Frequencies, whilst Adapting a Single α Value

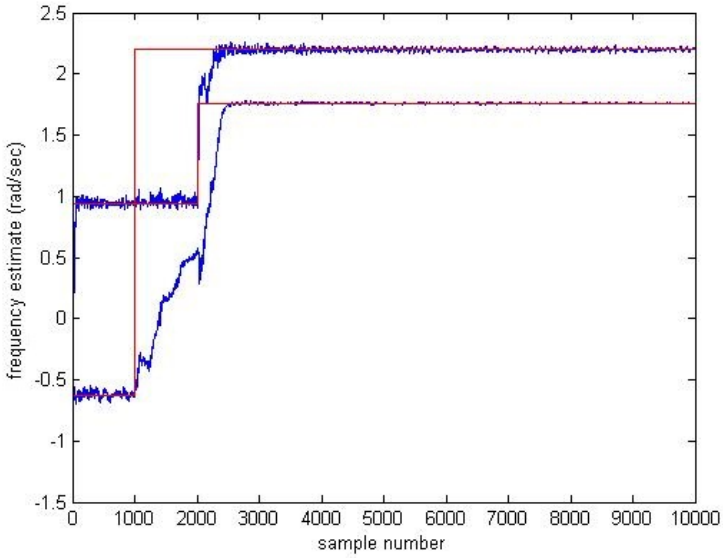
Now, the heuristic approach is considered to evaluate the inevitable reduction in performance as a result of simplifying the full gradient term; whilst updating a single value of α in CNF_1. The results for this simulation are shown in Figure 5.18, which confirms that this method provides a reasonable solution for this scenario.

The values used to create Figure 5.18 were: $\mu_\beta = 0.1$, $\mu_\alpha = 0.04$ and $\gamma = 0.8$ for the adaptation of α and β. The notch bandwidth parameter α was initialised at 0.8; then constrained in the same method as for the heuristic simplification when tracking a single CSS; which was shown in section 5.2.4.

Figure 5.18 shows that the heuristic simplification does appear to reduce the noise on both estimates, however, far more significantly on the lower value 1.7593.



(a) The adaptation of α in this CANF



(b) The tracking performance of the CANF

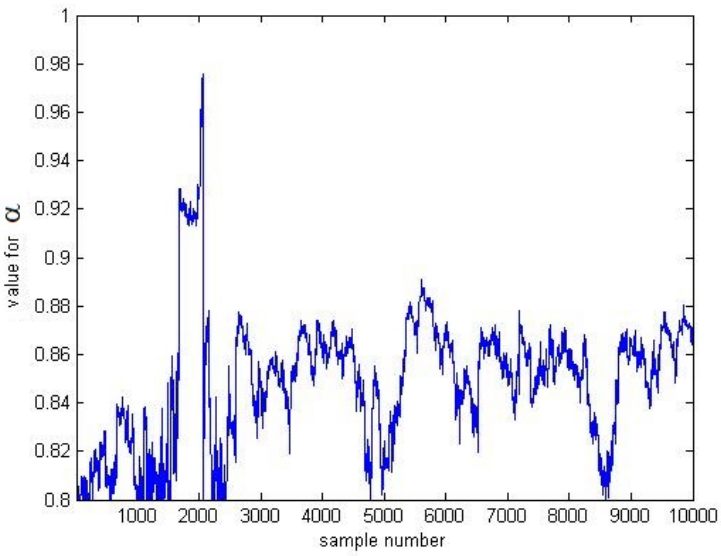
Figure 5.18. Tracking two frequencies whilst adapting a single value for α with the heuristic approach.

5.3.3 The Interconnected Parameter Approach for Tracking Two Complex Sinusoid Signals Frequencies, whilst Adapting a Single α Value

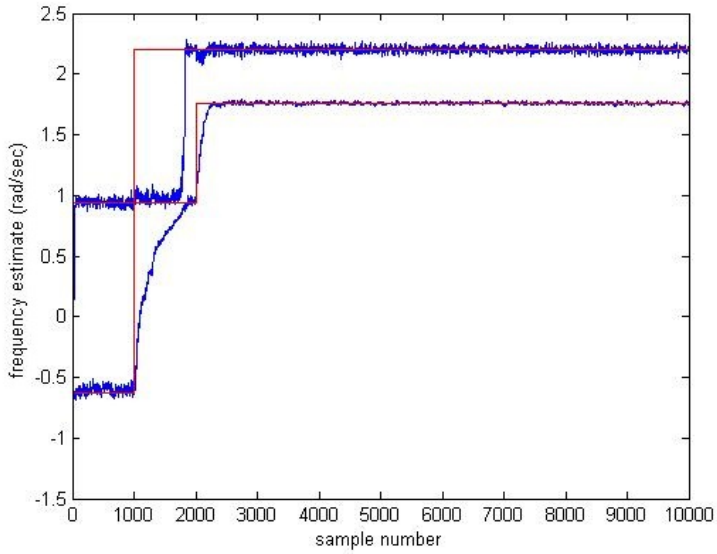
As two CSSs have been successfully tracked, with both the full gradient and heuristic simplification approaches; lastly the interconnected parameter approach is evaluated. To facilitate a fair comparison, again a single value of α is updated via the interconnected parameter approach, in the part of the structure shown as CNF_1: in Figure 5.16, throughout the entire structure.

The results for this experiment are shown in Figure 5.19. Now, by observing Figure 5.19, it can be concluded that the interconnected parameter approach does not appear to work well in this scenario, as α is not truly converging, which is shown in Figure 5.19 (a), whilst (b) clarifies that there is little improvement in the noise reduction.

This simulation was produced with μ_β set to 0.25; whilst μ_α was set to 0.04; as previously noted these are relatively large values, when compared to the other equivalent simulations. Also when tracking two CSSs with this method, more favourable results are achieved by reducing γ to 0.8: which was also the case with the previous result; and this value of γ was applied to the adaptation of both α and β. Again α was initialised at 0.8, then constrained by the same method applied for the interconnected approach when tracking a single CSS; and these constraints are included in section 5.2.5.



(a) The adaptation of α in this CANF



(b) The tracking performance of the CANF

Figure 5.19. Tracking two frequencies whilst adapting a single value for α with the interconnected parameter approach.

5.4 Tracking Two Complex Sinusoid Signals, whilst Adapting Unique α Values

In this section of the chapter, updating multiple values of α is considered. As two frequencies are being tracked by separate all-pass sections, it is clearly possible to update different values of α for each frequency that is being tracked; however, this will increase the complexity of the designs, for instance particularly the full gradient approach. Thus in Figure 5.16, α could be adapted in CNF_2 and CNF_1; instead of just being updated in CNF_1.

Please note, that since a value of alpha is being updated for θ_1 and θ_2, it follows that these unique values of α are referred to as α_1 and α_2, which naturally correspond to the θ values.

Therefore, the three approaches are again considered and simulated in this section; however, final values and variances have again been tabulated in the discussions section within Table 5.6, to clearly show a fair comparison between these structures; thus the full gradient approach is considered first.

5.4.1 The Full Gradient Approach for Tracking Two Complex Sinusoid Signals, whilst Adapting Unique α Values for each Signal

To begin this comparison on adapting multiple values of α, relating to each CSS being tracked, firstly the full gradient approach is considered. To implement this approach an additional notch structure is required, in the same manner an CNF_3; this extra structure is labelled as CNF_4, and also derives the gradient for α_2 from CNF_2. This complete structure required to implement the full gradient approach for tracking two CSSs, whilst adapting unique α values is shown on the next page as Figure 5.20.

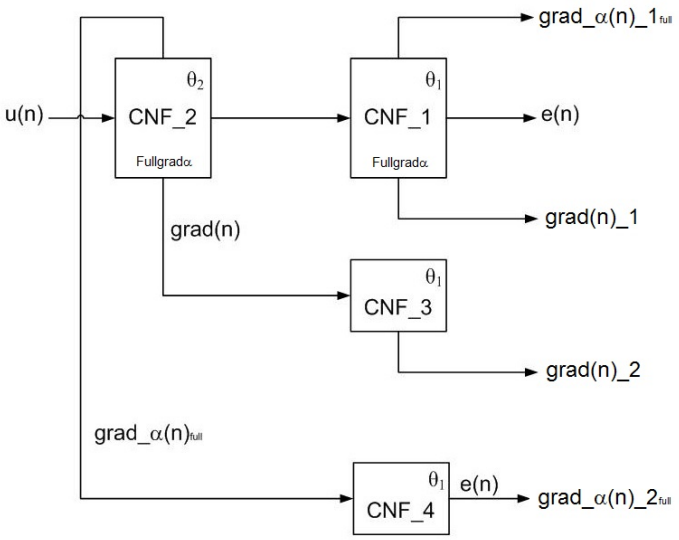
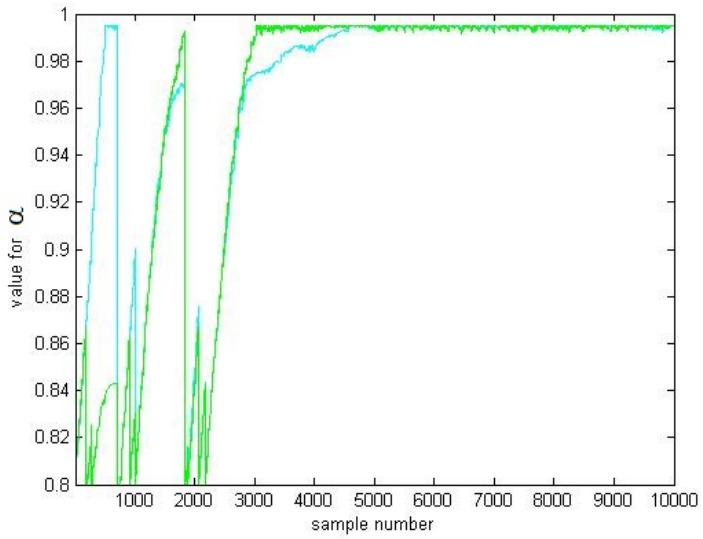


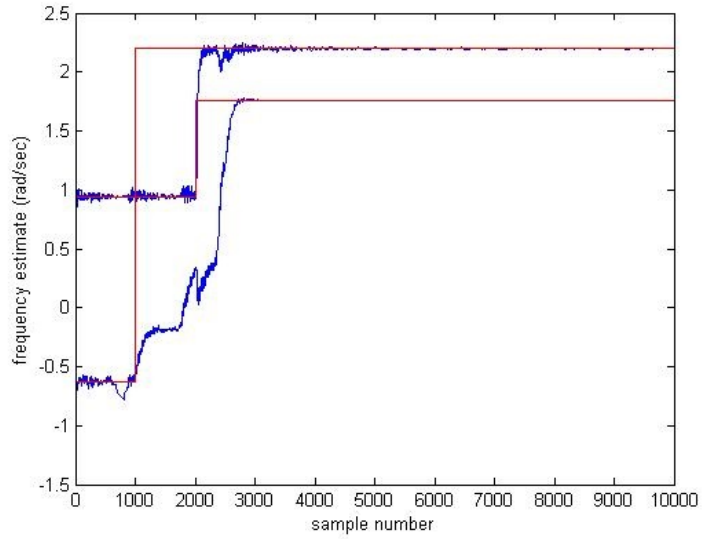
Figure 5.20. A structure capable of tracking two complex sinusoid signals, whilst adapting unique α values for each signal with the full gradient approach.

Please note that in Figure 5.20, CNF_1 and CNF_2 are effectively double CANF structures, as they derive the full gradient term for α, and these structures are shown earlier in Figure 5.7.

The simulation for this structure is included as Figure 5.21; which was produced with $\mu_\beta = 0.1$, $\mu_\alpha = 0.001$; γ was set to 0.8 for the adaptation of α, and 0.9 for the adaptation of β. Other constraints were as previously implemented for this approach.



(a) The adaptation of both α values in this CANF



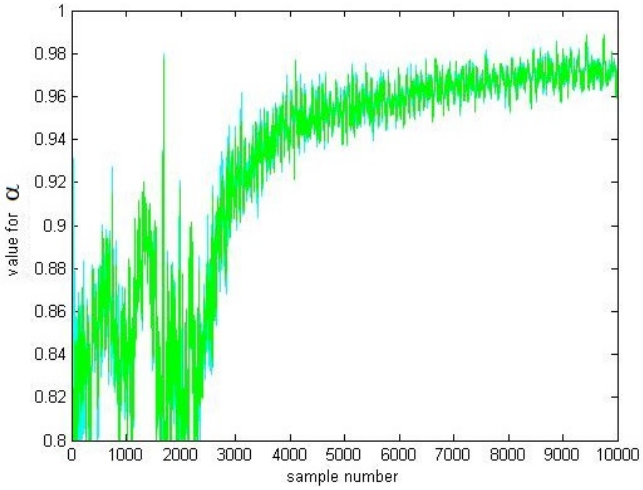
(b) The tracking performance of the CANF

Figure 5.21. Tracking two frequencies whilst adapting separate α values with the full gradient approach.

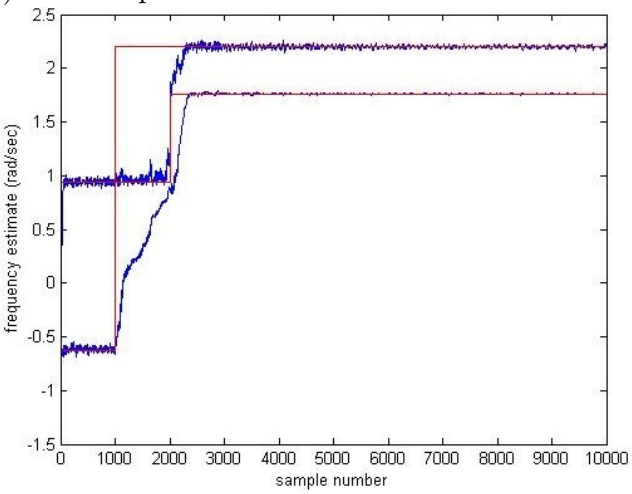
As expected, the full gradient approach produces a very powerful solution when adapting multiple values of α, as there is very little variance on the estimates after convergence; however, it also adds significant complexity.

5.4.2 The Heuristic Simplification Approach for Tracking Two Complex Sinusoid Signals, whilst Adapting Unique α Values for each Signal

Now, considering the heuristic simplification; which is implemented by updating α_2 directly in CNF_2, as adding a further section: which would effectively be CNF_4, did not improve the result further. It was discovered that this approach did achieve a slight overall improvement, and the simulation is included as Figure 5.22.



(a) The adaptation of both α values in this CANF



(b) The tracking performance of the CANF

Figure 5.22. Tracking two frequencies whilst adapting separate α values with the interconnected parameter approach.

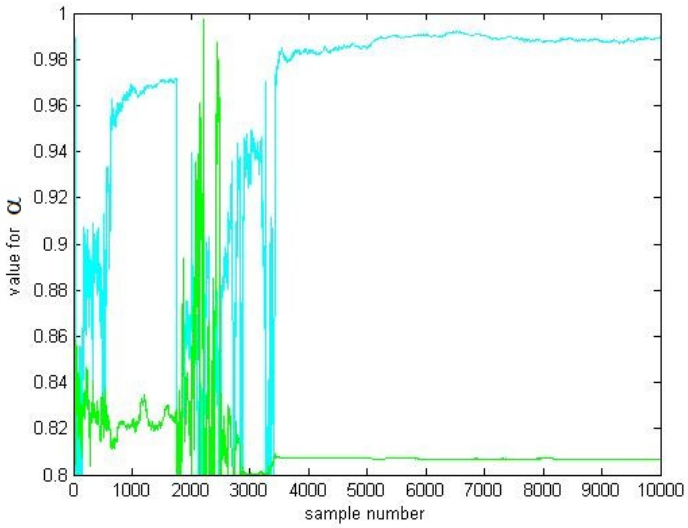
Figure 5.22 shows that one value improves much more than the other; as there is still clearly more noise on the estimate for 2.1991, when compared to its comparable value 1.7593.

This result was produced with $\mu_\beta = 0.1$, $\mu_\alpha = 0.03$, again γ was set to 0.8 for the adaptation of both α and β. Other constraints were as previously implemented for this approach.

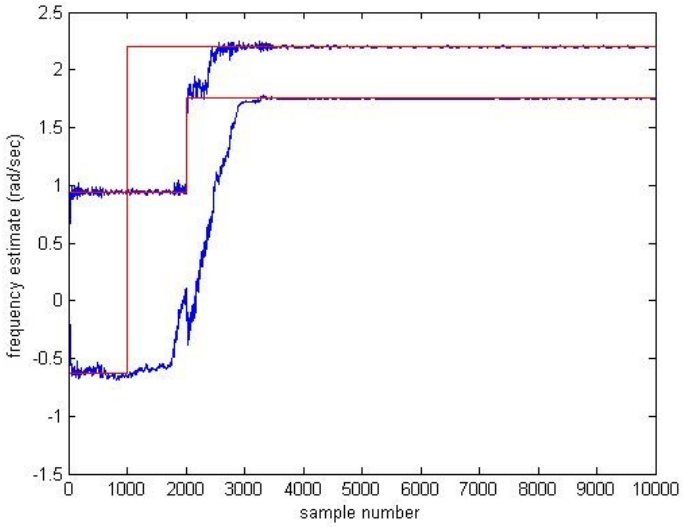
5.4.3 The Interconnected Parameter Approach for Tracking Two Complex Sinusoid Signals, whilst Adapting Unique α Values for each Signal

Implementing the interconnected parameter approach to track two CSSs, whilst updating multiple values of α can be achieved very simply, and produces pleasing results with a small amount of variance, which are shown in Figure 5.23; wherein (a) shows that as expected one value of α_1 converges to 0.99; however, interestingly the second value of α_2 is converging to 0.81. This convergence is peculiar, although produces quite a nice result as Figure 5.23 (b) shows, however the solutions are not converging to the exact true values, which is clarified in Table 5.6. Therefore, some accuracy in the final solution is lost with this approach.

This result was produced with $\mu_\beta = 0.1$, $\mu_\alpha = 0.15$, again γ was set to 0.8 for the adaptation of both α and β. Other constraints were as previously implemented for this approach.



(a) The adaptation of both α values in this CANF



(b) The tracking performance of the CANF

Figure 5.23. Tracking two frequencies whilst adapting separate α values with the interconnected parameter approach.

5.5 Discussions from Adapting the α and β Parameters Simultaneously

The results from these simulations are now discussed; where the results are embodied in tables, thus facilitating a clear comparison. Also, an analysis of

the multiple CSSs frequencies tracked in this chapter, is included in section 6.4.3 of this thesis; please also note that, figures for the additional computational complexity required to adapt the notch bandwidth parameter, are included in section 6.4.4 of this thesis.

5.5.1 Tracking a Single Complex Sinusoid Signal whilst also Updating α

Firstly, the results for tracking a single CSS whilst updating α, are summarised in Tables 5.3 and 5.4; and within the first Table 5.3 the final value tracked is -0.48, which when normalised by multiplying by 2π produces -3.0159: the value shown in the simulations.

Table 5.3. A comparison of methods for adapting α, with a final value of -0.48; wherein $\Delta\omega$ is the sample bias and σ^2 is the sample variance.

Method	a) Full gradient term	b) Heuristic simplification	c) Interconnected parameter
$\Delta\omega$	0.0001	0.0003	-0.0004
σ^2	5.8668×10^{-7}	4.0660×10^{-6}	5.1478×10^{-7}

It was observed that all the approaches, and particularly the heuristic approach did not adapt α as effectively when the target frequency is close to zero; therefore, an additional result is included as Table 5.4 to show a second set of results tracking -0.05, which when normalised is -0.3142.

Table 5.4. A comparison of methods for adapting α, with a final value of -0.05; wherein $\Delta\omega$ is the sample bias and σ^2 is the sample variance.

Method	a) Full gradient term	b) Heuristic simplification	c) Interconnected parameter
$\Delta\omega$	-0.0002	0.0005	-0.0003
σ^2	1.2722×10^{-6}	1.5679×10^{-5}	3.2575×10^{-6}

These results show that updating α, whilst tracking a single CSS does indeed generate improvements in the overall performance with all three adaptation methods for α. The full gradient approach is computationally the most

complex, as it requires an additional filter; however, this method achieves the best result, as there is less variance in the estimate. The interconnected approach provides a good simple solution, and the heuristic simplification creates some improvement reducing the variance slightly, for little additional computational complexity.

5.5.2 Tracking Two Complex Sinusoid Signals whilst Updating a Single Value of α in CNF_1

Firstly, the results are provided in Table 5.5 for tracking two CSSs, whilst updating a single value of α in the final section of a output-error structure, which is shown as CNF_1 in Figure 5.16. Please note, that the first value in Table 5.5 is for the original structure, where α is fixed at 0.8; and that the normalised values shown on the y-axis in the simulations, equate to 1.7593 for 0.28, and 2.1991 for 0.35.

Table 5.5. A comparison of methods for adapting α in CNF_1, whilst tracking two complex sinusoid signals; wherein $\Delta\omega_1$ & $\Delta\omega_2$ are sample biases and σ_1^2 & σ_2^2 are the sample variances; and a † labels a value smaller than 0.00005.

Method	The original structure	Adapting α via a) Full gradient term	Adapting α via b) Heuristic simplification	Adapting α via c) Interconnected parameter
$\Delta\omega_1$	0.0003	0†	0†	-0.0003
$\Delta\omega_2$	0.0010	0.0001	0†	0.0003
σ_1^2	0.0032	2.52×10^{-4}	0.0018	0.0026
σ_2^2	0.0013	7.52×10^{-4}	4.93×10^{-4}	0.0013

The results show that the full gradient approach achieves a definite improvement in both signals, whilst the heuristic approach only significantly improves one of the CSSs final variances; interestingly, the interconnected parameter approach shows little improvement in this scenario. Overall, some improvement can be achieved in adapting a single value of α, however, if this

is attempted the full gradient approach should generally be used.

5.5.3 Tracking Two Complex Sinusoid Signals, whilst Adapting Unique α Values for each Signal

In this section the results for tracking two CSSs, whilst updating separate values of α are detailed in Table 5.6; which also includes the original value for the structure, with a fixed value of α which is 0.8. Again, the normalised values shown on the y-axis in the simulations, equate to 1.7593 for 0.28, and 2.1991 for 0.35.

Table 5.6. A comparison of methods for adapting two α values, whilst tracking two complex sinusoid signals; wherein $\Delta\omega_1$ & $\Delta\omega_2$ are sample biases and σ_1^2 & σ_2^2 are the sample variances; and a † labels a value smaller than 0.00005.

Method	The original structure	Adapting α via a) Full gradient term	Adapting α via b) Heuristic simplification	Adapting α via c) Inter-connected parameter
$\Delta\omega_1$	0.0001	0†	0†	-0.001
$\Delta\omega_2$	0†	-0.0001	0.0004	0†
σ_1^2	0.0032	2.51×10^{-4}	0.0015	2.74×10^{-4}
σ_2^2	0.0013	1.06×10^{-4}	3.05×10^{-4}	5.34×10^{-4}

These results prove that the full gradient term produces the most accurate final solution, as although the interconnected parameter approach has achieved a similar variance, it is not precisely converging to the true value. However, if computation complexity is an issue both the heuristic simplification and interconnected approach should be considered. This result also demonstrates that, adapting multiple values of α does indeed improve the performance of a CANF when tracking multiple CSSs, however of course adding extra computational complexity[3].

[3]Figures for the additional computational complexity required to adapt the notch bandwidth parameter, are included in section 6.4.4 of this thesis.

5.5.4 Overall Observations for Adapting the Notch Bandwidth and Frequency Parameters Simultaneously whilst Tracking Hopping Complex Sinusoid Signals

Overall, it has been demonstrated that the performance of a CANF can be significantly improved if both the notch bandwidth and frequency parameters are updated simultaneously when tracking hopping CSSs, although this can significantly increase the complexity of the design. Adapting the notch bandwidth and frequency parameters also improves the ability of this structure for the tracking of CSSs which are further apart.

Three methods for updating the notch bandwidth parameter have been evaluated, firstly for tracking a single hopping CSS; then tracking two hopping CSSs whilst adapting a single notch bandwidth parameter for the overall structure, lastly individual notch bandwidth parameters were updated for each hopping CSS; this research also shows the strengths and weaknesses for each of the three approaches.

It has also been shown that a method of steepest ascent is required for the adaptation of the notch bandwidth parameter α.

5.6 Summary

This chapter considers adapting the notch bandwidth and frequency parameters simultaneously in a CANF. Three methods of adapting the notch bandwidth parameter have been demonstrated in this chapter, along with schemes for tracking two CSSs whilst updating an overall notch bandwidth parameter for the complete structure, or for each frequency which is being tracked.

In the next chapter of this thesis, tracking CVCSs is investigated; initially with only the parameter β adapting to track the signal, i.e. with a fixed notch bandwidth.

Chapter 6

TRACKING A COMPLEX-VALUED CHIRP SIGNAL

This chapter considers tracking a Complex-Valued Chirp Signal (CVCS), and begins with an introduction and initial results for tracking a CVCS; then the notch bandwidth parameter is adapted, as in the previous chapter: to attempt to improve the results further. Next, the chapter moves onto tracking a CVCS and a hopping CSS simultaneously; and lastly it includes frequency plots for the CSSs tracked in Chapters 4 to 6 of this thesis, and the computational complexity of adapting the notch bandwidth parameter when tracking one or two CSSs; before concluding with a discussion of the results and a summary.

So to begin with, the introduction defines a CVCS, then moves onto initial results; which compare Regalia's structure to the scheme developed in Chapter 4 of this thesis.

6.1 Introduction and Initial Results

In Regalia's paper [23], he demonstrated that the complex version of his design is capable of tracking a CVCS, which in this case is a quadratically

varying frequency of the form

$$u(n) = e^{j\phi(n)} + b(n); \tag{6.1.1}$$

wherein

$$\phi(n) = \phi_2 n^2 + \phi_3 n^3; \tag{6.1.2}$$

where $\phi_2 = -0.004$ and $\phi_3 = 1.2 \times 10^{-6}$.

Regalia's result, which is labelled as Fig. 3 in [23], has been included in this chapter as Figure 6.1, noting that if the NLMS algorithm is used to produce this simulation, a similar result can be achieved to the LMS algorithm's result; with the exception that the gain should be increased. Therefore, Regalia specifies that μ should be 0.02 for the LMS variant; however if the NLMS learning algorithm is selected μ should be increased to 0.08, and the value required for γ is 0.8.

Equivalently a comparable result may be produced with the proposed structure: which is shown in Figure 6.2; however, to produce this result, as the plotted value is the angle of the complex result; a phase shift of -2π is required when the plotted value exceeds one. Figure 6.2 was achieved by using the following parameters: $\mu = 0.15$, $\gamma = 0.8$ and $\alpha = 0.9$.

It is clearly visible when comparing Figure 6.1 with Figure 6.2, that both structures produce similar results. Please note that measured values for the simulations in sections 6.1 and 6.2 of this chapter are included in Table 6.1, which is located in the discussions section of this chapter: section 6.4.1.

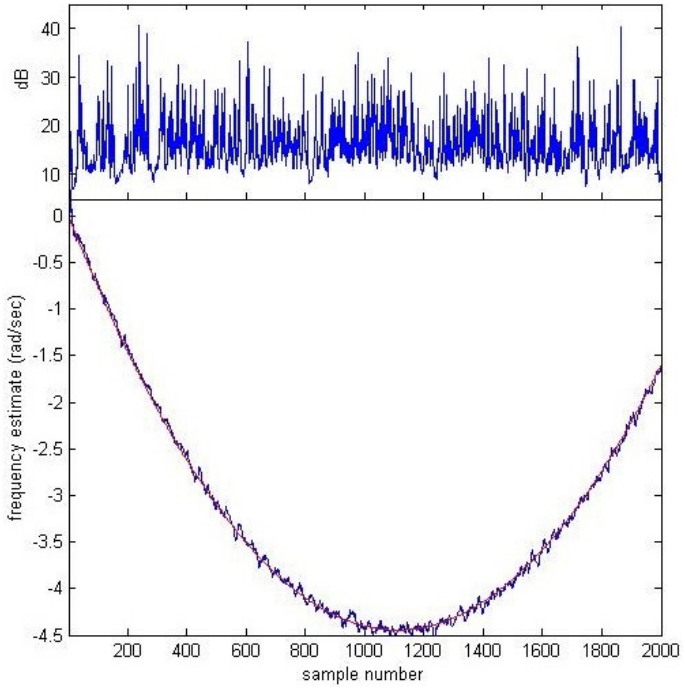


Figure 6.1. Tracking a complex-valued chirp signal with Regalia's structure.

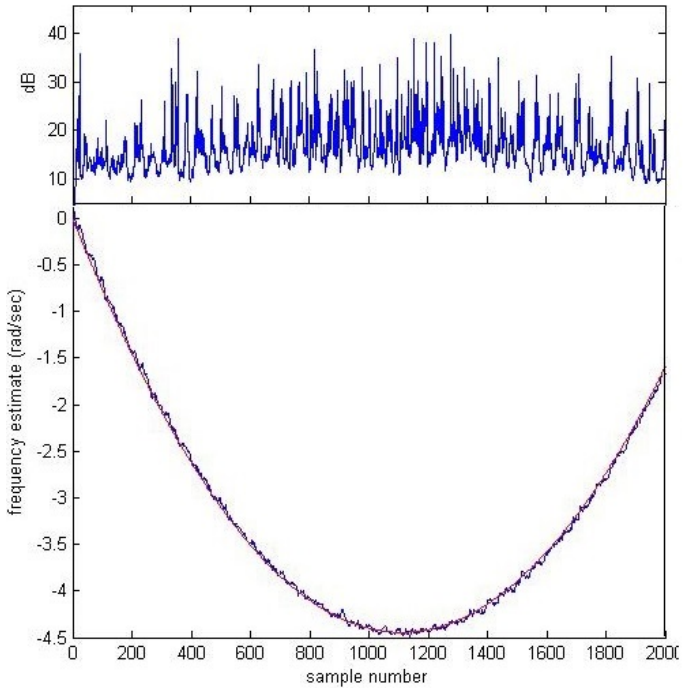


Figure 6.2. Tracking a complex-valued chirp signal with the proposed structure.

To create a full evaluation of tracking a CVCS, the full gradient term for β is also evaluated, to investigate if this outperforms the simplified term when tracking a CVCS. Please note that, the full gradient term for β is derived in Appendix 9.2 of this thesis.

In this analysis more noise was also added, to investigate if the full gradient term improved results in noisier environments, when tracking a CVCS. It was noted that the simplified β term and the full gradient term for β both tracked a CVCS for SNR up to approximately -12.5 dB, then both approaches struggled beyond this value.

The simulation of tracking a CVCS with the full gradient term for β is shown in Figure 6.3; which was produced with the following parameters: μ = 0.3, γ = 0.9 and α = 0.9.

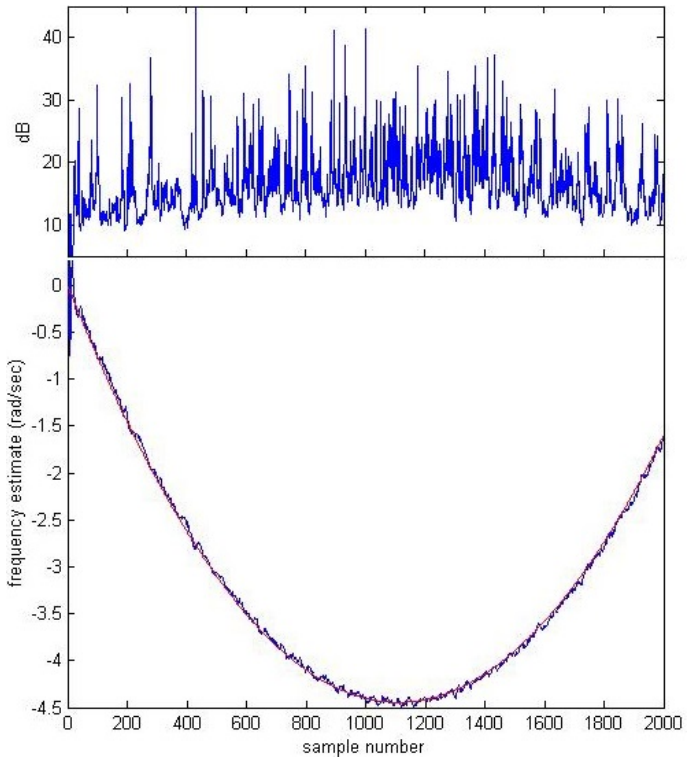


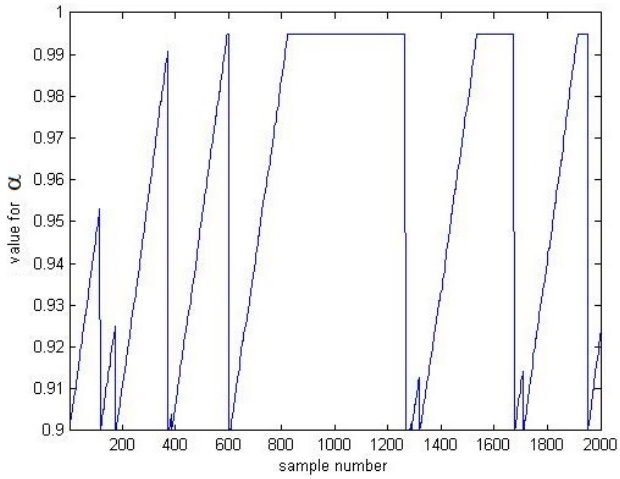
Figure 6.3. Tracking a complex-valued chirp signal with the full gradient term for β, within the proposed structure.

6.2 Tracking a Complex-Valued Chirp Signal whilst Adapting α

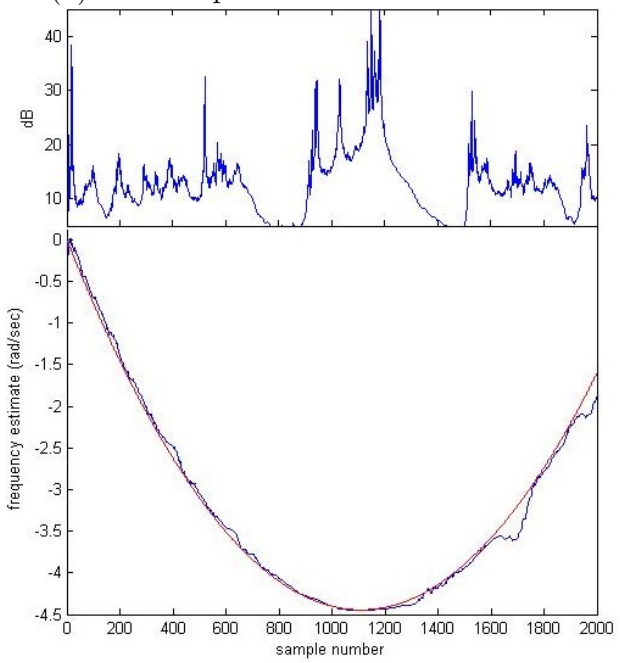
Following on from the result shown in Figure 6.2, and applying the simplified update for β; the variations of this structure which update α are simulated, to investigate if adapting α improves the results further. Again, this has been investigated for the three viable approaches, where firstly the full gradient approach is considered. Please note that measured values for the simulations in this section are included in Table 6.1, which is located in the discussions section of this chapter: section 6.4.1.

The Full Gradient Approach for Adapting α, whilst Tracking a Complex-Valued Chirp Signal

In this section of the chapter a CVCS is tracked with the full gradient approach for adapting α; the result to this simulation is shown in Figure 6.4. Wherein, the simulation was produced with the following parameters: $\mu_\beta = 0.1$, $\mu_\alpha = 0.0008$, $\gamma_\alpha = 0.9$ $\gamma_\beta = 0.7$. The parameter α was initialised at 0.9; and was constrained with the same method used previously for this approach.



(a) The adaptation of α in this CANF



(b) The tracking performance of the CANF

Figure 6.4. Tracking a complex-valued chirp signal whilst adapting α with the full gradient approach.

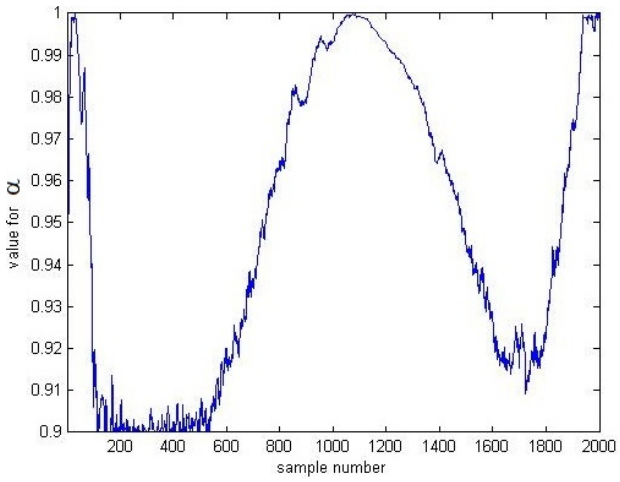
Please note, that in these simulations for tracking a CVCS, the lower limit for α has been increased to 0.9; which is the value Regalia applied in his publication [23].

The results show that, as α increases the noise reduces; however, the target signal is then lost so α is reset. Therefore, this does not produce an ideal result, as the overall variance from the target signal is increased when the tracking drifts, thus little improvement is shown in the overall variance estimates.

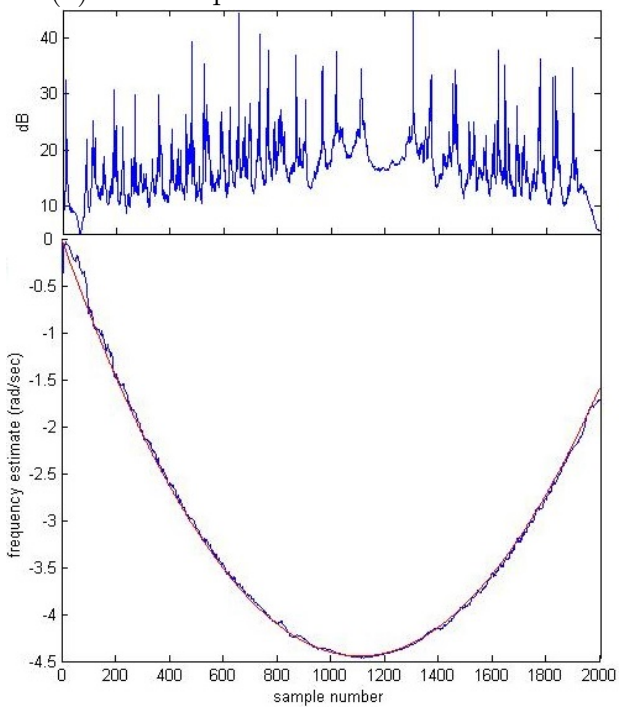
The Heuristic Approach for Adapting α, whilst Tracking a Complex-Valued Chirp Signal

In this section, the heuristic approach for updating α is trialled and the results for updating α whilst tracking a CVCS are shown in Figure 6.5; which shows that some improvement is achieved, once α exceeds the value of 0.9, which occurs at approximately 700 samples. However, the value of α does not fully stabilise with this method.

Figure 6.5 was produced with the following parameters: $\mu_\beta = 0.15$, $\mu_\alpha = 0.05$, $\gamma = 0.8$ for both the adaptation of α and β. The parameter α was initialised at 0.8; and was constrained with the same method used previously for this approach.



(a) The adaptation of α in this CANF

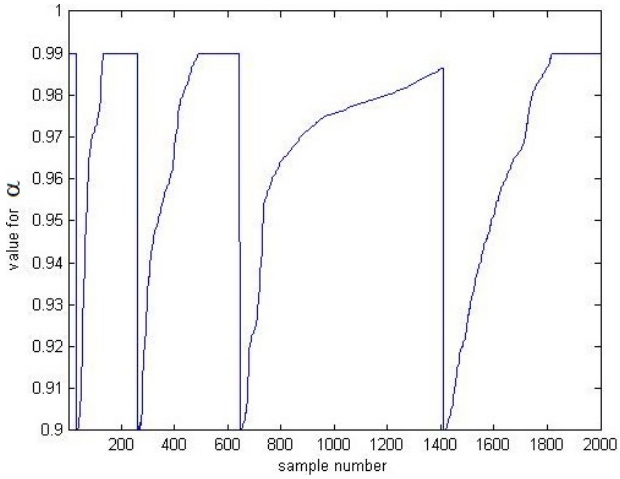


(b) The tracking performance of the CANF

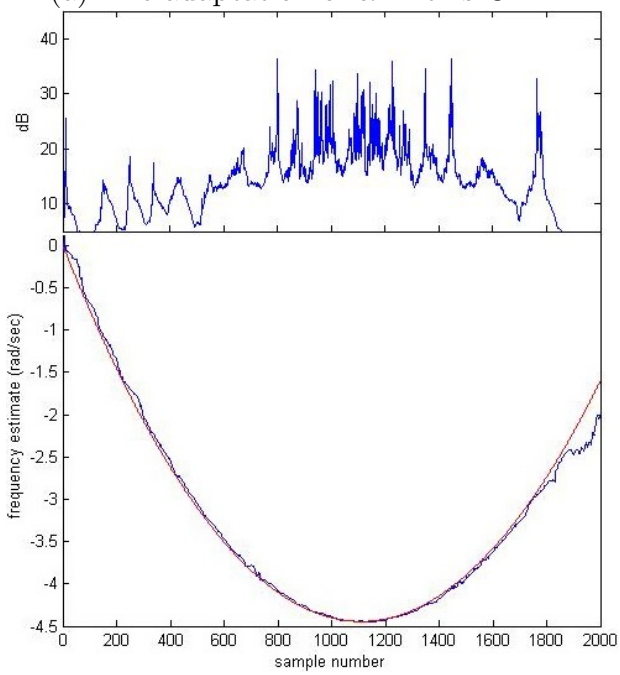
Figure 6.5. Tracking a complex-valued chirp signal whilst adapting α with the heuristic approach.

The Interconnected Parameter Approach for Adapting α, whilst Tracking a Complex-Valued Chirp Signal

As the heuristic approach for adapting α was capable of tracking a CVCS, now the interconnected parameter approach is investigated. Therefore, Figure 6.5 shows the results for updating α, whilst tracking a CVCS, with the interconnected parameter approach.



(a) The adaptation of α in this CANF



(b) The tracking performance of the CANF

Figure 6.6. Tracking a complex-valued chirp signal whilst adapting α with the interconnected parameter approach.

The result shows much improved tracking performance, except when α is constrained, i.e. being reduced to 0.9; as when this occurs the performance drops, which is apparent at samples 210, 650 and 1400. However, if these constraints are removed, the notch can loose the target signal altogether.

Figure 6.5 was produced with the following parameters: $\mu_\beta = 0.4$, $\mu_\alpha = 0.1$, $\gamma = 0.8$ for both the adaptation of α and β. The parameter α was initialised at 0.9; and was constrained with the same method used previously for this approach.

To summarise, these results demonstrate that a CVCS can be tracked with this proposed structure, although no improvement is gained by updating the notch bandwidth parameter.

6.3 Tracking a Chirp and a Hopping Signal, whilst Adapting Multiple α Values

As little improvement was produced when tracking a CVCS and adapting α, in this section tracking a CVCS at the same time as a frequency hopping CSS is investigated.

To begin with, tracking a CVCS and a hopping CSS without adapting α is considered, where the CVCS was produced as in (6.1.1), with the following parameters

$$\phi(n) = -0.0005n^2 + 0.1 \times 10^{-6}n^3. \tag{6.3.1}$$

Whilst the hopping signal initialises at 0.2, then hops to -0.2 at 1000 samples; the result for this simulation is shown below in Figure 6.7.

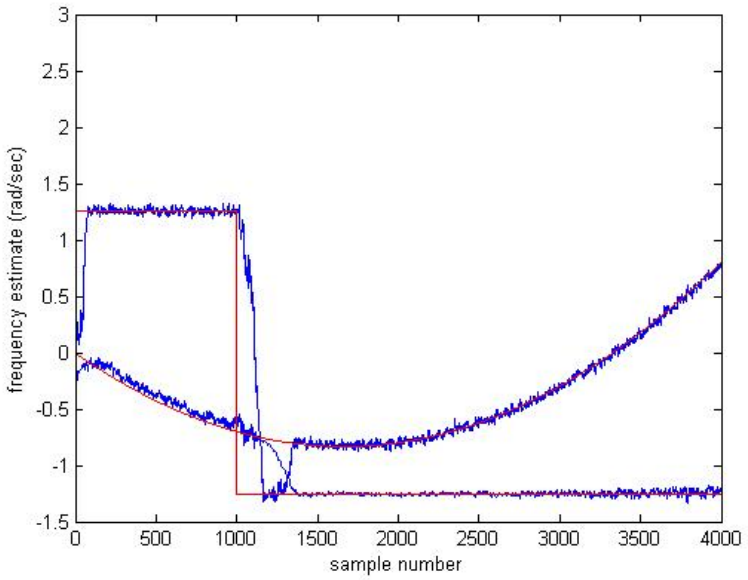


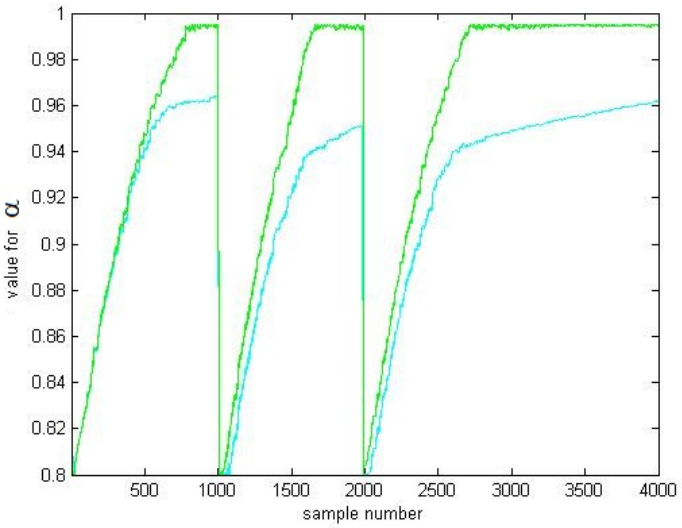
Figure 6.7. Tracking a complex-valued chirp signal and a hopping complex sinusoid signal simultaneously.

This provides a satisfactory result, which can be used as a starting point for comparison. As before, measured values are included in Table 6.2, which is located in the discussions section of this chapter: section 6.4.2.

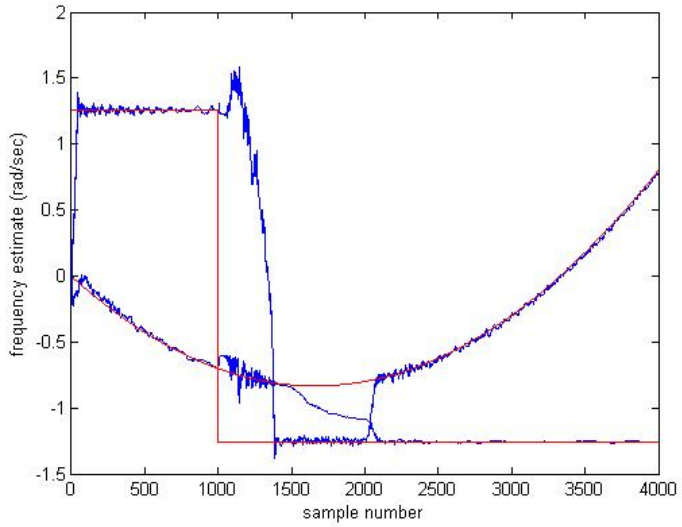
6.3.1 The Full Gradient Approach for Adapting α values, whilst Tracking a Complex-Valued Chirp Signal and a Hopping Complex Sinusoid Signal Simultaneously

In this first simulation two values of α are adapted by applying the full gradient approach, whilst simultaneously tracking a CVCS and a hopping CSS.

The result is shown as Figure 6.8, where the following parameters were used: $\alpha_1(0) = 0.8$, $\alpha_2(0) = 0.8$, $\mu_\beta = 0.1$, $\mu_\alpha = 0.001$, $\gamma_\beta = 0.9$ and $\gamma_\alpha = 0.8$.



(a) The adaptation of both α values in this CANF



(b) The tracking performance of the CANF

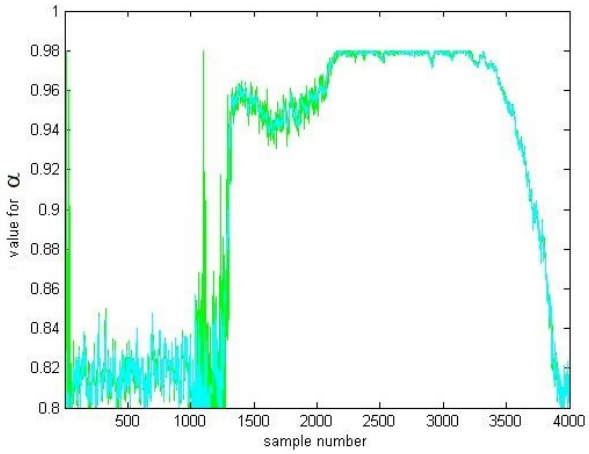
Figure 6.8. Tracking a complex-valued chirp signal and a hopping complex sinusoid signal simultaneously, whilst adapting multiple values of α with the full gradient approach.

Please take note that, the CVCS selected for this simulation has a lower gradient than the CVCS that was tracked in the previous section.

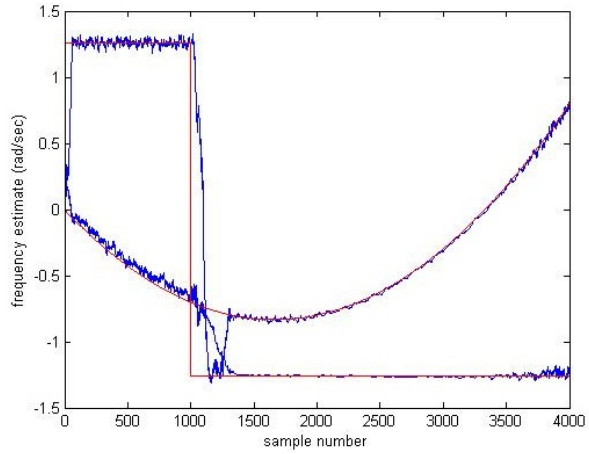
Figure 6.8 shows a pleasing result, as the amount of noise is clearly reducing on both signals.

6.3.2 The Heuristic Simplification Approach for Adapting α Values whilst Tracking a Complex-Valued Chirp Signal and a Hopping Complex Sinusoid Signal Simultaneously

In this second simulation two values of α are adapted via the heuristic simplification method, whilst tracking a CVCS and a hopping CSS simultaneously. This result is shown as Figure 6.9, where the following values were applied: $\alpha_1(0) = 0.8$, $\alpha_2(0) = 0.8$, $\mu_\beta = 0.1$, $\mu_\alpha = 0.015$ and $\gamma = 0.8$ for the adaptation of α and β.



(a) The adaptation of both α values in this CANF



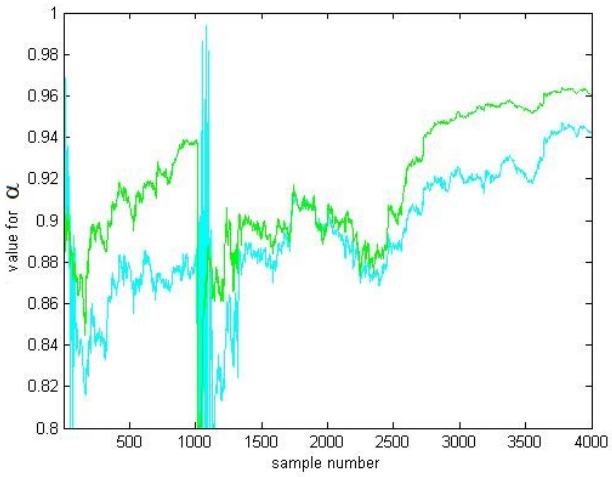
(b) The tracking performance of the CANF

Figure 6.9. Tracking a complex-valued chirp signal and a hopping complex sinusoid signal simultaneously, whilst adapting multiple values of α with the heuristic approach.

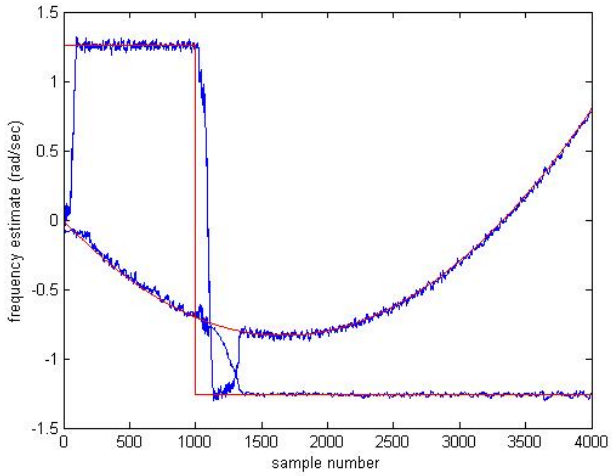
Figure 6.9 shows some success, however, both α values drift towards the end of the simulation i.e. after 3250 samples, thus the heuristic approach may have limited performance for tracking a CVCS and a hopping CSS simultaneously.

6.3.3 The Interconnected Parameter Approach for Adapting α Values, whilst Tracking a Complex-Valued Chirp Signal and a Hopping Complex Sinusoid Signal Simultaneously

In the last result of this chapter, adapting two values of α with the interconnected parameter approach whilst tracking a CVCS and a hopping CSS simultaneously is investigated. This result is shown as Figure 6.10, where the following parameters were set: $\alpha_1(0) = 0.8$, $\alpha_2(0) = 0.8$, $\mu_\beta = 0.1$, $\mu_\alpha = 0.15$ and $\gamma = 0.9$ for the adaptation of both α and β.



(a) The adaptation of both α values in this CANF



(b) The tracking performance of the CANF

Figure 6.10. Tracking a complex-valued chirp signal and a hopping complex sinusoid signal simultaneously, whilst adapting multiple values of α with the interconnected parameter approach.

Figure 6.10 shows a working solution, as the amount of noise is clearly reducing on both signals; however, both α values are only converging to around 0.94, which is not very close to unity; therefore, the variance will not be significantly reduced.

6.4 Discussions from Tracking a Complex-Valued Chirp Signal

This chapter now concludes, with discussions from this chapter on tracking a CVCS; along with the computation complexity of the algorithm variants for adapting both the notch bandwidth and frequency parameters.

6.4.1 Tracking a Complex-Valued Chirp Signal

This section tabulates the results for tracking a CVCS with Regalia's approach; then equivalently with the proposed structure, firstly with the simplified gradient term, then in a full gradient form[1]. Next, α is adapted for the results a) to c), utilising the successful methods that have previously been applied for adapting α when tracking CSSs.

Table 6.1. A comparison of methods for tracking a complex-valued chirp signal; wherein σ^2 is the sample variance.

Method	Regalia 2010	The original structure	Full gradient for β structure
σ^2	0.0026	0.0017	0.0016
Method	Adapting α via a) Full gradient term	Adapting α via b) Heuristic simplification	Adapting α via c) Interconnected parameter
σ^2	0.0074	0.0029	0.0036

The results show that, the proposed structure produces slightly superior results than Regalia's equivalent structure; although, adapting α via these trialled methods whilst tracking a CVCS, provides no real improvement.

Please note, that when tracking a CVCS the variance has been calculated from the 50th to the final sample; this is due to the fact that all the structures perform well over certain parts of the CVCS, therefore, to achieve a true comparison the variance for the complete CVCS simulation should be considered.

[1]The 'full gradient form' for β is derived in Appendix 9.2.

6.4.2 Tracking a Complex-Valued Chirp Signal and a Hopping Complex Sinusoid Signal Simultaneously whilst Adapting Multiple α Values

This section tabulates the results for tracking a CVCS and a hopping CSS simultaneously, firstly by only adapting the β parameter to track the signals. Then, separate values of α are adapted via methods a) to c), to evaluate if by simultaneously updating α, whilst still tracking signals with β, will instantiate further improvements in the results.

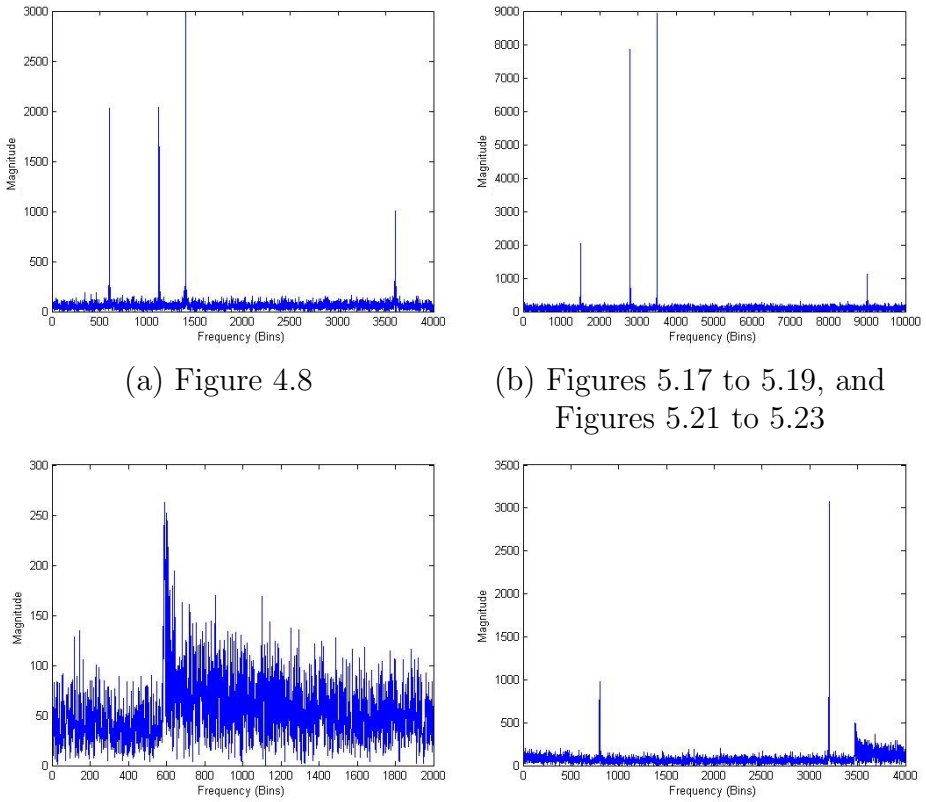
Table 6.2. A comparison of methods for adapting two α values, whilst tracking a complex-valued chirp signal and a hopping complex sinusoid signal; wherein σ_1^2 & σ_2^2 are the sample variances.

Method	The original structure	Adapting α via a) Full gradient term	Adapting α via b) Heuristic simplification	Adapting α via c) Inter-connected parameter
σ_1^2	7.02×10^{-4}	8.96×10^{-6}	7.82×10^{-4}	3.83×10^{-4}
σ_2^2	5.76×10^{-4}	6.45×10^{-5}	2.07×10^{-4}	1.39×10^{-4}

Table 6.2 demonstrates that the full gradient approach generates a significant improvement to both signals; whilst the heuristic approach struggles in this scenario: which is clear from 3500 to 4000 samples in the simulation; the interconnected approach generates some improvement reducing the noise on both signals by 50%.

6.4.3 Analysis of the Complex Sinusoid Signals Frequencies that have been Tracked

A further analysis of the multiple CSS frequencies tracked in the simulations from chapters 4-6 of this thesis is shown in Figure 6.11.



(a) Figure 4.8

(b) Figures 5.17 to 5.19, and Figures 5.21 to 5.23

(c) Figures 6.1 to 6.3

(d) Figure 6.10

Figure 6.11. Frequency analysis of the complex sinusoid signals tracked in chapters 4-6 of this thesis.

These results are included for completeness, to show any relationship in the selected frequencies: which may be useful in further analysis of convergence.

6.4.4 Computational Complexity of the Algorithms, including the Additional Calculations Required when Adapting α

This section contains the number of calculations required to track CSSs, whilst adapting parameters for all the scenarios considered in this chapter, and is also applicable to Chapter 5.

Therefore, tracking one and two CSSs is considered, whilst adapting a single value of α in CNF_1; then updating a values of α, for each CSS being

tracked i.e. updating multiple α parameters.

Firstly, Table 6.3 includes the complete number of calculations required for tracking a single CSS, first with α fixed: the original structure from Chapter 4; then with α simultaneously updating with β, via the three successful methods a) to c).

Table 6.3. The complete complexity for adapting α whilst tracking a complex sinusoid signal.

Method	$\div s$	$\times s$	$+$s
The original structure	2	10	8
Adapting α a) Full gradient approach	4	19	15
Adapting α b) Heuristic simplification	3	15	11
Adapting α c) Interconnected approach	4	12	10

Please note that constraints have not been included in these calculations, however, would add minimal computational complexity.

Secondly, Table 6.4 contains the additional complexity required to adapt α when tracking a CSS for the three successful methods.

Table 6.4. The additional complexity for adapting α whilst tracking a complex sinusoid signal.

Method	$\div s$	$\times s$	$+$s
Adapting α a) Full gradient approach	2	9	7
Adapting α b) Heuristic simplification	1	5	3
Adapting α c) Interconnected approach	2	2	2

Next, Table 6.5 includes the complete complexity required to adapt a single value of α when tracking two CSSs.

Table 6.5. The complete complexity for adapting a single value of α whilst tracking two complex sinusoid signals.

Method	$\div s$	$\times s$	$+$s
The original structure	4	24	21
Adapting α a) Full gradient approach	6	33	28
Adapting α b) Heuristic simplification	5	30	26
Adapting α c) Interconnected approach	6	26	23

Lastly, Table 6.6 shows the complete complexity required to adapt multiple values of α when tracking two CSSs.

Table 6.6. The complete complexity for adapting two values of α whilst tracking two complex sinusoid signals.

Method	$\div s$	$\times s$	$+$s
Adapting α a) Full gradient approach	8	43	42
Adapting α b) Heuristic simplification	6	34	28
Adapting α c) Interconnected approach	8	29	27

Table 6.6 demonstrates that adapting multiple values of α with the full gradient approach effectively doubles the computational complexity of the design.

Overall, these results show that if computational complexity is an issue in a specific application where updating α would be an advantage; then it may be worth considering the heuristic simplification or interconnected approaches, as the full gradient approach adds significant complexity particularly when tracking multiple CSSs.

6.4.5 Overall Observations for Tracking a Complex-Valued Chirp Signal

Lastly, some overall observations for tracking a CVCS are highlighted. This chapter has demonstrated the strong performance of this design for tracking a CVCS, and compared the approach to Regalia's scheme [23]

The chapter highlighted that it is possible to produce a full gradient term for the adaptation of β: using the method applied to derive the full gradient term for the notch bandwidth parameter: α, which may improve the results in tracking signals for certain scenarios, as this approach improved the results slightly for tracking a CVCS.

The results demonstrate that no real improvement is achieved, from updating both the α and β parameters when tracking a CVCS, with the meth-

ods evaluated herein; although some improvement was demonstrated, when tracking a less severe CVCS and a hopping CSS simultaneously.

The computational complexity for the proposed design with a fixed notch bandwidth parameter, and an adapting notch bandwidth parameter, for the tracking of one CSS and two CSSs, has also been included in this chapter; and it has been shown that adapting the notch bandwidth parameter can double the complexity of the design.

6.5 Summary

This chapter has demonstrated that the CANF structure developed earlier in this thesis is capable of tracking a CVCS, and has compared its performance to Regalia's scheme.

It has also considered adapting the notch bandwidth parameter when tracking a CVCS, and when tracking a hopping CSS and a CVCS; and that adapting the notch bandwidth parameter can facilitate a reduction in variance when tracking a hopping CSS and a CVCS.

Lastly, the chapter included frequency plots for the CSSs tracked in Chapters 4 to 6 of this thesis; then embodied the computational complexity of the approaches applied in these chapters into tables for convenient reference.

This concludes this section of research, and the next chapter of this thesis moves on to consider stochastic search methods, with the aim of facilitating faster convergence to an unknown target frequency after a sudden change, or frequency hop.

Chapter 7

STOCHASTIC SEARCH APPROACH TO LEARNING IN COMPLEX ADAPTIVE NOTCH FILTERS

7.1 Introduction

Historically, there are two major classes of optimisation algorithms; these are calculus-based techniques and enumerative techniques. The earlier chapters of this thesis have focused on calculus-based optimisation techniques, and employ the gradient-directed searching mechanism to minimise an error, which utilises a cost function. However, as local optima are frequently obtained, enumerative techniques such as dynamic programming can be applied, although this technique may break down on complex problems of moderate size. A third approach is genetic algorithms, where its associated algorithms are: simulated annealing [60], evolutionary strategies [61], and evolutionary programming [62] - [63]; and these types of approaches are classified as stochastic search or guided random techniques.

This chapter firstly introduces genetic algorithms, then specifically Particle Swarm Optimisation (PSO): which is the stochastic technique implemented in this chapter. Next it demonstrates how PSO can be exploited in a CANF, firstly in a pure form for tracking a single CSS; then in a hybrid form, for reinitialisation after a hop when tracking two CSSs, and the further refinements required in this scenario are discussed. Lastly, the chapter is concluded and summarised.

7.1.1 An Overview of Genetic Algorithms

Essentially, the principle of a genetic algorithm is to assess a range of values, where the values that provide the strongest solution succeed, as is the case in evolution. Genetic algorithms utilise the chromosomes of candidates, which crossover and mutate at recombination; and this principle can be applied to the adaptation of parameters in a man made system. Figure 7.1 illustrates the concept of using an evolutionary algorithm to solve a problem, and this illustration was obtained from [64].

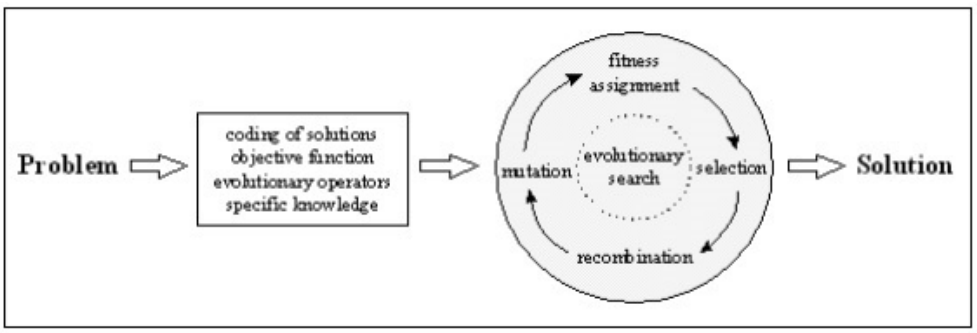


Figure 7.1. The concept of a problem solution using evolutionary algorithms [64].

Figure 7.1 shows the stages of a genetic algorithm to be: fitness assignment, selection, recombination and mutation. This approach can and has been applied to learning algorithms for some time, and a book published in 1975 entitled 'Adaptation in natural and artificial systems' [65], high-

lights the idea of applying the process observed in natural evolution to the learning of a system. An early reference [66], which was published in 1990, describes the applications of genetic algorithms to adapt intelligent systems, and defines applications for genetic algorithms in: neural networks, adaptive rule-based optimisation of combustion in multiple burner installations in the steel industry, and controlling a dynamical system.

It should also be acknowledged that the comprehensive article [67], provides an excellent source of background information on genetic algorithms; and highlights the application of genetic algorithms to: IIR adaptive filtering, non-linear model selection, time-delay estimation, active noise control and speech processing.

7.1.2 An Overview of Particle Swarm Optimisation

After this brief overview of genetic algorithms, swarm optimisation techniques are introduced; as this is the stochastic technique that has been implemented in this chapter; due to the fact that the potential solutions evolve in a more collective swarm like manner, to yield an improved solution.

Genetic algorithms consider chromosomes, whilst swarm optimisation techniques consider a distribution of particles; where both techniques are utilising their best results to converge upon the optimum solution.

In 1992 papers began to be published on swarm intelligence such as [68], although the concept of PSO was published in 1995 by Kennedy and Eberhart [69]. They considered applications such as: modelling human social behaviour, fish schooling and bird flocking.

The basic principle of PSO can be described as follows; initially you start with a distribution of particles, which are spread over the full potential solution space; where the initial distribution over the solution space should be referred to as the initialisation of the swarm. Then each of these particles position is passed through a fitness function to assess how well it performs, thus providing an overall 'best' solution.

Once the set of particles has been evaluated, each particle's velocity is updated so that it moves towards the 'best' solution, where each particle considers its best position and the overall best position of the swarm; therefore, global sharing of information takes place herein. Then the speed of the convergence is controlled via weighting parameters, which vary depending on which PSO algorithm is selected. Thus, each particle essentially moves towards the 'best' solution after each update.

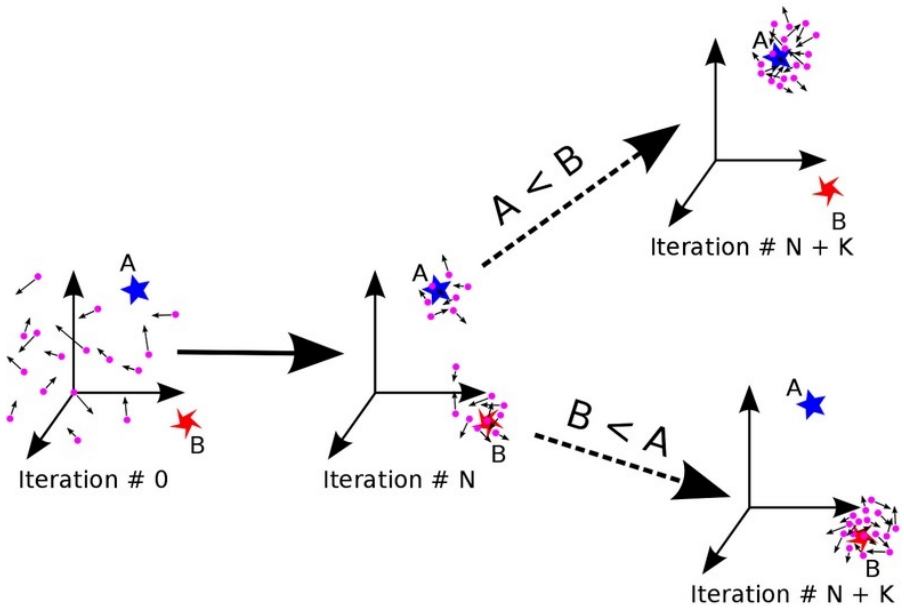


Figure 7.2. The concept of particle swarm optimisation [70].

The concept of PSO is shown in Figure 7.2, which illustrates a three dimensional problem; and was acquired from [70]. It shows a swarm initialisation at Iteration # 0, then at Iteration # N the swarm has split between

two points A and B, and at Iteration # N+K the two scenarios where $A < B$ and $B < A$ are shown; which illustrates the swarm converging on the most likely position overall. Generally, PSO works well in applications seeking an optimum position with two or three dimensions, which can be located over multiple time intervals; therefore, since a CANF has a 'best real position' and a 'best imaginary position' i.e. two dimensions, PSO is well suited to this particular application.

PSO is normally implemented by stopping the algorithm when a 'stop condition' is reached, or after a given number of samples; as in normal operation the computational complexity of this type of approach is high. Therefore to reduce the overall complexity, hybrid implementations such as [71] - [73] have been published, where these methods combine a genetic algorithm with a gradient descent approach; which will provide extra performance when required.

In many publications such as [74] and [75], the PSO process is shown with a flowchart, and this has been included in this chapter as Figure 7.3; which illustrates the standard PSO process.

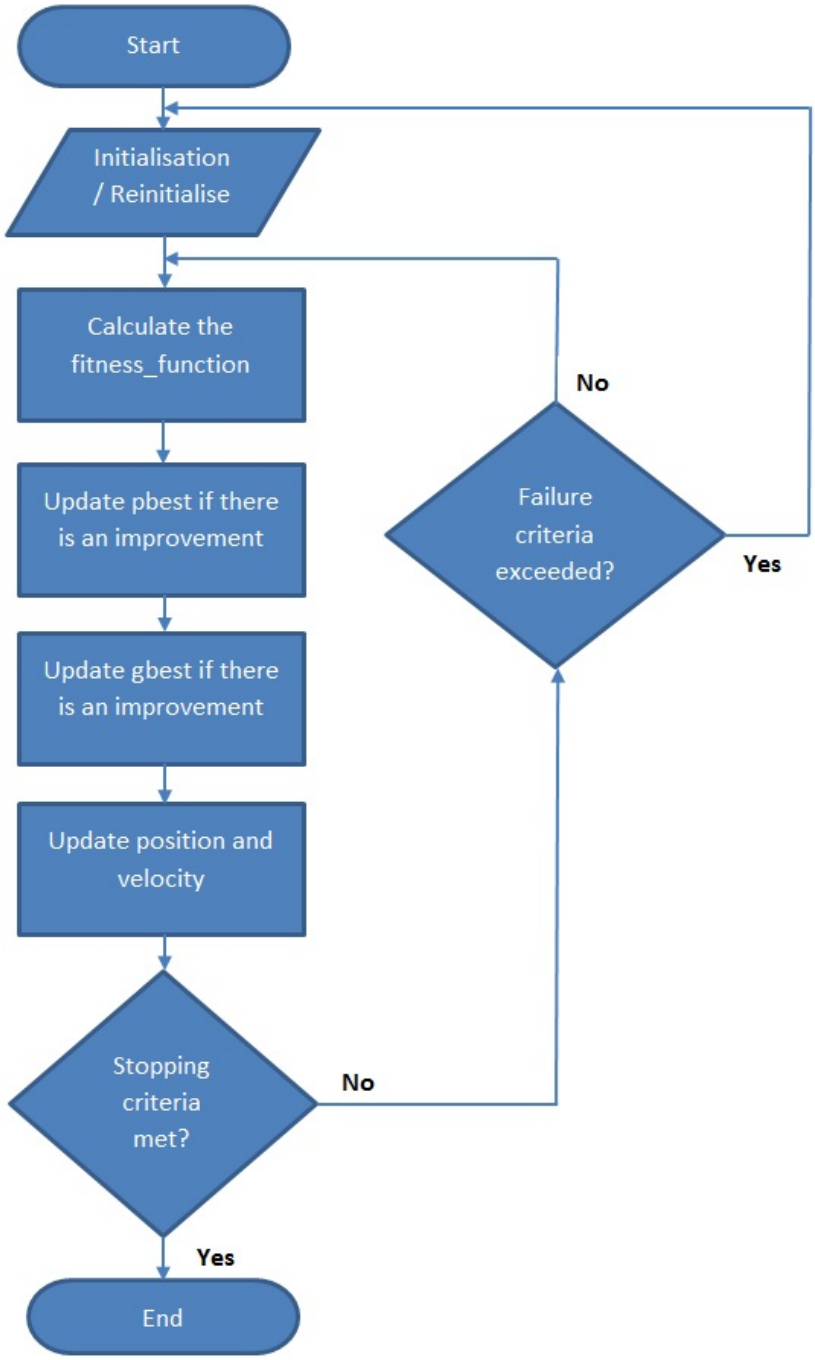


Figure 7.3. A flowchart representing the particle swarm optimisation process; where pbest is the current best position of an individual particle, and gbest is the current best position in the swarm.

One specific weakness of stochastic search approaches is that methods suffer from a delay, and this delay is due to fitness calculation for each value or particle that is being evaluated.

7.2 Particle Swarm Optimisation Theory

Now to describe PSO in more detail, the notations are defined next. These definitions begin with (i) which is used to denote each particle in the swarm, therefore it can take the values $i = 1, 2, ..., N$; in a swarm size of dimensions D, thus $d = 1, 2, ..., D$. The coordinates of each particle in the swarm is represented with the vector $X_i = (x_{i1}, x_{i2}, ...x_{iD})$, and the particle with the 'global best' value is denoted by the index g. The best previous position of each particle in the swarm is represented with the vector $P_i = (p_{i1}, p_{i2}, ...p_{iD})$, and the velocity of the i-th particle is $V_i = (v_{i1}, v_{i2}, ...v_{iD})$. It should be noted that, each swarm is evaluated for the time interval (n).

The PSO works by calculating the fitness of each particle, i.e. how well its position performs. Therefore, every PSO application must have a 'fitness function', which will indicate how well the position performs. The fitness function is defined as

$$fitness\_function(i) = f_x(i); \tag{7.2.1}$$

wherein, f_x is a specific function; and the target would normally be seeking a minimum of this function.

In PSO the particles velocities are updated by the equation

$$v_i(n)_{id} = \iota v_i(n-1)_{id} + rand.c_1(p_i(n-1)_{id} - x_i(n-1)_{id}) + rand.c_2(p_i(n-1)_{gd} - x_i(n-1)_{id}); \tag{7.2.2}$$

then the position of each particle is adjusted by the update

$$x_i(n)_{id} = x_i(n-1)_{id} + \chi v_i(n)_{id}; \tag{7.2.3}$$

where, c_1 and c_2 are two positive constants, χ is the constriction factor, used to control and constrict the velocities, $rand$ is a random number uniformly

distributed in the range [0,1], and ι is the inertia weight.

Considering (7.2.2) further, the first term $\iota v_i(n-1)_{id}$ is the particle's previous velocity weighted by the inertia weight ι; the second term, $(p_i(n-1)_{id} - x_i(n-1)_{id})$, is the distance between the particle's best previous position, and its current position; then the third term, $(p_i(n-1)_{gd} - x_i(n-1)_{id})$, is the distance between the swarm's best experience, and the i-th particle's current position. The parameters $rand.c_1$ and $rand.c_2$ provide randomness that make the technique more flexible but less predictable.

As described in [76], the constriction factor may be calculated as

$$\chi = \frac{1}{2 - c - \sqrt{c^2 - 4.c}}; \tag{7.2.4}$$

herein, $c = c_1 + c_2$, and $c > 4$. Additionally, [76] also highlights that it is also possible to update the inertia weight by finding the variance of the population fitness as

$$\sigma^2 = \sum_{i=1}^{N} \left(\frac{f_i - f_{avg}}{f} \right)^2; \tag{7.2.5}$$

wherein, f_i is the fitness of the i_{th} particle, f_{avg} is the average fitness of the population of particles, and f is a normalising factor used to limit the variance.

7.3 Applying Particle Swarm Optimisation to a Complex Adaptive Notch Filter

In this section, PSO is exploited within a CANF; and this section begins with a literature review of research relevant to this topic.

7.3.1 Tracking a Complex Sinusoid Signal whilst applying Particle Swarm Optimisation in a Complex Adaptive Notch Filter

Genetic algorithms have been implemented in ANFs for some time, and in the 1990s, Cain et al. adopted what they refer to as Darwinian adaptation in [77] and [78], where they essentially evaluated values related to the parameters of a notch filter, calculated each value's fitness, then converged upon the most successful value i.e. survival of the fittest, which is implemented by narrowing the width of a distribution: over several iterations. Their method adapts both the pole radius and frequency parameter in an ANF; however, this approach requires access to a separate interference signal, which is quite artificial and is unlikely to be possible in practice; thus this approach has not been considered in this chapter. It should also be noted that in [78], Cain et al. apply their approach to tracking two hopping CSSs.

PSO has previously been applied to adaptive filtering, where an example of one of these publications is [74], which appeared in 2006; where this paper highlights two weakness of PSO, which are: 1) when the particles move towards the 'best' solution, a key area could be missed 2) particles close to the 'best' solution tend to stagnate; then [74] suggests and evaluates modifications to the PSO approach. It highlights three ways of implementing convergence speed enhancements, and four ways to provide search capability enhancements. This publication also proposes a variation of PSO which they refer to as modified PSO, which introduces and prevents stagnation; however, only with smaller populations. They compare the performance of modified PSO, to PSO, and a genetic algorithm, for group sizes of ten and fifty, thus demonstrating the improvement from modified PSO for small populations.

Known PSO applications in adaptive notch filtering are Shi and Zhang [75]: which is a short paper that crudely highlights the potential performance advantage of PSO when compared to LMS in locating a target signal, and

provides a summary of both approaches. Another publication by Gaing and Chang exploits PSO in an ANF within a larger system's Proportional Integral Derivative (PID) controller, which is a high-speed rail pantograph system [79]. This publication models the pantograph system, then evaluates four algorithms which are: global-oriented PSO, traditional PSO, evolutionary programming, and differential evolution; and these algorithms are evaluated with refined integrated time weighted squared error performance criteria: although three other variations of this performance criteria are briefly considered. Gaing and Chang's publication [79], highlights an innovative solution to their application; and although they do briefly consider computational efficiency, they do not evaluate other potential solutions to their application, such as a gradient-directed searching mechanism.

Now, this section specifically demonstrates how PSO can be utilised to track a single CSS, using the CANF which this research has focussed on. Therefore, PSO is used to optimise the β parameter in this CANF, as shown in Figure 7.4.

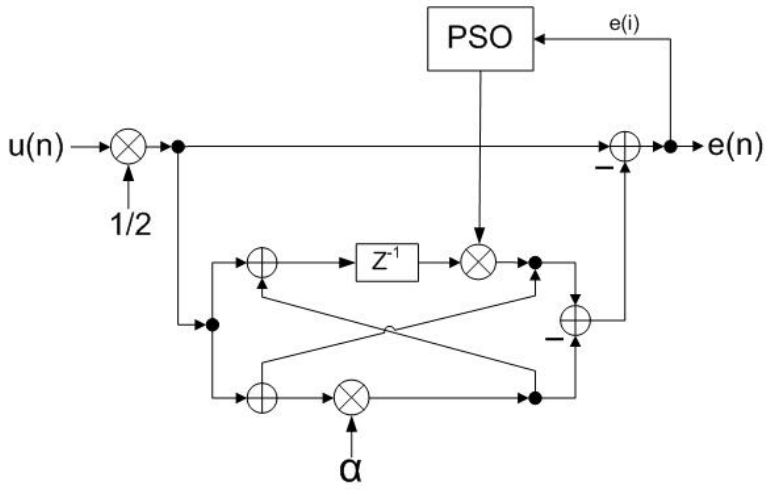


Figure 7.4. A complex adaptive notch filter employing particle swarm optimisation based adaptation.

Please take note that in this chapter, the PSO variation selected is con-

sidered the most simple variation of PSO, as stated in [80]; which the authors describe as "very intuitive and easy to program". This variation considers two parameters, which are: 1) a correction factor, this effects how much the velocity is updated, and 2) inertia, where inertia is the resistance to changing the particles direction.

When tracking a single CSS using a CANF applying PSO; the fitness function that showed the most robust performance was found to be

$$fitness\_function(i) = e(i).e^*(i). \tag{7.3.1}$$

Thus, this $fitness\_function$ is namely the instantaneous magnitude of the notch filter output squared.

In this application the fitness of each particle: which is referred to as the 'swarm particle fitness' ($swarm\_particle\_fitness$), is calculated over the 'stability length' ($stab\_length$); where the $stab\_length$ is the number of samples each particle is considered for before that particle moves to a new position; and when calculating the 'swarm particle fitness' the first three values are ignored to remove any initial transience caused by a particles new position; thus the 'swarm particle fitness' is calculated over the remaining stability length. Therefore, for a 'stability length' of L the 'swarm particle fitness' is

$$swarm\_particle\_fitness = \frac{1}{L-3}\sum_{n=4}^{L} fitness\_function(n); \tag{7.3.2}$$

then if the 'swarm particle fitness' is lower than the current 'pbest' or 'gbest' value, this particle's position becomes the new 'best' position. Now as described in the introduction, the PSO algorithm considers a 'pbest' and a 'gbest' when updating the swarm velocities; where 'pbest' is the particles best position, and 'gbest' is the overall best position achieved in the whole swarm. Thus, an improvement in 'pbest' is found by considering the particle's best position: $swm(i, pb)$, then if there is an improvement, the particle's

best Re position:

$(swm(i, pb\_Re))$, and the particle's best Im position: $(swm(i, pb\_Im))$ are updated; along with $swm(i, pb)$.

IF $swarm\_particle\_fitness < swm(i, pb)$ - if the new position is better
  $(swm(i, pb\_Re) = swm(i, vel\_Re(n))$; - update 'pbest' Re
  $(swm(i, pb\_Im) = swm(i, vel\_Im(n))$; - update 'pbest' Im
  $(swm(i, pb)) = swarm\_particle\_fitness$; - store 'pbest'
  $best = (swm(i, pb\_Re) + j.(swm(i, pb\_Im)$;
END;

Figure 7.5. Swarm code example 1 - Updating the particle's best value.

Then 'gbest' is kept up to date by checking for the optimum overall solution via the update

$[temp, gbest] = min(swm(i, pb))$; - global best position .

Figure 7.6. Swarm code example 2 - Updating the global best value.

Now, as the particles are constantly moving with a velocity, this velocity is updated so that they converge on the 'best' position; which is implemented firstly by updating each particles' real and imaginary velocity:
$swm(i, vel\_Re(n-1))$ and $swm(i, vel\_Im(n-1))$, via the update;

FOR $i = 1 : swarm\_size$

$$swm(i, vel\_Re(n)) = rand.\iota.swm(i, vel\_Re(n-1)) + \varsigma\,.rand.(swm(i, pb\_Re) - swm(i, Re(n-1)) + \varsigma.rand.(swm(gbest, Re) - swm(i, Re(n-1));$$

$$swm(i, vel\_Im(n)) = rand\,.\iota.swm(i, vel\_Im(n-1)) + \varsigma\,.rand.(swm(i, pb\_Im) - swm(i, Im(n-1)) + \varsigma.rand.(swm(gbest, Im) - swm(i, Im(n-1));$$

END;

Figure 7.7. Swarm code example 3 - Updating the particle's velocities.

herein ς is the correction factor, $rand$ is a random real value in the open interval (0,1), and ι is the inertia; then $swm(i, vel\_Re(n-1))$, is the previous Re velocity of the particle, $swm(i, vel\_Im(n-1))$, is the previous Im velocity of the particle; $(swm(gbest, Re))$ is the overall best Re position, $(swm(gbest, Im))$ is the overall best Im position; $(swm(i, Re(n-1)))$ is the particles previous Re position, and $(swm(i, Im(n-1)))$ is the particles previous Im position.

The particle's positions are then updated as follows

FOR $i = 1 : swarm\_size$

$\beta(i) = swm(i, x(n-1)) + j.swm(i, y(n-1));$

IF $newswarm >= stab\_length$

$swm(i, x(n)) = swm(i, Re(n-1)) + swm(i, vel\_Re)/1.3;$

- updates each particle's Re position

$swm(i, y(n)) = swm(i, Im(n-1)) + swm(i, vel\_Im)/1.3;$

- updates each particle's Im position

END.

Figure 7.8. Swarm code example 4 - Updating the particle's positions.

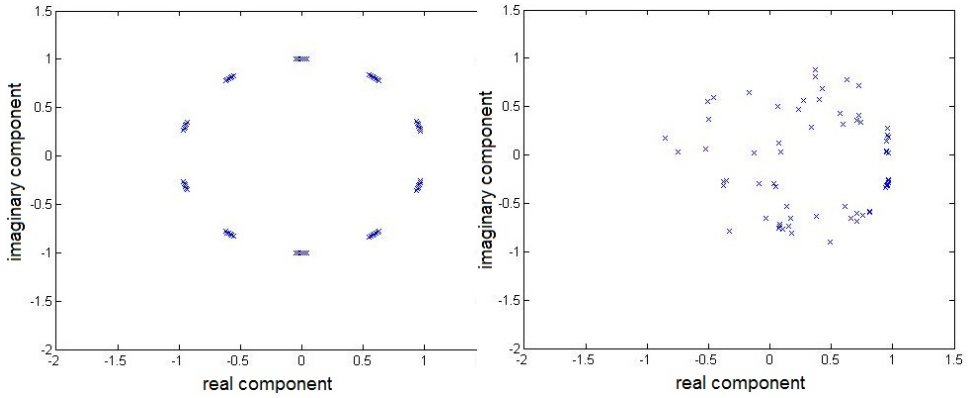
Please note that other methods were evaluated, such as only updating the velocities when an improvement was achieved. As a new particle position will not necessarily facilitate an improvement to the 'best' solution: where an improvement is a new 'best' position, which is achieved by a $fitness\_function$ for a particle producing a new 'best' position; however, these methods required extra complexity and fine tuning: particularly when tracking two CSSs. Therefore, a simple method with slow convergence on the 'best' position was found to provide the most robust overall performance, in the scenarios considered in this chapter.

When considering variations of PSO, a useful publication is [81] which considers the application of PSO to minimax problems. This publication

compares the performance of PSO to that of other established approaches, such as the sequential quadratic programming method and a recently proposed smoothing technique. It also compares the performance of three variants of PSO: one with inertia weight and without constriction factor, one with constriction factor and without inertia weight, and one with both constriction factor and inertia weight. This publication demonstrated that PSO is effective in solving minimax problems in cases where the gradient based techniques fail; however, it was outperformed by the much faster Sequential Quadratic Programming (SQP) method in less complex problems. With respect to the different variants of PSO, the one which utilised only constriction factor performed better, in terms of the cases that achieved the highest success rate, and it was always faster than inertia weighted variation. The variant in which both inertia weight and constriction factor were used, was faster than the other two variants, but had the worst success rates among all of them. Additionally it should be noted that, this publication provides further mathematical detail on PSO; and that the comprehensive article [81] is another good reference on this topic.

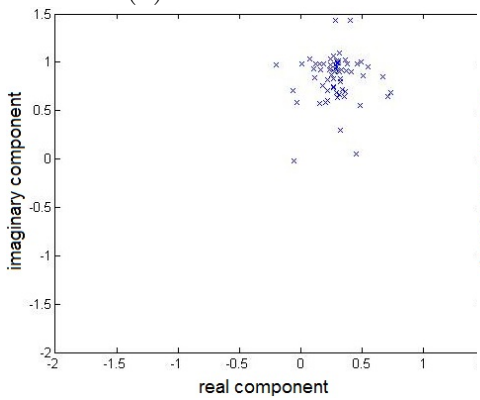
7.3.2 Simulation Results

This section of this chapter, contains simulation results for tracking a CSS with a CANF, whilst applying PSO to adapt the notch frequency parameters. In the following five figures the evolution of the swarm is plotted, to illustrate the principle of applying PSO to a CANF.

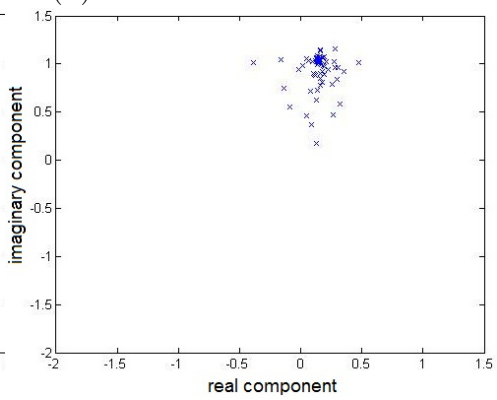


(a) Initialisation

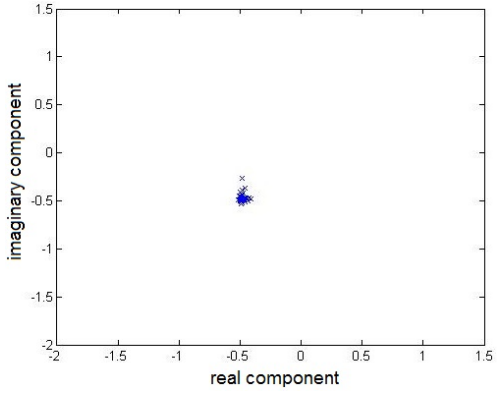
(b) The swarm's first move



(c) The swarm's second move



(d) Further swarm development



(e) The swarm's final position

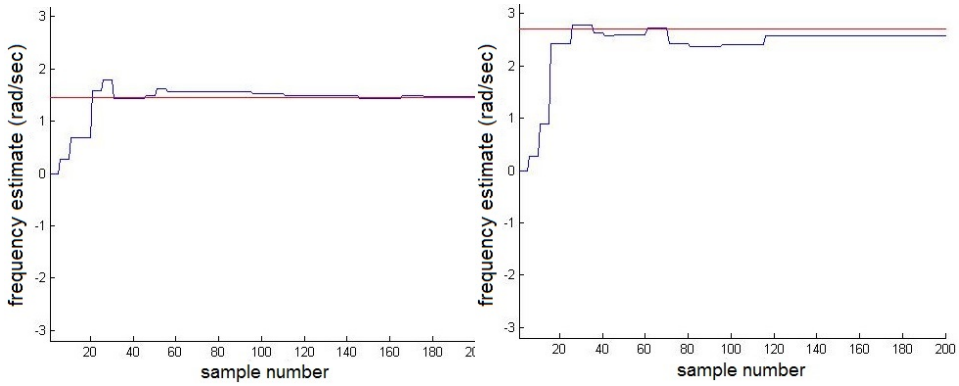
Figure 7.9. Particle swarm optimisation - Swarm evolution.

This result shows how a swarm is initialised over the complete potential solution in (a), then in (b) it begins to move towards the most likely solution; (c) and (d) show the swarm converging more, then (e) demonstrates the complete convergence of the swarm.

In these results, a 'stability length' of six samples was found sufficient to produce a result for the swarm to focus on. Also in these results, the inertia (ι) selected was 1.0; and the correctness factor (ς) was set to 0.75; and these values were found empirically.

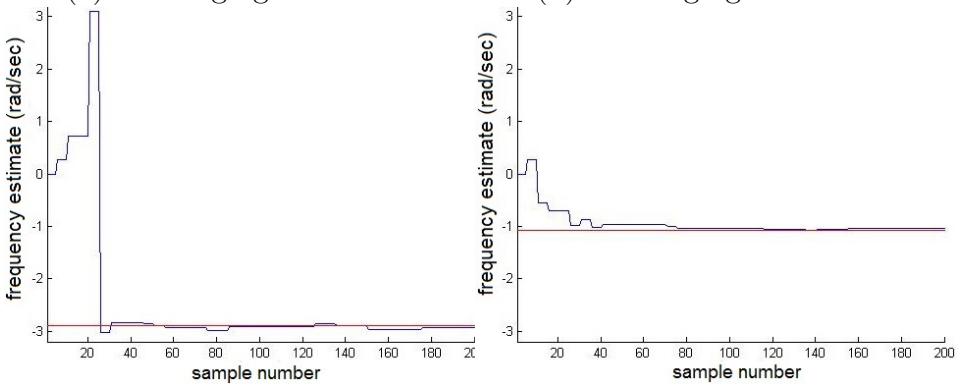
In all the results in this chapter, a 'swarm size' of 64 was selected; primarily as a value close to 50 is not excessive to run through at one time sample. Yet 50 is still large enough to provide a reasonable distribution, then 64 equates to 2^6; which is a sensible digital value to select.

In the next five figures, different CSS frequencies are located with PSO; thus demonstrating that PSO works over the full frequency range in a CANF. Therefore, five values have been selected in the range $-\pi$ to π; then the CANF locates these values utilising PSO, and Figure 7.10 demonstrates that this approach provides a viable solution over the full frequency range. In all five results the target frequency is located within 40 samples; however, when the result is close to zero as is the case in (e), the swarm does not appear to converge as well; as it appears to fluctuate around the solution more.



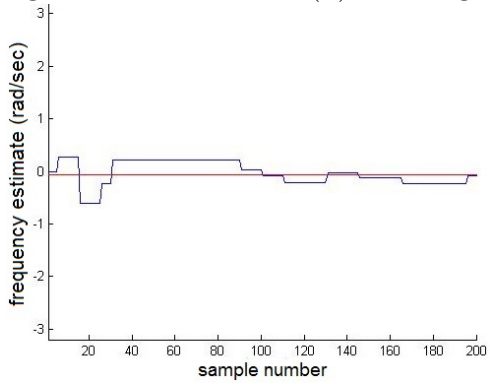
(a) Converging to 1.4451

(b) Converging to 2.7018



(c) Converging to -2.8903

(d) Converging to -1.0681



(e) Converging to -0.0628

Figure 7.10. Tracking a complex sinusoid signal with a complex adaptive notch filter exploiting particle swarm optimisation.

7.3.3 Particle Swarm Optimisation Hybrid Implementations in Adaptive Notch Filters

The PSO approach which has been implemented does occasionally fail, due to the swarm sometimes converging to an incorrect minimum point; and reinitialisation is required: see section 7.4.3; which is a weakness in this type of approach. Therefore, a hybrid arrangement is worth considering; however, this approach has previously been implemented in [71] - [73], although not for tracking a CSS.

Further reasons for applying a hybrid approach are: a) generally this approach does not fully converge to a solution as accurately as a gradient descent approach, b) a PSO approach also requires resetting when a frequency hop occurs, and c) the extra computation complexity required to implement a genetic algorithm is costly and unnecessary if continuously applied in a system; therefore, a PSO approach should only be utilised to initially locate unknown frequencies.

7.4 Tracking Two Complex Sinusoid Signals utilising a Hybrid Implementation

This hybrid approach has been implemented as a genetic algorithm is generally capable of locating large changes in hops more quickly than a gradient descent approach.

7.4.1 Implementing Particle Swarm Optimisation to Reinitialise a Notch Filter After a Frequency Hop when Tracking Two Complex Sinusoid Signals

In this section, the scenario where two hopping CSS are being tracked is considered; where when one CSS hops a stochastic search method is utilised

to quickly locate its new frequency, and this is achieved by reinitialising the CANF; which is novel, as a genetic algorithm has not been used to track two CSSs before. This scheme has also been implemented in a hybrid arrangement, and the reasons for this choice are to improve stability, and reduce complexity.

Tracking two CSSs via PSO requires a modification to the calculation of the fitness function; as when tracking a single CSS purely evaluating the magnitude of the notch filter output $e(n)$ is adequate. However, as Sayed et al. highlight in their publication [82], "there are applications where the squared-error is not the primary parameter affecting the performance of a system", and tracking two CSSs whilst utilising PSO to reinitialise after a hop in one of these frequencies, appears to be one of these applications.

Then considering their publication, Sayed et al. introduce a 'cost function' that is based on both the error magnitude and the phase error. This phase-error cost function is defined as

$$J_{pe}(\mathbf{w}) = E\angle d - \angle \mathbf{uw}|^m; \tag{7.4.1}$$

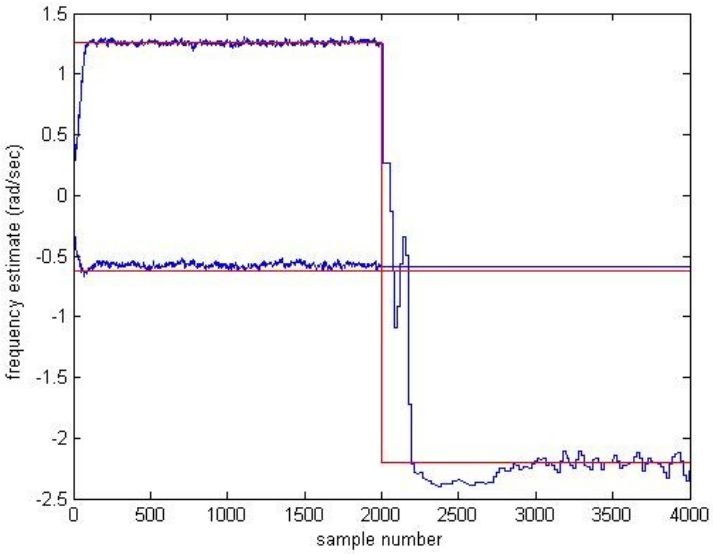
wherein, $\angle$ denotes the phase angle; the letter E denotes the expectation, with d being a scaler, u a $1 \times M$ vector, $m = 1, 2$: which is an increment that is discussed in their letter, and w is an unknown weight vector to be estimated.

Although, Sayed et al. apply this principle to a gradient descent approach; the concept of considering the phase can be applied to PSO in a 'fitness function'. Therefore, when tracking two CSSs and applying PSO to reinitialise the CANF after a single frequency hop, the fitness function that provides the most promising result is

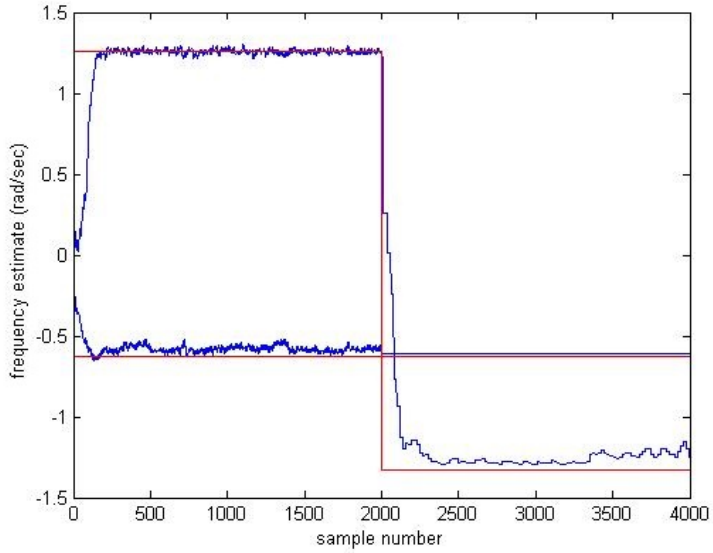
$$fitness\_function = \angle e(n); \tag{7.4.2}$$

where this fitness function is seeking the lowest phase angle of the notch filter output $e(n)$ that can be achieved. Further to this, in [82] it was stated that "The improvement is more significant when the adaptive filter is required to track fast channel variations, since the proposed structure can at least track the channel phase variations, even if it fails to properly track the channel magnitude variations"; which highlights two things: 1) they applied this method to an adaptive filter, and 2) this method works well for tracking channel variations, hence the application to frequency hops in this application.

Now Figures 7.11 and 7.12 demonstrate how PSO can reinitialise a CANF, thus locating an unknown frequency, when tracking two CSSs, in the scenario where one frequency hops; as in these figures the frequency that hops starts at 0.2, and hops at 2000 samples, whilst the second frequency is fixed at -0.1 throughout the simulation. When the target frequency hops at 2000 samples, the PSO algorithm is initiated to locate the new frequency; and when this occurs the second frequency is held at its current value. In all four of these examples, the PSO algorithm re-initialises the CANF after a hop quickly, certainly within 250 samples in Figure 7.11 and Figure 7.12 (a); although in Figure 7.12 (b), the PSO takes a little longer to fully converge: another 200 samples.

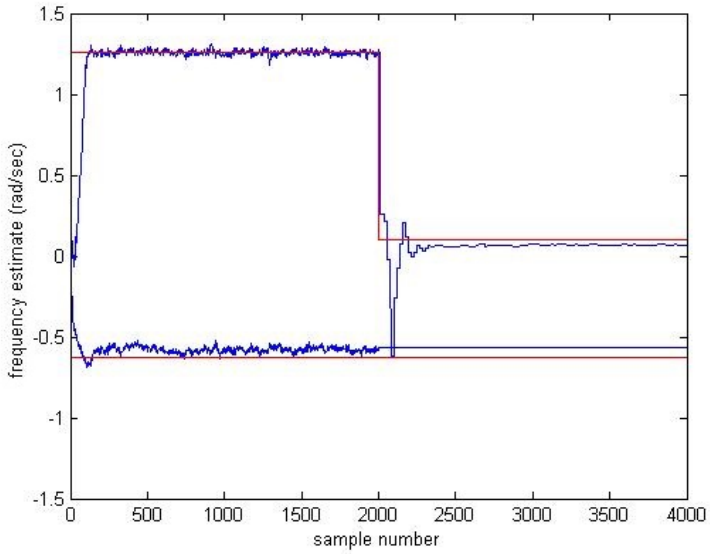


(a) Converging to -2.1991

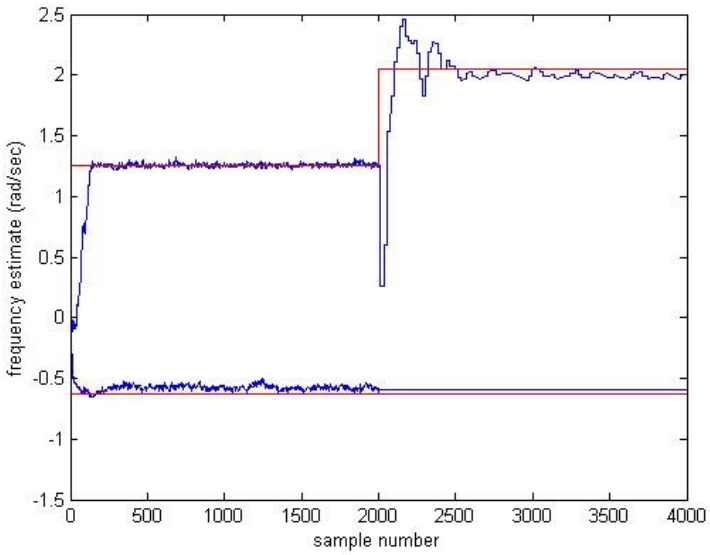


(b) Converging to -1.3320

Figure 7.11. Tracking two complex sinusoid signals, then applying particle swarm optimisation to reinitialise the notch filter when one frequency hops – Results one and two.



(a) Converging to 0.1005



(b) Converging to 2.0483

Figure 7.12. Tracking two complex sinusoid signals, then applying particle swarm optimisation to reinitialise the notch filter when one frequency hops – Results three and four.

In the next three parts of this section the main issues in the implementation of this method are discussed; which are: detecting which frequency has hopped, reinitialisation if the swarm converges to an incorrect solution, and switching back to the gradient descent approach, after the PSO algorithm has reinitialised the CANF, thus locating a hopping CSS.

7.4.2 Detecting which Frequency has Hopped

When tracking two CSSs, in the proposed scheme it is clearly necessary to be able to detect which CSS has frequency hopped; therefore, a suitable method to achieve this is to consider the gradients. Thus by adding a ψ value for each gradient then looking for a significant change in this value, the CSS which has hopped can be detected. Noting that, this method was previously implemented in Chapter 5 for $e(n)$, to detect a frequency hop when adapting the bandwidth parameter.

A frequency hop can be detected in either CSS by implementing the code IF $grad(n)\_1 > 1.25 \times \psi(n-1)_{grad\_1}$ will detect if CSS one frequency hops, and

IF $grad(n)\_2 > 1.25 \times \psi(n-1)_{grad\_2}$ will detect if the second CSS frequency hops; where the value 1.75 was found empirically: which works for most SNRs, then

$\psi(n)_{grad\_1} = \gamma.\psi(n-1)_{grad\_1} + (1-\gamma)grad(n)\_1.grad^*(n)\_1$, and similarly

$\psi(n)_{grad\_2} = \gamma.\psi(n-1)_{grad\_2} + (1-\gamma)grad(n)\_2.grad^*(n)\_2$.

In the structure utilised, the PSO is implemented in section CNF_1 of the block shown in Figure 7.13. Therefore, it may be necessary to swap the CSS frequency estimates: β_1 and β_2, to enable the PSO to function correctly; this works best by averaging the estimate of the frequency that has not hopped; considering one hundred samples, two hundred samples before the hop i.e. $(count-300 : count-200)$: where count is the current sample; then assigning this value to β_2: which is the value of β that is adapted in section

CNF_2; and this value is fixed whilst the PSO algorithm reinitialises the CANF. The hundred samples before the hop have been ignored to prevent any adaptation to the new CSS frequency offsetting the estimate.

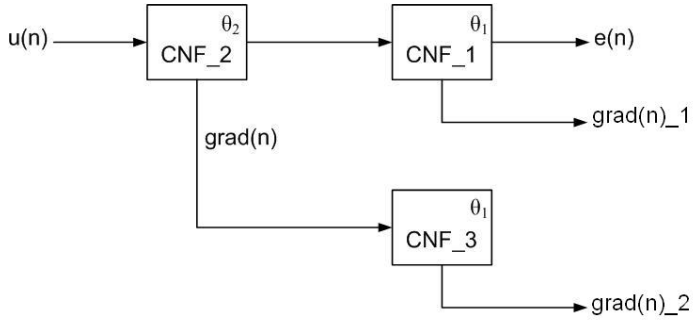


Figure 7.13. The structure for tracking two complex sinusoid signals.

7.4.3 Reinitialising the Particle Swarm Optimisation if the Swarm Converges to an Incorrect Solution

When applying PSO, occasionally the swarm converges to an incorrect solution; and if this occurs, the swarm should be reinitialised. This incorrect solution can be detected by the same manner that was applied to the bandwidth parameter in Chapter 5, i.e. by implementing a forgetting factor to the error; which is achieved as follows

$\psi_{error}(n) = \psi_{error}(n-1)\gamma + (1-\gamma).e_{conj}(n);$

if $e_{conj}(n) > 6.\psi_{error}$

Reinitialise swarm

end.

7.4.4 Switching back from Particle Swarm Optimisation to the Gradient Descent Approach

Problems may occur when you switch from the PSO algorithm back to the gradient descent approach in this hybrid arrangement, this is because the gradient descent approach has been stopped from controlling the β parameter; therefore, an initial spike occurs on the transition from a PSO algorithm back to a gradient descent algorithm.

Figure 7.14 clearly illustrates this issue of switching back from PSO to gradient descent; and within this figure, observe at 4000 samples where the switch occurs the CANF structure completely looses the CSS which has been located using PSO; however, this issue has been highlighted as an area for future work.

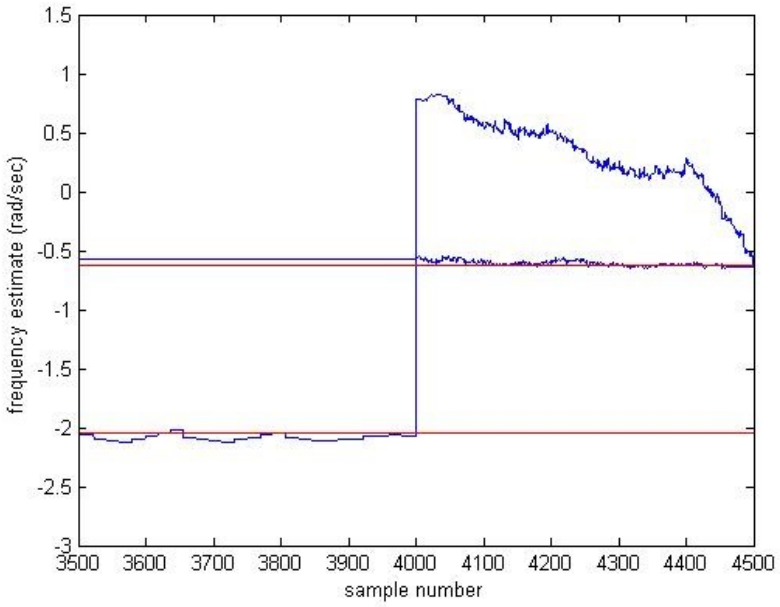


Figure 7.14. The issue of switching back from particle swarm optimisation to the gradient descent approach.

The stop condition

To prevent unnecessary iterations of the PSO algorithm a stop condition should be defined; this is generally applied in most PSO applications, and this stage is shown in Figure 7.3.

Either of two stop conditions should be applied to stop the PSO algorithm, and these are: 1) a limit should be applied to the number of iterations that the PSO algorithm can run for: such as 250, as the PSO algorithm will converge to a solution within 250 iterations, and 2) calculating a ψ_{error} then stopping the algorithm when this value is low enough. However, this is also an area for further work.

7.5 Discussions on Exploiting Particle Swarm Optimisation in Complex Notch Filters

The key points from this chapter are now discussed. PSO effectively tries a range of values which move towards its best solution, instead of applying a gradient based approach to calculate an optimum solution; therefore, PSO is costly computationally. However, PSO can quickly locate unknown frequencies.

This chapter has demonstrated that PSO can be utilised in a CANF to track a CSS; which provides an alternative to the gradient descent approaches that have been considered in earlier chapters of this thesis.

It has also demonstrated that by incorporating the phase information of the notch filter output $e(n)$; PSO can be applied to quickly lock onto an unknown target frequency in a system tracking two CSSs: when one frequency hops.

A method to detect which CSS has hopped when tracking two CSSs has been demonstrated, although switching back from PSO to a gradient descent approach in a hybrid arrangement has been highlighted as an area for future

work. It should also be mentioned that a PSO is unlikely to work well when tracking a CVCS.

7.6 Summary

This chapter firstly introduced genetic algorithms, and specifically PSO; then shows that PSO can be applied in an CANF to locate a CSS. It also considers applying PSO when tracking two CSSs, which involves authenticating when one of the frequencies has hopped, then demonstrates how to reinitialise a CANF by applying PSO to find one unknown frequency after a frequency hop: in a hybrid arrangement.

In the final chapter of this thesis: excluding the appendices, conclusions are highlighted, along with areas for future work.

Chapter 8

CONCLUSIONS AND POSSIBLE FUTURE WORK

This final chapter is split into two sections, where in the first section conclusions are drawn from the research related chapters of this thesis i.e. Chapters 3 - 7, then the second section discusses possible areas for further research, that were noted in this thesis. Thus, the first section considers these contributions.

8.1 Conclusions

In Chapter 3 of this thesis, an analysis of four ANF structures has been completed to assess their abilities to track between two and four RSSs. Part of this analysis included investigating the differences in performance between output-error and equation-error approaches. This study has demonstrated that each structure demonstrates strengths and weaknesses, which should be carefully noted.

This analysis of real structures can be summarised as: the most robust solution is provided by Chambers & Constantinides [1]; Regalia's design [2] is a computationally simple solution, however under certain conditions does not converge; Cho, Choi & Lee's structure [3] is a powerful solution, but shows the most variance in the final solution and is unstable initially, this structure also does not converge as consistently as [1], as occasionally this

structure fails to converge on solutions when tracking multiple RSSs. The DCS solution by Kwan and Martin [4] is generally outperformed, and is also computationally complex when compared to the all-pass approaches [1] - [3].

This comparison also highlighted that, the simplification facilitated by applying an equation-error approach: such as [2] or [3], may lead to multiple notch filters tracking the same signal, whilst missing other signals altogether.

Thus, in Chapter 3 four published structures have been implemented and a critical comparison of these key approaches has been successfully conducted; which highlighted the strengths and weaknesses of each approach.

Chapter 4 includes an original CANF structure; which was developed from [1]; and this new structure is based upon an output-error learning algorithm. Its superior tracking performance over Regalia's CANF solution [23] has been demonstrated; particularly for tracking two closely spaced frequencies.

Generally, an output-error approach will facilitate more reliable tracking than an equation-error approach for RSS and CSSs; however, output-error approaches are slightly more computationally complex. For practical applications the low complexity and robustness of the proposed structure are indeed very attractive. It should be noted that, an equation-error approach may have a wider basin of attraction than output-error methods; which can be an advantage, particularly when tracking a single CSS.

This scheme specifically allows two CSSs that are closer than the normalised value of 0.3770 to be tracked, which is not possible with Regalia's equation-error approach.

In Chapter 5, it was demonstrated that the performance of a CANF can be significantly improved if both the notch bandwidth and frequency parameters are updated simultaneously, when tracking hopping CSSs; although this can significantly increase the complexity of the design.

Adapting the notch bandwidth parameter also improves the ability of

this structure for the tracking of CSSs which are further apart. In addition, novelty has been demonstrated, as it has been shown that a method of steepest ascent is required for the adaptation of the notch bandwidth parameter.

Three methods for updating the notch bandwidth parameter have been evaluated, firstly for tracking a single hopping CSS; then tracking two hopping CSSs, whilst adapting a single notch bandwidth parameter for the overall structure, lastly, individual notch bandwidth parameters were updated for each hopping CSS; and this research showed strengths and weaknesses for each of the three methods.

Adapting the notch bandwidth parameter can reduce the variance of the estimate from 0.0013 to 1.06×10^{-4}: when tracking a single CSS, although this will slow the convergence by approximately 50 samples.

Chapter 6 of this thesis showed the strong performance of the proposed design for tracking a CVCS. The results demonstrate that, no real improvement is achieved from updating the notch bandwidth parameter when tracking a CVCS, with the methods evaluated herein; although some improvement was demonstrated when tracking a less severe CVCS and a hopping CSS simultaneously.

This chapter also included a result for the full gradient term for the adaptation of β: which is derived in Appendix 9.2 of this thesis, and this may improve the results in tracking signals for certain scenarios, as this approach improves the results slightly for tracking a CVCS.

Chapter 6 specifically demonstrated that when tracking a CVCS, the variance in Regalia's estimate could be reduced from 0.0026 to 0.0017 by utilising the proposed structure. Additionally, when tracking a CVCS and a hopping CSS, the variances could be improved from 7.02×10^{-4} and 5.76×10^{-4}, to 8.96×10^{-6} and 6.45×10^{-5} by adapting the notch bandwidth parameter.

In Chapter 7 of this thesis, a stochastic search approach was applied to learning in a CANF; which demonstrated that PSO can be exploited in an CANF to locate a CSS. This stochastic search method provides an alternative to the gradient descent approaches, that have been considered in earlier chapters of this thesis.

It has also demonstrated that by incorporating the phase information of the notch filter output $e(n)$; PSO can be applied to reinitialise a CANF, thus quickly locating a new unknown frequency; in a system tracking two CSSs, when one frequency hops.

By applying PSO, the number of samples required to locate a frequency hop can be reduced to 200 samples, which is a reduction from 1000 samples with a gradient descent method, in certain scenarios.

Issues around tracking multiple CSSs whilst applying PSO have been investigated, which include: a method to detect which CSS has hopped when tracking two CSSs, reinitialising the PSO algorithm when it fails; then lastly, switching back from PSO to a gradient descent approach in a hybrid arrangement. It should also be mentioned that PSO is unlikely to work well when tracking a CVCS.

This completes the conclusions observed from this thesis, and next topics for further research are discussed.

8.2 Further Related Research Topics

Further possible work on structures, could be initiated from a paper by Mitra [83]: who originally worked with Regalia; as this publication shows that a structure can be implemented in many ways, and further variations of the structure proposed in this thesis could be investigated. Also, a complex version of Cho, Choi & Lee's scheme [3] would be worth investigating i.e. producing a CANF version of [3]. Developing possible hybrid structures,

i.e. a combination of this structure and Regalia's approach [23], could be another area for future research: when tracking multiple CSSs.

The CANF structure in this thesis and Regalia's scheme [23], have the potential issue that the gain at dc: z = 1, and Nyquist: z = -1, is not unity; therefore, a second filter is required to provide a constant end to end SNR improvement ratio, which should be considered.

Comparing the performance of Nishimura's CANF structure [50], to Regalia's and the structure developed in this thesis, would certainly be of interest.

Completing further convergence analysis, is another area for further research; particularly with respect to tracking multiple CSSs. Please note that, some references to properties of ANFs are included in: Chapter 9, Appendix 9.1.

Implementing adaptive constraints that assist in the update of the notch bandwidth parameter, is an area that could be explored further; along with considering hyper-stable algorithms: which are described in Chapter 9 of Regalia's monograph [45].

An idea inspired from the ISP conference in December 2013, is that a system could be developed with a CANF structure to track an unknown number of CSSs, by first scanning the frequency domain with a method such as the paper presented at the ISP conference [84]; this would highlight the number of CSS and their current value, then a cascaded CANF structure could be implemented to track them, thus reducing the power and computations required by the system.

Tracking multiple CVCSs is very challenging, and is certainly an area that could be explored further. Also, considering different types of coloured noise is another area which could be investigated further; for example, the CANF may behave differently from Grey or Brownian noise.

Further work could be undertaken on applying Bayesian theory to PSO;

where if there is more than one likely overall best solution, the swarm splits to consider all the initial strong solutions; and this approach should improve the success of PSO techniques in CANFs.

Switching back from the PSO algorithm to the gradient descent approach requires further work, as there are instances where the gradient approach chooses to swap the estimates over; so ω_1 converges to ω_2, and ω_2 converges to ω_1: which is not ideal.

The primary weakness of the LMS algorithm, is the selection of a suitable adaptation gain; which if selected incorrectly, will produce slow convergence: if the value is too low; or oscillation around the solution: if the value is too high. Therefore, further types of learning algorithm could also be considered; where one example of types of algorithm that could be considered is the RLS / Newton type of algorithm [85].

Considering a specific application, is another suitable research topic; and some examples of the areas where ANFs have been utilised are summarised in: Chapter 1, section 1.2.

This concludes this thesis, which primarily developed three aspects of a CANF for the tracking of multiple CSSs; and these aspects were:

1. Developing a new CANF structure, which is capable of tracking closely spaced CSSs,

2. Adapting the notch bandwidth parameter with a method of steepest ascent, to further improve the performance of the CANF, and

3. Applying PSO to reinitialise a CANF, thus quickly locating a frequency hopping CSS, whilst tracking two CSSs.

In addition, tracking a CVCS has been considered; along with a critical comparison of four structures capable of tracking RSSs.

Chapter 9

APPENDICES

This final chapter of the thesis contains two appendices; where Appendix 9.1 contains details of further literature relevant to this thesis.

Then, Appendix 9.2 contains an extension relevant to Chapters 4, 5 and 6; which is a full gradient term for the adaptation of the β parameter.

9.1 Appendix A - Further Notable Literature

This first appendix reviews recent developments in ANFs, and searching on "Adaptive Notch Filters" (ANF) from 1991-2012 within the IEEE database returned 514 results. To reduce this number only journals were reviewed, which decreased the number of papers to 137. The most promising papers were then divided into four categories: structures, algorithms, properties of ANFs, and ANF applications. Structures and properties are discussed in the next two subsections; as this thesis focussed on stochastic search methods instead of algorithms, the algorithm papers were not considered further; and the applications have been listed in section 1.2 of Chapter 1; thus, papers on ANF structures from 1991-2012 are summarised first.

9.1.1 Publications on Adaptive Notch Filter Structures from 1991-2012

1991-1994

The first ANF structure located was by Kwan & Martin [86], and these researchers had previously developed the strong DCS solution [4]. However, this structure is geared towards an analogue implementation, which uses operational transconductance amplifiers; therefore, as this research is based around digital signal processing (DSP), this structure has not been considered further.

The next structure located is by Regalia [2], and this structure has been discussed in the Relevant Literature review in Chapter 2.

Interestingly, a phase locked loop (PLL) was located next as [87], which was published by Wulich et al. PLLs are a notable alternative to ANFs, although this PLL has in fact been built using a constrained notch filter and a complex adaptive notch filter (CANF); however, current issues around PLLs include: unpredictable electromagnetic emissions - created by constantly changing frequencies, inconsistent initialisation particularly at the temperature extremes, and jitter. Although please take note that, currently PLLs are generally implemented via voltage controlled oscillators.

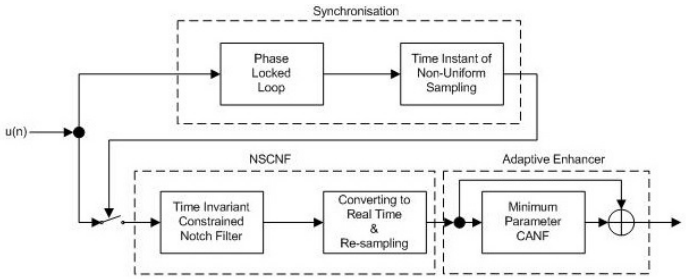


Figure 9.1. A block diagram of a phase locked loop by Wulich et al. [87].

Cho & Lee's structures are of particular interest as they provide a very powerful solution in [3], which is included in Chapter 3 in this research; and their next publication [55], focusses on output-error and equation-error approaches, considering the structures shown below; although structures are not the focus of this piece of research.

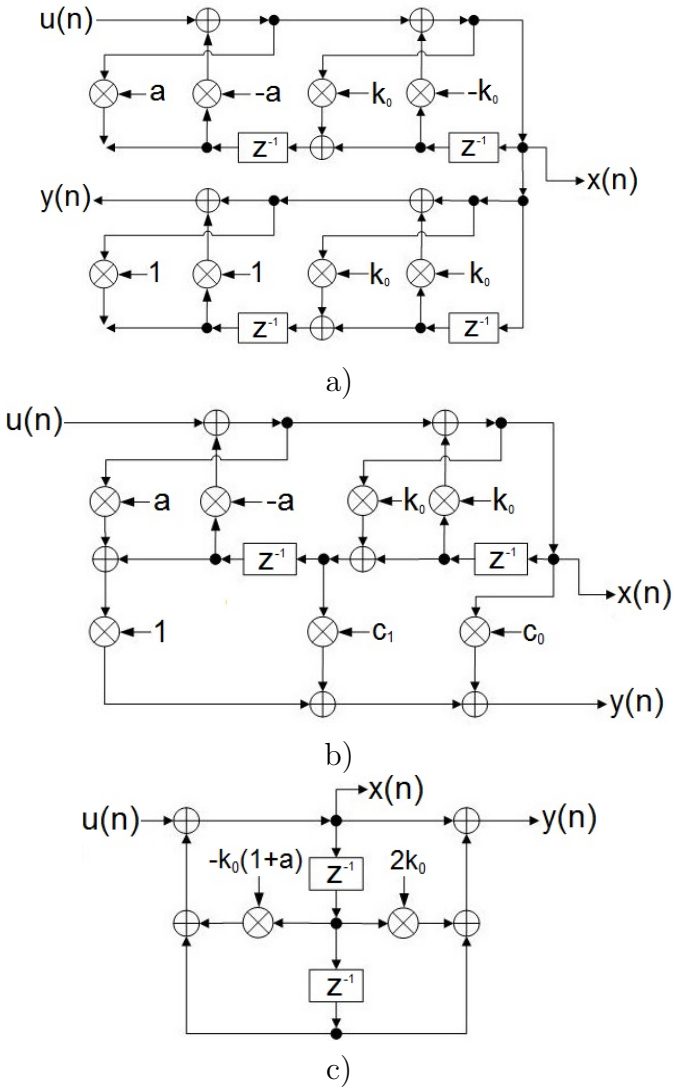


Figure 9.2. Improved lattice structures by Cho & Lee published in [55].

Next, a paper by Ko and Li [88] shows another equation-error approach. It would certainly be interesting to compare the performance of this design to Regalia's, however, this has not been completed in this thesis. The reason that this approach has not been considered further, is that this is purely a modification to the equation-error approach; which in fact adds slightly more complexity, hence is moving back towards an output-error approach in a hybrid arrangement. Also, this research has primarily focussed on output-error approaches.

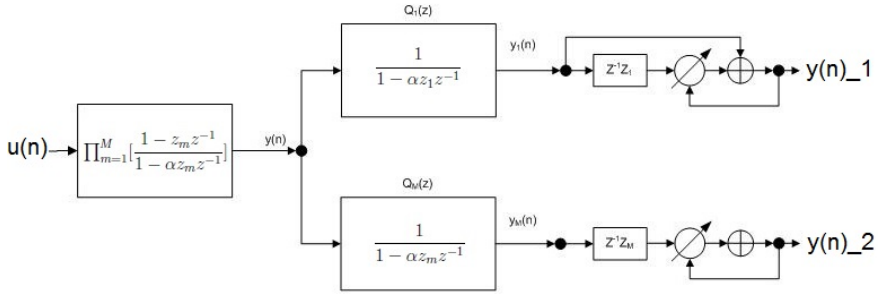


Figure 9.3. An equation-error structure by Ko and Li [88].

1995-1999

Linder et al. published a structure in 1996 [89], however, this is an analogue implementation in BiCMOS so is not relevant to this research.

Then in 1997 Soderstrand et al. developed Kwan & Martin's design further in [90], this is of interest although has not been comprehensively evaluated, as the improvement claimed is a reduction in complexity, not an improvement in performance: which is what would be required to make this DCS structure comparable to an all-pass design.

Please note that Nishimura's CANF structures which were published at this time are detailed in the literature review.

2000-2004

Karimi-Ghartemani et al. published another PLL in 2002 [91], this design is based on Phase Detection and utilises a Voltage Controlled Oscilator; and as previously mentioned PLLs are not evaluated in this research.

In 2004 Mojiri and Bakhshai modified Regalia's design into the analogue form shown below in [92]; herein, ζ is the damping ratio: which determines the notches depth. In addition he mentions that Bodson and Douglas also modified Regalia's design, which he cites as his third reference. Thus due to the fact that this structure is analogue, as it clearly contains integrators; it has been discounted from this research.

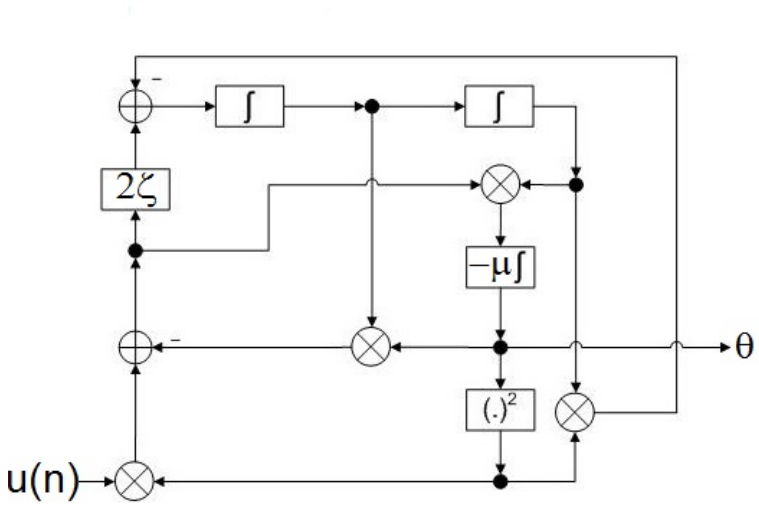


Figure 9.4. Mojiri's analogue implementation of Regalia's structure [92].

Lim et al. published a modification to the lattice structure in 2005 as [93], showing a new method for extracting information on the distance between the frequency of the input sinusoid and the zero of the notch, which is a notable concept.

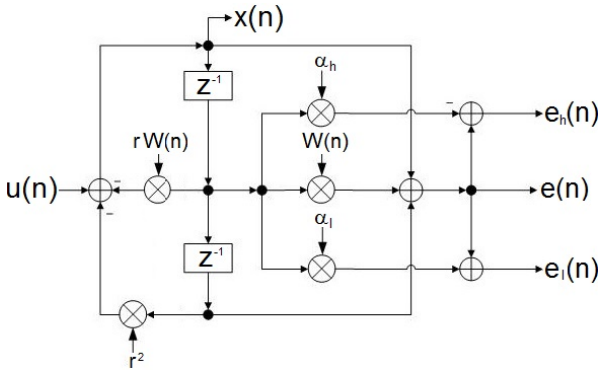


Figure 9.5. Lim's modification to the notch filter output in the lattice structure [93].

2005-2009

Mojiri et al. then published two further papers in 2007 continuing with the analogue form, where the first paper [94] allows the structure to implement a different learning algorithm; then Mojiri compares his design to a PLL stating some advantages of an ANF, which he claims are 'a more simple implementation and faster transient response'. In the second paper [95] Mojiri and Bakhshai demonstrate how their structure can be modified to track multiple RSSs.

Next Regalia's structure was modified by Yazdani et al. in 2008 as [96]; however, this was also in an analogue form, which he links to power grid applications, Yazdani also made a further publication on the same topic in 2009.

2010-2012

The last notable structure encountered is Regalia's modified structure [23], which has been discussed in detail in this thesis.

To summarise, if further work was completed on complex all-pass structures, a solid starting point for this work would be Mitra's publication [83]; as this paper includes further details on the many ways to synthesise complex all-pass structures, which certainly would be worth considering. In addition a CANF version of Cho & Lee's paper [55], would also be worth researching.

Other key notes from this summary of structures from 1991-2013 are: Regalia's Lattice appears to be popular, and PLLs are also widely used as an alternative to ANFs for the tracking of sinusoid signals .

9.1.2 Publications Related to the Properties of Adaptive Notch Filters from 1991-2012

ANF have several properties which can be analysed and developed, and this section now provides a short overview of some papers which have been published on this topic.

Analysing the convergence of an algorithm, is a property which certainly should be mentioned; and Ljung published a paper in 1977 as [97], which is still widely cited on this topic, as the paper provides a very comprehensive evaluation of recursive algorithms: as it is 25 pages in total. More recently, four papers of interest have been published by Xiao et al., where the first paper 'Steady state analysis of a plain gradient algorithm for a second-order adaptive IIR notch filter' [98], was published 2001; the second paper 'Tracking properties of a gradient-based second-order adaptive IIR notch filter with constrained poles and zeros' [99], was published in 2002; the third paper 'Performance analysis of the sign algorithm for a constrained adaptive IIR notch filter' [100], was published in 2003; and the forth paper

'Statistical performance of the memoryless nonlinear gradient algorithm for the constrained adaptive IIR notch filter' [101], was published in 2005. Then on a similar topic 'A structural view of stability in adaptive IIR filters' [102], was published in 2006 by Zhou et al.

Adapting the notch bandwidth parameter is another property of an ANF, and there are few publications on this topic. The adaptation of the notch bandwidth parameter, is formally considered with a separate literature review, in Chapter 5 of this thesis.

A recent paper [103], which was published in 2011; provides an evaluation of modern algorithms for creating smooth notch tracking, i.e. removing any jagged transitions in the tracking of a signal. Where, Niedźwiecki who is the author of this paper, has published a number of papers on this topic.

Preliminary Work Completed

This research programme: which was completed part-time, followed on from a Master of Science qualification in Digital Communication Systems at Loughborough University.

Other notable background theory completed included: three statistical signal processing assignments, and all thirty-four on-line lectures in the subject of further linear algebra presented by Professor Strang of the Massachusetts Institute of Technology [104]. There are several books written by Strang, which assist in following this course in linear algebra, and one of these books is [105]. Another piece of preliminary work completed compared the LMS and RLS algorithms, which was included in the first year report for this research programme.

A very useful book that provides a strong introduction to DSP is [106], which introduces many aspects of this subject, and also provides relevant information on all-pass structures; however, in 2012 a new edition of this book has been published.

9.2 Appendix B - Extension Related to Chapters 4, 5 & 6

9.2.1 The Full Gradient Term for β

This appendix derives the full gradient term for β, which has been derived to evaluate the effect of the simplified version of this term, that has been applied in the proposed structure.

The derivation begins with the transfer function of the structure, which is

$$H_{notch}(z) = \frac{E(z)}{U(z)} = \frac{1}{2}\frac{(1+\alpha)(1-z^{-1}\beta)}{1-\alpha z^{-1}\beta} \tag{9.2.1}$$

$$= \frac{1}{2}(1+\alpha)(1-z^{-1}\beta)(1-\alpha z^{-1}\beta)^{-1}. \tag{9.2.2}$$

Now to generate a derivative with respect to θ, firstly β must be replaced by θ, as $\beta = e^{j\theta}$ substituting θ back into (9.2.2) provides the form

$$H_{notch}(z) = \frac{1}{2}(1+\alpha)(1-z^{-1}e^{j\theta})(1-\alpha z^{-1}e^{j\theta})^{-1}. \tag{9.2.3}$$

Then by differentiating the two products containing θ from (9.2.3), with respect to θ; via the 'differentiation of products' rule, creates the expression

$$\frac{\delta}{\delta\theta} = \frac{(1+\alpha)}{2}[(1-z^{-1}e^{j\theta})(1-\alpha z^{-1}e^{j\theta})^{-2}(j\alpha z^{-1}e^{j\theta}) + (1-\alpha z^{-1}e^{j\theta})^{-1}(-jz^{-1}e^{j\theta})]. \tag{9.2.4}$$

Equation (9.2.4) can then be rearranged to the form

$$\frac{\delta}{\delta\theta} = \frac{(1+\alpha)}{2}\left[\frac{(1-z^{-1}e^{j\theta})(j\alpha z^{-1}e^{j\theta})}{(1-\alpha z^{-1}e^{j\theta})^2} + \frac{(-jz^{-1}e^{j\theta})}{(1-\alpha z^{-1}e^{j\theta})}\right]. \tag{9.2.5}$$

Now, creating a common denominator for (9.2.5), and replacing $e^{j\theta}$ with β

creates the expression

$$\frac{\delta}{\delta\theta} = \frac{(1+\alpha)}{2}\left[\frac{(1-z^{-1}\beta)(j\alpha z^{-1}\beta) + (-jz^{-1}\beta)(1-\alpha z^{-1}\beta)}{(1-\alpha z^{-1}\beta)^2}\right]; \quad (9.2.6)$$

then multiplying out and simplifying leaves

$$\frac{\delta}{\delta\theta} = \frac{(1+\alpha)}{2}\left[\frac{j\alpha z^{-1}\beta - jz^{-1}\beta}{(1-\alpha z^{-1}\beta)^2}\right]. \quad (9.2.7)$$

Thus, the full gradient term for β has been derived as

$$\begin{aligned} Grad_{full}\beta &= \frac{(1+\alpha)}{2}\left[\frac{jz^{-1}\beta(\alpha-1)}{(1-\alpha z^{-1}\beta)^2}\right] \\ &= \frac{(1+\alpha)}{2}\left[\frac{j}{(1-\alpha z^{-1}\beta)} \times \frac{z^{-1}\beta(\alpha-1)}{(1-\alpha z^{-1}\beta)}\right]. \end{aligned} \quad (9.2.8)$$

The structure required to implement this full gradient term is shown in Figure 9.6.

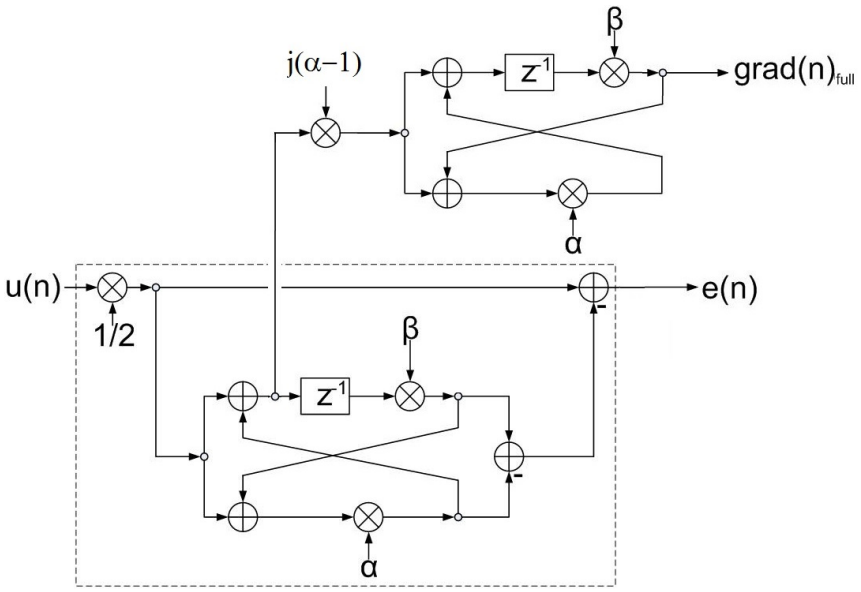


Figure 9.6. The structure required to implement the full gradient term for updating β.

Now, simulating this structure with the full gradient term for β produces the result shown in Figure 9.7. Wherein, Figure 9.7 was created with $\mu =$

0.25 and $\gamma = 0.7$, and for consistency a value of 0.8 was implemented for α in these results.

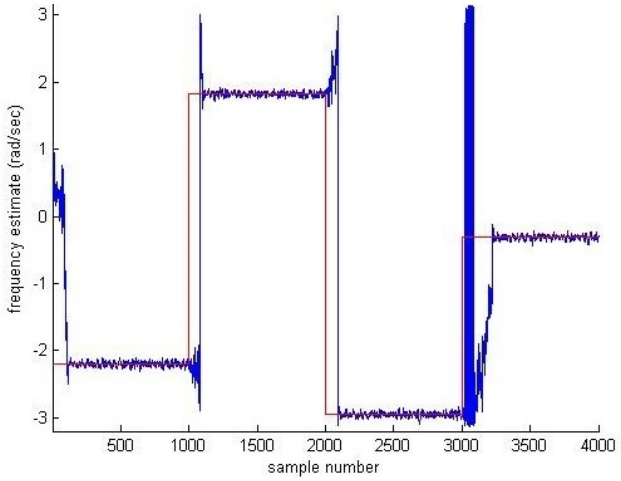


Figure 9.7. Full gradient approach for adapting β, whilst tracking a hopping complex sinusoid signal.

To facilitate a comparison with the partial gradient term for β the equivalent result for the partial gradient term is shown in Figure 9.8.

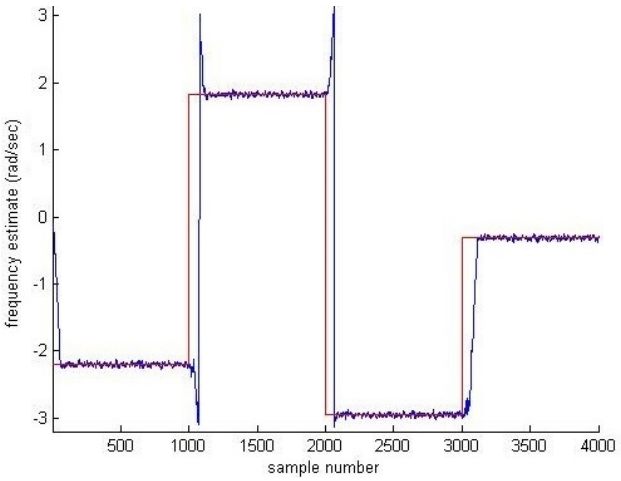


Figure 9.8. Partial gradient approach for adapting β, whilst tracking a hopping complex sinusoid signal.

From these results its clear that the full gradient term for β has more noise in its estimate, this is because the extra complexity in the full gradient term requires a larger adaptation gain, however, this full gradient term may facilitate solutions to specific applications, where the simplified term fails.

References

[1] J. A. Chambers, A. G. Constantinides, "Frequency tracking using constrained adaptive notch filters synthesized from allpass sections", Proceedings on Instrumentation Electronic Engineering F, vol.137, pp. 475–481, Dec. 1990.

[2] P. A. Regalia, "An improved lattice-based adaptive IIR notch filter", Signal Processing, IEEE Transactions on, vol.39, pp. 2124–2128, Sep. 1991.

[3] N. I. Cho, C.-H. Choi, S. U. Lee, "Adaptive line enhancement using an IIR lattice notch filter", Acoustics Speech and Signal Processing, IEEE Transactions on, vol.37, no.4, pp. 585–589, April 1989.

[4] T. Kwan, K. Martin, "Adaptive detection and enhancement of multiple sinusoids using a cascade IIR filter", Circuits and Systems, IEEE Transactions on, vol.36, no.7, pp. 937–947, July 1989.

[5] M. V. Dragosevic, "Autofocusing of moving objects in SAR data based on adaptive notch filtering", Aerospace and Electronic Systems, IEEE Transactions on, vol.44, no.1, pp. 384–392, Jan. 2008.

[6] W.-S. Ra, "Practical adaptive notch filter for missile bending mode rejection", Electronics Letters, vol.41, no.5, pp. 228–229, March 2005.

[7] D. Borio, L. Camoriano, L. L. Presti, "Two-Pole and Multi-Pole Notch Filters: A Computationally Effective Solution for GNSS Interference Detec-

tion and Mitigation", IEEE Systems Journal, vol.2, no.1, pp. 38–47, March 2008.

[8] Y. Zhenzhen, C. Duan, P. V. Orlik, J. Zhang, A. A. Abouzeid, "A Synchronization Design for UWB-Based Wireless Multimedia Systems", Broadcasting, IEEE Transactions on, vol.56, no.2, pp. 211–225, June 2010.

[9] J. Yogananda, "Flight Control Software: Mistakes Made and Lessons Learned", Software, IEEE , vol.30, no.3, pp. 67–72, May-June 2013.

[10] M. Ferdjallah, R. E. Barr, "Adaptive digital notch filter design on the unit circle for the removal of powerline noise from biomedical signals", Biomedical Engineering, IEEE Transactions on, vol.41, no.6, pp. 529–536, June 1994.

[11] P. S. Hamilton, "A Comparison of Adaptive and Nonadaptive Filters for Reduction of Power Line Interference in the ECG", IEEE Transactions on Biomedical Eng., vol.43, no.1, pp. 105–109, Jan. 1996.

[12] P. Laguna, R. Jane, P. Caminal, "Adaptive filtering of ECG baseline wander", Engineering in Medicine and Biology Society, 14th Annual International Conference of the IEEE, vol.2, pp. 508–509, Oct.-Nov. 1992.

[13] S. Nishimura, A. Mvuma, T. Hinamoto, "Frequency estimation of three-phase power systems using complex adaptive notch filters", Circuits and Systems (ISCAS), IEEE International Symposium on, pp. 2297–2300, June 2014.

[14] M. Karimi-Ghartemani, "A novel three-phase magnitude-phase-locked loop system," Circuits and Systems I: Regular Papers, IEEE Transactions on, vol.53, no.8, pp. 1792–1802, Aug. 2006.

[15] R. R. Pereira, C. H. da Silva, L. E. B. da Silva, G. Lambert-Torres, J. O. P. Pinto. "New Strategies for Application of Adaptive Filters in Active

Power Filters", Industry Applications, IEEE Transactions on, vol.47, no.3, pp. 1136–1141, May-June 2011.

[16] A. El-Nady, A. Noureldin. "Mitigation of Arc Furnace Voltage Flicker Using an Innovative Scheme of Adaptive Notch Filters", Power Delivery, IEEE Transactions on, vol.26, no.3, pp. 1326–1336, July 2011.

[17] K. Ohno, T. Hara, "Adaptive resonant mode compensation for hard disk drives", Industrial Electronics, IEEE Transactions on, vol.53, no.2, pp. 624–630, April 2006.

[18] J. Levin, P. Ioannou, "Multirate adaptive notch filter with an adaptive bandwidth controller for disk drives", American Control Conference, pp. 4407–4412, June 2008.

[19] J. Lee, E. Song, Y. Park, D. Youn, "Effective bass enhancement using second-order adaptive notch filter", Consumer Electronics, IEEE Transactions on, vol.54, no.2, pp. 663–668, May 2008.

[20] K. Seki, M. Iwasaki, M. Kawafuku, H. Hirai, K. Yasuda, "Adaptive Compensation for Reaction Force With Frequency Variation in Shaking Table Systems", Industrial Electronics, IEEE Transactions on, vol.56, no.10, pp. 3864–3871, Oct. 2009.

[21] M. Mojiri, M. Karimi-Ghartemani, A. Bakhshai, "Processing of Harmonics and Interharmonics Using an Adaptive Notch Filter", Power Delivery, IEEE Transactions on, vol.25, no.2, pp. 534–542, April 2010.

[22] A. L. Szczupak, L. W. P. Biscainho, "Adaptive IIR Notch Filters for tracking of quasi-harmonic signals", Signal Processing Conference (EUSIPCO), Proceedings of the 20th European, pp. 2119–2123, Aug. 2012.

[23] P. A. Regalia, "A Complex Adaptive Notch Filter", Signal Proc. Letters, IEEE, vol.17, pp. 937–940, Nov. 2010.

[24] T. Kailath, "A view of three decades of linear filtering theory", Information Theory, IEEE Transactions on, vol.20, no.2, pp. 146–181, March 1974.

[25] S. Haykin, "Adaptive Filter Theory", Prentice-Hall, Englewood Cliffs, N.J. 07632, USA, 1986.

[26] A. Papoulis, "Probability, Random Variables and Stochastic Processes", McGraw-Hill, 1984.

[27] O. Toeplitz, Wikipedia reference, *http://en.wikipedia.org/wiki/Otto_Toeplitz*, accessed 29 Jan. 2014.

[28] N. Levinson, "The Wiener RMS (root mean square) error criterion in filter design and prediction", J. Math. Phys. 25, pp. 261–278, 1947.

[29] J. V. Candy, "Signal processing: the model-based approach", McGraw-Hill, 1986.

[30] B. Widrow, M. E. Hoff Jr., "Adaptive switching circuits", IRE WESCON Conv. Rec., vol.4, pp. 96–104, 1960.

[31] H. Robbins, S. Monro, "A Stochastic Approximation Method", Ann. Math. Statist. vol.22, no.3, pp. 400–407, 1951.

[32] D. J. Sakrison, "The use of stochastic approximation to solve the system identification problem", Automatic Control, IEEE Transactions on, vol.12, no.5, pp. 563–567, Oct. 1967.

[33] D. Godard, "Channel Equalization Using a Kalman Filter for Fast Data Transmission", IBM Journal of Research and Development, vol.18, no.3, pp. 267–273, 1974.

[34] L. Ljung, T. Söderstrom, "Theory and Practice of Recursive Identification", Cambridge, MA : MIT Press, ISBN-13: 978-0262120951, 1983.

[35] D. D. Falconer, L. Ljung, "Application of Fast Kalman Estimation to Adaptive Equalization", Communications, IEEE Transactions on, vol.26, no.10, pp. 1439–1446, Oct. 1978.

[36] J. M. Cioffi, T. Kailath, "Fast, recursive-least-squares transversal filters for adaptive filtering", Acoustics Speech and Signal Processing, IEEE Transactions on, vol.32, no.2, pp. 304–337, April 1984.

[37] P. L. Feintuch, "An adaptive recursive LMS filter", Proc. IEEE, vol.64, no.11, pp. 1622–1624, Nov. 1976.

[38] S. A. White, "An adaptive recursive digital filter", Annual Asilomar Conf. on Circuits, Syst., and Computers, 9th, Pacific Grove, California, pp. 21–25, Nov. 1975.

[39] M. G. Larimore, J. R. Treichler, C. R. Johnson Jr., "An algorithm for adapting IIR digital filters", Acoustics Speech and Signal Processing IEEE Transactions on, vol.28, no.4, pp. 428–440, Aug. 1980.

[40] H. R. Abutalebi, H. Sheikhzadeh, R. L. Brennan, G. H. Freeman, "Affine Projection Algorithm for Oversampled Subband Adaptive Filters", IEEE International Conference on Acoustics Speech and Signal Processing, April 2003.

[41] B. Widrow, J. McCool, M. Ball, "The complex LMS algorithm", Proc. IEEE, vol.63, pp. 719–720, April 1975.

[42] J. J. Shynk, "A complex adaptive algorithm for IIR filtering", Acoustics Speech and Signal Processing, IEEE Transactions on, vol.ASSP-34, pp. 1342–1344, Oct. 1986.

[43] S.-C. Pei, C.-C. Tseng, "Complex adaptive IIR notch filter algorithm and its applications", Circuits and Systems II, IEEE Transactions on, vol.41, no.2, pp. 158–163, Feb. 1994.

[44] J. A. Chambers, "Digital Signal Processing Algorithms and Structures for Adaptive Line Enhancing", PhD Thesis, Imperial College London, May 1990.

[45] P. A. Regalia, "Adaptive IIR Filtering in Signal Processing and Control", New York: Marcel Dekker, ISBN-13: 978-0824792893, 1995.

[46] A. Nehorai, "A minimal parameter adaptive notch filter with constrained poles and zeros", Acoustics Speech and Signal Processing, IEEE Transactions on, vol.ASSP-33, pp. 983–996, Aug. 1985.

[47] P. A. Regalia, S. K. Mitra, P. P. Vaidyanathan, "The digital all-pass filter: A useful signal processing building block", Proc. IEEE, vol.76, pp. 19–37, Jan. 1988.

[48] S. Nishimura, J. Hai-Yun, "Gradient-based complex adaptive IIR notch filters for frequency estimation", Circuits and Systems, IEEE Asia Pacific Conference on, pp. 235–238, Nov. 1996.

[49] S. Nishimura, J. Hai-Yun, T. Hinamoto, "Performance analysis of complex adaptive IIR notch filters", Signal Processing Proceedings, ICSP Fourth International Conference on, vol.1, pp. 502–505, 1998.

[50] S. Nishimura, A. Mvuma, T. Hinamoto, "Tracking properties of complex adaptive notch filter for detection of multiple real sinusoids", Circuits and Systems (ISCAS), IEEE International Symposium on, pp. 2928–2931, May 2013.

[51] S. Nishimura, A. Mvuma, T. Hinamoto, "Complex coefficient adaptive IIR notch filter tracking characteristics", Circuits and Systems, MWSCAS 52nd IEEE International Midwest Symposium on, pp. 640–643, Aug. 2009.

[52] D. Hush, N. Ahmed, R. David, S. D. Stearns, "An adaptive IIR structure for sinusoidal enhancement, frequency estimation, and detection", Acous-

tics Speech and Signal Processing, IEEE Transactions on, vol.ASSP-34, pp. 1380–1390, Dec. 1986.

[53] M. H. Hayes, "Statistical Digital Signal Processing and Modeling", Wiley, 1996, ISBN 0-471 59431-8.

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

[55] N. I. Cho, S. U. Lee, "On the adaptive lattice notch filter for the detection of sinusoids", Circuits and Systems II: Analog and Digital Signal Processing, IEEE Transactions on, vol.40, no.7, pp. 405–416, July 1993.

[56] K. M. Knill, A. G. Constantinides, "Least-mean square adaptation of the orthogonal block adaptive line enhancer", Signals Systems and Computers, Conference Record of The Twenty-Seventh Asilomar Conference on, pp. 663–667 vol.1, Nov. 1993.

[57] A. Mvuma, S. Nishimura, T. Hinamoto, "Adaptive optimization of notch bandwidth of an IIR filter used to suppress narrow-band interference", Circuits and Systems, ISCAS IEEE International Symposium on, vol.5, pp. V-341–V-344, 2002.

[58] A. Mvuma, S. Nishimura, T. Hinamoto, "Adaptive IIR notch filter with controlled bandwidth for narrow-band interference suppression in DS CDMA system", Circuits and Systems, ISCAS Proceedings of the 2003 International Symposium on, vol.4, pp. IV-361–IV-364, May 2003.

[59] R. Punchalard, W. Lertvasana, P. Chumchu, "Convergence speed improvement for a variable step-size plain gradient algorithm by using variable notch bandwidth technique", Image and Signal Processing and Analysis, ISPA Proceedings of the 3rd International Symposium on, vol.2, pp. 788–792, Sept. 2003.

[60] S. Kirkpatrick, C.D. Gellat Jr., M.P. Vecchi, "Optimization by Simulated Annealing", Science, Vol. 220, No. 4598, pp. 671–680, 1983.

[61] H.P. Schwefel, "Numerical Optimization of Computer Models", John Wiley, Chichester, 1981.

[62] D. B. Fogel, "System Identification through Simulated Evolution: A Machine Learning Approach to Modeling", Ginn Press, Needham Heights, MA 02194, 1991.

[63] L. J. Fogel, A.J. Owens, M.J. Walsh, "Artificial Intelligence through Simulated Evolution, John Wiley & Sons, New York, 1966.

[64] Diagram showing the concept of a problem using evolutionary algorithms, *http://www.geatbx.com/docu/algindex.html*, accessed 13th June 2014.

[65] J. H. Holland, "Adaptation in natural and artificial systems", The University of Michigan Press, 1975.

[66] T. C. Fogarty, "Using the genetic algorithm to adapt intelligent systems", Symbols Versus Neurons, IEE Colloquium on, pp. 4/1–4/4, Oct 1990.

[67] K. S. Tang, K .F Man, S. Kwong, Q. He, "Genetic algorithms and their applications", Signal Processing Magazine, IEEE, vol.13, no.6, pp. 22–37, Nov. 1996.

[68] V. Genovese, P. Dario, R. Magni, L. Odetti, "Self Organizing Behaviour And Swarm Intelligence In A Pack Of Mobile Miniature Robots In Search Of Pollutants", Intelligent Robots and Systems, Proceedings of the 1992 lEEE/RSJ International Conference on, vol.3, pp. 1575–1582, July 1992.

[69] J. Kennedy, R. Eberhart, "Particle swarm optimization", Neural Networks, 1995. Proceedings., IEEE International Conference on, vol.4, pp. 1942–1948 vol.4, Nov.-Dec. 1995.

[70] The concept of particle swarm optimisation, *http://wirelesstechthoughts.blogspot.co.uk/2013/06/an-introduction-to-particle-swarm.html*, accessed 22nd June 2014.

[71] C. Benjangkaprasert, S. Phuvasitkul, W. Limwong, K. Janchitrapongvej, "Fast convergence algorithm for adaptive IIR notch filter using combination of genetic search and variable step-size algorithm", Intelligent Transportation Systems, Proceedings. 2003 IEEE, vol.2, pp. 948–952, Oct. 2003.

[72] C. Benjangkaprasert, S. Jorphochaudom, S. Chompoo, O. Sangaroon, K. Janchitrapongvej, "The combination of genetic algorithm and variable step-size algorithm for adaptive IIR notch filter", Neural Networks and Signal Processing, Proceedings of the 2003 International Conference on, vol.1, pp. 480–483, Dec. 2003.

[73] S. Jorphochaudom, C. Benjangkaprasert, O. Sangaroon, K. Janchitrapongvej, "Hybrid algorithm for adaptive IIR notch filter", Control, Automation and Systems, ICCAS 2008. International Conference on, pp. 2454–2458, Oct. 2008.

[74] D. J. Krusienski, W. K. Jenkins, "A modified particle swarm optimization algorithm for adaptive filtering", Circuits and Systems, ISCAS 2006. Proceedings. 2006 IEEE International Symposium on, pp. 137–140, May 2006.

[75] H. Shi, J. Zhang, "Research and application of the particle swarm optimization in adaptive notch filter design", Communications, Circuits and Systems, ICCCAS 2009. International Conference on, pp. 446–449, July 2009.

[76] B. Biswal, P. K. Dash, K. B. Panigrahi, "Power Quality Disturbance Classification Using Fuzzy C-Means Algorithm and Adaptive Particle Swarm Optimization", Industrial Electronics, IEEE Transactions on, vol.56, no.1, pp. 212–220, Jan. 2009.

[77] G. D. Cain, A. Yardim, J. Brun, B. Summers, "Real-time IIR notch filtering using Darwinian adaption", Circuits and Systems, IEEE International Sympoisum on, pp. 432–435 vol.1, June 1991.

[78] G. D. Cain, A. Yardim, L. Allen, "Dynamic convergence behaviour in Darwinian adaptive multi-notch filtering", New Directions in Adaptive Signal Processing, IEE Colloquium on, pp. 8/1–8/6, Feb. 1993.

[79] Z.-L. Gaing; R.-F. Chang, "Optimal PID controller for high-speed rail pantograph system with notch filter", TENCON 2009 - 2009 IEEE Region 10 Conference, pp. 1–6, Jan. 2009.

[80] J. L. Fernandez-Martinez, E. Garcia-Gonzalo, "Stochastic Stability Analysis of the Linear Continuous and Discrete PSO Models", Evolutionary Computation, IEEE Transactions on, vol.15, no.3, pp. 405–423, June 2011.

[81] E. C. Laskari, K. E. Parsopoulos, M. N. Vrahatis, "Particle swarm optimization for minimax problems", Evolutionary Computation, 2002. CEC '02. Proceedings of the 2002 Congress on, vol.2, pp. 1576–1581, 2002.

[82] A. Tarighat, A. H. Sayed, "Least mean-phase adaptive filters with application to communications systems", Signal Processing Letters, IEEE, vol.11, no.2, pp. 220–223, Feb. 2004.

[83] T. Samamäki, T.-H. Yu, S. K. Mitra, "Very Low Sensitivity Realization of IIR Digital Filter Using a Cascade of Complex All-Pass Structures", Circuits and Systems, IEEE Transactions on, vol.CAS-34, no.8, pp. 876–886, Aug. 1987.

[84] A. Ahmed, Y. Hu, P. Pillai, "3M relationship pattern for detection and estimation of unknown frequencies for unknown number of sinusoids based on Eigenspace analysis of Hankel matrix", Intelligent Signal Processing Conference 2013 (ISP 2013), IET, London, pp. 1–6, Dec. 2013.

[85] J. Kang, "A Newton-type iterative learning algorithm of output tracking control for uncertain nonlinear distributed parameter systems", Control Conference (CCC), 2014 33rd Chinese, pp. 8901–8905, July 2014.

[86] T. Kwan, K. Martin, "An adaptive analog continuous-time CMOS biquadratic filter", Solid-State Circuits, IEEE Journal of, vol.26, no.6, pp. 859–867, June 1991.

[87] D. Wulich, E. I. Plotkin, M. N. S. Swamy, W. Tong, "PLL synchronized time-varying constrained notch filter for retrieving a weak multiple sine signal jammed by FM interference", Signal Processing, IEEE Transactions on, vol.40, no.11, pp. 2866–2870, Nov. 1992.

[88] C. C. Ko, C. P. Li, "An adaptive IIR structure for the separation, enhancement, and tracking of multiple sinusoids", Signal Processing, IEEE Transactions on, vol.42, no.10, pp. 2832–2834, Oct. 1994.

[89] T. Linder, H. Zojer, B. Seger, "Fully analogue LMS adaptive notch filter in BICMOS technology", Solid-State Circuits, IEEE Journal of, vol.31, no.1, pp. 61–69, Jan. 1996.

[90] M. A. Soderstrand, T. G. Johnson, R. H. Strandberg, H. H. Loomis, K. V. Rangarao, "Suppression of multiple narrow-band interference using real-time adaptive notch filters", Circuits and Systems II: Analog and Digital Signal Processing, IEEE Transactions on, vol.44, no.3, pp. 217–225, March 1997.

[91] M. Karimi-Ghartemani, M. R. Iravani, "A nonlinear adaptive filter for online signal analysis in power systems: applications", Power Delivery, IEEE Transactions on, vol.17, no.2, pp. 617–622, April 2002.

[92] M. Mojiri, A. R. Bakhshai, "An adaptive notch filter for frequency estimation of a periodic signal", Automatic Control, IEEE Transactions on, vol.49, no.2, pp. 314–318, Feb. 2004.

[93] Y.C. Lim, Y. X. Zou, N. Zheng, "A piloted adaptive notch filter", Signal Processing, IEEE Transactions on, vol.53, no.4, pp. 1310–1323, April 2005.

[94] M. Mojiri, M. Karimi-Ghartemani, A. Bakhshai, "Estimation of Power System Frequency Using an Adaptive Notch Filter", Instrumentation and Measurement, IEEE Transactions on, vol.56, no.6, pp. 2470–2477, Dec. 2007.

[95] M. Mojiri, A. R. Bakhshai, "Estimation of n Frequencies Using Adaptive Notch Filter", Circuits and Systems II: Express Briefs, IEEE Transactions on, vol.54, no.4, pp. 338–342, April 2007.

[96] D. Yazdani, A. Bakhshai, G. Joos, M. Mojiri, "A Nonlinear Adaptive Synchronization Technique for Grid-Connected Distributed Energy Sources", Power Electronics, IEEE Transactions on, vol.23, no.4, pp. 2181–2186, July 2008.

[97] L. Ljung, "Analysis of recursive stochastic algorithms", Automatic Control, IEEE Transactions on, vol.22, no.4, pp. 551–575, Aug. 1977.

[98] Y. Xiao, Y. Takeshita, K. Shida, "Steady-state analysis of a plain gradient algorithm for a second-order adaptive IIR notch filter with constrained poles and zeros", Circuits Syst. II, Analog Digit. Signal Process., IEEE Transactions on, vol.48, no.7, pp. 733-740, July 2001.

[99] Y. Xiao, Y. Takeshita, K. Shida, "Tracking properties of a gradient-based second-order adaptive IIR notch filter with constrained poles and zeros", IEEE Transactions on Signal Process., vol.50, no.4, pp. 878-888, April 2002.

[100] Y. Xiao, R. K. Ward, A. Ikuta, "Performance analysis of the sign algorithm for a constrained adaptive IIR notch filter", IEEE Transactions on Signal Process., vol.51, no.7, pp. 1846–1858, July 2003.

[101] Y. Xiao, L. Ma, K. Khorasani, A. Ikuta, "Statistical Performance of the Memoryless Nonlinear Gradient Algorithm for the Constrained Adaptive IIR

Notch Filter", Circuits and Systems I: Regular Papers, IEEE Transactions on, vol.52, no.8, pp. 1691–1702, Aug. 2005.

[102] J. Zhou, G. Li, M. Sun, "A Structural View of Stability in Adaptive IIR Filters", Signal Processing, IEEE Transactions on, vol.54, no.12, pp. 4828–4835, Dec. 2006.

[103] M. Niedźwiecki, M. Meller, "New Algorithms for Adaptive Notch Smoothing", Signal Processing, IEEE Transactions on, vol.59, no.5, pp. 2024–2037, May 2011.

[104] G. Strang, "Linear Algebra Lectures", Massachusetts Institute of Technology, *http://ocw.mit.edu/courses/mathematics/18-06-linear-algebra-spring-2010/video-lectures/*, accessed June 2010.

[105] G. Strang, "Linear Algebra and its Applications", Fourth Edition, Brooks/Cole, ISBN-10: 0-03-010567-6, 2006.

[106] S. K. Mitra, "Digital Signal Processing a Computer-Based Approach", Third Edition, McGraw-Hill, ISBN 007-125579-6, 2006.